Deep learning- Assignment 2

Tamar Mano 206295917
Orgad Shlishman 303142897

Theoretical part

1. In RNN, and specifically when working on sequence-to-sequence tasks, one problem is the varying length of the sequences.

   a. Describe two ways (or more) to deal with variable-length **input** sequences.

      i. **Padding:** This method is getting input sequence and pad it (if needed) to fit to another chosen size. The idea is to pad all inputs that are smaller than the largest one. This way, all inputs will have same length.

      ii. **Bucketing:** This method is grouping sequences by their length, and then, calling doing all the needed operations per group of specific sequence length.

   b. Describe two ways (or more) to deal with variable-length output sequences.

      i. **Length Control:** This approach involves constraining the output sequence to a certain length or length range, either by adding a penalty term to the loss function or by adding a constraint during decoding.

      ii. **Fixed-length output sequence**: This approach can be achieved by truncating longer sequences or padding shorter ones.

2. Name two advantages of GRU over LSTM.

   i. **Simpler architecture**: The main difference between the architectures of LSTMs and GRUs are the gates – 2 in GRUs and 3 in LSTMs. Thus, training GRU is easier than training LSTM, and the performance are similar.

   ii. **Fewer parameters**: GRUs has fewer parameters in comparison to LSTMs, so it's less prone to overfitting.

3. A network based on a single LSTM cell uses a vector of size 200 to describe the current state. Its input size are 200 sized vectors. Considering only parameters related to the cell, how many parameters does it have?

The number of parameters depends on input size, layer connections and weights.
We have 4 FC layers, but each one of them (f, i, o, g) is doubled.
Which means:

$$\begin{aligned}
\mathbf{i}_{(t)} &= \sigma(\mathbf{W}_{xi}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hi}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_{i}) \\
\mathbf{f}_{(t)} &= \sigma(\mathbf{W}_{xf}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hf}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_{f}) \\
\mathbf{o}_{(t)} &= \sigma(\mathbf{W}_{xo}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{ho}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_{o}) \\
\mathbf{g}_{(t)} &= \tanh\,(\mathbf{W}_{xg}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hg}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_{g})
\end{aligned}$$

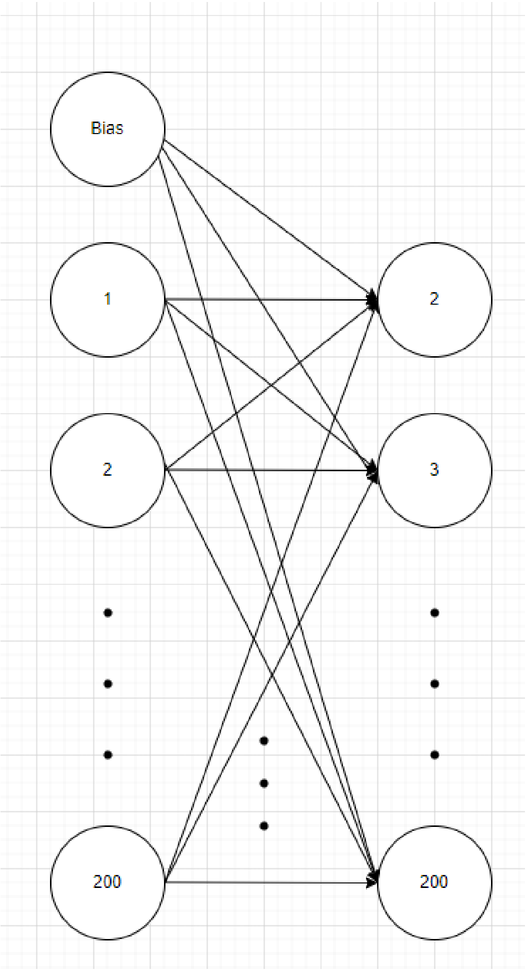
(These equations are given)

So, the number of parameters would be:

- Weights per layer: 200 x 200

- FC connected layers: 2 x 4
- Biases: 4 x 200

- In total: 2 x 4 x 200 x 200 + 4 x 200 = 320, 800



4. The GRU equations are given. We would like to calculate the gradients of GRU for back propagation. For simplicity, we ignore the bias, and calculate the gradients of the second time stamp only.

   Using the chain rule, Calculate:

   a. $\frac{\partial\epsilon_{(2)}}{\partial W_{xz}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\frac{\partial h_{(2)}}{\partial z_{(2)}}\frac{\partial z_{(2)}}{\partial W_{xz}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\left(h_{(t-1)} - g_{(t)}\right)\frac{\partial}{\partial W_{xz}}[\sigma(W_{xz}{}^{T}x_{(t)} + W_{hz}{}^{T}h_{(t-1)} + b_z)] = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\left(h_{(t-1)} - g_{(t)}\right)\sigma_{z_{(2)}}\left[1-\sigma_{z_{(2)}}\right]x_{(2)}$

   b. $\frac{\partial\epsilon_{(2)}}{\partial W_{hz}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\frac{\partial h_{(2)}}{\partial z_{(2)}}\frac{\partial z_{(2)}}{\partial W_{hz}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\left(h_{(t-1)} - g_{(t)}\right)\frac{\partial}{\partial W_{hz}}[\sigma(W_{xz}{}^{T}x_{(t)} + W_{hz}{}^{T}h_{(t-1)} + b_z)] = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\left(h_{(t-1)} - g_{(t)}\right)\sigma_{z_{(2)}}\left[1-\sigma_{z_{(2)}}\right]h_{(1)}$

   c. $\frac{\partial\epsilon_{(2)}}{\partial W_{xg}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\frac{\partial h_{(2)}}{\partial g_{(2)}}\frac{\partial g_{(2)}}{\partial W_{xg}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}(1-z_{(2)})\frac{\partial}{\partial W_{xg}}\left[tanh\left(W_{xg}{}^{T}x_{(t)} + W_{hg}{}^{T}(r_{(t)}h_{(t-1)} + b_g\right)\right] = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}(1-z_{(2)})\left(1-tanh_{g_{(t)}}^{2}\right)x_{(2)}$

   d. $\frac{\partial\epsilon_{(2)}}{\partial W_{hg}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\frac{\partial h_{(2)}}{\partial g_{(2)}}\frac{\partial g_{(2)}}{\partial W_{hg}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}(1-z_{(2)})\frac{\partial}{\partial W_{hg}}\left[tanh\left(W_{xg}{}^{T}x_{(t)} + W_{hg}{}^{T}(r_{(t)}h_{(t-1)} + b_g\right)\right] = \frac{\partial\epsilon_{(2)}}{\partial h_{(t)}}(1-z_{(2)})\left(1-tanh_{g_{(t)}}^{2}\right)(r_{(2)} \otimes h_{(1)})$

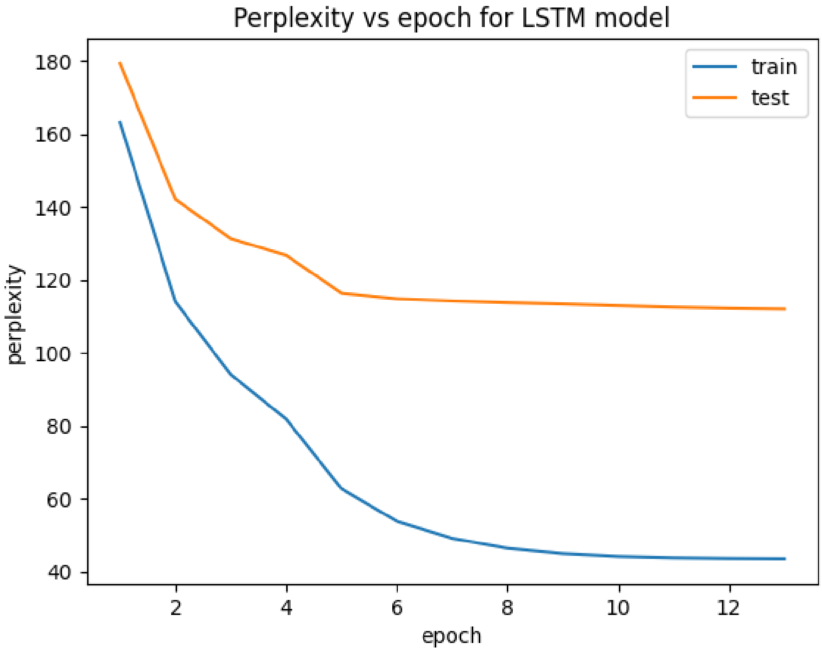
   e. $\frac{\partial\epsilon_{(2)}}{\partial W_{xr}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\frac{\partial h_{(2)}}{\partial g_{(2)}}\frac{\partial g_{(2)}}{\partial r_{(2)}}\frac{\partial r_{(2)}}{\partial W_{xr}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(t)}}(1-z_{(2)})\frac{\partial g_{(2)}}{\partial r_{(2)}}\frac{\partial r_{(2)}}{\partial W_{xr}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(t)}}(1-z_{(2)})\left((1 - tanh_{g_{(t)}}^{2})W_{hg}{}^{T}h_{(1)}\right)(\sigma_{r_{(2)}}(1-\sigma_{r_{(2)}})x_{(2)})$

   f. $\frac{\partial\epsilon_{(2)}}{\partial W_{hr}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(2)}}\frac{\partial h_{(2)}}{\partial g_{(2)}}\frac{\partial g_{(2)}}{\partial r_{(2)}}\frac{\partial r_{(2)}}{\partial W_{hr}} = \frac{\partial\epsilon_{(2)}}{\partial h_{(t)}}(1-z_{(2)})\left((1 - tanh_{g_{(t)}}^{2})W_{hg}{}^{T}h_{(1)}\right)(\sigma_{r_{(2)}}(1-\sigma_{r_{(2)}})h_{(1)})$

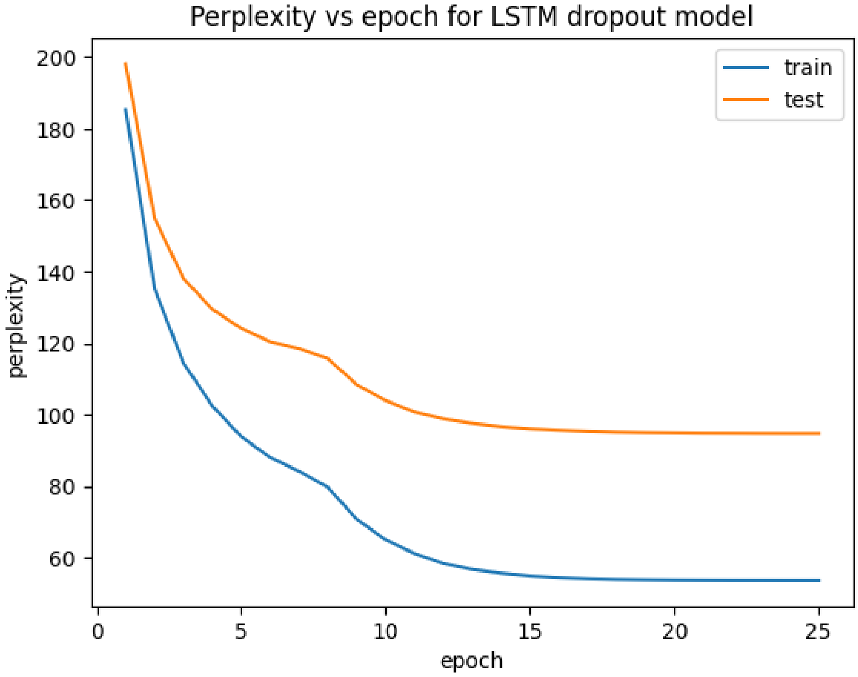
Practical part

1. Graphical results

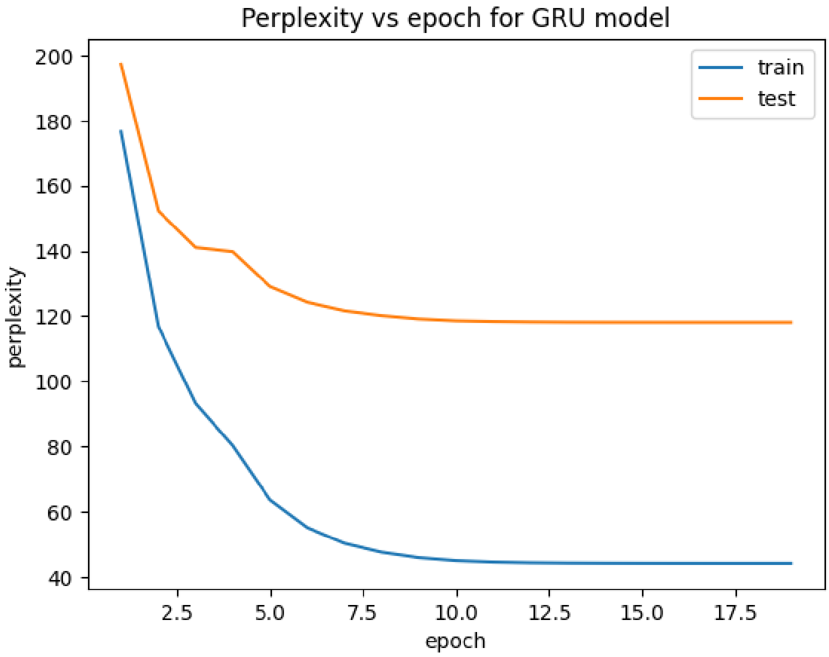
   The LSTM model without dropout was trained for a total of 13 epochs. The learning rate started at 1 and dropped by a factor of two after the 4th epoch.



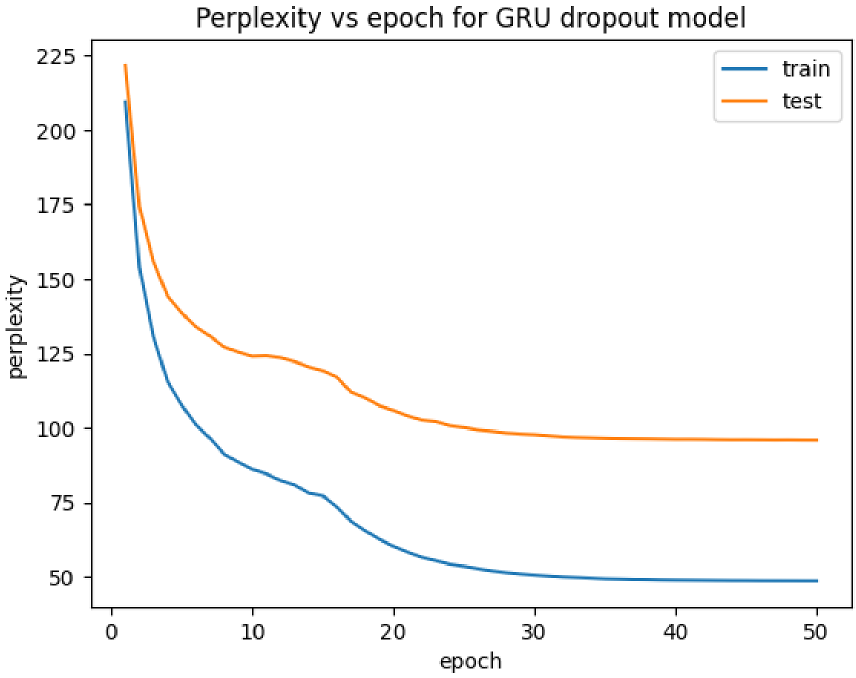
The LSTM model with dropout was trained with a 30% dropout rate for a total of 25 epochs. The learning rate started at 1 and dropped by a factor of 1.5 after the 8th epoch.



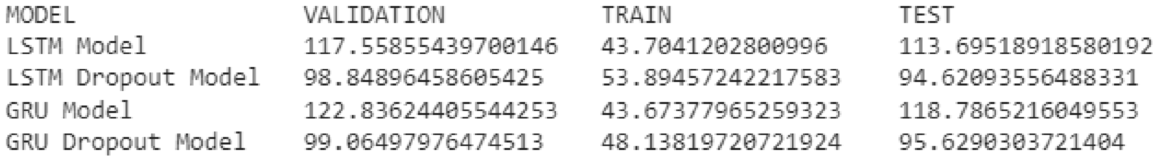
The GRU model without dropout was trained for a total of 20 epochs. The learning rate started at 0.4 and dropped by a factor of 1.9 after the 4th epoch.



The GRU model with dropout was trained with a 40% dropout rate for a total of 50 epochs. The learning rate started at 0.5 and dropped by a factor of 1.2 after 15 epochs.



2. Summarized perplexity table



```
MODEL                VALIDATION          TRAIN               TEST
LSTM Model           117.55855439700146  43.7041202800996    113.69518918580192
LSTM Dropout Model   98.84896458605425   53.89457242217583   94.62093556488331
GRU Model            122.83624405544253  43.67377965259323   118.7865216049553
GRU Dropout Model    99.06497976474513   48.13819720721924   95.6290303721404
```

3. Conclusions

For both the GRU and LSTM models, it can be clearly seen by looking at the graphs that correctly implementing dropout reduces overfitting on the test and validation data. The test data for models without dropout doesn't follow the trend of the train data- the perplexity of the test data stops dropping much earlier in training, pointing at overfitting of the model. For both the GRU and LSTM models, the validation perplexity was below 125 without dropout and below 100 with dropout. However, these results are not optimal- they were reached using trial and error and finding a good point at which to drop the learning rate for each model separately. These results could be further optimized by implementing an optimization method such as gradient or stochastic gradient descent.

Without dropout, we were able to reach a lower perplexity using the LSTM model than the GRU model- but this may be due to optimization of the parameters. With dropout we were able to reach similar perplexity scores, however the GRU model required many more epochs for training.