Adapting Machine Learning Techniques to Predict the Weather

Mini-Project: Climate Group

Name: Maruf Talukdar

Date: 06/01/2022

Contents

[Aims: 2](#_Toc93323570)

[Objectives: 2](#_Toc93323571)

[Introduction: 2](#_Toc93323572)

[Understanding the model: 3](#_Toc93323573)

[Method: 5](#_Toc93323574)

[Findings and Results – Part 1: 6](#_Toc93323575)

[Findings and Results – Part 2 10](#_Toc93323576)

[Conclusion: 12](#_Toc93323577)

[References: 13](#_Toc93323578)

# Aims:

The primary aim in this study is to create a machine learning algorithm that can predict the weather (defined as the weekly or monthly average) one year in advance. The second component of the study is to create a machine learning algorithm that can predict the weather better than assuming that the weather tomorrow will be the same as the weather today.

# Objectives:

The objective in this experiment is to build a machine learning model that trains on weather data from previous years from stations around the world and uses that data to predict what the future weather is going to look like. This study will output the predicted weather as both a monthly and a weekly average. The mean squared error statistic will be used to evaluate this model.

At the same time, a new model (one where it is assumed that the weather today is the same as the weather the previous day) will be created. The same mean squared error statistic will be used to evaluate this model. Finally, a numerical comparison between the two will determine whether a model built with machine learning techniques predicts whether better than assuming the weather tomorrow will be the same as the weather today.

# Introduction:

The data used in this study is obtained from Global Historical Climatology Network (GHCN) public database. The GHCN collects daily weather data from 100 000 stations dispersed over 180 countries. While some have only recently started recording data, the oldest stations have been recording since 175 years ago. The variables they record include (but not limited to) the maximum and minimum temperature, the daily precipitation and snowfall. Their data is also regulated by a common suite of quality assurance reviews.

Weather forecasting is predicting future atmospheric conditions at specific locations and times. There are many factors to consider when forecasting weather. Weather in a certain location could be affected by weather conditions on the opposite side of the world. Some of the factors to consider are pressure, temperature, tides, cloud cover, wind speed and direction. Due to the chaotic nature of the atmosphere and the large numbers of factors to consider, weather forecasting has been greatly inaccurate. In recent years, as technology improved, weather forecasting has switched to computer-based model, which are able to take into account more factors. The above means that weather forecasting has become increasingly more accurate. Still, the immensely large amount of computational power required to describe the current atmosphere, land and ocean means that we are still far from being able to predict the weather with 100% confidence. This confidence decreases exponentially the further in the future we try to predict.

This study takes a different approach to weather forecasting. It uses a ‘pattern recognition’ method to predict the weather. A machine learning algorithm will attempt to predict the weather by studying data from one location for one variable only, completely ignoring all other factors. Although the GHCN records many variables, they often don’t have enough data for a machine learning model to train on, therefore, the only variables that this study is going to train the model on is the maximum and the minimum temperature.

Weather forecasts have many applications. Projections could be made about droughts, spikes in temperatures etc… All of which would affect agriculture and hence the commodity market. Early warnings on agricultural challenges could prevent famines. Being able to predict which zones are likely to flood means we can take measures to decrease flood risk, evacuate the area. Blackouts would be more common if utility companies did not have temperature forecasts. Weather forecasts are also used in everyday life ranging from when and where to go on holidays down to what clothes to wear tomorrow or whether an umbrella is necessary.

# Understanding the model:

The machine learning model will be built on Python. Python was used due to its English like syntax, which makes it easy to learn and manipulate arrays of data. It executes the code line by line and stops when an error is spotted. This helps in debugging. Python also automatically assigns data types at execution. Although the line-by-line execution and automatic data type assignments make Python run slower than other languages (as it must complete extra tasks), the benefit of the added simplicity outweigh the slower run time. The amount of data that will be analysed in this report (although large) is not so large where the slower speed makes any noticeable difference.

As a trade-off to the increased simplicity, Python has a sub-standard efficiency when using available memory space. However, as mentioned previously, due to the limited size of the data, memory size will not affect this study in any way. Regardless of all the above advantages, the biggest benefit to Python is its large (over 100 000) standard libraries and its even larger (over 200 000) external packages. TensorFlow is one such external package.

TensorFlow is an open-source platform for machine learning. It allows to easily build and train machine learning models, widely used in large companies such as NASA and YouTube. Keras is a high-level application programming interface (API) running on top of TensorFlow. Like python, Keras focuses on simplicity, while retaining the essential tools to build customisable machine learning models. The machine learning model used in this study was built with Keras.

The simplest type of model is the sequential model, where layers are linearly stacked on top of each other. A sequential model is best fit when there is one input tensor and one output tensor such as in our case.

To predict future weather data a sequence model will be used (not to be confused with the abovementioned sequential models, which is a type of model in Keras). Sequence modelling is when a computer program can predict a sequence of data. For example, translating a sentence from English to French, completing a sentence, or predicting responses to common questions are all example of sequence modelling. The ‘autocorrect’ feature, available in almost all modern smartphones were built using sequential modelling. Long Short-Term Memory (LSTM) is a common algorithm used in sequence modelling. The first layer used in the model in this study is also an LSTM.

The advantage to LSTMs (compared to standard neural networks) is that the weights are shared across time. LSTMs can remember previous inputs for long periods of time, something a standard neural network is unable to do. LSTMs are also able to manipulate its memory. LSTMs have 3 gates, namely the:

1. Forget gate: used to remove information,
2. Input gates: used to add further information to the input,
3. Output gate: used to selectively output stored information.

The gating system described above allows LSTMs to selectively remove, change and output information and is what makes LSTMs one of the best at making time series predictions.

The next layer used in the model is a Flatten layer. A Flatten operation acted on a multidimensional input, would change it so the output has the same number of elements in one dimension. The Flatten layer is added so that the data can be input into the next Dense layer, which only accepts one dimensional data.

Chart, bar chart

Description automatically generated

Diagram

Description automatically generated

*[1a left]: shows how the Flatten layer changes the elements. If the input has shape 3x16, then the output is an one dimensional array with 48 elements.*

*[1b right]: is a cartoon showing how Flatten works.*

As mentioned before the next layer used in this model is the Dense layer. The Dense layer is described as the dot product between the input tensor and the weight matrix added to a bias vector, all passed through an activation function. The number of parameters in the Dense layer equals to the input shape multiplied by the weight matrix. The Dense layer is a deeply connected neural network, meaning that neurons in the dense layer receive inputs from all neurons in the previous layer. Equation (1) below shows how the Dense layer is computed.

(Dense layer) Output = f(W.X + b)

*Equation (1): Shows how the Dense layer is calculated.*

*W = Weight matrix, X = Input tensor, b = Bias vector, f = Activation function*

Text

Description automatically generatedThe metric used to evaluate this model is the Mean squared erroe (MSE). This is the weighted sum of the square of the difference between the predicted value and the real value. Equation (2) below shows how the MSE is calculated.

*Equation (2): Shows how the MSE is calculated.*

The ‘squaring’ removes the effect of the negative sign and places more weight on larger differences. The MSE is used to measure how close forecasts are to real values. The lower the MSE the more accurate the forecast is. An MSE value of 0 would indicate that the model is perfect. It would mean that the predicted weather is exactly equal to the real weather. There is no ‘good’ MSE, simply the lower the better. This is because MSE varies wildly based on the field and data that is being used on.

# Method:

The first variable this study will predict is the maximum temperature at the Clyde River Station situated in Canada. This station was chosen at random. The data will be split into training data which will encompass 75% of the available data and testing data (which will include the remaining 25%). The training data will be used by the machine learning algorithm to recognise patterns. It will study the way the maximum temperature changes over time. Then it will predict the ‘future’ maximum temperature. To evaluate the accuracy of the model, it is necessary to be able to compare predicted values with real values, this is why 25% of the available data has been split as the testing data.

For example, if the available data ranges from 2010 to 2014, then the machine learning model would train on data from 2010 to 2013 and use this data to predict the temperature from 2013 to 2014. But since real data for 2013 to 2014 is already available, then a comparison can be made between the predicted value and the real value.

The aim of this study is to predict the future weather as monthly or weekly average. The GHCN only provides daily data, therefore, daily data will first have to converted into weekly or monthly data. The mean of every 7 days is defined as the weekly data and the mean of every 30 days is the monthly data. This is called a running average. This only works under the following two assumptions:

1. There are little to no holes in the data. As stated before, some stations don’t collect data for all the variables (temperature, precipitation, snowfall etc…), and some stations have missing data. If the data for a date is missing, then it is called a hole. A running average will only reflect the true average if the size of the holes are small.

For example, if the data for 8/10/2010 to 14/10/201 is available, but the next available data is on 19/10/2010, then there is a hole of 5 days, and the running average will not accurately reflect the real average (average of 8/10/2010 to 15/10/2010).

1. That the data points are close to each other. A mean works best when range of the data points is small. The more anomalies there are the worse the mean is. And this reflects in the running average as well.

Another factor that needs to considered is the window size, in other words, how much history to consider when making the predictions. A compromise needs to be made between a window size that is too small (in which case the algorithm won’t be able to ‘see the big picture’) and a window size that is too big (where old, irrelevant data might affect the algorithms judgement). For example, it wouldn’t make much sense to consider data from 1975 to predict the weather in 2014.

Equal number of daily data points is being divided by 7 to obtain the weekly averages and divided by 30 to obtain the monthly averages, meaning there is more weekly data then monthly data. As there is more testing data for the weekly average, more data points will naturally be predicted for the weekly data. However, the period of forecast is going to remain the same for both (as both 52 weeks and 12 months approximates equal to 1 year).

# Findings and Results – Part 1:

The predicted maximum weekly temperatures below are obtained with a window size of a quarter of a year.

Graphical user interface

Description automatically generated

*Figure (1): Shows the predicted maximum temperature (in red, predicted one year in advance) compared to the real maximum temperature (in blue). Both the predicted and the real temperatures are given as a weekly average. The x-axis is the number of weeks, which shows the timeline (0 weeks on the far left is the earliest available data on GHCN). Also has a point of interest circled in green and labelled as A.*

A

The precision of the predicted weather is astonishing. While the numerical predicted temperature might be different than the real temperature, the shape the predicted temperature makes mirrors that of the real temperature quite remarkably. Notice point A, the model accurately predicts that the maximum temperature is going to oscillate several times before it begins its steep increase.

Chart, line chart

Description automatically generated

*Figure (2): Shows how the train (blue) and testing (orange) loss changes as number of epochs increases for weekly maximum temperature predictions with a window size of a quarter of a year.*

From figure (2), the training loss steadily decreases as epoch increase. This is within expectation (as the model is getting more confident in its predictions). The sharpest decrease in loss is within the first few epochs, also within expectation (as there are more inaccuracies to fix at the beginning). The test loss has been relatively steady (or slightly increasing) as the number of epochs increase. This is due to overfitting. The model is becoming more confident in the characteristics of the training data which often doesn’t carry over to the testing data.

The mean squared error (MSE) is 90 (to the nearest integer). While on its own it doesn’t mean much, when considering that there are 163 predicted data points, this is a very low value. With an average of 52 weeks per year, 146 data points means that this model has predicted the maximum temperature one year in advance for over 3 years. The figure below, shows the predicted monthly temperature, again with a window size of a quarter of a year.

Chart

Description automatically generated with medium confidence

*Figure (3): Shows the predicted maximum temperature (in red, predicted one year in advance) compared to the real maximum temperature (in blue). Both the predicted and the real temperatures are given as a monthly average. The x-axis is the number of months, which shows the timeline (0 months on the far left is the earliest available data on GHCN).*

Without knowing the MSE, it can be judged by eye that the predictions made as a monthly average is much worse than the weekly average. The more the red line overlaps with the blue line, the closer the prediction is to the real value. Clearly the red line in figure (1) overlaps with the blue line much more than the red line in figure (3). It can also be clearly seen that the number of data points predicted as a monthly average is much less than the weekly average (this is because there are fewer testing data points, discussed in more detail in page 5). The MSE, train and test loss is shown in the figure below:

Chart, line chart

Description automatically generated

*Figure (4): Shows how the train (blue) and testing (orange) loss changes as number of epochs increases for monthly maximum temperature predictions with a window size of a quarter of a year.*

Like figure (2), the train loss is gradually decreasing, and the test loss is keeping a steady level (very slight increase as the epochs increase). The MSE is 173 (to the nearest integer), almost twice than that of the weekly predictions. Therefore, the maximum temperature predictions made as a monthly average is much worse than when made with the weekly averages. This relates to one of the assumptions (the data points are close to each other) breaking down. The monthly average is the mean of 30 days. In 30 days, the temperature can change by a large margin and the range of values is bigger, hence the mean is less likely to reflect the actual temperature (or temperature change) during this month. The same process is repeated for the minimum temperature.

Graphical user interface

Description automatically generatedChart, line chart

Description automatically generated

*[5a left]: Shows how the train (blue) and testing (orange) loss changes as number of epochs increases for weekly minimum temperature predictions with a window size of a quarter of a year.*

*[5b right]: Shows the predicted minimum temperature (in red, predicted one year in advance) compared to the real minimum temperature (in blue). Both the predicted and the real temperatures are given as a weekly average. The x-axis is the number of weeks, which shows the timeline (0 weeks on the far left is the earliest available data on GHCN).*

Chart

Description automatically generated with low confidenceChart, line chart

Description automatically generated

*[6a left]: Shows how the train (blue) and testing (orange) loss changes as number of epochs increases for monthly minimum temperature predictions with a window size of a quarter of a year.*

*[6b right]: Shows the predicted minimum temperature (in red, predicted one year in advance) compared to the real minimum temperature (in blue). Both the predicted and the real temperatures are given as a monthly average. The x-axis is the number of months, which shows the timeline (0 months on the far left is the earliest available data on GHCN).*

The figures above reflect the conclusions previously drawn. Like in the maximum temperature predictions, the weekly average predictions in minimum temperature is more accurate than the monthly average (which has an MSE of almost 3 times greater than the weekly one). The training loss is steadily decreasing, and the testing loss is kept relatively steady (slight overall increase). This further confirms that the predictions made as a weekly average are more accurate than monthly average. Having determined which of the two gives a better prediction, the next part of the study is to determine what lookback period gives the lowest mean squared error.

Chart, line chart

Description automatically generated

*Figure (7): Shows MSE (y-axis) as a function of window size (x-axis), note that that window size is given as number of years. Also, it is assumed that 52 weeks and 12 months both equal to 1 year.*

*Maximum temperature weekly (dark blue), minimum temperature weekly (light blue), maximum temperature monthly (red), maximum temperature monthly (orange).*

It appears that a window size of a quarter of a year gives the lowest weekly MSE and a window size of half a year gives the lowest monthly MSE for both maximum temperature and minimum temperature. The above calculations can be done for other stations. See Jupiter Notebook [1].

# Findings and Results – Part 2

The second aim of this experiment is to determine whether the model can predict the weather better than assuming the weather tomorrow is going to be the same as the weather today. This can be approached in 2 ways:

1. One is to assume that today’s data is always the real data and that it updates at every data point. For example, assume that the maximum temperature today is 25 °C, then the predicted maximum temperature tomorrow is also 25°C. However, it happens that the real maximum temperature tomorrow is 28°C. Then to make the prediction for the next day, ‘todays’ temperature becomes 28°C (and not 25°C).
2. Assume that today’s data always stays constant. For example, if the maximum temperature today is 25°C, then the maximum temperature in all future predictions is going to be 25°C regardless of what the real maximum temperature is.

Chart, line chart

Description automatically generatedThe predicted data is going to be output as daily data. The mean squared error for the predicted maximum temperature is shown in the figure below. The window size is 1 year (365 days).

*Figure (8): Shows how the train (blue) and testing (orange) loss changes as number of epochs increases for daily maximum temperature predictions with a window size of one a year.*

The MSE for the predicted daily temperature is much lower than the weekly one (the weekly one is almost 2 times that of the daily one). This shows that forecasting with daily data gives much more accurate predictions. The accuracy of the daily predictions is further strengthened by figure (9) below which shows the red line overlapping the blue line.

Graphical user interface

Description automatically generated

*Figure (9): Shows the predicted maximum temperature (in red, predicted one year in advance) compared to the real maximum temperature (in blue). Both the predicted and the real temperatures are daily data points. The x-axis is the number of days, which shows the timeline (0 days on the far left is the earliest available data on GHCN).*

Graphical user interface

Description automatically generatedIn the first case, where it is assumed that the real temperature updates as the timeline progresses, it can be expected that the MSE is going to be very low. This is because the maximum temperature is not going to vary significantly within such a short period of time. For example, if the maximum temperature today is 25 °C, then you can not expect the maximum temperature tomorrow to be -5°C. It is going to be within reasonable distance 25°C. This is reflected in figure 10, which shows that the red line almost completely overlaps the blue line.

*Figure (10): Shows the previous days maximum temperature (red) and the real maximum temperature (blue). Both the predicted and the real temperatures are daily data points. The x-axis is the number of days, which shows the timeline (0 days on the far left is the earliest available data on GHCN).*

Graphical user interface

Description automatically generated with low confidenceThe MSE for case 1 is 11 (to the nearest integer). However, the aim of this study is to create a machine learning model that could predict the weather one year in advance. Being able to predict the weather tomorrow knowing the weather today is not very useful. Therefore case 2 (where todays temperature stays constant and doesn’t update as the timeline progresses) is more sensible. We can, however, expect the MSE to be very high in this case (case 2), as the temperature is not going to remain the same over long periods of time. The figure 11 below further reinforces that.

*Figure (11): Shows the predicted maximum temperature (red) and the real maximum temperature (blue). Both the predicted and the real temperatures are daily data points. The x-axis is the number of days, which shows the timeline (0 days on the far left is the earliest available data on GHCN).*

As can be seen from figure (11), the predicted temperature in this case not close to the real temperature, and as expected the MSE is very high, with a value of 238 (to the nearest integer). The same calculation can be done for minimum temperature and similar figures are obtained. See Jupiter Notebook [1].

# Conclusion:

In conclusion, the primary aim of this study has been met. A machine learning model that can predict the weather one year in advance has been created. This model can predict the maximum and minimum temperature both as a weekly average, monthly average, or daily data. It is obvious that when forecasting with daily data, the predictions are the most accurate. Followed closely by forecasting with weekly averages.

This study has not come to a definite conclusion as to what window size would provide the most accurate predictions. This is due to the limited amount of data. Window sizes greater than 1 year are hard to achieve as there are not enough data points for the monthly average. From the limited amount of data available, it appears that a window size of a quarter of a year (13 weeks) is best for the weekly average and a window size of half a year (6 months) is best for the monthly averages.

To improve this section of the study, analysis could me made on larger data sets and different variables (such as snowfall and snow depth) and different stations. Larger data sets would mean that testing a larger range of window sizes is possible, it would also mean that there is more data to train and test on. Testing on different stations would strengthen the conclusions drawn from this study. Testing on different variables would ensure that this model doesn’t just work on temperatures but on other factors too.

The second aim of this study would not be met under the assumption that the real temperature updates as the timeline progresses. However, as discussed above, this is akin to predicting only one day in advance, whereas the model is attempting to predict one year in advance. Therefore, a comparison between the 2 is not appropriate. If operating under the assumption that the real temperature stays constant as the timeline progresses, then the model can predict the future weather better.

This section of the experiment could be improved by creating model that can predict the weather one day in advance based on only the weather on the previous day. This can be achieved by reducing the window size to 1 and the timesteps (how far ahead to predict) to 1.

# References:

1. Maruf Talukdar. 2022. *Mini\_Project\_2 (2).* Jupiter Notebook
2. Team, K., n.d. *Keras documentation: About Keras*. [online] Keras.io. Available at: <<https://keras.io/about/>> [Accessed 17 January 2022].
3. TensorFlow. n.d. [online] Available at: < <https://www.tensorflow.org/> > [Accessed 17 January 2022].
4. Team, K., n.d. *Keras documentation: The Sequential model*. [online] Keras.io. Available at: <<https://keras.io/guides/sequential_model/> > [Accessed 17 January 2022].
5. Jayawardhana, S., 2020. *Sequence Models & Recurrent Neural Networks (RNNs)*. [online] Medium. Available at: < <https://towardsdatascience.com/sequence-models-and-recurrent-neural-networks-rnns-62cadeb4f1e1> > [Accessed 17 January 2022].
6. Deval, S., 2020. *Mean Squared Error - Explained | What is Mean Square Error?*. [online] GreatLearning Blog. Available at: < <https://www.mygreatlearning.com/blog/mean-square-error-explained/> > [Accessed 17 January 2022].
7. National Centers for Environmental Information (NCEI). n.d. *Global Historical Climatology Network daily (GHCNd)*. [online] Available at: < <https://www.ncei.noaa.gov/products/land-based-station/global-historical-climatology-network-daily> > [Accessed 17 January 2022].
8. PubMed. 1985. *China's one-child family: a famine-forced decision*. [online] Available at: <<https://pubmed.ncbi.nlm.nih.gov/12340334/> > [Accessed 17 January 2022].