Practice Final Examination – Solution, 2024
Exam Date:
Duration: 2 hours and 30 minutes (150 minutes)
Course: ECE421H1 F – Introduction to Machine Learning
Examiner: B. Frey, E. Meskar
Exam type: A (*i.e.*, "closed book" exam. No aids are permitted.)
Calculator type: 2 (*i.e.*, non-programmable electronic calculator is allowed)

DO NOT TURN THIS PAGE UNTIL YOU ARE TOLD TO DO SO

- There are **7 questions** and **17 pages** in this exam, including this one. When you receive the signal to start, please make sure that your copy of the examination is complete.
- Answer each question directly on the examination paper, in the space provided. Please note that the provided space **does not** necessarily reflect the expected length of your answer.
- This exam includes one extra page, *i.e.* page 15, for your scratch notes or in case you need extra space for any question, which must be submitted. **Do not remove any page**, including the scratch note page or the formula sheet page from your exam booklet.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total
Max	10	10	12	10	12	11	10	75
Score								

Q1 **[10 pts]** In this question, you will train a regularized linear regression model with an ℓ_1 regularization penalty. We are given the following training dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$, where $x_1 = -1$, $x_2 = 0$, $x_3 = 0$, with associated labels $y_1 = 0$, $y_2 = -2$, and $y_3 = 4$. We wish to fit a linear model $\hat{y} = w_0 + w_1 x$ by minimizing the regularized loss $J(\underline{w}) = \frac{1}{2}\left(\sum_{i=1}^{3}(y_i - \hat{y}_i)^2\right) + \|\underline{w}\|_1$, where $\underline{w} = (w_0, w_1)$ and $\|\underline{w}\|_1 = |w_0| + |w_1|$. Assume weight $\underline{w}$ is initially $(-1, 1)$. Specify the update rule of full-batch gradient descent for w_0 and w_1 and find the updated weight vector after one iteration of full-batch gradient descent with learning rate $\eta = 1$.

Answer. Observe that

$$\frac{\partial J(\underline{w})}{\partial w_0} = \begin{cases} -\sum_{i=1}^{3}(y_i - w_0 - w_1 x_i) + 1 & \text{if } w_0 > 0, \\ -\sum_{i=1}^{3}(y_i - w_0 - w_1 x_i) - 1 & \text{if } w_0 < 0, \end{cases}$$

$$\frac{\partial J(\underline{w})}{\partial w_1} = \begin{cases} -\sum_{i=1}^{3}(y_i - w_0 - w_1 x_i)x_i + 1 & \text{if } w_1 > 0, \\ -\sum_{i=1}^{3}(y_i - w_0 - w_1 x_i)x_i - 1 & \text{if } w_1 < 0. \end{cases}$$

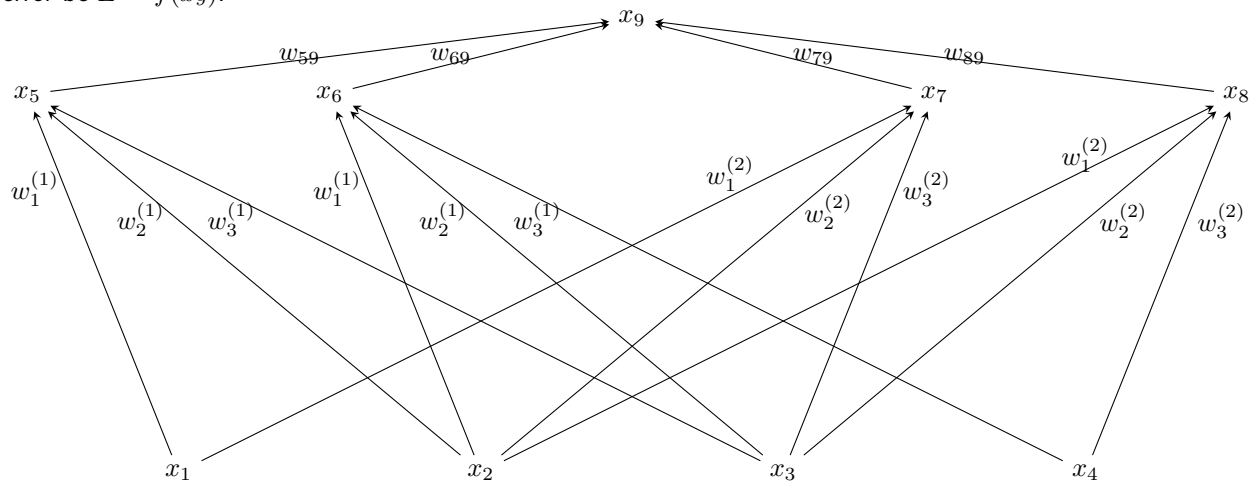
Thus, at $\underline{w} = (-1, 1)$,

$$\frac{\partial J(\underline{w})}{\partial w_0} = -\sum_{i=1}^{3}(y_i - w_0 - w_1 x_i) - 1 = -(0 + 1 + 1) - (-2 + 1 + 0) - (4 + 1 + 0) - 1 = -7,$$

$$\frac{\partial J(\underline{w})}{\partial w_1} = -\sum_{i=1}^{3}(y_i - w_0 - w_1 x_i)x_i + 1 = -(0 + 1 + 1)(-1) - (-2 + 1 + 0)0 - (4 + 1 + 0)0 + 1 = 3.$$

Thus, $\nabla_{\underline{w}} J((-1, 1)) = (-7, 3)$. Therefore, $\underline{w} \leftarrow (-1, 1) - \eta(-7, 3) = (6, -2)$.

Q2 **[10 pts]** Consider the following convolutional layer for a 1-D input sequence of length 4 that has 2 filters of width 3. We denote the weights of the first filter by $w_1^{(1)}, w_2^{(1)}, w_3^{(1)}$, and the weights of the second filter by $w_1^{(2)}, w_2^{(2)}, w_3^{(2)}$. The inputs are denoted by $x_1, \ldots, x_4$, and the output is denoted by x_9. Let the error be $E = f(x_9)$.



Assume that the activation function of the convolutional layer is $g(\cdot)$ and the output layer has linear activation function. Therefore, the relationship between x's are as follows:

$$
\begin{aligned}
x_5 =& w_1^{(1)} x_1 + w_2^{(1)} x_2 + w_3^{(1)} x_3, \qquad x_6 = w_1^{(1)} x_2 + w_2^{(1)} x_3 + w_3^{(1)} x_4, \\
x_7 =& w_1^{(2)} x_1 + w_2^{(2)} x_2 + w_3^{(2)} x_3, \qquad x_8 = w_2^{(2)} x_2 + w_2^{(2)} x_3 + w_3^{(2)} x_4, \\
x_9 =& w_{59} g(x_5) + w_{69} g(x_6) + w_{79} g(x_7) + w_{89} g(x_8).
\end{aligned}
$$

Find $\frac{\partial E}{\partial w_1^{(1)}}$ in terms of x, $g'(\cdot)$, $f'(\cdot)$,and w.

Answer.

$$
\begin{aligned}
\frac{\partial E}{\partial w_1^{(1)}} &= f'(x_9) \left(w_{59} g'(x_5) \frac{\partial x_5}{\partial w_1^{(1)}} + w_{69} g'(x_6) \frac{\partial x_6}{\partial w_1^{(1)}} + w_{79} g'(x_7) \frac{\partial x_7}{\partial w_1^{(1)}} + w_{89} g'(x_8) \frac{\partial x_8}{\partial w_1^{(1)}} \right) \\
&= f'(x_9) \left(w_{59} g'(x_5) \frac{\partial x_5}{\partial w_1^{(1)}} + w_{69} g'(x_6) \frac{\partial x_6}{\partial w_1^{(1)}} \right) \\
&= f'(x_9) \left(w_{59} g'(x_5) \frac{\partial (w_1^{(1)} x_1 + w_2^{(1)} x_2 + w_3^{(1)} x_3)}{\partial w_1^{(1)}} + w_{69} g'(x_6) \frac{\partial (w_1^{(1)} x_2 + w_2^{(1)} x_3 + w_3^{(1)} x_4)}{\partial w_1^{(1)}} \right) \\
&= f'(x_9) \left(w_{59} g'(x_5) x_1 + w_{69} g'(x_6) x_2 \right).
\end{aligned}
$$

Q3 **Deep Learning in Practice – [12 pts]** In class, we discussed various techniques, including weight initialization, weight penalties, Early stopping, Bagging, and dropout, to avoid overfitting. In a few sentences describe these five techniques and explain how each one of them helps with avoiding overfitting.

Answer. See week 8 notes.

Q4 [10 pts] Sequence models.

4.a [3 pts] Convolutional neural networks (CNN).

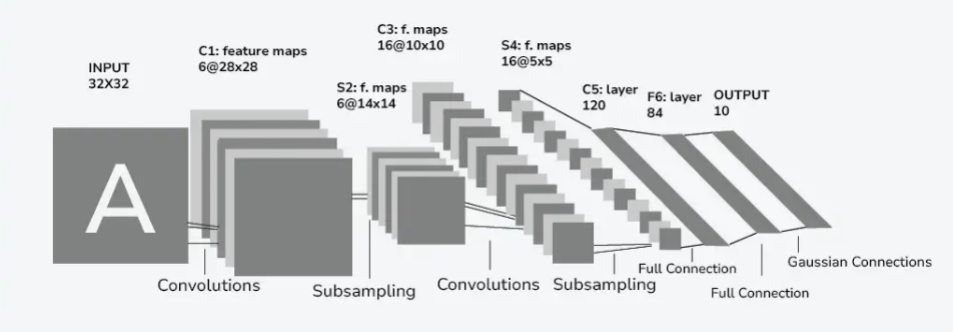
4.a.i [1 pts] What is the primary purpose of pooling (also known as downsampling) in a CNN? (select one):

- ☐ to increase the number of parameters in the network.
- ☐ to reduce the spatial dimensions of the feature map while preserving important information. **(correct answer)**
- ☐ to apply non-linear transformations to the feature map.
- ☐ to convolve the input image with multiple filters.

4.a.ii [1 pts] Consider a convolutional layer for the 2 by 3 input $\begin{bmatrix} -1 & 1 & 3 \\ 5 & 7 & 9 \end{bmatrix}$. The convolutional layer has one filter. Assuming that the output of this layer is $\begin{bmatrix} 0 & 4 \\ 12 & 16 \end{bmatrix}$, the filter size is (select one):

- ☐ 1 by 2 **(correct answer)**
- ☐ 2 by 1
- ☐ 2 by 2
- ☐ the filter size cannot be specified

4.a.iii [1 pts] Here is the historical LeNet Convolutional Neural Network architecture of Yann LeCun et al. for digit classification that we've discussed in class. Here, the INPUT layer takes in 32x32 image, and the OUTPUT layer produces 10 outputs. The notation 6@28x28 means 6 matrices of size 28x28. Assuming that there aren't any biases, how many independent parameters (i.e. weight) are in layer C3?



- ☐ 5×5
- ☐ $5 \times 5 \times 6$
- ☐ $5 \times 5 \times 6 \times 16$ **(correct answer)**
- ☐ $10 \times 10 \times 6 \times 16$

4.b [3 pts] Recursive neural networks. Assume the input is $x_1, x_2, \ldots$ and the output is $y_1, y_2, \ldots$

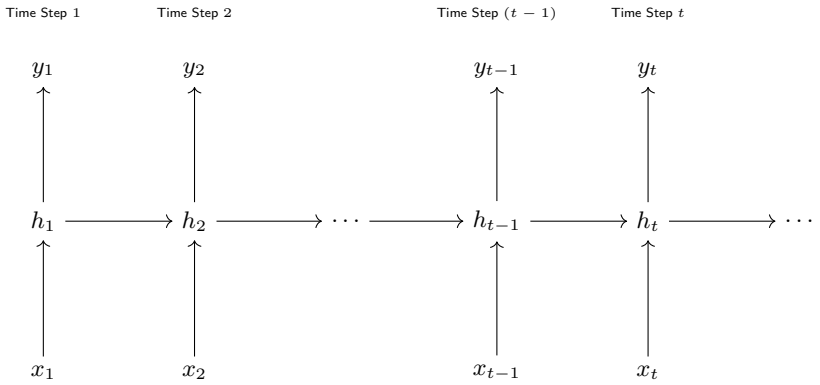
4.b.i [1 pts] What is the purpose of the hidden state in an RNN? (select one):

- ☐ to store the information from the previous time step **(correct answer)**
- ☐ to store the model's parameters during training
- ☐ to normalize the input data at each time step
- ☐ to generate random noise for improving generalization

4.b.ii [1 pts] What is the purpose of the bidirectional RNN architecture? (select one):

- ☐ to capture dependencies by processing the sequence in both forward and backward directions **(correct answer)**
- ☐ to reduce the computational complexity of the network
- ☐ to ensure the network uses only the most recent inputs for predictions
- ☐ to process sequences more quickly by skipping alternate time steps

4.b.iii [1 pts] You are training this RNN language model.



At the i^{th} time step, what is the RNN doing? (select one):

- ☐ estimating $\mathbb{P}(y_1, y_2, \ldots, y_{t-1})$
- ☐ estimating $\mathbb{P}(y_t)$
- ☐ estimating $\mathbb{P}(y_t \mid y_1, y_2, \ldots, y_{t-1})$ **(correct answer)**
- ☐ estimating $\mathbb{P}(y_t \mid y_1, y_2, \ldots, y_t)$

4.c [4 pts] Transformers for language modeling. In the following, $\underline{x}_i$ denotes the i^{th} vector-valued token, and $\underline{q}_i, \underline{k}_i$, and $\underline{v}_i$ denotes its query, key, and value vector, respectively.

4.c.i [1 pts] Indicate whether the following statement is true or false: Suppose you learn a word embedding for a vocabulary of 10000 words. Then the embedding vectors should be 10000 dimensional, so as to capture the full range of variation and meaning in those words.

☐ TRUE ☐ FALSE **(correct answer)**

4.c.ii [1 pts] The similarity $\alpha_{i,j}$ between query i and key j is (select one):

☐ $\alpha_{i,j} = \exp\left(\underline{q}_i^\top \underline{k}_j / \sqrt{d_k}\right)$

☐ $\alpha_{i,j} = \exp\left(\underline{q}_i^\top \underline{k}_j / \sqrt{d_k}\right) / \exp\left(\underline{q}_j^\top \underline{k}_i / \sqrt{d_k}\right)$

☐ $\alpha_{i,j} = \exp\left(\underline{q}_i^\top \underline{k}_j / \sqrt{d_k}\right) / \sum_t \exp\left(\underline{q}_i^\top \underline{k}_t / \sqrt{d_k}\right)$ **(correct answer)**

☐ $\alpha_{i,j} = \exp\left(\underline{q}_i^\top \underline{k}_j / \sqrt{d_k}\right) / \sum_t \exp\left(\underline{q}_t^\top \underline{k}_j / \sqrt{d_k}\right)$

4.c.iii [1 pts] The output of the attention layer for token i is (select one):

☐ $\underline{a}_i = \sum_j \alpha_{i,j}\ \underline{q}_i$

☐ $\underline{a}_i = \sum_j \alpha_{i,j}\ \underline{k}_i$

☐ $\underline{a}_i = \sum_j \alpha_{i,j}\ \underline{v}_j$ **(correct answer)**

☐ $\underline{a}_i = \sum_j \alpha_{i,j}\ \underline{x}_j$

4.c.iv [1 pts] Which one of the following statements is true about multi-headed attention? (select one):

☐ multi-head attention allows the model to focus on different parts of the input sequence simultaneously by using multiple attention heads. **(correct answer)**

☐ in multi-head attention, the number of heads directly determines the dimensionality of the output vector of the attention mechanism.

☐ multi-head attention reduces the number of parameters in the transformer model by using shared weights across all attention heads.

☐ none of the above

Q5 [12 pts] Guest Lecture Questions

5.a [1.5 pts] Dr. Celaj described the BigRNA model of RNA biology. If D is a gene's DNA sequence and R is the gene's RNA expression, which one of the following does BigRNA model? (select one)

- ☐ $\mathbb{P}(D, R)$
- ☐ $\mathbb{P}(R \mid D)$ **(correct answer)**
- ☐ $\mathbb{P}(D \mid R)$
- ☐ None of the above

5.b [1.5 pts] One example of the kinds of tokens that BigRNA pays attention to include (select one)

- ☐ Gene function
- ☐ Splice site **(correct answer)**
- ☐ Protein
- ☐ RNA molecule

5.c [1.5 pts] Which of the following is an application of BigRNA that Dr. Celaj described (select one):

- ☐ Designing an oligonucleotide therapeutic **(correct answer)**
- ☐ Predicting which disease a gene is associated with
- ☐ Mapping each gene to a latent space using single cell data
- ☐ Predicting the protein structure of a gene

5.d [1.5 pts] Dr. Gandhi described foundation models of single-cell data that can be used for multiple purposes. Which purpose did he NOT describe (select one):

- ☐ Perturbation effect prediction
- ☐ Evolutionary biology prediction **(correct answer)**
- ☐ Gene function prediction
- ☐ Gene regulatory network (GRN) inference

5.e [1.5 pts] In Dr. Gandhi's setup for foundation models of single-cell data, what corresponds to tokens and sentences? (select one)

- ☐ DNA as tokens, RNA as sentences
- ☐ Cells as tokens, DNA as sentences
- ☐ RNA as tokens, genes as sentences
- ☐ Genes as tokens, cells as sentences **(correct answer)**

5.f [1.5 pts] Dr. Gandhi described the application of single cell foundation models to which medical application? (select one)

- ☐ Alzheimer's disease
- ☐ Immunological disorders
- ☐ HIV/AIDs
- ☐ Cancer **(correct answer)**

5.g [1 pts] Dr. Wang described that modern uses of foundation models/LLMs in industry require "A-B-C-D". What does each letter stand for? (select one)

- ☐ Algorithm, Bidirectional RNN, Compute, Data
- ☐ Application, Business, Compute, Data
- ☐ Application, Bidirectional RNN, Compute, Data
- ☐ Algorithm, Business, Compute, Data **(correct answer)**

5.h **[1 pts]** What are the two steps of sc-GPT training that Dr. Wang described? (select one)

- ☐ Pretraining and finetuning **(correct answer)**
- ☐ Tokenization and validation
- ☐ Validation and testing
- ☐ Tokenization and testing

5.i **[1 pts]** Roughly how much hardware (GPU) resource did Dr. Wang say was needed to train sc-GPT? (select one)

- ☐ 1 GPU for 1 hour
- ☐ 5 GPUs for 1 day
- ☐ 20 GPUs for 2 weeks **(correct answer)**
- ☐ 100 GPUs for 3 months

Q6 **[MDP – 11 points]** There's a potential for a strike from Canada Post Worker Union. The union's negotiation team seeks assistance from ECE421 students to analyze the negotiation process with their employers, treating it as an MDP.

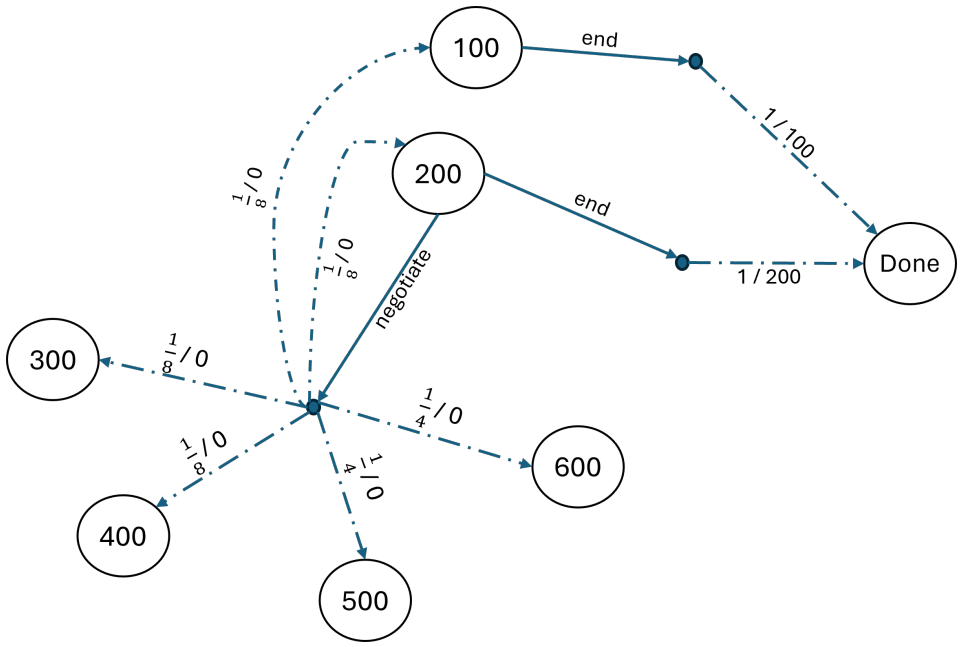
This MDP has states 10 states: 0, 100, 200, 300, 400, 500, 600, 700, 800, and *Done*. Each state corresponds to the annual increase in workers' salary offered by the employer, and also a *Done* state where the negotiation ends. The negotiating team had received an offer of \$200, *i.e.* the MDP starts at state 200. The union has two actions: *Stop* and *negotiate* for one more week, and is forced to take the *Stop* action at states 0, 100, and 800.

Choosing the *Stop* action transitions the union to the terminal state *Done* and rewards them with the amount corresponding to the state they transitioned from. For instance, *Stop* at state 300 yields \$300. The negotiation process concludes upon reaching the *Done* state.

The *Negotiate* action is viable between states 200 and 700. After each negotiation round, the union may receive offers of \$100, \$200, \$300, or \$400 with a probability of $\frac{1}{8}$ each, and \$500 or \$600 with a probability of $\frac{1}{4}$ each.

If the union chooses to *Negotiate* from state s and receives an offer of o, it transitions to state $s+o-200$, provided that $s+o-200 \leq 800$. Here, s represents the current state's dollar amount, o is the offered amount, and the deduction of 200 accounts for the price of negotiating for one week. If $s+o-200 > 800$, the union fails and transitions directly to the *Done* state without receiving any reward.

The provided diagram offers a partial illustration of the MDP, depicting transition probabilities and rewards for a better understanding. Note that the complete MDP graph is not displayed. The two numbers next to each edge represent the probability and reward of a transition. For instance, $\frac{1}{8}$ / 0 denotes that the probability of this transition is $\frac{1}{8}$ and the reward for it is 0.



Note that this figure does not show the complete MDP graph.

6.a [7 points] The union's initial policy π is given in the table below. Evaluate the policy at each state, with $\gamma = 1$. Note that the actions at state $0, 100$, and 800 are *Stop* and are fixed into the rule.

State	200	300	400	500	600	700
$\pi(s)$	*Negotiate*	*Negotiate*	*Stop*	*Stop*	*Stop*	*Stop*
$V_\pi(s)$						

Answer. *Done* is a terminal state without any actions in which the process ends. Thus, $V_\pi(\text{Done}) = 0$. By policy π, we choose action *Stop* in states 400, 500, 600, and 700, which deterministically leads to state *Done*. Thus,

$$\begin{aligned}
V_\pi(400) &= R(400, \text{Stop}, \text{Done}) + \gamma V_\pi(\text{Done}) = 400 \\
V_\pi(500) &= R(500, \text{Stop}, \text{Done}) + \gamma V_\pi(\text{Done}) = 500 \\
V_\pi(600) &= R(600, \text{Stop}, \text{Done}) + \gamma V_\pi(\text{Done}) = 600 \\
V_\pi(700) &= R(700, \text{Stop}, \text{Done}) + \gamma V_\pi(\text{Done}) = 700
\end{aligned}$$

Regarding states 200 and 300,

$$\begin{aligned}
V_\pi(200) &= \frac{1}{8}[V_\pi(100) + V_\pi(200) + V_\pi(300) + V_\pi(400)] + \frac{1}{4}[V_\pi(500) + V_\pi(600)] \\
&\Rightarrow 2700 = 7V_\pi(200) - V_\pi(300) \qquad (1) \\
V_\pi(300) &= \frac{1}{8}[V_\pi(200) + V_\pi(300) + V_\pi(400) + V_\pi(500)] + \frac{1}{4}[V_\pi(600) + V_\pi(700)] \\
&\Rightarrow 3500 = 7V_\pi(300) - V_\pi(200) \qquad (2)
\end{aligned}$$

Solving the system of linear equations derived from (1) and (2) would result in $V_\pi(200) = 1400/3$ and $V_\pi(300) = 1700/3$.

State	200	300	400	500	600	700
$\pi^i(s)$	*Negotiate*	*Negotiate*	*Stop*	*Stop*	*Stop*	*Stop*
$V_{\pi^i}(s)$	1400/3	1700/3	400	500	600	700

6.b [4 points] One of the members of union claims that the strategy given in the table in part a, *i.e.*, π, is the an optimal strategy. Is that true? Prove why?

Answer. We can show that it is not optimal by use of policy improvement. If updating the policy results in a different policy, then π is not optimal. Let's find the Q-values. Consider state 400 for instance.

$$\begin{aligned} Q_\pi(400, \text{Negotiation}) &= \frac{1}{8}[V_\pi(300) + V_\pi(400) + V_\pi(500) + V_\pi(600)] + \frac{1}{4}[V_\pi(700) + V_\pi(800)] \\ &= \frac{1}{8}[V_\pi(300) + 400 + 500 + 600] + \frac{1}{4}[700 + 800] \\ &= \frac{V_\pi(300)}{8} + \frac{1500}{8} + \frac{1500}{4}. \end{aligned}$$

$$Q_\pi(400, \text{Stop}) = V_\pi(400) = 400.$$

Since $V_\pi(300) > 0$, it is easy to see that $Q_\pi(400, \text{Negotiation}) > Q_\pi(400, \text{Stop})$. Therefore, $\pi(400)$ will be updated to *Negotiation*. Therefore, after one step of policy improvement, policy π will be updated to a different policy. Thus, π is not an optimal policy.

Q7 **[10 pts] Reinforcement Learning.** This question consists of three independent parts. Please note that **7.c** is on the next page.

7.a **[1 pts]** Recall that the expected utility recursion under the optimal policysatisfies:

$$V^\star(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma\ V^\star(s')].$$

Rewrite the above recursion for the $Q^\star$.
[**NOTE:** Your answer must be in terms of the functions T, R, $Q^\star$.]

[Write your answer to 7.a here.]

$$Q^\star(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q^\star(s',a')]$$

7.b **[3 pts]** Given the following list of Q-values for state s and the set of actions $\{\text{Left}, \text{Right}, \text{Fire}\}$:

$$Q(s,\text{Left}) = 0.15, \qquad Q(s,\text{Right}) = 0.95, \qquad Q(s,\text{Fire}) = 0.5$$

What is the probability that we will take each action on our next move when following an ϵ-greedy exploration policy? Show your process below. Assume all random movements are chosen uniformly from all actions. [**Relax!** To make your life easier, a correct answer for $(1-\epsilon)$-greedy exploration policy would also receive full mark.]

Answer.
Probability of taking the action Left $= \epsilon/3$
Probability of taking the action Right $= (1-\epsilon) + \epsilon/3$
Probability of taking the action Fire $= \epsilon/3$

7.c [6 pts] You are playing a peculiar card game, but unfortunately you were not paying attention when the rules were described. You did manage to pick up that for each round you will be holding one of three possible cards Ace, King, Jack (A, K, J, for short) and you can either Bet or Pass, in which case the dealer will reward you points and possibly switch out your card. You decide to use Q-Learning to learn to play this game, in particular you model this game as an MDP with states $\{\text{A}, \text{K}, \text{J}\}$, actions $\{\text{Bet}, \text{Pass}\}$ and discount $\gamma = 1$. To learn the game you use $\alpha = 0.25$.

7.c.i [4 pts] Say you observe the following rounds of play (in order):

s	a	s'	r
A	Bet	K	4
J	Pass	A	0
K	Pass	A	-4
K	Bet	J	-12
J	Bet	A	4
A	Bet	A	-4

What are the estimates for the following Q-values as obtained by Q-learning? All Q-values are initialized to 0. Show your process.

$$Q(J, Bet) =$$

Answer.

- After A, Bet, K, 4:
  $Q(A, \text{Bet}) \leftarrow (1-\alpha)Q(A, \text{Bet}) + \alpha[4 + \gamma \max_{a'} Q(K, a')]$
  Thus, $Q(A, \text{Bet}) \leftarrow (1-\alpha)0 + \alpha[4 + \gamma 0] = 1$.
- From (J, Pass, A, 0) to (K, Bet, J, -12):
  $Q(A, a)$'s won't get updated.
  $Q(J, \text{Bet})$ won't get updated.
- After J, Bet, A, 4:
  $Q(J, \text{Bet}) \leftarrow (1-\alpha)Q(J, \text{Bet}) + \alpha[4 + \gamma \max_{a'} Q(A, a')]$
  Thus, $Q(J, \text{Bet}) \leftarrow (1-\alpha)0 + \alpha[4 + \gamma 1] = 5/4$.

7.c.ii [2 pts] For this next part, we will switch to a feature based representation. We will use two features:

$$f_1(s,a) = 1,$$

$$f_2(s,a) = \begin{cases} 1, \text{if } a = \text{Bet} \\ 0, \text{if } a = \text{Pass} \end{cases}$$

Starting from initial weights of 0, compute the updated weights after observing the following samples:

s	a	s'	r
A	Bet	K	8
K	Pass	A	0

What are the weights after the first update (*i.e.*, after using the first sample), and after the second update (*i.e.*, after using the second sample). Show your process below. Only correct answer with correct process will receive full mark.

Answer.

- After A, Bet, K, 8: Let $\delta = [8 + \gamma \max_{a'} Q(K, a')] - Q(A, \text{Bet}) = 8$.
  - $w_1 \leftarrow w_1 + \alpha\delta f_1(A, \text{Bet})$. Assuming the same α as in part a, we would have $w_1 = 2$.
  - $w_2 \leftarrow w_2 + \alpha\delta f_2(A, \text{Bet})$. Thus, we would have $w_2 = 2$.
- After K, Pass, A, 0: Let $\delta = [0 + \gamma \max_{a'} Q(A, a')] - Q(K, \text{Pass})$. Note that $Q(A, \text{Bet}) = w_1 f_1(A, \text{Bet}) + w_2 f_2(A, \text{Bet}) = 4$, and $Q(A, \text{Pass}) = w_1 f_1(A, \text{Pass}) + w_2 f_2(A, \text{Pass}) = 2$. Thus, $\max_{a'} Q(A, a') = 4$. Furtheremore, $Q(K, \text{Pass}) = w_1 f_1(K, \text{Pass}) + w_2 f_2(K, \text{Pass}) = 2$. Therefore, $\delta = [0 + \gamma \max_{a'} Q(A, a')] - Q(K, \text{Pass}) = 4 - 2 = 2$.
  - $w_1 \leftarrow w_1 + \alpha\delta f_1(K, \text{Pass})$. Assuming the same α as in part a, we would have $w_1 = 2.5$.
  - $w_2 \leftarrow w_2 + \alpha\delta f_2(K, \text{Pass})$. Thus, we would have $w_2 = 2$.

[**This page is intentionally left blank. You may use it for your scratch notes or extra space for your answers.**]

Formula Sheet

- Most common activation functions:

Sigmoid	Tanh	ReLU	Leaky ReLU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = g(z) = \max(\epsilon z, z)$ with $\epsilon \ll 1$

- Matrix/vector manipulation

Rule	**Comments**
$(AB)^\top = B^\top A^\top$	order is reversed, everything is transposed
$(\underline{a}^\top B \underline{c})^\top = c^\top B^\top \underline{a}$	as above
$\underline{a}^\top \underline{c} = \underline{c}^\top \underline{a}$	(the result is a scalar, and the transpose of a scalar is itself)
$(A + B)C = AC + BC$	multiplication is distributive
$(\underline{a} + \underline{b})^\top C = \underline{a}^\top C + \underline{b}^\top C$	as above, with vectors
$AB \neq BA$	multiplication is **not** commutative

- Common vector derivatives

$f(\underline{x})$	$\underline{x}^\top \underline{a}$	$\underline{a}^\top \underline{x}$	$\underline{x}^\top \underline{x}$	$\underline{x}^\top A \underline{x}$
$\nabla_{\underline{x}} f(\underline{x})$	$\underline{a}$	$\underline{a}$	$2\underline{x}$	$(A^\top + A)\underline{x}$

- Probability density function of a normal distribution with mean $\underline{\mu}$ and covariance matrix Σ:

$$\mathcal{N}(\underline{x} \,|\, \underline{\mu}, \Sigma) = \frac{1}{|2\pi\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\underline{x} - \underline{\mu})^\top \Sigma^{-1} (\underline{x} - \underline{\mu})\right)$$

Note: For scalar random variable with normal distribution with mean μ and variance σ^2:

$$\mathcal{N}(x \,|\, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$