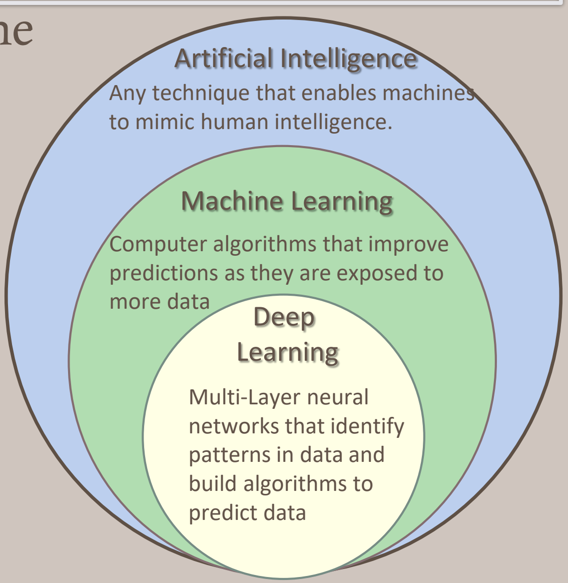
Neural Networks and Geospatial Data

Deep learning over space and time

K. Allison Lenkeit Meezan
Foothill College

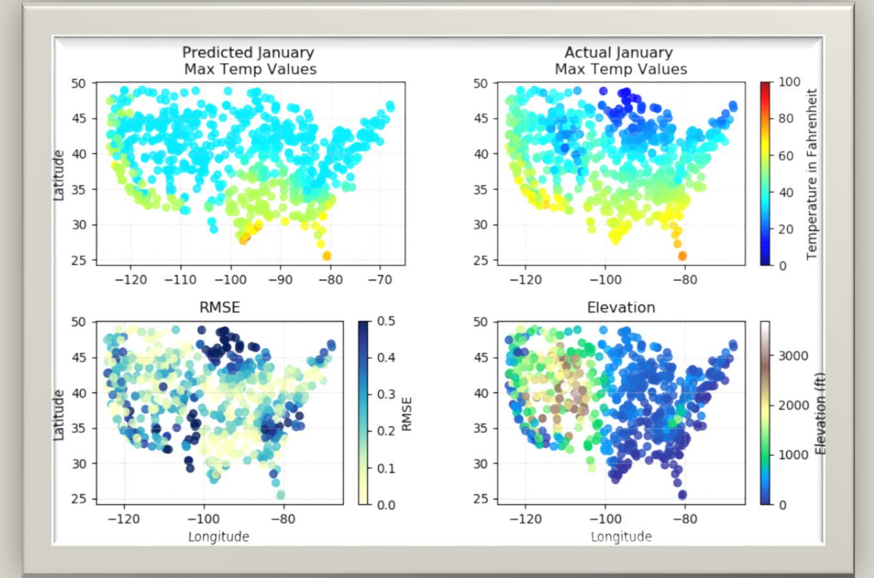
https://github.com/kmeezan/MachineLearningPub

This project began as an exercise to better learn Python. It has evolved into a fascination with deep learning with geospatial data sets. I have used it as a springboard to learn how to build neural networks with Tensor Flow on Jupyter notebooks and to explore data science. This is a summary of the work to date. Details and notebooks can be found on my GitHub page


Artificial Intelligence
Any technique that enables machines to mimic human intelligence.
Machine Learning
Computer algorithms that improve predictions as they are exposed to more data
Deep Learning
Multi-Layer neural networks that identify patterns in data and build algorithms to predict data

Coding Neural Networks with Python

As part of a six month sabbatical in winter/ spring 2019, I took a class in which we coded a feed forward/ back propagation neural network in Python. For the final project I trained the network to predict maximum average monthly temperature (TMAX) for Jan. 2019 using 4000 data points


Predicted January Max Temp Values
Actual January Max Temp Values
Temperature in Fahrenheit
RMSE
Elevation
Elevation (ft)
Latitude
Longitude

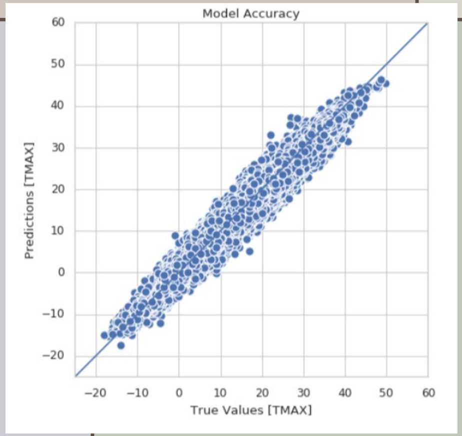
The network performed best with two hidden layers and five nodes. After training for 50,000 epochs it yielded an average RMSE of 0.2 F.

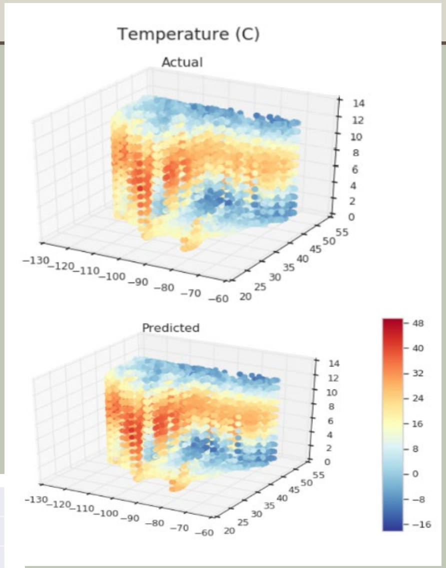
Neural Networks in Tensor Flow

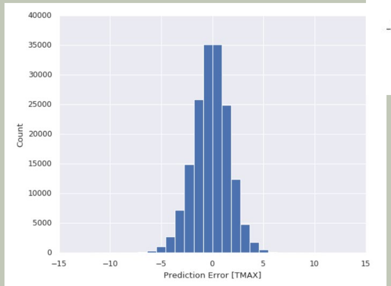
The next step in this project was to build a more extensive neural network using Tensor Flow. I used Google CoLab Jupyter notebooks to run Tensor Flow. The project utilized NOAA weather station data with input values of latitude, longitude, elevation and minimum average monthly temperature to predict maximum average monthly temperature for a given month. The data was trained on 20 years of weather station data.

Preprocessing data using Python

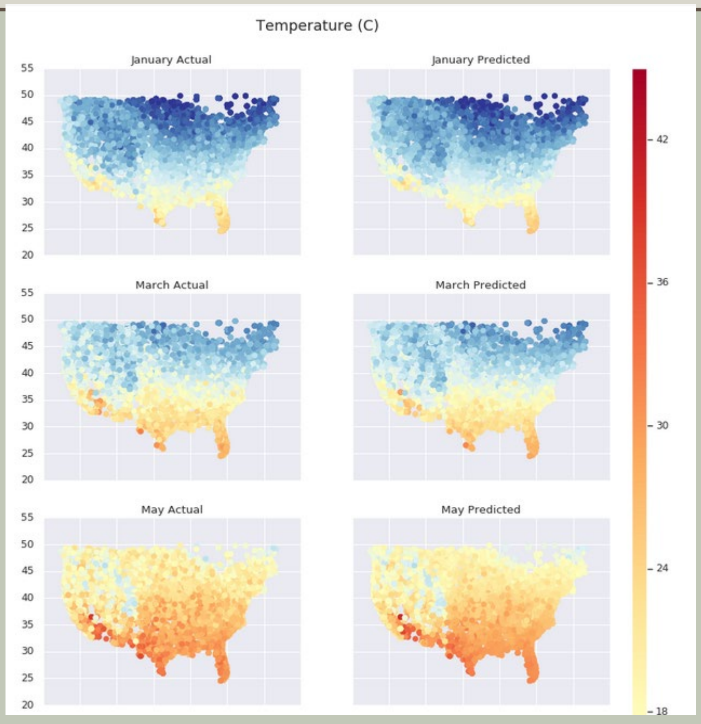
The data set was a total of 14.5GB of individual CSV files corresponding to weather stations around the world. Python was used to acquire the data from the files, filter latitude and longitude parameters and select a random list of 4000 stations in the continental USA which had data readings in the past 20 years. I wrote a Python program that used Pandas data frames to pull the relevant data from the selected stations, filtered out for null values and output the cleaned CSV files posted on GitHub for consumption by the CoLab notebook.







Adding patience to the model prevents overtraining. After around 60 epochs, the model yields a RMSE of 2.2C. The plotted error is Gaussian.


Temperature (C)
January Actual
January Predicted
March Actual
March Predicted
May Actual
May Predicted