Predicting The Minimum Daily Temperature Of Melbourne

Jiang, Yingyuan (yxj210005)
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
TX, USA
yxj210005@utdallas.edu

Kanamata Reddy, Vishnu Vardhan Reddy (vxk210042)
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
TX, USA
vxk210042@utdallas.edu

Kotra, Sai Charan (sxk210083)
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
TX, USA
sxk210083@utdallas.edu

Manthri, Satya Sai Bharadwaj (sxm210073)
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
TX, USA
sxm210073@utdallas.edu

***Abstract*—We are creating a perfect minimum daily temperature predictor for Melbourne. Our dataset is a record of the Daily minimum temperature of Melbourne city. This is a regression problem. A RNN is being used to predict the minimum daily temperature. RNN ("Recurrent Neural Network") is a variety of ANN ("Artificial Neural Network") extracted from the Feedforward Neural Network. RNN works along with Temporal sequence, and it shows temporal dynamic behavior. RNN's performance is better than many other algorithms for this case. We used hyperparameter tuning to get the best from the RNN. We got an average test loss of 5.707 from the RNN, which is an excellent value.**

Keywords— Recurrent Neural Network, Neural Network

I. Introduction and Background Work

.

Fig 1. Melbourne

Victoria's capital is Melbourne, which is located in south-eastern Australia. The Federation Square complex, located along the Yarra River and has plazas, pubs, and restaurants, is located in the city's heart. The Melbourne Arts Precinct, is in the Southbank neighbourhood, contains the Arts Centre Melbourne, and the National Gallery of Victoria, which showcases Australian and indigenous art.

Summers, winters in Melbourne are incredibly warm, crisp and blustery, cloudy is there throughout year. The temperatures goes from 6°C to 27°C, with temperatures only here and there falls under 1°C or ascends over 35°C. In the period December 15 to March 16, the hot murkiness runs for quite some time, with an everyday least typical high temperature of above 30°C.

February is the most blazing month in Melbourne, with normal highs of 25°C and lows of 15°C. From May 23 to September 7, the cold season endures 3.5 months, with normal everyday high temperatures underneath 16°C. July is the coolest month in Melbourne, with a typical low of 12°C and a high of 15°C.

A particular day's lowest temperature is called as Minimum temperature of that day. We are trying to create a machine learning model that will predict Melbourne city's daily minimum temperature.

Machine learning develop their forecast exactness without being explicitly fabricated. Machine learning algorithms utilize past information as a contribution to anticipate new objective qualities.

How conventional machine learning is frequently classified, an algorithm figures out how to turn out to be more exact in its expectations. The four major approaches are supervised, unsupervised, semi-supervised, and reinforcement learning. The algorithm that information researchers still up in the air by the kind of information they wish to foresee.

This particular case we considered comes under supervised learning and regression-type problems. Regression is a measurable strategy for deciding the connection between free factors or attributes and a reliant variable or result.

For building the ML model, we decide to go with the Recurrent Neural Network Algorithm. Because of the following reasons:

- RNN comes under supervised learning
- RNN works perfectly for regression
- RNN is a type of ANN
- RNN uses the backpropagation technique too

RNN is a variety of ANN extracted from the Feedforward Neural Network. RNN works along with Temporal sequence, and it shows temporal dynamic behavior.

A "neural network" is a collection of algorithms that attempts to identify basic relationships during a batch of data which emulates how the human brain works. Neural networks, during this situation, see systems of neurons that may be biological or counterfeit.

As neural networks can take change in input, It can give out the most effective outcome without any other activity. The AI-based notion of NNs are becoming popular everywhere in the world.

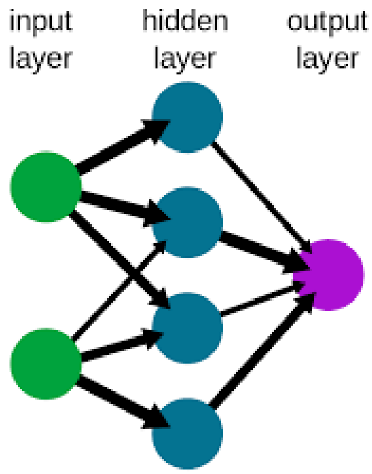

Fig 2. A simple neural network

A neural network has three components:

- Input layer.
- Hidden layer.
- Output layer.

Input Layer: Also referred to as Input nodes are the inputs/information from the surface world provided to the model to be told and derive conclusions from. Input nodes pass the knowledge to the subsequent one.

Hidden Layer: This layer is a set of neurons where all the computations are done on the inputs. There is any number of hidden layers during a neural network. Best NN will have a one-hidden layer.

Output layer: The output layer gives out outputs computed by the previously hidden layers. There will be single or multiple nodes within the output layer. Suppose we've got a binary classification problem. In that case, the output node is one, but within the case of multi-class classification, the output nodes may be over 1.

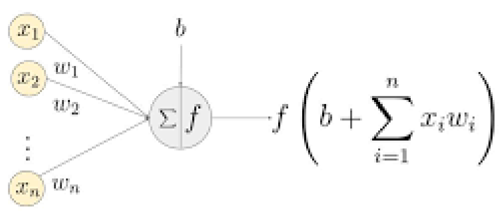

Fig 3. Working of Neural Network

The equation for the neural network could be a linear combination of independent variables and their respective weights and intercept term for every neuron. The neural network equation feels like this:

Z = W0+ W1X1 + W2X2 + …+ WnXn

where,

Wi - Weights

Xi - Inputs

W0 - Bias/intercept

There are 3 steps to do in NN:

- In order to calculate Ypred, we integrate the inputs&weights into Z = W0+ W1X1 + W2X2 + …+ WnXn.
- Compute loss/error term. (Error – Deviation between actual and predicted values)
- Reduce the loss/error term.

Backpropagation is one such essential property of the Neural Networks. Backpropagation helps us to build a Neural network with high accuracy.

A NN's weights are fine-tuned by means of back propagation, which analyzes the error-rate/loss attained within the previous iteration/epoch. As a result, the model is more reliable when the weights are properly tuned, reducing error rates.

II. THEORETICAL AND CONCEPTUAL STUDY

RNN saves the specific layer's output and feeds this back for backpropagation.

RNN was created because there have been some issues within the feedforward NN:

- NN can't handle sequential data
- NN cares only input
- NN doesn't have the power to memorize.

RNN is the one solution to the above issues. A RNN handles sequential inputs, taking this computer file and inputs which are received already. RNN memorize all inputs in internal memory.

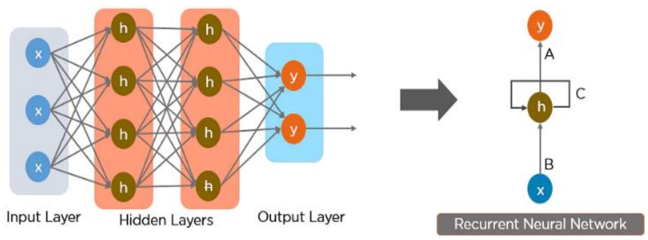

Fig 4. Working of an RNN

RNN are created by compacting the layers of nodes in a neural network. The network's parameters are A, B, and C.

Generally, Recurrent neural networks work similarly to a NN. A neural network uses the following steps for training:

1. Takes input
2. Does calculation with weights and bias
3. Gives out predicted result

4. Find error
5. Backpropagation
6. Repeat the process until convergence
7. Predict the results

In the above procedure, the only differences are some calculations and complexity.

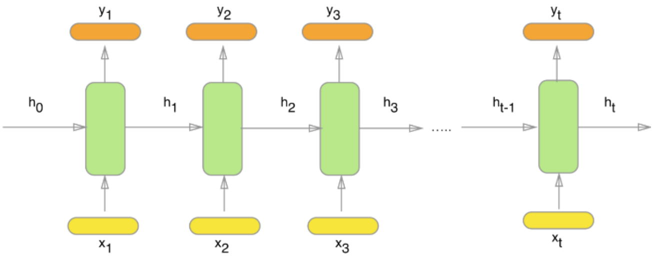

Fig 5. Detailed Reccurent Neural Network

In the above figure, *x1 to xt* represent inputs, *y1 to yt* represent predicted results, and *h0 - ht* has the information for the inputs.

Training equations of RNN are:

1) $$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$$

2) $$y_t = softmax(W^{(S)}h_t)$$

3) $$J^{(t)}(\theta) = \sum_{i=1}^{|V|} (y'_{t_i} log y_{t_i})$$

Fig 6. Equations of RNN

1st equation contains information about the sequence's preceding input. As you can see, the prior h (t-1) vector and the present input vector x t are used to calculate h t. The final summation is also given a non-linear activation function f (typically tanh or sigmoid). The assumption that h 0 is a vector of zeros is valid.

2nd equation calculates the anticipated input vector at t intervals. The softmax function creates a (V,1) vector in which all members add up to 1. This probability distribution offers us the vocabulary's most likely future input index.

3rd equation calculates the error between the expected and actual value using the cross-entropy loss function at each step t.

III. PROCEDURE

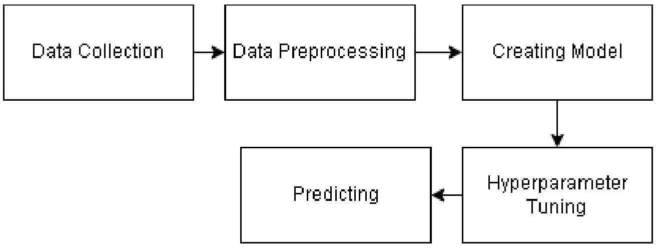

Fig 7. Workflow of building a model

- Data Collection
- Data preprocessing
- Creating Model
- Hyper Parameter Tuning
- Predicting

A. *Data Collection:*

When we decided to find the minimum daily temperature, we surfed the internet for an apt dataset. We found a perfect dataset on the city of Melbourne, Australia. The data source of this dataset is the" Australian Bureau of Meteorology." Dataset has the data entries of minimum daily temperatures from 1981 to 1990 in Melbourne city.

Dataset details:

- Number of Features - 2
  (Date, Daily minimum temperatures)
- Number of Instances - 3650

Name	Daily minimum temperatures
Count	3650.000000
Mean	11.177753
Std	4.071837
Min	0.000000
25%	8.300000
50%	11.000000
75%	14.000000
Max	26.300000

Table 1. Data distribution

B. *Data Preprocessing:*

This is a technique employed to remodel information in a valuable and well-organized format.

The steps of this process are:
1. Cleaning the Data:
Here, in this process we eliminate unnecessary and missing values of the data.

2. Transformation of Data:
Here, we perform tasks like Normalization, Attribute Selection, etc

3. Reduction of Data:
Here, we try to edit the data to save storage cost and analysis cost. This increases storage efficacy.

C. *Creating Model:*

We decided to use RNN as our algorithm to create a model, as it is the best for our case. We built an RNN without using any built-in packages.

D. Hyperparameter Tuning:

This is a process where we try all possible parameters for some methods which we give us the best outputs.

We have done this process on our parameters, and we calculated loss.

E. Predicting:

Finally, as our model is ready. We can perform our main aim of predicting the minimum daily temperatures of Melbourne.

IV. Result And Analysis

```
Epoch 0 => loss:1.2829
Epoch 20 => loss:0.8674
Epoch 40 => loss:0.8544
Epoch 60 => loss:0.8439
Epoch 80 => loss:0.8352
Epoch 100 => loss:0.8276
Epoch 120 => loss:0.8211
Epoch 140 => loss:0.8153
Epoch 160 => loss:0.8103
Epoch 180 => loss:0.8058
Epoch 200 => loss:0.8019
Epoch 220 => loss:0.7985
Epoch 240 => loss:0.7955
Epoch 260 => loss:0.7928
Epoch 280 => loss:0.7905
Epoch 300 => loss:0.7884
Epoch 320 => loss:0.7866
Epoch 340 => loss:0.7850
Epoch 360 => loss:0.7835
Epoch 380 => loss:0.7822
Epoch 400 => loss:0.7811
Epoch 420 => loss:0.7801
Epoch 440 => loss:0.7792
Epoch 460 => loss:0.7783
Epoch 480 => loss:0.7776
```

Fig 7. LOSS

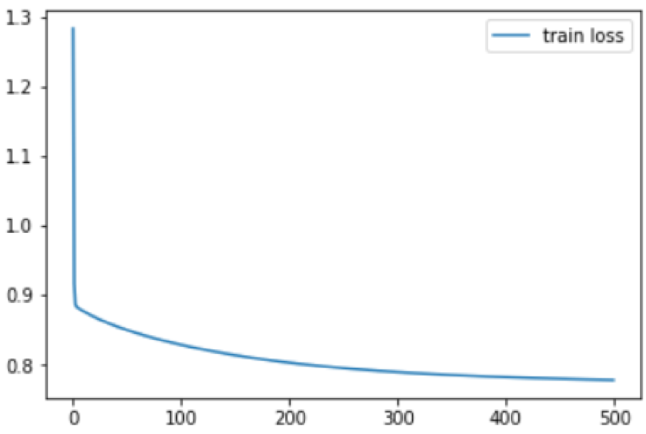

Fig 8. Training loss

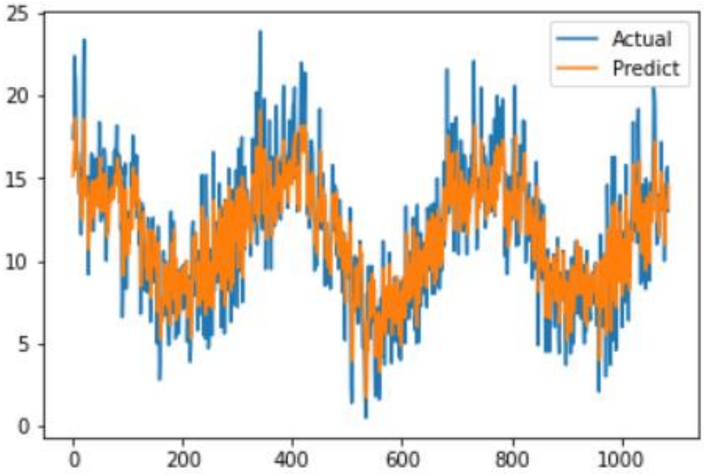

Fig 9. Actual data vs Predicted data

The test loss is 5.570578470121801.

Our RNN model uses the following parameters:

- Learning Rate
  - Values taken [1e-6, 1e-5, 1e-4, 1e-3]
- Step size
  - Values taken [30,31,29]
- epochs
  - Values taken [100, 200, 300, 400, 500]
- Number of Hidden layers
  - Value taken 50
- Activation function
  - Values taken[tanh, relu]
- Train/Test split
  - Value taken 0.7

After the hyperparameter tuning with a range of values for each parameters, the least Mean square error of test data we achieved is 5.57.

As we compared the results which can be seen the fig 9, our model produced promising results however, since its not possible to check every possible value of the hyperparameters, our model could have been stuck at a local minima.

Therefore, it can further improved by adding more training data, data preprocessing including feature engineering techniques and trying out different different neural network architectures such as GRU, LSTM and 1D-CNN's.

V. Conclusion & Future Scope

In the near future on the advancement of ML, we have scope to make a model with which can give out more results. The RNN can't remember previous data from a very far time. So, LSTM is better, which can set a threshold to store some history data from a long time ago.

REFERENCES

[1] https://towardsdatascience.com/learn-how-recurrent-neural-networks-work-84e975feaaf7

[2] https://www.analyticsvidhya.com/blog/2021/04/estimation-of-neurons-and-forward-propagation-in-neural-net/#:~:text=There%20are%20three%20steps%20to,values%2C%20called%20the%20Ypred.

[3] https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML#:~:text=Machine%20learning%20(ML)%20is%20a,to%20predict%20new%20output%20values.

[4] https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd

[5] https://machinelearningmastery.com/time-series-datasets-for-machine-learning/