Application of Deep Learning to Weather Forcasting Paramerisation Schemes

Student: John Griffith, Supervisor: Dr. Carl Henrik Ek

University of Bristol, Department of Computer Science

1. Introduction

Numerical weather prediction (NWP) is a computationally expensive problem. It involves vast quantities of data describing the exact state of the climate at any given location and time period. It's therefore necessary to break the given forecast area into discrete chunks. At a global scale the resolution of these chunks will start at 10 km in size, using smaller sizes quickly leads to impractical computational requirements.
The effect of atmospheric phenomena smaller than the available resolution must still be modelled. This is done through a **parameterisation** scheme which approximates the effect based off the available lower resolution data.

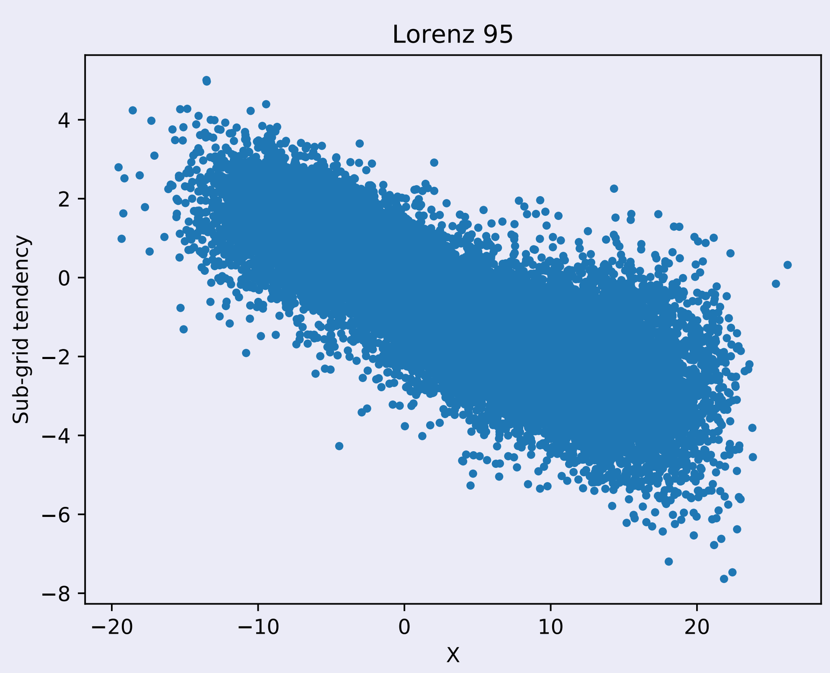
2. Lorenz Model

Lorenz 95' is a dynamic system which is often utilised by climatologists for testing NWP. The system is layered with each tier of variables influencing the layers above and below. In a climatic system a higher tier of variables could represent high amplitude, smaller phenomena which can't be resolved due to imaging resolution.
For a two tiered system, made up of sets of variables X and Y, the following update procedure is calculated on every time-step.

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - \frac{hc}{b}\sum_{j=J(j-1+1)}^{kJ} Y_j; \, k = 1, \ldots, K$$

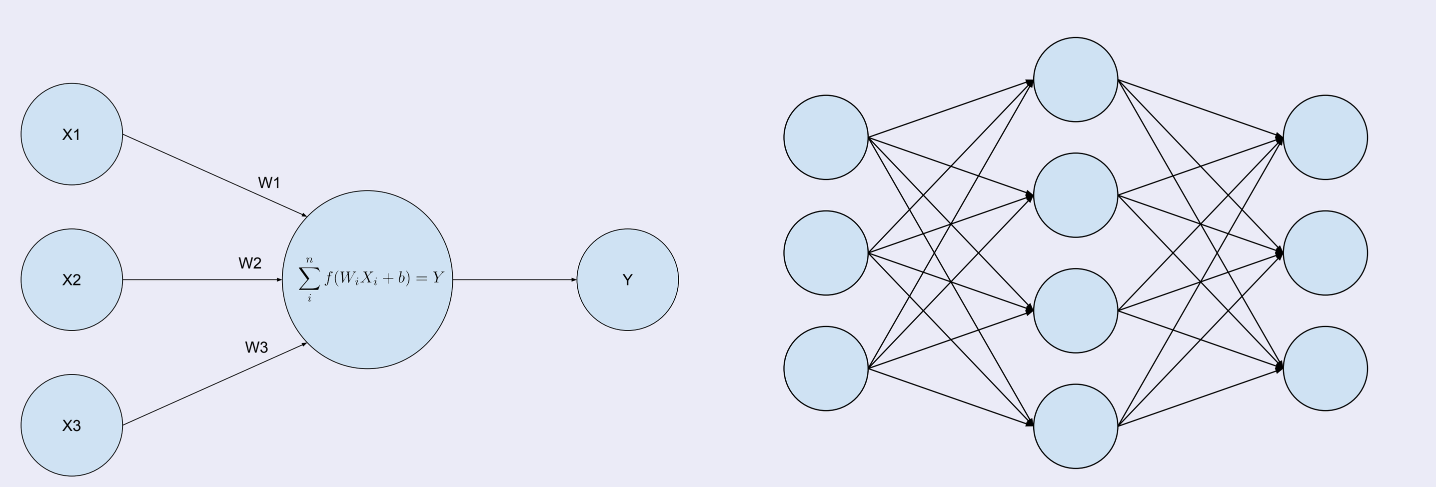
$$\frac{dY_j}{dt} = -cbY_{j+1}(Y_{j+2} - Y_{j-1}) - cY_j + \frac{hc}{b}X_{int[(j-1)/J]+1} j = 1, \ldots, JK$$

In the forecast model it is assumed that only the variables in X are resolved. The contribution from the variables in Y (known as sub-grid tendency) is replaced by a parametrisation scheme.



The shape of the sub-grid tendency is shown above.

3. Neural Network



A neural network is an interconnected group of nodes, each with adjustable weights and biases. Networks are typically formed in layers, with an input layer which is typically connected to one or more "hidden layers" and then finally an output layer.
Each neuron's output is a weighted balance of each input and is shifted with a bias term. An activation function f, such as Relu or Tanh, maps the output into a desired range.

$$Y = f\left(\sum_{n}^{i=1} W_i X_i + b\right)$$

The exact values of the weights and biases in each layer is obtained through training using a process called back propagation.

4. Uncertainty in Predictions

To capture the noisy data in the Lorenz model it is necessary to capture the distribution of the output. A deterministic neural network with a high number of degrees of freedom can be fitted to a wide variety of points, but does not indicate how certain it was of the result.
The spread of the results of a neural network with a stochastic output can be used to estimate the network's certainty for a given input. This can result in a more informed decision making process, especially in safety critical applications such as self driving cars and medical diagnosis.
Concrete Dropout and Bayes by Hypernet are two existing pieces of research that I'm currently investigating and building upon.

Concrete Dropout

Dropout is one possible technique to produce a stochastic output in a neural network. During the feed forward stage, nodes will be randomly "dropped" with probability p in each desired layer. Varying the dropout probability can allow a model to reduce the epidemic uncertainty; choosing an optimal value of p has been typically done through grid searching.
With Concrete Dropout the typically used Bernoulli Distribution is approximated with the continuous distribution called the "Concrete Distribution". This allows the inclusion of the dropout probability in the gradient descent optimisation process during training.

Bayes by Hypernet

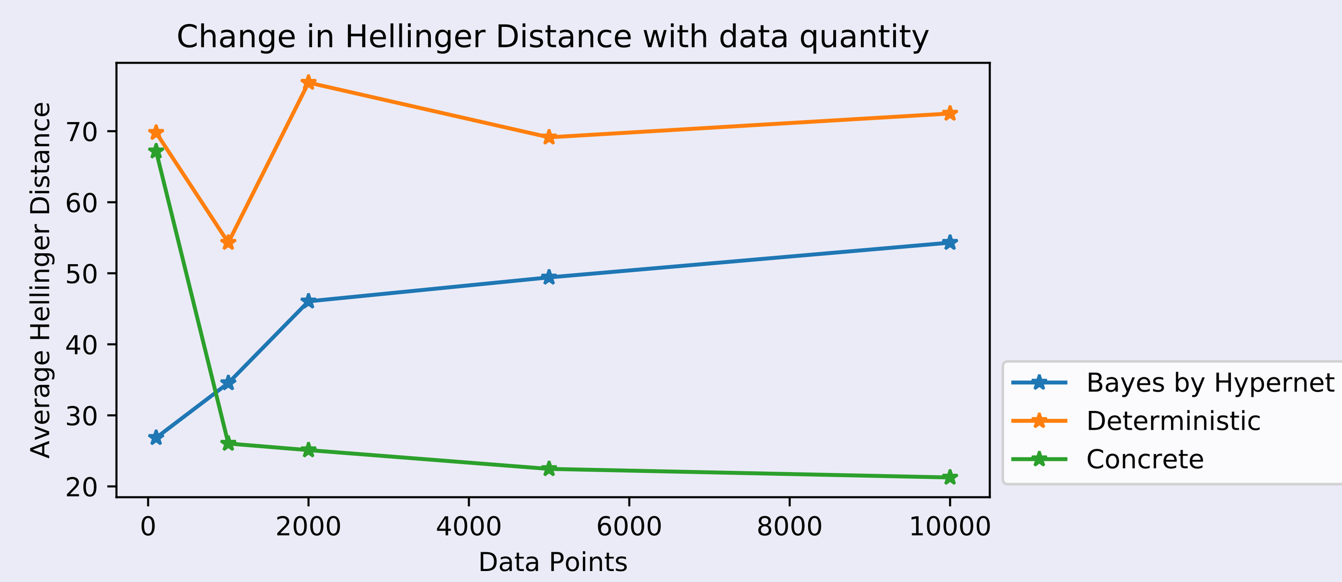
A HyperNet is an auxiliary model where the output is used to weight the main neural network. The HyperNet uses an implicit model which transforms noise, such as a Gaussian distribution, into a desired probability density function. This has been used extensively in Generative Adversarial Networks to transform Gaussian noise into desired output, such as images of faces. In this case the output is the set of weights used by the main network.

5. Preliminary Results

- Comparing the average mean squared error with different training dataset sizes

My results showed that the deterministic model, outperformed both techniques with the full dataset yet dropped off significantly with less than half the available data.

- Calculating the Hellinger Distance of the resulting sample distribution with different training dataset sizes



6. Future work

- Research the limitations of Concrete Dropout and Bayes by HyperNet
- Research why each technique produces the given results
- Continue building upon existing research, identifying improvements
- Continue to investigate how each technique responds to changes in data size

Yarin Gal, Jiri Hron, Alex Kendall. (2017). Concrete Dropout

Nick Pawlowski, Martin Rajchl, Ben Glocker. (2017). Implicit Weight Uncertainty in Neural Networks