Sequence Learning

Feed-Forward Networks for Sequence Data

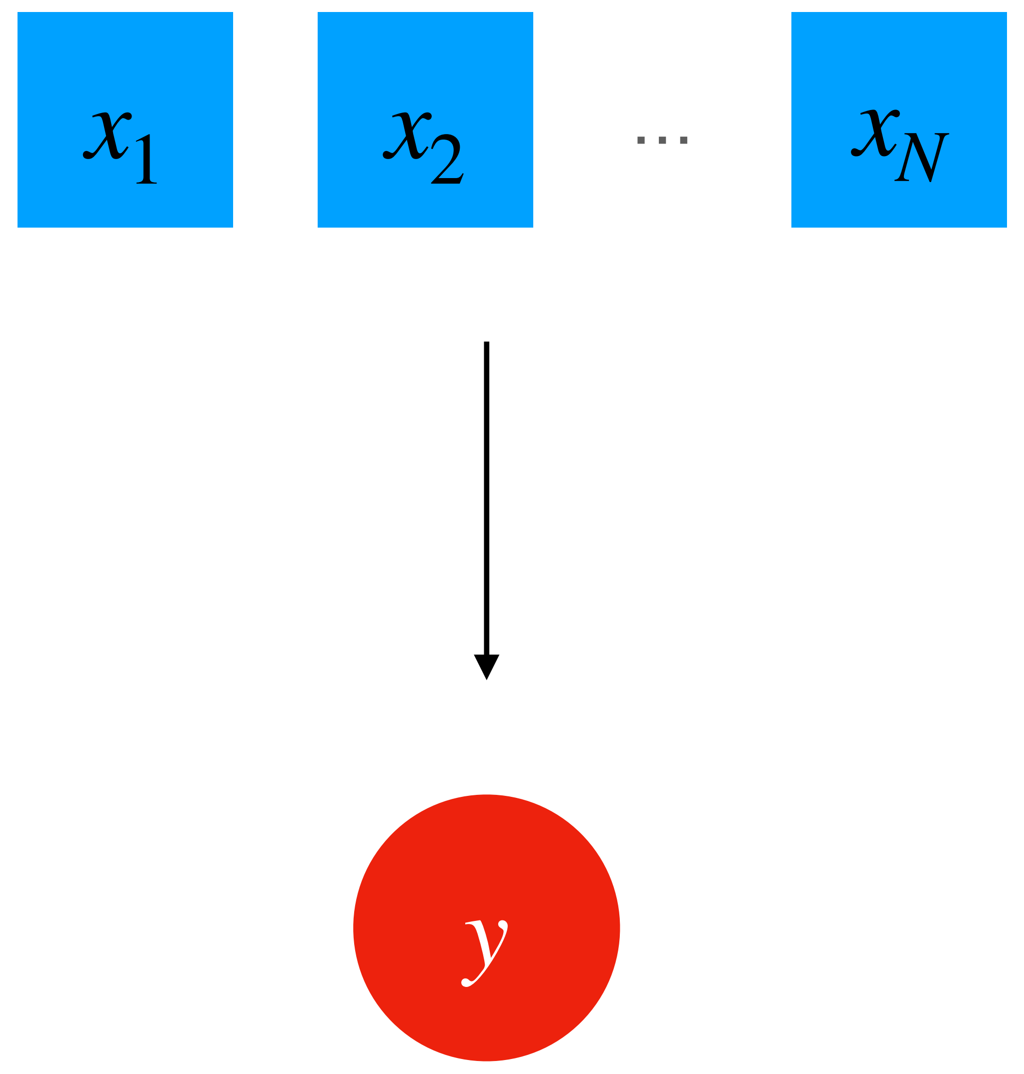
Korbinian Riedhammer

TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

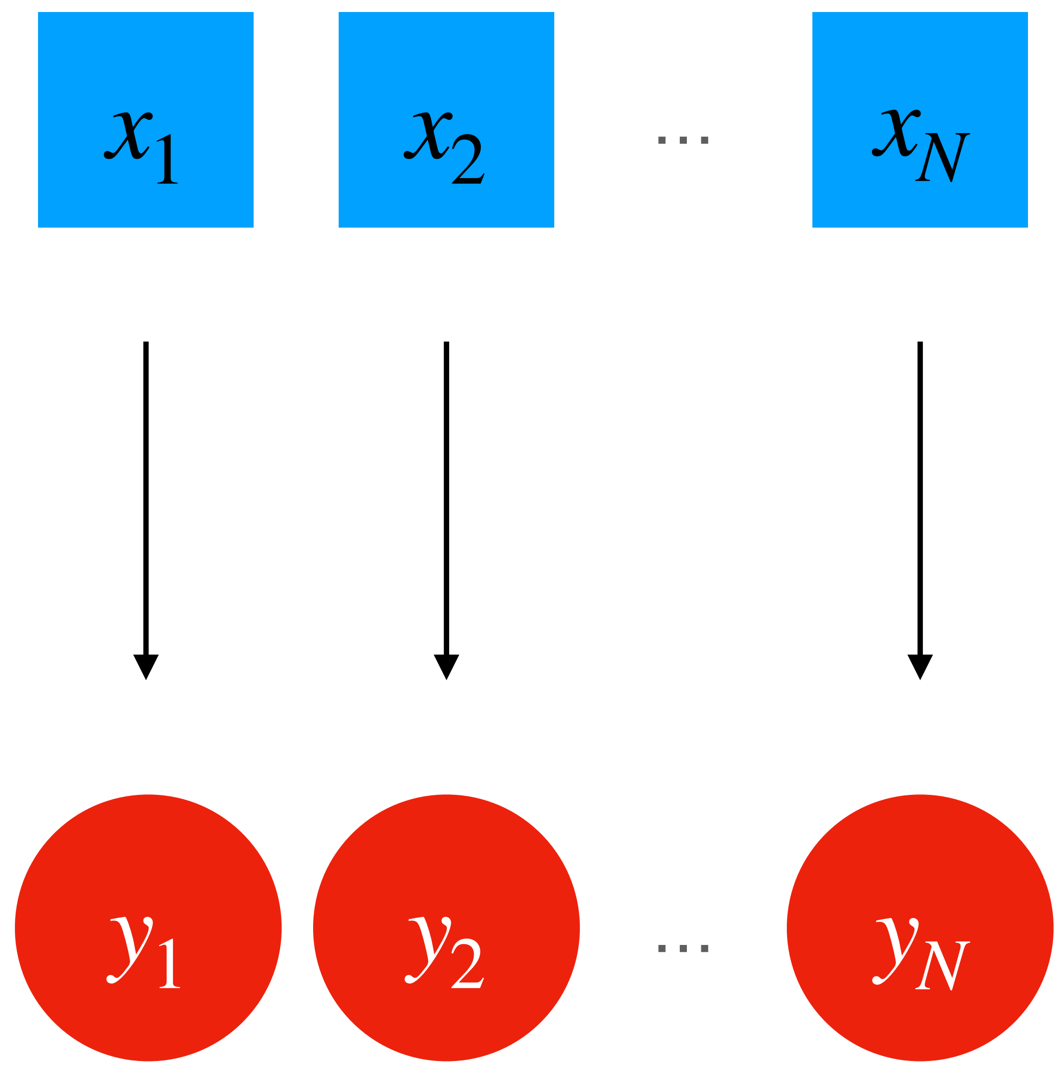
Todo

- Kurzeinführung (30-45min) Perzeption (1), Feed-Forward Netz (2; mit Aktivierungsfunktionen), Backprop (2-3)
- Fasttext (char-ngram word2vec); evtl. Konzept Byte-Pair-Encoding

Feed-Forward Networks
…on sequence data



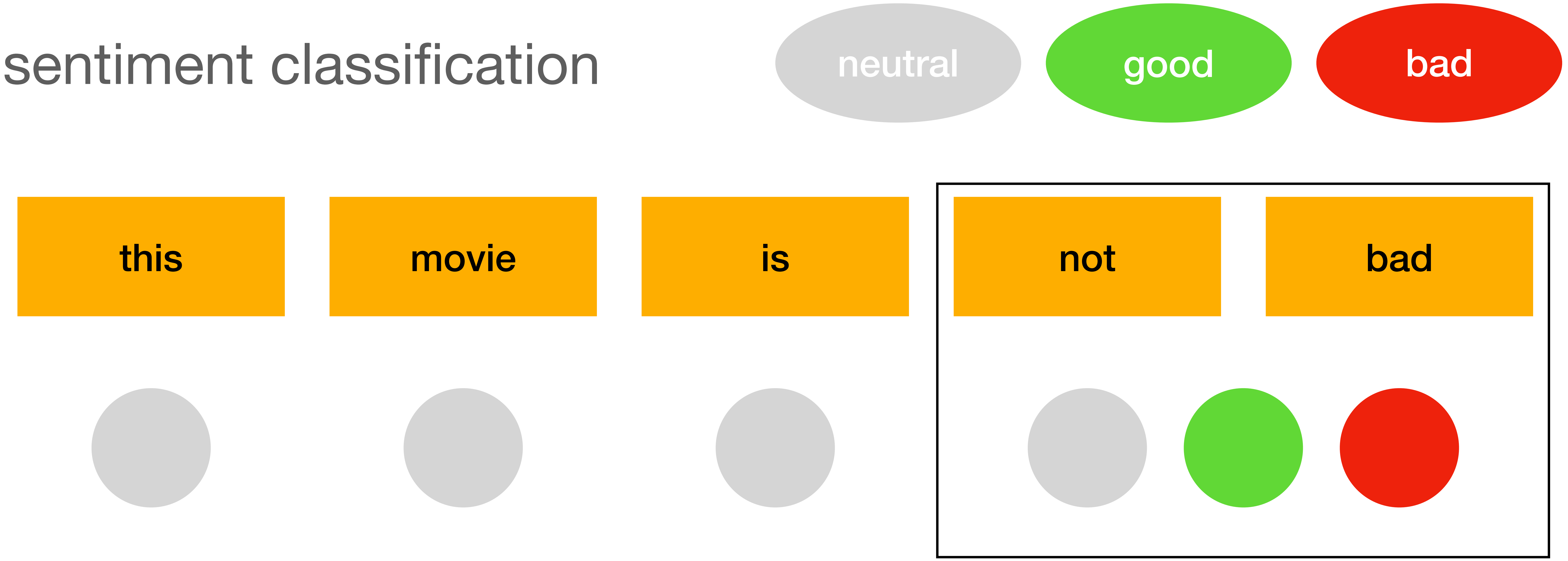
many-to-one



many-to-many

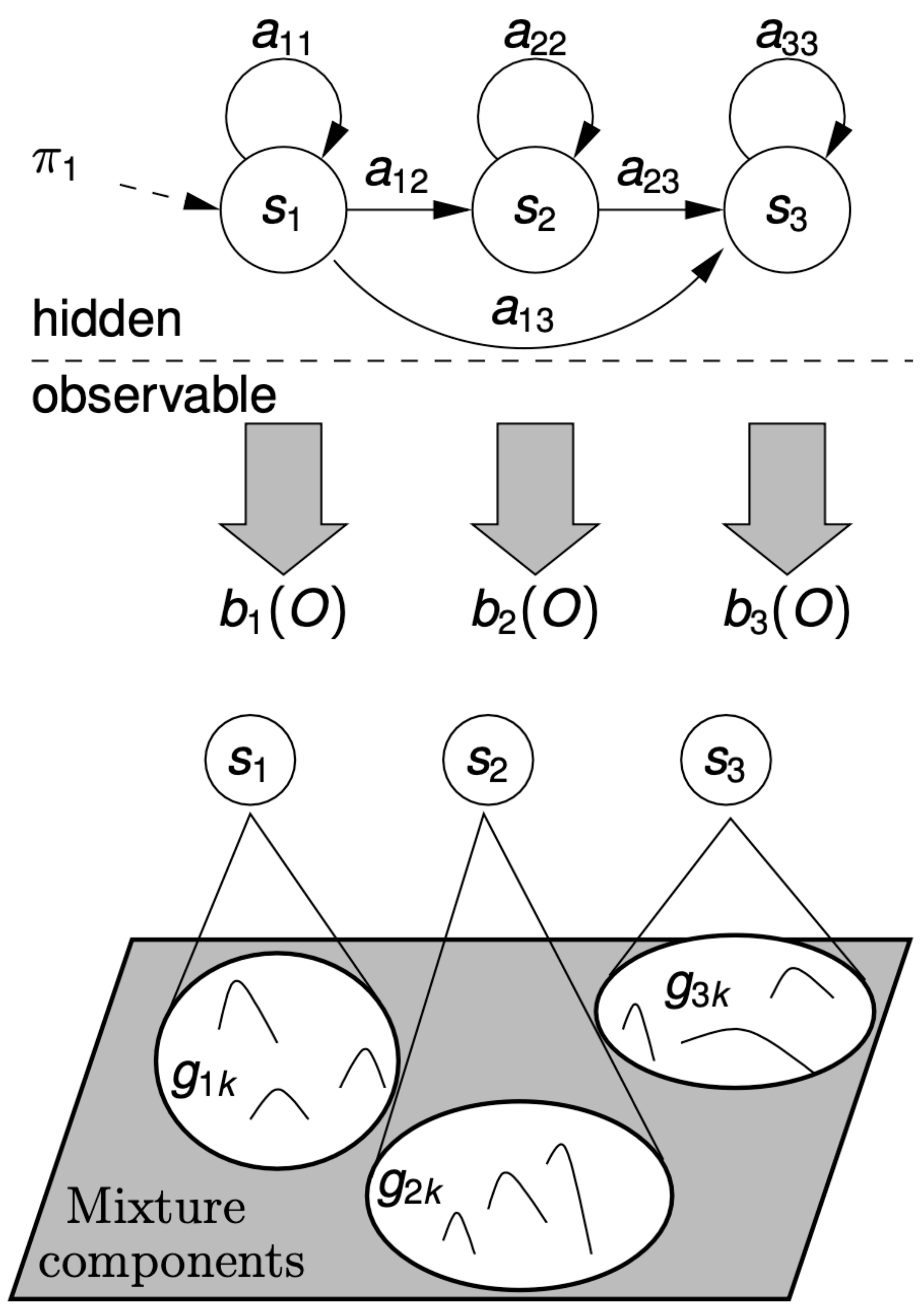
Context is Crucial

Example: sentiment classification



Solution: Use context windows to learn temporal relations

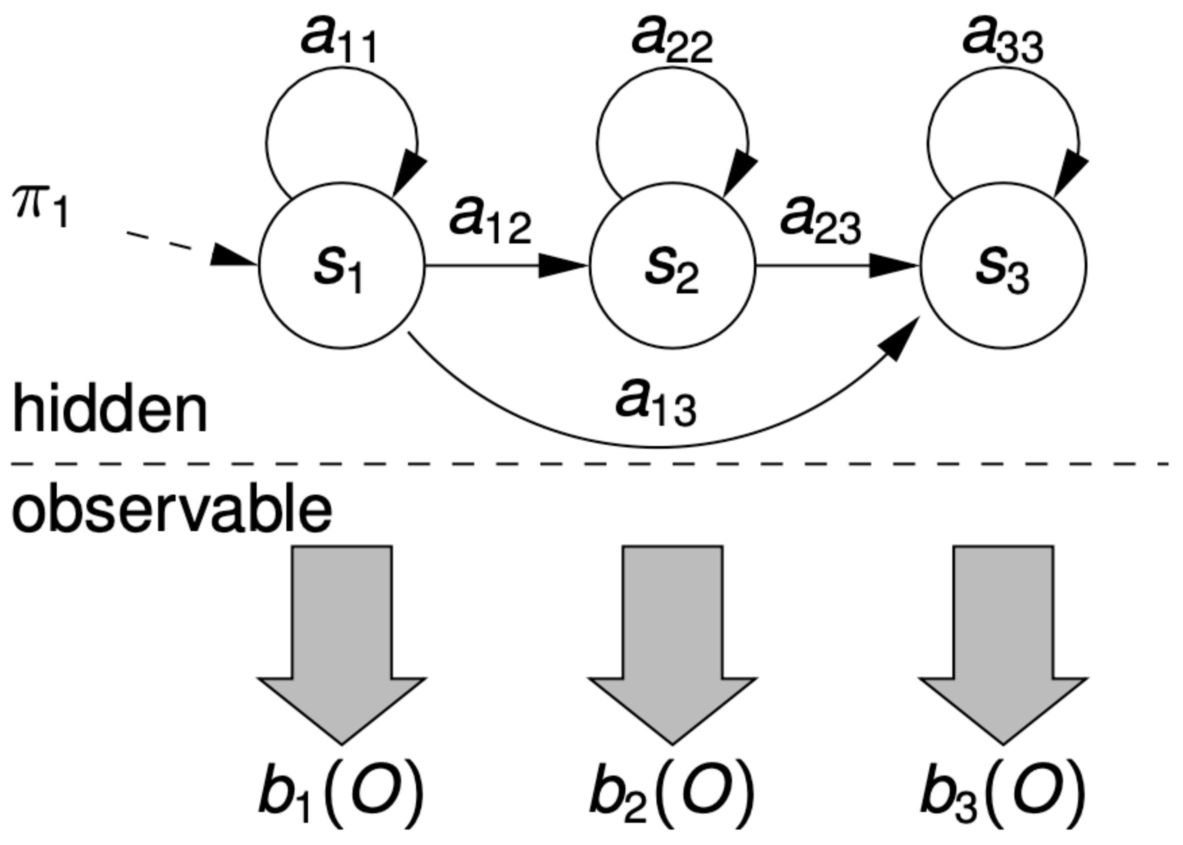
Connectionist HMM

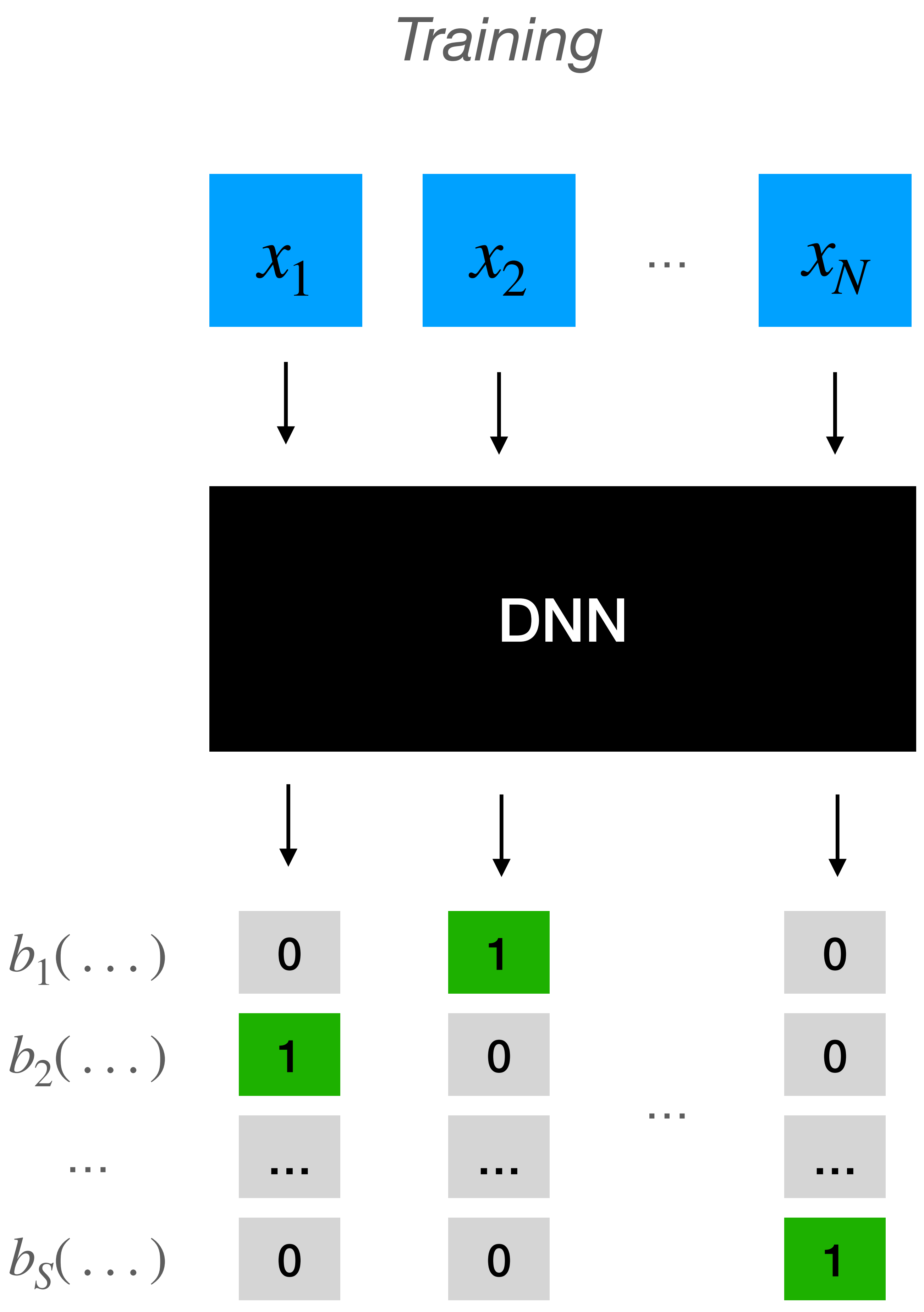


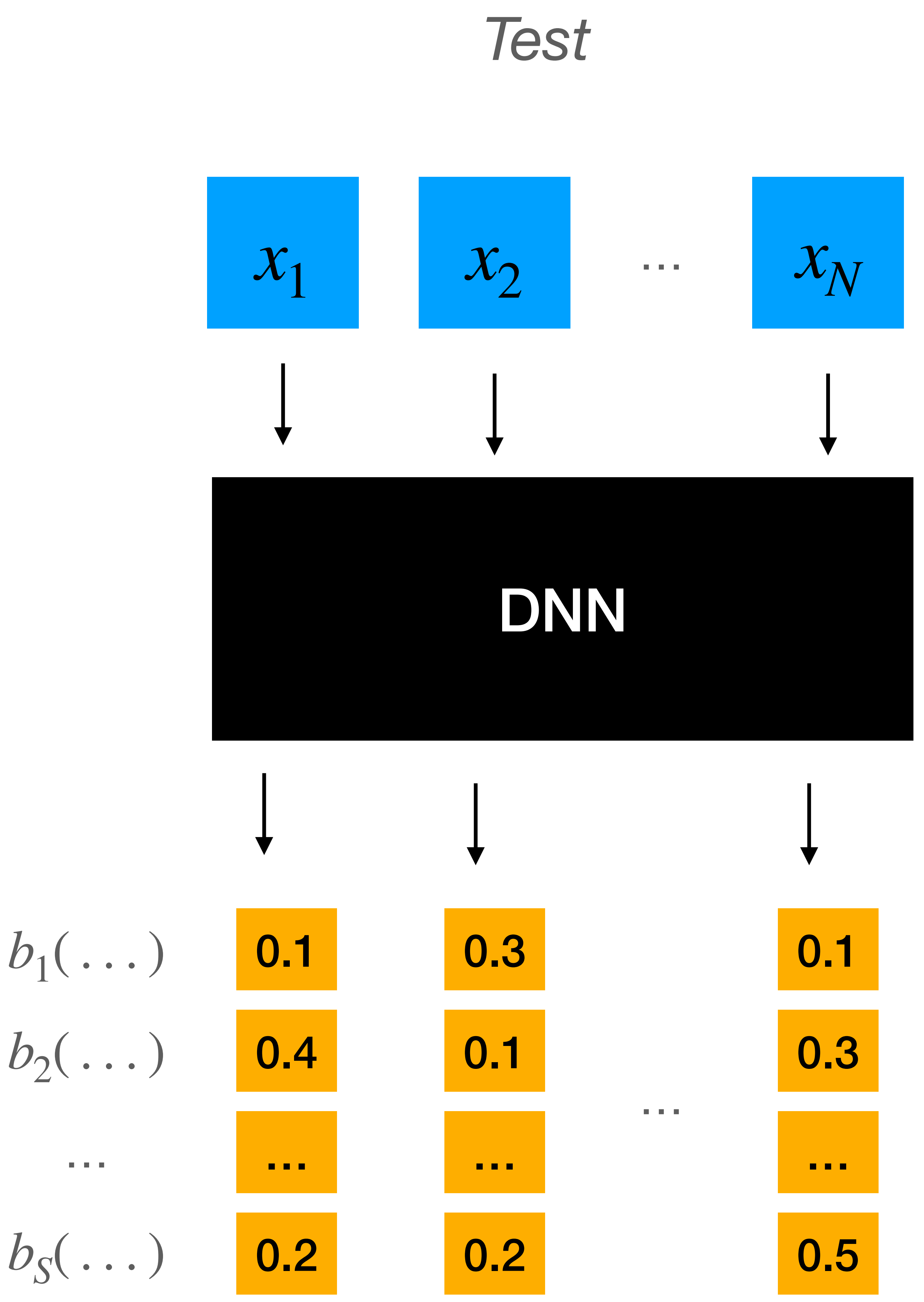
- *Observation*: At decoding time, we need emission probabilities of all (active) states
- *Problem*: GMMs don't generalize well
- *Idea*: Use NN to "predict" emission probs for all states at the same time

Renals et al., 1992: Connectionist Probability Estimation in the DECIPHER Speech Recognition System

Connectionist HMM







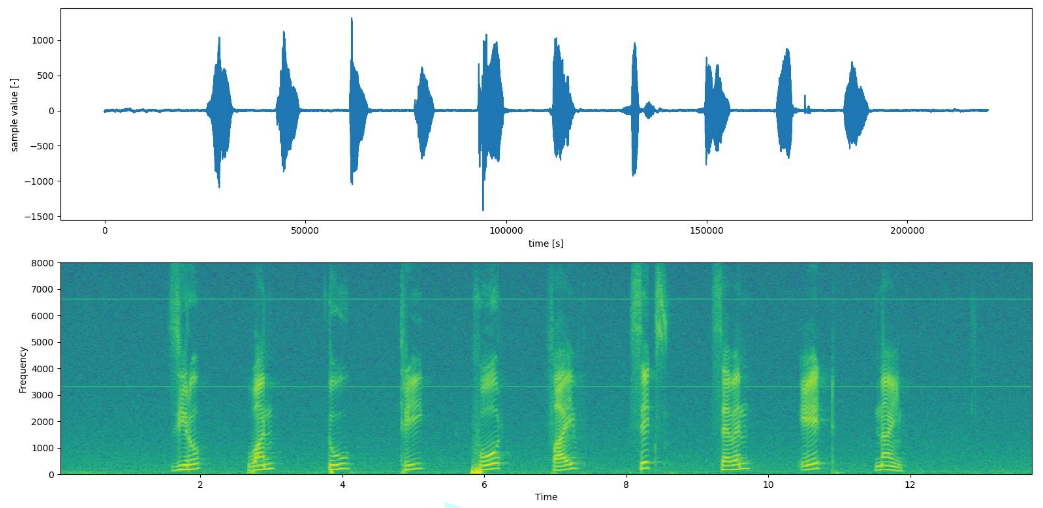
- requires alignment
- “one-hot encoding”
- cross-entropy loss

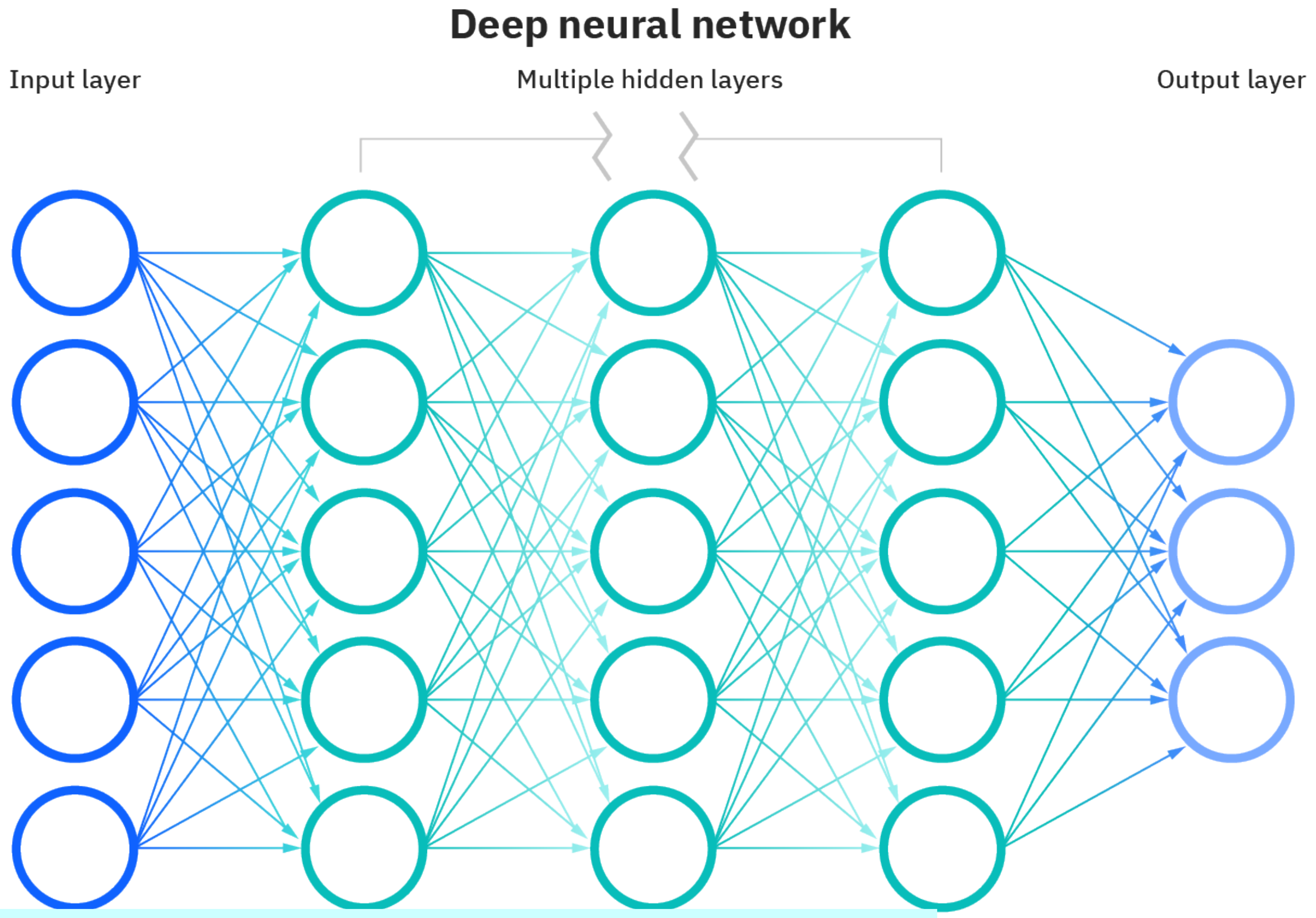
Word2Vec

- Recall n-gram probabilities: count observed ngrams, use back-off for unseen
- Bi-grams probabilities limit the context: $P(w_1, w_2, \ldots, w_n) = P(w_1) \prod_{i=2}^{N} P(w_i \mid w_{i-1})$
- How could we learn (not count) these?

Why word-embeddings?

[1, 1, 1, ..., -102, -99, -93]





Trommeln

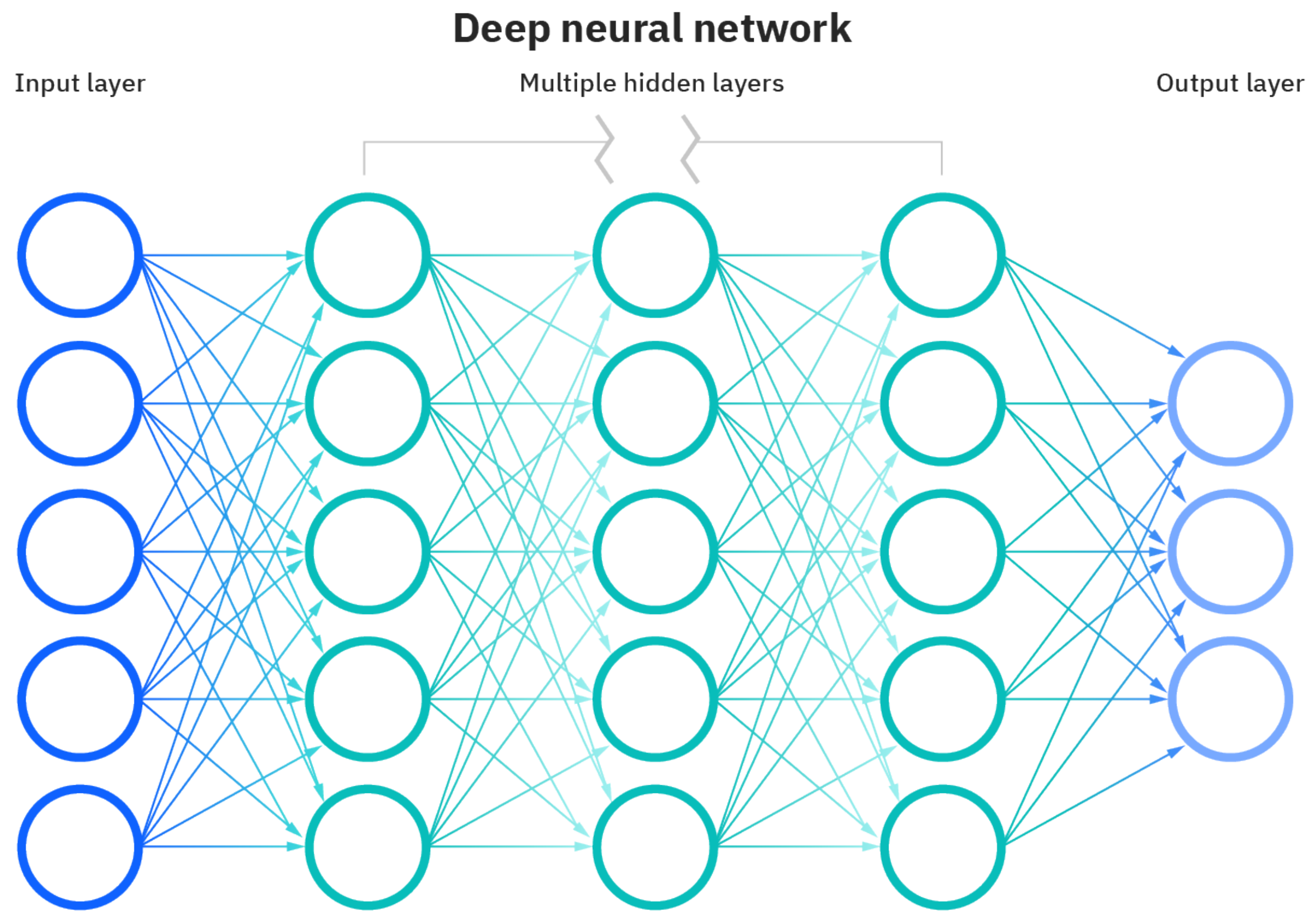
Trompete

Xylophon

```
[7.3945923e+03, 2.7395833e+03, 2.1257576e+04,
 4.2160831e+05, 2.9105340e+06, 4.6765578e+05],
[2.1494924e+04, 2.4632730e+04, 1.9541261e+05,
 3.1385060e+06, 8.9293340e+06, 4.5901940e+06],
[1.8762828e+05, 5.4574359e+04, 9.2627324e+03,
 2.8369732e+06, 2.3244162e+06, 2.3561962e+07]…
```

Why word-embeddings?

`Very enjoyable nonsense, this movie`



Neutral

Positiv

Negativ

One-hot representation

very	enjoyable	nonsense	this	movie
1	0	0	0	0
0	**1**	0	0	0
0	0	**1**	0	0
0	0	0	**1**	0
0	0	0	0	**1**

One-hot representation

very	enjoyable	nonsense	this	movie	film
1	0	0	0	0	
0	**1**	0	0	0	
0	0	**1**	0	0	
0	0	0	**1**	0	
0	0	0	0	**1**	

One-hot representation

Problems:

→ No relationships between words

(e.g., synonyms like film/movie)

→ Vocabulary size explodes

very	enjoyable	nonsense	this	movie	film
1	0	0	0	0	0
0	**1**	0	0	0	0
0	0	**1**	0	0	0
0	0	0	**1**	0	0
0	0	0	0	**1**	0
0	0	0	0	**0**	**1**

How to improve?

- fixed size vectors
- meaningful representations

dog	movie	film

How to improve?

- words
- meaning encoded in values
- **distributed representations**

dog	movie	film	
0.9	0.8	0.8	“moves”
0.0	0.6	0.6	art
0.9	0.8	0.2	US-English
0.0	0.0	1.0	creature
1.0	1.0	0.5	noun
…	…	…	

How would you automatically generate distributed representations?

Automatic generation of distributed representations

How?

dog
0.9
0.0
0.9
0.0
1.0
...

Behind the tree **hides** a **hairy, small Wolpertinger.**

Automatic generation of distributed representations

How?

dog
0.9
0.0
0.9
0.0
1.0
...

Behind the tree **hides** a **hairy, small Wolpertinger.**

A **small tabby cat hides** behind the barn.

Automatic generation of distributed representations

How?

dog
0.9
0.0
0.9
0.0
1.0
...

Behind the tree **hides** a **hairy, small Wolpertinger.**

A **small tabby cat hides** behind the barn.

A **scruff little dog hides** under the car.

Automatic generation of distributed representations

How?

dog
0.9
0.0
0.9
0.0
1.0
...

"You shall know a word by the company it keeps.“

J.R. Firth, A synopsis of linguistic theory 1930-55, 1957

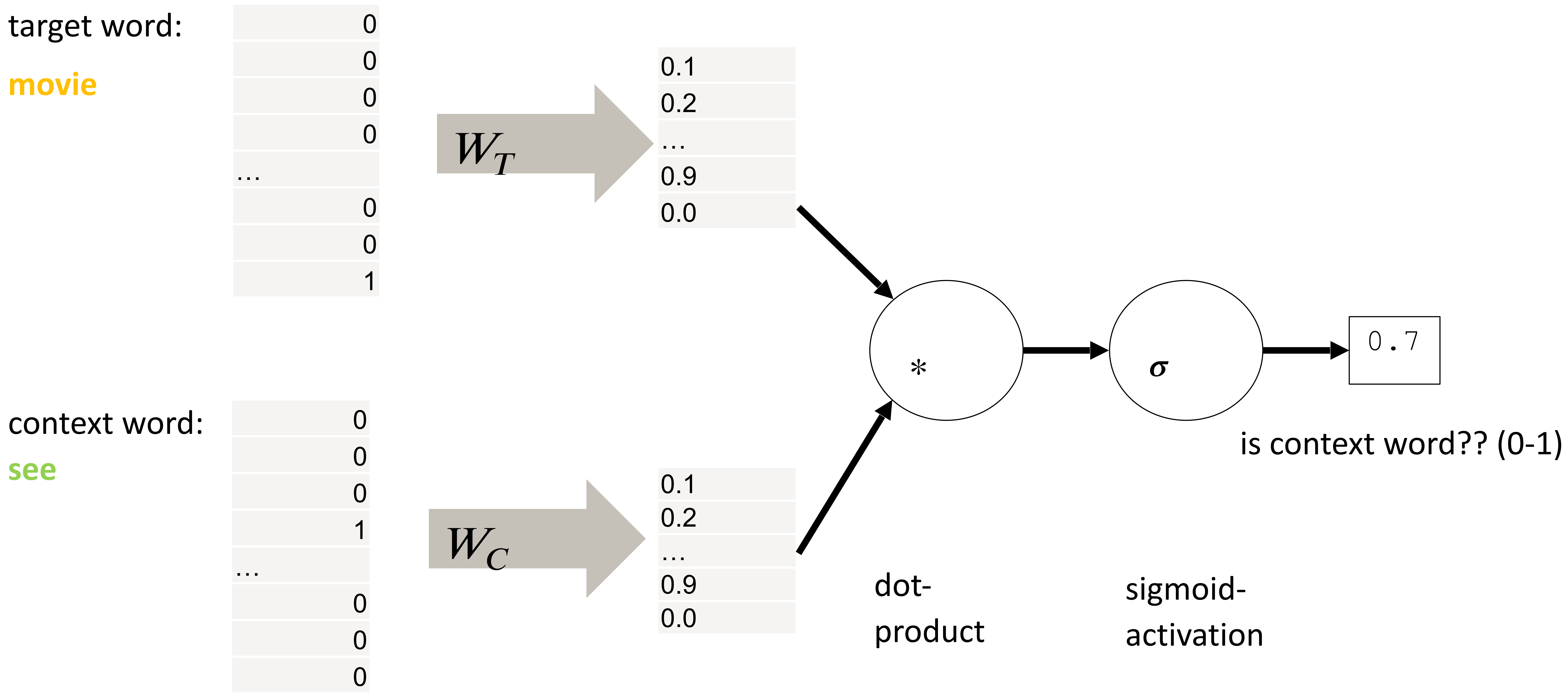
Word2Vec

- **General idea:**
  - Embeddings can be automatically learnt from data
  - Enough data represents covers many relationships
  - Include the context / context words

I would like a glass of apple juice. An apple grows on the tree. Yesterday, my father baked an apple pie.	She drank a glass of orange juice. There is an orange tree in the backyard. First, peel the orange.

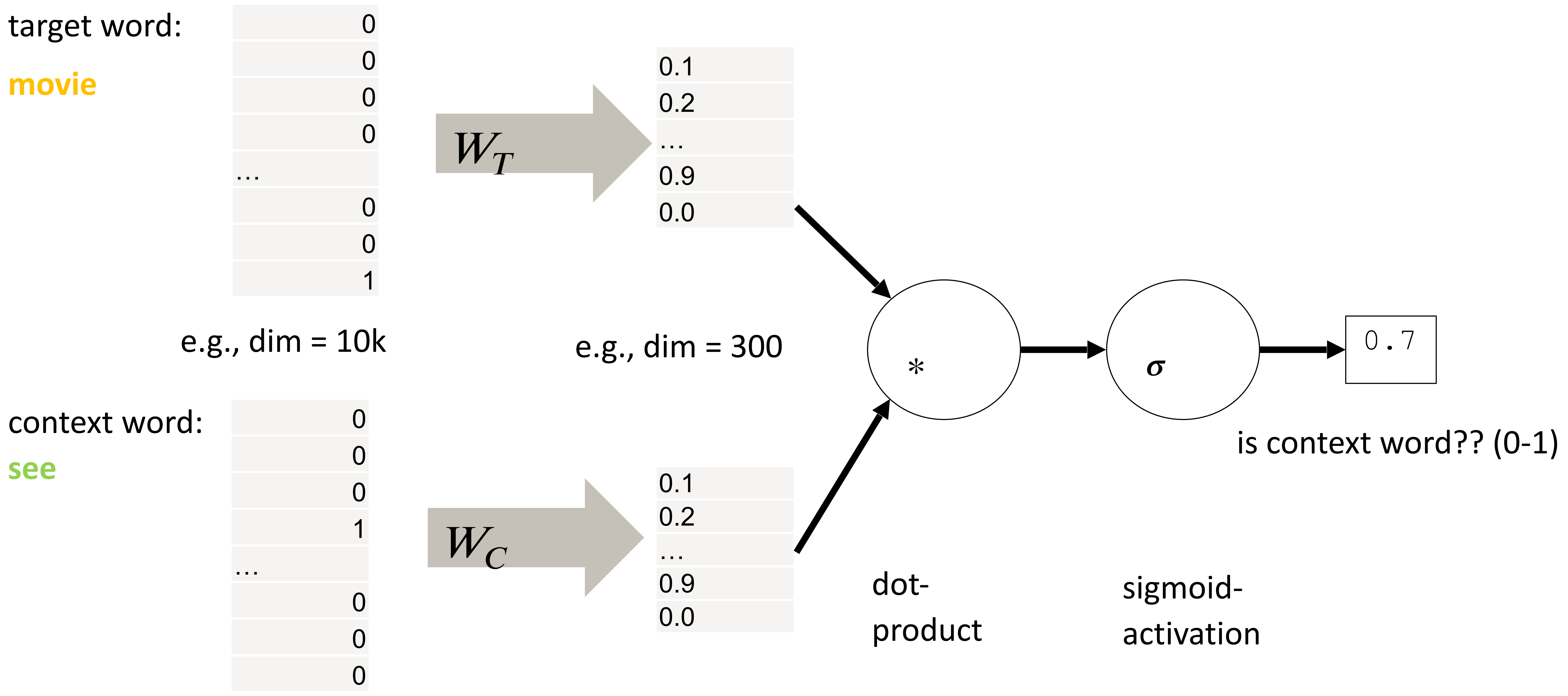
"Distributed Representations of Words and Phrases and their Compositionality", Tomas Mikolov et al., 2013

Word2Vec



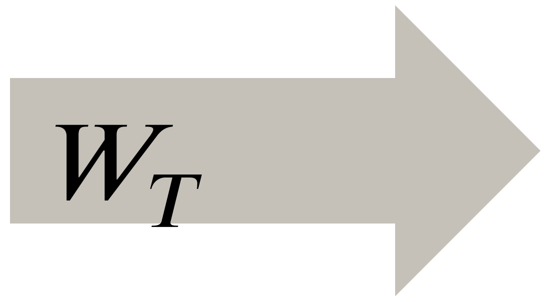
"Distributed Representations of Words and Phrases and their Compositionality", Tomas Mikolov et al., 2013

Word2Vec



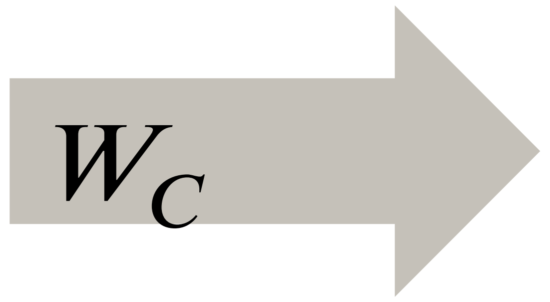
"Distributed Representations of Words and Phrases and their Compositionality", Tomas Mikolov et al., 2013

Word2Vec



Where to the projection matrices W_T and W_C come from?

→ They have to be learned!



Word2Vec
Skip-gram

- Skip-gram:
  - choose context words to generate positive samples
  - must be in relationship to target word, e.g., environment of +/- 2 words around the target word
  - Example:
    - Let‘s go $\boxed{\text{see a movie at the}}$ cinema →

Zielwort	Kontextwort	Label
movie	see	1

"Distributed Representations of Words and Phrases and their Compositionality”, Tomas Mikolov et al., 2013

Word2Vec

Negative sampling

- Negative sampling:
  - choose random words from the vocabulary
  - label as negative samples
  - Sampling frequency depending on the frequency of words in the dataset
  - Let‘s go **see** a **movie** at the cinema →

Zielwort	Kontextwort	Label
movie	see	1
movie	dear	0
movie	autotomy	0
movie	where	0

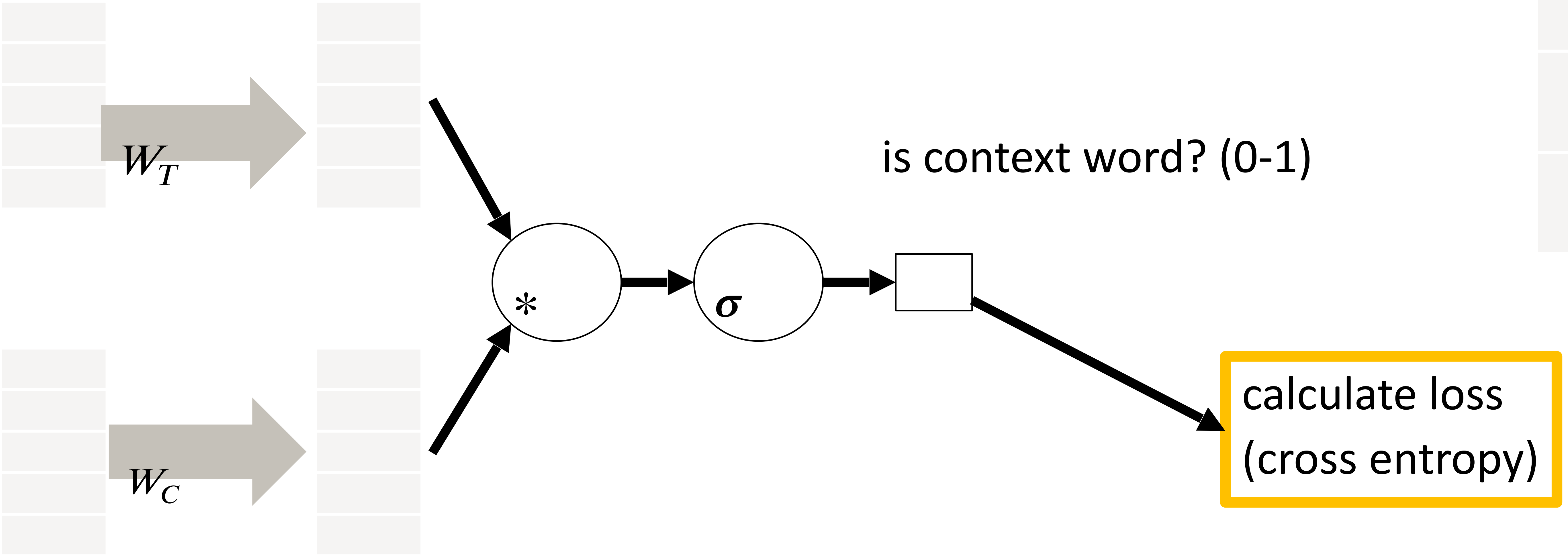
"Distributed Representations of Words and Phrases and their Compositionality”, Tomas Mikolov et al., 2013

Word2Vec

Training

Zielwort
movie
movie
movie
movie

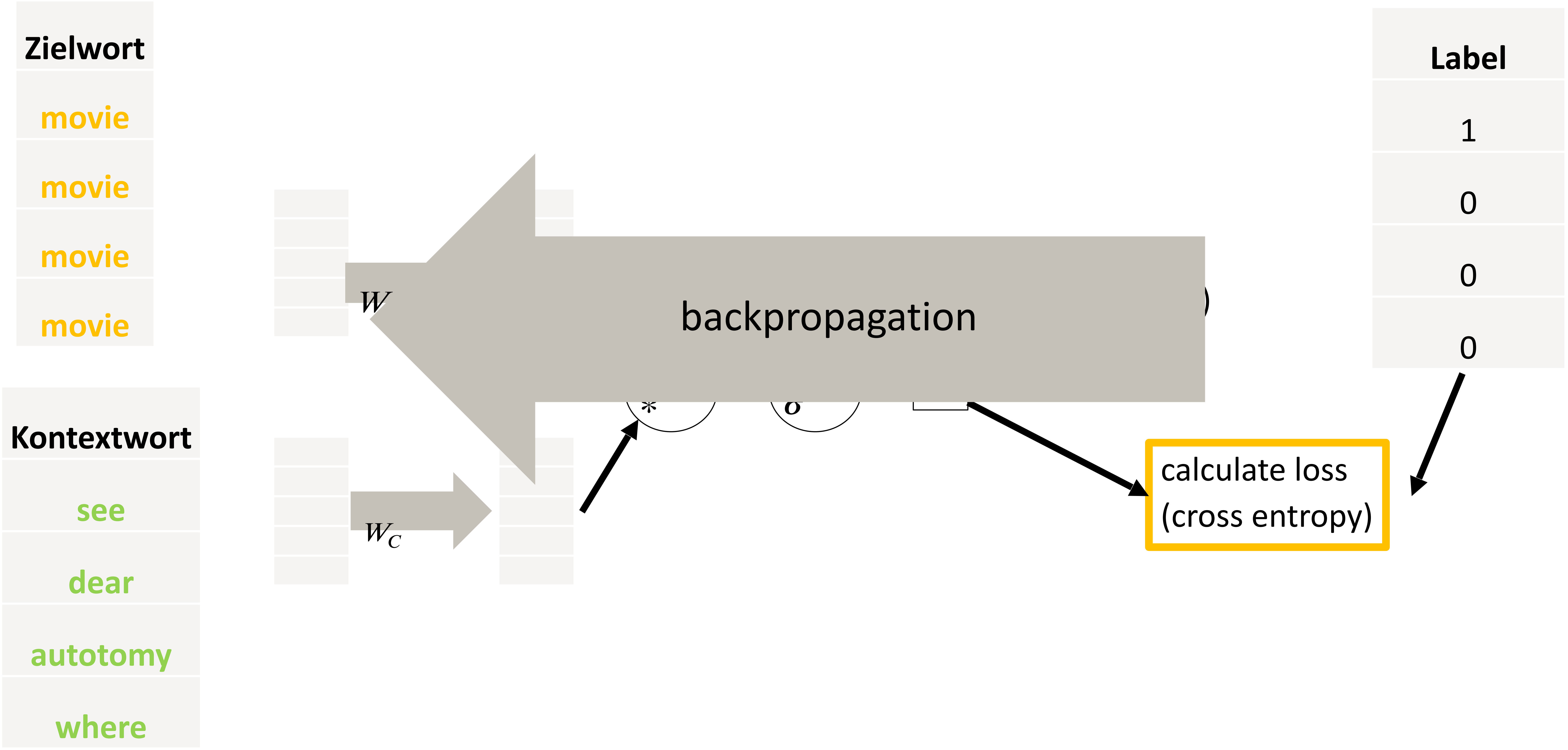
Kontextwort
see
dear
autotomy
where



Label
1
0
0
0

"Distributed Representations of Words and Phrases and their Compositionality", Tomas Mikolov et al., 2013

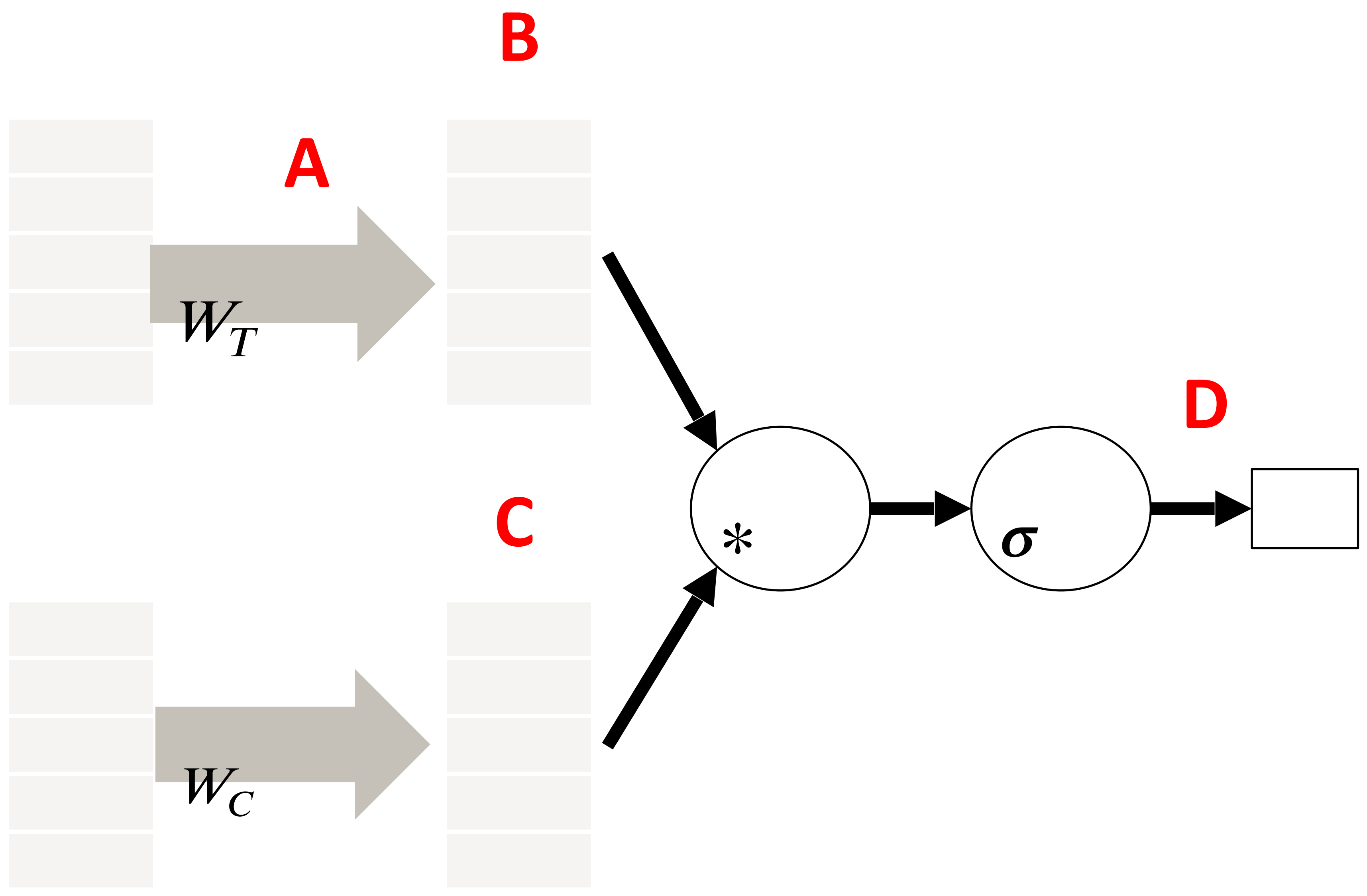
Word2Vec
Training



"Distributed Representations of Words and Phrases and their Compositionality", Tomas Mikolov et al., 2013

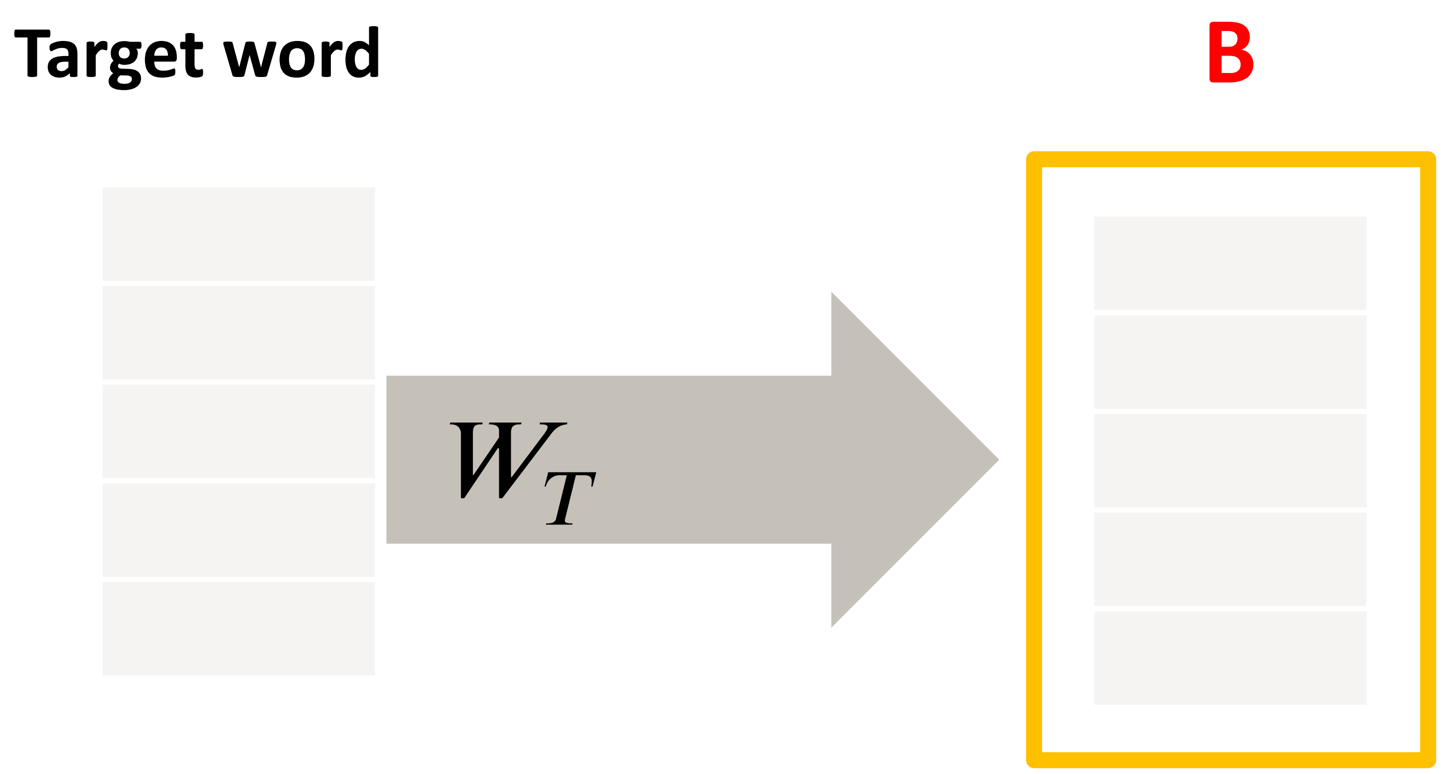
Word2Vec

Where will embeddings be extracted?



"Distributed Representations of Words and Phrases and their Compositionality", Tomas Mikolov et al., 2013

Word2Vec



- independent of vocabulary size
- smaller dimensionality than vocabulary size
- representation of relationships between words

"Distributed Representations of Words and Phrases and their Compositionality", Tomas Mikolov et al., 2013

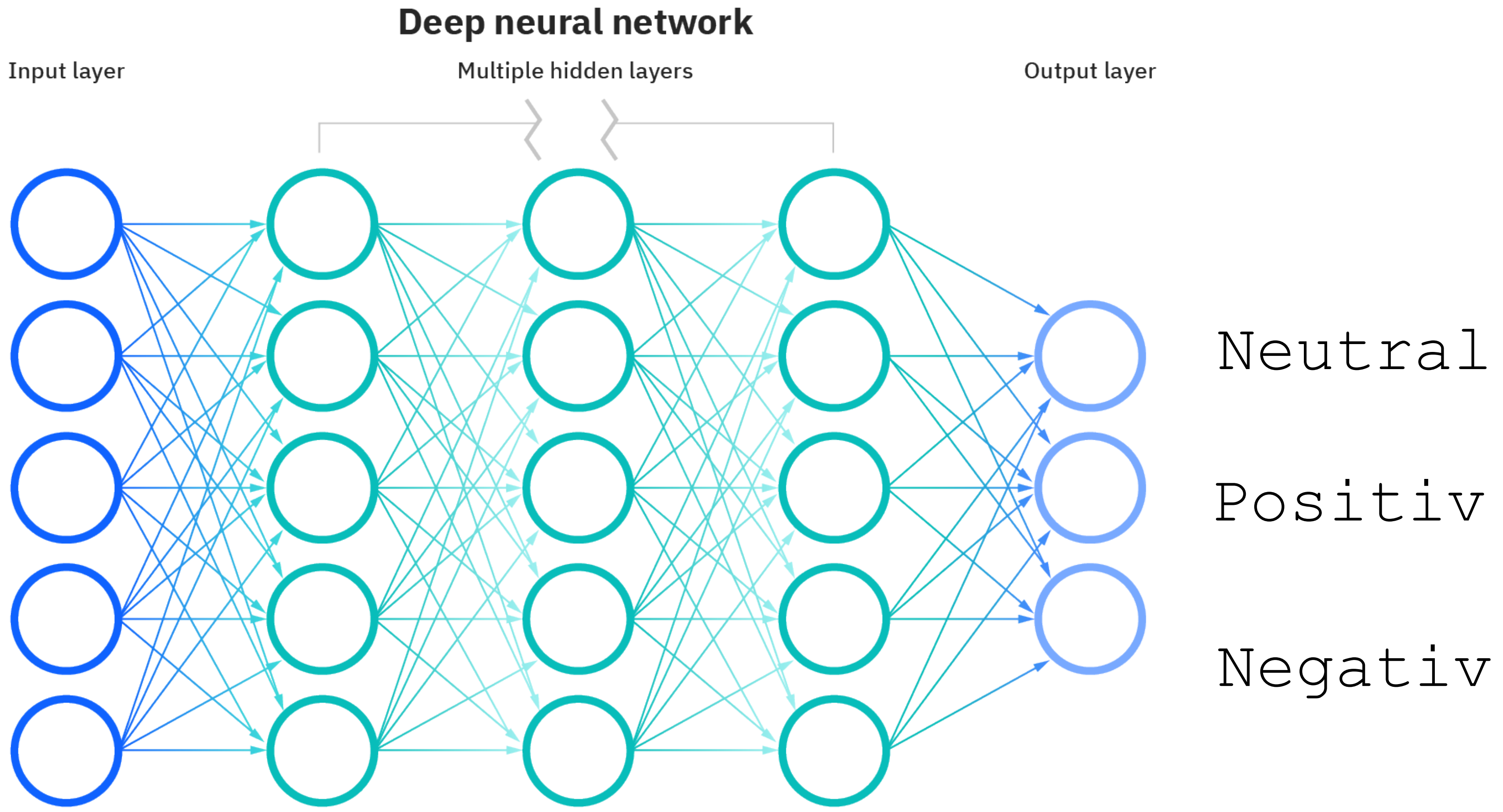
Word2Vec

Problem solved

`Very enjoyable nonsense, this movie`

W_T

very	enjoyable	nonsense	this	movie
0.6	0.01	0.03	0.3	0.01
0.02	0.9	0.32	0.88	0.12
0	0.2	0.25	0	0.25
0.22	0.33	0.8	0.1	0.2
0.88	0.65	0.23	0.24	0.1
0.01	0.23	0.65	0.44	0.9



Attention:

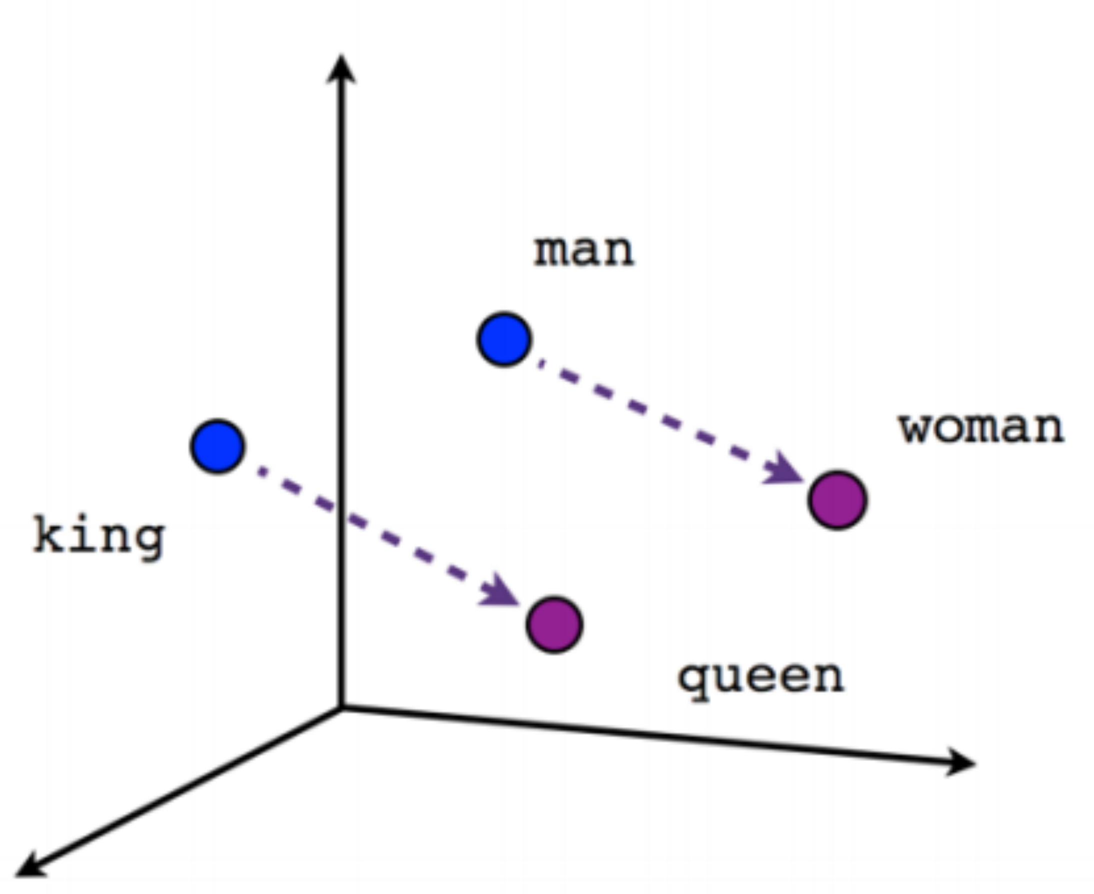
W_T is usually pre-trained on large databases, only "fine-tuning" necessary later

"Distributed Representations of Words and Phrases and their Compositionality", Tomas Mikolov et al., 2013

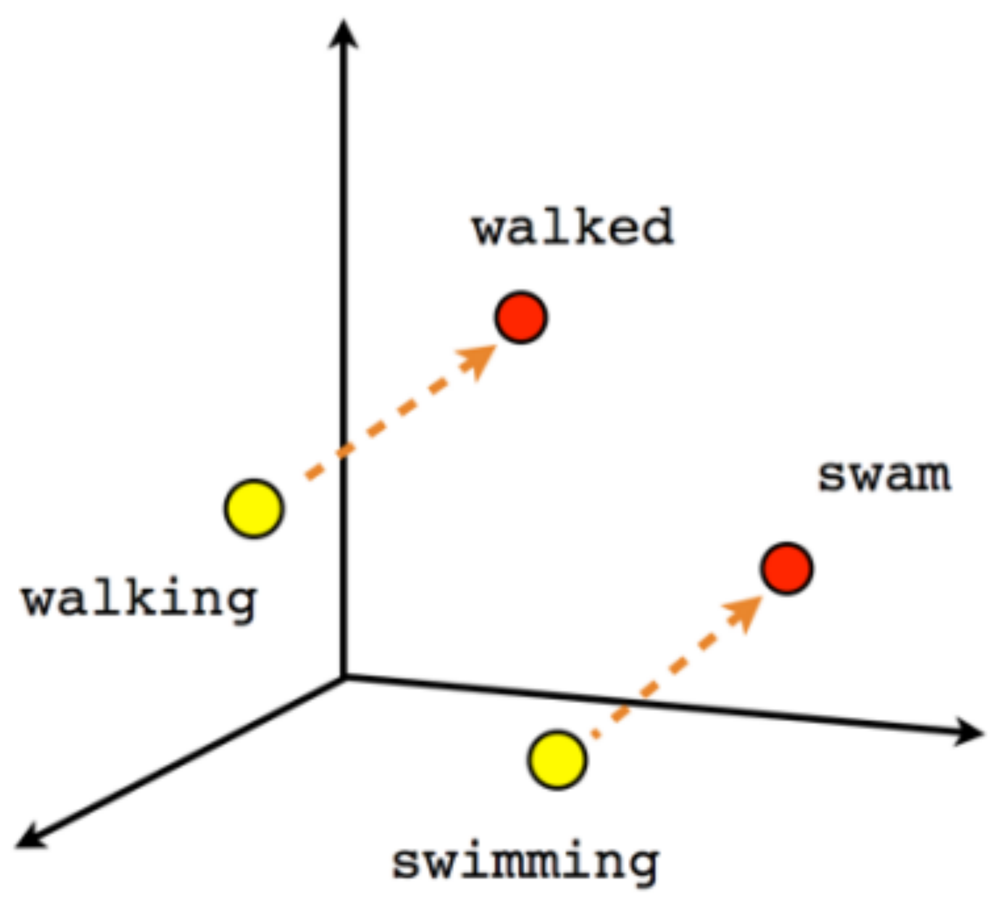
Word2Vec

Visualization of semantic relationships of words;

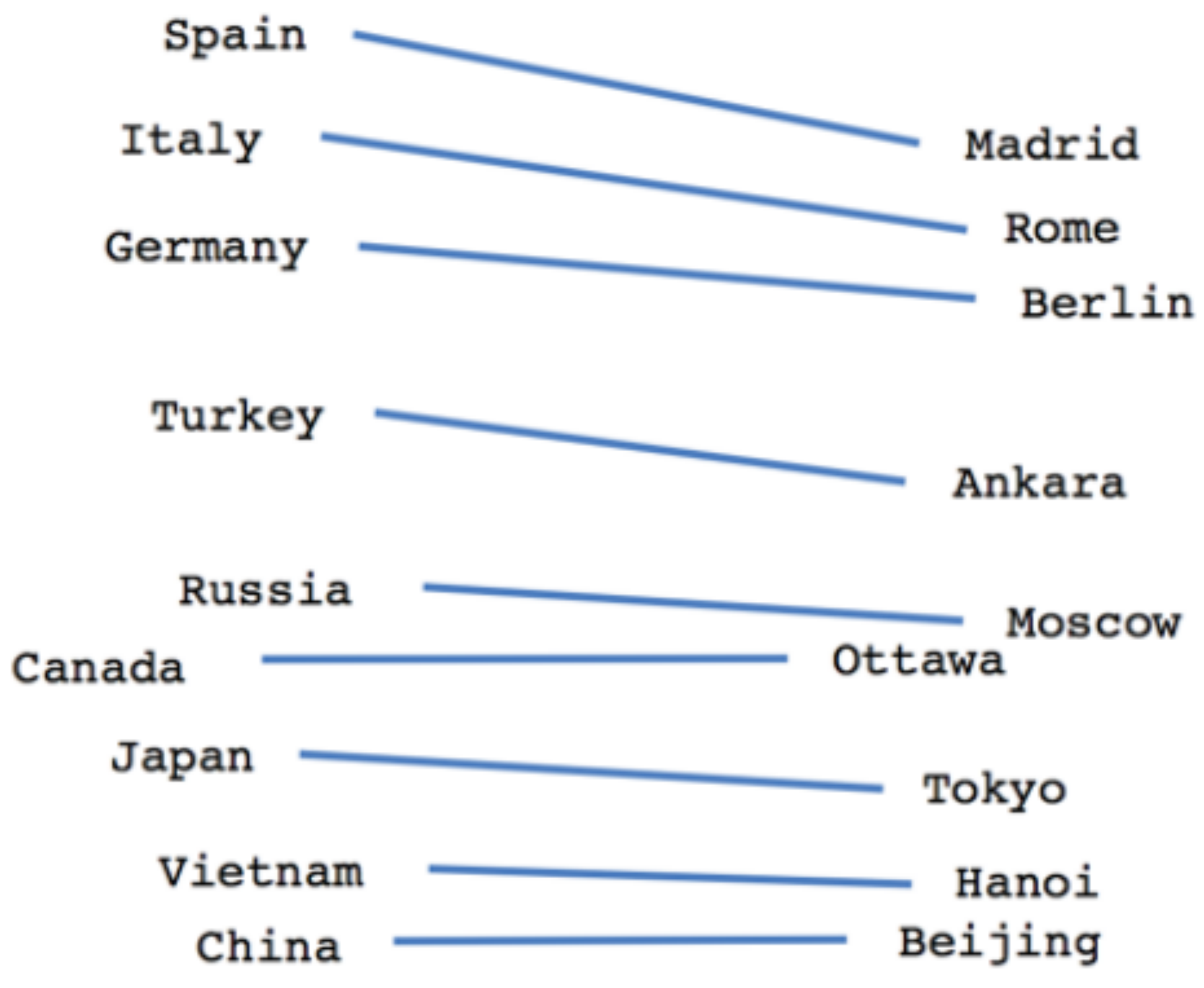
Good embeddings encode semantic relationships



Male-Female



Verb tense

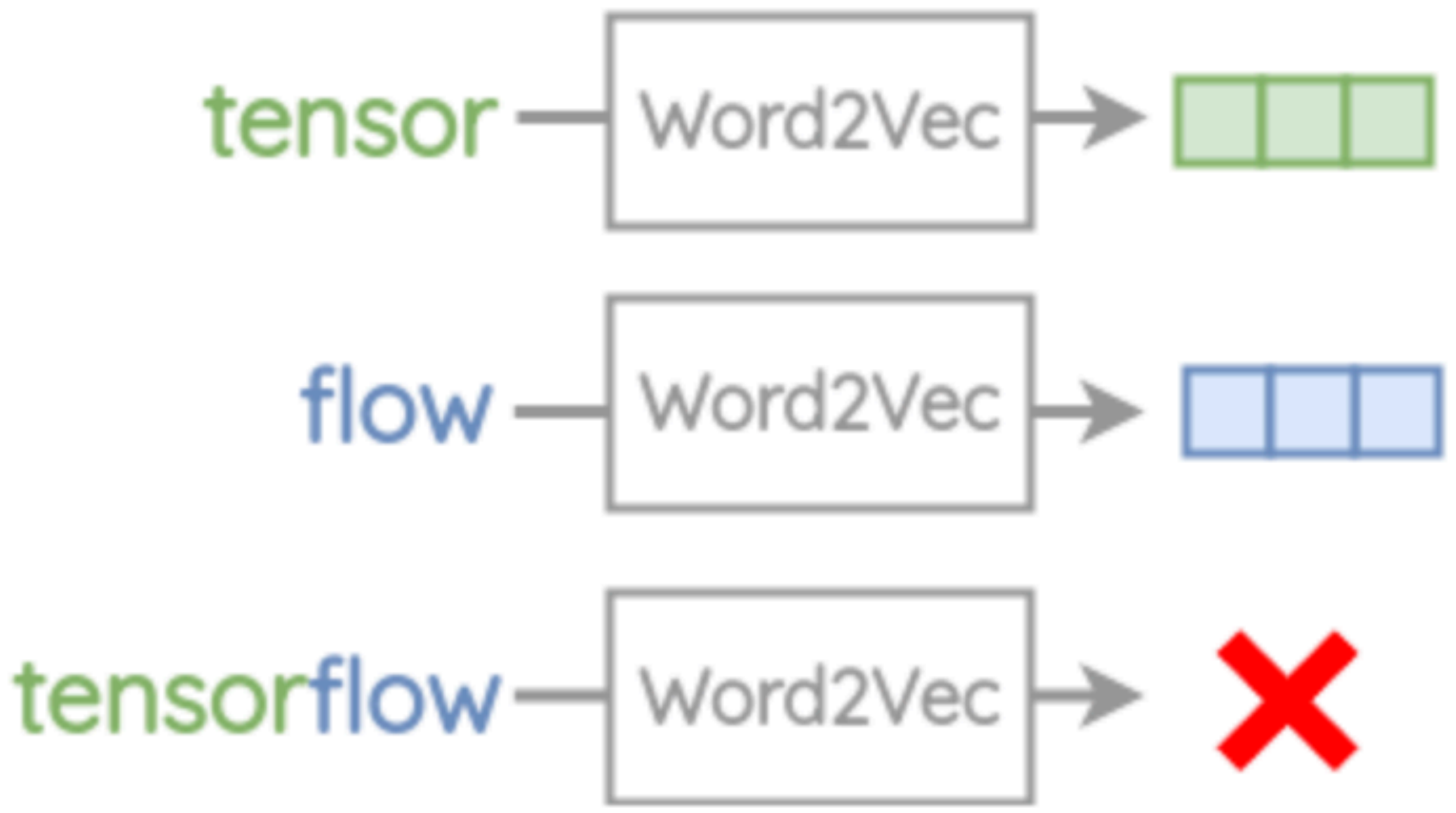


Country-Capital

https://www.tensorflow.org/images/linear-relationships.png

Word2Vec
Limitations

- Out-of-Vocabulary
  - Also: typos, compounds



- Morphology
  - Also: slang, shortening

Figures: https://amitness.com/2020/06/fasttext-embeddings/

FastText

- Observation: Words are inherently a problem (OOV, typos, morphology, etc.)
- Solution:
    - Use sub-words (character n-grams) instead
    - Re-use skip-gram and negative sampling
    - *Bojanowski 2017:* 3-6 grams

Bojanowski, Grave, Joulin and Mikolov, 2017: Enriching Word Vectors with Subword Information

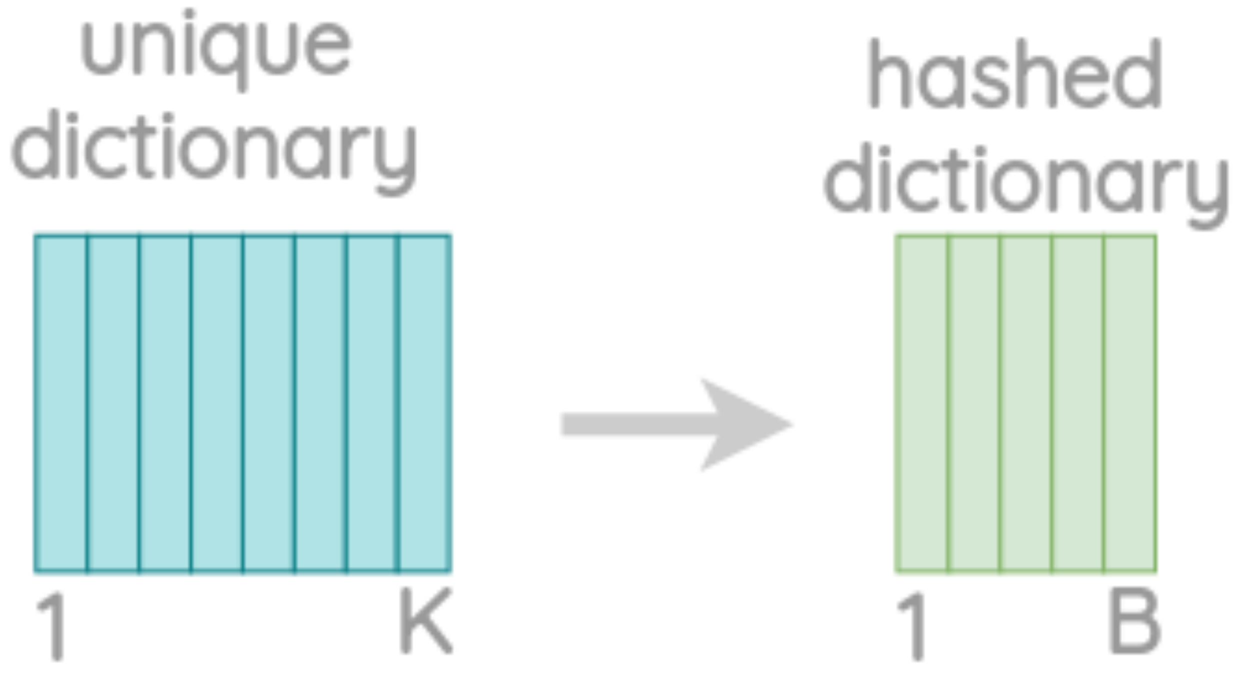
FastText

Step 1: Decompose to Sub-Words

- Enclose any word in the training set with <>

- Extract character n-grams with sliding window

- Use hashing to reduce memory; count for bin instead of actual token



FastText

Step 2: Modify Skip-Gram & Negative Sampling

- Sum up the n-gram vectors *and* the vector of the actual word

- Sample positive and negative context (word vectors)

- Compute dot-product for actual and negative context, and use SGD to update parameters

FastText

Insights

- *Improves* performance on **syntactic word analogy tasks** significantly for morphologically rich language like Czech and German

Singular/plural

cat → cats

dog → ?

Base/Comparative

good → better

rough → ?

	word2vec-skipgram	word2vec-cbow	fasttext
Czech	52.8	55.0	**77.8**
German	44.5	45.0	**56.4**
English	70.1	69.9	**74.9**
Italian	51.5	51.8	**62.7**

- *Degrades* performance on **semantic analogy tasks** compared to Word2Vec.

man → king

woman → queen

	word2vec-skipgram	word2vec-cbow	fasttext
Czech	25.7	**27.6**	27.5
German	66.5	**66.8**	62.3
English	**78.5**	78.2	77.8
Italian	52.3	**54.7**	52.3

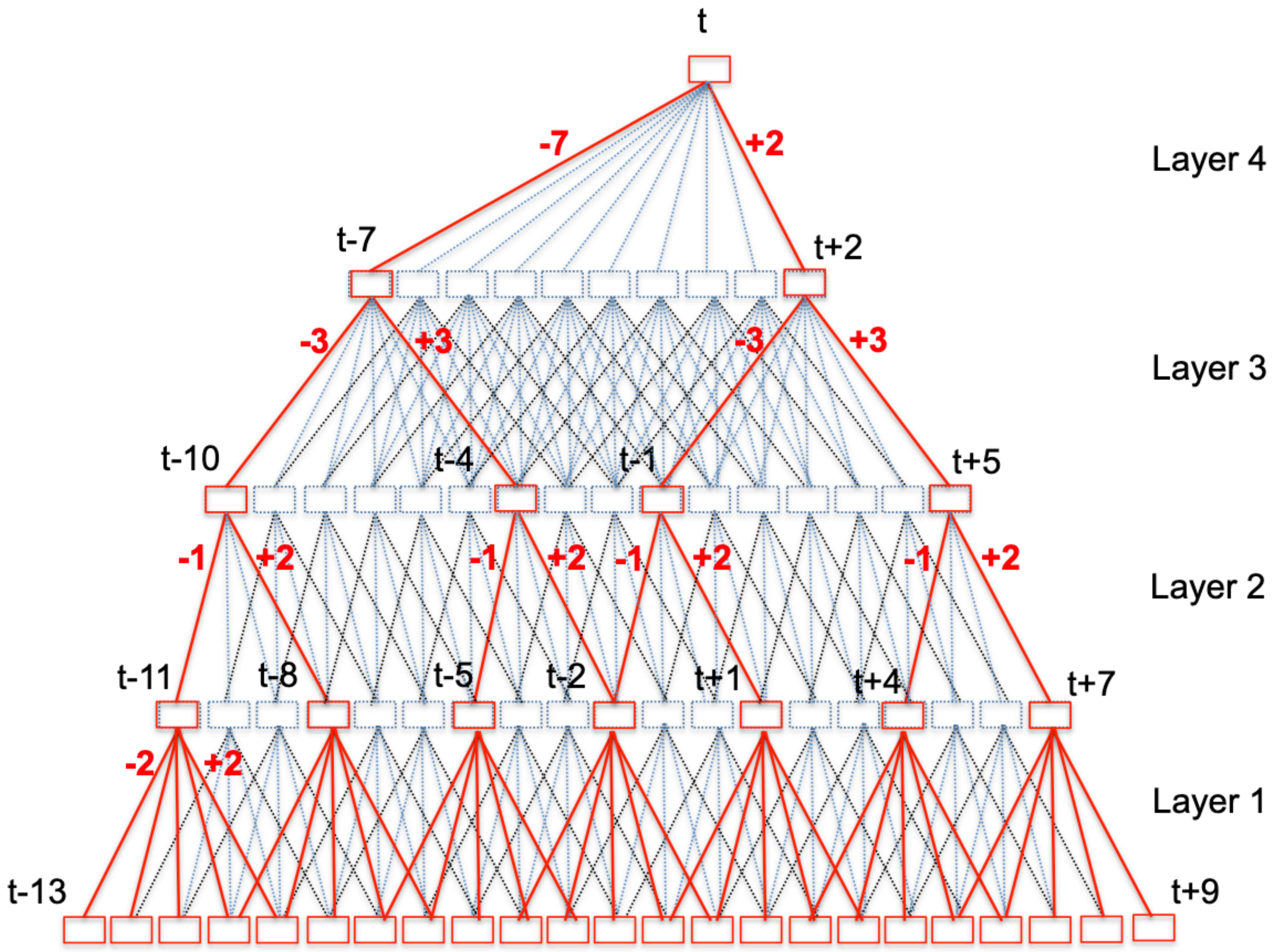
FastText
Insights

- Using sub-word information with character-ngrams has better performance than CBOW and skip-gram baselines on word-similarity task.
- Representing out-of-vocab words by summing their sub-words has better performance than assigning null vectors.

		skipgra	cbo	FT null	FT char
Arabic	WS353	51	52	54	**55**
	GUR35	61	62	64	**70**
German	GUR65	78	78	81	**81**
	ZG222	35	38	41	**44**
English	RW	43	43	46	**47**
	WS353	72	73	71	**71**
Spanish	WS353	57	58	58	**59**
French	RG65	70	69	75	**75**
Romani	WS353	48	52	51	**54**
Russian	HJ	69	60	60	**66**

Time-delay Neural Networks

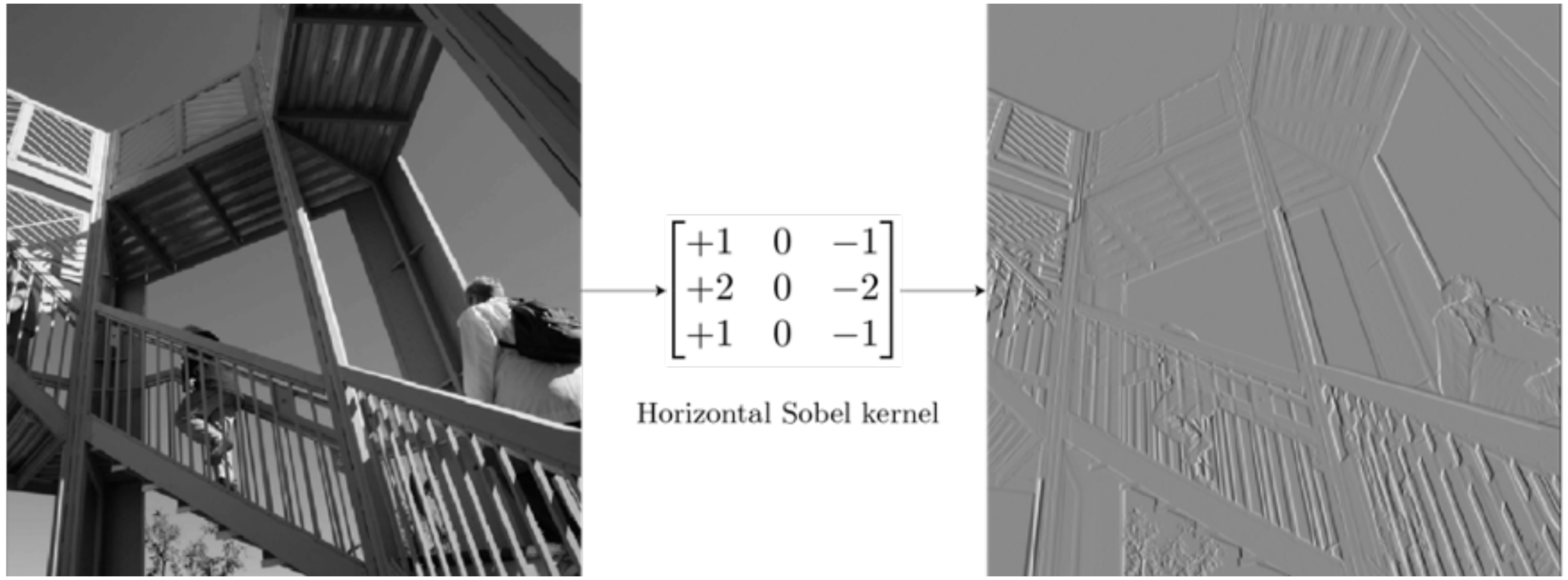
Waibel et al. 1989



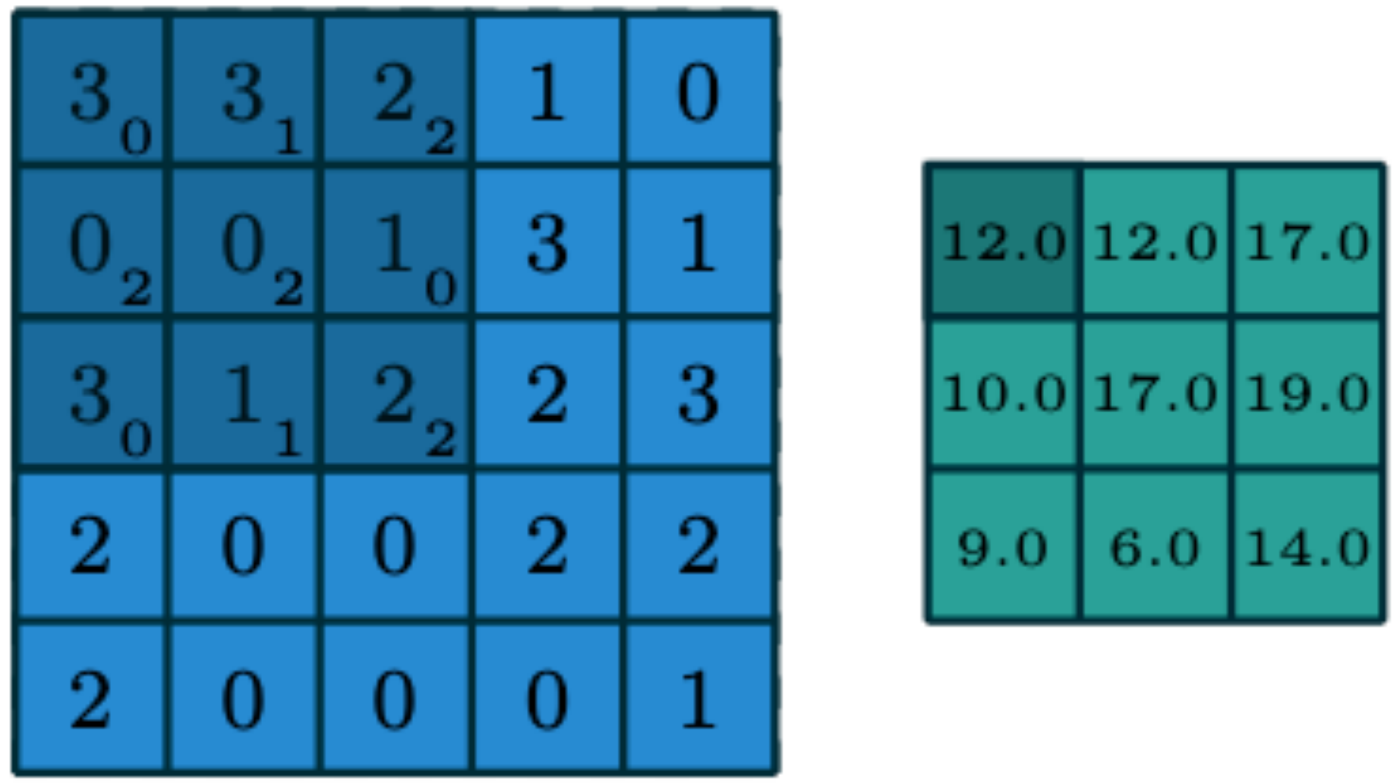
- Frames are typically features (MFCC, word embeddings, …)
- Concatenate frames to form contexts
- Go from narrow to wide with layers
- Lower layers learn "local" features
- Higher layers learn temporal relationships

Peddinti et al., 2015. "A time delay neural network architecture for efficient modeling of long temporal contexts"

ConvNets



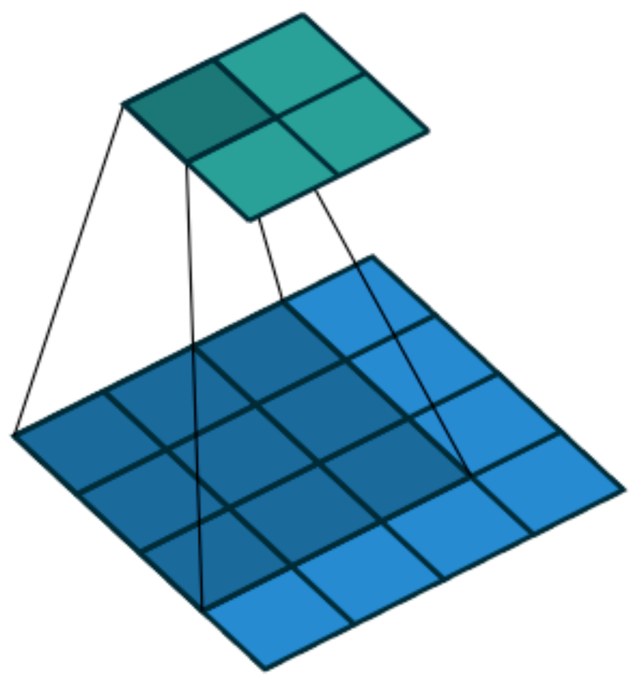
- Motivation:
  - Convolution of signal with special kernels can be a great feature
  - Well established in computer graphics (eg. Sobel edge detector)
- 1D time series: 1D convolutions
  - "within-feature convolutions"
- 2D image: 2D convolutions
  - "across-feature convolutions"

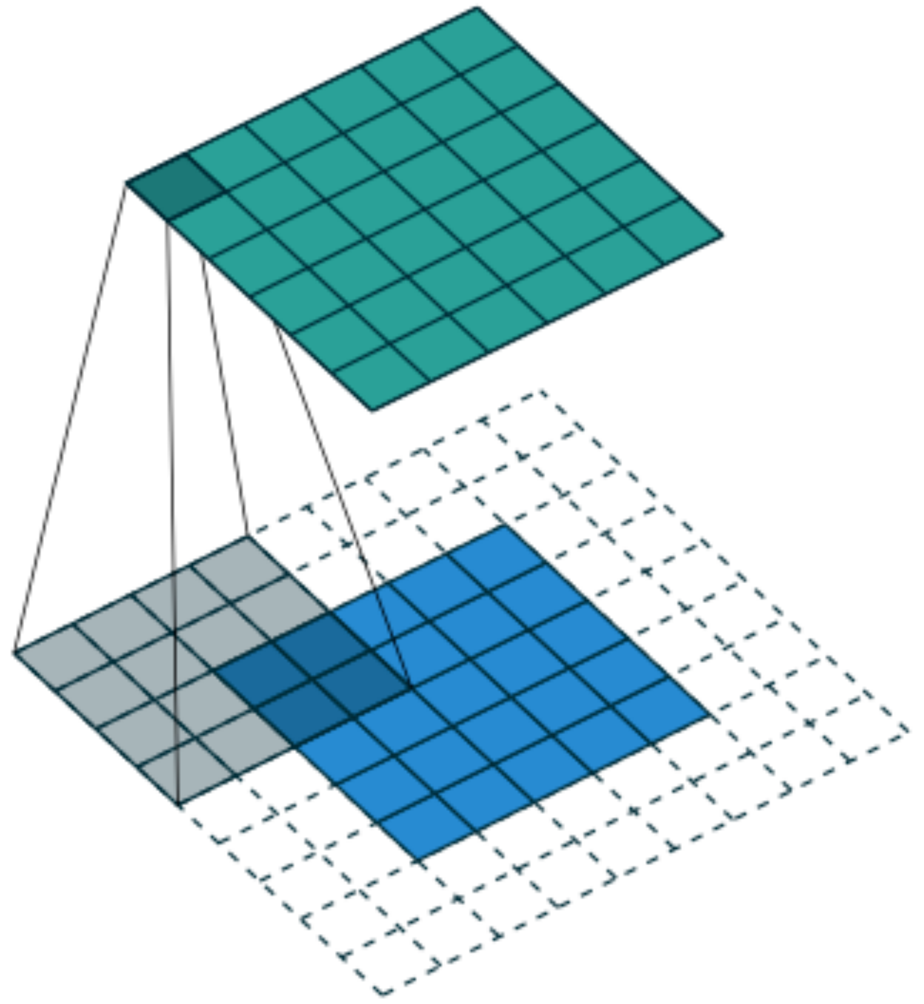


Dumoulin, V. and Visin, F. "A guide to convolution arithmetic for deep learning"

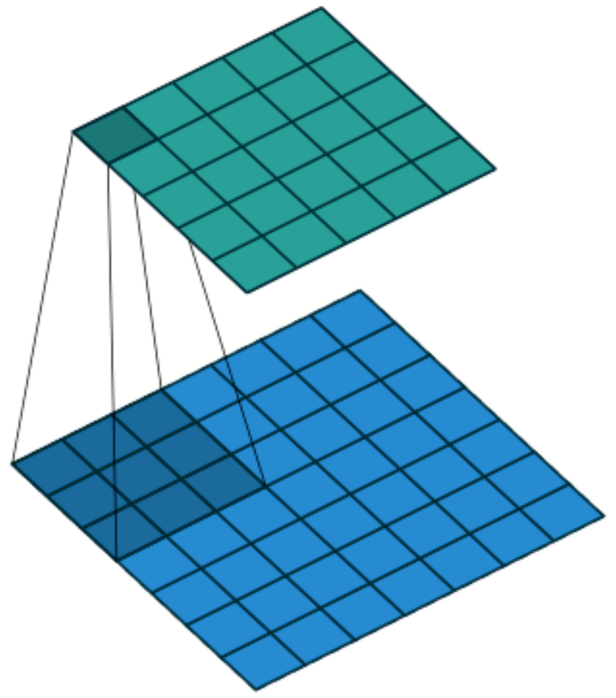
ConvNets Building Blocks

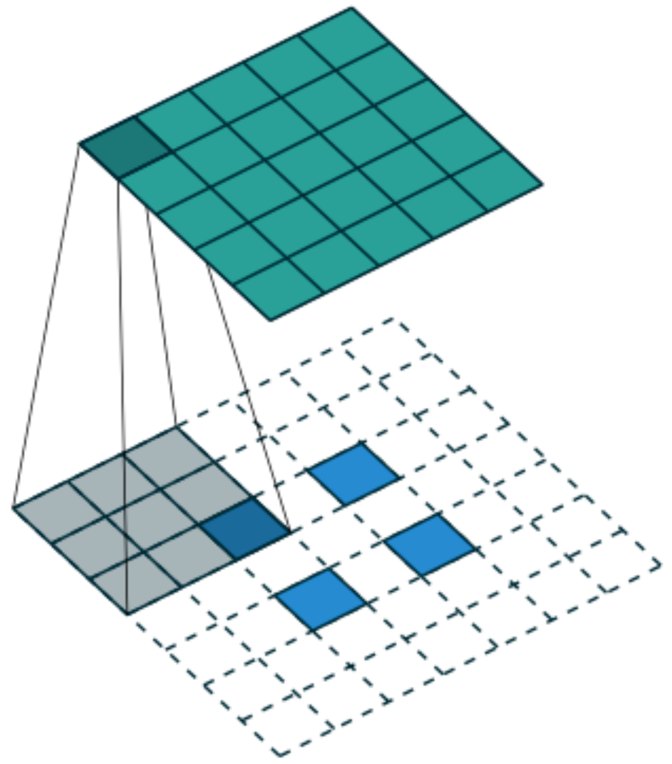
- Convolution:
    - kernel size, eg. 3x3, 1x3
    - stride, step size, eg. 1
    - padding, what to do at the edges? eg. zero-pad





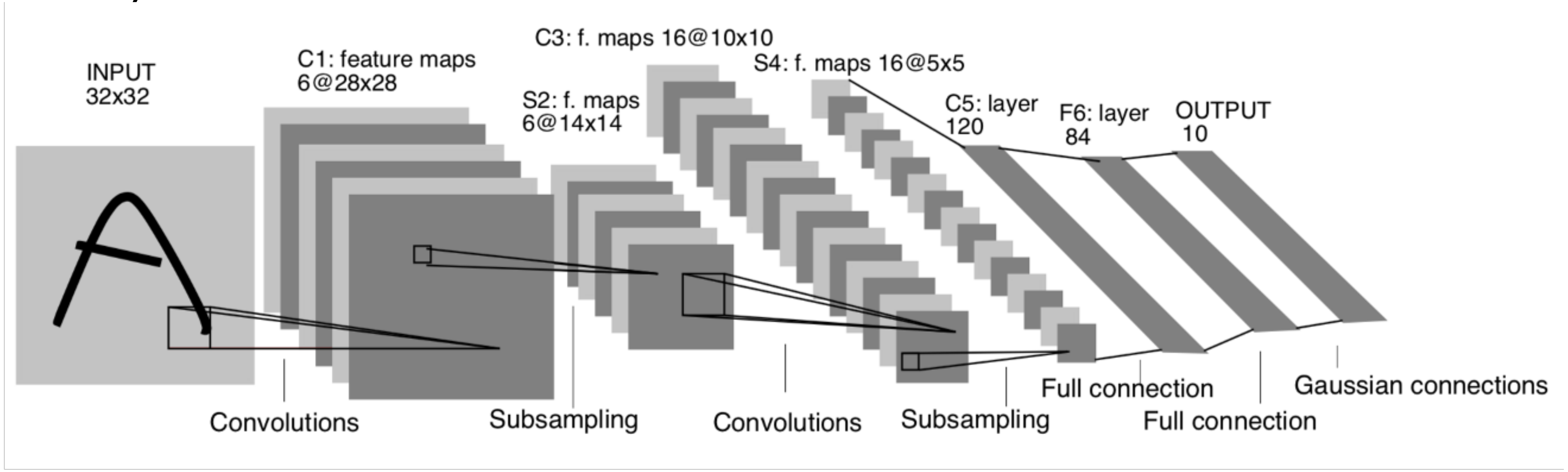
- Pooling to reduce/increase resolution
    - average, max, …





Historic Note

- TDNN (1989): effectively 1D convolutions
- LeCun at al., 1998: LeNet-5 architecture, MNIST error rate 0.8% (regular FF: 4.7%)



Recap

Feed-Forward Networks for Sequence Data

- Use context windows, eg. by concatenation
- Use embeddings for discrete symbols (which effectively use 1-hot)
- Use convolutions (1D, 2D) to extract temporal structure from context window
- Works for all modalities:
  - Audio: eg. MFB, MFCC
  - Text: Word Vectors