Neural Networks to predict game freeze in a two person fighter video game

## 1. Introduction and Summary

Game freeze interrupts game play and hinders enjoyment of the game. Since the causes of game freeze in a game engine are not always known, it is useful to develop and apply machine learning techniques to predict – and therefore avoid – game freeze. Here we report on the development and application of a neural network to predict game freeze during game play from as few as three game variables in a two person fighter video game.

First, the data were downloaded off of the Internet. Next the data were cleaned and parsed into CSV format. Next, we conducted exploratory data analysis (EDA) to isolate the key variables. Using both visualization and quantification, we reduced the features down to a three core variables candidate variables. We then encoded these variables into a format suitable for neural networks and applied the neural nets to the training data. We obtained a 98% accuracy on the available data.

## 2. Methods

There were three steps conducted in this analysis:

1. Downloading and cleaning the data
2. Exploratory analysis
3. Network training and evaluation

2.1 Downloading and cleaning the data The data were downloaded from

[https://github.com/billwillman/UnityMugen/blob/9a30e9d9deb54b48f7121172496b6564816d3340/Project/](https://github.com/billwillman/UnityMugen/blob/9a30e9d9deb54b48f7121172496b6564816d3340/Project/Assets/resources/mugen/char/@Master-Lee/Master-Lee.cns.txt#LL195C24-L195C24)

[Assets/resources/mugen/char/%40Master-Lee/Master-Lee.cns.txt#LL195C24-L195C24](https://github.com/billwillman/UnityMugen/blob/9a30e9d9deb54b48f7121172496b6564816d3340/Project/Assets/resources/mugen/char/@Master-Lee/Master-Lee.cns.txt#LL195C24-L195C24)

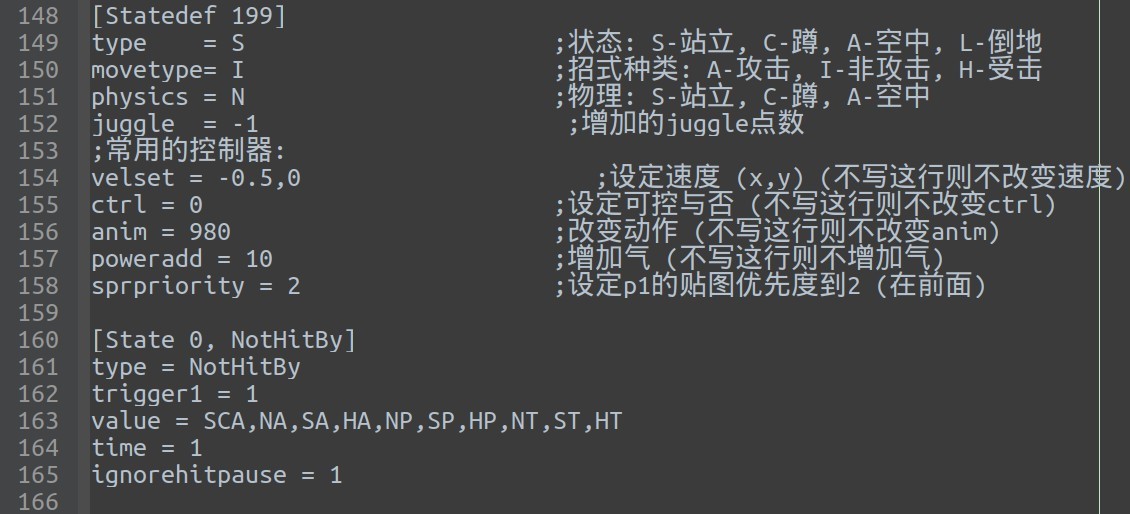


Figure 1. A screenshot of the downloaded data. Note that the comments must be removed, as well as correcting numerous formatting errors and irregularities.

The raw data consist of 8486 lines of scripting output from a two fighter video game engine. This raw data contains numerous foreign language comments, many formatting errors and irregularities. All of these must be corrected before the data can be analyzed. The data must be put into a uniform format.

A script was written in Python to perform these operations. The main purpose of the script was to remove the comments, remove irregular white-space, detect and correct the major formatting errors.

Next, the data were parsed. The parsing happened in two stages: parsing the entry header and parsing the entry body. For example, in Figure 1 above, line 160 is the header, consisting of the identification of the state (state=0) and the action (NotHitBy). The body contains a number of variables with their values. These were collimated and converted to CSV.

2.2 Exploratory analysis

In total, there are 157 different variables present in the data across 906 rows. These variables were reduced to 3 variables by the following procedure:

1. Variables were eliminated that did not appear in freeze rows
2. Variables were inspected numerically in spreadsheet form
3. Variables were quantitatively inspected in Python
4. Variables were plotted in order to identify trends and clusters.

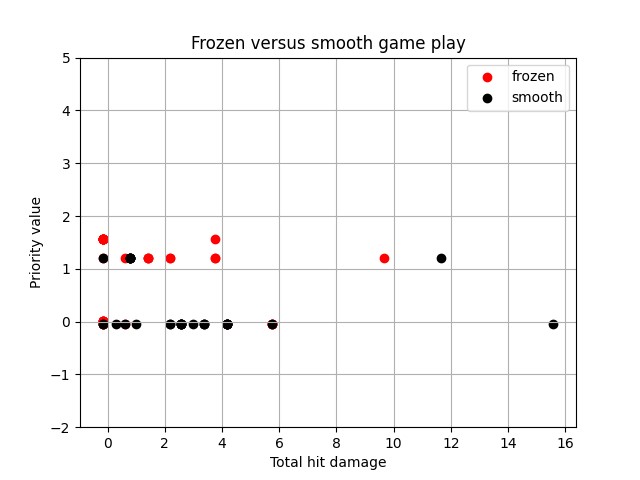


Figure 2. Scatter plot showing frozen game-play (red) alongside smooth game-play (black). The x-axis is the total hit damage and the y-axis is the priority value. We see the beginning of a clear separation with the majority of frozen games near the center of the y-axis and to the left on the x-axis.

2.3 Network training and evaluation

After the exploratory data analysis was complete, three key variables were identified as features for game freeze: damage, sparkxy, and priority. These variables were transformed into neural net variables by splitting and by standard normalization.

The data were separated into 4 groups:

* hit data (31)
* non-hit data (836)
* hit data frozen (39)
* near frozen data (2)

The goal of the classifier is to predict frozen frames from the hit data. That is, there are a total of 70 rows available for training and testing, 31 of which are smooth hit data and 39 of which are frozen hit data. From this small dataset, we randomly selected 50 for training and 20 for testing.

We repeated this test 5 times, obtaining the following accuracies: 0.935, 1.000, 1.000, 0.972,

0.972 for an average accuracy of 98%.

## 3. Conclusion

In conclusion, we have used a combination of data engineering, exploratory data analysis, data visualization and neural networks to clean, feature select, and train a neural network capable of identifying freeze behavior in a two person fighter game with 98% accuracy.