IMPLEMENTATION OF A

NEURAL NETWORK MODEL

TO **LEARN**

TO PLAY CART-POLE

Ashwin.M.S

1CR17CS024

Introduction to Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

**What are neural networks?**

Artificial neural networks (ANNs) are computing systems inspired by the biological neurons that constitute animal brains.Such systems "learn" i.e. progressively improve performance on tasks by considering examples, generally without task-specific programming.

Each connection (a simplified version of a synapse) between artificial neurons can transmit a signal from one to another. The artificial neuron that receives the signal can process it and then signal artificial neurons connected to it.

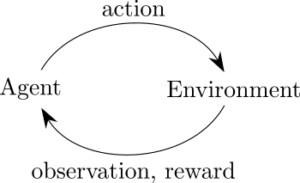
**OpenAI Gym**

**OpenAI** is a non-profit artificial intelligence (AI) research company that aims to promote and develop friendly AI in such a way as to benefit humanity as a whole. The organization aims to "freely collaborate" with other institutions and researchers by making its patents and research open to the public.

## Gym is a toolkit for developing and comparing reinforcement learning algorithms. The gym library is a collection of test problems — environments — that you can use to work out your reinforcement learning algorithms. These environments have a shared interface, allowing you to write general algorithms

Our own agent can be written using your existing numerical computation library, such as TensorFlow or Theano.

In this project we shall be implimenting the agent-environment loop as shown below:

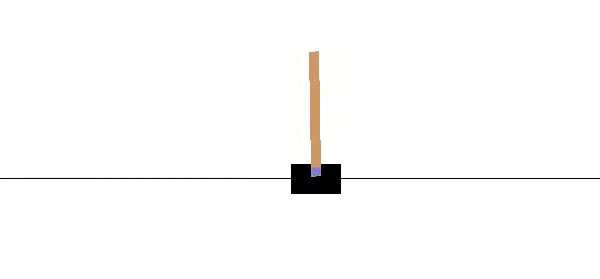


**Cart-PoleV1**

**Problem Description:**

A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The system is controlled by applying a force of +1 or -1 to the cart. The pendulum starts upright, and the goal is to prevent it from falling over. A reward of +1 is provided for every timestep that the pole remains upright. The episode ends when the pole is more than 15 degrees from vertical, or the cart moves more than 2.4 units from the center.

This environment corresponds to the version of the cart-pole problem described by Barto, Sutton, and Anderson("Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problem", IEEE Transactions on Systems, Man, and Cybernetics, 1983).



**The Approach**

**STEP 1 : Generation of Training Data**

The training dataset requried for the ML model is generated by making random moves in a game.The **score , observation** and **action** of each game is stored in the game memory. Defined number of games is played and data is logged after each game.

All the games whose final score are beyond the **required score** are used to populate the training set.The output from the observation is converted to one-hot form to efficiently manage categorical data.

**STEP 2 : Building the Neural Network Model**

The neural network model used is made up of 5 hidden layers of sizes 128 , 256 , 512 , 256 and 128 nodes.The nodes in the hidden layers are activated by the ReLU activation function.All the hidden layers have a dropout rate of 30%.This is done to avoid over-fitting of data.

The final-layer consists of 2 nodes and the softmax activation function is used to obtain the probability of each action(left or right).

Regression is performed on the NN to optimize for the loss function i.e categorical\_crossentropy.The loss function is optimized using the Adam optimizer.

**STEP 3 : Training the Model**

As the next step, segregate out training data into X(observations) and y(target mapping).If no model is initialized , build the model as described in step 2.The training data is then fit into the model and thenumber of epochs is defined.An **epoch** is a measure of the number of times all of the training vectors are used once to update the weights.

**STEP 4 : Testing the Model**

The model is tested by calling the model.predict() method of the trained model.This returns a set of predictions as a list.The list contains the probability of the best action that should be taken which will help maximize the score and minimize the loss function.

The model can be evaluated by taking the mean of all scores over a defined number of games.

**Key Terms and Functions**

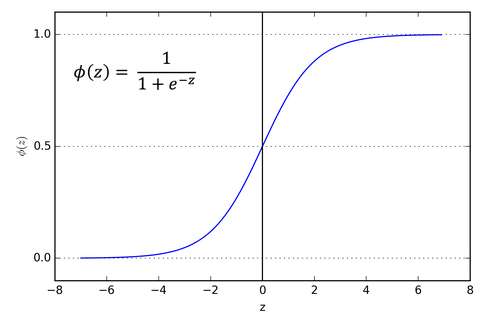
**1) Activation Functions**

They introduce non-linear properties to our Network.Their main purpose is to convert a input signal of a node in a A-NN to an output signal. That output signal now is used as a input in the next layer in the stack.

The activation functions used in the program are:

1.1) **Softmax or Sigmoid or Logistic Activation Function**

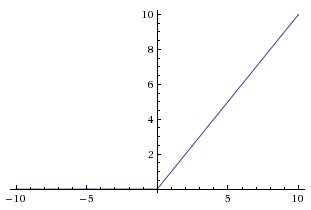
The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output.Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.



1.2) **Rectified Linear Units(ReLU)**

The Rectified Linear Unit is the most commonly used activation funciton in deep learning models.A rectified linear unit has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input.



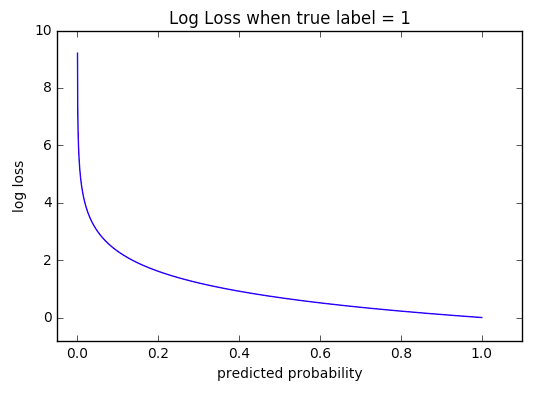


**2) Loss Function:**

A Loss Function is used to optimize the parameter values in a neural network model. Loss functions map a set of parameter values for the network onto a scalar value that indicates how well those parameter accomplish the task the network is intended to do.

**2.1) Categorical Cross Entropy :**

Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value. A perfect model would have a log loss of 0.



**3) Adam Optimizer**

The name Adam is derived from adaptive moment estimation.

Adam is an optimization algorithm that can used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.Adam is different to classical stochastic gradient descent.Stochastic gradient descent maintains a single learning rate (termed alpha) for all weight updates and the learning rate does not change during training.A learning rate is maintained for each network weight (parameter) and separately adapted as learning unfolds.

**References:**

Github Repository : <https://github.com/Pythonista7/Open-Ai-Experiments>

(contains jupyter notebook,documentation and .py file)

Sources : <https://gym.openai.com/>