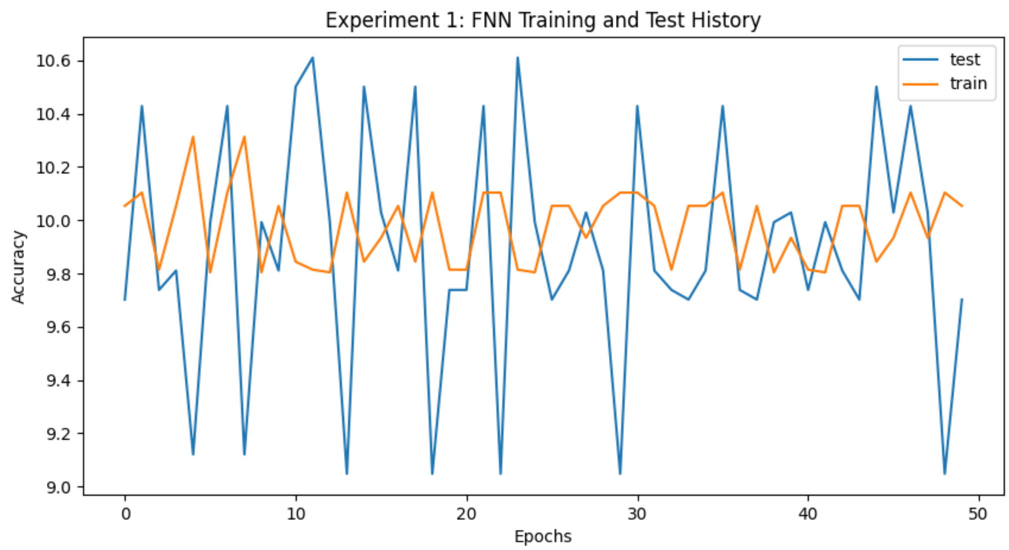
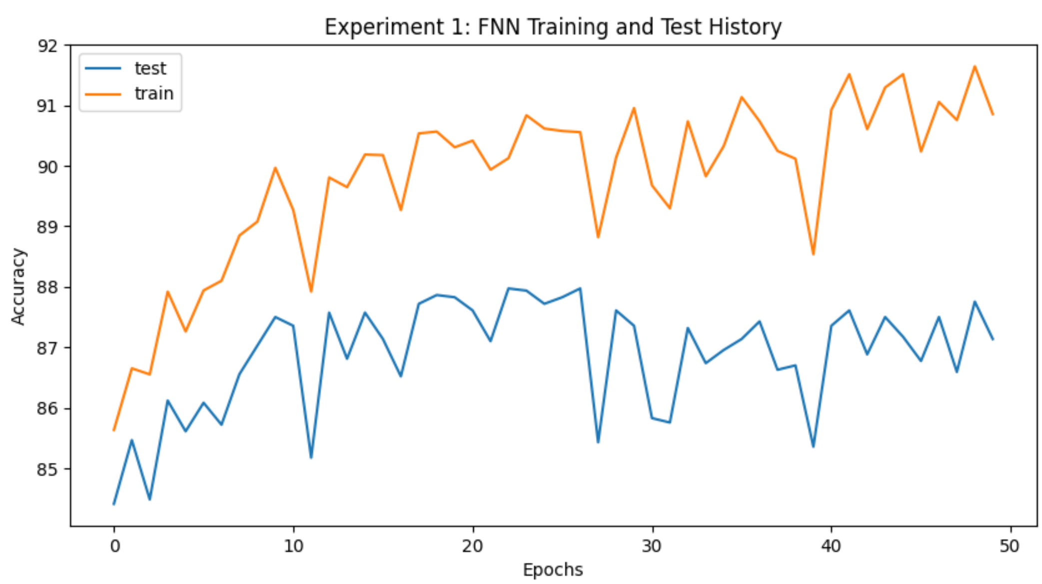
QA3 Testing/Training Model

Experiment: In this experiment, you will train the FNN in the image recognition task. Train your FNN model for a batch size of 32 (default) for 50 epochs (default) and the AdamW optimizer (Link to AdamW) for learning rate of 0.1, 0.01, 0.001, 0.0001. Plot and compare the training/testing history of these models, namely we want to see how the training progresses with different learning rates. The function name for this experiment will be *compare_lr()*. Comment on what you observed from the experiment and explain.

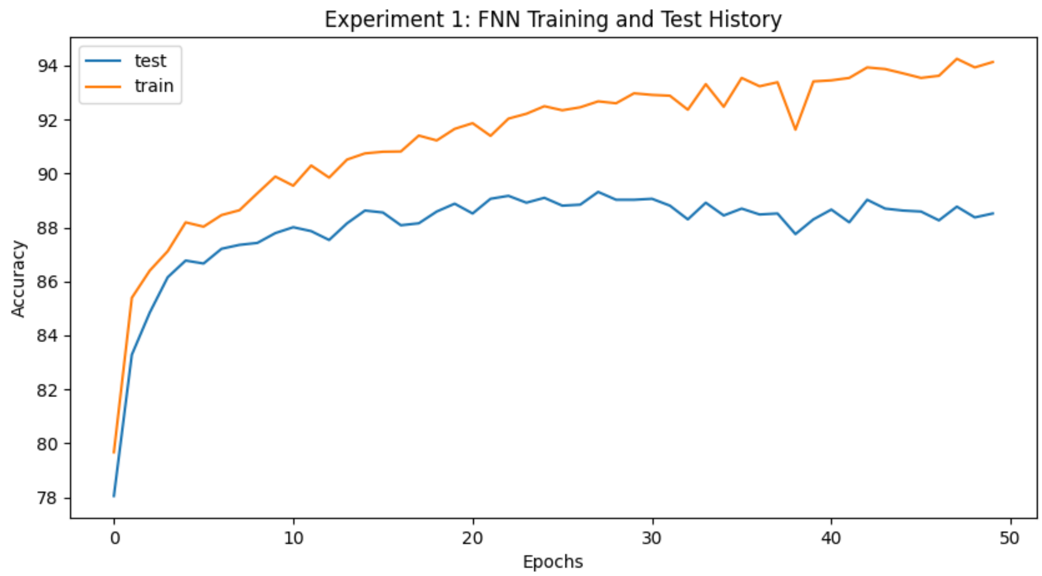
<u>FNN for Learning Rate of 0.1</u>



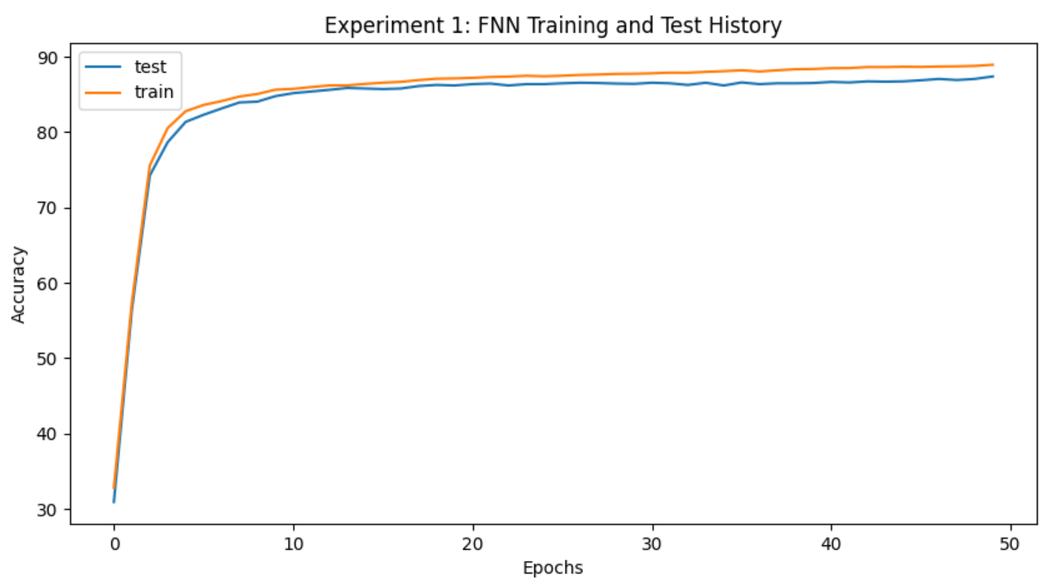
<u>FNN for Learning Rate of 0.01</u>



FNN for Learning Rate of 0.001



FNN for Learning Rate of 0.0001



Evaluation: The learning rate directly affects the step size taken during optimization, with **higher** rates causing **larger** updates and potentially overshooting the optimal solution, while **lower** rates result in **slower** convergence but may yield more **precise** solutions, emphasizing the importance of selecting an appropriate learning rate to balance between convergence speed and accuracy in training neural networks.

Observations:

1. Learning Rate = **0.1**:
   - Training accuracy tends to be very high initially but decreases rapidly over epochs, indicating overshooting or divergence.
   - Testing accuracy starts low and decreases further, suggesting the model's poor generalization.

2. Learning Rate = **0.01**:
   - Both training and testing accuracies increase steadily over epochs, indicating a stable learning process.
   - The model seems to find a good balance between learning from the training data and generalizing to unseen data.

3. Learning Rate = **0.001**:
   - Training accuracy increases gradually, indicating the model is learning effectively from the training data.
   - Testing accuracy also increases but at a slower pace, suggesting a slightly conservative learning approach.

4. Learning Rate = **0.0001**:
   - Training accuracy increases very slowly, indicating a slow learning process.
   - Testing accuracy also increases slowly, but the model might not reach its full potential due to the small learning rate.

Overall:
- A learning rate of 0.01 seems to provide the best balance between fast convergence and good generalization.
- Learning rates that are too high (e.g., 0.1) lead to overshooting and poor generalization.
- Learning rates that are too low (e.g., 0.0001) result in slow convergence, which might not be efficient for training.
- The choice of learning rate significantly impacts the training dynamics and the model's final performance. It's important to tune the learning rate carefully to achieve optimal results.