Deep Learning architectures for NLP

G. Shang, A. Tixier, M. Vazirgiannis

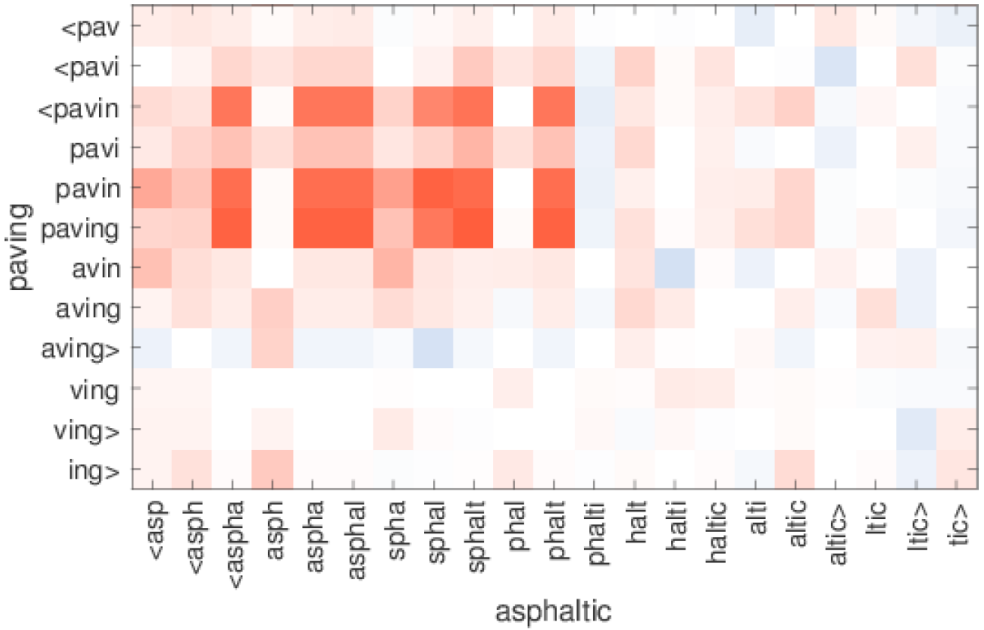
LIX, Ecole Polytechnique

December 2019

- Fast Text Embeddings – subword information

Enriching Word Vectors with Subword Information

FAIR 2017



X'11-MVA'12

Piotr Bojanowski

Facebook AI Research
Verified email at fb.com - Homepage

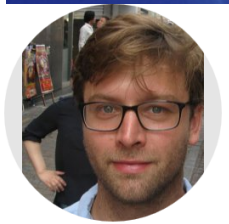
Computer Vision Machine Learning

X'9-MVA'10

Edouard Grave

Research Scientist

X'8-MVA'9



Armand Joulin

Research scientist at Facebook
Verified email at fb.com - Homepage

Artificial Intelligence Machine Learning

Tomas Mikolov

Research scientist, Facebook
Verified email at fb.com

Artificial Intelligence Machine Learning

Enriching Word Vectors with Subword Information

Simple problem: word2vec/glove etc. ignore the internal structure of words

E.g., knowledge about *luck* is not used when learning a representation for *unlucky or luckily*

=> parameters are not shared => difficult to learn good vectors for rare words, and impossible for out-of-vocabulary words

Simple solution: learn vectors for character n-grams. Compose word vectors from their n-gram vectors.

Enriching Word Vectors with Subword Information

Quick recap: in skip-gram (Mikolov et al. 2013), the objective is to maximize:

$$\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} \log p(w_c \mid w_t)$$

$w_1, \ldots, w_T$: the training corpus

C_t : set of indices of the context words around w_t

In English: the objective is to *predict well the context of a word given this word*

$p(w_c \mid w_t)$ is parameterized by the word vectors through a scoring function ***s***

$$s(w_t, w_c) = \mathbf{u}_{w_t}^{\top} \mathbf{v}_{w_c}$$

u and **v** above are taken from the input and output embedding matrices, resp.

Enriching Word Vectors with Subword Information

Proposed approach:

Each word is represented as a bag of **character n-grams**. E.g., for the word *where* and n=3:

<wh, whe, her, ere, re>

The < and > characters are added at the beginning and end of the word to keep prefix/suffix information.

New scoring function: $$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c$$

G_w is the set of character n-grams in word w. z_g is the vector of the g^{th} n-gram
V_c is the vector of the context word c
=> w is represented as the sum of its n-gram vectors

Enriching Word Vectors with Subword Information

Quantitative results: word similarity and word analogy tasks in 7 languages

- similarity: better than original skipgram and CBOW on 6/7 datasets
- analogy: - improves on original skipgram and CBOW for syntactic tasks
  - no improvement for semantic tasks

Qualitative results:

query	tiling	tech-rich	english-born	micromanaging	eateries	dendritic
sisg	tile flooring	tech-dominated tech-heavy	british-born polish-born	micromanage micromanaged	restaurants eaterie	dendrite dendrites
sg	bookcases built-ins	technology-heavy .ixic	most-capped ex-scotland	defang internalise	restaurants delis	epithelial p53

Nearest neighbors of rare words using subword (sisg) and original (sg) skipgram

Deep Contextualized Word Representations - ELMO

Deep Contextualized Word Representations

Best paper NAACL 2018

transfer learning has been used in vision since 2012 (ImageNet)
it is used in NLP since 2018! [1] [2]

ELMo: Embeddings from Language Models

Initial problem: traditional word vectors map each word to a single, context-independent vector

But some words have more than one sense! (polysemy)

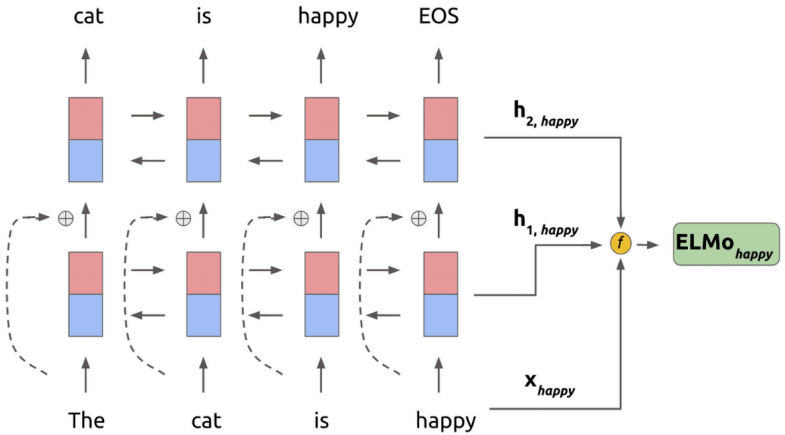
e.g., bank, get, wood, play, mean…

Solution:

- token assigned a representation based on entire input sentence
- use the internal representations of a RNN language model pre-trained on a large dataset

Deep Contextualized Word Representations

- bidirectional LSTM is trained with a coupled language model (LM) on a large text corpus - ELMo (Embeddings from Language Models) representations.

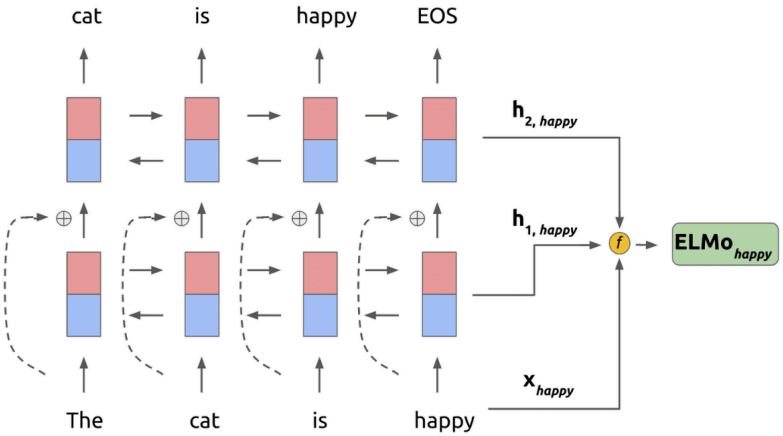


  - Forward $p(t_1, t_2..., t_N) = \prod_{k=1}^{N} p(t_k|t_1, t_2..., t_N)$
  - Bacward $p(t_1, t_2..., t_N) = \prod_{k=1}^{N} p(t_k|t_{k+1}, t_{k+2}..., t_{k+N})$

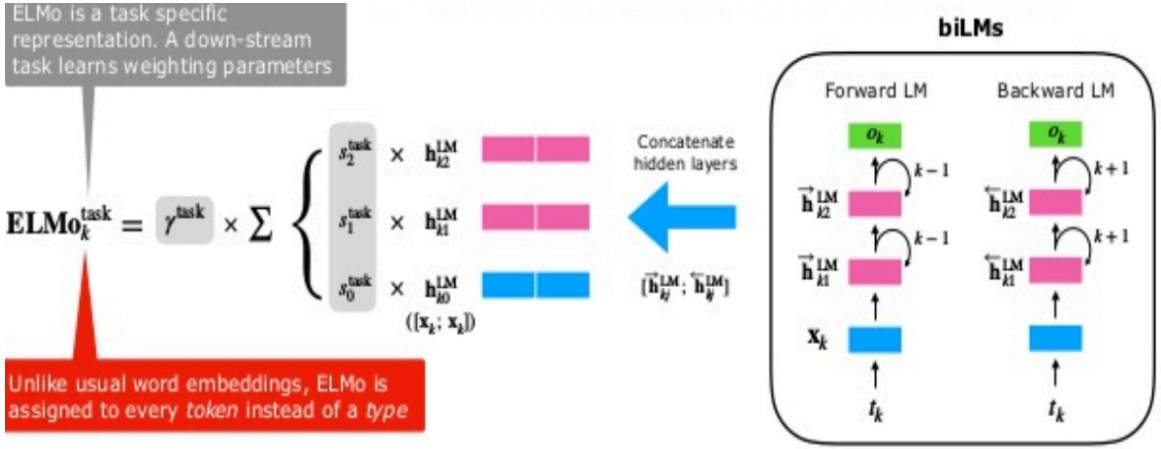
Graphic from: https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/

Deep Contextualized Word Representations

- ELMo representations are deep: function of all of the internal layers of the biLM.
  - learn a linear combination of the vectors stacked above each input word for each end task,
  - Combining the internal states in this manner allows for very rich word representations.
  - higher-level LSTM states ´context-dependent aspects of word meaning (e.g., they can be used without modification to perform well on supervised word sense disambiguation tasks)
  - lower level states model aspects of syntax (e.g., they can be used to do part-of-speech tagging).
  - Simultaneously exposing all of these signals is highly beneficial, allowing the learned models select the types of semi-supervision that are most useful for each end task.



Deep Contextualized Word Representations



Semi-supervised approach:

- 1) a deep bidirectional RNN language model is pretrained on a large dataset
- 2) the vector of each word in a given input sentence is computed as a weighted sum of the RNN hidden states
- 1) is *unsupervised*, 2) weights are learned in a *supervised* way on some task-specific dataset

Graphic from: https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/

$$\mathbf{ELMo}_k^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}$$

$h_{k,j}^{LM}$ is the k[th] hidden representation of the j[th] layer of the bi-RNN LM

s^{task} is the softmax weight vector, γ^{task} is a scaling parameter (for optimization)

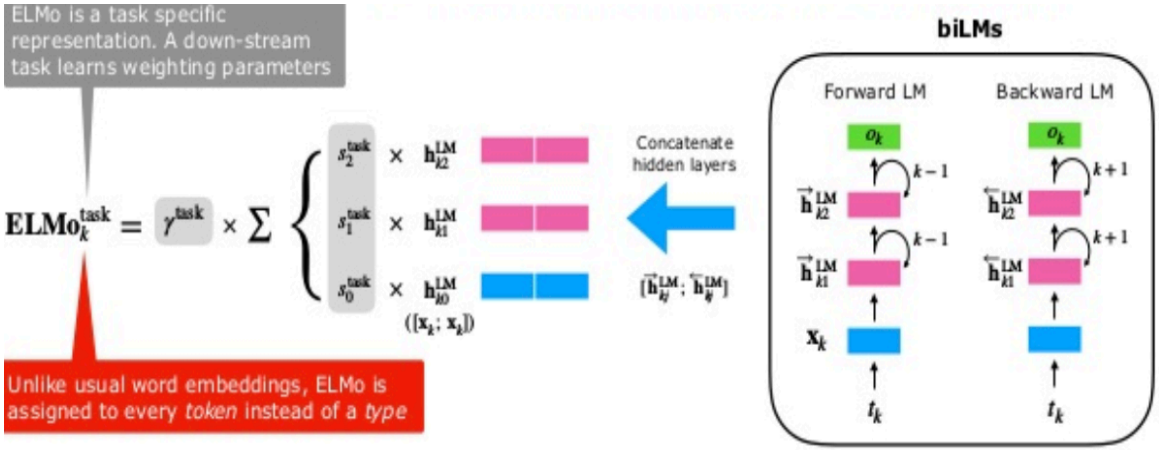
The authors use L=2 in all experiments

Deep Contextualized Word Representations

Use of ELMo in practice, on a task-specific dataset:

- 1) use the pretrained language model in prediction mode and store $h_{k,j}^{LM}$ for each word (each k) and each layer (each j)
- 2) concatenate $ELMo_k^{task}$ with the corresponding input vector* of whatever supervised model is used to solve the task (e.g., RNN, CNN, feed-forward…)
- 3) update s^{task} and γ^{task} with the other parameters of the supervised model during training

*given by word2vec, glove, etc.



Results:

ELMo improves over the baselines on 8 tasks, ranging from question answering to co-reference resolution, sentiment analysis, POS-tagging, and disambiguation

Deep Contextualized Word Representations

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Table 1: Test set comparison of ELMo enhanced neural models with state-of-the-art single model baselines across six benchmark NLP tasks. The performance metric varies across tasks – accuracy for SNLI and SST-5; F_1 for SQuAD, SRL and NER; average F_1 for Coref. Due to the small test sizes for NER and SST-5, we report the mean and standard deviation across five runs with different random seeds. The "increase" column lists both the absolute and relative improvements over our baseline.

SQuAD: Question answering, SNLI: Entailment, SRL: Semantic Role Labelling ("Who did what to whom"), NER: Name entity recognition, SST: Sentiment Analysis, Coref: task of clustering mentions in text that refer to the same underlying real world entities

Deep Contextualized Word Representations

Model	F_1
WordNet 1st Sense Baseline	65.9
Raganato et al. (2017a)	69.9
Iacobacci et al. (2016)	**70.1**
CoVe, First Layer	59.4
CoVe, Second Layer	*64.7*
biLM, First layer	67.4
biLM, Second layer	*69.0*

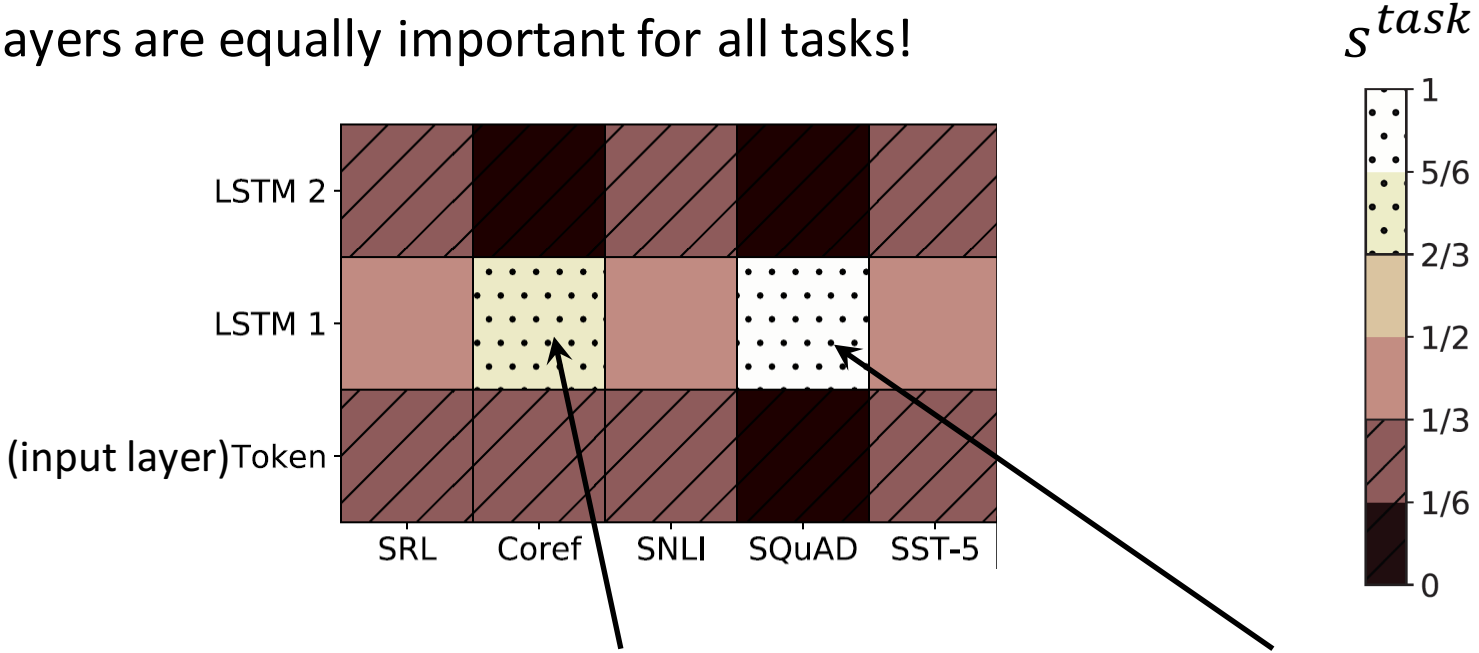
WSD

Model	Acc.
Collobert et al. (2011)	97.3
Ma and Hovy (2016)	97.6
Ling et al. (2015)	**97.8**
CoVe, First Layer	*93.3*
CoVe, Second Layer	92.8
biLM, First Layer	*97.3*
biLM, Second Layer	96.8

POS tagging

Deep Contextualized Word Representations

Not all layers are equally important for all tasks!



The 1st layer clearly dominates for co-reference resolution and question answering

For the other tasks, the weights are more evenly distributed among layers

ATTENTION BASED ARCHITECTURES

Attention is ubiquitous in DL today

Image captioning

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

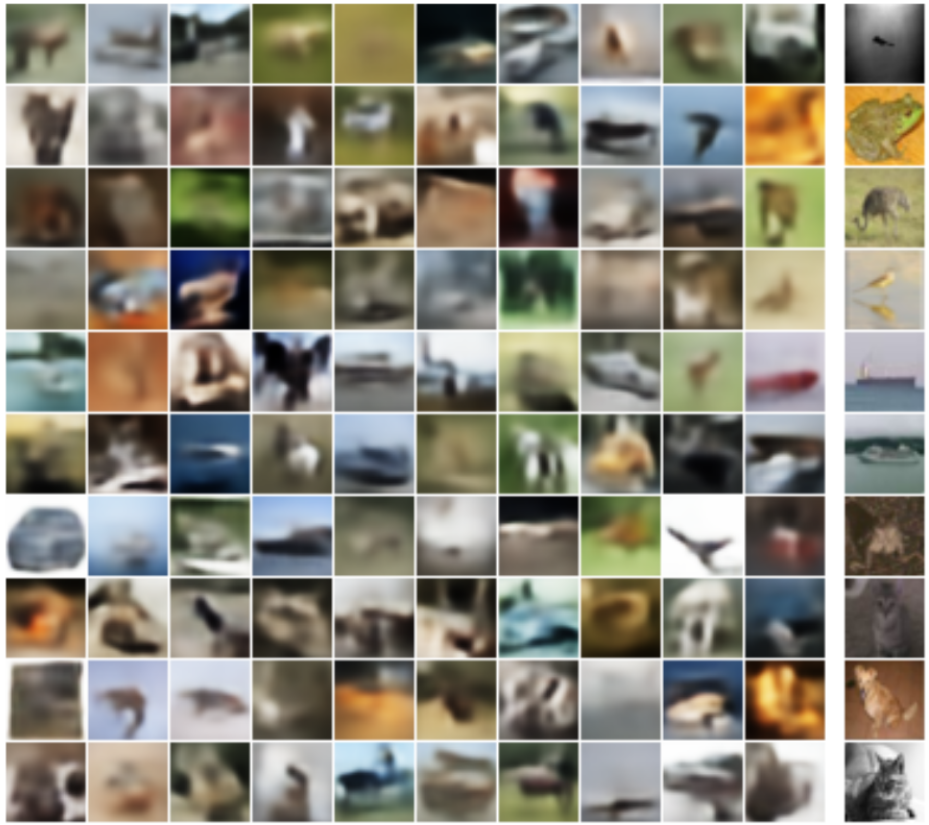
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Show, attend and tell: Neural image caption generation with visual attention (Xu et al. 2015)

Attention is ubiquitous in DL today

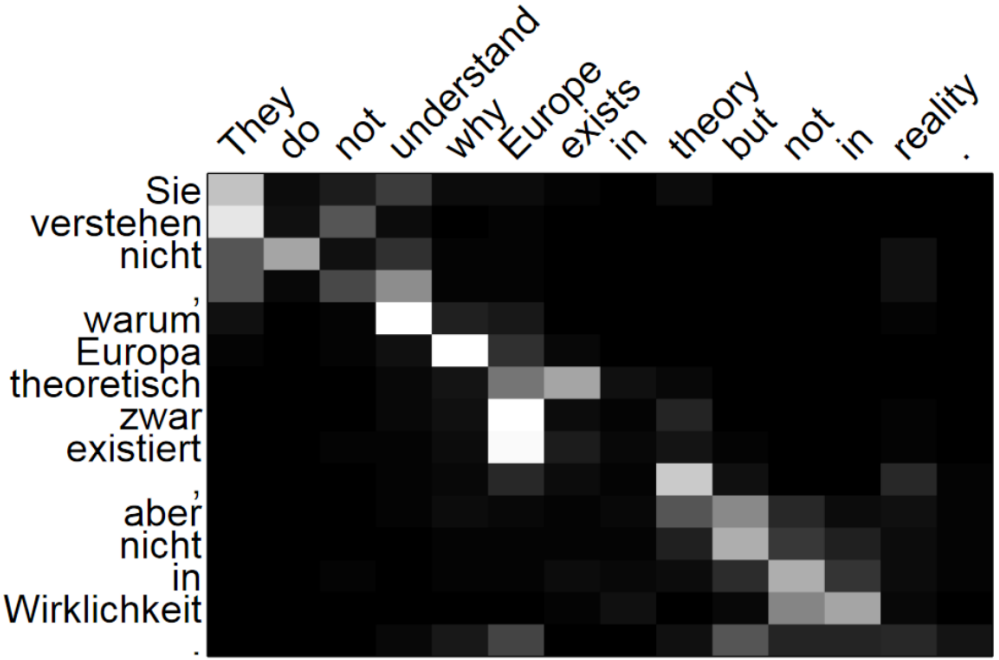
Image generation



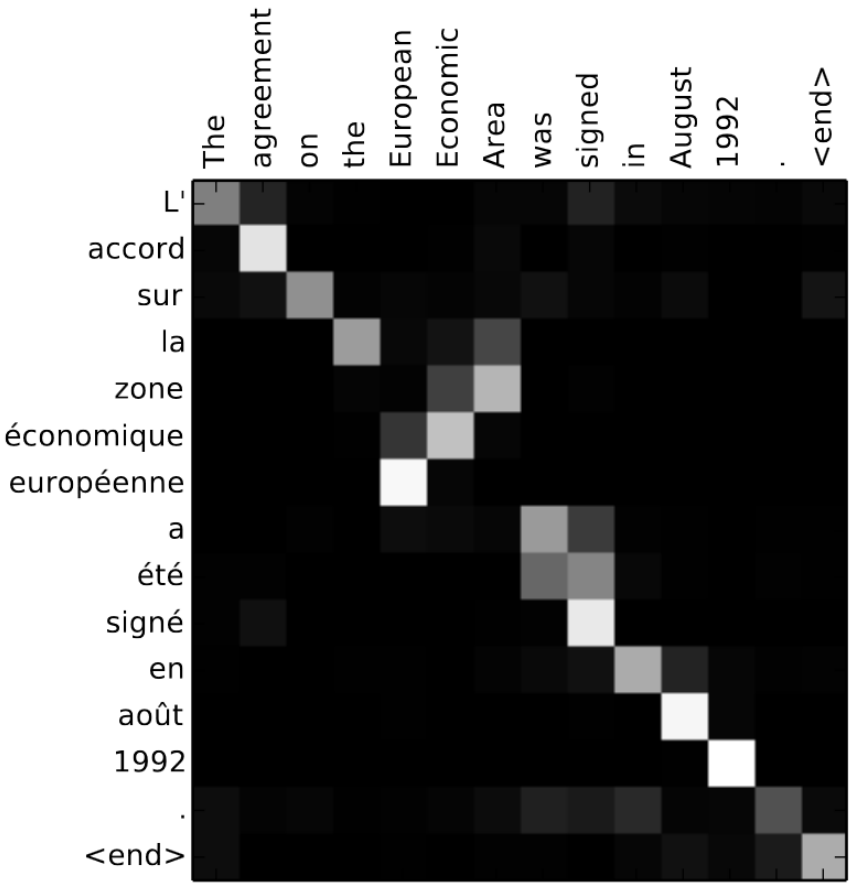
DRAW: A recurrent neural network for image generation (Gregor et al. 2015)

Attention is ubiquitous in DL today

Neural Machine Translation



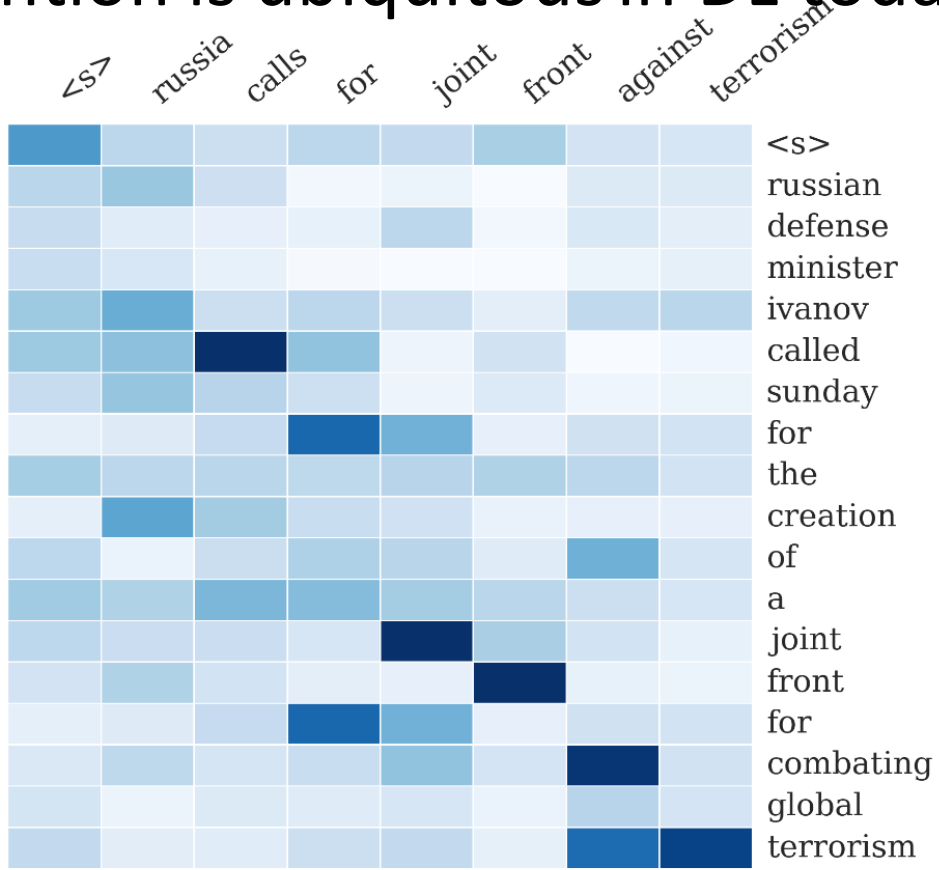
Effective approaches to attention-based neural machine translation (Luong et al. 2015)



Neural machine translation by jointly learning to align and translate (Bahdanau et al. 2014)

Attention is ubiquitous in DL today

Abstractive summarization



A neural attention model for abstractive sentence summarization (Rush et al. 2015)

Attention is ubiquitous in DL today

Sentiment analysis

GT: 4 Prediction: 4

pork belly = delicious .
scallops ?
i do n't .
even .
like .
scallops , and these were a-m-a-z-i-n-g .
fun and tasty cocktails .
next time i 'm in phoenix , i will go
back here .
highly recommend .

GT: 0 Prediction: 0

terrible value .
ordered pasta entree .
.
$ 16.95 good taste but size was
appetizer size .
.
no salad , no bread no vegetable .
this was .
our and tasty cocktails .
our second visit .
i will not go back .

Hierarchical attention networks for document classification (Yang et al. 2016)

Encoder-Decoder Architectures

General idea:

encoder **decoder**

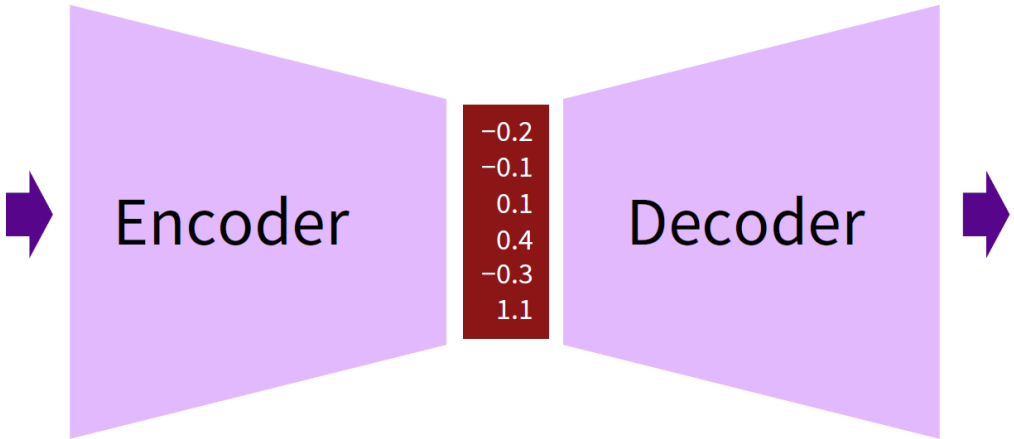
Input -> embedding -> Output

- image
- document
- source language
- …

- caption
- summary
- target language
- …

Known as *sequence-to-sequence* (seq2seq) when input and output are sequences (e.g., NLP applications)

The full architecture is differentiable => end-to-end training



What is attention?

Objective

- Traditional models (e.g., encoder) having to embed the input into a single fixed-length vector (lossy).
- Alternatively information can be kept and stored into multiple vectors.
- information can be retrieved later on (e.g., by the decoder). (Bahdanau et al. 2014)

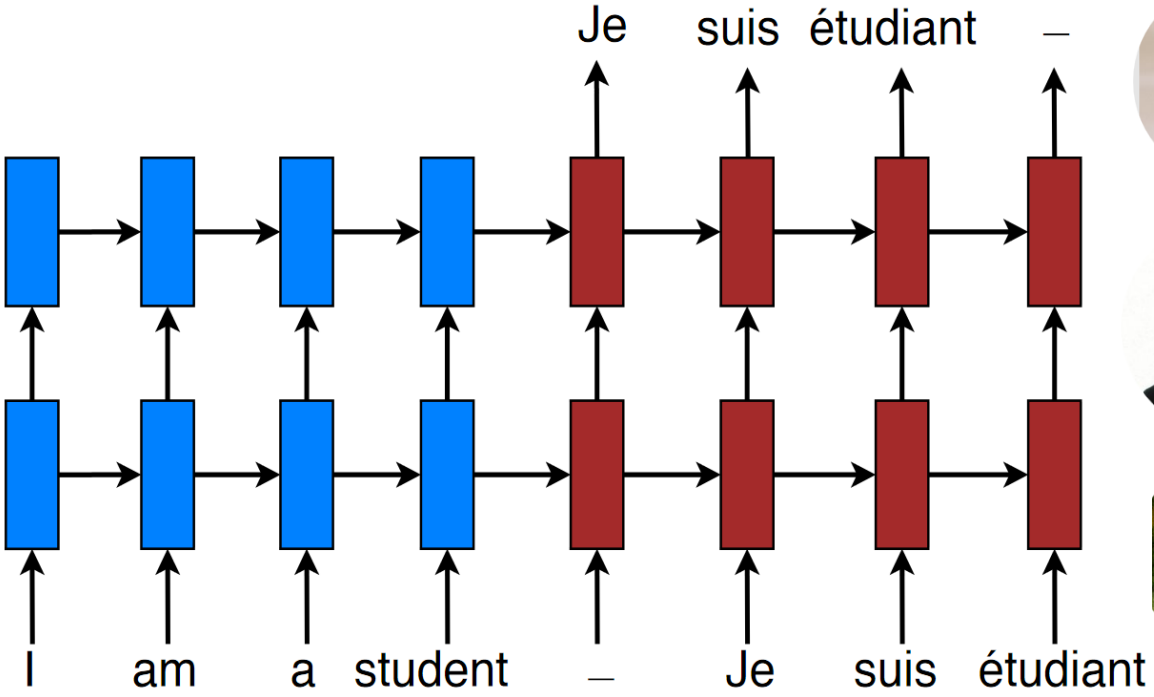
Quick history:

- developed in the context of **encoder-decoder** architectures for neural machine translation (Bahdanau et al. 2014)
- rapidly applied to naturally related tasks like image captioning (Xu et al. 2015) and summarization (*Luong et al. 2015*)
- also proposed for encoders only, e.g. for text classification (Yang et al. 2015) and representation learning (Conneau et al. 2017).
  Known as *self or inner attention* in such cases.

Encoder-Decoder for Neural Machine Translation

Encoder **Decoder**

Target *sentence (generated)*



Source *sentence (input)*

Minh-Thang Luong

Research Scientist at Google
Verified email at google.com - Homepage
Deep Learning Natural Language Processing

Hieu Pham

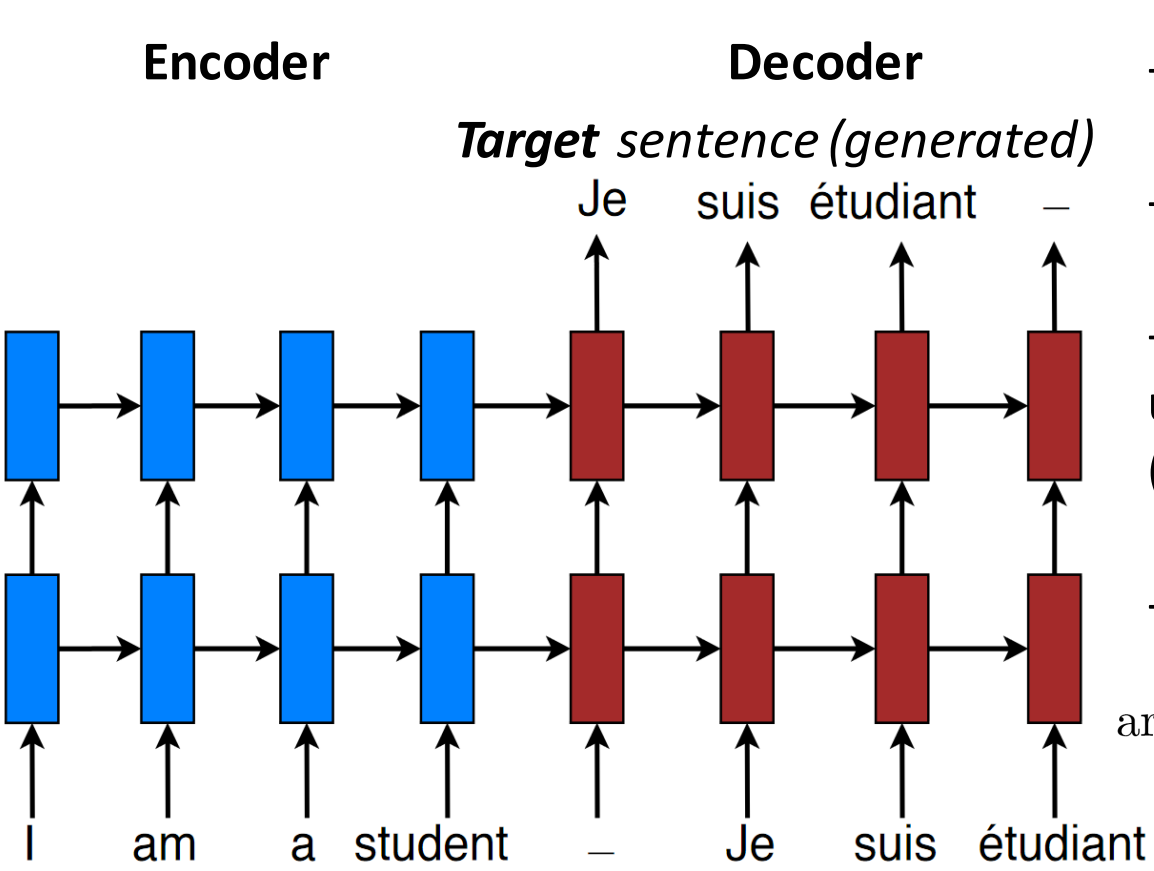
Carnegie Mellon University, Google Brain
Verified email at google.com
Machine Learning

Christopher D Manning

Professor of Computer Science and Linguistics
Verified email at stanford.edu - Homepage
Natural Language Processing

Effective approaches to Attention-Based Neural Machine Translation (2015)

Encoder-Decoder for Neural Machine Translation



- source: $\left(x_1, \ldots, x_{T_x}\right)$

- target: $\left(y_1, \ldots, y_{T_y}\right)$

- Both encoder & decoder are unidirectional deep RNNs (a.k.a. *stacking* RNNs)

- Training objective:

$$\text{argmax}_\theta \left\{ \sum_{(x,y)\in \text{corpus}} \log\, p(y|x;\theta) \right\}$$

Luong et al. (2015)

Encoder-Decoder for Neural Machine Translation

Encoder: usually: CNN, stacking RNN* with LSTM or GRU units…

* unidirectional (Luong et al. 2015) or bidirectional (Bahdanau et al. 2014).

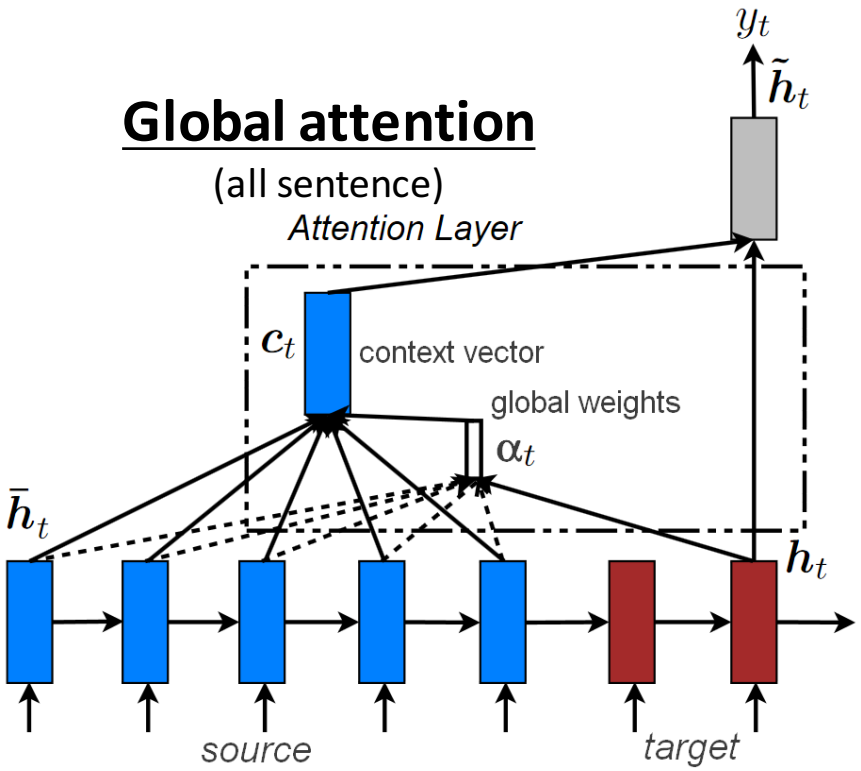
Decoder: unidirectional *RNN* (well suited to text generation), best if deep.

Generates the target sentence $\left(y_1, \cdots, y_{T_y}\right)$ one word at a time:

$$P\left[y_t | \{y_1, \ldots, y_{t-1}\}, c_t\right] = \text{softmax}\left(W_s \tilde{h}_t\right)$$

Luong et al. (2015)

Encoder-**Decoder** for Neural Machine Translation



$$P\left[y_t|\{y_1,...,y_{t-1}\},c_t\right] = \text{softmax}\left(W_s\tilde{h}_t\right)$$

attentional hidden state

$$\tilde{h}_t = \tanh\left(W_c\left[c_t;h_t\right]\right)$$

context vector

decoder hidden state

$$c_t = \sum_{i=1}^{T_x} \alpha_{t,i}\bar{h}_i$$

i^{th} encoder hidden state

alignment vector

$$\alpha_{t,i} = \frac{\exp\left(\text{score}\left(h_t,\bar{h}_i\right)\right)}{\sum_{i'=1}^{T_x}\exp\left(\text{score}\left(h_t,\bar{h}_{i'}\right)\right)}$$

$$\text{score}\left(h_t,\bar{h}_i\right) = h_t^\top \bar{h}_i$$

Luong et al. (2015)

Encoder-**Decoder** for Neural Machine Translation

$$P\left[y_t|\{y_1,...,y_{t-1}\},c_t\right] = \text{softmax}\left(W_s\tilde{h}_t\right)$$

attentional hidden state

$$\tilde{h}_t = \tanh\left(W_c\left[c_t;h_t\right]\right)$$

decoder hidden state

$$p_t = T_x \cdot \sigma\left(v_p^\top \tanh\left(W_p h_t\right)\right)$$

context vector

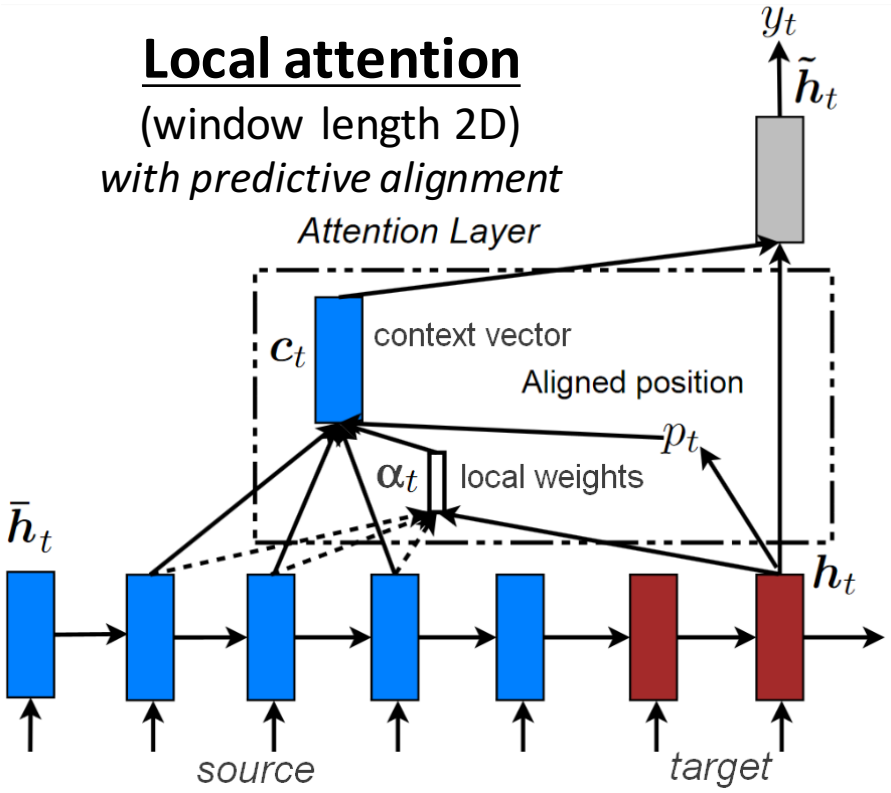
ith encoder hidden state

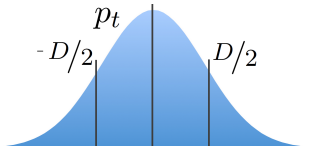
$$c_t = \sum_{i=p_t-D}^{p_t+D} \alpha_{t,i}\bar{h}_i$$

alignment vector

$$\alpha_{t,i} = \frac{\exp\left(\text{score}(h_t,\bar{h}_i)\right)}{\sum_{i'=p_t-D}^{p_t+D}\exp\left(\text{score}(h_t,\bar{h}_{i'})\right)}\exp\left(-\frac{(i-p_t)^2}{2(D/2)^2}\right)$$

$$\text{score}(h_t,\bar{h}_i) = h_t^\top W_\alpha \bar{h}_i$$





Luong et al. (2015)

Encoder-Decoder for Neural Machine Translation

Facts:

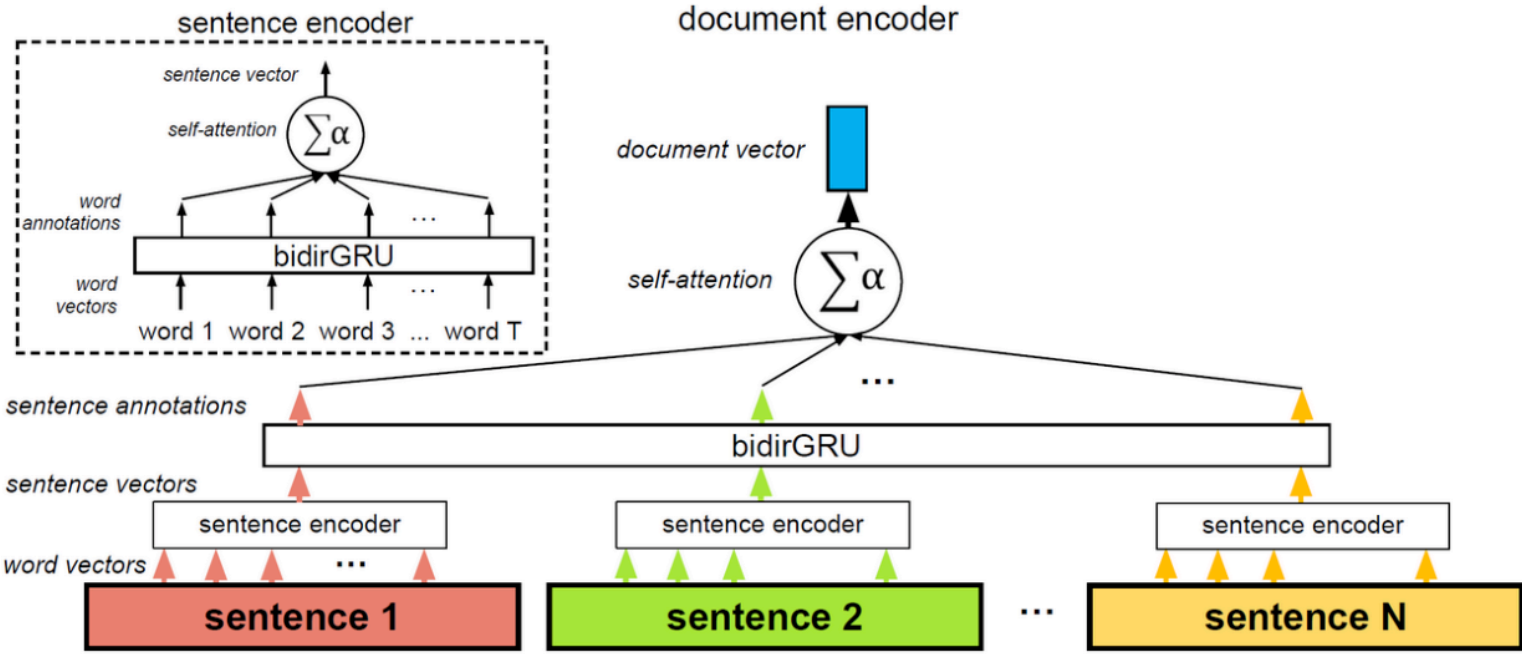
- Tested on the English <-> German task (WMT'14 dataset)
- 4.5M sentence pairs
- Encoder and decoder RNNs feature 4 layers of stacking and 1000-dimensional hidden states
- Window of size *D=10* for local attention
- Trained for 12 epochs (total of 7-10 days on a single GPU, at 1K target words/s)
- New state-of-the-art performance

Lessons learned:

- Local attention with predictive alignment gives better results than global attention
- Dot product ($\text{score}(h_t, \bar{h}_i) = h_t^\top \bar{h}_i$) works well for global attention
- The general formulation ($\text{score}(h_t, \bar{h}_i) = h_t^\top W_\alpha \bar{h}_i$) is better for local attention

Luong et al. (2015)

Self-attention for RNN encoders



Self-attention for RNN encoders

- Input is a sentence $(x_1, \ldots, x_T)$
- We're only interested in getting an embedding s of the sentence for some downstream task (e.g., classification)

encoder hidden state

context vector

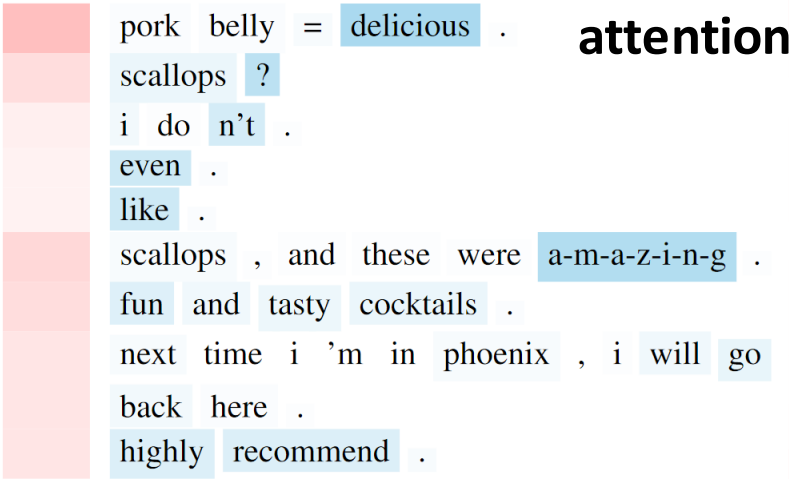
$$u_t = \tanh(W h_t)$$

$$\alpha_t = \frac{\exp(\text{score}(u_t, u))}{\sum_{t'=1}^{T} \exp(\text{score}(u_{t'}, u))}$$

$$s = \sum_{t=1}^{T} \alpha_t h_t$$

Where $\text{score}(u_t, u) = u_t^\top u$

The same process can be repeated over the sentence vectors s -> **hierarchical attention**



Attention is All You Need - Tranformer

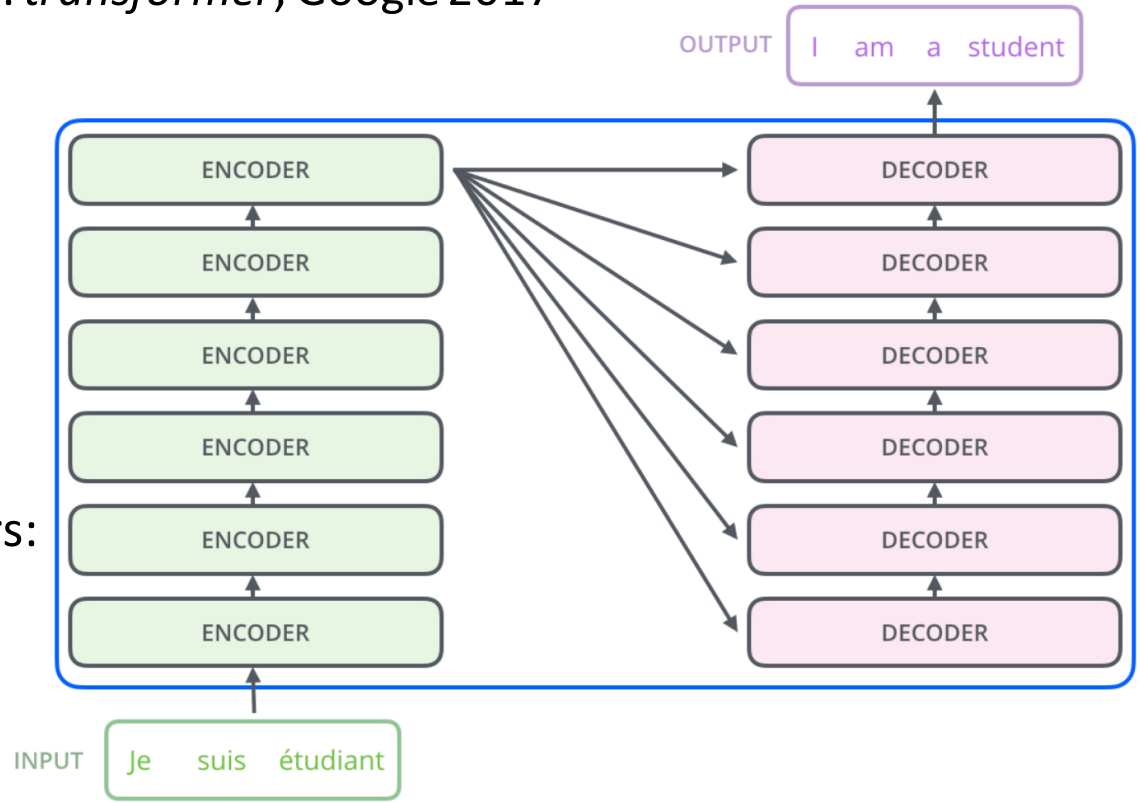
Attention is All You Need

a.k.a. *transformer*, Google 2017

Transformer is a seq2seq model that does not use recurrence!

It only uses:

- dense layers
- two types of attention layers: self-attention and encoder-decoder attention



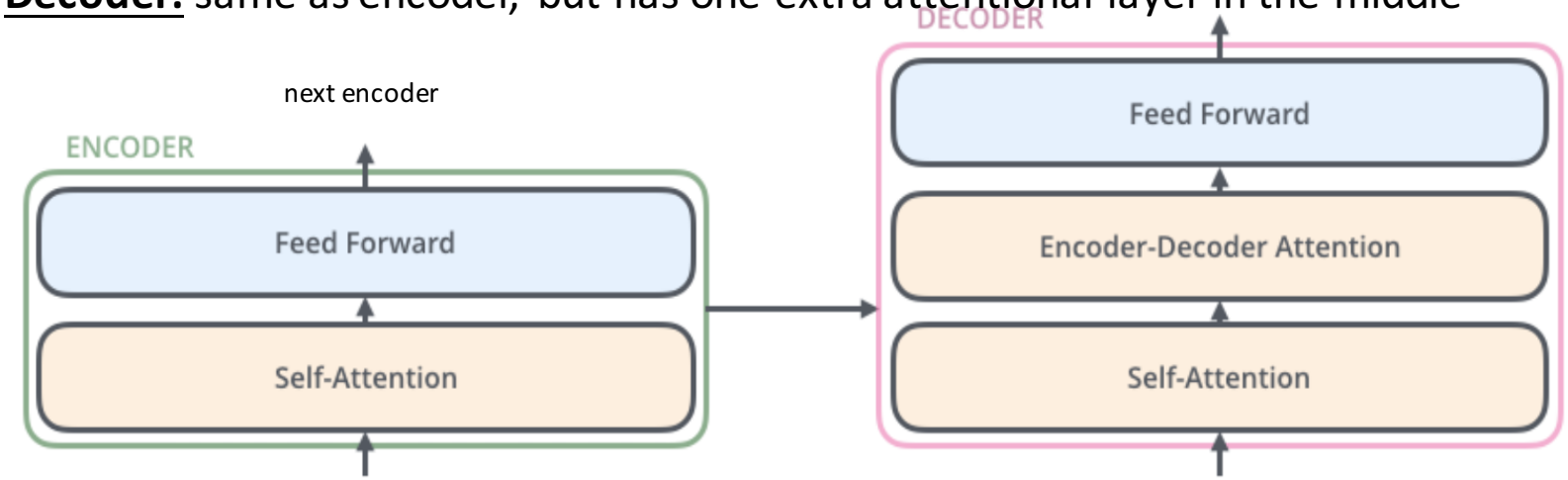
Figures in this section are taken from this excellent blogpost

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

Attention is All You Need

Encoder: (1) each word* of the input sequence is encoded 'in-context' by considering other words in the sequence via self-attention; (2) the output for each word is then passed through the same dense (FF) layer in parallel.

Decoder: same as encoder, but has one extra attentional layer in the middle



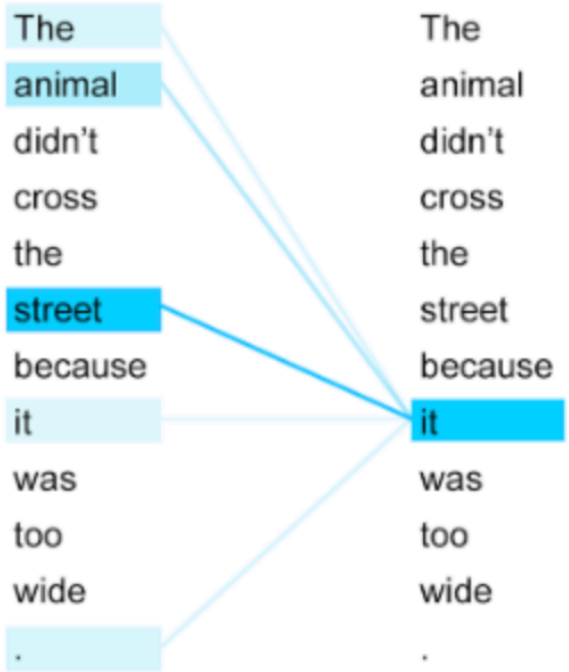
*each word is represented by its embedding + positional vector (both trainable). Positional vectors give the model a sense of word order

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

Attention is all you need - Transformer

- **Self Attention**
- *…, sometimes called intra-attention, is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence.”*

- https://arxiv.org/abs/1706.03762



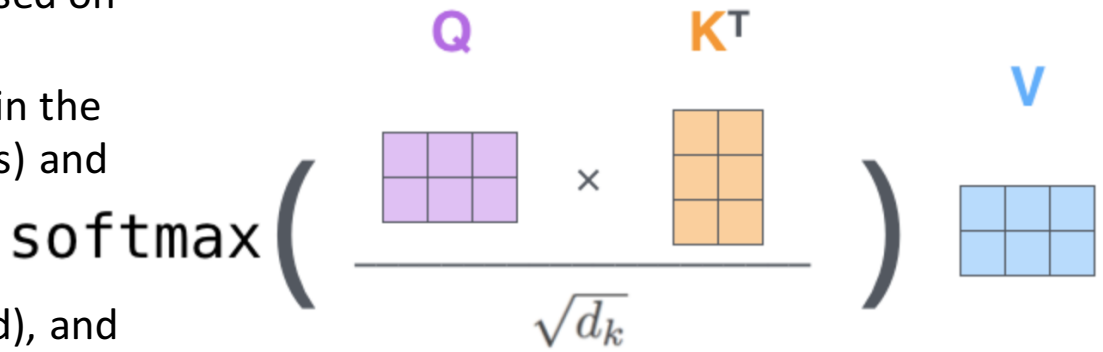
http://jalammar.github.io/illustrated-transformer/

Attention is All You Need

- General formulation for attention, based on query, key, and value matrices: Q,K,V.
  In these matrices, rows are positions in the input sequence (word representations) and columns are features.

- Q and K are first compared (multiplied), and the output of that comparison is normalized into a weight matrix.

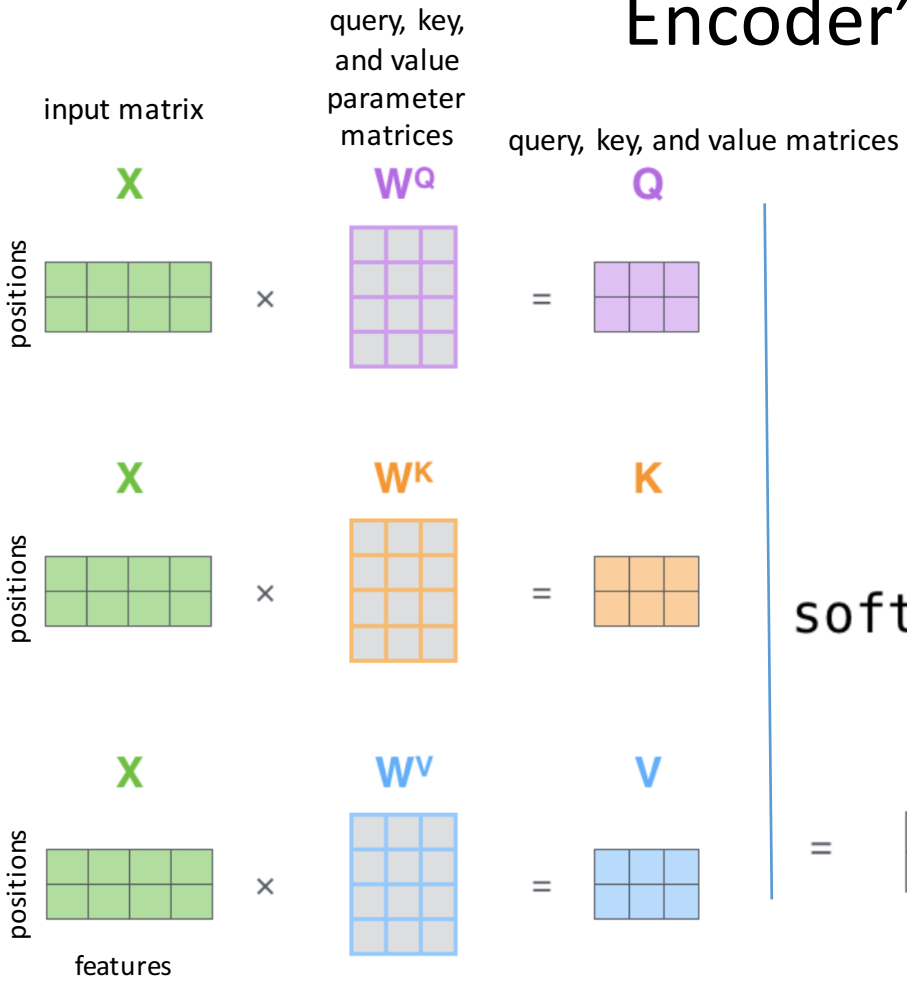
- The weight matrix is then multiplied with V, giving a matrix where each row is a weighted sum of the word representations at each position.



self-attention: Q,K, and V all come from the input (this results in a formulation different from the self-attention we've seen earlier)

encoder-decoder attention: K and V come from the top encoders' output, and Q comes from the current input of the decoder. This is similar to Luong et al. (2015)'s attention.

Encoder's self-attention



Example with an input sentence of two words and d_k=3.

In practice, multiple sets of Q,K,V matrices are used (multi-head attention) and the output vectors are concatenated.

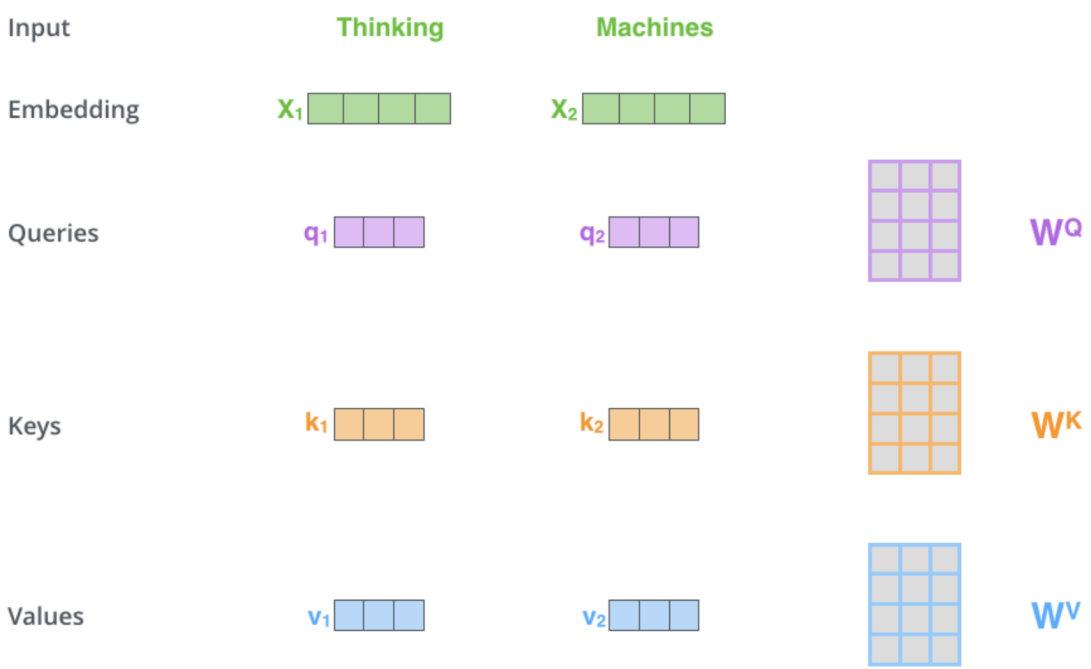
$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V$$

output of self-attention for word at position 1
output of self-attention for word at position 2

then, each vector independently passed to the FF layer

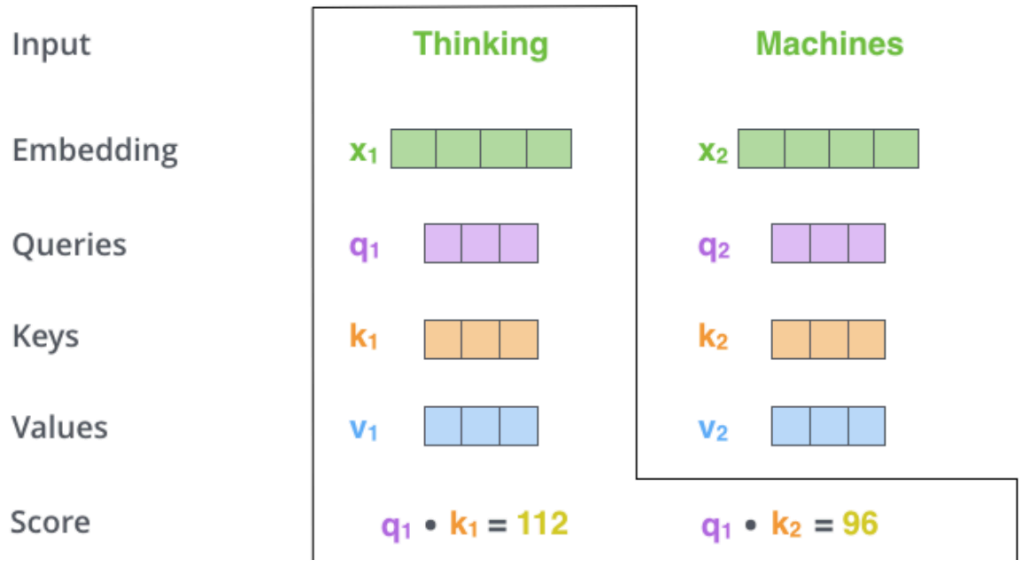
Self attention computation

- Consider phrase – “Thinking machines”.
- To calculate self-attention for “Action”, calculate scores for all words in the phrase with respect to “Action”.
- This score determines importance of other words when we are encoding a certain word in an input sequence



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

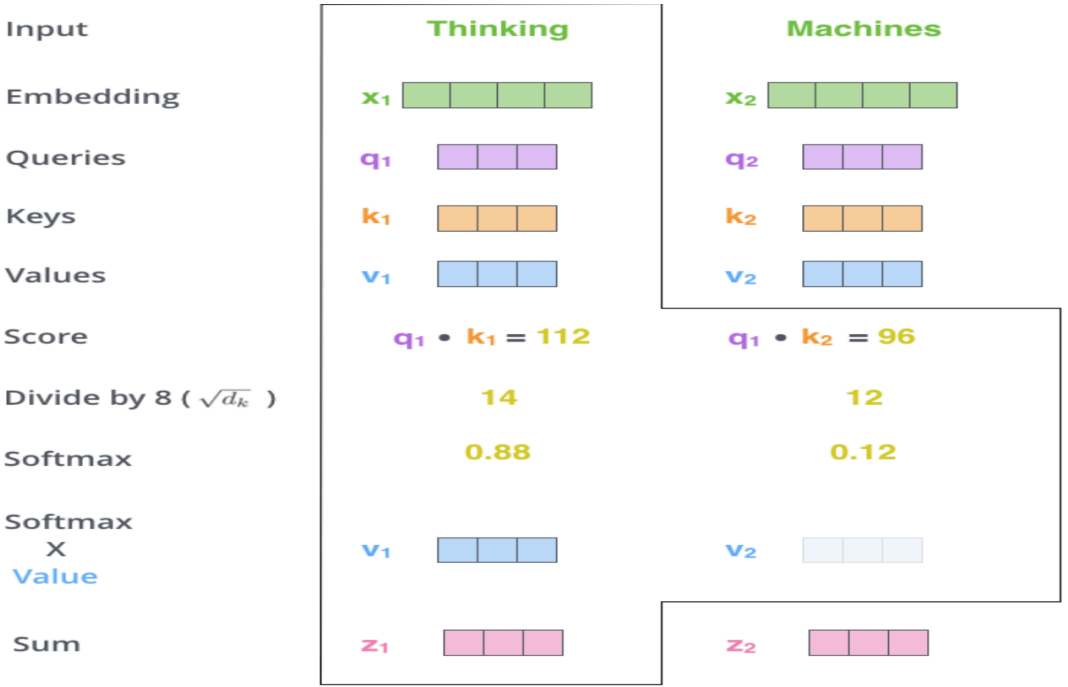
Self attention computation



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

Self attention computation

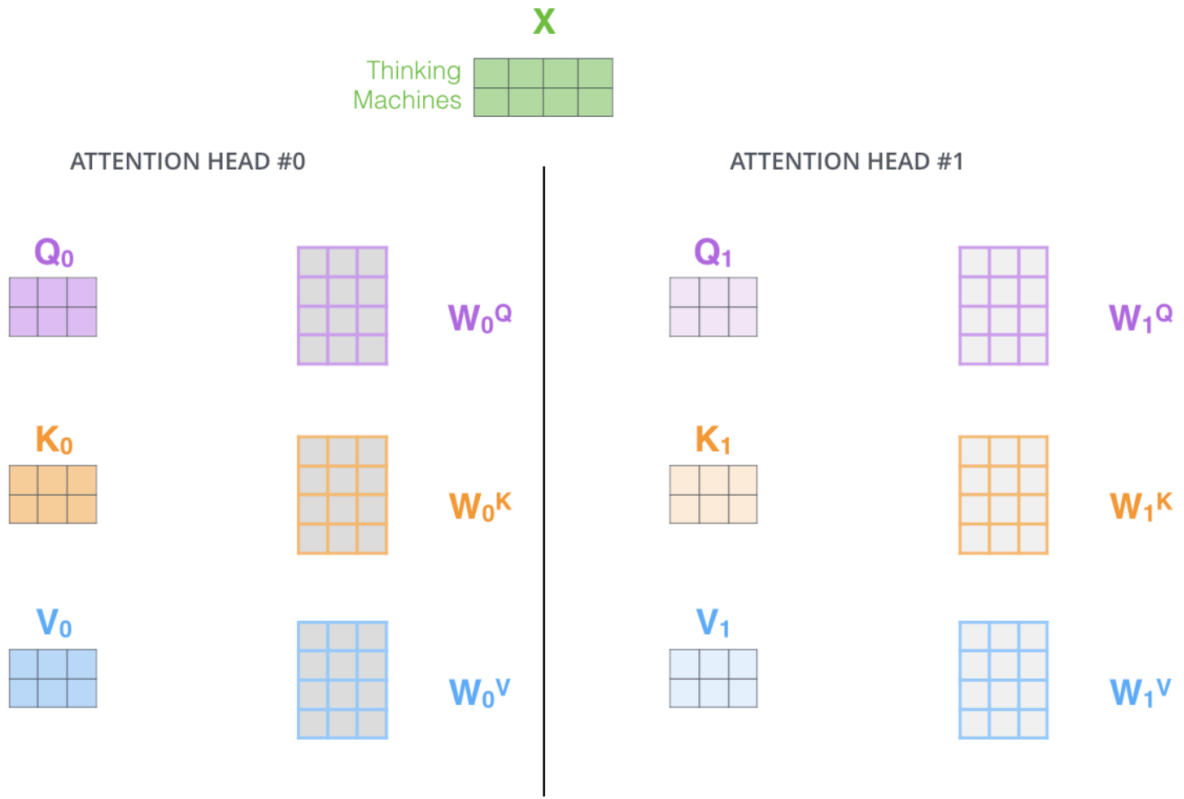
- Producing low dimensional vectors zi for each word
- Represent the encoding of a word within the sentence context.



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

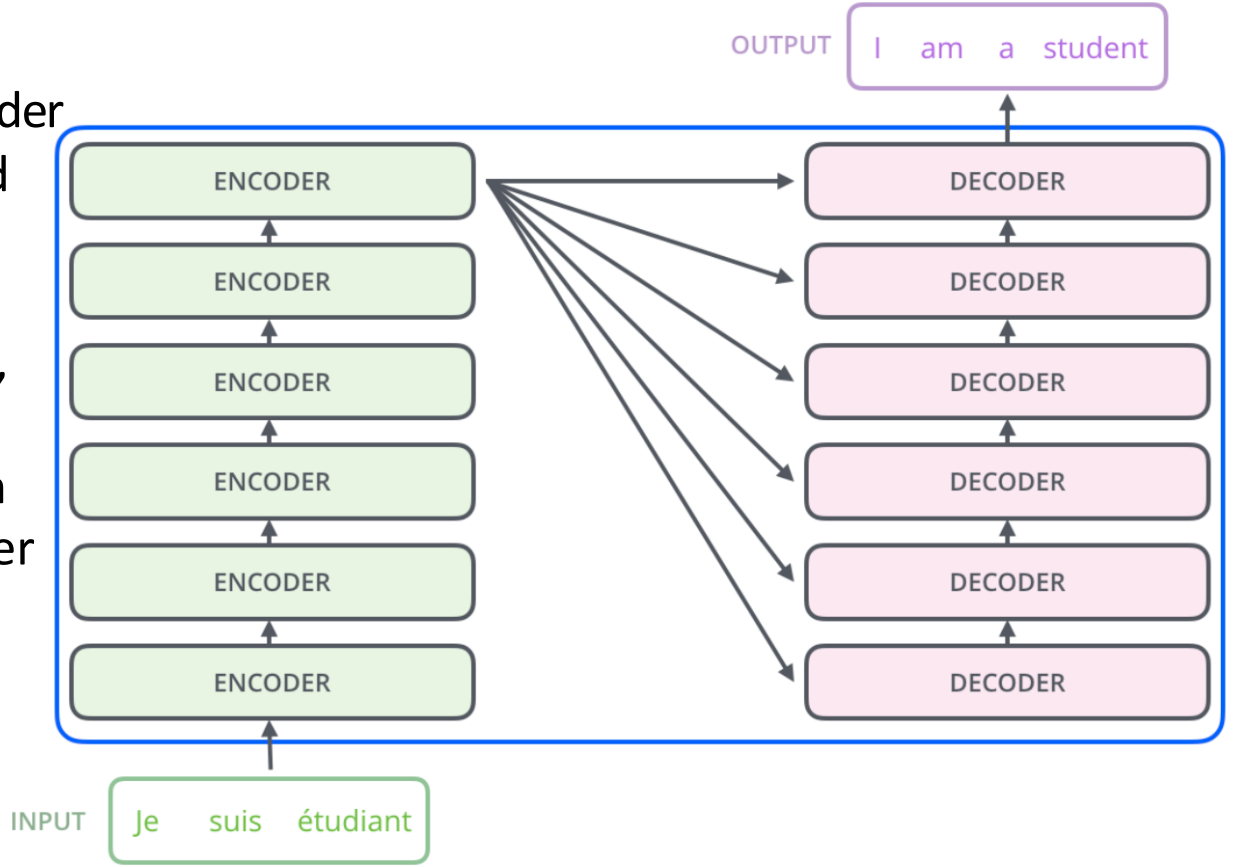
Multi-head attention

- Train multiple dependencies



Decoding phase: overview

- the output of the top encoder is transformed into key and value matrices (K,V).
- then, step-by-step process, where at each step, the encoder-decoder attention mechanism of each decoder uses K and V to transform the representation of its input (Q).



Transformer - experiments

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost

References

Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).

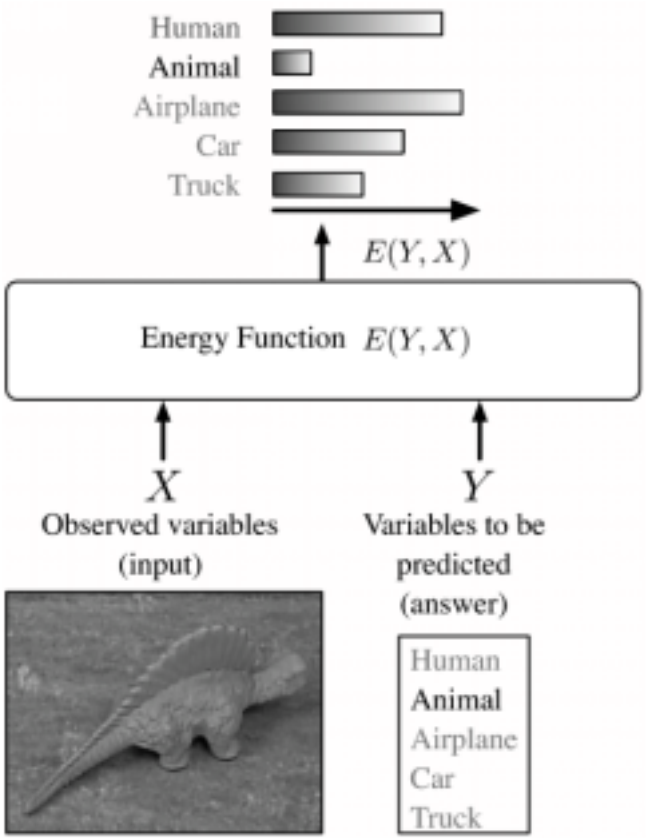
Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

ALTERNATIVE ARCHITECTURES

Energy-Based Models - EBM for classification



Model: Measures the compatibility between an observed variable X and a variable to be predicted Y through an energy function E(Y,X).

E(Y,X) assigns low energies to correct configurations of X and Y and higher energies to incorrect ones.

Inference process: Search for the Y that minimizes the energy function for a given X. In case of low cardinality, we can use exhaustive search.

$$Y^* = \text{argmin}_{Y \in \mathcal{Y}} E(Y, X).$$

A tutorial on energy-based learning (LeCun et al. 2006)

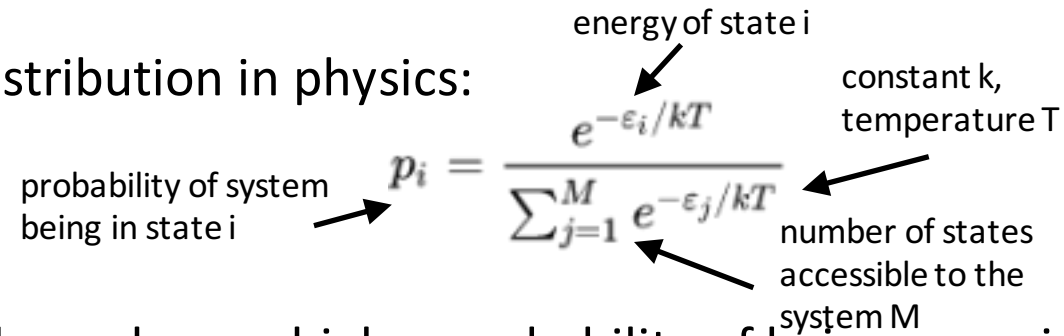
Energy-Based Models （EBMs）

- EBMs associate a scalar energy to each configuration of the variables of interest. Probability distribution is defined through an energy function:

$$P(\mathbf{x}) = \frac{e^{-Energy(\mathbf{x})}}{\sum_{\mathbf{x}} e^{-Energy(\mathbf{x})}} \qquad P(y|\mathbf{x}) = \frac{e^{-Energy(\mathbf{x},y)}}{\sum_{y} e^{-Energy(\mathbf{x},y)}}$$

Learning corresponds to modifying the energy function so that its shape has desirable properties.

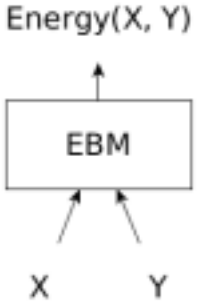
- Related to the Boltzmann distribution in physics:



States with lower energy will always have a higher probability of being occupied than the states with higher energy.

Learning deep architectures for AI (Bengio. 2009)

What Questions Can an EBM Answer?



1. Prediction, Classification & Decision Making:

- Which value of Y is most compatible with X?
- Training: give the lowest energy to the correct answer

2. Ranking:

- Is Y1 or Y2 more compatible with this X?
- Training: produce energies that rank the answers correctly

3. Detection:

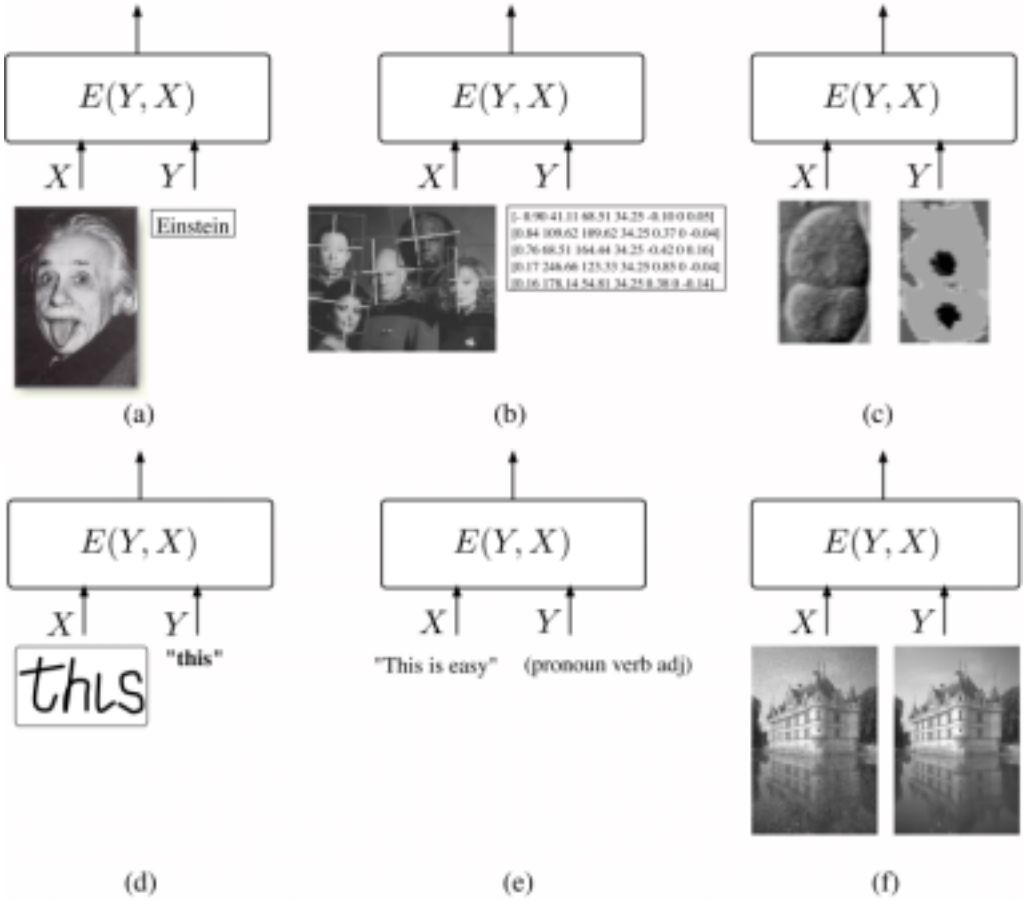
- Is this value of Y compatible with X? (e.g. face detection)
- Training: energies that increase as the image looks less like a face.

4. Conditional Density Estimation:

- What is the conditional distribution P(Y|X)?
- Training: differences of energies must be just so.

A tutorial on energy-based learning (LeCun et al. 2006)

Complex Tasks: Inference is non-trivial



- When the cardinality or dimension of Y is large, exhaustive search is impractical.
- We need to use a "smart" *inference procedure* to find the Y that (globally or locally) minimizes the energy function E(Y, X), such as gradient-based optimization, min-sum, Viterbi algorithms, etc. (depends on the internal structure of the model.)

A tutorial on energy-based learning (LeCun et al. 2006)

EBM Architecture, Training, Inference

- Parameterized energy functions $\mathcal{E} = \{E(W, Y, X) : \ W \in \mathcal{W}\}.$
- Training set $\mathcal{S} = \{(X^i, Y^i) : i = 1 \ldots P\}$
- Training process $W^* = \min_{W \in \mathcal{W}} \mathcal{L}(E, \mathcal{S})$
- Loss functional (*measures the quality of an energy function using the S)*

$$\mathcal{L}(E, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} L(Y^i, E(W, \mathcal{Y}, X^i)) + R(W).$$

Per-sample loss

Desired answer

Energy surface for a given Xi as Y varies

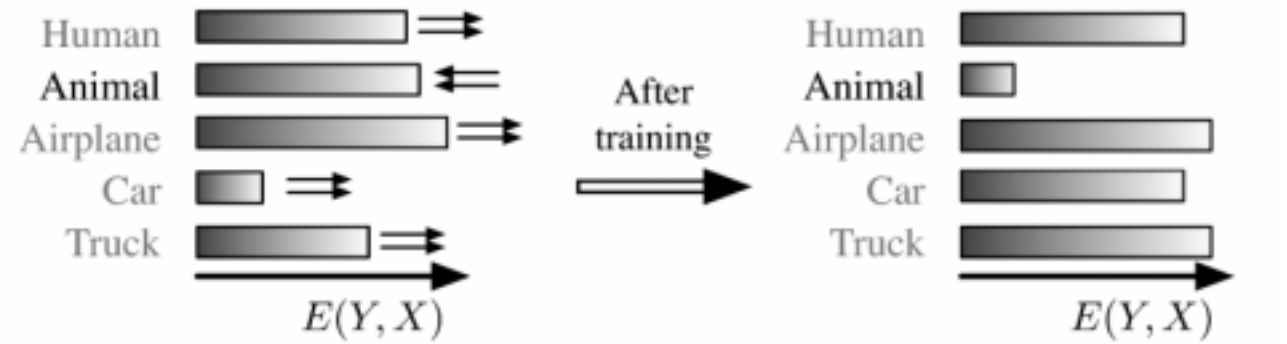
Regularizer

- Inference process

$$Y^* = \text{argmin}_{Y \in \mathcal{Y}} E(Y, X).$$

A tutorial on energy-based learning (LeCun et al. 2006)

Designing a Loss Functional



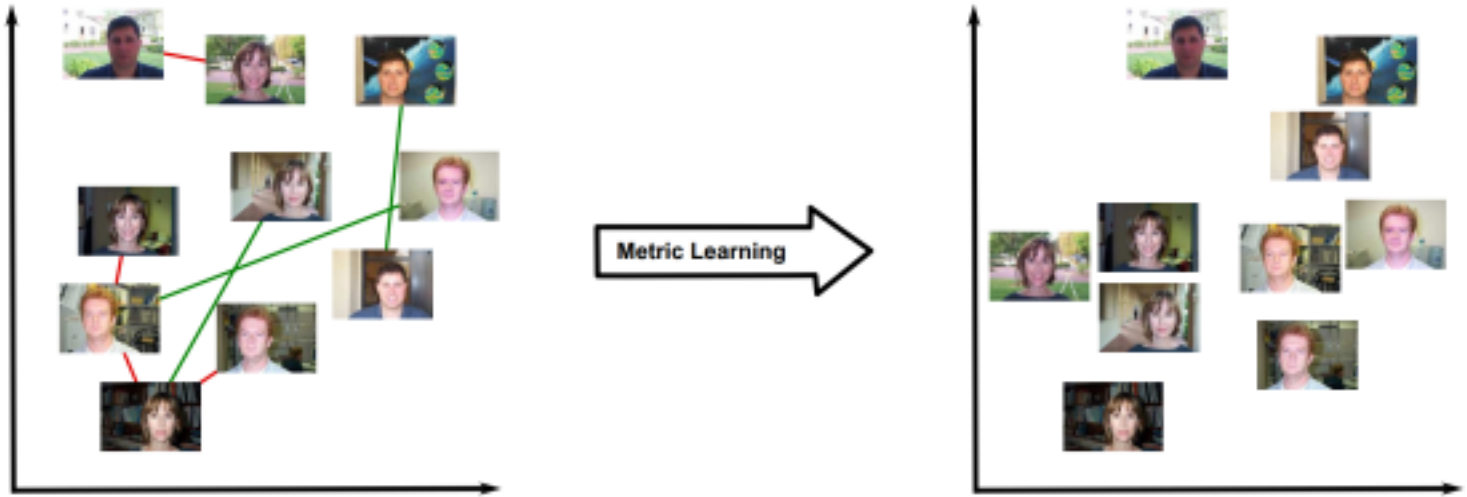
- Correct answer has the lowest energy -> LOW LOSS
- Lowest energy is not for the correct answer -> HIGH LOSS

A tutorial on energy-based learning (LeCun et al. 2006)

Deep Metric Learning

- Metrics are ubiquitous in machine learning, measuring the distance (or similarity) between objects. Each problem has its own semantic notion of similarity, which is often badly captured by standard metrics (e.g., Euclidean distance).

- Learn a **parameterized** distance function $D_w(x, x')$ that assigns small (resp. large) distance to pairs of examples that are *semantically* similar (resp. dissimilar).

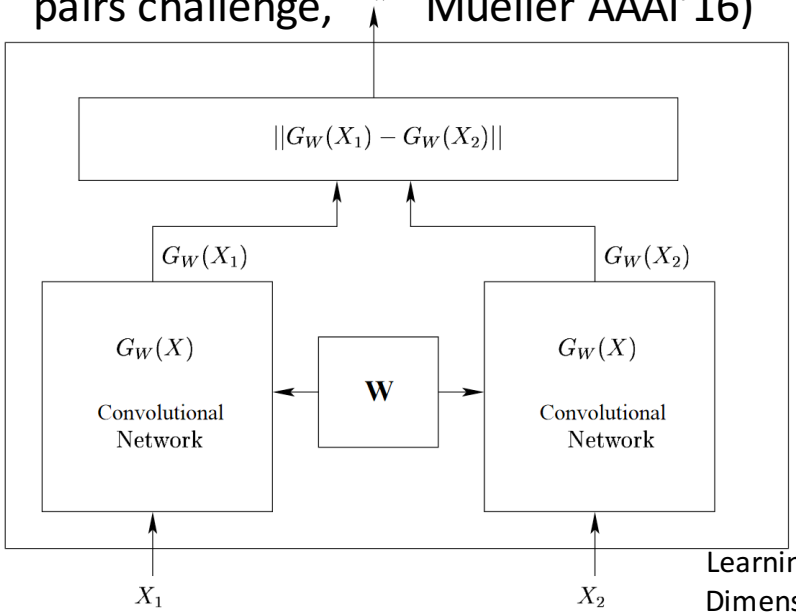


A survey on metric learning for feature vectors and structured data (Bellet. 2013)

Siamese architecture

Overview

- Two encoders **learn the similarity** between training examples passed as pairs
- Weights are *shared* between branches
- Developed in Computer Vision
- Application to learning text similarity (e.g., winners of the 2017 Kaggle Quora question pairs challenge, Mueller AAAI'16)



Sumit Chopra

Imagen Technologies
Verified email at imagen.ai - Homepage

Machine Learning Energy Based Models Deep Learning

Raia Hadsell

Google DeepMind
Verified email at google.com - Homepage

Artificial Intelligence Deep Learning Robotics

Yann LeCun

Chief AI Scientist at Facebook & Silver Professor at the Courant Institute New York University
Verified email at cs.nyu.edu - Homepage

AI machine learning computer vision robotics image compression

genuine (similar)

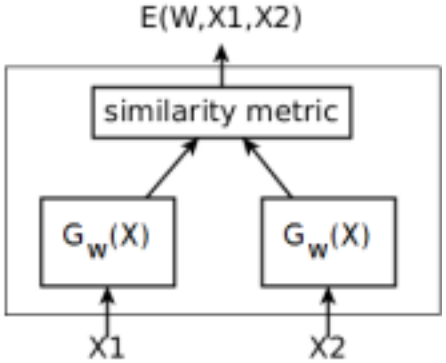
impostor (dissimilar)

Learning a similarity metric discriminatively, with application to face verification (2005)
Dimensionality reduction by learning an invariant mapping (2005)

Deep Metric Learning: Siamese network

- An architecture of EBMs, where variables X1 and X2 are passed through two instances of a same function G_w (dual-branch networks with tied weights).
- Energy function E(W, X1, X2) is defined as *symmetric* similarity metric between G_w(X1) and G_w(X2).
- Training set consists of labeled pairs, when X1 and X2 are similar (e.g. two pictures of the same person), Y = 0; when they are different, Y = 1.
- Training process involves finding an optimal W that assigns low energies to similar pairs and higher energies to dissimilar pairs. (in case of L1 distance)

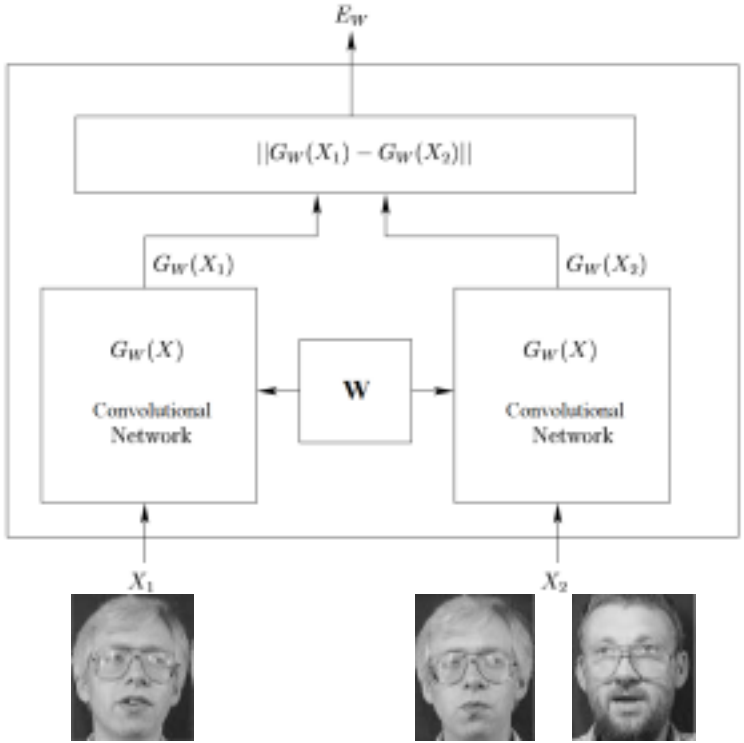
Originally designed for signature verification (Bromley et al., 1993)



Learning a similarity metric discriminatively, with application to face verification (Chopra et al. 2005)

Application in face verification (1/2)

(energy should be low given the *similar* pair, be high given the *dissimilar* pair)



Contrastive loss

An *energy function* E_W computes the L1 distance between the two embeddings $G_W(X_1)$ and $G_W(X_2)$

Images are separately encoded by a shared CNN

Input: pair of images (X_1, X_2) and label Y:

Y = 0 if X_1 and X_2 are similar

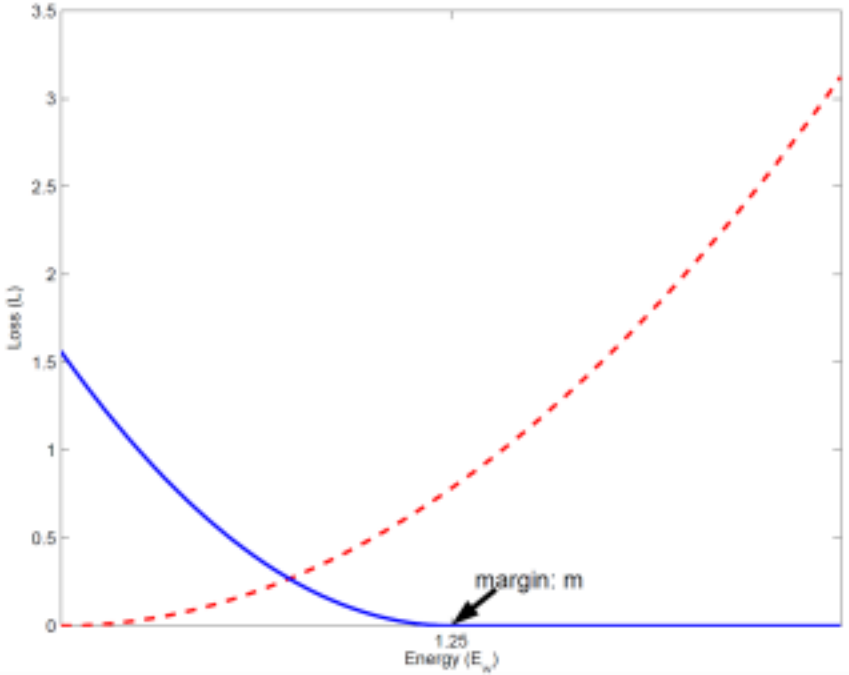
Y = 1 if they are dissimilar

Learning a similarity metric discriminatively, with application to face verification (Chopra et al. 2005)

Application in face verification (2/2)

Contrastive loss:

$$L(W,(Y,X_1,X_2)^i) = (1-Y)L_S\left(E_W^i\right) + YL_D\left(E_W^i\right)$$



L_S: partial loss for similar instances (Y=0)

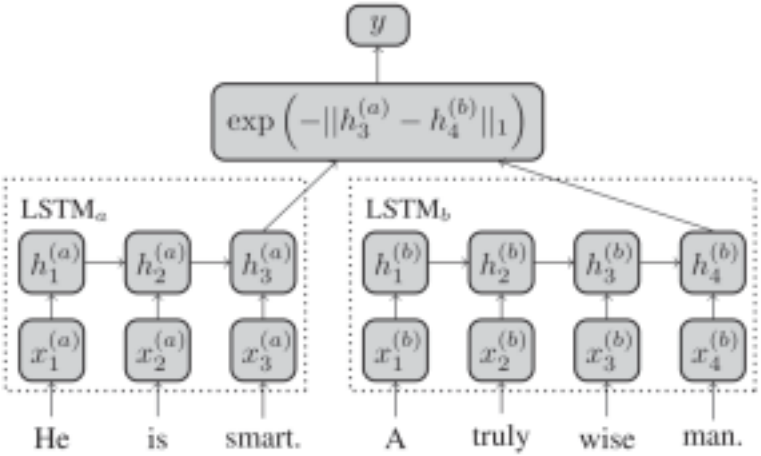
L_D: partial loss for dissimilar instances (Y=1)

- Goal: optimize W so that E_W is **low for similar pairs** and **high for dissimilar pairs**
- attraction/repulsion spring analogy

Learning a similarity metric discriminatively, with application to face verification (Chopra et al. 2005)

Applications in NLP (1/2)

Learning similarity metric between two sentences. (winners of the 2017 Kaggle Quora question pairs challenge)



Mean squared error loss

An *energy function* E_W computes the exponent negative manhattan distance between the two embeddings $G_W(X_1)$ and $G_W(X_2)$

Sentence are separately encoded by a shared LSTM

Input: pair of sentences (X_1, X_2) (represented as a sequence of word vectors) and label Y:

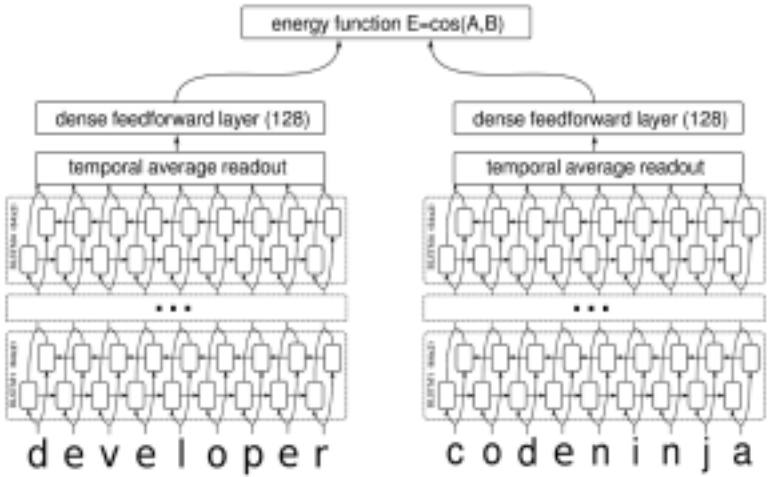
Y = 1 if X_1 and X_2 are similar

Y = 0 if they are dissimilar

Siamese Recurrent Architectures for Learning Sentence Similarity (Mueller and Thyagarajan. 2016)

Applications in NLP (1/2)

Learning similarity metric between two sequences of characters (job titles).



Contrastive loss

An *energy function* E_W computes the cosine similarity between the two embeddings $G_W(X_1)$ and $G_W(X_2)$

Words are separately encoded by shared four layers of Bidirectional LSTM nodes and one densely connected feedforward layer.

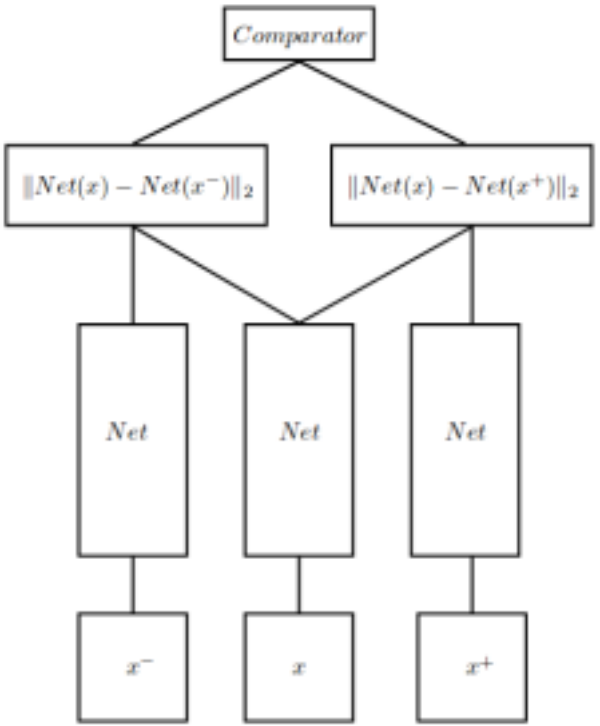
Input: pair of words (X_1, X_2) and label Y:

$Y = 1$ if X_1 and X_2 are similar

$Y = 0$ if they are dissimilar

Learning text similarity with siamese recurrent networks (Neculoiu et al. 2016)

Deep Metric Learning: Triplet network



A Triplet network is comprised of 3 instances of the same *Net(x)* (with shared parameters).

Training is performed by feeding the network with triplets (x, x^+, x^-) where x and x^+ are of the same class, and x^- is of different class.

Intermediate values consist of distances between each of x^+ and x^- against the reference x.

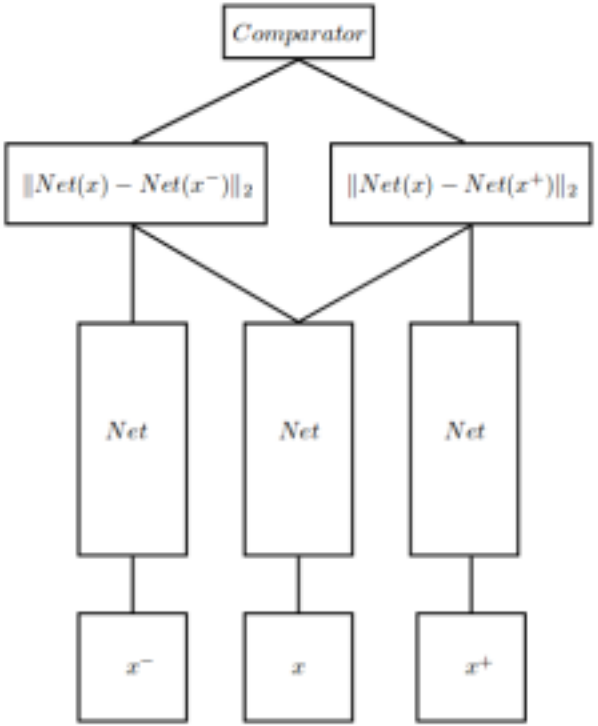
$$TripletNet(x, x^-, x^+) = \begin{bmatrix} \|Net(x) - Net(x^-)\|_2 \\ \|Net(x) - Net(x^+)\|_2 \end{bmatrix} \in \mathbb{R}^2_+$$

Siamese or Triplet ? Depending on data, training strategies, network design, ...

Deep metric learning using triplet network (Hoffer and Ailon. 2005)

Deep Metric Learning: Triplet network

The objective is to learn a metric embedding, the label determines which example is *closer* to x.



Loss function is MSE on the softmax result, compared to the (0, 1) vector:

$$Loss(d_+, d_-) = \|(d_+, d_- - 1)\|_2^2 = const \cdot d_+^2$$

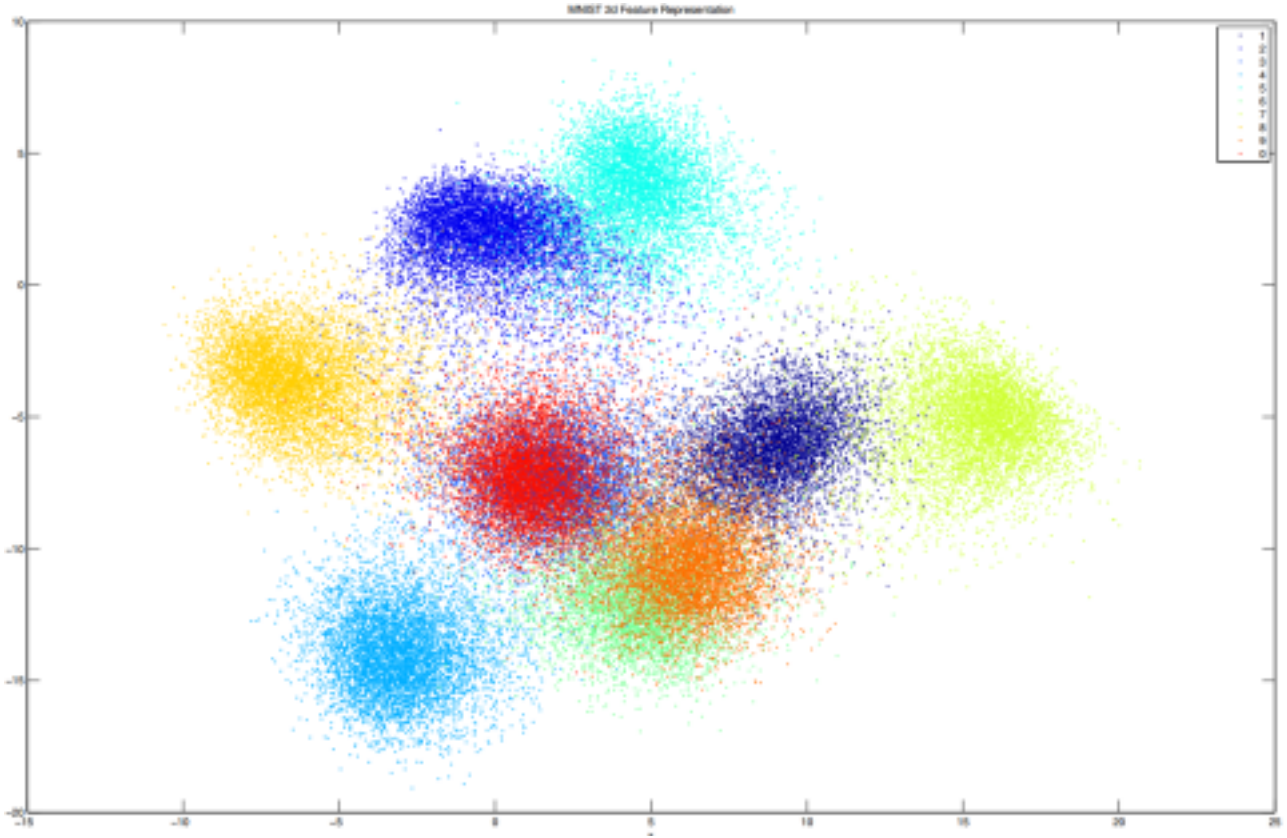
$$d_+ = \frac{e^{\|Net(x)-Net(x^+)\|_2}}{e^{\|Net(x)-Net(x^+)\|_2} + e^{\|Net(x)-Net(x^-)\|_2}}$$

$$d_- = \frac{e^{\|Net(x)-Net(x^-)\|_2}}{e^{\|Net(x)-Net(x^+)\|_2} + e^{\|Net(x)-Net(x^-)\|_2}}$$

FaceNet: A Unified Embedding for Face Recognition and Clustering (Schroff et al.. 2015)

Deep metric learning using triplet network (Hoffer and Ailon. 2015)

Application on MNIST dataset



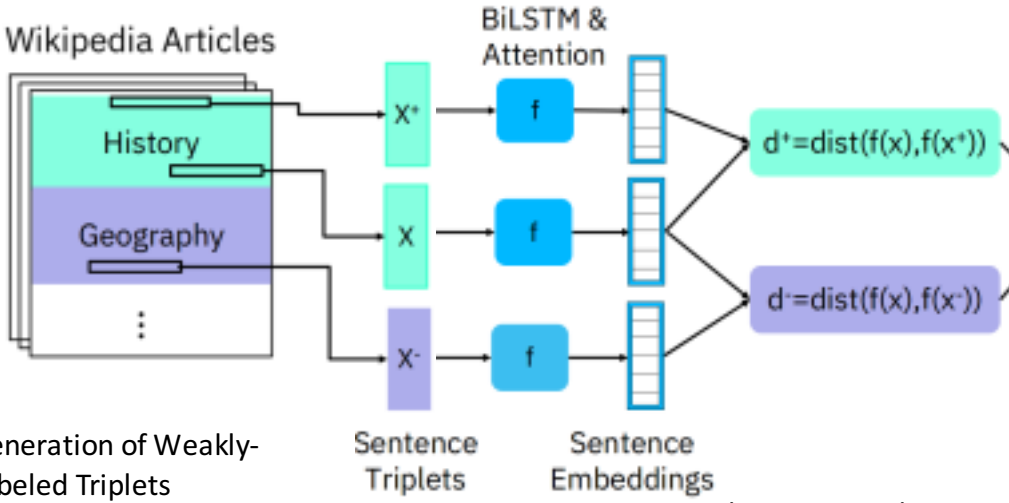
MNIST - Euclidean representation of embedded test data, projected onto top two singular vectors

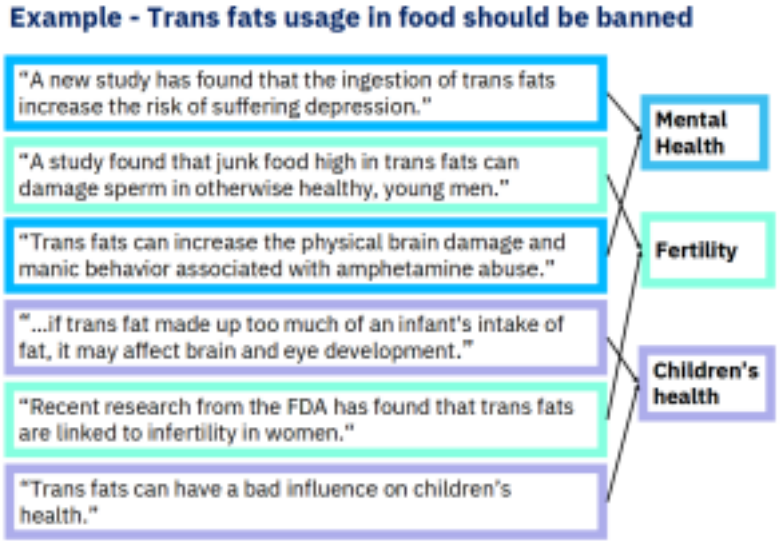
Deep metric learning using triplet network (Hoffer and Ailon. 2015)

Application in NLP

Argumentation mining:
Organize arguments into themes to generate a compelling argumentative narrative.

Multi-document summarization:
Organize selected sentences into meaningful sections and paragraphs.





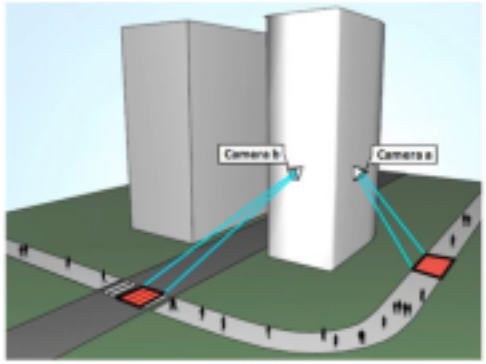
Learning a similarity metric between sentences dedicated to thematic clustering

Learning Thematic Similarity Metric from Article Sections Using Triplet Networks (Dor et al. 2018)

Deep Metric Learning: Quadruplet network

Probe Rank List
(groundtruth in green frames)

Results of triplet loss on Testing Set



Observation: The trained model (on the left figure) with triplet loss has low generalization ability on testing set.

- Some false positives are more similar to the probe image.
- These false positives never show up in training set.

Beyond triplet loss: a deep quadruplet network for person re-identification (Chen et al. 2017)

Deep Metric Learning: Quadruplet network

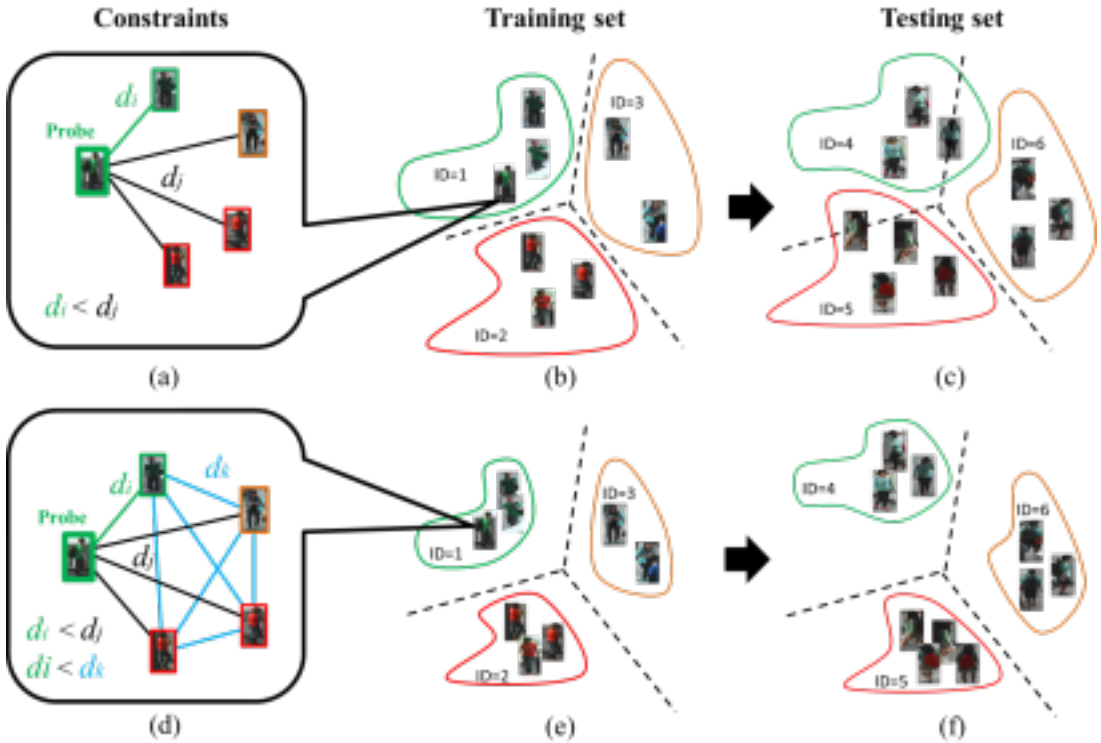
$$L_{trp} = \sum_{i,j,k}^{N} [g(x_i, x_j)^2 - g(x_i, x_k)^2 + \alpha_{trp}]_+$$

g(xi, xj) is leaned metric
[z]+ = max(z, 0)
The threshold α is a margin that is enforced between positive and negative pairs

$$L_{quad} = \sum_{i,j,k}^{N} [g(x_i, x_j)^2 - g(x_i, x_k)^2 + \alpha_1]_+ + \sum_{i,j,k,l}^{N} [g(x_i, x_j)^2 - g(x_l, x_k)^2 + \alpha_2]_+$$

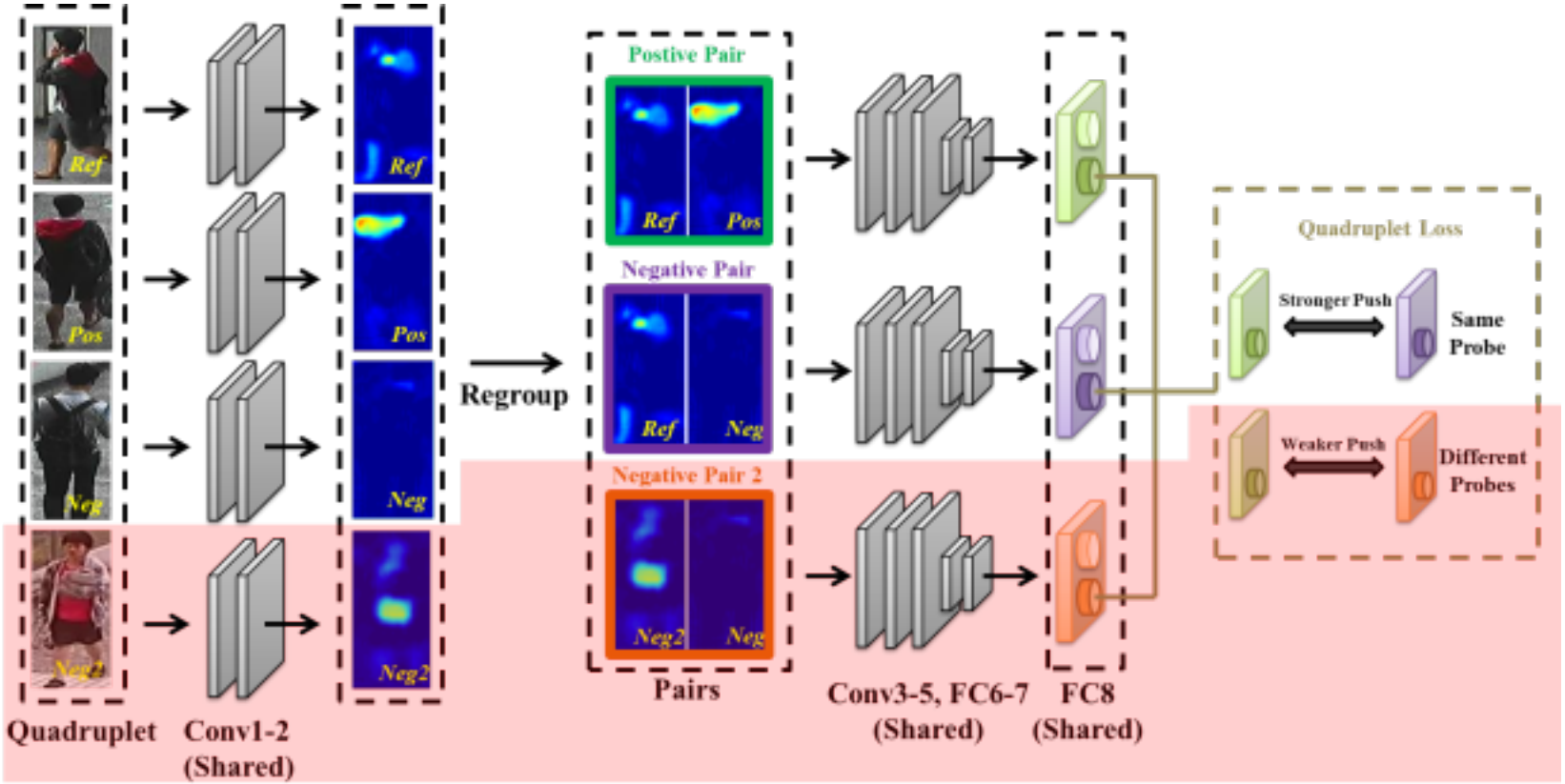
$$s_i = s_j, s_l \neq s_k, s_i \neq s_l, s_i \neq s_k$$

where α1 and α2 are the values of margins in two terms and si refers to the person ID of image xi



Beyond triplet loss: a deep quadruplet network for person re-identification (Chen et al. 2017)

Deep Metric Learning: Quadruplet network



Beyond triplet loss: a deep quadruplet network for person re-identification (Chen et al. 2017)

References (1/3)

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364.

Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., & Wierstra, D. (2015). DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*.

Hadsell, Raia, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant mapping." *Computer vision and pattern recognition, 2006 IEEE computer society conference on*. Vol. 2. IEEE, 2006.

Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).

Mueller, J., & Thyagarajan, A. (2016, February). Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI* (pp. 2786-2792).

References (2/3)

Neculoiu, Paul, Maarten Versteegh, and Mihai Rotaru. "Learning text similarity with siamese recurrent networks." *Proceedings of the 1st Workshop on Representation Learning for NLP*. 2016.

Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015, June). Show, attend and tell: Neural image caption generation with visual attention. In International Conference on Machine Learning (pp. 2048-2057).

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. NAACL 2016 (pp. 1480-1489).

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, *1*(0).

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, *2*(1), 1-127.

References (3/3)

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems* (pp. 737-744).

Neculoiu, P., Versteegh, M., & Rotaru, M. (2016). Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP* (pp. 148-157).

Hoffer, E., & Ailon, N. (2015, October). Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition* (pp. 84-92). Springer, Cham.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

Dor, L. E., Mass, Y., Halfon, A., Venezian, E., Shnayderman, I., Aharonov, R., & Slonim, N. (2018). Learning Thematic Similarity Metric from Article Sections Using Triplet Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Vol. 2, pp. 49-54).

Bellet, A., Habrard, A., & Sebban, M. (2013). A survey on metric learning for feature vectors and structured data.

Chen, W., Chen, X., Zhang, J., & Huang, K. (2017, July). Beyond triplet loss: a deep quadruplet network for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Vol. 2, No. 8).