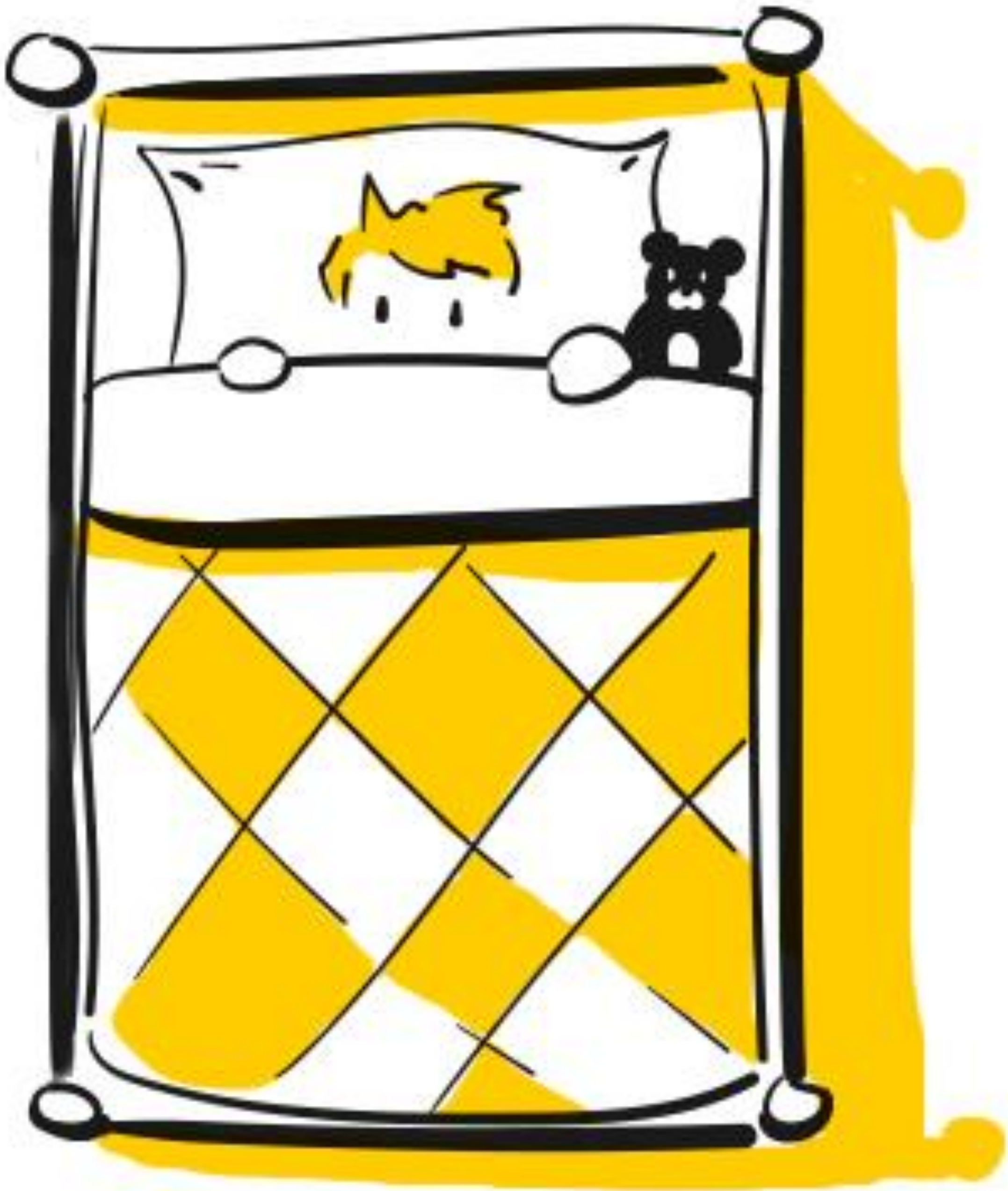
MAI Deep Learning

Embeddings

Dario Garcia Gasulla *dario.garcia@bsc.es*
Armand Vilalta

Embeddings

The idea



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Embeddings

The idea

- To firmly place something in a surrounding mass or environment
- To make something an integral part of a larger whole.

Embeddings

The maths

- In general topology, an embedding is a homeomorphism onto its image.
- More explicitly, an injective continuous map $\boldsymbol{f}: \boldsymbol{X} \to \boldsymbol{Y}$ between topological spaces $\boldsymbol{X}$ and $\boldsymbol{Y}$ is a **topological embedding** if $\boldsymbol{f}$ yields a **homeomorphism** between $\boldsymbol{X}$ and $\boldsymbol{f}(\boldsymbol{X})$.

Embeddings

The maths

- **Homeomorphism** is a continuous function between topological spaces that has a continuous inverse function.
  - $\boldsymbol{f}$ is a bijection (one-to-one and onto)
  - $\boldsymbol{f}$ is continuous
  - The inverse function $\boldsymbol{f^{-1}}$ is continuous

Embeddings

The maths

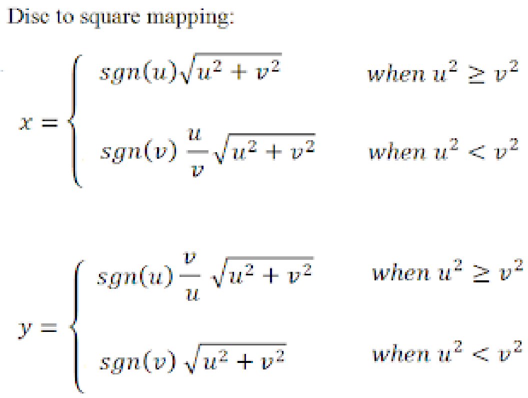
Examples of homeomorphism:

- The open interval (a, b) for any $a < b$ is homeomorphic to $\mathbb{R}$

$$f(x) = \frac{1}{a - x} + \frac{1}{b - x}$$

Embeddings

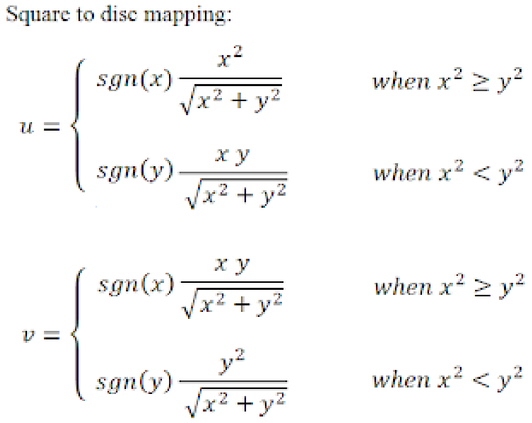
The maths



Disc to square mapping:

$$x = \begin{cases} sgn(u)\sqrt{u^2+v^2} & when\ u^2 \geq v^2 \\ sgn(v)\dfrac{u}{v}\sqrt{u^2+v^2} & when\ u^2 < v^2 \end{cases}$$

$$y = \begin{cases} sgn(u)\dfrac{v}{u}\sqrt{u^2+v^2} & when\ u^2 \geq v^2 \\ sgn(v)\sqrt{u^2+v^2} & when\ u^2 < v^2 \end{cases}$$



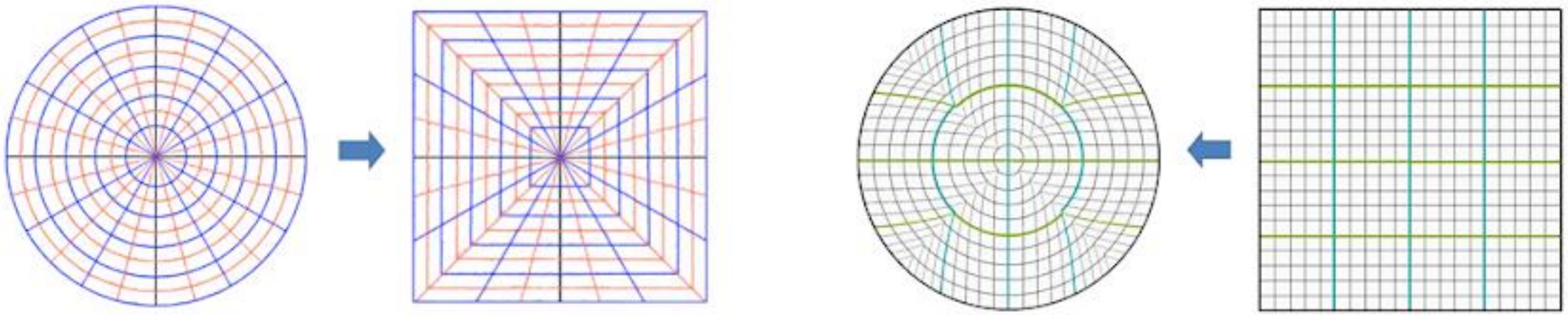
Square to disc mapping:

$$u = \begin{cases} sgn(x)\dfrac{x^2}{\sqrt{x^2+y^2}} & when\ x^2 \geq y^2 \\ sgn(y)\dfrac{x\,y}{\sqrt{x^2+y^2}} & when\ x^2 < y^2 \end{cases}$$

$$v = \begin{cases} sgn(x)\dfrac{x\,y}{\sqrt{x^2+y^2}} & when\ x^2 \geq y^2 \\ sgn(y)\dfrac{y^2}{\sqrt{x^2+y^2}} & when\ x^2 < y^2 \end{cases}$$

Examples of homeomorphism:

- The open interval (a, b) for any $a < b$ is homeomorphic to $\mathbb{R}$
- The unit 2-disc D^2 and the unit square in $\mathbb{R}^2$



Embeddings

The maths

Examples of homeomorphism:

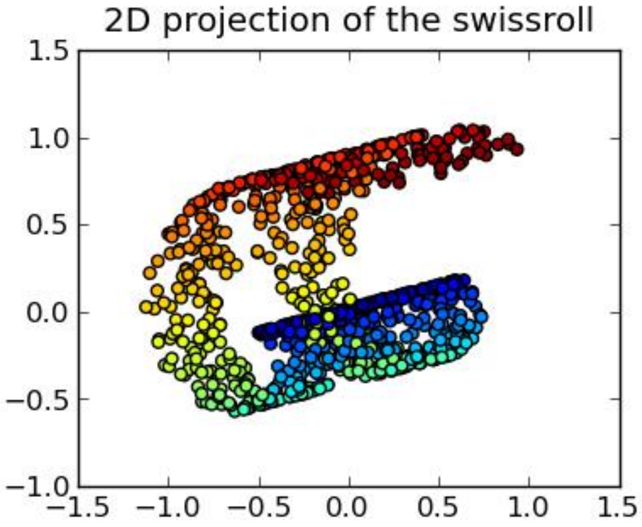
- The open interval (a, b) for any $a < b$ is homeomorphic to $\mathbb{R}$
- The unit 2-disc D^2 and the unit square in $\mathbb{R}^2$

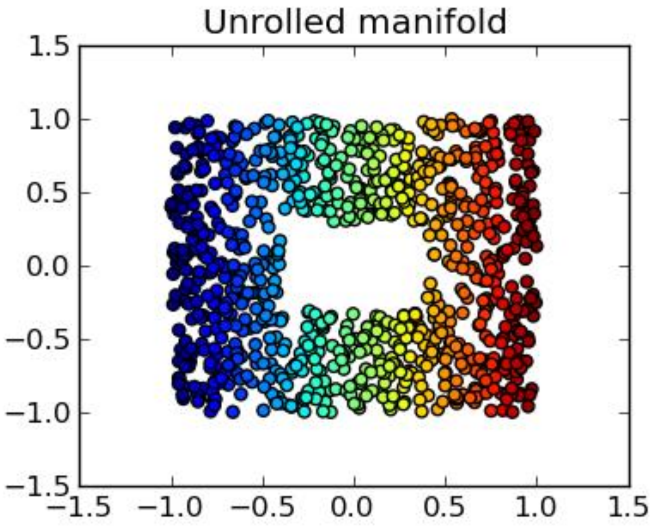
Examples of NOT homeomorphism:

- $\mathbb{R}^m$ and $\mathbb{R}^n$ are not homeomorphic for $m \neq n$
- The Euclidean real line in not homeomorphic to the unit circle as a subspace of $\mathbb{R}^2$ (line is not compact, circle is)

Embeddings

The maths
The manifold hypothesis





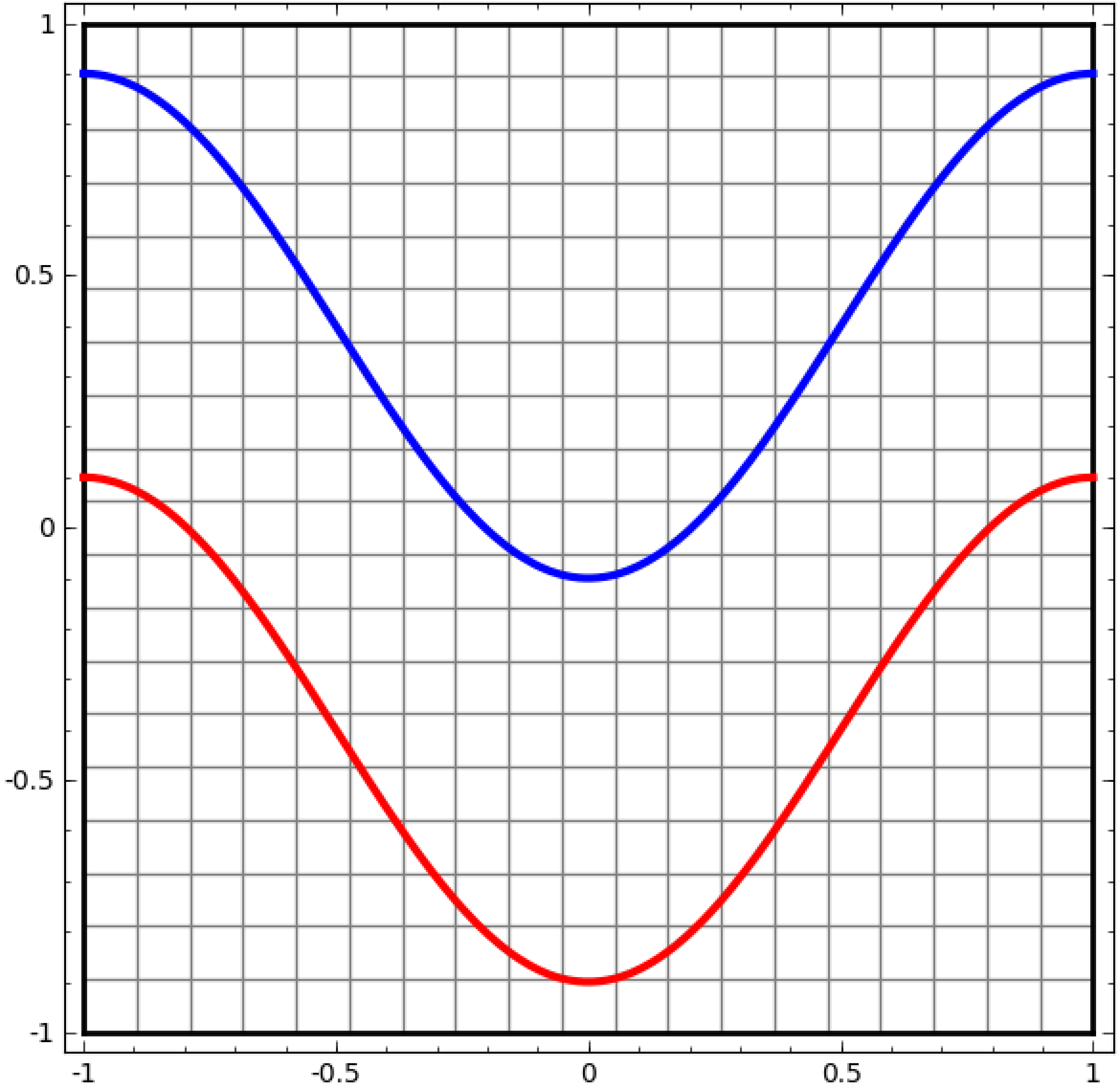
The **manifold hypothesis** is that **natural data** forms **lower-dimensional manifolds** in its embedding space.

There are both theoretical and experimental reasons to believe this to be true.

If you believe this, then the task of a **classification** algorithm is fundamentally to **separate a bunch of tangled manifolds**.

Embeddings

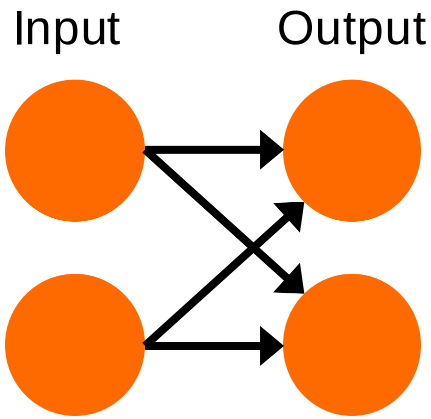
The maths
The neural networks

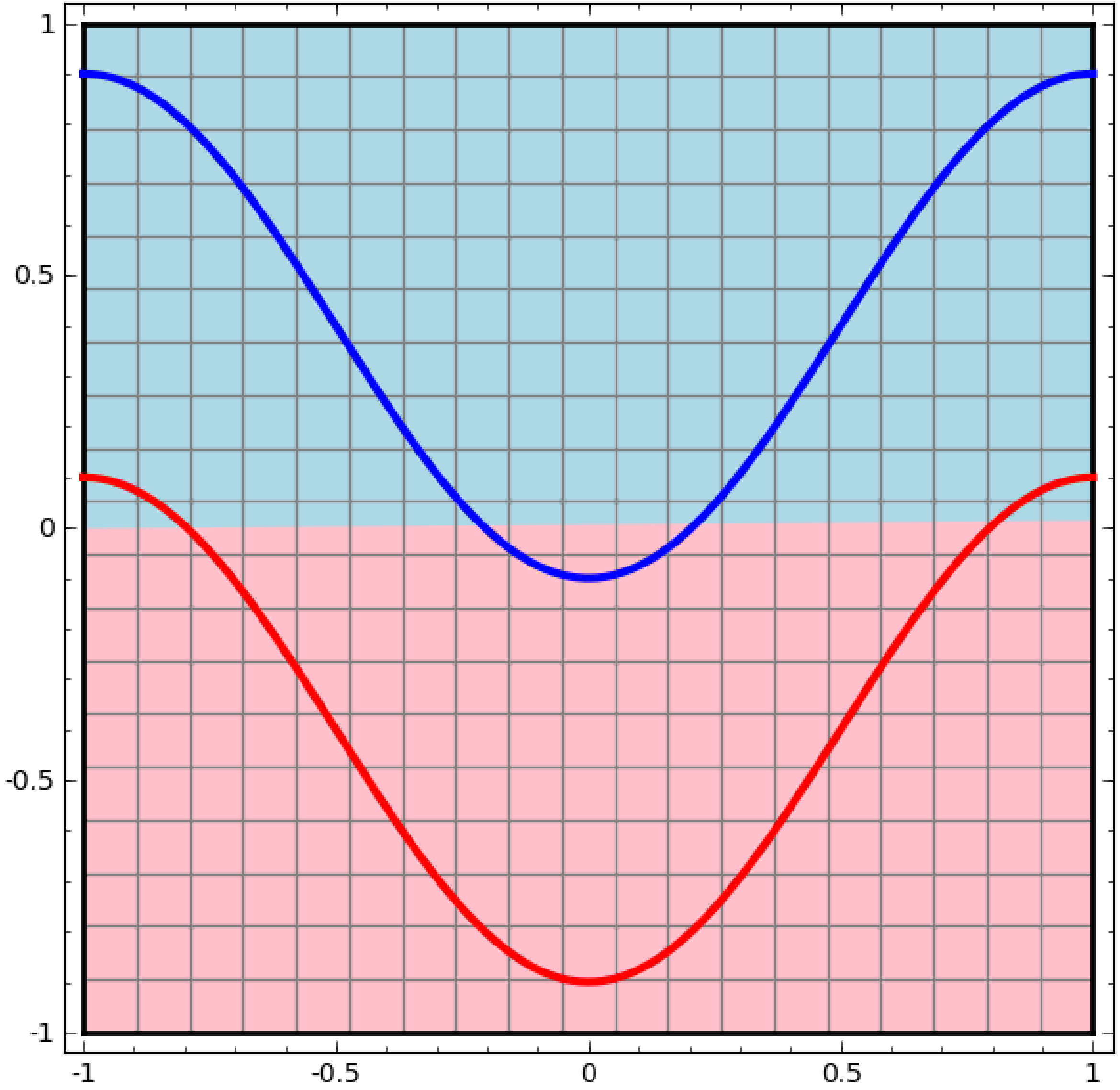


http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Embeddings

The maths
The neural networks

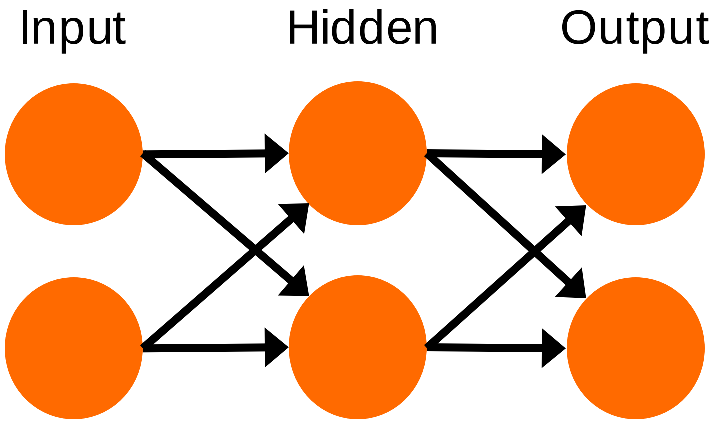


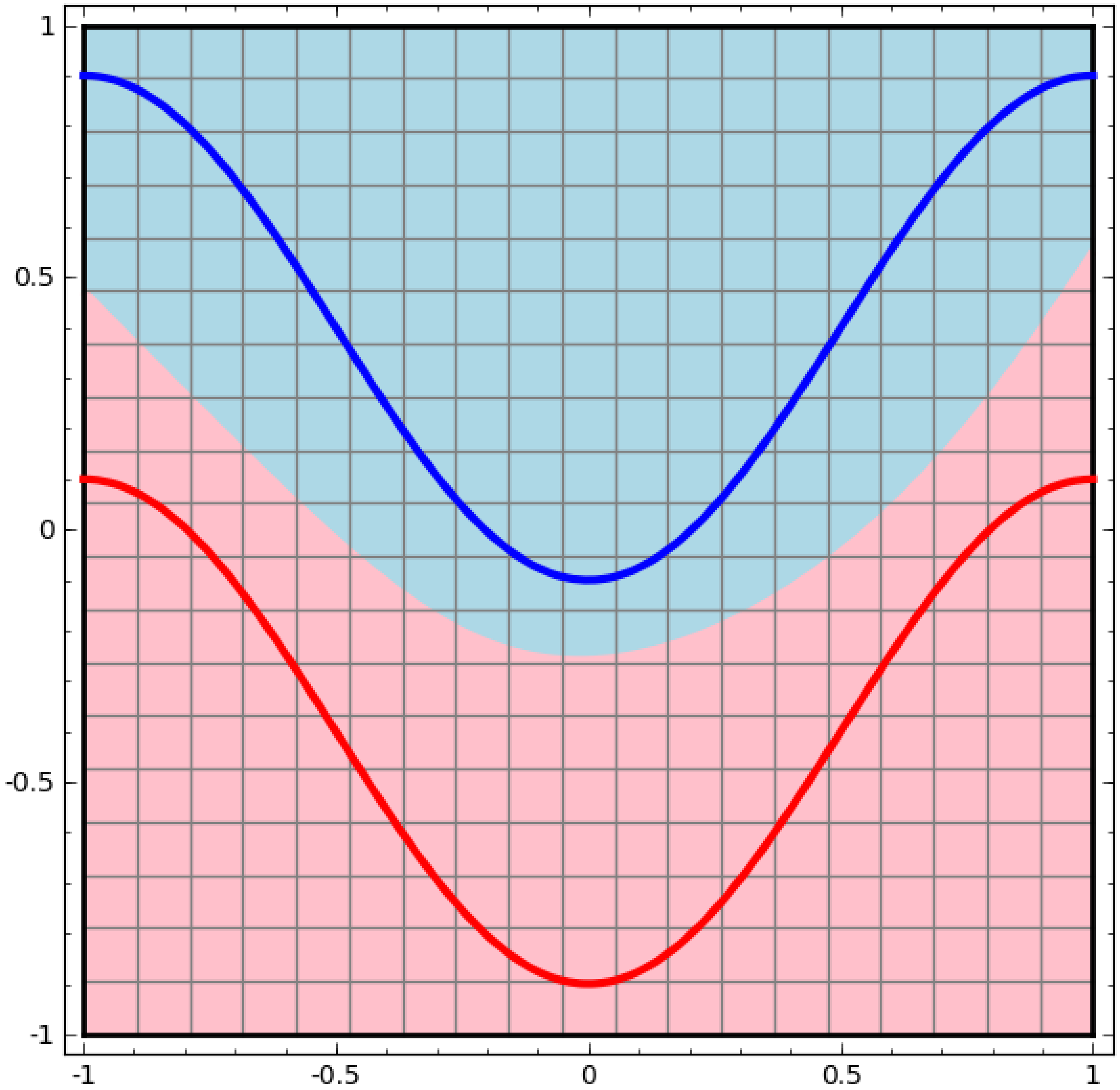


Embeddings

The maths

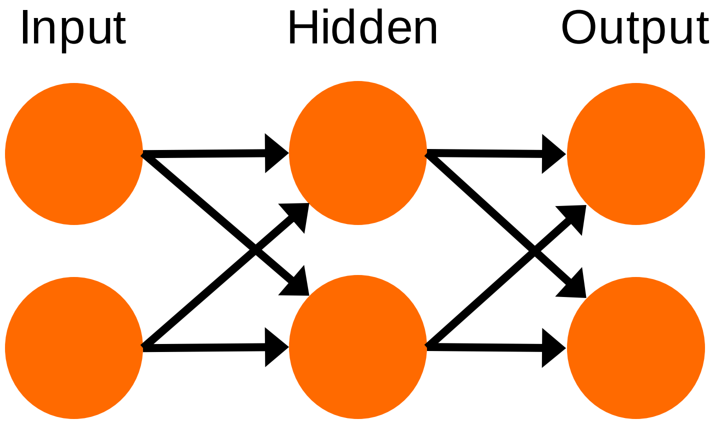
The neural networks

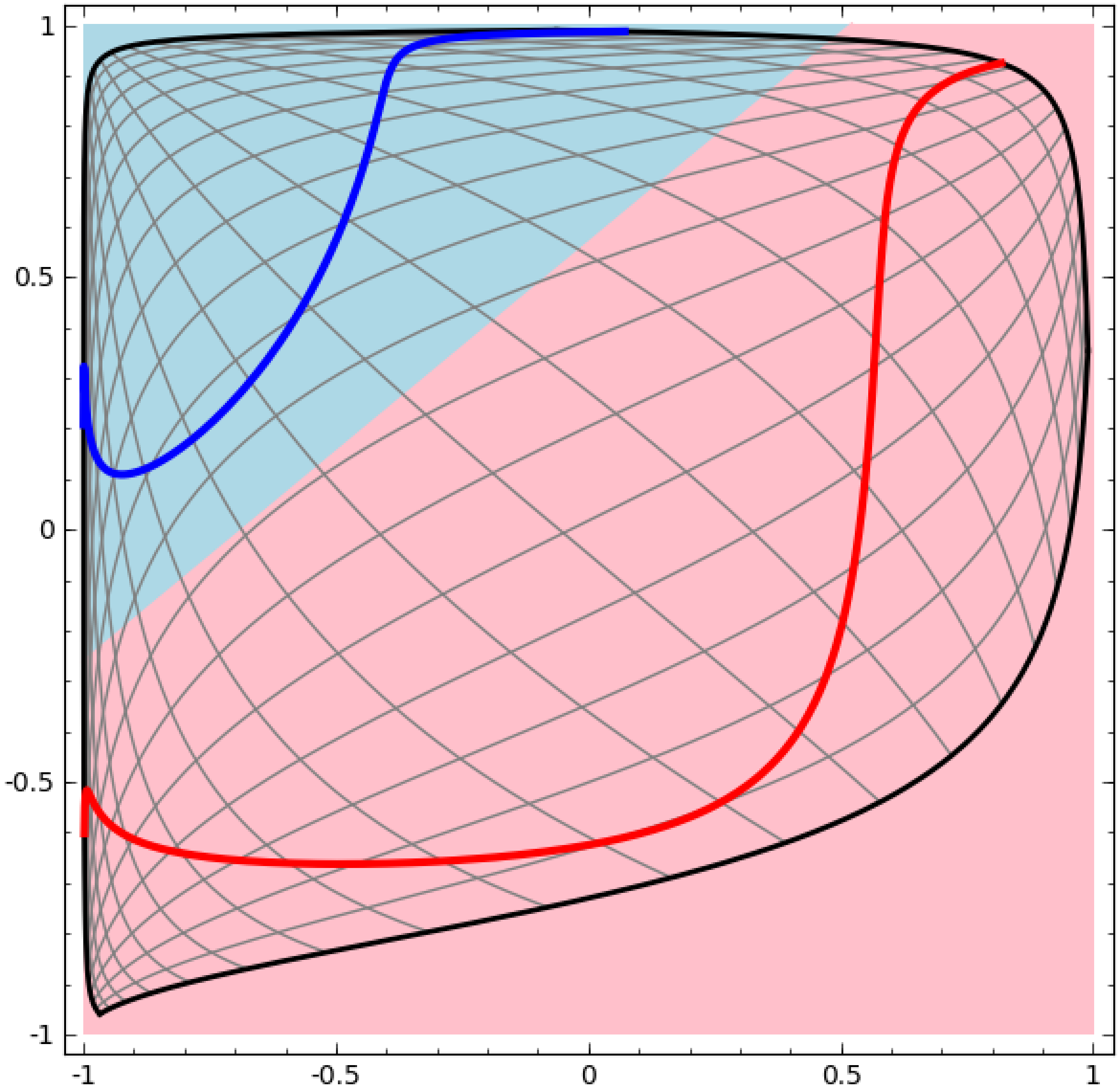




Embeddings

The maths
The neural networks

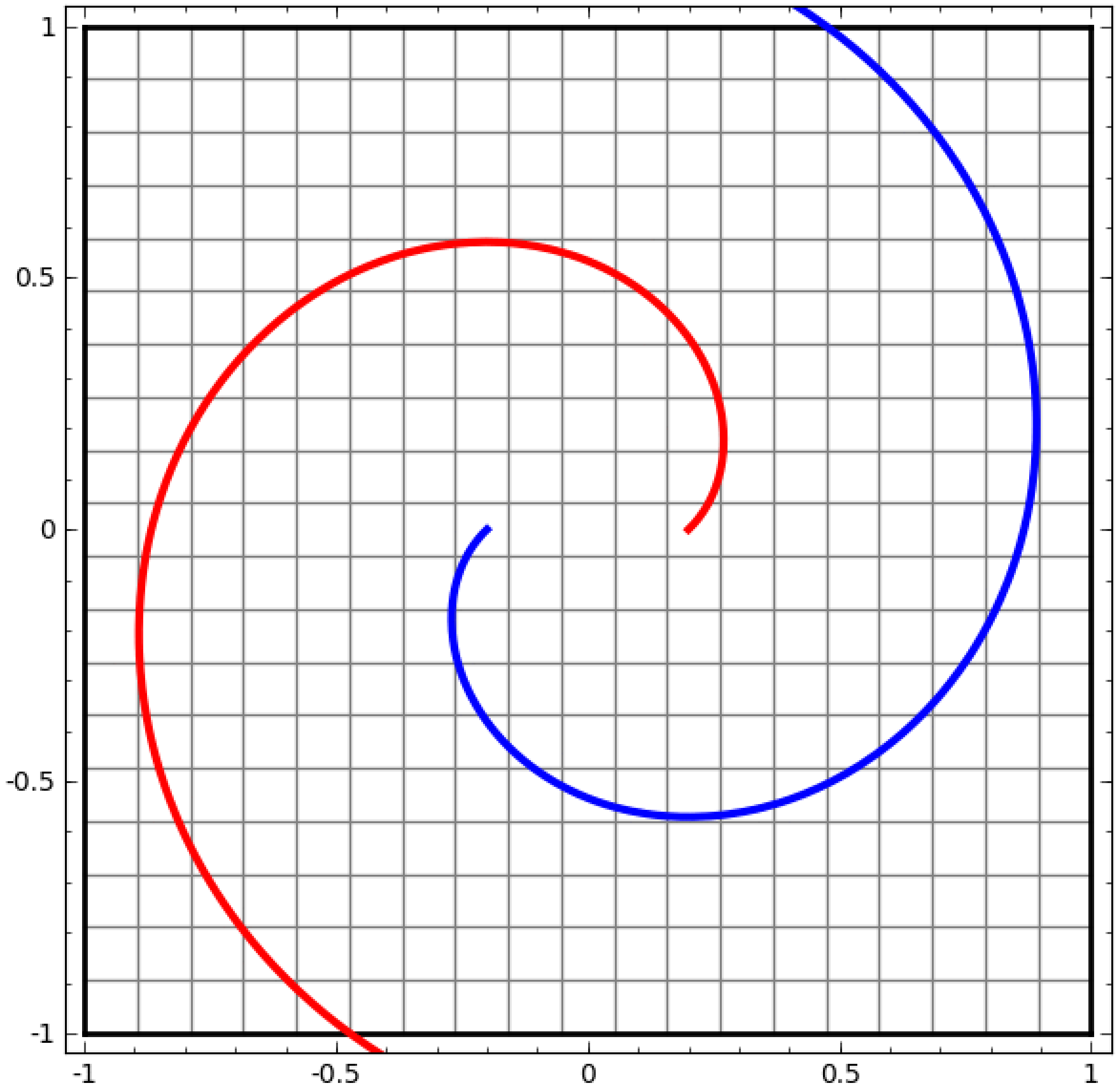




Embeddings

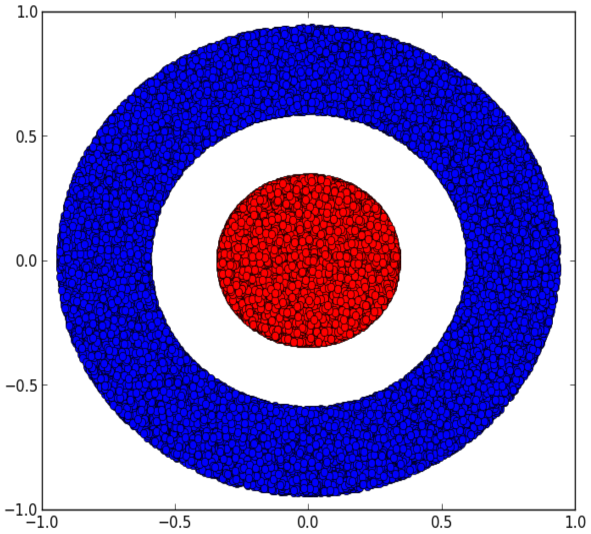
The maths
The neural networks

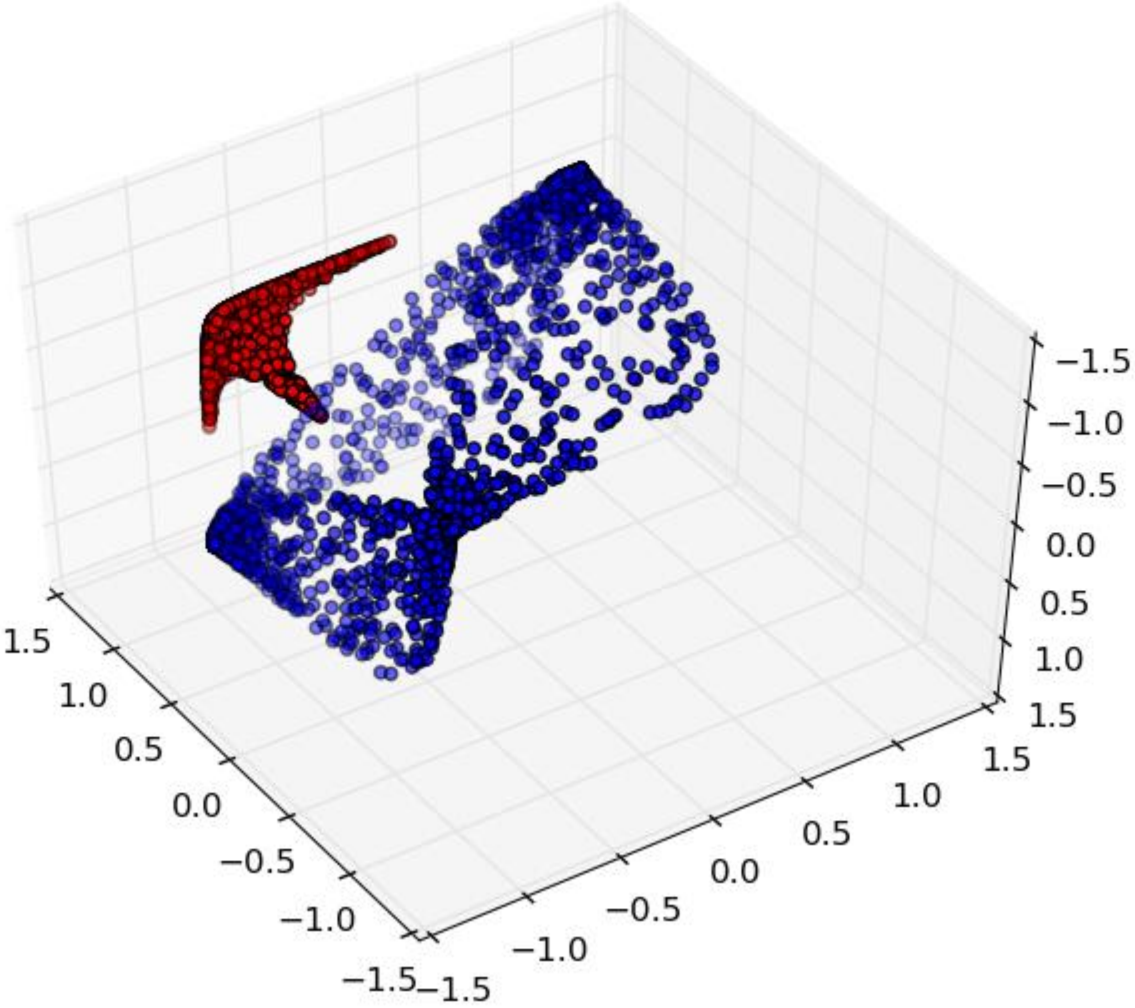
Classifying entangled spirals using 4 hidden layers



Embeddings

The maths
The neural networks





Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Embeddings

The maths

- In general topology, an embedding is a homeomorphism onto its image.
- More explicitly, an injective continuous map $\boldsymbol{f}\colon \boldsymbol{X} \to \boldsymbol{Y}$ between topological spaces $\boldsymbol{X}$ and $\boldsymbol{Y}$ is a **topological embedding** if $\boldsymbol{f}$ yields a **homeomorphism** between $\boldsymbol{X}$ and $\boldsymbol{f}(\boldsymbol{X})$.

Embeddings

Some references

- Colah’s blog about NN topology:

http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

- Andrej Karpathy tool to visualize NN embeddings:

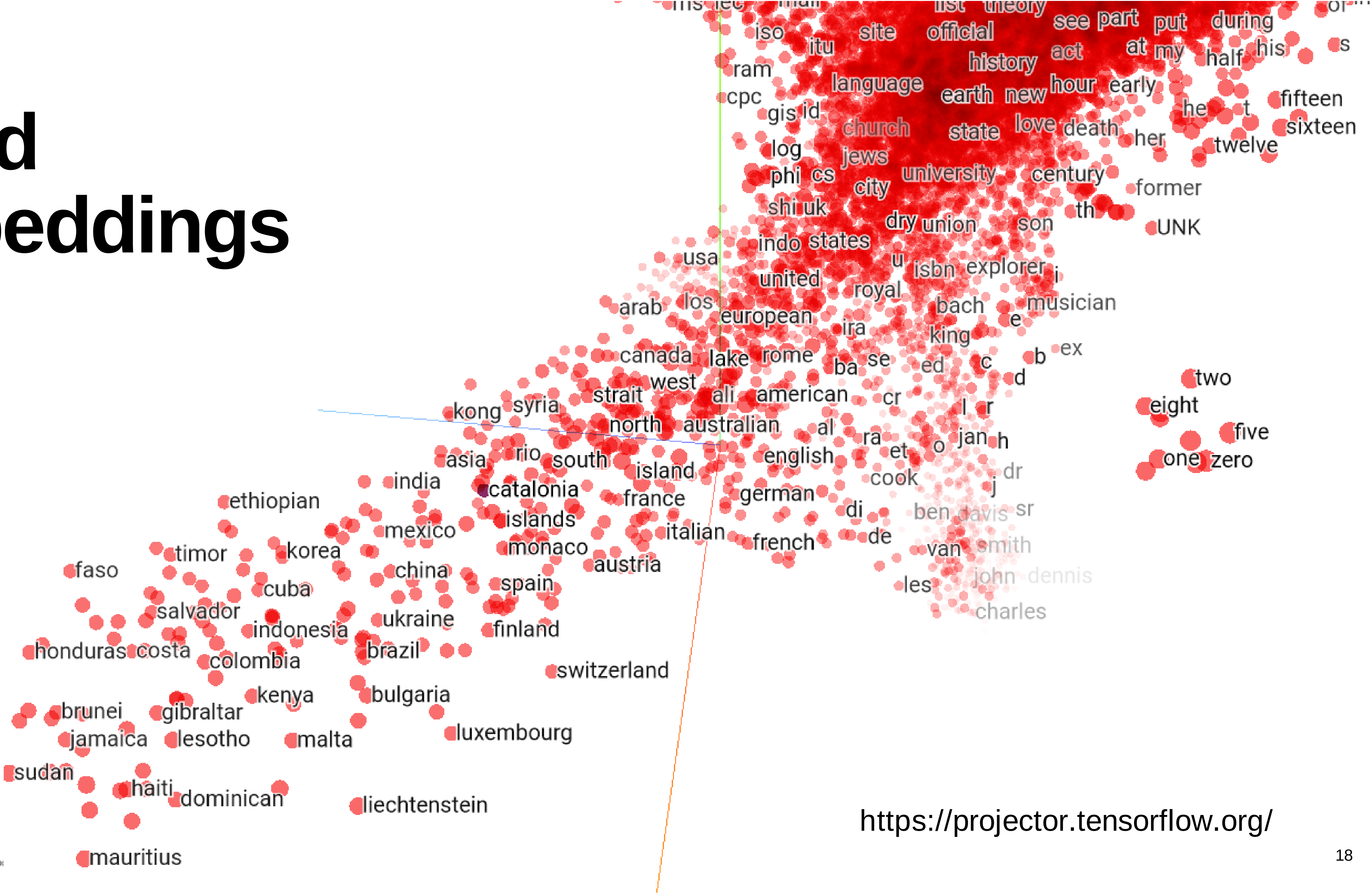
https://cs.stanford.edu/people/karpathy/convnetjs//demo/classify2d.html

- Mathematical articles on the manifold hypothesis:

http://www.mit.edu/~mitter/publications/121_Testing_Manifold.pdf

http://www.ams.org/journals/bull/2009-46-02/S0273-0979-09-01249-X/S0273-0979-09-01249-X.pdf

Word embeddings



https://projector.tensorflow.org/

Word embeddings

How can we represent words?

Word embeddings

How can we represent words?

- **One-hot** vector embedding

	1	2	3	4	5	6	7	8
man	1	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0
girl	0	0	0	1	0	0	0	0
prince	0	0	0	0	1	0	0	0
princess	0	0	0	0	0	1	0	0
queen	0	0	0	0	0	0	1	0
king	0	0	0	0	0	0	0	1

Word embeddings

How can we represent words?

- **One-hot** vector embedding

	1	2	3	4	5	6	7	8
man	1	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0
girl	0	0	0	1	0	0	0	0
prince	0	0	0	0	1	0	0	0
princess	0	0	0	0	0	1	0	0
queen	0	0	0	0	0	0	1	0
king	0	0	0	0	0	0	0	1

Word embeddings

How can we represent words?

- **One-hot** vector embedding

	1	2	3	4	5	6	7	8
man	1	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0
girl	0	0	0	1	0	0	0	0
prince	0	0	0	0	1	0	0	0
princess	0	0	0	0	0	1	0	0
queen	0	0	0	0	0	0	1	0
king	0	0	0	0	0	0	0	1

- Simple
- Each word is a new dimension

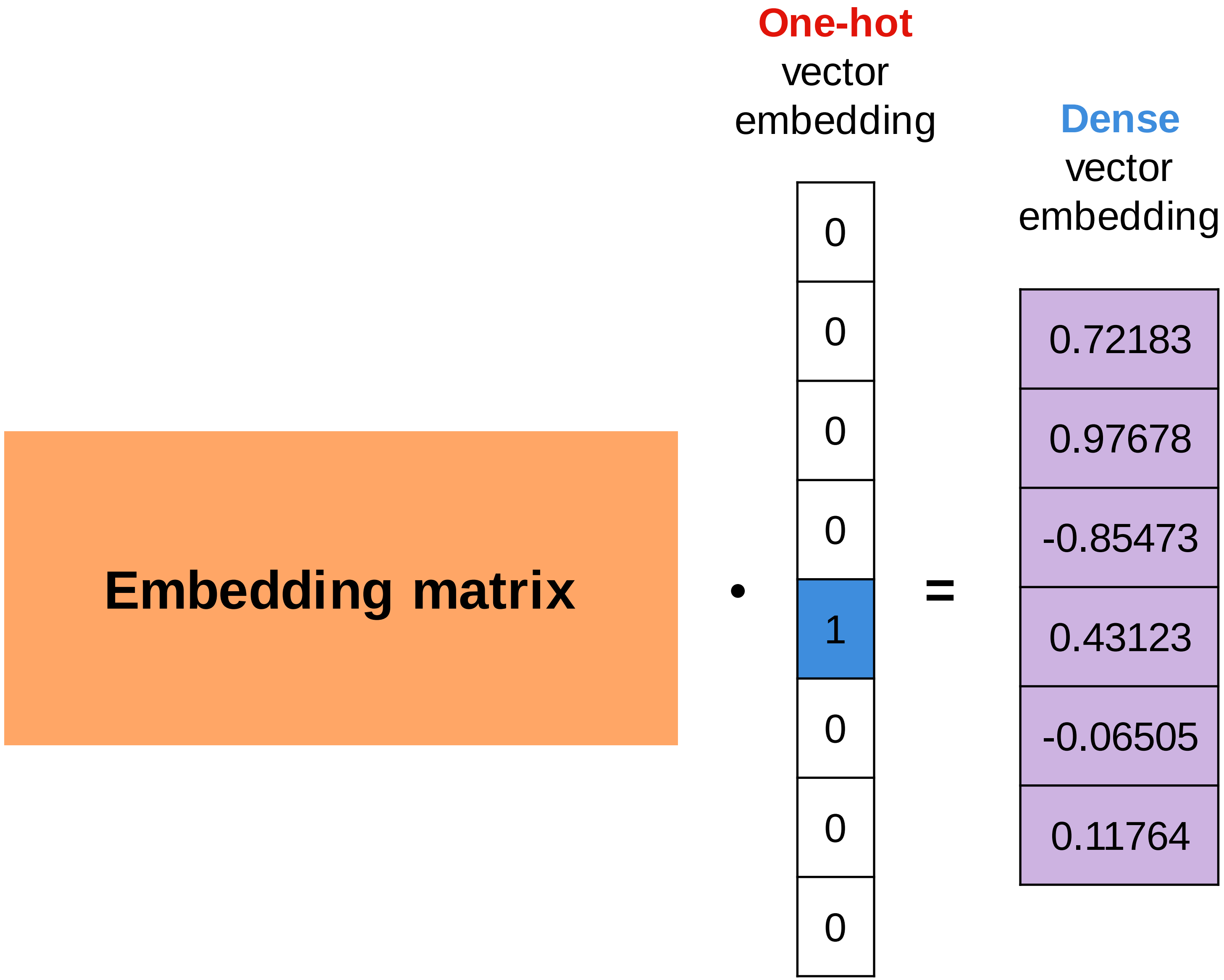
➔ high dimensionality

- Semantics are uncorrelated / orthogonal

Word embeddings

How can we represent words?

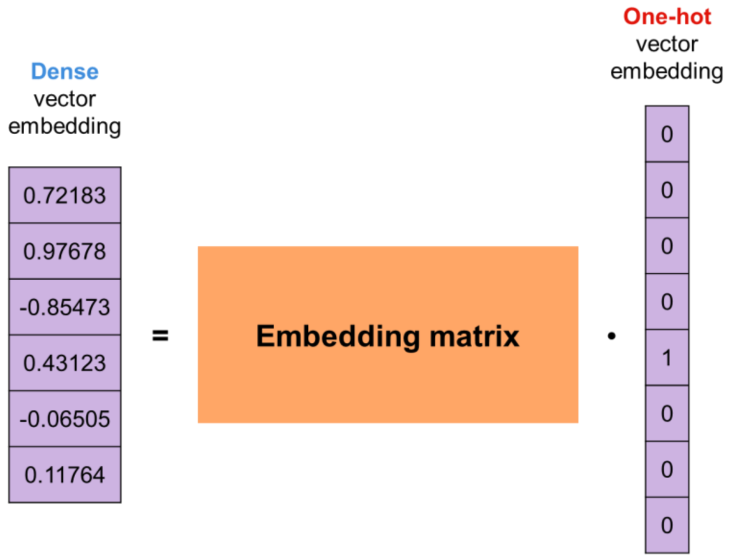
- **One-hot** vector embedding
- **Dense** vector embedding.



Word embeddings

How can we represent words?

- One-hot vector embedding
- Dense vector embedding.



- Needs an **appropriate** embedding matrix
- Reduced dimensionality compared to vocabulary (~300 dimensions)
- Semantics are correlated!

Word embeddings

Learning the embedding matrix

Semantics are correlated!

Word embeddings

Learning the embedding matrix

Semantics are correlated!

- Needs an appropriate embedding matrix where **spatial relations** between embedded words mimic **semantic relations** between words.

Word embeddings

Learning the embedding matrix

Semantics are correlated!

- Needs an appropriate embedding matrix where **spatial relations** between embedded words mimic **semantic relations** between words.
- Since this is deep learning course we would like to **learn this matrix** (also because defining it manually can be a humongous task).

Word embeddings

Learning the embedding matrix

Semantics are correlated!

- Needs an appropriate embedding matrix where **spatial relations** between embedded words mimic **semantic relations** between words.
- Since this is deep learning course we would like to **learn this matrix** (also because defining it manually can be a humongous task).
- So, we need a **task** to solve that **requires** a **semantic** representation of the embedding.

Word embeddings

Learning the embedding matrix

Semantics are correlated!

- Needs an appropriate embedding matrix where **spatial relations** between embedded words mimic **semantic relations** between words.
- Since this is deep learning course we would like to **learn this matrix** (also because defining it manually can be a humongous task).
- So, we need a **task** to solve that **requires** a **semantic** representation of the embedding.
- When we use the embedding in **another task** we are effectively doing **transfer learning.**

Word embeddings

Learning the embedding matrix

So, which is the task?

Word embeddings

Learning the embedding matrix

So, which is the task?

In general,

predict word sequences!

Word embeddings

Word2vec
Skip-gram model

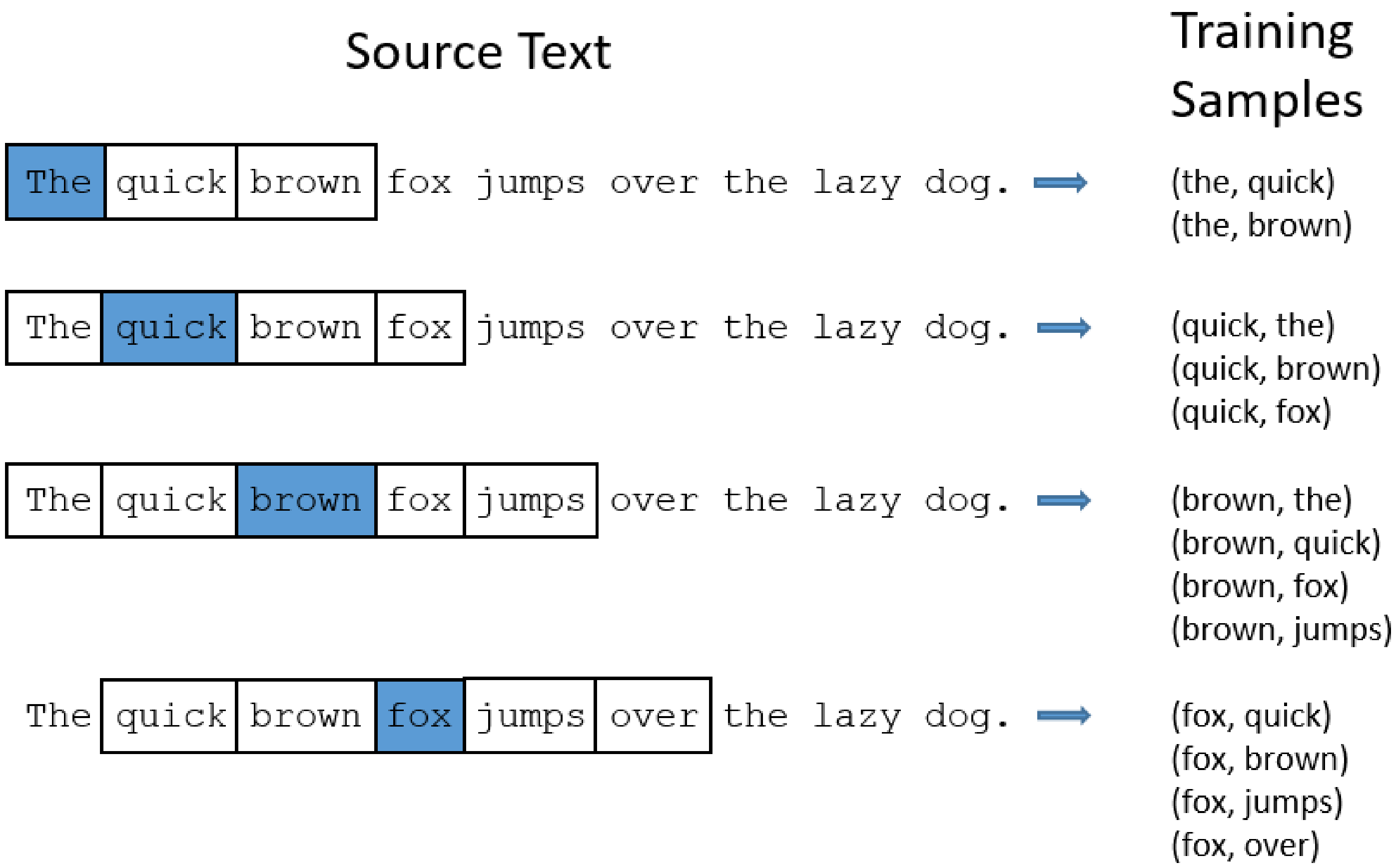
Task: learn the probability of a context of words given a source word.

Word embeddings

Word2vec
Skip-gram model

The **context of words** is defined using a **sliding window** of fixed length through a large **corpus** of text

Task: learn the probability of a context of words given a source word.



Source Text

Training Samples

The quick brown fox jumps over the lazy dog. ⟹ (the, quick) (the, brown)

The quick brown fox jumps over the lazy dog. ⟹ (quick, the) (quick, brown) (quick, fox)

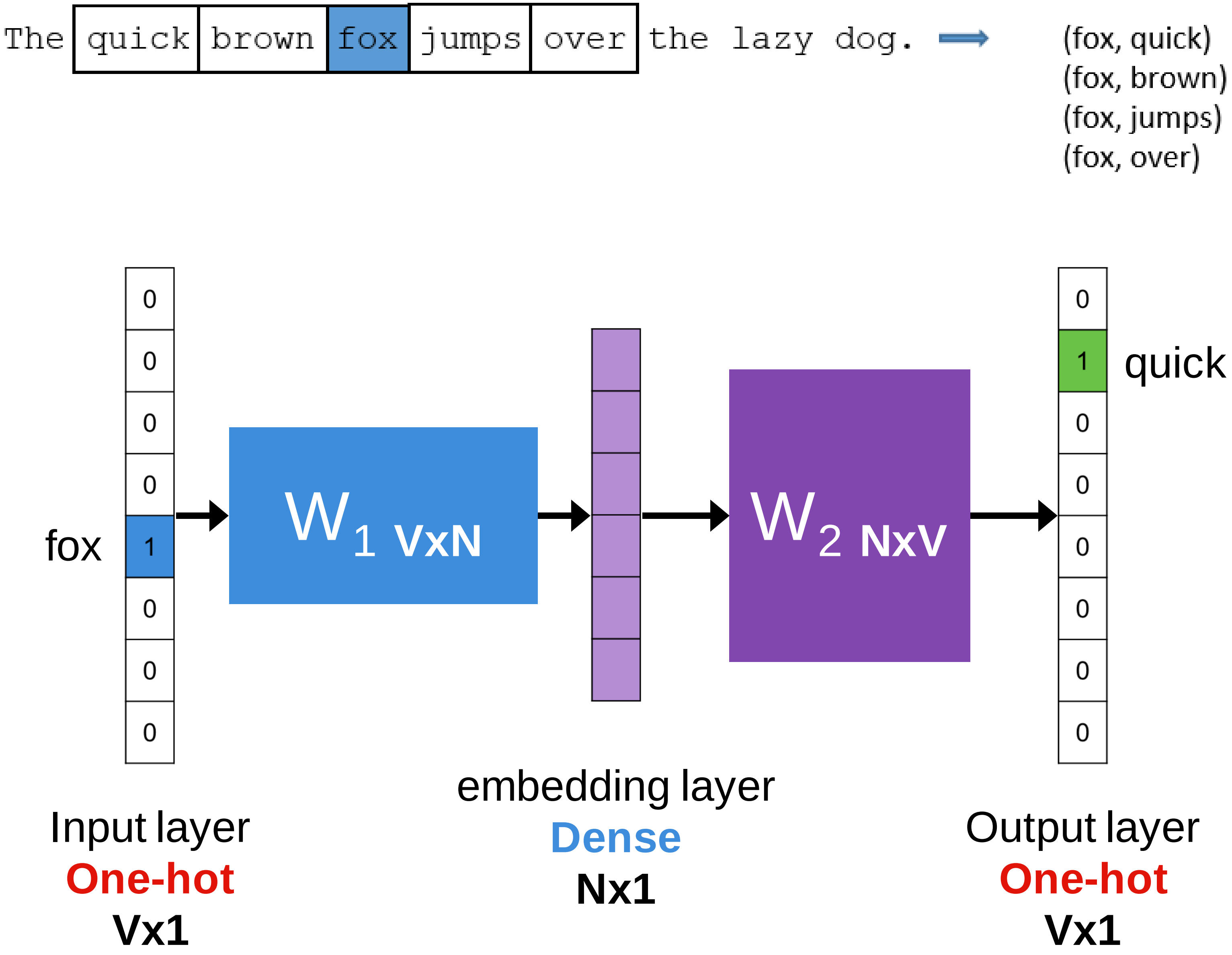
The quick brown fox jumps over the lazy dog. ⟹ (brown, the) (brown, quick) (brown, fox) (brown, jumps)

The quick brown fox jumps over the lazy dog. ⟹ (fox, quick) (fox, brown) (fox, jumps) (fox, over)

Word embeddings

Word2vec
Skip-gram model

Task: learn the probability of a context of words given a source word.



The quick brown fox jumps over the lazy dog.

(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

fox

$W_{1\ VxN}$

$W_{2\ NxV}$

quick

Input layer
One-hot
Vx1

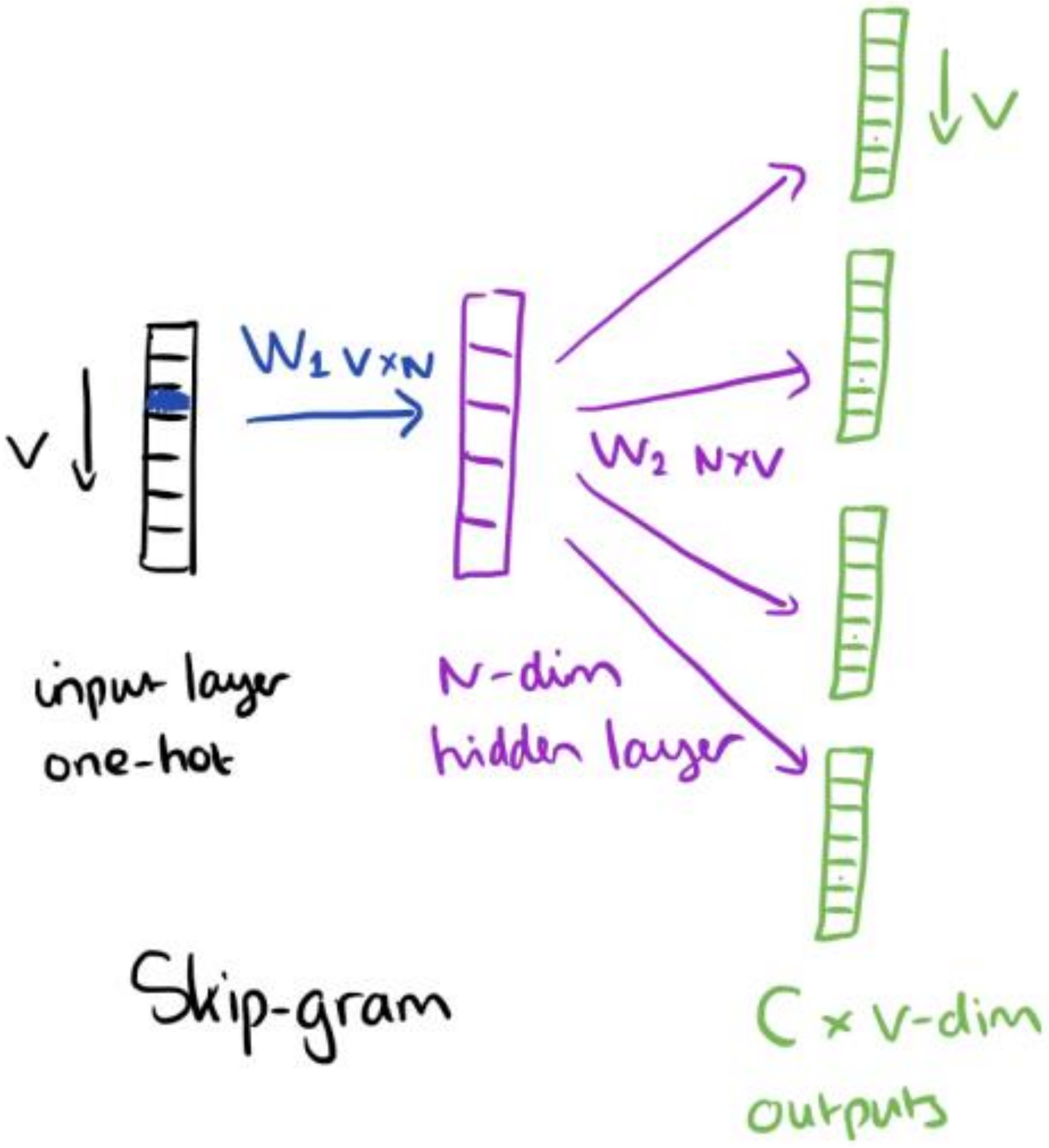
embedding layer
Dense
Nx1

Output layer
One-hot
Vx1

Word embeddings

Word2vec
Skip-gram model

Task: learn the probability of a context of words given a source word.



Word embeddings

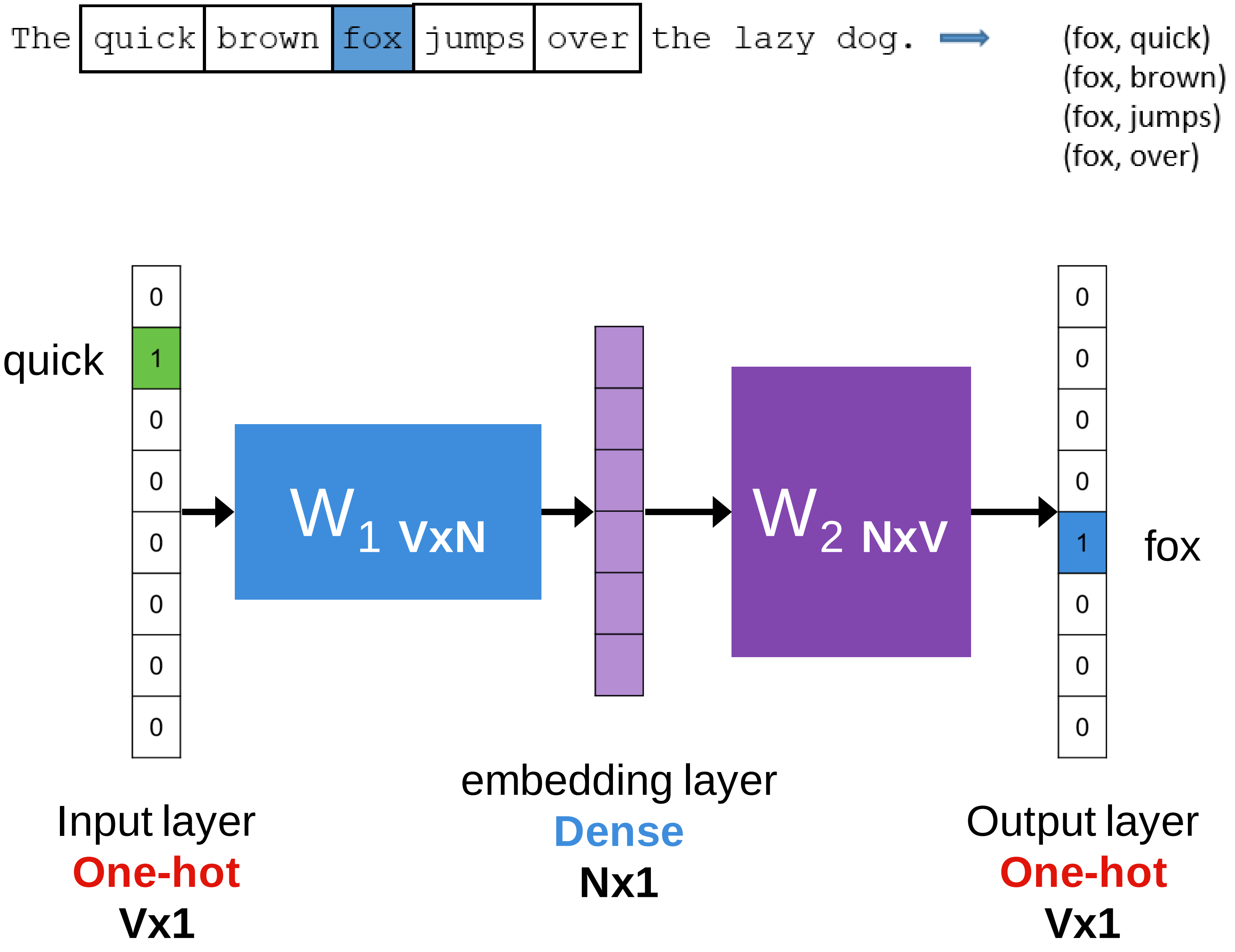
Word2vec
CBOW model

Task: learn the probability of a word given its context.

Word embeddings

Word2vec CBOW model

Task: learn the probability of a word given its context.



The quick brown fox jumps over the lazy dog. ⟹

(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

quick

$W_{1\ VxN}$

$W_{2\ NxV}$

fox

Input layer
One-hot
Vx1

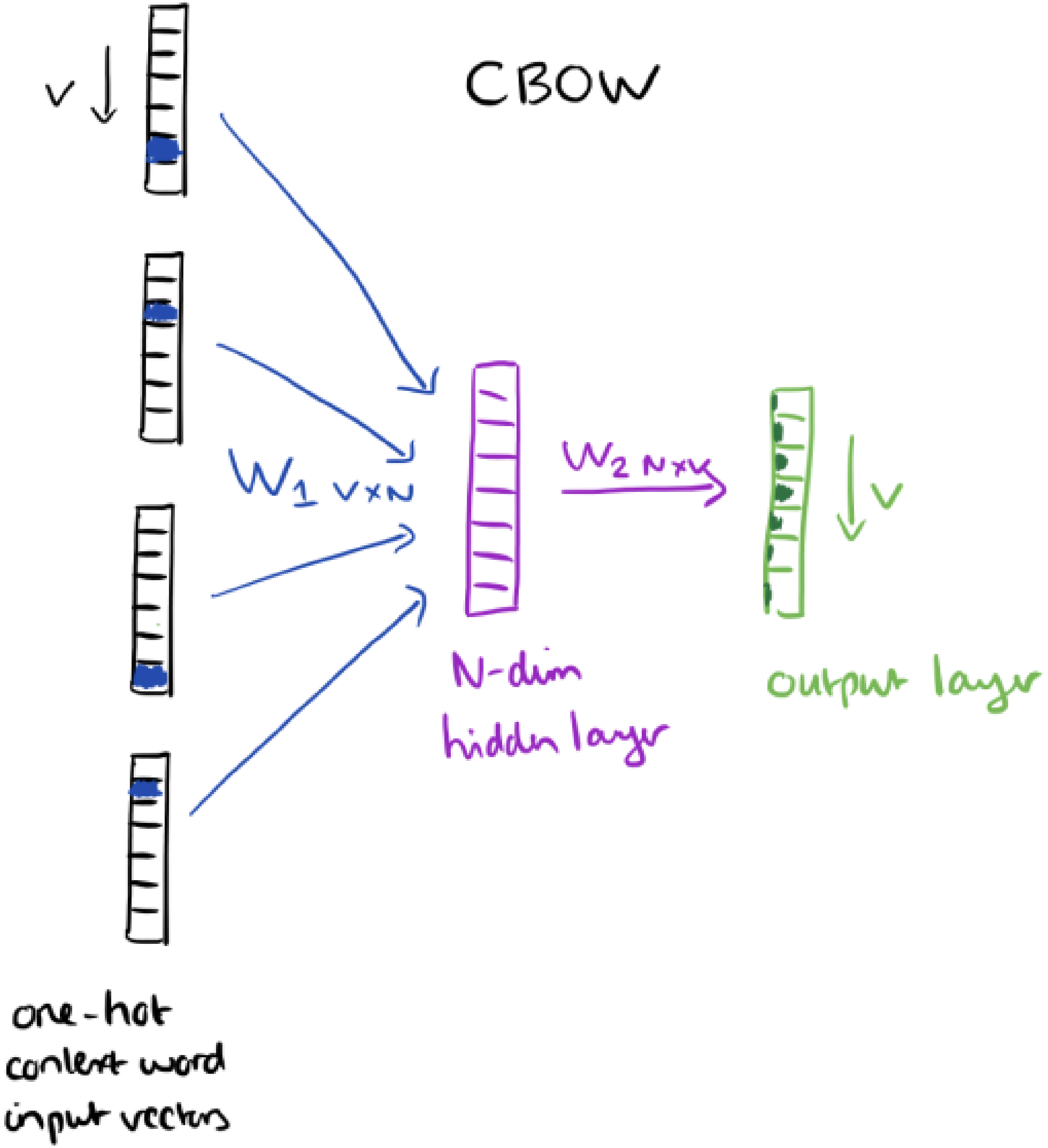
embedding layer
Dense
Nx1

Output layer
One-hot
Vx1

Word embeddings

Word2vec CBOW model

Task: learn the probability of a word given its context.



Word embeddings

Word2vec
CBOW model
Skip-gram model

Skip-gram works well with little training data and represents well rare words

CBOW is faster to train and has slightly better accuracies for frequent words

Word embeddings

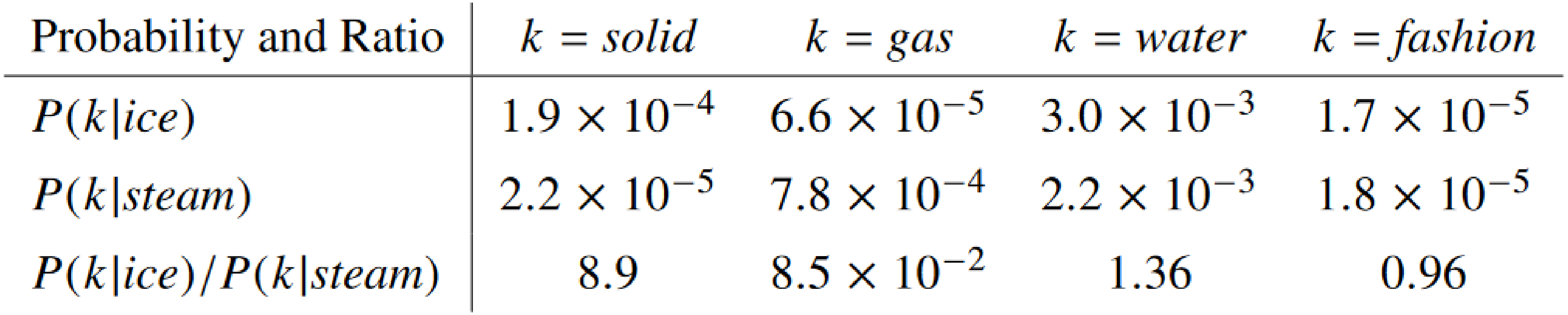
GloVe

Word-word co-occurrence probabilities can encode meaning

Word embeddings

GloVe

Word-word co-occurrence probabilities can encode meaning



Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Word embeddings

GloVe

Word-word co-occurrence probabilities can encode meaning

$$\underbrace{J}_{\text{Cost to minimize (AdaGrad)}} = \overbrace{\sum_{i,j=1}^{V}}^{\text{Sum over all pairs of words}} \underbrace{f\left(X_{ij}\right)}_{\text{Weighting function}} \Big(\overbrace{w_i^T \tilde{w}_j + b_i + \tilde{b}_j}^{\text{Vectors embeddings} = W + \widetilde{W}} - \underbrace{\log X_{ij}}_{\text{Log of co-occurrence}} \Big)^2$$

Word embeddings

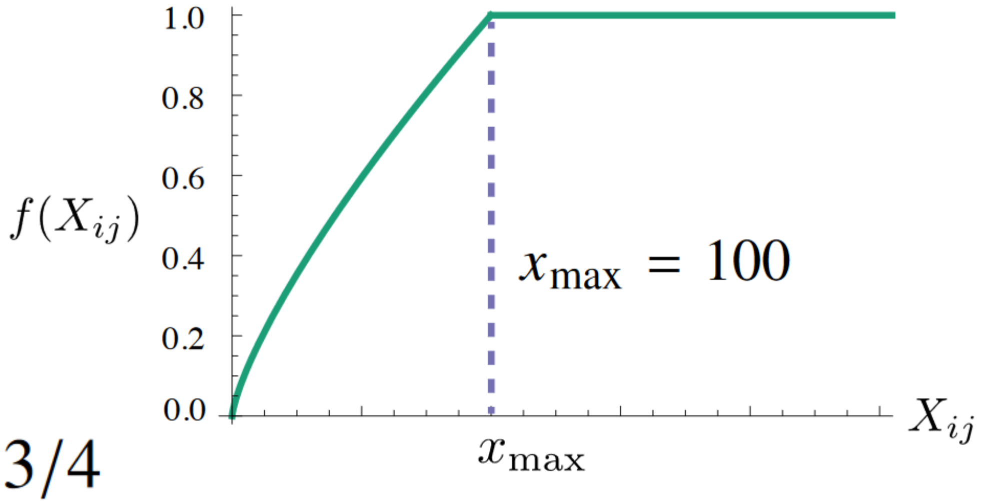
GloVe

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

Weighting function

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}$$

Word-word co-occurrence probabilities can encode meaning



$\alpha = 3/4$

Word embeddings

Linguistic regularities

- **Linguistic or semantic similarity:**

Nearest neighbours using Euclidean (cosine) distance

frog

1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus

Word embeddings

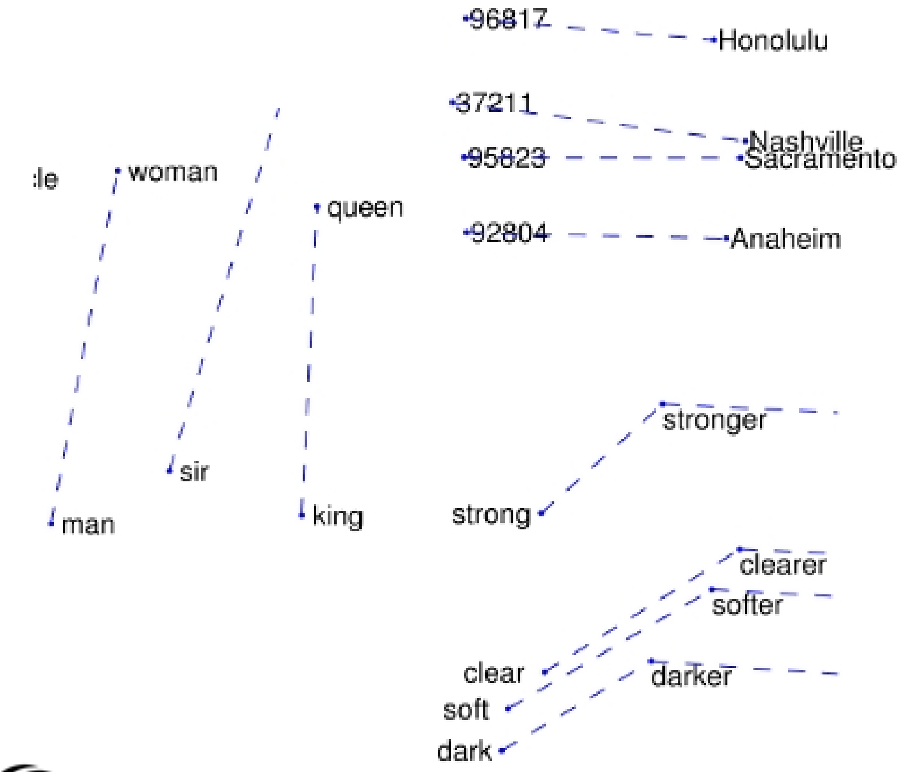
Linguistic regularities

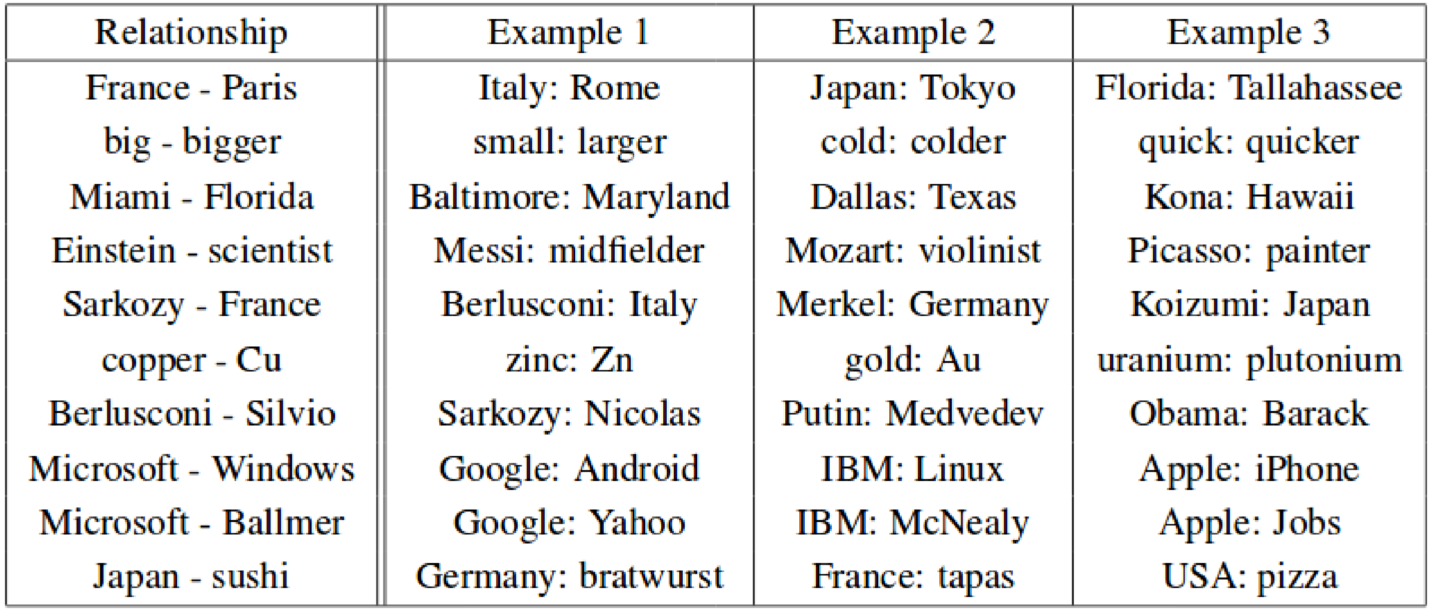
- **Linguistic or semantic similarity:**

  Nearest neighbours using Euclidean (cosine) distance

- **Linear substructures:**

  Semantic / syntactic relations





Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Word embeddings

Linguistic regularities

- **Linguistic or semantic similarity:**

  Nearest neighbours using Euclidean (cosine) distance

- **Linear substructures:**

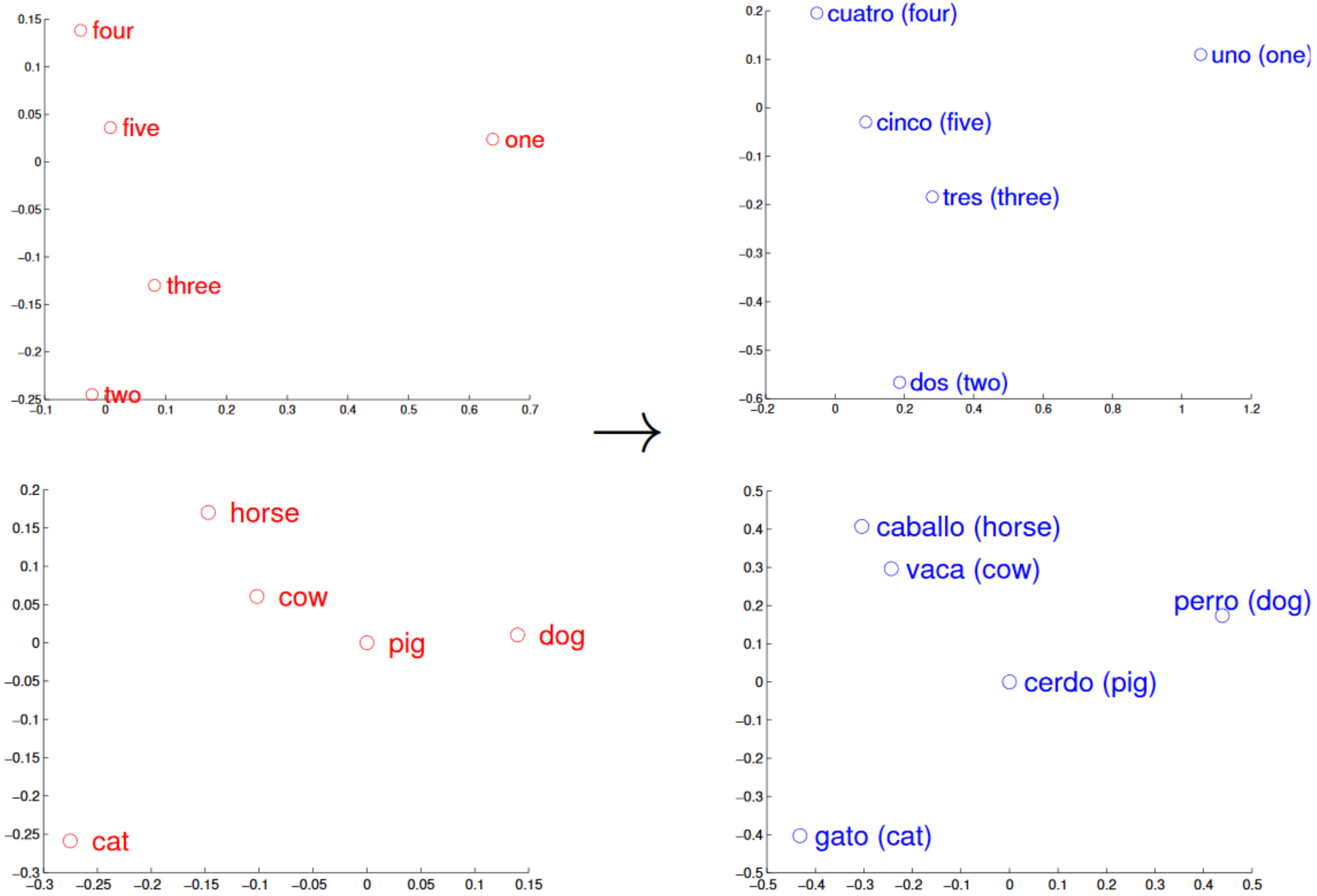
  Semantic / syntactic relations

- **Multilingual embeddings:**

  Word translation

Word embeddings

Linguistic regularities



Word embeddings

Linguistic regularities

- **Linguistic or semantic similarity:**

  Nearest neighbours using Euclidean (cosine) distance

- **Linear substructures:**

  Semantic / syntactic relations

- **Multilingual embeddings:**

  Word translation

Word embeddings

- **Linguistic or semantic similarity:**

  Nearest neighbours using Euclidean (cosine) distance

- **Linear substructures:**

  Semantic / syntactic relations

- **Multilingual embeddings:**

  Word translation

Word embeddings

Practical aspects

- Corpus:
  - General language
  - Specific / technical language
- Pre-trained models

  → do not train the wheel again
- Method performance depends on the problem.

  → try different models
- Language bias

  → IA inherits our biases

Word embeddings

Some references

- Mikolov, Tomas, et al. Distributed Representations of Words and Phrases and their Compositionality.

http://papers.nips.cc/paper/5021-distributed-representations-of-words-andphrases

- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

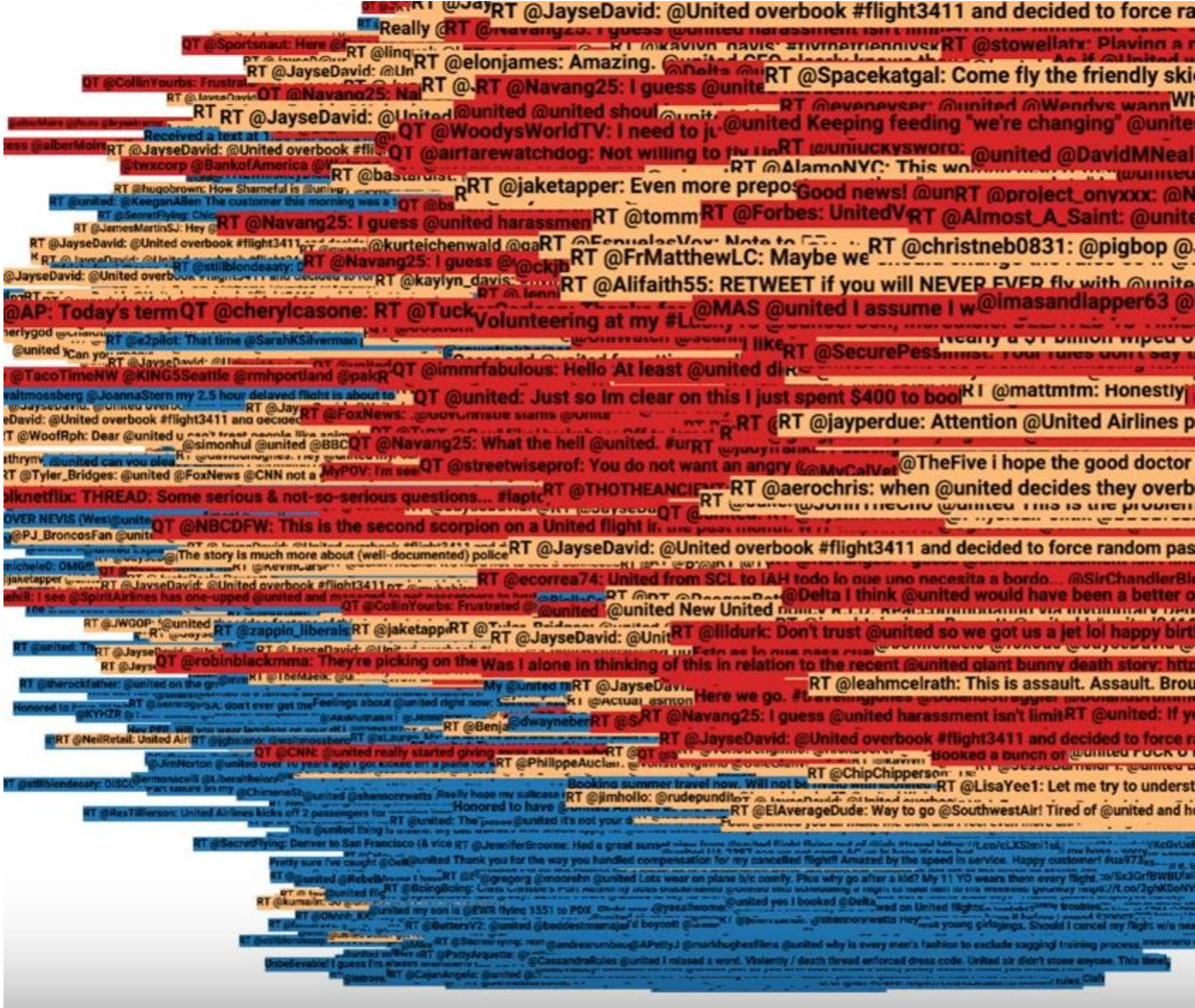
https://nlp.stanford.edu/projects/glove/

- word embedding visualization

https://ronxin.github.io/wevi/

https://projector.tensorflow.org/

Sentence embeddings



Sentence embeddings

How can we represent sentences, paragraphs or documents?

Sentence embeddings

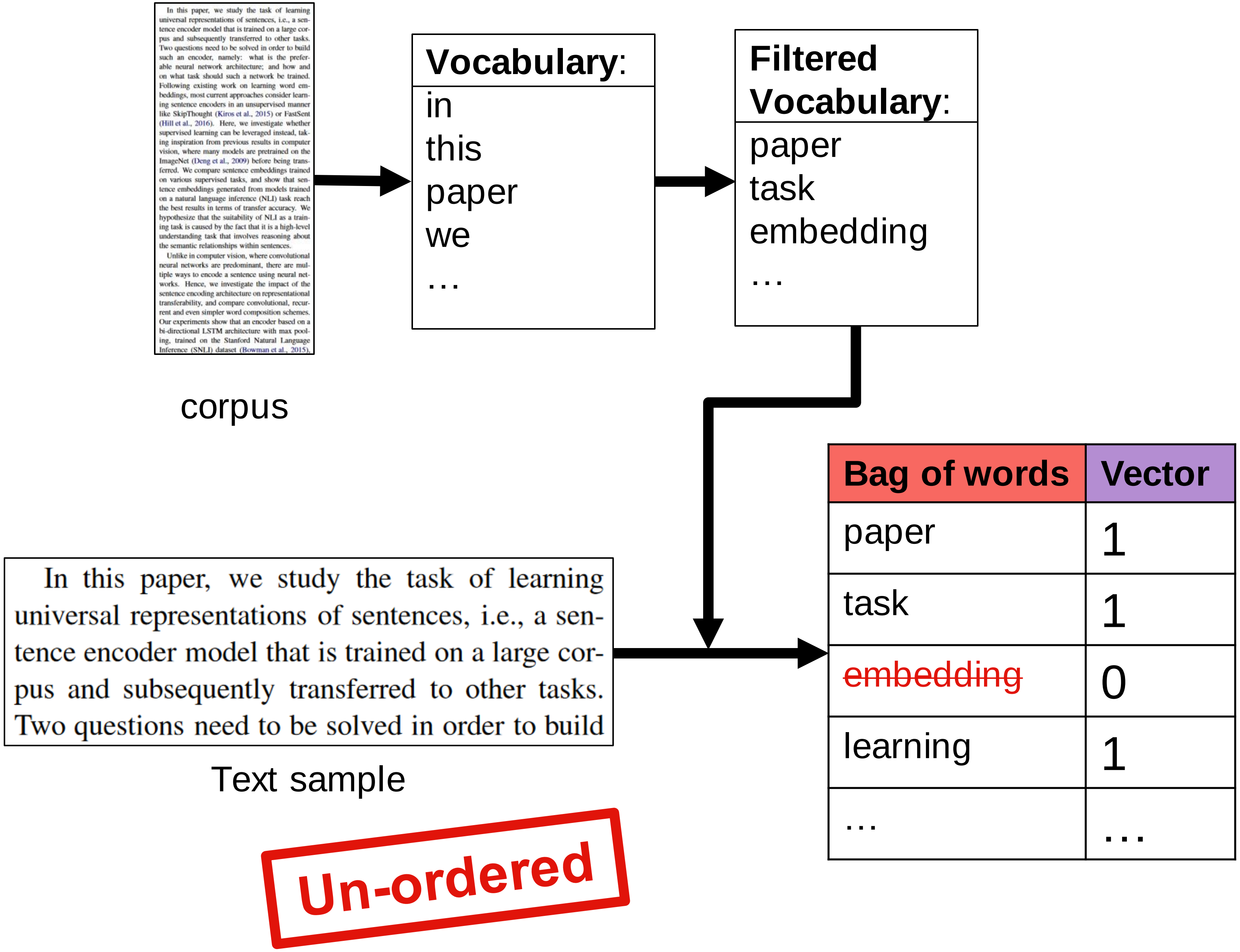
How can we represent sentences, paragraphs or documents?

Sentences have a different lengths!

Sentence embeddings

How can we represent sentences?

- **Bag of words – one-hot vector**



In this paper, we study the task of learning universal representations of sentences, i.e., a sentence encoder model that is trained on a large corpus and subsequently transferred to other tasks. Two questions need to be solved in order to build such an encoder, namely: what is the preferable neural network architecture; and how and on what task should such a network be trained. Following existing work on learning word embeddings, most current approaches consider learning sentence encoders in an unsupervised manner like SkipThought (Kiros et al., 2015) or FastSent (Hill et al., 2016). Here, we investigate whether supervised learning can be leveraged instead, taking inspiration from previous results in computer vision, where many models are pretrained on the ImageNet (Deng et al., 2009) before being transferred. We compare sentence embeddings trained on various supervised tasks, and show that sentence embeddings generated from models trained on a natural language inference (NLI) task reach the best results in terms of transfer accuracy. We hypothesize that the suitability of NLI as a training task is caused by the fact that it is a high-level understanding task that involves reasoning about the semantic relationships within sentences.

Unlike in computer vision, where convolutional neural networks are predominant, there are multiple ways to encode a sentence using neural networks. Hence, we investigate the impact of the sentence encoding architecture on representational transferability, and compare convolutional, recurrent and even simpler word composition schemes. Our experiments show that an encoder based on a bi-directional LSTM architecture with max pooling, trained on the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015).

corpus

Vocabulary:

in
this
paper
we
…

Filtered Vocabulary:

paper
task
embedding
…

In this paper, we study the task of learning universal representations of sentences, i.e., a sentence encoder model that is trained on a large corpus and subsequently transferred to other tasks. Two questions need to be solved in order to build

Text sample

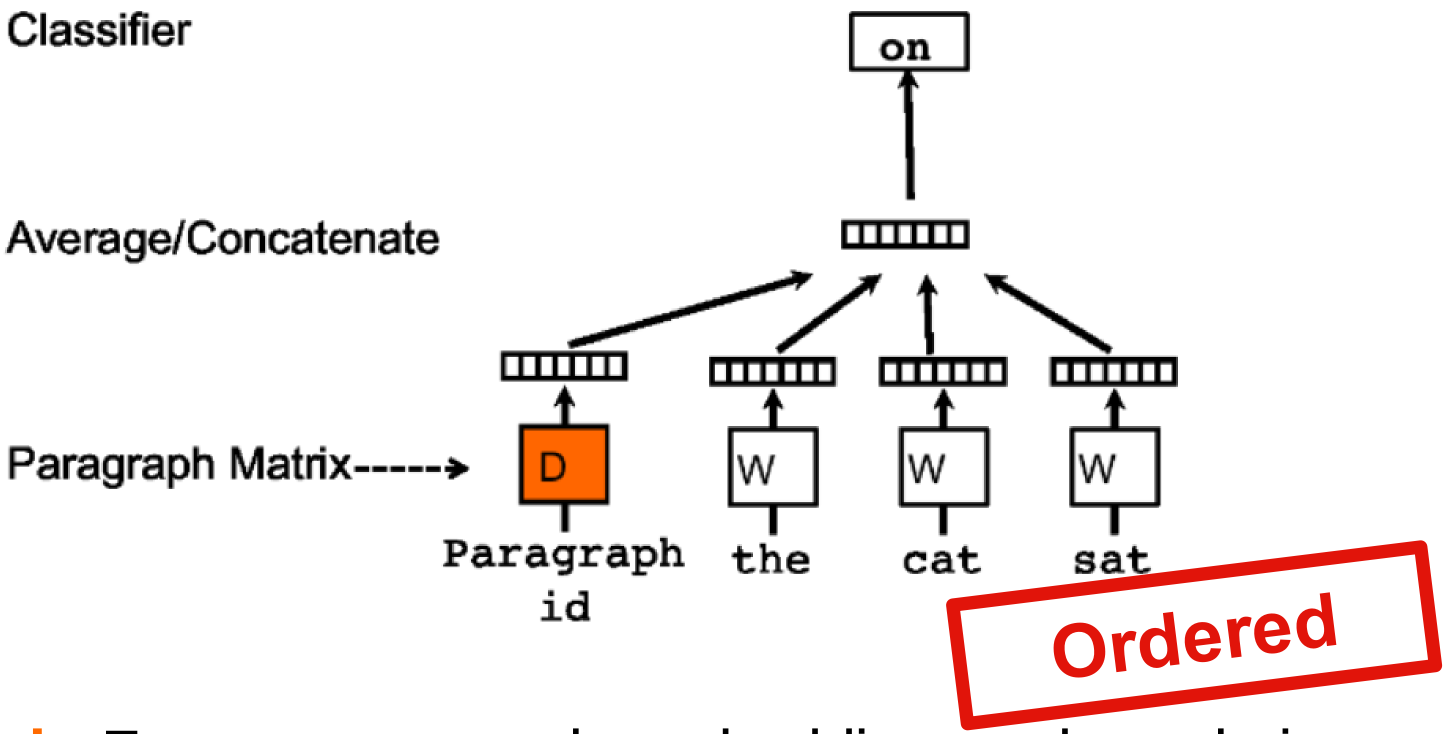
Bag of words	**Vector**
paper	1
task	1
~~embedding~~	0
learning	1
…	…

Un-ordered

Sentence embeddings

How can we represent sentences?

- Bag of words – one-hot vector
- Paragraph vector PV-DM (distributed memory)



Task: From a paragraph embedding and words in the sentence predict next word.

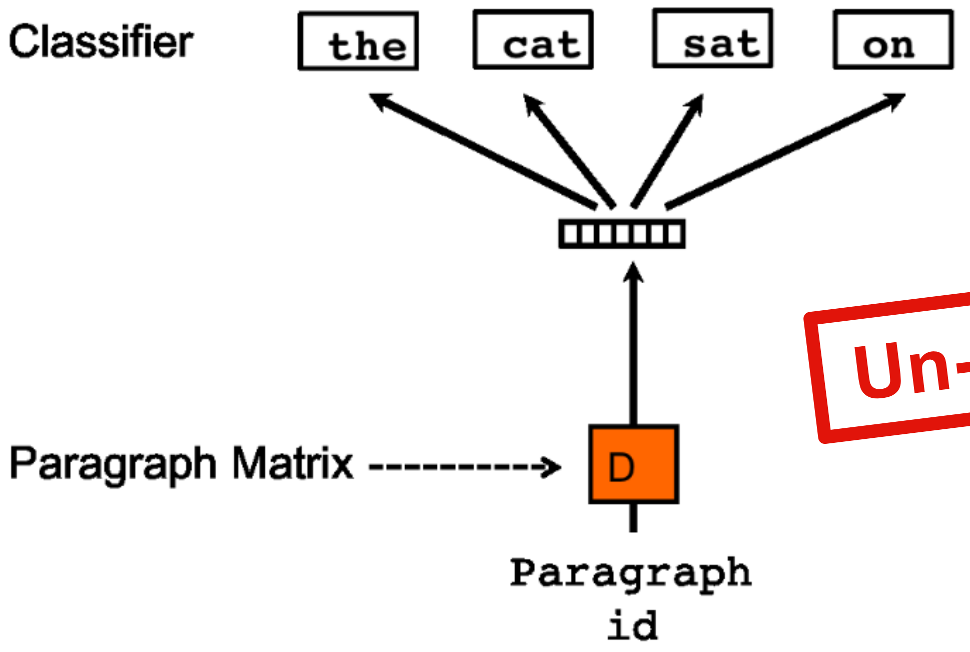
→ we learn the paragraph and word embeddings and softmax weights in the corpus.

For inference we fix all parameters but D and learn D by gradient descent.

Sentence embeddings

How can we represent sentences?

- Bag of words – one-hot vector
- Paragraph vector PV-DBOW (distributed BOW)



Un-ordered

Task: From a paragraph embedding predict random words from the paragraph.

→ we learn the paragraph embeddings and softmax weights in the corpus.

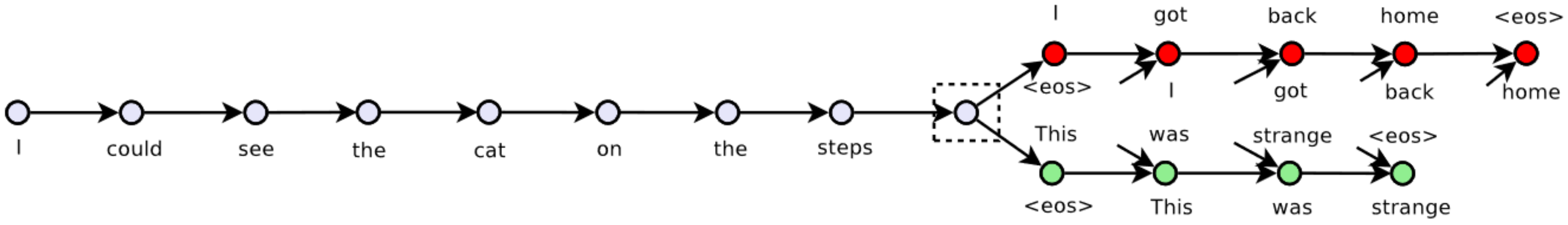
For inference we fix all parameters but D and learn D by gradient descent.

Combine PV-DBOW and PV-DM

Sentence embeddings

How can we represent sentences?

- **Bag of words – one-hot vector**
- **Paragraph vector**
- **Skip-thoughts**



Task: From a sentence predict next and previous sentences.

Encoder-decoder method:

ConvNet - RNN, RNN - RNN, LSTM – LSTM, **GRU - condGRU**

Unidirectional / bidirectional encoder or both

Objective: the sum of the log-probabilities for the forward and backward sentences conditioned on the encoder representation

$$\sum_t \log P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) + \sum_t \log P(w_{i-1}^t | w_{i-1}^{<t}, \mathbf{h}_i)$$

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Sentence embeddings

How can we represent sentences?

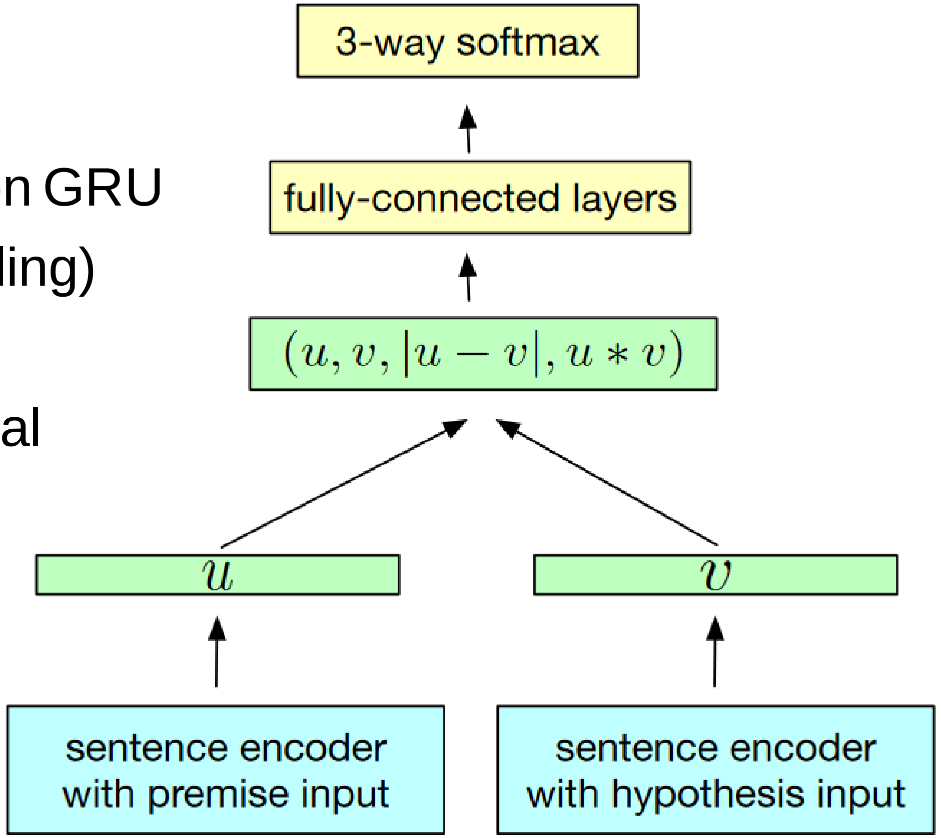
- Bag of words – one-hot vector
- Paragraph vector
- Skip-thoughts
- SNLI - BiLSTM

Task: SNLI: Stanford Natural Language Inference. 570K English sentences pairs labeled {entailment, contradiction, neutral}

Supervised

Sentence encoder architectures:

- LSTM
- GRU
- Concatenation 2-direction GRU
- **BiLSTM** (**max** - avg pooling)
- Self-attentive
- Hierarchical Convolutional

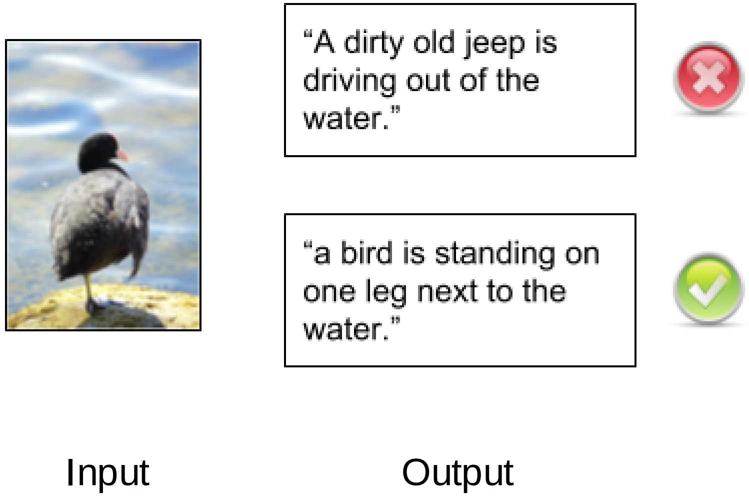

3-way softmax
fully-connected layers
$(u, v, |u - v|, u * v)$
u
v
sentence encoder with premise input
sentence encoder with hypothesis input

Multimodal embeddings

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Problem: Caption / Image retrieval

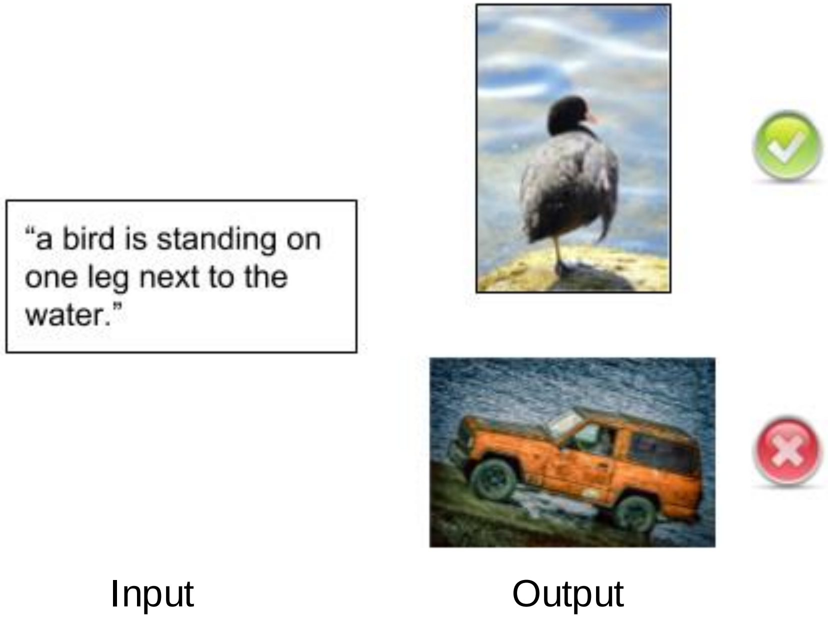
- A.K.A. Image Annotation
- For a given image, find the caption that best describes the image, from a set of defined captions.



“A dirty old jeep is driving out of the water.” ✗

“a bird is standing on one leg next to the water.” ✓

- A.K.A. Image Search
- For a given caption, find the image that is best described by the caption, from a set of given images.



“a bird is standing on one leg next to the water.”

Multimodal Embedding Space

- A common vector space.
- Learned on pairs of examples.
- The space is tuned to put similar items closer than different ones.
- Search is straight forward. Find nearest neighbours.

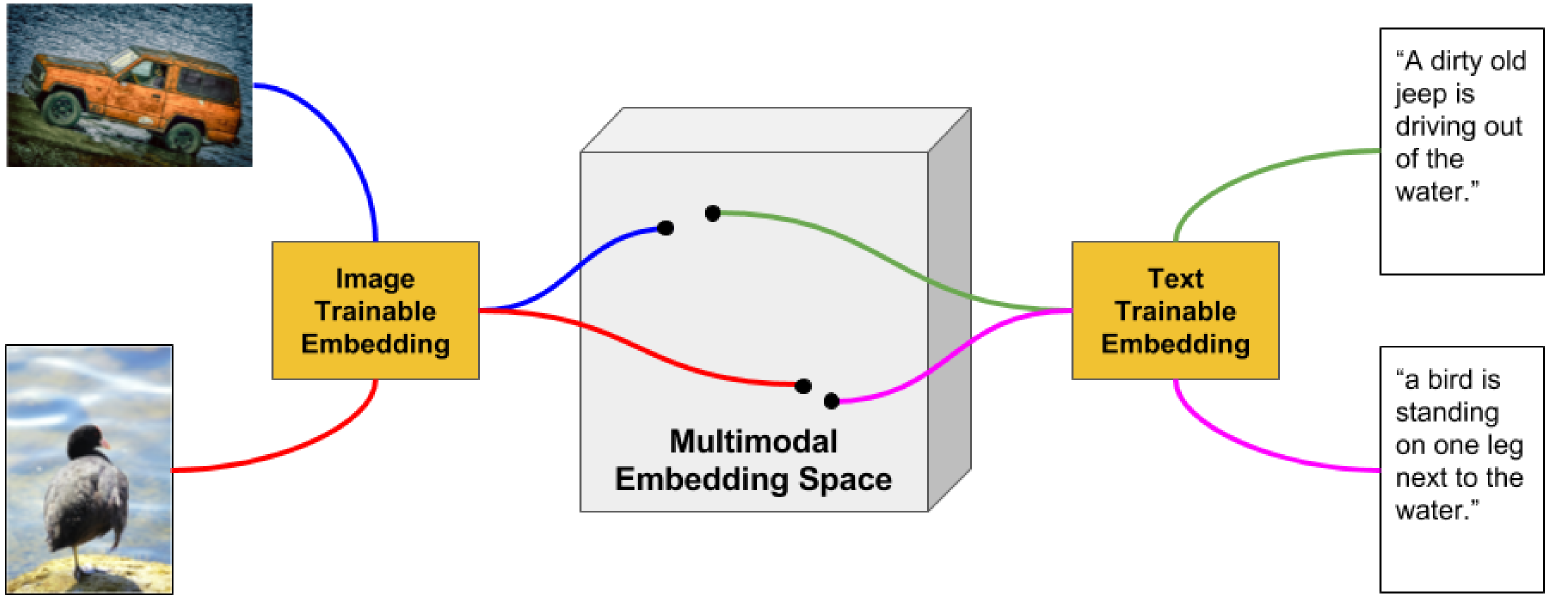
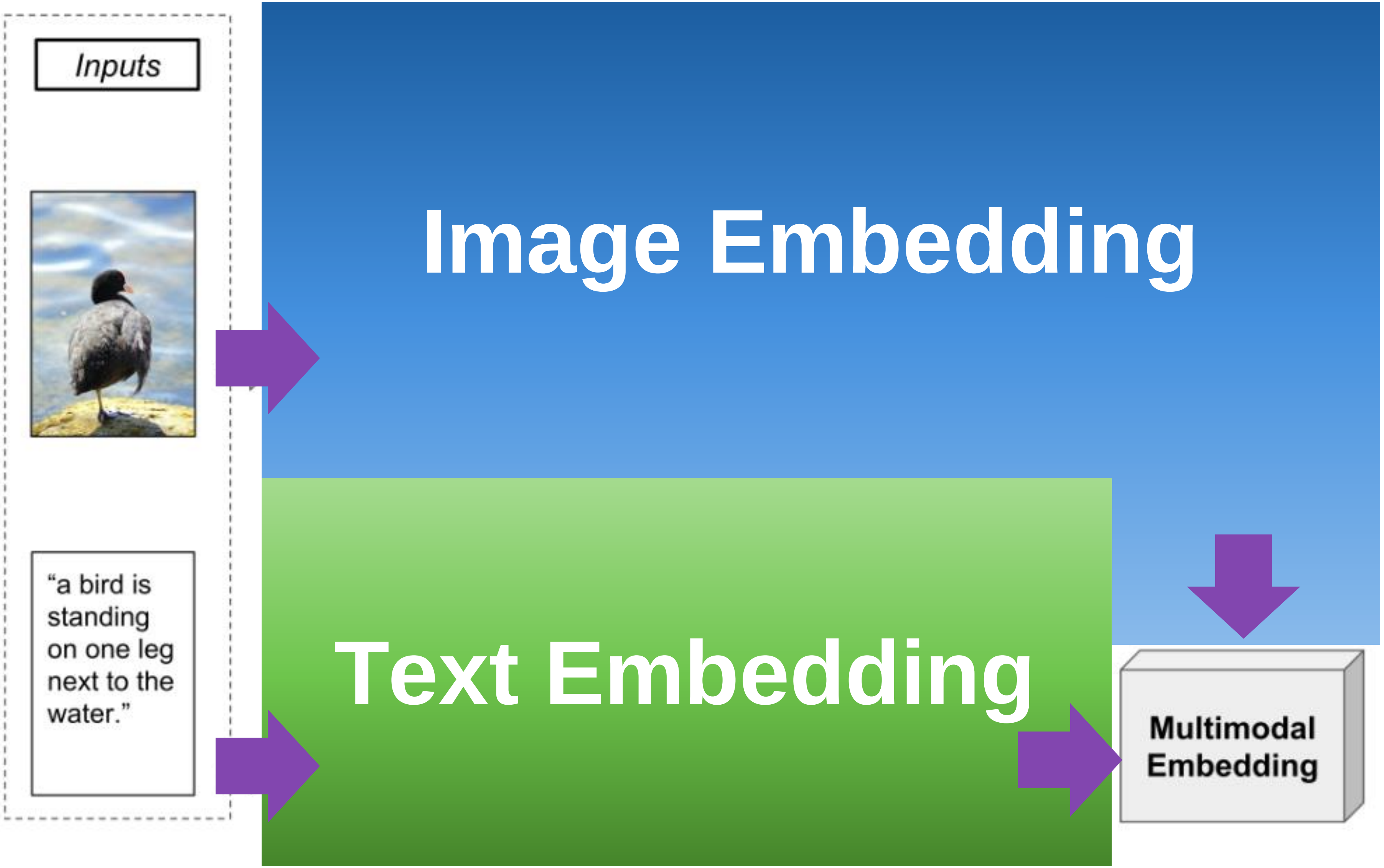


Image Trainable Embedding
Multimodal Embedding Space
Text Trainable Embedding
"A dirty old jeep is driving out of the water."
"a bird is standing on one leg next to the water."

The whole pipeline



Inputs

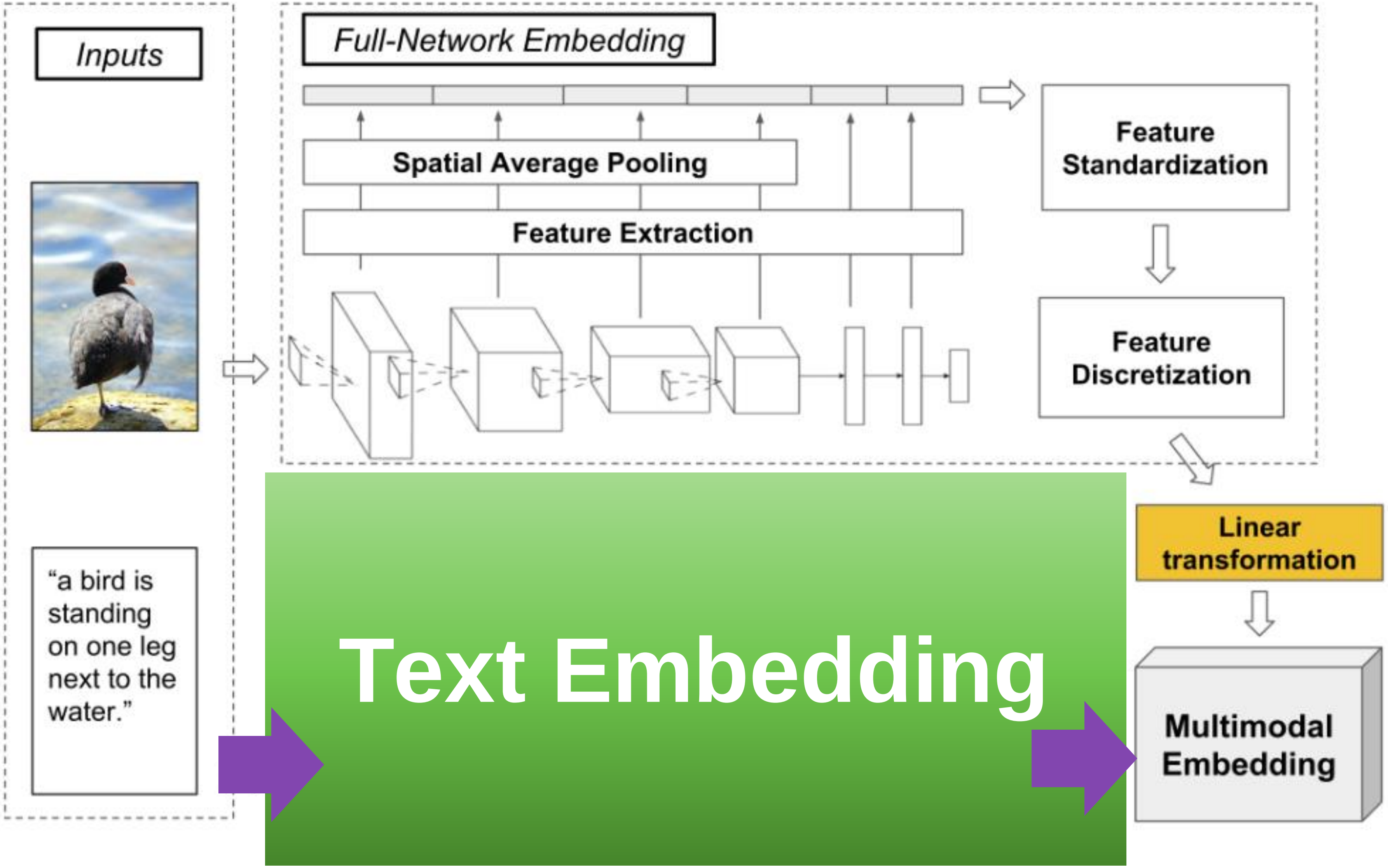
"a bird is standing on one leg next to the water."

Image Embedding

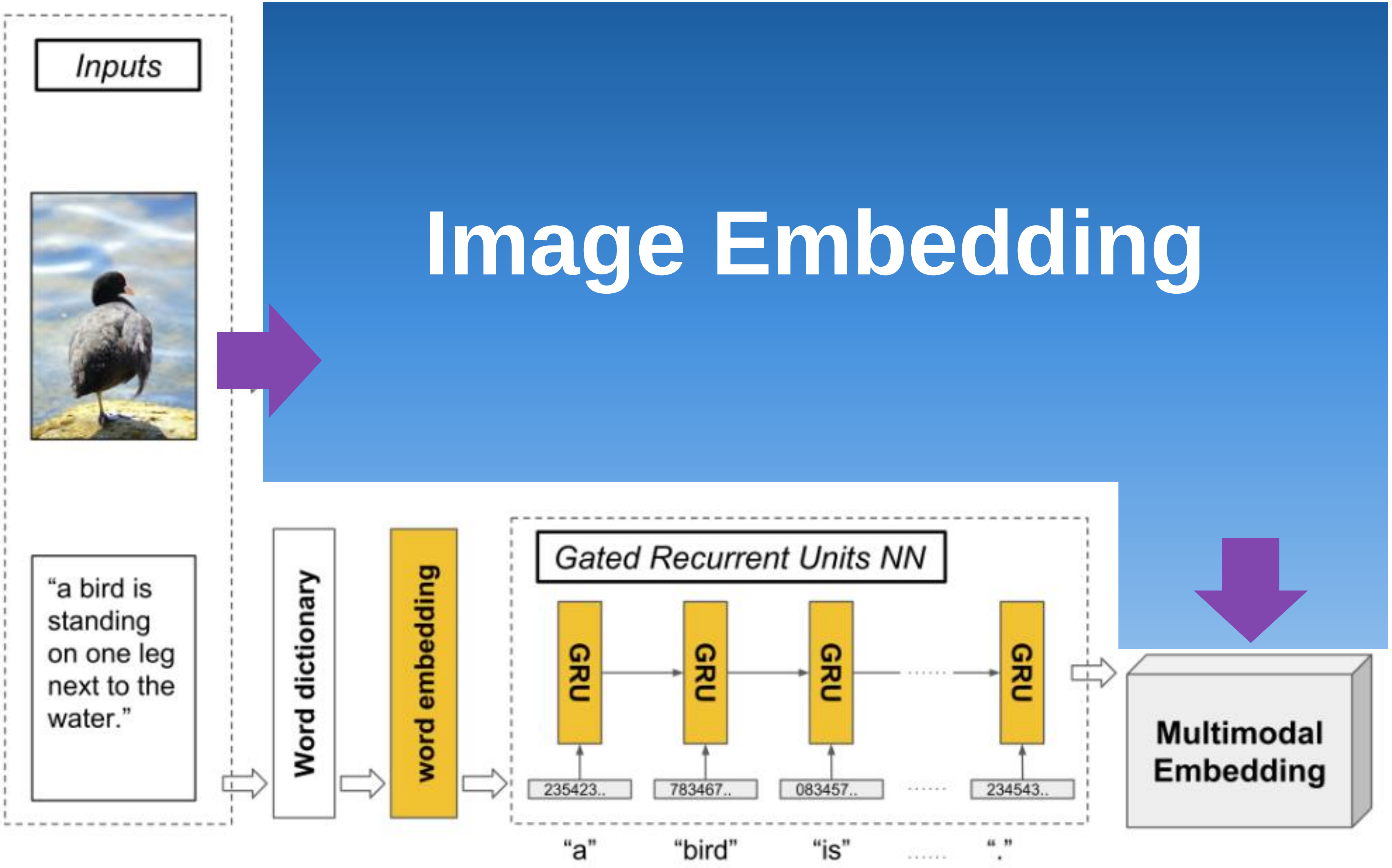
Text Embedding

Multimodal Embedding

The whole pipeline

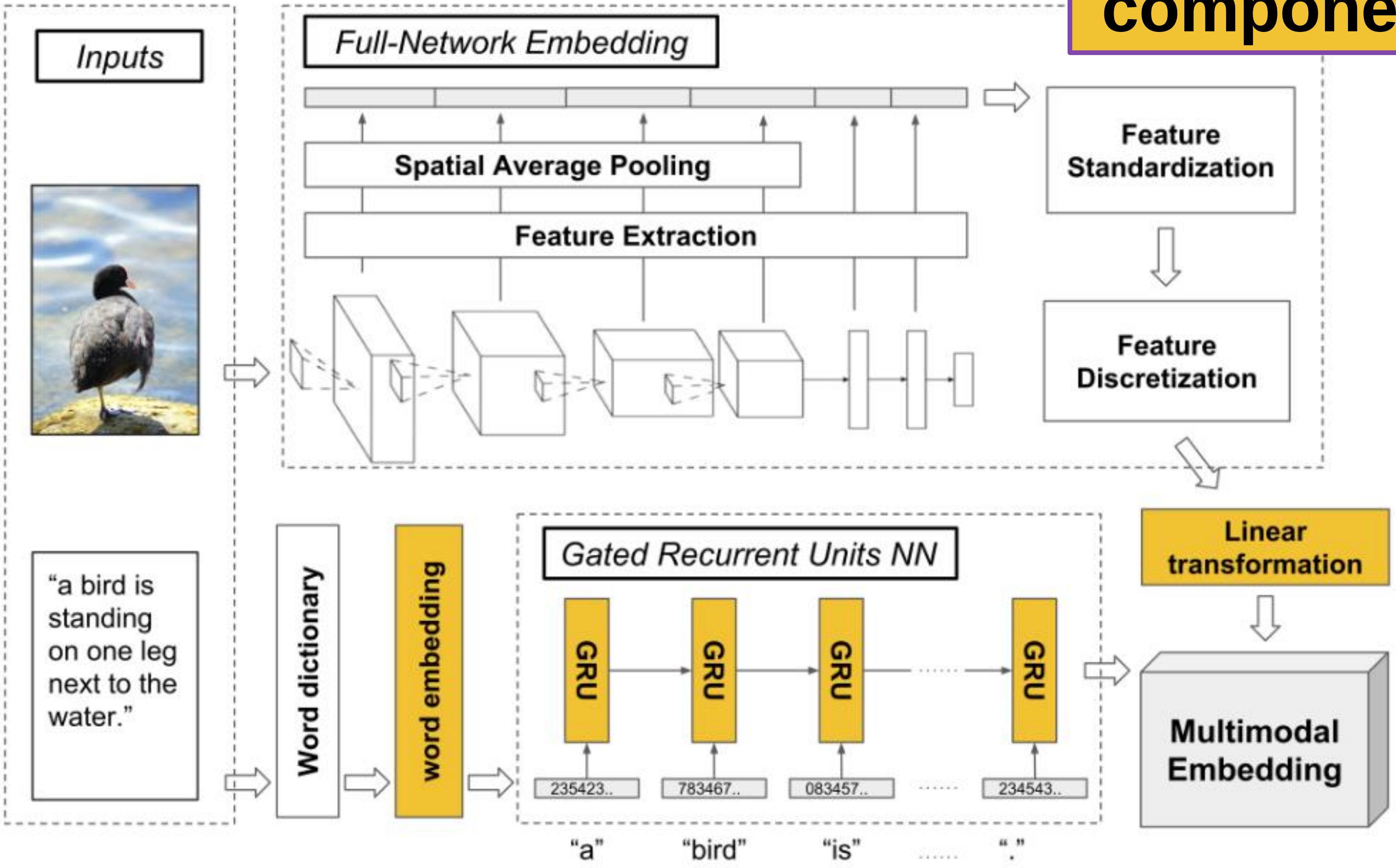


The whole pipeline



The whole pipeline

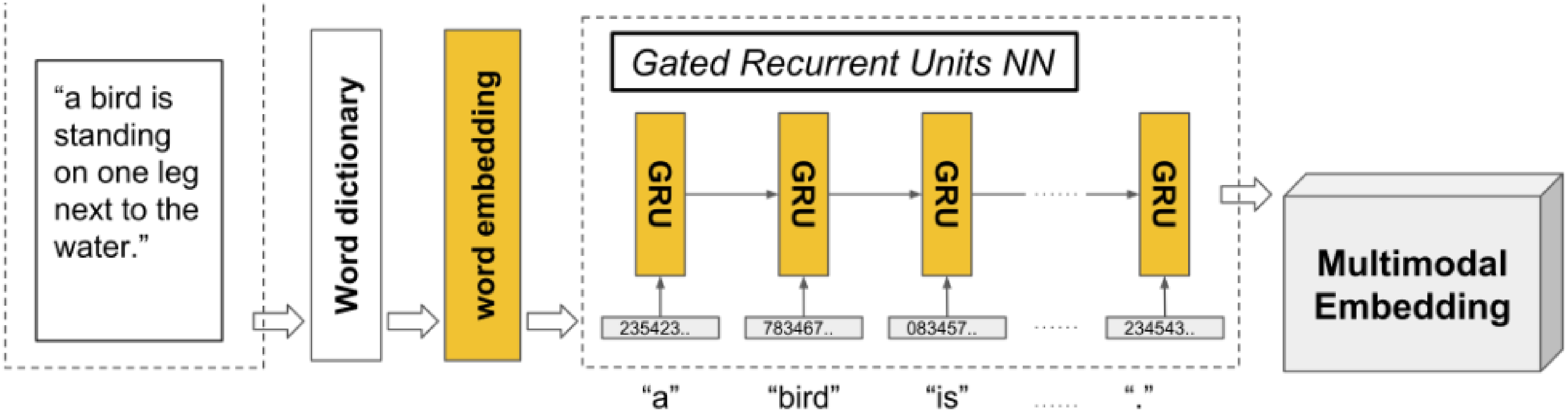
Trainable components



Text embedding

Following the approach described in:

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel.
Unifying visual-semantic embeddings with multimodal neural language models.

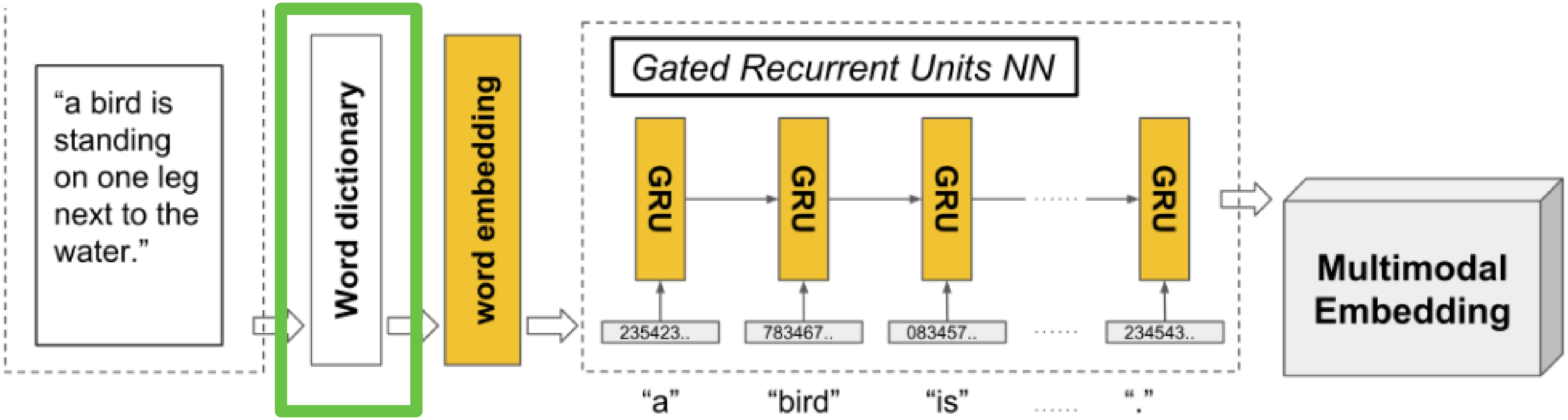


Text embedding

Following the approach described in:

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel.
Unifying visual-semantic embeddings with multimodal neural language models.

- Define a one-hot vector encoding of words via word dictionary.



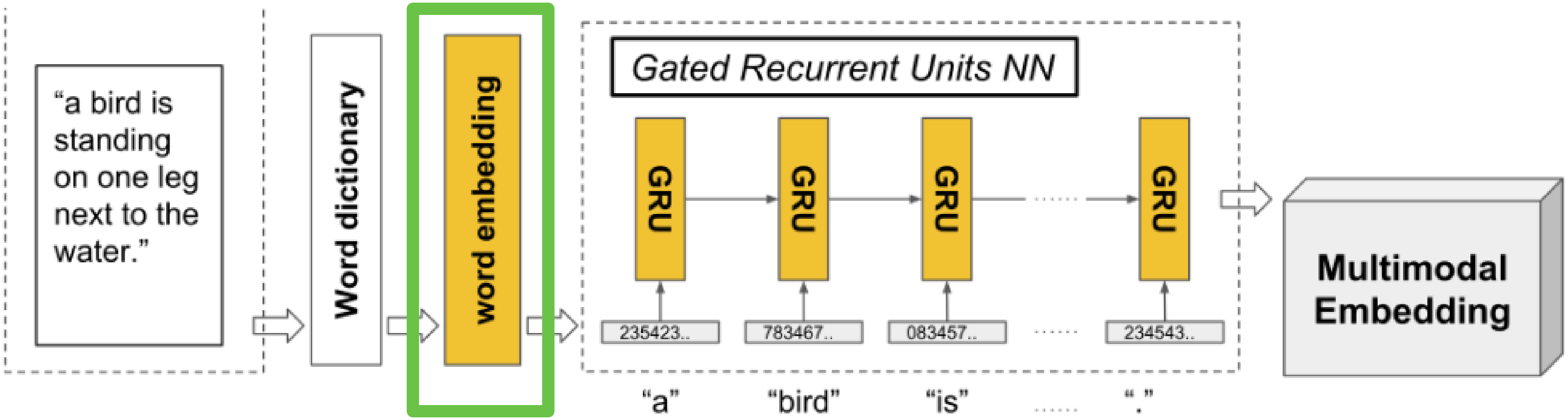
Text embedding

Following the approach described in:

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel.

Unifying visual-semantic embeddings with multimodal neural language models.

- Define a one-hot vector encoding of words via word dictionary.
- Obtain a dense representation using a trainable linear embedding.



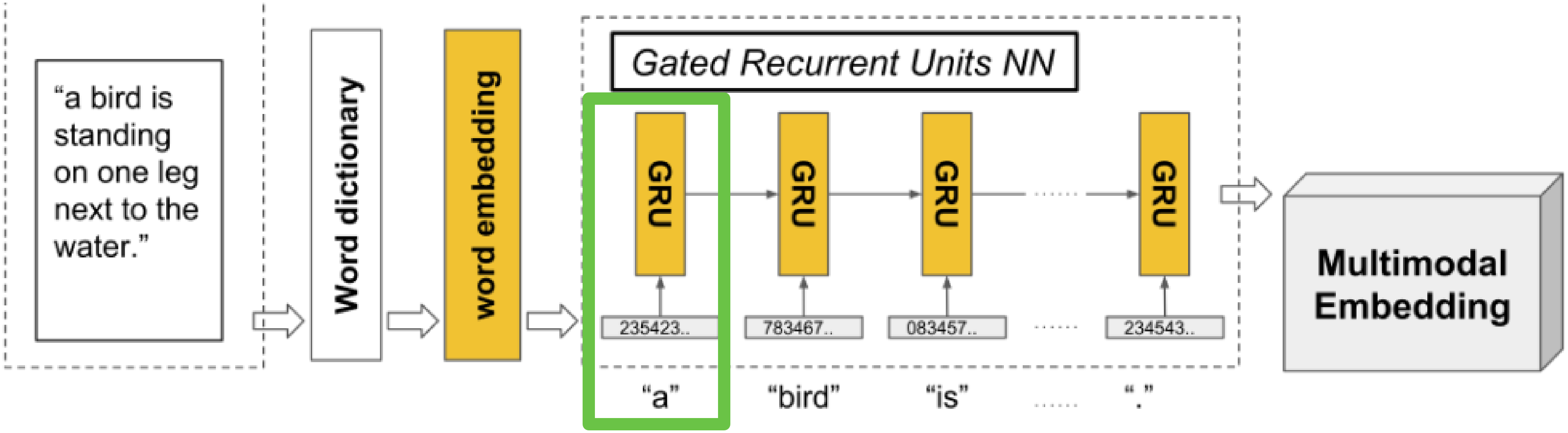
Text embedding

Following the approach described in:

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel.

Unifying visual-semantic embeddings with multimodal neural language models.

- Define a one-hot vector encoding of words via word dictionary.
- Obtain a dense representation using a trainable linear embedding.
- Feed the caption, word per word, to a Gated Recurrent Units (GRUs) Neural Network.



Text embedding

Following the approach described in:

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel.
Unifying visual-semantic embeddings with multimodal neural language models.

- Define a one-hot vector encoding of words via word dictionary.
- Obtain a dense representation using a trainable linear embedding.
- Feed the caption, word per word, to a Gated Recurrent Units (GRUs) Neural Network.
- Use the final hidden state of the network as text embedding.

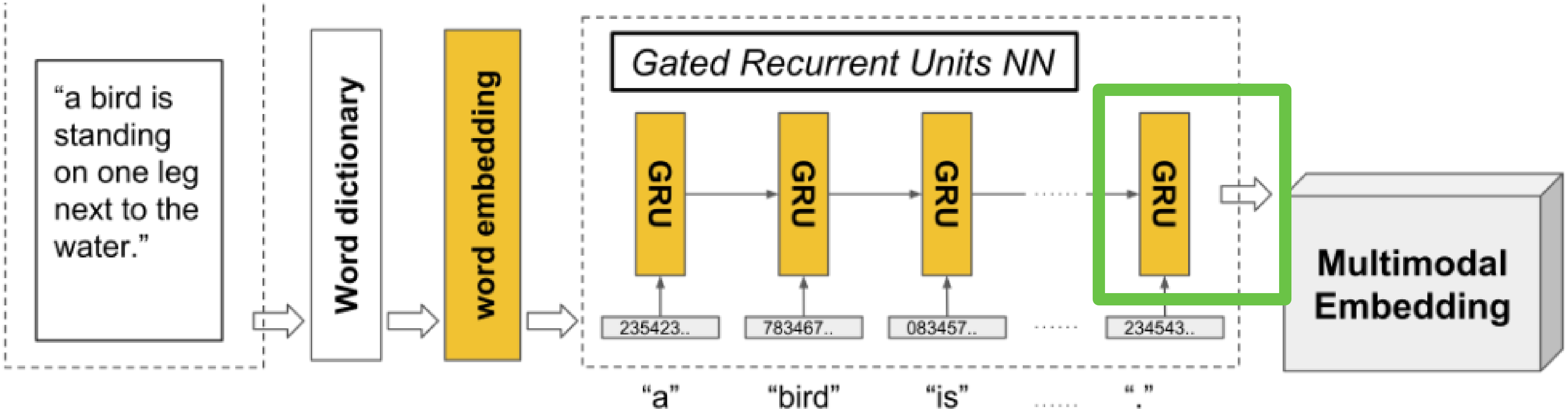


Image embedding

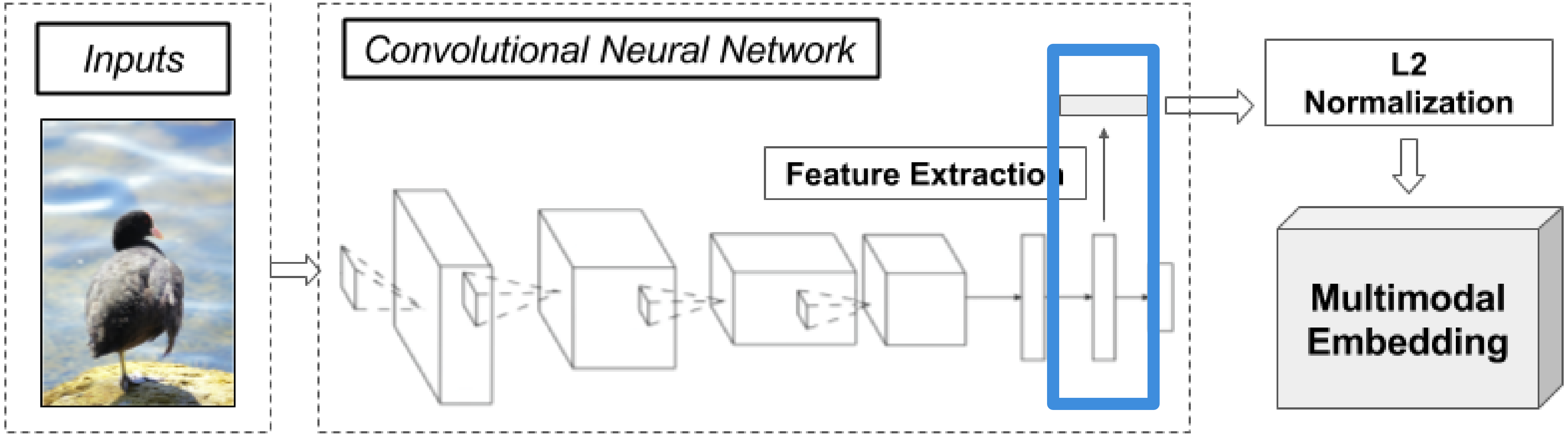
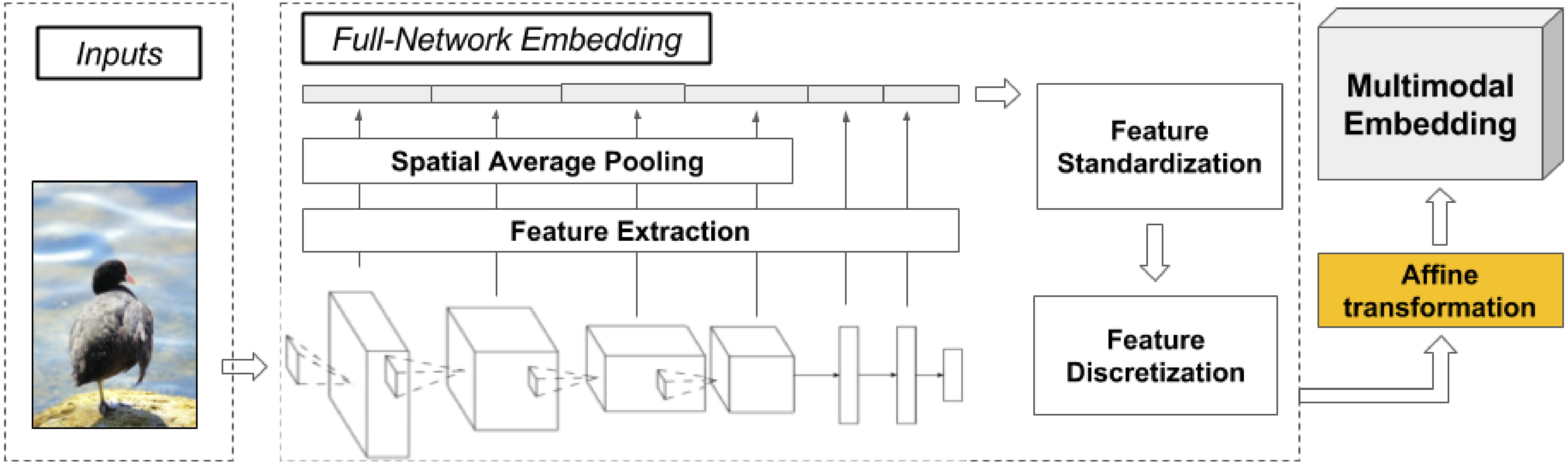


Image embedding



Questions?

Dario Garcia Gasulla dario.garcia@bsc.es
Armand Vilalta

Barcelona Supercomputing Center
Centro Nacional de Supercomputación