Assignment 2 - Training of Recurrent Perceptron

Sneha Oram, 23M2159, MS CMINDS-1
A Snegha, 23M2160, MS CMINDS-1
K Hemanth, 23M2164, MS CMINDS-1
A Shruthi, 23M1074, MTech EE-1

31st March 2024

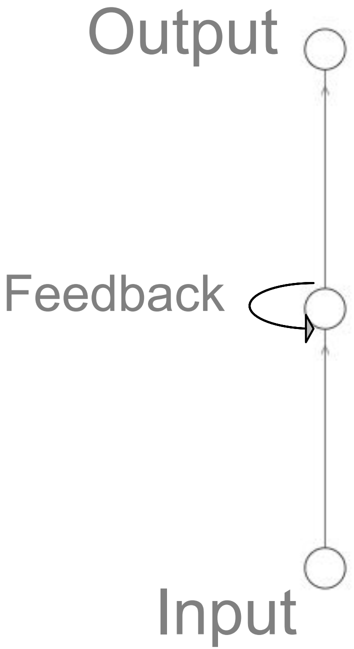
Problem Statement

Input: POS-tagged input tokens

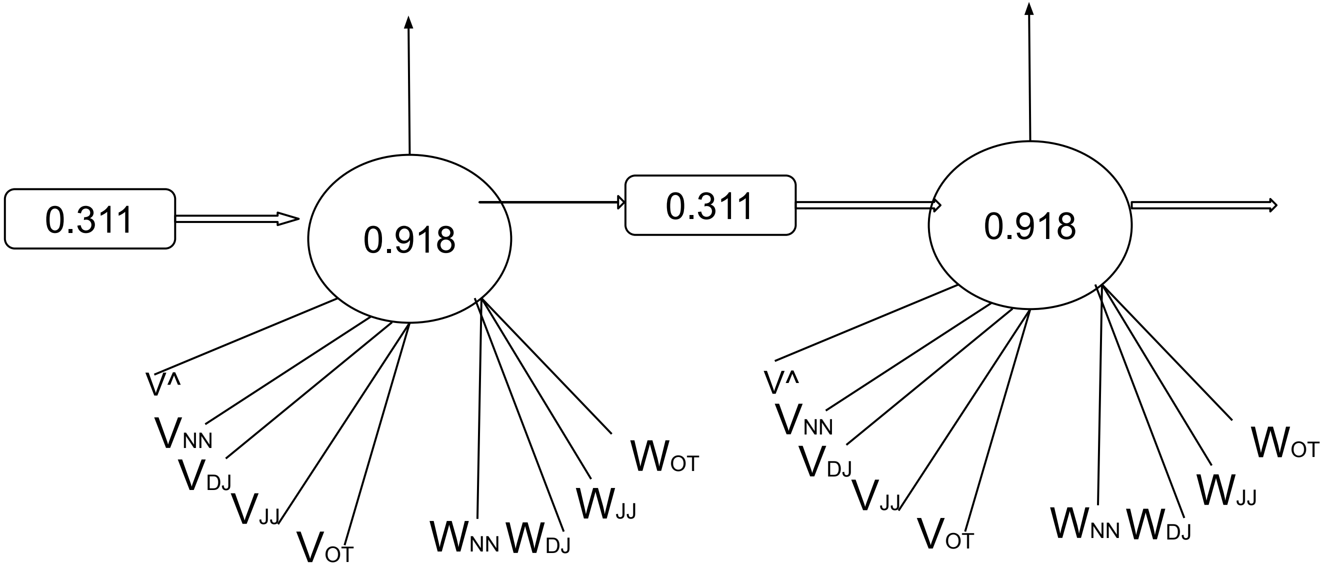
Output: Noun chunk labels on tokens .The beginning of the chunk will be labeled 1 and the rest of the words in the chunk will be labeled 0. All other words are labeled 1.

Implementation Details (1/3)

- **Loss Function:** Cross Entropy
- **Activation Function:** Sigmoid
- **Parameters:** W, W_feedback, bias
- **Epoch:** 5
- **Learning rate:** 0.001
- **Batch size:** 25



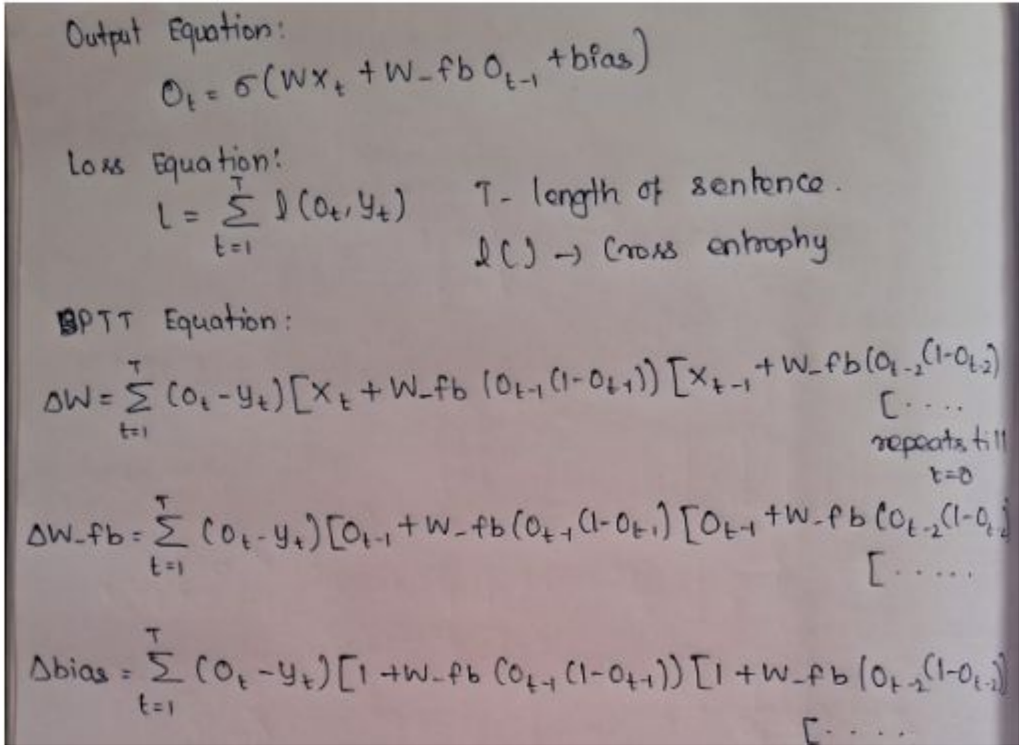
Implementation Details (2/3)



$V^\wedge$	0.644
V_{NN}	-0.83
V_{DT}	-1.569
V_{JJ}	-0.909
V_{OT}	1.677
W_{NN}	-1.928
W_{DJ}	-0.409
W_{JJ}	-0.790
W_{OT}	2.01

Implementation Details (3/3)

- Equations used in BPTT -



Output Equation:

$$O_t = \sigma(Wx_t + W\_fb\, O_{t-1} + bias)$$

Loss Equation:

$$L = \sum_{t=1}^{T} l(O_t, y_t)$$

T - length of sentence.

$l(\,)$ → Cross entrophy

BPTT Equation:

$$\Delta W = \sum_{t=1}^{T} (O_t - y_t)[x_t + W\_fb\,(O_{t-1}(1-O_{t-1}))[x_{t-1} + W\_fb(O_{t-2}(1-O_{t-2}) \; [\cdots$$

repeats till $t=0$

$$\Delta W\_fb = \sum_{t=1}^{T} (O_t - y_t)[O_{t-1} + W\_fb(O_{t-1}(1-O_{t-1})[O_{t-1} + W\_fb(O_{t-2}(1-O_{t-2}) \; [\cdots$$

$$\Delta bias = \sum_{t=1}^{T} (O_t - y_t)[1 + W\_fb\,(O_{t-1}(1-O_{t-1}))[1 + W\_fb(O_{t-2}(1-O_{t-2}) \; [\cdots$$

Overall performance

- Accuracy of Test Dataset: 84.12%
- Precision: 0.8097, Recall: 0.9888, F1 Score: 0.8903
- 5-fold cross validation

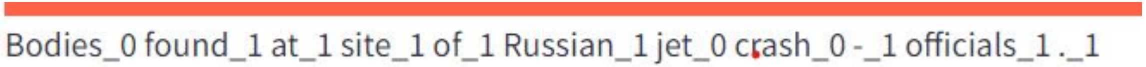
K-fold	Accuracy	Precision	Recall	F1 score
1/5	**81.8**	**0.788**	**0.997**	**0.881**
2/5	81.3	0.784	0.997	0.877
3/5	81.3	0.785	0.996	0.878
4/5	81.1	0.782	0.996	0.876
5/5	81.2	0.782	0.996	0.876

Language constraint table

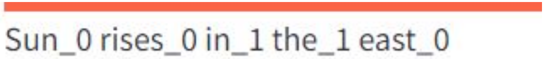
Current (W) Prev (V)	DT	JJ	NN	OT
^	$V_\wedge + W_{DT} > \theta$	$V_\wedge + W_{JJ} > \theta$	$V_\wedge + W_{NN} > \theta$	$V_\wedge + W_{OT} > \theta$
DT		$W + V_{DT} + W_{JJ} < \theta$	$W + V_{DT} + W_{NN} < \theta$	
JJ		$V_{JJ} + W_{JJ} < \theta$ $W + V_{JJ} + W_{JJ} < \theta$	$V_{JJ} + W_{NN} < \theta$ $W + V_{JJ} + W_{NN} < \theta$	
NN				$W + V_{NN} + W_{OT} > \theta$
OT	$W+V_{OT}+W_{DT} > \theta$	$W+V_{OT}+W_{JJ} > \theta$	$W+V_{OT}+W_{NN} > \theta$	$W+V_{OT}+W_{OT} > \theta$

$V_\wedge$	0.644
V_{NN}	-0.83
V_{DT}	-1.569
V_{JJ}	-0.909
V_{OT}	1.677
W_{NN}	-1.928
W_{DJ}	-0.409
W_{JJ}	-0.790
W_{OT}	2.01
w	0.311

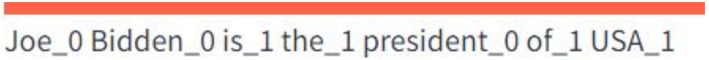
Error Analysis (1/2)



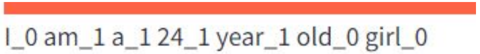
Bodies_0 found_1 at_1 site_1 of_1 Russian_1 jet_0 crash_0 -_1 officials_1 ._1



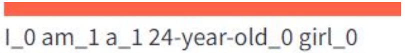
Sun_0 rises_0 in_1 the_1 east_0



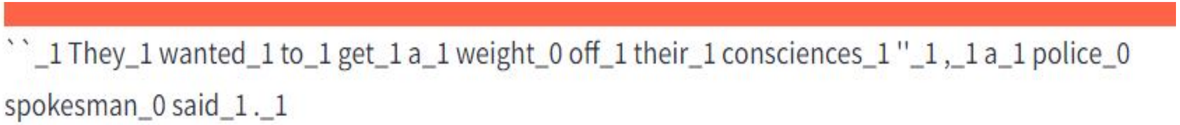
Joe_0 Bidden_0 is_1 the_1 president_0 of_1 USA_1



I_0 am_1 a_1 24_1 year_1 old_0 girl_0



I_0 am_1 a_1 24-year-old_0 girl_0



``_1 They_1 wanted_1 to_1 get_1 a_1 weight_0 off_1 their_1 consciences_1 ''_1 ,_1 a_1 police_0 spokesman_0 said_1 ._1

Error Analysis (2/2)

- Model is making mistake if the sentence starts with just ‘NN’ or ‘NN NN’ tag.
- Model is making mistake with numbers but when grouped along with other word, it is chunking correctly.
- Punctuations are considered as none Noun chunks.
- Maximum sentence length without any mistake is 67.

Learnings

- A single recurrent perceptron is able to detect noun chunks in a text successfully using bigram assumption.
- The algorithm of BPTT in single perceptron is handling dependencies upto 67 words.