# RL 2021 Exam

I am relatively confident of Q1 and Q2.

Q3 is blank atm so please feel free to add a solution :)

Q1

a) B,C ✓

b) A, B,D (when u aggregating all values I will get a lower variance) - I am not sure about the additional bias bit, because we are considering all possible actions, then surely the bias should be less? (ps: for whoever is wondering the second update rule is Expected Sarsa)

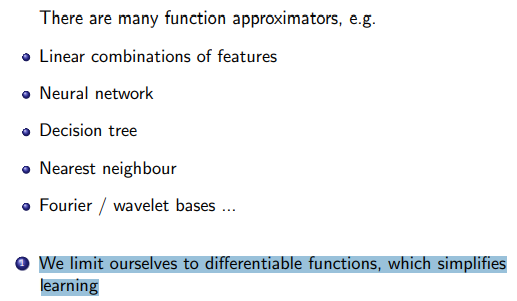
Graphical user interface, text, application

Description automatically generated

c) A,D (Could be B and C, not sure thou. The behaviour policy is based on Q1 and Q2 but both Q1 and Q2 is based on the probability)

d) A,B (might not be A because V(s,w) is not differentiable ) - I think C should not be selected, you can indeed have basis functions in linear methods

Not sure if the below means we can technically have non differentiable functions:



Graphical user interface

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

e) A,D,E

f) E

g) ii, Vi

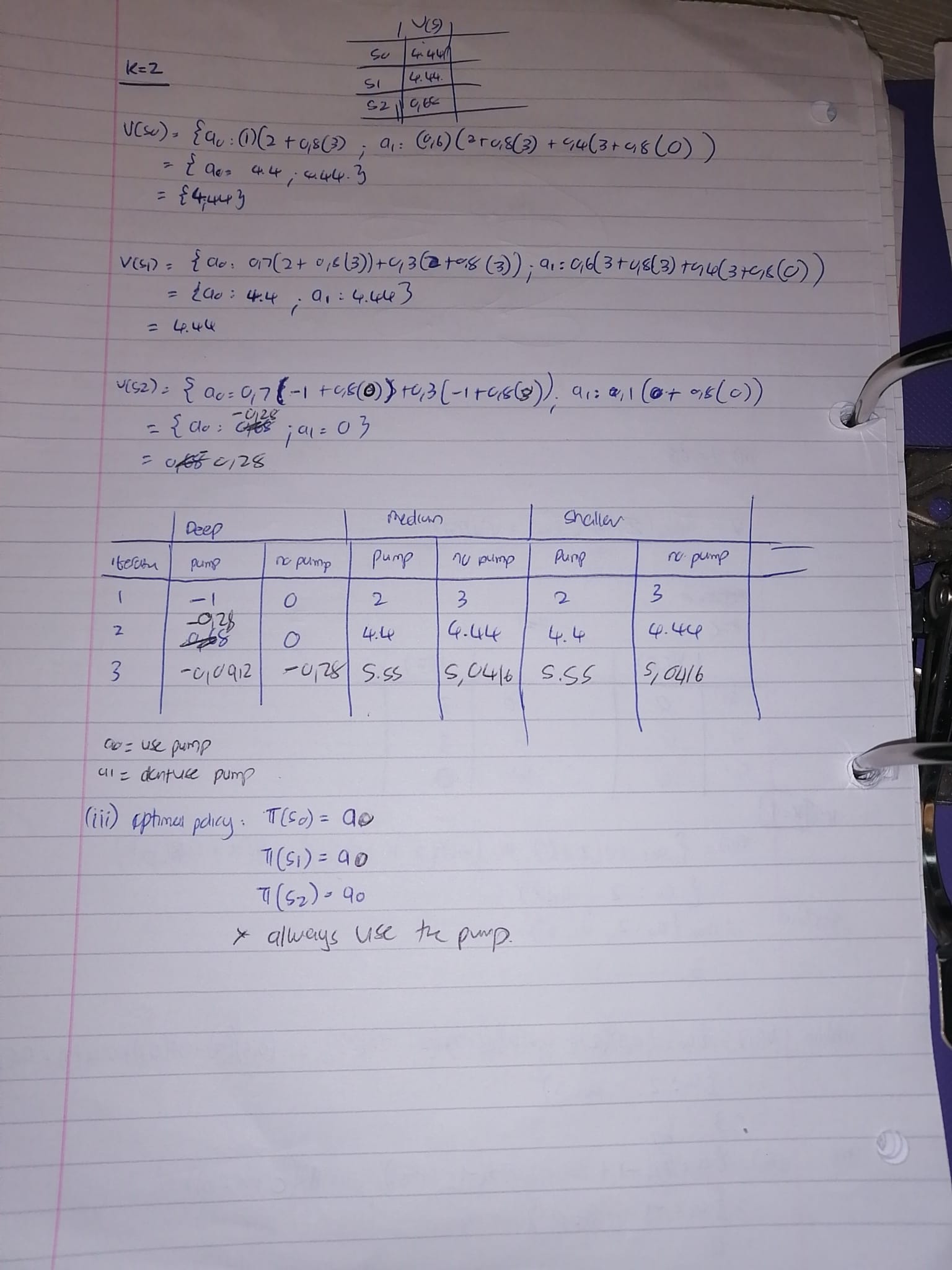
A piece of paper with writing on it

Description automatically generated

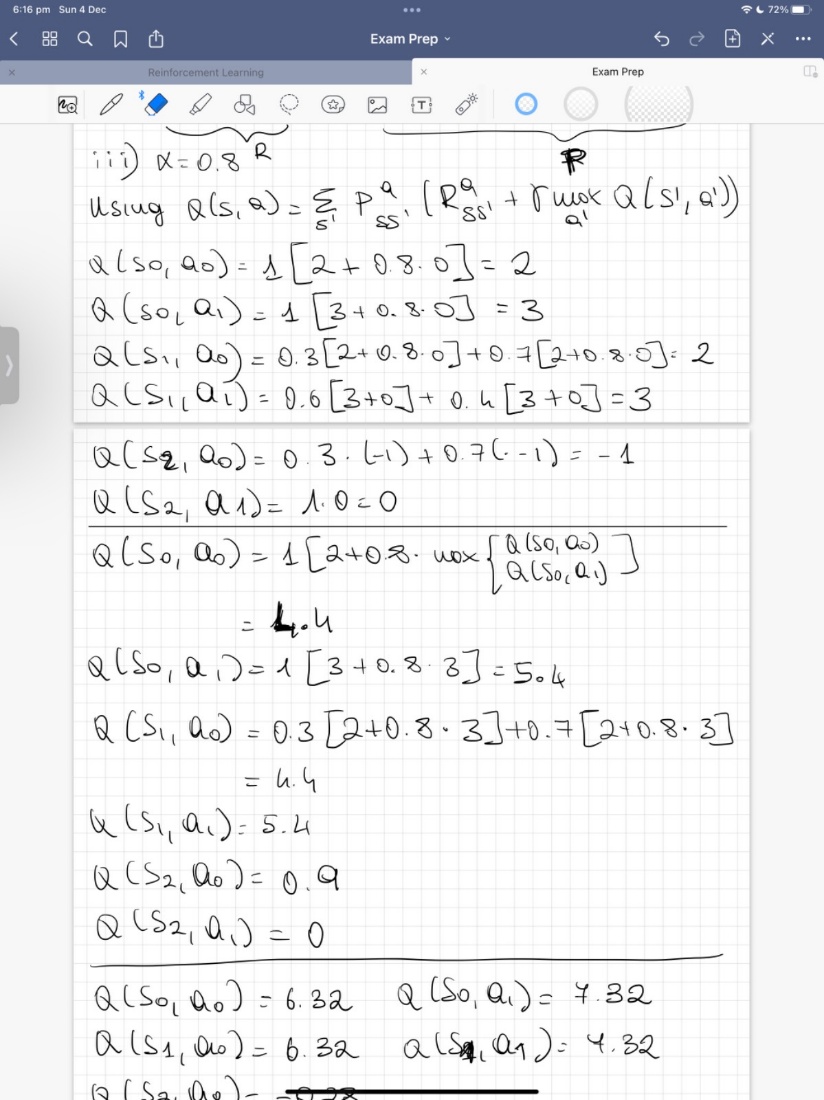
Q2b

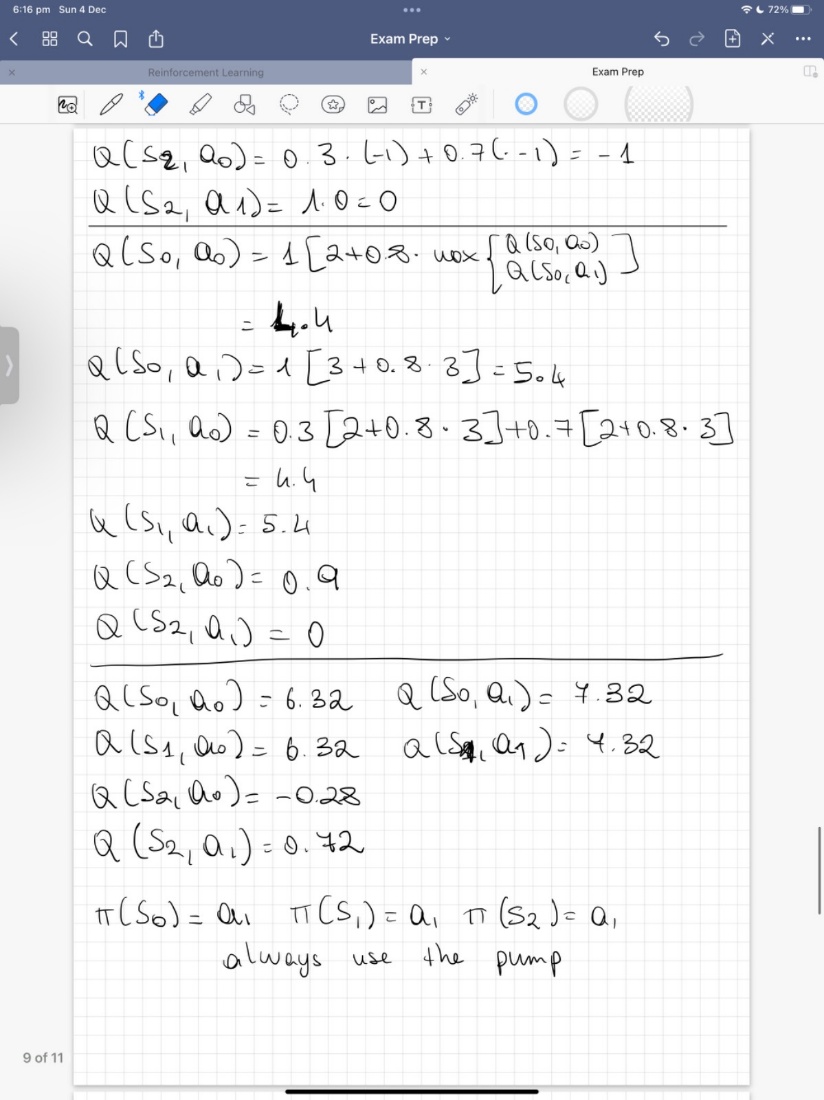
Diagram

Description automatically generated

Text, letter

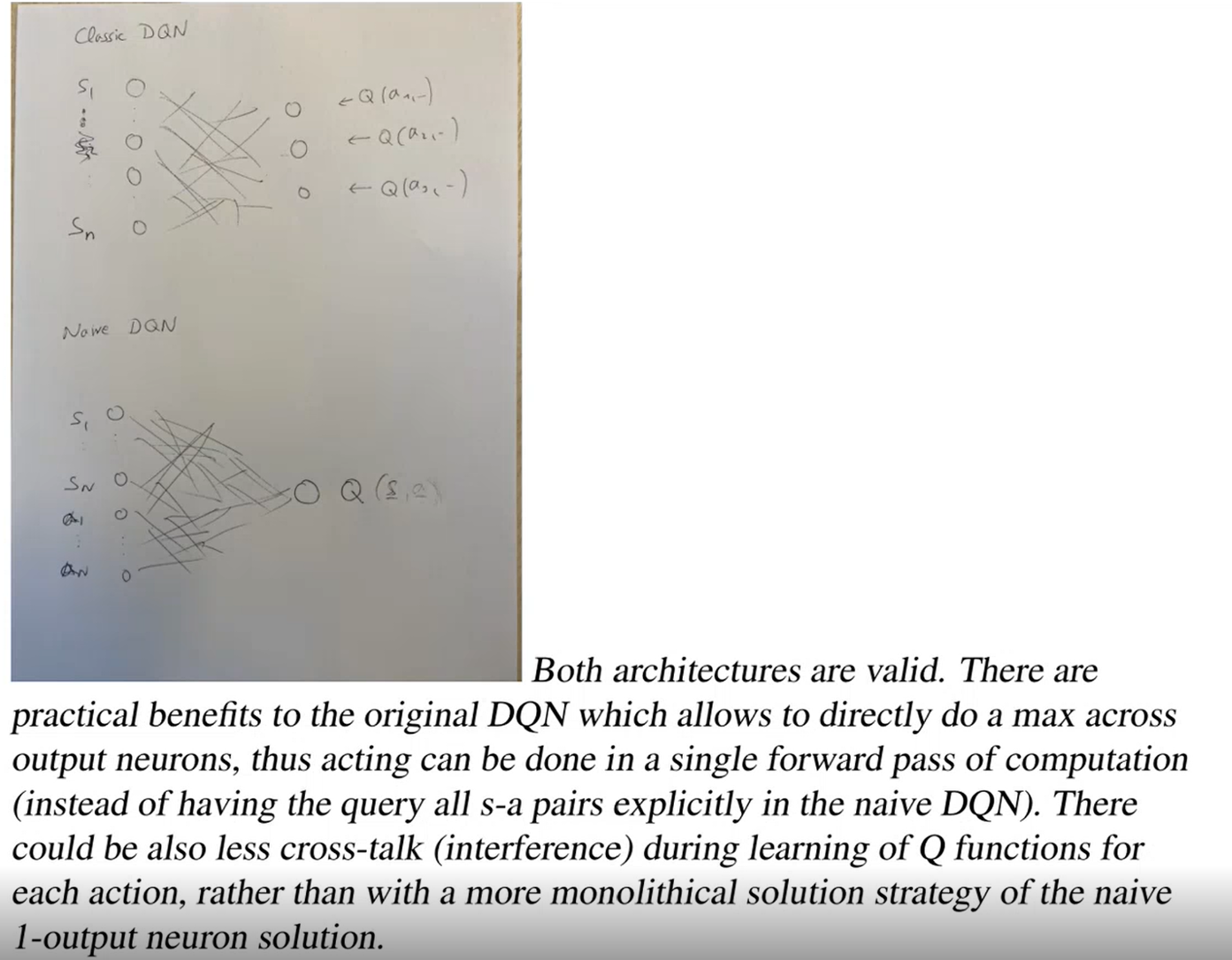
Description automatically generatedThe questions asks for the action-value function, not value function, therefore I did something different (also you forgot to do iteration 3).

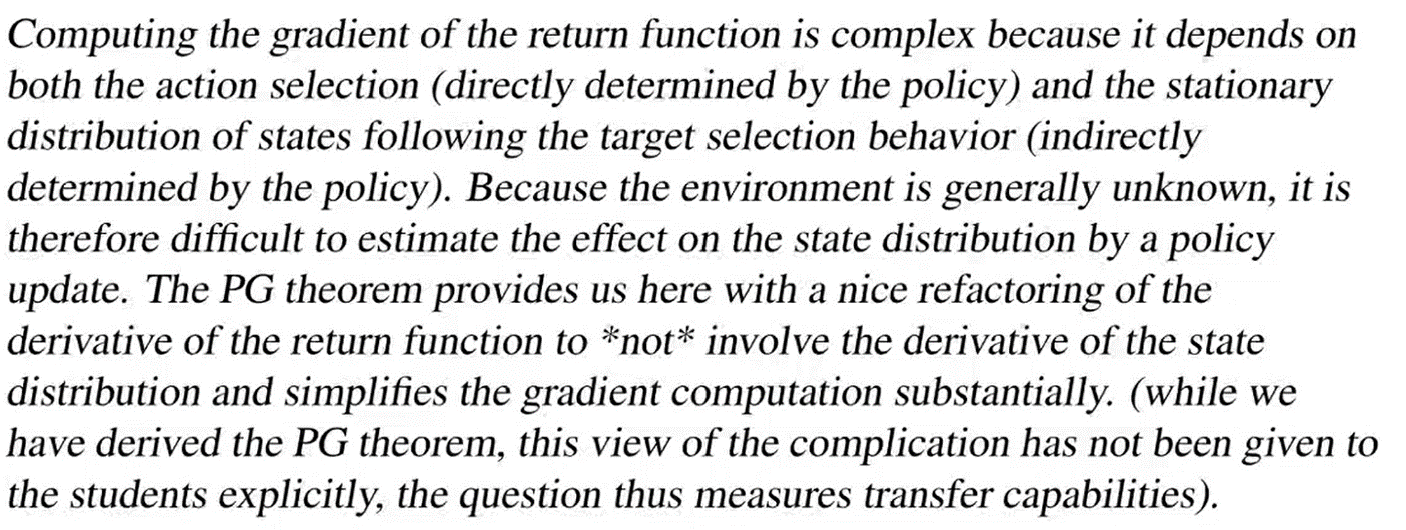
 x

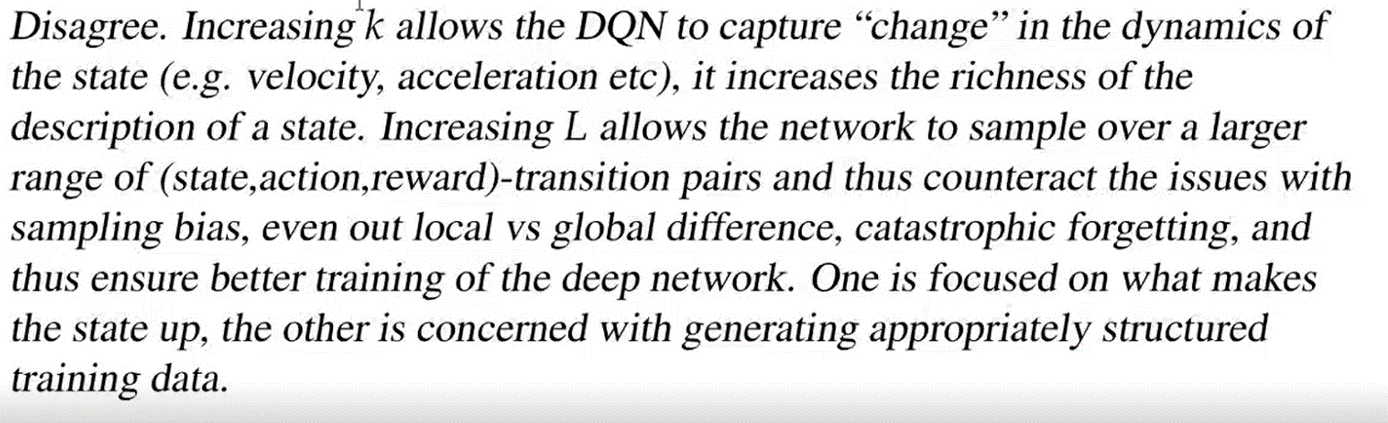


Question 3:

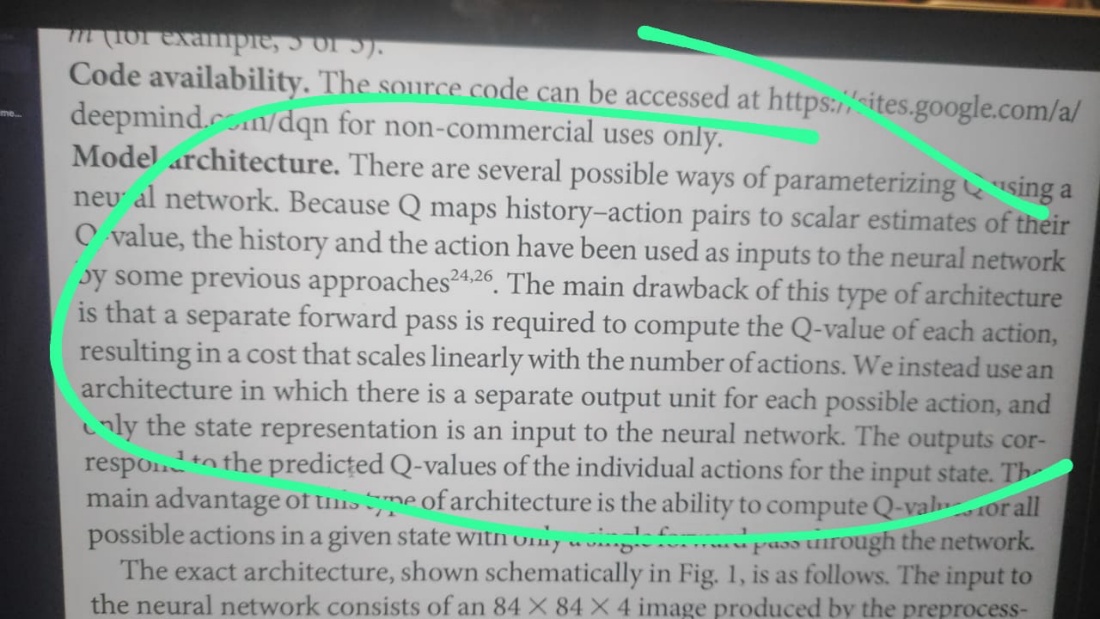
**Model Answers:**







Part 1



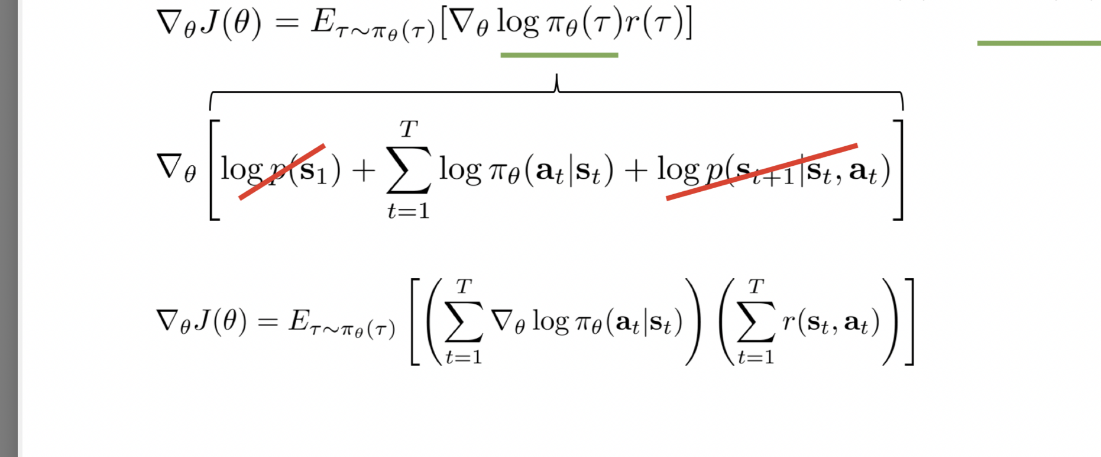
Snapshot from DQN paper

Another advantage might be that the original using Original DQN it is possible to see how the model is grouping the states by observing the high dimension representations learned by the DQN network using techniques like t-SNE (referenced in the DQN paper). With the Naïve model as action is also part of the input data, I don’t think this type of inference would be possible to make.

PS : Not sure about this reason though. (Professor confirmed it so it should be fine.)

Part 2

Policy Gradient theorem helps us to get rid of transition probability terms, thus we can just do simple simulations to calculate gradients of the performance measure



This part

Part 3

k means that we are skipping k frames and then stack them together, while in L we just keep on adding (s,a,s’,r) to our memory buffer. I think both have completely different purpose, by introducing k we reduce the computations required to compute an action also stacking multiple frames gives more information as compared to just a single frame. While memory buffer L stores the runs that our agent make so that later on we can sample randomly from the buffer, doing so helps us to get rid of correlations between input samples. Increasing or decreasing doesn’t seems to be fulfilling the same function as a buffer.