Abstract:

In the project, we implement the different neural-network methods, including Multi-Layer Perceptron(MLP) with different numbers of layers and activation functions, and Convolutional Neural-Networks(CNN) on a fashion dataset to investigate and evaluate the method performance.

We first process the dataset, construct the neural-network methods, and implement those with different parameters on the dataset to train the model. We investigate the performance of and choose the best architecture among several parameter combinations on the dataset, and our key takeaways include follows. The normalization of the dataset is necessary for classification; 2 hidden layers perform better than 1 hidden layer (or no hidden layer), but too many layers would make the model too complicated and leads to overfitting; Regularization would prevent overfitting and increase the testing accuracy; Leaky-ReLU and ReLU generate similar results for this dataset, but both perform better than tanh activation. Besides, we also found that more hidden units and epochs lead to better model performance.

Introduction:

In the assignment, we are assigned the tasks to implement different NNets on the Fashion_Mnist dataset to train the model and predict the label of the target variables. We are given one image dataset: The Fashion_Mnist dataset. It contains 60000 instances in training datasets and 10000 instances in test datasets. Its target variables include 10 categories of clothes.

After investigations and experiments, we have the following findings: First, in this case (for the Fashion Dataset), data normalization is very important since it could affect the result in the SoftMax function. Second, 2 hidden layers generate better performance than 0 and 1 hidden layer, but too many hidden layers would lead to the overfitting problem. Third, regularization would prevent such problems and help generate better results (higher test accuracy). Fourth, Leaky-ReLU and ReLU have same-level performances in this case, and both perform better than tanh activation. This could be because tanh works mainly for binary classification.Fifth, more epochs help the model perform better, both for MLP and CNN. Lastly, by doing additional research, we found that more hidden units lead to higher test accuracy for MLP, and more filter size leads to worse model performance for CNN.

Dataset:

We are given one dataset in the assignment: the Fashion_Mnist Dataset. It contains 60,000 instances in the training dataset and 10,000 in the test dataset. For each instance, it is an image with size 28*28. It contains 10 different labels under the target variables, each of which is a category of clothes. The dataset is balanced, with 6000(1/10) instances under each of the 10 labels.

To process the data, we first vectorize the image instances. We use the reshape function under the Numpy package to stretch the 28*28 matrix into a 784, making the final shape of the training dataset to be 60000*784 and the size of the testing dataset to be 10000*784.

Next, we normalize the data by dividing the standard error.

Results:

First, we construct and train the MLP with different hidden layers and parameters: No hidden layer, Single hidden layer with 128 units and ReLu activation, and Two hidden layers with the same number of units and activation function. The test accuracy score of these three models is as follows (**Task 3.1**):

	Single	One Hidden Layer	Two Hidden Layers
Testing Accuracy	0.6595	0.7286	0.7453

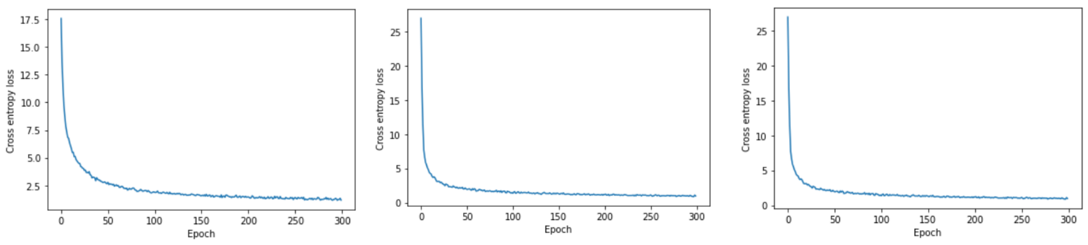


Figure 3.1 Charts of Entropy Cross Loss Function on three models.

From the result, we can see that with the increase of layers and attendance of non-linearity (activation function), the model's performance has been improved (the test accuracy increases from 0.65 to 0.74). The result also follows our expectations.

Next, we compare how different activation functions affect the model performance(**Task 3.2**). We use the NNets with 2 hidden layers and compare the performance difference between using the tanh and Leaky-ReLU. The test accuracy for the tanh is **0.6497** and for the Leaky-ReLU is **0.7443**, meaning that the Leaky performs better than the tanh in this case. Also, tanh takes longer time than Leaky ReLu activation.

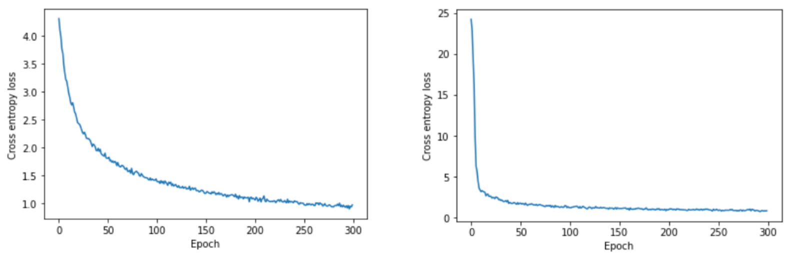


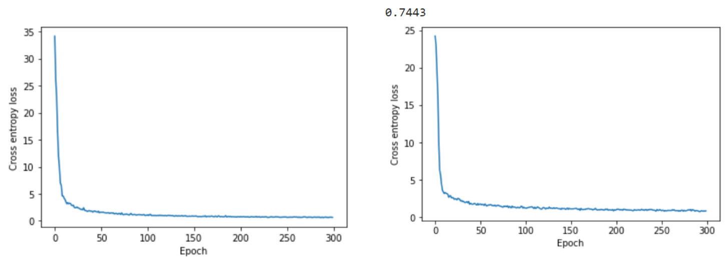
Figure 3.2 Charts of Entropy Cross Loss function on different activations,

However, in general cases, it is not always the case that Leaky-ReLu performs better than tanh activation. Tanh is mainly used in the classification between two labels, so in this dataset, with 10 labels for the target variable, it is expected that Leaky-ReLu provides better performance.

Then we check how L2 Regularization could improve the model performance (**Task 3.3**). We construct an MLP with 2 hidden layers with 128 units and ReLu activations, and this time plug in L2 Regularization. We choose the Lambda as 0.1. The result is shown below:

	Without L2 Regularization	With L2 Regularization
Test Accuracy	0.7453	0.7661

From the result, we could see that after adding the L2 Regularization, the performance slightly increases, from 0.74 to 0.76.



```
0.7443
```

Figure 3.3 Cost function with/without L2 Regularization

We also want to see the influence if we do not normalize the dataset (**Task 3.4**). By training the MLP with 2 hidden layers and 128 units with ReLU on the unnormalized dataset, we have an accuracy score of 0.1. Since the dataset is not normalized, we have outliers with very negative instances, which leads to the computation result of the exp function near 0 in the SoftMax Function, and thus leads to the divided by 0 error. Therefore we cannot estimate the result if we do not normalize the dataset, and if we replace the nan with other values, for example, 1, we neither gain a good result.

Next, we implemented CNN from the TensorFlow Package. We created a ConvNet with 2 convolutional and 2 fully connected layers, setting the hidden units as 128 and the activation as ReLu. The test accuracy reached 0.9336 in epoch 5 (**Task 3.5**). Using ConvNet greatly increases the accuracy compared to MLP.

Afterward, we move to construct the MLP architecture with the best performance by choosing the hyperparameter (**Task 3.6**). Since we already verified in Task 3.1 that the model has higher accuracy with more hidden layers, we would choose the number of hidden layers to be 2. Thus we would choose from Leaky-ReLU and ReLU as activation and test on different lambda values for L2 Regularization. The result is shown as below:

	0.01	0.1	1
Test Accuracy (ReLu)	0.7518	**0.7687**	0.4549
Test Accuracy (Leaky)	0.7592	0.7664	0.4720

So the MLP architecture we select is:

	Hidden Layer	Activation	Lambda	Test Accuracy
Parameters	2	ReLU	0.1	0.7687

Then we test on how epochs number affect the performance of both MLP and CNN (**Task 3.7**). In CNN, we see that the accuracy rate increases with increasing epochs. The results from epochs 1 to 5 are listed below(Blow (Table and Figure 3.4):

	Epoch 1/5	Epoch 2/5	Epoch 3/5	Epoch 4/5	Epoch 5/5
Test Accuracy	0.8770	0.8995	0.8965	0.9078	0.9057

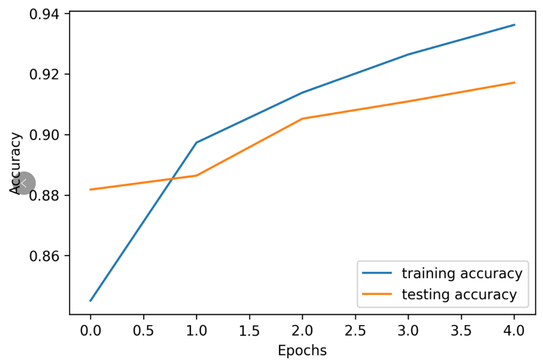


Figure 3.4 Test Accuracy w.r.t Epoches in CNN

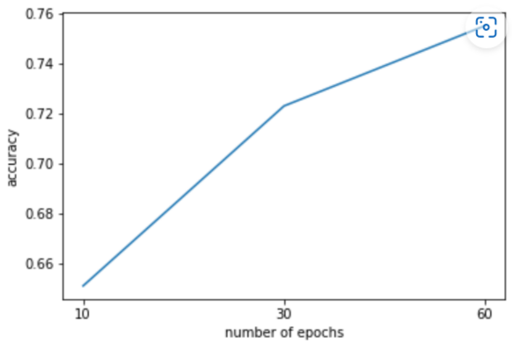


Figure 3.5 Test Accuracy w.r.t Epoches in MLP

For the MLP, we also construct a function to see the influence of epochs (see Figure 3.5 Above). Both of them show the pattern that larger epochs lead to higher test accuracy.

Besides from above, we also had some interesting findings for MLP and CNN. We test how hidden units affect the model result. By doing several trials from 16 to 128, we found that more hidden units lead to a better result. We also found that the large filter size in CNN leads to worse model performance (as shown below).

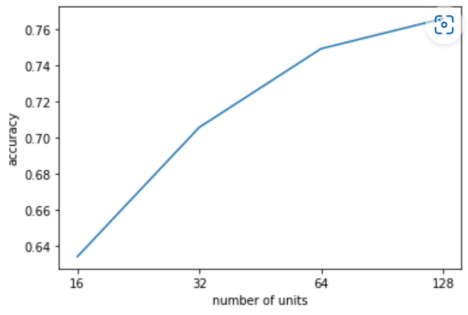


Figure 3.6 Accuracy of MLP w.r.t. Units

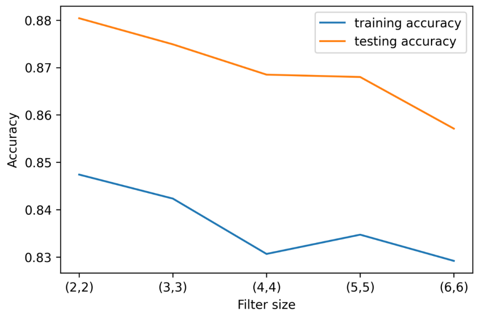


Figure 3.7 Accuracy w.r.t Filter Size

Discussion:

For the key learnings, we compare the performance between MLP and CNN and how different parameters (hidden units, hidden layers, Regularization, epochs) could affect the test accuracy. In this situation, it is essential to normalize the dataset since very negative outliers could cause nan problems in the following functions and lead to very low model performance. To specify, first, we check how hidden layers influenced MLP, finding that 2 hidden layers generate the best performances among 0,1, and 2 options. Even so, that does not mean that the more hidden layers the better since complicated models would cause the overfitting problem.
Then we moved to check on regularization, and we found that it could prevent the overfitting problem and thus improve the model performance. After this step, we check on the influence of activations. In this case, Leaky-ReLU and ReLU perform nearly the same, and both are better than tanh since tanh is mainly used in binary classification. Lastly, we tested different hidden units from 32 to 128. The result shows that the more hidden units, the better the performance.

We also implement the CNN method, finding that the larger the filter size, the worse the model performance. Also, for both CNN and MLP, larger epochs lead to better performance.

In future steps, we would like to explore what causes the difference in performance between CNN and MLP in our case. We have an accuracy score of 0.9 for CNN and an accuracy score of 0.8 for MLP, which is a relatively large difference. We would like to see the reasons behind this gap and further explore which scenarios best fit those two neural-network methods. We could also test and search for the most reasonable learning rate for the models.

Contribution:

Kevin and Yujie contributed most of the coding, and Zhixuan contributed a small part of the coding and finished the report.