

Collaborators: Henry Leiter

### Write-Up

The training data is sourced from a file named `training.txt`, where each entry records the human move, computer move, and the outcome of the game. Moves are encoded numerically: 0 for Rock, 1 for Paper, and 2 for Scissors. The outcome is represented as 1 for a human win, 0 for a tie, and -1 for a human loss. This data is transformed into one-hot encoded vectors to facilitate effective training of the neural network.

The Recurrent Neural Network (RNN) consists of an input layer with six neurons representing the one-hot encoded human and computer moves, a hidden layer with fifty neurons using the hyperbolic tangent activation function, and an output layer with three neurons corresponding to the possible moves (Rock, Paper, Scissors). The network employs softmax activation in the output layer to generate probability distributions over the possible moves.

Training involves forward propagation through the network to compute predictions, followed by backpropagation to adjust the weights based on the cross-entropy loss function combined with L2 regularization. The model is trained over 500 epochs with a learning rate of 0.01. Early stopping is implemented to halt training if the validation loss does not improve for ten consecutive epochs, preventing overfitting. Additionally, the learning rate decays by 10% every 60 epochs to fine-tune the model's convergence.

After training, the model evaluates its performance by calculating the number of wins, ties, and losses from the training data. Win percentages are computed both excluding ties and including them to provide a comprehensive view of the model's effectiveness. These metrics help in assessing the model's ability to predict moves that lead to favorable outcomes for the human player.

Upon completion of training, the RNN processes the entire input sequence to generate a probability distribution for the computer's next move. A move is then recommended to the human player based on these probabilities, enhancing the strategic decision-making process.

### Reflection

While it wasn't required of me to write any kind of neural network, having pursued ample research in the past on how they work, my partner and I decided to pursue a real one. We began working on the program together on a shared document initially, but upon reaching issues with the backpropagation, we decided to split our separate ways where I started over again with new structures.

Writing this was all over the place, but I truly enjoyed being challenged and having to research so many new things on the topic. Learning about the many different issues that can lead to overfitting forced me to implement new mathematical techniques that I found challenging yet rewarding in the results.