2) $\overrightarrow{h_t} = f(W_{fx} x_t + W_{fh} \overrightarrow{h_{t-1}} + b_f)$ $\quad \overleftarrow{h_t} = f(W_{bx} x_t + W_{bh} \overleftarrow{h_{t+1}} + b_b)$

$y_t = g(W_y [\overrightarrow{h_t}; \overleftarrow{h_t}] + b_y)$

; means concat

Tutorial 8

Question 1

RNNs have been used for many problems in NLP. For this exercise, we will consider RNNs for named entity recognition as shown in the Figure below, assuming there is only one type of named entity (person).

1) If W_{xh} and W_{hh} are the weight matrices for input to hidden layer and hidden to hidden layer, write down the composition function of a vanilla unidirectional RNN model.

2) Write the composition functions for a bidirectional RNN.

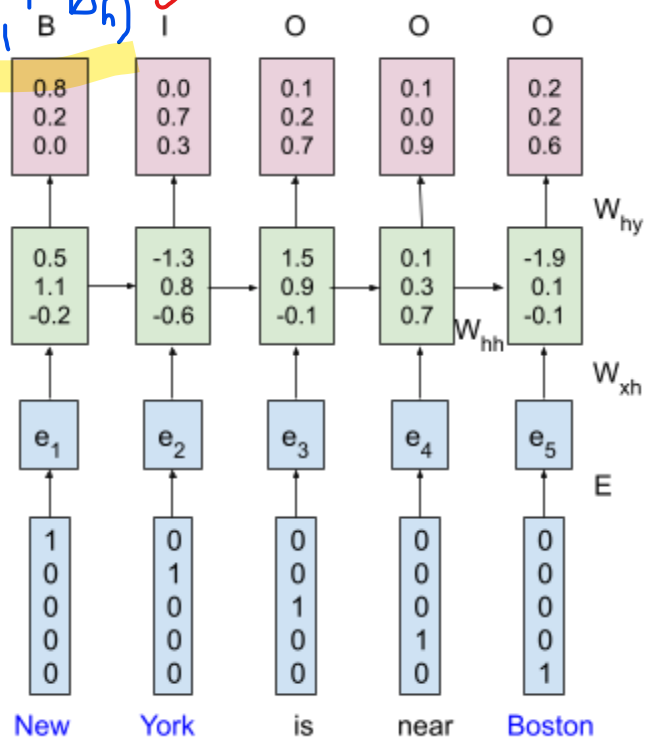
3) Describe the number of classes in this NER task and how the final label is predicted in the output layer given the hidden representation for each word.

4) F1-score is a commonly used metric to evaluate the NER performance. It is computed based on recall and precision: $F1 = \frac{2 \times recall \times precision}{recall + precision}$. Suppose your model predicts "B I O O O" whereas the gold label should be "B I O O B". What is the F1 score in this case?

1) $h_t = f(W_{xh} e_t + W_{hh} h_{t-1} + b_h)$

$y_t = W_{hy} h_t + b_y$

$y_t = softmax(W_{hy} h_t + b_y)$



3) There are 3 classes of B,I,O. When given the hidden representations of each word, the formulas are followed which returns the probability of each of the classes as y. The higher the probability, the more likely it is that class

4) Get TP, FP, TN for each class

	B	I	O
True Positive =	1	1	2
False Positive =	0	0	1
False Negative =	1	0	0

Precision = (2/3 + 1 + 1) / 3 = 0.88889

Recall = (1/2 + 1 + 1) / 3 = 0.83333

$P = \frac{TP}{TP + FP}$

$R = \frac{TP}{TP + FN}$

1.4) ONLY consider the named entities here

Predicted: B I O O O
Gold: B I O O B

Extract named entities from labels:
Predicted: New York (B I)
Gold: New York(B I). Boston(B)

Calculate Precision and Recall:
Precision = TP/(TP+FP) = 1 / (1+0) = 1
Recall = TP/(TP+FN) = 1 / (1+1) = 0.5

F1 = 2 * 1 * 0.5 / (1 + 0.5) = 0.67

$F1 = \frac{2PR}{P+R}$

Q2
1) Padding can be used so that all sequences are of the same length, else it goes into the next batch
A binary mask is used to indicate which positions are real data vs padding

Masks used to ignore the padded tokens, ensuring that they do contribute to the model's output or loss calculation when comp the loss/hidden states

2) The model might have a high training performance but a low testing performance.
This can be resolved by stacking with other models.
Eg. Encoder(RNN/Transformer) - Decoder(RNN) Architecture, Retrieval Augmented Generation: Input => Retrieve similar sequences => RNN + Retrieved Context =

Question 2

To actually implement RNNs in python, there are several points to take note of.

1) For mini-batch training, the input to RNNs contains a number of sentences with different lengths. In order to perform efficient batch training, how to handle variable-length inputs?

2) In language model training with RNNs, the model uses the correct previous tokens from the dataset to predict the next one. However, during testing, it must rely on its own predictions to generate future tokens. How might this difference between training and testing impact the model's performance? Can you think of a solution to alleviate this issue?

2. 2) This is a problem of exposure bias. Teacher forcing used in training, helping the mod learn to predict the next token based on current context. During testing, model uses its ov predictions as input. If mistake made early on, incorrect prediction becomes part of input context, leading to potential error propagation. As model does not see its own mistakes d training, it cannot adapt when mistakes occur during testing, generating less coherent and accurate sequences, especially for long outputs

Question 3

(see the word doc for this c

Imagine a simplified neural machine translation model that translates sentences from English to French. Consider that the model uses a single-layer RNN for both the encoder and the decoder. For simplicity, assume that each word in both languages has already been represented as a fixed-size vector (e.g., word2vec embedding).

Q3.1) Encoder produces an encoding of the source sentence and provides initial hidden state for Decoder, capturing all information about the source sentence

1) Describe the role of the encoder RNN in the neural machine translation process.

2) What is the purpose of the decoder RNN, and how does it differ in functionality from the encoder?

A decoder is a conditional language model that generates target sentence , conditioned on encoding

3) Imagine you have an English sentence: "I am learning". You wish to translate it into French: "Je suis en train d'apprendre". Describe the steps the NMT model would take to perform this translation.

4) How would the process change if the encoder is a bi-directional RNN?

5) What could be the problem of greedy decoding during translation generation? Can you come up with any possible solution?

A limited number of sentences will be generated. Noise can b introduced to the model through temperature sampling

6) What are some potential drawbacks or limitations of using RNNs for NMT?

Vanishing/Exploding gradients

Coding exercises: RNN
https://drive.google.com/file/d/1jZUpCxs8s7pDDHRdHrmyl1XkCafxmFXP/view?usp=sharing

2.2 Solution: Scheduled Sampling
Gradually replace ground truth tokens with the model's own predictions during training. At the beginning of training, model sees the correct tokens. As training progresses, it slowly relies on its own predicted tokens. This allows model to adapt to input conditions it will encounter during inference, reducing exposure bias