Neural Computation

Solutions

Mock Examination 2021

Neural Computation

Note that there are 60 points in total for the mock (and final real) exam, which will be rescaled to 80% in the end.

Question 1

Consider a set of two input-output pairs $\left(\binom{1}{2}, 1\right), \left(\binom{-1}{2}, -1\right)$. Let the loss function be the least square and we wish to find a linear model $\mathbf{x} \mapsto \mathbf{x}^\top \mathbf{w}$, where $\mathbf{x}^\top$ is the transpose of $\mathbf{x} \in \mathbb{R}^2$ and $\mathbf{w} = \binom{w_1}{w_2} \in \mathbb{R}^2$. Then the objective function becomes

$$C(\mathbf{w}) = \frac{1}{2}\left(\frac{1}{2}\Big((1,2)\mathbf{w} - 1\Big)^2 + \frac{1}{2}\Big((-1,2)\mathbf{w} + 1\Big)^2\right).$$

Let us build our prediction model by minimizing the above objective function.

(a) Simplify the objective function to the form of

$$C(\mathbf{w}) = c_1 w_1^2 + c_2 w_2^2 + c_3 w_1 + c_4 w_2 + c_5 w_1 w_2 + c_6,$$

where $c_k \in \mathbb{R}$ are coefficients, and w_i is the i-th coordinate of $\mathbf{w} \in \mathbb{R}^2$. After that, compute the gradient of $C(\mathbf{w})$ in terms of c_k.

[8 marks]

(b) Consider gradient descent with the initial point $\mathbf{w}^{(0)} = \binom{3}{1}$ and learning rate $\eta = 0.5$. Write down the process of calculating $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$. After that, calculate $C(\mathbf{w}^{(1)})$ and $C(\mathbf{w}^{(2)})$.

[6 marks]

(c) Consider gradient descent with momentum. Assume $\mathbf{w}^{(0)} = \binom{3}{1}$, learning rate $\eta = 0.5$ and momentum rate $\alpha = 0.5$. Write down the process of calculating $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$. After that, calculate $C(\mathbf{w}^{(1)})$ and $C(\mathbf{w}^{(2)})$.

[6 marks]

Model answer / LOs / Creativity:

(a) It is clear that

$$\begin{aligned} 4C(\mathbf{w}) &= \big(w_1 + 2w_2 - 1\big)^2 + \big(-w_1 + 2w_2 + 1\big)^2 \\ &= w_1^2 + 4w_2^2 + 1 - 2w_1 - 4w_2 + 4w_1 w_2 + w_1^2 + 4w_2^2 + 1 - 2w_1 - 4w_1 w_2 + 4w_2 \\ &= 2w_1^2 + 8w_2^2 - 4w_1 + 2. \end{aligned}$$

Therefore

$$C(\mathbf{w}) = \frac{1}{2}w_1^2 + 2w_2^2 - w_1 + \frac{1}{2} \quad \text{and} \quad \nabla C(\mathbf{w}) = \begin{pmatrix} w_1 - 1 \\ 4w_2 \end{pmatrix}.$$

- Type of question: Creative
- Learning outcomes: 2 and 3.
- Marking scheme: 4 marks for the simplification and 4 marks for the gradient.

(b) For the gradient descent, we know

$$\mathbf{w}^{(1)} = \mathbf{w}^{(0)} - 0.5\nabla C(\mathbf{w}^{(0)}) = \begin{pmatrix} 3 \\ 1 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \Longrightarrow C(\mathbf{w}^{(1)}) = 5/2,$$
$$\mathbf{w}^{(2)} = \mathbf{w}^{(1)} - 0.5\nabla C(\mathbf{w}^{(1)}) = \begin{pmatrix} 2 \\ -1 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 1 \\ -4 \end{pmatrix} = \begin{pmatrix} 1.5 \\ 1 \end{pmatrix} \Longrightarrow C(\mathbf{w}^{(2)}) = 17/8.$$

- Type of question: Creative
- Learning outcomes: 2 and 3.
- Marking scheme: 2 marks for $w^{(1)}$, 2 marks for $w^{(2)}$ and 2 marks for J.

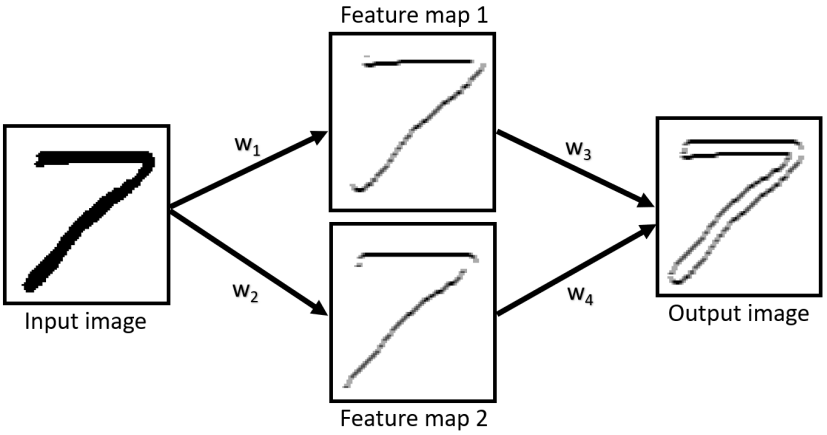
(c) For the gradient descent with momentum, we know

$$\mathbf{v}^{(1)} = \alpha\mathbf{v}^{(0)} - 0.5\nabla C(\mathbf{w}^{(0)}) = -\frac{1}{2}\begin{pmatrix} 2 \\ 4 \end{pmatrix}$$
$$\mathbf{w}^{(1)} = \mathbf{w}^{(0)} + \mathbf{v}^{(1)} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \Longrightarrow C(\mathbf{w}^{(1)}) = 5/2,$$
$$\mathbf{v}^{(2)} = \alpha\mathbf{v}^{(1)} - 0.5\nabla C(\mathbf{w}^{(1)}) = \frac{1}{2}\begin{pmatrix} -1 \\ -2 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 1 \\ -4 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$
$$\mathbf{w}^{(2)} = \mathbf{w}^{(1)} + \mathbf{v}^{(2)} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Longrightarrow C(\mathbf{w}^{(2)}) = 0$$

- Type of question: Creative
- Learning outcomes: 2 and 3.
- Marking scheme: 2 marks for $w^{(1)}$, 2 marks for $w^{(2)}$ and 2 marks for J.

Question 2

Answer the following questions related to convolutional neural networks (CNNs):



(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.

(i) 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]**

(ii) Design appropriate convolution kernels of size 3×3 for the output layer such that the output is that 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]**

(c) In CNNs, apart from ReLU there exist other nonlinear activation functions such as Sigmoid and Tanh. For the collection of neurons in a single layer, what would be the Jacobian matrices of the Sigmoid and Tanh functions, respectively. The Jacobian matrix is defined as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix},$$

Where f_i ($i = \{1, .., m\}$) denote the nonlinear activate function (Sigmoid or Tanh) and x_j ($j = \{1, .., n\}$) the neurons (variables) fed to that nonlinearity. Note: you can simplely use Sigmoid′ and Tanh′ to denote the derivatives of Sigmoid and Tanh, respectively. **[5 marks]**

Model answer / LOs / Creativity:

(a) (i)

$$W_1 = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \text{ and } W_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

One of kernels should detect dark/light horizontal boundaries, while the other should detect light/dark horizontal boundaries. It does not matter which one is W_1 or W_2. One kernel should have a positive gradient in the up-down direction while another kernel should have a negative gradient in the up-down direction.

- Type of question: Creative
- Learning outcomes: 2 and 3.
- Marking scheme: 2.5 marks for W_1 and 2.5 marks for W_2. 1.5 marks for one kernel that has a positive gradient in the up-down direction and 1.5 mark another kernel that has a negative gradient in the up-down direction.

(ii)

$$W_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \text{ and } W_4 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Two kernels that add the feature maps from the previous layer.

- Type of question: Creative
- Learning outcomes: 2 and 3.
- Marking scheme: 2.5 marks for W_3 and 2.5 marks for W_4.

(b) N = 10, M = 5, H2 = 1, W2 = 1, D2 = 10

- Type of question: Bookwork
- Learning outcomes: 1, 4.
- Marking scheme: 1 point for each value being correct.

(c) Both Jacobian matrices should be a diagonal matrix. For Sigmoid, the diagonal entries are Sigmoid'(x_i), where x_i denote input neurons. For tanh, the diagonal entries are tanh'(x_i).

- Type of question: Bookwork
- Learning outcomes: 2, 3.
- Marking scheme: 2.5 marks Sigmoid'(x_i) and 2.5 marks for tanh'(x_i)

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 standard Variational Auto-Encoder (VAE), with encoder $f_\phi(x)$ parameterized by ϕ that predicts mean $\mu_\phi(x)$ and standard deviation $\sigma_\phi(x)$ of a multi-dimensional Gaussian that is the conditional (posterior) $p_\phi(z|x) = N\left(\mu_\phi(x), \sigma^2_\phi(x)\right)$, and a decoder $g_\theta(z)$ parameterized by θ.
Assume that half-way through VAE training we process input x_1 and get encoder outputs $\mu_\phi(x_1) = (0.1, 0.2)$, and $\sigma_\phi(x_1) = (0.05, 0.2)$. Using the re-parameterization trick, we sample code $\tilde{z}$ to give as input to the decoder. What value of $\tilde{z}$ is most likely to give the best reconstruction of x_1? Explain why. **[4 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.

(i) 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]**

(ii) After the whole training process of the GAN has finished, assume that G has been optimized ideally. What would be the most likely value for D(G(z))? Explain why. **[4 marks]**

(c) Assume you are a Machine Learning Engineer. You are given a large database of photos of objects (all from same data distribution). You wish to create an object classifier based on neural-networks. The object class is labelled only on a few of the images. Assume the number of unlabelled data is high (no possible overfit). You are instructed to train an unsupervised model on the unlabelled data, and afterwards use its trained parameters to initialize a classifier, which you can then refine with supervised learning on the few labelled images. You can choose between a basic Auto-Encoder (AE), a Variational Auto-Encoder (VAE) and a Generative Adversarial Network (GAN) (basic versions taught).
What model would you choose? Explain why the other two are suboptimal. **[7 marks]**

Model answer / LOs / Creativity:

(a) The mean μ_ϕ predicted by the encoder. Explanation 1: According to the posterior $p_\phi(z|x)$ by encoder, the mean is the most probably code for input x, therefore will give the reconstruction that best matches x. Explanation 2: The mean is the z

that will be sampled the most during training when performing a forward pass on x, and therefore the value that will be most commonly reconstructed as x. **[4 marks]**

- Type of question: Creative
- Learning outcomes: 1 and 2.
- Marking scheme: 1 mark for correct answer about mean. Full marks for correct explanation (either of 2).

(b) (i) The value is likely to be closer to 0. This is because G with random weights will produce terrible images. D will easily learn to separate bad fakes from real examples, even with a few SGD iterations, therefore predicting $D(G(z)) \approx 0$. **[5 marks]**

- Type of question: Creative
- Learning outcomes: 1 and 2.
- Marking scheme: 1 mark for answering close to 0. 2 marks for explaining that at start of training G will produce bad images, 2 marks for explaining D will easily learn to identify the fakes.

(ii) The ideal value would be $D(G(z)) = 0.5$. Explanation 1: Ideally, G has learned to generated data that are are perfectly realistic. Then D cannot distinguish between real and fake data and its accuracy is chance, 50%. Explanations 2: It has been theoretically proven that the optimal discriminator predicts the ratio $p_{data}(x)/(p_{data}(x) + p_{model}(x))$. If the two distributions are the same for optimal G, the ratio is 0.5. **[4 marks]**

- Type of question: Creative
- Learning outcomes: 1 and 2.
- Marking scheme: 1 mark for answering 0.5. Full marks for correct explanation (either of 2).

(c) The AE is the most optimal. The classifier benefits by pre-trained parameters that tend to cluster the data. Clustering is a result of the reconstruction loss, which is optimized by AE. The VAE additionally minimizes a Regularizer, which opposes the reconstruction loss, therefore is less ideal. The basic GAN cannot be used, because it does not have an encoder to learn mapping $x \rightarrow z$ where z a potentially useful (e.g. clustered) representation of the data. **[7 marks]**

- Type of question: Creative
- Learning outcomes: 1,3 and 4.
- Marking scheme: 1 mark for correct choice. 2 marks for explaining clustering as result of reconstruction for AE. 2 marks explaining the regularizer of VAE is not useful. 2 marks for explaining GAN has no encoder.