Assignment II
COMP 5900 Advanced Machine Learning
Winter 2021

Note: Submit your assignment as a single typed pdf along with your code as a .zip file in Culearn. No handwritten note will be accepted. Do not include the pdf in the zip file. Your report (the pdf) must be self-contained.

Part I LSTM and RNN

In this part you will use RNN and LSTM for sentiment analysis of IMDB movies reviews. You will use Colab environment that you are familiar with from Assignment 1. Please download the Jupiter Notebook provided on the course website and save it in you Google Drive and open it in Colab. Go to Edit -> Notebook Setting and set the Runtime Type to Python 3 and Hardware Accelerator to GPU. The provided code classifies movie reviews to either positive or negative using 2 bidirectional LSTM layers stacked on top of each other.

Q1.1 (1 Mark) Run the cells one-by-one and follow the instructions provided in the notebook. Train the network for 5 epochs. How many parameters are in the two bi-LSTM layers? What is the accuracy on the test set? Write your answers in the table below. Hint: Your two-level LSTM looks like this:

```
LSTM(100, 200, num_layers=2, dropout=0.5, bidirectional=True)
```

Q1.2 (1 Mark) Modify the code such that it implements a 2-level bi-RNN. Train the network for 5 epochs. How many parameters are in the two bi-RNN layers? What is the accuracy on the test set? Write your answers in the table below. Hint: your two-level RNN should look like this:

```
RNN(100, 200, num_layers=2, dropout=0.5, bidirectional=True)
```

Q1.3 (1 Mark) Modify the code such that it implements a bi-RNN. Train the network for 5 epochs. How many parameters are in the bi-RNN layer? What is the accuracy on the test set? Write your answers in the table below. Hint: your one-level RNN should look like this:

```
RNN(100, 200, dropout=0.5, bidirectional=True)
```

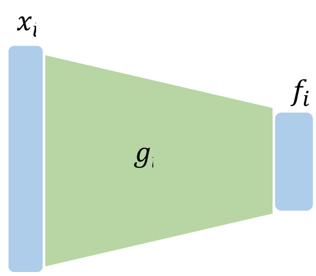
Q1.4 (1 Mark) Modify the code such that it implements a RNN. Train the network for 5 epochs. How many parameters are in the RNN layer? What is the accuracy on the test set? Write your answers in the table below. Hint: your one-level RNN should look like this:

```
RNN(100, 200, dropout=0.5)
```

	Total # of parameters in RNN/LSTM layer(s)	Test accuracy %
2-level bi-LSTM		
2-level bi-RNN		
bi-RNN		
RNN		

Part II Triplet Loss

In this Part you are going to explore how "triplet loss" works. There is no implementation in this part. You will learn more about triplet loss in class on March 1. Triplet loss is a loss function for artificial neural networks where a baseline (anchor) input is compared to a positive (truthy) input and a negative (falsely) input. The distance from the baseline (anchor) input to the positive (truthy) input is minimized, and the distance from the baseline (anchor) input to the negative (falsely) input is maximized. The loss function can be described using a Euclidean distance function. The triplet is formed by drawing an anchor input, a positive input that describes the same entity as the anchor entity, and a negative input that does not describe the same entity as the anchor entity. These inputs are then run through the network, and the outputs are used in the loss function.



Let g be an arbitrary model (e.g. neural network) and f_i be the embedding for the input sample x_i. We are going to learn the embeddings f_i. For simplicity, assume that g consists of only a single dense layer with parameters W. Therefor $f_i = Wx_i$.

Let $t = (f_a, f_p, f_n)$ be a triplet where f_a, f_p *and* f_n are embeddings corresponding to "anchor", "positive" and "negative" samples respectively. Let's define the loss for the t-th triplet as

$$\mathrm{L}^t = L^t(f_a, f_p, f_n) = \max\left(0, \frac{1}{2}\left\|f_a - f_p\right\|^2 + \alpha - \frac{1}{2}\|f_a - f_n\|^2\right), \tag{1}$$

and the overall loss $\mathrm{L} = \sum_{t=1}^{T} L^t$ where T is the total number of triplets.

Q2 (1.5 Marks) Determine $\frac{\partial L^t}{\partial f_a}, \frac{\partial L^t}{\partial f_p}$ and $\frac{\partial L^t}{\partial f_n}$, if $\frac{1}{2}\left\|f_a - f_p\right\|^2 + \alpha - \frac{1}{2}\|f_a - f_n\|^2 \le 0$.

Q3 (1.5 Marks) Determine $\frac{\partial L^t}{\partial f_a}, \frac{\partial L^t}{\partial f_p}$ and $\frac{\partial L^t}{\partial f_n}$, if $\frac{1}{2}\left\|f_a - f_p\right\|^2 + \alpha - \frac{1}{2}\|f_a - f_n\|^2 > 0$.

Assume that our dataset has six samples from two classes. Here is our dataset D.

Sample	x_1	x_2	x_3	x_4	x_5	x_6
Class Label	1	1	1	2	2	2

Q4 (1 Mark) How many unique triplets can be generated from dataset D?

Assume that only 8 out of all triplets satisfy $\frac{1}{2}\left\|f_a - f_p\right\|^2 + \alpha - \frac{1}{2}\|\boldsymbol{f}_a - f_n\|^2 > 0$. Here are the 8 triplets.

$t_1 = (x_1, x_2, x_5)$, $t_2 = (x_1, x_3, x_4)$, $t_3 = (x_2, x_1, x_6)$, $t_4 = (x_2, x_3, x_4)$, $t_5 = (x_3, x_1, x_5)$

$t_6 = (x_5, x_6, x_2)$, $t_7 = (x_6, x_4, x_3)$, $t_8 = (x_6, x_4, x_2)$.

We are going to update the weights W using gradient decent. The general procedure is as follows:

1. Perform the forward pass for the input data and compute the embeddings $f_1, \ldots, f_6$.
2. Compute the L^t for every triplet using Equation (1) and compute the overall loss L.
3. Compute the gradient of loss L for every training sample i.e. compute $\Delta_1, \cdots, \Delta_6$ where $\Delta_i = \frac{\partial L}{\partial f_i}\frac{\partial f_i}{\partial W}$.
4. Update W using gradient decent: $W^{new} = W^{old} - \eta\frac{1}{6}\sum_{i=1}^{6}\Delta_i$.

Let's take a closer look at step 3. In our simple network $f_i = Wx_i$. Therefore $\frac{\partial f_i}{\partial W} = x_i$. For computing $\Delta_i = \frac{\partial L}{\partial f_i}x_i$ we need to compute gradient of loss L with respect to the embedding of every sample, i.e. we need to compute $\frac{\partial L}{\partial f_1}, \cdots, \frac{\partial L}{\partial f_6}$. We want to minimize the loss for all triplets, so the loss is $\mathrm{L} = \sum_{t=1}^{8} L^t$ where L^t is the triplet loss that works on three samples of the triplet t. In Q6, you are going to compute $\frac{\partial L}{\partial f_1}, \cdots, \frac{\partial L}{\partial f_6}$ for our toy dataset and in Q5 you will compute some intermediate values needed to do so.

Q5 (4 Marks)

Q5.1 Determine $\frac{\partial L^1}{\partial f_1}, \frac{\partial L^1}{\partial f_2}, \frac{\partial L^1}{\partial f_3}, \frac{\partial L^1}{\partial f_4}, \frac{\partial L^1}{\partial f_5}$ and $\frac{\partial L^1}{\partial f_6}$.

Q5.2 Determine $\frac{\partial L^2}{\partial f_1}, \frac{\partial L^2}{\partial f_2}, \frac{\partial L^2}{\partial f_3}, \frac{\partial L^2}{\partial f_4}, \frac{\partial L^2}{\partial f_5}$ and $\frac{\partial L^2}{\partial f_6}$.

Q5.3 Determine $\frac{\partial L^3}{\partial f_1}, \frac{\partial L^3}{\partial f_2}, \frac{\partial L^3}{\partial f_3}, \frac{\partial L^3}{\partial f_4}, \frac{\partial L^3}{\partial f_5}$ and $\frac{\partial L^3}{\partial f_6}$.

Q5.4 Determine $\frac{\partial L^4}{\partial f_1}, \frac{\partial L^4}{\partial f_2}, \frac{\partial L^4}{\partial f_3}, \frac{\partial L^4}{\partial f_4}, \frac{\partial L^4}{\partial f_5}$ and $\frac{\partial L^4}{\partial f_6}$.

Q5.5 Determine $\frac{\partial L^5}{\partial f_1}, \frac{\partial L^5}{\partial f_2}, \frac{\partial L^5}{\partial f_3}, \frac{\partial L^5}{\partial f_4}, \frac{\partial L^5}{\partial f_5}$ and $\frac{\partial L^5}{\partial f_6}$.

Q5.6 Determine $\frac{\partial L^6}{\partial f_1}, \frac{\partial L^6}{\partial f_2}, \frac{\partial L^6}{\partial f_3}, \frac{\partial L^6}{\partial f_4}, \frac{\partial L^6}{\partial f_5}$ and $\frac{\partial L^6}{\partial f_6}$.

Q5.7 Determine $\frac{\partial L^7}{\partial f_1}, \frac{\partial L^7}{\partial f_2}, \frac{\partial L^7}{\partial f_3}, \frac{\partial L^7}{\partial f_4}, \frac{\partial L^7}{\partial f_5}$ and $\frac{\partial L^7}{\partial f_6}$.

Q5.8 Determine $\frac{\partial L^8}{\partial f_1}, \frac{\partial L^8}{\partial f_2}, \frac{\partial L^8}{\partial f_3}, \frac{\partial L^8}{\partial f_4}, \frac{\partial L^8}{\partial f_5}$ and $\frac{\partial L^8}{\partial f_6}$.

Q6 (3 Marks) Determine $\frac{\partial L}{\partial f_1}, \frac{\partial \mathrm{L}}{\partial f_2}, \frac{\partial L}{\partial f_3}, \frac{\partial \mathrm{L}}{\partial f_4}, \frac{\partial \mathrm{L}}{\partial f_5}$ and $\frac{\partial \mathrm{L}}{\partial f_6}$. The final expression should be in its simplest form.

$Hint: for\ example\ \frac{\partial L}{\partial f_1} = \frac{\partial L^1}{\partial f_1} + \frac{\partial L^2}{\partial f_1} + \cdots + \frac{\partial L^8}{\partial f_1}.$