Sequential Transfer Machine Learning in Networks: Measuring the Impact of Data and Neural Net Similarity on Transferability

Robin Hirt
KIT / prenode
Karlsruhe, Germany
robin@prenode.de

Akash Srivastava
MIT-IBM Watson AI Lab
Cambridge, USA
akash.srivastava@ibm.com

Carlos Berg
KIT
Karlsruhe, Germany

Niklas Kühl
KIT / IBM
Karlsruhe, Germany
niklas.kuehl@kit.edu

Abstract

In networks of independent entities that face similar predictive tasks, transfer machine learning enables to re-use and improve neural nets using distributed data sets without the exposure of raw data. As the number of data sets in business networks grows and not every neural net transfer is successful, indicators are needed for its impact on the target performance-its transferability. We perform an empirical study on a unique real-world use case comprised of sales data from six different restaurants. We train and transfer neural nets across these restaurant sales data and measure their transferability. Moreover, we calculate potential indicators for transferability based on divergences of data, data projections and a novel metric for neural net similarity. We obtain significant negative correlations between the transferability and the tested indicators. Our findings allow to choose the transfer path based on these indicators, which improves model performance whilst simultaneously requiring fewer model transfers.

1. Introduction

Machine learning is a main driver in the automation of process tasks across industries [1]. Although many industry players face similar problems with similar data structures in areas where machine learning can be utilized, every company typically solves these problems in an isolated manner [2]. From a systems perspective, these analytical tasks are well-comparable [3].

In an ideal world with an exhaustive exchange of all data across company borders, companies could solve similar problems in a more efficient manner [4, 5]. However, due to competition and first and foremost, due to the preservation of intellectual property and privacy, a sharing of raw data is not feasible. From an economic standpoint, this poses a significant inefficiency as similar problems are solved multiple times and no analytical knowledge is exchanged. Additionally, the creation of analytical models is typically costly. If every company builds its own models, every company would end up with an inferior model as substantially more data potentially exists in the entire ecosystem. Moreover, every company would also have to reinvent the wheel, thus resulting in higher costs for model creation. Therefore, the current industry practices result in an inefficient resource utilization from a system's viewpoint [6].

To address this challenge, we propose the utilization of sequential transfer machine learning, a technique that enables to reuse and improve predictive machine learning models using different, distributed data sets. Hereby, no raw data exchange between companies is required, yet the sequentially transferred model can be improved by leveraging these different data sets. Although different types of analytical models could be transferred, neural networks are especially suited for transfer machine learning and are thus subject of the majority of related work [7]. Multiple studies demonstrate the effectiveness and efficiency of transfer machine learning in well-known, well-formed data sets like MNIST or ImageNet, but a lack of real-world industry studies is evident. One reason, amongst others, is the question on when to transfer, since (naturally) not every neural net can be transferred to every data set [7]. As our research gap, we observe a lack of techniques for identifying the impact of a neural net transfer prior to the transfer itself—which can be described as the transferability of a neural net. For the work at hand, transferability in general can be defined as the estimation of the extent to which representations learned from a source task can help in learning a target task [8]. This is especially relevant when considering large numbers of participants in an ecosystem and a correspondingly high amount of potential neural nets candidates for transfer.

To address this gap, we perform an empirical study on a real-world use case with the aim to study the effects between different similarity measures and the transferability of neural nets. Precisely, we are interested in indicators for transferability of neural

URI: https://hdl.handle.net/10125/71472
978-0-9981331-4-0

nets that are based on a comparison of data and data projections as well as on the neural nets themselves. As a basis for this study, we consider a unique data set of an ecosystem of different restaurant branches owned by different legal entities, all of whom need to perform sales forecasts to improve their respective resource allocations. As owners fear to expose data outside their restaurant, they are not willing to share raw data. Therefore, they are in need of a pre-transfer analysis on the possibility of value-adding neural nets without having to access the raw data of the competitor.

The paper at hand is structured as follows: In the remainder of this section, we cover related work (section 1.1), elaborate on our contribution to theory, define prerequisites and derive hypotheses (section 1.2). Then, we introduce the data set (section 2.1), present the neural net structure and the transfer (section 2.2), and elaborate on indicators for transferability based on raw data, data projections (section 2.3) and neural nets (section 2.4). Afterwards, we present the results by first describing the performance impact of transferring neural nets in a business network (section 3.1). Then, we describe the impact of the tested indicators on transferability (section 3.2). After discussing our findings (section 4), we summarize the results, discuss their generalization, recognize limitations, and show future research prospects (section 5).

1.1. Related Work

The foundations of transfer learning are surveyed by [7] who provide a detailed overview on transfer learning. A wide variety of studies on the application of transfer learning can be identified: [9] present findings on the utilization of deep convolutional neural networks (CNN) in medical image analysis. They use large, general pre-trained sets and adapt them to a specific task to show that pre-trained CNNs using computer vision databases (e.g., ImageNet) are useful in medical image applications and that multi-view classification is possible without the pre-registration of the input images. [10] reports that pre-trained word vectors for sentence-level classification tasks can be seen as universal feature extractors that can be utilized for various classification tasks. In this study, we focus on investigating the transferability [8] of neural networks from a source to a target domain. Related work can be divided into three main aspects that can indicate the transferability, namely the task similarity, the data similarity and, recently, also the model similarity. Table 1 summarizes the related work on transferability in alignment with the aforementioned three main research categories. A variety of work covering the topic of task similarity in transfer learning exists. [11] classify tasks that are correlated and dependent, thus proving that concepts that were previously learned on one task may be transferred to other tasks. [12] state that the transferability is negatively affected by the specialization of higher layer neurons of their source task, which eventually leads to a performance decrease on the target task. Another way to determine the transferability of neural nets is to examine the source and target data set itself. [13] use the similarity among data points in order to update the detection score of the classifier and its classification boundary. [14] find suitable training instances from other domains by measuring the distance between the source and target data in the domain of oil-prize forecasting. [9] apply density ratio weighting to overcome the difference in marginal distributions However, there are more methods for comparing data distributions that could indicate transferability, such as divergences or distances [15].

Publication	Task Similarity	Net Similarity	Data Similarity
Dirichlet Priors [11]	x		
Transfer features[12]	x		
SVCCA [16]		x	
Representational similarity[17]		x	
Data set CV[13]			x
Cascade models [9]			x
This work		x	x

Table 1. Excerpt of related work on transferability.

Especially if the source data set is not available or cannot be accessed due to confidentiality reasons, examining a potential source neural network can be a way to gain insights on its transferability to a target data set. To the best of our knowledge, there is no work on finding indicators for transferability based on net structures. However, recent work shows possibilities for the comparison of neural net similarity using SVCCA [16] to interpret neural network representations. Hereby, given a joint input, SVCCA analyzes the different layer-wise output of two neural networks (their activations). Due to the applied transformations, SVCCA allows the comparison between different layers and networks and is fast to compute. Previous studies apply SVCCA to compare net similarity across a group of CNNs, demonstrating that networks that generalize converge to more similar representations than networks that memorize [17].

In the course of this work, we are interested in transferring models across different data sets for which the data distribution may vary, but not the task to be executed. As we assume, that different entities are trying to solve the same task but the data follows different distributions, we disregard methods that are purely based on task similarity. We are interested in finding ways to receive indications on the transferability in a case where data cannot be pooled (e.g. due to confidentiality issues). Additionally, as computational capacity might be limited in a distributed system, limiting the required computational complexity is desired. Thus, in this study, we focus on SVCCA as a means of comparing two data sets and finally learning about the transferability between them. To get an estimate of the basic indication of data similarity in transfer learning, we compare "raw" data sets. Then, in order to potentially reduce the amount of exposed information during the comparison, we examine ways to compare projections of those raw data sets. Given that even those projections might not be retrievable in some cases (e.g. in cases where only models are exchanged and initial training data is not accessible), we finally aim to find indicators for transferability based on the structure of a neural net.

The contribution of this work is threefold: 1 We develop and evaluate a multi-step system-wide transfer on a unique data set in the domain of sales forecasting. 2 We empirically show an association between the divergence of data distributions and the divergence of projection of data distributions with respect to the transferability of models. 3 We empirically show that the SVCCA is associated with the transferability.

1.2. Prerequisites and Research Design

In our case, we want to transfer neural networks across different distributed data distributions p_l of L companies:

$$\{p_l | l \in \{1, ..., L\}\}. \tag{1}$$

We define the input of L different data sets X^l that are composed of B samples of a neural network η as follows:

$$X^l = \{x_i\}_{i=0}^{B} | x^l \in \mathbb{R}^N. \tag{2}$$

The test inputs T^l and the corresponding true labels V^l are composed of $h < B$ samples and are constructed by sampling uniformly from X^l:

$$T^l = \{t_i\}_{j=0}^{h} | V^l = \{tr_j\}_{j=0}^{h} | h < B. \tag{3}$$

The performance M of a neural network η_{p_l} trained on p_l with predicted labels $\eta_{p_k}(X^l)$ is denoted as:

$$M(\eta_{p_l}(X^l), V^l) \tag{4}$$

The performance delta ΔM of a source neural network η_{p_k,p_z} which is trained on a distribution p_k and then transferred to a target distribution p_z is described as

$$\begin{aligned} \Delta M(\eta_{p_k}, \eta_{p_k,p_z}) = M(\eta_{p_k,p_z}(X^z), V^z) - \\ M(\eta_{p_z}(X^z), V^z) | \Delta M(\eta_{p_z}, \eta_{p_k,p_z} \in \mathbb{R}^N. \end{aligned} \tag{5}$$

We define $\Delta M(\eta_{p_k}, \eta_{p_k,p_z})$ as the transferability of a model that is trained on the source distribution p_k and transferred to the target distribution p_z. In our case, we therefore regard transferability as a performance increase of a neural network from one (source) distribution to another (target) distribution. The first goal of our work is to show that transferability, i.e. the performance increase of a transferred model, exists for the regarded problem/data set. Therefore, we formulate our first hypothesis as follows:

Hypothesis 1 (H1): A model η_{p_k,p_z} which is pre-trained on a distribution p_k and transferred to a distribution p_z outperforms a model η_{p_z} $\iff$ $\Delta(\eta_{p_z}, \eta_{p_k,p_z}) \geq 0$.

If this hypothesis can be confirmed, the next step of this work consists of identifying possible indicators for transferability in advance to the transfer itself. In order to do so, we analyze indicators for the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$ by comparing p_k and p_z directly as well as their respective projections. Hereby, a projection f maps a distribution p as follows:

$$f : \mathbb{R}^a \longrightarrow \mathbb{R}^b \tag{6}$$

The projected distribution is $f(x_i)$. To empirically test different projections, we apply Multidimensional Scaling (MDS), Principal Components Analysis (PCA) and t-stochastic Neighborhood Estimation (t-SNE).

$$f \in MDS, PCA, t - SNE. \tag{7}$$

To compare two distributions p_k and p_z and their respective projections we calculate their data divergence $D[p_k||p_z]$ and data projection divergence $D[f(x^k)||f(x^z)]$.

In this work, we aim to empirically examine the association between the divergence $D[p_k||p_z]$ of data distributions p_k and p_z, the divergence $D[f(x^k)||f(x^z)]$ of projected distributions $f(x^k)$ and $f(x^z)$ and the performance impact $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$. Accordingly, we formulate Hypothesis 2 and 3:

Hypothesis 2 (H2): The divergence of two distributions p_k and p_z, described as $D[p_k||p_z]$, correlates with the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$.

Hypothesis 3 (H3): The divergence of the projection of two distributions $f(x^k)$ and $f(x^z)$, described as $D[f(x^k)||f(x^z)]$ correlates with the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$.

Finally, we examine neural nets themselves without accessing the source data to find indicators for transferability. Therefore, we consider the Singular Value Canonical Correlation Analysis (SVCCA). SVCCA enables the comparison of the behavior of neural nets, derived by the activations of neurons with regard to a data input d^z. Let $\rho = (\eta_{p_k}, \eta_{p_z}, d^z)$ denote the result of an SVCCA between a net η_{p_k} and a net η_{p_z} based on a data sample $d^z \subseteq x^z$. Accordingly, we formulate Hypothesis 4:3

Hypothesis 4 (H4): The output of a SVCCA $\rho(\eta_{p_k}, \eta_{p_z}, d^z)$ correlates with the transferability $\Delta M(\eta_{p_z}, \eta_{p_k}, p_z)$.

For H1, we perform a two-sided one-sample t-test for the mean of all transferabilities to test if the average transferability significantly deviates from zero. For H2-4, we calculate Spearman's rank correlation coefficient as a non-parametric measure between the variables and test the significance of the calculated Spearman's rho r_s.

2. Experiment

In this chapter, we first give an overview of the data we examined and subsequently elaborate on the sales forecasting model design and the transfer mechanism. In conclusion, we describe how we compare data and data projections. Lastly, we present the applied variation of measuring the net similarity via SVCCA.

2.1. Data Set

We analyze unique daily sales data of six different restaurant branches of two particular restaurant chains that serve different types of food. The data set captures observations from 2013 until 2017.

Branch	1	2	3	4	5	6
Company	A	A	A	B	B	B
City	a	a	b	a	c	d

Table 2. Overview of available data for branch 1 to 6 (sales data from 2013-01-01 to 2017-12-31).

By precisely predicting the sales per day for each branch in the next week, month, or even year, several advantages can be leveraged: based on the revenue and demand, staff schedules can be optimized toward cost savings and a better experience for customers can be delivered. Additionally, the procurement of supplies can be improved, as spoiled food is a main cost-driver for restaurants. Thus, the management of restaurant chains has a major interest to forecast sales for their branches.

Table 2 gives an overview on all the available branch data we use in this work. Each of the two restaurant companies has three branches with different locations. Branch 1, 2 and 4 are located within the same city.

2.2. Forecasting Model Design and Transfer

We aim to build separate models for each data distribution, where one data distribution corresponds to the data set of one branch. Afterwards, these models are transferred to every other distribution and then re-trained. This procedure is repeated until every model has passed through every distribution exactly once (H1). To empirically study the effects of data, data projection divergence and net similarity on the transferability of models, we test all possible transfers in a brute-force attempt and analyze the results a posteriori (H2-H4).

Our goal is to develop a model that is able to forecast daily sales on a weekly basis. There are many ways to design a sales forecasting model, such as ARIMA models, additive, or logarithmic regressions. To simplify our research design, we focus solely on Convolutional Neural Networks (CNN) for multivariate forecast as they have proven to achieve superior results in similar problems in the past [18]. Here, the input X_i of a neural network η is $X_i = \{s_i^n, y_i, m_i, w_i\}$, where s_i^n is a vector of daily sales of the previous sales period, $y_i \in \mathbb{Z}$ denotes the year, $m_i \in Z$ the month and $w_i \in \mathbb{Z}$ the week of the observation. The complete data set can be described as $\{X_i\}_{i=0}^{B \times L}$ and $s_j^1 \in \mathbb{R}^7$. Then, the date and time index are adjusted and reformatted in line with the opening hours of the respective branches. The available variables are grouped by day in order to forecast the time series on a per day basis. We clean obvious errors in the data set by dropping erroneous values, such as negative daily revenues.

As a next step, we build a multi-head CNN model to forecast the daily sales of the next sales period. The model has four input variables: revenue of the previous sales period, month, weekday and year of the observation. Each variable is fed into a separate head. All heads consist of two one-dimensional convolutional layers with the same parameter configuration, followed by a max-pooling layer. The output of the pooling layers is flattened and merged by a concatenation layer. The merged heads' output is fed into a first fully connected layer followed by a second one to conduct the interpretation. Finally, the sales forecast for the next period is generated in form of an output vector.

In a pre-test, we determine the model hyperparameters by empirical testing as follows: the two one-dimensional convolutional layers both have 32 filter maps and a kernel size of 3. As activation function, rectified linear unit is applied to both convolutional layers. The pool size for the max-pooling layer is set to 2. The first fully-connected layer contains 200 neurons and the second one 100 neurons. The model is compiled with mean squared error (MSE) as loss function during

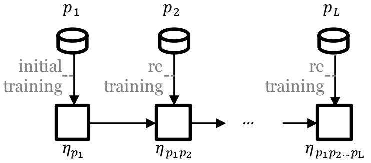

Figure 1. Overview of a possible transfer path for a model across different data distributions.

Degree of transfer	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	Total
Source models	6	30	120	360	720	120
Possible targets	5	4	3	2	1	-
Targets	30	120	360	720	720	1950

Table 3. Number of possible transfers.

training and uses Adam as optimizer. After compilation the model is fitted on the training data set for 20 epochs with a batch size of 16.

For the model training and re-training, we split the data into a training and a test set for each branch. As testing period, we choose the year 2017 consistently. The remaining data builds our training or re-training set. For every model η_{p_x}, we calculate its performance on the actual target data set and on the union of all test sets across all branches for comparability reasons.

To implement the sequential transfer, we re-train a source CNN on a target data set as depicted in Figure 1. Hereby, we do not freeze the layers to enable re-weighting of the neurons in the layers. We re-train the CNN model with the same number of epochs (25) and batch size (16) as in base model training. Note that it would also be possible to adaptively choose certain layers to freeze and dynamically adapt the learning rate. For this study, we chose not to change or vary the amount of training parameters or frozen layers for a transfer. The degree of transfer denotes the total amount of performed transfers per model. In Table 3 we give an overview of all transfers, their respective source models and the respective targets according the degree of transfer. Generally, the amount of transfers grows significantly with a growing number of data sets N and can be described by $\sum_{k=0}^{n-1} \frac{n!}{k!}$.

2.3. Data and Data Projection Divergence

In the following, we first introduce the utilized data divergence measure, which we apply on the unchanged data populations as well as on the projected data. Measuring the independence or divergence of two random variables or distributions can be conducted in different ways. In this work, we estimate the divergence of two data distributions using an energy distance meta estimator $D_{EnDist}(f_1, f_2)$ as equivalent to maximum mean discrepancy [19, 20], which is defined as follows:

$$D_{EnDist}(f_1, f_2) = 2[D_{MMD}(f_1, f_2)]^2 \tag{8}$$

$$D_{MMD} = || \int \{p_k K(;f(x^k))\delta f(x^k) - \int p_z K(;f(x^z))\delta f(x^z)\} ||^2_{\mathbb{H}} \tag{9}$$

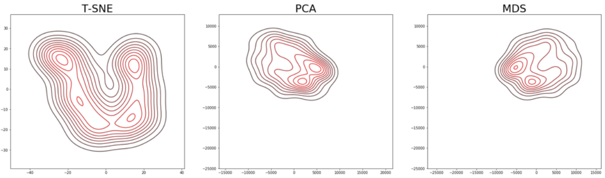

Figure 2. Bi-variate kernel density estimates of data projection (t-SNE, PCA, MDS) for data distribution δ_1 of the first branch.

Considering a scenario where data cannot be exchanged across entities of a system, it is not possible to compare two data sets simultaneously. To ensure a certain degree of confidentiality, a possible solution would be to compare only projected data, where critical information is already lost due to abstraction [21].

Thus, in an initial step we apply projections $f : \mathbb{R}^a \rightarrow \mathbb{R}^b$ raw data $\delta_x \in \mathbb{R}^a$ to retrieve abstractions $\delta_x \in \mathbb{R}^b$ where $a > b$. We use three established algorithms to calculate abstractions of the raw data, namely t-distributed stochastic neighbor embedding (t-SNE), multidimensional scaling (MDS) as well as principal component analysis (PCA). The t-SNE is a well-suited technique for the visualization of high-dimensional data to create meaningful intermediate results and is effective for interactive data analysis [22]. MDS is a technique used for analyzing similarity or dissimilarity of data. It attempts to model the relationship between data as distances in a geometric space [23]. Lastly, PCA decomposes a multivariate data set into a set of subsequent orthogonal components which explain a maximum amount of the variance in the data [24]. The projections for each technique applied to the first data distribution δ_1 are visualized in Figure 2.

Subsequently, we calculate divergences between the data projections. Lastly, for both the raw data and data projections, we evaluate whether a correlation to the transferability of models is given.

2.4. Neural Net Similarity

The SVCCA is a method for analyzing and comparing different representations learned by artificial neural networks [16]. It represents an amalgamation of a singular value decomposition (SVD) and a canonical correlation analysis (CCA) [25].

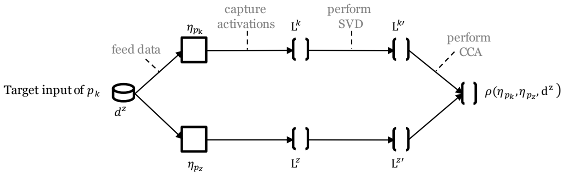

Figure 3. Procedure of comparing of comparing a potential source neural network η_{p_z} to a target net η_{p_k}.

In this work, we use SVCCA to determine the neural net similarity ρ of two networks η_{p_k}, η_{p_z} of two different branches. In Figure 3, we present an overview of the application of SVCCA on a potential source net to identify its transferability to a target distribution.

Two neural nets η_{p_k}, η_{p_z} that are to be compared are fed with data d^z. In this study we supply a data sample d^z which represents the sales of 2017 from the target distribution to the potential source net and capture the activation vectors for every layer.

The neurons' response is calculated as a representation over a finite set of inputs. The resulting activation vectors L^k for each layer of neurons are then processed by applying SVD. Similar to the eigenvalues, these characterize the properties of the matrix. This results in singular vectors $L^{k'} = (\{x'_1, ..., x'_{m'_1}\})$ with associated singular values for X and similarly for Y. Of these singular vectors we keep the top (m'_1), as 99% of variation of X is explained by the top (m'_1) vectors. This helps to remove directions with respect to neurons that are constantly zero or exhibit noise with small magnitudes [16].

Subsequently, CCA is applied to the sets of top singular vectors (m'_1). The CCA is a well-established method for understanding the similarity of two different sets of randomly distributed variables. Given the two sets of vectors $(\{x'_1, ..., x'_{m'_1}\}, \{y'_1, ..., y'_{m'_2}\})$, we wish to find linear transformations (W_X, W_Y) that maximally correlate with the sub-spaces. This can be reduced to an eigenvalue problem. Solving this problem results in linearly transformed sub-spaces with directions $(\tilde{x}_i, \tilde{y}_i)$ that are maximally correlated with one another. As a result, we ultimately obtain $\rho = (\eta_{p_k}, \eta_{p_z}, d^z)$ as the transferability of a source neural net η_{p_k} towards a target data set η_{p_z}.

3. Results and Discussion

We present the results of this study along two steps. First, we describe the result of the initial net training and the performed transfers—thus addressing H1. Second, we describe the output of the analysis on the association between data, data projection, neural net, and their impact on transferability—thus addressing H2-H4.

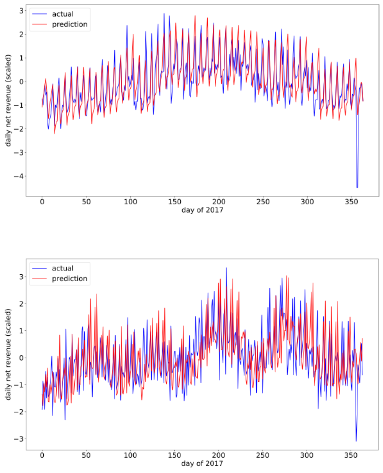

Figure 4. Scaled daily net revenue, actual and predicted; Above: branch 1, below: branch 4.

3.1. Base and Transfer Results (Hypothesis 1)

To measure the performance of the developed forecasting models, two metrics are used: Root Mean Squared Error (RMSE) [26] and Mean Absolute Percentage Error (MAPE) [27]. The RMSE is used to calculate the differences between values predicted by a model and the actual values observed and, in this work, is a basis during model optimization. RMSE as a scale-dependent measurement is not suitable for comparing forecasting errors across different data sets. Thus, to evaluate and compare the performance of different models on different data sets, we additionally calculate the MAPE. The MAPE delivers a very intuitive interpretation in terms of relative error and therefore MAPE is broadly used in practice

We train base models for every branch based on all available data including 2016. Then, we test the models on the full year of 2017 and calculate the MAPE and RMSE. In Figure 4, we depict the scaled daily net revenue (exemplarily) for branch 1 and branch 4. Both base models are seemingly good in predicting the actual value. However, it is also noticeable that between those two data distributions—and, thus, models—there are significant differences in sales patterns.

As shown in Table 3, the number of potential transfers and therefore the number of possible models that are evaluated grows exponentially. Not all transfers have a positive impact on the performance to a target distribution (see results of first degree in comparison to base models in Table 4). This indicates that transferability varies, depending on the association between the source and target distribution. Additionally, in practice it might not be feasible to test all possible transfer model candidates on a target set, as a transfer and re-training of a model is bound to a computational cost. Simply testing all possible combinations via a brute-force approach would therefore not be efficient.

In Table 4, the best results for each branch according the degree of transfer are presented. Note that we select the best performing model for every transfer step and every branch. For almost all branches, except branch 5, an increase in prediction performance can be observed with an increasing degree of transfer. In case of branch 5, we observe an increase of performance starting after the third transfer. It is noticeable that the increase steadily grows for every transfer step, albeit in some cases marginally.

With an increasing degree of transfer, we can observe that in some cases the same distributions are used to re-train models. If, for instance, we investigate target branch 1, we can observe that branch 4 and 5 seem to be good previous distributions to train a model on. However, as we always re-train the complete net, an information loss is likely to arise after multiple transfer steps. H1 states that a model η_{p_k,p_z} which is pre-trained on a distribution p_k and transferred to a distribution p_z outperforms a model $\eta_{p_z} \iff \Delta M(\eta_{p_k}, \eta_{p_k,p_z}) \geq 0$. Thus, a two-sided one-sample t-test for the mean of all transferabilities ΔM (N=1950) is conducted to test if the average transferability significantly deviates from zero. With a mean of 0.00894, a standard deviation of .06728 and a p-value ¡.0001, the test confirms that the average transferability is positive. Thus, H1 is supported. Although the mean of ΔM is only slightly above zero, Table 4 illustrates that there is a steady increase of performances with every further transfer step. However, in that scenario, the best performing models are cherry-picked. In reality, it would not be desirable to test all 1950 transferred models, e.g., due to computational cost. Thus, it is desirable to know in advance which models will perform best. This leads us to the study of association on transferability.

3.2. Associations on Transferability (Hypotheses 2-4)

Returning to our previously defined research gap, we aim to find an indicator of transferability between two data distributions without comparing them directly. By establishing and testing H1, we first show the utility of a transfer in our use case. Now, we empirically study the correlation between three influence factors on transferability: the data divergence (H2), the projected data divergence (H3) and the SVCCA (H4). For every hypothesis, we calculate Spearman's rank correlation coefficient between the transferability and the corresponding indicator. The coefficient describes both the strength and the direction of the relationship. The Spearman correlation evaluates the monotonic relationship between the two continuous variables: transferability and the corresponding indicator. The results are presented in Table 5. We split H3 into three sub-hypotheses corresponding to the differing data projection functions we examine: H3.1 corresponds to the T-SNE, H3.2 to the PCA and H3.3 to the MDS. For every hypothesis, we examine N=1950 transferred models.

Although we do not intend to find indications on raw data as it might not be feasible in business networks due to data confidentiality reasons, we formulate H2 to investigate whether or not there is an association without any transformation of data. H2 states that the divergence of two distributions p_k and p_z, described as $D[p_k||p_z]$, correlates with the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$. Results of the study indicate that there is a significant negative association between the data divergence $D[p_k||p_z]$ and the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$ (r_s=-.4294, p¡.0001).

By projecting data and thus masking confidential information, we state and test different techniques for transferability indicators through H3. Thus, H3 describes that the divergence of the projection of two distributions $f(x^k)$ and $f(x^z)$ described as $D[f(x^k)||f(x^z)]$ correlates with the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$. The sub-hypotheses H3.1-3.3 describe different projection functions, respectively. For H3.1, results indicate that there is a positive association between the projected data divergence $D_{TSNE}[f(x^k)||f(x^z)]$ based on the T-SNE projection and the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$ (r_s=-.0668, p¡.05). However, the Spearman's rho is rather low which indicates a weak correlation between the two variables. In the case of H3.2, however, the results paint a clearer picture: a negative correlation between the projected data divergence $D_{PCA}[f(x^k)||f(x^z)]$ and the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$ is present (r_s=-.3101, p¡.0001). A similar situation can be observed by considering the results of H3.3, where we find an even higher negative correlation between the projected data divergence $D_{MDS}[f(x^k)||f(x^z)]$ based on MDS to the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$ (r_s=-.3101, p¡.0001). Based on the results for H3.1-3.3, we can derive that the PCA and the MDS are better aligned with the identified

Target p_i	**base**	**1**st **degr.**	**2**nd **degr.**	**3**rd **degree**	**4**th **degree**	**5**th **degree**
Br. 1 (p_1)	9.59 (p_1)	9.18 (p_6,p_1)	9.08 (p_5,p_4,p_1)	8.98 (p_3,p_5,p_4,p_1)	8.96 (p_2,p_4,p_6,p_5,p_1)	8.96 (p_5,p_6,p_3,p_2,p_4,p_1)
Br. 2 (p_2)	13.31 (p_2)	12.52 (p_3,p_2)	11.87 (p_6,p_3,p_2)	11.73 (p_3,p_5,p_1,p_2)	11.65 (p_3,p_4,p_6,p_1,p_2)	11.70 (p_3,p_4,p_6,p_1,p_5,p_2)
Br. 3 (p_3)	13.94 (p_3)	13.84 (p_2,p_3)	13.76 (p_2,p_6,p_3)	13.38 (p_2,p_6,p_1,p_3)	13.25 (p_6,p_5,p_1,p_2,p_3)	13.01 (p_2,p_6,p_4,p_5,p_1,p_3)
Br. 4 (p_4)	11.88 (p_4)	10.64 (p_2,p_4)	10.33 (p_2,p_6,p_4)	10.18 (p_3,p_1,p_2,p_4)	10.22 (p_5,p_2,p_6,p_3,p_4)	10.03 (p_2,p_6,p_5,p_3,p_1,p_4)
Br. 5 (p_5)	23.00 (p_5)	24.71 (p_3,p_5)	23.19 (p_3,p_6,p_5)	22.42 (p_6,p_1,p_4,p_5)	22.16 (p_1,p_6,p_4,p_3,p_5)	21.98 (p_6,p_2,p_1,p_3,p_4,p_5)
Br. 6 (p_6)	13.26 (p_6)	12.82 (p_4,p_6)	12.95 (p_2,p_5,p_6)	12.42 (p_3,p_1,p_5,p_6)	12.49 (p_2,p_5,p_1,p_4,p_6)	12.21 (p_2,p_3,p_1,p_5,p_4,p_6)

Table 4. MAPE M (the lower the better) of best model along degrees of transfer for each distribution p_i with the corresponding transfer path in brackets.

H	**Transferability between** η_{p_k} **+** η_{p_z}	r_s
H2	Data divergence $D[p_k\|\|p_z]$	-.4294***
H3.1	$D[f_{TSNE}(x^k)\|\|f_{TSNE}(x^z)]$	.0668**
H3.2	$D[f_{PCA}(x^k)\|\|f_{PCA}(x^z)]$	-.2397***
H3.3	$D[f_{MDS}(x^k)\|\|f_{MDS}(x^z)]$	-.3101***
H4	$\rho(\eta_{p_z}, \eta_{p_k,p_z}, d^z)$	-.2245***

”*” means $p < .05$, ”**” means $p < .01$, ”***” and means $p < .001$.

Table 5. Spearman correlation of all tested indicators for transferability.

correlation between data divergence and transferability (H1), as the direction of their correlations towards the transferability is the same. Furthermore, in case of the T-SNE, we only see a weak positive monotonous association.

Through the comparison, although not exposing raw, but projected data, a possible breach of confidential information is not unlikely, as certain characteristics of the original data distribution are still extractable from the projection. Thus, we state and test H4 to find indications for transferability by the result of the SVCCA, a measure for neural net similarity. In case of H4, we state that the output of a SVCCA $\rho(\eta_{p_z}, \eta_{p_k,p_z}, d^z)$ correlates with the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z}, d^z)$. Our tests show a similar result as for H2, H3.2 and H3.3. We find a significant negative association between the neural net similarity $\rho(\eta_{p_z}, \eta_{p_k,p_z}, d^z)$ and the transferability $\Delta M(\eta_{p_z}, \eta_{p_k,p_z})$ (r_s=-.2245, p¡.0001).

In summary, we can reject the null hypothesis for H2-H4. However, we observe differences in the results for each tested association. There seems to be a clear negative correlation between the projected data divergence based on PCA and MDS and the transferability as compared to T-SNE. Here, we observe a positive correlation with a Spearman's rho value below .07 whereas PCA and MDS exhibit larger, yet negative Spearman's rho values. Hence, we observe the same direction of correlation between the net similarity and the transferability, which indicates stable results.

4. Discussion

A multitude of insightful results can be derived from the conducted empiric research. First and foremost, what sparks our interest the most is the observed dominant, negative correlation effect between the transferability and the data and data projection divergence and neural net similarity. Based on previous research, one would expect a positive correlation to be present [14]. However, in the regarded case, we assume that a neural network benefits from divergent or different observations which are not available in previous training data.

Additionally, in our case we consider sales data collected by different restaurants. Although the data sets originate from two different chains which serve different types of food, the underlying sales patterns might be quite comparable. Results indicate that the underlying data distribution cannot yet be learned by looking at an isolated data population. Thus, we hypothesize that if a neural net receives a larger amount of diverging observations as inputs, its generalization and hence its performance improve.

Another striking finding can be observed by visually inspecting the projections of data populations and their respective transferability and divergences. Exemplarily, we consider projections derived through MDS and compare a first degree of transfer. In Figure 5, we present two cases where the effect of projected data divergence and the transferability can be visually observed for particularly ”successful” transfers and ”unsuccessful” transfers. In the figure, we can detect a strong support for our hypothesis validation, as successful transfers occur when the data is extremely divergent and vice-versa, unsuccessful transfers occur when data is divergent. However, future work is necessary to further investigate this phenomenon.

Furthermore, the correlations of the data and data projection divergence and their transferability show the same direction as the correlation between the neural net similarity and the transferability. This gives us reasons to believe that the neural net similarity, as applied in this work with SVCAA, represents similar abstracted information as the divergence of data and its projection. It also aligns with previous work [16], by finding representations of features of a data set in a neuron's response. However, this assumption requires further confirmation in future work based on additional empirical research established through other data sets.

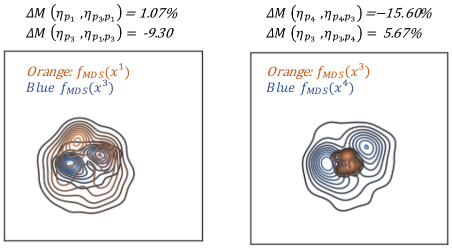

Figure 5. Overlay of bi-variate kernel density estimates of data projections (MDS) in the case of a) $f_{MDS}(x^1)$,$f_{MDS}(x^3)$ and b) $f_{MDS}(x^3)$, $f_{MDS}(x^4)$ and their respective bi-directional transferabilities.

5. Conclusion and Outlook

In this work, we utilize transfer machine learning on a unique sales data set. We do so to reveal two aspects of interest: first, the performance increase—labeled as transferability—of transferring models in general and second, the identification of indicators of a successful transfer prior to the transfer itself.

Therefore, we contribute to the body of knowledge in manifold ways. First, we implement a multi-step system-wide transfer on the sales data of different restaurants and restaurant chains. We successfully show the an empirically analysis the utility of transfers. This is in line with Hypothesis 1, which states that a model that is pre-trained on one distribution and subsequently transferred to another distribution outperforms the model built solely on the original distribution. Secondly, the association of divergence of data distributions as well as the divergence of projections of data distributions and their transferability is analyzed. We are able to confirm Hypothesis 2 and Hypothesis 3 for different sub-distributions, indicating a strong negative correlation between data divergence and data projection divergence and their transferability. Thirdly, we analyze with Hypothesis 4 whether the output of a Singular Value Canonical Correlation Analysis is associated with the transferability. Although we analyze only trained nets—and not data distributions or their projections—we are able to find an association between the neural net similarity and the transferability. In summary, this means for the regarded data set that we are now able to determine transferability of models without regarding raw data—prior to the transfer. As a result, predictions about the transferability for new data sets in a business network can be made, without exposing data distributions. Additionally, its application could allow for more efficiency across the overall system, as the same problem does not need to be solved multiple times: a once trained model can be re-applied several times for similar problems at each restaurant.

Despite the novelty of the approach, there are limitations. As we consider only one case, to theorize the process of general indicators for transferability, more examples are necessary. Additionally, for the time being, we only show an association between data, data projection and neural net similarity and the transferability. On the technical side, the currently implemented transfer mechanism exploits ”forgetting”, i.e., we do not dynamically adapt the frozen layers. Furthermore, the data and data projection association towards transferability neglects previous transfer steps of a model and is thus trivialized. Finally, while no raw data is shared, recent research shows the possibility to retrieve single instances, especially extreme points of a population [28].

Future research needs to address especially the last aspect. If we aim to allow privacy-preserving transfer machine learning, we need to incorporate differential privacy mechanisms into model training [29]. A further enhancement of the transfer mechanism could prove meaningful, for instance by including the freezing of certain layers, as well as adapting the learning rate or number of frozen layers with respect to the degree of transfer. Also, an in-depth investigation of the ”forgetting” aspects of networks could be interesting, e.g., how many transfer steps are required for a network to ”forget” information—and therefore limit the amount of transfers from the beginning. As mentioned previously, more and repeated empirical studies on other data sets, models, and net architectures are necessary to address the generalizability of the approach. Finally, an exploitation of the association between SVCCA and transferability would be preferable. First directions are shown in research to develop methods or search algorithms that utilizes it as a direction of search [30]. This would allow to choose the ”path of transfer” in advance—and result in higher model performances with less model transfer permutations.

Acknowledgements

The authors would like to thank John Cohn, Nghiah Hoang and Gerhard Satzger for their continuous support in this research endeavour.

The authors acknowledge the financial support by the Federal Ministry for Economic Affairs and Energy of Germany in the project Service-Meister (project number 01MK20008).

References

[1] A. Sanders, C. Elangeswaran, and J. Wulfsberg, “Industry 4.0 implies lean manufacturing: Research

activities in industry 4.0 function as enablers for lean manufacturing," *Journal of Industrial Engineering and Management*, 2016.

[2] R. Hirt and N. Kühl, "Cognition in the era of smart service systems: Inter-organizational analytics through meta and transfer learning," *Thirty Ninth International Conference on Information Systems*, 2018.

[3] R. Mizoguchi, J. Vanwelkenhuysen, and M. Ikeda, "Task ontology for reuse of problem solving knowledge," *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing*, vol. 46, no. 59, p. 45, 1995.

[4] R. Hirt, N. Kühl, Y. Peker, and G. Satzger, "How to learn from others: Transfer machine learning with additive regression models to improve sales forecasting," in *IEEE International Conference on Business Informatics (CBI)*, 2020.

[5] T. Karb, N. Kühl, R. Hirt, and V. Glivici-Cotruță, "A network-based transfer learning approach to improve sales forecasting of new products," in *European Conference on Information Systems (ECIS)*, 2020.

[6] J. R. Hicks, "The foundations of welfare economics," 1939.

[7] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data engineering*, pp. 1345–1359, 2009.

[8] Y. Bao, Y. Li, S.-L. Huang, L. Zhang, A. R. Zamir, and L. J. Guibas, "An information-theoretic metric of transferability for task transfer learning," in *International Conference on Learning Representations (ICLR) 2019*, 2019.

[9] E. Zhong, W. Fan, Q. Yang, O. Verscheure, and J. Ren, "Cross validation framework to choose amongst models and datasets for transfer learning," in *Machine Learning and Knowledge Discovery in Databases* (J. L. Balcazar and F. Bonchi, eds.), 2010.

[10] Y. Kim, "Convolutional neural networks for sentence classification," in *Conference on Empirical Methods in Natural Language Processing* , (Doha, Qatar), 2014.

[11] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-Task Learning for Classification with Dirichlet Process Priors," *Journal of Machine Learning Research*, vol. 8, pp. 35–63, 2007.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems 27*, NIPS Foundation, 2014.

[13] V. Jain and E. Learned-Miller, "Online domain adaptation of a pre-trained cascade of classifiers," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[14] J. Xiao, C. He, and S. Wang, "Crude oil price forecasting: A transfer learning based analog complexing model," in *2012 Fifth International Conference on Business Intelligence and Financial Engineering*, pp. 29–33, IEEE, aug 2012.

[15] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[16] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems 30*, pp. 6076–6085, Curran Associates, Inc., 2017.

[17] A. S. Morcos, M. Raghu, and S. Bengio, "Insights on representational similarity in neural networks with canonical correlation," in *Advances in Neural Information Processing Systems 31*, pp. 5727–5736, Curran Associates, Inc., 2018.

[18] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," in *Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence*, pp. 729–730, 2017.

[19] Z. Szabó, "Information theoretical estimators toolbox," *Journal of Machine Learning Research*, vol. 15, pp. 283–287, 2014.

[20] G. J. Székely and M. L. Rizzo, "Energy statistics: A class of statistics based on distances," *Journal of Statistical Planning and Inference*, vol. 143, no. 8, pp. 1249–1272, 2013.

[21] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings - IEEE Symposium on Security and Privacy*, 2008.

[22] N. Pezzotti, B. P. F. Lelieveldt, L. v. d. Maaten, T. Höllt, E. Eisemann, and A. Vilanova, "Approximated and user steerable tsne for progressive visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1739–1752, 2017.

[23] I. Borg and P. Groenen, "Modern multidimensional scaling: Theory and applications," *Journal of Educational Measurement*, vol. 40, no. 3, pp. 277–280, 2003.

[24] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.

[25] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[26] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, pp. 679–688, 2006.

[27] A. De Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, 2016.

[28] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333, ACM, 2015.

[29] I. Mironov, "Rényi differential privacy," *Proceedings - IEEE Computer Security Foundations Symposium*, pp. 263–275, 2017.

[30] F. Peters and R. Hirt, "A transfer machine learning matching algorithm for source and target (tl-mast)," in *International Conference on Machine Learning, Optimization, and Data Science*, Springer, 2020.