Module 2 :

Deep Networks

Recurrent Neural Networks

Géraldine Conti, August 2020

Discussion Session

- Review of Notebooks 6.1 and 6.2:
    - Computer vision (6.1) :
        - Convolution, padding, pooling layer
        - Using pretrained model (Resnet-50)
    - Data augmentation (6.2) :
        - Flipping, grayscale, saturation, brightness, rotation, cropping
        - Augment dataset and train with it

Bibliography

- Deep Learning book (Goodfellow, Bengio, Courville)
- Machine Learning @ Stanford (Prof Andrew Ng)
- Hands-On Machine Learning with Scikit-Learn & Tensorflow (Aurélien Géron)

Learning Objectives

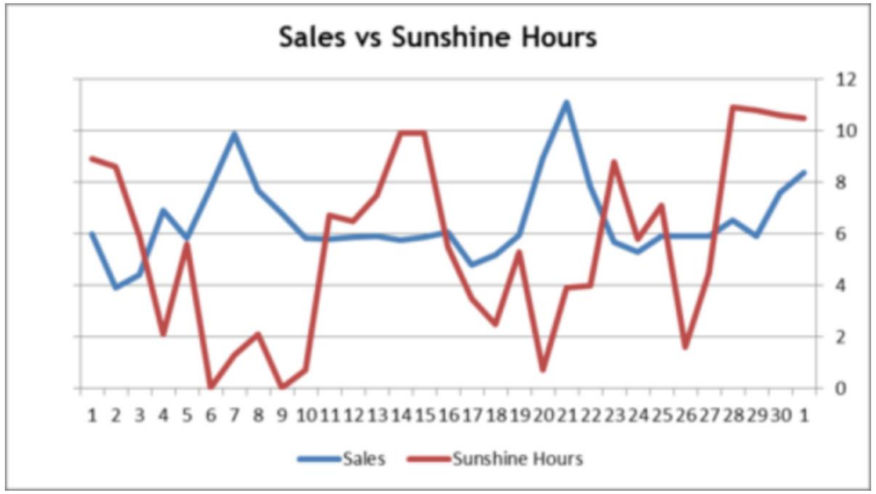
1. Recurrent Neural Networks components
2. Training RNNs
3. Optimization techniques
4. Examples
5. Natural Language Processing (NLP)

1. RNN components

- Recurrent Neurons
- Memory Cells
- Input/output sequences
- Examples

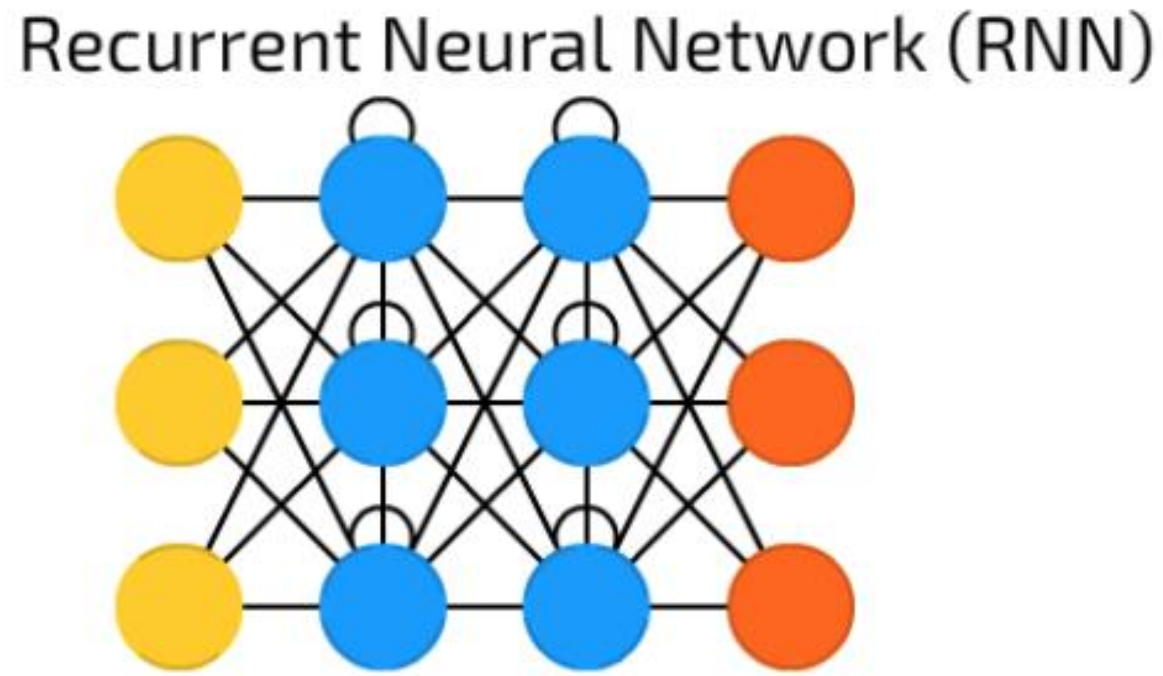
Introduction

- Class of nets that can predict the future
- Work on sequences or arbitrary lengths
- Inputs : sentences, documents, audio,…
- Use cases :
  - Can analyze time series data
  - Can anticipate car trajectories and help avoid accidents
  - Text analysis (sequences where context is important)



RNN

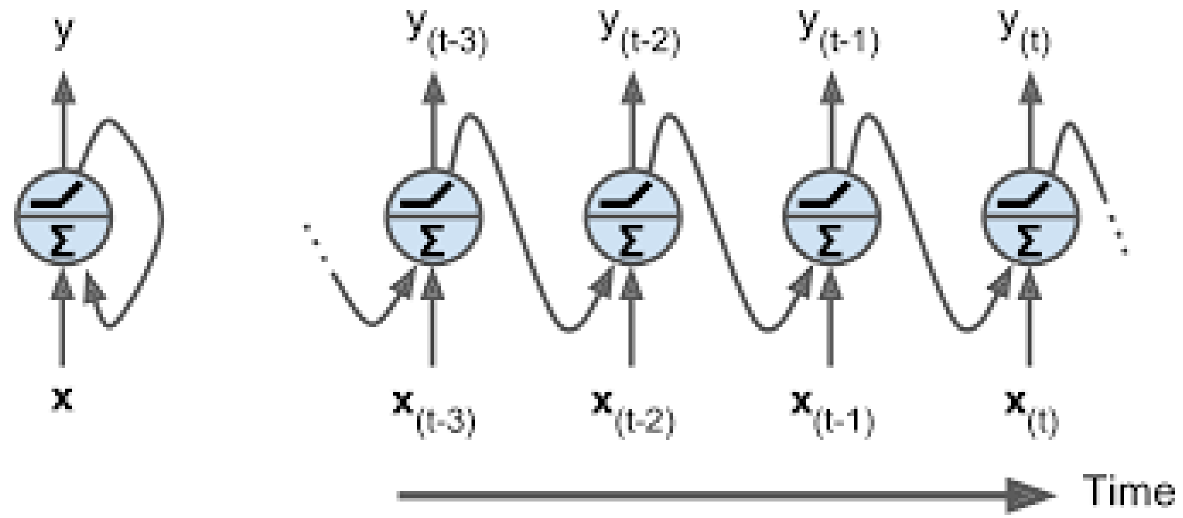
- connections between neurons include loops
- **Recurrent cells** (or memory cells) used
  - Weight sharing between *time-steps*



Recurrent Neurons

- At each time step t, the recurrent neuron receives the inputs $x_{(t)}$ as well as its own output from the previous time steps $y_{(t-1)}$

- Unrolling the network through time

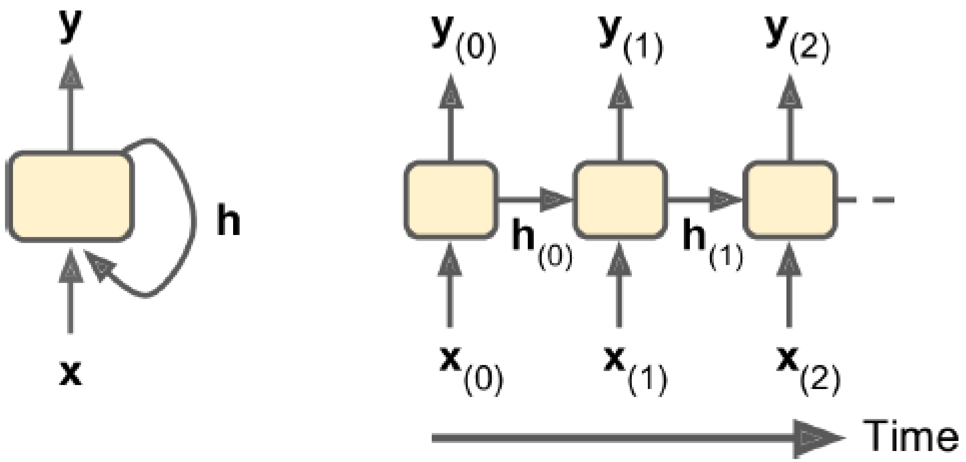


- Output at time step t is a function of previous states and current input → *memory*

$$y_t = \varphi(x_t^T \cdot w_x + y_{t-1}^T \cdot w_y + b)$$

  - Two sets of weights w_x and w_y

Memory Cells



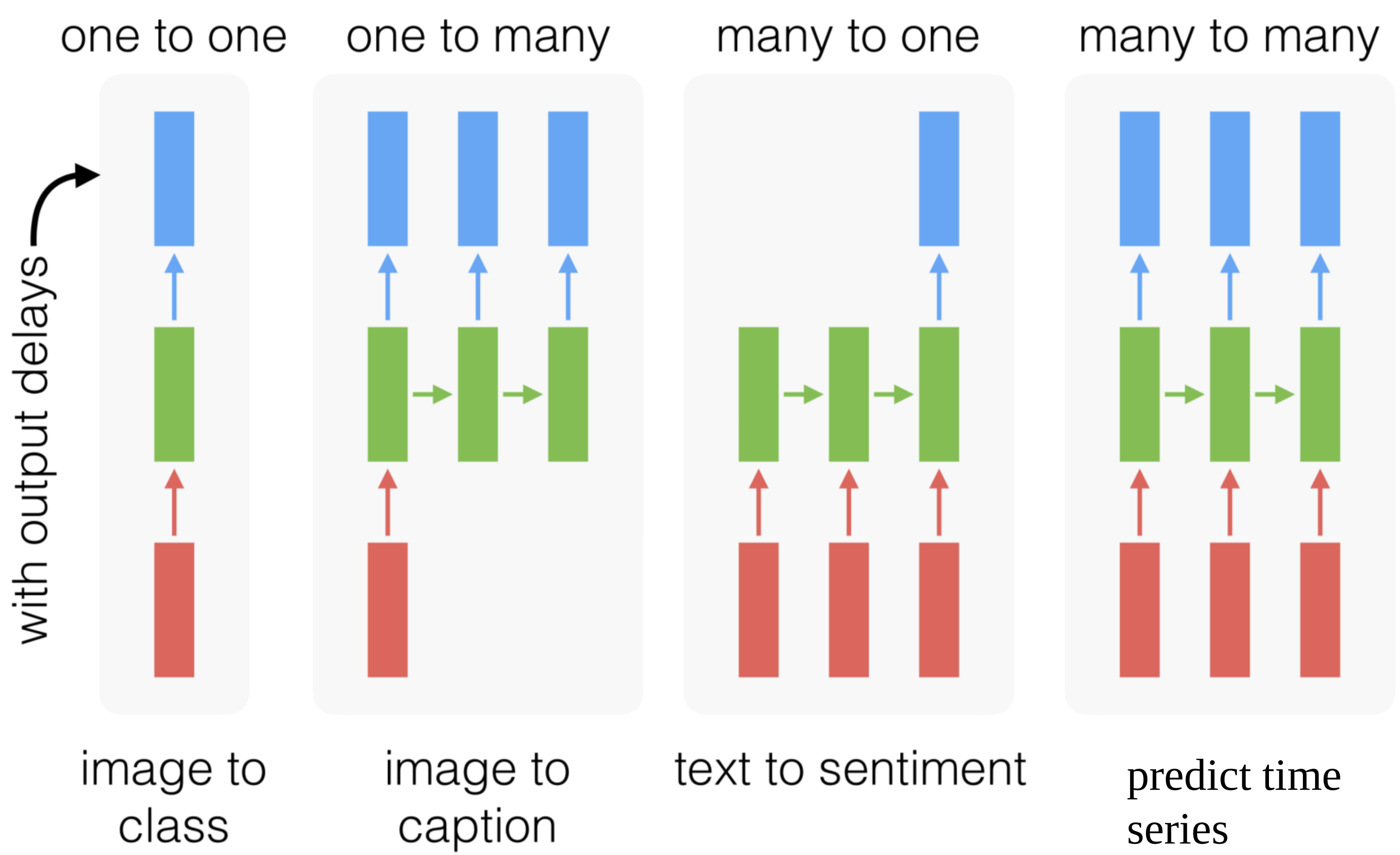
- Memory cell : part of a NN that preserves some states across time steps

- Cell state at time step t is a function of some inputs at that time steps and its state at the previous time step:

$$h_t = f(h_{t-1}, x_t)$$

Not necessarily equal to the output y_t

Input and Output Sequences

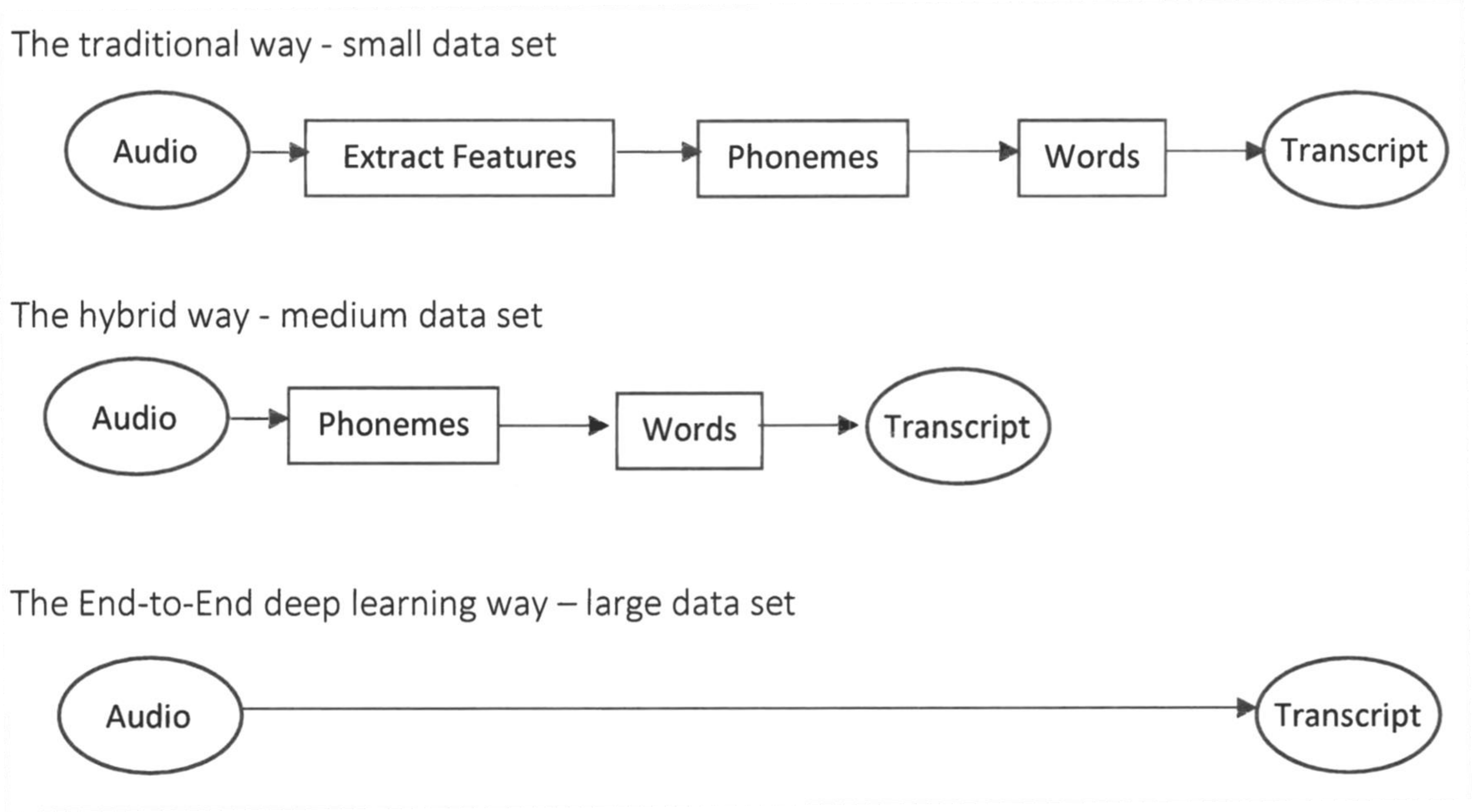


What would you use for language translation ?

Examples

<table>
<tr><th colspan="2">Input</th><th colspan="2">Target</th><th rowspan="2">Use Cases</th></tr>
<tr><th>Type</th><th>Elements</th><th>Type</th><th>Elements</th></tr>
<tr><td rowspan="4">Scalar</td><td rowspan="4">One</td><td>Trends</td><td>Many</td><td>Pattern generation</td></tr>
<tr><td>Audio</td><td>Many</td><td>Music Generation</td></tr>
<tr><td>Text</td><td>Many</td><td>Text Generation</td></tr>
<tr><td>Image</td><td>Many</td><td>Image generation</td></tr>
<tr><td rowspan="4">Trends</td><td rowspan="4">Many</td><td rowspan="2">Scalar</td><td rowspan="2">One</td><td>Stock Trading decisions</td></tr>
<tr><td>Forecasting KPI for fixed duration</td></tr>
<tr><td rowspan="2">Trends</td><td rowspan="2">Many</td><td>DNA Sequence analysis</td></tr>
<tr><td>Time series forecasts</td></tr>
<tr><td rowspan="13">Text</td><td rowspan="13">Many</td><td rowspan="3">Scalar</td><td rowspan="3">One</td><td>Sentiment Classification</td></tr>
<tr><td>Topic Classification</td></tr>
<tr><td>Answer Selection</td></tr>
<tr><td rowspan="8">Text</td><td rowspan="8">Many</td><td>Text Summarization</td></tr>
<tr><td>Machine translation</td></tr>
<tr><td>Chatbots</td></tr>
<tr><td>Name Entity Recognition</td></tr>
<tr><td>Subject Extraction</td></tr>
<tr><td>Part of Speech Tagging</td></tr>
<tr><td>Textual Entailment</td></tr>
<tr><td>Relation Classification</td></tr>
<tr><td>Trends</td><td>Many</td><td>Path Query Answering</td></tr>
<tr><td>Audio</td><td>Many</td><td>Speech Generation</td></tr>
<tr><td rowspan="4">Image</td><td rowspan="4">Many</td><td rowspan="2">Scalar</td><td rowspan="2">One</td><td>Facial expression tagging</td></tr>
<tr><td>Entity classification</td></tr>
<tr><td>Text</td><td>Many</td><td>Image Captioning</td></tr>
<tr><td>Image</td><td>Many</td><td>Image Modification</td></tr>
<tr><td rowspan="6">Audio</td><td rowspan="6">Many</td><td rowspan="3">Scalar</td><td rowspan="3">One</td><td>Sentiment Classification</td></tr>
<tr><td>Number of speaker tagging</td></tr>
<tr><td>Topic Classification</td></tr>
<tr><td rowspan="2">Text</td><td rowspan="2">Many</td><td>Speech Recognition</td></tr>
<tr><td>Conference Summarization</td></tr>
<tr><td>Audio</td><td>Many</td><td>Speech Assistant</td></tr>
<tr><td rowspan="2">Video</td><td rowspan="2">Many</td><td>Scalar</td><td>One</td><td>Activity Recognition</td></tr>
<tr><td>Text</td><td>Many</td><td>Subtitle generation</td></tr>
</table>

Speech Recognition Example



- End-to-End deep learning : simplification of a learning system into one NN
  - *Large dataset of labeled data is required*

Scalar → X

Input		Target		Use Cases
Type	Elements	Type	Elements	
Scalar	One	Trends	Many	Pattern generation
		Audio	Many	Music Generation
		Text	Many	Text Generation
		Image	Many	Image generation

Music Generator

Random seed = 5 → Sequence Model →

Text Captioning

Random seed = 10 → Sequence Model →

Neuron is a cell that can transmit electrical signals. They can also refer to units in ANN model................

Image Generator

Random seed = 15 → Sequence Model →

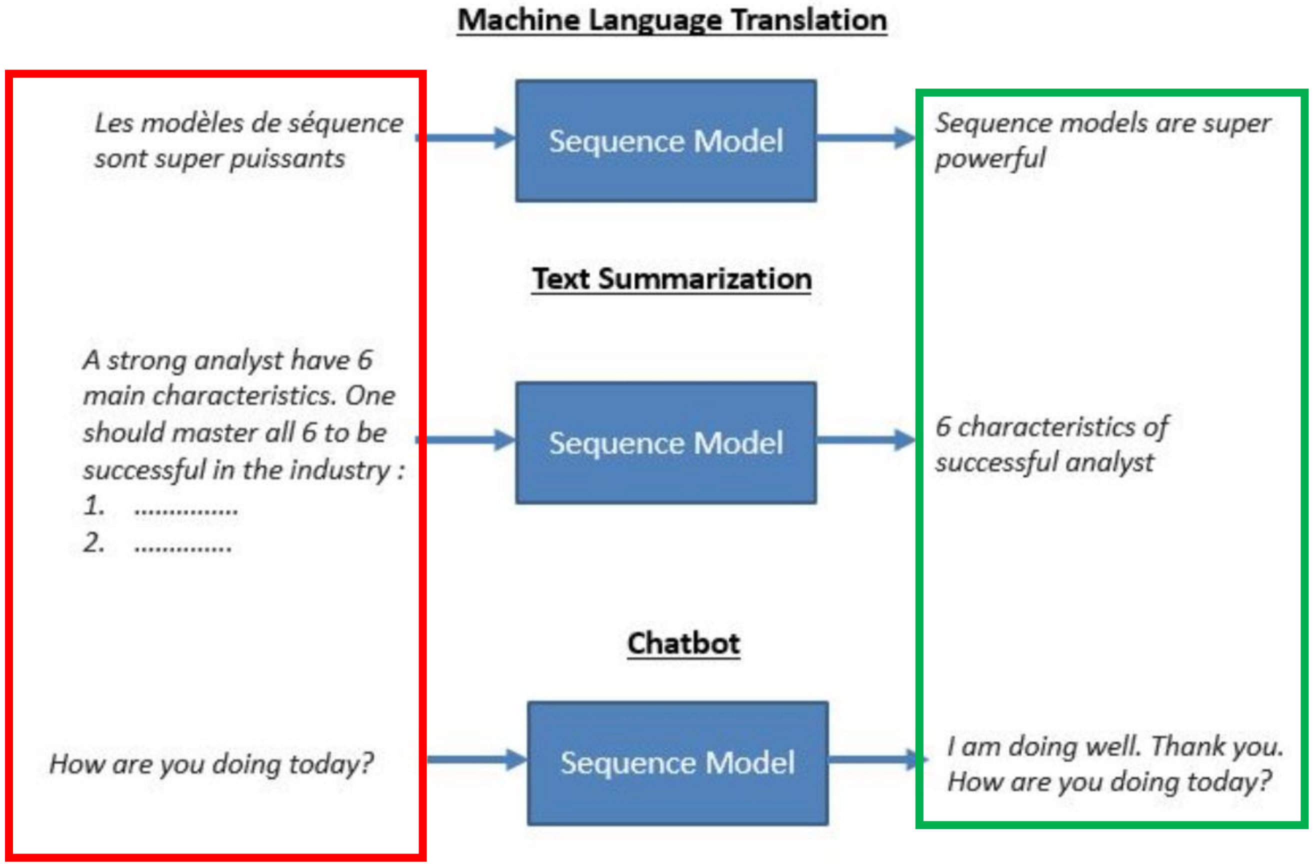
Text → Scalar

Input Type	Input Elements	Target Type	Target Elements	Use Cases
Text	Many	Scalar	One	Sentiment Classification
				Topic Classification
				Answer Selection
		Text	Many	Text Summarization
				Machine translation
				Chatbots
				Name Entity Recognition
				Subject Extraction
				Part of Speech Tagging
				Textual Entailment
				Relation Classification
		Trends	Many	Path Query Answering
		Audio	Many	Speech Generation

Sentiment	Tweets
Negative	@united is the worst. Nonrefundable First class tickets? Oh because when you select Global/FC their system auto selects economy w/upgrade.
	@united I will not be flying you again
Neutral	@VirginAmerica my drivers license is expired by a little over a month. Can I fly Friday morning using my expired license?
	@VirginAmerica any plans to start flying direct from DAL to LAS?
Positive	@VirginAmerica done! Thank you for the quick response, apparently faster than sitting on hold ;)
	@united I appreciate your efforts getting me home!

Text → Text

Input		Target		Use Cases
Type	Elements	Type	Elements	
Text	Many	Scalar	One	Sentiment Classification
				Topic Classification
				Answer Selection
		Text	Many	Text Summarization
				Machine translation
				Chatbots
				Name Entity Recognition
				Subject Extraction
				Part of Speech Tagging
				Textual Entailment
				Relation Classification
		Trends	Many	Path Query Answering
		Audio	Many	Speech Generation



Input		Target		Use Cases
Type	Elements	Type	Elements	
Image	Many	Scalar	One	Facial expression tagging
				Entity classification
		Text	Many	Image Captioning
		Image	Many	Image Modification
Audio	Many	Scalar	One	Sentiment Classification
				Number of speaker tagging
				Topic Classification
		Text	Many	Speech Recognition
				Conference Summarization
		Audio	Many	Speech Assistant
Video	Many	Scalar	One	Activity Recognition
		Text	Many	Subtitle generation

X → Text

Speech Recognition

Sequence Model → *I love oranges*

Image Captioning

Sequence Model → *Two dogs are playing with a ball*

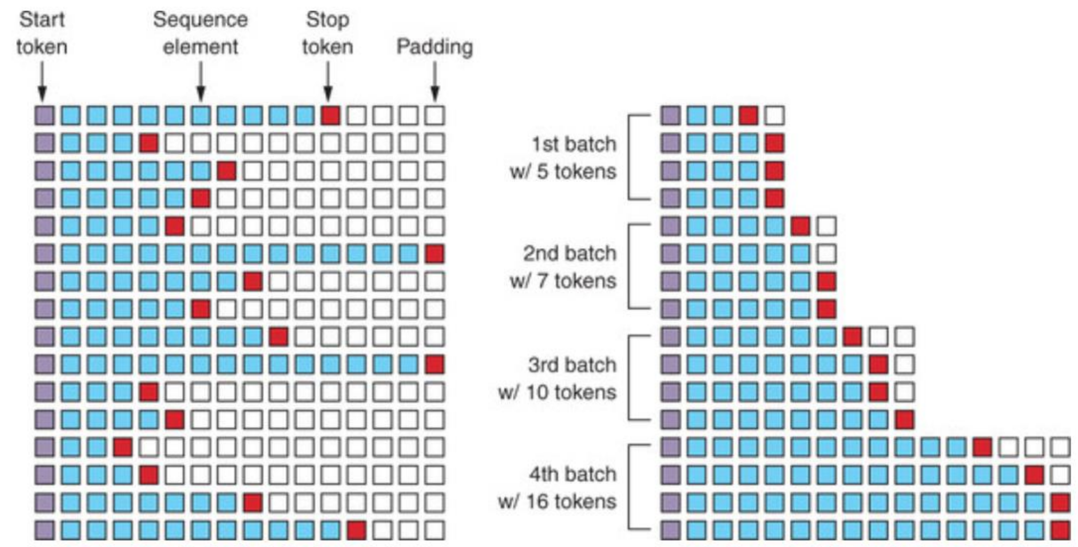
Subtitle Generator

Sequence Model → *How you doin?*

2. Training RNNs

- Input data :
  - Bucketing
  - Time series
- Training loop
- Bidirectional RNNs
- Deep RNNs
- LSTM/GRU Cells

Input Data : Bucketing



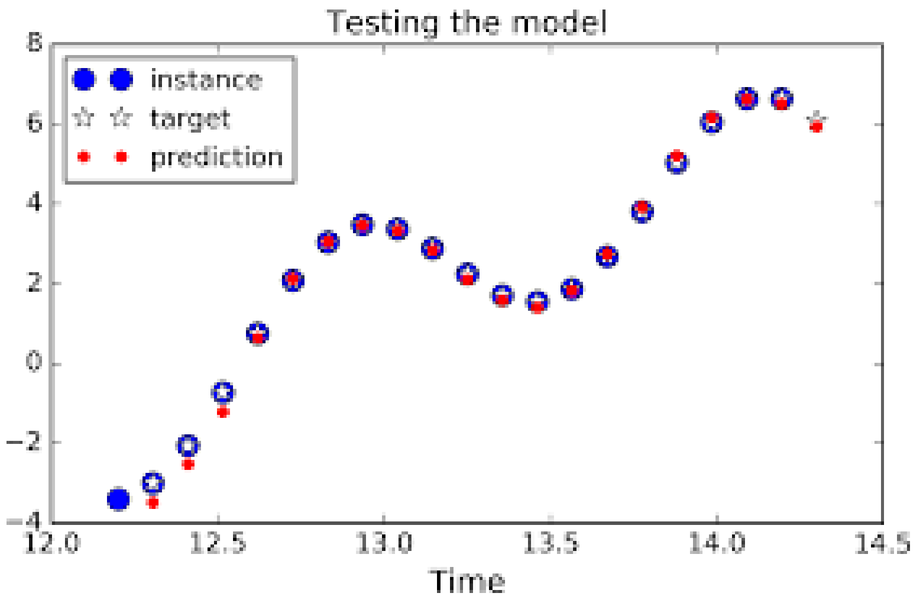
- Used in the case of variable-length sequences
  - Allows to *reduce the number of time steps* needed for a particular batch

- Recipe :
  - Sort the sequences by length
  - Group them in buckets
  - Pad them (add zeros) to the maximum token length for a particular bucket

Input Data : Time Series

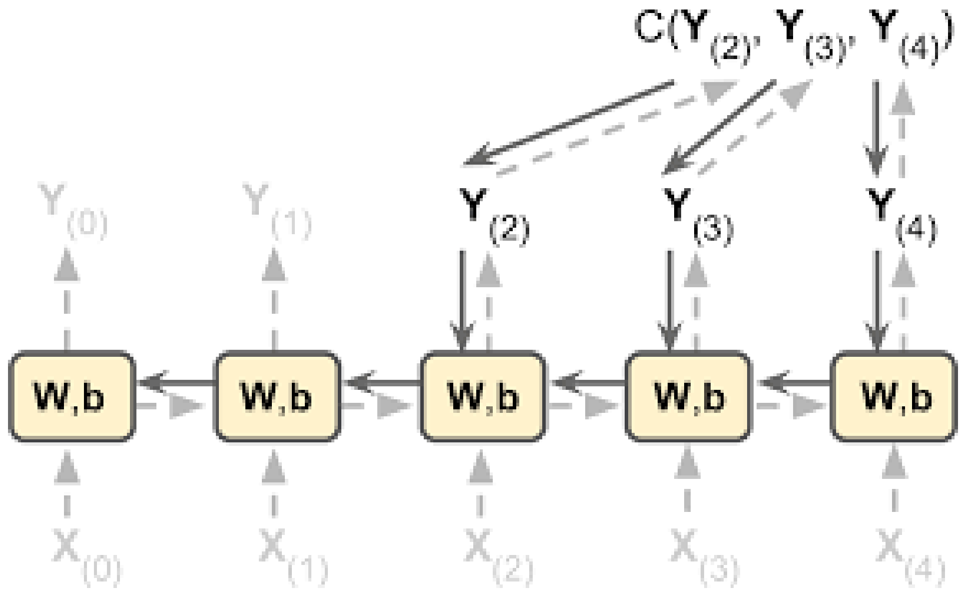
- Training instance : randomly selected sequence of 20 consecutive values from the time series
  - In principle, have several input features

- Target sequence : similar as the input sequence, except it is shifted by one time step into the future



Training Loop

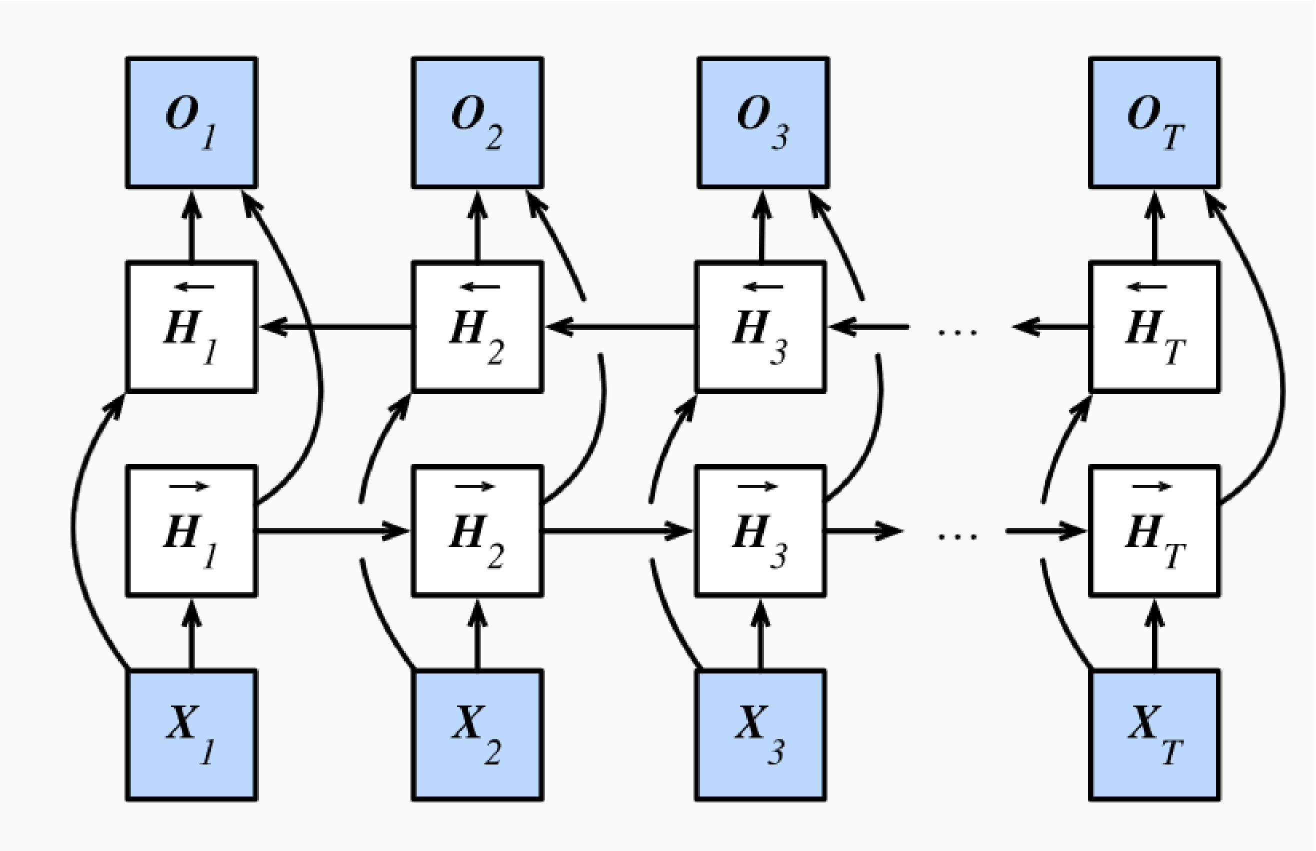
- Trick is to unroll it through time and use regular backpropagation (backpropagation through time)
  - First forward pass through the unrolled network
  - output sequence evaluated using a cost function
  - Gradients of the cost function are propagated backward through the unrolled network
  - Model parameters are updated using computed gradients



Bidirectional RNN

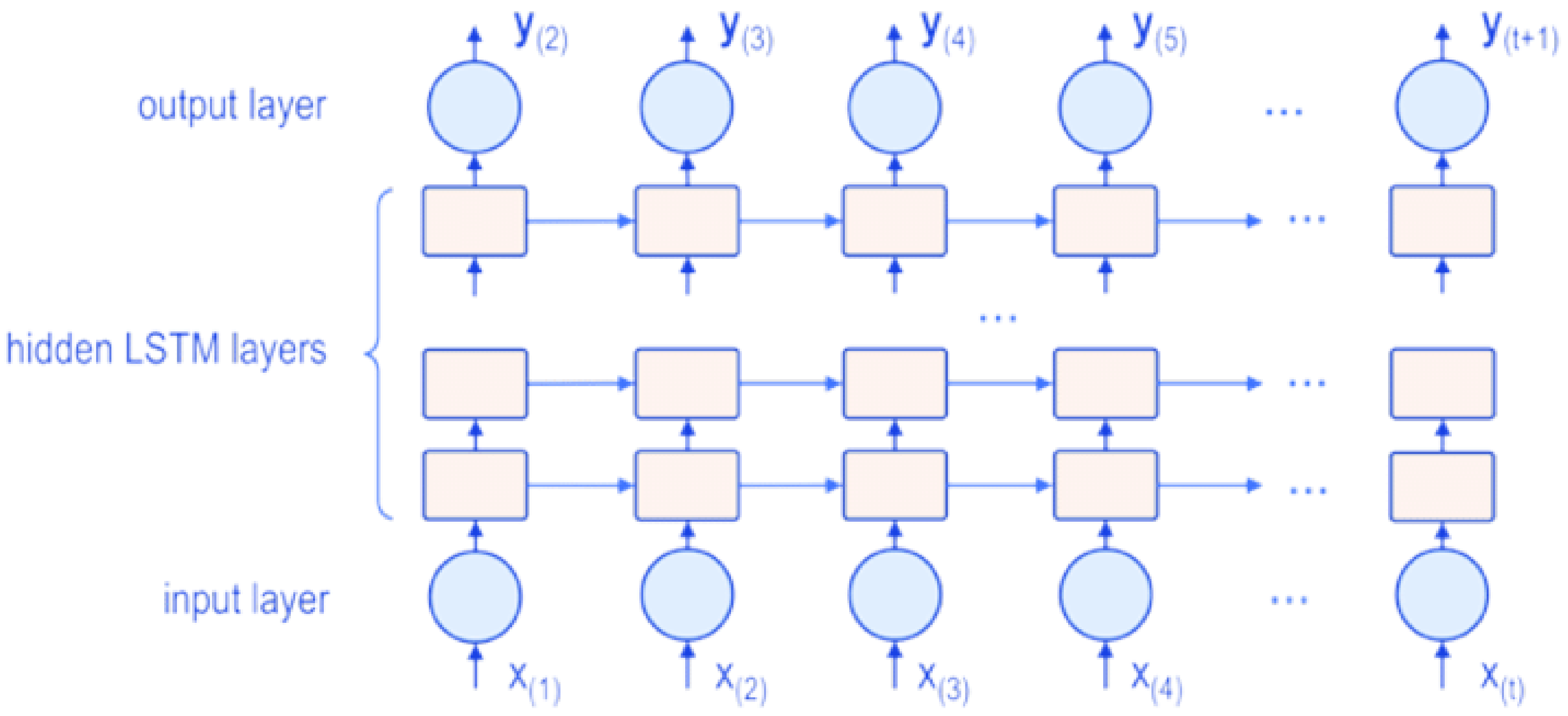
- A RNN limitation is that it only uses information from earlier in the sequence and not after

- Solution : bidirectional RNN



Deep RNNs

- Stack multiple layers of cells



- Apply dropout to reduce overfitting

Training over many time steps

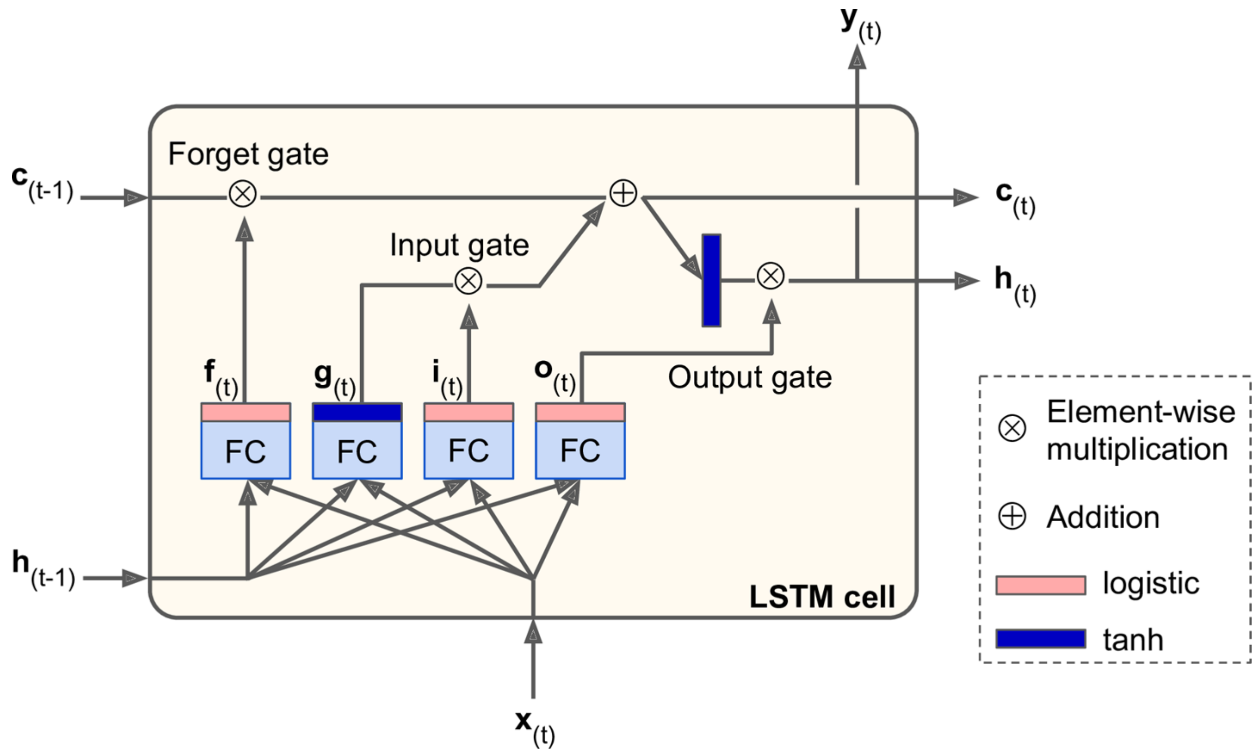
- Many time steps needed to train a RNN on long sequences
- Vanishing/exploding gradients

What tricks can you use ?

- One solution : truncated backpropagation through time : unroll the RNN only over a limited number of time steps during training
- Limits :
  - Missing part of crucial data in your training sample (specific events/dates,…)
  - Memory of the first inputs gradually fades away

LSTM Cell

- Long Short-Term Memory (LSTM) cell (1997)
  - Converges faster and detects long-term dependencies in the data

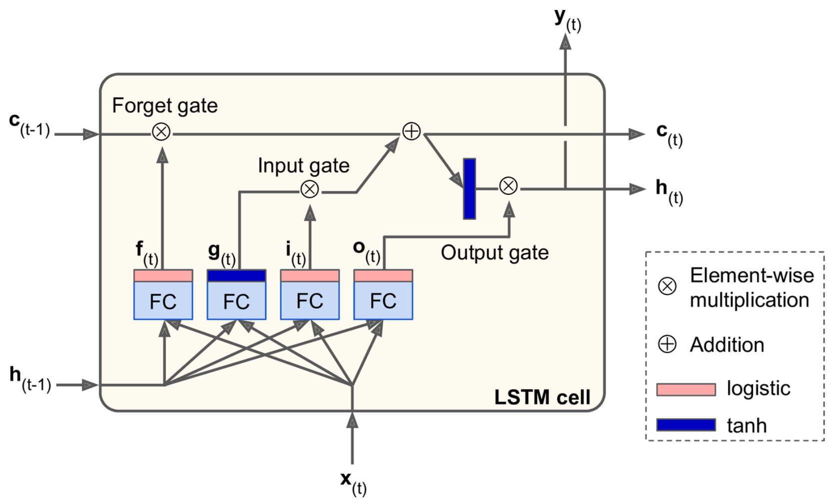


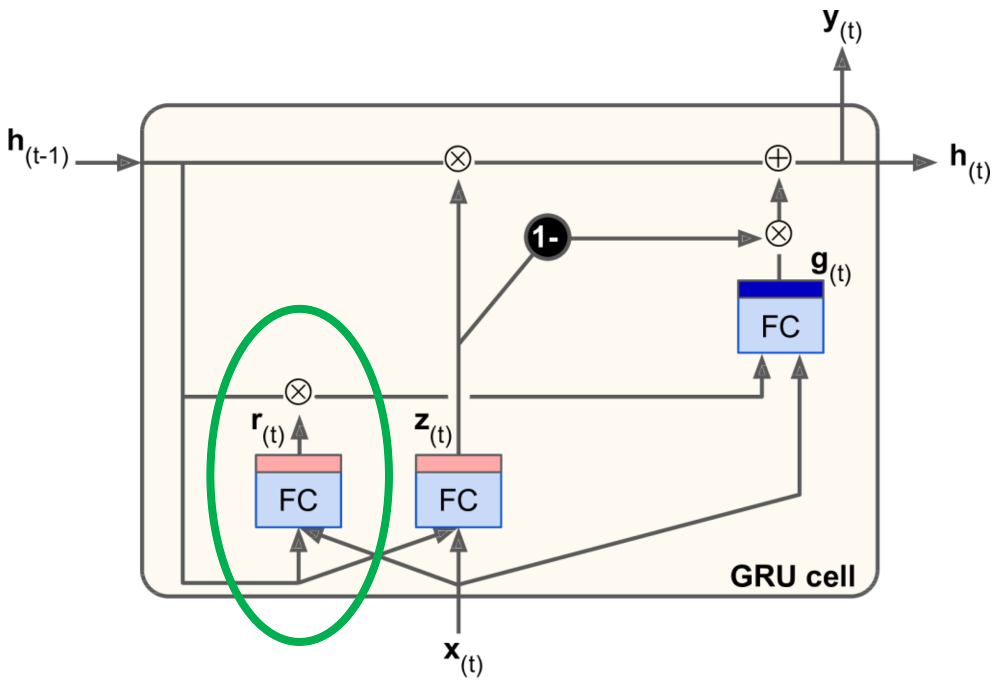
- Cell state is split in two vectors h_t (short-term state) and c_t (long-term state)

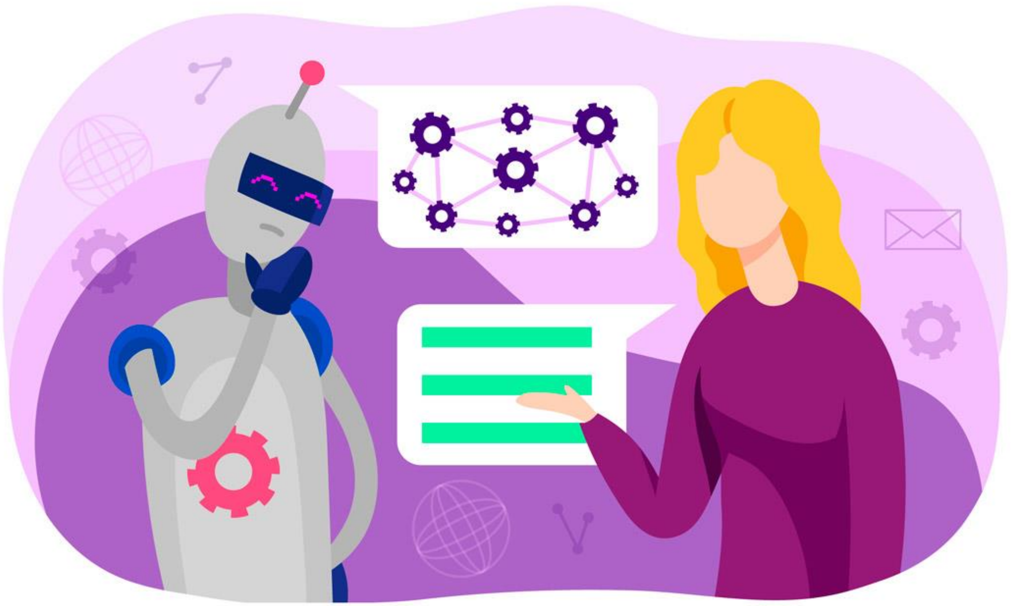
- Network that can learn what to store in the long-term state, what to throw away, and what to read from it

GRU Cell

- Gated Recurrent Unit (GRU) cell is a simplified version of the LSTM cell
- Both state vectors are merged into a single vector h_t
- A single gate controller controls both the forget gate and the input gate







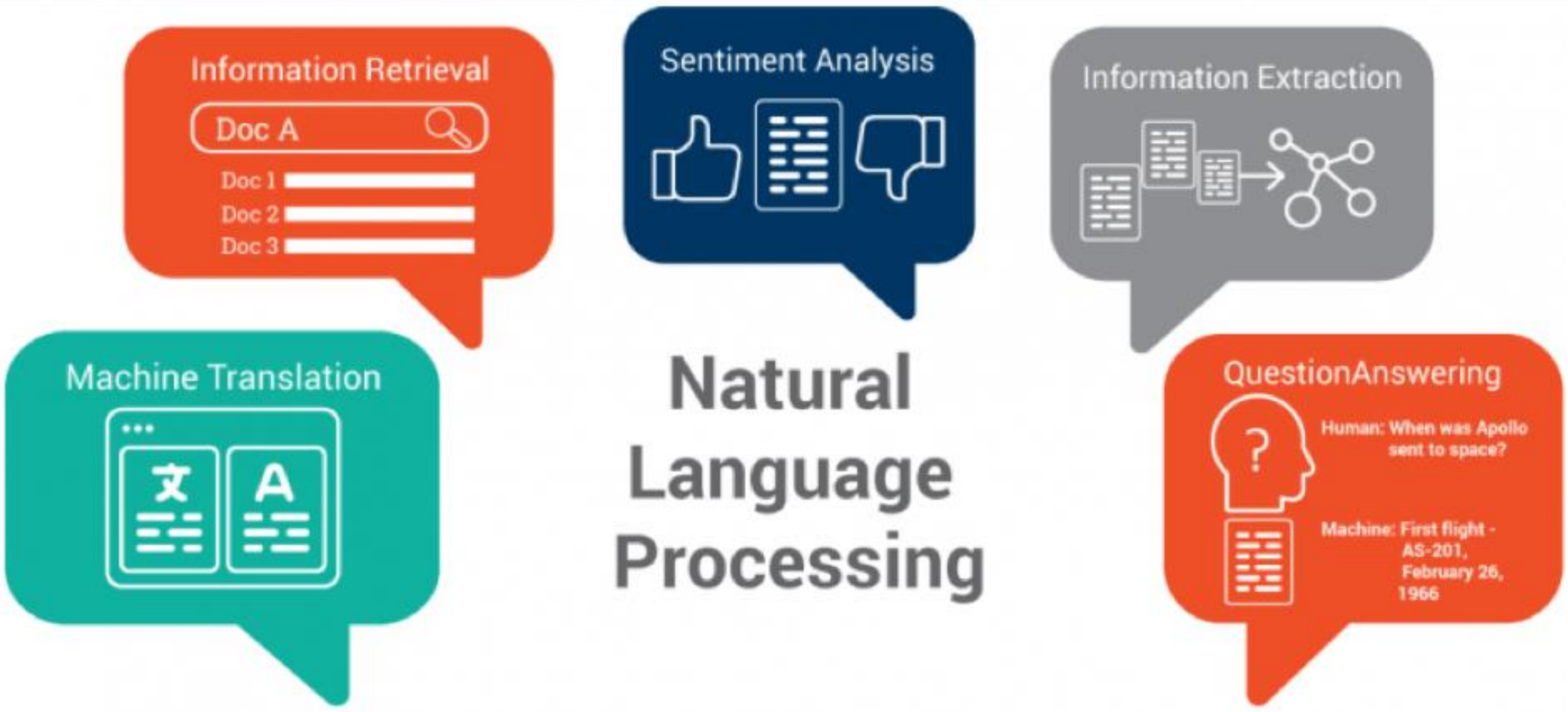
Natural Language Processing

- Word Embeddings
- Architectures for machine translation

Natural Language Processing

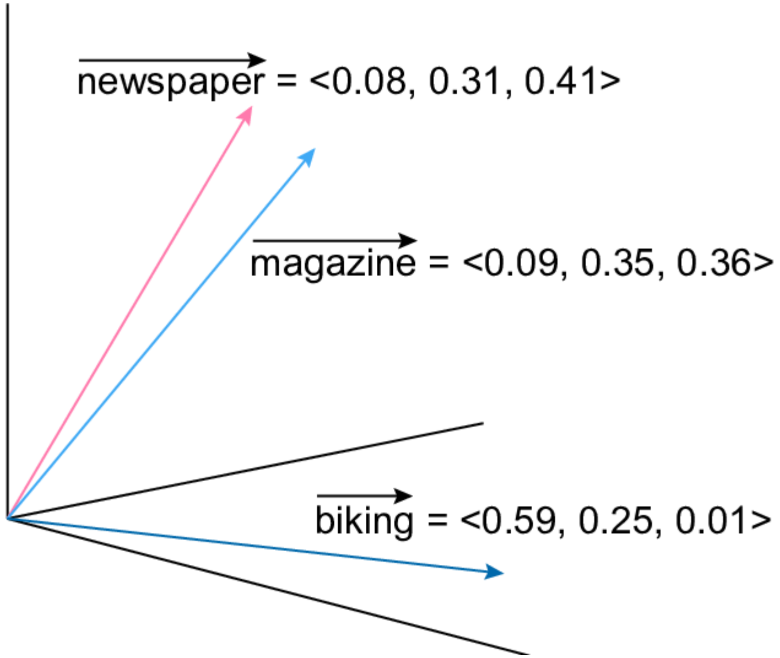
- Based (at least in part) on RNNs

- Deals with machine translation, automatic summarization, parsing, sentiment analysis,….



Word Embeddings

- Convert symbolic representations (words, dates, categories,…) into meaningful numbers
  - Capture the underlying semantic relations



- Several methods :

  1) Use a one-hot vector to represent words

  2) Feature vectors

  3) (Pre-trained) word embeddings

One-Hot Encoding

- Embed the colour “orange”

```
Orange = [1,0,0,0,0,0,…]
```

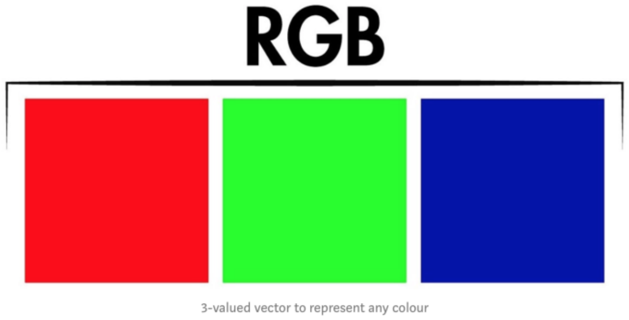
Each location in the vector represents a different colour

- Limitations :
  - Vectors can be prohibitively large
  - Assumption that there are no inherent relationships between any of the colours being embedded
    - Similarity between the vectors for “orange” and “red” will not be different to the similarity between the vectors for “orange” and “green”

Feature Vectors

- Vector able to represent any colour with only 3 values each time

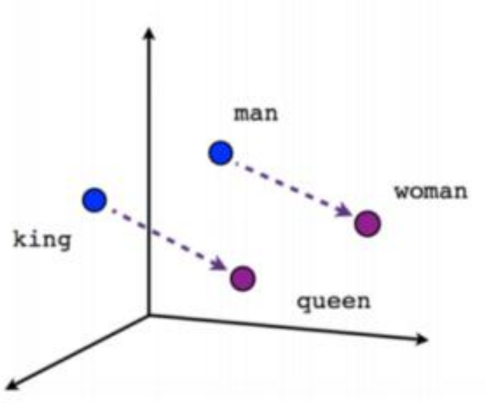
```
Red    = [1,0,0]
Green  = [0,1,0]
Orange = [1,0.5,0]
```



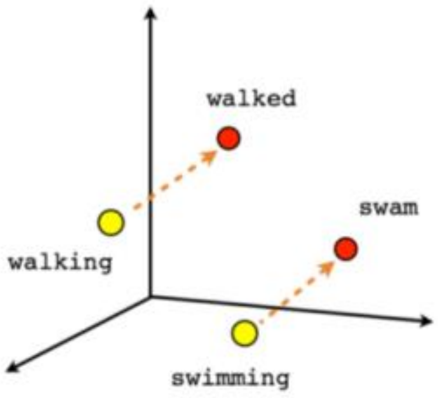
- Similarly, create fixed-length vectors that represent items like words (usually between 50 and 300 values)

Pre-trained Word Embeddings

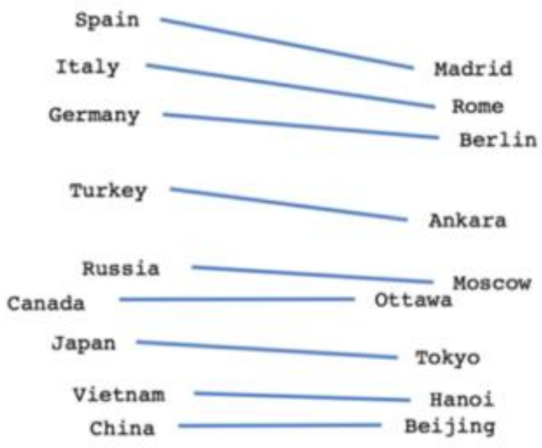
- Glove (2014) / Word2Vec (2014)



Male-Female

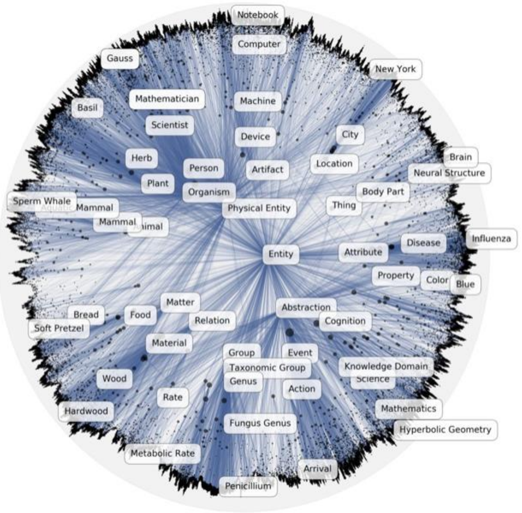


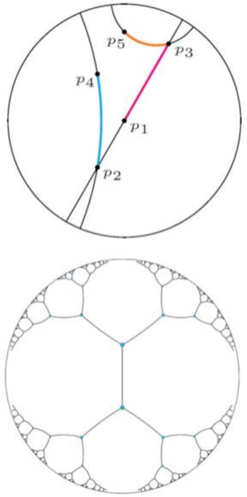
Verb tense



Country-Capital

- Poincaré





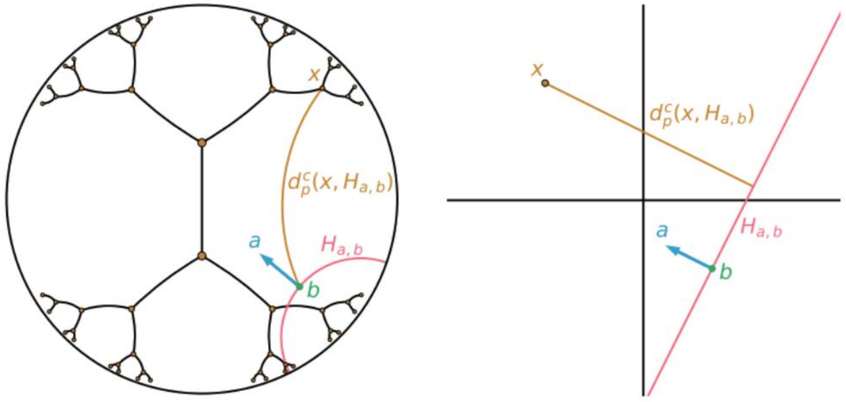
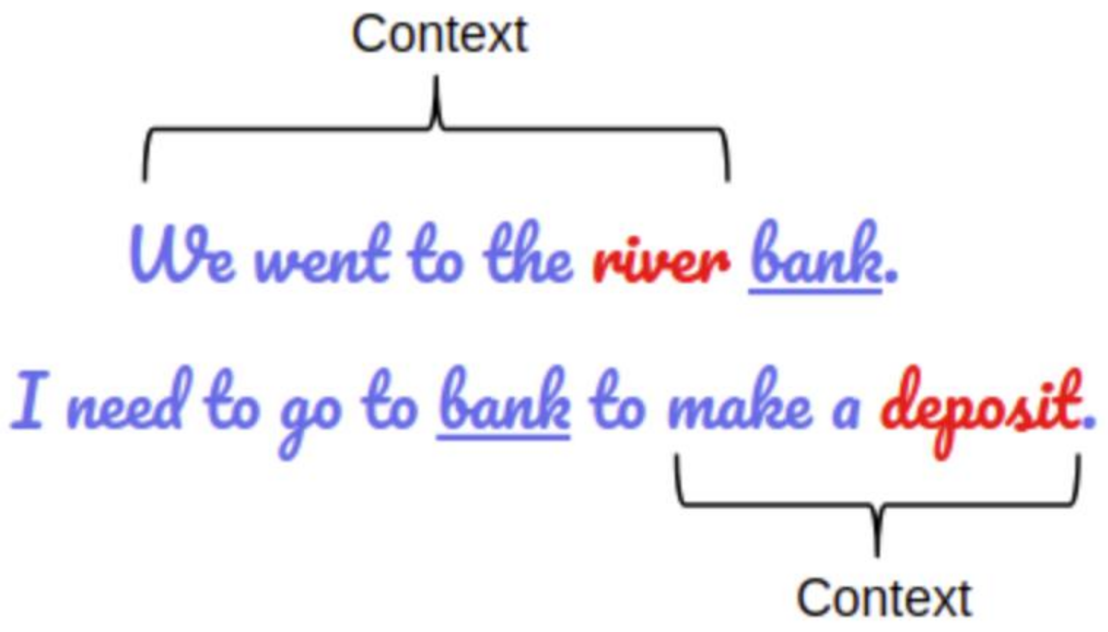


Figure 3: Illustration of an orthogonal projection on a hyperplane in a Poincaré disc $\mathbb{B}_c^2$ (Left) and an Euclidean plane (Right). Those hyperplanes are *decision boundaries*.

Limit of 2014 pre-trained word embeddings



- Do NOT take the context of the word into account
- Word2Vec will give the same vector for “bank” in both contexts

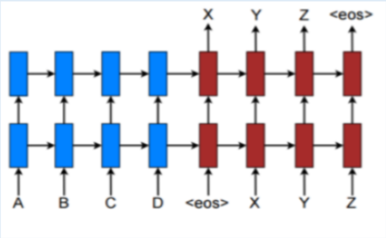
Machine Translation

(*French*) *Si mon tonton tond ton tonton, ton tonton sera tondu.*

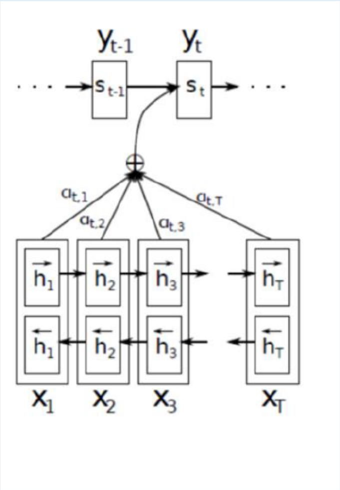
(*English*) *If my uncle shaves your uncle, your uncle will be shaved.*

$$\underset{t \in TARGET\_LANGUAGE}{\operatorname{argmax}} P(t|s) = \underset{t \in TARGET\_LANGUAGE}{\operatorname{argmax}} P(s|t) \times P(t)$$

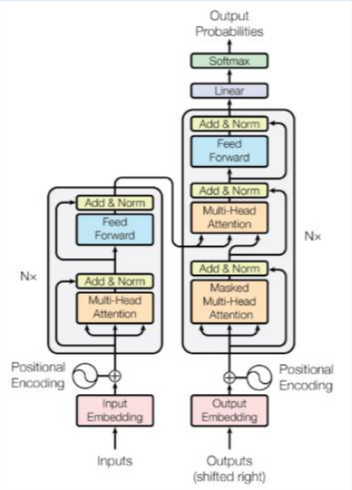
Phrase-based statistical MT with a translation model and a language model (Koehn et al. 2007)



RNN encoder-decoder with LSTM units (Suts-kever / Cho et al. 2014)



RNN encoder-decoder with attention model (Bahdanau et al. 2015)



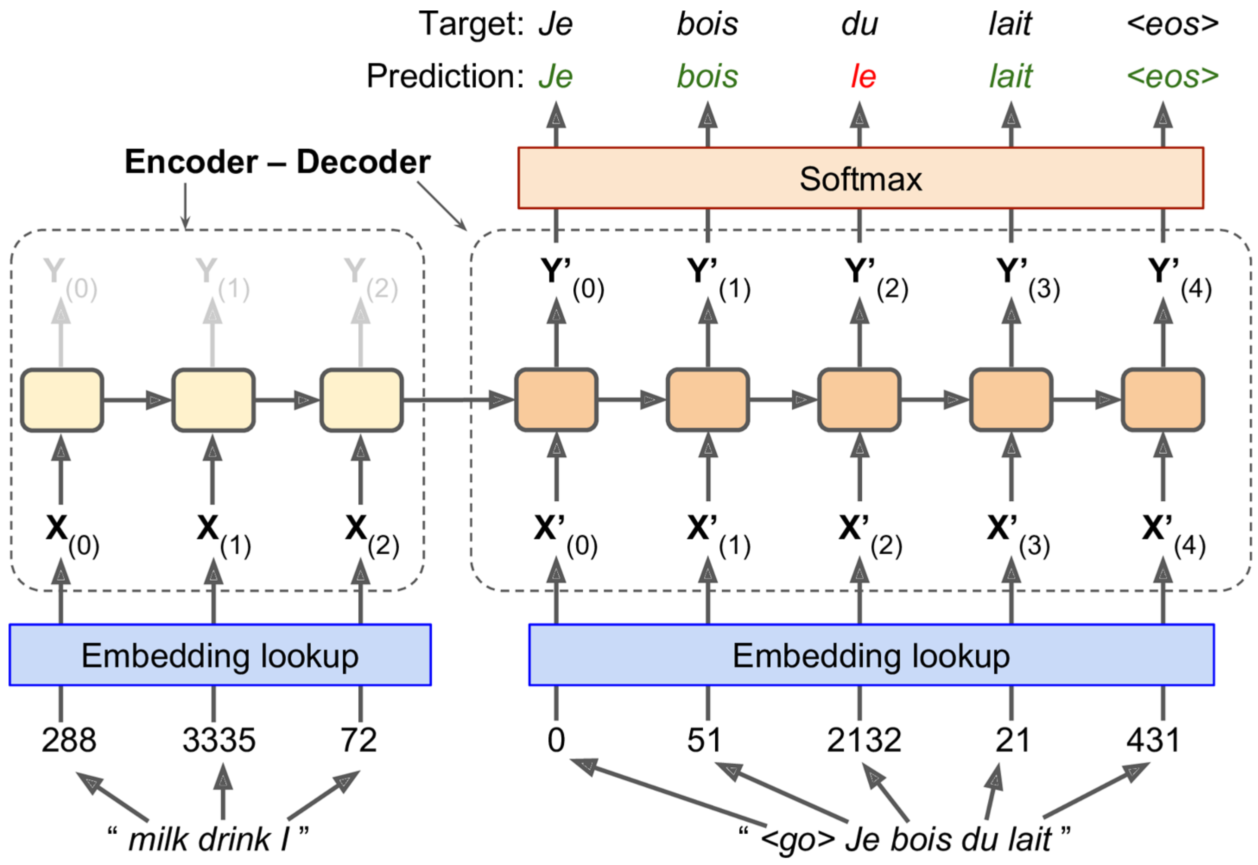
The Transformer: self-attention networks with positional encoding (Vaswani et al. 2017)

Courtesy Prof. A. Popescu

Time

RNN encoder-decoder (2014)

- Simple machine translation model (English → French)
  - English sentences fed to the encoder
  - French translations output by the decoder
  - French translations also used as inputs to the decoder (shifted by one step)



RNN encoder-decoder

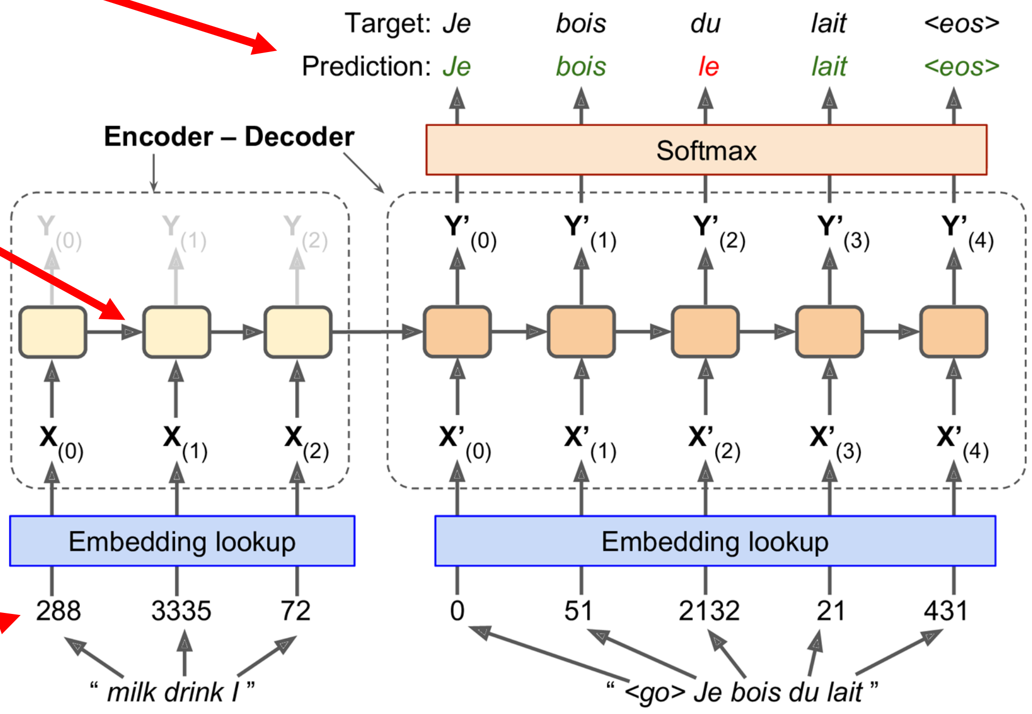
At each step, the decoder outputs a score for each word in the output vocabulary

The word with highest probability (softmax layer) is output

Hidden state used for the next input word

Word embedding that is fed to the encoder

Each word is initially represented by a simple integer identifier



Why is the English sentence reversed ?

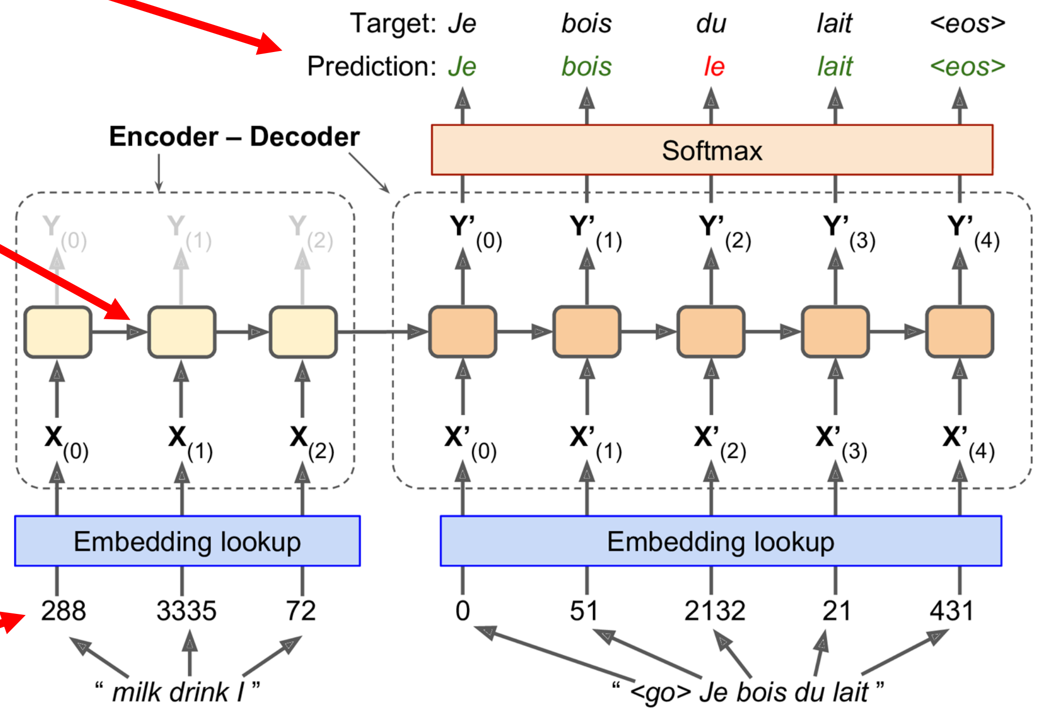
At each step, the decoder outputs a score for each word in the output vocabulary
The word with highest probability (softmax layer) is output

RNN encoder-decoder

Hidden state used for the next input word

Word embedding that is fed to the encoder

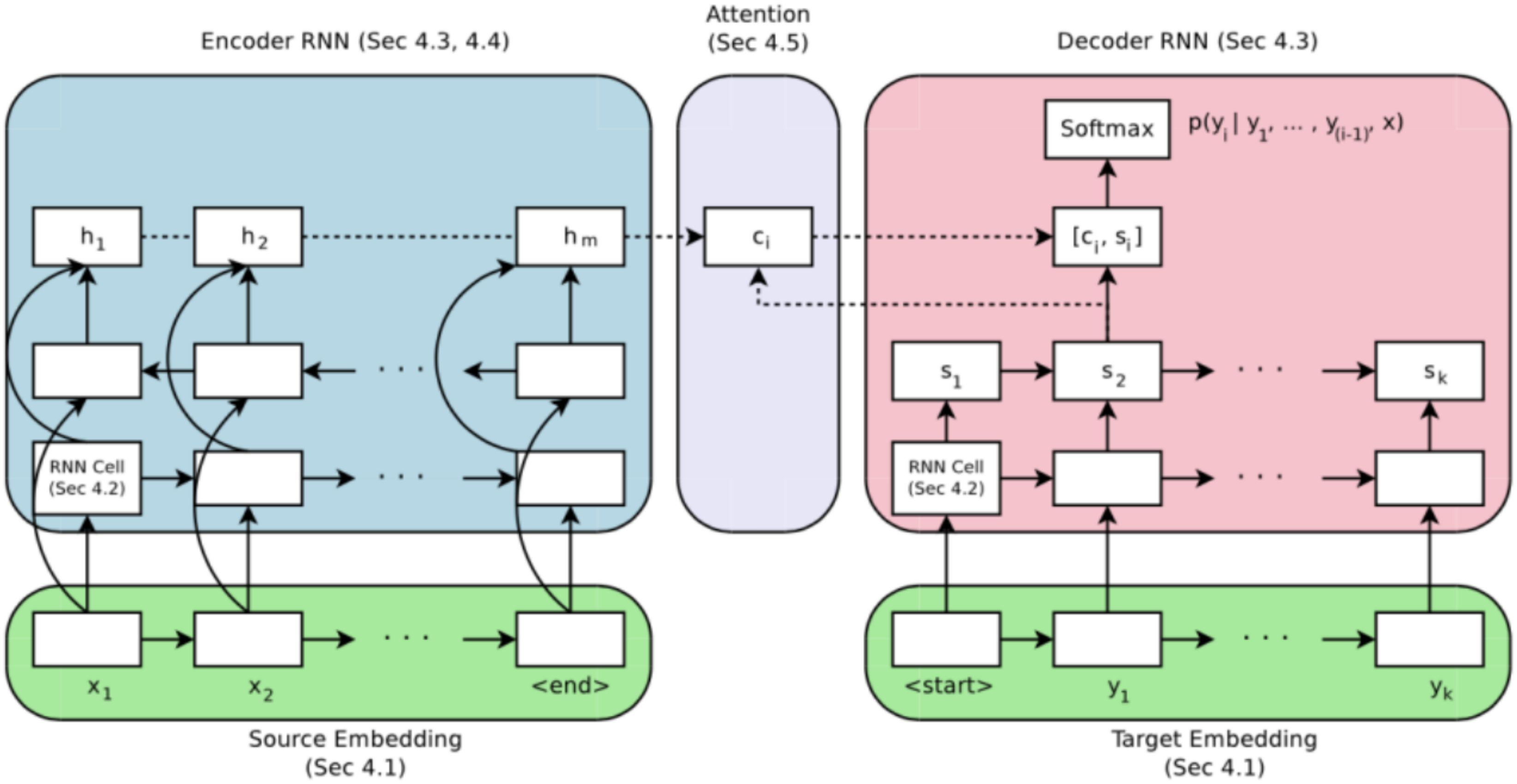
Each word is initially represented by a simple integer identifier



Why is the English sentence reversed ?

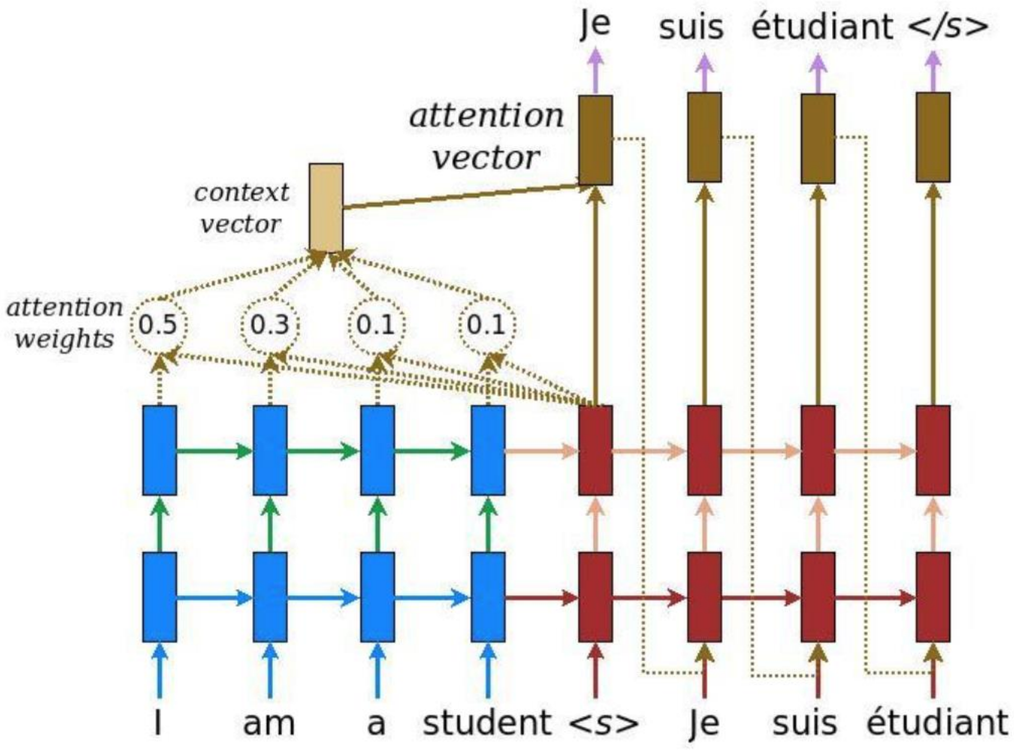
This ensures that the beginning of the English sentence will be fed last to the encoder, which is useful because it is the first thing that the decoder needs to translate.

RNN encoder-decoder with attention (2015)



Attention Mechanism

- Let the decoder learn to focus over a specific range of the input sequence

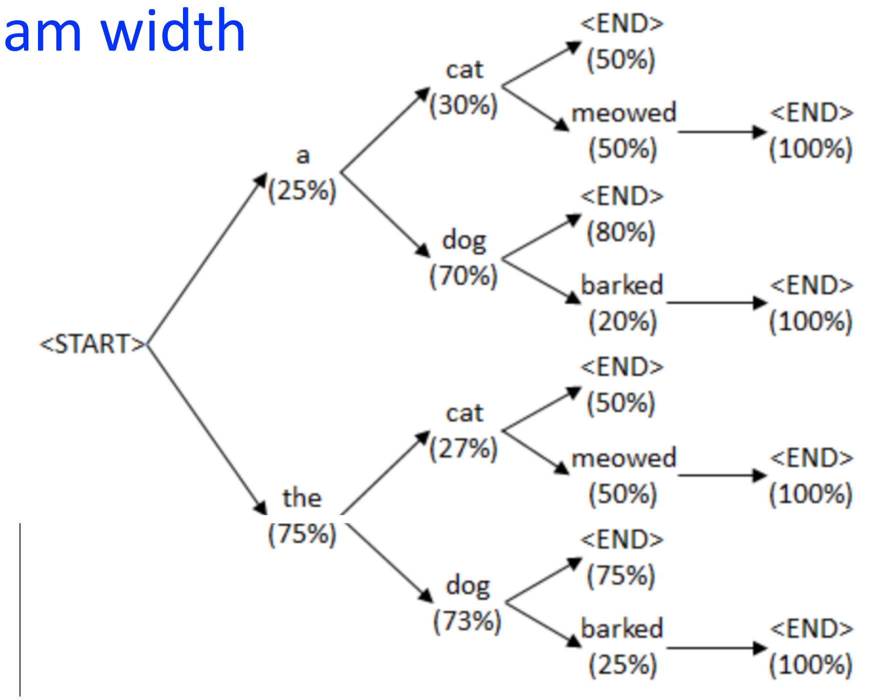


- Each input word is assigned a weight by the attention mechanism, which is then used by the decoder to predict the next word in the sentence

Beam Search

- Greedy search : always pick the *most probable word*
  - Known to NOT give the optimal solution

- Beam search (more exploratory) : search in a probability tree
  - *consider the top 10 most probable prefixes*
  - Maximize the total probability
  - Parameter : beam width



250,000 GPU hours on the standard WMT English to German translation task.

RNN encoder-decoder with attention

Hyperparameter	Value
embedding dim	512
rnn cell variant	LSTMCell
encoder depth	4
decoder depth	4
attention dim	512
attention type	Bahdanau
encoder	bidirectional
beam size	10
length penalty	1.0

Dim	newstest2013	Params
128	21.50 ± 0.16 (21.66)	36.13M
256	21.73 ± 0.09 (21.85)	46.20M
512	21.78 ± 0.05 (21.83)	66.32M
1024	21.36 ± 0.27 (21.67)	106.58M
2048	**21.86** ± 0.17 (22.08)	187.09M

Cell	newstest2013	Params
LSTM	**22.22** ± 0.08 (22.33)	68.95M
GRU	21.78 ± 0.05 (21.83)	66.32M
Vanilla-Dec	15.38 ± 0.28 (15.73)	63.18M

Beam	newstest2013	Params
B1	20.66 ± 0.31 (21.08)	66.32M
B3	21.55 ± 0.26 (21.94)	66.32M
B5	21.60 ± 0.28 (22.03)	66.32M
B10	21.57 ± 0.26 (21.91)	66.32M
B25	21.47 ± 0.30 (21.77)	66.32M
B100	21.10 ± 0.31 (21.39)	66.32M
B10-LP-0.5	21.71 ± 0.25 (22.04)	66.32M
B10-LP-1.0	**21.80** ± 0.25 (22.16)	66.32M

Attention	newstest2013	Params
Mul-128	22.03 ± 0.08 (22.14)	65.73M
Mul-256	22.33 ± 0.28 (22.64)	65.93M
Mul-512	21.78 ± 0.05 (21.83)	66.32M
Mul-1024	18.22 ± 0.03 (18.26)	67.11M
Add-128	22.23 ± 0.11 (22.38)	65.73M
Add-256	22.33 ± 0.04 (22.39)	65.93M
Add-512	**22.47** ± 0.27 (22.79)	66.33M
Add-1028	22.10 ± 0.18 (22.36)	67.11M
None-State	9.98 ± 0.28 (10.25)	64.23M
None-Input	11.57 ± 0.30 (11.85)	64.49M

The Transformer

- Model used for machine translation and text summarization (learns contextual relations between words in a text)
  - Building block of most state-of-the-art architectures in NLP replacing gated RNNs (LSTM)

- Based on the attention mechanism *without recurrent sequential processing*
  - does NOT require that the sequence be processed in order (unlike RNNs) → Parallelization (unlike RNNs)

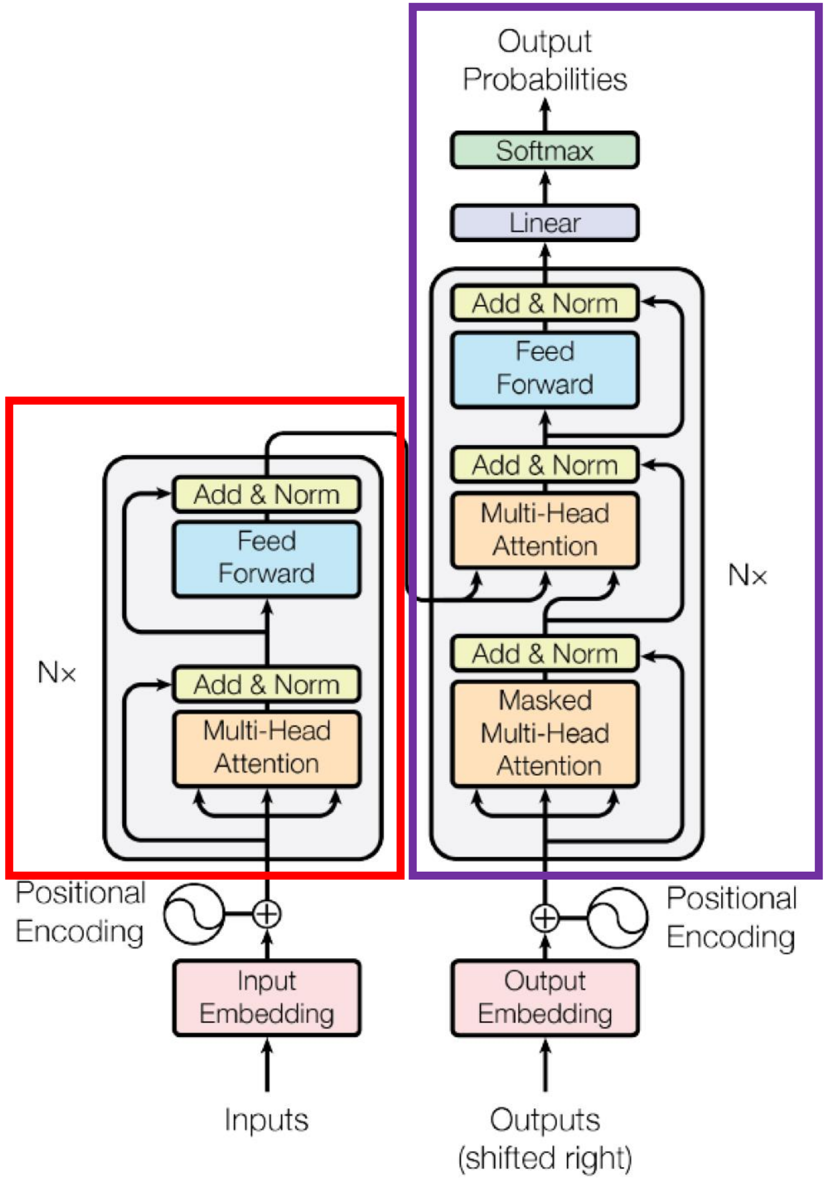
- All token processed at the same time and attention weights between them calculated
  - training on more data

• a set of encoders chained together and a set of decoders chained together

The Transformer: Architecture

Decoder : prediction for the task

- self-attention mechanism
- *attention mechanism over the encodings*
- feed-forward NN



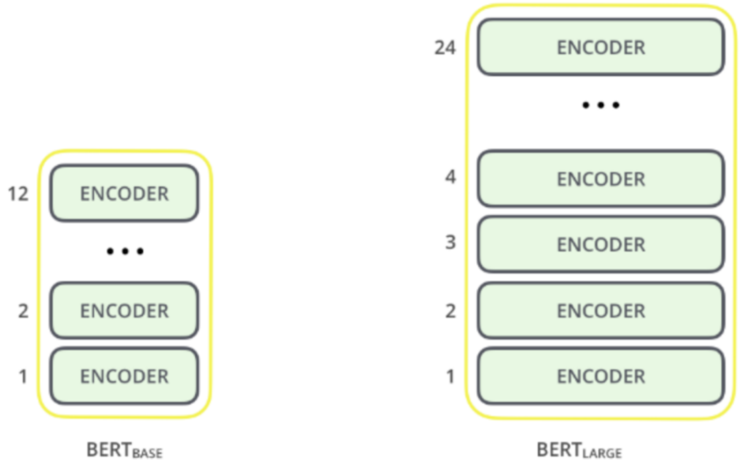
Encoder : reads the text input

- self-attention mechanism
- feed-forward NN

BERT (embedding)

- Bidirectional Encoder Representation from Transformers (BERT)
  - Based on the Transformer architecture (only encoder part)
  - Bidirectional training provides a deeper sense of language context (learns the context of a word based on all of its left and right surroundings)

110 mio parameters



340 mio parameters

Reading

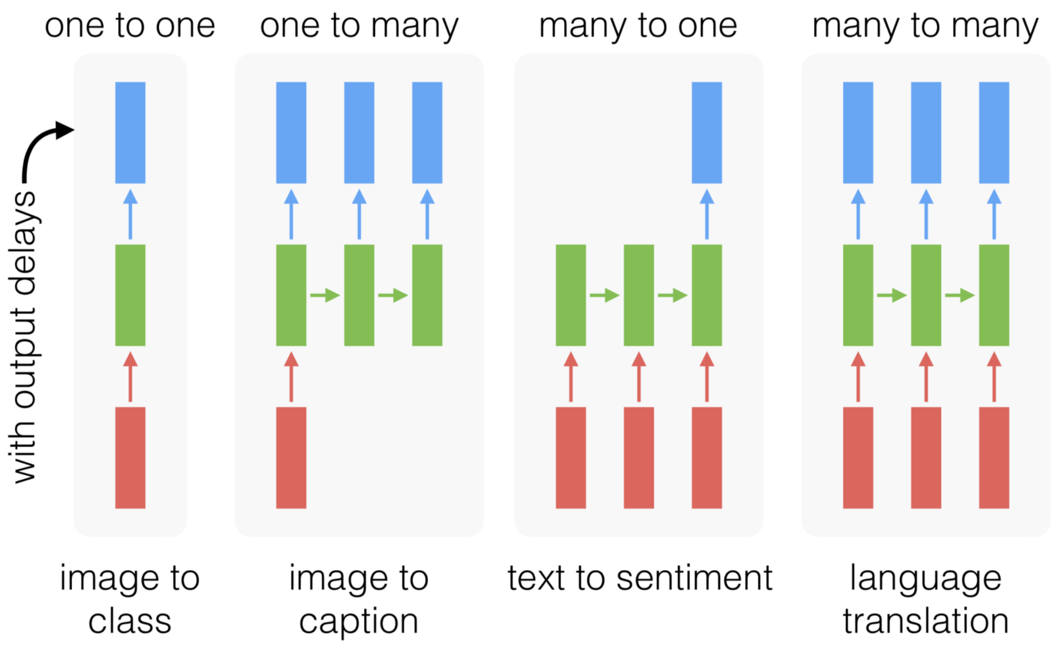
- Pre-trained on the entire Wikipedia (2500 million words) and Book Corpus (800 million words)

- can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks

Pretrained Word Embeddings

- word2vec (Google)
  - O. Levy & Y. Goldberg, “Neural Word Embeddings as ImplicitMatrix Factorization”, *NIPS*2014
- GloVe (Stanford)
  - J. Pennington, R. Socher, C. D. Manning, “GloVe: Global Vectorsfor Word Representation”, *EMNLP*2014
- fastText (Facebook)
  - P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, “EnrichingWord VectorswithSubwordInformation”, *TACL*2017
- ELMo (AllenNLP)
  - M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, “Deepcontextualizedwordrepresentations”, *NAACL*2018
- BERT (Google)
  - J. Devlin, M.W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of DeepBidirectionalTransformers for LanguageUnderstanding", *Arxiv*oct. 2018

More examples



- One-to-one : Text generation
- One-to-many : Image to caption

1. Text Generation

- Given a character, or a sequence of characters, what is the most probable next character?

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou that art now the world's fresh ornament,
And only herald to the gaudy spring,
Within thine own bud buriest thy content,
And tender churl mak'st waste in niggarding:
  Pity the world, or else this glutton be,
  To eat the world's due, by the grave and thee.

The Sonnets,
W. Shakespeare

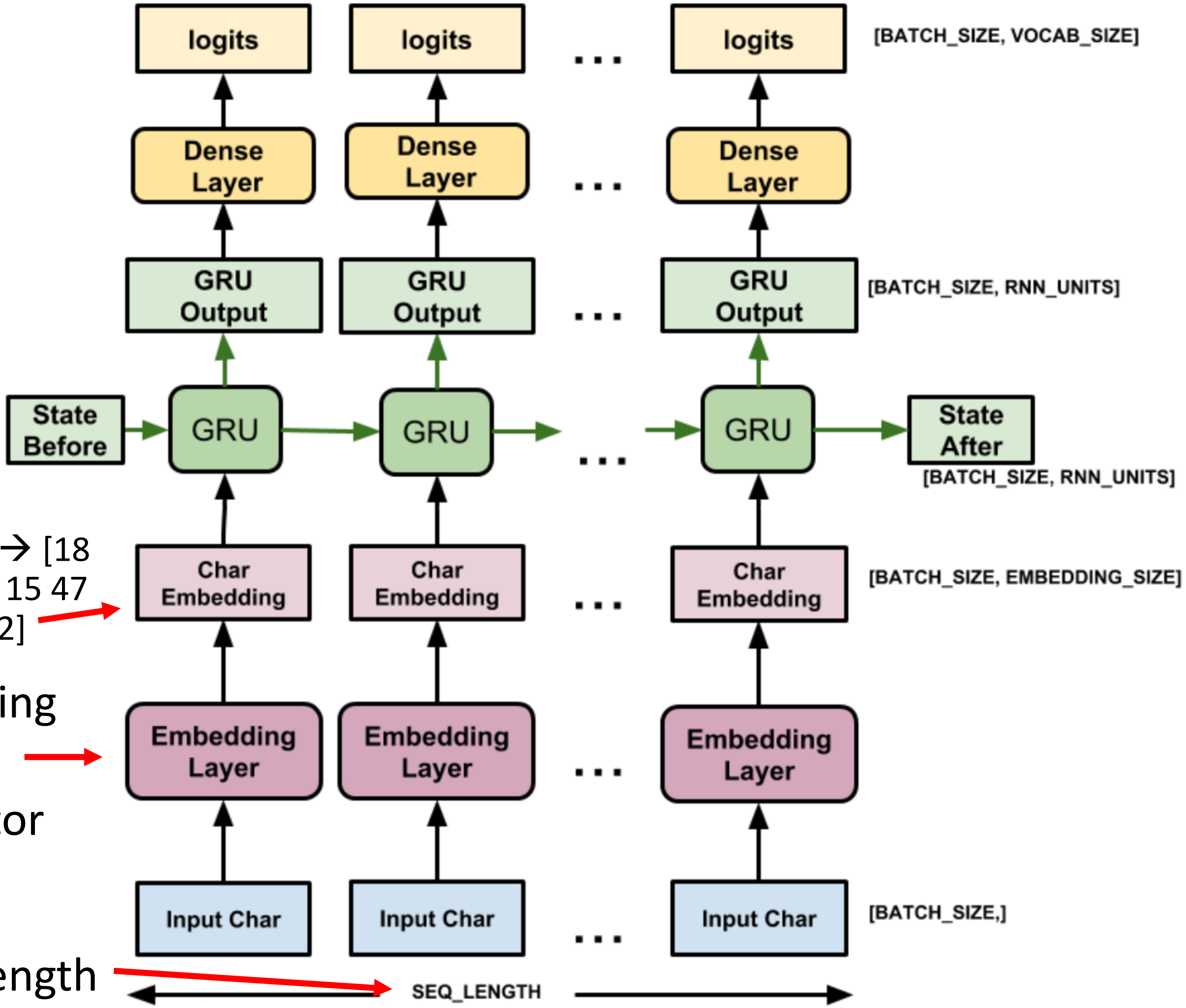
- For each sequence, the corresponding target contain the same length of text, except shifted one character to the right
  - Input sequence : "Hell"
  - Target sequence : "ello"

Training

'**First Citizen'** → [18 47 56 57 58 1 15 47 58 47 64 43 52]

Lookup table mapping the numbers of the characters to a vector

divide the text into sequences of given length



Result

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

Two-Minute Papers

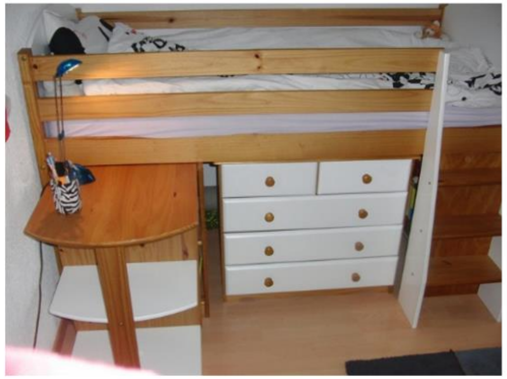
2. Image to Caption

- Given an image, what is the most probable caption describing it ?
- Microsoft-COCO dataset (>82000 images)


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

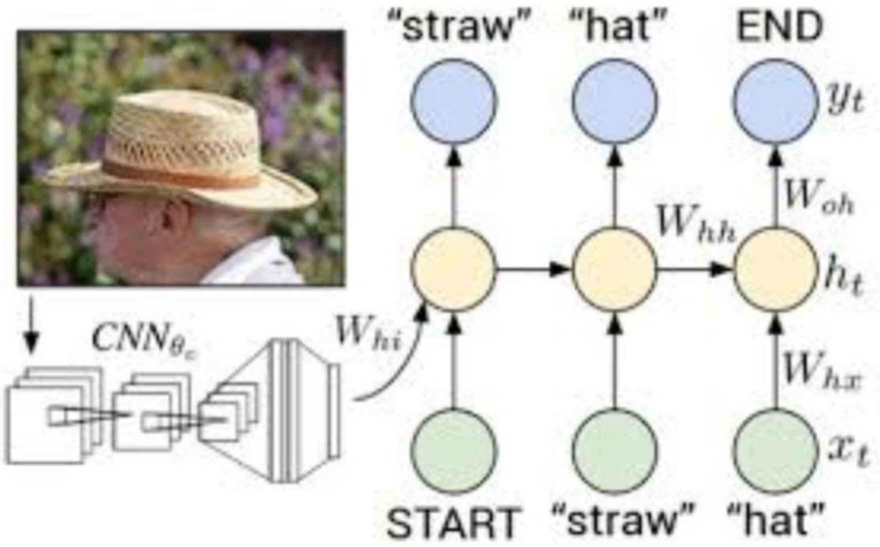

A large bus sitting next to a very tall building.

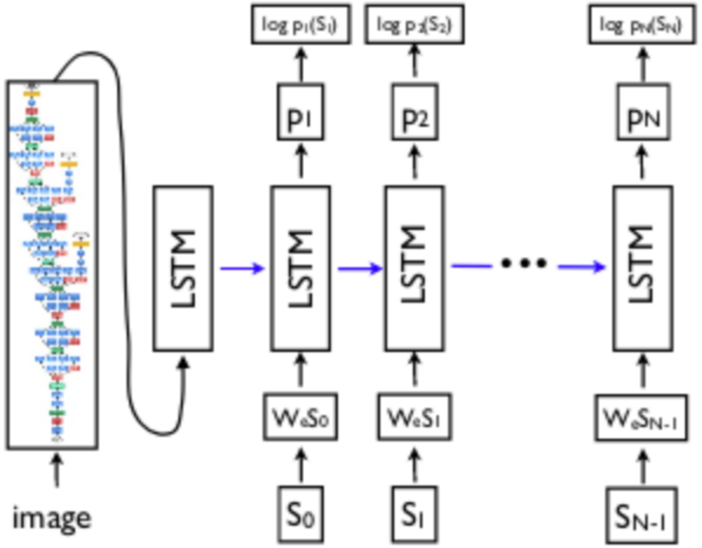

A horse carrying a large load of hay and two people sitting on it.


Bunk bed with a narrow shelf sitting underneath it.

Training

- Idea is to use a CNN as encoder and a RNN as decoder





- CNN encoder produces a representation of the input image by embedding it to a fixed-length vector
  - Inception network
  - Attention mechanism to increase performance → grid of vectors

- RNN decoder uses as input the last hidden layer of the CNN. It returns the predictions and the decoder hidden state.

Result

man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

two young girls are playing with lego toy.

boy is doing backflip on wakeboard.

Two-Minute Papers

Quiz

https://b.socrative.com/login/student/

Room : CONTI6128