Walking Humanoid

Team members: Milind Prajapat, Kartik Garg, Sachin Negi, Surabhit Gupta.

Abstract

The main aim of our project is to model linear motion of biped/humanoid over different terrains using real time data. The reason for us picking this topic was to understand the movement of legs in human body and make more fluid animation of real life-like computer graphics as well as visualisation of human gait data which can be used in research, in the field of prosthesis. To achieve our goal we have implemented deep learning algorithms with the help of TensorFlow in Pybullet environment.

Problem Addressed

The problem, our project addresses is the analysis and the representation of human gait data. We aimed to use real time data to make more fluid and lifelike animations as well as to make a tool for aiding in research involving gait analysis.

Our Approach

The approach we have used here is deep learning which we have implemented using TensorFlow. Deep learning and neural networks are a perfect fit for an AI which learns by itself and gets perfect over the time. Thus choosing a neural network with perfect hidden neurons, the input data, the output data, and the architecture makes an ideal AI. It is capable of adapting to any function and performing any task, whereas Pybullet provides us a good platform for simulating our model in an artificial world.

Key Concepts Used

Below we have described all the concepts that have been used in making of this project.

TensorFlow

It is a highly flexible system which provides multiple models or multiple versions of the same model that can be served simultaneously. The architecture of Tensor Flow is highly modular, which means you can use some parts individually or can use all the parts together. Tensor Flow has made it possible to play around an idea on your laptop without having any other hardware support. It runs on GPUs, CPUs, desktops, servers, and mobile computing platforms. It can also be used to train and serve models in live mode.

- It allows Deep Learning.

- It is reliable (and without major bugs).
- It is a skill recognized by many employers.
- It is easy to implement.

Hidden Layers and Hidden Neurons

There are really two decisions that must be made regarding the hidden layers: how many hidden layers to actually have in the neural network and how many neurons will be in each of these layers.

- 0 layers - Only capable of representing linear separable functions or decisions.
- 1 layer - Can approximate any function that contains a continuous mapping from one finite space to another.
- 2 layers - Can represent an arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping to any accuracy.

Using too few neurons in the hidden layers will result in something called under-fitting. Under-fitting occurs when there are too few neurons in the hidden layers to adequately detect the signals in a complicated data set. Using too many neurons in the hidden layers can result in several problems. First, too many neurons in the hidden layers may result in over-fitting. Over-fitting occurs when the neural network has so much information processing capacity that the limited amount of information contained in the training set is not enough to train all of the neurons in the hidden layers.

Choosing two hidden layers, and a total of 384 hidden neurons; 256, 128 hidden neurons in 1st and 2nd hidden layer respectively, makes our neural network an ideal network for performing the required task.

Input and Output Layer

Input data must be chosen intelligently, it can greatly affect the networks efficiency. Choosing data which is highly responsible for performing the desired task can be a great choice.

- As in our case, we have 44 neurons in the input layer:
- Relative position of COM (1)
- Sine and cosine of the angle made with target position (2)
- Rotational velocity of base multiplied by a scaling factor of 0.3 (3)
- Roll and Pitch values of base (2)
- Current relative position of each joint present in the humanoid in terms of angle, calculated from its lower and upper limits (17)
- Velocity of each joint present in the humanoid multiplied by a scaling factor of 0.1 (17)
- Bool values for contact of foots, 1 for in contact and 0 for not in contact (2)

This data is highly responsible for walking, running and balancing purpose. Also this data is been clipped in between -5 to +5 to avoid large calculation time. Similarly output layer should give out the output data which we will be using to control the humanoid, i.e., values of force for each motor to be rotated, giving us 17 outputs or neurons in the output layer. The output data is then further processed, first clipped in between -1 to +1 to avoid large changes as it is then multiplied by a huge scaling factor, that is different for each joint present in the humanoid and the resultant is further multiplied by a common scaling factor to have the desired output.

Pybullet

Pybullet is an easy to use Python module for physics simulation, robotics and deep reinforcement learning based on the Bullet Physics SDK. With Pybullet we were able to simulate the real time physics into an artificial world, providing us a great platform to execute our ideas and helped us to develop the correct approach to the problem. It acted as a base for our problem and deep learning algorithms as well.

Conclusion

Within these two weeks, we were able to complete this project and train a model which allows a biped/humanoid to walk on level ground. However, we believe that we have just scratched this project's potential and there is lot more to be achieved in line with our original goal.

Due to lack of time, and limited knowledge of TensorFlow and deep learning algorithms as well as limited availability of datasets, we were not able to train a model which would allow our biped/humanoid to move on different terrains, i.e., walking on a staircase, tackling different obstacles in its path.

We hope that, with continuous practice in TensorFlow and deep learning algorithms, in general and access to a larger and a better data set, we will be able to make a significance progress towards our goal over the time.

References

1. https://github.com/bulletphysics/bullet3/tree/master/examples/pybullet/gym/pybullet_envs/examples
2. https://www.tensorflow.org/

Acknowledgement

The project is the outcome of diligent and hard work of each team member. We all were inspired and motivated by each other's enthusiasm towards this project.