Machine Learning HW3 Report

108062135 呂佳恩

Basic

Problems that I encountered:

1. Understanding the idea of this time's homework was easy, however, it was hard for me to put my thoughts into code.
2. There were some notes when encountering the Sigmoid, Softmax implementation, however, I found out that not all were necessary to implement in this homework to make the code run. For me, I used the numeric stable sigmoid function which was mentioned in the note but the naïve implementation for the softmax function.
3. There were different methods to which achieved similar results such as
   np.random.normal() and np.random.randn
   np.dot and np.matmul
   I had to read articles to understand the differences

How I built the binary classifier:

Following the instructions, we do the following steps

1. First initialize the parameters
2. We have to implement the activation functions
3. Build the Cost function
4. Build the Forward model
5. Build the backward model
6. We have to update the parameters according to the results

Extra Efforts to implement my model:

I tried different iterations for the model in order to accomplish the highest accuracy.

Summary:

The homework was relatively easy after I understood the thought process of forward/backward propagation.

I read some articles online about different methods and libraries used in the implementations.

The biggest problem was the training time was so long for models with multiple layers and nodes.

BONUS

Extra Efforts to implement my model:

I set the learning rate to 0.0075 since if the learning rate is too big, it might overshoot or diverge, if the learning rate is small, we can overcome this with more iterations.

I slip the data into a 80-20 split and use the accuracy of the validation data as the evaluation method. The higher the accuracy is, the better. It is shown in the image below

	Learning Rate	Layer_dims	iterations	Accuracy
6 layers				
	0.0075	[64, 100, 100, 100, 10	40000	0.8502
	0.0075	[64, 100, 100, 100, 10	30000	0.85097
	0.0075	[64, 100, 100, 100, 10	25000	0.85246
	0.0075	[64, 100, 100, 100, 10	20000	0.8532
	0.0075	[64, 100, 100, 100, 10	15000	0.84724
	0.0075	[64, 200, 200, 200, 20	30000	0.8763
	0.0075	[64, 200, 200, 200, 20	25000	0.87556
	0.0075	[64, 200, 200, 200, 20	20000	0.87481
	0.0075	[64, 300, 300, 300, 30	25000	0.87034
	0.0075	[64, 300, 300, 300, 30	10000	0.86438
5 layers				
	0.0075	[64, 300, 200, 100, 4]	40000	0.889717
	0.0075	[64, 100, 100, 100, 4]	45000	0.89121
	0.0075	[64, 100, 100, 100, 4]	40000	0.891207
	0.0075	[64, 100, 100, 100, 4]	30000	0.88599
	0.0075	[64, 100, 100, 100, 4]	25000	0.8845
	0.0075	[64, 200,200, 200, 4]	30000	0.87407
4 Layers				
	0.0075	[64, 300, 300, 4]	70000	0.90611
	0.0075	[64, 300, 300, 4]	60000	0.90462
	0.0075	[64, 300, 300, 4]	55000	0.90164
	0.0075	[64, 300, 300, 4]	45000	0.90089
	0.0075	[64, 300, 300, 4]	40000	0.90015
	0.0075	[64, 300, 300, 4]	35000	0.8994
	0.0075	[64, 300, 300, 4]	30000	0.89791
	0.0075	[64, 100, 100, 4]	70000	0.90537
	0.0075	[64, 100, 100, 4]	30000	0.90015
	0.0075	[64, 500, 500, 4]	70000	0.91729