Date______ 20__
M T W T F S S

ASSIGNMENT #03

Basil Ali Khan
20K-0477

Question#01:

(1)

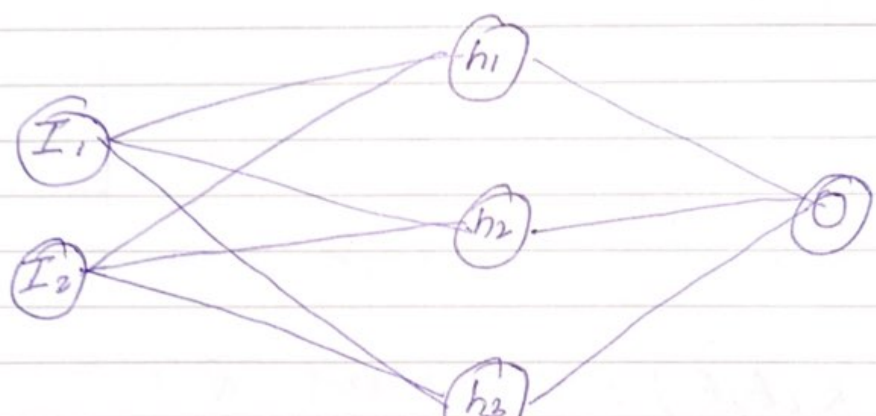
$a_1 + ib_1 = 3+4i$
$a_2 + ib_2 = 5+2i$
$a_3 + ib_3 = 2+i3.2.$

$s_1' \Rightarrow w = [0.2, 0.4, 0.6]$
$s_2' \Rightarrow w = [0.1, 0.9, 0.7]$

$s_1' = (0.2 \times 3 + 0.4 \times 5 + 0.6 \times 2) + (-2).$
$s_1' = \sigma(1.8) = 1/1+e^{-1.8} = 0.8581$

$s_2' = (0.1 \times 3 + 0.9 \times 5 + 0.7 \times 2) + (+2).$
$s_2' = \sigma(8.2) = 1/1+e^{-8.2} = 0.9992.$

(2)



(3)

$$\frac{\partial L}{\partial B^{(1)}} = \frac{\partial L}{\partial Z^{(2)}} \times \frac{\partial Z^{[2]}}{\partial A^{[2]}} \times \frac{\partial A^{(2)}}{\partial Z^{[1]}} * \frac{\partial Z^{[2]}}{\partial B^{[1]}}$$

Back propagation allows a network to maintain and updated its weight and biases in order to minimice the loss function. and propagate error backward from output layer to input

Question #02

$$
R = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array}
\begin{array}{c}
\begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\left[\begin{array}{cccccc}
-1 & -1 & -1 & -1 & 0 & -1 \\
-1 & -1 & -1 & 0 & -1 & 100 \\
-1 & -1 & -1 & 0 & -1 & -1 \\
-1 & 0 & 0 & -1 & 0 & -1 \\
0 & -1 & -1 & 0 & -1 & 100 \\
-1 & 0 & -1 & -1 & 0 & 100
\end{array}\right]
\end{array}
$$

$$
Q = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array}
\begin{array}{c}
\begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\left[\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}
$$

$$
\begin{aligned}
Q(B,F) &= R(B,F) + 0.8 * \max[Q(F,E), Q(E,F), Q(F,B)]. \\
&= 100 + 0.8 * \max[0] \\
&= 100 + 0 \\
&= 100
\end{aligned}
$$

$$
Q = \begin{array}{c c}
 & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} &
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 100 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

The article on Reinforcement Learning (RL) focuses on the approaches used to solve RL problems. The article begins by summarizing the previous article, which introduced basic concepts and terminology in RL and provided a framework for solving RL problems using Markov Decision Processes. The article then proceeds to discuss the solution approaches and categorization of RL algorithms.

The author categorizes RL solutions into two main groups: model-based and model-free approaches. Model-based approaches are used when the internal operation of the environment is known, allowing the exact outcome of each state-action interaction to be determined. On the other hand, model-free approaches treat the environment as a black box, where the internal dynamics are not known, and the agent learns by interacting with the environment.

Another distinction made in the article is between prediction and control problems. Prediction problems involve finding the value function given a policy, while control problems aim to explore the policy space and find the optimal policy. The majority of real-world RL problems fall under the control problem category.

The article then provides a taxonomy of well-known RL algorithms, categorizing them based on their characteristics. The categories include value-based methods, policy-based methods, actor-critic methods, and more. The focus of the article is primarily on model-free control solutions, as they are more relevant to practical RL problems.

Model-based approaches are briefly mentioned, highlighting that they can find a solution analytically without interacting with the environment. In contrast, model-free approaches rely on interacting with the environment to observe its behavior.

The article explains how model-free algorithms observe the environment's behavior by interacting with it one action at a time. The agent takes actions, observes the next state and reward, and repeats this process to acquire experience through trial and error. The trajectory of the agent's interactions becomes the algorithm's training data.

Next, the article introduces the Bellman Equation, which is the foundation for all RL algorithms. The Bellman Equation relates the return, value, and state-action value in RL problems. It explains how the return from a state can be decomposed into the immediate reward and the discounted return from the next state. This recursive relationship allows for the computation of optimal values and policies.

The article concludes by mentioning that the subsequent articles in the series will delve into specific RL algorithms, such as Q-learning, Deep Q Networks, Policy Gradient, and Actor-Critic methods.

Overall, the article provides a comprehensive overview of the solution approaches in RL, highlighting the categorization of algorithms and the importance of the Bellman Equation in RL problem solving.