Name:

Roll No.: **Dept.**:

IIT Kanpur
CS771A (IML)
Quiz-4
Date: November 14, 2023

Instructions: *Total:* **30 marks**

1. Duration is 60 minutes. Please write your name, roll number, department on **all pages**.
2. Write your answers clearly in the provided box. Keep your answer precise and concise.

Section 1 (Short/medium-length answer questions: 30 marks). .

1. Assuming each input $\boldsymbol{x}_n \in \mathbb{R}^D$, clearly write down all the operations (with expressions) that a multi-layer Perceptron (MLP) performs to compute the first hidden layer's output $\boldsymbol{h}_n^{(1)} \in \mathbb{R}^{K_1}$? **(2 marks)**

   $\boldsymbol{h}_n^{(1)} = g(\mathbf{W}^{(1)\top}\boldsymbol{x}_n)$ where $\mathbf{W}^{(1)}$ is a $D \times K$ matrix and g denotes some nonlinear activation function.

2. Between sigmoid activation and ReLU activation, which one is cheaper to compute and why? **(2 marks)**

   ReLU is cheaper since it only requires a max whereas sigmoid requires computing exp function.

3. Consider a single hidden layer MLP for multi-class classification where each input $\boldsymbol{x}_n \in \mathbb{R}^{100}$, the number of classes is 5, and the hidden layer's dimension is 10. How many weights (ignore the bias parameters) would be required for this MLP? Briefly explain the reason too. **(2 marks)**

   Note that the input layer is 100-dimensional, the hidden layer (only single hidden layer) is 10-dimensional, and the output layer is 5-dimensional. To connect input layer to hidden layer, we need $100 \times 10 = 1000$ weights, and to connect hidden layer to output layer, we need another $10 \times 5 = 50$ weights. Thus we need a total of 1050 weights.

4. For an input $\boldsymbol{x}_n = [1, 2, 3, 4]$, write down the feature map produced by applying a filter $\boldsymbol{w} = [1, 1]$ assuming no zero padding around the input, and a stride size of 1. **(2 marks)**

   Due to the size of the filter and stride size of 1, it can only be placed at 3 locations. It is easy to see that the resulting feature map is [3, 5, 7].

5. Write down the expression of how the hidden state $\boldsymbol{h}_t$ at step t of an RNN is computed (clearly showing the other quantities/parameters this computation depends on). **(2 marks)**

   $\boldsymbol{h}_t = g(\mathbf{W}\boldsymbol{x}_t + \mathbf{U}\boldsymbol{h}_{t-1})$ where $\mathbf{W}$ and $\mathbf{U}$ are matrices of appropriate sizes and g denotes some nonlinear activation function.

6. Given a training set $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, for a test input $\boldsymbol{x}_*$, the prediction by a standard K-nearest neighbors regression method can also be seen as using an attention mechanism. Briefly explain what the query, keys, and values would be in this case, and what will be the attention weights? **(2 marks)**

   For K-nearest neighbors regression y_* is a weighted average of the training labels of the K nearest neighbors. In the attention anology, we can think of the test input $\boldsymbol{x}_*$ as the query, the training inputs $\{\boldsymbol{x}_n\}_{n=1}^N$ as the N keys, and their respective labels $\{y_n\}_{n=1}^N$ as the values. The attention scores can be defined in many ways. For example, we can set them to 1 for each of the K-nearest neighbors and 0 for other inputs.

7. Why are positional encodings not needed in an RNN but needed in a transformer? **(2 marks)**

   RNN processes tokens in the input sequence in a sequential manner, so the positional encoding is not needed. Transformers process all the tokens in parallel and therefore positional encoding is needed to specify where in the input sequence a token appears.

8. In CNNs, after the convolution step, what is the next step and why is that step needed? **(2 marks)**

   Usually, the next step after the convolution step is pooling and it is needed to reduce the size of the feature maps produced by the convolution operations.

Name:

Roll No.: **Dept.**:

IIT Kanpur
CS771A (IML)
Quiz-4
Date: November 14, 2023

9. Given the same training data with classes that are not linearly seperable, briefly explain which of these two would have a larger bias (approximation error): linear SVM or a nonlinear SVM? **(2 marks)**

   Linear SVM will underfit in this case since the classes are not separable linearly and will thus linear SVM have a larger bias (approximation error) as compared to nonlinear SVM. Even adding more training data will not help linear SVM here.

10. Briefly explain the double descent phenomenon. **(2 marks)**

    As we increase the model complexity, the training error keeps on decreasing. The test error decreases at first but, beyond a point, it starts increasing due to overfitting. However, what is observed (especially with deep learning models) is that if we keep increasing the model complexity even further then the test error starts decreasing once again. Thus there are two descent phases for the test error (and hence the name "double descent").

11. What is the difference between a standard RNN and a bidirectional RNN? How is the hidden state $\boldsymbol{h}_t$ at step t represented in bidirectional RNN? **(2 marks)**

    In a standard RNN, the hidden states evolve only in the forward direction whereas in the bidirectional RNN, we also have another layer of hidden states that evolve in the reverse direction. Denoting the respective hidden state vectors as $\overrightarrow{\boldsymbol{h}}_t$ and $\overleftarrow{\boldsymbol{h}}_t$, we concatenate them to get the effective hidden state $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t]$.

12. Why do we need to perform normalization of the hidden layers of a deep neural network? **(2 marks)**

    Normalization of the hidden layer nodes is needed so that they are on the same scale (standardized) which makes training of deep neural networks easier.

13. For transformers, briefly explain which of the following two normalization schemes would you prefer: batch normalization or layer normalization? **(2 marks)**

    For transformers, batch normalization is difficult to apply since the each input sequence can possibly be of a different length. So it is better to perform normalization for one input sequence at a time, and thus layer normalization is preferred which operates for each input individually.

14. Given the pre-activation vector $\boldsymbol{z}_n = \mathbf{W}^\top \boldsymbol{x}_n \in \mathbb{R}^K$ and its corresponding activation vector $\boldsymbol{h}_n = g(\boldsymbol{z}_n)$ where g is the sigmoid function applied element-wise on $\boldsymbol{z}_n$, what will be the size of the derivative $\frac{\partial \boldsymbol{h}_n}{\partial \boldsymbol{z}_n}$. Also write down its expression. **(2 marks)**

    Size both $\boldsymbol{h}_n$ and $\boldsymbol{z}_n$ are of size K, the derivative will be a Jacobian matrix of size $K \times K$. Furthermore, since g is applied on the vector $\boldsymbol{z}_n$ element-wise, all off-diagonal terms $\frac{\partial h_{nk}}{\partial z_{nk'}}$ will be zero. Only the diagonal terms will be nonzero and the k-th diagonal entry of the Jacobian matrix will be equal to $g'(z_{nk})$. Since $g(z_{nk}) = \sigma(z_{nk}) = \frac{1}{1+\exp(-z_{nk})}$, $g'(z_{nk}) = \sigma(z_{nk})(1 - \sigma(z_{nk}))$.

15. Even though a neural network with a very wide single hidden layer can approximate any function, why might we instead prefer deeper neural networks with less wider hidden layers? **(2 marks)**

    The reason we prefer deeper networks, even if each layer is not so wide is because, with multiple layers, we can learn features at multiple levels of abstractions (due to property of compositionality - higher layer learn features that are compositions of lower layer features). Features at the higher layers of the network are much more meaningful/interpretable, and make prediction task much more easy.