CV

# Few-Shot Learning

College of Information Science and Engineering
Ocean University of China

http://vision.ouc.edu.cn

Chenchen Qiu

# CONTENT

01

# Introduction to Few-shot Learning

In this part, I will briefly introduce the fundamental concept of few-shot learning, especially few-shot and one-shot image classification.

# Comparison

**Deep Learning**

massive amounts of data
+
GPUs

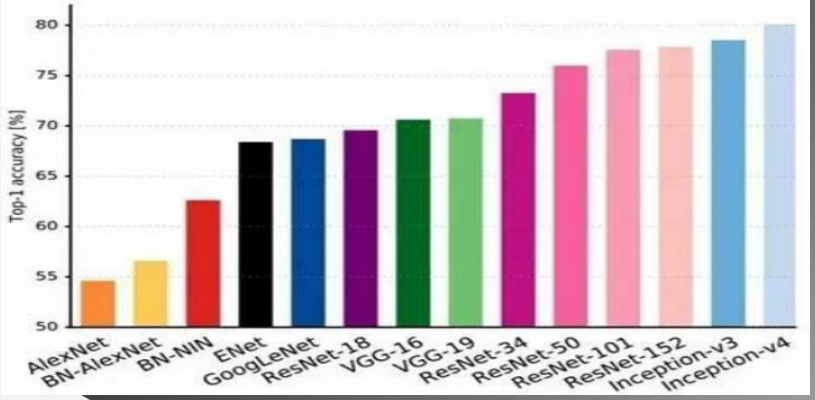
**Few-shot Learning**

Training of machine learning algorithms using a very small set of training data


**Zero-shot Learning**

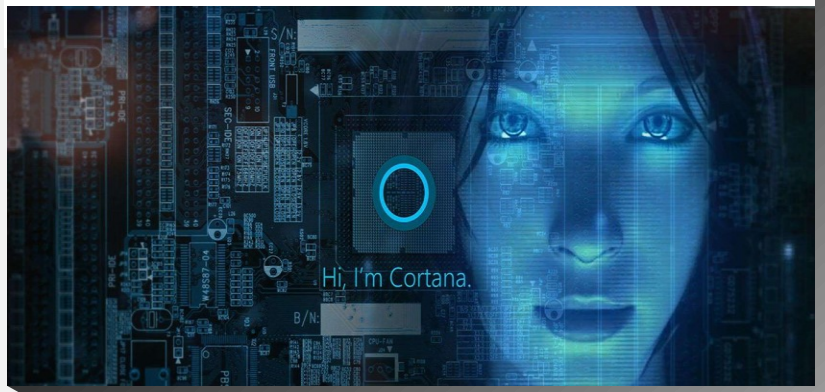
Solving a task despite not having received any training examples of that task

## ImageNet ILSVRC

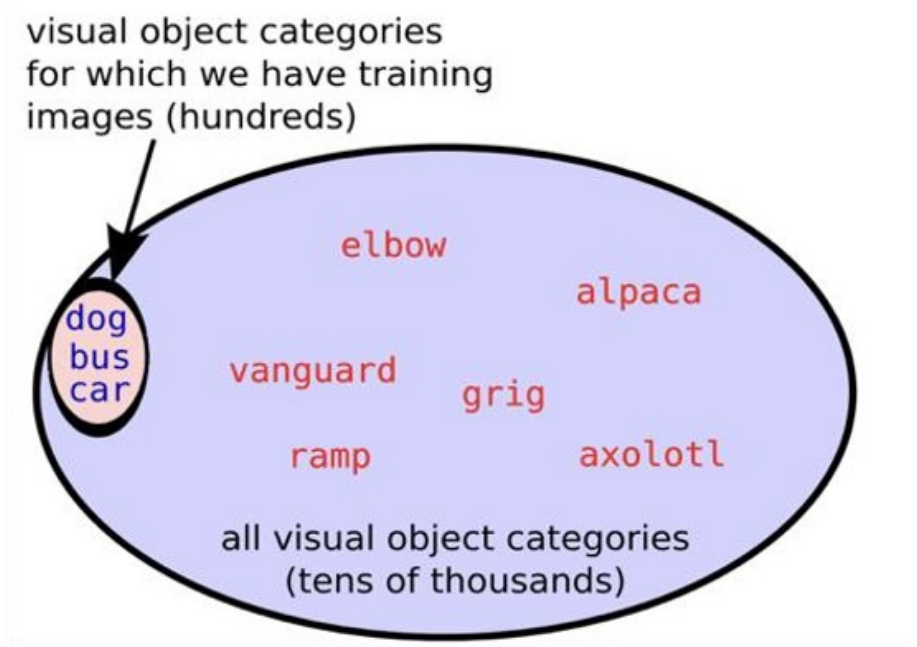



## Face Verification


## Voice Assistant



# Dilemma of Large-scale Supervised Recognition

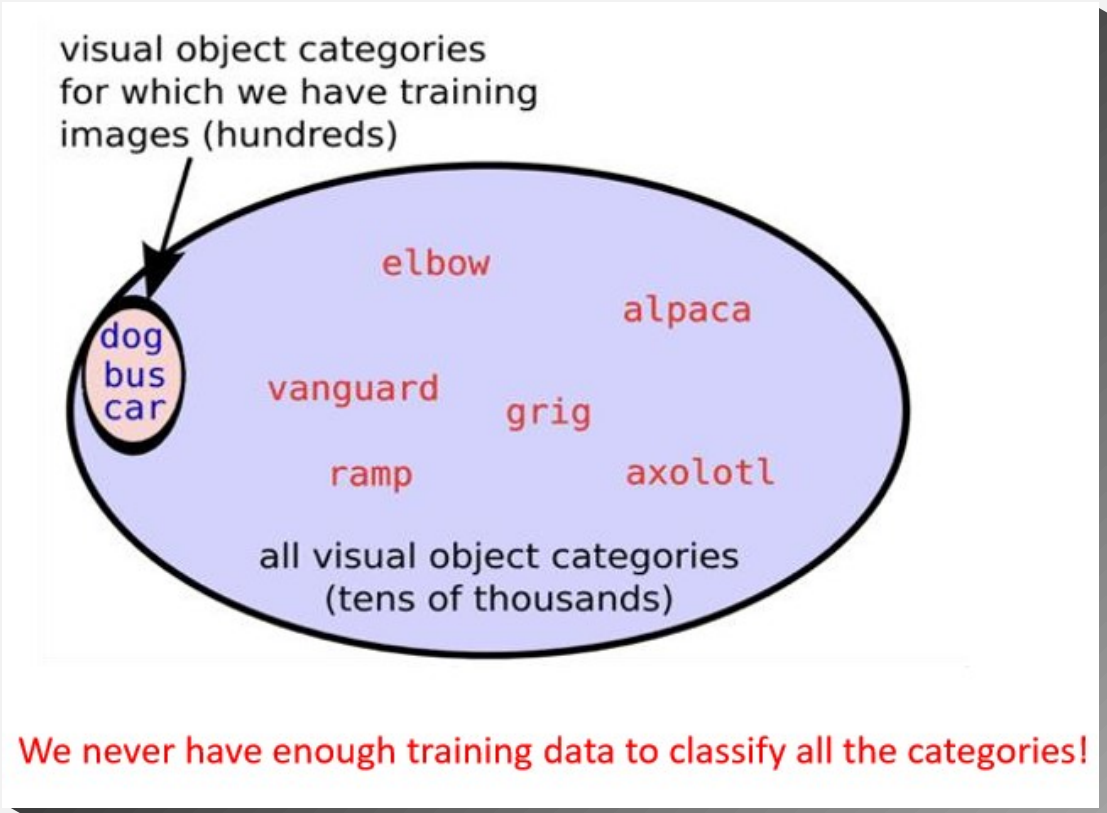



We never have enough training data to classify all the categories!

## Problems：

- **No memory: Knowledge learned is not retained**
- **Needs a large number of training examples**

# Dilemma of Large-scale Supervised Recognition




**What we want? Learn as humans do!**

- Humans have the ability to recognize without seeing examples (zero-shot learning);
- Retain learned knowledge from previous tasks & use it to help future learning(transfer learning);

# Our Goal

One-shot learning aims to learn information about object categories from one, or only a few, training images.

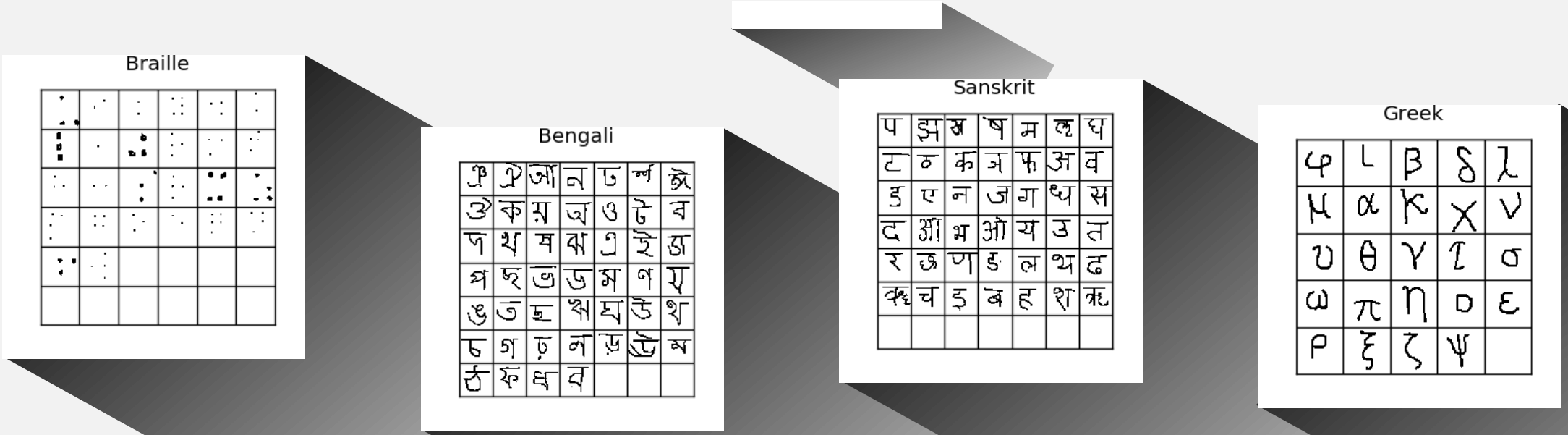
# Dataset

Omniglot

Minilmagenet

Animals with Attributes2

# Omniglot

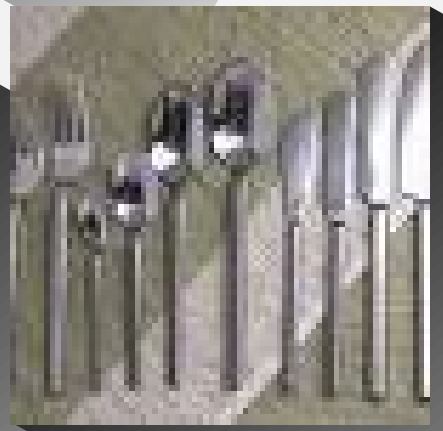
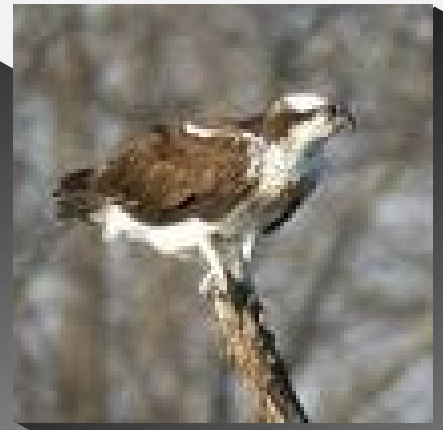



- This dataset contains 1623 different handwritten characters from 50 different alphabets.
- Each of the 1623 characters was drawn online via Amazon's Mechanical Turk by 20 different people.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. Science, 2015

# MiniImageNet



- The miniImageNet test used 60,000 images with 100 classes, each having 600 examples.
- 80 classes were used for training, and testing was done on the other 20.

L. Fei-Fei and O. Russakovsky, Analysis of Large-Scale Visual Recognition, Bay Area Vision Meeting, 2013

# Animals with Attributes2

| polar bear | |
|---|---|
| black: | no |
| white: | yes |
| brown: | no |
| stripes: | no |
| water: | yes |
| eats fish: | yes |

- This dataset provides a platform to benchmark transfer-learning algorithms, in particular attribute base classification and zero-shot learning .

- It consists of 37322 images of 50 animals classes.

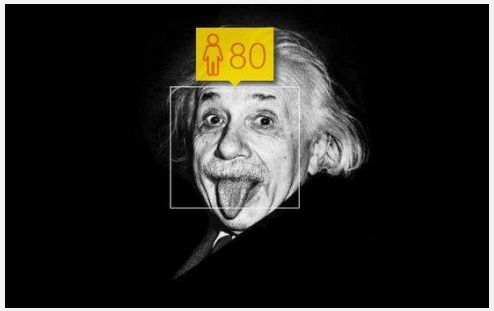
Y. Xian, C. H. Lampert, B. Schiele, Z. Akata. "Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly" arXiv:1707.00600

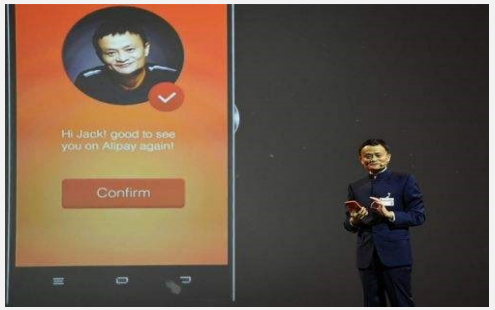
02

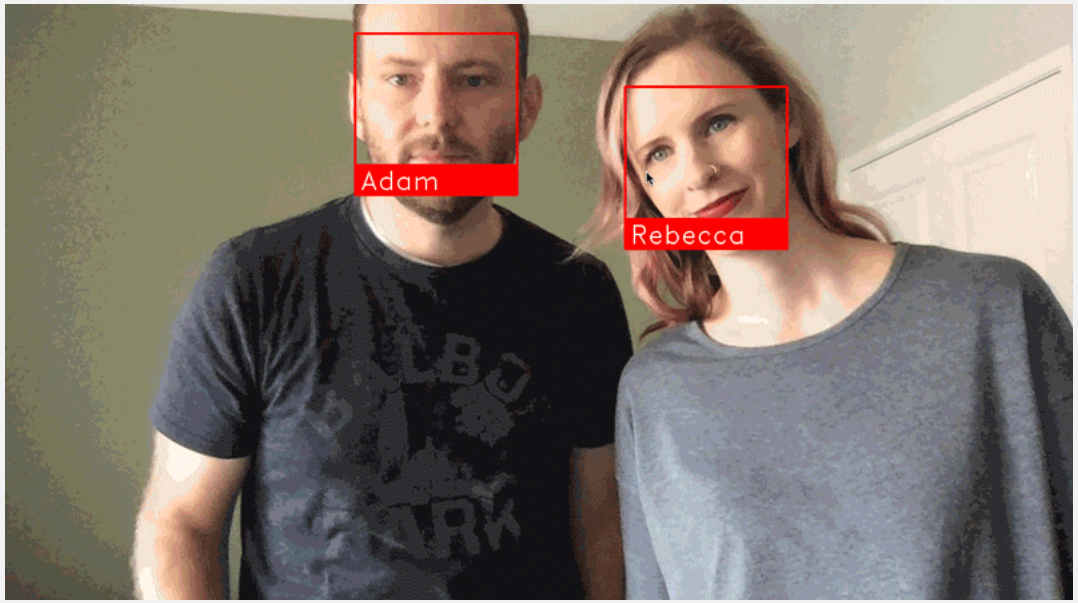
# Popular methods to Few-shot Learning

In this part, I shall describe some of the papers that I gathered.

# Face Recognition





Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In CVPR, 2014.

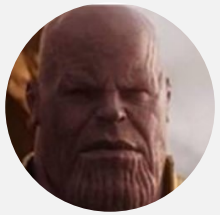
# Face Recognition

Bruce Banner

Ironman

Scarlet Witch

Black Widow



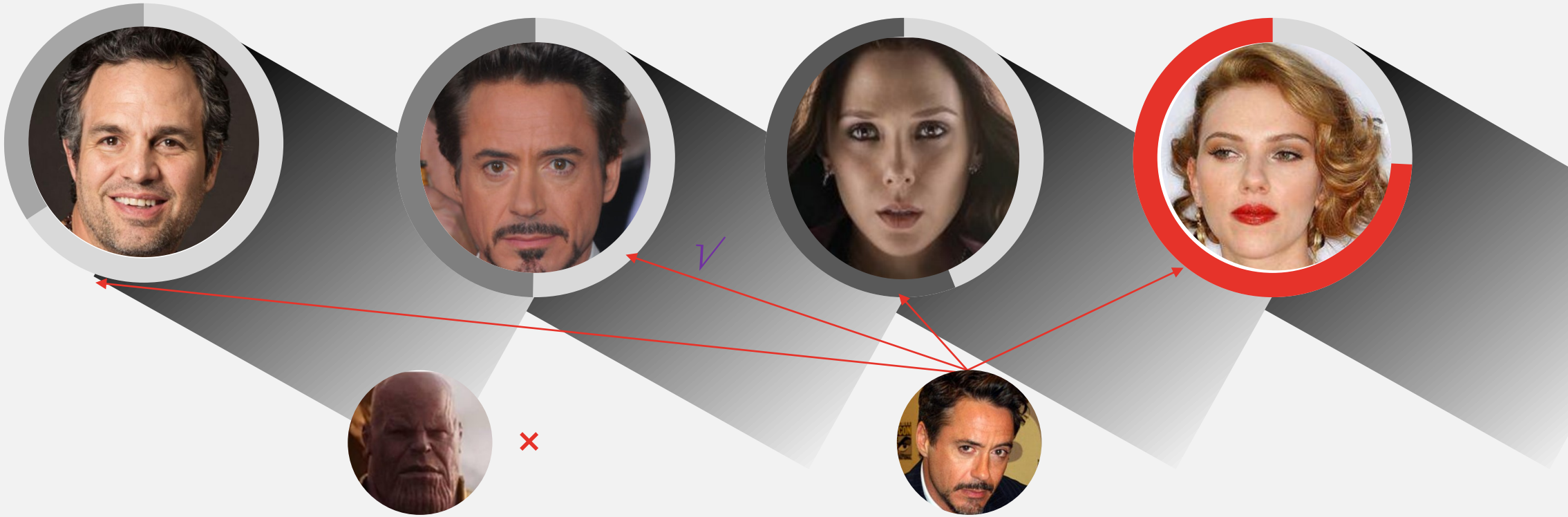
Learning from one example to recognize the person again.

# Learning a “similarity” function

d (img1,img2)=degree of difference between images

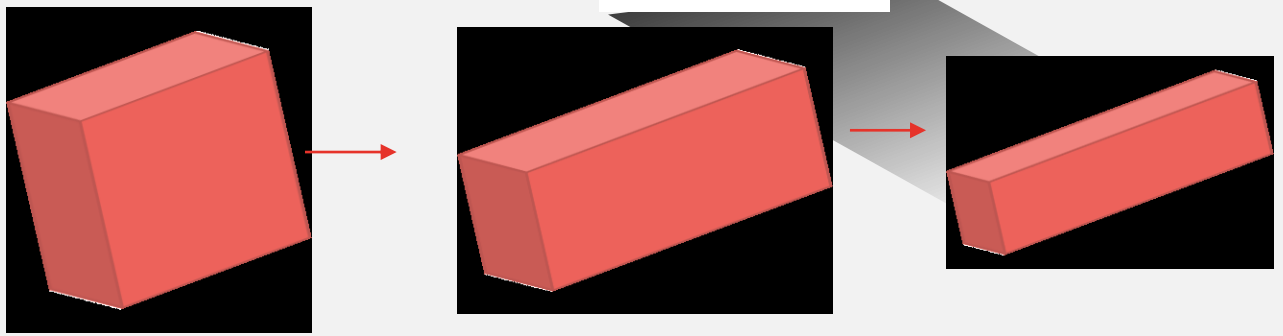
If d(img1,img2) ≤τ same
>τ different


***Face verification***

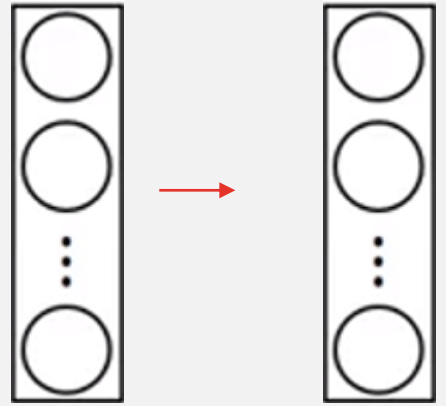


# Siamese Network

$x^{(1)}$ Ironman

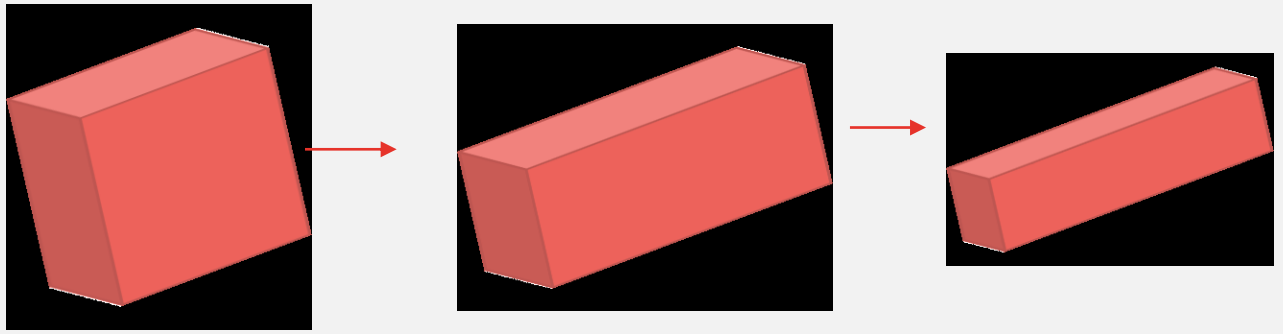


128

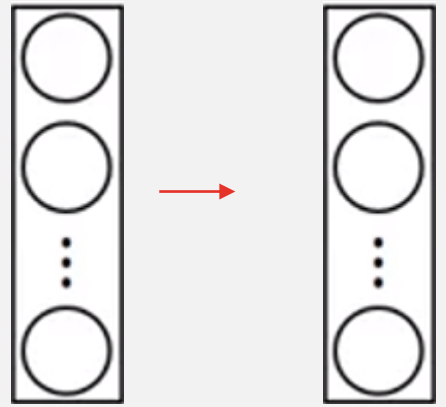


$x^{(1)}$

$x^{(2)}$ Black Widow



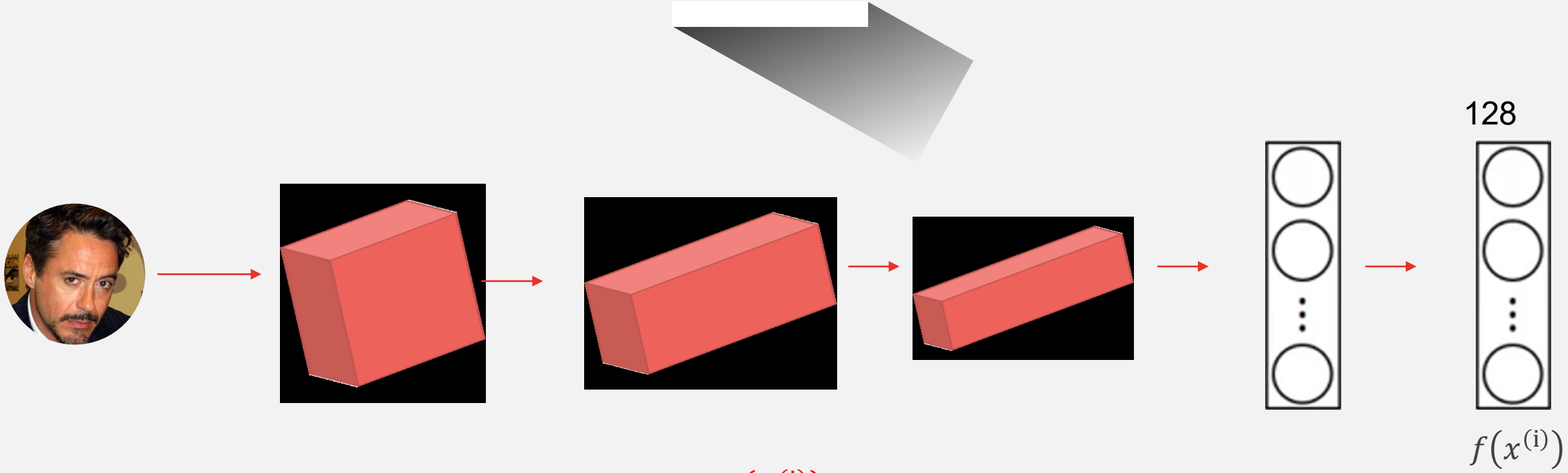
128



$$d(x^{(1)}, x^{(2)}) = ||f(x^{(1)}) - f(x^{(2)})||_d{}^2$$

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In CVPR, 2014.

# Goal of learning




- Parameters of NN define an encoding $f(x^{(i)})$
- Learn parameters so that:

  If $x^{(i)}, x^{(j)}$ are the person, $\|f(x^{(i)}) - f(x^{(j)})\|^2$ is small.

  If $x^{(i)}, x^{(j)}$ are different persons, $\|f(x^{(i)}) - f(x^{(j)})\|^2$ is large.

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In CVPR, 2014.

# Face Recognition


**Goal**: $\|f(A) - f(P)\|^2 \leq \|f(A) - f(N)\|^2$

$$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 \leq 0$$

$$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 \leq \alpha$$

F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In CVPR, 2015.

# Loss Function


$$L(A, P, N) = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0)$$

$$J = \sum_{i=1}^{m} L\left(A^{(i)}, P^{(i)}, N^{(i)}\right)$$

Training set： 10k pictures of 1k persons

F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In CVPR, 2015.

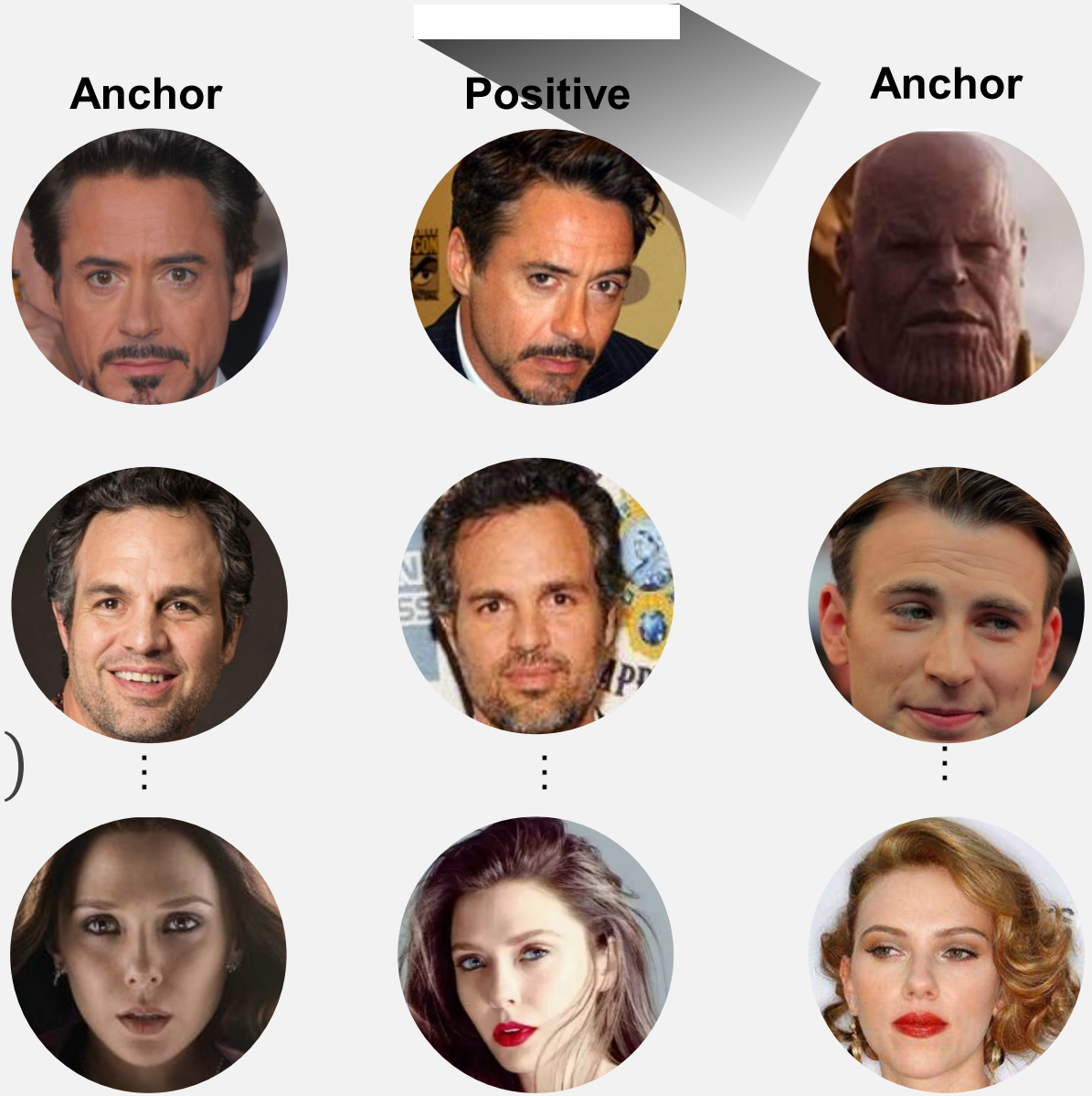
# Choosing the triplets A,P,N


During training, if A, P, N are chosen randomly,
$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha \leq 0$ is easily satisfied.

$$d(A, P) + \alpha \leq \mathrm{d}(A, N)$$

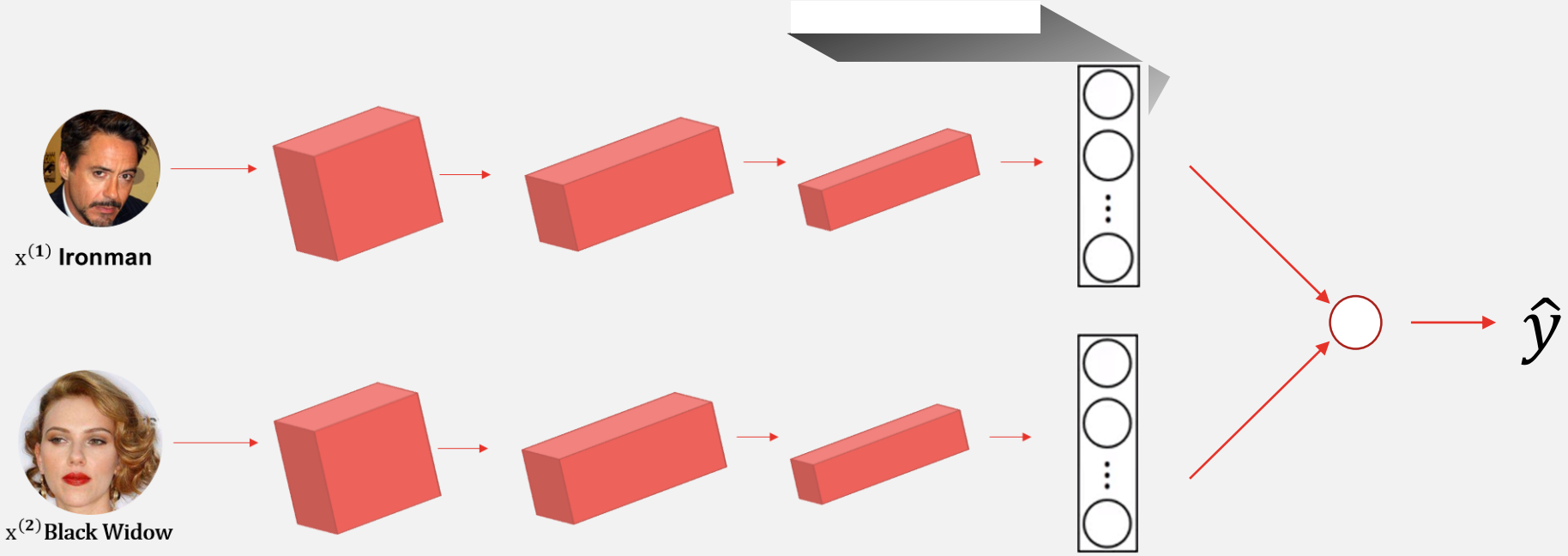
F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In CVPR, 2015.

# Training set using triplet loss




$$J = \sum_{i=1}^{m} L\left(A^{(i)}, P^{(i)}, N^{(i)}\right)$$

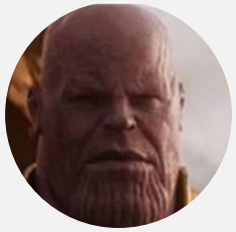
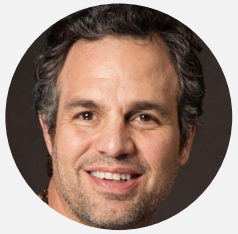
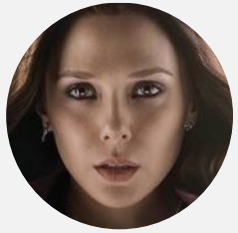
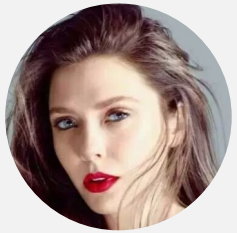
# Learning the similarity function




$$y = \sigma\left(\sum_{k=1}^{128} w_i \left| f\left(x^{(i)}\right)_k - f\left(x^{(j)}\right)_k \right| + b\right)$$

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In CVPR, 2014.

# Face verification supervised learning

| x | | y |
|---|---|---|
|  |  | 1 |
|  |  | 0 |
|  |  | 1 |
|  |  | 1 |

# Outlines of One-shot Learning Methods

**Directly Supervised approaches**

**Transfer Learning Approaches**

Do not use auxiliary data

Directly learn one-shot classier

Use knowledge from auxiliary data

The paradigm of meta-learning

Fu et al. Recent Advances in Zero-Shot Recognition: Toward Data-Efficient Understanding of Visual Content. In SPM, 2018.

# Directly supervised learning-based approaches

## Instance-based learning

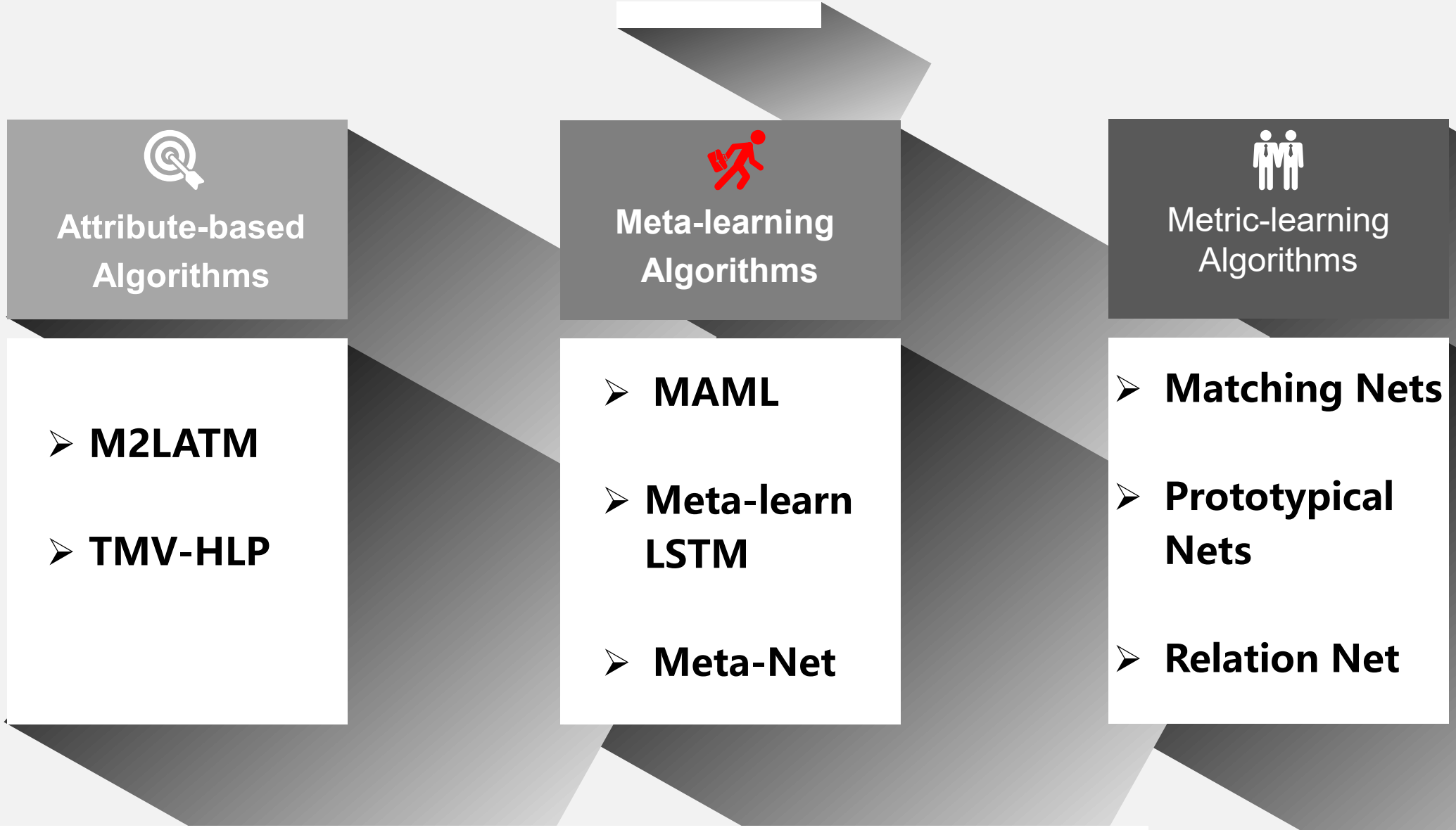
- K-nearest neighbor

### Non-parametric methods

- Fei-Fei et al. A Bayesian approach to unsupervised one-shot learning of object categories. In *CVPR*, 2003.
- Fei-Fei et al. One-shot learning of object categories. *TPAMI*, 2006.

Fu et al. Recent Advances in Zero-Shot Recognition: Toward Data-Efficient Understanding of Visual Content. In *SPM*, 2018.

# Transfer learning-based approaches




Fu et al. Recent Advances in Zero-Shot Recognition: Toward Data-Efficient Understanding of Visual Content. In *SPM*, 2018.

# Data Augmentation for One-shot Learning

01 Learning one-shot models by utilizing the **manifold information** of large amount of unlabeled data in a semi-supervised or transductive setting

02 Adaptively learning the one-shot classifiers from off-shelf trained models

03 Borrowing examples from relevant categories or semantic vocabularies to augment training set

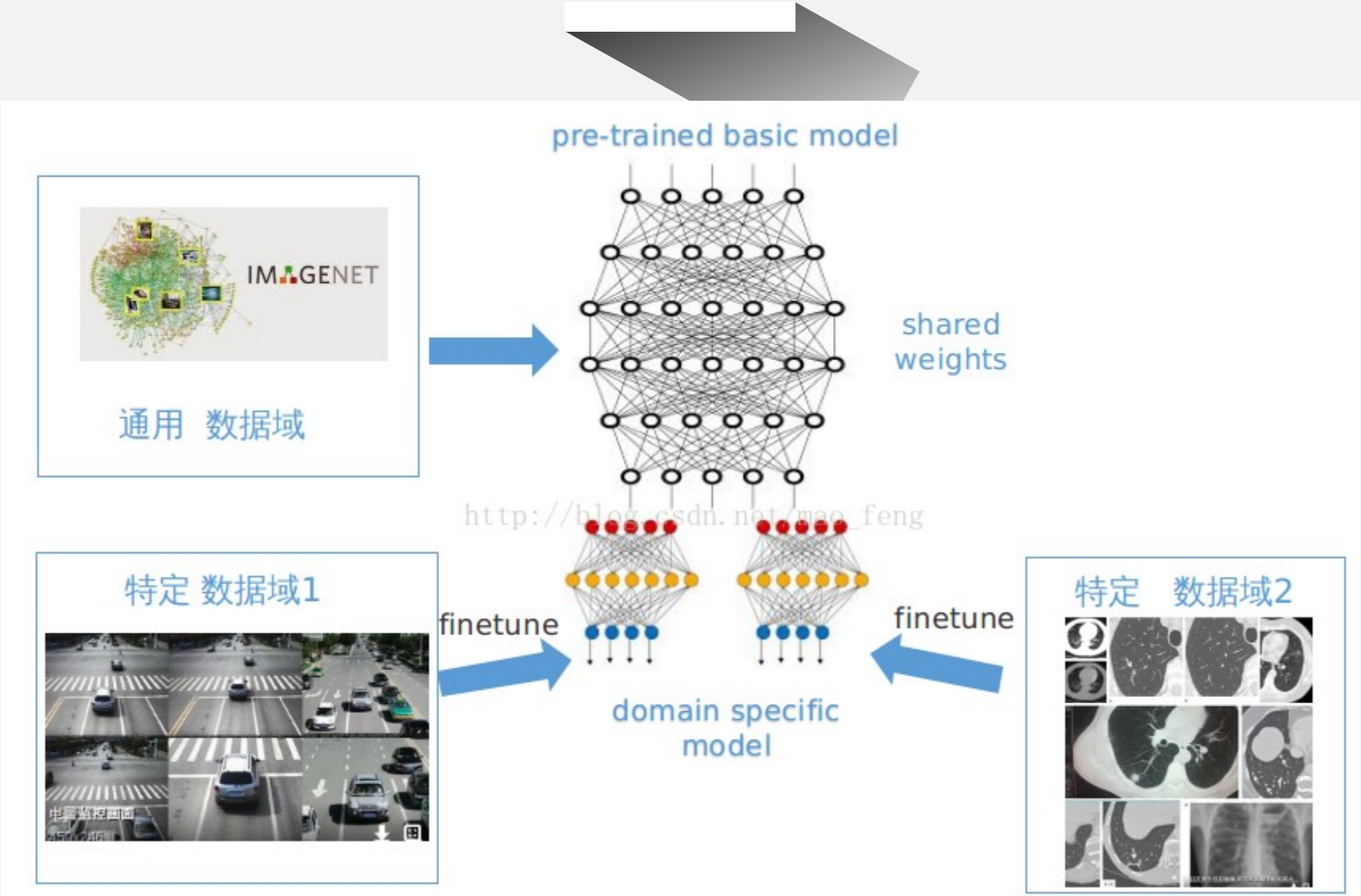
04 Synthesizing new labelled training data by rendering virtual examples or composing synthesized representations or distorting existing training examples;

05 Generating new examples using Generative Adversarial Networks (GANs);

06 Attribute-guided augmentation (AGA) to synthesize samples at desired values or strength

Chen et al. Semantic Feature Augmentation in Few-shot Learning. https://arxiv.org/abs/1804.05298, submitted to ECCV2018

# Finetune

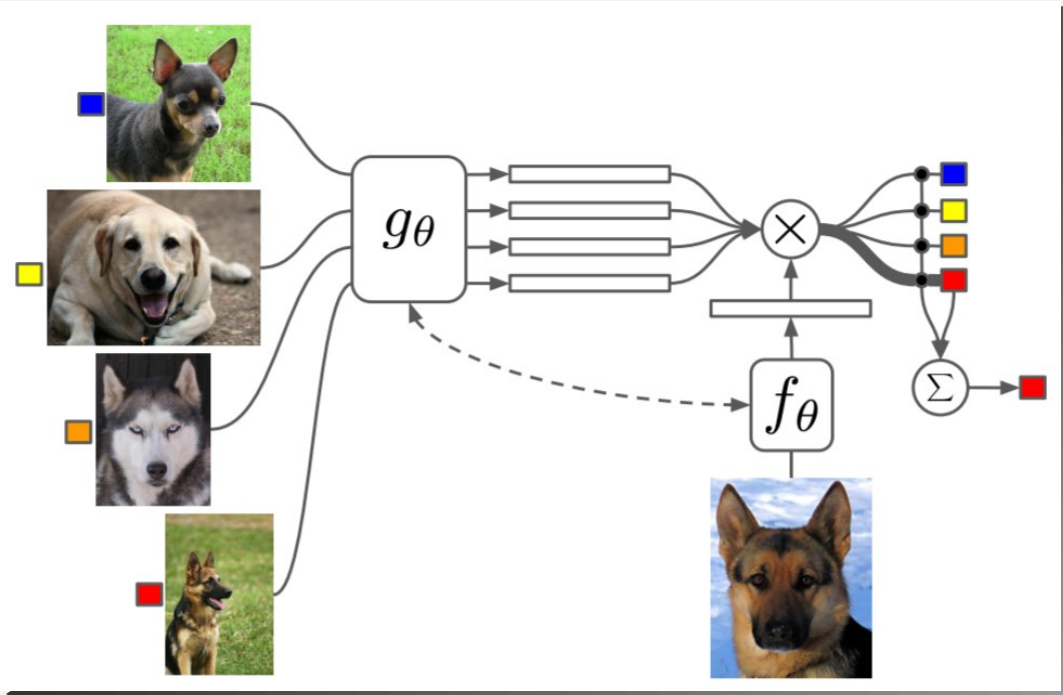

pre-trained basic model
IMAGENET
通用 数据域
shared
weights
http://blog.csdn.net/mao_feng
特定 数据域1
finetune
finetune
特定 数据域2
domain specific
model

# Metric-learning Algorithms

- **Siamese Neural Network**
- **Matching Networks**
- **Prototypical Networks**
- **Relation Network**

Fu et al. Recent Advances in Zero-Shot Recognition: Toward Data-Efficient Understanding of Visual Content. In *SPM*, 2018.

# Matching Networks



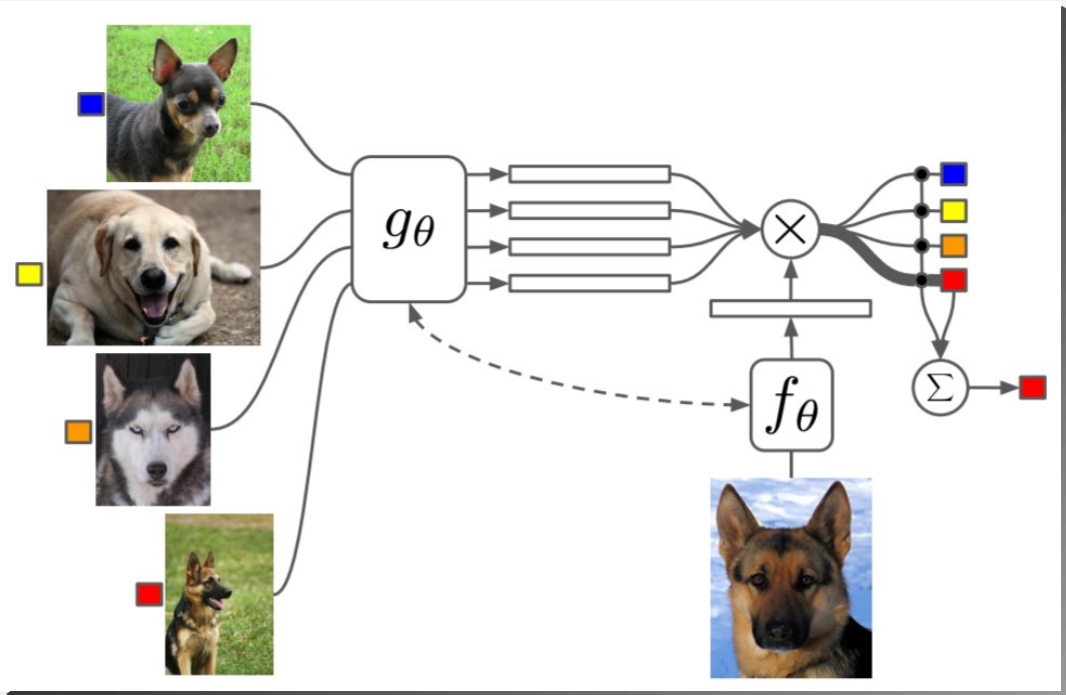

**Novelty** of their work

Modeling level

Training procedure

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In Advances in Neural Information Processing Systems, 2016.

# Matching Networks




**Novelty of Modeling level**

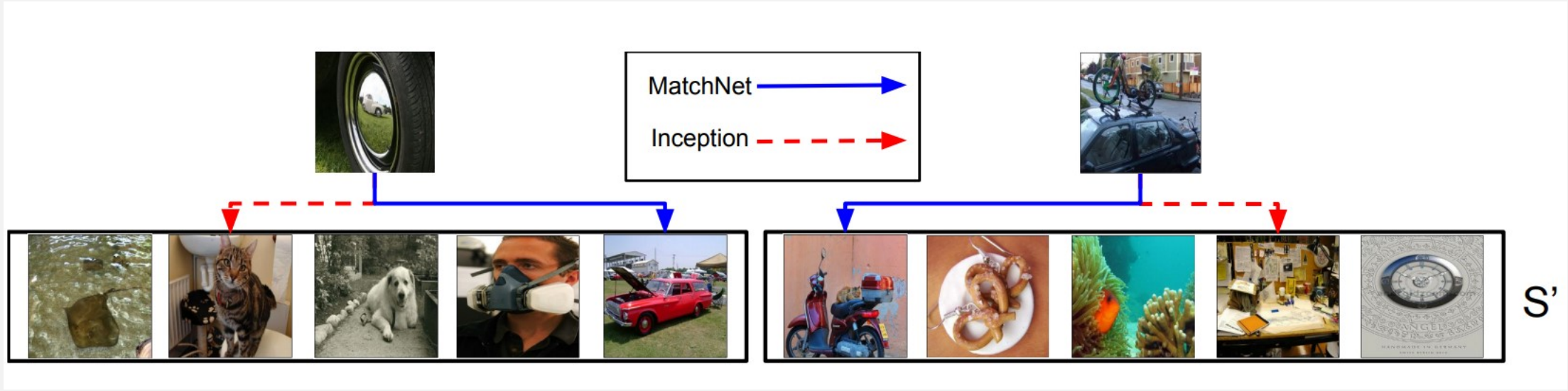
$$\arg max_y \quad P(y|\hat{x}, S), \qquad \hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

$$a(\hat{x}, x_i) = \frac{e^{c(f(\hat{x}), g(x_i))}}{\sum_{i=1}^{k} e^{c(f(\hat{x}), g(x_i))}}$$

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In Advances in Neural Information Processing Systems, 2016.

# Matching Networks

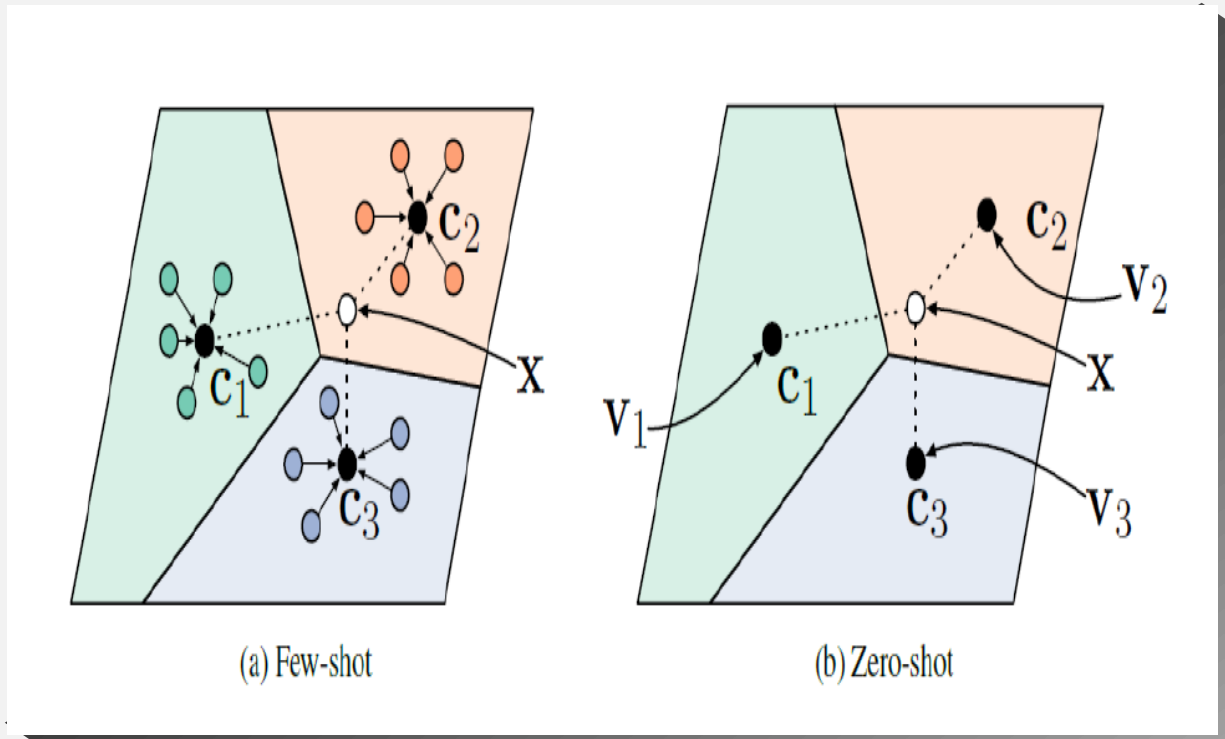
## Training procedure




$$\theta = \arg\max_{\theta} E_{L\sim T}\left[E_{S\sim L, B\sim L}\left[\sum_{(x,y)\in B} \log P_\theta\left(y|x,S\right)\right]\right]$$

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In Advances in Neural Information Processing Systems, 2016.

# Prototypical Networks for Few-shot Learning




**Novelty** of their work

[redacted]

[redacted]

Snell, Jake, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning, Advances in Neural Information Processing Systems. 2017.

# Prototypical Networks for Few-shot Learning

Table 1: Few-shot classification accuracies on Omniglot.

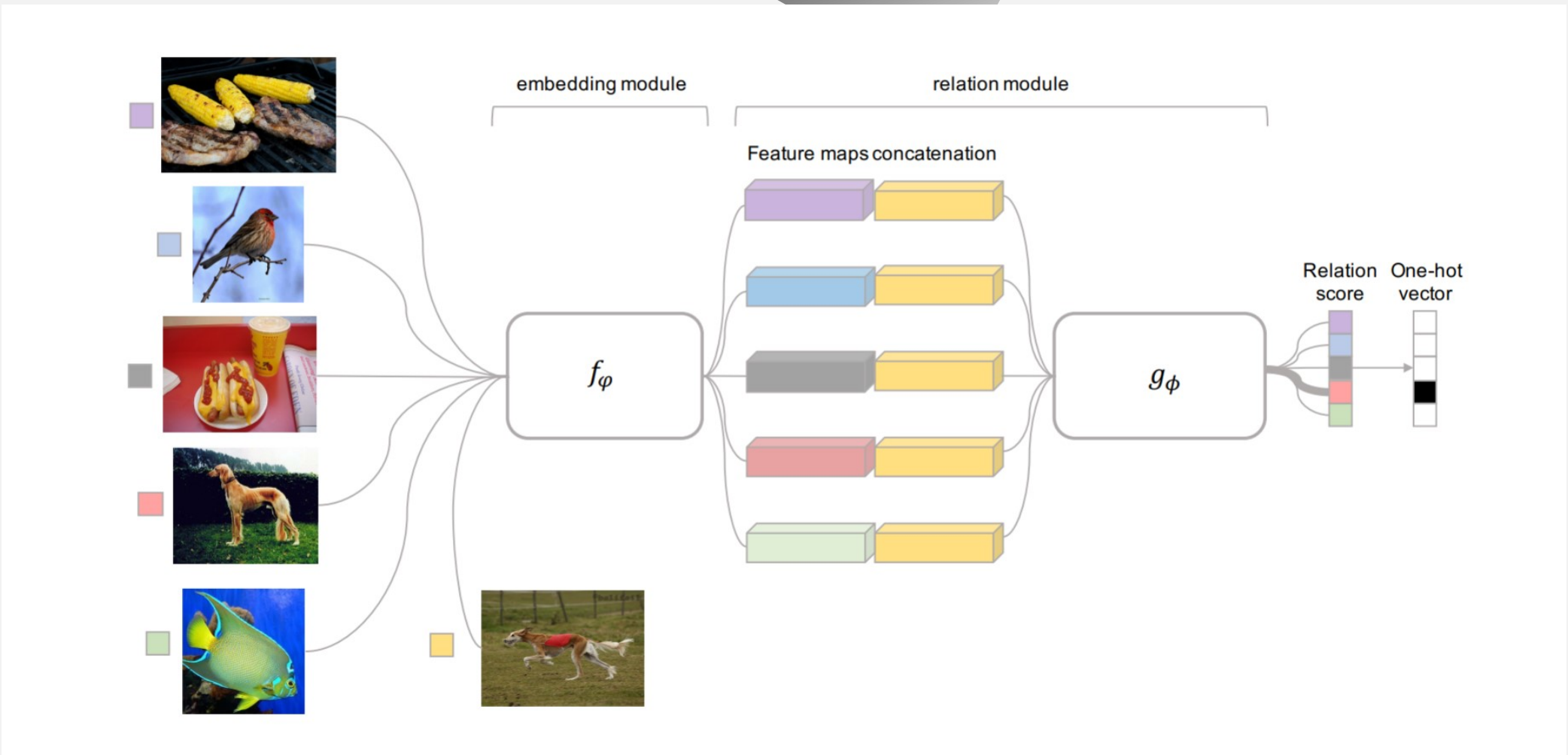
| Model | Dist. | Fine Tune | 5-way Acc. 1-shot | 5-way Acc. 5-shot | 20-way Acc. 1-shot | 20-way Acc. 5-shot |
|---|---|---|---|---|---|---|
| MATCHING NETWORKS [29] | Cosine | N | 98.1% | 98.9% | 93.8% | 98.5% |
| MATCHING NETWORKS [29] | Cosine | Y | 97.9% | 98.7% | 93.5% | 98.7% |
| NEURAL STATISTICIAN [6] | - | N | 98.1% | 99.5% | 93.2% | 98.1% |
| **PROTOTYPICAL NETWORKS (OURS)** | Euclid. | N | **98.8%** | **99.7%** | **96.0%** | **98.9%** |

Table 2: Few-shot classification accuracies on *mini*ImageNet. All accuracy results are averaged over 600 test episodes and are reported with 95% confidence intervals. *Results reported by [22].

| Model | Dist. | Fine Tune | 5-way Acc. 1-shot | 5-way Acc. 5-shot |
|---|---|---|---|---|
| BASELINE NEAREST NEIGHBORS* | Cosine | N | $28.86 \pm 0.54\%$ | $49.79 \pm 0.79\%$ |
| MATCHING NETWORKS [29]* | Cosine | N | $43.40 \pm 0.78\%$ | $51.09 \pm 0.71\%$ |
| MATCHING NETWORKS FCE [29]* | Cosine | N | $43.56 \pm 0.84\%$ | $55.31 \pm 0.73\%$ |
| META-LEARNER LSTM [22]* | - | N | $43.44 \pm 0.77\%$ | $60.60 \pm 0.71\%$ |
| **PROTOTYPICAL NETWORKS (OURS)** | Euclid. | N | $\mathbf{49.42 \pm 0.78\%}$ | $\mathbf{68.20 \pm 0.66\%}$ |

Snell, Jake, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning, Advances in Neural Information Processing Systems. 2017.

# Relation network for few-shot learning




Sung, Flood, et al. Learning to Compare: Relation Network for Few-Shot Learning. In CVPR, 2018.

# Relation network for few-shot learning

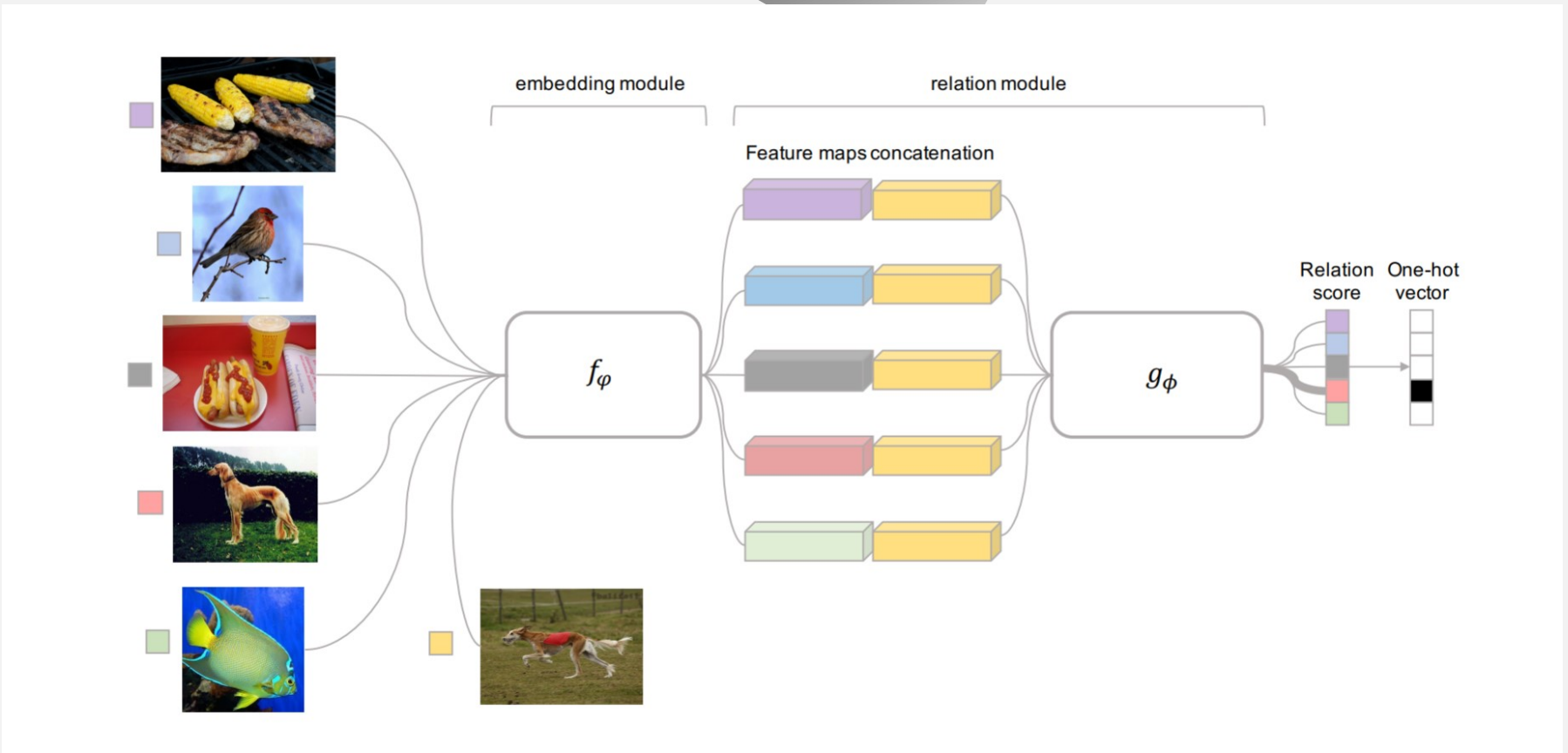
## motivation


一般般吧~

卤蛋精!

Sung, Flood, et al. Learning to Compare: Relation Network for Few-Shot Learning. In CVPR, 2018.

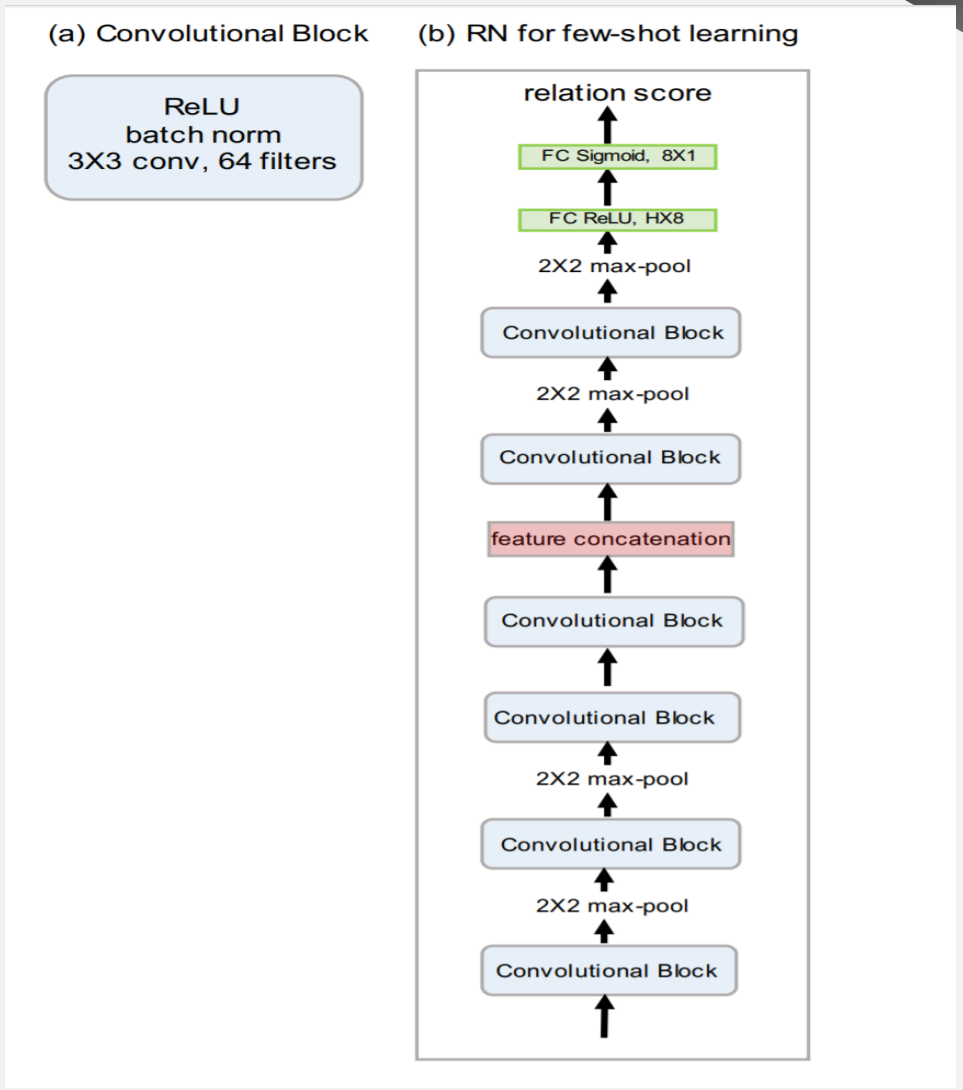
# Relation network for few-shot learning

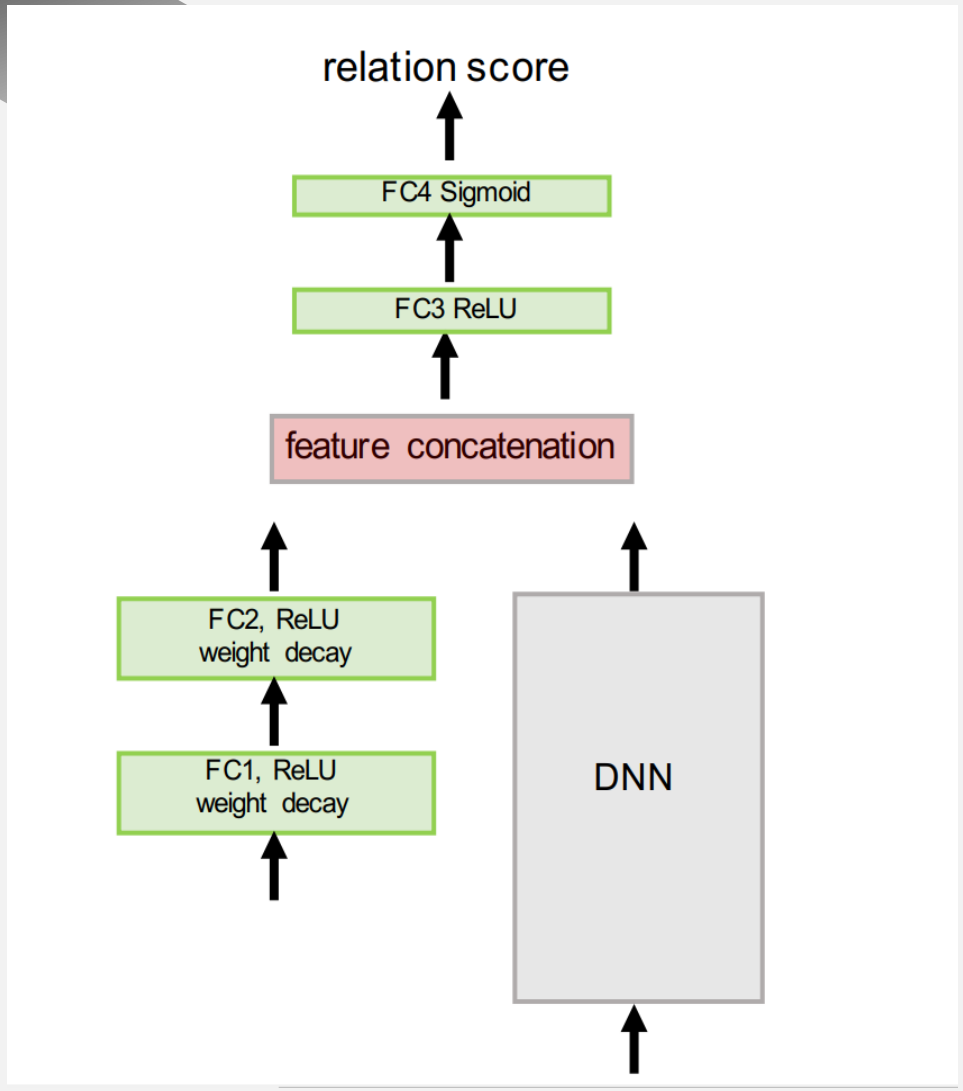



$$r_{i,j} = g_\phi\left(C\left(f_\varphi(v_c), f_\varphi(x_j)\right)\right), \qquad i = 1,2,3,\ldots,C$$

Sung, Flood, et al. Learning to Compare: Relation Network for Few-Shot Learning. In CVPR, 2018.

# Relation network for few-shot learning




Sung, Flood, et al. Learning to Compare: Relation Network for Few-Shot Learning. In CVPR, 2018.

# Relation network for few-shot learning

| Model | Fine Tune | 5-way Acc. | | 20-way Acc. | |
|---|---|---|---|---|---|
| | | 1-shot | 5-shot | 1-shot | 5-shot |
| MANN [32] | N | 82.8% | 94.9% | - | - |
| CONVOLUTIONAL SIAMESE NETS [20] | N | 96.7% | 98.4% | 88.0% | 96.5% |
| CONVOLUTIONAL SIAMESE NETS [20] | Y | 97.3% | 98.4% | 88.1% | 97.0% |
| MATCHING NETS [39] | N | 98.1% | 98.9% | 93.8% | 98.5% |
| MATCHING NETS [39] | Y | 97.9% | 98.7% | 93.5% | 98.7% |
| SIAMESE NETS WITH MEMORY [18] | N | 98.4% | 99.6% | 95.0% | 98.6% |
| NEURAL STATISTICIAN [8] | N | 98.1% | 99.5% | 93.2% | 98.1% |
| META NETS [27] | N | 99.0% | - | 97.0% | - |
| PROTOTYPICAL NETS [36] | N | 98.8% | 99.7% | 96.0% | 98.9% |
| MAML [10] | Y | 98.7 ± 0.4% | **99.9 ± 0.1%** | 95.8 ± 0.3% | 98.9 ± 0.2% |
| RELATION NET | N | **99.6 ± 0.2%** | **99.8± 0.1%** | **97.6 ± 0.2%** | **99.1± 0.1%** |

Fitting everything!

| Model | FT | 5-way Acc. | |
|---|---|---|---|
| | | 1-shot | 5-shot |
| MATCHING NETS [39] | N | 43.56 ± 0.84% | 55.31 ± 0.73% |
| META NETS [27] | N | 49.21 ± 0.96% | - |
| META-LEARN LSTM [29] | N | 43.44 ± 0.77% | 60.60 ± 0.71% |
| MAML [10] | Y | 48.70 ± 1.84% | 63.11 ± 0.92% |
| PROTOTYPICAL NETS [36] | N | 49.42 ± 0.78% | **68.20 ± 0.66%** |
| RELATION NET | N | **50.44 ± 0.82%** | 65.32 ± 0.70% |

Sung, Flood, et al. Learning to Compare: Relation Network for Few-Shot Learning. In CVPR, 2018.

03

# Prior Knowledge

In this part, I shall introduce the RNN, attention mechanism, VQA problem.

# Prior Knowledge

## RNN

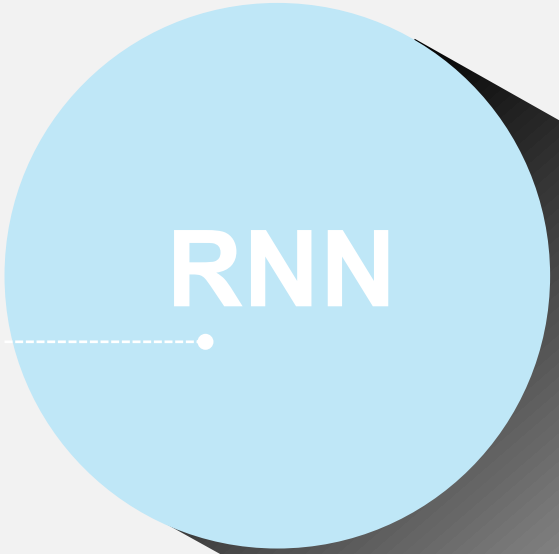
I'll introduce the recurrent neural network, including LSTM,GRU and word2vec.

## Attention mechanism

Summary of attention mechanism.

## VQA

I'll describe the visual question answering problem.



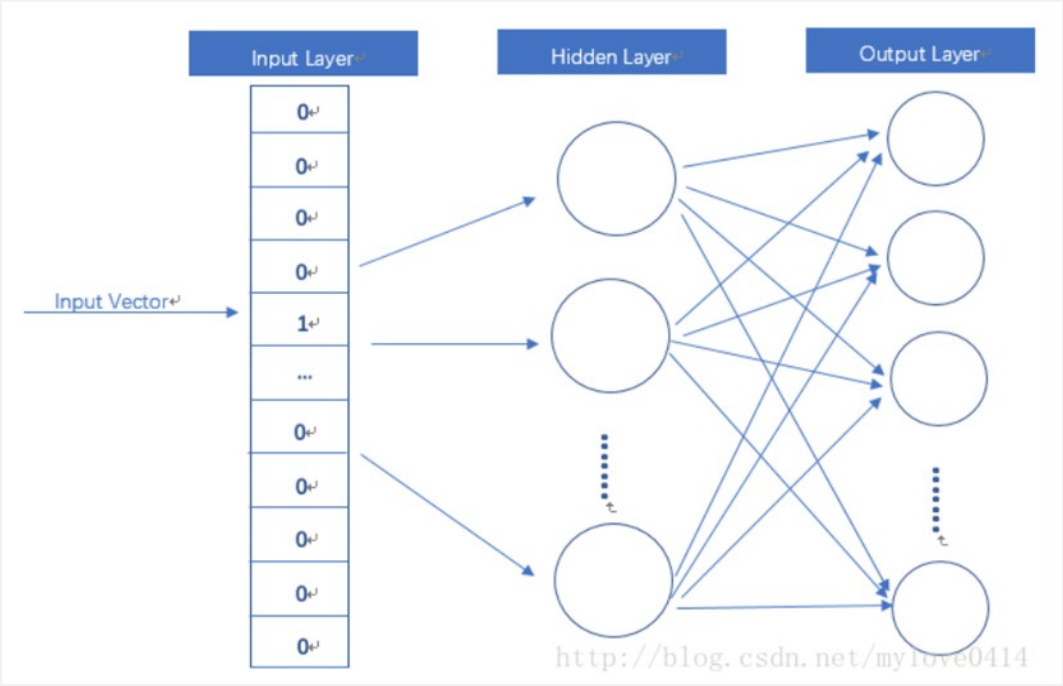

01 What is Recurrent Neural Network?

02 Several Units

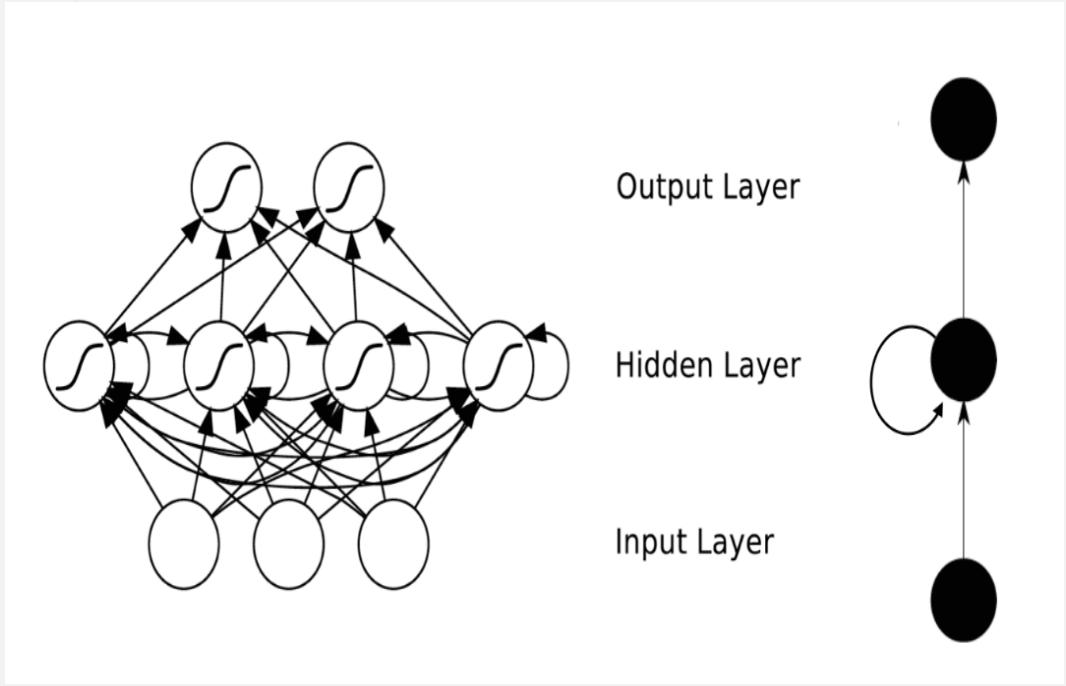
03 Representing words

04 Application of Recurrent Neural Network

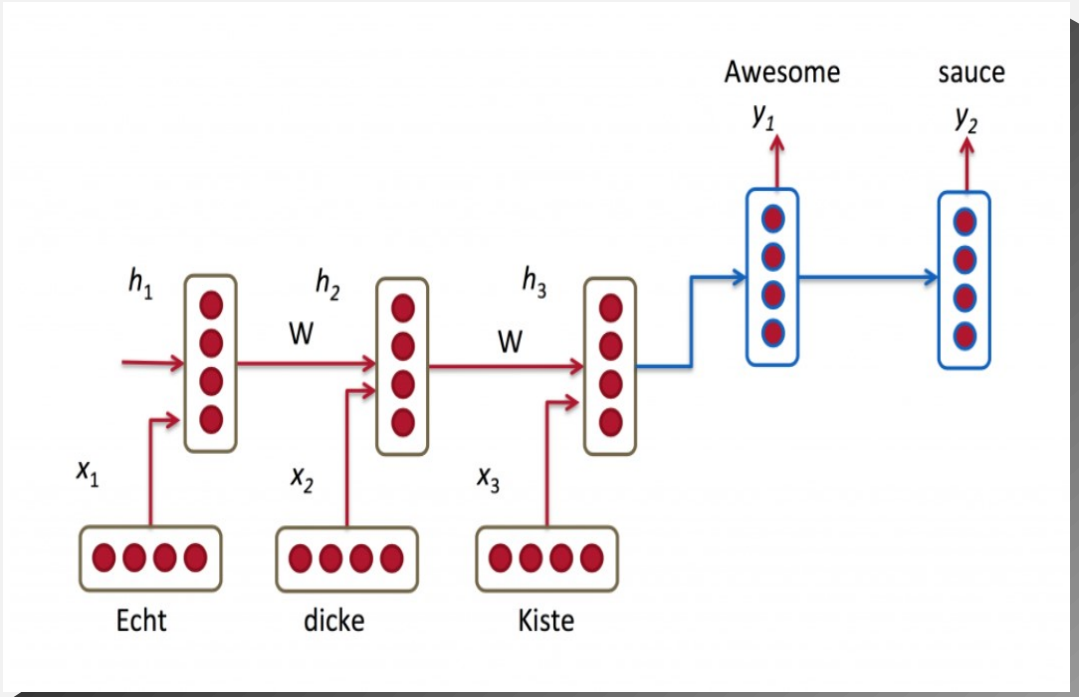
# What are RNNs?




VS



# What can RNNs do?

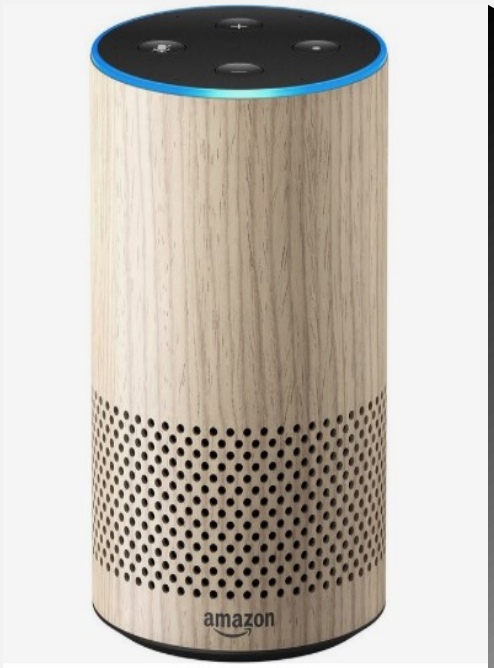



## Machine Translation

英 Hello world!

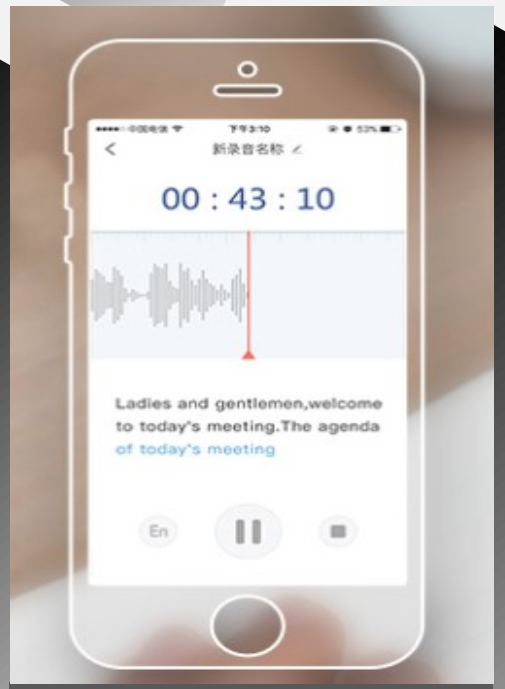
你好 世界! 中

# Speech Recognition




Smart speaker

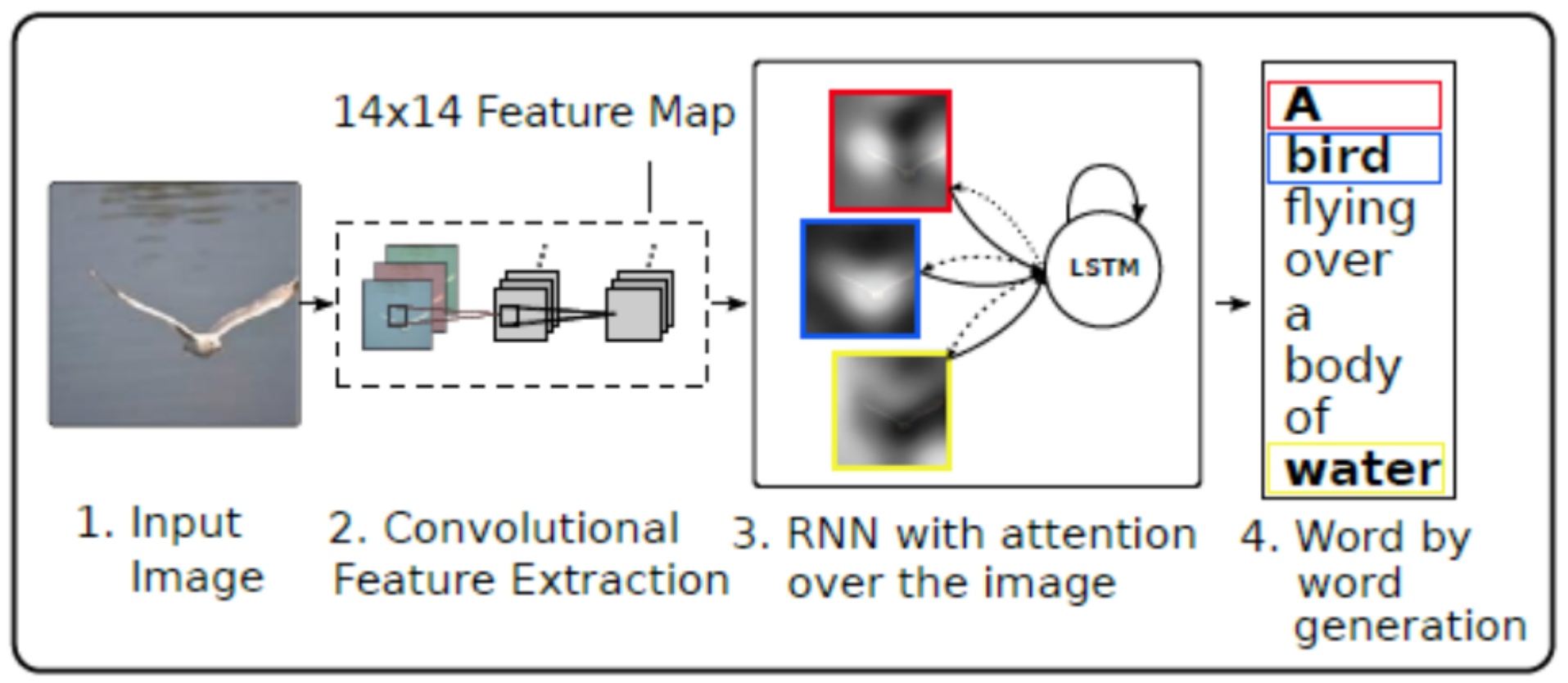

Voice assistant




Real-time ASR

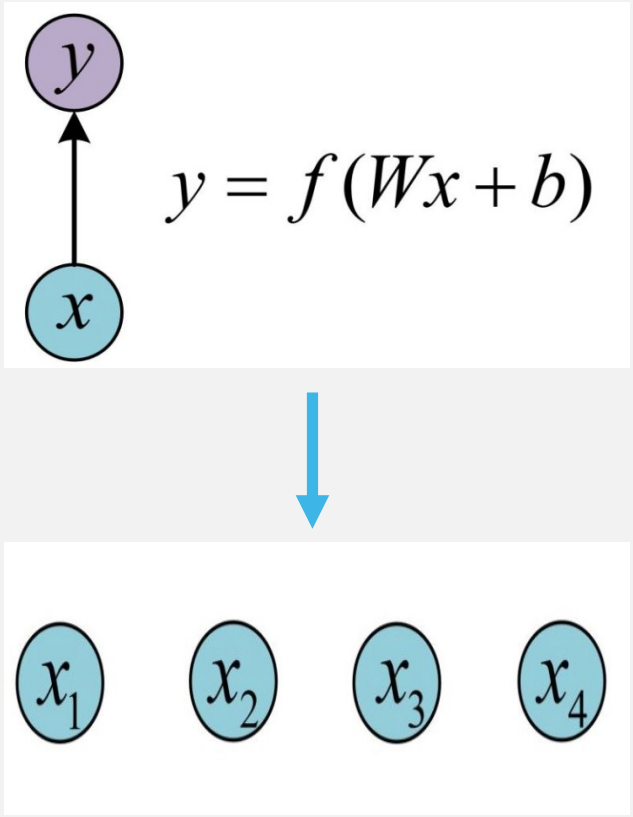

Realtime navigation

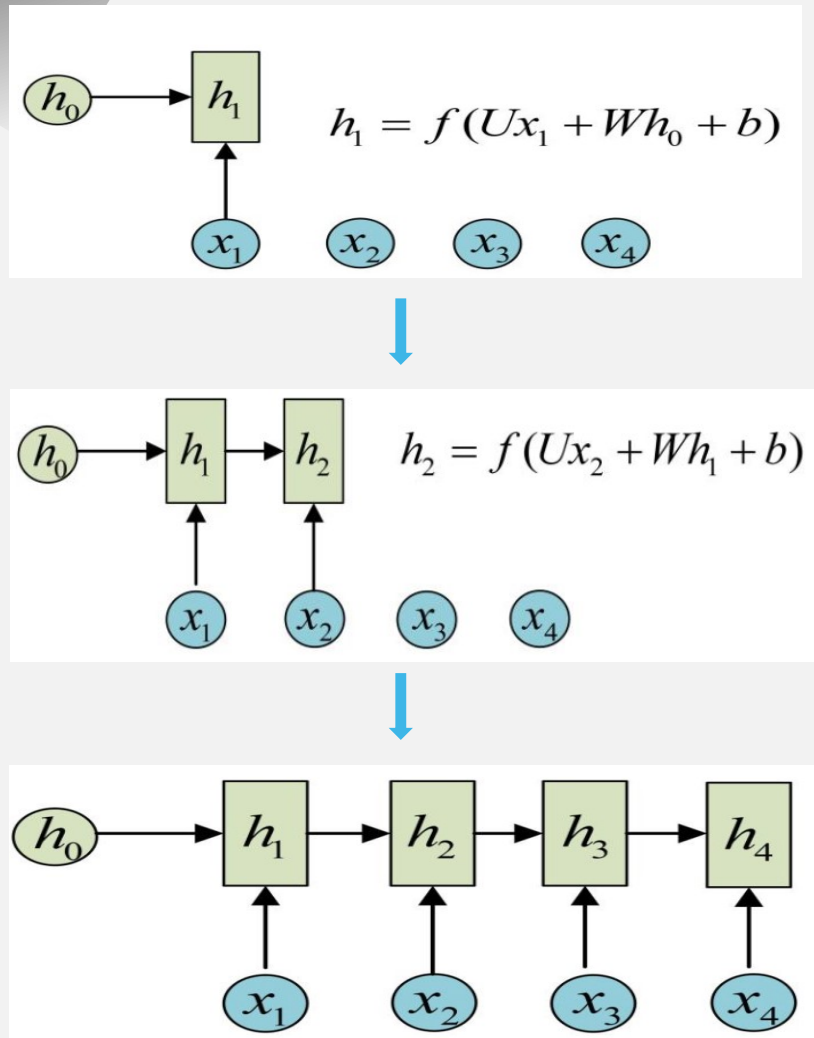
# Image Caption




* Image is from Xu, Kelvin, et al. Show, attend and tell: Neural image caption generation with visual attention. JMLR, 2015.

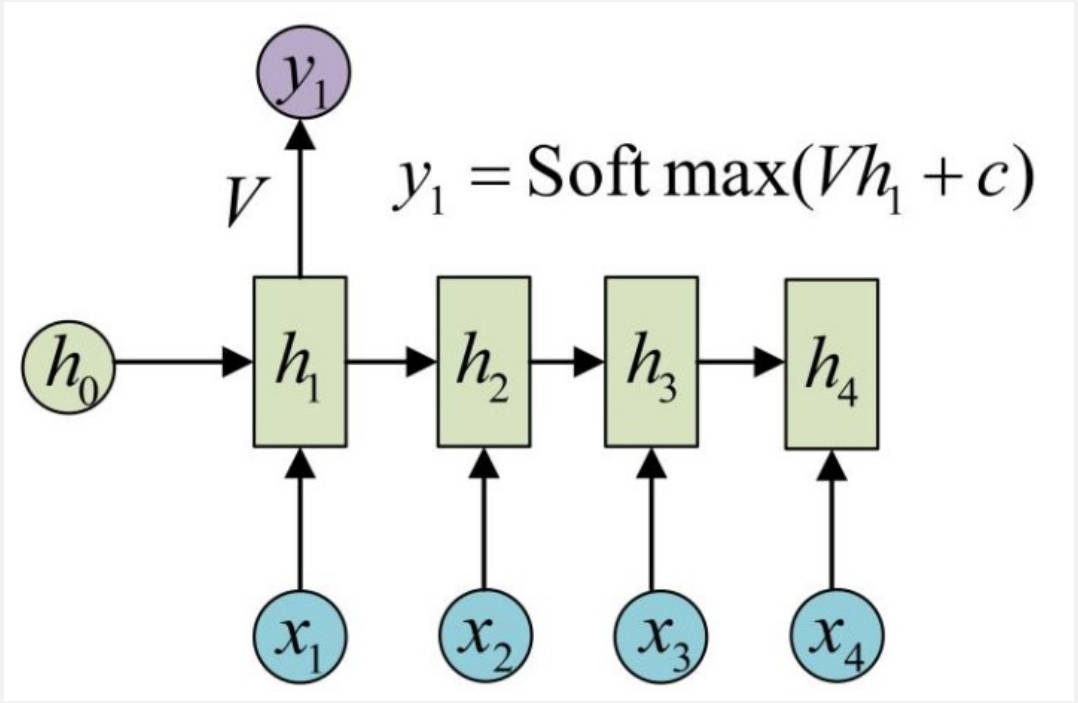
# Classical RNN structure

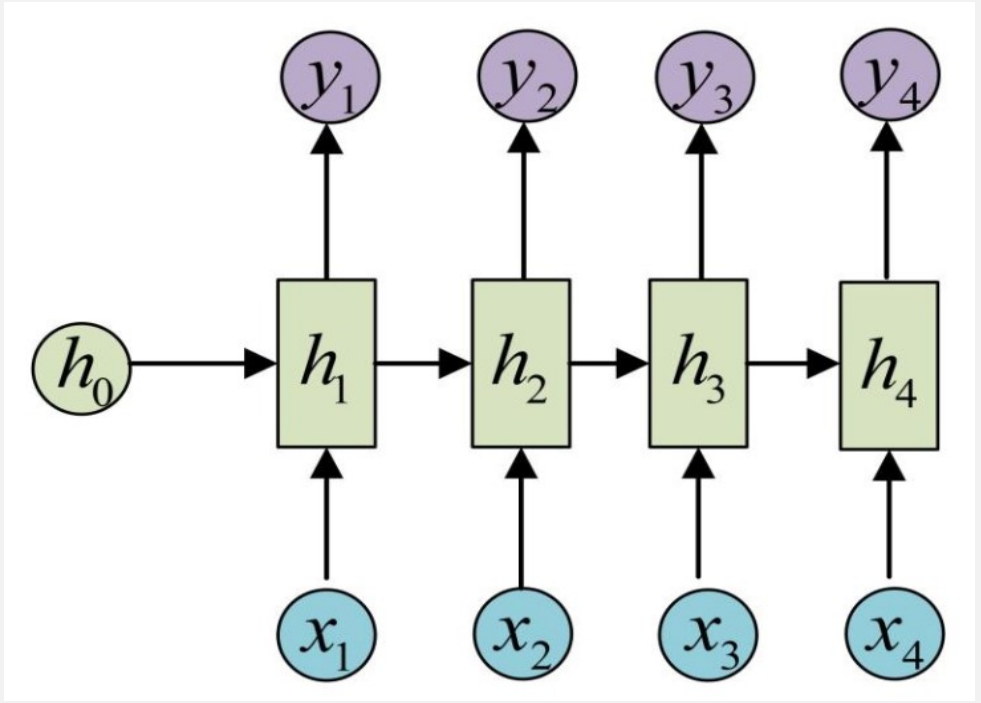



* Images are from https://zhuanlan.zhihu.com/p/28054589.

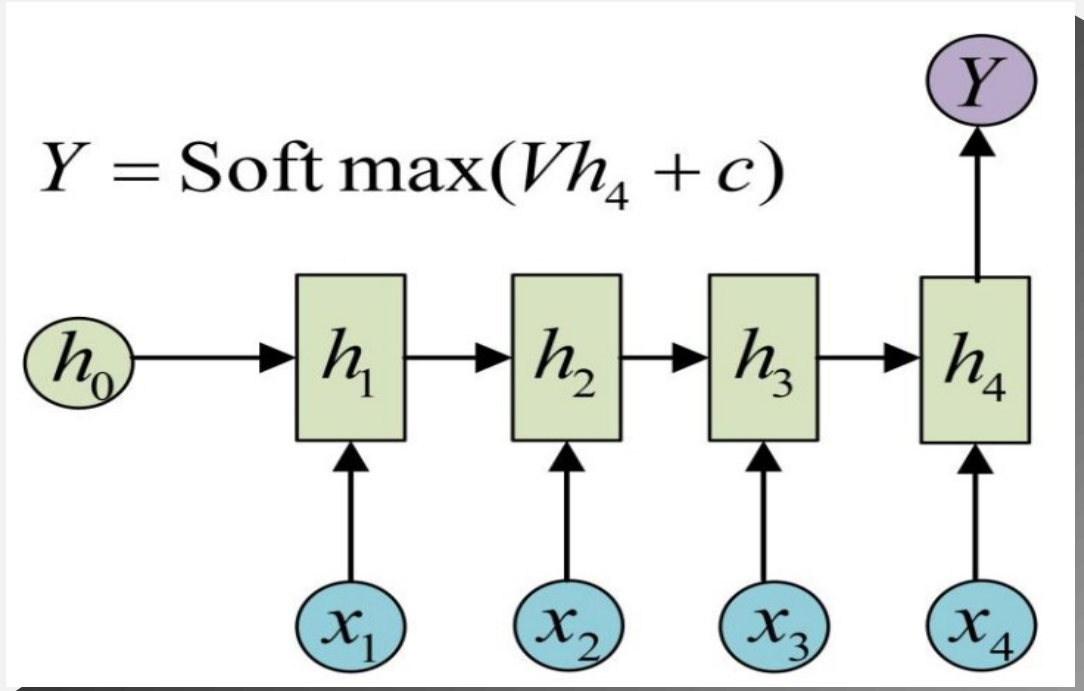
# Classical RNN structure




* Images are from https://zhuanlan.zhihu.com/p/28054589.

# Some variants



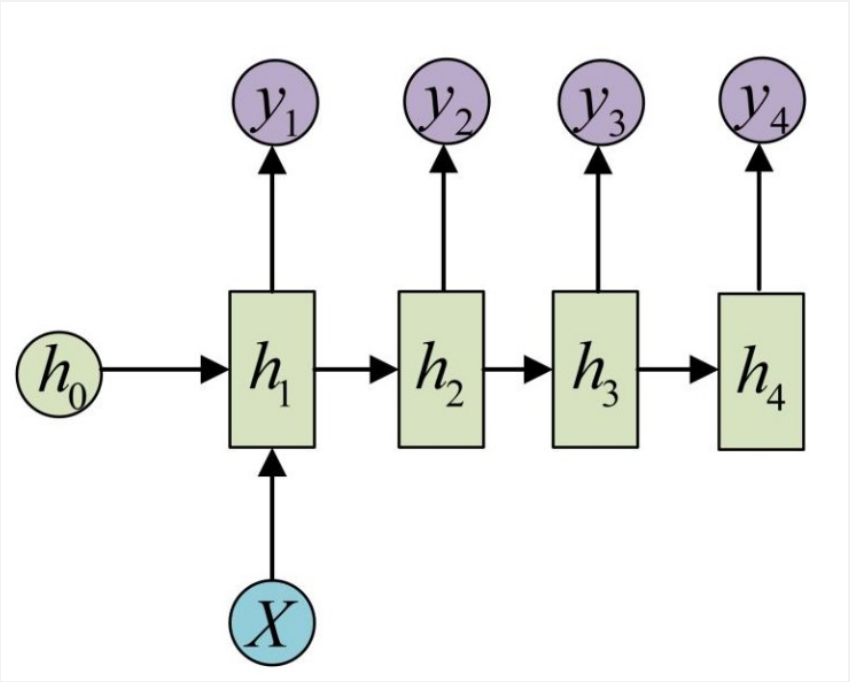

N VS 1

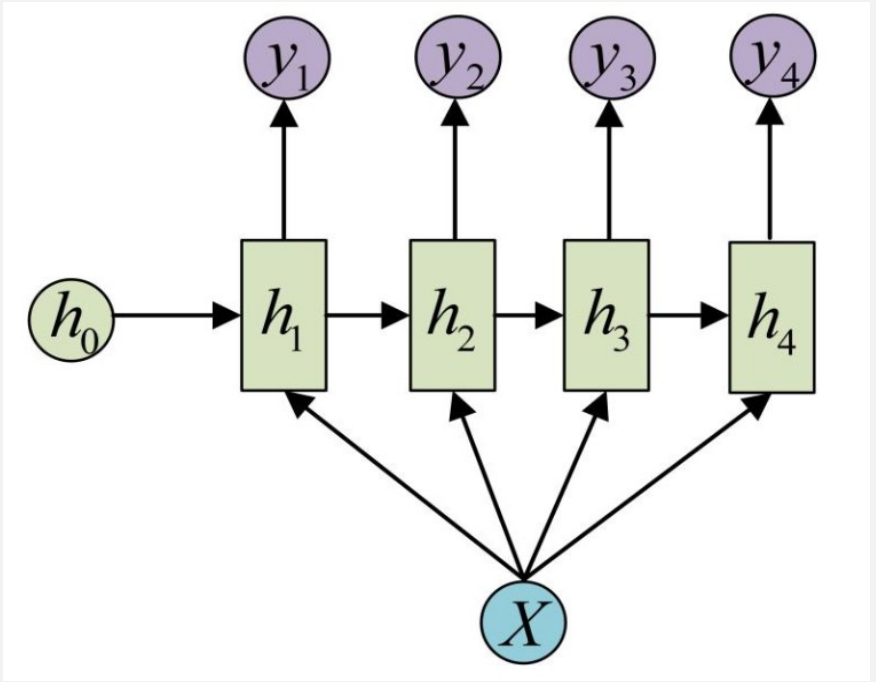
Text classification

Video classification

* Images are from https://zhuanlan.zhihu.com/p/28054589.

# 1 VS N



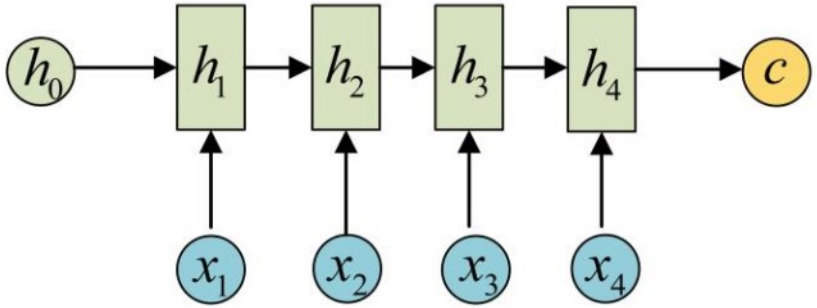

* Images are from https://zhuanlan.zhihu.com/p/28054589.

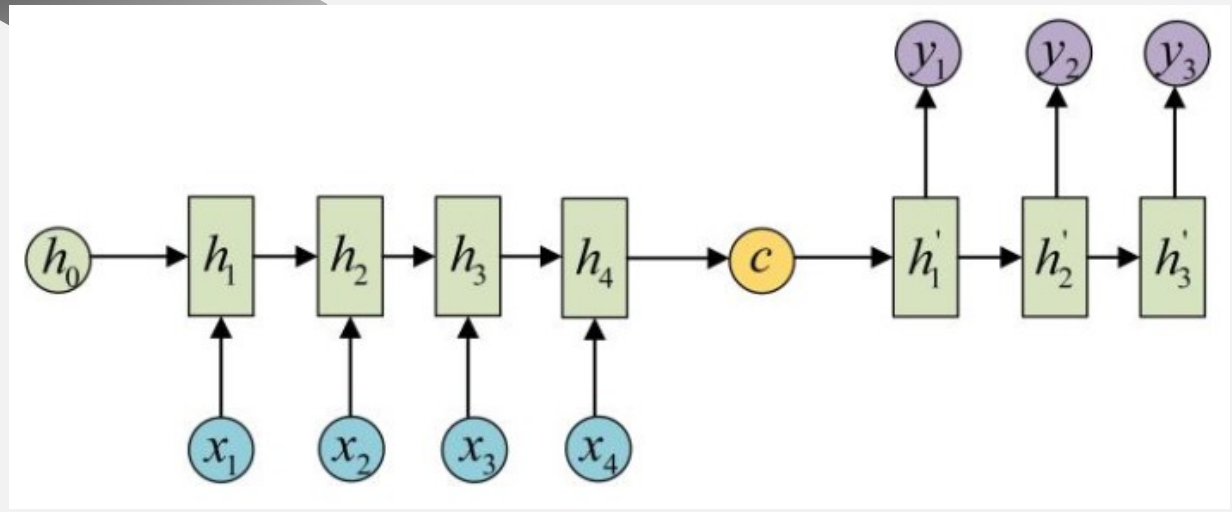
# Seq2Seq (N VS M)

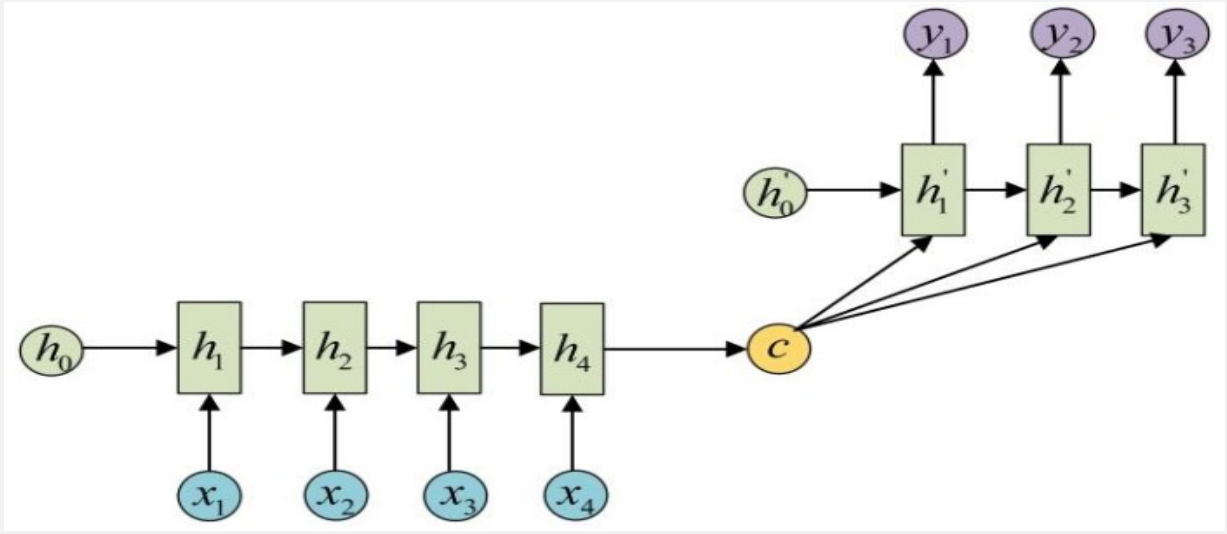
$$(1)\ c = h_4$$

$$(2)\ c = q(h_4)$$

$$(3)\ c = q(h_1, h_2, h_3, h_4)$$




* Images are from https://zhuanlan.zhihu.com/p/28054589.

## Machine Comprehension

Sam walks into the kitchen.
Sam picks up an apple.
Sam walks into the bedroom.
Sam drops the apple.
Q: Where is the apple?
A. Bedroom

## Auto Text Summarization

源文：

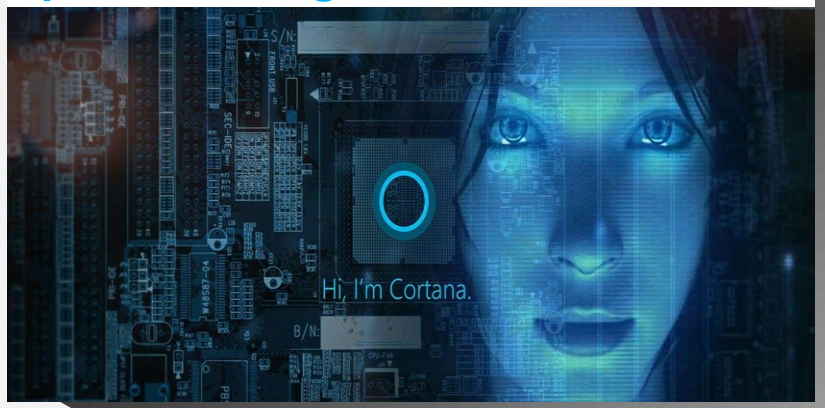
海湾报刊对美国新当选总统克林顿，能否帮助振兴中东和平进程感到怀疑，但也确实看到了一丝希望。

==========

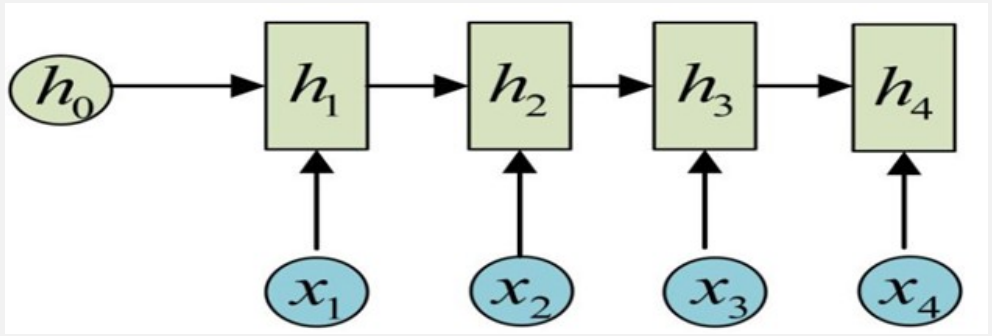
人类：海湾报刊对克林顿是否会恢复和平进程，持怀疑态度

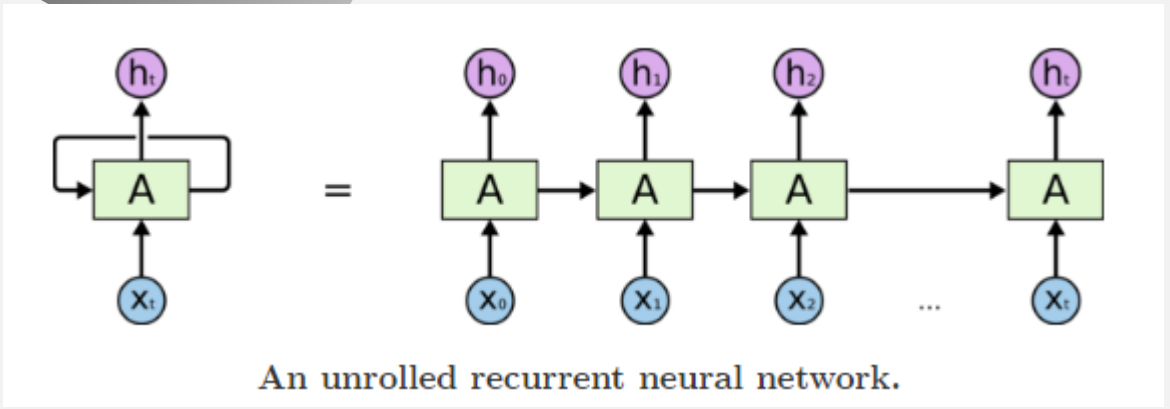
机器：海湾新闻界对克林顿恢复和平进程的前景，持怀疑态度

## Speech Recognition

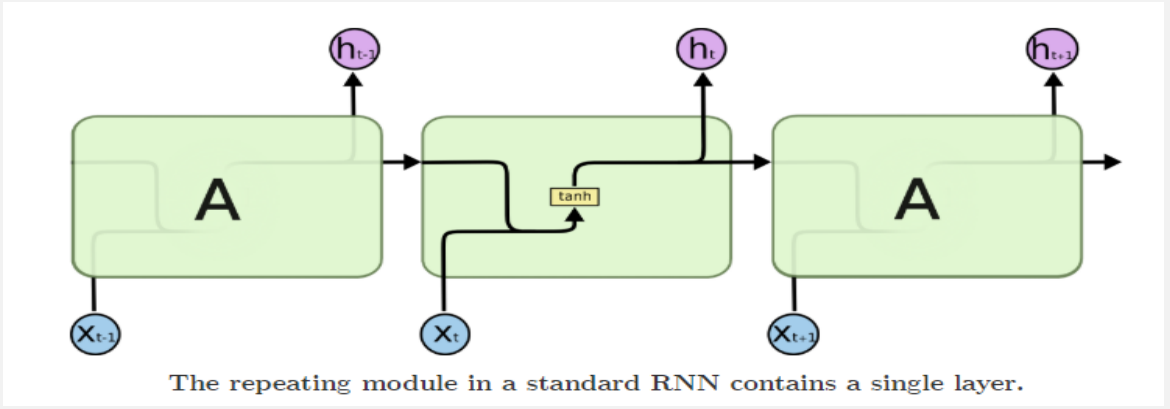


# Understanding LSTM Networks




An unrolled recurrent neural network.

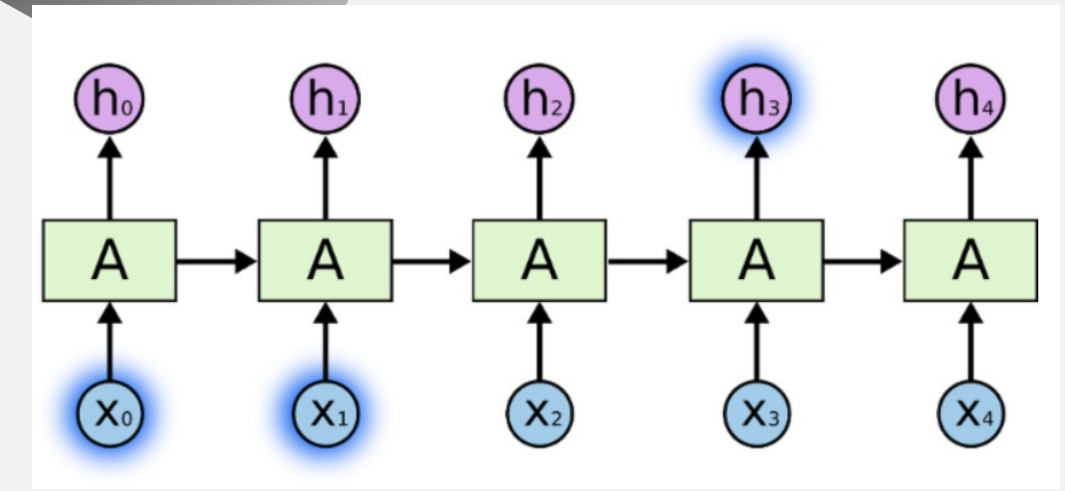



The repeating module in a standard RNN contains a single layer.

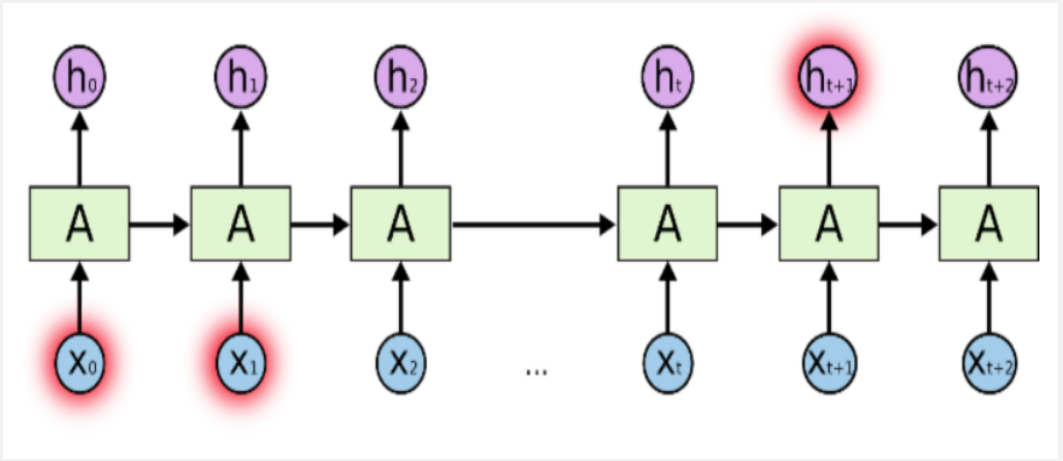
* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

# Understanding LSTM Networks

**The clouds are in the sky.**

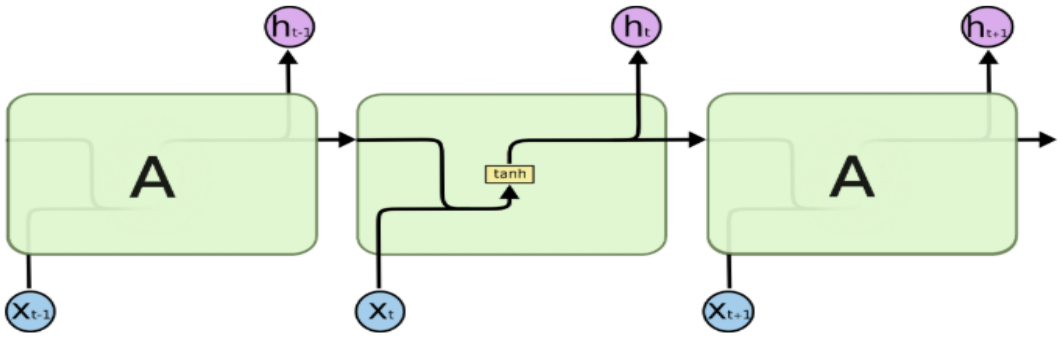



**I grew up in France... I speak fluent French.**

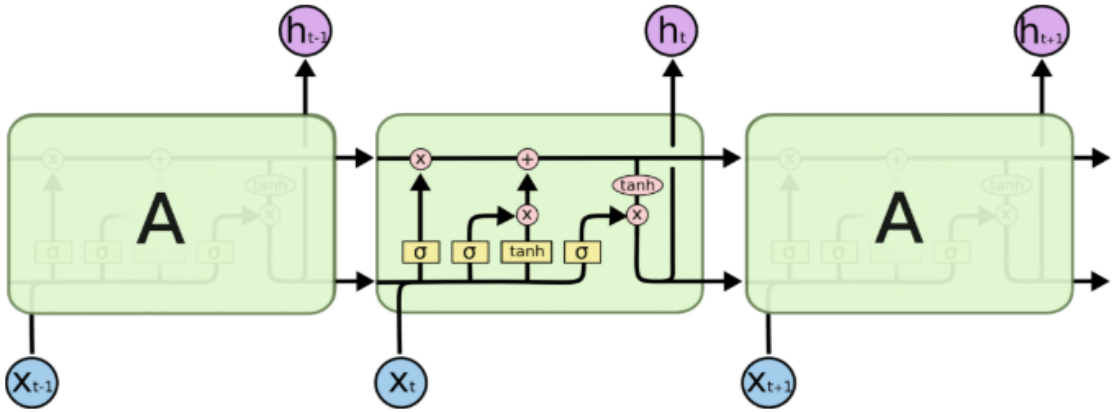



* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

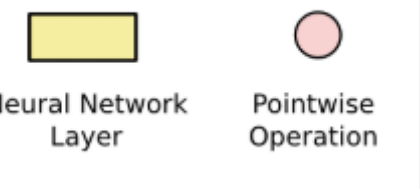
# Understanding LSTM Networks




The repeating module in a standard RNN contains a single layer.




The repeating module in an LSTM contains four interacting layers.

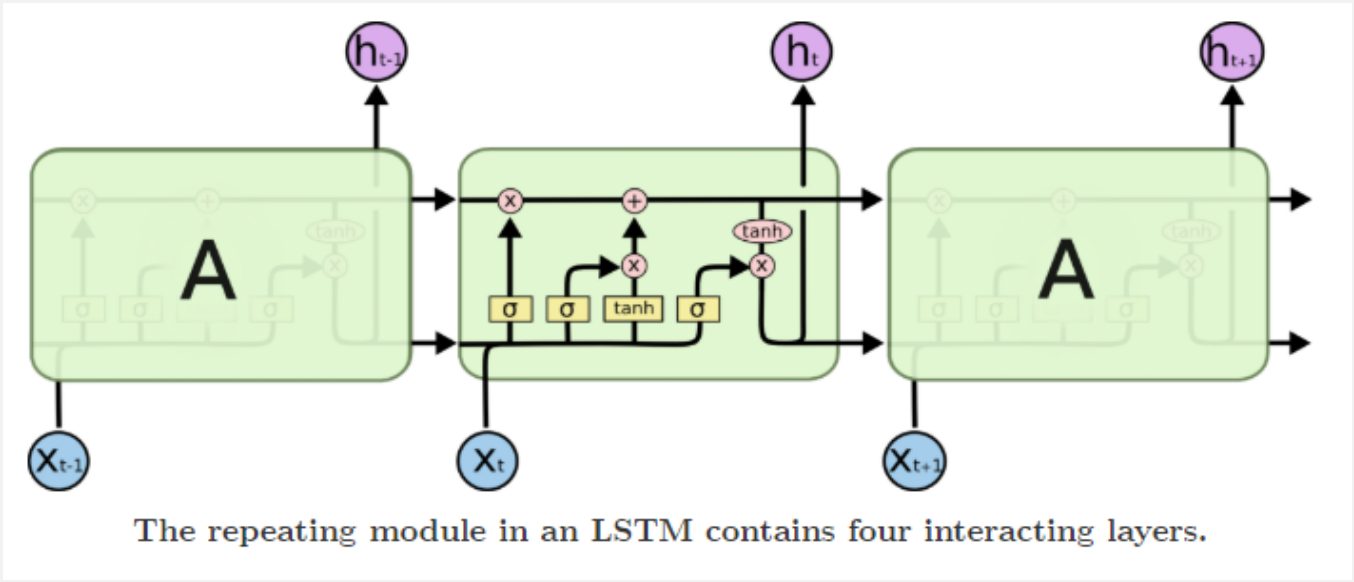




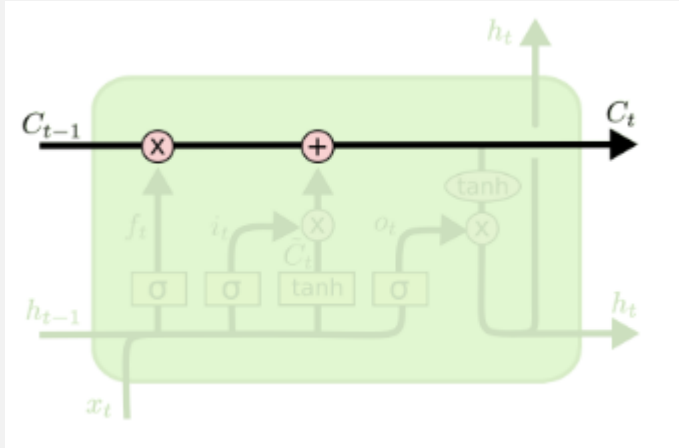
## Long Short Term Memory

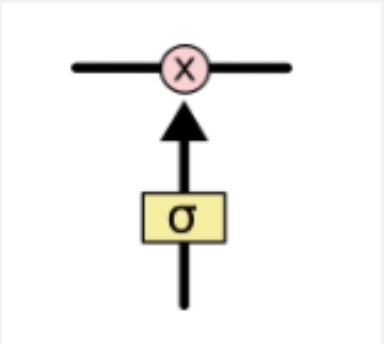
* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

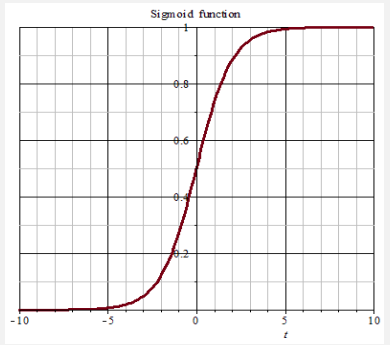
# Understanding LSTM Networks




The repeating module in an LSTM contains four interacting layers.

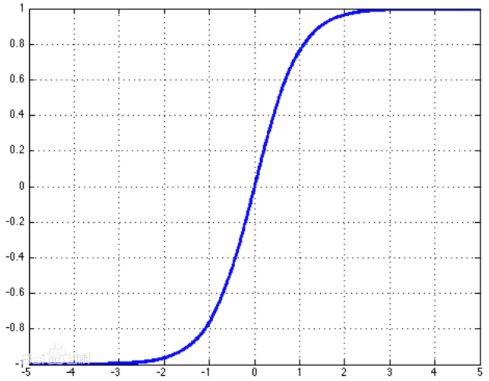


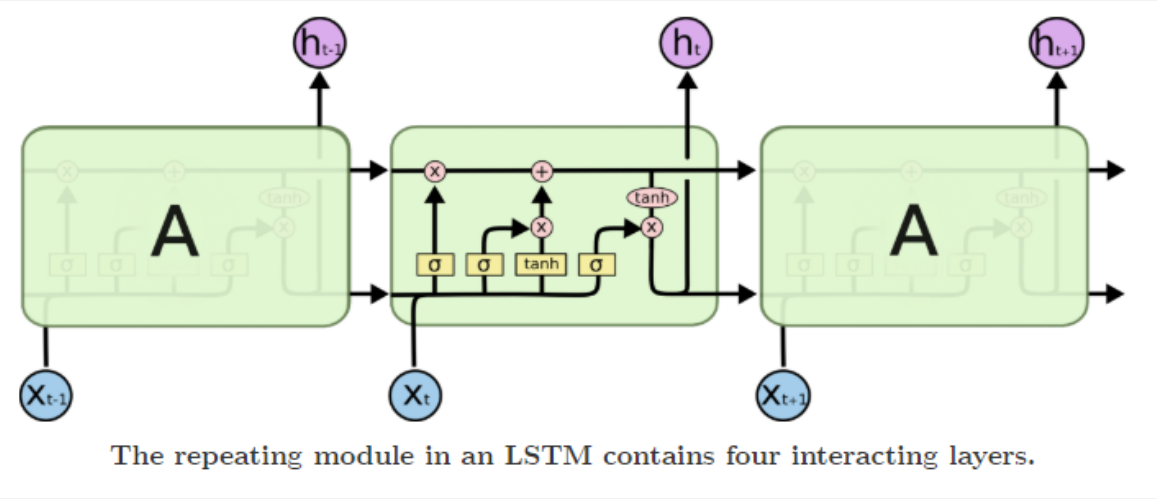



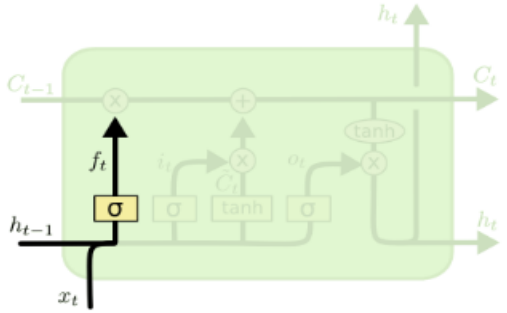

* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

# Understanding LSTM Networks

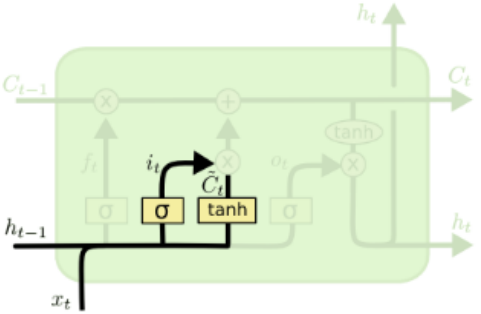





The repeating module in an LSTM contains four interacting layers.



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

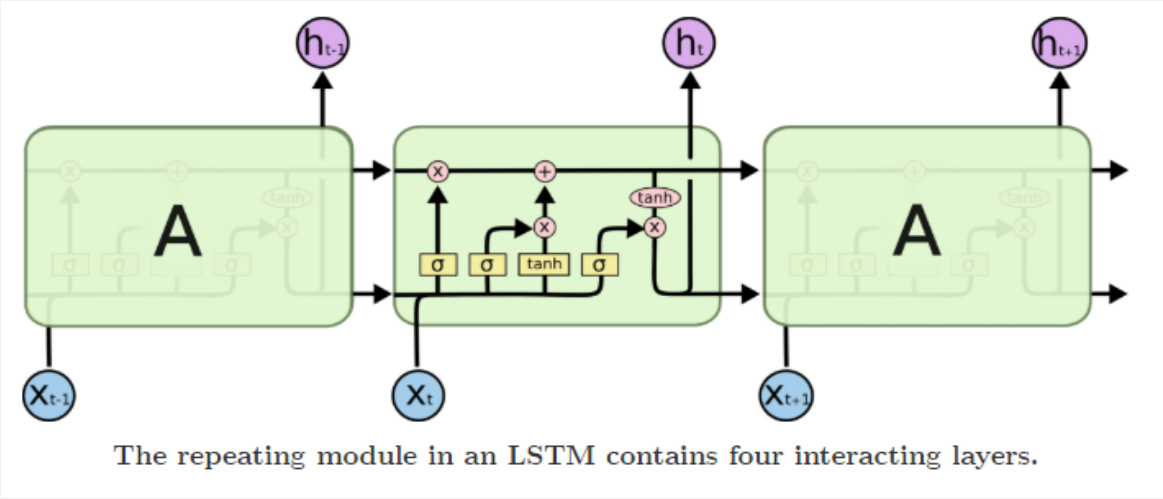


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

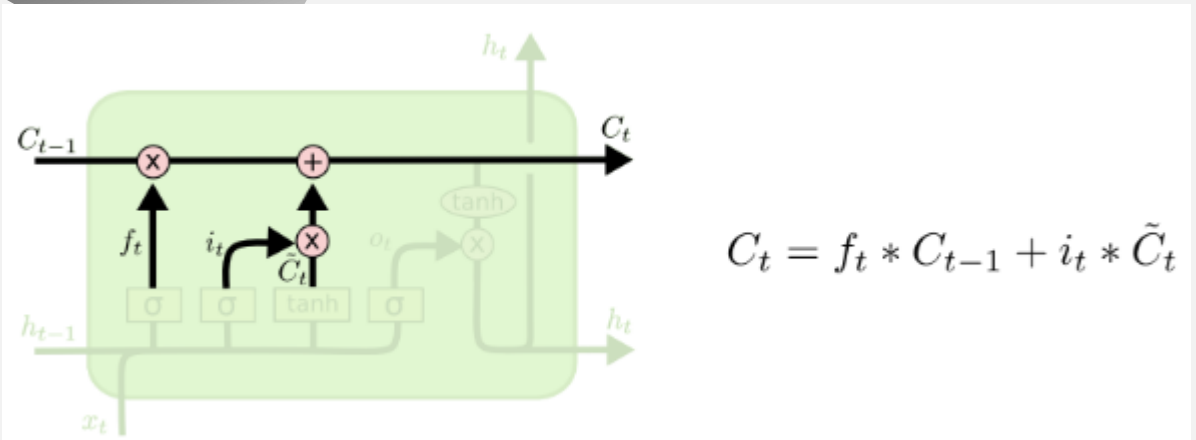
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/ and https://baike.baidu.com

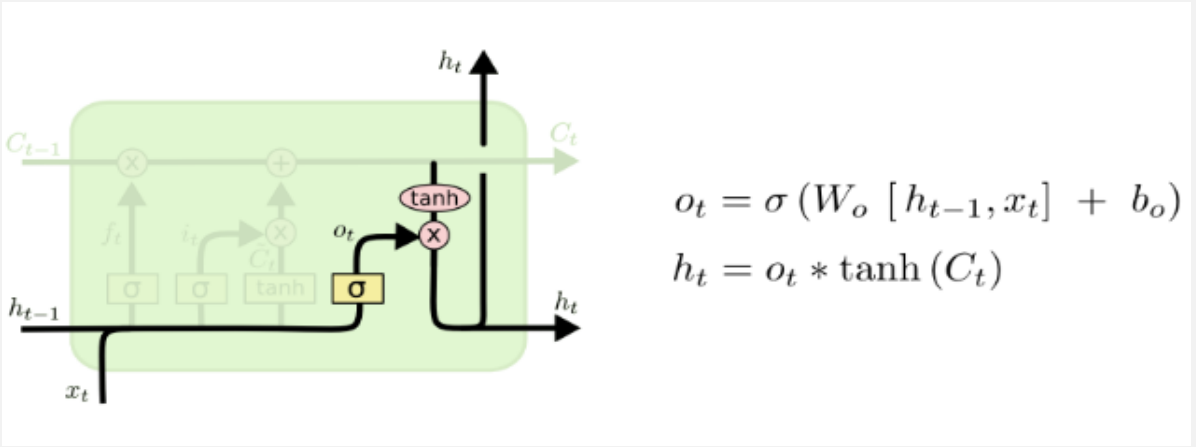
# Understanding LSTM Networks




The repeating module in an LSTM contains four interacting layers.



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

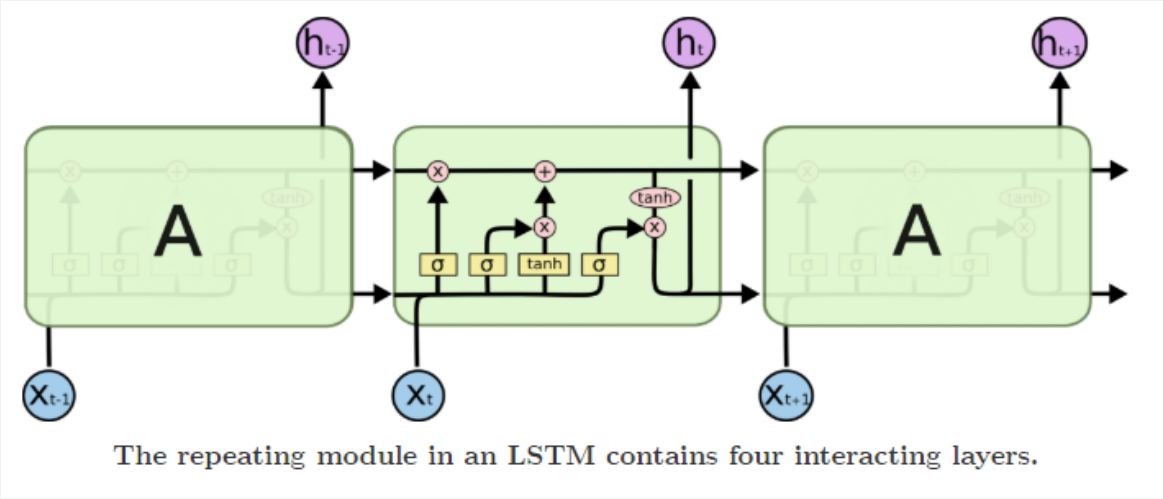


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

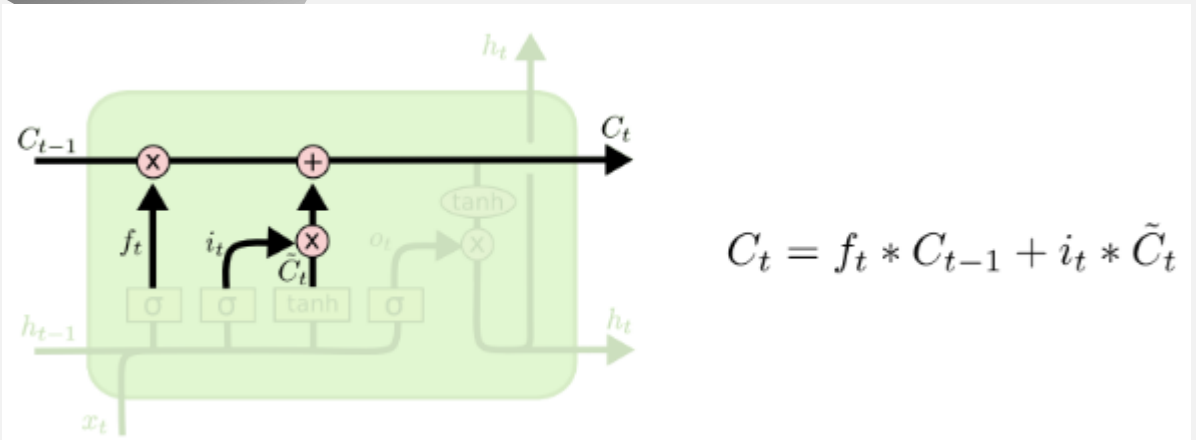
$$h_t = o_t * \tanh\left(C_t\right)$$

* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/ and https://baike.baidu.com

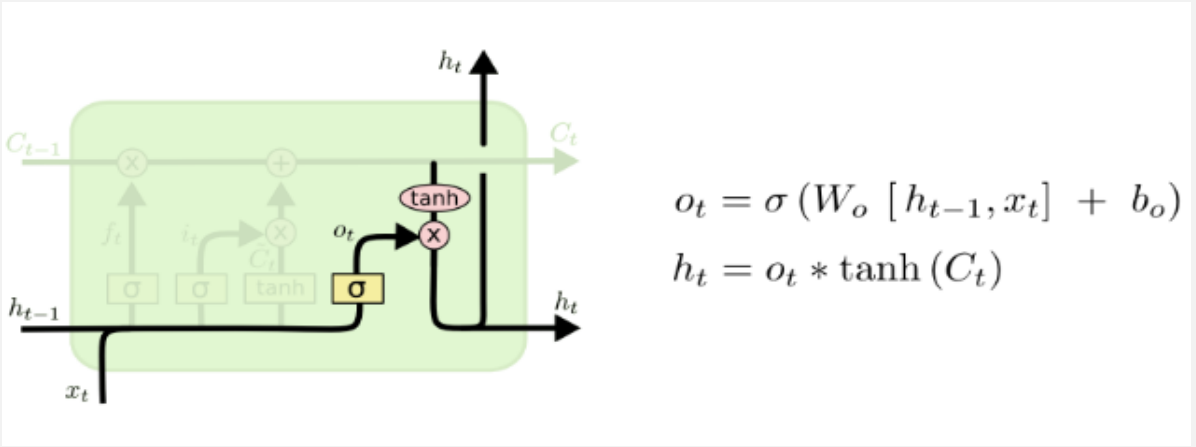
# Understanding LSTM Networks




The repeating module in an LSTM contains four interacting layers.



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

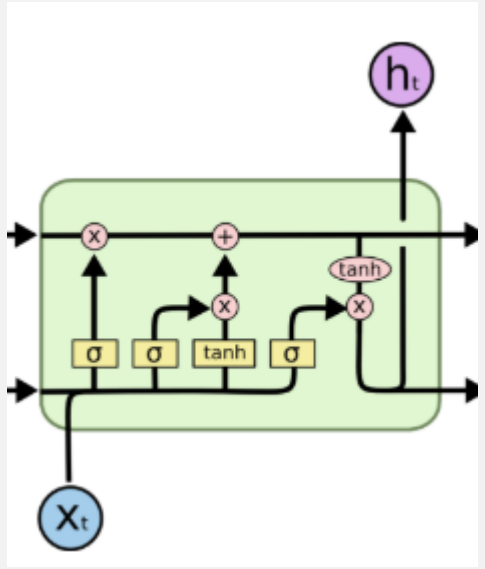


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

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

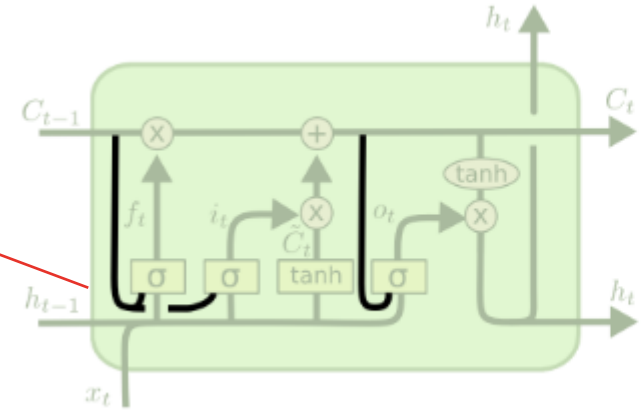
* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/ and https://baike.baidu.com

# Variants on Long Short Term Memory

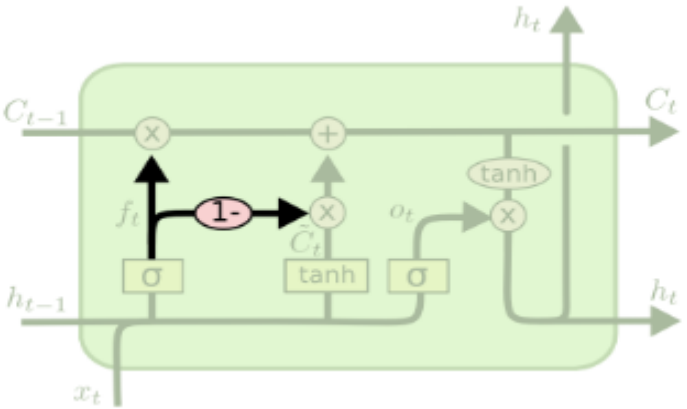



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

peephole



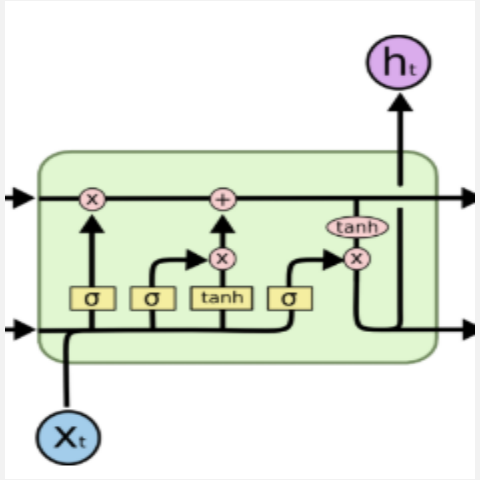

$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

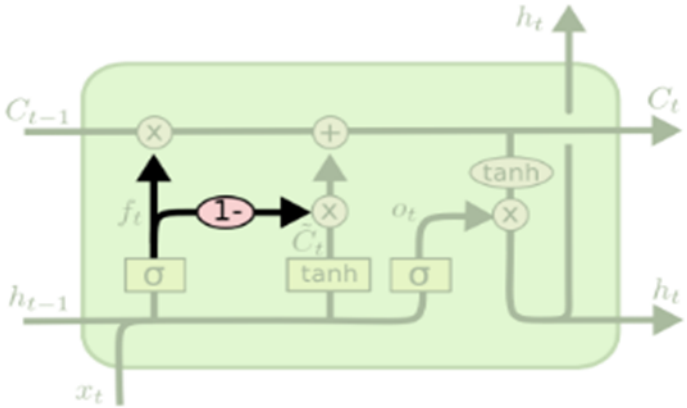



$$C_t = f_t * C_{t-1} + (\mathbf{1} - \boldsymbol{f_t}) * \tilde{C}_t$$

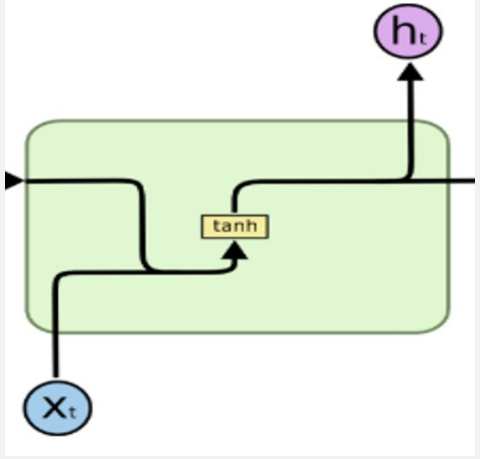
* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs

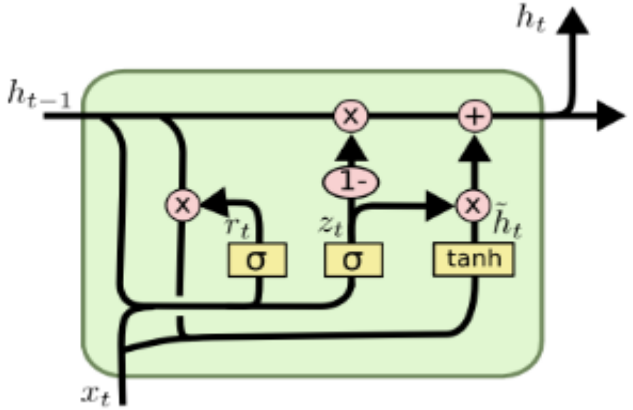
# GRU (Gated Recurrent Unit)




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

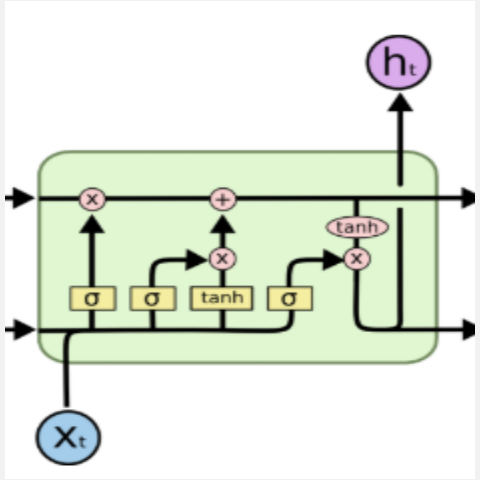


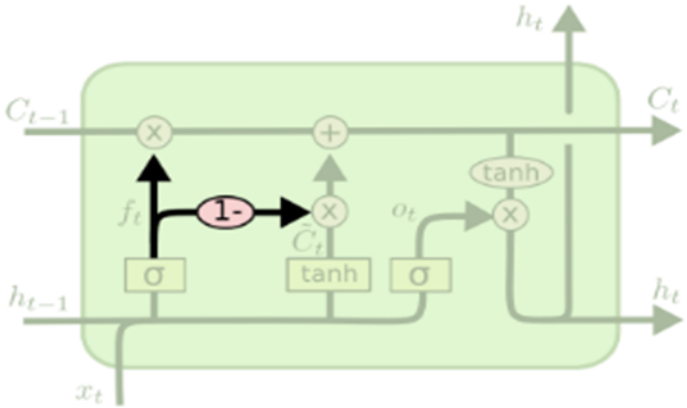




$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

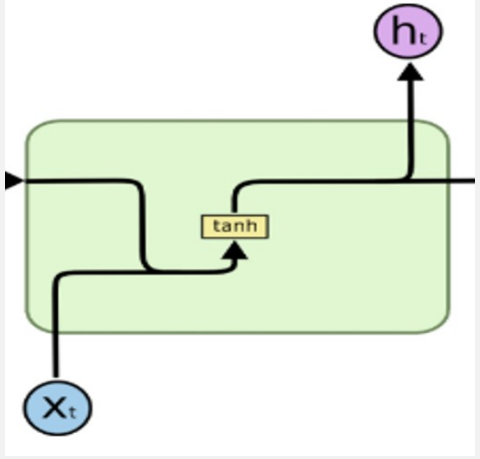
* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs

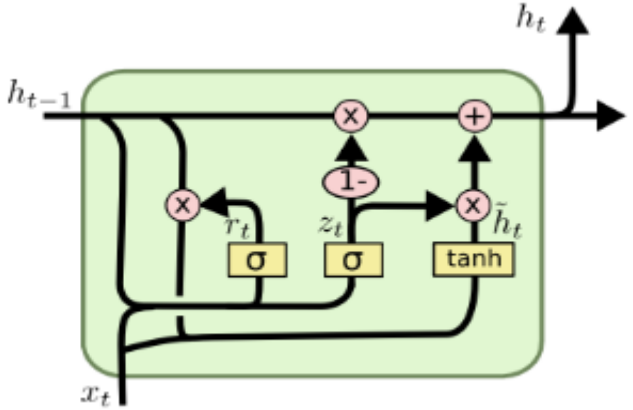
# GRU (Gated Recurrent Unit)




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




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

* Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs

# Representing words

***One-hot:***

X: Mike Guo are a good guy.

| Vocabulary | Number |
|---|---|
| A | 1 |
| And | 2 |
| … | … |
| Guo | 635 |
| Pooter | 636 |
| .. | … |
| Zulu | 10000 |

| Mike | Guo | are | … | guy |
|---|---|---|---|---|
| 0 | 0 | 0 | | 0 |
| … | 0 | 1 | | 0 |
| 0 | … | … | … | … |
| 0 | 1 | 0 | | 0 |
| … | 0 | 0 | | 1 |
| 1 | … | … | | .. |
| … | 0 | 0 | | 0 |

# Representing words

***One-hot:***

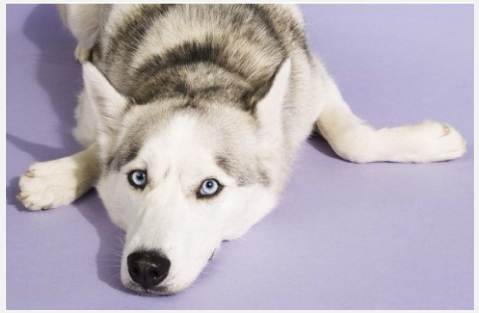
杭州 [0,0,0,0,0,0,0,1,0,……，0,0,0,0,0,0,0]

上海 [0,0,0,0,1,0,0,0,0,……，0,0,0,0,0,0,0]

宁波 [0,0,0,1,0,0,0,0,0,……，0,0,0,0,0,0,0]

北京 [0,0,0,0,0,0,0,0,0,……，1,0,0,0,0,0,0]

# The Curse of Dimensionality in classification



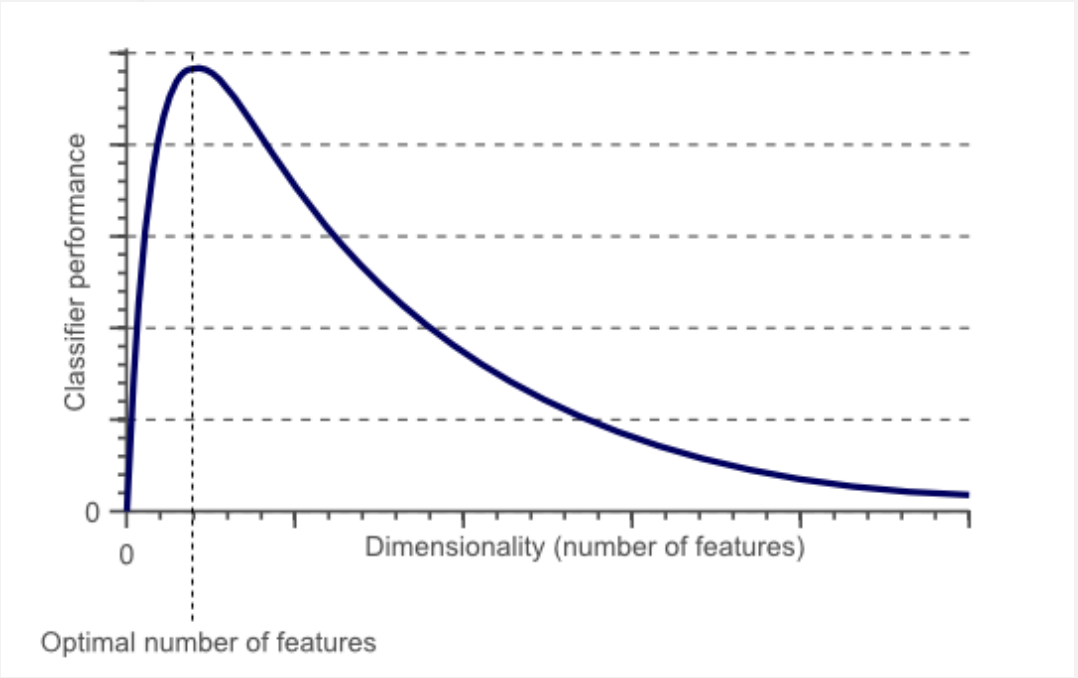
If 0.5*red + 0.3*green + 0.2*blue > 0.6 : return cat;
else return dog;

**More features**:

**dx**

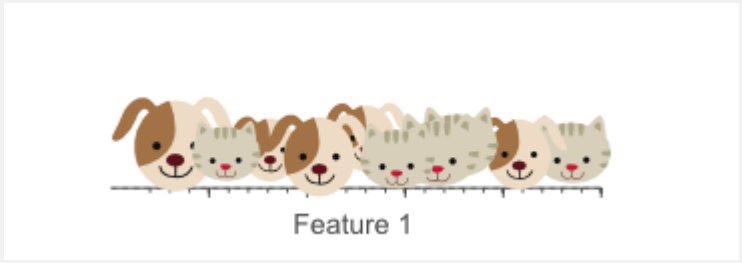
**dy**

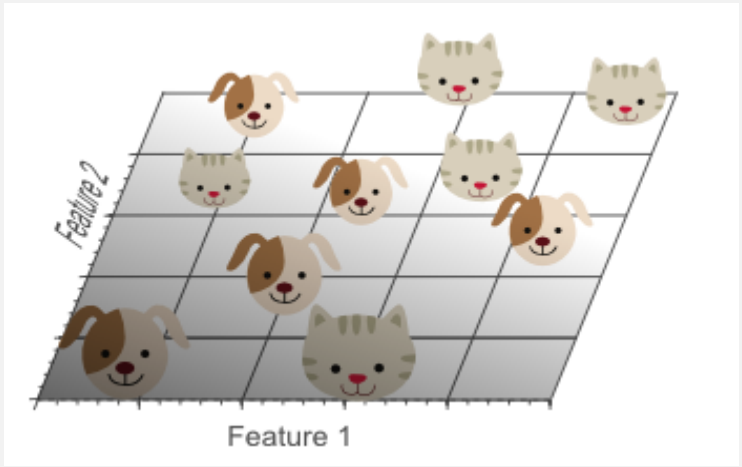
…

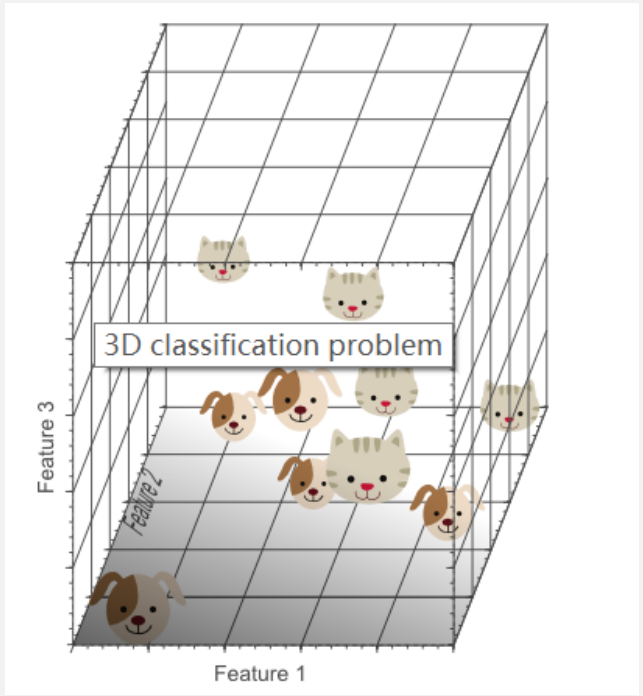


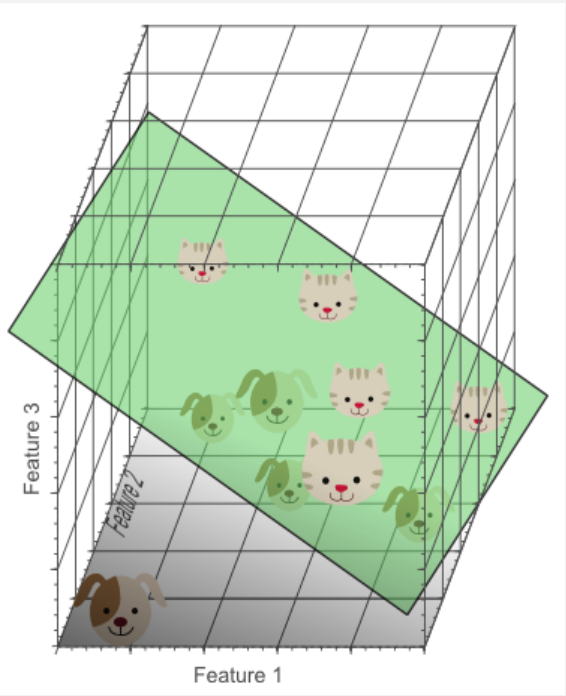

*Images from http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

# The Curse of Dimensionality in classification



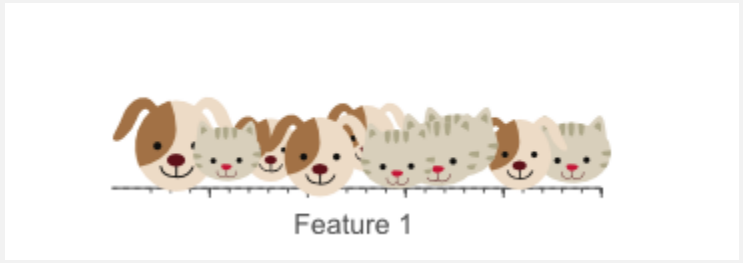



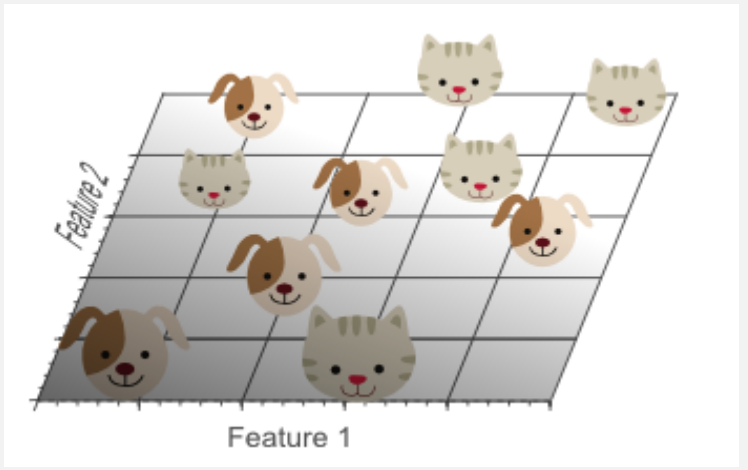




*Images from http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

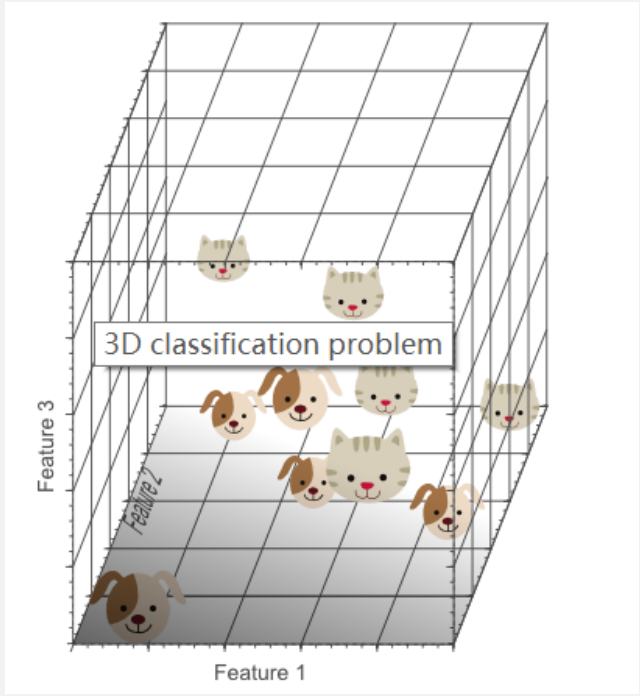
# The Curse of Dimensionality in classification




10÷5=2



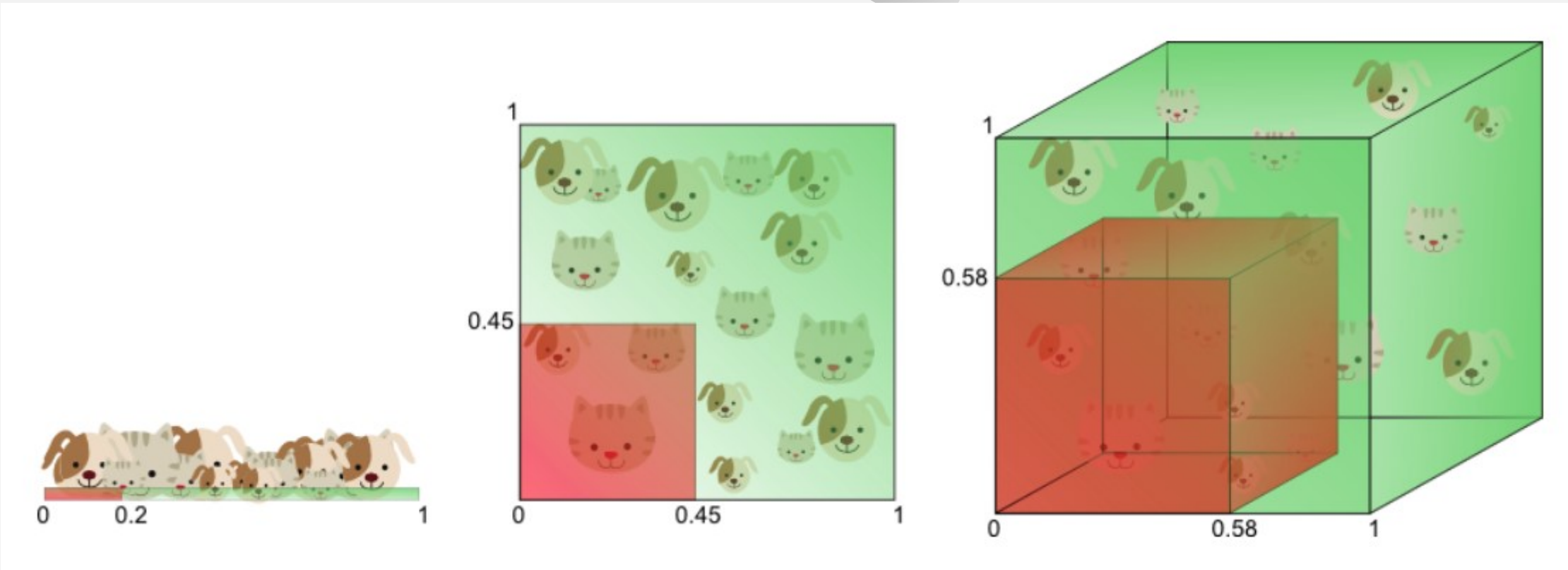

10÷25=0.4




10÷125=0.08

*Images from http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

# The Curse of Dimensionality in classification




The amount of training data needed to cover 20% of the feature range grows exponentially with the number of dimensions.

*Image is from http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

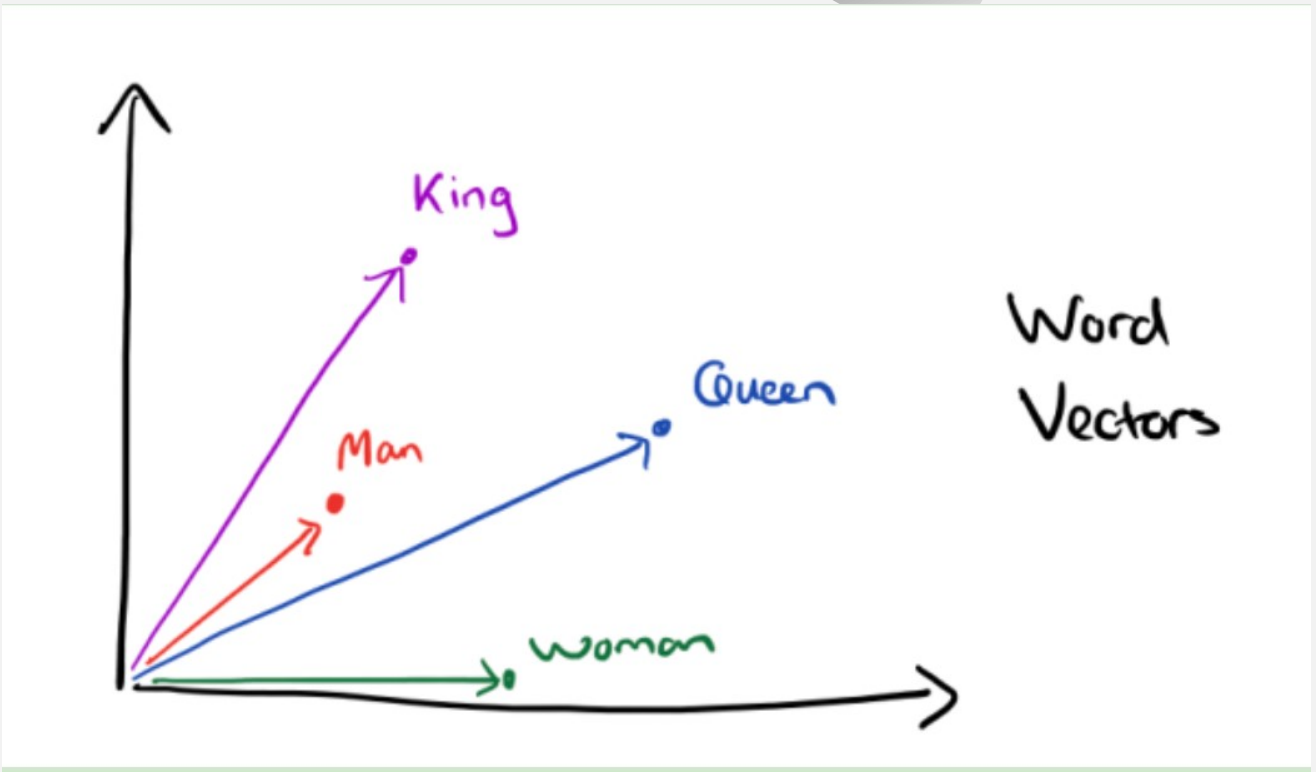
# Representing words

**Distributed representation**


*Image is from http://www.cnblogs.com/pinard/p/7160330.html

# Representing words

**Distributed representation**

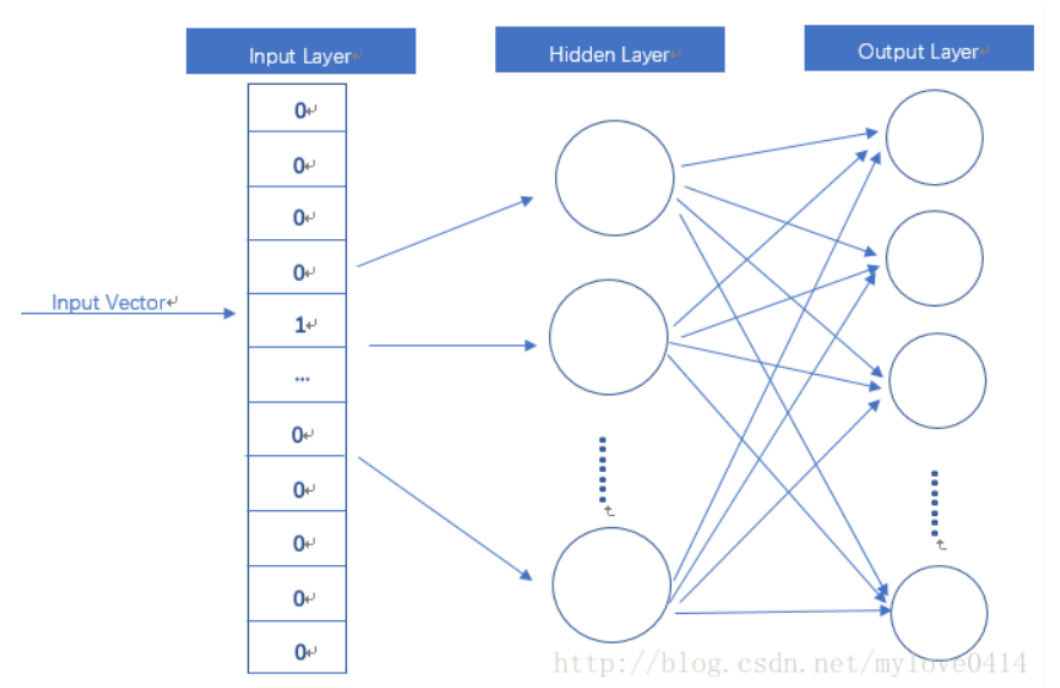



$$\vec{King} - \vec{Man} + \vec{Woman} = \vec{Queen}$$

*Image is from http://www.cnblogs.com/pinard/p/7160330.html

# Representing words

## Word2Vec，Embeddings



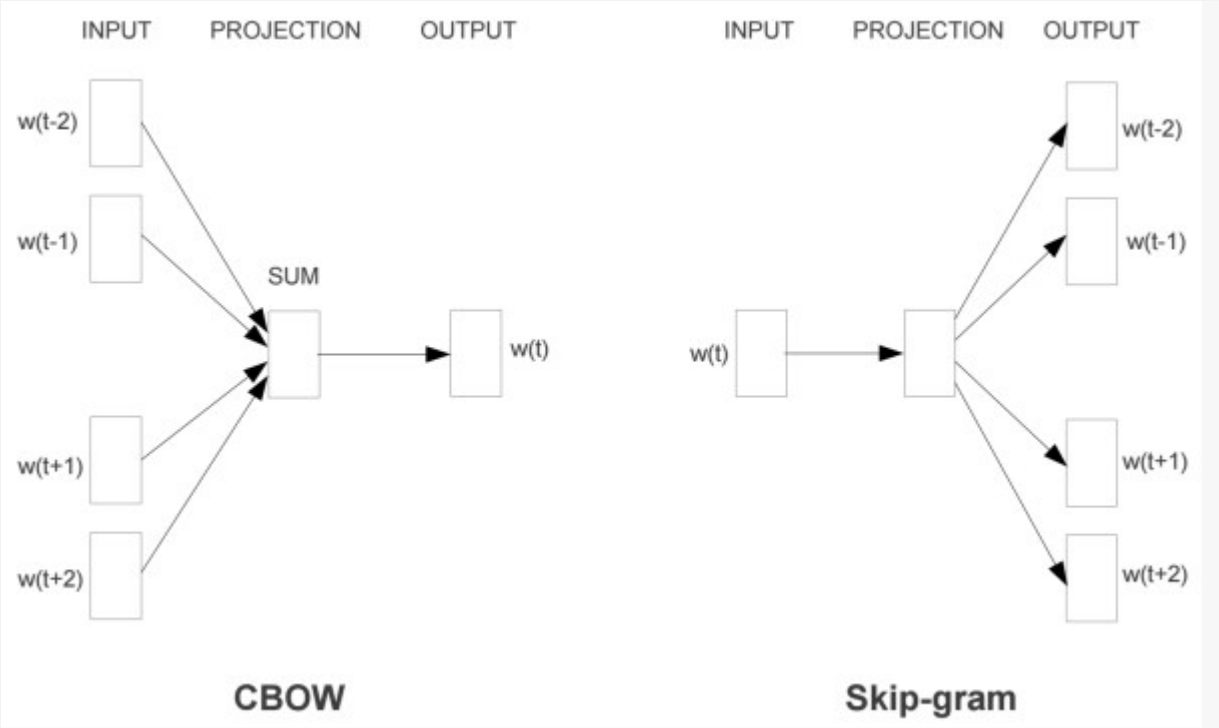

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$



*Images are from https://blog.csdn.net/mylove0414

# Representing words

## CBOW & Skip-gram




$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$



CBOW： Continuous Bag-of-Words Model

Skip-gram： Continuous Skip-gram Model

*Images are from https://blog.csdn.net/mylove0414

# Representing words

## Skip-gram

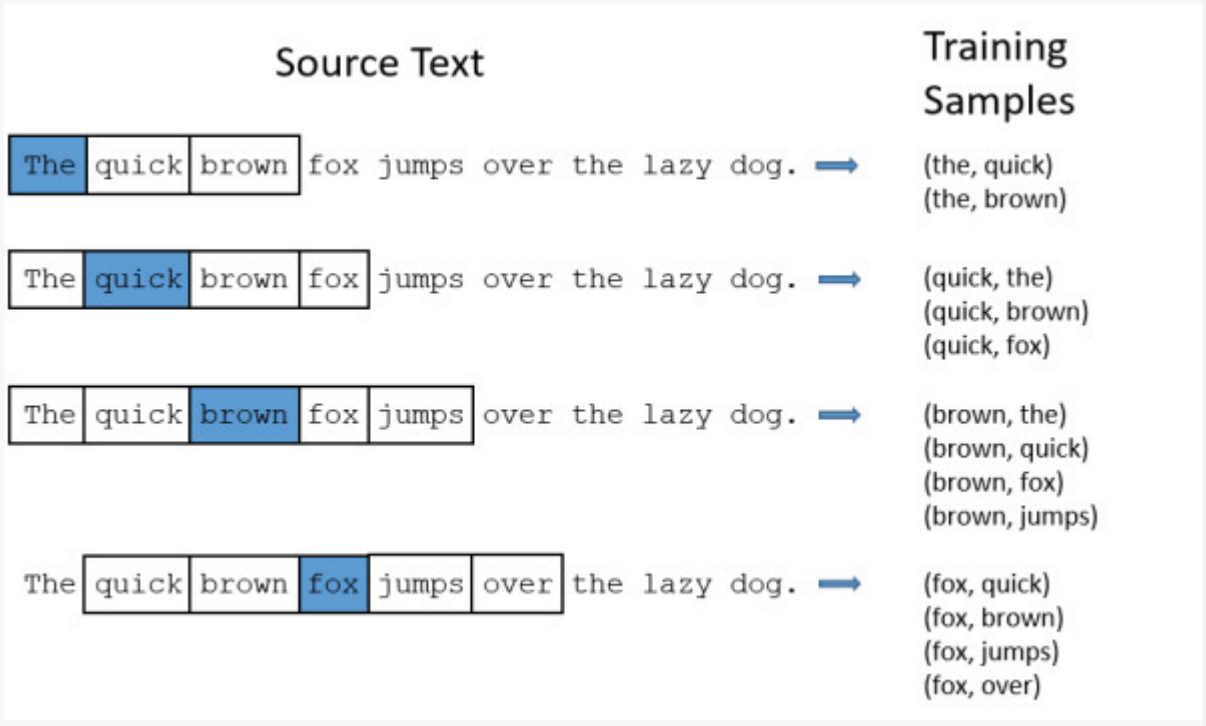
Input word

**The dog barked at the mailman**

Skip window=2 → **['The', 'dog', 'barked', 'at']**

Num skips=2

**('dog', 'barked'), ('dog', 'the')**




*Images are from https://blog.csdn.net/mylove0414

# Representing words

## Skip-gram

**The quick brown fox jumps over lazy dog**

Skip window=2

Union

Soviet

Russia

kangaroo

watermelon

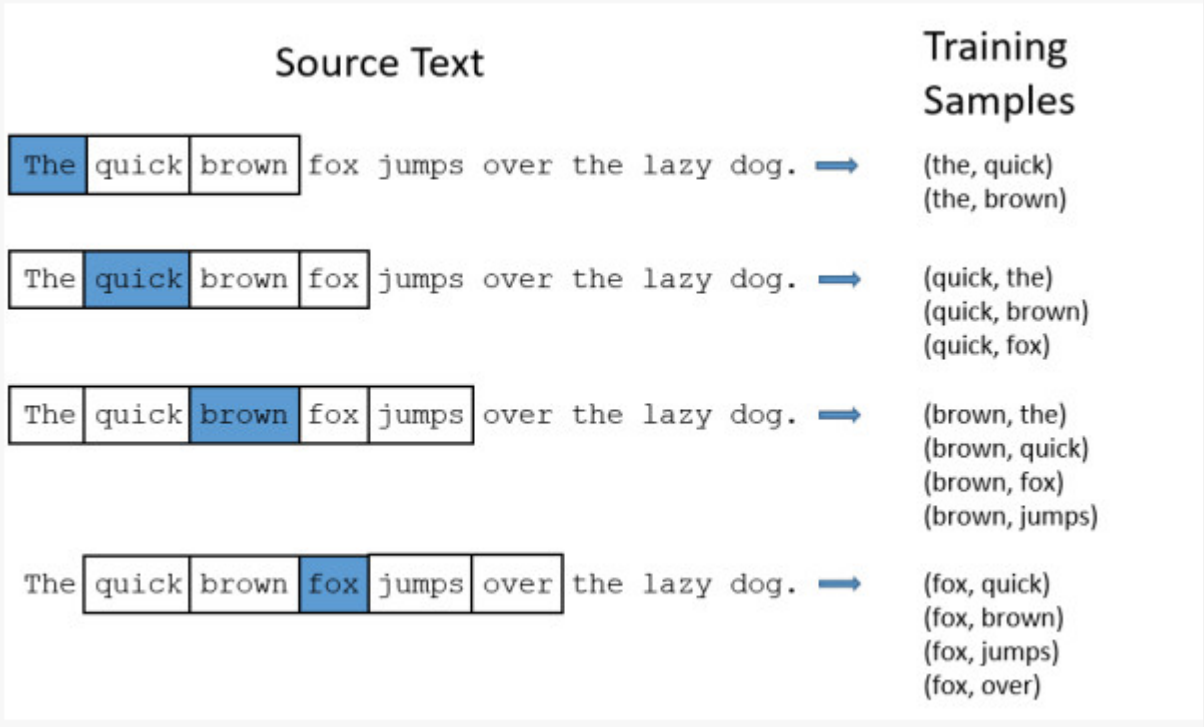



*Image is from https://www.leiphone.com/news/201706/PamWKpfRFEI42McI.html

# Representing words

## Skip-gram

**The dog barked at the mailman**

[0,1,0,0,0]

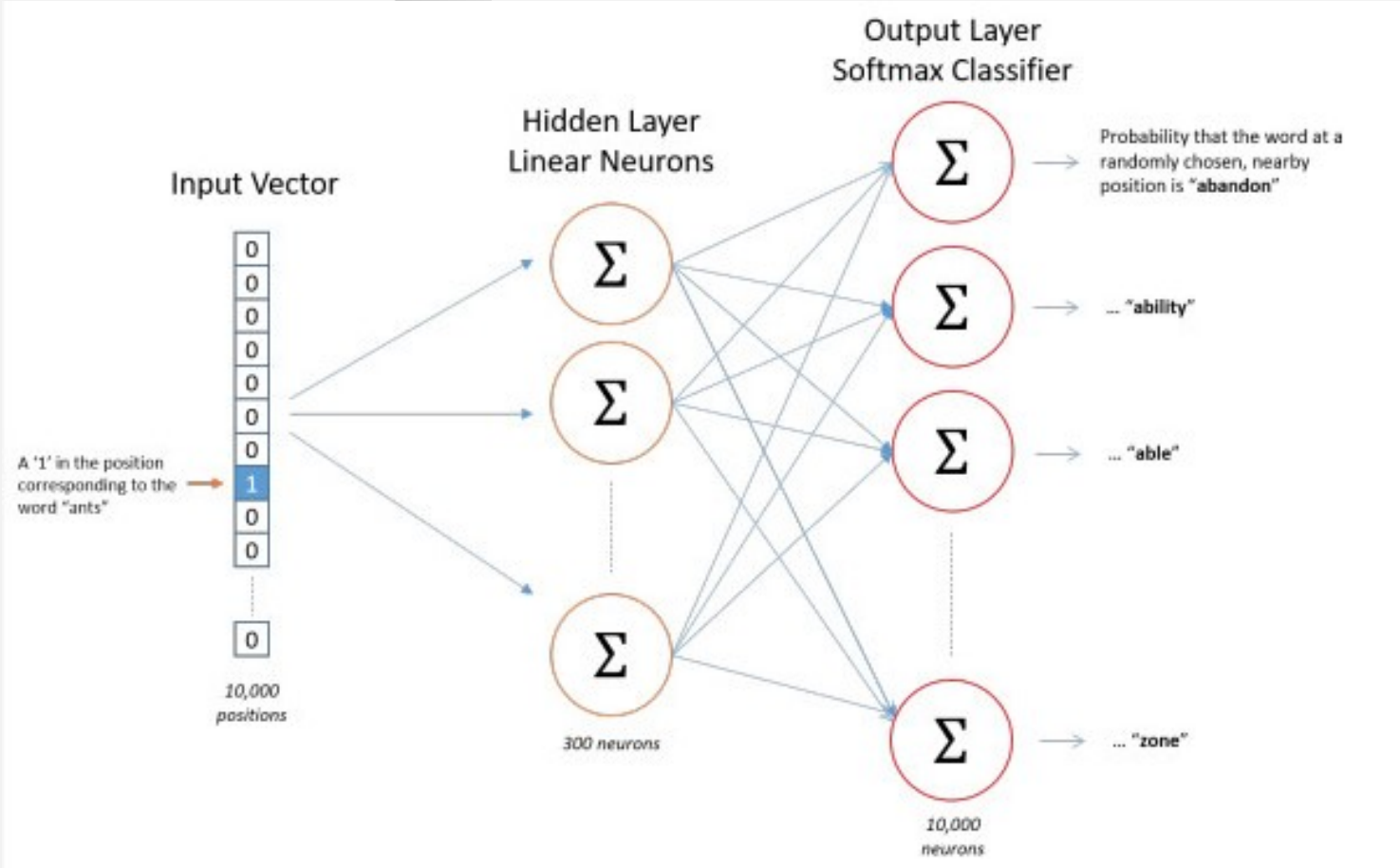
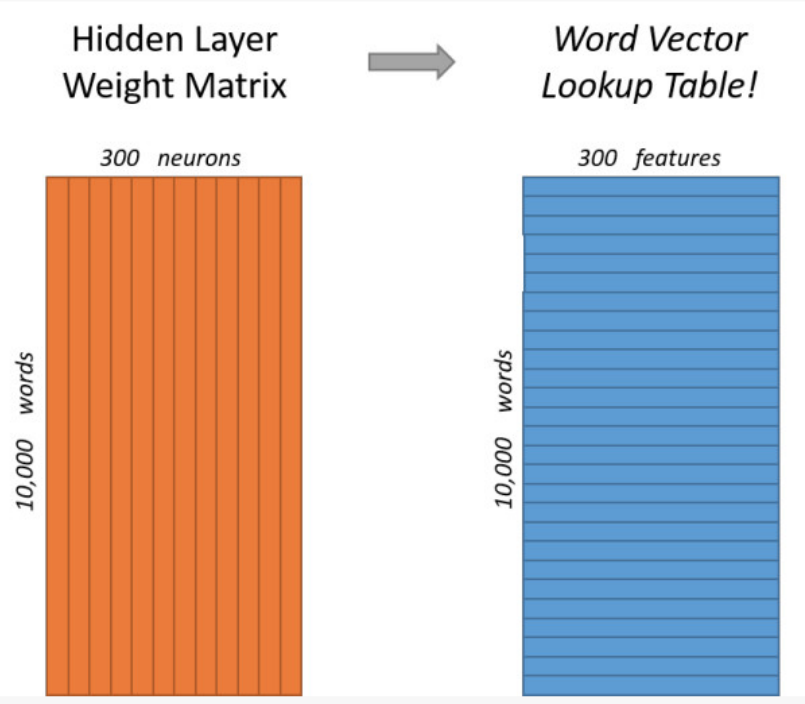



*Image is from https://www.leiphone.com/news/201706/PamWKpfRFEI42McI.html

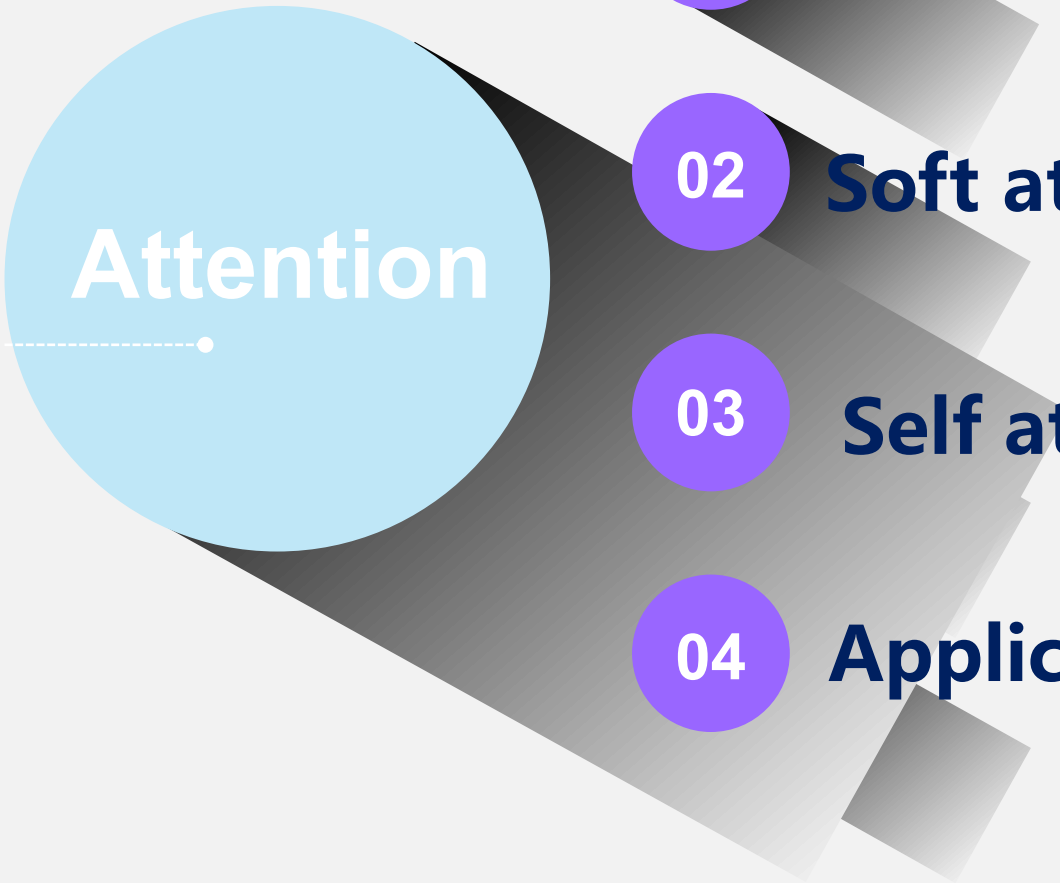
# Representing words

## Skip-gram




$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

*Image is from https://www.leiphone.com/news/201706/PamWKpfRFEI42McI.html



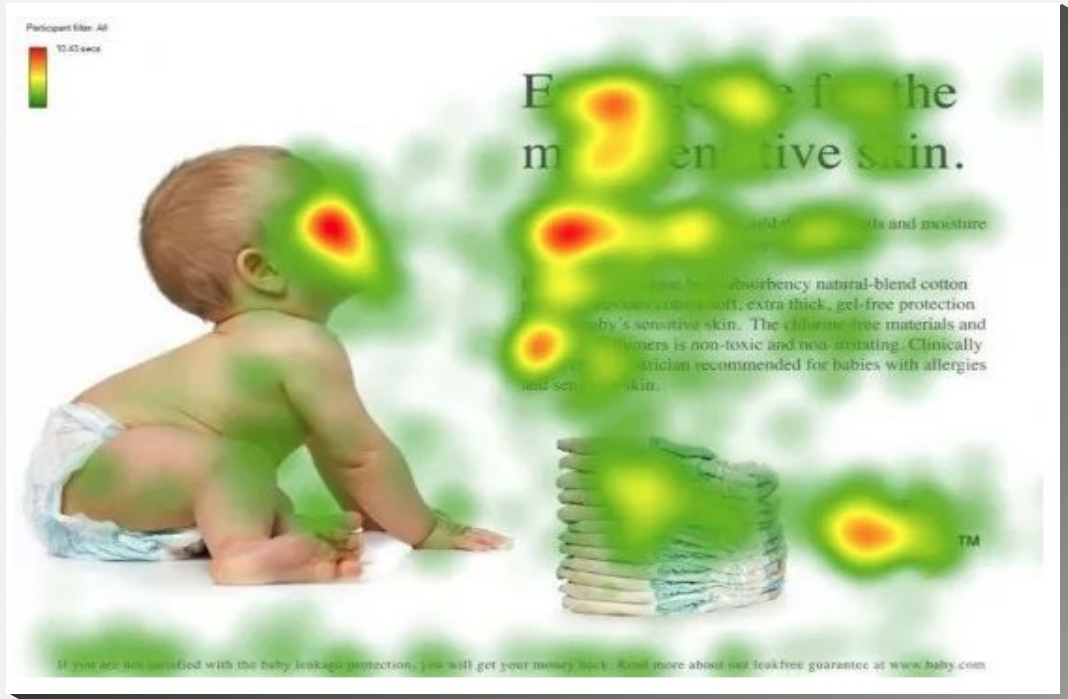

01 What are the attention mechanisms ?

02 Soft attention and hard attention

03 Self attention

04 Application of attention mechanism

# What are the attention mechanisms ?



**YI REN WEI BEN**

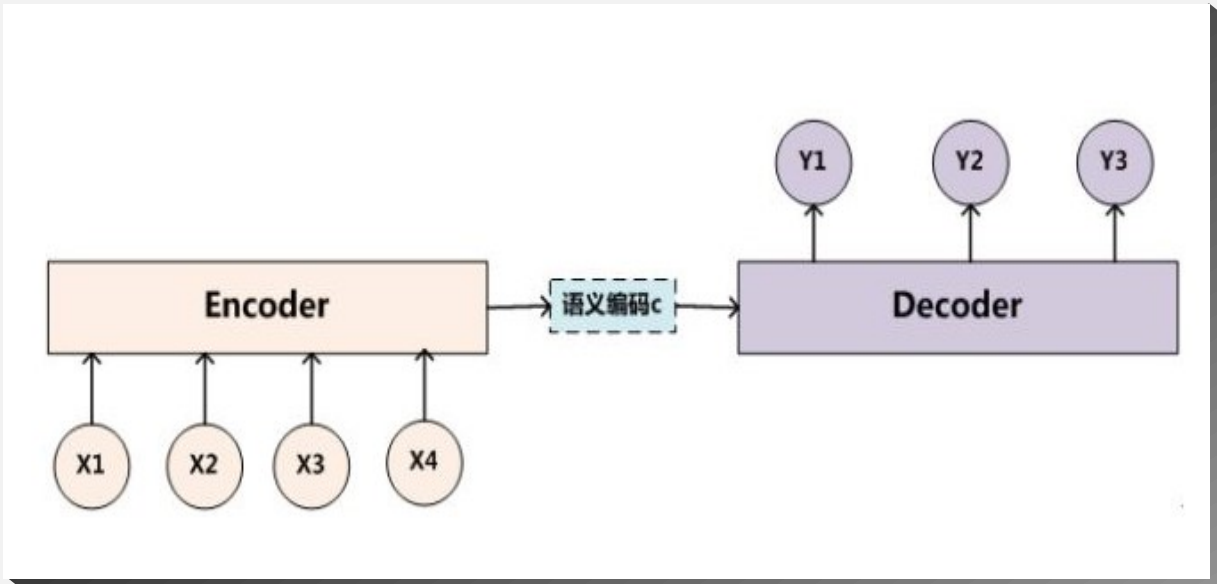
Attention Mechanisms in Neural Networks are (very) loosely based on the visual attention mechanism found in humans.

**Bottom-up**

**Top-down**

*Image is from https://zhuanlan.zhihu.com/p/3760

# Encoder-Decoder



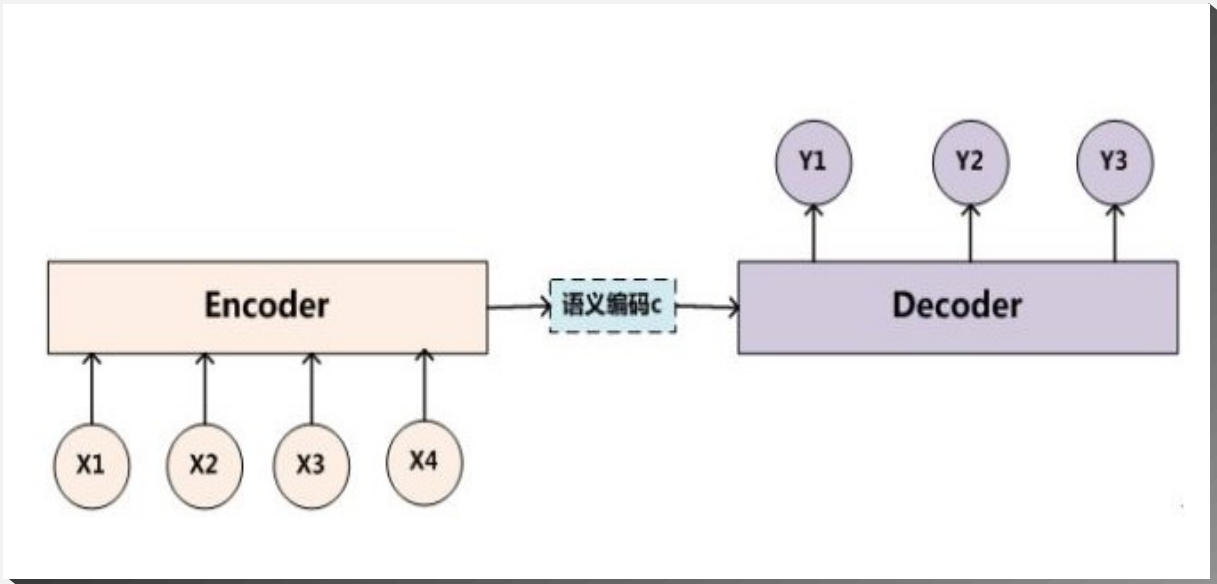

## Seq2Seq

Sequence-to-sequence (seq2seq) models have enjoyed great success in a variety of tasks.

- Source=<$x_1$ , $x_2$ … $x_m$>
- Target=< $y_1$ , $y_2$ … $y_m$ >
- C=F($x_1$ , $x_2$ … $x_m$)
- $y_i$= G(C, $y_1$ , $y_2$ … $y_{i-1}$ )

*Image is from https://zhuanlan.zhihu.com/p/37601161

# Attention mechanism




**Soft attention**

Multiplies features with a (soft) mask of values between zero and one.

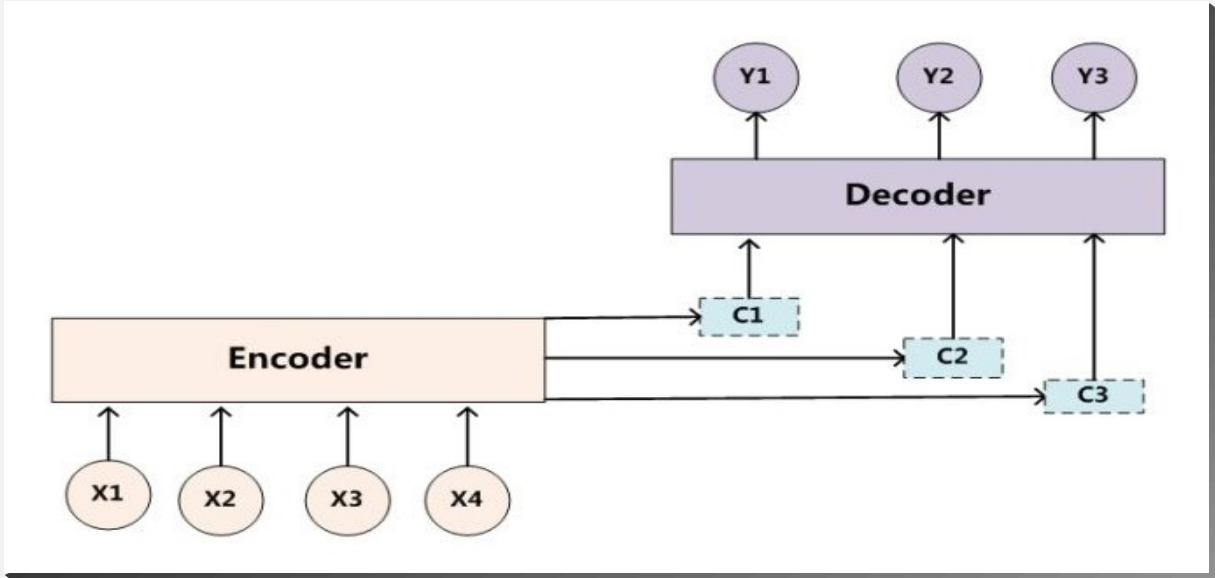
- $y_1 = f(C)$
- $y_2 = f(C, y_1)$
- $y_3 = f(C, y_1, y_2)$

Tom chase Jerry

汤姆追逐杰瑞

*Image is from https://zhuanlan.zhihu.com/p/37601161

# Attention mechanism




$$c_i = \sum_{j=1}^{Lx} a_{ij} h_j$$

## Soft attention

Multiplies features with a (soft) mask of values between zero and one.

- (Tom,0.3) (Chase,0.2) (Jerry,0.5)
- $y_1 = f(C_1)$
- $y_2 = f(C_2, y_1)$
- $y_3 = f(C_3, y_1, y_2)$

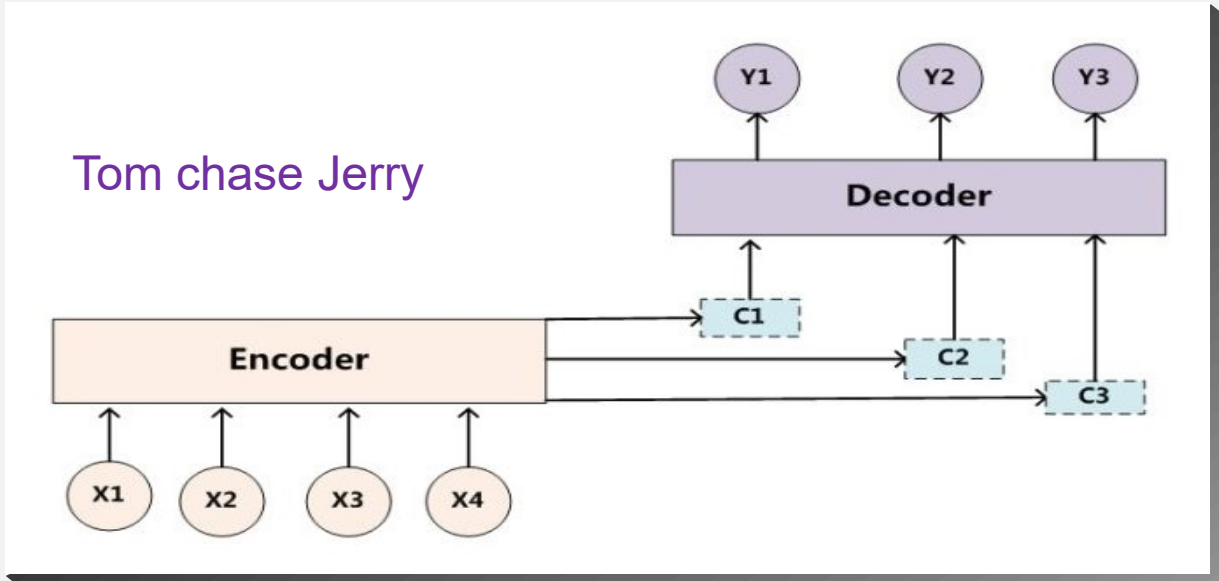
$$0.2 * f2(Chase), 0.2 * f2("Jerry"))$$

$$\text{om}"), 0.7 * f2(Chase), 0.1 * f2("Jerry"))$$

$$2("Tom"), 0.2 * f2(Chase), 0.5 * f2("Jerry"))$$

*Image is from https://zhuanlan.zhihu.com/p/37601161

# Attention mechanism




$$c_i = \sum_{j=1}^{Lx} a_{ij} h_j$$

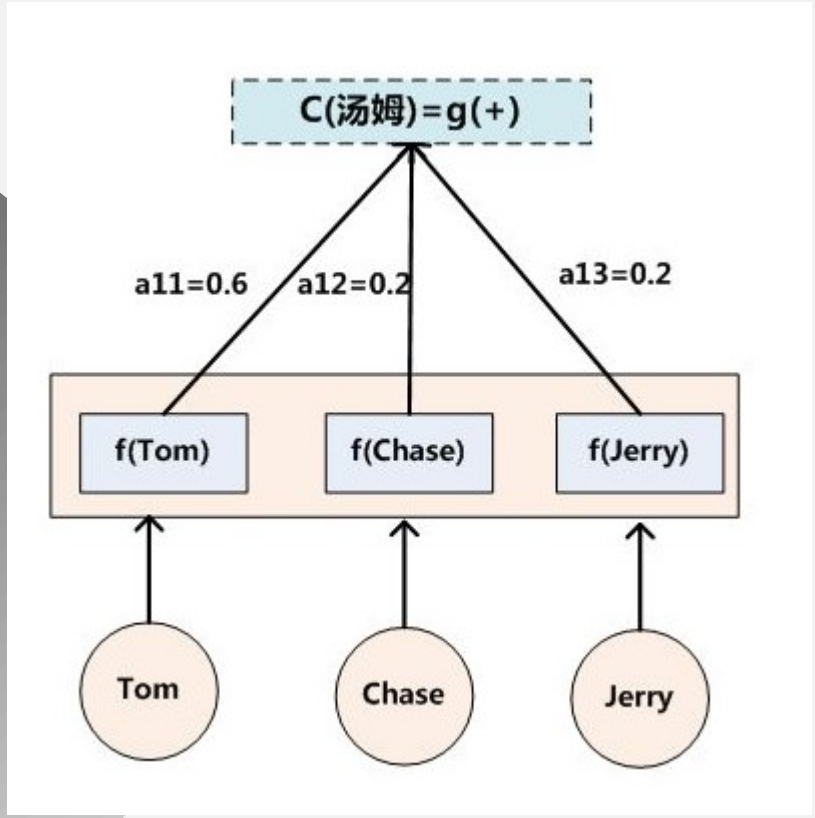
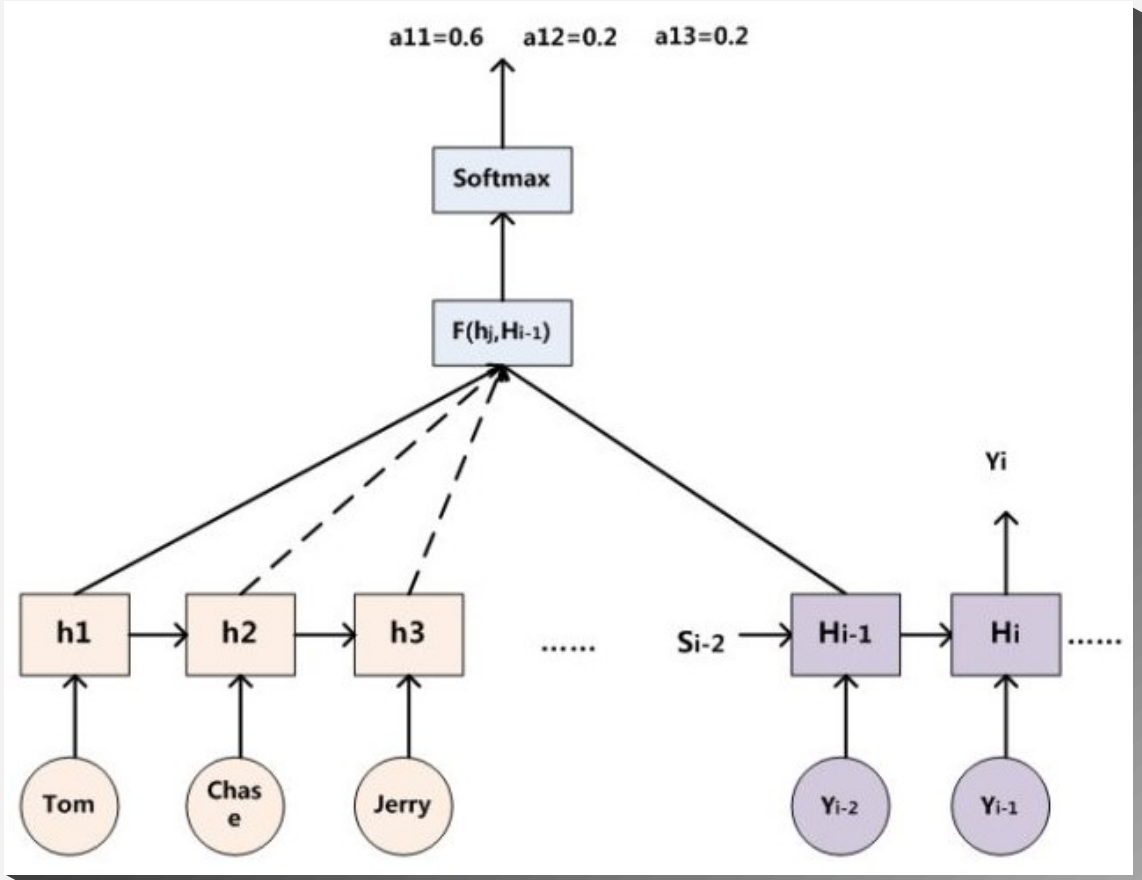



*Image is from https://zhuanlan.zhihu.com/p/37601161

# Attention mechanism



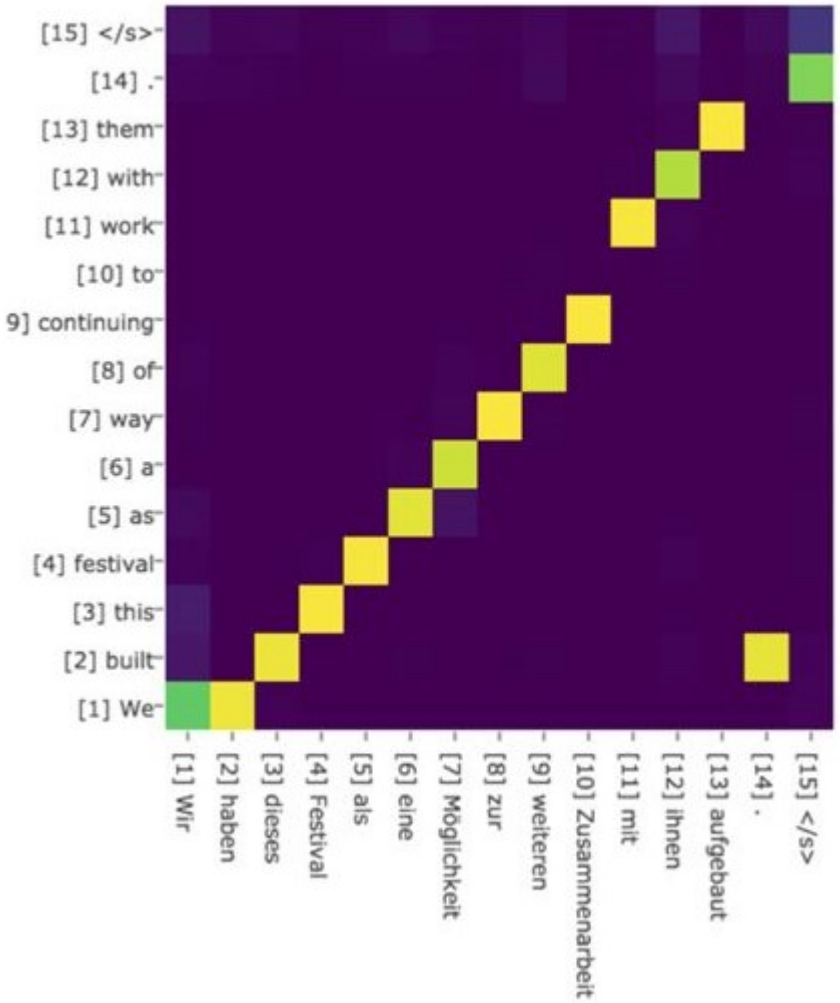
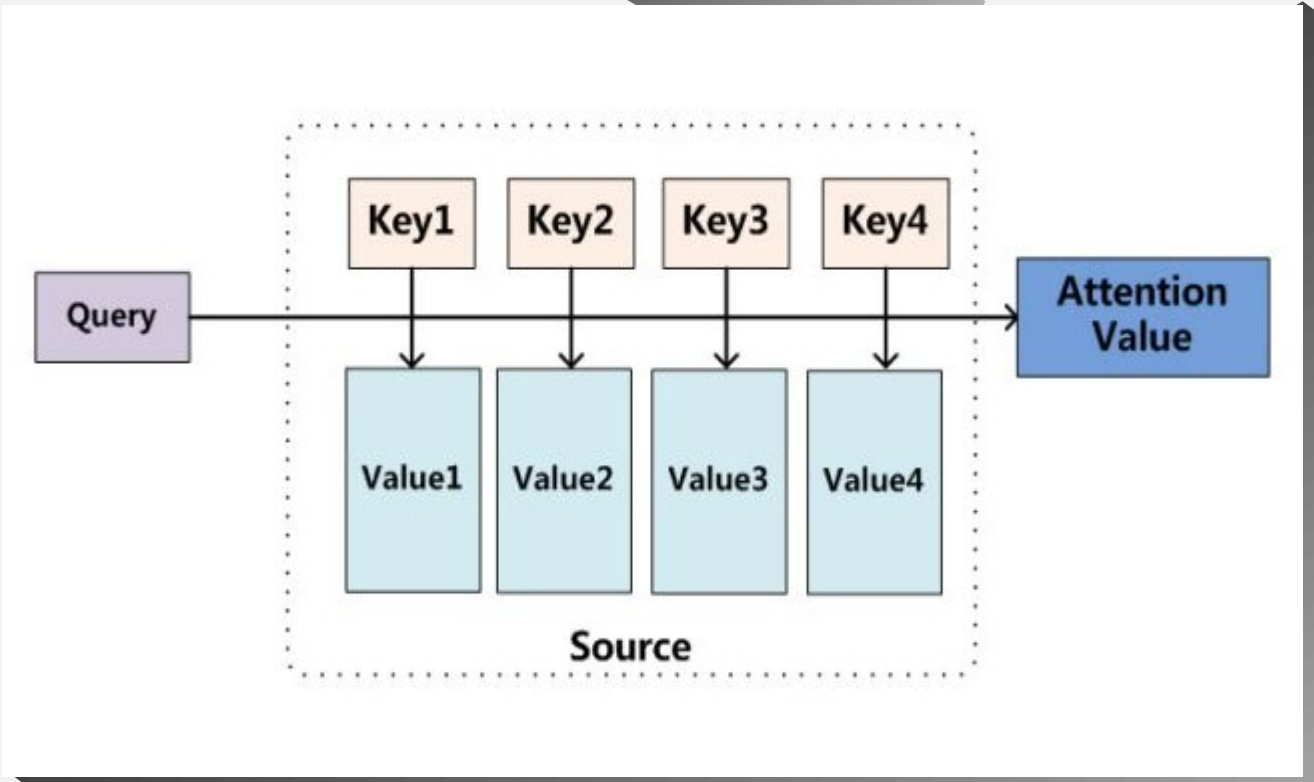




*Image is from https://zhuanlan.zhihu.com/p/37[illegible]

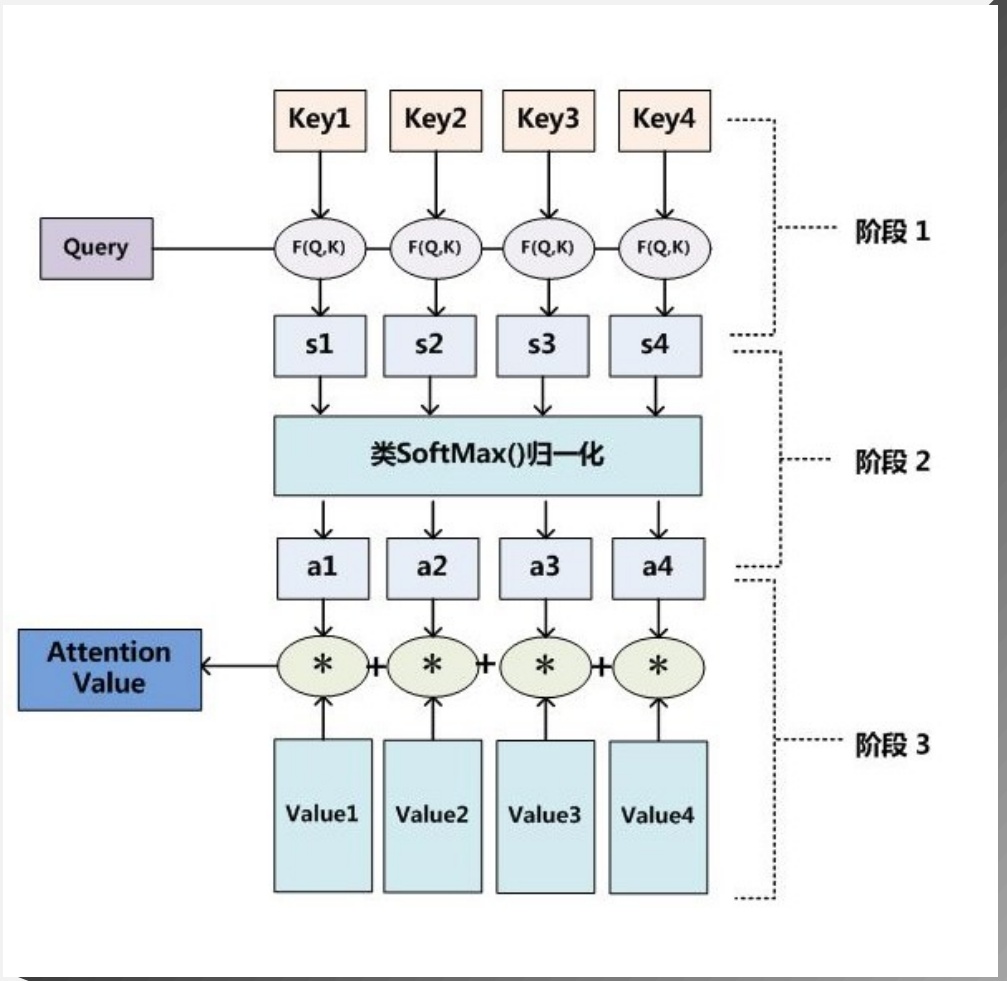
# Attention mechanism




$$Attention(Query, Source) = \sum_{i=1}^{L_x} Similarity(Query, Key_i) * Value_i$$

*Image is from https://zhuanlan.zhihu.com/p/37601161

# Attention mechanism




$$Similarity(Query, Key_i) = \text{Query} * Key_i$$

$$Similarity(Query, Key_i) = \frac{Query * Key_i}{||Query|| * ||Key_i||}$$

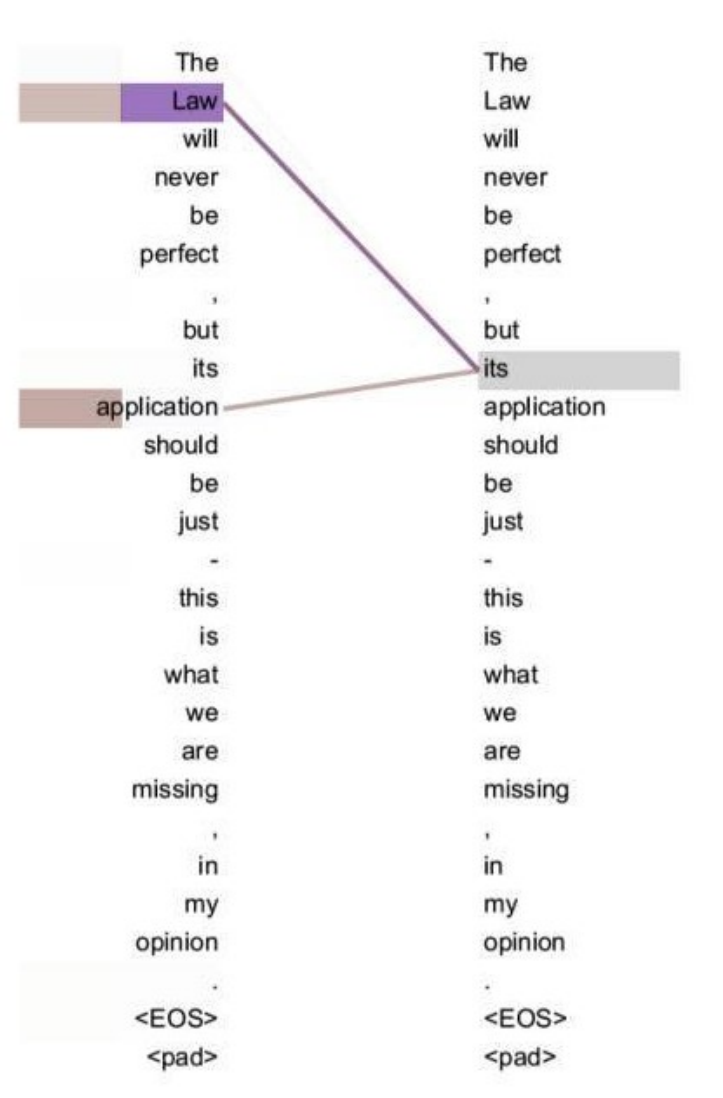
$$Similarity(Query, Key_i) = MLP(Query, Key_i)$$

$$a_i = softmax(Sim_i) = \frac{e^{Sim_i}}{\sum_{j=1}^{L_x} e^{Sim_i}}$$

$$Attention(Query, Source) = \sum_{j=1}^{L_x} a_i * Value_i$$

*Image is from https://zhuanlan.zhihu.com/p/37601161

# Self Attention



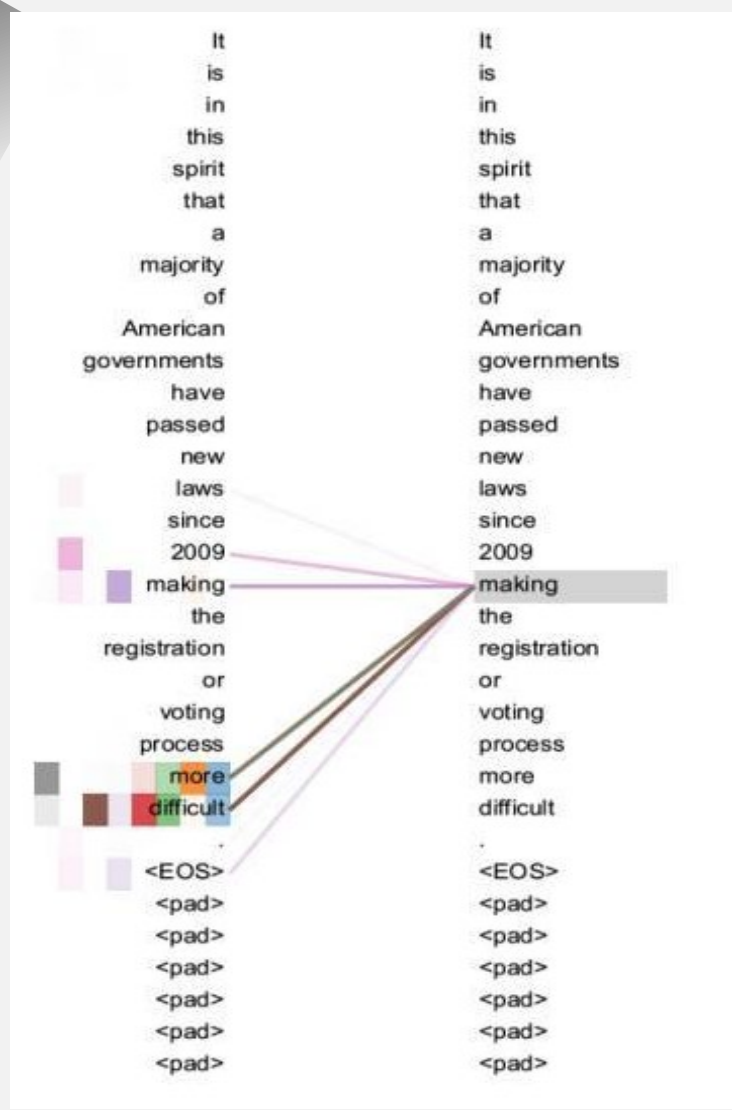
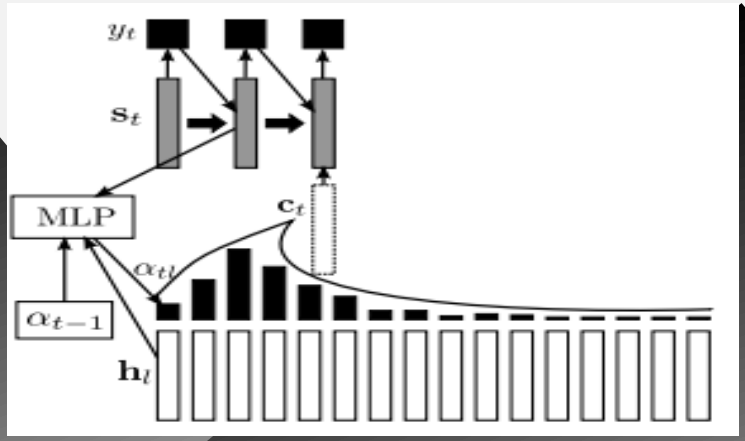




*Image is from https://zhuanlan.zhihu.com/p/37601161

# Application of attention mechanism


Image caption


Text2image




Speech recognition

# Image caption


**Image caption**

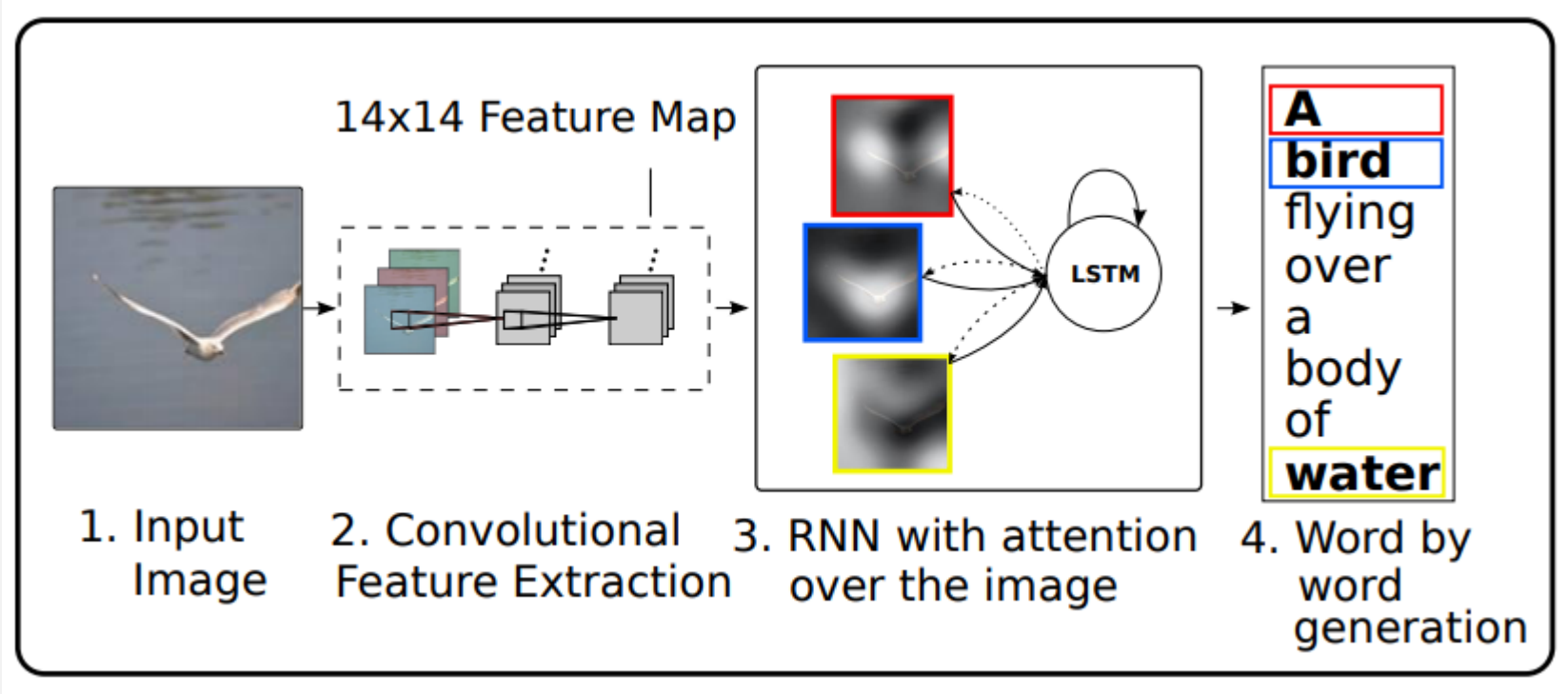
Automatically generating captions of an image is a task very close to the heart of scene understanding

**Encoder-decoder**

**Attention mechanism**

Xu, Kelvin, et al. Show, attend and tell: Neural image caption generation with visual attention, In ICML, 2015.

# Show, attend and tell

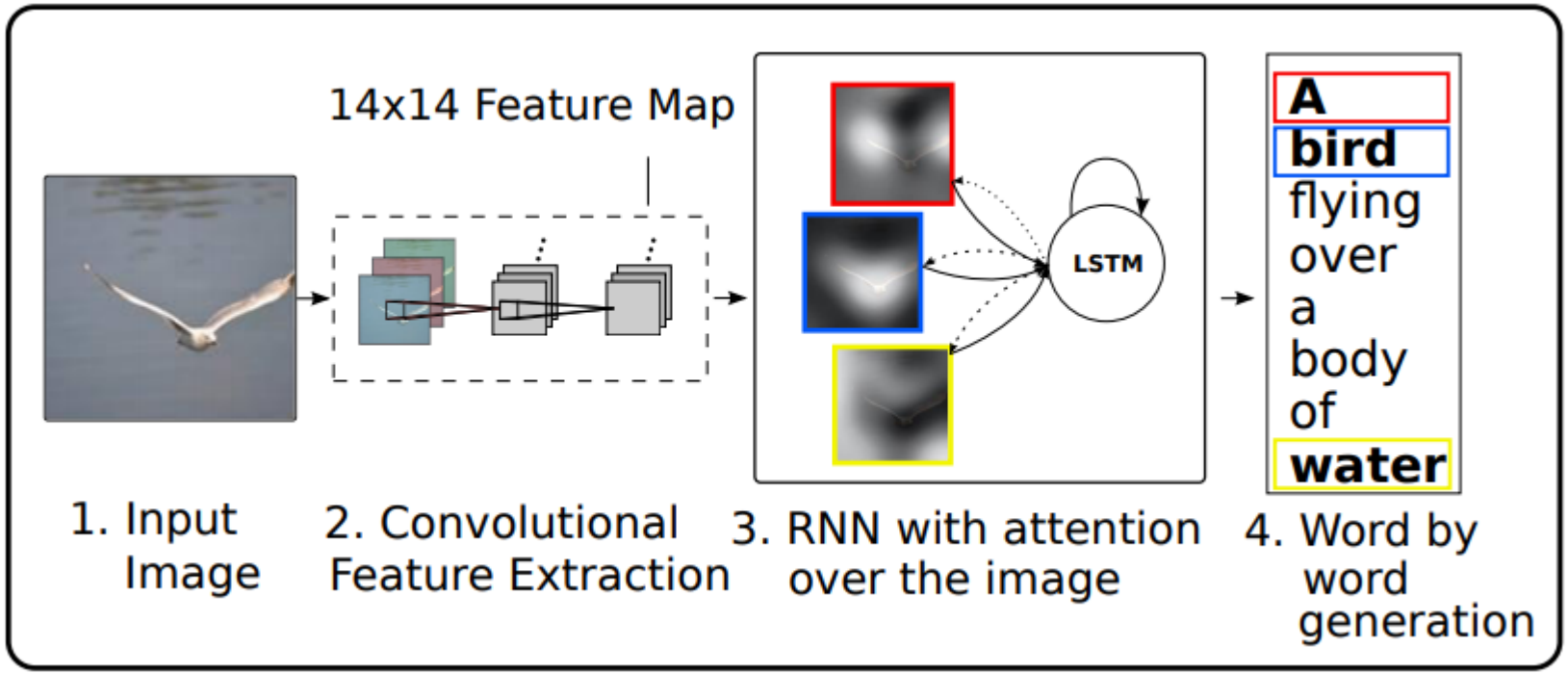



$$a = a_1, \ldots, a_L, \qquad a_i \in R^D$$

| VGG19 | | |
|---|---|---|
| 序号 | 层结构 | |
| 1 | conv1-1 | 1 |
| 2 | relu1-1 | |
| 3 | conv1-2 | 2 |
| 4 | relu1-2 | |
| 5 | pool1 | |
| 6 | conv2-1 | 3 |
| 7 | relu2-1 | |
| 8 | conv2-2 | 4 |
| 9 | relu2-2 | |
| 10 | pool2 | |
| 11 | conv3-1 | 5 |
| 12 | relu3-1 | |
| 13 | conv3-2 | 6 |
| 14 | relu3-2 | |
| 15 | conv3-3 | 7 |
| 16 | relu3-3 | |
| 17 | conv3-4 | 8 |
| 18 | relu3-4 | |
| 19 | pool3 | |
| 20 | conv4-1 | 9 |
| 21 | relu4-1 | |
| 22 | conv4-2 | 10 |
| 23 | relu4-2 | |
| 24 | conv4-3 | 11 |
| 25 | relu4-3 | |
| 26 | conv4-4 | 12 |
| 27 | relu4-4 | |
| 28 | pool4 | |
| 29 | conv5-1 | 13 |
| 30 | relu5-1 | |
| 31 | conv5-2 | 14 |
| 32 | relu5-2 | |
| 33 | conv5-3 | 15 |
| 34 | relu5-3 | |
| 35 | conv5-4 | 16 |
| 36 | relu5-4 | |
| 37 | pool5 | |
| 38 | fc6(4096) | 17 |
| 39 | relu6 | |
| 40 | fc7(4096) | 18 |
| 41 | relu7 | |
| 42 | fc8(1000) | 19 |
| 43 | prob(softmax) | |

Xu, Kelvin, et al. Show, attend and tell: Neural image caption generation with visual attention, In ICML, 2015.

# Show, attend and tell




**Hard attention**

$$\hat{z}_t = \sum_i s_{t,i} \mathbf{a}_i$$

**Soft attention**

$$L_{p(s_t|a)}[\hat{z}_t] = \sum_{i=1}^{L} \alpha_{t,i} \mathbf{a}_i$$

$$a = a_1, \ldots, a_L, \qquad a_i \in R^D$$

Xu, Kelvin, et al. Show, attend and tell: Neural image caption generation with visual attention, In ICML, 2015.

# Datasets

**IMAGE 6620094641**

The entity refers to other

**SENTENCES**

1. Two combatants in a rugby stadium making a move on each other , one wearing green with ball and defended by # 5 in the maroon colors .
2. Two muddy athletes from opposing teams going head to head in a game of rugby while fans watch expectantly .
3. Two soccer players on opposing teams trying to keep the soccer ball from one another .
4. A rugby player is running with the ball as a defender moves to intervene .
5. Two men from opposing teams with dirty uniforms in a match of rugby .

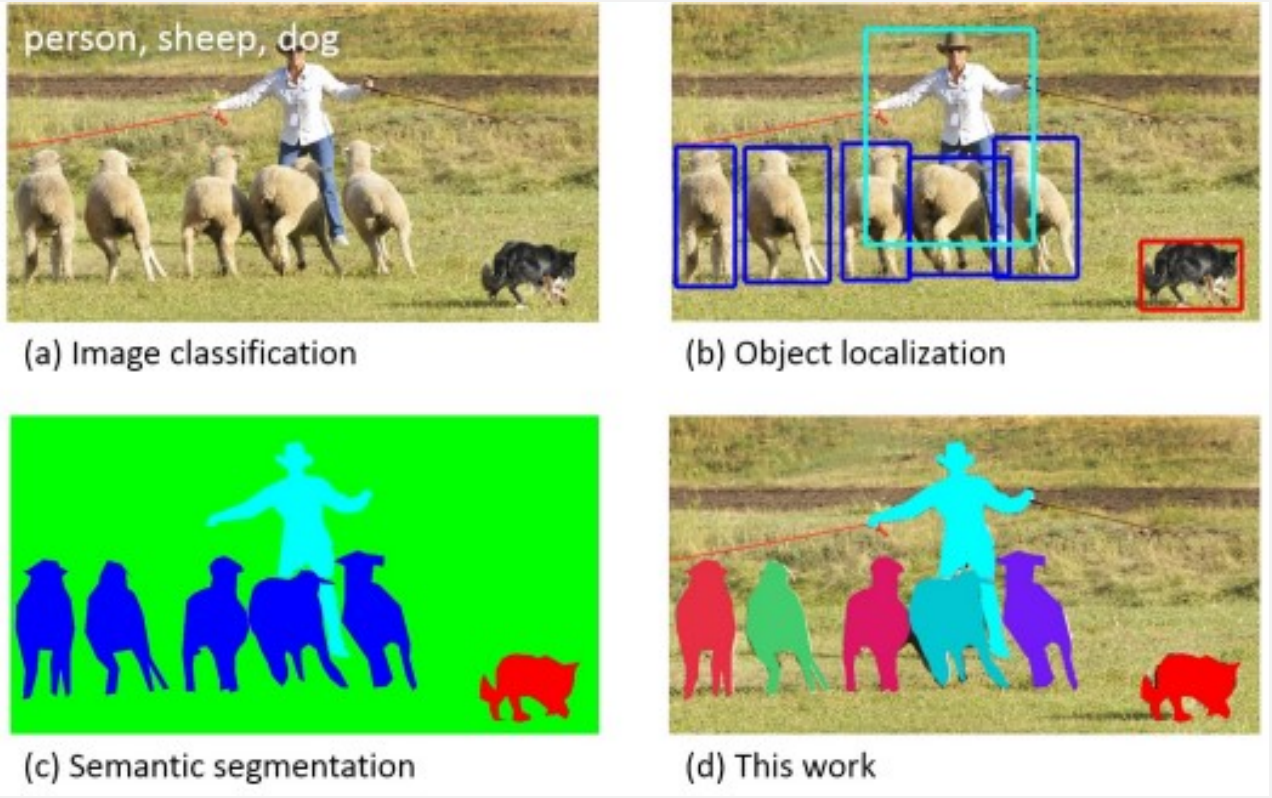
## Flickr30k Entities

The Flickr30k dataset has become a standard benchmark for sentence-based image description.

- 276k manually annotated bounding boxes
- 158k captions
- 244k coreference chains

*Image is from http://web.engr.illinois.edu/~bplumme2/Flickr30kEntities/

# Datasets




## Microsoft COCO

COCO is a large-scale object detection, segmentation, and captioning dataset.

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

*Image is from http://web.engr.illinois.edu/~bplumme2/Flickr30kEntities/

# Show, attend and tell

*Table 1.* BLEU-1,2,3,4/METEOR metrics compared to other methods, † indicates a different split, (—) indicates an unknown metric, ∘ indicates the authors kindly provided missing metrics by personal communication, Σ indicates an ensemble, $a$ indicates using AlexNet

| | | BLEU | | | | |
|---|---|---|---|---|---|---|
| Dataset | Model | B-1 | B-2 | B-3 | B-4 | METEOR |
| Flickr8k | Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$ | 63 | 41 | 27 | — | — |
| | Log Bilinear (Kiros et al., 2014a)$^{\circ}$ | 65.6 | 42.4 | 27.7 | 17.7 | 17.31 |
| | Soft-Attention | **67** | 44.8 | 29.9 | 19.5 | 18.93 |
| | Hard-Attention | **67** | **45.7** | **31.4** | **21.3** | **20.30** |
| Flickr30k | Google NIC$^{\dagger\circ\Sigma}$ | 66.3 | 42.3 | 27.7 | 18.3 | — |
| | Log Bilinear | 60.0 | 38 | 25.4 | 17.1 | 16.88 |
| | Soft-Attention | 66.7 | 43.4 | 28.8 | 19.1 | **18.49** |
| | Hard-Attention | **66.9** | **43.9** | **29.6** | **19.9** | 18.46 |
| COCO | CMU/MS Research (Chen & Zitnick, 2014)$^{a}$ | — | — | — | — | 20.41 |
| | MS Research (Fang et al., 2014)$^{\dagger a}$ | — | — | — | — | 20.71 |
| | BRNN (Karpathy & Li, 2014)$^{\circ}$ | 64.2 | 45.1 | 30.4 | 20.3 | — |
| | Google NIC$^{\dagger\circ\Sigma}$ | 66.6 | 46.1 | 32.9 | 24.6 | — |
| | Log Bilinear$^{\circ}$ | 70.8 | 48.9 | 34.4 | 24.3 | 20.03 |
| | Soft-Attention | 70.7 | 49.2 | 34.4 | 24.3 | **23.90** |
| | Hard-Attention | **71.8** | **50.4** | **35.7** | **25.0** | 23.04 |


Xu, Kelvin, et al. Show, attend and tell: Neural image caption generation with visual attention, In ICML, 2015.

# Show, attend and tell

Bleu

Example 1.
Candidate 1： It is a guide to action which ensures that the military always obeys the commands of the party.

Candidate 2: It is to insure the troops forever hearing the activity guidebook that party direct.

Reference 1: It is a guide to action that ensures that the military will forever heed Party commands.

modified n-gram precision

$$p_n = \frac{\sum_{C \in \{Candidate\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidate\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram)}$$

Example 2.
Candidate: the the the the the the the.

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

$$Bleu = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Xu, Kelvin, et al. Show, attend and tell: Neural image caption generation with visual attention, In ICML, 2015.

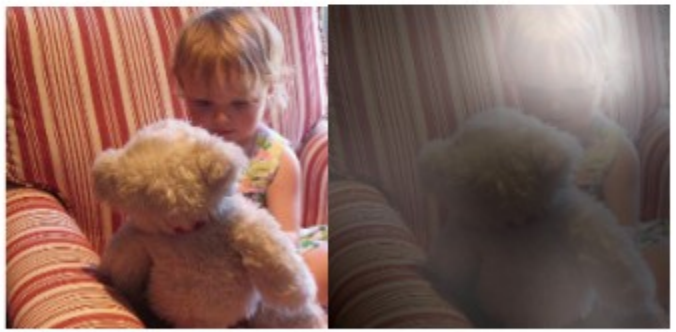
# Show, attend and tell

*Figure 3.* Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

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.



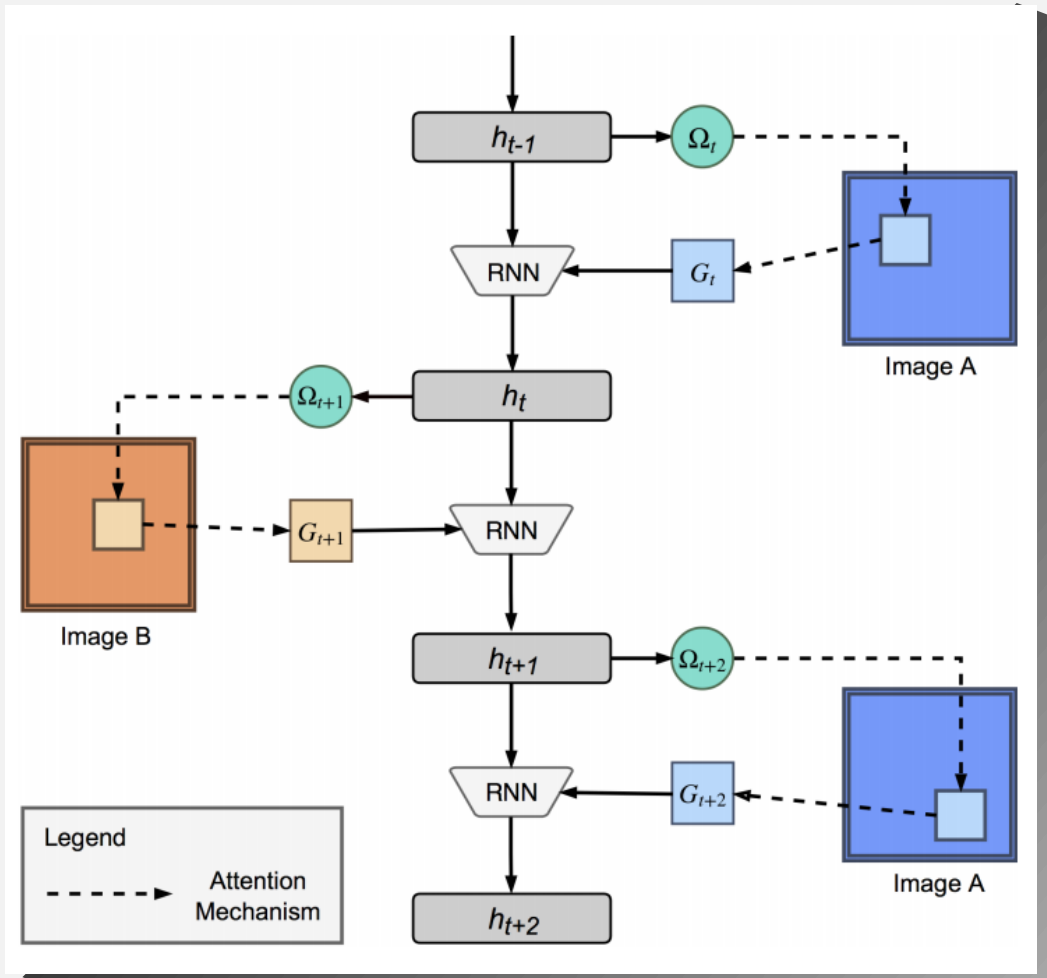
A little girl sitting on a bed with a teddy bear.

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

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

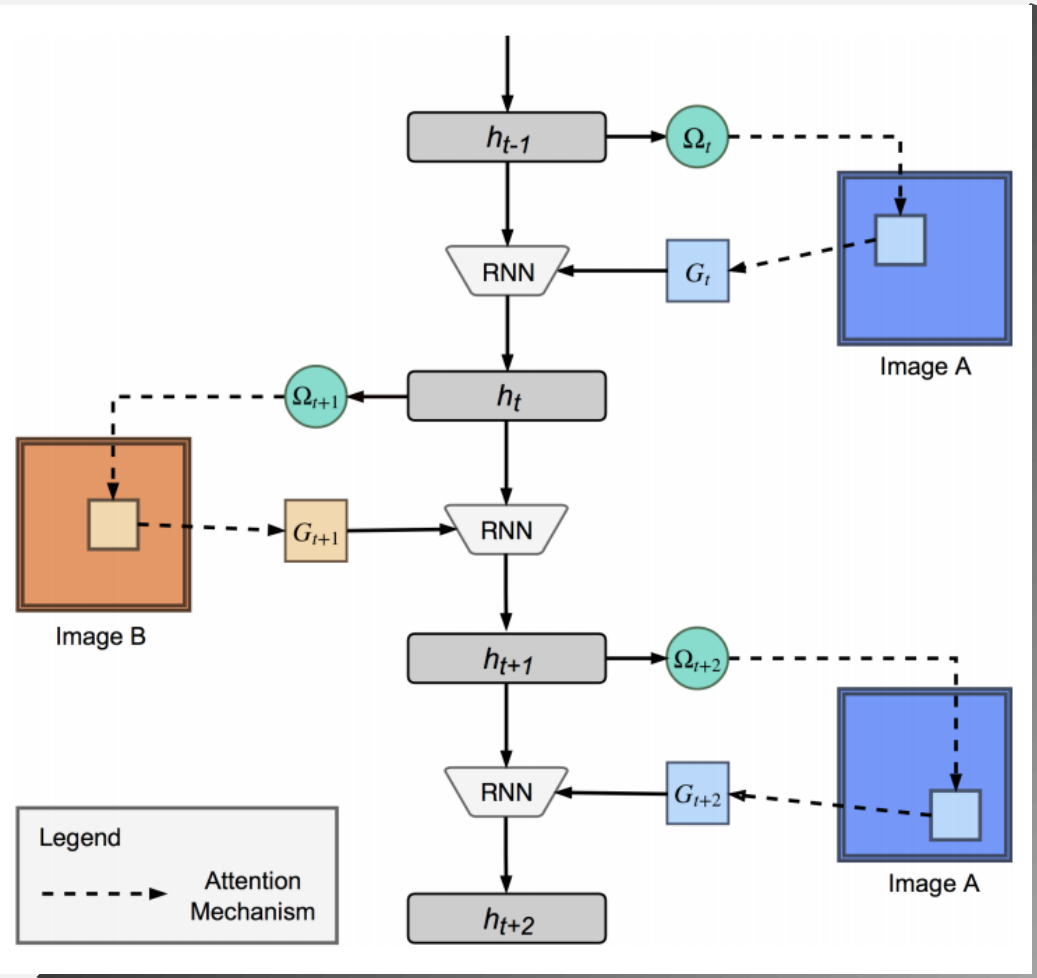
Xu, Kelvin, et al. Show, attend and tell: Neural image caption generation with visual attention, In ICML, 2015.

# Attentive Recurrent Comparators




Shyam, Pranav, S. Gupta, and A. Dukkipati. Attentive Recurrent Comparators, In CVPR 2017.

# Attentive Recurrent Comparators





$I_t \leftarrow x_a$ if t % 2 is 0 else $x_b$

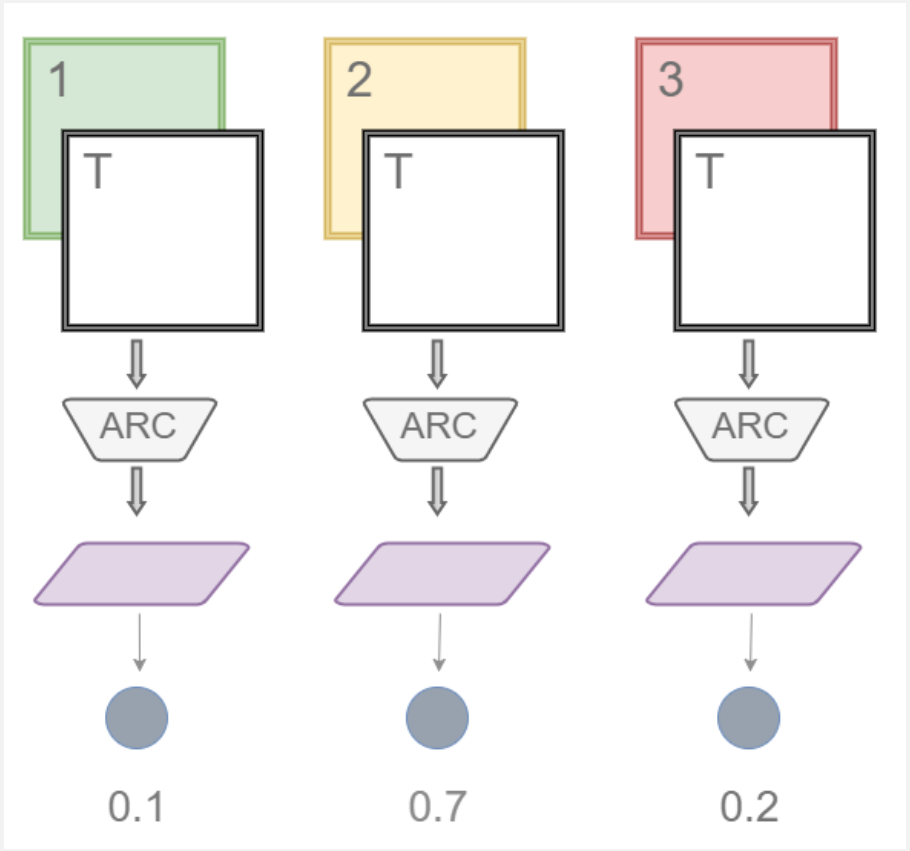
$G_t \leftarrow atten\,d(I_t, \Omega_t)$ where $\Omega_t = W_g h_{t-1}$

$h_t \leftarrow RNN(G_t, h_{t-1})$

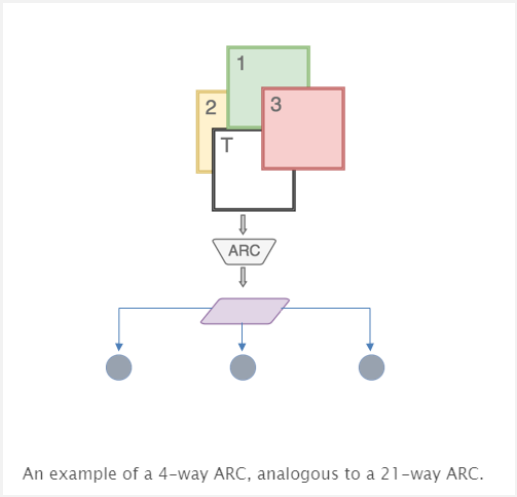
Shyam, Pranav, S. Gupta, and A. Dukkipati, Attentive Recurrent Comparators, In CVPR 2017.

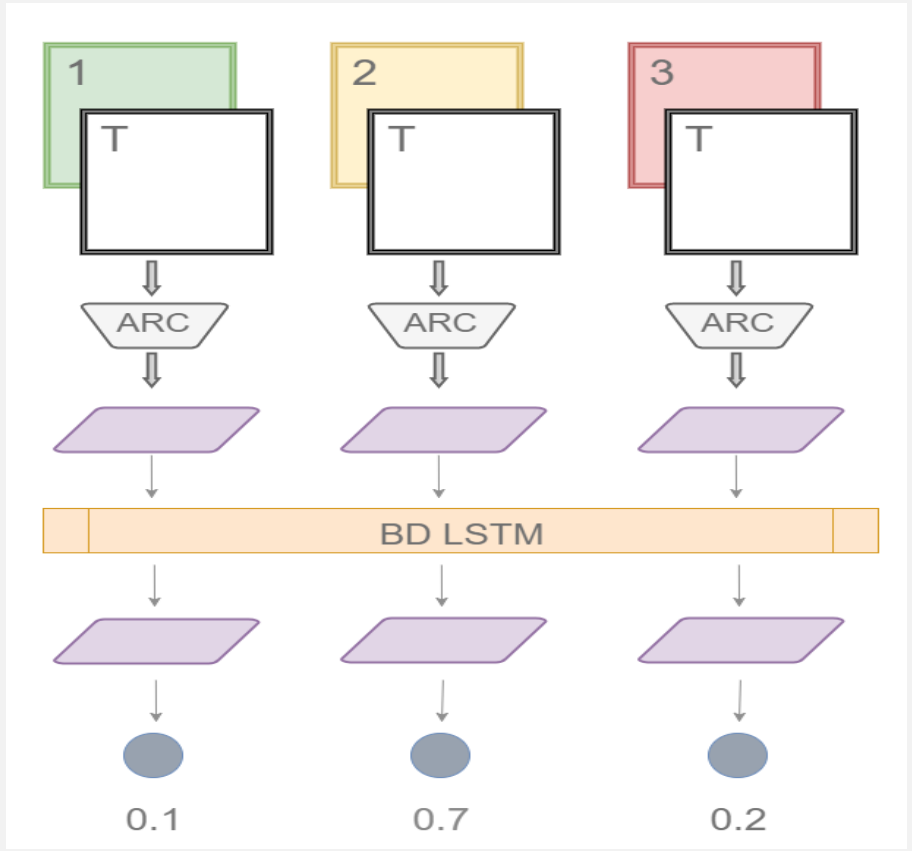
# Attentive Recurrent Comparators

20Way one-shot




Naive ARC

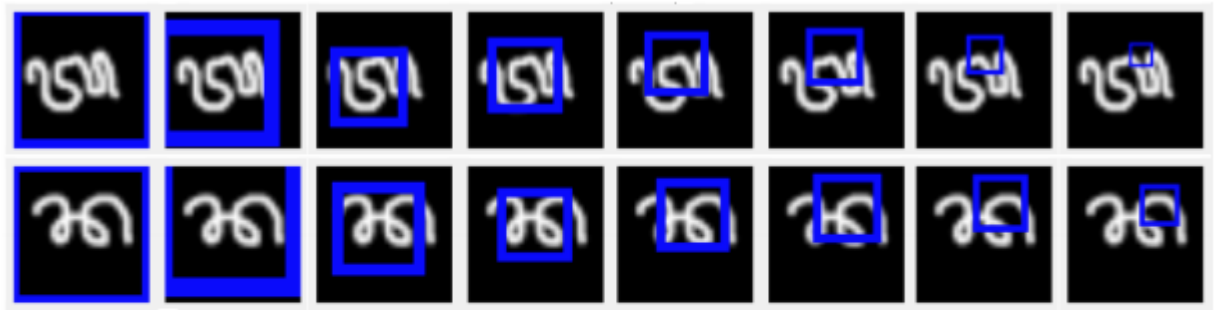



Full Context ARC

Shyam, Pranav, S. Gupta, and A. Dukkipati, Attentive Recurrent Comparators, In CVPR 2017.

# Attentive Recurrent Comparators



(a) It can be seen that the two characters look very similar in their stroke pattern and differ only in their looping structure. ARC has learnt to focus on these crucial aspects.

*Table 1.* Glimpses per image versus classification accuracy of ARC. Out of the 50 alphabets provided in the Omniglot dataset, 30 were used for training and validation and the last 20 for testing

| GLIMPSES | ACCURACY (TEST SET) |
|---|---|
| 1 | 58.2% |
| 2 | 65.0% |
| 4 | 80.8% |
| 6 | 89.25% |
| **8** | **92.08%** |

Shyam, Pranav, S. Gupta, and A. Dukkipati, Attentive Recurrent Comparators, In CVPR 2017.

# Attentive Recurrent Comparators

*Table 5.* One-shot classification accuracies of various methods and our ARC models on Omniglot dataset - Within Alphabets

| MODEL | ACCURACY |
|---|---|
| KNN | 21.7% |
| SIAMESE NETWORK | 58.3% |
| DEEP SIAMESE NETWORK (KOCH ET AL.) | 92.0% |
| HUMANS | 95.5% |
| BPL | 96.7% |
| NAIVE ARC | 91.75% |
| **NAIVE CONVARC** | **97.75%** |
| **FULL CONTEXT CONVARC** | **98.5%** |

*Table 6.* 5 way one-shot Classification accuracies of various methods and our ARC models - miniImageNet

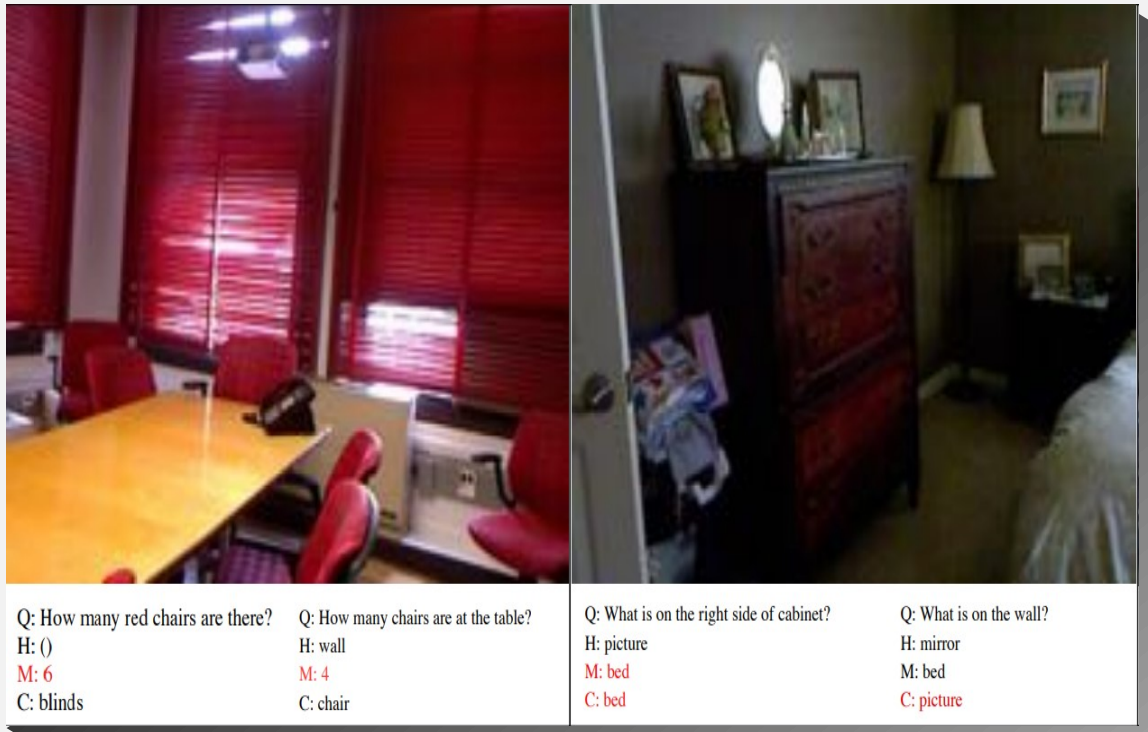
| MODEL | ACCURACY |
|---|---|
| RAW PIXELS W/ COSINE SIMILARITY | 23.0% |
| BASELINE CLASSIFIER | 38.4% |
| MATCHING NETWORKS | 46.6% |
| **NAIVE CONVARC** | **49.14%** |

Shyam, Pranav, S. Gupta, and A. Dukkipati, Attentive Recurrent Comparators, In CVPR 2017.

# Visual Question Answering

Image is from Kafle, K, and Christopher K. Visual question answering: Datasets, algorithms, and future challenges. Computer Vision and Image Understanding, 2017

# DAQUAR




**DAQUAR**

The DAtaset for QUestion Answering on Real-world images

12,000 question-answer pairs on RGBD images

Based on NYU-Depth V2

Kafle, K., & Kanan, C, Visual question answering: Datasets, algorithms, and future challenges. Computer Vision and Image Understanding, 2017.

# COCO-QA


COCO-QA: What does an intersection show on one side and two double-decker buses and a third vehicle,?
Ground Truth: Building

## COCO-QA

Exploring Models and Data for Image Question Answering

- Automatically generated from image captions.
- 123287 images
- 78736 train questions
- 38948 test questions
- 4 types of questions: object, number, color, location
- Answers are all one-word.

Image is from http://www.cs.toronto.edu/~mren/imageqa/data/cocoqa/

# VQA

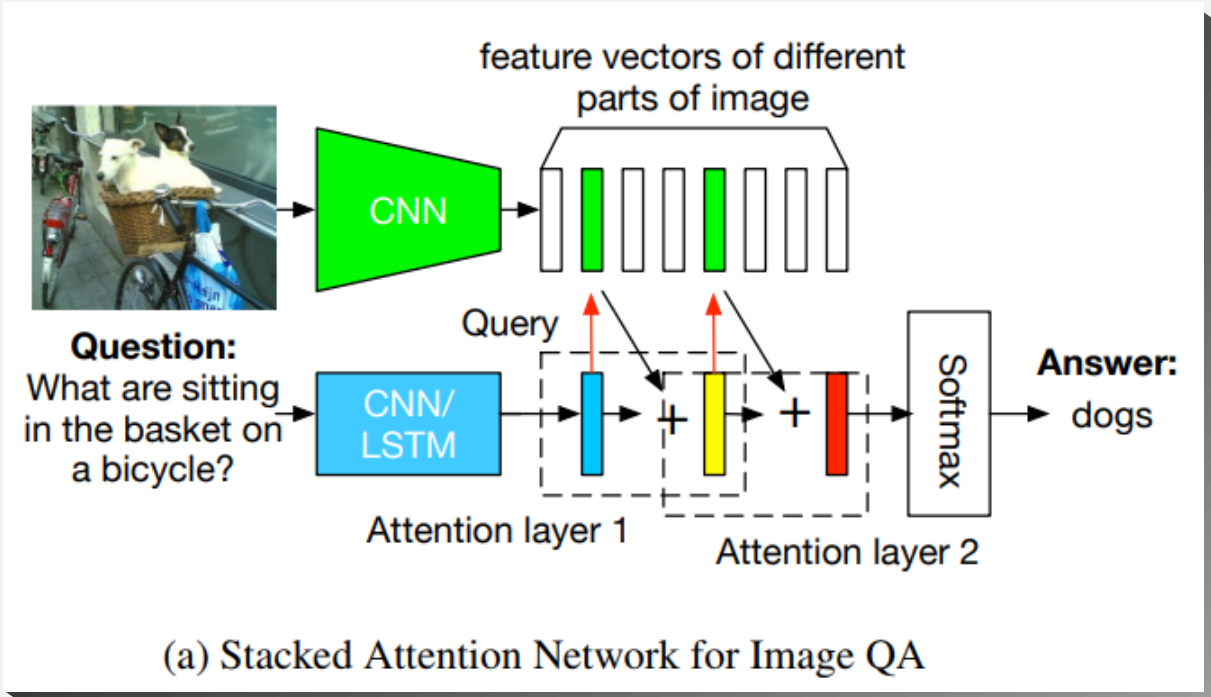

## VQA

VQA is a new dataset containing open-ended questions about images. These questions require an understanding of vision language and commonsense knowledge to answer.

- 265,016 images (COCO and abstract scenes)
- At least 3 questions (5.4 questions on average) per image
- 10 ground truth answers per question
- 3 plausible (but likely incorrect) answers per question
- Automatic evaluation metric

Image is from http://www.visualqa.org/

# Stacked Attention Networks for Image Question Answering




(a) Stacked Attention Network for Image QA

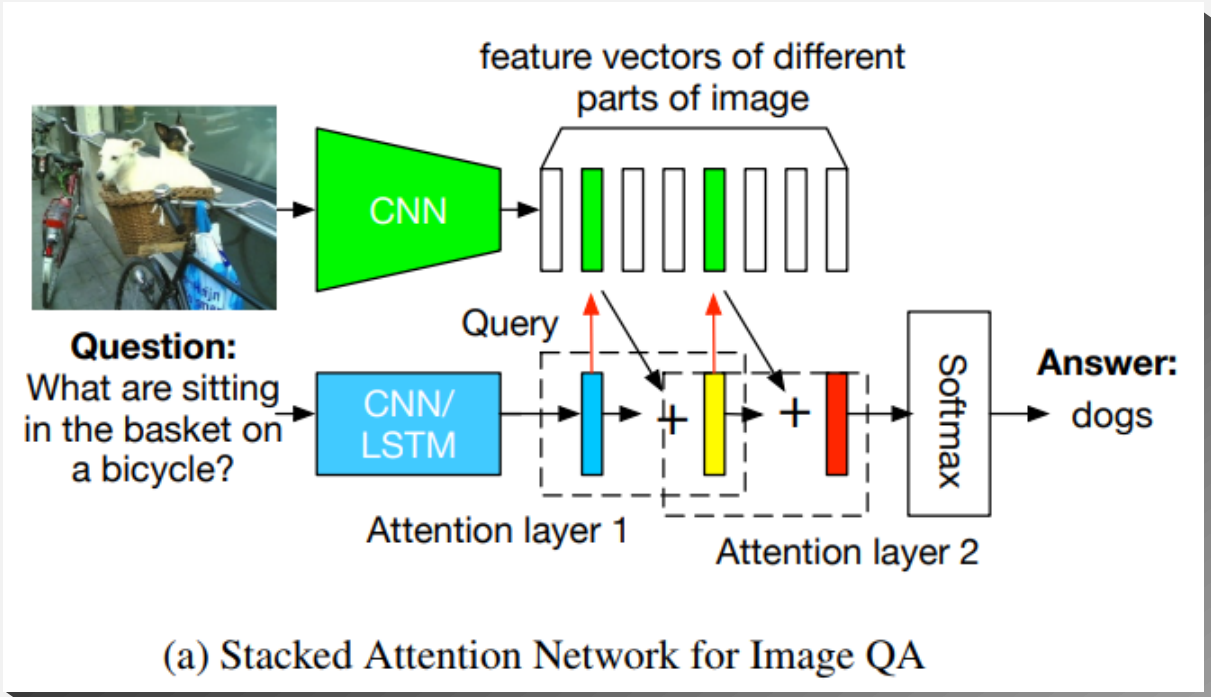
**SANs**

This paper presents stacked attention networks (SANs) that learn to answer natural language questions from images.

- Image model
- Question model
- Stacked attention networks

Yang, Z, et al. Stacked attention networks for image question answering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

# Image model




(a) Stacked Attention Network for Image QA

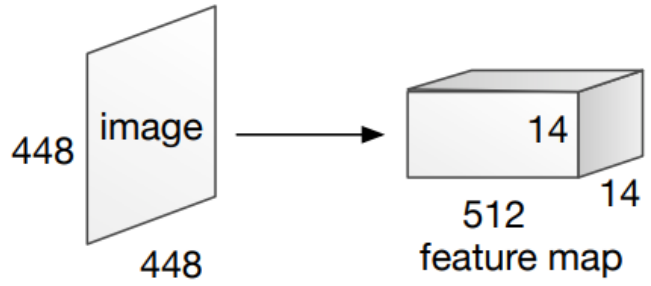


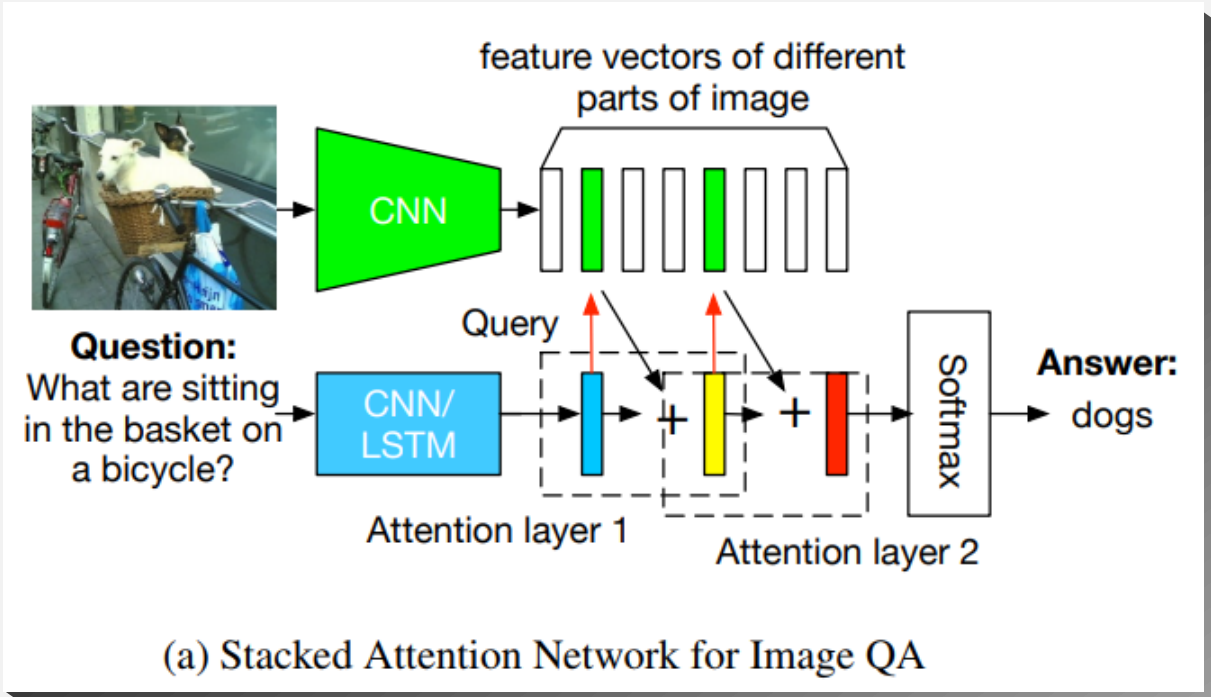

Figure 2: CNN based image model

$$f_I = \text{CNN}_{vgg}(I).$$

$$v_I = \tanh(W_I f_I + b_I),$$

Yang, Z, et al. Stacked attention networks for image question answering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

# Question model




(a) Stacked Attention Network for Image QA

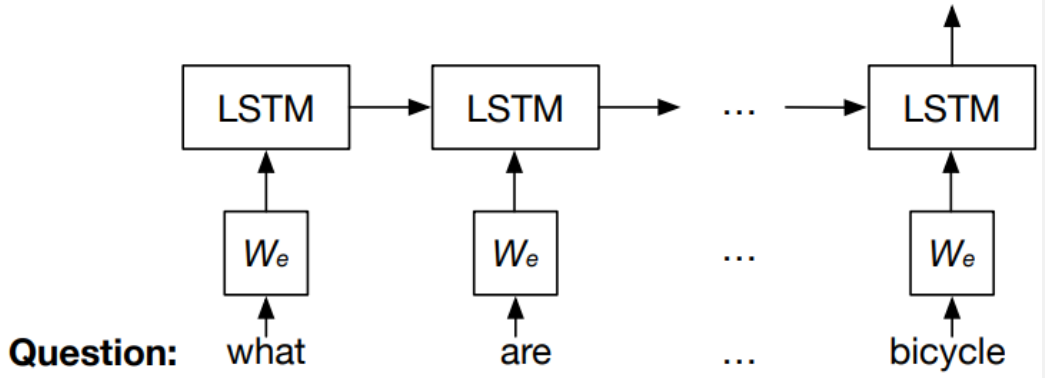
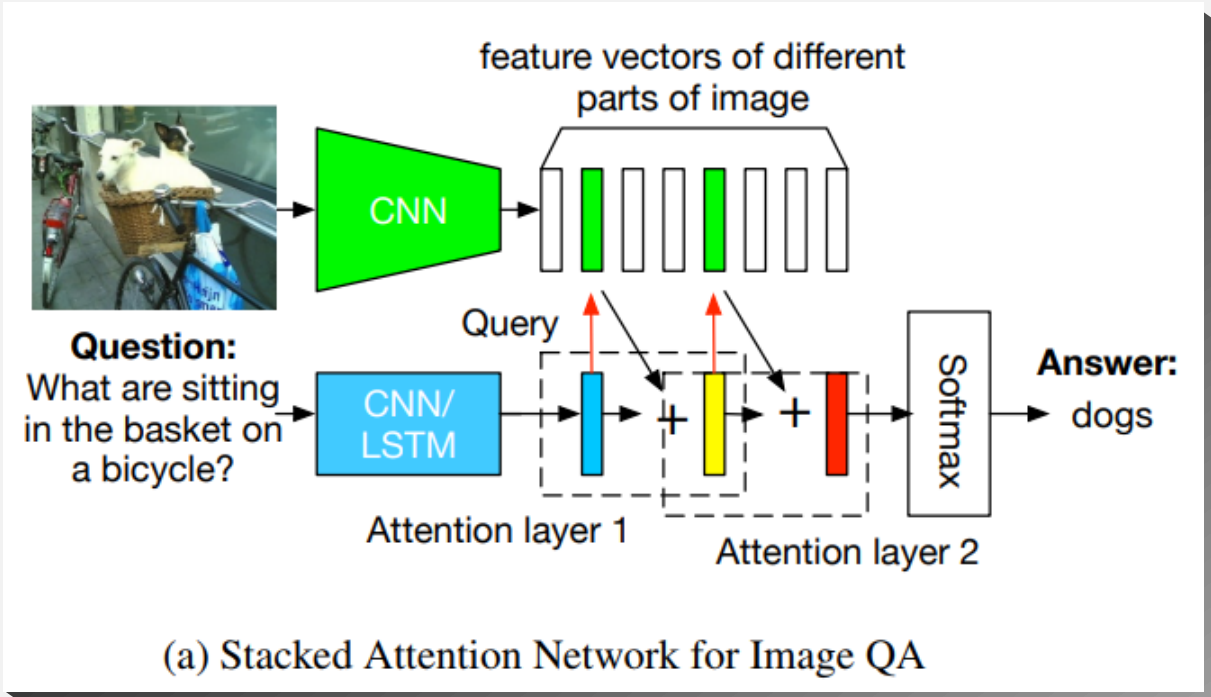



Figure 3: LSTM based question model

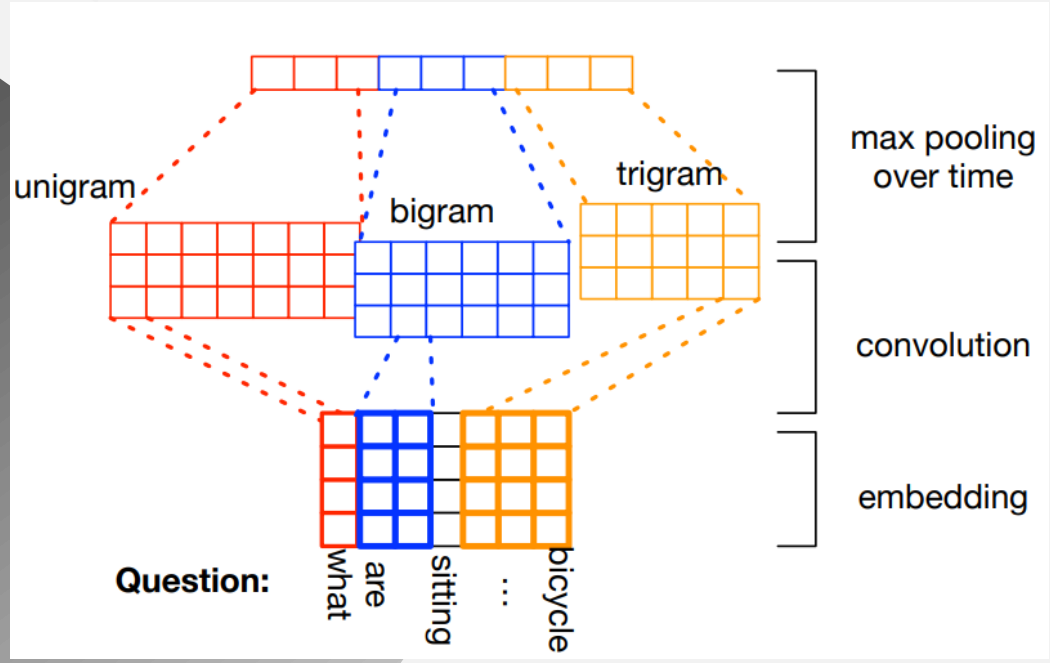
$$x_t = W_e q_t, t \in \{1, 2, ...T\},$$
$$h_t = \text{LSTM}(x_t), t \in \{1, 2, ...T\}.$$

Yang, Z, et al. Stacked attention networks for image question answering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

# Question model

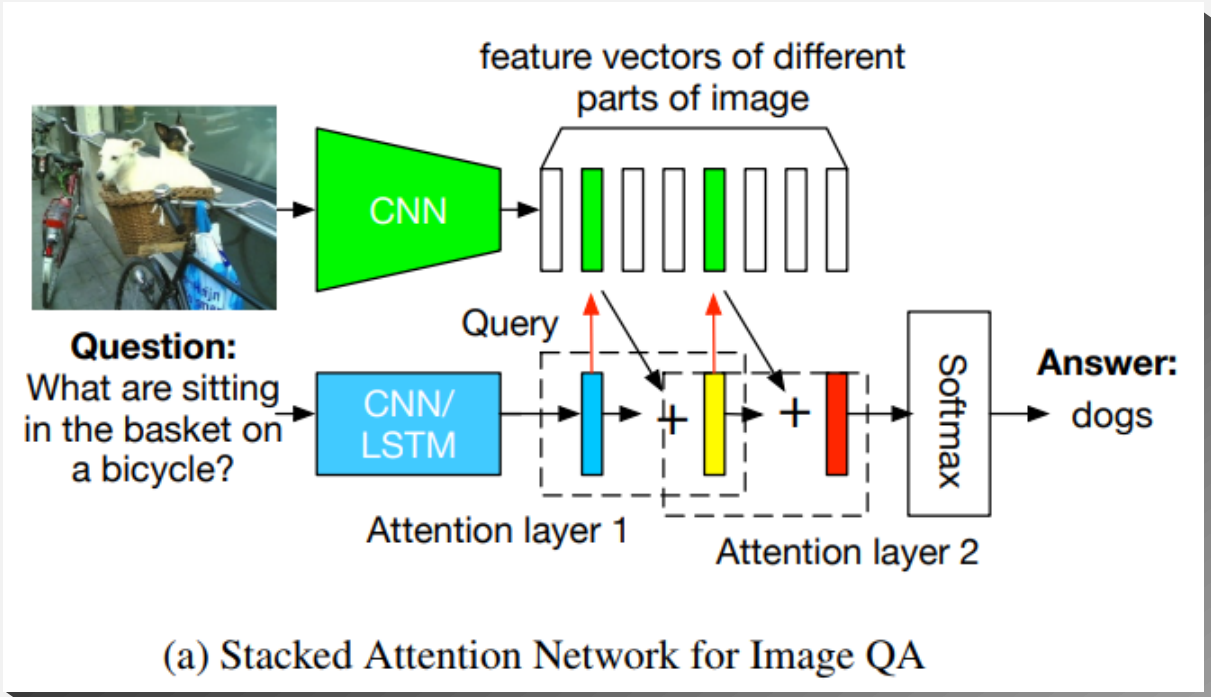



(a) Stacked Attention Network for Image QA




Yang, Z, et al. Stacked attention networks for image question answering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

# Stacked Attention Networks




(a) Stacked Attention Network for Image QA

$$h_A = tanh(W_{I,A} v_I \oplus (W_{Q,A} v_Q + b_A))$$

$$p_I = \text{softmax}(W_P h_A + b_P)$$

$$\tilde{v}_I = \sum_i p_i v_i$$

$$\mu = \tilde{v}_I + v_Q$$

$$h_A^k = tanh(W_{I,A}^k v_I \oplus (W_{Q,A}^k \mu^{k-1} + b_A^k))$$

$$p_I^k = \text{softmax}(W_P^k h_A^k + b_P^k)$$

$$\tilde{v}_I^k = \sum_i p_i^k v_i$$

$$\mu^k = \tilde{v}_I^k + \mu^{k-1}$$

$$p_{\text{ans}} = \text{softmax}(W_u u^K + b_u)$$

Yang, Z, et al. Stacked attention networks for image question answering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

# Stacked Attention Networks

| Methods | Accuracy | WUPS0.9 | WUPS0.0 |
|---|---|---|---|
| **VSE:** [21] | | | |
| GUESS | 6.7 | 17.4 | 73.4 |
| BOW | 37.5 | 48.5 | 82.8 |
| LSTM | 36.8 | 47.6 | 82.3 |
| IMG | 43.0 | 58.6 | 85.9 |
| IMG+BOW | 55.9 | 66.8 | 89.0 |
| VIS+LSTM | 53.3 | 63.9 | 88.3 |
| 2-VIS+BLSTM | 55.1 | 65.3 | 88.6 |
| **CNN:** [17] | | | |
| IMG-CNN | 55.0 | 65.4 | 88.6 |
| CNN | 32.7 | 44.3 | 80.9 |
| **Ours:** | | | |
| SAN(1, LSTM) | 59.6 | 69.6 | 90.1 |
| SAN(1, CNN) | 60.7 | 70.6 | 90.5 |
| SAN(2, LSTM) | 61.0 | 71.0 | 90.7 |
| SAN(2, CNN) | **61.6** | **71.6** | **90.9** |

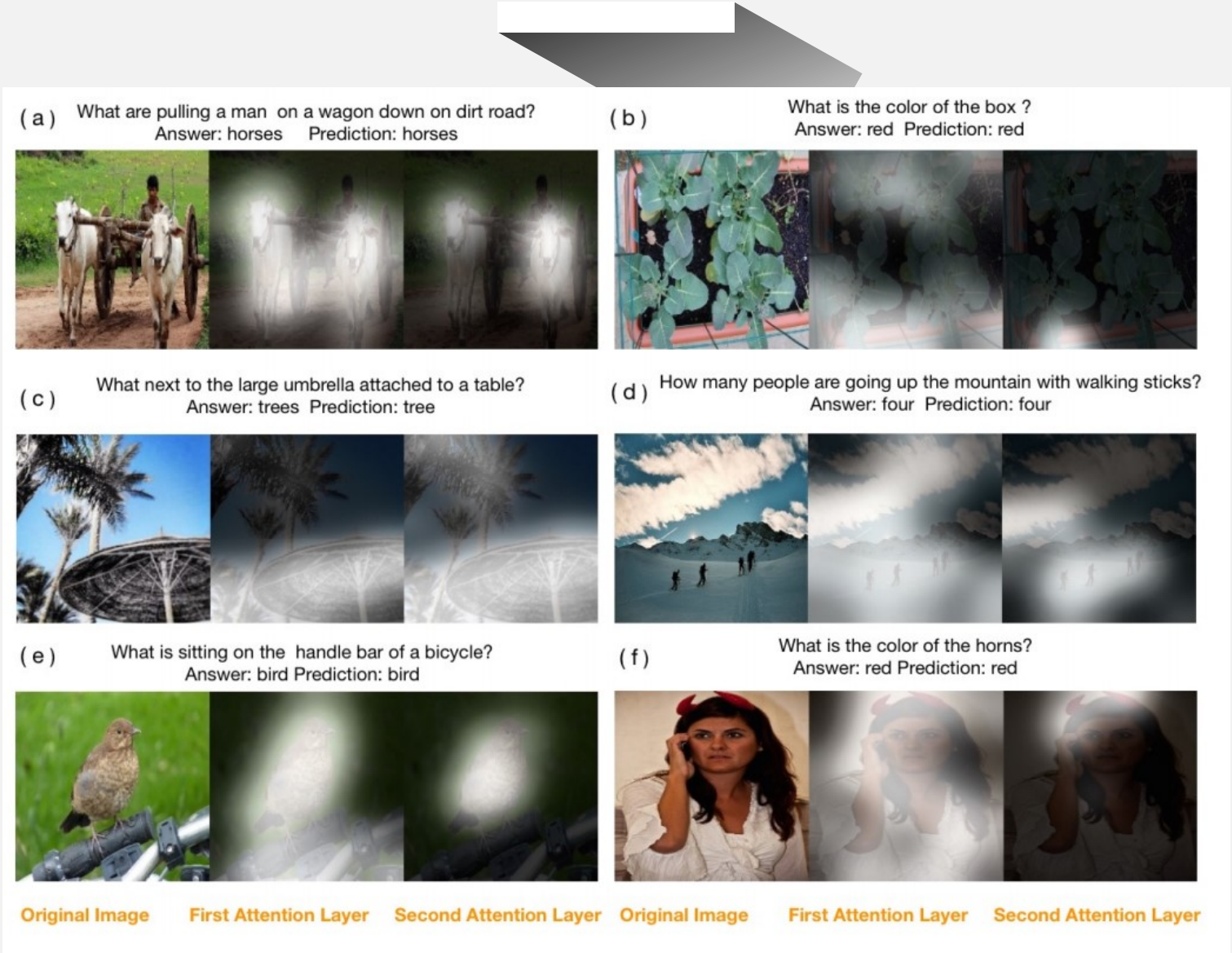
Table 3: COCO-QA results, in percentage

| Methods | All | Yes/No 36% | Number 10% | Other 54% |
|---|---|---|---|---|
| SAN(1, LSTM) | 56.6 | 78.1 | 41.6 | 44.8 |
| SAN(1, CNN) | 56.9 | 78.8 | 42.0 | 45.0 |
| SAN(2, LSTM) | 57.3 | 78.3 | **42.2** | 45.9 |
| SAN(2, CNN) | **57.6** | 78.6 | 41.8 | **46.4** |

Table 6: VQA results on our partition, in percentage

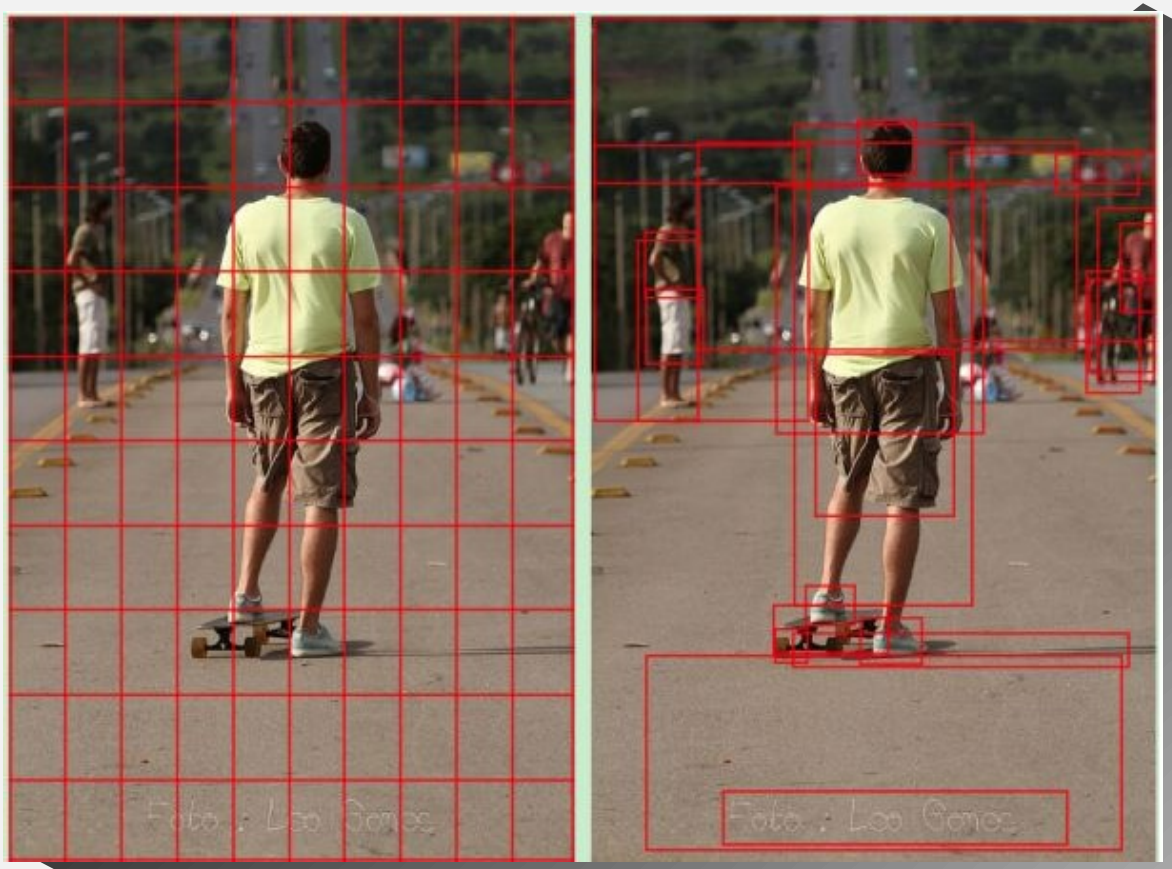
Yang, Z, et al. Stacked attention networks for image question answering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

# Stacked Attention Networks




Yang, Z, et al. Stacked attention networks for image question answering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

# Bottom-Up and Top-Down Attention



**Bottom-Up and Top-Down Attention**

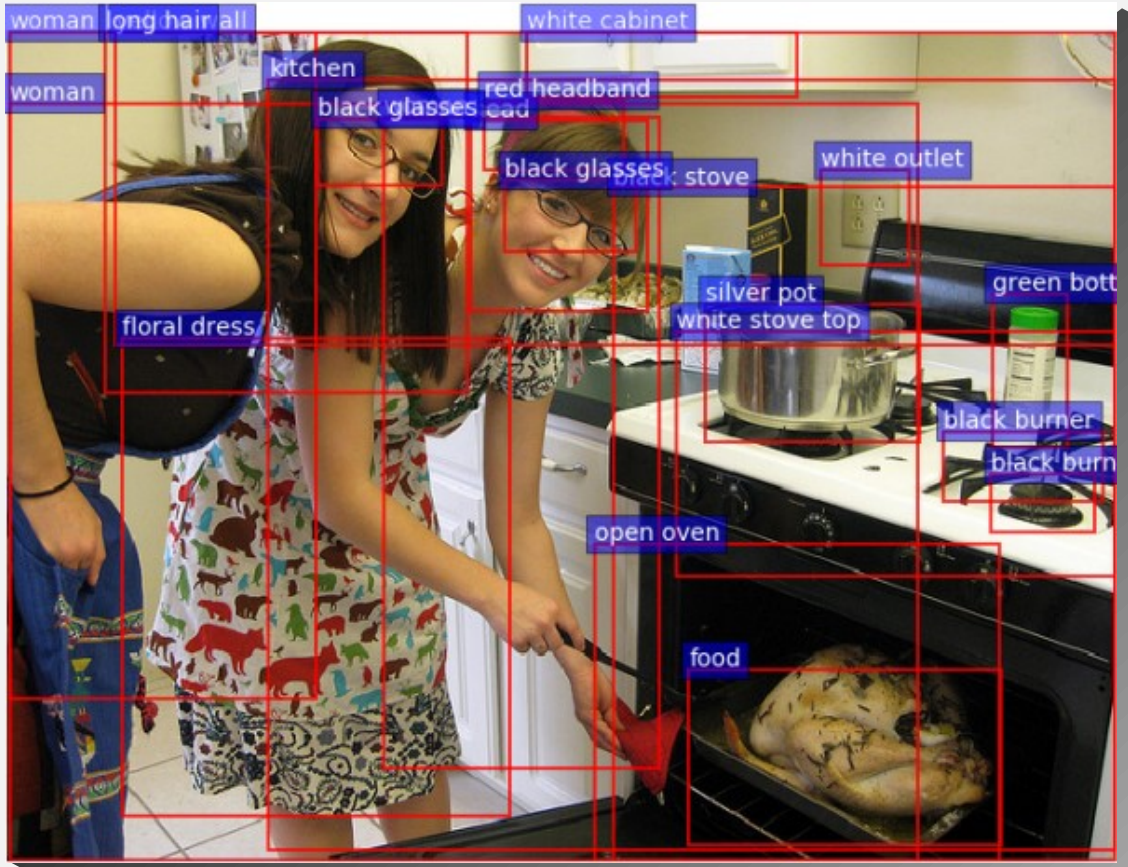
Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering

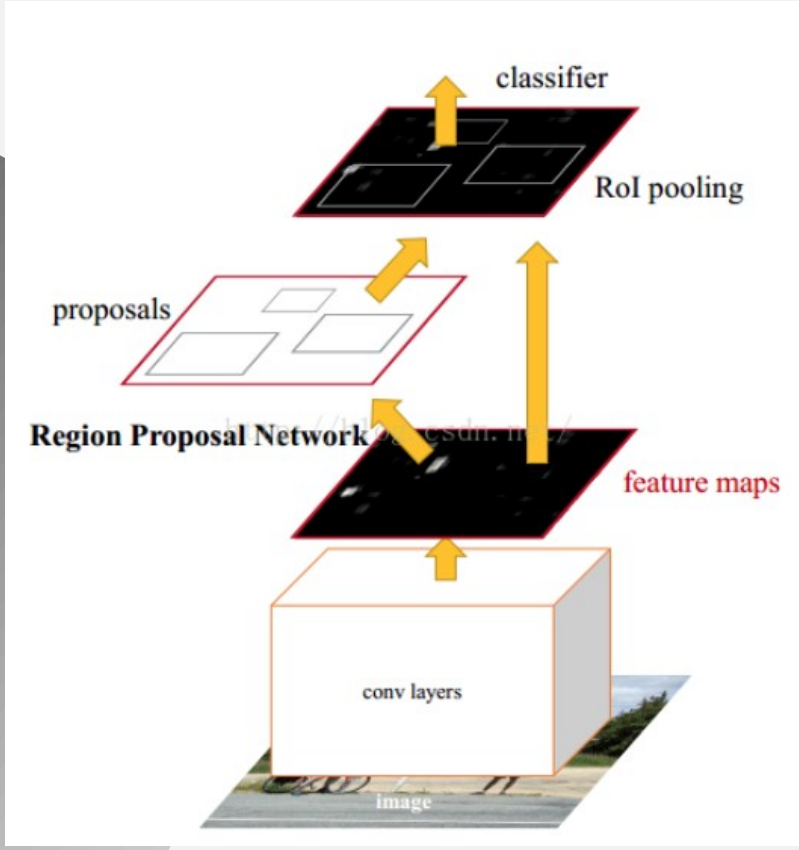
**Bottom-Up**

**Top-Down**

Anderson, P., He, X. et al., Bottom-up and top-down attention for image captioning and visual question answering, In CVPR 2018.

# Bottom-Up attention model




Anderson, P., He, X. et al., Bottom-up and top-down attention for image captioning and visual question answering, In CVPR 2018.

# Bottom-Up attention model

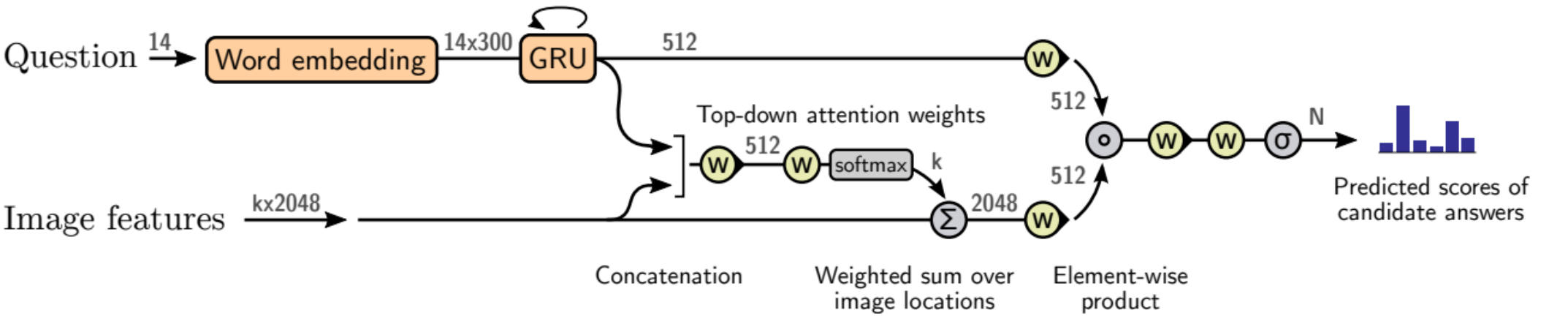



Figure 4. Overview of the proposed VQA model. A deep neural network implements a joint embedding of the question and image features $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_k\}$. These features can be defined as the spatial output of a CNN, or following our approach, generated using bottom-up attention. Output is generated by a multi-label classifier operating over a fixed set of candidate answers. Gray numbers indicate the dimensions of the vector representations between layers. Yellow elements use learned parameters.

Anderson, P., He, X. et al., Bottom-up and top-down attention for image captioning and visual question answering, In CVPR 2018.

# Bottom-Up attention model

| | Yes/No | Number | Other | Overall |
|---|---|---|---|---|
| Prior [13] | 61.20 | 0.36 | 1.17 | 25.98 |
| Language-only [13] | 67.01 | 31.55 | 27.37 | 44.26 |
| d-LSTM+n-I [28, 13] | 73.46 | 35.18 | 41.83 | 54.22 |
| MCB [12, 13] | 78.82 | 38.28 | 53.36 | 62.27 |
| UPMC-LIP6 | 82.07 | 41.06 | 57.12 | 65.71 |
| Athena | 82.50 | 44.19 | 59.97 | 67.59 |
| HDU-USYD-UNCC | 84.50 | 45.39 | 59.01 | 68.09 |
| Ours: Up-Down | **86.60** | **48.64** | **61.15** | **70.34** |

Table 5. VQA v2.0 test-standard server accuracy, ranking our submission against recent published and unpublished work for each question type. Our approach, an ensemble of 30 models trained with different random seeds, outperforms all other leaderboard entries.

Anderson, P., He, X. et al., Bottom-up and top-down attention for image captioning and visual question answering, In CVPR 2018.

# Bottom-Up attention model

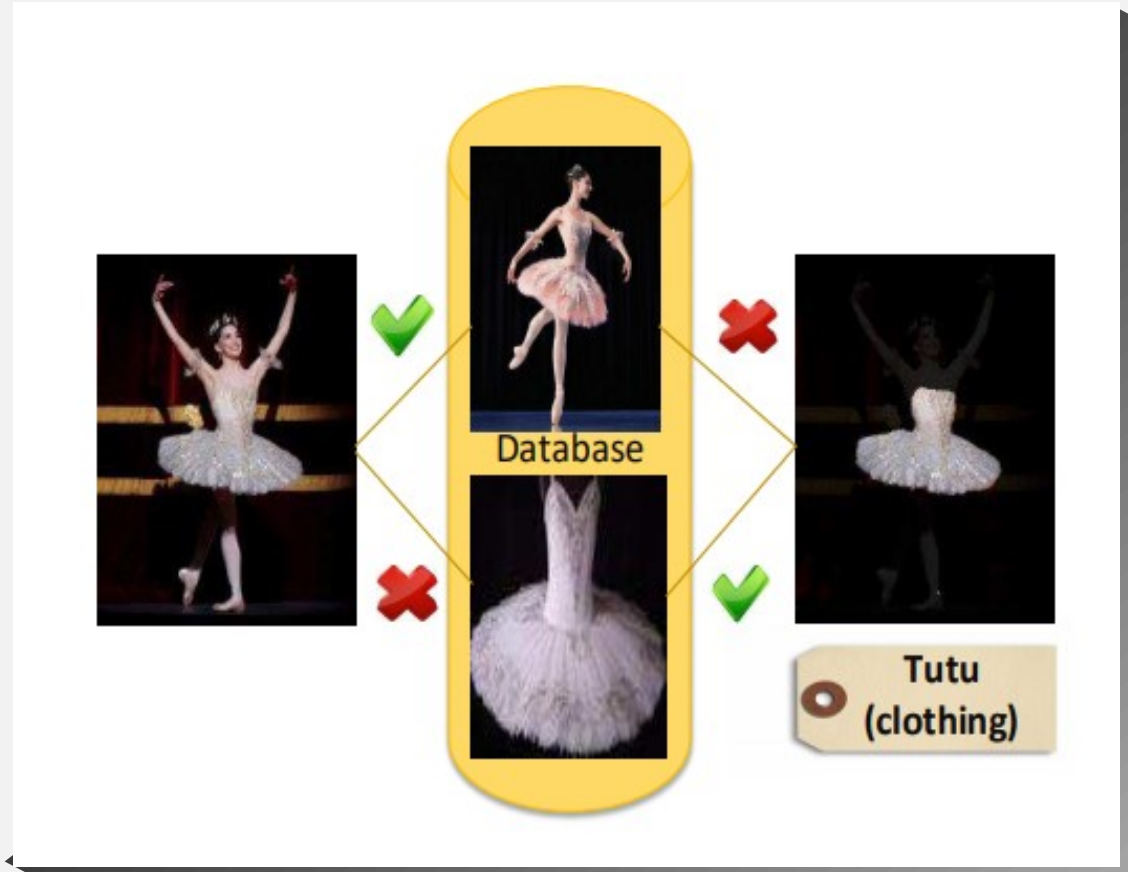

Anderson, P., He, X. et al., Bottom-up and top-down attention for image captioning and visual question answering, In CVPR 2018.

# Multi-attention network for one shot learning

Car=class 1

Motorcycle=class2

# Multi-attention network for one shot learning




**Multi-attention network**

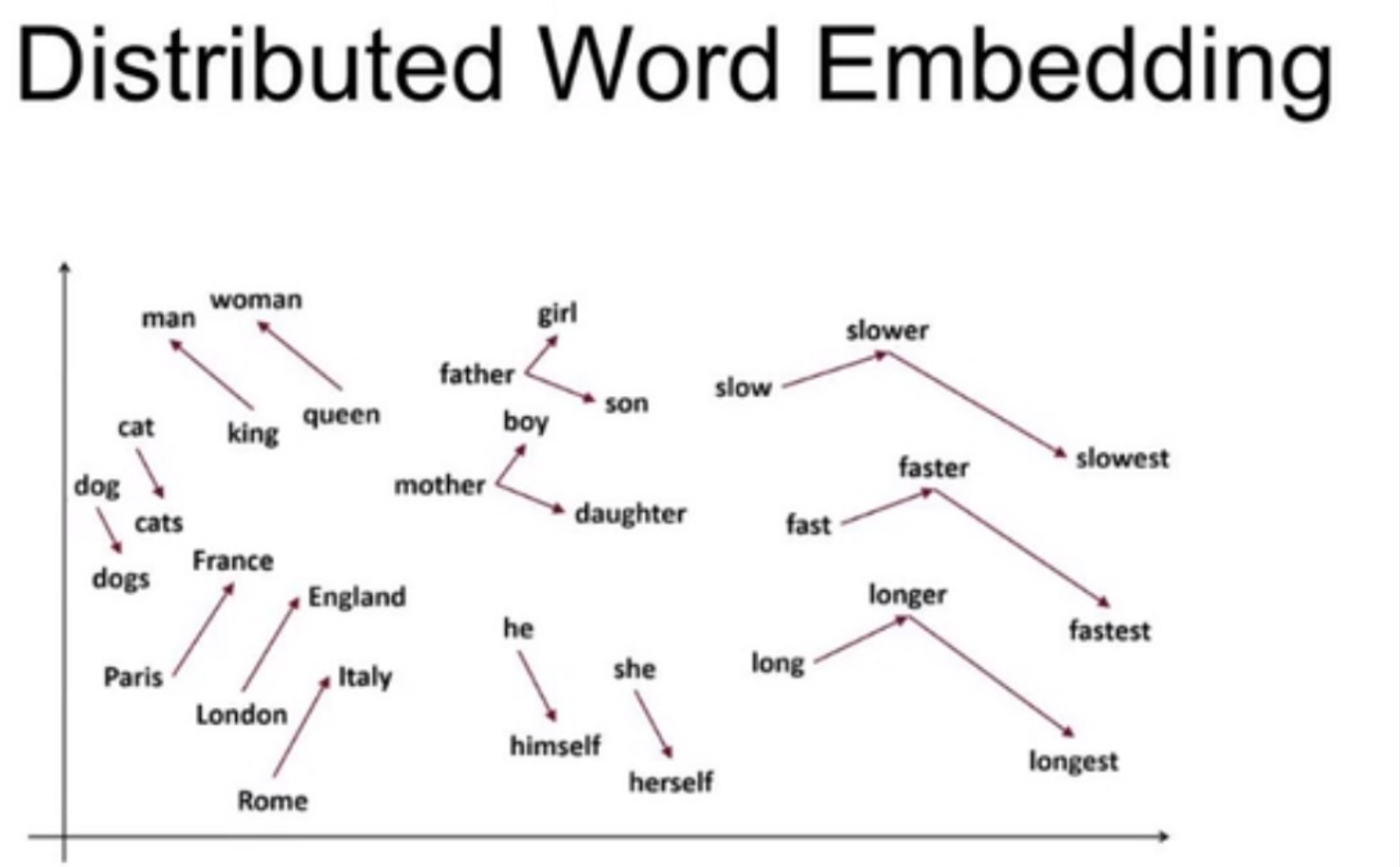
We design a neural network architecture which takes the semantic embedding of the class tag to generate attention maps.

**The use of the class tag**

**Visual attention**

Wang, P., Liu, L., Shen, C., Huang, Z., van den Hengel, A., & Shen, H. T. Multi-attention network for one shot learning. In CVPR, 2017.

# Multi-attention network for one shot learning




Wang, P., Liu, L., Shen, C., Huang, Z., van den Hengel, A., & Shen, H. T. Multi-attention network for one shot learning. In CVPR, 2017.

# Multi-attention network for one shot learning

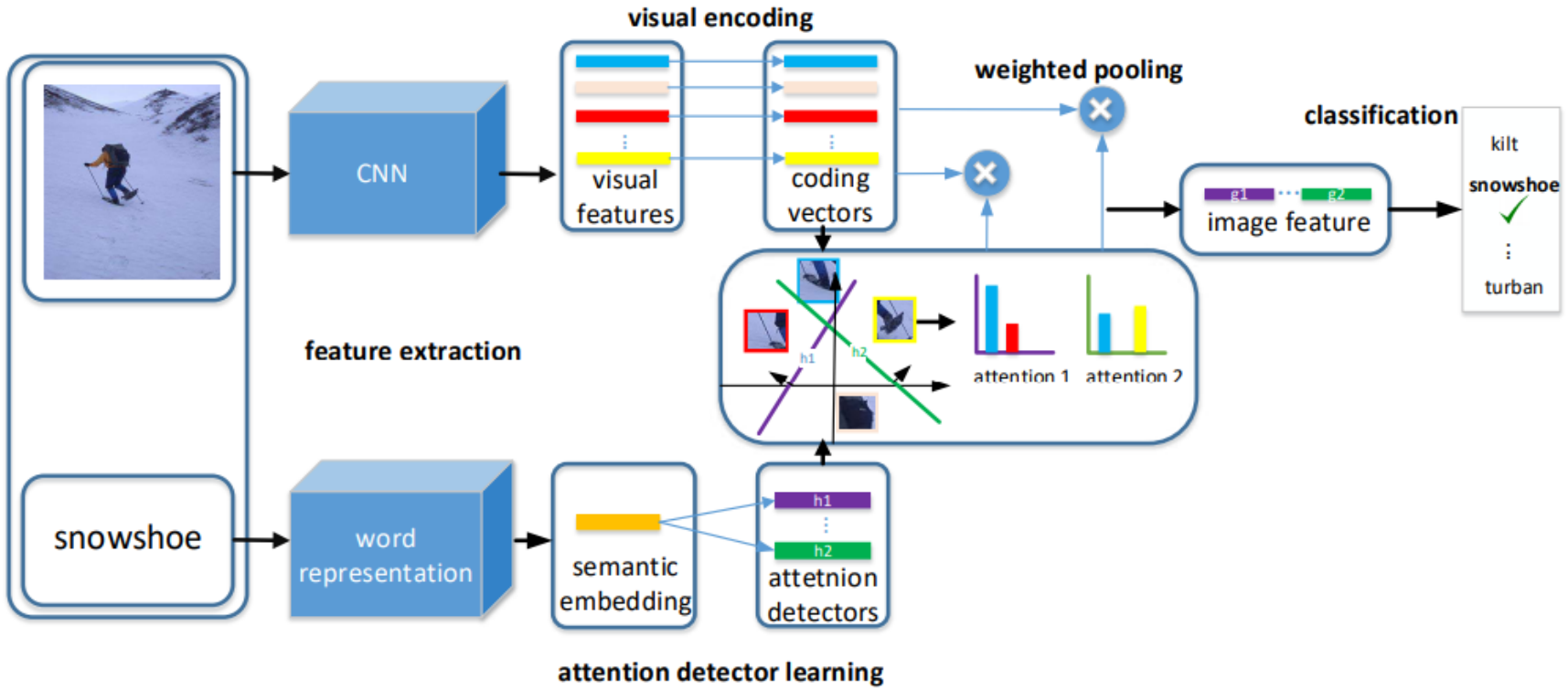



Figure 2. An illustration of the overview architecture of the proposed attention network.

$$\mathbf{v}_i = f(\mathbf{W}_v \mathbf{x}_i + \mathbf{b}_v),$$

$$\mathbf{h} = \mathbf{W}_s \mathbf{c} + \mathbf{b}_s,$$

$$a'_i = \mathbf{h}^\top \mathbf{v}_i.$$

$$a_i = \frac{b(a'_i)}{\sum_i b(a'_i)},$$

Wang, P., Liu, L., Shen, C., Huang, Z., van den Hengel, A., & Shen, H. T. Multi-attention network for one shot learning. In CVPR, 2017.

# Multi-attention network for one shot learning

Table 1. Comparison of the attention network to alternative solutions on the Animal Dataset.

| | | |
|---|---|---|
| close-world | Global FC | 68.9% |
| | SE | 67.7% |
| | Zero shot learning | 55.2% |
| | Attention (attribute) | **72.4%** |
| | Attention (word vector) | **74.0%** |
| | Multi-Attention (2) | **82.4%** |
| | Multi-Attention (5) | **77.7%** |
| open-world | Global FC | 42.8% |
| | SE | 42.1% |
| | Zero shot learning | 15.3% |
| | Attention (attribute) | **47.4%** |
| | Attention (word vector) | **49.0%** |
| | Multi-Attention (2) | **55.7%** |
| | Multi-Attention (5) | **56.8%** |

Table 2. Comparison of the attention network to alternative solutions on the Artifact Dataset.

| | | |
|---|---|---|
| close-world | SE (256D) | 27.8% |
| | SE (512D) | 28.2% |
| | SE + Joint Bayesian | 31.5% |
| | Attention | **34.5%** |
| | Attention + Joint Bayesian | **36.8%** |
| | Multi-Attention (2) | **50.5%** |
| open-world | SE (256D) | 11.6% |
| | SE (512D) | 12.2% |
| | SE + Joint Bayesian | 11.4% |
| | Attention | **15.2%** |
| | Multi-Attention (2) | **28.2%** |

Wang, P., Liu, L., Shen, C., Huang, Z., van den Hengel, A., & Shen, H. T. Multi-attention network for one shot learning. In CVPR, 2017.

# Thank you!

http://vision.ouc.edu.cn

Chenchen Qiu