

Identification and Attenuation of Multiple Reflections with Neural Networks

Identifizierung und Unterdrückung von Multiplen Reflexionen mit Neuronalen Netzen

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

der Fakultät für Physik der
Universität Karlsruhe (TH)
genehmigte

DISSERTATION

von

Dipl.-Geophys. Robert Essenreiter
aus Isny im Allgäu

Tag der mündlichen Prüfung:

9. Juli 1999

Referent:

Prof. Dr. Peter Hubral

Korreferent:

Prof. Dr. Friedemann Wenzel

Contents

| | |
|---|-----------|
| Zusammenfassung | I |
| 1 Introduction: Primaries and Multiples | 1 |
| 1.1 The Problem of Multiple Reflections | 1 |
| 1.2 Multiple Attenuation Methods | 4 |
| 1.3 Multiple Attenuation with Neural Networks | 8 |
| 2 Neural Networks | 11 |
| 2.1 Backpropagation Neural Networks | 11 |
| 2.1.1 Neural Networks as Non-Linear Filters: Theory | 11 |
| 2.1.2 The Multi-layer Perceptron | 15 |
| 2.1.3 Practical Aspects of Backpropagation | 20 |
| 2.2 Self-Organizing Maps (SOM) | 22 |
| 2.2.1 Principles of Self-Organizing Maps: Theory | 24 |
| 2.2.2 SOM Application Example | 31 |
| 3 Seismic Attributes | 35 |
| 3.1 Instantaneous Attributes | 36 |
| 3.2 Wavelet Attributes | 39 |
| 3.3 Wavefront Parameters | 39 |
| 4 Multiple Prediction and Attenuation with Backpropagation Neural Networks | 43 |
| 4.1 Trace by Trace Multiple Attenuation | 43 |
| 4.2 Multiple Attenuation with Neural Net Ensembles | 48 |
| 4.3 Conclusions | 51 |

| | | |
|----------|---|------------|
| 5 | Attribute based Multiple Prediction and Attenuation with Backpropagation | 53 |
| 5.1 | The Method | 53 |
| 5.2 | Multiple Attenuation in a Simple 1-D Elastic Data Set | 54 |
| 5.3 | Multiple Attenuation and Prediction in a 1-D Realistic Elastic Data Set . . | 57 |
| 5.3.1 | Multiple Attenuation | 57 |
| 5.3.2 | Multiple Prediction | 58 |
| 5.4 | Multiple Prediction in a Small 2-D Acoustic Data Set | 60 |
| 5.4.1 | Target-oriented Multiple Prediction | 61 |
| 5.5 | Multiple Prediction in a Full 2-D Acoustic Data Set | 62 |
| 5.6 | Conclusions | 63 |
| 6 | Attribute based Multiple Identification with Self-Organizing Maps | 65 |
| 6.1 | The Method | 65 |
| 6.2 | Application to Synthetic Data | 68 |
| 6.2.1 | Correlation Analysis | 72 |
| 6.2.2 | Results for Instantaneous Attributes | 74 |
| 6.2.3 | Results for Wavelet Attributes | 76 |
| 6.2.4 | Component Plane Analysis for Wavelet Attributes | 78 |
| 6.2.5 | Results for Wavelet and Wavefront Attributes | 80 |
| 6.3 | Application to Real Data | 87 |
| 6.4 | Conclusions | 90 |
| 7 | A Layer-stripping approach | 91 |
| 7.1 | The Method | 91 |
| 7.2 | Synthetic Data Application | 95 |
| 7.3 | Conclusions | 97 |
| | Conclusions | 99 |
| | Bibliography | 103 |

Zusammenfassung

Das Multiplenproblem

Multiple Reflexionen in der Seismik sind ein bestens bekanntes Problem, da sie im Seismogramm einen Typ von Signalen darstellen, der durch seine starke Energie andere Signalanteile, die Primärreflexionen, überlagert und oft zum Verschwinden bringt. Trotzdem ist dieses Problem bis heute nicht zufriedenstellend gelöst, und die Erdölindustrie unternimmt große Anstrengungen, um Verfahren zu entwickeln, die wenigstens bei bestimmten geologischen Gegebenheiten ein verbessertes Abbild des Untergrundes erzeugen können. Eine Primärreflexion ist definitionsgemäß die Antwort auf eine Wellenfront, die nach nur einmaliger Reflexion an einer Inhomogenität registriert wird. Die Wellenausbreitung wird in diesem Zusammenhang im allgemeinen Fall durch die elastodynamische, inhomogene, anisotrope Wellengleichung (Aki and Richards, 1980) mit bestimmten Randbedingungen beschrieben:

$$(c_{ijkl} u_{k,l})_{,j} + f_i = \rho \ddot{u}_i, \quad i = 1, 2, 3, \quad (1)$$

wobei u_k die Komponenten des Verschiebungsvektors darstellen; $(\)_{,j}$ ist die räumliche Ableitung in j -Richtung, $u_{k,l}$ ist die räumliche Ableitung der k -ten Komponente von u in l -Richtung (Summenkonvention), \ddot{u}_i ist die zweite Ableitung der i -ten Komponente nach der Zeit. c_{ijkl} sind die Komponenten des Elastizitätstensors, f die Quellfunktion und ρ die Dichte des Mediums. Eine multiple Reflexion ist demnach die Antwort auf eine Wellenfront, die mehrmals an Diskontinuitäten reflektiert wurde, bevor sie von einem Empfänger aufgezeichnet wird.

Heutzutage wird der Großteil der Öl- und Gasvorräte vor den Küsten und in den kontinentalen Schelfgebieten im relativ flachen Wasser (50 m - 300 m) entdeckt und gefördert.

Allerdings geht der Trend zur Ressourcengewinnung in tieferem Wasser, da eine ausgereifte Technik das erlaubt, und die fortschreitende Ausbeutung in anderen Gebieten dies notwendig macht. Bevor jedoch ein öl- oder gasproduzierendes Bohrloch abgeteuft werden kann, müssen Geophysiker ein Abbild der physikalischen Eigenschaften (Gesteinsporositäten und -permeabilitäten) des Untergrundes liefern, das anzeigt, wo ein Reservoir erwartet werden kann.

In der marinen Exploration stossen wir oft auf das Problem, daß sich das Meer wie ein Wellenleiter verhält (Backus, 1959), in dem die seismischen Wellen mehrfach zwischen der Meeresoberfläche und dem Meeresboden hin- und herreflektiert werden (siehe Abb. 1). Dies liegt daran, daß die Wasseroberfläche ein Reflektor mit sehr hohem Impedanzkontrast ist, an dem ein sehr großer Teil der Wellenenergie wieder nach unten zurückreflektiert wird, und auch der Meeresboden oft einen relativ starken Impedanzkontrast aufweist. Wellen, die durch den Meeresgrund transmittiert werden, können dann auch zwischen tieferen Schichtgrenzen reverberieren. Diese Art von Multiplen bezeichnet man als interne Multiple, wohingegen Multiple, die mindestens einmal von der Wasseroberfläche nach unten reflektiert wurden als Freie-Oberflächen-Multiplen bezeichnet werden. Generell wird auch unterschieden zwischen Multiplen 1. Ordnung, 2. Ordnung, usw., wobei die Ordnung die Anzahl der Rückstreuungen der Welle von einer Diskontinuität nach unten angibt (siehe Abb. 2, rechts). Die Energie dieser internen Multiplen und besonders der Wasserreverberationen

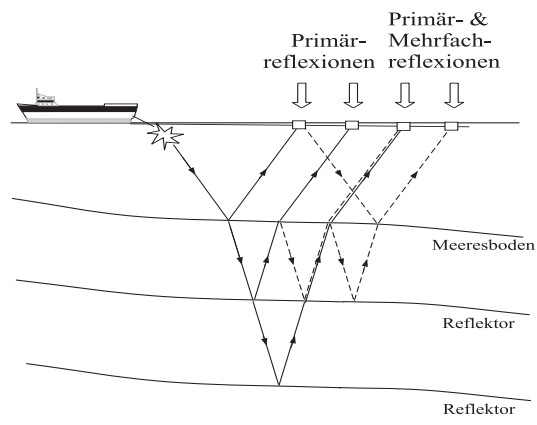


Abb. 1: Das Multiplenproblem: Eine Welle, die von einer Quelle zu den Empfängern propagiert, wird entweder nur einmal gestreut - das sind die Primärreflexionen, deren Laufwege mit einer durchgezogenen Linie dargestellt sind, oder es kommt zu Mehrfachreflexionen an den Schichtgrenzen. Die Laufwege dieser Multiplen sind als gestrichelte Linien dargestellt. In isotropen Medien sind die Strahlen auf der propagierenden Wellenfront senkrecht stehende Trajektorien.

kann so stark werden, daß Primärreflexionen von tieferen Reflektoren, auch von Schichtgrenzen, die potentielle Öl- und/oder Gasfallen darstellen, komplett verdeckt werden, was zu falschen Interpretationen der Messungen führen kann.

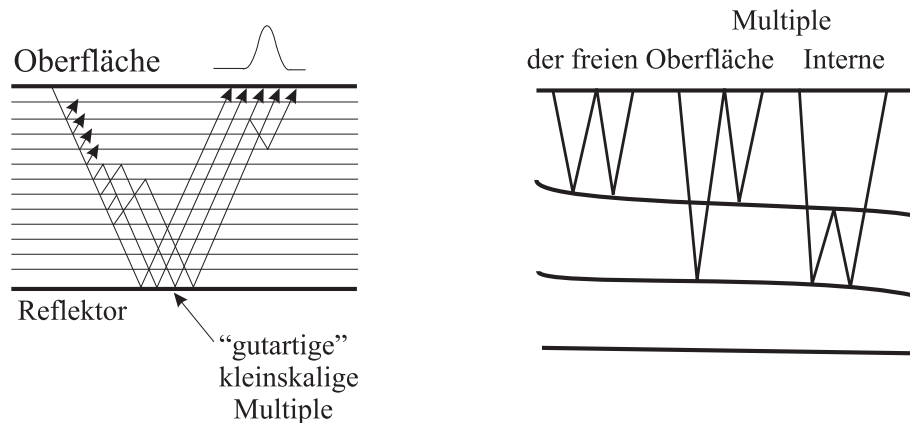


Abb. 2: Links: Kurzperiodische Multiple. Wellenanteile, die zwischen Schichtgrenzen hin- und herreflektiert werden, deren Abstand größer ist als eine Wellenlänge, werden als Störsignal betrachtet. Die kleinskaligen Reverberationen zwischen dünnen Schichten dagegen sind für das Entstehen der Primärreflexionen verantwortlich. Rechts: Die verschiedenen Arten von multiplen Reflexionen: 1) Multiplen der freien Oberfläche, wie z.B. Wasserreverberationen (links), Peglegmultiple (mitte), und 2) interne Multiple (rechts).

Um einen gewünschten Reflektor, der unter Umständen eine Lagerstätte anzeigen könnte, richtig zu lokalisieren, müssen diese interferierenden Multiplen unterdrückt werden, oder, da dies nur selten vollständig möglich ist, zumindest abgeschwächt werden. Ein wichtiger Punkt hierbei ist jedoch, daß das Primärsignal nicht beschädigt oder gar mitunterdrückt wird. Das wird zu einer Herausforderung, wenn Multiple und Primäre direkt interferieren, also an sämtlichen Empfängern etwa zur selben Zeit ankommen, d.h. ihre Laufzeitkurven weisen über die ganze Empfängerauslage hinweg nur wenig Abweichung voneinander auf. Dann ist es oft schwer zu entscheiden, ob ein Primärsignal oder ein multiples Störsignal vorliegt, oder eine Überlagerung von beiden.

Im Falle von Multiplen spielt jedoch die Skala, in der diese auftreten, eine große Rolle. Wir wollen nicht *alle* Arten von Multiplen unterdrücken: Es gibt Multiple, die als Störsignal betrachtet werden, aber ebenso gibt es "gutartige" Multiple. Als störend gelten langperiodische Multiple. Das sind diejenigen Multiplen, die zwischen Materialdiskontinuitäten hin- und herreflektiert werden, deren Abstand größer als eine Wellenlänge ist. Die gutartigen Multiplen hingegen sind diejenigen, ohne die gar kein Primärsignal an

der Oberfläche ankommen würde. Diese Multiplenart ist extrem kurzperiodisch und reverberiert zwischen dünngeschichteten Strukturen, aus denen im Prinzip jeder, für die Erdölexploration interessante, sedimentäre, geologische Untergrund aufgebaut ist (siehe Abb. 2, links). Die nach oben zurückgestreuten Wellenfelder vieler dieser kleinskaligen Reverberationen summieren sich kohärent auf. Dadurch wird durch konstruktive Interferenz genügend Energie gesammelt und wieder nach oben zurückreflektiert, um an der Oberfläche registriert werden zu können (Shapiro and Hubral, 1999; O'Doherty and Anstey, 1971).

In geophysikalischen Darstellungen werden Laufwege von Wellen oft als Strahlen dargestellt. Diese Strahlen stehen für isotrope Medien senkrecht auf der jeweiligen Wellenfront der sich ausbreitenden Elementarwelle. Das legt nahe, daß das Problem der Trennung von Primär- und multiplen Reflexionen mit Methoden aus der geometrischen Optik angegangen wird und formell mit Hilfe der Wellentheorie beschrieben werden kann.

Die Ausbreitung von Wellen in elastischen, inhomogenen, anisotropen Medien wird durch die elastodynamische Wellengleichung (Gleichung 1) beschrieben. Ist der Quellterm f in dieser Gleichung eine zeitliche und räumliche Impulsfunktion, dann ist die Lösung eine spezielle Funktion, die Green'sche Funktion, die auch als Impulsantwortfunktion oder Übertragungsfunktion bezeichnet wird. Für Probleme mit konstanten Koeffizienten kann ein analytischer Ausdruck für die Green'sche Funktion für isotrope, homogene Medien gefunden werden. Im allgemeinen Fall ist dies allerdings nicht möglich, und die Green'sche Funktion kann nur näherungsweise bestimmt werden, zum Beispiel mit Hilfe der Strahlentheorie.

Eine Welle, die sich durch ein inhomogenes Medium ausbreitet, wird an Streuzentren gestreut und erzeugt ein gestreutes Wellenfeld. Wird das Wellenfeld nur einmal zwischen Quelle und Empfänger gestreut (oder wie oben erwähnt mehrfach zwischen dünnen Schichten), wird dieser Teil des Wellenfeldes als Primäranteil bezeichnet. Sobald das Wellenfeld aber von einem zweiten Streuzentrum gestreut wird, das weiter als eine Wellenlänge entfernt ist, gehört es zum multiplen Wellenfeld. Das Problem, das primäre vom multiplen Wellenfeld zu trennen, läuft letztendlich auf die Aufspaltung der Green'schen Funktion hinaus, die die Wellengleichung (Gleichung 1) für eine impulsförmige Anregungsfunktion löst. Dabei wird die Greensche Funktion in einen Anteil für das primäre Wellenfeld G_P , einen Anteil für das multiple Wellenfeld G_M , und einen Anteil

G_R für den Teil des Wellenfelds, der weder dem primären noch dem multiplen Energieanteil zugeordnet werden kann, aufgespalten:

$$G = G_P + G_M + G_R. \quad (2)$$

Die Green'sche Funktion für das multiple Wellenfeld G_M kann noch weiter aufgespalten werden in Green'sche Funktionen der Teilwellenfelder von Multiplen der freien Oberfläche und internen Multiplen:

$$G_M = G_F + G_I, \quad (3)$$

wobei G_F für den Teil der Green'schen Funktion für die Multiplen der freien Oberfläche steht, und G_I die Green'sche Funktion für die internen Multiplen darstellt. Natürlich ist diese Trennung der Green'schen Funktion in verschiedene Anteile auf analytischem Wege nur in besonderen Fällen mit einfacher Geometrie, konstanten Koeffizienten, und in homogenen, isotropen und unbeschränkten Medien möglich.

Im ersten Kapitel werden sieben bestehende Ansätze zur Lösung des Problems der Herausstrennung spezieller multipler Streuanteile im Wellenfeld beschrieben. Alle diese Methoden haben eines gemeinsam: Sie funktionieren nur in Spezialfällen. Die Grundlage jeder dieser Methoden ist ein physikalisches Modell, das in der Realität nur begrenzte Gültigkeit besitzt. Diese hängt von der Geometrie im Untergrund, den Messparametern und den elastischen Eigenschaften des Mediums ab. Nimmt man das physikalische Modell als bekannt an, so bedeutet das, die modellbeschreibenden Parameter exakt genug zu kennen, was nicht immer der Fall ist. Oft sind die Modelle zu grob und/oder zu stark vereinfacht, so daß die darauf basierenden Methoden versagen.

Methoden zur Multiplenunterdrückung

Traditionelle Methoden zur Unterdrückung oder Dämpfung von multiplen Reflexionen in seismischen Daten können grob in zwei Klassen eingeteilt werden: Die erste Klasse von Verfahren versucht, eine physikalische Eigenschaft auszunutzen, mit deren Hilfe sich Primäre und Multiple unterscheiden lassen. Es handelt sich dabei meist um Filtermethoden, wie zum Beispiel das Verfahren der "predictive deconvolution" mit Wiener-Filtern (Robinson and Treitel, 1980; Peacock and Treitel, 1969). Dabei wird ein Optimalfilter entworfen, das die Multiplen vorhersagt und anschließend vom Signal subtrahiert. Es

wird davon ausgegangen, daß der Multiplenanteil im Signal streng periodisch ist, was allerdings nur für kleine Quelle-Empfänger-Entfernungen gilt. Auch stößt dieses Verfahren schnell an seine Grenzen, sobald das von der Quelle ausgesandte Signal seine Amplitude, seinen Frequenzgehalt oder seine Phasencharakteristik verändert.

Um eine simulierte “zero-offset” Sektion zu erhalten, die schon ein ungefähres Abbild vom tatsächlichen Untergrund darstellt, werden die seismischen Daten gestapelt, d.h. es wird entlang von Laufzeithyperbeln im “common-midpoint (CMP) gather” aufsummiert. Die Krümmung dieser Hyperbeln wird von einem Makrogeschwindigkeitsmodell bestimmt, das zum Beispiel aus einer Geschwindigkeitsanalyse gewonnen wurde. Stimmt dieses Geschwindigkeitsmodell genügend gut mit der Realität überein, dann werden Primärsignale kohärent aufsummiert, sofern die Laufzeitfunktion hyperbolisch sind. Multiple weisen meist geringere Stapelgeschwindigkeiten auf als Primäre, besitzen daher stärker gekrümmte Laufzeithyperbeln und werden daher in der gestapelten Sektion unterdrückt (Schneider et al., 1965).

Die zweite Klasse von Multiplenunterdrückungsmethoden basiert auf der Extrapolation des an der Oberfläche gemessenen Wellenfeldes zurück in die Tiefe auf der Grundlage der Wellentheorie. Das multiple Wellenfeld wird vorhergesagt und anschließend vom gesamten Wellenfeld abgezogen. Die Daten werden zum Beispiel einmal durch die Wasserschicht hinunter und wieder nach oben propagiert. Dies lässt aus den Primären Multiplen erster Ordnung werden, die dann vom ursprünglichen Wellenfeld mit Hilfe eines adaptiven Subtraktionsverfahrens abgezogen werden können. Dies kann zu Fehlern führen, wenn die Adaption des zu subtrahierenden Signals an das Originalsignal nicht optimal funktioniert. Außerdem müssen teilweise Meeresbodentopologie, Oberflächengeschwindigkeiten, oder die Quellfunktion f bekannt sein oder es müssen Daten zwischen den bestehenden Empfängern interpoliert werden.

Alle diese Annahmen und Parameter, die oft nur ungenau abgeschätzt werden können beeinträchtigen die Anwendungsmöglichkeiten dieser Methoden beträchtlich und limitieren ihren Einsatz auf Sonderfälle, in denen das physikalische Modell eng genug mit der Realität übereinstimmt. Oft trifft dies aber schon nicht mehr zu, wenn zum Beispiel die Daten verrauscht sind. Selbst eine Kombination dieser Verfahren würde dieses Problem nicht lösen. Deshalb wird in dieser Dissertation ein Verfahren vorgestellt, das von

einem völlig anderen und neuen Ansatz ausgeht: die Unterdrückung von multiplen Reflexionen mit Neuronalen Netzen.

Multiplenunterdrückung mit Neuronalen Netzen

In der Realität sind die physikalischen Gesetze, die das Problem der multiplen Reflexionen bestimmen, nicht in dem Maße bekannt, daß sich robuste und universell einsetzbare Algorithmen zu dessen Lösung entwickeln liessen. Wir brauchen eine Methode, die zum Teil auf einem physikalischen Modell beruht, aber trotzdem adaptiv ist und die Möglichkeit gibt, Nebenbedingungen mit einfließen zu lassen. Das Neuronale Netz stellt ein attraktives Konzept dar, das alle diese Eigenschaften besitzt.

Eine weitere Motivation besteht darin, daß es im Gegensatz zu den herkömmlichen Verfahren zur Multiplenunterdrückung erstmals möglich wird, Kriterien zur Unterscheidung von Primärsignalen und Multiplen aus mehreren verschiedenen Parameterbereichen zu kombinieren. Jede der oben beschriebenen Methoden arbeitet in nur einem Parameterbereich: im Raum-Zeit-Bereich ($x - t$), im Frequenz-Wellenzahl-Bereich ($f - k$), im Zeit-Slowness-Bereich ($\tau - p$) oder im Geschwindigkeits-Zeit-Bereich ($v - t$). Dadurch werden auch nur Diskriminierungseigenschaften dieses einen Raumes benutzt, wie zum Beispiel Periodizität oder unterschiedliche Laufzeitdifferenzen an den Empfängern.

Idealerweise macht man sich aber alle diese Eigenschaften, die Multiple von Primären trennen können, gleichzeitig zu Nutze. Der Formalismus der Neuronalen Netze ist speziell dafür ausgelegt, aus möglichst vielen dieser physikalischen Attribute die gewünschte Information zu extrahieren.

Der *Backpropagation*-Algorithmus ist in der Lage, die physikalischen Eigenschaften des zugrundeliegenden Modells zu erlernen, indem es aus den Daten selber die Regeln und physikalischen Gesetze extrahiert. Am Anfang steht nur eine ungenaue ("fuzzy") Menge von Beispielpaaren, die Eingaben für das Problem und bekannte spezielle Lösungen für diese Eingaben beinhalten.

Die selbst-organisierende Karte (*self-organizing map*, SOM) analysiert die physikalischen Eigenschaften des Problems und versucht selber Regeln aufzustellen, mit denen das Problem charakterisiert werden kann. Als Eingabe braucht sie nur die Messwerte

selber, oder Attribute, die daraus berechnet wurden, und findet dann Zusammenhänge, Ähnlichkeiten und Gegensätze in den Daten. Das SOM bietet eine elegante Methode zur Klassifizierung und Clusterung von Messdaten.

Im Rahmen dieser Dissertation wurden vier verschiedene Verfahren zur Multiplenunterdrückung entwickelt, getestet und analysiert. Diese werden im Folgenden kurz beschrieben (siehe auch Abb. 1.3 in Kapitel 1).

Verfahren 1: Voraussage und Unterdrückung von Multiplen mit Backpropagation-Netzen (Kapitel 4)

Das Ziel dieses Verfahrens ist es, aus vorhandenen Bohrlochdaten, wie Geschwindigkeits- und Dichtelogs, die in einzelnen Bohrlöchern gemessen wurden, auf die Struktur des Untergrundes zwischen den Bohrlöchern zu schließen. Dies wird nicht getan, indem einfach zwischen den Bohrlöchern interpoliert wird. Die gewünschte Information wird aus seismischen Daten extrahiert, die über das ganze relevante Gebiet hinweg gemessen wurden. Dabei beschränke ich mich auf das Erstellen eines multiplenenfreien Pre-stack Datensatzes, der dann mit der entsprechenden Technik gestackt (gestapelt) und migriert werden kann.

Kernstück dieser Methode ist ein Backpropagation Neuronales Netz, das so trainiert wird, daß es lernt, ein gemessenes seismisches Wellenfeld in das Primärfeld und das Multiplenenfeld aufzuspalten. Der Multiplenenanteil wird unterdrückt während der Primärenanteil ausgegeben wird. Das Backpropagation Netz braucht dazu Beispiele, aus denen es die physikalischen Gesetze lernen kann, die dann in den Netzkoeffizienten (den Gewichten) gespeichert werden. Diese Beispiele werden durch Modellierung von synthetischen seismischen Daten auf der Basis der Bohrlochdaten generiert. Mittels des Reflectivity-Verfahrens (Fuchs and Müller, 1971), das auf der elastischen Wellengleichung in 1-dimensionalen Medien beruht, werden Seismogramme modelliert, wobei die Möglichkeit besteht, die Rückstreuung an bestimmten Diskontinuitäten aus- bzw. einzuschalten. Dies wird durch Nullsetzen der entsprechenden Reflexionskoeffizienten in der Propagatormatrix erreicht (siehe Abschnitt 4.2). Damit lassen sich Seismogramme erstellen, die jede mögliche Kombination aus Primärsignalen, Multiplen der freien Oberfläche und internen Multiplen enthalten. Auf diese Weise wurden zwei synthetische Datensätze generiert: ein Datensatz,

der das volle Wellenfeld beinhaltet, und einer, der nur aus dem Primärfeld besteht. In das Neuronale Netz werden nun spurweise die Daten mit dem vollen Wellenfeld (Primäre und Multiple) eingegeben. Gleichzeitig präsentiert man dem Netz die jeweilige gewünschte Ausgabe: die zugehörige Spur aus dem Primärfeldseismogramm. So lernt das Neuronale Netz aus einer Anzahl von Beispielen, die gewünschten Primärreflexionen aus den Daten herauszupräparieren und die unerwünschten Multiplen zu unterdrücken.

Das erste Anwendungsbeispiel zeigt einen Vergleich der Unterdrückung von Multiplen mit dem Neuronalen Netz und der Methode der “predictive deconvolution” mittels Wiener-Filtern (Robinson and Treitel, 1980) anhand von Zero-Offset-Daten. In der zweiten, realistischeren Anwendung, wird ein Ensemble von Neuronalen Netzwerken verwendet, bei dem jedes Netz für einen Offset im Common-Midpoint-(CMP)-Gather zuständig ist. Im Vergleich zum Wiener-Filter, der auf sehr strengen Annahmen beruht, zeigt das Neuronale Netz, besonders in Anwesenheit von Rauschen, sehr robuste und gute Ergebnisse.

Verfahren 2: Voraussage und Unterdrückung von Multiplen mit Backpropagation-Netzen auf der Basis von Attributen (Kapitel 5)

Im Gegensatz zu Verfahren 1, das nur im Raum-Zeit-Bereich arbeitet, kommt in diesem Verfahren der oben erwähnte Ansatz der Verwendung von Attributen aus mehreren verschiedenen Parameterbereichen zum Einsatz. Problemspezifisch werden physikalische Attribute ausgewählt und für jedes potentielle Reflexionsereignis (Primäre und Multiple) berechnet. Je nachdem welche gewünschte Ausgabe man zum Training des Backpropagation-Netzes verwendet, hat man die Wahl, ob man Multiplen unterdrücken oder voraussagen will. Die vorausgesagten Multiplen kann man dann mit geeigneten Filtertechniken vom Originaldatensatz subtrahieren. Zur Multiplenvorhersage benötigt man eine Abschätzung der zu behandelnden Multiplen für einen bestimmten Teil der Seismogramme. Diese kann entweder durch Modellierung (1D und 2D) auf der Basis von Bohrlochdaten gewonnen werden, oder aber durch die Anwendung eines herkömmlichen Multiplenvorhersageverfahrens (vgl. Abschnitt 1.2). Für die Unterdrückung der Multiplen braucht man, wie in Verfahren 1, das Primärfeld aus Modellberechnungen an Bohrlochlokalationen.

Anhand von drei verschiedenen synthetischen Datenbeispielen wird sowohl das Verfahren zur Multiplenunterdrückung als auch zur Multiplenvorhersage demonstriert. Die Methode funktioniert sehr gut für die Vorhersage von Multiplen der freien Oberfläche und zur Multiplenunterdrückung. Mit den internen Multiplen hat sie, wie fast alle existierenden Verfahren, Probleme, da diese meist sehr schwache Signalamplituden aufweisen und deshalb sehr stark von Rauschen überlagert sind. Die hier vorgestellte Methode ist auch hervorragend für eine zielorientierte Anwendung geeignet: Man kann sich bestimmte Reflexionsereignisse herausuchen, die aus explorationstechnisch interessanten Tiefen kommen, in denen Lagerstätten vermutet werden. Es lässt sich dann untersuchen, ob es sich dabei wirklich um ein Primärsignal handelt oder nur um eine Multiple.

Verfahren 3: Identifizierung von Multiplen mit selbst-organisierenden Karten auf der Basis von Attributen (Kapitel 6)

Diese Methode ist für die praktische Anwendung das interessanteste Verfahren zur Trennung von Primärfeld und multipltem Wellenfeld. Im Gegensatz zu den Methoden, die auf den Backpropagation-Netzwerken beruhen, wird hier keine Abschätzung des gewünschten Ergebnisses an bestimmten Punkten benötigt. Mit Hilfe der selbst-organisierenden Karte können allein aus dem aufgezeichneten Wellenfeld, bzw. aus den daraus berechneten physikalischen Attributen, die einzelnen Signale getrennt und klassifiziert werden. Dazu werden in einem seismischen Datensatz unter Zuhilfenahme der Geschwindigkeitsanalyse (siehe Abschnitt 3.3) alle Reflexionsereignisse (Primäre und Multiple) automatisch gepickt. Entlang den hyperbolischen Laufzeitkurven dieser Ereignisse werden im CMP-Gather eine Anzahl von Attributen berechnet (siehe Kapitel 3). Mittels einer Korrelationsanalyse wird entschieden, welche der Attribute aussagekräftig genug sind und welche redundante Information beinhalten und deswegen weggelassen werden können. Nachdem die so gewonnenen Daten normalisiert worden sind, wird eine selbst-organisierende Karte (Abschnitt 2.2) damit trainiert. Diese führt eine Clusterung der Eingabedaten, also der Reflexionsereignisse, durch. Auf einer zwei- oder dreidimensionalen Karte werden die Daten so abgebildet, daß Abhängigkeiten innerhalb der Daten sichtbar werden. Aus der Auswertung dieser Topologie-erhaltenden Abbildung gewinnt man dann schließlich eine farbcodierte Zero-Offset Sektion, in der die einzelnen Teilfelder des Wellenfeldes unterschiedliche Farben aufweisen. Auf diese Weise ist sowohl eine qualitative als auch

eine quantitative Trennung von Primären und Multiplen möglich.

Verfahren 4: Unterdrückung von Multiplen mit einem Schicht-für-Schicht-Verfahren (Kapitel 7)

Diese Methode stellt eine sehr praktische Anwendung eines überwachten Lernverfahrens dar, bei dem dieselben Eingabedaten wie in Verfahren 3 verwendet werden, diese aber nicht mit einer selbst-organisierenden Karte, sondern mit einem Backpropagation-Netz prozessiert werden. Der Datensatz besteht also aus einer Reihe von Attributen, die aus einem gemessenen Seismogramm berechnet wurden und aus wenigen Signaleinsätzen, die von einem erfahrenen Interpreten identifiziert wurden. Dabei reicht es aus, Reflexionsereignisse aus geringen Tiefen zu verwenden, da diese meistens sehr verlässlich zu bestimmen sind. Diese sollten jedoch sowohl Primär- als auch Mehrfachreflexionen beinhalten. Der Algorithmus arbeitet sich dann Schicht für Schicht von oben nach unten durch. Dazu wird ein Neuronales Netz mit den ausgewählten Ereignissen trainiert und dann mit allen Reflexionsereignissen getestet. Alle Einsätze, die innerhalb eines bestimmten Vertrauensintervalls entweder als Primäre oder als Multiple identifiziert wurden, werden zusätzlich für den nächsten Trainingslauf verwendet. Dabei kann der Benutzer jederzeit eingreifen, wenn er meint, daß ein Signal falsch klassifiziert worden ist. Die Prozedur wird solange wiederholt, bis alle Reflexionsereignisse identifiziert worden sind.

Alle vier Verfahren demonstrieren die Anwendbarkeit von Neuronalen Netzen, sowohl überwachter als auch unüberwachter Lernalgorithmen, für die Identifizierung und Unterdrückung von multiplen Reflexionen. Je nach Vorhandensein von Zusatzinformationen aus Bohrlochdaten ist das eine oder das andere Verfahren vorzuziehen. Können aus mehreren Geschwindigkeits- und Dichtelogs zuverlässige Modellierungen durchgeführt werden, so liefern die überwachten Verfahren (Verfahren 1 & 2: Voraussage und Unterdrückung von Multiplen mit Backpropagation-Netzen) sehr zuverlässige Abschätzungen des primären beziehungsweise des multiplen Wellenfeldes. Sind keine Bohrlochdaten vorhanden, bietet das unüberwachte Verfahren (Verfahren 3: Identifizierung von Multiplen mit selbst-organisierenden Karten auf der Basis von Attributen) eine sehr effiziente Methode, Reflexionsereignisse zu klassifizieren und die Wellenfelder zu trennen. Dies

geschieht nur unter Verwendung der gemessenen seismischen Daten und einiger weniger interpretierter Ereignisse zur Etkettierung der Klassen. Auch das vierte Verfahren (Verfahren 4: Unterdrückung von Multiplen mit einem Schicht-für-Schicht-Verfahren) verfolgt diese Philosophie, jedoch unter Verwendung einer modifizierten, überwachten Lernmethode.

Chapter 1

Introduction: Primaries and Multiples

1.1 The Problem of Multiple Reflections

In seismic exploration the problem of multiple reflections contaminating seismograms and thus disguising important information about subsurface reflectors is well-known but not yet solved satisfactorily. Today, the majority of all oil and gas resources are discovered offshore in continental shelf areas in shallow water, although the trend is moving into deeper water. Before oil-producing wells can be drilled, geophysicists have to provide an image of the physical properties in the subsurface that shows where reservoirs can be expected. In a marine exploration setting we encounter the problem that the water layer often behaves as a wave trap (Backus, 1959), where seismic waves are multiply reflected between sea surface and sea bottom. Waves that are transmitted through the sea bottom can also reverberate between deeper reflectors. The energy of these interbed multiples and water layer reverberations can become so strong that the primary reflection arrivals of deeper target reflectors become completely invisible. As a result, marine seismograms often show a ringy character with strong multiples superposed on most of the primary arrivals from deeper reflectors.

For correctly locating a target reflector that might indicate an oil reservoir, these interfering multiple reflections have to be eliminated, or since this is only rarely possible, they have to be at least attenuated. Figure 1.1 shows a typical marine data acquisition method. Travel paths of primary reflections (solid lines) and several multiple reflections (dashed lines) illustrate the problem of interference of these two types

of arrivals. Especially when primary and multiple reflections arrive at the same zero-offset travel time and show little move-out difference, it is very hard to separate the two signals or even to tell if the event is a primary or a multiple.

An important distinction has to be made, however. We do not want to eliminate *all* types of multiples. There are *bad* multiples, but also *good* multiples. The bad ones are the long-period multiples depicted in Figure 1.1, which are reverberating between material discontinuities that are separated from each other by more than a wavelength. The good multiples are the ones, without which there would be no primary signal arriving at the surface. This type of multiple is extremely short-period and reverberates in the fine-layered structures which make up the subsurface (Shapiro and Hubral, 1999; O'Doherty and Anstey, 1971). The upward reflections of many of these small-scale reverberations sum up coherently, and thus enough of the wave energy is reflected back upward again to be recorded at the surface. The difference between good and bad multiples is shown in Figure 1.2.

The travel paths drawn in Figures 1.1 and 1.2 are those trajectories that are perpendicular to the corresponding wavefront of the propagating wave. This implies the treatment of the problem of separating primary and multiple reflections by means of techniques based on geometrical optics. However, in a general sense, without assuming a special geometry of scatterers forming reflectors, a formulation of the problem in terms of wave theory is possible as well.

The propagation of waves in elastic, inhomogeneous, anisotropic media follows the elastodynamic wave equation

$$(c_{ijkl} u_{k,l})_{,j} + f_i = \rho \ddot{u}_i, \quad i = 1, 2, 3 \quad , \quad (1.1)$$

where u_k are the components of the displacement vector, $(\)_{,j}$ the spatial derivative in

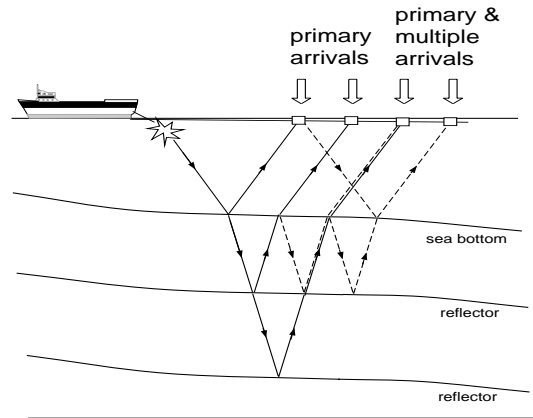


Figure 1.1: *Marine data acquisition: travel paths of the primary reflections (solid lines) and several multiple reflections (dashed lines). For isotropic media, the travel paths are trajectories perpendicular to the corresponding wavefronts.*

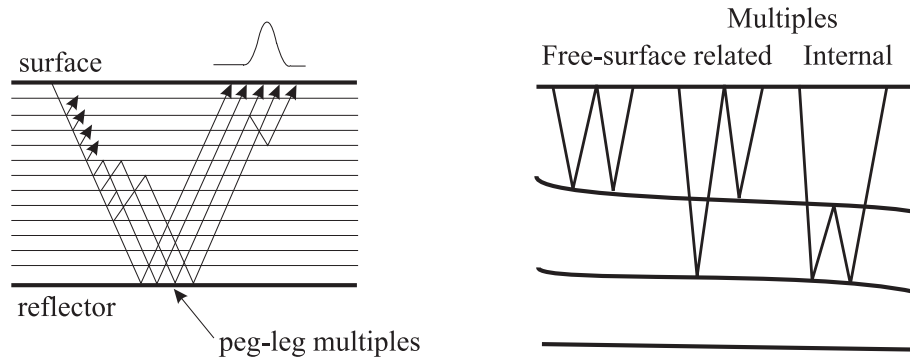


Figure 1.2: *Left: short period peg-leg multiples. The multiples reverberating between layer boundaries have a long period compared to the wavelength of the seismic wave. These are the multiples we consider as unwanted noise, whereas the “reflection generating” small-scale reverberations naturally should not be suppressed. Right: different types of multiples: 1) free-surface related multiples such as water reverberations, peg-leg multiples, and 2) internal multiples. A first-order multiple is a multiple that is reflected downward only once, whereas a second-order multiple is scattered back downward twice from any one of the discontinuities, etc.*

j -direction, $u_{k,l}$ the spatial derivative of component k in the l -direction (using the summation convention), and \ddot{u}_i is the second time derivative of the i th component of the vector u . c_{ijkl} are the components of the elasticity tensor, f the source function and ρ the density (Aki and Richards, 1980). If the source term f is an impulsive function in space and time, then the solution is a special result called the Green’s function G . For problems with constant coefficients, homogeneous, isotropic and unbounded media, the Green’s function can be found analytically and the problem is solved. In general, this is not the case and it has to be approximated, e.g. using ray theory.

A wave propagating through an arbitrary inhomogeneous medium will be scattered at inhomogeneities (scatterers) thus producing a scattered wavefield. If the wavefield is only scattered once between source and receiver (or, as shown in Figure 1.2, left, multiply on a small scale), this is the primary part of the wavefield. As soon as it is scattered from a second scatterer, which is at a distance larger than a wavelength, it becomes part of the multiple wavefield.

The problem of separating the primary wavefield from the multiple wavefield amounts to separating the Green’s function that solves equation 1.1 for an impulsive source into the Green’s function for the primary wavefield, the Green’s function for the multiple

wavefield, and a Green's function for that part of the wavefield that doesn't belong to either primary or multiple energy:

$$G = G_P + G_M + G_R, \quad (1.2)$$

where G_P is the part for the primary, G_M the part for the multiple wavefield, and G_R the part for the remaining wavefield that cannot be associated with one of the two (i.e. noise or effects not described by the model). The Green's function term for the multiple wavefield G_M can be split further into Green's functions for the partial wavefields of e.g. free-surface related multiples and internal multiples:

$$G_M = G_F + G_I, \quad (1.3)$$

where G_F is the part for the free-surface multiples, and G_I for the internal multiples. The different types of multiples together with their naming convention are shown in Figure 1.2, right. This separation of Green's functions is only possible in an analytical way for certain simple geometries, of course.

In the next section I shall introduce a number of existing techniques to solve the problem of separating out particular multiple scattering contributions. They all share one common feature: they only work in special cases. The basis for each method is a physical model that holds for only a limited number of real-life cases (depending on subsurface geometry, acquisition parameters, elastic parameters), and is often too rough and too simplified. Assuming the physical model to be known implies knowledge about the governing parameters, which might or might not be available.

1.2 Multiple Attenuation Methods

Methods for multiple attenuation either try to exploit a physical property or feature that differentiates primaries from multiples by the use of filtering algorithms, or they aim to predict the multiples by using modelling or inversion techniques. Such predicted multiples are later subtracted from the recorded seismic data.

So far there is no multiple attenuation technique that works universally. Common methods used in the industry today can be divided into seven different major categories:

1. Methods exploiting periodicity

These methods are based on the periodicity of multiples in contrast to the primary arrivals. Predictive deconvolution using Wiener filters (Robinson and Treitel, 1980; Peacock and Treitel, 1969) is one successful approach that designs a filter operator which predicts the multiples and subtracts them from the seismic trace. However, at far offsets this method fails, since multiples are no longer periodic. Another drawback of this method is that it requires assumptions such as that the seismic trace be stationary and the reflectivity be a random series of spikes. Taner (1980) showed a way to overcome the non-periodicity problem at far offsets by applying predictive deconvolution in the radial trace space. Similarly, in the $\tau - p$ domain multiples are periodic for all slowness (p) values (Carrion, 1986), but a horizontally layered medium must be presumed.

2. Stacking methods

The Stacking methods exploit the move-out (travel time) difference between the primary and the multiple reflection hyperbolae in a CMP gather: In principle, these methods sum over all possible hyperbolae by scanning through all zero-offset times and curvatures of all theoretically possible hyperbolae. If the summation coincides with an actual reflection hyperbola, a high amplitude signal is generated, while in all other cases the signal will be weaker. Since the curvature of the hyperbola relates to the velocity of the corresponding arriving wave, this method is called velocity analysis (Yilmaz, 1987, e.g.). In the velocity spectrum an experienced interpreter can distinguish primaries (generally with higher apparent velocities) from multiples (generally with lower apparent velocities), and can thus produce a macro-velocity model. Normal move-out (NMO) correction with the picked primary velocity model (macro model) and subsequent common-midpoint (CMP) stacking already reduces the amount of multiple energy (Schneider et al., 1965). If the NMO correction is applied in such a way that primaries are overcorrected and multiples are undercorrected, the two signals map onto different half-spaces in the $f - k$ domain (Yilmaz, 1987). However, these correlation methods often fail because arrivals from deeper reflectors or complex geologic structures increasingly deviate from a simple hyperbolic relationship. Thus primaries often cannot be separated from multiples in the velocity or $f - k$ spectrum, mainly due to interference effects.

Moreover, in cases where internal multiples are created at velocity inversions they show higher apparent velocities than primaries appearing at the same zero-offset time. This leads to incorrect velocity models and to amplification of such multiples instead of their attenuation.

3. Prediction Methods based on wavefield parameters

This method is based on the idea that all multiple travel times - for both free-surface and internal multiples - can be constructed by adding or subtracting primary travel times to or from each other (Keydar et al., 1998). In order to find the relevant primaries the method compares the emergence angles of the wavefronts arriving at the surface. If a number of primaries can be combined to build a certain type and order of multiple, the corresponding events are the searched primaries, and the multiple can be predicted in a kinematic sense. The essential parameter is the wavefront emergence angle, which is obtained - along with the radius of curvature of the reflected wavefront - with a local moveout correction computed in the common shot domain. The power of this method is its independence of the macro-model and the practical aspect to suppress targeted multiples with the interpreter's input of relevant primaries, which can at the same time be a drawback. This method also requires the near-surface velocity, which is no problem in marine settings, but is often not known well enough for land data.

4. Wavefield extrapolation methods

These methods predict multiples by extrapolation of the wavefield into the subsurface and subsequently subtract them from the data (Berryhill and Kim, 1986; Wiggins, 1988): Here, the wavefield is propagated down and up through the water layer so that the primaries become first-order multiples which are then subtracted adaptively from the original data. An estimate of the water-bottom topography and reflectivity is needed, and only water-bottom multiples can be removed. This method relies on an adaptive subtraction algorithm, which can produce an incorrect signal if it is not working perfectly.

5. Methods based on autoconvolution

These methods predict free-surface related multiples by iterative autoconvolution in time and space (Berkhout and Verschuur, 1997; Verschuur and Berkhout, 1997).

They assume knowledge about the source signature and about the existence of zero- and near-offset data, which are almost impossible to record in the field. Extrapolation to zero-offset and interpolation between the recorded traces can create large errors that can severely degrade this method in practice.

6. **Methods based on coherency**

These methods are based on computing coherency measures (using Singular Value Decomposition) after an NMO correction that flattens multiple reflections (Kneib and Bardan, 1997). The eigenimages with the highest eigenvalues (pertaining to the multiples) are removed, leaving the primary information in the ideal case, but not always in reality, since the largest eigenvalues may also contain multiple energy due to incorrect NMO correction.

7. **Inverse scattering methods**

Inverse scattering methods express the total wavefield as the sum of the wavefield from a smooth known background plus the wavefield generated by scatterers (Weglein et al., 1992). Multiples are represented by higher order terms in a non-linear inverse scattering series and can thus be removed. However, here both source signature and a background model have to be known. Similar to the autoconvolution method, this technique also requires near-offset traces. So far there do not exist near-trace extrapolation methods for shallow water, so in that case the method is likely to fail.

Approaches exist where multiples are not treated as noise that has to be eliminated but regarded as signal that has travelled through a certain part of the subsurface many times more than a primary and thus can also give us information about the geology (Helbig and Brouwer, 1993).

An unconventional but potentially promising approach to suppress multiples is based on the use of artificial neural networks (Calderón-Macias et al., 1997). In this dissertation I introduce four novel methods, all based on such neural networks.

1.3 Multiple Attenuation with Neural Networks

All the conventional techniques for multiple attenuation mentioned in the previous section fundamentally depend on a physical model that is usually not known exactly and whose governing parameters can often only be estimated imprecisely. In those cases where the simple model is sufficiently close to the reality and/or the greater part of the underlying assumptions is fulfilled these methods produce good results. However, already in the presence of noise, very few physical models still hold, and the applicability decreases dramatically. Even a combination of these methods would not overcome this problem inherent in this kind of approach.

In reality the physical laws governing the problem of multiple reflections are not known to an extent allowing the development of robust and universally applicable algorithms. What we need is a method that is partly model-based, but still adaptive and includes the possibility of posing constraints. The neural network technique is an appealing approach that encompasses all of that. The *backpropagation* neural network (see section 2.1) is able to mimick the physics of the underlying model; it tries to extract the rules and physical laws governing the problem. At the outset there is a fuzzy set of examples, consisting of inputs to the problem and known solutions for these particular inputs. A *self-organizing map* (SOM) algorithm (see section 2.2) analyzes the problem and tries to establish rules for the characterization of the problem. It makes use of the measurements as such or attributes computed from them and finds interrelations and connections between the data.

Within the scope of this dissertation four different multiple attenuation and identification algorithms based on backpropagation neural networks and self-organizing maps have been developed, tested and analyzed. Figure 1.3 shows diagrammatically the four different methods employed.

The first method (see chapter 4) is a trace-by-trace multiple attenuation scheme using supervised backpropagation neural networks. On the basis of well-log data, networks are trained to act as a non-linear filter attenuating free-surface as well as internal multiples present in a full elastic wavefield recorded in the space-time domain.

The other three methods employ seismic attributes (described in chapter 3) from various other data domains using backpropagation neural networks and self-organizing maps.

| <i>Neural Net Input: Attributes from</i> | | | |
|---|--|---|------------------------------------|
| <i>x-t domain</i> | <i>various domains</i> | | |
| Trace-by-trace & Ensemble Backpropagation | Attribute-based Backpropagation | Attribute-based self-organizing Maps (SOM) | Layer-stripping Backpropagation |
| <i>Multiple Attenuation</i> | <i>Multiple Prediction & Attenuation</i> | <i>Primary/Multiple Identification & Classification</i> | <i>Multiple Attenuation</i> |
| Chapter 4 | Chapter 5 | Chapter 6 | Chapter 7 |

Figure 1.3: Overview of the methods investigated in this dissertation: It is possible to perform multiple prediction, classification and attenuation. By using attributes from different domains, and different neural networks we can perform one or more of these tasks.

Recognizing that information about the multiple content of the data is present in various data domains and attributes (e.g. instantaneous attributes, wavelet attributes, wavefront parameters, velocity spectrum) computed in these domains, I train the networks with combinations of such inputs. Since these input domains constitute a rather large parameter space, the information is compressed by using sparse attributes and segments of these input domains. The idea of using physically meaningful attributes is to combine the different discriminatory powers in the various domains to exploit the redundancy of information inherent in the seismic data. Every conventional technique only relies on a single discrimination criterion provided by one parameter space (e.g. f-k filtering), and neglects the possibilities offered by other domains. The performance of all methods is demonstrated on a number of synthetic data sets.

In the attribute-based multiple prediction and attenuation technique using backpropagation I assemble a supervised learning algorithm which produces estimates of the desired wavefields (chapter 5). Depending on the desired output provided for the neural network, there exists the possibility of either attenuating or predicting free-surface and internal multiples. Basis for multiple attenuation is a modeled primary section obtained from well-log information at certain locations in the exploration area. For the prediction of the multiples an estimate of the desired multiples has to be provided using conventional

multiple prediction methods for a few seismic sections. The neural network then extracts either the primaries or the predicted multiples directly from the attributes computed from the seismograms for the remaining major part of the data set.

The self-organizing maps represent an unsupervised learning method that analyzes the seismic data and tends to extract information with the help of a sophisticated clustering technique. In chapter 6, a new technique is introduced, that uses only the seismic data or attributes derived from the seismograms, and does not require any a priori estimate of the primary or multiple wavefield. Using an automatic picking algorithm, all relevant events, primaries and multiples, are picked in a seismic data set. Then a number of carefully selected attributes is computed for each event. The self-organizing map arranges the data in an ordered manner, and forms clusters that not only allow to separate primaries from multiples, but also to distinguish between different types of multiples.

A practical method for multiple attenuation using supervised backpropagation neural networks is shown in chapter 7. The input data consist, as for the SOM in chapter 6, of attributes computed for a set of reflection events. The desired output is provided by an interpreter in the form of a few picked and classified events. The neural net is then trained with this information in a layer-stripping manner. It works its way top down through the data set by learning from the given information, classifying new events, taking them for training as soon as they have been classified with a certain confidence, learning again, and so on, until all events have been processed. This method also allows user interference between the individual training runs, e.g. when an event can be definitely labelled as a primary or multiple on the basis of the events already classified by the algorithm.

Chapter 2

Neural Networks

Among the numerous types of neural networks now in use, I apply two to the multiple problem: backpropagation neural networks and self-organizing maps. The former technique is the most robust of the feed-forward supervised learning algorithms. In fact it is nothing more and nothing less than a non-linear adaptive filter, and provides the possibility of comparison with linear filtering methods, such as the Wiener filter. On the other hand, the self-organizing maps (or Kohonen feature maps) represent one of the more sophisticated unsupervised learning techniques used for cluster analysis.

2.1 Backpropagation Neural Networks

2.1.1 Neural Networks as Non-Linear Filters: Theory

Typically, in classical inverse theory (Tarantola, 1987) it is assumed that we possess relatively accurate knowledge of the physical model underlying a certain problem. This knowledge is expressed explicitly as an operator \mathcal{A} , and can be written in matrix form. If this operator is applied to a model vector \vec{x} , it roughly reproduces the data vector \vec{d} consisting of measured data:

$$\mathcal{A} \vec{x} \approx \vec{d} \tag{2.1}$$

In a classical inverse problem we know the measured data \vec{d} and we also know precisely

the operator \mathcal{A} , which could be, for example, a normal move-out (NMO) - operator (Calderòn-Macias, 1997, shows a neural net approach to NMO correction), or a filter to predict and subtract multiples. We want a good model for the data \vec{d} : the model parameters \vec{x} . Equation 2.1 can be inverted in a formal manner for \vec{x} ,

$$\vec{x} \approx \mathcal{A}^{-1}\vec{d}. \quad (2.2)$$

This is usually an approximation, since there is always noise present in seismic data. By minimizing an error measure $\|\mathcal{A}\vec{x} - \vec{d}\|^p$ ($p = 1, 2, 3, \dots$) we can find the best approximation of a model fitting the measured data.

To illustrate this principle, we take a look at the NMO correction (Claerbout et al., 1997), expressed in matrix form (here $\mathcal{A} = \mathcal{NMO}$):

$$\vec{d} = \mathcal{NMO} \vec{x} = \begin{bmatrix} \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix}. \quad (2.3)$$

The 1's in the NMO-operator are arranged along the hyperbola $t^2 = t_0^2 + x_0^2/v^2$ and the dots stand for zeros. The vector \vec{x} represents one trace in a CMP gather, which gives after application of the NMO-operator the NMO-corrected trace \vec{d} .

Inverting equation 2.3 for \vec{x} gives $\tilde{\vec{x}}$:

$$\tilde{\vec{x}} = \mathcal{NMO}^{-1}\vec{d} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \frac{1}{3} & \frac{1}{3} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \frac{1}{3} & \frac{1}{3} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \frac{1}{3} & \frac{1}{3} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \end{bmatrix}, \quad (2.4)$$

where \mathcal{NMO}^{-1} is the pseudo-inverse of \mathcal{NMO} .

The mapping rule from the model space to the data space is known and can be expressed as a matrix operator \mathcal{A} . However, in many cases this relationship is *not* known or only known empirically, thus it cannot be expressed explicitly in mathematical form. In such a situation classical inverse theory as described above fails, and the neural networks can be used advantageously.

The action of a neural network can be described by the same equation as classic inverse theory (eq. 2.1). However, we do not know the operator \mathcal{A} , but have a set of examples available in the form of data from an input space x and corresponding data from an output space d . These examples should be an exhaustive representation of the mapping from input to output space. The task of the neural network is then to find this relationship (the operator \mathcal{A}) by learning from training examples. \vec{x} is one of many input data vectors of the training set, and \vec{d} is the corresponding desired output vector. The operator \mathcal{A} represents here the weight matrix of the neural network, which has to be optimized in order to guarantee not only correct mapping of the training data, but also generalization to data not included during training. This is accomplished by a learning rule often based on gradient descent algorithms which compute the gradient and descend in the negative direction of the largest gradient in order to find the minimum. One of the numerous advantages of the neural net approach is that it is not restricted to the L2-norm (p=2), so that the error measure can be adapted to the specific problem.

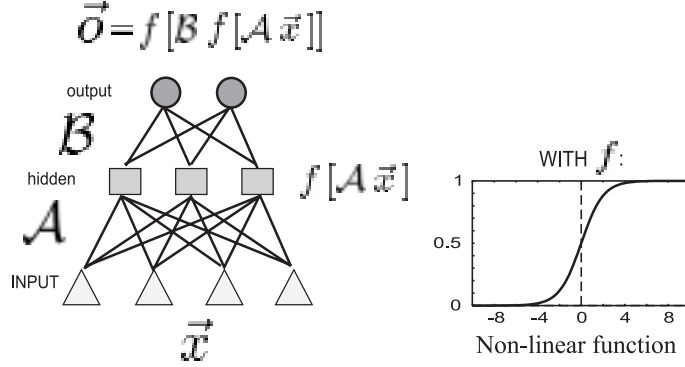


Figure 2.1: The three-layer backpropagation neural network with the input vector \vec{x} , the output vector \vec{o} , the weight matrix \mathcal{A} associated with the connections from the input layer to the hidden layer, the weight matrix \mathcal{B} from the hidden to the output layer, and a non-linear function f .

Normally a neural network consists of three or four layers - an input layer, one or two hidden layers (called hidden, since they don't have a direct connection to the outside world), and an output layer - as shown in Figure 2.1. The layers are fully connected by weights. Therefore, the weight matrix is more complex in this situation than in eq. 2.1. The network can be expressed as

$$\tilde{\mathcal{B}} [\tilde{\mathcal{A}} \vec{x}] = \vec{o} \quad (2.5)$$

$$\text{or } f[\mathcal{B} f[\mathcal{A} \vec{x}]] = \vec{o}, \quad (2.6)$$

where \mathcal{A} is the weight matrix associated with the connections from the input layer to the hidden layer, \mathcal{B} is the weight matrix associated with the connections from the hidden to the output layer, \vec{x} is the input, \vec{o} the output of the network, and f is a non-linear function (shown in Figure 2.1). The matrices $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ are related to the original matrices \mathcal{A} and \mathcal{B} via the non-linearity f . An example from geophysical data processing reveals the analog formulation:

$$MIG [STACK [DMO [NMO \vec{x}]]] = \vec{o}. \quad (2.7)$$

It represents the successive application of NMO, DMO, Stacking and Migration operators to the data \vec{x} . The non-linearity f could be included in the form of trace amplitude manipulations, such as automatic gain control (AGC) at certain steps.

2.1.2 The Multi-layer Perceptron

In 1911 Ramón y Cajál came up with the idea of the neuron as the basic component of the brain. Today we know that a human brain contains some 100 billion (10^{11}) neurons with about 1,000 to 10,000 connections each (resulting in a total of 10^{14} - 10^{15} interconnections). Although one neuron is about 10^6 times slower than a transistor of a computer (neuron: 10^{-3} sec, transistor: 10^{-9} sec) the massively parallel processing capability of the brain gives it a much higher efficiency.

Simulating neurons on the computer began with the pioneering paper of McCulloch and Pitts (1943) who described the formal theory of neural networks. Then, in 1958, Rosenblatt invented the perceptron (see Figure 2.2), an artificial neuron, whose inputs x_i are multiplied by weights w_{ji} . These weights are the only variables and determine the contribution of the individual inputs. These N weighted inputs are simply summed inside the neuron, and θ_j is a suitable threshold. The “activation” a_j of the neuron j is then given by

$$a_j = \sum_{i=1}^N w_{ji}x_i + \theta_j. \quad (2.8)$$

Usually, the threshold or bias θ_j is realized as an additional input x_0 , which is set to 1 and multiplied by an additional weight w_{j0} :

$$a_j = \sum_{i=0}^N w_{ji}x_i \quad (2.9)$$

This sum of weighted inputs is then passed through a non-linearity f . The result is the output of neuron j :

$$o_j = f(a_j) \quad (2.10)$$

This non-linearity allows the neural net to adapt itself to a variety of data that can show extremely non-linear behavior.

In the case of backpropagation this non-linear function has to be continuously differentiable, since the first derivative f' appears in the learning rule and also because we want a continuous output (Cichocki and Unbehauen, 1993). A popular choice for the non-linearity is the sigmoid function, also known as the Fermi-function in physics, which is linear around zero and asymptotically reaches 1 or 0, respectively for arguments going to $\pm\infty$ (cp. Figure 2.1).

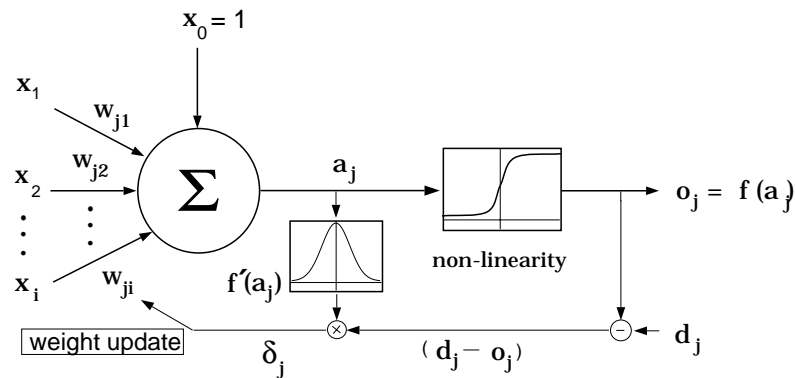


Figure 2.2: *The Perceptron: forward propagation of the input x_i . x_0 is the threshold, w_{ji} are the weights, a_j is the activation function, f is the non-linear transfer-function and o_j represents the output of neuron j . Backpropagation: The difference between the desired output d_j and the actual output o_j is multiplied by the first derivative of the non-linearity $f'(a_j) = o_j(1.0 - o_j)$. This is the local error δ_j that is “backpropagated” (cp. equations 2.11 and 2.12) from layer l to layer $l - 1$ (here shown for output layer to hidden layer).*

The individual artificial neurons can be interconnected in many different ways leading to a variety of neural networks with different architectures, learning rules and abilities. The most important ones are: Feedforward networks, Adaptive Resonance Theory networks (ART), Hopfield nets, Kohonen’s self-organizing feature maps, Radial Basis Functions (RBF), Boltzmann-machines, and Cascade-correlation networks (Haykin, 1994; Zell, 1994).

A simple way to organize the neurons in several layers is shown in Figure 2.1. This architecture is called a feed-forward network, since neurons of one layer are only connected with neurons of the succeeding layer, without any recurrent connections. Normally, these nets consist of one input layer, one or two hidden layers and one output layer. With such a net, input data are mapped from the n -dimensional input space to an m -dimensional output space. This network now has to learn to produce a certain desired output for each input pattern presented at the input layer.

The detailed training procedure is as follows:

1. Split the data set into a training set and a test set. Normally, the training set is larger than the test set. Often the desired outputs have to be normalized to the range $[0 : 1]$ since the sigmoid function only returns values in this range ¹. The input patterns do not have to be normalized.
2. Initialize all weights, including all biases, to small random values (usually in the range of $[-1 : +1]$). This determines the starting point on the error surface for the gradient descent method, whose position can be essential for the convergence of the network.
3. Forward propagation of the first input pattern of the training set from the input layer via the hidden layer(s) to the output layer, where each neuron sums the weighted inputs, passes them through the non-linearity and passes this weighted sum to the neurons in the next layer.
4. Calculation of the difference between the actual output of each output neuron and its corresponding desired output. This is the error associated with each output neuron.
5. Backpropagating this error through each connection by using the backpropagation learning rule (described below) and thus determining the amount each weight has to be changed in order to decrease the error at the output layer.
6. Correcting each weight by its individual weight update.
7. Presenting and forward propagating the next input pattern ...

Repeat steps 3-7 until a certain stopping criterion is reached, for example that the error falls below a predefined value.

The one-time presentation of the entire set of training patterns to the net constitutes a training epoch.

¹Normalization from range $[min:max]$ to range $[MIN:MAX]$ can be accomplished by:
$$y = \frac{MAX-MIN}{max-min}(x - min) + MIN$$
 , where $min \leq x \leq max$ and $MIN \leq y \leq MAX$

After terminating the training phase the trained net is tested with new, unseen patterns from the test data set. The patterns are forwardpropagated, using the weights now available from training, and the error at the output layer is determined (no weight-update is performed!). If performance is sufficiently good, the net is ready-for-use. If not, it has to be retrained with the same patterns and parameters or something has to be changed (e.g. number of hidden neurons, additional input patterns, different kinds of information contained in the input patterns, ...).

The Backpropagation Learning Rule

For every neuron the weighted input is summed, passed through the non-linearity f , yielding the actual output which is subtracted from the desired output. This error is backpropagated from the output layer to the hidden layer and from the hidden layer to the input layer by correcting each weight after the n th epoch

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \quad (2.11)$$

by the weight update

$$\Delta w_{ji}(n) = \eta \delta_j x_i. \quad (2.12)$$

Here x_i is the input to neuron j and δ_j is the local error, whose form depends on whether neuron j is an output or a hidden neuron:

1. The neuron j is an output neuron:

δ_j is the product of $f'(a_j)$ and the error $(d_j - o_j)$

For the case of sigmoidal non-linearities: $\delta_j = (d_j - o_j) o_j (1.0 - o_j)$

2. The neuron j is a hidden neuron:

δ_j is the product of $f'(a_j)$ and the weighted sum of the δ 's of the neurons in the next layer (in the backward direction). This next layer has k neurons.

For the case of sigmoidal non-linearities: $\delta_j = [o_j (1.0 - o_j)] [\sum_k \delta_k w_{kj}]$

The Rprop learning rule

Instead of using the standard backpropagation learning algorithm I employed the Rprop (resilient propagation) algorithm (Riedmiller and Braun, 1993) which shows considerably faster convergence. In contrast to other gradient-descent algorithms, Rprop does not use the magnitude of the gradient, but only its sign. It begins with an initial small update value, and then increases this value by a factor greater than 1 (typical value: 1.2), if the current gradient has the same direction (sign) as the previous gradient, but decreases this value by a factor smaller than 1 (typical value: 0.5), if the gradient has the opposite direction (sign).

This update then is added to the weight, if the gradient is negative (progression in the positive direction towards the minimum), and subtracted from the weight, if the gradient is positive (progression in the negative direction towards the minimum). The principle is shown in Figure 2.3.

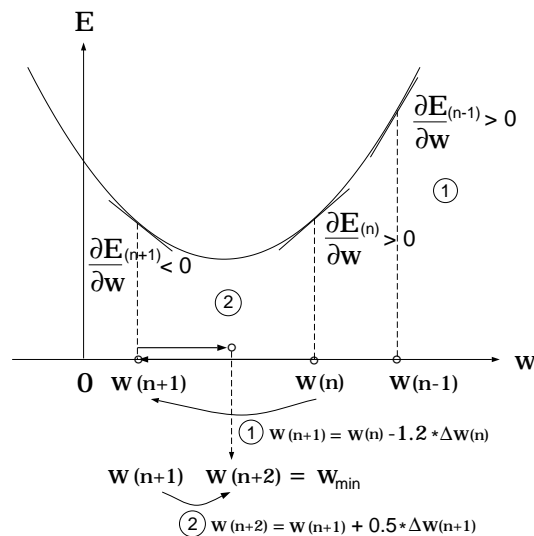


Figure 2.3: Principle of the Rprop-algorithm: ① The gradients of the steps $n - 1$ and n are both positive, so the last update $\Delta w^{(n)}$ is increased by the factor 1.2 and subtracted (because the current gradient is positive) from the current weight $w^{(n)}$. ② The gradients of the steps n and $n + 1$ have different signs, so the last update $\Delta w^{(n+1)}$ is decreased by the factor 0.5 and added (because the current gradient is negative) to the current weight $w^{(n+1)}$. E is the total error.

2.1.3 Practical Aspects of Backpropagation

Parallel Computing

We often find the statement in the literature about Neural Networks: “*neural networks are ideally suited for parallel computing applications*”. That is true, since every computing node can perform the task of one single neuron of the network and there are already hardware implementations of parallel neural networks, e.g. for speech recognition, automatic car steering, etc.

The practical use of backpropagation neural networks can be separated into the following steps:

Network Design

When experimenting with neural networks, the design of the network is often interactive and it is mostly the skill of the user that determines the speed of this development process. The user has to decide on the type of input and output and the type of networks to be used. This decision process is more or less user and problem driven.

Generation and Preprocessing of Training Data

The next step is to generate sufficient training and test data in order to achieve good neural network performance, which generally means that it can generalize from the learned training data to unseen test data. The training data have to be representative of the physical problem we want to solve. For example, if the multiple suppression problem is essentially caused by a 1D subsurface structure depending on $v_p(z)$, $v_s(z)$ and $\rho(z)$, we need to model as well as possible the full wavefields and the primary wavefields in a 1D elastic earth. If the subsurface exhibits 2D or 3D velocity and density variations $v_p(x, y, z)$, $v_s(x, y, z)$ and $\rho(x, y, z)$, then the 1D modeling algorithms will not be sufficient any longer and other modeling algorithms which are adequate for 2D and 3D structures have to be used. Training the network with 1D examples, whereas the real problem is 2D or 3D, will result in a failure of the neural network to cope with the problem. We can then expect unsatisfactory results when applied to the real data. An important point is that the training data must be chosen in such a way that they span the whole experiment space, which is a function of the elastic parameters, acquisition parameters, wavetypes, etc.

Training the Network

Once the neural network has been designed and the training data set has been generated, the network can be trained. We perform the training process generally non-interactively in batch-mode. The training is stopped when the error at the output layer does not change significantly anymore, or - and this is very important - when the error of the validation data set starts to increase again after going through a minimum. This is the point where overtraining of the network starts to occur.

Testing the Trained Network

After the neural network has been trained, we apply it to some synthetically created test cases. Here we can quantify directly how the network has performed, because we know input, output and desired output. Thus we have a direct error measure (least squares error). A very extensive but thorough method to estimate the performance of the network and to obtain confidence bounds is the k-fold cross validation technique (Aminzadeh, 1997). In a seismic example this would require for example computing large numbers of synthetic seismograms by perturbing reservoir properties and feeding the corresponding attributes to the network.

Applying obtained Networks to Real Data

After successful performance on the synthetic test data, the trained network is ready to process real data.

2.2 Self-Organizing Maps (SOM)

Self-organizing maps are a class of artificial neural networks bearing a much closer relation to the actual biological brain than do the backpropagation networks, which are merely biologically inspired non-linear filters. An important class of techniques simulating memory is the associative memory, which will be discussed first, before the theory and practice of the self-organizing maps as such is presented.

Associative Memory

The concept of associative memory is very old. Already the Greek philosophers (e.g. Aristotle, 384 - 322 B.C.) formulated the principle:

Events that tend to coincide in time are connected by the human brain.

If you hold a finger into a flame, this will “immediately” lead to pain and to the withdrawal of the finger. Associative learning essentially is the building of connections between an external stimulus and the reaction to it (Thompson, 1993). In order to memorize these connections, some neural alterations (changes of synaptic connections) have to take place inside the brain. When a certain stimulus/reaction pair is learned, it is stored in a particular region in the brain, from where it can be recalled later when required.

Even when a stimulus or pattern presented to this memory is noisy or imperfect, the associative memory is able to correctly recall the stored pattern that is associated with this stimulus. This feature makes simulations of associative memories prime candidates for pattern recognition applications. If we assume the memory operator to be linear, we can express it as

$$\vec{b} = M\vec{a}, \quad (2.13)$$

with the input stimulus \vec{a} , the response \vec{b} , and the memory matrix M . In the non-linear case we have

$$\vec{b} = f(M, \vec{a})\vec{a}, \quad (2.14)$$

where f is a non-linear function of M and \vec{a} .

The linear case is depicted in Figure 2.4 where every neuron acts as a linear combiner, similar to a perceptron with linear transfer function f (cp. section 2.1.2). If we want to store a whole set of pattern vectors $\vec{a}_1, \vec{a}_2, \dots, \vec{a}_k$, with the corresponding memorized

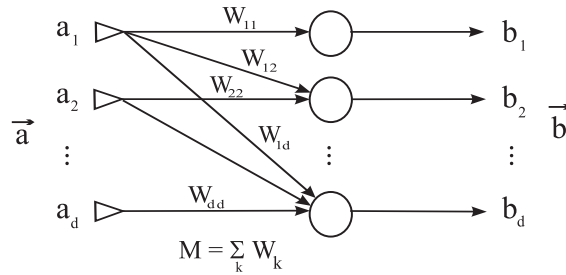


Figure 2.4: Model of an associative memory in the form of a neural network.

patterns $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_k$, we can store a certain number q of these patterns, which is always smaller than the dimensionality d of the network.

The memory matrix M is the sum of all the individual weight matrices W_k that associate \vec{a}_k with \vec{b}_k :

$$\vec{b}_k = W_k \vec{a}_k \quad (2.15)$$

$$M = \sum_{k=1}^q W_k \quad (2.16)$$

This means that the memory matrix M contains a little piece from every learned stimulus/reaction pair.

We can now construct an estimate of the memory matrix M_{est} :

$$M_{est} = \sum_{k=1}^q b_k a_k^T \quad (2.17)$$

The memory matrix is based on the outer product of the input and output patterns, and thus also called the *correlation matrix memory* (Haykin, 1994). Actually, correlation in general is the basis for learning, recalling memory and pattern recognition.

Another important aspect of associative memory is that it is distributed. That means different external stimuli are mapped onto different parts of the memory. Research of the past years has shown that the human cerebral cortex is organized in different areas where certain brain actions take place. There are special brain areas that perform specialized tasks, for example, the processing of sensory signals (visual, sensory, somatosensory, auditory, ...), speech, motor functions, thinking, long term and short term memory (see Figure 2.5).

It is thought that this topographical ordering of the brain is a very important feature and thus plays also an essential role in the simulation of intelligent systems. The

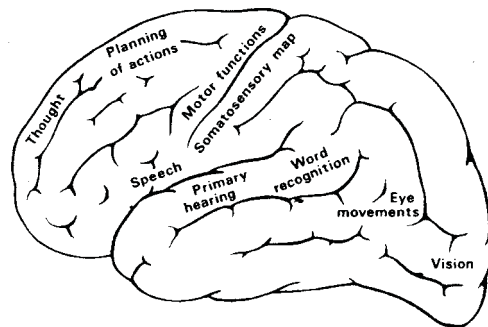


Figure 2.5: *The human brain with its receptive fields (from Kohonen, 1998).*

“self-organizing map”, developed mainly by Teuvo Kohonen, is such an artificial neural network whose topographical organization of its neurons is its essential aspect.

2.2.1 Principles of Self-Organizing Maps: Theory

The self-organizing map (SOM) belongs to the class of unsupervised learning techniques of artificial neural networks, since there is no desired output given during learning. The network “organizes itself”, as the name already implies.

A self-organizing map usually consists of a 2-dimensional grid (feature map) of neurons. Sometimes problem dependent 1-dimensional feature maps are used, however, maps in higher dimensions can be implemented as well. A self-organizing map can be used to perform a dimensionality reduction, allowing us to project high-dimensional data onto a 2-dimensional map, which is easy to display. Each of these feature map neurons is connected to each input neuron (i.e. there are as many input neurons as one input vector has components), as is indicated in Figure 2.6.

Clustering

The goal of training a self-organizing map is to separate the input data into several distinct clusters, which can be - in the 2D case - visualized on the 2-dimensional map. That means a multi-dimensional input space is mapped onto a 2-D output space. As an example, Figure 2.6 displays the clustering of input vectors containing primary wavefield information into one area of the map (left), and those input vectors containing multiple

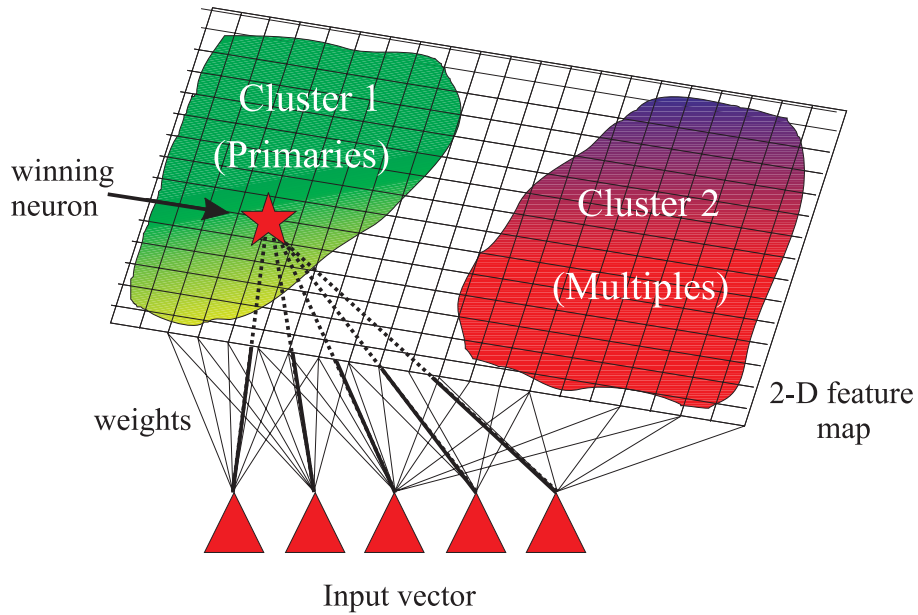


Figure 2.6: A self-organizing map (Kohonen network) with a 2-dim. feature map and a 5-component input layer. Every input neuron is connected with every neuron of the feature map.

wavefield information into another area (right).

In general, the clustering aspect is more important than the desire to visualize it. If we use a 2-dimensional map while the dimension of the data being higher, it can be difficult to obtain optimal clustering: e.g. data points lying very close together in a higher-dimensional space might be located far from each other on a 2-dimensional map that is folded into that space. Figure 2.7 a) shows the principle. The dimension we are dealing with here is not of the dimension of the input vectors, but the fractal information dimension (Peitgens et al., 1992), which is defining the information content in the data.

This clustering is carried out by computing differences (or similarities, respectively) between all input vectors and each of the set of weights connecting the input layer with a feature map neuron. A commonly used criterion is the Euclidean distance

$$d = \sqrt{\sum_j [x_j - w_j]^2}, \quad (2.18)$$

where j denotes the j th vector component. If d is zero, the input \vec{x}_i and the weight \vec{w}_i are identical. Alternatively, the vector product of \vec{x}_i and \vec{w}_i can be used. If it is zero, the two vectors are orthogonal.

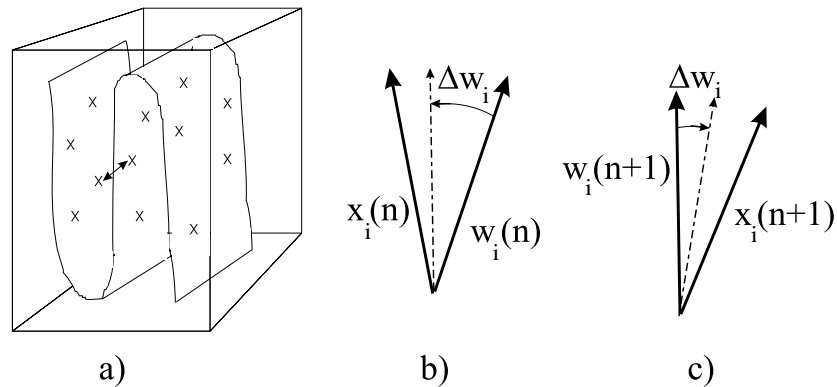


Figure 2.7: a) A data set with a fractal dimension of 3, trying to cluster on a 2-D map. The two points that are close together in 3-D are far apart on the 2-D map. b) Weight update: the weight $\vec{w}_i(n)$ is moved into the direction of the input vector $\vec{x}_i(n)$ producing $\vec{w}_i(n+1)$ which in turn is updated during the next iteration: c) The next input vector $\vec{x}_i(n+1)$ leads to an update of the weight into its direction (n is the iteration index).

The weights are initialized randomly and should - in the end - form a topologically ordered clustering of the input vectors.

An important remark: the location of the neurons on the map and the weight vectors are two different things that should not be confused. Every neuron has its fixed location on the grid and is associated with a weight vector having as many components as the input vector.

Training

For training the neural network, we present all the input vectors, one at a time, to the net. Each input vector is compared to every weight vector associated with every neuron, i.e. the Euclidean distance is computed. The one feature map neuron having the weight vector with the smallest difference (or highest similarity, respectively) to the current input vector is the winning neuron. This is the “winner takes all” concept of neural computation. The weight of this winning neuron is now updated in the direction of the input vector. That means, if this input vector is presented to the net a second time, this neuron is very likely to be the winner again, and thus represent the class (or cluster) for this particular input vector. Clearly, similar input vectors will be associated with winning neurons that are close together on the map.

The weight update is performed, using the difference of the current input vector \vec{x}_i and the current weight $\vec{w}_i^k(n)$, where n is the iteration number:

$$\vec{w}_i^k(n+1) = \vec{w}_i^k(n) + \alpha(n)[\vec{x}_i - \vec{w}_i^k(n)]. \quad (2.19)$$

The index k denotes the winning neuron, and $\alpha(n)$ is the learning rate, which should decrease with increasing number of iterations.

An important point, related to the topology-ordering feature of the SOM, is that the neurons are also connected to their neighboring neurons. In other words, during a weight update not only is the weight vector of the winning neuron updated, but also the weight vectors of the neighboring neurons are updated in accordance with a “neighborhood function”. This neighborhood function often is chosen to be Gauss-shaped, so that the weight-update becomes smaller with increasing distance from the winning neuron.

This results in a special kind of vector quantization such that the weight vectors become ordered in a way that they represent the input vectors on an “elastic” grid. If there are changes at one location on the grid, this change affects the neighborhood of this neuron. However, the further away it is located the less influence this change has.

In this way, a map evolves, where every region represents a class of input vectors, or, in other words, we try to represent a data set (the input vectors) by a number of weights, each of which (or several of them) represents the mean of a certain class or cluster of input vectors. By training we try to establish a configuration where each weight lies in the middle of a cluster of input vectors. If we present during training an input vector, one weight vector will be closest to this input vector, this is the weight pertaining to the winning neuron. Thus, in order to get this weight vector into the middle of its class, we must move it into the direction of this input vector. If it is the winning neuron again, when presenting another input vector, we will move it into the direction of this input vector (i.e. maybe a little away from the first input vector, but hopefully into the middle of the two), and so on.

In the beginning of the training not only the weight of the winning neuron is moved but also the weights of the neighboring neurons within a certain radius. This strategy improves the convergence behaviour (Haykin, 1994). This radius is decreased during learning. Thus in the beginning there is a lot of movement and the weight vectors can be ordered roughly to their final locations. In the end only the single weight vectors are

moved (fine tuning).

The **training procedure** is given as follows:

1. Normalize the input data so that all values of each single component of all input vectors have an RMS value equal to one.
2. Initialize connection weights
 - a) randomly or
 - b) using a subset of input vectors.
3. Compute the similarity between the first input vector and all connection weights by using either
 - (a) the Euclidian distance $d = \sqrt{(x_1 - w_1)^2 + (x_2 - w_2)^2 + \dots + (x_N - w_N)^2}$.
If $d = 0$ then the input vector is equal to the weight vector.
If $d \neq 0$ then the input vector points into another direction as the weight vector.
 - (b) or the vector product $\vec{x} \cdot \vec{w}$.
If $\vec{x} \cdot \vec{w} = 0$ then the two vectors are orthogonal.
If $\vec{x} \cdot \vec{w} = 1$ then the two vectors are parallel.
4. The neuron having the connection weights with the highest similarity to the input vector is the winning neuron (“winner takes all”).
5. Move the winning neuron in the direction of the input vector and also the neighboring neurons according to a neighborhood function with a certain radius.
6. Decrease the radius of the neighborhood function (i.e. the radius of the neurons that are updated along with the winning neurons).
7. Feed the neural net with the next input vector and go back to step 3.

Repeat this procedure until convergence, e.g. until the error between the input data and the corresponding neuron representing their class falls below a certain threshold.

Using the trained SOM

Once the SOM is trained, it can be used to cluster the training data or other data. For the backpropagation network it does not make much sense to test the net with the training data, since we know the result already, and we want the network to generalize to new data, making predictions using physical laws extracted from the training data.

The SOM behaves differently: we do not have a desired output, since it is an unsupervised learning technique. However, we have to extract the information on how the data have been clustered from the SOM. This is done as follows (see also Figure 2.9):

The neurons of the SOM feature map are colored according to the clusters that have emerged from training. A means of visualizing these clusters is the unified-distance-matrix (U-matrix) U_{ij} . It contains the mutual distances d_{ij} between the weights of all neighboring neurons for $i \neq j$:

$$U_{ij} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \dots & d_{1j} \\ d_{21} & d_{22} & d_{23} & \dots & d_{2j} \\ d_{31} & d_{32} & d_{33} & \dots & d_{3j} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{i1} & d_{i2} & d_{i3} & \dots & d_{ij} \end{bmatrix}, \quad (2.20)$$

for a feature map with $i \times j$ neurons. The values of U_{ij} for $i = j$ are the normalized standard deviations of all weights connected to this neuron. Figure 2.8 shows an example of a U-matrix, where the locations of the neurons are marked with little white squares. A cluster is defined by an area on the U-matrix where the distances of the weights connected to neighboring neurons is less than a specified threshold value. As soon as the distance exceeds this threshold, a new cluster begins. Every cluster is then given a different color from the available color table.

Each of these colored neurons is associated with a weight vector \vec{w} , whose components are displayed as little spheres of various sizes in Figure 2.9. If we present an input vector \vec{x} , which has the same number of components as \vec{w} , to the SOM, it is compared to all the weight vectors. The neuron with the most similar weight is the winning neuron and the input vector \vec{x} is tagged with the color of this neuron. In this way all the input vectors get an individual color code, based on where on the map they are classified to.

However, to perform a decisive clustering analysis, some of the input vectors have to be labelled. In the case of differentiating primary events from multiple events, we have

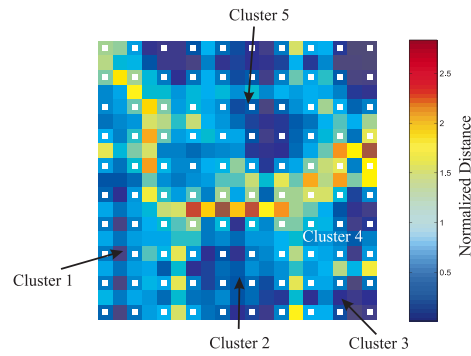


Figure 2.8: Example of an U -matrix or unified distance matrix, which contains the distances between neighboring neurons. It is used to visualize the cluster structure of the map.

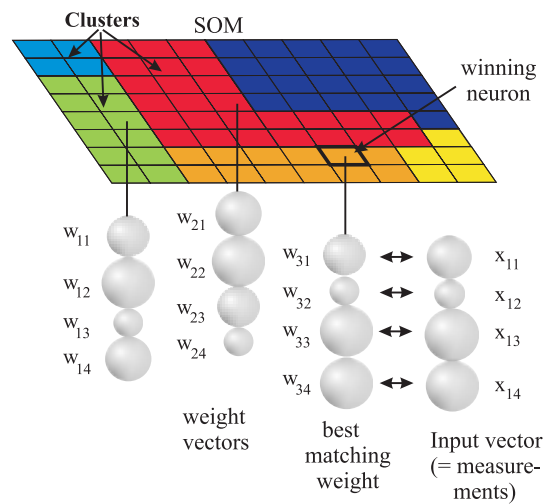


Figure 2.9: Applying the trained SOM: the neurons of the feature map are colored according to the clusters defined by the U -matrix, and each of them is associated with a weight vector, here with 4 components corresponding to a 4-component input vector. The input vector gets the color of that neuron whose weight is most similar to this input vector.

to know that some events in the data are definitely primaries and others multiples. This gives us the possibility to label the various areas on the trained SOM, and classify all the other events, not knowing before if they are primaries or multiples.

2.2.2 SOM Application Example

This simple example demonstrates the use of the self-organizing map. The eight training patterns are the coordinates of the vertices of a three-dimensional cube of size $2 \times 2 \times 2$ centered at the origin (see Figure 2.10 a), with vertex coordinates from $(-1, -1, -1)$ to $(1, 1, 1)$. The self-organizing map is a two-dimensional 16×16 grid of neurons. The determination of the size of the map is somewhat intuitive, but can be constrained by the expected number of classes. If it is too small, classes could overlap, whereas for large networks training time will increase quite fast. The three-dimensional input vectors are mapped to different positions on the two-dimensional map. Since this method belongs to the class of unsupervised learning, the training patterns consist only of input data and no desired output data.

The network orders the input vectors conserving the relationship to each neighbor class. If the corresponding contours are plotted (Figure 2.10 c), d) and e)) we can see that the vertices of the cube are ordered in a way that different sides of the cube are separated (upper side - lower side, right side - left side, front side - back side).

What we see in these Figures are the component maps. These component maps show the magnitude of the connection weights between only one component of the input vector and all feature map neurons. So, in this example, the first component map (Figure 2.10 c)) shows the weight amplitudes for the z -coordinate of the input patterns (the vertices of the cube). The second component map (Figure 2.10 d)) shows the weight amplitudes for the y -coordinate, and the third (Figure 2.10 e)) for the x -coordinate.

Another illustrative example, using the MATLAB Toolbox for Self-Organizing Maps written by the SOM working group of the Helsinki University of Technology (Kohonen, 1997b), is shown in Figure 2.11. The 3-dimensional input data we want to cluster on a 2-dimensional feature map are the x -, y -, and z -coordinates of random data points, which are roughly located in three cubic volumes in 3-D space (see Figure 2.11 a)). Is the SOM able to separate the three different clusters on a map with one dimension less than we have in the data? The answer to this question is very interesting, because it tells us if

the dimension of the information in the data really is 3, or if it actually only is 2. Figure 2.11 b) gives the answer: On the upper left panel we display the U-matrix or unified distance matrix, which contains the distances between neighboring units. It is used to see the cluster structure of the map. It contains information from all components. As we can see the upper part of the U-Matrix forms a cluster separated from the two clusters in the lower part. So the SOM was able to separate the data into three clusters (separated by light blue areas) on the 2-D map. The other three panels show the component maps, already explained above. Also here we can see three distinct regions on every component map.

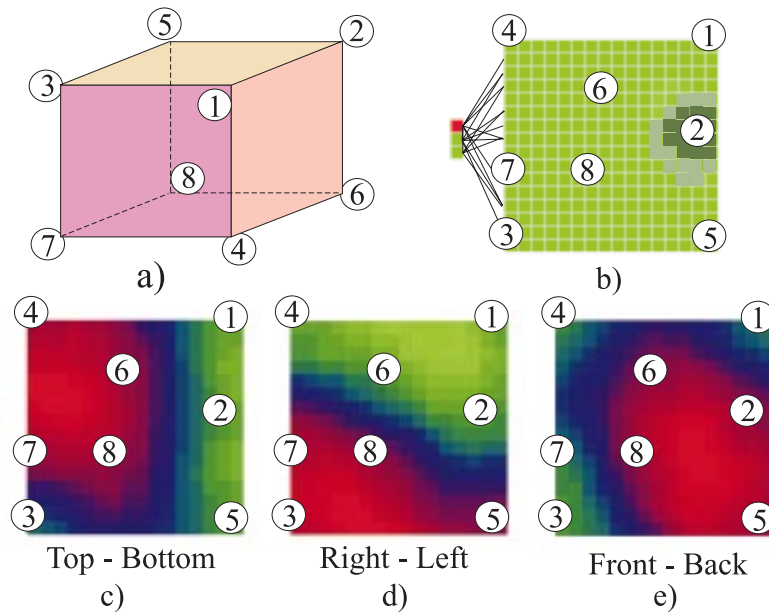


Figure 2.10: a) The coordinates of the eight vertices of this cube form the input for the network. b) The trained network shows the result for vertex no.2 as input. The numbers represent the winning neurons for the corresponding input patterns. c) d) and e) show component maps: c) 1,2,3,5 form the upper side and 4,6,7,8 the lower side of the cube, i.e. here the z-component is shown. d) 1,2,4,6 form the right side and 3,5,7,8 the left side of the cube, i.e. here the y-component is shown. e) 1,3,4,7 form the front side and 2,5,6,8 the back side of the cube, i.e. here the x-component is shown.

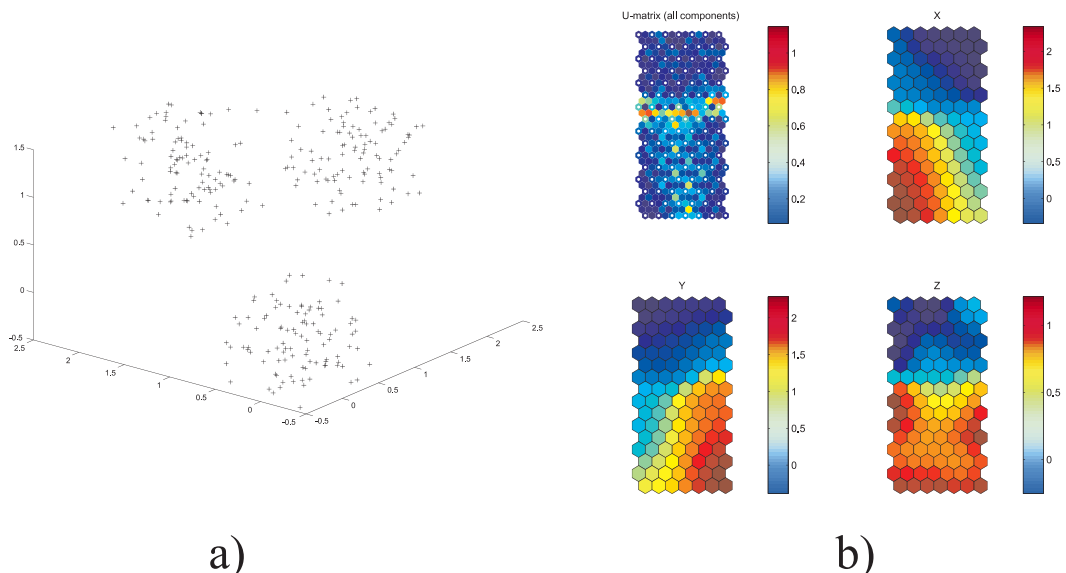


Figure 2.11: a) Input data for the SOM: three clusters of random points within cubes of fixed size. b) Top left: U-Matrix, showing the SOM feature map with the colors representing the relative distance of the corresponding neuron to the neighboring neurons. Top right: component map for the x-coordinate. Bottom left: component map for the y-coordinate. Bottom right: component map for the z-coordinate. (Remark: in this example a hexagonal grid instead of a rectangular one was used.)

Chapter 3

Seismic Attributes

Since Taner, Koehler & Sheriff in 1979 introduced complex seismic trace analysis in Geophysics, the development of seismic attributes has found a multitude of applications, especially for reservoir characterization. A recent publication (Chen and Sidney, 1997) listed over a hundred different attributes. In general, all quantities that we can compute from seismic measurements - 2-D or 3-D, prestack or poststack, time-migrated or unmigrated - are seismic attributes. Many of them have a physical meaning, and only those will be considered in this work. They vary from the instantaneous amplitude of the seismic trace to such computationally complex parameters as for example the radius of curvature of a normal-incidence wavefront.

Physical attributes are computed from seismic traces. The theoretical model of this seismic trace consists of a reflectivity series that is the impulse response of an elastic or acoustic model of the subsurface with the corresponding elastodynamic (equation 1.1) or acoustic wave equation underlying it. This reflectivity series convolved with the source wavelet is defined to be the seismic trace. Each seismic trace is a recording of information from the subsurface. Physical attributes constitute a means of extracting this information without applying complicated inversion procedures. Important properties such as energy transport, phase changes, frequency content, amplitude attenuation, wavefront characteristics, and others, provide us with the necessary information to distinguish between individual parts of the seismic wavefield.

An important class of attributes is based on the complex seismic trace, and will be described next. Another set of attributes, the wavefront attributes, are treated in a later

section (3.3). The complex seismic trace, or analytic signal (Taner et al., 1979) is defined as

$$F(t) = r(t) + ig(t), \quad (3.1)$$

where the real part $r(t)$ is the recorded seismic signal and the imaginary part $g(t)$ is the Hilbert-transform of $r(t)$ (also called the quadrature trace). The Hilbert-transform (Bracewell, 1965) can be realized by an allpass filter with a 90° - phase shift in the time domain, for example.

Attributes computed from the complex seismic trace can be subdivided into instantaneous (referring to an instant of time) and wavelet attributes (referring to a time window).

3.1 Instantaneous Attributes

Instantaneous attributes are computed directly from the seismic trace. The ones used in this dissertation are:

1. Envelope

$$E(t) = \sqrt{r^2(t) + g^2(t)} \quad (3.2)$$

represents the total instantaneous energy and describes the energy flux in the subsurface. It is independent of the phase of the signal.

2. First derivative of the envelope

$$\frac{dE(t)}{dt} \quad (3.3)$$

represents the rate of change of the envelope, and tends to show the onset of wavelets in a seismic trace.

3. Second derivative of the envelope

$$\frac{d^2E(t)}{dt^2} \quad (3.4)$$

tends to show all peaks of the envelope, thus emphasizing all reflections in the seismogram.

4. Instantaneous phase

$$p(t) = \arctan\left(\frac{g(t)}{r(t)}\right) \quad (3.5)$$

Using the quantities above the analytic signal can be also expressed as

$$F(t) = E(t)e^{ip(t)} \quad (3.6)$$

leading to the alternative formulation

$$p(t) = \Im [\log F(t)], \quad (3.7)$$

where \Im denotes the imaginary part and \log is the principal value of the complex logarithm. The instantaneous phase is a measure of continuity of the propagating wave in the medium. At every discontinuity the wave suffers a phase change.

5. Instantaneous frequency

$$f(t) = \frac{dp(t)}{dt} \quad (3.8)$$

represents the temporal rate of change of the instantaneous phase. For practical applications the difference formula is preferable:

$$\begin{aligned} \frac{dp(t)}{dt} &= \Im \left\{ \frac{1}{F(t)} \frac{dF(t)}{dt} \right\} \\ &\approx \frac{2}{\Delta t} \Im \left\{ \frac{F(t) - F(t - \Delta t)}{F(t) + F(t - \Delta t)} \right\} \\ &= \frac{4}{\Delta t} \frac{r(t - \Delta t)g(t) - r(t)g(t - \Delta t)}{(r(t) + r(t - \Delta t))^2 + (g(t) + g(t - \Delta t))^2}, \end{aligned} \quad (3.9)$$

with Δt the sample rate. The instantaneous frequency often shows large fluctuations, especially in areas with low signal-to-noise ratio, but indicates every change in the phase behavior. Therefore, it is used as an indicator for the bedding thickness of the layers.

6. Envelope weighted instantaneous frequency

$$\frac{\sum_T [E(t)f(t)]}{\sum_T E(t)} \quad (3.10)$$

where T is the window length, depending on the desired degree of smoothing. This is a smoothed version of the rather rapidly varying instantaneous frequency attribute.

7. Acceleration of phase

$$\frac{df(t)}{dt} = \frac{d^2p(t)}{dt^2} \quad (3.11)$$

shows jumps in the instantaneous frequency, which indicates events arriving close after another. Thus, this attribute is often used as a thin-bed indicator.

8. Dominant frequency

$$\hat{f} = \int_{f=0}^{\infty} f P(f) df / \int_{f=0}^{\infty} P(f) df, \quad (3.12)$$

where $P(f)$ is the power spectrum of the signal and where we assume that $P(f)$ is uni-modal, or

$$\hat{f} = \frac{1}{\hat{T}}, \quad (3.13)$$

with \hat{T} denoting the dominant period measured between two successive peaks or troughs.

9. Bandwidth

$$bw(t) = \int_{f=0}^{\infty} (f - \hat{f})^2 P(f) df / \int_{f=0}^{\infty} P(f) df \quad (3.14)$$

$$bw(t) = \frac{dE(t)}{dt} / (2\pi E(t)) \quad (3.15)$$

For a propagating wave, the earth acts as a low-pass filter. Reflections coming from deep reflectors are lacking the high frequencies, whereas the bandwidth for shallow reflections is considerably broader. Therefore, also multiples should show a tendency to larger bandwidth, compared to primaries appearing at the same zero-offset time.

10. Instantaneous Q-factor

$$Q(t) = \frac{\pi f(t)}{\kappa v} \quad (3.16)$$

with the absorption coefficient κ and the velocity v . The attenuation of the amplitude of seismic waves in elastic media is frequency-dependent. Thus, we can expect a different Q-factor for primaries and multiples, since they often have different frequency content (or bandwidth).

11. Apparent polarity

Defined as the relation of a peak or trough of a seismic reflection to the sign of the reflection coefficient, i.e. a peak would indicate a positive reflection coefficient and vice versa. (Sheriff, 1991).

3.2 Wavelet Attributes

The seismic trace is a superposition of a number of *wavelets*. Their temporal length can be defined as the time from one minimum of the envelope of the seismic trace to the next. We compute only one attribute value per wavelet. Wavelet attributes are thus the instantaneous attributes picked at the envelope peaks.

The formulas for the wavelet attributes I use are the analogs of the instantaneous attributes listed in the previous section.

3.3 Wavefront Parameters

Stacking Velocity

In the seismic processing flow there is an important step called velocity analysis. It is used for creating a macro-velocity model that consists of the stacking velocities for selected reflection events. Stacking these events with the correct stacking velocity (which is related to NMO and root-mean-square (RMS)-velocity) in the common-midpoint (CMP) gather produces a simulated zero-offset (ZO) section. Conventional velocity analysis is based on the assumption that an event in a CMP gather has an approximately hyperbolic travel-time curve:

$$t^2(x) = t_0^2 + \frac{x^2}{v_{Stack}^2}, \quad (3.17)$$

where t_0 is the zero-offset time, x is the offset and v_{Stack} the stacking velocity.

Scanning through all relevant velocities from v_{Stack}^{min} to v_{Stack}^{max} and performing a coherency analysis (e.g., semblance) along the corresponding hyperbolae for each zero-offset time t_0 , we get one value for v_{Stack} at which the coherency measure is at a maximum. This is the optimal stacking velocity of this event and represents a very significant attribute that is used later in this thesis for primary-multiple discrimination.

In general, primary events have higher stacking velocities than multiples appearing at similar zero-offset times. However, for internal multiples created at a velocity inversion, the opposite situation occurs. An example of a velocity analysis result, the velocity spectrum, is shown along with the corresponding CMP gather in Figure 3.1.

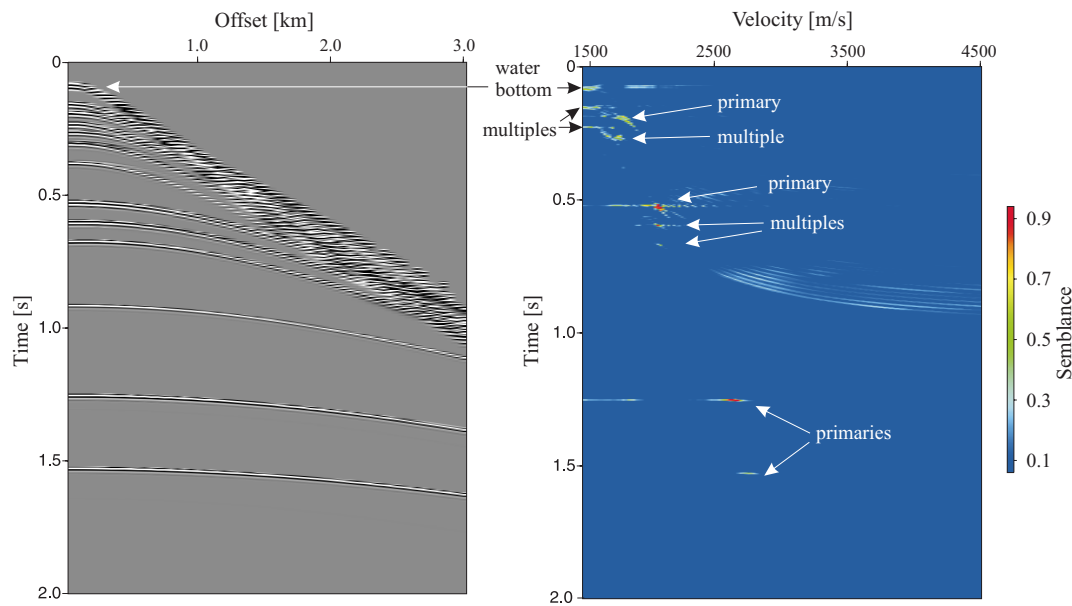


Figure 3.1: *Left: CMP gather with primary and multiple events. Right: corresponding velocity spectrum. Some events are labelled. There are some artefacts due to cutting effects in the velocity spectrum and some weak events have very low semblance.*

Emergence Angle and Radii of Curvature

Using an alternative description of rays in laterally inhomogeneous media, Tygel et. al (1997) showed that traveltimes of a reflected event can be expressed in terms of the three wavefront parameters,

1. the emergence angle β_0 of the zero-offset ray,
2. the radius of curvature of the normal-incidence-point wave R_{NIP} , and
3. the radius of curvature of the normal wave R_N :

$$t^2(x_m, h) = \left(t_0 + 2 \frac{\sin \beta_0}{v_0} (x_m - x_0) \right)^2 + 2t_0 \frac{\cos^2 \beta_0}{v_0} \left(\frac{(x_m - x_0)^2}{R_N} + \frac{h^2}{R_{NIP}} \right), \quad (3.18)$$

where t_0 is the zero-offset time, v_0 the near-surface velocity at x_0 , x_m the midpoint and h the half-offset coordinate. The NIP-wave and the N-wave are hypothetical waves introduced by Hubral (1983). Figure 3.2 shows a sketch of the important parameters and waves.

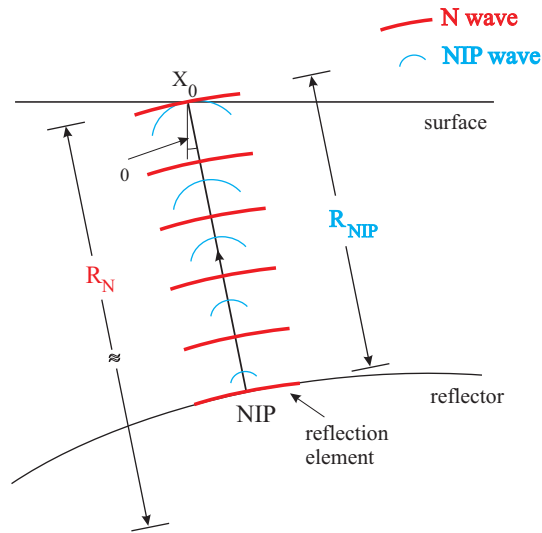


Figure 3.2: The wavefront parameters are computed at the zero-offset location x_0 . The emergence angle β_0 is measured between the normal ray (going from x_0 to the normal-incidence point NIP) and the surface-normal. R_{NIP} is the radius of curvature of the wave generated by a point source at NIP. R_N denotes the radius of curvature of a wave generated by the “explosion” of a reflector element. Here a constant velocity v_0 is assumed between surface and reflector.

Instead of having only one parameter (v_{Stack}) in the normal-moveout formula, we now have three parameters (β_0 , R_{NIP} and R_N) that provide essential information about the reflected event and can serve as attributes for the discrimination of primaries and multiples. However, this implies that we have to perform a three-parameter search over all gathers and zero-offset times instead of only scanning through one parameter in order to find the optimal stacking hyperbola. In practice this causes problems due to immense computation time (e.g. the range of the radii is $[-\infty : +\infty]$). Müller (1999) solved this problem by conducting three one-parameter searches instead of one three-parameter search. First, by searching only in the CMP domain, equation 3.18 reduces to equation 3.3, where v_{Stack} is a combination of the two parameters β_0 and R_{NIP} . When the optimal value of this combined parameter is found, a one parameter search in the ZO domain is carried out. The restriction of equation 3.18 to the ZO-configuration results in a formula

for two unknown parameters: β_0 and R_N . In a first-order approximation in $(x_m - x_0)$ this description can be further reduced to depend on the emergence angle β_0 only. After having found the optimal value for β_0 we can calculate R_{NIP} from the combined parameter determined in the first step. In a final step, we use the exact formulation in the ZO domain which is dependent on β_0 and R_N , and search for R_N to determine this last parameter.

These steps produce first estimates of the three wavefront parameters, which can now be refined by use of optimization techniques. However, tests have shown that the first estimates do not deviate much from the optimized values. Since in this thesis these parameters are used as inputs to neural networks, which generally work better and are more robust in the presence of weak noise, I decided to use the non-optimized parameter values.

Chapter 4

Multiple Prediction and Attenuation with Backpropagation Neural Networks

4.1 Trace by Trace Multiple Attenuation

To investigate the behavior of the neural network and - for comparison - the Wiener filter with the attenuation of multiple reflections in synthetic data, I generated a subsurface model and computed a number of seismograms. The model is made up of three deep reflectors. The two-way travel-time (TWT) of the primary reflection of each reflector is allowed to vary ± 100 milli-seconds. The reflectors have a variability of 0.2 in the reflection coefficients. The water surface reflection coefficient is -1. The sea bottom is at a fixed TWT of 90 ms with the reflection coefficient varying in the range [-0.5 : -0.3]. Table 1 shows the model. The relatively high reflection coefficients were chosen in order to see clearly how either the neural net or the Wiener filter affects the signal.

TABLE 1. *Synthetic model for trace-by-trace multiple attenuation.*

| <i>reflector</i> | <i>TWT [ms]</i> | <i>refl. coeff.</i> |
|-------------------|-----------------|---------------------|
| water surface | 0 | -1 |
| water bottom | 90 | [-0.5 : -0.3] |
| 1. deep reflector | 220 - 420 | [0.6 : 0.8] |
| 2. deep reflector | 550 - 750 | [-0.6 : -0.4] |
| 3. deep reflector | 760 - 960 | [0.7 : 0.9] |

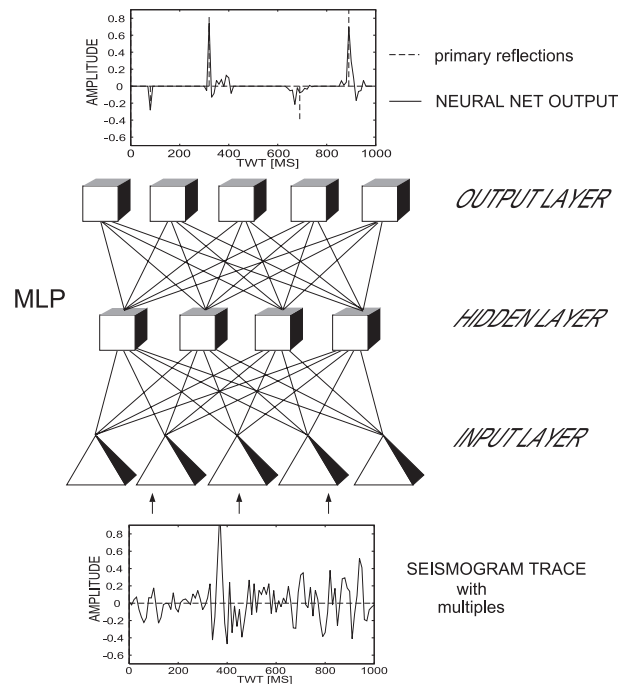


Figure 4.1: Training of the neural net (multi-layer perceptron, MLP): Presentation of the training set consisting of 100 seismic traces containing free-surface and internal multiple reflections and different amounts of noise at the input layer and simultaneously providing the desired output, i.e. the arrival time and reflection strength of the primary reflections.

By propagating a vertically incident plane wave through different reflector configurations (i.e. depth and reflection coefficient are chosen randomly within their respective ranges), a set of 200 different seismograms was calculated, convolved with a zero-phase wavelet, perturbed by various amounts of noise, and finally split into one set of 100 training patterns and another set of 100 test patterns. A pattern is defined here as a single seismic trace. Figure 4.1 at the bottom shows one example of such a seismic trace, containing free-surface as well as internal multiples (cp. section 1.1). The signal-to-noise ratio is $S/N = 2.0$ in this example.

As depicted in Figure 4.1, the neural net input is the seismic trace containing free-surface related and internal multiple reflections and noise, and the desired output is the seismic trace with only the primary reflection events, which are known from the given synthetic model. According to the problem, the neural network consists of 100 input neurons, between 10 and 30 hidden neurons, and 100 output neurons. The training time

varied according to the contamination of the seismograms with noise. For the noise-free training set about 100 to 200 training epochs were sufficient, whereas for the noisy data sets (e.g. $S/N = 0.6$) the network needed up to 1000 such epochs.

The neural net output gives the arrival times as well as the reflection strengths of the desired primary reflections. However, the amplitude characteristic of the seismic wavelet is destroyed. Thus, this method is not applicable if in a subsequent processing step true amplitudes of the seismic trace are required, although the reflection strengths of the individual reflectors are reproduced quite reliably.

The set containing the 100 test patterns was not only presented to the trained neural net but also to a Wiener prediction filter (Robinson and Treitel, 1980), with prediction distance $\alpha = 80$ ms, and of length 300 ms. This parameter choice proved to yield the best results for this data set.

Figure 4.2 shows the result for a given trace for neural net testing on the left, and for Wiener filtering on the right. The figure shows one example out of 100 test traces with signal-to-noise ratios ranging from the noise free case (trace 2) to a S/N ratio of 0.6 (trace 5). Trace 1 is the desired output, i.e. the actual reflectivity series for this example.

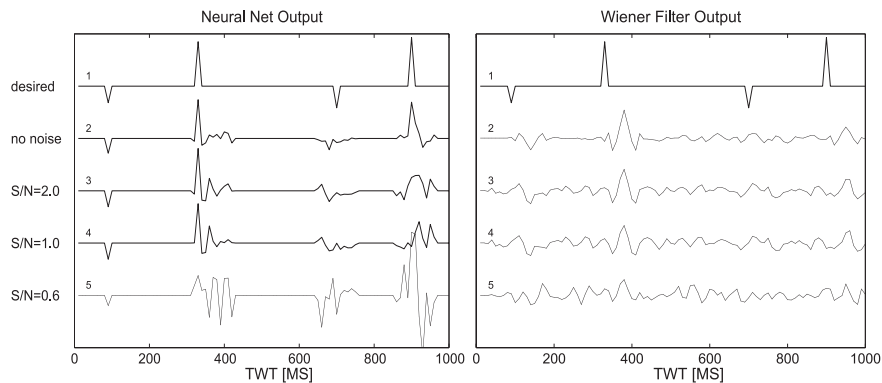


Figure 4.2: *The neural net performance (to the left) and the Wiener filter (predictive deconvolution) performance (to the right) on deconvolution for synthetic seismograms with various amounts of noise. Only one example out of 100 is shown for each signal-to-noise ratio. Trace 1 is the true reflectivity, traces 2 - 5 show the neural net output or Wiener filter output, respectively.*

The performance of the neural net and of the Wiener filter for the whole test set is pictured in Figure 4.3, which shows the percentage of correctly detected reflectors from

all 100 test traces. The percentage of correct neural net detections is plotted versus the percentage of correct Wiener filter detections. The 45-degree line separates the area where the neural net performance is better (white area) from the area where the Wiener filter yields higher detection rates (gray area). On the two diagrams, significantly more dots lie in the white area, demonstrating that the neural net shows an overall better performance. The left part of Figure 4.3 shows the performance with respect to the different amounts of noise in the data. It can be seen that with increasing amount of noise, the performance of both the neural net and the Wiener filter generally becomes worse. However, the noise factor is not as crucial for reliable event detection as the travel-time of the respective reflection event. The right part of Figure 4.3 depicts the performance with respect to the four different reflectors. The water bottom (o), for example, is always detected to 100% by the neural net, while the Wiener filter only scores 100% for the noise-free case (68%, 51%, and 37 %, respectively for increasing noise).

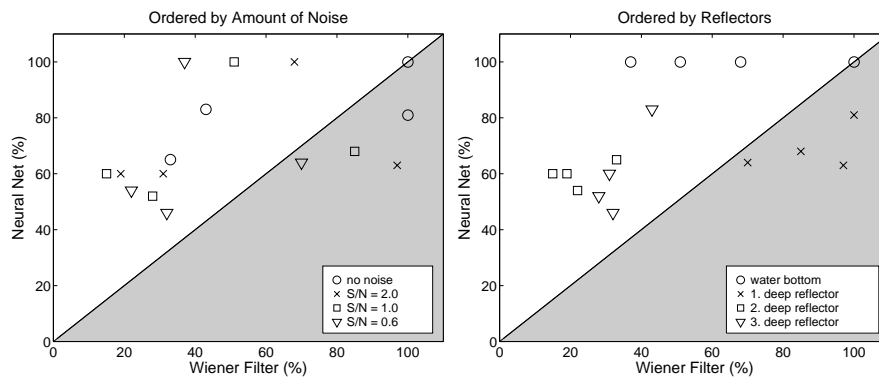


Figure 4.3: The neural net performance plotted versus the Wiener filter (predictive deconvolution) performance in % of correctly detected events. Left: detection percentage ordered by noise content of the seismic data. Right: detection percentage ordered by reflector depth. The 45-degree line separates the area where the neural net performance is better (white area) from the area where the Wiener filter yields higher detection rates (gray area).

To create the plots of Figure 4.3, each output trace was divided into windows centered at the true reflection event and only the neurons with the highest output in the corresponding windows were taken into account. The diagram displays how often (in %) the neural net activated the correct neuron (i.e. within a deviation of ± 1 time samples from the correct value). Since the water bottom does not change depth, it is detected with a score of 100% by the neural net. This provides a criterion to assess the proper functioning of the

algorithm. The Wiener filter, on the other hand, does a better job only for the shallowest deep reflector (\times). This reflector has a very high reflection coefficient and thus the reflection event is not distorted much by multiples. The second deep reflector (\square) is the hardest to detect correctly for both the neural net as well as for the Wiener filter because the signal here directly interferes with a strong multiple. Around 60% of the neural net picks lie within a range of ± 1 time samples around the true value, whereas the Wiener filter performance only lies between 18% and 35%. For the third deep reflector (∇) the neural net shows detection rates that are similar to those for the second deep reflector. The Wiener filter also performs better than for the second reflector, possibly because there is not as much interference with multiple energy.

For the noisy data sets the neural net performance, as well as the Wiener filter performance, decreases with increasing noise levels. It has to be noted that the score for the Wiener filter always has its peak value at a deviation of 50 ms from the correct value (see the traces in the right panel of Figure 4.2). The reason for this is the difficulty in picking the onset of a seismic wavelet, which the Wiener filter tries to restore in the filter process. Therefore I decided to pick the peak value of the wavelet, which is at a constant offset of 50 ms. The output from predictive deconvolution has to be processed further with a spiking deconvolution operator in order to achieve even sharper events.

Due to its high reflection coefficient, the first deep reflector is detected quite reliably by the Wiener filter. However, in the presence of noise it fails almost completely for reflectors at greater depth, whereas the neural net still shows a clear trend for correct detection. It is hard to give an objective measure of comparison between the results of the neural net on the one hand and the Wiener filter on the other, because in seismics the success of a processing step is often judged by visual inspection. Criteria that involve input data for success assessment are biased by the underlying assumptions of the algorithm and thus cannot really provide an independent measure (an example is the residual wavelet). Thus, if a method reveals subsurface structure or information that was previously hidden or distorted, it is rated as successful, even if the overall change in the data is very small.

Table 2 shows the correlation coefficients between the desired and actual output for the neural net and the Wiener filter. The tabulated coefficients represent average values for the entire test set of 100 traces.

TABLE 2. Correlation coefficients between the desired and actual output for the neural net and for the Wiener filter. The tabulated coefficients represent average values for the entire test set of 100 traces.

| | <i>neural network</i> | <i>Wiener filter</i> |
|-------------|-----------------------|----------------------|
| no noise | 0.67 | 0.49 |
| $S/N = 2.0$ | 0.36 | 0.35 |
| $S/N = 1.0$ | 0.28 | 0.28 |
| $S/N = 0.6$ | 0.27 | 0.25 |

This test represents a situation where the Wiener filter should have performed perfectly, since all assumptions were satisfied: the multiples were strictly periodic, the reflections sparse, and the wavelet was not changing its shape. In the presence of noise the physical model underlying the Wiener filter theory does not describe the situation precisely enough.

4.2 Multiple Attenuation with Neural Net Ensembles

In real seismic data processing we not only deal with seismograms from vertically incident plane waves (as in the previous section). We normally record seismic data in shot gathers with offsets from 100 m up to a few km. A typical shot gather is shown in Figure 4.4, right. Now the attenuation of multiple reflections in marine seismic sections is shown for an entire profile by training an ensemble of neural networks. I demonstrate the method on a synthetic data set and train different neural nets for different offsets in the seismograms. This corresponds to performing neural net deconvolution in the common-offset domain. After deconvolving all common-offset gathers, the data are resorted into common-midpoint gathers and then stacked.

To test the performance of neural network deconvolution on synthetic data, I created a subsurface model, consisting of several reflectors. The sea bottom dips from 200 m to 300 m in depth, and the six deeper interfaces have the form shown in Figure 4.6, left. The total depth of the model is about 4 km. This 2-D model was then assumed to be horizontally stratified locally (within 4-km intervals in x -direction). This was done in order to use

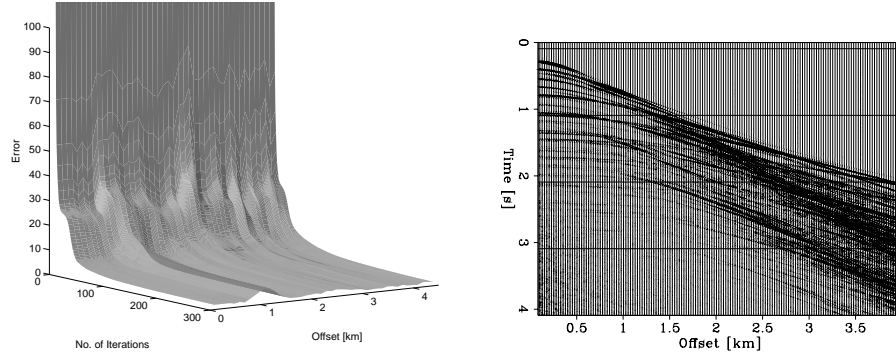


Figure 4.4: *Left: convergence behavior of all 40 neural networks. Right: a typical shot gather containing free-surface and internal multiples and wave conversions.*

the reflectivity method (Fuchs and Müller, 1971), which requires a 1-D subsurface geometry with $v_P(z)$, $v_S(z)$ and $\rho(z)$ for computing the reflection seismograms. I calculated a set of 100 reflection seismograms (CMP gathers), containing free-surface multiples, internal multiples and converted waves, which were then resorted into common-offset (CO) gathers.

The reflectivity method can switch on or off the various kinds of multiples and/or conversions. The reflectivity matrix $\mathbf{r}_i^{D/U}$ contains the reflection coefficients r_i for the i th layer for downward (D) or upward (U) propagation of the wavefield given in the form:

$$\mathbf{r}_i^{D/U} = \begin{pmatrix} r_{PP,i} & r_{PS,i} \\ r_{SP,i} & r_{SS,i} \end{pmatrix}, \quad (4.1)$$

with PP denoting incident and reflected P-wave, PS incident P and reflected S-wave, SP incident S-wave and reflected P-wave, and SS incident and reflected S-wave.

If we want to switch off internal multiples, we have to set the reflection coefficients for downward reflection

$$r_{PP}^D, r_{SP}^D, r_{PS}^D, r_{SS}^D = 0 \quad \text{for all layers } i.$$

Setting the respective downward reflection coefficients of the free surface to zero, produces a seismogram without free-surface multiples. This method is therefore a practical tool for testing neural network multiple attenuation, since it permits to produce input data (full wavefield) and desired output data (primaries only) with the same algorithm. The section with the smallest offset containing free-surface and internal multiples and conversions is shown in Figure 4.5, left. This is the input for a given neural network. The stack of all CMP gathers containing only primary reflections, which is the desired result

for the deconvolution efforts, is shown in Figure 4.6, left. Figure 4.5 (right) shows a stack of the data after normal move-out (NMO) correction with the known velocity model. If the section in Figure 4.5, left is compared with this stacked section, we observe that many multiple reflections have already been attenuated. However, it was assumed that we know the subsurface velocity model. In the very common case when we do not have a correct velocity model available, the NMO-stack yields far worse results. If we have some information about the subsurface from well-logs in the region, the neural net method produces better results when we do not know the exact 2-D velocity model for the whole profile.

For suppressing the multiple energy in the data an ensemble of neural networks was designed to perform deconvolution in the common offset domain. This makes more sense than processing in the CMP domain, where the neural net would have to average over all offsets. Since the structure of the information varies greatly from near to far offset, a neural net would have to adapt to a large pattern space, which leads to poorer results. This latter way of processing would be analogous to the application of a single prediction filter to all offsets in the conventional case.

After resampling the data from a 4 ms to a 16 ms time interval, and from a spatial interval of 25 m to one of 100 m, 40 sections were obtained, each containing 100 traces. These data were split into a training set containing every 4th trace of each section and a test set containing the remaining traces. This would correspond to a real world situation where we have well-log information from boreholes every 4 km, with the aim to interpolate the subsurface structure from the seismic data between the boreholes. Although such well-spacing exists in some oil fields, in general the number of boreholes is very small, since wells are expensive to drill. My experiments suggest that it is not necessary to have more than a few boreholes available. A neural network was trained with seismograms that were computed from artificial variations of real well-log data and achieved good results. The well-log variations should mimic the geologic situation in the investigation area, since a reflector that does not appear in the training data is unlikely to be detected. On the other hand, the neural net is not a mere interpolator between well-log data in the traditional sense. It learns to extract the physical laws and relationships leading to subsurface information from seismic data, and additionally makes use of the available geologic data from borehole measurements (sonic and density logs).

For the different offsets in the CMP gathers different neural networks were trained

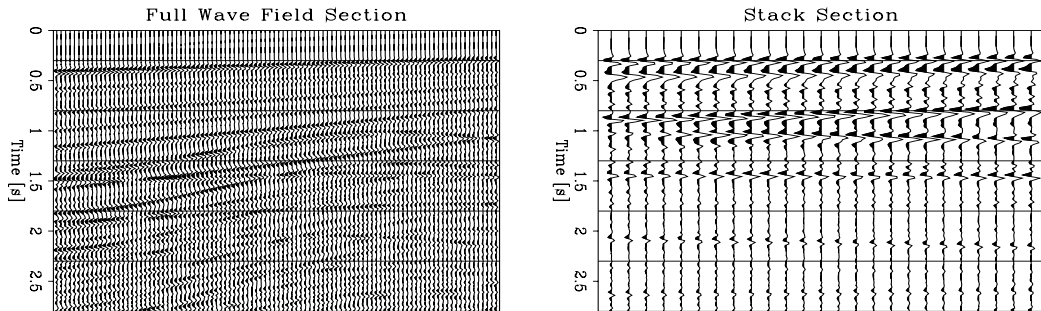


Figure 4.5: Left: input for the neural network (CO gather of the nearest offset, containing the full wavefield). Right: NMO-stack of the input data using the known velocity model.

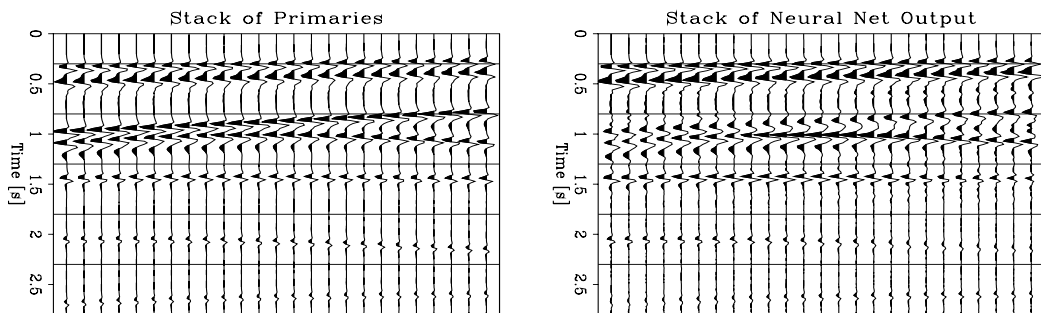


Figure 4.6: Left: stack of the CMP gathers containing primaries only (desired output). Right: after neural network deconvolution (stack of the deconvolved CMP gathers).

- one for each offset. The Rprop algorithm (Riedmiller and Braun, 1993, chapter 2) proved to be most efficient for this task. I tested different network configurations, of which a three-layer network with one hidden layer converged to the lowest minimum and produced the best results. The stack of the neural net output is shown in Figure 4.6, right.

4.3 Conclusions

Several deep primary reflections were detected, and multiple energy was attenuated significantly. The convergence behavior of all networks for the different offsets is shown in Figure 4.4, left: The training error is plotted versus the number of iterations and the offset (in the CMP gathers). Convergence normally was reached after already 100 iterations, but for offsets around 1 km the error remained at a higher level. For this offset many reflection hyperbolae intersect, so that the information provided to the neural net is ambiguous (compare with the shot gather in Figure 4.4, right).

This example demonstrates that the neural network can detect primary reflection events in data highly contaminated by multiple energy. Even the shape of completely hidden deep reflectors was revealed, given some information from well-logs. This specific method cannot work without any subsurface information from sonic or density logs. The borehole data represent constraints for the inversion of the seismic data, while the seismic data is used to infer the structure in the time section between the wells.

Chapter 5

Attribute based Multiple Prediction and Attenuation with Backpropagation

5.1 The Method

An alternative approach to the method of trace-by-trace multiple attenuation described in the previous chapter is the attribute based multiple prediction and attenuation method, developed within the scope of this thesis. All traditional multiple attenuation techniques based on filtering have one thing in common: they remain in one parameter domain and depend on the assumption that the transformation of the data into this domain separates primary from multiple reflections (e.g. $f - k$ filtering). The idea of the method described in this chapter is to combine the different discriminatory powers in the various domains and use meaningful attributes from each data space. If a multiple cannot be separated from a primary in the velocity spectrum, this might be possible in the f - k domain or vice-versa. Feeding the neural net with selected attributes from as many different domains as possible leaves the decision with the neural net to determine which combination of attributes corresponds to a primary or to a multiple.

This approach has the beauty that, depending on the choice of the desired output, we can do multiple prediction or alternatively multiple attenuation. Figure 5.1 depicts the principle. The input is always a number of carefully selected attributes computed from the full wavefield seismograms. If we have an estimate of free-surface and/or internal multiples available, e.g. from other multiple prediction methods, these are used as the

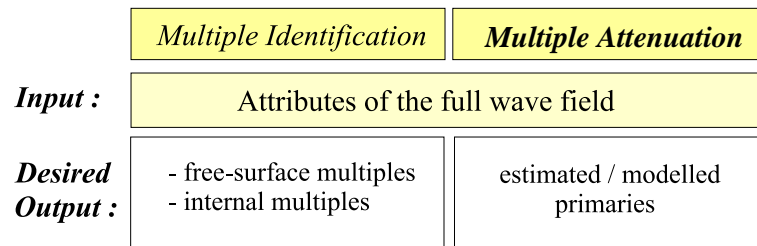


Figure 5.1: *Multiple identification and attenuation with backpropagation neural networks:*

desired output and the algorithm predicts the desired type of multiple. Modelling on the other hand can be used to produce sections with primary information. Using these as the desired output provides the possibility of multiple attenuation.

5.2 Multiple Attenuation in a Simple 1-D Elastic Data Set

For a simple elastic synthetic data set (a representative shot gather is shown in Figure 5.3 on the left hand side), generated with the reflectivity method (Kennett, 1979) which is based on the elastodynamic wave equation in 1-D media the following instantaneous attributes (cp. chapter 3) were computed for each time sample:

- the amplitudes of the five near-offset traces in each seismic section,
- the travel-time,
- the envelope of the first trace,
- the maximum value of the velocity spectrum (semblance) for each time sample,
- the peak location in the velocity spectrum for each time sample (i.e. the velocity corresponding to the highest peak in the spectrum),
- a horizontal window from the velocity spectrum for each time sample,
- the instantaneous phase, and
- the instantaneous frequency.

This resulted in an input vector with 58 components for each time sample. For this data set, the instantaneous attributes were used and produced the most consistent results. The principle of my approach is shown in Figure 5.2. All attributes were normalized in such a way that each attribute had zero mean and a standard deviation of 1. The velocity spectrum is computed using semblance as the coherency criterion, which is the ratio of the output to the input energy of an M-channel signal f within a time gate of length N (Neidell and Taner, 1971):

$$S = \frac{\sum_{j=1}^N \left\{ \sum_{i=1}^M f_{i,j(i)} \right\}^2}{M \sum_{j=1}^N \sum_{i=1}^M f_{i,j(i)}^2}. \quad (5.1)$$

Attributes derived from this velocity spectrum are particularly meaningful since the move-out information is contained there. The different moveout of a primary and a multiple manifests itself in different stacking velocities and thus in distinct spots in the velocity spectrum:

$$\text{Primary:} \quad t^2(x) = t_0^2 + \frac{x^2}{v_{primary}^2} \quad (5.2)$$

$$\text{Multiple:} \quad t^2(x) = t_0^2 + \frac{x^2}{v_{multiple}^2} \quad (5.3)$$

All seven reflectors were picked from the synthetic sections containing only the primaries, and used this information as the desired output for the training. Then I split the whole data set into a training and a test set, and trained a network with 58 input neurons (for the 58 attribute values), 8 to 20 hidden neurons and one output neuron. After several tries with different configurations this network showed best convergence and produced the best results. The single-node output of the net can be configured to deliver a “one” for the presence of a primary event and a “zero” for its absence. However, to obtain a feeling for network classification reliability, I allowed a continuous output instead of the hard-limited zero or one. The results for one example of a shot gather are shown in Figure 5.3.

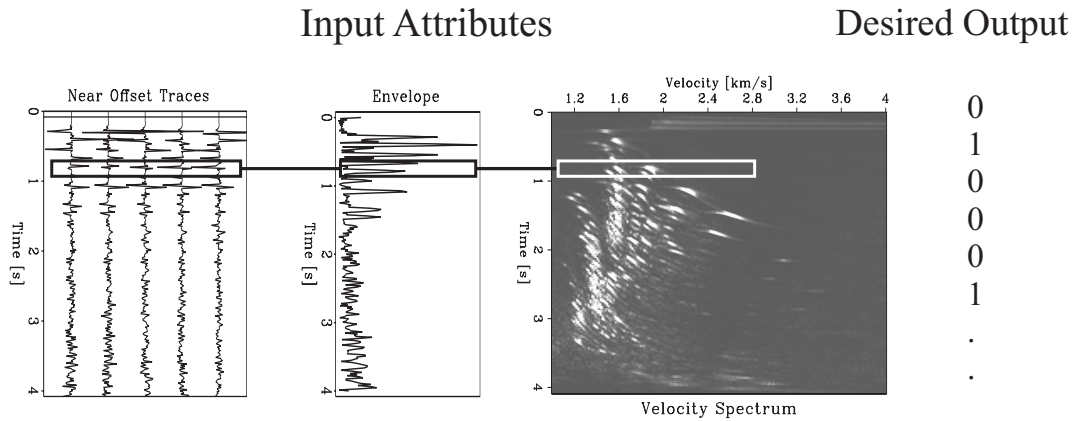


Figure 5.2: Attribute based multiple attenuation. The input consists of a number of attributes from different domains, e.g. here are shown the amplitude of 5 near-offset traces, the envelope of the first trace, and the velocity spectrum from which different attributes were computed. The desired output for each event is 1 for a primary and 0 for a multiple.

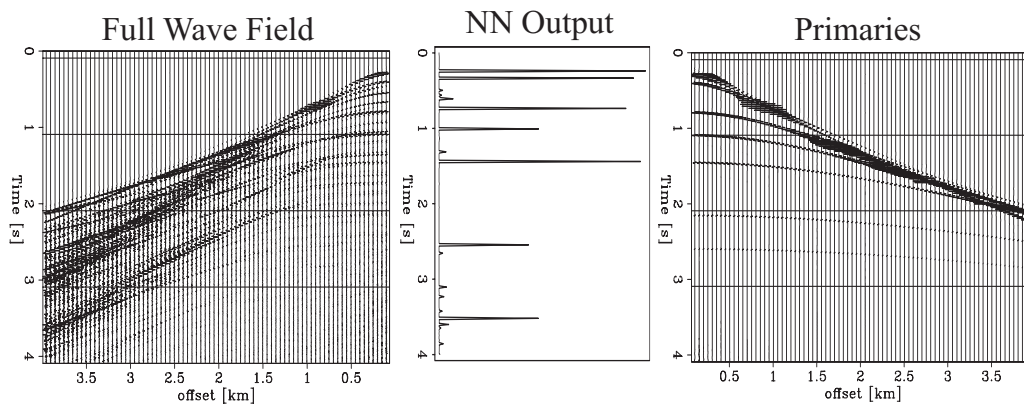


Figure 5.3: Results for one typical shot gather using the attribute based multiple attenuation method. The left seismogram shows the raw data with all multiples and converted waves. On the right seismogram we see the desired output, i.e. the primaries. In the center the corresponding classification result of the neural net is displayed. It reliably detected the upper five reflection events and also the lowest one, but failed on the sixth event. The lowermost spike is a misclassification of a multiple.

5.3 Multiple Attenuation and Prediction in a 1-D Realistic Elastic Data Set

This section shows the application of the attribute based neural net multiple identification and attenuation method on an elastic full wave synthetic CMP gather shown in Figure 5.4, left. It was modeled on the basis of well-log data, using the reflectivity method. The synthetic CMP gather, containing the primary P-waves only (Figure 5.4, right), was used as the desired output information.

For obtaining attributes for the neural net training, the velocity spectrum was computed, and the location and height of the peaks for each zero-offset time was determined. In addition, I calculated the envelope of the first trace of the input CMP gather. A window of length 5 samples from the first trace of the full wave CMP gather (the signal amplitude) was also used as input for the neural net. The 5-samples window is about the length of a wavelet in this data set.

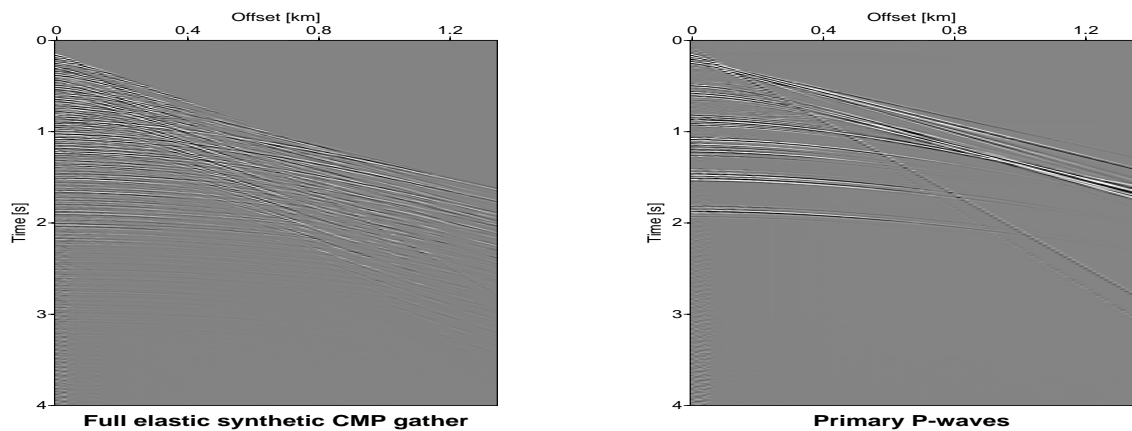


Figure 5.4: *Left: full elastic synthetic CMP gather. Input for the neural net are selected attributes computed from these gathers. Right: primary P-wave arrivals. This is the desired output for the neural net.*

5.3.1 Multiple Attenuation

According to the resulting data set a backpropagation neural network was designed with 8 input neurons for the attributes mentioned above, 4 hidden neurons, and 1 output neuron.

The output is one time sample of the near offset trace for each input vector. According to the desired output it is either a time sample of the near-offset trace of the primary CMP gather for multiple attenuation, or a time sample of the near-offset trace of the CMP gather containing the estimated multiples. These in turn have been obtained by subtracting the modelled primary wavefield from the full wavefield. Training for 300 epochs, and using the training CMP gather as the test data set, resulted in the trace shown in the center of Figure 5.5. For comparison, the input trace is shown to the right and the desired output trace to the left of the neural net output. Here, I performed multiple attenuation, since I used the primary section as the desired output for the training.

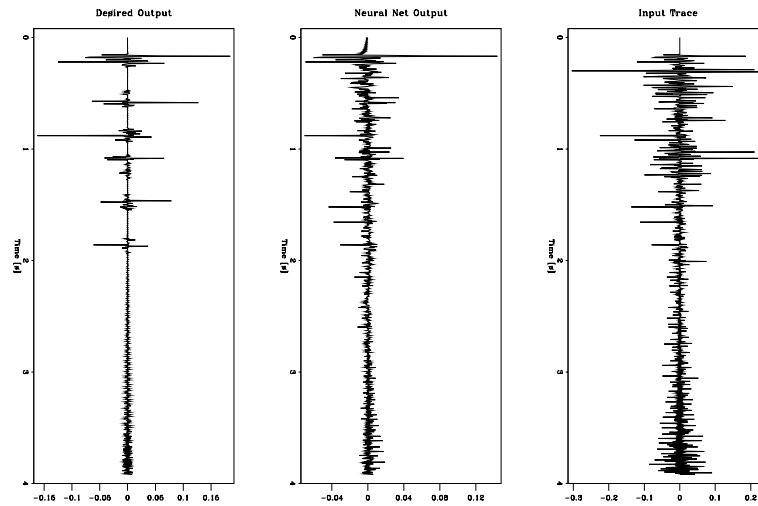


Figure 5.5: *Multiple attenuation. Left: first trace of the primary section (desired output). Center: neural net output (the trace with the attenuated multiples). Right: first trace of the full wavefield section (input).*

5.3.2 Multiple Prediction

In order to test if it is also possible to predict the multiples with the method described above, I trained the neural network to predict the free-surface multiples instead of the primary arrivals. I used the first trace of the CMP gather containing only free-surface multiples as the desired output. The result is shown in Figure 5.6. Nearly all phases and polarities of the predicted multiples are correctly determined by the neural network. The attenuation of free-surface multiples on this data set worked very well, the multiple energy has been reduced considerably and the primaries maintain their signal strength. Similarly, I tried to predict the internal multiples. Figure 5.7 depicts the result.

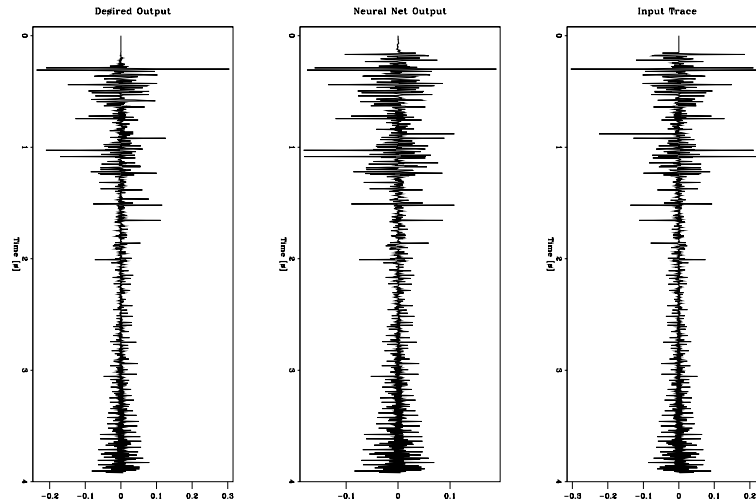


Figure 5.6: Multiple prediction: free-surface multiples. Left: First trace of the CMP gather containing only free-surface multiples (desired output). Center: neural net output (predicted free-surface multiples). Right: first trace of the full wavefield section (input).

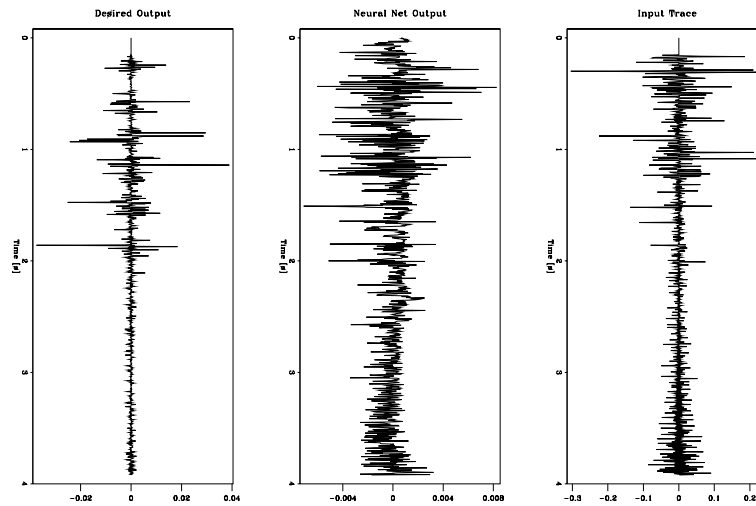


Figure 5.7: Multiple prediction: internal multiples. Left: first trace of the CMP gather containing only internal multiples and converted waves (desired output). Center: neural net output (predicted internal multiples). Right: first trace of the full wavefield section (input).

The prediction of multiples for the case of internal multiples failed. This suggests that the neural net, in common with most other multiple prediction algorithms, has problems with this type of multiples. However, it must be noted that the net was only trained with data from a single CMP gather, and that generally a neural net needs much more training data in order to perform better. On the other hand, the prediction of free-surface multiples is very satisfactory. The predicted multiples match the desired output at nearly every sample of the trace. Together with data that were preprocessed by another multiple prediction algorithm on a selected number of CMP gathers, this would be a fast and cheap method to predict the free-surface multiples on the remaining (major) part of the data set.

It should be mentioned that this method automatically produces a near-offset section containing either the primaries and attenuated multiples or the predicted multiples by using information from the full offset range and various attributes. In the next sections multiple prediction for 2-D data is shown.

5.4 Multiple Prediction in a Small 2-D Acoustic Data Set

This section demonstrates the application of the neural net multiple prediction method to a set of 2-D acoustic full wave synthetic CMP gathers, obtained from a finite difference (FD) modeling scheme. The neural network was trained with a set of synthetic CMP gathers containing the full wavefield as input. In order to predict the free-surface multiples, a set of synthetic CMP gathers, containing free-surface multiples only (Figure 5.8, right), was used as the desired output information. These gathers were obtained from FD modeling with an absorbing boundary at the surface and subsequent subtraction of the resulting data set from the data containing the full wavefield.

To obtain attributes for the neural net training, the velocity spectrum was computed, and the location and height of the peaks was determined for each zero-offset time. In addition I calculated the envelope of the first trace of each input CMP gather, and I used a window of length 5 samples from the first trace of the full wave CMP gathers as additional input for the neural net. Then I split the data set into a training set with 10 CMP gathers and a test set containing the remaining 10 CMP gathers. With this data set a backpropagation neural network was designed with 8 input neurons for the attributes, 4 hidden neurons, and 1 output neuron. Training for 300 epochs resulted in the traces shown in

Figure 5.8, left, where the general appearance of the multiples is predicted with only relatively few training gathers. For travel-times between 0.5 and 2.5 seconds the amplitudes of the predicted multiples are in part higher than the desired signal.

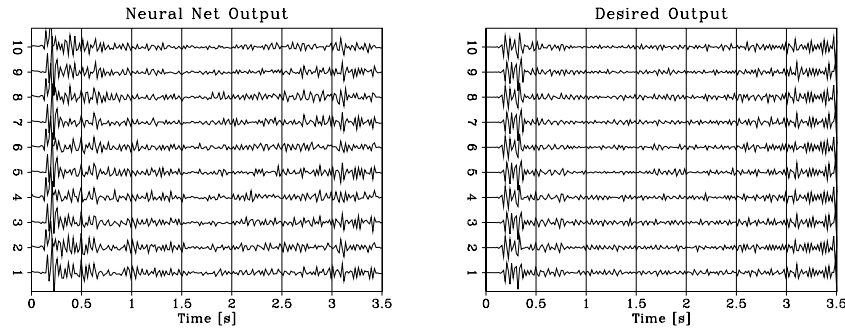


Figure 5.8: *Left: neural net predicted free-surface multiples of the near-offset trace of the 10 test CMP gathers. Right: free-surface multiples of the near-offset trace of the 10 test CMP gathers (the desired output).*

5.4.1 Target-oriented Multiple Prediction

The attribute based multiple attenuation and prediction method can be applied in a target-oriented fashion. Figure 5.9 shows the multiple prediction on traces from 10 CMP gathers for the first 500 ms. The neural network was trained with the traces of 10 CMP gathers (computed attributes in the first 500 ms) and then tested with 10 traces from different CMP gathers. The left panel of Figure 5.9 shows the free-surface multiples (desired output), and in the right panel the neural net predicted free-surface multiples are depicted. The relatively coarse appearance is caused by the extremely coarse sampling of the original synthetic data. The equivalent results are shown in Figure 5.10 for the target zone between 1500 and 3000 ms. The neural network output seems to extrapolate information horizontally. Multiples with relatively low degree of dip are recognized and predicted correctly. With the limited information available for the network it is able to extract just the most dominant features.

The trend of the multiples is predicted very nicely by the neural net, although the absolute amplitudes partly differ. Due to the few training patterns, the neural net tends to emphasize strong events and to smear them horizontally. This can be overcome by a more suitable training data choice.

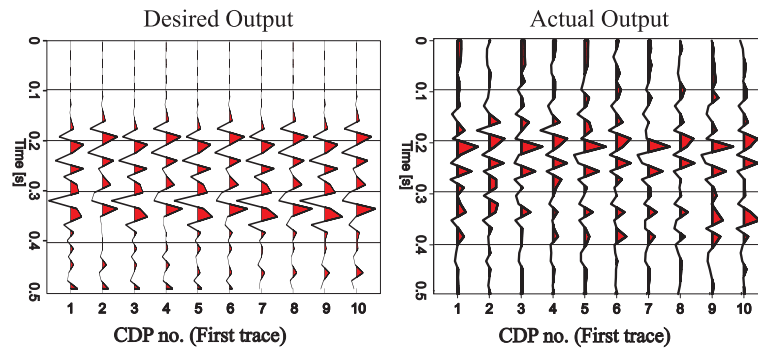


Figure 5.9: Left: free-surface multiples (desired output) within the first 500 ms, right: neural net predicted free-surface multiples.

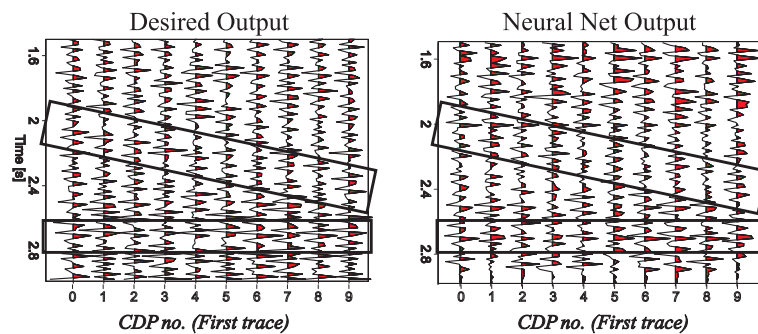


Figure 5.10: Left: free-surface multiples (desired output) within the target zone, right: neural net predicted free-surface multiples.

5.5 Multiple Prediction in a Full 2-D Acoustic Data Set

The multiple prediction method was also applied to a full 2-D synthetic data set, consisting of 1300 CMP gathers, that were computed using a finite difference algorithm. I selected a training set of 100 CMP gathers spread equally over the data set. This is less than 8 per cent of the full data set. The desired output is the first traces of all CMP gathers containing only free-surface multiples. They are shown in Figure 5.11. The result is not a stacked section, since the neural net output consists of one near-offset trace per CMP location.

The neural net output predicting the full data set, by training with the subset of 100 CMP gathers, is shown in Figure 5.12. Most of the structure across the data set has been reconstructed very reliably by the neural net, although again dominant horizontal streaks are visible, which relate to the tendency of the net to smear strong events horizontally.

However, shallow as well as deep multiples have been predicted, even the structure in the first strong multiple between CMP no. 800 to 1200 is reconstructed in the neural net output.

5.6 Conclusions

The attribute based multiple attenuation and prediction method using backpropagation neural networks is a supervised technique that preprocesses the seismic data by computing a selected number of attributes from various parameter domains, learns from a small subset of the data and tries to attenuate or predict multiples on the whole section. Depending on the supplied desired output we are either able to perform multiple attenuation or multiple prediction. In the first case the desired output consists of an estimate of the desired primary information obtained from modeling on the basis of well-log data. In the latter case the desired output is an estimate of the kind of multiple we want to predict. This information can be provided by modeling or by using various existing multiple prediction techniques.

The attenuation of multiples on the elastic synthetic data gave good results, but the prediction of internal multiples was a failure. On the other hand, prediction of free-surface multiples worked very well, and suggests that good results may well be achieved with other data sets.

The prediction of multiples on the full data set is based on an acoustic model. Here the free-surface multiples were predicted. The method worked very well along the entire profile.

An interesting possibility might be to use the output of a self-organizing map (SOM) as the input for a backpropagation network. This SOM, in turn, would have undergone unsupervised training with a set of attributes obtained from a seismic data set. This would use the SOM as a sort of pre-processor similar to the principal-component analysis (PCA) approach, that reduces the dimensionality of the input data and thus facilitates the task for the supervised learning scheme.

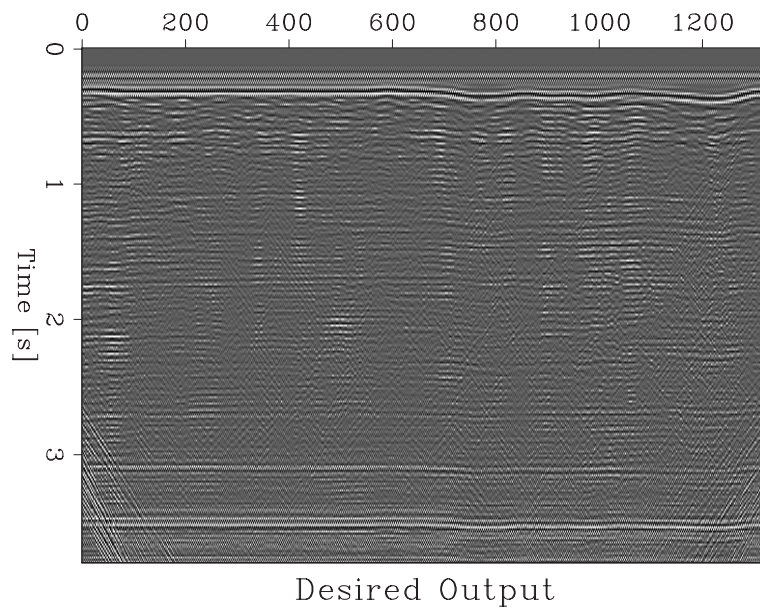


Figure 5.11: *The first traces of all CMP gathers containing only free-surface multiples (desired output).*

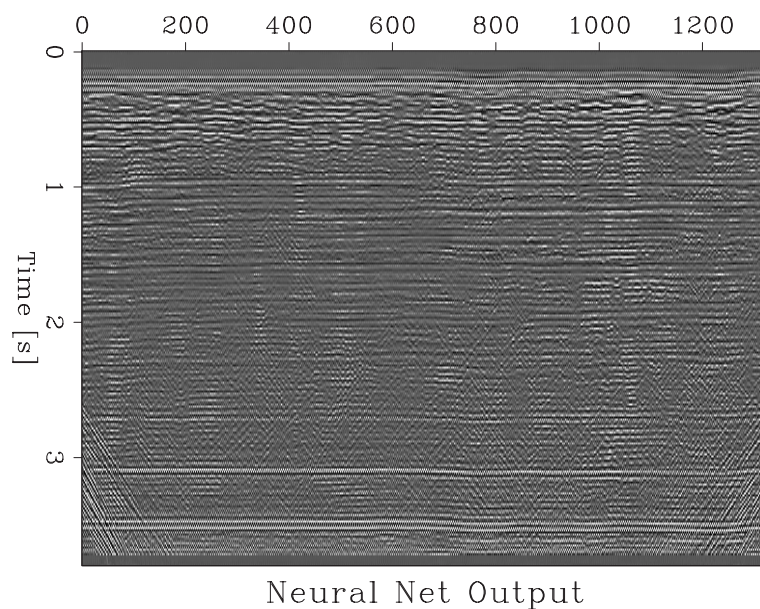


Figure 5.12: *The neural net predicted free-surface multiples.*

Chapter 6

Attribute based Multiple Identification with Self-Organizing Maps

In this chapter I introduce a new method that employs an unsupervised learning algorithm to extract information from the seismic data by means of clustering of primaries and multiples. I show that this method can be used to separate primary from multiple energy in a way that is easy to analyse for an interpreter or a processor. There is no prespecified desired output, but we merely present the network with the total seismic wavefield in the form of selected characteristic attributes. Labelling the formed clusters with a few picked primaries and multiples allows the classification of the large remaining portion of reflection events. In a subsequent step the identified multiples can be removed, e.g. with a filter in the parabolic $\tau - p$ domain, or other suitable methods.

6.1 The Method

Event Picking

Starting with a data set in CMP configuration and the corresponding velocity spectra (cp. section 3.1.), an automatic picking algorithm picks all peaks in the velocity spectra above a certain threshold. This provides zero-offset traveltimes and stacking velocities of all prominent events - primary and multiple reflections. Figure 6.1, left, shows all picked events for the synthetic data set (described in section 6.2) in its zero-offset location. The deep reflectors are not continuous since the semblance value was too small at certain CMP

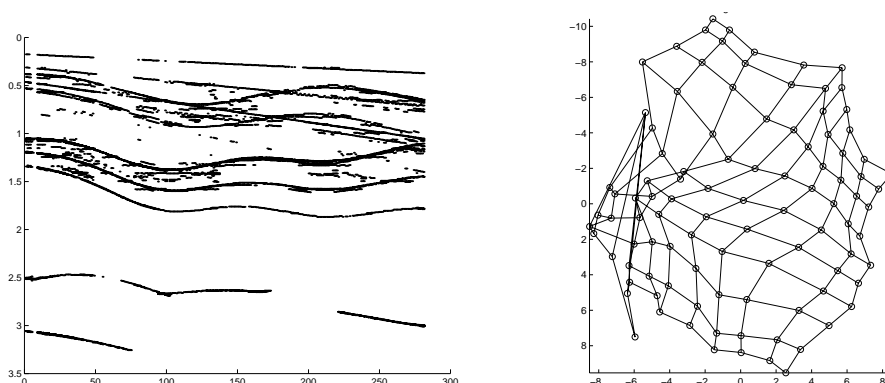


Figure 6.1: *Left: all picked events for the whole data set as a zero-offset section (primaries and multiples). Right: projection of the SOM weights after training had finished.*

locations to be picked by the automatic picking algorithm.

Computing Attributes

The picked zero-offset traveltimes and stacking velocities define hyperbolae for each picked event in the respective CMP gathers. Along these hyperbolae a number of selected attributes (see chapter 3) is computed for a set of offsets in a window around the hyperbola. Then we take the median of the attribute for each offset. This procedure provides robust attribute values. The wavefront attributes, emergence angle β_0 , and the radii of curvature R_{NIP} and R_N , are computed as described in section 3.3.

Correlation Analysis

In order to determine if some of the attributes contain redundant information, I performed a correlation analysis, which is formally shown below for a synthetic data set. Computing the linear correlation coefficient of one attribute for all offsets and events shows the behavior of the attribute over the offset. Thus, it can be judged if all attribute information from near to far offset is needed and which offsets show strong or weak correlations. Computing the correlation coefficient of all attributes for a specific offset and all events allows us to determine the contribution of the individual attributes to the information content of the data set to be used for SOM training. If two attributes are strongly correlated, we might disregard one of them and thus decrease the size of the data set. This leads to decreased computation times and avoids over-parameterization of the estimation process.

Training and Analysis

The next step is the design of the self-organizing map. First the dimension of the map has to be determined. This could be done by computing the fractal information dimension (Peitgens et al., 1992) that defines the dimension of the information content of the data. However, this procedure proved to be computationally too intensive. For most seismic data sets this dimension is assumed to be between 2 and 3, so that we generally start off with a two-dimensional map. If the network is not able to cluster the data, we can add one dimension and train the net again. The number of neurons should not be too small in order to represent all clusters that are potentially possible. An important parameter is the initial size of the neighborhood where the neighboring weights are updated along with the winning neuron's weight. If it is too small the map will not be ordered globally but splits up into a mosaic of very small clusters (Kohonen, 1997a). My experience shows that a good starting value is about half the map size, with a linear decrease of the neighborhood size with training time. The learning rate is often chosen to be 1 and decreases linearly with time.

A criterion for stopping the training is given by the visualization of a projection of the multi-dimensional network weights on a 2-D display. On the right hand side of Figure 6.1 the SOM weights after training are displayed for the synthetic example. They show mostly satisfactory unraveling. The optimum would be an approximately grid-like appearance in the form of the (here) rectangular map.

Another means of analysis is the visualization of the component planes where the values of all weights connected with one input component are shown. Figure 6.2 a) and b) shows the two component planes of the attributes "zero-offset time" and "stacking velocity". The values of the weights connecting every neuron on the feature map with an input vector component are depicted. Here, the attributes "zero offset time" and "stacking velocity" are taken as input vector components.

Figure 6.2 c) depicts a histogram showing the frequency of classified events to each neuron (a large square means many events have been classified to this neuron, small ones mean only a few). The U-Matrix showing the mutual distances of the weights of the feature map (cp section 2.2.2) is displayed in d). It serves as the display of the main

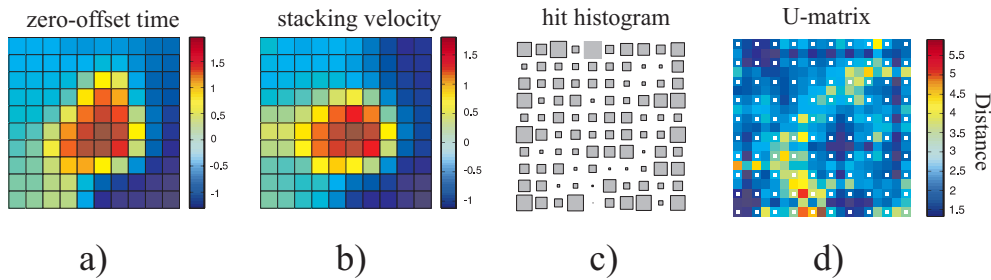


Figure 6.2: *Component planes: a) values of the weights connecting every neuron on the feature map with the input neuron with the attribute “zero offset time”. b) values of the weights connecting every neuron on the feature map with the input neuron with the attribute “stacking velocity”. c) histogram displaying the frequency of classifications to each neuron. d) U-Matrix showing the mutual distances of the weights of the feature map.*

individual clusters with “ravines” of large distances (displayed in yellow/red colors) separating the cluster “hills” (blue color shades). On the basis of the U-matrix the neurons of the trained feature map are then assigned to different clusters. Every input vector that is mapped to a neuron belonging to a certain cluster, is tagged with the color of this cluster.

6.2 Application to Synthetic Data

As an application for the method described above, I designed a synthetic subsurface model that is 10 km wide and 5 km deep. The seven layers each have a constant P- and S-velocity, as well as a constant density. The grid interval is 10 m. The P-velocity of the model is shown in Figure 6.3, left.

Using this model, a set of shot gathers was computed via ray tracing. For the ray tracing algorithm every ray code we want to have in the seismogram later has to be specified. Apart from the primaries, I chose water multiples up to the 4th order, three types of peg-leg multiples, and four types of internal multiples. The dominant frequency of the wavelet was 40 Hz, the sample rate 4 ms, and the total recording time 4 seconds. There was a shot every 25 m and a receiver spacing of 12.5 m. This resulted in a total of 400 shot gathers, which were resorted into 400 CDP gathers. Finally 281 CDP gathers were selected that had full coverage from 0 to 3 km offset (60 traces per CDP gather). A typical CDP gather is shown in Figure 6.3, right. The stacked section, using an automatic CMP stack that sums along that event that has the highest semblance, is shown in Figure 6.4.

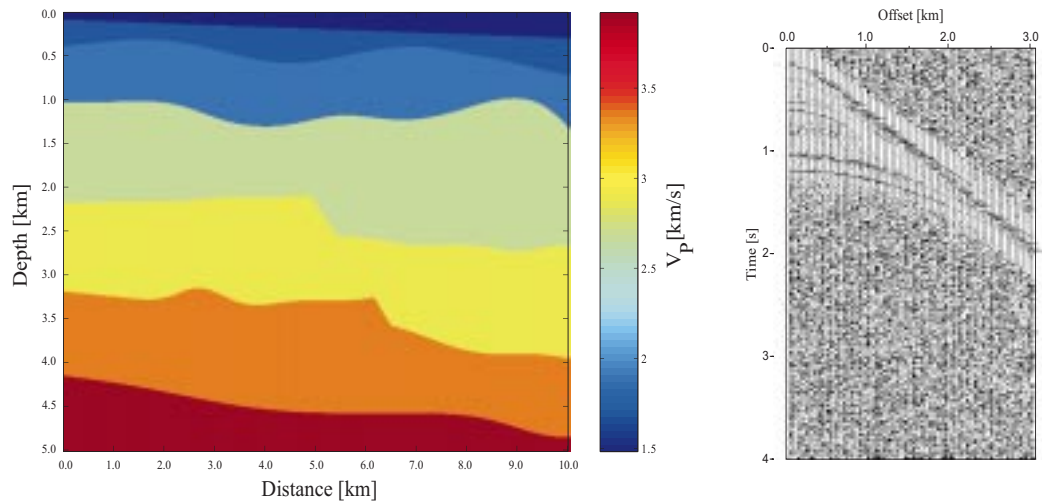


Figure 6.3: Left: synthetic pressure wave velocity model. Right: CMP gather plus noise at location 1.5 km.

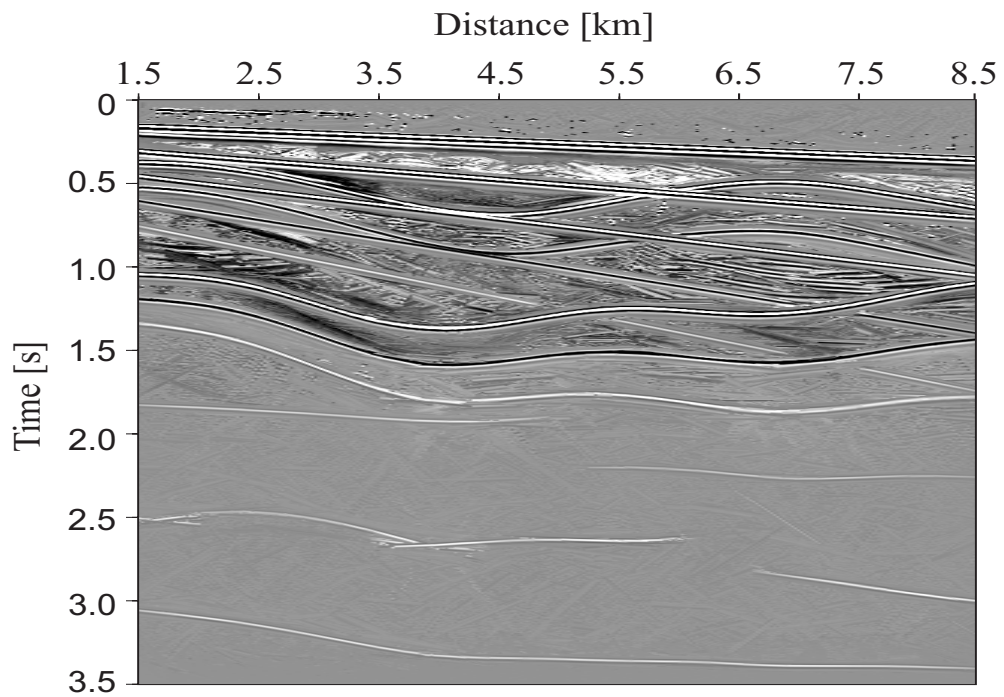


Figure 6.4: Automatically stacked section (unmigrated).

With this technique the multiples are stacked as well as the primaries, and thus can be clearly recognized.

From the CDP gathers a number of attributes was computed using the attribute software package of the Seismic Research Corporation (Taner, 1998):

Instantaneous Attributes

real part of complex trace
 imaginary part of complex trace
 trace envelope
 time derivative of envelope
 second derivative of envelope
 instantaneous phase
 instantaneous frequency
 envelope weighted inst. frequency
 acceleration of phase
 dominant frequency
 bandwidth
 instantaneous Q-factor
 normalized amplitude
 envelope amplitude modulated phase
 relative acoustic impedance

Wavelet Attributes

trace envelope
 time derivative of envelope
 second derivative of envelope
 instantaneous phase
 instantaneous frequency
 envelope weighted inst. frequency
 acceleration of phase
 dominant frequency
 bandwidth
 instantaneous Q-factor

as well as the three **wavefront parameters**: emergence angle β_0 , and the radii R_{NIP} , and R_N .

Some selected attributes are shown in the following Figure 6.5 for the CDP gather of Figure 6.3. From these CDP gathers no obvious differences between primaries and multiples are directly visible to the human eye.

Next, the velocity spectra of all 281 CDP gathers were computed. From these velocity spectra the algorithm picked automatically all values that were higher than a specified clip value. Before that I balanced the amplitude of the shallow peaks against the deeper peaks using AGC (automatic gain control). These picks constitute the two-way traveltime and the velocity of each event, i.e. the zero-offset time and the curvature of each hyperbolic event in the CDP gather. This information was used to determine the hyperbolic events

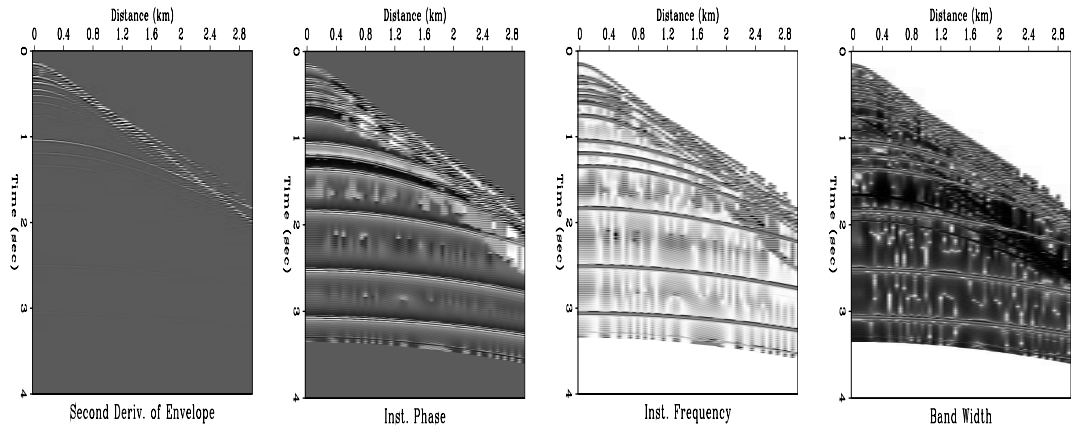


Figure 6.5: *From left to right: second derivative of envelope, instantaneous phase, instantaneous frequency and bandwidth*

in the computed attribute gathers. All attribute values for all offsets of each event were written to a data cube, whose x-dimension is the offset, whose y-dimension is the attribute, and whose z-dimension is the event number. For reasons of data reduction the data were resampled along the offset axis to every 10th sample.

Using these data I created a pattern file for the input to a self-organizing map. This pattern file consisted of 7034 events from 281 CDP gathers (on average 25 events per CDP gather). Each event represents one input vector to the SOM and has the following components:

- zero-offset time
- stacking velocity
- attribute no. 1 (for every 10th offset)
- attribute no. 2 (for every 10th offset)
- ⋮
- attribute no. N (for every 10th offset)

6.2.1 Correlation Analysis

Before training the SOM I performed a correlation analysis in order to find out, which attributes show a mutual correlation and thus might represent redundant information.

The linear correlation coefficients C_{ij} between each combination of two wavelet attributes y_i and y_j was computed for all $N=7034$ events and summed over all L offsets:

$$C_{ij} = \frac{1}{L} \sum_{l=1}^L \frac{1}{N} \sum_{k=1}^N (y_{ikl} - \bar{y}_{il})(y_{jkl} - \bar{y}_{jl}) \quad (6.1)$$

$$\mathcal{C} = \hat{C}_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}} \quad (6.2)$$

for $i, j = 1, \dots, M$, with $\bar{y} = 1/N \sum_{k=1}^N y_k$, and M the number of attributes. The resulting matrix \mathcal{C} containing the normalized correlation coefficients \hat{C}_{ij} in a color-coded form is shown in Figure 6.6. Every matrix element represents the crosscorrelation coefficient of two respective individual attributes. It shows how well a pair of attributes correlates. On the diagonal the correlation of an attribute with itself shows a unique linear dependency. In the off-diagonal elements we can see how attributes correlate. Red values show high positive correlation, whereas blue values indicate strong negative correlation. An analysis of the behavior of the attributes with offset, including the computation of scatter plots, shows that the correlation patterns change dramatically with offset. When all offsets are considered together, then the far offsets dominate the structure.

If we want to interpret this correlation analysis with respect to the composition of the pattern file for SOM training, two (mutually exclusive) strategies suggest themselves:

1. We eliminate those attributes that correlate strongly with another attribute.
2. We use all attributes no matter what correlation they may have with other attributes.

Strategy 1 aims at the reduction of the data set, and thus has a better chance to classify the patterns in a space with reduced dimensionality. Strategy 2 is based on the argument that every attribute contributes some information, however small it might be, and that this piece of information should be kept in order to ease the classification task for the SOM. In consideration of the fact that we do not have thousands of attributes, but only between 10 and 20, I favour strategy 2. Here I only show the correlation analysis for the wavelet attributes; the results for the instantaneous attributes look similar.

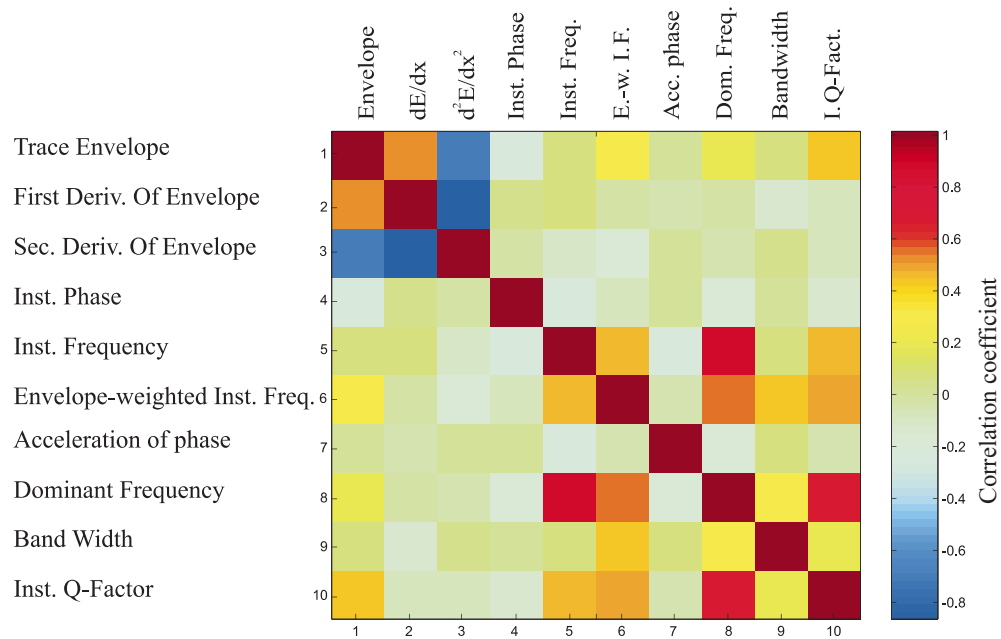


Figure 6.6: Correlation coefficients for all ten wavelet attributes. This matrix is symmetric and one square signifies the crosscorrelation of two attributes summed over all offsets.

In the correlation coefficient matrix above we can see some strong correlations between certain attributes:

- positive correlation:
- trace envelope & time derivative of the envelope
 - trace envelope & instantaneous Q-factor
 - instantaneous frequency & envelope weighted inst. frequency
 - instantaneous frequency & dominant frequency
 - instantaneous frequency & instantaneous Q-factor
 - envelope weighted inst. frequency & dominant frequency
 - envelope weighted inst. frequency & bandwidth
 - envelope weighted inst. frequency & instantaneous Q-factor
 - dominant frequency & instantaneous Q-factor
- negative correlation:
- trace envelope & second derivative of the envelope
 - time derivative of the envelope & second derivative of the envelope

The positive and negative correlations of the envelope and the first and the second derivative, respectively, were expected and can be easily understood with the help of a simple example using the sine-function: $\frac{d \sin(x)}{dx} = \cos(x)$, $\frac{d \cos(x)}{dx} = \frac{d^2 \sin(x)}{dx^2} = -\sin(x)$, i.e. the first derivative (cosine-function) has a positive correlation to the original signal, whereas the second derivative shows a clear negative correlation. The correlations of the instantaneous frequency with its smoothed versions, the dominant frequency and the envelope-weighted frequency are obvious. From the definitions of the attributes bandwidth, dominant frequency and Q-factor, we can as well expect some correlation, since they are all a function of frequency.

6.2.2 Results for Instantaneous Attributes

In a first investigation I used a number of instantaneous attributes: the imaginary part of the complex trace (the real trace with 90 degrees phase shift), the trace envelope, the second derivative of the trace envelope, the instantaneous phase, the instantaneous frequency, bandwidth, and the relative acoustic impedance. For the final labelling of the clusters that formed on the map, we used a few interpreted events, that are taken preferably from shallow depths (i.e. the water bottom) since they are easier to identify.

Figure 6.7 shows the results for the first two CDP gathers. In the left panel of each figure the picks in the velocity spectrum are shown. Due to the smearing in the velocity spectrum the automatic picking algorithm occasionally picked several events for the same ZO-time, but with different velocities. This might explain the problems that later occur with the SOM results. Since not all of these picks are “real” events, i.e. which follow a hyperbola in the corresponding CDP gather, these “fake” events do not contain reasonable values for the attributes and thus might degrade the classification.

In the right panel of each figure the classification result is displayed. Each event picked in the left panel is classified and associated with a colored line. The results for these first two CDP gathers are consistent, and do not show random classification of the events. The classification result for the instantaneous attributes for the entire model is shown in Figure 6.8 as a zero-offset section.

This result is not overwhelming, since it is hard to see clear classification trends. For this reason I disregard the class of instantaneous attributes for further classification and use the wavelet attributes and the wavefront parameters (see section 3.2 and 3.3).

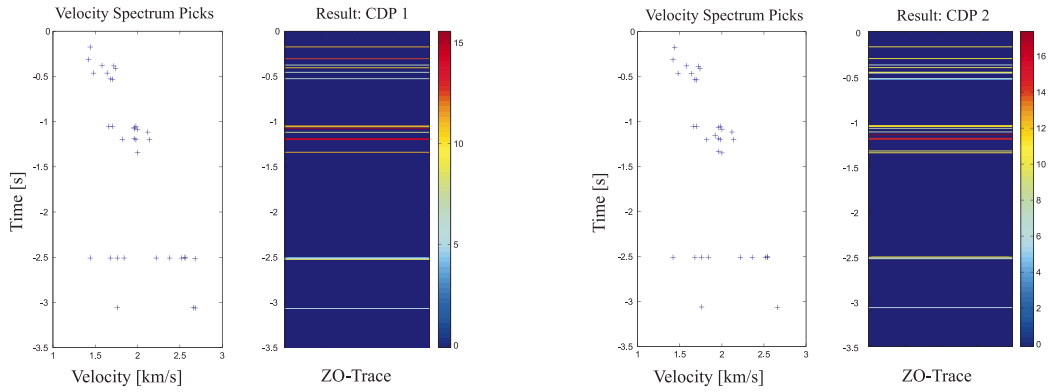


Figure 6.7: Results for the first two consecutive CDP gathers. Left Panels: picks from the velocity spectrum. Right Panels: classified events. Different colors mean different clusters.

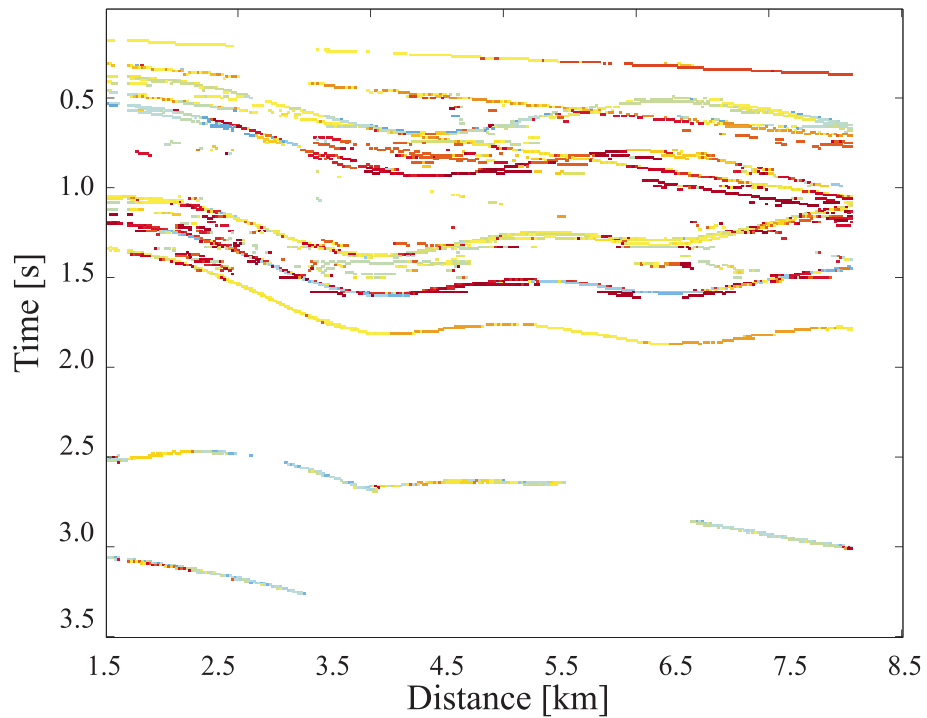


Figure 6.8: Zero-offset section of classification using instantaneous attributes. The colors represent different classes of events. The individual events do not show coherent classification to a particular class. For this reason the instantaneous attributes are disregarded from further classification analyses.

6.2.3 Results for Wavelet Attributes

The result for training a SOM with all ten wavelet attributes plus zero-offset time and stacking velocity is shown in Figure 6.9. We observe that the SOM classified groups of primaries along with their corresponding multiples. This is shown by color. For example, the first event is the sea bottom dipping from left to right. It is displayed in light blue, as well as its multiples below it. The second event is the primary from the curved second interface. It is displayed in yellow, and its multiple (a peg-leg multiple) is also yellow/orange. The third primary from the third curved interface shows up as red and dark blue, as do its three multiples below it.

Most events are now consistently displayed in a single color. However, the primary for the third interface has some blue overlays. This might be due to what I mentioned above, namely that several events were picked at one zero-offset time because of smearing in the velocity spectrum.

In conclusion, use of wavelet attributes plus zero-offset time and stacking velocity does produce clustering of multiples along with their multiple-generating primary. The assumption that zero-offset time might be a strong and dominant attribute leads to the idea of rerunning the SOM training without this particular attribute. The result is shown in Figure 6.10. Now the SOM can in fact distinguish between a given primary and its corresponding multiples, which is a very desirable classification result if it can be shown to hold for real data. Thus, the sea bottom is shown in orange and its multiples in yellow/light blue. The second primary is yellow, whereas its multiple is light blue. The third primary is also yellow and its multiples are red. For practical application these two approaches can be combined: First the SOM is trained including the zero-offset time as an attribute in order to extract the multiple generating set (primary plus corresponding multiples). Then the SOM is trained without the zero-offset time attribute, in order to separate the primary from the multiples within each set.

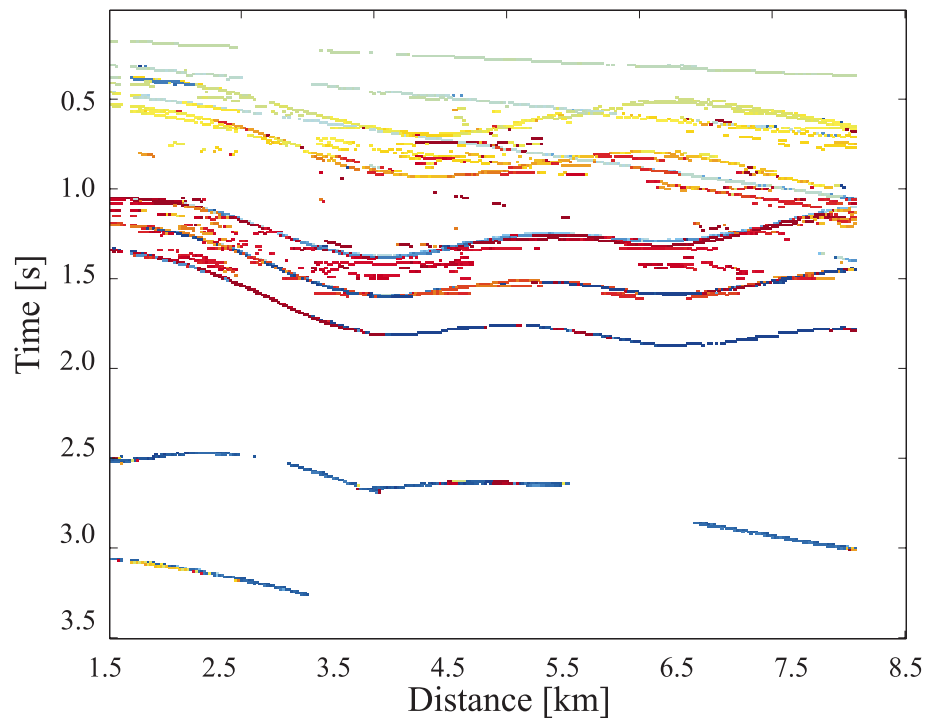


Figure 6.9: Zero-offset section of classification using wavelet attributes. The colors represent different classes of signals.

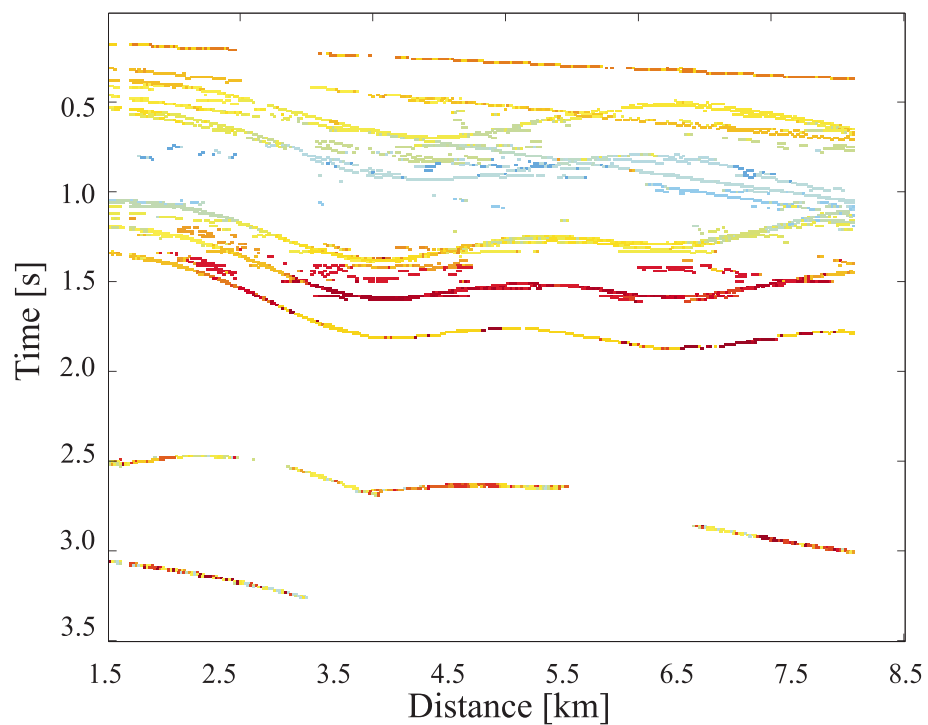


Figure 6.10: Zero-offset section of classification using wavelet attributes (the same as in Figure 6.9), but without zero-offset time as an input attribute. The colors represent different classes of signals.

6.2.4 Component Plane Analysis for Wavelet Attributes

Component planes display one component of each weight vector after training. Each neuron of the map is connected with every component of the input vector. The values associated with each connection, compose the weight vector. Thus, the weight vector has as many components as the input vector. The display of one component of the weight vector, i.e. the connection of every neuron with one component of the input vector, provides the possibility to analyze the clustering behavior according to this component. In the synthetic example here, the input vector has 65 components: zero-offset time, stacking velocity, 10 attributes over 6 different offsets, and three wavefront parameters. Accordingly, each weight vector also has 65 components.

We can now investigate, how the algorithm separates the input data with respect to every individual attribute. Each neuron is labelled with the corresponding type of input vector: “P” for a primary event, and “M” for a multiple. The red areas are separated from the blue areas on the component planes by yellow and green “trenches”. The labelling of the neurons is performed by using the majority-principle: a neuron receives the label of that type of input vector with the greater hit count.

Figure 6.11 shows three component planes for the three components “envelope near offset” (left), “envelope mid-offset” (center), and “envelope far offset” (right). The near offset shows a trend for separating primaries from multiples, whereas the mid and far offsets separate different areas with multiples from the rest. Similarly, Figure 6.12 displays three of the six component planes for the attribute “bandwidth”. These plots and the analysis of the other component planes showing an attribute over offset lead to the conclusion that there is no redundant information in the different offsets. Different offsets cluster different types of multiples and/or primaries. This can be used to further analyze the separation of the multiple wave field into the wavefields of internal and free-surface multiples.

Figure 6.13 shows three component planes with three different attributes for one single offset, here the near offset. As can be expected, the coloring of the component plane for the second derivative of the envelope is inverse to the colors of the envelope and first derivative of the envelope planes. This is due to the negative correlation of these attributes.

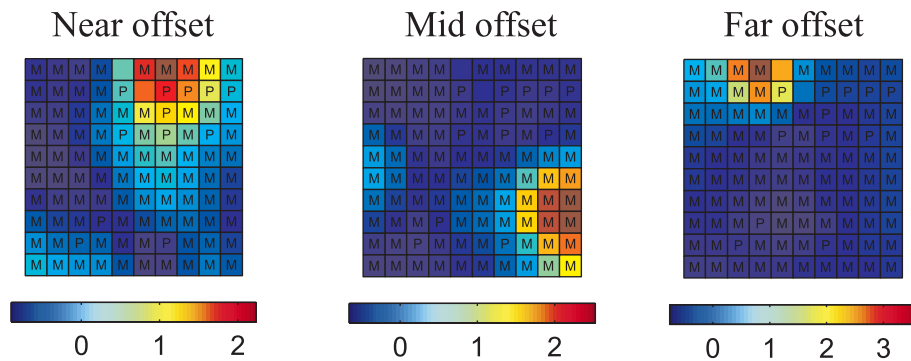


Figure 6.11: Component planes for the attribute "envelope" for near, mid-, and far offset. Also displayed are the labels of that class of input vectors that has been mapped to the corresponding neuron: *P* signifies that the majority of all events mapped to this neuron is a primary, *M* stands for multiple.

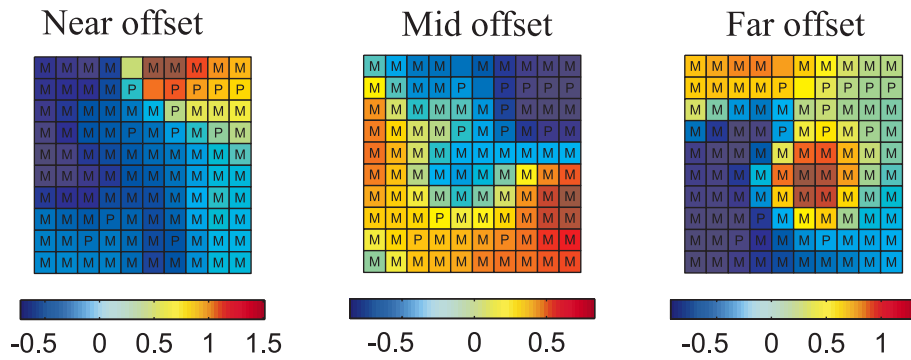


Figure 6.12: Component planes for the attribute "bandwidth" for near, mid-, and far offset.

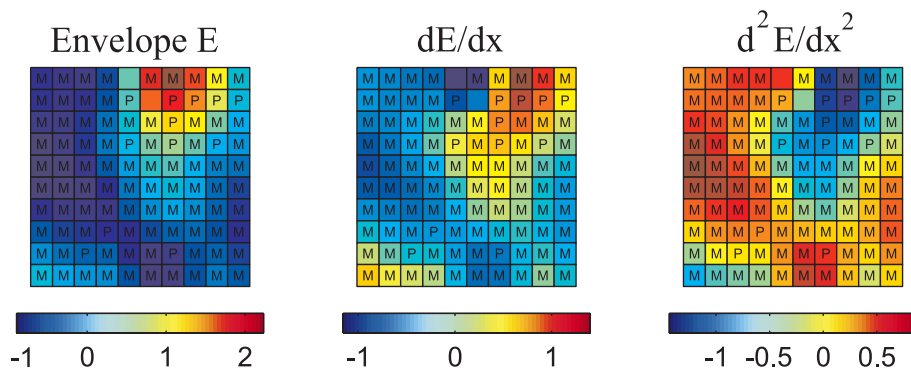


Figure 6.13: Component planes for the near offset for three attributes: envelope, first and second derivative of the envelope.

6.2.5 Results for Wavelet and Wavefront Attributes

In addition to the wavelet attributes (plus ZO-time and stacking velocity) used in the last section, here the wavefront attributes (section 3.3) are used for multiple identification and classification. This combination of attributes proved to optimally exploit the differentiation criteria contained in the data, as is shown below.

The separation of primaries and multiples is demonstrated for two different synthetic 2-D data sets: 1) the data set based on the model with the seven layers, already used above, and 2) a realistic data set based on a model that was designed to match the field data. The first data set was computed using ray tracing, the second with finite differences. Whereas for data set no. 1 we know every individual event, since we provided the ray codes, for data set no. 2 only a number of picked primary events from an interpreted stacked section was available.

The result of the self-organizing map multiple classification for the first data set is shown in Figure 6.14. On the left hand side at the top, the U-matrix after training is displayed. The clusters inferred from the hills and ravines of the U-matrix are shown at top right. Every cluster is labelled with the corresponding events that have been classified to that respective area on the map. The entire ZO-section is shown at the bottom. All events have the same color as the cluster it belongs to. Obviously, the primary events (water bottom WB, primaries of the second reflector P2, the third, fourth and fifth reflector P3, P4 and P5) cluster in an area bottom left/left to top right on the map, whereas the multiples (water bottom multiples WBM1 and WBM2, internal and pegleg multiples M3 and M4) are located at top left and bottom right. Primaries are colored in blue and yellow colors, the multiples in various shades of red. Except P2, which is heavily corrupted by water reverberations, all events show a continuous color and the primaries can be clearly separated from the multiples. The two separate wavefields are shown in Figures 6.15 (the primary events only) and 6.16 (the multiples only). The corresponding traveltime hyperbolae are displayed in the CMP gather at location 1.5 km in Figure 6.17. This demonstrates that, assuming hyperbolic traveltime curves, the method allows identifying primaries and multiples in the pre-stack data set. Using an adequate filtering technique (e.g. in the parabolic $\tau - p$ domain), the identified multiples can be filtered out.

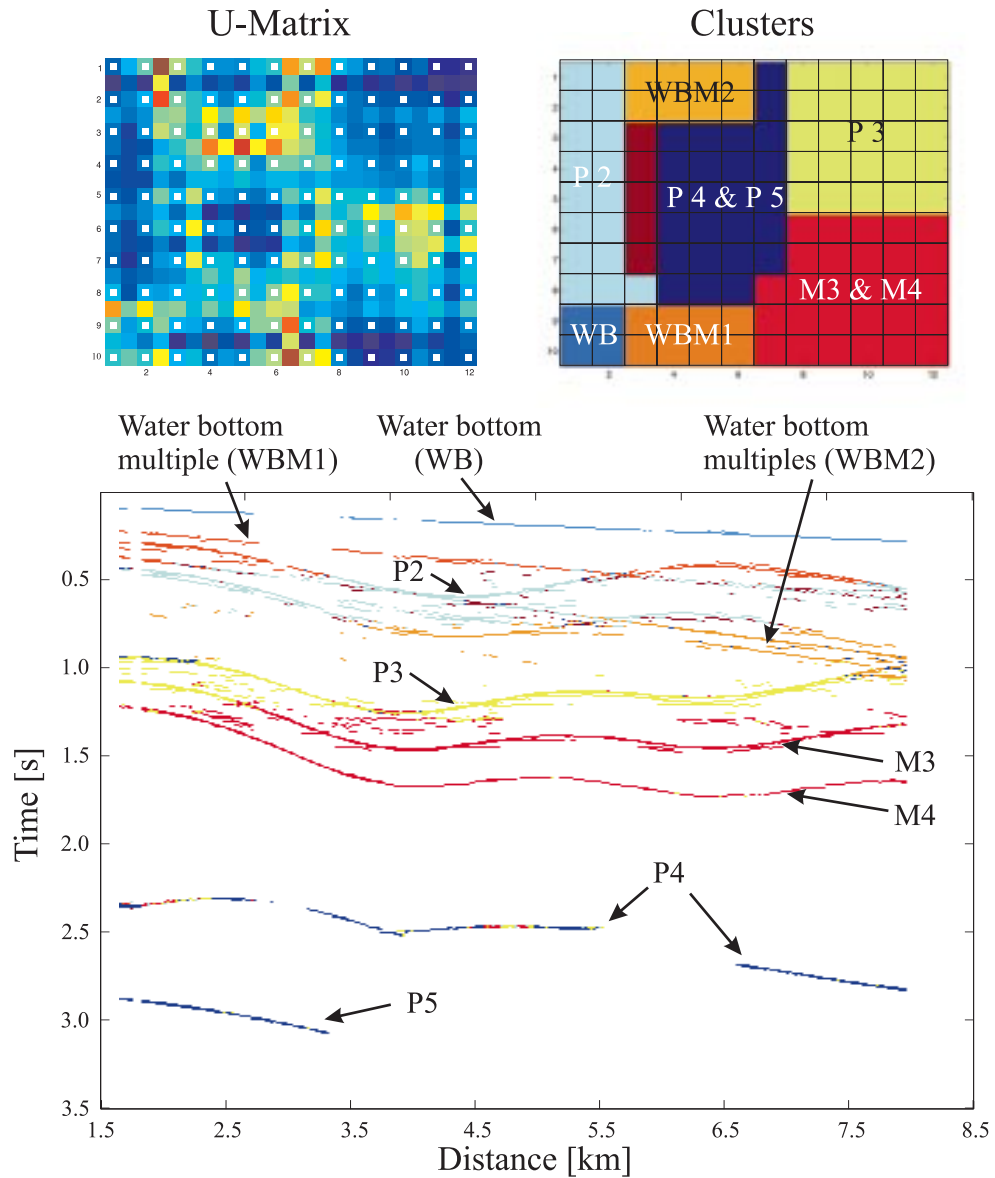


Figure 6.14: First data set: Cluster analysis for using all wavelet and wavefront attributes, including zero-offset time and stacking velocity. The clusters on the map (top right) have been determined on the basis of the U-matrix (top left). The events in the ZO-section below have the same colors as the clusters they have been classified to. Primaries mainly cluster from bottom/left to top/right on the map, whereas multiple clusters are located at bottom/right and top/left.

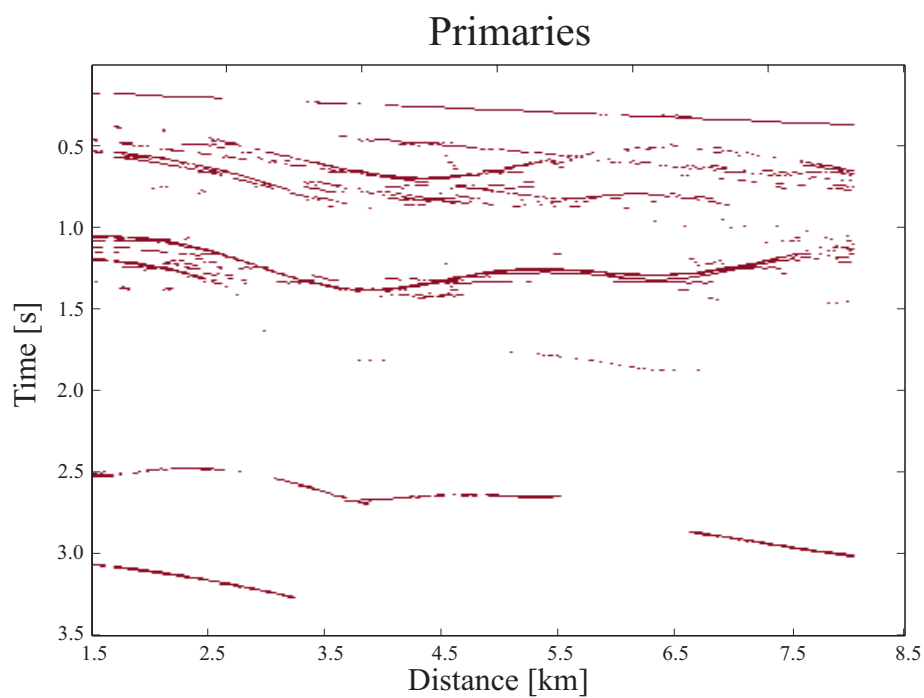


Figure 6.15: *The primary wavefield after separating out the multiple wavefield.*

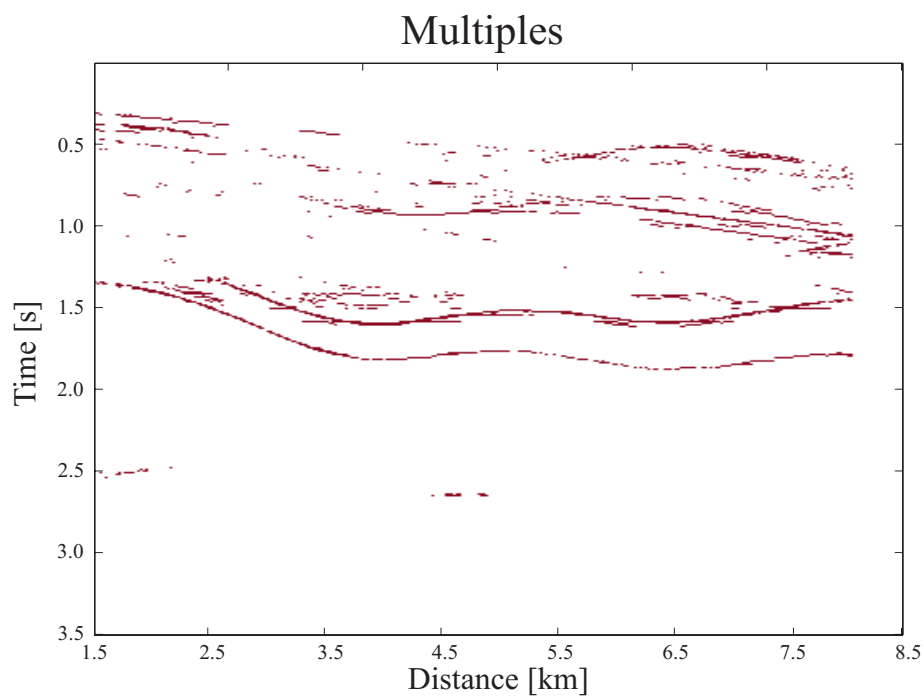


Figure 6.16: *The multiple wavefield after separating out the primary wavefield.*

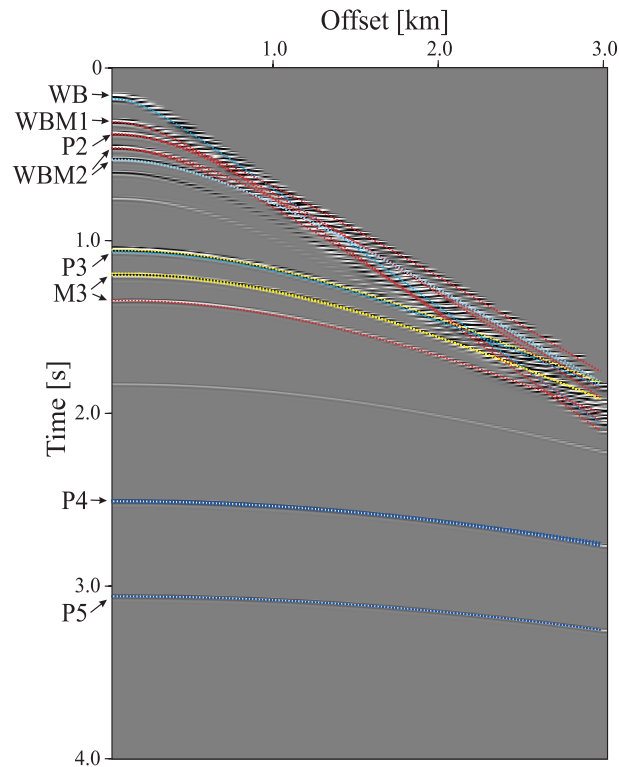


Figure 6.17: *CMP gather at location 1.5 km with all classified events.*

The second data set was processed in the same manner, and the result is displayed in Figure 6.18. Again, at the top the U-matrix and the resulting clusters show a very good separation of the two wavefields. Primaries (WB, P1, P2, P3, and P4) cluster at the bottom and in the center, with the multiple clusters (WBM1, WBM2, M2, M3 and M4) surrounding them. In the colored ZO section below, the individual events are labelled and displayed in the same color as the cluster they have been classified to. The deeper multiples are all comprised in the class M4, which could, according to the cluster map, be subdivided and further analysed to investigate the origin of these multiples. The wavefields have been separated into the primaries only (Figure 6.19) and the multiples only (Figure 6.21). For comparison, the primaries picked from an interpreted section are shown in Figure 6.20. Note that this information was not used for the clustering process. Only one pick from each of the horizons served to label the clusters on the map. Figure 6.23 depicts the automatically stacked section.

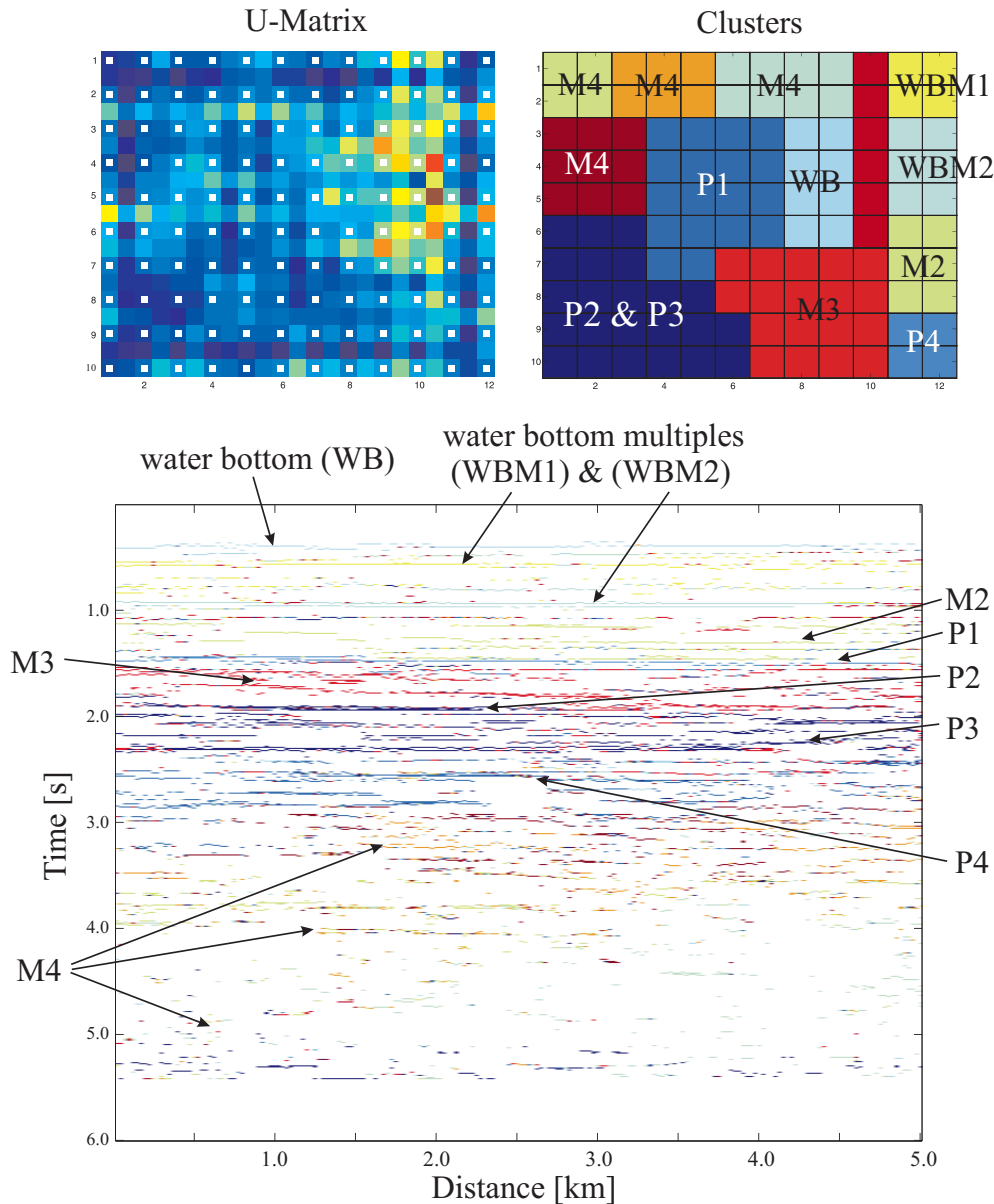


Figure 6.18: Second data set: Cluster analysis for using all wavelet and wavefront attributes, including zero-offset time and stacking velocity. The clusters on the map (top right) have been determined on the basis of the U-matrix (top left). The events in the ZO-section below have the same colors as the clusters they have been classified to. Primaries mainly cluster at the bottom/left and the center of the map (with the exception of P4), whereas multiple clusters are located at the top, on the right side and at the bottom right of the map.

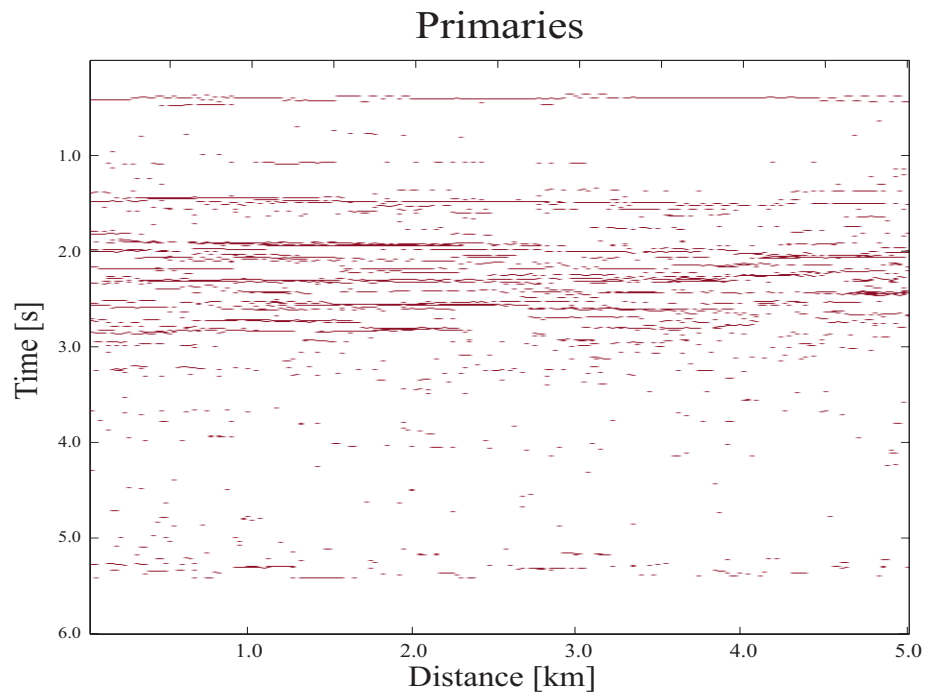


Figure 6.19: *The primary wavefield after separating out the multiple wavefield.*

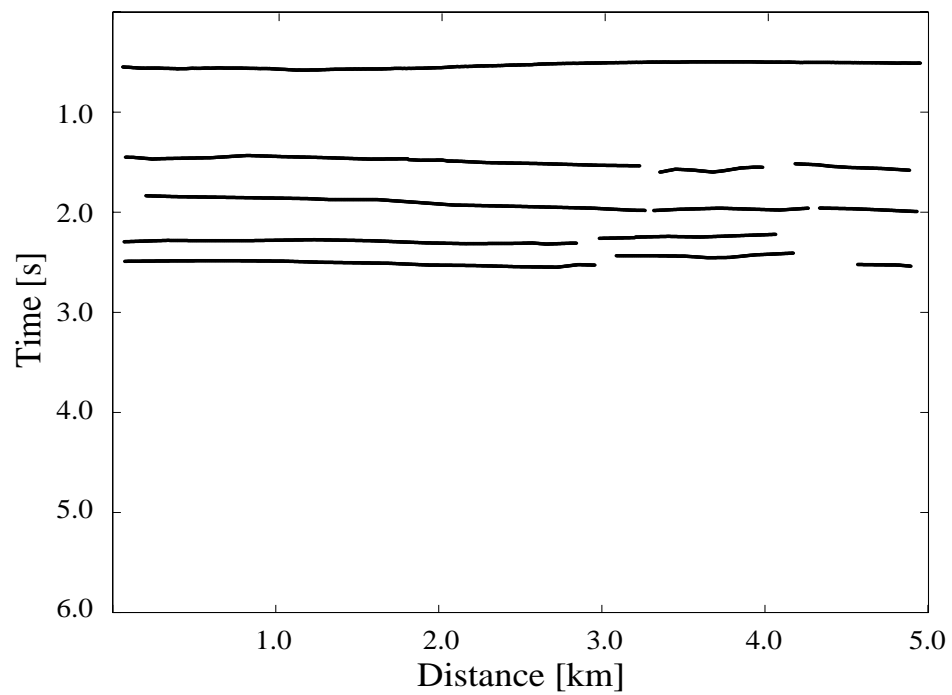


Figure 6.20: *The primary events picked by an interpreter.*

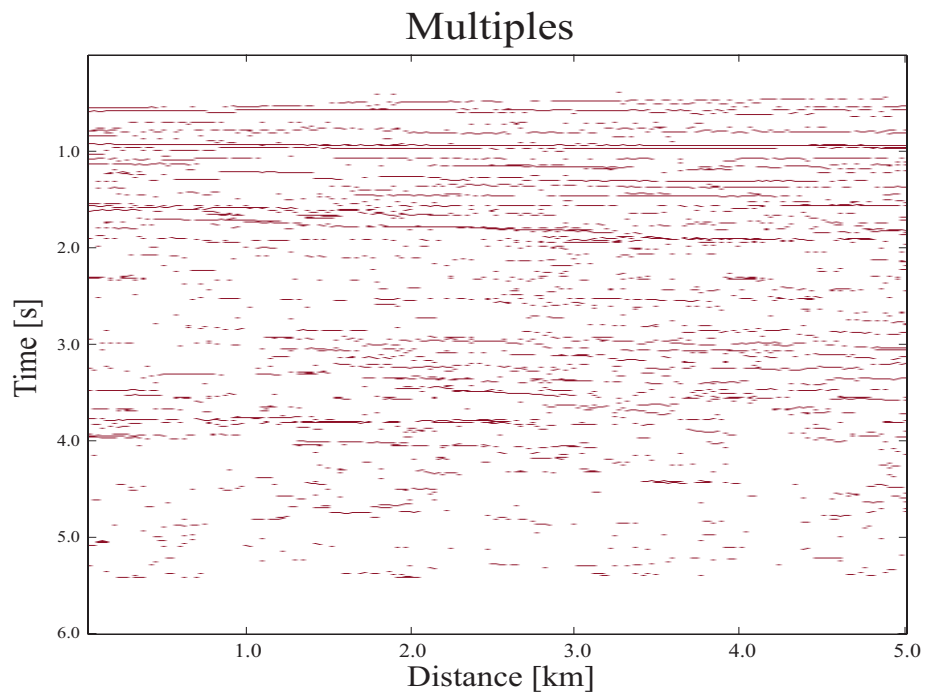


Figure 6.21: *The multiple wavefield after separating out the primary wavefield.*

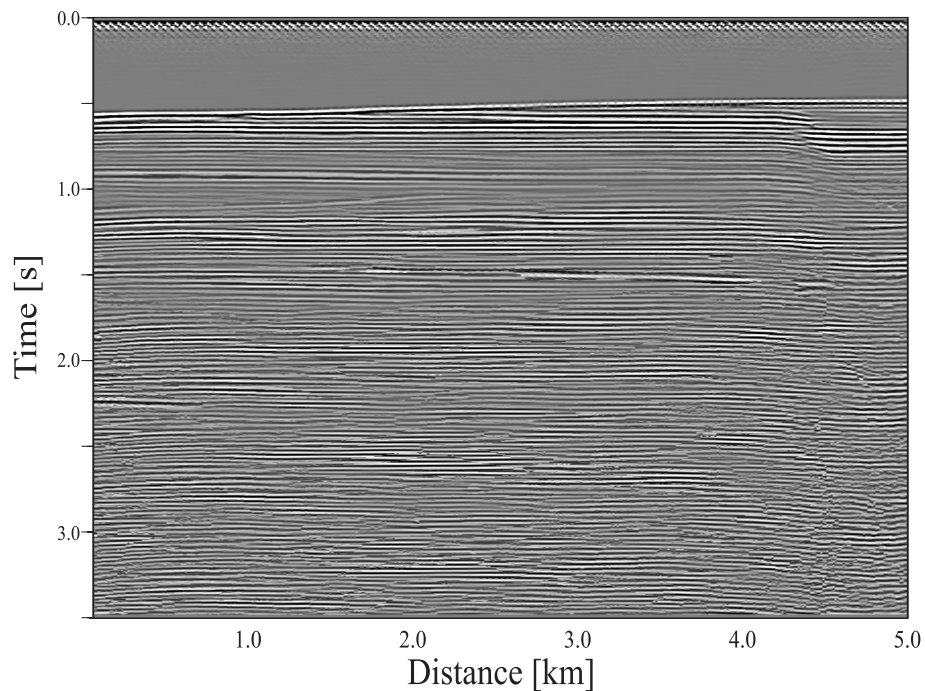


Figure 6.22: *Automatically stacked section of the second data set. (Note: only the upper 3.5 seconds are depicted.)*

6.3 Application to Real Data

In order to test the performance of the technique in the real world, the same procedure as described above was applied to a real data set. The automatically stacked section is shown in Figure 6.23. For this data set, about 10,000 events have been picked in 200 CMP gathers. All wavelet attributes and wavefront parameters were computed. Then a self-organizing map of size 10×12 was trained. The result is shown in Figure 6.24. The individual events are not as coherent over distance as in the synthetic data set of the previous section. As a result also the classification of the events by the SOM is not as coherent. However the water bottom and the primary at 1.45 seconds show up in blue color, and also the primaries at 2.3 and 2.5 seconds can be recognized as blue bands. The multiples generally show red color shades. A separation of the primary and the multiple wavefield can be seen in Figures 6.25 and 6.26, respectively.

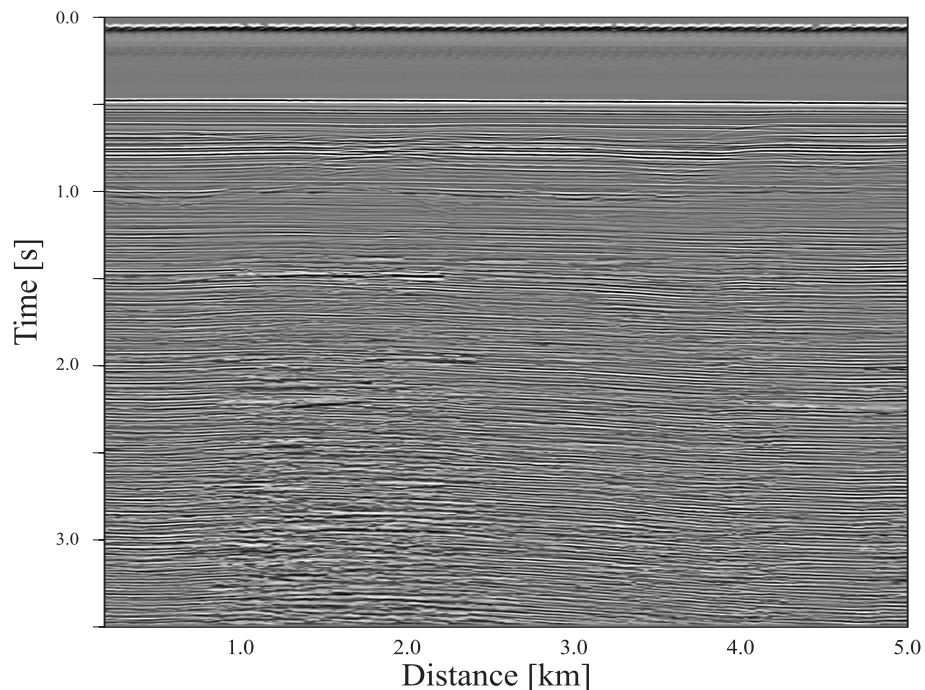


Figure 6.23: Automatically stacked section of the real data set. (Note: only the upper 3.5 seconds are depicted.)

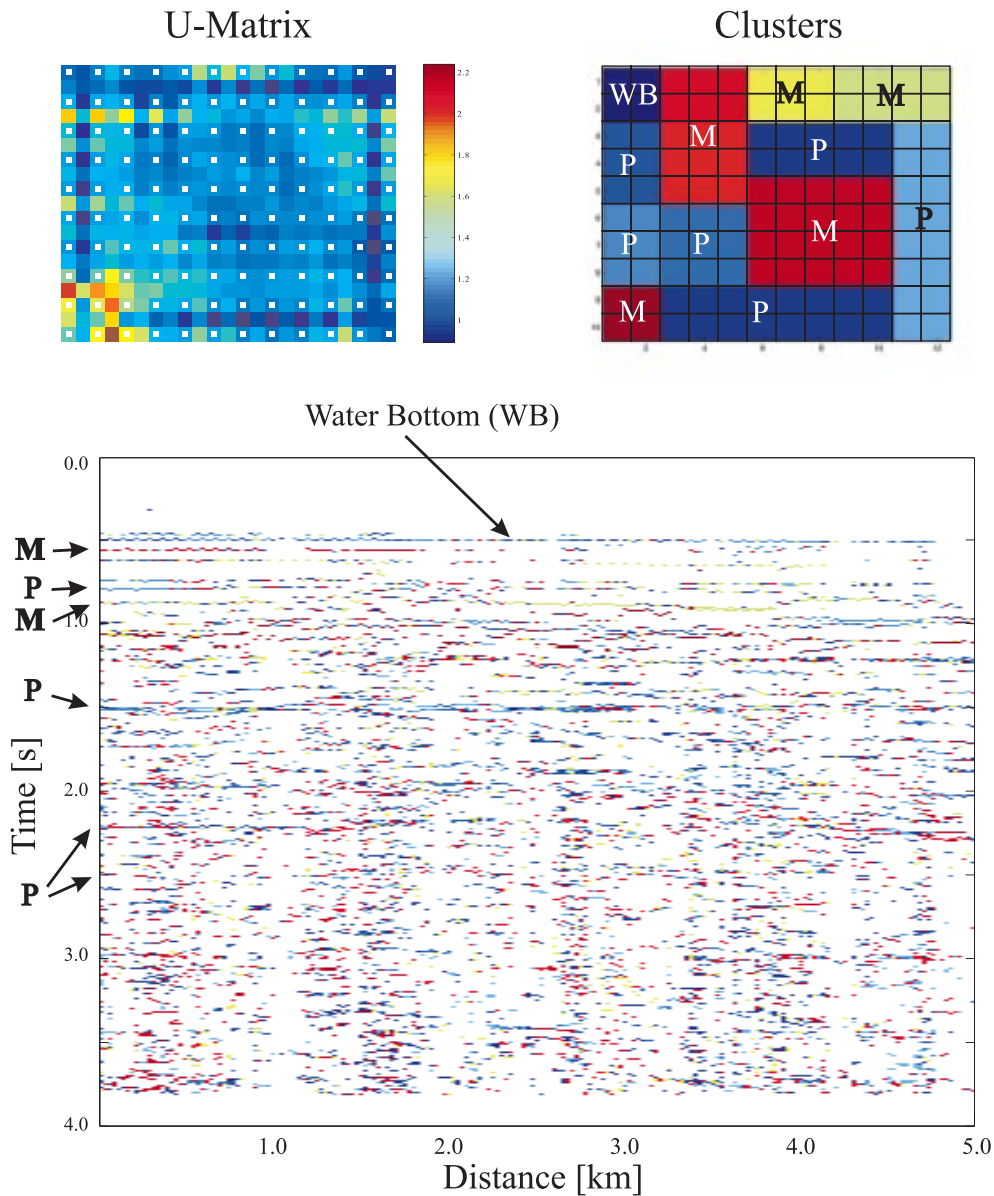


Figure 6.24: Real data set: Cluster analysis for using all wavelet and wavefront attributes, including zero-offset time and stacking velocity. The clusters on the map (top right) have been determined on the basis of the U-matrix (top left). The events in the ZO-section below have the same colors as the clusters they have been classified to. Primaries (“P”) mainly cluster at the left, the bottom and the right side of the map (mainly in blue colors), whereas multiple (“M”) clusters are located at the top and in the center (all other color shades).

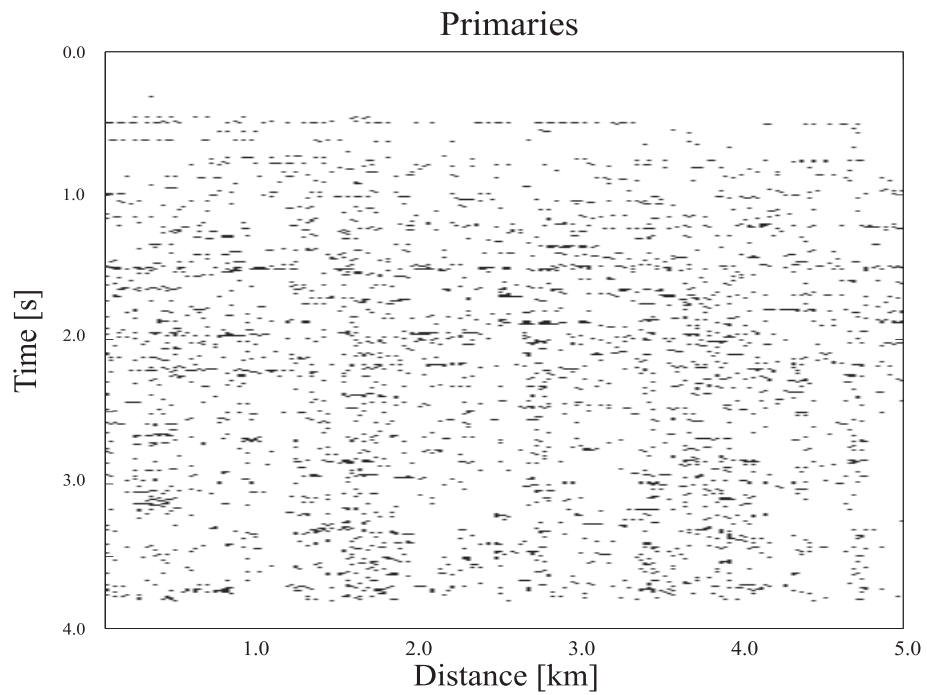


Figure 6.25: The primary wavefield after separating out the multiple wavefield.

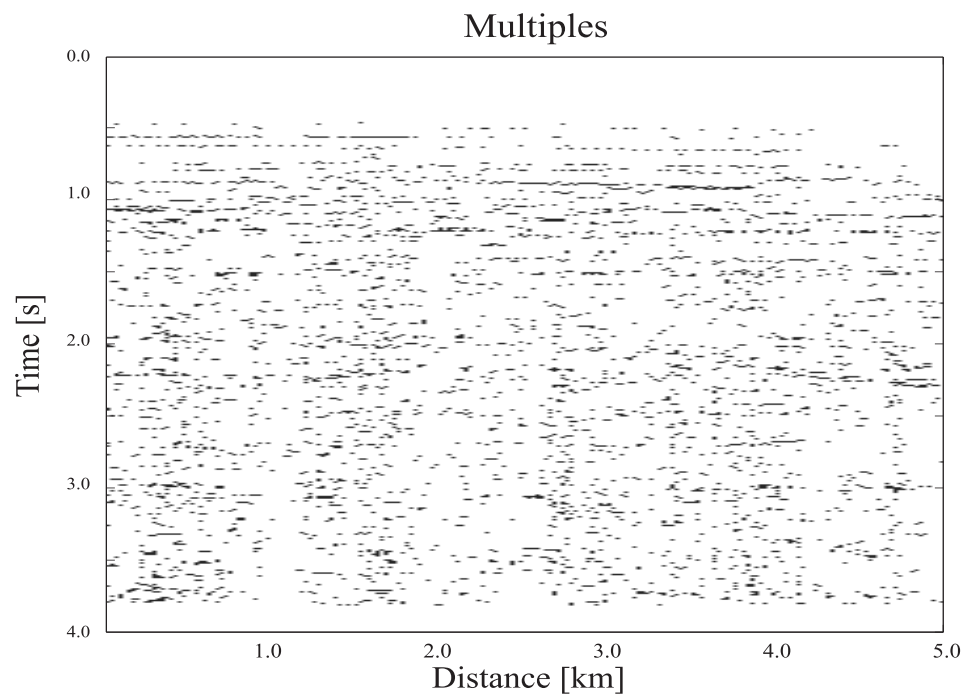


Figure 6.26: The multiple wavefield after separating out the primary wavefield.

6.4 Conclusions

A by-product of investigating the clustering ability of the attribute based self-organizing map, is that by judging the quality of the class separation we obtain criteria to judge the significance of a particular attribute to separate primaries from multiples. In other words, if we train the net by switching on and off certain attributes, we can obtain a measure of the discriminatory power associated with a particular attribute.

An important point is that, when we have trained the self-organizing map with a certain data set, and have obtained a feature map after satisfactory convergence, we need some means of labelling the different clusters on the map.

One possibility is that we color-code the entire map, so that input patterns (attributes of a sample of a seismic trace) falling into a certain region of the map are associated with a particular color. Then we plot the corresponding seismic trace sample in this same color, leading to a seismogram in the colors of the map, with for example blue signifying primaries and red multiples.

Another, more sophisticated method is to label the map with input data where we already know to which class they belong, i.e. if a particular event is a primary or a multiple. This information can be retrieved, for example, from interpreted sections where the interpreter indicates probable candidates for primaries and/or multiples. Data that are already preprocessed with a multiple prediction algorithm also can be used to label input for the SOM feature map.

Chapter 7

A Layer-stripping approach

7.1 The Method

After routine processing of the seismic data, it is often possible for an experienced interpreter to identify certain particular events in a stacked zero-offset section. This judgment is often based on a priori knowledge about the geologic setting in the region. Therefore, I developed a scheme for identifying primaries and multiples by a supervised technique using a backpropagation neural net in a layer-stripping manner utilizing this available information from the interpreter.

A supervised neural net is trained with a few interpreted events - preferably shallow events, since they are easier to interpret. For example the water bottom always can be identified reliably as a primary arrival. The input for the neural net is the same data set as for the SOM classification procedure (chapter 6), i.e. it consists of attributes computed for a set of reflection events.

The details of the method are described in the following:

1. Identify some prominent primaries and multiples in your data set (this is the interpreter's input), preferably for shallow events, since they are often easier to identify.
2. Train a backpropagation neural network with the attribute vectors computed from these events as input and the interpreter's information as the desired output.
3. After convergence of the training, test the trained network with your whole data set, consisting of interpreted events and unidentified events.

4. The algorithm now automatically decides on which events are used for the next training run in addition to the previous training set: these are all events which are classified close enough to 0 or close enough to 1, where 0 signifies a multiple and 1 a primary (the neural net output ranges between [0 ; 1]). In other words, if the algorithm classifies an event with a certain confidence (i.e. in the range [0.8 ; 1] or [0 ; 0.2]) as a primary or a multiple, respectively, we take it into the next training file. The layer-stripping aspect comes in by the restriction that we only consider events in a certain time window starting from $t=0$. This time window is increased for every new training run.
5. Repeat steps 2 - 4 until all events are processed.

Obviously, there is the danger that the algorithm wrongly classifies an event with high confidence and consequently takes it as a training pattern for the next training run. This might lead to further misclassifications. However, there is always the possibility of interference by the user: if a clear misclassification is noticed, he can correct for this in the training file that is used in the next training run. The training is stopped when the time window comprises the whole data set and the classification results do not change any more. Figure 7.1 shows a flowchart of the method.

In Figure 7.2 a simple example is shown with only one CDP gather containing 12 events (4 primaries and 8 multiples). The desired output is depicted with red dots, whereas the actual output of the neural net is represented by black dots. A "1" stands for a primary whereas a "0" is a multiple. The gray areas comprise the events of the training set that was used in that training run. The figure at the top on the left shows the initial situation: the interpreter provides the first two primaries (event no. 1 and 2) and the first multiple (event no. 3) for the training set (gray area). The other events are not part of the training set. The panel at the top on the right shows the result of testing all 12 patterns (gray and white area) on the net trained with the first 3 patterns (gray area). The algorithm would incorrectly classify event no. 4 as a multiple and take it for training as such. Here, I interfered and set this particular desired output to 1, and trained now with the first four events. The bottom panel on the left shows the result after testing. Now, event no. 5 is misclassified as a primary (but it is a multiple). I interfered again and set the desired

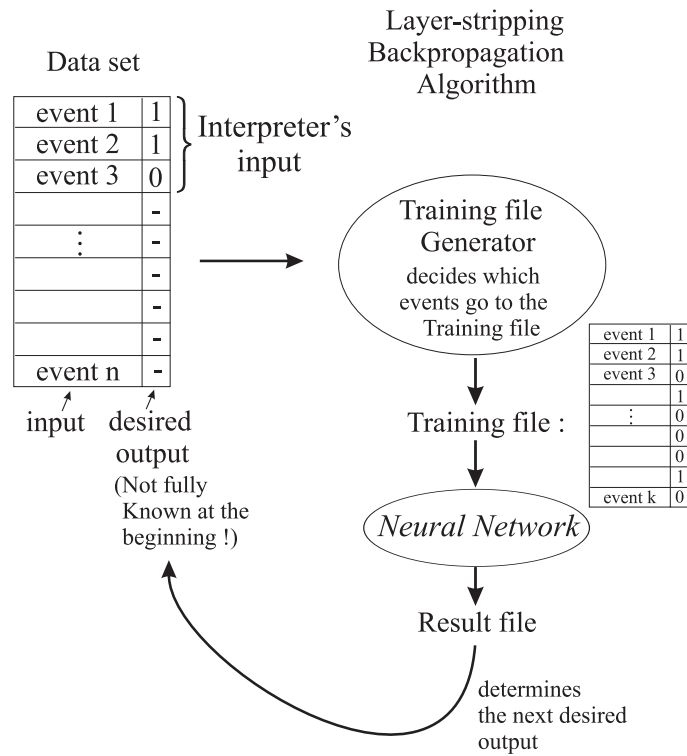


Figure 7.1: Flowchart of the multiple identification method using layer-stripping with backpropagation. For the first training run, only the events that have been identified by the interpreter enter the training file. For all consecutive training runs, those events that are classified within a certain confidence interval are added to the training file.

output for training to 0 for this event. The result of the third training run is shown at the bottom on the right. With the first five events used for training the method correctly classifies most of the other events. Only for event no. 12 the algorithm could not decide if it is a primary or a multiple. We must consider that we only used 12 events, which is no actual basis for neural net usage. This method will now be applied to a larger data set.

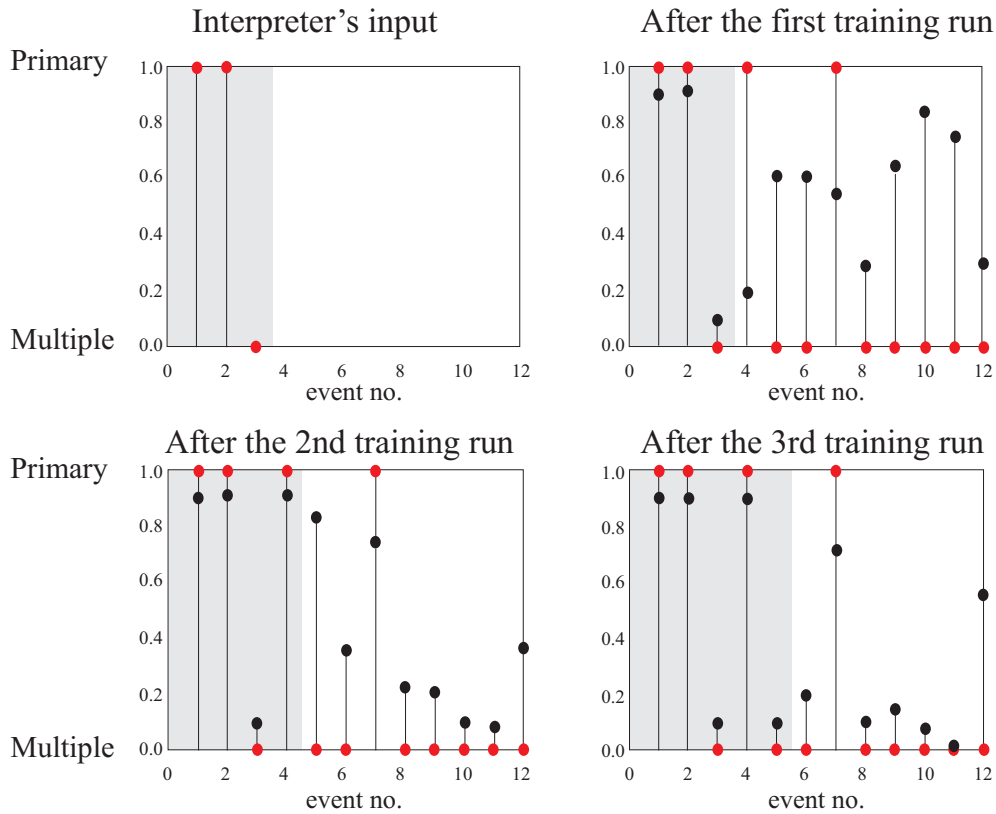


Figure 7.2: Top left: Initial data set for training: the first two primaries and the first multiple are given by the interpreter. (Red is for desired output, black stands for actual output. The gray areas underlie those events used for training in that training run.) Top right: After the first training run, using only the first three events and testing with all 12 events. Bottom left: After the second training run, with the third primary as additional training pattern (given by the interpreter), and testing with all 12 events. Bottom right: After the third training run, with the same data as before plus the second multiple as additional training pattern (given by the interpreter), and testing with all 12 events.

7.2 Synthetic Data Application

The method of separating primary from multiple reflections by using a layer-stripping algorithm on the basis of a backpropagation neural network is applied to the synthetic data set described in the previous chapter (chap. 6). The water bottom and parts of the second primary were considered to be already identified as primaries. The first-order and a part of the second-order water multiple was also taken as the “interpreter’s input”. This was the training set for the first training run (see Figure 7.3). After training the backpropagation network with this data set, it was tested with the entire set of events. Depending on the classification of this test run, the data were split into primaries (for an output in the range [0.8:1.0]) and multiples (for an output in the range [0.0:0.2]). The result is shown in Figure 7.4. Taking the classified data in a time window from 0 seconds down to 2 seconds (see Figure 7.5), a second training run was started. Figure 7.6 shows the result of testing the trained network with the entire data set. There are still some misclassified multiples in the primaries section, but the primaries have a much stronger appearance and thus the multiples have been attenuated considerably compared to the input data.

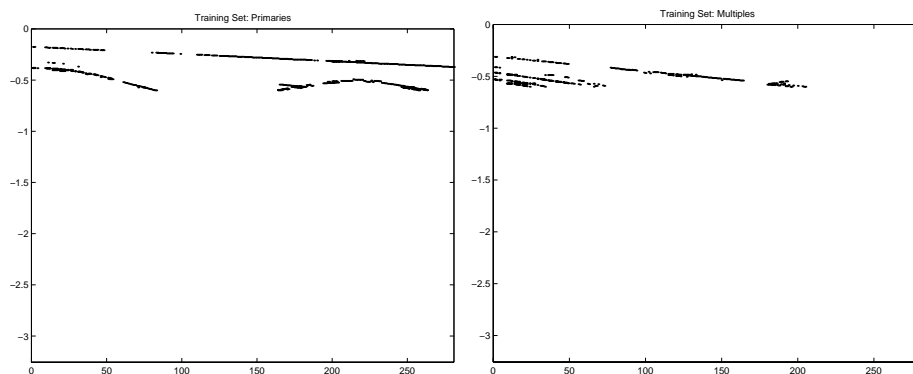


Figure 7.3: Training set for the first training run. Left: primaries, right: multiples.

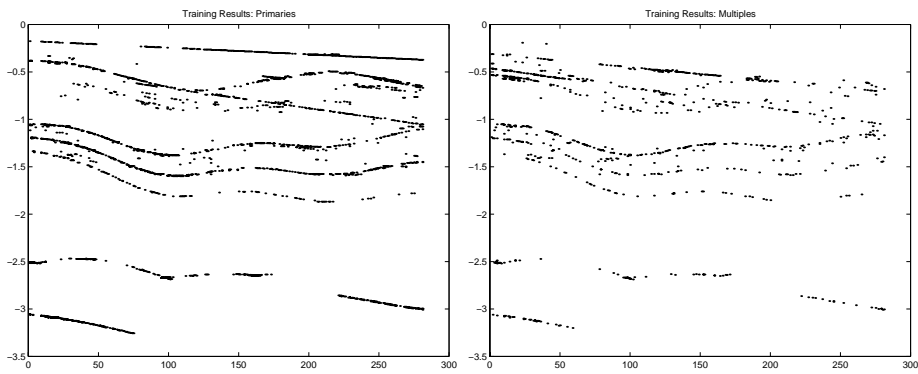


Figure 7.4: Result after the first training run. Left: primaries, right: multiples.

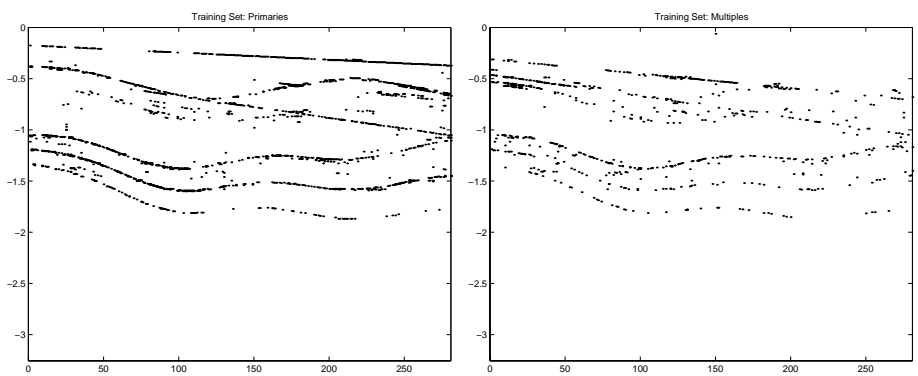


Figure 7.5: Training set for the second training run. Left: primaries, right: multiples.

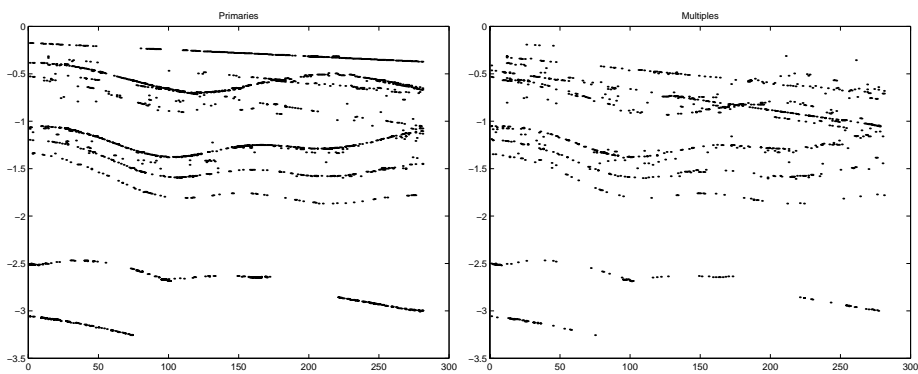


Figure 7.6: Result after the second training run. Left: primaries, right: multiples.

7.3 Conclusions

This method is easy to apply, and relies on the - generally available - interpreted sections as the basis for training a neural network to classify the remaining set of unidentified events. The necessary information is taken from meaningful attributes computed from a set of picked events (primaries and multiples). The algorithm automatically detects if an event belongs to the class “primary” or “multiple”. There is no well-log information needed or any modelling involved. The fact that it is always possible to interfere between subsequent training runs and to correct for obvious misclassifications, make this technique a practical tool for the interpretation of seismic data. The user has the chance to learn together with the neural net - layer by layer - which event is probably a primary and which can be safely regarded as a multiple.

Conclusions

The attenuation and prediction of multiple reflections and the recovery of the desired primary reflections from prestack seismic data has been demonstrated to work successfully on 1-D and 2-D data sets. The methods developed and described in this dissertation can handle both free-surface and internal multiple reflections. Each one of the four presented methods

- multiple prediction and attenuation with backpropagation neural networks,
- attribute based multiple prediction and attenuation with backpropagation,
- attribute based multiple identification with self-organizing maps, and
- the layer-stripping approach using backpropagation

demonstrates the applicability of neural networks for the identification and attenuation of multiple reflections.

Depending on the availability of additional information from well-log measurements, one or the other method is preferable. If there is the possibility to perform high-quality modeling on the basis of sonic and density logs - using the reflectivity method (1D), finite difference schemes (2D, 3D), or ray tracing algorithms (2D, 3D), the supervised techniques (using backpropagation, chapters 4 & 5) perform a reliable separation of the primary and the multiple wavefields. We do not need modeled data for the entire exploration area, but only a few synthetic seismograms at the borehole locations. The neural net then acts as a non-linear filter that interpolates between these points by extracting the necessary information from the field data recorded between the boreholes, or a set of characteristic attributes computed from these data. The modeling of entire profiles in this thesis was performed for validation purposes. This method works fully automatic without user interaction.

In cases where no additional subsurface information in the form of well-logs is available, the unsupervised method (self-organizing map, chapter 6) provides an elegant possibility to classify reflection events and to separate primary and multiple wavefields. The input for this algorithm only consists of a number of selected attributes computed from the seismic data. For labelling the classes, that have been automatically determined by the algorithm, the method requires a few interpreted events. These can be events from shallow depths that are easy to identify. Moreover, this provides the possibility of user interaction with the chance of including a priori knowledge of the geological setting in the exploration area. The layer-stripping approach (chapter 7) pursues the same philosophy. It shows consistent attenuation of the multiple wavefield and provides a powerful tool for processing critical data that do not allow the application of fully automatic algorithms. Working its way down layer by layer, the user provides his interpretation to the half-automatic procedure, that in turn gives back its interpretation and so on.

The idea to combine the discriminatory powers of several parameter domains in the form of attributes proved to be a success. Due to the vast number of existing attributes a careful selection of meaningful attributes that are capable to distinguish between primary and multiple events was necessary. The class of instantaneous attributes only produced good results in combination with attributes computed from the velocity spectrum. The wavelet attributes in combination with the wavefront parameters, the zero-offset time and the stacking velocity demonstrated very consistent results, especially for the unsupervised multiple identification method using self-organizing maps. They should be used for any attribute based wavefield separation technique.

My results show that the neural net approach for multiple removal is promising, especially since it can easily handle non-linear data interrelations. The neural net generalizes from relatively few input seismograms, and tries to remove multiple energy on the remaining major part of the data set on the basis of empirically learned rules. In the case of zero-offset data as well as for entire seismic sections, the neural net method shows that it can reveal the desired information even if data are heavily corrupted by noise. The unsupervised neural net technique only depends on attributes computed from seismic data, and the supervised technique additionally needs a few modeled seismograms. Thus, it can be easily extended to the application in 3-D, since both the computation of attributes and high-quality modeling tools are available for three dimensions.

By using not only seismic data as such, but in addition a number of given attributes, it is possible to improve the discriminatory power of the neural net and to unveil information deeply hidden in the obscurity of the raw seismograms.

Bibliography

- Aki, K., and Richards, P. G., 1980, Quantitative seismology, theory and methods: Freeman, New York.
- Aminzadeh, F., 1997, Model-based neural network and data mining applied to the reservoir parameter estimation problem: 59th EAGE Conference, EAGE, Proceedings of the Workshop on Geophysical Applications of Artificial Neural Networks and Fuzzy Logic, 39.
- Backus, M. M., 1959, Water reverberations - their nature and elimination: *Geophysics*, **XXIV**, no. 2, 233–261.
- Berkhout, A. J., and Verschuur, D. J., 1997, Estimation of multiple scattering by iterative inversion, part I: Theoretical considerations: *Geophysics*, **62**, no. 5, 1586–1595.
- Berryhill, J. R., and Kim, Y. C., 1986, Deep-water peg legs and multiples: Emulation and suppression: *Geophysics*, **51**, no. 12, 2177–2184.
- Bracewell, R., 1965, The fourier transform and its applications: McGraw-Hill.
- Calderòn-Macias, C., and Sen, M. K., 1993, Geophysical interpretation by artificial neural systems: A feasibility study: 65th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 254–257.
- Calderòn-Macias, C., Sen, M. K., and Stoffa, P. L., 1997, Hopfield neural networks, and mean field annealing for seismic deconvolution and multiple attenuation: *Geophysics*, **62**, no. 3, 992–1002.
- Calderòn-Macias, C., 1997, Artificial neural systems for interpretation and inversion of seismic data: Ph.D. thesis, University of Texas at Austin.
- Carrion, P. M., 1986, A layer-stripping technique for suppression of water-bottom multiple reflections: *Geophysical Prospecting*, **34**, 330–342.

- Chen, Q., and Sidney, S., 1997, Seismic attribute technology for reservoir forecasting and monitoring: *The Leading Edge*, **16**, no. 5, 445–456.
- Cichocki, A., and Unbehauen, R., 1993, *Neural networks for optimization and signal processing*: John Wiley & Sons.
- Claerbout, J. F., Black, J., and Biondi, B., 1997, *Basic earth imaging*: Stanford University.
- Essenreiter, R., 1996, *Geophysical deconvolution and inversion with neural networks*: Master's thesis, University of Karlsruhe.
- Fuchs, K., and Müller, G., 1971, Computation of synthetic seismograms with the reflectivity method and comparison with observations: *Geophys.J.R.astr.Soc.*, **23**.
- Haykin, S., 1994, *Neural networks, a comprehensive foundation*: Prentice-Hall.
- Helbig, K., and Brouwer, J. H., 1993, Surface multiples as a tool of calibration in shallow avo studies: 55th Mtg. Eur. Assoc. Expl Geophys., Abstracts, Session:A040.
- Hubral, P., 1983, Computing true amplitude reflections in a laterally inhomogeneous earth: *Geophysics*, **48(8)**, 1051–1062.
- Kennett, B. L. N., 1979, Theoretical reflection seismograms for elastic media: *Geophysical Prospecting*, **27**, 301–321.
- Keydar, S., Landa, E., Gelchinsky, B., and Belfer, I., 1998, Multiple prediction using the homeomorphic-imaging technique: *Geophysical Prospecting*, **46**, 423–440.
- Kneib, G., and Bardan, V., 1997, 3d targeted multiple attenuation: *Geophysical Prospecting*, **45**, 701–714.
- Kohonen, T., 1997a, *Self-organizing maps*: Springer Series in Information Sciences.
- Kohonen, T., 1997b, Som toolbox for matlab: <http://www.cis.hut.fi/projects/somtoolbox/>.
- Müller, T., 1999, *The common reflection surface stacking method, seismic imaging without explicit knowledge of the velocity model*: Ph.D. thesis, University of Karlsruhe.
- Neidell, N. S., and Taner, M. T., 1971, Semblance and other coherency measures for multichannel data: *Geophysics*, **36**, 482–497.

- O'Doherty, R. F., and Anstey, N. A., 1971, Reflections on amplitudes: *Geophysical Prospecting*, **19**, 430–458.
- Peacock, K. L., and Treitel, S., 1969, Predictive deconvolution: Theory and practice: *Geophysics*, **34**, 155–169.
- Peitgens, H. O., Jürgens, H., and Saupe, D., 1992, *Chaos and fractals*: Springer.
- Riedmiller, M., and Braun, H., 1993, A direct adaptive method for faster backpropagation learning: The rprop algorithm: *IEEE International Conference on Neural Networks*, San Francisco, CA, March 28-April 1, 1993, IEEE.
- Robinson, E. A., and Treitel, S., 1980, *Geophysical signal analysis*: Prentice-Hall.
- Schneider, W. A., Prince, E. R. J., and Giles, B. F., 1965, A new data-processing technique for multiple attenuation exploiting differential normal moveout: *Geophysics*, **30**, 348–362.
- Shapiro, S. A., and Hubral, P., 1999, *Elastic waves in random media - fundamentals of seismic stratigraphic filtering*: Springer Lecture Notes in Earth Sciences.
- Sheriff, R. E., 1991, *Encyclopedic dictionary of exploration geophysics*: SEG Geophysical References Series.
- Taner, M. T., Koehler, F., and Sheriff, R. E., 1979, Complex seismic trace analysis: *Geophysics*, **44**, 1041–1063.
- Taner, M. T., 1998, *Program: Interactive geophysical systems*: Seismic Research Corporation (SRC).
- Tarantola, A., 1987, *Inverse problem theory*: Elsevier.
- Thompson, R. F., 1993, *The brain*: Freeman & Co., New York.
- Tygel, M., Müller, T., Hubral, P., and Schleicher, J., 1997, Eigenwave based multiparameter traveltimes expansions: 67th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1770–1773.

- Verschuur, D. J., and Berkhout, A. J., 1997, Estimation of multiple scattering by iterative inversion, part II: Practical aspects and examples: *Geophysics*, **62**, no. 5, 1596–1611.
- Wang, L.-X., and Mendel, J. M., 1992, Adaptive minimum prediction-error deconvolution and source wavelet estimation using hopfield neural networks: *Geophysics*, **57**, no. 4, 670–679.
- Weglein, A. B., Carvalho, P. M., and Stolt, R. H., 1992, Nonlinear inverse scattering for multiple suppression: Application to real data. part i: 62nd Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1093–1095.
- Wiggins, J. W., 1988, Attenuation of complex water-bottom multiples by wave-equation-based prediction and subtraction: *Geophysics*, **53**, no. 12, 1527–1539.
- Wu, Y., and McMechan, G. A., 1996, Prestack full-wave inversion of seismic data with strong multiple reflections: *Journal of Seismic Exploration*, **5**, 275–290.
- Yilmaz, O., 1987, *Seismic data processing*: Society of Exploration Geophysicists, Tulsa, OK.
- Zell, A., 1994, *Simulationen Neuronaler Netze*: Addison Wesley.

Danksagung

Herrn Prof. Dr. Peter Hubral danke ich für die Übernahme des Hauptreferats, für die Schaffung einer sehr angenehmen Arbeitsatmosphäre am Institut und daß er mir die Möglichkeit gab, auf zahlreichen internationalen Kongressen Erfahrungen zu sammeln und Kontakte zu knüpfen. Auch bei Frau Payne möchte ich mich herzlich für die ständige Unterstützung bedanken.

Herrn Prof. Dr. Friedemann Wenzel danke ich für die Übernahme des Korreferats.

Ganz herzlich bedanke ich mich bei Martin Karrenbach, der diese Arbeit betreut hat, und durch seine vielen Ideen und Anregungen sehr geprägt hat. Vielen Dank für Dein stetiges Interesse, die kritischen Korrekturen, und die Möglichkeit zu Projekttreffen und Kursen auf der ganzen Welt zu fahren. Es hat ganz einfach immer Spaß gemacht, mit Dir zusammenzuarbeiten.

Sven Treitel gilt mein besonderer Dank. Schon seit meiner Diplomarbeit sind Sie immer für mich da, wenn ich Rat brauche, und durch Ihre kritischen Fragen habe ich eine Menge gelernt. Für das Versehen meiner Arbeit mit "roten Ornamenten" schulde ich Ihnen noch ein paar Gläschen guten Chianti.

I want to thank Josef Paffenholz and Chad Harding for the constant interest in my work, for the chance to work for three weeks at BHP (Houston), for the ideas, productive discussions and for a great time in a great project.

Many thanks to Tury Taner for providing his attribute program, for his constructive ideas,

for the great discussions, and for his exceptionally good coffee.

Bei Rainer Jäger und Thilo Müller bedanke ich mich für die Bereitstellung ihrer Programme zur Berechnung von Wellenfront-Attributen.

Bei Robert Suna vom Forschungszentrum Informatik bedanke ich mich für die sehr produktiven Gespräche über die Neuronalen Netze und für das Interesse an der Anwendung in der Geophysik.

Joerg Zaske danke ich für die Hilfe mit den Wavefront-Parametern, für die gute Zeit in Karlsruhe, Houston und Tel Aviv und für die lange Freundschaft.

Ein ganz großes Dankeschön an Matthias Riede, Ellen Gottschämmer, Johannes Haury und Kai-Uwe Vieth für das hilfreiche Korrekturlesen dieser Arbeit, vor allem aber für eine sehr schöne Zeit in Karlsruhe. Bessere Freunde und Kollegen kann man sich nicht wünschen. Es geht auch nichts über Hackbraten im Klinikum oder LKW auf dem Sofa.

Lebenslauf

Persönliche Daten:

Name: Robert Essenreiter
Geburtsdatum: 17.04.1968
Geburtsort: Isny im Allgäu
Nationalität: Deutsch

Schulbildung:

1974 – 1978 Grundschule in Christazhofen
1978 – 1987 Gymnasium Isny im Allgäu

Studium und Promotion:

1989 – 1996 Studium der Geophysik an der
Universität Karlsruhe
1993 – 1994 Studium der Geophysik an der
University of Hawaii at Manoa
27.08.1996 Diplom
seit 1996 Doktorand an der Fakultät für
Physik der Universität Karlsruhe