CS772: Deep Learning for Natural Language Processing (DL-NLP)

Introduction cntd, flavour of neural computation, perceptron

Pushpak Bhattacharyya

Computer Science and Engineering Department

IIT Bombay

Week 1 of 2nd Jan, 2023

Course Content: Task *vs.* Technique Matrix

Task (row) vs. Technique (col) Matrix	**Rules Based/Knowledge-Based**	**Classical ML**				**Deep Learning**		
		Perceptron	Logistic Regression	SVM	Graphical Models (HMM, MEMM, CRF)	Dense FF with BP and softmax	RNN-LSTM	CNN
Morphology								
POS								
Chunking								
Parsing								
NER, MWE								
Coref								
WSD								
Machine Translation								
Semantic Role Labeling								
Sentiment								
Question Answering								

Books

- 1. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016.

- 2. Dan Jurafsky and James Martin, Speech and Language Processing, 3rd Edition, 2019.

Books (2/2)

- 4. Christopher Manning and Heinrich Schutze, Foundations of Statistical NaturalLanguage Processing, MIT Press, 1999.

- 5. Pushpak Bhattacharyya, Machine Translation, CRC Press, 2017.

Journals and Conferences

- Journals: Computational Linguistics, Natural Language Engineering, Journal of Machine Learning Research (JMLR), Neural Computation, IEEE Transactions on Neural Networks

- Conferences: ACL, EMNLP, NAACL, EACL, AACL, NeuriPS, ICML

Useful NLP, ML, DL libraries

- NLTK
- Scikit-Learn
- Pytorch
- Tensorflow (Keras)
- **Huggingface**
- Spacy
- Stanford Core NLP

Nature of DL-NLP

3 Generations of NLP

- Rule based NLP is also called Model Driven NLP
- Statistical ML based NLP (*Hidden Markov Model, Support Vector Machine*)
- Neural (Deep Learning) based NLP

Illustration with POS tagging

Neural Parsing

Data

[

[The man]$_{NP}$

[

[

saw$_{VBD}$

[[the boy]$_{NP}$

]$_{VP}$

[with [a telescope]$_{NP}$]$_{PP}$

]$_{VP}$

]$_{S}$

Classification Decisions

- Are there any brackets to be inserted at a position *p*?
- If the answer to (a) is yes, which bracket- opening or closing?
- If closing bracket, which label to insert

Steps (1/2)

- In the first pass, the representation from two consecutive word-units is obtained by (a) concatenating the vectors of these words, and (b) passing the concatenation through the recurrent n/w.
- The resulting combination-unit is (a) pre-multiplied by a *learnt* weight vector, (b) the product added with a bias term, (c) the result passed through a non-linear function, to obtain a score for the unit.

Steps (2/2)

- The highest scoring combination-unit is retained and a new sequence obtained by deleting the word-units constituting the combination-unit.
- The new sequence is treated like in the previous pass, combining bi-grams.
- Retained combination-units also pass through a feedforward network with softmax final layer, to obtain the labels *NP, VP, PP* etc.
- The process stops with the finding of the start symbol *S*.

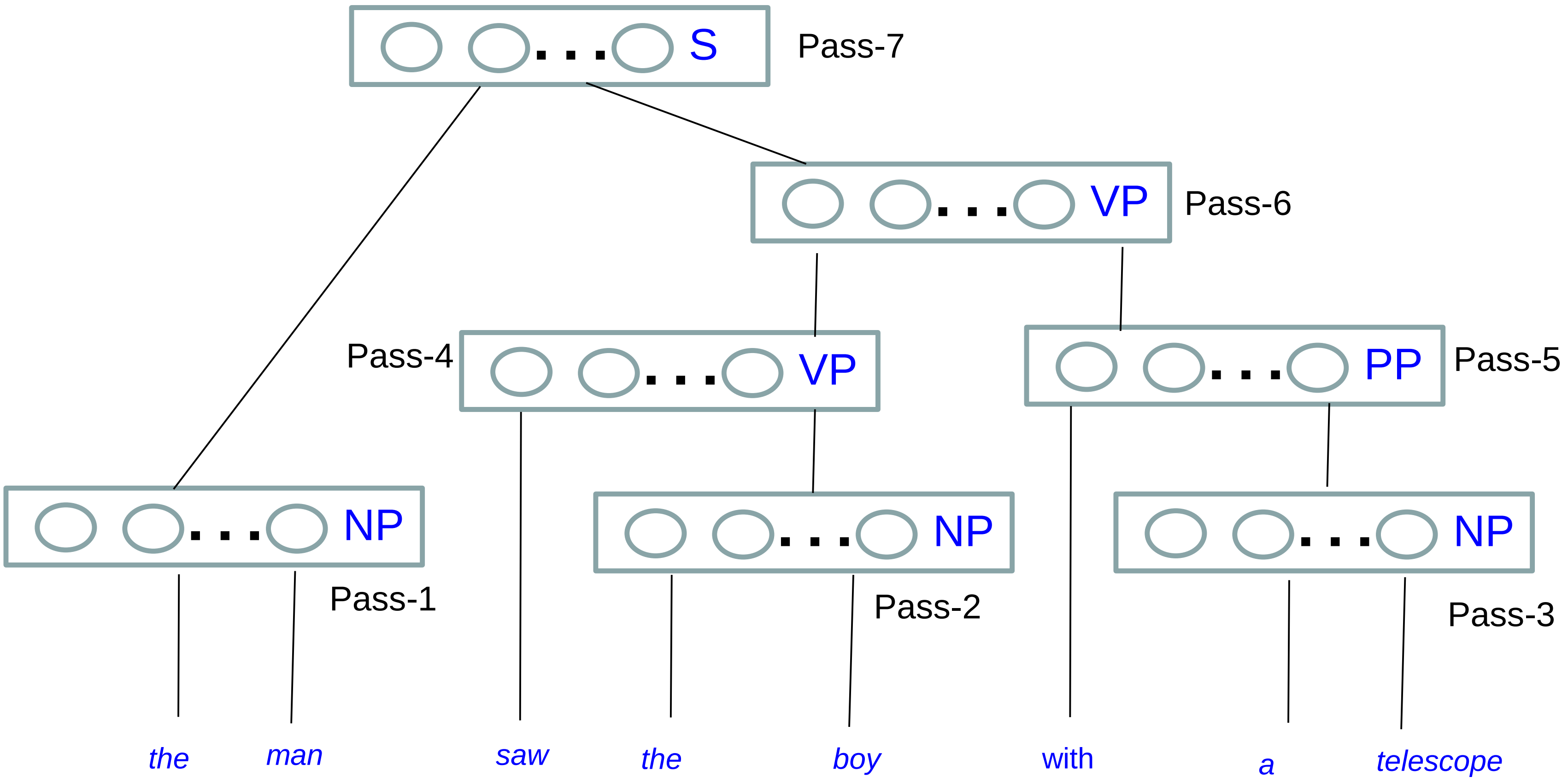
Example (1/2)

- $_0$ *the* $_1$ *man* $_2$ *saw* $_3$ *the* $_4$ *boy* $_5$ *with* $_6$ *a* $_7$ *telescope* $_8$
- $_0\, C^1_{02}\ _1\, C^1_{13}\ _2\, C^1_{24}\ _3\, C^1_{35}\ _4\, C^1_{46}\ _5\, C^1_{57}\ _6\, C^1_{68}\ _7$; assume C^1_{02} (*'the man'*) has the highest score; the upper right suffix '1' indicates pass-1; '*the man*' is replaced with its representation C^1_{02} along with the label *NP*
- $_0\, C^1_{02}$_*NP* $_1$ *saw* $_2$ *the* $_3$ *boy* $_4$ *with* $_5$ *a* $_6$ *telescope* $_7$; new sequence
- (after combining, scoring and filtering) $_0\, C^1_{02}$_*NP* $_1$ *saw* $_2$ C^2_{24}_*NP* $_3$ *with* $_4$ *a* $_5$ *telescope* $_6$; upper right suffix '2' indicates pass-2

Example (2/2)

- $_0 C^1{}_{02}$_*NP* $_1$ *saw* $_2 C^2{}_{24}$_*NP* $_3$ *with* $_4 C^3{}_{46}$_*NP* $_5$; 3rd pass; '*a telescope*' is an *NP*
- $_0 C^1{}_{02}$_*NP* $_1 C^4{}_{13}$_*VP* $_2$ *with* $_4 C^3{}_{46}$_*NP* $_5$; 4th pass; '*saw*' and *NP* ('*a boy*') give rise to a *VP*
- $_0 C^1{}_{02}$_*NP* $_1 C^4{}_{13}$_*VP* $_2 C^5{}_{25}$_*PP* $_3$; 5th pass; '*with*' and NP ('*a telescope*') produce a VP
- $_0 C^1{}_{02}$_*NP* $_1 C^6{}_{13}$_*VP* $_2$; 6th pass; *VP ('saw the boy')* + *PP ('with a telescope')* → *VP*
- $_0 C^7{}_{02}$_*S*; 7th pass; *S* →*NP VP*; *S* found; TERMINATE

RcNN based parse tree of "*the man…*": Parse Tree-1 (man has telescope)



Neural parsing objective function

$$J = \sum_i \left[s(x_i, y_i) - \max_{y \in A(x_i)} \left(s(x_i, y) + \Delta(y, y_i) \right) \right]$$

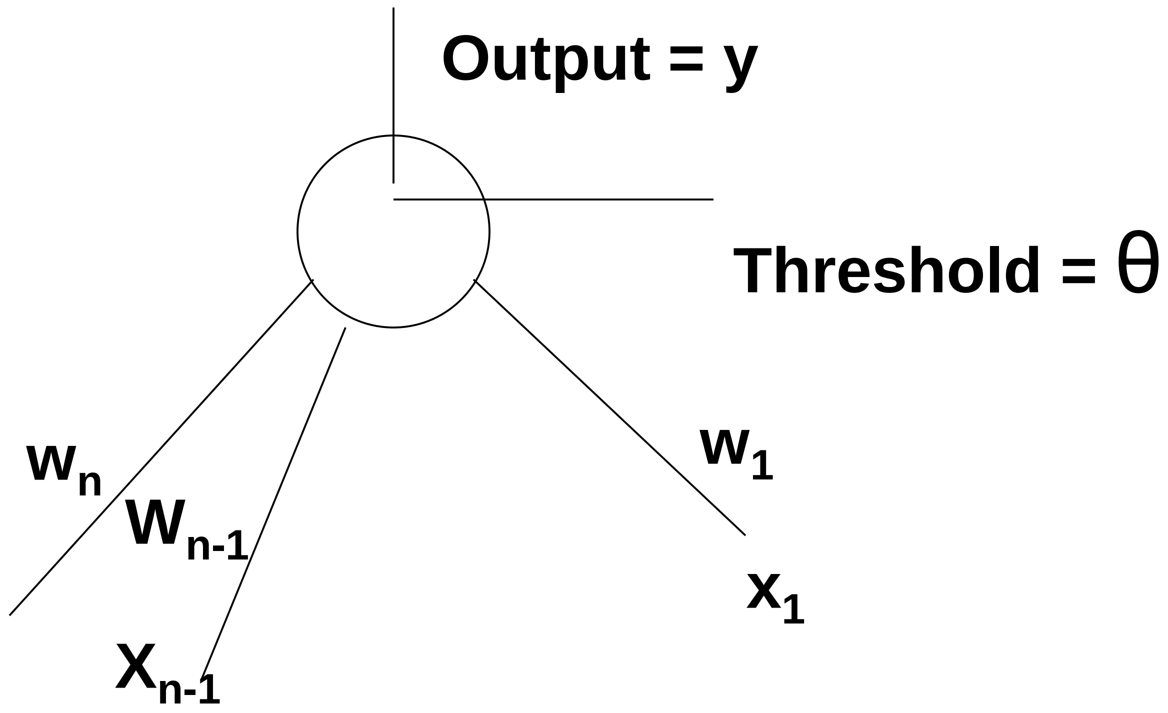
$$s(x_i, y_i) = \sum_{d \in T(y_i)} s_d(c_p, c_q)$$

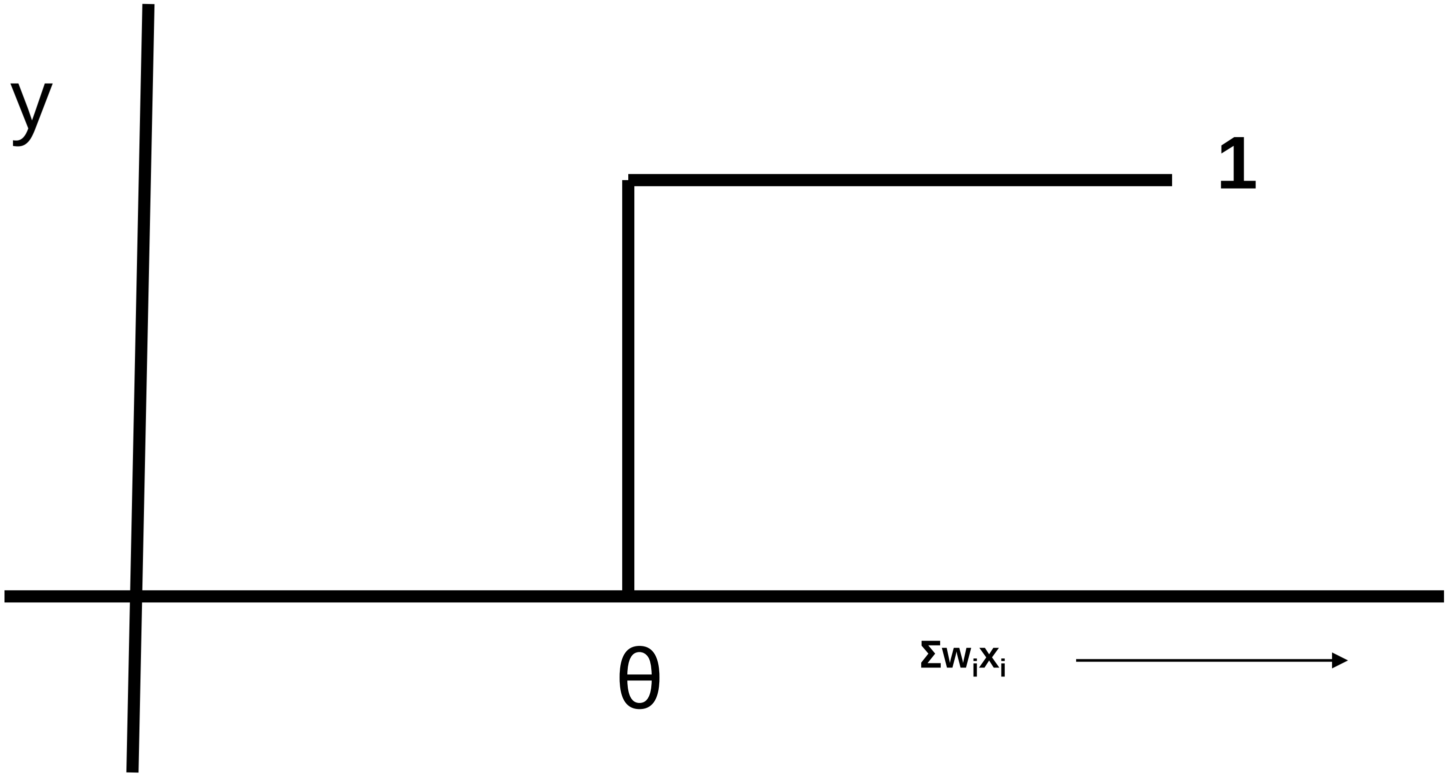
RcNN→RNN→FFNN→Perceptron

The Perceptron

The Perceptron Model

- A perceptron is a computing element with input lines having associated weights and the cell having a threshold value. The perceptron model is motivated by the biological neuron.





- Step function / Threshold function
- y = 1 for Σwixi >=θ
- =0 otherwise

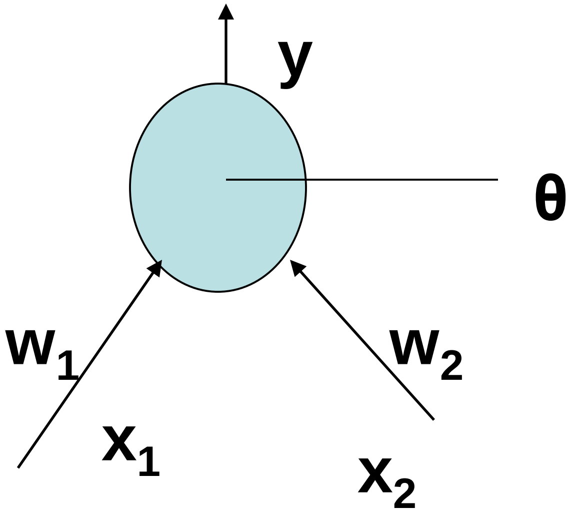
Features of Perceptron

- Input output behavior is discontinuous and the derivative does not exist at $\Sigma w_i x_i = \theta$
- $\Sigma w_i x_i - \theta$ is the net input denoted as net
- Referred to as a linear threshold element - linearity because of x appearing with power 1
- y= f(net): Relation between y and net is non-linear

Computation of Boolean functions: AND

X1	x2	**y**
0	0	0
0	1	0
1	0	0
1	1	1

The parameter values (weights &thresholds) need to be found.



Computing parameter values

- $w1 * 0 + w2 * 0 \leq \theta \rightarrow \theta \geq 0$; since $y=0$
- $w1 * 0 + w2 * 1 \leq \theta \rightarrow w2 \leq \theta$; since $y=0$
- $w1 * 1 + w2 * 0 \leq \theta \rightarrow w1 \leq \theta$; since $y=0$
- $w1 * 1 + w2 * 1 > \theta \rightarrow w1 + w2 > \theta$; since $y=1$
- $w1 = w2 = \quad = 0.5$
- satisfy these inequalities and find parameters to be used for computing AND function.

Other Boolean functions

- OR can be computed using values of w1 = w2 = 1 and = 0.5
- XOR function gives rise to the following inequalities:

$w1 * 0 + w2 * 0 <= \theta \Rightarrow \theta >= 0$

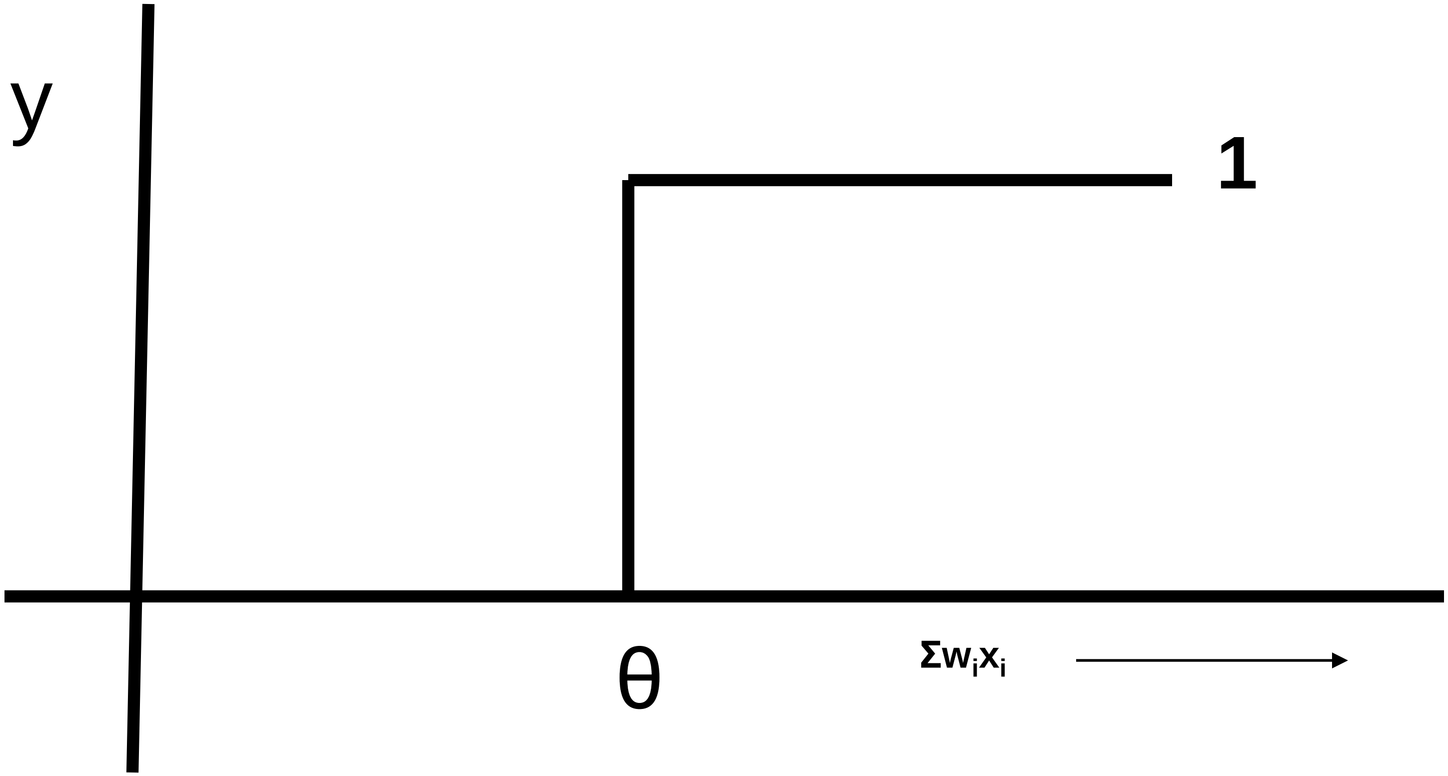
$w1 * 0 + w2 * 1 > \theta \Rightarrow w2 > \theta$

$w1 * 1 + w2 * 0 > \theta \Rightarrow w1 > \theta$

$w1 * 1 + w2 * 1 <= \theta \Rightarrow w1 + w2 <= \theta$

Threshold functions

- n   # Boolean functions (2^2^n) #Threshold Functions (2n2)
- 1   4   4
- 2   16   14
- 3   256   128
- 4   64K   1008
- Functions computable by perceptrons- threshold functions,
- #TF becomes negligibly small for larger values of #BF.
- For n=2, all functions except XOR and XNOR are



- Step function / Threshold function
- y = 1 for $\Sigma w_i x_i \geq \theta$
- =0 otherwise

Features of Perceptron

- Input output behavior is discontinuous and the derivative does not exist at $\Sigma wixi = \theta$
- $\Sigma_{1,n} w_i x_i - \theta$ is the net input denoted as net

- Referred to as a linear threshold element - linearity because of x appearing with power 1
- y= f(net): Relation between y and net is non-linear

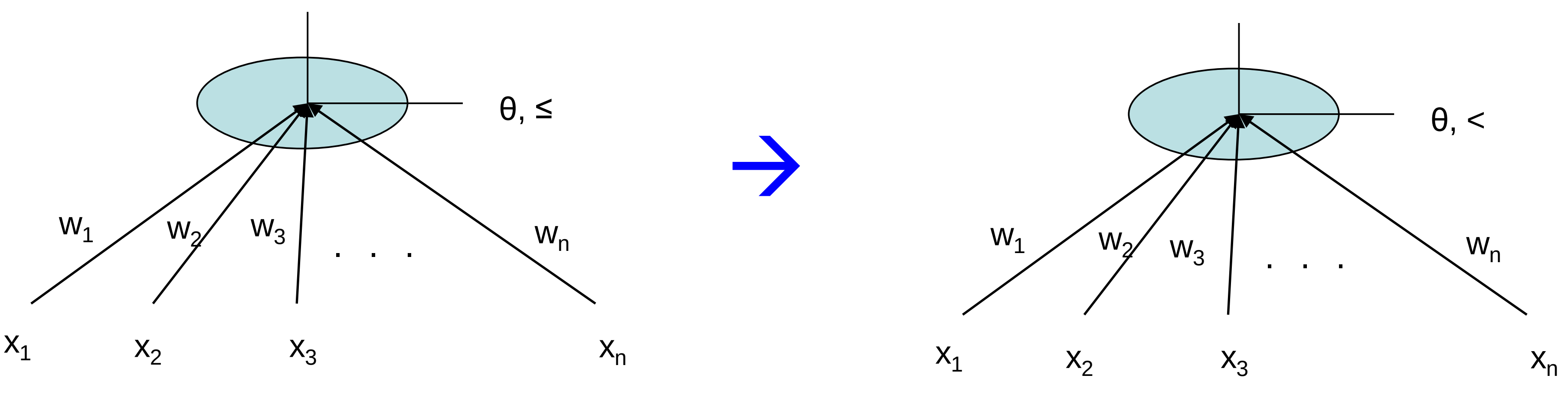
Perceptron Training Algorithm (PTA)

Preprocessing:

1. The computation law is modified to

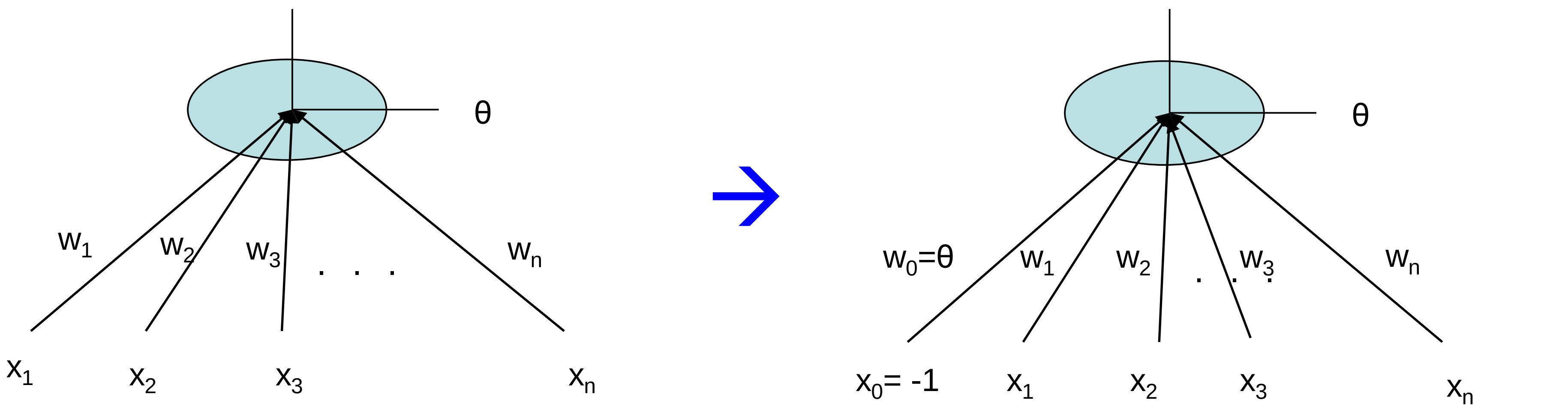
$$y = 1 \text{ if } \sum w_i x_i > \theta$$

$$y = o \text{ if } \sum w_i x_i < \theta$$



PTA – preprocessing cont…

2. Absorb θ as a weight



3. Negate all the zero-class examples

Example to demonstrate preprocessing

- **OR perceptron**

1-class <1,1> , <1,0> , <0,1>

0-class <0,0>

Augmented x vectors:-

1-class <-1,1,1> , <-1,1,0> , <-1,0,1>

0-class <-1,0,0>

Negate 0-class:- <1,0,0>

Example to demonstrate preprocessing cont..

Now the vectors are

	x_0	x_1	x_2
X_1	-1	0	1
X_2	-1	1	0
X_3	-1	1	1
X_4	1	0	0

Perceptron Training Algorithm

1. Start with a random value of w

   ex: <0,0,0…>

2. Test for $wx_i > 0$

   If the test succeeds for i=1,2,…n

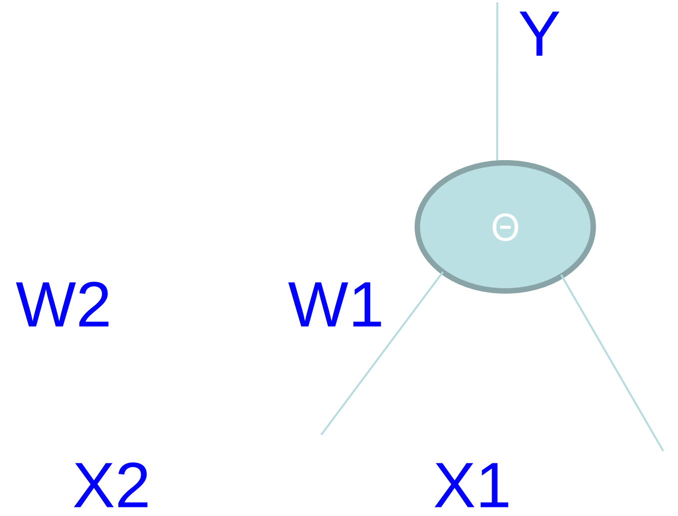
   then return w

3. Modify w, $w_{next} = w_{prev} + x_{fail}$

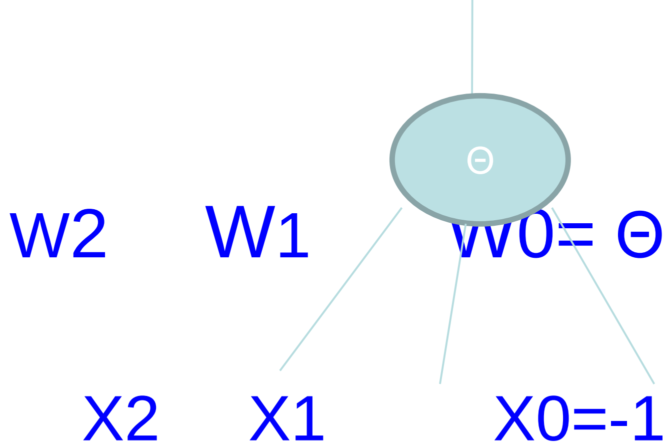
PTA on NAND

NAND:

X2	X1	Y
0	0	1
0	1	1
1	0	1
1	1	0



Converted To



Preprocessing

NAND Augmented:

X2	X1	X0	Y
0	0	-1	1
0	1	-1	1
1	0	-1	1
1	1	-1	0

NAND-0 class Negated

	X2	X1	X0
V0:	0	0	-1
V1:	0	1	-1
V2:	1	0	-1
V3:	-1	-1	1

Vectors for which W=<W2 W1 W0> has to be found such that $W. V_i > 0$

PTA Algo steps

Algorithm:

1. Initialize and Keep adding the failed vectors until W. Vi > 0 is true.

Step 0: $W = <0, 0, 0>$

$W_1 = <0, 0, 0> + <0, 0, -1>$ {V_0 Fails}

$= <0, 0, -1>$

$W_2 = <0, 0, -1> + <-1, -1, 1>$ {V_3 Fails}

$= <-1, -1, 0>$

$W_3 = <-1, -1, 0> + <0, 0, -1>$ {V_0 Fails}

$= <-1, -1, -1>$

$W_4 = <-1, -1, -1> + <0, 1, -1>$ {V_1 Fails}

$= <-1, 0, -2>$

Trying convergence

$W_5 = \langle -1, 0, -2 \rangle + \langle -1, -1, 1 \rangle$ {V_3 Fails}

$= \langle -2, -1, -1 \rangle$

$W_6 = \langle -2, -1, -1 \rangle + \langle 0, 1, -1 \rangle$ {V_1 Fails}

$= \langle -2, 0, -2 \rangle$

$W_7 = \langle -2, 0, -2 \rangle + \langle 1, 0, -1 \rangle$ {V_0 Fails}

$= \langle -1, 0, -3 \rangle$

$W_8 = \langle -1, 0, -3 \rangle + \langle -1, -1, 1 \rangle$ {V_3 Fails}

$= \langle -2, -1, -2 \rangle$

$W_9 = \langle -2, -1, -2 \rangle + \langle 1, 0, -1 \rangle$ {V_2 Fails}

$= \langle -1, -1, -3 \rangle$

Trying convergence

$W_{10} = \langle -1, -1, -3 \rangle + \langle -1, -1, 1 \rangle$ {V_3 Fails}

$= \langle -2, -2, -2 \rangle$

$W_{11} = \langle -2, -2, -2 \rangle + \langle 0, 1, -1 \rangle$ {V_1 Fails}

$= \langle -2, -1, -3 \rangle$

$W_{12} = \langle -2, -1, -3 \rangle + \langle -1, -1, 1 \rangle$ {V_3 Fails}

$= \langle -3, -2, -2 \rangle$

$W_{13} = \langle -3, -2, -2 \rangle + \langle 0, 1, -1 \rangle$ {V_1 Fails}

$= \langle -3, -1, -3 \rangle$

$W_{14} = \langle -3, -1, -3 \rangle + \langle 0, 1, -1 \rangle$ {V_2 Fails}

$= \langle -2, -1, -4 \rangle$

W15 = <-2, -1, -4> + <-1, -1, 1> {V3 Fails}

= <-3, -2, -3>

W16 = <-3, -2, -3> + <1, 0, -1> {V2 Fails}

= <-2, -2, -4>

W17 = <-2, -2, -4> + <-1, -1, 1> {V3 Fails}

= <-3, -3, -3>

W18 = <-3, -3, -3> + <0, 1, -1> {V1 Fails}

= <-3, -2, -4>

W2 = -3, W1 = -2, W0 = Θ = -4

Succeeds for all vectors

PTA convergence

Statement of Convergence of PTA

- Statement:

  Whatever be the initial choice of weights and whatever be the vector chosen for testing, PTA converges if the vectors are from a linearly separable function.

Proof of Convergence of PTA

- Suppose w_n is the weight vector at the n^{th} step of the algorithm.
- At the beginning, the weight vector is w_0
- Go from w_i to w_{i+1} when a vector X_j fails the test $w_i X_j > 0$ and update w_i as

  $$w_{i+1} = w_i + X_j$$

- Since Xjs form a linearly separable function,
- there exits w* s.t. $w^* X_j > 0$ for all j

Proof of Convergence of PTA (cntd.)

- Consider the expression

$$G(w_n) = \frac{w_n \cdot w^*}{|w_n|}$$

where w_n = weight at nth iteration

- $$G(w_n) = \frac{|w_n| \cdot |w^*| \cdot \cos\theta}{|w_n|}$$

where □ = angle between w_n and w^*

- $G(w_n) = |w^*| \cdot \cos\theta$
- $G(w_n) \leq |w^*|$ (as $-1 \leq \cos\theta \leq 1$)

Behavior of Numerator of G

$w_n \,.\, w^* \;=\; (w_{n-1} + X^{n-1}_{fail}) \,.\, w^*$

$= \; w_{n-1} \,.\, w^* + X^{n-1}_{fail} \,.\, w^*$

$= \; (w_{n-2} + X^{n-2}_{fail}) \,.\, w^* + X^{n-1}_{fail} \,.\, w^* \; \ldots..$

$= \; w_0 \,.\, w^* + (X^0_{fail} + X^1_{fail} + \ldots. + X^{n-1}_{fail}) .\, w^*$

$w^*.X^i_{fail}$ is always positive: note carefully

- Suppose $|X_j| \geq \delta_{min}$, where δ_{min} is the minimum magnitude.
- Num of G $\geq |w_0 \,.\, w^*| + n\, \delta_{min}|w^*|$
- So, numerator of G grows with n.

Behavior of Denominator of G

- $|w_n| = (w_n \cdot w_n)^{1/2}$

$= [(w_{n-1} + X^{n-1}_{fail})^2]^{1/2}$

$= [(w_{n-1})^2 + 2.\ w_{n-1.}\ X^{n-1}_{fail} + (X^{n-1}_{fail})^2]^{1/2}$

$\leq [(w_{n-1})^2 + (X^{n-1}_{fail})^2]^{1/2}$ (as $w_{n-1.}\ X^{n-1}_{fail} \leq 0$)

$\leq [(w_0)^2 + (X^0_{fail})^2 + (X^1_{fail})^2 + \ldots. + (X^{n-1}_{fail})^2]^{1/2}$

- $|X_j| \leq \delta_{max}$ (max magnitude)
- So, Denom $\leq [(w_0)^2 + n\ \delta_{max}{}^2)]^{1/2}$
- Denom grows as $n^{1/2}$

Some Observations

- Numerator of G grows as n
- Denominator of G grows as $n^{1/2}$

  => Numerator grows faster than denominator
- If PTA does not terminate, $G(w_n)$ values will become unbounded.

Some Observations contd.

- But, as $|G(w_n)| \leq |w^*|$ which is finite, this is impossible!
- Hence, PTA has to converge.
- Proof is due to Marvin Minsky.

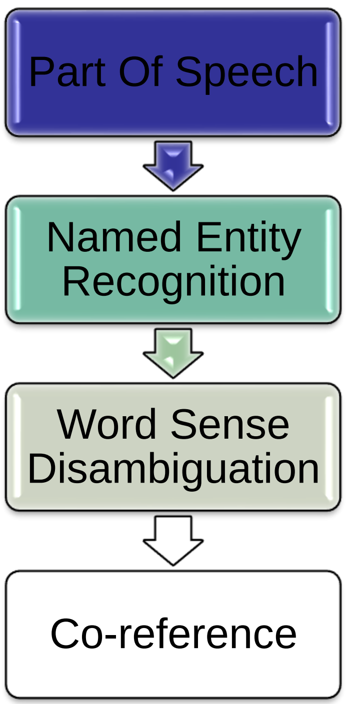
Convergence of PTA proved

- *Whatever be the initial choice of weights and whatever be the vector chosen for testing, PTA converges if the vectors are from a linearly separable function.*

Possible project ideas

Semantics Extraction using Universal Networking Language

Sentence: *I went with my friend, John, to the bank to withdraw some money but was disappointed to find it closed.*



Current work:

Combine Machine learning with rule Based technique
(Janardhan)

Agt(go,I)
Ptn(go,friend)
Nam(friend,John)
Plt(go,bank)
Pur(go, withdraw)
Obj(withdraw,money0
Mod(money,some)
And(go,disappoint)

Sentiment Analysis

“The water is boiling.”: Objective

“He is boiling with anger.”: Negative

Current work:

1. *Tweet and Blog Sentiment*
2. *Indian Language Sentiment Analysis*
3. *Word Sense and Sentiment*
4. *Thwarting and*

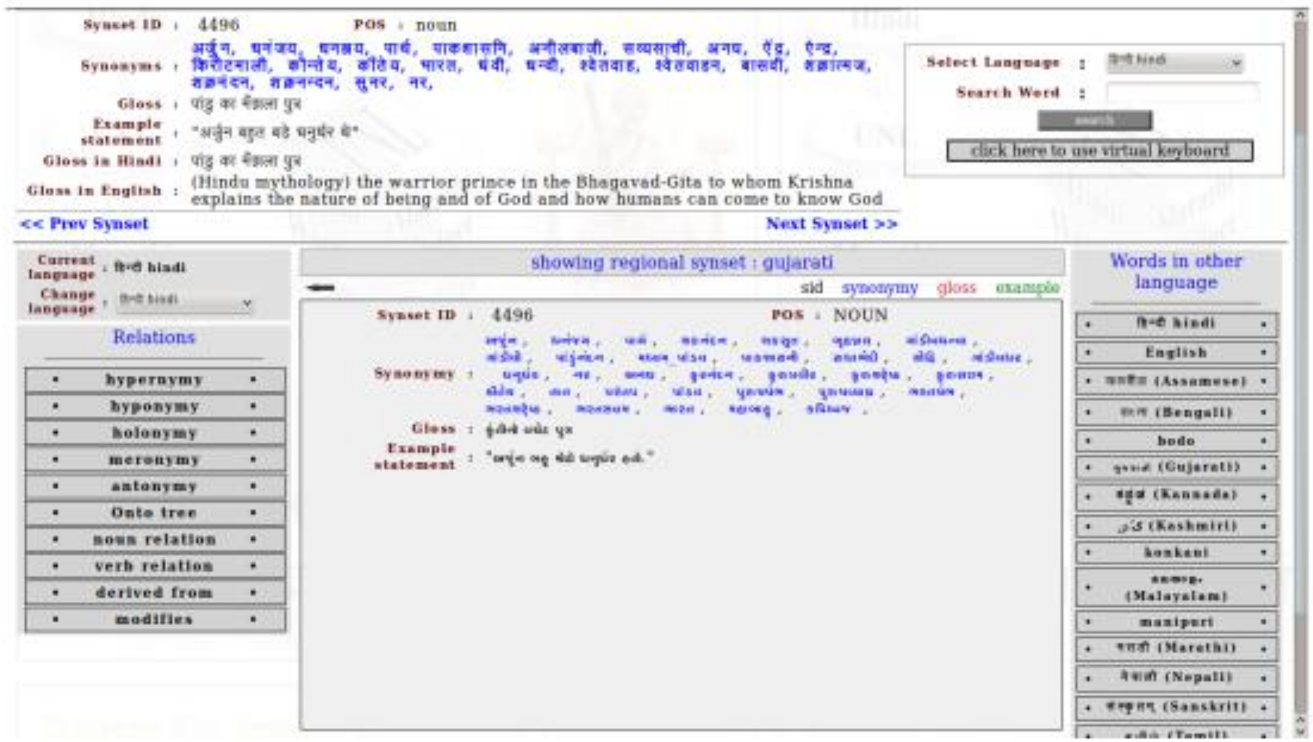
   (Subhabrata and Akshat, Balamurali)

Text Entailment

	TEXT	HYPOTHESIS	ENTAIL-MENT
1	*. The Hubble is the only large visible light and ultra-violet space telescope we have in operation.*	*Hubble is a Space telescope.*	True
2	*Google files for its long awaited IPO.*	*Google goes public.*	True
3	*After the deal closes, Teva will earn about $7 billion a year, the company said.*	*Teva earns $7 billion a year.*	False

Current work: Do entailment from Semantic Graphs (Arindam, Janradhan)

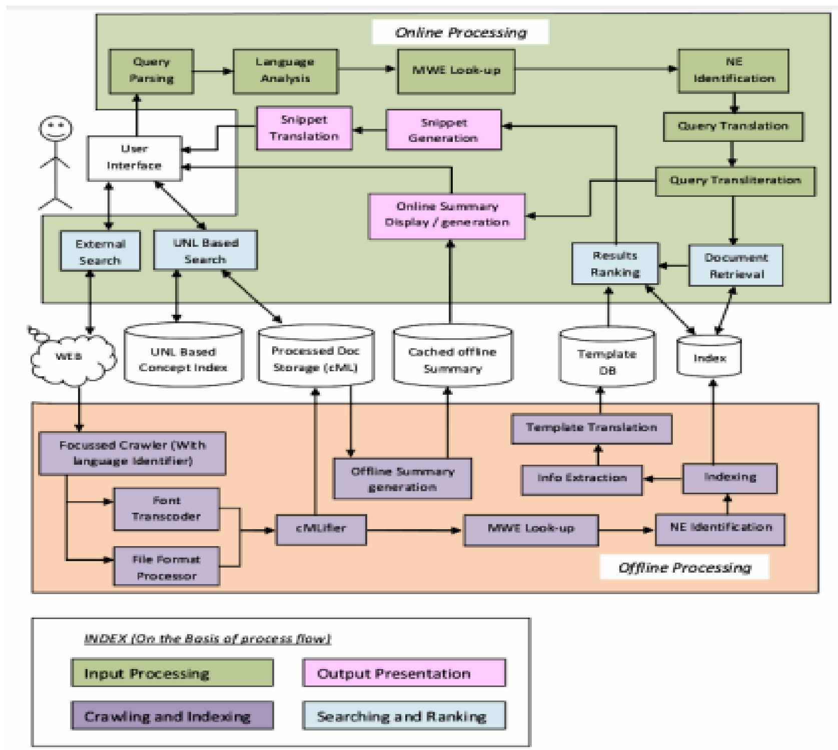
Indowordnet and Multilingual Word Sense Disambiguation



Current work: Linking wordnets with SUMO Ontology; using resources of one Language for another for WSD (Salil Joshi, Arindam Chatterjee, Brijesh, Mitesh)

Cross Lingual Information Retrieval

Architecture of Sandhan



Current work: Performance Enhancement; Query expansion and disambiguation
(Yogesh, Arjun, Swapnil)

Machine Translation

Large Projects funded by

Yahoo, Xerox, Ministry of IT

Current work:

1. *Indian Language to Indian Language*
2. *Statistical MT*
3. *Crowdsourcing and MT*
4. *Semantics and SMT*

(Mitesh, Anoop, Victor, Somya, Abhijit, Raj, Rahul)

Sites:

http://www,cse.iitb.ac.in/~pb
http://www.cfilt.iitb.ac.in