CS 7015 - Deep Learning - Endsem **Duration: 3 Hours**

Instructions: Write your answers in the space provided. If you use more space than what is provided then your answer **will not be evaluated**. You will be provided additional sheets for rough work. First write the answers in the rough sheet and once you are sure of the concise precise phrasing of the answer write in the provided space. Write your answers in good legible handwriting (don't scribble). If needed, we will provide you 30 minutes extra to copy your answers from the rough sheet in good handwriting. The font size of your answers should not be smaller than the font size used for the questions (we have used the same font size for the answers and were able to fit them in the provided space).

1. **(0.5 Mark)** What is the difference between McCulloch Pitt's neuron and Perceptron ?

   Solution:

2. **(0.5 Mark)** Write down the update rules for Adam ?

   Solution:

3. **(0.5 Mark)** Write down the best rank-1 approximation of a matrix where the error in approximation is measured using Frobenius norm ?

   Solution:

4. **(0.5 Mark)** What is the technical name for the problem which occurs when all the weights of a neural network are initialized to 0 ?

   Solution:

5. **(0.5 Mark)** Which form of regularization ensures sparse weights ?

   Solution:

6. **(0.5 Mark)** Training an ensemble of large neural networks is obviously very hard. Suggest a method for doing this efficiently.

   Solution:

7. **(0.5 Mark)** In LSTMs what controls the flow of information and gradients ?

   Solution:

8. **(0.5 Mark)** Given the input dimensions W and H, filter size F, padding P, stride S used in a colvolution layer, give a formula for computing the dimensions of the output W' and H' of the convolution layer.

Solution:

9. **(0.5 Mark)** What type of models are RBMs (generative or discriminative) ?

   Solution:

10. **(0.5 Mark)** Who proved Pythagoras theorem :-) ?

    Solution:

11. The encoder for the VAE learns the mean and variance for the input data distribution and the decoder samples points from this distribution. However, the sampling procedure is not differentiable and hence we cannot backpropagate through the sampling layer. To solve this issue, we use the reparameterization trick, i.e. push the sampling procedure to the input.

    (a) **(1 Mark)** VAE models the sampling distribution $Q(z|X)$ using a normal distribution. Write down the equation for moving out the sampling layer and explain why we can backpropagate after reparameterization.

    Solution:

    (b) **(2 Marks)** Write down one example modelling choice for $Q(z|X)$ where the reparameterization trick would not work?

    Solution:

12. **(3 Marks)** Derive the expression for the KL divergence between two univariate gaussian distributions $p = \mathcal{N}(\mu_1, \sigma_1)$ and $q = \mathcal{N}(\mu_2, \sigma_2)$ i.e $KL(p||q)$. Note that,

$$KL(p||q) = \int [\log p(x) - \log q(x)]p(x)\,dx$$

    Solution:

13. **(1 Mark)** Prove that the negative log probability $P(X|z)$ (given the assumptions made in VAE's) is proportional to euclidean distance between $f(z)$ and X, (Notations as discussed in class, $f(z)$ computed using a neural network and $P(X|z)$ assumed to be a normal distribution)

Solution:

14. The seq2seq models that we discussed in class involved sequences with a natural order both at the source and target sequence. One could think of settings where we might not necessarily have an order in the input sequence or we might want to alter the input sequence to make the learning problem easier. For example, it was observed that doing so improves English to French translation by 5-10% .

    (a) **(1 Mark)** We noted that there is some particular order of the input for which learning becomes easier (i.e. model performs better). Say that you do not know what is the "right" order of input for the model. Given the encoding of the input, i.e. output(s) of the encoder, suggest a technique to help the model choose the right order for decoding. $f(z)$ computed using a neural network and $P(X|z)$ assumed to be a normal distribution)

    Solution:

    (b) **(2 Marks)** Recurrent networks encode a variable length sequence to a fixed length vector which can then be used as a representation for the input sequence. It is reasonable to expect that as the length of the input increases, the fixed length vector representation will not be able to capture the entire sequence well. One approach is to use attention mechanism to allow the model to peep into the encoder states. Suggest an extension to the attention model which could potentially help improve the performance of the model. (Hint: Ideas discussed in class)

    Solution:

15. **(4 Marks)** For a neural network trained using dropout at test time, we run the model with all the weights, with the modification that the weight outgoing from unit i is multiplied by probability of including the unit i. This rule is known as the weight scaling rule and works well in practice but there is no theoretical justification for the

same. However, we can prove that this rule is exact for some class of models. Consider the softmax regression classifier with n input variables represented by vector v.

$$P(\mathrm{y} = y|\mathbf{v}) = \text{softmax}\big(W^\top v + b\big)_y$$

Each of the models in the ensemble can be represented by an element wise multiplication by a binary vector d.

$$P(\mathrm{y} = y|\mathbf{v}; \mathbf{d}) = \text{softmax}\big(W^\top (d \odot v) + b\big)_y$$

The prediction of the ensemble of models obtained by taking all possible vectors d is given by the geometric mean over all ensemble member's prediction

$$P_{\text{ensemble}}(\mathrm{y} = y|\mathbf{v}) = \frac{\tilde{P}_{\text{ensemble}}(\mathrm{y} = y|\mathbf{v})}{\sum_{y'} \tilde{P}_{\text{ensemble}}(\mathrm{y} = y'|\mathbf{v})}$$

where

$$\tilde{P}_{\text{ensemble}}(\mathrm{y} = y|\mathbf{v}) = \sqrt[\frac{1}{2^n}]{\prod_{d \in \{0,1\}^n} P(\mathrm{y} = y|\mathbf{v}; \mathbf{d})}$$

Given this above expression, prove that the weight scaling rule is exact.

Solution:

16. **(3 Marks)** We saw that for a given decoder timestep t, the attention weights over the S encoder states (timesteps) are constrained to sum up to 1 (i.e., $\sum_{i=1}^{S} \alpha_{it} = 1$). However, there is no constraint on how the attention weights behave across timesteps. This could result in a situation where the attention weights are always concentrated on the same set of encoder states across different decoder timesteps. This may be undesirable (for example, in the case of translation this might mean that some words in the source sentence never get attention). Suggest a way to tweak the loss function to explictly prevent this problem?

Solution:

17. **(4 Marks)** The attention function that we saw in class was of the following form: $e_{jt} = ATT(h_{t-1}, s_j)$ where h_{t-1} is the state of the decoder at timestep t and s_j is the state of the j-th input timestep. This choice of the attention function requires the decoder state h_t to fulfill several purposes at the same time: i) encode a distribution to predict the next token ($\because y_t$ is a function of h_t), ii) serve as a key to compute the attention vector ($\because e_{jt}$ is a function of h_{t-1}), iii) encode relevant content to inform future predictions. It is observed that such overloaded use of the decoder state makes training the model difficult. Propose a modification to the attention mechanism which separates these different functions of the decoder explicitly (we are looking for equations, not a verbose description).

Solution:

18. **(2 Marks)** Let y be the true output and $\hat{f}(x)$ be the output prediced by the models. Prove that $(E[(y - \hat{f}(x))^2] = Bias^2 + Variance + \sigma^2$ (irreducible error))

Solution:

19. **(2 Marks)** Earlier we saw that a fully connected layer of a convolutional neural network can be implemented as a convolutional layer. What is the advantage of doing this (as opposed to just implementing it as a feedforward layer) ?

Solution:

20. **(3 Marks)** Prove that word2vec does implict matrix factorization.

Solution:

21. **(2 Marks)** We saw various encoder-decoder models in class and analyzed them using the *data-model-parameter-loss-algorithm* framework. Now consider the hierarchical encoder decoder with attention models used for document classification. Write down the equations of this model (*i.e.*, the equations which connect the input x to the output y).

Solution: