Assignment #3

ZIAUL HASAN HASHMI

① A window into NER

(a) (i) " Pearson rejected my manuscript but sent me an encouraging response", Here " Pearson " could be a person (PER) or a publishing group (ORG).

" Washington, Brace Yourself for extreme cold." Here it is ambiguous whether "Washington" is a person (PER) or a city (LOC).

(ii) Using additional features such as case information and parts of speech tags can help generalize the system and can classify uncommon words that would normally be not named entities. As an example: "apple" is simply a noun whereas "Apple" is an (Capital A) ORG

(iii) part of speech tags, prefixes and suffixes and case information of the current and the surroundings.

(b) (i) dimension of $e^{(t)}$ = $1 \times (2W+1)D$

dimension of W = $(2W+1)D \times H$

dimension of U = $H \times C$

Using the dimensions that satisfy the given equations

(ii) (Based on Instructor's note on piazza that it takes $O(D)$ time to grab a D-dimensional embedding from E) @1047

For a single window:

Complexity to compute $e^{(t)} = O((2W+1)D)$

Complexity to compute $h^{(t)} = O((2W+1)DH)$ (ignore bias terms)

Complexity to compute $\hat{y}^{(t)} = O(HC)$ [ignoring bias additions]

So overall complexity for a single window $= O((2W+1)DH + HC) = O((2W+1)DH)$ Assuming $D \gg C$.

For a sentence of length T

Complexity would hence be $\boxed{O((2W+1)DHT)}$

(c) (i) Please check code for implementation

(ii) " "

(iii) " "

(iv) " "

Please check "window_predictions.conll"

(d)(i) Best entity level F1 score is 83% and the corresponding confusion matrix is as follows

go\gu	PER	ORG	LOC	MISC	O
PER	2953.00	63.0	59.00	17.00	57.00
ORG	145.00	1683.00	98.00	50.00	116.00
LOC	54.00	117.00	1861.00	18.00	44.00
MISC	47.00	77.00	43.00	993.00	108.00
O	48.00	60.00	15.00	33.00	42603.00

The confusion matrix shows that the biggest source of error (145) is due to organizations being confused as person, followed by locations being confused as organizations. Also non-entities are rarely classified as entities.

(ii) Window based model cannot make use of decisions made in neighboring windows. For eg. in this sentence:

Next week Kansai Electric Power and .....

y^*	O	O	ORG	ORG	ORG	O	...
$\hat{y}$ (pred)	O	O	PER	ORG	ORG		

Here "Kansai" be easily inferred as "ORG" based on subsequent decisions on "Electric Power" as ORG | ORG but the window based model cannot use this information from neighboring windows

Window based model also cannot utilize information from distant parts of the sentence. For eg:

"I told Monica ... win," King said

y^*	O	O	PER	...	O	PER	O	
$\hat{y}$ (pred)	O	O	PER	...	O	O	O	

Here "I told" is referring to King but window based model fails to get that.

② Recurrent Neural Nets for NER

(a)(i) RNN has DXH parameters for W_e
and HXH parameters for W_h
and HXC parameters for U
Total = $DH + H^2 + HC$

In comparison window based model has
$(2W+1)DH + HC$ model parameters

So RNN has only DH parameters instead of $(2W+1)DH$ parameters for weight matrix and has a H^2 more parameters for W_h.

(ii) For one time step:

complexity for calculating $e^{(t)} = O(D)$
complexity for calculating $h^{(t)} = O(H^2 + DH + H)$
complexity for calculating $\hat{y}^{(t)} = O(HC + C)$

So overall complexity for one time step $= O(H^2 + DH + HC)$
$= O(H^2 + DH)$ if $D >> C$ (only higher order terms)

So for a sentence of length T, we need to do this for T time steps. So overall complexity is $\boxed{O(TH^2 + DHT)}$ for the full sentence

(b)(i) Let us take the following example sentence

"The American Airlines"

Gold standard	O	ORG	ORG
Prediction 1	O	O	O
Prediction 2	O	O	ORG

Suppose we were making prediction 1 and we then switched to prediction 2. In this case, cross entropy will decrease for sure because we have one more correct token label. Lets look at F1 score:

Entity level Precision: Goes down because we make error over the entity and it is counted towards precision

Entity level Recall: Stays the same because we still make the same error over the whole entity

So, F1 score goes down

(ii) Since F1 depends on the whole dataset, we cannot stochastic/mini-batch gradient descent like algorithms so it would be really slow. Also we need raw counts of true positives, false positives & false negatives so the objective will be a combinatorial optimization problem.

(c) Please check code for implementation

(d) (i) If we do not use masking then gradients from padding units would flow down the network affecting the RNN parameters and hence their learning.

The loss would include the terms due to the inaccurate predictions of 0-vector y for padding.

Masking helps to remove such loss terms by multiplying them with 0 and hence also stops the gradient flow.

(ii) Please check code for implementation

(e) Please check code.

(f) Please check "rnn_predictions.conll"

(g)

(i) & (ii)

This model cannot fix or update decisions once it makes correct decisions in future. Consider the following example:

Kansai Electric Power Corporation

Model Output PER PER ORG ORG

Future decisions can be used by training the model both on forward & reversed sentences. A bidirectional RNN could be used.

The model cannot make of collective words to make decisions (which include in / of etc), Eg.

Department of Defense

Model Output ORG O ORG

This can probably by considering pairwise smoothening of words such as 'in' or "of" that are sandwiched between two named entities.

③ Grooving with GRUs

(a) For this part lets assume $\sigma(x) \cong \tanh(x) \cong \varphi(x)$

where $\varphi(x) \rightarrow \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$

(i) For RNN

$$h^{(t)} = \varphi\left(x^{(t)} w_h + h^{(t-1)} u_h + b_h\right)$$

There could be four cases :

* $h^{(t-1)} = 0$, $x^{(t)} = 0$ then $h^{(t)} = 0$

$$\Rightarrow \quad \varphi(b_h) = 0$$

$$\Rightarrow \quad b_h \leq 0 \quad \text{——} \quad (1)$$

* $h^{(t-1)} = 0$, $x^{(t)} = 1$, then $h^{(t)} = 1$

$$\Rightarrow \quad \varphi(w_h + b_h) = 1$$

$$\Rightarrow \quad w_h + b_h > 0 \quad \text{——} \quad (2)$$

* $h^{(t-1)} = 1$, $x^{(t)} = 0$, then $h^{(t)} = 1$

$$\Rightarrow \quad \varphi(u_h + b_h) = 1$$

$$\Rightarrow \quad u_h + b_h > 0 \quad \text{——} \quad (3)$$

* $h^{(t-1)} = 1$, $x^{(t)} = 1$ then $h^{(t)} = 1$

$$\Rightarrow \quad \varphi(w_h + u_h + b_h) = 1$$

$$\text{or} \quad w_h + u_h + b_h > 0 \quad \text{——} \quad (4)$$

but (4) can be derived by combining (1), (2), (3)

(1) & (3) $\Rightarrow u_h > 0$ adding this to (2) gives (4)

Combining ①, ②, ③ & ④ We can say that the values that can help replicate this automation are given by

$$b_h \leq 0$$
$$w_h + b_h > 0$$
$$u_h + b_h > 0$$

④ Can be derived from ①, ②, & ③.

(ii) $[\sigma \cong \tanh \cong \psi]$ Since $W_r = U_r = b_r = b_z = b_h = 0$

$$\Rightarrow \quad r^{(t)} = \psi(0) = 0$$
$$z^{(t)} = \psi\left(x^{(t)} W_z + h^{(t-1)} U_z\right)$$
$$\tilde{h}^{(t)} = \psi\left(x^{(t)} W_h\right)$$
$$h^{(t)} = z^{(t)} h^{(t-1)} + \left(1 - z^{(t)}\right) \tilde{h}^{(t)}$$

Four Cases:

* $h^{(t-1)} = 0$, $x^{(t)} = 0$ then $h^{(t)} = 0$

$$\Rightarrow \quad z^{(t)} = \psi(0) = 0$$
$$\tilde{h}^{(t)} = \psi(0) = 0$$

So $h^{(t)} = 0$ [automatically 0, satisfies all the time]

* $h^{(t-1)} = 0$, $x^{(t)} = 1$, then $h^{(t)} = 1$

$$\Rightarrow \quad z^{(t)} = \psi(W_z)$$
$$\& \quad \tilde{h}^{(t)} = \psi(W_h)$$
$$\text{and} \quad h^{(t)} = \left(1 - \psi(W_z)\right) \psi(W_h)$$

Since $h^{(t)} = 1$

So both $1 - \psi(w_z) = 1$ or $\psi(w_z) = 0$

and $\psi(w_h) = 1$

$\Rightarrow$ $w_z \leq 0$ & $w_h > 0$ — (1)

★ $h^{(t-1)} = 1$, $x^{(t)} = 0$ then $h^{(t)} = 1$

$\Rightarrow$ $z^{(t)} = \psi(u_z)$

$\tilde{h}^{(t)} = \psi(0) = 0$

$h^{(t)} = z^{(t)}$

Since $h^{(t)} = 1$

$\Rightarrow$ $\psi(u_z) = 1$

or $u_z > 0$ — (2)

★ $h^{(t-1)} = 1$, $x^{(t)} = 1$ then $h^{(t)} = 1$

$z^{(t)} = \psi(w_z + u_z)$

$\tilde{h}^{(t)} = \psi(w_h) = 1$ from (1)

$h^{(t)} = z^{(t)} + (1 - z^{(t)})$

$= 1$ always so it satisfies

Therefore required conditions are

$w_z \leq 0$

$w_h > 0$ & u_h can be any real number

$u_z \geq 0$

(b) $\sigma \cong \tanh \cong \varphi$

(i) If we input $x^{(t)} = 0$, state remains the same

ie $h^{(t)} = 0$ if $h^{(t-1)} = 0$

$h^{(t)} = 1$ if $h^{(t-1)} = 1$

$\Rightarrow \quad \varphi(u_h + b_h) = 1$

& $\varphi(b_h) = 0$

$\Rightarrow \quad b_h \leq 0$

& $u_h + b_h > 0$

or $u_h > 0$ ——— (1)

If we input $x^{(t)} = 1$, state toggles

ie $h^{(t)} = 0$ if $h^{(t-1)} = 1$

$h^{(t)} = 1$ if $h^{(t-1)} = 0$

$\Rightarrow \quad \varphi(w_h + b_h) = 1$

& $\varphi(w_h + u_h + b_h) = 0$

or $w_h + b_h > 0$

& $w_h + u_h + b_h \leq 0$

$\Rightarrow \quad u_h < 0$ ——— (2)

but (1) & (2) are in contradiction

So 1 D RNN cannot replicate this behavior.

(ii) $\sigma \equiv \tanh \equiv \psi$

$W_r = U_r = b_z = b_h = 0$

Then

$$z^{(t)} = \psi\left(x^{(t)} W_z + h^{(t-1)} U_z\right)$$

$$r^{(t)} = \psi(b_r)$$

$$\tilde{h}^{(t)} = \psi\left(x^{(t)} W_h + r^{(t)} h^{(t-1)} U_h\right)$$

$$h^{(t)} = z^{(t)} h^{(t-1)} + (1 - z^{(t)})\, \tilde{h}^{(t)}$$

Again taking four possible cases

* $h^{(t-1)} = 0$ $x^{(t)} = 0$ then $h^{(t)} = 0$

We have, $z^{(t)} = \psi(0) = 0$

$\tilde{h}^{(t)} = \psi(0) = 0$

So $h^{(t)} = 0$ (always satisfied)

* $h^{(t-1)} = 0$, $x^{(t)} = 1$, then $h^{(t)} = 1$

We have, $z^{(t)} = \psi(W_z)$

$r^{(t)} = \psi(b_r)$

$\tilde{h}^{(t)} = \psi(W_h)$

$$1 = h^{(t)} = (1 - \psi(W_z))\, \psi(W_h)$$

This is true iff $\psi(W_z) = 0$ & $\psi(W_h) = 1$

or $W_z \leq 0$ & $W_h > 0$ — (1)

* $h^{(t-1)} = 1$, $x^{(t)} = 0$ then $h^{(t)} = 1$

We have, $z^{(t)} = \psi(U_z)$

$r^{(t)} = \psi(b_r)$

$\tilde{h}^{(t)} = \psi(r^{(t)} U_h)$

So $h^{(t)} = z^{(t)} + (1 - z^{(t)}) \psi(r^{(t)} U_h)$

$h^{(t)}$ can be 1

iff $z^{(t)} = 1$ — (i)

or $\psi(r^{(t)} U_h) = 1$ & $z^{(t)} = 0$ — (ii)

(i) means $U_z > 0$

OR

(ii) means $U_z \leq 0$ and $b_r > 0$ and $U_h > 0$ ← (2)

* $h^{(t-1)} = 1$, $x^{(t)} = 1$ then $h^{(t)} = 0$

$z^{(t)} = \psi(W_z + U_z)$

$\tilde{h}^{(t)} = \psi(W_h + r^{(t)} U_h)$

$0 = h^{(t)} = z^{(t)} + (1 - z^{(t)}) \tilde{h}^{(t)}$

$z^{(t)} \to 0$, $\tilde{h}^{(t)} \to 0$ → Can only happen

So both $z^{(t)}$ & $\tilde{h}^{(t)}$ must be 0

i.e $W_z + U_z \leq 0$

and $W_h + r^{(t)} U_h \leq 0$ → (3)

From ① $W_h > 0$

So from ④ $W_h + r^{(+)} U_h \leq 0$

$\Rightarrow r^{(+)} U_h < 0$

This can only be satisfied when $r^{(+)} = 1$

& $U_h < 0$

i.e $b_r > 0$ & $U_h < 0$ — ④

So ② (ii) Cannot be satisfied.

Gathering ①, ②, ③, ④

$W_h > 0$

$U_Z > 0$

$W_Z + U_Z \leq 0$

$b_r > 0$

$U_h < 0$

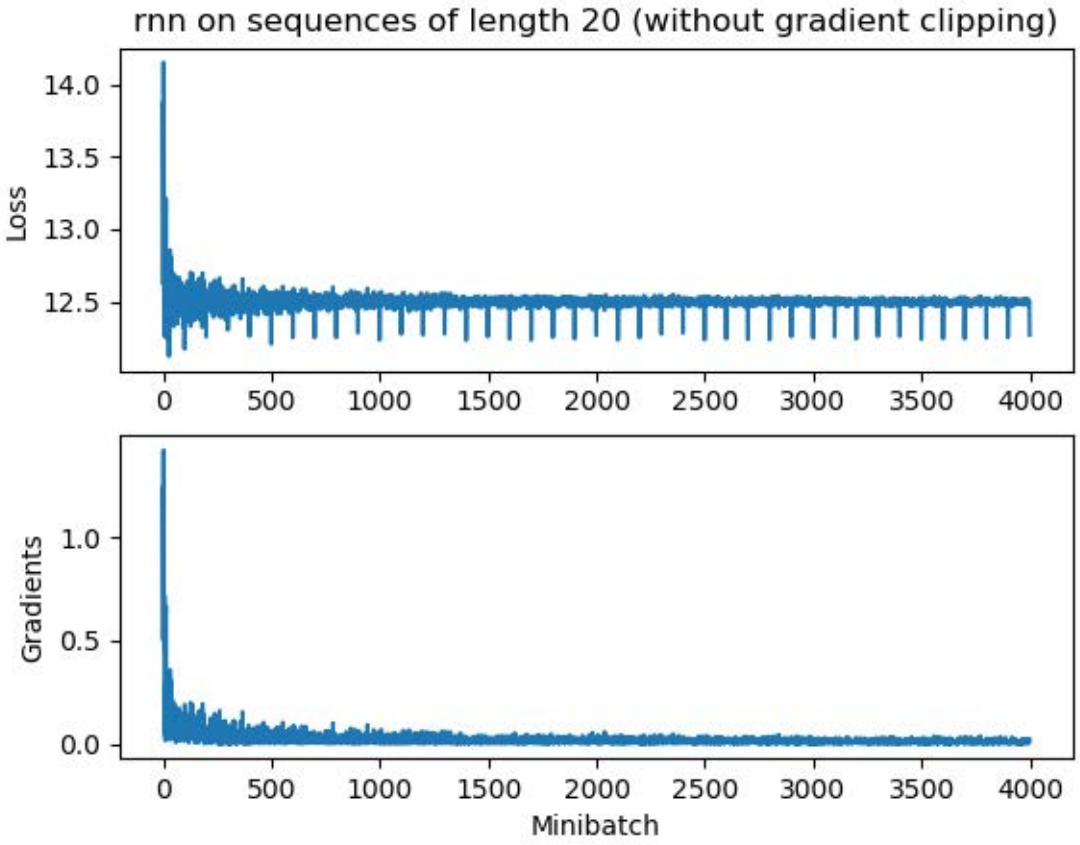
— Final Conditions

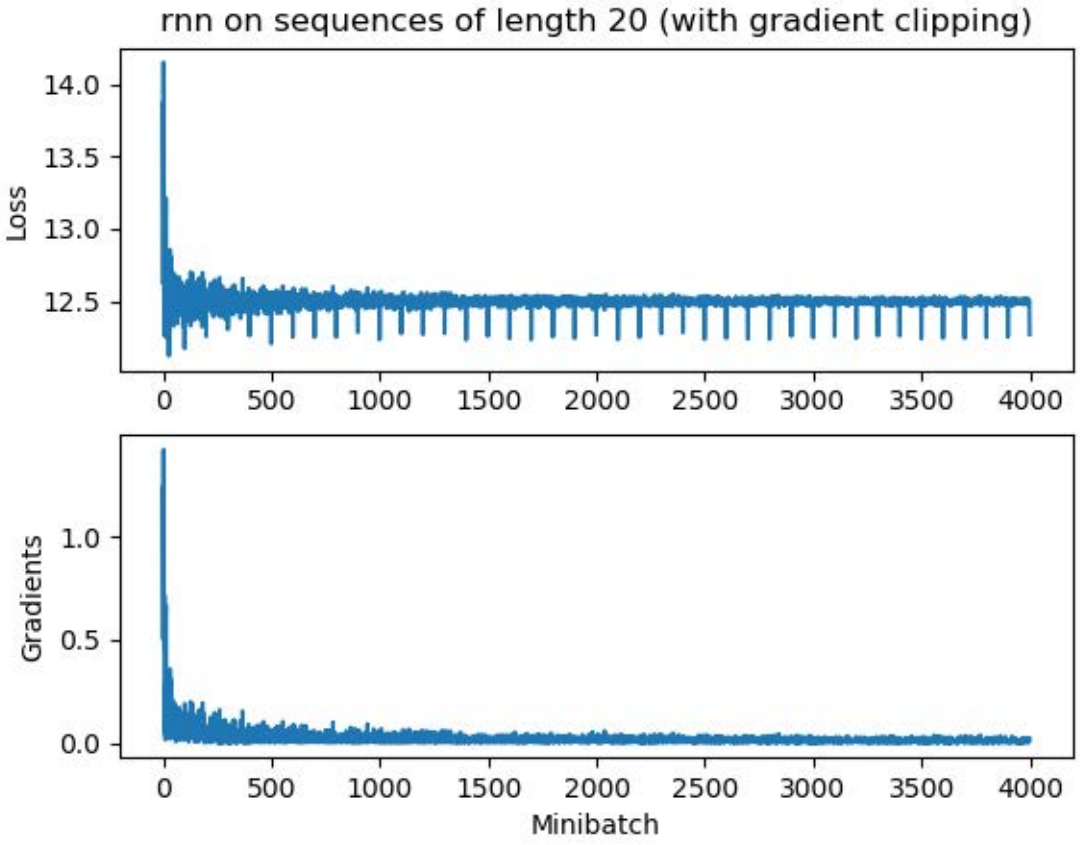
This was also verified by the following sample values for model parameters:

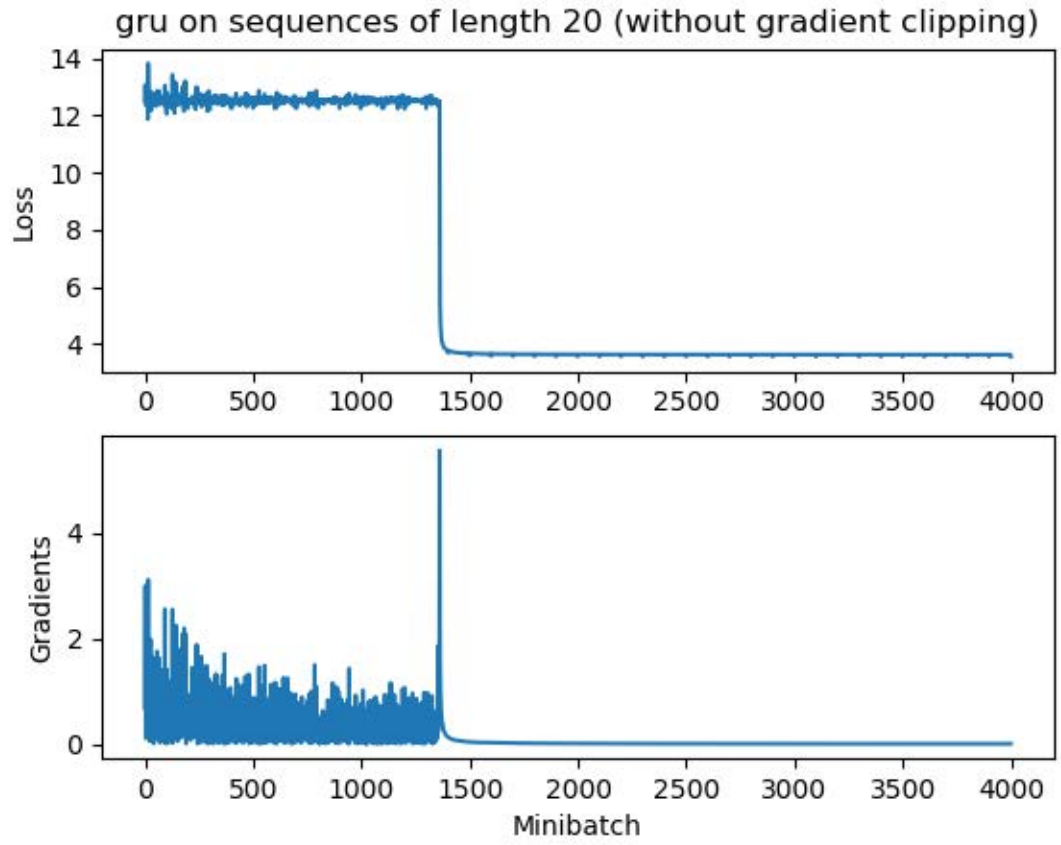
$W_h = 1$ $U_Z = 1$ $W_Z = -2$, $b_r = 1$ & $U_h = -1$

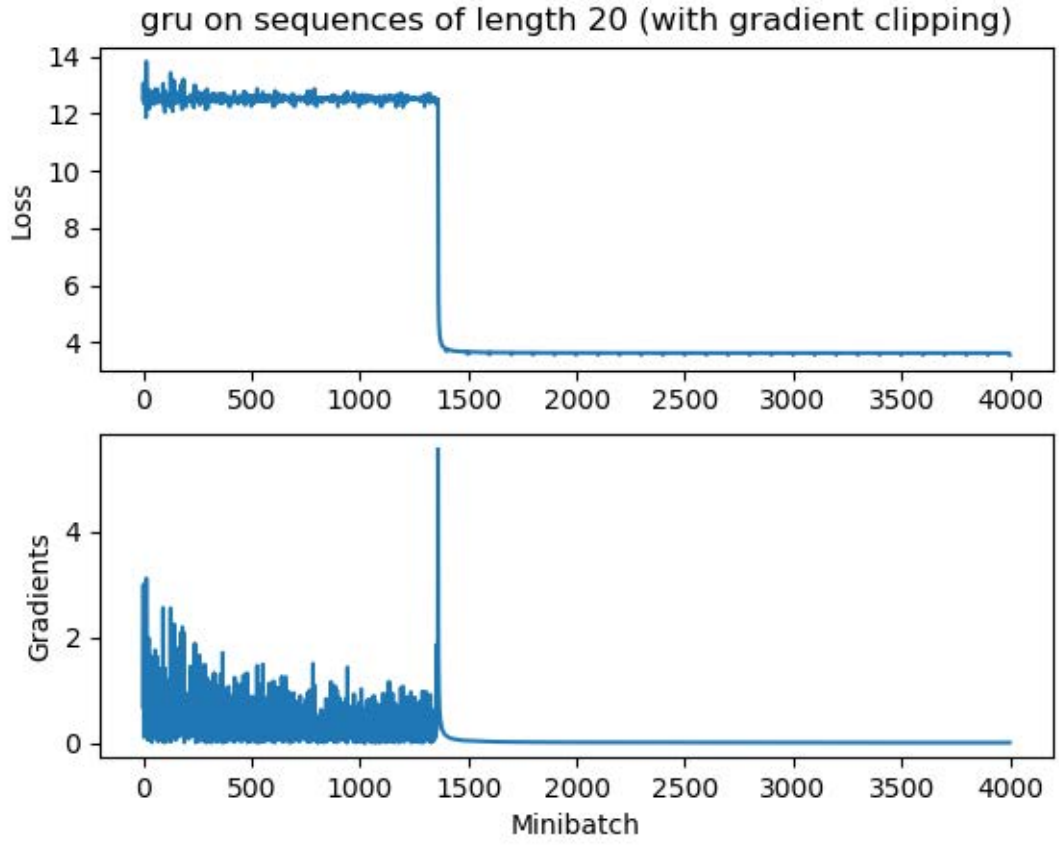
(c) Please check code

(d) Please check code & plots


rnn on sequences of length 20 (without gradient clipping)
Loss
14.0
13.5
13.0
12.5
0
500
1000
1500
2000
2500
3000
3500
4000
Gradients
1.0
0.5
0.0
0
500
1000
1500
2000
2500
3000
3500
4000
Minibatch


rnn on sequences of length 20 (with gradient clipping)
Loss
14.0
13.5
13.0
12.5
0
500
1000
1500
2000
2500
3000
3500
4000
Gradients
1.0
0.5
0.0
0
500
1000
1500
2000
2500
3000
3500
4000
Minibatch


gru on sequences of length 20 (without gradient clipping)
Loss
14
12
10
8
6
4
0
500
1000
1500
2000
2500
3000
3500
4000
Gradients
4
2
0
Minibatch


gru on sequences of length 20 (with gradient clipping)
Loss
14
12
10
8
6
4
0
500
1000
1500
2000
2500
3000
3500
4000
Gradients
4
2
0
Minibatch

(e) (i) RNN model seems to have vanishing gradient problem because the loss doesn't go to zero but gradients become zero so the model stops improving.

GRU model seems to have a exploding gradient problem at around 1500th minibatch but it is less than max_norm set in config file so gradient clipping doesn't do anything. GRU has this problem due to abrupt change in the dynamics of the GRU due to small variation in the parameters or accumulated gradients.

Gradient clipping doesn't kick in for GRUs because gradients didn't exceed the max_norm (=5) parameter.

For RNNs, gradients quickly vanish so gradient clipping is not used.

(ii) GRU does better overall than RNN since loss goes further down.

This is because for GRUs, gradients corresponding to historical inputs & states are carried forward and maintained so it can further improve the loss. In summary, GRUs have better control over the gradient flow and they enable better preservation of "long-range dependencies".

(f) Please check "gru-predictions.conll"