Machine Learning Spring 2022
Assignment 2 Report

Ali Khalid
MSDS21001

May 8, 2022

1 Task # 1

1.1 Model Parameters

- Learning rate = 0.001
- Optimizer = Adam
- Epochs = 10
- model Layer with neurons in each layer:
  - Input Layer: 3*218*178
  - 3 Hidden Layer: [1000, 800, 400]
  - Output Layer: 120
- activation Function
  - Input Layer: No Activation function
  - 3 Hidden Layer: Relu
  - Output Layer: softmax

1.2 Loss Curves

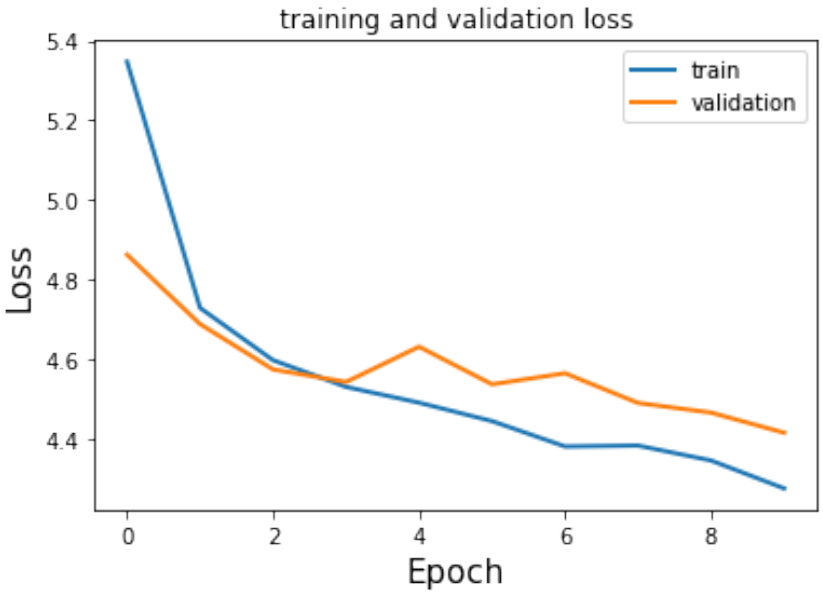

Figure 1: Task 1 loss curves

1.3 Accuracy Curves

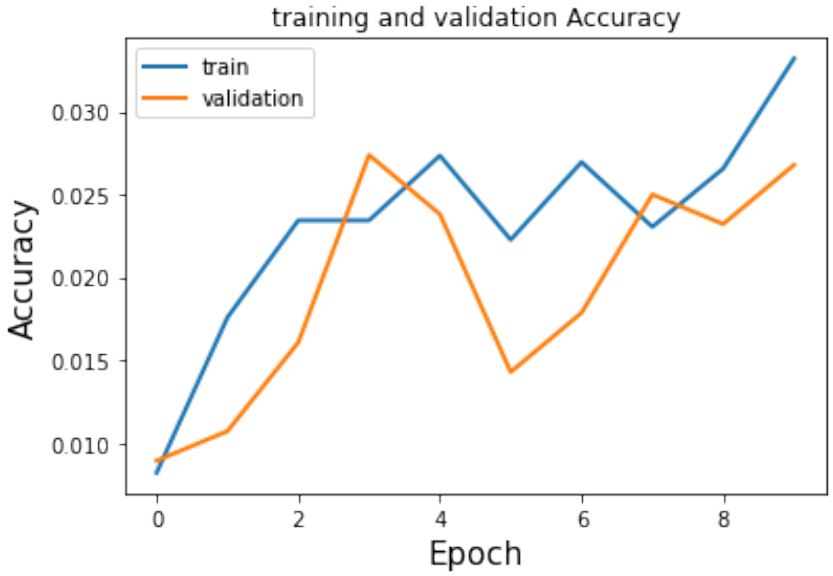

Figure 2: Task 1 accuracy curves

2 Task # 2

2.1 Model Parameters

- Learning rate = 0.001
- Optimizer = Adam
- Epochs = 10
- model Layer with neurons in each layer:
  - Input Layer: 3*218*178
  - 4 Hidden Layer: [1000, 800, 400, 200]
  - Output Layer: 10. Each image is passed from the network and then the difference of two output layers is taken and after that the sum of all 10 values in the difference vector is computed.
- activation Function
  - Input Layer: No Activation function
  - 4 Hidden Layer: Relu
  - Output Layer: Sigmoid. The final output after difference and summation, which is a single value, is subjected to sigmoid.

2.2 Loss Curves

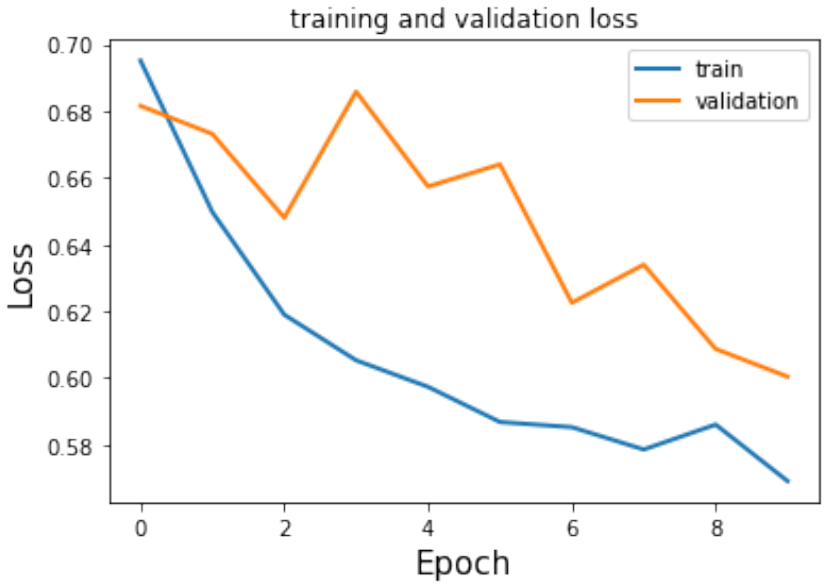

Figure 3: Task 2 loss curves

2.3 Accuracy Curves

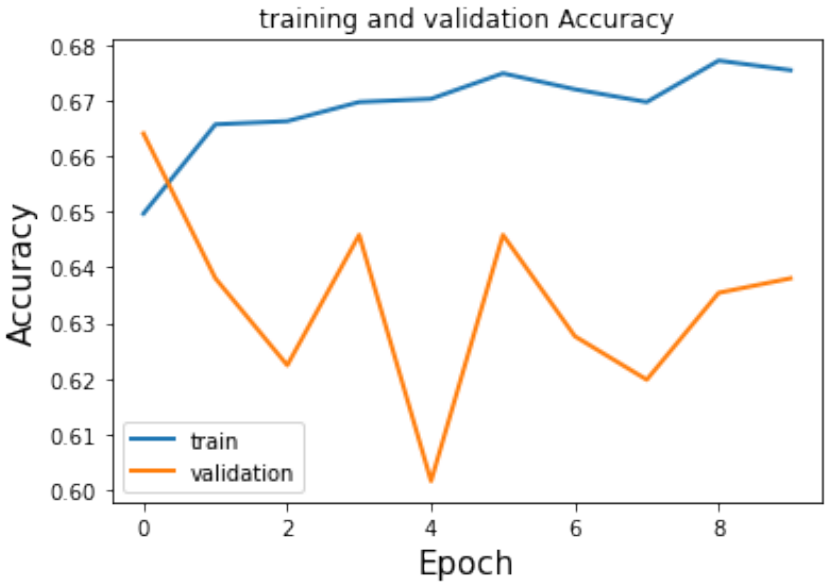

Figure 4: Task 2 accuracy curves

3 Task # 3

3.1 Model Parameters

- Learning rate = 0.0001
- Optimizer = Adam
- Epochs = 10

- model Layer with neurons in each layer:
  - Input Layer: 3*218*178
  - 4 Hidden Layer: [1000, 800, 400, 200]
  - Output Layer: 7
- activation Function
  - Input Layer: No Activation function
  - 4 Hidden Layer: Relu
  - Output Layer: sigmoid

3.2 Loss Curves

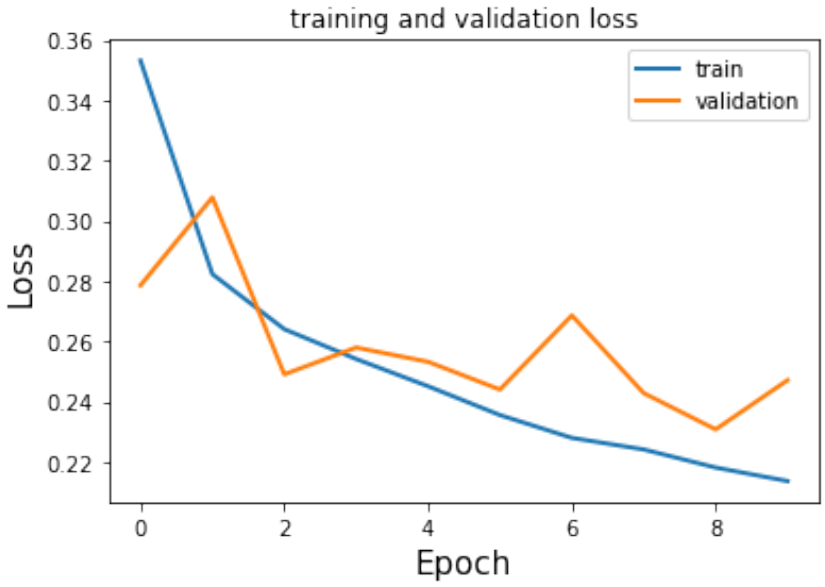

Figure 5: Task 3 loss curves

3.3 Accuracy Curves

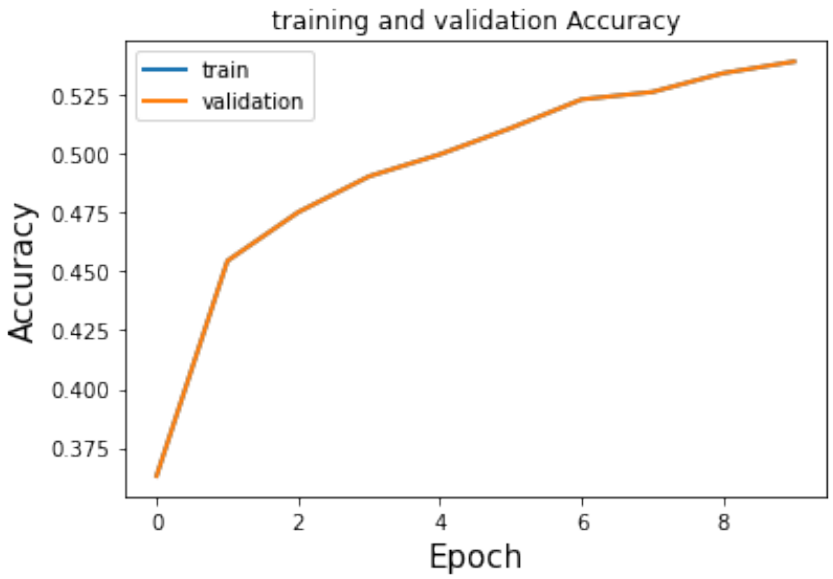

Figure 6: Task 3 accuracy curves

This model took 14+ hours to train and unfortunately I saved validation accuracy in both train and validation accuracy variables, hence the curves overlap.

4 Experiments

4.1 Task 1

The model architecture is kept same and only learning rate and epochs are changes

- Learning rate 0.001, epoch 1
  - train Accuracy: 0.8%
  - Validation Accuracy: 0.7%
  - test Accuracy: 0.7%
- Learning rate 0.001, epoch 10
  - train Accuracy: 3.3%
  - Validation Accuracy: 2.7%
  - test Accuracy: 1.1%

It is to be noted that loss curve is showing a decreasing trend and accuracy curve is showing an increasing trend as shown above. Therefore, increasing number of epochs will improve the accuracy but because of time limitations results is shown for max 10 epochs.

4.2 Task 2

The model architecture is kept same and only learning rate and epochs are changes.

- Learning rate 0.001, epoch 10
  - train Accuracy: 49%
  - Validation Accuracy: 50.8%
  - test Accuracy: 49.4%

- Learning rate 0.0001, epoch 10
  - train Accuracy: 67.5%
  - Validation Accuracy: 63.8%
  - test Accuracy: 63%

It is to be noted that loss curve is showing a decreasing trend and accuracy curve is showing an increasing trend as shown above. Therefore, increasing number of epochs will improve the accuracy but because of time limitations results is shown for max 10 epochs.

4.3 Task 3

The model architecture is kept same as described above. I tried to run my code on Google colab but I was facing some issues there. Therefore, I ran the code on local machine without GPU and it took 14+ hours for training with learning rate 0.0001 and epoch 10. I mistakenly saved test accuracy in the validation accuracy during training as well. That's why both the curves coincides, as shown above.

- Learning rate 0.001, epoch 10
  - train Accuracy: 53.9%
  - Validation Accuracy: 52.5%
  - test Accuracy: 51.6%