**Mortality prediction based on weather data**

*An end to end model for mortality prediction using deep learning*

Bruchental Robert-Ionut

*Faculty of Mathematics and Computer Science | Babes-Bolyai*

**1.Goals**

From weather data collected from 7 cities across 14 years, an RNN model is proposed to try predict the mortality rate based on gender and age group.

**2.First steps**

Based on the research conducted by Zheng Chu and Jiong Yu at the Xinjiang University[[1]](#footnote-1), it can be concluded that the study of weather-based predictions can be conducted using deep learning. Knowing that we first tried to do a simple neural network that can predict if it will rain the next day, based on weather data collected during a year period.

From inputs:MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, WindGustDir, WindGustSpeed, WindDir9am, WindDir3pm, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Cloud9am, Cloud3pm, Temp9am, Temp3pm, RainToday, RISK\_MM the model managed to predict the output RainTomorrow with an accuracy of 75% using a simple network architecture consisting of three fully connected layers after which the sigmoid activation function is applied.

**3.Actual data**

**3.1.Mortlity data**

When working with the actual data the first thing that had to be done was clean the data, by adding 0 to the columns where the information was missing. The reason for not completing it with the mean of the column was because the original data also had a total column, that we didn’t use anymore as it was redundant.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | F<19 | M<19 | T<19 | F19-65 | M19-65 | T19-65 | F>65 | M>65 | T>65 |
| 01/01/2000 |  |  | 0 | 1 | 3 | 4 | 1 | 6 | 7 |
| 01/02/2000 |  |  | 0 | 1 | 3 | 4 | 5 | 3 | 8 |
| 01/03/2000 |  |  | 0 | 2 | 4 | 6 | 3 |  | 3 |

**3.2.Weather data**

The weather data had no incomplete information so no additional inputs where necessary, the only additional action that was needed was exctracting the number values from columns which also contained text.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Date | MaxT | MinT | Prec | Pressure | WindDir | WindSpeed | CloudCoverage |
| 19/08/2000 | 36.7 | 14.4 | 0.0 | 1018.8 Hpa | 4º(N) | 3 | 1/8 |
| 20/08/2000 | 37.6 | 15.1 | 0.0 | 1017.2 Hpa | 88º(E) | 4 | 1/8 |
| 21/08/2000 | 38 | 14.9 | 0.0 | 1015.8 Hpa | 76º(E) | 4 | 0/8 |

Then the two datasets are combined, uniting the information based on the date column. The data is then normalized using min-max and split into train and test data.

A diagram of a computer

Description automatically generated with medium confidence

**4.The model**

**4.1.Model architecture**

As the problem was a time-series related one, we introduce a gated recurrent unit layer that processes sequential input data. It has 2 layers, each with 128 units and a dropout rate of 0.01.

Then there are two fully connected layers, between which we apply a batch normalization to speed up training and a relU activation function for the second fully connected layer.

The model is trained using the Adam optimizer and the Huber loss function, with a learning rate of 0.001. The hyperparameters(dropout rate and learning rate) were found by using optuna over 20, 50 and 100 epochs and 10 iterations, but usually after the first 5 the outputs were pruned.

**4.2.Results**

For lower age groups the model predicts pretty accurately, especially lower values.

A graph with blue dots

Description automatically generated

A graph of blue dots

Description automatically generated But for age groups which have a larger spread of data, it is biased towards the lower values

A diagram of a graph

Description automatically generated with medium confidence

The predictions it makes do increase with the extremity of the weather, but for cases where the mortality rate is high, but the weather does not seem to be under any extreme conditions it isn’t able to find a correlation

**4.3. Previous models**

The first models were fully connected with the best performing one having a 4 layer architecture with batch normalization between and a dropout rate of 0.01.

The biggest change was seen when changing the loss function from mseLoss to Huber loss, as the model was able to more accurately predict outliers.

A graph with red and blue dots

Description automatically generated

With Huber:

Loss: 0.008(MSE for normalized data)

A graph of red and blue dots

Description automatically generated

With MSE:

Loss: 0.02(MSE for normalized data)

**5.Conclusion**

In conclusion, it was possible to model a deep learning algorithm with the data provided, but further research is needed for improvement.

1. https://www.sciencedirect.com/science/article/abs/pii/S0168169920300314 [↑](#footnote-ref-1)