ML Challenge | Land Classification

Name: Pruthviraj R Patil

Btech-CSE, VIT CHENNAI,

Mail: [pruthvirajr.patil2016@vitstudent.ac.in](mailto:pruthvirajr.patil2016@vitstudent.ac.in)

***The output data:***

*The output is inserted in the folder as csv file with the name: “final\_submission\_nn.csv”*

*Jupyter notebook as well as the HTML file is inserted too.*

1**. The first 3 things I did to understand the data better (including any graphs you plot or summaries you generated)**

* Plotting the Scatter plot between features using HUE as the target variable.
* Some of the features were found to be positively correlated and some were found to be negatively correlated. The classes were completely overlapped and hence by this we can interpret that a function that projects it into the higher dimension so that it can be separable there that leads us to go towards MLP.
* It is seen that the first 6 variables contains values that are continuous but they aren’t properly scaled when compared to the next 6 variables. The scattered Data said that there was more variation in few data features. Hence correlation of the data is used to plot the heat map and hence to implement PCA to eliminate most varying data points.

2. **Pre-processing involved**

**STEPS:**

1. Check for null values: There weren’t any
2. Check for the Variations of the data as there were many, PCA needed to be applied
3. Standard Scaler is used
4. PCA applied on standard scaled Data
5. Graphs drawn to interpret.

**3. The different models I tried - what worked and what did not**

Planned to try:

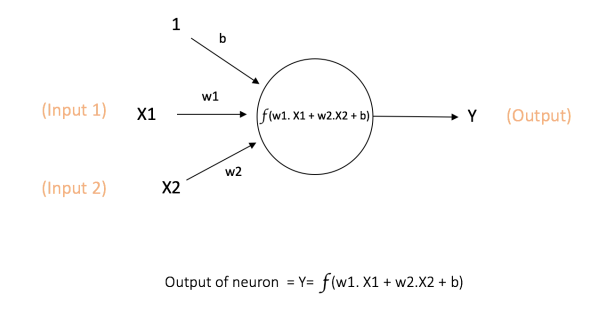
1. MLP(ANN) with single hidden layer on raw data without PCA applied(92.3% accuracy)
2. MLP with single hidden layer applying grid search to get the best hyper parameters (95% accuracy)
3. Grid search’s best Model on PCA(97.8 % accuracy)
4. MLP with hidden layers(98.5% accuracy)

This did not work:

CNN couldn’t be used as there weren’t any time variant data or any shift variant and the space variant. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in image and video recognition, recommender systems and natural language processing.

**4. Explain the basics of the model you used and how you measured the error**

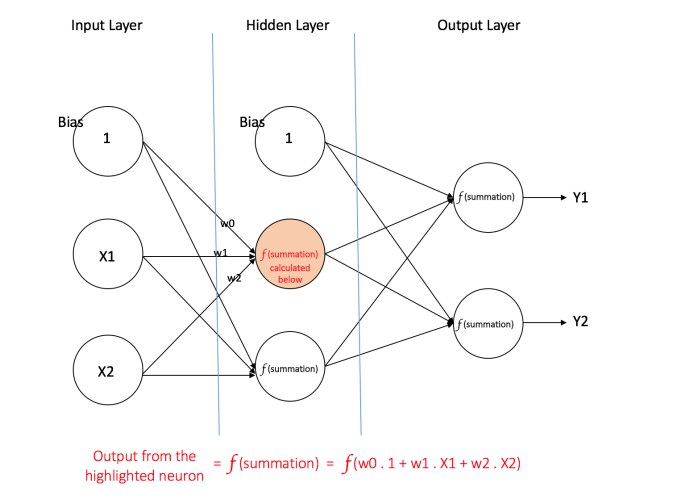
**MLP** is also known as Artificial Neural Network (ANN). It is a computational model which is a neural network and the basic unit of computation in this neural network is the neuron, often called a node or unit that applies a function f (defined below) to the weighted sum of its inputs as shown:



The function f is non-linear and is called the Activation Function. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

Mainly 3 types of Activation functions, they are tanh, relu, sigmoid.

A Multi-Layer Perceptron (MLP) contains one or more hidden layers (apart from one input and one output layer). While a single layer perceptron can only learn linear functions, a multi-layer perceptron can also learn non – linear functions.

So, there are manly 3 types of layers, I/p, Hidden, Output layer.

There may be multiple Hidden layers.

Hidden layers are comprised of the bias and input variables' combination.

The process by which a Multi-Layer Perceptron learns is called the **Back propagation** algorithm.

So, initially all the edge weights are randomly assigned. For every input in the training dataset, the ANN is activated and its output is observed. This output is compared with the desired output that we already know, and the error is “propagated” back to the previous layer. This error is noted and the weights are “adjusted” accordingly. This process is repeated until the output error is below a predetermined threshold.

So, the error is measured and adjusted using the back propagation algorithm and are adjusted propagating in that manner itself.

**5. How further could this have been improved and the code optimized for speed**

We shall use regularization, optimization like Adam’s Optimization as well as initialization like Xavier Initialization and as well as He Initialization to improve the speed of the algorithm.

We shall further tune the Hidden layers’ size and shape to get the proper fit.

**6. Anything else I think I did differently:**

Tried grid search over PCA’d values over MLP Classifier to get the best hyper parameters. Hence Better results were expected. To improve the speed of the grid search, early stopping is kept toggled to true, and while using the best model out of it early stopping is turned off (False).