**Peter Dorich**

**Garrett Bauer**

**James May**

**Assignment 3:**

**Part 1:** Since the program took about 10 minutes to finish running, I only had the opportunity to test with 10 epochs, just as the example had done as well. What I noticed is as the learning rate goes down, the loss is far greater. With each additional epoch, up to 5, the loss generally lowered from the previous one. However it is worth noting that these statements are very general, the actual results are far more visible in the graphs below.

As for the average loss, it corresponds with the accuracy. The goal here is to minimize the loss, so the smaller the better. The best time to stop training, obviously, is when the loss begins to grow, or the accuracy starts to decline. Usually this was after about 3 epochs, from what I noticed in the data.

I’d say after examining all the data, the best learning rate would be between 0.1 and 0.001 according to the information. Beyond that the lost was just too great to justify.

**Because of the amount of time it takes to run the program with 10 Epochs, I chose to run them at 5 each to save time. This will clearly limit the amount of information gathered for each Learning rate, but we can at the very least compare them with the information available.**

**Learning Rate = 0.1:**

Accuracy Plot:

1.0

1.5

2.0

2.5

3.0

3.5

4.0

4.5

5.0

34

36

38

40

42

44

46

Loss Plot:

1.0

1.5

2.0

2.5

3.0

3.5

4.0

4.5

5.0

1.60

1.65

1.70

1.75

1.80

1.85

1.90

**Learning Rate: 0.001**

Accuracy Plot:

1.0

1.5

2.0

2.5

3.0

3.5

4.0

4.5

5.0

33

34

35

36

37

38

39

Loss Plot:

1.0

1.5

2.0

2.5

3.0

3.5

4.0

4.5

5.0

1.70

1.75

1.80

1.85

1.90

1.95

**Learning Rate: 0.0001**

Accuracy Plot:

1.0

1.5

2.0

2.5

3.0

3.5

4.0

4.5

5.0

21

22

23

24

25

26

27

28

29

30

Loss Plot:

1.0

1.5

2.0

2.5

3.0

3.5

4.0

4.5

5.0

2.00

2.05

2.10

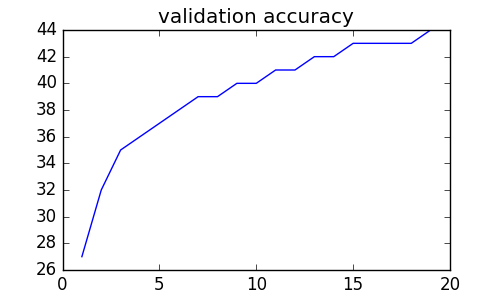
2.15

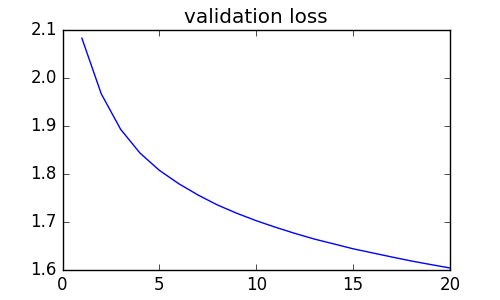
2.20

2.25

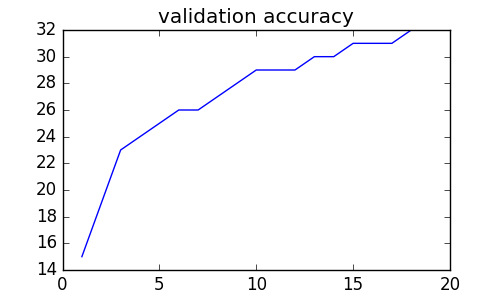
**Part 2:** Using the Relu activation function, it appears that the best value for the learning rate is 0.1. After testing over 20 epochs, this gives the best validation accuracy at 44% as well as the smallest loss at about 1.6. As the learning rate decreases by a factor of 10, the accuracy decreases as well while the average loss increases. I have included plots of accuracy and loss vs. number of epochs below to verify my findings.

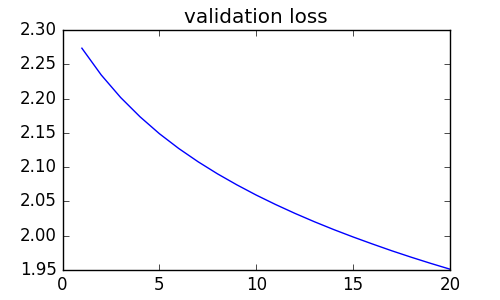
**Learning Rate: 0.1**



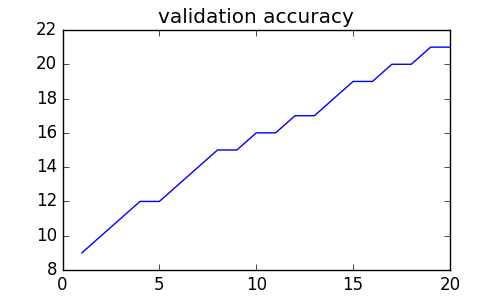


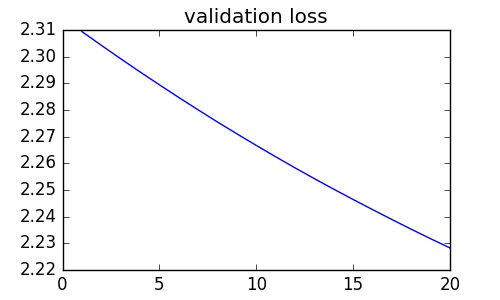
**Learning Rate: 0.01**



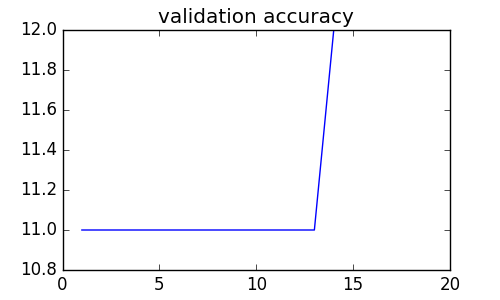


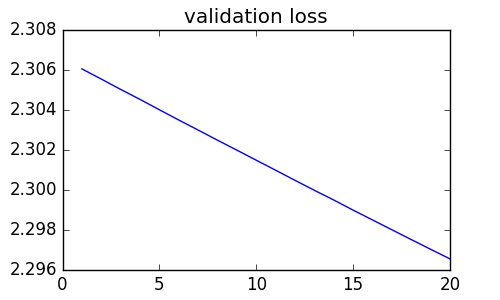
**Learning Rate: 0.001**





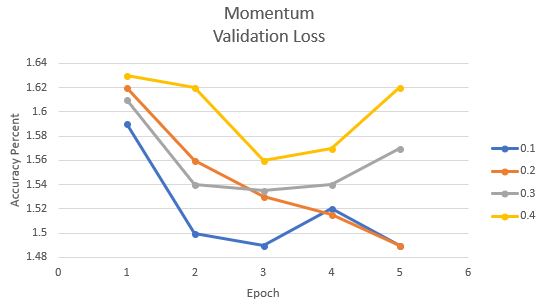
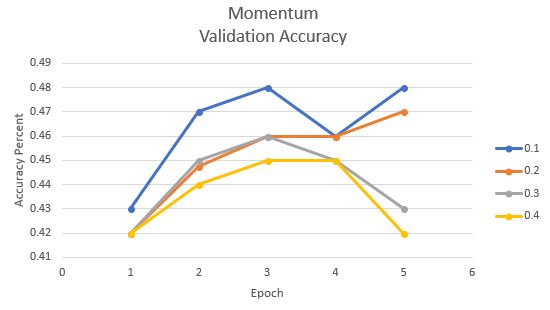
**Learning Rate: 0.0001**



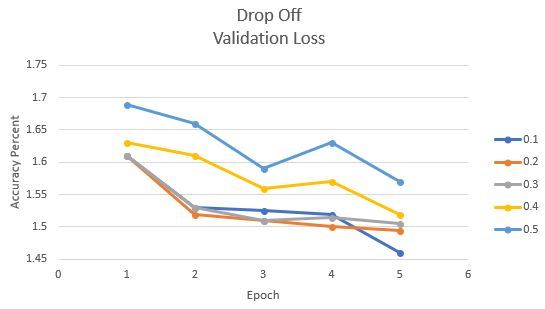
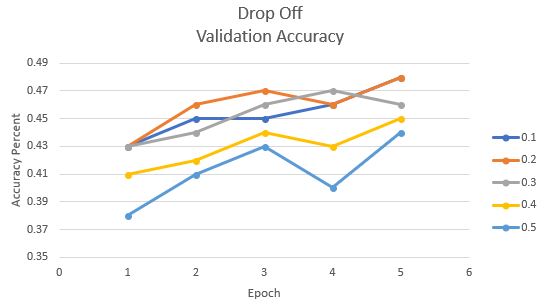


**Part 3:**

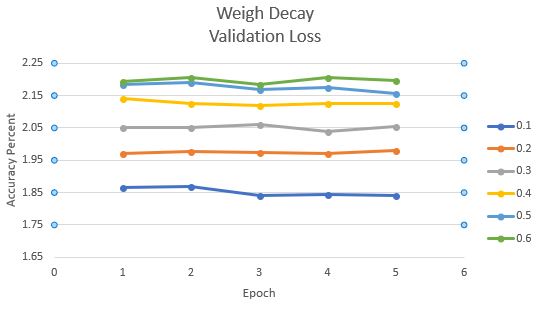
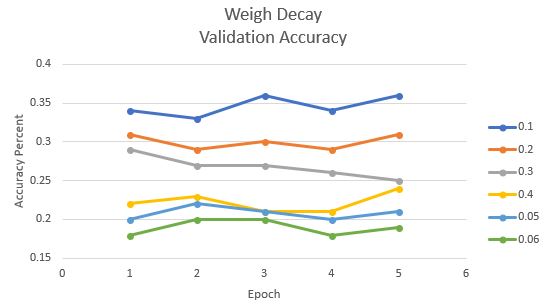
To improve the network efficiency, we varied the momentum, drop-off, and weight decay coefficients. Due to the small amount of influence on each other, each coefficient was varied independently and after graphing each term as seen in the 6 figures below, we concluded that for maximum performance, momentum = .1, drop-off = .1 and the weight decay = .1. The number of epochs was limited to 5 due to time constraints.



The momentum coefficient is more efficient when lower since our learning rate is high and we are more likely so jump over local minima.



The graphs show that after about .2 we start losing too many weights and underfitting becomes an issue.



Weight decay is similar to the momentum in that with a high learning rate, it should remain small.

**Part 4:** For this problem, we used the Relu activation function and a learning rate of 0.1. We noticed that it took a little longer to run using this 3-layer network than the 2-layer network previously, even though they have the same number of nodes. If this is because of random chance or because the 3-layer network is more computationally expensive, we can’t quite say. Comparing against the 2-layer network with the same number of nodes, the 3-layer network is slightly less accurate and has a higher average loss. Given this information for this specific classification problem, we would use the 2-layer network over the 3-layer network. Here are graphs of the accuracy and loss for the 3-layer network using the Relu activation function and a learning rate of 0.1.

