University of Ljubljana, Faculty of Computer and Information Science

Recurrent neural networks for text

Prof Dr Marko Robnik-Šikonja

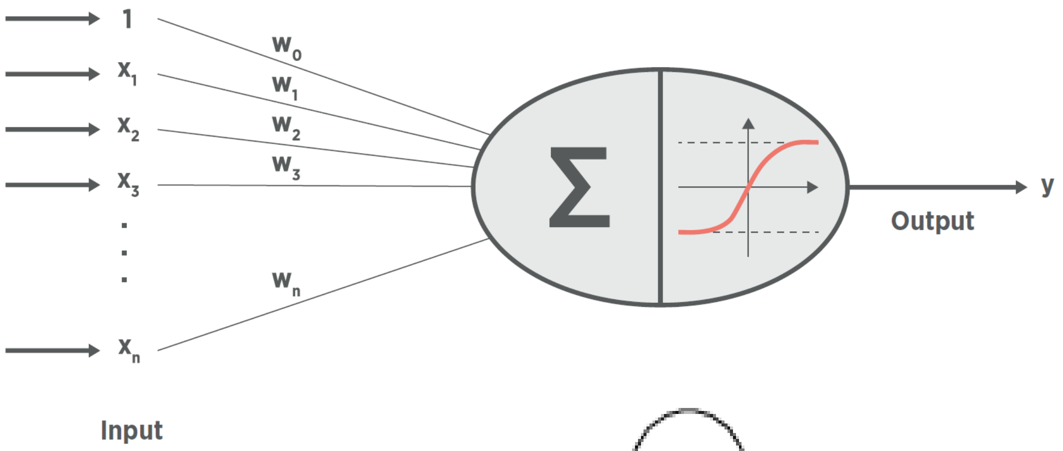
Natural Language Processing, Edition 2023

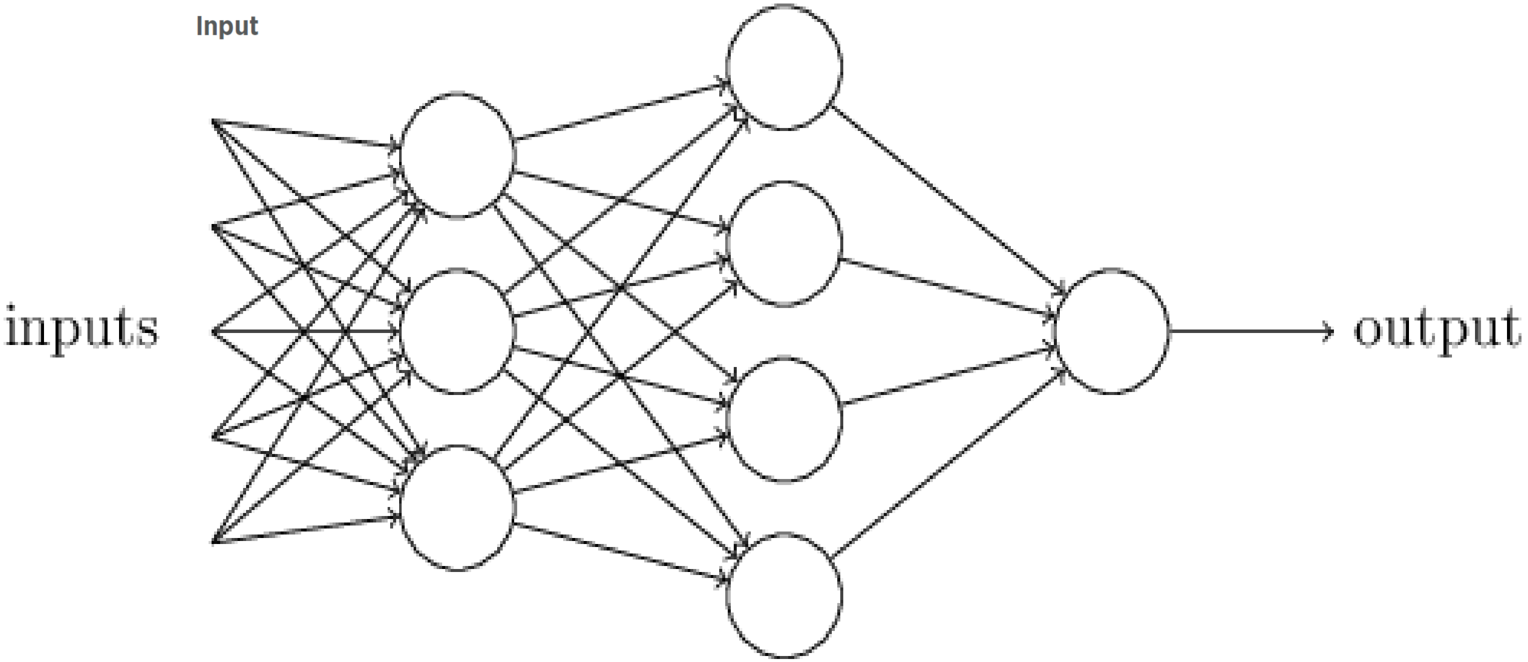
Contents

- recurrent neural networks
- LSTM and GRU networks

read Chapter 9 in Jurafsky & Martin, 3rd edition

Revision: artificial neural networks

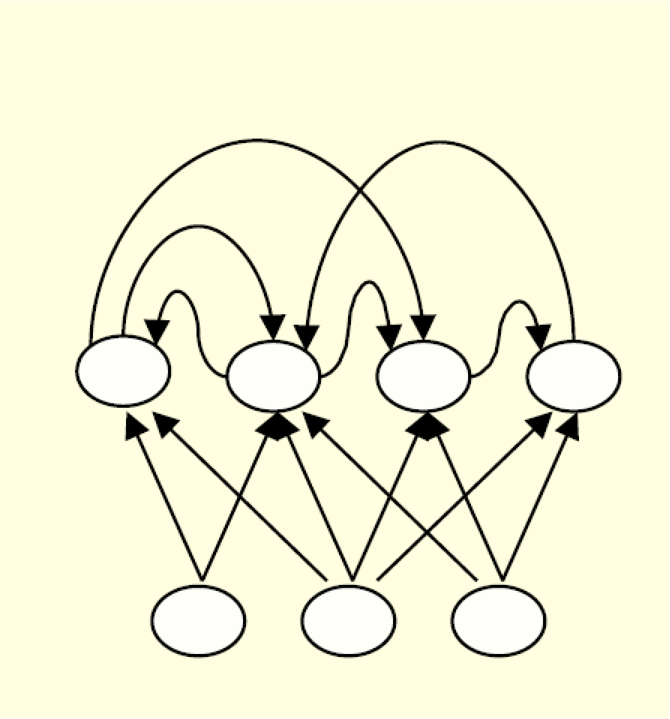




besides feed-forward networks, there are many other more complex network architectures

Recurrent network (RNN)

- back connections
- biologically more realistic
- store information from the past – back connections correspond to memory
- more difficult to learn



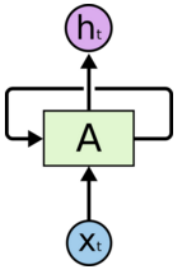
Recurrent Neural Networks

- The applications of standard Neural Networks (and also Convolutional Networks) are limited due to:
  - They only accepted a fixed-size vector as input (e.g., an image or a fixed number of words) and produce a fixed-size vector as output (e.g., probabilities of different classes).
  - These models use a fixed amount of computational steps (e.g. the number of layers in the model).

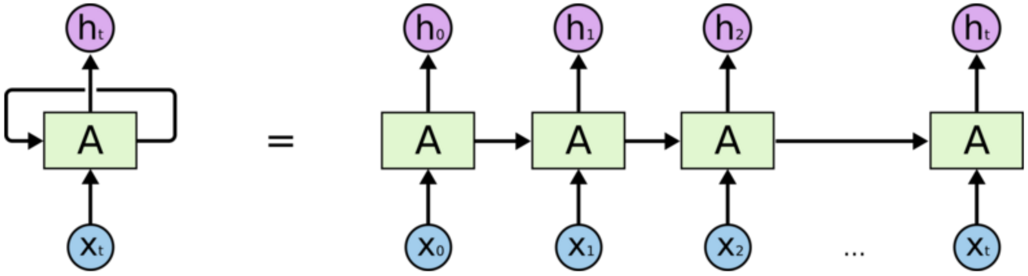
- Recurrent Neural Networks are unique as they allow us to operate over sequences of vectors.
  - Sequences in the input, the output, or in the most general case both

Recurrent Neural Networks

- Recurrent Neural Networks are networks with loops in them, allowing information to persist.



Recurrent Neural Networks have loops.



An unrolled recurrent neural network.

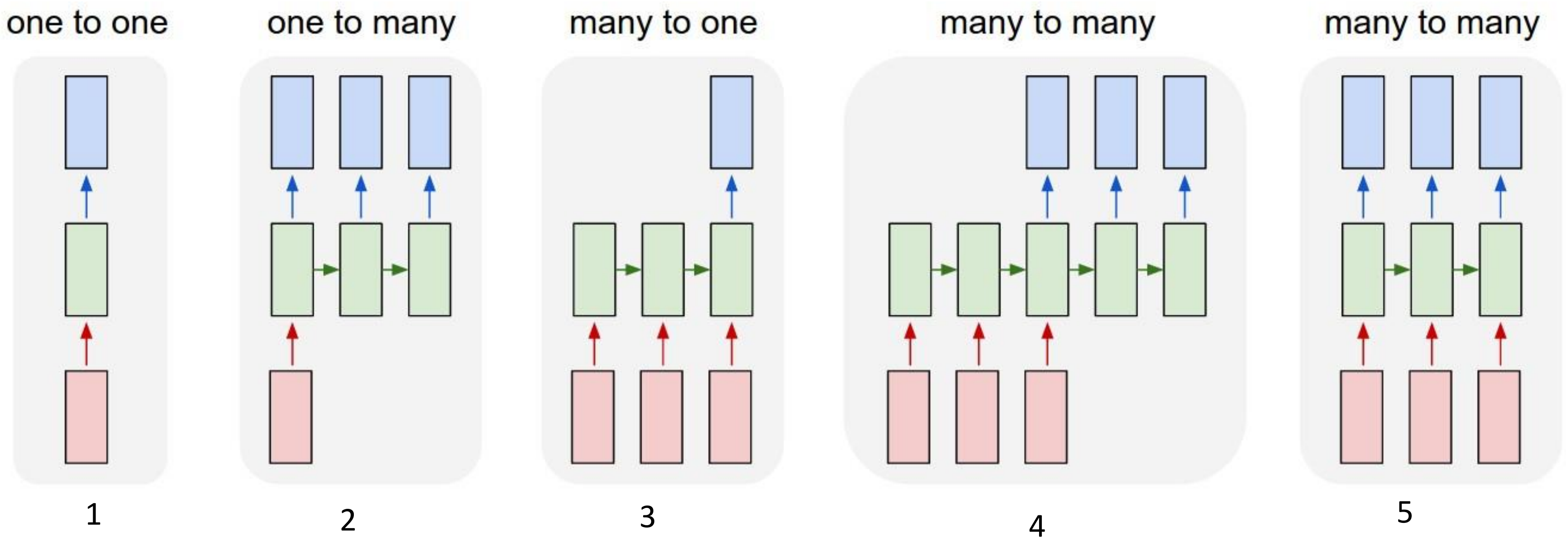
In the above diagram, a chunk of neural network, A, looks at some input x_t and outputs a value h_t.
A loop allows information to be passed from one step of the network to the next.

A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.
The diagram above shows what happens if we unroll the loop.

Recurrent Neural Networks

- Intuition of Recurrent Neural Networks
  - Human thoughts have persistence; humans don’t start their thinking from scratch every second.
    - As you read this sentence, you understand each word based on your understanding of previous words.
  - One of the appeals of RNNs is the idea that they are able to connect previous information to the present task
    - E.g., using previous video frames to inform the understanding of the present frame.
    - E.g., a language model tries to predict the next word based on the previous ones.

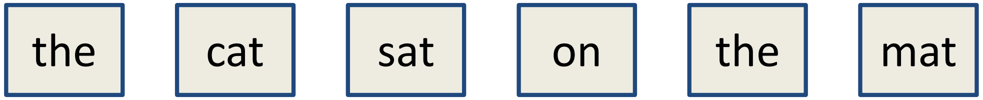
Examples of Recurrent Neural Networks



- Each rectangle is a vector and arrows represent functions (e.g. matrix multiply).
- Input vectors are in red, output vectors are in blue and green vectors hold the RNN's state

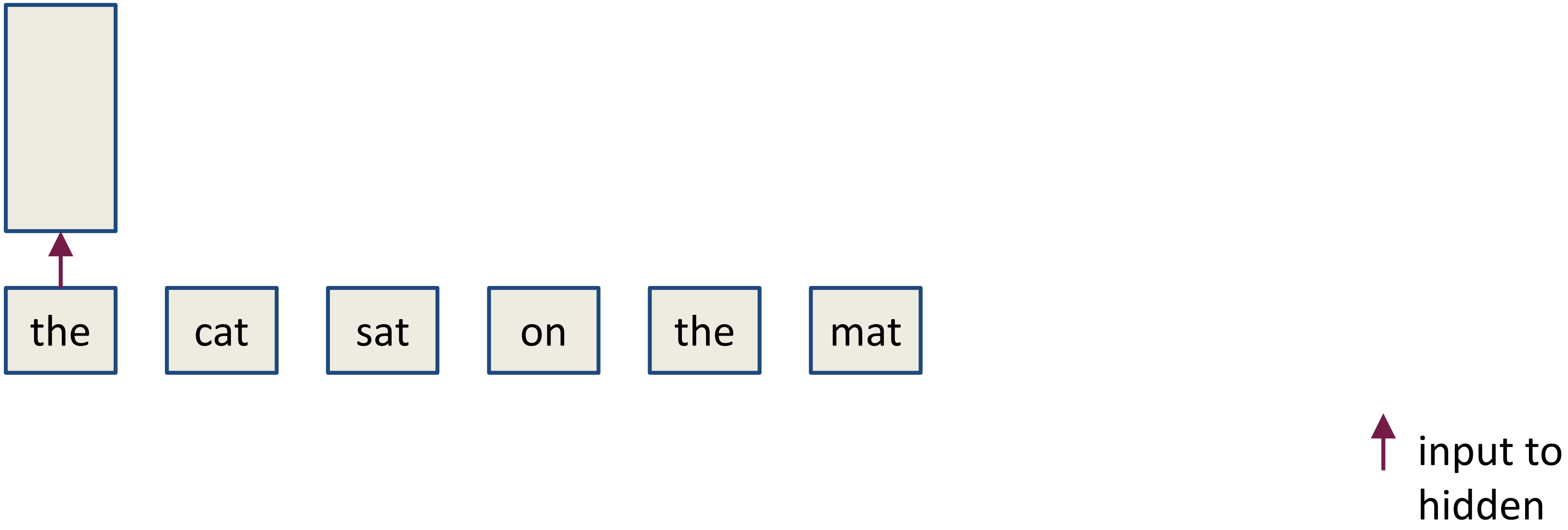
1. Standard mode of processing without RNN, from fixed-sized input to fixed-sized output (e.g. image classification).
2. Sequence output (e.g. image captioning takes an image and outputs a sentence of words).
3. Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment).
4. Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French).
5. Synced sequence input and output (e.g. video classification where we wish to label each frame of the video).

How an RNN works for text



the cat sat on the mat

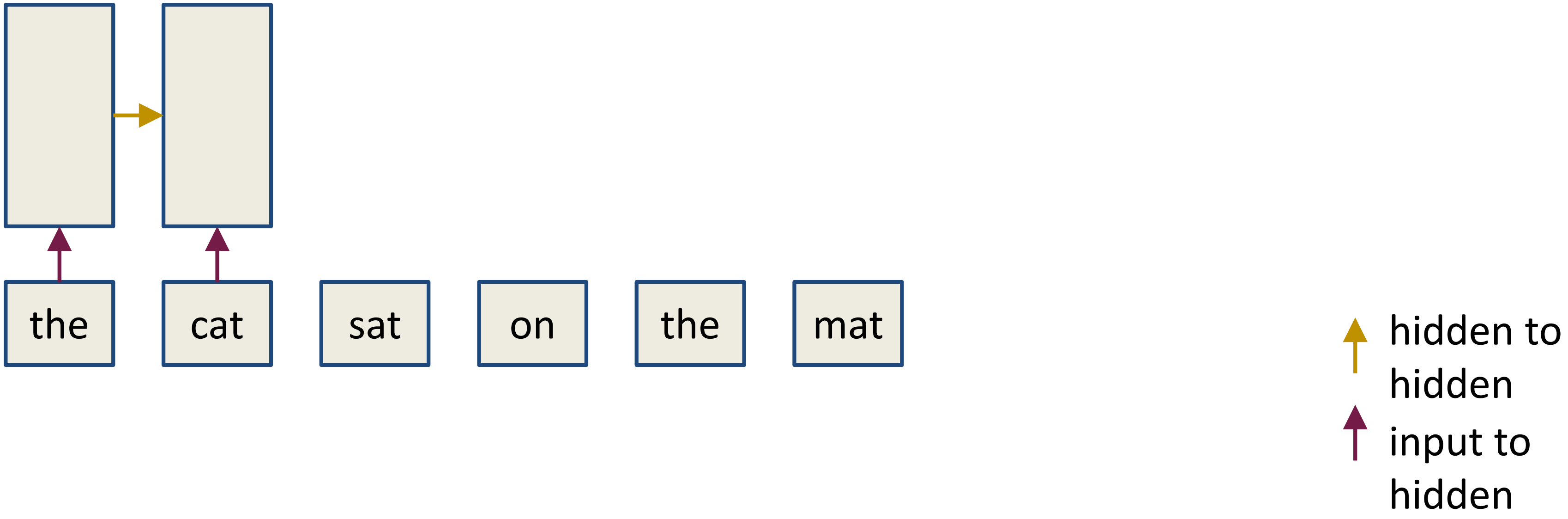
How an RNN works



the cat sat on the mat

input to hidden

How an RNN works

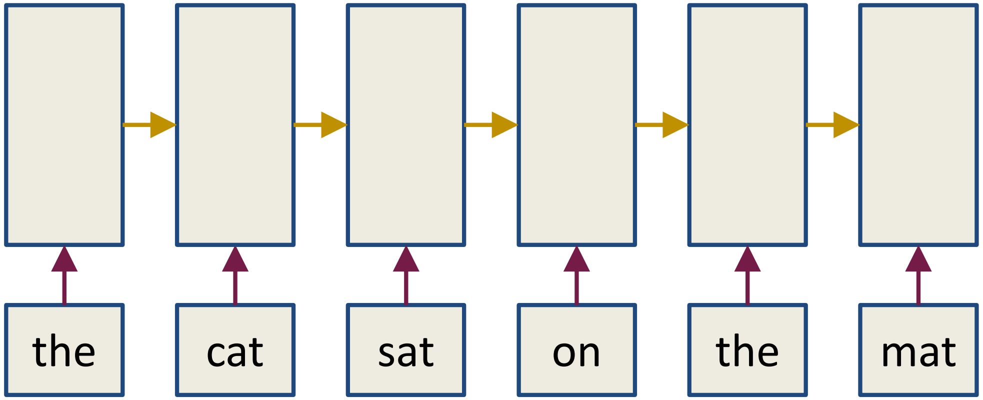


the cat sat on the mat

hidden to hidden

input to hidden

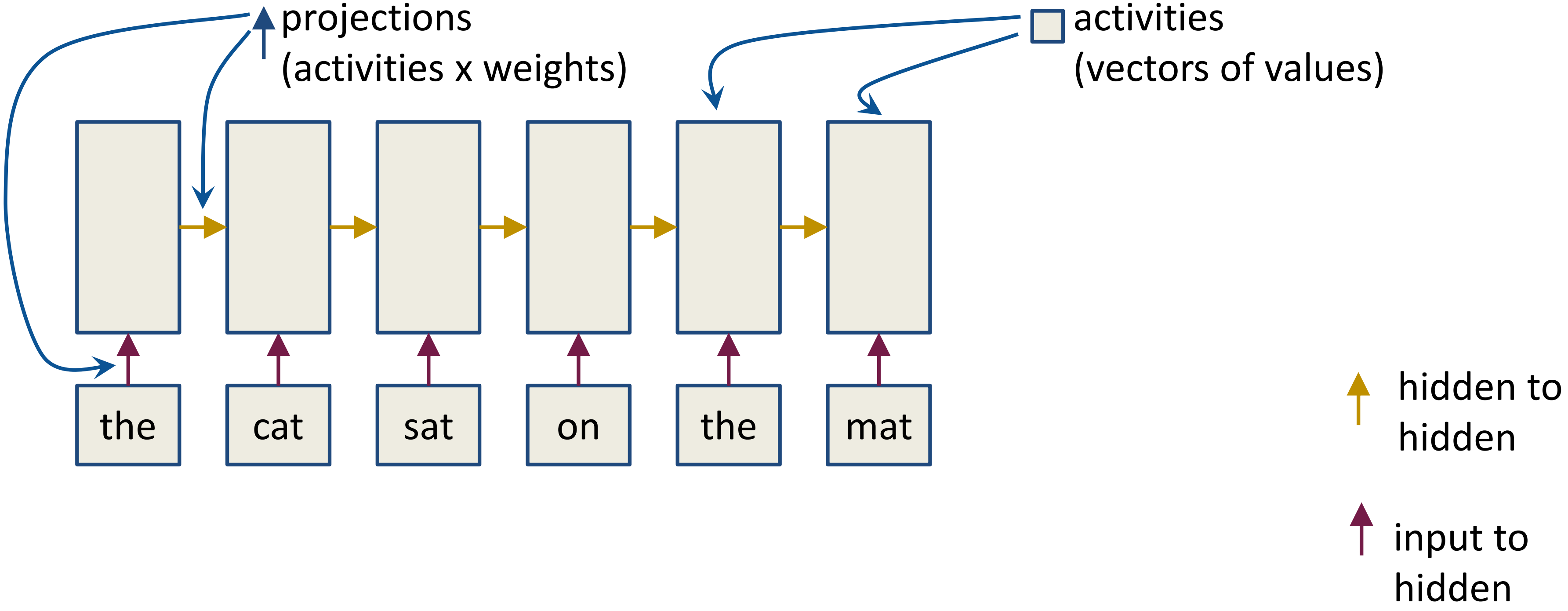
How an RNN works



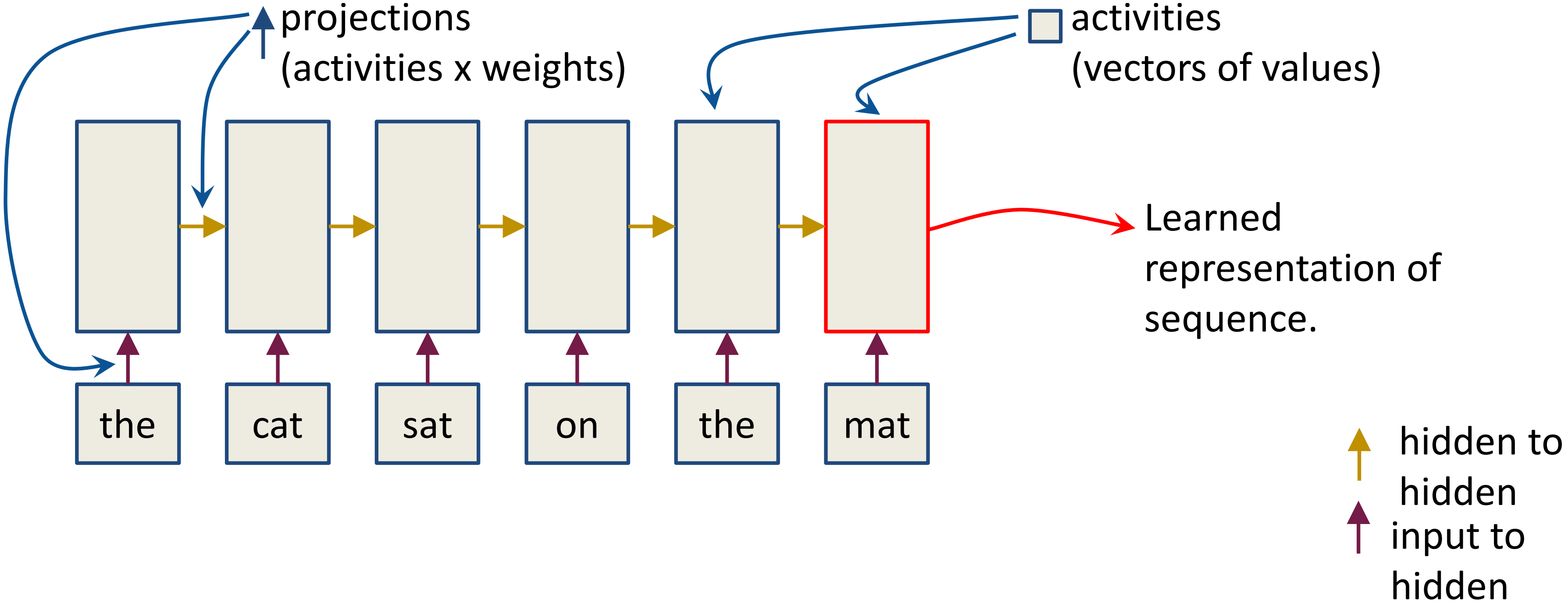
↑ hidden to hidden

↑ input to hidden

How an RNN works



How an RNN works



projections (activities x weights)

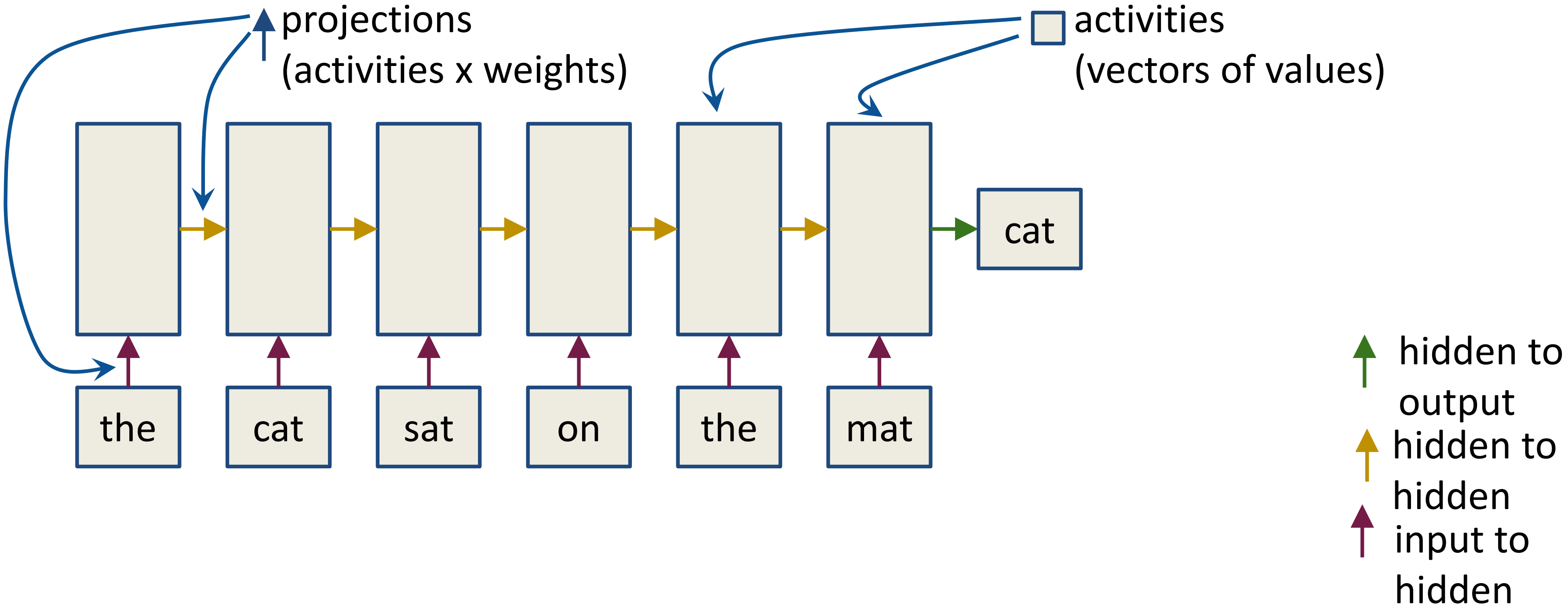
activities (vectors of values)

Learned representation of sequence.

hidden to hidden

input to hidden

How an RNN works



projections
(activities x weights)

activities
(vectors of values)

the cat sat on the mat

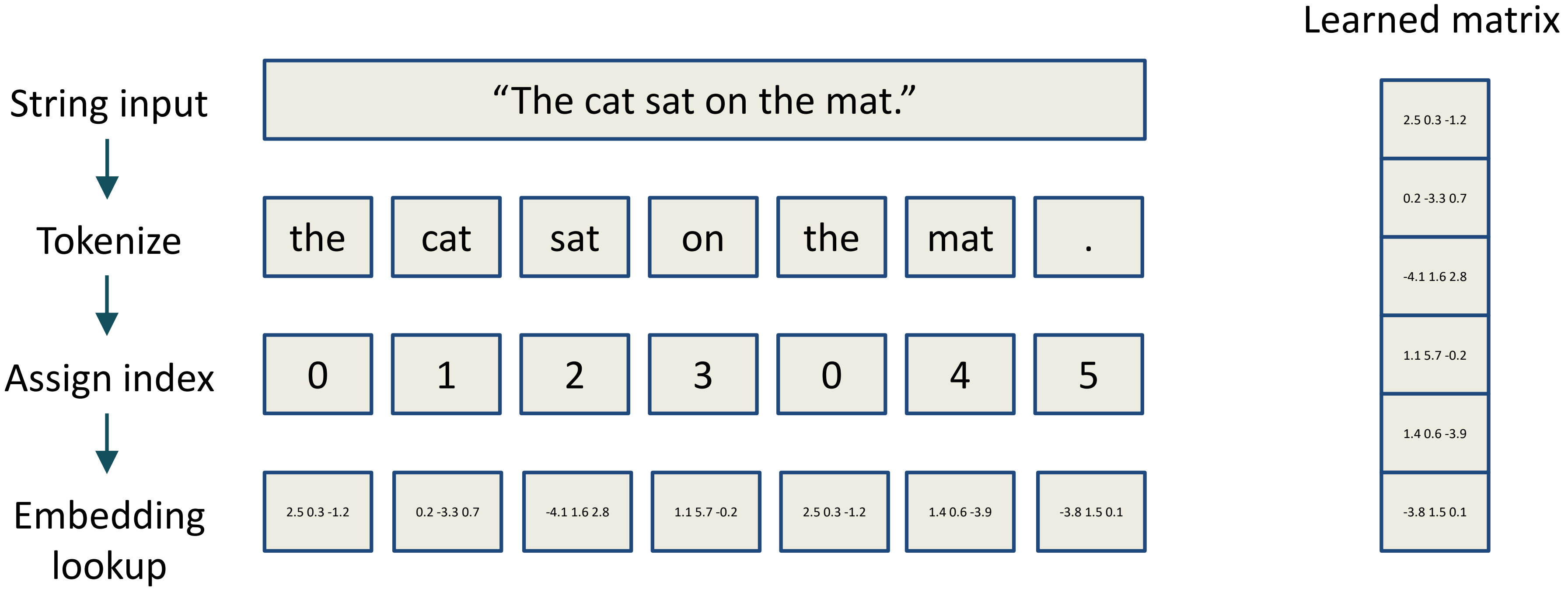
cat

hidden to output

hidden to hidden

input to hidden

From text to RNN input



Learned matrix

String input

“The cat sat on the mat.”

Tokenize

the cat sat on the mat .

Assign index

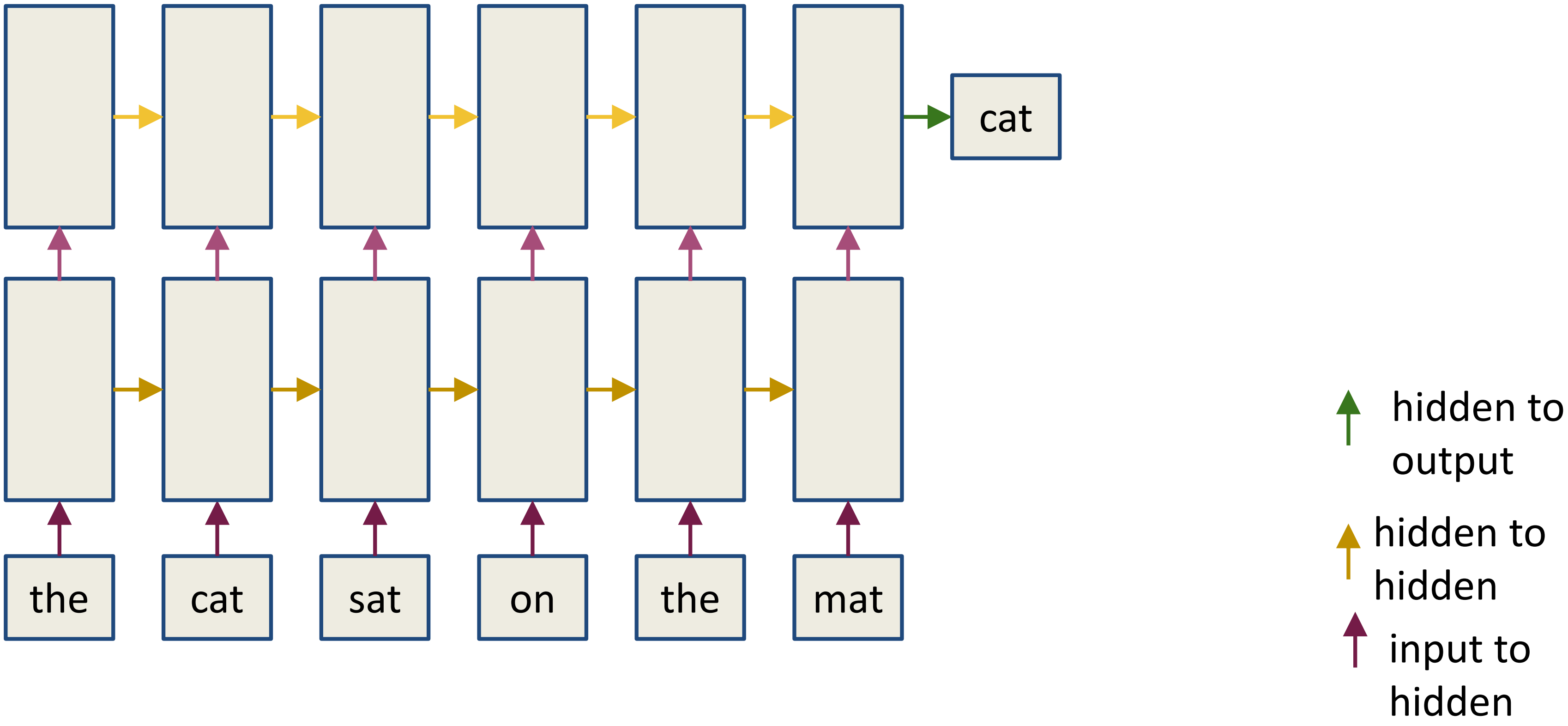
0 1 2 3 0 4 5

Embedding lookup

2.5 0.3 -1.2 | 0.2 -3.3 0.7 | -4.1 1.6 2.8 | 1.1 5.7 -0.2 | 2.5 0.3 -1.2 | 1.4 0.6 -3.9 | -3.8 1.5 0.1

2.5 0.3 -1.2
0.2 -3.3 0.7
-4.1 1.6 2.8
1.1 5.7 -0.2
1.4 0.6 -3.9
-3.8 1.5 0.1

You can stack them too



cat

the

cat

sat

on

the

mat

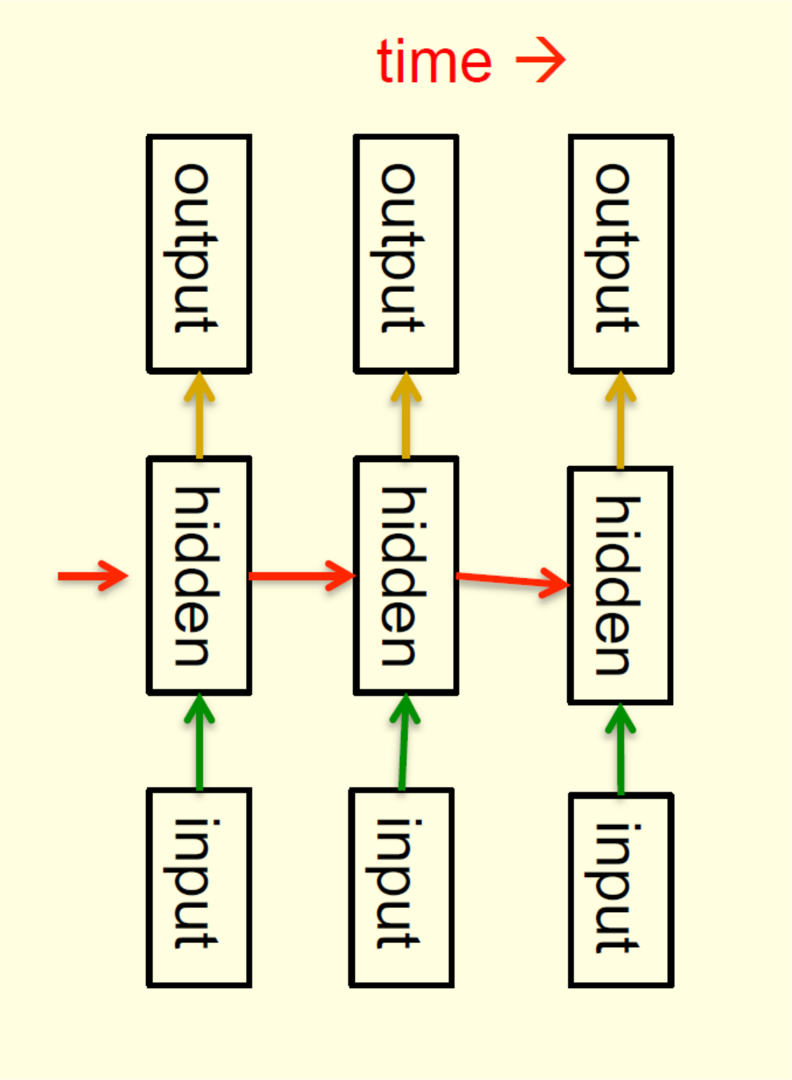
hidden to output

hidden to hidden

input to hidden

Recurrent networks for sequence learning

- equivalent to deep networks with one hidden level per time slot
- but: hidden layers share weight (less parameters)



Example: RNN as a LM

- Ilya Sutskever (2011): RNN as a character-based language model – prediction of the next character
- training set: half a billion characters from English Wikipedia
- used for text generation:
  - predict probability distribution for the next character
  - sample from that distribution

RNN of Ilya Sutskever - sample output

In 1974 Northern Denver had been overshadowed by CNL, and several Irish intelligence agencies in the Mediterranean region. However, on the Victoria, Kings Hebrew stated that Charles decided to escape during an alliance. The mansion house was completed in 1882, the second in its bridge are omitted, while closing is the proton reticulum composed below it aims, such that it is the blurring of appearing on any well-paid type of box printer.

Next word in speech recognition

- difficult recognition in noisy environment
- many words with similar acoustic signal
- humans use semantics, context, and prediction to recognize the correct right word
- wreck a nice beach.
- recognize a speech

- automatic speech recognizers (ASR) need information about sensible next words
- classical approach was the trigram LM

Trigram LM

- using a large corpora we compute the probability of trigrams

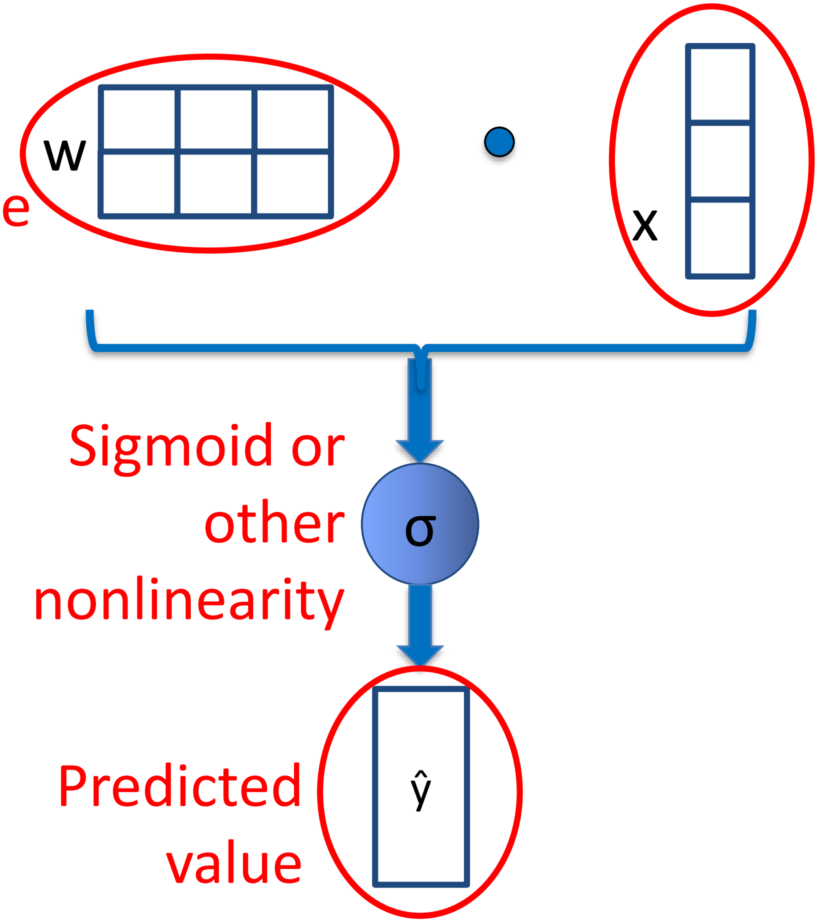
$$\frac{p(w_3 = c \mid w_2 = b, w_1 = a)}{p(w_3 = d \mid w_2 = b, w_1 = a)} = \frac{count(abc)}{count(abd)}$$

- some combinations do not appear in the corpus but that does not mean that they are not possible

Weaknesses of trigram model

- Seeing a sentence
  - “the cat walked over the garden on friday”
- shall increase the probability of a sentence
  - “the dog ran through the yard on monday”
- trigram model does not recognize the similarity between
  - cat/dog walked/ran garden/yard friday/monday
- we need information about the past to predict the future
- traditional approach: construct semantic and syntactic features to capture longer context
- recently: use DNNs

Schematic view of NNs



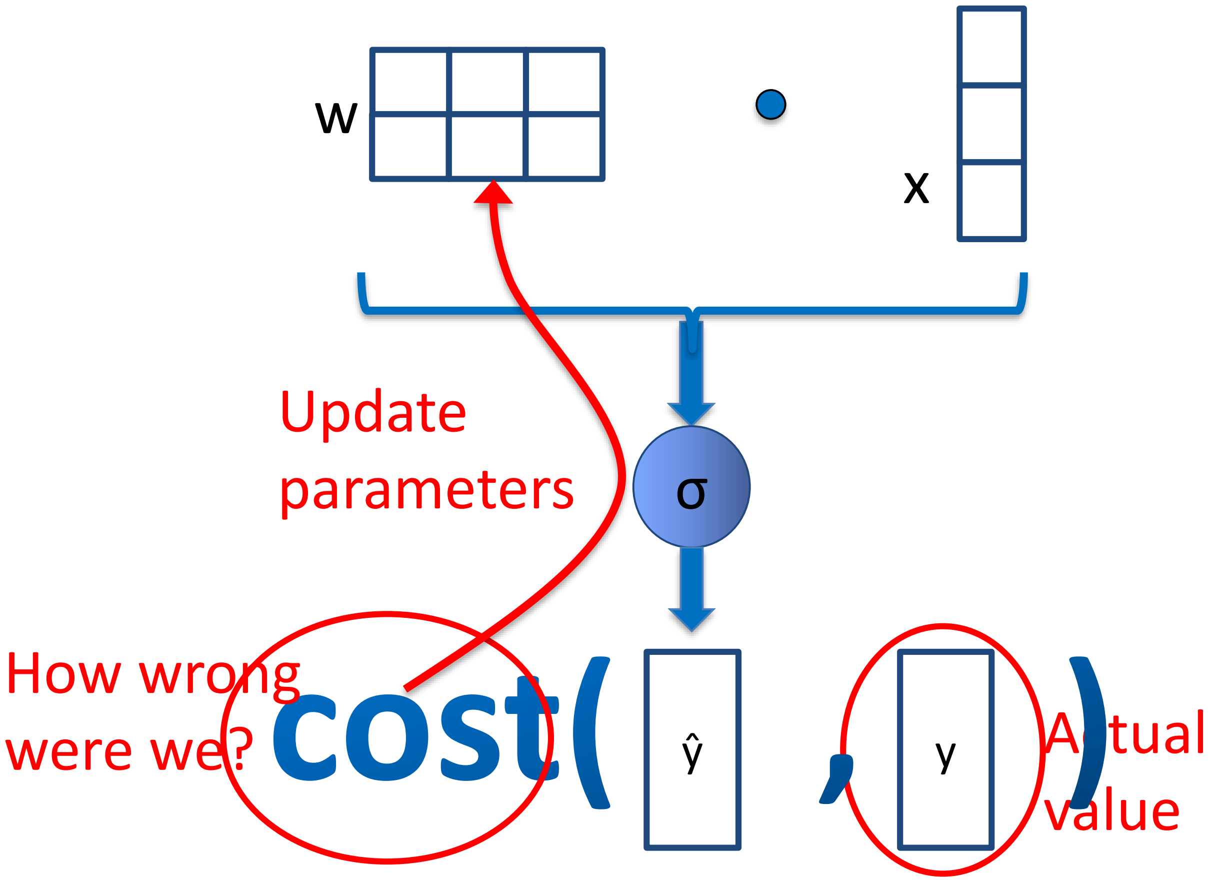
Parameters (things we're learning)

Input feature vector

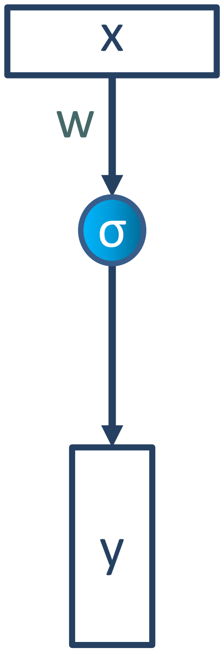
Sigmoid or other nonlinearity

Predicted value

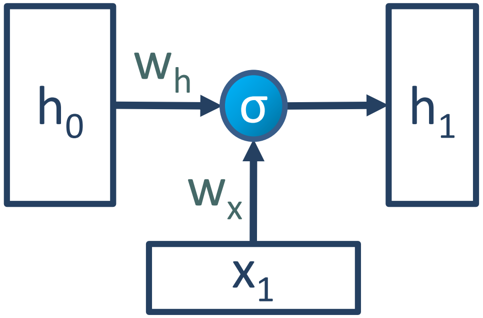
Schematic view of NNs



A Simplified Diagram

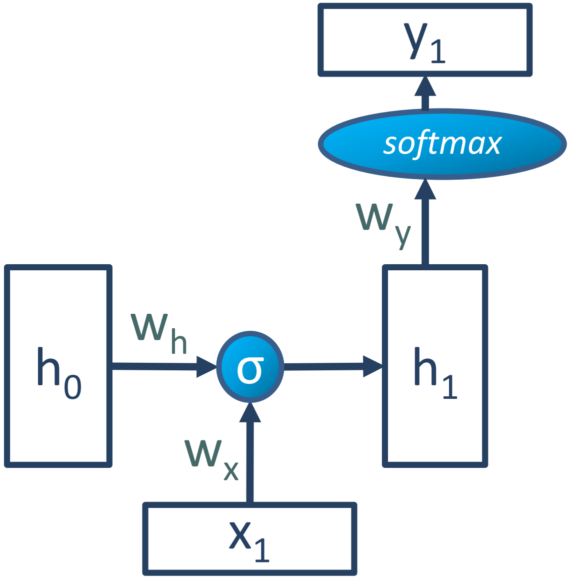


Recurrent Neural Networks (RNNs)



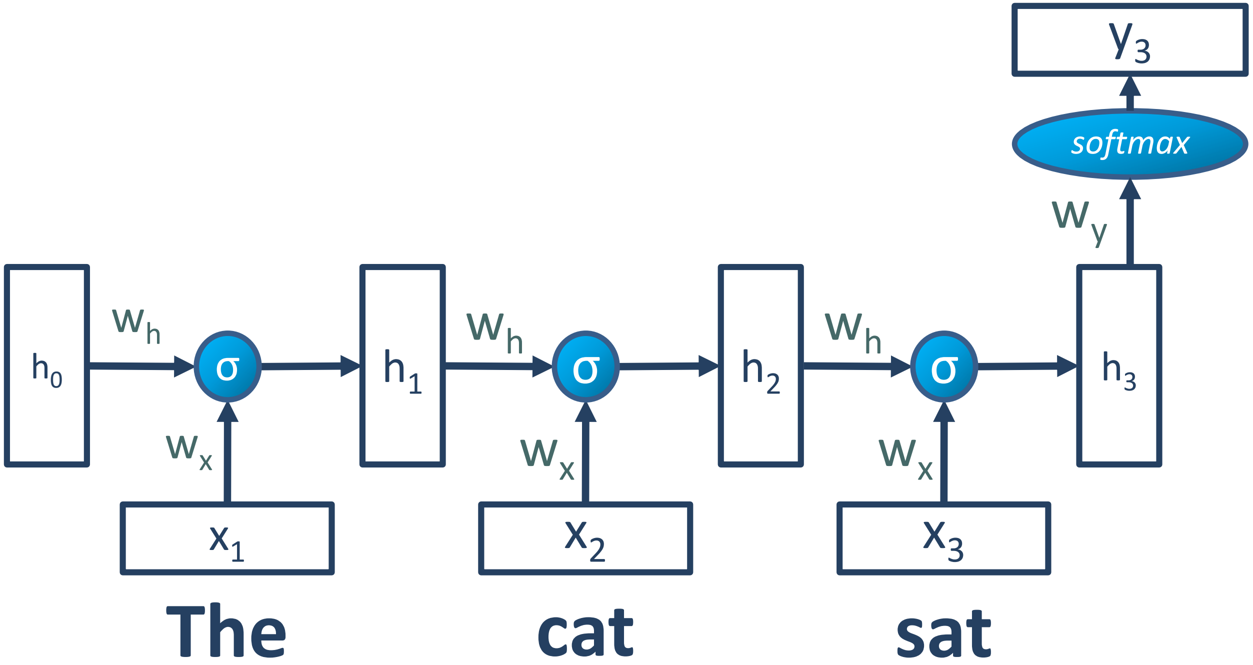
$$h_t = \sigma(W_h h_{t-1} + W_x x_t)$$

RNN

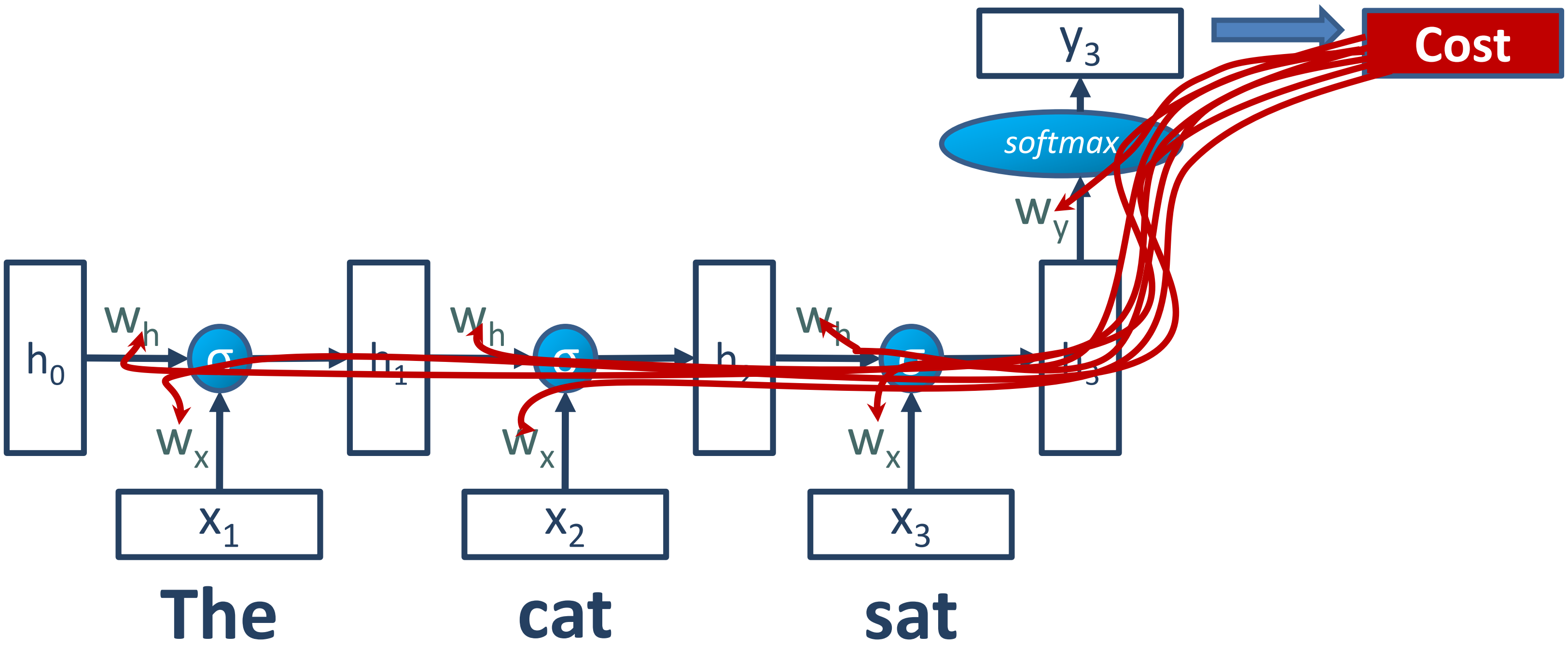


$$h_t = \sigma(W_h h_{t-1} + W_x x_t)$$
$$y_t = softmax(W_y h_t)$$

RNN

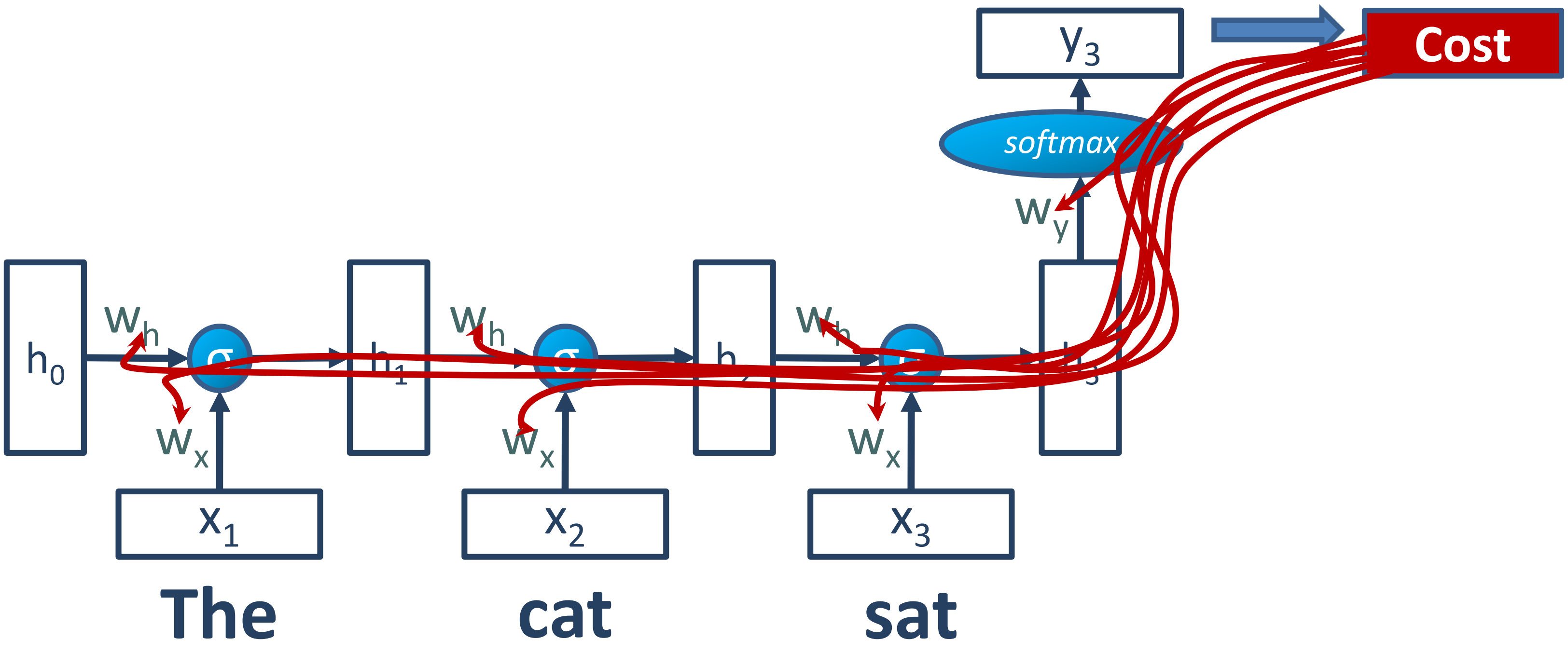


Updating Parameters of an RNN



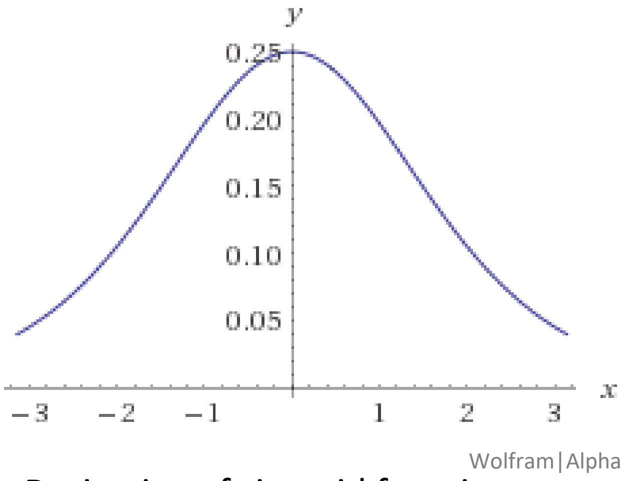
LSTM Motivation

Remember how we update RNN?



The Vanishing Gradient Problem

- Deep neural networks use backpropagation.
- Back propagation uses the chain rule.
- The chain rule multiplies derivatives.
- Often these derivatives are between 0 and 1.
- As the chain gets longer, products get smaller
- until they disappear.



Derivative of sigmoid function

Or do they explode?

- With gradients larger than 1,
- you encounter the opposite problem
- with products becoming larger and larger
- as the chain becomes longer and longer,
- causing overlarge updates to parameters.
- This is the exploding gradient problem.

Vanishing/Exploding Gradients Are Bad.

- If we cannot backpropagate very far through the network, the network cannot learn long-term dependencies.

  - My dog [chase/chases] squirrels. ✓

  vs.

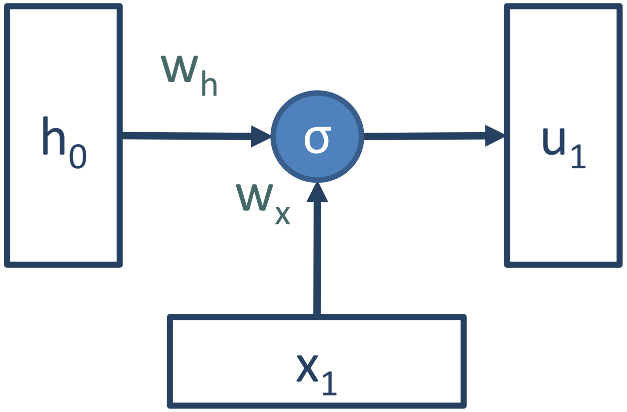
  - My dog, whom I adopted in 2009, [chase/chases] squirrels. ✖

LSTM Solution

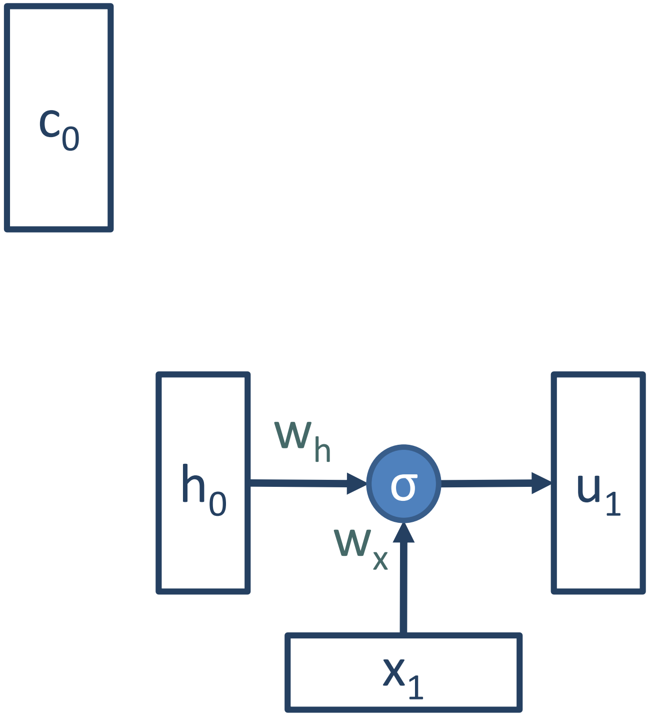
- Use memory cell to store information at each time step.
- Use “gates” to control the flow of information through the network.
  - Input gate: protect the current step from irrelevant inputs
  - Output gate: prevent the current step from passing irrelevant outputs to later steps
  - Forget gate: limit information passed from one cell to the next

Transforming RNN to LSTM

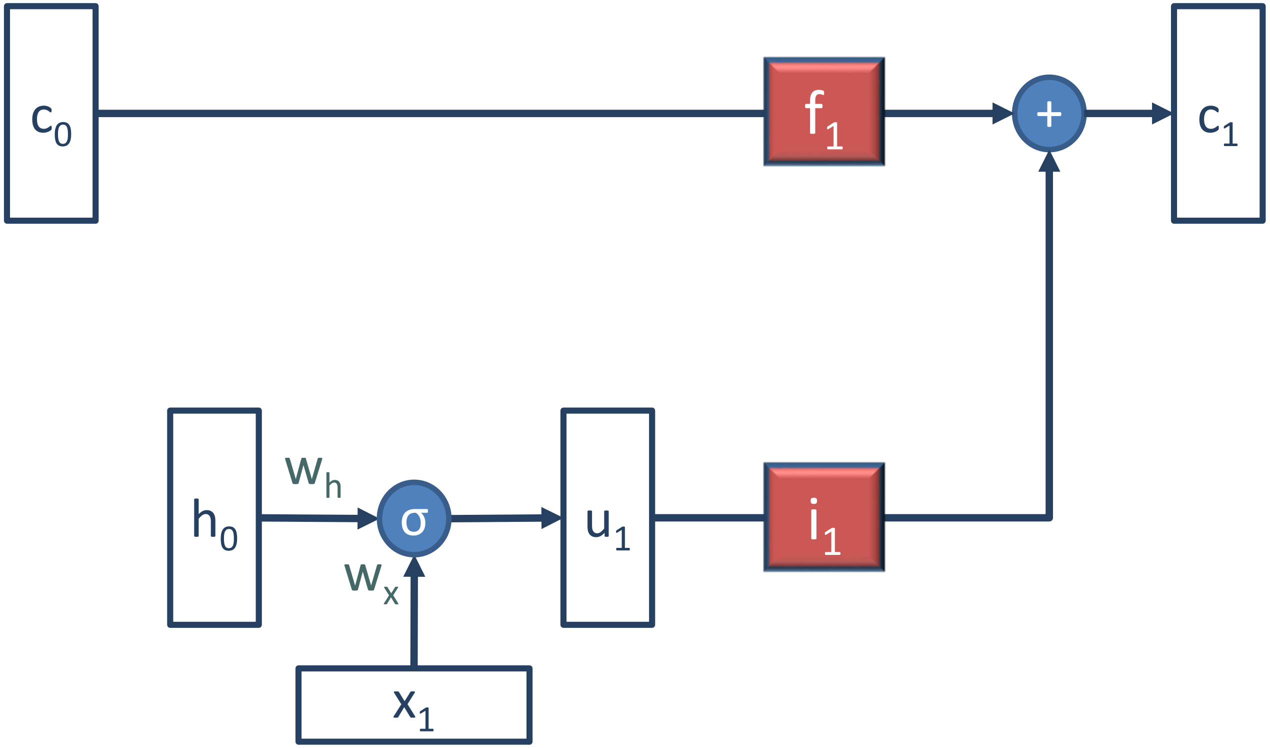
$$u_t = \sigma(W_h h_{t-1} + W_x x_t)$$



Transforming RNN to LSTM

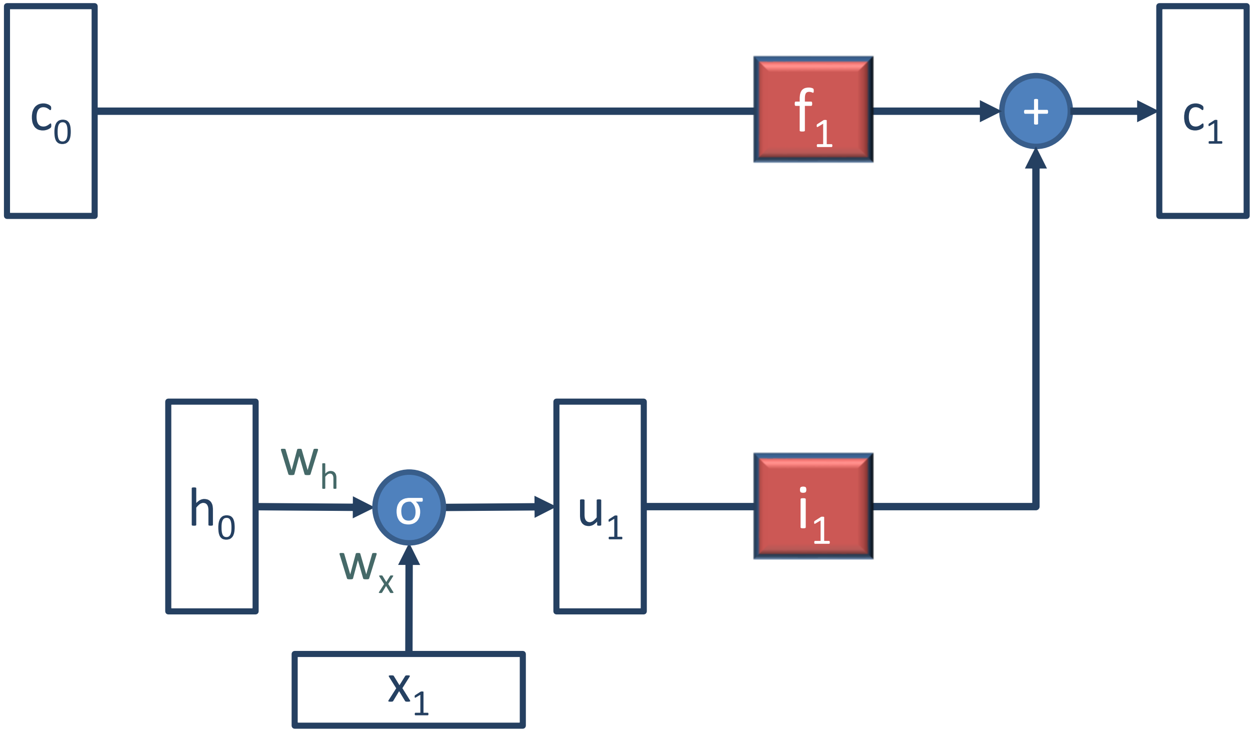


Transforming RNN to LSTM



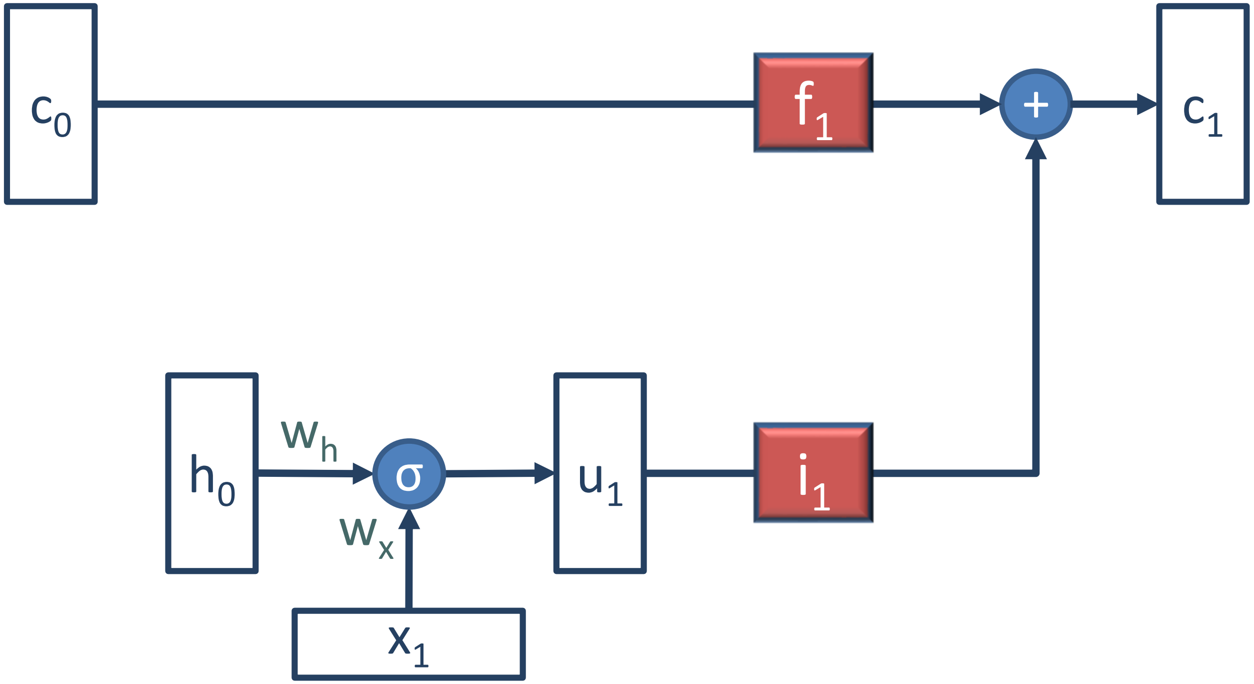
$$c_t = f_t \odot c_{t-1} + i_t \odot u_t$$

Transforming RNN to LSTM



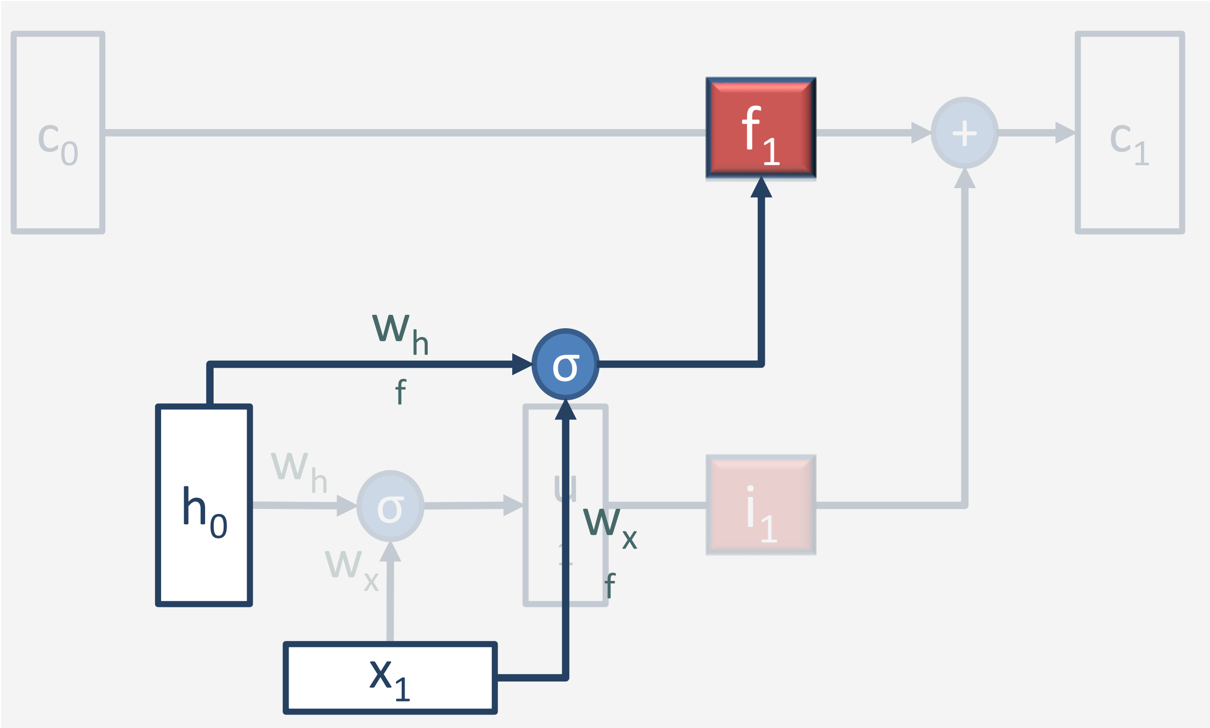
$c_t = f_t \odot c_{t-1} + i_t \odot u_t$

Transforming RNN to LSTM



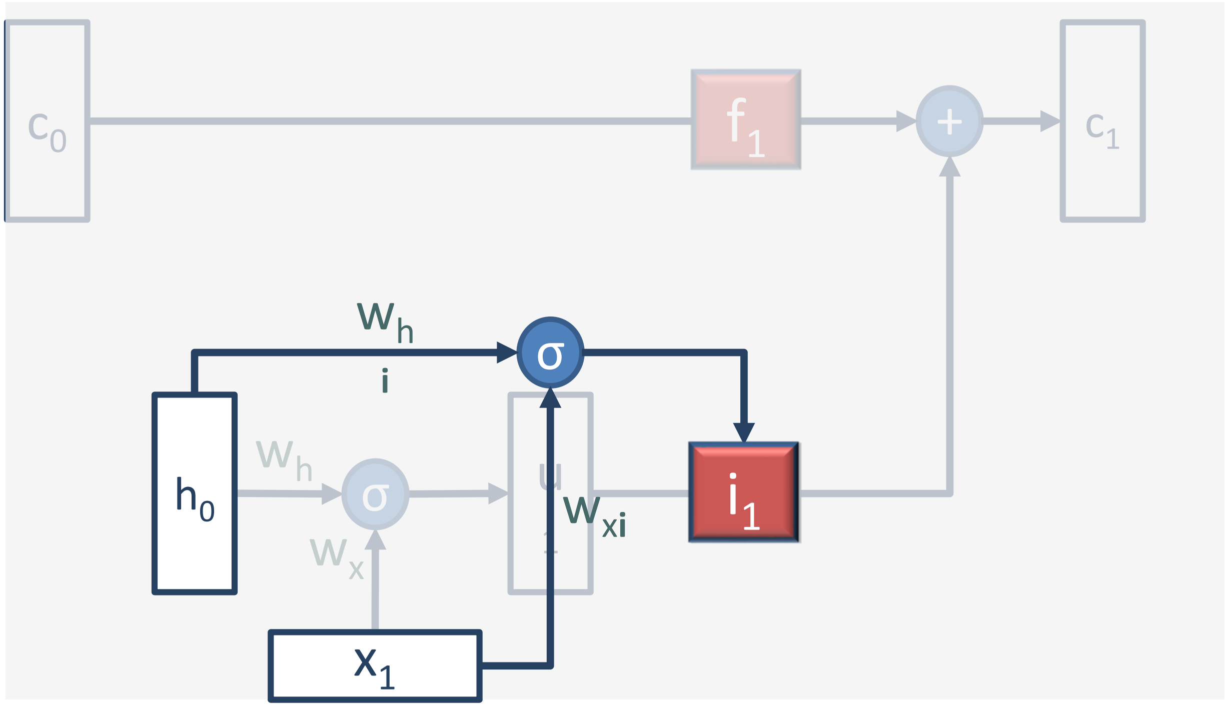
$$c_t = f_t \odot c_{t-1} + i_t \odot u_t$$

Transforming RNN to LSTM



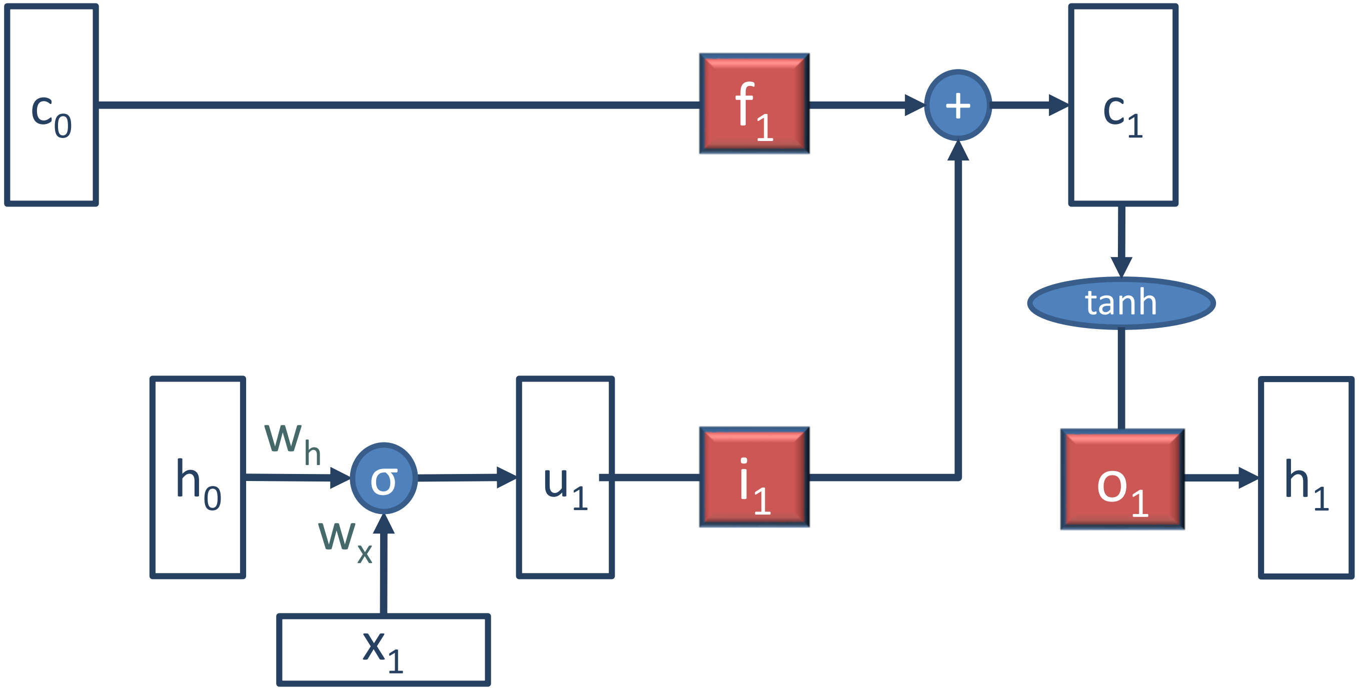
$$f_t = \sigma(W_{hf}h_{t-1} + W_{xf}x_t)$$

Transforming RNN to LSTM



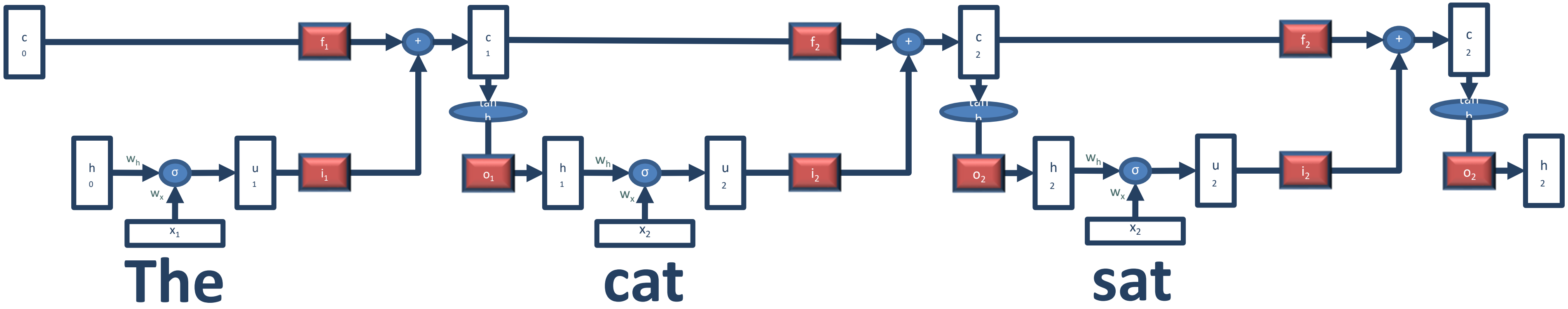
$$i_t = \sigma(W_{hi}h_{t-1} + W_{xi}x_t)$$

Transforming RNN to LSTM



$$h_t = o_t \odot \tanh c_t$$

LSTM for Sequences

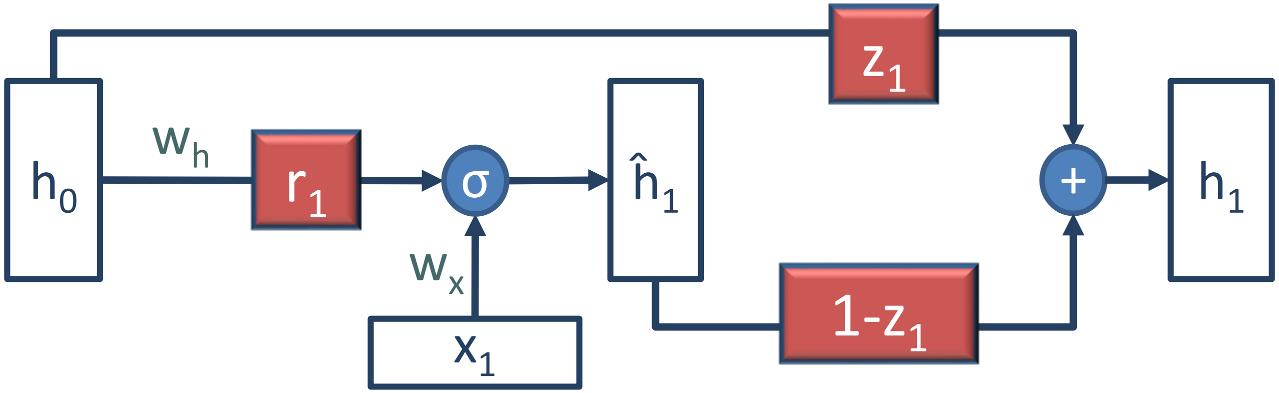


LSTM Applications

Handwriting generation

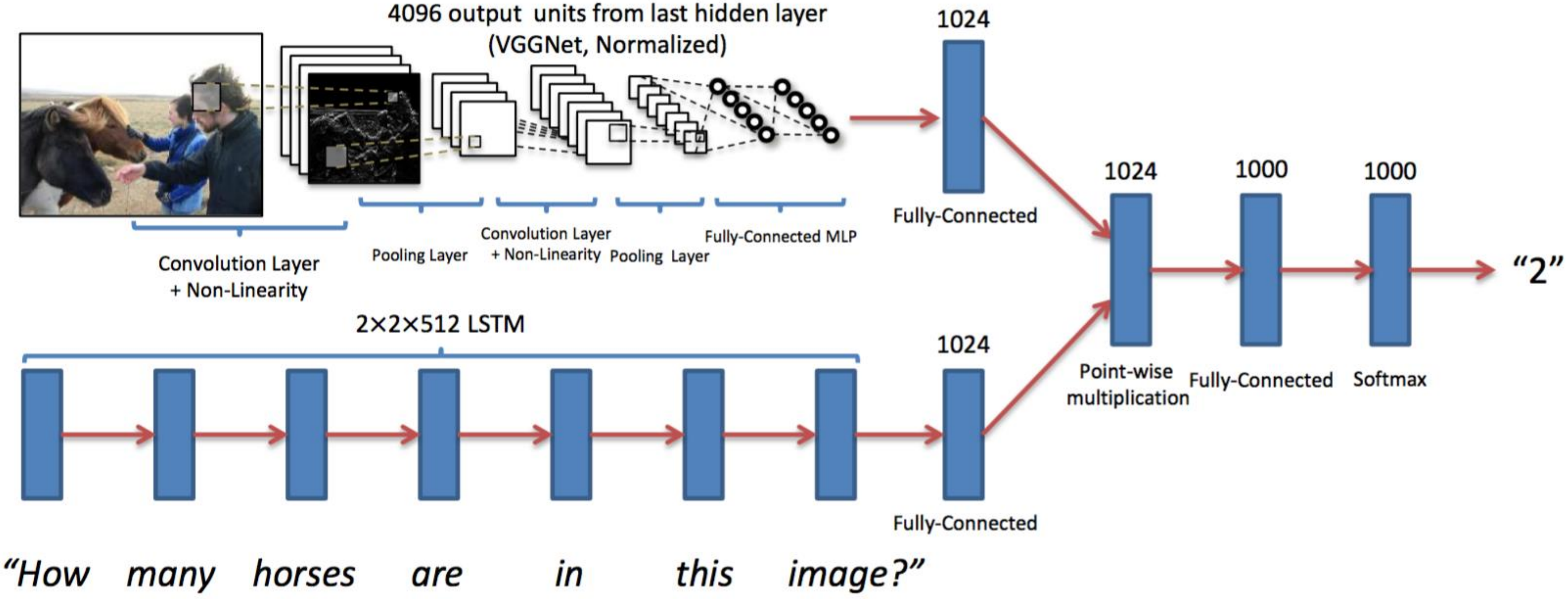
- Language identification (Gonzalez-Dominguez et al., 2014)
- Paraphrase detection (Cheng & Kartsaklis, 2015)
- Speech recognition (Graves, Abdel-Rahman, & Hinton, 2013)
- Handwriting recognition (Graves & Schmidhuber, 2009)
- Music composition (Eck & Schmidhuber, 2002) and lyric generation (Potash, Romanov, & Rumshisky, 2015)
- Robot control (Mayer et al., 2008)
- Natural language generation (Wen et al. 2015)
- Named entity recognition (Hammerton, 2003)
- And many more, but transformers are replacing them

Related Architectures: GRU



- only two gates, reset and update
- Chung et al. (2014) reports comparable performance to LSTM
- later use confirms that, but LSTMs were often slightly better and presented the first choice in NLP until transformers

Visual Question Answering


4096 output units from last hidden layer
(VGGNet, Normalized)
1024
Convolution Layer
+ Non-Linearity
Pooling Layer
Convolution Layer
+ Non-Linearity
Pooling Layer
Fully-Connected MLP
Fully-Connected
1024
1000
1000
"2"
2×2×512 LSTM
1024
Point-wise
multiplication
Fully-Connected
Softmax
Fully-Connected
"How
many
horses
are
in
this
image?"