Due Date: Monday, 15th November, 11pm ET

Instructions

- *For all questions, show your work!*
- *Please use a document preparation system such as LaTeX, unless noted otherwise.*
- *Unless noted that questions are related, assume that notation and defintions for each question are self-contained and independent*
- *Submit your answers electronically via Gradescope.*
- *TAs for this assignment are* ***Leo Feng and Milad Aghajohari****.*

Question 1 (4-8-8)**.** Consider the following Bidirectional RNN:

$$\begin{aligned}\boldsymbol{h}_t^{(f)} &= \sigma(\boldsymbol{W}^{(f)}\boldsymbol{x}_t + \boldsymbol{U}^{(f)}\boldsymbol{h}_{t-1}^{(f)})\\ \boldsymbol{h}_t^{(b)} &= \sigma(\boldsymbol{W}^{(b)}\boldsymbol{x}_t + \boldsymbol{U}^{(b)}\boldsymbol{h}_{t+1}^{(b)})\\ \boldsymbol{y}_t &= \boldsymbol{V}^{(f)}\boldsymbol{h}_t^{(f)} + \boldsymbol{V}^{(b)}\boldsymbol{h}_t^{(b)}\end{aligned}$$

where the superscripts f and b correspond to the forward and backward RNNs respectively and σ denotes the logistic sigmoid function. Let $\boldsymbol{z}_t$ be the true target of the prediction $\boldsymbol{y}_t$ and consider the sum of squared loss $L = \sum_t L_t$ where $L_t = ||\boldsymbol{z}_t - \boldsymbol{y}_t||_2^2$.

In this question our goal is to obtain an expression for the gradients $\nabla_{\boldsymbol{W}^{(f)}} L$ and $\nabla_{\boldsymbol{U}^{(b)}} L$.

1. First, complete the following computational graph for this RNN, unrolled for 3 time steps (from $t = 1$ to $t = 3$). Label each node with the corresponding hidden unit and each edge with the corresponding weight. Note that it includes the initial hidden states for both the forward and backward RNNs.
2. Using total derivatives we can express the gradients $\nabla_{\boldsymbol{h}_t^{(f)}} L$ and $\nabla_{\boldsymbol{h}_t^{(b)}} L$ recursively in terms of $\nabla_{\boldsymbol{h}_{t+1}^{(f)}} L$ and $\nabla_{\boldsymbol{h}_{t-1}^{(b)}} L$ as follows:

$$\nabla_{\boldsymbol{h}_t^{(f)}} L = \nabla_{\boldsymbol{h}_t^{(f)}} L_t + \left(\frac{\partial \boldsymbol{h}_{t+1}^{(f)}}{\partial \boldsymbol{h}_t^{(f)}}\right)^\top \nabla_{\boldsymbol{h}_{t+1}^{(f)}} L$$

$$\nabla_{\boldsymbol{h}_t^{(b)}} L = \nabla_{\boldsymbol{h}_t^{(b)}} L_t + \left(\frac{\partial \boldsymbol{h}_{t-1}^{(b)}}{\partial \boldsymbol{h}_t^{(b)}}\right)^\top \nabla_{\boldsymbol{h}_{t-1}^{(b)}} L$$

   Derive an expression for $\nabla_{\boldsymbol{h}_t^{(f)}} L_t$, $\nabla_{\boldsymbol{h}_t^{(b)}} L_t$, $\frac{\partial \boldsymbol{h}_{t+1}^{(f)}}{\partial \boldsymbol{h}_t^{(f)}}$ and $\frac{\partial \boldsymbol{h}_{t-1}^{(b)}}{\partial \boldsymbol{h}_t^{(b)}}$.
3. Now derive $\nabla_{\boldsymbol{W}^{(f)}} L$ and $\nabla_{\boldsymbol{U}^{(b)}} L$ as functions of $\nabla_{\boldsymbol{h}_t^{(f)}} L$ and $\nabla_{\boldsymbol{h}_t^{(b)}} L$, respectively.
   Hint: It might be useful to consider the contribution of the weight matrices when computing the recurrent hidden unit at a particular time t and how those contributions might be aggregated.

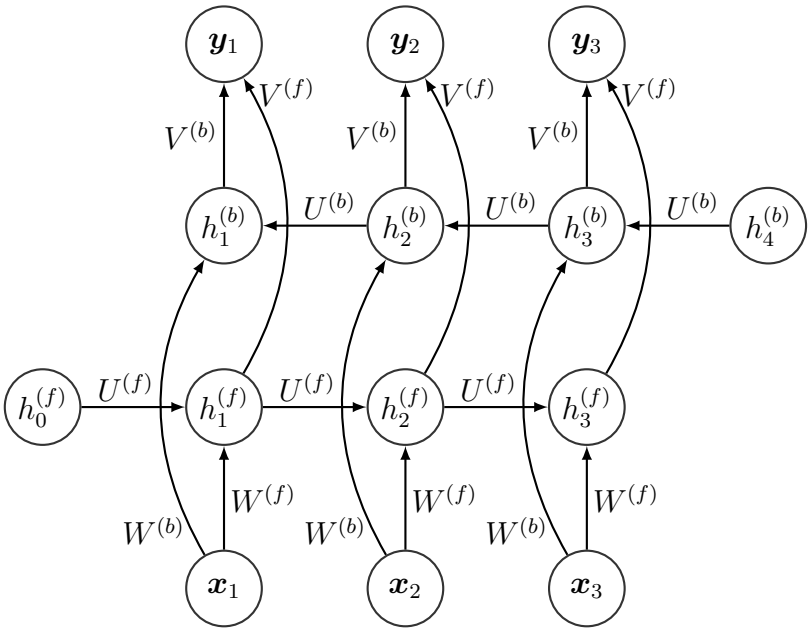


FIGURE 1 – Computational graph of the bidirectional RNN unrolled for three timesteps.

Answer 1. 2.

$$
\begin{aligned}
\nabla_{h_t^{(f)}} L_t &= \frac{\partial}{\partial h_t^{(f)}} L_t \\
&= \frac{\partial}{\partial h_t^{(f)}} \|z_t - y_t\|_2^2 \\
&= \frac{\partial}{\partial h_t^{(f)}} \|z_t - V^{(f)} h_t^{(f)} - V^{(b)} h_t^{(b)}\|_2^2 \qquad y_t = V^{(f)} h_t^{(f)} - V^{(b)} h_t^{(b)} \\
&= \frac{V^{(f)} h_t^{(f)} + V^{(b)} h_t^{(b)} - z_t}{\|z_t - V^{(f)} h_t^{(f)} - V^{(b)} h_t^{(b)}\|_2} \qquad \text{(from Homework 0)} \\
&= \frac{y_t - z_t}{\|z_t - y_t\|_2}
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{h_t^{(b)}} L_t &= \frac{\partial}{\partial h_t^{(b)}} L_t \\
&= \frac{\partial}{\partial h_t^{(b)}} \|z_t - y_t\|_2^2 \\
&= \frac{\partial}{\partial h_t^{(b)}} \|z_t - V^{(f)} h_t^{(f)} - V^{(b)} h_t^{(b)}\|_2^2 \qquad y_t = V^{(f)} h_t^{(f)} - V^{(b)} h_t^{(b)} \\
&= \frac{V^{(f)} h_t^{(f)} + V^{(b)} h_t^{(b)} - z_t}{\|z_t - V^{(f)} h_t^{(f)} - V^{(b)} h_t^{(b)}\|_2} \qquad \text{(from Homework 0)} \\
&= \frac{y_t - z_t}{\|z_t - y_t\|_2}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial h_{t+1}^{(f)}}{\partial h_t^{(f)}} &= \frac{\partial}{\partial h_t^{(f)}} h_{t+1}^{(f)} \\
&= \frac{\partial}{\partial h_t^{(f)}} \sigma(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)}) \qquad (h_{t+1}^{(f)} = w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)}) \\
&= \frac{\partial}{\partial h_t^{(f)}} \frac{1}{1 + e^{-(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)})}} \\
&= (-1)(1 + e^{-(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)})})^{-2} \times -U^{(f)} e^{-(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)})} \\
&= \frac{U^{(f)} e^{-(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)})}}{(1 + e^{-(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)})})^2} \\
&= U^{(f)} \times \frac{1}{1 + e^{-(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)})}} \times \frac{e^{-(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)})}}{1 + e^{-(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)})}} \\
&= U^{(f)} \times \sigma(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)}) \times (1 - \sigma(w^{(f)} x_{t+1} + U^{(f)} h_t^{(f)})) \\
&= U^{(f)} h_{t+1}^{(f)} (1 - h_{t+1}^{(f)})
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial h_{t-1}^{(b)}}{\partial h_t^{(b)}} &= \frac{\partial}{\partial h_t^{(b)}} h_{t-1}^{(b)} \\
&= \frac{\partial}{\partial h_t^{(b)}} \sigma(w^{(b)} x_{t-1} + U^{(f)} h_t^{(b)}) \qquad (h_{t-1}^{(b)} = w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)}) \\
&= \frac{\partial}{\partial h_t^{(b)}} \frac{1}{1 + e^{-(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)})}} \\
&= (-1)(1 + e^{-(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)})})^{-2} \times -U^{(f)} e^{-(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)})} \\
&= \frac{U^{(b)} e^{-(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)})}}{(1 + e^{-(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)})})^2} \\
&= U^{(b)} \times \frac{1}{1 + e^{-(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)})}} \times \frac{e^{-(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)})}}{1 + e^{-(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)})}} \\
&= U^{(b)} \times \sigma(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)}) \times (1 - \sigma(w^{(b)} x_{t-1} + U^{(b)} h_t^{(b)})) \\
&= U^{(b)} h_{t-1}^{(f)} (1 - h_{t-1}^{(b)})
\end{aligned}
$$

3.

$$
\begin{aligned}
\nabla_{w^{(f)}} L &= \sum_{t=1}^{T} \frac{\partial L_t}{\partial h_t^{(f)}} \frac{\partial h_t^{(f)}}{\partial w^{(f)}} \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \frac{\partial h_t^{(f)}}{\partial w^{(f)}} \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \frac{\partial}{\partial w^{(f)}} h_t^{(f)} \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \frac{\partial}{\partial w^{(f)}} \sigma(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)}) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \frac{\partial}{\partial w^{(f)}} \frac{1}{1 + e^{-(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)})}} \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \left(\frac{-1}{(1 + e^{-(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)})})^2} \times -x_t e^{-(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)})} \right) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \left(\frac{x_t e^{-(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)})}}{(1 + e^{-(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)})})^2} \right) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \left(x_t \times \frac{1}{1 + e^{-(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)})}} \times \frac{e^{-(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)})}}{1 + e^{-(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)})}} \right) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \Big(x_t \times \sigma(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)}) \times (1 - \sigma(w^{(f)} x_t + U^{(f)} h_{t-1}^{(f)})) \Big) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \Big(x_t \times h_t^{(f)} \times (1 - h_t^{(f)}) \Big)
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{w^{(b)}} L &= \sum_{t=1}^{T} \frac{\partial L_t}{\partial h_t^{(b)}} \frac{\partial h_t^{(b)}}{\partial w^{(b)}} \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(b)}} L_t \frac{\partial h_t^{(b)}}{\partial w^{(b)}} \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(b)}} L_t \frac{\partial}{\partial w^{(b)}} h_t^{(b)} \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(b)}} L_t \frac{\partial}{\partial w^{(b)}} \sigma(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)}) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(b)}} L_t \frac{\partial}{\partial w^{(b)}} \frac{1}{1 + e^{-(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)})}} \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(b)}} L_t \left(\frac{-1}{(1 + e^{-(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)})})^2} \times -h_{t-1}^{(b)} e^{-(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)})} \right) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \left(\frac{h_{t-1}^{(b)} e^{-(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)})}}{(1 + e^{-(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)})})^2} \right) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(f)}} L_t \left(h_{t-1}^{(b)} \times \frac{1}{1 + e^{-(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)})}} \times \frac{e^{-(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)})}}{1 + e^{-(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)})}} \right) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(b)}} L_t \left(h_{t-1}^{(b)} \times \sigma(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)}) \times (1 - \sigma(w^{(b)} x_t + U^{(b)} h_{t+1}^{(b)})) \right) \\
&= \sum_{t=1}^{T} \nabla_{h_t^{(b)}} L_t \left(h_{t-1}^{(b)} \times h_t^{(b)} \times (1 - h_t^{(b)}) \right)
\end{aligned}
$$

Question 2 (1-12-2-6)**.** Suppose that we have a vocabulary containing N possible words, including a special token `<BOS>` to indicate the beginning of a sentence. Recall that in general, a language model with a full context can be written as

$$p(w_1, w_2, \ldots, w_T \mid w_0) = \prod_{t=1}^{T} p(w_t \mid w_0, \ldots, w_{t-1}).$$

We will use the notation $\boldsymbol{w}_{0:t-1}$ to denote the (partial) sequence $(w_0, \ldots, w_{t-1})$. Once we have a fully trained language model, we would like to generate realistic sequences of words from our language model, starting with our special token `<BOS>`. In particular, we might be interested in generating the most likely sequence $\boldsymbol{w}_{1:T}^{\star}$ under this model, defined as

$$\boldsymbol{w}_{1:T}^{\star} = \arg\max_{\boldsymbol{w}_{1:T}} p(\boldsymbol{w}_{1:T} \mid w_0 = \texttt{<BOS>}).$$

For clarity we will drop the explicit conditioning on w_0, assuming from now on that the sequences always start with the `<BOS>` token.

2.1 How many possible sequences of length $T+1$ starting with the token `<BOS>` can be generated in total ? Give an exact expression, without the O notation. Note that the length $T+1$ here includes the `<BOS>` token.

2.2 In this question only, we will assume that our language model satisfies the *Markov property*

$$\forall\, t > 0,\ p(w_t \mid w_0, \ldots, w_{t-1}) = p(w_t \mid w_{t-1}).$$

Moreover, we will assume that the model is time-homogeneous, meaning that there exists a matrix $\boldsymbol{P} \in \mathbb{R}^{N\times N}$ such that $\forall\, t > 0,\ p(w_t = j \mid w_{t-1} = i) = [\boldsymbol{P}]_{i,j} = P_{ij}$.

2.2.a Let $\boldsymbol{\delta}_t \in \mathbb{R}^N$ be the vector of size N defined for $t > 0$ as

$$\delta_t(j) = \max_{\boldsymbol{w}_{1:t-1}} p(\boldsymbol{w}_{1:t-1}, w_t = j),$$

where $\boldsymbol{w}_{1:0} = \emptyset$. Using the following convention for $\boldsymbol{\delta}_0$

$$\delta_0(j) = \begin{cases} 1 & \text{if } j = \texttt{<BOS>} \\ 0 & \text{otherwise,} \end{cases}$$

give the recurrence relation satisfied by $\boldsymbol{\delta}_t$ (for $t > 0$).

2.2.b Show that $w_T^\star = \arg\max_j \delta_T(j)$.

2.2.c Let $\boldsymbol{a}_t \in \{1, \ldots, N\}^N$ be the vector of size N defined for $t > 0$ as

$$a_t(j) = \arg\max_i P_{ij}\delta_{t-1}(i).$$

Show that $\forall\, 0 < t < T$, $w_t^\star = a_{t+1}(w_{t+1}^\star)$.

2.2.d Using the previous questions, write the pseudo-code of an algorithm to compute the sequence $\boldsymbol{w}_{1:T}^\star$ using dynamic programming. This is called *Viterbi decoding*.

2.2.e What is the time complexity of this algorithm ? Its space complexity ? Write the algorithmic complexities using the O notation. Comment on the efficiency of this algorithm compared to naively searching for $\boldsymbol{w}_{1:T}^\star$ by enumerating all possible sequences (based on your answer to question 1).

2.2.f When the size of the vocabulary N is not too large, can you use this algorithm to generate the most likely sequence of a language model defined by a recurrent neural network, such as a GRU or an LSTM ? Why ? Why not ? If not, name a language model you can apply this algorithm to.

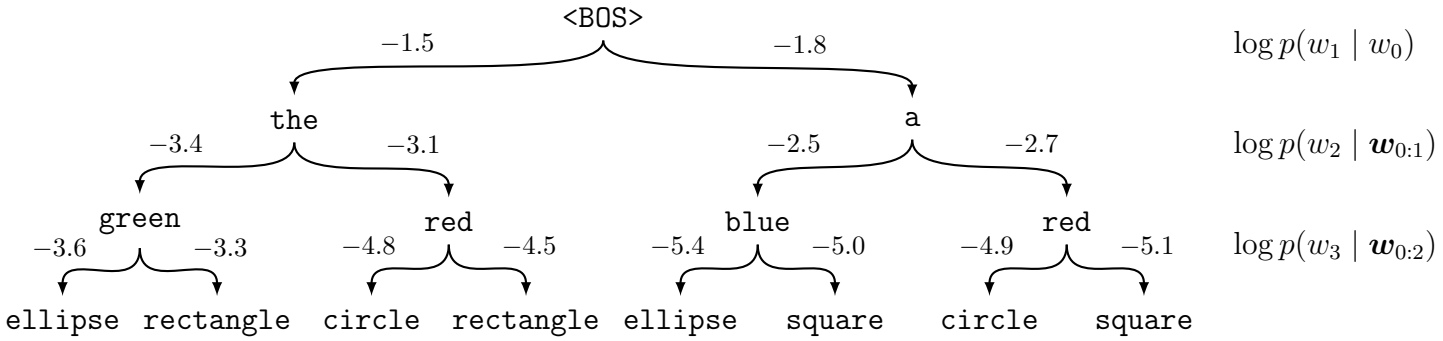
2.3 For real-world applications, the size of the vocabulary N can be very large (e.g. $N = 30$k for BERT, $N = 50$k for GPT-2), making even dynamic programming impractical. In order to generate B sequences having high likelihood, one can use a heuristic algorithm called *Beam search decoding*, whose pseudo-code is given in Algorithm 1 below

Algorithm 1: Beam search decoding

Input: A language model $p(\boldsymbol{w}_{1:T} \mid w_0)$, the beam width B
Output: B sequences $\boldsymbol{w}_{1:T}^{(b)}$ for $b \in \{1, \dots, B\}$
Initialization: $w_0^{(b)} \leftarrow$ `<BOS>` for all $b \in \{1, \dots, B\}$
Initial log-likelihoods: $l_0^{(b)} \leftarrow 0$ for all $b \in \{1, \dots, B\}$
for $t = 1$ **to** T **do**
  for $b = 1$ **to** B **do**
    for $j = 1$ **to** N **do**
      $s_b(j) \leftarrow l_{t-1}^{(b)} + \log p(w_t = j \mid \boldsymbol{w}_{0:t-1}^{(b)})$
  for $b = 1$ **to** B **do**
    Find (b', j) such that $s_{b'}(j)$ is the b-th largest score
    Save the partial sequence b': $\widetilde{\boldsymbol{w}}_{0:t-1}^{(b)} \leftarrow \boldsymbol{w}_{0:t-1}^{(b')}$
    Add the word j to the sequence b: $w_t^{(b)} \leftarrow j$
    Update the log-likelihood: $l_t^{(b)} \leftarrow s_{b'}(j)$
  Assign the partial sequences: $\boldsymbol{w}_{0:t-1}^{(b)} \leftarrow \widetilde{\boldsymbol{w}}_{0:t-1}^{(b)}$ for all $b \in \{1, \dots, B\}$

What is the time complexity of Algorithm 1 ? Its space complexity ? Write the algorithmic complexities using the O notation, as a function of T, B, and N. Is this a practical decoding algorithm when the size of the vocabulary is large ?

2.4 The different sequences that can be generated with a language model can be represented as a tree, where the nodes correspond to words and edges are labeled with the log-probability $\log p(w_t \mid \boldsymbol{w}_{0:t-1})$, depending on the path $\boldsymbol{w}_{0:t-1}$. In this question, consider the following language model (where the low probability paths have been removed for clarity)



2.4.a If you were given the whole tree, including the log-probabilities of all the missing branches (e.g. $\log p(w_2 =$ `a` $\mid w_0 =$ `<BOS>`$, w_1 =$ `red`$)$), could you apply Viterbi decoding from question 2 to this language model in order to find the most likely sequence $\boldsymbol{w}_{1:3}^\star$? Why ? Why not ? Find $\boldsymbol{w}_{1:3}^\star$, together with its corresponding log-likelihood $\log p(\boldsymbol{w}_{1:3}^\star) = \max_{\boldsymbol{w}_{1:3}} \log p(\boldsymbol{w}_{1:3})$.

2.4.b *Greedy decoding* is a simple algorithm where the next word $\overline{w}_t$ is selected by maximizing the conditional probability $p(w_t \mid \overline{\boldsymbol{w}}_{0:t-1})$ (with $\overline{w}_0 =$ `<BOS>`)

$$\overline{w}_t = \arg\max_{w_t} \log p(w_t \mid \overline{\boldsymbol{w}}_{0:t-1}).$$

Find $\overline{\boldsymbol{w}}_{1:3}$ using greedy decoding on this language model, and its log-likelihood $\log p(\overline{\boldsymbol{w}}_{1:3})$.

2.4.c Apply beam search decoding (question 3) with a beam width $B = 2$ to this language model, and find $\boldsymbol{w}_{1:3}^{(1)}$ and $\boldsymbol{w}_{1:3}^{(2)}$, together with their respective log-likelihoods.

2.4.d Compare the behaviour of these 3 decoding algorithms on this language model (in particular greedy decoding vs. maximum likelihood, and beam search decoding vs. the other two). How can you mitigate the limitations of beam search ?

Answer 2. 1. The number of possible sequences for $T+1$, starting with the <BOS> token is N^T. N is the number of words in our vocabulary and T is the length of a single sequence.

$$\prod_{t=1}^{T} N = N \times N \times N... \qquad \text{(T times)}$$
$$= N^T$$

(a)
$$\delta_t(j) = \max_i P_{ij} \times \delta_{t-1}(i)$$

(b) At $t = 0$, the most likely word would be the <BOS> token. δ_0 would returns 1 given j as the <BOS> token, otherwise 0.

The subsequent words following the <BOS> token form the most likely sequence as δ_t ($\forall t > 0$) maximizes the joint distribution of the $t-1$ sequence given the t^{th} word. The word which maximizes the likeliness of the sequence is then chosen as the t^{th} word (choosing the word which maximizes the joint distribution of the entire sequence).

Hence, it return the most lilely T^{th} word.

$$w_T^* = \arg\max_j \delta_T(j)$$

(c) $a_t(j)$ maximizes the likeliness of the word preceding the t^{th} word denoted by j in the equation. In other words, a_t gives the most likely preceding word to the input word j. As such, if we wished to find the most likely current t^{th} word, passing as input to a_{t+1}, the most likely next $(t+1)^{th}$ word would result in us receiving the most likely t^{th}.

Hence,
$$w_t^* = a_{t+1}(w_{t+1}^*$$

Question 3 (4-6-6)**.** This question is about normalization techniques.

3.1 Batch normalization, layer normalization and instance normalization all involve calculating the mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma^2}$ with respect to different subsets of the tensor dimensions. Given the following 3D tensor, calculate the corresponding mean and variance tensors for each normalization technique: $\boldsymbol{\mu}_{batch}$, $\boldsymbol{\mu}_{layer}$, $\boldsymbol{\mu}_{instance}$, $\boldsymbol{\sigma}^2_{batch}$, $\boldsymbol{\sigma}^2_{layer}$, and $\boldsymbol{\sigma}^2_{instance}$.

$$\left[\begin{bmatrix}1,3,2\\1,2,3\end{bmatrix}, \begin{bmatrix}3,3,2\\2,4,4\end{bmatrix}, \begin{bmatrix}4,2,2\\1,2,4\end{bmatrix}, \begin{bmatrix}3,3,2\\3,3,2\end{bmatrix}\right]$$

The size of this tensor is 4 x 2 x 3 which corresponds to the batch size, number of channels, and number of features respectively.

3.2 For the next two subquestions, we consider the following parameterization of a weight vector $\boldsymbol{w}$:

$$\boldsymbol{w} := \gamma \frac{\boldsymbol{u}}{||\boldsymbol{u}||}$$

where γ is scalar parameter controlling the magnitude and $\boldsymbol{u}$ is a vector controlling the direction of $\boldsymbol{w}$.

Consider one layer of a neural network, and omit the bias parameter. To carry out batch normalization, one normally standardizes the preactivation and performs elementwise scale and shift $\hat{y} = \gamma \cdot \frac{y-\mu_y}{\sigma_y} + \beta$ where $y = \boldsymbol{u}^\top \boldsymbol{x}$. Assume the data $\boldsymbol{x}$ (a random vector) is whitened ($\text{Var}(\boldsymbol{x}) = \boldsymbol{I}$) and centered at 0 ($\mathbb{E}[\boldsymbol{x}] = \boldsymbol{0}$). Show that $\hat{y} = \boldsymbol{w}^\top \boldsymbol{x} + \beta$.

3.3 Show that the gradient of a loss function $L(\boldsymbol{u}, \gamma, \beta)$ with respect to $\boldsymbol{u}$ can be written in the form $\nabla_{\boldsymbol{u}} L = s\boldsymbol{W}^\perp \nabla_{\boldsymbol{w}} L$ for some s, where $\boldsymbol{W}^\perp = \left(\boldsymbol{I} - \frac{\boldsymbol{u}\boldsymbol{u}^\top}{||\boldsymbol{u}||^2}\right)$. Note that [1] $\boldsymbol{W}^\perp \boldsymbol{u} = \boldsymbol{0}$.

1. As a side note: $\boldsymbol{W}^\perp$ is an orthogonal complement that projects the gradient away from the direction of $\boldsymbol{w}$, which is usually (empirically) close to a dominant eigenvector of the covariance of the gradient. This helps to condition the landscape of the objective that we want to optimize.

Answer 3. 1. B = 4, C = 2, F= 3

$$\mu_{batch} = \frac{1}{B \times F} \sum_{i=1}^{B} \sum_{j=1}^{F} x_{ij}$$
$$= \begin{bmatrix} \frac{1+3+2+3+3+2+4+2+2+3+3+2}{12} \\ \frac{1+2+3+2+4+4+1+2+4+3+3+2}{12} \end{bmatrix}$$
$$= \begin{bmatrix} 2.5 \\ 2.583 \end{bmatrix}$$
$$\sigma^2_{batch} = \frac{1}{B \times F} \sum_{i=1}^{B} \sum_{j=1}^{F} (x_{ij} - \mu)^2$$
$$= \begin{bmatrix} 0.5833 \\ 1.0763 \end{bmatrix}$$
$$\mu_{layer} = \frac{1}{C \times F} \sum_{i=1}^{C} \sum_{j=1}^{F} x_{ij}$$
$$= \begin{bmatrix} \frac{1+3+2+1+2+3+4}{6} & \frac{3+3+2+2+4+4}{6} & \frac{4+2+2+1+2+4}{6} & \frac{3+3+2+3+3+2}{6} \end{bmatrix}$$
$$= \begin{bmatrix} 2.0 & 3.0 & 2.5 & 2.6667 \end{bmatrix}$$
$$\sigma^2_{layer} = \frac{1}{C \times F} \sum_{i=1}^{C} \sum_{j=1}^{F} (x_{ij} - \mu)^2$$
$$= \begin{bmatrix} 0.6667 & 0.6667 & 1.25 & 0.2222 \frac{3+3+2+3+3+2}{6} \end{bmatrix}$$
$$\mu_{instance} = \frac{1}{F} \sum_{i=1}^{F} x_i$$
$$= \begin{bmatrix} \begin{bmatrix} \frac{1+3+2}{3} \\ \frac{1+2+3}{3} \end{bmatrix} \begin{bmatrix} \frac{3+3+2}{3} \\ \frac{2+4+4}{3} \end{bmatrix} \begin{bmatrix} \frac{4+2+2}{3} \\ \frac{1+2+4}{3} \end{bmatrix} \begin{bmatrix} \frac{3+3+2}{3} \\ \frac{3+3+2}{3} \end{bmatrix} \end{bmatrix}$$
$$= \begin{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 2.6667 \\ 3.333 \end{bmatrix} \begin{bmatrix} 2.6667 \\ 2.333 \end{bmatrix} \begin{bmatrix} 2.66667 \\ 2.66667 \end{bmatrix} \end{bmatrix}$$
$$\sigma^2_{layer} = \frac{1}{C \times F} \sum_{i=1}^{C} \sum_{j=1}^{F} (x_{ij} - \mu)^2$$
$$= \begin{bmatrix} \begin{bmatrix} 0.6667 \\ 0.6667 \end{bmatrix} \begin{bmatrix} 0.2222 \\ 0.8889 \end{bmatrix} \begin{bmatrix} 0.8889 \\ 1.5556 \end{bmatrix} \begin{bmatrix} 0.2222 \\ 0.2222 \end{bmatrix} \end{bmatrix}$$

2.

$$
\begin{aligned}
\hat{y} &= \gamma \times \frac{y - \mu_y}{\sigma_y} + \beta \\
&= \gamma \times \frac{u^T x - \mu_y}{\sigma_y} + \beta \\
&= \gamma \times \frac{u^T x}{\sigma_y} + \beta \qquad (\mathbb{E}[x] = 0) \\
&= \gamma \times \frac{u^T x}{\sqrt{Var(y)}} + \beta \\
&= \gamma \times \frac{u^T x}{\sqrt{Var(u^T x)}} + \beta \\
&= \gamma \times \frac{u^T x}{\sqrt{(u^2)^T Var(x)}} + \beta \qquad (Var(ax) = a^2 Var(x)) \\
&= \gamma \times \frac{u^T x}{\sqrt{u^T u \cdot Var(x)}} + \beta \\
&= \gamma \times \frac{u^T x}{\sqrt{u^T u}} + \beta \qquad (Var(x) = I) \\
&= \gamma \times \frac{u^T x}{\|u\|} + \beta \\
&= \gamma \times \frac{u^T}{\|u\|} \cdot x + \beta \\
&= w^T x + \beta
\end{aligned}
$$

3.

$$
\begin{aligned}
\frac{\partial L(u,\gamma,\beta)}{\partial u} &= \frac{\partial L(u,\gamma,\beta)}{\partial w}\frac{\partial w}{\partial u} \\
&= \nabla_w L \frac{\partial w}{\partial u} \\
&= \nabla_w L \frac{\partial}{\partial u} \gamma \frac{u}{\|u\|} \\
&= \gamma \nabla_w L \frac{\|u\| \frac{\partial u}{\partial u} - \frac{\partial \|u\|}{\partial u} u}{\|u^2\|} \\
&= \frac{\gamma}{\|u\|} \nabla_w L \left(\frac{\partial u}{\partial u} - \frac{u}{\|u\|} \frac{\partial \|u\|}{\partial u} \right) \\
&= \frac{\gamma}{\|u\|} \nabla_w L \left(\frac{\partial u}{\partial u} - \frac{u}{\|u\|} \frac{u^T}{\|u\|} \frac{\partial u}{\partial u} \right) \\
&= \frac{\gamma}{\|u\|} \nabla_w L \left(I - \frac{u u^T}{\|u\|^2} \cdot I \right) \\
&= s W^{\perp} \nabla_w L \qquad (s = \frac{\gamma}{\|u\|})
\end{aligned}
$$

Question 4 (7-5-5-3)**.** The point of this question is to understand and compare the effects of different regularizers (specifically dropout and weight decay) on the weights of a network. Consider a linear regression problem with input data $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, weights $\boldsymbol{w} \in \mathbb{R}^{d \times 1}$ and targets $\boldsymbol{y} \in \mathbb{R}^{n \times 1}$. Suppose that dropout is applied to the input (with probability $1-p$ of dropping the unit i.e. setting it to 0). Let $\boldsymbol{R} \in \mathbb{R}^{n \times d}$ be the dropout mask such that $\boldsymbol{R}_{ij} \sim \text{Bern}(p)$ is sampled i.i.d. from the Bernoulli distribution.

For a squared error loss function with dropout, we then have:

$$L(\boldsymbol{w}) = ||\boldsymbol{y} - (\boldsymbol{X} \odot \boldsymbol{R})\boldsymbol{w}||^2$$

4.1 Let Γ be a diagonal matrix with $\Gamma_{ii} = (\boldsymbol{X}^\top \boldsymbol{X})_{ii}^{1/2}$. Show that the *expectation (over $\boldsymbol{R}$)* of the loss function can be rewritten as $\mathbb{E}[L(\boldsymbol{w})] = ||\boldsymbol{y} - p\boldsymbol{X}\boldsymbol{w}||^2 + p(1-p)||\Gamma\boldsymbol{w}||^2$. *Hint: Note we are trying to find the expectation over a squared term and use* $\text{Var}(Z) = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2$.

4.2 Show that the solution $\boldsymbol{w}^{\text{dropout}}$ that minimizes the expected loss from question 4.1 satisfies

$$p\boldsymbol{w}^{\text{dropout}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda^{\text{dropout}}\Gamma^2)^{-1}\boldsymbol{X}^\top \boldsymbol{y}$$

where λ^{dropout} is a regularization coefficient depending on p. How does the value of p affect the regularization coefficient, λ^{dropout} ?

4.3 Express the loss function for a linear regression problem without dropout and with L^2 regularization, with regularization coefficient λ^{L_2}. Derive its closed form solution $\boldsymbol{w}^{L_2}$.

4.4 Compare the results of 4.2 and 4.3: identify specific differences in the equations you arrive at, and discuss qualitatively what the equations tell you about the similarities and differences in the effects of weight decay and dropout (1-3 sentences).

Answer 4.