Forest Fire Prediction Using Multilayer Neural Network

Mehmet Şerbetçioğlu - 040160056
Veli Bulur - 040150051

The Problem

Forest fire prediction depends on multiple variables. These variables may or may not correlate with each other and the correlation is unclear. Variables such as the vegetation, time of the day, meteorological data can affect the chances of a fire starting and the area of spread. When faced with such difficult correlations, machine learning becomes a possible solution to make predictions about the subject.

The Project

In this project, we have constructed a Multilayer Neural Network in order to make predictions about the size of burnt area after a wildfire. As our data, we will be using a forest fire area dataset [1], which have information regarding spatial, temporal and meteorological data as well as the size of burnt area from each fire. We will categorize the burnt area logarithmically and train the network to predict the categories.

There have been different studies that show the correlation between variables [2][3] and predict the burnt area using different methods such as random forest, multilayer neural network and support vector machines [4][5].

The Dataset

The dataset [1] contains information about 517 fires from the Montesinho natural park in Portugal. There are 13 variables present in the dataset. These variables are listed.

FFMC, DMC, DC and ISI indexes are indexes that are denoted by Fire Weather Index System [6]. These indexes are used by National Wildfire Coordinating Group in order to rate the danger of a fire.

Also, there is a large amount of samples with 0 valued burnt area in the dataset. What this means is that the burnt area for that particular fire was smaller than 100m2.

Dataset Contents:

- X Coordinate
- Y Coordinate
- Month
- Day
- FFMC Index
- DMC Index
- DC Index
- ISI Index
- Temperature
- Relative humidity
- Wind
- Rain
- Burnt Area

Multilayer Neural Network

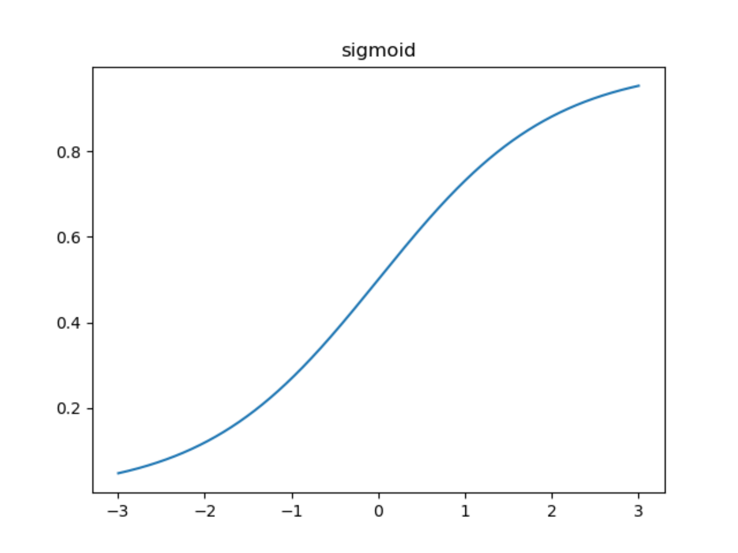
Multilayer Neural Network is a construct that is originally designed to simulate the behavior of a human brain. The network is formed of layers and each layer has a number of neurons in them. With proper excitation, these neurons fire a signal forward, which then feeds into other neurons.

Each neuron has an input vector, a weight vector and an activation function. The input vector is multiplied by weight vector in order to get the argument of activation function. Output of the activation function determine the excitation of the neuron. The selection of activation function is important for constructing an efficient neural network.

Activation Functions

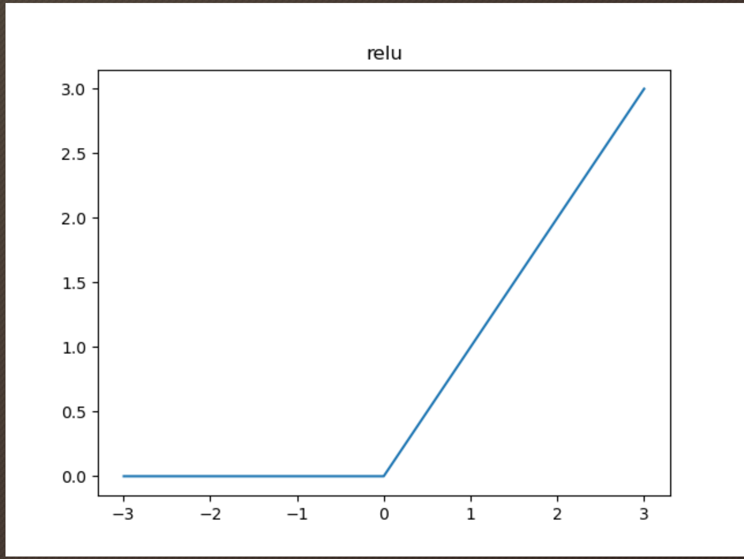
Traditionally, logistic activation functions such as sigmoid and tangent hyperbolic functions are used in multilayer neural networks. However, more modern networks use more efficient activation functions such as rectified linear unit (ReLU) and leaky rectified linear unit (LReLU) functions. This project will cover different types of activation functions and their performance for the present problem.

Activation Functions



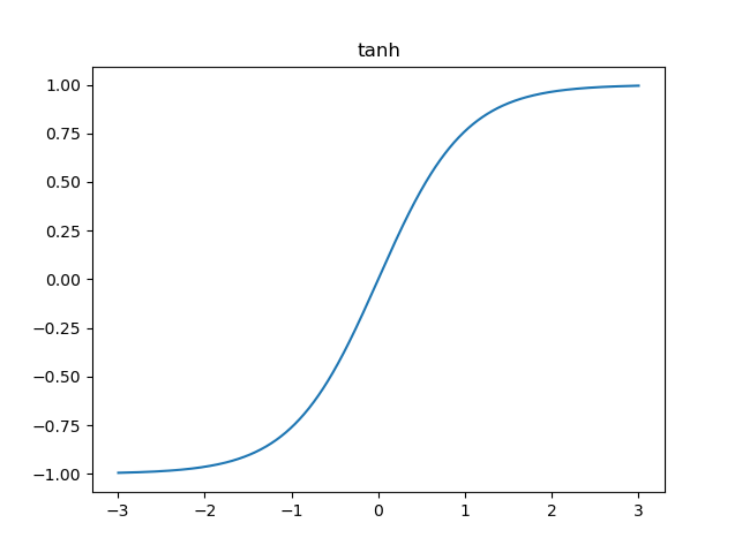
Sigmoid:

$$f(x) = \frac{1}{1 + e^{-1}}$$



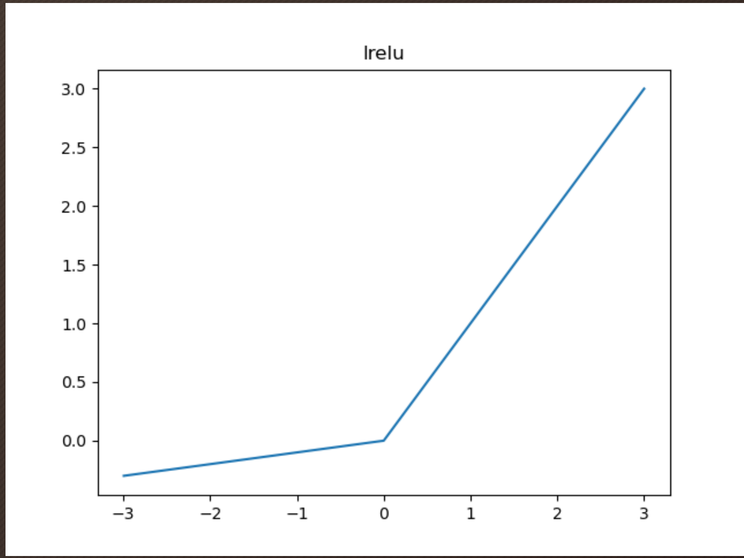
ReLU:

$$f(x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$$



Tanh:

$$f(x) = \frac{1 - e^{-1}}{1 + e^{-1}}$$



LReLU:

$$f(x) = \begin{cases} 0.1x \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$$

Multilayer Neural Network

The network in this project will utilize back propagation method in order to train the weight vectors. Back propagation method involves calculating the derivative of the error with respect to each of the weights in order to determine the direction each weight should change towards.

The network will also utilize a momentum term in order to increase the learning speed and decrease the chance of getting stuck in local minima. Momentum is an adaptive term that increase or decrease the change in weights, depending on the difference between current and previous weights.

Size of the burnt areas will be categorized into four different categories, while x is the size of the burnt area in hectares;

- $x < 1$ => Category 1
- $1 \leq x < 10$ => Category 2
- $10 \leq x < 100$ => Category 3
- $100 \leq x < 1000$ => Category 4

Roughly half the data from each category will be selected for the training set while the rest will be added to testing set. Since the amount of data at each of the categories is highly sided towards category 1, data selection is important for the healthy training of the network.

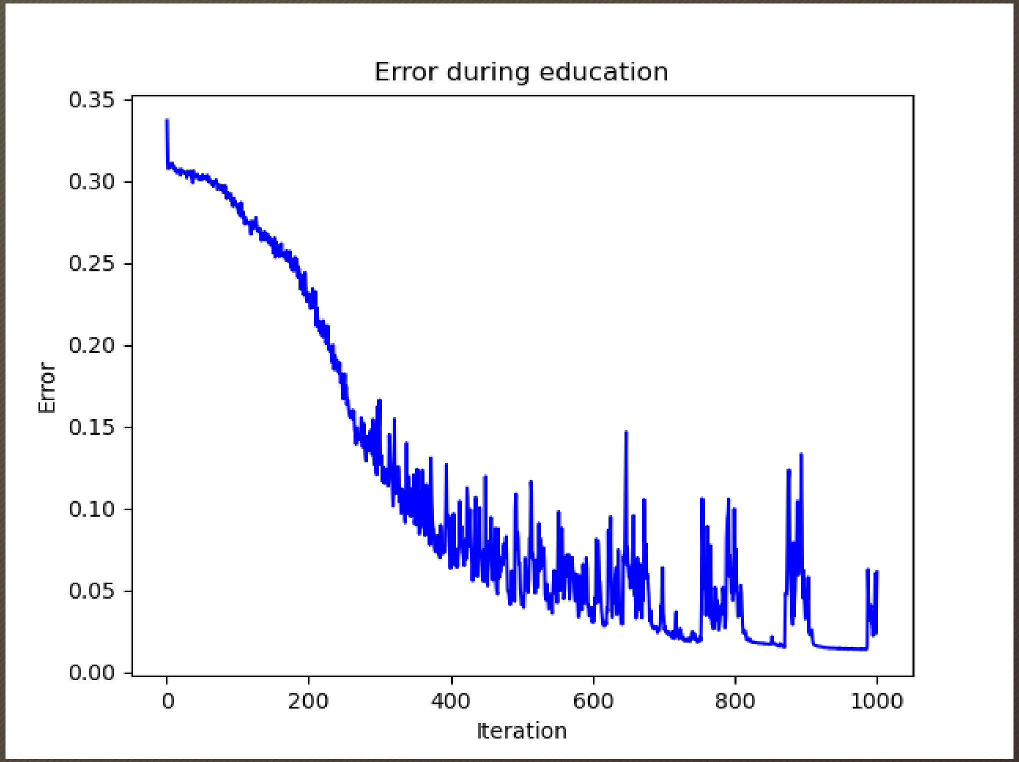
Multilayer Neural Network

Network will have an input layer, an output layer and two hidden layers in between. Since there are four categories of fires, the output layer will contain 4 neurons with binary outputs. Therefore, for the output layer, sigmoid function will be used through the whole project. For other layers, different activation functions will be tested for performance. Network parameters will be selected as;

- Learning Rate = 0.01
- Momentum Constant = 0.9
- Neuron Count of Hidden Layer 1 = 12
- Neuron Count of Hidden Layer 2 = 12

Also a maximum iteration number and an error threshold will be selected to stop the learning. The education will stop if the maximum number of iterations is reach or the mean squared error becomes less than the error threshold. For these parameters, the maximum iteration number will be selected as 1000 and the error threshold as 0.01.

The Results – Training Error



Best performing activation function: Tanh(x)

ReLU, LReLU, sigmoid and tanh functions were tested. ReLU and LReLU has shown good regression results but only tanh function was able to get under 0.05 error mark. Sigmoid function performed the worst with only one test getting near 0.25 error mark.

Another thing to note is that sigmoid function show less fluctuation in the error during training. This helps with constant regression but causes the network to be stuck at local minima on some occasions.

The graphic on the left is from a single test with tanh function, other functions' performances can be found at the associated github repository[7]

The Results – Education Duration

Activation Function	Education Time [s]
Sigmoid	400.44
Tanh	399.44
ReLU	285.64
LReLU	289.69

When education duration for each activation function is compared, it is clear that ReLU and LReLU perform better than both logistic functions. The algorithm for logistic functions involve multiple mathematical calculations involving exponentials while ReLU and LReLU contain only a couple of if statements. It is important to note that the education times on the table are measured with by averaging the times of educations with 1000 iterations.

The Results – Test Error

Activation Function	Test Error [%]
Sigmoid	48.13
Tanh	57.4
ReLU	52.77
LReLU	57.14

Test error percentages show that the network was not capable of reliably predicting the category of fire. Tests with lower education error have higher test error, which indicate that the data is not reliably correlative. For a more successful prediction, a better data representation is recommended.

Bibliography

[1] S. Mishra, "Forest Fire Area", *Kaggle.com*, 2020. [Online]. Available: https://www.kaggle.com/sumitm004/forest-fire-area. [Accessed: 21- Jan- 2021].

[2] S. Mishra, "Forest Fire Area Predictions and EDA", *Kaggle.com*, 2021. [Online]. Available: https://www.kaggle.com/sumitm004/forest-fire-area-predictions-and-eda. [Accessed: 21- Jan- 2021].

[3] "Starter: Forest Fire Area", *Kaggle.com*, 2020. [Online]. Available: https://www.kaggle.com/kerneler/starter-forest-fire-area-926a18c9-0. [Accessed: 21- Jan- 2021].

[4] K. Kumar, "Forest Fire Prediction Using Random Forest", *Kaggle.com*, 2020. [Online]. Available: https://www.kaggle.com/kishore1999/forest-fire-prediction-using-random-forest. [Accessed: 21- Jan- 2021].

[5] P. Cortez, A. Morais. “A Data Mining Approach to Predict Forest Fires using Meteorological Data”, in *Portuguese Conf. on A.I.*, J. Neves, M. F. Santos and J. Machado Eds., Guimarães, Portugal, Dec. 2007, pp. 512-523. Available: http://www.dsi.uminho.pt/~pcortez/fires.pdf

[6] “Fire Weather Index (FWI) System,” *NWCG*. [Online]. Available: https://www.nwcg.gov/publications/pms437/cffdrs/fire-weather-index-system. [Accessed: 22-Jan-2021].

[7] M. Şerbetçioğlu and V. Bulur, "Multilayer Neural Network", *GitHub*, 2021. [Online]. Available: https://github.com/meserbetcioglu/MultilayerNN. [Accessed: 22- Jan- 2021].