CMPT 413/825 Natural language processing
Homework 3, Fall 2020

Due: October 29th, 2020

Instructions Answers are to be submitted by 11:59pm of the due date as a PDF file. Please submit your answers using either Gradescope (preferred) or Coursys. You do not need to submit your answers using both platforms, just one.

- **Gradescope**: Go to https://www.gradescope.ca/, select **HW3** and submit your PDF. Please make sure to assign the pages to corresponding questions.
- **Coursys**: Go to Coursys, select the **Homework 3** activity and submit your answer as **answer.pdf**.

This assignment is to be done individually.

1. Hidden Markov Model (12 points)

Hidden Markov Models can be used to perform named entity recognition by identifying entities in a piece of text. In this question, there are two types of entity types: *person* `(P)` and *city* `(C)`.
Assume, we have the following annotated training data:

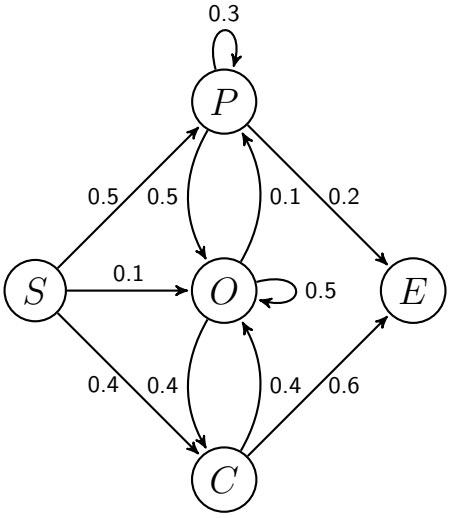
- `<SOS>` Florence`(C)` is smaller than Washington`(C)` `<EOS>`
- `<SOS>` Madison Avenue in New`(C)` York`(C)` is a nice place to live `<EOS>`
- `<SOS>` Florence`(P)` Madison `(P)` went to Washington`(C)` `<EOS>`
- `<SOS>` Robert`(P)` Washington`(P)` stayed at the Madison `<EOS>`
- `<SOS>` Nick`(P)` Madison`(P)` met Denzel`(P)` Washington`(P)` `<EOS>`

All words which are not annotated belong to the class O (=other words).

(a) (4 points) Draw the HMM with nodes for starting state S and final state E (for beginning and end of a sentence) and the "hidden" states P, C and O. Label the edges with the probabilities from the training data given above.

(b) (2 points) Specify the emission probabilities for C and P.

(c) (6 points) Given an HMM similar to the one that you have constructed in a), calculate the most likely annotation for the sentence "`<SOS>` Florence travels to Alberta `<EOS>` " using the Viterbi algorithm.



Assume that the emission probabilities are as follows:

- Start and Final states:
$$P(\texttt{<SOS>}|S) = P(\texttt{<EOS>}|E) = 1$$
- People:
$$P(\text{“Lydia”}|P) = 0.25, \quad P(\text{“Sue”}|P) = 0.5, \quad P(\text{“Florence”}|P) = 0.15 \quad P(\text{“Alberta”}|P) = 0.1$$
- Cities:
$$P(\text{“Washington”}|C) = 0.4, \quad P(\text{“Paris”}|C) = 0.3, \quad P(\text{“Florence”}|C) = 0.2, \quad P(\text{“Alberta”}|C) = 0.1$$
- Other words:
$$P(\text{“went”}|O) = 0.2, \quad P(\text{“to”}|O) = 0.2, \quad P(\text{“travels”}|O) = 0.3, \quad P(\text{“in”}|O) = 0.3$$

2. LSTM Gradient (10 points)

In this question, you will derive the Backprop Through Time (BPTT) equations for the univariant version of the Long Short Term Memory (LSTM) architecture.

First let's review the BPTT equations for the vanilla RNN. For convenience, we will use the notation $\overline{v} \triangleq \frac{\partial L}{\partial v}$, where L is the function and v the variable that we are taking the gradient with respect to[1]. Often L will be the function that we are optimizing and v a weight or parameter we are optimizing over.

Let's consider the vanilla RNN, which performs the following computations.

$$\begin{aligned} z^{(t)} &= ux^{(t)} + wh^{(t-1)} \\ h^{(t)} &= \tanh\left(z^{(t)}\right) \\ L &= f(h^{(t)}) \end{aligned}$$

Note that $\tanh'(z) = 1 - \tanh^2(z)$. Using this notation, we get:

$$\begin{array}{lll} \overline{h^{(t)}} = \frac{\partial L}{\partial h^{(t)}} & = f'(h^{(t)}) + \frac{\partial L}{\partial z^{(t+1)}} \frac{\partial z^{(t+1)}}{\partial h^{(t)}} & = f'(h^{(t)}) + \overline{z^{(t+1)}} w \\ \overline{z^{(t)}} = \frac{\partial L}{\partial z^{(t)}} & = \frac{\partial L}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial z^{(t)}} = \overline{h^{(t)}} \tanh'(z^{(t)}) & = \overline{h^{(t)}}(1 - \tanh^2(z^{(t)})) \end{array}$$

For the purposes of this problem, we will ignore the $f'(h^{(t)})$ term[2]. Using the same procedure for the weights, u and w, we get the following set of BPTT equations for the vanilla RNN (again ignoring the $f'(h^{(t)})$ term):

$$\begin{aligned} \overline{h^{(t)}} &= \overline{z^{(t+1)}} w \\ \overline{z^{(t)}} &= \overline{h^{(t)}}(1 - \tanh^2(z^{(t)})) \\ \overline{u} &= \sum_{t=1}^{T} \overline{z^{(t)}} x^{(t)} \\ \overline{w} &= \sum_{t=1}^{T-1} \overline{z^{(t+1)}} h^{(t)} \end{aligned}$$

(a) (6 points) Now, let's derive the Backprop Through Time (BPTT) equations for the activations and the gates for the LSTM. For reference, here are the computations it performs:

[1] See details about why this notation was introduced and is convenient in Roger Grosse's lecture notes on Backpropagation `http://www.cs.toronto.edu/~rgrosse/courses/csc421_2019/readings/L04%20Backpropagation.pdf`

[2] See `http://www.cs.toronto.edu/~rgrosse/courses/csc421_2019/readings/L13%20Recurrent%20Neural%20Nets.pdf` for the full example with the complete loss

$$
\begin{aligned}
i^{(t)} &= \sigma\left(w_{ix}x^{(t)} + w_{ih}h^{(t-1)}\right) \\
f^{(t)} &= \sigma\left(w_{fx}x^{(t)} + w_{fh}h^{(t-1)}\right) \\
o^{(t)} &= \sigma\left(w_{ox}x^{(t)} + w_{oh}h^{(t-1)}\right) \\
g^{(t)} &= \tanh\left(w_{gx}x^{(t)} + w_{gh}h^{(t-1)}\right) \\
c^{(t)} &= f^{(t)}c^{(t-1)} + i^{(t)}g^{(t)} \\
h^{(t)} &= o^{(t)}\tanh\left(c^{(t)}\right)
\end{aligned}
$$

Please provide the Backprop Through Time (BPTT) equations for the activations and the gates for the LSTM. You should write your answers in terms of the other gradients (e.g. $\overline{h^{(t)}}$, $\overline{c^{(t)}}$, etc). You do not need to show the partial derivatives (e.g. $\frac{\partial L}{\partial z^{(t)}}$) in your answer.

$$
\begin{aligned}
\overline{h^{(t)}} &= \\
\overline{c^{(t)}} &= \\
\overline{g^{(t)}} &= \\
\overline{o^{(t)}} &= \\
\overline{f^{(t)}} &= \\
\overline{i^{(t)}} &=
\end{aligned}
$$

(b) (2 points) Derive the BPTT equation for the weight w_{ix}:

$$\overline{w_{ix}} =$$

(c) (2 points) Explain how the design of the LSTM cells help alleviate the issue of exploding and vanishing gradients.

3. Neural sequence models (14 points)

(a) (2 points) Suppose you want to build a French to English translator system using an LSTM based sequence to sequence model. What would be fed to the decoder as input at each time step t? Note there should be two components to the input.

(b) (2 points) Consider the two components of the input to your decoder from part (a). How does each component affect the generalization of the model during inference?

(c) (4 points) Describe two different decoding strategies for generating English translations with your decoder from part (a). For each strategy, explain when you would want to use it and what is a possible drawback of that decoding strategy.

(d) (2 points) You are building a text classifier using a simple single-layer, unidirectional RNN. Your friend recommend that you used a GRU cell instead of a LSTM cell. Under what circumstance might the GRU work better than the LSTM?

(e) (4 points) You notice that the performance of your classifier from part (d) is not very good. Describe two extensions that you can make to your GRU model from part (d) to improve the performance of your classifier. For each extension, explain why it might help.

4. BLEU score (14 points)

BLEU score is the most commonly used automatic evaluation metric for NMT systems. It is usually calculated

across the entire test set, but here we will consider BLEU defined for a single example.[3] Suppose we have a source sentence $\mathbf{s}$, a set of k reference translations $\{\mathbf{r}_1, \ldots, \mathbf{r}_k\}$, and a candidate translation $\mathbf{c}$. To compute the BLEU score of $\mathbf{c}$, we first compute the *modified n-gram precision* p_n of $\mathbf{c}$, for each of $n = 1, 2, 3, 4$, where n is the n in n-gram:

$$p_n = \frac{\sum_{\text{ngram} \in \mathbf{c}} \min\left(\max_{i=1,\ldots,k} \text{Count}_{\mathbf{r}_i}(\text{ngram}),\ \text{Count}_{\mathbf{c}}(\text{ngram}) \right)}{\sum_{\text{ngram} \in \mathbf{c}} \text{Count}_{\mathbf{c}}(\text{ngram})} \tag{1}$$

Here, for each of the n-grams that appear in the candidate translation $\mathbf{c}$, we count the maximum number of times it appears in any one reference translation, capped by the number of times it appears in $\mathbf{c}$ (this is the numerator). We divide this by the number of n-grams in $\mathbf{c}$ (denominator).

Next, we compute the *brevity penalty* BP. Let $\text{len}(c)$ be the length of $\mathbf{c}$ and let $\text{len}(r)$ be the length of the reference translation that is closest to $\text{len}(c)$ (in the case of two equally-close reference translation lengths, choose $\text{len}(r)$ as the shorter one).

$$BP = \begin{cases} 1 & \text{if } \text{len}(c) \geq \text{len}(r) \\ \exp\left(1 - \frac{\text{len}(r)}{\text{len}(c)}\right) & \text{otherwise} \end{cases} \tag{2}$$

Lastly, the BLEU score for candidate $\mathbf{c}$ with respect to $\mathbf{r}_1, \ldots, \mathbf{r}_k$ is:

$$\text{BLEU} = BP \times \exp\left(\sum_{n=1}^{4} \lambda_n \log p_n \right) \tag{3}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are weights that sum to 1. The log here is natural log.

(a) (5 points) Please consider this example:

Source Sentence $\mathbf{s}$: **el amor todo lo puede**
Reference Translation $\mathbf{r}_1$: *love can always find a way*
Reference Translation $\mathbf{r}_2$: *love makes anything possible*
NMT Translation $\mathbf{c}_1$: *the love can always do*
NMT Translation $\mathbf{c}_2$: *love can make anything possible*

Please compute the BLEU scores for $\mathbf{c}_1$ and $\mathbf{c}_2$. Let $\lambda_i = 0.5$ for $i \in \{1, 2\}$ and $\lambda_i = 0$ for $i \in \{3, 4\}$ (**this means we ignore 3-grams and 4-grams**, i.e., don't compute p_3 or p_4). When computing BLEU scores, show your work (i.e., show your computed values for p_1, p_2, $\text{len}(c)$, $\text{len}(r)$ and BP). Note that the BLEU scores can be expressed between 0 and 1 or between 0 and 100. The code is using the 0 to 100 scale while in this question we are using the **0 to 1** scale.

Which of the two NMT translations is considered the better translation according to the BLEU Score? Do you agree that it is the better translation?

(b) (5 points) Our hard drive was corrupted and we lost Reference Translation $\mathbf{r}_2$. Please recompute BLEU scores for $\mathbf{c}_1$ and $\mathbf{c}_2$, this time with respect to $\mathbf{r}_1$ only. Which of the two NMT translations now receives the higher BLEU score? Do you agree that it is the better translation?

(c) (2 points) Due to data availability, NMT systems are often evaluated with respect to only a single reference translation. Please explain (in a few sentences) why this may be problematic.

(d) (2 points) List two advantages and two disadvantages of BLEU, compared to human evaluation, as an evaluation metric for Machine Translation.

[3] This definition of sentence-level BLEU score matches the `sentence_bleu()` function in the `nltk` Python package. Note that the NLTK function is sensitive to capitalization. In this question, all text is lowercased, so capitalization is irrelevant.
`http://www.nltk.org/api/nltk.translate.html#nltk.translate.bleu_score.sentence_bleu`