# Predicting future warming and climate in the Arctic circle

**Mujahed Darwaza**

## Introduction

As our concentration of carbon dioxide in the atmosphere has reached 420ppm and keeps on increasing, our planet is going to keep on warming and the effects of climate change have become clearer than ever. The average temperature has risen by more than 1 degree in the past century, and this warming is yet to slow down. One region that has been particularly affected by this warming is the Arctic circle. Because of climatic conditions in that part of the world, the Arctic has observed much bigger temperatures rises, the increase reaching up to 14 degrees during winter.

To understand how these increases are affecting the world, and model future climates, climate scientists rely mostly on Global Climate Models (GCM). These very complex models simulate the planet's climate by taking into account the atmosphere, the oceans, the sun and more. They are becoming ever more accurate, but one flaw that has been recurrent in this field is their computational cost. An emerging solution would be to use modern Machine Learning techniques to understand and predict future climates instead of relying on physical models. Among the current research, a paper by Mansfield et al stands out as they strive to predict global climate patterns by using Ridge Regression and Gaussian Processes. Inspired by this paper, I have decided to try implement other machine learning algorithms to predict future climate variables in the Arctic region.

## Context and variables

The increase of greenhouse gases in the atmosphere has caused a serious warming effect in the Arctic. In effect, this region used to benefit from a very high reflectivity of sunlight thanks to the ice it contains (climate scientists would talk of high albedo). But as the warming intensifies, more of this ice sheet has been melting. This caused a reduction in the albedo of the region, which in turn allowed for an even stronger warming. This positive feedback loop is the main reason the Arctic has been warming up much faster than other parts of the world.

In this study, I will try to take this effect into account. A first important variable that I will study will be CO2 levels. To predict future warming of the planet, it can be useful to start understanding how current emissions account for today's warming. The second and third most important variable will be temperature and albedo. Temperature and albedo are the key ingredients in the feedback loop described previously, and thus must be accounted for in this model. As most of the effect happens on land, this study will only consider these variables for land, neglecting the effects of oceans on the system. Finally, the warming results from the interaction of incoming radiation with the atmosphere, and of the amount that is lost to the system in various ways such as evaporation of water. These variables will also be added to the system as they might have a clear way of explaining the temperature rise. Since temperature, net radiation and evaporation are seasonal, the variables are expected to vary a lot over the course of a year.

## Dataset used and pre-processing

The dataset that was used to conduct most of our study is extracted from the Copernicus Climate Data Store. Among the various datasets present in their database, I have used the ERA5 land average hourly data from 1950 to today. This dataset contained all the variables needed for my project except for CO2 levels, which I was able to extract from the Mauna Loa database. It contained all daily CO2 levels from 1973 to today.

Both datasets need a pre-processing phase as they were unusable in their first version. The CO2 levels were not very accurate, and a very big amount of the points had an absent value (represented by a value of -999.99). This was problematic as it would not work in the learning algorithms and would probably create an additional bias in the learning phase.

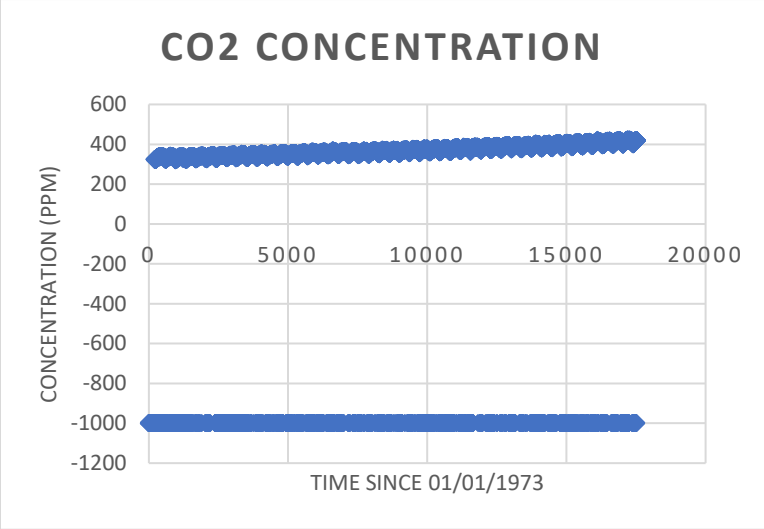



*Fig 1. $CO_2$ concentration since 1973 (Mauna Loa)*

To counter this problem, a solution I have found was to replace the absent values by the smallest value that occurred in the next 100 points. In effect, the concentration is increasing so it seemed relevant for the purpose of this study to do such an approximation. The new dataset once processed looked as follows :

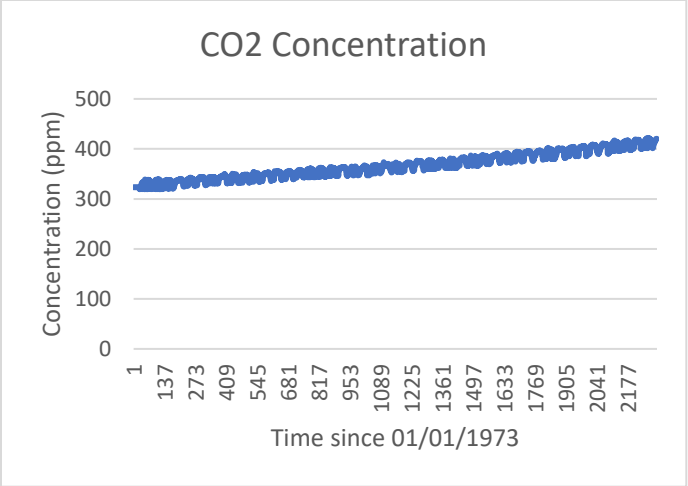



*Fig 2. $CO_2$ concentration since 1973 after processing*

As for the climate variables, the initial dataset was too heavy for the purpose of this simulations: it was over 50GB of data when considering variables for every day since 1973. This amount of information would not have been useful for the learning and would have probably lengthen the algorithm time completion rendering it useless. Instead, I have selected four time occurrences per month : noon and midnight, 1st and 15th of every month. The number of points was thus divided by a factor of 5. The same processing has been made eventually for the $CO_2$ levels to have matching datasets. This was an improvement, but the data was still very heavy, as it turned out that the maps were much more highly defined than necessary. The second processing phase was then to redefine the resolution of the map, reducing it to 0.5 degrees in longitude and latitude instead of 0.1 degrees. Here is an example of the final map of temperature when compared from 1973 to 2021 on January 1st :

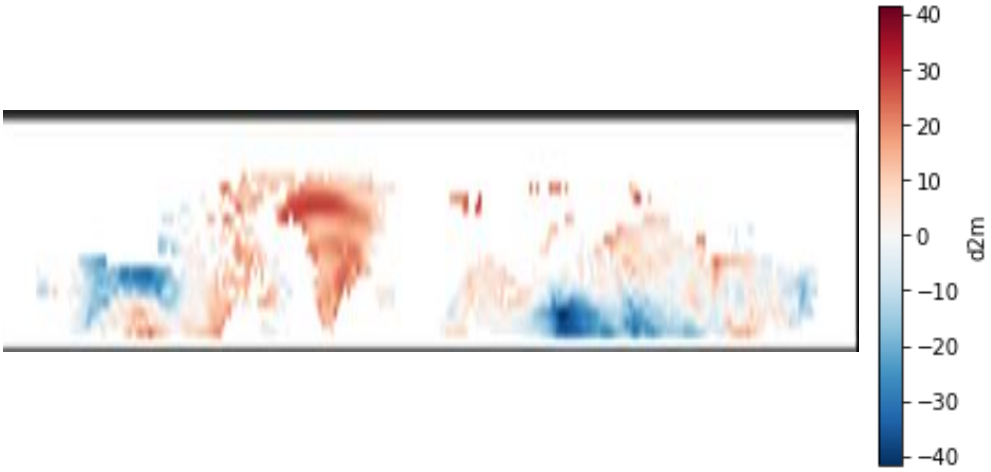



*Fig.3 Temperature increase in the Arctic from 01/01/1973 to 01/01/2021*

## Machine Learning algorithms

Finding the right machine learning algorithm for this study was not easy. The dataset I had contained both time and space variables and was not always of the same type (maps for the climate variables, and simple vectors for the $CO_2$ concentration. My initial ambition was to reproduce the results of the Mansfield et al paper to have in output of my algorithm a map of temperatures in the Arctic 10 years from now. This turned out to be a very complex task that required either using multiple different forms of networks merged into one (LSTM, CNN, and GAN), or to implement a Gaussian Process such as the one in the paper. After multiple trial and error, I finally settled on reassessing my ambitions.

Instead of benefiting from the space-dimension of my data, I have reduced it to a single value for each year by taking the mean on all the available space. This was a major reduction of the available information but allowed for the use of algorithms that were more relevant for having only the mean temperature of the region as an output.

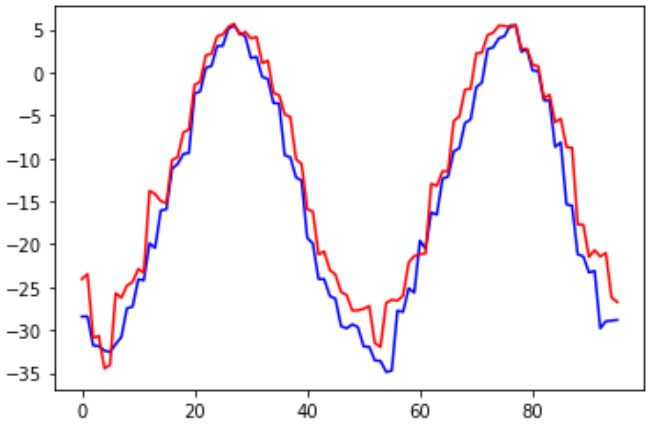



*Fig.4 Temperature over the course of two years in 1973 (blue) and 2020 (red)*

I started by implementing three algorithms for the $CO_2$ dataset that would enable us to create different possible future scenarios of $CO_2$ emissions. The first algorithm was a neural network that took in input the $CO_2$ concentration at time t and tried to predict the $CO_2$ concentration at time t+1 : $CO_2(t) \rightarrow\therefore\rightarrow CO_2(t+1)$

The second algorithm that was implemented was also a neural network. This time, instead of using the concentration at time t, it could use multiple concentrations from the past to predict the future value : $CO_2(t-length), \cdots, CO_2(t) \rightarrow\therefore\rightarrow CO_2(t+1)$

These algorithms were not enough as they did not take into advantage the time-series format of the data. To improve the algorithms, I have implemented a Long-Short-Term-Memory (LSTM) network for the $CO_2$ concentration. This type of network is a form of recurrent networks : it uses previous predictions as inputs for the next prediction. Its great strength compared to other RNNs is that it does not have a vanishing gradient or exploding gradient effect. This is because of the memory structure of the network that keeps from letting gradients impact one another too much.

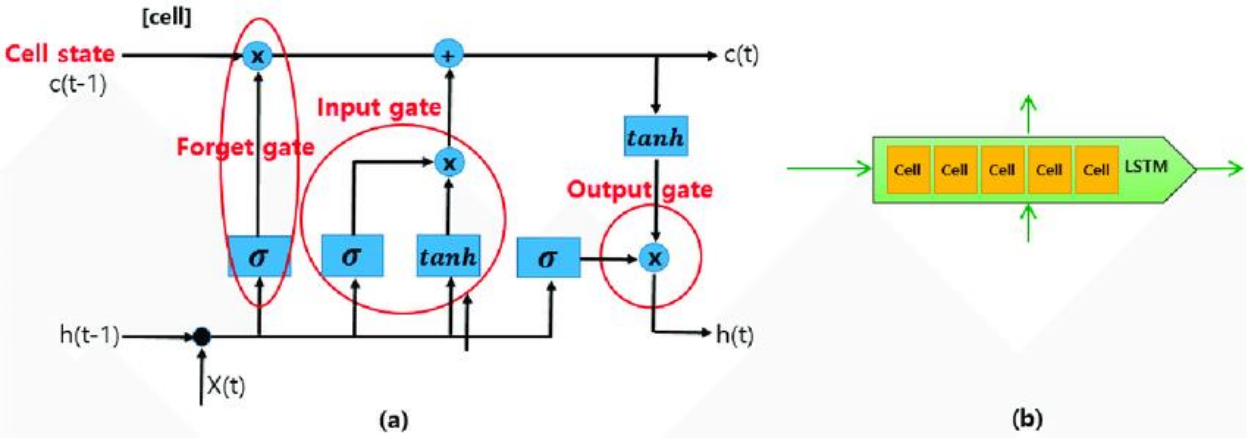



*Fig.5 Structure of a LSTM network (https://www.researchgate.net/)*

The same algorithms were then readapted to take into account the climate variables I had. This time, the algorithm would try to predict future temperature, albedo, net solar radiation, and evaporation by using $CO_2$ levels as well as past versions of those said variables.

For all the algorithms, I have implemented the early stopping method to avoid overfitting. Some algorithms also use dropout, but this technique usually sacrificed performance and was never useful for the study.

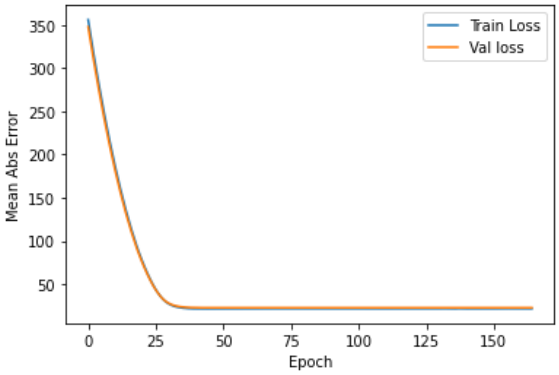
## Learning and efficiency

The algorithms used had varying degrees of accuracy and did not use the same number of parameters to predict $CO_2$ levels and the climate variables.

### Single-date Neural Networks

This algorithm had the overall worse performance, as we would expect. Since it only took into account the previous data to predict the next one, it usually was unable of assessing the trend of the variables I was using and even at its best was still very poor in its predictions.

For the $CO_2$ predictions, it was clear that the algorithm had to use a Rectified Linear Unit (ReLu) activation function. When it was using a Hyperbolic Tangent (tanh) or Sigmoid function, the mean absolute error was absurdly high (21.37), and the algorithm would converge to a value that had no sense with regards to the data.



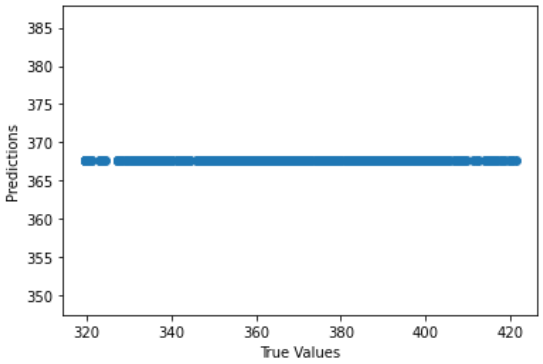
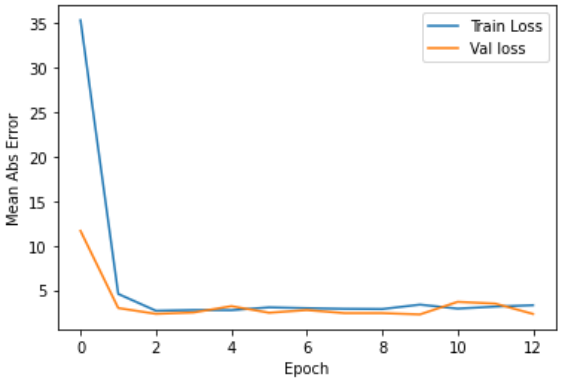




*Fig. 6 & 7 Single date Neural Network performance for $CO_2$ predictions with tanh activation*

The best accuracy was achieved for a network with two layers containing 5 neurons and the ReLu activation function. (mae = 2.3)



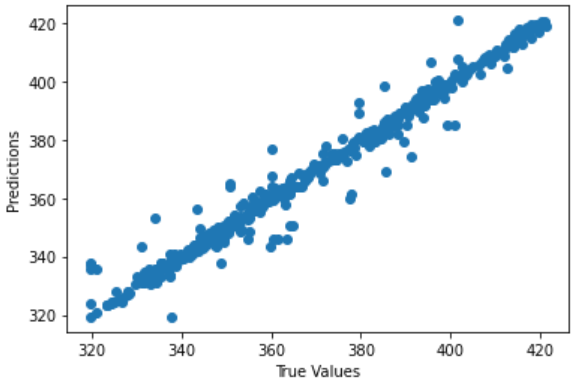




*Fig. 8 & 9 Single date Neural Network performance for $CO_2$ predictions with ReLu activation*

As for the climate variables, trying out different activation functions showed that tanh and ReLu were equivalent for this kind of network. The best mean absolute error was obtained for a 25 neuron, two layered network, and was equal to 0.13.

**Multiple dates Neural Network**

Putting as input multiple dates at the same time had different effects whether looking at the $CO_2$ predictions or the climate predictions. I would have expected to see an overall

improvement, but this was strangely not the case for the $CO_2$. It appears it had already reached the best possible scenario with a single date for that variable, which is unusual.

In the case of the climate predictions, we can see a net improvement when implementing the multiple input neural network. By studying different sizes of inputs, it seems that using 8 previous dates is the ideal. Sigmoid and ReLu activation functions give similar results, which are overall better than tanh. By trying different network sizes, it seems that the best results are obtained for a two-layer, 75 neurons per layer network. This gives us a mean absolute error of 0.064. This is 50% of the previous error which is promising.

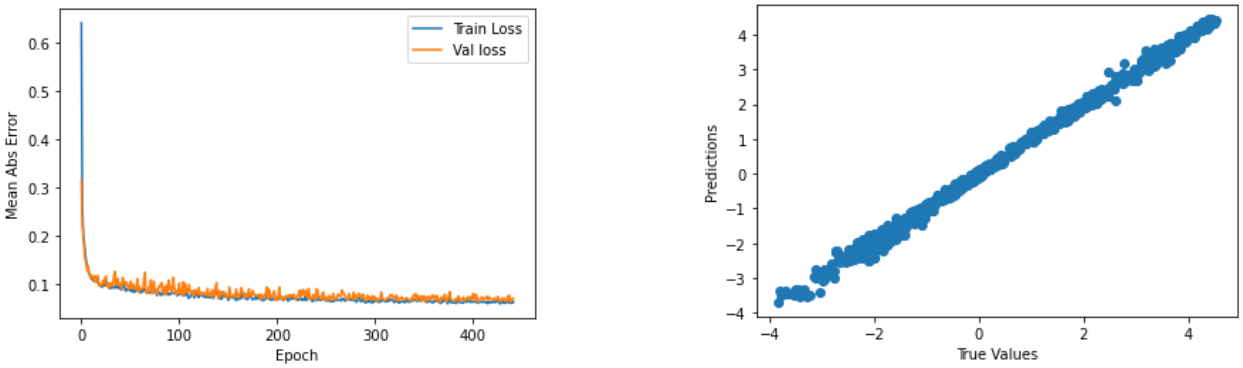



*Fig. 10 & 11 Best multiple dates Neural Network performance for climate predictions*

**LSTM models**

Finally, the implementation of the LSTM was only useful for the climate predictions. LSTM networks work better in batches that are not too large, so in my algorithm I implemented batches of size varying from 4 to 96. Th best mean absolute error was 0.028 and it was obtained when training the network with only one LSTM layer of size 150 followed by a dense network with 120 neurons , the tanh activation function, and batches of size 48. We can see that once again this is a much better performance than the previous one.

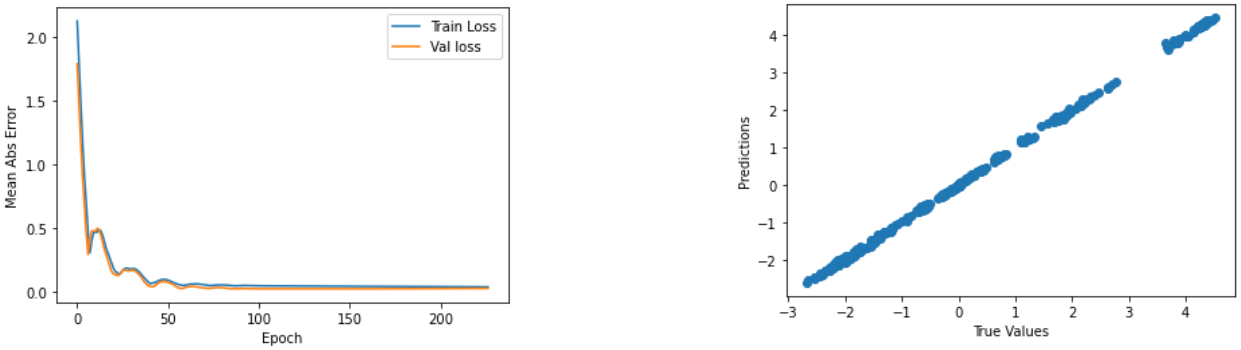



*Fig.12 & 13 Best LSTM network performance for climate predictions*

## $CO_2$ predictions + scenarios

Once the network has been trained, I decided to simulate some of the potential scenarios of future $CO_2$ levels over the next 10 years. The idea was to have different forms of future emissions, to later predict how the Arctic region would react to these emissions. I decided to use the single date neural network as it gave the best performance in my previous analysis. The initial result did not take into account the periodicity of the variables, so I had to add a sinusoidal white noise to do so. To give different scenarios, I kept the prediction as outputted by the network, and then added an exponentially decreasing factor to one of them, and an exponentially increasing factor to the last. Here is the obtained result :

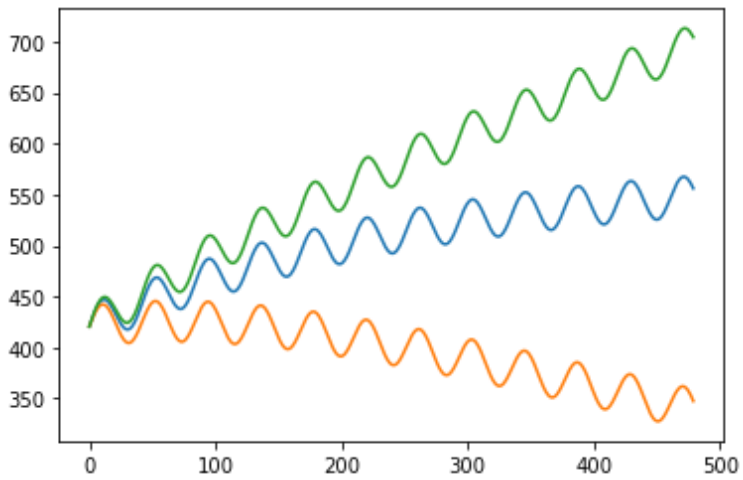



*Fig. 13 $CO_2$ concentration (ppm) scenarios using Neural Networks*

As we can see, the “normal” prediction has an expected behavior. It keeps on increasing through time at a rate that is a little high (+100ppm concentration in 10 years). The exponentially increasing one goes very high very fast. As for the exponentially decreasing one, it increases a little then starts to decrease slowly. All rates of change are too high for what we would expect for the climate system to react to in a decade, but this gives us an initial way to predict future climate.

As a test, I tried predicting the $CO_2$ levels using the LSTM model with a high batch size (thus having multiple time occurrences ). This time I did not add any white noise but kept both exponentials with a same factor. The result was more satisfactory than the neural network, especially as the batch size increased. This shows that although the LSTM model has a higher mean absolute error, it can model the change of carbon dioxide as good as the neural network.

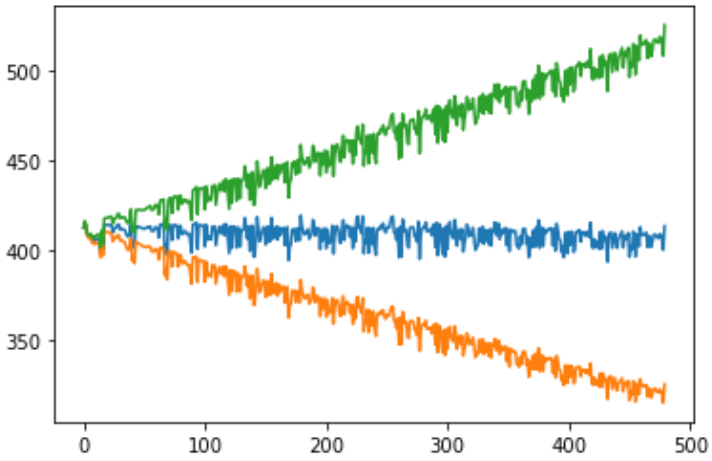



*Fig. 14 $CO_2$ concentration (ppm) scenarios using LSTM*

## Climate predictions

With these different scenarios, I later tried to predict temperature, albedo, net solar radiation, and evaporation over the next 10 years. Since the LSTM models had the lowest error, they were the ones I implemented to predict these variables. The batch size had a very strong effect on the quality of the prediction, the smaller they were, the more unpredictable and fluctuating the predictions were.

At first, I used the results of the neural network to predict the variables. The result was promising and didn't seem absurd. For each scenario, the temperature behaved in the way we would expect it : increasing if $CO_2$ levels were increasing and decreasing otherwise. However, the range of data was completely offset. As seen in the previous temperature profile of the Arctic, the temperatures are supposed to vary between 5°C and -35°C. In the prediction, the temperature was always between 0°C and -3°C.

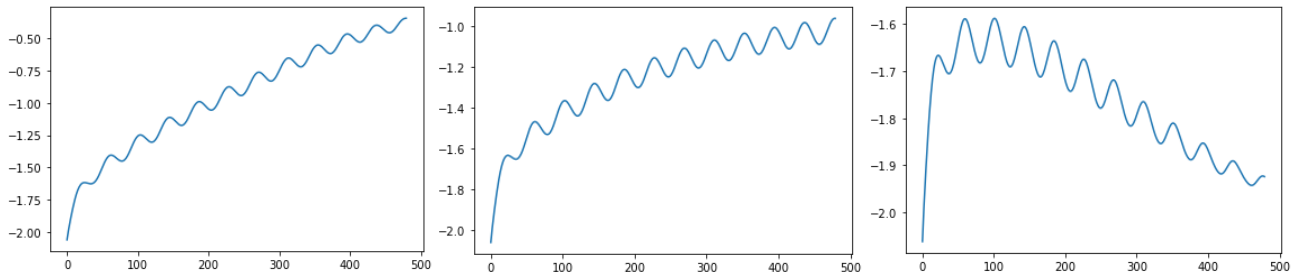



*Fig. 15, 16 & 17 Temperature increase by scenarios*
*(exponentially increasing/normal/exponentially decreasing)*

If we look at the other variables we can see that the results are problematic : the net solar radiation seems to increase when $CO_2$ decreases, and they are not varying as we would expect them to do seasonally.

Using the other scenarios that were outputted from the LSTM model does not give better results.

## Conclusion

The results obtained from the climate predictions are not conclusive. The models can recognize the sinusoidal shape that is present in the $CO_2$ levels, but it seems that this variable doesn't have a strong influence on the output of the model. This results in a simulation that is lacking and not capable of computing the feedback loop effect we have described in the beginning. In addition to this, the $CO_2$ levels vary too much for the period we are studying, which means that the same algorithms are not performing well and do not simulate the increase on a longer time scale as they should.

Many different reasons can explain the lack of physical meaning to these results. First, the size of the dataset might have been not adapted to this study, and the choice of dates seems to create more complexity than necessary. If we were to look at the temperature for a same given month, there would be less seasonal variability, and a more coherent increase of temperature with regards to the $CO_2$ increase. In addition to this, implementing another machine learning algorithm such as Gaussian Processes for the $CO_2$ levels might have given better results, as it has already been tested and gave promising results in the past.

As for the climate predictions, the algorithms are not giving enough importance to the $CO_2$ levels that are increasing and have lost the periodicity that they initially had over the course of a year. Using a Gaussian Process might have been more adapted to the study because of this, as the choice of kernel for these kind of networks can give them very good performance to simulating periodicity.