**Assignment 2**

Data Analytics Foundation

Mandeep Sarangal

251000108

***Description of the forecasting problem and data***

Weather data for the city of Szeged in Hungary, from 2006 to 2016, is available. This forecasting problem deals with predicting the **Apparent Temperature** value given the values of 6 parameters which are **Humidity, Temperature, Wind speed and Pressure** on that day.

Y = Temperature

X = Humidity, Apparent temperature, Wind speed, Wind Bearing, Visibility and Pressure

Where X represents all the independent variables/attributes, also known as predictor variables. Based on the values of X, we will predict the values of Y which is our variable of interest.

Description of available data:

|  |  |  |
| --- | --- | --- |
| **Column name** | **Type** | **Description** |
| Formatted Date | DateTime | Date and Time of the day |
| Summary | String | Short Summary of the day |
| Precip type | String | Type of precipitation |
| Temperature | Numeric | Actual Temperature |
| Apparent Temperature | Numeric | Temperature perceived by humans |
| Humidity | Numeric | Value of humidity |
| Wind Speed | Numeric | Speed of Wind |
| Wind Bearing | Numeric | Direction of the Wind |
| Visibility | Numeric | Distance at which an object or light can be clearly discerned |
| Loud Cover | Numeric | Total cover |
| Pressure | Numeric | Value of atmospheric pressure |
| Daily Summary | String | Overall summary for the day |

***Overview of network architecture and parameters that were tuned***

**Tuned parameters:**

* *Hidden Layers*

A thumb rule for deciding the number of hidden layers required in a problem statement is as follows:

= = 2

So I experimented with 1 and 2 hidden layers at a given time.

Also, if the data is linear, it does not need any hidden layers at all. But if the nature of data is more complex It will need hidden layers. Each progressive hidden layer captures increasingly more complex features and patterns in the data which help making accurate predictions in the output layer.

* *Neurons*

Deciding the number of neurons in the hidden layers is a very important part of deciding your overall neural network architecture. Using too few neurons in the hidden layers will result in something called Underfitting. Underfitting occurs when there are too few neurons in the hidden layers to adequately detect the signals in a complicated data set.

Using too many neurons in the hidden layer can cause two problems. First, it can result in Overfitting the data. Overfitting is a condition when the neural network has so much information processing capacity that training set is too small to train all the neurons in the hidden layers. Second, even if the training data is sufficient, large number of neurons might drastically increase the training time of the neural network, which is not desirable.

Ultimately, the selection of number of neurons in each hidden layer neural network comes down to trial and error.

* *Back propagation Algorithm*

I tried 2 variations of back propagation algorithms, namely:

1. Normal Back propagation (algorithm=’backprop’)
2. Resilient Back propagation (algorithm=’ rprop+’)

Resilient Back propagation gave a better result by reducing the error value

* *Activation Function*

Choosing a good activation function allows training better and efficiently.

I tried 3 variations of activation functions

1. Sigmoid: It is a probabilistic function and is most useful in classification problems
2. TanH: This activation function gave me the best result. The resulting error was the lowest in this case.
3. Relu: It is the most commonly used function but in this particular problem, the results given were not optimal.
4. MSE : 0.0000807523271

***Description of the process, including, data pre-processing, model training and evaluation.***

**Step 1**

In order to select relevant independent attributes, I calculated the Correlation Co-efficient for all the independent variables among each other. Only those attributes were selected for whom the correlation co-efficient was not strong on either sides i.e. it was neither very close to 1 nor to -1. It’s a good practice to choose independent attributes which are not strongly linear in relationship to each other.

If any 2 independent attributes had a very high correlation co-efficient, one of them was dropped.

After above analysis, 4 attributes were shortlisted for further data processing:

* Temperature
* Humidity
* Wind Speed
* Pressure

**Step 2**

For the optimization to proceed numerically smooth, all the attribute values were normalized using min max scaling. Normalizing essentially means scaling the values such that they have zero mean and equal variance.

**Step 3**

The dataset was divided into Training, Validation and Test datasets in the ratio 3:1:1 respectively.

Training data was used to train the Algorithm