Simulation Research on Influence of Neural Networks Scale on its Performance

Xilei Wu
College of Computer Science and Electronic Engineering, Hunan University, Tianma Student Apartment, Juzizhou Street, Yuelu District, Changsha, Hunan.
Email: angela@cas-harbour.org

Abstract: When using deep learning methods to deal with specific problems, the structure and configuration complexity of neural networks model have a great impact on the performance of the system. This paper constructed a multi-layer perceptual neural networks model basing on classification problem of film and television reviews. The paper also analyzed the forward and back propagation, processing algorithm and key parameter optimization method, and built the simulation environment. By adjusting the network depth and the number of neurons in each layer, the paper analyzed the influence of the complexity of the neural networks model on the network performance. Then the paper gave the analysis results under the condition of different model scale. Finally the paper purported conclusion in the last section that the performance of the neural networks model changes greatly with number of neurons in each layer changed, but the performance changes little with network depth changed. It can provide an important reference for two classification problems in different fields.

1. Introduction
Nowadays, the deep learning method based on artificial neural networks develops rapidly and is widely applied[1]. When using deep learning methods to solve binary classification problems such as comments classification of film, television program, literature as well as art, and recognition of emotions and images[2]. The structure and configuration complexity of neural networks model have a great impact on the performance of the system. Many articles focus on the improvement of neural networks forward and back propagation algorithm on the basis of deep learning[3], or focus on application studies in specific fields such as sea load data processing[4] and etc. However, there is little quantitative analysis of the impact of network complexity on system performance. Considering the influence of model structure and scale on model performance, it is generally believed that multi-layer perception can have better performance with the same input [5]. In this paper, through the construction of multi-layer perceptual neural networks model and the establishment of simulation environment, by adjusting the depth of the networks and the number of neurons in each layer, we quantitatively analyze the impact of neural networks complexity on system performance. The models, simulation analysis methods and conclusions in this paper bring convenience to those who seek to find proper neural networks scale for their dichotomous or multiple classification applications and more importantly have a wide range of reference and application value in solving various binary classification and even multi-classification problems.

2. Principle Analysis of Model
Artificial Neural Networks (ANNs) is also called connection model. It is an algorithm mathematical
model of distributed parallel information processing [6]. A typical neural networks consists of several neurons distributed in several layers and connections with different weights between adjacent layers. Figure 1 shows a multi-layer perception neural networks model which is also the theoretical analysis model of our research target.

Figure 1.A typical model of multi-layer perceptron neural networks.

The model consists of n layers in total, including an input layer, an output layer and n-2 middle layers which are also called hidden layers. \( m_p \) ( \( p = 1, 2, 3, \ldots, n \) ) represents number of neurons in the \( p \)th layer \( L_p \). Vector \( x = (x_1, x_2, x_3, \ldots, x_{m_1}) \) is the input of the neural networks, and vector \( y = (y_1, y_2, y_3, \ldots, y_{m_n}) \) is the output of the neural networks.

The process of neural network optimization is divided into two stages: forward propagation and back propagation. In the first stage, the predicted value is calculated by the forward propagation algorithm, and the error between the predicted value and the real value is obtained. In the second stage, the cost function and the gradient of each parameter that needs to be adjusted (such as weight \( w \) and bias \( b \)) are calculated by back propagation. And then the weight and bias of each connection are updated according to the gradient and learning rate by using the gradient descent algorithm.

Figure 2 represents the model of transmission and processing procedure of a certain neuron in forward propagation, which reflects the relationship among neuron input, weight connection, processing and neuron output.

In the model shown in figure 2, all the weights \( w_{lk}^{(p)} \) ( \( k = 1, 2, 3, \ldots, m_{p-1} \) ) are randomly distributed. With input vector \( a^{(p-1)} = (a_1^{(p-1)}, a_2^{(p-1)}, \ldots, a_{m_{p-1}}^{(p-1)})^T \), the status \( z_i^{(p)} \) and output \( a_i^{(p)} \) of the \( i \)th neuron in the \( p \)th layer are respectively represented in formula (1) and (2).

\[
z_i^{(p)} = \sum_{j=1}^{m_{p-1}} w_{ij}^{(p)} a_j^{(p-1)} + b_i^{(p)}
\]

(1)

\[
a_i^{(p)} = f(z_i^{(p)}) = f \left[ \sum_{j=1}^{m_{p-1}} w_{ij}^{(p)} a_j^{(p-1)} + b_i^{(p)} \right]
\]

(2)
Figure 2. A neuron in the forward propagation.

In formula (1), $w_{ij}^{(p)}$ represents the weight of connection between the $j^{th}$ neuron in the $(p-1)^{th}$ layer and the $i^{th}$ neuron in the $p^{th}$ layer; $b_i^{(p)}$ represents the input bias from the $i^{th}$ neuron in the $(p-1)^{th}$ layer to the $i^{th}$ neuron in the $p^{th}$ layer; $z_i^{(p)}$ represents the status of the $i^{th}$ neuron in the $p^{th}$ layer; $a_i^{(p)}$ represents the input (or say activation) of the $i^{th}$ neuron in the $p^{th}$ layer; $f(\cdot)$ is the activation function as which we often use sigmoid, ReLU and so on. In figure 2, when $L_{p-1}$ is the first layer (input layer), input vector is $\mathbf{a}^{(1)} = \mathbf{x} = (x_1, x_2, \ldots, x_m)^T$.

Figure 3. A neuron in the back propagation.

Figure 3 shows the model of transmission and processing procedure of a certain neuron in the error back propagation. In figure 3, the output $\delta_i^{(p)}$ can be expressed as formula (3).

$$
\delta_i^{(p)} = \sum_{j=1}^{m_{p+1}} \delta_j^{(p+1)} W_{ji}^{(p+1)} \cdot f'(z_i^{(p)})
$$

(3)

In the formula (3), we use $\delta^{(p+1)}$ in layer $L_{p+1}$ to calculate $\delta^{(p)}$ in layer $L_p$. Then, through training
from input layer to output layer and iteration of error back propagation, the parameters of neural networks are finally optimized. In back propagation, cost function reflects the difference between the expected value and actual value. And in order to optimize the performance of the neural networks model, making the actual value close to expected value, we should keep the cost function as small as possible.

The total average cost function of the network is expressed as formula (4)

$$E = \frac{1}{N} \sum_{k=1}^{N} E(k)$$

In the formula (4), $E(k) = \frac{1}{2} \sum_{i=1}^{m} [y_i(k) - a_i^{(n)}]^2$ is the cost function of the network under the kth training dataset.

Then according to gradient descent algorithm, calculation and optimization of weight $w$ and bias $b$ are expressed as formula (5) and (6).

$$\frac{\partial E}{\partial w_{ij}^{(p)}} = \delta_i^{(p-1)}$$

$$\frac{\partial E}{\partial b_i^{(p)}} = \delta_i^{(p)}$$

3. Simulation and Analysis

3.1. Simulation Process and Related Parameters

In this section, the paper describes the simulation preparation process and how the model of neural networks is constructed. A multi-layer perceptual neural network model is constructed based on the classification problem of film and television reviews, and relative model parameters are given. sentiment text classification of film comments is a dichotomous problem with only positive and negative target categories [7]. The working process of the neural network is given in figure 4.
In figure 4, IMDB (Internet Movie Database) contains several film comments data in the Keras library. In the model, select 10000 high-frequency English words for simplifying data processing, and vectorize the data of film comments to fit the requirement of neural networks input. Then use K-fold validation to set part of the data to validation data. The dataset have enough training data (25000 film comments), in order that simply dividing the data into training data and validation data. What to do next is build a model of neural networks for training data. Keras has highly modular and packaged model function, so it can just import model from Keras and add layers to it. Relative parameters of the model are presented in table 1 below.

| Dataset                          | Activator of hidden layers | Activator of output layer | Optimizer    | Loss function       | Metrics (parameters to monitor) | Number of training epochs |
|----------------------------------|-----------------------------|---------------------------|--------------|---------------------|---------------------------------|---------------------------|
| IMDB dataset in Keras Library    | ReLU (Rectified Linear Units) | Sigmoid                  | RMSprop (Root Mean Square Prop) | Binary crossentropy | Accuracy & Loss                     | 20                        |

In the model, the number of neurons in each layer is changed firstly and then the number of layers in the model. Then the author builds 3 models with same parameters in the table 1 and with number of neurons or number of layers changed. Table 2 and Table 3 show experimental variables settings in part of the evaluations.

| Model                | First evaluation | Second evaluation | Third evaluation |
|----------------------|------------------|-------------------|------------------|
| 1st model (model 0)  | 2                | 4                 | 4                |
| 2nd model (model 1)  | 2                | 8                 | 16               |
| 3rd model (model 2)  | 2                | 256               | 1024             |

| Model                | First evaluation | Second evaluation | Third evaluation |
|----------------------|------------------|-------------------|------------------|
| 1st model (model 0)  | 16               | 1                 | 2                |
| 2nd model (model 1)  | 16               | 2                 | 3                |
| 3rd model (model 2)  | 16               | 3                 | 5                |

Finally, results of accuracy and loss which are the metrics to evaluate performance of the model are plot in curve graphs. In order to have a better view of the comparison of the results, in one evaluation the accuracy and loss of these 3 models are plot in the same curve graph simultaneously, which will be presented in the next section. The specific outcomes of sentiment judgment of each film comment will not be shown in the results displayed in the next section because they are not in the scope of this paper.
3.2. Results Analysis
In this section, the paper gives part of the experiment results and results analysis. First, the number of neurons in each layer is changed and the number of layers remains the same. Figure 5 - 7 respectively show the accuracy and loss results of the 1st, 2nd and 3rd evaluation.

Figure 5. Result of the 1st evaluation with 4, 8, 256 neurons in each layer of model 0, 1, 2 respectively.

Figure 6. Result of the 2nd evaluation with 4, 16, 1024 neurons in each layer of model 0, 1, 2 respectively.

Figure 7. Result of the 3rd evaluation with 4, 16, 4096 neurons in each layer of model 0, 1, 2 respectively.
As shown in figure 5 - 7, a clear comparison of accuracy and loss of each model is displayed. Generally speaking, when dealing with a dichotomous problem, first, the more neurons there are, the larger the capacity of the neural networks will be, and the loss function of training data rapidly reaches the minimum value with the accuracy rate of training data quickly going to the top. When there are relatively less neurons in each layer, the increasing rate of accuracy and the declining rate of loss function get smaller, requiring more epochs of training to have an ideal training performance. Second, in terms of validation data, when there are more neurons in each layer, the loss function soon drop to the minimum and over-fit occurs. After over-fit, loss function rise sharply and the gradually flatten. Accuracy of validation data can just be improve to a limited extent ( about 87% ) and it raises fast to the maximum and then remains nearly invariable with bigger capacity of neural networks. Besides, even though not presented in the paper, when adding excessive neurons in each layer, such as 8096 neurons, the model loses the ability to learn from the training data. It also occurs when adding too little neurons to each layer.

Then, the number of layers in each model is changed while number of neurons remains the same in each layer. Accuracy and loss results of the 1st, 2nd and 3rd evaluation are presented in figure 8 - 10 respectively.
Figure 10. Result of the 3rd evaluation with 2,3,9 layers in model 0, 1, 2 respectively.

When number of layer increases, training accuracy and loss function curve are nearly the same, even if number of neurons is set to a bigger value like 256 or 512. In terms of the validation data, then loss function deteriorate sooner with more layers, while accuracy change little. Compared to the change of number of neurons in each layer, the change of number of layers has less effect on validation performance. What’s more, similarly, model is unable to be trained if number of layers is excessive, such as 30 layers in a model.

4. Conclusions
The paper assesses the performance of neural networks which deals with a dichotomous problem by modeling training and validation process, and part of the evaluation results are presented. According to the results above, with bigger number of neurons in each layer, the validation loss function deteriorates more quickly and seriously. However, what is noticeable is that the general validation accuracy in each experiment set is higher than the other two models of smaller scale. As a consequence, the author thinks that the common statement that over-fit will leads to bad performance of neural networks is not precise. Moreover, the same experiment on another multi-classification problem (handwritten digits recognition) proves that it is very important to select the appropriate size of neural networks in accordance with different data scale to improve model performance. The above process has shown that changing the number of neurons and changing the number of neural networks layers have different effect on model performance, which may provide a reference for selecting different neural network structures for different data. Besides, more research on other kind of problems have to be carried out to further validate the conclusion that over-fitting does not necessarily lead to performance degradation.

Reference
[1] Ma,S.L., Li,X.P., (2016) Overview of big data and deep learning [J]. Journal of intelligent systems, 11(6):728-741.
[2] He,K., Zhang,X., Ren,S., et al. (2016) Identity Mappings in Deep Residual Networks[C]. European Conference on Computer Vision. Springer,Cham., 630-645.
[3] Kobayashi,N., Inui,K., Matsumoto,Y., et al. (2004) Collecting Evaluative Expressions for Opinion Extraction[J].Journal of Natural Language Processing, 12: 596-605.
[4] Liu, Y., Xu,L.X., (2018) High-performance back-propagation neural network algorithm for classification of sea load data [J]. Automation of power system, 42(21):96-103.
[5] Shan,S.G., Meina,K., Liu,X., (2016) Deep learning: Renaissance and reform of multi-layer neural network[J]. Science and technology herald, 34 (14):60-70.
[6] Zhang,L., Zhang,Y., (2016) Big data analysis by infinite deep neural networks [J]. Journal of Computer Research and Development.,53 (1):68-79.
[7] Zan,H.Y., Guo,M., Chai,Y.M., et al. (2010) An investigation on the emotional orientation of news reports [J]. Computer engineering,36(15):20-22.