Natural Language Processing

Soboleva Natalia

Neural Language Models

Language Models

Word ordering: $P(\text{the cat is small}) > P(\text{small the is cat})$

Word choice: $P(\text{walking home after school}) > P(\text{walking house after school})$

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i | w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i | w_{i-(n-1)}, \ldots, w_{i-1})$$

$\{w_1, \ldots, w_m\}$ – sequence of m words

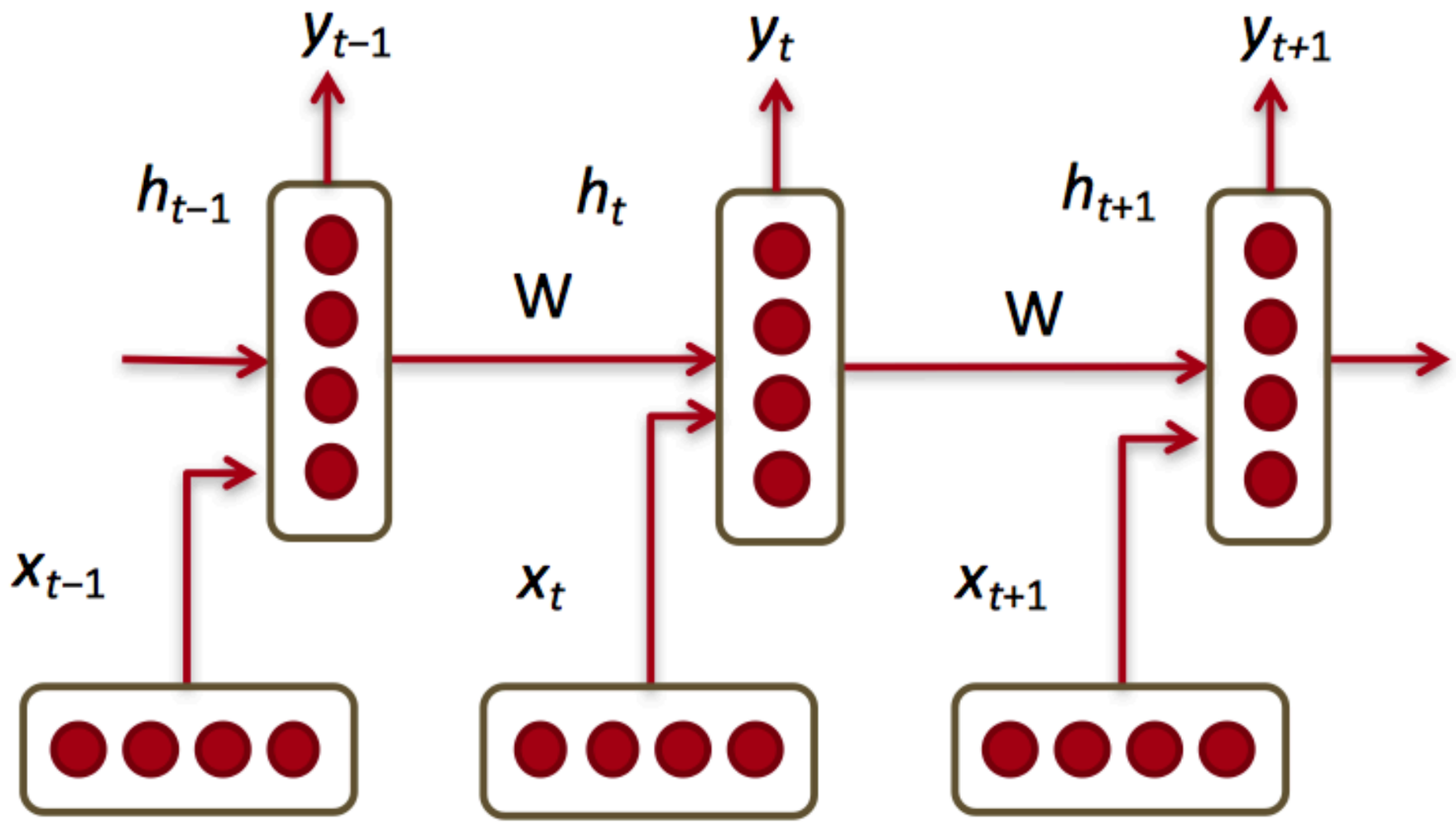
$$P(w_2|w_1) = \frac{count(w_1, w_2)}{count(w_1)}$$

$$P(w_3|w_1, w_2) = \frac{count(w_1, w_2, w_3)}{count(w_1, w_2)}$$

What is the main problem with this approach?

Next obvious step is…

Recurrent Neural Networks



Basic RNN. Image Source: http://cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf

Simple RNN

$x_1, \ldots, x_{t-1}, x_t, x_{t+1}, \ldots, x_T$ – the word vectors

$$h_t = \sigma(W^{(hh)} h_{t-1} + W^{(hx)} x_{[t]})$$

$$\hat{y}_t = softmax(W^{(S)} h_t)$$

$x_t \in \mathbb{R}^d$ – input word vector at time t
$x_{[t]}$ – column vector of L at index $[t]$ at time step t
$W^{hx} \in \mathbb{R}^{Dh \times d}$ – weights matrix used to condition x_t
$W^{hh} \in \mathbb{R}^{Dh \times Dh}$ – weights matrix used to condition h_{t-1}
$W^{(S)} \in \mathbb{R}^{|V| \times Dh}$ – network's weights matrix
$h_{t-1} \in \mathbb{R}^{Dh}$ – output of the non-linear function at the previous time-step
$h_0 \in \mathbb{R}^{Dh}$ – initialization vector for the hidden layer at time-step $t = 0$
σ – the non-linearity function (sigmoid here)

How Do We Train It?

$$\hat{P}(x_{t+1} = v_j \mid x_t, \ldots, x_1) \quad = \quad \hat{y}_{t,j}$$

- Probability distribution over the vocabulary: $\hat{y} \in \mathbb{R}^{|V|}$

- Cross entropy loss function but predicting words instead of classes:

$$J^{(t)}(\theta) = -\sum_{j=1}^{|V|} y_{t,j} \log \hat{y}_{t,j}$$

- Evaluation could just be negative of average log probability over dataset of size (number of words) T:

$$J = -\frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{|V|} y_{t,j} \log \hat{y}_{t,j}$$

Vanishing Gradient & Gradient Explosion Problems

Sentence 1

«Jane walked into the room. John walked in too. Jane said hi to **John** »

Sentence 2

"Jane walked into the room. John walked in too. It was late in the day, and everyone was walking home after a long day at work. Jane said hi to _____»

Who was that again?

Sentence 3

«Coldplay gave a really memorable performance at the Rose Bowl last night. It was a sell out *…random…text…something…something…*

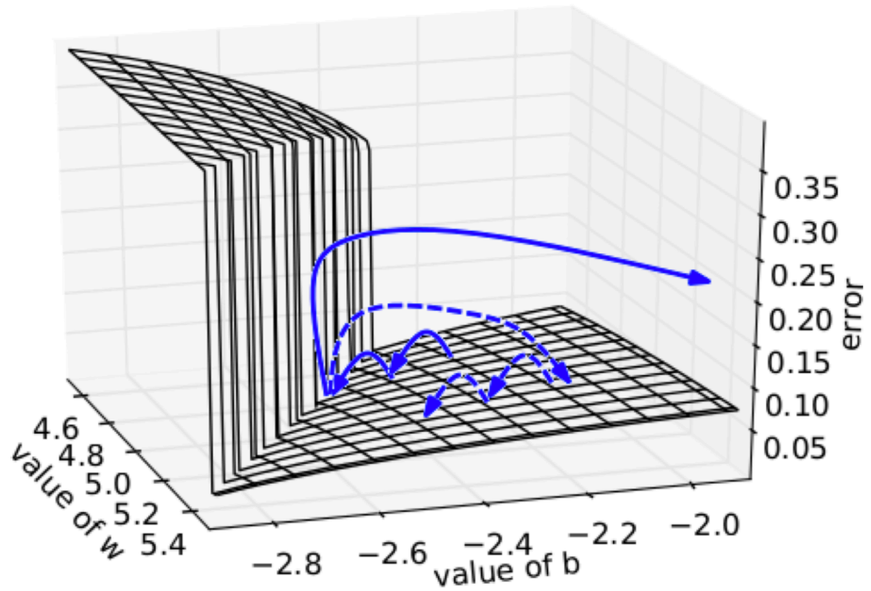
…think…of…it…as…a…really… long…text…

…almost…the…end…of…the…story…almost…there…and…finally…it…ends!»

Who played at the concert?

Solution to the Exploding & Vanishing Gradients

- Gradient clipping

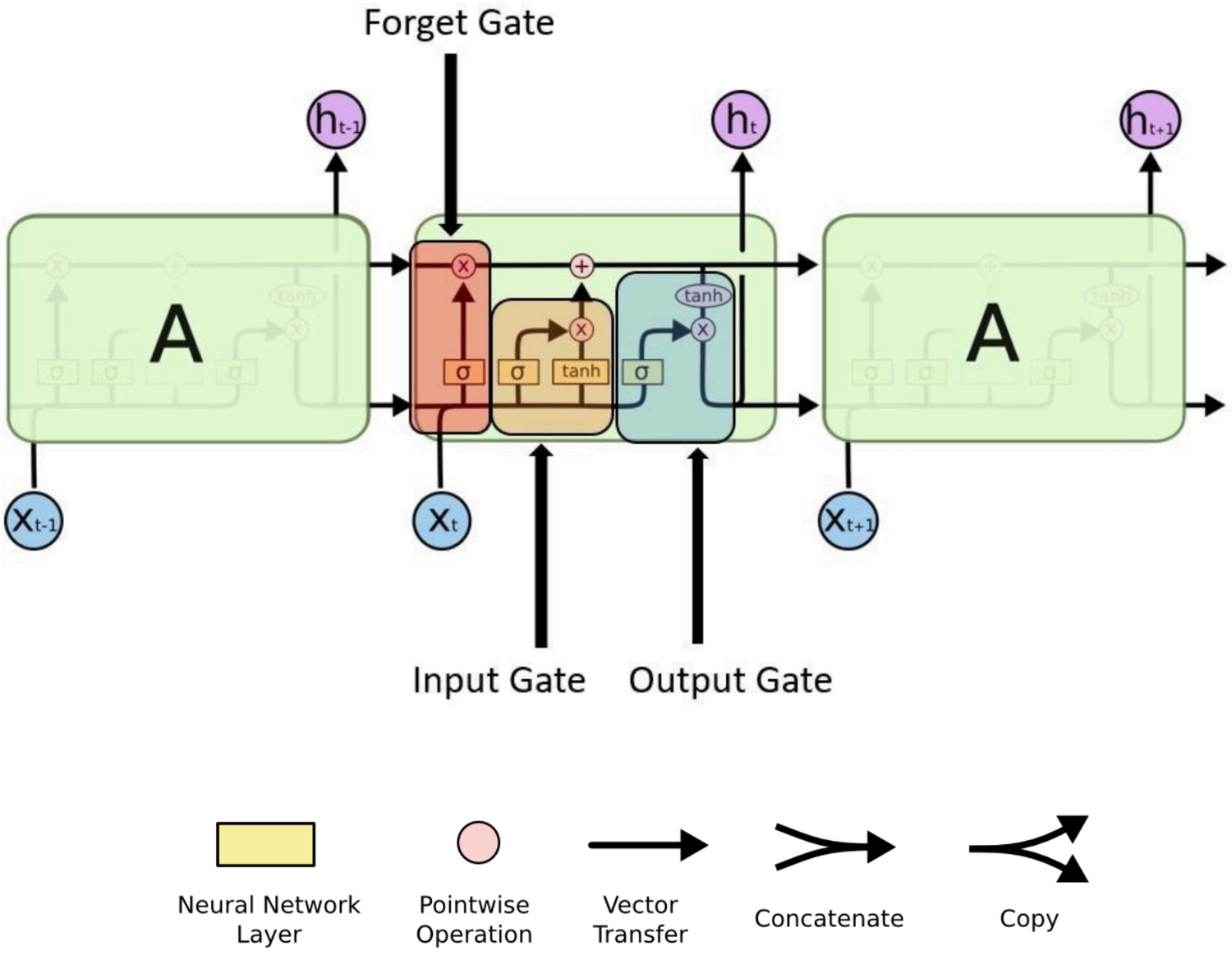


Gradient Clipping RNN. Image Source: http://arxiv.org/pdf/1211.5063.pdf

- Instead of initializing $W^{(hh)}$ randomly, start off from an identify matrix initialization

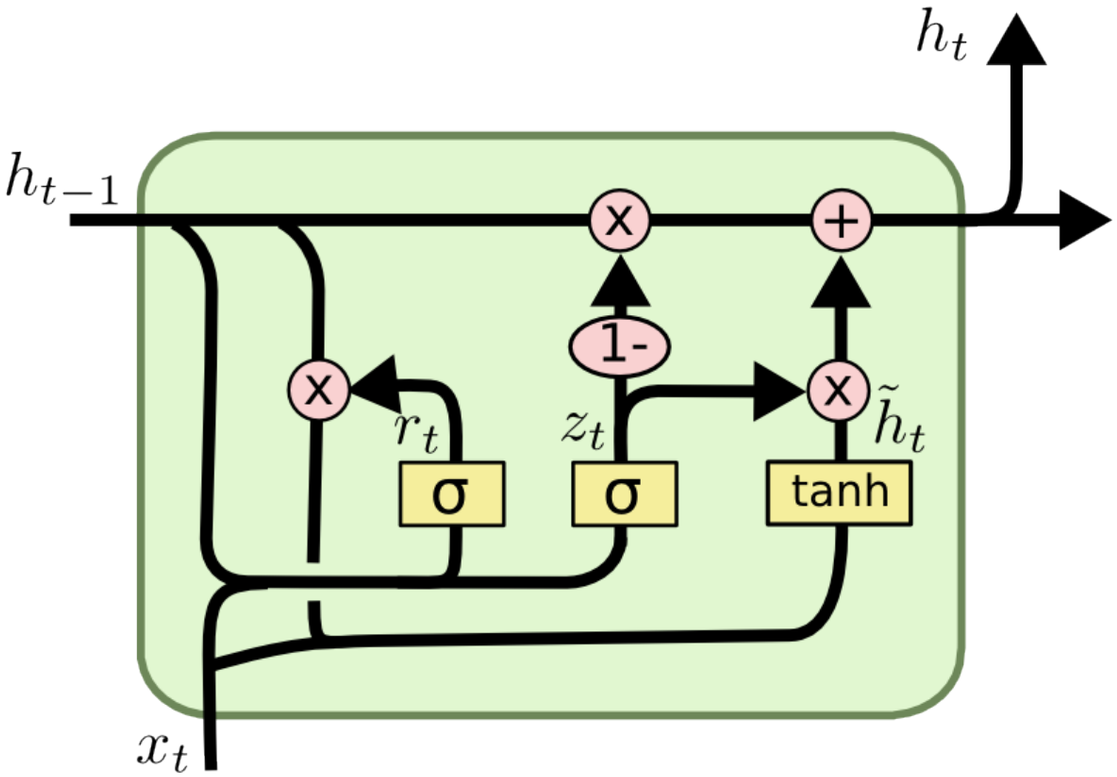
- Use the Rectified Linear Units (ReLU) instead of the sigmoid function

Long-short-term-memory (LSTM)



3- node LSTM model. Image Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Units (GRU)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

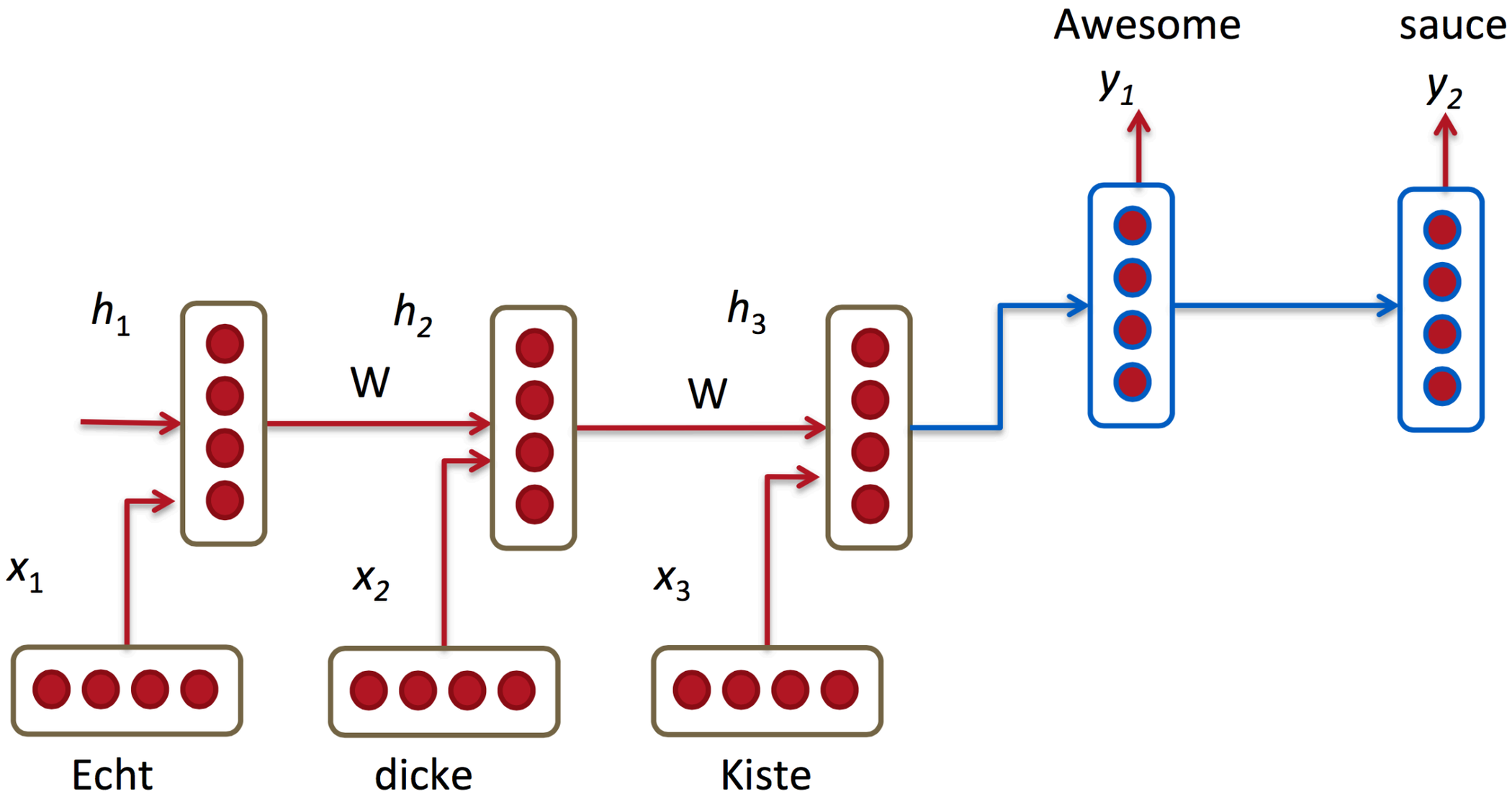
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

GRU model. Image Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

- Keep around memories to capture long distance dependencies
- Allow error messages to flow at different strengths depending on the inputs

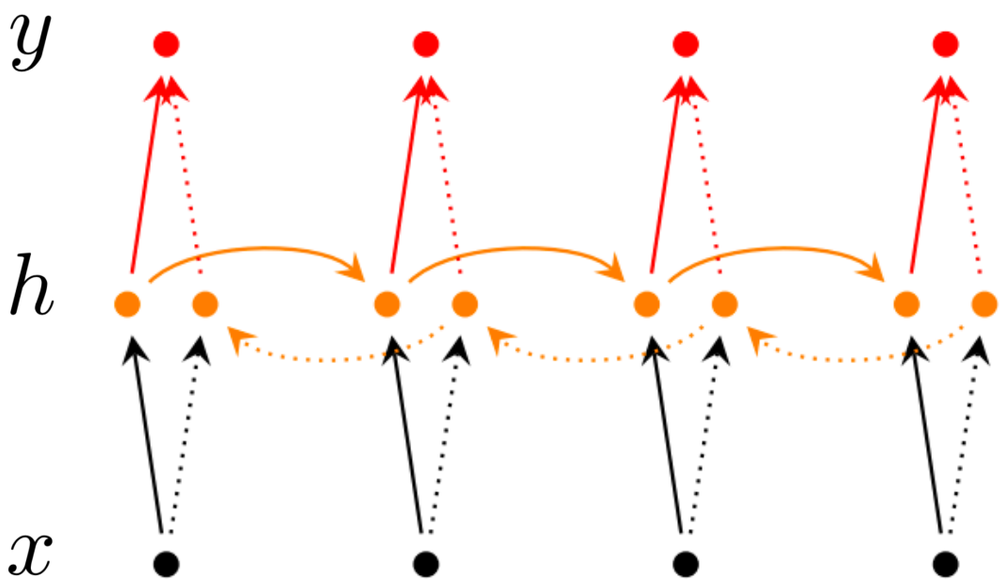
RNN for Machine Translation



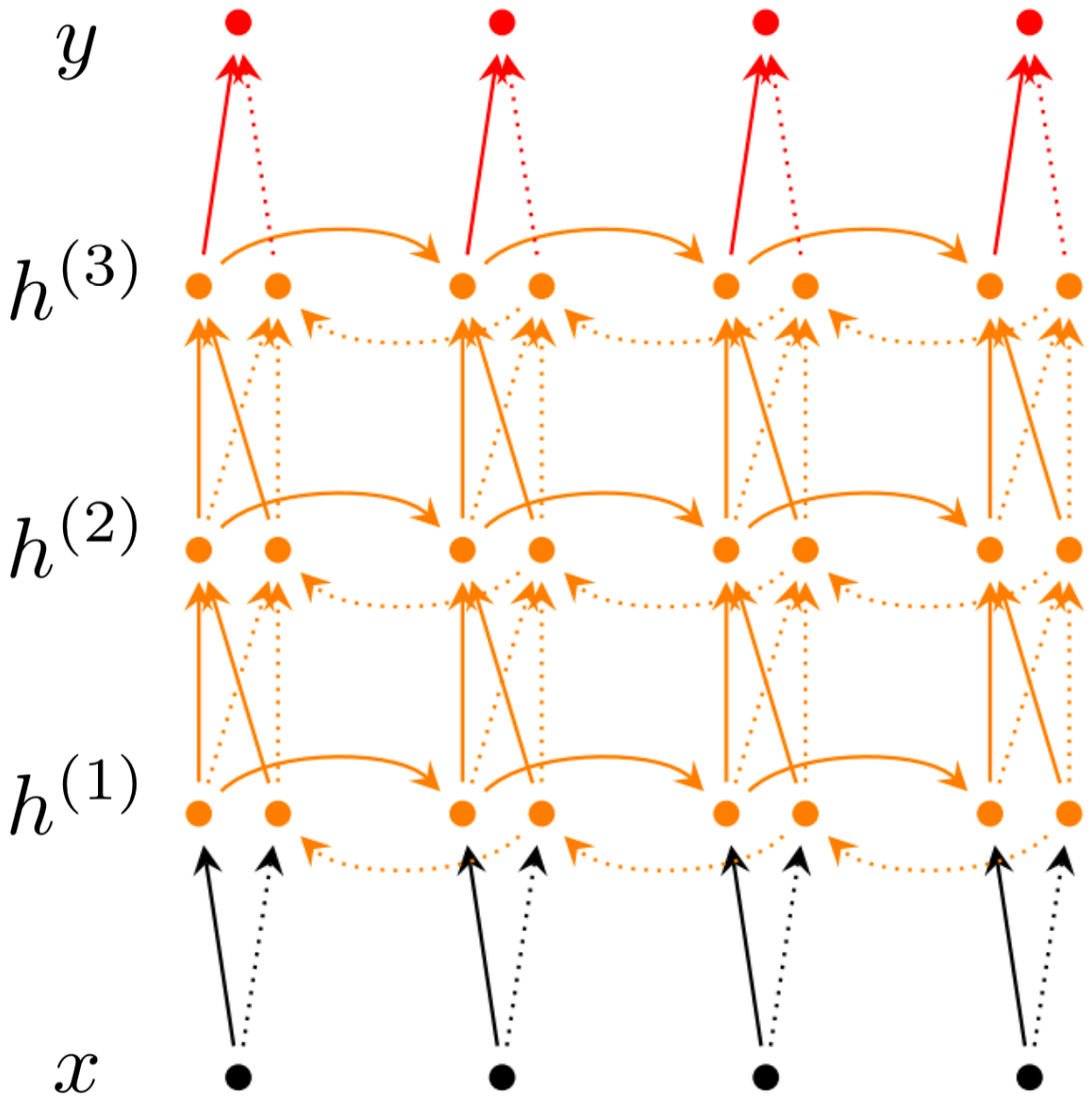
RNN for Machine Translation. Image Source: http://cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf

What’s the problem with going only forward?

Bidirectional RNNs



A bi-directional RNN model. Image Source: http://cs224d.stanford.edu/lecture_notes/notes4.pdf



A deep bi-directional RNN model with three RNN layers . Image Source: http://cs224d.stanford.edu/lecture_notes/notes4.pdf

Tasks for RNNs

- **Word-level classification:**
    1. bidirectional LSTM for NER

        $[\text{Jim}]_{\text{Person}}$ bought 300 shares of $[\text{Acme Corp.}]_{\text{Organization}}$ in $[\text{2006}]_{\text{Time}}$.

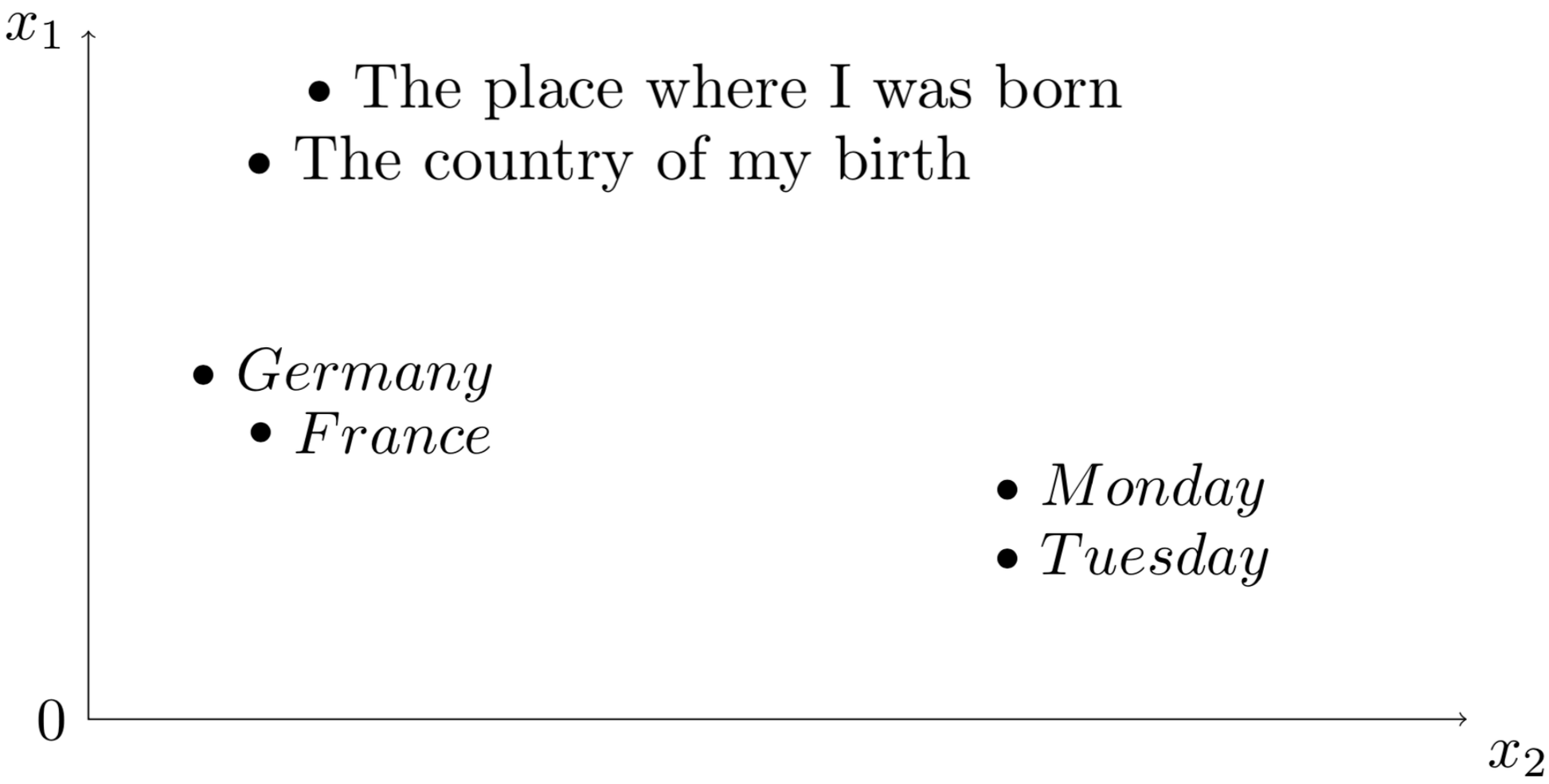
    3. modeling complex sequences lead to MT
    4. character- based representations
- **Sentence-level classification:**
    1. tweets with LSTM
    2. Dual-LSTM for semantic matching between texts
- **Generating language:**
    1. Deep LSTMs have been shown to generated reasonable task-specific text
    2. machine translation, QA, dialog systems

Are Languages Recursive?

[The man] + [from the company] + [that] + [you] + [spoke with] + [about the project] + [yesterday]

[The man from [the company that you spoke with about [the project] yesterday]

Recursive Neural Networks

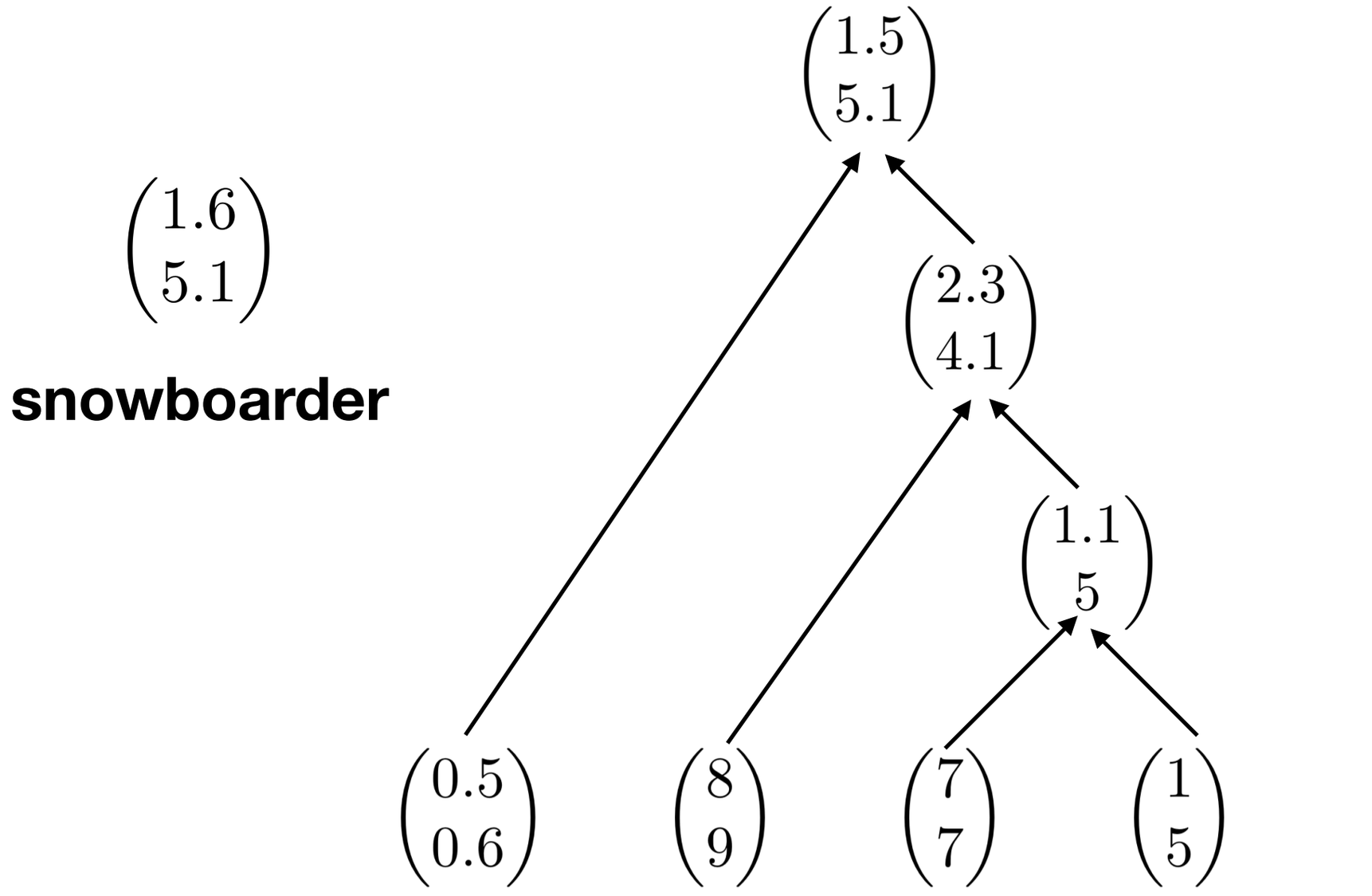


Principle of compositionality

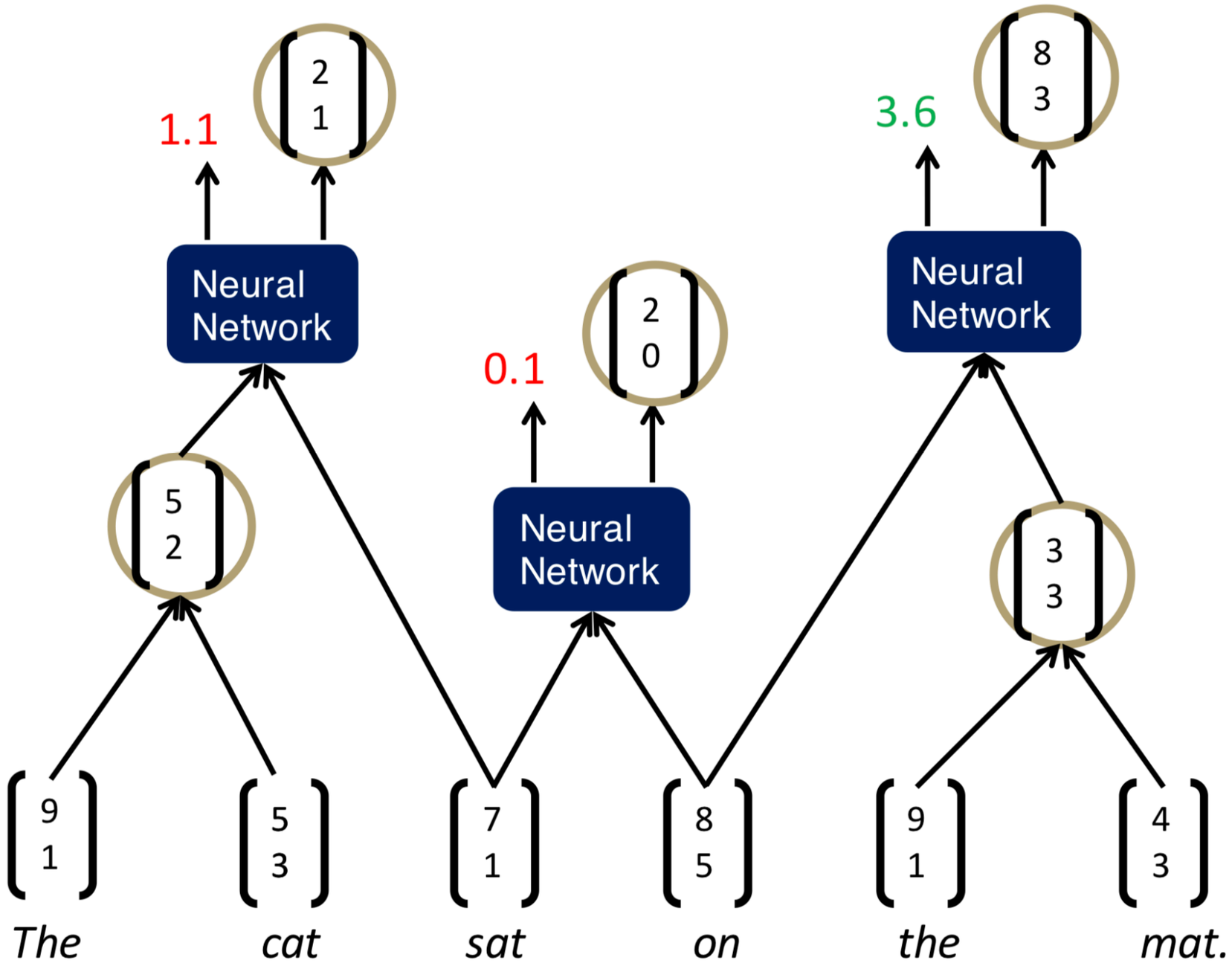
- The meanings of the words
- The rules that combine them

*The **snowboarder** is leaping over a mogul*

*A **person on a snowboard** jumps into the air*

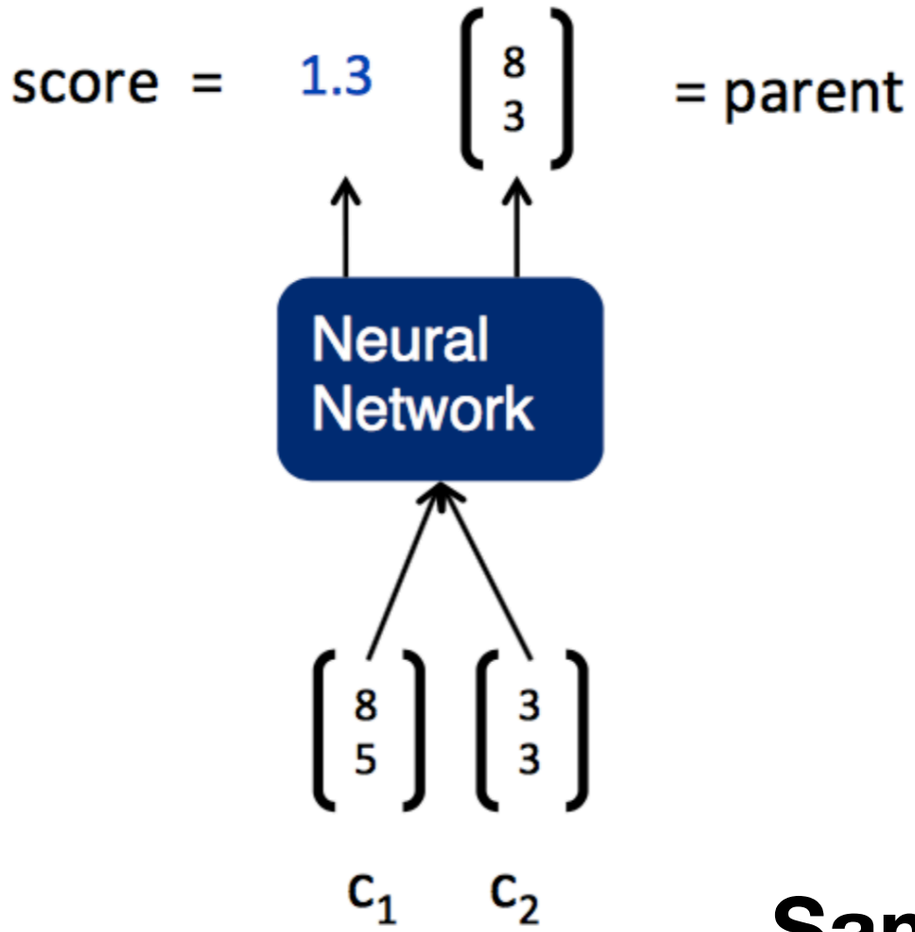


Let's Parse The Sentence



Recursive neural network example. Image Source: http://cs224d.stanford.edu/lectures/CS224d-Lecture10.pdf

How does it work?



Recursive neural network example. Image Source:http://cs224d.stanford.edu/lectures/CS224d-Lecture10.pdf

$$score = U^T h$$

$$h = \tanh\left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right)$$

$$W \in \mathbb{R}^{d \times 2d}, b \in \mathbb{R}^d, h \in \mathbb{R}^d$$

Same W parameters at all nodes of the tree

Max-Margin Framework

x – sentence, y – parse tree

$$s(x, y) = \sum_{n \in nodes(y)} score_n$$

$$J = \sum_i s(x_i, y_i) - \max_{y \in A(x_i)} (s(x_i, y) + \delta(y, y_i))$$

The loss $\delta(y, y_i)$ penalizes all incorrect decisions

Similar to max-margin parsing, a supervised max-margin objective

Backpropagation Through Structure (BPTS)

Three differences resulting from the recursion and tree structure:

1. Sum derivatives of W from all nodes

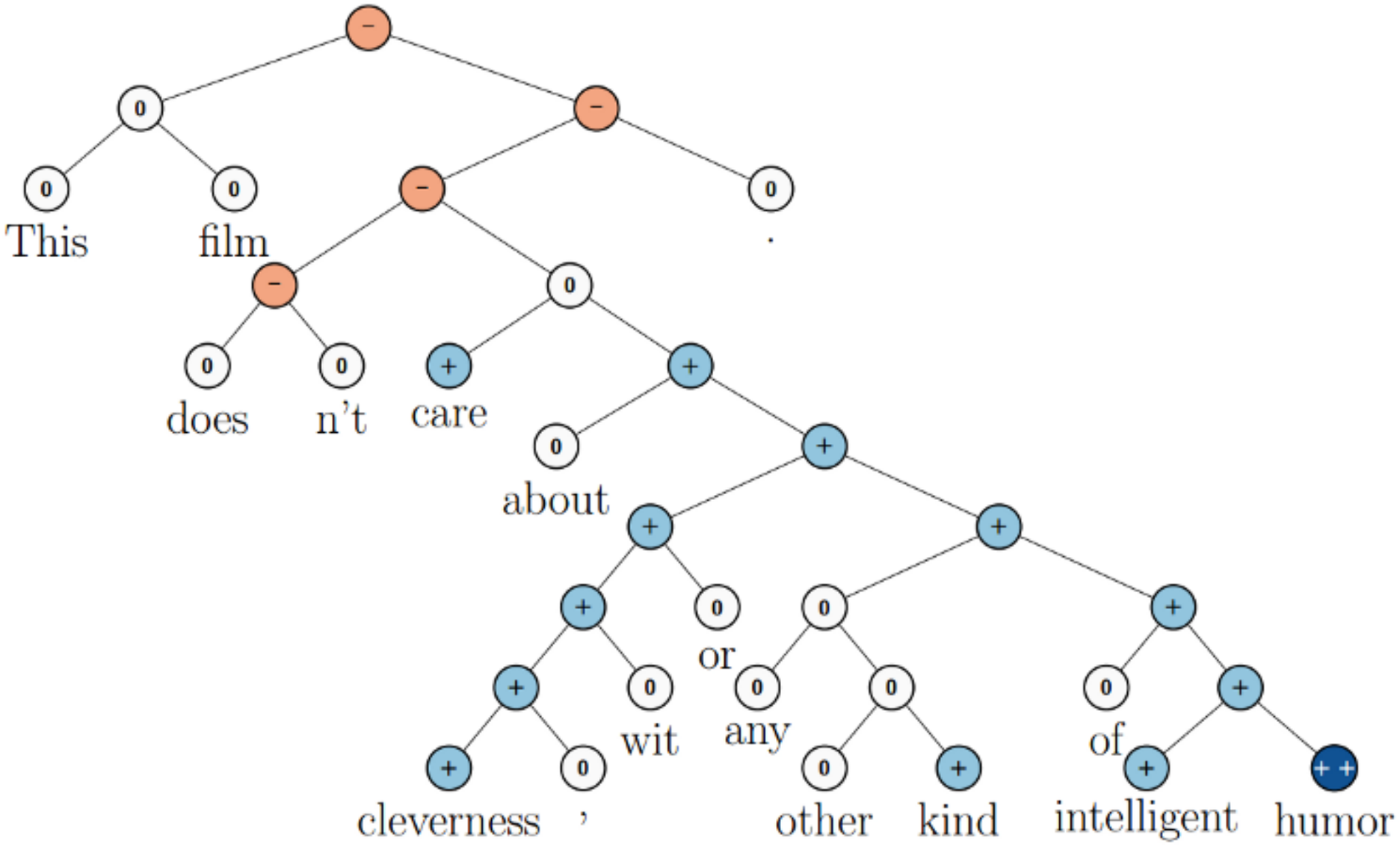
2. Split derivatives at each node

3. Add error messages from parent + node itself

$$\delta^{(l)} = \left((W^{(l)})^T \delta^{(l+1)}\right) \circ f'(z^{(l)}),$$

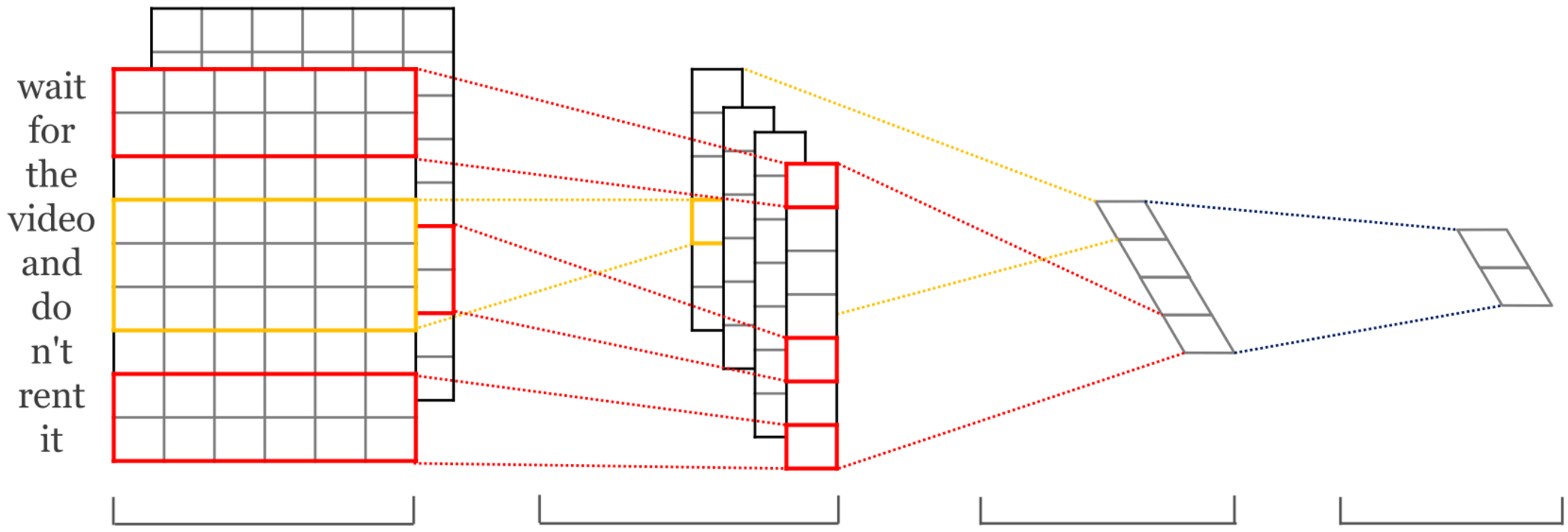
$$\frac{\partial}{\partial W^{(l)}} E_R = \delta^{(l+1)} (a^{(l)})^T + \lambda W^{(l)}$$

RNTNs (Recursive Neural Tensor Network)



Sentiment Trees. Image Source: http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

Convolutional Neural Networks



CNN structure. Source: Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_n$$

$x_{1:n} \in \mathbb{R}^k$ – the k-dimensional word vector
n – length of the sentence

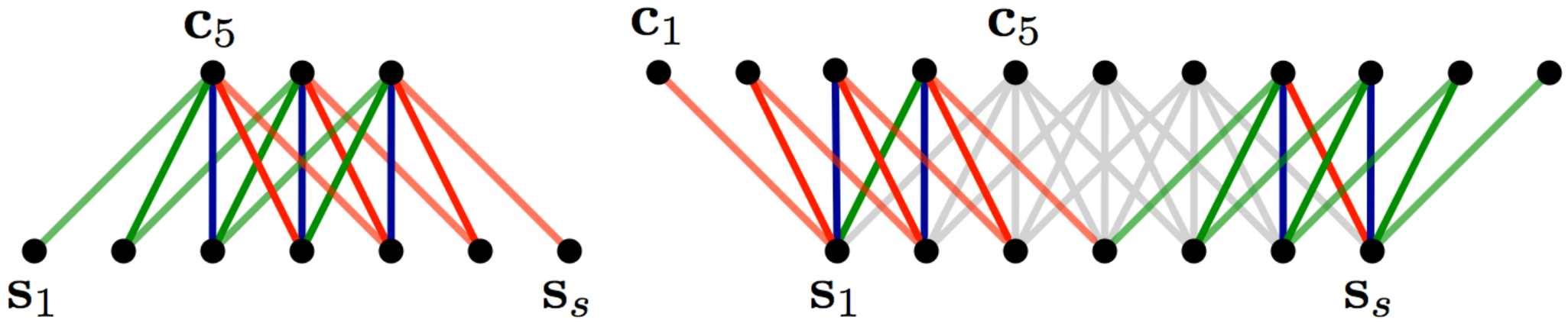
$$c_i = f(\mathbf{w}^T \mathbf{x}_{i:i+h-1} + b)$$

$w \in \mathbb{R}^{hk}$ – filter
c_i – new feature

$$\mathbf{c} = [c_1, c_2, \ldots, c_{n-h+1}]$$

feature map

Narrow vs. Wide convolution

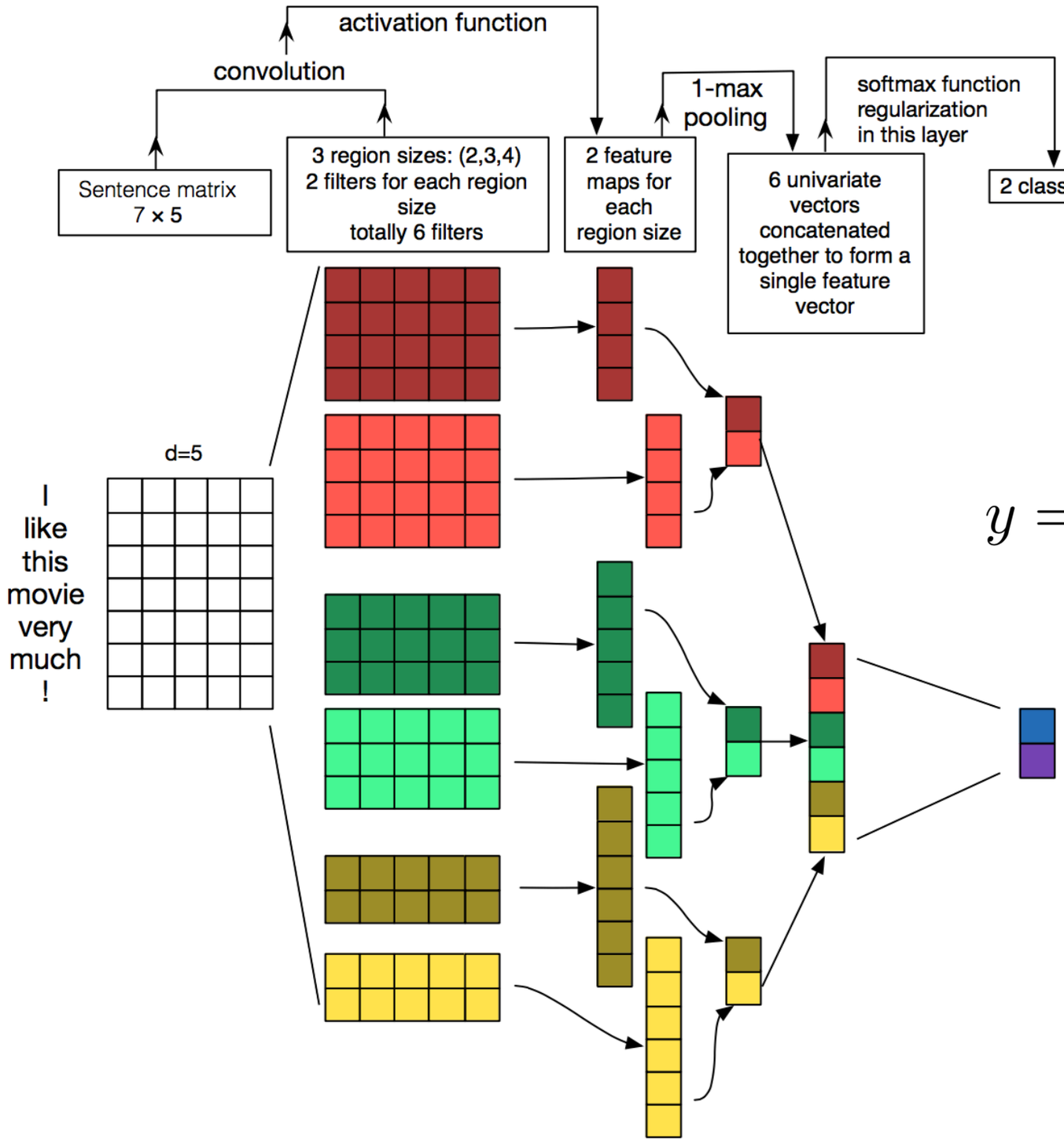


Narrow vs. Wide Convolution. Filter size 5, input size 7. Source: A Convolutional Neural Network for Modelling Sentences (2014)

Pooling strategy

$$\hat{c} = \max\{c\}$$

1. Fixed-length output, required for classification
2. Reduces the output's dimensionality while keeping the most salient n-gram features across the whole sentence



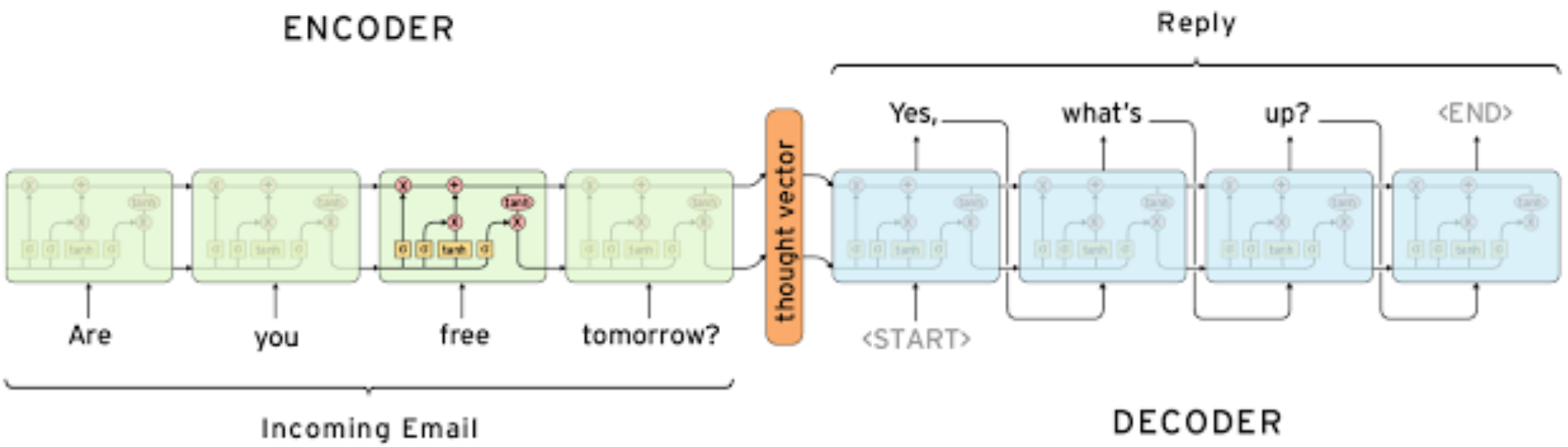
$$\mathbf{z} = [\hat{c}_1, \ldots, \hat{c}_m]$$

$$y = softmax\left(W^{(s)}\mathbf{z} + b\right)$$

Illustration of a Convolutional Neural Network (CNN) architecture for sentence classification. Source: Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification.

Deep Reinforcement Learning

Deep Reinforcement Learning for Dialogue Generation



LSTM for dialogue generation. Source: http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/

4-layer LSTM encoder-decoder

A: how old are you?

B: I'm 16.

A: 16?

B: I don't know what you are talking about.

A: You don't know what you are saying.

B: I don't know what you are talking about.

A: You don't know what you are saying.

Probable answer != Good answer

Which Answer Is Good?

- Reasonable:

  $P(response|input)$ is high according to the $seq2seq$ model

- Non-repetitive:

  similarity between response and previous messages is low

- Easy to answer

  $P(IDK|response)$ is low

Scoring function:

$$R(response) = reasonable\_score + non\_repetitive\_score + easy\_to\_answer\_score$$

Add Reinforcement Learning

A: how old are you?

B: I'm 16. Why are you asking?

A: I thought you were 12.

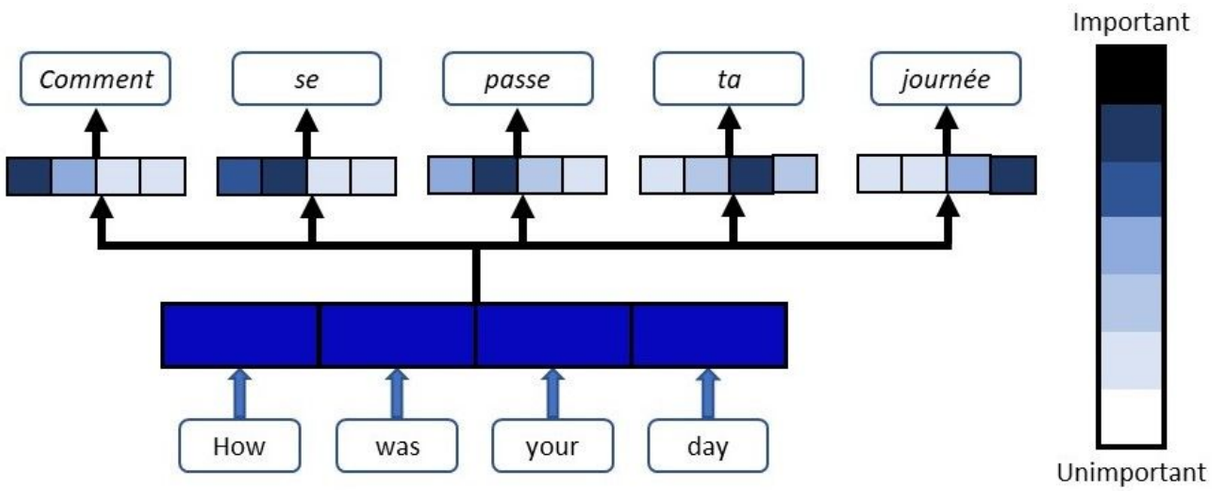
B: What made you think so?

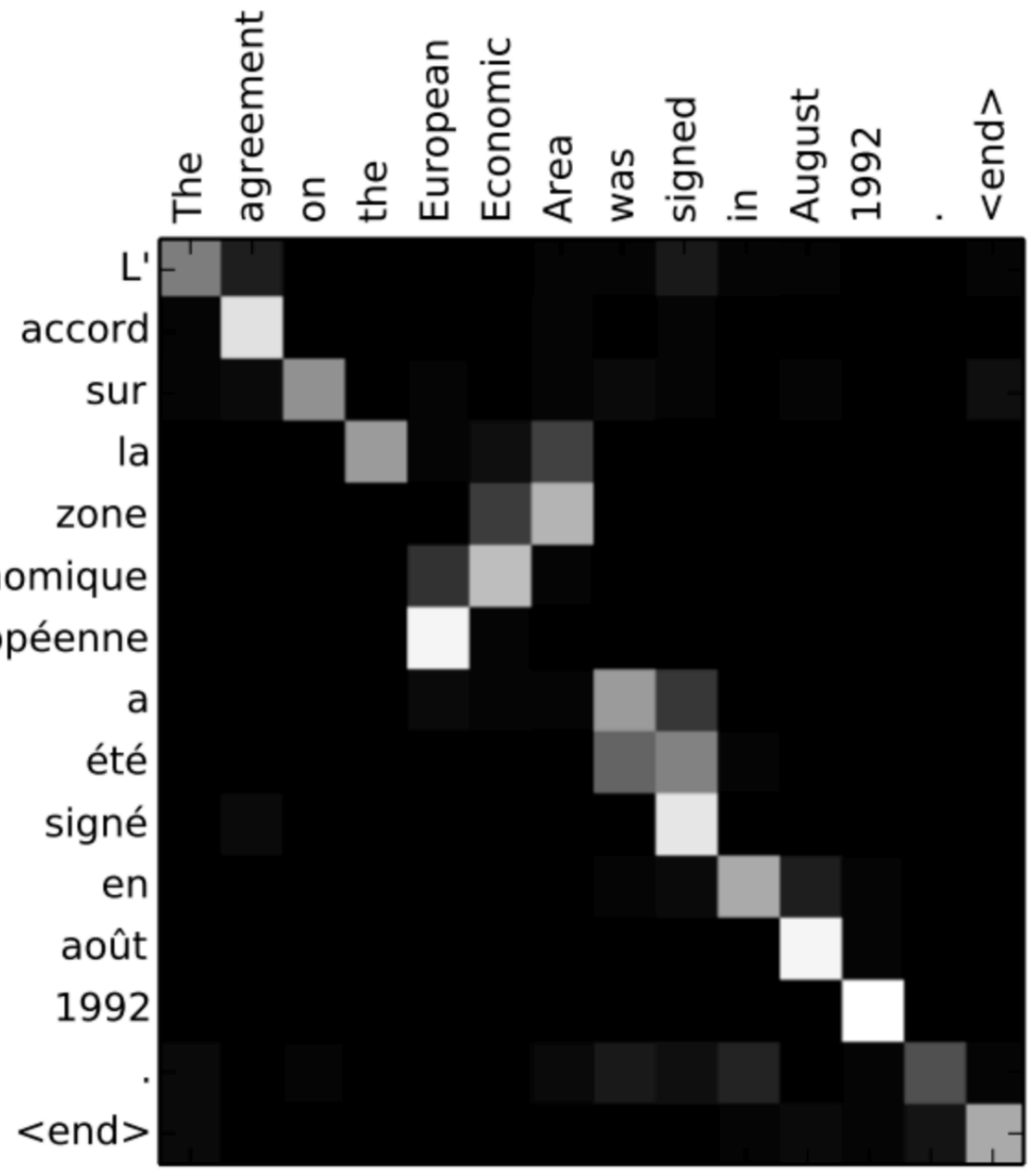
A: I don't know what you are talking about.

B: You don't know what you are saying.

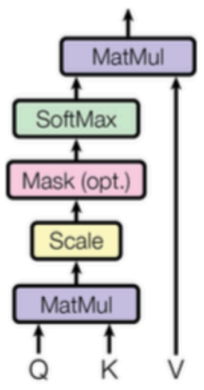
Attention Mechanism

Attention Mechanism

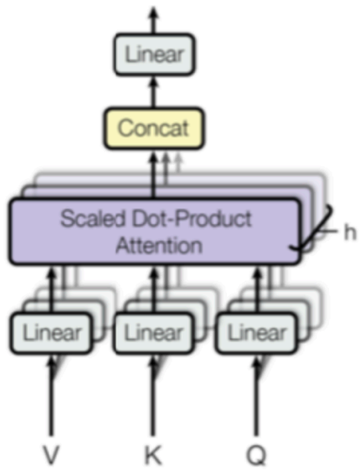




Scaled Dot-Product Attention

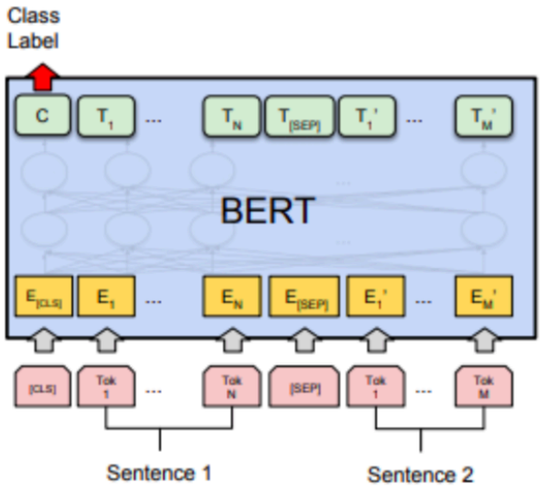


Multi-Head Attention

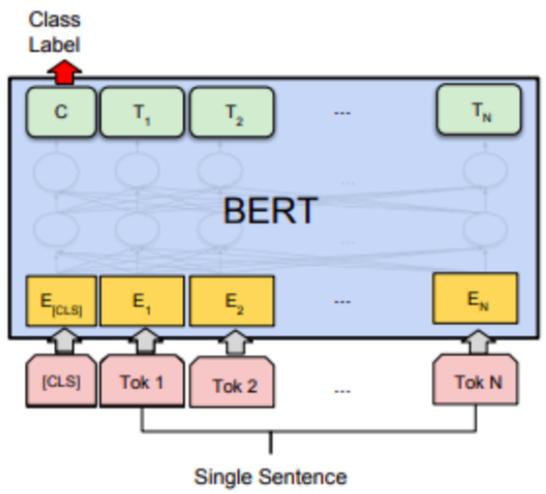


The BERT model

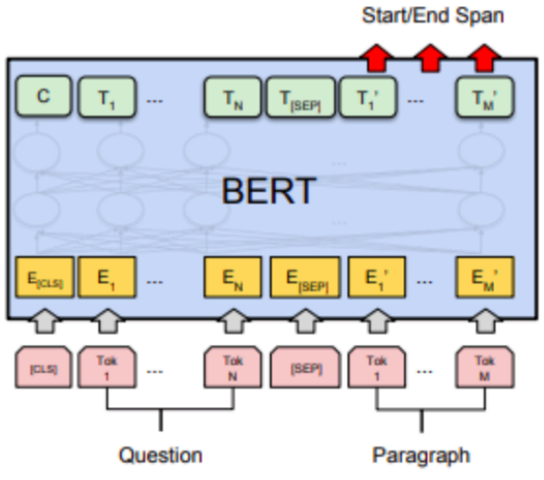
Bidirectional Encoder Representations from Transformers



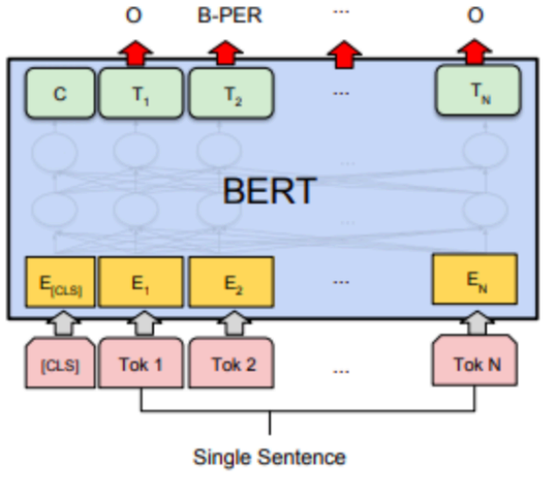
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



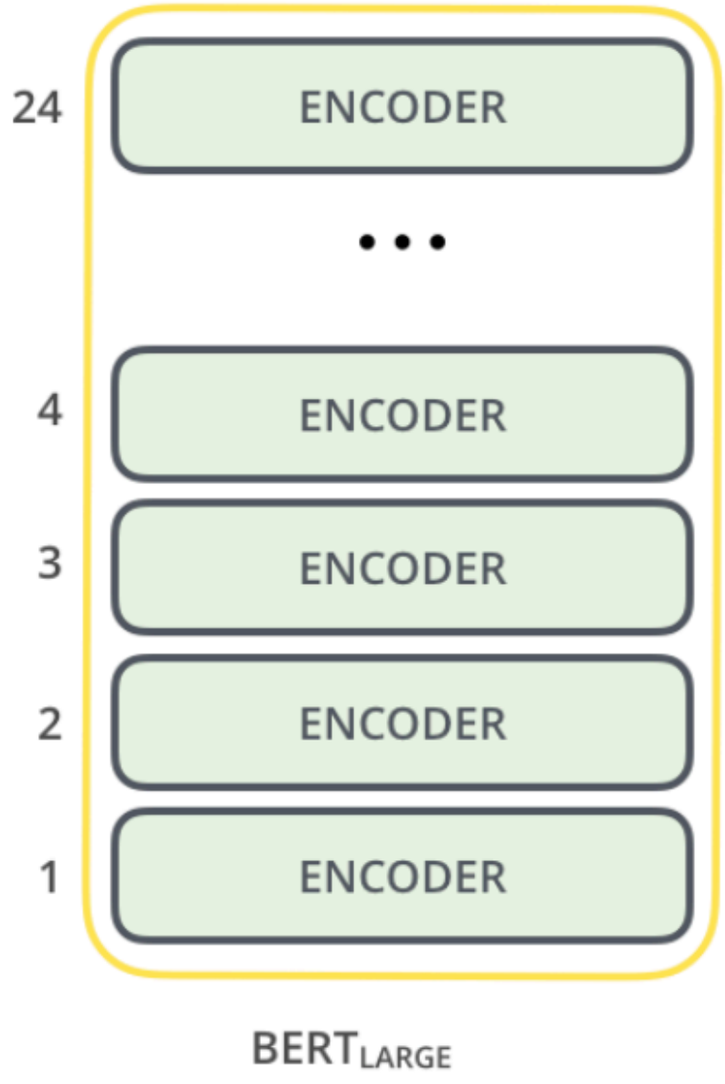
(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER



$BERT_{LARGE}$

http://cs224d.stanford.edu

https://arxiv.org/pdf/1702.01923.pdf

https://arxiv.org/pdf/1511.08630.pdf

https://nlp.stanford.edu/pubs/tai-socher-manning-acl2015.pdf

http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

http://aclweb.org/anthology/D14-1181

http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

https://github.com/UKPLab/deeplearning4nlp-tutorial/blob/master/2015-10_Lecture/Lecture5/2015-11-02_Convolutional_NN.pdf

https://cs224d.stanford.edu/reports/GuptaDesai.pdf

https://arxiv.org/pdf/1503.00075v3.pdf

https://arxiv.org/pdf/1708.02709.pdf

https://www.nyu.edu/projects/bowman/NLU%20Guest%20Lecture%20S%272015.pdf