

Assignment 1:

Answer 1.2 Text Book

Tic Tac game is played on 3X3 board. A game is considered to be won when a player has 3 consecutive strikes (O/X) row wise/column wise or diagonally. There are overall 9 moves available in total for both player and opponent. If by end of 9 moves, no one achieves a win, we called the board state as *draw*. If the opponent achieves 3 strike before the player, it's declared as a *loss* for the player.

Task is anything that refers to performing work. In this example, making quality moves on the board in search of a win is the *task*. We achieve this by selecting a board position among many available legal board positions using Equation 1. The board position that corresponds to maximum value of the equation is our best board position.

Performance Measure is a scale that is used to measure the quality of machine learning algorithm found. In this example, performance is measured against percentage of win that we have achieved using equation 1.

Training Experience is the process of gaining knowledge by performing tasks of different complexity. In this example, machine gains the experience to play the game efficiently by training with different board states. In doing so, it updates the weight values for every move that is made on the board. Here, error is said to be zero if the difference between actual output value and predicted output value is zero. Actual output value is found using equation 1, while predicted output is assumed to be equal to the boards successor state. If there difference is positive, we increase the weight values in a bid to achieve zero error. Else, we decrease the weight values. This is done with the help of below equation:

$$w_i \leftarrow w_i + (\hat{V}(b)_{train} - \hat{V}(b)) * x_i * n$$

Where,

n : learning factor

$\hat{V}(b)_{train}$: actual output

$\hat{V}(b)$: predicted output

Learning factor is used to limit the magnitude of changes to values of the weights.

Target function to be learnt is the way to select the best board state based on the corresponding board moves values.

It is represented by the below equation:

$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6 \quad \cdot \text{Equation 1}$$

Where,

$\hat{V}(b)$ = Target Function

$w_0 - w_6$ = weights

$x_1 - x_6$ = board features

x_1 = number of X on board

x_2 = number of O on board

x_3 = number of 3 X's in a row /column on board

x_4 = number of 3 O's in a row /column on board

x_5 = number of 2 X's in a row /column with subsequent being empty on board

x_6 = number of 2 O's in a row /column with subsequent being empty on board

Trade-offs considered in this task is related to selecting board states. Board states representing the guidance of a teacher and non-teacher is the main trade-off here. If we train the board, using board states given by a teacher, we learn the game efficiently on teacher's terms. We don't explore other possible board scenarios. On the other hand, when we are given a random board state without teacher's guidance, we learn game techniques while playing. Over a period of time the algorithm we design with this approach, would be efficient and would have covered all possible board situations. Also, with teacher as guidance, we always know the end result of our game. But without teacher, we may end up with win/loss/draw though our initial board state is bad. Thus, this is a trial and error way to learn things which improves Performance significantly. Hence, when we come up against an opponent like world champion, we are better off with this approach as we would have explored many situations.

1.4 Text Book

a) Random board positions

If the experiment generator suggests new board positions randomly, it gives us an opportunity to test the efficiency of our algorithm against new board scenarios that we could have missed earlier while training. It also lets us to update our algorithm design for the better. On the other hand, if the generator generates bad quality board position, it doesn't help us to improve our algorithm. In fact, if we use these board positions to train our system, quality of algorithm that we intend to design will be low.

Thus, this method generates plenty of different board positions of ambiguous quality to develop and test our algorithm.

b) Generating a position by picking a board state from previous game, then apply one of the moves that was not executed

If the experiment generator follows this method, we can see the effects of choosing a board position that is better/worse than that experiment generator had chosen previously. If the quality of new board position is good, it helps us to update our weight values of the algorithm for better. Else, training the system against this new board position becomes repetitive without any again.

Thus, this method lets us explore the different board positions which we had refrained from testing our algorithm earlier and make a conclusion regarding our decision.

c) My design

I would make the experiment generator to generate board positions such that they help the algorithm to learn the games rule easily. For example, I will generate board position that would simulate board positions for wins, losses and draws equally. This will allow us to develop accurate algorithm. The trade-off to be considered here is that we can't simulate every game situation possible. As a result, quality of algorithm is good to the level that we have trained it to.

Answer 1c Assignment

Weight values without teacher :

```
w2=-13.448397372565559
w1=-14.853010692273617
w0=-3.578954231030347
w6=-3.79834448849391
w5=-3.8351567984749866
w4=-0.39342359122570614
w3=-2.9215921633144184
```

Weight values with teacher :

```
w2=0.03
w1=0.03
w0=-4.978756648867395
w6=0.03
w5=0.03
w4=0.03
w3=0.03
```

Weight values learned using with teachers help results in better output in terms of performance as we have supplied all the board states to the system with teacher guidance. As a result, percentage of win is higher with teachers t weight values.

While, weight values learnt without help of teacher results in lesser percentage of win comparatively. This is because the board moves which we decide here are done without teacher's help. As a result, we can't guarantee a win at the earliest. Rather, we guarantee over early win by training over system over larger training data set.