IMPERIAL COLLEGE LONDON

TIMED REMOTE ASSESSMENTS 2020-2021

MEng Honours Degree in Electronic and Information Engineering Part IV
MEng Honours Degree in Mathematics and Computer Science Part IV
MEng Honours Degrees in Computing Part IV
MSc Advanced Computing
MSc Artificial Intelligence
MSc in Computing (Specialism)
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant assessments for the Associateship of the City and Guilds of London Institute

PAPER COMP70010=COMP97111=COMP97112

DEEP LEARNING

Tuesday 23 March 2021, 10:00
Duration: 105 minutes
Includes 15 minutes for access and submission

Answer ALL TWO questions
Open book assessment

This time-limited remote assessment has been designed to be open book. You may use resources which have been identified by the examiner to complete the assessment and are included in the instructions for the examination. You must not use any additional resources when completing this assessment.

The use of the work of another student, past or present, constitutes plagiarism. Giving your work to another student to use constitutes an offence. Collusion is a form of plagiarism and will be treated in a similar manner. This is an individual assessment and thus should be completed solely by you. The College will investigate all instances where an examination or assessment offence is reported or suspected, using plagiarism software, vivas and other tools, and apply appropriate penalties to students. In all examinations we will analyse exam performance against previous performance and against data from previous years and use an evidence-based approach to maintain a fair and robust examination. As with all exams, the best strategy is to read the question carefully and answer as fully as possible, taking account of the time and number of marks available.

Paper contains 2 questions

1 Convolutions

a i) Convolve the 2D input tensor with the 2D filter kernel shown in Figure 1a. Stride = 1. Use convolution, not cross-correlation!

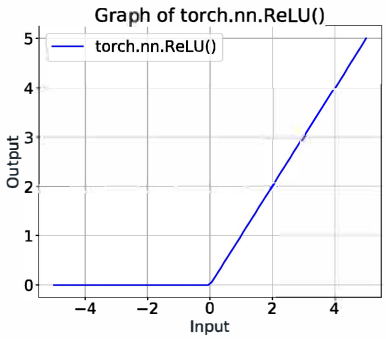
Input tensor

1	1	1	1	1	1
1	1	1	1	1	1
0	1	0	0	1	0
1	1	0	0	1	1
1	1	1	1	1	1
1	1	1	1	1	1

(a)

filter kernel

0	1	0
1	-2	1
0	0	0



(b)

Fig. 1: (a) 2D input tensor and 2D filter kernel, (b) activation function.

ii) Use the activation function in Figure 1b to activate the output of this convolution.

iii) Which of the following activation functions can lead to vanishing gradients?

- ☐ **ReLU**
- ☐ **Tanh**
- ☐ **Leaky ReLU**
- ☐ **Sigmoid**
- ☐ **SELU**

iv) What is the number of multiplication operations required to apply the filter and activation function from Figure 1 to the input tensor? (Assume stride=1, zero padding; count multiplications by zeros for simplicity).

v) What would you need to do to get the same input and output tensor shape?

vi) Suppose you have a single-channel input tensor of size $M \times N$ and a filter of size $m \times n$. Assume M, N, m, and n are odd numbers. What is the output tensor size if stride=1 and no zero padding is used?

vii) List two reasons why max pooling operation is often used in Convolutional Neural Networks.

b i) Given a layer in a Neural Network that processes a $64 \times 64 \times 1$ input tensor, how many parameters need to be learned for

A) a fully connected layer

B) a convolutional layer with kernel size 5×5

ii) A Convolutional Neural Network can extract latent feature representation tensors of shape (1024×1). How many training samples would at least be required to approximate a differentiable continuous function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ with this network to achieve an error as low as $\epsilon \sim 0.1$,?

- ☐ $\sim 10^{42}$
- ☐ $\sim 10^{10}$
- ☐ $\sim 10^{36000000}$
- ☐ $\sim 10^{78}$
- ☐ $\sim 10^{82}$

iii) Consider the binary classification problem in Figure 3 which is for a linear classifier impossible to solve.

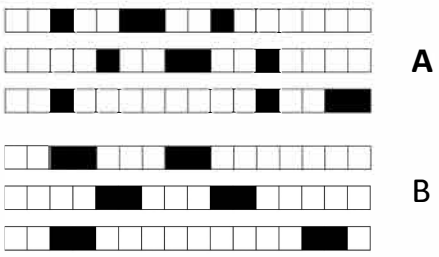


Fig. 2: binary classification problem

The training set consists of patterns A and B in all possible translations, warped around at the boundaries. Consider a neural network that consists of a 1D convolution layer with a linear activation function, followed by a linear layer with a logistic output. Can such an architecture perfectly classify all of the training examples? Why or why not?

iv) For small batch sizes, the number of iterations required to reach the target loss decreases as the batch size increases. Why is that?

v) For large batch sizes, the number of iterations does not change much as the batch size is increased. Why is that?

vi) Would you gain any regularization benefit from using Dropout and BatchNorm within the same network architecture? Why or why not?

c i) With a neural network we try to approximate the function $f(x)$. Which function does a residual neural network (ResNet) block approximate?

ii) List three advantages of the ResNet architecture.

iii) When using BatchNorm in a ResNet, what would be the effect of choosing a mini-batch size of 1.

The three parts carry, respectively, 40%, 40%, and 20% of the marks.

2 Generative models, RNNs and Attention

a Consider the latent variable model with the graphical model visualised below:

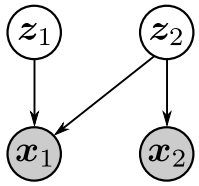


Fig. 1: Graphical model visualisation of $p(x_1, x_2, z_1, z_2)$

i) Write down the factorisation structure of $p(x_1, x_2, z_1, z_2)$ (in terms of product of conditional distributions).

ii) State the VAE objective by assuming the encoder as $q_\phi(z_1, z_2|x_1, x_2)$. Assume the data distribution is $p_{\text{data}}(x_1, x_2)$.

iii) Assume the q distribution is factorised as

$$q_\phi(z_1, z_2|x_1, x_2) = q_{\phi_1}(z_1|x_1, x_2)q_{\phi_2}(z_2|x_2).$$

Also assume sampling $z_1, z_2 \sim q$ can be done as $z_1 = T_{\phi_1}(x_1, x_2, \epsilon_1), \epsilon_1 \sim \pi_1(\epsilon_1)$ and $z_1 = T_{\phi_2}(x_2, \epsilon_2), \epsilon_2 \sim \pi_2(\epsilon_2)$. Then what are the gradients of the VAE loss with respect to ϕ_1 and ϕ_2?

iv) Which of the following statements are TRUE for VAEs?

☐ **The optimal encoder minimises** $\text{KL}[p(z|x)||q(z|x)]$**;**

☐ **To train the encoder with gradient ascent, the** $q(z|x)$ **must be reparameterisable;**

☐ **The KL term in the VAE loss is always regarded as a regulariser for the encoder;**

☐ **Reconstruction error is the best evaluation metric for a VAE's test performance;**

☐ **None of the above**

b Consider training a GAN for conditional image generation tasks:

i) Assume the data distribution is $p_{data}(x, y)$ where x represents an image and y represent its label. Define the GAN generator distribution as $p_\theta(x|y)p(y)$. How do you construct a discriminator and what is its optimal form?

ii) What is the optimal generator $p_\theta(\boldsymbol{x}|y)$ when the adversarial training reaches an equilibrium (assume all the networks in the conditional GAN have infinite capacity)?

iii) Which of the following statements are FALSE?

☐ **The original GAN objective in Goodfellow et al. (2014) is equivalent to minimising the Jensen-Shannon divergence for the generator;**

☐ **With the original GAN loss, saturated gradients occur for the generator only when the fake images look similar to the real ones;**

☐ **Normalisation layers are often used to stabilise GAN training;**

☐ **For GAN losses in general, we can swap the ordering of min-max to max-min;**

☐ **None of the above**

iv) State in detail the main differences between VAEs and GANs.

c Consider the following RNN to process the input sequence $\boldsymbol{x}_{1:T}$:

$$\boldsymbol{h}_t = \sigma(W_h \boldsymbol{h}_{t-1} + W_x \boldsymbol{x}_t + \boldsymbol{b}_h), \quad t = 1, ..., T$$

with the convention that $\boldsymbol{h}_0 = \mathbf{0}$, and $\sigma(\cdot)$ is the activation function. Assume the final output is $\boldsymbol{y} = \boldsymbol{h}_T$ and there is a loss function $\mathcal{L}(\boldsymbol{y})$ applied to it.

i) Derive the gradient of $\mathcal{L}(\boldsymbol{y})$ with respect to W_h.

ii) Now assume $\sigma(\cdot)$ is identity mapping, $\boldsymbol{h}_t \in \mathbb{R}^{3\times 1}$ and the recurrent weight matrix is

$$W_h = \begin{pmatrix} -0.84 & -0.26 & -1.32 \\ -0.76 & 0.08 & -0.96 \\ 0.42 & -1.86 & -0.4 \end{pmatrix}.$$

State whether gradient vanishing/explosion would happen when $T \to +\infty$ and explain why.

iii) Which of the following solutions can help address the gradient vanishing/explosion problem in practice?

☐ **Truncated back-propagation through time**

☐ **Use gates for the recurrent units**

☐ **Use ReLU activations**

☐ **Use layer normalisation**

☐ **Use orthonormal matrix for W_h**

iv) Consider building a Sequence-to-Sequence model with encoder/decoder LSTMs. Explain why and how this model can handle input/output sequences of arbitrary length.

d Attention mechanisms & Transformers:

i) Consider scaled dot-product hard attention: compute its output using the following inputs:

$$Q = \begin{pmatrix} -2 & 1 & 1 & -3 \\ -3 & 0 & -3 & -3 \end{pmatrix}, \quad K = \begin{pmatrix} -3 & 2 & -2 & -5 \\ -3 & 4 & 1 & -4 \\ -1 & 2 & 3 & -2 \end{pmatrix}, \quad V = \begin{pmatrix} -2 & -2 \\ 2 & 3 \\ 0 & 2 \end{pmatrix}.$$

ii) Assume the query vectors are column vectors of D dimensions. To achieve the same order of time complexity when compared with single-head attention, what is the dimensionality of the query projection matrix in multi-head attention (if using H heads)?

iii) Assume we want to build a Transformer-like neural network to process images. In this case an input image of size $H \times W$ is viewed as a collection of small patches of size $h \times w$. Assume no padding is used, and the small patches are overlapping with stride 1. What is the time complexity of computing the dot product for self-attention on these patches?

iv) Which of the following statements are TRUE for position encoding?

- ☐ **We need to use position encoding because dot-product attention is permutation invariant;**
- ☐ **When applied to spatial indices, different dimensions in the coordinates need to be encoded separately;**
- ☐ **Learnable position encoding can only be used if the maximum index is known;**
- ☐ **Position encodings can be added or concatenated to the original query inputs;**
- ☐ **The sinusoid encoding with fixed output dimensionality is a one-to-one mapping;**

The four parts carry, respectively, 25%, 25%, 30%, and 20% of the marks.