Characters, Spirals and Unit Dynamics by training neural network model

Zhe Luo¹

¹ Data Science and Engineering, University of New South Wales, 2052 NSW, Australia

Abstract. Deep learning is an integral part of machine learning by building a structure like the neural network in the human brain. Amounts of different neural network models have been emerged, improved, and transformed. This article will implement various neural network models for three different specific tasks (Japanese character recognition/double helix task/remote unit dynamics) and analyse the fine-tuning parameters process (different network layers, different types of network layers). The result is that generally, models with more layers would perform better. However, other parameter could have significant effects on the performance. For example, activation function, different numbers of hidden nodes, different initial weights, and different periods, etc.

1. Introduction

Nowadays, machine learning could make good use of data that people have ignored for years. However, when it comes to the picture, typical machine learning techniques would not satisfy the high time cost caused by extensive computation. Then, deep learning comes. In June 2012, the project named ‘Brian’ was revealed by media in Google company [1], which played a significant role in developing deep learning. Over 1600 CPU cores were used at that time from a deep neural network learning model. It turns out that this project won a big success in image and sound recognition. Also, in the same year, Krizhevsky and Hinton proposed AlexNet CNN structure [2], which won the championship with 10.9% higher over the second place in ImageNet Large-Scale Visual Recognition Challenge. The remarkable progress in ImageNet proved the future and capability of deep learning. However, unsolved problems still exist in the deep learning part. More research is needed to be applied in real life to test the feasibility of different advanced models in the future. This paper aims to study the different deep learning models from two layers to the convolution layer and then to the entire connection layer. In addition, the number of parameters will be changed to test their impact on the deep learning model. For example, the activation function (tanh / Relu / sigmoid) and the different numbers of nodes in each layer. Thus, three tasks could help us evaluate by testing and comparing different accuracies.

2. Thesis

2.1 KMNIST, Two Spirals Problem and Hidden Unit Dynamics

The majority of old Japanese published could not be read over 150 years ago because of the long-time evolved education system. Many pictures of old Japanese character are included in Kuzushiji-MNIST(KMNIST). This data set is adapted from the Kuzushiji data set, and it is provided in original MNIST and NumPy format [3]. We would use the easier one: 10 Hiragana characters with 7000 samples
per class. Then, Lang and Witbrock proposed the two spiral problems in 1988 [4]. The distinction of two other intertwined spirals is an efficient way to exhibit network models and compare the learning speed of variants of the back-propagation algorithm. The advantage is that we could make analytic visually. Also, A hidden unit corresponds to the output of single filter. After exploring different dynamic hidden units with different periods, the activation (point) and output boundary of hidden units are described to move with the training process. The random number size or parameter type of tensor will produce different types of images.

2.2 CNN
From LeNet-5 Architecture [5] by Yann LeCun to Google Net architecture by Christian Szegedy et al. from Google Research [6], from ResNet by Kaiming He et al. in 2015 [7] to Squeeze-and Excitation Network (SENet) in 2017 [8], with the development of the increasing capacity of the computer and the increasing size of the training set, CNN has evolved and improved for generations. It could achieve advanced humanization in many complex tasks.

3. Progress and Result

3.1 Japanese character recognition
Download the KMNIST dataset and use the terminal implementation model to calculate the linear function of the pixels in the image, and then log SoftMax. The final loss is 1.0098, and the accuracy is 70% (6968/10000). Then, a fully connected 2-layer network with tanh activation function is implemented at the hidden node, and SoftMax is recorded at the output node. Try different values for the number of hidden nodes to achieve an accuracy higher than 84%. The result is that when the value of hidden nodes is 20/50/150/200/500, the accuracy would be 76%/81%/84%/85%/85%. After that, implement a CNN with two convolutional layers and one FCL with Relu activation function and epochs larger than 10. The absolute accuracy is 93% (9270/10000). The relative accuracy of these models should be observed and discussed. At the same time, the confusion matrix is used to find the features that are most likely to be misjudged by the model and analyze them. As a result, we could change some parameters like different numbers of hidden nodes or different numbers of different layers to analyze.

3.2 Two Spirals Problem
Using RawNet function to operates on the raw input without converting to polar coordinates and PolarNet function to operate as follows: First, the input is converted to polar co-ordinates with \( r = \sqrt{x^2 + y^2} \), \( a = \text{atan2}(y, x) \). Next, \((r, a)\) is fed into a fully connected neural network with one hidden layer using tanh activation, followed by a single output using sigmoid activation. Then, find the minimum number of hidden nodes required to correct the training data within 20000 epochs. Then select a value as the number of hidden nodes with the size of the initial weight within 20000 epochs and make the correct classification. Because when the number of hidden nodes is ten and the initial weight is 0.11, the accuracy is stable at 53.61%, so different initial weights are used. The result shows in Table 1.

| Initial weight | Final epoch |
|----------------|-------------|
| 0.11           | 20700/20000 |
| 0.12           | 18400/20000 |
| 0.13           | 18100/20000 |
| 0.15           | 13900/20000 |

3.3 Hidden Unit Dynamics
This task is a 9-2-9 encoder with input and target determined by one-hot encoding. Save the most
representative images generated (epochs from 50 to 15070). Analyze the outputs and illustrate the movement of hidden unit activations (dots) and output boundaries during training. Here are two results:

4. Analysis

4.1 Japanese character recognition
As for the model 1/model 2/model 3, the accuracy is 70%/86%/93%. The differences between the models in Table 3 indicate that different neural networks perform differently in the same data set. Comparing the model number and the accuracy we could get: A. The accuracy of the three models is increasing. B. For images, the convolutional network has excellent advantages. It has the highest accuracy and uses a small computational cost. C. The two-tier network function is behaving generally compared to the best network function. D. The first model has the worst performance and is not suitable for images. Pictures with higher resolution would help the model perform better. Therefore, after experimenting with this data set and other architectures and parameters, it is found that: For model 2: tried many different hidden nodes. When the number of hidden nodes is less than 200, the accuracy will decrease as the number of hidden nodes decreases. For model 3: When the top pooling layer is deleted in the third model, and the input feature of linear layer one is changed to 12800, the accuracy becomes 92%, which means that the pooling layer is accommodating to improve the accuracy.

4.2 Two Spirals Problem
For functions calculated by PolarNet, each hidden node is not linear. However, for the function calculated by RawNet, each hidden node in hidden layer 1 is linear. All hidden nodes cannot learn the correct classification. After trying different initial weights, it turns out that the initial weights are not necessarily related to the learning speed of RawNet. Sometimes, the larger the initial weight, the faster the speed. As for the success of learning, the experimental results show that they are related. For example, when the initial weight is 0.1, 0.15 and 0.2, the epoch would be 6500, 5200, 14200. After testing other changes, it turned out that: a. For the PolarNet function, as the batch size increases from 97 to 194, the epoch decreases. Each different batch size is responsible for a different epoch. The experimental results show that the larger the recommended batch, the faster the speed. b. Because the function with SGD could not achieve 100%, SGD could be worse than Adam to achieve 100% accuracy especially when they have same epochs. d. As for the activation function, after changing tanh to Relu, Relu is worse than tanh. E.g., In the PolarRaw function, Relu needs more time to reach 100% accuracy.

4.3 Two Spirals Problem
From Figure 1 and Figure 2 we could propose that the hidden units gather at one place at the beginning. The reason is that when the initial weight is about 0.001, the activation function is tanh, which leading
the result to be 0. As the training iteration of the neural network, most of the spots spread to the outer edges of the hidden unit space with an increasing speed. However, the two dots move slowly. As a result, a Nine-angled picture was generated. Every dot was separated from others by boundary lines, which illustrates that the input and target output could be determined by one-hot coding.

5. Conclusion and Future
The main content is implementing and training various neural network models for three different specific tasks: Japanese character recognition, Twin Spirals Task, and Hidden Unit Dynamics. Then analyze the results by fine-tuning the parameters. The structure of the neural network is like human brain. Generally, models with more hidden layers perform better. However, we need to consider issues such as cost and prevention of over-assembly. For the computer part, we could use K-fold cross-validation like RandomizedSearchCV or other open Python libraries. For the human part, we could increase the number of hidden layers from one. The reason is that if a hidden layer has enough nodes, the model’s performance would be better. Also, we could try different values of the number of nodes from small to large until overfitting. It is easier for us to choose a model with more layers and nodes than we need and uses early stopping and other regularization techniques to prevent the model from overfitting, which is usually more straightforward and effective [10]. In the future, more research and practice are needed to test the influence of different fine-tuning hyperparameters on different models. For example, in DeepMind's 2017 paper [11], the author used hyperparameters to optimize many models. On the other hand, evolutionary algorithms will play an essential role in this regard.

References
[1] MarkoffJ. How many computers to identify a cat? [N] The NewYork Times, 20212-06-25
[2] Alex Krizhevsky et al., “ImageNet Classification with Deep Convolutional Neural Networks”, Proceedings of the 25th International Conference on Neural Information Processing Systems 1 (2012): 1097–1105.
[3] KMNIST Dataset, http://codh.rois.ac.jp/kmnist/index.html.en
[4] Kevin J. Lang and Michael J. Witbrock, Learning to Tell Two Spirals Apart, 1988
[5] Yann LeCun et al., “Gradient-Based Learning Applied to Document Recognition”, Proceedings of the IEEE 86, no.11 (1998): 2278–2324. K. Elissa, “Title of paper if known,” unpublished.
[6] Christian Szegedy et al., “Going Deeper with Convolutions”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015): 1-9. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[7] Kaiming He et al., “Deep Residual Learning for Image Recognition”, arXiv preprint arXiv: 1512: 03385 (2015). M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
[8] Jie Hu et al., “Squeeze-and-Excitation Networks”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018): 7132–7141.
[9] Aurélien Géron et al., “The Architecture of the Visual Cortex”, Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow (2019): 447
[10] Aurélien Géron et al., “Fine-Tuning Neural Network Hyperparameters”, Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow (2019): 323-324
[11] Max Jaderberg et al., “Population Based Training of Neural Networks,” arXiv preprint arXiv:1711.09846 (2017).