Neural Networks With Super Mario Bro's

User Manual

Pre-requisites:

1. Lua version 2.5 or higher: https://www.lua.org/download.html
2. BizHawk Emulator: https://github.com/TASVideos/BizHawk/releases
3. Super Mario Bro's ROM file(A quick google search will find this available for download)
4. Java jdk 1.7 or higher: http://www.oracle.com/technetwork/java/javase/downloads/index.html
5. A emulator save state file for the start of the first level saved in the systems save_states folder called "SMB_L1-1.state"

Installation Instructions

1. Download the current release at https://github.com/Eoin-Mur/SMB-Neural-Network/releases
2. Extract the archive file to your desired location. This will extract the necessary scripts and Java classes and also the directory structure that the system uses for storing files
3. navigate to the Java-files folder and compile the two Java files with: " javac *.java "

Contents

User Guide

1 Getting Started

1.1 Loading The Super Mario Bro's ROM

Open BizHawk and from File->Select Rom select the ROM from the window explorer

1.2 Loading The Main Script

1. In BizHawk open the lua console from Tools → Lua Console
2. In the Lua console open the script Lua-Scripts/SMB_Neural_Net.lua from Script → Open Script
3. You should see the main user interface in the emulator window as below

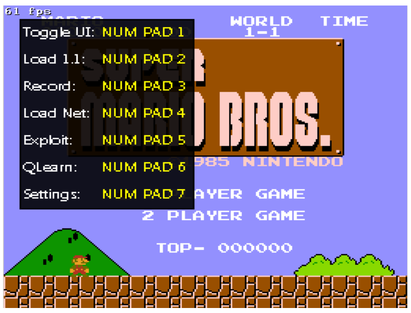
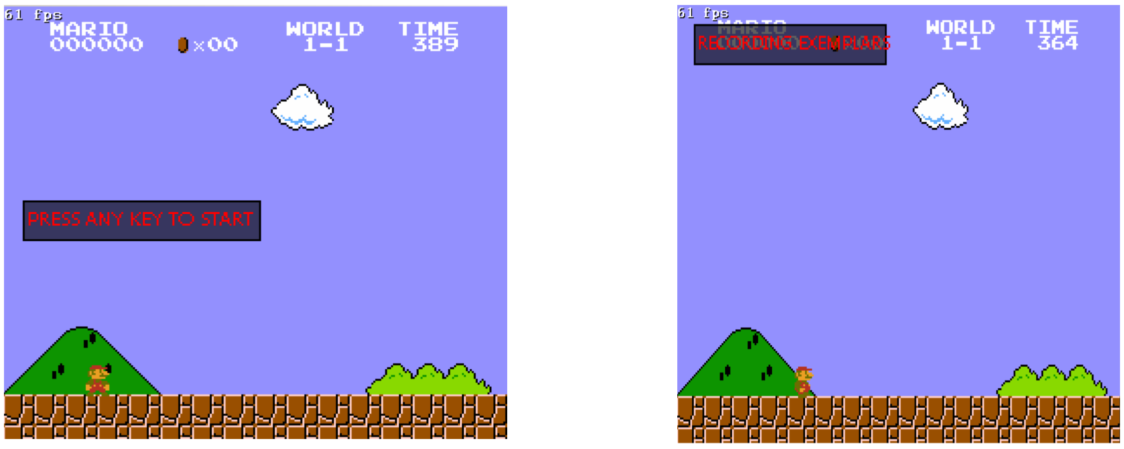


Figure 1: Main User Interface

1.3 Navigating The System Functions

From the main user interface you will see 7 options these option are selected by pressing the corrisponding key on the number pad.

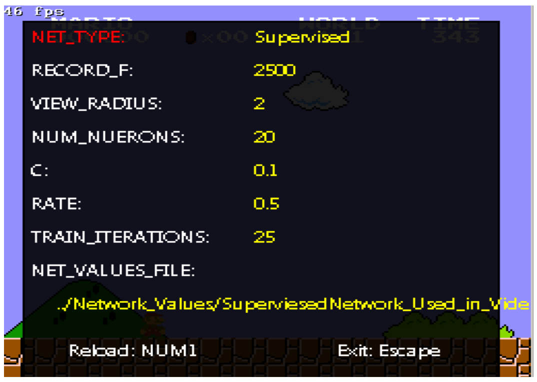
1. Toggle UI: This hides/Displays the user interface
2. Load 1.1: This loads the save state file "save_states/SMB_L1-1.state"
3. Record: This loads the level 1 save state placing the user at the start of the first level. The user is prompted to press a key to begin recording their game as the exemplars for training the network with supervised learning(Figure 2)
4. Load Net: This loads the network values file specified in the configuration file to the system
5. Exploit: This executes the network currently loaded in the system
6. Qlearn: This begins training the network using Q-learning the network type varaible in the configuration file must be set to either "Reinforcment" or "ReinforcmentMul"
   1. Reinforcment: uses the network architecture to pass the actions in with the inputs
   2. ReinforcmentMul: uses multiple Networks representing each of the actions
7. Settings: Displays important configuration variables currently specified in the configuration file(Figure 3). Pressing numpad 1 in this screen will reload the configuration file updating the system with the new values. Press esc to exit



PRESS ANY KEY TO START

RECORDING EXEMPLARS

Figure 2: Recording exemplars screens



```
NET_TYPE:            Supervised
RECORD_F:            2500
VIEW_RADIUS:         2
NUM_NUERONS:         20
C:                   0.1
RATE:                0.5
TRAIN_ITERATIONS:    25
NET_VALUES_FILE:
  ../Network_Values/SuperviesedNetwork_Used_in_Vide

     Reload: NUM1                    Exit: Escape
```

Figure 3: Settings Screen

1.4 Recording Exemplars

1. In the configuration file: “config.txt” under the variable RECORD_F enter the number of frames of gameplay you wish to record for the exemplars. *NB: It is important to make user there is a space between the variable name and the value when modifying configuration values as if there is no space the value will not be read correctly*
2. Check that the correct number of frames to record is loaded into the system by checking the settings screen(NUMPAD 7). If the value shown is not the same as what is save simply hit NUMPAD 1 to reload the file and you should see your value displayed. Hit esc to return to the main window
3. In the main window press the **record** key (NUMPAD 3)
4. this will load up level one and prompt you begin playing the level
5. While playing the level you will see the notification that they system is recording the exemplars once the desired number of frames is reached this prompt will disappear and the main UI will redisplay.
6. Once finished recording you can find you exemplar file saved in the folder Exemplar_Files with the current time and date in the file name.

NOTE: The system will record exemplars with the maximum tile radius of 7. if you wish to reduce these to a smaller radius please see the next section **Pre-Processing Exemplars** for instructions on how to do so.

2 Pre-Processing Exemplars

The folder Java-Files contains two java classes PreProcess_Exemplars and ReduceInputs

2.1 Reducing The Inputs In Exemplars

After recording exemplars you will find that the exemplars are recorded with the largest radius that being of 7 and you may wish to use a smaller radius and thus a smaller number of inputs in you network.

1. Run the Java class ReduceInputs passing the exemplar filename and the radius you wish to reduce it to the lowest possible radius is 2.
2. Once the program finishes you will find your reduced exemplar file in the folder Pre-Processed_Exemplars. It will have the same file name excepted extended onto it will say reduced-Inputs followed by the new size.

2.2 Pre-Processing exemplars

The pre processing function included in the system works by removing the duplicate exemplars and if an two exemplars are present with the same input but different outputs the program will select the exemplar which has the highest number of “on”(1) outputs.

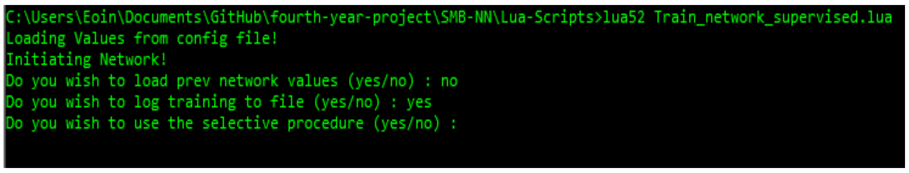
1. Run the Java file PreProcess_Exemplars passing the exemplar filename in the arguments.
2. Once the program finishes it will output two files in the Pre-Processed_Exemplars folder the first being the exemplar file with no duplicates indicated by the ND added to the file name and the second being the exemplar file with no duplicates and exemplars which have the same input the on with the highest number of “on” outputs chooses indicated by ND_HOOC suffix in the file name

This is just one pre-processing method, and you are not limited to using it you can develop your own! This system is completely free to tinker and change with all source code available to view and edit.

3 Training your network

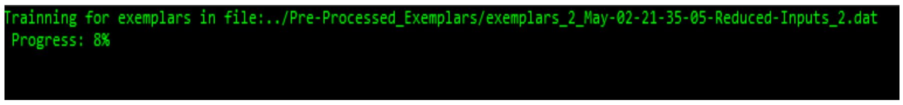
3.1 Supervised Learning

1. In the configuration file config.txt specify the values for the following variables
   1. TRAINING_FILE: the file location of the exemplar file you wish to use to train the network
   2. VIEW_RADIUS: the view radius of tiles around mario in the exemplars file. It is very important that this value matches the view radius of the radius in you exemplar file and this also determines the number of inputs to your network. If you have a larger input radius specified then what is in you exemplars you will find the network will not function as expected. A radius smaller then then what is in the exemplars file will result in an array.outOfBoundsException.
   3. NUM_NUERONS: this is the number of neurons you want to be in you networks hidden layer. Remember the higher this value goes the more computational heavy your network will be and will have an effect on training times.
   4. C: this is the value which determines the initial starting weights of the network it is recommended to leave this at its default value of 0.1
   5. RATE: This is the learning rate used in the backpropigation algoritm this value should always be between 0.0 and 1.0
   6. SIGMOID_TYPE: this is the type of sigmoid function you want to use for you activation function. Again it is recommended to use the default binary value
   7. TRAIN_ITERATIONS: this is the number of times you want to show the network your exemplars. To few times and the network might not classify the inputs correctly to many times and the network may over train and work only for what it was shown and not very well for values it has never seen before.
   8. NET_VALUES_FILE: if you wish to retrain a network previously trained specify the file location here.
2. With your network configurations set and saved run the script found in the folder Lua-Scripts Train_network_supervised.lua
3. You will be presented with 3 options (Figure 4):
   1. Do you wish to load prev network values(yes/no)
   2. Do you wish to log training to file (yes/no)
   3. Do you wish to use selective procedure(yes/no)
4. Once you have entered your answer for the three options the script will begin training the network and provide feedback on its progress(Figure 5)
5. Once finished you will find your trained networks values in the folder Network_Values in a file called “NET-Values.xml”
6. If you said yes to the option to log training you can find this file in the Training_Logs folder with the filename “NET-Learn”followed by the date and time that training began.



```
C:\Users\Eoin\Documents\GitHub\fourth-year-project\SMB-NN\Lua-Scripts>lua52 Train_network_supervised.lua
Loading Values from config file!
Initiating Network!
Do you wish to load prev network values (yes/no) : no
Do you wish to log training to file (yes/no) : yes
Do you wish to use the selective procedure (yes/no) :
```

Figure 4



```
Trainning for exemplars in file:../Pre-Processed_Exemplars/exemplars_2_May-02-21-35-05-Reduced-Inputs_2.dat
 Progress: 8%
```

Figure 5

NOTE: It is important to make sure there is a space between the variable name and the value when modifying configuration values as if there is no space the value will not be read correctly

3.2 Q-Learning

1. In the configuration files set the following values:
    1. VIEW_RADIUS: the radius of the tiles around Mario you wish to show to the agent.
    2. NUM_NUERONS: this is the number of neurons you want to be in you networks hidden layer
    3. C: this is the value which determines the initial starting weights of the network it is recommended to leave this at its default value of 0.1
    4. RATE: This is the learning rate used in the back-propagation algorithm this value should always be between 0.0 and 1.0
    5. DISCOUNT_FACTOR: this is the value for determining wheter the agent should choose immediate rewards over later rewards. This value represents GAMMA in the q-learning equations.
    6. TRAIN_ITERATIONS: in q-learning this is the number of runs of the level you want the agent to perform
    7. RUN_EXPEREINCES: This is the number of experiences you want the agent to experience at each run
    8. EXPEREINCE_REPLAY: this is the number of times you want to replay the experiences that the agent experienced at the end of each run.
    9. TEMPATURE_CEILING: this value is used in the Boltzmann distribution it is recommended to have this value double the total number experiences the agent will experience during training.
    10. NET_TYPE: This value must be either 'Reinforcment' or 'ReinforcmentMul' This determines how the actions are passed into the network. As described in (1.3) It is recommended to use ReinforcmentMul as this shows the best performance for Q-learning as the networks finds it very difficult to separate the inputs from actions
2. Once these values are set and saved load the settings by either reloading the script SMB_Neural_Net.lua script in the lua console or by entering the settings screen and press the reload key.
3. Once loaded press the key on the number pad corresponding to Qlearn on the

number pad from the main UI.

4. This will load the first level and begin training
5. During training the networks operations will be written to a file that can be found in the Training_Logs folder the file name will be “Q_learn” with the date and time that training began.

4 Running you network

Once you have finished training your network you will want to now run it and see how well you trained it and how far it can get in the level.

4.1 Loading your network

1. In the Configuration file edit the following values:
   1. NET_TYPE: set this value to supervised if your trained you network with supervised learning or ReinforcmentMul if you trained it with Q-Learning.
   2. NET_VALUES_FILE: set this value to the location of the networks value file that you want to load to the system.
   3. VIEW_RADIUS: make sure this value is the same as what the network was trained on
   4. NUM_NUERONS: make sure this values is the same as what the network was trained on
2. Reload the configuration file
3. Press the key on the number pad corresponding to the Load Net function in the main UI. You should see a on screen prompt saying the system loaded the network values. There will also be a message in the Lua console saying it was loaded. If you receive and error check that the VIEW_RADIUS and NUM_NUERONS are the same as what the network was trained on or that the file location is correct.

4.2 Exploiting Your Network

Once you have the network loaded into the system press the numpad button corresponding to the Exploit function. You should see the first level being loaded and the network begin to play the level.