Machine Learning Nanodegree Capstone Proposal

Kaushik Prakash

December 22, 2017

1 Domain Background

The weather is a very influential part of any person's life. It may affect the decisions they make on a daily basis. These decisions may be about whether or not to go to work or if it is worth it go shop that day. Thus, being able to accurately predict weather is essential for many people. Machine learning has been growing in popularity over the past decade. It can be applied to a myriad of things, weather being one of them. Currently, the weather is predicted on a real time basis based on the settings of any given day.

There has been extensive research on predicting weather. Such research has led to the use of very expensive satellites and sensors that are all compiled to make a prediction. This is a very exorbitant process and not always the most accurate for predicting far into the future. This is because the prediction is made solely on sensor data, as opposed to any a combination of both patterns in the past and sensor data. Although global warming may cause patterns to change, any algorithm will be account for this ahead of time based on the way it has changed in the past. Personally, this is a project I am very interested in because I can use it on a daily basis and compare it to the weather forecast. Simply the curiosity of identifying patterns in the weather is enough for me to continue through with this project.

2 Problem Statement

The goal of this project is to be able to accurately predict the temperature based off of past weather data. The solution to predicting temperatures that will be explored in this project is training a convolutional neural network on the past five years of weather data for a certain area, in this case Morganville. The input features for the data will be characteristics on a given day such as barometric pressure and humidity. The output will be the predicted temperature on that day. The temperatures can also be analyzed over an extended period time in order to find how global warming has affected the change in temperatures.

3 Datasets and Inputs

The dataset for this project will be obtained using the Open Weather Map API. The python wrapper for this API is by Github user Claudio Sparpaglione and will be used in this project. To access bulk history data, the API requires me to pay $10 per city. The API will return all kinds of information about a certain time period. These features will then be mapped to the temperature on any specific day. The city to be analyzed in this project is Morganville, NJ. Once the API key for bulk history has been obtained I can access the past five years of weather data. The data can be returned in either a JSON or a .csv file. I will be receiving the data in a .csv file because I am more familiar on how to use it with pandas. Using Scikit-learn the data will then be split into training, validation, and testing sets.

The weather data will contain all kinds of information about Morganville from average temperature to barometric pressure. Features that influence the temperature will all be used as input neurons for the neural network. The plan is to create a model that learns the relationship between these features and the corresponding temperature. The model will be used to predict the average temperature on a daily basis. Thus, the dataset will have information about each day for the past five years. To constantly improve the accuracy of the model, the testing set will be updated on a daily basis to keep up with the current patterns. This will be done by creating a function that checks the date and time and makes an API call depending on whether or not a day has passed. If so, the data point will be added to the testing set. After a week or a month has passed the network will run through the testing set again and update the weights through the use of backpropagation.

4 Solution Statement

I will be using a convolutional neural network in this project to accurately predict the daily temperature. The Deep Learning library, Keras, will be used to create the convolutional network. This network will be utilizing feed-forward backpropagation. Hence, it will be constantly updating and improving itself. Creating a complex network that is very computationally heavy is not of concern because in this case speed does not matter. Additionally, rectified linear unit, sigmoid, and/or softmax activation functions will be used for the neurons at each hidden layer. Training such a complex network on a large dataset will require a lot of compute power. Consequently, I will be using either Amazon Web Services or Paperspace to train and test the model. The convolutional network will find patterns and relationships between the temperature and the weather over the course of five years. My goal is for the network to be able to predict a reasonably accurate temperature, while also understanding that over time the temperature changes by a very small coefficient which we know to be global warming.

5 Benchmark Model

There are multiple options for a benchmark for this project. One possible benchmark model may be the weather channel and comparing my network's prediction to their prediction. This would be a very easy way to assess accuracy on a daily basis. Another possible benchmark might be to use random weights for the network to determine if its doing better than if a random temperature was guessed. This will tell me if my network has actually learned or not.

6 Evaluation Metrics

Prediction accuracy will be one of the main methods of evaluating my CNN's understanding of the data. The predicted temperature can be compared to both benchmark models mentioned in the above section. The prediction that the weather channel gives and the prediction that my model gives can both be compared to the actual temperature. Obviously there will be room for slight error since temperature has so many factors that can easily change. Additionally, patterns aren't strictly followed in the real world.

7 Project Design

7.1 All libraries

- Keras
- Tensorflow
- Numpy
- Pandas
- PyOwm
- scikit-learn

The structure of my project will begin with procuring the data. Using OpenWeatherMap's Weather API, I will pull bulk history from the past five years. The name of the file will be bulk-history.py.

```
#Pseudocode
function get_weather_data(place, start, end):
  owm = openweathermap object
   weather_objects = owm.weather_history(place, start, end)
   weather_data_csv = convert_to_csv(weather_objects)
   return weather_data_csv
```

Once the data has been obtained and saved as a .csv file it will be ready to be preprocessed for keras. The preprocessing will convert the data into a numpy array. The array will have the characteristics of weather as the feature and the temperature as the label It will then be split into training, validation, and testing sets. Once the data has been properly split, I will begin to create my network. The CNN will have to be built from ground up considering that the model has specific features that are all related to an output. The network will have 4 to 5 input neurons and only 1 output neuron. The input neurons will all be passed through a few hidden layers each with their own activation function. This will be the basic structure of the entire project. As I continue to develop some things will change slightly and these will all be noted in the Capstone Project Report