Project Requirements Document

Project: Wildfire Prediction

Team Name: Flare

Team Members:

- Kelly Lin (Team Lead)
- Alvin Tu (Scribe)
- Nick Ong
- Shuya Shou
- Steven Chang

Introduction

Background

Wildfires are a serious environmental concern that causes not only economic and ecological harm, but also puts human lives in danger. With the growing climate change predicament, wildfires are an increasingly dangerous problem, especially in wildfire-prone states like California. They pose a risk to local businesses by putting both structures and workers at risk. Various conditions increase the likelihood of wildfires, such as a large presence of flammable materials or dry climate. Creating predictive models can help authorities estimate the long term effects of climate change on wildfire occurrences, which can help reduce the damage on the local businesses, workers, and the ecosystem of the forests.

Problem Being Solved

Currently, many areas are in danger from wildfires, with few means to detect and prevent them in the long term. As climate change worsens, we need to prepare for the increased occurrences of wildfires since it will not only help reduce business risks caused from wildfires, but also potentially save lives and protect the environment. Our team will tackle this problem by using deep learning approaches to predict potential wildfires and use that information to help make crucial decisions.

Why is This Important?

Mitigating the risk and damages that occur from wildfires is urgent. Over the past ten years, the U.S. has averaged an annual 7.4 million acres burned, costing the country $2.4 billion a year. Additionally, the world's largest corporations face up to $1 trillion in damages from climate change. However, the most impacted group are individuals who have lost their homes, savings, and lives from fires. Investing in methodologies to predict wildfires would minimize the toll of these catastrophic events.

Existing Solutions

Studies have shown that the main causes of wildfires include climate change, human negligence, and natural causes like thunderstorms. Additionally, meteorological measurements such as temperature, relative humidity, rain, and wind are all known to impact forest fires. Most of the current solutions rely on a simulation method and mathematical models, which can be expensive and hard to maintain. Additionally, these tools face many problems such as low accuracy and long execution time. Other common methods include using Support Vector Machines, Decision Trees, Random Forests, and genetic algorithms. However, as an increasing amount of meteorological and climate data becomes available, it becomes difficult for these methods to remain effective, thus encouraging deep learning approaches. Recurrent Neural Networks (RNN) have been used in the past to do time-series forecasting, which uses past observed values to make future predictions. However, RNNs tend to suffer from long term dependency problems. On the other hand, Long Short Term Memory (LSTM) networks is a type of RNN that reduces this issue, which makes it a more suitable choice for our task.

Project Goals

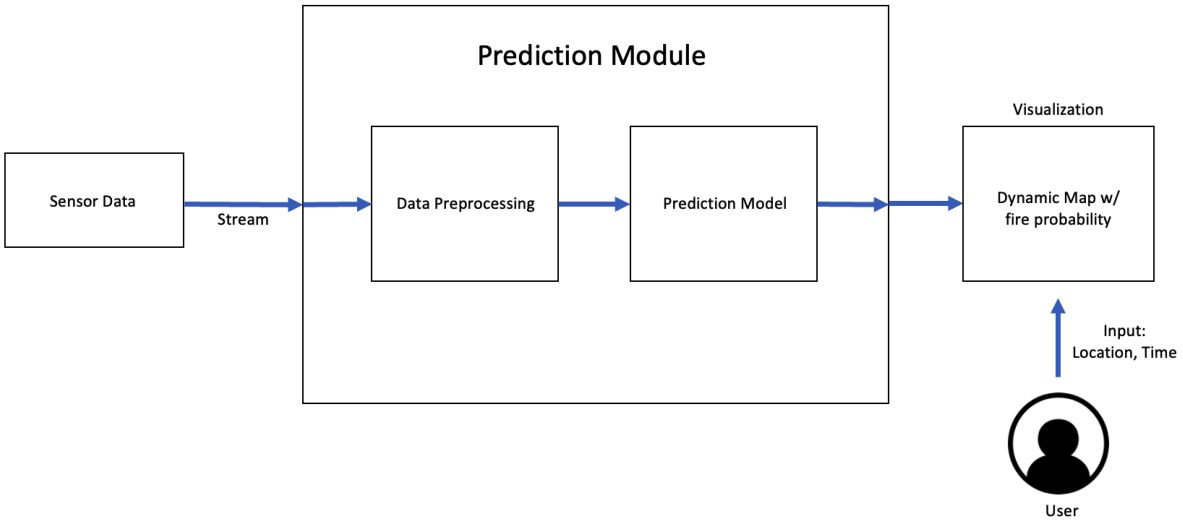
The goal of this project is to create a LSTM neural network to predict wildfire occurrences using time series climate data combined with weather variables, and other data including land cover types and topography. A wildfire prediction model will be developed for some states and then tested on those same states. Finally, the model will be improved to ensure robustness against outliers through data that predicts variances in the features we use, ultimately helping to guarantee the model will not fail in the future.

Assumptions

Our system will be built under a few assumptions. The biggest assumption is that climate change is one of the most important causes of wildfires, which is why the model will be trained with climate data. Additionally, because we are using a LSTM model for time-series forecasting, we are assuming that there is a long term trend in the climate data, and that this trend directly correlates with the probability of wildfire occurrences.

Overview of System Architecture & User Interaction

The diagram below shows an overview of our system. On the left side, we have live sensor data regularly streaming into our prediction module. Our prediction module contains two parts – a data preprocessing module and a prediction model. The data preprocessing module takes the incoming sensor data and processes them so that they can be used by the prediction model to make predictions. The prediction module is a network pre-trained to do time-series forecasting, using climate data gathered over the span of 10 years. The output of the prediction module is the probability of wildfires at different locations and time stamps. By inputting a specific location and time, the user can visualize the model output on a dynamic map.



Requirements

Note: We are filling in user stories with tasks because a machine learning model does not have many user stories

---

Research Tasks

1. Research past work on wildfire prediction using machine learning
    a. **Acceptance criteria:** Writing up detailed notes about past wildfire prediction research papers and how we can apply this knowledge.
    b. Github issue
2. Research open source climate/wildfire datasets
    a. **Acceptance criteria:** Writing up detailed notes about datasets with pros and cons of each in order to pick one or mix multiple.
    b. Github issue
3. Research climate specific domain knowledge
    a. **Acceptance criteria:** Writing up detailed notes on climate specific variables from research papers that affect wildfires to better understand how to analyze data
    b. Github issue
4. Research Pytorch and machine learning tools
    a. **Acceptance criteria:** Writing up detailed notes on tools and libraries that could aid in a wildfire prediction model and teaching the other members how to use them.
    b. Github issue

Model Building

5. Data analysis and feature selection
    a. **Acceptance criteria:** Thoroughly analyze datasets and select most relevant and impactful features to use to train models.
    b. **Scenario 1:** Select data from only one dataset
        i. Given dataset is thorough
        ii. We will select and analyze data from what we deem as the most relevant dataset
    c. **Scenario 2:** Select data from multiple datasets
        i. Given multiple datasets have useful data

        ii. We will pick and choose data from multiple datasets to combine into one dataframe for data analysis
    d. Github issue
6. Build LSTM model
    a. **Acceptance criteria:** Build a basic LSTM machine learning model using the analyzed dataset mentioned in user story 5 with training and validation samples.
    b. Github issue
7. Build model pipeline
    a. **Acceptance criteria:** Create a pipeline to streamline model training and hyperparameter tuning to easily create multiple models
    b. Github issue
8. Build testing pipeline
    a. **Acceptance criteria:** Create a pipeline to streamline testing
    b. Github issue

Using Model

9. As a PwC climate scientist, I can input data that matches the required features we have chosen
    a. **Acceptance criteria:** Successfully have method to input data
    b. **Scenario 1:** User inputs correct data
        i. Given the model is currently working
        ii. Given there is a method to input data
        iii. The user will be given long-term wildfire predictions for given data
    c. **Scenario 2:** User inputs incorrect data
        i. Given that model is currently working
        ii. Given model has been trained on specific data
        iii. The user will be given an error that the data is not correct

Visualizing Data

10. As a user, I can view areas of wildfire risks on a dynamic map
    a. **Acceptance criteria:** Map successfully outputs areas of wildfire risks in an easy to read format
    b. **Scenario 1:** User opens up map
        i. Given model is working correctly
        ii. Given model has made up-to-date predictions

iii. The user will be able to open the map and explore wildfire probabilities in different locations

c. **Scenario 2:** User zooms in on map
   i. Given map is up-to-date
   ii. The user will be able to see more specific cities and locations

d. **Scenario 3:** User clicks on specific area
   i. Given map is up-to-date
   ii. Given data is open source
   iii. The user will be able to see specific details as to why a probability for the clicked location was given

System Models

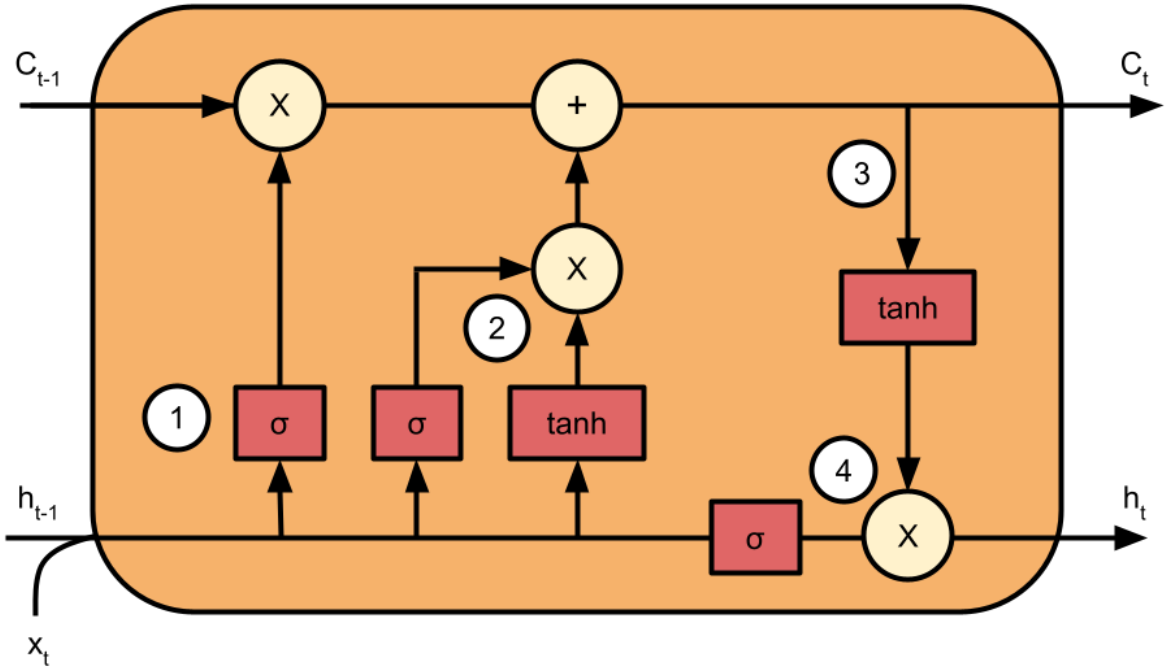


Figure 2. LSTM Architecture

C_{t-1} and C_t represent the input and output of the cell, with C_{t-1} being the output of the previous cell. Information relevancy comes in through h_{t-1} and x_t, in which the sigmoid (σ) and tanh functions determine their importance to the cell's state. To put it broadly, the top line represents the information stored, and the bottom line determines how that information is transformed

within the cell and beyond.

1) This sigmoid function is the forget layer where the output is between 0 and 1, with 0 meaning none of the information is remembered and 1 indicating that all of it should persist. This output is multiplied by C_{t-1} and determines how much of the previous information is remembered.

2) This sigmoid function is known as the input gate layer and determines which values will be updated. The tanh function next to it creates new candidate values. The product of these functions represent the new information that is added on top of the previous information, if persisted.

3) Together, the addition of the historical and new information make up the output C_t and are transferred through the right-most tanh function. This tanh function decides how the current information will affect the next cell.

4) The sigmoid function determines what part of the cell state will be outputted. It is then multiplied by output of the tanh function to become the h_t, one of the inputs of the next cell state.

Appendices

Technologies

- Python
    - Pytorch
    - Pandas
    - Numpy
- Google APIs (required for certain datasets)
- Github

References

- Understanding LSTM Networks -- colah's blog