CS224N Assignment 3

Kangwei Ling

March 4, 2017

1 A window into NER

(a) i. • **The Smith's** makes top-notch bread.

ii. The context matters, words may have different meaning in different context.

iii. Context words, POS tags

(b) i.

$$\begin{aligned} \boldsymbol{e}^{(t)} &: (2w+1)D \\ \boldsymbol{W} &: (2w+1)DH \\ \boldsymbol{U} &: HC \end{aligned}$$

ii.

$$T \cdot [(2w+1)VD + (2w+1)DH + H + HC + C]$$

(c) `q1_window.py`

(d) i. Best F_1 score:

	P	R	F_1
Entity-level	0.81	0.84	0.83

Confusion Matrix:

Table 1: My caption

go\gu	PER	ORG	LOC	MISC	O
PER	2959.00	39.00	72.00	14.00	65.00
ORG	158.00	1624.00	110.00	65.00	135.00
LOC	46.00	112.00	1862.00	26.00	48.00
MISC	43.00	60.00	44.00	1012.00	109.00
O	45.00	41.00	16.00	26.00	42631.00

With an inspection of the confusion matrix, I found that the model make mistakes mostly by recognizing ORG as PER, LOC as ORG, ORG as LOC, ORG as O, MISC as O.

ii. All in all, the training data is skewed, as most of words are O (non-entity). There should be no doubt that the model works well on capturing non-entity words. For the named entities, window size throttles the performance of this model, as some organizations have longer names, which have some location words(e.g. city names) in them, confusing the model.

- misclassify ORG for LOC.

```
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
```

- missclassify LOC for PER, as some places are named after famous people.

```
x : Washington
y*: PER
y': LOC
```

2 Recurrent neural nets for NER

(a) i. $H^2 - 2wDH$ more.

ii. $T(VD + H^2 + DH + 2H + HC + C)$

(b) i. For the example below, cross-entropy loss is decreased but F_1 score is also decreased.

```
correct: ORG ORG ... ORG ORG ... PER PER
before:  ORG ORG ... O   O   ... O   O
after:   ORG O   ... ORG O   ... ORG O
```

ii. F_1 score has to be evaluated on large dataset, so minibatch can't completely.

iii. Not easy to compute gradients. F_1 score function may not be easily optimized.

(c) `q2_rnn_cell.py`

(d) i. Without masking, loss and gradient updates will be evaluated on many non-existed data (padding). Masking make sure that the loss generated at padding tokens is zero and no loss is back propagated to timestamps before.

ii. `q2_rnn.py`

(e) `q2_rnn.py`

(f) `q2_rnn.py`

(g) i.

ii.

3 Grooving with GRUs

(a) i. A simple setup: $w_h = 1, u_h = 1, b_h = 0$.

ii. Set all to 1.

(b) i. For toggling, it must hold that:

$$\begin{cases} u_h + w_h + b_h & \leq 0 \\ u_h + b_h & > 0 \end{cases}$$

which implies that $w_h < 0$. For staying when 0 arrives, it must hold that:

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

which implies $w_h > 0$, leads to contradiction.Therefore, a 1D RNN can not replicate the toggling behavior.

ii.

$$\begin{cases} u_z & = -1 \\ w_z & = 1 \\ b_r & = 1 \\ u_h & = 1 \\ w_h & = -1 \end{cases}$$

(c) `q3_gru_cell.py`

(d) `q3_gru.py`

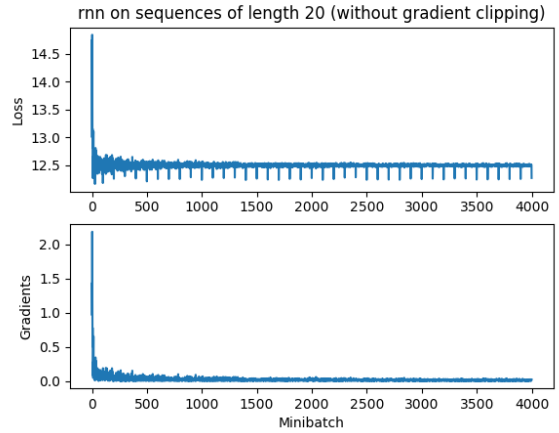

Figure 1: rnn no clip

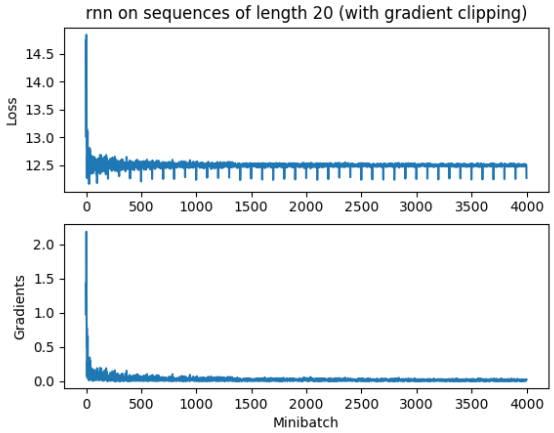

Figure 2: rnn with clip

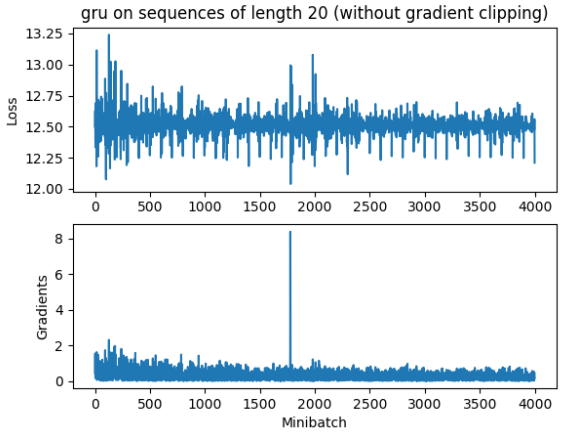

Figure 3: gru no clip

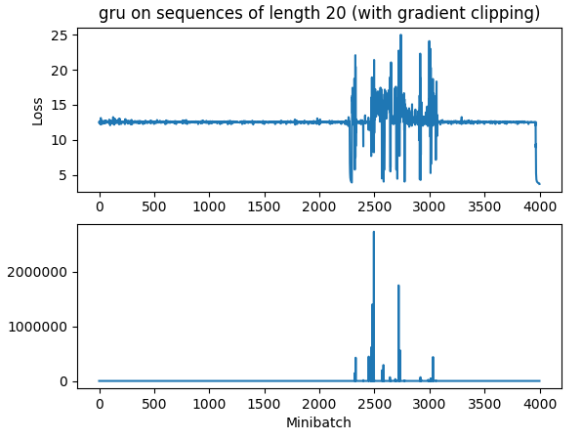

Figure 4: gru with clip