A gentle introduction to Deep Learning

Part 2: Natural Language Processing

Alina Miron
alina.miron@brunel.ac.uk

16th of November 2021

Sequence models

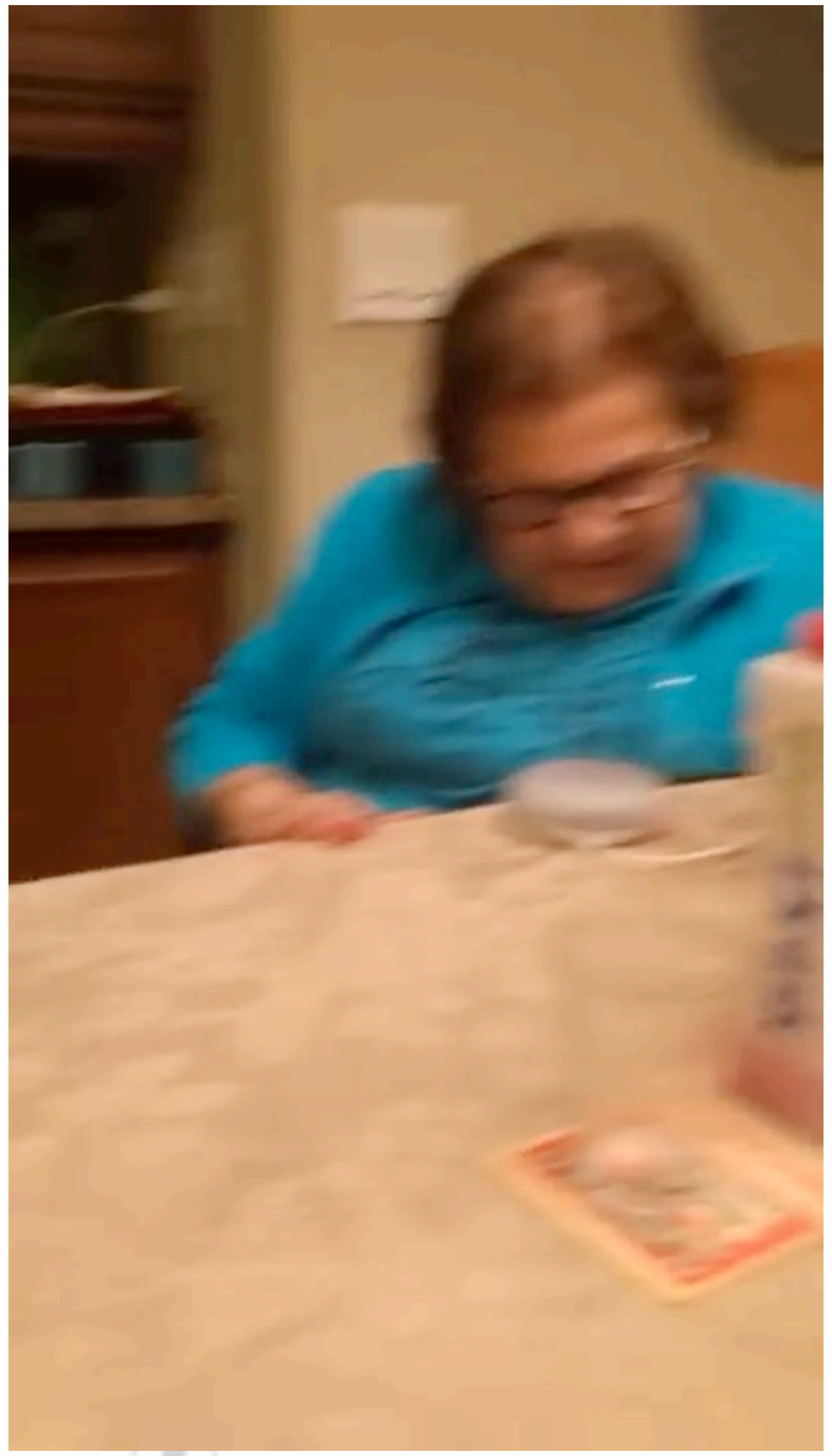
Video processing

Two-Stream RNN/CNN for Action Recognition in 3D Videos

Rui Zhao, Haider Ali, Patrick van der Smagt

German Aerospace Center, Technische Universität München

Sound processing



"Okay Google"

Italian grandmother learning to use Google home

4,731,819 views • 27 Dec 2017

Ben Actis
2.94K subscribers

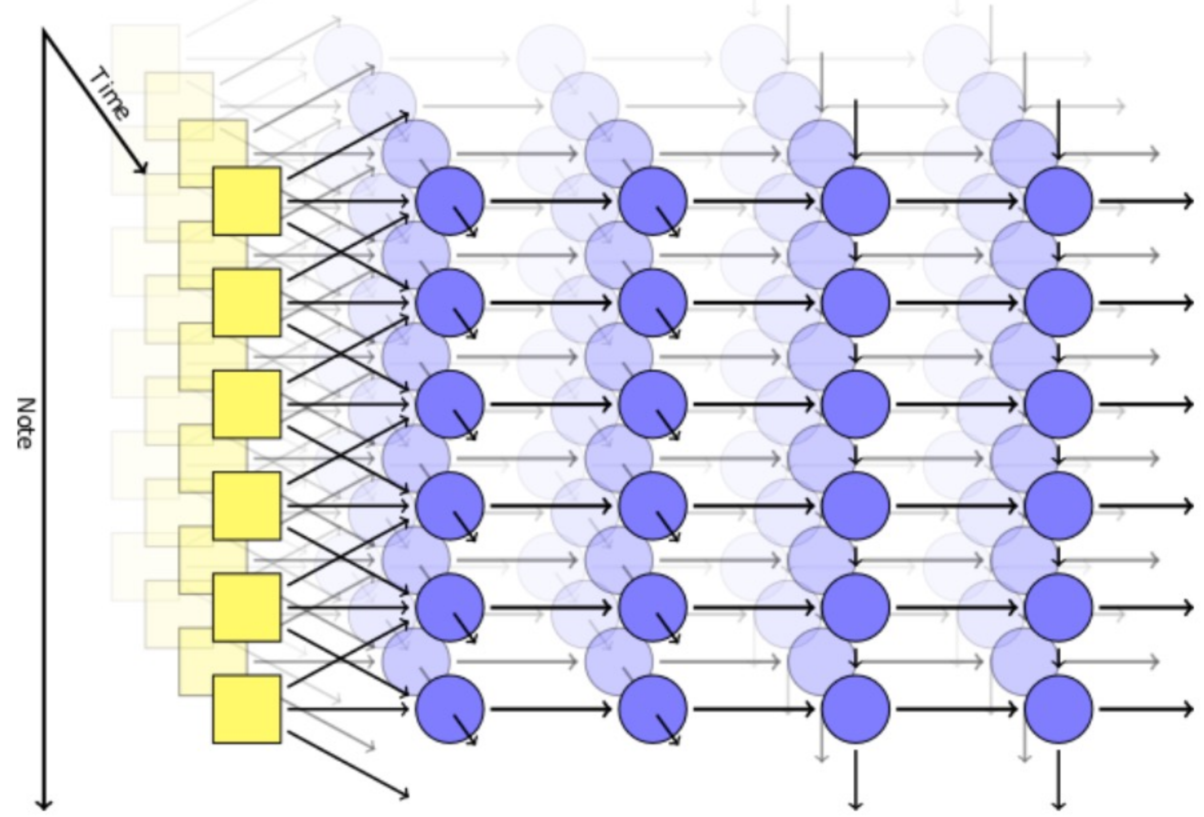
My 85 year old Italian grandmother learns to use Google home

SHOW MORE

https://www.youtube.com/watch?v=e2R0NSKtVA0

Sound generation

Composing Music With Recurrent Neural Networks



(Update: A paper based on this work has been accepted at EvoMusArt 2017! See here for more details.)

It's hard not to be blown away by the surprising power of neural networks these days. With enough training, so called "deep neural networks", with many nodes and hidden layers, can do impressively well on modeling and predicting all kinds of data. (If you don't know what I'm talking about, I recommend reading about recurrent character-level language models, Google Deep Dream, and neural Turing machines. Very cool stuff!) Now seems like as good a time as ever to experiment with what a neural network can do.

http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks/

Sentiment classification

User Reviews

The Fellowship of the Ring: Not just a Movie, but the Door to another Dimension
14 April 2006 | by bonnie91 – See all my reviews

The first part of the Lord of the Rings trilogy, the Fellowship of the Rings opened the door to a whole new world for me. I'd never read any of Tolkien's books when I saw the film for the first time at the theatre and, now that I've read them, in retrospect I think being a neophyte to the mythology made my LOTR movie experience all the more miraculous.

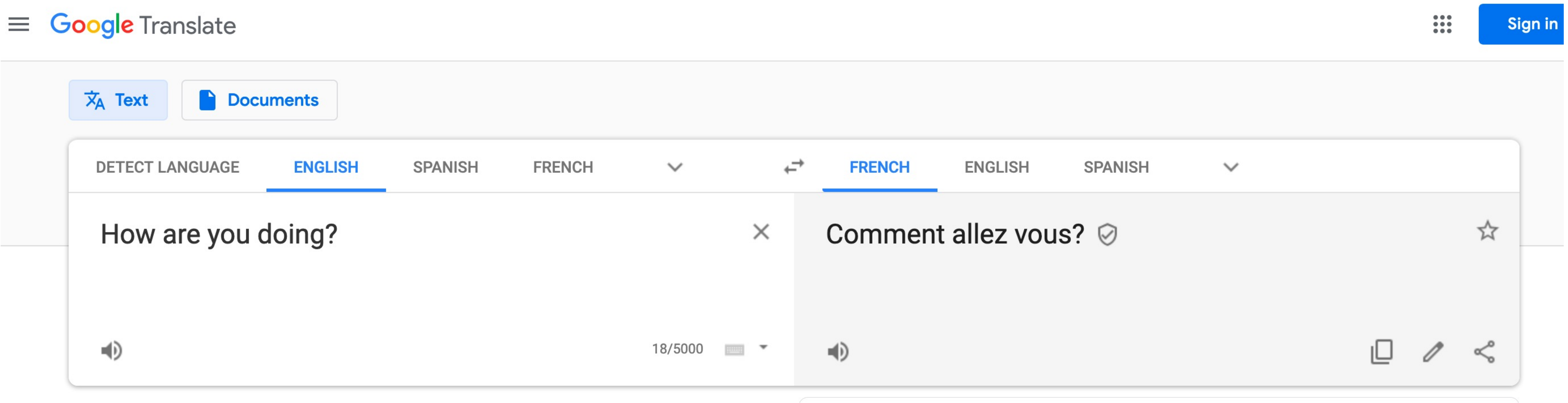
Overlong epic
salomeuk 5 June 2004

I can see why people love this movie. Perhaps I'd feel the same if it came in under 2 hours. As it is, it was far too long. I had trouble staying awake, and one of my party left halfway through as she was so uncomfortable sitting so long in the cinema. That's something I've never know happen before or since!

I can't really comment on the movie itself too much, as I'm not sure I've actually seen it all....sat through it, yes. But much of it didn't register after the first half hour.

https://www.imdb.com/title/tt0120737/?ref_=nv_sr_1?ref_=nv_sr_1

Machine translation



Natural Language Processing

Natural Language Processing (NLP)

"We will take Natural Language Processing —
or NLP for short –in a wide sense to **cover any kind of computer manipulation of natural language**. At one extreme, it could be as simple as counting word frequencies to compare different writing styles. At the other extreme, NLP involves "understanding" complete human utterances, at least to the extent of being able to give useful responses to them."

Bird, Steven, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

Natural Language Processing (NLP)

- It is a branch of Artificial Intelligence
- Large amounts of data is textual data
  - NLP helps computers and humans communicate
- An important percentage of textual data is unstructured/unannotated
- Not all NLP tasks are solved by Deep Learning!

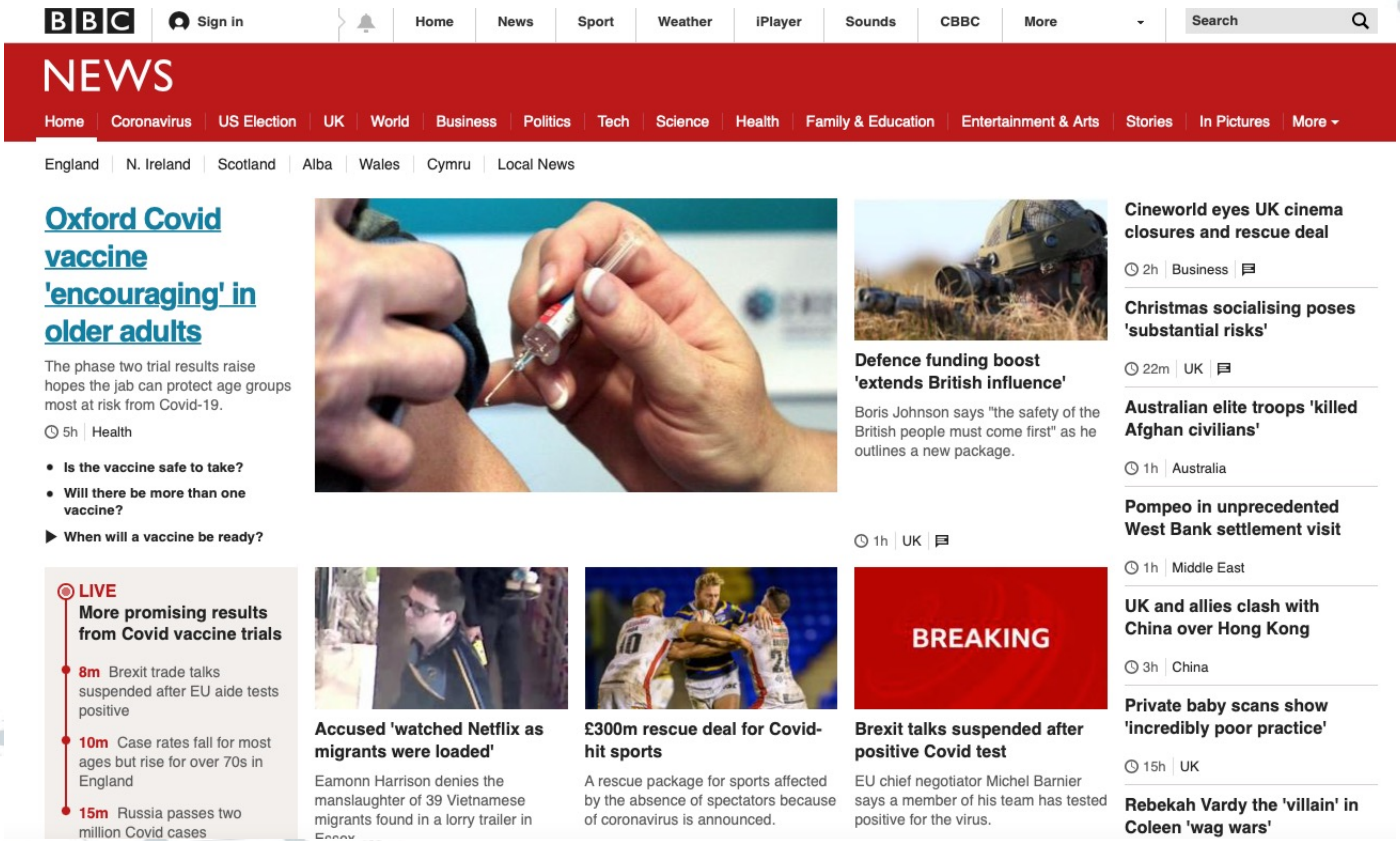
Application: Named Entity Recognition

◎Recognise named entities in text into predefined categories, like persons names, companies names, locations, time etc.

- Can help in NLP to automatically search for specific persons/locations/organisations

Application: Named Entity Recognition

- Recognise named entities in text



Application: Named Entity Recognition

- Let’s consider the simplified problem of NER:

“Given a text, find the people’s names”

aka classify each word in a sentence as being a name (label 1) or not a name (label 0)

Application: Named Entity Recognition

- “My name is Teddy”.

  0 0 0 1

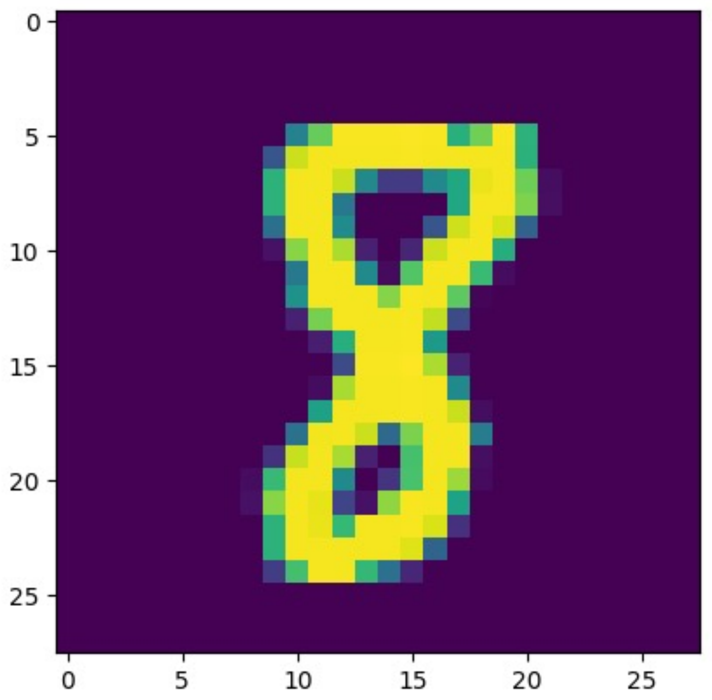
- Claire said: “Teddy bears are my favourite toys.”

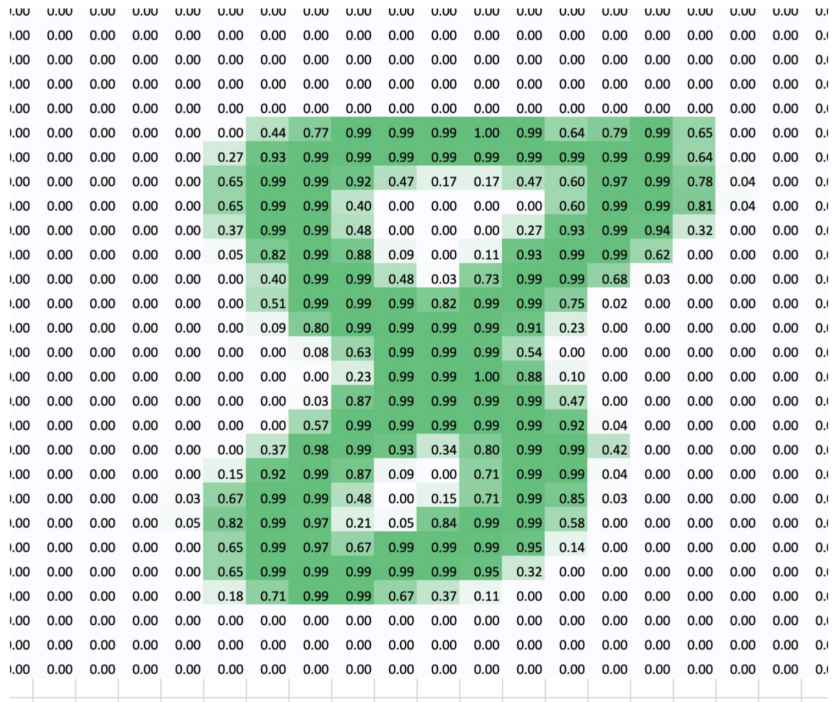
  1 0 0 0 0 0 0 0

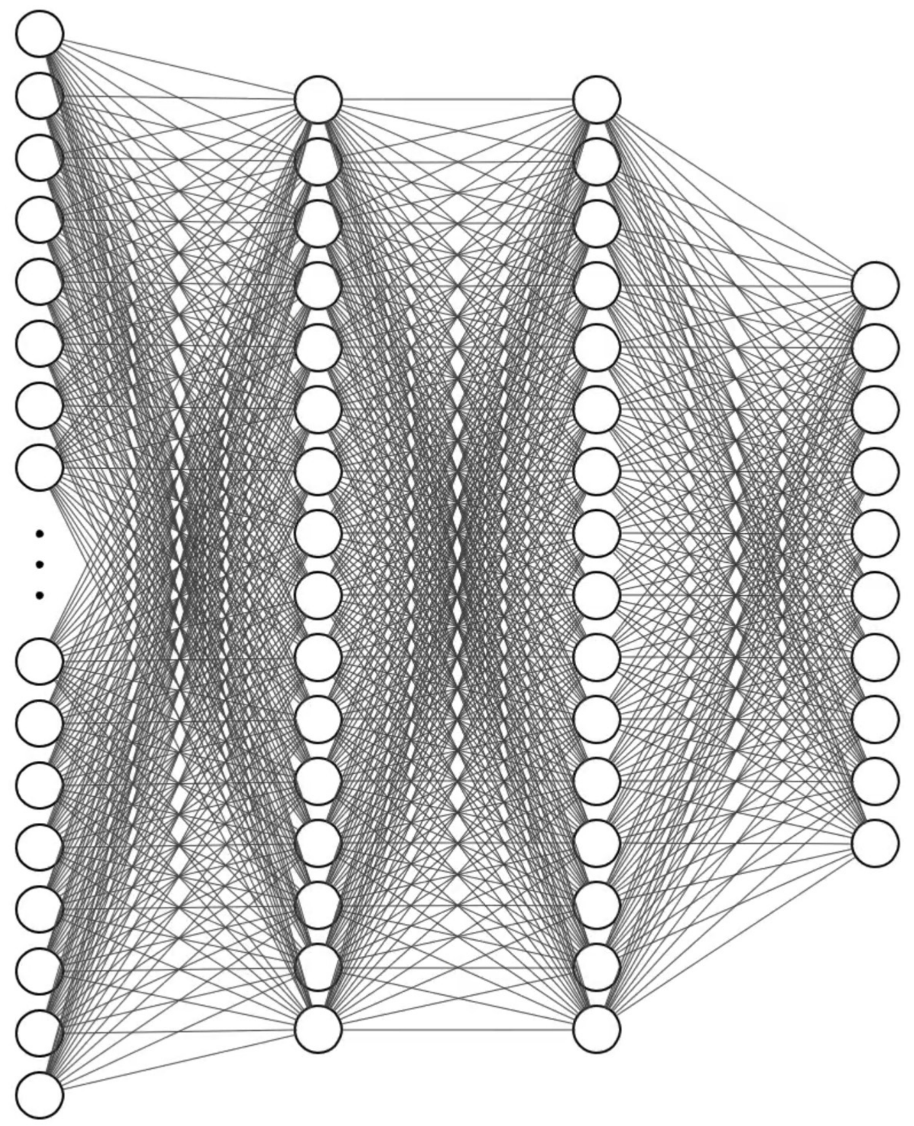
1 - Name

0 - Not Name

Text and Neural Networks



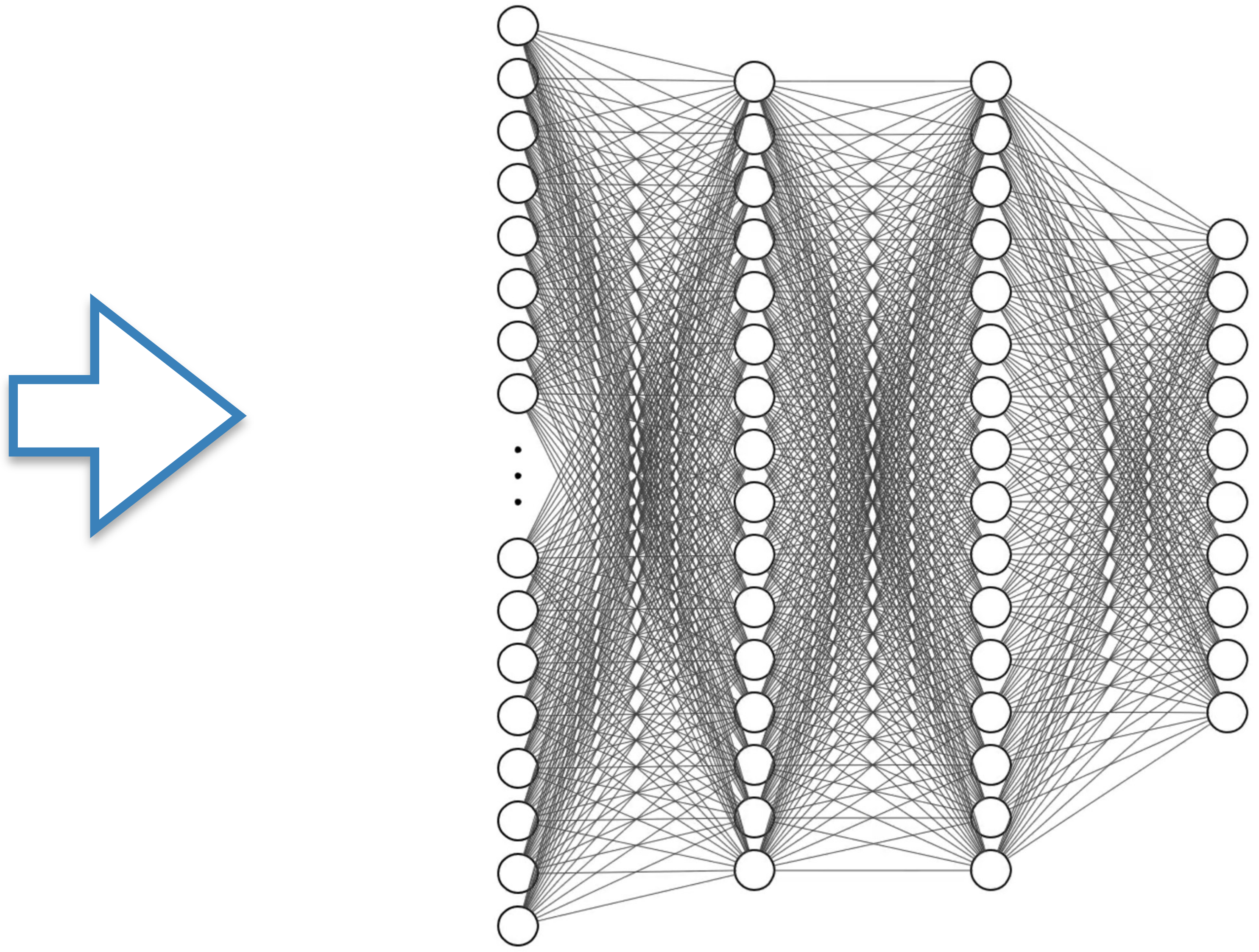




An image is just a collection of numbers (the value of each pixel)

Text and Neural Networks

“My name is Teddy”.



How do we input text to a neural network?

From text… to numeric data

From text… to numeric data

One-hot encoding

One-hot encoding

- Imagine you have a dictionary of 10000 words, sorted alphabetically

A	→	1
About	→	2
….		
Be	→	105
Because	→	106
…		
Could	→	256
…..		…..
What	→	9996
When	→	9997
Year	→	9998
Zoo	→	9999
Zyzzyva	→	10000

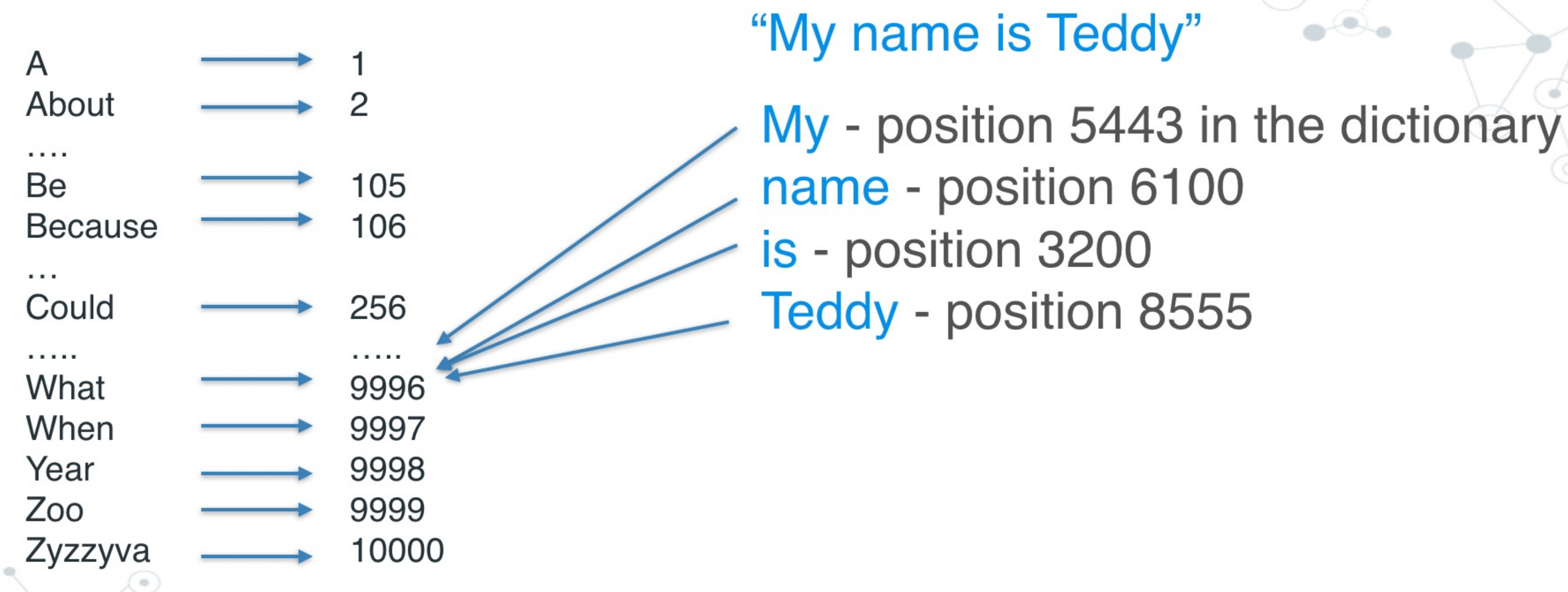
One-hot encoding

A	1
About	2
….	
Be	105
Because	106
…	
Could	256
…..	…..
What	9996
When	9997
Year	9998
Zoo	9999
Zyzzyva	10000

"My name is Teddy"

My - position 5443 in the dictionary
name - position 6100
is - position 3200
Teddy - position 8555

One-hot encoding



- Represent each word as a vector of zeros and ones
- The vector size will be the size (N) of the dictionary (in our example N=10000)
- Commercial NLP systems work with dictionary size of over 100000 words

One-hot encoding

A	1
About	2
….	
Be	105
Because	106
…	
Could	256
…..	…..
What	9996
When	9997
Year	9998
Zoo	9999
Zyzzyva	10000

"My name is Teddy"

My - position 5443 in the dictionary
name - position 6100
is - position 3200
Teddy - position 8555

$$\text{My} \Rightarrow \left[\begin{array}{c}0\\0\\\vdots\\0\\1\\0\\\vdots\\0\end{array}\right]\begin{array}{l}\\ \\ \\ \\ \leftarrow 5443\\ \\ \\ \\ \end{array}$$

$$\text{name} \Rightarrow \left[\begin{array}{c}0\\0\\\vdots\\0\\\vdots\\1\\\vdots\\0\end{array}\right]\begin{array}{l}\\ \\ \\ \\ \\ \leftarrow 6100\\ \\ \\ \end{array}$$

One-hot encoding: Disadvantages

A	1
About	2
….	
Be	105
Because	106
…	
Could	256
…..	…..
What	9996
When	9997
Year	9998
Zoo	9999
Zyzzyva	10000

"My name is Teddy"

My - position 5443 in the dictionary
name - position 6100
is - position 3200
Teddy - position 8555

- Dimensionality
- Words are embedded in isolation from the other words.

From text… to numeric data

TF-IDF

TF-IDF (term frequency–inverse document frequency)

- A statistic that is used to compute how **important a word is to a document <u>given a collection of documents</u>**

TF-IDF (term frequency–inverse document frequency)

◎TF - Term frequency:

$$TF = \frac{(Number of times term t appears in the document)}{(Total number of terms in the document)}$$

◎IDF - Inverse document frequency

$$IDF = \log \frac{(Total number of documents)}{(Total number of documents with term t)}$$

◎TF-IDF

$$TFIDF = TF \cdot IDF$$

TF-IDF (term frequency–inverse document frequency)

◎TF - Term frequency:

$$TF = \frac{(Number of times term t appears in a document)}{(Total number of terms in the document)}$$

Document1: “My name is Teddy, Teddy Boss”.

Document2: Claire said: “Teddy bears are my favourite toys.”

TF-IDF (term frequency–inverse document frequency)

Document1: “My name is Teddy, Teddy Boss”.

Document2: Claire said: “Teddy bears are my favourite toys.”

	are	claire	boss	bears	favourite	is	my	name	said	teddy	toys
Document1:	0	0	1	0	0	1	1	1	0	2	0
Document2:	1	1	0	1	1	0	1	0	1	1	1

TF-IDF (term frequency–inverse document frequency): How to compute TF

		are	claire	boss	bears	favourite	is	my	name	said	teddy	toys
"My name is Teddy, Teddy Boss".	Document1:	0	0	1	0	0	1	1	1	0	2	0
Claire said: "Teddy bears are my favourite toys."	Document2:	1	1	0	1	1	0	1	0	1	1	1

tf("is",Document1) = 1/6 = 0.166
tf("teddy",Document1) = 2/6 = 0.333
tf("teddy", Document2) = 1/8 = 0.125

$$TF = \frac{(Number of times term t appears in a document)}{(Total number of terms in the document)}$$

TF-IDF (term frequency–inverse document frequency): How to compute IDF

		are	claire	boss	bears	favourite	is	my	name	said	teddy	toys
“My name is Teddy, Teddy Boss”.	Document1:	0	0	1	0	0	1	1	1	0	2	0
Claire said: “Teddy bears are my favourite toys.”	Document2:	1	1	0	1	1	0	1	0	1	1	1

idf(“is”) = log(2/1) = 0.301

idf(“teddy”) = log(2/2) = 0

$$IDF = \log \frac{(Total number of documents)}{(Total number of documents with term t)}$$

TF-IDF (term frequency–inverse document frequency): How to compute TF-IDF

Sentence	Document	are	claire	boss	bears	favourite	is	my	name	said	teddy	toys
"My name is Teddy, Teddy Boss".	Document1:	0	0	1	0	0	1	1	1	0	2	0
Claire said: "Teddy bears are my favourite toys."	Document2:	1	1	0	1	1	0	1	0	1	1	1

$$tfidf(\text{“is”}, Document1) = tf(\text{“is”}, Document1) * idf(\text{“is”}) = 0.166 * 0.301 = 0.0499$$

$$tfidf(\text{“teddy”}, Document2) = tf(\text{“teddy”}, Document2) * idf(\text{“teddy”}) = 0.125 * 0 = 0$$

TF-IDF (term frequency–inverse document frequency): How to compute TF-IDF

TF

	are	bears	boss	claire	favourite	is	my	name	said	teddy	toys
Document1	0	0	0.167	0	0	0.167	0.167	0.167	0	0.333	0
Document2	0.125	0.125	0	0.125	0.125	0	0.125	0.000	0.125	0.125	0.125

IDF

are	bears	boss	claire	favou rite	is	my	name	said	teddy	toys
0.301	0.301	0.301	0.301	0.301	0.301	0.301	0.301	0.301	0	0.301

TF-IDF

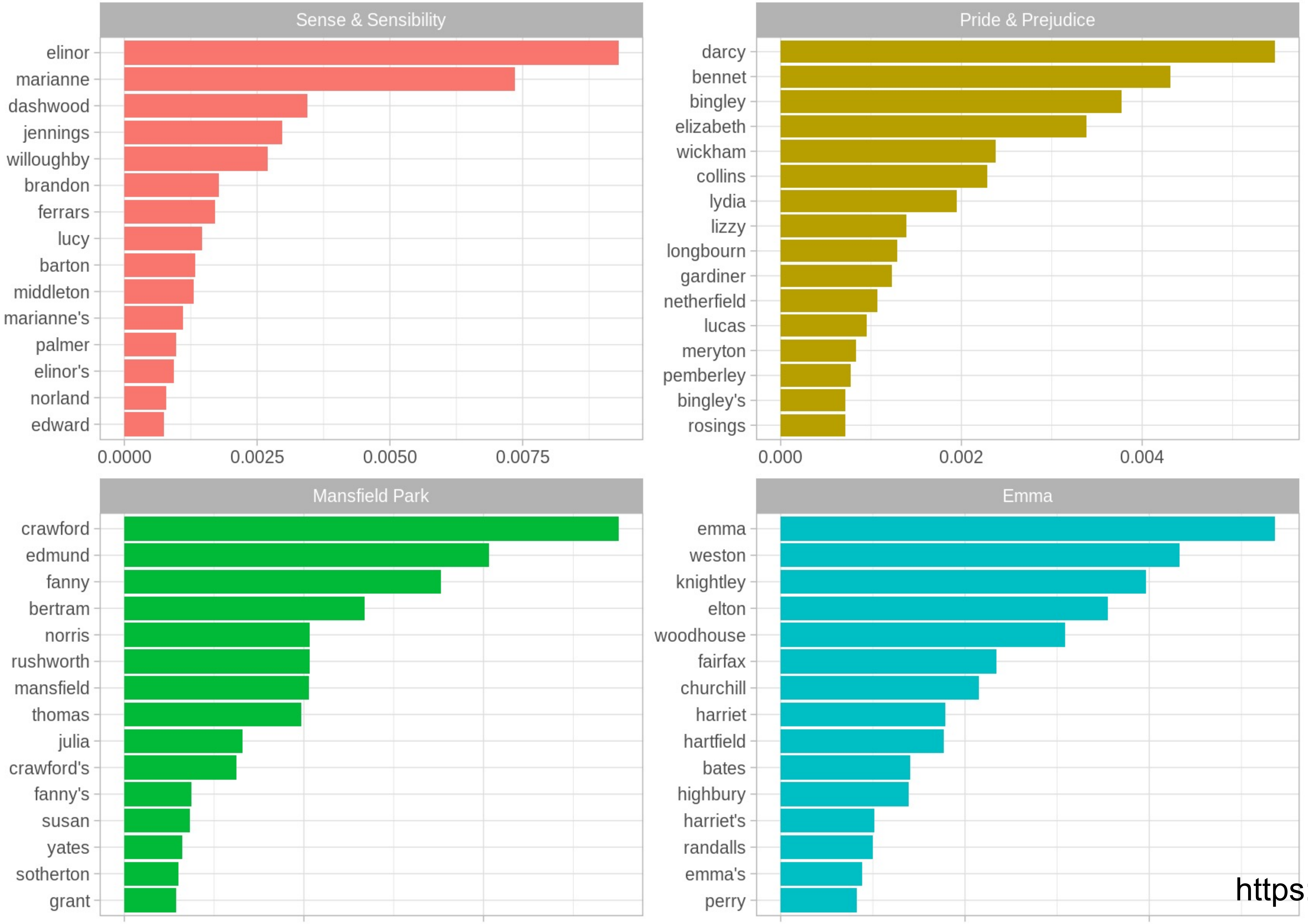
	Document1	Document2
are	0	0.0376
bears	0	0.0376
boss	0.0502	0
claire	0	0.0376
favourite	0	0.0376
is	0.0502	0
my	0.0502	0.0376
name	0.0502	0.000
said	0	0.0376
teddy	0	0
toys	0	0.0376

TF-IDF (term frequency–inverse document frequency):

- TF-IDF is usually computed across **many** documents
- It is efficient and straightforward to compute
- It can be used:
  - In search engines to retrieve partial matches
  - Summarise documents
  - Automatic tag extraction etc.

TF-IDF (term frequency–inverse document frequency): Application example

Highest tf-idf words in each of Jane Austen's Novels



https://www.tidytextmining.com/tfidf.html

TF-IDF (term frequency–inverse document frequency):

◎Disadvantages:

- It assumes that word frequency is relevant for document similarity
- It might be slow for large vocabularies (but it can be parallelised)
- It does not take into consideration semantic similarities

From text… to numeric data

Word2Vec

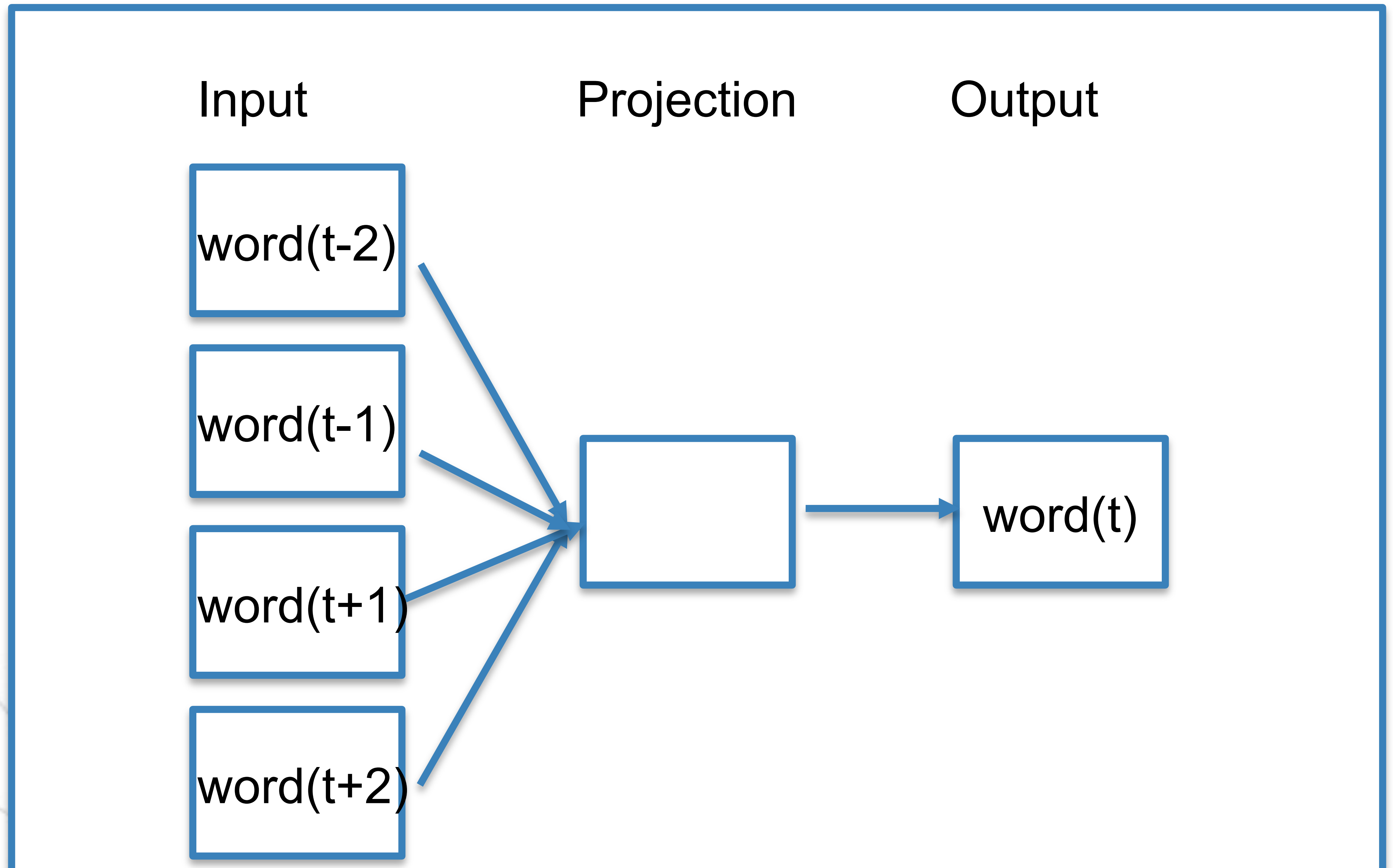
Word embedding: word2vec

- Word2vec is a model used to learn vector representation of the words
- It uses a simple 3 layer neural network (input, hidden layer, output layer)
- The size of the vector representation for each word is usually between 100 to 1000
- The idea is to have a vector representation where similar words will have similar vectors

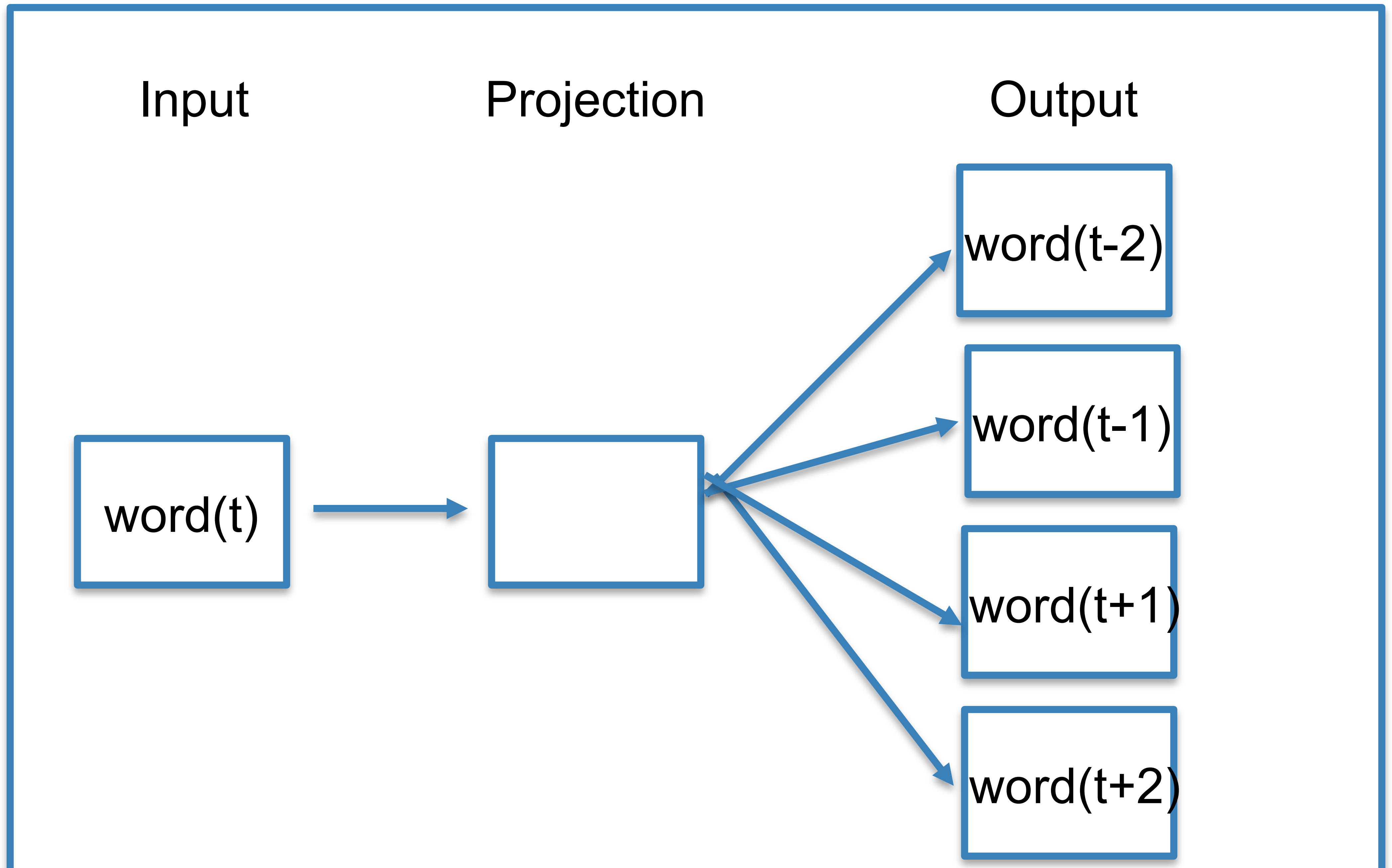
Word embedding: word2vec

◎There are two approaches to compute the word embeddings:

- CBOW: continuous bag of words
- Skip-gram model



CBOW

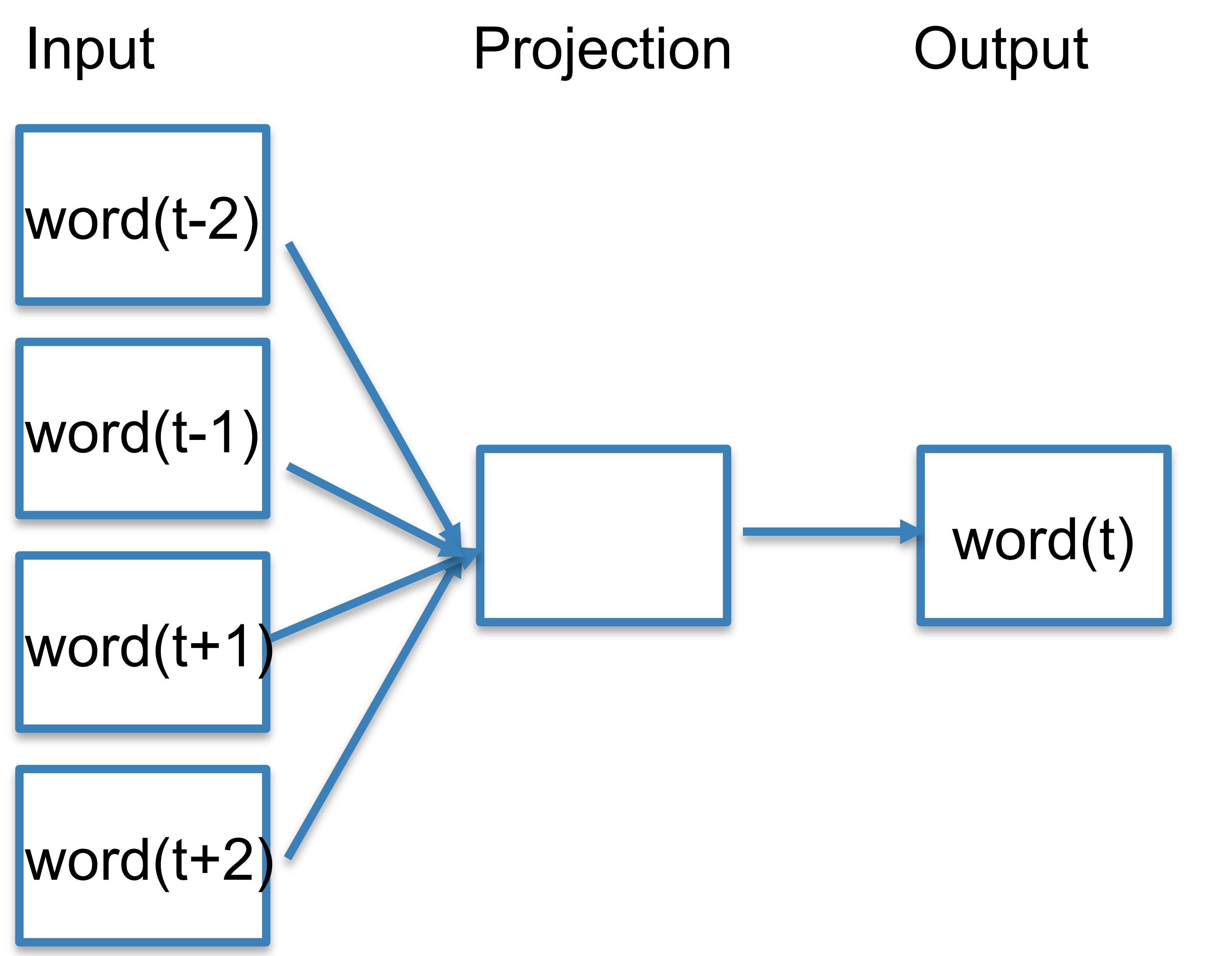


Skip-gram

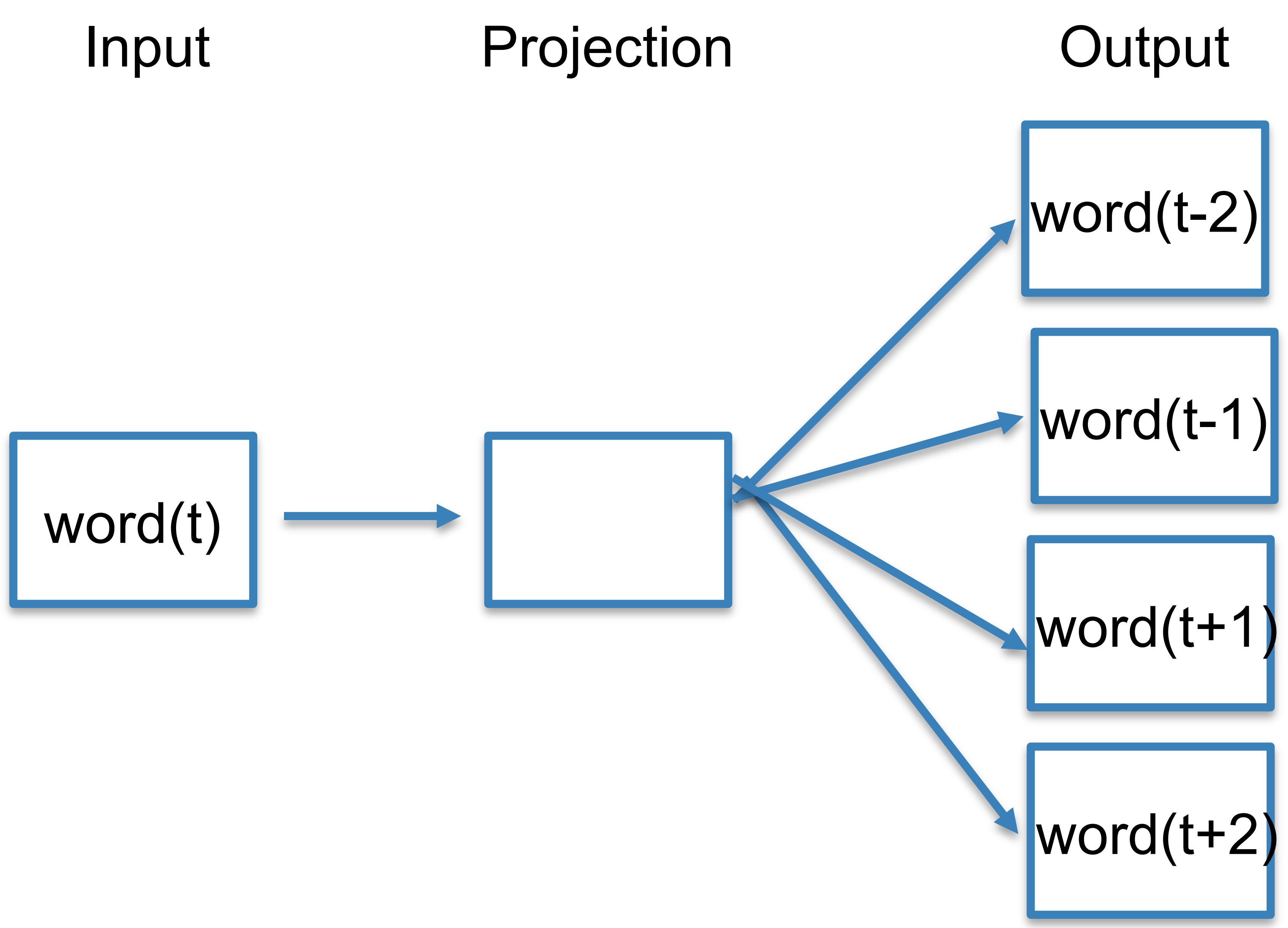
Word embedding: word2vec

◎There are two approaches to compute the word embeddings:

- CBOW: continuous bag of words
- Skip-gram model



CBOW



Skip-gram

Word embedding: word2vec

◎There are two approaches to compute the word embeddings:

- CBOW: continuous bag of words
- Skip-gram model

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

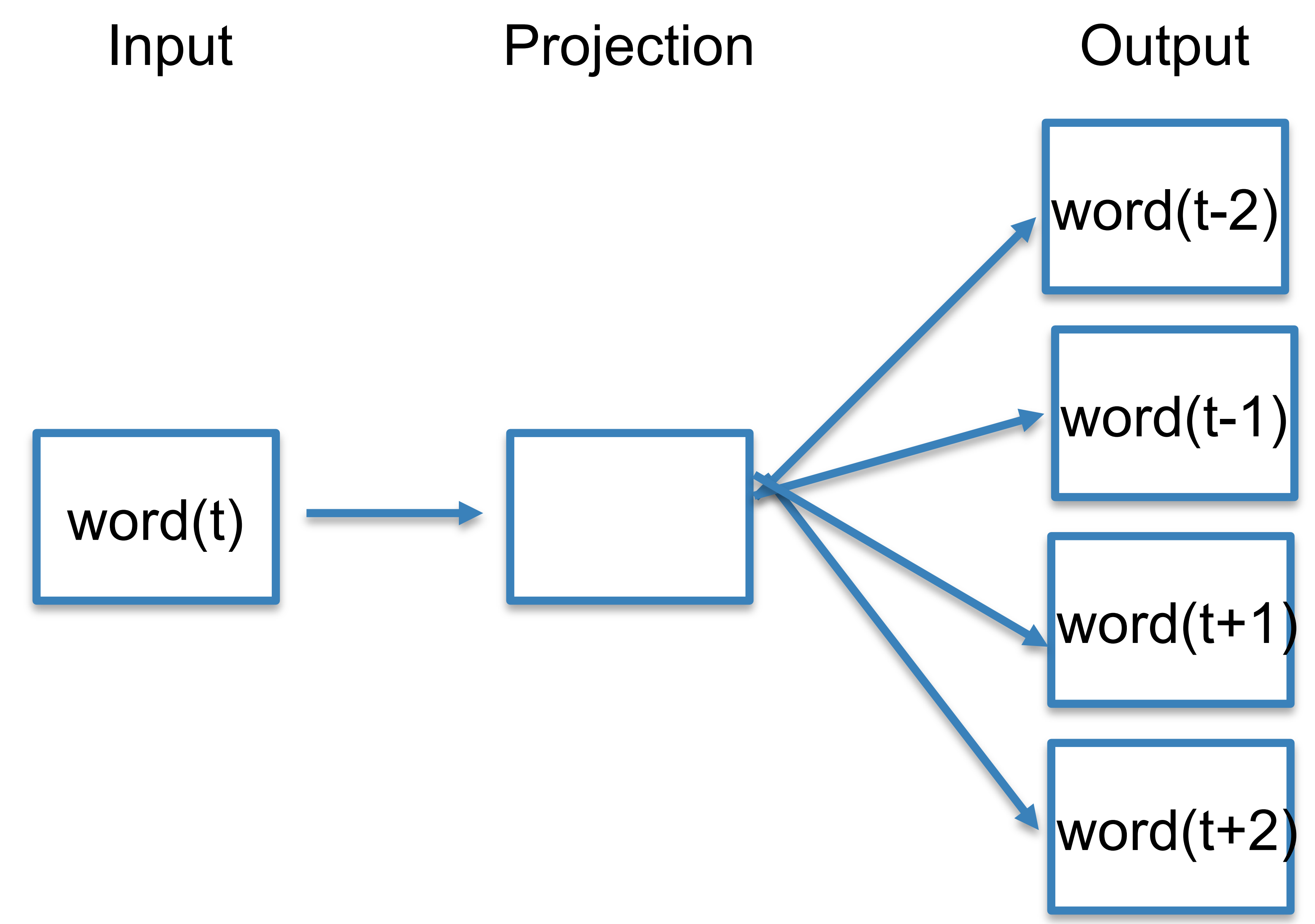
Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

Word embedding: word2vec
How training works (skip-gram)

- Create a 3 layer neural network
- The input is a word
- The output should be another word
- The task is to learn to predict a word from its neighbourhood



Skip-gram

Word embedding: word2vec
How training works (skip-gram)

- Given the text…
  - “The five boxing wizards jump quickly."
- …and a window of size 2
- Generate the training set:
  - “The five boxing wizards jump quickly."

Word embedding: word2vec
How training works (skip-gram)

- “The five boxing wizards jump quickly."

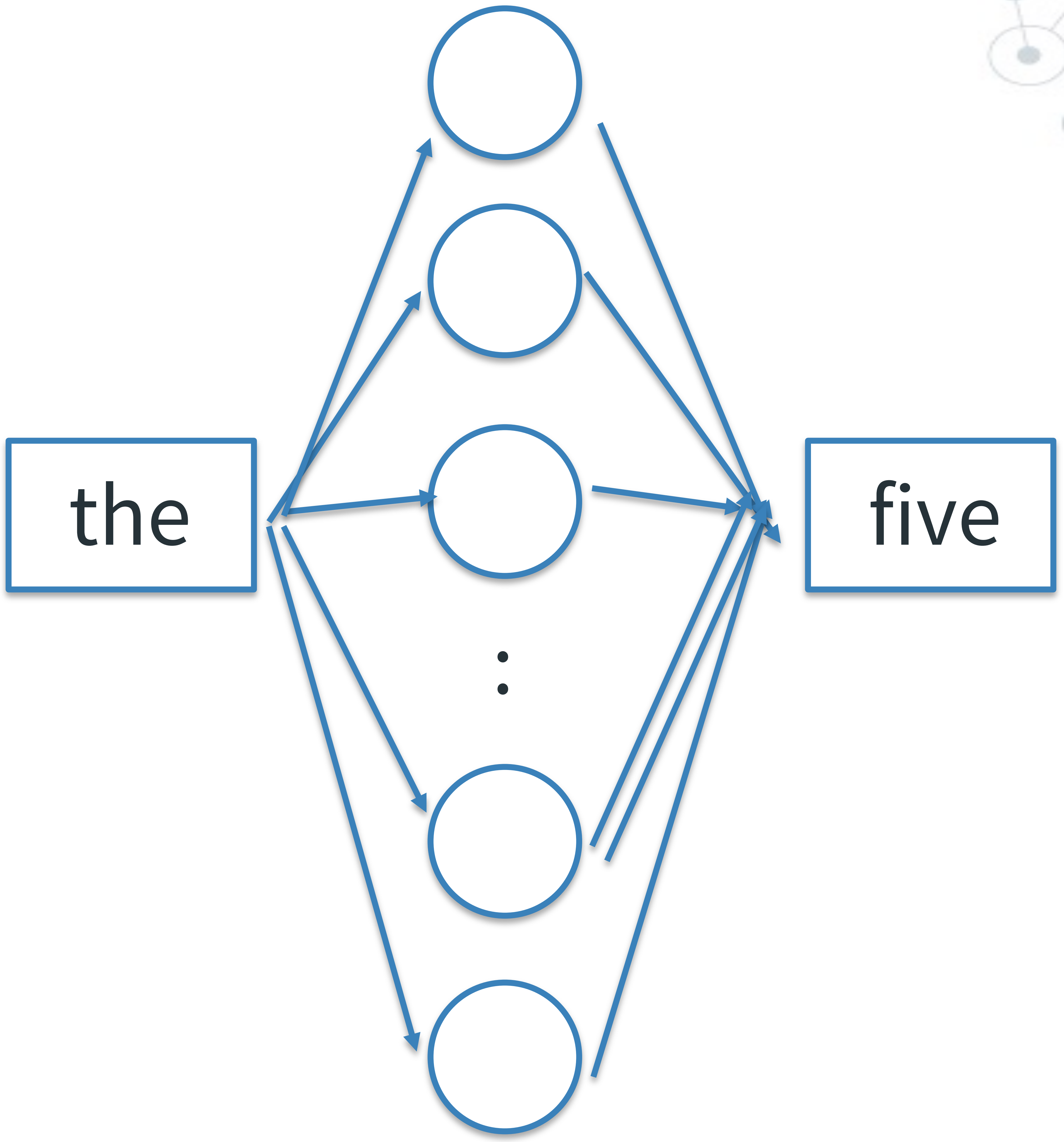
Training set:

The five boxing wizards jump quickly.	(the, five); (the, boxing)
The five boxing wizards jump quickly.	(five, the); (five, boxing); (five, wizards)
The five boxing wizards jump quickly.	(boxing, the); (boxing, five); (boxing, wizards), (boxing, quickly)
The five boxing wizards jump quickly.	(wizards, five); (wizards, boxing); (wizards, jump), (wizards, jump)

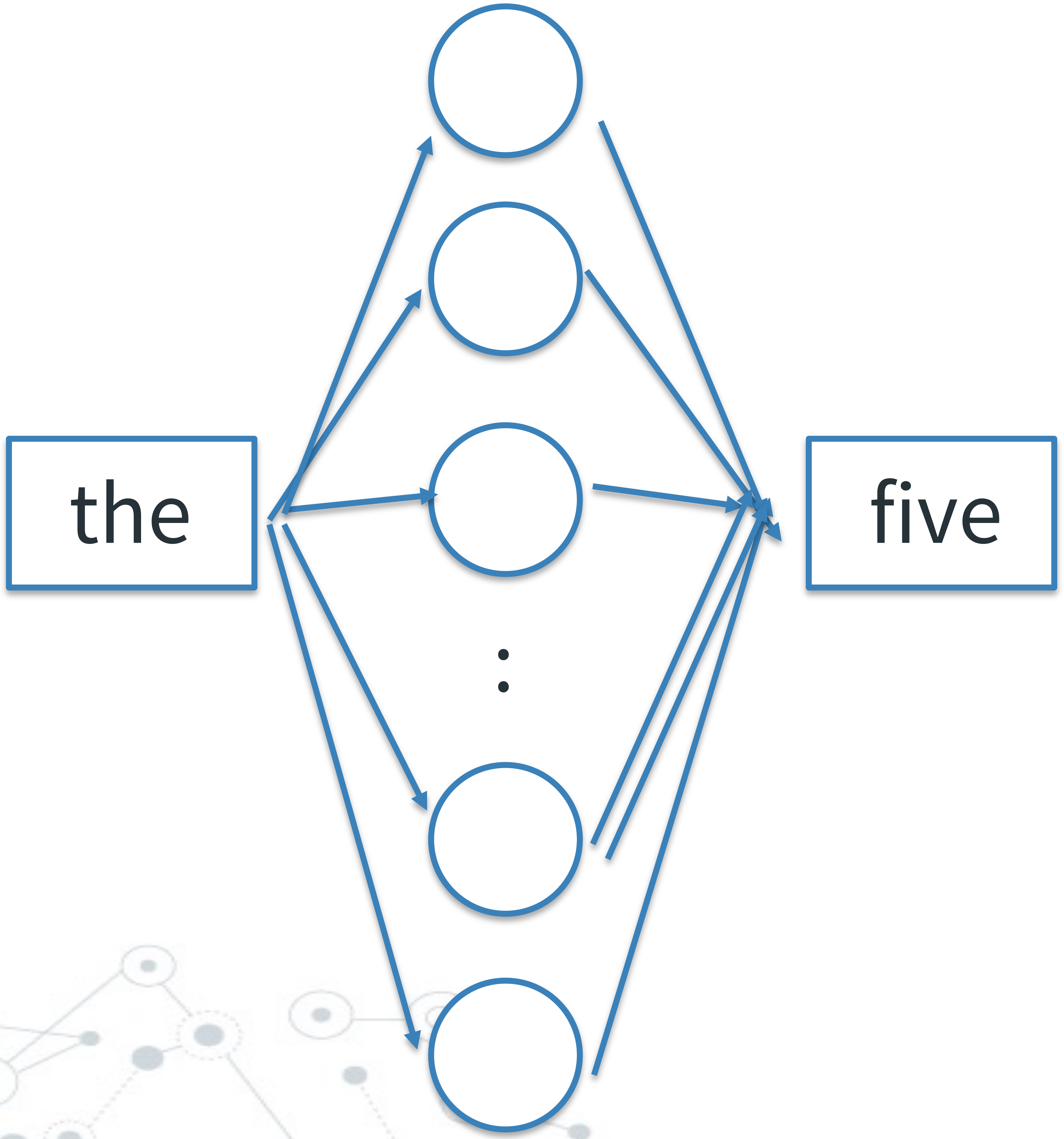
Word embedding: word2vec
How training works

Training set:

(the, five); (the, boxing); (five, the); (five, boxing); (five, wizards); (boxing, the); (boxing, five); (boxing, wizards), (boxing, quickly); (wizards, five); (wizards, boxing); (wizards, jump), (wizards, jump)

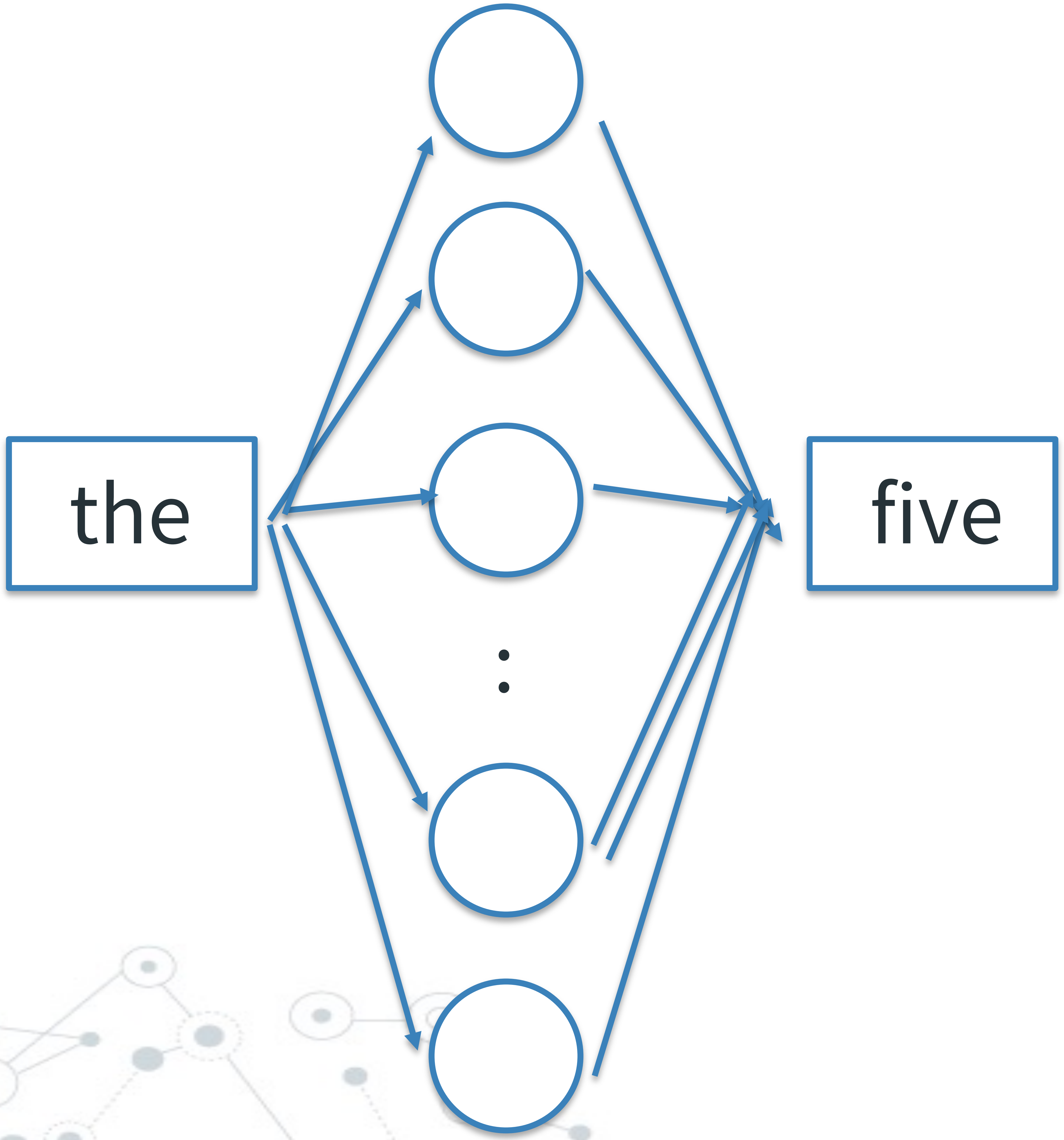


Word embedding: word2vec
How training works



The input are not the actual words, but a one-hot vector representation!

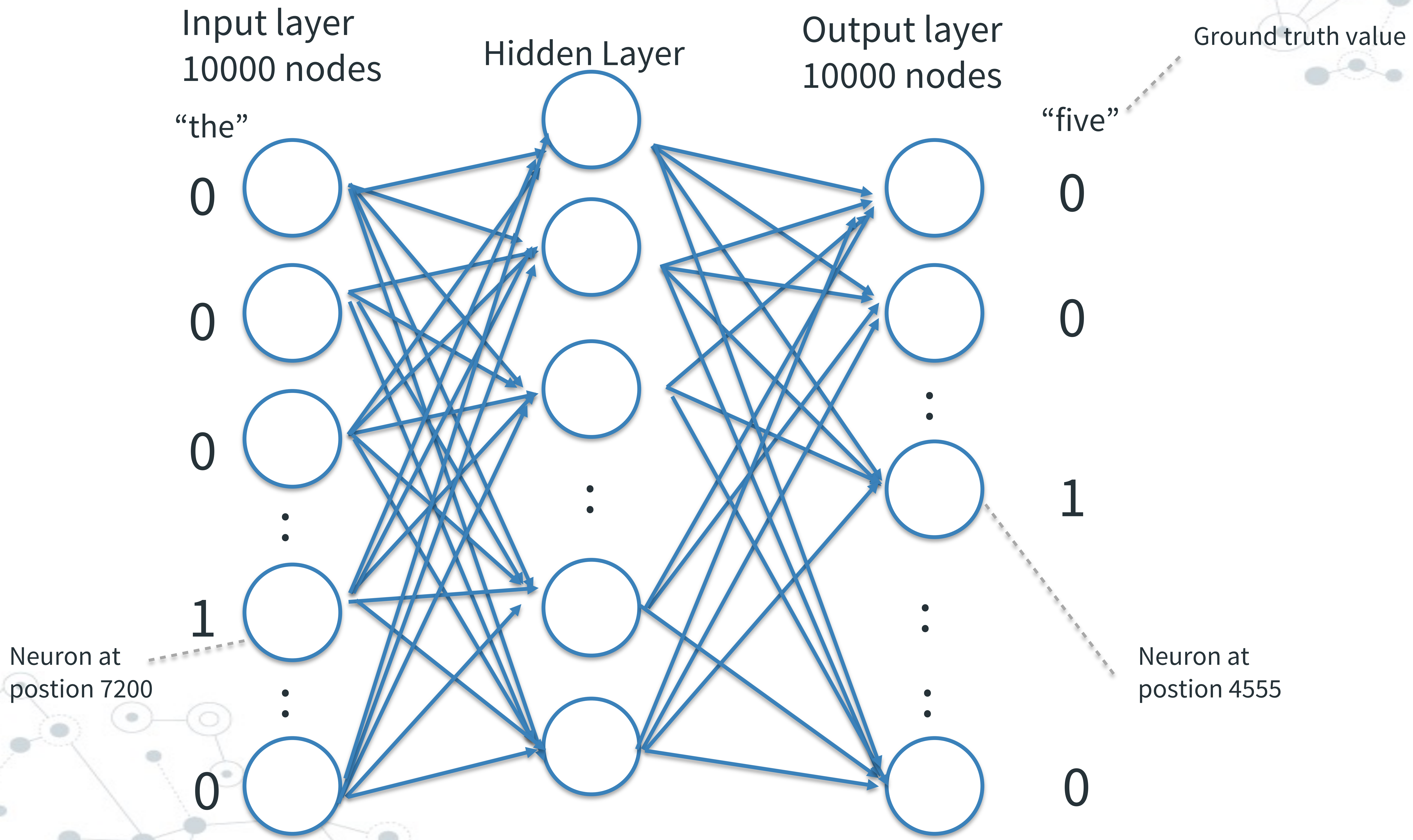
Word embedding: word2vec
How training works



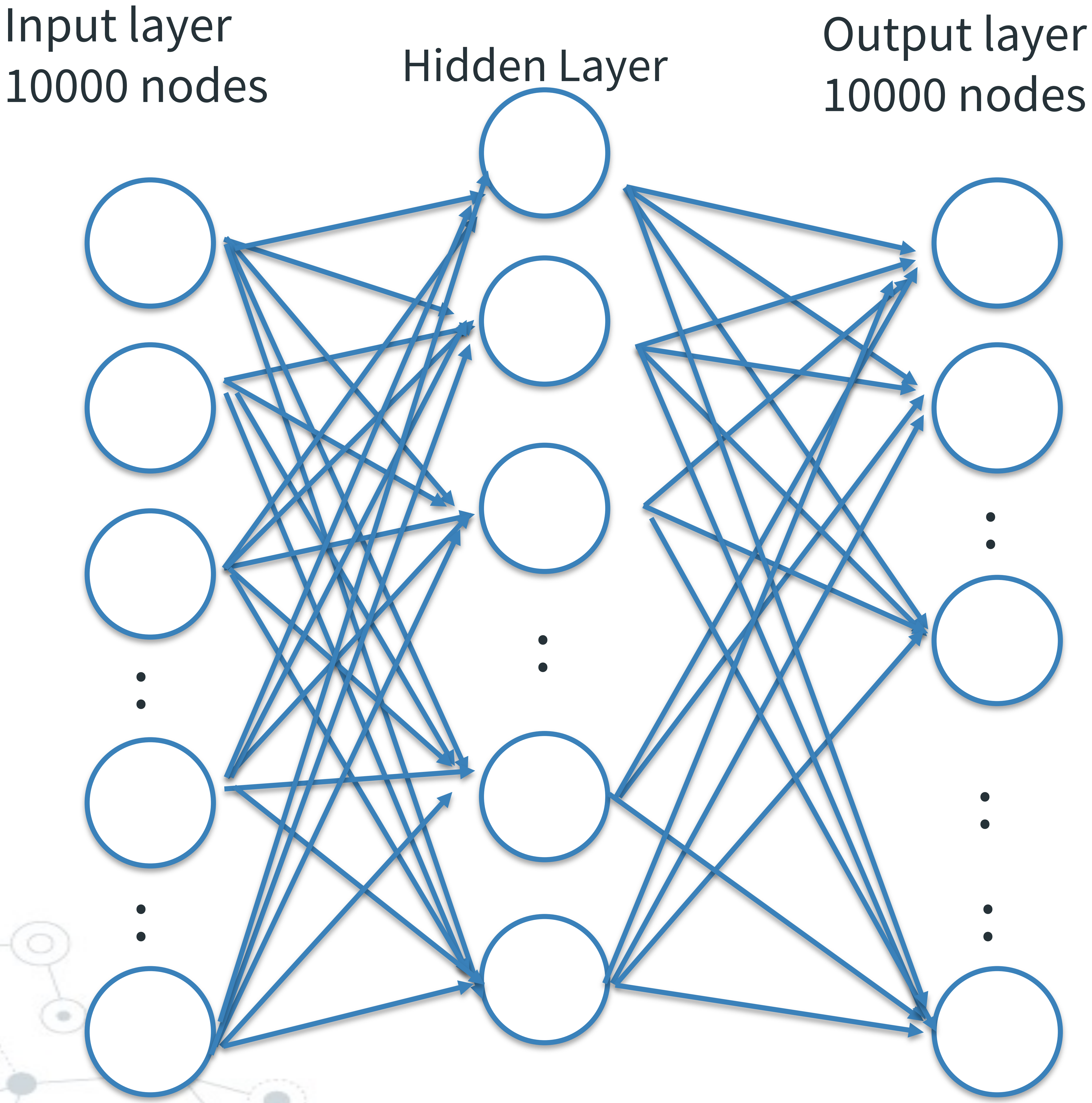
Let's assume we have a dictionary of 10000 elements
The - word at position 7200
Five - word at position 4555

We use the one-hot vector representation for input and output!

Word embedding: word2vec
How training works



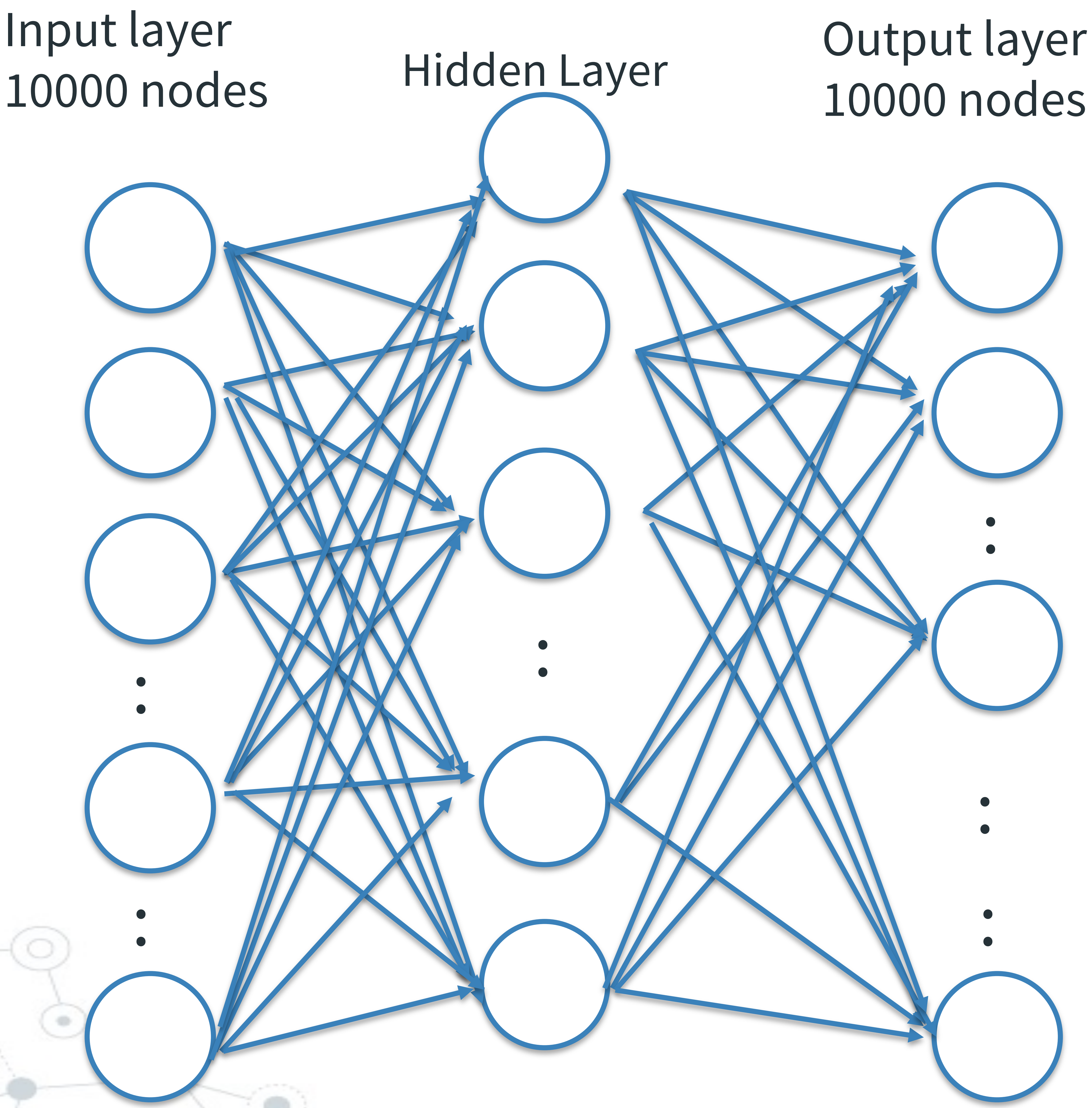
Word embedding: word2vec
How training works



When training we pass all the word pairs from the training set through the network

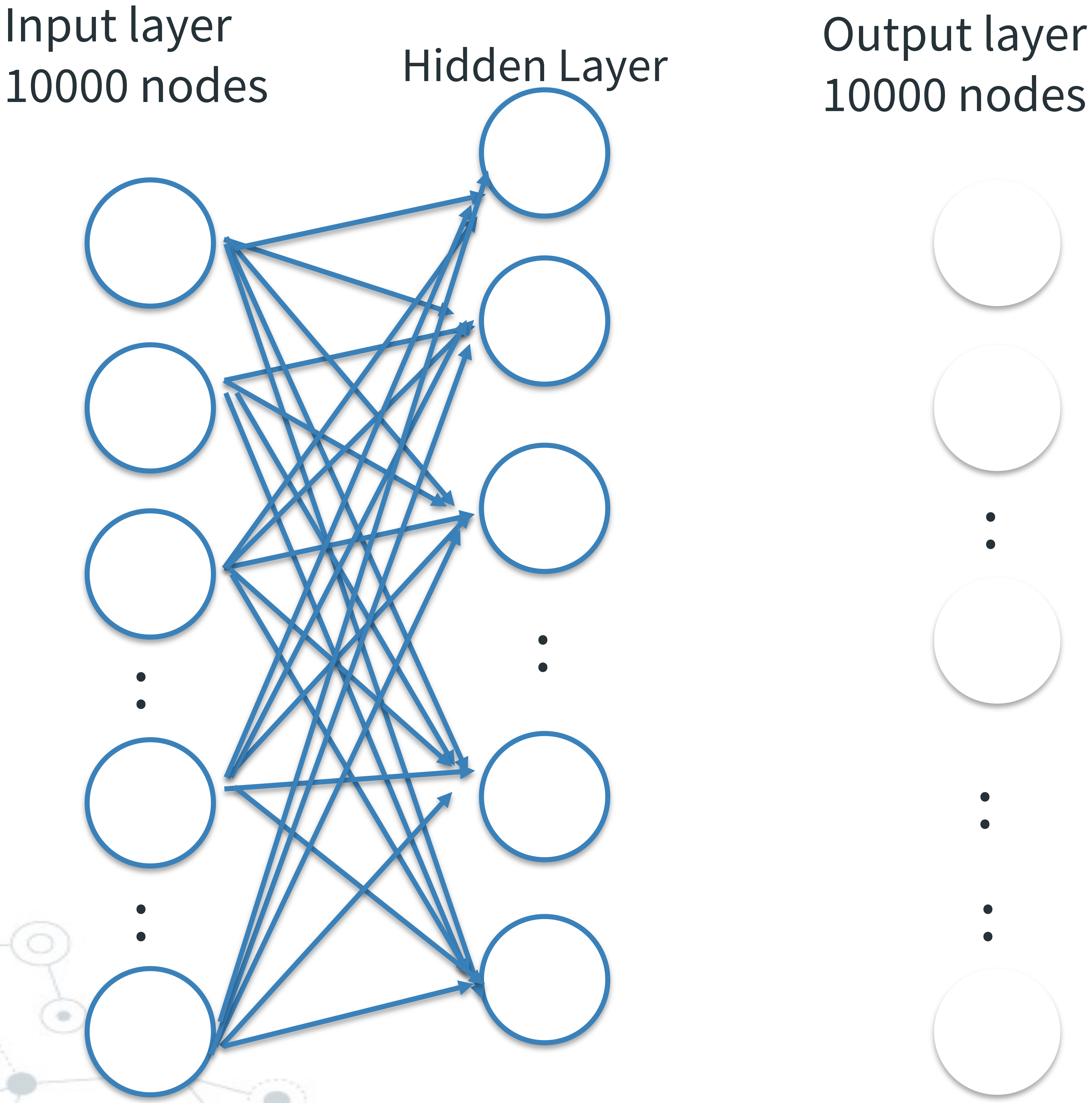
The network is trained using back-propagation

Word embedding: word2vec
How training works

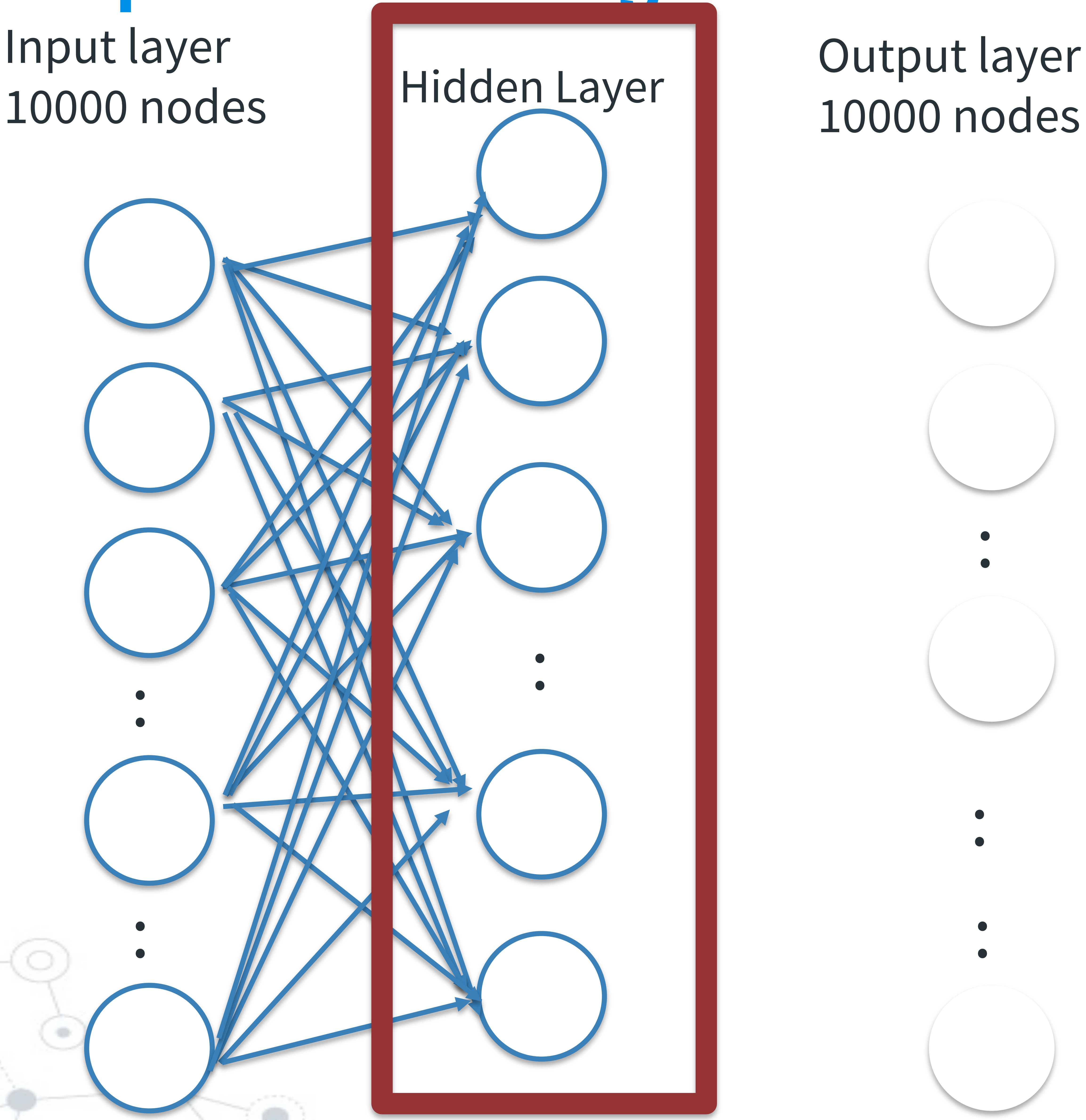


The output layer will contain the probabilities that the words are in the neighbourhood of the input word

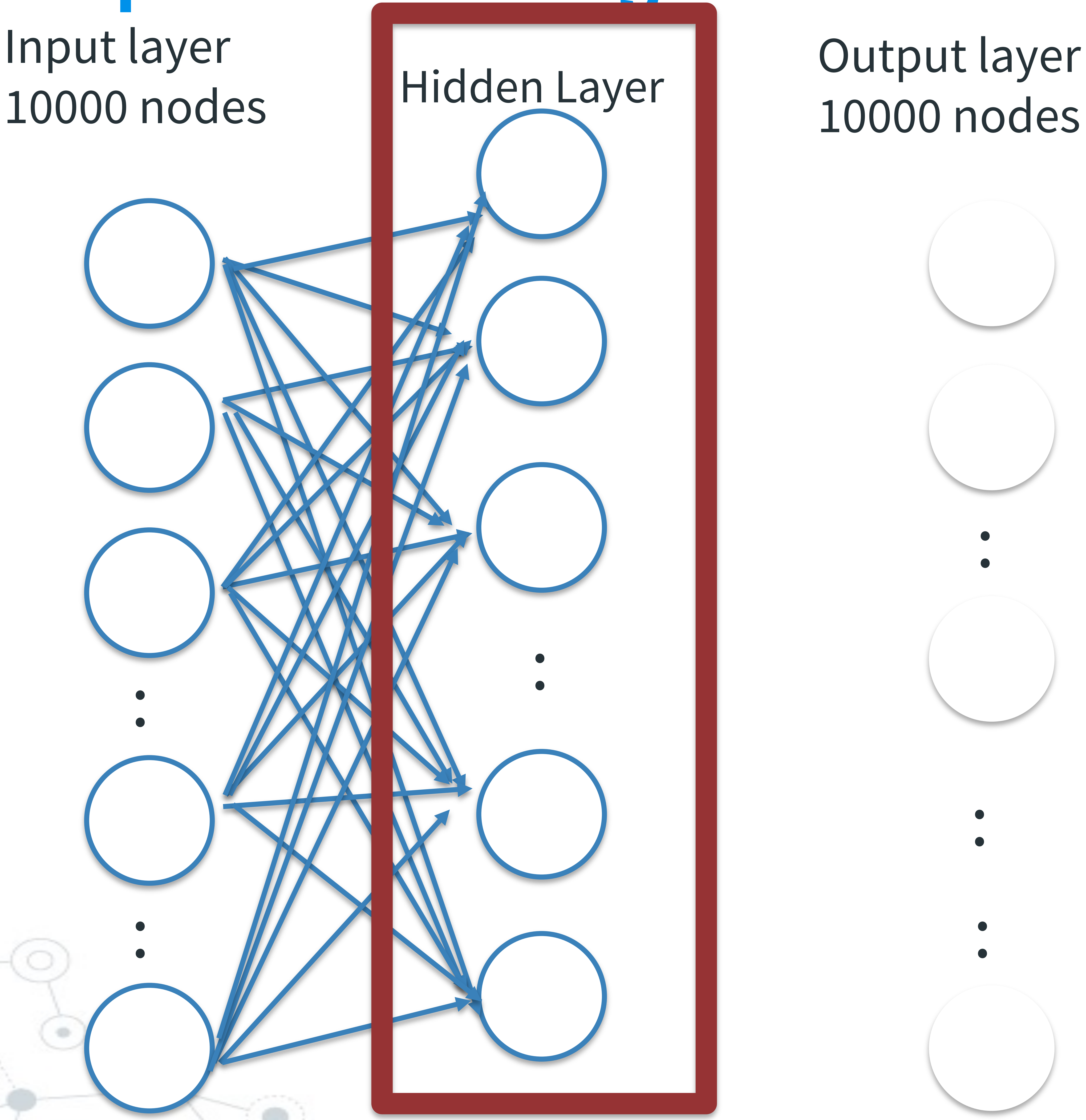
Word embedding: word2vec
What we keep after training



Word embedding: word2vec
What we keep after training



Word embedding: word2vec
What we keep after training



After training, the hidden layer is our word-vector representation!

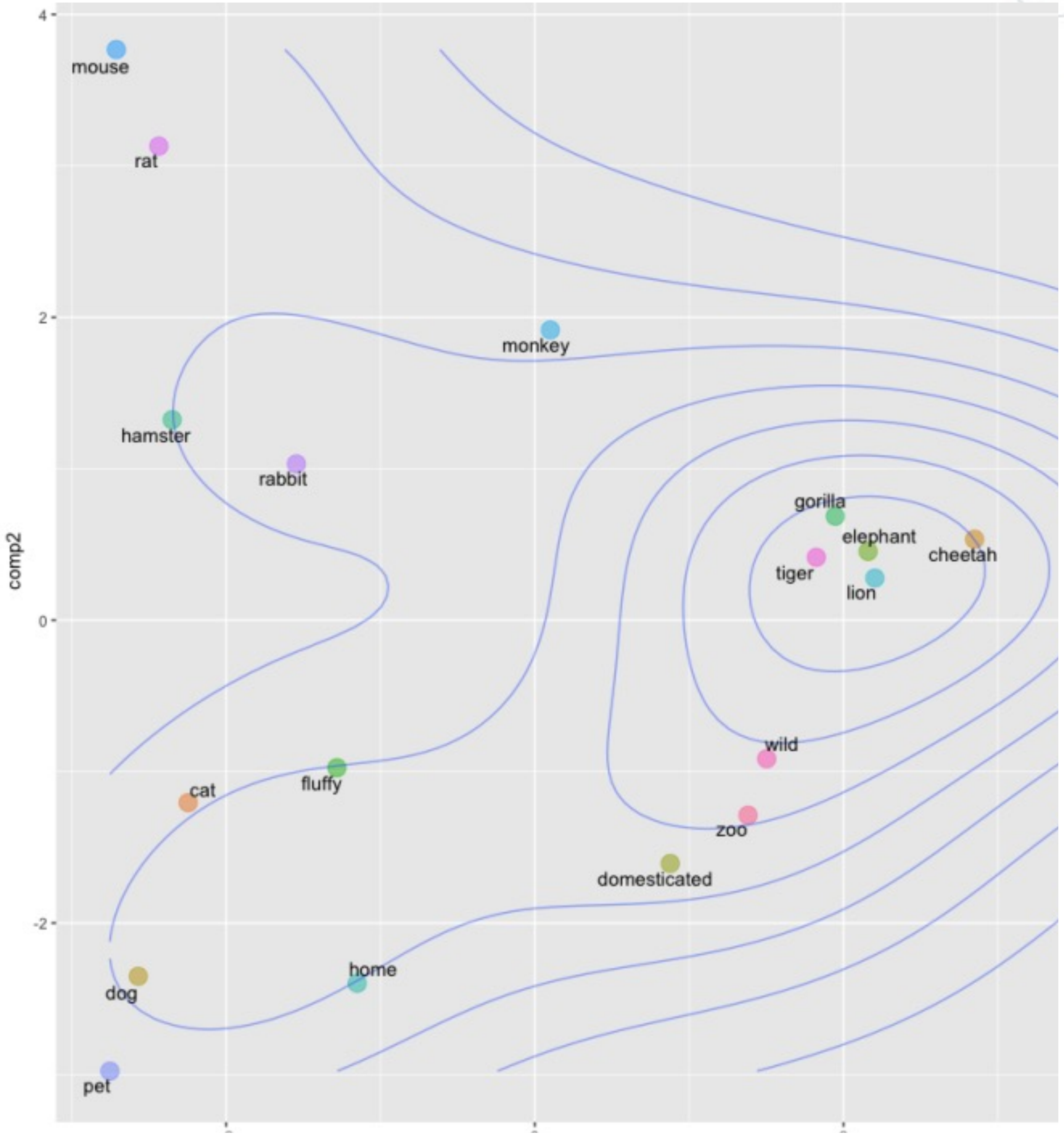
The number of neurons in the hidden layer represent the size of our vector encoding

The beauty of representing words as vectors is that they lend themselves to mathematical operators

Jayesh Bapu Ahire, Introduction to Word Vectors

Word embedding: word2vec

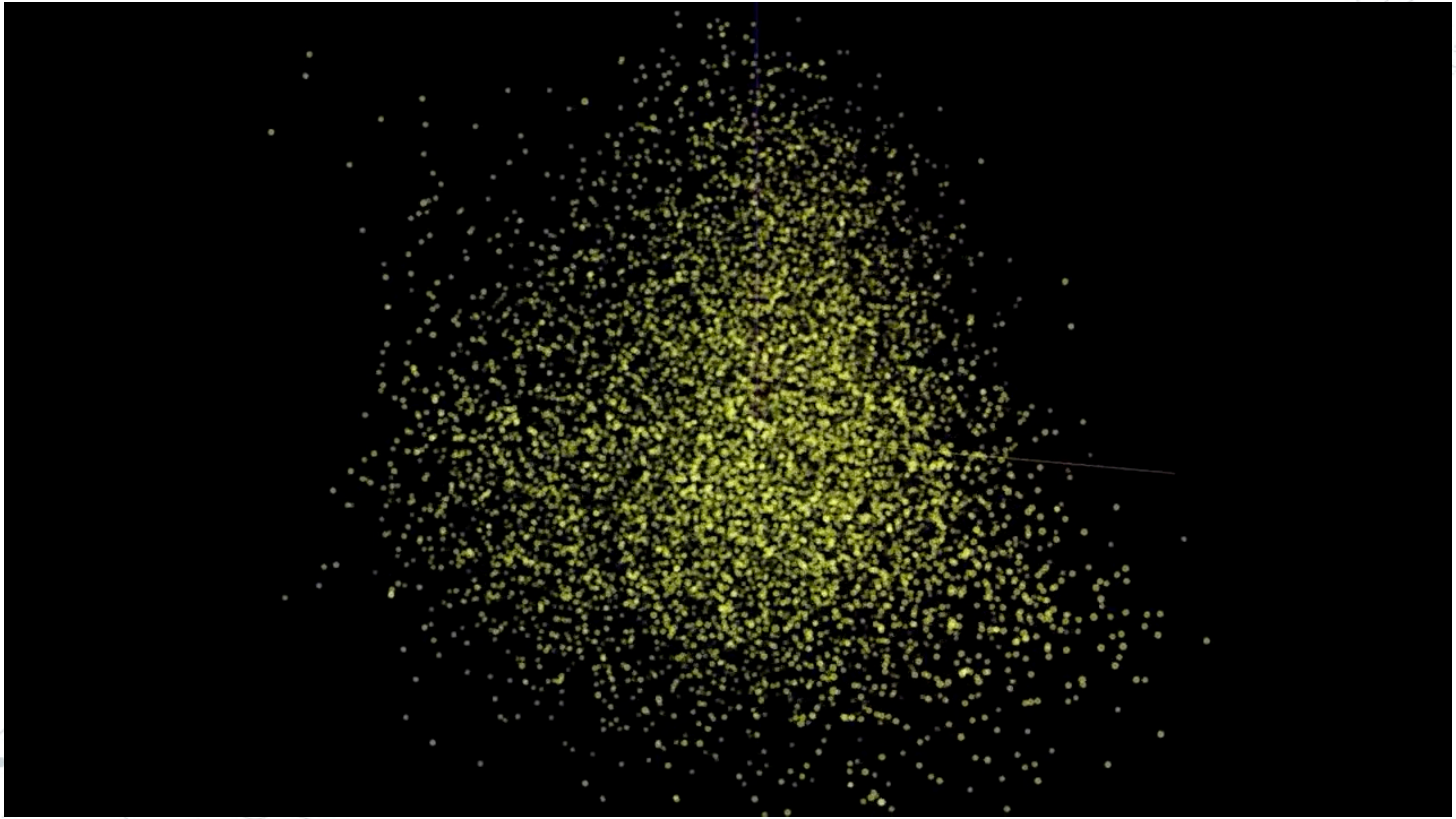
In the vector space, similar words will be closer.



Jayesh Bapu Ahire, Introduction to Word Vectors


comp2
0
-2
hamster
rabbit
gorilla
elephant
cheetah
tiger
lion
wild
cat
fluffy
zoo
domesticated
home
dog

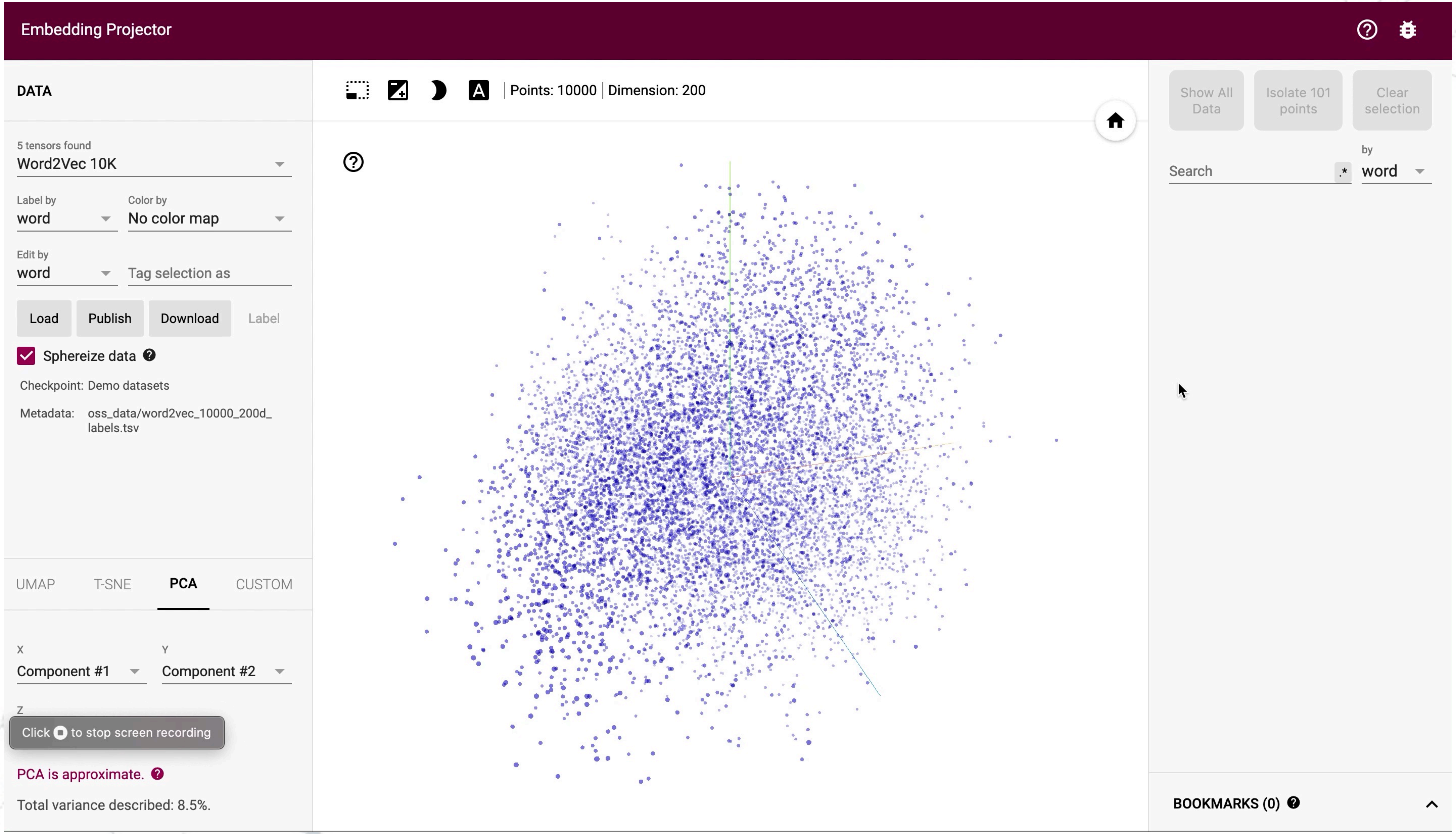
Word embedding: word2vec



A.I. Experiments: Visualizing High-Dimensional Space

https://www.youtube.com/watch?v=wvsE8jm1GzE

Word embedding: word2vec



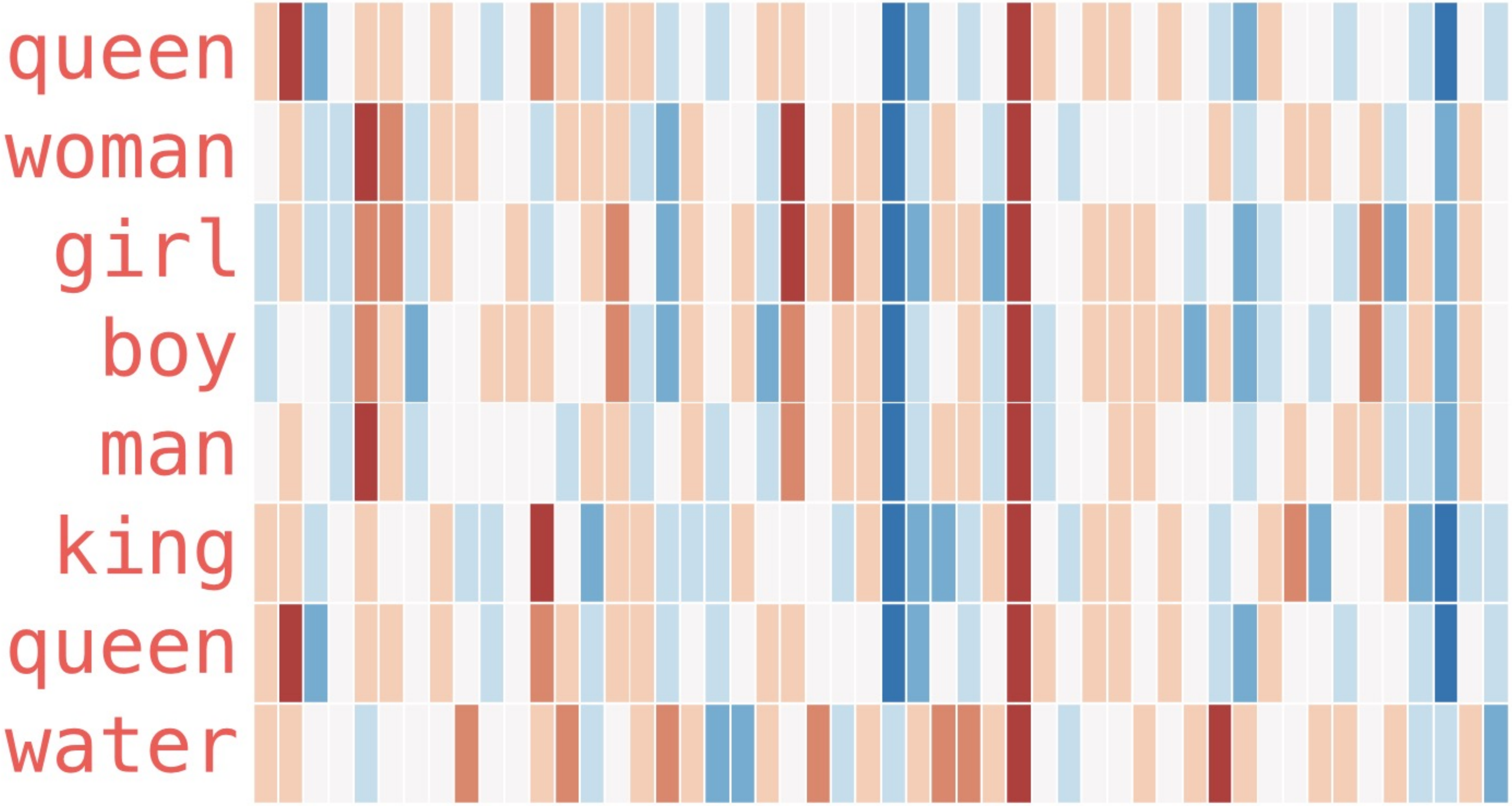
http://projector.tensorflow.org/

Word embedding operations

Jay Alammar, The Illustrated Word2vec

http://jalammar.github.io/illustrated-word2vec/

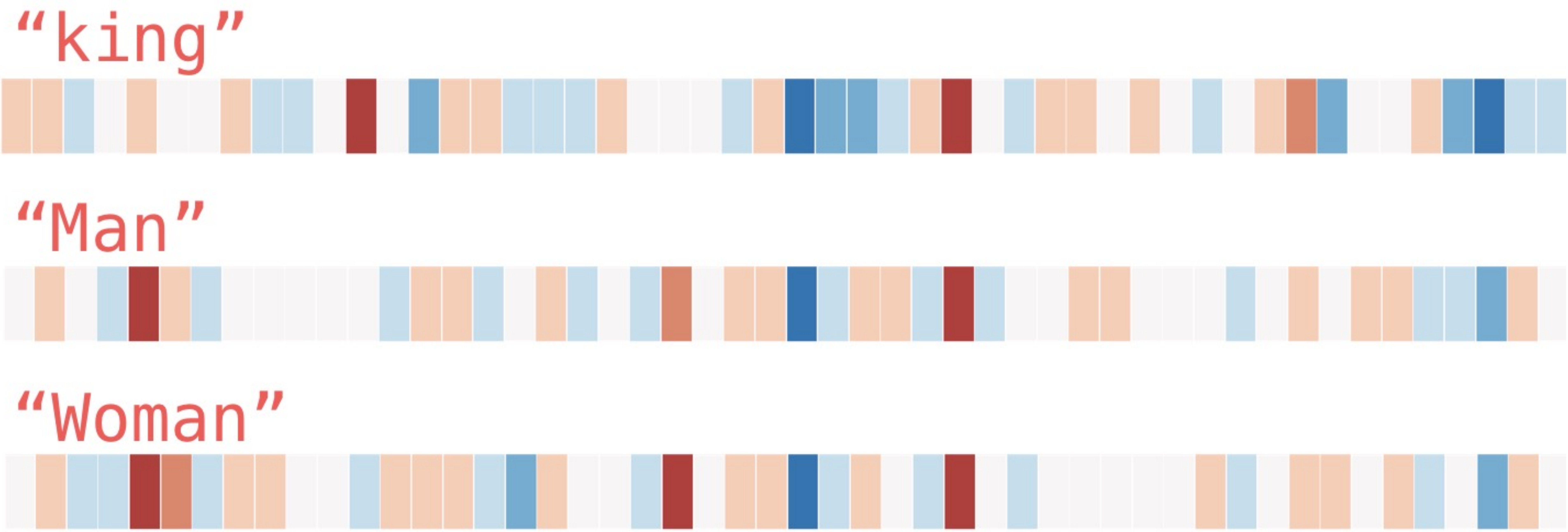
Word embedding operations



Jay Alammar, The Illustrated Word2vec

http://jalammar.github.io/illustrated-word2vec/

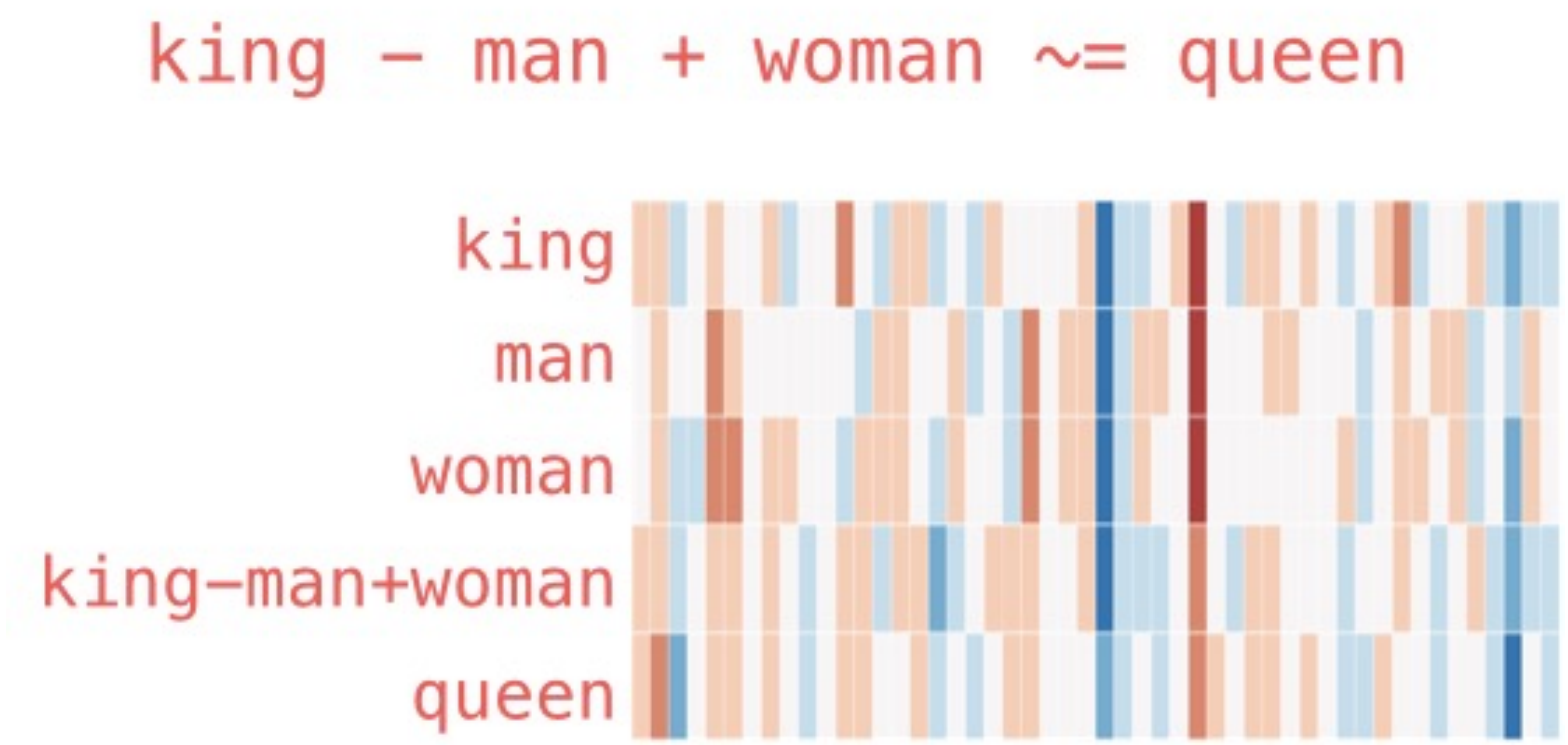
Word embedding operations



Jay Alammar, The Illustrated Word2vec

http://jalammar.github.io/illustrated-word2vec/

Word embedding operations




```
#%%

import gensim

path = "/Users/alina/Desktop/Lab2/GoogleNews-vectors-negative300.bin"
model = gensim.models.KeyedVectors.load_word2vec_format(path, binary=True)

#%%

# "King - man + woman" looks like:
model.most_similar_cosmul(positive=['king', 'woman'], negative=['man'])
```

Jay Alammar, The Illustrated Word2vec

http://jalammar.github.io/illustrated-word2vec/

Word embedding operations: DEMO

```python
#%%

import gensim

path = "/Users/alina/Desktop/Lab2/GoogleNews-vectors-negative300.bin"
model = gensim.models.KeyedVectors.load_word2vec_format(path, binary=True)

#%%

# "King - man + woman" looks like:
model.most_similar_cosmul(positive=['king', 'woman'], negative=['man'])
```

Recurrent Neural Networks

Why do we need Recurrent Neural Networks?

- "My name is Teddy".

0 0 0 1

1 - Name
0 - Not Name

- Claire said: "Teddy bears are my favourite toys."

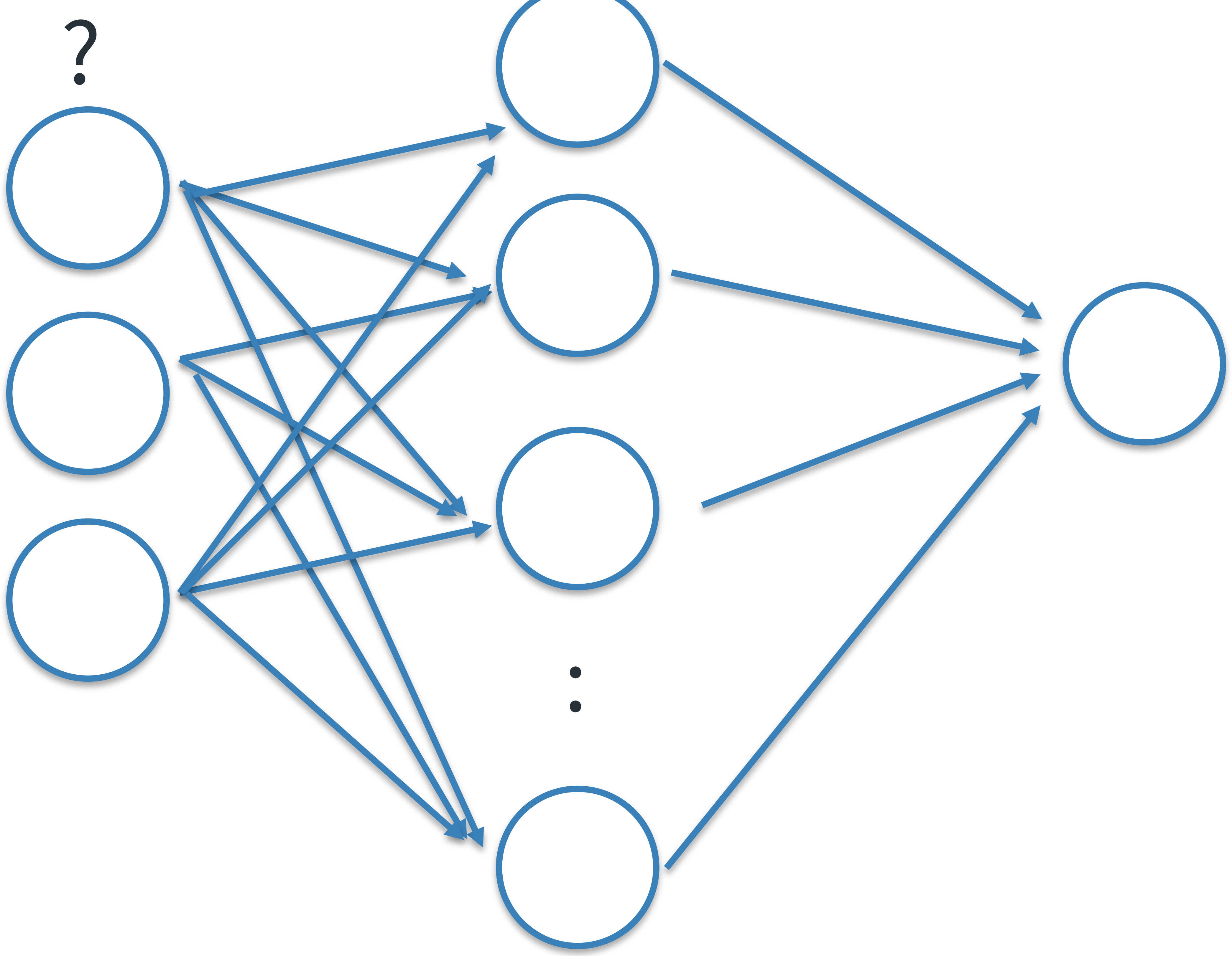
1 0 0 0 0 0 0 0

Why do we need Recurrent Neural Networks?

- “My name is Teddy”.
- Claire said: “Teddy bears are my favourite toys.”

How do we give input to a Neural Network a sentence?

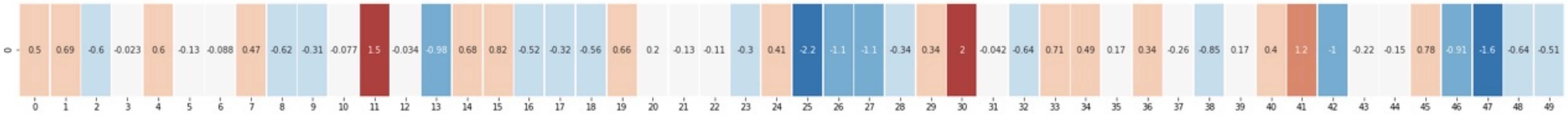
Main problems:
- Variable size of the input, variable size of the output
- The order of the words is important!



Recurrent Neural Networks

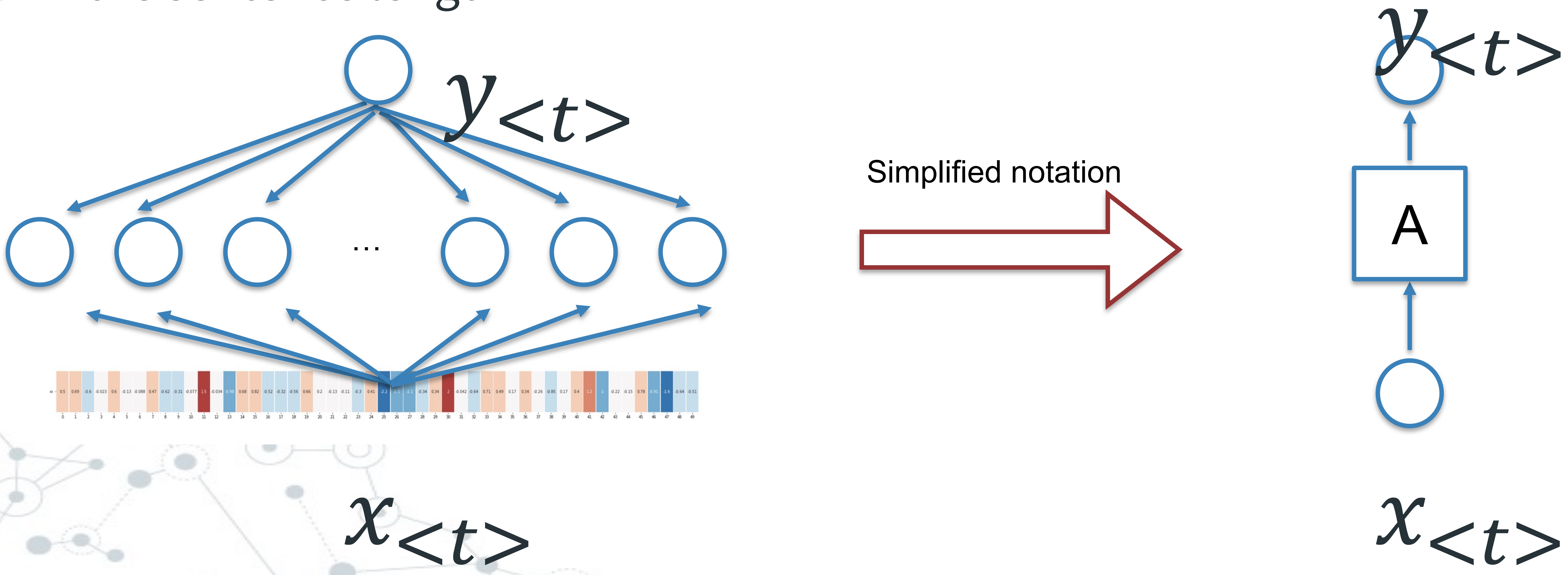
◎"My name is Teddy".

◎Let $x_{<t>}$ be the word embedding for the word at position t:



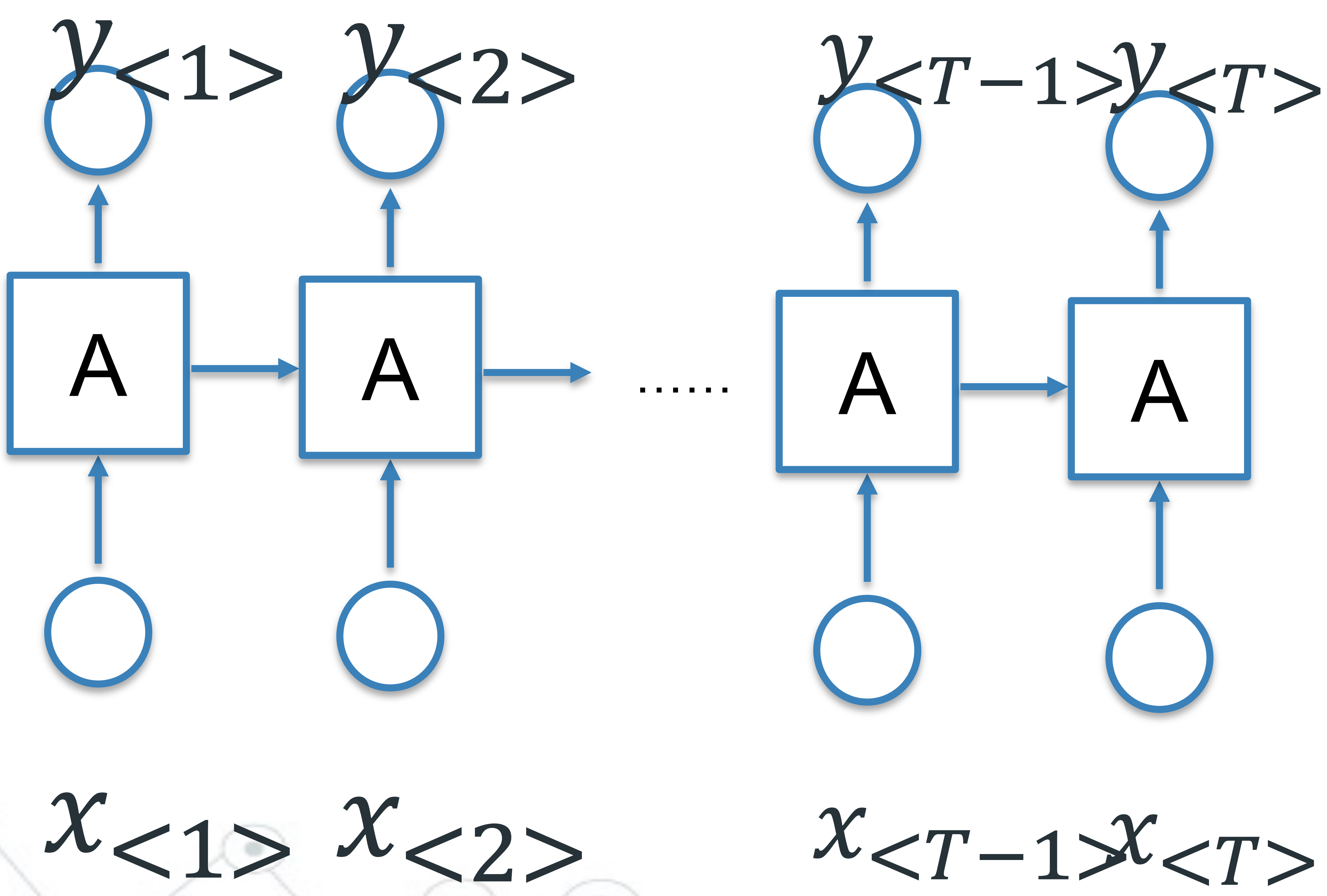
Recurrent Neural Networks

- "My name is Teddy".
- Let $x_{<t>}$ be the word embedding for the word at position t
- Let $y_{<t>}$ be the output value for the input $x_{<t>}$
- T - the sentence length

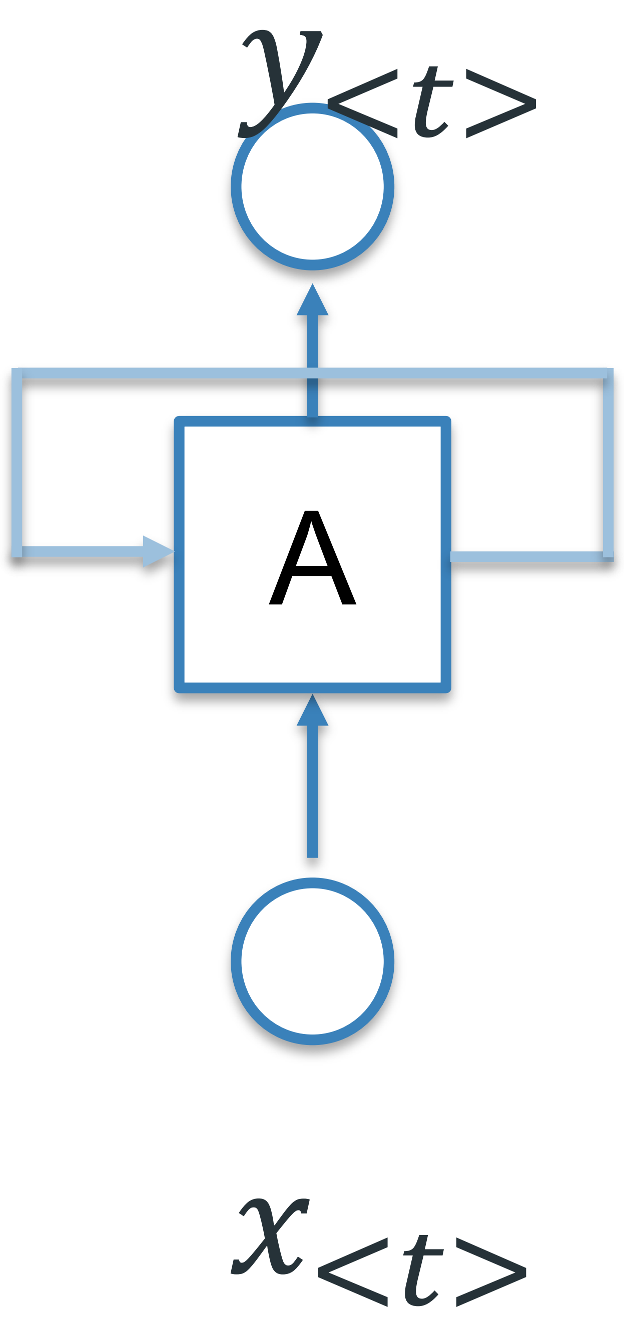


Recurrent Neural Networks

Unrolled form of the RNN

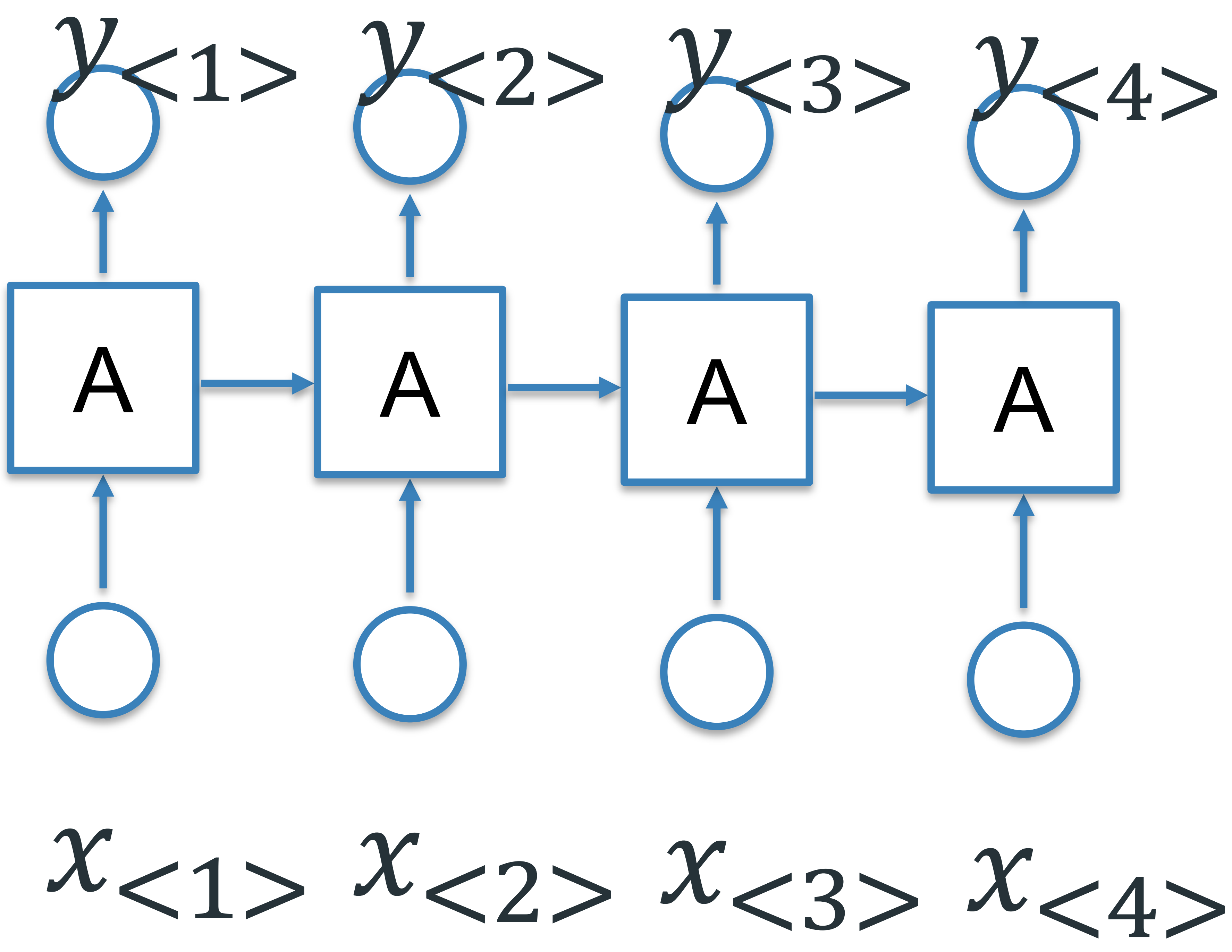


Rolled form of the RNN



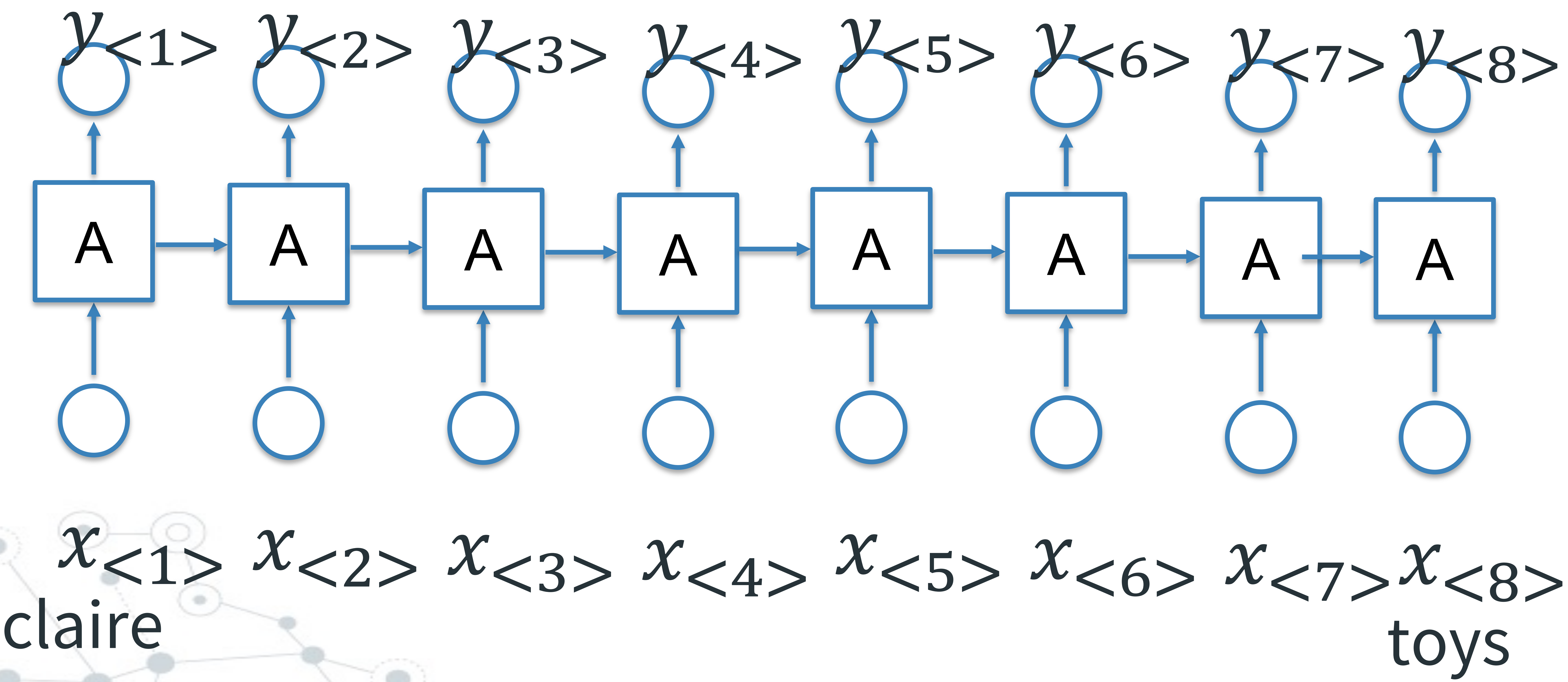
Recurrent Neural Networks

- "My name is Teddy".



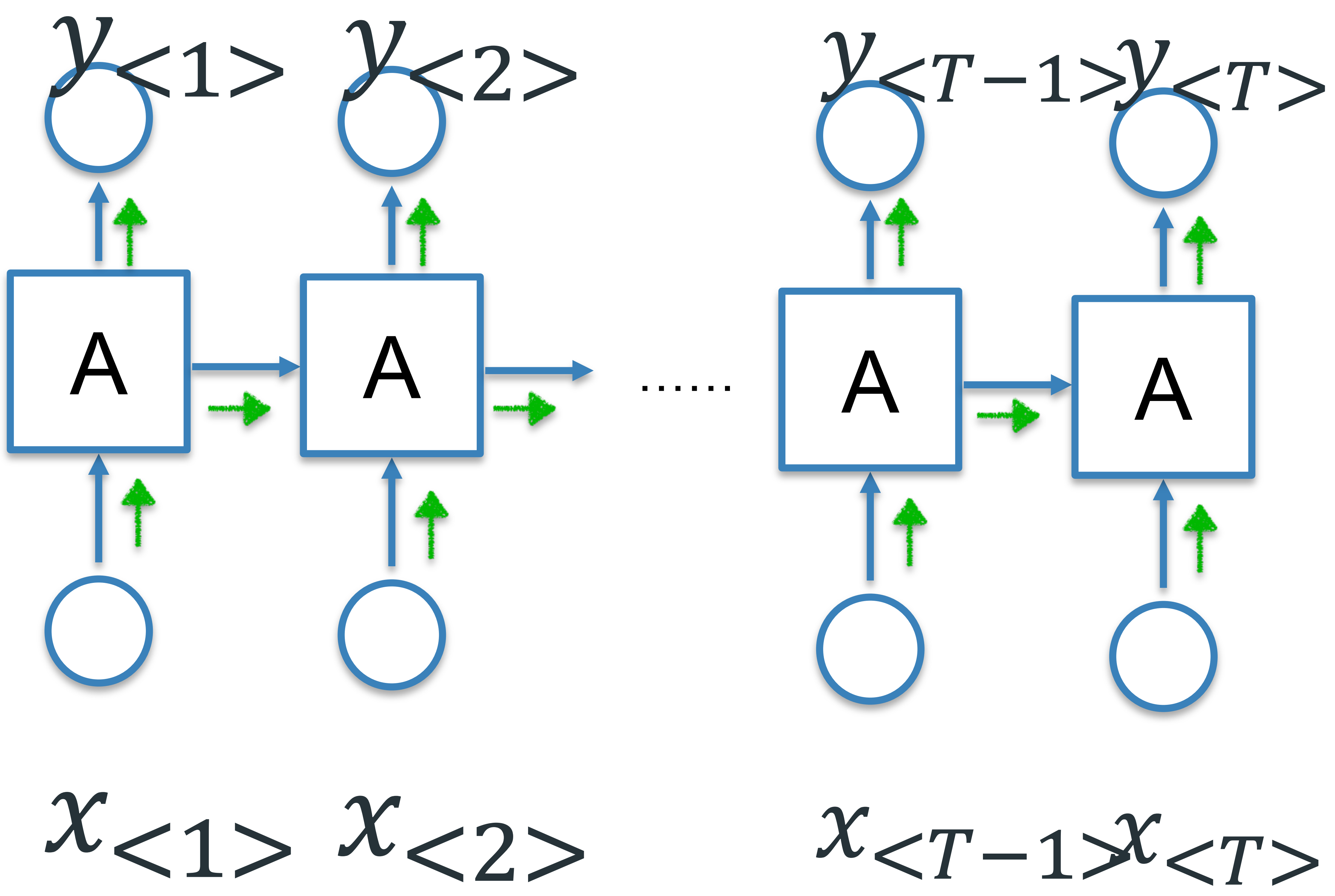
Recurrent Neural Networks

◎ Claire said: “Teddy bears are my favourite toys.”



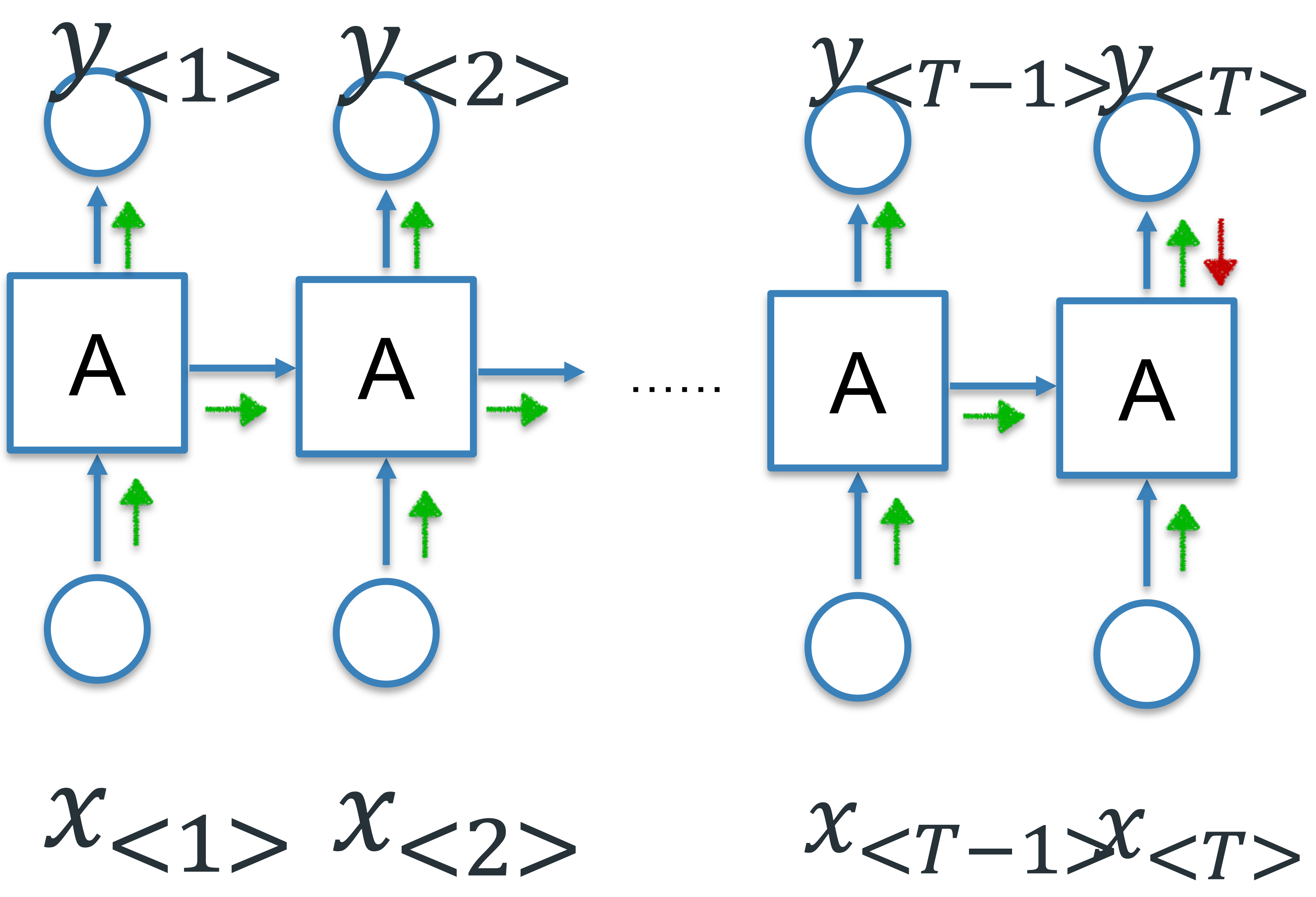
Back-propagation for RNN

◎Back-propagation through time



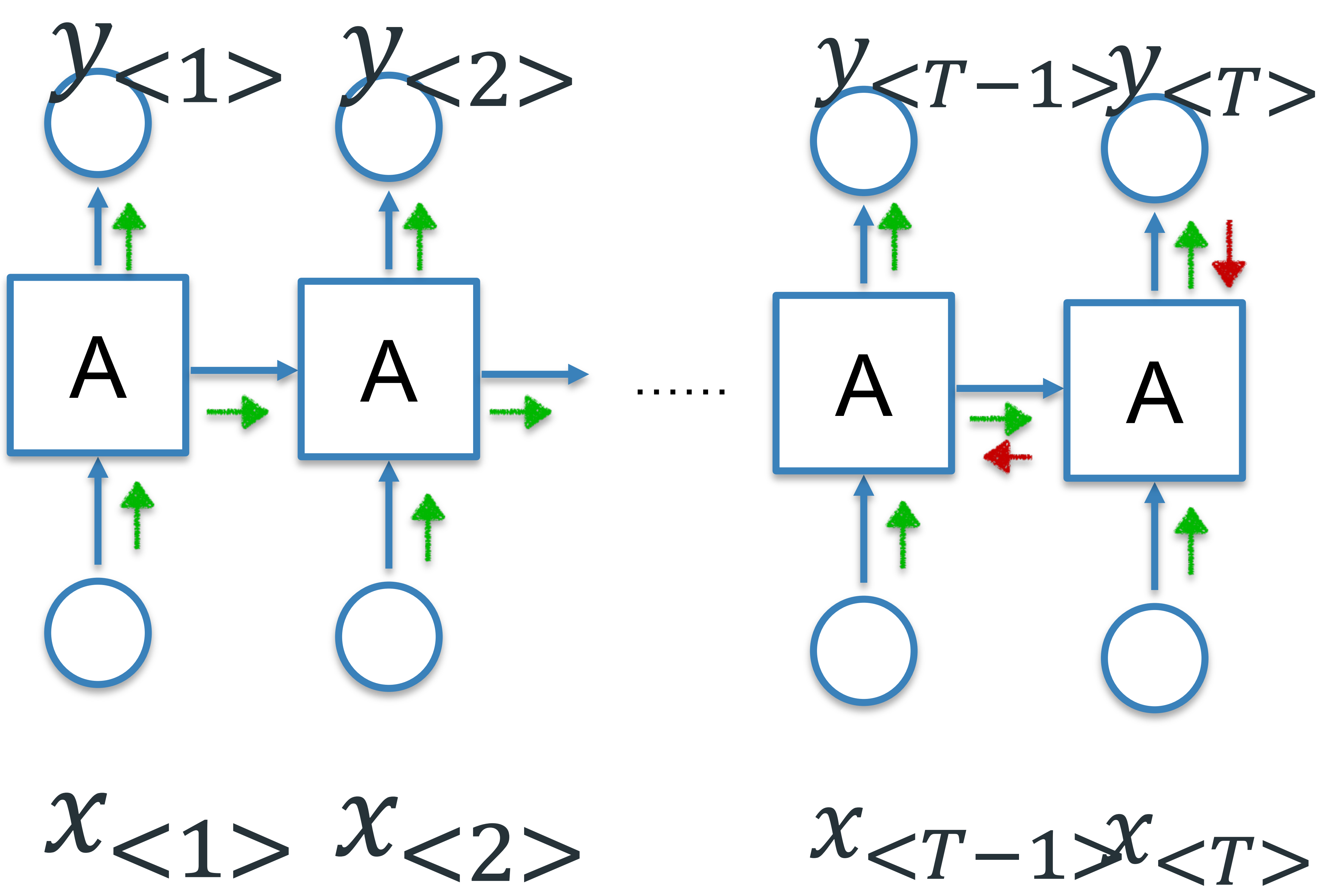
Back-propagation for RNN

◎Back-propagation through time



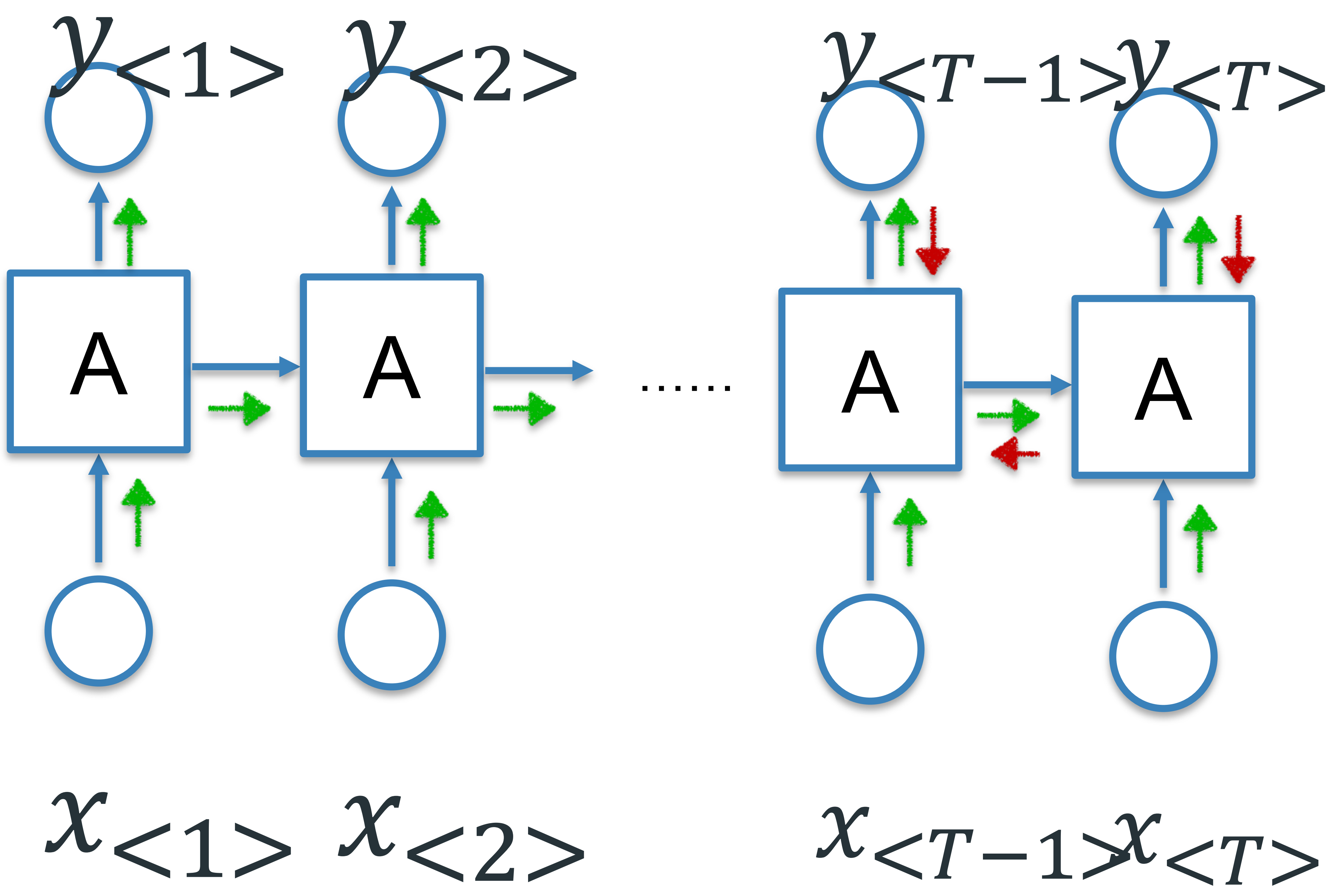
Back-propagation for RNN

◎Back-propagation through time



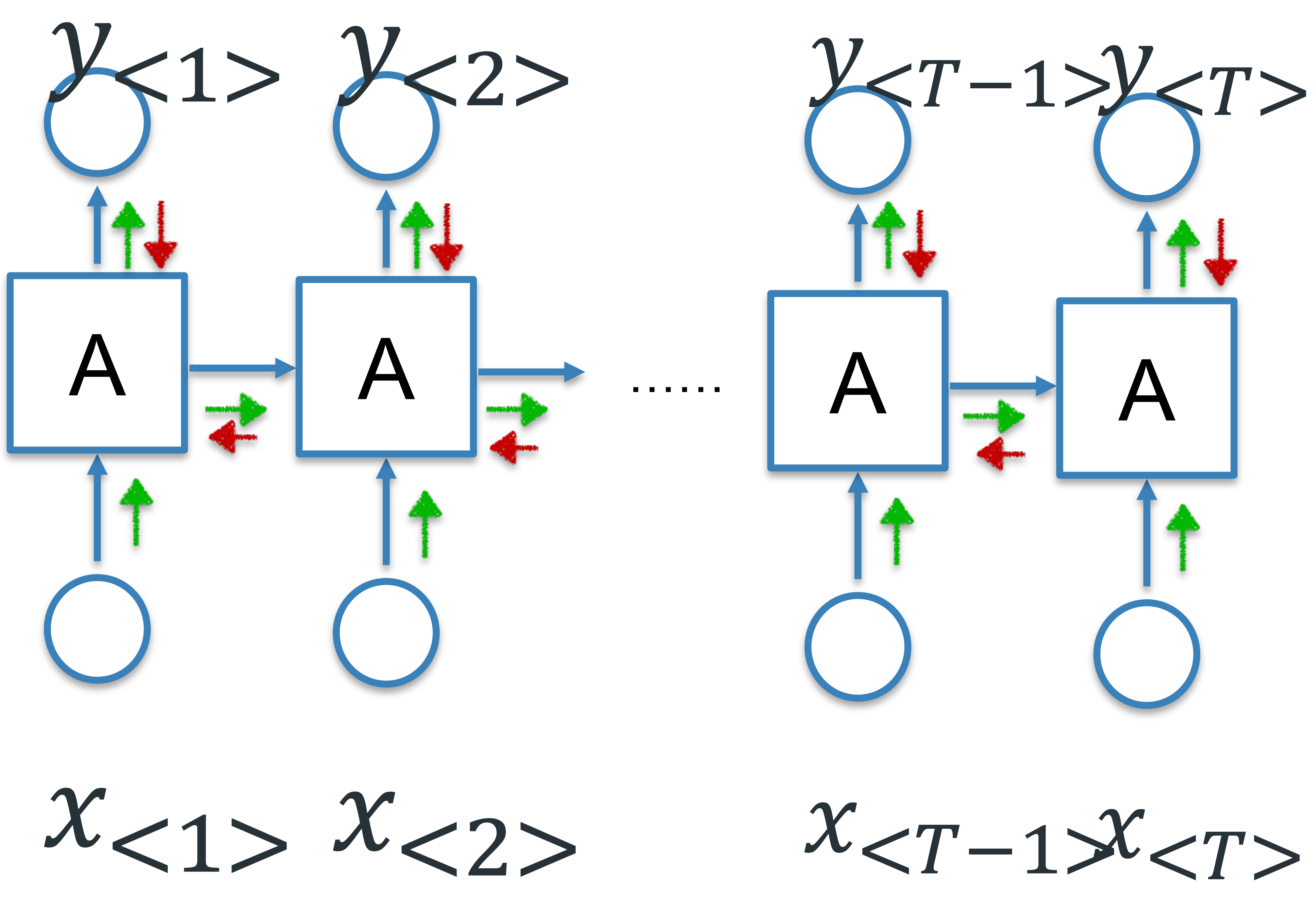
Back-propagation for RNN

◎Back-propagation through time



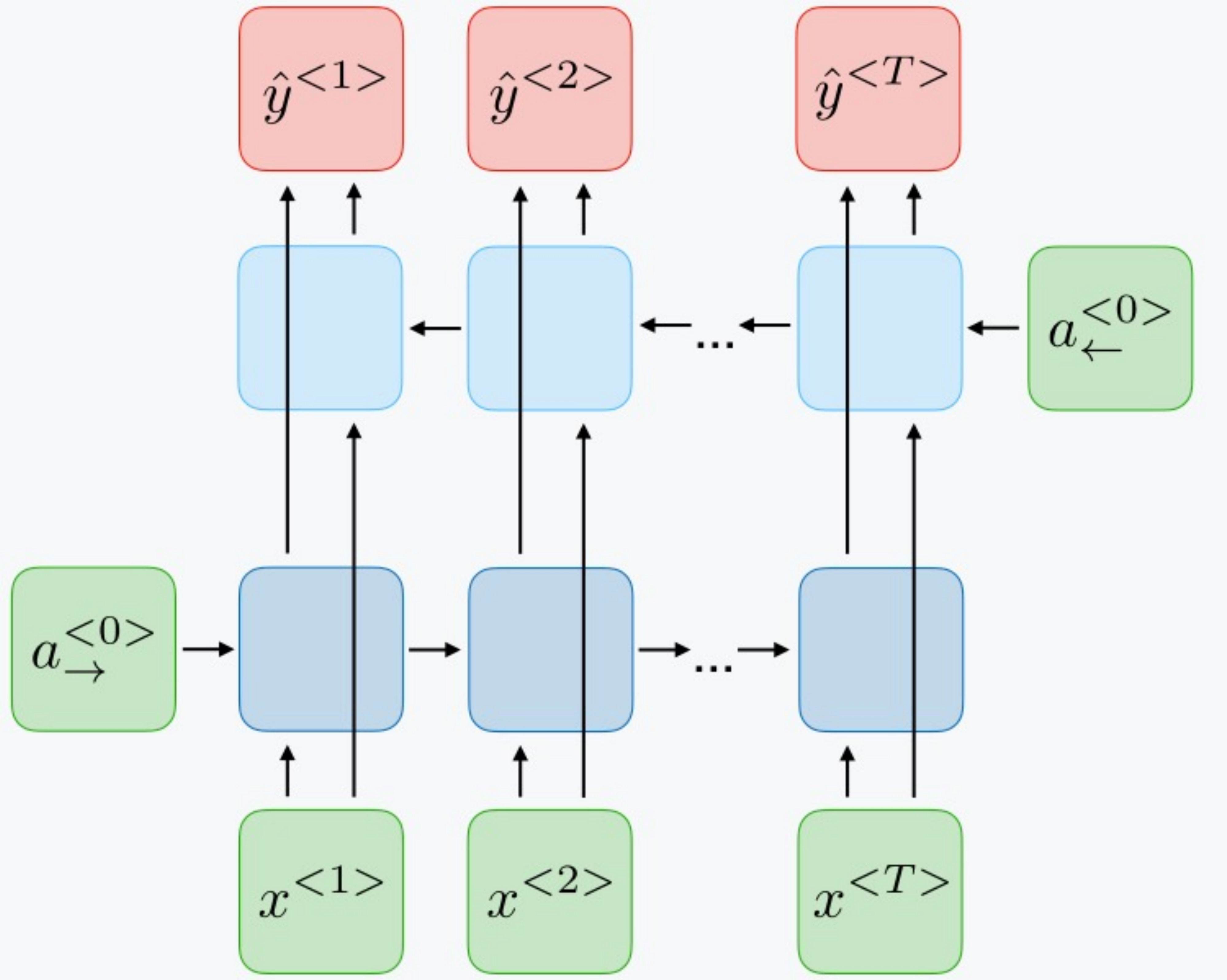
Back-propagation for RNN

◎Back-propagation through time



Other types of RNN

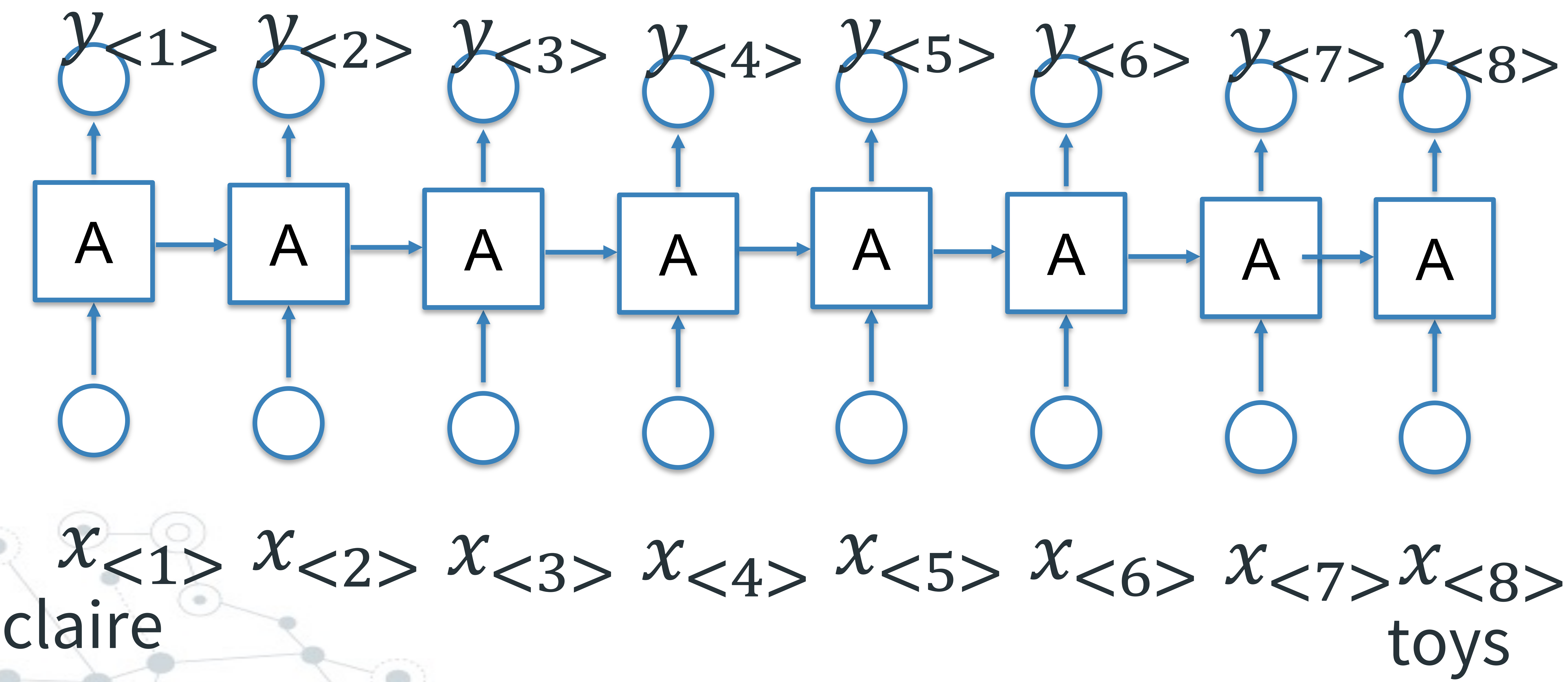
Bidirectional RNN



https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

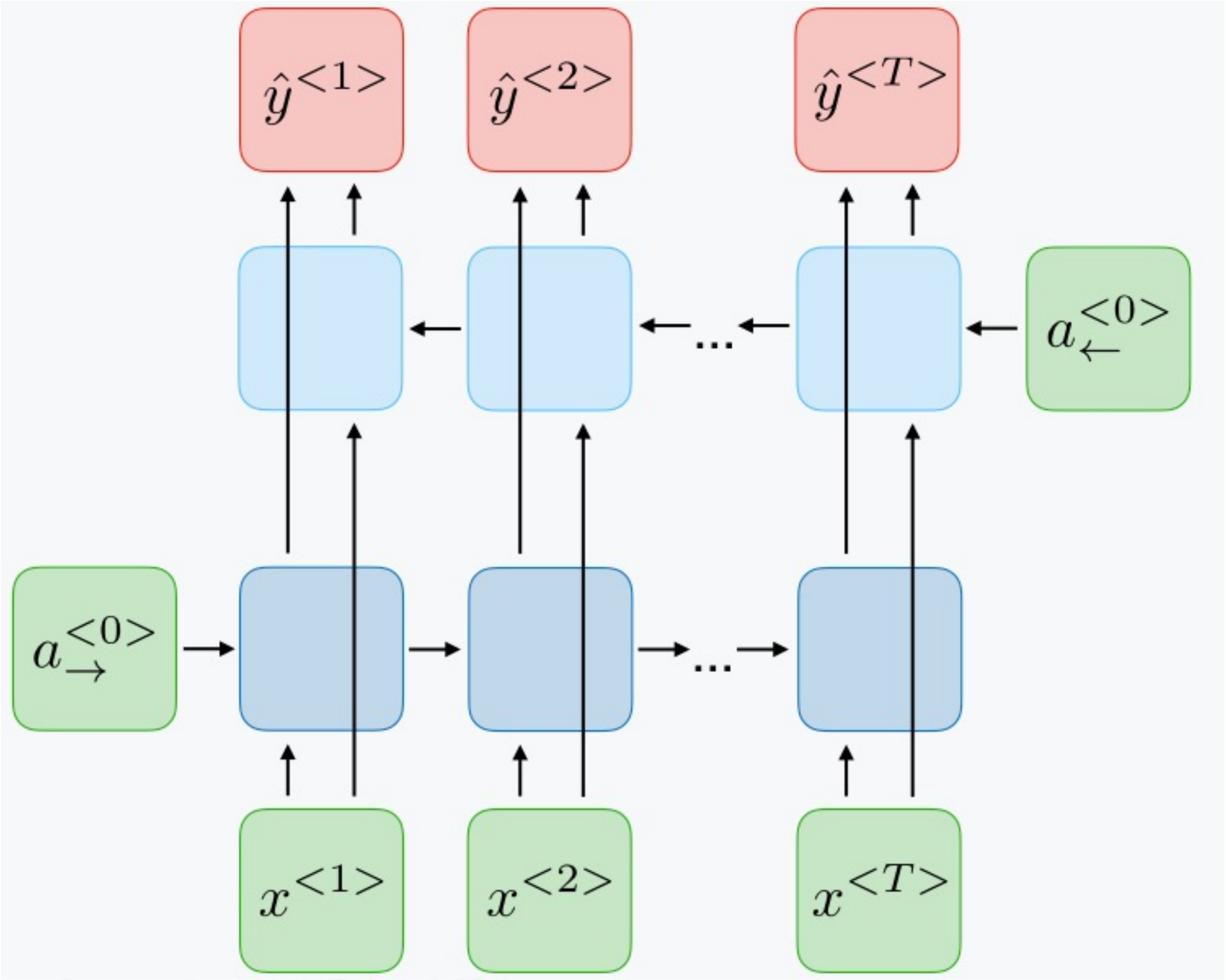
Recurrent Neural Networks

- Claire said: “Teddy bears are my favourite toys.”



Other types of RNN

Bidirectional RNN

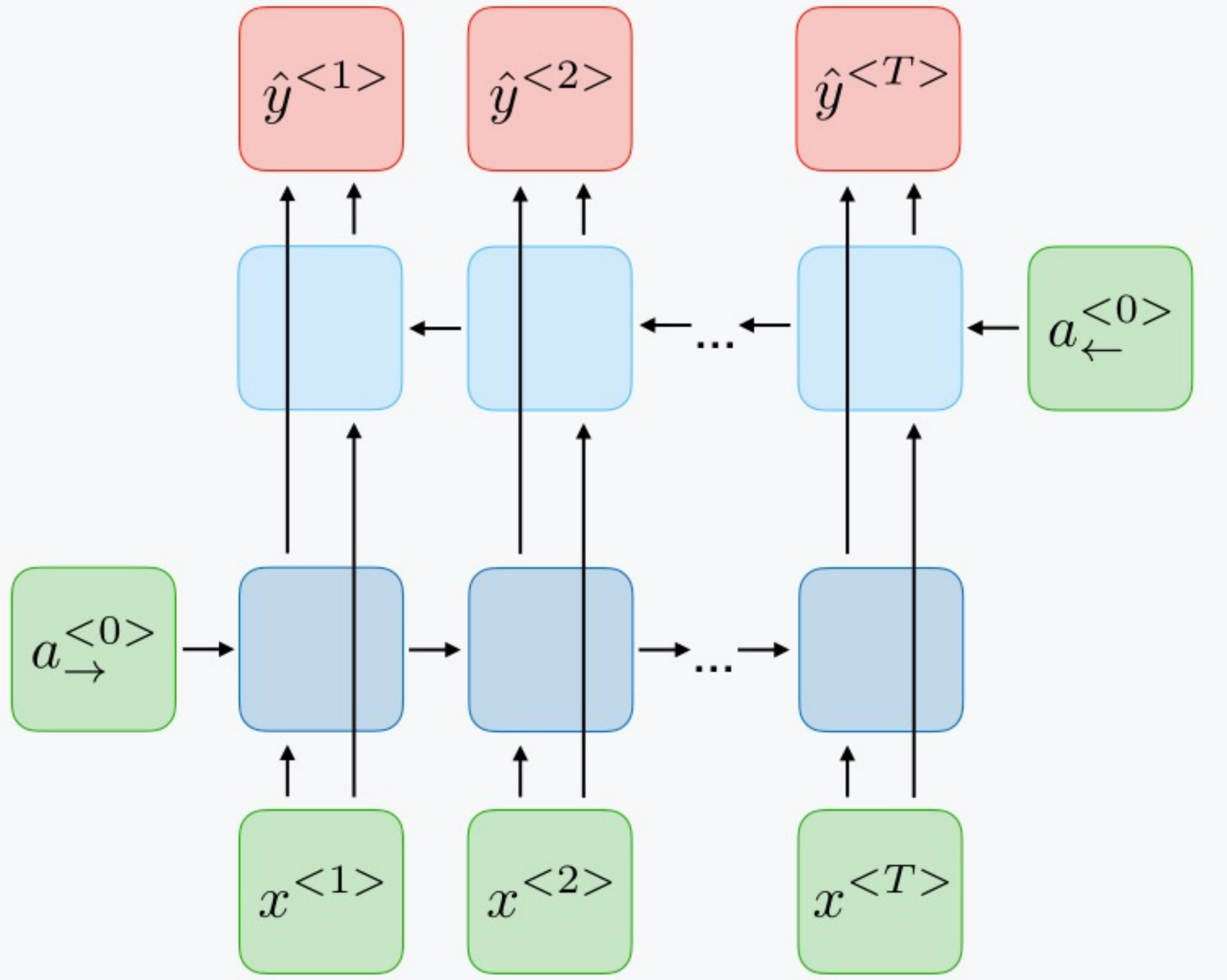


- Claire said: "Teddy bears are my favourite toys."

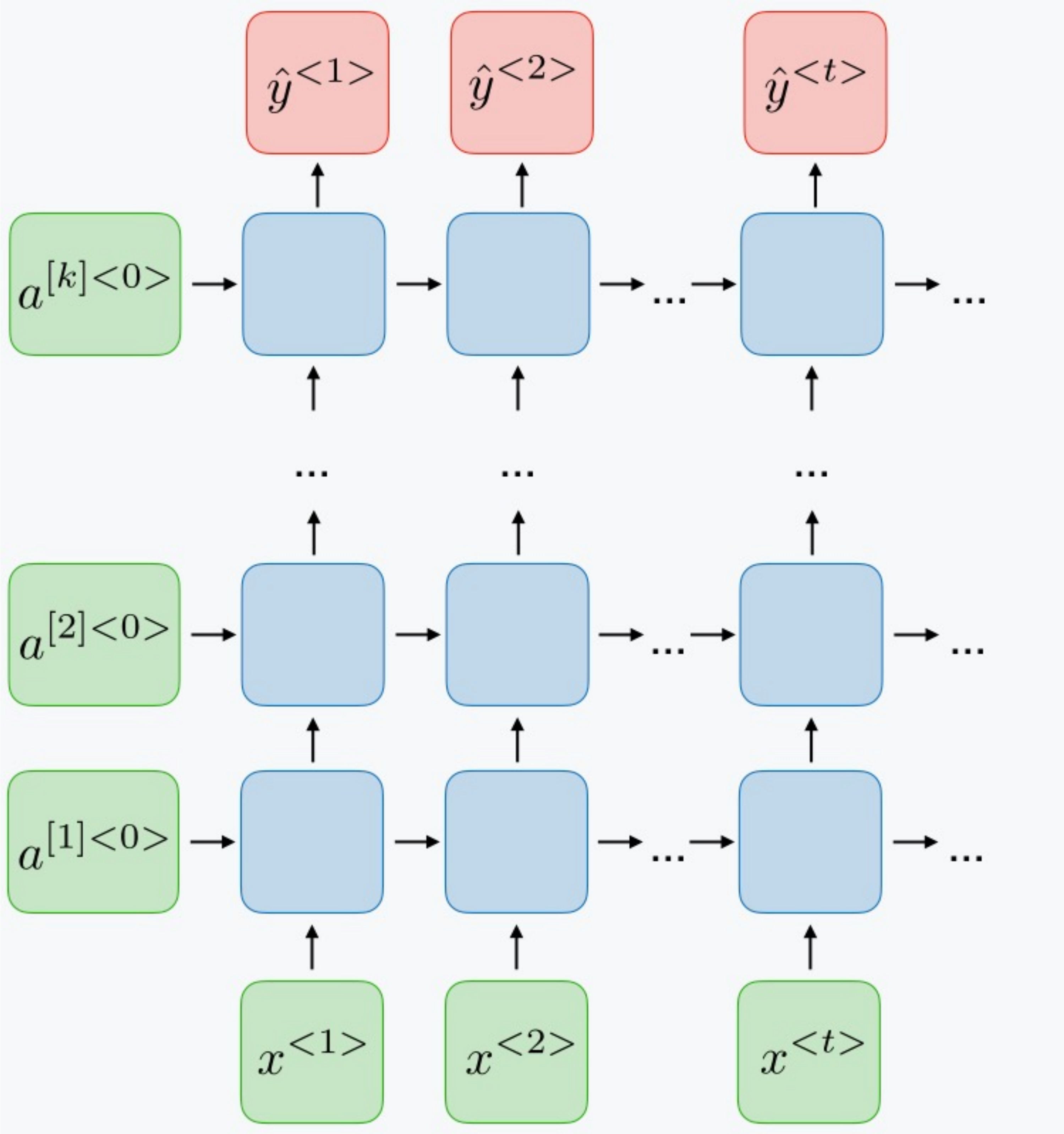
https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Other types of RNN

Bidirectional RNN

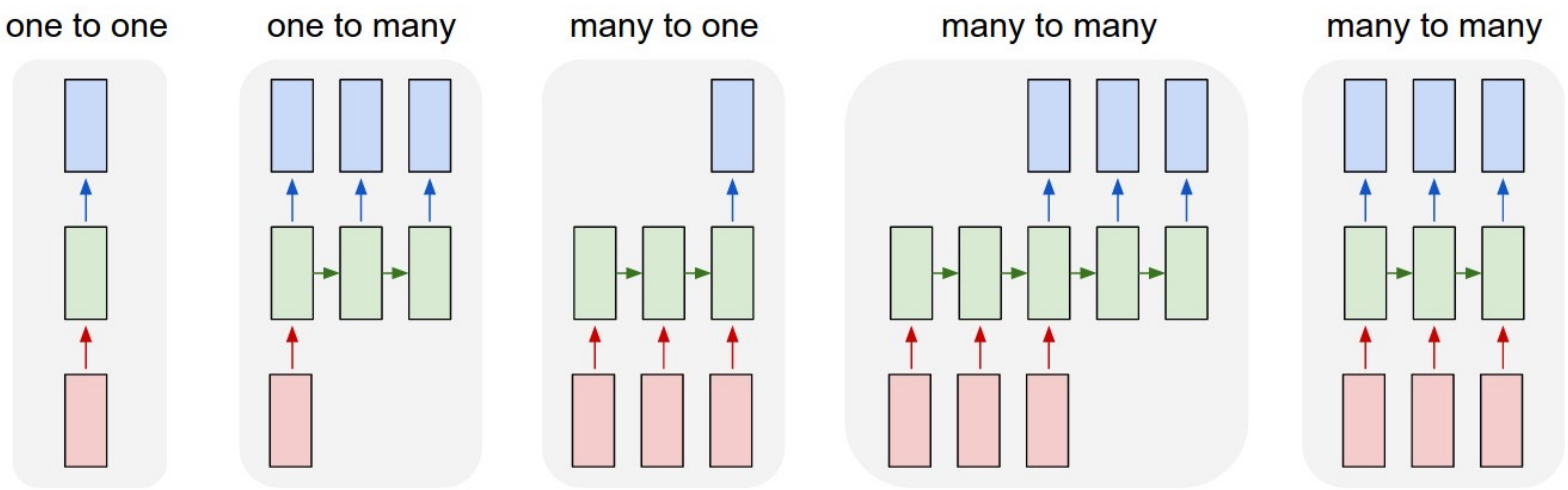


Deep RNN



https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Types of RNN



Traditional NN
Without any loops

Text
generation

Sentiment
Classification;
Action
classification

Machine
translation

Named entity
recognition; Part of
words tagging

The Unreasonable Effectiveness of Recurrent Neural Networks

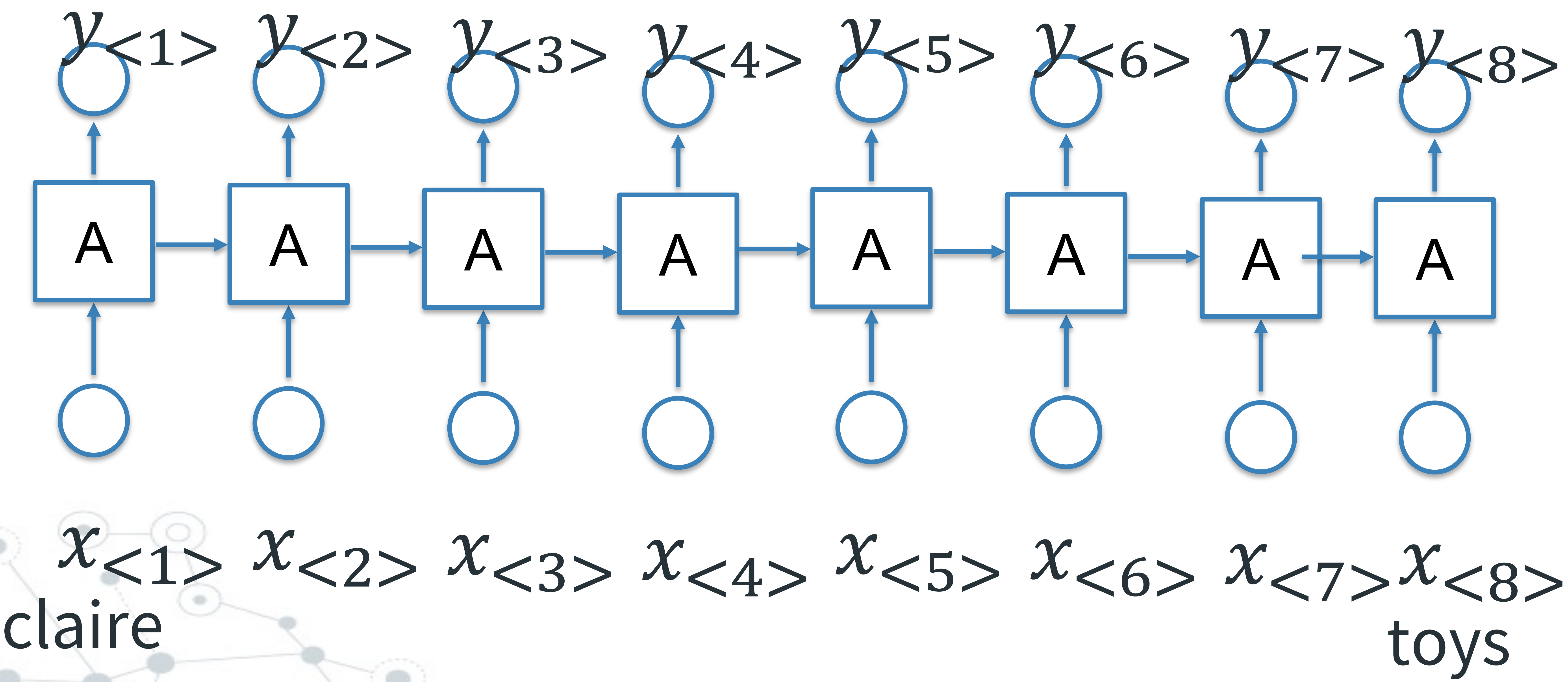
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Problems with RNN

- Vanishing Gradient: the lower your gradient is and the harder it is to train the weights

Recurrent Neural Networks

- Claire said: “Teddy bears are my favourite toys.”



Problems with RNN

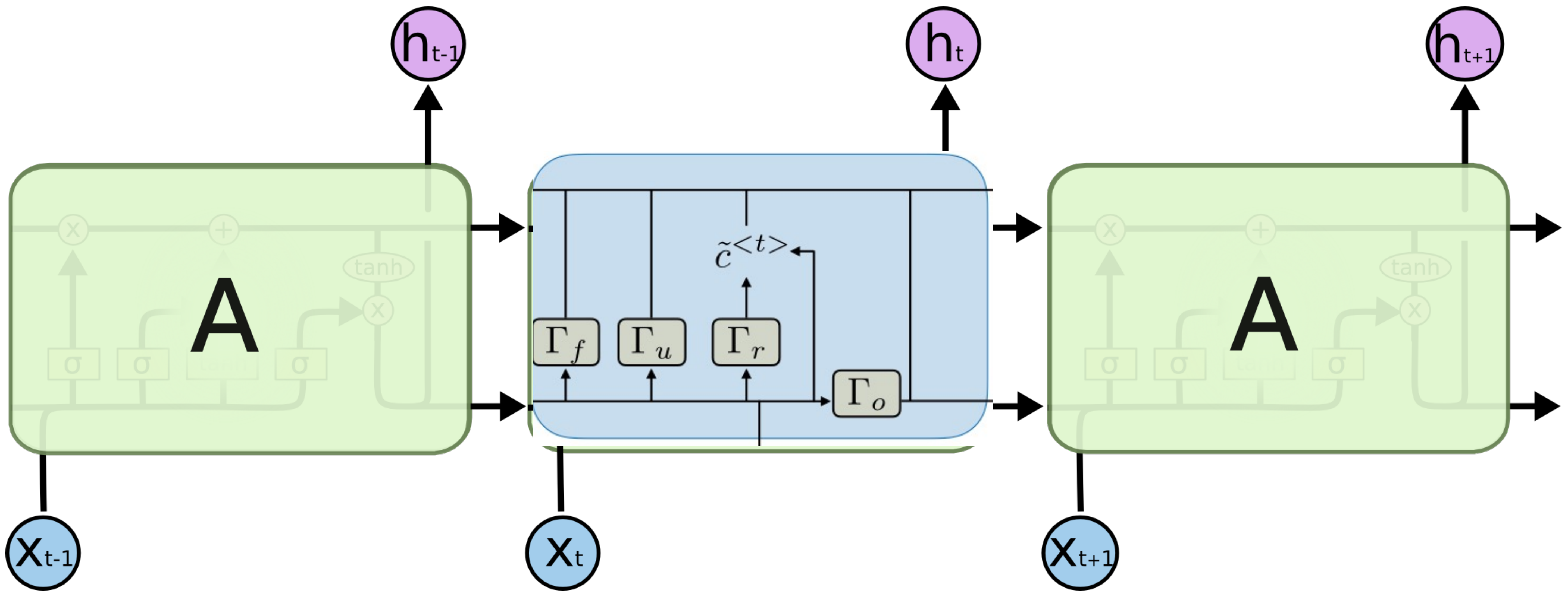
◎Vanishing Gradient: the lower your gradient is and the harder it is to train the weights

- Solution: Long Short-Term Memory Networks; ResNet (Residual Networks); Attention Neural Networks

◎It can be difficult to train

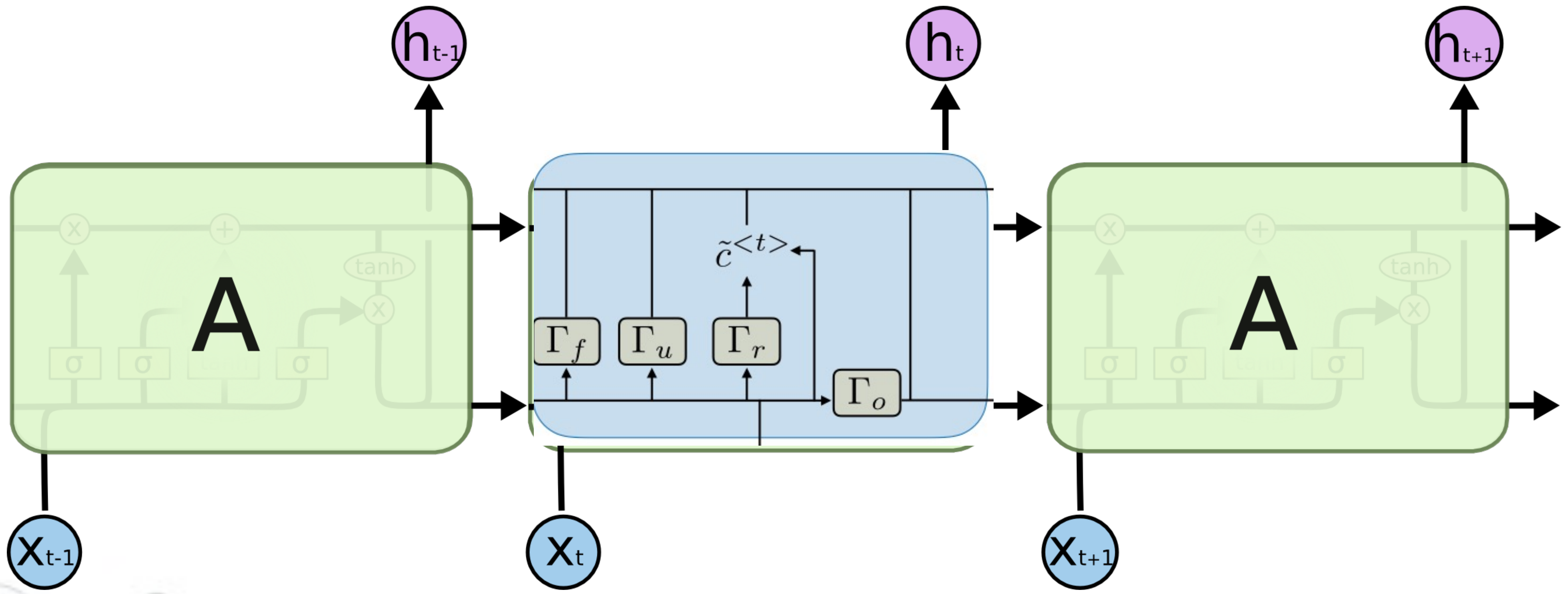
◎Are not able to keep track of long-time dependencies (LSTMs and Attention based Neural Network solve this problem partially)

Long-Short Term neural networks



https://analyticsindiamag.com/overview-of-recurrent-neural-networks-and-their-applications/

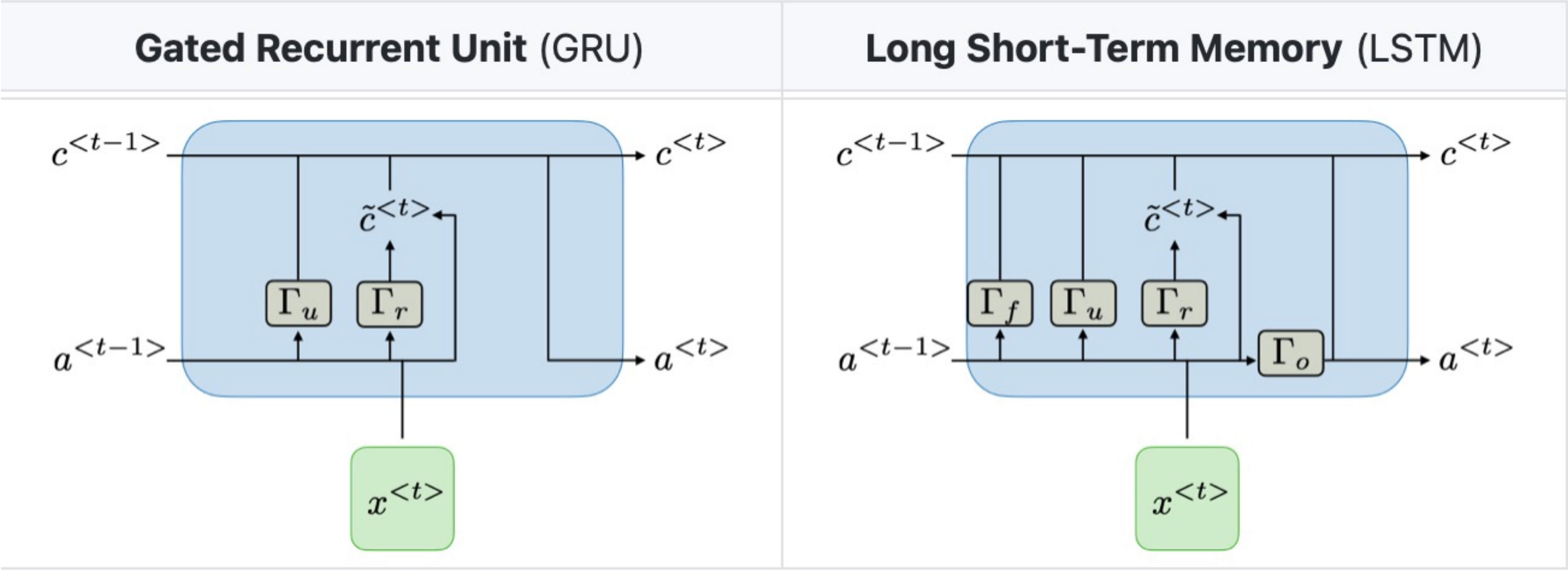
Long-Short Term neural networks



- LSTMs are similar with RNN
- Instead of a simple Layer of neurons, it has some particular type of neurons/cells:
  - How much the past should matter at this point? Γ_u (update gate)
  - Can we just drop the previous information? Γ_r (relevance gate)
  - Erase altogether some previous information: Γ_f (forget gate)
  - How much of this information should we pass as output? Γ_o (output gate)

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#architecture

GRU (Gated Recurrent Units) vs LSTM

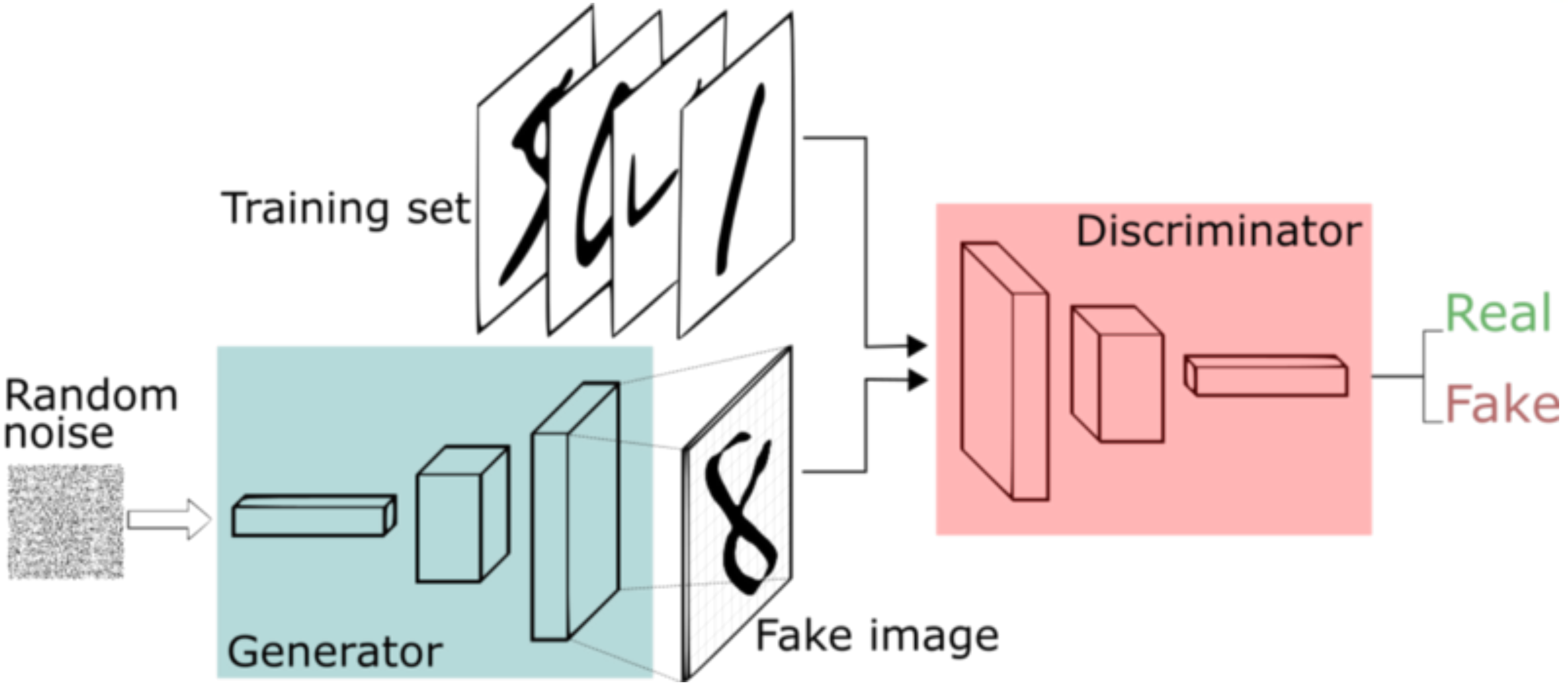


https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#architecture

Other network architectures…

Generative adversarial neural networks (GANs)

Generative Adversarial Networks



An intuitive introduction to Generative Adversarial Networks (GANs)

https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/

Image-to-Image Translation with Conditional Adversarial Networks

Phillip Isola   Jun-Yan Zhu   Tinghui Zhou   Alexei A. Efros

Berkeley AI Research (BAIR) Laboratory, UC Berkeley

{isola,junyanz,tinghuiz,efros}@eecs.berkeley.edu

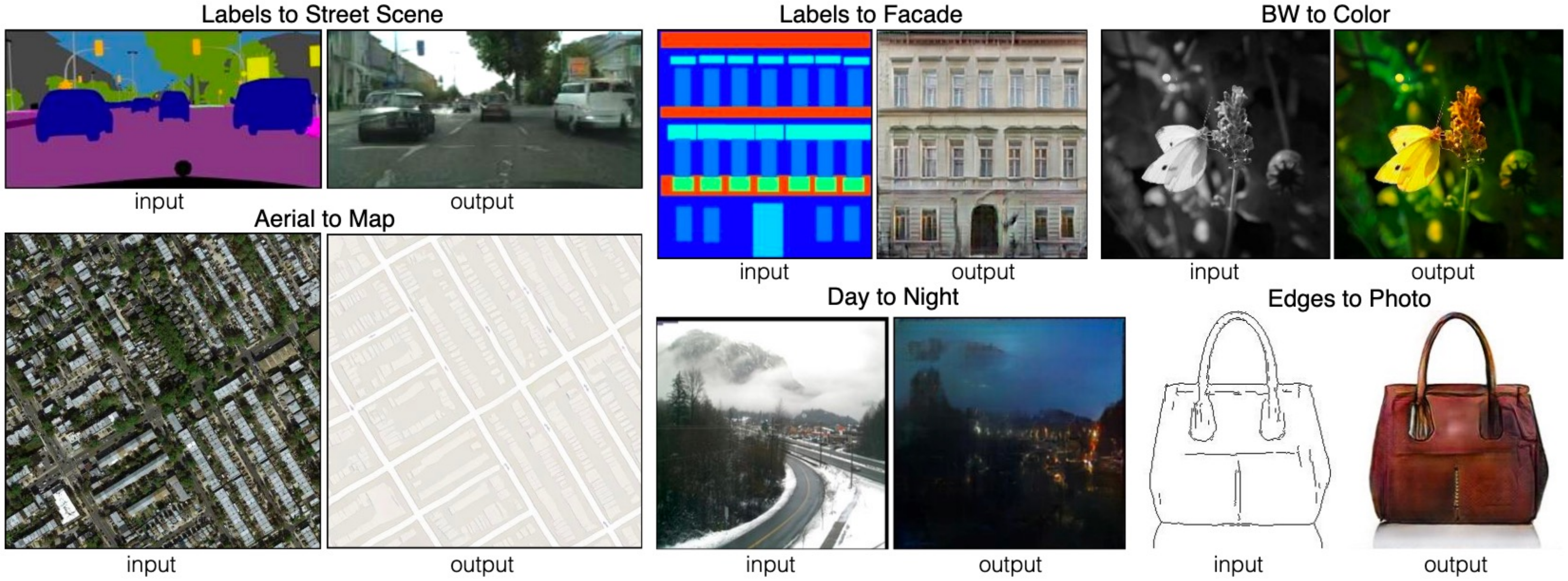


Figure 1: Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. Conditional adversarial nets are a general-purpose solution that appears to work well on a wide variety of these problems. Here we show results of the method on several. In each case we use the same architecture and objective, and simply train on different data.

https://arxiv.org/pdf/1611.07004.pdf

Image-to-Image Demo

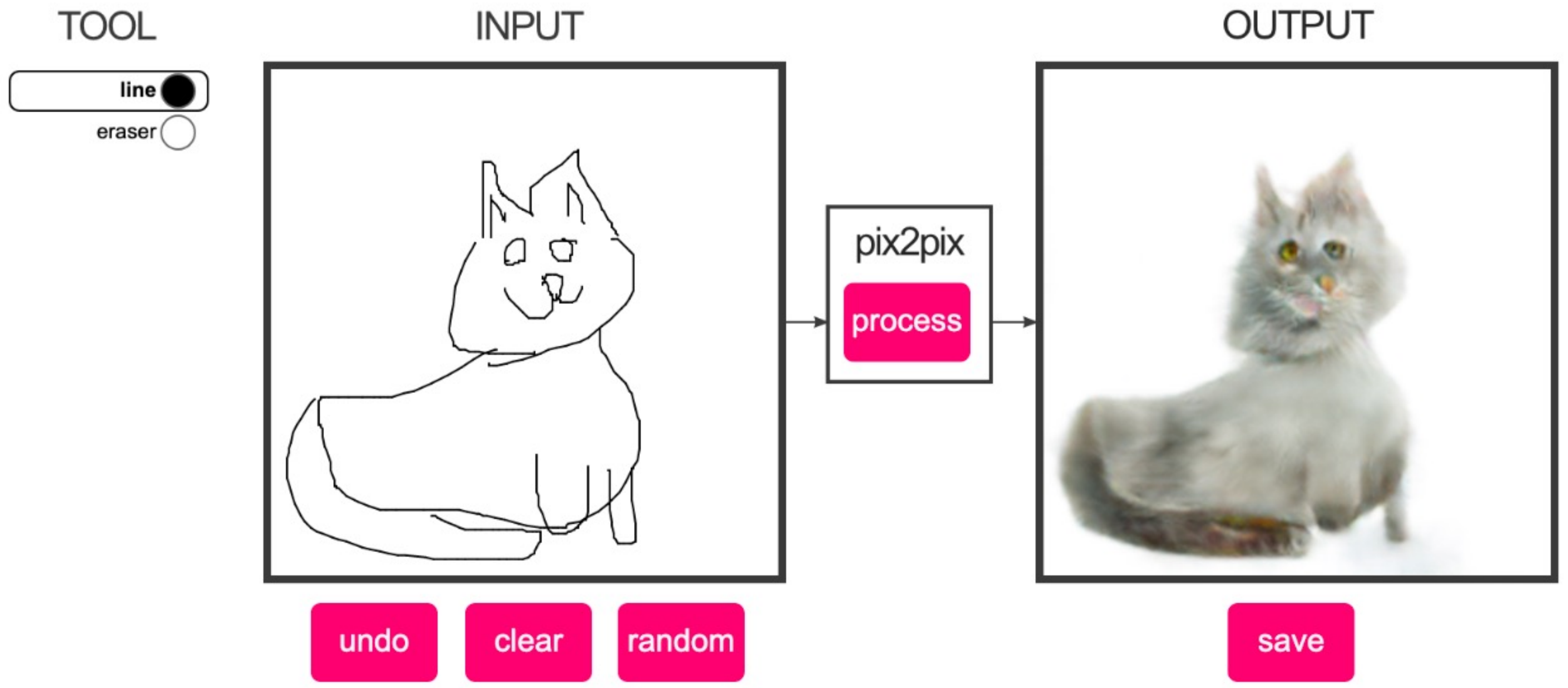
Interactive Image Translation with pix2pix-tensorflow

Written by Christopher Hesse — February 19th, 2017

Recently, I made a Tensorflow port of pix2pix by Isola et al., covered in the article Image-to-Image Translation in Tensorflow. I've taken a few pre-trained models and made an interactive web thing for trying them out. Chrome is recommended.

The pix2pix model works by training on pairs of images such as building facade labels to building facades, and then attempts to generate the corresponding output image from any input image you give it. The idea is straight from the pix2pix paper, which is a good read.

edges2cats



https://affinelayer.com/pixsrv/

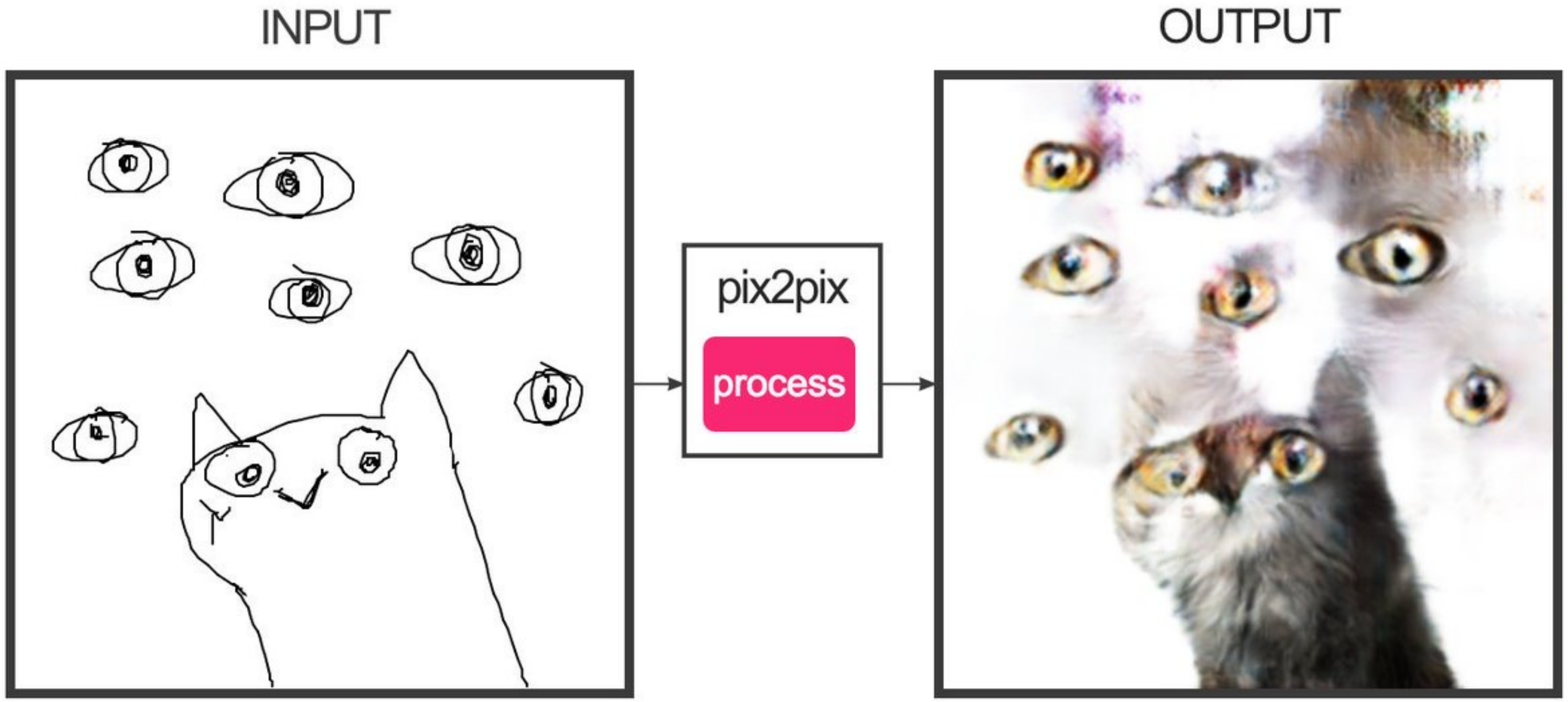
Image-to-Image Demo

Interactive Image Translation with pix2pix-tensorflow

Written by Christopher Hesse — February 19th, 2017

Recently, I made a Tensorflow port of pix2pix by Isola et al., covered in the article Image-to-Image Translation in Tensorflow. I've taken a few pre-trained models and made an interactive web thing for trying them out. Chrome is recommended.

The pix2pix model works by training on pairs of images such as building facade labels to building

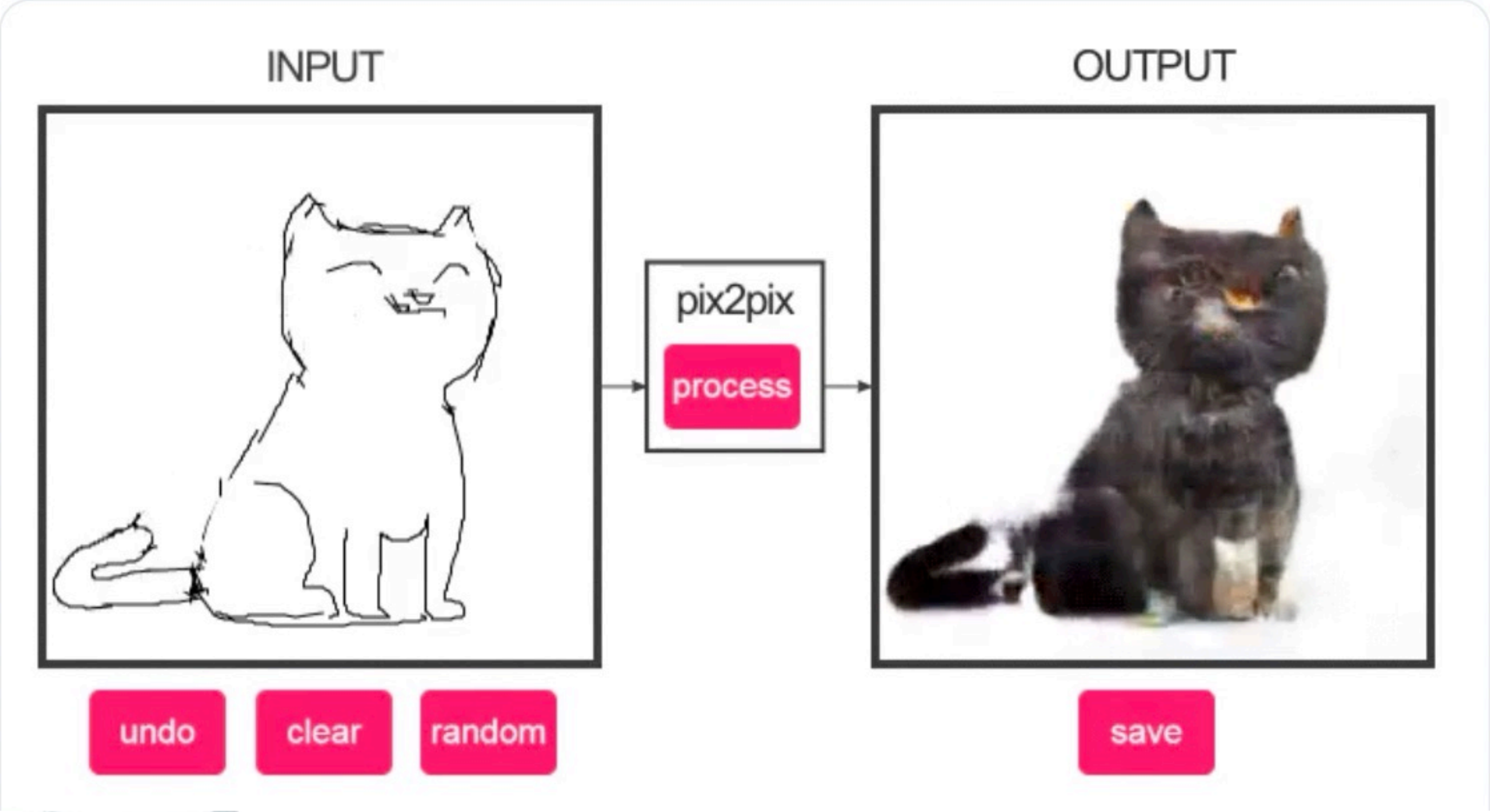


https://twitter.com/lauriepink/status/834941922070364161

Calamity 🐱ri-Warui
@oleivarrudi

Follow

i animated a cat (badly) and shoved it into affinelayer.com/pixsrv/ to make it even more catty
#pix2pix #edge2cats #NeuralNetworks #cat

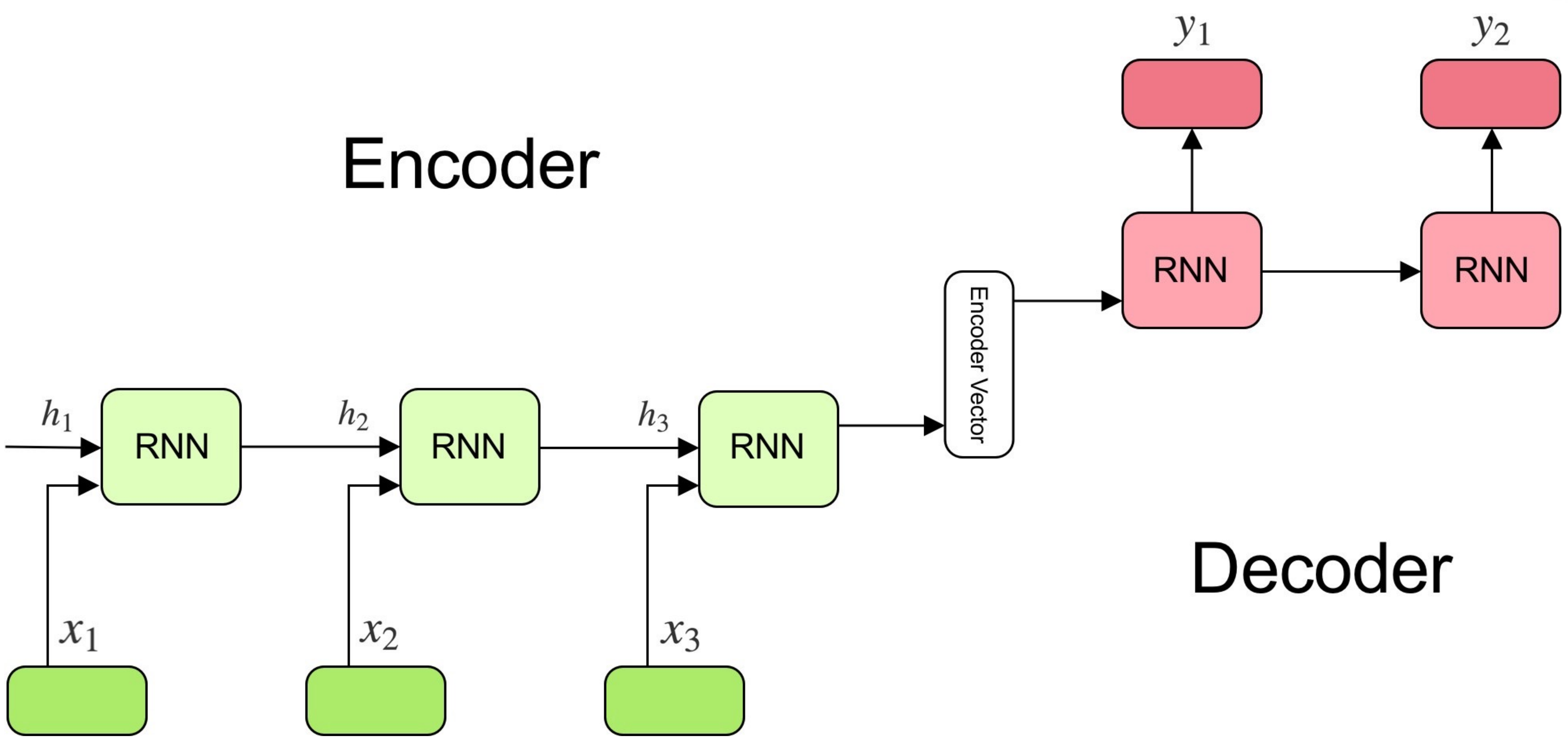


ı/lauriepink/status/8349419
22070364161

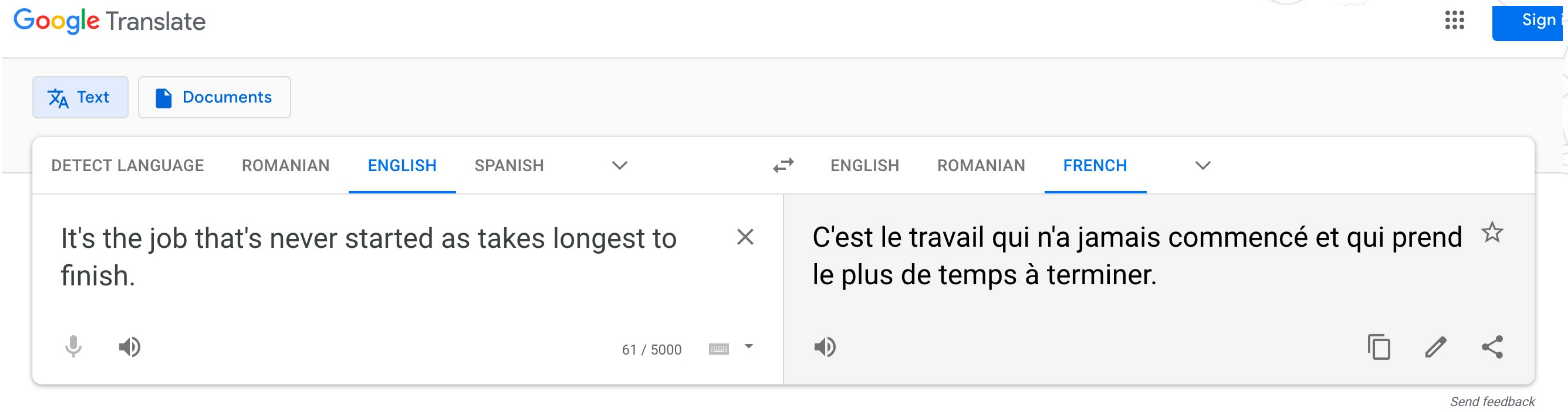
https://twitter.com/patricktresset/status/902107222792179713

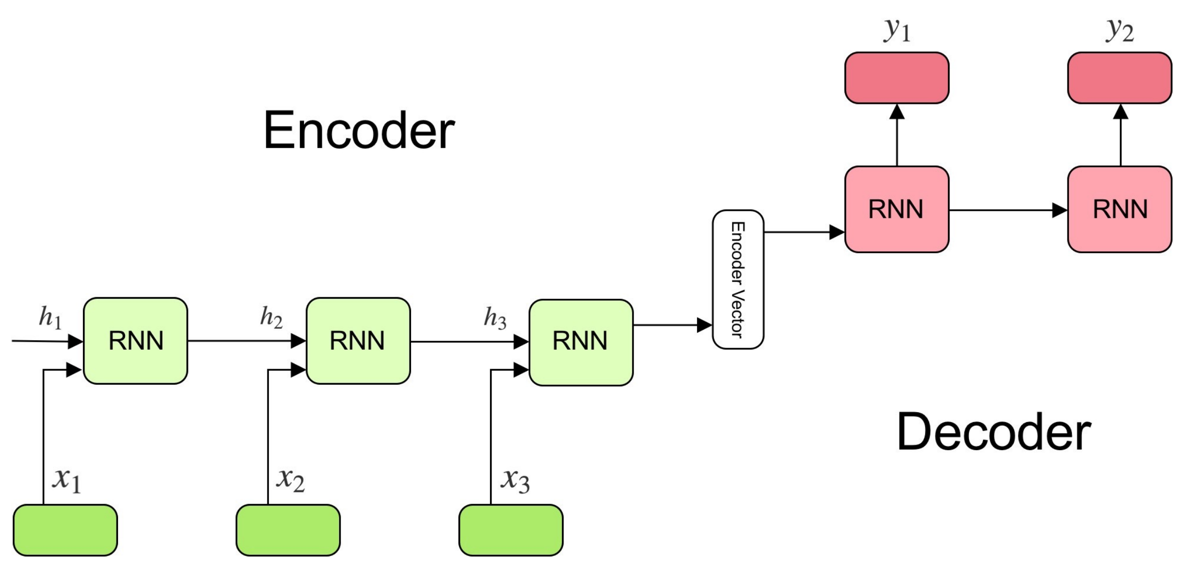
Encoder-decoder
Sequence to Sequence Model

Sutskever, I., Vinyals, O. and Le, Q.V., 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).

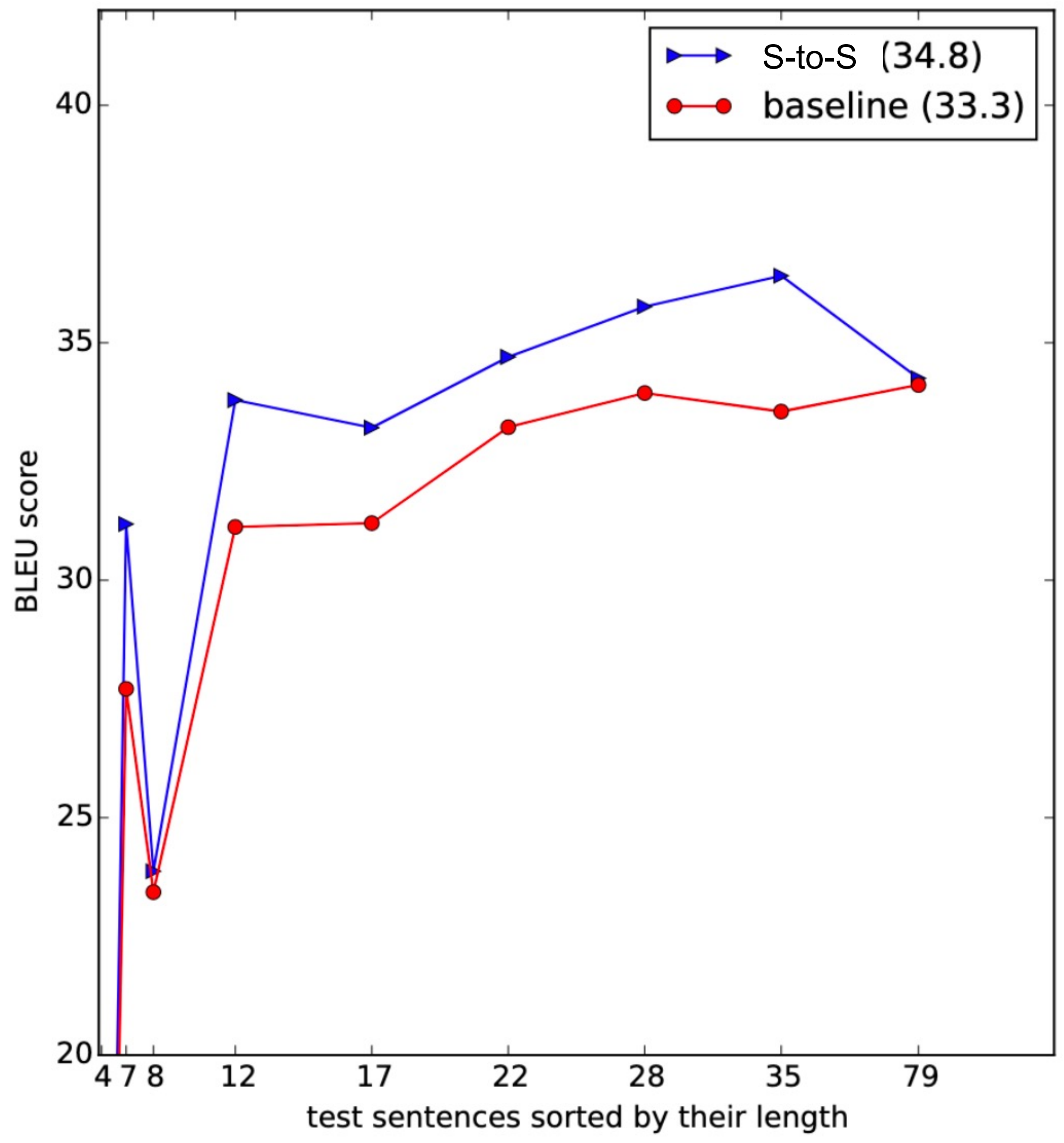


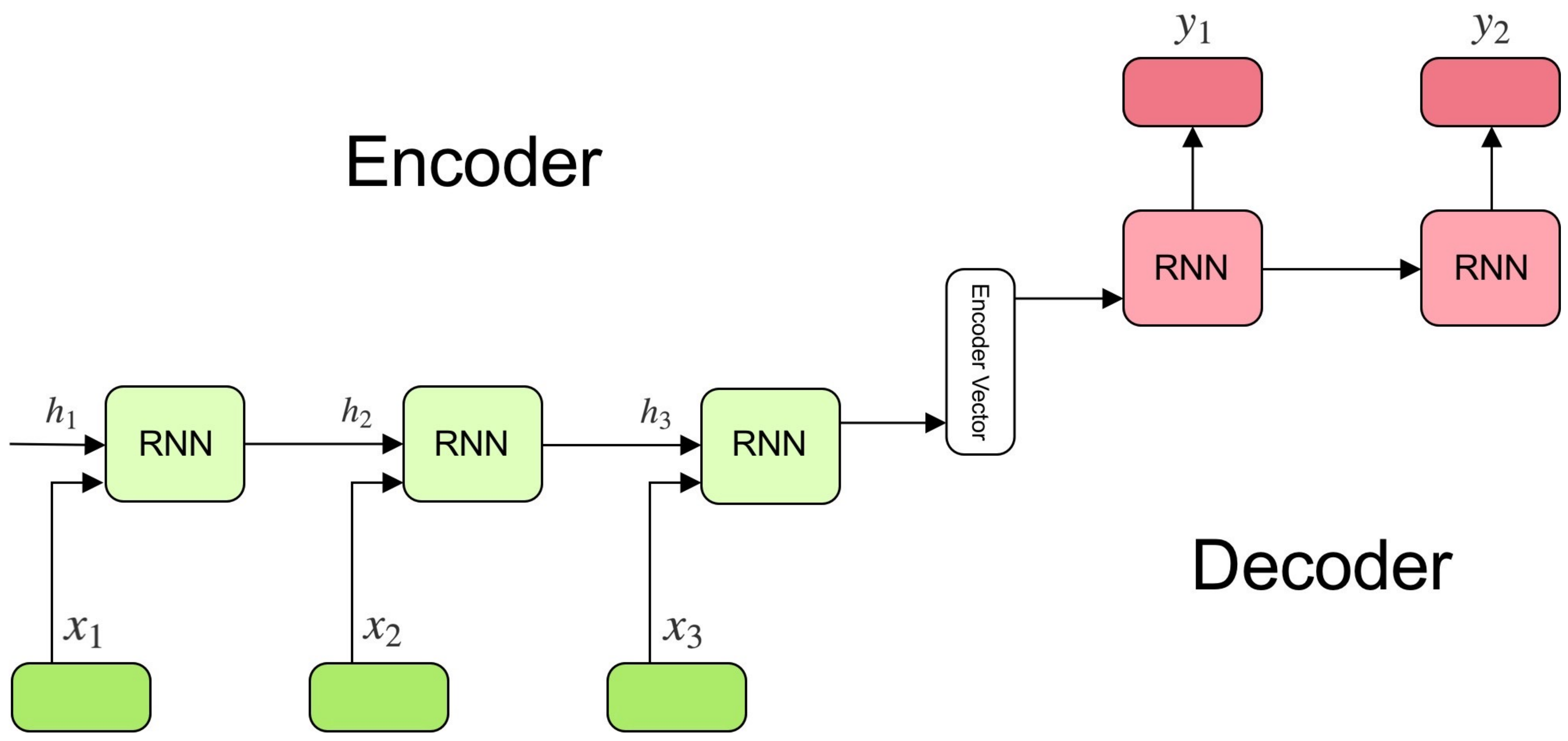
Sutskever, I., Vinyals, O. and Le, Q.V., 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).





Sutskever, I., Vinyals, O. and Le, Q.V., 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).





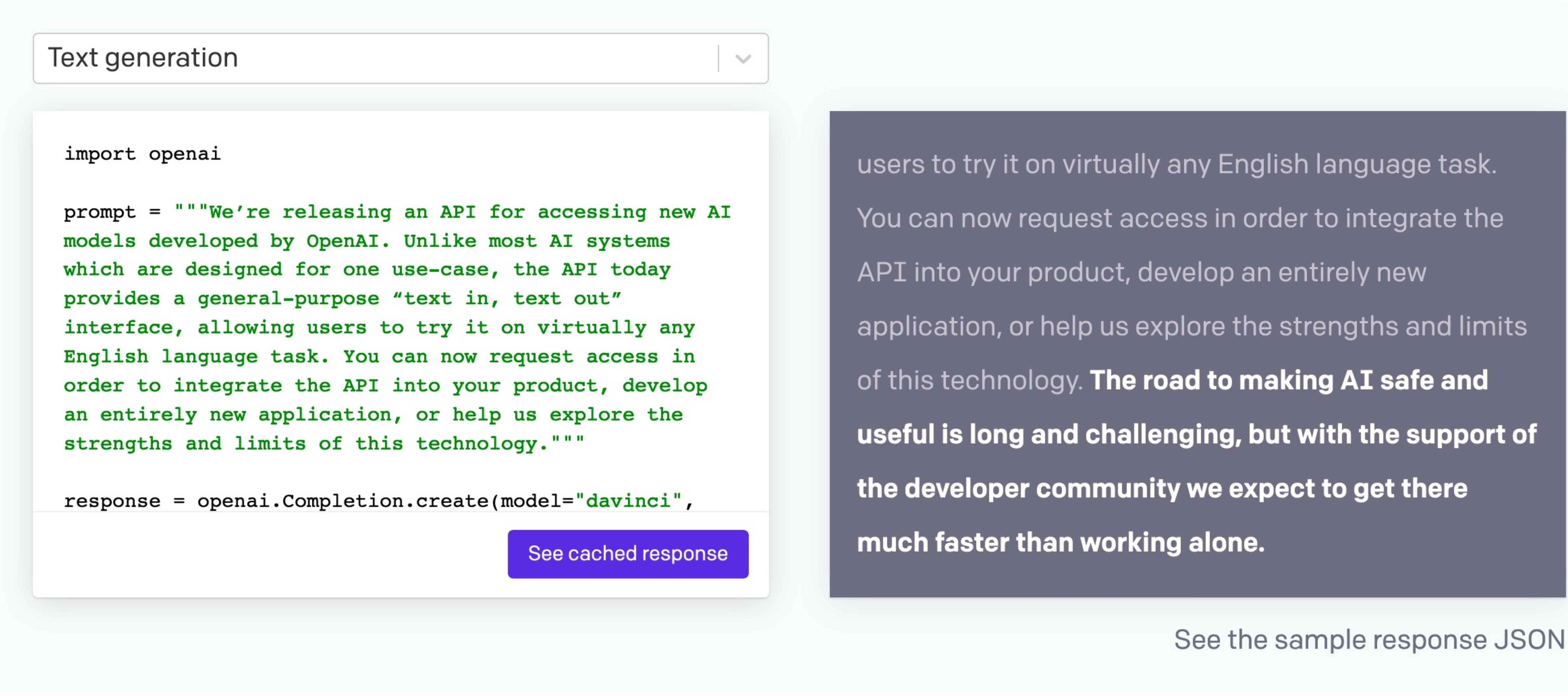
Transformer networks

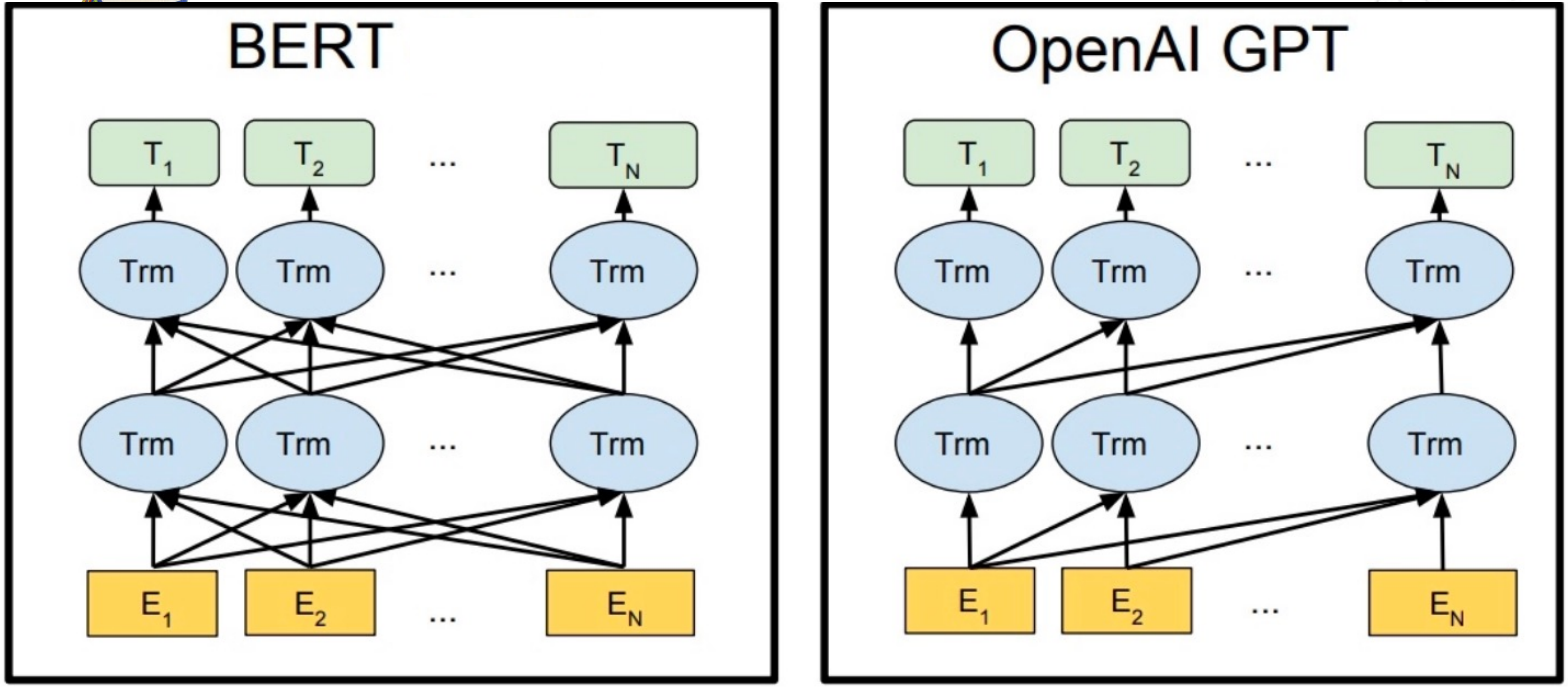
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com
Karthik Narasimhan OpenAI karthikn@openai.com
Tim Salimans OpenAI tim@openai.com
Ilya Sutskever OpenAI ilyasu@openai.com

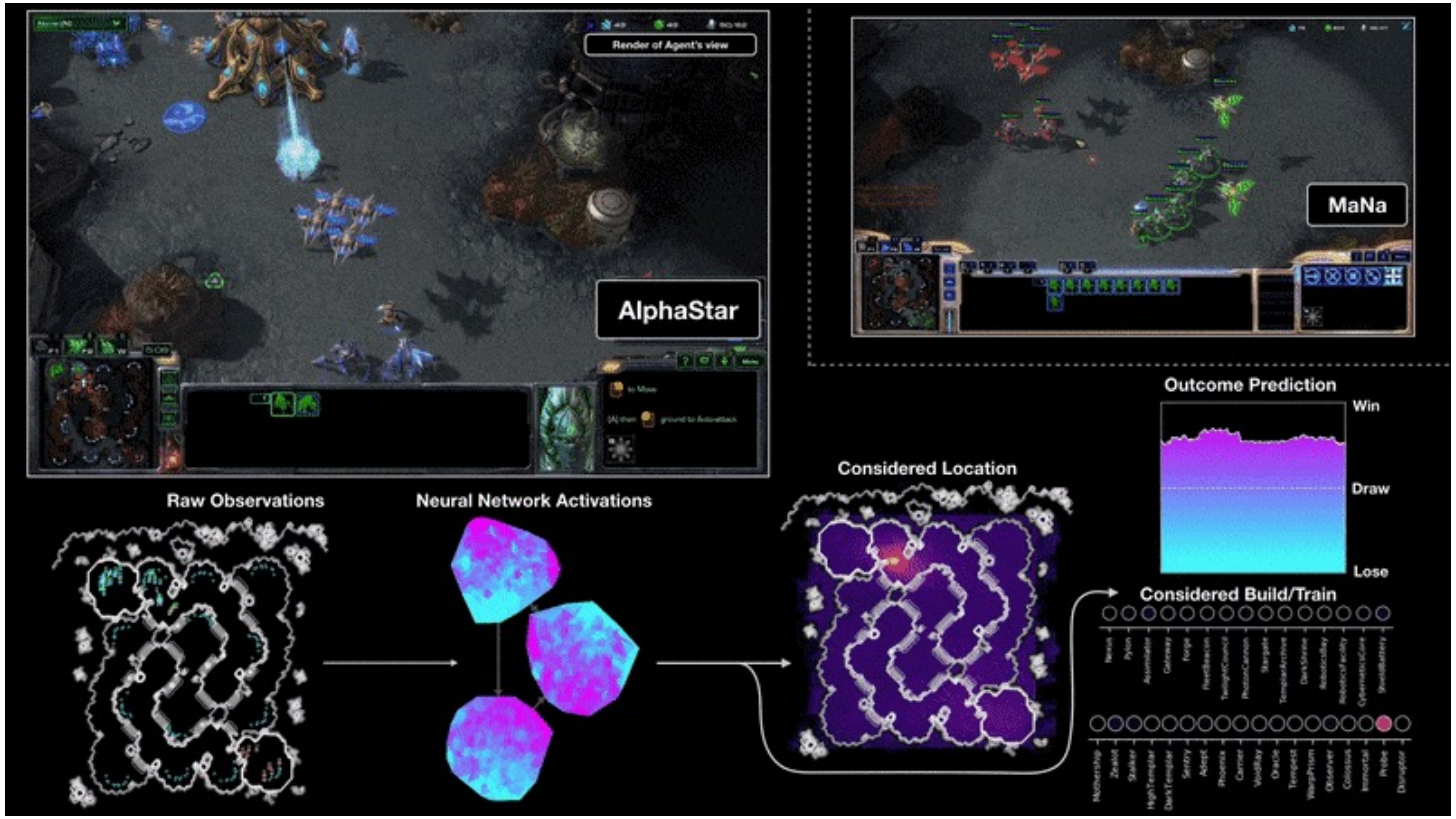




Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Deep Reinforcement learning

Deep Reinforcement learning



Laboratory

Image-to-Image Demo

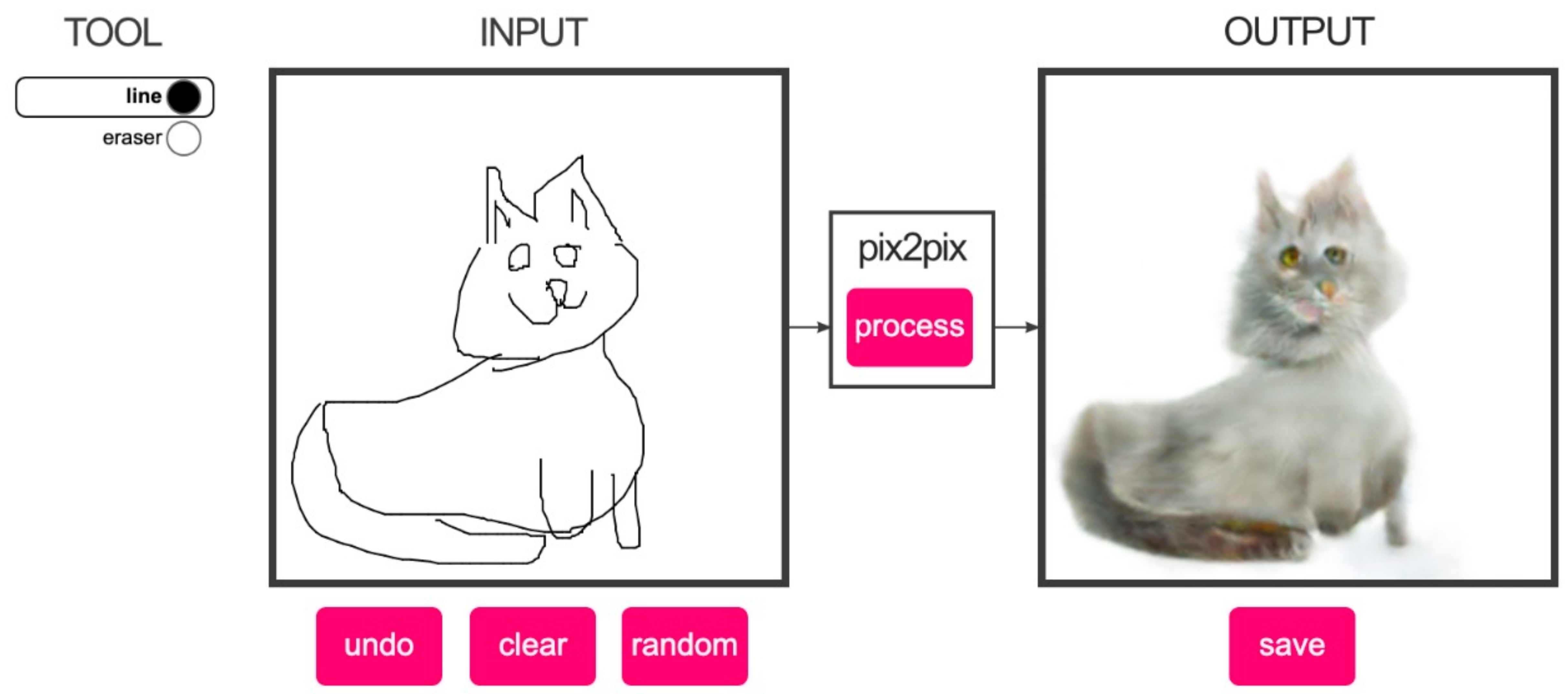
Interactive Image Translation with pix2pix-tensorflow

Written by Christopher Hesse — February 19th, 2017

Recently, I made a Tensorflow port of pix2pix by Isola et al., covered in the article Image-to-Image Translation in Tensorflow. I've taken a few pre-trained models and made an interactive web thing for trying them out. Chrome is recommended.

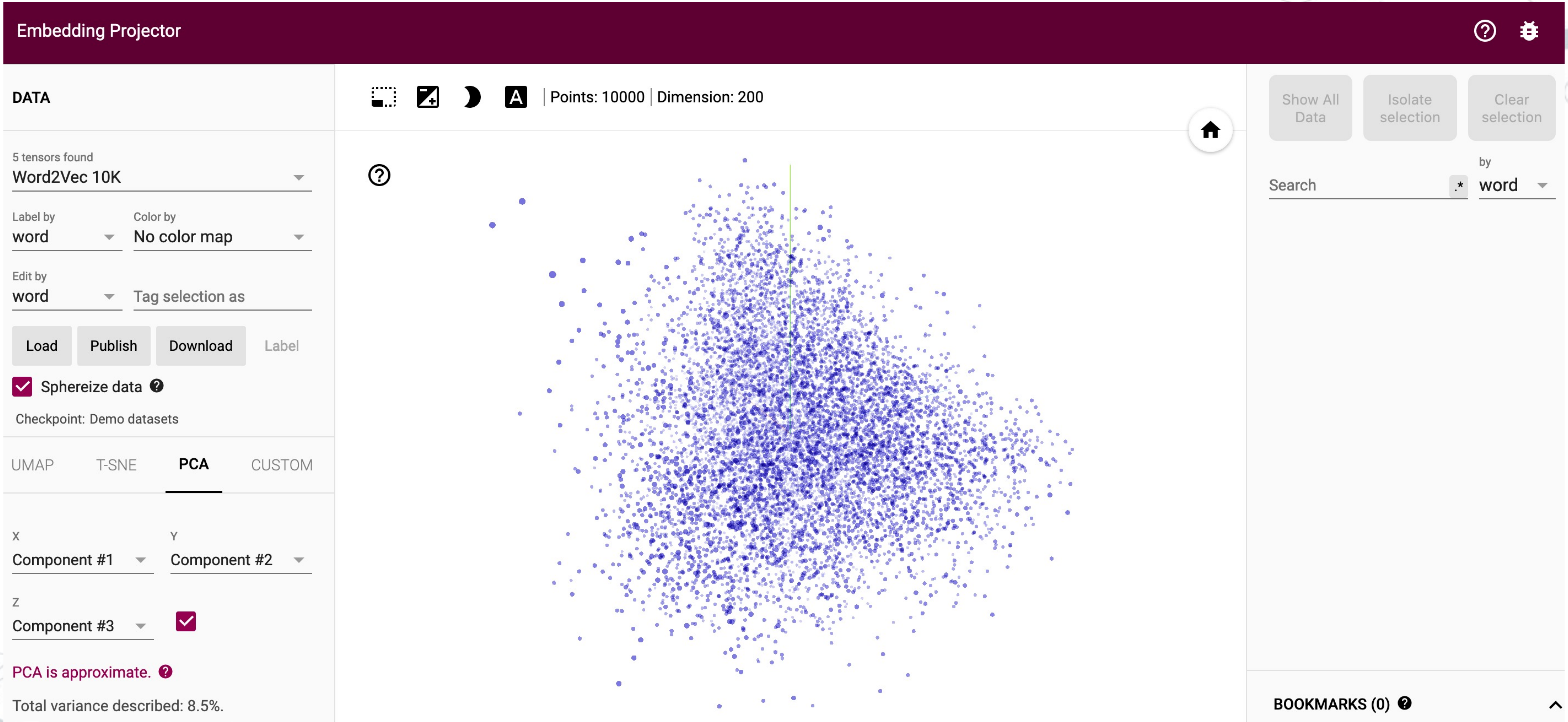
The pix2pix model works by training on pairs of images such as building facade labels to building facades, and then attempts to generate the corresponding output image from any input image you give it. The idea is straight from the pix2pix paper, which is a good read.

edges2cats



https://affinelayer.com/pixsrv/

Play with word2vec



Sentiment analysis

User Reviews

★★★★★★★★★★ **The Fellowship of the Ring: Not just a Movie, but the Door to another Dimension**
14 April 2006 | by bonnie91 – See all my reviews

The first part of the Lord of the Rings trilogy, the Fellowship of the Rings opened the door to a whole new world for me. I'd never read any of Tolkien's books when I saw the film for the first time at the theatre and, now that I've read them, in retrospect I think being a neophyte to the mythology made my LOTR movie experience all the more miraculous.

Positive

★ 4/10

Overlong epic
salomeuk 5 June 2004

I can see why people love this movie. Perhaps I'd feel the same if it came in under 2 hours. As it is, it was far too long. I had trouble staying awake, and one of my party left halfway through as she was so uncomfortable sitting so long in the cinema. That's something I've never know happen before or since!

I can't really comment on the movie itself too much, as I'm not sure I've actually seen it all....sat through it, yes. But much of it didn't register after the first half hour.

Negative

Thank you!