

Teaching a Neural Network to Fly Autopilot

Design and Extraction of Training Data:

The training data consists of samples of the game state and the actions taken by a player. The `sample()` method extracts features from the current game state, Distances to the top and bottom of the opening.

These features are combined into a double array and stored alongside the action taken (up, down, or stay) during data collection. This approach allows the neural network to learn between game state features and actions for not hitting the walls.

Neural Network Topography and Configuration:

An input layer with a size matching the number of features extracted from the game state (5 – `immediateTop`, `immediateBottom`, `avgTop`, `avgBottom`, `playerRow`).

A hidden layer with 30 neurons to capture complex relationships between features.

An output layer with a single neuron using a TANH activation function. The output value represents the probability of moving up (-1) or down (1) or stay (0).

The code also utilizes a simple normalization technique where each feature in the feature vector (vector) is divided by `MODEL_HEIGHT`. This approach prevents features with larger scales (like distances) from overloading the training process compared to the player's position. By having all features on a similar scale the network can learn the relationships between them more effectively.

Training Hyperparameters:

Learning rate: Controls the magnitude of weight updates during training. A small learning rate ensures gradual improvement while a high rate might lead to instability.

Training epochs: The number of times the entire training dataset is passed through the network for learning. More epochs allow for better convergence but increase training time.

These hyperparameters are tuned through experimentation to find a balance between learning speed and achieving good performance.

Rationale: Carefully chosen hyperparameters prevent the network from underfitting (not learning enough) or overfitting (memorizing the training data without generalizing well).

How to Run:

Windows:

```
java -cp ./encog-core-3.4.jar;./ai.jar ie.atu.sw.Runner
```

Linux / Mac:

```
java -cp ./encog-core-3.4.jar:./ai.jar ie.atu.sw.Runner
```

Game is set to autopilot (true) via flag in `gameWindow.java`. If you wish to play manually and train the neural network set the flag to false. Click 'S' to start collecting data and click 'S' again in this mode to save and train the neural network whilst playing the game.