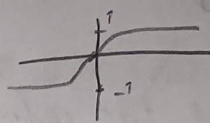
Amin Hosseiny Marani

(1)

1- using tanh faction will map inputs to -1, 1 →



2- if we remove tanh, the values of hidden layer will get bigger till they become NAN (±infinity)

3- These are values when the activation function applies tanh:

time	value		
1	0.98	-.99	-0.98
2	0.86	-0.96	-0.95
3	0.99	-0.44	-0.99
⋮	⋮	⋮	⋮
9	0.85	0.86	-0.66
10	-0.69	0.86	0.36

4- These are values in absence of tanh:

time	values		
1	-2.46	-4.72	-2.34
2	6.66	-8.37	-8.34
3	17.00	-6.66	-20.43
⋮	⋮	⋮	⋮
9	-539.22	-119.24	767.12
10	-536.11	-1078.15	624.04

5- activation function is used to indicate when and how much a neuron should fire. Later hidden states are used to produce output within V (weight) matrix. with these values getting larger not only soon O (output matrix) will become NAN, but also make gradient in backe propagation exploding.

(1) cont'd

6. plus the restriction of output for hiddien states (-1,1 for tanh), tanh function centers the feature values on 0 while removing it will not give the model this centrality.

Summary:

removing tanh $\xrightarrow{I}$ will make outputs NAN at some point.

$\xrightarrow{II}$ It will cause gradinet exploding.

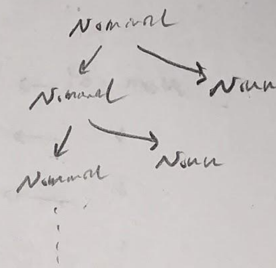
$\xrightarrow{III}$ one large value of input can affect hidden states values largly.

2

I - Prove the CFG at figure 1 allow derivations of infinite lengths?

Ans: Yes, it allows. Considering "Nominal → Nominal Noun" we can create an infinite length:

(1)
Nominal → Nominal Noun
→ Nominal Noun Noun
→ ... → Nominal $(\text{Noun})^N$



So using (1), and S → NP VP, and NP → Det Nominal we will have

S → NP VP
→ Det Nominal VP
→ $\boxed{\text{Det } \underbrace{\text{Nominal } (\text{Noun})^N}_{\text{ends at } (\text{Noun})^{N+1}} \text{ VP}}$ → for an infinite number N this can go on.

* this is just one way/example.
we can make more than 1 derivation with infinite length given this CFG.

II Justify or disaprove the existence of such derivations.

Ans: we know in English language we can make compound nouns. For example, "radar system design expert" consists of 4 nouns.

② Cont'd

on the other hand even if theorically having N nouns in a compound noun is possible, its rare to see compound nouns with $N > 5$. So, one way of solving this would be:

Nominal → Noun
" → Noun Noun
" → " " Noun
⋮
" → " " " Noun Noun

this may prevent this specific CFG from generating infinite length sentece, but this need extra exploration to make sure these new rules can do exactly similar what "nominal → Nominal Noun" could do. meaning that if there is a compound noun with $N > 7$ we have to add a rule for that, as well.

Samary:

"Nominal → Nominal Nound" existence would save time and need of more rules, therefore it is not neccessary. it also can generate infinite length sentences, so it may cause problems.

3- To show how we can fill the (n+1) * (n+1) CYK parsing matrix, I start with an empty matrix, L1 in CNF (figure 13.3 of SLP3), and Lexicon table in figure 13.1 of SLP3 (as your hint in the Piazza suggested). In fact, we have to consider variables -> terminals rules to be able to answer this question.

$\mathcal{L}_1$ **in CNF**
S → NP VP
S → X1 VP
X1 → Aux NP
S → book | include | prefer
S → Verb NP
S → X2 PP
S → Verb PP
S → VP PP
NP → I | she | me
NP → TWA | Houston
NP → Det Nominal
Nominal → book | flight | meal | money
Nominal → Nominal Noun
Nominal → Nominal PP
VP → book | include | prefer
VP → Verb NP
VP → X2 PP
X2 → Verb NP
VP → Verb PP
VP → VP PP
PP → Preposition NP

Lexicon
Det → that | this | the | a
Noun → book | flight | meal | money
Verb → book | include | prefer
Pronoun → I | she | me
Proper-Noun → Houston | NWA
Aux → does
Preposition → from | to | on | near | through

Book	The	Flight	Through	Houston
[0,1]	[0,2]	[0,3]	[0.4]	[0,5]
	[1,2]	[1.3]	[1.4]	[1,5]
		[2,3]	[2,4]	[2,5]
			[3,4]	[3,5]
				[4,5]

Step 1, we have to complete the bottom of each column. Meaning that finding the Non-Terminal to Terminal (words) derivations for [0,1], [1,2], [2,3], [3,4], and [4,5] cells.

Cell [0,1] is “book” and by looking at the figures in the previous page, we will have:
Noun -> book, Verb -> book, VP -> book, Nominal -> book, and S -> book
So cell[0,1] = Noun, Verb, VP, Nominal, and S

The rest will be filled the same way and the results is listed in the table below:

Book	The	Flight	Through	Houston
Noun,Verb, VP, Nominal, S [0,1]	[0,2]	[0,3]	[0.4]	[0,5]
	Det [1,2]	[1.3]	[1.4]	[1,5]
		Noun, Nominal [2,3]	[2,4]	[2,5]
			Prep [3,4]	[3,5]
				NP, Proper-Noun [4,5]

Step 2:
Now we can start working on cell [3,5] and find derivation of Prep NP, or Prep Proper-Noun. But, because the question asks for cell [0,1] and [0,3] we can skip a bunch of steps and go to build cell[1,3].
To do that we have to find a Nonterminal on the left side of the figure in the previous page that results in Det Noun or Det Nominal as its right hand side. This would be NP as you can see in the table below.

Amin Hosseiny Marani

Book	The	Flight	Through	Houston
Noun,Verb, VP, Nominal, S [0,1]	[0,2]	[0,3]	[0.4]	[0,5]
	Det [1,2]	NP [1.3]	[1.4]	[1,5]
		Noun, Nominal [2,3]	[2,4]	[2,5]
			Prep [3,4]	[3,5]
				NP, Proper-Noun [4,5]

Step 3, cell[0,2]:
Since Det only appears in one rule (NP -> Det Nominal and not the other way around), there is no connection between cell [1,2] and [0,1]. So this makes cell [0,2] to remain empty as a fencepost.

Step 4, cell [0,3]:
Now we have to find any rule that has a right hand side of combinations cell[0,1] and [1,3] with . These are the ones we could use, S ->Verb NP, VP -> Verb NP, and X2 -> Verb NP. This makes the results for [0,3] as S, VP, and X2 as you can see below:

Book	The	Flight	Through	Houston
Noun,Verb, VP, Nominal, S [0,1]	[0,2]	S,VP, X2 [0,3]	[0.4]	[0,5]
	Det [1,2]	NP [1.3]	[1.4]	[1,5]
		Noun, Nominal [2,3]	[2,4]	[2,5]
			Prep [3,4]	[3,5]
				NP, Proper-Noun [4,5]

4-
To compute each cell we have to sum over all possible CFG parse trees with $P(N \ -> \ N_1 N_2) \ * \ \beta_{N_1}(1, p) * \beta_{N_2}(p + 1, q)$.

Step0:
To start we have to fill the diagonal cells as below. For that we are going to use the probability table below in the figure 4.1 and we have to put the probabilities as the table below. For example, for cell [5,5] : we only have NP -> stars with the probability of 0.18; so, we will put that in the table.
We have to put all possible parsing trees. For example, cell[2,2] has the word “saw”with two parsing trees : NP -> saw , and V -> saw. We will add both of them.

S → NP VP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → *astronomers*	0.1
VP → V NP	0.7	NP → *ears*	0.18
VP → VP PP	0.3	NP → *saw*	0.04
P → *with*	1.0	NP → *stars*	0.18
V → *saw*	1.0	NP → *telescopes*	0.1

Figure 4.1 - the probability of parsing trees

	1	2	3	4	5
1	$\beta_{NP} = 0.1$				
2		$\beta_{NP} = 0.04$ $\beta_V = 1.0$			
3			$\beta_{NP} = 0.18$		
4				$\beta_P = 1.0$	
5					$\beta_{NP} = 0.18$
	Astronomers	Saw	Stars	With	Ears

Step 1:
To compute cell[1,3] we have to compute cell [2,3] first. For that we have to look for possible parsing trees; 1) NP NP and 2) V NP. We only have VP -> V NP in the figure 4.1 so we will have

to compute $P(VP\ -{>}\ V\ NP)\ *\ \beta_V(2,2)\ *\ \beta_{NP}(3,3)\ =\ 0.7\ *\ 1.0\ *\ 0.18\ =\ 0.126$. We will update the table with this value as below:

	1	2	3	4	5
1	$\beta_{NP} =\ 0.1$				
2		$\beta_{NP} =\ 0.04$ $\beta_V =\ 1.0$	$\beta_{VP} =\ 0.126$		
3			$\beta_{NP} =\ 0.18$		
4				$\beta_P =\ 1.0$	
5					$\beta_{NP} =\ 0.18$
	Astronomers	Saw	Stars	With	Ears

Step 2:
Now we can finally compute cell[1,3] using cells [1,1] and [2,3]. We have to find parsing trees with the right hand side of NP VP or NP NP. Again we only have S -> NP VP with a value of 1.0 * 0.1 * 0.126 = 0.0126. So the table would be update as below:

	1	2	3	4	5
1	$\beta_{NP} =\ 0.1$		$\beta_S =\ 0.0126$		
2		$\beta_{NP} =\ 0.04$ $\beta_V =\ 1.0$	$\beta_{VP} =\ 0.126$		
3			$\beta_{NP} =\ 0.18$		
4				$\beta_P =\ 1.0$	
5					$\beta_{NP} =\ 0.18$
	Astronomers	Saw	Stars	With	Ears

*columns 4 and 5 were not completed because it is not necessary for this question.

5-
S= “telescopes saw stars”

S → NP VP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → *astronomers*	0.1
VP → V NP	0.7	NP → *ears*	0.18
VP → VP PP	0.3	NP → *saw*	0.04
P → *with*	1.0	NP → *stars*	0.18
V → *saw*	1.0	NP → *telescopes*	0.1

I used a similar table to the table in the question 4 to find possible trees.

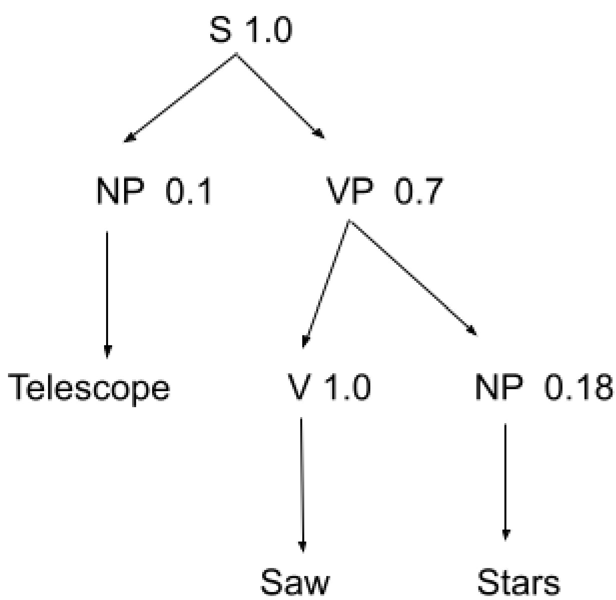
	1	2	3
1	$\beta_{NP} = 0.1$		
2		$\beta_{NP} = 0.04$ $\beta_{V} = 1.0$	
3			$\beta_{NP} = 0.18$
	telescopes	Saw	Stars

I continued and filled the other cells (this is pretty much similar to the table in question 4).

	1	2	3
1	$\beta_{NP} = 0.1$		$\beta_{S} = 0.0126$ Comes from (s -> NP VP) and 1.0 * 0.1*0.126
2		$\beta_{NP} = 0.04$ $\beta_{V} = 1.0$	$\beta_{VP} = 0.126$ Comes from (VP -> V NP) and 0.18*1.0 * 0.7

3			$\beta_{NP} = 0.18$
	telescopes	Saw	Stars

This will lead us to one tree:



P(T) = 1.0 * 0.1*0.7*1.0*0.18 = 0.0126