UNIVERSITY OF BIRMINGHAM

School of Computer Science

Neural Computation

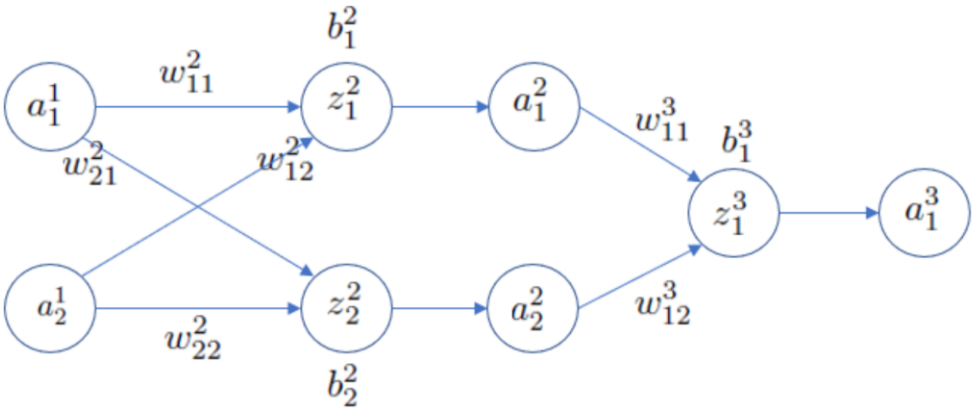
Mock Examination 2022

Total time: 60 minutes

Answer ALL questions. In the mock exam, each question is marked out of 10. The mock exam paper will be marked out of 30, which will be rescaled to a mark out of 100. The mock exam is scheduled for 60 minutes while the real exam will be scheduled for 120 minutes.

Q1: NN, GD, Optimisation

Let us consider solving regression problems with a neural network. In particular, we consider a neural network of the following structure:



As illustrated in the lecture, we have the following relationship between variables in the neural network.

$$\mathbf{z}^2 = \begin{pmatrix} z_1^2 \\ z_2^2 \end{pmatrix} = \begin{pmatrix} \omega_{11}^2 & \omega_{12}^2 \\ \omega_{21}^2 & \omega_{22}^2 \end{pmatrix} \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix} + \begin{pmatrix} b_1^2 \\ b_2^2 \end{pmatrix}, \quad \mathbf{a}^2 = \begin{pmatrix} a_1^2 \\ a_2^2 \end{pmatrix} = \begin{pmatrix} \sigma(z_1^2) \\ \sigma(z_2^2) \end{pmatrix},$$

where σ is the activation function. For simplicity of computation, we always use $\sigma(x) = x^2$ in this neural network. In a similar way, there is also a relationship between z_1^3, a_1^3 and $\mathbf{a}^2$.

(a)

Compute the number of trainable parameters required in determining this neural network. Please explain your answer. **[3 marks]**

(b)

Suppose

$$\mathbf{W}^2 = \begin{pmatrix} \omega_{11}^2 & \omega_{12}^2 \\ \omega_{21}^2 & \omega_{22}^2 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad \mathbf{W}^3 = (\omega_{11}^3, \omega_{12}^3) = (1, 1),$$

$$\mathbf{b}^2 = \begin{pmatrix} b_1^2 \\ b_2^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad b_1^3 = -3.$$

Consider the training example

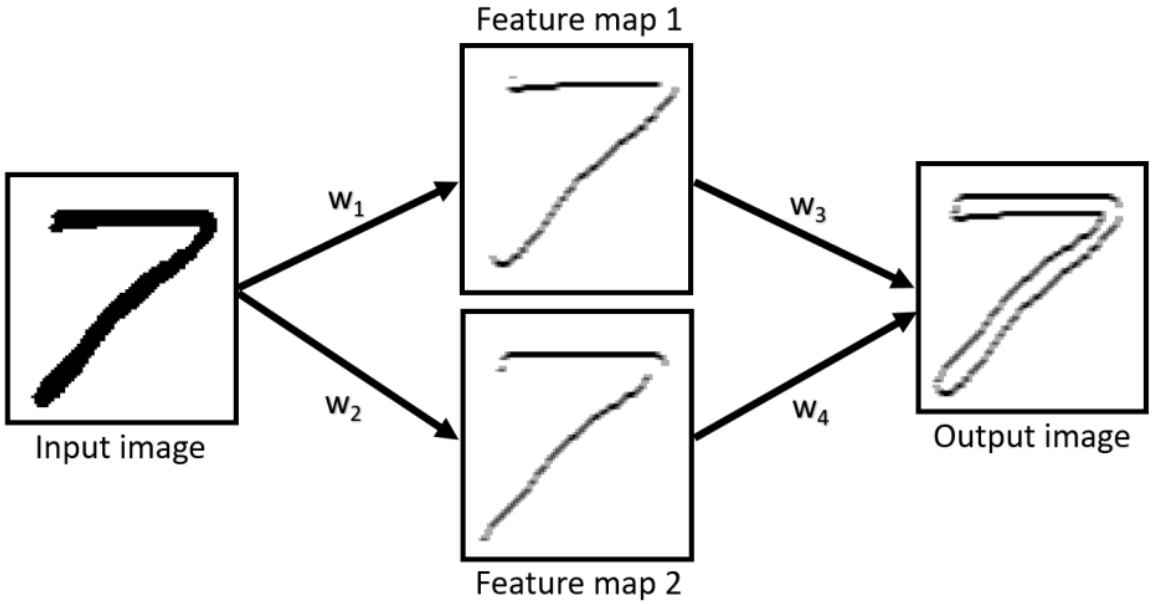
$$\mathbf{x} = \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad y = 1.$$

Let us consider the square loss function $C_{\mathrm{x},y}(\mathbf{W}, \mathbf{b}) = \frac{1}{2}(a_1^3 - y)^2$, where $\mathbf{W} = \{\mathbf{W}^2, \mathbf{W}^3\}, \mathbf{b} = \{\mathbf{b}^2, b_1^3\}$. Use the forward propagation algorithm to compute $\mathbf{a}^2$, a_1^3 and the loss $C_{\mathrm{x},y}(\mathbf{W}, \mathbf{b})$ for using the neural network to do prediction on the above example $(\mathbf{x}, y)$. Please write down your step-by-step calculations. **[7 marks]**

Q2: CNN, RNN, LSTM

(a)

A CNN is shown in the figure below. We use the nonlinear ReLU activation function in the first layer and the linear activation function in the output layer. Note that for better visualisation, in the images we use white regions to denote 0 and darker regions to denote larger values.



Design appropriate convolution kernels of size 3×3 for the first layer such that the feature maps 1 and 2 are these displayed in the figure. Please justify your answer. **[5 marks]**

(b)

We apply N convolutional kernels with stride=1 and padding = 0 to a 11 by 11 colour image, which results in a 5 by 5 feature map, with 10 channels. We then apply M convolutional kernels of size (H, W, D) to this feature map with stride = 2 and padding = 1. This results in a 4 by 4 output, with 5 channels. Please identify the values for (N, M, H, W, D). Please justify your answer. **[5 marks]**

Q3: VAE, AE, GAN

(a)

Consider the Variational Auto-Encoder (VAE), with encoder $f_\phi(x)$ that predicts mean $\mu_\phi(x)$ and standard deviation $\sigma_\phi(x)$ of a multi-dimensional Gaussian that is the conditional $p_\phi(z|x)$, and decoder $g_\theta(z)$. The VAE's loss for each d-dimensional input vector x is:

$$\mathcal{L}_{VAE} = \lambda_{rec}\mathcal{L}_{rec}(x) + \lambda_{reg}\mathcal{L}_{reg}(x), \tag{2}$$

where $\mathcal{L}_{rec}(x) = \frac{1}{d}\sum_{j=1}^{d}(x^{(j)} - g_\theta^{(j)}(\tilde{z}))^2$ for sample $\tilde{z} \sim p_\phi(z|x)$ is reconstruction loss, $\mathcal{L}_{reg}(x) = \frac{1}{2}\sum_{j=1}^{v}\left[(\mu_\phi^{(j)}(x))^2 + (\sigma_\phi^{(j)}(x))^2 - 2\log_e \sigma_\phi^{(j)}(x) - 1\right]$ is the regularizer, z is a v-dimensional vector and $\log_e$ is the natural logarithm. λ_{rec}, λ_{reg} are non-trainable scalars for weighting $\mathcal{L}_{rec}$ and $\mathcal{L}_{reg}$. $h^{(j)}$ denotes the j-th element of vector h.

Assume you are given an implementation of the above VAE with a bottleneck (i.e. $v < d$). You are asked to train the VAE so that it will be as good as possible for the task of compressing data (via bottleneck) and uncompressing them with fidelity. Generation of fake data or other applications are not of interest. What values would you choose for λ_{rec} and λ_{reg}? For each, specify either *equal to* 0 or *greater than* 0. Explain why. **[5 marks]**

(b)

Consider a Generative Adversarial Network (GAN) that consists of Generator G that takes as input noise vector z, and of a Discriminator D that given input x it outputs $D(x)$. We assume that value $D(x) = 1$ means that D predicts with certainty that input x is a real data point, and $D(x) = 0$ means D predicts with certainty that x is a fake, generated sample.

Assume that at the beginning of training, parameters of G and D are initialized randomly. Then, D is trained for few SGD iterations, while G remains fixed (untrained). After the few updates to D's parameters, is the value D(G(z)) likely to be closer to 0 or 1? Explain why. **[5 marks]**