THE UNIVERSITY OF HONG KONG

FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE

DASC7606 Deep Learning
(Subclass D)

Date: (May, 8th, 2023) **Time: 6.30pm – 8.30pm**

There are 3 questions in total. The first question gives 20/100 points, the second one 30/100, while the third one 50/100.

Answer ALL questions. They are all COMPULSORY.

Write your answers on the exam paper in the corresponding space after the question.

Only approved calculators as announced by the Examinations Secretary can be used in this examination. It is candidates' responsibility to ensure that their calculator operates satisfactorily, and candidates must record the name and type of the calculator used on the front page of the examination script.

Question 1 (Gradient descent, 20/100). Consider the following function $y=x^2$ to which we would like to use gradient descent and gradient descent with momentum in order to compute a local minimum. We recall the equations for the latter one as follows:

$v_{t+1} = \rho v_t + \nabla y(x_t)$, $x_{t+1} = x_t - \alpha v_{t+1}$

where ρ is the friction factor and α is the learning rate.

Starting from $x_0=-1$ do the following:

a) (4/20) Report the value of the function at the end of the third step (i.e. x_3^2) of the gradient descent algorithm when the learning rate $\alpha=1$.

---

b) (4/20) Report the value of the function at the end of the third step of the gradient descent algorithm when the learning rate $\alpha=1/2$.

---

c) (4/20) Report the value of the function at the end of the third step of the gradient descent algorithm when the learning rate $\alpha=1/4$.

---

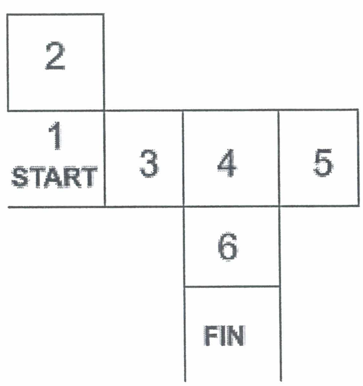
d) (4/20) Report the value of the function at the end of the third step of the gradient descent algorithm *with momentum* when the learning rate $\alpha=1/4$, $v_0=0$, $\rho=1$.

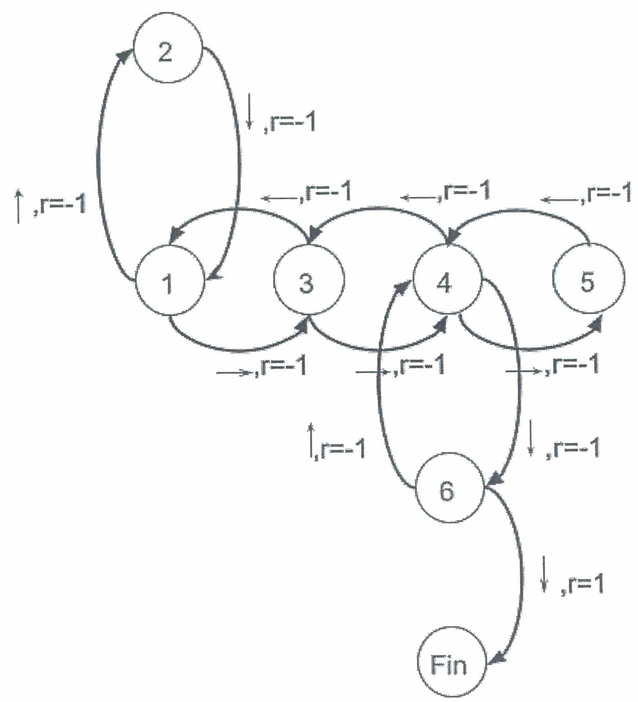
---

e) (4/20) Mention one variant of the gradient descent algorithm (among the ones presented in our course) which aims at alleviating the main issue in d).

Question 2 (Reinforcement Learning, 30/100). The Markov Decision Process of a robot exploring an environment is given as follows:

- $S = \{1,2,3,4,5,6,Fin\}$
- $A = \{\leftarrow,\rightarrow,\uparrow,\downarrow\}$
- The robot starts in state 1.
- The actions are deterministic, i.e., for every state and action the robot moves to the corresponding state receiving the corresponding reward r, as follows





Let q*(s,a) be the max expected cumulative reward starting from state s and performing action a following an optimal policy. We recall the Bellman equation:

$$q^*(s,a)=\max_{\pi} \; E_{s'\sim P(\cdot|s,a)} \left[r + \gamma \max_{a'} q^*(s',a') \mid s,a \right]$$

a) (10/100) What is the main goal of the robot?

b) (10/100) What is an optimal policy?

---

c) (10/100) Run the *value iteration* algorithm with $\gamma=1/2$ until an optimal policy can be inferred from the $q(s,a)$ values. What is the value of $q(1,\rightarrow)$ and $q(1,\uparrow)$?

Question 3 (RNN and Language Models, 50/100). Consider an RNN where x_t , y_t , s_t ,denote the input, the output and the state of the RNN at step *t* respectively. Recall that given matrices W,U,V we have that:

$s_t = g(Wx_t+Us_{t-1})$ and $y_t = g(Vs_t)$,

at every step t>0, where we are ignoring the biases, for simplicity. We consider a language model (LM) where the input x_t is a character in the English alphabet and y_t is a probability vector over a subset of those characters. Recall that at step *t,* a character is sampled from y_t and then fed in input to the RNN at step *t+1*. For this question, we consider only the following characters with the following ordering (in alphabetical order except <start> and <stop>):

<start>, 'd', 'h', 'l', 'm', 'o', 'r', 't', 'w', 'y', <stop>

with <start>,<stop> being special characters denoting the beginning and the end of the sequence, respectively. Your task is to provide the values for W,U,V so that our RNN will produce one of the following three strings chosen uniformly at random (that is, each with probability 1/3): "ylhd", "todlr", "tomlwd" which stand for "yesterday I have dived", "today I rest", "tomorrow I will dive", respectively. We shall denote with a_1,a_2,a_3, the three strings, respectively.

Activation function: We use pointwise ReLu as activation function, i.e. $g(x_1,...,x_d) = (\max(0,x_1),...,\max(0,x_d))$.

x_t vector: We use the one-hot encoding representation for the 11 input characters, using the ordering specified above. E.g. x_t='10000000000' for <start>, x_t='01000000000' for 'd', x_t='00100000000' for 'h', x_t='00000000001' for <stop>...

s_t vector: We let s_t be a 13-dimensional binary vector $s_t^1,s_t^2,...,s_t^{13}$ s_t^j in {0,1}:

- s_t^1,s_t^2,s_t^3 denote which sequence the LM is producing. In particular, s_t^i=1 if the LM 'has decided' to produce the sequence a_i , i in {1,2,3}. Initially, $s_t^1 = s_t^2 = s_t^3 = 0$, however, when processing 'l' exactly one of the s_t^i is equal to 1.
- s_t^4 = 1 if and only if the previous character is the special character <start>.
- $s_t^5,...,s_t^{13}$ represent the previous character in the sequence. In particular, s_t^5=1 if and only if the previous character is 'd', s_t^6=1 if and only if the previous character is 'h' and so on using the ordering specified above with s_t^{13} =1 if the previous character is 'y' (no need to retain any info about <stop>).

y_t vector: probability distribution over ‘d’, ‘h’, ‘l’, ‘m’, ‘o’, ‘r’, ‘t’, ‘w’, ‘y’, <stop>, stored in that order, that is y_t^1 is the probability that ‘d’ is generated next, y_t^2 the probability that ‘h’ is generated next, etc.

Questions:

a) (7/100) What is the size of W,U and V?

---

b) (7/100) Provide the 2nd row of W and U. (Hint: they correspond to a_2).

---

c) (7/100) Provide the first row of V (Hint: this corresponds to ‘d’).

---

d) (7/100) Provide the row of V corresponding to ’h’,

---

e) (7/100) Provide the row of V corresponding to ’t’

---

f) (7/100) Provide the row of V corresponding to ’r’

---

g) (8/100) Provide the row of V corresponding to ‘<stop>’

– END of PAPER –