Deep Learning for Language

e.g. Natural Language Processing and Information Retrieval

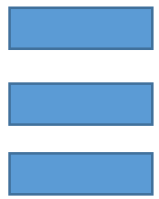
What is Machine Learning

- Supervised Learning **with label**
- Unsupervised Learning **without label**
- Reinforced Learning with **delayed label**.

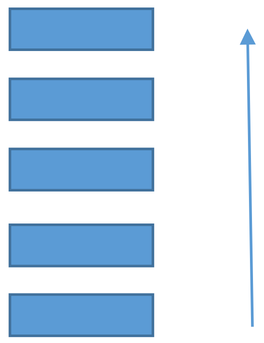
Machine Learning

- Linear Regression
- Naïve Bayes
- Decision Tree
- Support vector machine
- **Artificial neural Network**

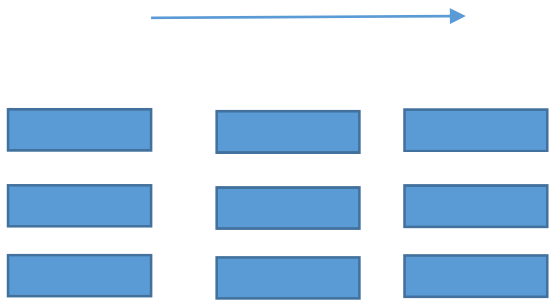
What is Deep Learning



Artificial neural Network

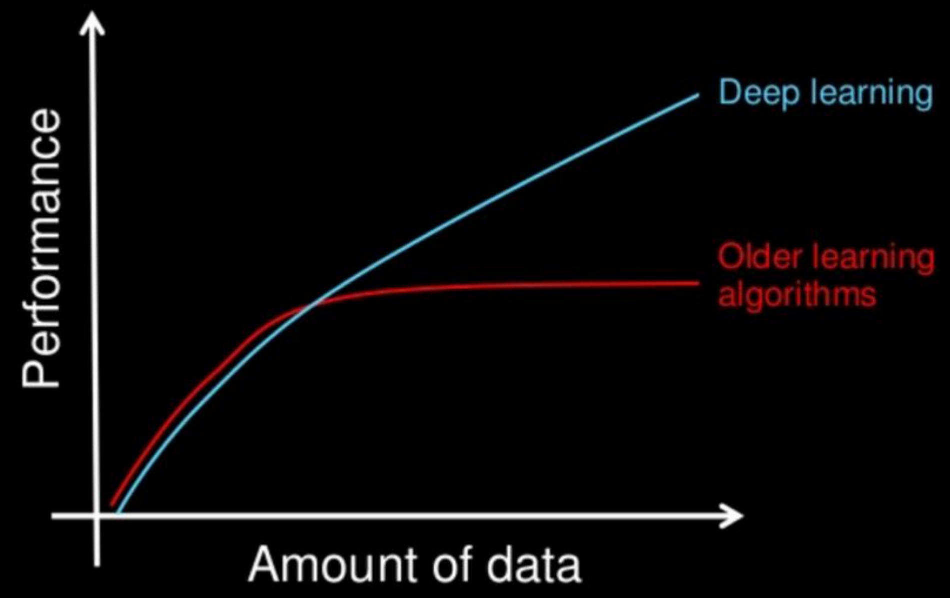


Convolutional neural Network



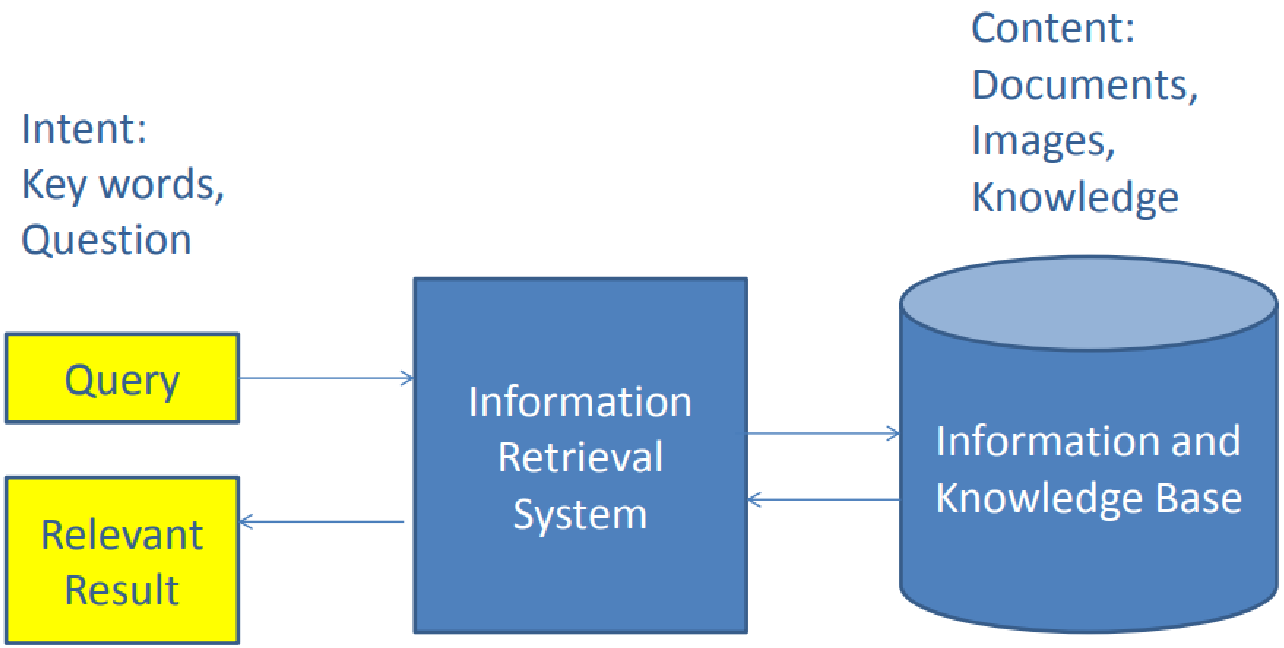
Recurrent neural Network

Why deep learning



How do data science techniques scale with amount of data?

IR background



Key Questions: How to Represent Intent and Content, How to Match Intent and Content

Traditional IR – Tfidf example

Document:

Star Wars: Episode VII
Three decades after the defeat of the Galactic Empire, a new threat arises.

Query:
star wars the force awakens reviews

$$q = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \xrightarrow{f(q,d)} d = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$f_{VSM}(q,d) = \frac{\langle q,d \rangle}{\| q \| \cdot \| d \|}$$

- Representing query and document as word vectors
- calculating cosine similarity between them

Modern IR – Learn to Rank

Query:
star wars the force awakens reviews

Document:

Star Wars: Episode VII
Three decades after the defeat of the Galactic Empire, a new threat arises.

(star wars)
(the force awakens)
(reviews)

$$
\begin{array}{ccc}
q & & d \\
v_{q1} & \xrightarrow{\vec{f}(q,d)} & v_{d1} \\
\vdots & & \vdots \\
v_{qm} & & v_{dn}
\end{array}
$$

- Conducting query and document understanding
- Representing query and document as multiple feature vectors
- Calculating multiple matching scores between query and document
- Training ranker with matching scores as features using learning to rank

Features + Ranking

Features：
- Language model
- BM25
- Title/Snippet/Document
- Pagerank

Ranking:
- Point-wise
- Pair-wise
- List-wise

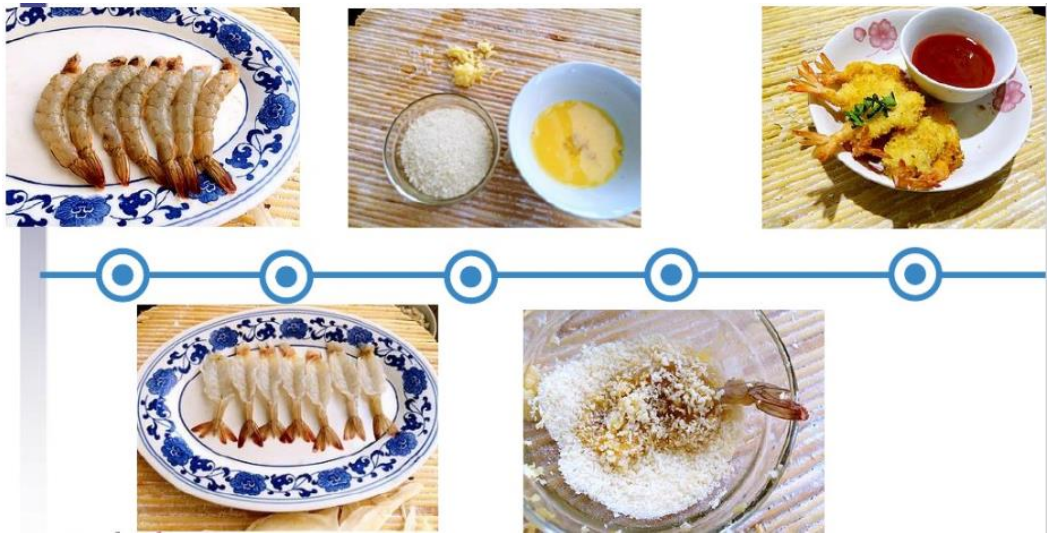
Credited to Prof. Songchun Zhu

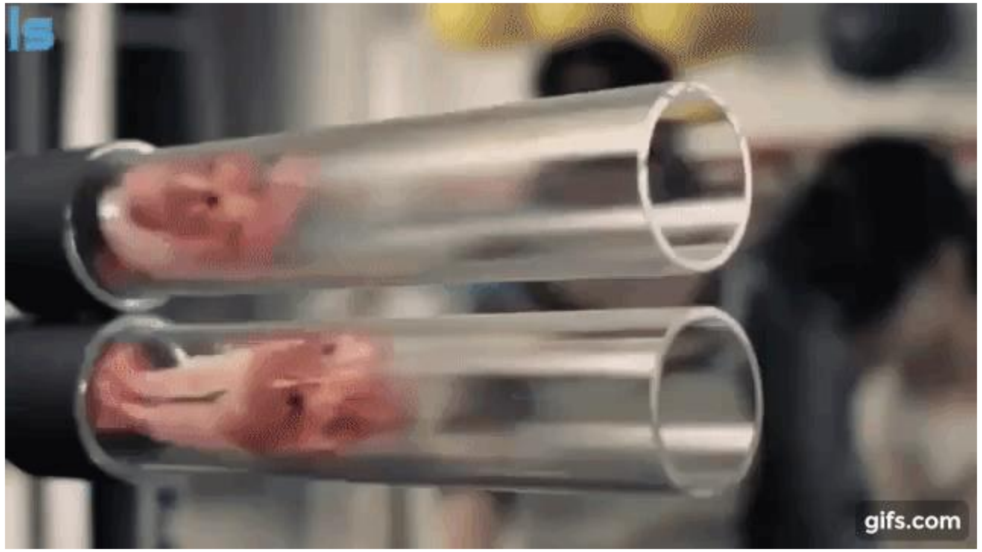
Example of **Mismatch**

Query	Document	Term Matching	Semantic Matching
seattle best hotel	seattle best hotels	no	yes
pool schedule	swimmingpool schedule	no	yes
natural logarithm transformation	logarithm transformation	partial	yes
china kong	china hong kong	partial	no
why are windows so expensive	why are macs so expensive	partial	no

Hang li, http://www.hangli-hl.com/uploads/3/4/4/6/34465961/tsinghua_opportunities_and_challenges_in_deep_learning_for_information_retrieval.pdf

End-to-end





https://www.youtube.com/watch?v=TYpBJ71VW9g

The inputting features are also **learnable/trainable**

Credited to Dr. Naiyan Wang

Trends for Neural IR

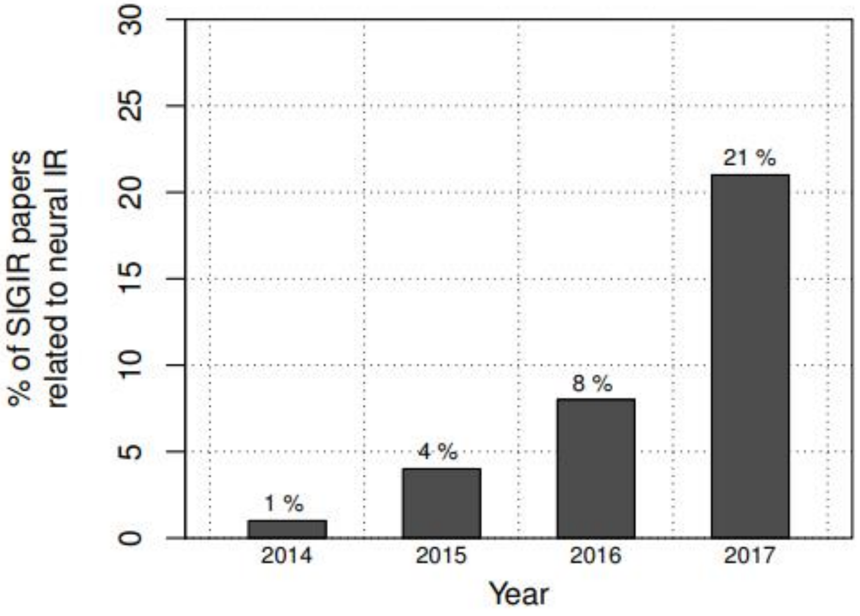
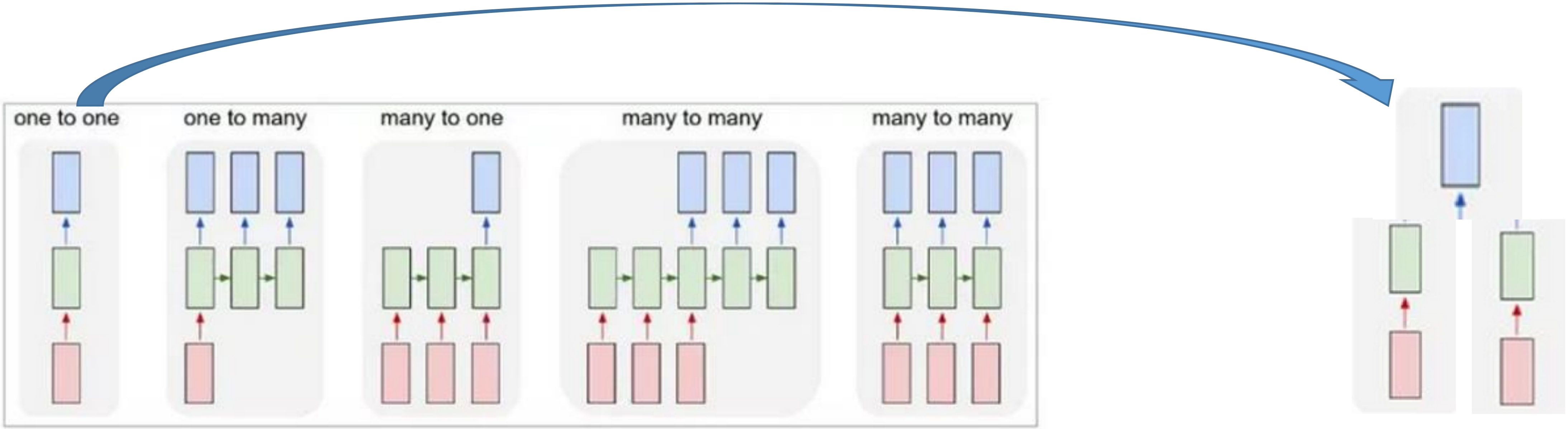


Figure 1: The percentage of neural IR papers at the ACM SIGIR conference—as determined by a manual inspection of the paper titles—shows a clear trend in the growing popularity of the field.

Mitra B, Craswell N. Neural Models for Information Retrieval[J]. arXiv preprint arXiv:1705.01509, 2017.

Tasks in IR/NLP



- Classification: assigning a label to a string

$$s \rightarrow c$$

- Matching: matching two strings

$$s, t \rightarrow \mathbf{R}^{+}$$

- Translation: transforming one string to another

$$s \rightarrow t$$

- Structured prediction: mapping string to structure

$$s \rightarrow s'$$

Credited by Hang li

Fundamental Demo In Code with PyTorch pseudo code

- Model = LSTM/CNN/Capsule/…
- text,lable = Dataset.nextBatch()
- representation = Model(text)

- Classification = FC(representation) FC : Mapping to label size

- Translation = Decode(representation)

- Matching = Cosine(representation1, representation2)

- Sequential_labelling = FCs(representation**s**)

Background of Neural IR

- Trends of DL for IR
- **Word embedding**
- Neural network
- DL for IR/NLP

Localist representation

Size color … unknown

- BMW [1, 0, 0, 0, 0] [.3, .7, .2, .1, .5]
- Audi [0, 0, 0, 1, 0] [.5, .3, .2, .1, .0]
- Benz [0, 0, 1, 0, 0] [.2, .0, .31, .03, .01]
- Polo [0, 0, 0, 1, 0] [.1, .1, .5, .5, 0.2]

http://www.cs.toronto.edu/~bonner/courses/2014s/csc321/lectures/lec5.pdf

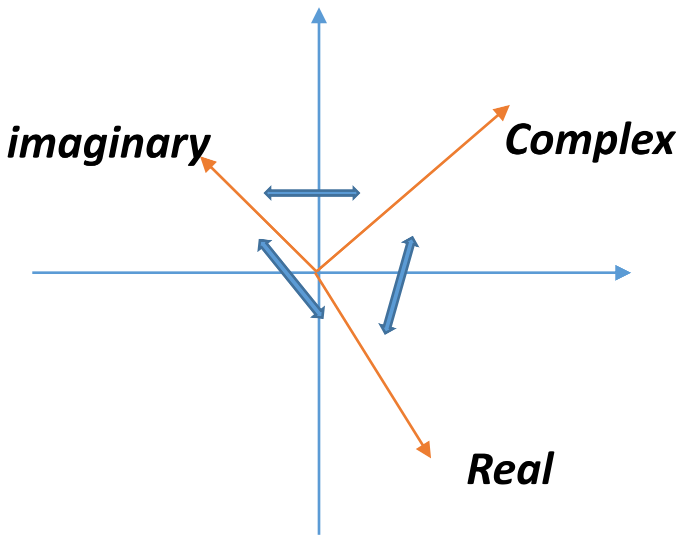
Distributed representation

Size color ... unknown

- BMW [1, 0, 0, 0, 0] [.3, .7, .2, .1, .5]
- Audi [0, 0, 0, 1, 0] [.5, .3, .2, .1, .0]
- Benz [0, 0, 1, 0, 0] [.2, .0, .31, .03, .01]
- Polo [0, 0, 0, 1, 0] [.1, .1, .5, .5, 0.2]

Embedding

Distributional hypothesis *linguistic items with similar distributions have similar meanings*



*Life is **complex**. It has both **real** and **imaginary** parts*

https://en.wikipedia.org/wiki/Distributional_semantics

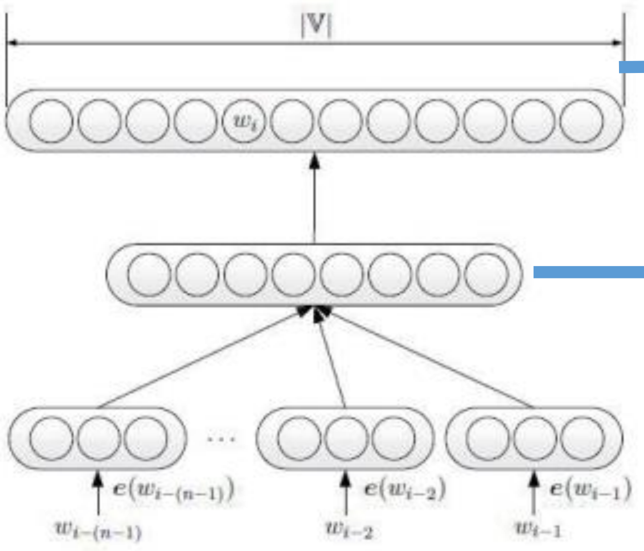
How to get Distributed representation

- Matrix Factorization
  - Word-word Matrix
  - Document-word Matrix
    - PLSA
    - LDA
- Sample-based Prediction
  - NNLM
  - C & W
  - Word2vec

Glove is a combination between these two schools of approaches

Levy, Omer, and Yoav Goldberg. "Neural word embedding as implicit matrix factorization." *Advances in neural information processing systems*. 2014.

NNLM to Word2vec



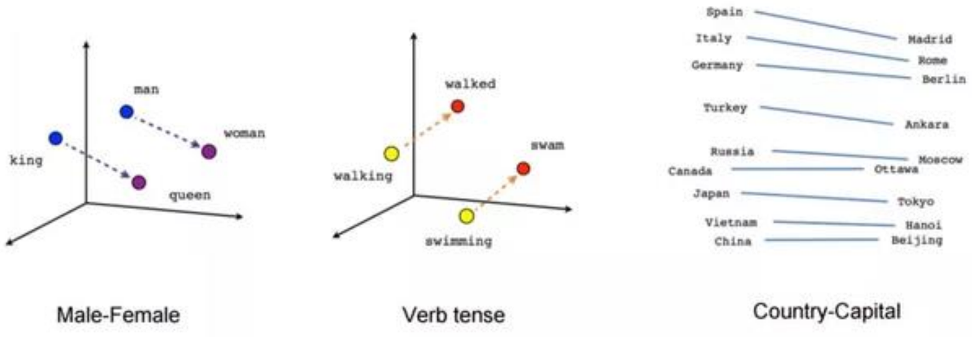
Negative sampling
Hierarchical softmax

Concat -> mean

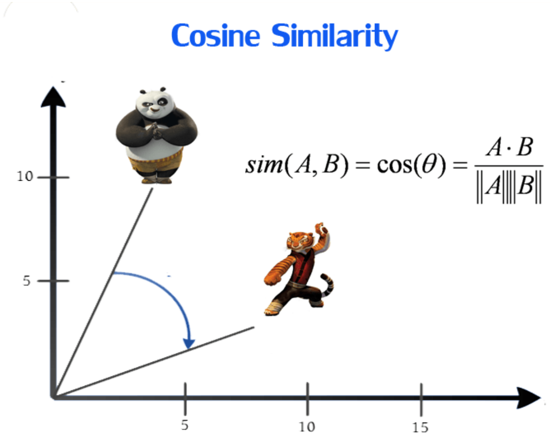
Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of machine learning research, 2003, 3(Feb): 1137-1155.
Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

Advantage of word embedding

- Linguistic regulation
  - $\overrightarrow{king} - \overrightarrow{man} = \overrightarrow{queen} - \overrightarrow{woman}$



- Semantic matching
  - As the initial input Feature/**Weight** for NN



Only Word Embedding ?

Which is the most similar word of “Italy” ?

Maybe “Germany” or “Pasta” ?

You cannot **guarantee** that each similar word pair could help your matching ?

Nie Jianyun said in SIGIR 2016 Chinese-Author Workshop, Tsinghua University, Beijing

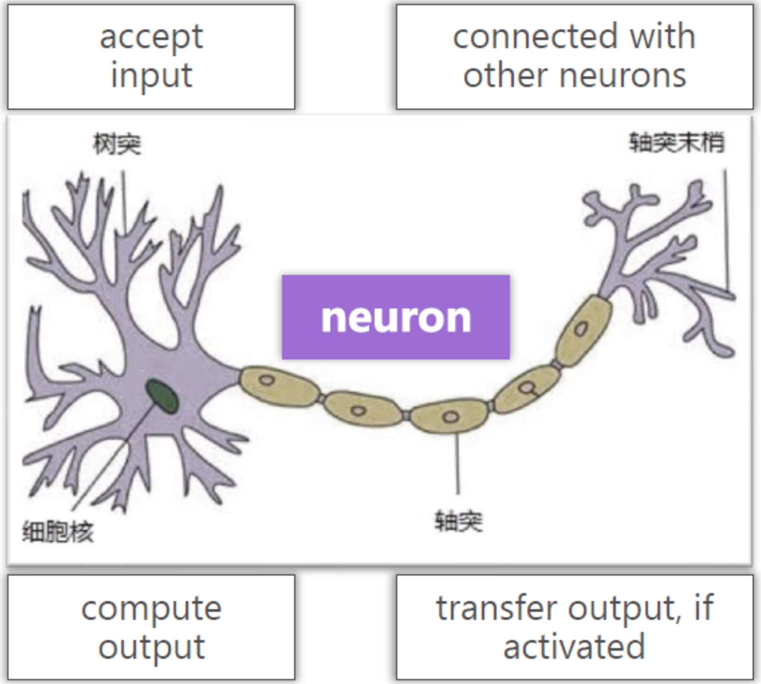
Background of Neural IR

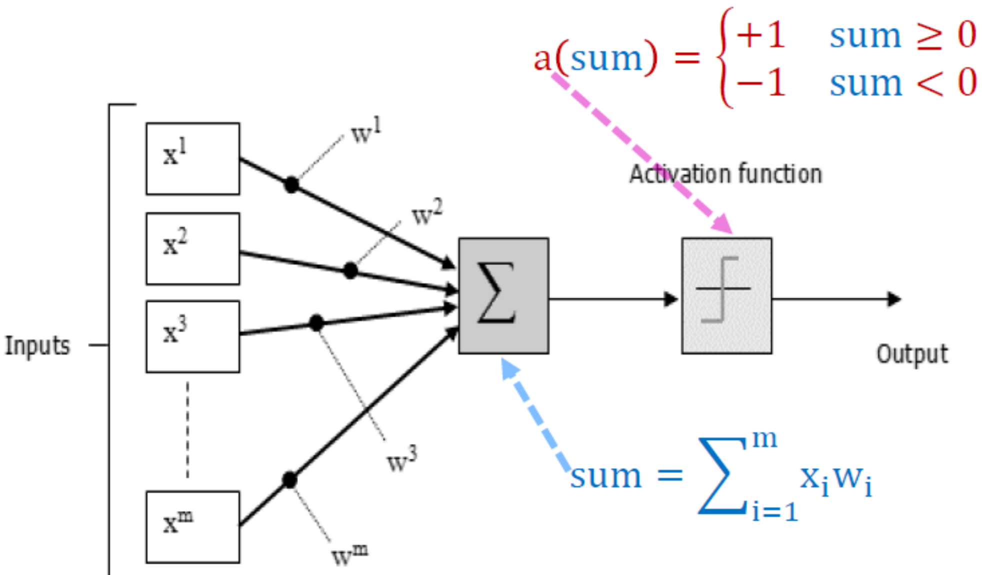
- Trends of DL for IR
- Word embedding
- **Neural network**
- DL for IR/NLP

Neural Network

- MLP
- CNN
  - **Shift/Space invariant**
- Recurrent NN -   [LSTM/GUR]
  - **Time-sensitive**
- Recursive NN
  - **Structure-sensitive**
- Special Case
  - Seq2seq
  - GAN
  - Reinforced Learning

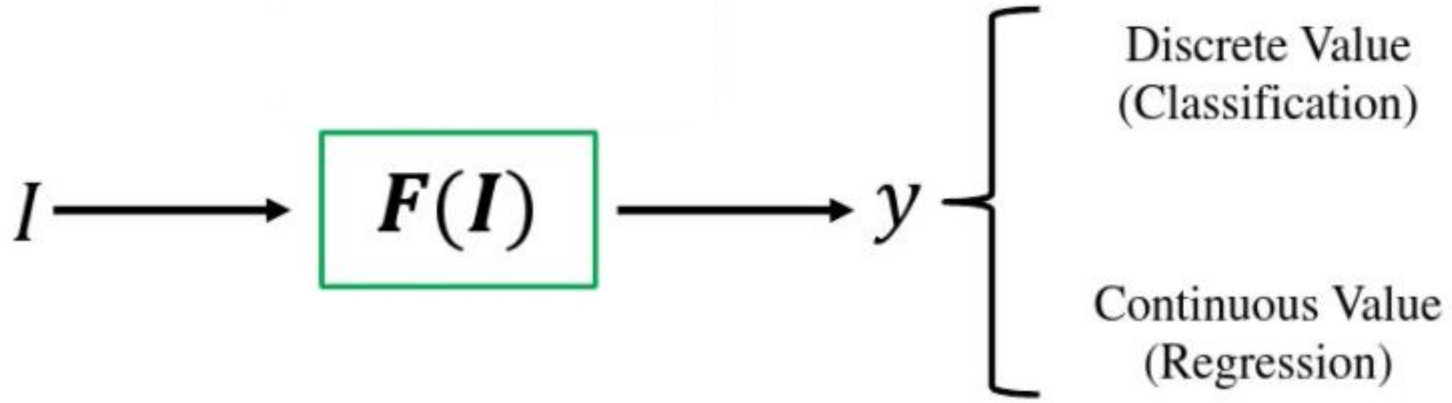
MLP





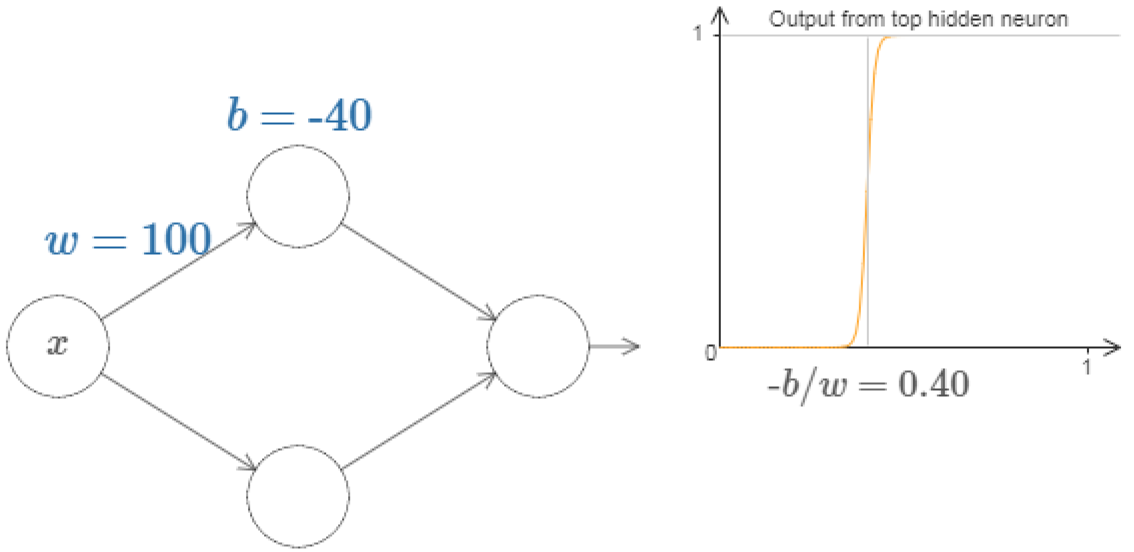
Perception (感知机)

UAT in MLP



Multi-layer Non-linear Mapping - > **U**niversal **A**pproximation **T**heorem

A sample of θ(wx+b)

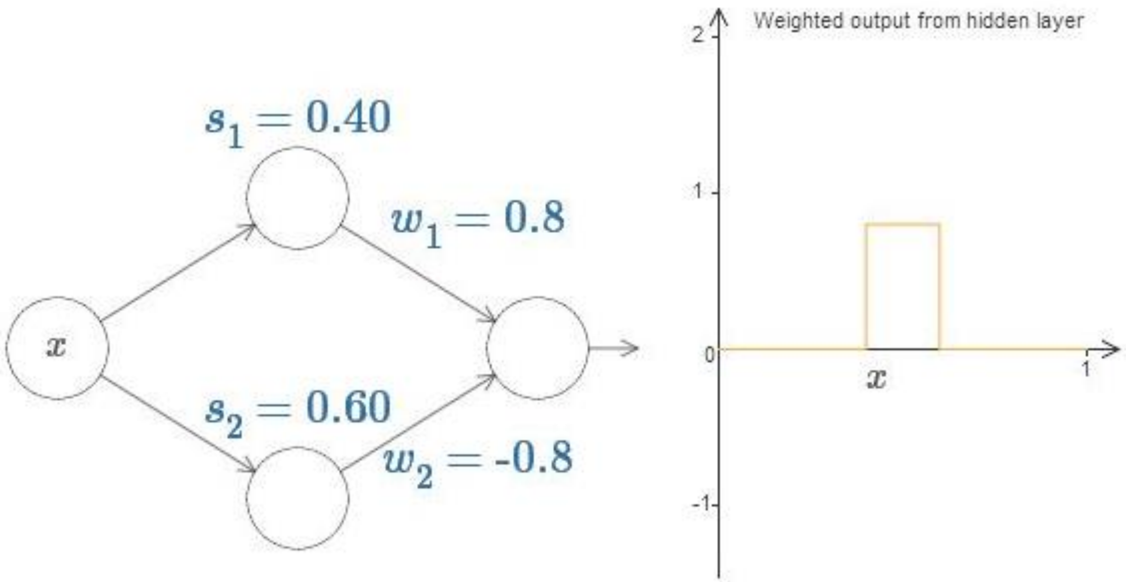


$$s = -b/w.$$

$$\sigma(wx + b), \text{ where } \sigma(z) \equiv 1/(1 + e^{-z})$$

http://neuralnetworksanddeeplearning.com/chap4.html

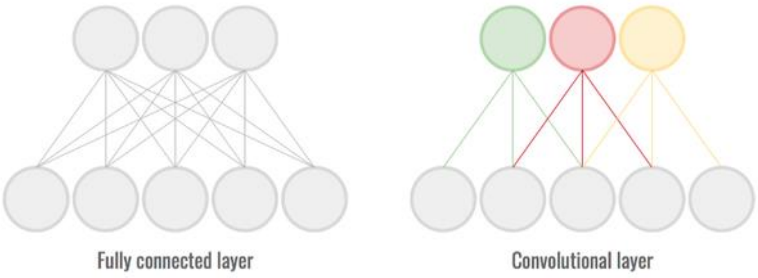
An another sample

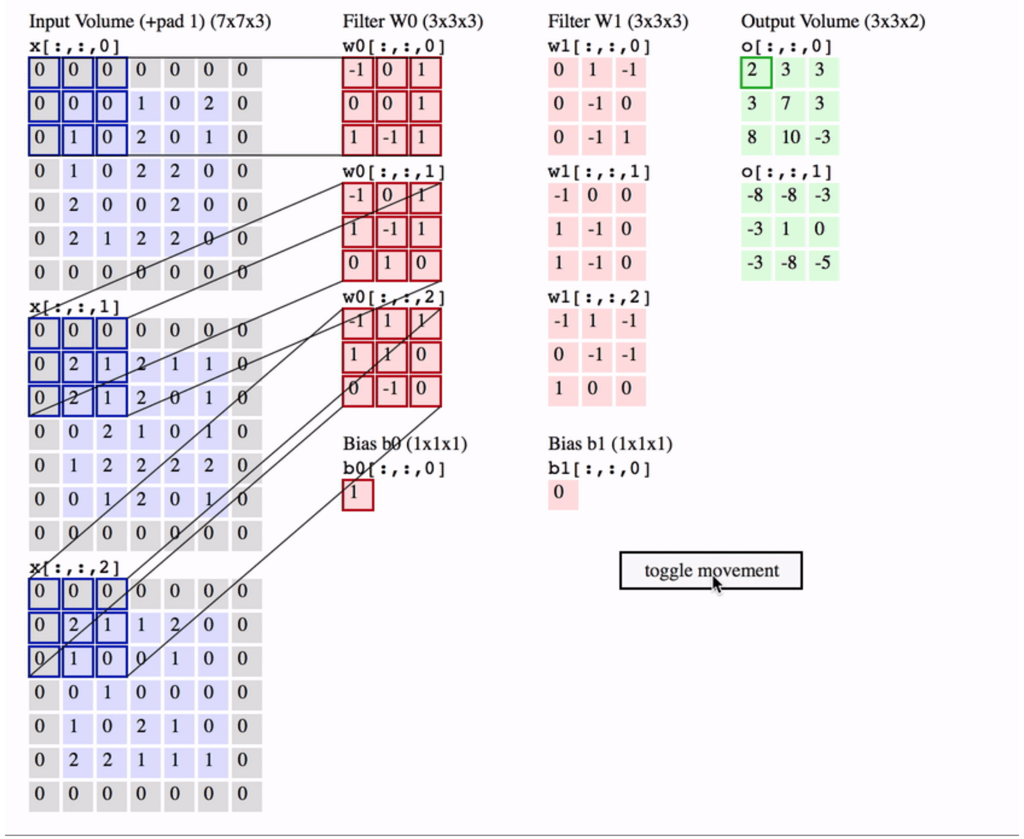


$$\sigma(wx+b), \text{ where } \sigma(z) \equiv 1/(1+e^{-z})$$

From MLP to CNN

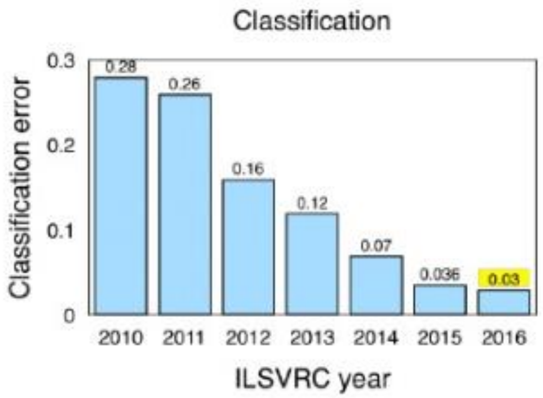
- Local connection
- Shared weight
- Pooling strategy



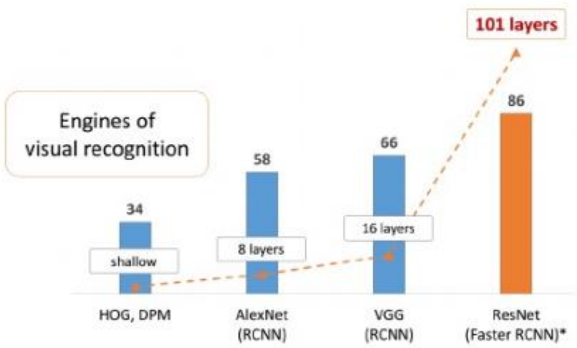


Deep NN in CV

Top 5 error in ImageNet classification

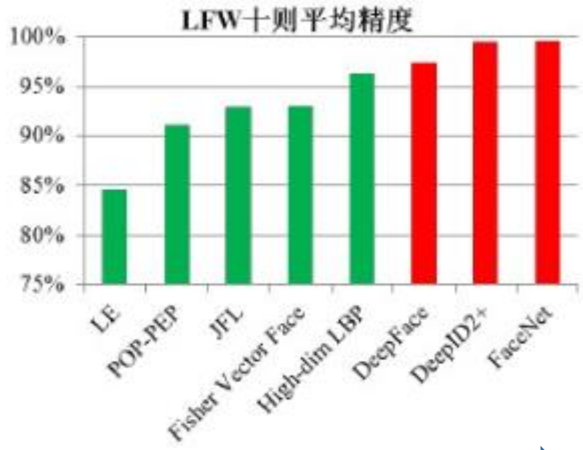


Deep NN



MAP in Pascal VOC visual recognition

10-fold mean precision Face recognition LFW dataset



Credited to Prof. Shiguang Shan with modified

End-2-end in CV

- Tradition CV

- Modern CV： Unsupervised mid-representation

- DNN CV： end-2-end

CNN

- Basic CNN
- Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences[J]. arXiv preprint arXiv:1404.2188, 2014
- Kim CNN
- VDCNN

CNN [kim EMNLP 2014]

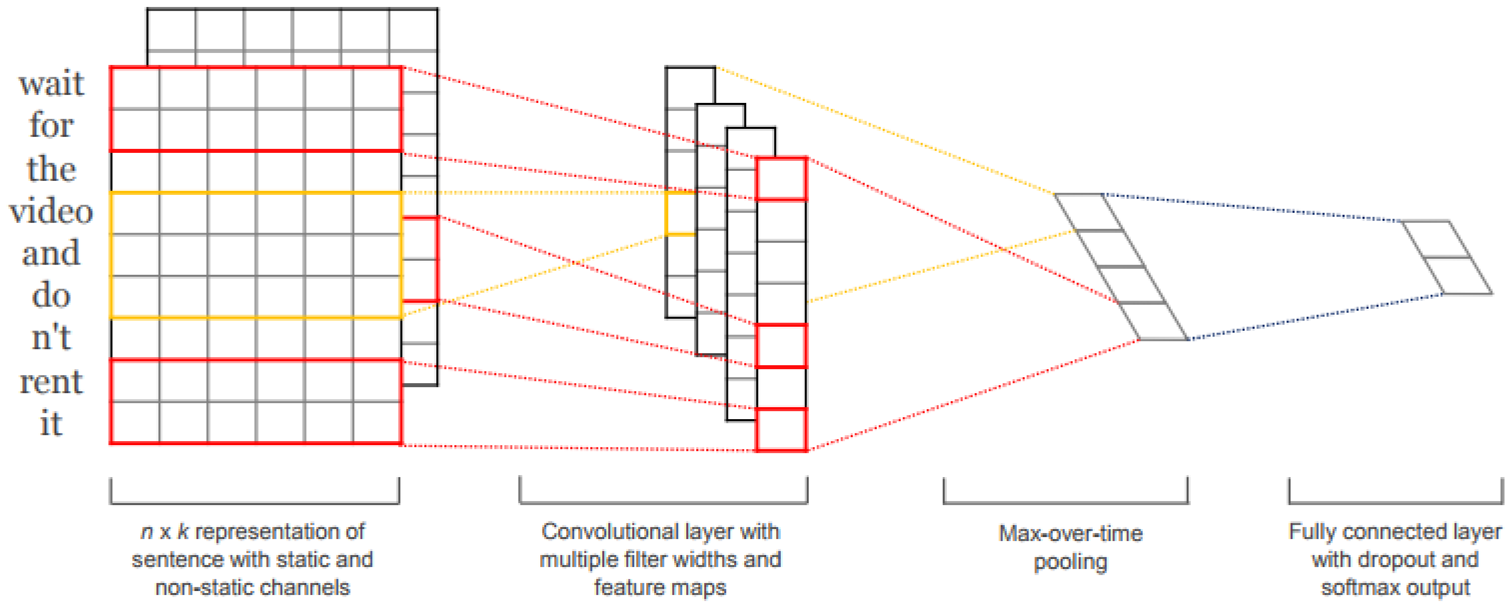


Figure 1: Model architecture with two channels for an example sentence.

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	–	–	–	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	–	–	–	–
RNTN (Socher et al., 2013)	–	45.7	85.4	–	–	–	–
DCNN (Kalchbrenner et al., 2014)	–	48.5	86.8	–	93.0	–	–
Paragraph-Vec (Le and Mikolov, 2014)	–	**48.7**	87.8	–	–	–	–
CCAE (Hermann and Blunsom, 2013)	77.8	–	–	–	–	–	87.2
Sent-Parser (Dong et al., 2014)	79.5	–	–	–	–	–	86.3
NBSVM (Wang and Manning, 2012)	79.4	–	–	93.2	–	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	–	–	**93.6**	–	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	–	–	93.4	–	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	–	–	**93.6**	–	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	–	–	–	–	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	–	–	–	–	–	82.7	–
SVM_S (Silva et al., 2011)	–	–	–	–	**95.0**	–	–

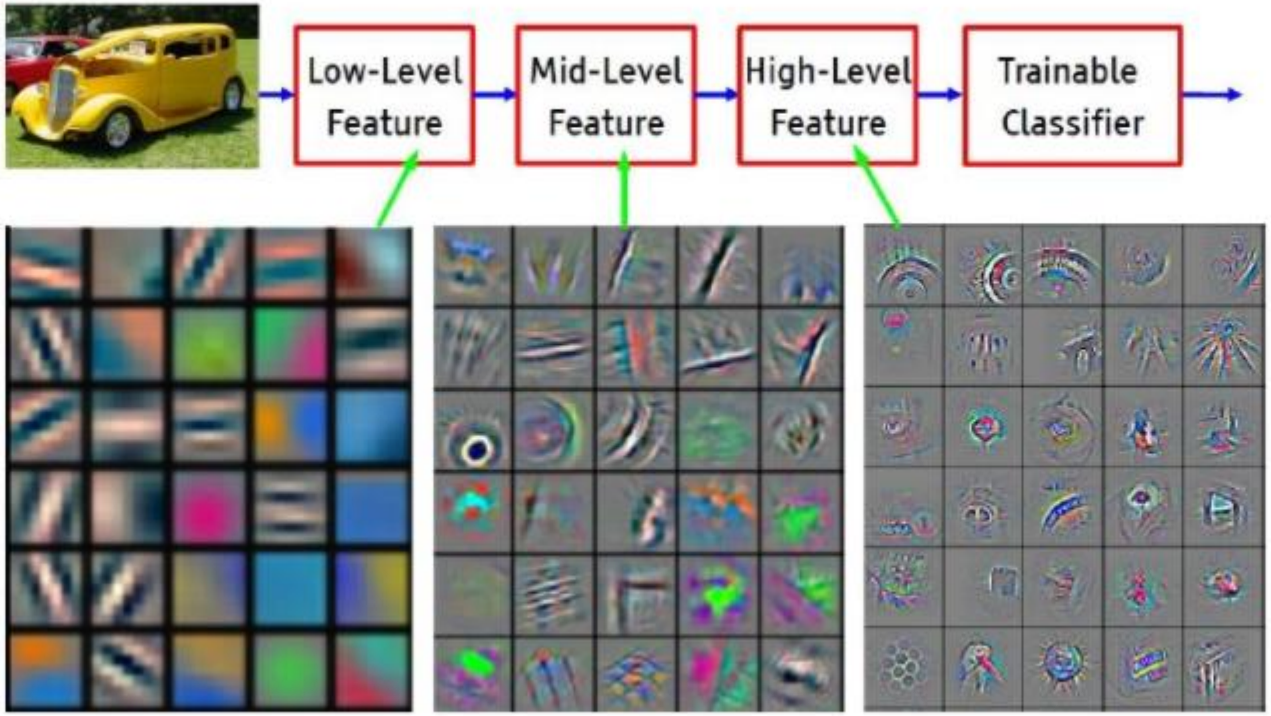
Go deeper or not?

- DEEP
  - Slower
  - Overfitting
    - More Parameters, more data need to feed
  - Hard for convergence
    - Highway network
    - Residual Block
    - Inception

- Shallow: one-layer
  - Fast
  - Less data, es. Fastext.

Go deeper or not?

- Image recognition: Pixel → edge → texton → motif → part → object
- Text: Character → word → word group → clause → sentence → story
- Speech: Sample → spectral band → sound → … → phone → phoneme → word



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

Modified from Prof. LeCun and Prof. Bengio

Very Large CNN [Conneau EACL]

Corpus:	AG	Sogou	DBP.	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
Method	n-TFIDF	n-TFIDF	n-TFIDF	ngrams	Conv	Conv+RNN	Conv	Conv
Author	[Zhang]	[Zhang]	[Zhang]	[Zhang]	[Zhang]	[Xiao]	[Zhang]	[Zhang]
Error	7.64	2.81	1.31	4.36	37.95*	28.26	40.43*	4.93*
[Yang]	-	-	-	-	-	24.2	36.4	-

Table 4: Best published results from previous work. Zhang et al. (2015) best results use a Thesaurus data augmentation technique (marked with an *). Yang et al. (2016)'s hierarchical methods is particularly adapted to datasets whose samples contain multiple sentences.

Depth	Pooling	AG	Sogou	DBP.	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
9	Convolution	10.17	4.22	1.64	5.01	37.63	28.10	38.52	4.94
9	KMaxPooling	9.83	3.58	1.56	5.27	38.04	28.24	39.19	5.69
9	MaxPooling	9.17	3.70	1.35	4.88	36.73	27.60	37.95	4.70
17	Convolution	9.29	3.94	1.42	4.96	36.10	27.35	37.50	4.53
17	KMaxPooling	9.39	3.51	1.61	5.05	37.41	28.25	38.81	5.43
17	MaxPooling	8.88	3.54	1.40	4.50	36.07	27.51	37.39	4.41
29	Convolution	9.36	3.61	1.36	4.35	**35.28**	27.17	37.58	**4.28**
29	KMaxPooling	**8.67**	**3.18**	1.41	4.63	37.00	27.16	38.39	4.94
29	MaxPooling	8.73	3.36	**1.29**	**4.28**	35.74	**26.57**	**37.00**	4.31

Table 5: Testing error of our models on the 8 data sets. No data preprocessing or augmentation is used.

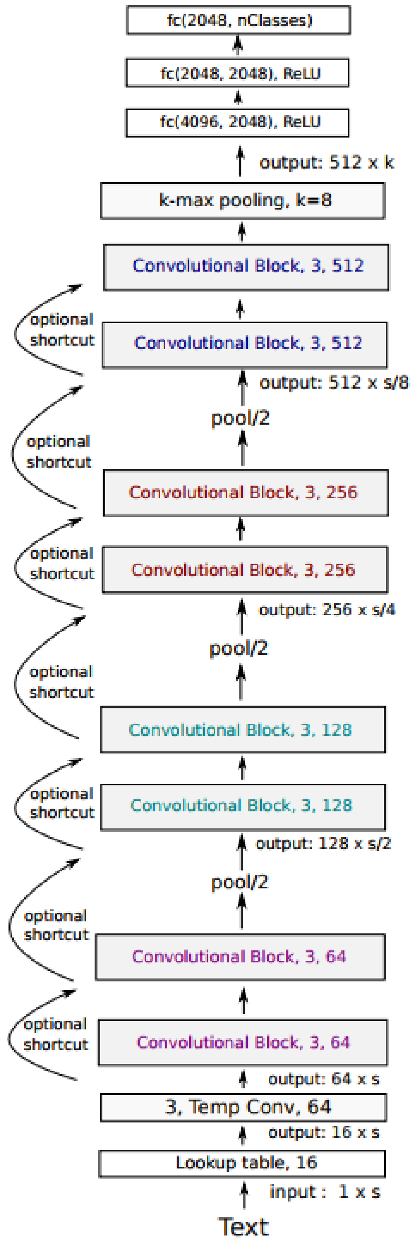


Figure 1: VDCNN architecture.

FASTEX [EACL 2017]

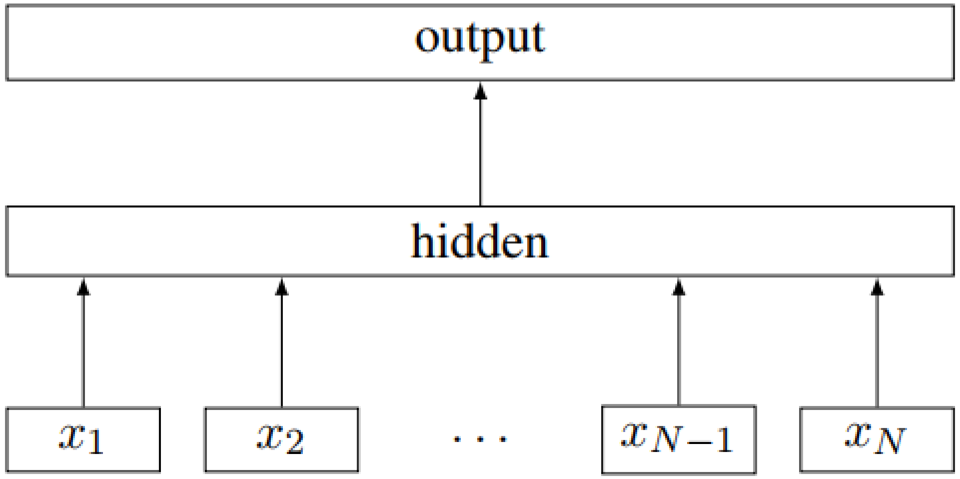
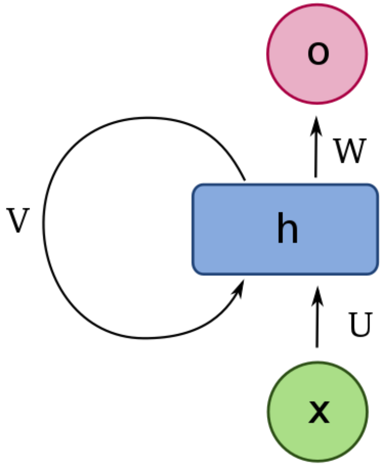


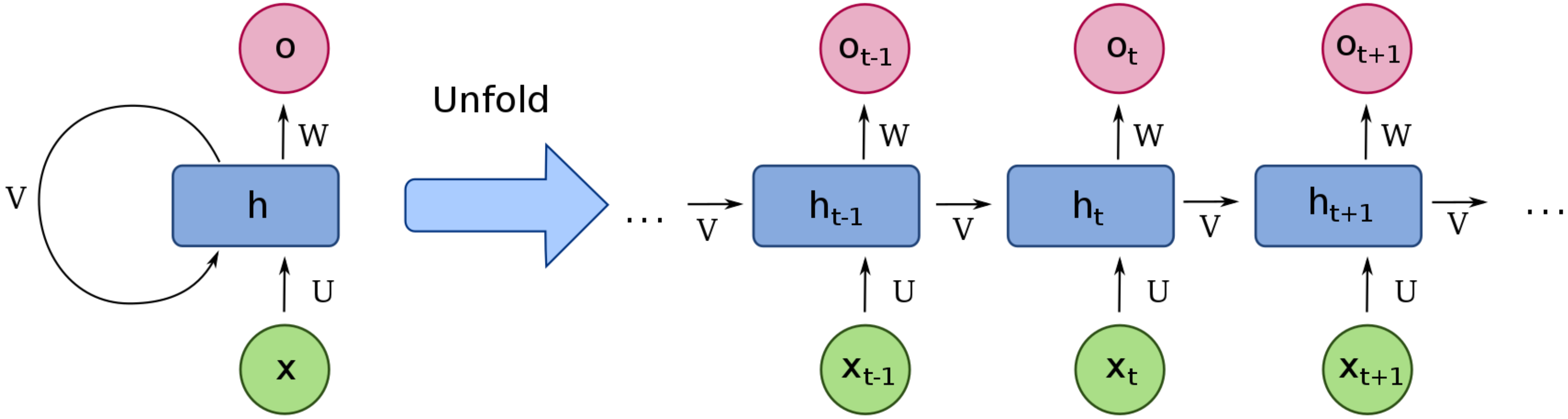
Figure 1: Model architecture of fastText for a sentence with N ngram features $x_1, \ldots, x_N$. The features are embedded and averaged to form the hidden variable.

Model	Yelp'13	Yelp'14	Yelp'15	IMDB
SVM+TF	59.8	61.8	62.4	40.5
CNN	59.7	61.0	61.5	37.5
Conv-GRNN	63.7	65.5	66.0	42.5
LSTM-GRNN	65.1	67.1	67.6	45.3
fastText	64.2	66.2	66.6	45.2

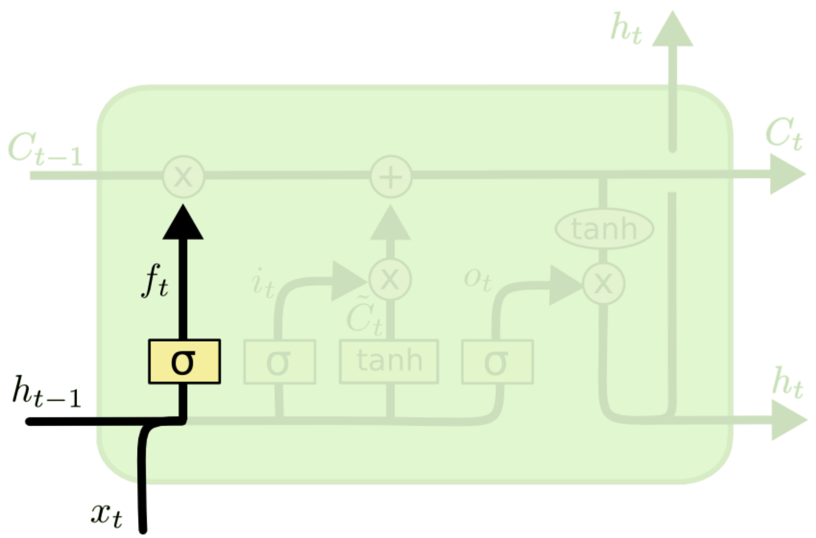
RNN



RNN

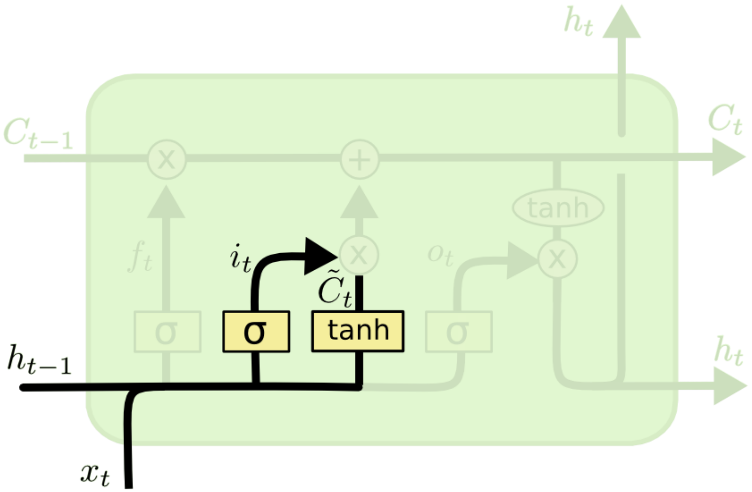


Forget gate



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Input gate



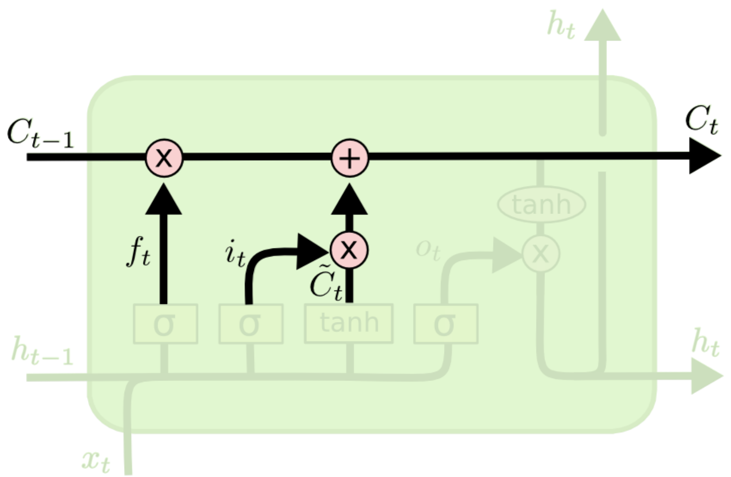
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

replace tanh with **softsign** (not softmax) activation for prevent **overfitting**

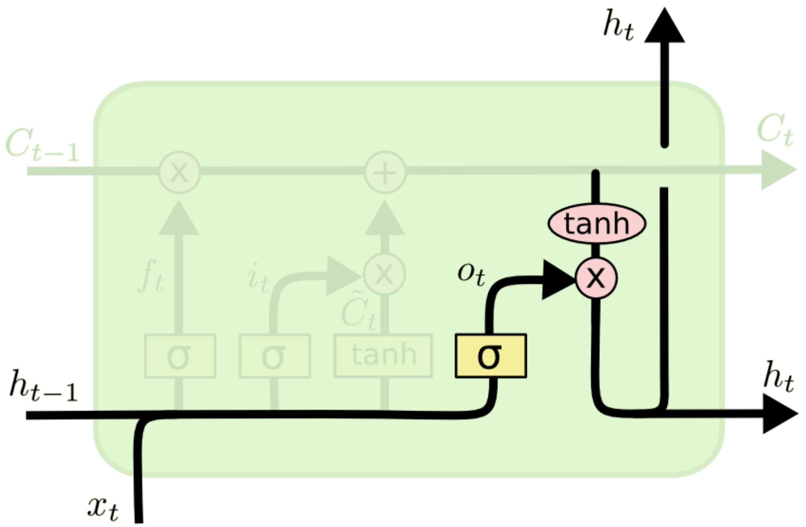
https://zhuanlan.zhihu.com/p/21952042

Forgotten + input



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

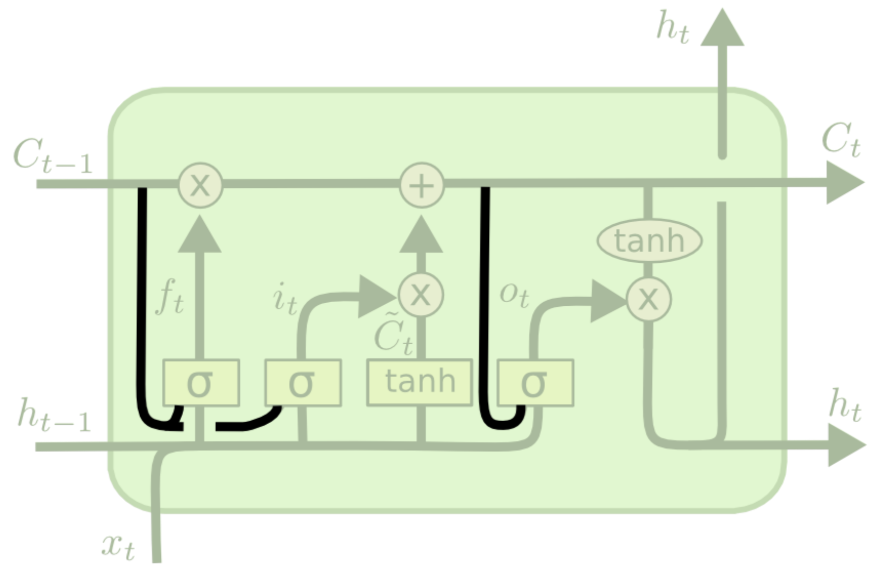
Output Gate



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

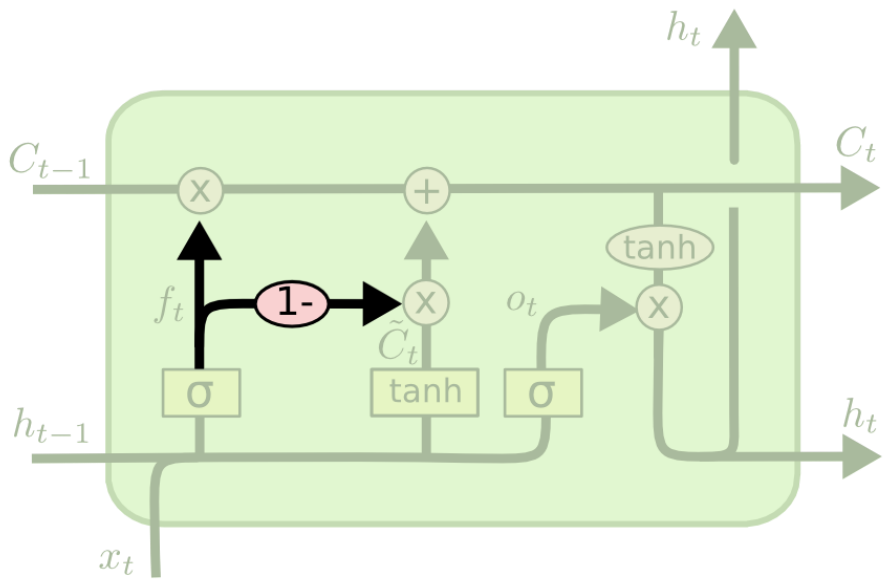
$$h_t = o_t * \tanh\left(C_t\right)$$

LSTM **Variants:** Peephole connections



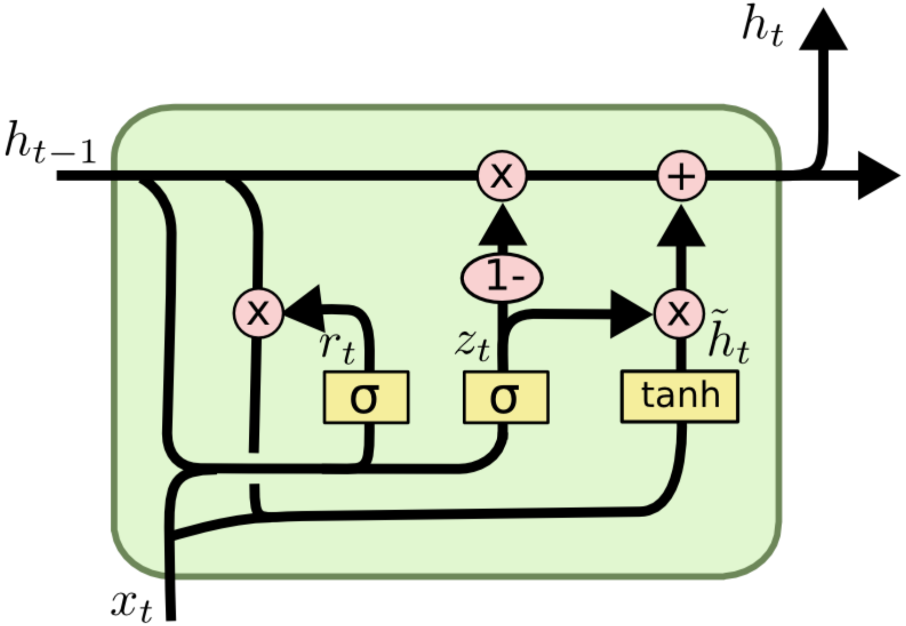
$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

LSTM **Variants:** coupled forget and input gates



$$C_t = f_t * C_{t-1} + (\mathbf{1} - \boldsymbol{f_t}) * \tilde{C}_t$$

LSTM Variants: GRU



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

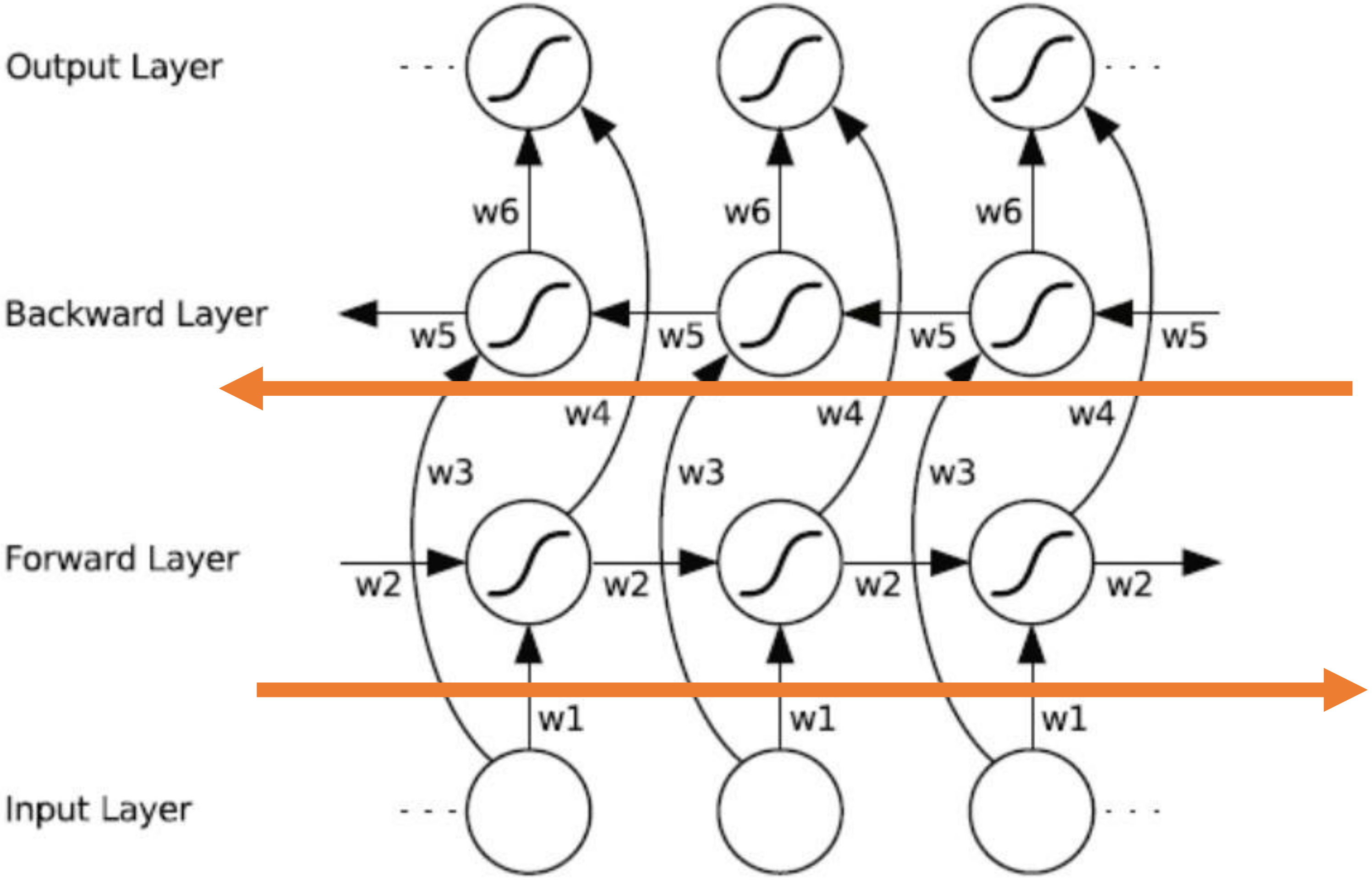
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

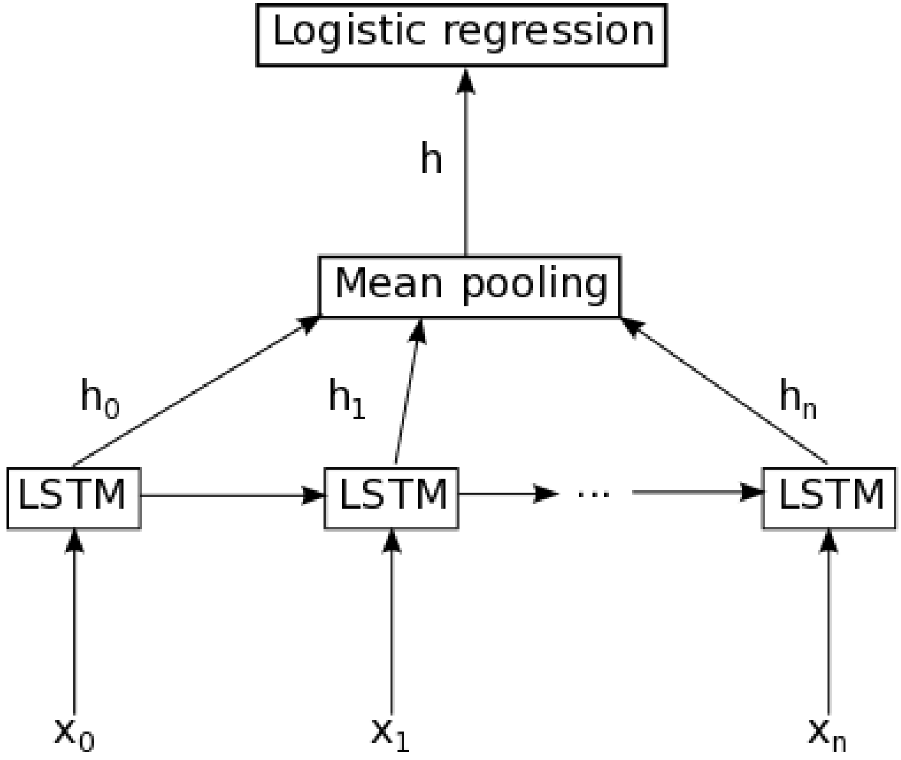
- ✓ Hidden = Cell
- ✓ Forget gate + input gate =1

BiLSTM



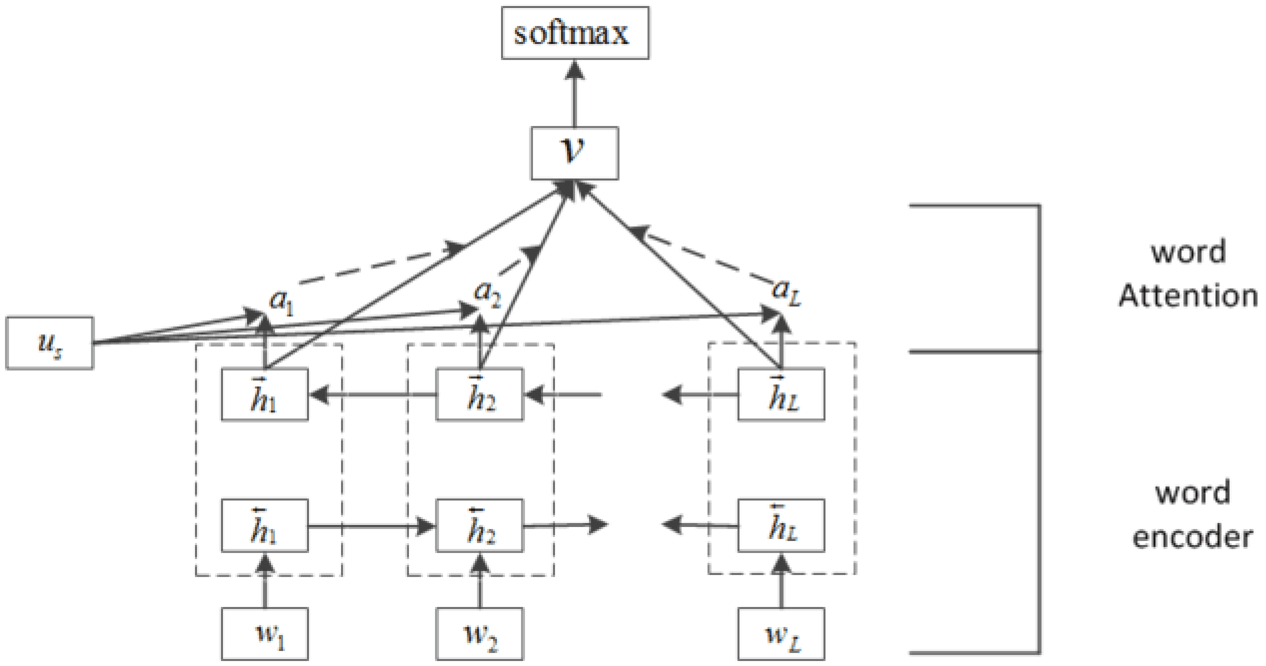
Output Layer
Backward Layer
Forward Layer
Input Layer
w6
w6
w6
w5
w5
w5
w5
w4
w4
w4
w3
w3
w3
w2
w2
w2
w2
w1
w1
w1

Last or Mean?



Logistic regression
h
Mean pooling
h0
h1
hn
LSTM
LSTM
...
LSTM
x0
x1
xn

RNN/LSTM with Attention



BIGRU : 93%

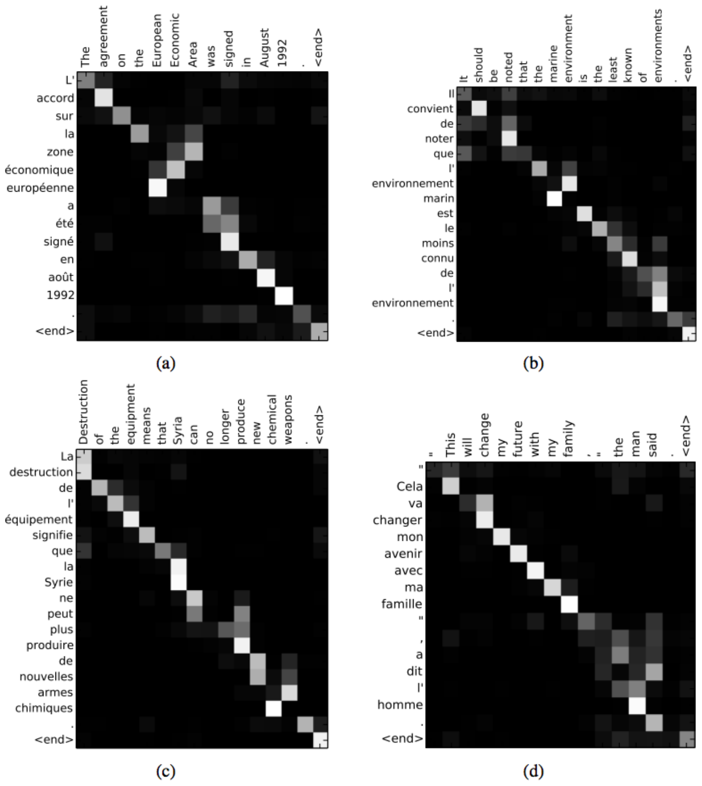
BILSTM : 91.43%

BIGRU_ATTENTION : 95.4%

BILSTM_ATTENTION : 96.2%

https://www.jianshu.com/p/4fbc4939509f

Visualization of Attention in RNN/LSTM



Machine Translation

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.

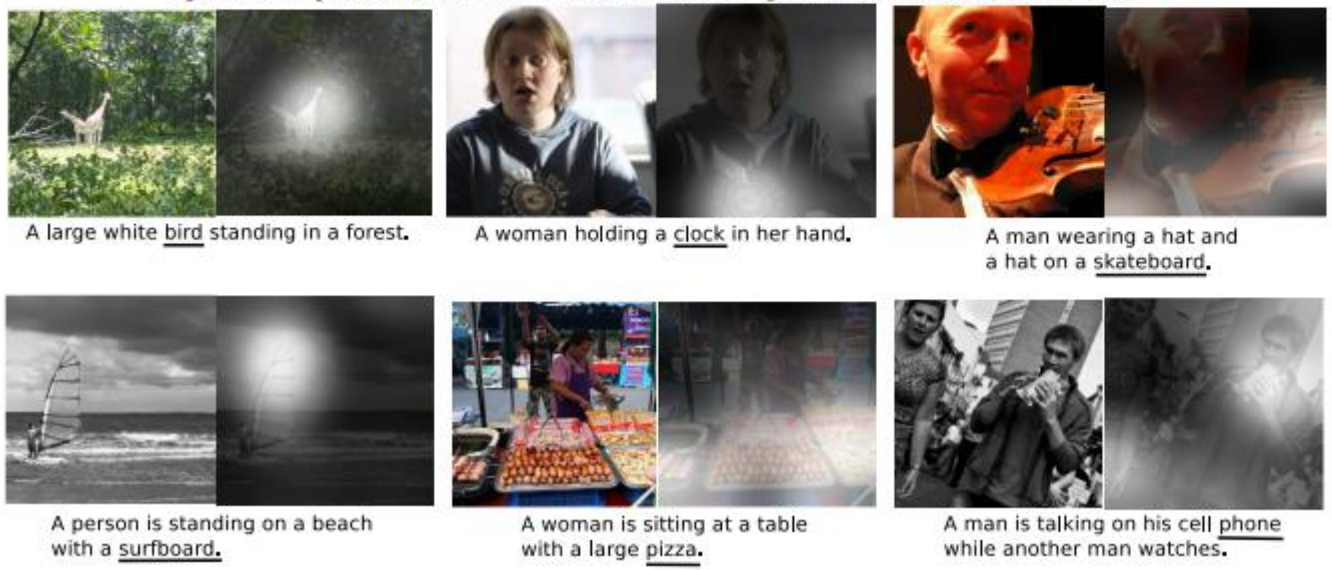
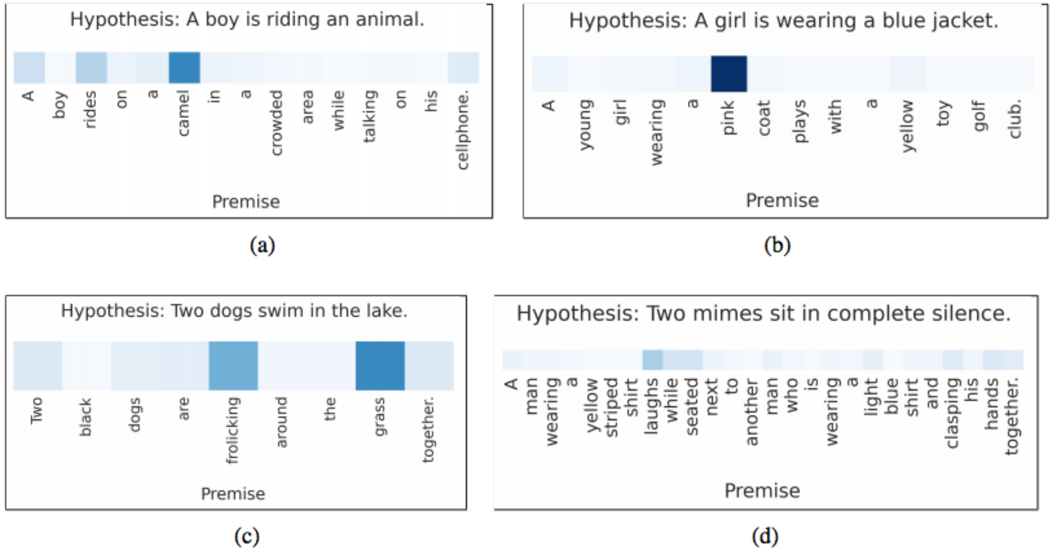
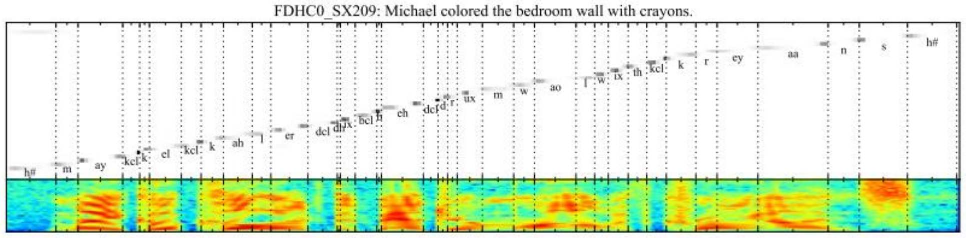


Image Caption

Visualization of Attention in RNN/LSTM

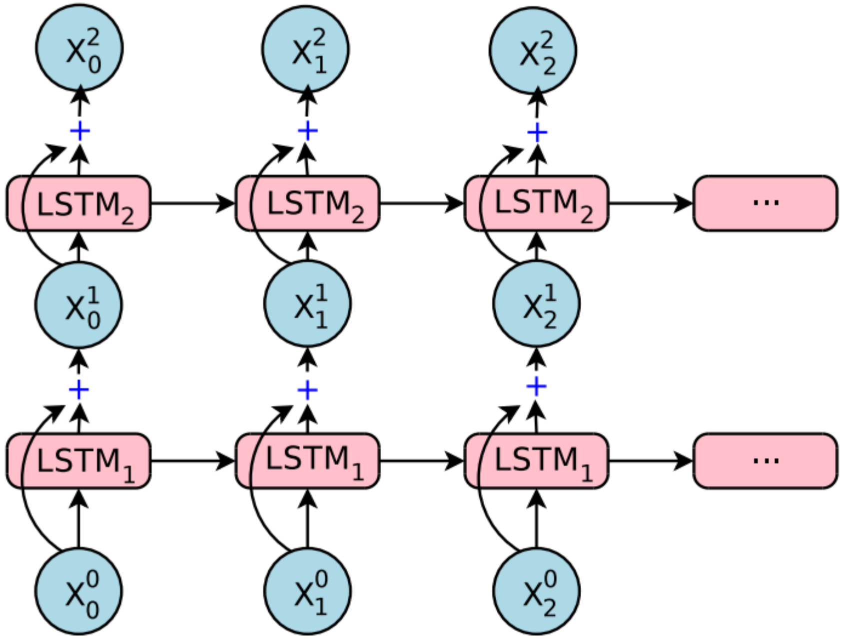


Sematic Entailment



Speech Recognition

Deeper LSTM

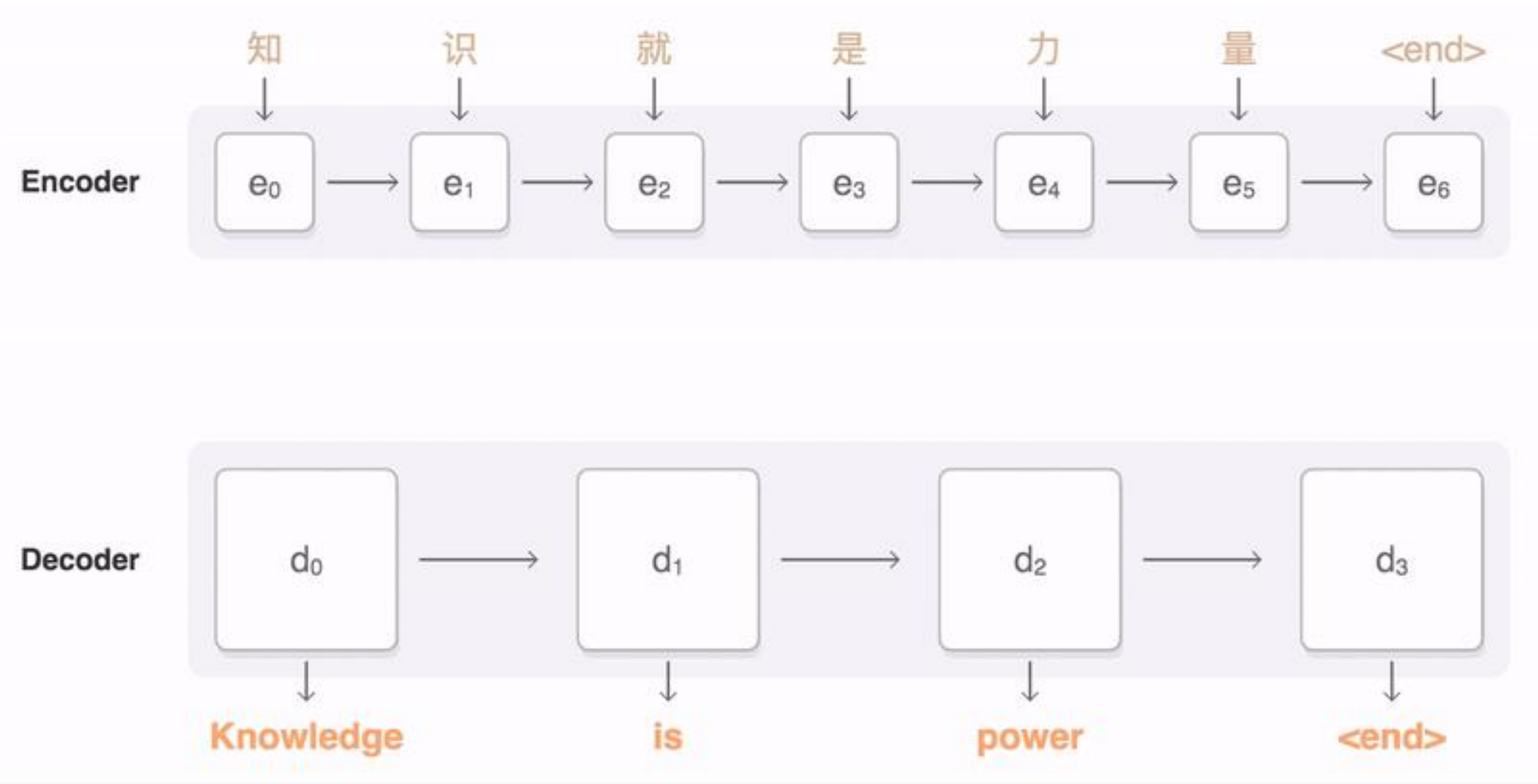


Deep is not necessary, but more feeding data!!!

Background of Neural IR

- Trends of DL for IR
- Word embedding
- Neural network
- **DL for IR/NLP**

Seq2seq



知
识
就
是
力
量
<end>
Encoder
e0
e1
e2
e3
e4
e5
e6
Decoder
d0
d1
d2
d3
Knowledge
is
power
<end>

State-of-art DL models in NLP

- Reading comprehension
  - 536 wiki articles.
  - 10 questions asked by Human

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, Il milione (or, The Million, known in English as the Travels of Marco Polo), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

How did some suspect that Polo learned about China instead of by actually visiting it?

Answer: through contact with Persian traders

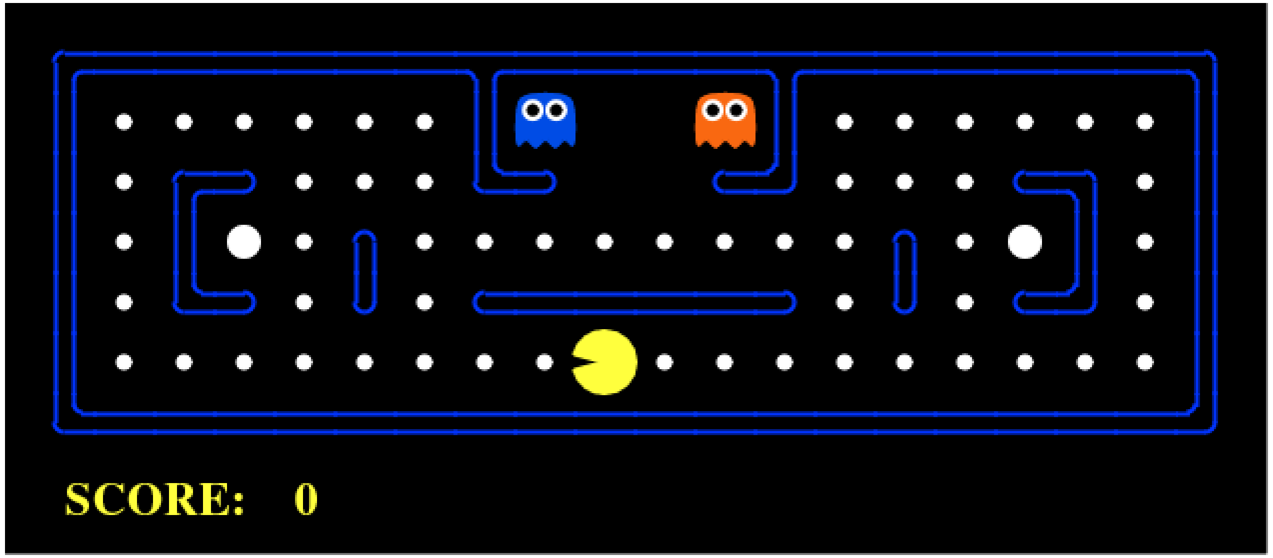
SQuAD1.1 Leaderboard

Since the release of SQuAD1.0, the community has made rapid progress, with the best models now rivaling human performance on the task. Here are the ExactMatch (EM) and F1 scores evaluated on the test set of v1.1.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Oct 05, 2018	BERT (single model) *Google AI Language* https://arxiv.org/abs/1810.04805	85.083	91.835
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
2 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.954	91.677
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490
4 Jul 08, 2018	r-net (ensemble) *Microsoft Research Asia*	84.003	90.147
5 Mar 19, 2018	QANet (ensemble) *Google Brain & CMU*	83.877	89.737
5 Sep 09, 2018	nlnet (single model) *Microsoft Research Asia*	83.468	90.133
5 Jun 20, 2018	MARS (ensemble) *YUANFUDAO research NLP*	83.982	89.796

https://rajpurkar.github.io/SQuAD-explorer/

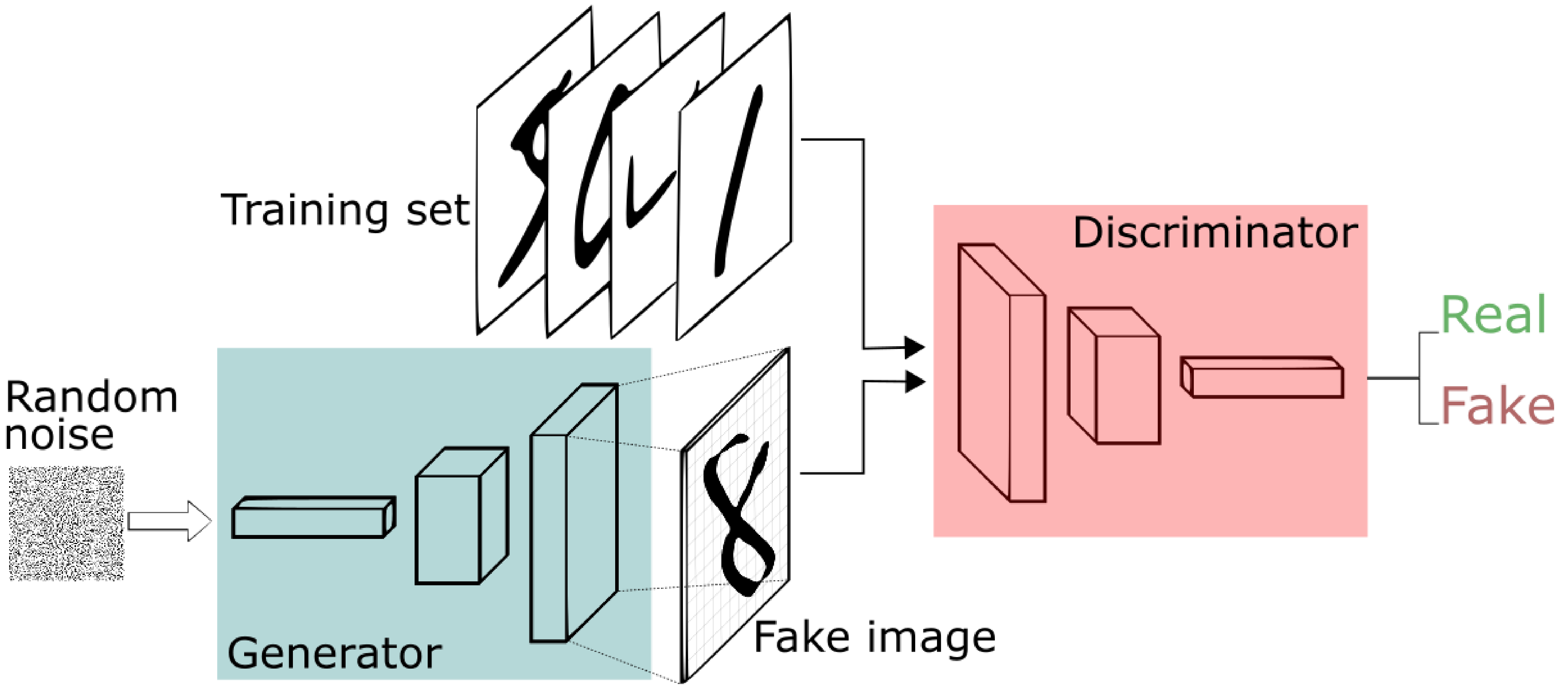
Reinforced learning



Compared to the supervised learning:
You can not know the current reward from the current action, namely a delayed reward,
only in the case that the game is finished.

https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html

GAN


Training set
Random noise
Generator
Fake image
Discriminator
Real
Fake