1. Play around with different Leaky ReLU slopes. What is the best slope you could find? What happens if you set the slope > 1? What about slope < 0. Theoretically, what happens if you set slope = 1?

Leaky ReLU takes advantages of avoiding zero-slope parts and making training faster. Through several testing, I found that Leaky ReLU may not be always consistent when comparing to ReLU. I found for some slope values the accuracy increases at first but drops very quick due to the abnormal cost. When 0 < slope < 0.000001 and even slope < 0, i.e. small slopes, the epoch accuracy becomes more reasonable. When slope > 1, the accuracy is low with huge loss. Among several Leaky ReLU slopes, the best slope among my testing is -0. 000001. Theoretically, when the slope is set to be 1, the Leaky ReLU function will always return the input no matter how it is compared with value 0.

2. Set PReLU to take 1 slope per layer. After 20 epochs, what were your PReLU slopes? Does this correspond with what you found in question 1?

My initital slope is set to 0.01. After 20 epochs, the last PReLU slope is -0.022. Comparing with the findings in question 1, I noticed that the slopes are updated in a negative trending. Since PReLU is a general case of Leaky ReLU but in which slopes are not fixed, then it can make sense that slopes are converging to negative values.

3. If you add more layers and more epochs, what accuracy can you reach? Can you get to 99%? What is your best network layout?

Linear Layer 1000 → Linear Layer 100 → final classification, 50 epochs, 94.2%
Linear Layer 1000 → Linear Layer 100 → final classification, 100 epochs, 95.2%
Linear Layer 1000 → Linear Layer 500 → Linear Layer 100 → final classification, 50 epochs, 95.3%
Linear Layer 1000 → Linear Layer 500 → Linear Layer 100 → final classification, 100 epochs, 96.5%
……

Among several testing, my best network is as following:

1. A full connected layer with 1000 output
2. A full connected layer with 500 output
3. A full connected layer with 100 output
4. A final classification layer

I set the number of epochs to 100. It came with a final accuracy of 96.5%. I found that adding more epochs increases the final accuracy and adding more layers makes the accuracy converge faster. It is possible to get the accuracy of 99% once the number of epochs is larger, because the convergence of accuracy become slower as the number of epochs increases.