Project Proposal

Udacity Machine Learning Engineer Nanodegree
Johnathan Widney

Overview:

- The project's **domain background** — the field of research where the project is derived;
- A **problem statement** — a problem being investigated for which a solution will be defined;
- The **datasets and inputs** — data or inputs being used for the problem;
- A **solution statement** — a the solution proposed for the problem given;
- A **benchmark model** — some simple or historical model or result to compare the defined solution to;
- A set of **evaluation metrics** — functional representations for how the solution can be measured;
- An outline of the **project design** — how the solution will be developed and results obtained.

Domain Background:

The domain background for this project is the heart of computer science: Code. This project will encompass programming, binary, machine language, metaprogramming, organic algorithms, and other core concepts.

Problem Statement:

This project will attempt to use machine learning to learn machine language. Using a deep reinforcement learning framework, I believe I can teach an AI agent how to program using binary. The problem I wish to solve is that our world is becoming more and more advanced in terms of technology. With quantum computing around the corner, we will need AIs that can process and understand these complex systems. This is the first step. Teaching an AI the core fundamentals of programming, and how to accomplish this efficiently at the machine language level, will drastically improve computer systems; and prepare it for the future.

Datasets and Inputs:

The datasets for this project will be the official Python documentation library. From fundamental functions, methods, and objects, to sample code provided by the documentation. Each code snippet will be preprocessed by compiling it into binary. The goal of the AI is to take the binary code and determine what task it is associated to. For example, I will have a code snippet that (if executed) will power a particular LED light on a computer. The AI will have this represented by binary, but not the named task. It will have to learn it and adequately provide its own program to accomplish the task.

Solution Statement:

In the learning phase, it will learn binary programming and learn the associated tasks to the code snippets. It will then go through a phase to validate its new learned skills. In the testing phase, the agent will be given a requested task, and it will program its own code. The code will be tested against example code snippets.

Benchmark Model:

The BAYOU AI created by researchers from Rice University made an agent that would be able to take a desired program from a user, using only a few key words, and output Java code that was very close to the desired outcome. While I will be taking this a bit further by going to the machine language level, the concept is similar.

Evaluation metrics:

The AI will be penalized for any errors that are raised in its code. Fatal errors will be penalized heavily. The AI will be rewarded for programming a function that performs the desired task, without any errors. The AI will also have a penalty for the code failing the task, even if no errors are raised.

Project Design:

I will bring a list of sample code that is sourced from the official Python docs website, and other public domain websites that have free sample code. Those code snippets will then be put through a compiler to convert them into binary. That will be the dataset. There will be 3 phases to the project: Training, Validation, and Testing. After splitting the dataset into training and testing data, they will be put through the agent model. The agent will learn the patterns of the binary code snippets and devise its own programming structure. Once it has its own programming methodology, it will move to the validation phase, the agent will be given sample tasks and it will provide its own code to accomplish the task. This will be validated against code snippets that truly accomplish that task. The validations will be used to further improve the model in its ability to code. The final testing phase will be similar to the validation phase, again the agent will be given tasks that are requested, and the accuracy of the agent will be tested against these code snippets. The target accuracy for this project is 87% or greater. Points will be awarded to the agent for code that has 0 raised errors, a program that completes the requested task, and done with a lower total file size in the code. The agent will be penalized for any and all errors that may arise in the code, certain errors will have larger penalties. It will also have penalties for writing a program that fails to complete the requested task, and for writing a program with many lines of code and having a high file size.