Deep Learning for Natural Language Processing

(2/2)

Alexis Conneau
PhD student @ Facebook AI Research

Master MVA, 2018

Outline

Class 1

- Overview of some classical NLP tasks
- Word2vec: word embeddings
- Bag-of-words representations

Class 2

- Recurrent Neural Networks (RNNs, LSTMs)
- Language Modelling/Generation
- Encoders and decoders: AutoEncoder/NMT/Seq2Tree/Seq2Img
- Going further: Attention mechanism and Byte Pair Encoding

Applications

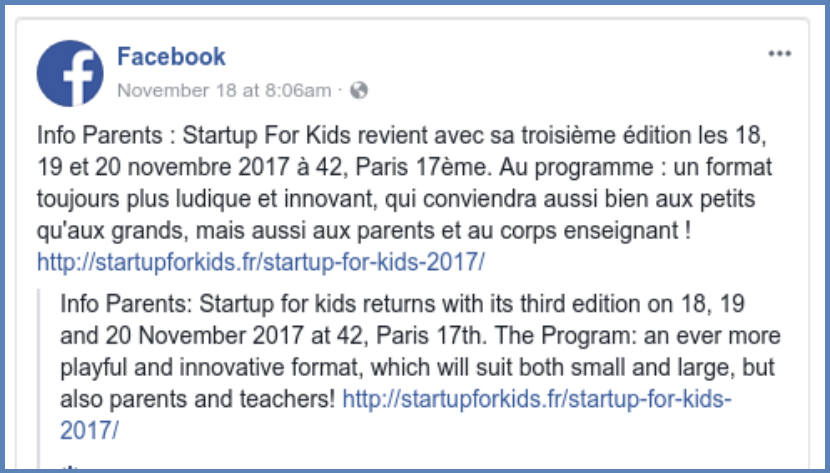
- Sentence classification
- Sentiment analysis
- Answer selection

Applications

Machine translation

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever, Oriol Vinyals, Quoc V. Le



French

English

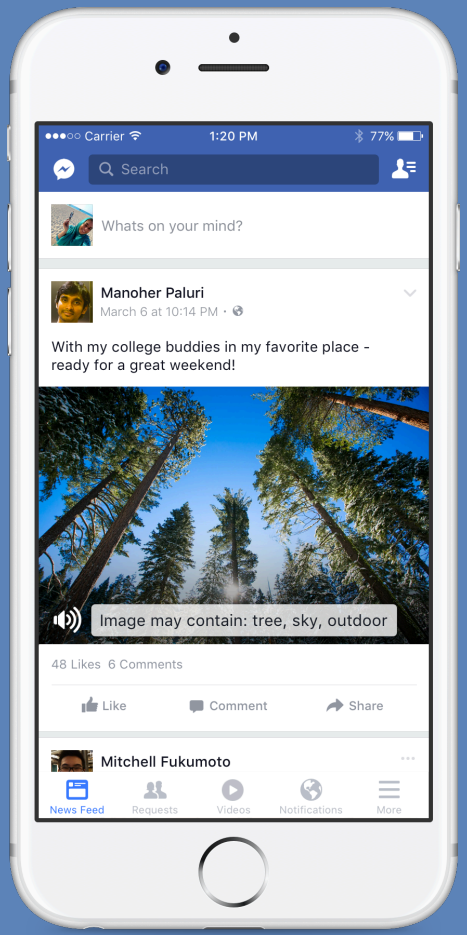
Applications

Image captioning

Making Facebook visual content accessible to visually impaired

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.



Outline

Class 1

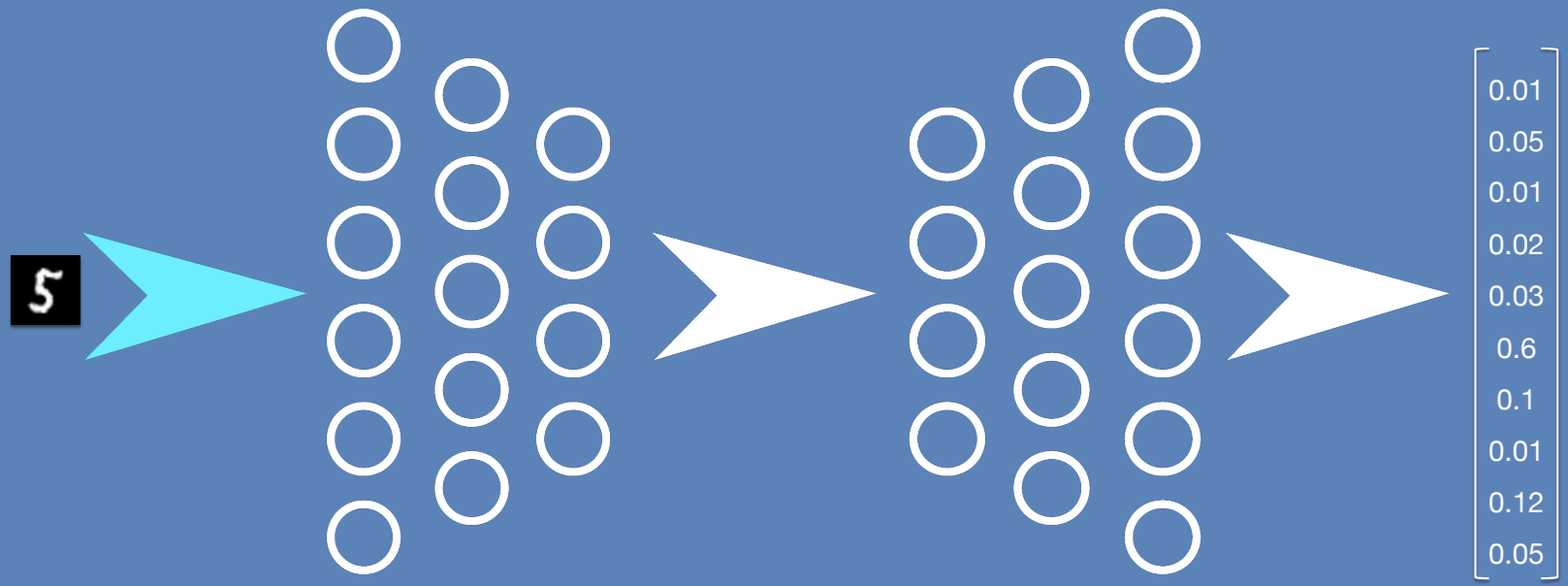
- Overview of some classical NLP tasks
- Word2vec: word embeddings
- Bag-of-words representations

Class 2

- Recurrent Neural Networks (RNNs, LSTMs)
- Language Modelling/Generation
- Encoders and decoders

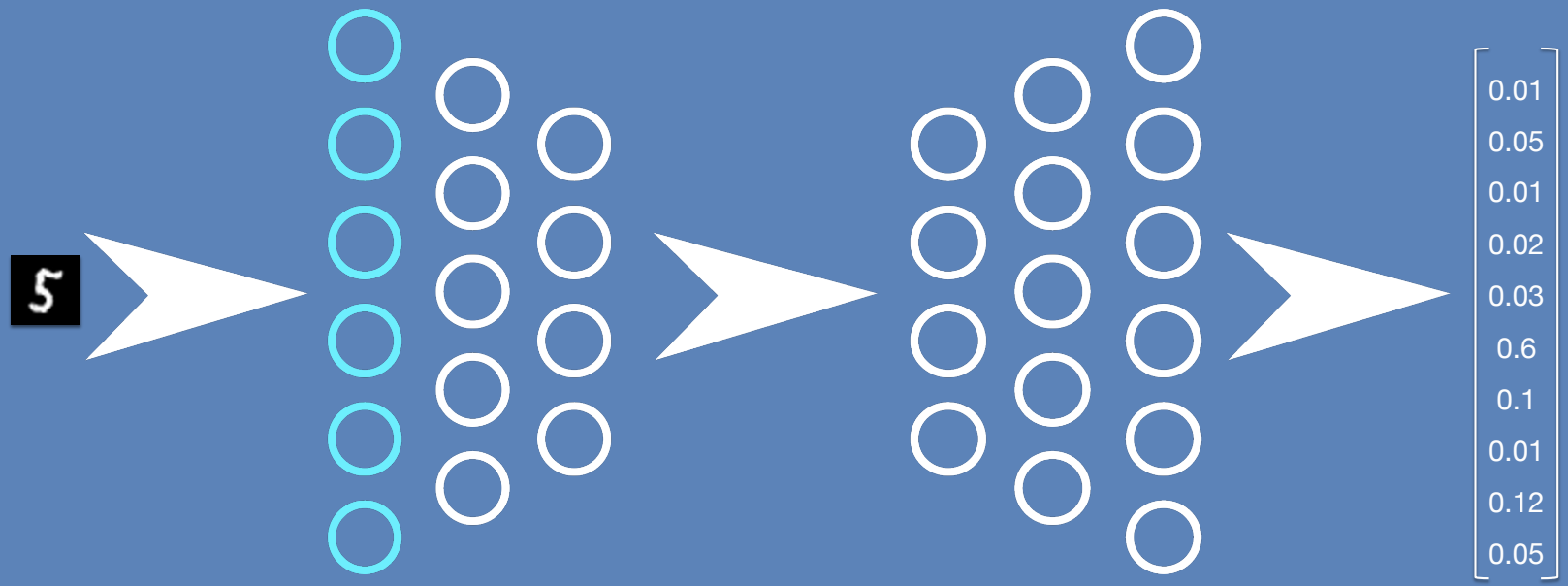
Forward Pass

Reminder



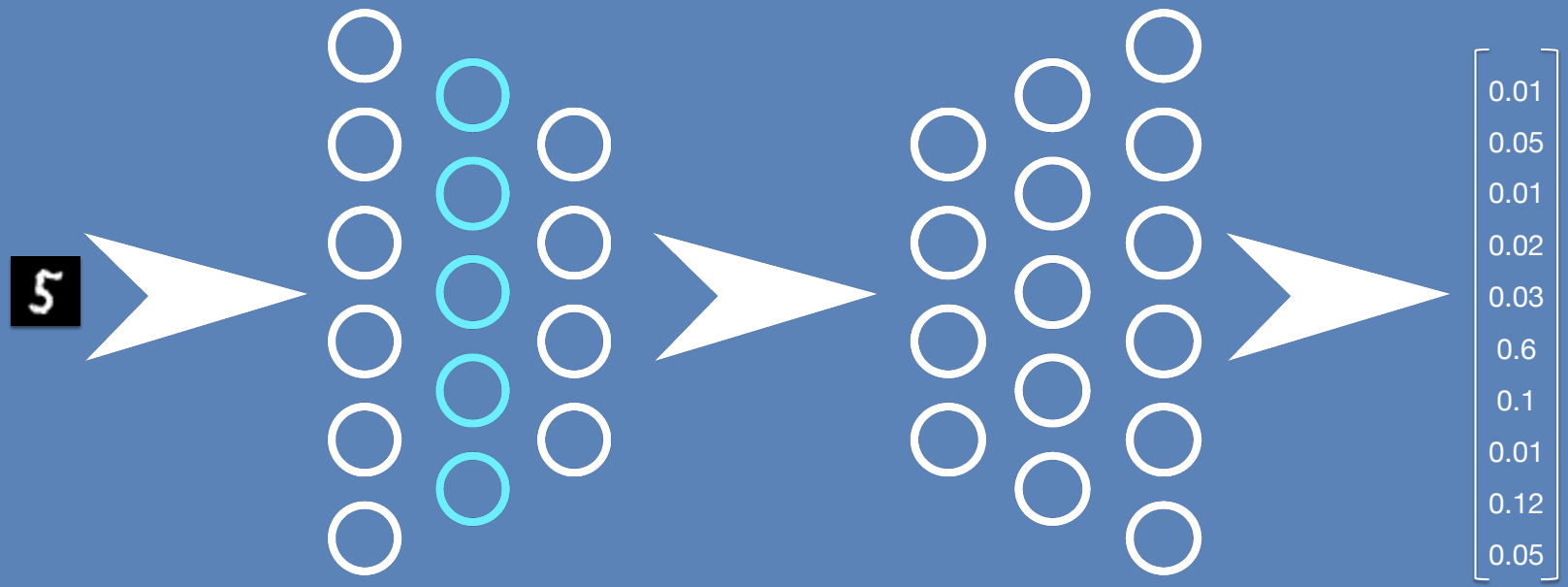
Forward Pass

Reminder



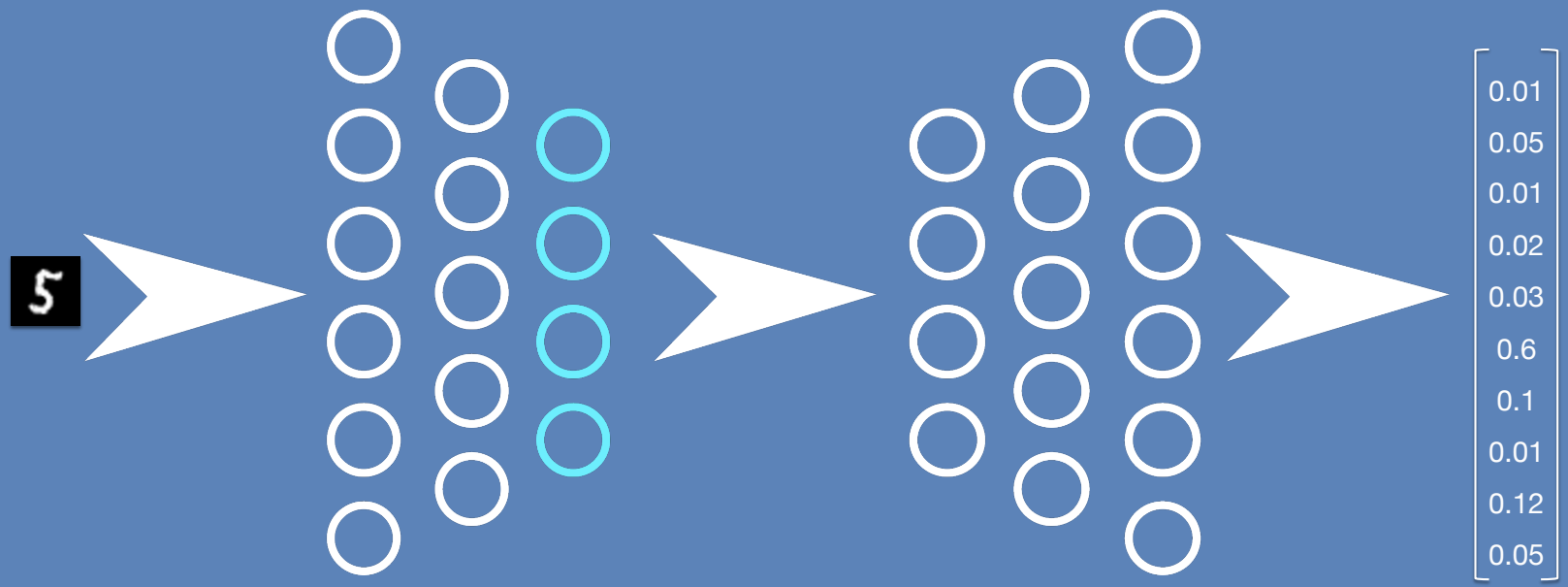
Forward Pass

Reminder



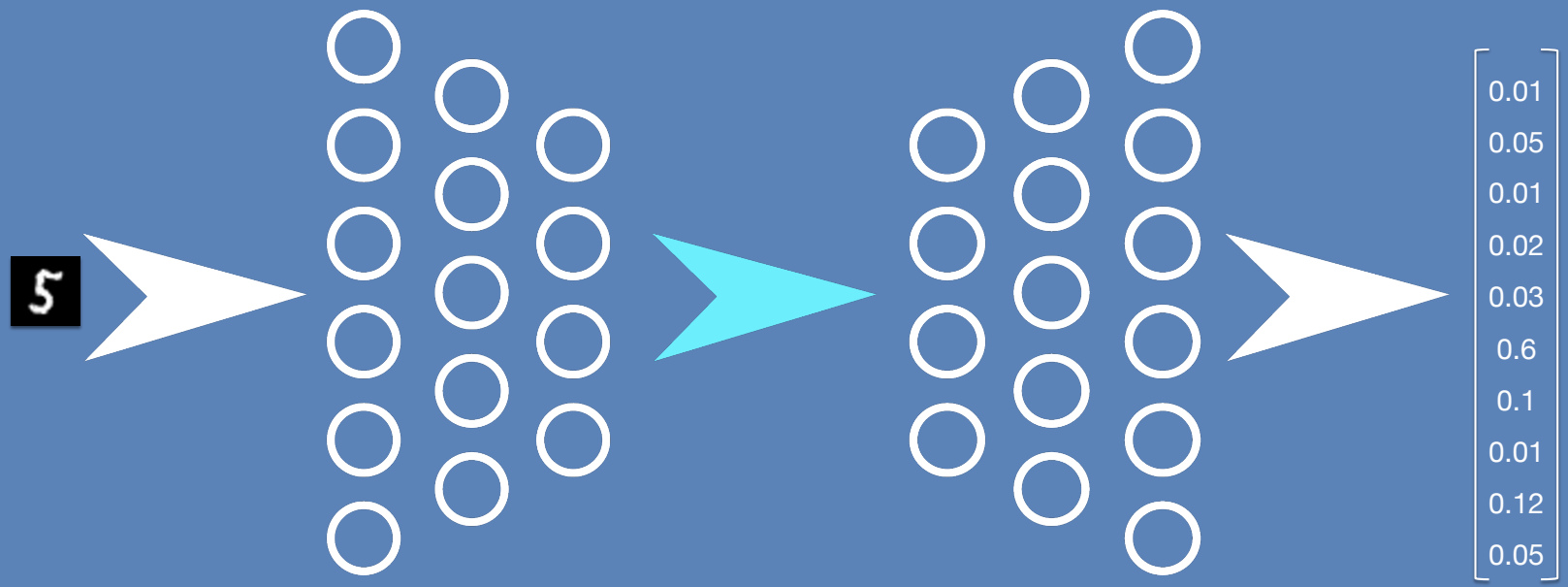
Forward Pass

Reminder



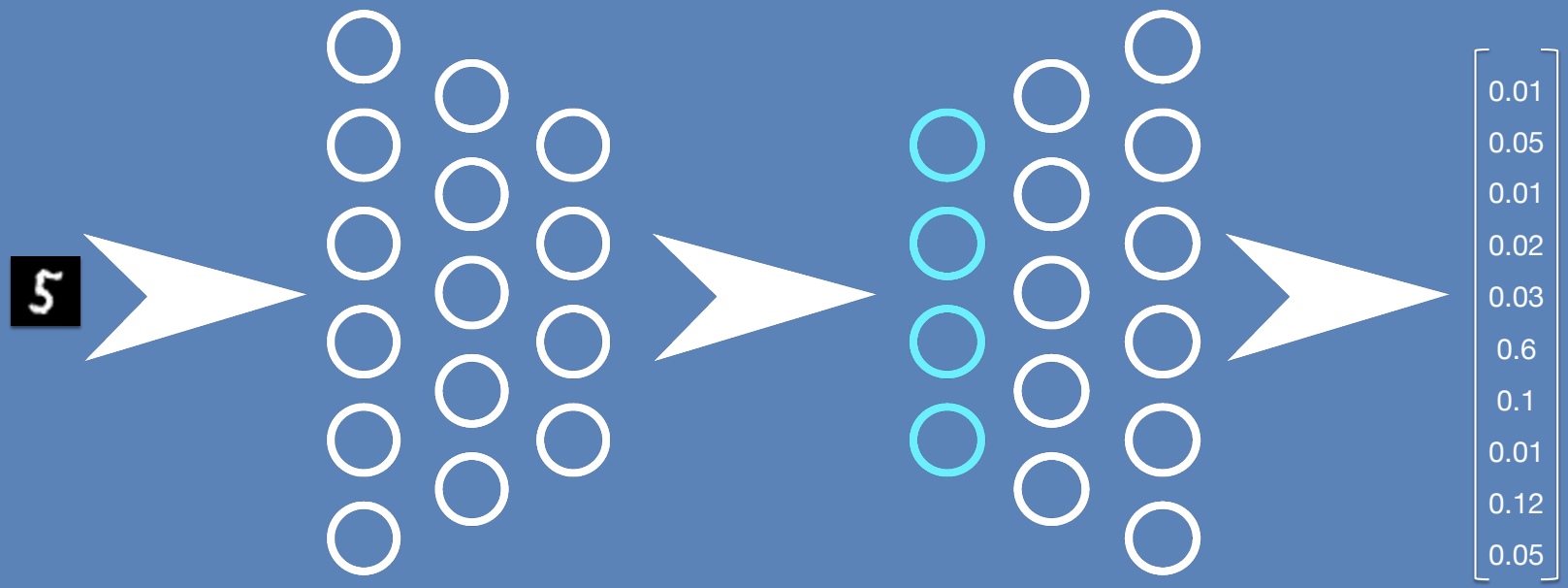
Forward Pass

Reminder



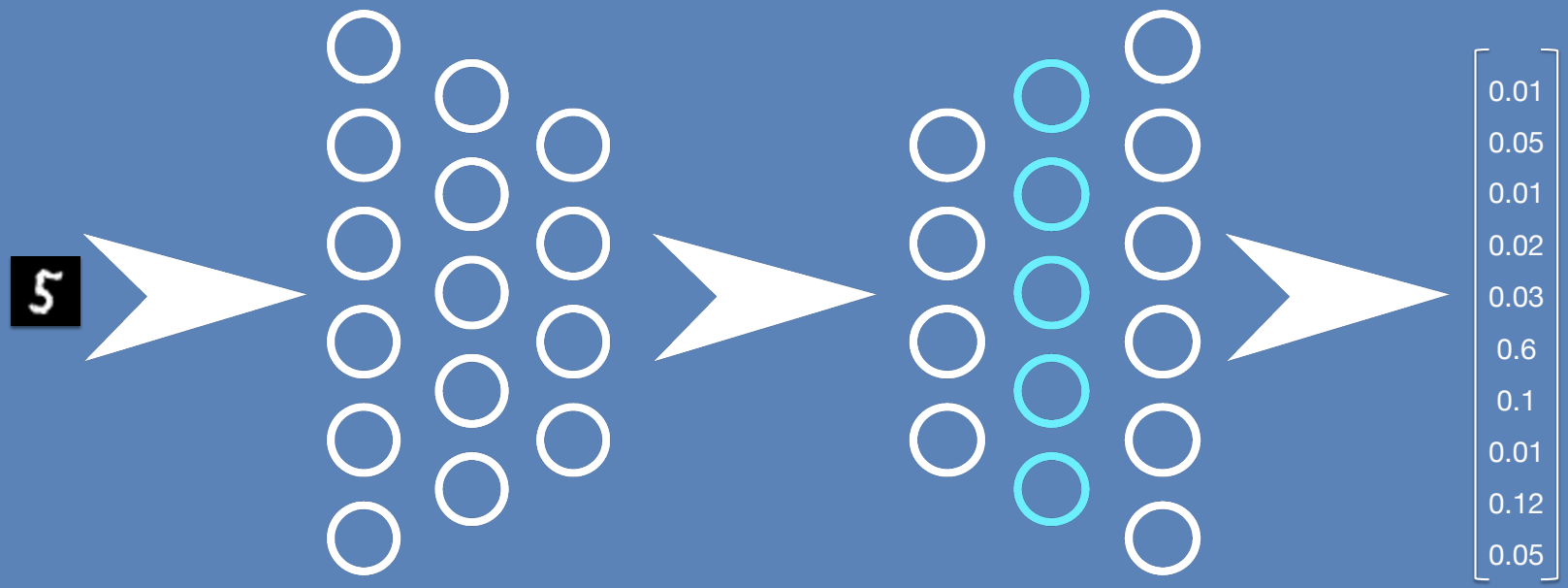
Forward Pass

Reminder



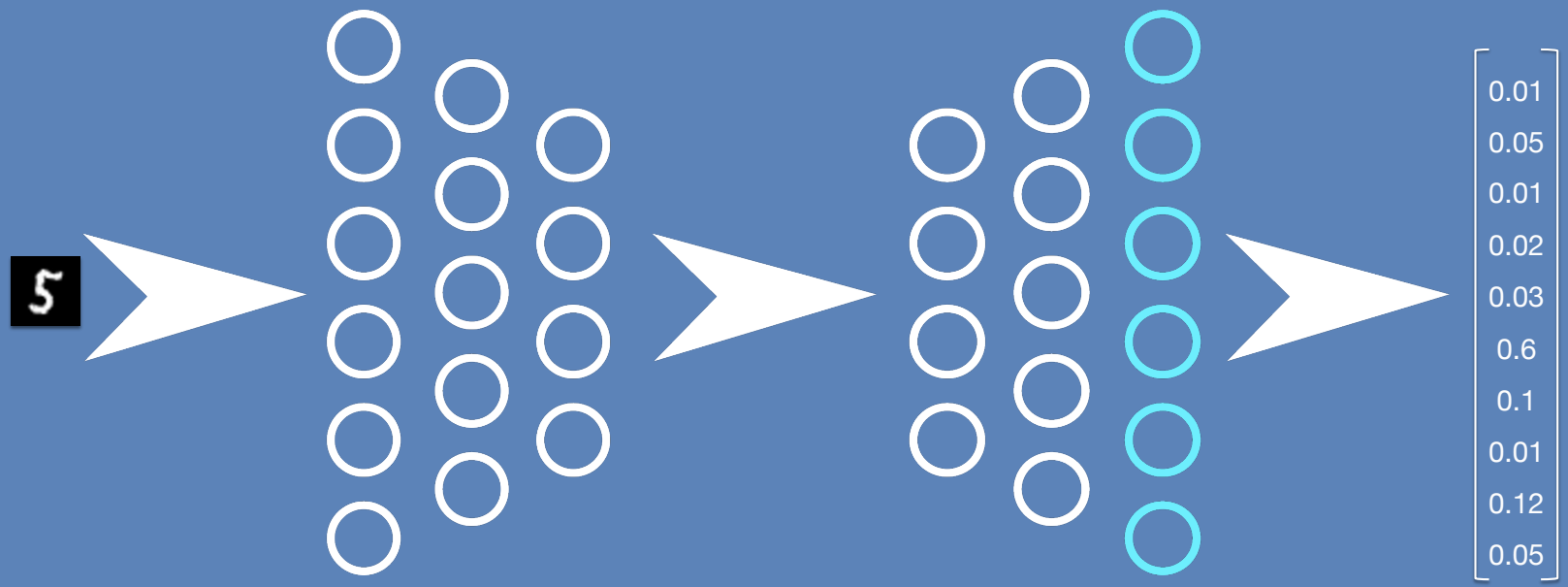
Forward Pass

Reminder



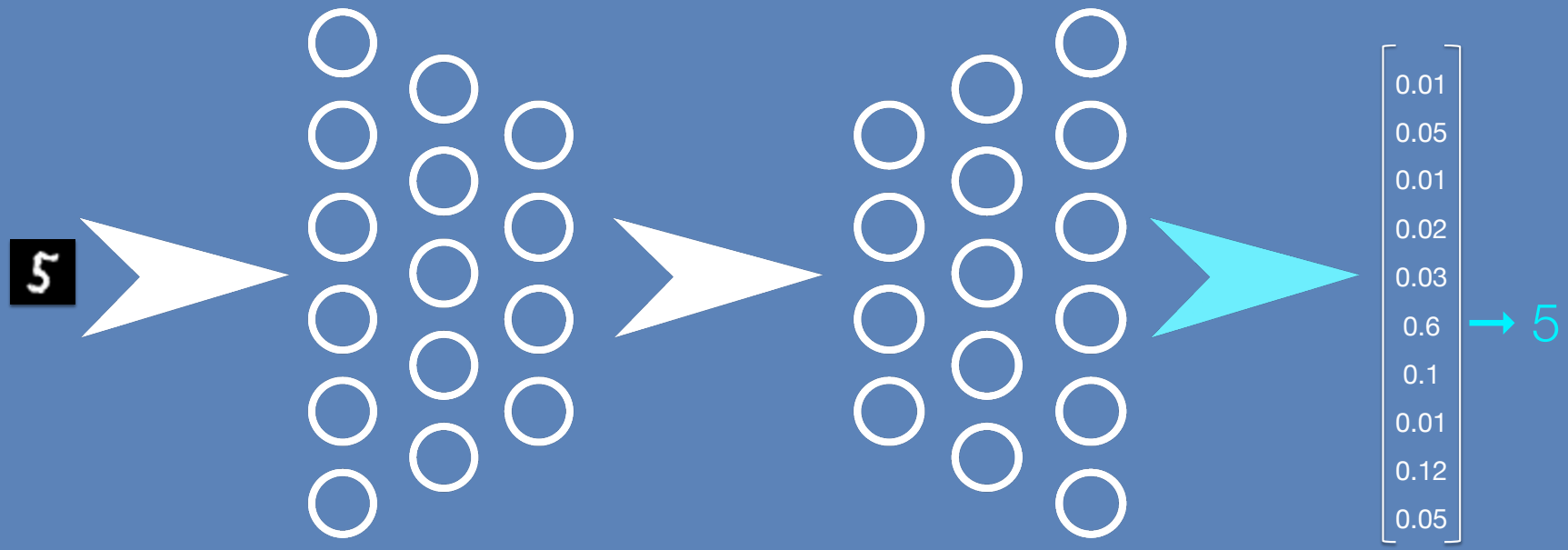
Forward Pass

Reminder



Forward Pass

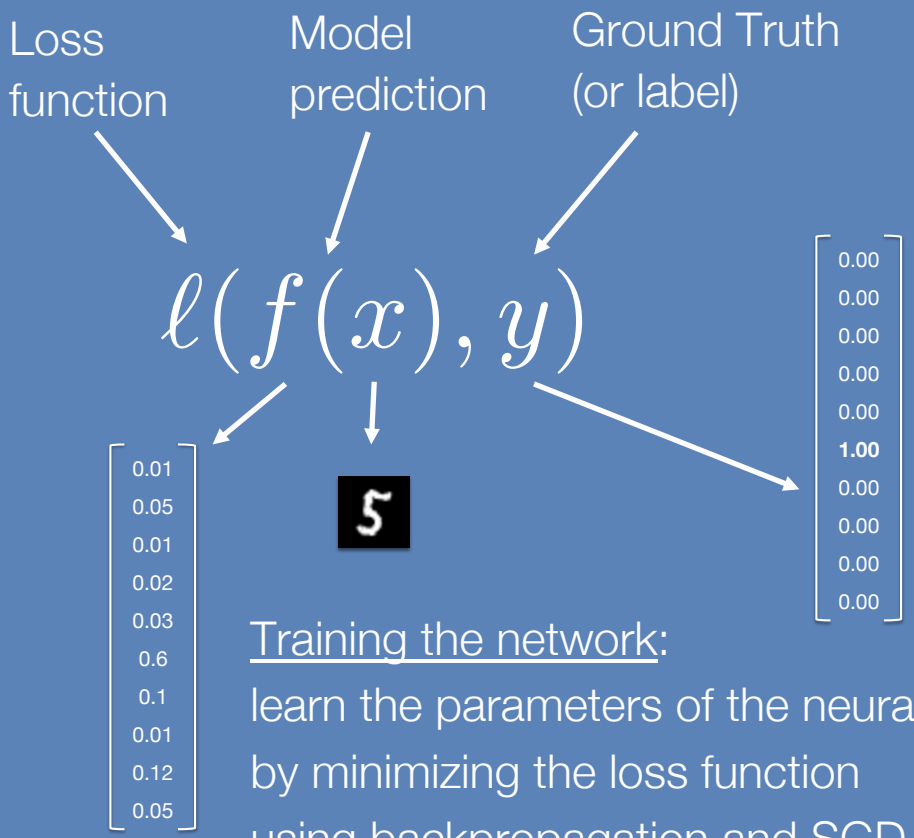
Reminder



Loss function

Reminder

The loss defines a distance between your prediction and the ground truth



Training the network:

learn the parameters of the neural networks by minimizing the loss function using backpropagation and SGD

Reminders

Class 1

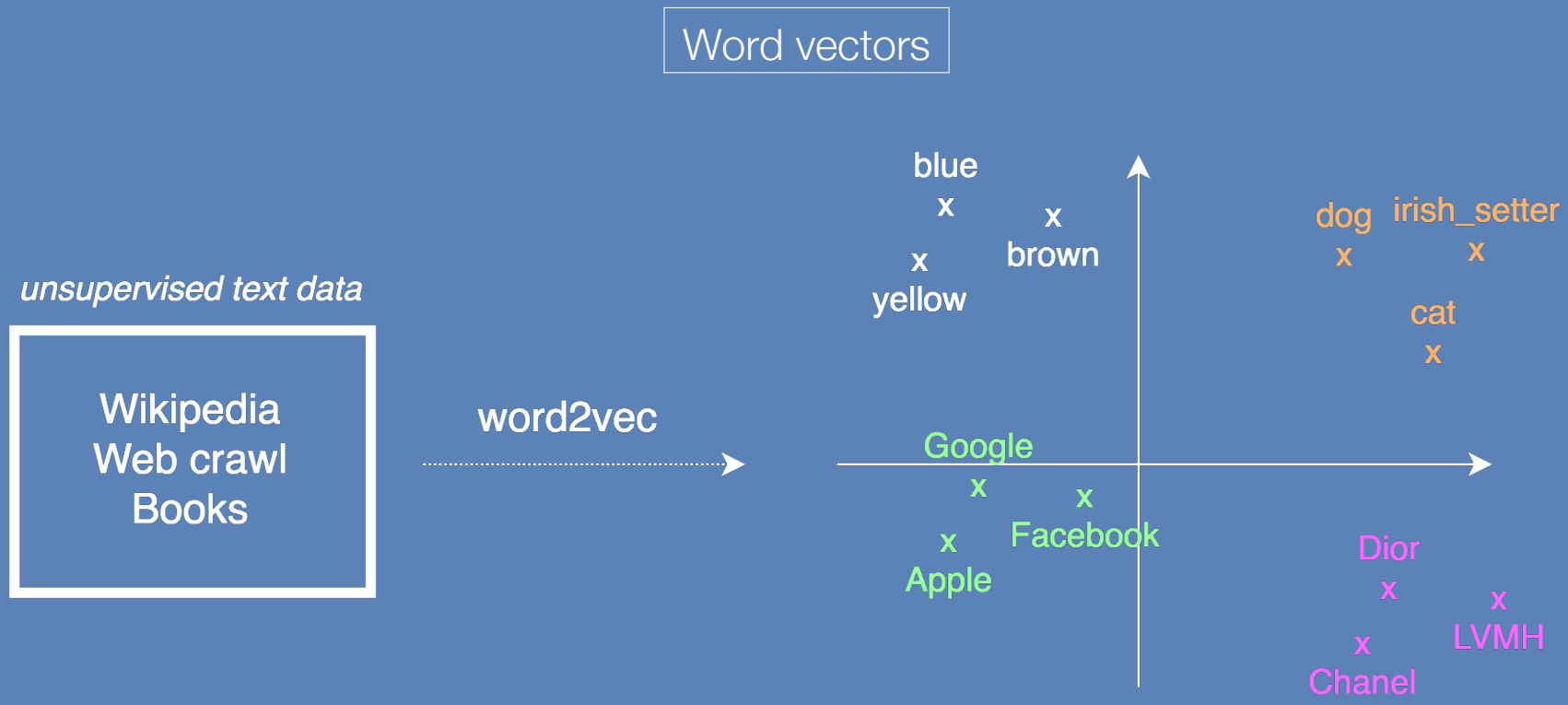
- Overview of some classical NLP tasks
- Word2vec: word embeddings
- Bag-of-words representations

Class 2

- Recurrent Neural Networks (RNNs, LSTMs)
- Language Modelling/Generation
- Encoders and decoders

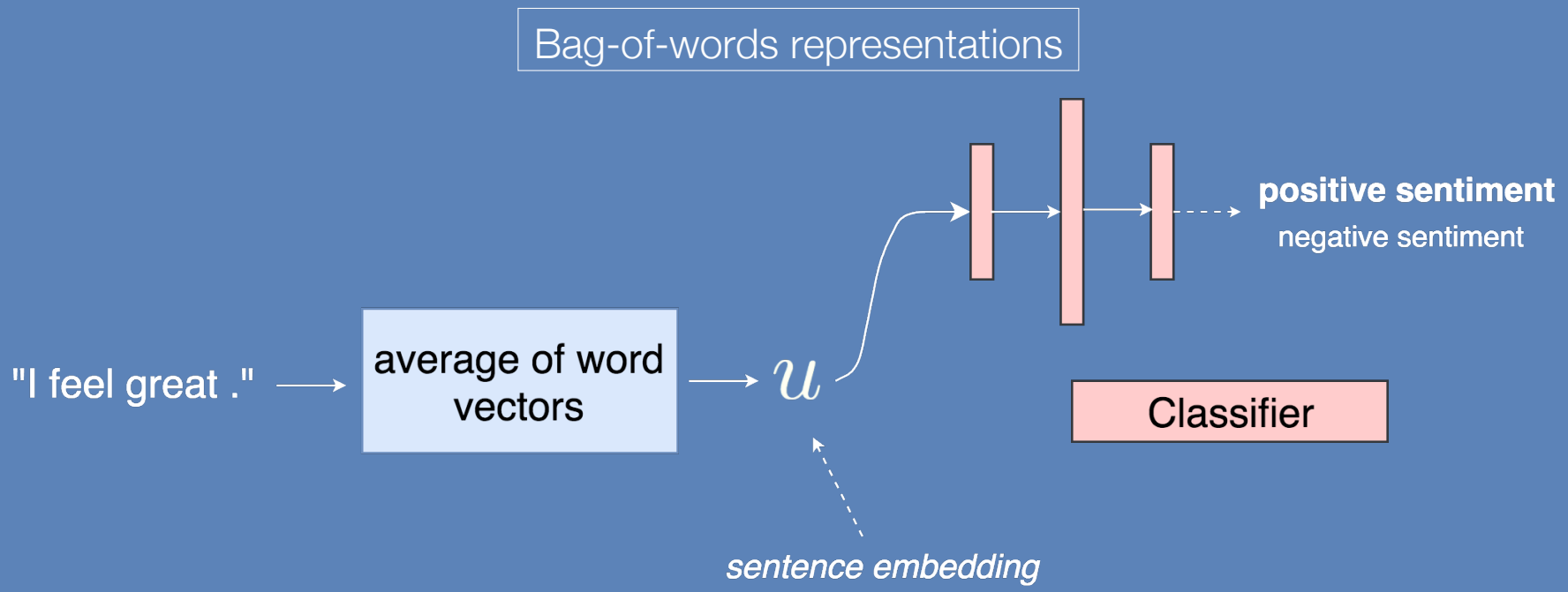
NLP 1/2

Reminder: word vectors



NLP 1/2

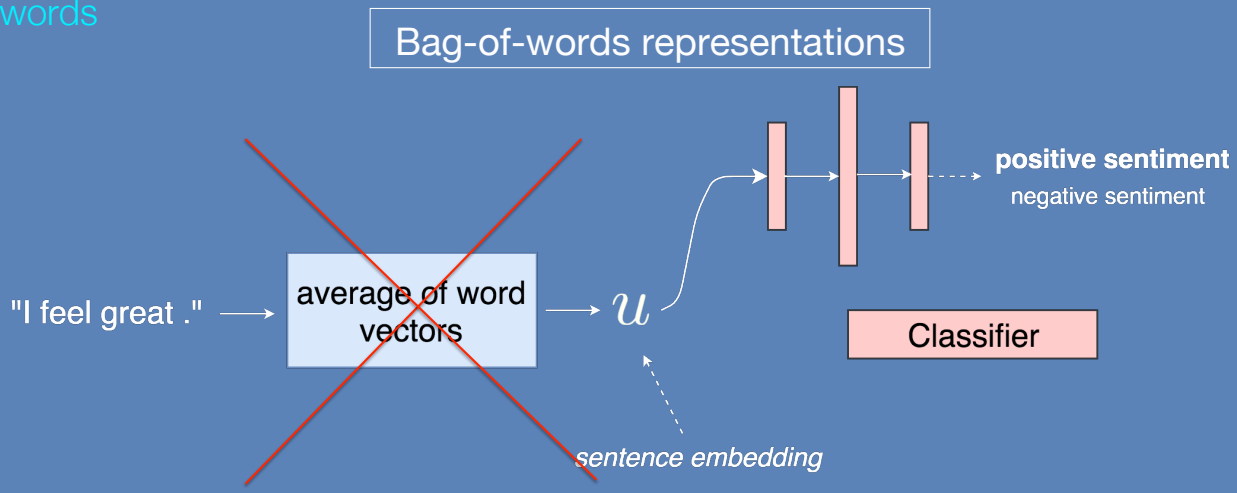
Reminder: bag-of-words



Continuous bag-of-words representations: average of word vectors

NLP 2/2

Beyond bag-of-words

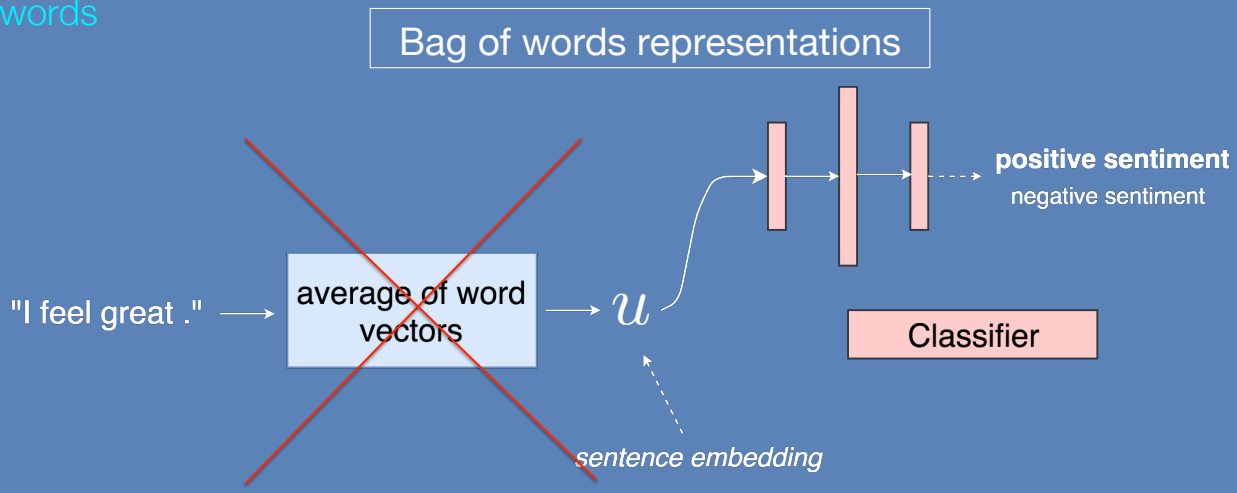


- Bag-of-words are limited (word order, context, …)

The cat is chasing the dog. versus The dog is chasing the cat.

NLP 2/2

Beyond bag-of-words



- Bag-of-words are limited (word order, context, …)
- Goal: capture more structure of input sentence
- Approach: sentence as a sequence of words

NLP 2/2

Recurrent Neural Networks

Three main types of neural networks:

- Multi-layer perceptron (MLP)
- Convolutional neural networks (CNNs)
- Recurrent Neural Networks (RNNs)

handle variable-length sequences

NLP 2/2

Recurrent Neural Networks

- Rich literature on sentence modelling: *RNNs, HMMs, CRFs* …
- Multiple points of view for a sentence: *set, sequence, tree* …
- RNNs: state-of-the-art in many NLP tasks

Outline

What we will see

- Part 1: Recurrent Neural Networks: RNNs (and LSTMs)
- Part 2: Language modelling and language generation
- Part 3: Encoder-Decoder applications

Chapter 1: RNNs

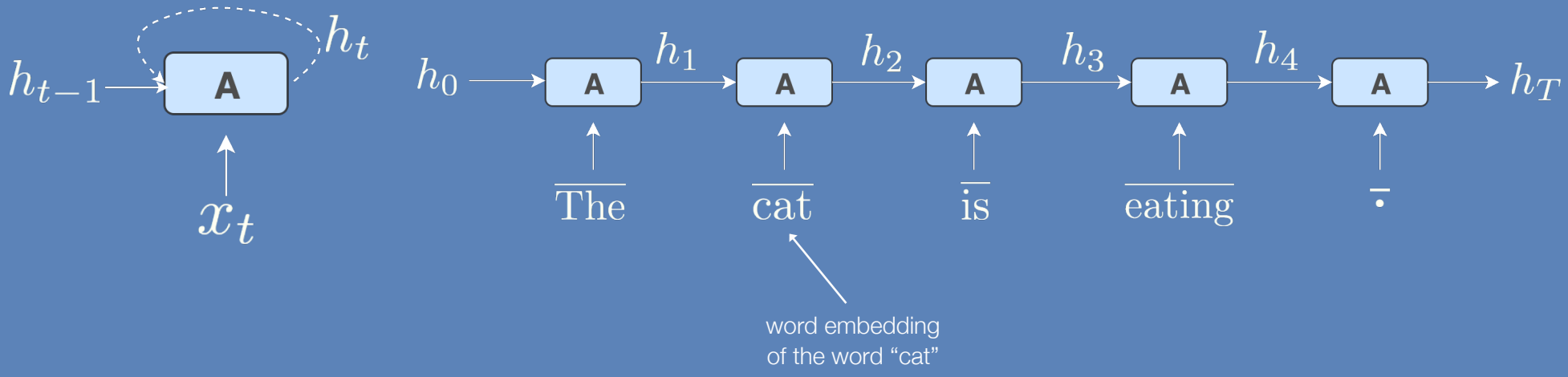
- Recurrent Neural Networks (RNN)
- 1st application: sentence encoder

Recurrent Neural Networks

(Elman, 1990)

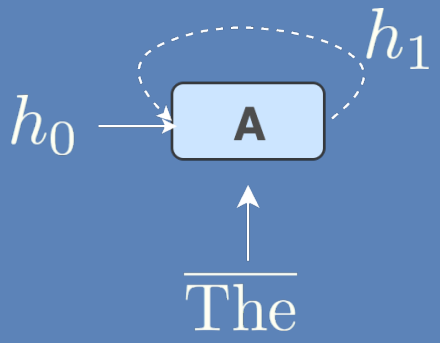
"recurrent" view of a RNN

"unfolded" view of a RNN

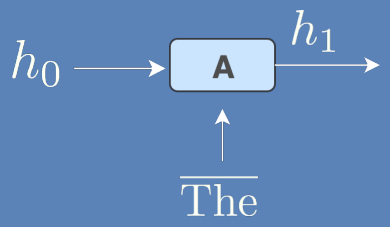


Recurrent Neural Networks in action

"recurrent" view of a RNN

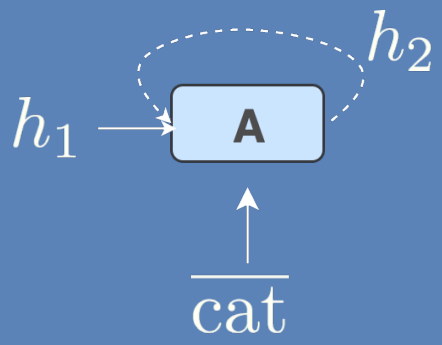


"unfolded" view of a RNN

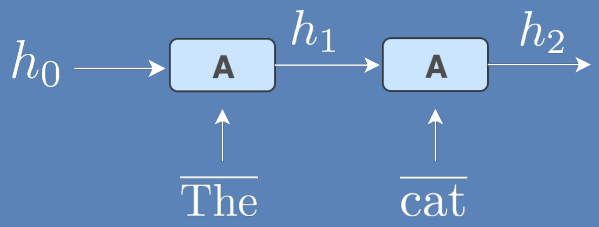


Recurrent Neural Networks in action

"recurrent" view of a RNN

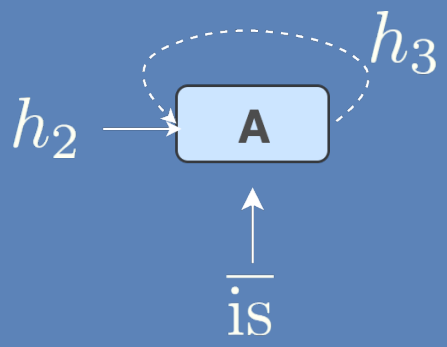


"unfolded" view of a RNN

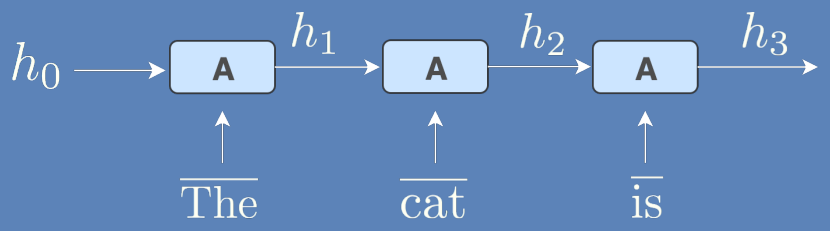


Recurrent Neural Networks in action

"recurrent" view of a RNN

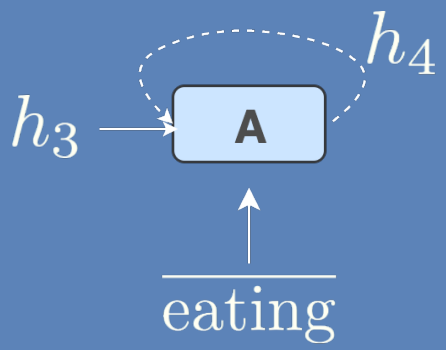


"unfolded" view of a RNN

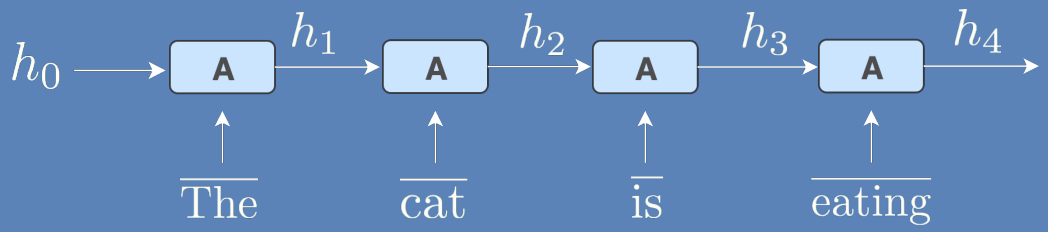


Recurrent Neural Networks in action

"recurrent" view of a RNN

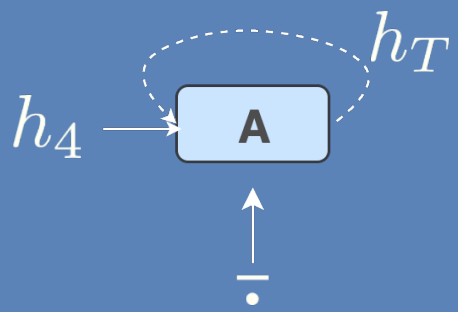


"unfolded" view of a RNN

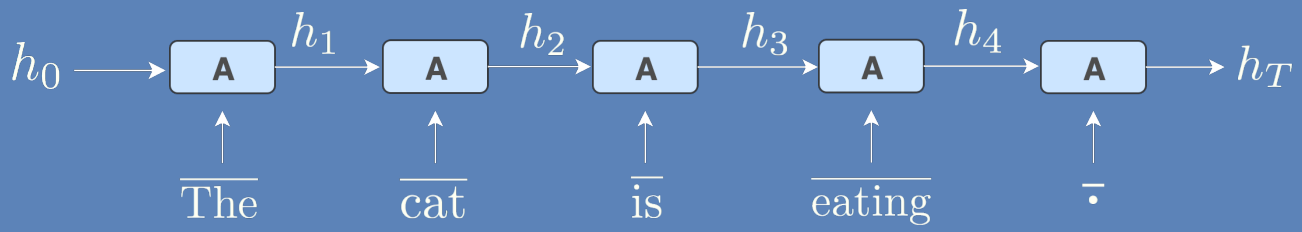


Recurrent Neural Networks in action

"recurrent" view of a RNN



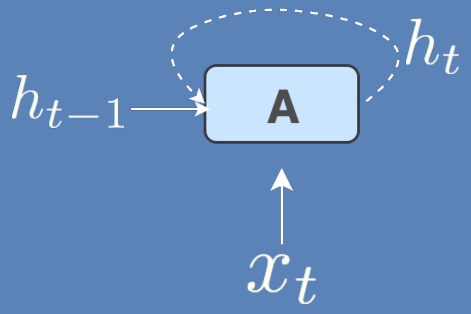
"unfolded" view of a RNN



Recurrent Neural Networks in action

inputs and outputs

"recurrent" view of a RNN

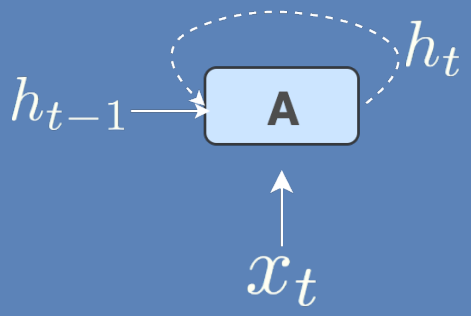


Three elements in the RNN:

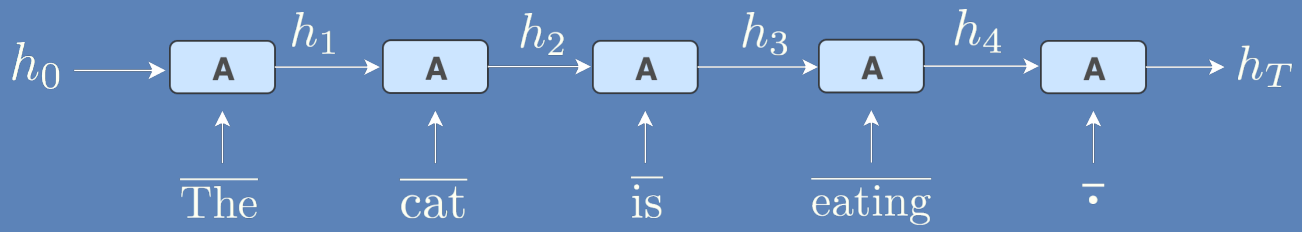
- Input sequence (The, cat, is, eating, .)
- Hidden states $(h_0, h_1, \ldots, h_T)$
- Cell function A : $h_t = A(x_t, h_{t-1})$

Recurrent Neural Networks in action

"recurrent" view of a RNN



"unfolded" view of a RNN

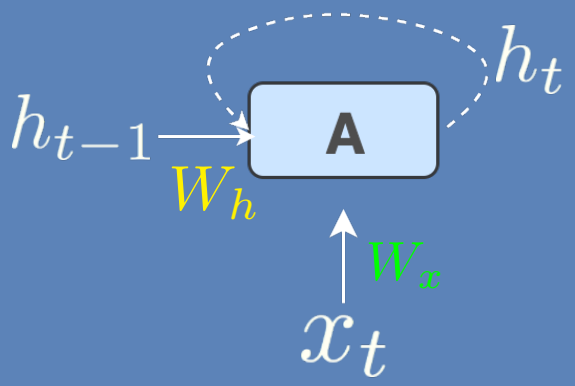


What is happening in the RNN cell?

Recurrent Neural Networks

RNN cell

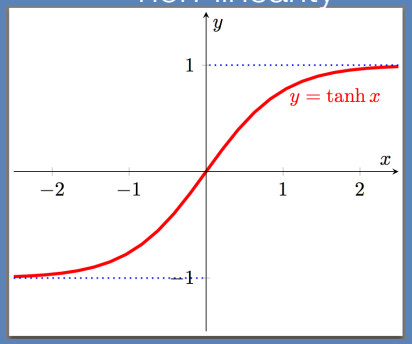
"recurrent" view of a RNN



What is happening in the RNN cell?

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

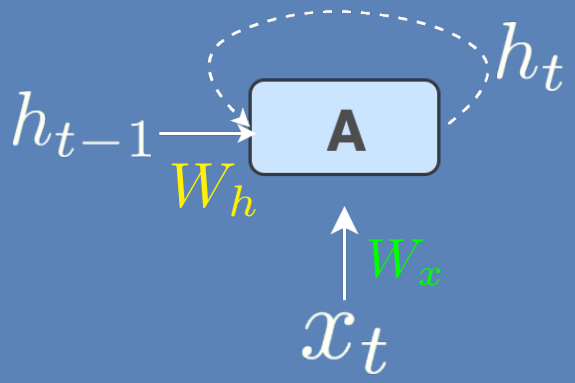
non-linearity



Recurrent Neural Networks

RNN cell

"recurrent" view of a RNN



What is happening in RNN cell?

The cell contains the parameters W_x W_h

d number of "hidden units" (e.g 512), size of h_t

W_x matrix of size (512, 300)

x_t vector of size (300)

} $W_x x_t$ of size (512)

W_h matrix of size (512, 512)

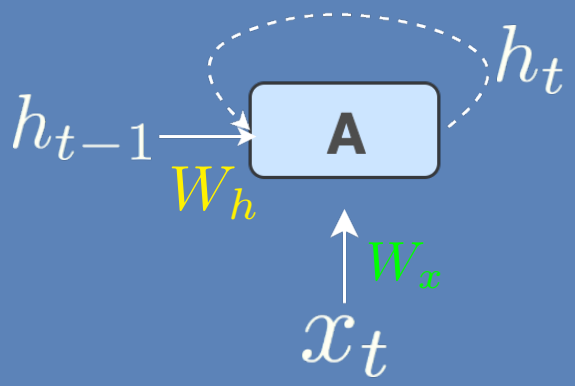
h_{t-1} vector of size (512)

} $W_h h_{t-1}$ of size (512)

Recurrent Neural Networks

RNN cell

"recurrent" view of a RNN



What is happening in RNN cell?

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

h_t of size (512)

Recurrent Neural Network in action

(Elman, 1990)

Initialize the hidden state h_0 (= vector of zeros)

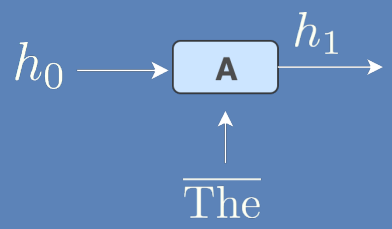
$h_0 \longrightarrow$

Recurrent Neural Network in action

(Elman, 1990)

stores information of "The"

$$h_1 = \tanh(W_x \overline{\text{The}} + W_h h_0)$$

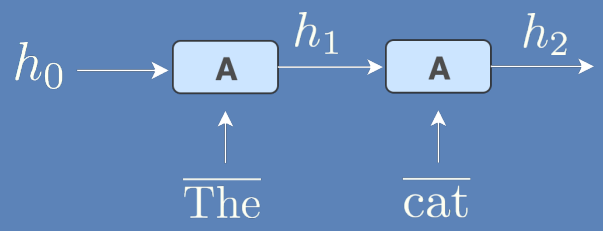


Recurrent Neural Network in action

(Elman, 1990)

stores information of "The cat"

$$h_2 = \tanh(W_x \overline{\text{cat}} + W_h h_1)$$

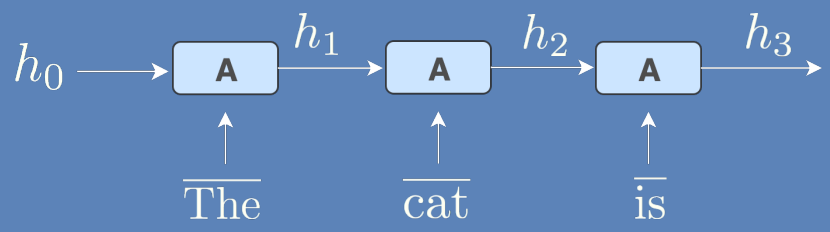


Recurrent Neural Network in action

(Elman, 1990)

stores information of "The cat is"

$$h_3 = \tanh(W_x \overline{\text{is}} + W_h h_2)$$

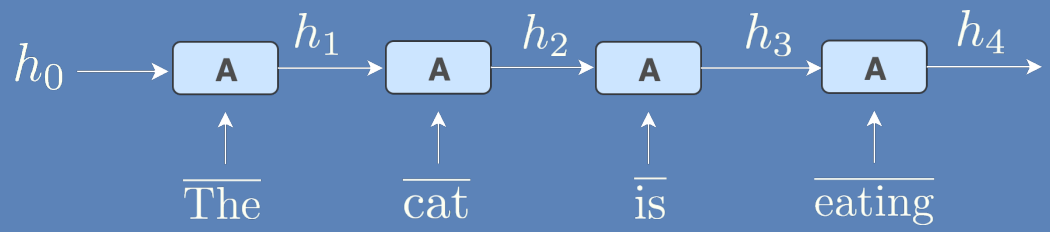


Recurrent Neural Network in action

(Elman, 1990)

stores information of “The cat is eating”

$$h_4 = \tanh(W_x \overline{\text{eating}} + W_h h_3)$$

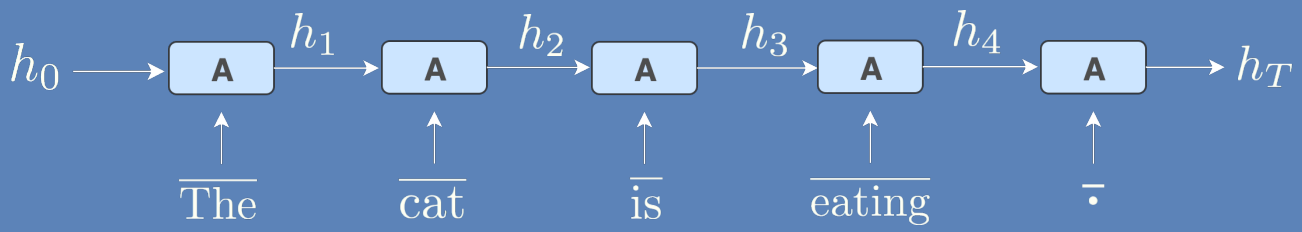


Recurrent Neural Network in action

(Elman, 1990)

stores information of "The cat is eating ."

$$h_T = \tanh(W_x \overline{\cdot} + W_h h_4)$$



Recurrent Neural Networks

Keras example

```
from keras.models import Sequential
from keras.layers import Embedding, RNN

model = Sequential()
model.add(Embedding(vocab_size, embed_dim))
model.add(RNN(nhid))
```

RNN Encoder

Recurrent Neural Networks (RNNs):

- Handle variable-length sequences
- Preserve word order
- Capture sequential information
- Encode sentence information in continous vectors

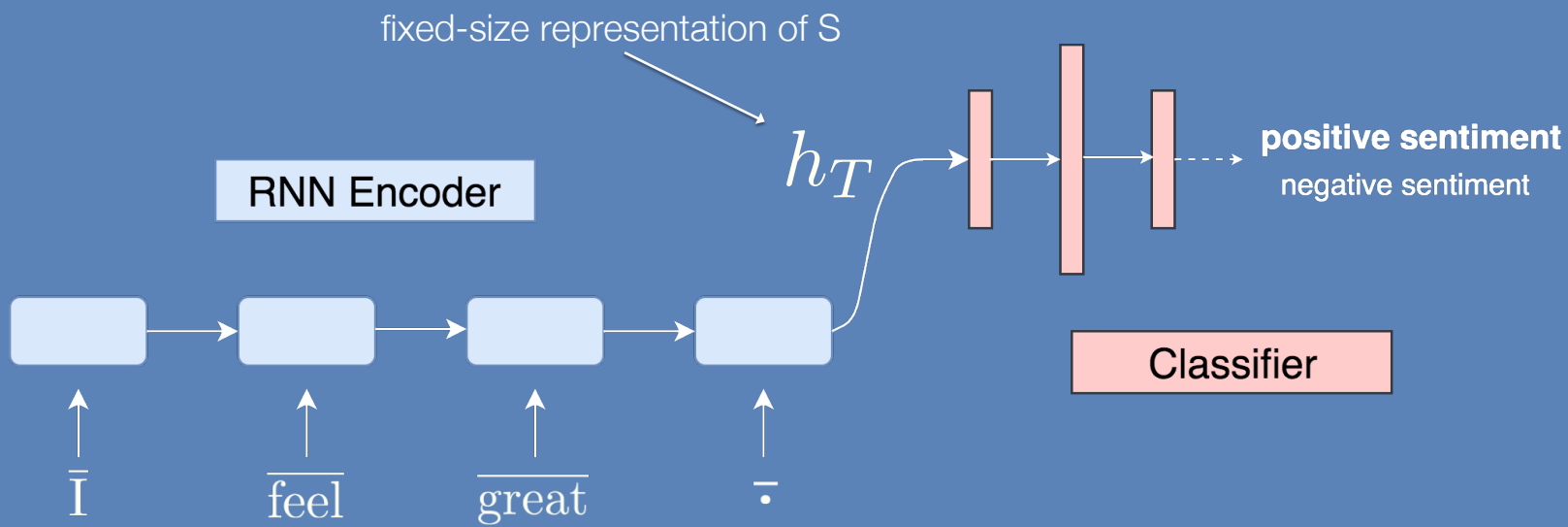
Applications

- Sentence classification
- Sentiment analysis
- Answer selection

RNN Encoder: sentence classification

Last hidden state h_T = summary of the input sentence S

Trained with backpropagation and SGD



RNN **Encoder**: Sentiment Analysis results

Stanford Sentiment Treebank dataset

Model	Fine-grained accuracy	Positive/negative accuracy
Bag-of-words	73.3	85.1
RNN	**79.0**	**86.1**

Input sentence	label
The story loses its bite in a last-minute happy ending that's even less plausible than the rest of the picture.	1
Watching Haneke's film is, aptly enough, a challenge and a punishment.	1
Morton is a great actress portraying a complex character, but Morvern Callar grows less compelling the farther it meanders from its shocking start.	3
Gosling provides an amazing performance that dwarfs everything else in the film.	5
A taut, intelligent psychological drama.	5

Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)

RNNs have a difficulty to summarize long sentences in a fixed-size vector.

RNN $$h_t = A_{\text{RNN}}(x_t, h_{t-1})$$

LSTMs are RNNs augmented with a memory cell.

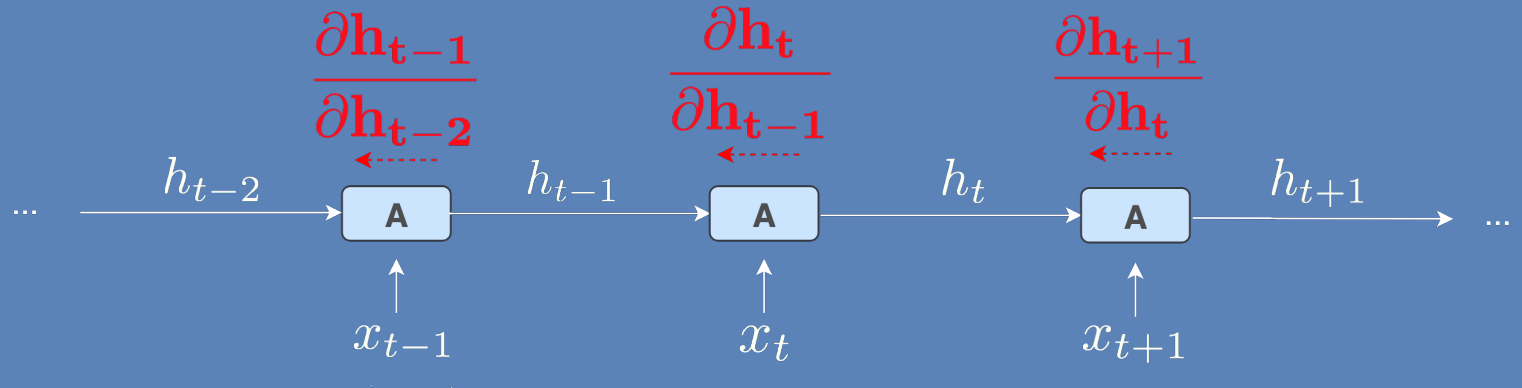
They can “remember” long-term dependencies.

memory cell

LSTM $$(h_t, c_t) = A_{\text{LSTM}}(x_t, (h_{t-1}, c_{t-1}))$$

Back-Propagation Through Time

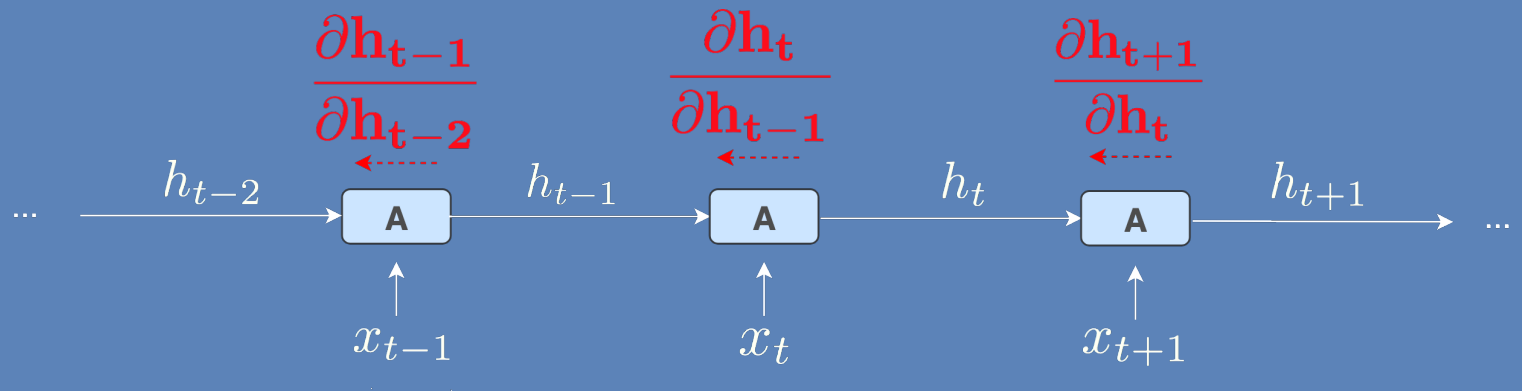
Hochreiter (1991)



- In that context: backpropagation = back-propagation through time (BPTT)
- Gradients are backpropagated using the chain rule

The problem of vanishing gradient

(Hochreiter and Schmidhuber, 1997)

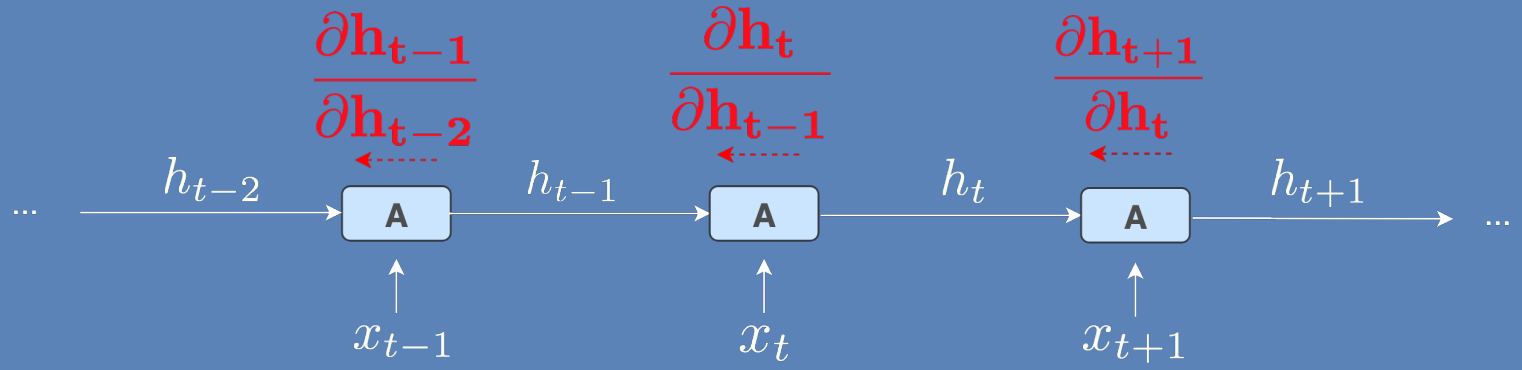


- Vanishing gradient = large decrease (to zero) of the norm of the gradient during training

--> the network does not learn long-term dependencies

The problem of vanishing gradient

(Hochreiter and Schmidhuber, 1997)



output (time T)

$$\frac{\partial o_T}{\partial h_1} = \frac{\partial o_T}{\partial h_T} \underbrace{\frac{\partial h_T}{\partial h_{T-1}} \cdots \frac{\partial h_2}{\partial h_1}}$$

Product of (T-1) jacobian matrices

The problem of vanishing gradient

(Hochreiter and Schmidhuber, 1997)

output (time T)

$$\frac{\partial \mathbf{o_T}}{\partial \mathbf{h_1}} = \frac{\partial \mathbf{o_T}}{\partial \mathbf{h_T}} \underbrace{\frac{\partial \mathbf{h_T}}{\partial \mathbf{h_{T-1}}} \cdots \frac{\partial \mathbf{h_2}}{\partial \mathbf{h_1}}}$$

Product of (T-1) jacobian matrices

$$\frac{\partial \mathbf{h_{t+1}}}{\partial \mathbf{h_t}} = \frac{\partial \tanh(\mathbf{W_x x_t} + \mathbf{W_h h_t})}{\partial \mathbf{h_t}}$$

$$= \mathbf{W_h} \tanh'(\mathbf{W_x x_t} + \mathbf{W_h h_t})$$

In the same way a product of (T-1) real numbers can shrink to zero or explode to infinity, so does this product of matrices

The problem of vanishing gradient

(Hochreiter and Schmidhuber, 1997)

$$\frac{\partial \mathbf{o_T}}{\partial \mathbf{h_1}} = \frac{\partial \mathbf{o_T}}{\partial \mathbf{h_T}} \underbrace{\frac{\partial \mathbf{h_T}}{\partial \mathbf{h_{T-1}}} \cdots \frac{\partial \mathbf{h_2}}{\partial \mathbf{h_1}}}$$

Product of (T-1) jacobian matrices

$$\frac{\partial \mathbf{h_{t+1}}}{\partial \mathbf{h_t}} = \frac{\partial \tanh(\mathbf{W_x x_t} + \mathbf{W_h h_t})}{\partial \mathbf{h_t}}$$

$$= \mathbf{W_h} \tanh'(\mathbf{W_x x_t} + \mathbf{W_h h_t})$$

- Two factors affect the magnitude of the gradients (the weights and activation tanh)
- If either of these factors is smaller than 1, then the gradients may in time.
  For example, the tanh derivative is < 1 for all inputs except 0.

Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)

RNNs have a difficulty to summarize long sentences in a fixed-size vector.

RNN $$h_t = A_{\text{RNN}}(x_t, h_{t-1})$$

LSTMs are RNNs augmented with a memory cell.

They can “remember” long-term dependencies.

memory cell

LSTM $$(h_t, c_t) = A_{\text{LSTM}}(x_t, (h_{t-1}, c_{t-1}))$$

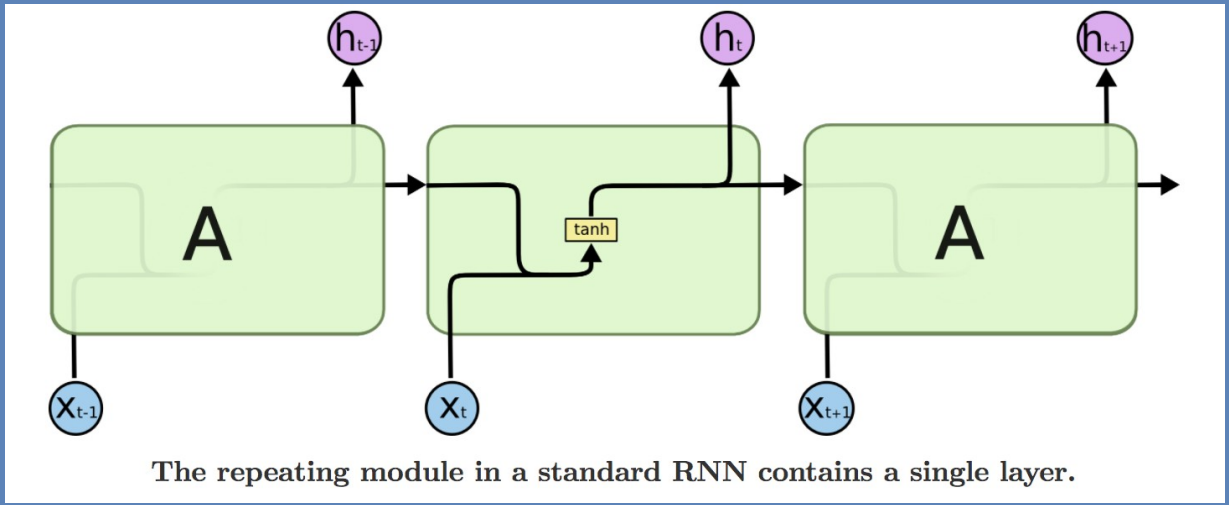
Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)

- An LSTM is a type of RNN, whose recurrent cell use **gates** (f_t, i_t, o_t) and an additional **memory cell** (c_t) to account for longer temporal dependencies.
- The gates protect and control the memory of the cell state.
- Given a sequence of inputs ($x_1, \ldots, x_T$), the LSTM computes $((h_1, c_1) \ldots, (h_T, c_T))$ using gates at each time step :
  - **Forget gate** f_t : controls what the memory cell c_t forgets.
  - **Input gate** i_t : controls what is added to the memory cell c_t .
  - **Output gate** o_t : controls the computation of the ouput h_t from c_t .

Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)



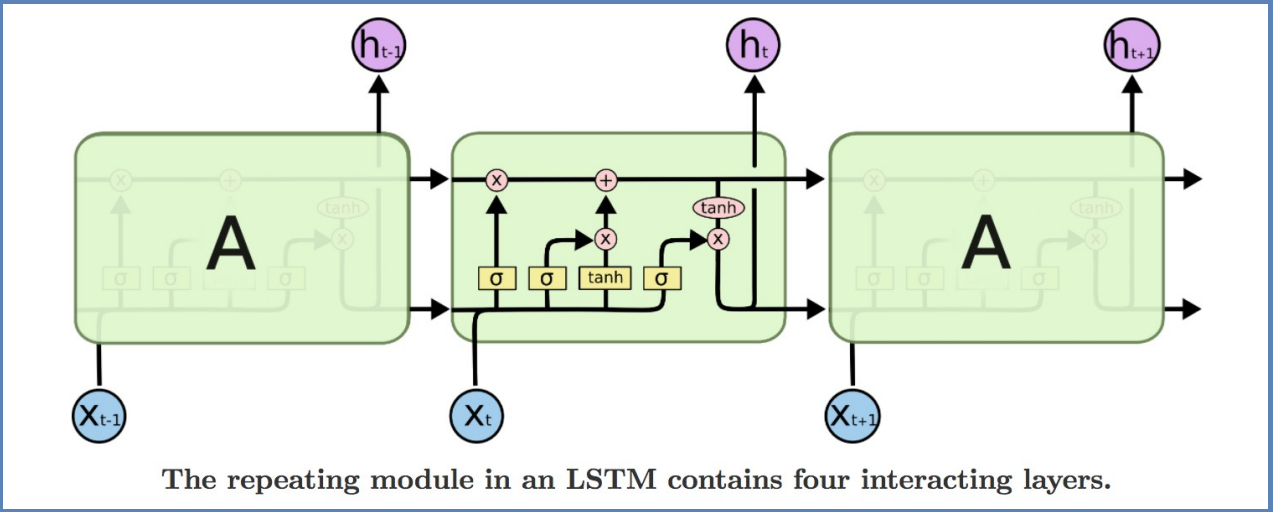
The repeating module in a standard RNN contains a single layer.

Equations of RNN

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)



The repeating module in an LSTM contains four interacting layers.

Equations of LSTM

Gates

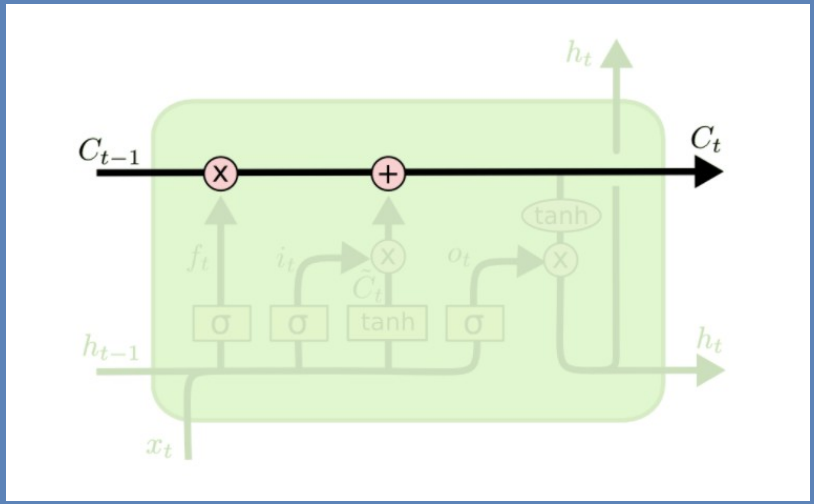
$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

Updates

$$\tilde{c_t} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$\mathbf{c_t = f_t c_{t-1} + i_t \tilde{c}_t}$$
$$\mathbf{h_t = o_t \tanh(c_t)}$$

Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)



Equations of LSTM

Gates

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

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

$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$

Updates

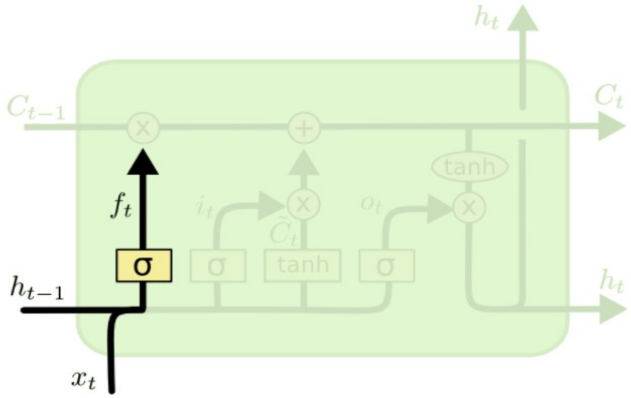
$\tilde{c_t} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$

$\mathbf{c_t = f_t c_{t-1} + i_t \tilde{c_t}}$

$\mathbf{h_t = o_t \tanh(c_t)}$

Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Equations of LSTM

Gates

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

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

$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$

Updates

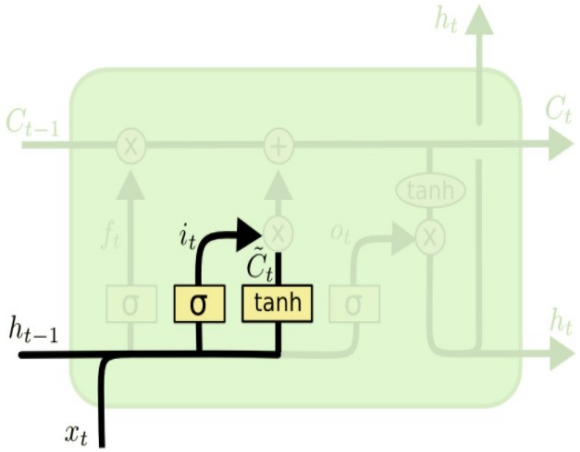
$\tilde{c_t} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$

$\mathbf{c_t = f_t c_{t-1} + i_t \tilde{c_t}}$

$\mathbf{h_t = o_t \tanh(c_t)}$

Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Equations of LSTM

Gates

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

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

$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$

Updates

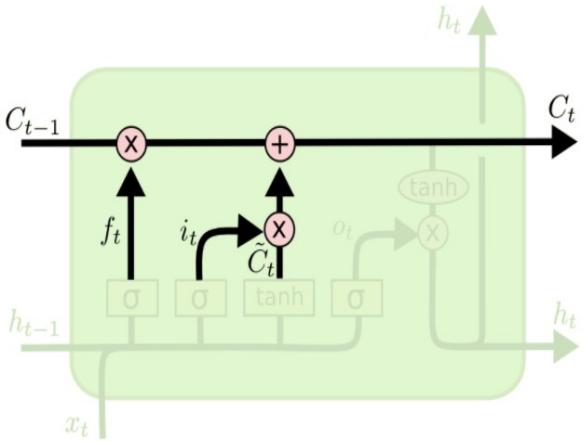
$\tilde{c_t} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$

$\mathbf{c_t = f_t c_{t-1} + i_t \tilde{c_t}}$

$\mathbf{h_t = o_t \tanh(c_t)}$

Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Equations of LSTM

Gates

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

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

$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$

Updates

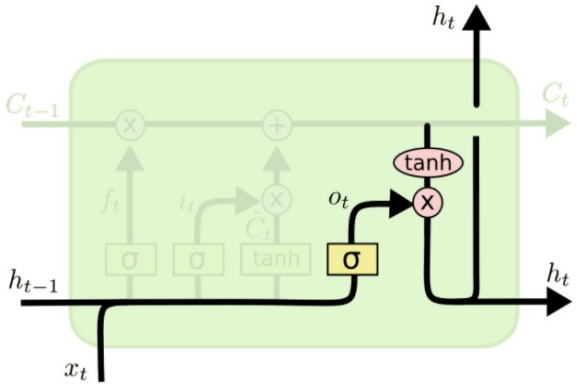
$\tilde{c_t} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$

$\mathbf{c_t = f_t c_{t-1} + i_t \tilde{c_t}}$

$\mathbf{h_t = o_t \tanh(c_t)}$

Long Short Term Memory (LSTMs)

(Hochreiter and Schmidhuber, 1997)



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Equations of LSTM

Gates

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

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

$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$

Updates

$\tilde{c_t} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$

$\mathbf{c_t = f_t c_{t-1} + i_t \tilde{c_t}}$

$\mathbf{h_t = o_t \tanh(c_t)}$

LSTM – vanishing gradient

(Hochreiter and Schmidhuber, 1997)

Reminder:

- Two factors affect the magnitude of the gradients (the weights and activation tanh)
- If either of these factors is smaller than 1, then the gradients may vanish in time. For example, the tanh derivative is < 1 for all inputs except 0.

LSTM – vanishing gradient

(Hochreiter and Schmidhuber, 1997)

$$c_t = f_t\, c_{t-1} + i_t\, \tilde{c}_t$$

- In the recurrency of the LSTM the activation function is the identity with a derivative of 1.0. So, the backpropagated gradient neither vanishes or explodes when passing through, but remains constant.

- The effective weight of the recurrency is equal to the forget gate activation. The forget gate thus controls the amount of gradient that is backpropagated.

Long Short Term Memory (LSTMs)

Keras example


```
from keras.models import Sequential
from keras.layers import Embedding, LSTM

model = Sequential()
model.add(Embedding(vocab_size, embed_dim))
model.add(LSTM(nhid))
```

Chapter 2: Language generation

- Language modelling
- Language generation

Language modelling - motivations

What does it mean that a machine understands language?

Language modelling - motivations

It's about telling how likely a sentence is.

A **language model** is a probability distribution over sequences of words (trained on large corpora)

$P(\text{the, cat, is, eating, .}) \longrightarrow$ high (likely)

$P(\text{cat, the, eating, ., is}) \longrightarrow$ low (unlikely)

$P(\text{the, cat, is, coding, in, python, .}) \longrightarrow$ medium

Language modelling

Conditional probabilities

$$P(A, B) = P(A|B)P(B)$$

The probability of a sentence is the product of conditional probabilities:

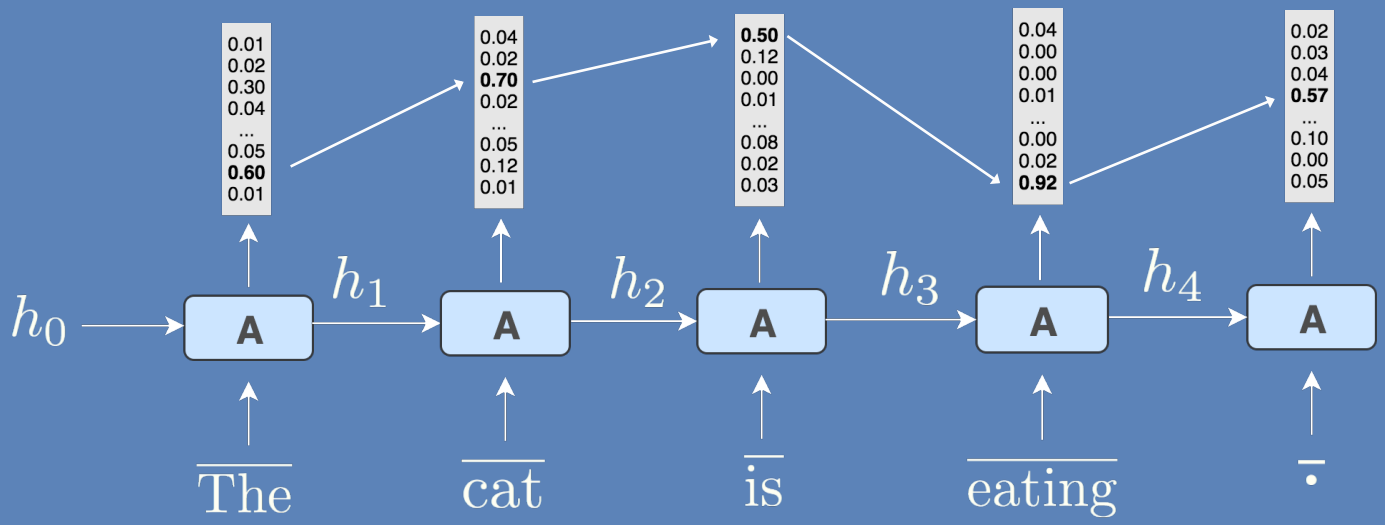
$$P(w_1, \ldots, w_T) = \prod_{t=1}^{T} P(w_t|w_1, \ldots, w_{t-1})$$

$$P(\text{the, cat, is, eating, .}) = P(.|\text{the, cat, is, eating})P(\text{eating}|\text{the, cat, is})P(\text{is}|\text{the, cat})P(\text{cat}|\text{the})P(\text{the})$$

Language modelling - RNN LM

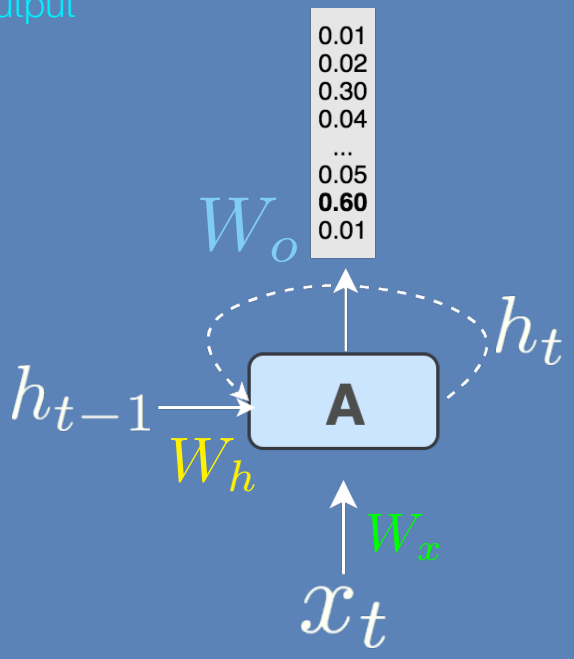
RNN language model

$$P(w_1, \ldots, w_T) = \prod_{t=1}^{T} P(w_t | w_1, \ldots, w_{t-1})$$



Recurrent Neural Networks

Output



“recurrent” view of a RNN

$$\begin{matrix}0.01\\0.02\\0.30\\0.04\\...\\0.05\\\mathbf{0.60}\\0.01\end{matrix} = \text{softmax}(W_o\, h_t)$$

W_o matrix of size (V, 512)

h_t vector of size (512)

$W_o\, h_t$ of size (V)

Language modelling - RNN

Summary

- Language model: trained on a large amount of unsupervised data
- Learns to predict the next word in a sentence using the previous words
- Able to compute the likelihood of a sentence

Evaluate language model

Perplexity

In Information Theory, **perplexity** is a measure of how well a probability distribution or probability model predicts a sample

$$\text{Perplexity(S)} = \mathbf{2^{H(S)}} \qquad \mathbf{H(S)} = -\frac{\mathbf{1}}{\mathbf{N}} \log \mathbf{P(w_1, \ldots, w_N)}$$

$$\text{Perplexity(S)} = \sqrt[n]{\frac{1}{\prod_{i=1}^{N} P(w_i | w_1, \ldots, w_{i-1})}}$$

Language generation - RNN

From language modelling to language generation

Language modelling

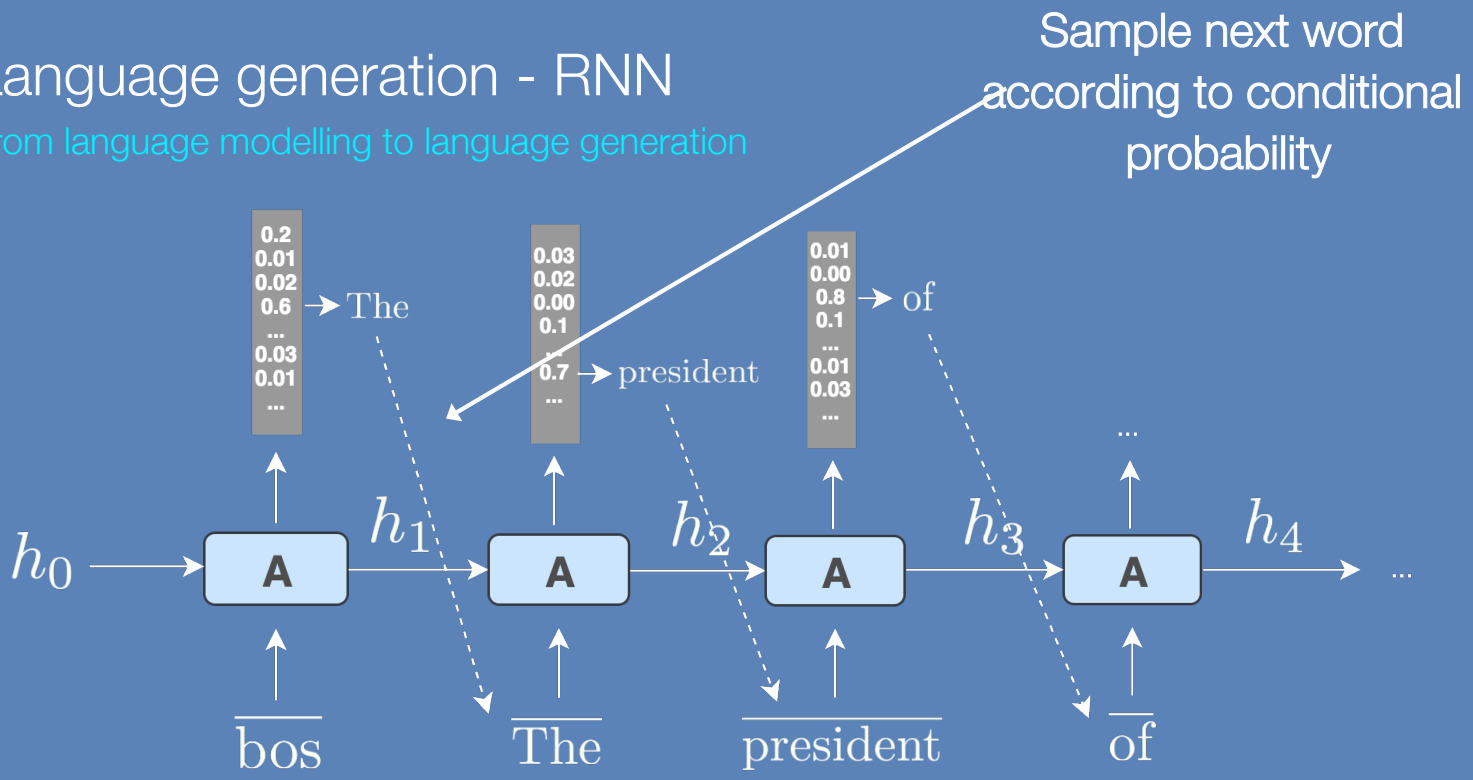
gives us the probability of the next word given the previous words.

Language generation

Can you guess a simple way to generate a sentence?

Language generation - RNN

From language modelling to language generation



Language modelling - RNN

Shakespeare language modelling (Karpathy)

- Concatenate all the work from *Shakespeare* (4.4 Mb file)
- Learn a language model
- Generate sentences using the language model

Language modelling - RNN

Shakespeare language modelling (Karpathy)

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.
```

Language modelling - RNN

Shakespeare language modelling (Karpathy)

```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.
```

Chapter 3: Encoder-decoder applications

- Sequence to sequence
- Image to sequence

Conditional language modelling

Question answering

Many generation tasks can be seen as conditional language models

Example 1: How likely is this sentence as an answer to the question?

Q. : “Who is the President of France?”

- Likely answer : “Emmanuel Macron is the President.”
- Unlikely answer : “Donald Trump is the President.”

Conditional language modelling

Image captioning

Many generation tasks can be seen as conditional language models

Example 2: How likely is this sentence given this view?

- Likely: “Two dolphins are diving.”
- Unlikely: “Two birds are flying.”

Conditional language modelling

Neural machine translation

Many generation tasks can be seen as conditional language models

Example 3: How likely is this sentence as a translation of the source sentence?

Source: “The cat is eating.”

- Likely answer : “Le chat mange.”
- Unlikely answer : “Le chat boit.”

Sequence to sequence learning

Encoder-Decoder architecture

RNN Encoder

can capture the information of an input sentence into a fixed-size vector

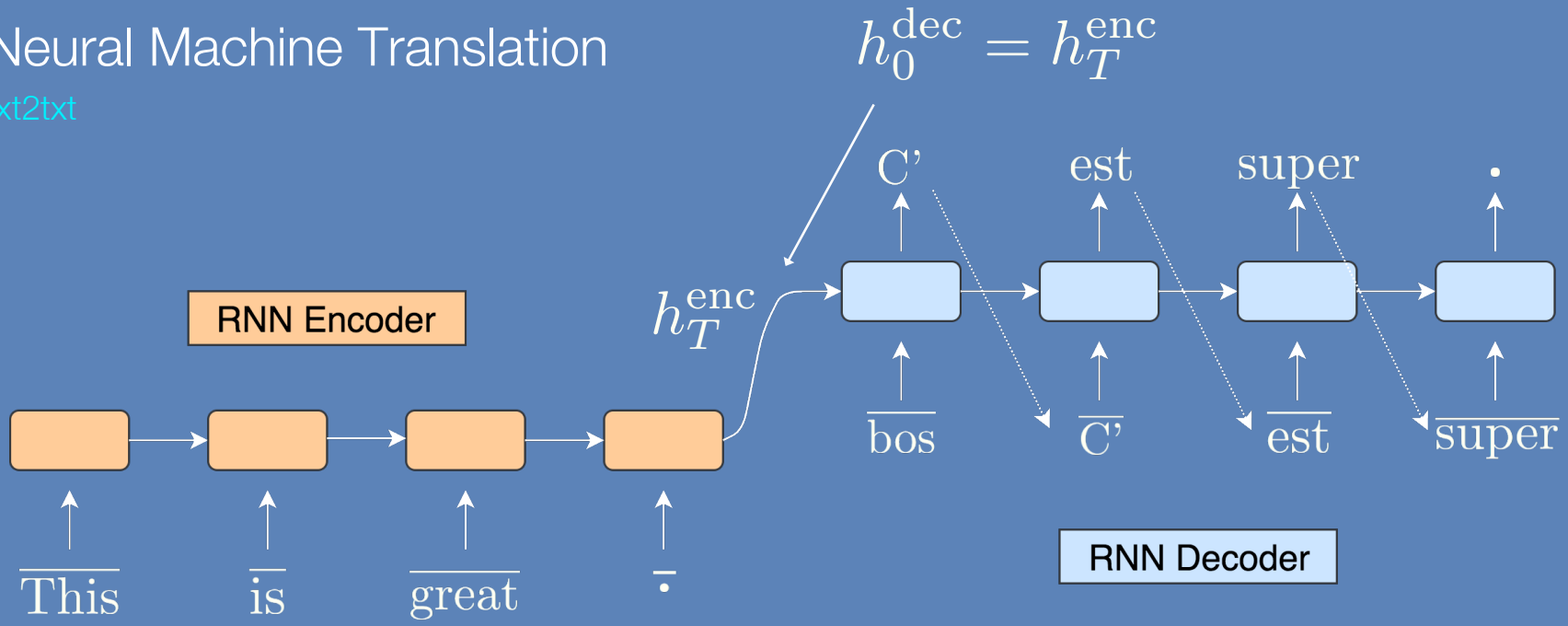
RNN Decoder

can generate a sentence word by word when initialized with a vector h0

Can we send the information of the encoder to the decoder?

Neural Machine Translation

txt2txt

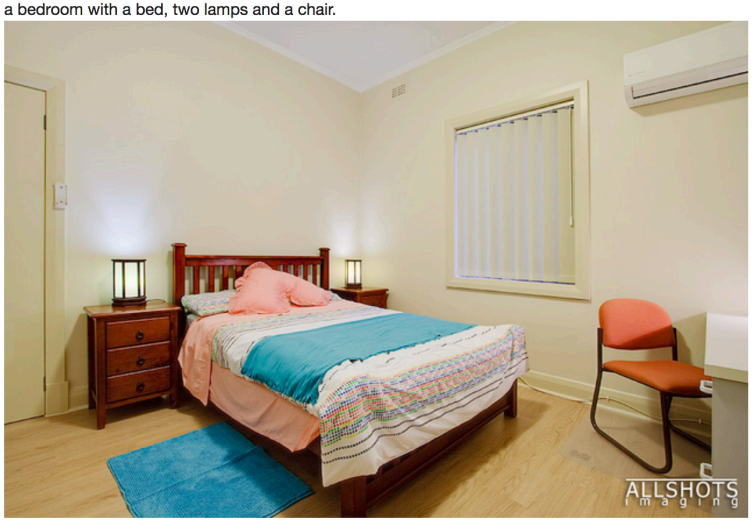


Solution: Initialize first hidden state of decoder using last hidden state of encoder

Image captioning

COCO dataset

a small bedroom with a made bed and chair
a neatly made bed with a blue rug and chair.
the interior set up of a modern bedroom.
a bed is shown in a simple bedroom with minimal furniture.
a bedroom with a bed, two lamps and a chair.



a large plane is flying in the sky
a large military aircraft at low altitude flight.
dark grey military plane flies off in overcast sky.
an air force jet is taking off into the sky.
a military plane is flying through the air.

Image captioning

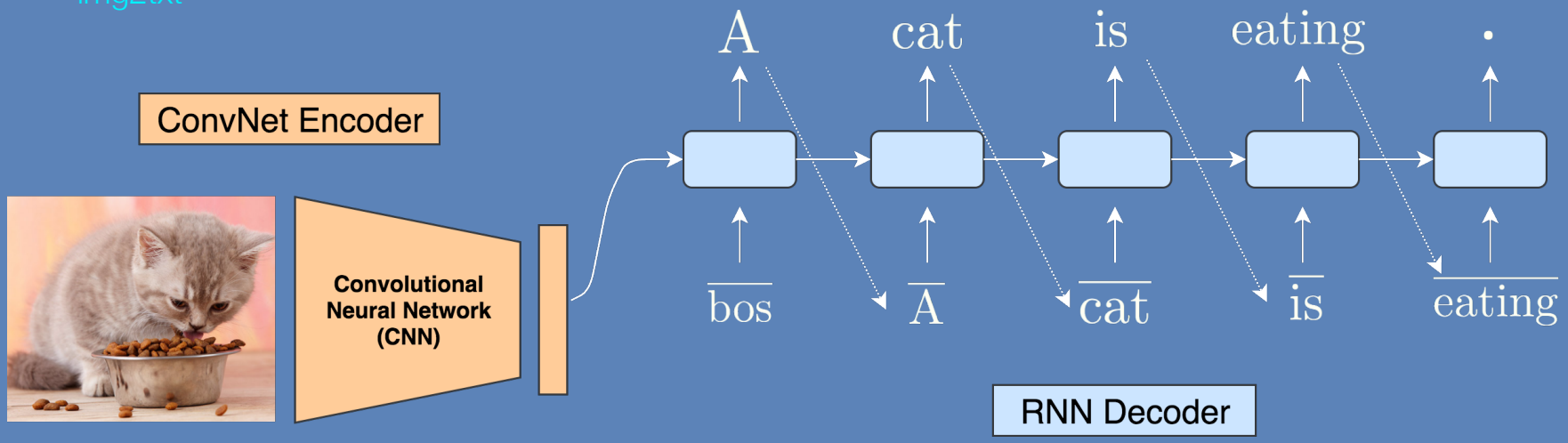
COCO dataset

a man with a tennis racket and the ball in front of him.
a tennis player getting ready to swing at the ball.
a tennis player in action on the court.
a man holding a tennis racquet on a tennis court.
a man raises his tennis racquet as a tennis ball comes towards him.

an elephant with a group of people riding on it's back.
people sit under umbrellas atop a giant wooden elephant.
there is a large decorated elephant with people sitting on it
some people are on a large wooden float shaped like an elephant
a very large wooden elephant with people on top.

Image captioning

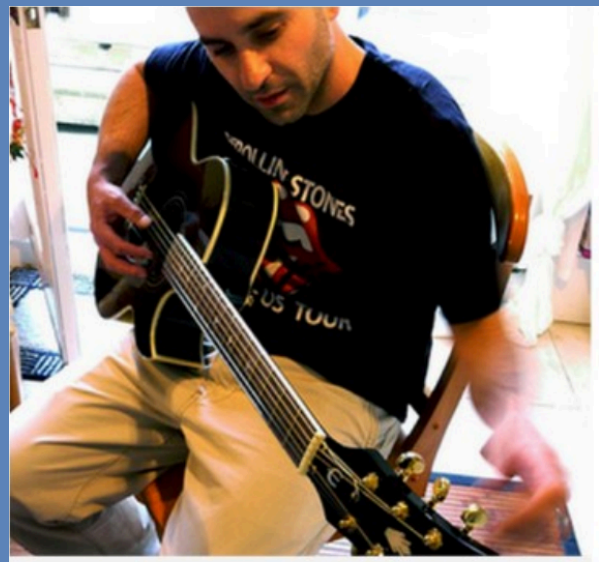
img2txt



Solution: Initialize first hidden state of decoder using output of the ConvNet encoder

Image captioning

results



man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

Image captioning

results

two young girls are playing with lego toy.

boy is doing backflip on wakeboard.

Image captioning

limitations

Image captioning

Limitations

- Model limited to its training data
- Need a lot of supervision (expensive) to generalize to all images …

Attention mechanism

Getting closer to NMT state-of-the-art systems (FB, Google, Linguee …)

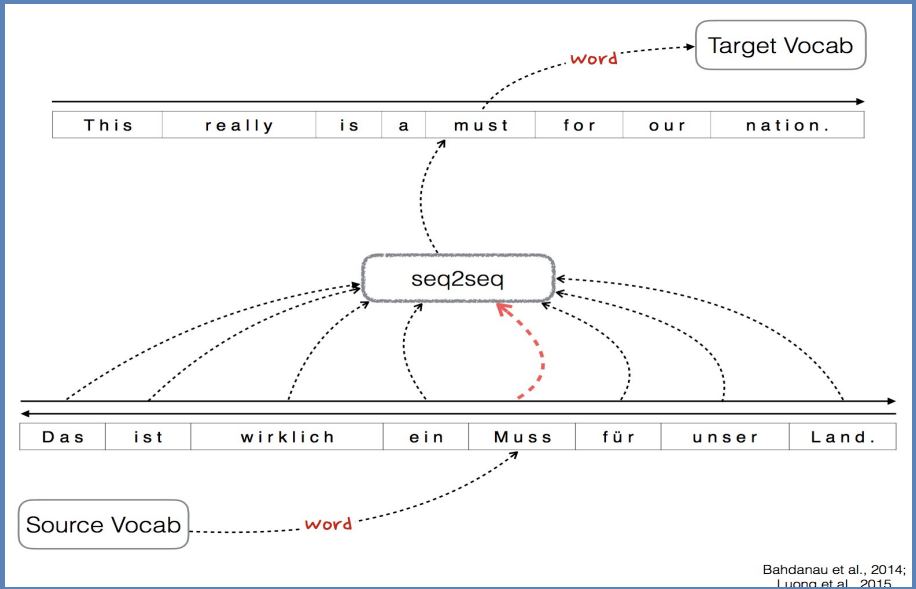
- Question 1: Is a fixed-size vector desirable to encode the information of an object?
  Answer: Instead, encode the input as a set of vectors (variable-size)

- Question 2: How to dynamically select the information relevant to your task, in this set?
  Answer: **Attention** mechanism

Attention mechanism

Bahdanau et al.

- Attention mechanism: **At each time step** prediction, we give the decoder the ability to **focus** its **attention on the source words that are relevant to make its prediction**



* Bahdanau et al. (ICLR 2015) – Neural Machine Translation by Jointly Learning to Align and Translate

Attention mechanism

Attention vector

- Without attention:

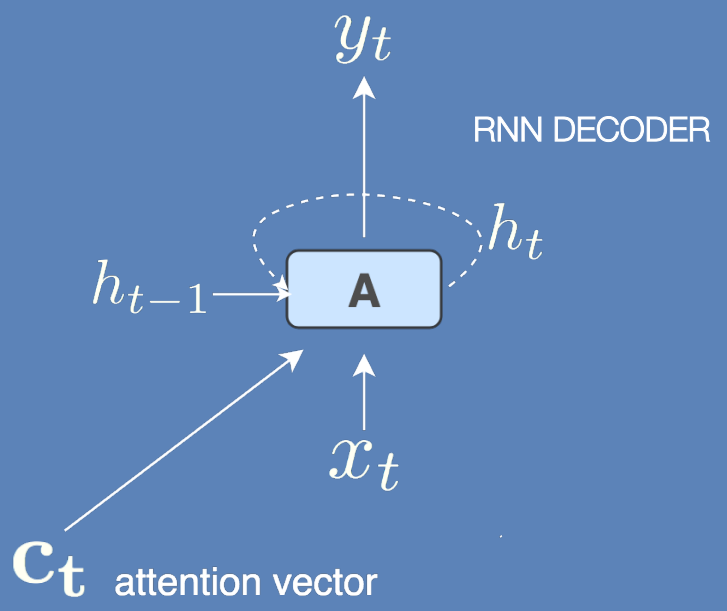
$$\tanh(\mathbf{W_x x_t} + \mathbf{W_h h_{t-1}} + \mathbf{c})$$

Last hidden state of encoder

- With attention:

$$\tanh(\mathbf{W_x x_t} + \mathbf{W_h h_{t-1}} + \mathbf{W_c c_t})$$

Changes at every time step



* Bahdanau et al. (ICLR 2015) – Neural Machine Translation by Jointly Learning to Align and Translate

Attention mechanism

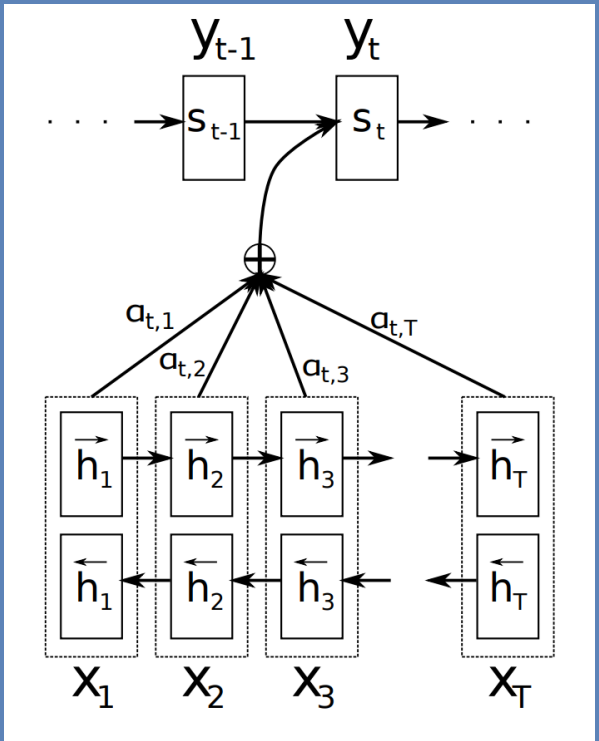
Attention vector

$$\mathbf{c_i} = \sum_{\mathbf{j=1}}^{\mathbf{T_s}} \mathbf{a_{ij}} \mathbf{h_j^{enc}}$$ context vector at t = i

$$\mathbf{e_{ij}} = \mathbf{a}(\mathbf{s_{i-1}}, \mathbf{h_j^{enc}})$$ a: feedforward network

$$\alpha_{\mathbf{ij}} = \frac{\exp(\mathbf{e_{ij}})}{\sum_{\mathbf{k=1}}^{\mathbf{T_x}} \exp(\mathbf{e_{ik}})}$$ importance of $\mathbf{w_j^{source}}$ to produce $\hat{\mathbf{w}}_\mathbf{i}^{\mathbf{target}}$

All of this is trained end-to-end with BP and SGD

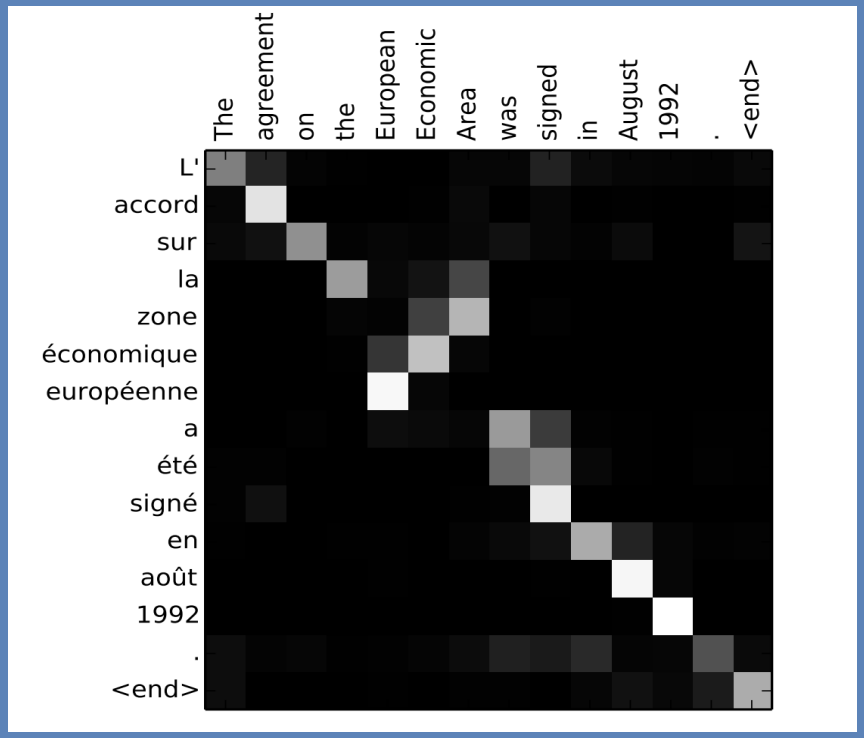


* Bahdanau et al. (ICLR 2015) – Neural Machine Translation by Jointly Learning to Align and Translate

Attention mechanism

Visualization of the alignment matrix

French to English

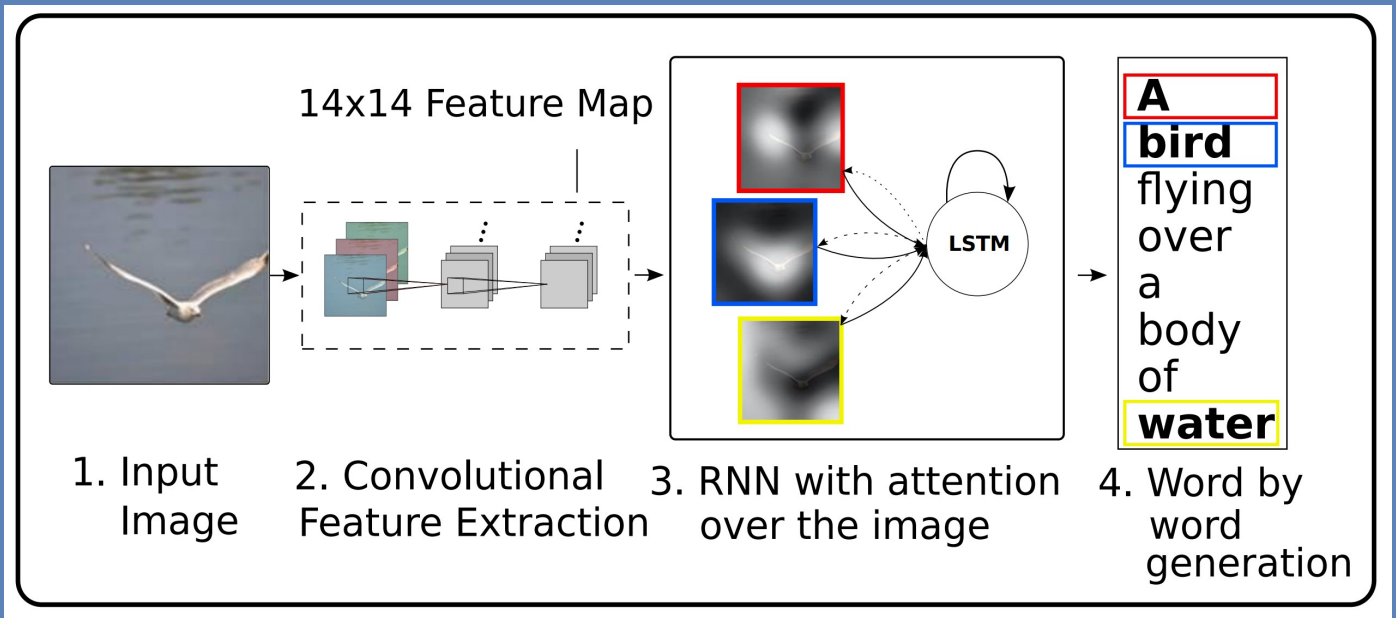


Visualization of the $\mathbf{a}_{(\mathbf{i},\mathbf{j})}$

Attention mechanism

Image captioning

Image-captioning with attention on the feature maps of the convolutional image encoder



* Xu et al. (NIPS 2015) – Show Attend and Tell: Neural Image Caption Generation with Visual Attention

Attention mechanism

Image captioning

Image attention visualization

Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. "soft" (top row) vs "hard" (bottom row) attention. (Note that both models generated the same captions in this example.)

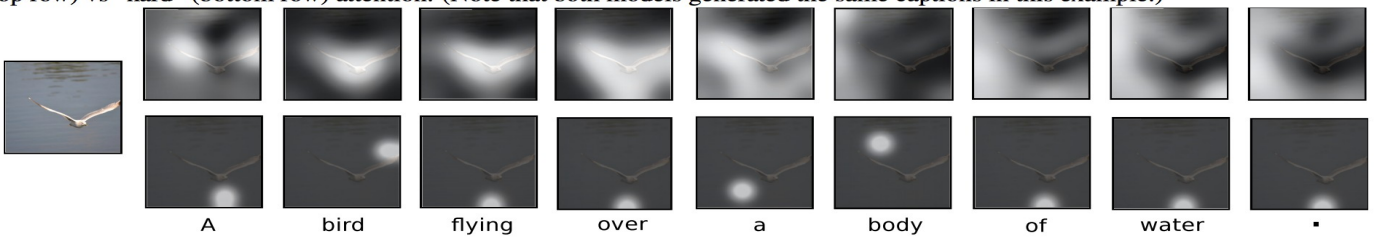


Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

* Xu et al. (NIPS 2015) – Show Attend and Tell: Neural Image Caption Generation with Visual Attention

Subwords units in NMT

Byte Pair Encoding (BPE)

Subword units

- Softmax in decoder (for creating the probability of the next word) is expensive

- Instead of considering V=100k, 150k … words, we limit ourselves to ~40k **subword units**

Learn BPE on your corpus

- Start with a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z
- Count the number of times two consecutive tokens appear in the text
- Add the most frequent pair to your list of "merge operations"
- Repeat, and keep the (say) 40k most frequent

Apply BPE on your (or any) corpus

- For each word, split it into subword units using the "merge operations" learned above

* Sennrich et al (ACL 2016) – Neural Machine Translation of Rare Words with Subword Units

Subwords units in NMT

Byte Pair Encoding (BPE)

https://github.com/rsennrich/subword-nmt

Tireraient profit	Tir@@ eraient profit
balayons devant notre porte	bal@@ ayons devant notre porte
Aufwandsentschädigung	Auf@@ wand@@ s@@ entschädigung
Berufungsmöglichkeiten	Berufungs@@ möglichkeiten

* Sennrich et al (ACL 2016) – Neural Machine Translation of Rare Words with Subword Units

Sequence to sequence tasks

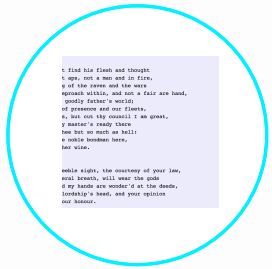
Examples of seq2seq tasks

task	source	target
AutoEncoder	I myself was out on an island in the Swedish archipelago , at Sandhamn .	I myself was out on an island in the Swedish archipelago , at Sand@ ham@ n .
NMT En-Fr	I myself was out on an island in the Swedish archipelago , at Sandhamn .	Je me trouvais ce jour là sur une île de l' archipel sué-dois , à Sand@ ham@ n .
NMT En-De	We really need to up our particular con-tribution in that regard .	Wir müssen wirklich unsere spezielle Hilfs@ leistung in dieser Hinsicht aufstocken .
NMT En-Fi	It is too early to see one system as a uni-versal panacea and dismiss another .	Nyt on liian aikaista nostaa yksi järjestelmä jal@ usta@ lle ja antaa jollekin toiselle huono arvo@ sana .
* SkipThought	the old sami was gone , and he was a different person now .	the new sami didn 't mind standing barefoot in dirty white , sans ra@ y-@ bans and without beautiful women following his every move .
** Seq2Tree	Dikoya is a village in Sri Lanka .	(ROOT (S (NP NNP)NP (VP VBZ (NP (NP DT NN)NP (PP IN (NP NNP NNP)NP)PP)NP)VP .)S)ROOT

BPE splits

* Kiros et al. (NIPS 2015) – Skip-Thought vectors

** Vinyals et al. (NIPS 2016) – Grammar as a Foreign Language



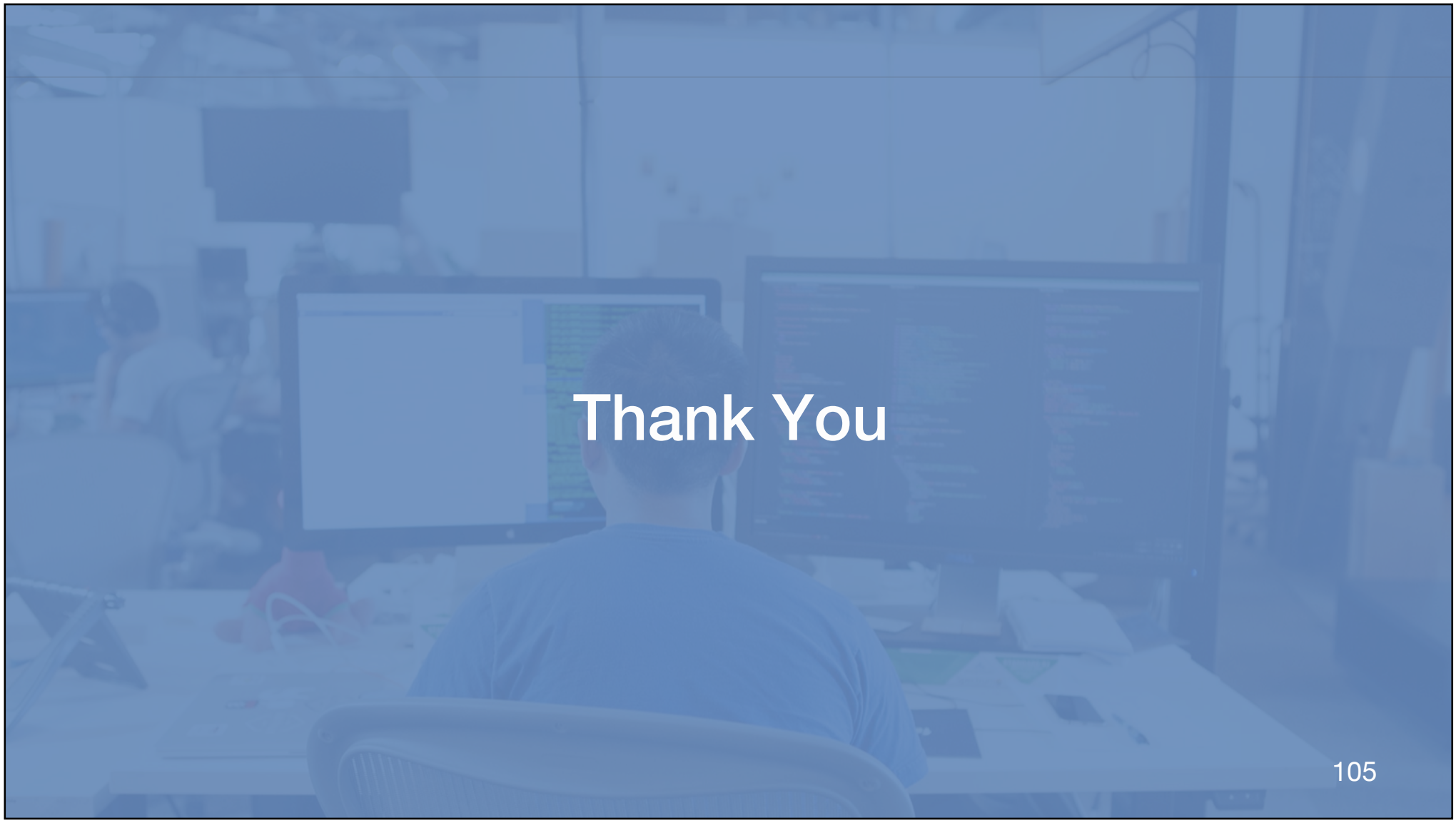
Conclusion

- What we have seen
- What we haven’t seen
- Going further

What we have seen

- RNNs can handle word order and context: better than bag-of-words
- RNNs can be used for text generation and language modelling
- Encoder-decoder: AutoEncoder/NMT/Seq2Tree/Seq2Img
- More advanced: Attention mechanism and BPE

Tools for Data Science


Thank You