Task 4

Garvit Meena (B20ME033)

5th November 2021

1 Describing and Suggesting a machine learning technique that may be applied in this paper:

In this work, we must offer a model connected to the study that is more effective and efficient than the research paper's "opponent modelling" technique. The "technique" presented in the study article primarily focuses on four components: pre-flop evaluation, hand strength and hand potential, betting strategy, and opponent modeling. Weighting the Enumeration, Computing Initial Weights, and Re-weighting are three steps in the opponent modelling process that work together to give you a sense of all the various hands you can make in successive rounds. This model provides a good notion of the methods that might be utilised against a poker player.When confronted with a world-class player, meanwhile, the player changes his tactics at different stages of the game, which is a drawback of this model. As a result, a more appropriate and realistic model was necessary. As a result, we'll go through this concept in more detail below.

Now, there are a variety of different approaches that may be used with this model, such as pattern recognition, tress, or neural networks. So, in accordance with the rules, we must propose and explain a model that is relevant to the course material and has a reasonable level of effectiveness when compared to the model used in this paper.

As a result, we're using and summarizing "Neural Networks" in the given task to comply with the standards.

We'll talk about how to use neural networks to simulate opponents in Texas Hold'em in this section.We'll then go over the experiments mentioned in this paper.

1.1 About the model:

Many additional contextual elements, including as the number of active players, relative betting position, pot size, and community board card qualities (such as the presence of ush or straight draws), might also influence a player's conduct. Testing and fine-tuning each of these variables would be time-consuming and uninteresting from a scientific standpoint.

Simple rule-based methods are essentially flawed, resulting in a system loaded with faults and biases. We believe that explicit human knowledge should be avoided wherever possible in favour of more computer-based solutions. Almost every important achievement in high-performance gaming systems has historically validated this approach.

Playing poker at a world-class level will require dynamic learning as the match continues, as well as the capacity to react to changing situations. As a result, we've started looking into alternative means of doing these jobs, which could provide far more freedom than the current structure.

One advantage of using a neural network is that many different factors can be provided as input, and the weighing will be performed immediately to maximise the accuracy of the target output.

Because neural networks may be trained with data from specific players as well as data from a balanced set of players, they can be used for both generic and specific opponent modeling. A good generic opponent model must include a wide range of players (weak and strong, loose and tight).

1.2 Describing Model:

An input layer, one or more hidden layers, and an output layer are the layers of nodes that comprise a neural network. In our case, each game state parameter to be studied will have its own node in the input layer. Such parameters include the game stage (flop, turn, or river), the number of players in the pot, and the pot ratio.

Three nodes in the output layer correspond to the possible opponent actions of Fold, Call, and Raise.

When these numbers are added together and averaged, they form a probability triple. The network's output is this.

Every node in a layer is connected to every node in the layer before it and layer after it. Every connection has a specific strength or weight. These weights define how powerful a node's influence on next node is.

By running training sets through the network, the network can be trained. Each training set contains the game's input parameters as well as the opponent's real activity in that situation. Each weight is then adjusted so that the network's output more strongly matches the training set's output. Because it tries to reduce the gap between the desired and actual outcomes, this process is known as backward propagation of errors.

1.3 Determining Accuracy:

Running a test item through the network and comparing the output to the correct result can be used to determine the accuracy of a neural network. The network has made a correct prediction if the output node with the greatest value is the opponent's proper action.

The squared difference between the output and the correct answer is defined as the network error.A useful way to format the accuracy of a network is with

	F	C	R	Freq.
F	13.0	0.3	0.3	13.6%
C	0.0	58.4	3.3	61.8%
R	0.0	10.5	14.1	24.7%
Freq.	13.0%	69.3%	17.7%	85.6%

Figure 1: Confusion Matrix

a confusion matrix:

The proportions predicted by the network for Fold, Call, and Raise are represented in columns F, C, and R. The opponent's real action is shown in rows F, C, and R. As a result, the diagonal contains the moments when the network correctly predicted an opponent's action. The opponent's action distribution is shown in the Freq. column, and the network's predictions are shown in the percentual distribution in the bottom row. The percentage sum of correct predictions in the bottom right cell represents the network's overall accuracy.

A confusion matrix shows the kind of errors that the neural network produces. This information can be used to update the probability triple to account for this sort of error. As a result, we can make a more conservative prediction by updating the probability triple.

Loki's neural network was able to predict the opponent's action much more reliably, with an accuracy of about 81 percent, as compared to the old opponent modeling system, which has a accuracy of 57.3 percent. The results also showed that two important factors, such as the opponents previous action and the previous amount to call, were not taken into account by the old model. These parameters were then included into Loki's calculations and incorporated into its own analysis.

This can be solved by training several different neural networks at the same time, but in different ways. These networks compete to be used, and are selected based on their accuracy in predicting the most recent actions.

As a result, we can conclude that neural networks outperform the approaches described in the paper.