Improving Classification Accuracy of Deep Neural Networks by Transferring Features from a Different Distribution

Chetak Kandaswamy[1]
chetak.kand@gmail.com
Luís M. Silva[1]
lmas@ua.pt
Jaime S. Cardoso[2]
jaime.cardoso@inescporto.pt

[1] Instituto de Engenharia Biomédica (INEB),
Rua do Campo Alegre, 823
Porto, Portugal

[2] INESC TEC and Faculdade de Engenharia,
Universidade do Porto, Rua Dr. Roberto Frias, 378
Porto, Portugal

Abstract

Deep Neural Networks (DNN) are feed-forward, artificial neural networks that allow learning of multiple levels of abstraction that help to make sense of data such as images, sound, and text. We study the performance of DNNs using transfer learning approaches. Transfer learning is a process where a network trained on a source problem is reused to solve a new target problem by applying minor modifications to the network. Generally, transfer learning is done when the distributions between the source and target are similar and the task is equal. In this paper, we hypothesis that, if the distance between the distributions is different, and the tasks are different transfer learning will help to improve classification performance and speed up the process.

For this purpose we propose unsupervised feature transference using Stacked Denoising Autoencoder (SDA). In unsupervised feature transference approach we explore: 1) transfer learning between completely different tasks drawn from different distributions and, 2) transfer learning between equal tasks drawn from different distributions. We achieved significant improvement on average error rate and on average computation time using SDA on two types of transfer learning approaches to test our hypothesis.

1 Introduction

The study of transfer learning was inspired by the ability of humans to reuse prior experience under different environments. Naturally, the transfer learning paradigm implies reusing learning machines previously trained for a given source problem S in order to solve, with minor modifications, a different target problem T. An ideal transfer learning method should improve the reused classifier over the one trained from scratch.

Traditionally, transfer learning is applied between problems with same classification task and drawn from similar distributions. In this paper we primarily explore transfer learning between *completely different tasks* drawn from *different distributions* [1], i.e., a classifier learns to perform a task on training instances drawn from the source problem and then reused to perform a different task on a target problem instances drawn from a different distribution. Secondly, we explore transfer learning between problems with same classification task but drawn from different distributions.

We analyze Deep Transfer Learning (DTL) (transfer learning with deep architectures) using an unsupervised feature transference (USDA) approach [1]. We use state-of-the-art deep learning methods like SDA (see [2],) that learn high-level features from large datasets. In order to distinguish how different the target distribution is from the source distribution, we use Jensen-Shannon divergence (JSD) as a metric to measure the degree of heterogeneity between distributions.

2 Problem Formulation: Transfer Learning

Given an input space X and a set of labels Y, a classifier is any function $g(\mathbf{x}) : X \rightarrow Y$ that maps instances $\mathbf{x} \in X$ to labels. Essentially, Y is a coding set for the labels using some one-to-one mapping (e.g., $\Omega = \{\text{“}equilateral\text{”}, \text{“}circle\text{”}, \text{“}square\text{”}\} \rightarrow Y = \{0,1,2\}$ with number of labels $c = 3$). We assume that n_{ds} instances are drawn by an i.i.d. sampling process from the input space X with a certain probability distribution $P(X)$, thus giving a design data set $X_{ds} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_{ds}}\}$ which is accompanied by a set of label codes $Y_{ds} = \{y_1, \ldots, y_{n_{ds}}\}$ for each instance. The classifier performance is measured using error rate ε and computation time on a test set $X_{ts} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_{ts}}\}$ with n_{ts} unlabeled instances drawn from the same distribution $P(X)$.

Traditionally, the goal of transfer learning is to transfer the learning (knowledge) from a source problem input space X_S to one or more problems or distributions to efficiently develop an effective hypothesis for a new task, problem or distribution. In this framework of transfer learning, the source and target problems may come from equal or different distributions. In supervised learning problems, the source Y_S and target Y_T labels may be equal or different. We focus mainly on two cases where the distributions are different:

1. **(TL: case 1)** The distributions are different $P_S(X) \neq P_T(X)$ and the labels are different $Y_S \neq Y_T$.

2. **(TL: case 2)** The distributions are different $P_S(X) \neq P_T(X)$ and the labels are equal $Y_S = Y_T$.

Under such hypothesis, our goal is to obtain an accurate classification for target-domain instances by exploiting labeled training instances from the source-domain.

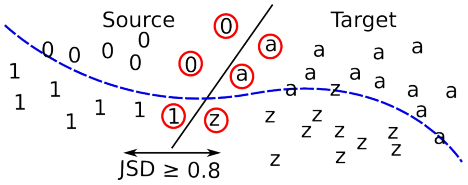

Figure 1: (TL: Case 1) Classify letters reusing digits

Comparing distributions: We use JSD as a measure to compute the difference between the distributions of two datasets. Given two probability functions $P_S(X)$ and $P_T(X)$, when $D_{JS}(S||T) = 0$, we can consider that two distributions are identical and when $D_{JS}(S||T) < 0.5$, the distributions are similar and when $D_{JS}(S||T) \geq 0.5$, the distributions are different.

2.1 Datasets

We use seven datasets of color and gray scale images, as described in Table 1. These seven datasets are either distinct in number of labels or distributions.

Latin and Arabic datasets are representative names for the well-known MNIST[1] and MADbase[2] datasets of handwritten Latin and Arabic digits, respectively. The original Chars74k[3] dataset of typed characters was resized to 28×28 pixels and then broken into three smaller datasets: Digits dataset contains digits from 0-to-9, the Lowercase dataset contains lowercase letters from a-to-z and finally, the Uppercase dataset contains uppercase letters from A-to-Z.

Shape1 and Shape2 are from Baby AI shape dataset[4] which was used for shape recognition tasks. Shape1 consists of images of canonical shapes (equilateral triangles, circles and squares) while Shape2 consists of images of non-canonical shapes (ellipsis, rectangles and triangles). Both Shape1 and Shape2 were generated with 28×28 pixels, with variation of colors: 0 to 7, position: left extreme to right extreme, rotation: 0 to 360^o.

Network Architecture: The SDA network had three hidden layers and one output layer with [576, 400, 256, c] neurons amounting to 784,384 connections. The induced random corruption levels for each of the three

[1] http://yann.lecun.com/exdb/mnist/
[2] http://datacenter.aucegypt.edu/shazeem/
[3] We acknowledge Microsoft Research India for Chars74k dataset.
[4] http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/BabyAIShapesDatasets

Table 1: Dataset characteristics with number of instances.

Data set		Labels		c	Instances		
Distribution		Ω			Train	Validation	Test
Latin	P_L	0-to-9	Ω_{09}	10	50,000	10,000	10,000
Arabic	P_A	•-to-9	$\Omega_{\bullet 9}$	10	50,000	10,000	10,000
Latin-2	P_{L2}	0-to-9	Ω_{09}	10	13,208	6,604	10,000
Digits	P_D	0-to-9	Ω_{09}	10	5,080	2,540	2,540
Lowercase	P_{LC}	a-to-z	Ω_{az}	26	13,208	6,604	6,604
Uppercase	P_{UC}	A-to-Z	Ω_{AZ}	26	13,208	6,604	6,604
Shape1	P_{Sh1}	'eqt','cir','sqr'	Ω_{sh1}	3	10,000	5,000	5,000
Shape2	P_{Sh2}	'tri','ell','rec'	Ω_{sh2}	3	10,000	5,000	5,000

hidden layers inputs are [10%, 20%, 30%] respectively. We used pre-training and fine-tuning learning rates of 0.001 and 0.1, respectively. The stopping criteria for pre-training was fixed to 40 epochs; stopping criteria for fine-tuning was set to a maximum of 1000 epochs with the validation dataset. Each of these experiments is repeated 10 times and performed student t-test with confidence interval of 0.05 to give some statistical significance.

We used Theano, a GPU compatible machine learning library to perform all our experiments on a i7-377 (3.50GHz), 16GB RAM and GTX 770 GPU processor.

3 Proposed approach and results

We propose a feature transference approach which enables deep neural networks to transfer hidden layers features (weights) of a classifier trained in unsupervised way [1]. For that purpose, SDA's are trained on a source problem and its features transferred to help in solving a target problem. We represent this transference by $\mathbf{w}_S \Rightarrow \mathbf{w}_T$. In this paper we explore feature transference in SDA at the pre-training stage, denoted $U(\mathbf{w})$, a process we call unsupervised feature transference (USDA) [1].

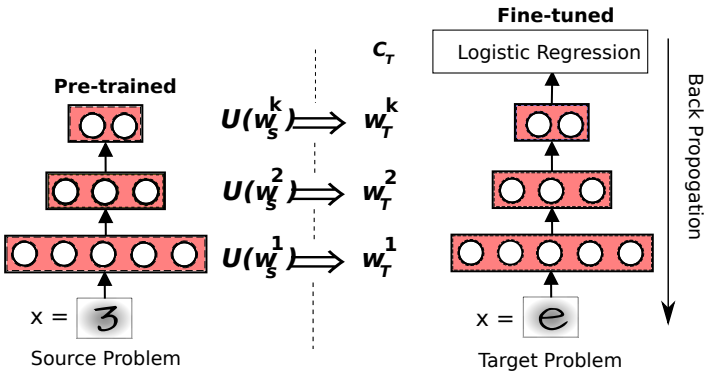

Figure 2: Unsupervised feature transference using SDA

In the USDA approach we transfer the unsupervised features of the SDA model from the source to the target problem, i.e., $U(\mathbf{w}_S) \Rightarrow \mathbf{w}_T$ as depicted in Fig.2. Once the features are transferred to the target problem, we add a logistic regression layer on top of the transferred machine. We fine-tune this entire classifier as a multi-layer perceptron using back-propagation. We compare the performance of USDA approach (TL) to the one obtained from a classifier built from scratch for the target problem, the baseline approach (BL).

3.1 TL: case 1

Classify letters reusing digits: We classify images of typed letters by reusing unsupervised features of a machine trained with handwritten digits from 0-to-9.

The performance is listed in Table 2. The average error rate of recognizing uppercase letters, 4.31±0.16% by reusing a machine pre-trained with Latin digits is significantly *lower* than baseline, 5.01±0.27%. Similar results are obtained from recognizing the lowercase letters. In both cases the significance level allows rejecting the null hypothesis of equal error rates.

It is interesting to note the difference between, reusing either Arabic or Latin dataset. The Latin dataset with smaller JSD value leads to better classification performance than the Arabic dataset. It would be interesting to study, does smaller JSD value leads to better TL even when the tasks are different.

Table 2: Average classification test error using SDA (%) ($\overline{\varepsilon}$) obtained for a target problem using UFT approach for different combinations of: target data distribution (P_T); target label set (Ω_T); source distribution (P_S); source label set (Ω_S); The difference between distributions is given by Jensen-Shannon divergence (JSD).

Label space		P_S	Ω_S	P_T	Ω_T	$\overline{\varepsilon}$		JSD
$Y_S \neq Y_T$	BL			P_{UC}	Ω_{AZ}	5.01±0.27		
	TL	P_L	Ω_{09}	P_{UC}	Ω_{AZ}	**4.31±0.16**	⇑	0.79
	TL	P_A	$\Omega_{\bullet 9}$	P_{UC}	Ω_{AZ}	4.41±0.22	⇑	0.99
	BL			P_{LC}	Ω_{az}	4.95±0.16		
	TL	P_L	Ω_{09}	P_{LC}	Ω_{az}	**4.37±0.13**	⇑	0.80
	TL	P_A	$\Omega_{\bullet 9}$	P_{LC}	Ω_{az}	4.43±0.11	⇑	0.99
$Y_S = Y_T$	BL			P_D	Ω_{09}	1.88±0.14		
	TL	P_L	Ω_{09}	P_D	Ω_{09}	1.78±0.21	○	0.88
	TL	P_A	$\Omega_{\bullet 9}$	P_D	Ω_{09}	**1.75±0.21**	○	0.99
	BL			P_{Sh1}	Ω_{sh1}	**7.88±0.93**		
	TL	P_{Sh2}	Ω_{sh2}	P_{Sh1}	Ω_{sh1}	7.96±0.93	○	0.99

⇑, ⇓, ○ statistically significant improvement or degradation or no change than baseline. The best $\overline{\varepsilon}$ obtained for a target dataset is marked in bold.

3.2 TL: case 2

Classify typed digits reusing handwritten digits: We classify images of typed digits by reusing unsupervised features of a machine trained with handwritten digits from 0-to-9. We observe slight performance improvement in the average error rate of recognizing typed digits.

Classify canonical shapes reusing geometrical shapes: We classify images of geometrical shapes (Shape2 dataset) reusing unsupervised features of a machine trained with canonical shapes (Shape1 dataset). We observe slight performance degradation in the average error rate of recognizing canonical shapes as listed in Table 2.

4 Conclusions

We analyzed the performance of USDA approach on two types of transfer learning approaches. TL case 1: when the source and target distributions and their labels are different. For example, the Jensen-Shannon distance for Arabic dataset with typed lowercase dataset is 0.99 and for Latin dataset with typed lowercase dataset is equal to 0.80, in both cases we observe significant improvement using transfer learning approach. It supports our hypothesis that, reusing the source dataset for the target dataset, even when the distributions and tasks are different, USDA approach helps improve classification performance of the target problem (and also around 50% reduction in computation time, see [1]) . Similar results were observed in case of uppercase dataset, this strengthens our previous argument.

On the other hand, TL case 2: when the source and target distributions are different yet the tasks are same, we observe slight improvement using transfer learning approach, i.e., reusing handwritten digits to classify typed digits.

Acknowledgment

The authors would like to thank Neural Network Interests Group of INEB for their valuable and constructive suggestions during the development of this research work. This work was financed by FEDER funds through the Programa Operacional Factores de Competitividade – COMPETE and by Portuguese funds through FCT – Fundação para a Ciência e a Tecnologia in the framework of the project PTDC/EIA-EIA/119004/2010.

References

[1] Chetak Kandaswamy, Luís M Silva, Luís A Alexandre, Ricardo Sousa, Jorge M Santos, and Joaquim Marques de Sá. Improving transfer learning accuracy by reusing stacked denoising autoencoders. *SMC conference, IEEE*, 2014.

[2] Pascal Vincent et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.