CS F425 Deep Learning Final Test
AY 2021-22 Sem-II

Department of CS & IS, BITS Pilani, K.K. Birla Goa Campus

IC: Tirtharaj Dash [tirtharaj@goa.bits-pilani.ac.in]

May 14, 2022 (Forenoon)

Q. Which of the following can learn representation of data?

(a) Multilayer Perceptron

(b) Transformer and BERT

(c) Random Forest

(d) k-Nearest Neighbour

(e) None of the above

Ans: a,b
Reason: DNNs (a and b) convert data in such a form that it would be better to solve the desired problem. This is called representation learning.

Q. Which of the following choice(s) can help alleviate the exploding gradient problem in deep networks?

(a) Using residual connections in the network

(b) Using batch normalisation

(c) Using `ReLU` activations instead of `sigmoid`

(d) Using LSTMs instead of standard RNN cells

(e) None of these

Ans: e
Reason: possible solutions: gradient clipping, truncated backprop

Q. An auto-encoder can compress the input data in its hidden representation if:

(a) If the input features are not correlated

(b) If the input features are independent

(c) If the input features are unrelated

(d) None of these

Ans: d
Reason: straightforward

Q. Given the following two statements, select the correct options:

S1: Auto-encoders are supervised learning techniques.
S2: Auto-encoder's output is exactly the same as the input.

(a) Only S1 is true

(b) Only S2 is true

(c) Both S1 and S2 are true

(d) Both S1 and S2 are false

Ans: d
Reason: Autoencoders are an unsupervised learning technique. The output of an autoencoder are indeed pretty similar, but not exactly the same.

Q. Given the following two statements, select the correct options:

S1: Sparse autoencoders introduce information bottleneck by reducing the number of nodes at hidden layers.
S2: The idea is to encourage the network to learn an encoding and decoding, which only relies on activating a small number of neurons.

(a) Only S1 is true

(b) Only S2 is true

(c) Both S1 and S2 are true

(d) Both S1 and S2 are false

Ans: b
Reason: Sparse autoencoders introduce an information bottleneck without requiring a reduction in the number of nodes at hidden layers. It encourages the network to learn an encoding and decoding, which only relies on activating a small number of neurons. Interestingly, this is a different approach towards regularisation, as we normally regularise the weights of a network, not the activations!

Q. You have constructed a very large and complex neural network model for representation learning. Learning adequately sparse representations for inputs has a lot of benefits in deep learning, such as extracting discriminating representation for any given input. Let $L(\mathbf{x}, \hat{\mathbf{x}})$ denote the loss function for your present model where $\mathbf{x}$ denotes the input. Which of the following loss functions allow(s) the model to learn sparse representation (select all that apply)?

(a) Modify the loss to $L' = L - \lambda \sum_{i=1}^{m} |h_i|$

(b) Modify the loss to $L' = L - \frac{\lambda}{2} \sum_{i=1}^{m} h_i^2$

(c) Modify the loss to $L' = L + \lambda \sum_{i=1}^{m} |h_i|$

(d) Modify the loss to $L' = L + \frac{\lambda}{2} \sum_{i=1}^{m} h_i^2$

(e) None of these

[Here: m is the size of the representation layer, and h_i is the value of the ith hidden unit, and λ is the regularisation parameter.]

Ans: c,d
Reason: Sparse representations can be learned by enforcing a constraint on the number of hidden (representation) units activated given an input instance. This is done by penalizing the *activations* of the neural network, so that only a small subset of the neurons fire for any given data instance. The simplest way to achieve sparsity is to impose an L_1-penalty on the hidden units. Therefore, the original loss function L is modified to the regularized loss function L' as $L' = L + \lambda \sum_{i=1}^{m} |h_i|$ or $L' = L + \frac{\lambda}{2} \sum_{i=1}^{m} h_i^2$.

Q. A contractive autoencoder is a heavily regularized encoder (normally, overcomplete) in which the hidden representation is not very sensitive to changes in input values. Consider an autoencoder with single hidden layer. Let the input vector be defined as $\mathbf{x} = (x_1, \ldots, x_d)$, and the hidden units be $\mathbf{h} = (h_1, \ldots, h_k)$ (usually, $k \geq d$).
The reconstruction loss for a single input instance is defined as:

$$L = \sum_{i=1}^{d} (x_i - \hat{x}_i)^2,$$

where, $\hat{\mathbf{x}}$ is the reconstructed version of the input.
The contractive autoencoder defines a new loss (let's call it L') by adding a regularising term to L. Which of the following regularisations help(s) the contractive encoder achieve robustness? [choose all that apply]

(a) $L' = \sum_{i=1}^{d} (x_i - \hat{x}_i)^2 + \frac{\lambda}{2} \sum_{j=1}^{k} \left(\frac{\partial h_j}{\partial h_{j-1}} \right)^2$

(b) $L' = L + \frac{\lambda}{2} \sum_{i=1}^{m} h_i^2$

(c) $L' = \sum_{i=1}^{d} (x_i - \hat{x}_i)^2 + \frac{\lambda}{2} \sum_{i=1}^{d} \sum_{j=1}^{k} \frac{\partial h_j}{\partial x_i}$

(d) None of these

Ans: d
Reason: This is standard definition of contractive loss. The derivative of h w.r.t. x conveys the fact that the representation is not sensitive to changes in the denominator (input). The answer (c) won't work as the derivative could be negative as well.

Q. Variational Autoencoder (VAE) is a neural generative model. Here are a few computational steps that are involved behind the working of VAE (showing one pass through the computational graph assuming Gaussian prior and posterior):

S1	Push $\mathbf{x}$ through the encoder	$\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x = M(\mathbf{x}), \Sigma(\mathbf{x})$
S2	Sample $\mathbf{z}$	$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x)$
S3	Push $\mathbf{z}$ through the decoder	$\hat{\mathbf{x}} = p_{\boldsymbol{\theta}}(\mathbf{x}\|\mathbf{z})$
S4	Compute reconstruction loss	$\mathcal{L}_{recon} = MSE(\mathbf{x}, \hat{\mathbf{x}})$
S5	Compute variational loss	$\mathcal{L}_{var} = -\mathrm{KL}[\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x)\|\|\mathcal{N}(0, I)]$
S6	Compute VAE loss	$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{var}$

Which of the above step will cause difficulty during backpropagation, and how is it solved in practice?

(a) S1 (solved by setting it to 1)

(b) S2 (solved by reparameterisation trick)

(c) S3 (solved by weight tying)

(d) S5 (solved by approximation trick)

(e) None of the above

Ans: e

Reason: Backpropagation cannot handle sampling. Replace the step with $\mathbf{z} = \boldsymbol{\mu}_x + \boldsymbol{\sigma}_x \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$

Q. For variational inference in the context of a VAE, we use a proposal distribution (usually denoted by q). Which of the following are some desired properties of a good q?

(a) Easy to sample from

(b) Differentiable w.r.t. the parameters

(c) Given a training sample $\mathbf{x}$, the sampled $\mathbf{z}$ is likely to have a non-zero $p(\mathbf{x}|\mathbf{z})$

(d) It must look like a Gaussian distribution

(e) None of these

Ans: a,b,c
Reason: straightforward: lecture

Q. For a VAE with latent dimension $[1 \times 3]$, consider that the encoder outputs $\boldsymbol{\sigma} = [0.1, 0.2, 0.1]$ and sampled $\boldsymbol{\epsilon} = [0.01, 0.02, 0.03]$ from $\mathcal{N}(\mathbf{0}, I)$. What is the encoder's output mean vector to produce the latent vector $\mathbf{z} = [0.3, 0.1, 0.5]$?

(a) $[0.040, 0.040, 0.080]$

(b) $[0.301, 0.104, 0.503]$

(c) $[0.299, 0.096, 0.097]$

(d) $[0.200, -0.100, 0.400]$

(e) None of these

Ans: e
Reason: $\boldsymbol{\mu} = z - \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}$

Q. Consider a machine translation model with an attention mechanism. Roughly speaking, while generating the word in the target language at time step t, the network gets the attention weights $\alpha(t, t')$ from the source language words from time $t' = 0$ to $t' = n$ (where n is the length of the sentence in the source language). Which of the following is/are true?

(a) $\sum_t \alpha(t, t') = 1$

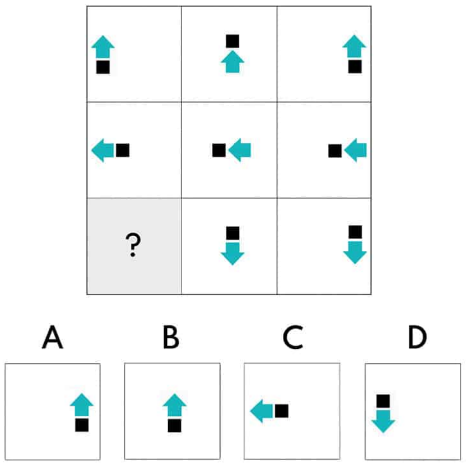
(b) $\sum_{t'} \alpha(t, t') = 1$

(c) Attention weights are same across all the time steps

(d) Attention weights are different across all the time steps

Ans: b,d
Reason: Recall classroom discussion.

Q. We are all familiar with questions related to "reasoning". We want to solve some reasoning tasks using a purely deep learning approach. Consider a reasoning task: Given a 3×3 grid of mini-images consisting of geometric shapes and of different colours, with one of the 9 grids blank, the goal is to fill the blank grid with the correct image from a given choice of 4 images. See one such instance below:



A dataset for this problem will consist of several such instances. Which of the following do you think is or are relevant to approaching this problem?

(a) Representation of the inputs and outputs

(b) Choice of a deep neural network class

(c) The type of learning (end-to-end, greedy pre-training, compositional learning, etc.)

(d) Looking for domain-knowledge

(e) Possibility of generating more data of the same kind

(f) None of the above

Ans: a–e
Reason: All these are necessary to think of an approach to solve the reasoning problem.

Q. Which of the following are possible applications of Restricted Boltzmann Machines (RBMs)?

(a) Dimensionality reduction

(b) Data classification

(c) Collaborative filtering

(d) Feature learning

(e) Topic modelling

(f) None of these

Ans: a–e
Reason: RBM can do all these. (Source: Wikipedia)

Q. Given a problem of simplifying mathematical expressions (see below for an example), which of the following class of deep network could be used to solve the problem?

Input Expression	Simplified Expression
$7x^3 + 3x^2 - 5x^2 - 5$	$7x^3 - 2x^2 - 5$
$4x - 5 - 4x + 5$	0
$\dots$	$\dots$

(a) Multilayer Perceptron

(b) Recurrent Neural Network

(c) Transformer

(d) Convolutional Neural Networks

(e) Graph Neural Networks

(f) None of these

Ans: b–e
Reason: MLPs are not fit for sequences. GNNs can work on parse trees.

Q. In what category(ies) of problems do we need to use a deep-learning solution?

(a) Problems with large-amount of data

(b) Problems where a prediction needs a human-readable explanation

(c) Problems that require answering "how" the model reached a decision

(d) Problems that are simple to be solved (e.g. well-separable data)

(e) None of these

Ans: a
Reason: b,c with DNN is still not easy; d is not advisable.