Cheruvu Internship Assignment

Round 1

Submitted by: Rishabh Sobti

This document consists of a brief explanation of the code and methodology used in solving the two questions provided as per the round one assignment of Cheruvu ML Internship opportunity.

Question 1

Question statement: Consider that you have a neural network with two hidden layers and let **X** be the input, **W1, W2, W3** be the weights of the hidden layers, **b1, b2, b3** be the corresponding biases and **y** be the output. Implement the forward pass and backward pass (you can assume any loss function and activation function). You don't have to test your code on any data. We will just check the implementation and ask questions based on that (If possible vectorize your code using numpy).

Explanation of the implementation: I have considered two hidden layers, each containing 3 hidden nodes/neurons. The input X and the 'target' are taken from a random distribution. The weights are initialized randomly using a normal distribution and the biases (i.e., the weights associated with bias neurons) are initialized at zero. The network is set to run for 900 iterations/epochs, with a learning rate of 0.005. The variables ending in 'error' show the actual deviation from result. The terms ending in 'error_term' show error in the output's effect on the input to the neuron. And the terms starting with 'del' actually shows the effect of these errors which back-propagates to the activation of previous layer.

The loss function is chosen as "squared-error" loss function, and sigmoid activations are chosen for each neuron.

Question 2

Question statement: The goal of this question is to check your knowledge of existing algorithms, your ability to come up with new algorithms and efficiency of your code. Attempt the following kaggle competition and document the results: https://www.kaggle.com/c/afsis-soil-properties

Objective

The problem was to predict some soil properties (Ca, P, pH, SOC, Sand) based on the Near Infrared (NIR) data. Spectral features and spatial features were present for analysis and training. There were 1158 instances in train data and 728 instances in test data with 3578 spectral features and 16 spatial features

Evaluation Metric

Submissions are scored on MCRMSE (mean column wise root mean squared error):

$$MCRMSE = \frac{1}{5}\sum_{j=1}^{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(p_{ij} - a_{ij}\right)^2}$$

Where ‘p’ is predicted value and ‘a’ is actual value.

Preprocessing

I have carried out three tasks to preprocess the data:

- Label Encoding – The ‘Depth’ field had two values ‘topsoil’ and ‘subsoil’, which were encoded in binary fashion to ‘0’ and ‘1’.
- PCA – Principal Component Analysis was carried out to reduce the high dimensionality of data. At the end, only 10 mixed-components were chosen for training, as they constituted more than 99% of the variance in the data.
- Normalization – The newly obtained 10 features from PCA were normalized for efficiency.

Modeling Algorithms

I have used an ensemble of 2 algorithms in 5 setups, to get the results. The different setups, along with their weights/contributions in prediction, are given below:

- SVR_1: Stands for Support Vector Regressor setup 1, uses the support vector machine for regression algorithm as provided by sklearn.svm.SVR. The C is set to 5000 and epsilon to 0.5. The contribution is (0% for Ca, P, pH, 80% in SOC and 60% in Sand)
- SVR_2: Stands for Support Vector Regressor setup 2, uses the support vector machine for regression algorithm as provided by sklearn.svm.SVR. The C is set to 1 and epsilon to 0.1. The contribution is (30% for Ca, 10% for P, 10% for pH, 0% for SOC and 0% for Sand)

- Net1: Stands for Multi-Layer Perceptron Neural Network setup 1, uses the fully-connected layering code as given in the code file. It has 2 hidden layers with 5 neurons each and dropout probability of 0.5. It has sigmoid activations. The contribution is (30% for Ca, 30% for P, 40% for pH, 0% for SOC and 0% for Sand)
- Net2: Stands for Multi-Layer Perceptron Neural Network setup 2, uses the fully-connected layering code as given in the code file. It has 1 hidden layer with 20 neurons and dropout probability of 0.5. It has sigmoid activations. The contribution is (10% for Ca, 30% for P, 10% for pH, 10% for SOC and 10% for Sand)
- Net3: Stands for Multi-Layer Perceptron Neural Network setup 3, uses the fully-connected layering code as given in the code file. It has 2 hidden layers with 5 neurons each and dropout probability of 0.5. It has relu activations. The contribution is (30% for Ca, 30% for P, 40% for pH, 10% for SOC and 20% for Sand)

Result

The test set predictions have an MCRMSE (private) of 0.79 on the kaggle.com website. The public score is 0.84.