Research of The Deeper Neural Networks

You Rong XIAO[1], Qing Xiu WU[1,a], Shu Qing LI[1] and Jun OU[1]

[1] *Hainan College of Software Technology, QiongHai, Hainan, 571400, China*

Abstract. Neural networks (NNs) have powerful computational abilities and could be used in a variety of applications; however, training these networks is still a difficult problem. With different network structures, many neural models have been constructed. In this report, a deeper neural networks (DNNs) architecture is proposed. The training algorithm of deeper neural network insides searching the global optimal point in the actual error surface. Before the training algorithm is designed, the error surface of the deeper neural network is analyzed from simple to complicated, and the features of the error surface is obtained. Based on these characters, the initialization method and training algorithm of DNNs is designed. For the initialization, a block-uniform design method is proposed which separates the error surface into some blocks and finds the optimal block using the uniform design method. For the training algorithm, the improved gradient-descent method is proposed which adds a penalty term into the cost function of the old gradient descent method. This algorithm makes the network have a great approximating ability and keeps the network state stable. All of these improve the practicality of the neural network.

1 Introduction

Neural networks are based on Multilayer Perceptron. Hubel and Wiesel [1] have researched the cat's visual cortex. It is known that the complex cells belong to visual cortex. The cells are apt to catch the little sub-regions of the visual object. The captured field is named with receptive field. In spatial filed, the characters of the sub-regions are correlated with the whole visual object. Also, the cells is divided into two cells: the simple cells and the complex cells. The former is sensitive to the edge of the receptive field; the latter covers the most of receptive fields. The receptive field cells produce the strongest response to the exact position of the visual object, but are unresponsive to the outside of receptive field.

2 Related work

2.1 Neural networks

The function can be represented in the form of a network diagram as shown in Figure 1. The process of evaluating can then be interpreted as followed.

As illustrated in Figure 1, input layer is consisting of vector x=$[x_0......x_n]^T$. Layer n computes an output vector a^n using the output a^{n-1} of the previous layer, with parameters b^n (a vector of offsets) and θ^n (a matrix of weights). The parameters θ^n as weights and the parameters b^n as biases. The bias parameter allows for any fixed offset in the data. The quantities a^n are known as activations. The formula is

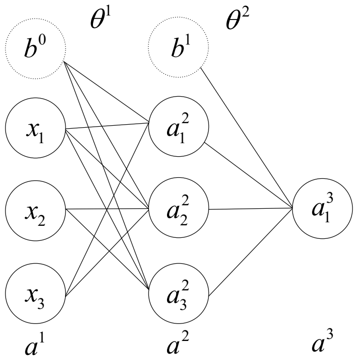

Figure1. Network diagram for neural network.

$$a^n = \text{sigmoid}(b^n + \theta^n a^{n-1}). \quad (1)$$

The logistic sigmoid function defined by

$$\sigma(a) = \frac{1}{1+\exp(-a)} \cdot \quad (2)$$

It can be replaced by tanh(a)=2$\sigma(a)$-1, and so a general linear combination of logistic sigmoid function is equivalent to a general linear combination of 'tanh' functions. The activation function is used to emulate the frequency of the impulses sent to another neuron (or group of neurons) or sampled by the external environment. The sigmoid function $\sigma(a)$ has a real-valued input (the signal's final sum) and squashes it to range from value=0 to value=1. Linear functions are that the dependent variable (y) changes with independent

[a] Corresponding author: xhogh@hotmail.com

variable (x); otherwise, nonlinear functions mean that the dependent variable (y) is not a corresponding fixed change with the variable (x).

2.2 Convolutional neural networks

Convolutional neural networks contain convolutional layers, pooling layers, fully-connected layers. The features of one part are densely correlated with the other part in the input image. Learned from one part of the image [2], the features of one part of the image can applied to the other parts of the image. For example, there is a 5×5 input image. 3×3 patches represent the learned features which have been sampled randomly from the image. 3×3 patches slide over the large image. The new feature image can be produced.

The first value of the final matrix is computed as followed: [4]=1×1+1×0+1×1+0×0+1×1+1×0+0×1+0×0+ 1×1. The other values can be calculated by the same method. Then, the different feature activation value will be obtained in the image.

Pooling layers are usually used after the convolutional layers. The layers simplify the number of neurons. It can obtain the condensed feature map from the convolutional layer. As a concrete example, max-pooling always is exploited. A pooling unit outputs the maximum activation in 2×2 input region. Obviously, 14×14 neurons outputs from the pooling layers while 28×28 neurons as input from the convolutional layers. The convolutional layers always produce many feature maps. Max-pooling is applied to the feature maps individually [3].

Fully-connected layers are the final layer of connections in the network. In one word, the layers connects every neuron from the max-pooled layers to every one of the output neurons.

Given N×N neuron layer, (N-k+1) × (N-k+1) neurons will output from the convolutional layer with a filter of k×k size. Suppose that a standard two-layer network of the convolutional net. The weight parameters in the first layer of the network are shared between the various outputs. The first layer of the network can be thought as performing a nonlinear feature extraction, and the sharing of features between the different outputs can save on computation and can also lead to improved generalization.

2.3 Deeper neural networks

In this report, the architectures of deeper neural networks will be designed. Its structure splits the two hidden layers. One of the hidden layers is a feed-back hidden layer and the other one is a self-organizing hidden layer. With many 2×2 blocks, the feed-back hidden layer gives the network a weak stability and wealthy dynamics. The self-organizing hidden layer can transform the information from different blocks, improve the approximating ability and organize the different dynamic subsystems to work. Considering the features of the deeper neural networks, the dynamic subsystem of the network is defined in this report. The network adjusts the structure by adding the subsystems. Additionally, this report also proposed another self-organizing method to adjust the structure by activating the different subsystems when the network is used to approximate different system. These make the network can adjust its structure depending on the task.

Gradient descent is local-optimization algorithm. Function G(x) decreases fastest if goes along the negative gradient direction. By the method, G(x) obtains the min or max value. A modified gradient-based learning algorithm is proposed for the two- layer interval perceptron. In the modified learning algorithm, for the radius of each interval weight, the function of the absolute value in the original learning algorithm is replaced by a quadratic term. This modified learning algorithm, compared with original learning algorithm, can prevent the weights oscillation during the learning procedure. The monotonicity of the error function and convergence for the modified learning algorithm are proved [4].

A deeper neural networks model (DNNs) is proposed, which could recognize images from multiple problem fields. Using a neural network based dimension reduction technique, low-dimensional embedding for arbitrary reorder features can be learned; through a multi-layer network, these feature embedding are integrated with word embedding features into a linear ordering reorder models. According to the truth that DNNs captures image features level-by-level, a special method is used to combine DNNs of multiple fields. The bottom level of DNNs are shared to capture the low-level features in data space, while the high level neurons are still detached to capture the high-level features of the specific fields. Besides capturing low-level features, shared neurons also lead input data into correct high level neurons, which will finally make the reorganization in specific fields. This cross-field DNNs is trying to simulate human visual system, which has bionic meanings. Meanwhile, cross-field unlabeled sample sets could be used to train this model due to the shared neurons, which could improve the recognition performance in current internet environment with large scale cross-field unlabeled images. Because of the detached neurons, paralleled computing is possible, which could reduce the time cost of training. The DNNs is designed and realized with MATLAB library. Finally it is tested in MNIST data set of handwriting characters and COIL data set of object images. Experimental shows that the DNNs is more efficient than the traditional neural network and convolutional neural networks. In a deep neural network, the supervised training of bottom neurons could be replaced by unsupervised training. According to this, some improvements are applied on DNNs training algorithm, which made the paralleled computing of DNNs easier. The DNNs model's parallelization can be realized by (Compute Unified Device Architecture) CUDA framework.

2.4 Model test

The deeper neural networks (DNNs) can be constructed by MATLAB or C++. A graphics processing unit (GPU) can accelerate MATLAB function. MATLAB Executable (MEX) is intended to directly use C/C++ and FORTRAN codes within the MATLAB environment to accomplish higher executing speed and avoid application bottlenecks. CMEX is called for the C/C++ MEX, and C-MEX is only for the purpose of deploying a GPU device. Since c-mex requires building C/C++ executable and CUDA requires hardware-specific (NVIDIA GPU) codes, extra installation steps is needed in addition to a standard MATLAB installation. Table 1 shows the image recognition result using the DNNs built.

Table 1. The image recognition result using the DNNs.

	Times(s)	Error rate	Epoch
DDNS	2.4295	0.068	1
NNS	0.8518	0.2367	5

From the above compared data, the running time of DNNs is longer than normal neural networks. But Full-batch train mean squared error on training set of the former is better than the latter. In short, the DNNs run well than the normal neural network.

2.5 Applications of model

2.5.1 Online course quality evaluation

Based on neural networks model, the thesis puts forward the "Resource Strategies", "Teaching Strategies" for improving online course perceived service quality with the calculation and analysis of the index system. The evaluation process is shown in Figure 2.

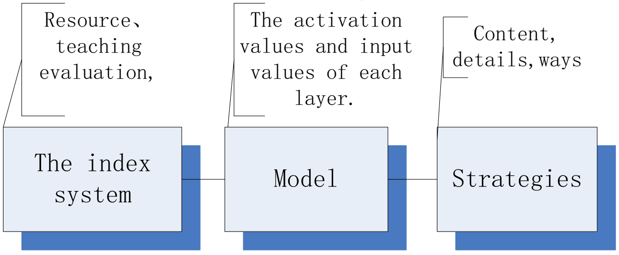
Resource、 teaching evaluation,
The activation values and input values of each layer.
Content, details, ways
The index system
Model
Strategies

Figure 2. Online Course Quality Evaluation.

The index system includes the evaluation of learning resource (content resources, positioning, easy or difficult), learning process (the actual operation, communication and interaction), and personnel. It can analyze the rules and trend of the learners visiting the website, then find out the problems in the learing course, and provides some strategies. The strategies can strengthen the quality of the content resources, improve the user operation,and provide clearly detailed study guide to enhance interaction. It can modify and provide a reliable basis for making the adjustment curriculum of the network, in order to improve the credibility of the learners.

2.5.2 Water and soil data analysis

Water and soil data can be inputted the model by the sensors. The model can put forward water diagnosis and soil conditions. In the conclusions, water quality belongs to which kind of water. The proportion of salt and water can be obtained in the soil tested. The process is shown in Figure 3.

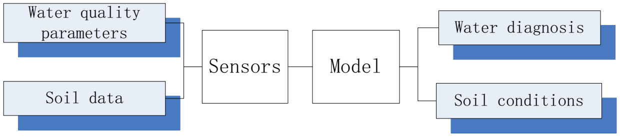

Figure 3. Water and Soil Data Analysis.

Water parameters include chemical indicators: the first kind is organic matter and the other is inorganic index.The former includes biochemical oxygen demand(BOD), chemical oxygen demand (COD), total organic carbon (TOC), and total oxygen content (TOD). The latter includes plant nutrient elements that is N element and P element of wastewater, PH which indicates the PH of the water samples, and heavy metal. Heavy metals mainly refer to the elements of mercury, cadmium, lead, chromium, nickel, and arsenic and other biological toxic elements. Soil parameters include soil moisture, temperature, PH, and salt content. According to the data, the soil tested will be irrigated or carrying out soil fertilization. The measures would enhance the soil properties.

3 Conclusion

Based on the study of neural network, a deeper neural network architecture has been designed. The method can be applied to image auto-annotation [5][6]. It can be an efficient method to solve semantic gap problems. Image auto-annotation can decomposed into two steps, that is, the basic image auto-annotation based on the relationship between image and tag, and the annotation enhanced based on the mutual information of the tags. The basic image auto-annotation is viewed as a multi-labelled problem. Therefore the prior knowledge of the tags, the dependent relationship of the tags is used to improve the annotation result. The experiments prove that the network has a great approximating ability and a weakly stable condition. It can efficiently solve the image recognition and labeling.

Acknowledgment

This work is supported by the Natural Sciences Foundation of Hainan Province under Grant No. 20156240, No.614238, No.20161009 and No.613167. The work is also supported by the Scientific Research Project of Hainan Province Colleges under Grant No.Hnky2015-79. *Corresponding author at: Department of Network Engineering, Hainan College of Software Technology; QiongHai. Hainan Province, 571400, P.R.China; E-mail address: xhogh@hotmail.com (Qx Wu).

References

1. Hubel, D. and Wiesel, T. (1968). Receptive fields and functional architecture of monkey striate cortex. Journal of Physiology (London), 195, 215–243.

2. http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/.
3. H Lee, R. Grosse, R. Ranganath, and A.Y. Ng, "Convolutional deep belief networks for scalable unsupervise learning of hierarchical representations.", ICML 2009.
4. H Lee, C. Ekanadham, and A.Y. Ng., Sparse deep belief net model for visual area V2, in Advances in Neural Information Processing Systems (NIPS) 20, 2008.
5. Y.Bengio, X. Glorot, Understanding the difficulty of training deep feedforward neuralnetworks, AISTATS 2010.
6. Socher, R., Huval, B., Bhat, B., Manning, C. D., and Ng,A. Y. Convolutional-recursive deep learning for 3d object classification. In Advances in Neural Information Processing Systems (NIPS). 2012.