UNIVERSITY OF PADOVA

Homework 1 Report

Subject: INP9086459 - NEURAL NETWORKS AND DEEP LEARNING 21/22
Professor: ALBERTO TESTOLIN
Student: Menglu Tao N°: 2041389

1-Regression Task

1. Introduction

The goal is to train a neural network to approximate an unknown function:

$$f: \mathbb{R} \rightarrow \mathbb{R}$$

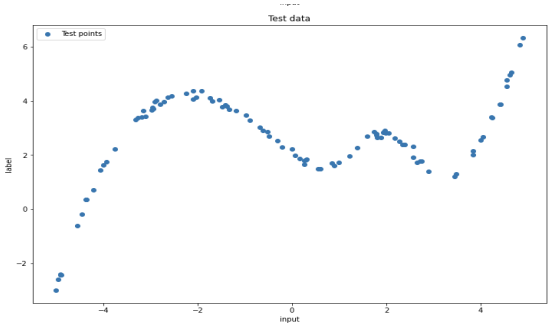
$$x \rightarrow y = f(x)$$

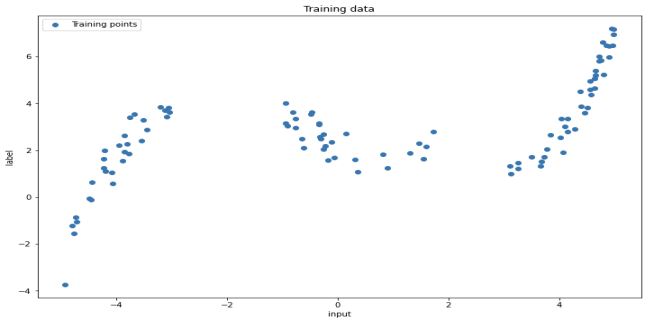
$$\text{network}(x) \approx f(x)$$

As training point, we only have noisy measures from the target function.

$$\hat{y} = f(x) + noise$$

In this task, we can observe the train dataset is very small, and the range is between (-5,5) regarding to x coordinate and (-4,7) regarding to y coordinates. Also when comparing train dataset and test dataset, there are some points are missing when x=2 and x=-2 in train dataset, which will bring some challenge to the predicting process.





2. Solutions

Based are the analysis above, given the situation that we're faced with a small dataset, then cross validation must be applied to avoid overfitting issue.

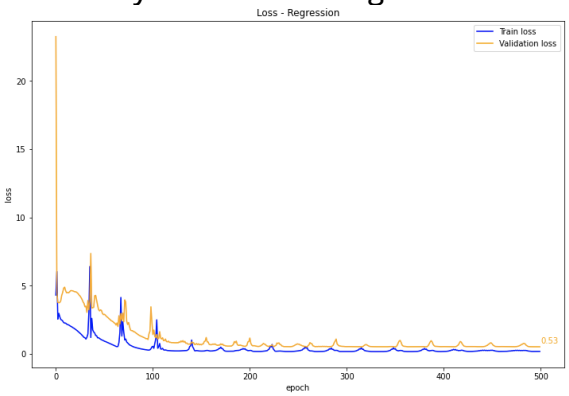
Network architecture: A fully connected networks with 3 hidden layers

Configurations of the network model:

- **First layer number of neurons**: 256
- **Second layer number of neurons**: 512
- **Third layer number of neurons**: 768

- **Activation methods**: tahn / softplus / Relu , last layer didn't use activation
- **Optimizer**: Adam
- **Learning rate**: 1e-3
- **Regularization**: 1e-7
- **Criterion**: MSELoss
- **Max epoch**: 500
- **Early stopping**: 400 epochs

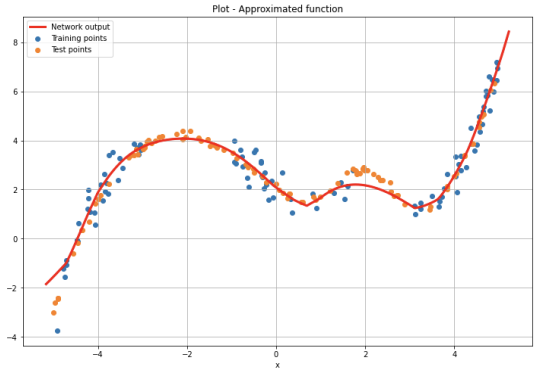
And with this configuration we can observe that the model itself already has achieved a good accuracy with:



- **Train Loss**: 0.194
- **Validation loss**: 0.532
- **Test loss**: 0.135

Considering the time consuming we didn't set up a complex grid search,yet we still try to find the best paras combination Thus the **grid search** runs as below:

- **optimizer__lr**: [0.001, 0.0013, 0.0015],
- **optimizer__weight_decay**: [1e-5, 1e-7],
- **batch_size**: [16, 32, 64]



And the results given to use the best combination is:
batch size = 16, learning rate = 0.0015, weight decay = 1e-5.
And using these new parameters set, we run the model again, The new results are:

- **Train Loss**: 0.209
- **Validation Loss**: 0.592
- **Test Loss**: 0.122

The test loss improved a little by 0.0013

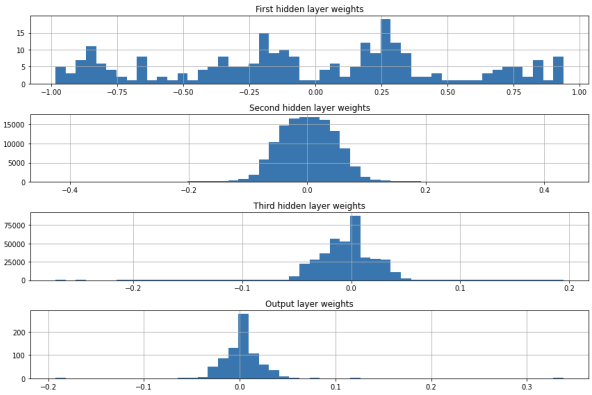
3.Analysis

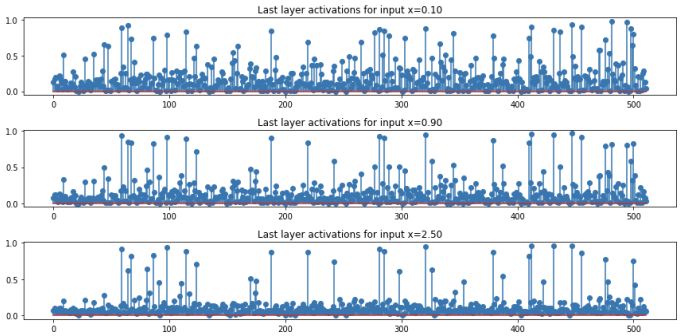
Weight histogram:

As we can observe from histogram, the weights distributions are normal, the first hidden layer is distributed evenly and last three are distributed more concentrated in one area.

Activation analysis:

The activation diagram shows that all the neurons are actively used, which shows the complexity of the networks is okay.





2-Classification Task

1.Introduction

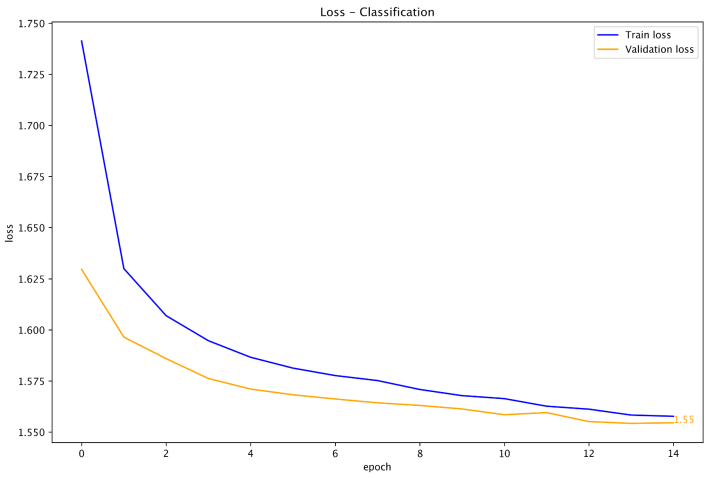
The goal is to train a neural network that maps an input image (from fashionMNIST) to one of ten classes. Since in image classification area, it's been tested that CNN is a good solution for this kind of problem because Its built-in convolutional layer reduces the high dimensionality of images without losing its information. So in the classification task we decided to use CNN model.

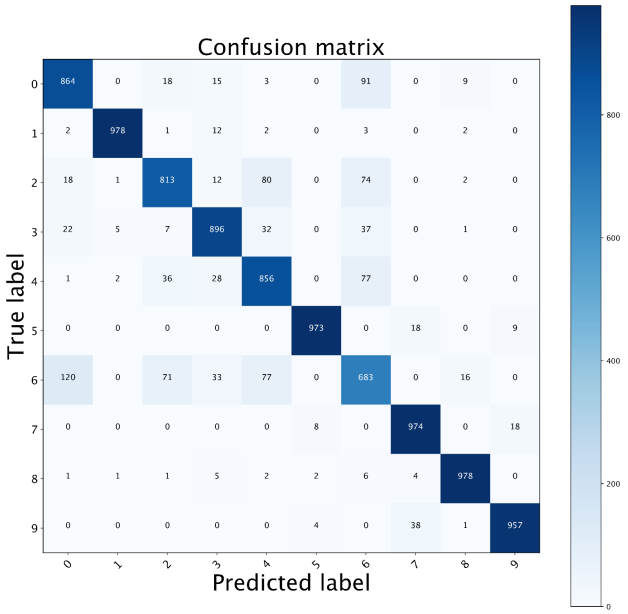
2.Solutions

Network structure: 2 convolutional layers with 2 max pooling layers and ended with 2 fully connected layers.
Using grid search trying to find the best parameters combination and the best one is as below:

Best Configurations for the network model:

- **Convolutional layer 1**: 32 filters with kernel size 3*3
- **Convolutional layer 2**: 48 filters with kernel size 3*3
- **Max_pool**: 2
- **FC layer 1 neuron number**: 128
- **FC layer 2 neuron numbe**r: 256
- **Dropout**: 0.2
- **Activation method**: Relu / softmax
- **Optimizer**: Adam
- **Learning rate**: 1e-4
- **Regularization L2**: 1e-7
- **Criterion**: CrossEntropyLoss
- **Max epoch**: 15
- **Early dropping** :500





Using the configuration above, train and test the model, we got the results below:

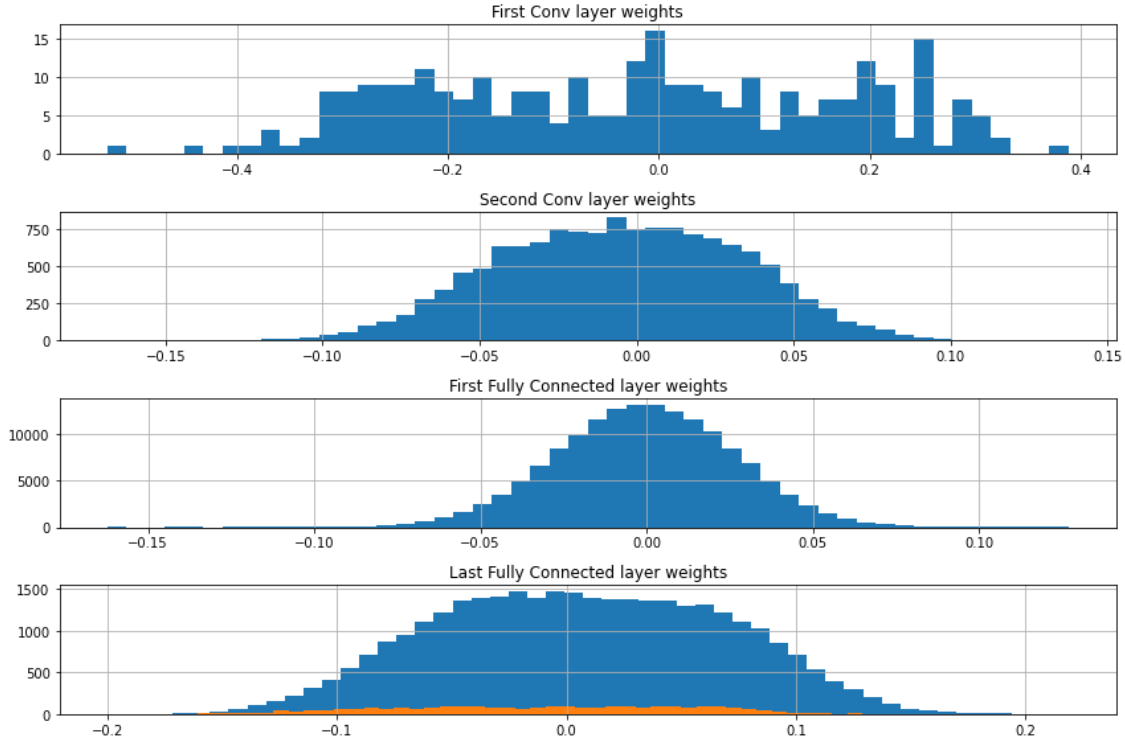
- **Train Loss**: 1.558
- **Validation Accuracy:** 0.906
- **Test Accuracy:** 0.898

And based on the generated confusion matrix, showing that label 6, label 4 , label 2 and label 0 have most misclassifications.

3.Analysis

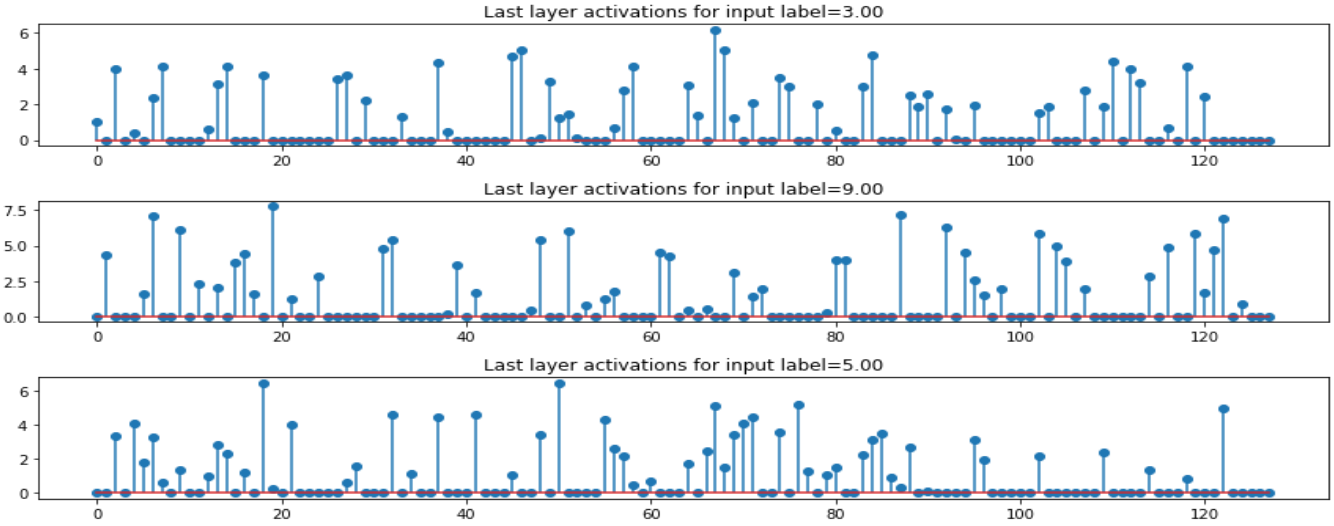
Weight histogram:

As we can observe from histogram, the weights distributions are normal, the 4 layers' weights are distributed evenly.



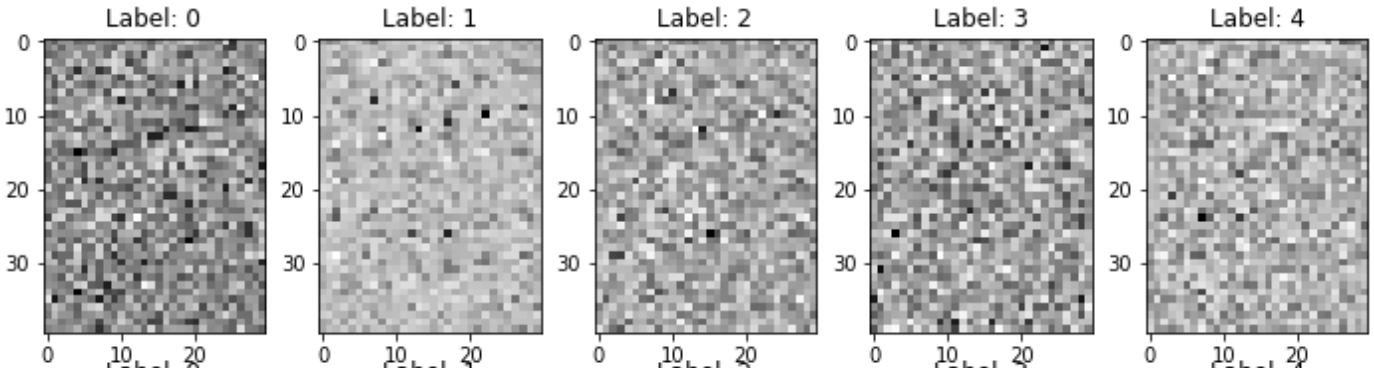
Activation analysis:

The activation diagram shows that many of the neurons are used, but some are not used, indicating that maybe we can change to a smaller network.

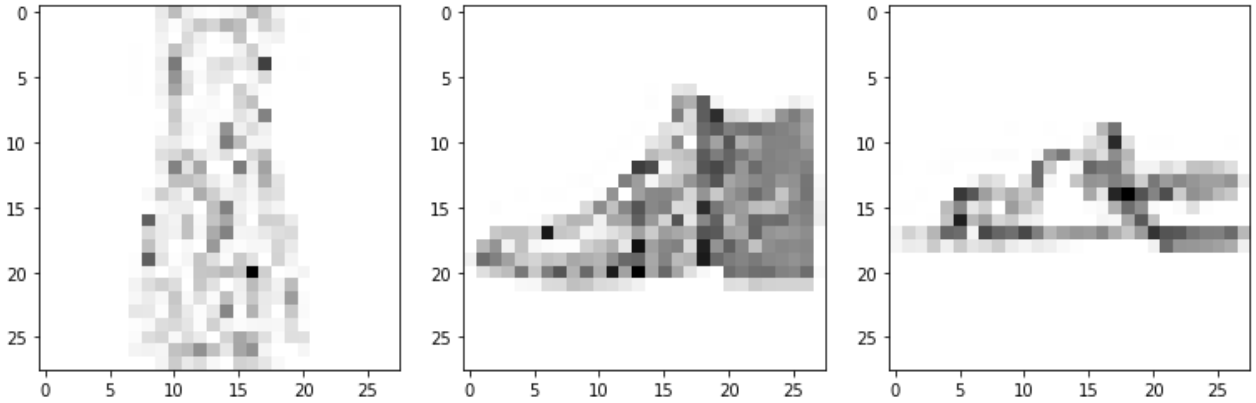


Receptive fields of the last layer :

We can see that the images are quite unclear, and can't tell the shape.

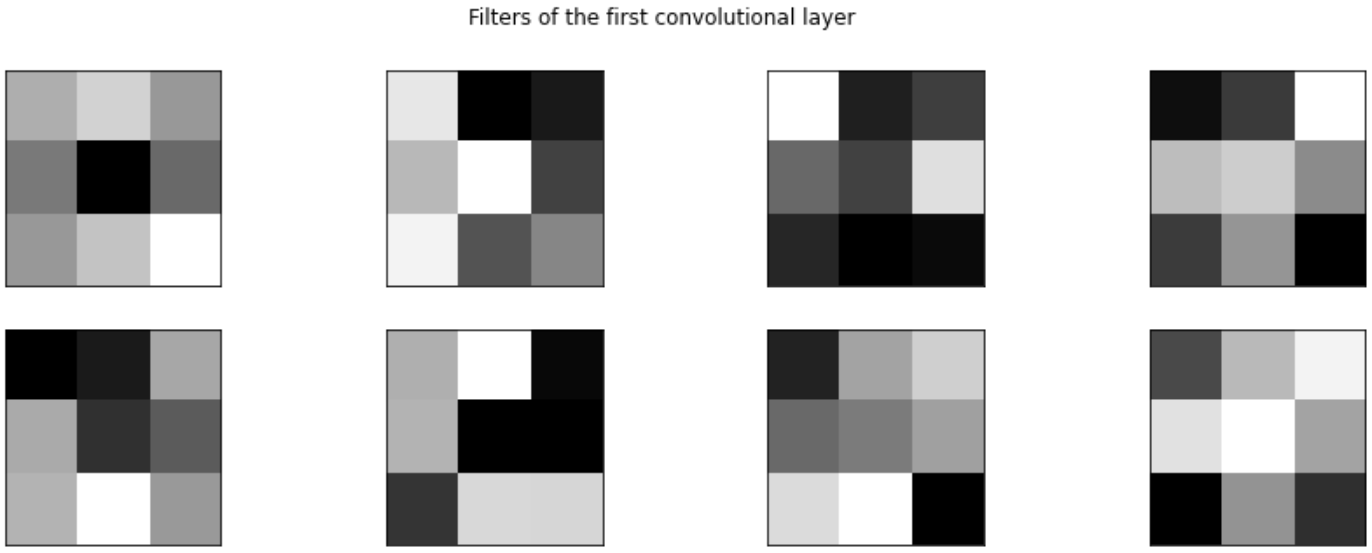


But after we visulize it, now we can have a clearer look.



Filters and feature maps:

Apply 3*3 filters of the first convolutional layer:



Feature maps of the first convolutional layer:
Shows that after applying the 8 filters, how the internal representation of input image look like.