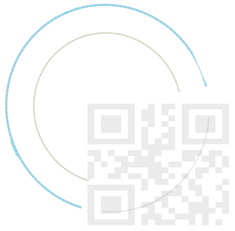
Neural Network

玖强

Outline

- Convolutional Neural Network
- Recurrent Neural Network
- Deep learning toolbox

Convolutional Neural Network



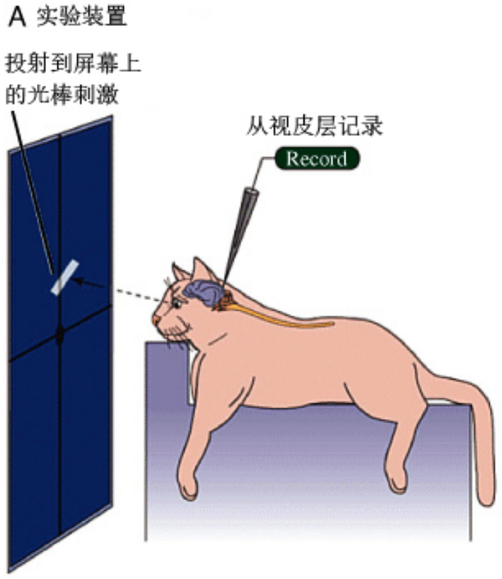
The History of CNN

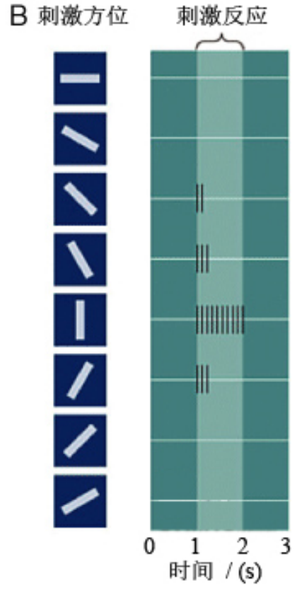
- Lecun并不是第一个提出CNN思想的人
  - 虽然，他1989年发表了《Backpropagation Applied to Handwritten Zip Code》
- Kunihiko Fukishima(福岛邦彦)提出的Neocognitron（新认知机）才是鼻祖.
  - 1980-Fukushima-Neocognitron A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position》
  - Lecun能把Neocogniron的精华提取出来然后加上其1986的BP算法,然后把CNN发扬光大.

The History of CNN

- **Receptive field/感受野（1958-1962）**

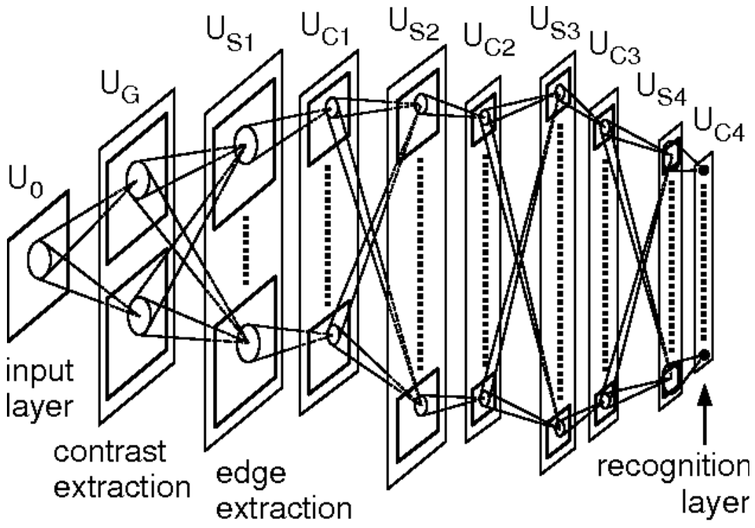
Hubel-Wiesel starting out

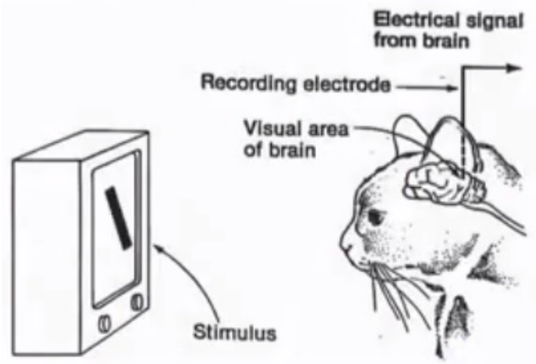




The History of CNN

- **Neocognitron /认知机（1980）**





(Hubel and Wiesel, c. 1965)

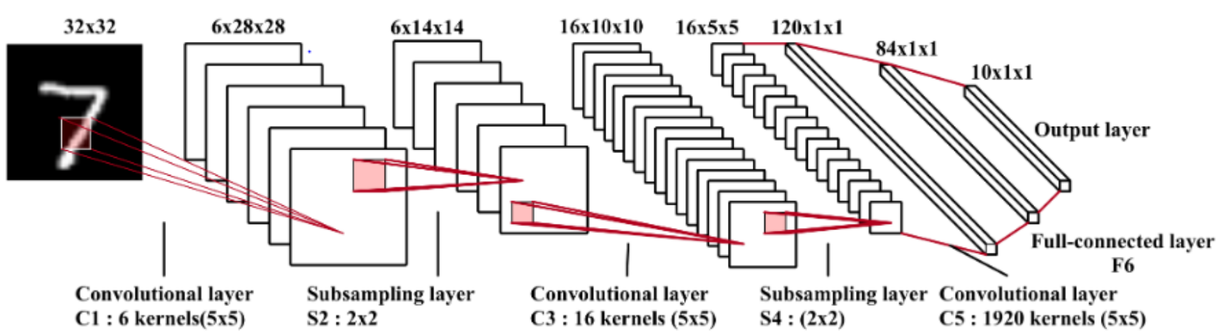
The History of CNN

❑ Lecun's LeNet-5 (1989)

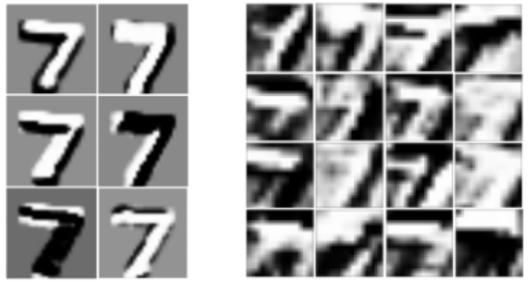
1989，Published the seminal paper establishing the modern framework of CNN

后来在1998，Gradient based learning applied to document recognition改进，这就是著名

卷积、激活、池化和全连接



(a) LeNet-5 network

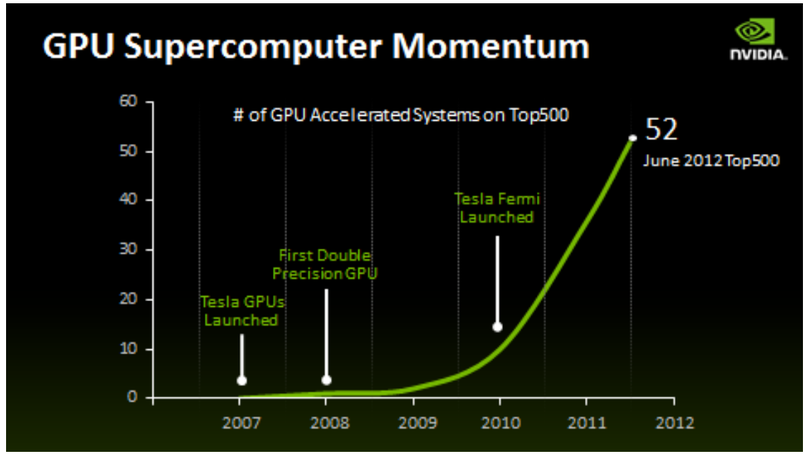


(b) Learned features

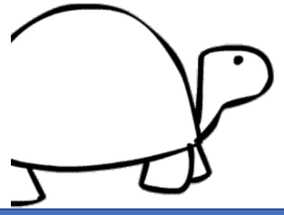
The History of CNN

No Data
No Powerful PC
No GPU
….

No Hope!!!



LeNet-5 (1998)



在1998年以后，深度学习并没有太多的突破

2006
Hinton

2012
Hinton's
Student

加速发展

The History of CNN

- **AlexNet (1989)**
  - **AlexNet是这一拨深度神经网络在视觉识别上大规模应用的开端**

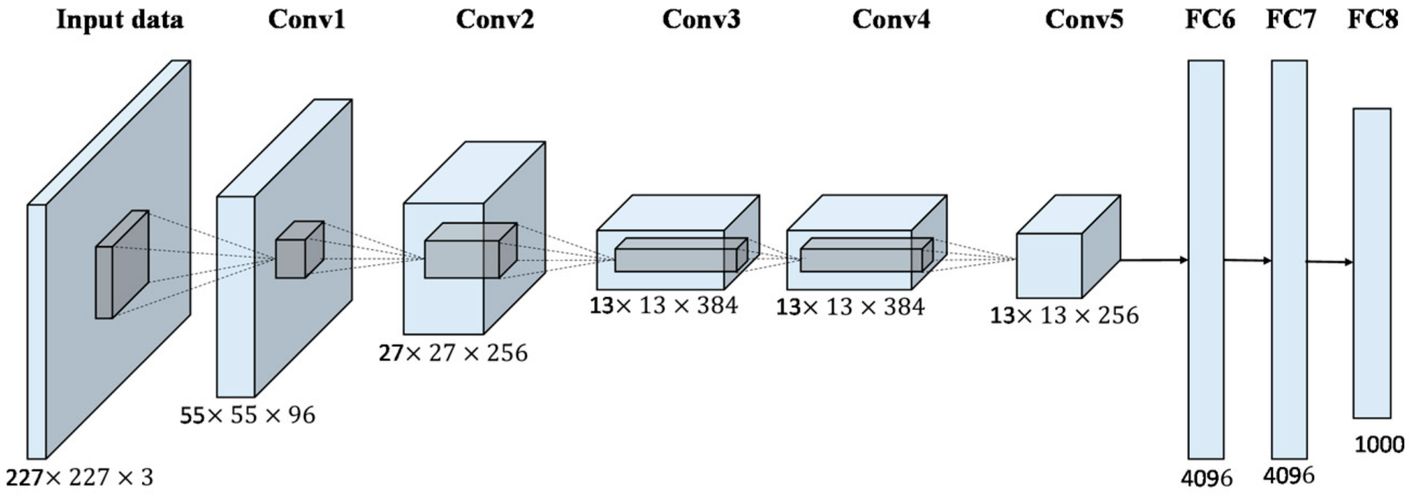
Imagenet classification with deep convolutional neural networks
A Krizhevsky, I Sutskever, GE Hinton - Advances in neural ..., 2012 - papers.nips.cc
Abstract We trained a large, deep convolutional neural network to classify the 1.3 million high-resolution images in the LSVRC-2010 ImageNet training set into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 39.7\% and 18.9\%
☆ 99 Cited by 21201 Related articles Web of Science: 12

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

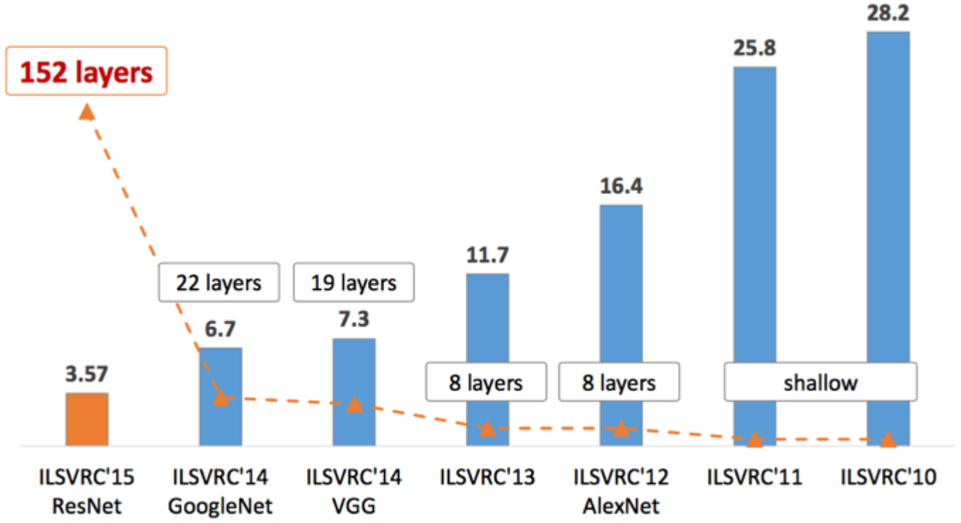
Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

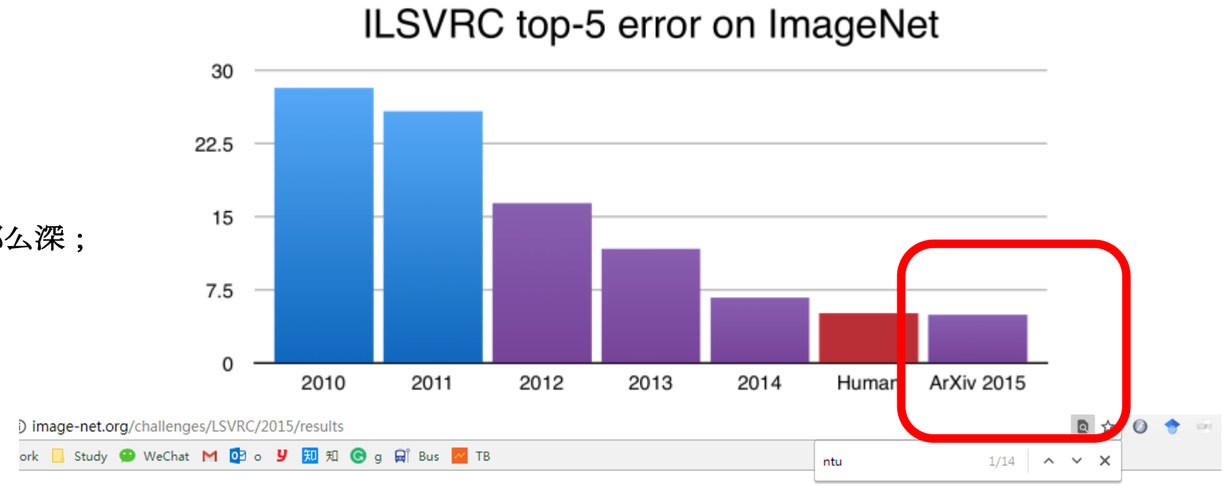


The History of CNN

❑ AlexNet为何这么厉害？

1. 大量的数据，深度学习领域应该感谢李飞飞团队弄出来的**ImageNet;**
2. **GPU**，这种高度并行的计算神器给与我们洪荒之力，没有它，**Alex**也不敢弄那么深；
3. 算法、算法还是算法！更深的网络、数据增强、**ReLU**、**Dropout**等





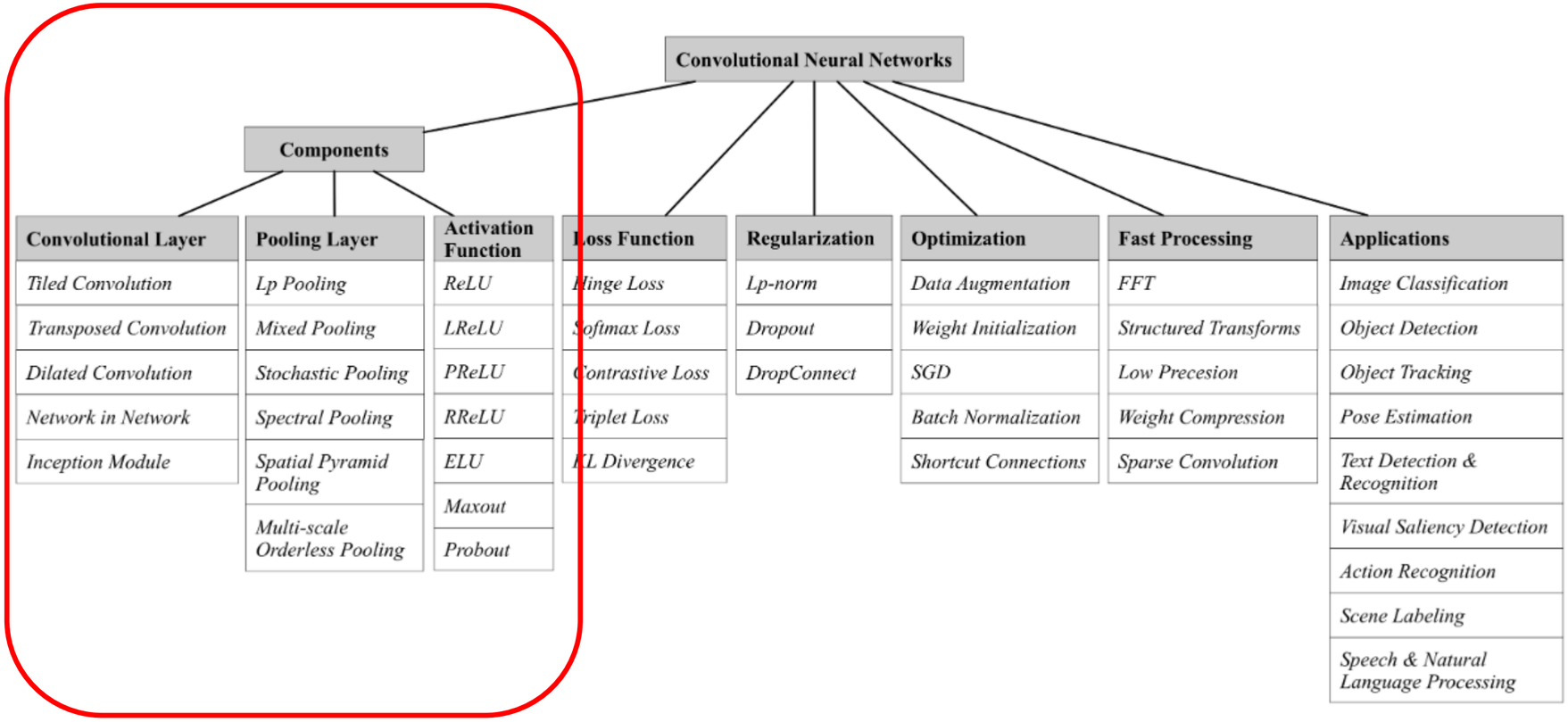
Scene Classification (Scene)[top]

Task 4a: Scene classification with provided training data

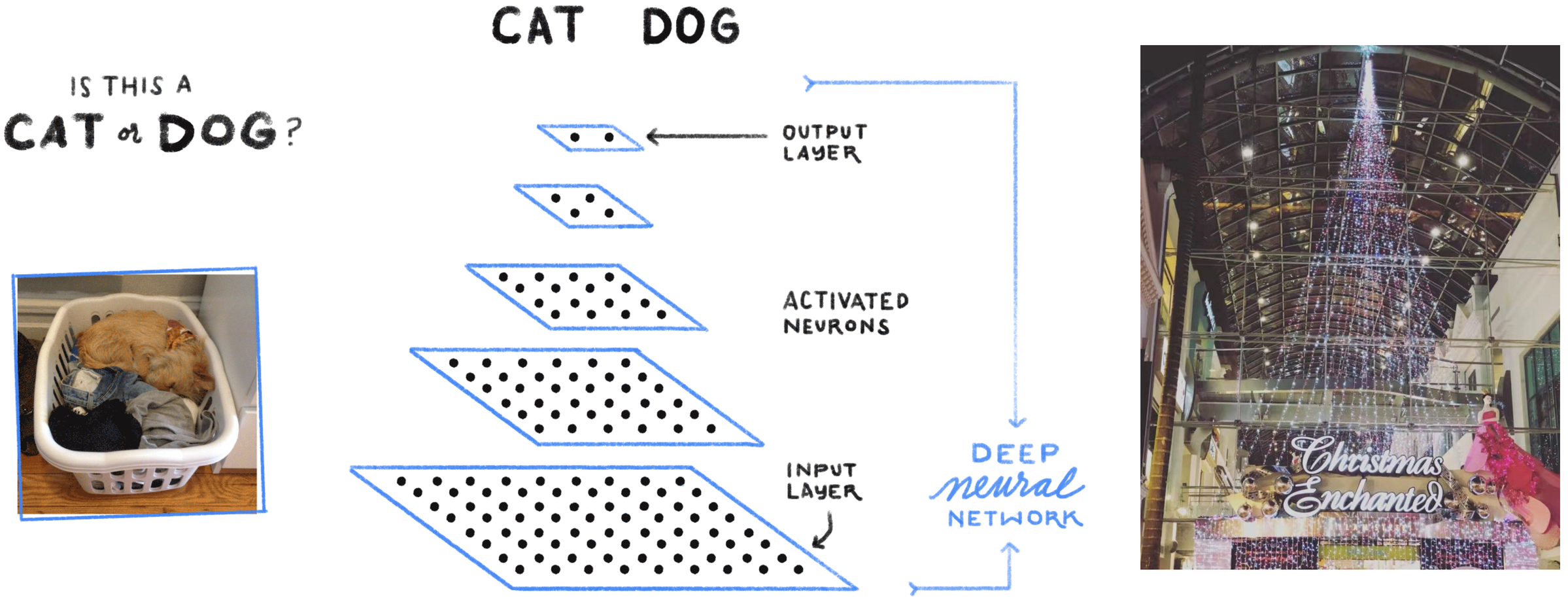
Team name	Entry description	Classification error
WM	Fusion with product strategy	0.168715
WM	Fusion with learnt weights	0.168747
WM	Fusion with average strategy	0.168909
WM	A single model (model B)	0.172876
WM	A single model (model A)	0.173527
SIAT_MMLAB	9 models	0.173605
SIAT_MMLAB	13 models	0.174645
SIAT_MMLAB	more models	0.174795
SIAT_MMLAB	13 models	0.175417
SIAT_MMLAB	2 models	0.175868
Qualcomm Research	Weighted fusion of two models. Top 5 validation error is 16.45%.	0.175978
Qualcomm Research	Ensemble of two models. Top 5 validation error is 16.53%.	0.176559
Qualcomm Research	Ensemble of seven models. Top 5 validation error is 16.68%	0.176766
Trimps-Soushen	score combine with 5 models	0.179824
Trimps-Soushen	score combine with 8 models	0.179997
Trimps-Soushen	top10 to top5, label combine with 9 models	0.180714
Trimps-Soushen	top10 to top5, label combine with 7 models	0.180984
Trimps-Soushen	single model, bn07	0.182357
ntu_rose	test_4	0.193367
ntu_rose	test_2	0.193645
ntu_rose	test_5	0.19397
ntu_rose	test_3	0.194262

Basic CNN Components of Convolutional Neural Network

CNN基本组成

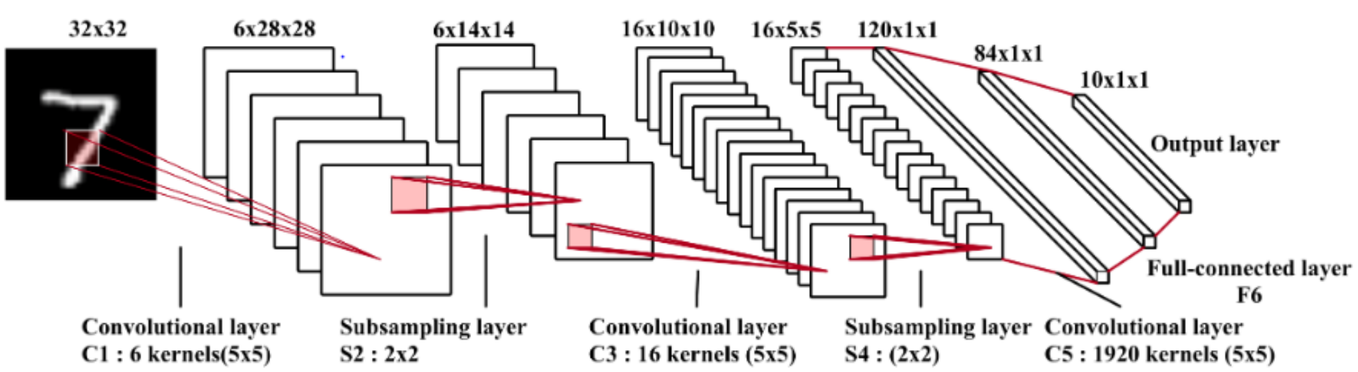

Convolutional Neural Networks
Components
Convolutional Layer
Tiled Convolution
Transposed Convolution
Dilated Convolution
Network in Network
Inception Module
Pooling Layer
Lp Pooling
Mixed Pooling
Stochastic Pooling
Spectral Pooling
Spatial Pyramid Pooling
Multi-scale Orderless Pooling
Activation Function
ReLU
LReLU
PReLU
RReLU
ELU
Maxout
Probout
Loss Function
Hinge Loss
Softmax Loss
Contrastive Loss
Triplet Loss
KL Divergence
Regularization
Lp-norm
Dropout
DropConnect
Optimization
Data Augmentation
Weight Initialization
SGD
Batch Normalization
Shortcut Connections
Fast Processing
FFT
Structured Transforms
Low Precesion
Weight Compression
Sparse Convolution
Applications
Image Classification
Object Detection
Object Tracking
Pose Estimation
Text Detection & Recognition
Visual Saliency Detection
Action Recognition
Scene Labeling
Speech & Natural Language Processing

CNN基本组成

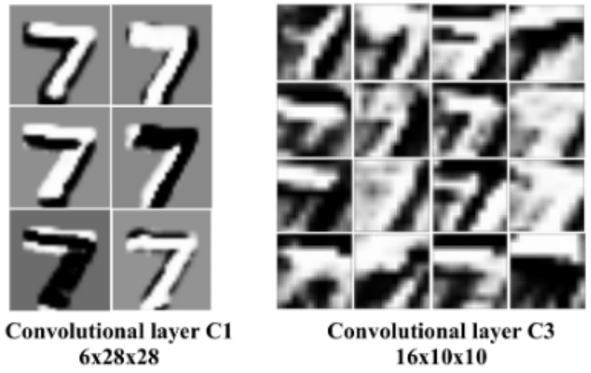

IS THIS A
CAT or DOG?
CAT DOG
OUTPUT LAYER
ACTIVATED NEURONS
INPUT LAYER
DEEP neural NETWORK
Christmas Enchanted

CNN基本组成

❑ LeNet-5

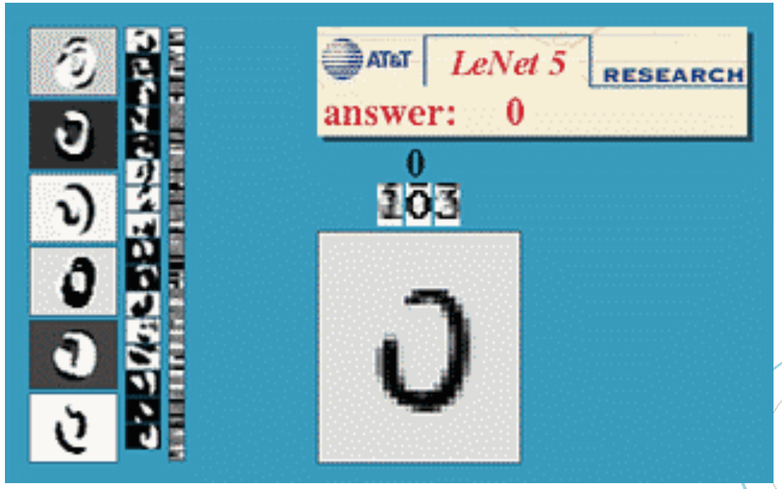


(a) LeNet-5 network

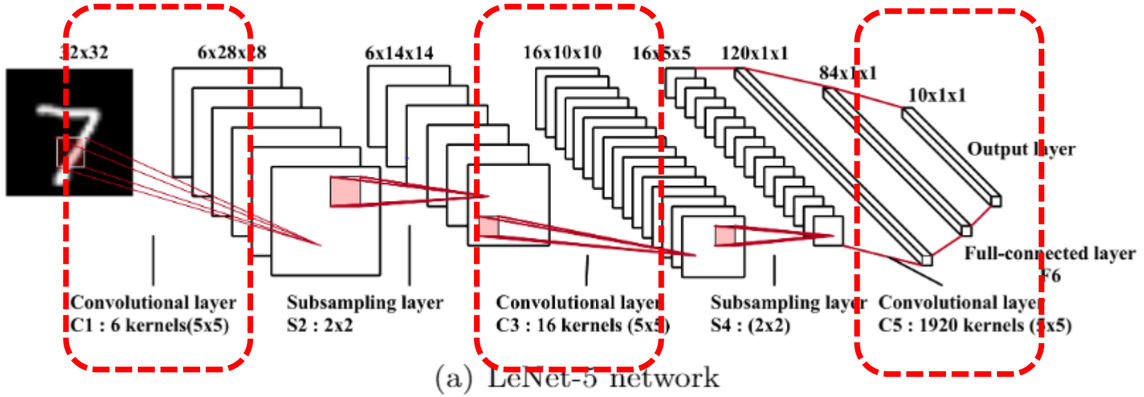


(b) Learned features

- 输入尺寸：32*32
- 卷积层/Convolutional Layer：3个
- 降采样层/Downsampling/Subsampling/Pooling：2个
- 全连接层/Fully-connected Layer：1个
- 输出/Output layer (SVM etc.)：10个类别（数字0-9的概率）

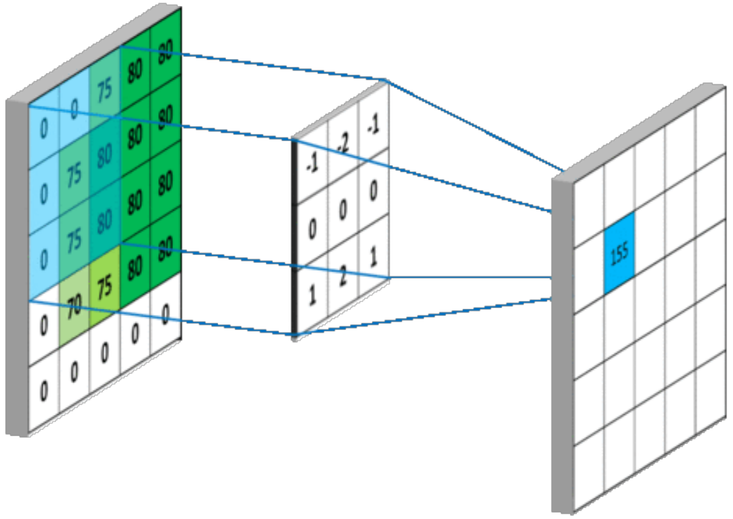


CNN基本组成–卷积层 (C1,C3, C5)

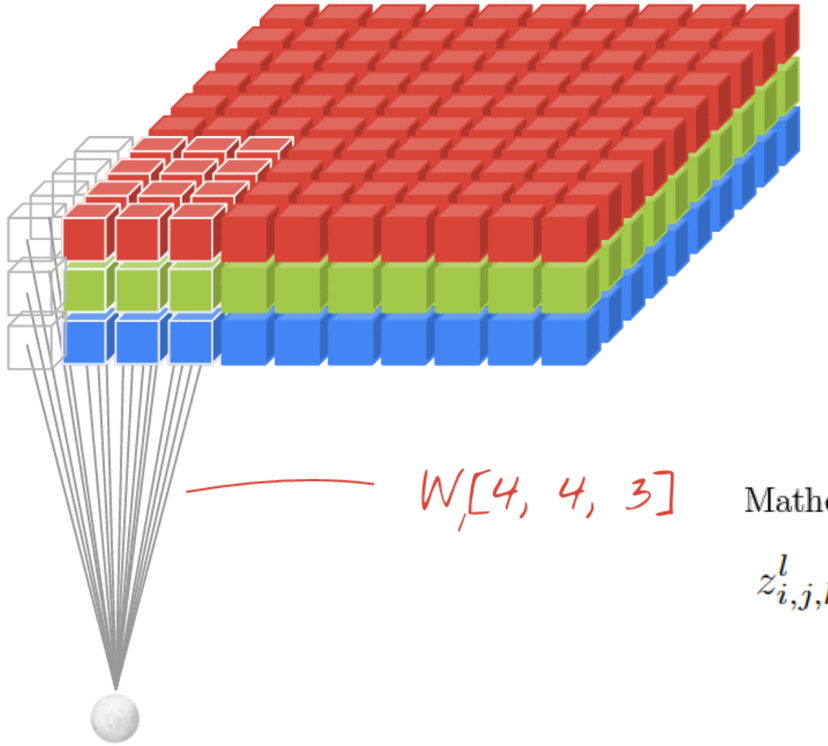


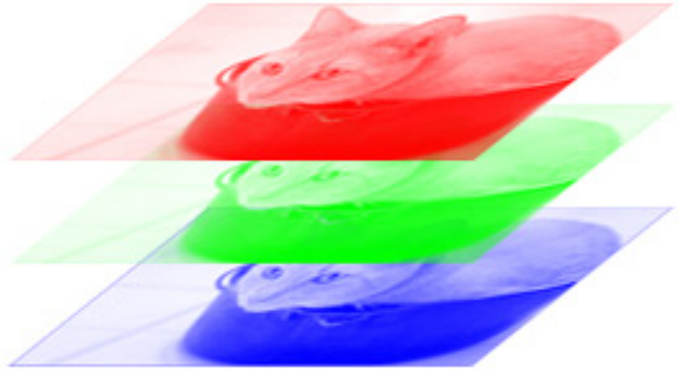
(a) LeNet-5 network

- Convolution layer 1层，也就是卷积层的第一层。
- 一共有6个Map，每个feature map分辨率是28*28
- 每个神经元的分辨率则是（32-28+1）*（32-28+1）=5*5，我们可以把这个神经元看作一个滤波器，而这就是局部感受野，因为一个滤波器只感受5*5的风景。
- 又因为权值共享，同Map下所有的神经元感受的特征都是一样的，所以这整个Map都只能算一个滤波器。
- 每个Map算一个滤波器，每个滤波器有（5*5+1）个参数
- **28*28个神经元是重复被6个滤波器使用的**
- 每个神经元一共有（5*5+1）*6=156个参数



CNN基本组成—卷积：2D Conv with 3D-inputs





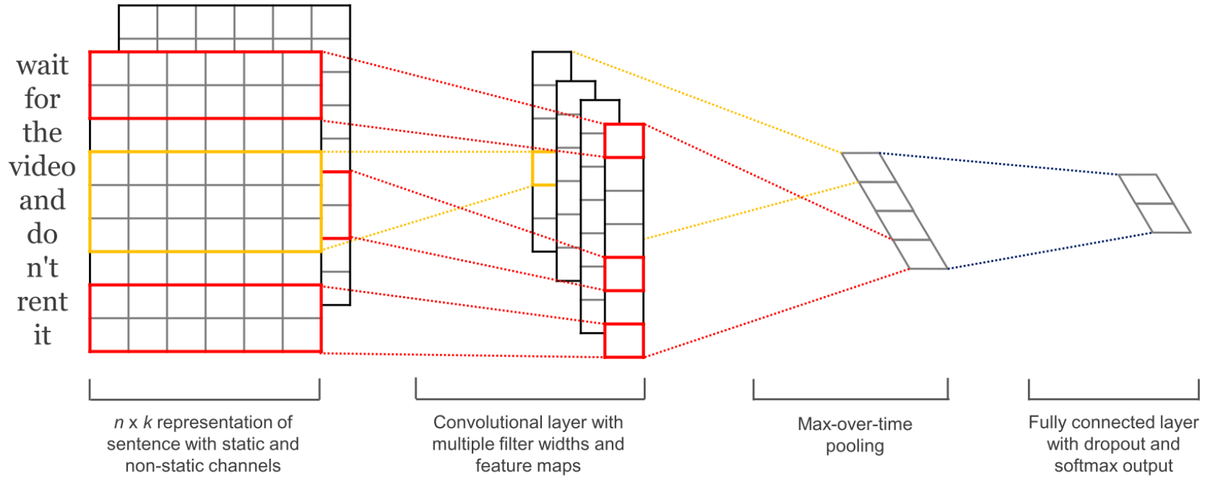
=

Mathematically, the feature value atlocation (i,j) in the k−th feature map ofl-th layer, $z^l_{i,j,k}$, is calculated by:

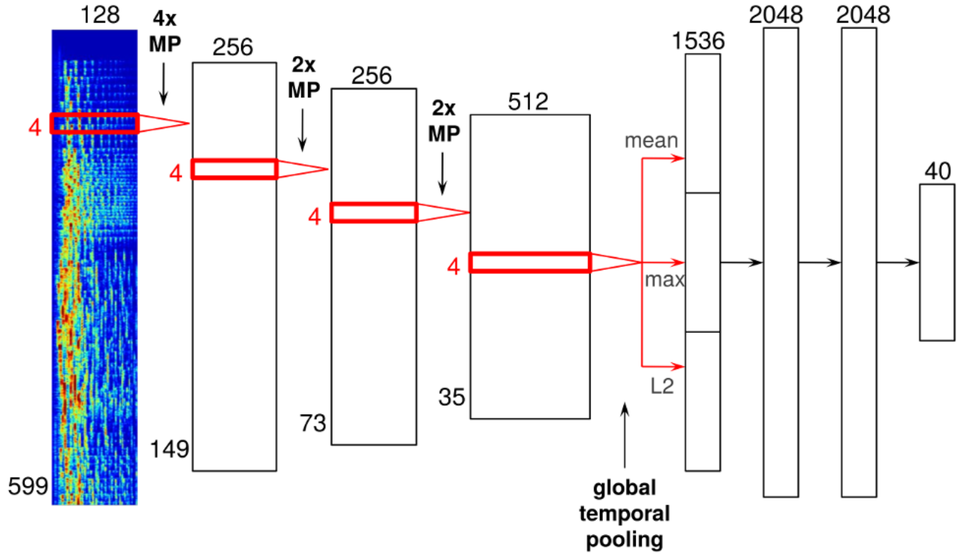
$$z^l_{i,j,k} = \mathbf{w}^{l\,T}_k \mathbf{x}^l_{i,j} + b^l_k$$

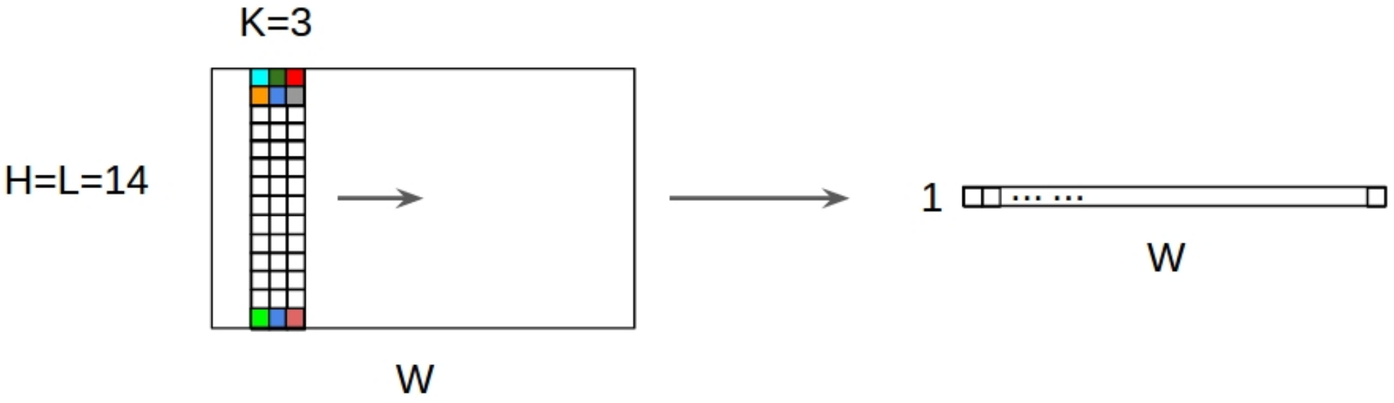
CNN基本组成–卷积：1D Convolutions with 2D INPUT

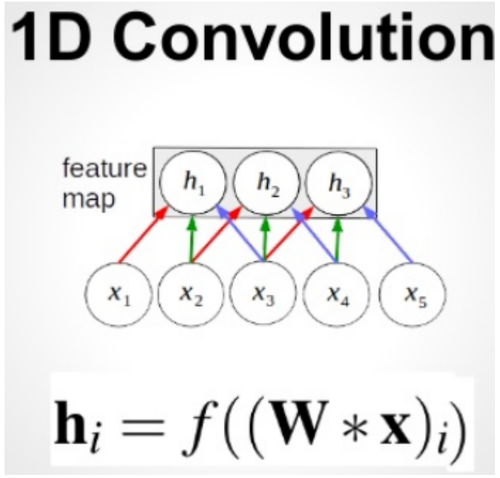
文本



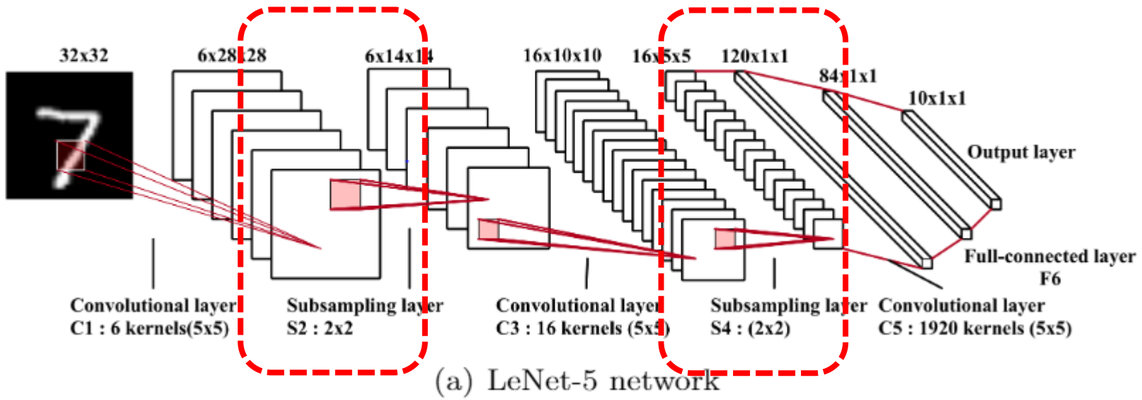
语音



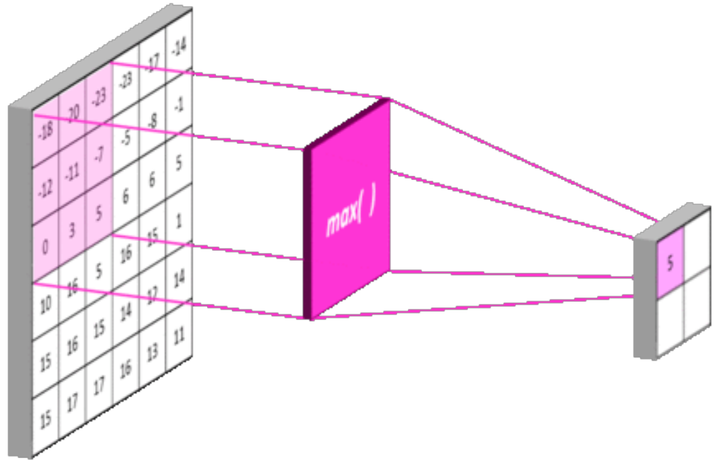




CNN基本组成–池化层 (S2,S4)



(a) LeNet-5 network



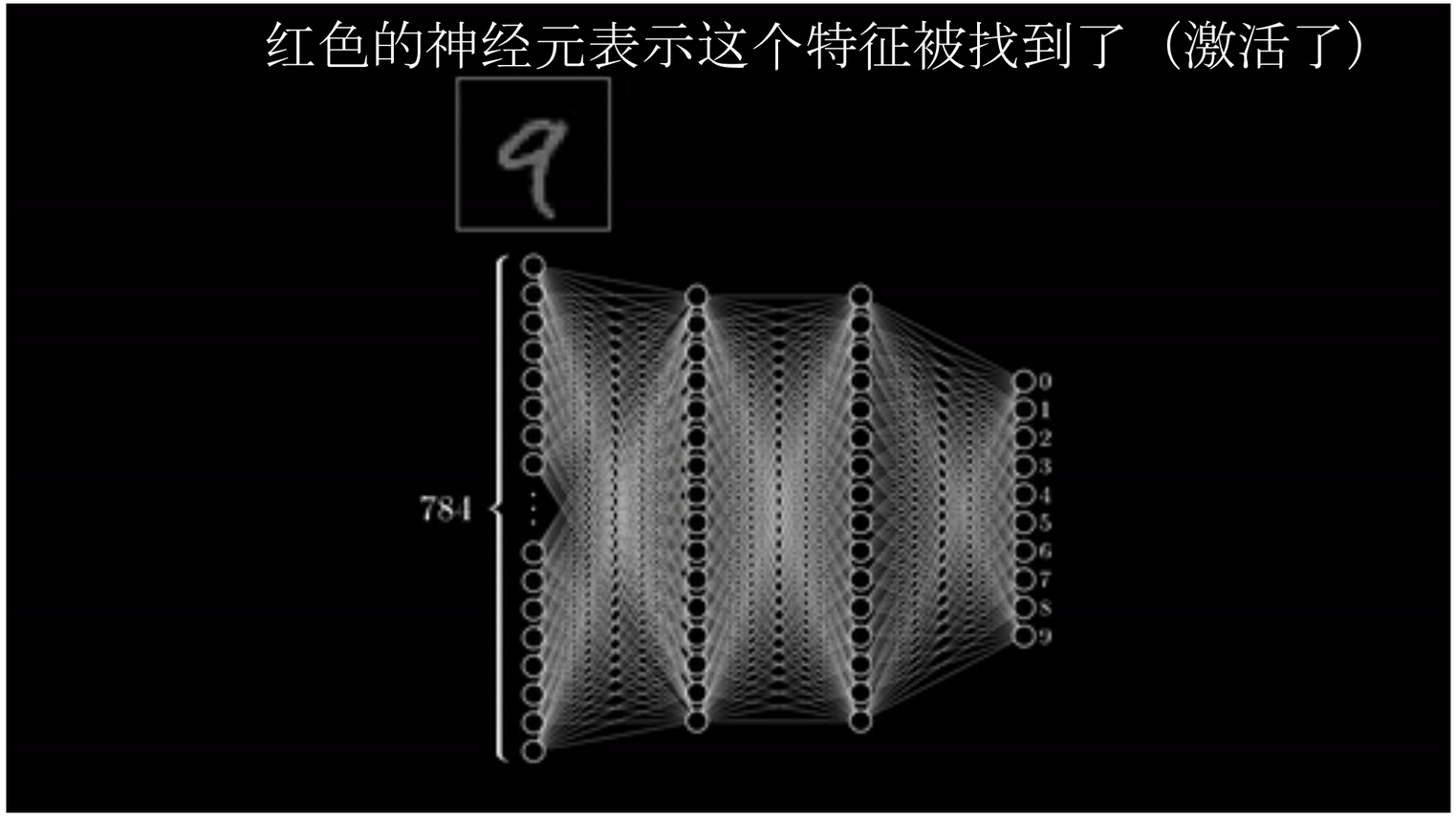
- S2层叫做Pooling Layer or Downsampling Layer or Subsampling Layer
- 提高泛化性。6个Map，每个Map14*14，size=2*2，卷积层有重叠，而采样层无重叠，所以每个Map=上一层Map分辨率28*28/size 2*2=14*14。(Max-pooling)
- 而采样层又是特殊的卷积层，只不过是卷积核为2*2（pool size）

1. 略微提高transformation invariance
2. 视皮层有类似的侧抑制效应
3. **减少参数防止过拟合**

CNN基本组成—全连接层

全连接层一般负责维度变换→分类或者回归

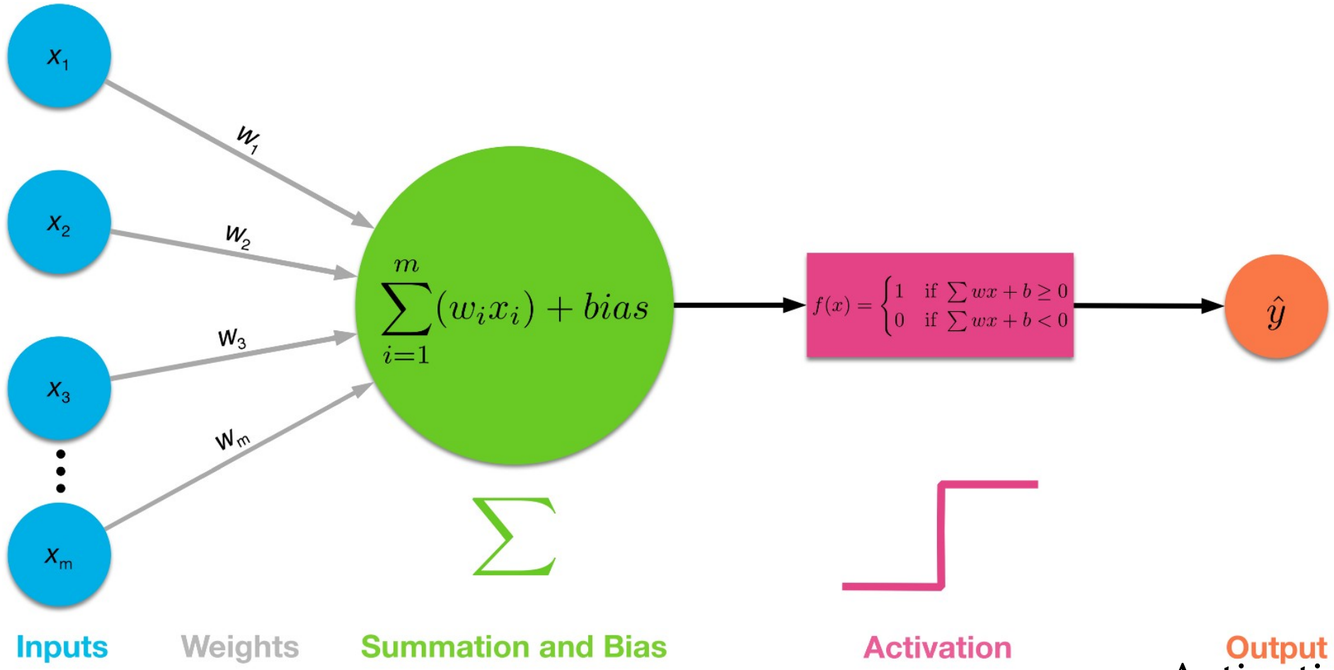
--卷积层本来就是全连接的一种简化形式:不全连接+参数共享，同时还保留了空间位置信息。这样大大减少了参数并且使得训练变得可控。

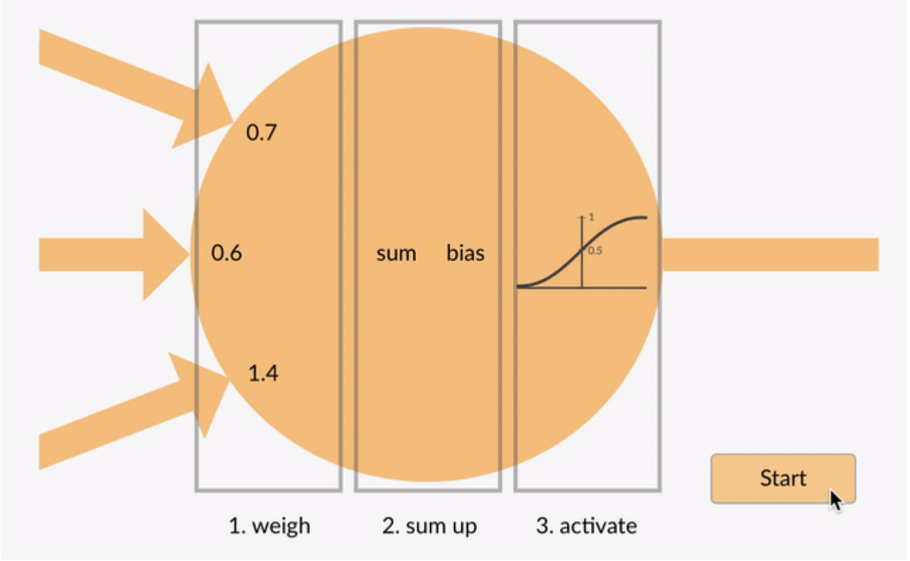


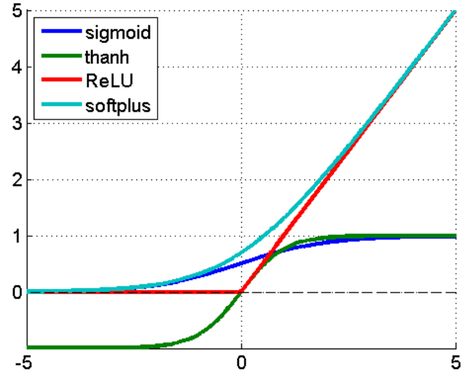
全连接层参数特多（可占整个网络参数80%左右），近期一些性能优异的网络模型如ResNet和GoogLeNet等均用全局平均池化（global average pooling，GAP）取代全连接层来融合学到的深度特征

CNN基本组成—激活函数

激活函数是用来加入非线性因素的，因为线性模型的表达能力不够







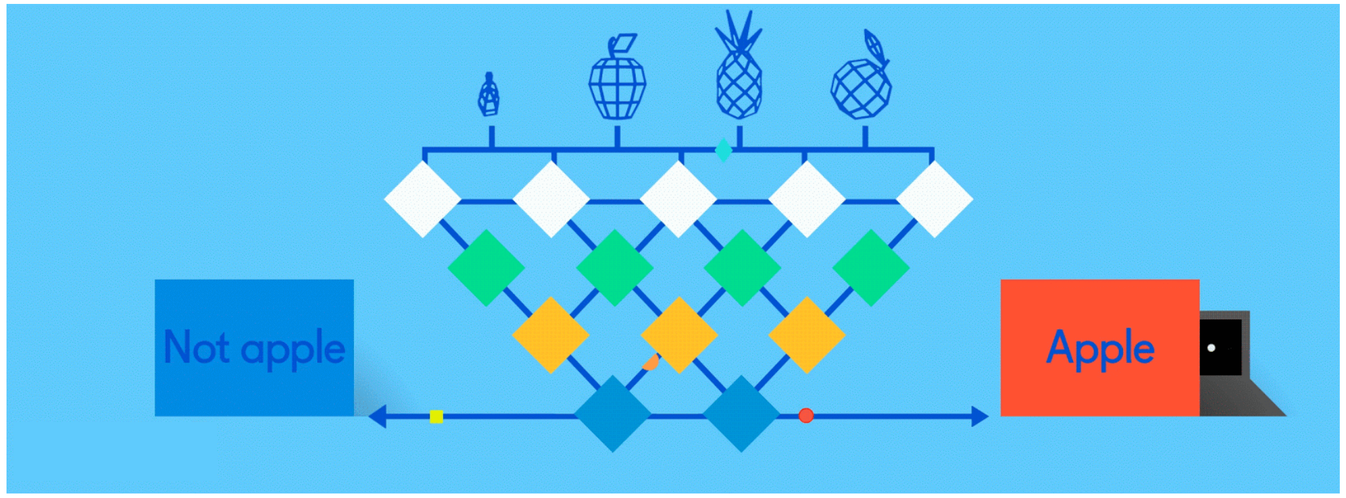
Activation function/翻译成激活函数

不要误解是指这个函数去激活什么，而是指如何把“**激活的神经元的特征**”通过**函数把特征保留并映射出来**，这是神经网络能解决非线性问题关键。

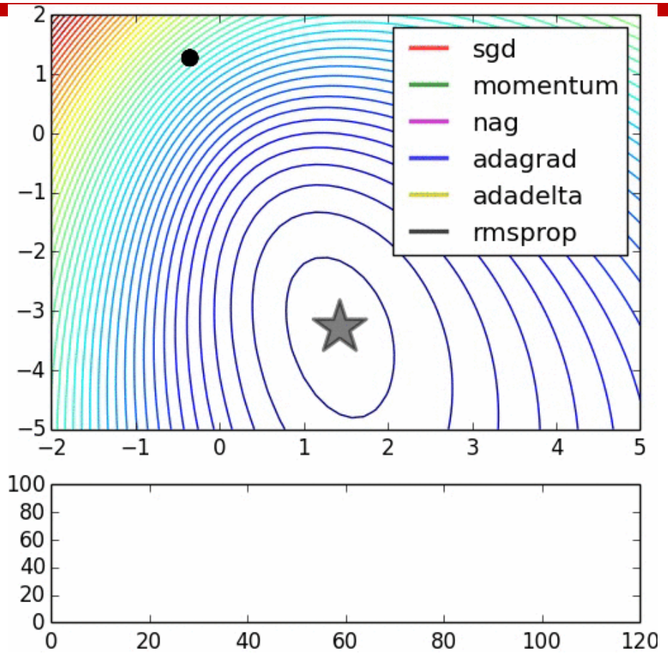
激活函数众所周知有tanh, sigmoid, ReLU等。

1. Tanh 双切正切函数，取值范围[-1,1]
2. Sigmoid 采用S形函数，取值范围[0,1]
3. ReLU 简单而粗暴，大于0的留下，否则一律为0。

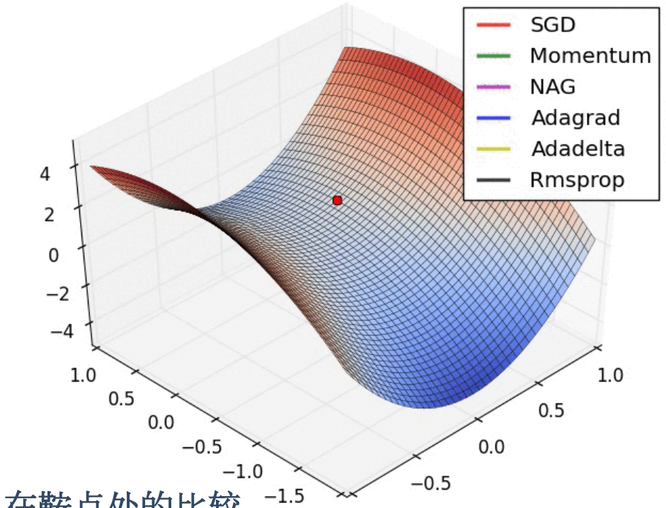
CNN训练（待续）



1. **SGD:mini-batch gradient descent**. SGD就是每一次迭代计算梯度，然后对参数进行更新，是最常见的优化方法了。
2. **Momentum**: momentum是模拟物理里动量的概念。它在相关方向加速SGD，抑制振荡，从而加快收敛. 在梯度指向同一方向的维度，momentum项增加; 在梯度改变方向的维度，momentum项减少更新.
3. **Adagrad**: 此方法能对不常见的参数进行较大的更新，对于常见参数更新较小，不用手动调节学习率
4. **Adadelta**: Adadelta是对Adagrad的扩展。Adagrad会累加之前所有的梯度平方，而Adadelta只累加固定大小的项，并且也不直接存储这些项，仅仅是计算对应的平均值。Adadelta甚至不用设置默认值。
5. **RMSprop**: RMSprop类似于Adadelta
6. **Adam**: Adam(Adaptive Moment Estimation)加上了bias校正和momentum，在优化末期，梯度更稀疏时，它比RMSprop稍微好点 (**RNN用的比较多**)

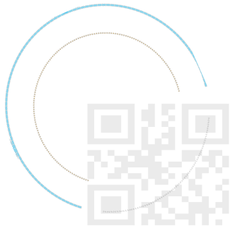


损失平面等高线

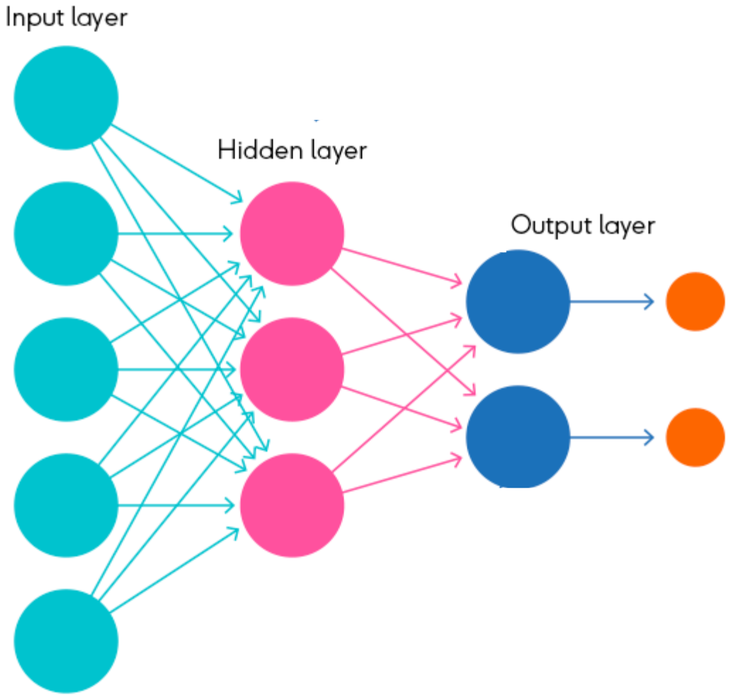


在鞍点处的比较

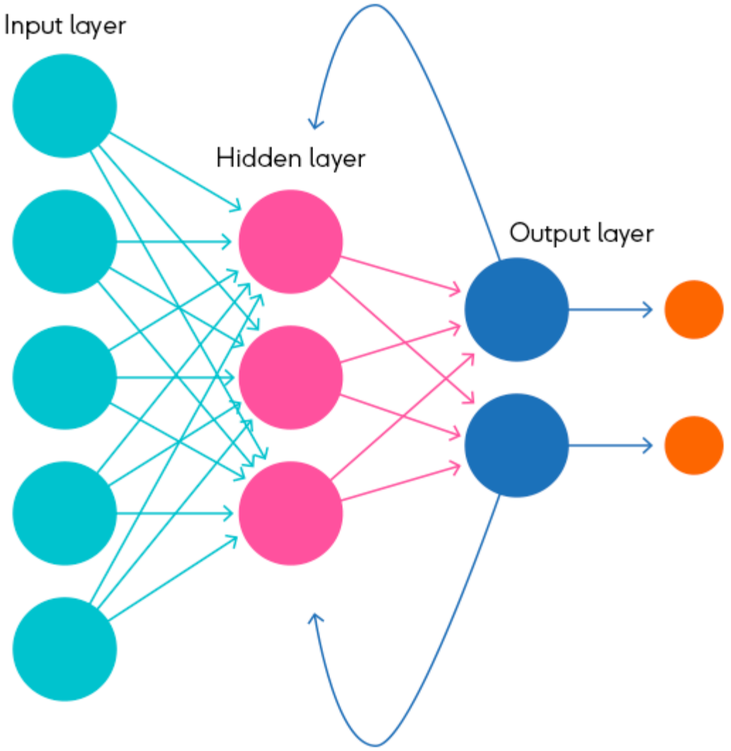
Recurrent Neural Network



什么是RNN？



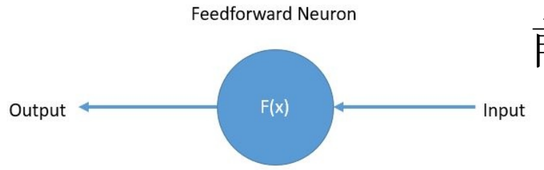
Feedforward NN 的结构是 DAG (有向无环图)



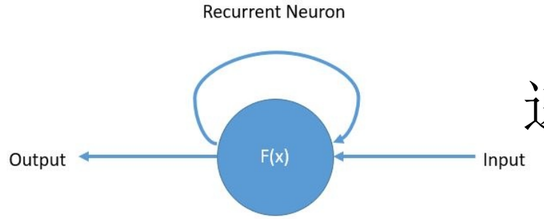
Recurrent NN 的结构中至少有一个环

什么是RNN？

- 递归神经网络（RNN）是两种人工神经网络的总称。一种是时间递归神经网络（recurrent neural network）；
- 另一种是结构递归神经网络（recursive neural network）。
- 时间递归神经网络的神经元间连接构成矩阵，而结构递归神经网络利用相似的神经网络结构递归构造更为复杂的深度网络。

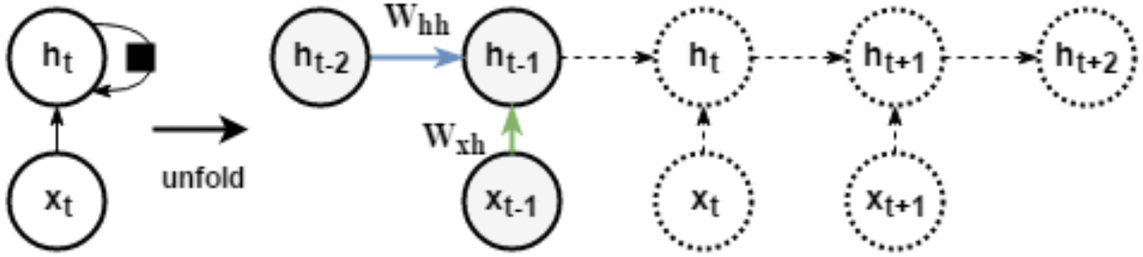


前向网络

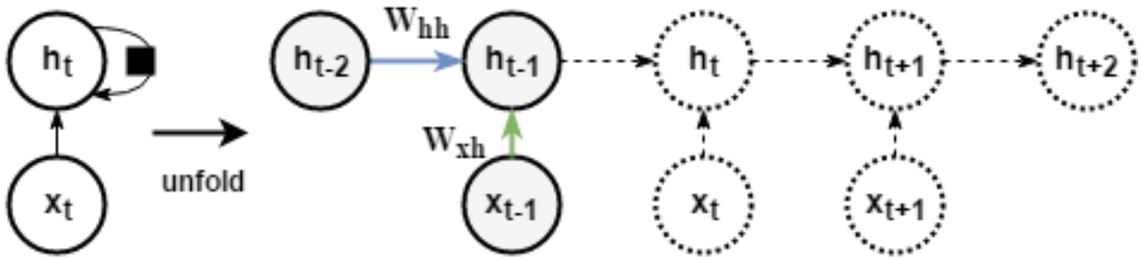


递归网络

RNN一般指代时间递归神经网络。单纯递归神经网络因为无法处理随着递归，权重指数级爆炸或消失的问题（Vanishing gradient problem），难以捕捉长期时间关联；**而结合不同的LSTM可以很好解决这个问题。**



Basic RNN的具体表达式



$$
\begin{aligned}
h_*^t &= W_{hx}x^t + W_{hh}h^{t-1} + b_h \\
h^t &= \sigma(h_*^t) \\
o_*^t &= W_{oh}h^t + b_o \\
o^t &= \theta(o_*^t)
\end{aligned}
$$

其中，x^t 表示 t 时刻的输入，o^t 表示 t 时刻的输出，h^t 表示 t 时刻 Hidden Layer 的状态。

Basic RNN的具体训练方法

RNN的反向传播算法Backpropagation Through Time (BPTT)

- RNN基本传递函数

$$s_t = tanh(Ux_t + Ws_{t-1})$$

$$\hat{y}_t = softmax(Vs_t)$$

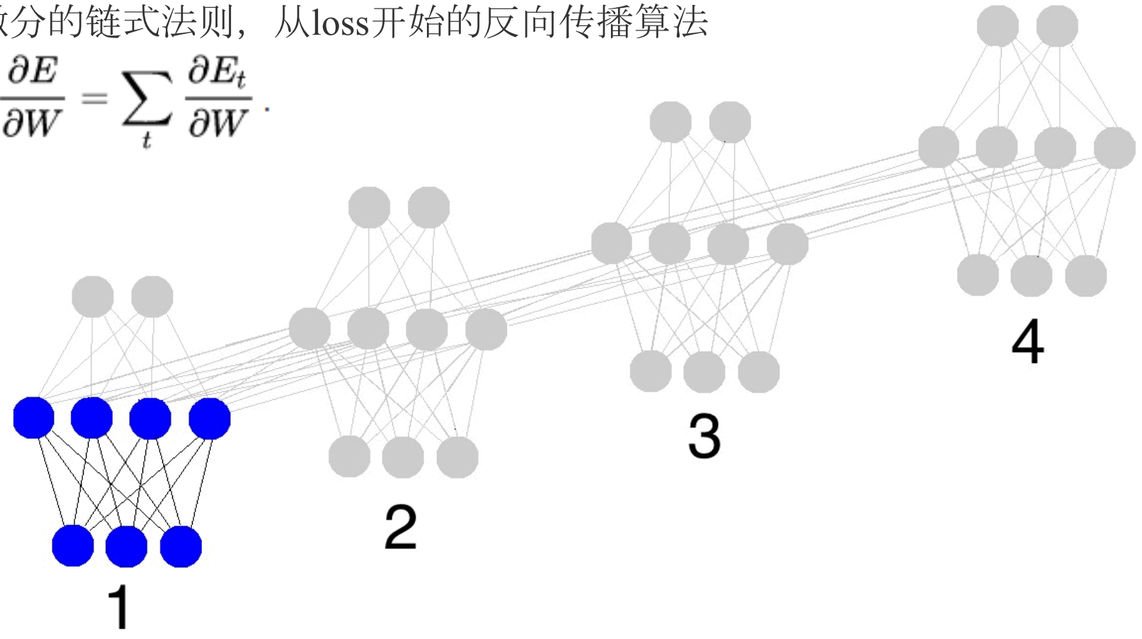
- 使用交叉熵作为代价函数:

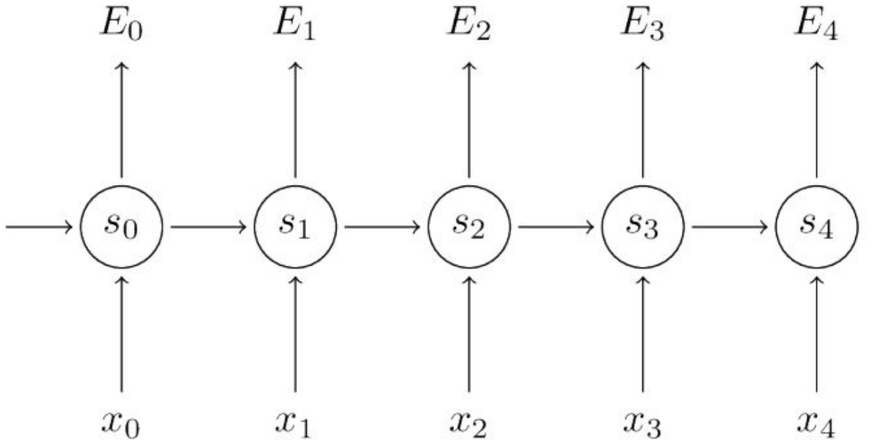
$$E_t(y_t, \hat{y}_t) = -y_t \log \hat{y}_t$$

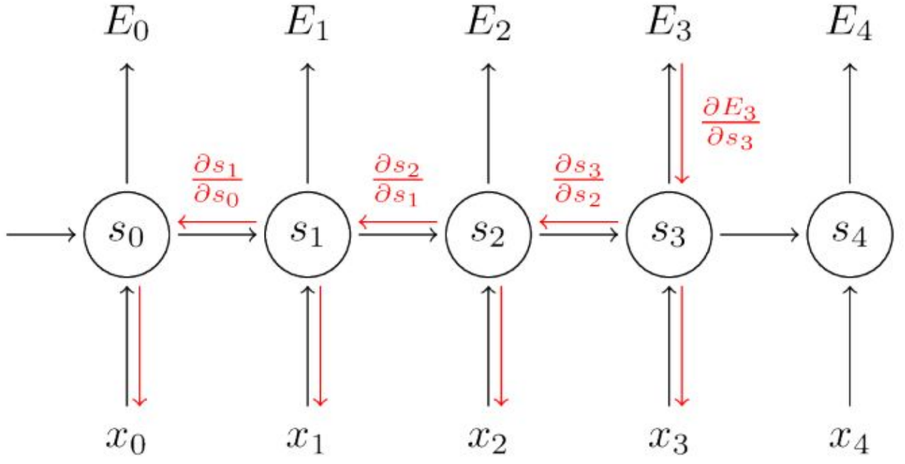
$$E(y, \hat{y}) = \sum_t E_t(y_t, \hat{y}_t) = -\sum_t y_t \log \hat{y}_t$$

- 微分的链式法则，从loss开始的反向传播算法

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W}.$$

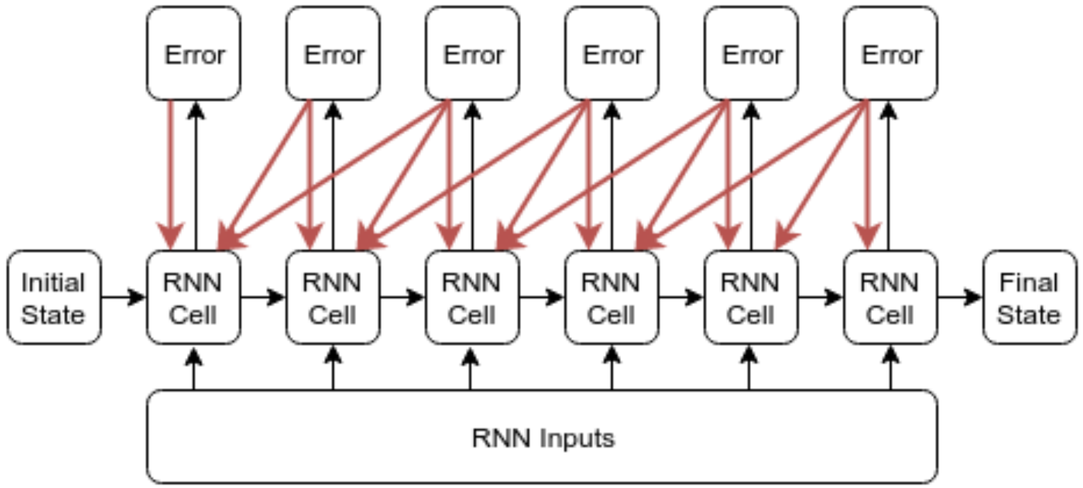






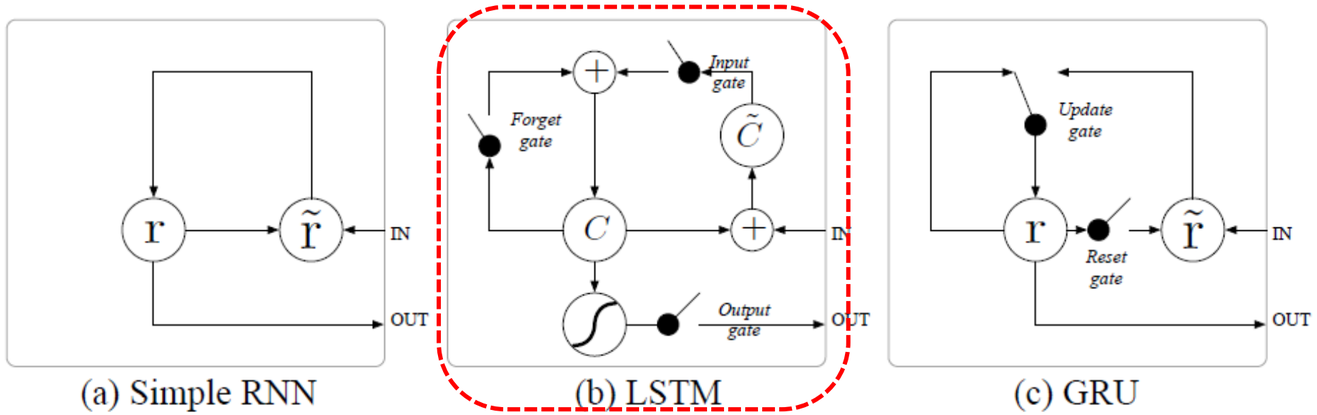
$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W}$$

Basic RNN的具体训练方法



3 types of RNNs For NLP

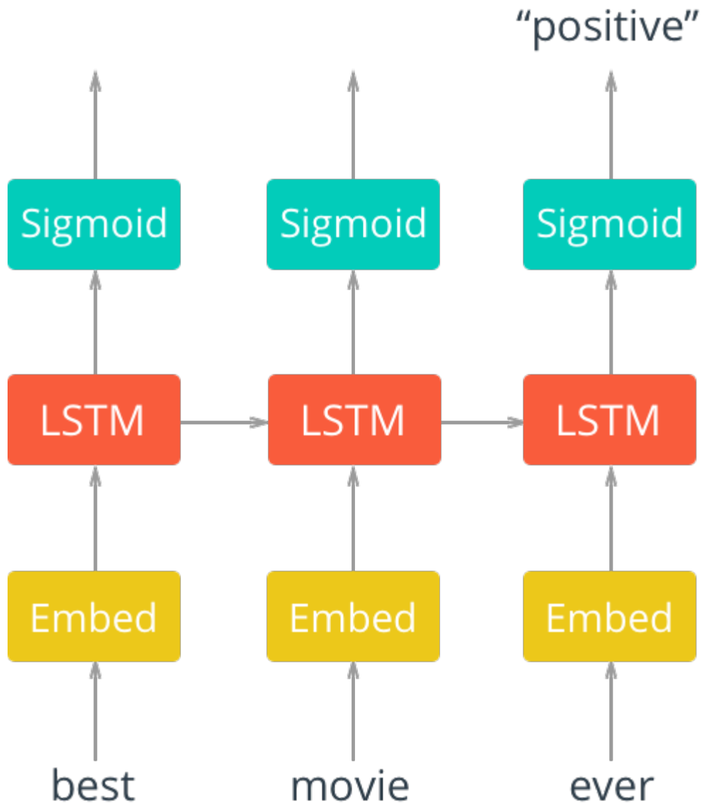
Long short-term memory (LSTM)



(a) Simple RNN (b) LSTM (c) GRU

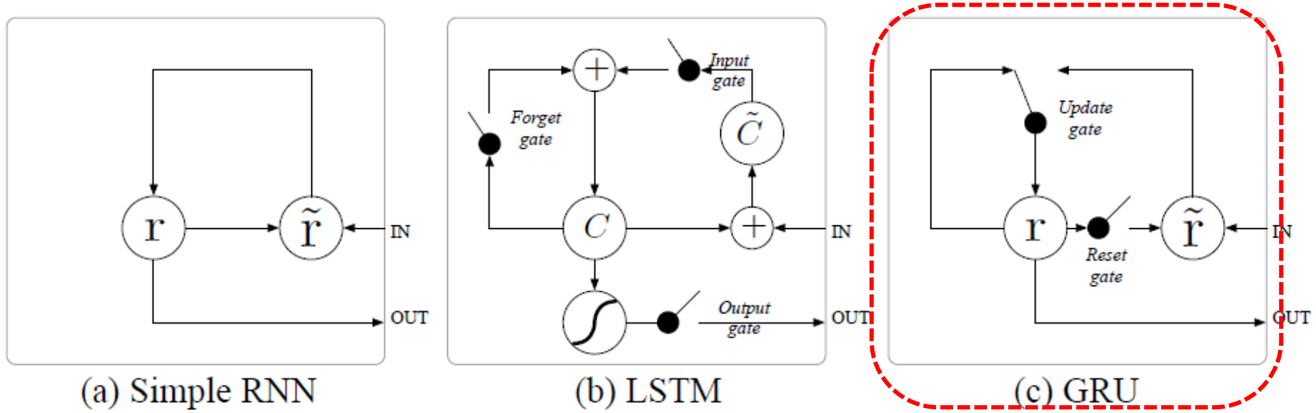
$$f_t = \sigma(X_t * U_f + H_{t-1} * W_f)$$
$$\bar{C}_t = \tanh(X_t * U_c + H_{t-1} * W_c)$$
$$I_t = \sigma(X_t * U_i + H_{t-1} * W_i)$$
$$O_t = \sigma(X_t * U_o + H_{t-1} * W_o)$$

$$C_t = f_t * C_{t-1} + I_t * \bar{C}_t$$
$$H_t = O_t * \tanh(C_t)$$



3 TYPES OF RNNs FOR NLP

Gated recurrent neural network (GRU)



(a) Simple RNN (b) LSTM (c) GRU

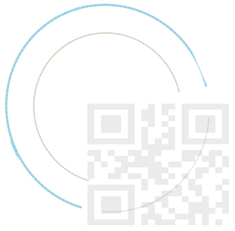
GRU（Gate Recurrent Unit）是循环神经网络（Recurrent Neural Network, RNN）的一种。和LSTM（Long-Short Term Memory）一样，也是为了解决长期记忆和反向传播中的梯度等问题而提出来的。

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

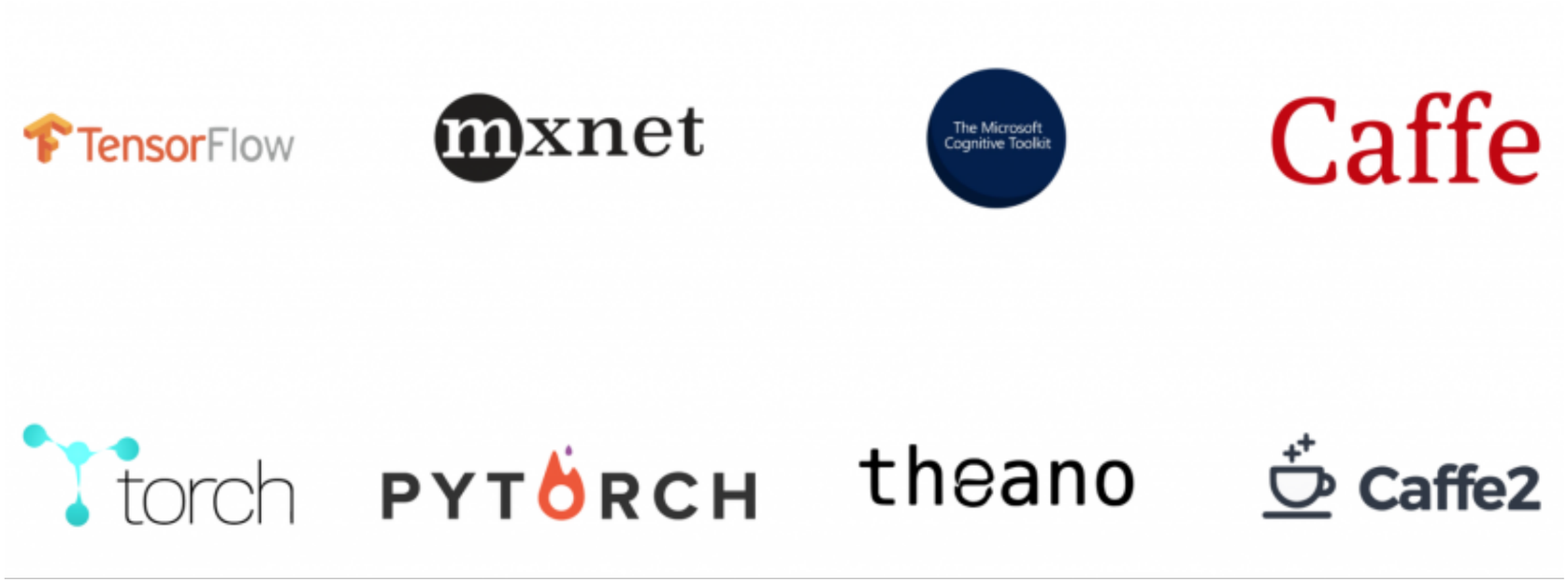
We choose to use Gated Recurrent Unit (GRU) (Cho et al., 2014) in our experiment since it performs similarly to LSTM (Hochreiter & Schmidhuber, 1997) but is computationally cheaper.

MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS（2017）

Deep Learning Toolbox



Deep learning tools



哪个最好？

当然，没有最好的，只有合适的

“你要知道梨子的滋味，你就得变革梨子，亲口吃一吃。你要知道原子的组成同性质，你就得实行物理学和化学的实验，变革原子的情况。你要知道革命的理论和方法，你就得参加革命。一切真知都是从直接经验发源的。但人不能事事直接经验，事实上多数的知识都是间接经验的东西，这就是一切古代的和外域的知识。”——《实践论》毛泽东

END