UNIVERSITY OF BIRMINGHAM

School of Computer Science

Third Year Undergraduate

06-32167

32167 LH Neural Computation

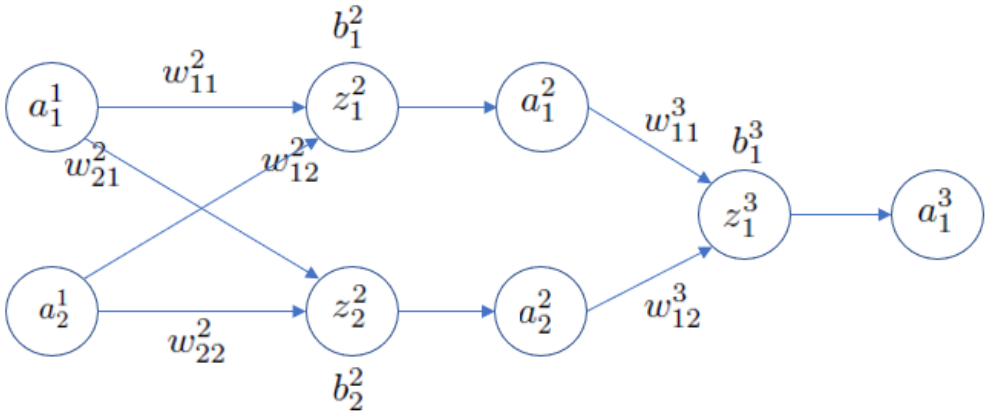
Main Summer Examinations 2022

[Answer all questions]

32167 LH Neural Computation

Question 1

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]**

(c) Let us consider the neural network with the above $\mathbf{W}^2, \mathbf{W}^3, \mathbf{b}^2, b_1^3$ and the above training example $\mathbf{x}, y$. Use the back propagation algorithm to compute the gradients. For simplicity, we only require you to compute the explicit number of

$$\frac{\partial C_{\mathrm{x},y}(\mathbf{W},\mathbf{b})}{\partial z_1^3}, \quad \frac{\partial C_{\mathrm{x},y}(\mathbf{W},\mathbf{b})}{\partial z_1^2}, \quad \frac{\partial C_{\mathrm{x},y}(\mathbf{W},\mathbf{b})}{\partial z_2^2}, \quad \frac{\partial C_{\mathrm{x},y}(\mathbf{W},\mathbf{b})}{\partial \omega_{11}^3}, \quad \frac{\partial C_{\mathrm{x},y}(\mathbf{W},\mathbf{b})}{\partial \omega_{12}^3}.$$

Please write down your step-by-step calculations. **[10 marks]**

Question 2

Given the weights (w_1, w_2, w_3) and the biases (b_2, b_3), we have the following recurrent neural network (RNN) which takes in an input vector x_t and a hidden state vector h_{t-1} and returns an output vector y_t:

$$y_t = \mathbf{g}(w_3\mathbf{f}(w_1 x_t + w_2 h_{t-1} + b_2) + b_3), \tag{1}$$

where $\mathbf{g}$ and $\mathbf{f}$ are activation functions. The following computational graph depicts such a RNN.

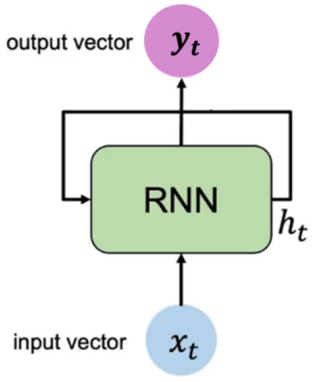


Figure 1: RNN Computational Graph

(a) Write down clearly which part of Equation (1) defines the current (updated) hidden state vector h_t shown in Figure 1. **[3 marks]**

(b) When $t = 3$ (starting from 1), please show how information is propagated through time by drawing an unfolded feedforward neural network that corresponds to the RNN in Figure 1. Please make sure that hidden states, inputs and outputs as well as network weights and biases are annotated on your network. **[4 marks]**

(c) Assume x_t, h_{t-1}, h_t and y_t are all scalars in Equation (1), and the activation functions are a linear unit and a binary threshold unit, respectively defined as:

$$\mathbf{g}(x) = x,$$
$$\mathbf{f}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}.$$

When $t = 3$ (starting from 1), please calculate the values of the outputs (y_1, y_2, y_3) given $(w_1 = 1, w_2 = -3, w_3 = 5)$, $(b_2 = 1, b_3 = 3)$, $(x_1 = 5, x_2 = 3, x_3 = 1)$ and $h_0 = 0$. Please show your calculations in detail. **[3 marks]**

(d) Again let us assume x_t, h_{t-1}, h_t and y_t are all scalars with $h_0 = 0$ and the activation functions the same as above. Compute (w_1, w_2, w_3) and (b_2, b_3) such that the network outputs 0 initially, but when it receives an input of 1, it outputs 1 for all subsequent time steps. For example, if the input is 00001000100, the output will be 00001111111. Please justify your answer.

Note: here we want a solution that satisfies (1) the hidden state h_t is zero until the input x_t becomes 1, at which point the hidden state changes to 1 forever, and (2) the output always predicts the same as the hidden state, i.e. $y_t = h_t$. **[10 marks]**

Question 3

Note: Each item below can be answered with approximately 5 lines of text. This is only an informal indication, not a hard constraint. Length of answers will not influence marks.

(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.

(i) If you train minimizing only the regularizer $\mathcal{L}_{reg}$ (i.e. $\lambda_{rec} = 0$ and $\lambda_{reg} > 0$), what values do you expect the encoder will tend to predict for means $\mu_\phi(x)$ and standard deviations $\sigma_\phi(x)$? Explain why. **[4 marks]**

(ii) Assume that z is 2 dimensional (i.e. $v = 2$). Assume that for an input data point x_1 the encoder outputs vectors $\mu_\phi(x_1) = (0.5, 0.1)$ and $\sigma_\phi(x_1) = (0.1, 0.3)$. Calculate the value of $\mathcal{L}_{reg}(x_1)$. Show the steps of the calculation. (Note: For simplicity, use $\log_e 0.1 \approx -2.3$ and $\log_e 0.3 \approx -1.2$) **[4 marks]**

(iii) 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 a Generator G that takes input noise vector z and outputs $G(z)$, and Discriminator D that given input x it outputs $D(x)$. We assume that $D(x) = 1$ 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. Figure 2 shows two loss functions that could be used for training G. Which of the two loss functions in Figure 2 is more appropriate for training G in practice? Explain why, based on the gradients for lowest and highest values of $D(G(z))$ and how they would influence training. **[7 marks]**

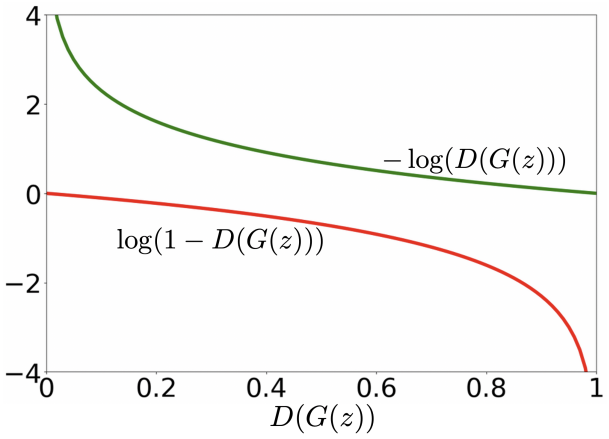


Figure 2: Two loss functions that could be used (minimized) for training Generator G of a GAN. Shown as a function of Discriminator's D predicted probability that a generated sample $G(z)$ is real. On the x-axis, $D(G(z)) = 1$ means D predicts with certainty that $G(z)$ is real, whereas $D(G(z)) = 0$ means D predicts with certainty that $G(z)$ is fake.