Activities report from March 2013 to March 2014

Chetak Kandaswamy

10th March 2014

Project “Reusable Deep Neural Networks: Applications to Biomedical Data”
(PDTC/EIA-EIA/119004/2010)

Instituto de Engenharia Biomédica (INEB)
Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

Contents

1 Introduction

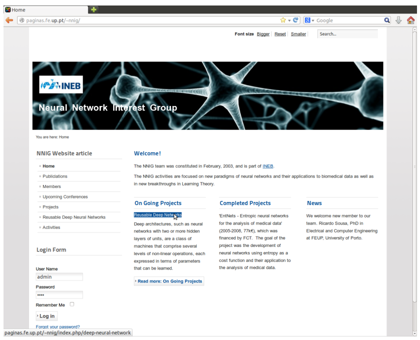
This report presents an overview of the work done during the research assistant grant started in March 15th within the project "Reusable Deep Neural Networks: applications to biomedical data" with reference PDTC/EIA-EIA/119004/2010. In the following sections I describe the work done, including an overview of the papers submitted to international conferences.

2 Creation and maintenance of the group's webpage

The webpage was created in Joomla version 1.0 open source content management system and hosted at FEUP server, gnome.fe.up.pt. The webpage host address is http://paginas.fe.up.pt/~nnig/. I along with Alexandre Castro managed MYSQL database for group website. The components of the Joomla 1.0 and its database was outdated leading to disconnection from the server. I migrated to Joomla version 2.5 and hosted at another FEUP server, paginas.fe.up.pt. The code's are managed in a open source centralized online repository Bitbucket with version control system.

The webpage site map:

- Homepage:
  - Publications
  - Members
  - Upcoming Conferences
  - Projects
    - On Going Projects
    - Completed Projects
  - Activities
- Reusable Deep Neural Networks
  - Team Members
  - Publications
  - Activities Reports
  - Upcoming Conferences
- Login Form



(a) Group Homepage

Figure 1: Group Homepage: http://paginas.fe.up.pt/~nnig/

3 Paper overview: Using Different Cost Functions to Train Stacked Auto-encoders

In this section present the overview of a paper [1] presented at the Mexican International Conference on Artificial Intelligence (MICAI 2013). In this work, we investigate the use of different cost functions by comparing the performances of squared errors (SSE), cross-entropy (CE), and exponential (EXP) costs when employed both in the unsupervised pre-training and in the supervised fine-tuning of deep networks whose hidden layers are regarded as a stack of auto-encoders. Using a number of artificial and real-world data sets, we compared pre-training and fine-tuning cost functions in terms of their impact on the reconstruction performance of hidden layers.

3.1 Experiments on CPU with MLPython library

The installation of MLPython library required Python, Numpy, SciPy, and MatPlotLib installations. The experiments were run on Eclipse/PyDev. The environment variables for PyDev was set via Window > Preferences > PyDev > Interpreter - Python > Environment > New. The MLPython v0.1 was used from http://www.dmi.usherb.ca/~larocheh/mlpython/ code and modified for the problem at hand.

3.2 Tuning Parameters for Deep Neural Networks

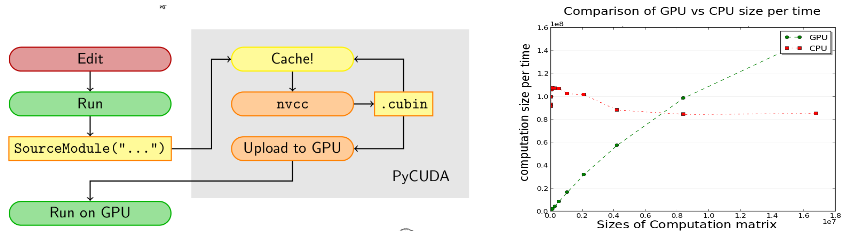
We performed a grid search for tuning hyper-parameters of a Stacked Denoising Autoencoder (SDA) for identifying best Cost function. The hyper-parameters such as number of hidden layers and size, pre-training and fine-tuning learning rate and tau (exponential cost function parameter). The results of SDA on various costs like SSE or MSE, CE, and EXP are presented in the technical report: Tunning Parameters of Deep Neural Network Algorithm for identifying best Cost function[2].

3.3 Conclusion

In general, the best layer-wise reconstruction performance was achieved by SSE pre-training, though with binary data CE yielded the lowest errors for the first hidden layer. Classification performance was found to vary little with the combination of pre-training and fine-tuning costs. When pre-training with CE, fine-tuning via SSE was found not to be a good choice. In general, the choice of the same pre-training and fine-tuning costs yielded classification errors with lower variance.

4 GPU Parallel Processing

Neural Network's with millions of neural connections training on large datasets with millions of instances would take several days on a machine using CPU. Generally, a GPU has 1000's of cores that can be processed in parallel computation, where as a CPU has 8 cores with serial computation. Using Theano a GPU parallel processing machine learning library allows faster training of large deep neural networks like Convolutional Neural Network's or Stacked denoising Autoencoder's. We have compared the performance of a CPU with i7-377 (3.50GHz) 16GB RAM Vs a GTX 770 GPU based on the size of the processed data. It is interesting to see that initially CPU performs well for small sizes of computations, but eventually GPU speeds up the process over several times as shown in below figure.



(a) A simple workflow of a GPU (b) Performance of GPU over CPU

Figure 2: Workflow and Performance comparision of GPU over CPU

4.1 Testing Symbolic Differentiation of Theano

In this section we present the migrate from MLPython to Theano and test the symbolic differentiation method on both the systems. The results are presented in the report: Testing Symbolic Differentiation of Theano With Denoising Autoencoder Model.

5 Paper overview: Improving Accuracy on Transductive Transfer Learning by Reusing SDA

In this section present the overview of a paper [3] submitted to International Joint Conference on Neural Networks conference (IJCNN 2014).

In this work we analyze feature transference using Stacked Denoising Autoencoders (SDA) for two different approaches: 1) unsupervised feature transference (USDA); 2) supervised layer based feature transference (SSDA). For that purpose we have carried out experiments to study the transductive transfer learning of arbitrary distribution of source and target problems for both USDA and SSDA approaches, for example, by training a machine to classify images of digits 0-to-9 and reusing these trained features to classify images of English characters a-to-z. We also performed experiments by reversing the problem roles: training a machine with images of English characters a-to-z and reusing the features to classify images of digits 0-to-9. Furthermore, we also studied inductive transfer learning of different but related problem for USDA approach.

We show that the unsupervised feature transference outperforms randomly initialized machine on a new problem. We achieved 7% relative improvement on average error rate and 50% on average computation time with uppercase letters dataset. In the case of supervised feature transference, we achieved 5.7% relative improvement for average error rate by reusing first or second hidden layer to classify the uppercase letters dataset, and 8.5% relative improvement for average error rate by reusing all three hidden layers of a problem that was fine-tuned again with the uppercase letters dataset.

6 Paper overview: Improving Deep Neural Network Performance by Reusing Features Trained with Transductive Transference

In this section present the overview of a paper [4] submitted to 24th International Conference on Artificial Neural Networks. (ICANN 2014). Training a machine to solve a specific problem and reuse this training with minor modifications to solve different but related problem. We propose a novel feature transference approach which enable deep neural networks to transfer either low-dimensional or high-dimensional features for a machine trained in either unsupervised or supervised fashion. Applying this feature transference approach on Convolutional Neural Network and Stacked Denoising Autoencoder on four different datasets, we achieve lower classification error rate with significant reduction in computation time with low-dimensional features trained in supervised fashion and high-dimensional features trained in unsupervised fashion for classifying images of uppercase and lowercase characters. The results are based from the technical report[5].

References

[1] T. Amaral, L. M. Silva, L. A. Alexandre, C. Kandaswamy, J. M. Santos, and J. Marques de Sa'. Using different cost functions to train stacked auto-encoders. In *Mexican International Conference on Artificial Intelligence (MICAI)*, pages 114–120, 2013. 3

[2] C Kandaswamy. Tunning parameters of deep neural network algorithm for identifying best cost function. Technical Report 2/2013, Instituto de Engenharia Biomedica / NNIG, April 2013. 4

[3] C. Kandaswamy, J. Marquesde Sa', L. M. Silva, L. A. Alexandre, and J. M. Santos. Improving accuracy on transductive transfer learning by reusing sda. *(Under Review)*, . 5

[4] C. Kandaswamy, L. M. Silva, L. A. Alexandre, J. Marques de Sa', and J. M. Santos. Improving deep neural network performance by reusing features trained with transductive transference. *(Under Review)*, . 6

[5] C. Kandaswamy, L. M. Silva, and L. A. Alexandre. Report: Improving cnn by reusing features trained with transductive transfer setting. Technical Report 2/2014, Instituto de Engenharia Biomedica / NNIG, Febyrary 2014. 6