PREDICTING CLIMATE AND WEATHER USING MACHINE LEARNING

McNeil, Ethan

Department of Physics and Astronomy, University College London, Gower Street, WC1E 6BT

Word Count - 5212

Table of Contents

Abstract – *Machine learning techniques were successfully applied to time series weather data to make predictions about the future. Neural network's ability to weekly and monthly averaged temperature data better than assuming the next month will be the same was proven by returning lower mean absolute errors. On daily timeframes the model straggled due to the more stochastic nature of the daily data. Precipitation data proved difficult to model alone however considering the interdependencies with temperature data, better predictions could be made.*

I Introduction

The purpose of this project is to apply machine learning techniques to datasets of historical weather and climate metrics to make predictions about the future. It is important to distinguish the difference between weather and climate; in the context of this project, weather describes the immediate weather conditions on a daily timeframe whereas climate refers to longer-term trends – over months, years, and decades. This project attempts to characterise and analyse the effectiveness of machine learning as a technique to facilitate these predictions.

Predicting weather and climate has vast and far-reaching applications. On short timeframes, daily weather changes influence industry, people's lives, and daily activities. On longer timeframes, understanding the climate is essential and underpins industries such as agriculture and allows people to be prepared for seasons of high risk such as flooding or extreme cold or heat. Pertinent to contemporary society, it also allows us to analyse the effects of human activities and therefore mitigate and adapt accordingly.

I.1 Why machine learning

At the core machine learning is a programming technique that 'teaches' machines to make predictions and decisions. A machine learning algorithm mimics the structure of a human brain, consisting of neurons that together form a 'neural network' (a brain). It removes the need to use unadaptable rule-based systems and can handle enormous quantities of data, as will be seen through this project.

By feeding our network with data, it learns in three main ways: supervised learning (learning with training sets that contain labels corresponding to the correct output) unsupervised learning (learning with unlabelled data. In this case the network must identify any hidden patterns or structures without labels) and reinforcement training (the network is forced to make decisions and is 'rewarded' when correct and has 'penalties' applied when incorrect decisions are made).

In the context of this project, a neural network is apt due to the large quantities of data required to be processed, and underlying patterns in weather and climate data that can be exploited. Further to this, high volumes of data allow for plenty of training material, ensuring the network can be sufficiently trained.

A neural network's ability to identify patterns aligns well with weather data which contains seasonal to daily structures that can be extrapolated to make predictions. In short time frames, however, the ability of a network to make predictions begins weakening. Long-term climate, daily and hourly weather becomes rather unpredictable despite contributing to more predictable longer-term trends. This will be investigated throughout this project.

I.2 Neural Networks in Depth

As discussed, networks are comprised of neurons. These neurons are arranged into layers that subsequently make up the network. When all the neurons in a layer are interconnected this is called a 'dense layer' and when a network comprises only dense layers, this is called a 'fully connected network.' [1] Values are fed through the network from an input layer and outputted in an output layer. The intermediate collection of layers is usually referred to as a ‘black box’ for simplicity. [fig 1]

Fig 1, visualizing a neural network “black box”

Neurons possess two main properties, a weight, and an activation function. The activation function is a mathematical function that determines the output of a single neuron. These can be linear or sigmoidal amongst other forms. Non-linear activation functions allow the network to learn complex relationships within datasets [1].

The weight of a neuron is initially an arbitrary value; however, it is updated as the network learns and is indicative of the importance and strength of the neuron output. The weights are adjusted in order to minimise the error associated with a prediction made by the model with reference to known labelled data throughout the training process. This process is repeated iteratively, reinforcing the learning, and subjecting the model to new data [2].

Throughout this project, various types of neural networks that are adapted to specific tasks are implemented. At the forefront of this project are LSTM (Long short-term memory) neural networks. These are networks that are very well adapted to time-series data, as is our weather data.

LSTMs are a form of recurrent neural network (RNN). Each LSTM unit has a memory cell that can store information for long periods. This allows LSTMs to remember and learn from information over long intervals making it effective for time series data where gaps between relevant information may be large.

LSTM cells are comprised of multiple collaborating cells, unlike traditional neurons. Below is a simple LSTM cell consisting of Candidate gates (C), Hidden gates (H) and Input gates (x). The flow of data through these gates is governed by the sigmoid neural net layer that regulates information passing through the cell. The cell state is passed along the chain of LSTM nodes, interacting with the various gates and retaining the important data. [3][4]

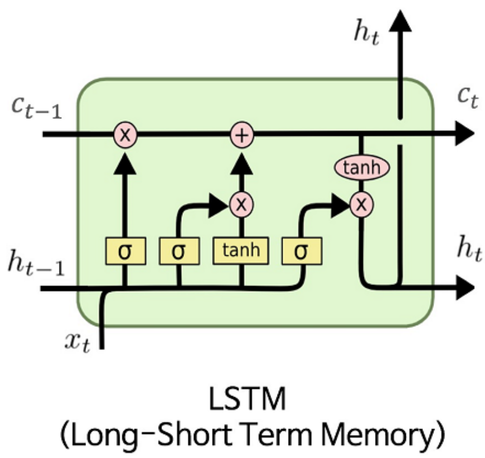


Fig 2, showing schematic of an LSTM “neuron”

I.3 Data Format

Our prediction model is trained on a database from the GHCN [5] (Global Historical Climatology Network. GHCN is a database of daily weather recordings from 990 weather stations across the globe (see below for global distribution). GHCN contains various metrics used to describe the weather and climate, for the purpose of this project maximum daily temperature (TMAX), minimum daily temperature (TMIN) and daily rainfall (PRCP) were used. Below is an example of formatted data extracted from the GHCN archive provided.

Date	PRCP	TMAX	TMIN
1946-08-26	0.0	19.8	0.0
1946-08-27	0.0	14.8	8.3
1946-08-28	0.0	14.5	4.3
1946-08-29	0.0	16.3	4.3
1946-08-30	0.0	18.3	3.3
...	...	...	...
2020-11-10	0.0	37.1	23.0
2020-11-11	0.8	32.9	23.3
2020-11-12	0.0	27.9	15.1
2020-11-13	0.0	24.3	11.2
2020-11-14	0.0	29.0	9.2

27006 rows × 3 columns

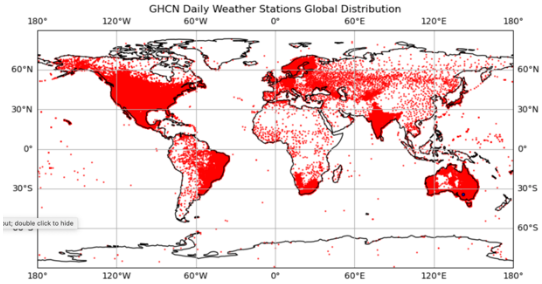


Fig 3a/b, Showing example of data frame format for reference and scatter plot of all 990 weather stations and the station analysed in this project (blue (Australia)).

II Method and Objectives

This project attempts to apply machine learning techniques to historical weather data in order to make predictions about future weather. As mentioned above, the weather data is obtained from GHCN [5].

The main goals of this project were to predict climate metrics a year in advance using a machine learning model and evaluate the accuracy of this. Predictions up to 10 years in advance wren carried out to test the limits of the model.

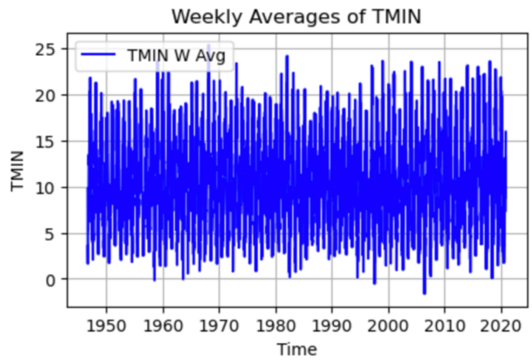
The next step was to compare whether the machine learning model performed better than just assuming that the weather at one time-step ahead was the same.

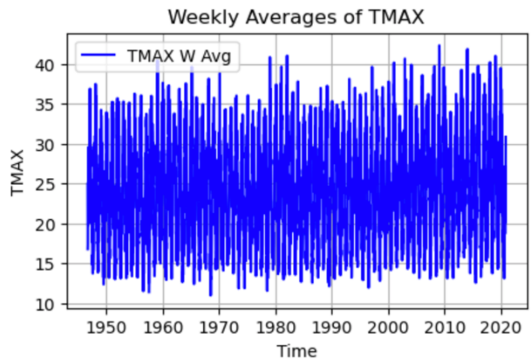
Predicting on precipitation data posed unique difficulties and so a new training regime was designed to make valid predictions on precipitation using both known and predicted temperature data.

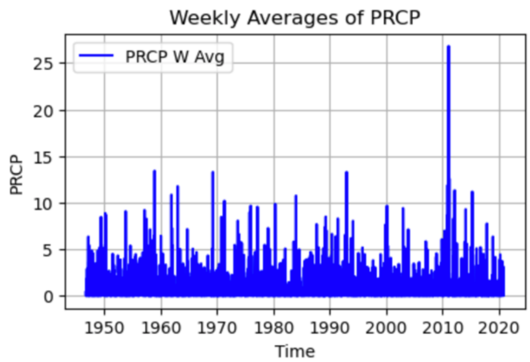
These steps were applied to the three climate metrics described above. The findings and method follow.

II.1 Data Extraction

Initially, I had to extract data from the data set. This was done using the read file functi on. This function takes the file path directo ry and returns a data frame of dates and the climate metrics mentioned above. The Pan das library is imported and used to facilitat e this. This data can be plotted and visualis ed using the averages_plotter function. Thi s function is designed to plot daily data or weekly, monthly, or yearly averages of TM AX, TMIN, or PRCP. For demonstration p urposes, I used ASN00076031, Mildura Ai rport, Australia. This is easily changed and any of the other 990 locations can be used with this code.







Figs 4a/b/c, showing weekly averages of TMAX, TMIN and PRCP for ASN00076031

II.2 Cleaning Data

Importantly, at this step, the data must be cleaned to ensure NaN entries are dealt with. These NaN values create 'holes' in the dataset, causing an interruption in the time series nature of the data. Further to this NaN values cannot be used with a neural network and will return an error. This is done using the function interpolate_time_series. interpolate_time_series utilizes linear interpolation to fill gaps in data. Linear interpolation is useful for filling relatively small gaps in data sets. When a gap in data is present, it is filled by assuming the relationship between the points is linear, inserting a new point according to this relationship. This is done both by reading the data forward (forward fill) and backwards (backward fill). Any remaining NaN value after interpolation is replaced with a median of the preceding and following points.

Once cleaned the data can be used to teach our network, enabling it to identify patterns and trends. The train_and_test_predict function is defined to do this. The make_timeseqs function takes the raw data frame and ensures it is correctly shaped to be compatible with train_and_test_predict function. This function separates the data into windowed time sequences and an offset point. Windowing the data allows the LSTM and network to accept data in a manageable format. Further to this, training on sections of data increases chances of local structure recognition and allows of real time data prediction if needed.

The offset can be set an arbitrary distance into the future and is predicted upon using the time sequence and acts as the 'training label' for this sequence. The data is used to train the model and then make predictions on the test set. Comparisons with the true data is made and conclusions about the quality of the model are drawn.

II.3 Model Design

The model itself was designed to be optimized for time series data. The model summary can be seen in [Fig 5]. LSTM layers and regular dense layers are both used. The number of training epochs is set to 15 and the batch size for all modelling was set to 70. A smaller batch size runs quicker however this exposes the loss function to larger fluctuations during the minimization process. Increasing the batch size reduces this issue but takes more computational power. Throughout this project a batch size of 70 and 15 epochs were used, as this proved a happy medium between model speed and producing the desired outputs.

```
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_8 (LSTM)               (None, 128)               66560

 dense_32 (Dense)            (None, 64)                8256

 dense_33 (Dense)            (None, 64)                4160

 dense_34 (Dense)            (None, 64)                4160

 dense_35 (Dense)            (None, 1)                 65

=================================================================
Total params: 83201 (325.00 KB)
Trainable params: 83201 (325.00 KB)
Non-trainable params: 0 (0.00 Byte)
```

Fig 5, showing the model structure, each component is discussed in more detail below.

II.3.1 Input Layer

The input layer consists of LSTM neurons accepting an input shape of one training window of data. From the code, we can see this as input_shape=(time_seq, 1).

An LSTM layer is used to help identify long-term trends using its memory capabilities. The LSTM cell retains data that can be recalled later in the training process.

II.3.2 Dense Layers

Fully connected dense layers follow this. These layers learn patterns and trends in the data, each layer subsequently attempting to learn higher-level trends.

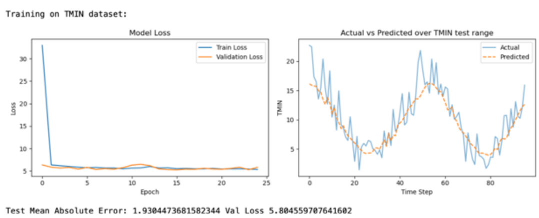
II.3.3 Output layer

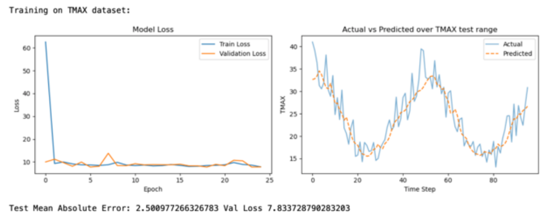
The output layer of a single neuron allows for the output of a single value, in this case, a single value, our weather metric prediction.

III Results and Analysis

III.1 Predicting climate one time step in advance

My first training and testing run was to train the model on a year of weekly averages (time_seq = 12) and predict the average for the following week (offset = 0). This was carried out using a test set of 96 weeks, which can be seen in [Figs 6]. The metrics used to analyse the quality of the model were mean absolute error (MAE) and value loss. The MAE allows for comparison between predicted data and unseen test data and is useful for comparing the quality of models with different offset values. The validation loss signifies how well the model predicts unseen data.





Figs 6a/b showing predictions on TMIN and TMAX datasets on a weekly timeframe. time_seqs = 12 and offset = 0, providing prediction one time-step in advance

From [Figs 6], the model is well trained on the data and can accurately predict one time-step ahead for both TMIN and TMAX on weekly averages.

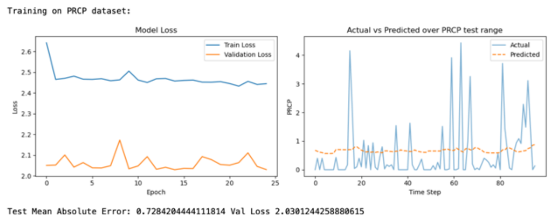


Fig 6c showing predictions on PRCP dataset on a weekly timeframe. time_seqs = 12 and offset = 0, providing prediction one week in advance. NOTE - poor predictions made.

For the precipitation data, however, it would be fair to say the model struggles to provide meaningful predictions. The mean absolute error value is low; however, this doesn't signify a good model as can be seen from [Fig 6c]. The predictions appear to hover around a mean value. The mean of the test set was calculated to be 0.78, which roughly aligns with the predicted values. This suggests the model failed to learn any structures other than an approximate value for PRCP at a given time.

Upon visual inspection of the PRCP graphs [fig 6c], they appear far more stochastic than the temperature graphs. The TMIN and TMAX data follow the seasons with regular sinusoidal patterns, allowing the neural network to learn this structure easily. In contrast, the precipitation data doesn't appear to possess any regularity.

To make better predictions on precipitation, the model was trained on monthly averages of PCRP to attempt to encapsulate any long-term trends. [Fig 7] Whilst slightly better than on a weekly timeframe, the model still appears to struggle with predictions. There appears to be some correlation between small peaks in the predicted data and the larger actual peaks, with a slight offset, however the predictions don't provide much insight into the actual structure of the PRCP data and as before the predictions lie around a mean value of 0.78, implying constant rain (which is obviously wrong).

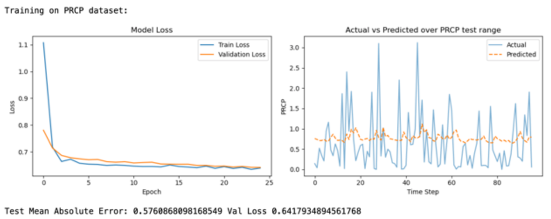


Fig 7 showing predictions on PRCP dataset on a monthly timeframe. time_seqs = 4 and offset = 0, providing prediction one month step in advance. NOTE – predictions still quite poor.

III.2 Predicting climate one year in advance

The results from predicting weekly climate data a year in advance can be seen below. For these predictions, time_seq = 12 and offset = 52. The findings are very similar to those predicting just one month ahead.

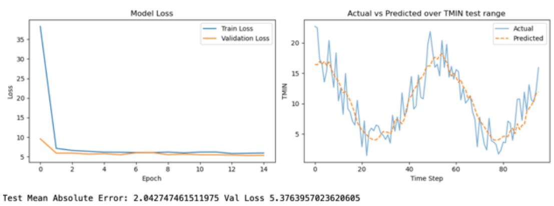


Fig 8a showing predictions on the TMIN dataset on a weekly timeframe. time_seqs = 12 and offset = 52 providing prediction one year in advance

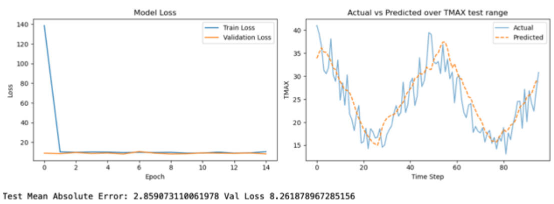


Fig 8b showing predictions on TMIN and TMAX datasets on a weekly timeframe. time_seqs = 12 and offset = 52, providing prediction one year in advance

The predictions are very similar to predicting one time step ahead, with slightly higher loss metrics which may as expected due to increased chances if long term trends changing with a larger offset.

I would hypothesize that, due to the magnitude of the dataset (thousands of weeks of data) and the long-term high regularity of the data, altering the prediction offset from one week to one year would have a negligible effect on the quality of the model's prediction.

For the PRCP data, again monthly averages were used, and as before no meaningful conclusions can be extracted.

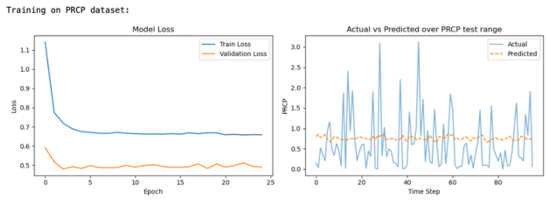
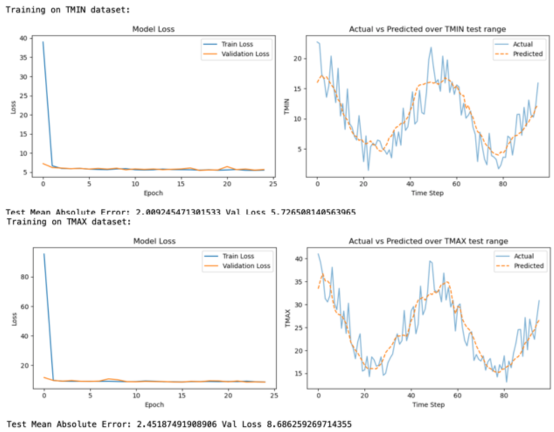


Fig 8c showing predictions on PRCP dataset on a monthly timeframe. time_seqs = 4 and offset = 12, providing prediction one year in advance. NOTE – poor predictions made.

I return to an updated method for predicting on the PRCP dataset later.

III.3 Predicting climate ten years in advance

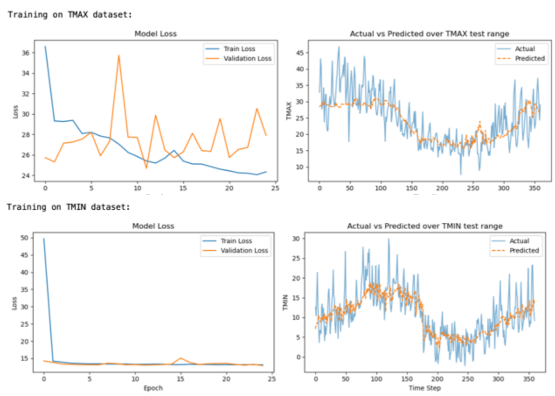
Lastly predictions made 10 years (offset = 520 and time_seq = 12) in advance were made as an extension. As before for both TMIN and TMAX, the predictions were of a good quality and were very poor for PRCP. Again 10 years is small compared to the magnitude of the entire dataset. For the temperature data, the periodicity and magnitude of the oscillations do not vary much over a period of 10 years, allowing valid predictions to still be made. It is possible, from visual inspection, a slight shift between the predicted data and actual data can be observed which is not present on the shorter prediction timescales. Perhaps long-term seasonal pattern changes caused by climatic oscillations could impact when seasons commence and end. [6]



Figs 9a/b showing predictions on TMAX and TMIN datasets on a weekly timeframe. time_seqs = 12 and offset = 520 (10 years), providing prediction 10 years in advance

Shortcomings were identified in areas such as predicting a year in advance on a daily timeframe with time_seq = 30. This returns an MAE of 3.51, significantly higher than using weekly averages [figs 8] This is partly due to the sequence lengths having to be much smaller. For weekly predictions time_seq = 12 is used to return accurate predictions; averaging one point every 7 days, whereas on the daily timeframe 1 point is used per day. This means 7 times as much data must be used to train the model on long-term trends.

Further to this, daily data is much more suspectable to seemingly random fluctuations, whereas over longer time frames these fluctuations are averaged out providing a smoother curve. Despite contributing the longer-term overall trends, on a daily level, it is hard to identify structure on daily timeframes [figs 10].



Figs 10a/b showing predictions on TMIN and TMAX datasets on a daily timeframe. time_seqs = 30 and offset = 3600, providing prediction 10 years in advance

For precipitation data, as before, no accurate predictions could be made by solely training on the PCRP dataset on any timeframe. I will explore later how to improve PCRP predictions.

III.4 Evaluating Predictions

Comparing the quality of the LSTM model with assuming the weather at the next time step will be the same as the previous. This is an arbitrary form of predicting the next day's weather, however, it turns out to be relatively accurate – think from experience, if it's warm one day, it will likely be warm the next. Thus, any model that can predict better than this, may have some merit.

From the mean absolute error (MAE) between the true data and predicted data, we can compare this with the mean absolute error of assuming the weather at the next time step was the same as the previous. This was achieved by defining a function called mae_current_next; which simply 'rolls' the actual data set to the right and uses this new dataset to calculate the mean absolute error with the original dataset.

Applying this function to the daily TMIN values of the ASN00076031 dataset, an MAE error of 2.79 was given.

Training the model with a training sequence of 1 day and an offset of 0 gives an MAE of 2.95 and a Val loss of 12.18 [fig 11].

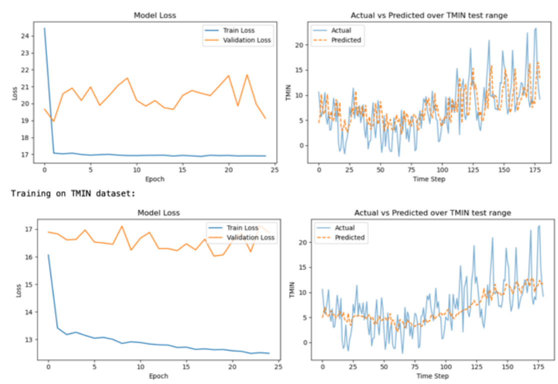


Fig 11a/b showing TMIN predictions, on a daily timeframe with offset = 1 and time_seqs = 1 (a) and offset = 1 and time_seqs =

30 (b) over a 100-day testing period. (100 days was chosen as it is long enough to encapsulate long term trends and short enough to see shorter term trends)

Increasing the sequence length, decreases the MAE however it drastically increases the training time. At time_seq = 30, the MAE drops to as low as 2.52 and the Val loss to 9.16. Interestingly, the 30-day window loses the detail of daily fluctuations but has a more defined longer-term trend whilst the 1-day window captures daily fluctuations well but loses clarity over the long term trend.

This high MAE Is reflective of the small timespan of the training sequence. Despite containing 30 training points, it is hard to practically identify long term patterns and structures from only one month of daily data, as discussed previously.

Using daily averages requires a lot more computational power due to the increased number of data points over a given period. To begin to identify a long-term trend that the neural network can learn, hundreds of daily data points must be used. Beyond 30 data points, the training and testing time becomes far too long to be reasonable.

Using weekly averages however, we can train a model to predict much better than assuming the next week will be the same as the current. From [Fig 6], in which we trained a model over 12 weeks, we can see the MAE of predicting one week in advance is 2.02 for the TMIN data. Using mae_current_next, a value of 2.46 was obtained, implying training the model can yield far more accurate predictions on these larger time windows. This is likely partly because data averaged over a week ‘smooths’ it out and reduces outliers, providing a much more regular series of data that the network can learn from. Further to this, weather on larger time scales is prone to more change from longer-term patterns, so the idea of assuming the next time step will be the same loses validity.

I did not subject PRCP data to this test since no accurate models could be procured and thus no valid conclusions can be drawn.

IV Better Precipitation Predictions

IV.1 Using Known Temperature Data

To extend the project and aim to better predict precipitation data, I postulated that temperature and precipitation are not mutually exclusive events. Thus, by training a model on all the metrics at once, perhaps more accurate predictions could be made.

As PRCP data doesn’t appear to exility show any long-term trends, I decided to explore whether any dependence of PRCP on temperature metrics could be exploited to make better predictions on PRCP.

In short, can better predictions at a given timestep of PRCP be made, given a model that is trained on TMAX, TMIN and PRCP given also current temperature metrics.

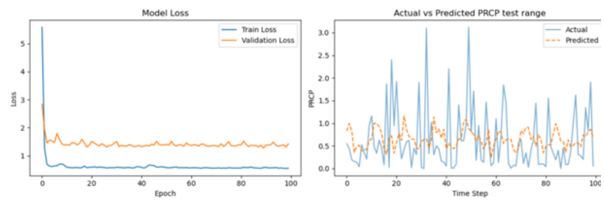
IV.2 Model

The model used had the same structure to that used in Part (II.3) and shown in [Fig 5], with some key differences, however. The input shape had to be altered in order to satisfy the higher dimension input on the multi-column data frame. In this case input_shape = (train_data.shape[1], train_data.shape[2])

Initially, the model is trained on TMIN, TMAX and an average of TMIN and TMAX called TMIN_TMAX_AVG and a ‘lagged’ PRCP dataset. All these metrics had time sequences of 20 and a prediction offset of 1. The lagged PRCP dataset is the regular PRCP dataset with all values shifted forward one time step. This allows us to make predictions of the following PRCP value given current time steps TMIN and TMAX values and the previous PRCP value.

IV.3 Predicting one timestep ahead

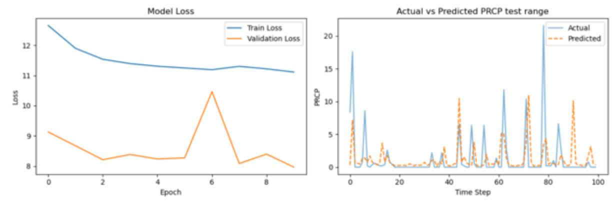
Initially the model was trained on monthly averages [Fig 12]. When compared with [Fig 7], it is hard to conclude which models PRCP data better. It appears the updated model may have attempted to identify an oscillating trend, however many of the peaks do not line up with the actual data. The mean absolute error is lower than previously, perhaps signifying a better fit however the validation loss is higher signifying the model may struggle on unseen data. As before, although not the same extent, the predictions fluctuate around the mean value of 0.78.



Test Mean Absolute Error: 0.5345320971413383 Val Loss 1.4196254014968872

Fig 12 showing PRCP predictions, training on known temperature and PRCP data on a monthly timeframe with offset = 1 and time_seqs = 20 over a 100-month testing period.

As evident in [Fig 13], this model seems to work far better on daily timeframes. Over longer time frames of weeks and months, predicting rain at the following timestep may be harder as the average conditions of the previous timestep may not have much influence on the next. However, on a daily scale, the model may be able to identify signature behaviours that precede rainfall events. For example, warmer or colder temperatures than usual combined with low rainfall over the training window may signify rain the following day. The model has also moved away from predicting fluctuating values around the mean of 0.78. In this case the model has learnt that low values and tall peaks are possible, whilst conserving the mean.



Test Mean Absolute Error: 1.6165140866041185 Val Loss 7.964073181152344

Fig 13 showing PRCP predictions, training on known temperature data as well as PRCP data on a daily timeframe with offset = 1 and time_seqs = 20 over a 100-day testing period.

IV.3.1 Evaluating Daily Prediction

Despite the MAE and Val loss being higher, the structure of the predicted rainfall is much more akin to that of the actual data than when compared to previous predictions. Whilst the magnitude of individual peaks is not accurate in most cases, the frequency, distribution, and shape are far more accurate suggesting the model has learnt nuances of the data that it could not previously identify.

The model correctly predicts some level of rainfall in 13 out of 14 of the actual peaks. There is one instance of a ‘false peak’ in which the model predicted rainfall, however none was actually recorded.

The MAE of these predictions when compared with the actual data is 1.74. When assuming the next step is the same as the previous a mean absolute error is 1.26. This low value of 1.26 is partly due to thee being many successive days of zero rainfall, as seen in [Fig 13], meaning that the assumption that consecutive days will have the same rainfall is a good one and thus making it very hard to train a model to predict better than this.

IV.4 Predicting two days ahead

Increasing the prediction offset drastically effects the validity of the models’ predictions. This may be since older PRCP data has an increasingly smaller impact on rainfall in the future.

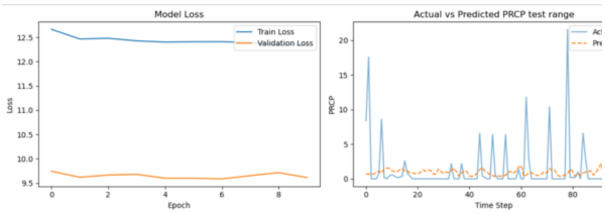


Fig 14 showing PRCP predictions, training on known temperature and PRCP data on a daily timeframe with offset = 2 and time_seqs = 20 over a 100-day testing period.

It could be concluded that, on this timescale, TMAX, TMIN and PRCP may exhibit identifiers that precede the following precipitation events very closely and are closely interdependent.

It is hard to conclude at a given time whether a rainfall event will occur in two days later given current temperature and precipitation metrics from two days prior.

IV.5 Using Predicted Temperature to Predict PRCP

In the above predictions we are assuming we know the temperature at the prediction time step, so it would be more meaningful to use predicted temperature data to make further predictions on PRCP. Using predicted temperature data to predict PRCP allows us to make complete future predictions that do not rely on knowing any data about the prediction period. pred_prcp_using_pred was defined to do this.

Using predictions of temperature, obtained from the train_test_predict function, a new data frame was created combining the predictions along with the actual lagged PRCP Values. To ensure the values all corresponded to the correct time step I used the last 100 values of the original data frame as my test set.

This data frame was then predicted on using a pre-trained model. This is the same model used to predict with actual temperature data in Part (IV.2), as this model was trained on known data, allowing it to identify the relevant structures and interdependencies between temperature and precipitation.

The PRCP predictions are not as good quality as in Part (IV.3) There are no prominent peaks however the form of the data appears somewhat valid. The predicted data is not structured as in Parts (III.2) and (III.3) in which predictions oscillate around a mean value.

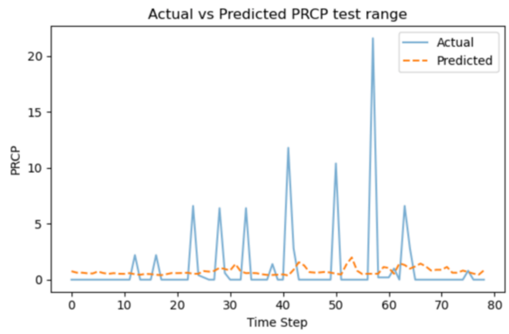


Fig 15 showing PRCP predictions, predicting on predicted temperature data as well as known PRCP data on a daily timeframe with offset = 1 and time_seqs = 20 over a 100-day testing period and using a model trained on known data.

Analysing the predicted data used to make these further predictions, we can see that the modelled temperature data misses detail provided by the actual data. Whilst the overall trend is correct, the predicted data is rather under exaggerated.

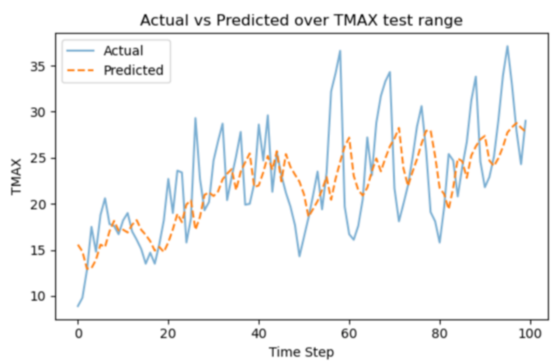


Fig 16 showing TMAX predictions (orange) and actual data (blue) used to make more meaningful perditions of PRCP

This is important as these small details and patterns may provide some dependency for the PRCP data and so missing them may affect the accuracy of our PRCP predictions. This could be amended by better training of the temperature model or considering other weather metrics that fluence temperature to achieve better predictions. For example, the roles could be reversed and known PRCP could be used to make even more accurate predictions of temperature on short timeframes.

V Conclusions

On small scales, temperature exhibits chaotic behaviours that over larger time frames are averaged out, creating long terms trends and patterns. The ability for a neural network to predict weather on short timeframes, such as daily, is limited by this chaos. Over long timeframes however, periodic trends that form are easily predicted by neural networks. The periodic nature of temperature over long timeframes is easy for neural networks to learn.

The increased ability of the network to make predictions is reflected by the decreasing MAE on longer time frames.

It proves very hard for machine learning models to identify trends in solely precipitation data. Unlike temperature, no largescale patterns are easily identifiable in precipitation data. It appears however, combining temperature and precipitation data allows interdependencies to be learnt and exploited when making predictions the day of or before. This implies the model may recognized behaviours in the temperature and precipitation data sets that precede and together signify rainfall events.

Increasing the prediction offset of this model however rapidly decreases the validity of the predictions and a new model would have to be devised to tackle this. This new model would have to consider offset values of temperature alongside lagged precipitation values and it will likely remain very hard to predict precipitation with an offset larger than one day.

Neural networks can be an incredibly useful when predicting weather. As seen, they must be used on accordance with their strengths. By definition, neural networks learn on patterned and structured data and very much struggle with stochastic datasets. These patterns and structures may not be obvious; thus it is important to try to identify places where they may be concealed before training.

References

Fig [2] Rahuljha (2020). *LSTM Gradients*. towardsdatascience. Available at:

https://towardsdatascience.com/lstm-gradients-b3996e6a0296.

[1] Alex Wilson. (2019). *What is a Dense Layer in Neural Networks?* Available at: https://www.alexdwilson.dev/learning-in-public/what-is-a-dense-layer-in-neural-networks [Accessed 2 Apr. 2024].

[2] Sharma, Sagar, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks." https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf*Towards Data Sci* 6.12 (2017): 310-316.

[3] Francis, Louise. "The basics of neural networks demystified." https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=cd6633e9872468d411aab8e13e18e05e3afbad84*Contingencies* 11.12 (2001): 56-61.

[4]Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." https://direct.mit.edu/neco/article/31/7/1235/8500/A-Review-of-Recurrent-Neural-Networks-LSTM-Cells*Neural computation* 31.7 (2019): 1235-1270.

[5] https://www.ncei.noaa.gov/products/land-based-station/global-historical-climatology-network-daily

[6] Qian, WeiHong, and Bo Lu. "Periodic oscillations in millennial global-mean temperature and their causes." https://link.springer.com/article/10.1007/s11434-010-4204-2 *Chinese Science Bulletin* 55 (2010): 4052-4057.