Project Proposal

CSCI 470/575: Introduction to Machine Learning

Defunct Duderanch

Parker Epps

Kiersten Gaspar

Evan Bergo

Daniel Personius

Carson Stevens

Applied Project

Preference: Primary

Topic Area: Image Recognition, geolocation

Project Name: GeoKnower

Problem Statement: "Placed" in a random location in the United States, and given the ability to "move" around in Google Street View, can a model accurately predict where it is? A version of this game, GeoGuessr, exists, but is currently only playable by humans.

Proposed Solution: This project could be completed by building a classification model with the ability to interact with Google Street View, extract information from a series of images, combine information from multiple images, and predict its location based on previous data encountered. This is also known as image localization. Google's own version called PlaNet uses 126 million photos in their dataset and covers the entire planet. Some things that may differ widely between locations are types of vegetation, man-made infrastructure such as roads, stoplights, and architecture, people themselves such as clothing, markings, signs, languages, and weather. Specifically, this would be a multi-class, multi-label classification model, wherein it would report for each US state the probability the model is currently in that state.

The main model that we will adapt will most likely be a CNN or convolution neural network as this is primarily used when creating image recognition models.This is also the approach used in a paper focusing on a similar problem giving us confidence that this is a feasible approach and solution to the problem.[1]

In terms of developing the model, our primary choice will most likely be TensorFlow as it has a robust library for CNNs and image recognition.

Data: The model would be fed images from the Google Street View API, along with their associated metadata, such as latitude, longitude, and elevation. Each image would arrive as a 360 degree view of a location, which would be split into four images, each corresponding to a cardinal direction. Depending on the accuracy of this method, another dataset with the 4 cardinal direction images combine into a single panorama can be obtained and used. Also available to the model would be topographic data for each US state. This dataset can be used to determine state's border locations. As the model moves through space, it could record how its elevation changes over time and compare that to changes in elevation present in the topographic data. Another source of data cleaning, refinement, and selection could be population density found from this dataset or on ArcGIS. Sources include the United States Geological Survey(USGS) and ArcGIS, a popular geospatial platform.

[1] https://www.groundai.com/project/deepgeo-photo-localization-with-deep-neural-network/1

Applied Project

Preference: Secondary

Topic Area: Text prediction/generation

Project Name: Television Episode Writer

Problem Statement: Writing a script for a TV show requires a lot of resources. Doing so requires a team of several writers, all of which must agree on a general plot and character development curves. Writing scripts can be seen as a giant resource dump of time and money if not done efficiently. Furthermore, once a show has well established patterns, the actual content of a show can become quite predictable (especially in more profit driven, less artful shows). So why not try and save money by simply generating these scripts rather than hiring writers for multiple seasons.

Proposed Solution: This machine learning project serves to solve the stated problem by creating a machine learning algorithm that has previously written scripts as the input and a new script as the output. This could potentially be done through unsupervised learning or reinforcement learning. Traditional methods based purely on word frequency could be used but would not form quite as coherent sentence structure or overall episode plot.

Machine learning has already been used to develop a more effective car commercial by analyzing which commercials were effective and what aspects of the commercial struck the strongest chord with consumers. [2] So, another avenue of this project could be to use a pre-existing script reader/generator and from that analyze the new output episodes for how effective they would be. From there, the script generator could be modified to "optimize" the newly generated output script.

Upon further research trying to figure out how to get started with a script generator, we came across an existing one in a GitHub repository.[3] From this, we can determine whether or not this is a good script writer based on human judgement. We could then determine where the existing script generators falter and find aspects to create a better one using either our own modifications of this generator or by using machine learning itself to find what would make a better script.

Data: The following website has several TV scripts from 'The Best TV shows'. This dataset can potentially be used to download some scripts to gain some ideas. Then some further research for data can be conducted.

[2] https://www.latimes.com/business/hollywood/la-fi-ct-machine-learning-hollywood-20190411-story.html
[3] https://medium.com/deep-writing/how-to-write-with-artificial-intelligence-45747ed073c

Timeline

Critical Path The critical path of the project with expected completion dates of each task is:

1. Problem and solution research - 09/01 - 09/14
    a. Research papers
    b. Github repositories
    c. Dataset research and/or collection
2. Data collection, exploration, and processing - 09/15 - 10/05
    a. File download
    b. API connections
    c. Data cleaning, formatting, and normalization
    d. Dimensionality reduction (contingent)
    e. Train-test-validation split
3. Model initialization - 10/06 - 11/02
    a. Define performance metrics
    b. Choose multiple possible model types
    c. Feed data to model
    d. Train model
    e. Evaluate initial results
4. Evaluate and adjust model - 11/03 - 11/30
    a. Fine-tune hyperparameters
    b. Choose best-performing model
    c. Test against human performance (GeoGuessr only)
5. Present results - 11/30 - 12/05
    a. Create presentation
    b. Give presentation
6. Deploy to production (optional) - Future