Weather prediction

Aim

Make a model for predicting weather at the Lucas Heights weather station (max temperature and if there is going to be over 1mm of rainfall) on a day from the past 2 weeks of weather at the Lucas Heights weather station (max temperature and amount of rainfall) and find out how effective it is.

Background Information and Introduction

Weather forecasting has been around for millennia. Different methods were used for predicting the weather by different ancient civilisations. The modern age of weather forecasting began in 1835 with the invention of the telegraph. Messages about weather in different places could be sent almost instantaneously. By the late 1840s, the telegraph allowed reports of weather conditions from a wide area to be received almost instantaneously, allowing forecasts to be made from knowledge of weather conditions further upwind. Now weather forecasting is done in simulations that run on powerful supercomputers. The simulations take data from various sources (weather stations, satellites, pilot reports from routine aircraft flites etc.)

This model, however, will be built with neural networks. They require less computational power to train and almost none when in use. They are also a better choice for this case because data from only one weather station will be used. A neural network is essentially a self-training function. Neural networks are really good at finding patterns in datasets, which it hopefully will do in the weather data. The neural network type that will be used is a *Feedforward Neural Network* (FFNN). An FFNN contains 3 types of layers: input, output and hidden. Information can only go through FFNN’s in one direction: from inputs through the hidden layers to the outputs. Each neuron on a layer is connected to each other neuron on the previous layer through axons. Each axon has a weight on it. The value of a neuron is the sum of the value of all neurons of the previous layer multiplied by the weights of the axons that connect plus the bias of the neuron. Usually activation functions are applied onto the final value of the neuron. For more information see the Wikipedia page on Neural Networks (link in references). In this experiment the logistic function [y=1/(1+e^(-x))] (places the value between 0 and 1) will be used as the activation function as it is the most widely used activation function and negative outputs are not needed in this model. The neural network will have 28 inputs which is 14 (2 weeks) times 2 (rain and temp.), 2 layers, first one with 30 neurons, second with 20 (a balance between complexity and training speed), and 2 outputs (rain and temp. on the next day). Weights of the axons are adjusted by using the gradient descent algorithm where the value of the weight is pushed to its local minimum by subtracting the derivative of it multiplied by the learning rate which started at 0.001 and then got changed to 0.0001 when the neural network reached the precision limit at that learning rate.

Method

1. Download historical weather data
2. Make a reader and converter for the data to convert it into a readable format for my program
3. Write the neural network code
4. Write the code for passing values into the neural network
5. Train for a week
6. After a couple of weeks pass in the new weather data that the Neural Network didn’t train on from the Lucas Heights weather station and see how accurate it is.

Results

The spreadsheets attached have the predicted weather data. The different documents come from different amounts of training (‘Weather (5h training)’ from 5 hours of training, ‘Weather (10h training)’ from 10 hours etc.), ‘Weather (trained, 40 days)’ is 40 days of information that was not used in the training tested on the trained neural network. ‘Expected Rain’ and ‘Expected Temp’ are the rain and temp. passed through the logistic function. ‘Converted Predicted Temp’ is the predicted temperature passed through the inverse of the logistic function. ‘Data’ means information that was there for reference (the date and rainfall), ‘Output Data’ is data that was produced by the Neural Network, ‘Formula’ is data converted in the spreadsheet with the use of a formula. Graphs from ‘Weather (trained, 40 days)’ are also on the next page.

Conclusion

The ‘>1mm rain Correct (%)’ was surprisingly high, proving how predictable rainfall is. The temperature on the other hand, was more difficult as the exact number needs to be known, leading to around 1.5° average error. The average error also seemed to vary in different seasons as different seasons have slightly different weather conditions and the Neural Network had no way of finding out exactly what season it is.

References

*Weather forecasting*[*https://en.wikipedia.org/wiki/Weather\_forecasting*](https://en.wikipedia.org/wiki/Weather_forecasting)

*Artificial neural networks*[*https://en.wikipedia.org/wiki/Artificial\_neural\_network*](https://en.wikipedia.org/wiki/Artificial_neural_network)

*Feedforward neural networks*[*https://en.wikipedia.org/wiki/Feedforward\_neural\_network*](https://en.wikipedia.org/wiki/Feedforward_neural_network)

*Logistic function*<https://en.wikipedia.org/wiki/Logistic_function>

Here are the graphs from ‘Weather (trained, 40 days)’: