# Assignment 3 writeup

Thomas Lu

## 1 Problem 1

(a)

- (i)
  - Belgium beat Brazil 2-1 in today's World Cup match. (Does "Belgium" refer to the country, a location, or the Belgian national team, an organization? Similarly for Brazil.)
  - If you want the best diamonds, go to Tiffany. (Is "Tiffany" a person, or does it refer to the jewelry store, which is a location?)
- (ii) The word itself might appear in different sentences as different types of entities, and the only way to distinguish which type of entity it is representing is through examining the context.
- (iii)
  - If a noun does not have a dependent determinant (the/a/an/etc.), it is more likely to be a named entity.
  - A noun followed by an appositive phrase (a descriptive phrase contained between two commas) is more likely to be a named entity, especially if the appositive phrase doesn't contain any specifying pronouns (who, which, etc.).

(b)

- (i) $e^{(t)}$ will be $1 \times D(2w+1)$, $W$ will be $D(2w+1) \times H$, and $U$ will be $H \times C = H \times 5$.
- (ii) The matrix multiplications (which dominate the asymptotic computational complexity) required are:
  - $T$ multiplications $e^{(t)}W$ of a $1 \times D(2w+1)$ vector by a $D(2w+1) \times H$ matrix, which contributes $O(TD(2w+1)H) = O(TDwH)$ complexity.
  - $T$ multiplications $h^{(t)}U$ of a $1 \times H$ vector by a $H \times C$ matrix, which contributes $O(THC)$ complexity. Taking $C = 5$ to be constant, this is $O(TH)$.

  Thus the total complexity is $O(TH(Dw+1)) = O(THDw)$.

(d)

- (i) The best dev entity-level F1 scores achieved was 0.84. The token-level confusion matrix was:

| go\gu | PER | ORG | LOC | MISC | O |
|---|---|---|---|---|---|
| PER | 2947.00 | 33.00 | 97.00 | 10.00 | 62.00 |
| ORG | 129.00 | 1659.00 | 122.00 | 54.00 | 128.00 |
| LOC | 34.00 | 89.00 | 1895.00 | 14.00 | 62.00 |
| MISC | 41.00 | 59.00 | 55.00 | 1006.00 | 107.00 |
| O | 39.00 | 44.00 | 20.00 | 37.00 | 42619.00 |

From the confusion matrix, we can see that the model is commonly mispredicting instances of ORG as PER, LOC, or O, and commonly mispredicts instances of MISC as O. (This is corroborated by the token-level scores, which show low recall for the ORG and MISC categories.)

- (ii)
  - Because of the short window, the model has trouble detecting longer entity names, for example, "Duke of Norfolk's XI" in the below:

    ```
    x : May 15 v Duke of  Norfolk 's  XI  ( at Arundel )
    y*: O   O  O ORG  ORG ORG     ORG ORG O O  LOC     O
    y': O   O  O ORG  O   LOC     O   ORG O O  LOC     O
    ```

    This was particularly harmful to organizations as organizations often have longer names than other named entities.
  - If some prefix of an entity name can be the name of a different type of entity (e.g. "Hong Kong" is a LOC, but "Hong Kong Open" is a MISC), and only this prefix appears in the window, the model can make a mistake, as with the example below:

    ```
    x : SQUASH - HONG KONG OPEN QUARTER-FINAL RESULTS .
    y*: O      O MISC MISC MISC O             O       O
    y': O      O LOC  MISC MISC O             O       O
    ```

# 2 Problem 2

(a)

- (i) $b_1, b_2, U$ do not change, $W_e$ has $DH$ parameters instead of $(2w+1)DH$, we have a new parameter $W_h$ with $H^2$ parameters. Thus the RNN has $H^2 - 2wDH$ more parameters.
- (ii) At each time step $t$, the model must perform three matrix multiplications:
  - $h^{(t-1)}W_h$, which is a $1 \times H$ vector multiplied by a $H \times H$ matrix, for complexity $O(H^2)$
  - $e^{(t)}W_e$, which is a $1 \times D$ vector multiplied by a $D \times H$ matrix, for complexity $O(DH)$
  - $h^{(t)}U$, which is a $1 \times H$ vector multiplied by a $H \times C$ matrix, for complexity $O(HC)$

  Thus the total complexity is $O(T(H^2 + DH + HC) = O(TH(H + D + C))$.

(b)

- (i) Suppose the distribution is very skewed, with 99.9% of samples being negative and only 0.1% being positive. Suppose we were simply provided a constant hypothesis $p$; $p = 0.001$ (leading to a negative 0 for every sample) would minimize the cross-entropy, but would lead to a F1 score of 0, while $p = 0.5$ (and resolving this to a positive or negative prediction with equal probabiliity) would lead to a higher cross-entropy but a F1 score of 0.5.
- (ii) The F1 score is not differentiable; it shows a step change when any single prediction crosses the 0.5 threshold.

(d)

If we didn't use masking, our output would continue producing outputs $\hat{y}^{(t)} = \text{softmax}(\sigma(W_h h_{t-1} + W_e x_0 + b))$ for $T < t \leq M$ for each sentence in a batch with length $T < M$, where $M$ is the maximum length across sentences in the batch and $x_0$ is the embedding of the null token. We would require our model to learn that

when the null token is passed in, it should return the null output, regardless of state. Masking allows our model to ignore this restriction.

(g)

- (i)
  - The RNN model has limited ability to look into the future, since it only uses a window size of 1. Thus (similarly to the window-based model) it can get tripped up on entities who have prefixes that are a different type of entity, such as "Hong Kong Open" in the below example:

    ```
    x : SQUASH - HONG KONG OPEN QUARTER-FINAL RESULTS .
    y*: O      O MISC MISC MISC O             O       O
    y': O      O LOC  MISC MISC O             O       O
    ```

  - The model also has trouble with named entities that come at the beginning of sentences. Oftentimes locations and organizations are named after people, so there isn't really any helpful information in the words themselves, and the model has no other information to work on. Example:

    ```
    x : Apic Yamada - 6mth parent forecast .
    y*: ORG  ORG    O O    O      O        O
    y': PER  PER    O O    O      O        O
    ```

- (ii) Both of these issues can be addressed by allowing the model to see farther into the future. There are a number of ways this can be done. Two examples are:
  - Increase the window size.
  - Make the RNN a bidirectional RNN.

# 3 Problem 3

(a)

- (i) We have $h^{(t)} = \sigma(x^{(t)}w_h + h^{(t-1)}u_h + b_h)$. The values $w_h = 1, u_h = 1, b_h = -0.5$ will satisfy the requirements of the problem: if either $x^{(t)}$ or $h^{(t-1)}$ is 1, then $h^{(t)}$ will be 1, and if both are 0, then $h^{(t)}$ will be 0.
- (ii) $w_r = u_r = b_r = 0$ yields $r^{(t)} = \sigma(x^{(t)}w_r + h^{(t-1)}u_r + b_r) = \sigma(0) = 0$, so we have the following equations (recall $b_z = b_h = 0$):

$$
\begin{aligned}
z^{(t)} &= \sigma(x^{(t)}w_z + h^{(t-1)}u_z) \\
\tilde{h}^{(t)} &= \sigma(x^{(t)}w_h) \\
h^{(t)} &= z^{(t)}h^{(t-1)} + (1 - z^{(t)})\tilde{h}^{(t)}.
\end{aligned}
$$

  It is not difficult to verify that setting $w_z = 0, u_z = 1, w_h = 1$ gives the desired behavior. The value of $u_h$ is inconsequential, as it is zeroed out by $r^{(t)}$ always evaluating to 0. (Actually, $u_z$ and $w_h$ can be any positive value.)

(b)

- (i)

We have $h^{(t)} = \sigma(w_x x^{(t)} + w_h h^{(t-1)} + b_h)$. We need $h^{(t)} = 0$ when $x^{(t)}$ and $h^{(t-1)}$ are both 0 or both 1, and $h^{(t)} = 1$ when one of $x^{(t)}$ and $h^{(t-1)}$ 0 and the other is 1. $h^{(t)} = 0$ is equivalent to $w_x x^{(t)} + w_h h^{(t-1)} + b_h \leq 0$, so the previous four conditions become:

$$\begin{aligned} b_h &\leq 0 \\ w_x + w_h + b_h &\leq 0 \\ w_h + b_h &> 0 \\ w_x + b_h &> 0 \end{aligned}$$

From the first, third, and fourth conditions, $b_h$ must be nonpositive $w_h$ and $w_x$ must be positive. However, this makes it impossible to simultaneously satisfy the second and third conditions, so the 1D RNN cannot model toggling behavior.

- (ii)

  With our simplifications, the GRU conditions become:

  $$\begin{aligned} z^{(t)} &= \sigma(w^{(t)} w_x + h^{(t-1)} u_z) \\ r^{(t)} &= \sigma(b_r) \\ \tilde{h}^{(t)} &= \tanh(x^{(t)} w_h + r^{(t)} h^{(t-1)} u_h) \\ h^{(t)} &= z^{(t)} h^{(t-1)} + (1 - z^{(t)}) \tilde{h}^{(t)} \end{aligned}$$

  We now need to satisfy four conditions:

  - When $x^{(t)} = h^{(t-1)} = 0$ we need $h^{(t)} = 0$. The precondition yields the simplifications $z^{(t)} = 0$ and $\tilde{h}^{(t)} = 0$, so for this condition, the values of $w_z, u_z, w_h, u_h, b_r$ are inconsequential.
  - When $x^{(t)} = 1, h^{(t-1)} = 0$ we need $h^{(t)} = 1$. The precondition yields the simplifications $z^{(t)} = \sigma(w_z)$, $\tilde{h}^{(t)} = \tanh(w_h)$, $h^{(t)} = (1 - z^{(t)})\tilde{h}^{(t)}$. We see that we need $z^{(t)} = 0$ and $\tilde{h}^{(t)} = 1$, so we must have $w_h > 0$, $w_z \leq 0$.
  - When $x^{(t)} = h^{(t-1)} = 1$ we need $h^{(t)} = 0$. The precondition yields the simplifications $z^{(t)} = \sigma(w_z + u_z)$, $\tilde{h}^{(t)} = \tanh(w_h + r^{(t)} u_h)$, and $h^{(t)} = z^{(t)} + (1 - z^{(t)})\tilde{h}^{(t)}$. We see that we must have $z^{(t)} = 0$, implying $u_z + w_z \leq 0 \Rightarrow u_z \leq -w_z$ and $\tilde{h}^{(t)} = 0$, which in turn implies $w_h + r^{(t)} u_h = w_h + \sigma(b_r) u_h \leq 0 \Rightarrow b_r > 0, u_h \leq -w_h$.
  - When $x^{(t)} = 0, h^{(t-1)} = 1$ we need $h^{(t)} = 1$. The precondition (taken with $b_r > 0$, $u_h \leq -w_h < 0$) yields the simplifications $z^{(t)} = \sigma(u_z)$, $\tilde{h}^{(t)} = \tanh(u_h) = 0$, and $h^{(t)} = z^{(t)}$. This yields $u_z > 0$.

  Our final set of conditions on $w_z, u_z, w_h, u_h, b_r$ are thus:

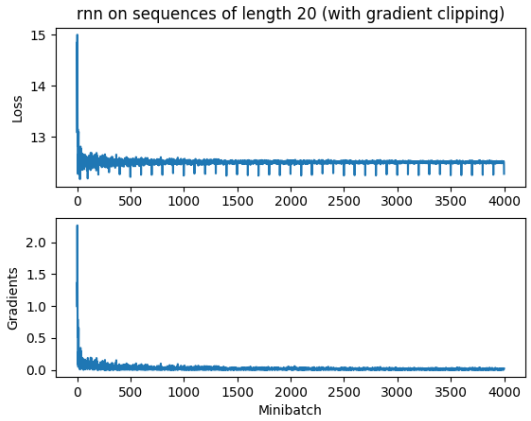
  $$\begin{aligned} b_r &> 0 \\ w_h &> 0 \\ w_z &\leq 0 \\ 0 < u_z &\leq -w_z \\ u_h &\leq -w_h \end{aligned}$$

  It can be mechanically verified that any values $b_r, w_h, w_z, u_z, u_h$ satisfying these conditions will allow the GRU to model toggling behavior.

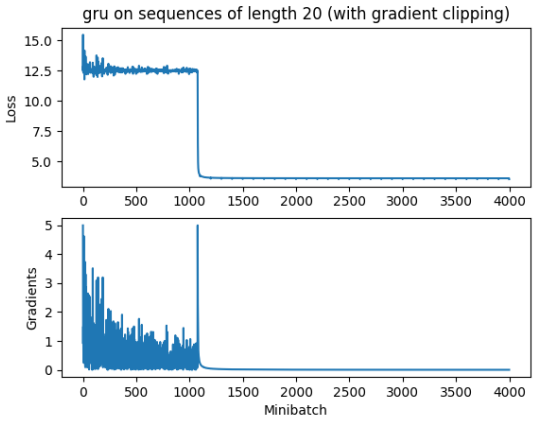
(d)

The results of the experiment are shown in Figure 1.

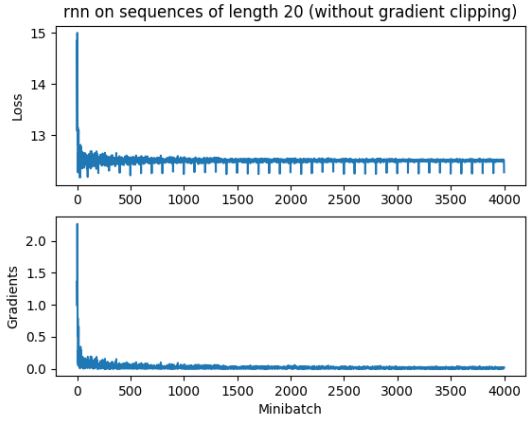
(e)



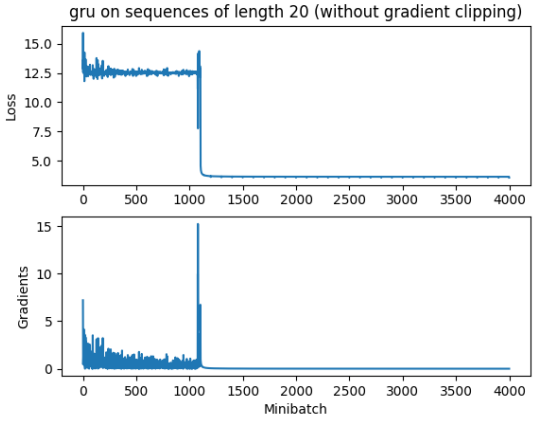

(a)




(b)




(c)




(d)

Figure 1: Gradients and losses from the RNN/GRU experiments with and without gradient clipping.

- (i) It seems like all models experienced vanishing gradients. The non-clipped GRU model had a few somewhat large gradients (maximum norm around 15), but nothing that would warrant a description of "exploding." Since there were no exploding gradients, gradient clipping was not very helpful.
- (ii) Clipped models and unclipped models performed similarly. The GRUs performed better than the RNNs, probably because they were better able to maintain memory throughout the sequence - you can see that the RNNs quickly entered a state where all the gradients vanished, while the GRUs kept significant gradients through many more training examples.