A Comparative Study between Multilayer Perceptron and recurrent neural networks trained with backpropagation through time

Li Guo
Li.Guo@city.ac.uk

Abstract
This paper aims to discuss a critical evaluation of two algorithm models performed in a supervised pattern recognition task on Fashion MNIST dataset. The two models are Multilayer Perceptron (MLP) and recurrent neural networks trained with backpropagation through time (RNNs). Each model is trained with same parameters and given results of accuracy under same conditions. The tested results from different models are given in figures to help compare the suitability for this dataset, and as a result, the MLP algorithm appears to be more suitable than the RNNs in this dataset.

1. Introduction

The MNIST dataset is usually the first dataset that researchers try to validate their algorithms as a benchmark, and the researchers will say that "if the algorithm does not work on MNIST, it will not work at all". However, based on the situation of MNIST dataset has been overused and is too easy to deal with, Fashion MNIST dataset has been presented by Han to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms sharing the same image size, data format and the structure of training and testing splits [1].

1.1. Multilayer perceptron (MLP)

Multilayer perceptron (MLP) is one of the popular Artificial Neural Networks (ANNs), which applies a supervised training procedure using examples of known outputs. With input data, multilayer perception generates a nonlinear function model to give the prediction of output data. It passes the input data unidirectionally from the input layer to the output layer through the hidden layers, however, the backpropagation, which is the widely known MLP learning algorithm, works the opposite way of feedforward neural network to correct the weight by propagating the errors from layer to layer starting with the output layer [2].

1.2. RNNs trained with backpropagation through time (RNNs)

Recurrent neural networks (RNNs) are capable of detect patterns in a time series data with sequential models, for example, the backpropagation through time (BPTT), which is usually used for learning RNNs, is developed from backpropagation algorithm to predict time series data by considering both the latest input and all the previous input in the network [3]. For each time step in the procedure of BPTT can be considered as limiting the numbers of hidden layers is because that BPTT can basically unfolds the RNN to create a new layer in the process [4].

2. Dataset

The dataset used in this project is based on the Fashion MNIST dataset from GitHub presented by Xiao, H., Rasul, K. and Vollgraf, R. this dataset contains 70,000 images of fashion products from 10 categories, with the size of each image with 28X28 grayscale. Depending on the intention of Fashion MNIST is to serve as a direct drop-in dataset as original MNIST dataset, there will be no need to pre-process this dataset [1].

In Figure 1, is shows that there are 10 labels for Fashion MNIST, under each label, there are 6000 images for training, and the rest of 1000 images are for testing.

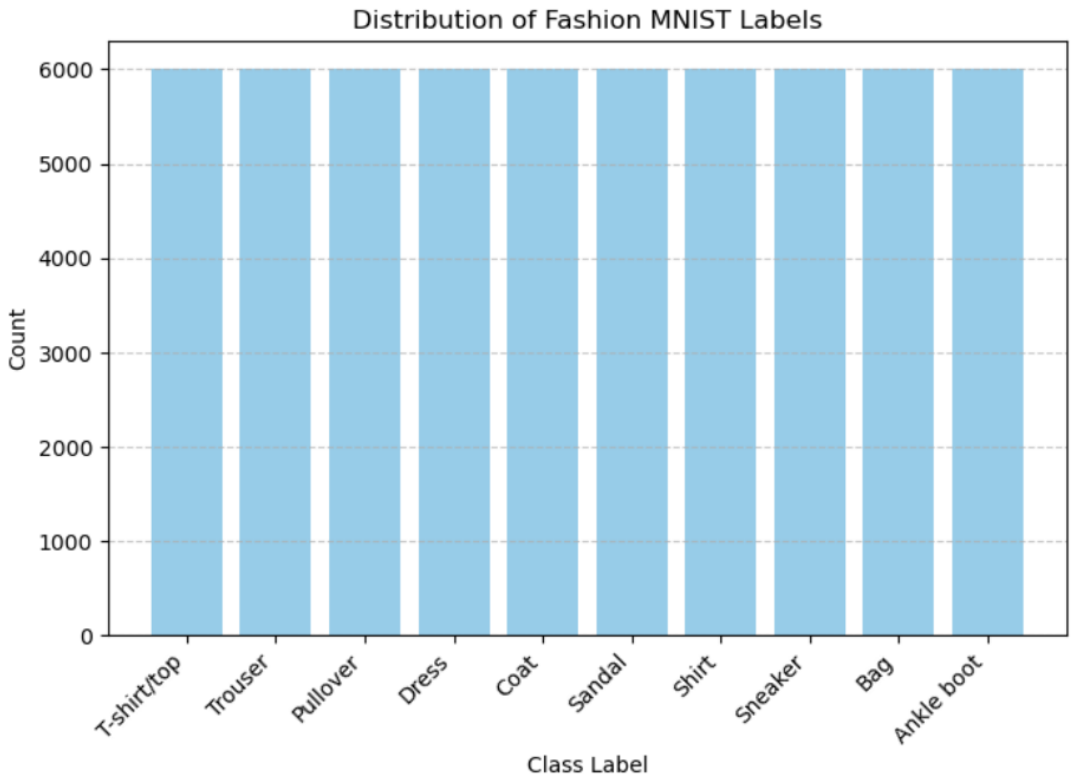

Figure 1 All the categories and images count in Fashion MNIST dataset

3. Methods

In this section, the details of training, architecture and hyperparameters of the algorithms will be provided to build the MLP and RNNs.

3.1. Methodology

The dataset has been divided into two parts, 14% of the dataset is the testing data, which means that 10,000 images are the testing data, and the rest 60,000 images are the training data.

The model selection process the backpropagation by using array to put all the necessary hyperparameters into it and give the array to the model to train two models. The range of all the parameters are set as the same to make sure the result can be comparable depending on the same input parameters.

For the algorithm comparison, the models are retrained using the same dataset, the same amount of input hyperparameters, and give the same kind of figures to show the accuracy of the MLP and RNNs, by comparing the best results of the models, the results of the comparison can be concluded.

3.2. Architecture and parameters used for the MLP

In the MLP model, first create the MLP model with forward and backward function to define the MLP model as backpropagation. In the backward function, it will calculate the weight and bias of every step in the procedure including the output layer and hidden layer. Then define the hyperparameters of learning rates from 0.1 to 1, the hidden layers from 64, 128 and 256. In the training process, the model will first calculate the loss of the data and then find out the accuracy of the models.

3.3. Architecture and parameters used for the RNNs

In the RNNs training with BPTT model, first create the normal RNN model with forward function, after defining the hyperparameters including hidden sizes from 64, 128 and 256, learning rates from 0.001 to 0.01, then calculate the losses of hidden sizes and learning rates, and finally calculate the accuracy of the mode.

4. Results, Findings and Evaluation

4.1. Model selection

In the training process, the print of the result will give all the training accuracy in two different types of figures. For the MLP model, according to figure 2 and figure 4, they show that the highest validation accuracy is 0.8556 and the validation loss is 0.0208 under the hyperparameters of learning rate with 1 and hidden layer size with 128. According to figure 3 and figure 5, they show that the highest validation accuracy is 0.6943 and training loss is 0.8227 under the hyperparameters of learning rate with 0.001 and the hidden layer size of 256.

For the MLP model, most of the accuracy are over 0.8, however for the RNNs model, the accuracy varies differently from 0.35 to 0.7, which means that RNNs model is more sensitive than MLP model, seems that the learning rate do not play a role in MLP models.

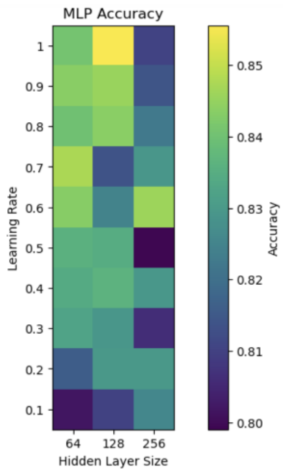

Figure 2 Accuracy for MLP

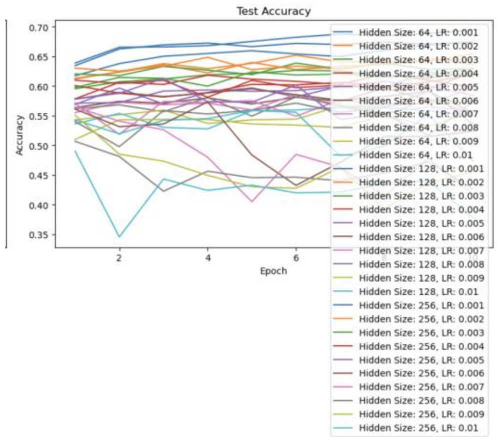

Figure 3 Accuracy for RNNs

```
Learning Rate: 1, Hidden Layer Size: 128, Epoch: 8/10, Validation Loss: 0.0210, Validation Accuracy: 0.8540
Learning Rate: 1, Hidden Layer Size: 128, Epoch: 9/10, Validation Loss: 0.0208, Validation Accuracy: 0.8564
Learning Rate: 1, Hidden Layer Size: 128, Epoch: 10/10, Validation Loss: 0.0208, Validation Accuracy: 0.8556
Learning Rate: 1, Hidden Layer Size: 128, Test Accuracy: 0.8421
Learning Rate: 1, Hidden Layer Size: 256, Epoch: 1/10, Validation Loss: 0.0380, Validation Accuracy: 0.7613
```

Figure 4 Highest Accuracy for MLP

```
Hidden Size: 256, Learning Rate: 0.001, Epoch: 7/10, Test Loss: 0.8390, Test Accuracy: 0.6876
Hidden Size: 256, Learning Rate: 0.001, Epoch: 8/10, Test Loss: 0.8357, Test Accuracy: 0.6850
Hidden Size: 256, Learning Rate: 0.001, Epoch: 9/10, Test Loss: 0.8353, Test Accuracy: 0.6891
Hidden Size: 256, Learning Rate: 0.001, Epoch: 10/10, Test Loss: 0.8227, Test Accuracy: 0.6943
Hidden Size: 256, Learning Rate: 0.002, Epoch: 1/10, Test Loss: 0.9745, Test Accuracy: 0.6302
Hidden Size: 256, Learning Rate: 0.002, Epoch: 2/10, Test Loss: 0.9738, Test Accuracy: 0.6251
```

Figure 5 Highest Accuracy for RNNs

4.2. Algorithm comparison

By comparing from Figure 2 and 3, MLP models can present the narrowed range of accuracy based on the constrained learning rate and hidden layer sizes, however, the same conditions of constrained learning rate and hidden layer sizes for RNNs training with BPTT can present the accuracy significantly. The best accuracy of MLP is 0.8556 (4) and the best accuracy of RNNs is 0.6943 (5) are significantly different, shows that RNNs train the dataset worse than MLP.

5. Conclusion

In conclusion, this study shows that Multilayer Perceptron is better for training Fashion MNIST dataset than recurrent neural networks training with backpropagation through time, even though both models show the better result in training than in testing. Eather way, the MLP appears to be a better algorithm than the RNNs in this certain situation, depending on the better accuracy the MLP has presented.

There also few flaws in both models, the hyperparameters can be constrained with wider range to train the model in order to present the higher accuracy than current result, and the functions of both models can be improved to reach better training result.

6. References

[1] Xiao, H., Rasul, K., Vollgraf, R., Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms (2017). Doi: 10.48550/arXiv.1708.07747

[2] Taud, H., Mas, J. (2018). Multilayer Perceptron (MLP). In: Camacho Olmedo, M., Paegelow, M., Mas, JF., Escobar, F. (eds) Geomatic Approaches for Modeling Land Change Scenarios. Lecture Notes in Geoinformation and Cartography. Springer, Cham. https://doi.org/10.1007/978-3-319-60801-3_27

[3] W. M. Septiawan and S. N. Endah, "Suitable Recurrent Neural Network for Air Quality Prediction With Backpropagation Through Time," 2018 2nd International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2018, pp. 1-6, doi: 10.1109/ICICOS.2018.8621720.

[4] Schmidt, M. R., Recurrent Neural Networks (RNNs): A gentle introduction and Overview (2019). Doi: 10.48550/arXiv.1912.05911