Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Natural Language Processing
Spring 2021

Prof. Ryan Cotterell

Course Assignment

July 10, 2021

Karol Borkowski
nethz Username: kborkowski
Student ID: 00000000

Collaborators:
Other student 1
Other student 2

By submitting this work, I verify that it is my own. That is, I have written my own solutions to each problem for which I am submitting an answer. I have listed above all others with whom I have discussed these answers.

Part I

Course Assignment Episode 1

Question 1

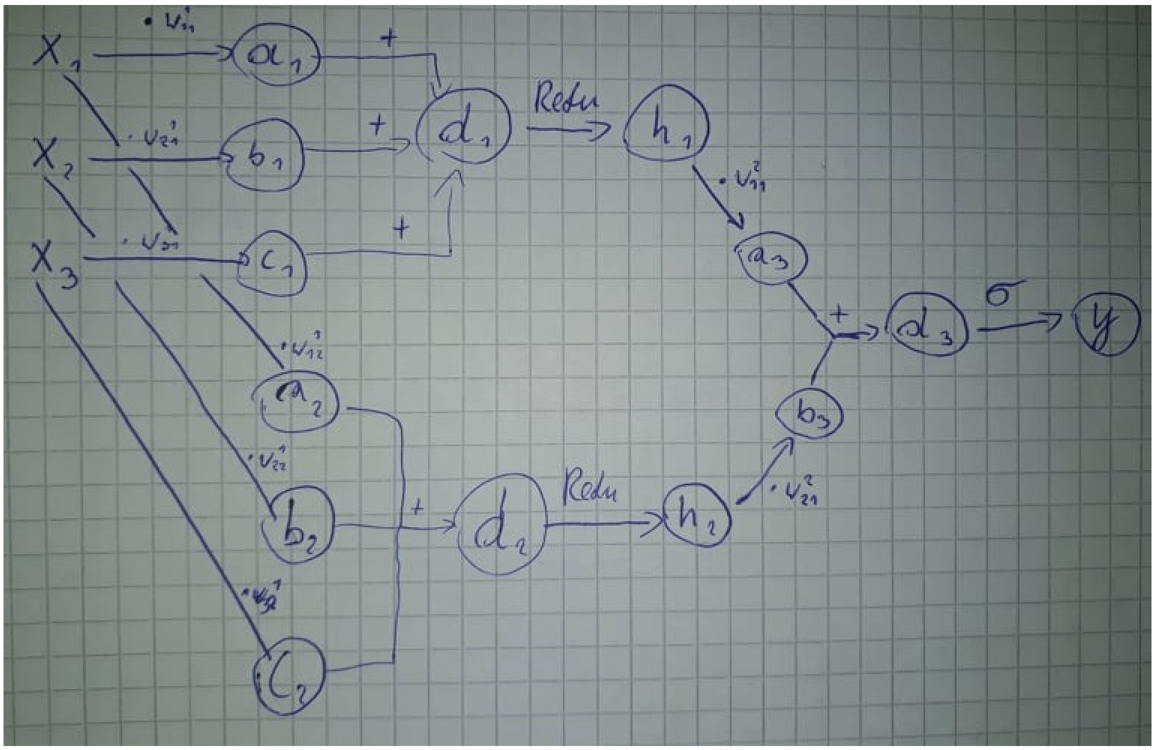

Figure 1: The computation graph of f.

(a) $a_1 = x_1 \cdot w_{11}^1 \quad b_1 = x_2 \cdot w_{21}^1 \quad c_1 = x_3 \cdot w_{31}^1$

$d_1 = a_1 + b_1 + c_1$

$h_1 = ReLu(d_1)$

$a_2 = x_1 \cdot w_{12}^1 \quad b_2 = x_2 \cdot w_{22}^1 \quad c_2 = x_3 \cdot w_{32}^1$

$d_2 = a_2 + b_2 + c_2$

$h_2 = ReLu(d_2)$

$a_3 = h_1 \cdot w_{11}^2 \quad b_3 = h_2 \cdot w_{21}^2$

$d_3 = a_3 + b_3$

$y = \sigma(d_3)$

(b) (i) $h_1 = ReLu(1 + 1 + 1) = 3 \quad h_2 = ReLu(1 + 1 + 1) = 3$

$y = \sigma(3 + 3) \simeq 0.99753$

(ii) $\frac{\partial L}{\partial \hat{y}} = \frac{-y}{\hat{y}} - \frac{1-y}{1-\hat{y}} \simeq -404$

$\frac{\partial y}{\partial d_3} = \sigma(d_3) \cdot (1 - \sigma(d_3)) \simeq 0.00247$

$\frac{\partial d_3}{\partial a_3} = 1$

$\frac{\partial d_3}{\partial b_3} = 1$

$\frac{\partial a_3}{\partial w_{11}^2} = h_1 = 3$

$\frac{\partial b_3}{\partial w_{21}^2} = h_2 = 3$

$\frac{\partial a_3}{\partial h_1} = w_{11}^2 = 1$

$\frac{\partial b_3}{\partial h_2} = w_{21}^2 = 1$

$\frac{\partial h_1}{\partial d_1} = 1$

$\frac{\partial h_2}{\partial d_2} = 1$

$\frac{\partial d_1}{\partial a_1} = 1$

$\frac{\partial d_1}{\partial b_1} = 1$

$\frac{\partial d_1}{\partial c_1} = 1$

$\frac{\partial d_2}{\partial a_2} = 1$

$\frac{\partial d_2}{\partial b_2} = 1$

$\frac{\partial d_2}{\partial c_2} = 1$

$\frac{\partial a_1}{\partial w_{11}^1} = 1$

$\frac{\partial b_1}{\partial w_{21}^1} = 1$

$\frac{\partial c_1}{\partial w_{31}^1} = 1$

$\frac{\partial a_2}{\partial w_{12}^1} = 1$

$\frac{\partial b_2}{\partial w_{22}^1} = 1$

$\frac{\partial c_2}{\partial w_{32}^1} = 1$

$\frac{\partial a_1}{\partial x_1} = 1$

$\frac{\partial b_1}{\partial x_2} = 1$

$\frac{\partial c_1}{\partial x_3} = 1$

$\frac{\partial a_2}{\partial x_1} = 1$

$\frac{\partial b_2}{\partial x_2} = 1$

$\frac{\partial c_2}{\partial x_3} = 1$

(iii) $\frac{\partial L}{\partial \hat{y}} = \frac{-y}{\hat{y}} - \frac{1-y}{1-\hat{y}} \simeq -404$

(iv) $\Delta w_{ij} = -\eta \frac{\partial L}{\partial w_{ij}}$

$\frac{\partial L}{\partial w_{11}^2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial d_3} \cdot \frac{\partial d_3}{\partial w_{11}^2} = -0.99788$

$\Delta w_{11}^2 = 0.1 \cdot -0.99788 = -0.099788$

$\frac{\partial L}{\partial w_{21}^2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial d_3} \cdot \frac{\partial d_3}{\partial w_{21}^2} = -0.99788$

$\Delta w_{21}^2 = 0.1 \cdot -0.99788 = -0.099788$

$\frac{\partial L}{\partial w_{11}^1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial d_3} \cdot \frac{\partial d_3}{\partial a_3} \cdot \frac{\partial a_3}{\partial h_1} \cdot \frac{\partial h_1}{\partial d_1} \cdot \frac{\partial d_1}{\partial a_1} \cdot \frac{\partial a_1}{\partial w_{11}^1} = -0.99788$

$\Delta w_{11}^1 = 0.1 \cdot -0.99788 = -0.099788$

Since the input values are the same and the weight values are also the same, the rest of the gradient changes have the same value. The calculation are analogous to the above ones.

(c) The symmetry of weights is not broken, so the hidden units can be replaced with a single unit producing the same output. The benefits of such solution are: faster training, more reliable model due to fewer parameters.

Question 4

(a) (i) Let's define an auxiliary function $J_w = \begin{cases} 1, & \text{if } w \textit{ is drawn} \\ 0, & \text{otherwise} \end{cases}$

The number of unique tokens X is equal to:

$X = \sum_{w=1}^{|V|} J_w$

Using the linearity of expectations, the expected value of X is equal to:

$E[X] = E[\sum_{w=1}^{|V|} J_w] = \sum_{w=1}^{|V|} E[J_w]$

where:

$E[j_w] = P(J_w = 1) = P(\textit{at least one w selected}) = 1 - P(\textit{w not selected}) = 1 - \frac{|V|-1}{|V|}$

Finally: $X = |V| \cdot (\frac{|V|-1}{|V|})^n$

(ii) P(all words appear) = 1 - P(at least one w doesn't appear) $= 1-(P(!w_1) \bigcup P(!w_2) \bigcup ... \bigcup P(!w_|V|) = 1 - |\bigcup_{i=1}^{|V|} P(!w_i)|)$

where '!' denotes that the word is not selected. The above union can be found using the inclusion-exclusion principle.

$|\bigcup_{i=1}^{|V|} P(!w_i)| = \sum_{J \in \{1,...,|V|\}} (-1)^{|J|+1} |\bigcup_{j \in J} P(!w_i)|) =$

(b) (i) A - expected additional draws if just selected any word other than 'work'

B - expected additional draws if just selected 'work'

A is equal to the initial draw + P(w != 'hard') * A + P(w = 'hard') * B, what yields:

$A = 1 + \frac{|V|-1}{|V|} \cdot A + \frac{1}{|V|} \cdot B$

B is equal to one draw in case that we select word 'hard' + P(w != 'hard') * A, what yields:

$B = 1 + \frac{|V|-1}{|V|} \cdot A$

Solving the above system of equations in terms of A, we get:

$A = \frac{|V|^2-1}{|V|-1}$

(ii) TO DO

(c) (i) A - expected number of draws before the first selection

B - expected additional number of draws after at least one word has been selected.

A is equal to B + the initial draw:

$A = B + 1$

B is equal to P(w is the same as the previous one) + P(w is different) * A:

$B = \frac{1}{|V|} + \frac{|V|-1}{|V|} \cdot A$

The solution of the above set of equations in terms of A is:

$A = |V| + 1$

(ii) I assume that the input is an integer corresponding to a given word index.

$w_0 = 1$

$w_1 = -1$

$w_2 = 0$

$w_0 = 0$

$b_0 = 0$

$f(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise} \end{cases}$

$g(x) = x$

(iii) $w_3 = 1$

$w_4 = 1$

$w_5 = 1$

$b_2 = 0$

$b_0 = 0$

$g(x) = x$

$h(x) = x$

A non-linear activation is required because the output cannot be express as a linear function of the inputs. It is similar to the XOR problem.

(iv) A non-uniform has a yields greater chances of sequentially drawing the same token. In can be intuitively depicted using an extreme case, when always the same word is drawn.

Question 5

(a) (i)

number	**sample strings**	**accepted**	**weight**
1	educational is this not		
2	is this assignment educational		
3	not educational is not educational		
4	this assignment is not educational		
5	is this assignment educational		
6	this assignment course is educational		
7	is this assignment not educational		
8	this assignment not		
9	this course assignment is not educational		
10	this course is not not educational		
11	not educational is this		
12	course assignment is not educational		
13	not this assignment is educational		
14	not not not educational		
14	is this course assignment not educational		
15	course assignment is this		
16	this course is interesting		
17	this course assignment not educational		

Table 1: Some strings from $\mathcal{Y}_{\geq 2, \leq 6}$

iteration: n = 0, weight matrix

	a this	**b** assign-ment	**c** course	**d** is	**e** not	**f** edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 0, backtracking matrix

	a this	**b** assign-ment	**c** course	**d** is	**e** not	**f** edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 1, weight matrix

	a this	**b** assign-ment	**c** course	**d** is	**e** not	**f** edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 1, backtracking matrix

	a this	**b** assign-ment	**c** course	**d** is	**e** not	**f** edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 2, weight matrix

	a this	**b** assign-ment	**c** course	**d** is	**e** not	**f** edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 2, backtracking matrix

	a this	**b** assign-ment	**c** course	**d** is	**e** not	**f** edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 3, weight matrix

	a this	**b** assign-ment	**c** course	**d** is	**e** not	**f** edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 3, backtracking matrix

	a this	**b** assign-ment	**c** course	**d** is	**e** not	**f** edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

Figure 2: Floyd-Warshall algorithm, iteration 0 to 3; left column matrix should contain weights after iteration n; right column matrix should be iteratively filled for backtracking each path

iteration: n = 4, weight matrix

	a this	b assign-ment	c course	d is	e not	f edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 4, backtracking matrix

	a this	b assign-ment	c course	d is	e not	f edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 5, weight matrix

	a this	b assign-ment	c course	d is	e not	f edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 5, backtracking matrix

	a this	b assign-ment	c course	d is	e not	f edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 6, weight matrix

	a this	b assign-ment	c course	d is	e not	f edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 6, backtracking matrix

	a this	b assign-ment	c course	d is	e not	f edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 7, weight matrix

	a this	b assign-ment	c course	d is	e not	f edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

iteration: n = 7, backtracking matrix

	a this	b assign-ment	c course	d is	e not	f edu-cational
a this						
b assignment						
c course						
d is						
e not						
f educational						

Figure 3: Floyd-Warshall algorithm, iteration 4 to 7; left column matrix should contain weights after iteration n; right column matrix should be iteratively filled for backtracking each path