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. **(3 marks)** In class we discussed a technique (Word2Vec) to obtain word embeddings given a text corpus. We noted that these representations have interesting properties like syntactically similar words are "closer" to each other compared to dissimilar words. Say we wish to extend this idea to obtain representations for words across 2 languages (English and French). Further we wish to have similar words across language to have representations that are "close".

   We are given a corpus of English and French text. We are also given a dictionary $\mathcal{D}$ of words that maps words in English to equivalent words in French. Write down the objective function for training a Word2Vec model for the above task.

   Solution:

2. A language model assigns a probability score for a sequence to words in a language. A good language model assigns higher probability for "valid" sentences in a language and low probabilities for sentences which are incorrect (for instance - sentences which do not follow language grammar rules).

   An n-gram model makes an order n Markov assumption for sentences. In particular we assume the probability of a word w given the past $n-1$ words is independent of the rest of words that appeared in the past.
   The probability of a sentence $s = w_1 w_2 w_3 \dots w_T$ is given by

   $$\Pr(s) = \prod_{i=1}^{T} \Pr(w_i | w_{i-n+1}, \dots w_{i-1})$$

   You can assume that the sentence is padded with $n-1$ start symbols to make the above equation to work. The above equation follows from the Markov assumption and chain rule for expressing a joint distribution. Notice that the language modelling task reduces to estimating the conditional probabilities from the above equation.

(a) **(1 mark)** We can estimate the conditional probability by counting the number of times the given sequence appears in our training corpus.

$$P(w_i|w_{i-n+1},\ldots,w_{i-1}) = \frac{c(w_{i-n+1}w_{i-n}\ldots w_i)}{c(w_{i-n+1}\ldots w_{i-1})}$$

where $c(w_{i-n+1}\ldots w_{i-1})$ is the count of number of times the sequence of words $w_{i-n+1}\ldots w_{i-1}$ appear in the training corpus. What is a potential problem with using the above method for estimating the conditional probabilities ?

Solution:

(b) **(1 mark)** The continuous bag of words model that we discussed in class is an alternative approach for estimating the conditional probability using a neural network . The words in the sequence are represented using real vectors and the concatenation of these vectors form the input to the network. A final softmax layer gives the probability of the next word in the sequence given the current $n-1$ words. How does this approach solve the problems with the count based models?

Solution:

3. **(2 marks)** Convolutional neural networks consist of a series of convolutional layers and max pooling layers. The max pooling layers do not contain any weights. Say the output of one such convolution layer is a 4×4 feature map denoted by h (comprising of $h_{11}, h_{12}, ..., h_{44}$). On top of this we have a 2×2 max pooling layer whose output is denoted w.r.t. m (comprising of $m_{11}, m_{12}, m_{21}, m_{22}$). Say we have computed the derivative of the loss w.r.t. the output m of the max-pool layers. Express the derivative of the loss w.r.t. the output (h) of the convolution layer.

Solution:

4. **(2 marks)** Consider a feedforward neural network fully connecting a m dimensional input layer to a n dimensional output layer. Represent this feedforward network as a convolutional neural network. Specifically write the number of filter(s) to be used and the sizes of these filters.

Solution:

5. **(2 marks)** Consider an input volume of size $W \times H \times D$. Now consider two convolutional neural networks. The first one uses one $7 \times 7 \times D$ filter on the input image followed by a max pooling layer. The second one uses three $3 \times 3 \times D$ filters on the input image followed by a max pooling layer. Do you see any advantage of using one of these networks over the other. Explain your answer.

Solution:

6. **(2 marks)** A typical softmax layer on $|V|$ output classes involves $|V| - 1$ computations. As $|V|$ increases it might be computationally expensive to perform these computations. Morin and Bengio proposed an approximation to softmax inspired by binary trees known as Hierachical softmax. Here they build a binary tree with $|V|$ leaves (depth $\log_2 |V|$). The leaves of the tree denote the probabilities of the specific class and the probability of a given class is given by the product of probabilities of the path from the root to the specific leaf. At each node we compute the probability of travesing on the left sub-tree and the right sub-tree. It can be shown that the effective computation needed goes down from $|V| - 1$ to $\log_2 |V|$. It seems like we can get further improvement by increasing the branching factor on the tree and reduce the number of computations further. Is a binary tree the best possible branching solution we can use ? Justify your claim.

Solution:

7. **(3 marks)** Inspired by the idea used in contractive autoencoders (which focuses on the Jacabian $\frac{\partial h}{\partial x}$) suggest a method for solving the vanishing/exploding gradient problem in Recurrent Neural Networks.

Solution:

8. **(3 marks)** We saw that Residual Networks use identity connections to facilitate better flow of information (and gradients). On other hand, LSTMs use gates to facilitate better flow of information. One potential problem with Residual Networks is that we hardcode the manner in which the information should flow (for example, by always adding connections from layer $k-2$ to layer $k \quad \forall k$). Suggest a way to make this more adaptive (instead of hardcoding the connections).

Solution: