

CS 398 Deep Learning @ UIUC

Homework 3 Convolution Neural Network for MNIST Dataset

Name: Yawen Duan UIN: 655877290

HW3 Description:

Implement and train a convolution neural network from scratch in Python for the MNIST dataset (no PyTorch). You should write your own code for convolutions (e.g., do not use SciPy's convolution function). The convolution network should have a single hidden layer with multiple channels. It should achieve at least 96% accuracy on the Test Set. For full credit, submit via Compass (1) the code and (2) a paragraph (in a PDF document) which states the Test Accuracy and briefly describes the implementation. Due Thursday, February 14 at 5:00 PM.

Implementation

In my code, I defined an object `CNN` to represent the model and contain its parameters. I first initialize a random set of parameters, and then I use stochastic logistic regression algorithm to train the convolutional neural network model with data replacement. Then I test the data based on the training dataset to get the accuracy score. Below are the related parameters I used.

```
batch_size = 1
num_iterations=150000
learning_rate=0.008
stride=1
padding=0
dim_kernal=3
num_kernels=5
dim_inputs=28
input_chanl=1
len_outputs=10
```

I wrote 8 methods including `Softmax(z)`, `activfunc(self,z,type = 'ReLU')`, `cross_entropy_error(self,v,y)`, `forward(self,x,y)`, `convolution(self,x,kernels)`, `back_propagation(self,x,y,f_result)`, `optimize(self,b_result, learning_rate)`, `train(self, X_train, Y_train, num_iterations = 1000, learning_rate = 0.5)`, `testing(self,X_test, Y_test)` to handle initialization, model fitting and testing.

Test Accuracy

The test accuracy and value of loss function with respect to the number of iterations within one time of modeling are shown as follows. Note the test eventually has achieved an accuracy score of around 94%.

Trained for 30000 times, loss = NA, test = 0.9226 Trained for 60000 times, loss = NA, test = 0.9242 Trained for 90000 times, loss = NA, test = 0.9217 Trained for 120000 times, loss = NA, test = 0.9343