Machine Translation 2: Statistical MT: Neural MT and Representations

Ondřej Bojar
bojar@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University, Prague

Outline of Lectures on MT

1. Introduction.
   - Why is MT difficult.
   - MT evaluation.
   - Approaches to MT.
   - Document, sentence and esp. word alignment.
   - Classical Statistical Machine Translation.
     - Phrase-Based MT.
2. Neural Machine Translation.
   - Neural MT: Sequence-to-sequence, attention, self-attentive.
   - Sentence representations.
   - Role of Linguistic Features in MT.

Outline of MT Lecture 2

1. Fundamental problems of PBMT.
2. Neural machine translation (NMT).
   - Brief summary of NNs.
   - Sequence-to-sequence, with attention.
   - Transformer, self-attention.
   - Linguistic features in NMT.

Summary of PBMT

Phrase-based MT:

- is a log-linear model
- assumes phrases relatively independent of each other
- decomposes sentence into contiguous phrases
- search has two parts:
  - lookup of all relevant translation options
  - stack-based beam search, gradually expanding hypotheses

To train a PBMT system:

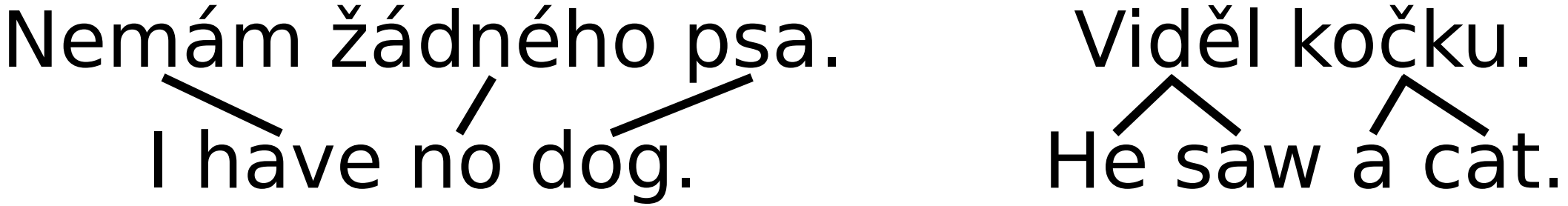
1. Align words.
2. Extract (and score) phrases consistent with word alignment.
3. Optimize weights (MERT).

1: Align Training Sentences

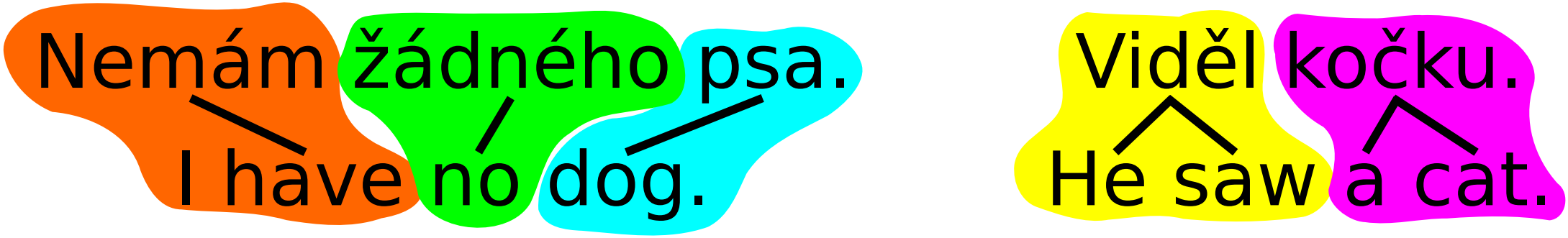
Nemám žádného psa. Viděl kočku.

I have no dog. He saw a cat.

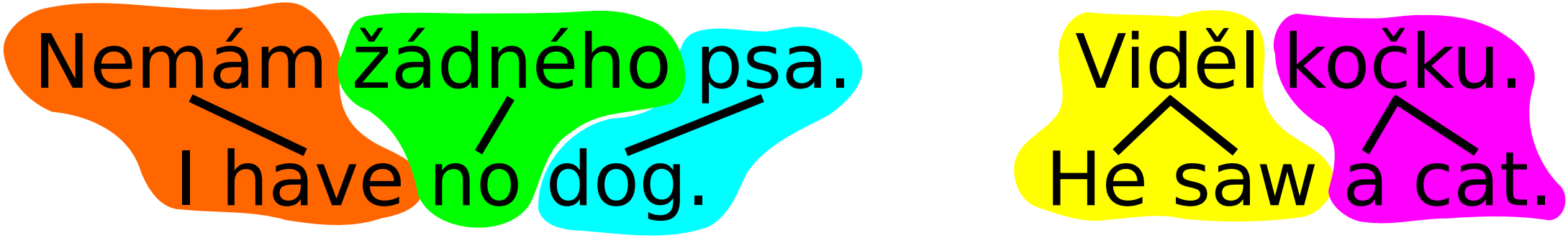
2: Align Words



3: Extract Phrase Pairs (MTUs)

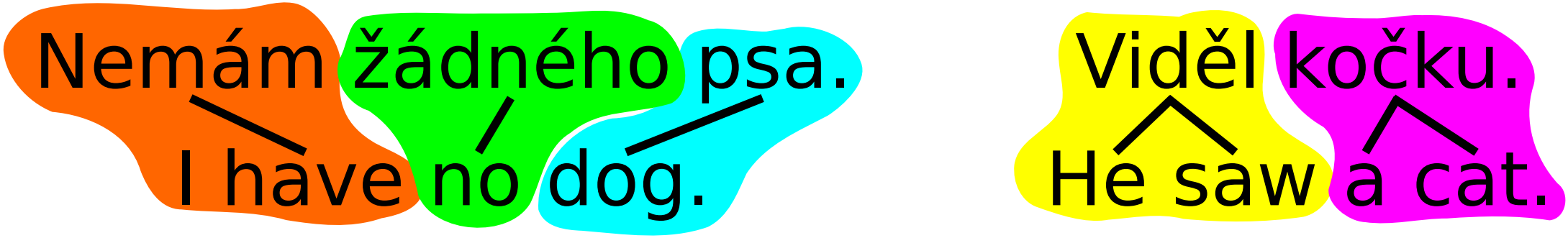


4: New Input



New input: Nemám kočku.

4: New Input



... I don't have cat.

New input: Nemám kočku.

5: Pick Probable Phrase Pairs (TM)

6: So That n-Grams Probable (LM)

Meaning Got Reversed!

What Went Wrong?

$$\hat{e}_1^{\hat{I}} = \underset{I,e_1^I}{\operatorname{argmax}}\, p(f_1^J|e_1^I)p(e_1^I) = \underset{I,e_1^I}{\operatorname{argmax}} \prod_{(\hat{f},\hat{e})\in\text{phrase pairs of } f_1^J,e_1^I} p(\hat{f}|\hat{e})p(e_1^I) \quad (1)$$

- Too strong phrase-independence assumption.
  - Phrases do depend on each other.
    Here "nemám" and "žádného" jointly express one negation.
  - Word alignments ignored that dependence.
    But adding it would increase data sparseness.
- Language model is a separate unit.
  - $p(e_1^I)$ models the target sentence independently of f_1^J.

Redefining $p(e_1^I|f_1^J)$

What if we modelled $p(e_1^I|f_1^J)$ directly, word by word:

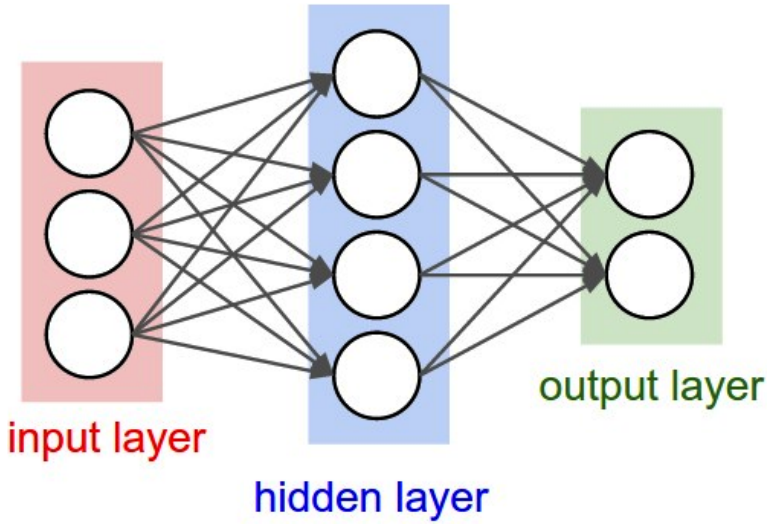
$$\begin{aligned} p(e_1^I|f_1^J) &= p(e_1, e_2, \dots e_I|f_1^J) \\ &= p(e_1|f_1^J) \cdot p(e_2|e_1, f_1^J) \cdot p(e_3|e_2, e_1, f_1^J) \dots \\ &= \prod_{i=1}^{I} p(e_i|e_1, \dots e_{i-1}, f_1^J) \end{aligned} \tag{2}$$

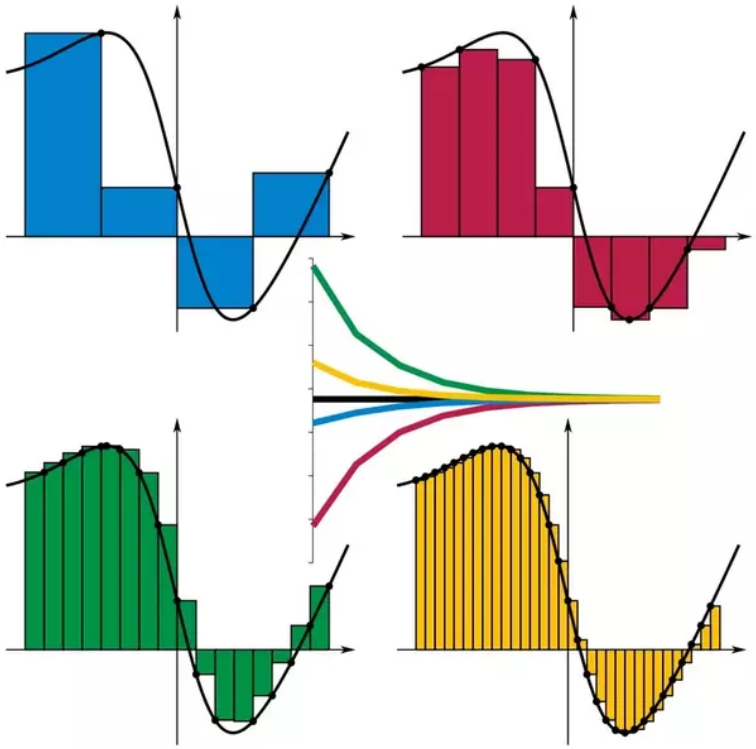
…this is "just a cleverer language model:" $p(e_1^I) = \prod_{i=1}^{I} p(e_i|e_1, \dots e_{i-1})$

Main Benefit: All dependencies available.

But what technical device can learn this?

NNs: Universal Approximators

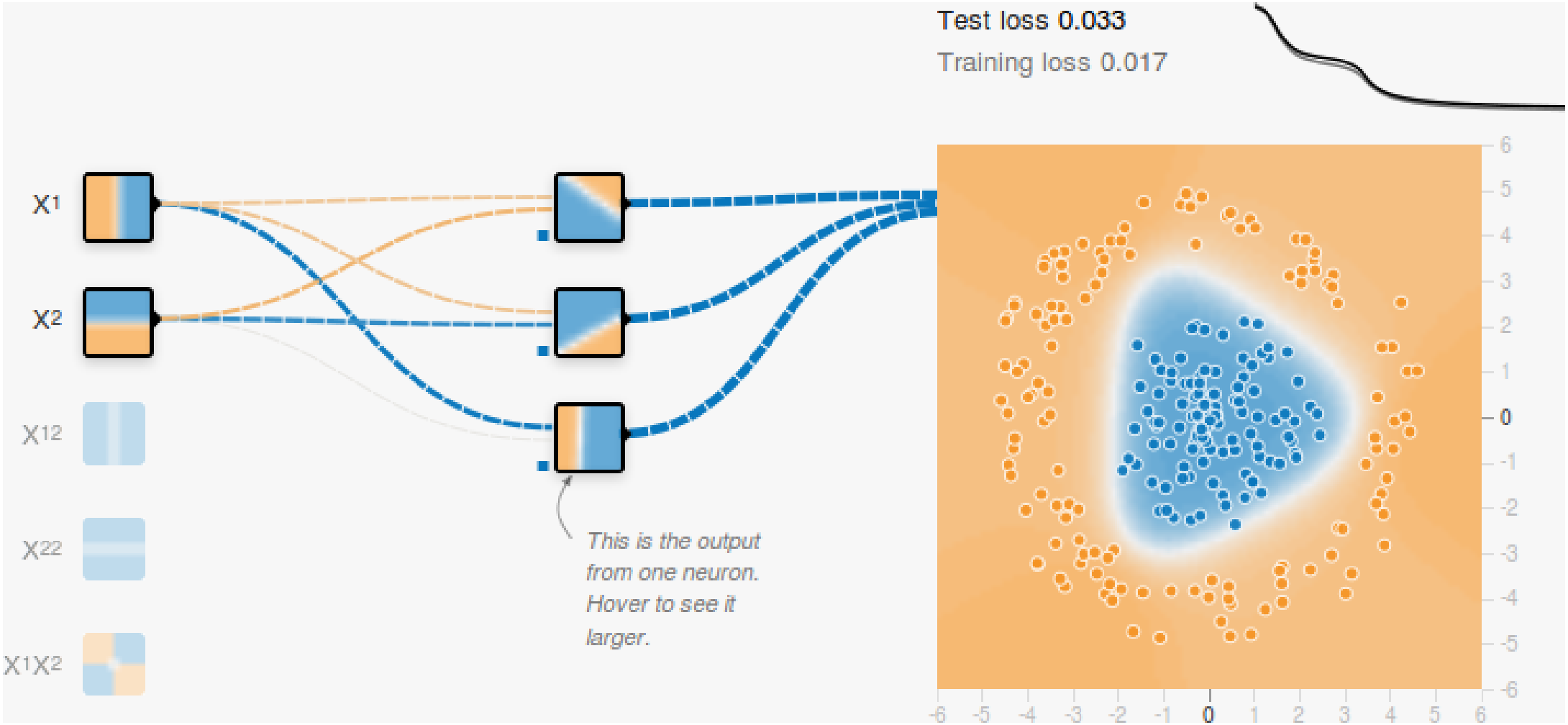




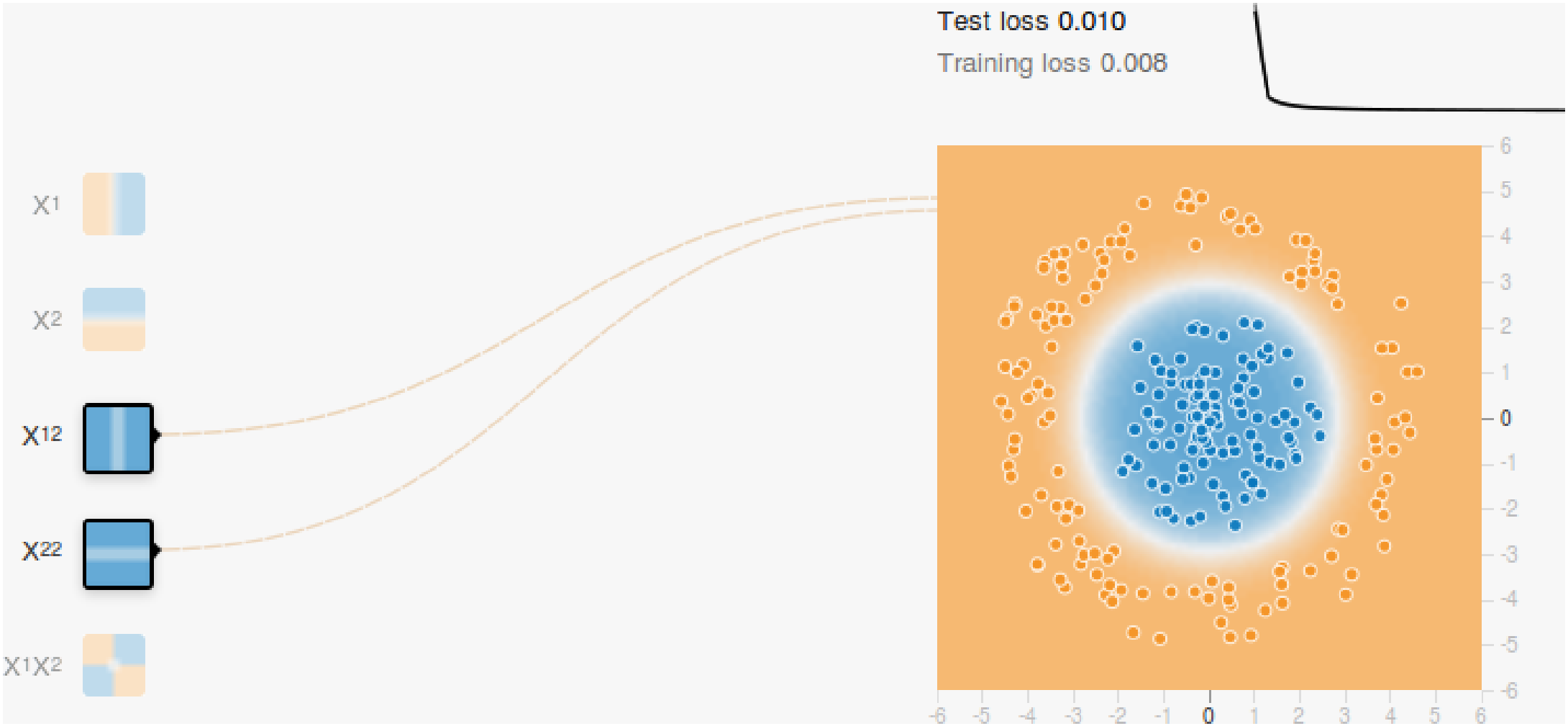
- A neural network with a single hidden layer (possibly huge) can approximate any continuous function to any precision.
- (Nothing claimed about learnability.)

https://www.quora.com/How-can-a-deep-neural-network-with-ReLU-activations-in-its-hidden-layers-approximate-any-function

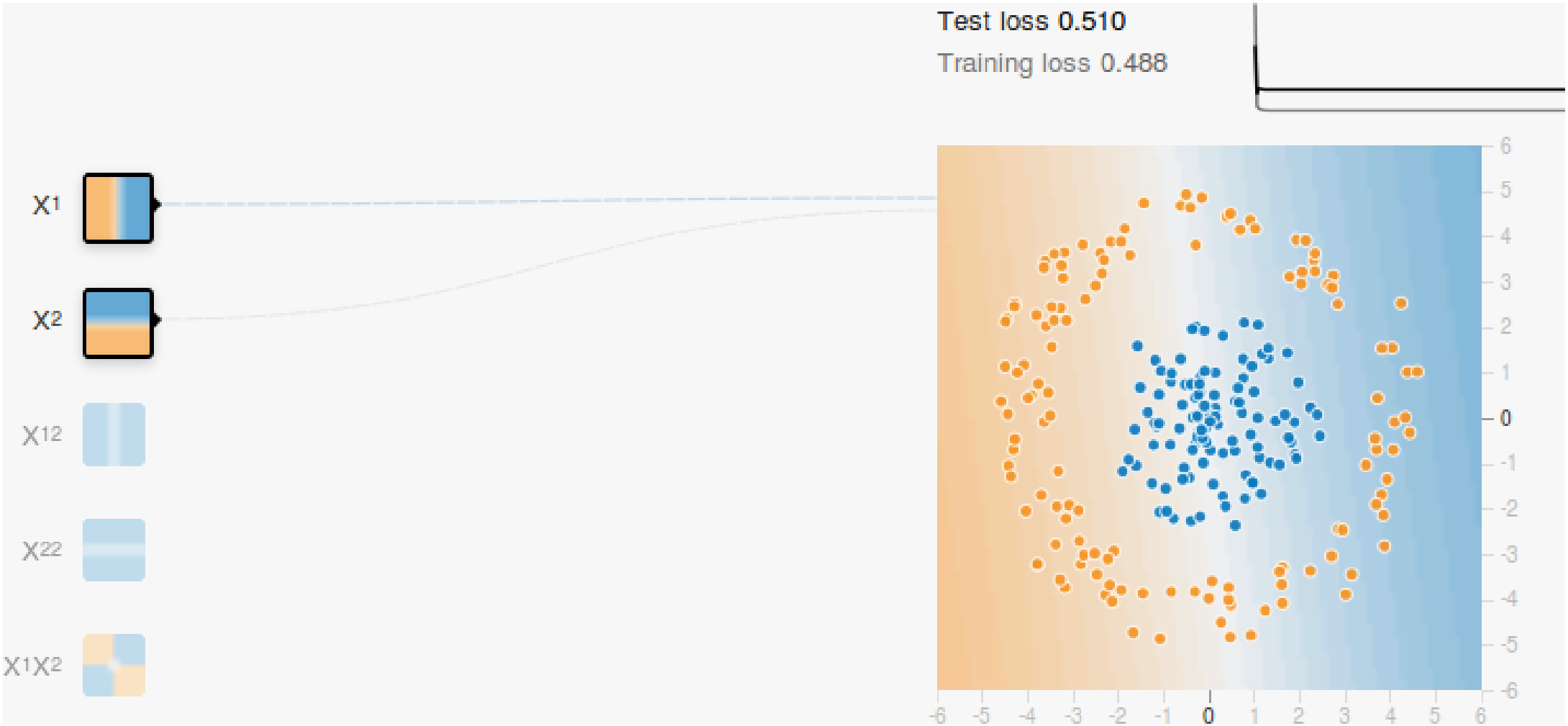
playground.tensorflow.org



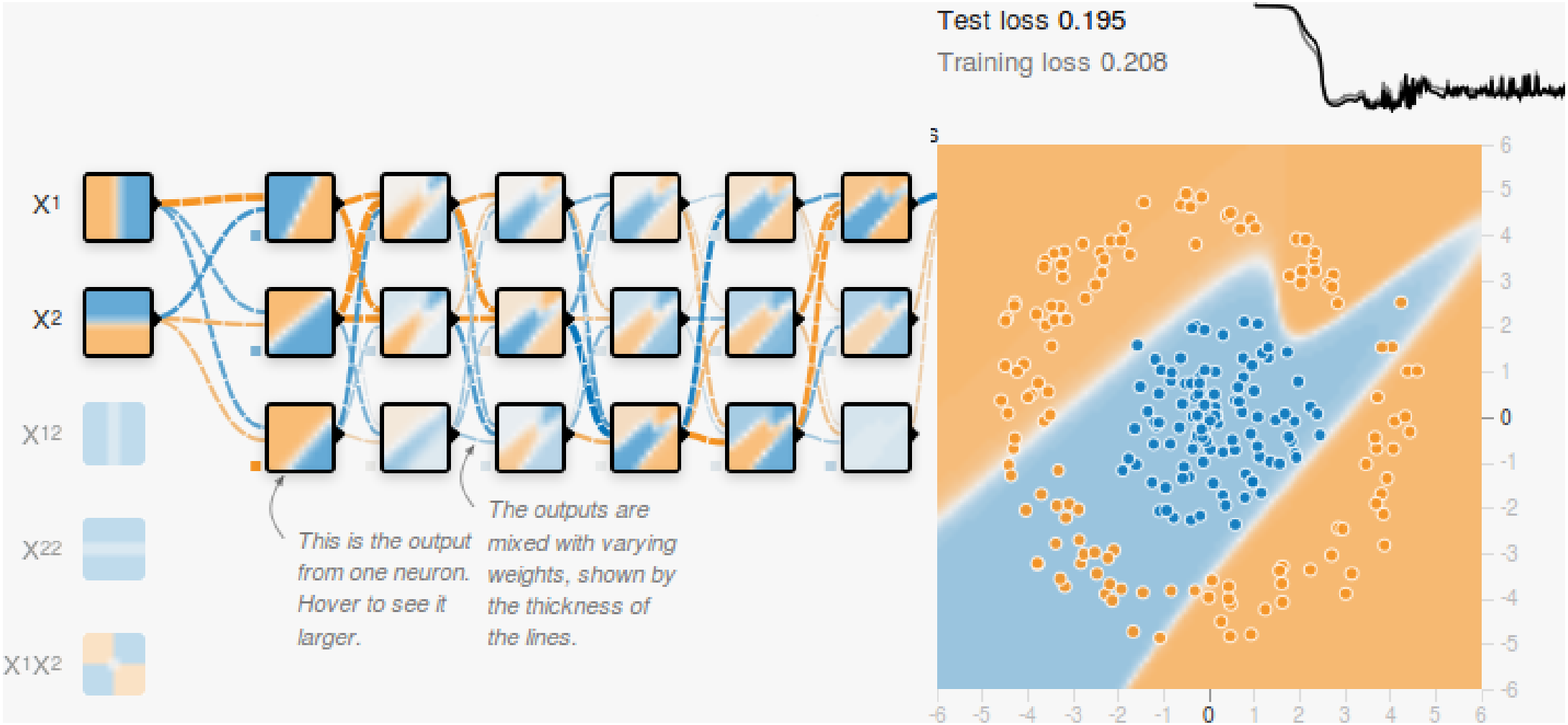
Perfect Features



Bad Features & Low Depth

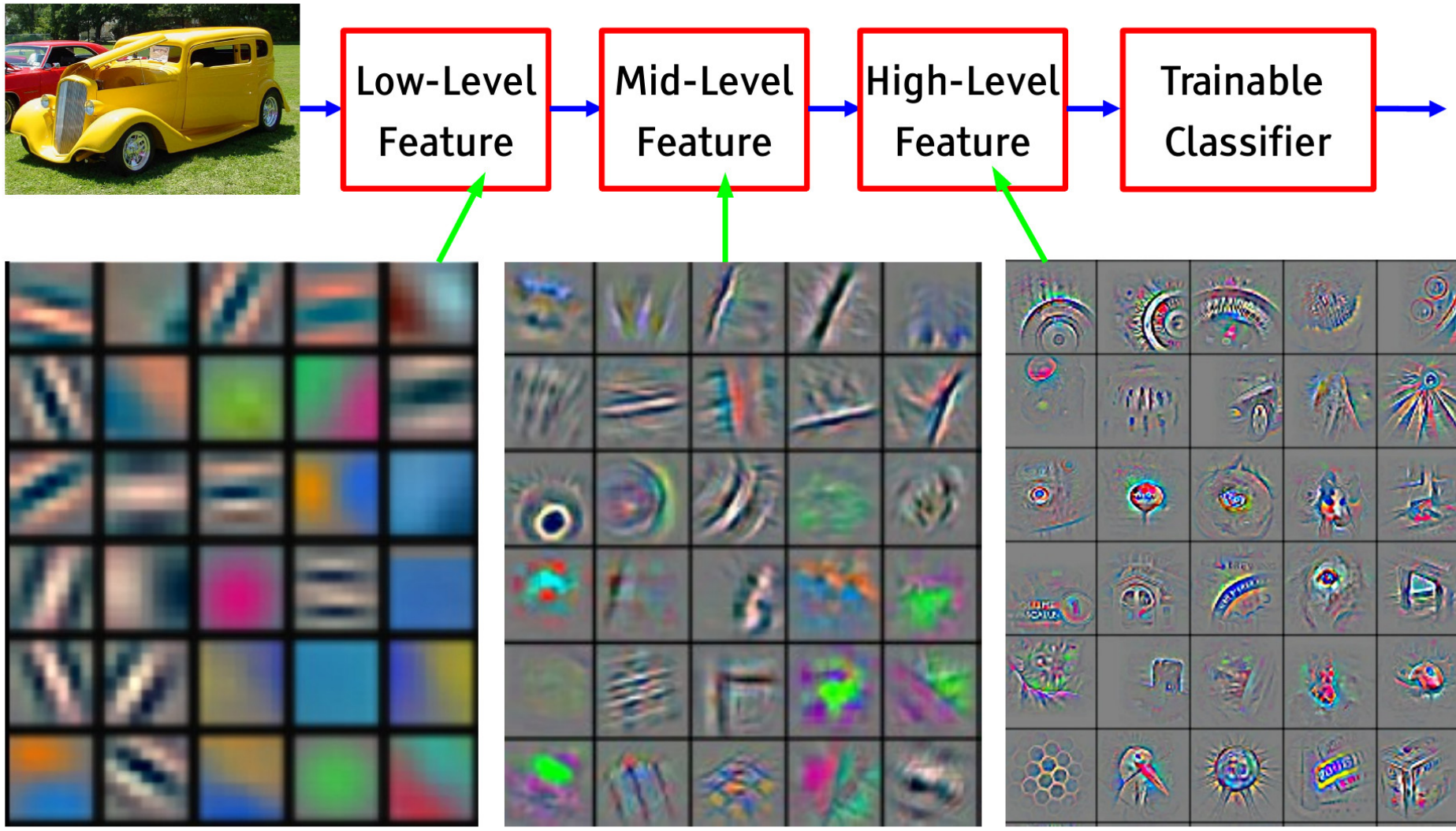


Too Complex NN Fails to Learn



Deep NNs for Image Classification

- It's deep if it has more than one stage of non-linear feature transformation



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Representation Learning

- Based on training data
  (sample inputs and expected outputs)
- the neural network learns by itself
- what is important in the inputs
- to predict the outputs best.

A "representation" is a new set of axes.

- Instead of 3 dimensions (x, y, color), we get
- 2000 dimensions: (elephantity, number of storks, blueness, ...)
- designed automatically to help in best prediction of the output

One Layer $\tanh(Wx + b)$, 2D→2D

Skew:

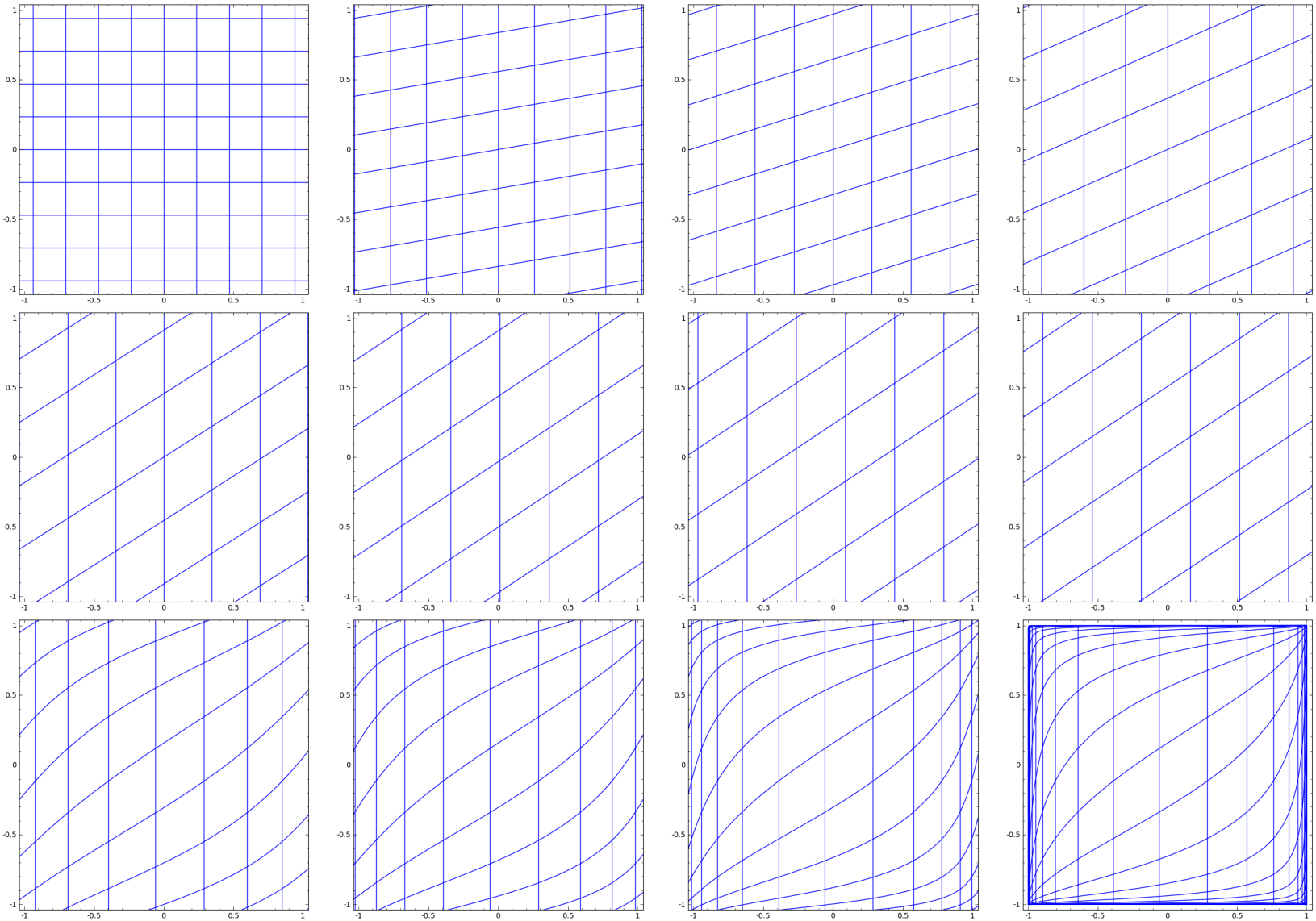
W

Transpose:

b

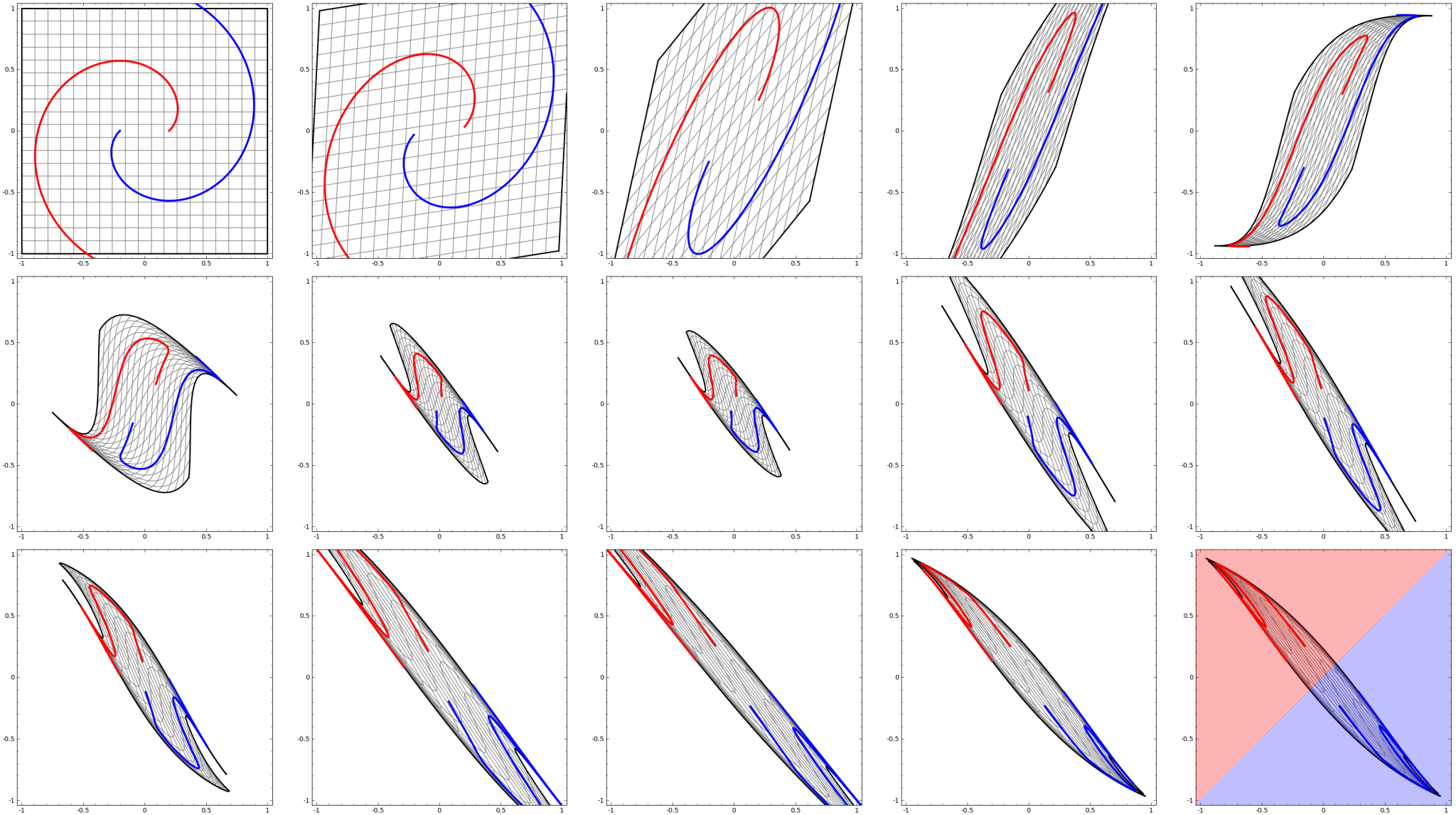
Non-lin.:

tanh



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

Four Layers, Disentagling Spirals



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

Processing Text with NNs

- Map each word to a vector of 0s and 1s ("1-hot repr."):

$$\text{cat} \mapsto (0, 0, \ldots, 0, 1, 0, \ldots, 0)$$

- Sentence is then a matrix:

		the	cat	is	on	the	mat
↑	a	0	0	0	0	0	0
	about	0	0	0	0	0	0
	...	...	...	...	...	...	...
	cat	0	1	0	0	0	0
	...	...	...	...	...	...	...
	is	0	0	1	0	0	0
	...	...	...	...	...	...	...
	on	0	0	0	1	0	0
	...	...	...	...	...	...	...
	the	1	0	0	0	1	0
	...	...	...	...	...	...	...
↓	zebra	0	0	0	0	0	0

Processing Text with NNs

- Map each word to a vector of 0s and 1s ("1-hot repr."):

$$\text{cat} \mapsto (0, 0, \ldots, 0, 1, 0, \ldots, 0)$$

- Sentence is then a matrix:

		the	cat	is	on	the	mat
↑	a	0	0	0	0	0	0
	about	0	0	0	0	0	0
	...	...	...	...	...	...	...
	cat	0	1	0	0	0	0
Vocabulary size:	...	...	...	...	...	...	...
1.3M English	is	0	0	1	0	0	0
2.2M Czech	...	...	...	...	...	...	...
	on	0	0	0	1	0	0
	...	...	...	...	...	...	...
	the	1	0	0	0	1	0
	...	...	...	...	...	...	...
↓	zebra	0	0	0	0	0	0

Processing Text with NNs

- Map each word to a vector of 0s and 1s ("1-hot repr."):

$$\text{cat} \mapsto (0, 0, \ldots, 0, 1, 0, \ldots, 0)$$

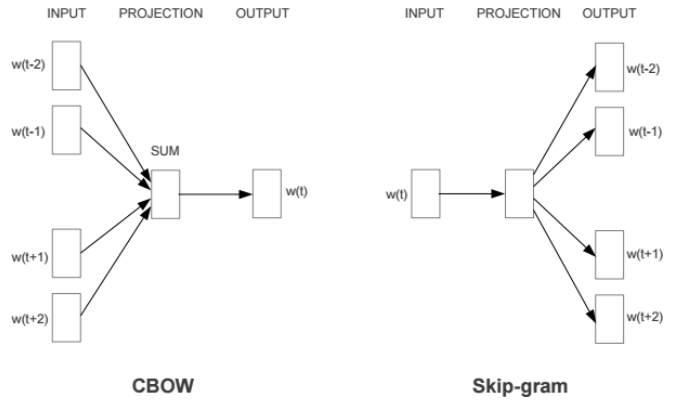
- Sentence is then a matrix:

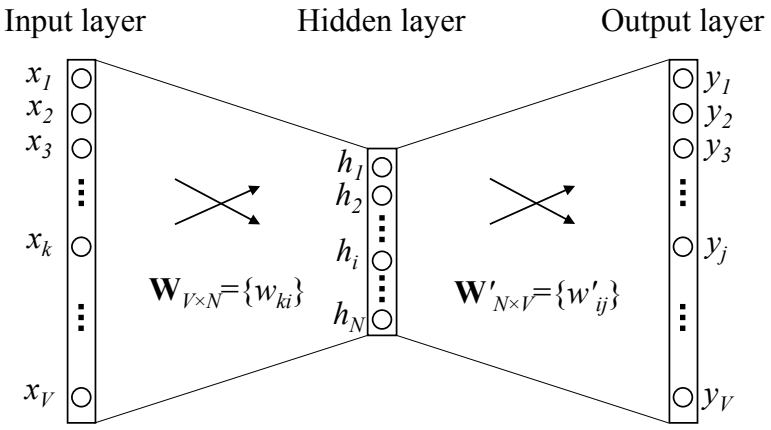
		the	cat	is	on	the	mat
↑	a	0	0	0	0	0	0
	about	0	0	0	0	0	0
	...	...	...	...	...	...	...
	cat	0	1	0	0	0	0
Vocabulary size:	...	...	...	...	...	...	...
1.3M English	is	0	0	1	0	0	0
2.2M Czech	...	...	...	...	...	...	...
	on	0	0	0	1	0	0
	...	...	...	...	...	...	...
	the	1	0	0	0	1	0
	...	...	...	...	...	...	...
↓	zebra	0	0	0	0	0	0

Main drawback: No relations, all words equally close/far.

Solution: Word Embeddings

- Map each word to a dense vector.
- In practice 300–2000 dimensions are used, not 1–2M.
  - The dimensions have no clear interpretation.
- Embeddings are trained for each particular task.
  - NNs: The matrix that maps 1-hot input to the first layer.
- The famous word2vec (Mikolov et al., 2013):
  - CBOW: Predict the word from its four neighbours.
  - Skip-gram: Predict likely neighbours given the word.



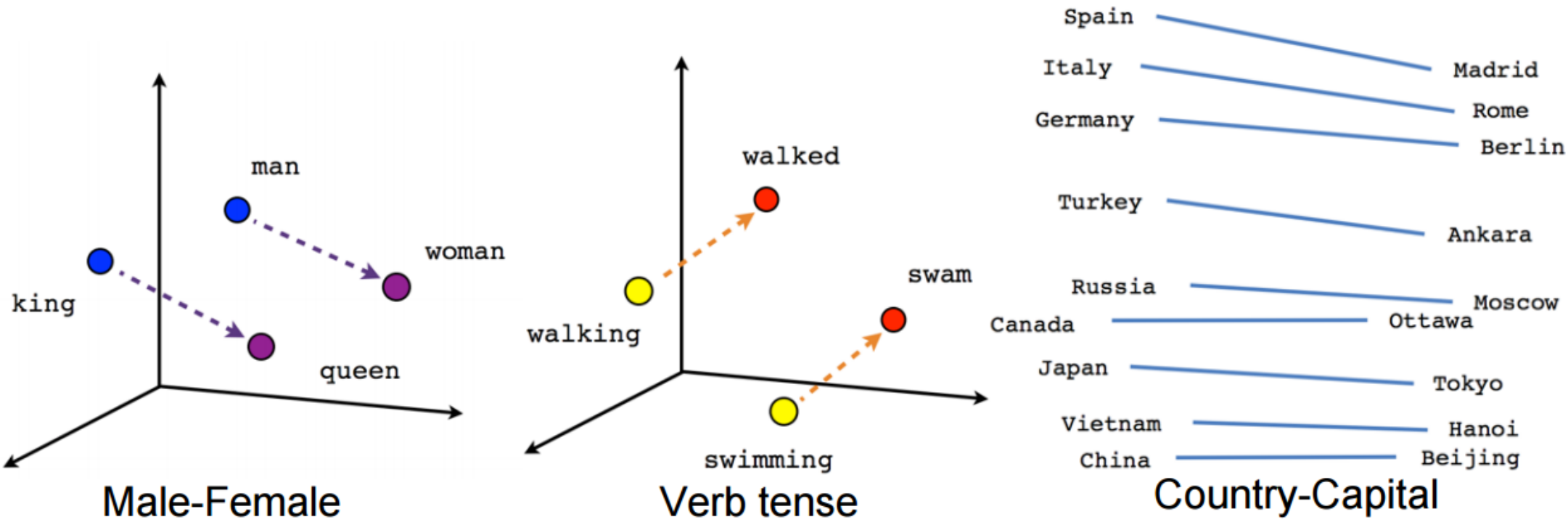


Right: CBOW with just a single-word context (http://www-personal.umich.edu/~ronxin/pdf/w2vexp.pdf)

Continuous Space of Words

Word2vec embeddings show interesting properties:

$$v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen}) \tag{3}$$



Illustrations from https://www.tensorflow.org/tutorials/word2vec

Further Compression: Sub-Words

- SMT struggled with productive morphology (>1M wordforms).

  nejneobhodpodařovávatelnějšími, Donaudampfschifffahrtsgesellschaftskapitän

- NMT can handle only 30–80k dictionaries.

⇒ Resort to sub-word units.

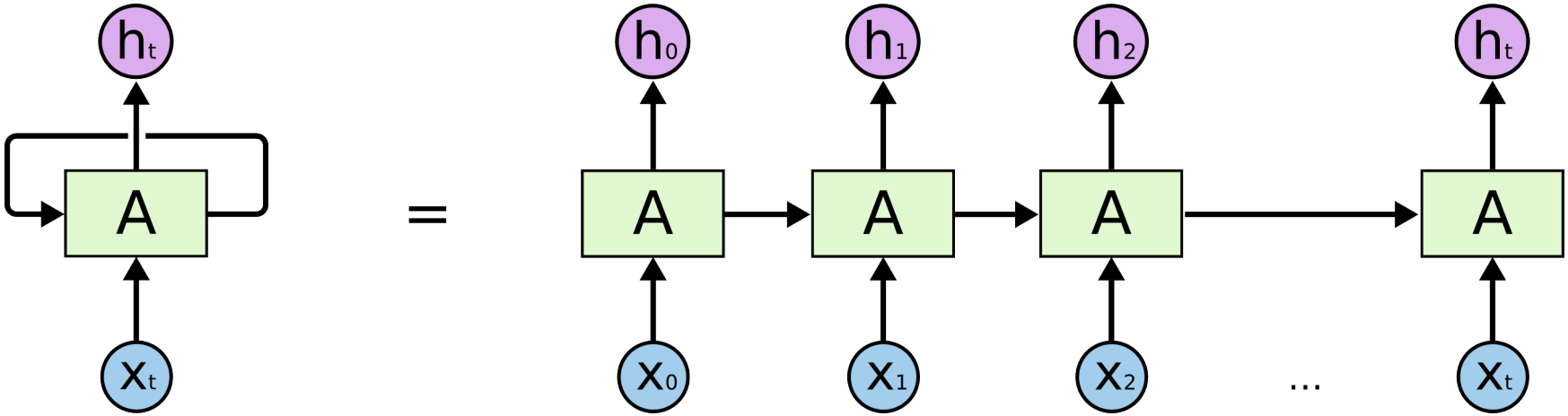
Orig	český politik svezl migranty
Syllables	čes ký ␣ po li tik ␣ sve zl ␣ mig ran ty
Morphemes	česk ý ␣ politik ␣ s vez l ␣ migrant y
Char Pairs	če sk ý ␣ po li ti k ␣ sv ez l ␣ mi gr an ty
Chars	č e s k ý ␣ p o l i t i k ␣ s v e z l ␣ m i g r a n t y
BPE 30k	český politik s@@ vez@@ l mi@@ granty

BPE (Byte-Pair Encoding) uses n most common substrings (incl. frequent words).

Variable-Length Inputs

Variable-length input can be handled by recurrent NNs:

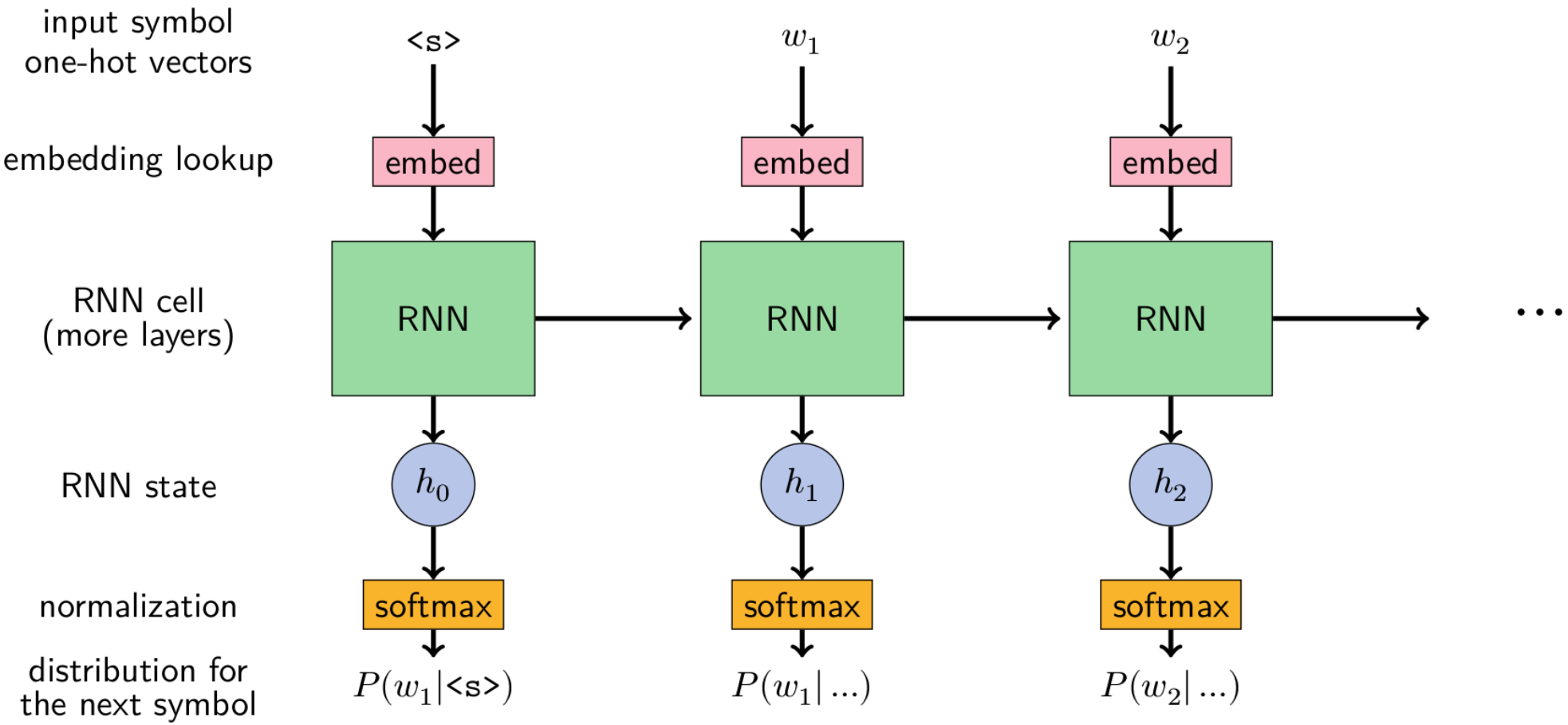
- Reading one input symbol at a time.
  - The same (trained) transformation A used every time.
- Unroll in time (up to a fixed length limit).



Vanilla RNN:

$$h_t = \tanh\left(W[h_{t-1}; x_t] + b\right) \tag{4}$$

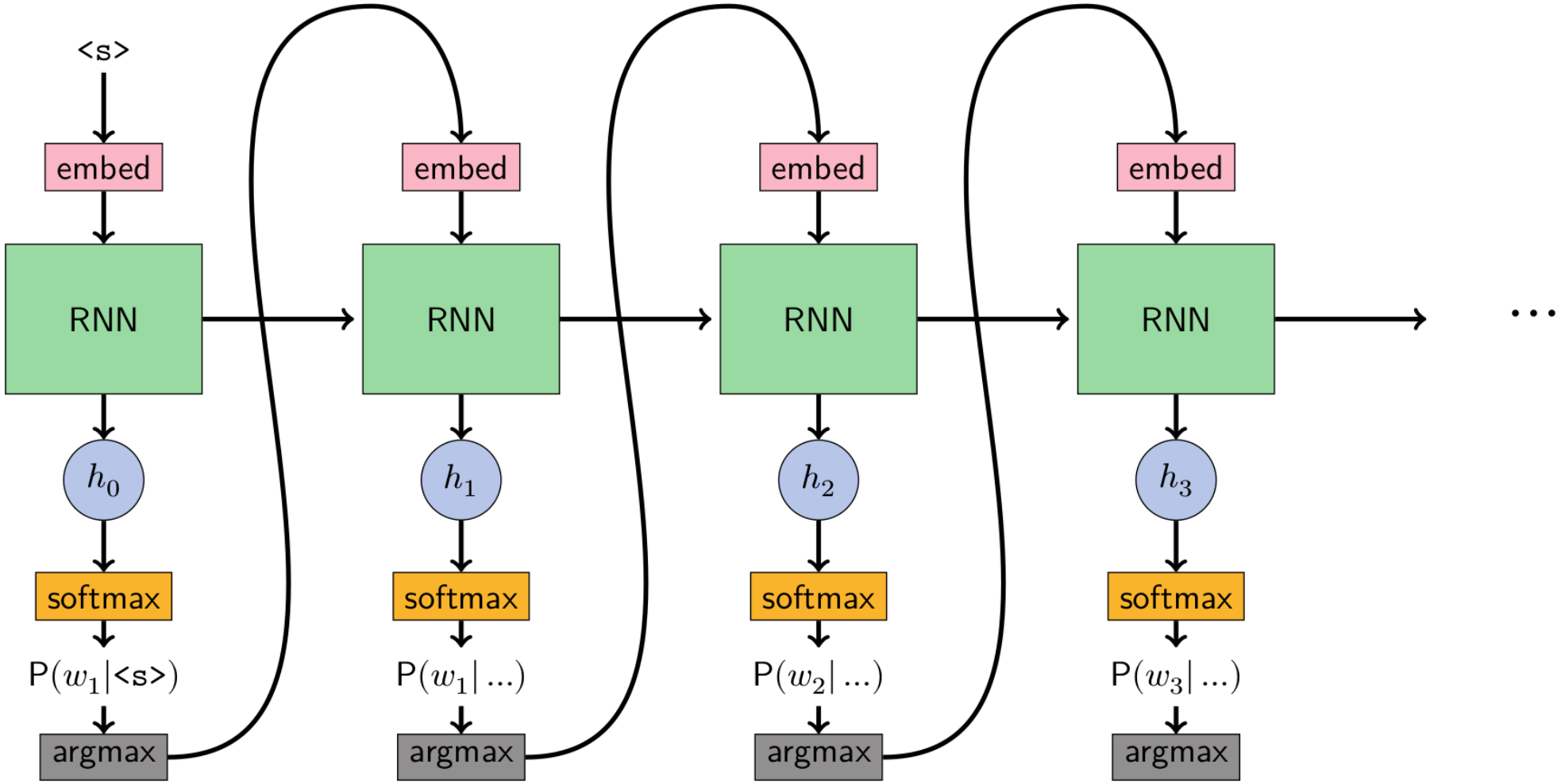
Neural Language Model



- estimate probability of a sentence using the chain rule
- output distributions can be used for sampling

Thanks to Jindřich Libovický for the slides.

Sampling from a LM



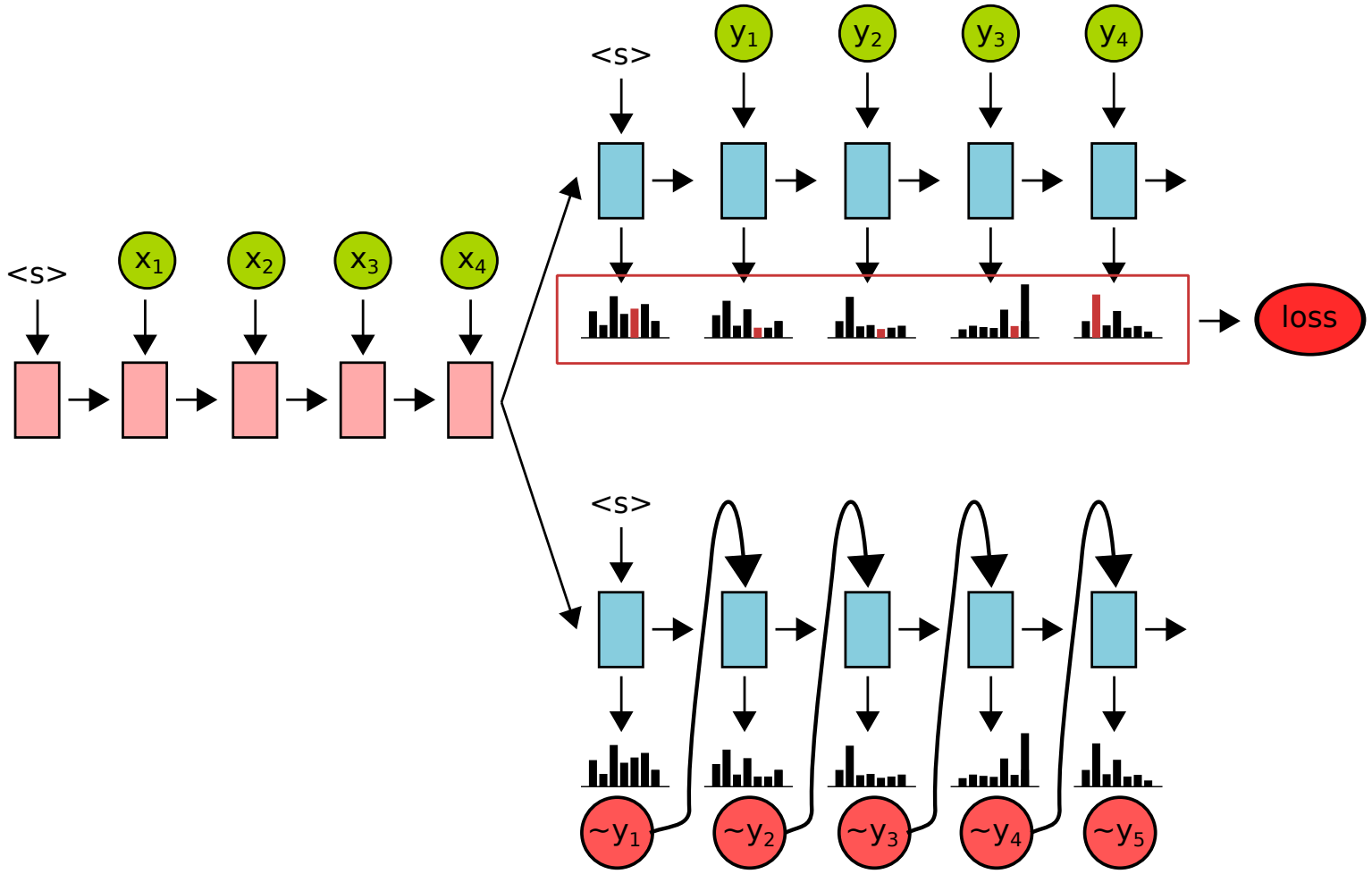
- "Autoregressive decoder" = conditioned on its preceding output.

Autoregressive Decoding

```
last_w = "<s>"
while last_w != "</s>":
    last_w_embedding = target_embeddings[last_w]
    state, dec_output = dec_cell(state,
                                 last_w_embedding)
    logits = output_projection(dec_output)
    last_w = np.argmax(logits)
    yield last_w
```

RNN Training vs. Runtime

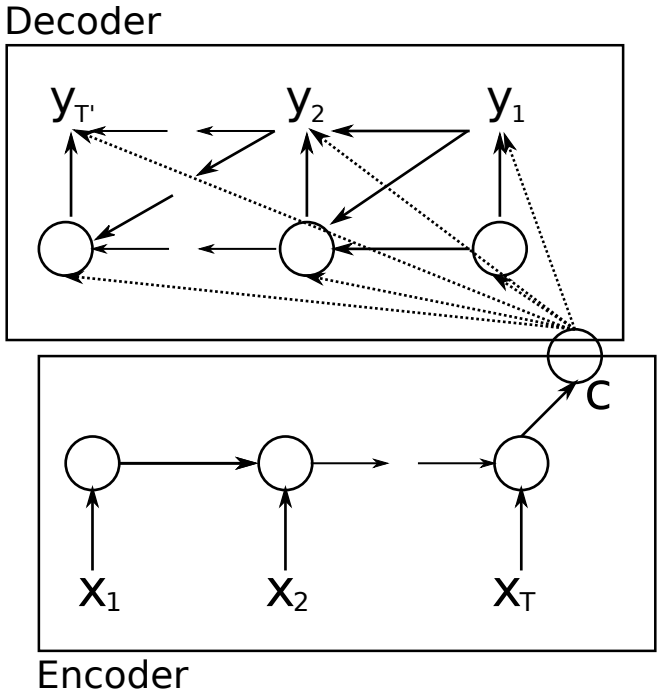
runtime: $\hat{y}_j$ (decoded) $\times$ training: y_j (ground truth)



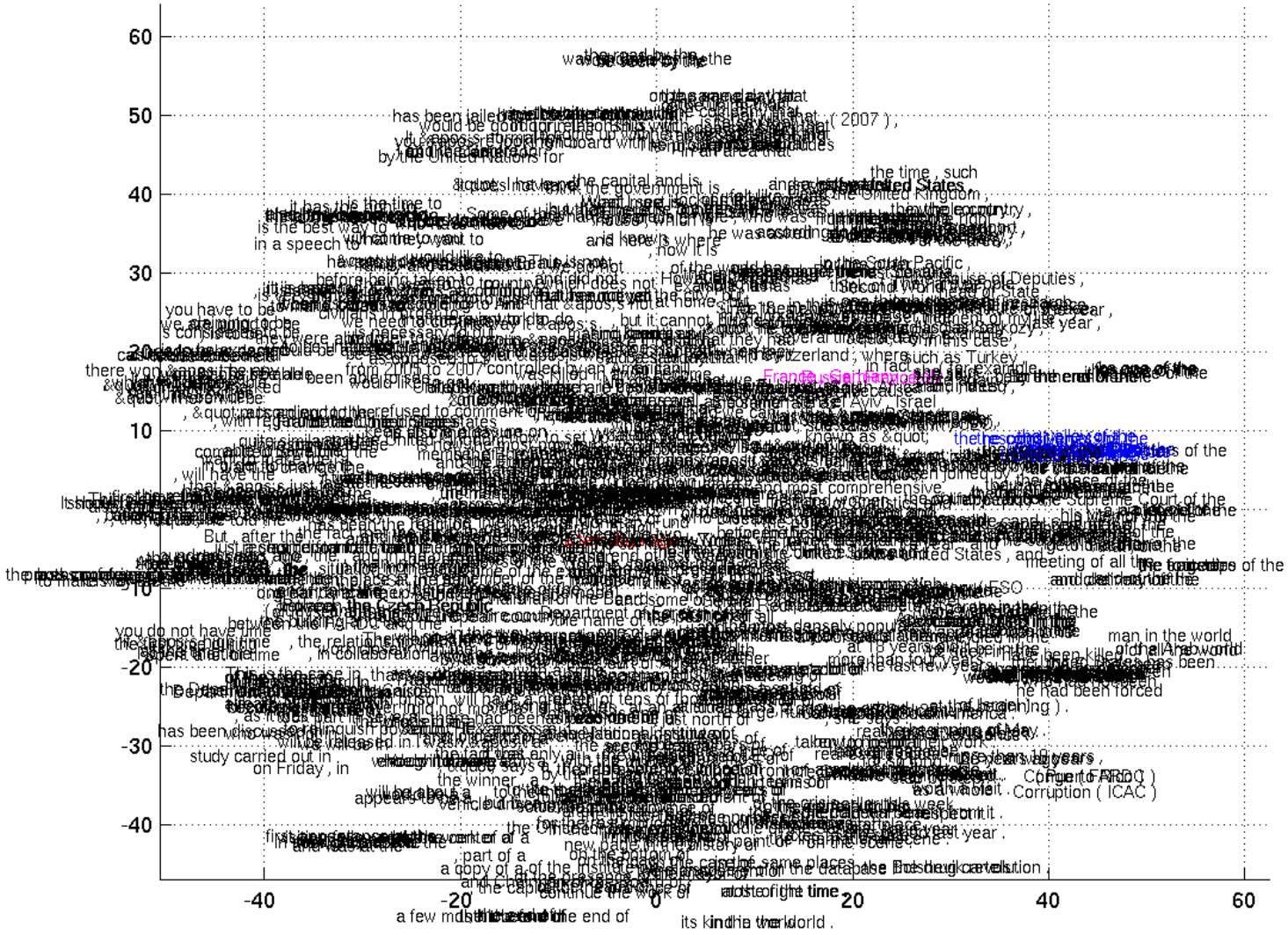
NNs as Translation Model in SMT

Cho et al. (2014) proposed:

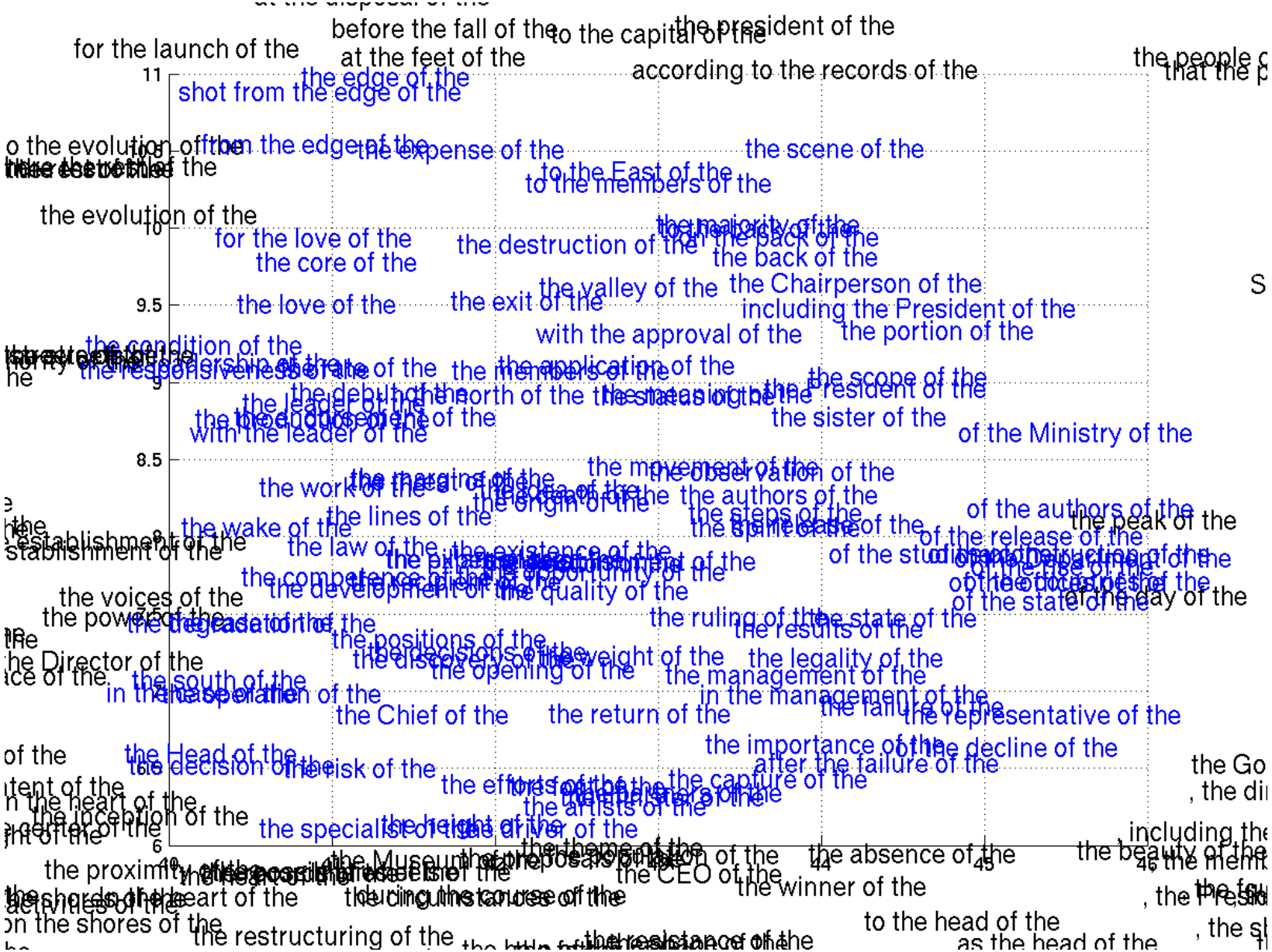
- encoder-decoder architecture and
- GRU unit (name given later by Chung et al. (2014))
- to score variable-length phrase pairs in PBMT.



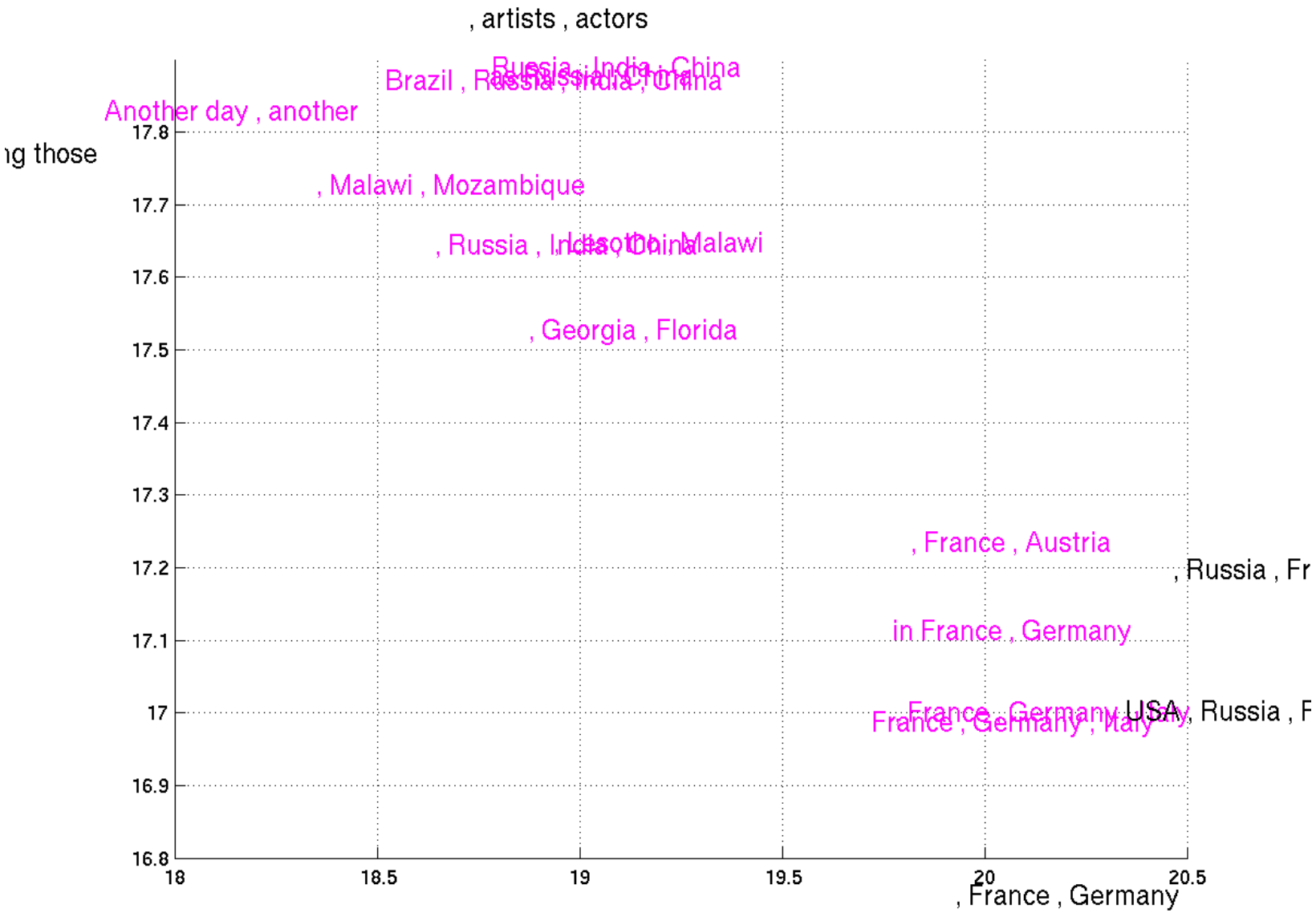
⇒ Embeddings of Phrases



⇒ Syntactic Similarity ("of the")



⇒ Semantic Similarity (Countries)

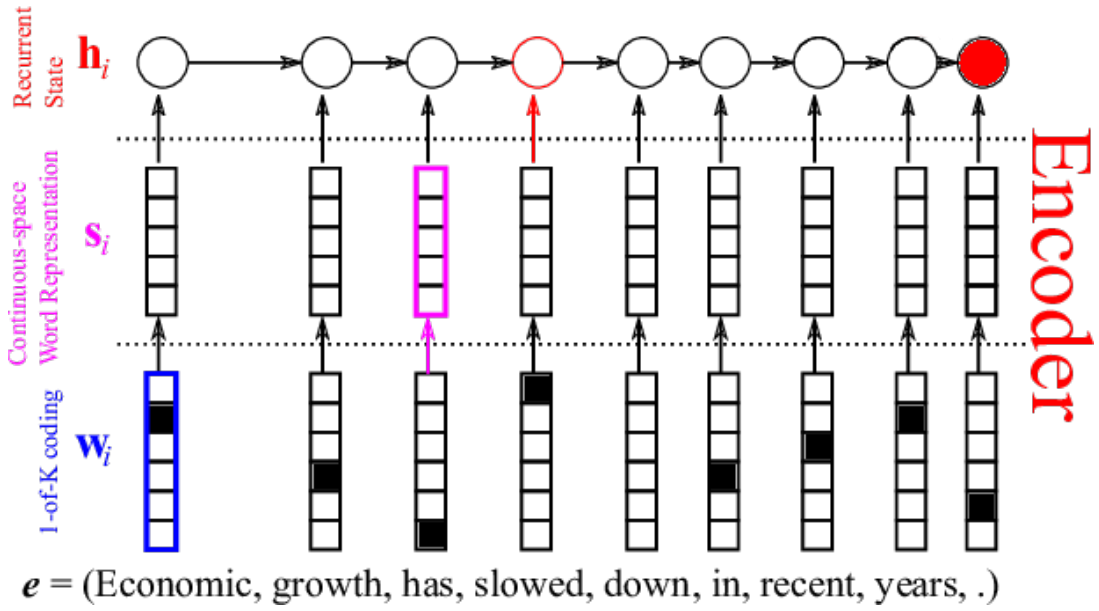


NMT: Sequence to Sequence

Sutskever et al. (2014) use:

- LSTM RNN encoder-decoder
- to consume
  and produce variable-length sentences.

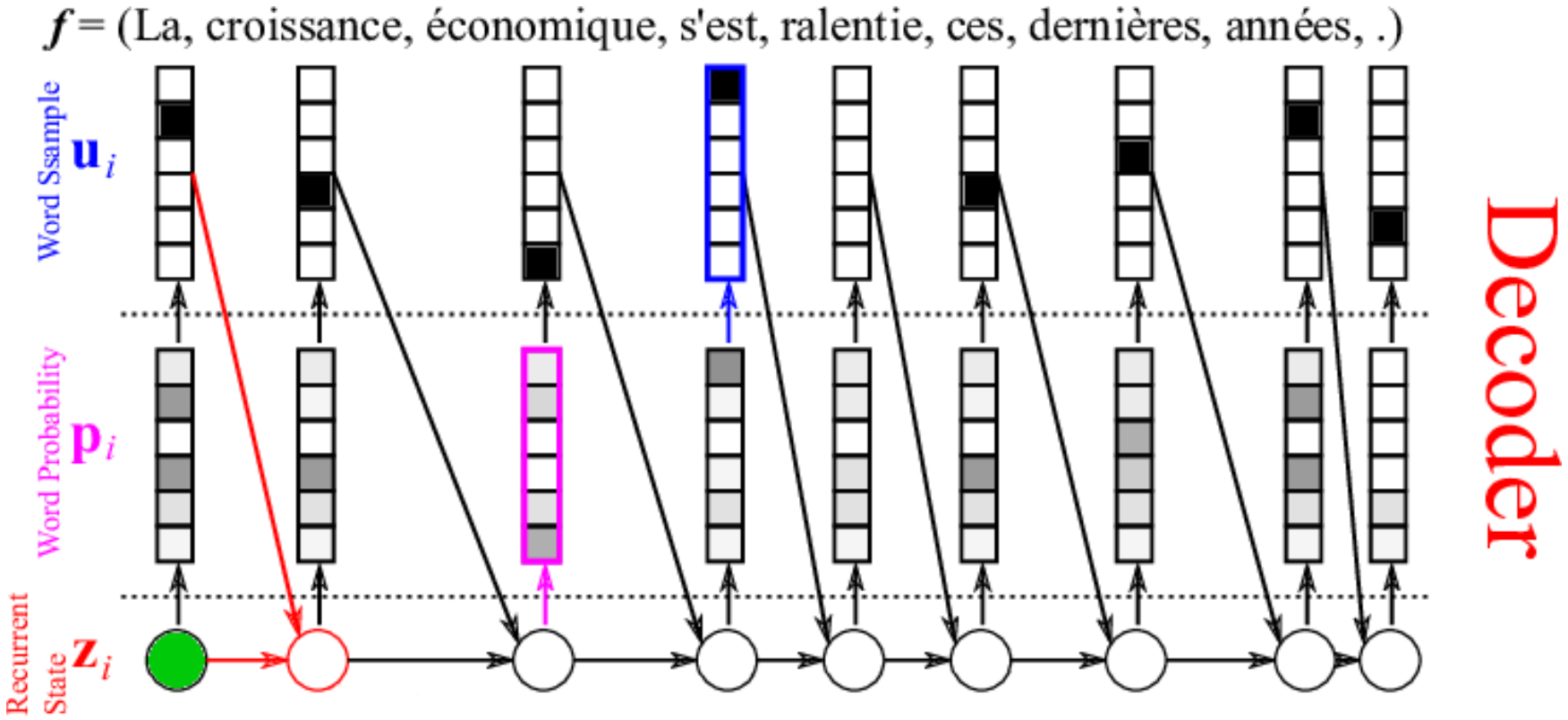
First the Encoder:



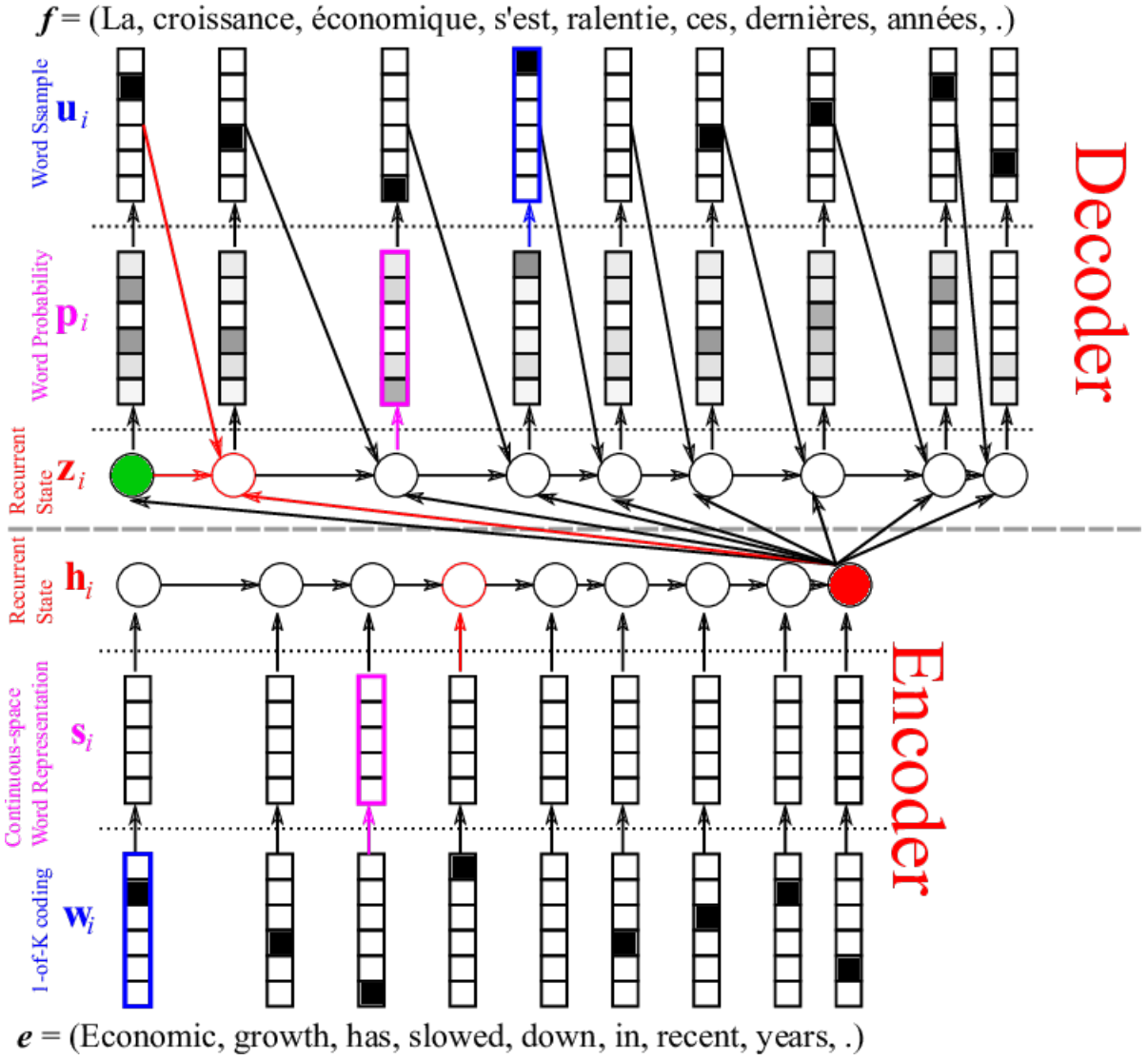
Then the Decoder

Remember: $p(e_1^I|f_1^J) = p(e_1|f_1^J) \cdot p(e_2|e_1, f_1^J) \cdot p(e_3|e_2, e_1, f_1^J) \ldots$

- Again RNN, producing one word at a time.
- The produced word fed back into the network.
  - (Word embeddings in the target language used here.)

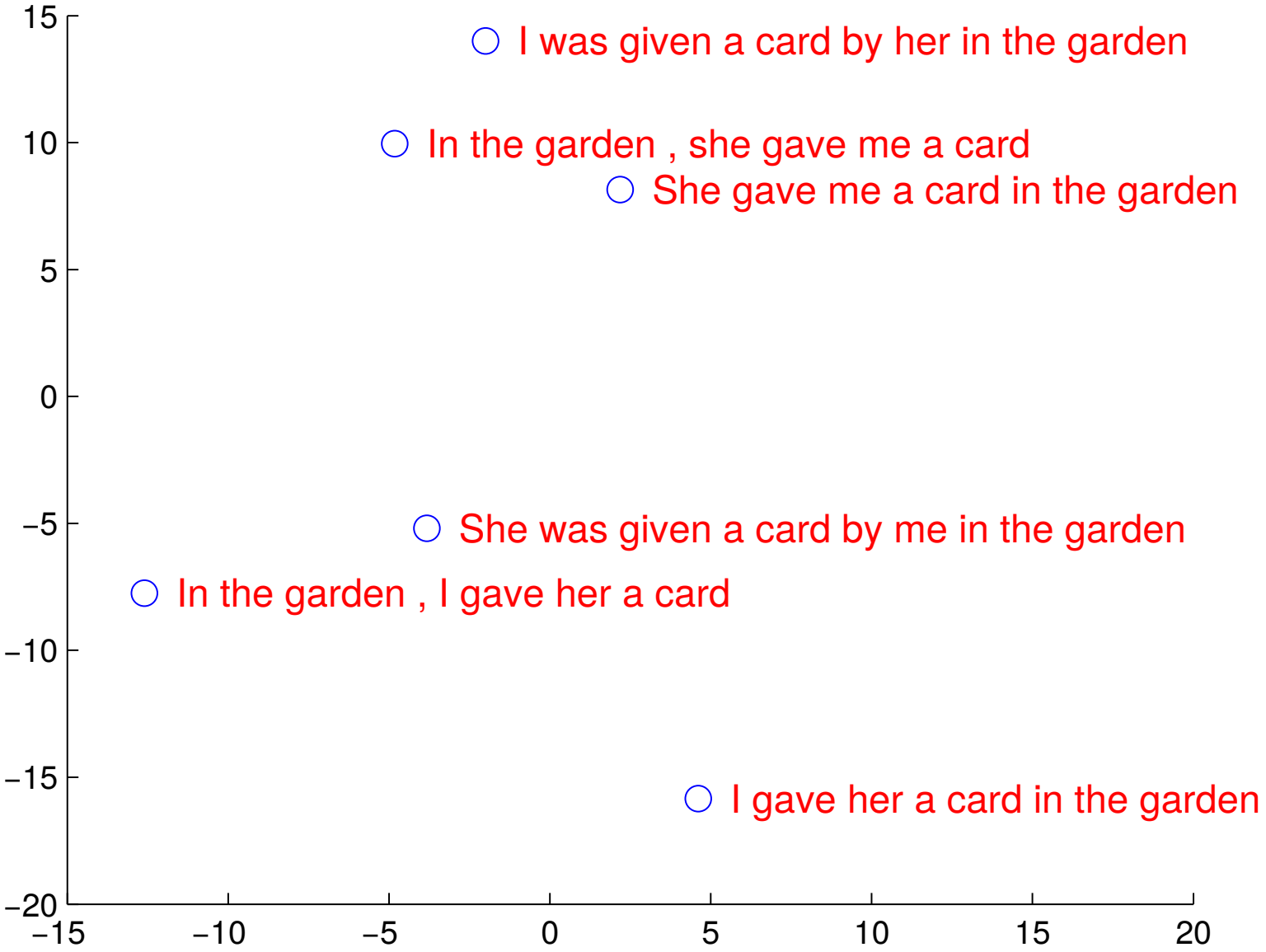


Encoder-Decoder Architecture



https://devblogs.nvidia.com/parallelforall/introduction-neural-machine-translation-gpus-part-2/

Continuous Space of Sentences



2-D PCA projection of 8000-D space representing sentences (Sutskever et al., 2014).

Architectures in the Decoder

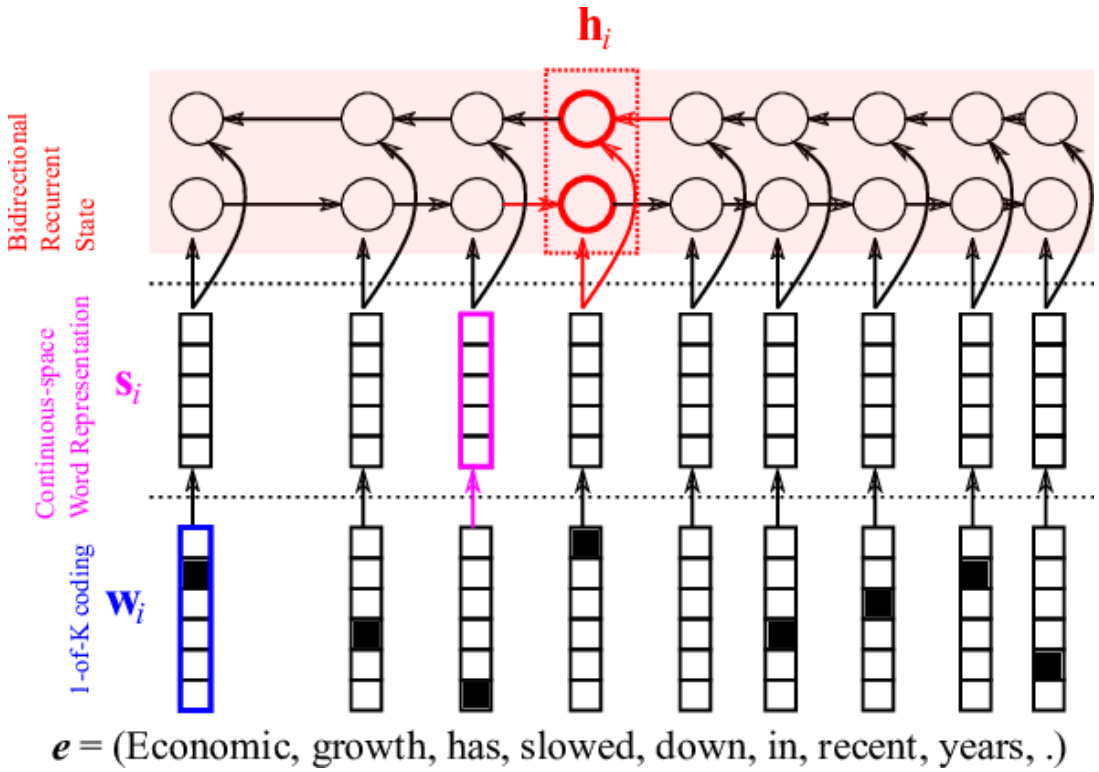
- RNN – original sequence-to-sequence learning (2015)
  - principle known since 2014 (University of Montreal)
  - made usable in 2016 (University of Edinburgh)
- CNN – convolution sequence-to-sequence by Facebook (2017)
- Self-attention (so-called Transformer) by Google (2017)

Attention (1/3)

- Arbitrary-length sentences fit badly into a fixed vector.
- Reading input backward works better.

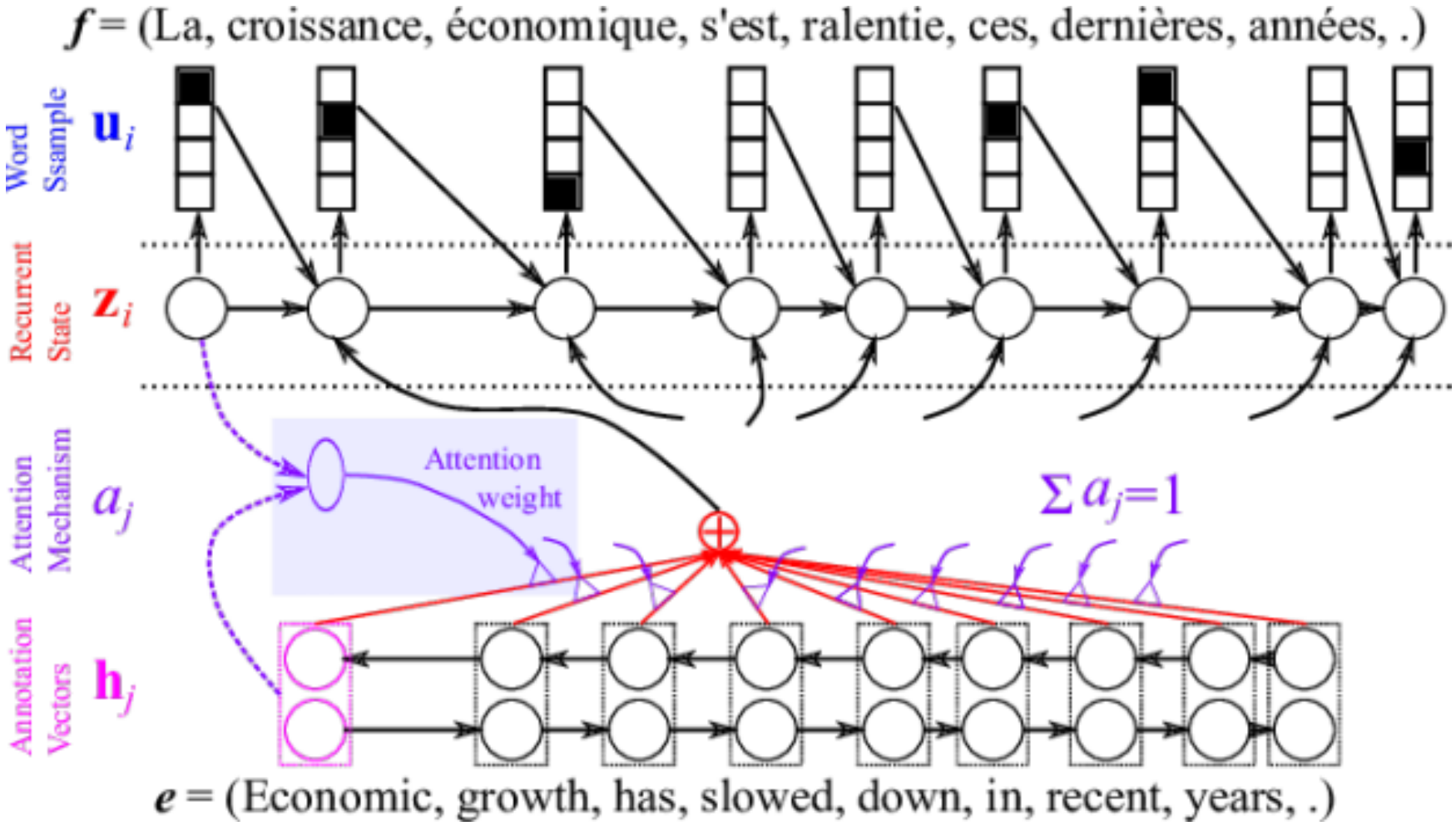
  … because early words will be more salient.

⇒ Use Bi-directional RNN and "attend" to all states h_i.

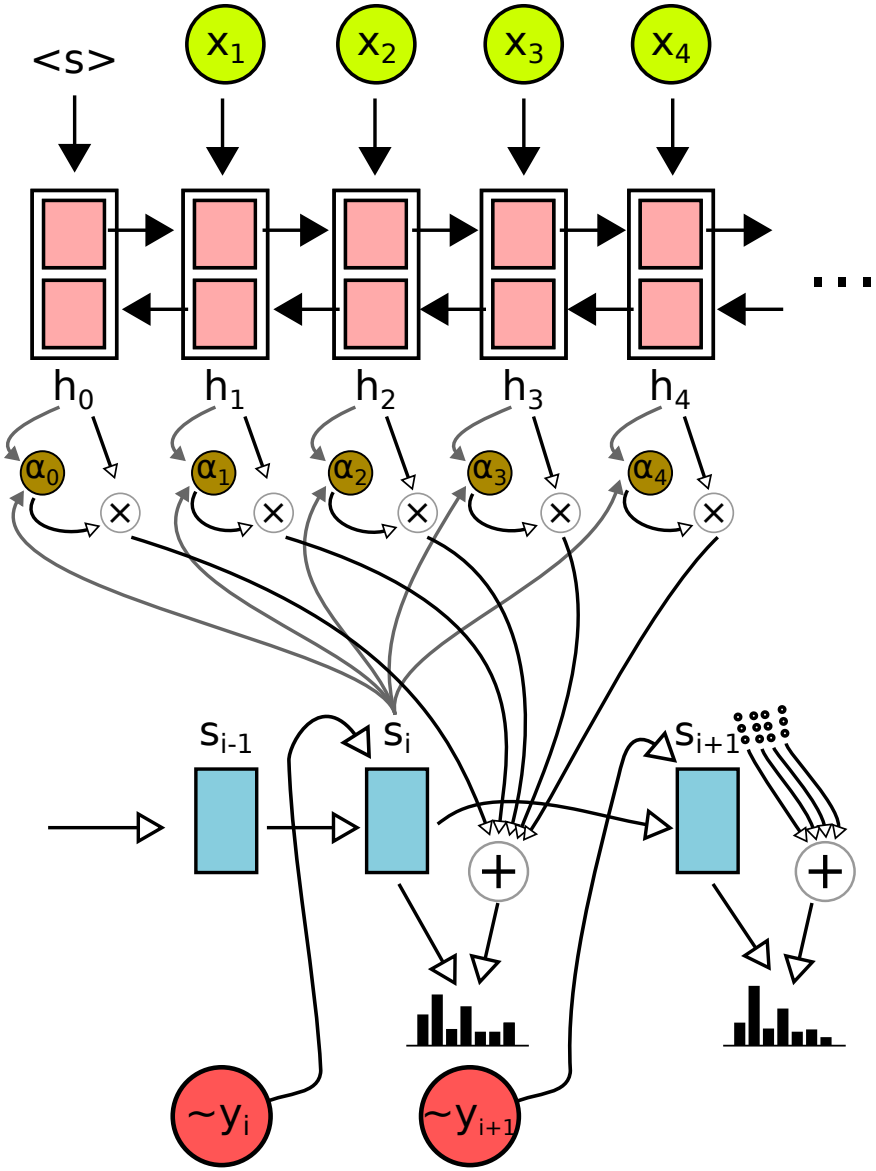


Attention (2/3)

- Add a sub-network predicing importance of source states at each step.



Attention (3/3)



Attention Model in Equations (1)

Inputs:

decoder state: s_i, encoder states: $h_j = \left[\overrightarrow{h_j}; \overleftarrow{h_j}\right] \quad \forall i = 1 \ldots T_x$

Attention energies:

$e_{ij} = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j + b_a\right)$

Attention distribution: $\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp(e_{ik})}$

Context vector: $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$

Attention Model in Equations (2)

Decoder state:

$$s_i = \tanh\left(U_d s_{i-1} + V_d E y_{i-1} + C_d c_i + b_d\right)$$

...attention is mixed with the hidden state

Output projection:

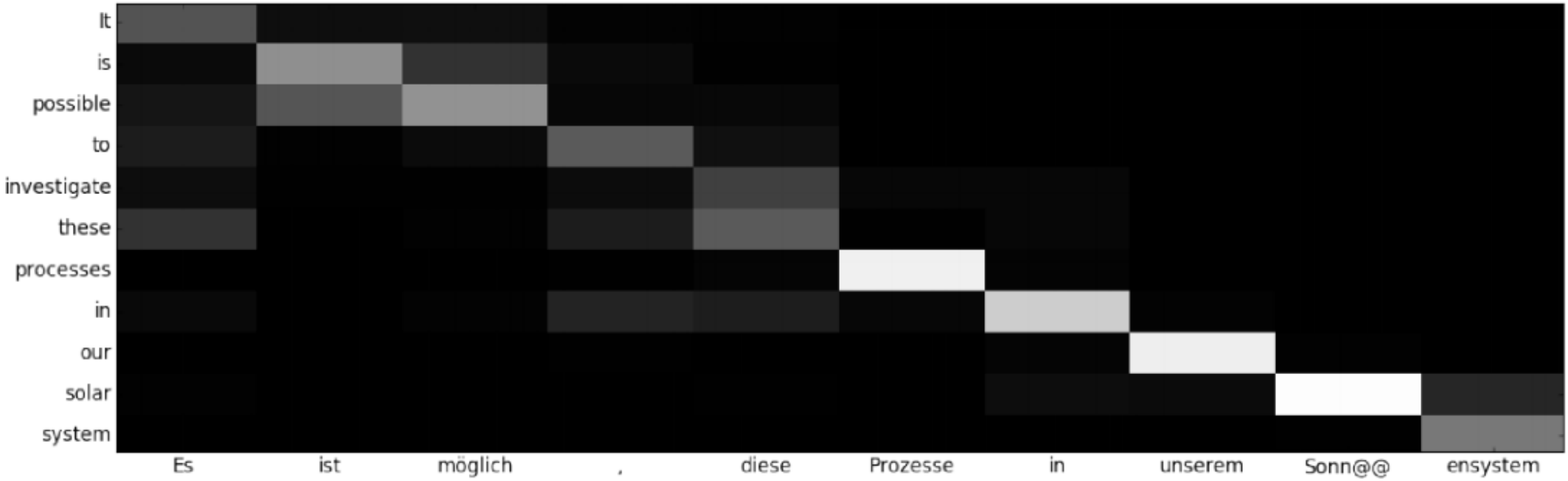
$$t_i = \tanh\left(U_o s_i + V_o E y_{i-1} + C_o c_i + b_o\right)$$

Output distribution:

$$p\left(y_i = k \mid s_i, y_{i-1}, c_i\right) \propto \exp\left(W_o t_i\right)_k + b_k$$

Attention ≈ Alignment

- We can collect the attention across time.
- Each column corresponds to one decoder time step.
- Source tokens correspond to rows.



Transformer Model

See slides from NPFL087:

https://ufal.mff.cuni.cz/courses/npfl087
#08 Transformer and Syntax in NMT

- Transformer model.
- Self-Attention.
- Options for linguistics in NMT:
  - Constrain network structure. …usually too limited.
  - Richer input. …not so much needed.
  - Multi-task to predict. …promising, but serious baselines needed.

Is NMT That Much Better?

The outputs of this year's best system: http://matrix.statmt.org/

SRC	A 28-year-old chef who had recently moved to San Francisco was found dead in the stairwell of a local mall this week.
	Osmadvacetiletý kuchař, který se nedávno přestěhoval do San Francisca, byl tento týden nalezen mrtvý na schodišti místního obchodního centra.
	Osmadvacetiletý šéfkuchař, který se nedávno přistěhoval do San Franciska, byl tento týden Ø schodech místního obchodu.
SRC	There were creative differences on the set and a disagreement.
	Došlo ke vzniku kreativních rozdílů na scéně a k neshodám.
	Na place byly tvůrčí rozdíly a neshody.

Is NMT That Much Better?

The outputs of this year's best system: http://matrix.statmt.org/

SRC	A 28-year-old chef who had recently moved to San Francisco was found dead in the stairwell of a local mall this week.
MT	Osmadvacetiletý kuchař, který se nedávno přestěhoval do San Francisca, byl tento týden nalezen mrtvý na schodišti místního obchodního centra.
REF	Osmadvacetiletý šéfkuchař, který se nedávno přistěhoval do San Franciska, byl tento týden Ø schodech místního obchodu.

SRC	There were creative differences on the set and a disagreement.
REF	Došlo ke vzniku kreativních rozdílů na scéně a k neshodám.
MT	Na place byly tvůrčí rozdíly a neshody.

Luckily ;-) Bad Errors Happen

SRC ... said Frank initially stayed in hostels...
MT ... řekl, že Frank původně zůstal v Budějovicích...

SRC Most of the Clintons' income...
MT Většinu příjmů Kliniky...

SRC The 63-year-old has now been made a special representative...
MT 63letý mladík se nyní stal zvláštním zástupcem...

SRC He listened to the moving stories of the women.
MT Naslouchal pohyblivým příběhům žen.

Catastrophic Errors

SRC Criminal Minds star Thomas Gibson sacked after hitting producer

REF Thomas Gibson, hvězda seriálu Myšlenky zločince, byl propuštěn po té, co uhodil režiséra

MT Kriminalisté Minsku hvězdu Thomase Gibsona vyhostili po zásahu producenta

SRC ...add to that its long-standing grudge...

REF ...přidejte k tomu svou dlouholetou nenávist...

MT ...přidejte k tomu svou dlouholetou záštitu...
(grudge → zášť → záštita)

German→Czech SMT vs. NMT

- A smaller dataset, very first (but comparable) results.
- NMT performs better on average, but occasionally:

SRC	Das Spektakel ähnelt dem Eurovision Song Contest.
REF	Je to jako pěvecká soutěž Eurovision.
SMT	Podívanou připomíná hudební soutěž Eurovize.
NMT	Divadlo se podobá Eurovizi Conview.

SRC	Erderwärmung oder Zusammenstoß mit Killerasteroid.
REF	Globální oteplení nebo kolize se zabijáckým asteroidem.
SMT	Globální oteplování, nebo srážka s Killerasteroid.
NMT	Globální oteplování, nebo střet s zabijákem.

SRC	Zu viele verletzte Gefühle.
REF	Příliš mnoho nepřátelských pocitů.
SMT	Příliš mnoho zraněných pocity.
NMT	Příliš mnoho zraněných ∅.

Ultimate Goal of SMT vs. NMT

Goal of "classical" SMT:

Find minimum translation units ~ graph partitions:

- such that they are frequent across many sentence pairs.
- without imposing (too hard) constraints on reordering.
- in an unsupervised fashion.

Goal of neural MT:

Avoid minimum translation units. Find NN architecture that

- Reads input in as original form as possible.
- Produces output in as final form as possible.
- Can be optimized end-to-end in practice.

Summary

- What makes MT statistical.

Two crucially different models covered:

- Phrase-based: contiguous but independent phrases.
  - Bayes Law as a special case of Log-Linear Model.
  - Hand-crafted features (scoring functions); local vs. non-local.
  - Decoding as search, expanding partial hypotheses.
- Neural: unit-less, continuous space.
  - NMT as a fancy Language Model.
  - Word embeddings, subwords.
  - RNNs for variable-length input and output.
  - Attention model and self-attention.
- Linguistic features in NMT.

References

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR, abs/1412.3555.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In Advances in neural information processing systems, pages 3104–3112.