Text Similarity

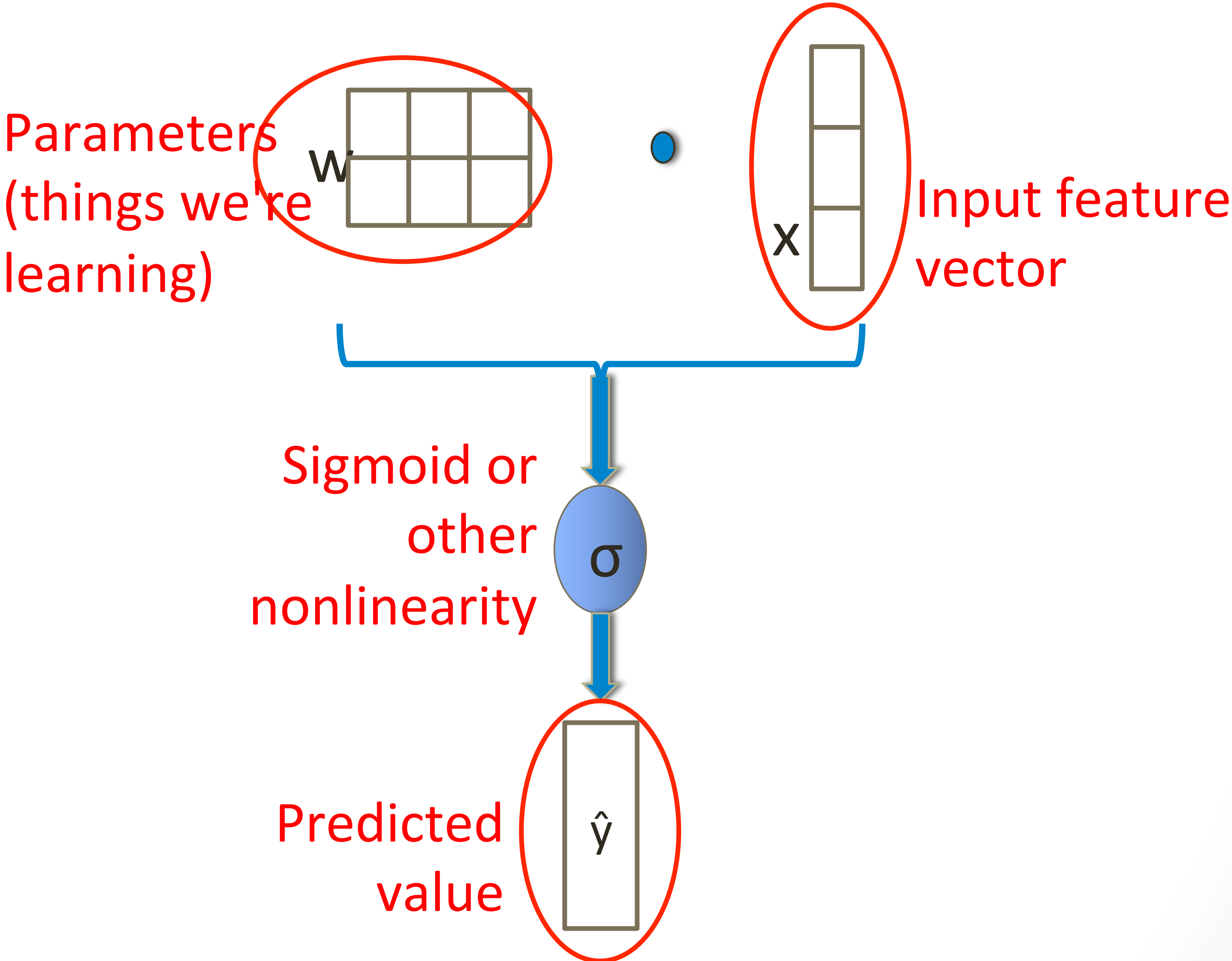
Announcements

- Note problems with Midterm multiple choice questions 2 and 3. If you got them wrong, you will get credit. Bring your exam back to TA hours.
- There will be a recitation tomorrow on HW3. Be sure to provide your interest and availability on Piazza.
- You have 2 weeks for HW3. Due date on the assignment is correct. I have updated web site.

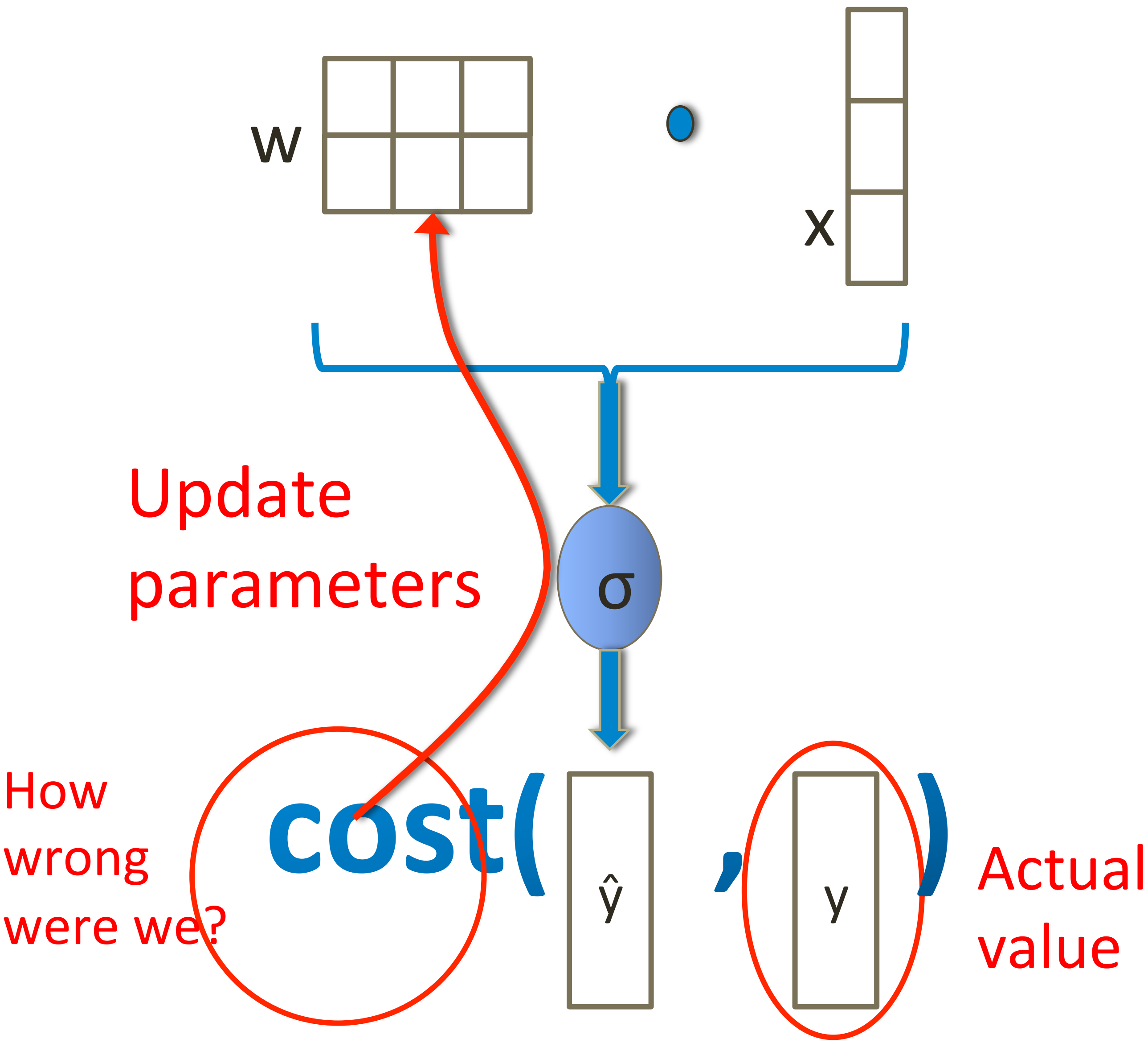
Time to Reflect

- Why does a neural net work?
- What is it good at?
- Empirical vs theory: what do we know?

Supervised Machine Learning



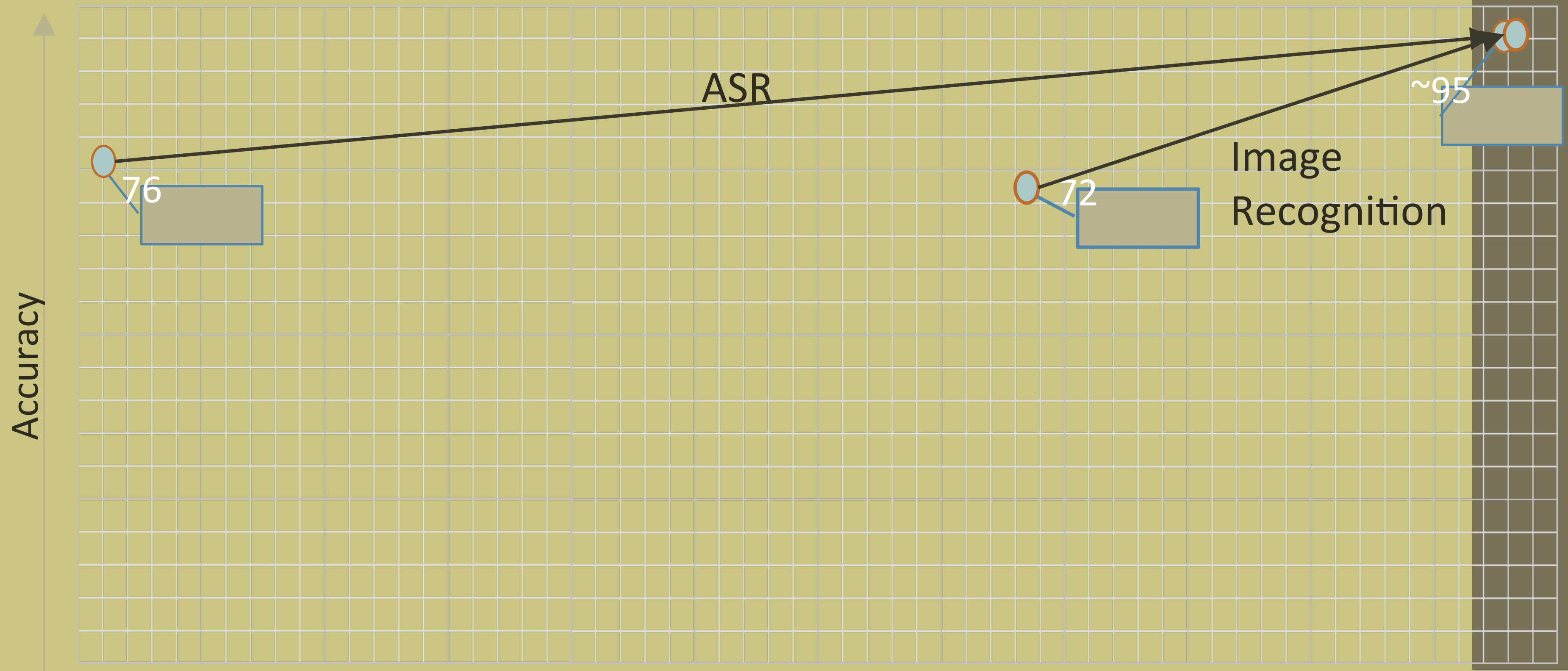
Supervised Machine Learning



Highlights of Neural Nets

- Learn a representation, not just to predict
- Critical component is the embedding layer
  - Mapping from discrete symbols to continuous vectors in low-dimensional space
  - *Semantic representation: Distributed*
- Feed-forward neural networks (multi-layer perceptron) can be used anywhere a linear classifier is used
  - Superior performance often due to non-linearity
- Which parameter values, which neural net (RNN, CNN, LSTM) are best for a task is determined experimentally

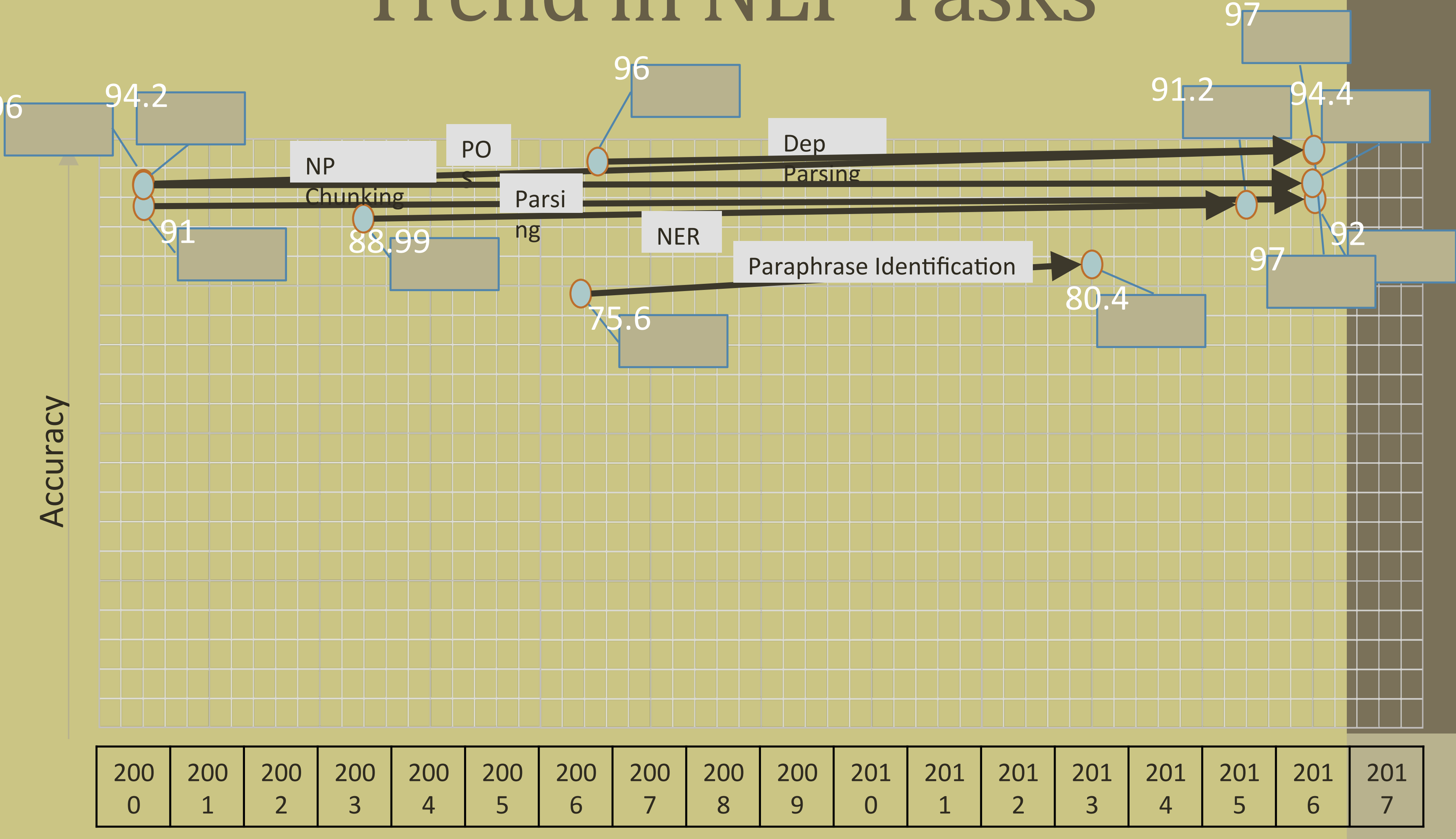
Huge leap forward in 'Speech Recognition' and 'Image Recognition'



2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017

Slide credit: Omid Bakhshandeh

Trend in NLP Tasks



Slide credit: Omid Bakhshandeh

Time to Reflect

- Your reactions to neural nets so far
  - Are they still confusing?
  - Do you need to see more?
  - Are you convinced (yet)?
  - Are they intriguing?
  - Do you want to see more?
  - Success is empirically determined: is empirical vs theoretical problematic?

What is your reaction to neural nets?

Start the presentation to activate live content

If you see this message in presentation mode, install the add-in or get help at PollEv.com/app

Synonomy and Paraphrase

- A critical piece of text interpretation
- Can be domain-specific

  - Word synonomy:
    - General domain: “hot” “sexy” but biology?

	General	Biology
hot	Warm, sexy, exciting	Heated, warm, thermal
treat	Address, handle	Cure, fight, kill
head	Leader, boss, mind	Skull, brain, cranium

Examples from Pavlick

Sentential Paraphrase

- Paraphrases extracted from different translations of the same novel

Emma burst into tears and he tried to comfort her, saying things to make her smile.	Emma cried, and he tried to console her, adorning his words with puns
And finally, dazzlingly white, it shone high above them in the empty sky.	It appeared white and dazzling, in the empty heavens.
People said "The Evening Noise is sounding, the sun is setting."	"The evening bell is ringing" people used to say.

Examples from Barzilay

Phrasal paraphrases

King's son	Son of the king
In bottles	bottled
Start to talk	Start talking
Suddenly came	Came suddenly
Make appearance	appear

Examples from Barzilay

Types of Text Similarity

- Many types of text similarity exist:
  - Morphological similarity (e.g., respect-respectful)
  - Spelling similarity (e.g., theater-theatre)
  - Synonymy (e.g., talkative-chatty)
  - Homophony (e.g., raise-raze-rays)
  - Semantic similarity (e.g., cat-tabby)
  - Sentence similarity (e.g., paraphrases)
  - Document similarity (e.g., two news stories on the same event)

Slide from Radev

Tasks requiring text similarity

- Information retrieval
- Machine translation
- Summarization
- Inference

Using word embeddings to compute similarity

- Cosine similarity

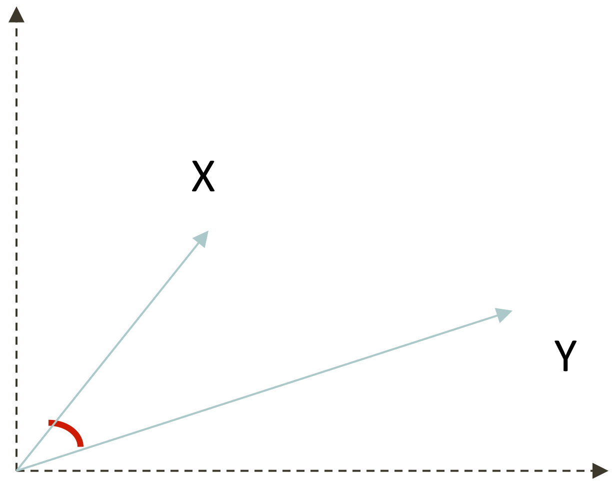
$$\mathrm{sim}_{\cos}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}$$

- When vectors have unit length, cosine similarity is the dot product
- Common to normalize embeddings matrix so that each row as unit length

Similarity Measures (Cont.)

- Cosine similarity: similarity of two vectors, normalized

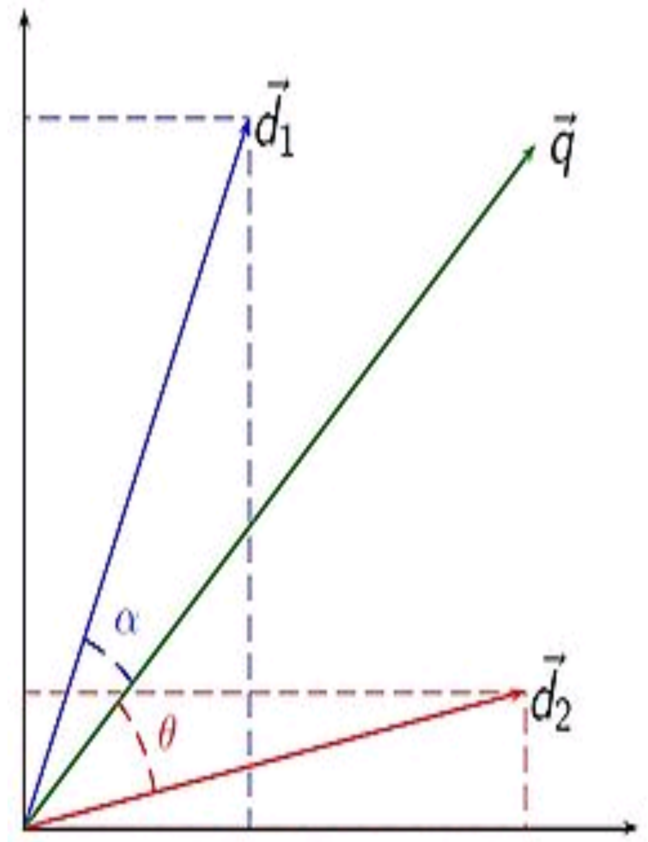
$$\cos(X,Y) = \frac{x_1 y_1 + x_2 y_2 + \ldots + x_n y_n}{\sqrt{x_1^2 + \ldots + x_n^2} \cdot \sqrt{y_1^2 + \ldots + y_n^2}} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}$$



Slide from Radev

Document Similarity

- Used in information retrieval to determine which document (d_1 or d_2) is more similar to a given query q.
- Documents and queries are represented in the same space.
- Angle (or cosine) is a proxy for similarity between two vectors



Slide from Radev

Quiz

- Given three documents

  D_1 = <1,3>

  D_2 = <10,30>

  D_3 = <3,1>
- Compute the cosine scores

  $\sigma(D_1,D_2)$

  $\sigma(D_1,D_3)$
- What do the numbers tell you?

Slide from Radev

t is the cosine similarity between D1 = and D

0

1

.5

Start the presentation to activate live content

If you see this message in presentation mode, install the add-in or get help at PollEv.com/app

What is the similarity between D1 and D3

0

1

.3

Start the presentation to activate live content

If you see this message in presentation mode, install the add-in or get help at PollEv.com/app

Quiz

- What is the range of values that the cosine scores can take?

Slide from Radev

he range of values that the cosine similarity c

[-1,1]

[0,1]

[-1,1]

Start the presentation to activate live content

If you see this message in presentation mode, install the add-in or get help at PollEv.com/app

Finding Similar Words

- $$\text{sim}_{\cos}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}$$

  Finding k most similar words where E an embedding matrix for all words.
  - $w = E_{[w]}$
  - $S = Ew$
    - A vector of similarities
    - $S_{[i]}$ = similarity of w to ith word
    - K-most similar words?
- How can we find the k-most similar words that are also orthographically similar?

can we find the k-most similar words that are orthographically similar

Start the presentation to activate live content

If you see this message in presentation mode, install the add-in or get help at PollEv.com/app

Similarity to a group of words

- Given: $w_i \ldots w_k$ that are semantically similar
- Find w_j such that it is the most semantically similar to the group
- Define similarity as average similarity to the group: $1/k\ \Sigma_{i-1}^{k}\ sim_{cos}(w,w_i)$

  $s= E(w)\ E(w_1 + w_2 + \ldots + w_k)/k$

- How would we compute odd word out?

Short Document Similarity

- We can train a model or *we can just use word embeddings*
- Suitable for very short texts such as queries, newspaper headlines or tweets
- Similarity = the sum of the pairwise similarities of all words in the document

Computing Document Similarity

- Where $D_1 = w^1_1 \ldots w^1_m$ and $D_2 = w^2_1 \ldots . w^2_n$

$$\mathrm{sim}_{\mathrm{doc}}(D_1, D_2) = \sum_{i=1}^{m} \sum_{j=1}^{n} \cos(\mathbf{w_i^1}, \mathbf{w_j^2})$$

- Equivalent to:

$$\mathrm{sim}_{\mathrm{doc}}(D_1, D_2) = (\sum_{i=1}^{m} \mathbf{w_i^1}) \cdot (\sum_{j=1}^{n} \mathbf{w_j^2})$$

- Allows: Document collection D is a matrix where each row i is a document. Similarity with a new document: $\mathbf{s} = \mathbf{D} \cdot (\sum_{i=1}^{n} \mathbf{w_i'})$

Analogy Solving Task

- $$\mathbf{w}_{\text{king}} - \mathbf{w}_{\text{man}} + \mathbf{w}_{\text{woman}} \approx \mathbf{w}_{\text{queen}}$$

$$\text{analogy}(m : w \rightarrow k :?) = \underset{v \in V \setminus \{m,w,k\}}{\text{argmax}} \cos(\mathbf{v}, \mathbf{k} - \mathbf{m} + \mathbf{w})$$

- Equivalent to (COS-ADD) (Levy and Goldberg 2014)

$$\text{analogy}(m : w \rightarrow k :?) = \underset{v \in V \setminus \{m,w,k\}}{\text{argmax}} \cos(\mathbf{v}, \mathbf{k}) - \cos(\mathbf{v}, \mathbf{m}) + \cos(\mathbf{v}, \mathbf{w})$$

- "… it is not clear what success on a benchmark of analogy tasks says about the quality of word embeddings beyond their suitability for solving this particular task." (Goldberg 2017)

Using WordNet and other paraphrase corpora

- (PPDB) Penn Paraphrase Database (Pavlick and Callison-Burch)
- Can we use word pairs that reflect similarity better for the task?
  - Pre-trained embeddings E
  - Graph G representing similar word pairs
  - Search for a new word embedding matrix E' whose rows are close to E but also close to G
- Methods for combining pre-trained word embeddings with smaller, specialized embeddings

Caveats

- Don't just use off-the-shelf word embeddings blindly
- Experiment with corpus and hyper-parameter settings
- When using off-the-shelf embeddings, use the same tokenization and normalization

Resources

- Word embeddings
  - https://code.google.com/p/word2vec/
  - http://nlp.stanford.edu/projects/glove/images/comparative_superlative.jpg
- Neural net platforms
  - Kerras https://keras.io/
  - Pytorch http://pytorch.org/
  - Tensorflow https://www.tensorflow.org/
  - Theano http://deeplearning.net/software/theano/

Language is made up of sequences

- So far we have seen embeddings for words
  - (and methods for combining through vector concatenation and arithmetic)

- But how can we account for sequences?
  - Words as sequences of letters
  - Sentences as sequences of words
  - Documents as sequences of sentences

Recurrent Neural Networks

- Represent arbitrarily sized sequences in fixed-size vector

- Good at capturing statistical regularities in sequences (order matters)

- Include simple RNNs, Long short-term memory (LSTMs), Gated Recurrent Unit (GRUs)

Learning word meaning from their morphs

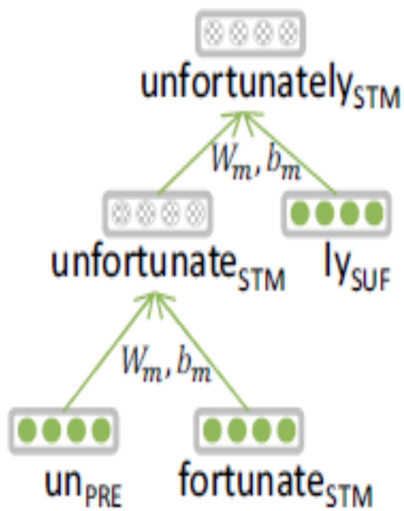


Figure 1: **Morphological Recursive Neural Network.** A vector representation for the word "unfortunately" is constructed from morphemic vectors: $un_{\texttt{pre}}$, $fortunate_{\texttt{stm}}$, $ly_{\texttt{suf}}$. Dotted nodes are computed on-the-fly and not in the lexicon.

[Thang et al. 2013]

Logical entailment using compositional semantics via RNNs

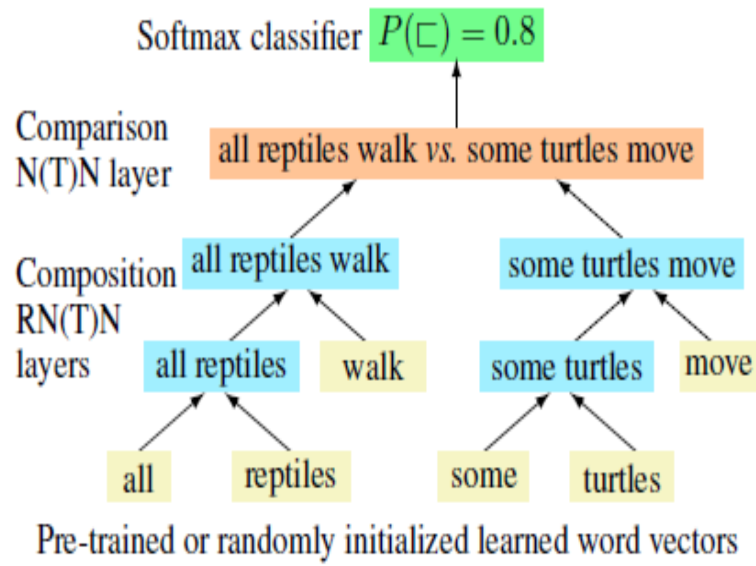


Figure 1: In our model, two separate tree-structured networks build up vector representations for each of two sentences using either NN or NTN layer functions. A comparison layer then uses the resulting vectors to produce features for a classifier.

[Bowman et al. 2014]

Machine Translation (Sequences)

- Sequence-to-sequence
  - Sutskever et al. 2014

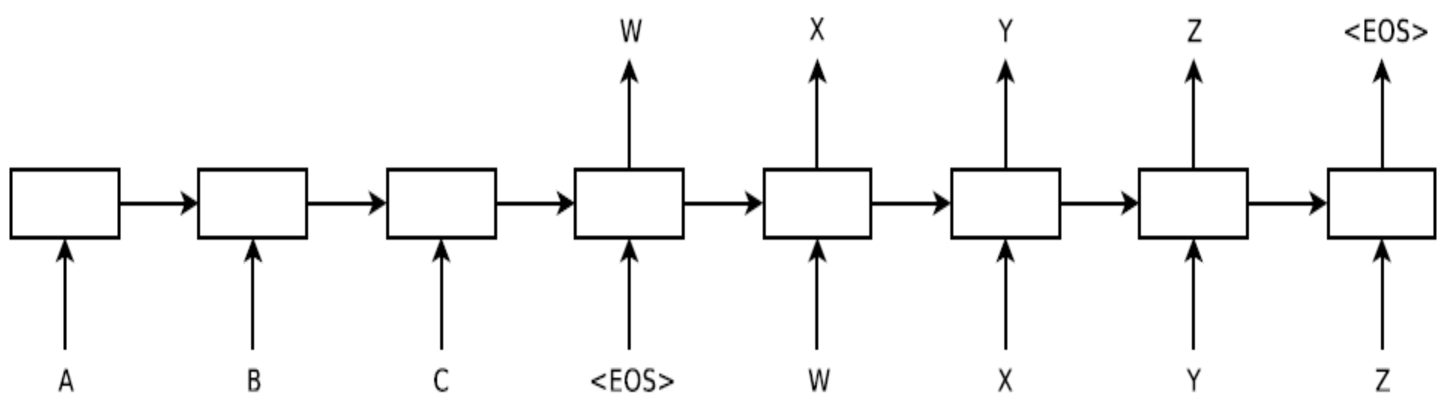


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

RNN Abstraction

- RNN is a function that takes an arbitrary length sequence as input and returns a single d_{out} dimensional vector as output
  - Input: $x_{1:n} = x_1\ x_{2\ \dots}\ x_n$ (x_i ε $R^{d\text{-}in}$)
  - Output: y_n ε $R^{d\text{-}out}$

$$\mathbf{y_n} = \mathrm{RNN}(\mathbf{x_{1:n}})$$

$$\mathbf{x_i} \in \mathbb{R}^{d_{\mathrm{in}}} \quad \mathbf{y_n} \in \mathbb{R}^{d_{\mathrm{out}}}$$

$$\mathbf{y_{1:n}} = \mathrm{RNN}^{\star}(\mathbf{x_{1:n}})$$

$$\mathbf{y_i} = \mathrm{RNN}(\mathbf{x}_{1:i})$$

$$\mathbf{x_i} \in \mathbb{R}^{d_{\mathrm{in}}} \quad \mathbf{y_i} \in \mathbb{R}^{d_{\mathrm{out}}}$$

Output vector y used for further prediction

RNN Characteristics

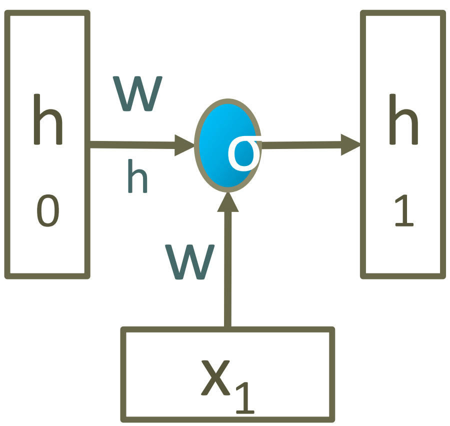
- Can condition on the entire sequence without resorting to the Markov assumption
- Can get very good language models as well as good performance on many other tasks

RNNs are defined recursively

- By means of a function R taking as input a state vector h_{i-1} and an input vector x_i
- Returns a new state vector h_i
- The state vector can be mapped to an output vector y_i using a simple deterministic function
- And fed through softmax for classification.

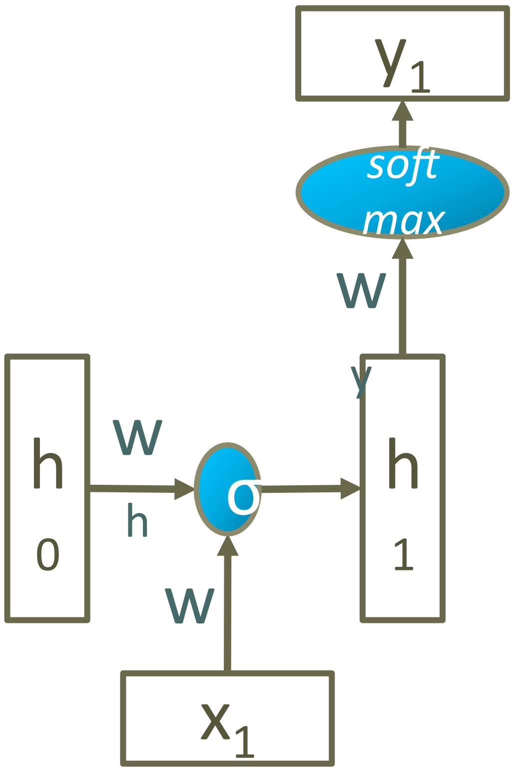
Recurrent Neural Networks

$$h_t = \sigma(W_h h_{t-1} + W_x x_t)$$



Slide from Radev

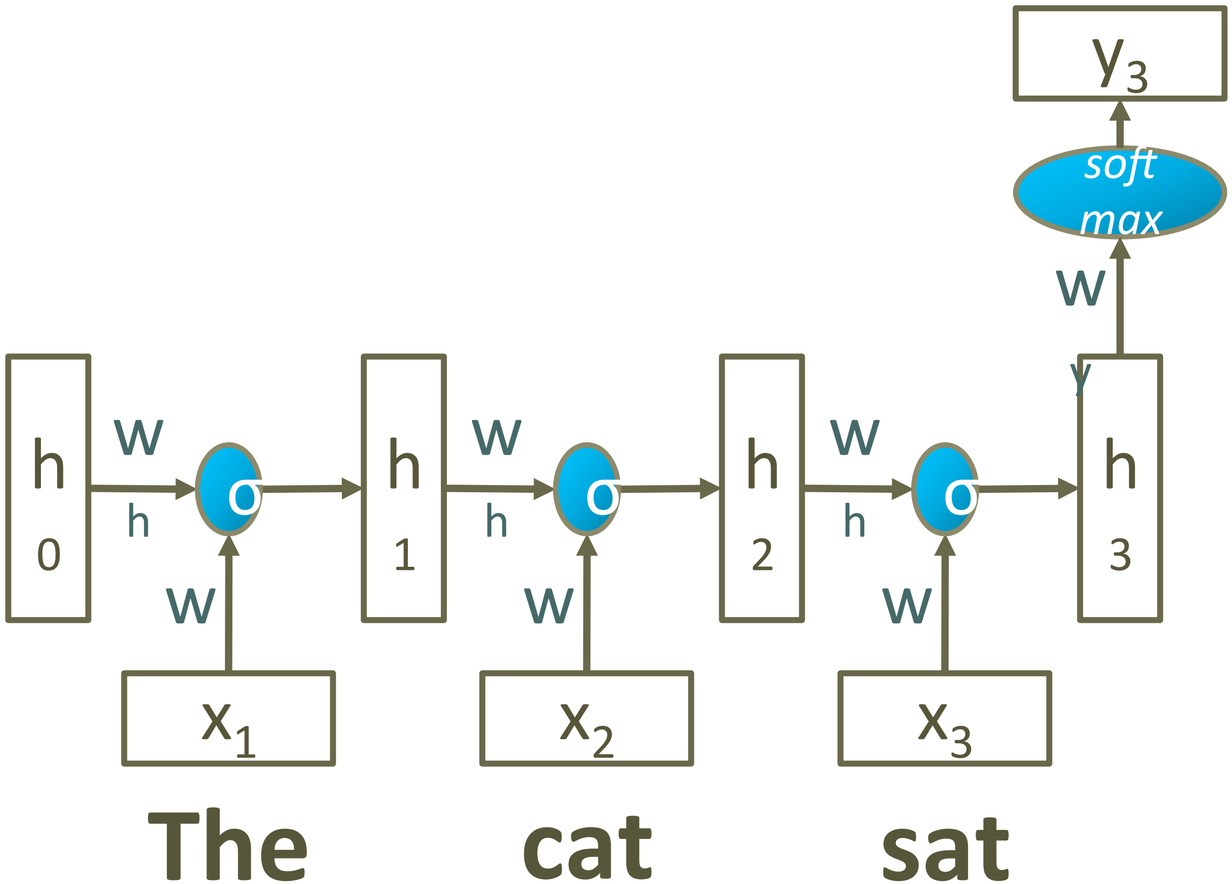
RNN



$$h_t = \sigma(W_h h_{t-1} + W_x x_t)$$
$$y_t = softmax(W_y h_t)$$

Slide from Radev

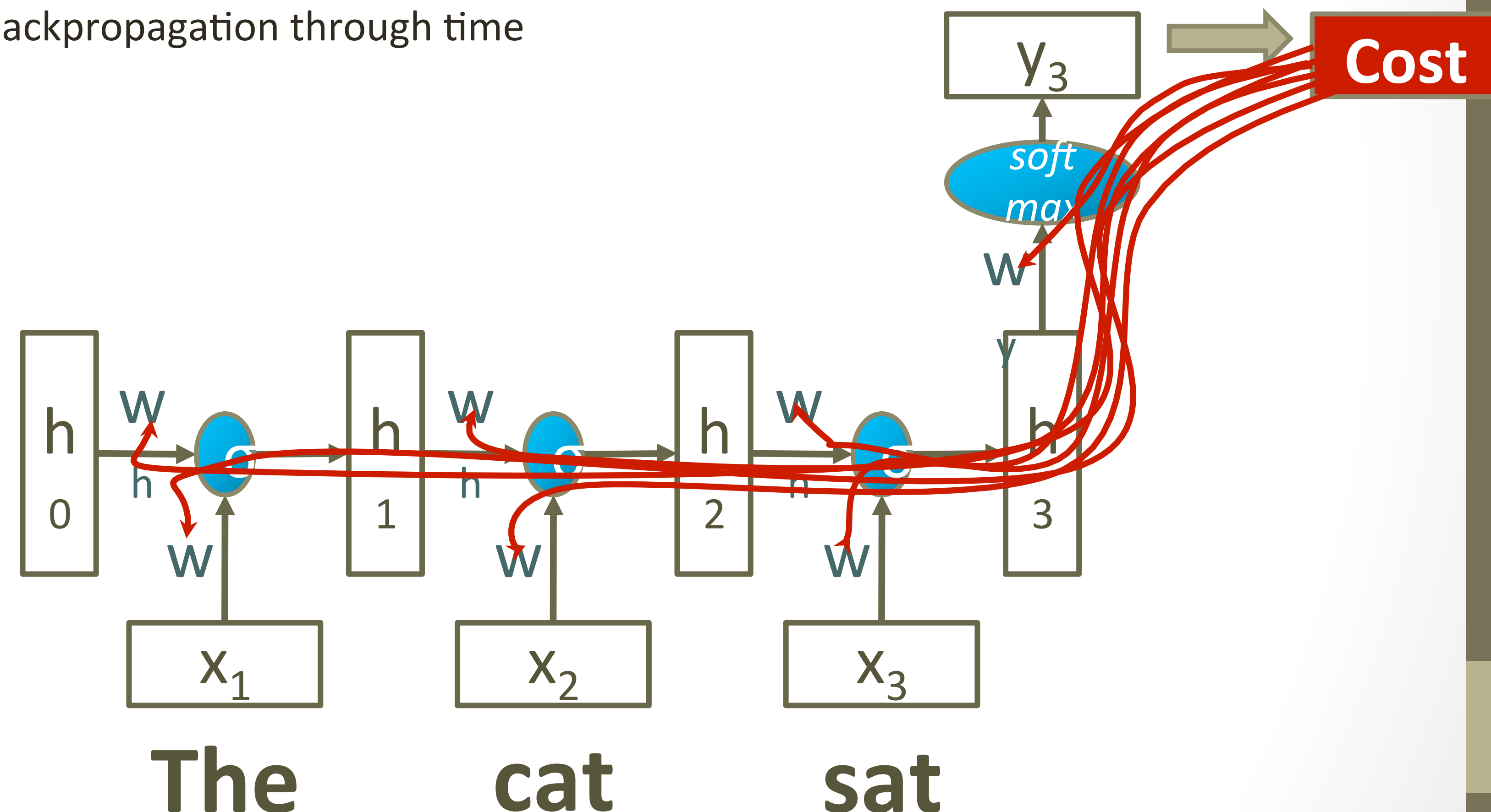
RNN



Slide from Radev

Updating Parameters of an RNN

Backpropagation through time



Slide from Radev

Next Time

- More on RNNs and their use in sentiment analysis