CS 224N: Assignment 3

PETER888@STANFORD.EDU SATURDAY 24TH FEBRUARY, 2018

Problem 1. A window into NER (30 points)

1.1 (a) Understanding NER (5 points, written)

1.1.1 i) Ambiguous Examples (2 points)

Answer:

1. ”Walt Disney has just bought Fox for 52bn”, here ”Walt Disney” and ”Fox” are ambiguous type. ”Walt Disney” can stands for person, the founder of Walt Disney Company, or can stands for ”Walt Disney” organization, the company.

2. ”What does the Fox say”, here ”Fox” is ambiguous type. ”Fox” can stands for the animal, or can stands for ”Fox” organization, the company.

1.1.2 ii) Why use features (1 point)

Answer:

Words can have multiple means depending on it’s context. We need predict named entity labels from their context as well.

1.1.3 iii) Feature examples (2 points)

Answer:

1. Adjacent Verb. For example ”Rose said”, here Rose should be a person, not the flower rose, because it’s adjacent verb ”said”

2. Location. If ”Chase” happening on highway, that’s normally the verb chase, instead of the bank ”Chase”

1.2 (b) Computational complexity (5 points, written)

1.2.1 i) Dimensions (2 points)

Answer:

$\boldsymbol{e}^{(t)} \in \mathbb{R}^{1\times((2\omega+1)\times D)}$, $\boldsymbol{W} \in \mathbb{R}^{(2\omega+1)\times H}$, $\boldsymbol{U} \in \mathbb{R}^{H\times C}$

1.2.2 ii) Complexity (3 point)

Answer:

For each step $\boldsymbol{e}^{(t)}$ requires $\boldsymbol{O}((2\omega+1)\times D)$ computation, $\boldsymbol{h}^{(t)}$ requires $\boldsymbol{O}((2\omega+1)\times D\times H)$ computation, and $\hat{\boldsymbol{y}}^{(t)}$ requires $\boldsymbol{O}(H\times C)$ computation.

In total, requires $\boldsymbol{O}(T\times((2\omega+1)\times D+(2\omega+1)\times D\times H+H\times C))$ computation. Ignore the constants and $D, H >> C$, the computation complexity of whole sentence is $\boldsymbol{O}(T\times\omega\times D\times H)$

1.3 (c) Implement model(15 points, code)

Answer:
See code: ~/`q1_window.py`.
And deliverable ~/`window_predictions.conll`

1.4 (d) Analyze the predictions (5 points, written)

1.4.1 i) Report best F_1 score (1 point)

Answer: *Get best F_1 score 0.84.*

Token-level confusion matrix:					
go/gu	**PER**	**ORG**	**LOC**	**MISC**	**O**
PER	**2935.00**	38.00	90.00	16.00	70.00
ORG	135.00	**1659.00**	120.00	51.00	127.00
LOC	36.00	118.00	**1863.00**	14.00	63.00
MISC	45.00	52.00	56.00	**999.00**	116.00
O	36.00	43.00	20.00	23.00	**42637.00**

Explanation: *The major errors which we can see from the confusion matrix are predict ORG as PER, O or LOC, and still errors are predict LOC to ORG, MISC as O.*

1.4.2 ii) Describe limitation (4 points)

Answer:

Limitation 1. Window based prediction is limited by it's window size, human can easy understand ”Sixflags Discovery Kingdom” should be an organization based on the coming word ”located”, but our model only have window size 2, so it can't do well on window longer than that.

Limitation 2. Context and surrounding tags influence. Humans are easily predict ”Kingdom” should related with ”Discovery” and should connected to each other in same entity name, and should be ORG

x :	Sixflags	Discovery	Kingdom	is	located	at	northern	California
y':	PER	ORG	LOC	O	O	O	O	LOC

Problem 2. Recurrent neural nets for NER (40 points)

2.1 (a) Computational complexity (4 points, written)

2.1.1 i) How many more (1 point)

Answer:

RNN model has $(H \times H) - (2\omega \times D \times H)$ more parameters than window-based model.

2.1.2 ii) Complexity (3 point)

Answer:

The un-simplified complexity is $O(T \times (D + ((D + H) \times H) + (H \times C + C)))$, ignore those minor ones, the complexity is $O(T \times (D \times H + H \times H + H \times C)$

2.2 (b) F_1 score (2 points, written)

2.2.1 i) When CE cost and F_1 decreasing at same time (1 point)

Answer:

Consider predicting tokens entity being all 'O's to predicting some tokens correct. Token accuracy will increase, that can lower the cross-entropy lost. But the model will have lower precision, and recall still the same.

2.2.2 ii) Why not F_1 (1 point)

Answer:

F1 is not a convex function, make it hard to optimize. And F1 is a over-all-metric, make it hard to batch or parallelize.

2.3 (c) RNN cell (5 points, code)

Answer: See code: ~/`q2_rnn_cell.py`.

2.4 (d) RNN model (8 points, code/written)

2.4.1 i) Loss and Gradient Update (3 points, written)

Answer:

The loss also include those padded zeros. And will then impact the gradient and will propagated back and impact the learning of the weights. After apply the masking, the loss come with the padded zeros will be zero, thus don't impact the learning.

2.4.2 ii) (5 points, code)

Answer: See code: ~/q2_rnn.py.

2.5 (e) Full RNN model (12 points, code)

Answer: See code: ~/q2_rnn.py.

2.6 (f) Train RNN model (3 points, code)

Answer: See result file: ~/rnn_predictions.conll.

2.7 (g) RNN's limit (6 points, written)

2.7.1 i) RNN limitations (3 points, written)

Answer:

Limitation 1. RNN is just have forward impact when learning. It can't predict ”Chenzhou” is a location, for sentence ”Chenzhou/0 is/0 a/0 small/0 city/0”, but according to the future information ”city”, it can predict it is a location.

Limitation 2. This RNN model doesn't group nearby tokens. Such as ”Sixflags/PER Discovery/ORG Kingdom/LOC is/0 in/0 northern/0 California/LOC”, but human can know the ”Sixflags” ”Discovery” and ”Kingdom” should group together form a ORG name.

2.7.2 ii) Suggestions (3 points, written)

Answer:

1. To fix the one way impact issue, we can use Bi-direction model like Bi-RNN

2. To fix the nearby token issue, we can use token-pair to nearby token group, in loss function or argument in the training data.

Problem 3. Grooving with GRUs (30 points)

3.1 (a) Modeling latching behavior (4 points, written)

3.1.1 i) RNN cell values (1 point)

Answer:

There are four x, h^{t-1} value pairs:

$x = 0, h^{t-1} = 0$, expect $h^t = \sigma(b_h) = 0$, thus $b_h \leqslant 0$
$x = 0, h^{t-1} = 1$, expect $h^t = \sigma(U_h + b_h) = 0$, thus $U_h + b_h > 0$
$x = 1, h^{t-1} = 0$, expect $h^t = \sigma(W_h + b_h) = 1$, thus $W_h + b_h > 0$
$x = 1, h^{t-1} = 1$, expect $h^t = \sigma(W_h + U_h + b_h) = 1$, thus $W_h + U_h + b_h > 0$

So, the values should be $b_z \leqslant 0$, $U_h + b_h > 0$, and $W_h + b_h > 0$.
One example could be $W_h = 1, U_h = 1$ and $b_h = -0.5$

3.1.2 ii) GRU cell values (3 points)

Answer:

Let $W_r = U_r = b_r = b_z = b_h = 0$, get new equations:

$z^{(t)} = \sigma(x^{(t)}W_z + h^{(t-1)}U_z)$
$r^{(t)} = 0$
$\tilde{h}^{(t)} = tanh(x^{(t)}W_h + r^t \circ h^{(t-1)}U_h) = tanh(x^{(t)}W_h)$
$h^{(t)} = z^{(t)} \circ h^{(t-1)} + (1 - z^{(t)}) \circ \tilde{h}^{(t)}$

There are four x, h^{t-1} value pairs

$x = 0, h^{t-1} = 0$, expect $h^t = 0$,
have $z^{(t)} = \sigma(0) = 0$, $\tilde{h}^{(t)} = tanh(0) = 0$, $h^{(t)} = 0$
$x = 0, h^{t-1} = 1$, expect $h^t = 1$,
have $z^{(t)} = \sigma(U_z)$, $\tilde{h}^{(t)} = tanh(0) = 0$, $h^{(t)} = z^{(t)} \circ h^{t-1} = \sigma(U_z) = 1$
$\rightarrow U_z > 0$
$x = 1, h^{t-1} = 0$, expect $h^t = 0$,
have $z^{(t)} = \sigma(W_z)$, $\tilde{h}^{(t)} = tanh(W_h)$, $h^{(t)} = (1 - z^{(t)}) \circ \tilde{h}^{(t)} = 0$
$\rightarrow W_z > 0$ or $W_h \leqslant 0$
$x = 1, h^{t-1} = 1$, expect $h^t = 1$,
have $z^{(t)} = \sigma(W_z + U_z)$, $\tilde{h}^{(t)} = tanh(W_h)$, $h^{(t)} = z^{(t)} + (1 - z^{(t)}) \circ \tilde{h}^{(t)} = 1$
$\rightarrow W_z + U_z > 0$ or ($W_z + U_z \leqslant 0$ and $W_h > 0$)

3.2 (b) Modeling toggling behavior (6 points, written)

3.2.1 i) 1D RNN (3 points)

Answer:

3.2.2 ii) GRU cell values (3 points)

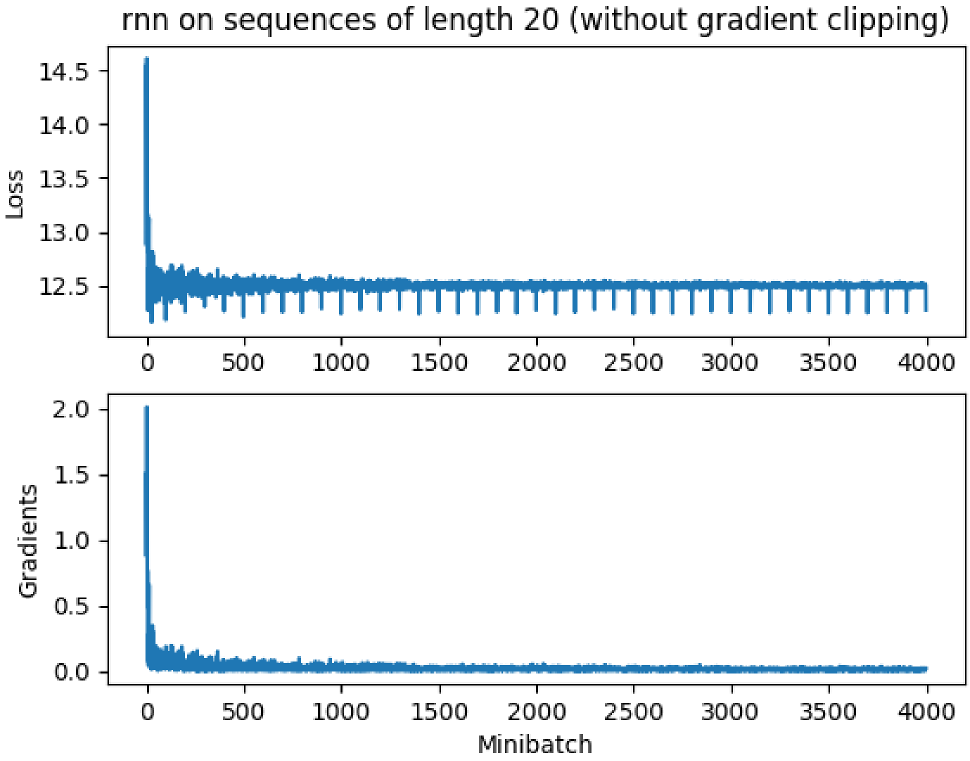
Answer:

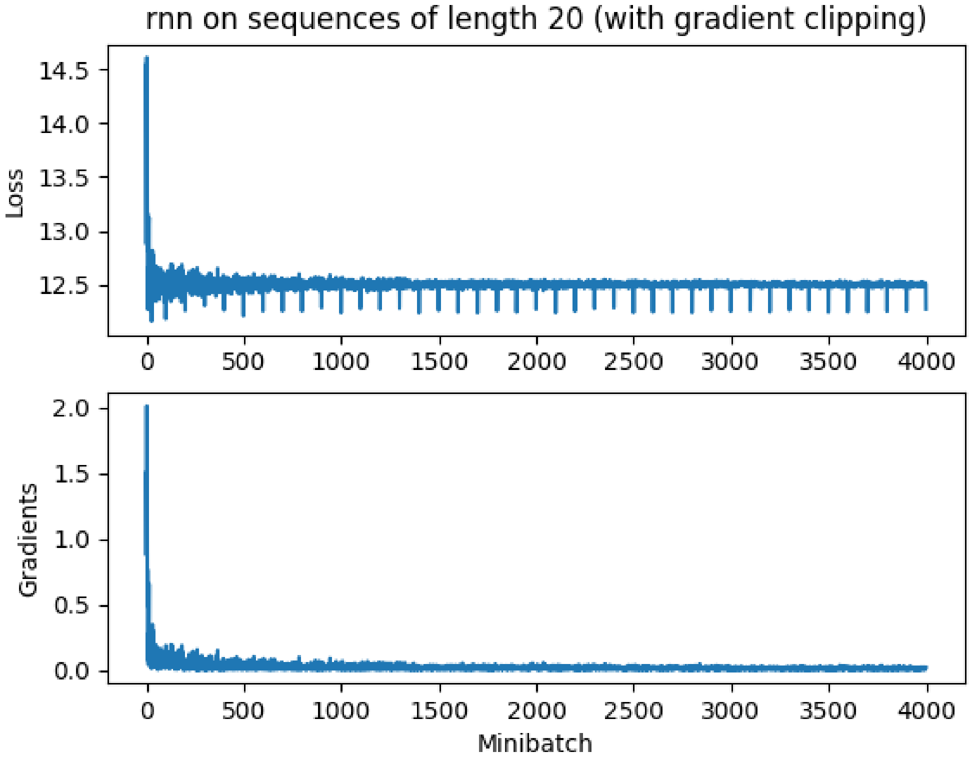
3.3 (c) GRU cell (6 points, code)

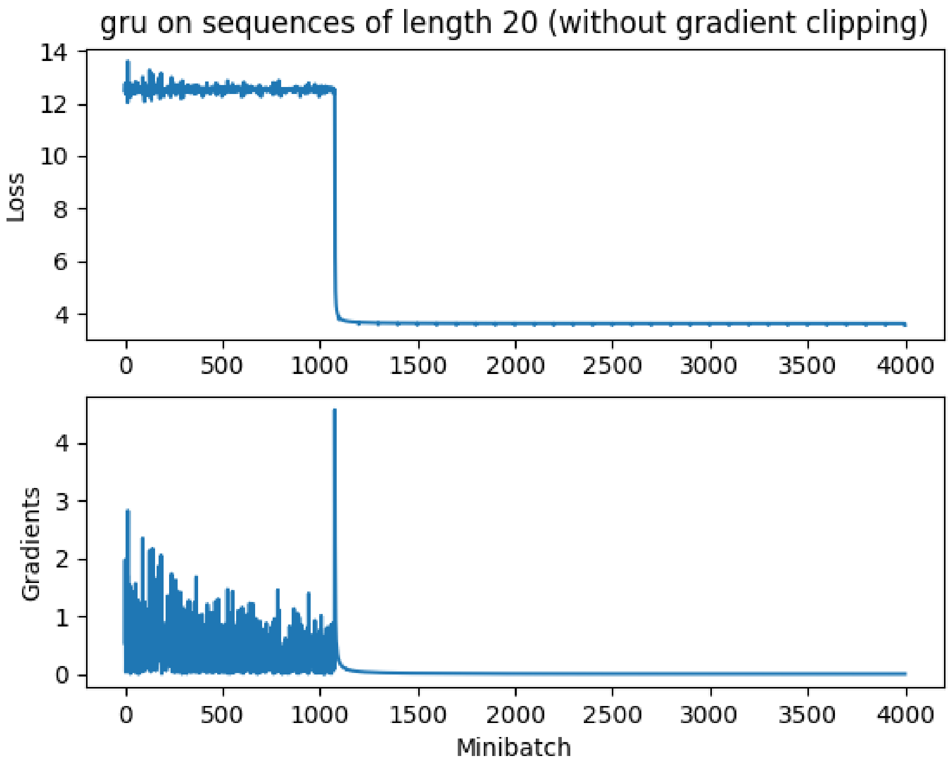
see code ∼/`q3_gru_cell.py`

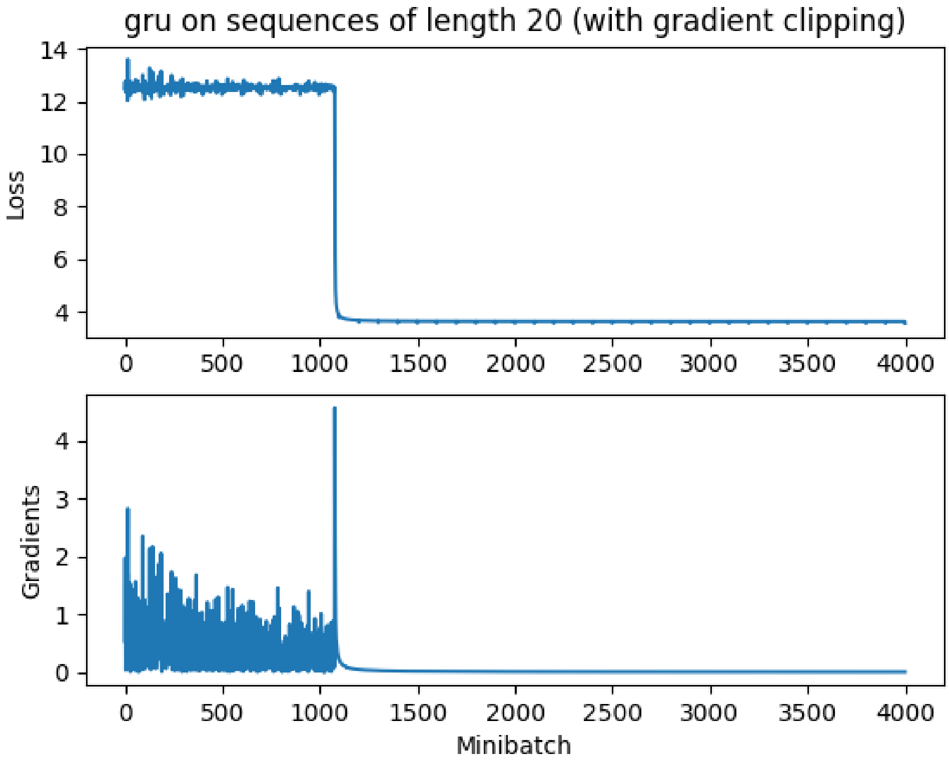
3.4 (d) Dynamic RNN (6 points, code)

see code ∼/`q3_gru.py` and next page plots (total 4)


rnn on sequences of length 20 (without gradient clipping)
Loss
14.5
14.0
13.5
13.0
12.5
Gradients
2.0
1.5
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
14.5
14.0
13.5
13.0
12.5
Gradients
2.0
1.5
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
3
2
1
0
0
500
1000
1500
2000
2500
3000
3500
4000
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
3
2
1
0
0
500
1000
1500
2000
2500
3000
3500
4000
Minibatch

3.5 (e) Analyze graphs (5 points, written)

Answer:

It's fine if GRU gradients are still below 5.0 without gradient clipping; just make conclusions based on what you see in your graphs. A few things to consider:

- Are gradients increasing or decreasing for the GRU with and without gradient-clipping? Is this expected?
- Does loss improve or stay constant for the RNN vs. GRU? Does gradient-clipping affect the loss? Why might this be the case?

3.6 (f) Train GRU (3 points, code)

Answer: See file: ~/`gru_predictions.conll`.