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?

   I find that slope between 0.8 to -0.7 is good to get higher accuracy. And they still have non-linear property. If slope is greater than 1, the df of negative part is bigger than positive.

	slope=1.5				slope=-0.8			
	train accuracy	train loss	epoch accuracy	epoch loss	train accuracy	train loss	epoch accuracy	epoch loss
epoch1	0.81	0.867	0.904	0.368	0.726	1.346	0.909	0.316
epoch2	0.905	0.344	0.913	0.306	0.922	0.263	0.931	0.233
epoch3	0.914	0.304	0.919	0.287	0.94	0.2	0.941	0.195
epoch4	0.92	0.284	0.924	0.271	0.95	0.167	0.944	0.179
epoch5	0.924	0.268	0.927	0.26	0.957	0.143	0.947	0.169
epoch6	0.928	0.256	0.93	0.25	0.963	0.124	0.953	0.154
epoch7	0.931	0.245	0.933	0.241	0.968	0.11	0.955	0.143
epoch8	0.934	0.236	0.934	0.234	0.972	0.098	0.957	0.138
epoch9	0.936	0.227	0.936	0.227	0.975	0.087	0.958	0.134
epoch10	0.938	0.218	0.938	0.22	0.978	0.078	0.958	0.133
epoch11	0.94	0.21	0.939	0.214	0.98	0.071	0.959	0.131
epoch12	0.942	0.202	0.94	0.208	0.982	0.064	0.961	0.129
epoch13	0.945	0.195	0.941	0.203	0.984	0.059	0.962	0.128
epoch14	0.947	0.188	0.943	0.197	0.986	0.054	0.962	0.126
epoch15	0.949	0.182	0.945	0.192	0.987	0.05	0.963	0.125
epoch16	0.951	0.176	0.946	0.186	0.988	0.047	0.963	0.124
epoch17	0.953	0.17	0.947	0.182	0.989	0.043	0.964	0.122
epoch18	0.954	0.164	0.948	0.177	0.991	0.04	0.964	0.12
epoch19	0.955	0.159	0.95	0.172	0.992	0.037	0.965	0.116
epoch20	0.957	0.154	0.951	0.168	0.993	0.034	0.966	0.114

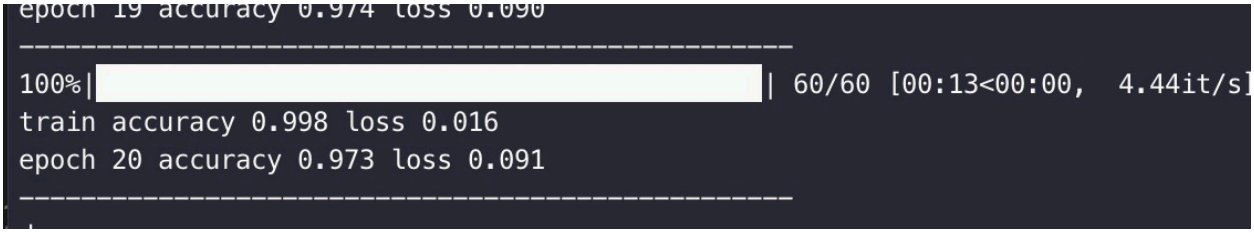
   If the slope is 1, it is a linear layer.

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 run it after 20 times, i find that the first layer should set as 0.47, the second should set as the -0.63.

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?

   I have try to add one linear layers and one Relu layer. As a result, after running 20 epoch, I get: (lr =0.02)



   It increases a little accuracy (from about 96.5% to 97.3%). I don't think deep layer of neural network will help us find a higher accuracy. Because the deeper neural network should to deal with the error vanish and error explosion.

   For me, the best network layout by using the 3 linear layer and 2 Leaky ReLU layer(one slope is 0.47, another is -0.63) with learning rate of 0.05 after 28 epochs will do the best network layout. The accuracy of the train set is 1 and the accuracy of the test set is 0.98