CS 224N

1.a) i. "JORDAN IS BEAUTIFUL"
PER OR LOC

"I TOOK A LOAN FROM GOLDMAN".
ORG OR PER

ii. IT MIGHT BE IMPORTANT TO USE THOSE WORDS TO IDENTIFY THE CONTEXT AND CLEAR THOSE AMBIGUITIES.

iii. → THE FOLLOWING WORD'S POS (E.G., IF IT IS A VERB, THE CHANCES OF THE PRECEDING WORD BEING A NAMED ENTITY IS HIGHER)

→ THE PREVIOUS WORD'S POS (E.G., IF IT IS A WORD SUCH AS "TO", "FROM", "IN", THE CHANCES OF THE FOLLOWING WORD BEING A NAMED ENTITY IS HIGHER).

1.b) i. $e^{(t)}: 1 \times (2w+1) \times D$ $h^{(t)}: 1 \times H$

$W: (2w+1)D \times H$

$U: H \times C$

ii. $e^{(t)}.W \Rightarrow O((2w+1)HD)$

$h^{(t)}.U \Rightarrow O(HC)$

} THOSE OPERATIONS DOMINATE THE COMPL

WE HAVE T WORDS IN THE SENTENCE.

THEREFORE, OVERALL COMPLEXITY IS $O(TH \cdot (C + (2w+1)D))$

CS224N

1.d) i. BEST F1 SCORE: 0.83

TOKEN-LEVEL CONFUSION MATRIX.

GO \ GU (PREDICTED)	PER	ORG	LOC	MISC	O
PER	2921	50	54	21	103
ORG	135	1657	99	55	146
LOC	43	112	1866	19	54
MISC	45	60	38	1011	114
O	37	55	13	26	42628

(ACTUAL)

THE MODEL OFTEN PREDICTS "O" WHEN IT'S ACTUALLY "ORG" OR "MISC", AND ALSO PREDICTS "PER" WHEN IT'S ACTUALLY "ORG" VERY FREQUENTLY

CS 224N

1.d)ii.

- If the window size is not large enough, the model might miss some long continuous named entities with connector words inside it.

Example:

x:	May	15	&	Duke	of	Norfolk	...
y*:	O	O	O	ORG	ORG	ORG	...
y':	O	O	O	ORG	O	ORG	...

- The model might not have enough memory to understand context and discriminate between different types of named entities.

Example:

x:	...	until	the	sixth	game,	when	Washington,	after	winning	...
y*:	...	O	O	O	O	O	PER	O	O	...
y':	...	O	O	O	O	O	LOC	O	O	...

CS 224N

2.a) i. Window based model: $(2w+1)DH$ (from W)
$H \times C$ (from U)
H (from b_1)
C (from b_2)

Not considering L, since it's the same for both

RNN: $H \times H$ (from W_h)
$D \times H$ (from W_x)
H (from b_1)
$H \times C$ (from U)
C (from b_2)

So we have $H^2 + DH + H + HC + C - C - H - HC - (2w+L)DH =$

$H \cdot (H - 2wD)$ more parameters.

2.a) ii. We need to go over each of the T words. So we will have a factor of $O(T)$. Now, for each word, the dominating operations are

$h^{(t-1)} W_h \Rightarrow O(H^2)$ ($1 \times H$, $H \times H$)

$e^{(t)} \cdot W_x \Rightarrow O(DH)$ ($1 \times D$, $D \times H$)

$h^{(t)} U \Rightarrow O(HC)$ ($L \times H$, $H \times C$)

Total complexity: $O(TH(H+D+C))$.

CS 224N

2.b)i.

FROM	y	PER	O	PER	PER	PER	PER
	$\tilde{y}$	PER	O	O	O	O	O
TO	y	PER	O	PER	PER	PER	PER
	$\tilde{y}$	O	O	O	PER	PER	PER

THE ENTITY LEVEL F_1 IS NOW 0 (DECREASED), EVEN THOUGH THE CROSS ENTROPY COST WOULD HAVE DECREASED (ASSUMING $\hat{y}_{PER} = 1$ IF $\tilde{y} = PER$, $\hat{y}_O = 1$ IF $\tilde{y} = O$).

2.b)ii. IT'S DIFFICULT TO DIRECTLY OPTIMIZE FOR F_1 BECAUSE IT IS A NON-DIFFERENTIABLE FUNCTION (NEITHER PRECISION OR RECALL ARE).

CS 224N

2. d) i.

Since those new ys are filled with 0s, and

$$CE(y^{(t)}, \hat{y}^{(t)}) = \sum_i y_i^{(t)} \log(\hat{y}_i^{(t)}) = \sum_i 0 \cdot \log(\hat{y}_i^{(t)}) = 0,$$

the loss J doesn't change. However, the gradients do change: as we saw before, if $\hat{y} = \text{softmax}(\theta)$ and $J = CE$ loss, $\frac{\partial J}{\partial \theta} = (\hat{y} - y)$. Even though y is filled with 0s, $\hat{y}$ isn't (since it is the output of a softmax). So if we don't apply the mask, we will see some non-zero gradient updates on the parameters when we do backpropagation after the pass of one of those "new" pair of vectors (x, y). Masking solves the problem by having on $m^{(t)} = 1 - \mathbb{1}\{t > T\}$ multiplying the loss (and, thus, the gradients) for each pass $t = 1, \ldots, T$. That would set to 0 both the loss and the gradients for those "new" pairs (x, y), solving the issue.

CS 224N

2.g)i, ii. • the model only reads the words until the current word t, i.e., just learn the past context.

Example:

x	...	the	Stars	and	Stripes	flag	was	...
y^*	...	O	MISC	MISC	MISC	O	O	...
y'	...	O	LOC	O	MISC	O	O	...

Model extension: build a <u>bidirectional</u> recurrent neural network.

• For long sentences, the model might be losing information about the context as it reaches the end of the sentence, which could be due vanishing gradients or just the fact that it might not be capturing the relevant information that it should save for the future.

Example:

x	...[long sentence]...	government	of	Fulgencio	Batista	.
y^*	...	O	O	PER	PER	O
y'	...	O	O	ORG	ORG	O

Model extensions: using ReLu as the activation function, or even move to more sofisticated models such as GRUs or LSTMs

CS 224N

3. a) i. Possible cases:

→ First element is 0: $h^1 = 0 = \sigma(0 + 0 + b_h) \Rightarrow b_h \leq 0$

→ First element is 1: $h^1 = 1 = \sigma(U_h + 0 + b_h) \Rightarrow U_h + b_h > 0$

→ Keep previous element 1: $\begin{cases} h^t = 1 = \sigma(1 + W_h + b_h) \Rightarrow W_h + b_h > 0 \\ h^t = 1 = \sigma(U_h \cdot - + W_h + b_h) \Rightarrow U_h + W_h + b_h > 0 \end{cases}$

→ Keep previous element 0 (analogous to 1st case).

Setting values: $\boxed{b_h = -1 \; ; \; U_h = 2 \; ; \; W_h = 2}$

CS 224N

3.a) ii. $w_r = u_r = b_r = b_z = b_h = 0$:

$$z^t = \sigma(x^t u_z + h^{t-1} w_z)$$

$$r^t = \sigma(0 + 0 + 0) = \sigma(0) = 0$$

$$\tilde{h}^t = \tanh(x^t u_h + 0 + 0)$$

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

Possible cases:

→ Keep previous (current) 0: $h^t = 0 = \underbrace{\sigma(0)}_{z^t}.0 + (1-0).\underbrace{\tanh(0)}_{=0} = 0 \Rightarrow$ O.K.

→ $x^t = 0$, $h^{t-1} = 1$: $h^t = 1 = \sigma(w_z).1 + (1-\sigma(w_z)).\underbrace{\tanh(0)}_{=0} \Rightarrow w_z > 0$

→ $x^t = 1$, $h^{t-1} = 0$: $h^1 = 1 = \sigma(u_z).0 + (1-\sigma(u_z)).\tanh(u_h) \Rightarrow \begin{cases} u_z \leq 0 \\ u_h > 0 \end{cases}$

Possible solution:

$$\boxed{w_z = 1;\ u_z = -1;\ u_h = 1;\ w_h = 0}$$

$w_h \Rightarrow$ any value

CS 224N

3. b) i.

Possible cases:

→ $h^{t-1}=0, x=0$: $h^t = 0 = \sigma(0+0+b_h) \Rightarrow b_h \leq 0$ (I)

→ $h^{t-1}=0, x=1$: $h^t = 1 = \sigma(u_h+0+b_h) \Rightarrow u_h + b_h > 0$ (II)

→ $h^{t-1}=1, x=0$: $h^t = 1 = \sigma(0+w_h+b_h) \Rightarrow w_h + b_h > 0$ (III)

→ $h^{t-1}=1, x=1$: $h^t = 0 = \sigma(u_h+w_h+b_h) \Rightarrow u_h + w_h + b_h \leq 0$ (IV)

$-(IV)+(II)-(III)$: $-u_h - w_h - b_h + u_h + b_h + w_h + b_h \geq 0 \Rightarrow b_h \geq 0$.

From (I), we get that $b_h = 0$.

So, looking again at (II)+(III), we get $u_h + w_h > 0$, which is impossible since, from (IV), $u_h + w_h \leq 0$.

CS 224N

3.b) ii. $w_r = u_r = b_z = b_h = 0$

$$z^{(t)} = \sigma(x^{(t)} u_z + h^{(t-1)} w_z)$$

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

$$\tilde{h}^{(t)} = \tanh\left(x^{(t)} u_h + \sigma(b_r) w_h \cdot h^{(t-1)}\right)$$

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

Cases:

$\rightarrow h^{(t-1)} = 0;\ x^{(t)} = 0 \Rightarrow h^{(t)} = 0 = \sigma(0) \cdot 0 + \overbrace{(1-\sigma(0))}^{1} \cdot \overbrace{\tanh(0)}^{0} = 0$

$\rightarrow h^{(t-1)} = 0;\ x^{(t)} = 1$:

$$h^{(t)} = 1 = \overbrace{\sigma(u_z) \cdot 0}^{0} + (1 - \sigma(u_z)) \cdot \tanh(u_h + 0) \Rightarrow u_z \ll 0,\ u_h > 0$$

$\rightarrow h^{(t-1)} = 1;\ x^{(t)} = 0$:

$$h^{(t)} = 1 = \sigma(w_z) \cdot \ldots + (1 - \sigma(w_z)) \cdot \tanh(\sigma(b_r) w_h) \Rightarrow \begin{cases} w_z > 0 \\ \text{or} \\ b_r > 0,\ w_h > 0 \end{cases}$$

$\rightarrow h^{(t-1)} = 1;\ x^{(t)} = 1$:

$$h^{(t)} = 0 = \sigma(u_z + w_z) + (1 - \sigma(u_z + w_z)) \cdot \tanh(u_h + \sigma(b_r) w_h) \Rightarrow \begin{cases} u_z + w_z \ll 0 \\ \text{and} \\ u_h + \sigma(b_r) w_h \ll 0 \end{cases}$$

POSSIBLE SOLUTION:

$b_r = 1$	$u_z = -1$	$u_h = 1$
$w_z = 1$	$w_h = -1$	

CS 224N

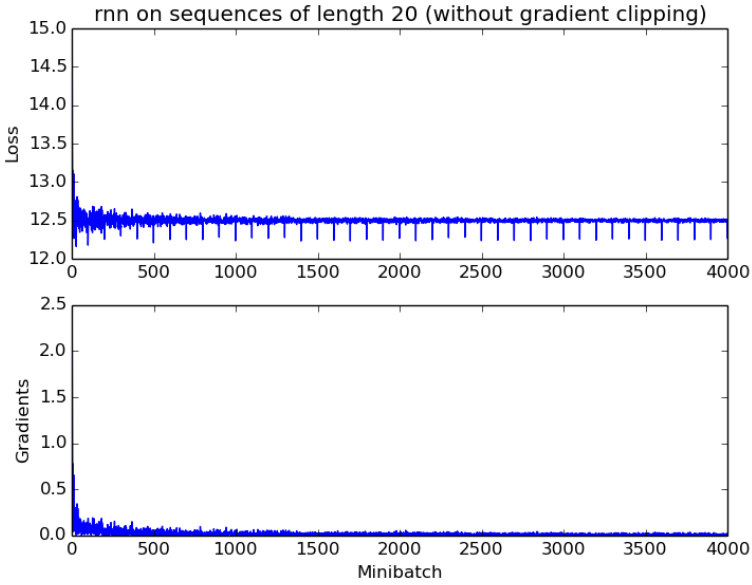
2.d) ii.

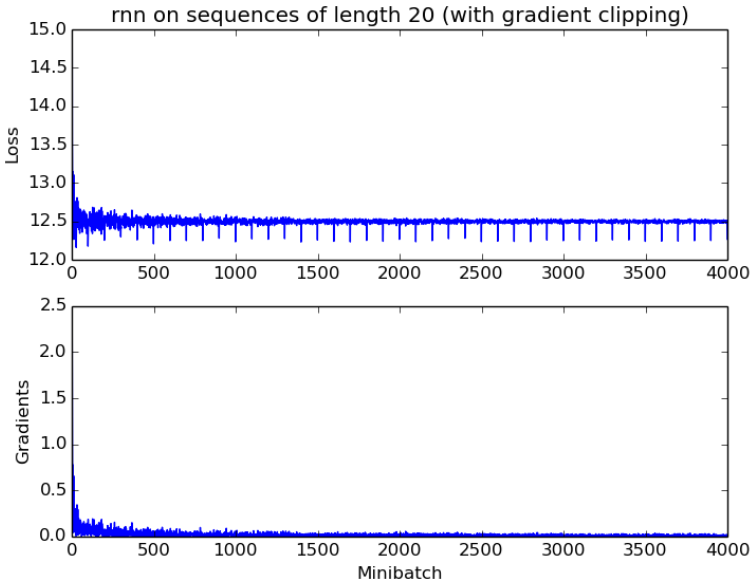
If we did not use masking, the loss would increase, since we would be adding $M-T$ terms of the form $\hat{y}^{(t)}$, $t>T$.

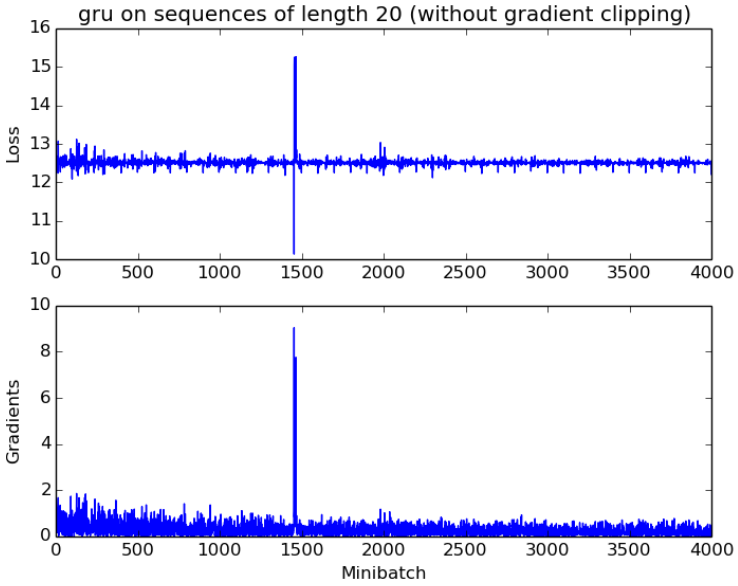
Also, the gradient would change. As we saw before, if $\hat{y} = \text{softmax}(\theta)$ and $J = CE$ loss, $\frac{\partial J}{\partial \theta} = (\hat{y} - y)$. So not only y has a 1 in the "NULL" position, but $\hat{y}$ is filled with nonzero elements as well. This first error δ backpropagates and will end up affecting most parameters gradients.

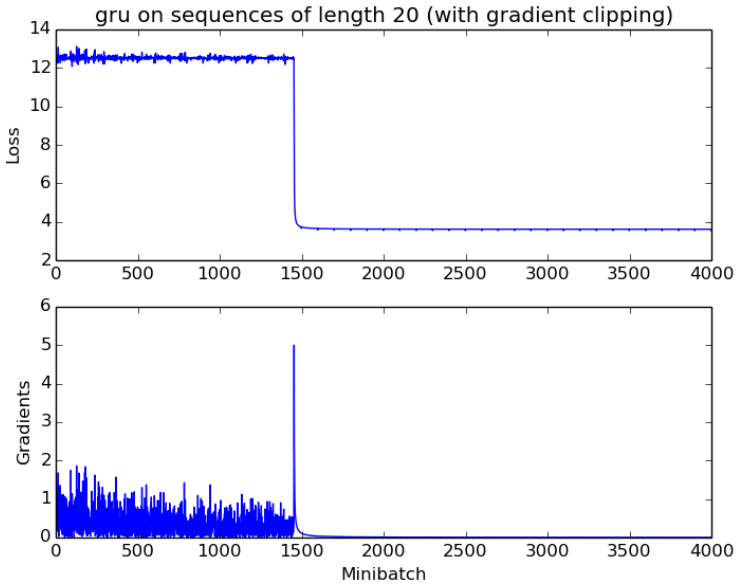
Masking solves the problem by having an $m^{(t)} = \mathbb{1}\{t \leq T\}$ multiplying each term of the loss (and, thus, the gradients), for each pos $t = 1, \ldots, M$. That would set to 0 both loss and gradients for $t = T+1, \ldots, M$ for $(x^{(t)}, y^{(t)})$, solving the issue (i.e., the new gradients and loss would be equivalent as if we didn't do the augmentation).

Question 3.d)


rnn on sequences of length 20 (without gradient clipping)
Loss
15.0
14.5
14.0
13.5
13.0
12.5
12.0
0
500
1000
1500
2000
2500
3000
3500
4000
Gradients
2.5
2.0
1.5
1.0
0.5
0.0
Minibatch


rnn on sequences of length 20 (with gradient clipping)
Loss
15.0
14.5
14.0
13.5
13.0
12.5
12.0
0
500
1000
1500
2000
2500
3000
3500
4000
Gradients
2.5
2.0
1.5
1.0
0.5
0.0
Minibatch


gru on sequences of length 20 (without gradient clipping)
Loss
Gradients
Minibatch


gru on sequences of length 20 (with gradient clipping)
Loss
Gradients
Minibatch

Question 3.e)i.

It seems like the RNN model experience vanishing gradients, since the gradients become very small and the model get stuck in some local minima. On the other hand, the GRU model seems to experience exploding gradient, since at some point it reaches a loss significantly better than before, but a large gradient pushes it to a worse local minimia region. Clipping does help, as we can see from the last pair of graphs: after it reaches this better loss, the clipped gradient is not large enough to push it to the worse local minima, and then we stay in the same region, eventually converging to a significantly better loss.

Question 3.e)ii.

GRU does better since it is able to achieve a much lower loss. This doesn't necessarily guarantees a better generalization error, but the fit is definitely better, we achieve a much lower error and the model seems to allow for better accuracy.