Deep Neural Network Classifier of Handwritten digits

{ Mohammad Reza Sanatkar, Mengke Lian } Duke University - ECE Department

Objectives

This course project aims at dealing with hand writing recognition task. Deep neural networks have garnered a lot of attention in the field of machine learning in the recent years. In this project, we implement a multi-layer neural networks to perform the task of handwritten digits. Our project can be divided in the following main steps:

- Implementation of a general class of multi-layer neural networks in C++.
- Implementation of feed-forward and back-propagation algorithm to train such networks as discriminative classifiers.
- Implementation of learning algorithm for RBM to train neural networks as generative classifiers.

Introduction

Multi-layer neural networks represent the consistent effort of human beings to mimic the brain. Below, you see the neural networks which is used in this project. This network consists of three hidden layers.

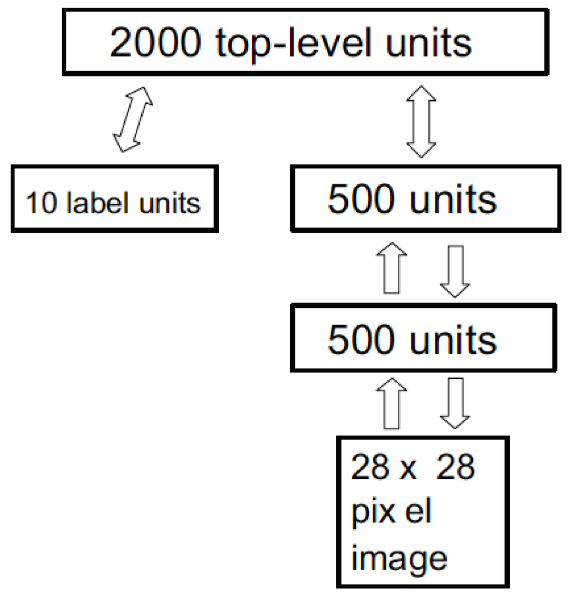

Figure 1: Neural Network implemented as a classifier of handwritten digits [1]

Restricted Boltzmann Machine (RBM)

Deep belief networks outperform the traditional neural networks because of taking benefits from RBMs. What is a RBM? RBM is a bipartite undirected inference graph which is used to learn a multivariate joint distribution satisfying the conditional independence imposed by the bipartite structure of the graph. In deep neural networks, every pair of successive layers is considered as a RBM. The first and most important step of training a deep neural network is to train these RBMs.

How to learn a RBM in a reasonable time : Contrastive Divergence Learning Procedure

$$\Delta w_{ij} = \epsilon(< v_i h_j >^0 - < v_i h_j^1)$$

- Start with available data on the visible units
- Compute the hidden nodes in parallel
- Compute the visible units (Reconstruction)
- Update the hidden nodes again

First, we train the RBM corresponding to the input layer and the first hidden layer. After learning the generative model for the first RBM, we use this set of learned weights to generate inputs for the second RBM. Having inputs for the second RBM, the set of weights of edges entering the nodes in the second hidden layer are learnt. These steps are repeated for all the other RBMS too.

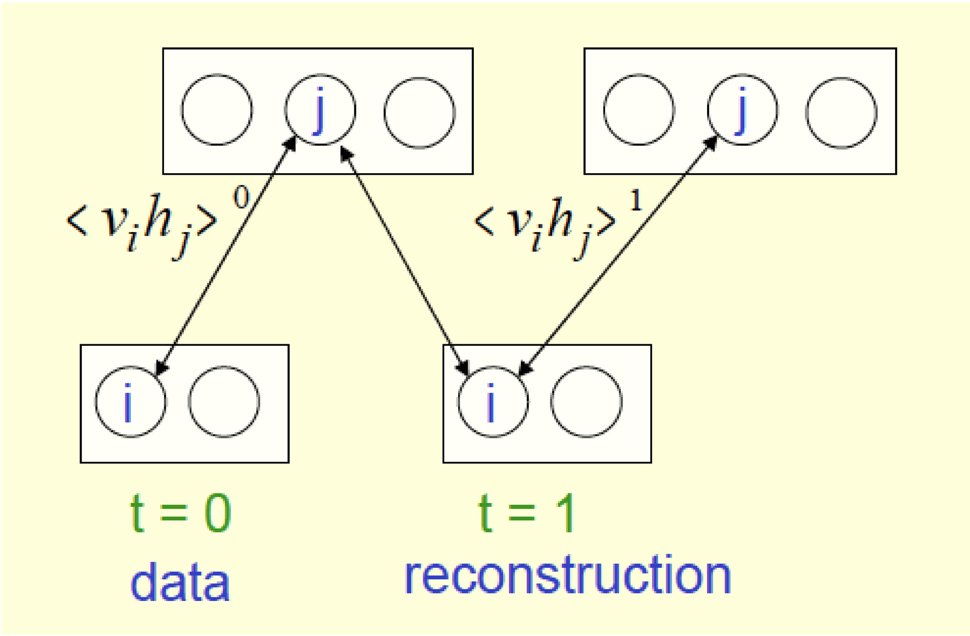

Figure 2: Contrastive Divergence Learning[1]

MNIST Database

The MNIST database of handwritten digits contains a training set of 60,000 examples, and a test set of 10,000 examples.

Each image is transformed from bilevel (black and white) image from NIST database by renormalizing to fit in a 20×20 pixel box while preserving aspect ration, which result into a 20×20 grey-scaled image. And then this 20×20 grey-scaled image is centered in a 28×28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28×28 field. The dataset consists of:

- 60000 training images of handwritten digits as well as their labels
- 10000 images of handwritten digits as test data

Back-Propagation

Back-propagation calculates the gradient of a loss function with respects to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function.

Denote the loss function as J, the basic iteration step in gradient descent is given by:

$$\Delta w = \ell \nabla J, \quad \Delta w_i = \ell \frac{\partial J}{\partial w_i}$$

For training images $\{(X_i, y_i)\}_{i=1}^N$, each time we use X_i as input and get the output $\hat{y}_i$ by feed-forward. Since we know the y_i, once $\hat{y}_i$ is known then J is determined and back-propagation is applied to update the weights.

Conclusion

Neural networks became popular in the community of compute scientists after derivation of back-propagation by Hinton in 1985. However, many researchers abandoned it because of emergence of SVM in 1990's. SVM worked better than trained neural networks by back-propagation. In 2002, Hinton came up with a fast method to train RBM neural networks which outperformed SVM classifiers.
We need to note that the significant increase in computational power of the computers in the recent years had a key role to make RBM neural networks practical. Training a deep neural networks as a generative classifier needs a huge amount of computations in parallel. Below, you can see the comparison between the RBM neural network and the other methods for handwritten digits recognition of MNIST dataset. the RBM deep neural network works better than other methods.

- Generative model based on RBMs 1.25%
- Support Vector Machine (Decoste et. al.) 1.4%
- Backprop with 1000 hiddens (Platt) 1.6%
- Backprop with 500 –>300 hiddens 1.6%
- K-Nearest Neighbor 3.3%

References

[1] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 2006.

Future Research

Even though employing multi-layer neural networks have improved the accuracy of object detection systems, the success rate of such systems is about 50 percent. This gives rise to the idea of a new class of neural networks with more similar structure to brain.

Contact Information

Email reza.sanatkar@duke.edu
Email mengke.lian@duke.edu