MPCS 51087
Problem Set 3
Machine Learning for Image Classification Milestone 2

Sonia Sharapova

CNetID: sharapova

Changes from previous milestone:

In this millstone, we replaced the Sigmoid activation function with ReLu, implemented Softmax for the last layer, and backpropagated with the cross-entropy loss.

The weights were initialized using the Kaiming initialization scheme.

Parameters:

Number of hidden layers: 1
Dimension of hidden layers: 800
Number of epochs: 5
Batch Size: 256 or 512
Alpha: 0.01

Running the program:

```
$ module load openblas
```

For sigma function:

```
$ gcc -o sig sig_nn.c -lopenblas -lm -O3
$ ./sig 1 800 5 512 0.01
```

For ReLu function:

```
$ gcc -o relu relu_nn.c -lopenblas -lm -O3
$ ./relu 1 800 5 512 0.01
```

Performances:

CPU Version:

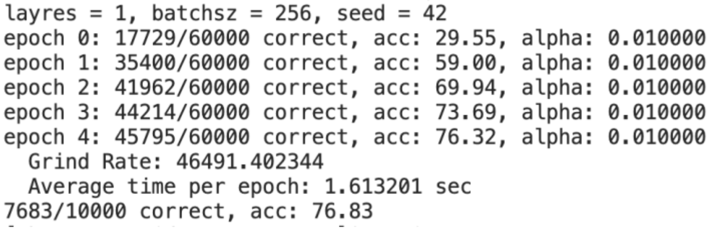
Baseline:

Sigmoid: Manual dot-product

Batch Size 256:

- Grind Rate: 46491.402
- Average time per epoch: 1.61 sec
- Accuracy on test set: 76.83%

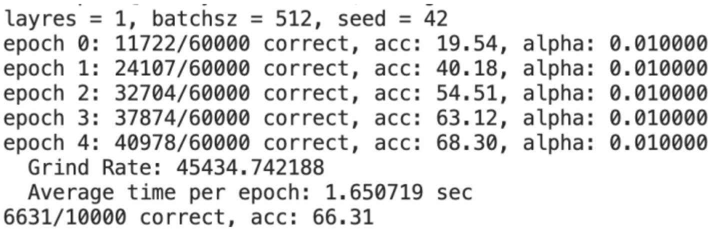
Training accuracies:



```
layres = 1, batchsz = 256, seed = 42
epoch 0: 17729/60000 correct, acc: 29.55, alpha: 0.010000
epoch 1: 35400/60000 correct, acc: 59.00, alpha: 0.010000
epoch 2: 41962/60000 correct, acc: 69.94, alpha: 0.010000
epoch 3: 44214/60000 correct, acc: 73.69, alpha: 0.010000
epoch 4: 45795/60000 correct, acc: 76.32, alpha: 0.010000
  Grind Rate: 46491.402344
  Average time per epoch: 1.613201 sec
7683/10000 correct, acc: 76.83
```

Batch Size 512:

- Grind Rate: 45434.742
- Average time per epoch: 1.65 sec
- Accuracy on test set: 66.31%
-



```
layres = 1, batchsz = 512, seed = 42
epoch 0: 11722/60000 correct, acc: 19.54, alpha: 0.010000
epoch 1: 24107/60000 correct, acc: 40.18, alpha: 0.010000
epoch 2: 32704/60000 correct, acc: 54.51, alpha: 0.010000
epoch 3: 37874/60000 correct, acc: 63.12, alpha: 0.010000
epoch 4: 40978/60000 correct, acc: 68.30, alpha: 0.010000
  Grind Rate: 45434.742188
  Average time per epoch: 1.650719 sec
6631/10000 correct, acc: 66.31
```

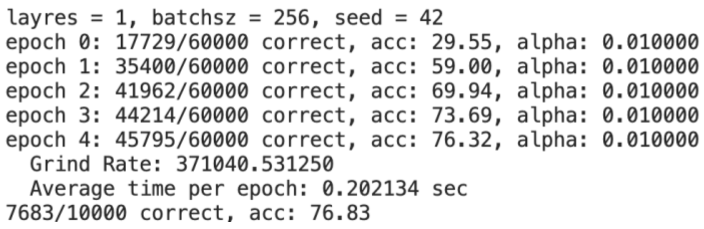
With batch size 512 on 20 epochs:

- Grind Rate: 38254.128906
- Average time per epoch: 1.651009 sec
- Accuracy on test set: 77.90%

Sigmoid: With BLAS

Batch Size 256:

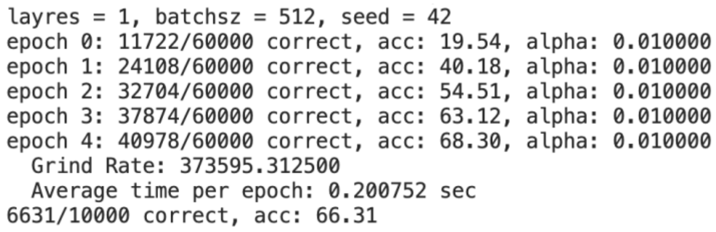
- Grind rate: Grind Rate: 371040.531
- Average time per epoch: 0.202sec
- Accuracy on test set: 76.83%



```
layres = 1, batchsz = 256, seed = 42
epoch 0: 17729/60000 correct, acc: 29.55, alpha: 0.010000
epoch 1: 35400/60000 correct, acc: 59.00, alpha: 0.010000
epoch 2: 41962/60000 correct, acc: 69.94, alpha: 0.010000
epoch 3: 44214/60000 correct, acc: 73.69, alpha: 0.010000
epoch 4: 45795/60000 correct, acc: 76.32, alpha: 0.010000
  Grind Rate: 371040.531250
  Average time per epoch: 0.202134 sec
7683/10000 correct, acc: 76.83
```

Batch Size 512:

- Grind rate: 373595.313
- Average time per epoch: 0.20 sec
- Accuracy on test set: 66.31%



```
layres = 1, batchsz = 512, seed = 42
epoch 0: 11722/60000 correct, acc: 19.54, alpha: 0.010000
epoch 1: 24108/60000 correct, acc: 40.18, alpha: 0.010000
epoch 2: 32704/60000 correct, acc: 54.51, alpha: 0.010000
epoch 3: 37874/60000 correct, acc: 63.12, alpha: 0.010000
epoch 4: 40978/60000 correct, acc: 68.30, alpha: 0.010000
  Grind Rate: 373595.312500
  Average time per epoch: 0.200752 sec
6631/10000 correct, acc: 66.31
```

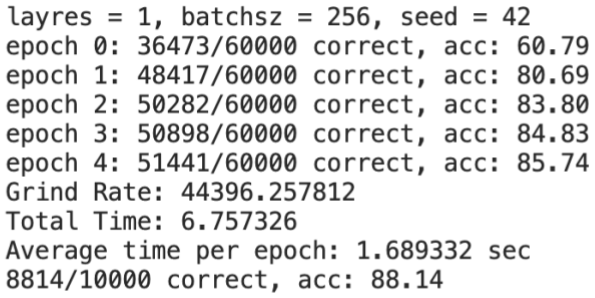
With batch size 512 on 20 epochs:

- Grind Rate: 314197.937
- Average time per epoch: 0.201 sec
- Accuracy on test set: 77.90%

Softmax: Manual dotprod:

Batch Size 256:

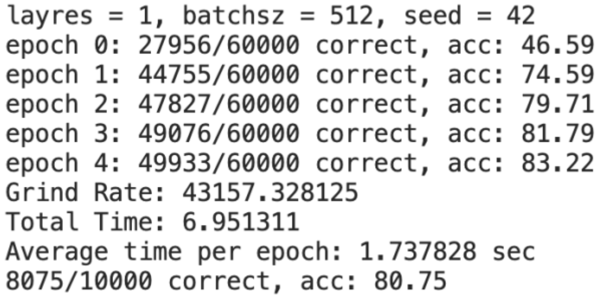
- Grind Rate: 44396.26
- Total Time: 6.76 sec
- Average time per epoch: 1.69 sec
- Accuracy on test set: 88.14%



```
layres = 1, batchsz = 256, seed = 42
epoch 0: 36473/60000 correct, acc: 60.79
epoch 1: 48417/60000 correct, acc: 80.69
epoch 2: 50282/60000 correct, acc: 83.80
epoch 3: 50898/60000 correct, acc: 84.83
epoch 4: 51441/60000 correct, acc: 85.74
Grind Rate: 44396.257812
Total Time: 6.757326
Average time per epoch: 1.689332 sec
8814/10000 correct, acc: 88.14
```

Batch Size 512:

- Grind Rate: 43157.328
- Total Time: 6.95 sec
- Average time per epoch: 1.74 sec
- Accuracy on test set: 80.75%



```
layres = 1, batchsz = 512, seed = 42
epoch 0: 27956/60000 correct, acc: 46.59
epoch 1: 44755/60000 correct, acc: 74.59
epoch 2: 47827/60000 correct, acc: 79.71
epoch 3: 49076/60000 correct, acc: 81.79
epoch 4: 49933/60000 correct, acc: 83.22
Grind Rate: 43157.328125
Total Time: 6.951311
Average time per epoch: 1.737828 sec
8075/10000 correct, acc: 80.75
```

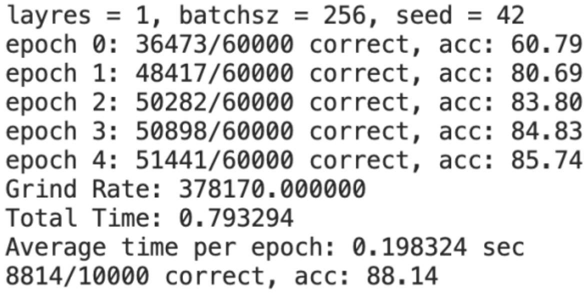
With batch size 512 on 20 epochs:

- Grind Rate: 36243.238
- Total Time: 33.11 sec
- Average time per epoch: 1.74 sec
- Accuracy on test set: 85.69%

Softmax: With BLAS

Batch Size 256:

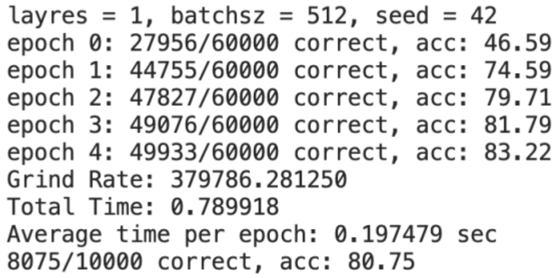
- Grind Rate: 378170.000000
- Total Time: 0.793294
- Average time per epoch: 0.198324 sec
- 8814/10000 correct, acc: 88.14%
-



```
layres = 1, batchsz = 256, seed = 42
epoch 0: 36473/60000 correct, acc: 60.79
epoch 1: 48417/60000 correct, acc: 80.69
epoch 2: 50282/60000 correct, acc: 83.80
epoch 3: 50898/60000 correct, acc: 84.83
epoch 4: 51441/60000 correct, acc: 85.74
Grind Rate: 378170.000000
Total Time: 0.793294
Average time per epoch: 0.198324 sec
8814/10000 correct, acc: 88.14
```

Batch Size 512:

- Grind Rate: 379786.281
- Total Time: 0.789 sec
- Average time per epoch: 0.19 sec
- Accuracy on test set: 80.75%



```
layres = 1, batchsz = 512, seed = 42
epoch 0: 27956/60000 correct, acc: 46.59
epoch 1: 44755/60000 correct, acc: 74.59
epoch 2: 47827/60000 correct, acc: 79.71
epoch 3: 49076/60000 correct, acc: 81.79
epoch 4: 49933/60000 correct, acc: 83.22
Grind Rate: 379786.281250
Total Time: 0.789918
Average time per epoch: 0.197479 sec
8075/10000 correct, acc: 80.75
```

With batch size 512 on 20 epochs:

- Grind Rate: 255689.641
- Total Time: 4.693 sec
- Average time per epoch: 0.247 sec
- Accuracy on test set: 85.69%

CUDA Performance

I was unable to implement a fully working CUDA version, but I will add the version I have now. The issue I came across was flattening my arrays from the serial implementation, but the fix should not be too hard to do. CUDA will be implemented for the matrix-matrix operations using BLAS.

Observations:

- The use of BLAS showed a major speedup on computation for the neural network.
- The use of the ReLu activation function also showed better accuracies when compared to the Sigmoid accuracies.
- Extra: I also found that running on more epochs or with a higher learning rate gave higher accuracies. When running on an alpha of 1 or 0.1, every model reached an accuracy of at least 90%.