EX 2 – RNN

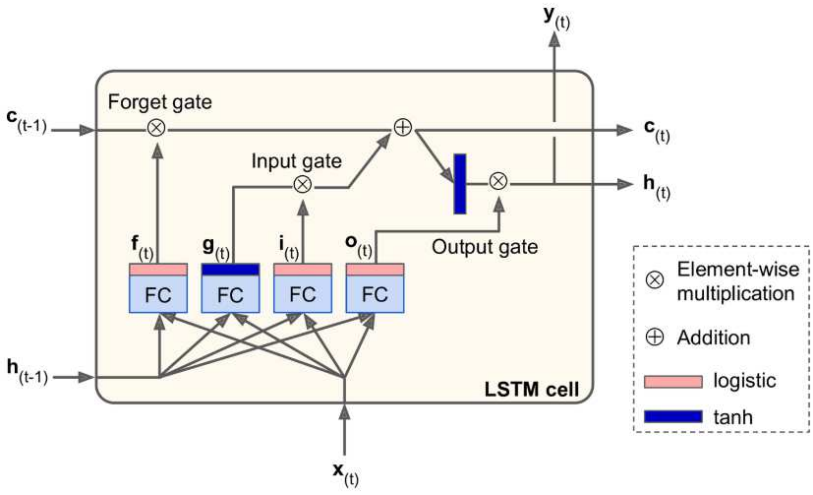
Please read the submission guidelines before you start.

Theory

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.
    b. Describe two ways (or more) to deal with variable-length **output** sequences.
2. Name two advantages of GRU over LSTM.
3. The LSTM cell equations are as follows:

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

An illustration of an LSTM cell:



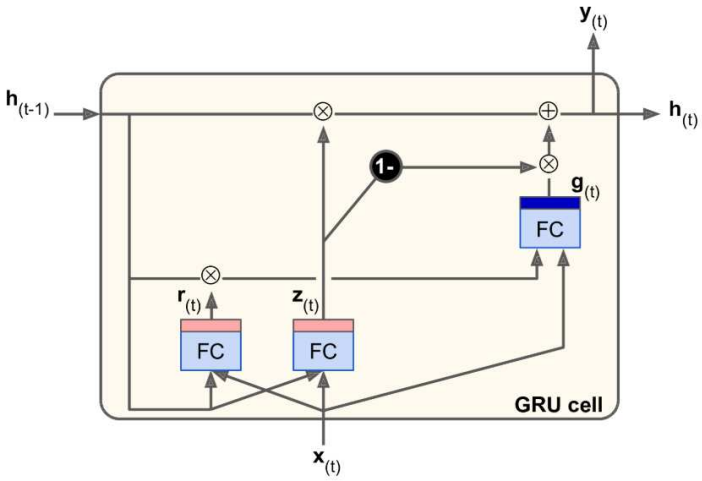
A network based on a single LSTM cell uses a vector of size 200 to describe the current state. Its input size are 200 sized vectors. Considering only parameters related to the cell, how many parameters does it have?

4. The GRU equations are as follows:

$$\mathbf{z}_{(t)} = \sigma(\mathbf{W}_{xz}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hz}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_z)$$
$$\mathbf{r}_{(t)} = \sigma(\mathbf{W}_{xr}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hr}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_r)$$
$$\mathbf{g}_{(t)} = \tanh(\mathbf{W}_{xg}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hg}{}^{T} \cdot (\mathbf{r}_{(t)} \otimes \mathbf{h}_{(t-1)}) + \mathbf{b}_g)$$
$$\mathbf{h}_{(t)} = \mathbf{z}_{(t)} \otimes \mathbf{h}_{(t-1)} + (1 - \mathbf{z}_{(t)}) \otimes \mathbf{g}_{(t)}$$

Where $\sigma(x) = \frac{1}{1+e^{-x}}$

An illustration of a GRU cell:



Consider GRU network with two timestamp (e.g. two iterations of the GRU cell), with a defined loss $\epsilon_{(t)}$ (e.g., the l_2 loss: $\frac{1}{2}(h_{(t)} - y_t)^2$).

Assume the gradient $\frac{\partial \epsilon_{(2)}}{\partial h_{(2)}}$ is given.

We would like to calculate the gradients of GRU for back propagation. For simplicity, you may ignore the bias, and calculate the gradients of the second time stamp only. Using the chain rule, Calculate:

a. $\frac{\partial \epsilon_{(2)}}{\partial W_{xz}}$

b. $\frac{\partial \epsilon_{(2)}}{\partial W_{hz}}$

c. $\frac{\partial \epsilon_{(2)}}{\partial W_{xg}}$

d. $\frac{\partial \epsilon_{(2)}}{\partial W_{hg}}$

e. $\frac{\partial \epsilon_{(2)}}{\partial W_{xr}}$

f. $\frac{\partial \epsilon_{(2)}}{\partial W_{hr}}$

Practical

We would like to experiment with the Penn Tree Bank data set. In this data set, we predict the next word. Instead of accuracy, we measure the performance by perplexity, which is quite similar to cross-entropy loss done in classification based NN.

To make sure you get the correct and updated version of the dataset, please download it from the course moodle.

For architecture, implement the "small" model as described in "Recurrent Neural Network Regularization", by Zaremba et al. It is the same as the "medium" and "large" models, only it has 200 hidden units (instead of 650 and 1500 accordingly).

a. For the four following settings, present a convergence graph (as described at ex1) for both the train and test Perplexity (8 in total), and for each one write the learning rate and dropout keep_prob.
   - LSTM based network without dropout.
   - LSTM based network with dropout.
   - GRU based network without dropout.
   - GRU based network with dropout.

b. Summarize the results by a table, with the resulted Perplexity on the train set, validation, and test.

c. Make conclusions regarding the results.

Comments:

- Describe in the readme file how to train each setting, and how to test it with the saved weights.
- All graphs should be clear with a proper heading. It is highly recommended (but not mandatory) to plot the train and test graphs for each technique together in the same plot (only 4 plots in total).
- You may need more than 13 epochs when using dropout. Change the learning rate appropriately in this case.
- No need to find the absolute best perplexity for each setting, but make sure that without dropout the **validation** perplexity is below 125, and with it is below 100.

Good Luck!