ECE 421 Programming Assignment Question

Member One: Mark Qi

Student ID: 1006764645

Member Two: Richard Zhao

Student ID: 1006750614

**Part 3 Questions:**

**Experiment 1 Code**

# Experiment 1

def compare\_arch():

# Train Model

model\_experiment\_1\_CNN, acc\_hist\_experiment\_1\_CNN = experiment(model\_type = 'CNN', learning\_rate = 0.0001, dropout\_rate = 0.0, weight\_decay = 0.0, num\_epochs = 50, verbose = False)

model\_experiment\_1\_FNN, acc\_hist\_experiment\_1\_FNN = experiment(model\_type = 'FNN', learning\_rate = 0.0001, dropout\_rate = 0.0, weight\_decay = 0.0, num\_epochs = 50, verbose = False)

# Plot Result

plot(acc\_hist\_experiment\_1\_CNN)

plot(acc\_hist\_experiment\_1\_FNN)

# Return Result

return model\_experiment\_1\_CNN, acc\_hist\_experiment\_1\_CNN, model\_experiment\_1\_FNN, acc\_hist\_experiment\_1\_FNN

**Experiment 1 Model Result**

|  |  |  |
| --- | --- | --- |
|  | Training Accuracy Plot | Testing Accuracy Plot |
| CNN |  |  |
| FNN |  |  |

**Experiment 1 Conclusion**

From the generate accuracy plots, we can observe that both training and testing accuracy plot for FNN were smoother than CNN. The training and testing accuracy for CNN model fluctuated a lot.

**Experiment 2 Code**

# Experiment 2

def compare\_dropout():

# Variable Declaration

dropout\_list = [0.5, 0.8, 0.95]

model\_list = []

acc\_hist\_list = []

# Train Model

for dropout\_rates in dropout\_list:

model, acc\_hist = experiment(model\_type = 'CNN', learning\_rate = 0.0001, dropout\_rate = dropout\_rates, weight\_decay = 0.0, num\_epochs = 50, verbose = False)

model\_list.append(model)

acc\_hist\_list.append(acc\_hist)

# Plot Result

for acc\_hist in acc\_hist\_list:

plot(acc\_hist)

# Return Result

return model\_list, acc\_hist\_list

**Experiment 2 Model Result**

|  |  |  |
| --- | --- | --- |
|  | Training Accuracy Plot | Testing Accuracy Plot |
| CNN Dropout rate = 0.5 |  |  |
| CNN Dropout rate = 0.8 |  |  |
| CNN Dropout rate = 0.95 |  |  |

**Experiment 2 Conclusion**

CNN with dropout rate of 0.5 and 0.8 performed both resulted in smooth training and testing accuracy plot, while CNN with dropout rate of 0.95 resulted in fluctuating training and testing accuracy plot.

CNN with dropout rate of 0.5 had a slightly higher training and testing accuracy rate than CNN with dropout rate of 0.8.

Thus, we conclude that CNN with dropout rate of 0.5 is the best model and the higher the dropout rate, the more fluctuation in accuracy and lower final accuracy.

**Experiment 3 Code**

# Experiment 3

def compare\_l2():

# Variable Declaration

weight\_decay\_list = [0.1, 1.0, 10.0]

model\_list = []

acc\_hist\_list = []

# Train Model

for weight\_decays in weight\_decay\_list:

model, acc\_hist = experiment(model\_type = 'CNN', learning\_rate = 0.0001, dropout\_rate = 0.0, weight\_decay = weight\_decays, num\_epochs = 50, verbose = False)

model\_list.append(model)

acc\_hist\_list.append(acc\_hist)

# Plot Result

for acc\_hist in acc\_hist\_list:

plot(acc\_hist)

# Return Result

return model\_list, acc\_hist\_list

**Experiment 3 Model Result**

|  |  |  |
| --- | --- | --- |
|  | Training Accuracy Plot | Testing Accuracy Plot |
| CNN Weight decay = 0.1 |  |  |
| CNN Weight decay = 1.0 |  |  |
| CNN Weight decay = 10.0 |  |  |

**Experiment 3 Conclusion**

CNN with weight decay of 0.1performed both resulted in smooth training and testing accuracy plot, while CNN with weight decay of 1.0 and 10.0 resulted in much lower training and testing accuracy plot. The accuracy plummeted to 0.10 which is the same as random guesses.

Thus, we conclude that CNN with weight decay of 0.1 is the best model and the higher the weight decay, the accuracy will get worse.