Models

Introductions

We use different AI models to create trading bots in our product.

These architectures can have different capabilities and properties. In this document we will outline some of these properties and give an overview of the architecture of each at a low level.

Details

All models are trained using Evolutionary Strategies.

Common Hyperparameters: * Epochs: 5000 * Population: 256 * Sigma: 0.05 * Lr: 0.01

Checkpoints are versioned using 2 integers, one for the training version (V) and a data version (D), that update training set and feature input respectively.

The fist number after the model name will be part of the hyperparameters to the specific model, generally the amount of hidden units.

These architectures are generally stored in the evosim module.

MLP

A shallow neural network with just 2 fully connected layers.

```
[Features]
    |
  Linear
    |
  ReLU
    |
  Linear
    |
[Action]
```

Filter Network

A simple deep neural network based on the Filter Blocks from WaveNet. Instead of convolutional layers it uses fully connected layers as these are more efficient on CPUs. These filter block act like gates by using 2 different activation functions and multiplying them. In WaveNet they are used in parallel to accumulate features of an input wave.

It has an optional `[Aux]` path for adding recurrent functionality, however this isn't currently utilized.

Architecture

```
 [Features] [Aux]
      |       |
    Linear  Linear
      |       |
    Tanh     Tanh
      |       |
      + -----/
      |
    FilterBlock x3
      |
      |
      |-------\
      |        |
    Linear    Linear
      |        |
    Sigmoid   Sigmoid
      |        |
    [Action]  [Aux]
```

Filter block

```
        |
      Linear
        |
       PReLU
        |
    |--------|
  Linear    Linear
    |        |
  Tanh    Sigmoid
    |        |
     ----*---
         |
```

GRU/LSTM

A proper recurrent neural network architecture. GRU a relatively new type of recurrent neural network, and LSTM a similar but older recurrent neural network.

GRU is claimed to be more efficient than LSTM while providing similar performance, but this can be problem-dependant. LSTM uses 2 hidden states, a long and short term memory (LSTM).

Recurrent neural networks traditionally are harder to train than regular ‘stateless’ neural networks because the gradient decays after a few iterations. With our training method this isn't a problem, however we do have the problem which is outlined in the paper 'Safe mutations in deep and recurrent neural networks through output gradients'

!Architure Image

Source code

Code is available at: https://git.fhict.nl/I404788/evosim/-/tree/master/models