HW4

Questions

2.1.1 What are the training and test accuracy values you get? Why might we be interested in both training accuracy and testing accuracy? What do these two numbers tell us about our current model?

I got 90.34% for training accuracy and 90.91 for test accuracy.

We want to know how well the model is for the training data and test data. If the training accuracy is largely higher than test accuracy, this might imply there is overfitting.

2.1.2 Try varying the model parameter for learning rate to different powers of 10 (i.e. 10^1, 10^0, 10^-1, 10^-2, 10^-3) and training the model. What patterns do you see and how does the choice of learning rate affect both the loss during training and the final model accuracy?

Learning rate	Training accuracy	Test accuracy
0.1	92.07%	91.71%
0.01	90.34%	90.91%
0.001	85.90%	86.69%
0.0001	73.21%	74.25%

The higher learning rate achieves higher accuracy for both training and test. A lower learning rate might cause insufficient updates to the model weights while training.

2.1.3 Try varying the parameter for weight decay to different powers of 10: (10^0, 10^-1, 10^-2, 10^-3, 10^-4, 10^-5). How does weight decay affect the final model training and test accuracy?

Decay	Training accuracy	Test accuracy
0.1	90.31%	90.89%
0.01	90.34%	90.92%
0.001	90.34%	90.91%

0.0001	90.34%	90.91%

Reduce the weight decay just slightly affect the test accuracy. However, increase the weight decay will lead to decrease on both training and test accuracies.

2.2.1 Currently the model uses a logistic activation for the first layer. Try using a the different activation functions we programmed. How well do they perform? What's best?

	Training accuracy	Test accuracy
Logistic	88.93%	89.49%
RELU	92.61%	92.81%
LRELU	92.43%	92.63%

RELU perform the best on training accuracy of 92.61% and the test accuracy of 92.81%.

2.2.2 Using the same activation, find the best (power of 10) learning rate for your model. What is the training accuracy and testing accuracy?

Learning rate	Training accuracy	Test accuracy
0.1	95.09%	94.39%
0.01	92.61%	92.81%
0.001	86.70%	87.66%
0.0001	52.80%	52.82%

When learning rate is 0.1 perform the best. Training accuracy is 95.09% and test accuracy is 94.39%.

2.2.3 Right now the regularization parameter `decay` is set to 0. Try adding some decay to your model. What happens, does it help? Why or why not may this be?

Decay	Training accuracy	Test accuracy
0	92.61%	92.81%
0.1	92.48%	92.75%
0.01	92.59%	92.80%

0.001	92.61%	92.82%

With adding the decay values seems just slightly decrease both the accuracies. The decay value can prevent the model from overfitting.

2.2.4 Modify your model so it has 3 layers instead of two. The layers should be `inputs -> 64`, `64 -> 32`, and `32 -> outputs`. Also modify your model to train for 3000 iterations instead of 1000. Look at the training and testing error for different values of decay (powers of 10, 10^-4 -> 10^0). Which is best? Why?

Decay	Training accuracy	Test accuracy
0	97.19%	96.47%
0.1	96.89%	96.28%
0.01	97.20%	96.45%
0.001	97.21%	96.50%

The decay value with 0.001 performs the best since that 0.001 offers the best compromise in managing the bias variance trade off.

3.1.1 What is the best training accuracy and testing accuracy? Summarize all the hyperparameter combinations you tried.

Learning rate	Decay	Training accuracy	Test accuracy
0.1	0.001	18.21%	18.08%
0.01	0.001	46.25%	44.47%
0.001	0.001	39.94%	39.25%
0.01	0	46.01%	44.57%
0.01	0.1	46.03%	44.67%
0.01	0.01	46.35%	44.22%
0.01	0.001	46.25%	44.47%
0.015	0.1	46.73%	45.58%
0.02	0.1	46.08%	44.97%
0.03	0.1	42.51%	41.56%

We can find out that the learning rate of 0.015 and decay with 0.1 perform the best. If we modify the model to train for 30000 iterations instead of 3000, the performance can increase to 52.55% in training accuracy and 48.11% in

test accuracy.