Machine Learning of Cloud Organisation

An industrial project with the University of Bath & the Met Office

Matthew Coward – Supervised by: Kwinten Van Weverberg, Chris Budd, Lisa Kreusser, Teo Deveney

INTRODUCTION

The aim of the project was to use machine learning to gain more insight into the organisation of clouds. Current weather forecast models divide the Earth into grid squares and predict weather for each entire square. However, they have a blind spot when it comes to clouds, because almost all cloud organisation occurs at a subgrid level. The entire grid square will just be labelled 100% cloud or no cloud. Small shallow, cumulus clouds over warm, dry land are particularly troublesome because of their tiny size but they also frustratingly have a large impact on climate. The current way to overcome this is to use a cloud fraction parameterization to specify what proportion of the grid box is cloudy.

While it would have been far beyond the scope of the project to predict the entire distribution of cloud, we used machine learning predict the surface area, also known as "cloud perimeter". This value is insightful in both cloud and radiation schemes. Using the cloud *fraction* (which we already model) combined with *perimeter* offers a good insight into organisation; a fraction of cloud close to 0.5 and a very large surface area might suggest shallow cumulus clouds, which have an almost checkerboard-like pattern.

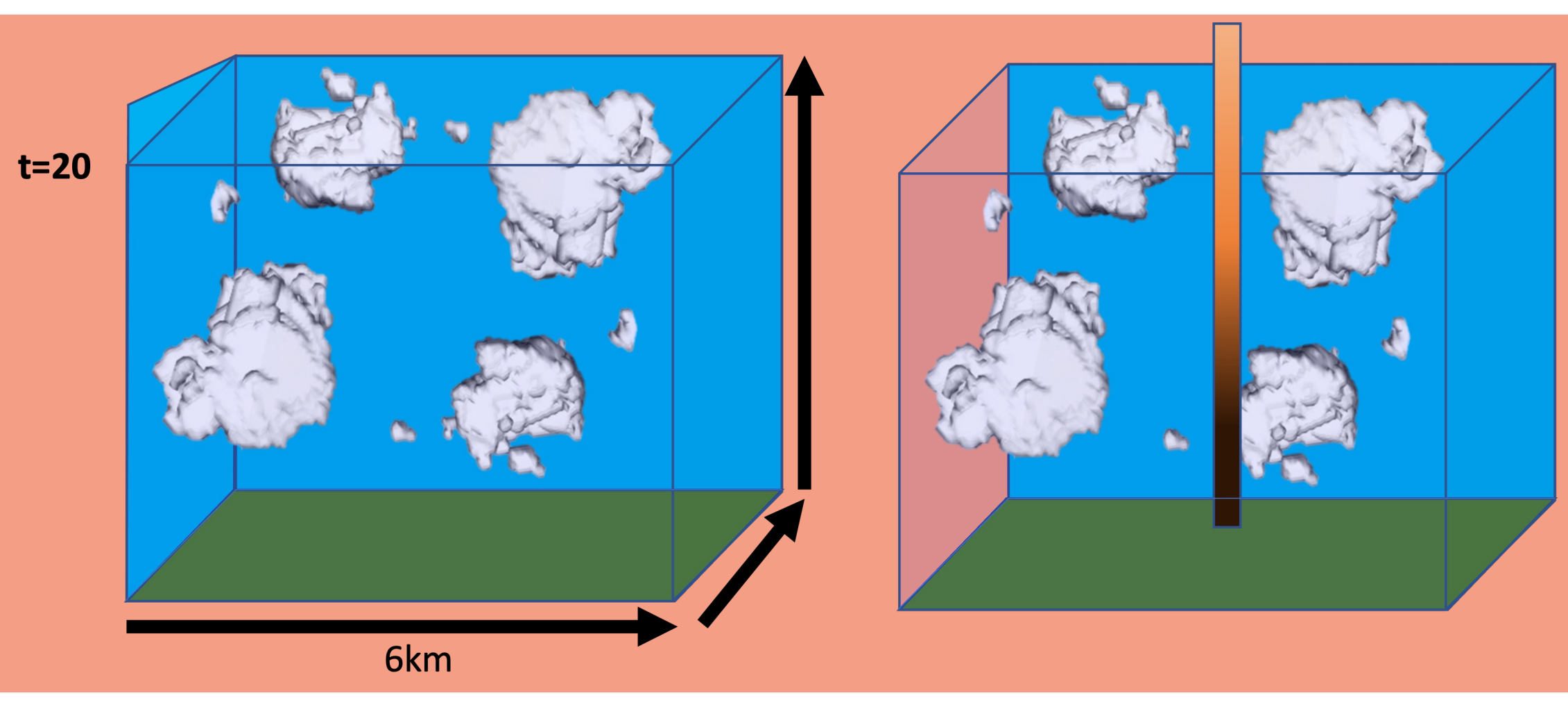
It is also a useful parameter to improve other parameterizations, such as the cloud-erosion parameterization, improvements in which in turn have been found to improve the overall cloud scheme. In addition to this, the SPeedy Algorithm for Radiative Transfer through Cloud Sides (or SPARTACUS) which can approximate 3D radiative transfer in climate models, requires an estimate of the cloud perimeter. **Hence our work could improve its performance and even improve the modelling of clouds' radiative effects.**

OBJECTIVE

Overall, a successful project would therefore be one that ultimately built a reliable machine learning model which could predict the cloud perimeter within a grid square, using information such as cloud fraction and various other atmospheric properties such as temperature and pressure.

DATA PROCESSING & METHODOLOGY

Our decision to use machine learning was mainly due to the recent availability of a dataset by the US Department of Energy. **Using cameras and stereophotogrammetry, the locations of clouds over time down to 50m accuracy on a 6km^3 site in Oklahoma have been collected. This is the Clouds Optically Gridded by Stereo (or COGS) data set.** Records of clouds were taken every 20 seconds for nearly 3 years, in a vast matrix containing 1s and 0s for "cloud" and "no cloud". This is an entirely unique insight into the lifecycle of clouds.

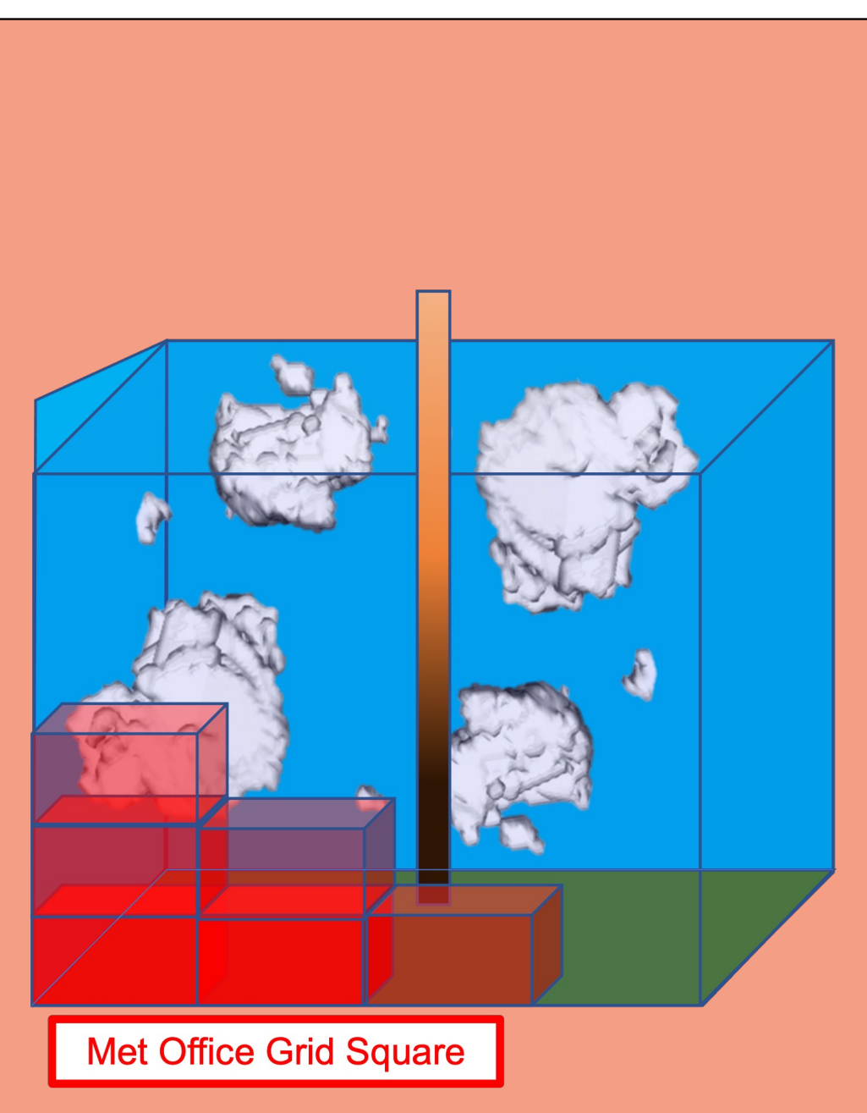


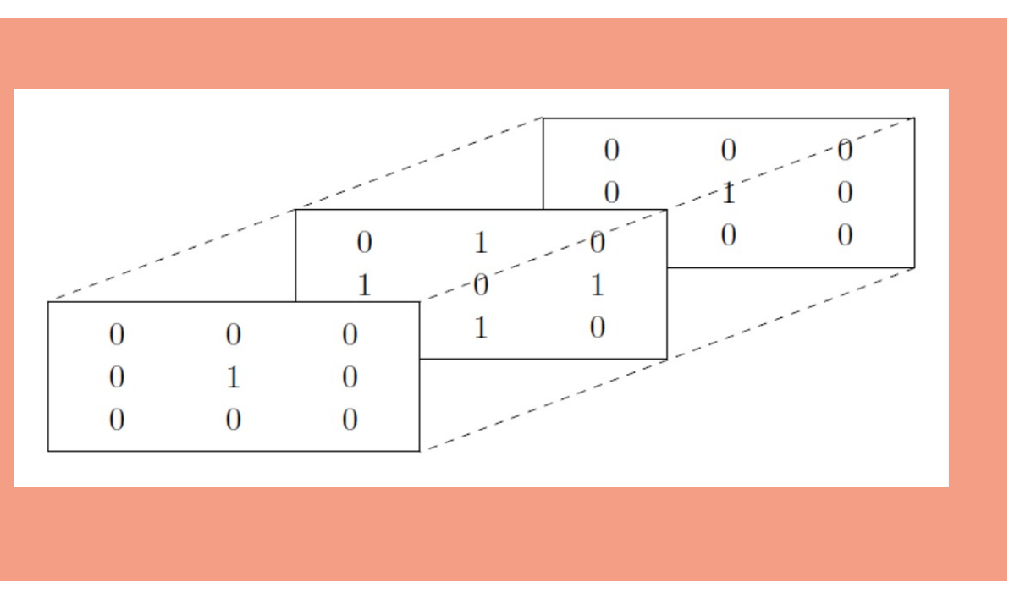
We also had data available on atmospheric conditions such as temperature and humidity. This data unfortunately only existed in a single vertical profile, located at the centre of the 6km-cubed domain. The data derived from a balloon sounding.

Image displaying the COGS domain (left) with atmospheric profile (right)

The first approach might be to compute the cloud fraction and perimeter on the entire 6km^3 region at each time step and use these values as the data with which to train the model.

However, it was decided that generating a single value for perimeter and fraction for the *entire* 6km-cubed region was not entirely sensible. Ideally, we wanted to implement this model within the Met Office's existing climate models, and the Met Office grid squares are considerably smaller than 6km^3 - 300m high by 1500m by 1500m horizontally. Therefore, a single Met Office grid square located above the surface at the boundary layer height will likely have a much higher proportion of cloud than a 6km^3 region which is may be mostly empty space. A model trained only seeing a 6km^3 region with barely any cloud will not perform well when asked to predict the cloud perimeter in a much smaller grid square, located precisely in an area filled with cloud. As a result, we decided to divide the 6km^3 region into smaller cells with the same dimensions as the Met Office grid squares. For each time step, our 6km^3 region contains 320 of these cells.

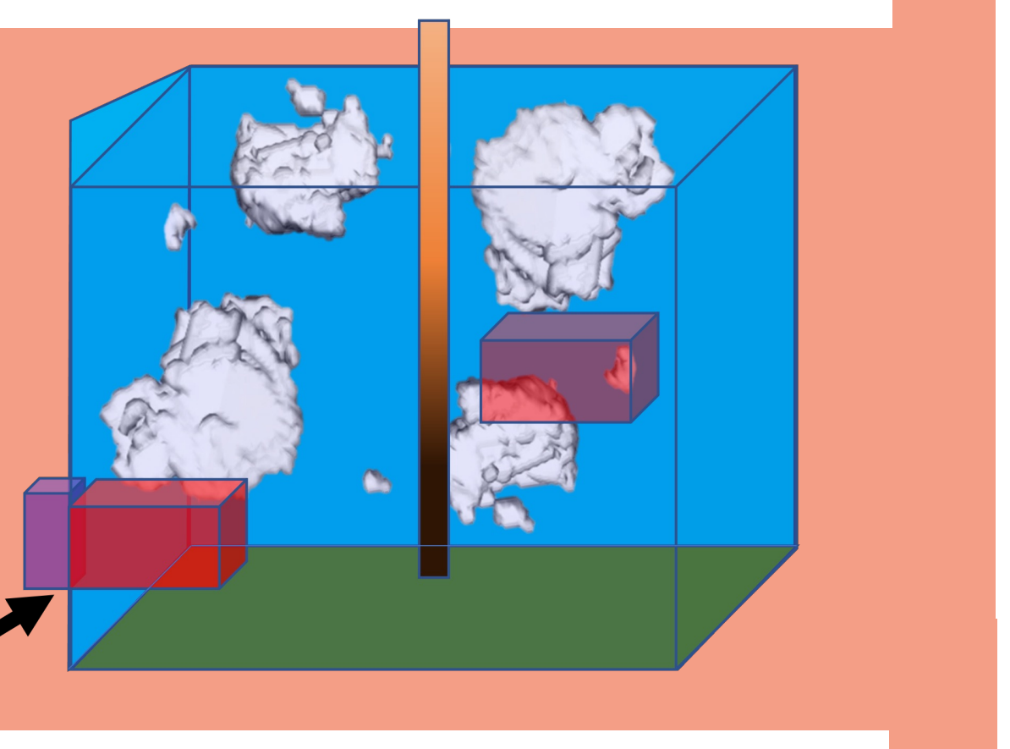




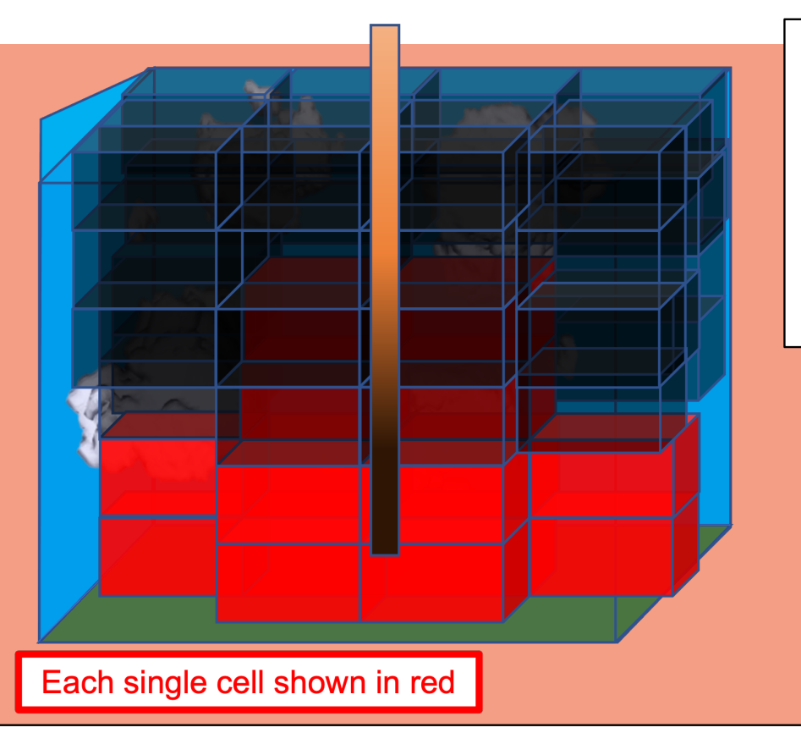
On each cell, the cloud fraction and cloud perimeter needed to be computed. The cloud fraction was straightforward – this is the number of 1s in the cell divided by the size of the cell. The cloud perimeter had to be computed by a convolutional kernel (left) over the domain. For each cloudy pixel, the kernel computed the number of neighbouring 0s storing these values in a new matrix. This matrix was then summed and we get the cloud perimeter over the entire cell.

It was thought likely that the height of the cell within the domain would be useful for the perimeter prediction. Hence, this was also used as an explanatory variable, encoded as an integer from 1 to 20 for where it occurred in the vertical stack of cells in the 6km^3 domain. The full atmospheric profile was used as a predictive variable for all cells because, for example, pressure lower down near the surface affects what clouds look like higher up. The single available atmospheric profile had to be collocated in time with the COGS time steps.

A key modelling decision was how to 'pad' the domain, in order to apply the kernel to the outside of the domain to compute the perimeter. We don't know whether a 1 on the outside lies next to a 0 or a 1. We decided to only run the kernel up to the penultimate slices of the 6km^3 domain so as not to add any artefacts to the data.



An image displaying problematic cells lying on the edge of the domain. The values in purple would need to be known in order to apply the kernel to elements on the outside of the domain.



Each single cell shown in red

A visualisation of the division of the domain into cells on which the cloud perimeter, cloud fraction and height were computed, along with a single vertical profile for each atmospheric condition, used as an explanatory variable for all cells

The cloud reconstructions required the triangulation of 6 cameras, thus the region of overlap was more like a pyramid rather than a perfect 6km cube. Thus, many of the 320 cells contained missing values that could not be reconstructed. We ignored any of the 320 cells which contain a missing value, since this still left 98 cells per time step (left). **For every COGS time step, the cell height, cloud fraction and cloud perimeter were computed for all 98 cells. Each cell represented one training point, with two scalar – cloud fraction and height – and one array of atmospheric conditions (the same used by all cells at a given time step) as explanatory variables. These were used to predict the response variable – the cloud perimeter.**

PRE-PROCESSING

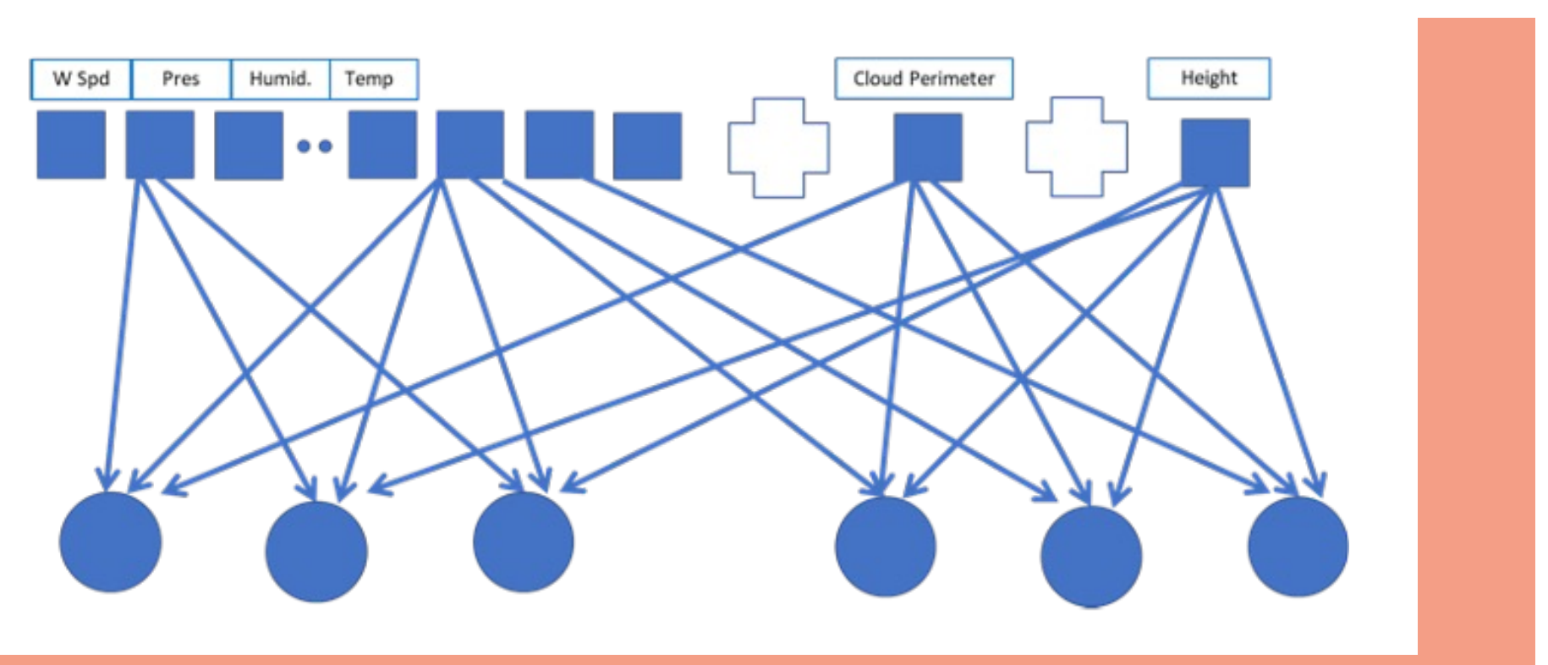
After fitting a neural network, the decision was made to pre-process the data prior to inputting into the model. A large portion of the data points, around 90%, had a cloud perimeter of 0. As a result, every model learnt to predict 0 perimeter for all data points. Since cloud fraction is provided as an explanatory variable, if a data point has 0 cloud fraction, then we want our model to predict 0 cloud perimeter. Almost like "bypass" layer built on top of the network, any incoming data gets checked to see whether its cloud fraction is 0. If so, a cloud perimeter of 0 is predicted and if not, it is inputted into the neural network.

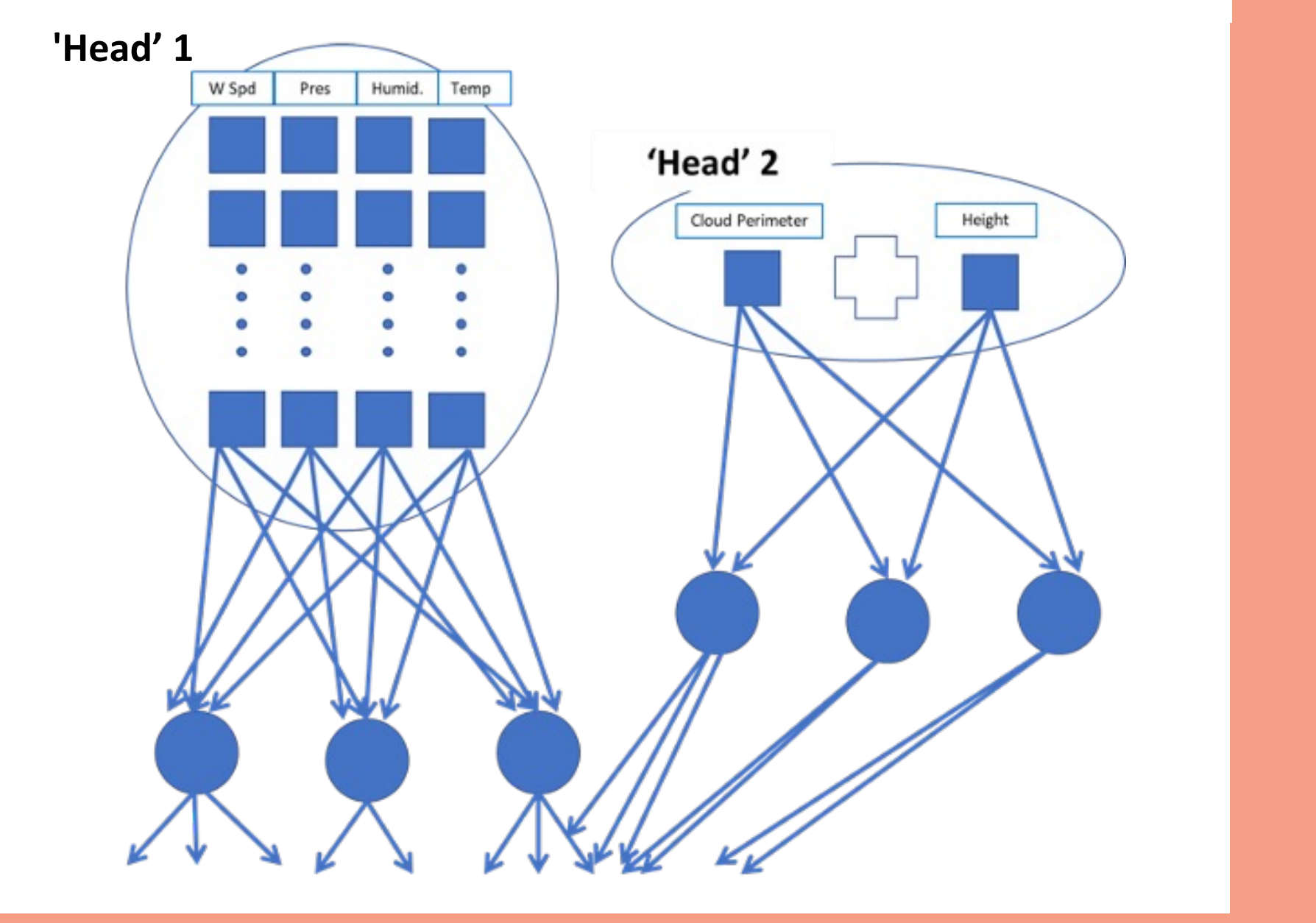
This was also found to partly solve some RAM issues, since reducing the size of the training set in this way allowed the neural network to be trained on the entire training set simultaneously, as it was now under 24GB.

MODEL FORMULATION

The first model devised was a simple neural network. It took a single vector as the input. In order to get a vector as the input, the vertical profiles of atmospheric conditions had to be stacked one on top of the other and then the cloud fraction and height appended onto the end.

It would be good if the network took the atmospheric profiles as an array so that it could see the vertical structure of the atmosphere without it being flattened into a vector like it was in the previous network. The problem is that we now have two floating scalars which cannot be inputted into a neural network with an array of data. **To solve this, a multi-input neural network or MNN was used**. This has two separate network inputs which are entirely independent. Both data types can be inputted simultaneously. It also means convolutional layers could be applied to the atmospheric data, to leverage the spatial aspect of the data. **Therefore, two models were trained – a simple neural network and the convolutional MNN.**



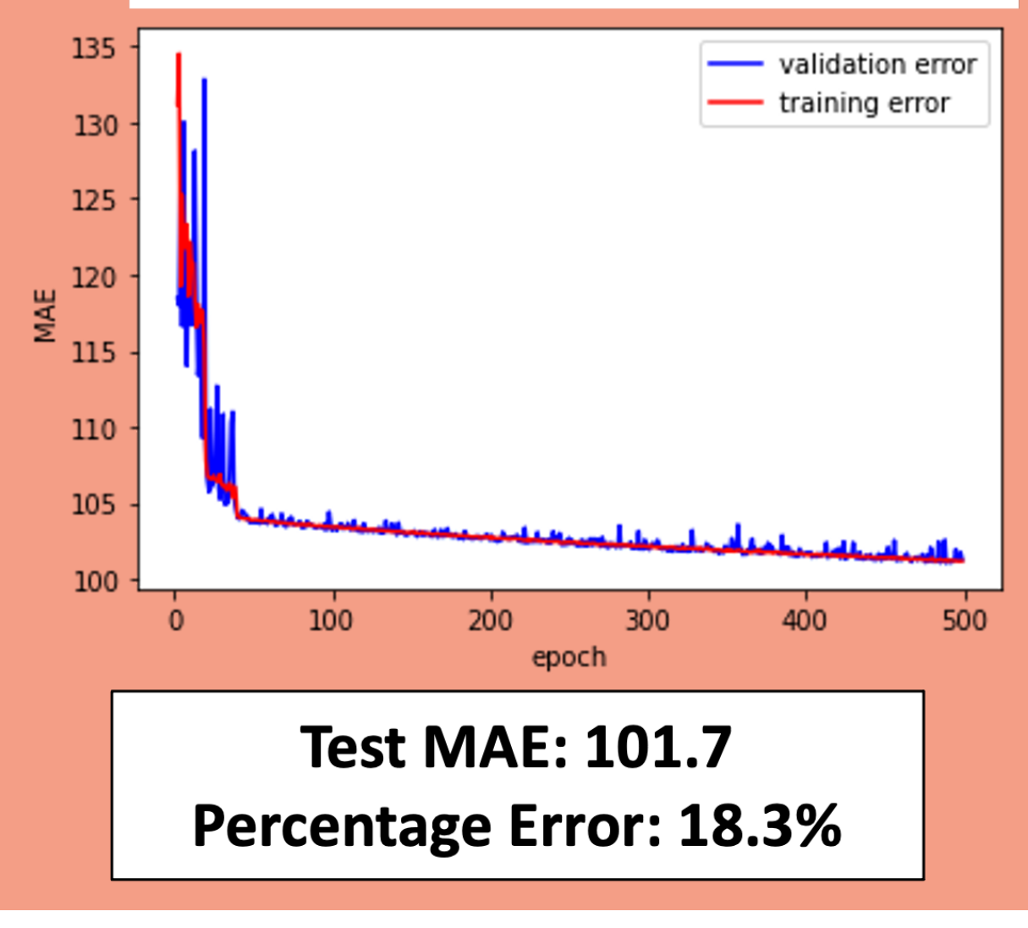


^ ANN Model   MNN Model >

RESULTS

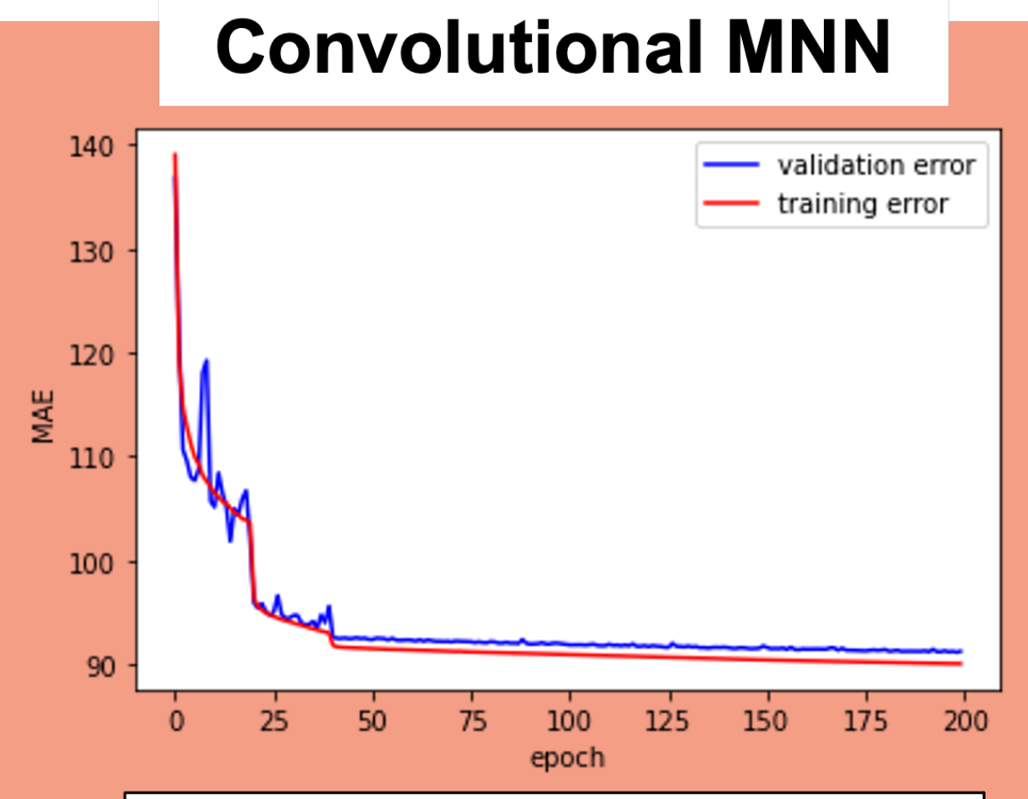
The accuracy on the test set was assessed via the mean absolute error - the average absolute difference between the true perimeter and the predicted perimeter over the entire test set. This was divided by the mean value of the perimeter on the test set to give a relative average percentage error in the prediction which was slightly more interpretable. **The simpler model managed to achieve a percentage error of 18.3%. The more complicated convolutional MNN managed to improve this accuracy to 16.1%.** The errors achieved by the two models, along with two plots containing the validation and training error of the model against the epoch number, are given below.

Simple Neural Network



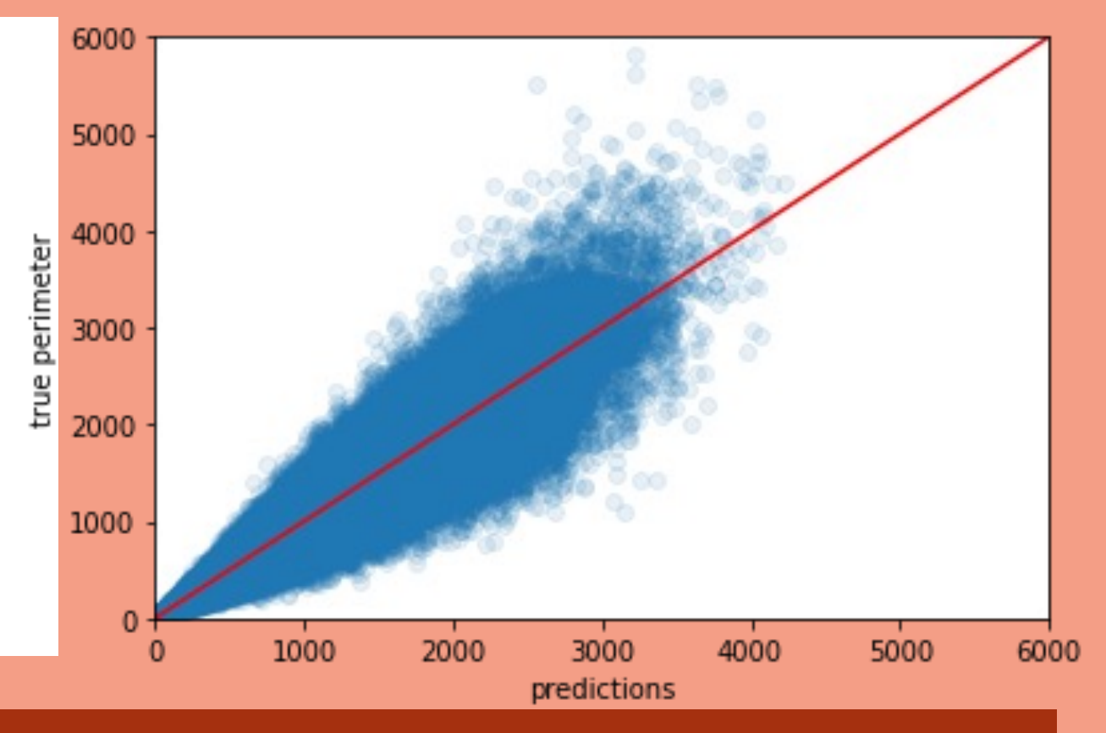
Test MAE: 101.7
Percentage Error: 18.3%

Convolutional MNN



Test MAE: 89.6
Percentage Error: 16.1%

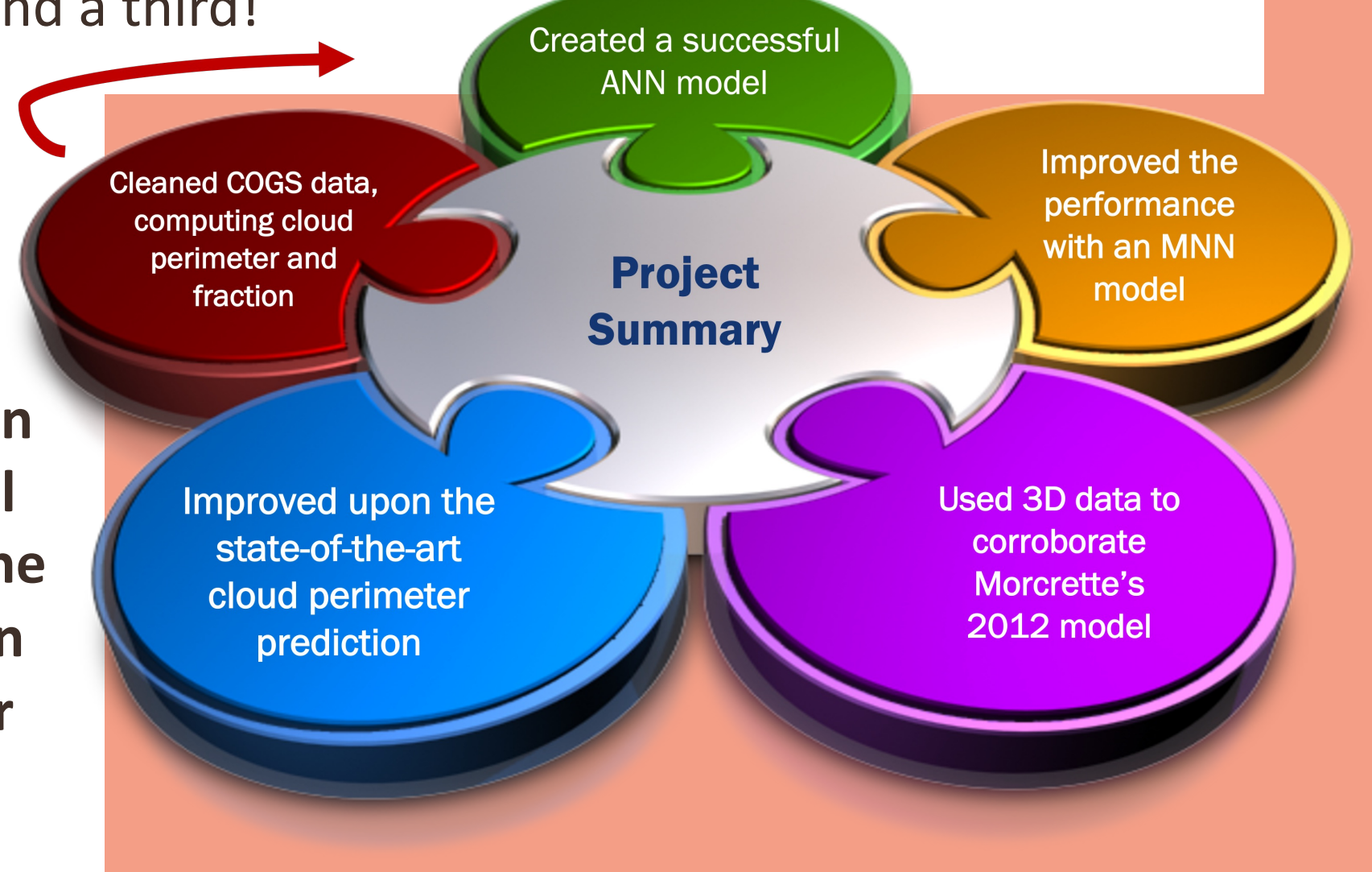
A key plot for validating the results was the true perimeters against the predicted perimeters (right). A perfect predictor would have all points along the red $y = x$ line. Of course, this isn't the case, but **there is a very strong correlation, and there doesn't appear to be any systematic bias.** Some of the largest true perimeters are under predicted by the model. However, it isn't surprising that the more extreme (and likely unusual) values are those that the model cannot accurately predict.



CONCLUSIONS

How good is the 16% error? **The most naïve prediction would be to simply predict the mean perimeter of the training data for all of the test set, which has a percentage error of just 93%!** So we have certainly beaten this baseline. In practice though, this naïve model would never be used. In 2012, Cyril Morcrette presented a relationship between cloud fraction and cloud perimeter, $P = \alpha F(1-F)$, where P and F are the perimeter and fraction respectively [1]. α is a parameter which can be adjusted to fit closely to the training data – the optimal alpha was found to be 0.64. The result of our work presented an opportunity to corroborate this hypothesis on 3D data for the first time and improve upon the prediction. **The results were that the model could achieve a percentage error of 23.8%!** So, it is clear that the relationship generalises and offers an accurate prediction of the cloud perimeter on 3D data. Not only has the model been corroborated, but our new machine learning model has further reduced this error by around a third!

Overall, we have cleaned the COGS data, in order to compute the cloud perimeters and cloud fractions. These have been used to train a machine learning model to parameterize the cloud perimeter. Two models were produced, with the more complicated MNN performing best. The cleaned data has been used to corroborate the results of the $P = \alpha F(1-F)$ model which was originally proposed for 2D data [1]. Not only has the strong performance of this pre-existing model been proved on our 3D data, but our new model has now been able to further reduce the error of the state-of-the-art prediction by a third.



Project Summary

Cleaned COGS data, computing cloud perimeter and fraction

Created a successful ANN model

Improved the performance with an MNN model

Used 3D data to corroborate Morcrette's 2012 model

Improved upon the state-of-the-art cloud perimeter prediction

REFERENCE LIST:

[1] Morcrette, C. J., 2012. Improvements to a prognostic cloud scheme through changes to its cloud erosion parametrization. *Atmospheric science letters*, 13(2), pp. 95-102.