An introduction to deep learning

Francis Quintal Lauzon
Laboratoire de vision et d'intelligence artificielle, École de Technologie Supérieure
francis.quintal-lauzon@polymtl.ca

Abstract

Deep learning allows automatically learning multiple levels of representations of the underlying distribution of the data to be modeled. In this work, a specific implementation called stacked denoising autoencoders is explored. We contribute by demonstrating that this kind of representation coupled to a SVM improves classification error on MNIST over the usual deep learning approach where a logistic regression layer is added to the stack of denoising autoencoders.

1. Introduction

Deep learning algorithms have shown superior learning and classification performance in areas such as transfer learning, speech and handwritten character recognition among others.

This paper introduces deep learning with focus on Stacked Denoising Autoencoders (SdA), applied to classification on the MNIST database [1]. Besides briefly introducing deep learning, the main contribution of this work is to suggest that the use of SVM as final classification layer can improve classification result over the usual logistic regression layer generally used with deep learning algorithms

Deep learning is about automatically learning multiple levels of representations of the underlying distribution of the data to be modeled [2]. In other words, a deep learning algorithm automatically extracts the low- and high-level features necessary for classification. By high level features, one means feature that hierarchically depends on other features. For instance, in the context of computer vision, this implies that a deep learning algorithm will learn its own low level representations from a raw image (such as edge detector, gabor filters, etc...), then build representations that depend on those low level representations (such as a linear or non-linear combinations of those low-level representations), and successively repeat the same process for higher levels.

Automatic representation learning is key point of interest of this kind of approach as the need for potentially time consuming handcrafted feature design is eliminated.

2. Theory

The training of a deep neural network for classification involves two steps, which are detailed in this section.

1. Unsupervised training of a SdA
2. Use the weights of the trained SdA to initialize a multilayer neural network and train with gradient descent.

According to a characterization made by Erhan *et al.* in [3], this two steps training process has a beneficial regularization effect in that it initialize the multi-layer neural network weights (of step 2) close to a local minimum of the loss function that will offer better generalization when compared to regular supervised training with random weights initialization.

UNSUPERVISED TRAINING

Autoencoders are neural networks trained to reproduce its input as accurately as possible. In order to do this well, the autoencoder must capture the important factors of variation of the data (in a way similar to what PCA does).

An autoencoder is made of a hidden layer, called the encoder and an output layer, called the decoder. Denoising Autoencoders (dA) are basically autoencoders for which noise is added to the training set so that the autoencoder must learn to reconstruct the uncorrupted input. This forces the encoder to learn robust representations that will usually generalize better than regular autoencoders [4]. The training process of a dA is done using standard gradient descent algorithm and is illustrated in Figure 1.

In the context of deep learning, the idea is to stack autoencoders one on top of each other so that the input of a given autoencoder is based on the output of the one below. This way, a hierarchical representation of the information is achieved.

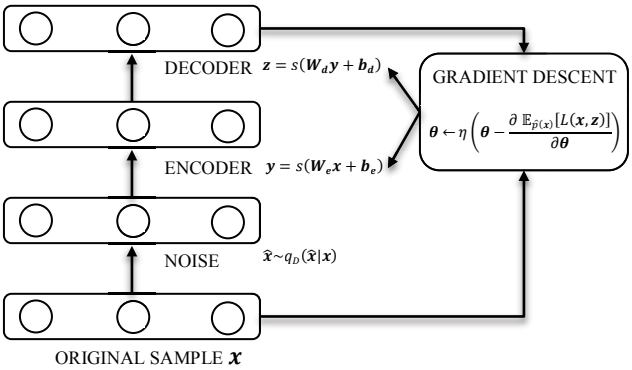

Figure 1 – dA training process with corrupted input. Note that the the difference between the first stage (input) and the second stage is only the addition of noise.

978-1-4673-0382-8/12/$31.00 ©2012 IEEE

The training of a SdA is done one encoder layer at a time. For the first encoder layer, a decoder (initialized with random weights) is attached to it and trained with gradient descent. When the training is completed, the decoder is replaced with a second encoder layer and a new decoder layer is added on top of that new encoder layer (in the same fashion it was done for the first layer). When training the second layer, the weights of the first layer are fixed so that only the second encoder layer and its decoder change. This process is then repeated for all SdA layers.

SUPERVISED TRAINING

At this point, the SdA cannot classify the data since it has not learnt to associate an input to a class. Rather, it only has the ability to reconstruct its input. In order to bridge this gap, the strategy is to add a final classification layer on top of the last encoding layer of the SdA and perform standard supervised multi-layer neural network training using gradient descent as illustrated in Figure 2.

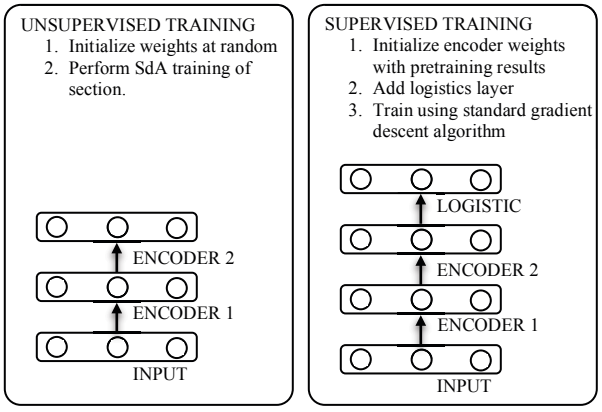

Figure 2 – Unsupervised and supervised training procedure procedure of a multi-layer neural network.

Once supervised training has been done, the network is ready for classification. Noting that the top layer of a multi-layer neural network acts as a linear separator, nothing forbids replacing it with a better performing classifier.

With this in mind, an experiment was run where a SVM layer replaces the top logistic regression layer after supervised training. Results are presented in the next section.

3. Results

The classification performance of a deep network initialized with a SdA is compared to popular classifiers, that is, the quadratic Bayes, k-NN and SVM. For the later, feature extraction using retina and dimensionality reduction using PCA was performed.

As for the deep networks, no pre-processing was applied, which is inline with the idea of automatic representation learning. Results were generated using a logistic regression layer and SVM layer as top classification layer.

The classification results are presented in Table 1. It is shown that deep networks show superior classification performance over the classifiers that use handcrafted features. Furthermore, the use of a SVM significantly improves classification error over a logistic regression.

Classifier comparison on MNIST

Classifier	Pre-processing	Error rate
Quadratic Bayes	Retina 10x10 + PCA	4.17%
k-NN	Retina 10x10 + PCA	3.37%
SVM	Retina 10x10 + PCA	1.94%
3 layers SdA + logistic regression	None	1.41%
3 layers SdA + SVM	None	1.16%

Table 1 - Comparaison between quadratic bayes, k-NN, SVM and SdA on MNIST.

4. Conclusion

In this work, deep learning along with the theory behind SdA were briefly introduced. Test result shows that a deep learning approach allows better classification than popular classifiers on the handcrafted features chosen in this work. This is a significant advantage over the typical classification approach that requires careful (and possibly time consuming) selection of features.

The fact that a SVM allows better classification than the standard logistic regression layer can be a useful tool for the deep learning practitioner. The natural next step to this work will be to test this approach to other datasets to investigate how general this result is.

5. References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the Ieee,* vol. 86, pp. 2278-2324, Nov 1998.

[2] Y. Bengio, "Deep Learning of Representations for Unsupervised and Transfer Learning," in *Proceedings of the Unsupervised and Transfer Learning challenge and workshop*, 2011.

[3] D. Erhan, Y. Bengio, A. Courville, P. A. Manzagol, P. Vincent, and S. Bengio, "Why Does Unsupervised Pre-training Help Deep Learning?," *Journal of Machine Learning Research,* vol. 11, pp. 625-660, Feb 2010.

[4] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research,* vol. 11, 2010.