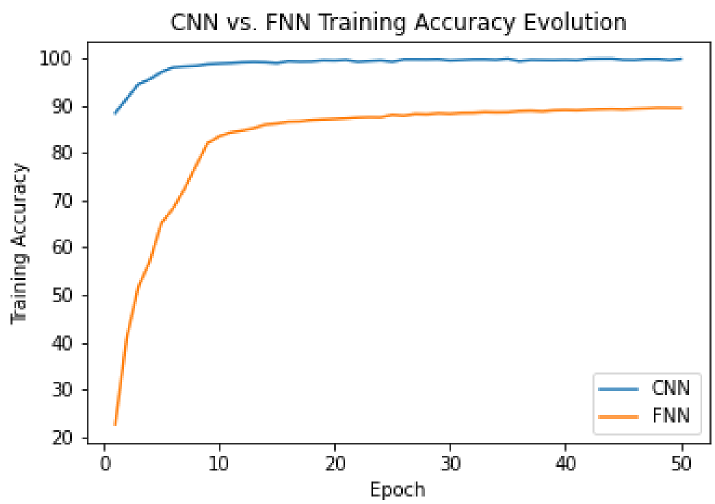
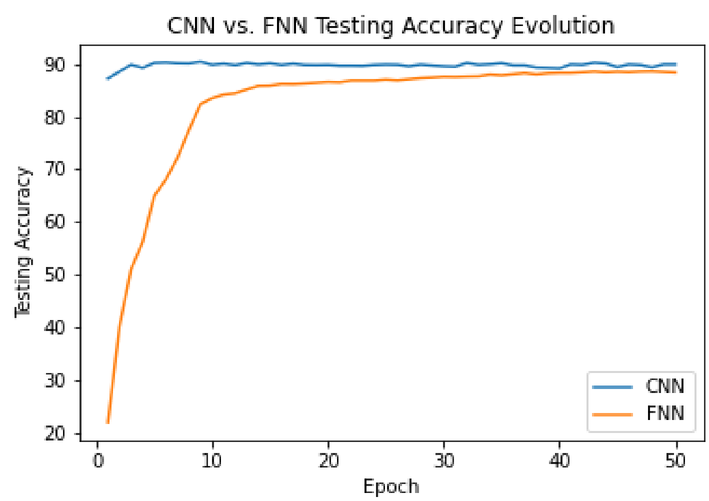
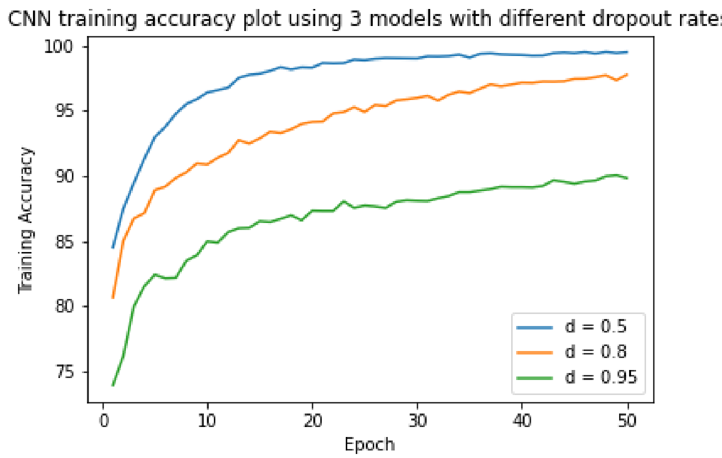
Experiment 1: In this experiment, you will compare the performance between CNN and FNN in the image recognition task. Train your CNN and FNN model for a batch size of 32 (default), for 50 epochs (default) and the AdamW optimizer (Link to AdamW) for learning rate of 0.0001. Set the dropout rate=0.0 and weight decay=0.0. Plot and compare the training/testing history of both models. The function name for this experiment will be compare arch().

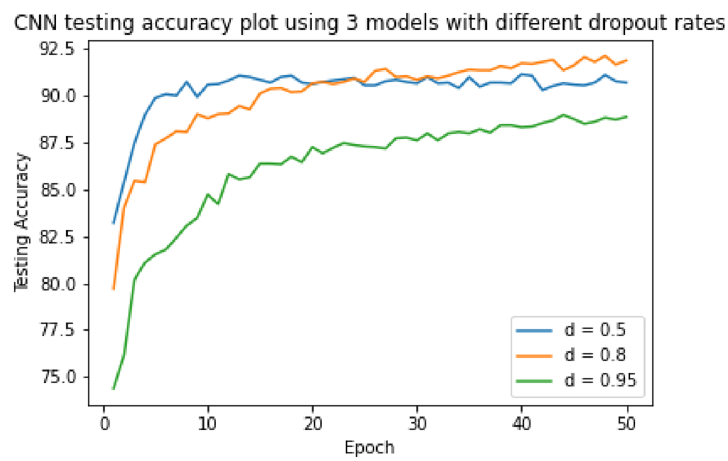




For the training set, CNN outperforms FNN by roughly 10%. CNN also starts off with a better accuracy and converges faster. For the testing set, CNN outperforms FNN with a smaller margin compared to the training set. It also starts off with a better accuracy and converges faster. This is because CNN captures spatial information, therefore it is more advantageous when dealing with image input.

Experiment 2: In this experiment, you will study the effects of dropout rate on your CNN model. Train your CNN model for a batch size of 32 (default), for 50 epochs (default) and the AdamW optimizer (Link to AdamW) for learning rate of 0.0001. Fix your weight decay=0.0 and train 3 different models with dropout rate=0.5,0.8,0.95 respectively. Plot and compare the training/testing history of these three models. Explain the influence of dropout rate to the model's accuracy. The function name for this experiment will be compare dropout().

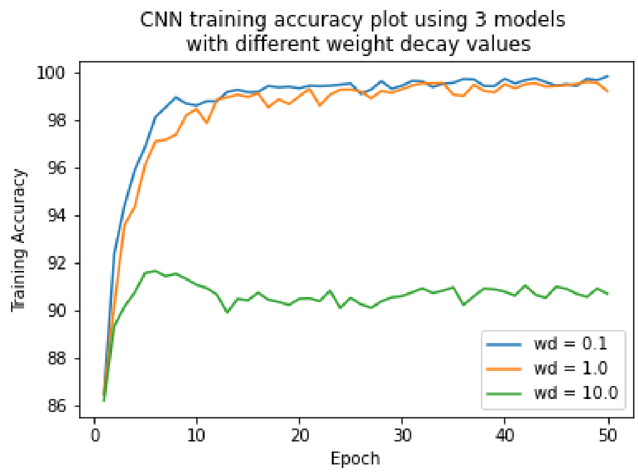


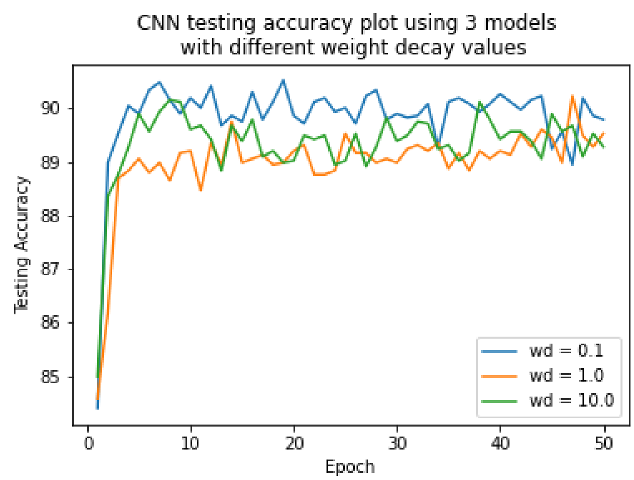


For the training set, smaller dropout rates result in higher training accuracies (d = 0.5 highest, d = 0.95 lowest). This intuitively makes sense because the less nodes within the neural network (neurons) we drop, the more information we capture on the training data, thus resulting in better accuracy.

For the testing set, a dropout rate of 0.8 performs the best in the end and a dropout rate of 0.95 performs the worst. A dropout rate of 0.95 performs the worst because it zeros-out too many nodes in the neural network, resulting in an overly simplistic model which failed to capture key features of new input. A dropout rate of 0.5 performs worse than 0.8 in the testing set even though 0.5 outperforms 0.8 in the training set. A dropout rate of 0.5 seems to be not sufficiently high for this dataset, resulting in a model that likely ends up overfitting the training set.

Experiment 3: In this experiment, you will study the effects of weight decay on your CNN model. Train your CNN model for a batch size of 32 (default), for 50 epochs (default) and the AdamW optimizer (Link to AdamW) for learning rate of 0.0001. Fix your drop rate=0.0 and train 3 different models with weight decay=0.1,1.0,10.0 respectively. Plot and compare the training/testing history of these three models. Explain the influence of L2 regularization rate to the model's accuracy. The function name for this experiment will be compare l2().





L2 regularization penalizes models with large weight values. If the weight decay value is too high, it may lead to underfitting by overly constraining the model, resulting in reduced accuracy. On the other hand, if the weight decay value is too low, the model may still overfit to the training data and lead to reduced accuracy on new data.

For the training set, weight decay values of 0.1 and 1.0 perform similarly on the accuracy without a significant difference. However, a weight decay value of 10.0 performs roughly 10% worse.

For the testing set, all three weight decay values perform similarly. It seems like these three particular weight decay values do not have significant effect on the testing accuracy for the dataset.