Programming Assignment 2

Cloth Classification using Deep Neural Networks in numpy

Imports:

Use only numpy and a few other basic python libraries. Do not use any automatic differentiation libraries like Tensorflow or Pytorch.

Task:

Build a neural network with 1 hidden layer and perform the cloth classification task Neural network specifics:

- 1. No of hidden layers : 1
- 2. Non-linearity in all layers : relu
- Forward pass equation : y_ = softmax((W₂(relu(W₁x + b₁)) + b₂) wherein x is your input item and y_ is the vector of same size as that of no of classes. It gives the probability that x belongs to a particular class.
- 4. Use softmax at the output of the final layer to calculate probability of the item belonging to each class. (Be careful and check that your softmax function calculates correct values.)
- 5. Loss function: categorical cross entropy
- 6. Regularization: Use L2 regularization to limit the capacity of your model to prevent overfitting.

Data: https://goo.gl/dYFTP7

- train_images: 60000 examples of 28x28 grayscale image. The data is of size 60000 x 784. Each pixel has a single intensity value which is an integer between 0 and 255.
- train_labels: 60000 labels from 10 classes for the images in given in train_images. Class details are mentioned below.
- test_images: 10000 examples of 28x28 grayscale image. The data is of size 10000 x 784. Each pixel has a single intensity value which is an integer between 0 and 255.
- test_labels: 10000 labels from 10 classes for the images in given in test_images. Class details are mentioned below.

Labels

Each training and test example is assigned to one of the following labels:

- 0 T-shirt/top
- 1 Trouser
- 2 Pullover
- 3 Dress
- 4 Coat
- 5 Sandal
- 6 Shirt
- 7 Sneaker
- 8 Bag
- 9 Ankle boot

Important Note: Please make sure that the data for training and testing is placed in '../data/' directory i.e. present in the parent directory of python files being created. You don't have to submit this data files in your final submission. Also do not change the names of any files.

Code Outline:

Code outline for all files in implementation section is provided here: <u>https://github.com/cs60010/Tutorials/tree/master/code</u>

Load Data:

Function for reading data from zip files have been implemented in the code outline that is provided. It will work if all the zip files for train and test data are in the data directory.

Implementation

Create the following files for writing your code:

- 1. data_loader.py : Use this file to load all datasets and create mini batches
- module.py : Create a neural network class here. Initialize all weights in the __init__() constructor. Create a member function forward() where you do a forward pass of the neural network. Create a member function backward() where you compute the loss and do a backward pass(backpropagation) of the neural network and update all weights.
- 3. train.py : Implement a simple minibatch SGD loop and train your neural network. Finally it should print the final accuracy after training.

Training and testing the network: *python train.py* should train the neural network, print out all losses on train data and the final accuracy.

Store the final accuracy in a file 'accuracy.txt'. Submit this along with your code. We should get a similar accuracy as reported in 'accuracy.txt' when we run your code. Make sure you use a fixed random seed in your code.

Reading Resources:

- For calculating vectorized gradients during backprop: <u>http://www.psi.toronto.edu/~andrew/papers/matrix_calculus_for_learning.pdf</u>
- 2. A more detailed matrix calculus tut: http://cs231n.stanford.edu/handouts/derivatives.pdf

Submission Instructions

- You have to use **only** numpy and basic python functions and **no** other imports are allowed.
- Implement the forward and backward passes of your network using vectorised numpy operations.
- Please make sure that the **path** for the data is **same** as demonstrated in the command to read labels and images in the code outline.
- Please submit the completed compressed file named as 'Assignment2_<Roll No>.zip' containing the files mentioned in implementation section. Please don't submit any data files.

Bonus points:

Make some interesting observations or provide some insights about the model that you have built.