IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2019

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 in Advanced Computing
MSc in Computing Science (Specialist)
for Internal Students of the Imperial College of Science, Technology and Medicine

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

PAPER C460

DEEP LEARNING

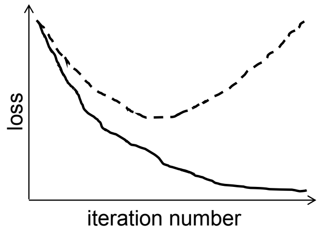
Wednesday 20th March 2019, 14:00
Duration: 120 minutes

Answer THREE questions

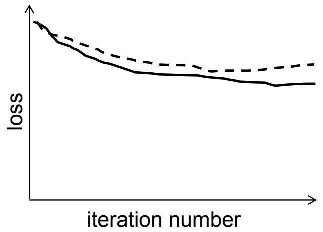
Paper contains 4 questions
Calculators not required

1 Basics of machine learning and neural networks

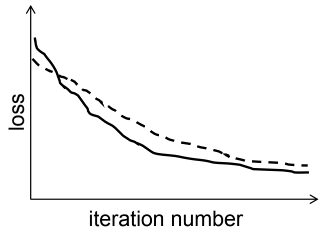
a Suppose you train three different neural network architectures on the same dataset and observe the following qualitative behavior of the training loss (solid curve) and validation loss (dashed curve):



Model A



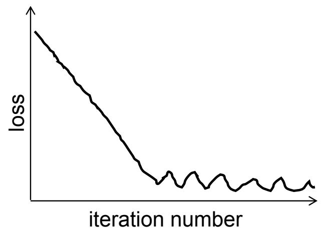
Model B



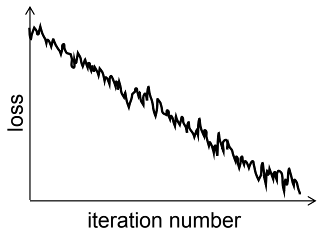
Model C

Characterize the behavior of each of the models. Which of the models would generalize well on test data?

b Suppose you are training a neural network on the same data using different settings and observe the following different behaviors. Assume that the loss (y-axis) is plotted on log scale:



Setting A



Setting B

i) Which of the settings is likely using batch mode gradient descent and which mini-batch (stochastic) gradient descent? Explain.

ii) What is the order of convergence rate in the beginning of the training? Explain.

iii) Which of the models is likely to have a larger learning rate? Explain.

iv) What is the likely reason for loss function oscillations in setting A? Explain.

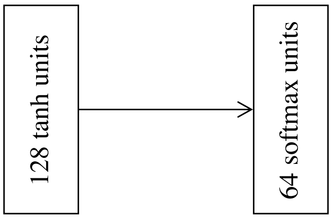
c You have a neural network that uses tanh as activation functions. A modification of this architecture is proposed in which the tanh is replaced with ReLU.

i) Assuming the same optimization settings are used, is the new architecture likely to train faster? Explain.

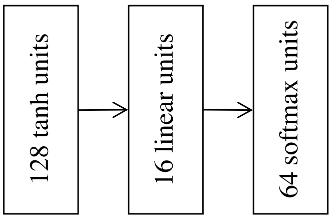
ii) What is the vanishing gradient phenomenon and how the new architecture may help to solve it? Explain.

iii) Provide another way of dealing with the vanishing gradient problem.

d Consider the following two neural network models:



Model A



Model B

i) Which model is more general (i.e. can approximate a larger class of functions)?

ii) What is the computational complexity in terms of multiplication operations of model A and model B? Ignore the cost of computing the activations.

iii) Which model has more parameters?

iv) Describe another advantage of Model B.

The four parts carry, respectively, 10%, 25%, 25%, and 40% of the marks.

2 Convolutional neural networks

a Suppose you have the following single-channel image:

1	0	1	2	1	0	0
0	1	3	4	3	1	1
1	1	4	5	4	1	1
2	4	6	6	6	5	1
2	6	7	7	6	5	1
1	2	5	6	6	2	1
1	1	2	3	2	1	1

i) Compute the result of max pooling of size 3×3 with stride=2.

ii) Suppose you are given the following 3×3 filter:

0	1	0
1	-2	0
0	1	0

Compute the result of convolution of the input image with this filter using stride=2, no zero padding.

b Suppose you have a single-channel image of size $M \times N$ and a filter of size $m \times n$. Assume M, N, m, and n are odd numbers.

i) What is the output image size if stride=1 and no zero padding is used?

ii) What is the output image size if stride=2 and zero padding is used?

iii) What is the number of multiplication operations required to apply the filter to the image? (Assume stride=1, zero padding; count multiplications by zeros for simplicity).

iv) The $m \times n$ 2D filter is now implemented as a separable filter, consisting of a pair of 1D filters: first, apply a vertical $m \times 1$ filter and then, apply a horizontal 1 × n filter. What is the number of multiplication operations required in this case, under the same assumptions?

v) Would the filtering result be the same if we first apply the horizontal filter and then the vertical filter instead of first applying the vertical filter and then the horizontal filter? Explain.

vi) Suppose now that the first 1D filter is followed by a non-linear activation function, and then the second 1D filter is applied. Would the filtering result still be the same if the order of the horizontal and vertical filters is exchanged? Explain.

vii) Is it always possible to write a 2D filter in a separable form as two 1D filters? Explain.

viii) For each of the 3x3 2D filters below, write its separable version (3x1 vertical and 1x3 horizontal filter). If you believe that some of the filters below cannot be written in this form, explain why.

$$H_A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad H_B = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad H_C = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 4 & 1 \end{bmatrix}$$

c i) List two reasons why max pooling operation is often used in CNNs.

ii) Suppose you are given two alternative CNN architectures:

Model A has three convolutional layers with filters of size 3 × 3 with stride=1 and ReLU activation; each convolutional layer is followed by 3 × 3 max pooling with stride=2.

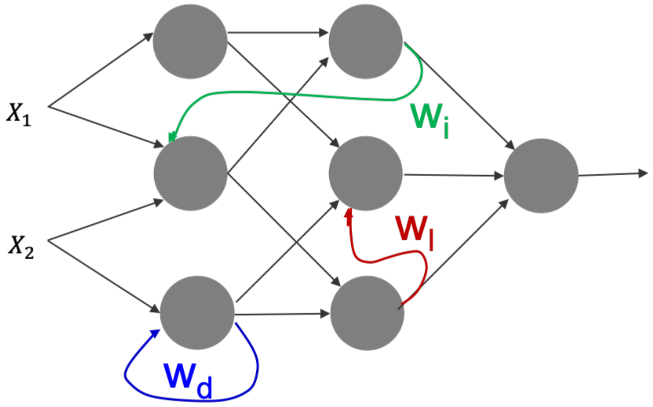
Model B (called 'all convolutional') has three convolutional layers with filters of size 3 × 3 with stride=2 and ReLU activation (no max pooling).

Explain two advantages of Model B compared to Model A.

The three parts carry, respectively, 20%, 60%, and 20% of the marks.

3 Recurrent Neural Networks

a The pictures below shows a Recurrent Neural Network (RNN) with the two input features x_1 and x_2 and the weights w_i, w_l, w_d.



i) How is the connection with weight w_d called? What is its function?

ii) How is the connection with weight w_i called? What is its function and consequence?

iii) How is the connection with weight w_l called? What is its function and the consequence?

b Recurrent Neural Networks can be trained via Backpropagation Through Time (BPTT).

i) Say in a few words what happens in the BPTT forward pass.

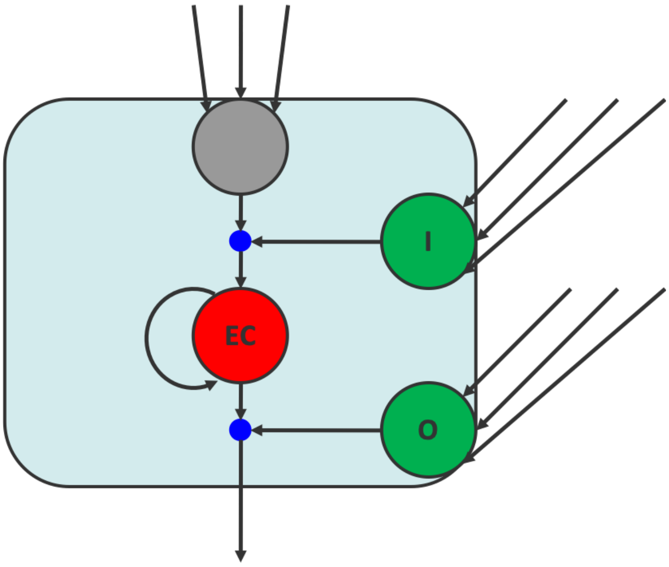
ii) Say in a few words what happens in the BPTT backward pass.

iii) What leads potentially to vanishing and exploding gradients in BPTT.

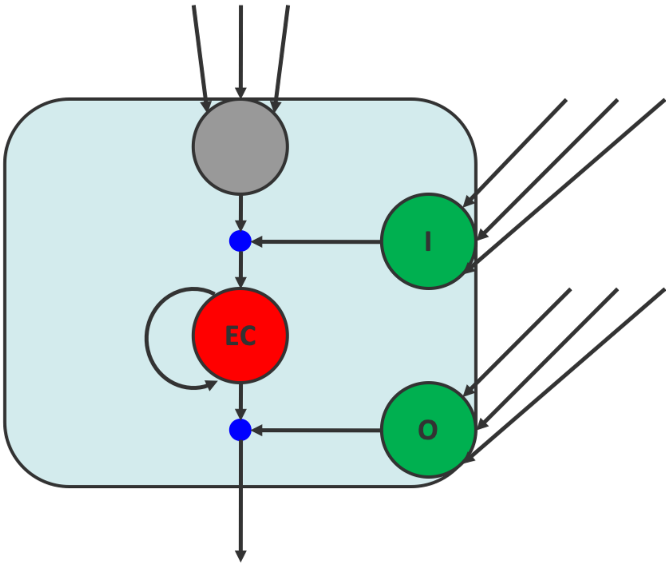
iv) Name a method that designed such that the gradients do not decay in BPTT.

c The pictures below each show a Long Short-Term Memory (LSTM) cell with an input (I) and output (O) gate as well as the (constant) error carousal (EC).

i) Draw the peep-hole connections into the cell and explain their function.



ii) Draw the forget gate (F) into the picture and explain why it is needed.



iii) Which gate of the above named gates I, O, F is missing in a Gated Recurrent Unit (GRU)? What is the immediate consequence and advantage coming from this?

d Connectionist Temporal Classification (CTC) can be applied to address temporal classification problems without the need for frame-level alignments, and normally, the output sequence is much shorter than the input sequence.

i) Name two main reasons to avoid manual pre-segmentation of temporal data.

ii) CTC introduces a special symbol - blank (“”) - as an additional label, meaning no (actual labels) are assigned to a frame. What would be the CTC output of the sequence I I C C C L?

iii) How can CTC decoding be improved? Name one method and describe their principle mechanism.

The four parts carry, respectively, 25%, 20%, 35%, and 20% of the marks.

4 Generative Models

a i What is the difference between Variational Auto-Encoders (VAE) and Generative Adversarial Networks (GANs) in terms of the structure and the loss function they optimise (Please write down the loss function of an VAE and a GAN)?

ii We want to generate samples from a set of predefined classes (e.g., different types of chairs) using a single generative model. What kind of GAN would be suitable? What would be the loss functions for the Generator and the Discriminator?

b Assume a GAN that comprises of a generator and a discriminator. Assume also that the GAN optimises the binary classification cross-entropy loss for real and fake (i.e., produced by the generator) images. Prove that for a fixed Generator the optimal Discriminator $D^*(\mathbf{x})$ is

$$D^*(\mathbf{x}) = \frac{p_d(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x})}$$

where $p_d(\mathbf{x})$ is the true data distribution and $p_g(\mathbf{x})$ is the distribution of the generated samples. What is the value of the discriminator for the optimal generator?

The two parts carry equal marks.