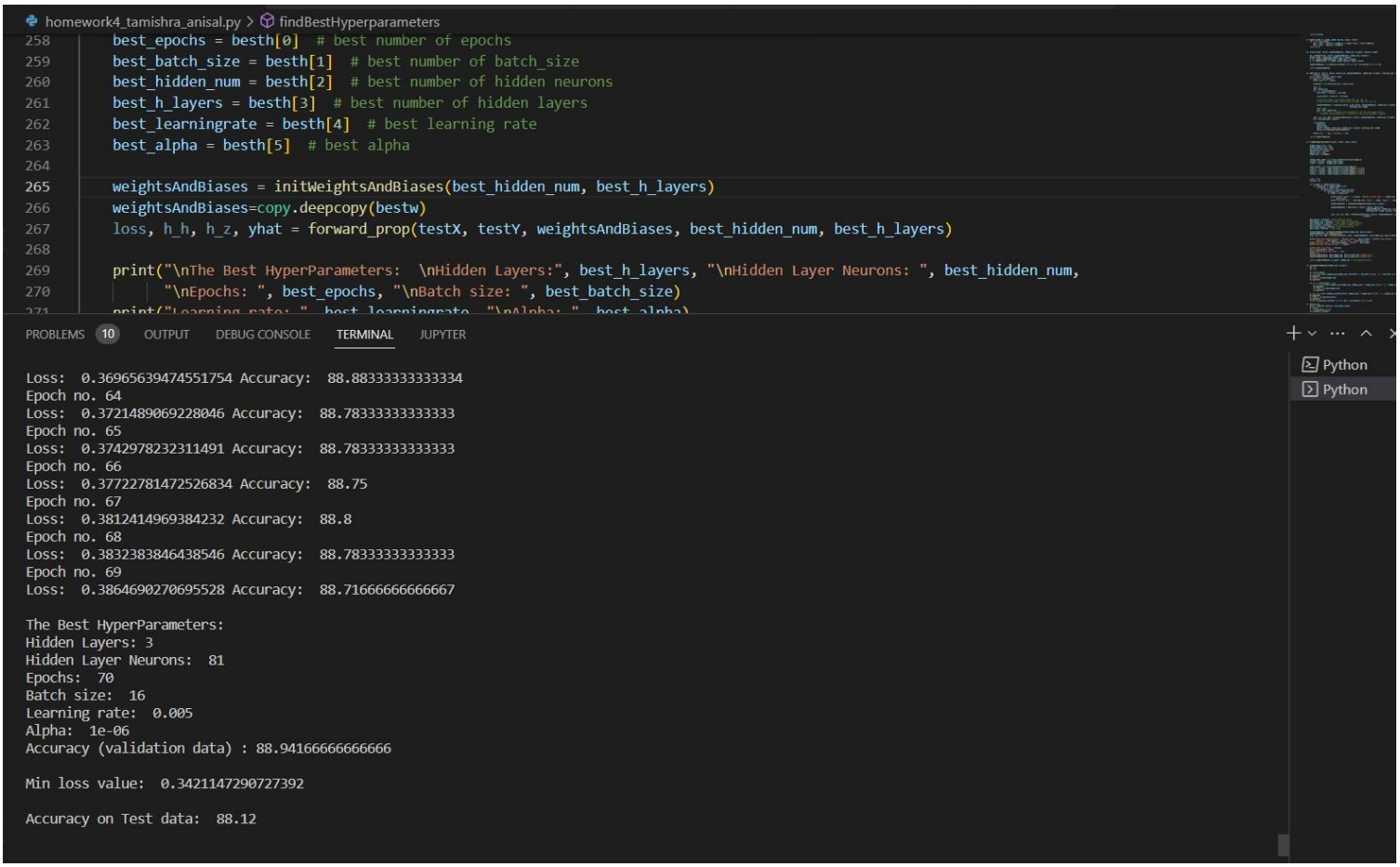
CS541 HW4 Screenshots



Best Accuracy and hyperparameters along loss and accuracy values after each iteration

homework4_tamishra_anisal.py > findBestHyperparameters

```
def saveVal(nested_list, name):
    with open(name, 'w') as file:
        for inner_list in nested_list:
            line = ','.join(map(str, inner_list)) + '\n'
            file.write(line)


if __name__ == "__main__":
    # TODO: Load data and split into train, validation, test sets
    X_tr = np.reshape(np.load("fashion_mnist_train_images.npy"), (-1, 28 * 28)) / 255
    trainX = X_tr.T
    ytr = np.load("fashion_mnist_train_labels.npy")
    train_Y = ytr
    X_te = np.reshape(np.load("fashion_mnist_test_images.npy"), (-1, 28 * 28)) / 255
    testX = X_te.T
    yte = np.load("fashion_mnist_test_labels.npy")
    test_Y = yte

    # onehot encoding
    trainY = np.zeros((train_Y.size, train_Y.max() + 1))
    testY = np.zeros((test_Y.size, test_Y.max() + 1))
    trainY[np.arange(train_Y.size), train_Y] = 1
    testY[np.arange(test_Y.size), test_Y] = 1
    trainY = trainY.T
    testY = testY.T
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
KeyboardInterrupt
PS C:\Users\Tanish A. Mishra\Documents\College Stuff\WPI\CS541\HW4\New folder> & "C:/Users/Tanish A. Mishra/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Tanish A. Mishra/Documents/College Stuff/WPI/CS541/HW4/New folder/homework4_tamishra_anisal.py"
Gradient Check:
2.4632077529957962e-06


Hidden Layers:  3 Neurons in each layer:  64 Batch_size= 16
Learning Rate:  0.01 Alpha:  1e-06 Epochs:  70
Epoch no. 0
Loss:  0.6439679579053116 Accuracy:  76.00833333333334
Epoch no. 1
Loss:  0.5092111516426224 Accuracy:  81.96666666666667
Epoch no. 2
```

Gradient Check

homework4_tamishra_anisal.py > findBestHyperparameters

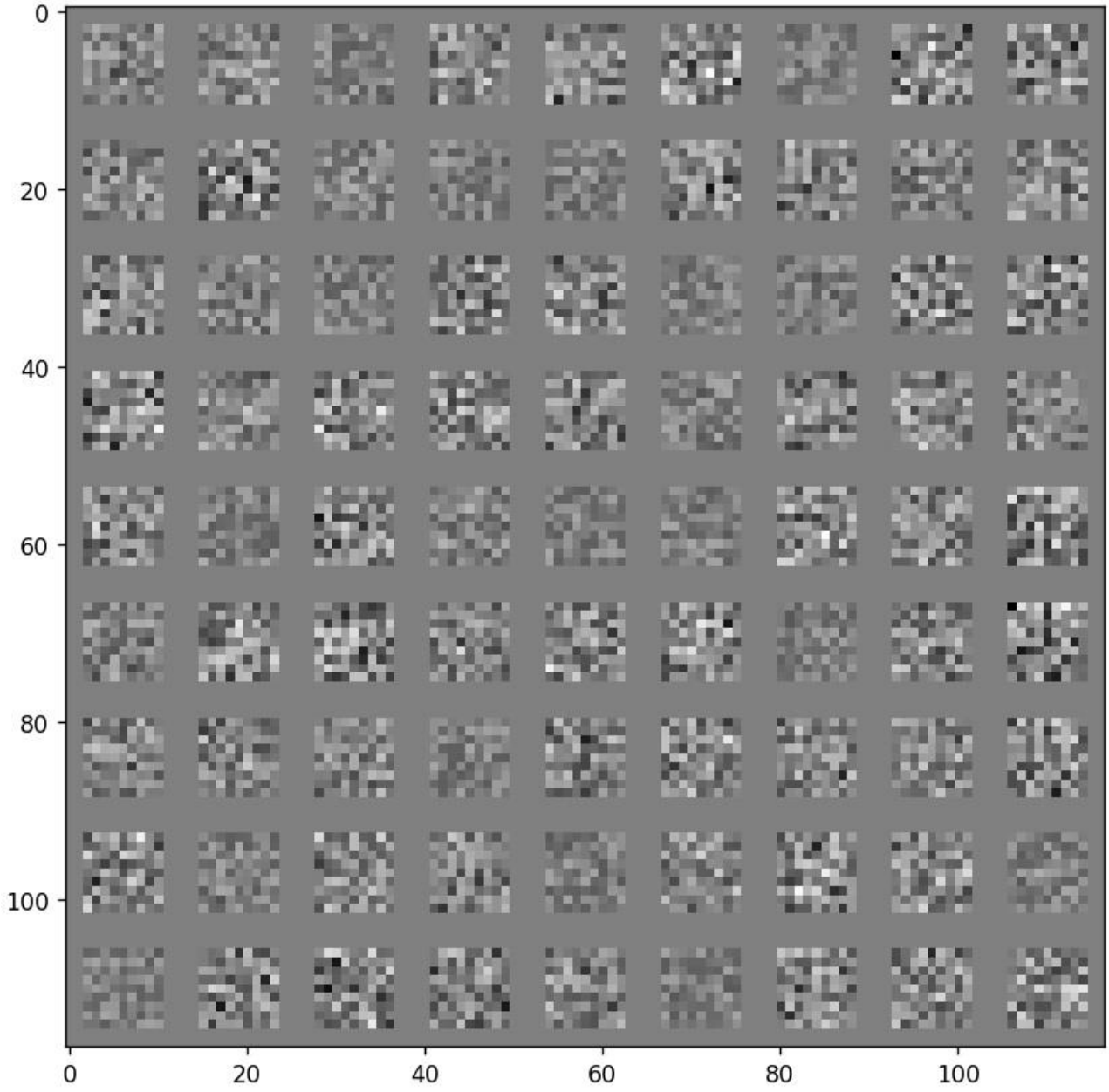
```
    best_epochs = besth[0]  # best number of epochs
    best_batch_size = besth[1]  # best number of batch_size
    best_hidden_num = besth[2]  # best number of hidden neurons
    best_h_layers = besth[3]  # best number of hidden layers
    best_learningrate = besth[4]  # best learning rate
    best_alpha = besth[5]  # best alpha

    weightsAndBiases = initWeightsAndBiases(best_hidden_num, best_h_layers)
    weightsAndBiases=copy.deepcopy(bestw)
    loss, h_h, h_z, yhat = forward_prop(testX, testY, weightsAndBiases, best_hidden_num, best_h_layers)

    print("\nThe Best HyperParameters:  \nHidden Layers:", best_h_layers, "\nHidden Layer Neurons: ", best_hidden_num,
          "\nEpochs: ", best_epochs, "\nBatch size: ", best_batch_size)
    print("Learning rate: ", best_learningrate, "\nAlpha: ", best_alpha)
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Python
Python

```
Loss:  0.612957024217922 Accuracy:  77.63333333333334
Epoch no. 1
Loss:  0.49761346621199776 Accuracy:  82.1
Epoch no. 2
Loss:  0.45303496480016675 Accuracy:  83.94166666666666
Epoch no. 3
Loss:  0.4258982726267441 Accuracy:  85.03333333333333
Epoch no. 4
Loss:  0.4067922182723614 Accuracy:  85.6
Epoch no. 5
Loss:  0.39024930229295923 Accuracy:  86.24166666666666
Epoch no. 6
Loss:  0.37759410068489463 Accuracy:  86.675
Epoch no. 7
Loss:  0.368900839349503 Accuracy:  86.83333333333333
Epoch no. 8
Loss:  0.363024347769612 Accuracy:  87.025
Epoch no. 9
Loss:  0.3557897101714316 Accuracy:  87.25
Epoch no. 10
Loss:  0.35233104370306584 Accuracy:  87.375
Epoch no. 11
Loss:  0.3511380542168628 Accuracy:  87.30833333333334
Epoch no. 12
Loss:  0.3458163661724052 Accuracy:  87.54166666666667
Epoch no. 13
Loss:  0.3448192707985614 Accuracy:  87.75833333333334
Epoch no. 14
```

Iterations



Visualisation of $W^{(1)}$ with 81 neurons in each hidden layer