Neural Machine Translation for language professionals

A GALA talk by
Roland Meertens

Contents

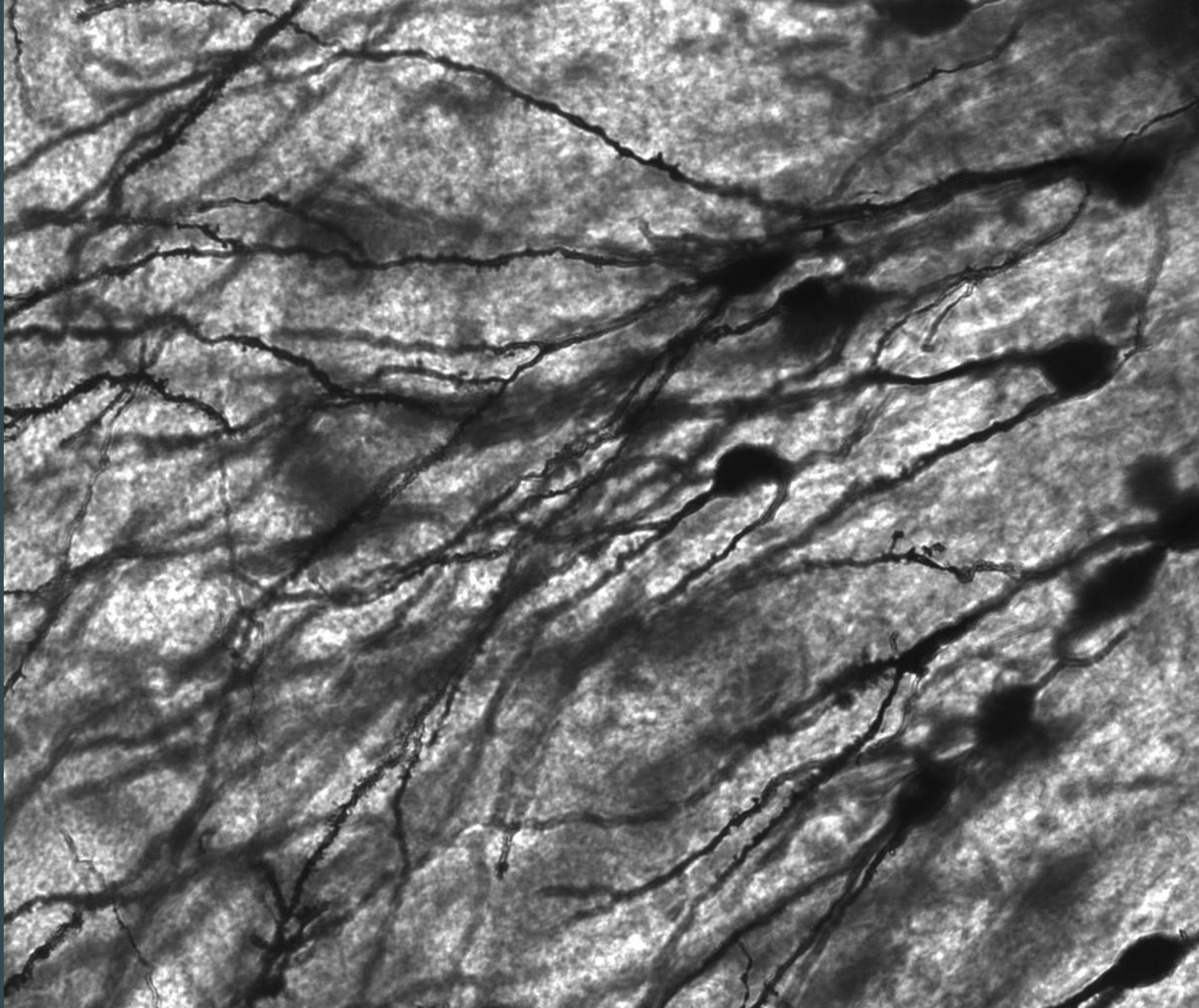
- Basics of neural networks
- Application of neural networks to machine translation
- Comparison to statistical methods
- Other applications of neural networks for natural language
- Reason for this ‘revolution’

Basics of neural networks

Brain

This is how your brain works:

Each brain cell (neuron) is connected to many other neurons, and can get activated by many other neurons.



Let us try to predict if a sentence from an email is spam.

Weights are randomly initialised.

Basic task

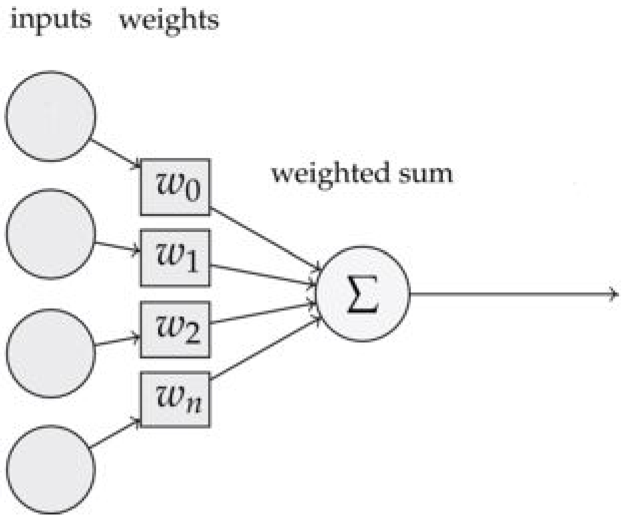
Contains... (0 or 1)

Buy

Enlargement

Meeting

Manager



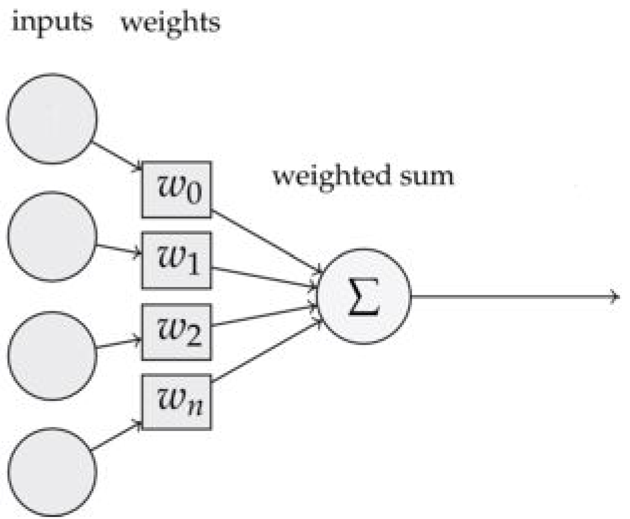
Probability of spam

Let us try to predict if a sentence from an email is spam.

Example:
"Buy this enlargement today for only X dollars"

Basic task

Contains... (0 or 1)	
Buy	1.0
Enlargement	1.0
Meeting	0.0
Manager	0.0



Probability of spam

1 * w0 + 1 * w1 + 0 * w2 + 0 * wn = spam score

Let us try to predict if a sentence from an email is spam.

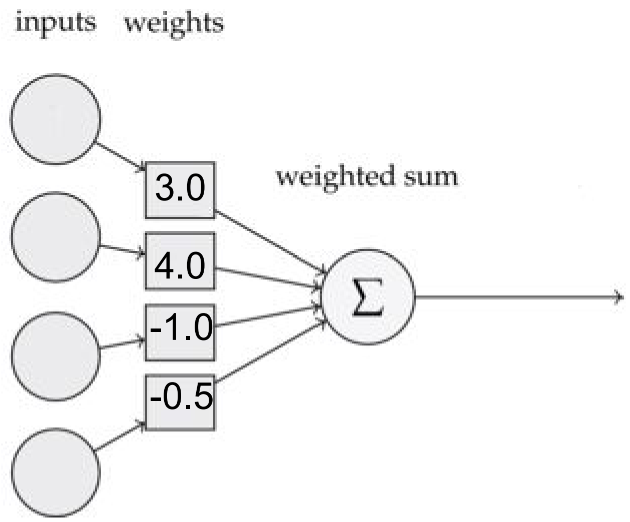
Initially all inputs are random, so all output is random.

Example:
"Buy this enlargement today for only X dollars"

Basic task

Contains... (0 or 1)

Buy	1.0
Enlargement	1.0
Meeting	0.0
Manager	0.0



Probability of spam

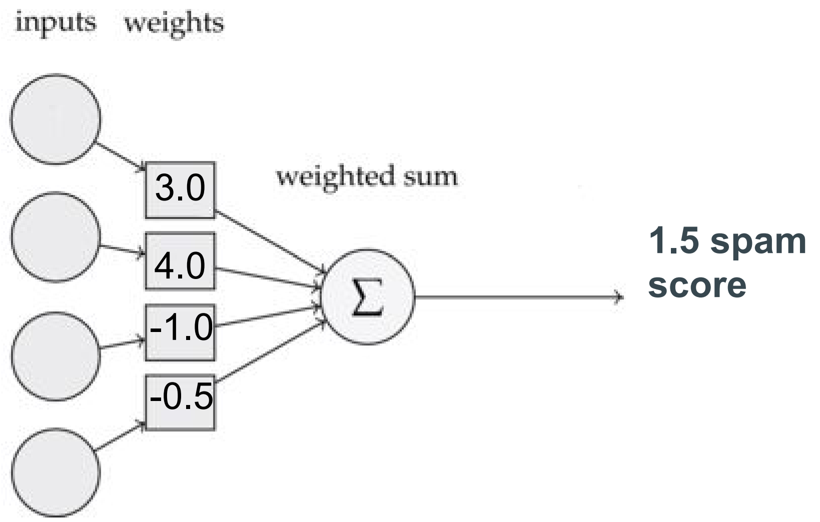
1 * 3.0 + 1 * 4.0 + 0 * -1.0 + 0 * -0.5 = 7.0 spam score

Let us learn if a sentence from an email is spam.

Example:
"Tomorrow we have a meeting to buy something for the birthday of our manager"

Feedback

Contains... (0 or 1)	
Buy	1.0
Enlargement	0.0
Meeting	1.0
Manager	1.0



1.5 spam score

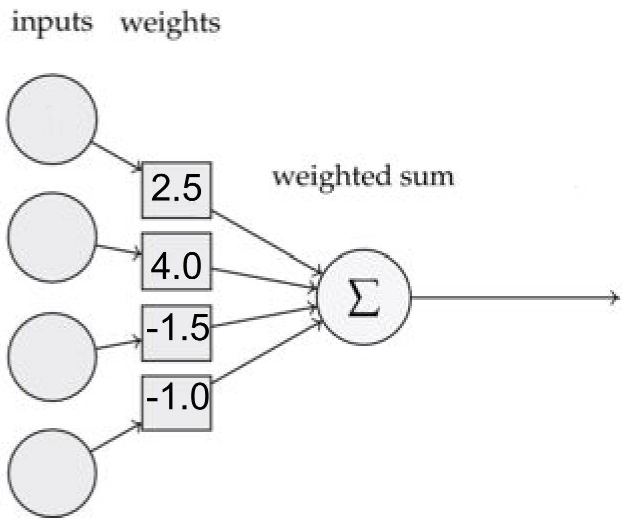
This sentence should lead to a negative response. Let us update the "Buy" weight, the "Meeting" weight, and the "Manager" weight.

Let us learn if a sentence from an email is spam.

Example:
"Tomorrow we have a meeting to buy something for the birthday of our manager"

Feedback

Contains... (0 or 1)	
Buy	1.0
Enlargement	0.0
Meeting	1.0
Manager	1.0



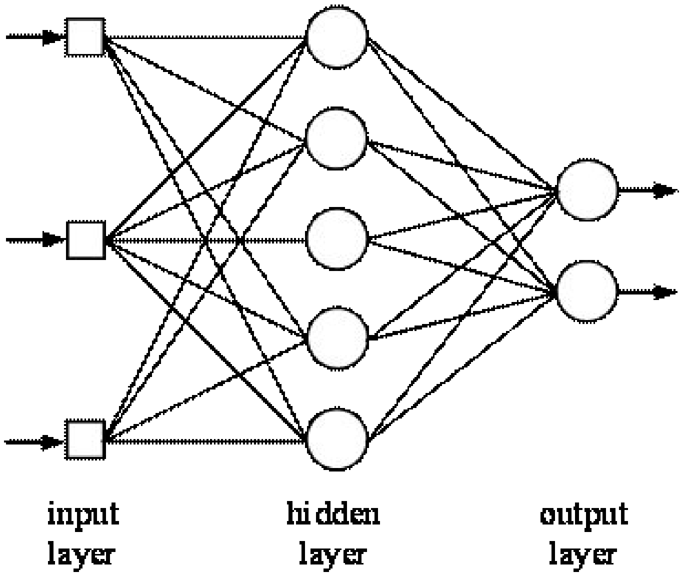
0 spam score

After adjusting our weights based on this one sentence, our network is a bit better!

Note: in this case a combination of words makes this sentence good

Combinations of inputs

Spam is based on co-occurring words. Making each co-occurrence a separate input leads to a REALLY big network. The network can learn these co-occurrences with a "hidden layer"

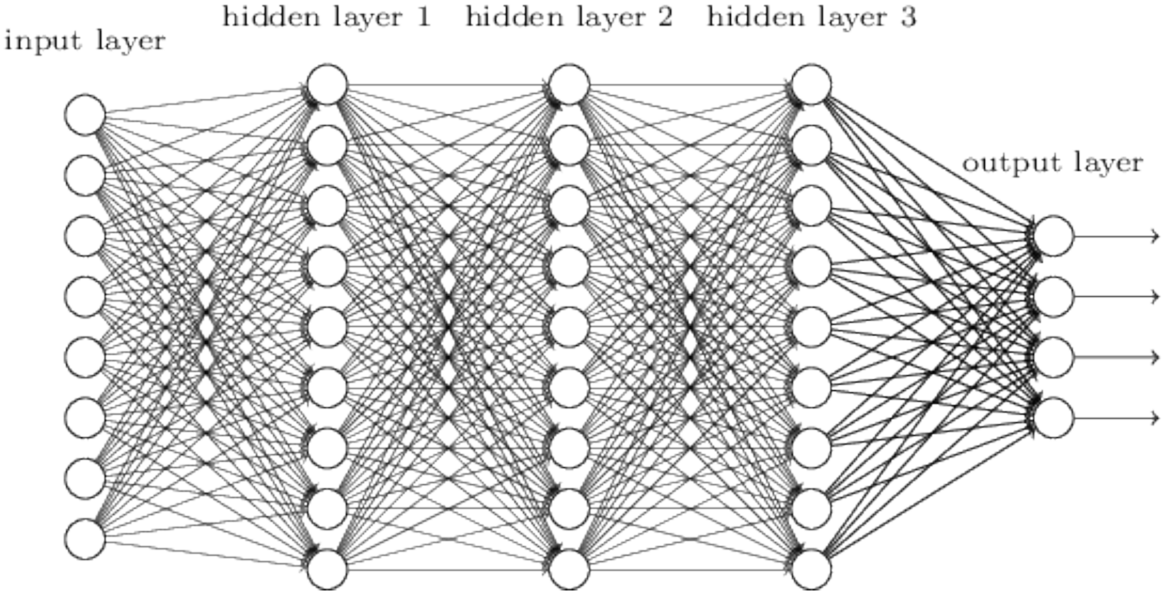


Each line is a "weight", this sample has 25 weights the network has to "learn".
Note that it does not manually set what co-occurrences are important. It can learn this from a corpus.

What about output?

Neural networks can give output to multiple neurons.

For example, instead of only determining which email is spam, a neural network can also categorize all your email into separate folders.



Neural networks

Everything that is a "numbers in, numbers out" problem could be learned with a neural network.

Neural networks

Everything that is a "numbers in, numbers out" problem could be learned with a neural network.

Note the "could be learned": large networks require a lot of data.

Neural networks

Everything that is a "numbers in, numbers out" problem could be learned with a neural network.

Important: you cannot define a "rule" (such as: enlargement always means spam) in your network, they always have to learn everything.

Words and neural networks

The English language has many words. The spam filter built before would become very big if it kept encoding each word as a one or a zero.

Smart tricks

For machine learning a really large amount of inputs and a really large amount of outputs would be terrible, as a large amount of parameters is difficult to learn.

Let us use embeddings instead of words!

Embed

Represent words using an N-dimensional space.

Similar words should be close together.

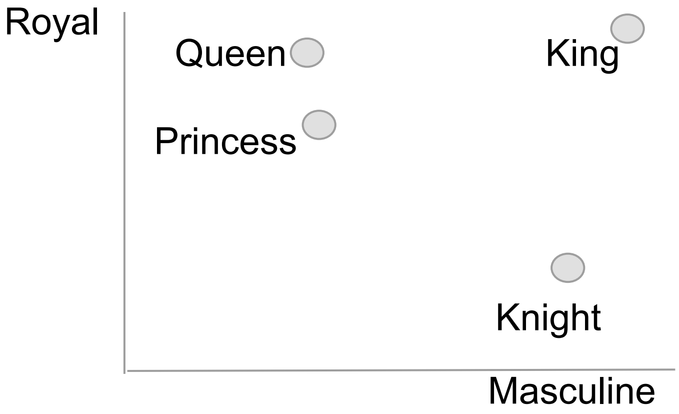
Embed

Represent words using an N-dimensional space.

Example words:

King
Queen
Princess
Knight

Possible representation



Embed

Represent words using an N-dimensional space.

Example words: Possible representation

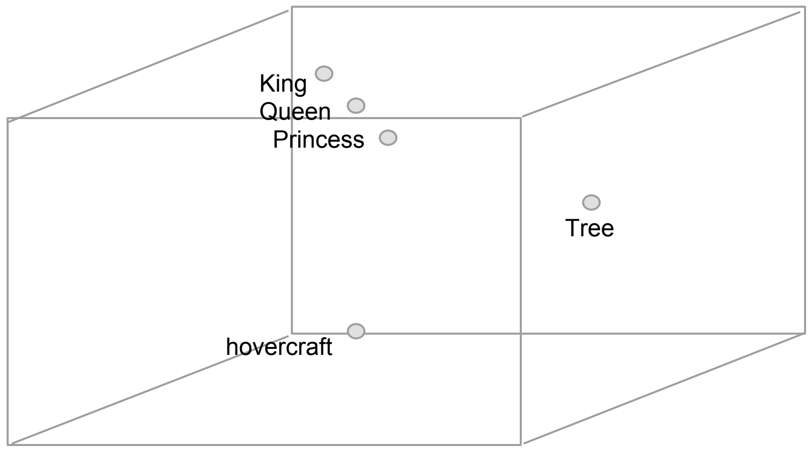
King

Queen

Princess

Hovercraft

Tree

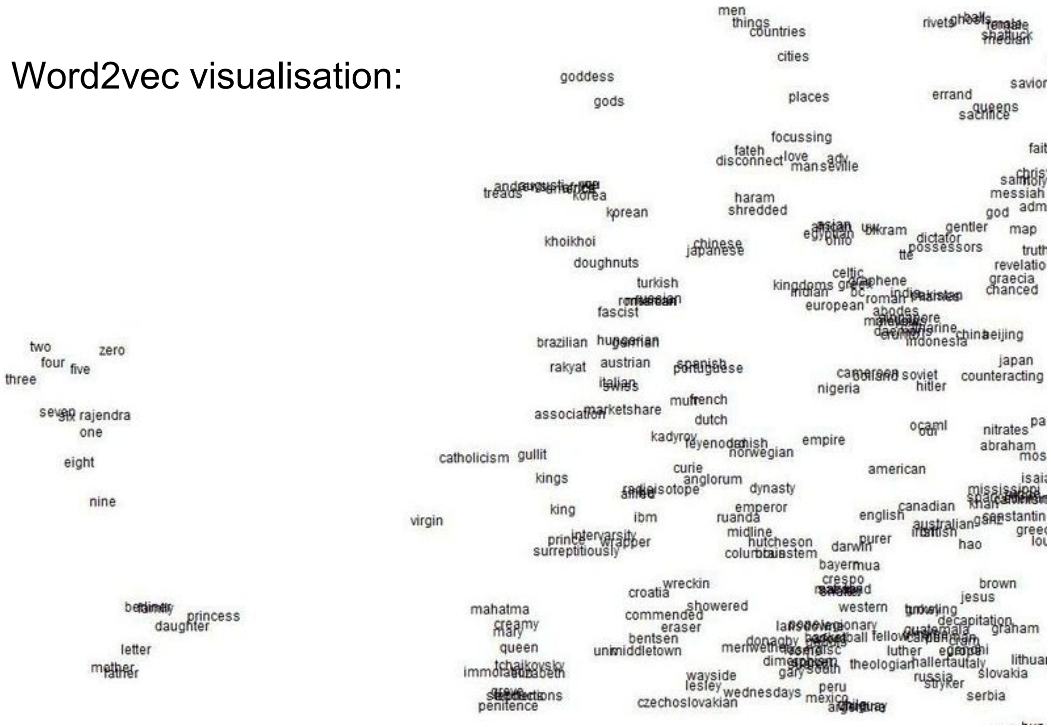


Embeddings

Good way of representing each word as a set of N-numbers (example: 32 numbers) reducing our input from thousands of numbers per word to 32 per word.

Embeddings

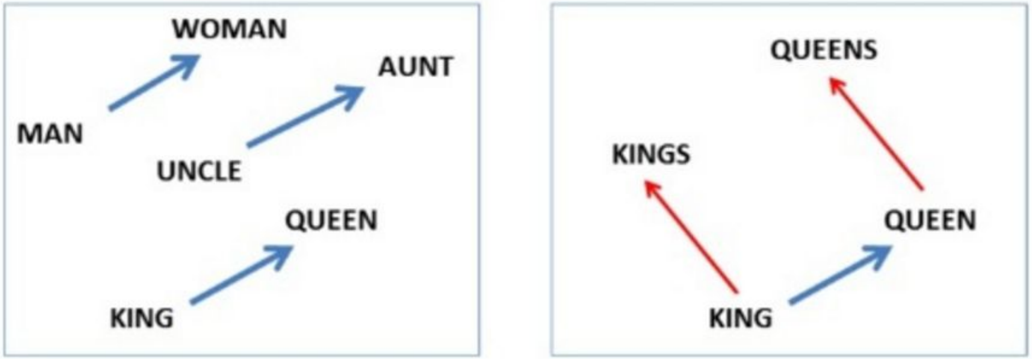
Word2vec visualisation:



Embeddings

Word2vec visualisation:

vec(“man”) – vec(“king”) + vec(“woman”) = vec(“queen”)



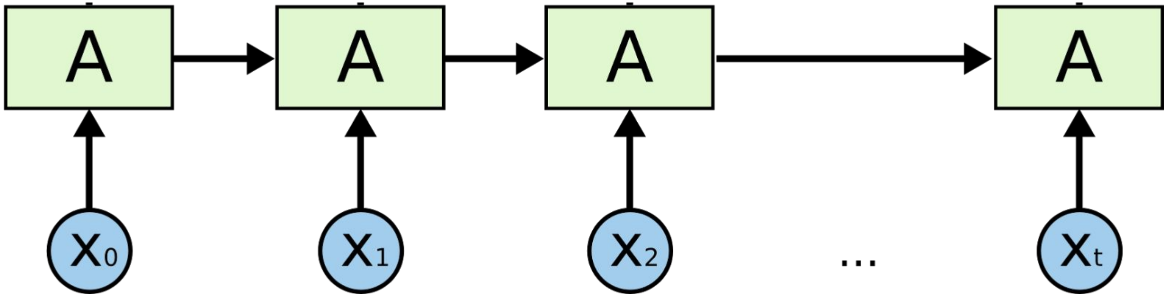
Summary

So far we have covered neural network and embeddings. Next up are the recurrent neural networks with which one can read a sentence or create a sentence.

Application of neural networks to machine translation

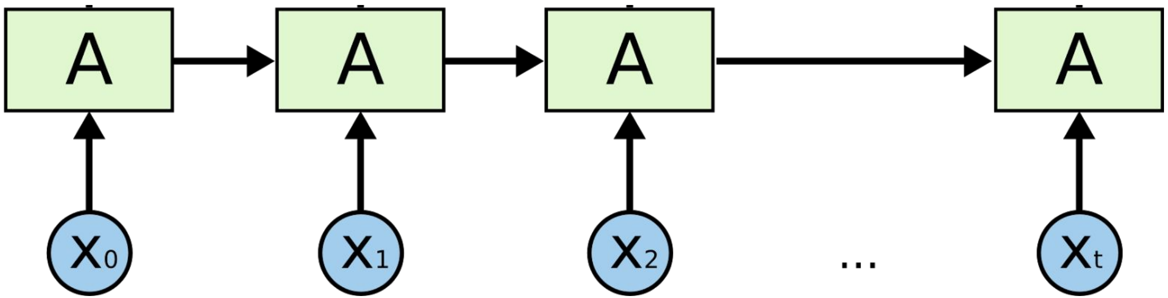
Encoding

Neural machine translation works by reading a sentence word for word, and saving the representation of each word you read so far.



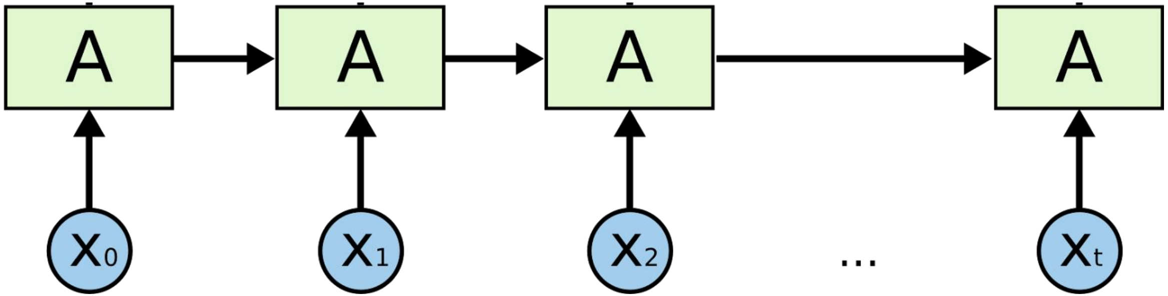
Encoding

These can be seen as "painting a picture". Each word adds and removes something to the painting.



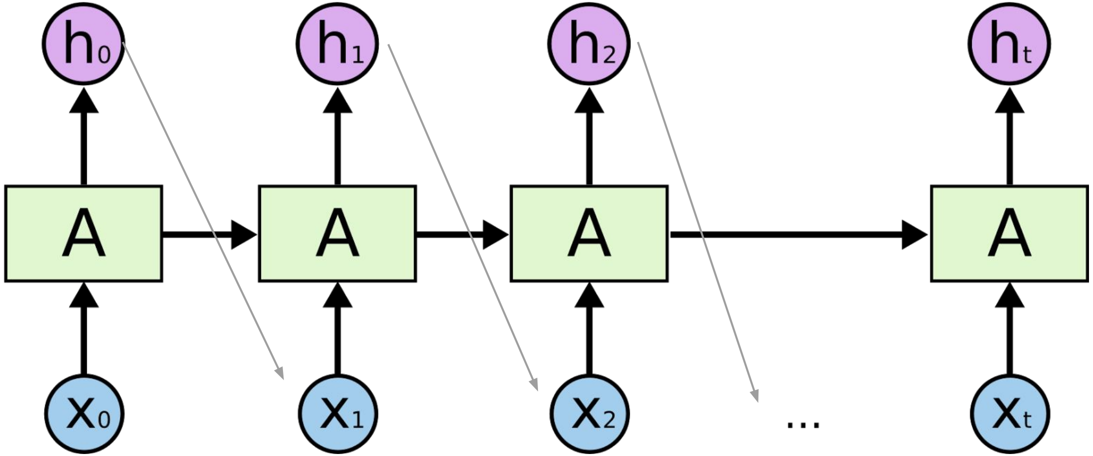
Encoding

The result is a list of numbers, or "the painting". This is a numerical representation of the input sentence. (The last A is the representation of the whole sentence).



Decoding

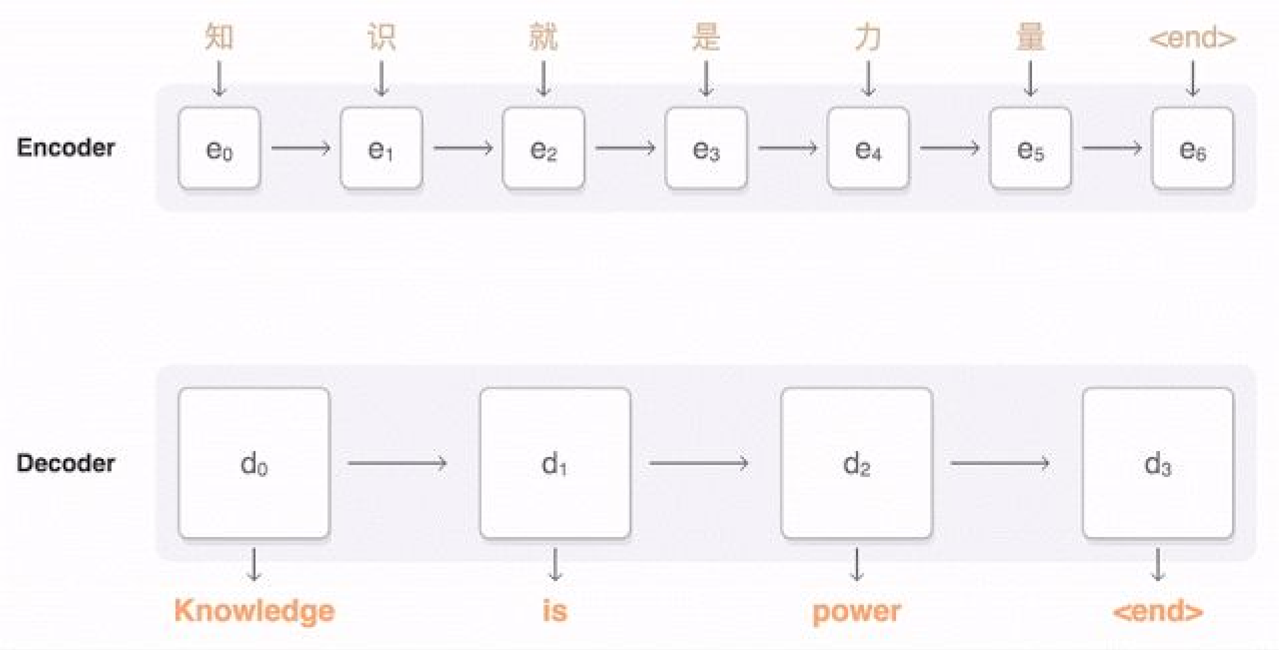
With this input you can create a sequence generating network that, based on the previous word generates the next word.



Attention mechanism

Words have a certain order. Aligning input and output is important. This alignment can be seen as an "attention mechanism". Based on the outputted word, where do you focus on the input? And, this too, can be learned using neural networks.

Attention mechanism



Comparison to statistical methods

SMT and NMT

Errors SMT makes:
- Poor grammar
- Wrong word choices
- Name translation (Mister Butters) -> (Meneer Boters)
- Unknown words

Infrequent words problem

SMT simple takes the translation

NMT has to learn what the word mean

Solution: subwords!

English	Dutch
Interactive	Interactief
Internationaal	International
Internet	Internet
Interesting	Interessant

Infrequent words problem

SMT simple takes the translation

NMT has to learn what the word mean

Solution: subwords!

English	Dutch
Interactive	Interactief
Inter@@ active	Inter@@ actief
Interesting	Interessant
Inter@@ esting	Inter@@ ess@@ ant

Other applications for neural networks
&
Reason for this revolution

Other applications of neural networks for natural language

- Automatic post-editing
  - The ability to translate using a traditional engine, neural networks fix the mistakes this engine makes
- Quality estimation
  - Estimation of its quality after translating a sentence
- Part Of Speech tagging
  - The ability to aid traditional translation engines

Reason for this revolution

- Discovery of embeddings
- More knowledge about sequence to sequence
- Faster computers (thanks gamers!)

Summary

- NMT outperforms SMT, but:
- Needs a lot of data
- Has problems with rare words