Bowei Wang 1462495
cmput466 mini-project final draft:

description of the dataset:

The dataset I have chosen is mnist, which is a very representative dataset for image identification. Mnist is a bunch of hand-written digits which range from 0-9. I think this is a very typical classification problem.

problem specification:

I am curious about the effect of neural network on image identification. Although people usually use convolution neural network which we haven't learn, I would like to try to compare the effect of simple neural network with logistic regression to figure out which one is better.
mini-batch gradient descent is popular in neural network, but I don't think it works better than sgd. I would like to compare the effect of them to see if mini-batch is really the best one.

question asked:

Why does the neural network usually used in image identification? Is neural network works much better than other classification algorithms like simple logistic regression?
How to choose either mini-batch gd or sgd while trying to do some learning?

methods:

The algorithm I would like to compare is: basic logistic regression with sgd, neural network with sgd and neural network with mini-batch gd.

reason of choosing these algorithem:

logistic regression with sgd :Basic logistic regression is one of the most easy and typical method of classification algorithms. It is great to choose this algorithm as the beginning of this problem.

neural network with sgd:Neural network is popular recently. Besides, nn is usually chosen to deal with image identification problems. I would like choose nn with sgd as a comparison with the logistic regression with sgd. By this comparision I can approximately know which algorithm is better on image identification.

neural network with mini-batch gd: While I also need to compare the effect of sgd and mini-batch gd. It is reasonable to use mini-batch gd on the same neural network and compare the effect of them.

design of experiments:

data split: since the mnist already have the test dataset（10000）and training dataset（60000）. I would just split 10000 data from the training as the validation set, therefore we have 50000 data as training set, 10000 data as validation set and 10000 data as test set.

statistical significance tests: While comparing two algorithms, first use the Welch’s t-test in the first 50 epochs to measure which algorithm can learn faster at beginning.
Then use Student’s t-test on 450-500 epochs(where I believe the std is pretty small and can be regarded same as each other after running on some small datasets) to compare the final effect of the algorithm.

My target of using t-test is to find if there is any significant difference between logistic regression and nn, and difference between sgd and mini-batch.

In the first 50 epochs, the accuracy might increasing very fast, so I assume the std between two model is different. In the last 50 epochs, the accuracy shouldn’t change very much, so I assume the std between two model is same and close to 0.

The process should be look like this: For the first 50 epochs, use data from validation set to evaluate the learning ability at the end of each epoch. Use these 50 results to do the Welch’s t-test to measure the learning speed.
Then, for 50-450 epochs, just do simple training. In 450-500 epoch, use the data from validation set to get the accuracy of each algorithm, put the result into the Student’s t-test to measure which one has the better final performance.
After 500 epochs, stop training, put the data from test set into the model, and compare the final result of each model. If nothing works wrong, the result from the test dataset should be same as the result from the t-test.

Specification of neural network:

The neural network for both sgd and mini-batch sgd has same structure. The only difference is the process of updating the weight.

The input layer has size 784, this cannot be changed because the dataset has 784 variables.

The first layer has size 80, this value is the one that I found works pretty good. If I found a better one latter, I may change this value.
The output layer has size 10, this cannot be changed because the mnist dataset has 10 numbers in it(0-9), so there are 10 possible choices. It looks like [0,1,0,0.....0], only one number is one and all the others are all 0.
If predict[index] == 1, then index is the number that we predict the data to be.

The activation of nn is sigmoid for first layer and softmax for output layer.
Actually, both sigmoid and softmax are works well on classification problems. However I think that use softmax as the output layer could learn faster on complex classification problems.

Result and analysis:

	Accuracy at 10 epochs	Accuracy at 50 epochs	Accuracy at 100 epochs	Accuracy at 400 epochs	Accuracy on test dataset
Logistic regression	92.04%	91.79%	92.02 %	91.21 %	91.59%
nn with sgd	89.12 %.	93.04 %.	94.12%	94.54%	94.07%
nn with mini-batch	80.59%	89.81%	91.75%	94.2%	93.67%

For first 50 epochs' Welch's t-test, we have:
statistic=2.5116250232548736, pvalue=0.013652030764066686
While comparing the logistic regression with neural network. That means that we should reject the null hypothesis of equal averages. Logistic regression works better.

statistic=4.284390150562798, pvalue=4.2846658034225385e-05
While comparing the sgd with mini-batch under neural network. That means that we should reject the null hypothesis of equal averages. SGD works better.

For last 50 epochs' Student's t-test, we have:
statistic=-81.27862590128701, pvalue=9.713622995246992e-92
While comparing the logistic regression with neural network. That means that we should reject the null hypothesis of equal averages. Neural network works better.

Besides, we have:
statistic=43.803766234289796, pvalue=3.611856340925435e-66
While comparing the sgd with mini-batch under neural network. That means that we should reject the null hypothesis of equal averages. SGD works better.

Unexpected factors found:

1.Use large mini-batch size will cause the learning process too slow. The neural network is written on numpy instead of tensorflow so the learning process for nn is not that fast. Learning for 500 epoch may cost 4 hrs(mini-batch with size 4 nearly converge after 500 epoch, if the batch size becomes larger, it may cost more epochs to learn). Therefore I decided to use 4 as the mini-batch size to decrease the learning time.

2. The logistic regression works much better than I think, it can converge in a pretty small epoch. Although the final result is a bit worse than neural network. I think one of the reason that cause this is that the mnist has large enough training data (50000 data) and logistic regression is much more sensitive than nn so it can learn very quickly.

3. mnist dataset is not normalized, so I need to normalize it or the sigmoid or softmax function will overflow.

4. The result data of mnist is numbers like 0-9 . It should be transferred to the format like[0,0.0....1...0] to match the output of softmax.

Summary:

From the result chart and t-test we can figure out that the neural network works better than simple logistic regression on mnist dataset if the epoch is large enough. Although the logistic regression can learn faster, but for the final accuracy, the neural network is better. That might because the logistic regression is simpler. Therefore, it can learn faster, but cannot represent complex models very well. Neural network is more complex, although learn slower, it can represent complex models better.

Another comparison I have done is between sgd neural network and minibatch neural network. The sgd works better on both begin and the end of the learning process. I believe if I have large enough epochs(like training for a whole day) the final result of mini-batch gd should be better, but that cost too much time and the improvement on performance is not proportional. That might because the mini-batch gradient descent is a kind of adaptive step-size gd, although it does not go over the optimal point, its stepsize could be too small. Therefore it is reasonable to use sgd if you would like to get a good result in a short time but use too much time to get a optimal result.

The conclusion that I can draw from this project is that neural network indeed has some advantage on handling image identification problems. Even I just have simple structure, it works better than logistic regression obviously.

Another conclusion that I can draw is that when the dataset is large enough and the target model is pretty complex(I think mnist graphic identification model is too complex to a two-layer neural network). Using sgd is enough. If you have some good designed neural network or a smaller dataset, mini-batch gradient descent could be a better choice.