1. Pay 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?

The best slope I have is 0.1. When I have slope > 1, I ran into overflow problem. When I have slope < 0, I still get similar performance as slope > 0.

Theoretically, if slope =1, then the neural network is just solving a linear system of equations WX+b = Y.

Leaky ReLU with momentum optimizer

Alpha (layer1)	Alpha (layer2)	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
0.01	0.01	0.888	0.913	0.922	0.930	0.934
0.05	0.05	0.891	0.915	0.925	0.931	0.934
0.1	0.1	0.889	0.911	0.921	0.929	0.934
0.2	0.2	0.891	0.911	0.921	0.926	0.932
0.9	0.9	0.898	0.909	0.913	0.915	0.917
0.3	0.2	0.891	0.912	0.921	0.927	0.932
-0.01	-0.01	0.892	0.912	0.925	0.931	0.936
-0.1	-0.1	0.890	0.915	0.926	0.933	0.940

Leaky ReLU with sgd optimizer

Alpha (layer1)	Alpha (layer2)	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
0.01	0.01	0.703	0.805	0.842		
0.1	0.1	0.708	0.811	0.841		
1.0	1.0	0.793	0.842	0.862		

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?

After 20 epochs, PReLU slope for layer 1 is 0.3 and for layer 2 is 0.2.
Not the same as question 1.

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?

No matter what I do, I could not surpass 97% accuracy it plateaus at that accuracy. The best network is 96.7% and the layout:

```
LinearLayer(28 * 28, 1000),
ReLULayer(),
LinearLayer(1000, 784),
ReLULayer(),
LinearLayer(784, 250),
ReLULayer(),
LinearLayer(250, 100),
ReLULayer(),
LinearLayer(100, 10),
```