Training Deep Belief Network with Sparse Hidden Units

Zhen Hu, Wenzheng Hu, and Changshui Zhang

The State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList) and the Department of Automation, Tsinghua University, Beijing, China, 100084

Abstract. In this paper, we proposed a framework to train Restricted Boltzmann Machine (RBM) which is the basic block for Deep Belief Network (DBN). By introducing sparsity constraint to the Contrastive Divergence algorithm (CD algorithm), we trained RBMs with better performance than the off-the-shelf model in MNIST handwritten digit data set. The sparse model suffer from saturation slightly, however, by using a trade-off coefficient, the saturation problem can be solved well. To our knowledge, the sparsity constraint was first introduced to the hidden units of RBM.

Keywords: sparsity, DBN, mnist.

1 Introduction

It’s well-known that (sigmoid) Neural Network is highly non-convex and thus it’s easy to get stuck to a local optima when trained by the widely used Backpropagation algorithm (BP algorithm) [1]. The deeper the neural network is, the more concave and the apter to get stuck. In 1998 Y. Lecun et al. proposed the Convolutional Neural Network model [2] (CNN) and made training deep neural network possible. CNN models are fast because of the parameter-sharing strategy and hence decrease the number of parameters needed. Later in 2006, G. E. Hinton et. al proposed the pre-training strategy to accelerate the training process of deep neural network. Hinton et. al showed that by properly pre-trained, the deep neural network, which was a 9-layer model, converged quickly. And the pre-training strategy made training deep neural network into reality. Since then researchers have found that deep architecture was powerful in feature learning in many areas, such as image recognition proposed in [3], [4] and [5], speech recognition in [6] and [7] and music information retrieval in [8] and [9].

We proposed a new framework to pre-train Restricted Boltzmann Machine (RBM) which is the basic component of a kind of deep neural network named as Deep Belief Network (DBN). In our framework, we modified the contrastive divergence algorithm (CD algorithm) proposed by G. E. Hinton in [10] and [11]. The contrastive divergence algorithm, or CD algorithm, is an asymptotic algorithm to train RBM. As the complexity of the deep network raises the gap between the result of CD algorithm and the true model becomes a key problem.
We can run a longer Markov chain to solve the problem to some extent, but we have to spend more time. A large number of experiments showed that neural network in deep architecture was easy to be over-fitting. Some researchers argued that the expressive power of full-linked neural network can be limited by introducing some sort of penalty such as sparsity in [12]. So in our framework we introduced a penalty expression into the optimization formulation for the pre-training and proposed a new pre-training framework for RBM. We found that the penalized model outperformed the original model.

The following paper will be arranged as follows, in section 2 we review Restricted Boltzmann Machine to show how to restrict the hidden units. In section 3 we propose our framework and some necessary analysis. And in section 4 we will make a comment of sparsity and saturation. We show our experiments and results in section 5.

2 Restricted Boltzmann Machine

Restricted Boltzmann Machine is a kind of energy-based neural network as described in [13]. It contains two layers which are visible layer and hidden layer. All connections are inter-layer connection and no connection between two nodes from the same layer. See figure 1 for an example. RBMs are stacked to build DBN which is a deep neural network.

The energy function for RBM was defined as follows,

$$E(v, h) = -h^T W v - b^T v - C^T h.$$  

Where \( v \) denotes the value of all visible nodes, and \( h \) denotes value of all hidden nodes. The values of all nodes can be either 0 or 1. The probability of state of the network is then defined as follow,

$$P(v, h) = \frac{1}{Z} exp(-E(v, h)).$$

![Restricted Boltzmann Machine](image)

**Fig. 1.** An example for Restricted Boltzmann Machine