Phi Duong
2/4/2025
CSE5693

Machine Learning Written Assignment 1

(a) 1.2. Pick some learning task not mentioned in this chapter. Describe it informally in a paragraph in English. Now describe it by stating as precisely as possible the task, performance measure, and training experience. Finally, propose a target function to be learned and a target representation. Discuss the main tradeoffs you considered in formulating this learning task.

For this learning task, I will be creating a model that will play tic-tac-toe using a machine learning algorithm. The performance measure will be tested by playing against a computer with different starting moves. Its training experience will be conducted using two different methods: indirect and direct training. A direct training method will be given to the model using a teacher that knows the best moves to play in the game. The machine learning model is trained with certain game states and the probabilistic values of the model winning if played perfectly. For indirect training. The model will play against itself for multiple rounds. The model will update its values using the set of boards states based on the model's move choices and if the result ended in a win, loss, or tie.

The target function will be a linear function consisting of 16 variables with their corresponding weights. 8 of them are going to be the number of player pieces in each row (3 each), column (3 each) and diagonal (2 each) of the board. The other 8 are the same except for the checks for opponent pieces. The Model will use a LMS algorithm to update the weights for the function. Since a single linear function could not represent ideas like 3 in a row. It's better to represent the board in a way to show possible ways to win and how close the player is/ opponent is to win. The trade off is a lot more computations to calculate the estimate training function, but it should benefit over just giving a value per board position.

(b) 1.4. Consider alternative strategies for the Experiment Generator module of Figure 1.2. In particular, consider strategies in which the Experiment Generator suggests new board positions by

- Generating random legal board positions
- Generating a position by picking a board state from the previous game, then applying one of the moves that was not executed
- A strategy of your own design

Discuss tradeoffs among these strategies. Which do you feel would work best if the number of training examples was held constant, given the performance measure of winning the most games at the world championships?

Generating random legal board positions would give it a variety of board states to look into. With limited examples to use, it would be useful at giving all possible board states. The problem with pure randomness is that examples given could vary in usefulness and would not give the model an effective idea of how the game works. In addition, it would have varying amounts of accuracy and consistency due to a lack of control of the example set. Generation through previous games helps the algorithm hone in on reaching better

outcomes in future tests, but it would only improve on current board states. The board states that are completely different from the current one will cause problems for the current model as it was never trained on these edge cases. A better strategy would be to classify the board state into categories and make sure an even amount of each category gets trained on. For example, games would typically have different types of board states as the game progresses. The simplest form of this is the beginning, intermediate, and end state. Training the model equally witch each state should allow the model to gain its variety while increasing its consistency when generating the model.

(c) from your programming assignment:
   i. State the learned weight values without a teacher
   ii. State the learned weight values with a teacher
   iii. Discuss why each of the weight values makes sense or not.

Current weight values for the model without a teacher:
[-4.1, -4.2, -4.2, -4.1, -4.2, -4.2, -4.2, -4.2, -4.2, -4.2, -4.2, -4.2, -4.2, -4.2, -4.2]
Current weight values for the model with a teacher:
[-18.4, 18.1, -27.1, -18.4, 18.1, -27.1, 10.6, 12.3, -8.8, 30.3, -21.1, -8.8, 30.3, -21.1, -13.3, -11.1]
This makes sense because the model without the teacher does not know what move caused the win or loss, so it is hesitant on giving stronger weights to any one variable. With the teacher, the model is given a strong direction on which board state is winning or not.