An Introduction to Deep Learning

Marc'Aurelio Ranzato
Facebook AI Research
ranzato@fb.com

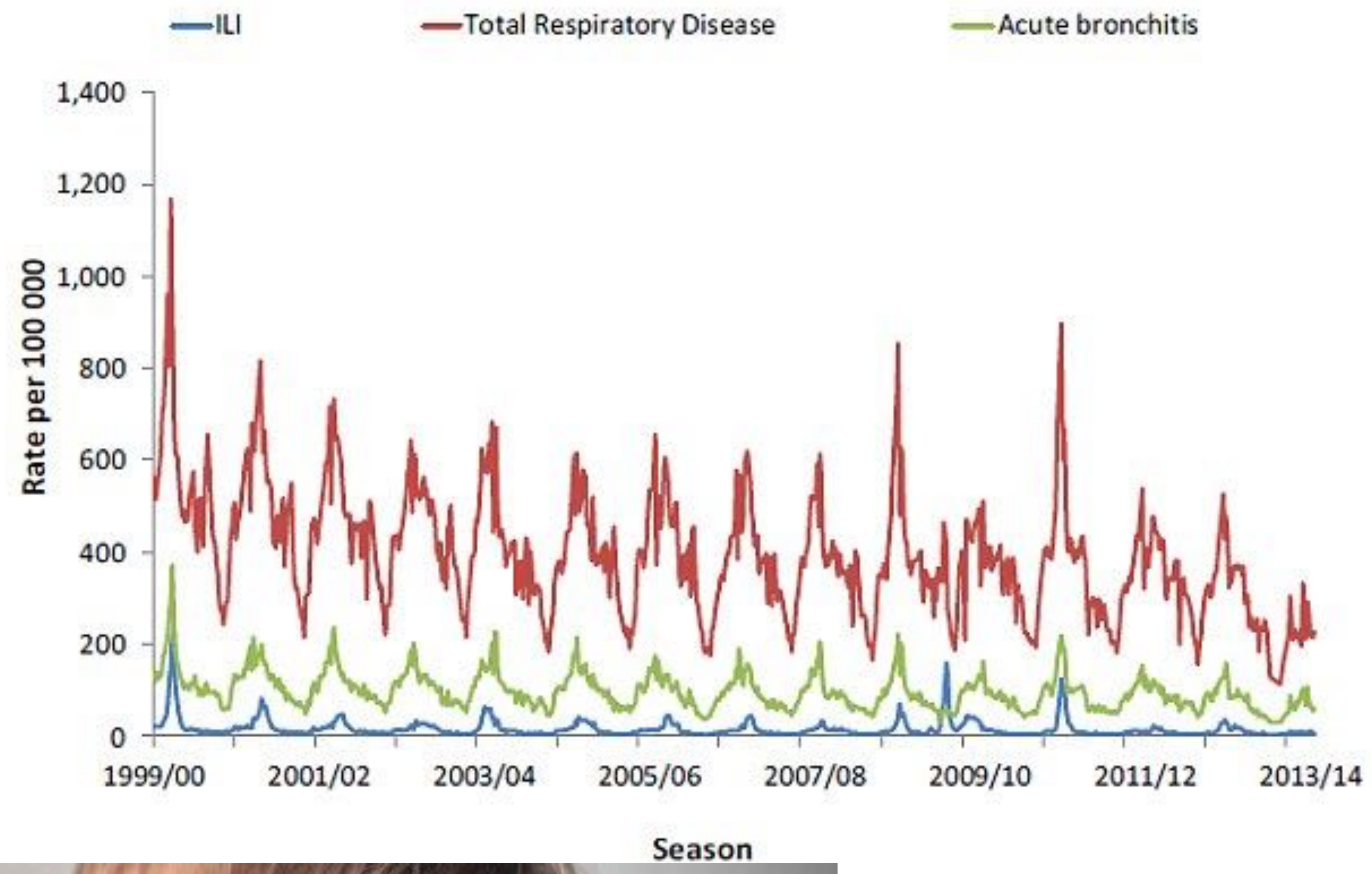
DeepLearn Summer School - Bilbao, 17 July 2017

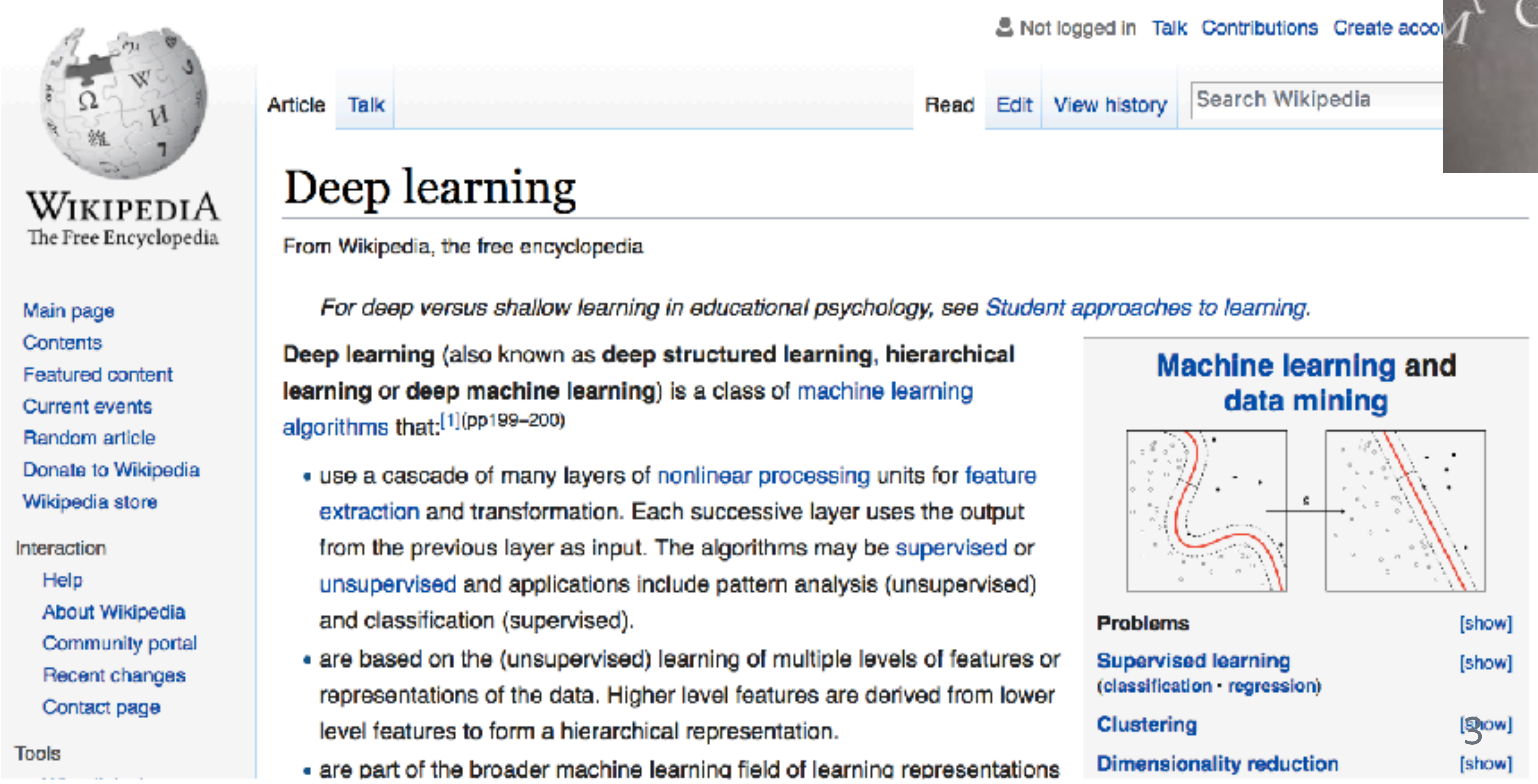
Outline

- **PART 0** [lecture 1]
  - Motivation
  - Training Fully Connected Nets with Backpropagation
- **Part 1** [lecture 1 and lecture 2]
  - Deep Learning for Vision: CNN
- **Part 2** [lecture 2]
  - Deep Learning for NLP
- **Part 3** [lecture 3]
  - **Modeling sequences**

Motivation

Data is often sequential in nature





Questions

— Deep learning tools to learn from and to predict sequences

- can standard tools like CNNs suffice?
- how about RNNs?

— fundamental problems when dealing with sequences

- is the sequential structure important for the prediction task?
- how to leverage structure at the input?
- how to deal with large output spaces? how to predict and what loss function to use?
- how to deal with variable length inputs/outputs? how to align sequences?

TL;DR...

There is no general rule of thumb, it depends on the task and constraints at hand.

Next, we will learn by reviewing several examples.

Learning Scenarios

Output Sequential?

Input Sequential?	no	yes
no		?
yes	?	?

Learning Scenarios: sequence -> single label

Output Sequential?

Input Sequential?		**no**	yes
	no		?
	yes	X	?

Examples:
- text classification
- language modeling
- action recognition
- music genre classification

Sequence->Single Label: Text Classification

Examples

Sentiment analysis

"I've had this place bookmarked for such a long time and I finally got to go!! I was not disappointed... " -> positive rating

Text classification

"Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on ...." -> science

General problem:

Given a document (ordered sequence of words), predict a single label.

Challenge:

Efficiency VS accuracy trade-off.

Sequence->Single Label: Text Classification

Examples

Sentiment analysis

"I've had this place bookmarked for such a long time and I finally got to go!! I was not disappointed... " -> positive rating

Text classification

"Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on ...." -> science

Approach:

Embed words in R^d -> average embeddings -> apply a linear classifier.

Word order is lost. This partially remedied by embedding n-grams.

Bag of tricks for efficient text classification, Joulin et al. 2016

Sequence->Single Label: Text Classification

Examples

Sentiment analysis

negative

"I've had this place bookmarked for such a long time and I finally got to go!! I was not disappointed… " -> positive rating

Text classification

"Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on ….” -> science

Approach:

Embed words in R^d -> average embeddings -> apply a linear classifier.
Word order is lost. This partially remedied by embedding n-grams.

Bag of tricks for efficient text classification, Joulin et al. 2016

Sequence->Single Label: Text Classification

Examples

negative

Sentiment analysis

"I've had this place bookmarked for such a long time and I finally got to go!! I was not disappointed… " -> positive rating

positive

Text classification

"Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on ….." -> science

Approach:

Embed words in R^d -> average embeddings -> apply a linear classifier.

Word order is lost. This partially remedied by embedding n-grams.

Bag of tricks for efficient text classification, Joulin et al. 2016

Sequence->Single Label: Text Classification

Examples

negative

Sentiment analysis

“I’ve had this place bookmarked for such a long time and I finally got to go!! I was not disappointed...“ -> positive rating

positive

Text classification

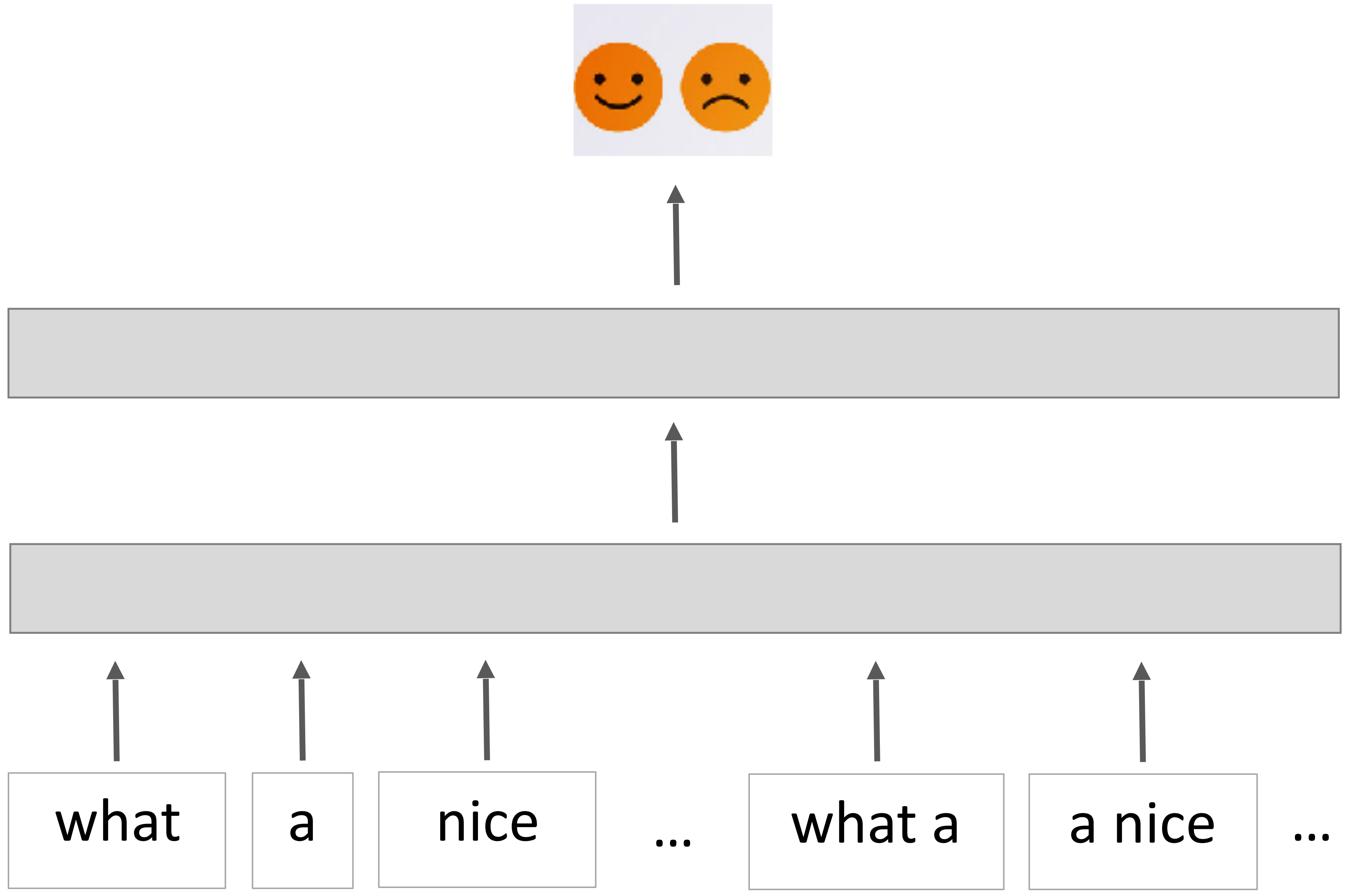
“Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on ....” -> science

Conclusion:

In this application (so far), bagging n-grams (n=1, 2, ...) works the best and is very efficient. No need to deal with sequential nature of the input!

fastText

- n-gram features at the input
- hashing
- hierarchical softmax
- product quantization of weights
- asynchronous training, “Hogwild”
- available at https://github.com/facebookresearch/fastText



Hogwild!..., Niu et al. 2011

Bag of tricks for efficient text classification, Joulin et al. 2016

FastText.zip: compressing text classification models, Joulin et al. 2017

credit: A. Joulin

fastText: results

CNN

	Zhang et al. (2015)		Conneau et al. (2016)		fastText	
AG	87.2	3h	91.3	51m	92.5	1s
Amz. F.	59.5	5d	63.0	7h	60.2	9s
DBpedia	98.3	5h	98.7	1h	98.5	2s
Yah. A.	71.2	1d	73.4	2h	72.3	5s
Yelp F.	62.0	-	64.7	1h12	63.9	4s

Accuracy and train time

Same accuracy – **1k-10K times faster!**

Bag of tricks for efficient text classification, Joulin et al. 2016
FastText.zip: compressing text classification models, Joulin et al. 2017

credit: A. Joulin

fastText: results

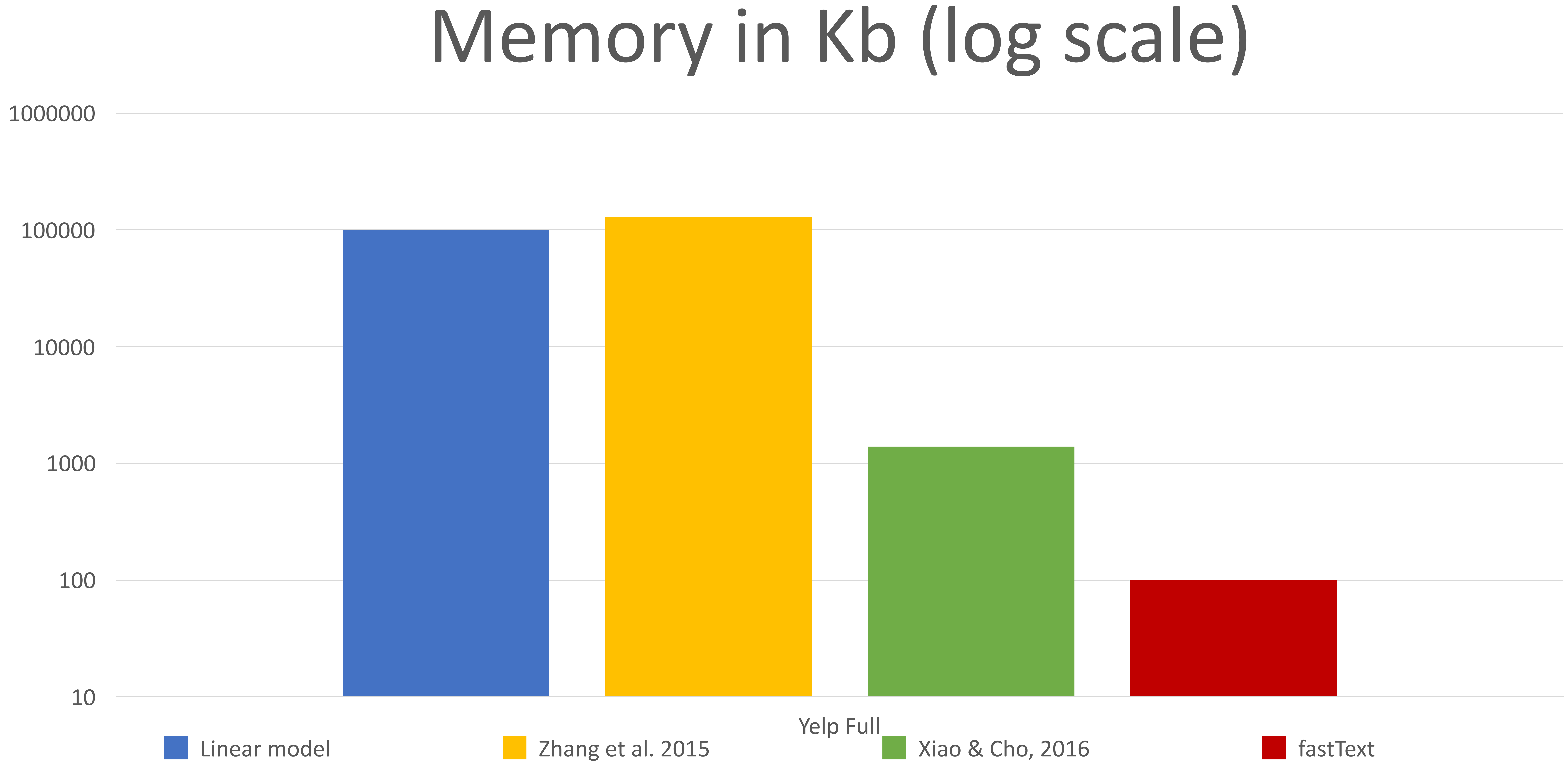
Model	prec@1	Running time	
		Train	Test
Freq. baseline	2.2	-	-
Tagspace (Weston et al., 2011)	35.6	5h32	15h
fastText	46.1	13m38	1m37

Results on Flickr. Prediction on 300K+ hashtags

Bag of tricks for efficient text classification, Joulin et al. 2016
FastText.zip: compressing text classification models, Joulin et al. 2017

credit: A. Joulin

fastText: results



Same accuracy – **1k-10K times faster + 10-100x smaller**

credit: A. Joulin

Bag of tricks for efficient text classification, Joulin et al. 2016

Sequence->Single Label: Language Modeling

Example

"Neural networks or connectionist systems are a computational ???" Task: replace ??? with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

$$p(w_t | w_{t-1} \dots w_1)$$

Challenges:

- very large vocabularies (> 100,000 words)
- long range dependencies (overall if working at the character level)

Sequence->Single Label: Language Modeling

Example

"Neural networks or connectionist systems are a computational ???" Task: replace ??? with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

$$p(w_t | w_{t-1} \ldots w_1)$$

Approaches:

- n-grams
- RNNs Exploring the limits of language modeling, Jozefowicz et al. 2016
- CNNs (more recently) Language modeling with gated convolutional networks, Dauphin et al. 2016

Sequence->Single Label: Language Modeling

Example

“Neural networks or connectionist systems are a computational **???**” Task: replace **???** with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

Approaches:

- **n-grams:** count-based, works well for head of distribution.

In order to estimate:

$$p(w_t | w_{t-1} \dots w_1)$$

we first make the Markov assumption that:

$$p(w_t | w_{t-1} \dots w_1) = p(w_t | w_{t-1} \dots w_{t-n+1})$$

and then we simply count:

$$p(w_t | w_{t-1} \dots w_{t-n+1}) = \frac{\text{count}(w_{t-n+1} \dots w_t)}{\text{count}(w_{t-n+1} \dots w_{t-1})}$$

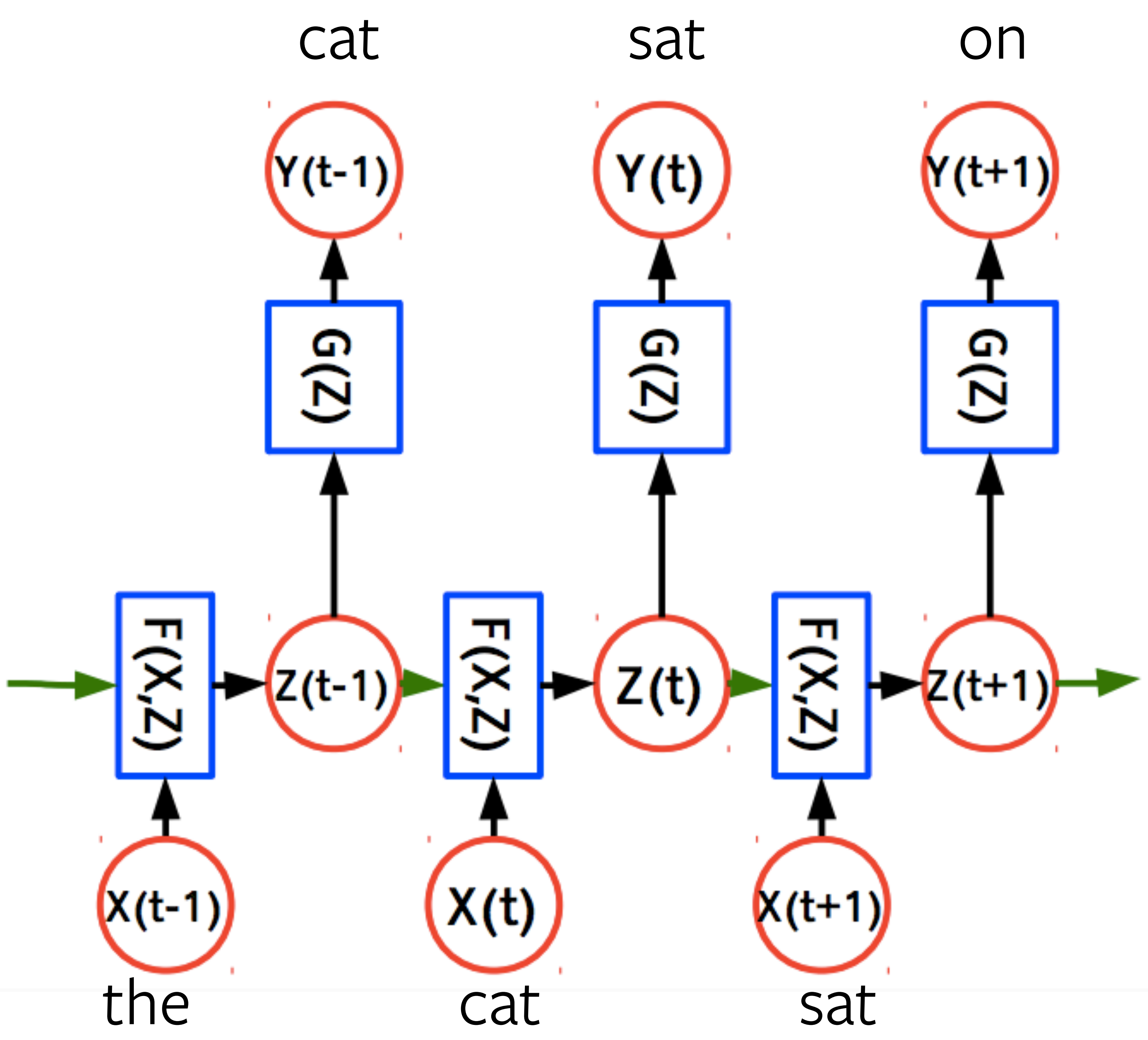
Sequence->Single Label: Language Modeling

Example

“Neural networks or connectionist systems are a computational ???” Task: replace ??? with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

Approaches:

- RNNs



Y. LeCun’s diagram

Sequence->Single Label: Language Modeling

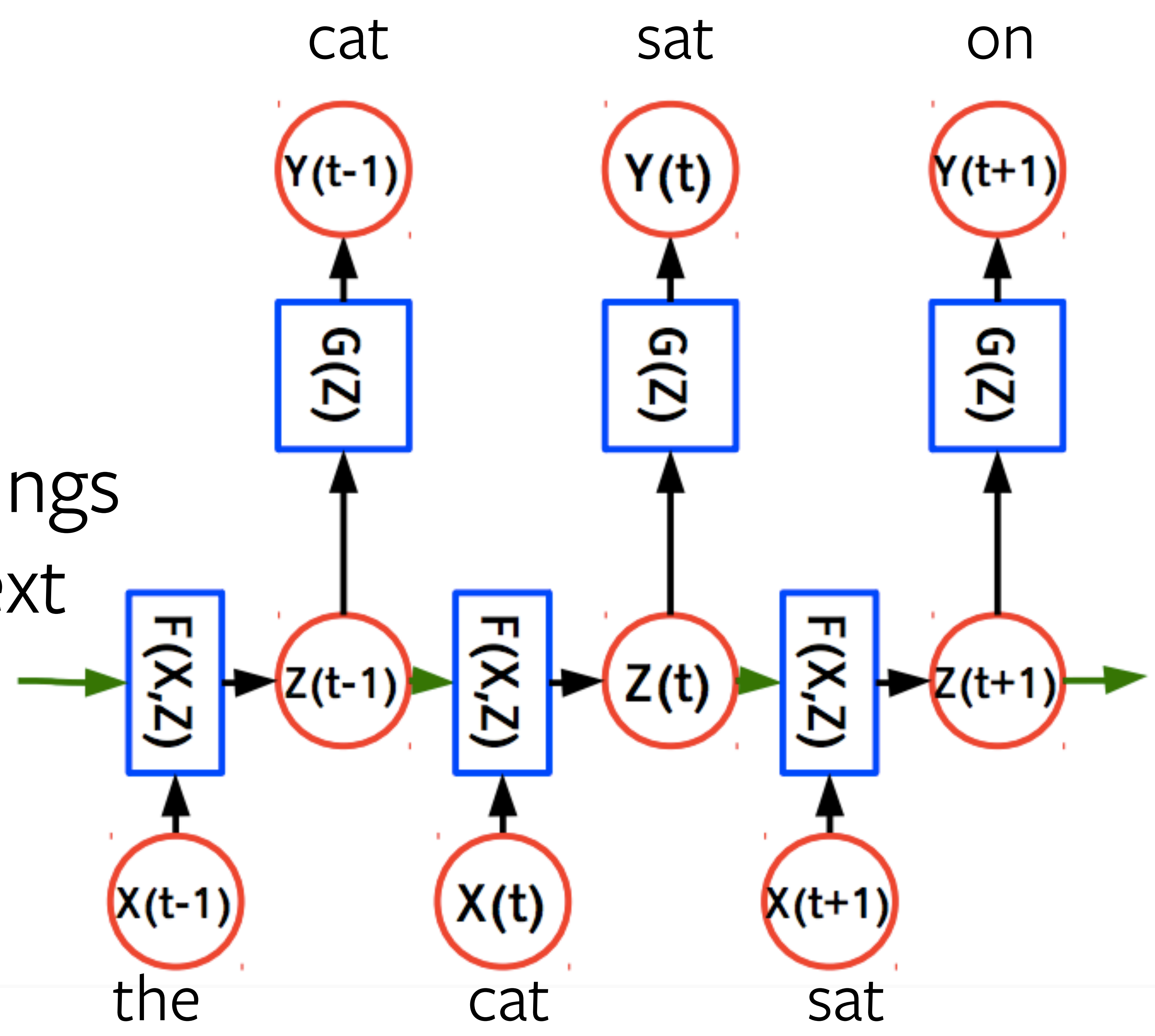
Example

"Neural networks or connectionist systems are a computational ???" Task: replace ??? with the correct word from the dictionary (useful for type-ahead and ASR, for instance).

Approaches:

- RNNs

+ it generalizes better thanks to embeddings
+ it can more easily capture longer context
- it's sequential, tricky to train



Y. LeCun's diagram

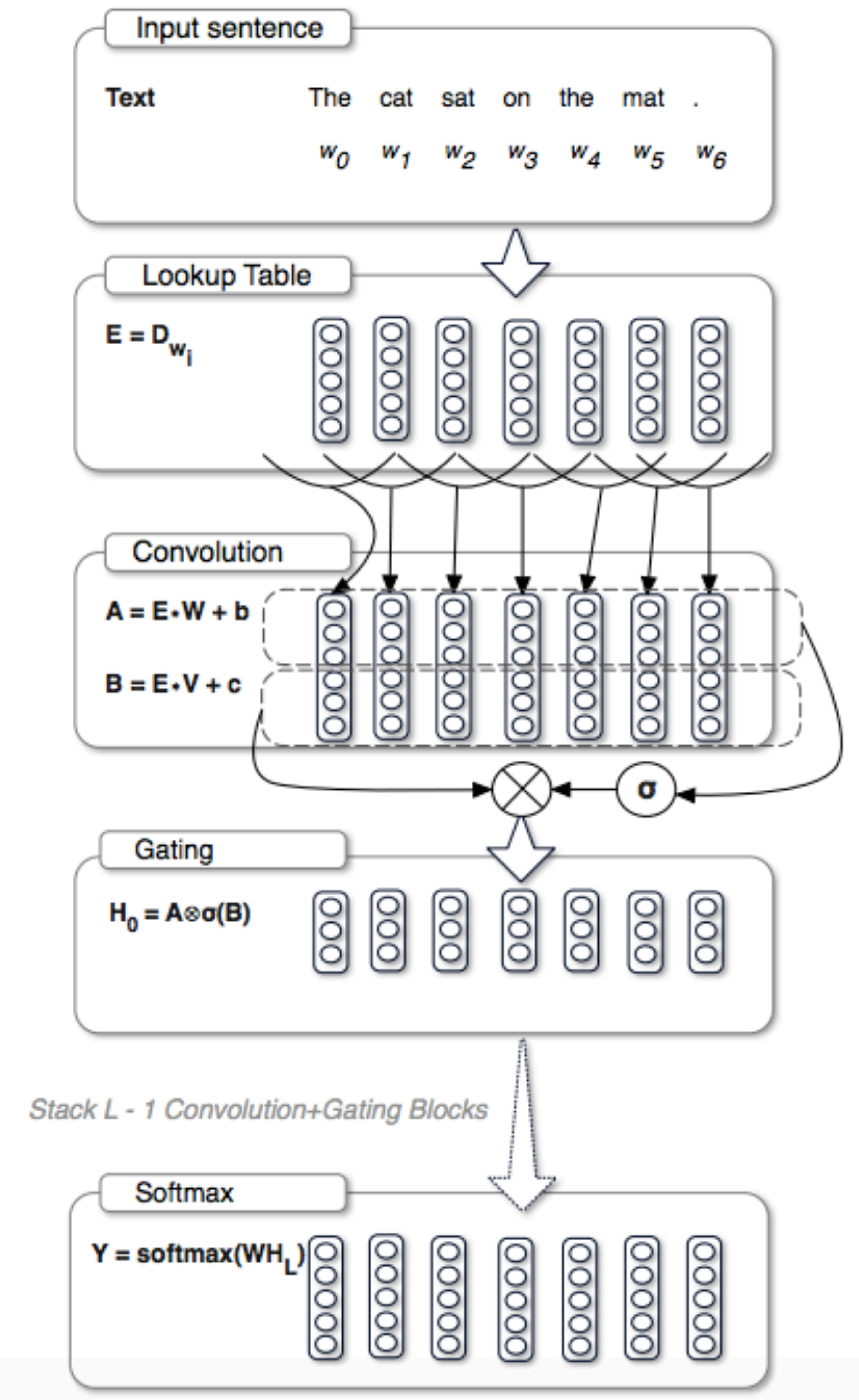
Fun demo with a charRNN:
http://www.cs.toronto.edu/~ilya/rnn.html

Sequence->Single Label: Language Modeling

Approaches:

- CNNs

+ same generalization as RNN
+ more parallelizable than RNNs
- fixed context (but it does not matter)



Language modeling with gated convolutional networks, Dauphin et al. 2016

Sequence->Single Label: Language Modeling

Model	Test PPL	Hardware
Sigmoid-RNN-2048 (Ji et al., 2015)	68.3	1 CPU
Interpolated KN 5-Gram (Chelba et al., 2013)	67.6	100 CPUs
Sparse Non-Negative Matrix LM (Shazeer et al., 2014)	52.9	-
RNN-1024 + MaxEnt 9 Gram Features (Chelba et al., 2013)	51.3	24 GPUs
LSTM-2048-512 (Jozefowicz et al., 2016)	43.7	32 GPUs
2-layer LSTM-8192-1024 (Jozefowicz et al., 2016)	30.6	32 GPUs
BIG GLSTM-G4 (Kuchaiev & Ginsburg, 2017)	23.3*	8 GPUs
LSTM-2048 (Grave et al., 2016a)	43.9	1 GPU
2-layer LSTM-2048 (Grave et al., 2016a)	39.8	1 GPU
GCNN-13	38.1	1 GPU
GCNN-14 Bottleneck	31.9	8 GPUs

Table 2. Results on the Google Billion Word test set. The GCNN outperforms the LSTMs with the same output approximation.

Language modeling with gated convolutional networks,
Dauphin et al. 2016

Sequence->Single Label: Language Modeling

Conclusion:

In language modeling, it is essential to take into account the sequential structure of the input.

RNNs/CNNs work the best at the moment.

Sequence->Single Label: Action Recognition

Playing Tennis

Challenges:

- how to aggregate information over time
- computational efficiency

Two stream convolutional network for action recognition in videos. Simonyan et al. NIPS 2014

Sequence->Single Label: Action Recognition

Playing Tennis

Approaches:

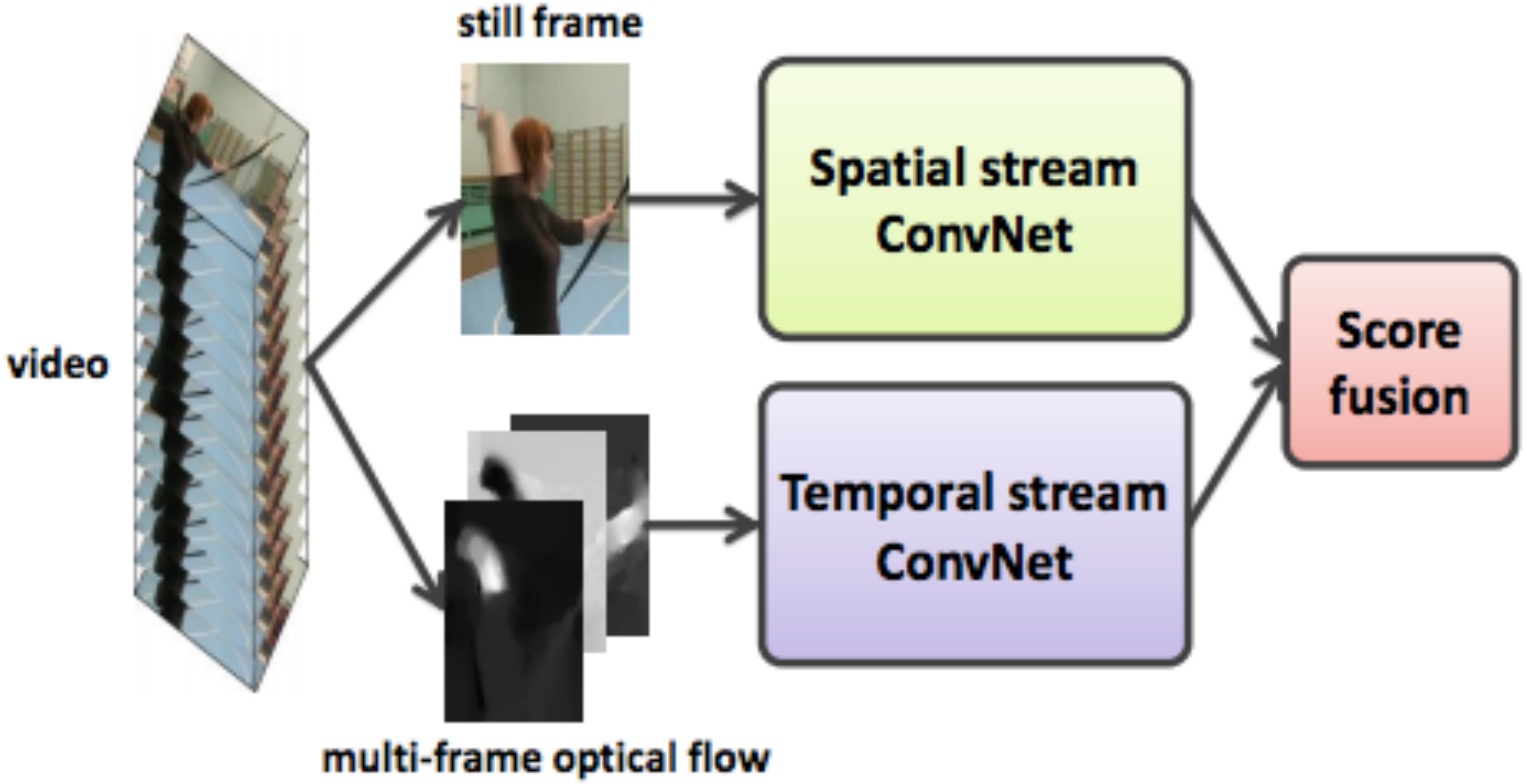
- CNN on static frames -> feature pooling over time -> classification. Possibly augmented with optical flow or (learned) temporal features.

Current large datasets have peculiar biases. E.g.,: one can often easily recognize the action from static frames by just looking at the context....

Two stream convolutional network for action recognition in videos. Simonyan et al. NIPS 2014

Sequence->Single Label: Action Recognition

Playing Tennis



Two stream convolutional network for action recognition in videos. Simonyan et al. NIPS 2014

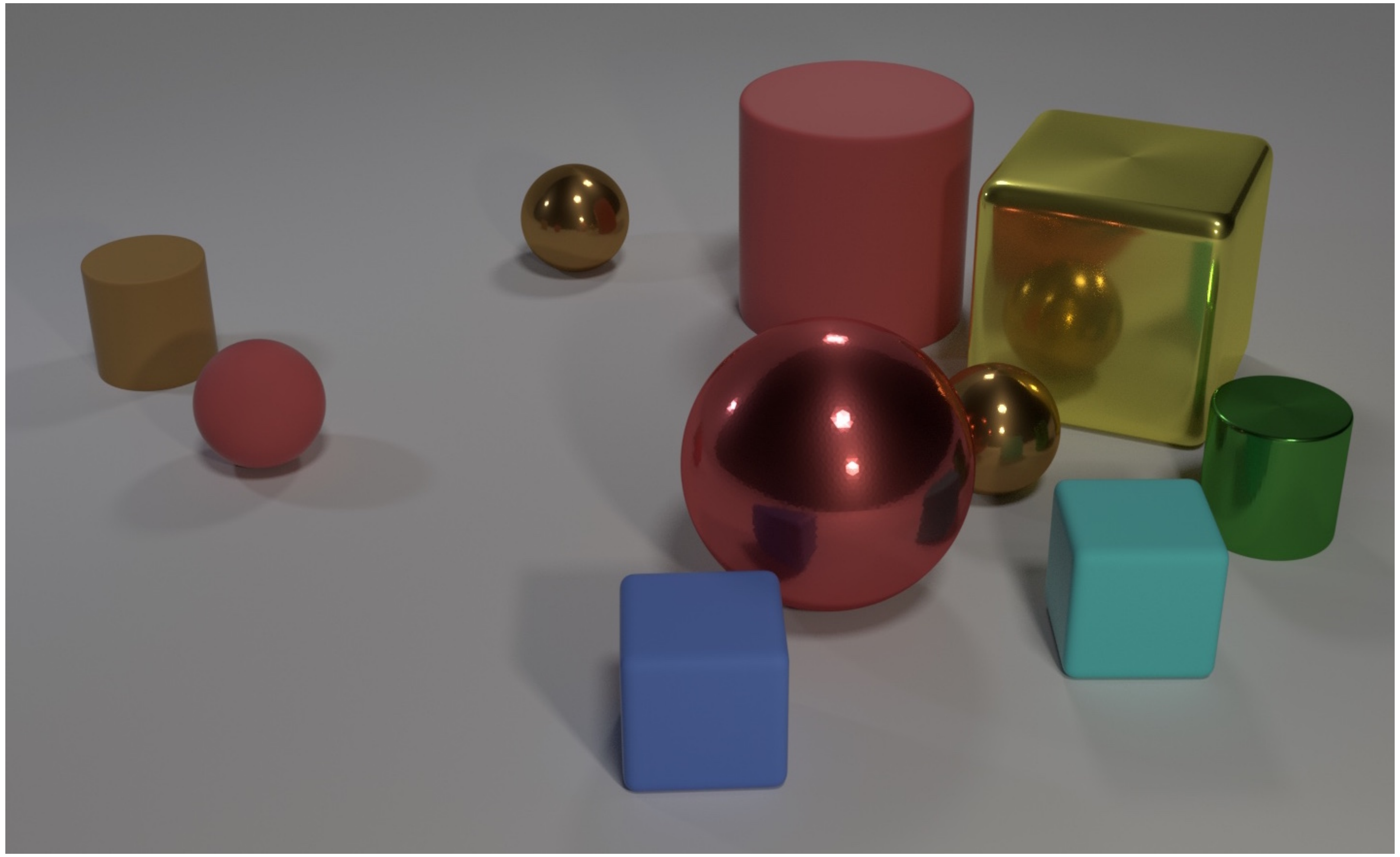
Sequence->Single Label: Action Recognition

Playing Tennis

Conclusion:

Methods and approaches heavily depend on the dataset used. Sometimes, the sequential structure does not add much information, if the label already correlates well with what can be found in static frames.

Sequence->Single Label: VQA



Q: Are there an equal number of large things and metal spheres?

Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere?

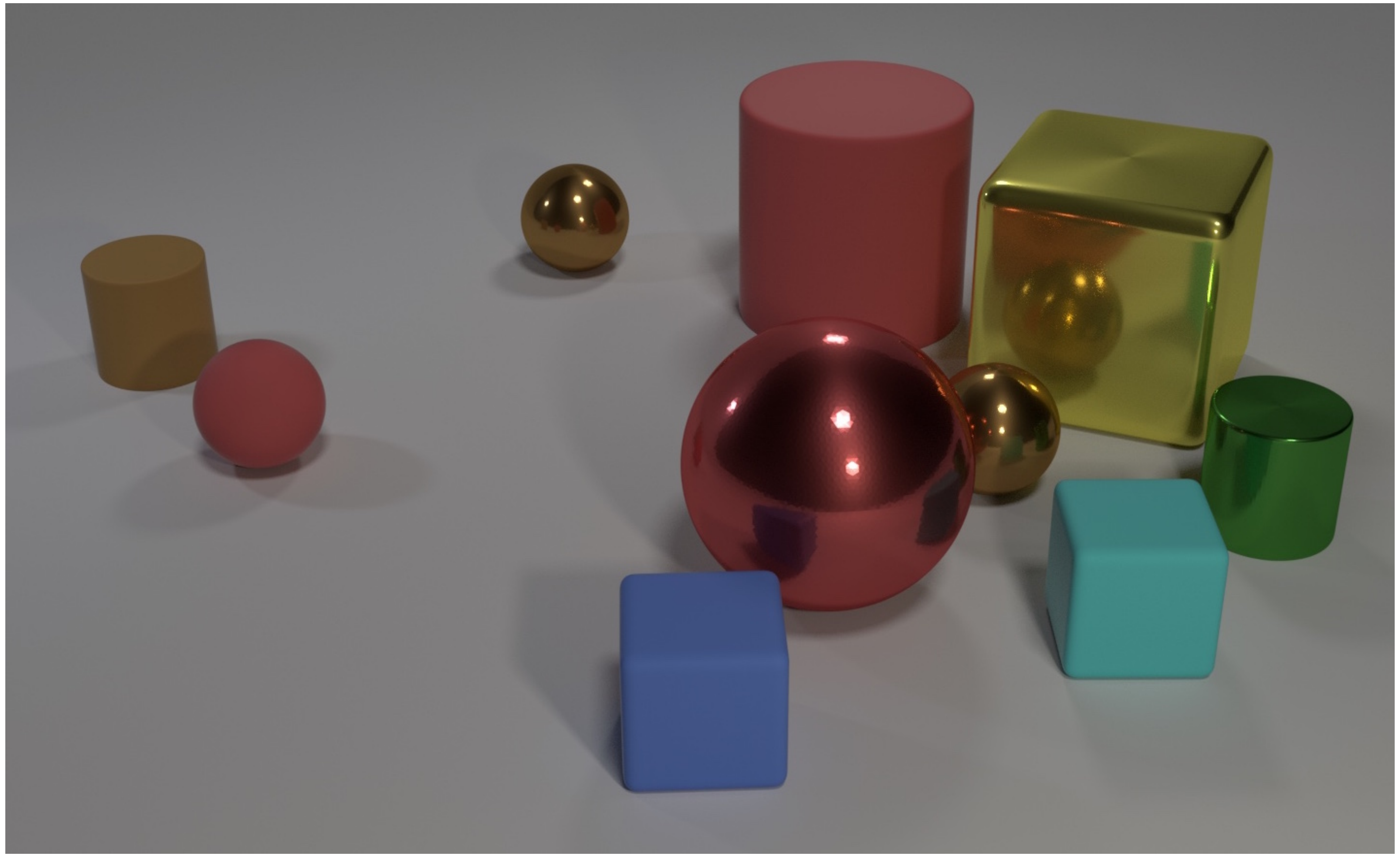
Q: There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?

Q: How many objects are either small cylinders or metal things?

Johnson et al, “CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning”, CVPR 2017

credit: R. Girshick

Sequence->Single Label: VQA



Q: Are there an **equal number** of **large** things and **metal spheres**?

Q: **What size** is the **cylinder that is left of** the **brown metal** thing **that is left of** the **big sphere**?

Q: There is a **sphere** with the **same size as** the **metal cube**; is it **made of the same material** as the **small red sphere**?

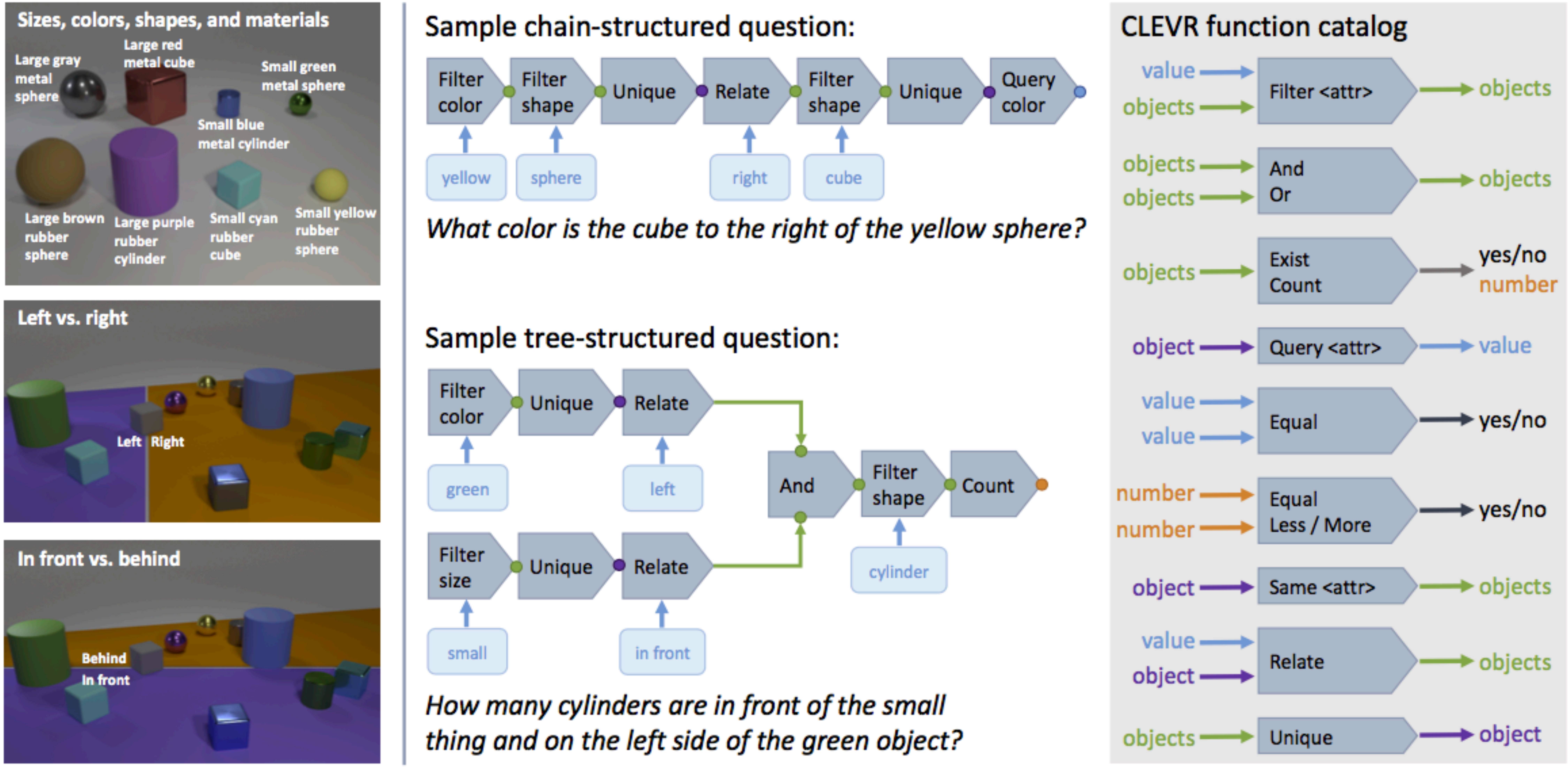
Q: **How many** objects are **either small cylinders or metal things**?

Attributes **Counting** **Comparison**
Spatial Relationships **Logical Operations**

Johnson et al, "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning", CVPR 2017

credit: R. Girshick

Sequence->Single Label: VQA

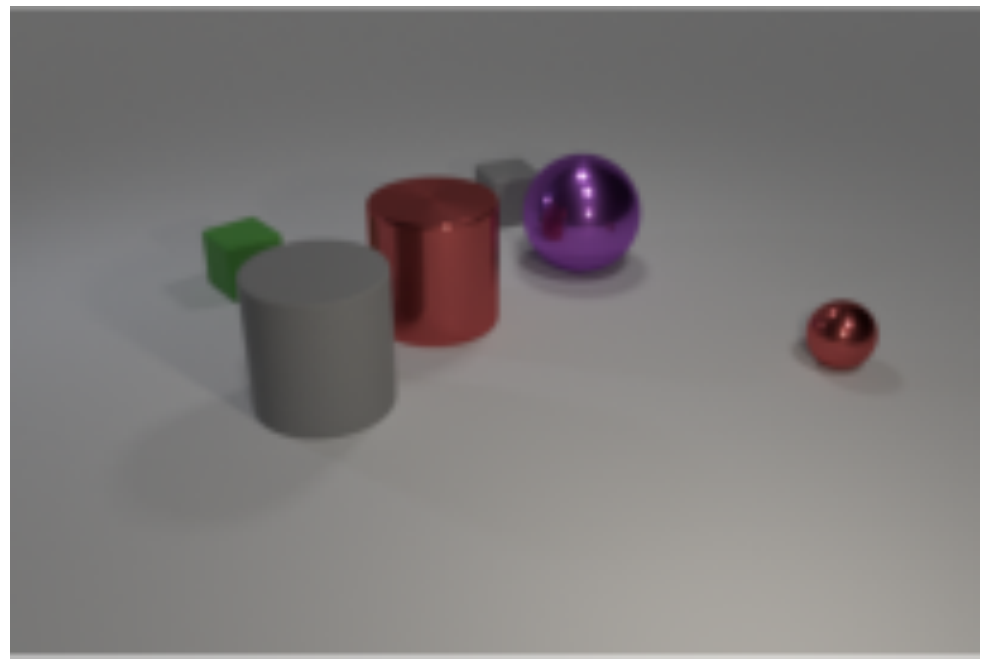


credit: R. Girshick

Sequence->Single Label: VQA

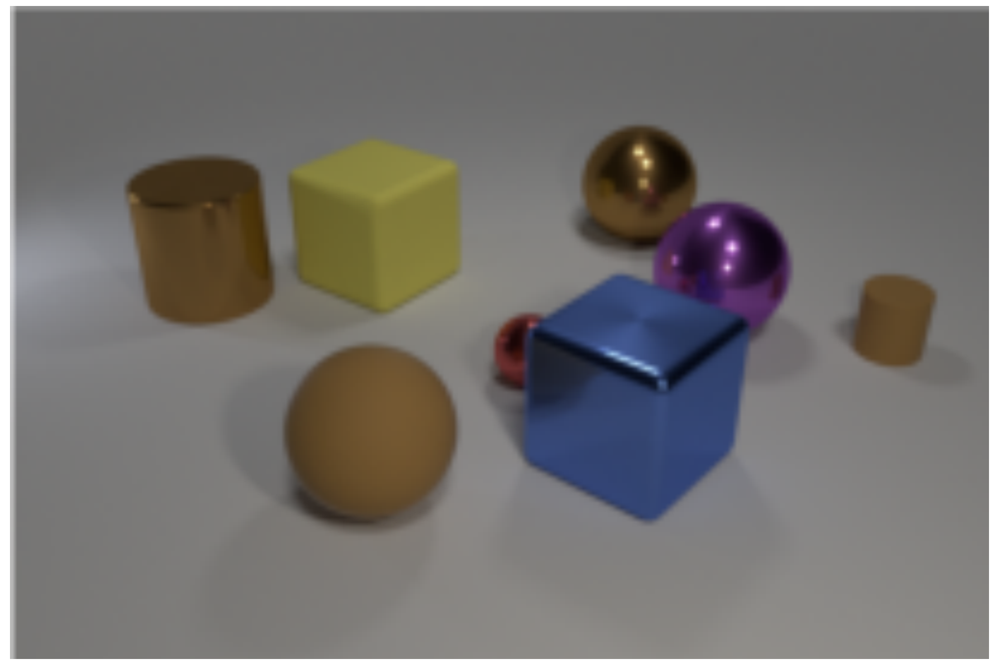
Question Types

Exist



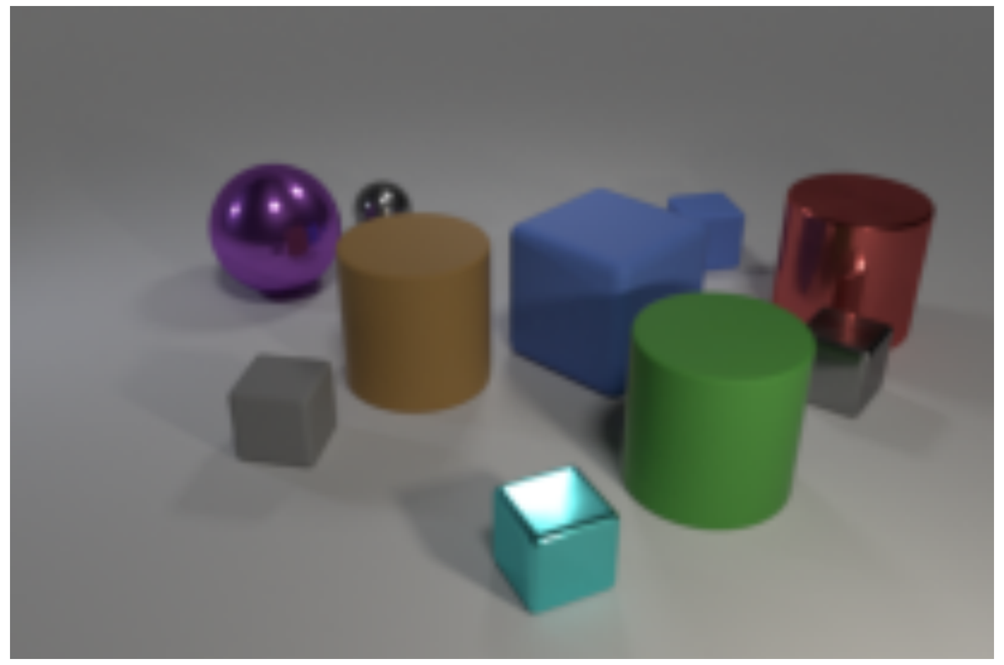
Q: Is there another green rubber cube that has the same size as the green matte cube?

Count

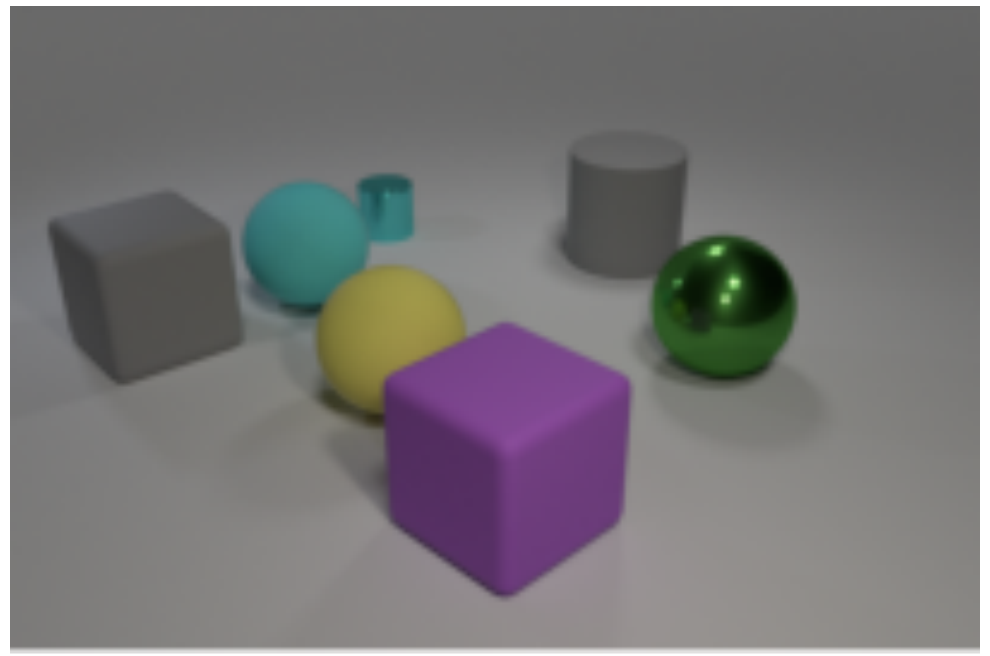


Q: There is a large cube that is right of the red sphere; what number of large yellow things are on the right side of it?

Compare number

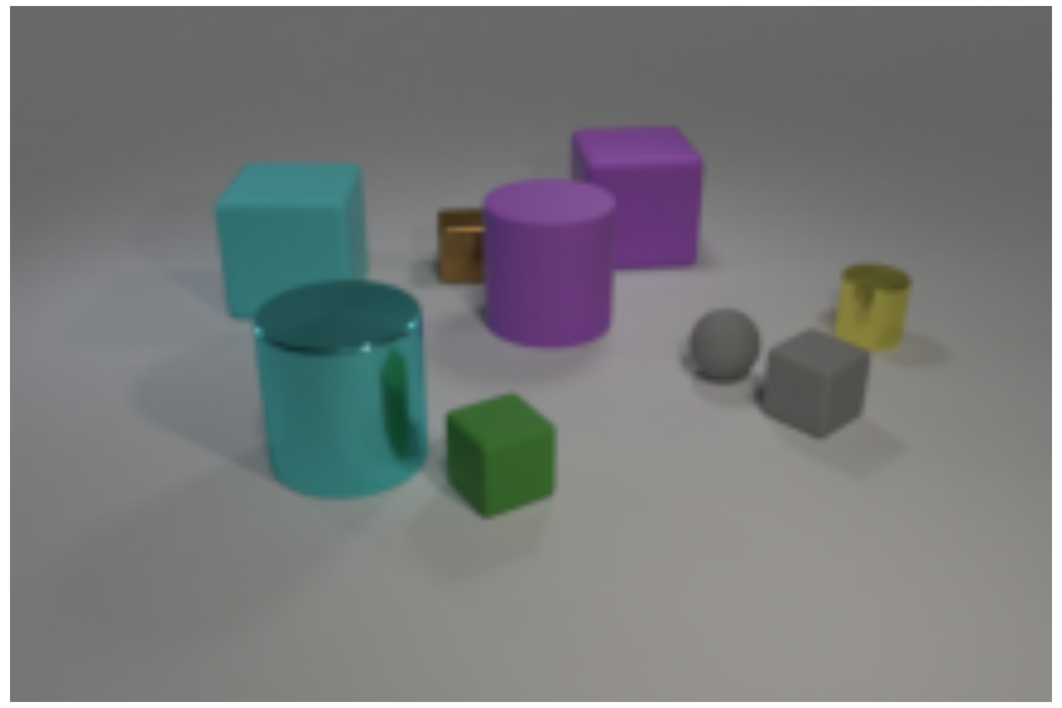


Q: Are there more metallic objects that are right of the large red shiny cylinder than gray matte objects?



Query

Q: There is a sphere to the right of the large yellow ball; what material is it?



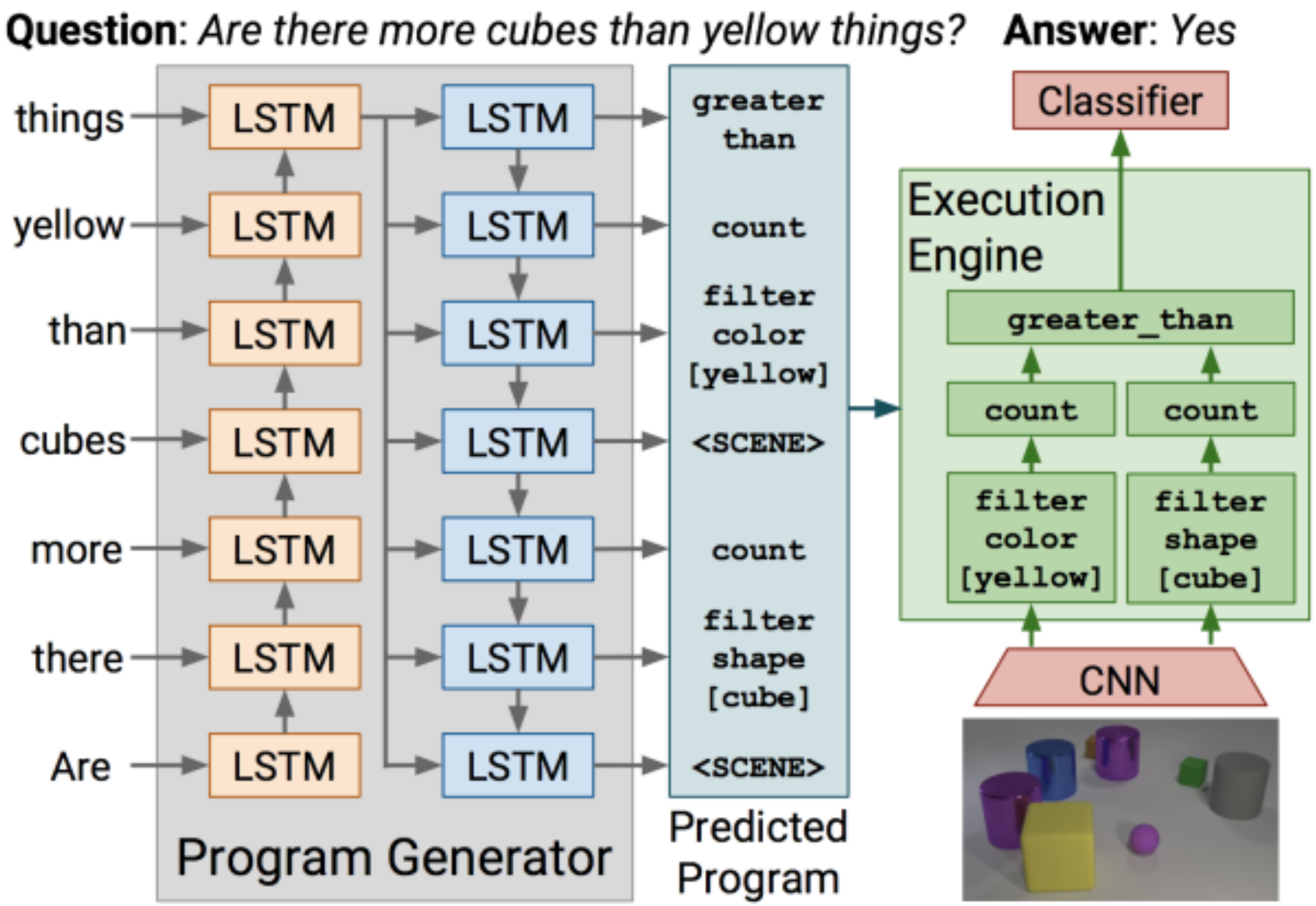
Compare Attribute

Q: Is the size of the cyan cube the same as the metal cylinder that is behind the cyan cylinder?

credit: R. Girshick

Sequence->Single Label: VQA

Compositional Reasoning: Model



credit: R. Girshick

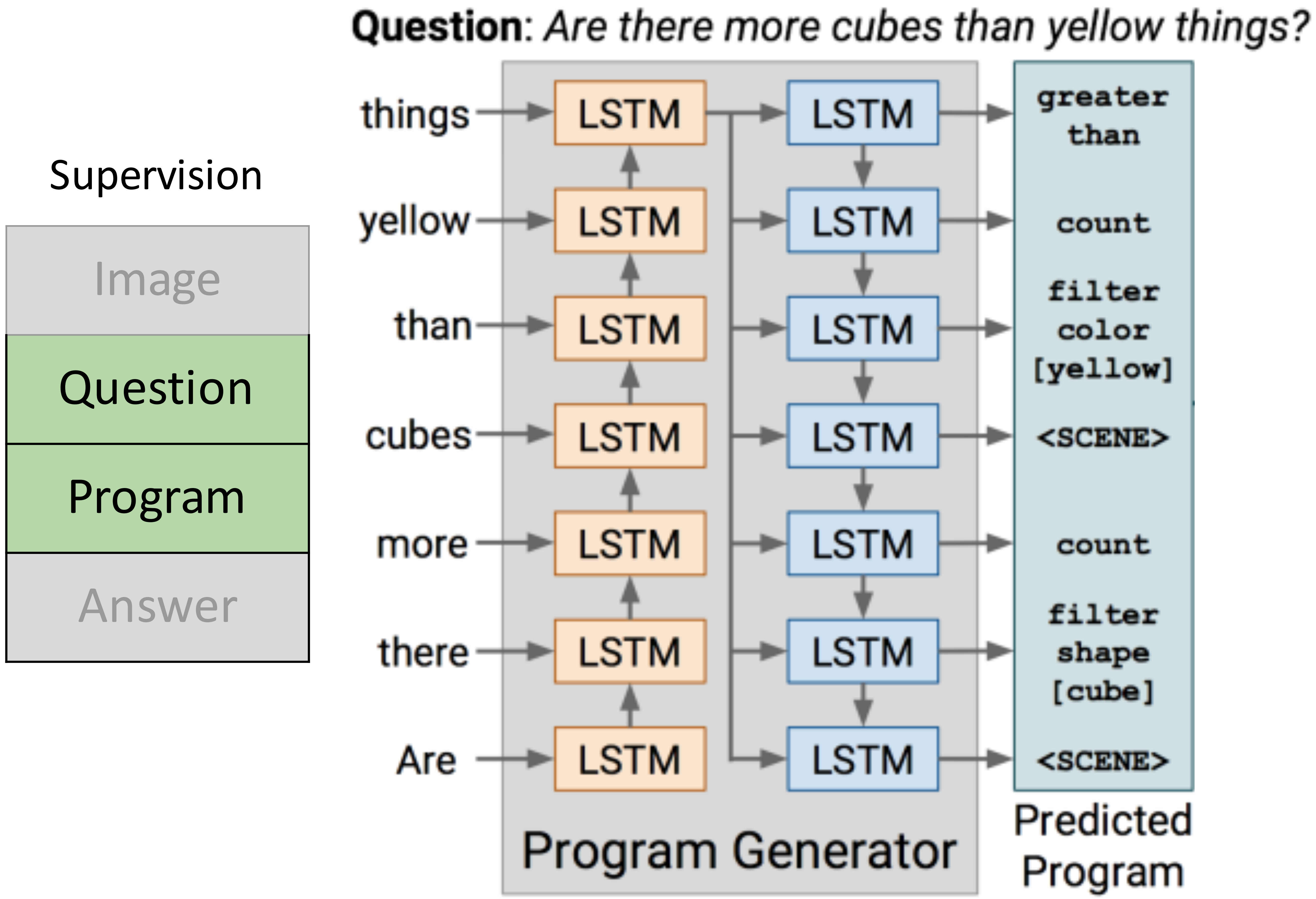
Johnson et al, "Inferring and Executing Programs for Visual Reasoning". 2017

Andreas et al, "Neural Module Networks", 2016

Andreas et al, "Learning to Compose Neural Networks for Question Answering", 2016

Sequence->Single Label: VQA

Step 1: Train Program Generator



(Using a small fraction of ground-truth programs)

Johnson et al, "Inferring and Executing Programs for Visual Reasoning". 2017

Andreas et al, "Neural Module Networks", 2016
Andreas et al, "Learning to Compose Neural Networks for Question Answering", 2016

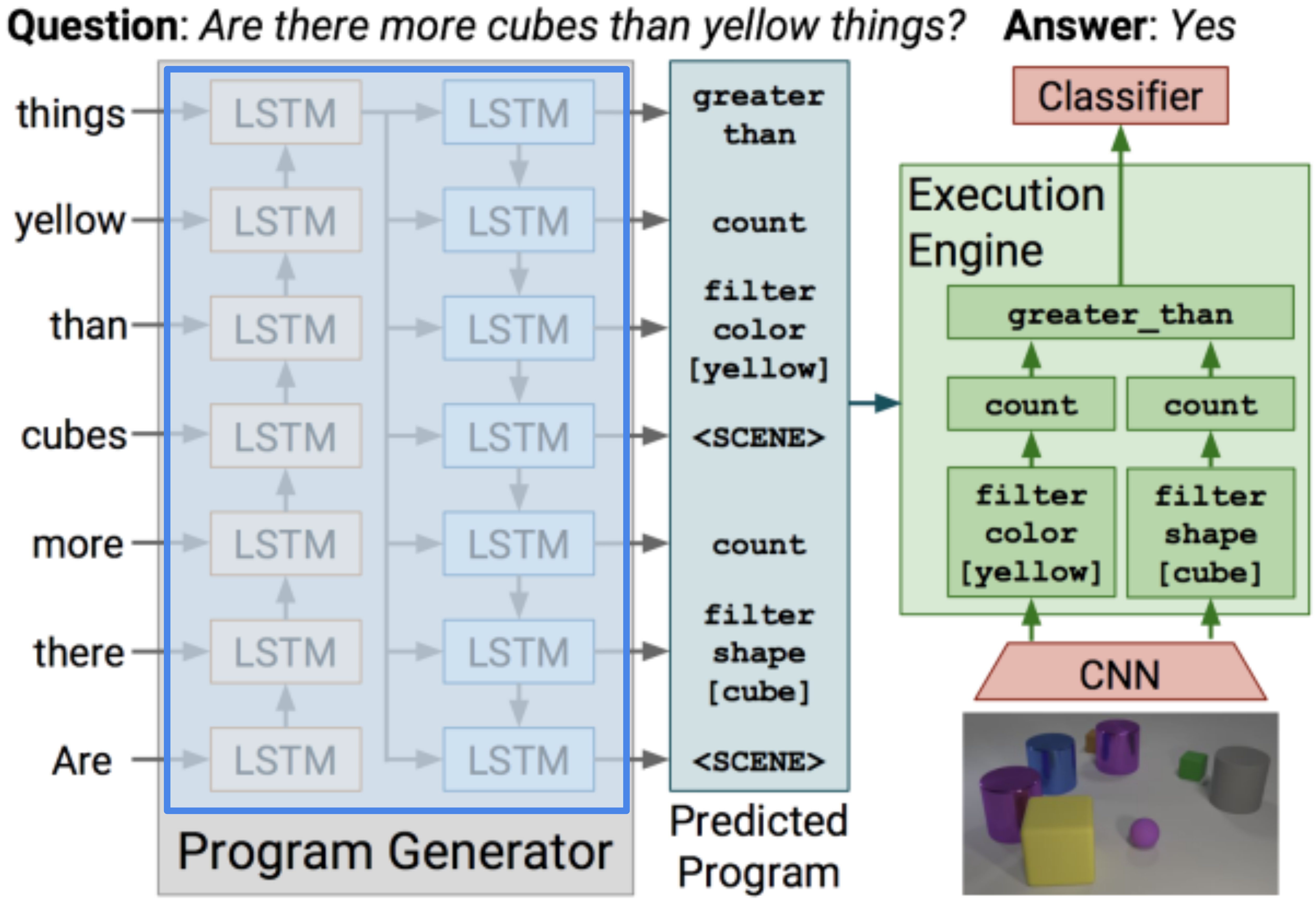
credit: R. Girshick

Sequence->Single Label: VQA

Step 2: Freeze PG, train Execution Engine

Supervision

Image
Question
Program
Answer



Johnson et al, "Inferring and Executing Programs for Visual Reasoning". 2017

Andreas et al, "Neural Module Networks", 2016
Andreas et al, "Learning to Compose Neural Networks for Question Answering", 2016

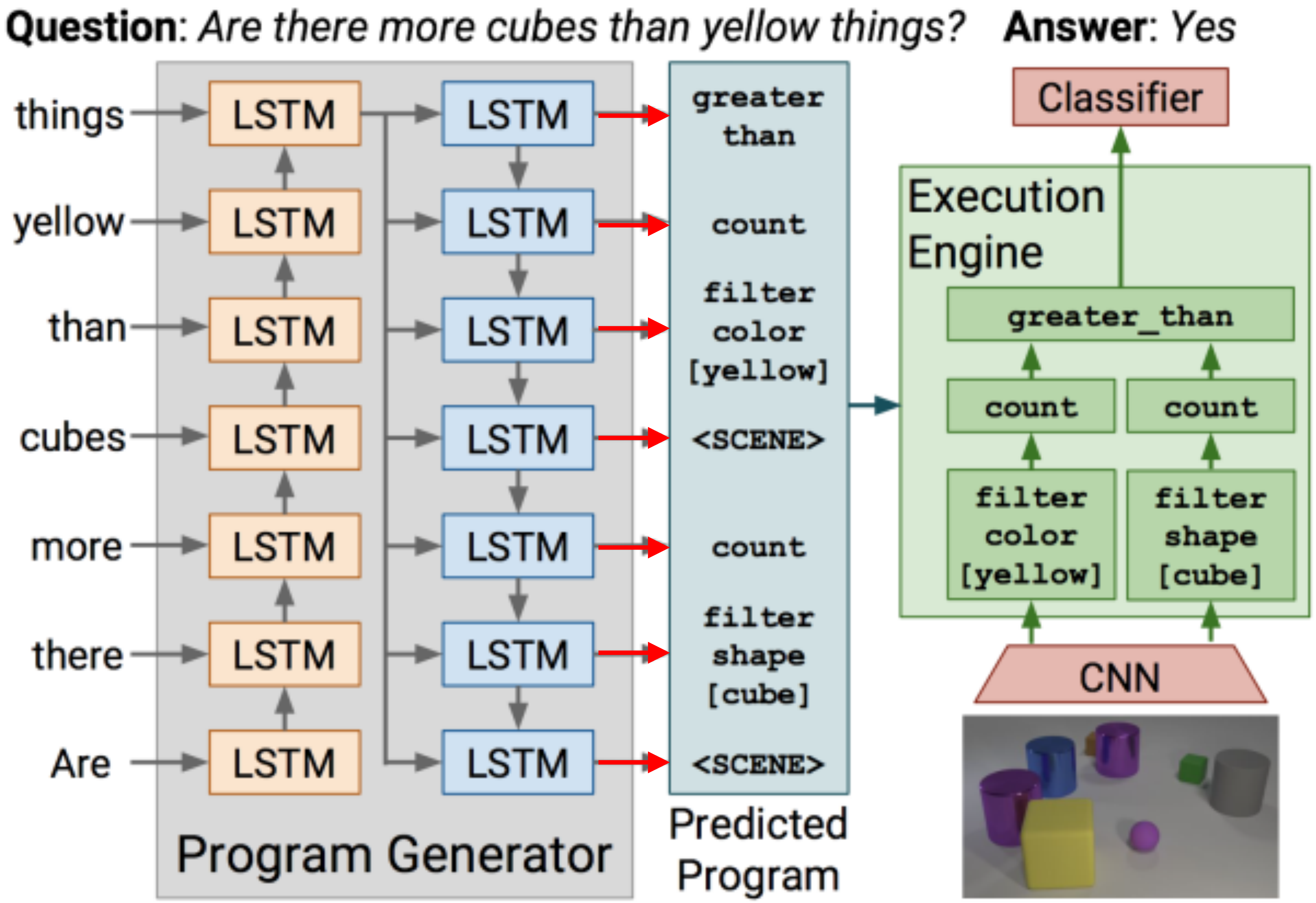
credit: R. Girshick

Sequence->Single Label: VQA

Step 3: Train jointly with REINFORCE

Supervision

Image
Question
Program
Answer



Andreas et al, “Neural Module Networks”, 2016

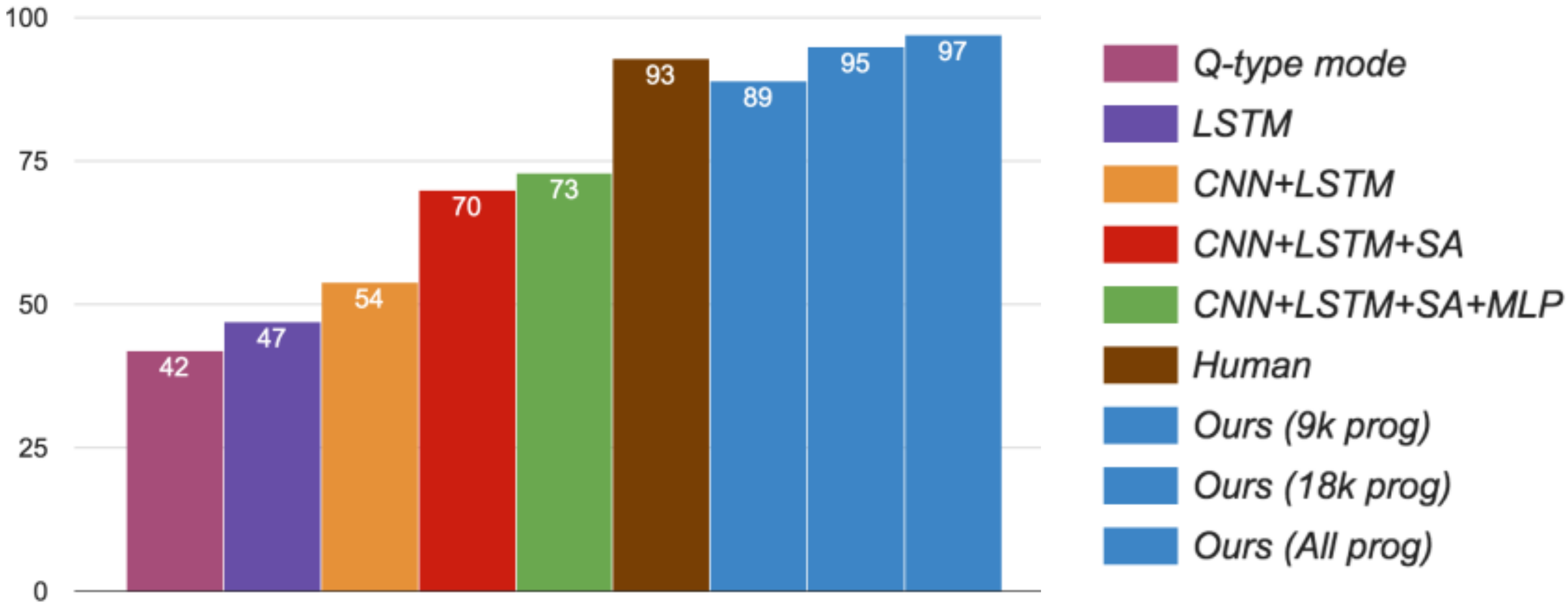
Johnson et al, “Inferring and Executing Programs for Visual Reasoning”. 2017

Andreas et al, “Learning to Compose Neural Networks for Question Answering”, 2016

credit: R. Girshick

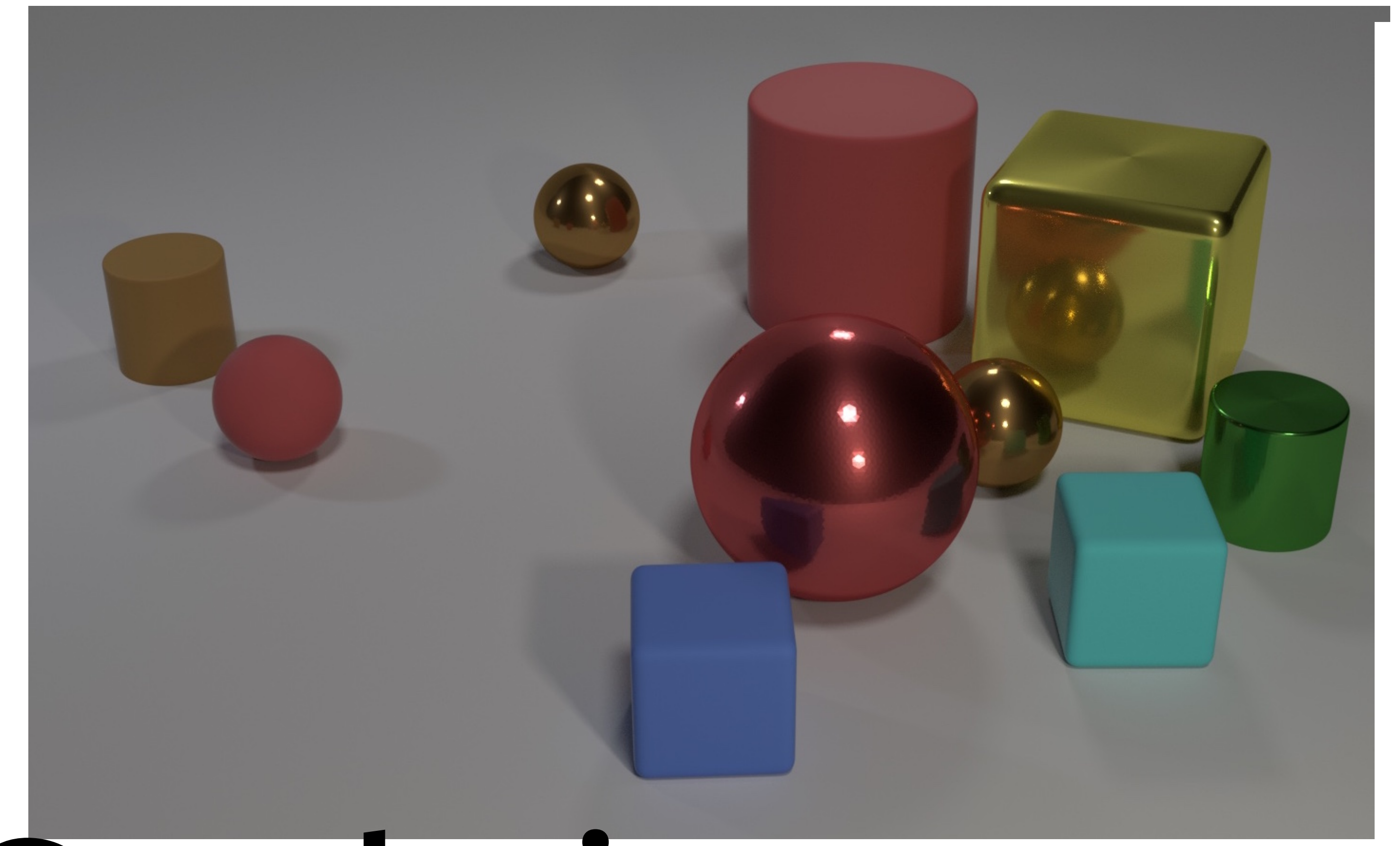
Sequence->Single Label: VQA

Accuracy on CLEVR



credit: R. Girshick

Sequence->Single Label: VQA



Conclusion:

In order to support compositional reasoning (even on rather artificial datasets like CLEVR), current models use rather complicated architectures (RNN + CNN + composable CNN).
Accuracy is good but supervision is unrealistically strong.
Recent approaches seem to reduce the amount of required supervision and replace top level composable CNN with a gated CNN. Unclear whether they can compose.

Learning visual reasoning without strong priors, Perez et al., arXiv 2017

Learning Scenarios: single input -> sequence

Input Sequential? \ Output Sequential?	no	yes
no		X
yes	X	?

Example:

- image captioning

Single input -> sequence: image captioning

Example:

A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.

Challenge:

- how to deal with multiple modalities.
- what to look for and where to look in the input image.
- uncertainty in the output: there are many good captions for a given image.
- What is a good metric of success?

Single input -> sequence: image captioning

Example:

A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.

Approach:

Pre-train a CNN to extract features from the image, and generate text conditioning an RNN with the image features.

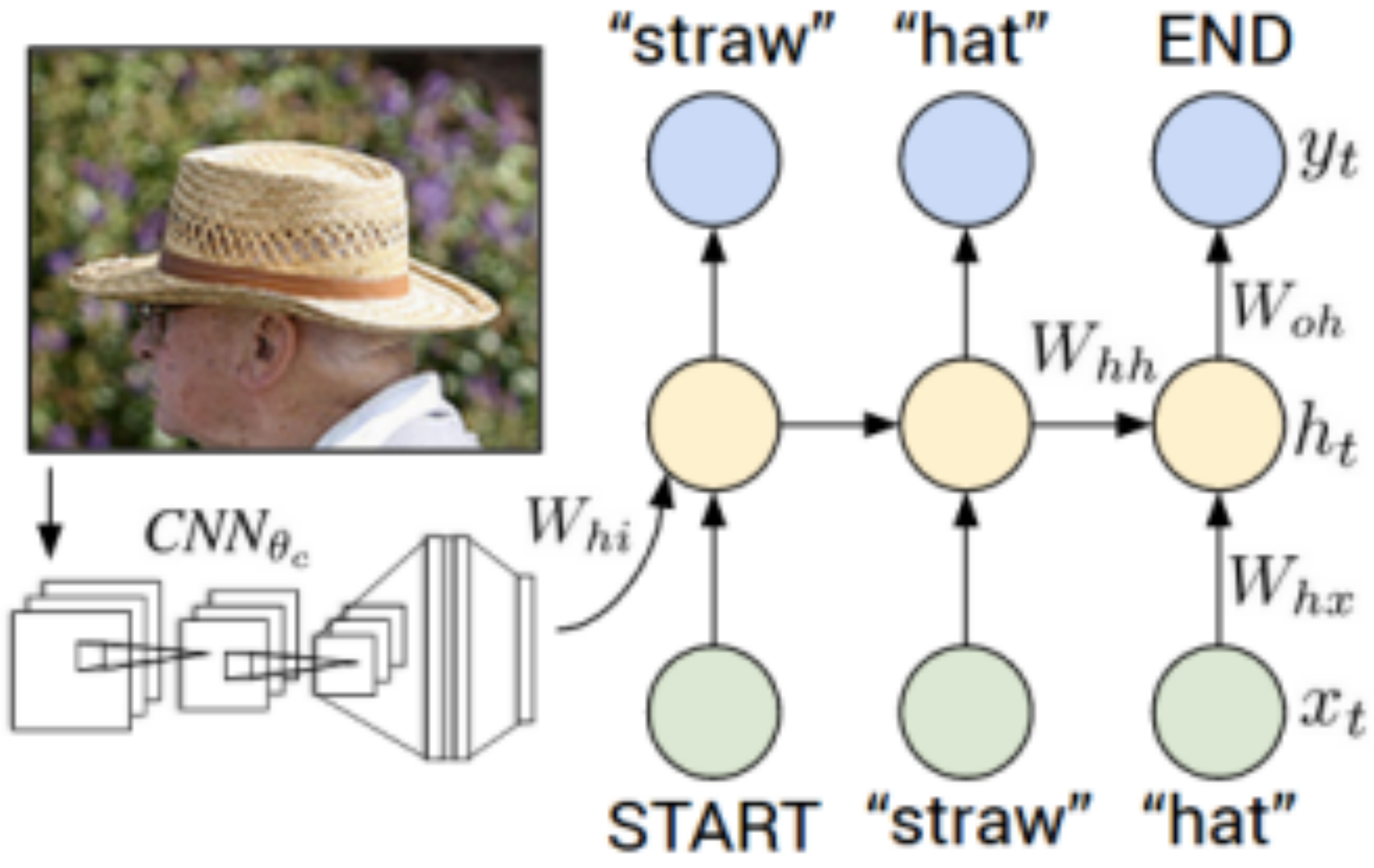
Deep visual semantic alignments for generating image descriptions, Karpathy et al. CVPR 2015

Single input -> sequence: image captioning

Example:

A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.

Approach:



Deep visual semantic alignments for generating image descriptions, Karpathy et al. CVPR 2015

Single input -> sequence: image captioning

Example:

A square with a fountain and tall buildings in the background, with some trees and a few people hanging out.

Conclusion:

It is easy to condition a language model (RNN or CNN based) with additional context, and ultimately map a static object into a sequence.
This however heavily relies on good pre-trained (on large labeled datasets) image features.

Learning Scenarios: sequence -> sequence

Output Sequential?

Input Sequential?	no	yes
no		X
yes	X	X

Examples:

- machine translation
- summarization
- speech recognition
- OCR
- video frame prediction

Sequence -> Sequence: machine translation

Example:

ITA: Il gatto si e' seduto sul tappetino.

EN: The cat sat on the mat.

Challenges:

- alignment: input/output sequences may have different length
- uncertainty (1-to-many mapping: many possible ways to translate)
- metric: how to automatically assess whether to sentences mean the same thing?

Sequence -> Sequence: machine translation

Example:

ITA: Il gatto si e' seduto sul tappetino.

↓

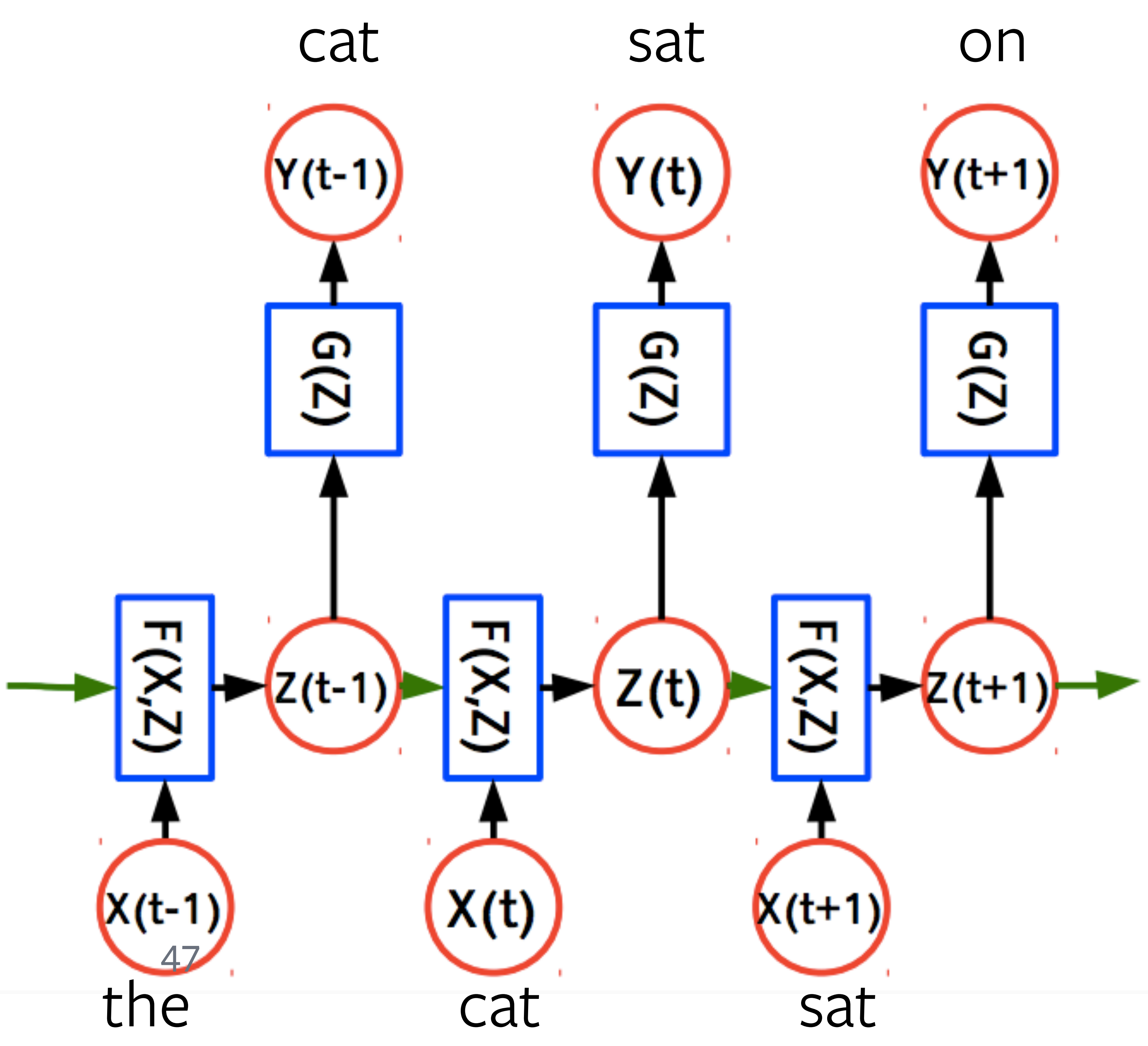
EN: The cat sat on the mat.

Approach:

Have one RNN to encode the source sentence, and another RNN to predict the target sentence. The target RNN learns to (soft) align via attention.

Neural machine translation by jointly learning to align and translate, Bahdanau et al. ICLR 2015

Sequence -> Sequence: machine translation

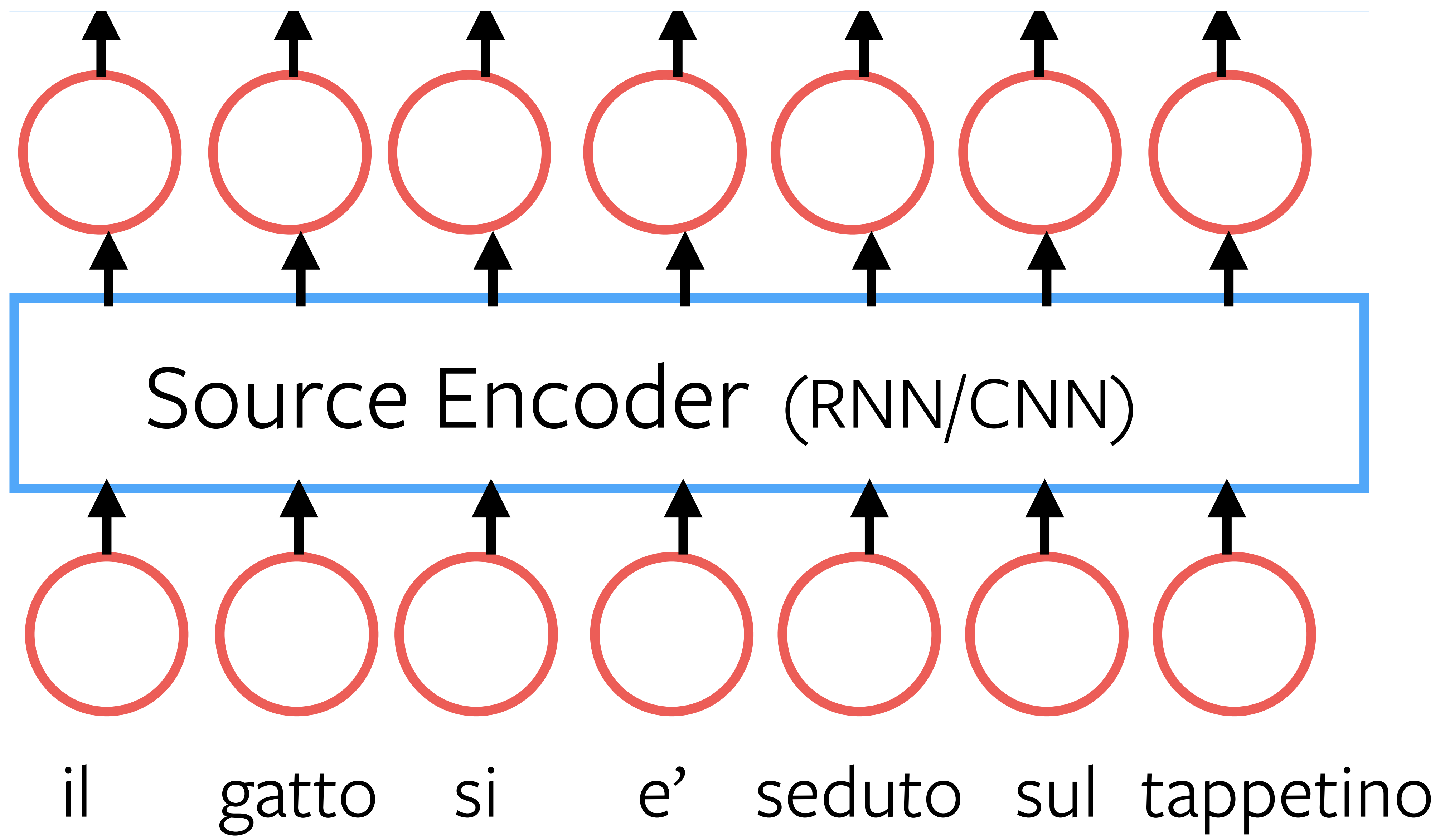


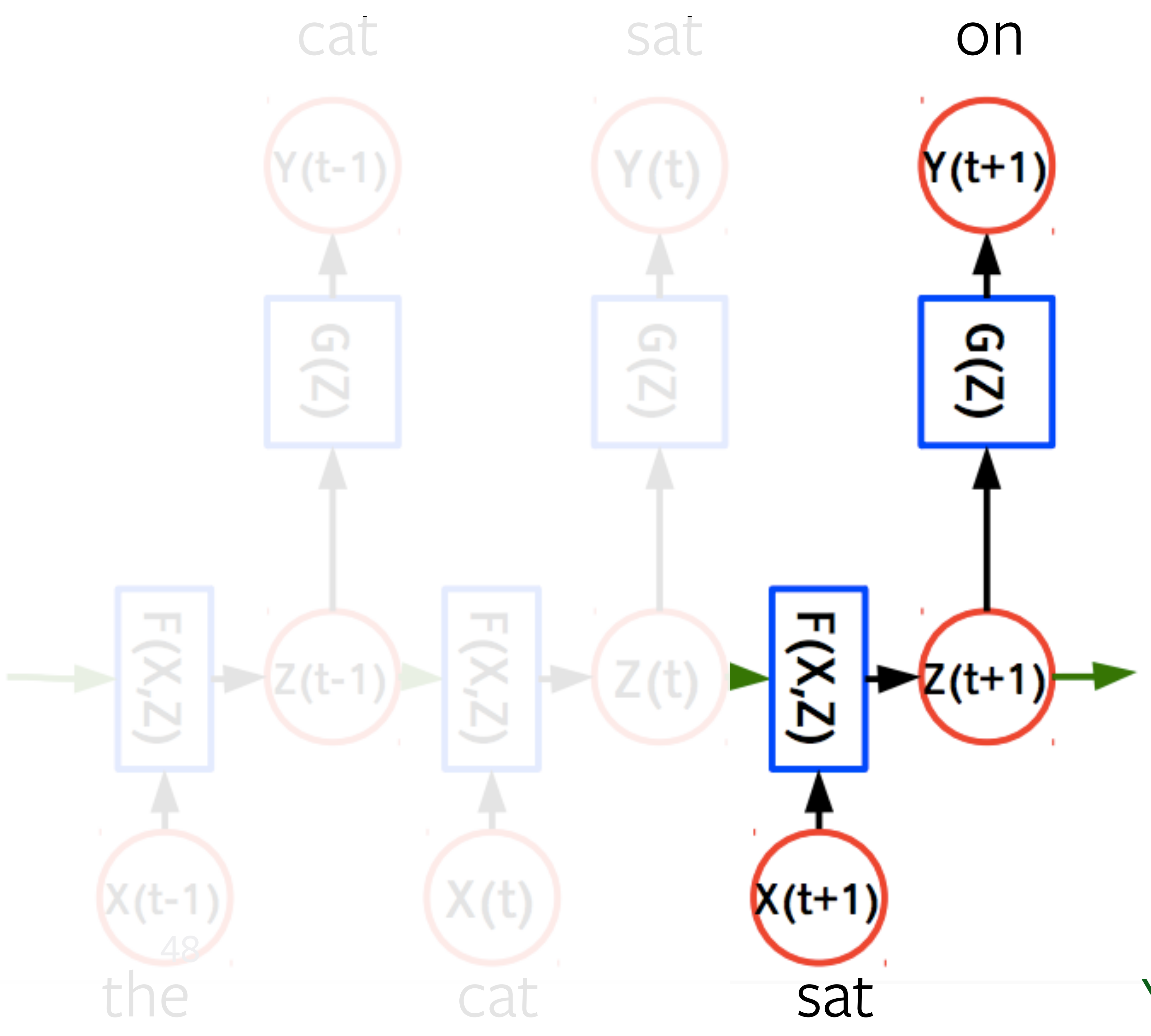
Y. LeCun's diagram

Source

Target

1) Represent source



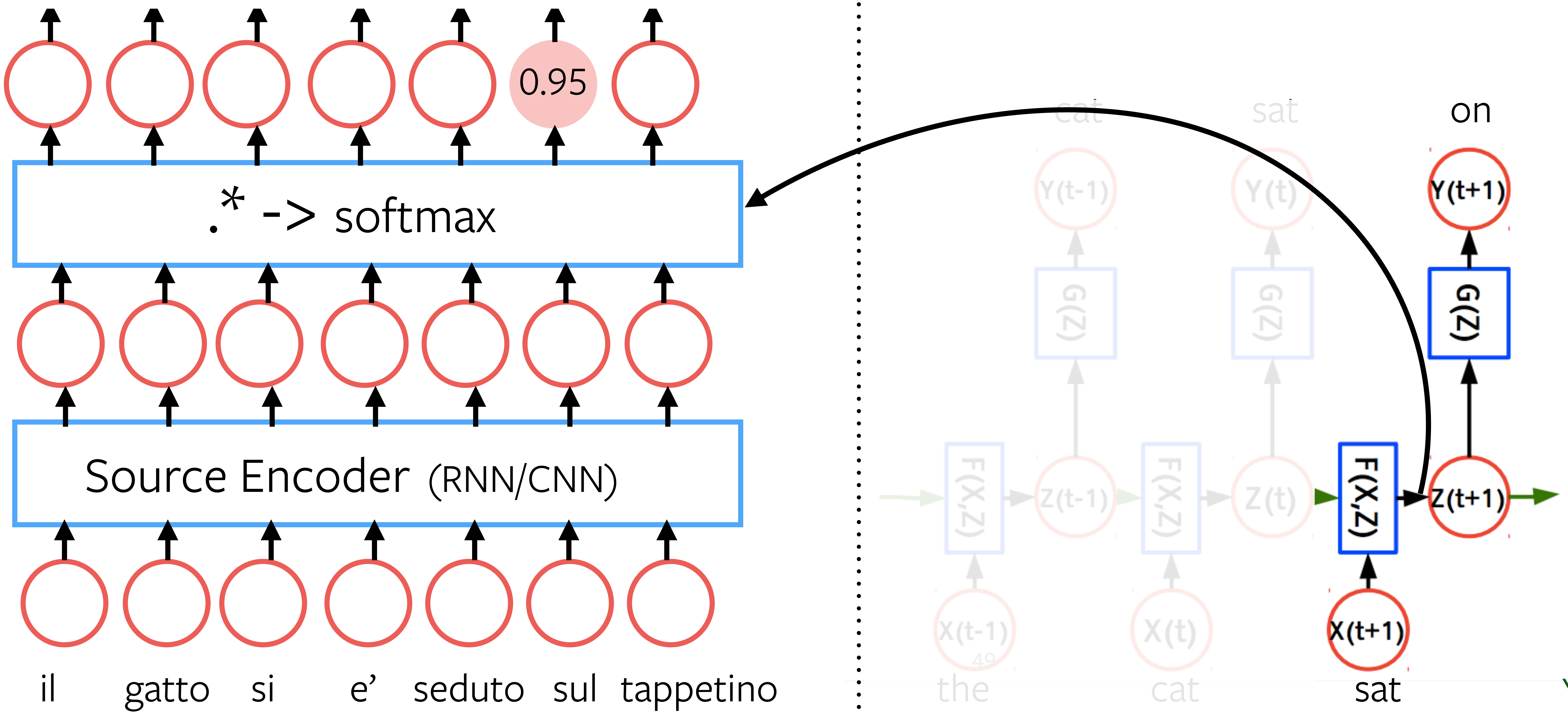


Y. LeCun’s diagram++

Source

Target

2) score each source word (attention)



Y. LeCun’s diagram++


Source
Target
3) combine target hidden with source vector
Sum
0.95
.* -> softmax
Source Encoder (RNN/CNN)
il
gatto
si
e'
seduto
sul
tappetino
cat
sat
on
Y(t-1)
Y(t)
Y(t+1)
G(Z)
G(Z)
G(Z)
F(X,Z)
F(X,Z)
F(X,Z)
Z(t-1)
Z(t)
Z(t+1)
X(t-1)
X(t)
X(t+1)
the
cat
sat
Y. LeCun's diagram++

Sequence -> Sequence: machine translation

Example:

ITA: Il gatto si e' seduto sul tappetino.

EN: The cat sat on the mat.

Notes:

+ source and target sentence can have any length, it works well on long sentences too!
+ it learns to align implicitly.
+ RNN can be replaced with CNNs. A convolutional encoder model for NMT, Gehring et al. 2016
+ it generates fluent sentences.
- It has trouble dealing with rare words, exact choice of words.
- It is typically trained like a language model (cross-entropy), good for scoring but not for generation.

Sequence -> Sequence: machine translation

WMT'16 English-Romanian	BLEU
Sennrich et al. (2016b) GRU (BPE 90K)	28.1
ConvS2S (Word 80K)	29.45
ConvS2S (BPE 40K)	29.88

WMT'14 English-German	BLEU
Luong et al. (2015) LSTM (Word 50K)	20.9
Kalchbrenner et al. (2016) ByteNet (Char)	23.75
Wu et al. (2016) GNMT (Word 80K)	23.12
Wu et al. (2016) GNMT (Word pieces)	24.61
ConvS2S (BPE 40K)	25.16

WMT'14 English-French	BLEU
Wu et al. (2016) GNMT (Word 80K)	37.90
Wu et al. (2016) GNMT (Word pieces)	38.95
Wu et al. (2016) GNMT (Word pieces) + RL	39.92
ConvS2S (BPE 40K)	40.46

Table 1. Accuracy on WMT tasks comapred to previous work. All results are averages over several runs.

	BLEU	Time (s)
GNMT GPU (K80)	31.20	3,028
GNMT CPU 88 cores	31.20	1,322
GNMT TPU	31.21	384
ConvS2S GPU (K40) $b = 1$	33.45	327
ConvS2S GPU (M40) $b = 1$	33.45	221
ConvS2S GPU (GTX-1080ti) $b = 1$	33.45	142
ConvS2S CPU 48 cores $b = 1$	33.45	142
ConvS2S GPU (K40) $b = 5$	34.10	587
ConvS2S CPU 48 cores $b = 5$	34.10	482
ConvS2S GPU (M40) $b = 5$	34.10	406
ConvS2S GPU (GTX-1080ti) $b = 5$	34.10	256

Table 3. CPU and GPU generation speed in seconds on the development set of WMT'14 English-French. We show results for different beam sizes b. GNMT figures are taken from Wu et al. (2016). CPU speeds are not directly comparable because Wu et al. (2016) use a 88 core machine compared to our 48 core setup.

Convolutional sequence to sequence learning, Gehring et al. arXiv 2017, https://github.com/facebookresearch/fairseq

Sequence -> Sequence: machine translation

Conclusions:

+ attention (gating) mechanism is rather general and it can be used for:
    + dealing with variable length inputs, as it "softly select one"
    + implicit alignment, which is discovered by the model as needed
    + to perform rounds of "reasoning" (e.g., "hops" in memory networks)
+ the same mechanism has been used to image captioning, summarization, etc.

- word level loss function (cross entropy for predicting the next word) is sub-optimal for the generation task.

Sequence level training with RNNs, Ranzato et al. ICLR 2016
An actor-critic algorithm for sequence prediction, ICLR 2017
Sequence-to-sequence learning as beam-search optimization, EMNLP 2016

Sequence -> Sequence: OCR

Example 1

Sequence -> Sequence: OCR

Example 2

Sequence -> Sequence: OCR

Example 2

"200"

Sequence -> Sequence: OCR

“200”

Challenges:

- digit segmentation is not observed; there can be several segmentations that are correct (i.e., yield correct transcription).
- variable length.
- design of loss function.
- very large number of valid output sequences.

Sequence -> Sequence: OCR

“200”

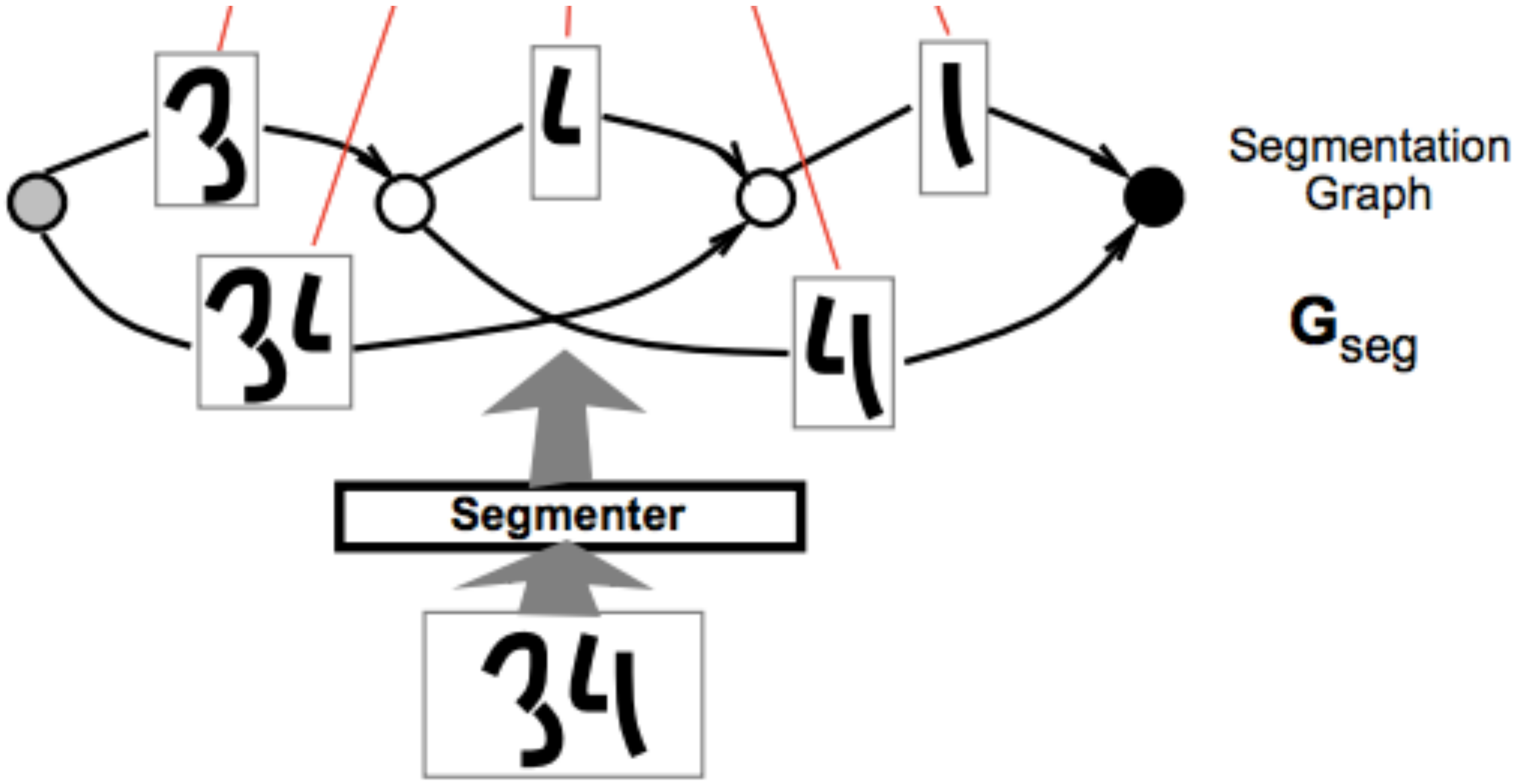
Approach:

- pre-train a CNN on single handwritten digits.
- over-segment and produce a lattice of possible “interpretations”.
- apply graph-transformer networks with a log-likelihood loss over sequences or margin loss.

Global training of document processing systems with graph transformer networks, Bottou et al. CVPR 1997
Gradient-based learning applied to document recognition, LeCun et al. IEEE 1998
Deep structured output learning for unconstrained text recognition, Jaderberg et al. ICLR 2015

Sequence -> Sequence: OCR

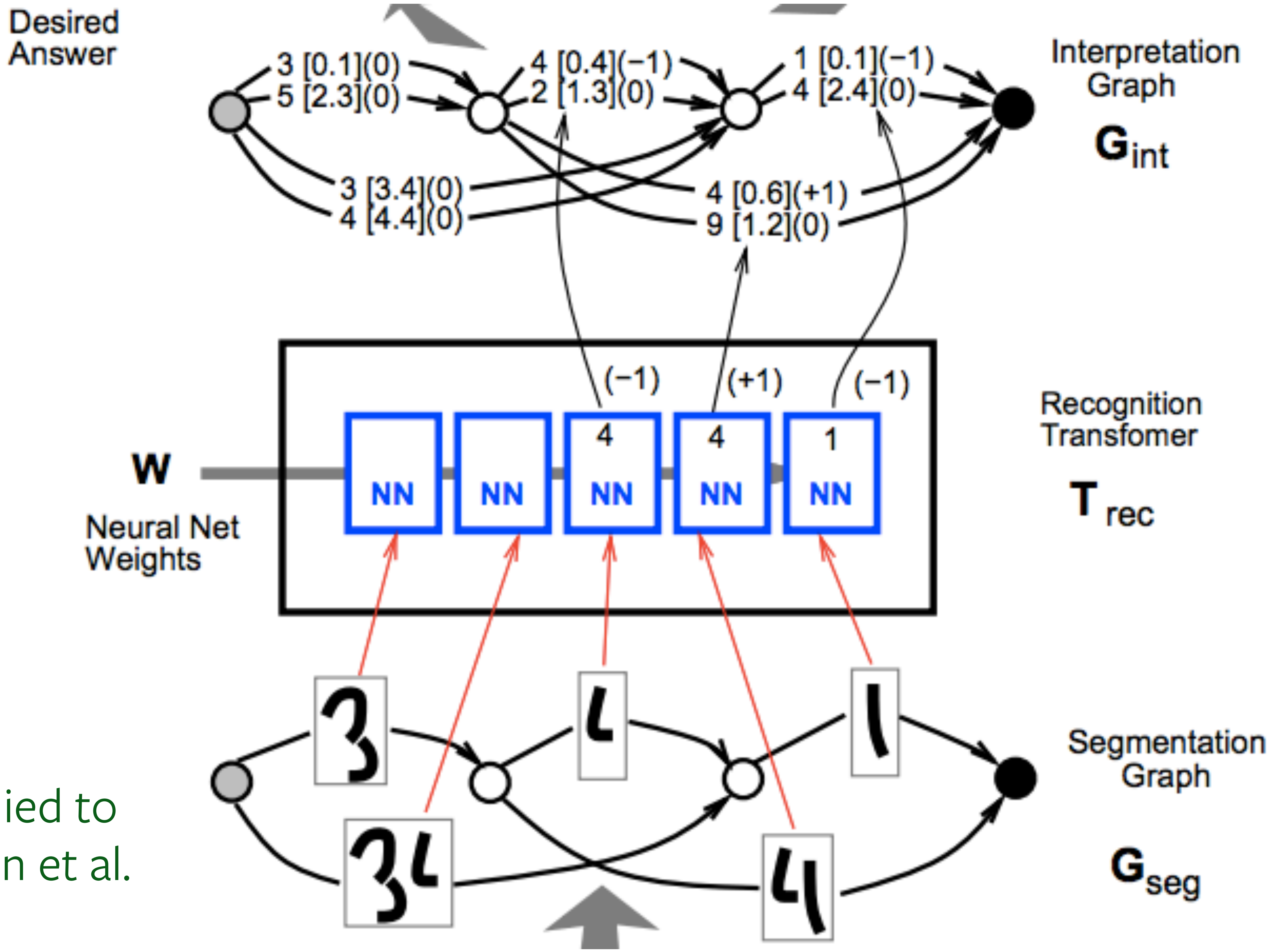
Step1: over-segment & produce lattice of interpretations



Gradient-based learning applied to document recognition, LeCun et al. IEEE 1998

Sequence -> Sequence: OCR

Step2: score each hypothesis

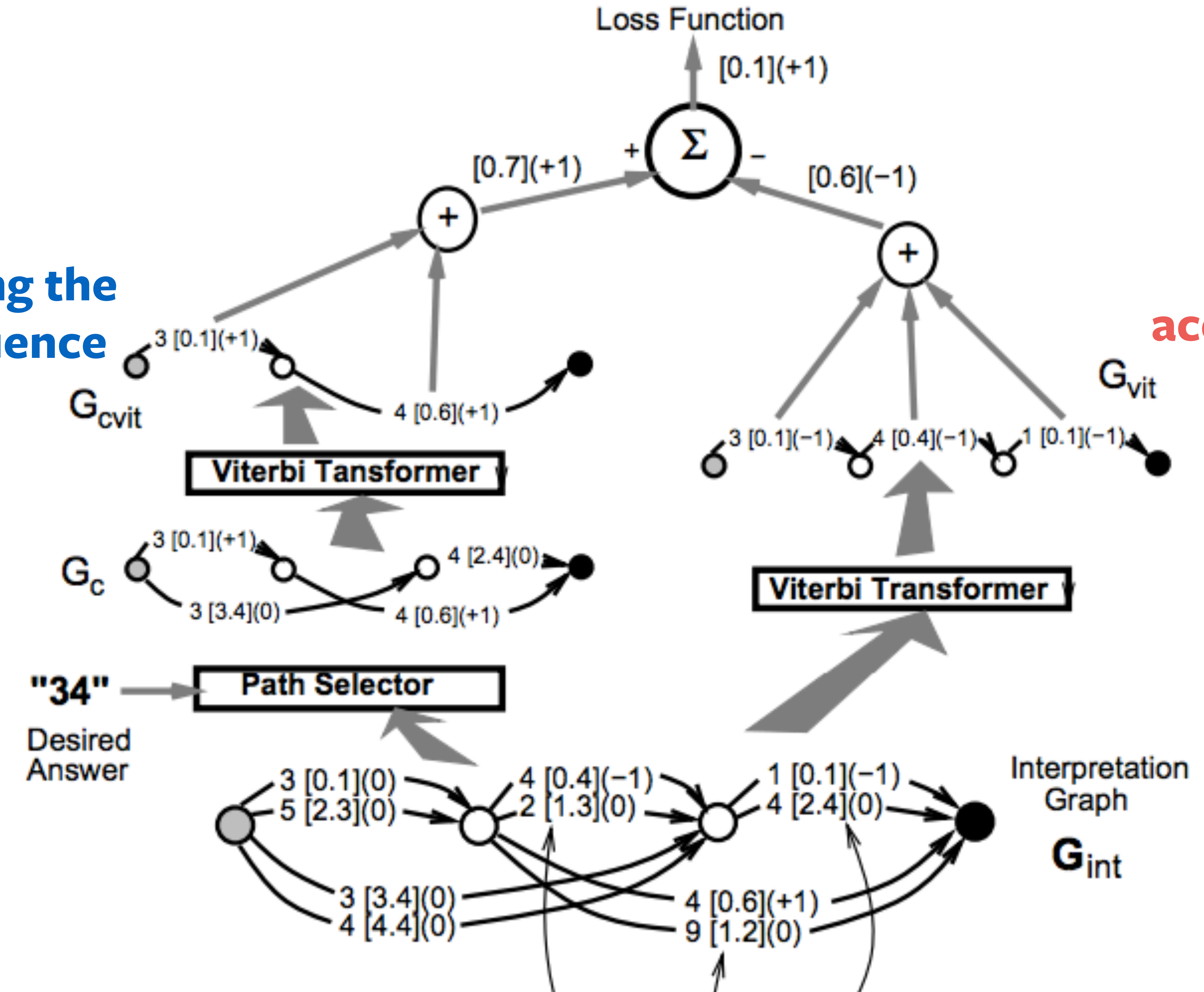


Gradient-based learning applied to document recognition, LeCun et al. IEEE 1998

Sequence -> Sequence: OCR

Step3: compute loss and gradients

Find all paths yielding the correct output sequence



Find the best path according to the model

Sequence -> Sequence: OCR

Conclusions:

- problem may have latent variables (segmentation), over which one can minimize or marginalize over.
- structure prediction is well expressed in terms of weighted lattices, and bprop still applies (GTN).
- loss functions and EBMs can straightforwardly be extended to handle sequences. This is one of the best examples of training at the sequence level.
- search over best hypothesis of the system can be expensive; marginalization can be intractable. It’s problem and model dependent.

Conclusions

- sequences can appear at the input, output, or both.
- structured outputs are the most difficult case, overall when there may be several plausible predictions for the same input (e.g., MT, image captioning).
- sometimes, we do not need to bother taking into account the sequential aspect of the data, if the prediction task is well correlated to variables present in static input.
- it's possible to learn to generate sequences, to search in the space of sequences, and to still train by back-propagation as in GTNs.
- ultimately, there is no general model/loss that work in all cases. They should be designed for the task at hand.
- there are lots of demos and code available to reproduce these examples. See pytorch and torch tutorials, for instance.

Questions?

Thank you!

Acknowledgements

I would like to thank Armand Joulin for sharing material about FastText.