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

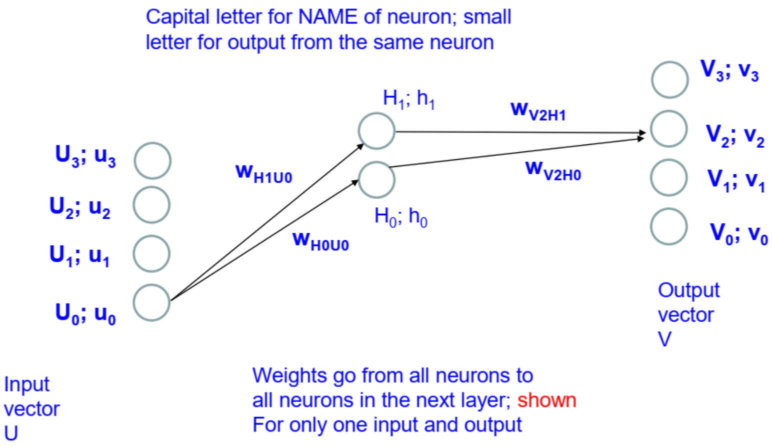
RNN, Sequence Processing, Representation Learning, Some BP points

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Week 7 of 12feb24

1-slide recap

- Derivation of weight change rule for skip gram



$$net_{V_0} = W_{U_0}.W_{V_0}^T$$

$$net_{V_1} = W_{U_0}.W_{V_1}^T$$

$$net_{V_2} = W_{U_0}.W_{V_2}^T$$

$$net_{V_3} = W_{U_0}.W_{V_3}^T$$

$$\Delta w_{V_2H_0} = \eta(1-v_2).w_{H_0U_0} = \eta(1-v_2)o_{H_0}$$

$$\Delta w_{H_0U_0}$$
$$= \eta[(1-v_2)w_{V_2H_0} + (0-v_0)w_{V_0H_0} + (0-v_1)w_{V_1H_0} + (0-v_3)w_{V_3H_0}].u_0$$

Sequence Processing

- Sequence Labelling
  - **Single label**: e.g., sentiment classification
  - **Multiple labels**
    - At each position, e.g., POS tagging Part of Speech.
    - At select positions, e.g., Named Entity marking (NP, VB, AD, etc)
- Sequence mapping
  - **Within the same modality**
    - Within the same language, e.g., summarization, question answering
    - Bilingual, e.g., Machine Translation
  - **Multiple modality**, e.g, visual question answering, speech to text, text to speech, image and video captioning, image and/or video generation from text and/or speech, video narration

Sequence Labelling Tasks

Named Entity Recognition

```
[PERS Pierre Vinken] , 61 years old , will join
[ORG IBM] 's board as a nonexecutive director
[DATE Nov. 2] .
```

Shallow Parsing

```
[NP Pierre Vinken] , [NP 61 years] old , [VP will join]
[NP IBM] 's [NP board] [PP as] [NP a nonexecutive
director] [NP Nov. 2] .
```

NP Chunking

```
[NP Pierre Vinken] , [NP 61 years] old , will join
[NP IBM] 's [NP board] as [NP a nonexecutive director]
[NP Nov. 2] .
```

https://courses.engr.illinois.edu/cs447/fa2018/Slides/Lecture07.pdf

The BIO encoding

We define three new tags:

– **B-NP**: beginning of a noun phrase chunk
– **I-NP**: inside of a noun phrase chunk
– **O**: outside of a noun phrase chunk

```
[NP Pierre Vinken] , [NP 61 years] old , will join
[NP IBM] ‘s [NP board] as [NP a nonexecutive director]
[NP Nov. 2] .
```

```
Pierre_B-NP Vinken_I-NP ,_O 61_B-NP years_I-NP
old_O ,_O will_O join_O IBM_B-NP ‘s_O board_B-NP as_O
a_B-NP nonexecutive_I-NP director_I-NP Nov._B-NP
29_I-NP ._O
```

https://courses.engr.illinois.edu/cs447/fa2018/Slides/Lecture07.pdf

Recurrent Neural Network

Acknowledgement:

1. http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

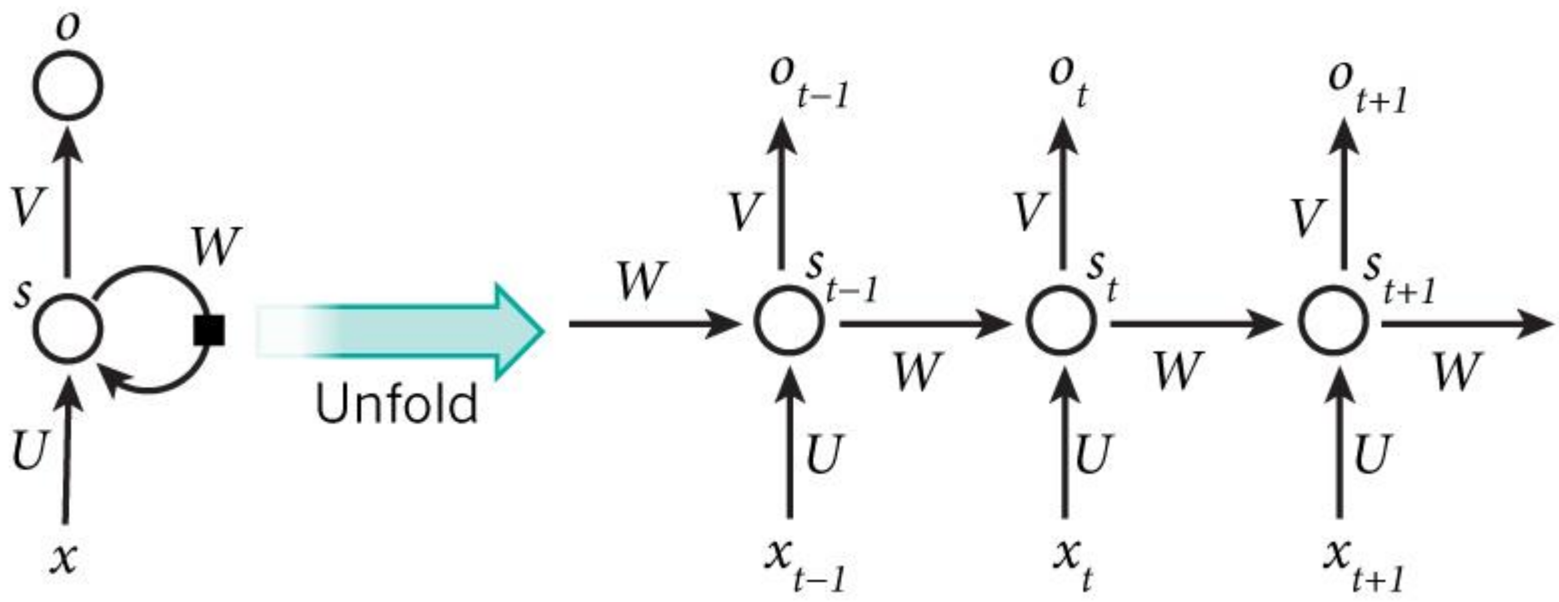
   By Denny Britz

2. Introduction to RNN by Jeffrey Hinton

http://www.cs.toronto.edu/~hinton/csc2535/lectures.html

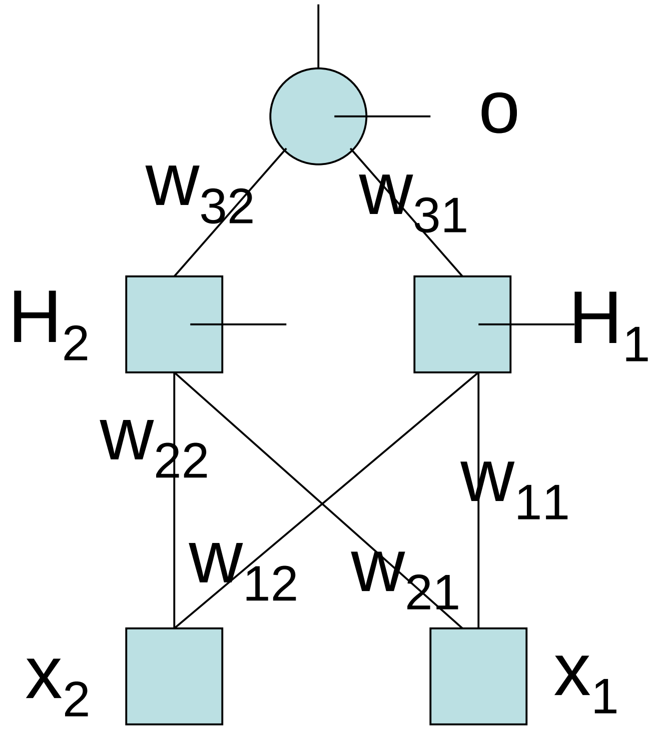
3. Dr. Anoop Kunchukuttan, Microsoft and ex-CFILT

Sequence processing m/c

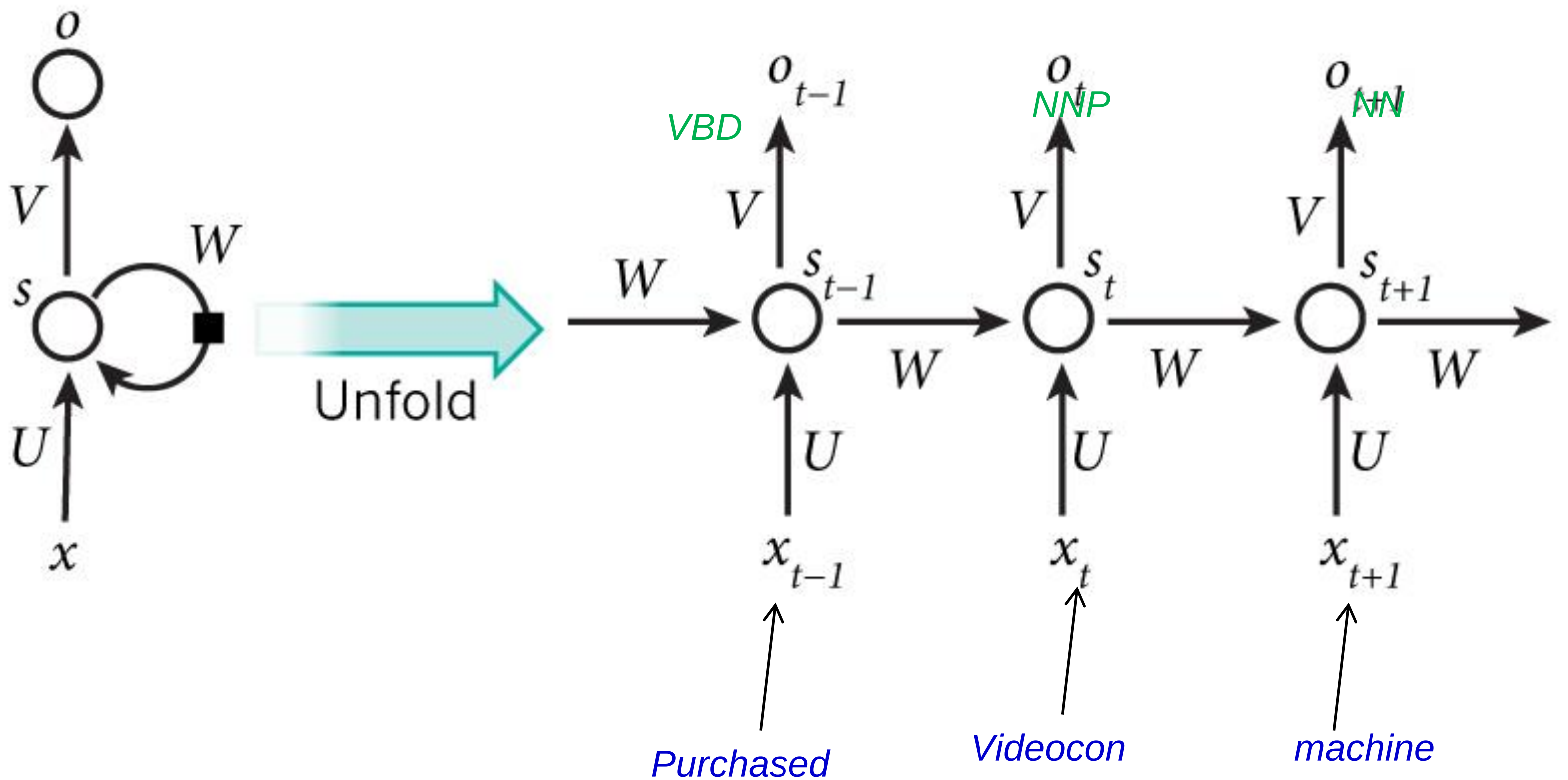


Meaning of state

- State vector → constituted of states of neurons
- State of a neuron → activation, i.e., output of the neuron corresponding to an input
- E.g., state vector for the XOR n/w is $<h_1, h_2, o>$



E.g. POS Tagging



Note that POS of “purchased” is ambiguous with possibilities as VBD or VBN or JJ

“I purchased Videocon machine” vs. “my purchased Videocon machine is running well”

POS Annotation

- *Who_WP is_VZ the_DT prime_JJ minister_NN of _IN India_NNP ?_PUNC*

- Becomes the training data for ML based POS tagging

3 Generations of POS tagging techniques

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

Noisy Channel Model

W Noisy Channel **T**

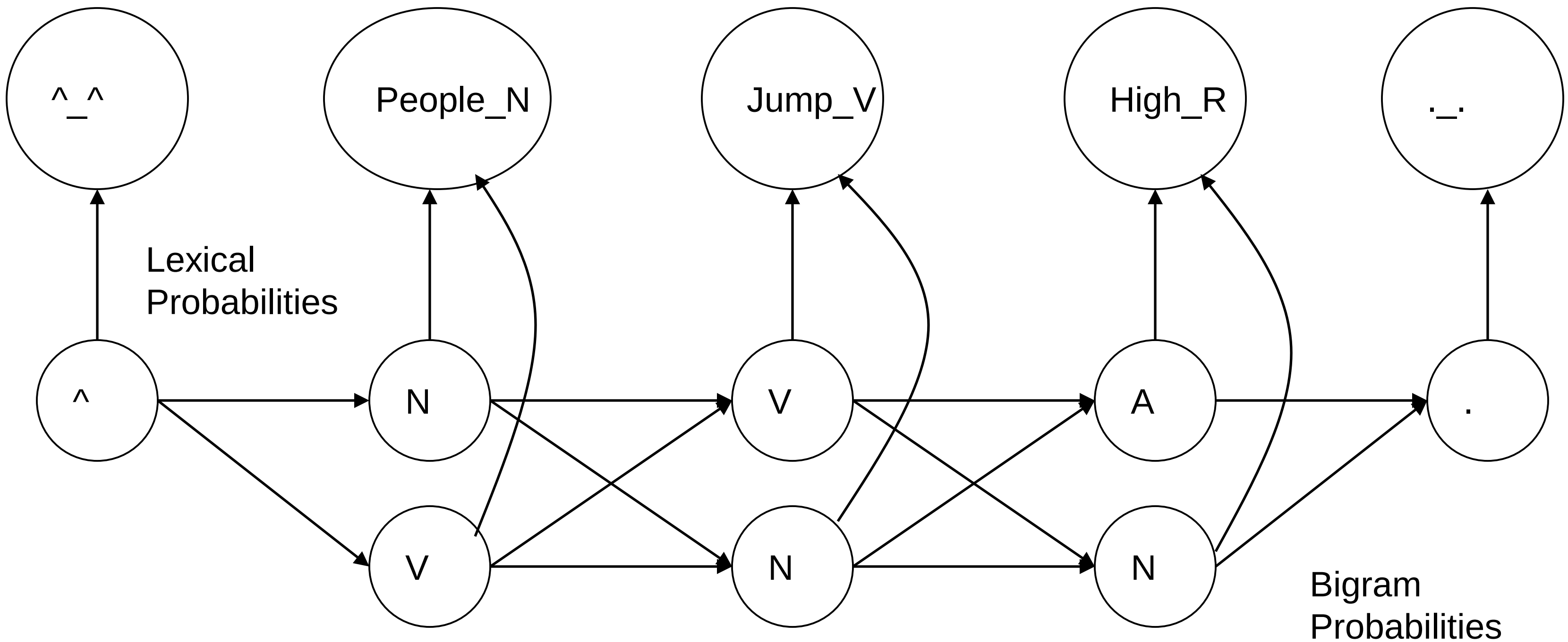
$(w_n, w_{n-1}, \ldots, w_1)$ $(t_m, t_{m-1}, \ldots, t_1)$

Sequence *W* is transformed into sequence *T*

$$T^* = \underset{T}{\operatorname{argmax}}(P(T|W))$$

$$W^* = \underset{W}{\operatorname{argmax}}(P(W|T))$$

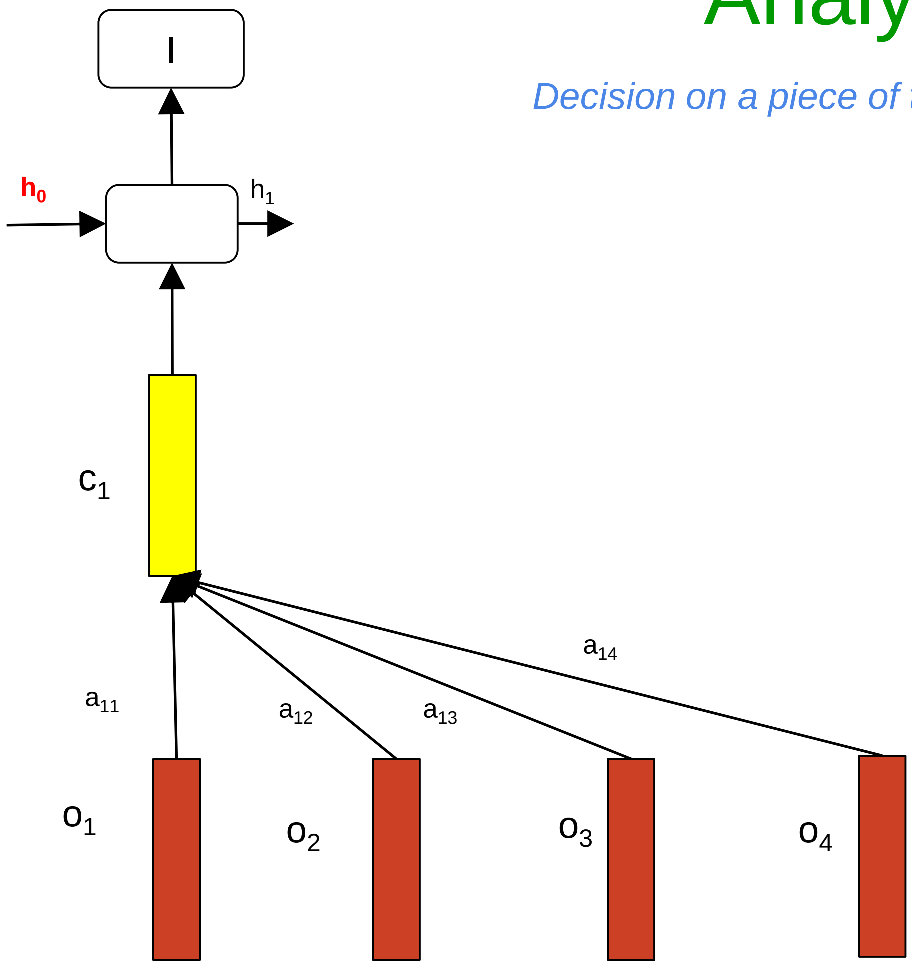
HMM: Generative Model

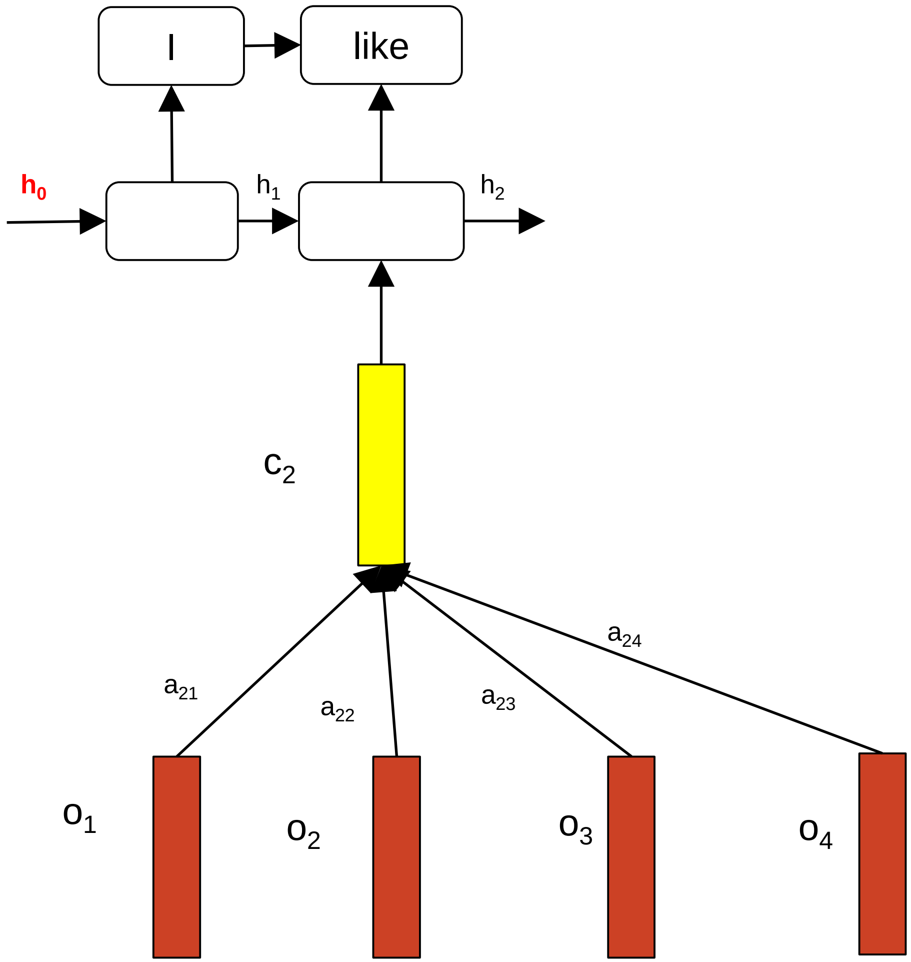


This model is called Generative model.
Here words are observed from tags as states.
This is similar to HMM.

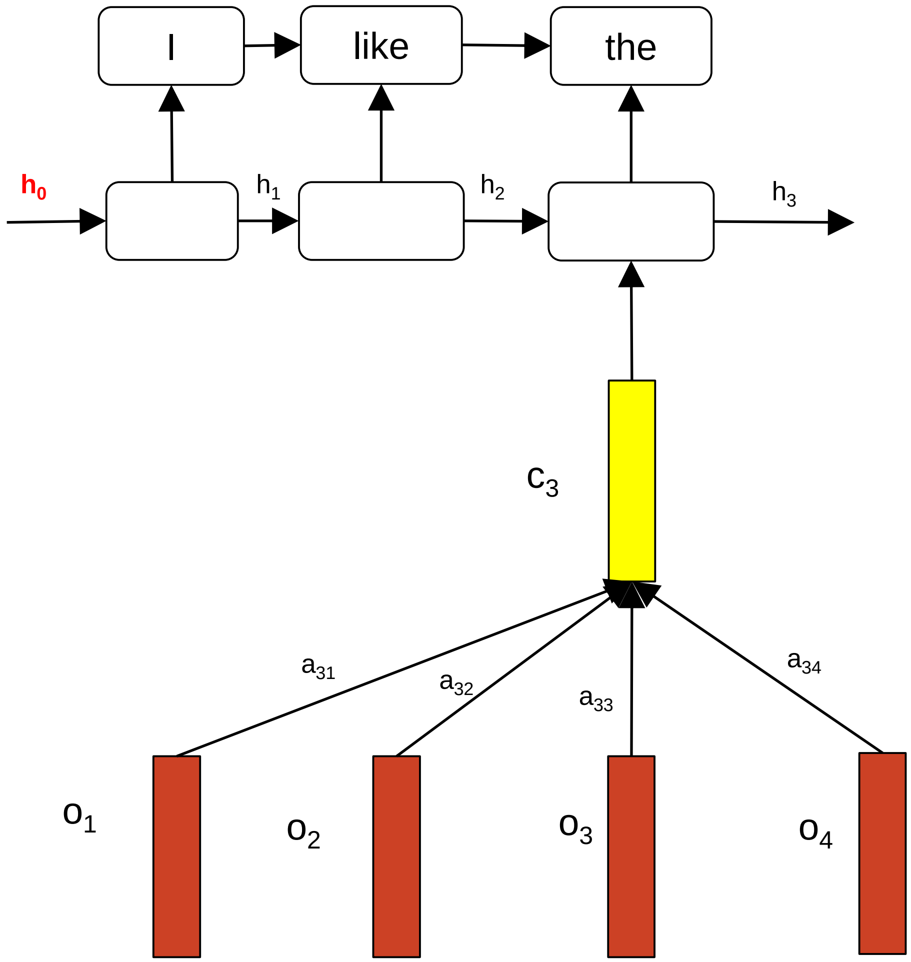
Sequence labeling with RNN: Sentiment Analysis

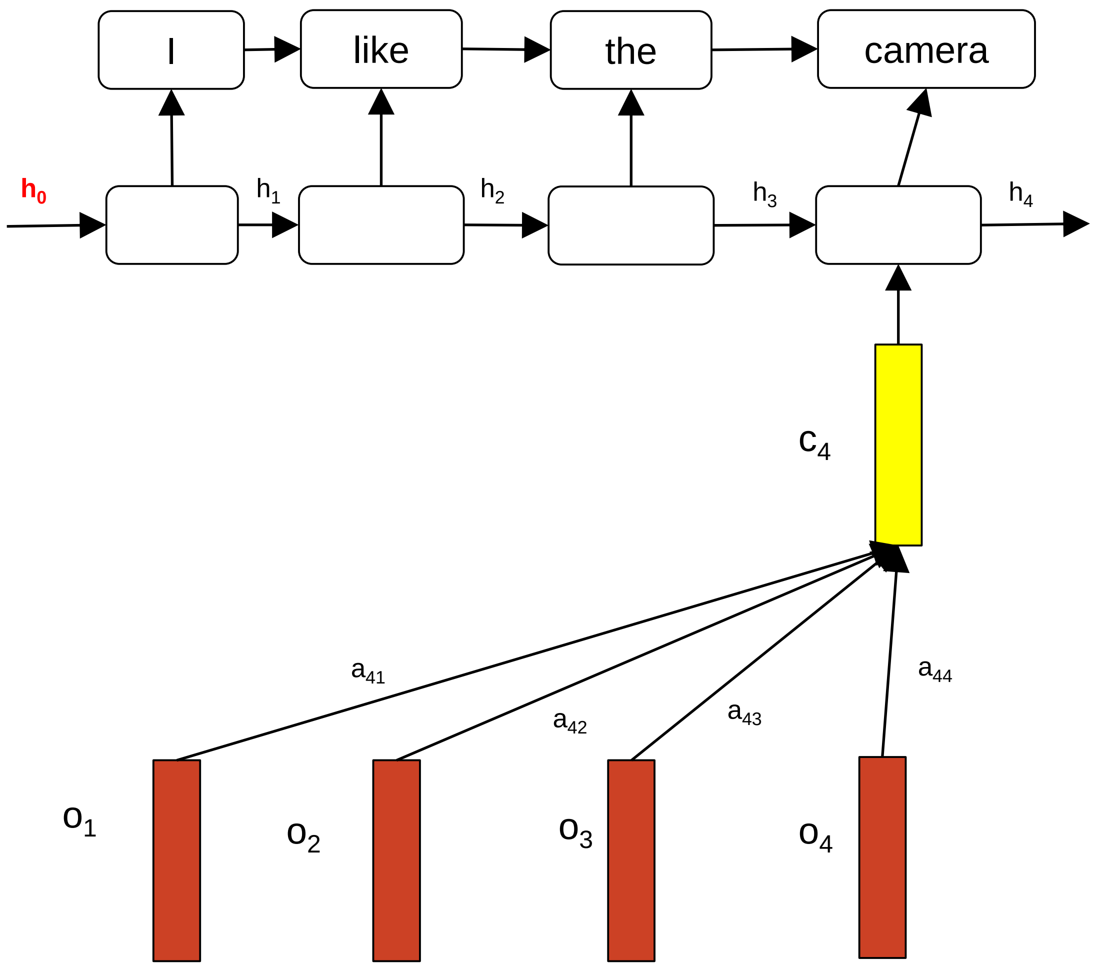
Decision on a piece of text

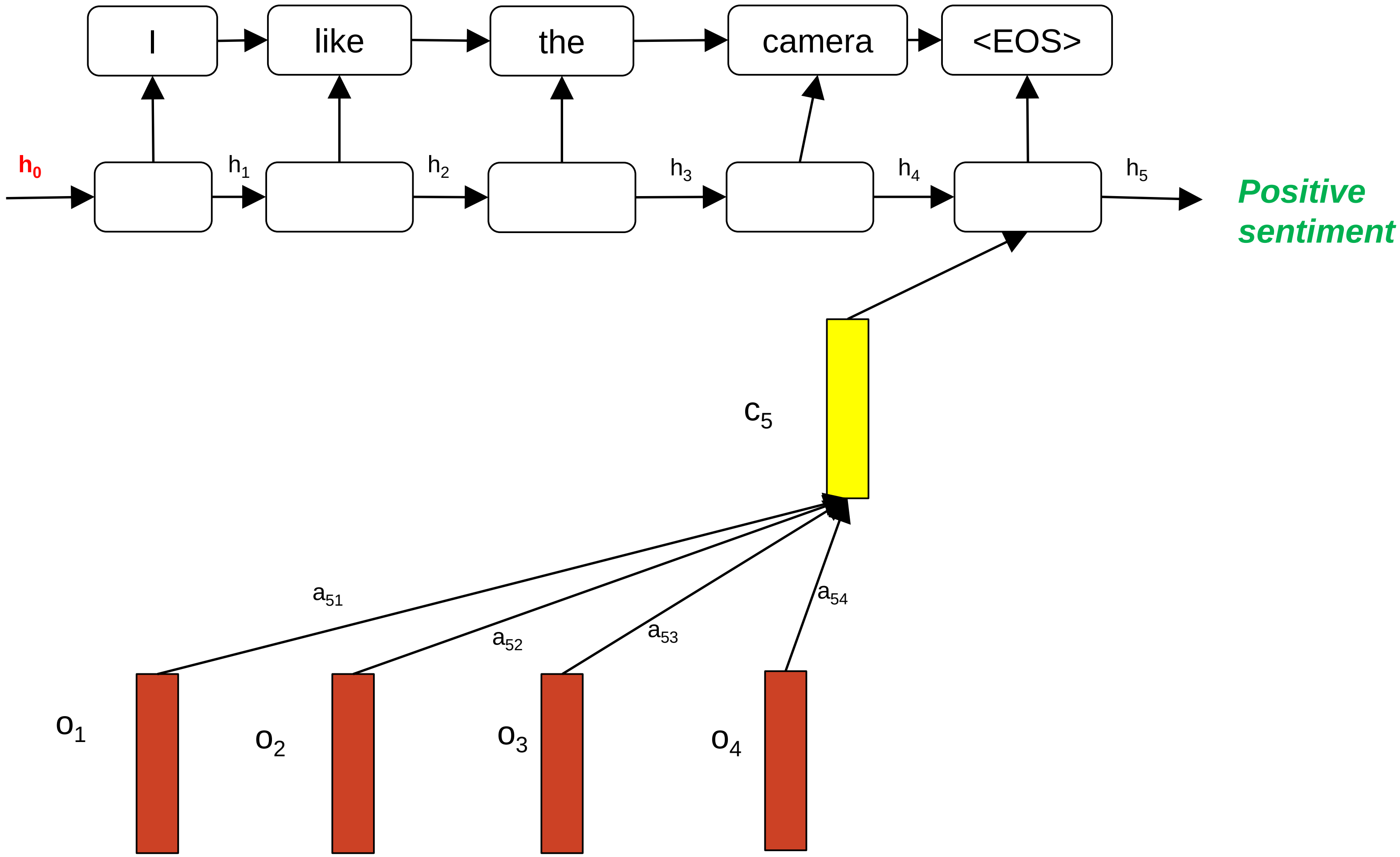




I
like
h0
h1
h2
c2
a21
a22
a23
a24
o1
o2
o3
o4

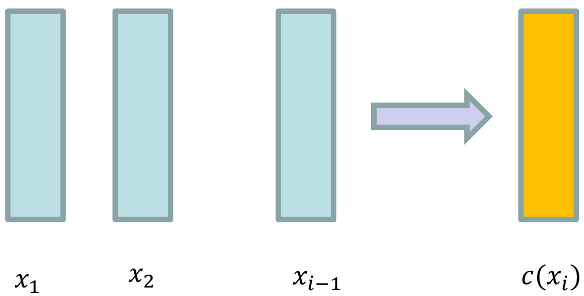

I
like
the
h0
h1
h2
h3
c3
a31
a32
a33
a34
o1
o2
o3
o4


I
like
the
camera
h0
h1
h2
h3
h4
c4
a41
a42
a43
a44
o1
o2
o3
o4


I
like
the
camera
<EOS>
h_0
h_1
h_2
h_3
h_4
h_5
Positive sentiment
c_5
a_{51}
a_{52}
a_{53}
a_{54}
o_1
o_2
o_3
o_4

Recurrent Neural Networks: two key Ideas

1. Summarize context information into a single vector



$$c(x_i) = F(x_1, x_2, \dots, x_{i-1})$$

$$P(x_i | c(x_i))$$

Function G requires all context inputs at once

How does RNN address this problem?

Nature of $P(.)$

n-gram LM: look-up table

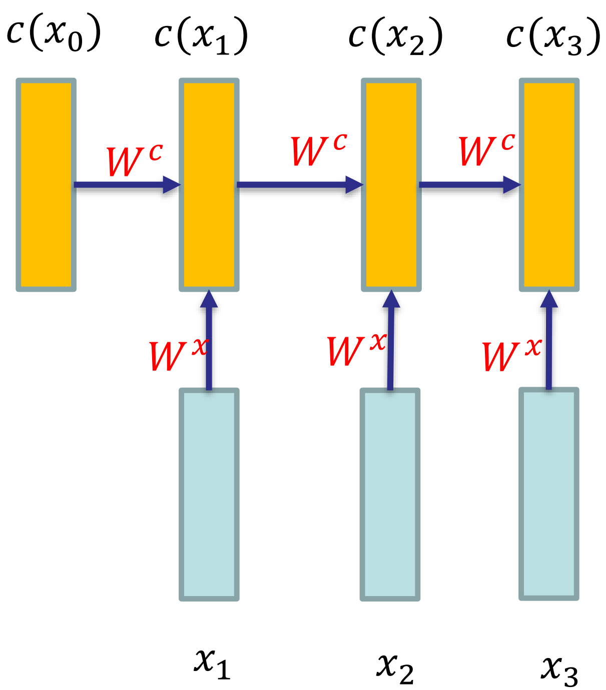
FF LM: $c(x_i) = G(x_{i-1}, x_{i-2})$ (trigram LM)

RNN LM: $c(x_i) = F(x_1, x_2, \dots, x_{i-1})$ (unbounded context)

Two Key Ideas (cntd)

2. Recursively construct the context

$$c(x_i) = F(c(x_{i-1}), x_i)$$



We just need two inputs to construct the context vector:

- Context vector of previous timestep
- Current input

The context vector ➔ state/hidden state/contextual representation

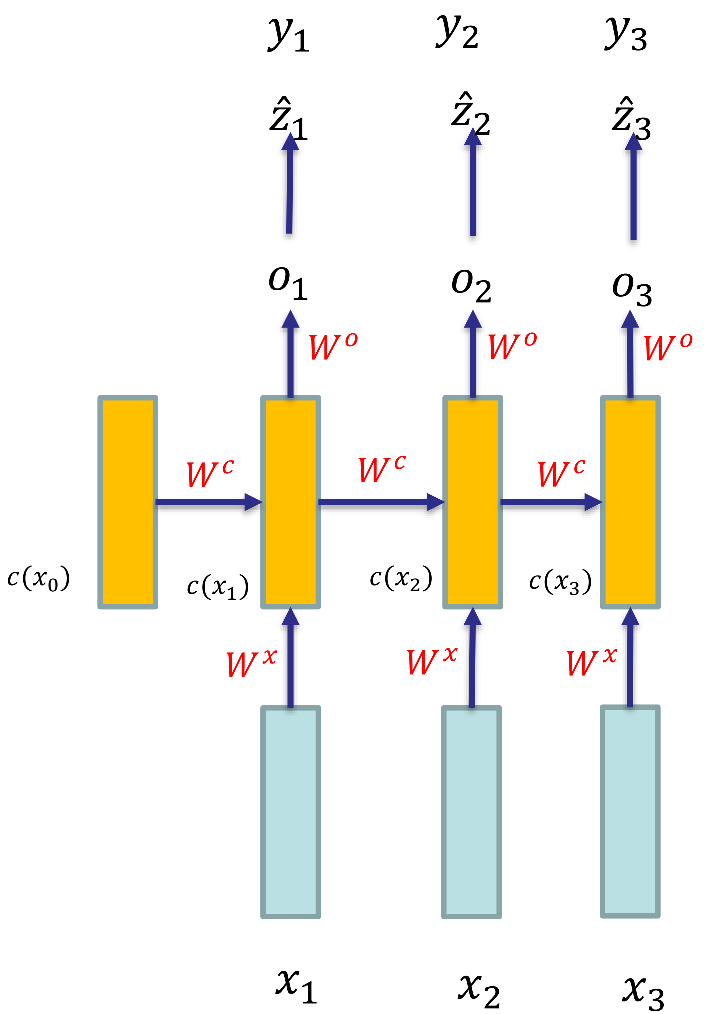
$F(.)$ can be implemented as

$$c(x_i) = \sigma(W^c c(x_{i-1}) + W^x x_i + b_1)$$

Like a feed-forward network

Generate output give the current input and state/context

W^o=wt. for output layer;
W^c= wt. for generating next state (context);
W^x= wt. for the input layer



$$o(x_i) = W^o c(x_i) + b_2$$

We are generally interested in categorical outputs

$$\hat{z}_i = softmax(o(x_i)) = P(y_i|ctx(x_i))$$

$$\widehat{z_i^W} = P(y_i = w|ctx(x_i))$$

The same parameters are used at each time-step

Model size does not depend on sequence length

Long range context is modeled

Sequence Labelling Task

Input Sequence: $(x_1 \;\; x_2 \;\; x_3 \;\; x_4 \ldots\ldots x_i \ldots\ldots\ldots x_N)$

Output Sequence: $(y_1 \;\; y_2 \;\; y_3 \;\; y_4 \ldots\ldots y_i \ldots\ldots\ldots y_N)$

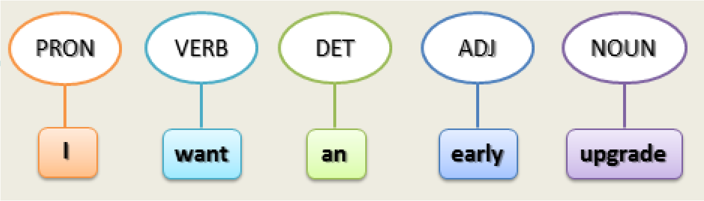
Input and output sequences have the same length

Variable length input

Output contains categorical labels

Output at any time-step typically depends on neighbouring output labels and input elements

Part-of-speech tagging



Recurrent Neural Network is a powerful model to learn sequence labelling tasks

How do we model language modeling as a sequence labeling task?

The	capital	of	Maharashtra	is	Mumbai	<EOS>
<BOS>	The	capital	of	Maharashtra	is	Mumbai

The output sequence is one-time step ahead of the input sequence

Training Language Models (1/2)

Input: large monolingual corpus

- Each example is a tokenized sentence (sequence of words)
- At each time step, predict the distribution of the next word given all previous words
- Loss Function:
  - Minimize cross-entropy between actual distribution and predicted distribution
  - Equivalently, maximize the likelihood

Training Language Models (2/2)

At a single time-step:

$$J_i(\theta) = CE(z_i, \hat{z}_i) = -\sum_{w \in V} z_i^w \log \widehat{z_i^w} = -\log \widehat{z_i^L}$$

Average over time steps for example n:

$$J^n(\theta) = \frac{1}{T} \sum_{i=1}^{T} J_i(\theta)$$

Average over entire corpus:

where $y_i = L$

$$J(\theta) = \frac{1}{N} \sum_{k=1}^{N} J^n(\theta)$$

How do we learn model parameters?
More on that later!

Evaluating Language Models

How do we evaluate quality of language models?

↓

Evaluate the ability to predict the next word given a context

↓

Evaluate the probability of a testset of sentences

Standard test sets exist for evaluating language models:
Penn Treebank, Billion Word Corpus, WikiText

Evaluating LM (cntd.)

- Ram likes to play -----
  - Cricket: high probability, low entropy, low perplexity (relatively very high frequency for ‘like to play cricket’)
  - violin: -do- (relatively high frequency possibility for ‘like to play violin’
  - Politics: moderate probability, moderate entropy, moderate perplexity (relatively moderate frequency ‘like to play politics’
  - milk: almost 0 probability, very high entropy, very high perplexity (relatively very low possibility for ‘like to play milk’

So an LM that predicts ‘milk’ is bad!

Language Model Perplexity

Perplexity: $\exp\big(J(\theta)\big)$

$J(\theta)$ is cross-entropy on the test set

Cross-entropy is measure of difference between actual and predicted distribution

Lower perplexity and cross-entropy is better

Model	Perplexity
Interpolated Kneser-Ney 5-gram (Chelba et al., 2013)	67.6
RNN-1024 + MaxEnt 9-gram (Chelba et al., 2013)	51.3
RNN-2048 + BlackOut sampling (Ji et al., 2015)	68.3
Sparse Non-negative Matrix factorization (Shazeer et al., 2015)	52.9
LSTM-2048 (Jozefowicz et al., 2016)	43.7
2-layer LSTM-8192 (Jozefowicz et al., 2016)	30
Ours small (LSTM-2048)	43.9
Ours large (2-layer LSTM-2048)	39.8

https://research.fb.com/building-an-efficient-neural-language-model-over-a-billion-words/

RNN models outperform n-gram models

A special kind of RNN network – LSTM- does even Better

Importance of Probabilistic Language Modelling (1/2)

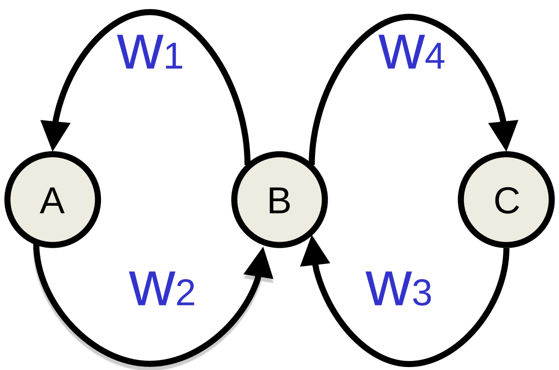
- Context free grammar for language models
  - Is a given string of words in language or not
  - Example:
    - *Ram saw Shyam* (correct word order)
    - *Ram Shyam saw* (incorrect word order)
  - However, belongingness to language is not a black and white issue
  - There are no grammatical and ungrammatical sentences, only sentences with probabilities

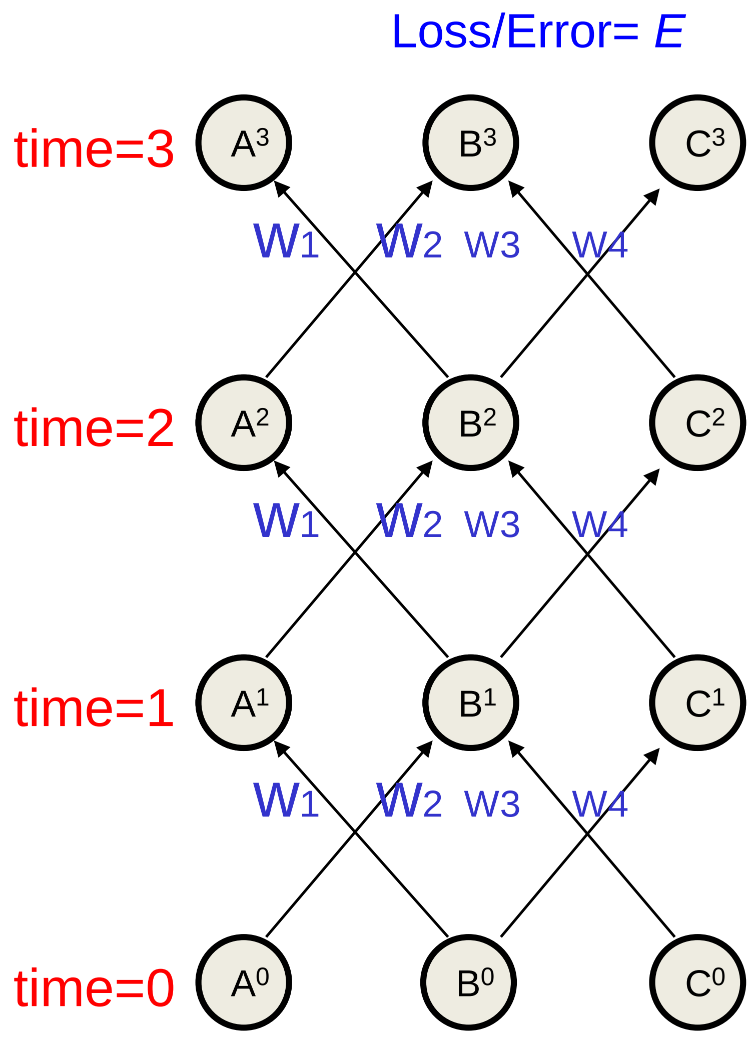
Importance of Probabilistic Language Modelling (2/2)

- Example:
  - Indian English: *You will go to the movie, no?*
  - US/UK English: *You will go to the movie, won't you?*
- English has different forms through differences in regional dialects and even through periods of time
  - English language evolves every year, new words and their different sentence positions are introduced
- Hence we cannot assign 0/1 value to sentences
  - But we can assign probabilities to word orders
  - Equivalent to Prob (Wn | W1,W2,...Wn-1)

BPTT

The equivalence between feedforward nets and recurrent nets





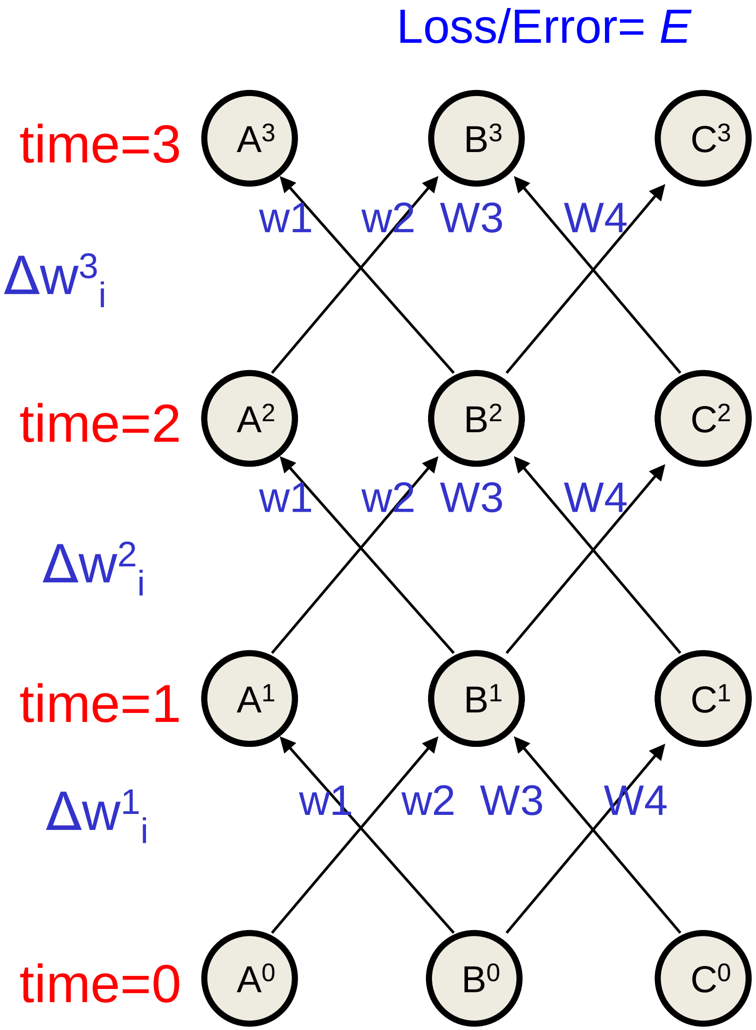
Assume that there is a time delay of 1 in using each connection.

The recurrent

BPTT illustration

$$\Delta w_i = \Delta w^3_i + \Delta w^2_i + \Delta w^1_i$$

Vanishing/Exploding Gradient can strike!!!



BPTT important points

- The forward pass at each time step.
- The backward pass computes the error derivatives at each time step.
- After the backward pass we add together the derivatives at all the different times for each weight.

Long word sequences

- The famous book by Charles Dickens “A Tale of Two Cities” starts the book with the famous sentence “This was the best of times, this was the worst of times….”
- The sentence has 119 words
- “It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way--in short, the period was so far like the present period that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.

The "best of times…" sentence

- Vanishing gradient will surely strike!!

  ↳ very long sentences will have vanishing gradient problem.

- Exercise: give an example from NLP, where exploding gradient will strike!!

Language translation, mitigated using attention mechanism, gradient clipping and transformers. (Document identification etc.) → where a lot of input is fed into the network.

Representation Learning

Basics

- What is a good representation? At what granularity: words, n-grams, phrases, sentences
- Sentence is important- (a) *I <u>bank</u> with SBI; (b) I took a stroll on the river <u>bank</u>; (c) this <u>bank</u> sanctions loans quickly*
- Each 'bank' should have a different representation
- We have to LEARN these representations

Principle behind representation

- Proverb: “A man is known by the company he keeps”
- Similarly: “A word is known/represented by the company it keeps”
- “Company” → Distributional Similarity

Starting point: 1-hot representation

- Arrange the words in lexicographic order
- Define a vector *V* of size |*L*|, where *L* is the lexicon
- For word w_i in the i^{th} position, set the ith bit to 1, all other bits being 0.
- Problem: cosine similarity of ANY pair is 0; wrong picture!!

Representation: to learn or not learn?

- 1-hot representation does not capture many nuances, e.g., semantic similarity
  - But is a good starting point
- Co-occurences also do not fully capture all the facets
  - But is a good starting point

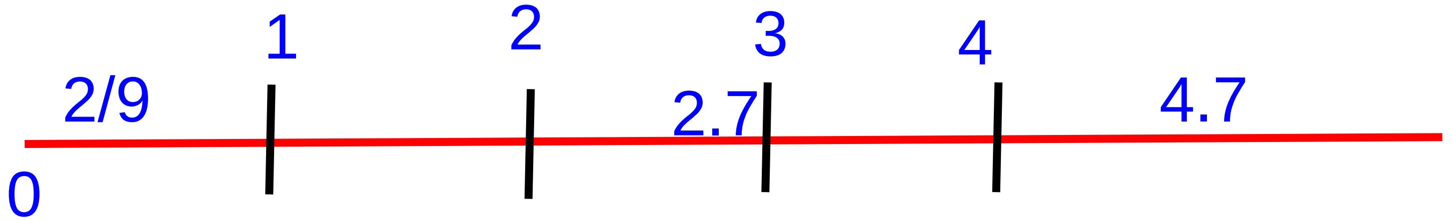
So learn the representation…

- Learning Objective

- ***MAXIMIZE CONTEXT PROBABILITY***

Foundations-1: Embedding

- Way of taking a discrete entity to a continuous space
- E.g., 1, 2, 3, 2.7, 2/9, $22^{1/2}$, … are numerical symbols
- But they are points on the real line
- Natural embedding
- Words' embedding not so intuitive!



Foundations-2: Purpose of Embedding

- Enter geometric space
- Take advantage of “distance measures”- Euclidean distance, Riemannian distance and so on
- “Distance” gives a way of computing similarity

Foundations-3: Similarity and difference

- Recognizing similarity and difference- foundation of intelligence
- Lot of Pattern Recognition is devoted to this task (Duda, Hart, Stork, 2nd Edition, 2000)
- Lot of NLP is based on Text Similarity
- Words, phrases, sentences, paras and so on (verticals)
- Lexical, Syntactic, Semantic, Pragmatic (Horizontal)

Similarity study in MT

English:

This blanket is very soft

Hindi:

yaha kambal bahut naram hai

Bangla:

ei kambal ti khub naram <null>

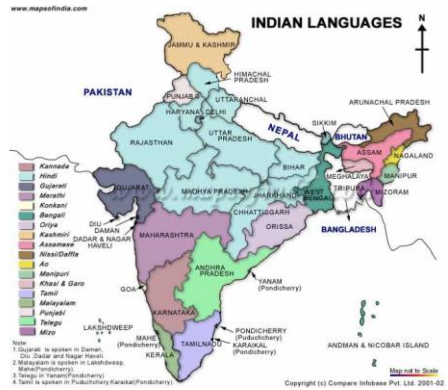
Marathi:

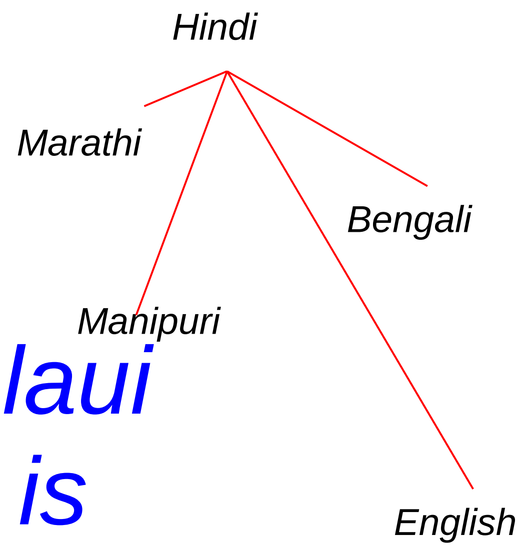
haa kambal khup naram aahe

Manipuri:

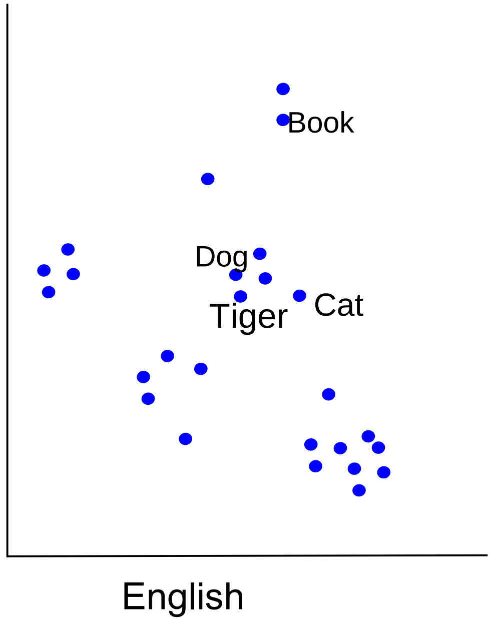
kampor asi mon mon laui

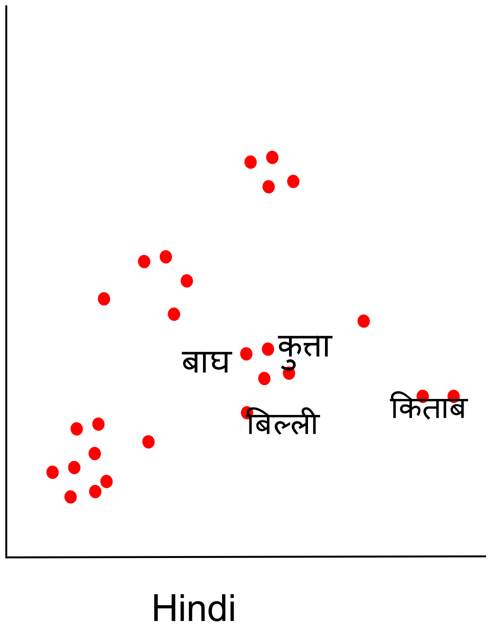
blanket this soft soft is





ISO-Metricity



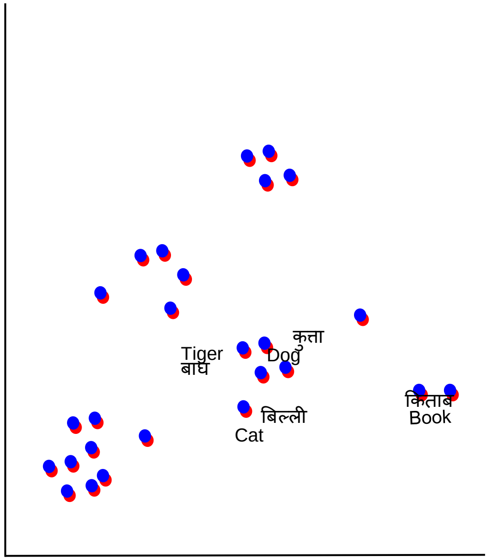


Across Cross-lingual Mapping

This involves strong assumption that embedding spaces across languages are isomorphic, which is not true specifically for distance languages (Søgaard et al. 2018). However, without this assumption unsupervised NMT is not possible.

Neural

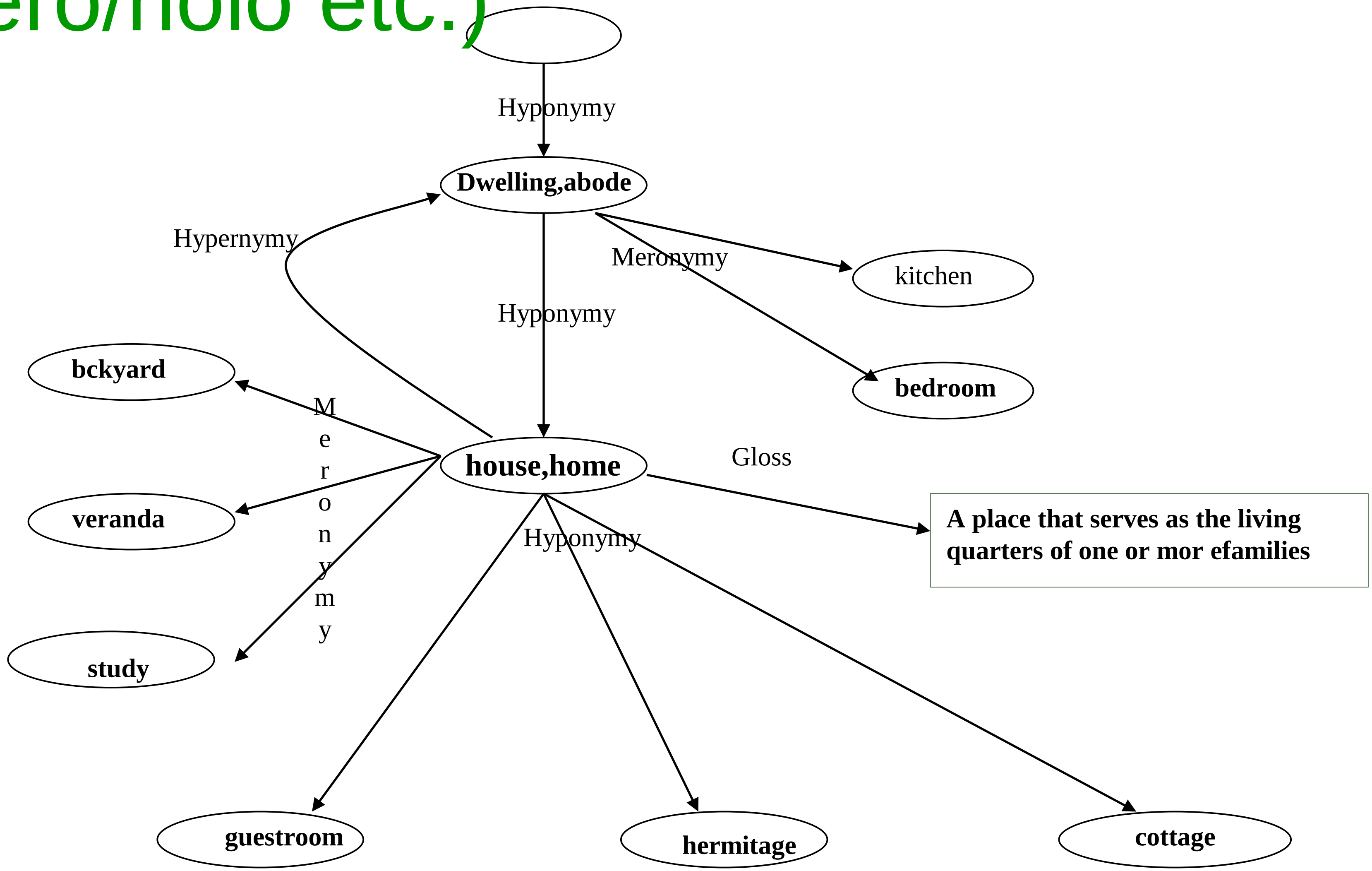
Søgaard, Anders, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. ACL



Foundations-4: Syntagmatic and Paradigmatic Relations

- Syntagmatic and paradigmatic relations
  - Lexico-semantic relations: synonymy, antonymy, hypernymy, mernymy, troponymy etc. **CAT is-a ANIMAL**
  - Coccurence: **CATS MEW**
- Wordnet: primarily paradigmatic relations
- ConceptNet: primarily Syntagmatic Relations

WordNet Sub-Graph with lexico-semantic relations (hyper/hypo, mero/holo etc.)



Lexical and Semantic relations in wordnet

1. Synonymy (e.g., *house, home*)
2. Hypernymy / Hyponymy (kind-of, e.g., *cat* ←→ *animal*)
3. Antonymy (e.g., *white and black*)
4. Meronymy / Holonymy (part of, e.g., *cat and tail*)
5. Gradation (e.g., *sleep*→*doze*→*wake up*)
6. Entailment (e.g., *snoring* → *sleeping*)
7. Troponymy (manner of, e.g., *whispering and talking*)

1, 3 and 5 are lexical (*word to word)*, rest are semantic (*synset to synset).*

‘Paradigmatic Relations’ and ‘Substitutability’

- Words in paradigmatic relations can substitute each other in the sentential context
- E.g., ‘The cat is drinking milk’ → ‘The animal is drinking milk’
- Substitutability is a foundational concept in linguistics and NLP

Foundations-5: Learning and Learning Objective

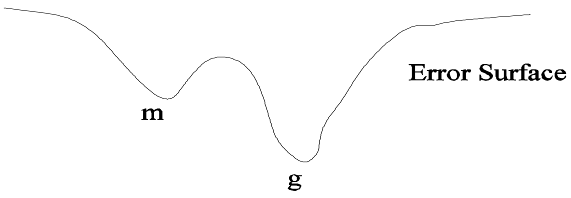
- Probability of getting the context words given the target should be maximized (skip gram)

- Probability of getting the target given context words should be maximized (CBOW)

Important concepts associated with FFNN-BP

Local Minima

Due to the Greedy nature of BP, it can get stuck in local minimum *m* and will never be able to reach the global minimum *g* as the error can only decrease by weight change.



m- local minima, g- global minima

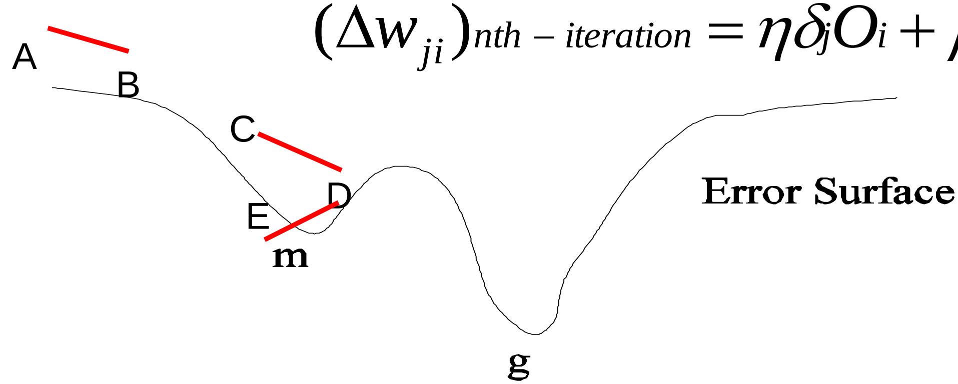
Figure- Getting Stuck in local minimum

Momentum factor

1. Introduce momentum factor.

- Accelerates the movement out of the trough.
- Dampens oscillation inside the trough.
- Choosing β : If β is large, we may jump over the minimum.

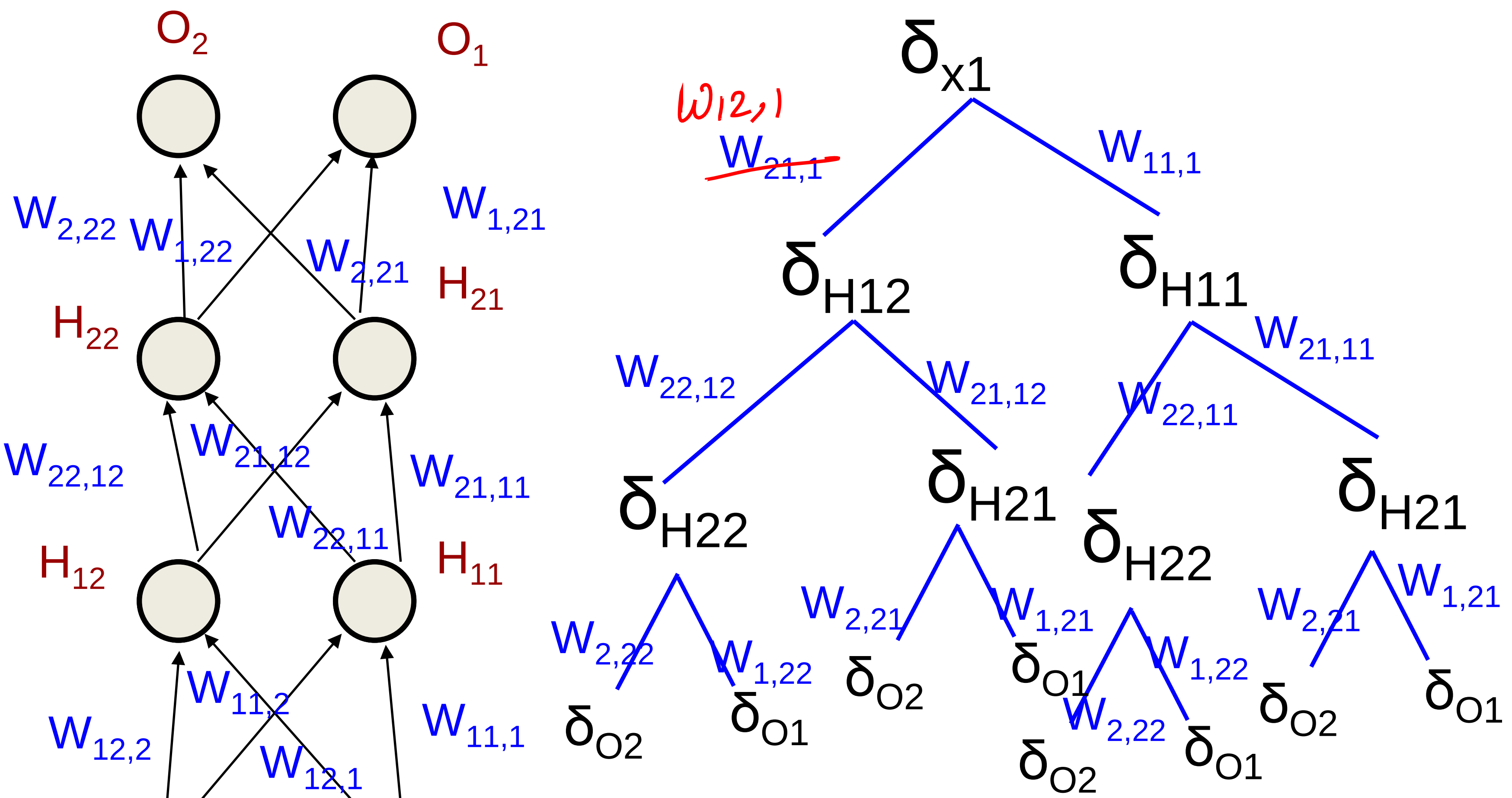
$$(\Delta w_{ji})_{nth-iteration} = \eta \delta_j O_i + \beta (\Delta w_{ji})_{(n-1)th-iteration}$$



m- local minima, g- global minima

Figure- Getting Stuck in local minimum

Vanishing/Exploding Gradient



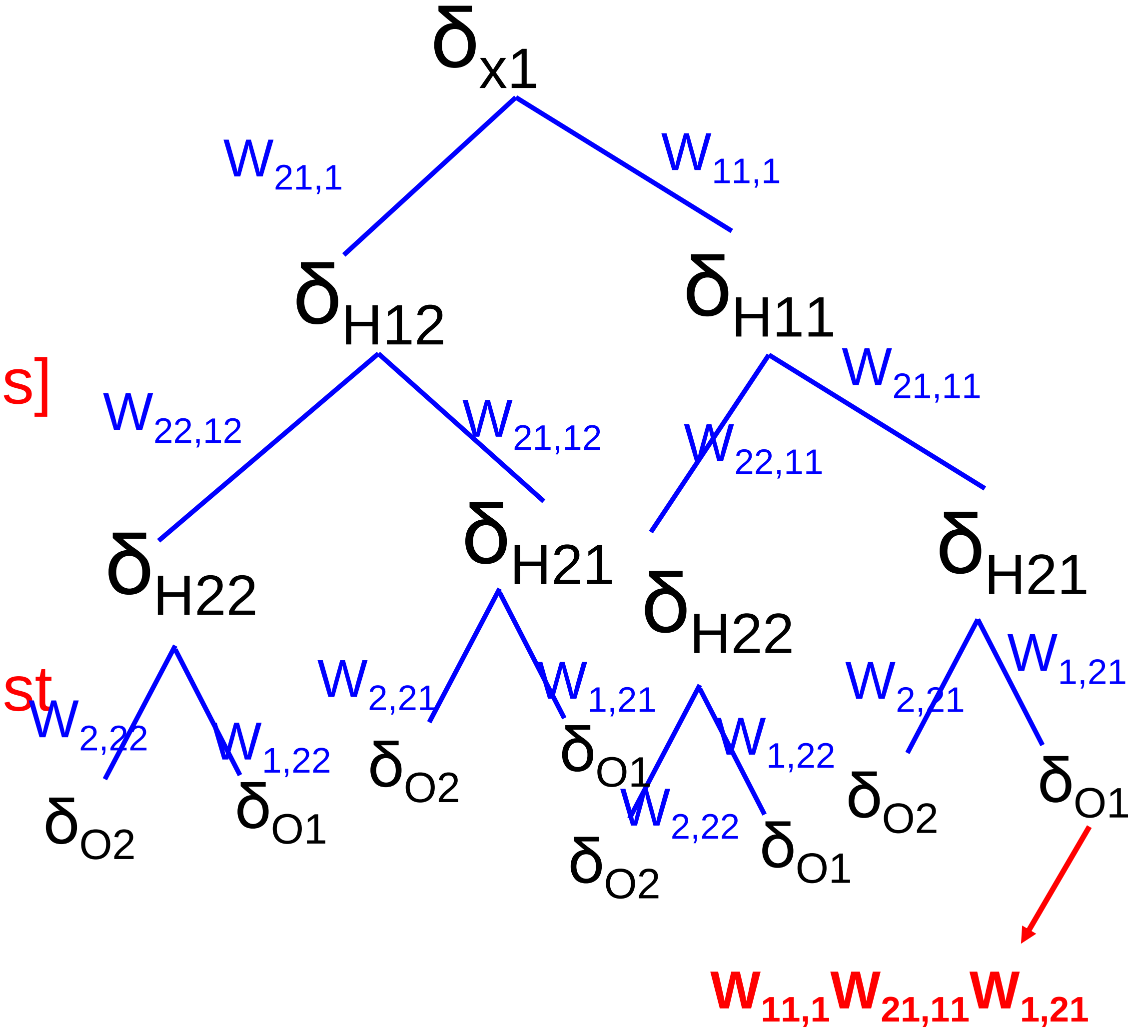
$\delta_{x1}=W_{11,1}\delta_{H11}f'(.)+W_{21,1}\delta_{H1}$ $f'(.)$, $f'(.)$ is derivative of sigmoid

Vanishing/Exploding Gradient

$\delta_{x1}=W_{11,1}\delta_{H11}f'(.)+W_{21,1}\delta_{H12}f'(.)$ [2 terms]

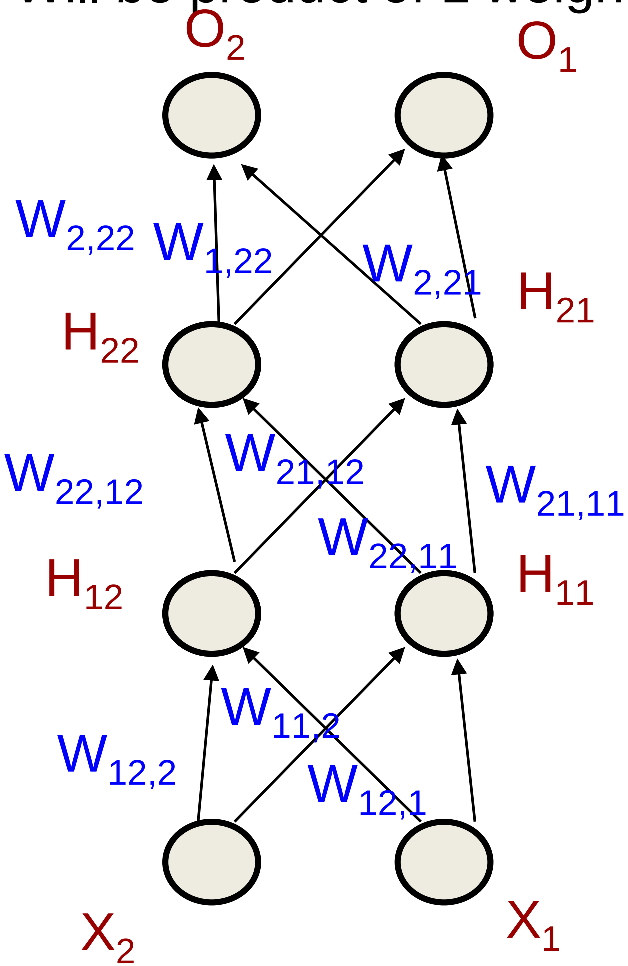
$=W_{11,1}(W_{21,11}\delta_{H21}f'(.)+W_{22,11}\delta_{H22}f'(.))f'(.)+W_{21,1}(W_{21,12}\delta_{H21}f'(.)+W_{22,12}\delta_{H22}f'(.))\ f'(.)$ [4 terms]

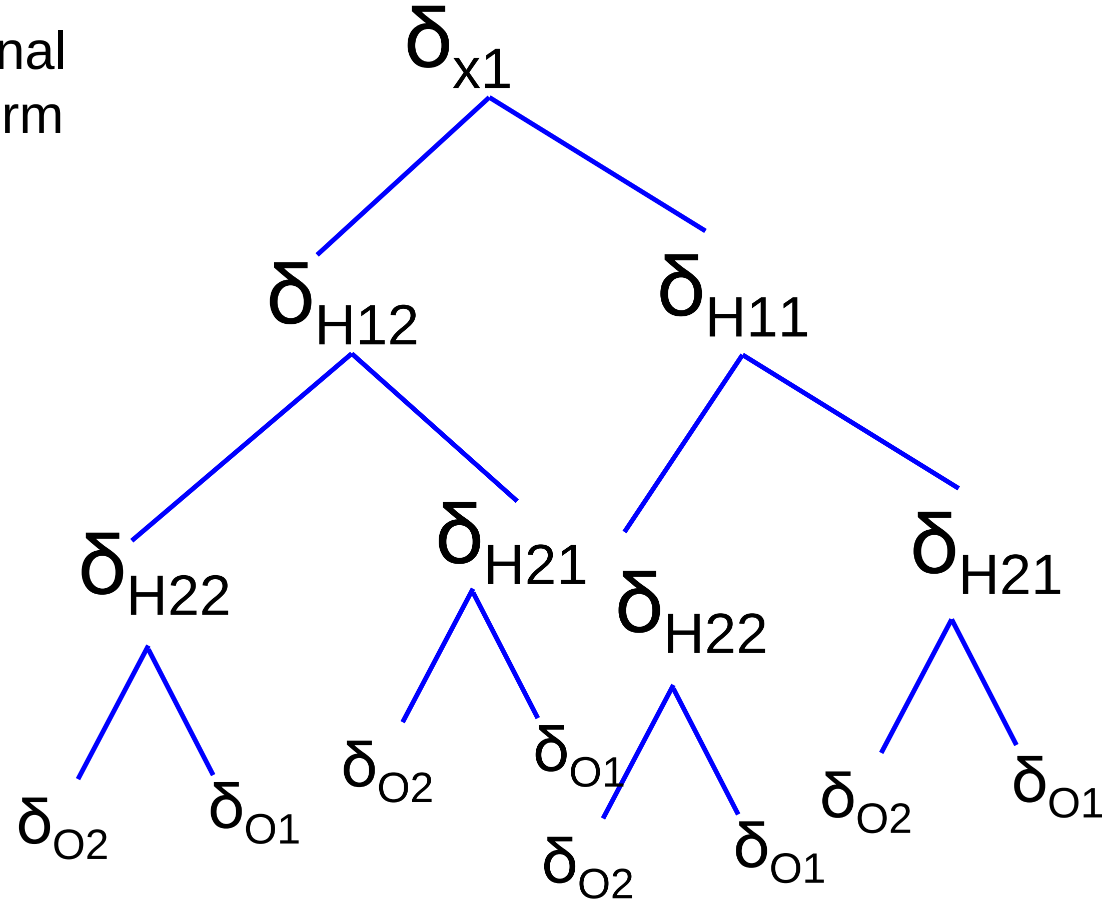
= (4 terms with δ_{o1}) + (4 terms with δ_{o2}; one term shown for the leftmost leaf's weight); also each term has product of derivatives



Vanishing/Exploding Gradient

With '*B*' as branching factor and
'*L*' as number of levels,
There will be B^L terms in the final
Expansion of δ_{x1}. Also each term
Will be product of L weights

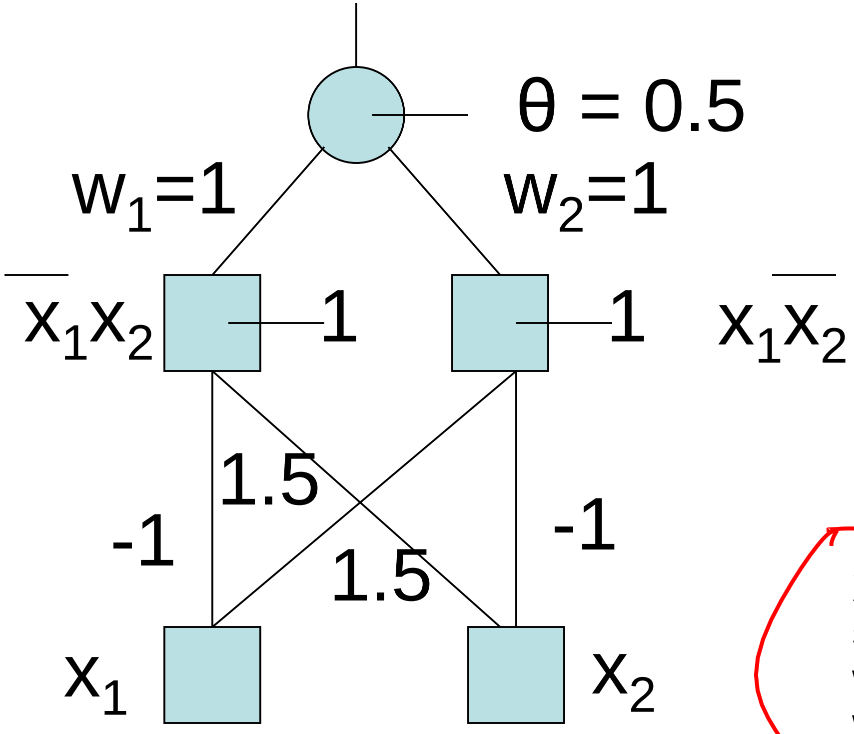




Each term also gets multiplied with
product of derivatives of sigmoid L times.
These products can vanish or explode.

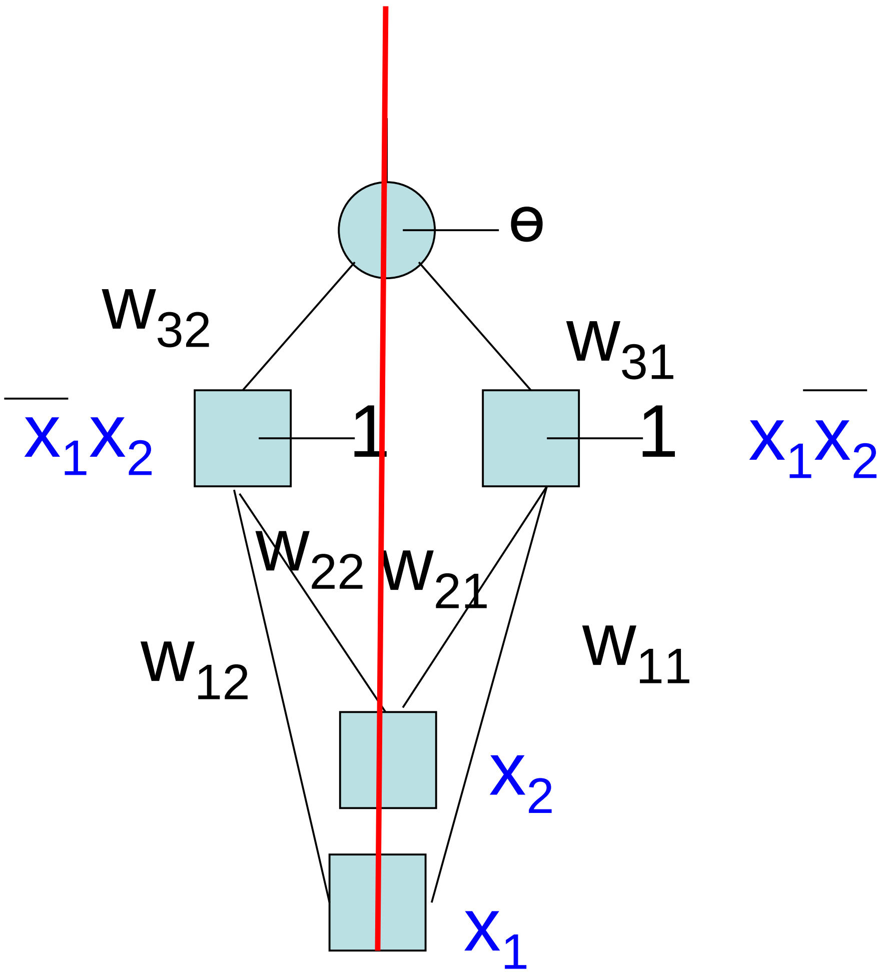
Symmetry breaking

- If mapping demands different weights, but we start with the same weights everywhere, then BP will never converge.



XOR n/w: if we s started with identical weight everywhere, BP will not converge

Symmetry breaking: understanding with proper diagram



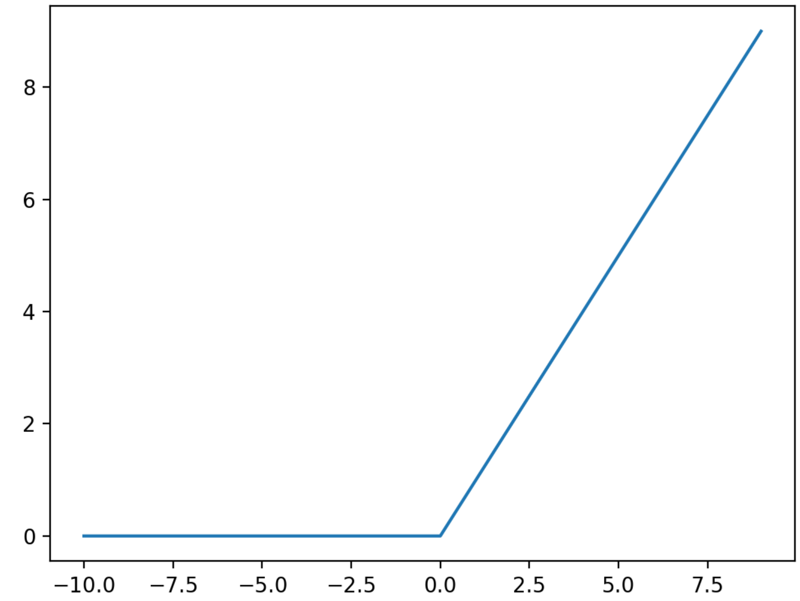
Symmetry About The red Line should Be broken

Working with RELU

Rectifier Linear Unit

What is RELU

y=relu(x)=max(0,x)



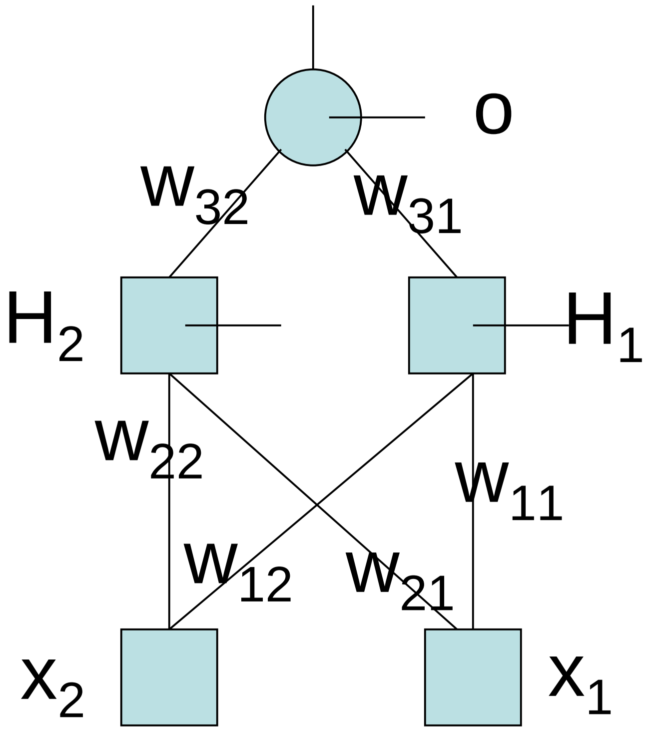
dy/dx

= *0 for x<o*

= *1 for x>0*

= *0* (forced to be 0 at x=0, though does not exit)

Output sigmod and hidden neurons as RELU



$$\Delta w_{ji} = -\eta \frac{\delta E}{\delta w_{ji}}$$

$$\eta = \text{learning rate}, 0 \le \eta \le 1$$

$$\frac{\delta E}{\delta w_{ji}} = \frac{\delta E}{\delta net_j} \times \frac{\delta net_j}{\delta w_{ji}}$$

$$net_j = \text{input at the j}^{th}\text{ neuron)}$$

$$\frac{\delta E}{\delta net_j} = -\delta j$$

$$\boxed{\Delta w_{ji} = \eta \delta j \frac{\delta net_j}{\delta w_{ji}} = \eta \delta j o_i}$$

Backpropagation – for outermost layer

$$\delta j = -\frac{\delta E}{\delta net_j} = -\frac{\delta E}{\delta o_j} \times \frac{\delta o_j}{\delta net_j} (net_j = \text{input at the j}^{th} \text{ layer})$$

$$E = \frac{1}{2}\sum_{p=1}^{m}(t_p - o_p)^2$$

$$\text{Hence, } \delta j = -(-(t_j - o_j)o_j(1-o_j))$$

$$\Delta w_{ji} = \eta(t_j - o_j)o_j(1-o_j)o_i$$

Backpropagation – for hidden layers

$$\Delta w_{ji} = \eta \delta j o_i$$

$$\delta j = -\frac{\delta E}{\delta net_j} = -\frac{\delta E}{\delta o_j} \times \frac{\delta o_j}{\delta net_j}$$

$$= -\frac{\delta E}{\delta o_j} \times (1\, or\ 0)$$

$$= -\sum_{k \in \text{next layer}} \left(\frac{\delta E}{\delta net_k} \times \frac{\delta net_k}{\delta o_j}\right) \times (1\, or\ 0)$$

$$\text{Hence, } \delta_j = -\sum_{k \in \text{next layer}} (-\delta_k \times w_{kj}) \times (1\, or\ 0)$$

This recursion can give rise to vanishing and exploding Gradient problem

$$= \sum_{k \in \text{next layer}} (w_{kj} \delta_k)\ or\ 0$$

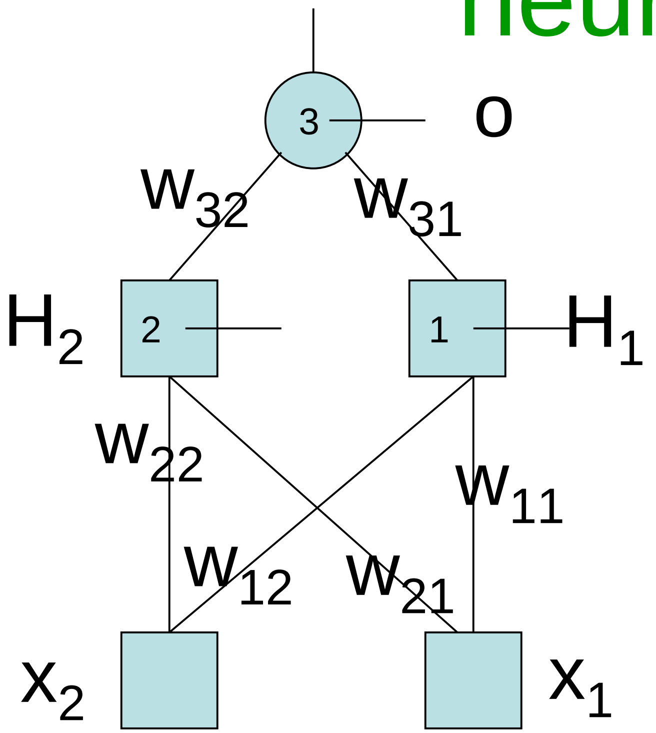
Backpropagation Rule for weight change with RELU, Sigmod and TSS

$$\Delta w_{ji} = \eta \delta j o_i$$

$$\delta_j = (t_j - o_j) o_j (1 - o_j)$$ for outermost layer

$$= \sum_{k \in \text{next layer}} (w_{kj} \delta_k) \; or \; 0$$ for hidden layers

Output sigmoid and hidden neurons as RELU



$$\Delta w_{ji} = \eta \delta j o_i$$

$$\delta_j = (t_j - o_j) o_j (1 - o_j)$$

for outermost layer

$$= \sum_{k \in \text{next layer}} (w_{kj} \delta_k) \; or \; 0$$

for hidden layer

Why is RELU a solution for vanishing or exploding gradient?

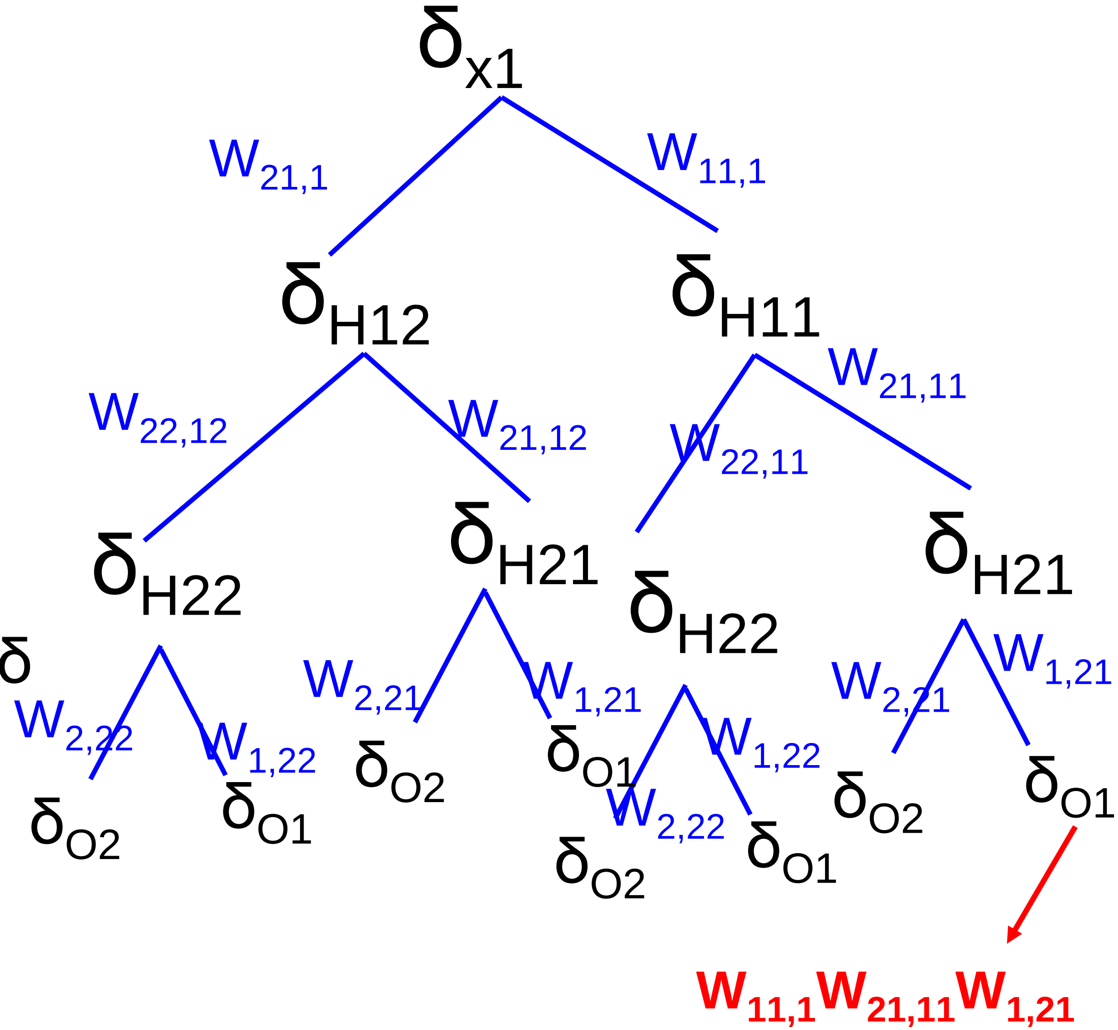
Vanishing/Exploding Gradient

$\delta_{x1}=W_{11,1}\delta_{H11}.1/0+W_{21,1}\delta_{H12}.1/0$ [2 terms]

$=W_{11,1}(W_{21,11}\delta_{H21}.1/0+W_{22,11}\delta_{H22}.1/0).1/0+W_{21,1}(W_{21,12}\delta_{H21}+W_{22,12}\delta_{H22})\ .1/0$ [4 terms]

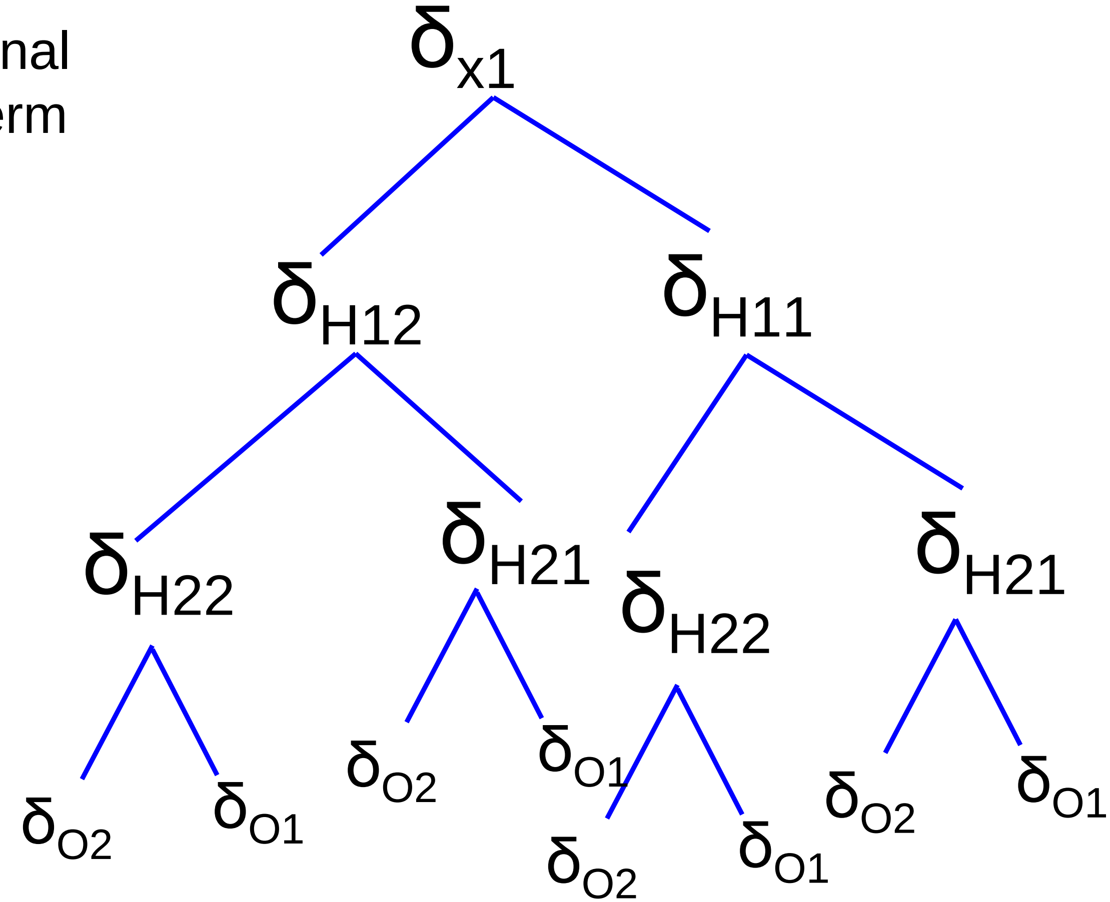
$= W_{11,1}W_{21,11}\delta_{H21}+W_{11,1}W_{22,21}\delta_{H22}+W_{21,1}W_{21,12}\delta_{H21}+W_{21,1}W_{22,12}\delta_{H22}$

= (4 terms with δ_{o1}) + (4 terms with δ_{o2}; one term shown for the leftmost leaf's weight); also each gets multiplied with 1/0



Vanishing/Exploding Gradient

With '*B*' as branching factor and '*L*' as number of levels, There will be B^L terms in the final Expansion of δ_{x1}. Also each term Will be product of L weights



If we use sigmoid, the products of derivatives of sigmoid can have vanishing or exploding effect. In RELU also each term gets multiplied with product of derivatives of RELU L times. These products can be 0 or 1. Thus the delta from output layer gets backpropagated either as such or not at all. This is the crux of the matter of RELU use.

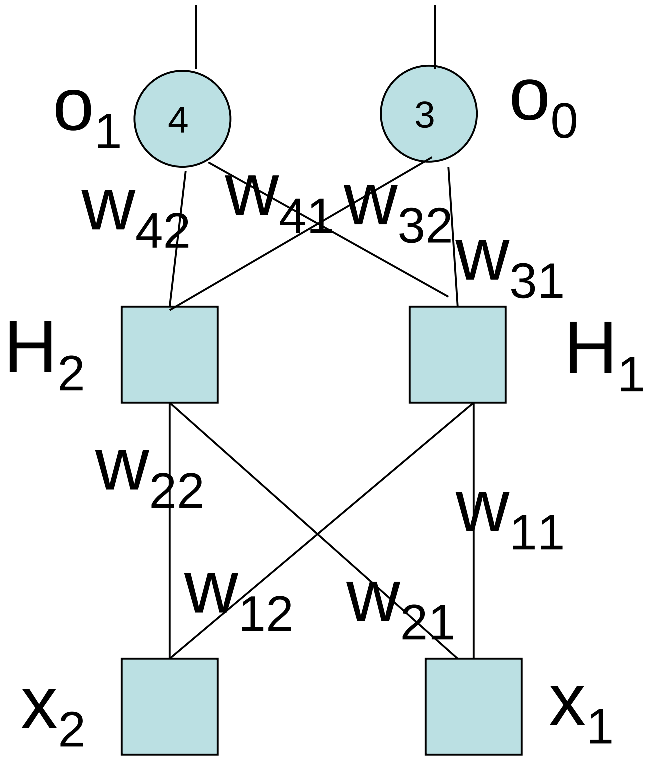
Hence ReLU is used to mitigate the vanishing gradient problem.

Softmax, Cross Entropy and RELU

$$E = -t_1 \log o_1 - t_0 \log o_0$$

$$o_1 = \frac{e^{net_1}}{e^{net_1} + e^{net_0}}, \; o_0 = \frac{e^{net_0}}{e^{net_1} + e^{net_0}}$$

$$\frac{\partial E}{\partial w_{31}} = -\frac{t_1}{o_1}\frac{\partial o_1}{\partial w_{31}} - -\frac{t_0}{o_0}\frac{\partial o_0}{\partial w_{31}}$$



$$\frac{\partial o_1}{\partial w_{31}} = \frac{\partial o_1}{\partial net_1} \cdot \frac{\partial net_1}{\partial w_{31}} + \frac{\partial o_1}{\partial net_0} \cdot \frac{\partial net_0}{\partial w_{31}} = o_1(1-o_1)h_1 + 0$$

$$h_1 = output\ of\ \ H_1$$

$$\frac{\partial o_0}{\partial w_{31}} = \frac{\partial o_0}{\partial net_1} \cdot \frac{\partial net_1}{\partial w_{31}} + \frac{\partial o_0}{\partial net_0} \cdot \frac{\partial net_0}{\partial w_{31}} = -o_1 o_0 h_1 + 0$$

$$\Rightarrow \frac{\partial E}{\partial w_{31}} = -t_1(1-o_1)h_1 + t_0 o_1 h_1$$

$$= -t_1(1-o_1)h_1 + (1-t_1)o_1 h_1$$

$$= [-t_1 + t_1 o_1 + o_1 - t_1 o_1]h_1$$

$$= -(t_1 - o_1)h_1$$

$$\Delta w_{31} = -\eta \frac{\partial E}{\partial w_{31}} = \eta(t_1 - o_1)h_1$$