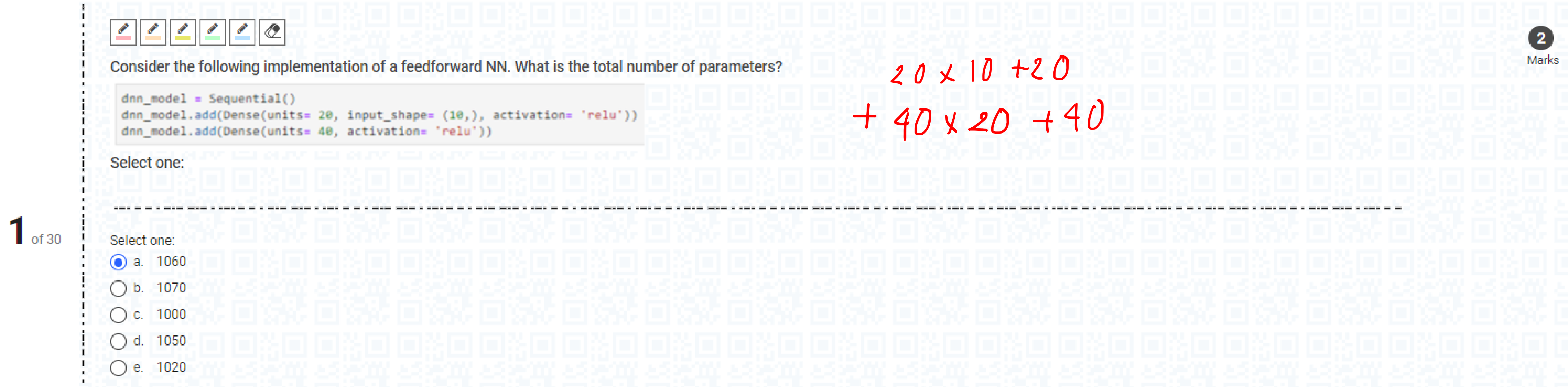
Solutions for FIT5215
Semester 2, 2021

Part I



2 Marks

1 of 30

Consider the following implementation of a feedforward NN. What is the total number of parameters?

```
dnn_model = Sequential()
dnn_model.add(Dense(units= 20, input_shape= (10,), activation= 'relu'))
dnn_model.add(Dense(units= 40, activation= 'relu'))
```

Select one:

$20 \times 10 + 20$

$+ 40 \times 20 + 40$

Select one:

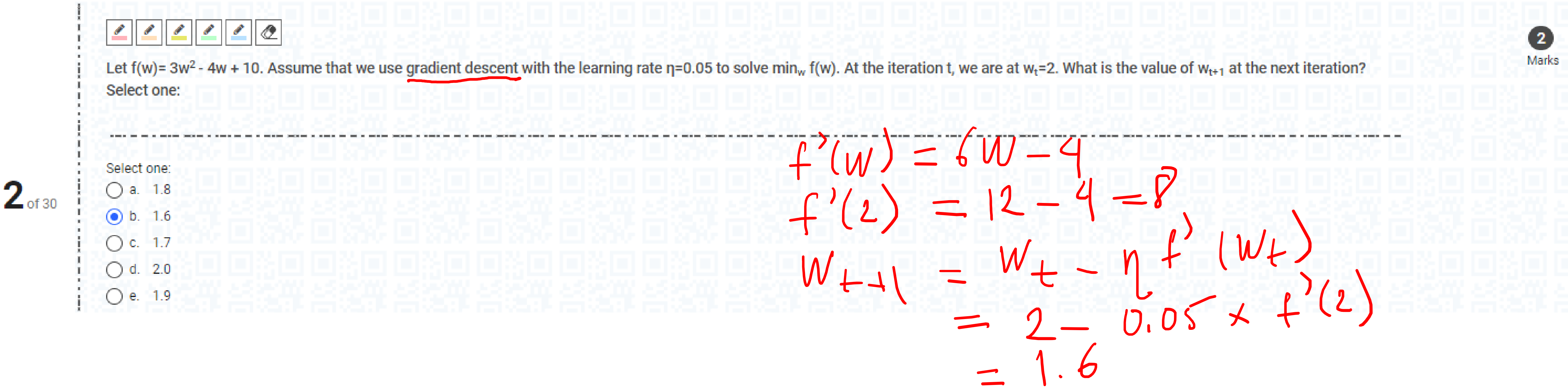
● a. 1060

○ b. 1070

○ c. 1000

○ d. 1050

○ e. 1020



2 Marks

2 of 30

Let f(w)= 3w^2 - 4w + 10. Assume that we use gradient descent with the learning rate η=0.05 to solve min$_w$ f(w). At the iteration t, we are at w$_t$=2. What is the value of w$_{t+1}$ at the next iteration?

Select one:

$f'(w) = 6w - 4$

$f'(2) = 12 - 4 = 8$

$w_{t+1} = w_t - \eta f'(w_t)$

$= 2 - 0.05 \times f'(2)$

$= 1.6$

Select one:

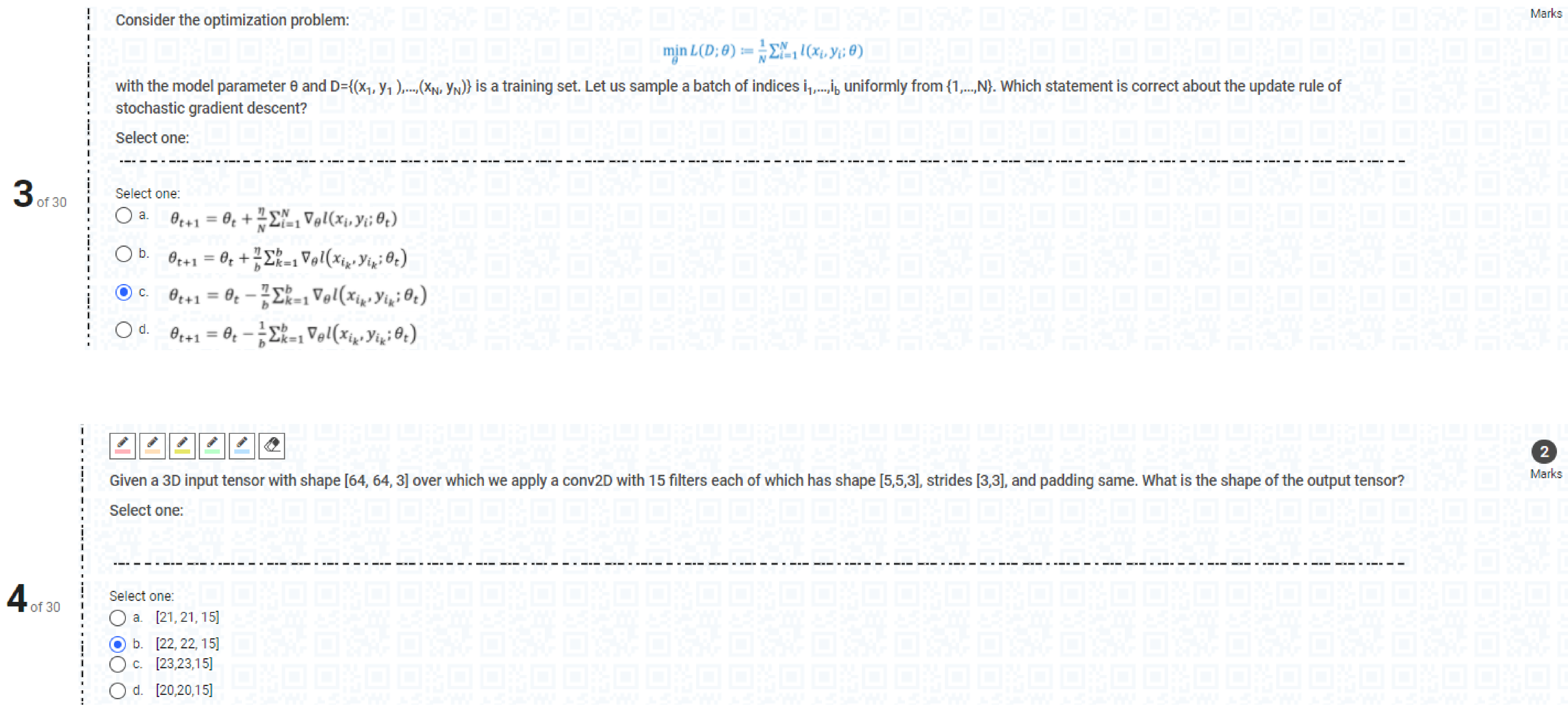
○ a. 1.8

● b. 1.6

○ c. 1.7

○ d. 2.0

○ e. 1.9



Marks

3 of 30

Consider the optimization problem:

$$\min_{\theta} L(D;\theta) := \frac{1}{N}\sum_{i=1}^{N} l(x_i, y_i; \theta)$$

with the model parameter θ and D={(x_1, y_1),...,(x_N, y_N)} is a training set. Let us sample a batch of indices i_1,...,i_b uniformly from {1,...,N}. Which statement is correct about the update rule of stochastic gradient descent?

Select one:

Select one:

○ a. $\theta_{t+1} = \theta_t + \frac{\eta}{N}\sum_{i=1}^{N} \nabla_\theta l(x_i, y_i; \theta_t)$

○ b. $\theta_{t+1} = \theta_t + \frac{\eta}{b}\sum_{k=1}^{b} \nabla_\theta l(x_{i_k}, y_{i_k}; \theta_t)$

● c. $\theta_{t+1} = \theta_t - \frac{\eta}{b}\sum_{k=1}^{b} \nabla_\theta l(x_{i_k}, y_{i_k}; \theta_t)$

○ d. $\theta_{t+1} = \theta_t - \frac{1}{b}\sum_{k=1}^{b} \nabla_\theta l(x_{i_k}, y_{i_k}; \theta_t)$

4 of 30

2 Marks

Given a 3D input tensor with shape [64, 64, 3] over which we apply a conv2D with 15 filters each of which has shape [5,5,3], strides [3,3], and padding same. What is the shape of the output tensor?

Select one:

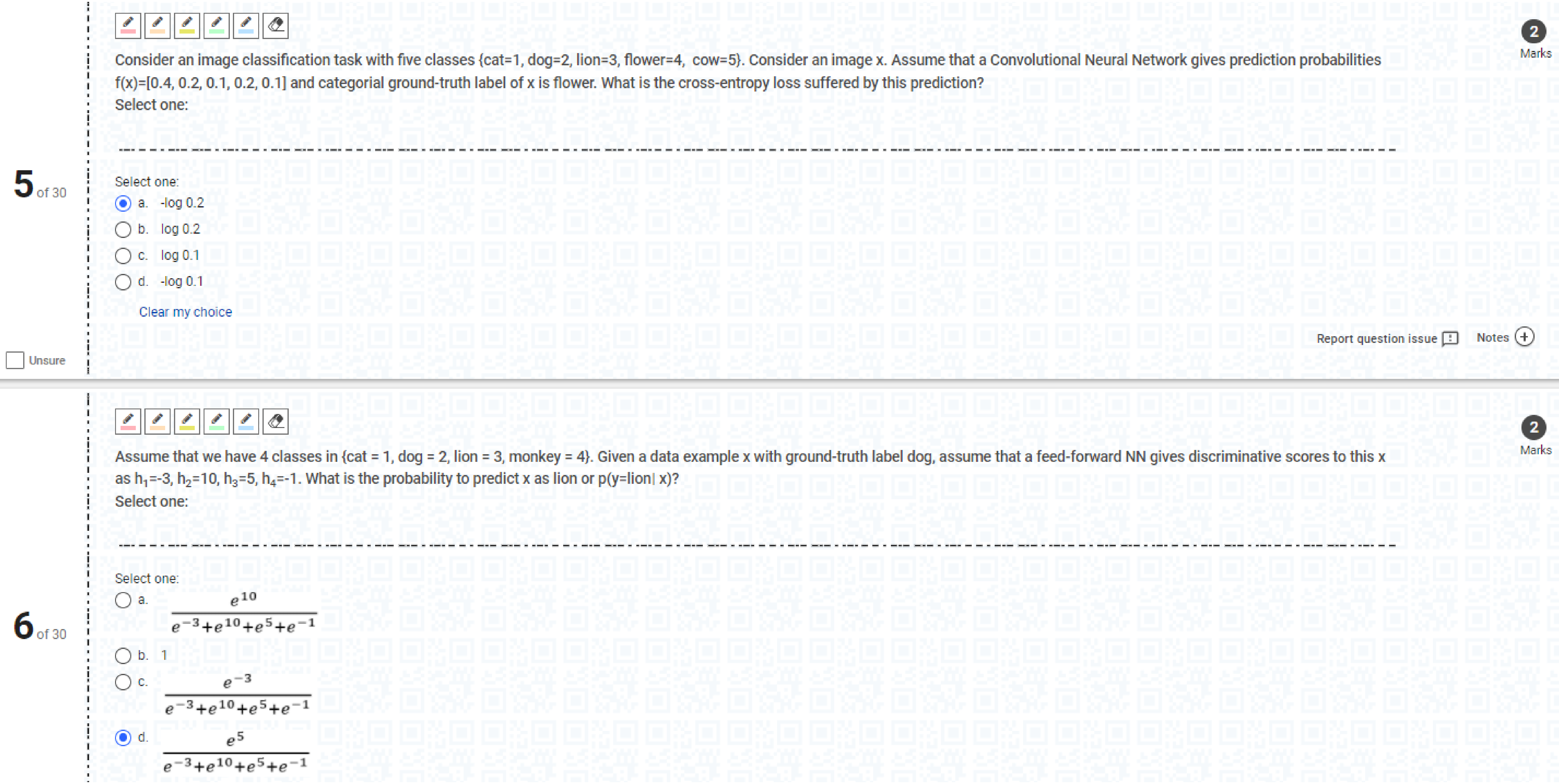
Select one:

○ a. [21, 21, 15]

● b. [22, 22, 15]

○ c. [23,23,15]

○ d. [20,20,15]



5 of 30

2 Marks

Consider an image classification task with five classes {cat=1, dog=2, lion=3, flower=4, cow=5}. Consider an image x. Assume that a Convolutional Neural Network gives prediction probabilities f(x)=[0.4, 0.2, 0.1, 0.2, 0.1] and categorial ground-truth label of x is flower. What is the cross-entropy loss suffered by this prediction?

Select one:

Select one:

● a. -log 0.2

○ b. log 0.2

○ c. log 0.1

○ d. -log 0.1

Clear my choice

□ Unsure

Report question issue

Notes

6 of 30

2 Marks

Assume that we have 4 classes in {cat = 1, dog = 2, lion = 3, monkey = 4}. Given a data example x with ground-truth label dog, assume that a feed-forward NN gives discriminative scores to this x as h_1=-3, h_2=10, h_3=5, h_4=-1. What is the probability to predict x as lion or p(y=lion| x)?

Select one:

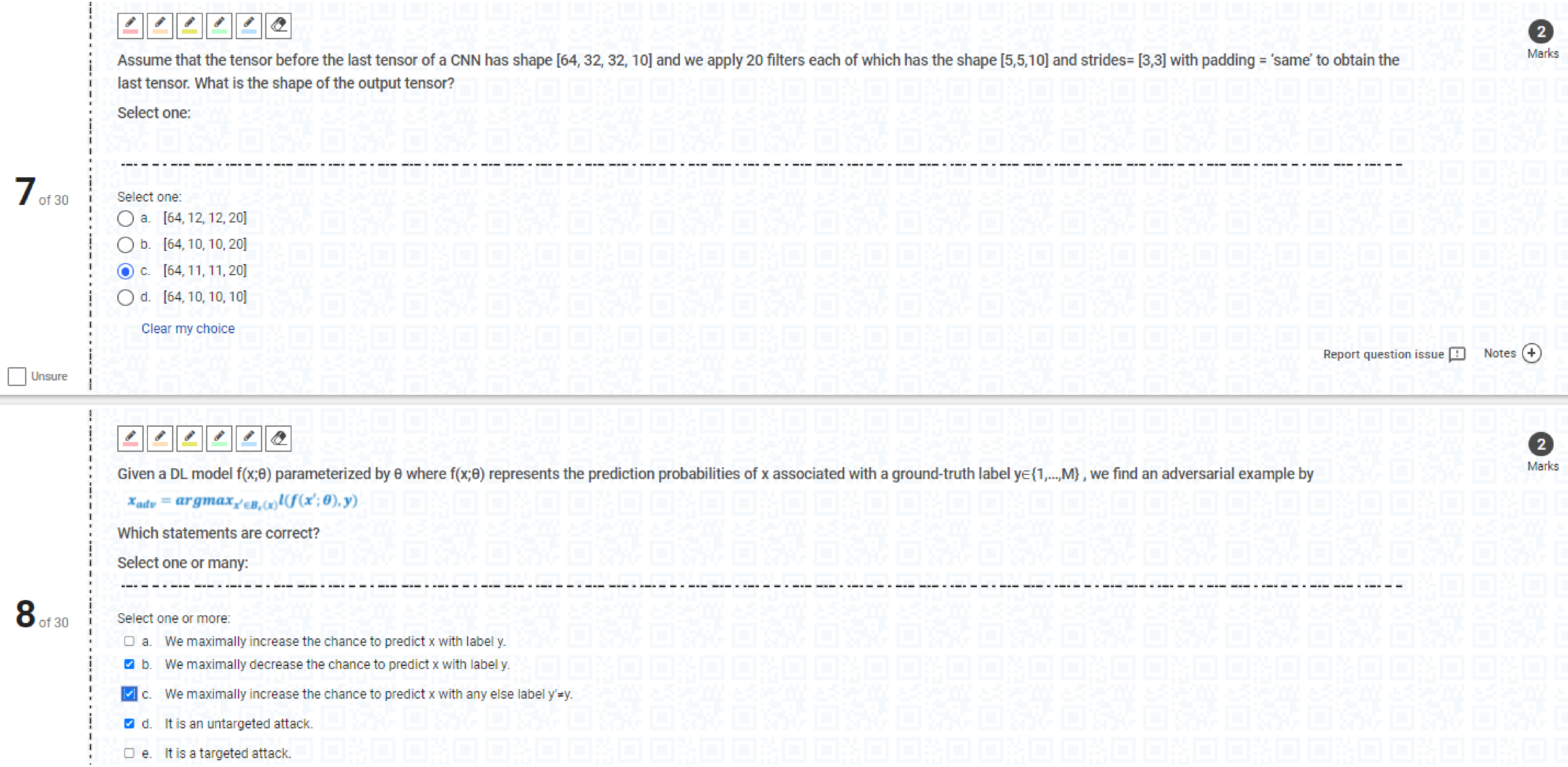
Select one:

○ a. $\frac{e^{10}}{e^{-3}+e^{10}+e^{5}+e^{-1}}$

○ b. 1

○ c. $\frac{e^{-3}}{e^{-3}+e^{10}+e^{5}+e^{-1}}$

● d. $\frac{e^{5}}{e^{-3}+e^{10}+e^{5}+e^{-1}}$



7 of 30

2 Marks

Assume that the tensor before the last tensor of a CNN has shape [64, 32, 32, 10] and we apply 20 filters each of which has the shape [5,5,10] and strides= [3,3] with padding = 'same' to obtain the last tensor. What is the shape of the output tensor?

Select one:

Select one:

○ a. [64, 12, 12, 20]

○ b. [64, 10, 10, 20]

● c. [64, 11, 11, 20]

○ d. [64, 10, 10, 10]

Clear my choice

□ Unsure

Report question issue

Notes

8 of 30

2 Marks

Given a DL model f(x;θ) parameterized by θ where f(x;θ) represents the prediction probabilities of x associated with a ground-truth label y∈{1,...,M} , we find an adversarial example by

$x_{adv} = argmax_{x' \in B_\epsilon(x)} l(f(x';\theta), y)$

Which statements are correct?

Select one or many:

Select one or more:

□ a. We maximally increase the chance to predict x with label y.

☑ b. We maximally decrease the chance to predict x with label y.

☑ c. We maximally increase the chance to predict x with any else label y'≠y.

☑ d. It is an untargeted attack.

□ e. It is a targeted attack.

9 of 30

2 Marks

Given a skip-gram model with vocabulary size 500 and embedding size 150, we consider a pair of target and context words with indices 2 and 8 respectively. Let U and V be two weight matrices connecting the input to hidden layers and hidden to output layers. What statements are correct?

Select one or many:

Select one or more:

☑ a. The hidden value h is the row 2 of the matrix U.

□ b. The hidden value h is the row 8 of the matrix U

□ c. Shape of U is [500,500] and shape of V is [150,150]

□ d. Input to the network is one-hot vector 1_8.

☑ e. Shape of U is [500,150] and shape of V is [150,500]

☑ f. Input to the network is one-hot vector 1_2.

Report question issue

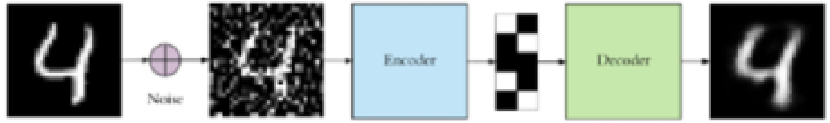
Notes

□ Unsure

10 of 30

3 Marks

How to train a denoising auto-encoder with encoder f_θ and decoder g_Φ?



Select one or many:

Select one or more:

☑ a. $\min\limits_{\theta,\Phi} \mathbb{E}_{x\sim\mathbb{P}}\left[\mathbb{E}_{x'\sim N(x,\eta I)}[d(x, g_\Phi(f_\theta(x')))]\right]$

□ b. $\min\limits_{\theta,\Phi} \mathbb{E}_{x\sim\mathbb{P}}\left[\mathbb{E}_{x'\sim N(x,\eta I)}[d(x, f_\theta(g_\Phi(x')))]\right]$

☑ c. $\min\limits_{\theta,\Phi} \mathbb{E}_{x\sim\mathbb{P}}\left[\mathbb{E}_{\epsilon\sim N(0,\eta I)}[d(x, g_\Phi(f_\theta(x+\epsilon)))]\right]$

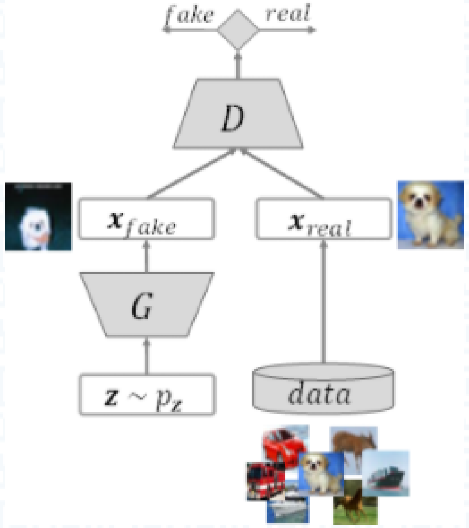
□ d. $\min\limits_{\theta,\Phi} \mathbb{E}_{x\sim\mathbb{P}}\left[\mathbb{E}_{\epsilon\sim N(0,\eta I)}[d(x, f_\theta(g_\Phi\ (x+\epsilon)))]\right]$

□ e. $\min\limits_{\theta,\Phi} \mathbb{E}_{x\sim\mathbb{P}}\left[\mathbb{E}_{\epsilon\sim N(0,\eta I)}[d(x, f_\theta(g_\Phi\ (x+\epsilon)))]\right]$

11 of 30

2 Marks

How to train GANs?



Select one or many:

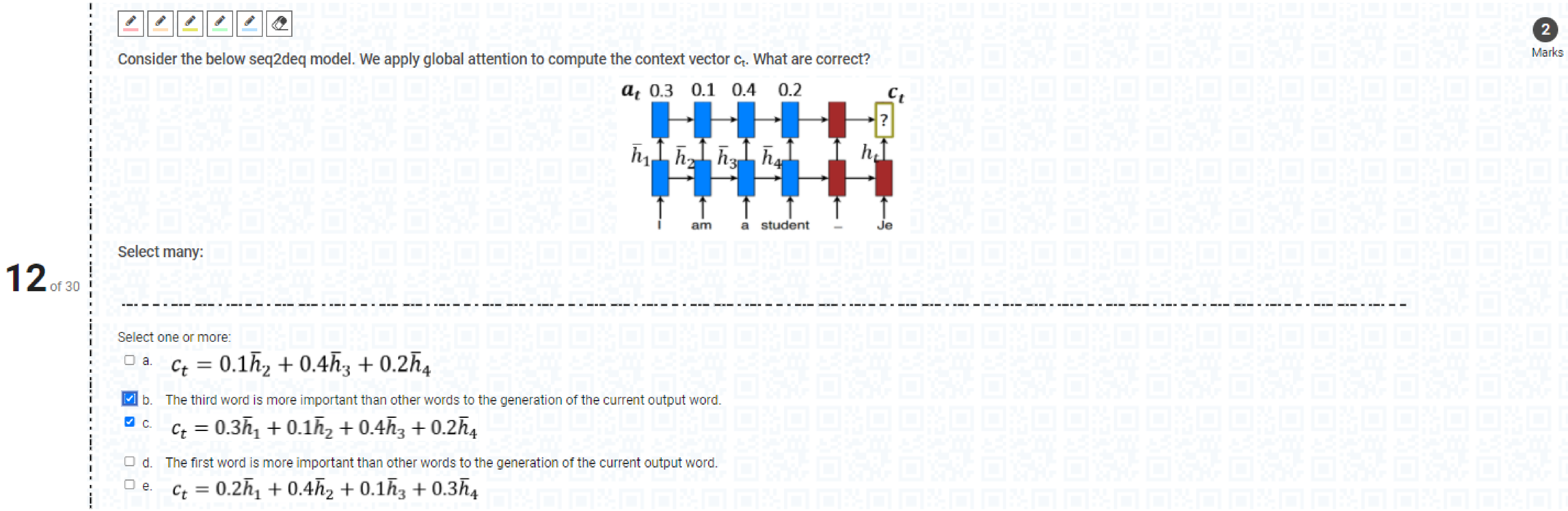
Select one or more:

☑ a. $\min_{G} \max_{D} J(G,D) = \mathbb{E}_{x \sim p_d(x)}[\log(1 - D(\boldsymbol{x}))] + \mathbb{E}_{z \sim p(z)}[\log D(G(\boldsymbol{z}))]$

□ b. $\max_{G} \min_{D} J(G,D) = \mathbb{E}_{x \sim p_d(x)}[\log D(\boldsymbol{x})] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(\boldsymbol{z}))]$

□ c. $\min_{\theta,\Phi} \mathbb{E}_{x\sim\mathbb{P}}[d(\tilde{x}, g_{\Phi}(f_{\theta}(\boldsymbol{x}))]$

☑ d. $\min_{G} \max_{D} J(G,D) = \mathbb{E}_{x \sim p_d(x)}[\log D(\boldsymbol{x})] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(\boldsymbol{z}))]$



12 of 30

2 Marks

Consider the below seq2deq model. We apply global attention to compute the context vector c_t. What are correct?

Select many:

Select one or more:

□ a. $c_t = 0.1\bar{h}_2 + 0.4\bar{h}_3 + 0.2\bar{h}_4$

☑ b. The third word is more important than other words to the generation of the current output word.

☑ c. $c_t = 0.3\bar{h}_1 + 0.1\bar{h}_2 + 0.4\bar{h}_3 + 0.2\bar{h}_4$

□ d. The first word is more important than other words to the generation of the current output word.

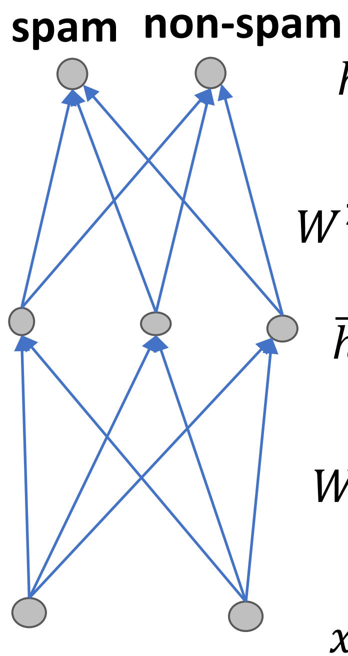
□ e. $c_t = 0.2\bar{h}_1 + 0.4\bar{h}_2 + 0.1\bar{h}_3 + 0.3\bar{h}_4$

Part II

II.1. Computational process of MLP

- Consider a feed-forward neural network as shown in the figure for spam email detection with two labels (spam=1 and non-spam=2). Assume that we feed a feature vector $x = [-1,1]^T$ with true label $y = 2$ to the network.

1. What are the formulas and the values of $\bar{h}^1, h^1$? **[4 points]**
2. What are the formulas and the values for the logit h^2 and the prediction probability p? **[4 points]**
3. What is the predicted label $\hat{y}$ and the cross-entropy loss for this prediction? Is it a correct or incorrect prediction? **[2 points]**



$h^2, p = softmax(h^2)$

$W^2 = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, b^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$\bar{h}^1, h^1 = ReLU(\bar{h}^1)$

$W^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix}, b^1 = \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix}$

$x = [-1,1]^T = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, y = 2$

Solution

1. $\bar{h}^1 = W^1x + b^1 = [-1\ 1\ 1]^T + [-2\ 0\ 1]^T = [-3\ 1\ 2]^T$ (**1.5 points**)

$h^1 = ReLU\left(\bar{h}^1\right) = [0\ 1\ 2]^T$ (**1.5 points**)

2. $h^2 = W^2h^1 + b^2 = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 1 \end{bmatrix} [0\ 1\ 2]^T + [0\ 1]^T = [1\ 4]^T$ (**1.5 points**)

$p = softmax(h^2) = \left[\frac{e}{e+e^4}\ \ \frac{e^4}{e+e^4}\right] = [0.047\ 0.953]$ (**1.5 points**)

3. The prediction label $\hat{y} = 2$ (**1 point**) and correct prediction (**1 point**).

$l(y, p) = CE\left(1_y, p\right) = -\log\frac{e^4}{e+e^4}$ (**2 points**)

II.2. CNN

- Consider an 4D tensor of a CNN with shape [32, 64, 64, 10]. We apply a Conv2D with 10 filters each of which has the shape [5,5,10] and strides= [3,3] with padding = ‘valid’ to obtain an another tensor. On top of this tensor, we apply MaxPool2D with strides= (3,3), pool/kernel size = (2,2), and padding= ‘same’ to gain an 4D tensor. Finally, we flatten the last tensor to obtain a dense layer. What is the number of neurons on the dense layer? Show the steps of your answer.

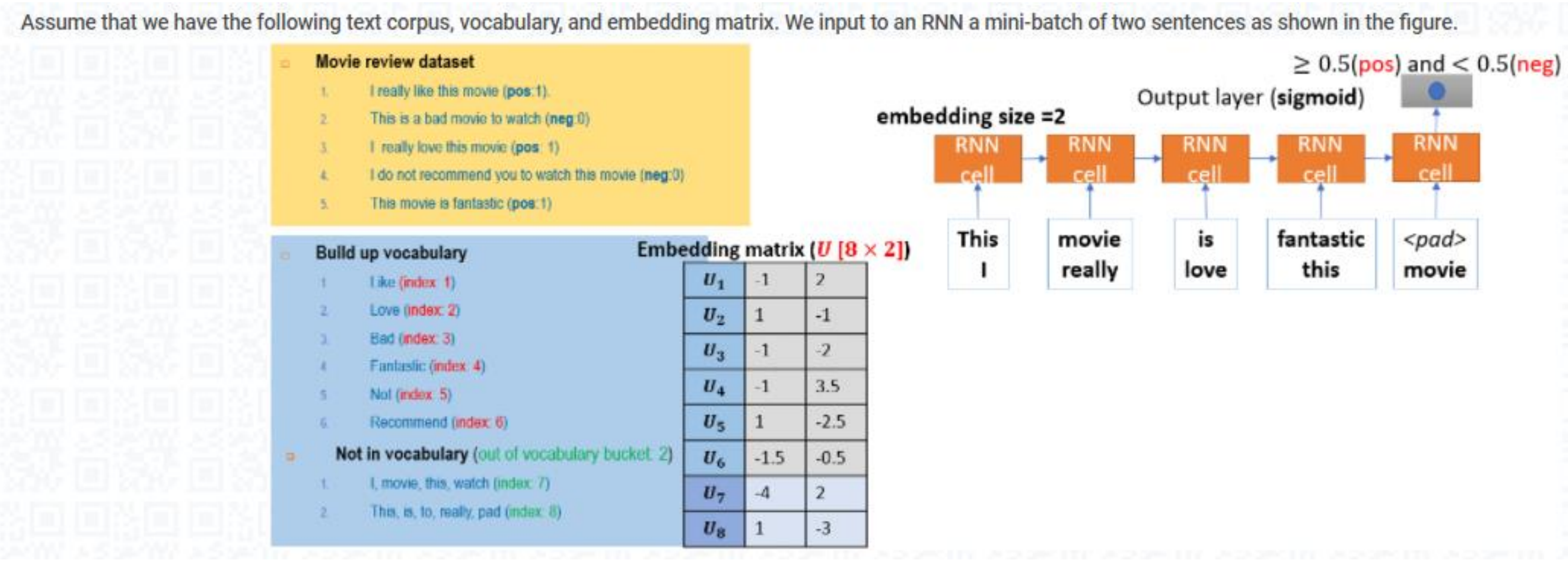
[5 points]

Solution

- First 4D tensor: [32, 20, 20, 10] (**2 points**)
    - Width and height are correct (**1 points**)
- Second 4D tensor: [32, 7, 7, 10] (**2 points**)
    - Width and height are correct (1 points)
- Number of neurons of the flattened layer (**1 point**)
    - $7 \times 7 \times 10 = 490$

II.3. RNN

- Assume that we have the following text corpus, vocabulary, and embedding matrix. We input to an RNN a mini-batch of two sentences as shown in the figure.
  1. What are the numeric input items at each timestep really fed to the RNN? **[2 points]**
  2. What is the 3D tensor with the shape $[batch\_size, seq\_len, embed\_size]$ over timesteps that is fed to the RNN? **[3 points]**



Assume that we have the following text corpus, vocabulary, and embedding matrix. We input to an RNN a mini-batch of two sentences as shown in the figure.

Movie review dataset

1. I really like this movie (**pos**:1).
2. This is a bad movie to watch (**neg**:0)
3. I really love this movie (**pos**: 1)
4. I do not recommend you to watch this movie (**neg**:0)
5. This movie is fantastic (**pos**:1)

Build up vocabulary

1. Like (index: 1)
2. Love (index: 2)
3. Bad (index: 3)
4. Fantastic (index: 4)
5. Not (index: 5)
6. Recommend (index: 6)

Not in vocabulary (out of vocabulary bucket: 2)

1. I, movie, this, watch (index: 7)
2. This, is, to, really, pad (index: 8)

Embedding matrix (U [8×2])

U_1	-1	2
U_2	1	-1
U_3	-1	-2
U_4	-1	3.5
U_5	1	-2.5
U_6	-1.5	-0.5
U_7	-4	2
U_8	1	-3

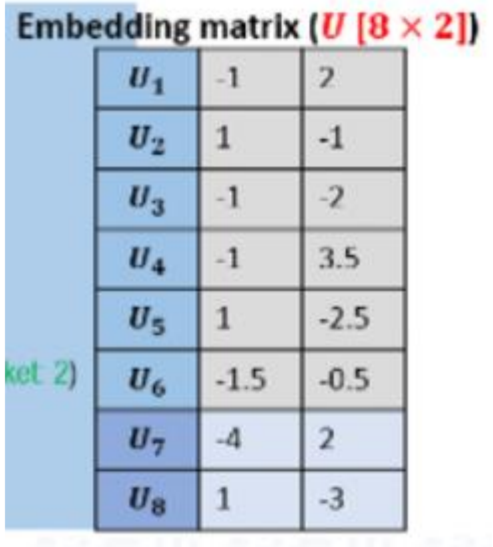
Solution

1. $[8,7]^T, [7,8]^T, [8,2]^T, [4,8]^T, [8,7]^T$ (**2 points**)

Or $\begin{bmatrix}U_8\\U_7\end{bmatrix}, \begin{bmatrix}U_7\\U_8\end{bmatrix}, \begin{bmatrix}U_8\\U_2\end{bmatrix}, \begin{bmatrix}U_4\\U_8\end{bmatrix}, \begin{bmatrix}U_8\\U_7\end{bmatrix}$

2. [[[1,-3],[-4,2],[1,-3],[-1,3.5],[1,-3]], (**3 points**)

[[-4,2],[1,-3],[1,-1],[1,-3],[-4,2]]]



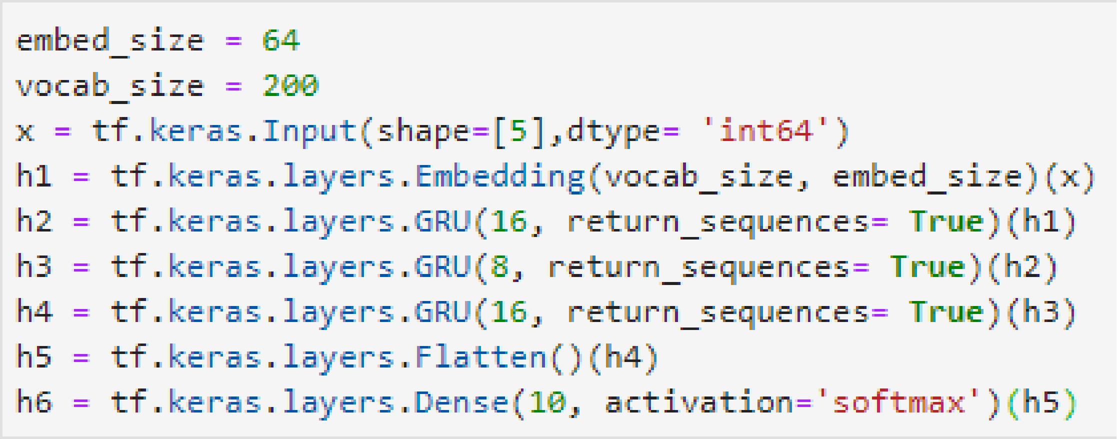
Embedding matrix (U [8 × 2])

U_1	-1	2
U_2	1	-1
U_3	-1	-2
U_4	-1	3.5
U_5	1	-2.5
U_6	-1.5	-0.5
U_7	-4	2
U_8	1	-3

II.4. Reading code for RNN

- Read the following code and provide the shape of the tensors x, h1, h2, h3, h4, h5, h6.

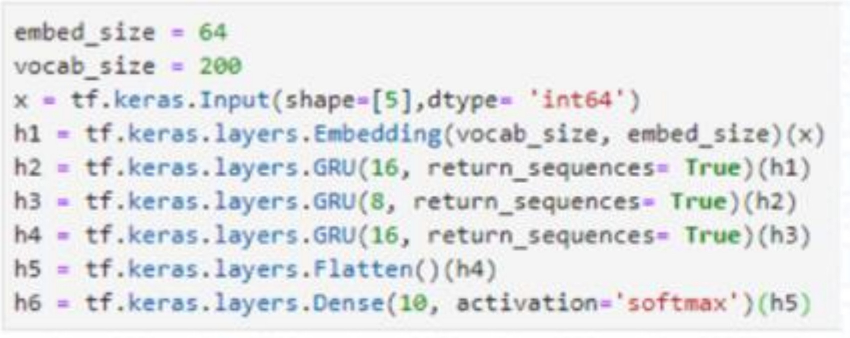
[5 points]



```
embed_size = 64
vocab_size = 200
x = tf.keras.Input(shape=[5],dtype= 'int64')
h1 = tf.keras.layers.Embedding(vocab_size, embed_size)(x)
h2 = tf.keras.layers.GRU(16, return_sequences= True)(h1)
h3 = tf.keras.layers.GRU(8, return_sequences= True)(h2)
h4 = tf.keras.layers.GRU(16, return_sequences= True)(h3)
h5 = tf.keras.layers.Flatten()(h4)
h6 = tf.keras.layers.Dense(10, activation='softmax')(h5)
```

Solution

- x: [None, 5]   (**1 point**)
- h1: [None, 5, 64] (**1 point**)
- h2: [None, 5, 16] (**0.5 point**)
- h3: [None, 5, 8] (**0.5 point**)
- h4: [None, 5, 16] (**0.5 point**)
- h5: [None, 80] (**1 point**)
- h6: [None, 10] (**0.5 point**)

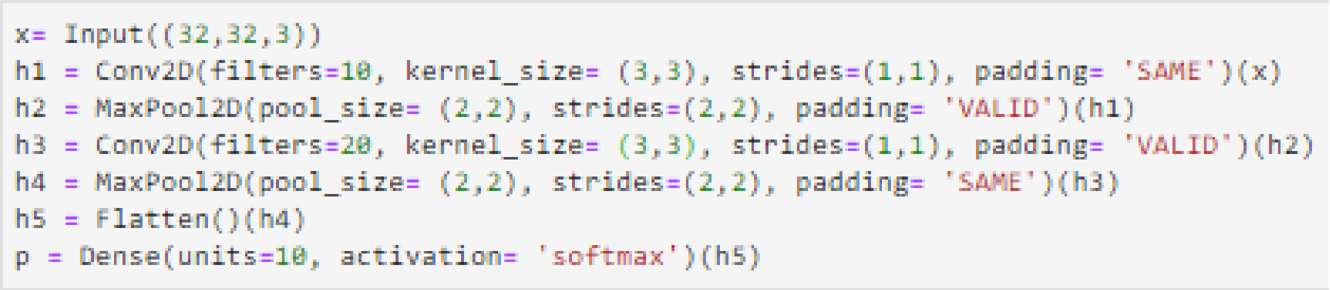


```
embed_size = 64
vocab_size = 200
x = tf.keras.Input(shape=[5],dtype= 'int64')
h1 = tf.keras.layers.Embedding(vocab_size, embed_size)(x)
h2 = tf.keras.layers.GRU(16, return_sequences= True)(h1)
h3 = tf.keras.layers.GRU(8, return_sequences= True)(h2)
h4 = tf.keras.layers.GRU(16, return_sequences= True)(h3)
h5 = tf.keras.layers.Flatten()(h4)
h6 = tf.keras.layers.Dense(10, activation='softmax')(h5)
```

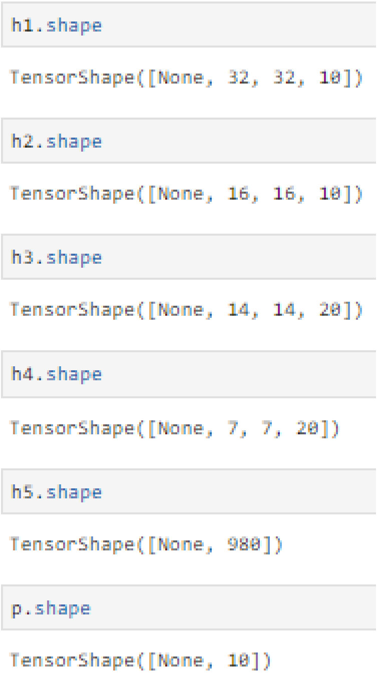
II.5. Reading code for CNN

- Read the following code and provide the shape of the tensors x, h1, h2, h3, h4, h5, p.

[5 points]

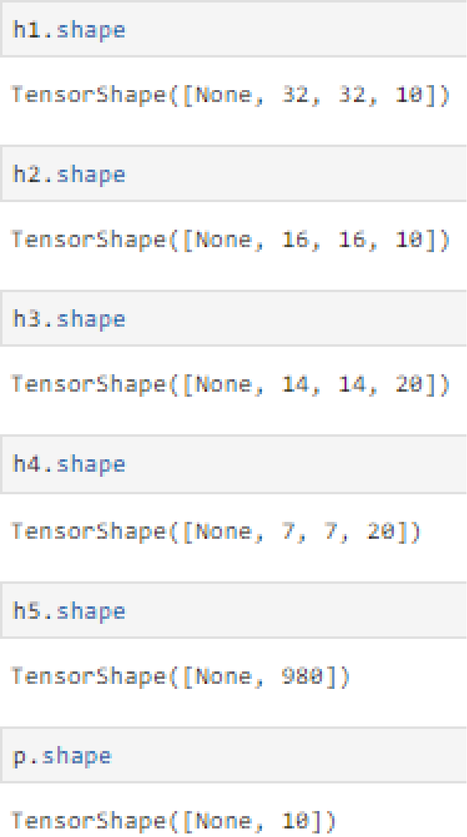


```
x= Input((32,32,3))
h1 = Conv2D(filters=10, kernel_size= (3,3), strides=(1,1), padding= 'SAME')(x)
h2 = MaxPool2D(pool_size= (2,2), strides=(2,2), padding= 'VALID')(h1)
h3 = Conv2D(filters=20, kernel_size= (3,3), strides=(1,1), padding= 'VALID')(h2)
h4 = MaxPool2D(pool_size= (2,2), strides=(2,2), padding= 'SAME')(h3)
h5 = Flatten()(h4)
p = Dense(units=10, activation= 'softmax')(h5)
```



```
h1.shape

TensorShape([None, 32, 32, 10])

h2.shape

TensorShape([None, 16, 16, 10])

h3.shape

TensorShape([None, 14, 14, 20])

h4.shape

TensorShape([None, 7, 7, 20])

h5.shape

TensorShape([None, 980])

p.shape

TensorShape([None, 10])
```

Solution



```
h1.shape
```

```
TensorShape([None, 32, 32, 10])
```

0.9 point

```
h2.shape
```

```
TensorShape([None, 16, 16, 10])
```

0.9 point

```
h3.shape
```

```
TensorShape([None, 14, 14, 20])
```

0.8 point

```
h4.shape
```

```
TensorShape([None, 7, 7, 20])
```

0.8 point

```
h5.shape
```

```
TensorShape([None, 980])
```

0.8 point

```
p.shape
```

```
TensorShape([None, 10])
```

0.8 point

II.6. CNN

- Assume that we have [6,6,1] input tensor as shown below and applying max pooling or average pooling with kernel size = [2,2], strides = [2,2], padding= valid.

1. What are the output tensors if we apply max pooling?

   Note: you can answer the output tensor by listing its rows (row1=[], row2=[], and so on)

   [2.5 points]

1. What are the output tensors if we apply average pooling?

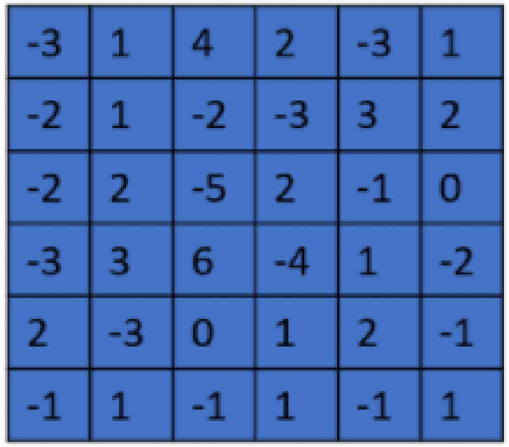
   [2.5 points]

-3	1	4	2	-3	1
-2	1	-2	-3	3	2
-2	2	-5	2	-1	0
-3	3	6	-4	1	-2
2	-3	0	1	2	-1
-1	1	-1	1	-1	1

Solution

1. The output of max pooling layer (**2.5 points**)
   - row1= [1 4 3] (**0.8 points**)
   - row2= [3 6 1] (**0.8 points**)
   - row3=[2 1 2] (**0.9 points**)

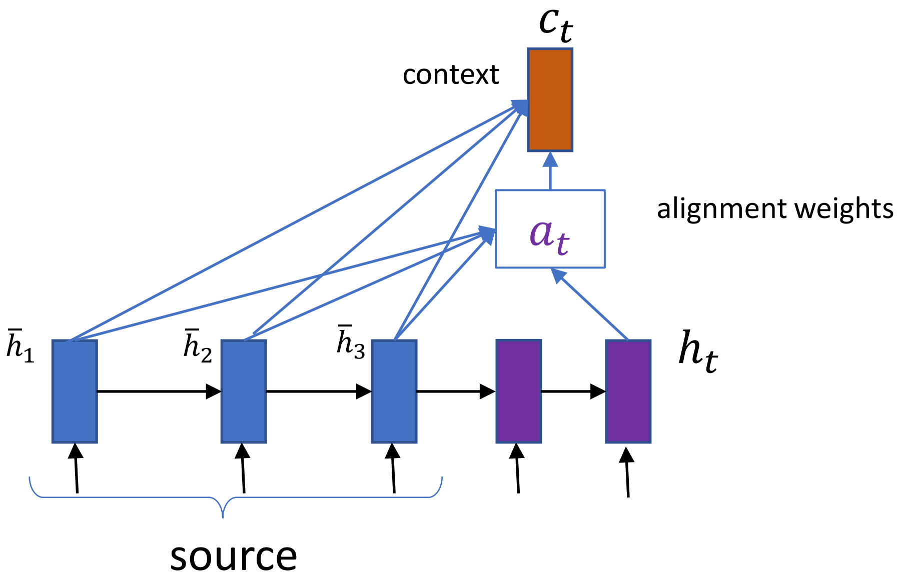
2. The output of max pooling layer (**2.5 points**)
   - row1= [-0.75 0.25 0.75] (**0.8 points**)
   - row2= [0 -0.25 -0.5] (**0.8 points**)
   - row3=[-0.25 0.25 0.25] (**0.9 points**)



-3	1	4	2	-3	1
-2	1	-2	-3	3	2
-2	2	-5	2	-1	0
-3	3	6	-4	1	-2
2	-3	0	1	2	-1
-1	1	-1	1	-1	1

II.7. Global Attention

- Attention mechanism allows the decoding network to look back to input sequence and is one of the most important techniques that has helped to achieve recent breakthroughs in NLP and seq2seq models. Consider the following encoding sequence of length 3 from the source and we are interested in using a global attention mechanism to decode the first target output. Answer the following questions:



II.7.a

- Assume that the encoder and decoder hidden states have the same dimension and we use the dot product to compute the alignments scores between them. Write down the analytical expressions to compute the alignment scores $score(h_t, \bar{h}_s)$ and alignment weights $a_t(s)$ for $s = 1{,}2{,}3$. (**2 points**)

Solution

- $score\left(h_t, \bar{h}_s\right) = h_t^T \bar{h}_s =< h_t, \bar{h}_s >$ for s=1,2,3 (**1 point**)
- $a_t(s) = \frac{\exp\{score(h_t, \bar{h}_s)\}}{\sum_{i=1}^{3} \exp\{score(h_t, \bar{h}_i)\}}$ for s=1,2,3 (**1 point**)

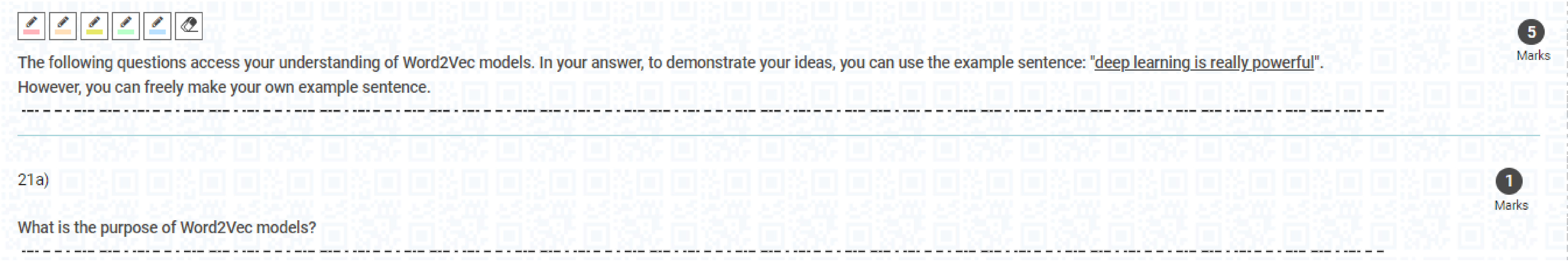
II.7.b

- For simplicity, assuming the encoder hidden states are scalars with the following values $\bar{h}_1 = 1, \bar{h}_2 = -1, \bar{h}_3 = 2$, calculate the alignment weight vector a_t and then the context vector c_t with the decoder hidden state $h_t = 1$. Note that we assume using the sign score function $score(h_t, \bar{h}_s) = sign(h_t \bar{h}_s)$ where $sign(x)$ returns 1 if $x \geq 0$ and -1 if otherwise. (**3 points**)

Solution

- $score(h_t, \bar{h}_1) = sign(h_t \bar{h}_1) = 1$ (**0.4 point**)
- $score(h_t, \bar{h}_2) = sign(h_t \bar{h}_2) = -1$ (**0.4 point**)
- $score(h_t, \bar{h}_3) = sign(h_t \bar{h}_3) = 1$ (**0.45 point**)
- $a_t(1) = \frac{e}{e+e+e^{-1}} = 0.47$ (**0.4 point**)
- $a_t(2) = \frac{e^{-1}}{e+e+e^{-1}} = 0.06$ (**0.4 point**)
- $a_t(3) = \frac{e}{e+e+e^{-1}} = 0.47$ (**0.45 point**)
- $c_t = \sum_{s=1}^{3} a_t(s)\, \bar{h}_s = 0.47 \times 1 + 0.06 \times (-1) + 0.47 \times 2 = 1.35$ ***(0.5 point)***

Part III



5

Marks

The following questions access your understanding of Word2Vec models. In your answer, to demonstrate your ideas, you can use the example sentence: "deep learning is really powerful". However, you can freely make your own example sentence.

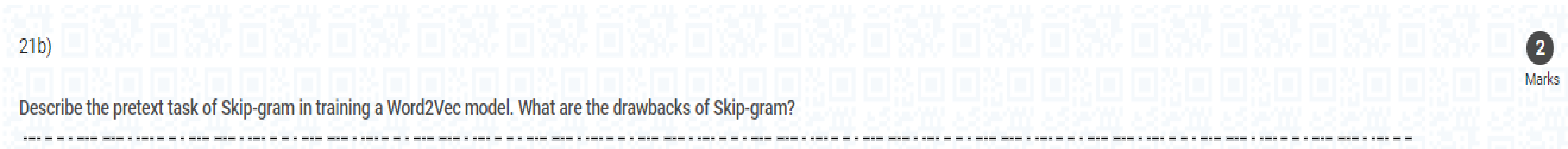
21a)

1

Marks

What is the purpose of Word2Vec models?

- Transform symbolic representation of a word to numeric representation (or vector) which preserve the semantic and syntactic relationships carried in a text corpus.



21b)

2 Marks

Describe the pretext task of Skip-gram in training a Word2Vec model. What are the drawbacks of Skip-gram?

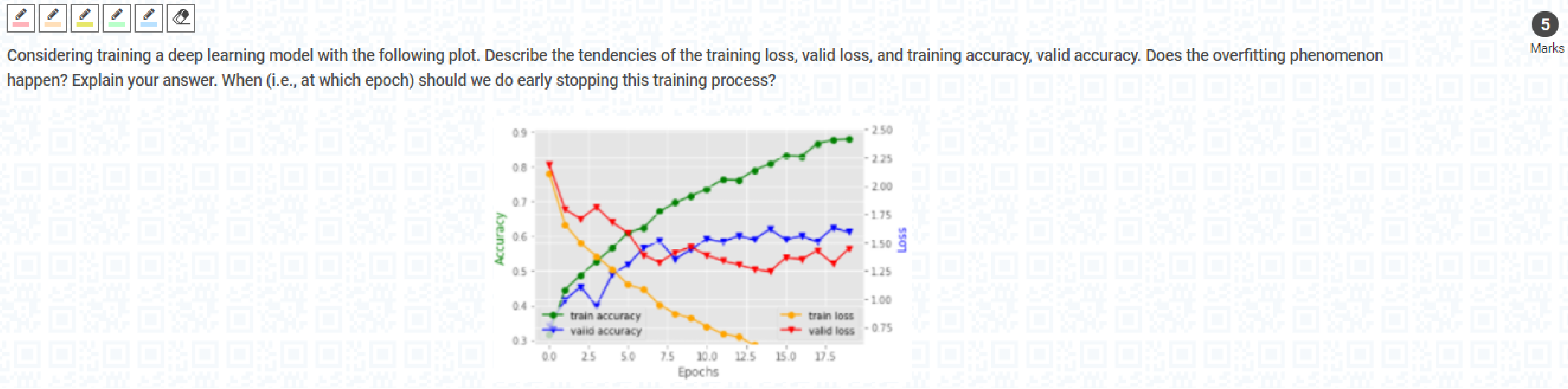
- Skip-gram uses the target word to predict the context words in a window.
- Deep learning is really powerful→ Use is to predict deep, learning, really, powerful.
- Two drawbacks
  - The output layer has N neurons where N is a large vocabulary size. Compute the prediction probabilities $p = [p_1, \ldots, p_N]$ by applying softmax to the output which is very computationally expensive.
  - The values of $p_1, \ldots, p_N$ are very tiny → hard to train.

21c)

2 Marks

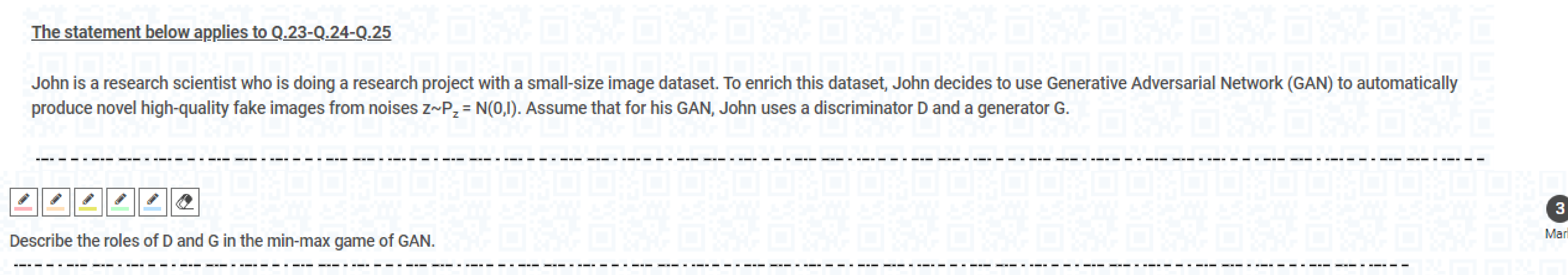
Describe Skip-gram with negative sampling.

- Turn multi-class classification to binary classification using negative words.
- Given a pair of (context word, target word), we sample some negative words from the vocabulary. Assign the pair (context word, target word) to label 1 and the pairs (context word, negative word) to label 0. Cast to a binary classification problem.



Considering training a deep learning model with the following plot. Describe the tendencies of the training loss, valid loss, and training accuracy, valid accuracy. Does the overfitting phenomenon happen? Explain your answer. When (i.e., at which epoch) should we do early stopping this training process?

- Training loss is decreasing, while valid loss is fluctuating.
- Training accuracy is increasing, while valid accuracy is fluctuating.
- Overfitting phenomenon is happening and we need to apply early stopping.
- We should apply early stopping at epoch 8
  - Students can answer 7, 9,10 or even 13



The statement below applies to Q.23-Q.24-Q.25

John is a research scientist who is doing a research project with a small-size image dataset. To enrich this dataset, John decides to use Generative Adversarial Network (GAN) to automatically produce novel high-quality fake images from noises $z \sim P_z = N(0,I)$. Assume that for his GAN, John uses a discriminator D and a generator G.

Describe the roles of D and G in the min-max game of GAN.

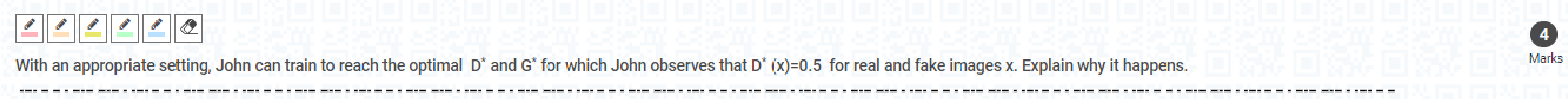
- The discriminator D aims to distinguish the real and fake examples, while the generator G aims to fool the discriminator D by generating realistic fake images indistinguishable from real images.

3

Marks

What are the optimization problems to train D and G?

$$\min_{G}\max_{\mathrm{D}} J(G,D) = \mathbb{E}_{\boldsymbol{x}\sim p_d(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\,[\log(1-D(G(\boldsymbol{z})))\,]$$

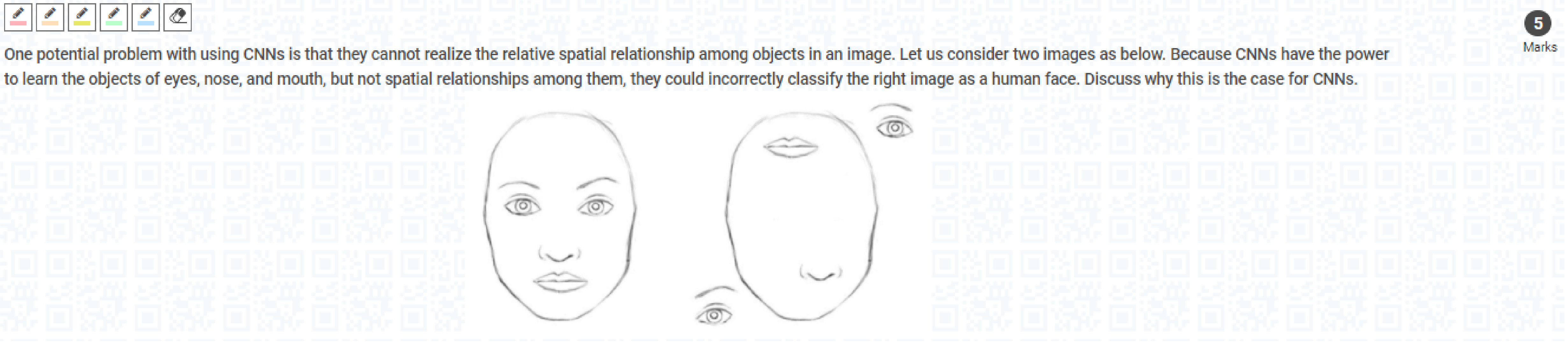


4

Marks

With an appropriate setting, John can train to reach the optimal D^* and G^* for which John observes that D^* (x)=0.5 for real and fake images x. Explain why it happens.

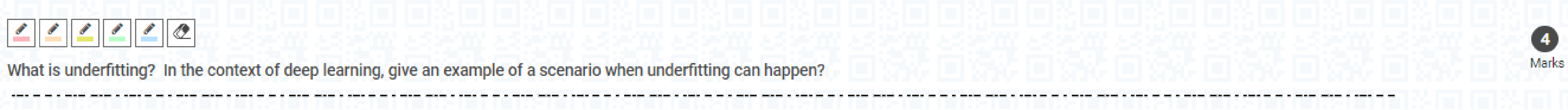
- The solution of the minimax problem $\boldsymbol{min_G max_D J(D,G)}$
  - Nash equilibrium point (D^*, G^*) which satisfies
    - $p_{g^*} = p_d$
    - $D^*(\boldsymbol{x}) = \frac{p_d(\boldsymbol{x})}{p_d(\boldsymbol{x})+p_{g^*}(\boldsymbol{x})} = 0.5, for\ all\ x$



5 Marks

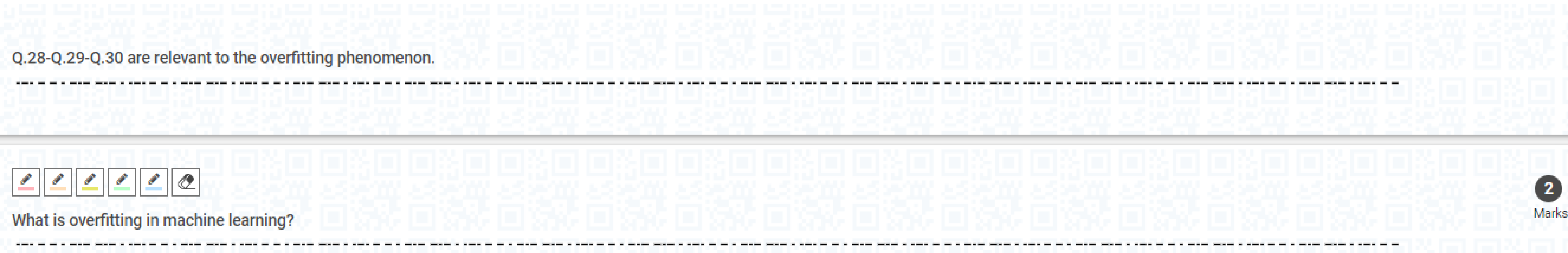
One potential problem with using CNNs is that they cannot realize the relative spatial relationship among objects in an image. Let us consider two images as below. Because CNNs have the power to learn the objects of eyes, nose, and mouth, but not spatial relationships among them, they could incorrectly classify the right image as a human face. Discuss why this is the case for CNNs.

- Low level filters can detect small details for example edges, lines, corners
- Higher level features combine low level features
- The context is expanded locally
  - Edges for lips → upper lip, lower lip → entire lip
- After being able to learn and detect local objects on an image (left eye, right eye, nose, and mouth), CNN has no way to expand their contexts to detect larger objects and realize the spatial relations among these local objects.



What is underfitting? In the context of deep learning, give an example of a scenario when underfitting can happen?

- The model family is overly simple to characterize the data. Underfitting happens when using too simple models to learn from more complex data.
- In deep learning, when using a feed forward NN without any activation function to learn from complex data, underfitting happens. The reason is that a feed forward NN without any activation function can only represent a linear function.
    - Need to give an example in the context of DL.



Q.28-Q.29-Q.30 are relevant to the overfitting phenomenon.

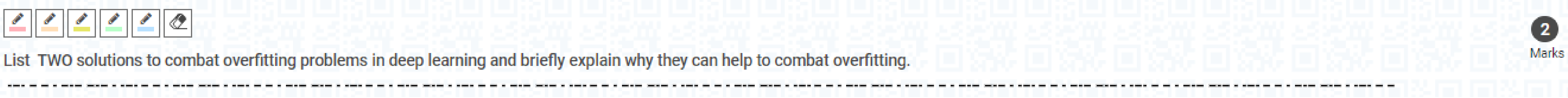
What is overfitting in machine learning?

- A model overlearns a training set and tends to use overly detailed characteristics and features of training examples to make prediction. Therefore, this model cannot generalize to predict well a separate testing set
  - Students can give the answers in different ways. As long as they indicate the characteristic of overfitting, the full marks are given.

2 Marks

In the context of deep learning, give an example of a scenario when overfitting can happen?

- Overfitting can happen in DL when we use a complex deep net with many layers and neurons per layer to learn from a relatively simpler training set.



2 Marks

List TWO solutions to combat overfitting problems in deep learning and briefly explain why they can help to combat overfitting.

- Data augmentation
- Use regularization term
- Dropout
- Batch norm
- Collect more data
- Data mix-up
- Label smoothing
- Just need to list two of them and give brief explanations.