Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Neuron Learning Machine for Representation Learning

Jia Liu,[1] **Maoguo Gong,**[1] **Qiguang Miao**[2]
[1]Key Laboratory of Intelligent Perception and Image Understanding, Xidian University, Xi'an, China
[2]School of Computer Science and Technology, Xidian University, Xi'an, China
omegaliuj@gmail.com, gong@ieee.org, qgmiao@mail.xidian.edu.cn

Abstract

This paper presents a novel neuron learning machine (NLM) which can extract hierarchical features from data. We focus on the single-layer neural network architecture and propose to model the network based on the Hebbian learning rule. Hebbian learning rule describes how synaptic weight changes with the activations of presynaptic and postsynaptic neurons. We model the learning rule as the objective function by considering the simplicity of the network and stability of solutions. We make a hypothesis and introduce a correlation based constraint according to the hypothesis. We find that this biologically inspired model has the ability of learning useful features from the perspectives of retaining abstract information. NLM can also be stacked to learn hierarchical features and reformulated into convolutional version to extract features from 2-dimensional data.

Introduction

The learning behaviours of neurons have been researched for a long time for revealing the mechanism of human cognition. One of the most famous theory is the Hebbian learning rule (Hebb 1949) proposed by Donald Olding Hebb. Hebbian learning rule can be summarised by the most cited sentence in (Hebb 1949): "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased." By applying the Hebbian rule in the study of artificial neural networks, we can obtain powerful models of neural computation that might be close to the function of structures found in neural systems of many diverse species (Kuriscak et al. 2015). Recent works on Hebbian learning are the series on "biologically plausible backprop" (Scellier and Bengio 2016) by Bengio, et al. They linked Hebbian and other biologically formulated rules to back-propagation algorithm and used them to train the neural network. Hebbian rule gives the updating gradient of the connecting weights. We establish the model with the single layer network and attempt to model the updating gradient as the objective function. We call the model neuron learning machine (NLM). Hebbian rule itself is unsupervised and we find that NLM has many similar properties to unsupervised feature learning models. From the analysis and experiments, NLM is capable of learning useful features.

The Model

In the training of an artificial neural network, Hebbian rule describes the updating gradient of each connecting weight $W_{ij} = W_{ij} + \Delta W_{ij}$. This can be simplified by the product of the values of the units at the two ends of the connection, i.e., $\Delta W_{ij} = \alpha h_i v_j$, where α is the learning rate.

With the gradient, a model can be obtained by integrating over ΔW. However, it is of great difficulty to derive the integral. The artificial neural networks are highly simplified models inspired from real neural networks. Therefore when the Hebbian learning rule is applied to the simplified network, many considerations have to be considered, i.e., locality, cooperativity, weight boundedness, competition, long term stability, and weight decrease and increase (Kuriscak et al. 2015). For convenience, we assume that the connecting weight has negligible influence on the connected hidden unit. Under this omitting assumption, h can be taken as a set of constants wrt W and the integral is apparent, i.e., $\max \sum_{ij} \int_{W_{ij}} h_i v_j = \sum_{ij} W_{ij} h_i v_j = v^T W h$. Although this term is derived under the hypothesis, optimizing this term will increase the correlation between W_{ij} and $v_j h_i$ which is consistent with the Hebbian rule.

In order to make the network approximate the assumption and guarantee the long term stability of W and h, a correlation based constraint is introduced. The assumption omits the effect of W on the hidden units h. Therefore, both positive and negative correlations between W and h should be minimized which can be represented by the cosine between the two vectors $\cos^2(h, W_{:j})$ where $W_{:j}$ is the j-th row of the matrix W. Meanwhile, the length of W and h should be limited in order to prevent the infinite increase or decrease of W. Then the correlation constraint can be formulated as the square of inner product between the two vectors $\sum_j \langle h, W_{:j} \rangle^2 = \sum_j \|W_{:j}\|_2^2 \|h\|_2^2 \cos^2(h, W_{:j}) = \|Wh\|_2^2$. Then we can obtain the objective function of NLM:

$$\max J(W, b) = v^T W h - \lambda \|Wh\|_2^2 \tag{1}$$

where λ is a user defined parameter that controls the importance of the two terms.

In convolutional neural networks (CNN), the feature maps are obtained by $\boldsymbol{H}_i = s(I * \boldsymbol{W}_i + b_i)$ where $\boldsymbol{H}_i$ denotes the

i-th feature map and $\boldsymbol{W}_i$ denotes the i-th convolution kernel. The convolutional NLM (CNLM) can be obtained by summing over all the pixels in feature maps. Then the first term can be denoted by $\sum_{(x,y)} \sum_i (I * \boldsymbol{W}_i)(x,y)\boldsymbol{H}_i(x,y) = \sum_i tr(I * \boldsymbol{W}_i \boldsymbol{H}_i^T)$, where (x,y) denotes the pixel position in $\boldsymbol{H}$ and $tr()$ is the trace of a matrix. For the second term, the convolutional version is also easy to derive, i.e., $\sum_{(x,y)} \sum_{(i,j)} \langle \boldsymbol{W}(i,j), \boldsymbol{H}(x,y) \rangle^2$, where (i,j) denotes the position in convolution kernels. By combining the two terms, the CNLM is formulated. NLM and CNLM can be optimized by the widely used stochastic gradient descent algorithm.

Analysis

As described in (Vincent et al. 2010), one natural criterion that we may expect any good representation to meet, at least to some degree, is to retain a significant amount of information about the input. In NLM, maximizing Eq. (1) amounts to increasing the cosine correlation coefficient between v and Wh, i.e., $\cos(v, Wh) = \frac{v^T Wh}{\|v\|_2 \|Wh\|_2}$. Then the conditional probability $p(v|Wh)$ is enlarged. With a distribution of visible data $q(v)$, $\mathbb{E}_{q(v)}[\log p(v|Wh)]$ can be increased. As discussed in (Vincent et al. 2010), this amounts to increasing the lower bound on the mutual information between v and h. Consequently, the representation h learned by NLM can retain information of input data v.

Another criterion of a good representation is to eliminate irrelevant variabilities of the input data, i.e., abstraction and invariance (Bengio, Courville, and Vincent 2013). In NLM, for each visible unit v_j, the squared inner product between h and $W_{:j}$ is minimized which breaks the relationship between h and $v_j W_{:j}$. Then minimizing the constraint term amounts to minimizing the summed conditional probability $\sum_j p(h|v_j; W_{:j})$. Meanwhile, maximizing the first term will increase the correlation between $v^T W$ and h which increases the conditional probability $p(h|v; W)$. This means that h responds to most relevant components which omitting the relatively irrelevant components in v.

Experimental Study

In NLM, we claim that h responds to most relevant components which omitting the relatively irrelevant components in v. Then we select two images from the MNIST dataset and CIFAR-10 dataset respectively, and plot $\cos^2(h, W_{:j})$ for each j as the response to each v_j during the iterations. Figure 1 shows such plots of the initialization, 10th iteration (N=10), 50th iteration (N=50) and the last iteration (N=100), respectively. Since v is 2 dimensional data, the plots are exhibited in the 3 dimensional form with the heave representing the response to each pixel in v.

Then we apply NLM to learn hierarchical representations. Two NLMs are stacked with the hidden units being 500 and 300 respectively. We set $\lambda = 1$ and the learning rate to be 0.01. The compared models, i.e., autoencoder (AE), restricted Boltzmann machine (RBM), SR-RBM, and MO-SFL, are also stacked with the same architecture. SR-RBM and MO-SFL are recently proposed sparse versions of RBM and AE respectively. The models are trained by the 60000 training images in the MNIST dataset. The learned hierarchical features are fed into a linear classifier and we use randomly sampled 10000 images and 60000 images to train the classifier. The test error rates are listed in Table 1. For efficiently learning features from images, we use the architecture of CNN and learn the features by CNLM. CNLM is implemented on the MNIST and CIFAR-10 datasets. The baselines are LeNet and CIFAR-10 Quick respectively. The learned features are classified by SVM. We compare CNLM with the convolutional versions of AE and RBM . The test error rates are listed in Table 1.

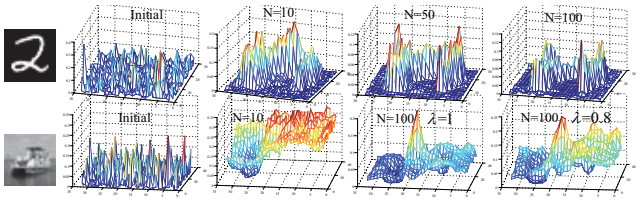

Figure 1: 3 dimension plots of $\cos^2(h, W_{:j})$ on each pixel of the input image. Two images from the MNIST dataset and the CIFAR-10 dataset respectively are exhibited. With the increase of iterations, i.e., N=0, 10, 50, and 100, the irrelevant pixels are restrained. With different λ, the response intensity of each pixel is different.

Table 1: Classification results of deep networks including stacked and convolutional version of different feature learning models. In the experiments on convolutional versions, the CIFAR-10 dataset is also used.

	stacked		convolutional	
models	10000	60000	version	CIFAR-10
AE	3.02	1.63	0.76	21.8
RBM	3.06	1.68	0.83	25.2
SR-RBM	2.97	1.61	–	–
MO-SFL	2.91	1.57	–	–
NLM	**2.89**	**1.53**	**0.67**	**21.1**

References

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE Trans. PAMI* 35(8):1798–1828.

Hebb, D. O. 1949. *In The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.

Kuriscak, E.; Marsalek, P.; Stroffek, J.; and Toth, P. G. 2015. Biological context of Hebb learning in artificial neural networks, a review. *Neurocomputing* 152:27–35.

Scellier, B., and Bengio, Y. 2016. Towards a biologically plausible backprop. *arXiv:1602.05179*.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR* 11:3371–3408.