CLASSIFYING FOREST SUPRA-TYPE FROM AUGMENTED SATELLITE DATA: COMPARING SIMPLE NEURAL NETWORK AND CASPER TECHNIQUE

J. He

College of Engineering and Computer Science, Australian National University

**Abstract**

Geographical data, especially information with detailed terrain type has become intensely important for various applications and research work. With structured and known satellite data, more detailed information can be generated.

The task is to classify the forest supra-type among five options based on geographical features mainly collected by satellite.

We apply two kinds of neural networks, a two-layer neural network and another with the Casper technique, to give a classification of Forest supra-type based on GIS dataset. We describe the performance of both methods and analyze the benefits or lack of benefits. Some comparison with other results performed on the same dataset are also made.

**Keywords**

Neural Network; CasPer; GIS dataset

# Data and Preprocessing

The Geographic Information System data was collected from an area in the Nullica State Forest on the south coast of New South Wales. The data was integrated from satellite imagery, soil maps and aerial photographs [1].

Among the spots explored, this dataset gives us 190 sets of features and forest supra-type classifications. In the raw data, there are 16 values for each plot, including aspect, sin and cos of aspect, altitude, topographic position, slope degree, geology descriptor, rainfall, temperature, and Landsat TM bands 1 to 7.

Some of the features have already been encoded by the geographers who provided the data. We have to decrypt the data and restructure the inputs and outputs. The preprocessing procedures performed on each feature are described as below:

* Aspect: It’s originally represented in degrees which has a problem of continuity. To deal with this, the aspect is encoded as four inputs, in which each represents a major direction and spread adjacent directions accordingly [2].
* Sin and Cos of aspect: Redundant with Aspect. Not used in our classification.
* Altitude: Normalized to 0-1.
* Topographic position: Normalized to 0-1.
* Slope degree: Normalize to 0-1.
* Geology descriptor: Unknown meaning in values. Since the values appear to be categorical and has no particular distribution, we encode it as four inputs, distinguishing between the popular types and the rare ones [2].
* Rainfall: Normalize to 0-1.
* Temperature: Normalize to 0-1.
* Landsat bands 1-7: Normalize to 0-1.

The output categories are encoded by 4 units, in the purpose of avoiding sparse output vectors by equilateral coding. All units have some activation on each pattern, and the maximum distance between vectors is maintained. The final category can be computed through Euclidean distance between predicted values and the five different types [2].

As the number of samples is very small (only 190) in this dataset, 5 cross validation is used in order to avoid bad splitting of data.

# Simple Neural Network Method

We perform a vanilla neural network on the dataset for cross validation first. This is initially a classification problem, in which forest supra-type is given as target, however, by equilateral encoding discussed above, it is modified into a regression problem.

Parameters

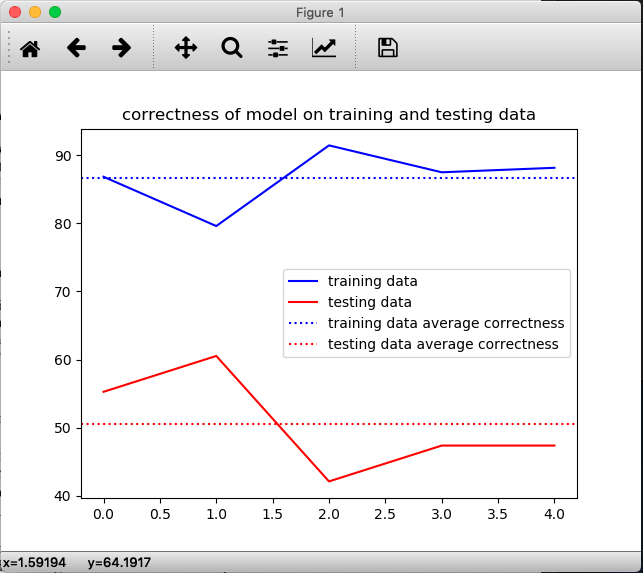
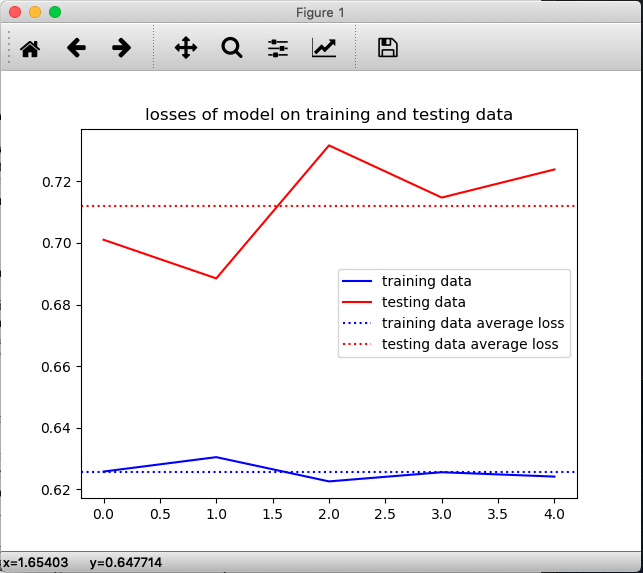
The regression model is defined with the following parameters:

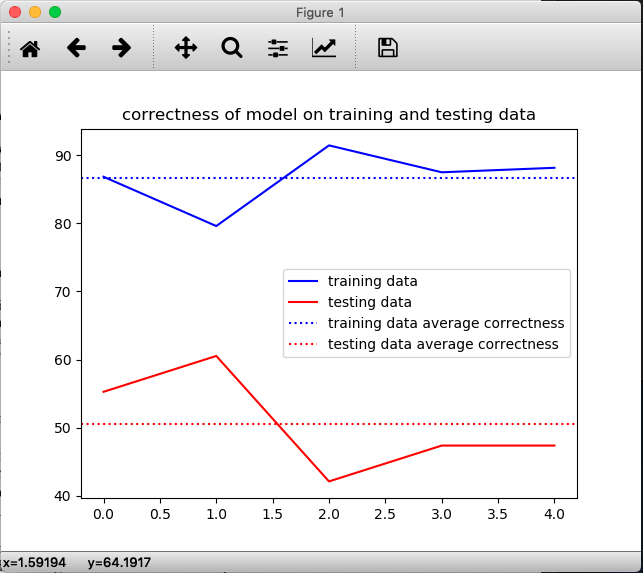
* Input size: 20; hidden size: 12; output size: 4
* Number of epochs: 1000 (number 500 and 1500 have also been tried, in balancing learning and over-fitting, 1000 is set as a preferred choice.)
* Loss function: binary cross entropy. In PyTorch, we use BCEWithLogitsLoss, since this includes the activation function (Sigmoid) of last layer, and ensures numerical stability by combining the operations into one single layer.
* Optimizer: Adaptive Moment Estimation (Adam in PyTorch). This avoids the problem of getting stuck in local minima in stochastic gradient descent by adapting learning rate [3].

Performance

The neural network regression model on GIS dataset performs quite well through parameters described above. In the five cross validation processes, the average loss and correctness (proportion of correct classification) on training and testing data are displayed in the following table, detailed data are displayed in figure 2.2.1 and 2.2.2:

|  |  |  |
| --- | --- | --- |
|  | Average loss value | Average correctness (%) |
| On training data | 0.6257 | 86.7105 |
| On testing data | 0.7120 | 50.5263 |

 Figure 2.2.1

 Figure 2.2.2

CasPer Technique