CS 547 Homework 2

Name: Jiashuo Tong
Date Submitted: Sep., 16, 2019

Neural Network Implementation

Function Definitions

1. def loadData(url, nimage): loadData reads a image and formats it into a 28x28 long array
2. def loadLabels(url, nimage): loadLabels reads the corresponding label data, one for each image
3. def download(dataurl, labelsurl, nimage): download uses loadData and loadLabels to generate raw data to be used later
4. def tanh(M): tanh is an activation function defined as 0 for x < 0 and x for x > 0
5. def tanh_p(M): tanh_p is the derivative of relu function
6. def softmax(z): softmax is a function that converts the raw scores for each category to probabilities that sum up to 1
7. def feedforward(x,y): feedforward feeds MNIST data to the neural network and computes values at each intermediate layers Z, H, U and the probability vector f.
8. def backpropagate(x,y,Z,H,f): backpropagate uses the training data along with the results (Z,H and f) from feedforward to compute the gradients of the loss to each parameter array (W,K and b) of the neural network. It outputs the updates for the parameter arrays, namely dr/db, dr/dW and dr/dK.
9. def conv(X, K, iteratable, l_f): conv function implements the convolution operation on the input image X and filter K. It takes in the l_f parameter which stands for the length of the filter. It uses numpy.tensordot function to speed up tensor computation.
10. def predict(Xdata, Ydata): predict function predicts the categories for a dataset using the trained CNN and calculate the accuracy on the dataset by comparing the results to the labels.

Main Block

The main block starts by downloading data from LeCun's website and preparing the data so they can be fed into the neural network. A for loop is used to do multiple epochs on the training dataset. Inside the for loop, it first calls the feedforward function to compute the intermediate layers Z, H and U. Then, it calls the backpropagate function to compute the gradients of the loss to the parameter arrays. Finally, the parameters are subtracted by the learning rate times the gradients according to the SGD algorithm.

The code has achieved an 94% accuracy on the test set in the 2nd epoch since CNN is very efficient in capturing features of handwritten digits.

Results

Although the training process is very long (2 hr/epoch), I have achieved a test accuracy of > 94% in the 2nd epoch.