Deep Vision Learning: From CNNs to Transformers

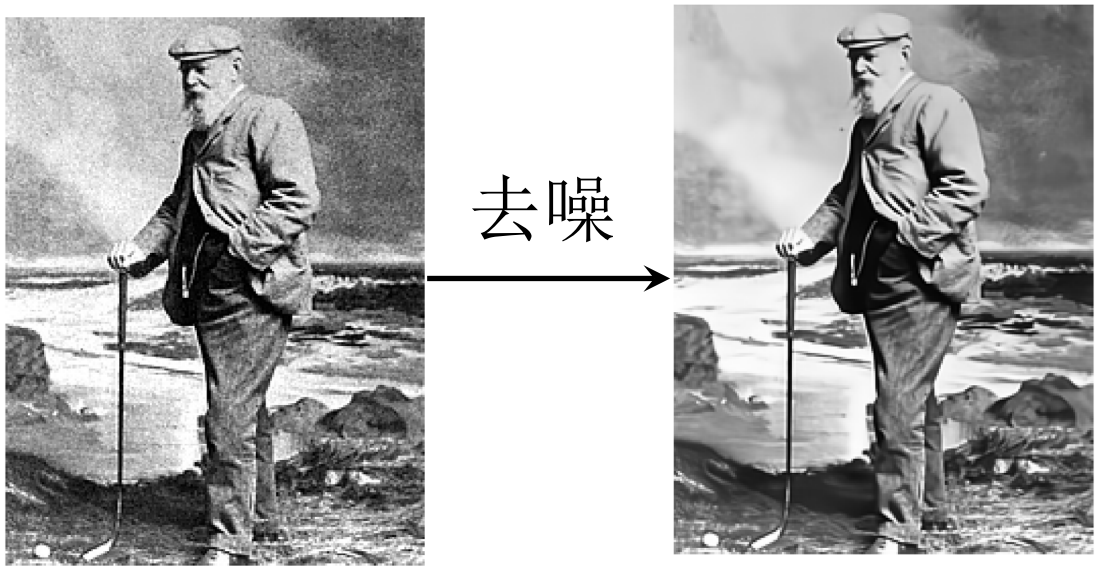
Wangmeng Zuo

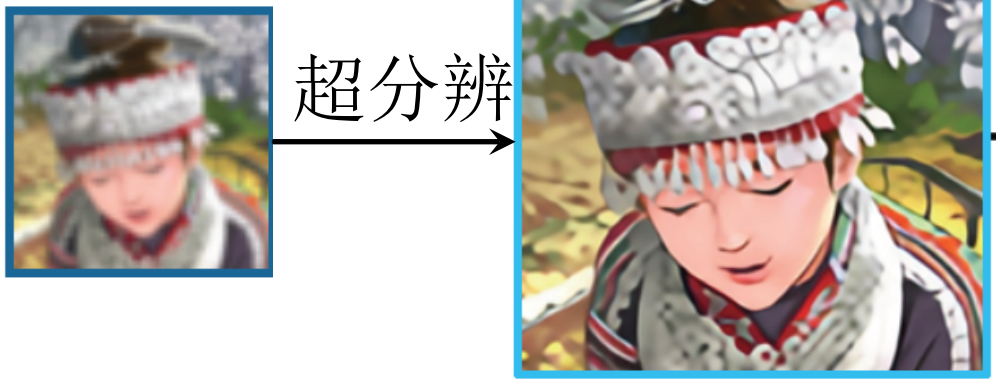
Center on Machine Learning Research
Harbin Institute of Technology

Content

- Introduction to Computer Vision
- Traditional Neural Networks and Their Limitations
- CNN and Recent Progress
  - CNN
  - Rectified Linear Units (ReLU)
  - Dropout / Batch Normalization
  - Representative Network Architectures
- Vision Transformers and Recent Progress

典型视觉学习任务（底层视觉）



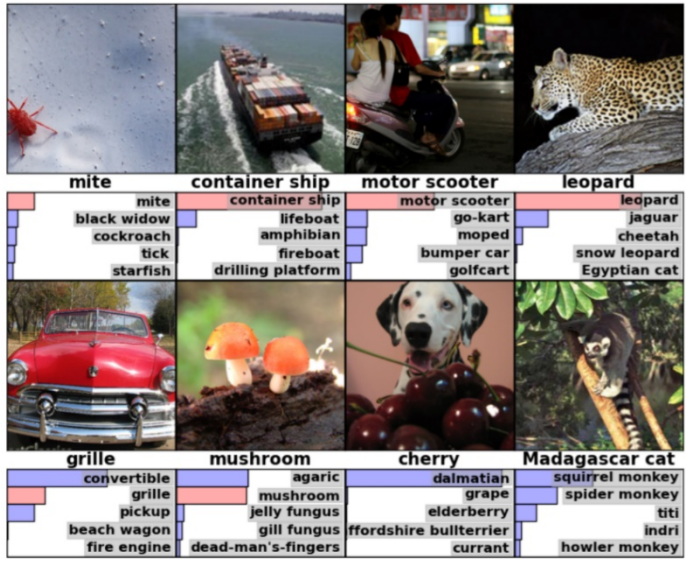


风格迁移

修复

典型视觉学习任务（视觉理解）

图像级分类



边界框级检测

物体关系预测

像素级分割

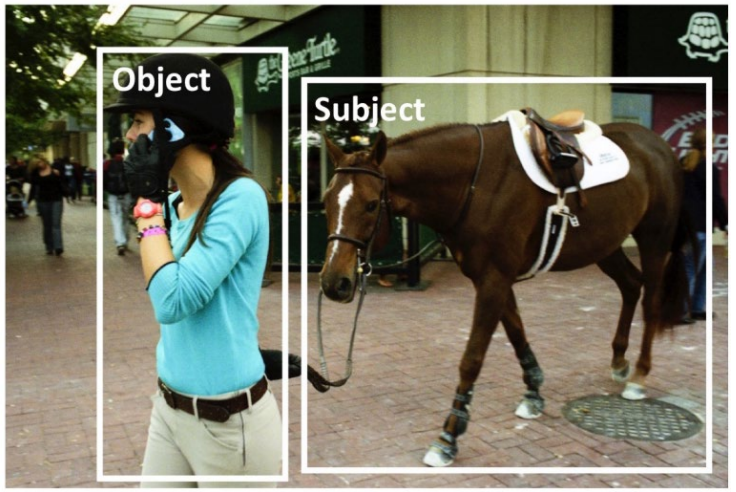
典型视觉任务（视觉-语言）

自然语言描述

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."



指代消解

视觉问答

Q: Does this foundation have any sunscreen?
A: yes

Q: What is this?
A: 10 euros

Q: What color is this?
A: green

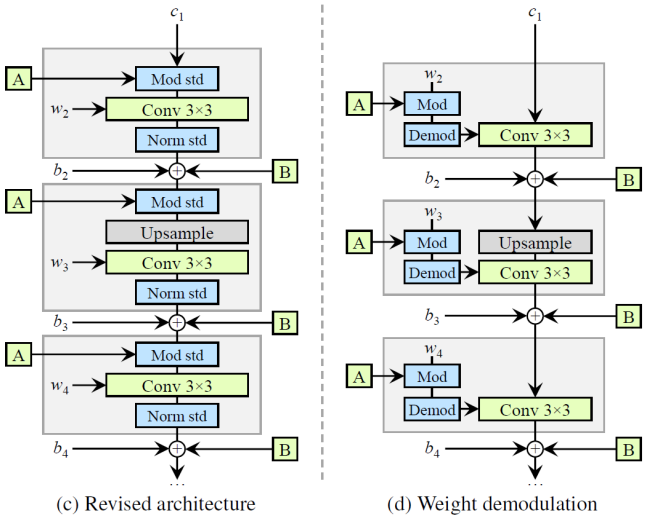
Q: Please can you tell me what this item is?
A: butternut squash red pepper soup

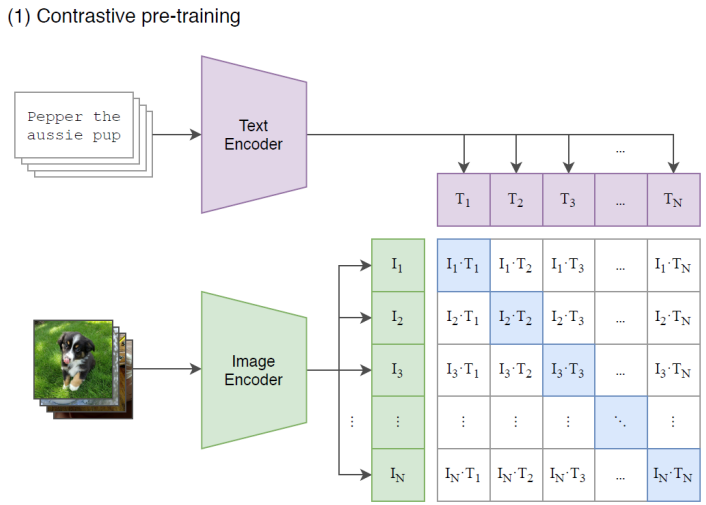
Q: Is it sunny outside?
A: yes

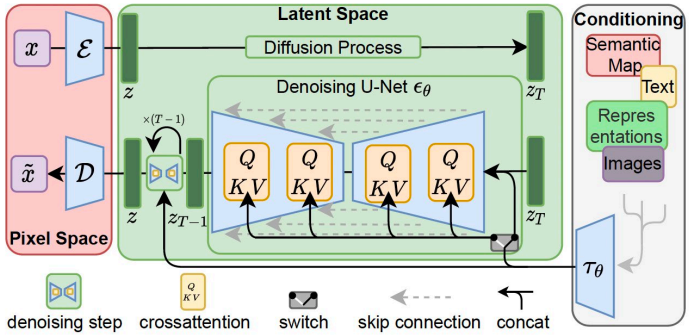
Q: Is this air conditioner on fan, dehumidifier, or air conditioning?
A: air conditioning

Progress in VL Pretraining

- Image Generation
  - StyleGAN2
- Image-Text Matching
  - CLIP
- Text-to-Image Generation
  - DALLE-2
  - Stable Diffusion







DALL.E-2: unCLIP

an espresso machine that makes coffee from human souls, artstation

panda mad scientist mixing sparkling chemicals, artstation

a corgi's head depicted as an explosion of a nebula

a dolphin in an astronaut suit on saturn, artstation

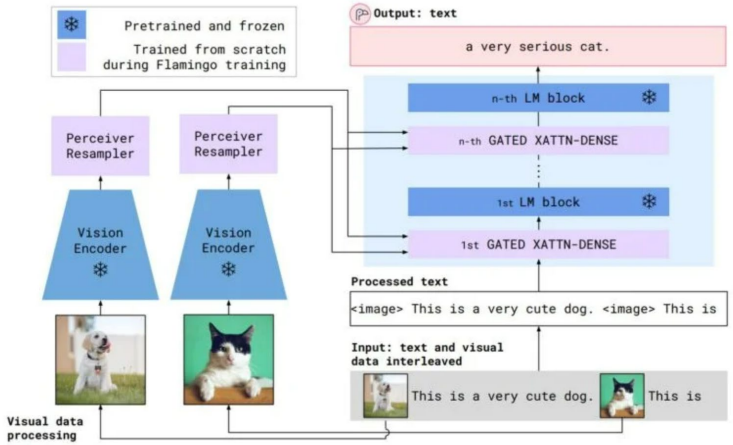
a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese

a teddybear on a skateboard in times square

Progress in VL Pretraining

- Large Language Model
  - BERT
  - ChatGPT

- Vision-Language Pre-training
  - Flamingo, GATO
  - GPT-4



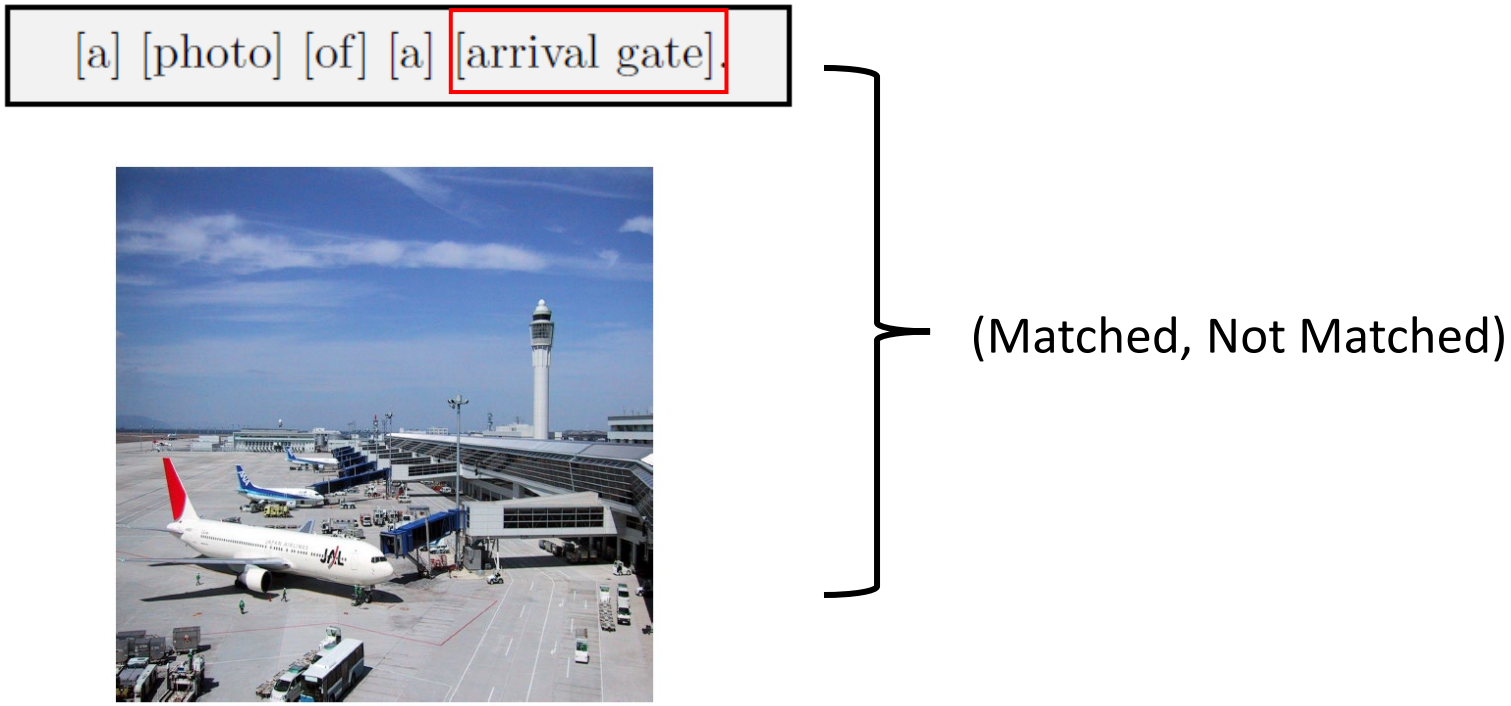
GPT-4 visual input example, Extreme Ironing:

User What is unusual about this image?

Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg

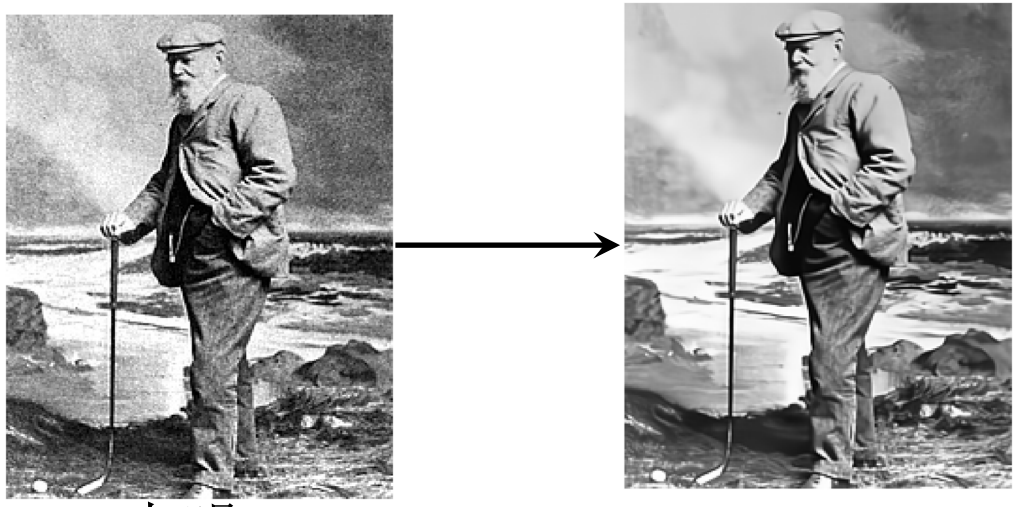
GPT-4 The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

视觉学习任务的统一处理范式



- Similar to other tasks
- Most vision tasks can be reformatted from vision-language perspective
- Also, many vision-language resources are available

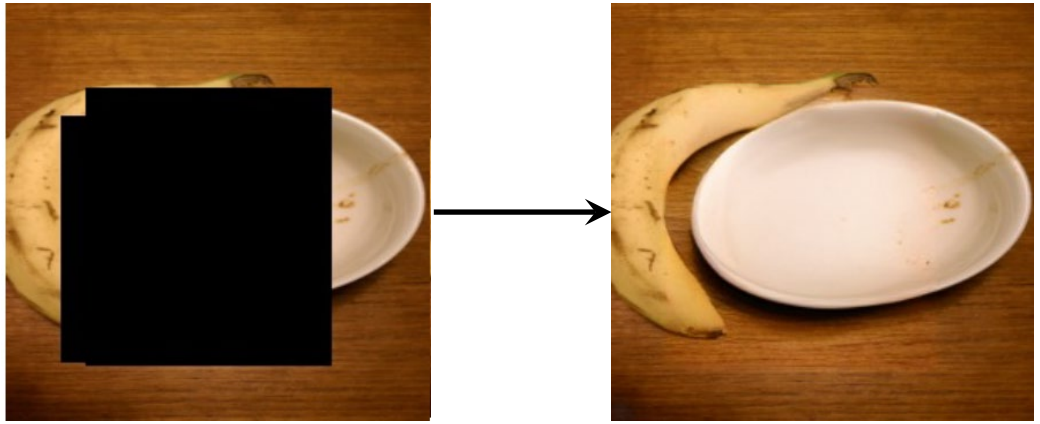
典型视觉学习任务（底层视觉）

- 

去噪

超分辨

修复



The banana is laying next to an almost empty bowl.

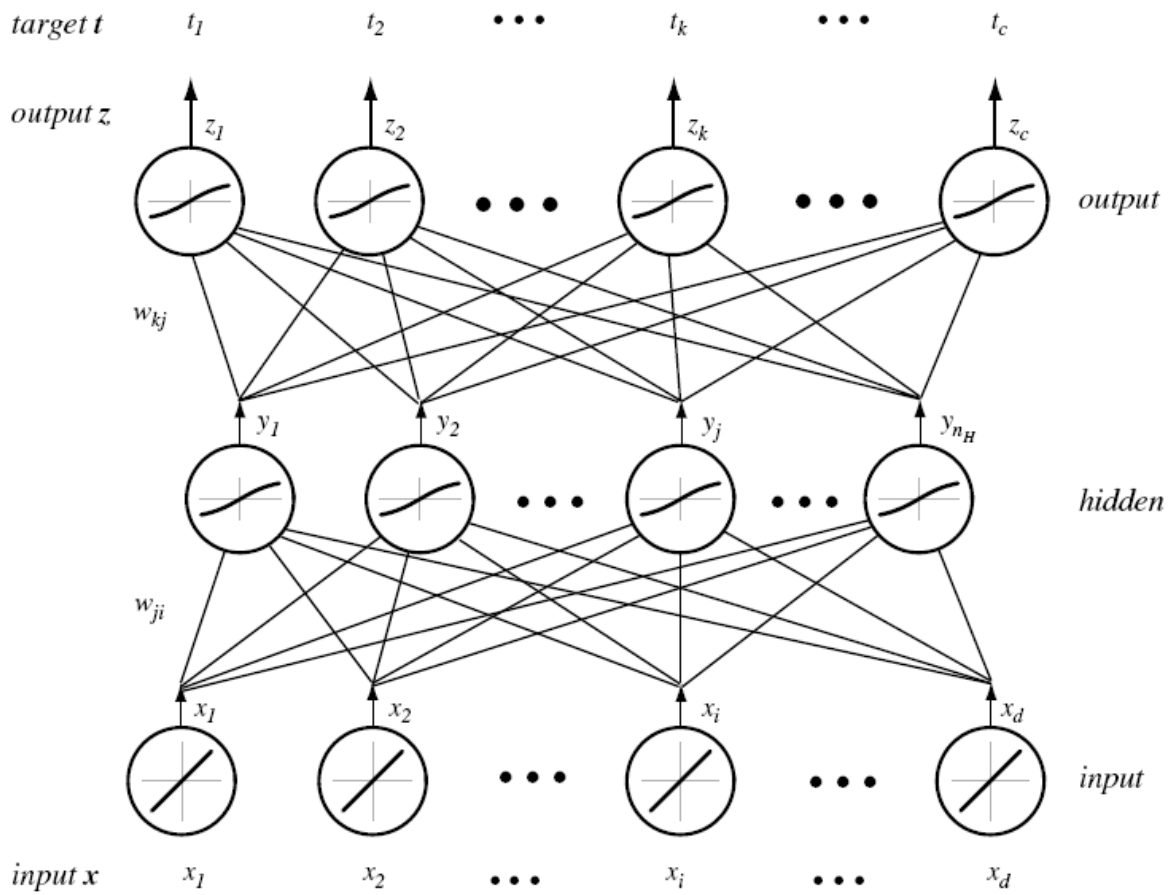
Course Arrangement

- 7. CNNs -> Transformers (2)
- 8. Vision Tasks and Learning (2)
- 9. Generative Adversarial Networks (2)
- 10. Self-Supervised Learning (2)
- 11. Vision-Language Pretraining (2)
- 12. Learning with Low-Resource Data (2)

Content

- Introduction to Computer Vision
- Traditional Neural Networks and Their Limitations
- CNN and Recent Progress
  - CNN
  - Rectified Linear Units (ReLU)
  - Dropout / Batch Normalization
  - Representative Network Architectures
- Vision Transformers and Recent Progress

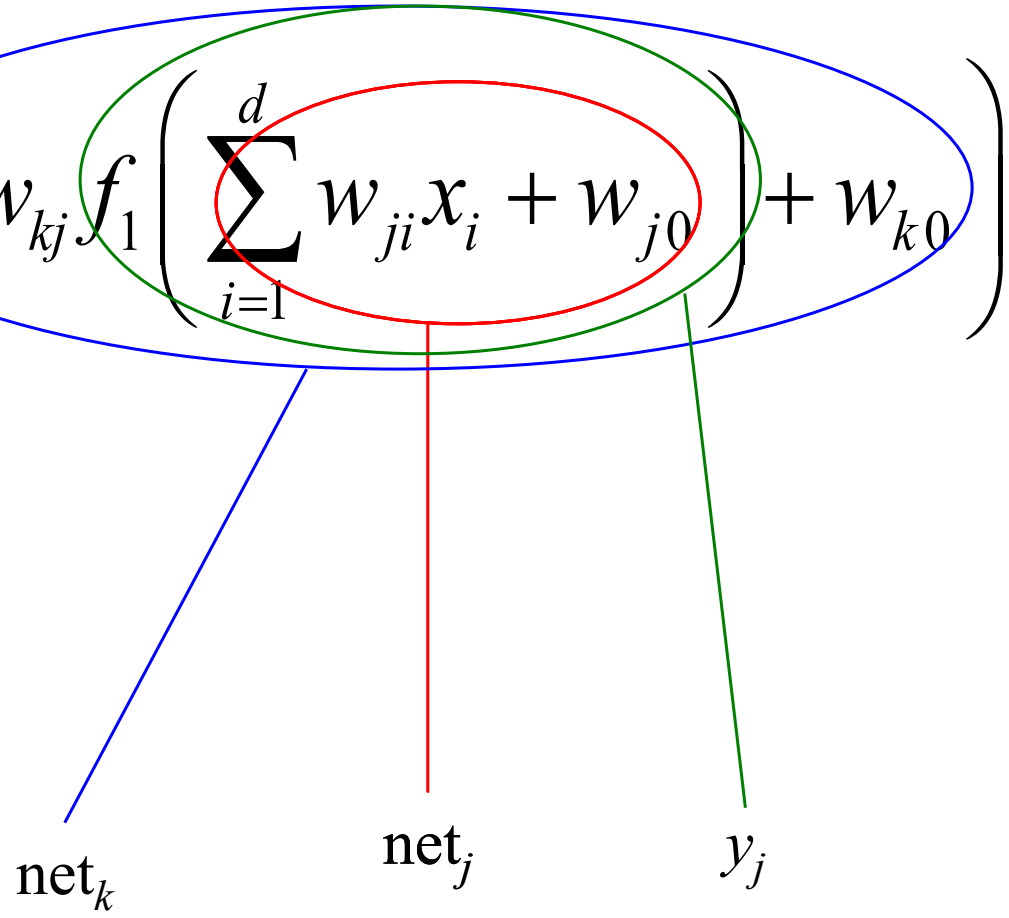
Multiple Layer Perception



D. E. Rumelhart, G. E. Hinton & R. J. Williams, Learning representations by back-propagating errors, Nature 323, 533 - 536 (09 October 1986)

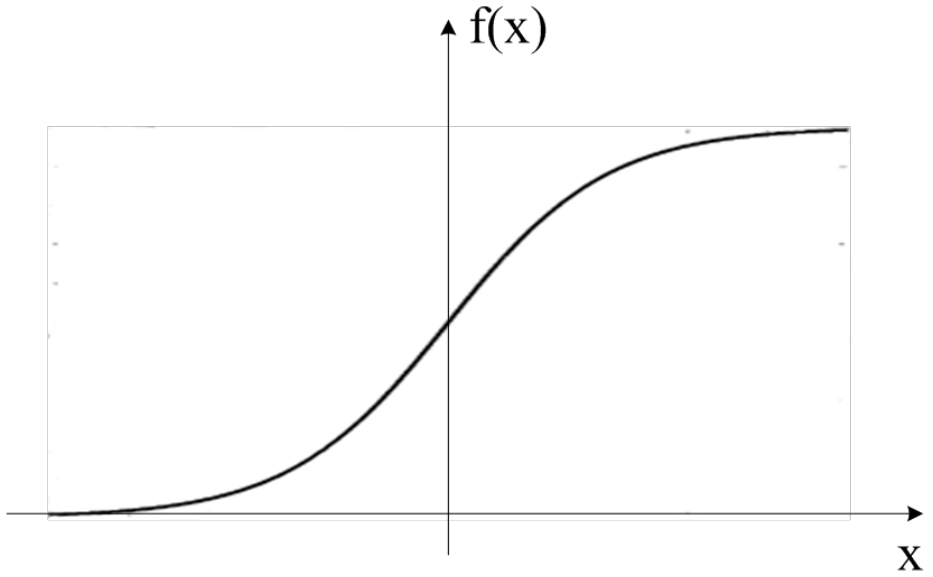
Back-Propagation

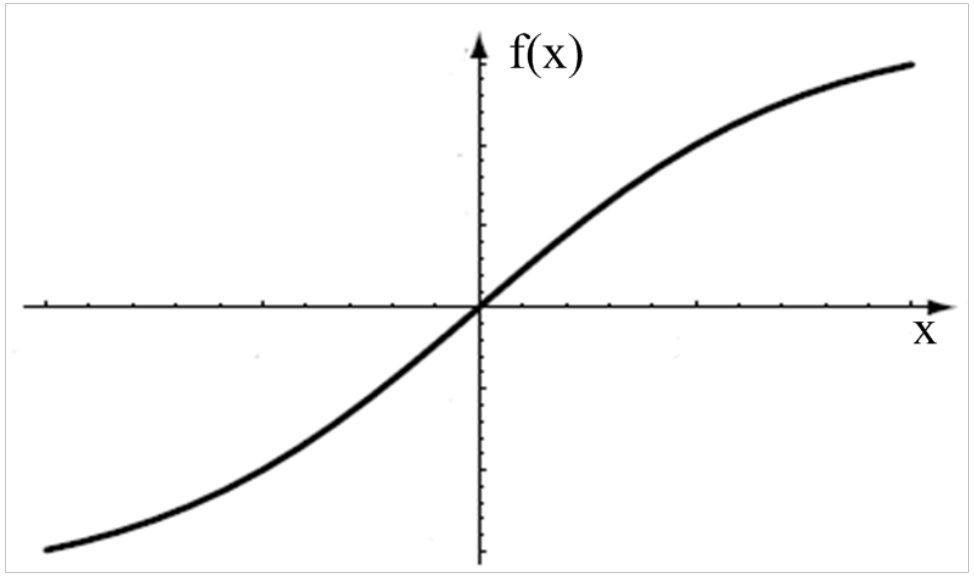
$$g_k(\mathbf{x}) = f_2\left(\sum_{j=1}^{n_H} w_{kj} f_1\left(\sum_{i=1}^{d} w_{ji} x_i + w_{j0}\right) + w_{k0}\right)$$



Activation Functions

- Gradient exploding/vanishing





Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$

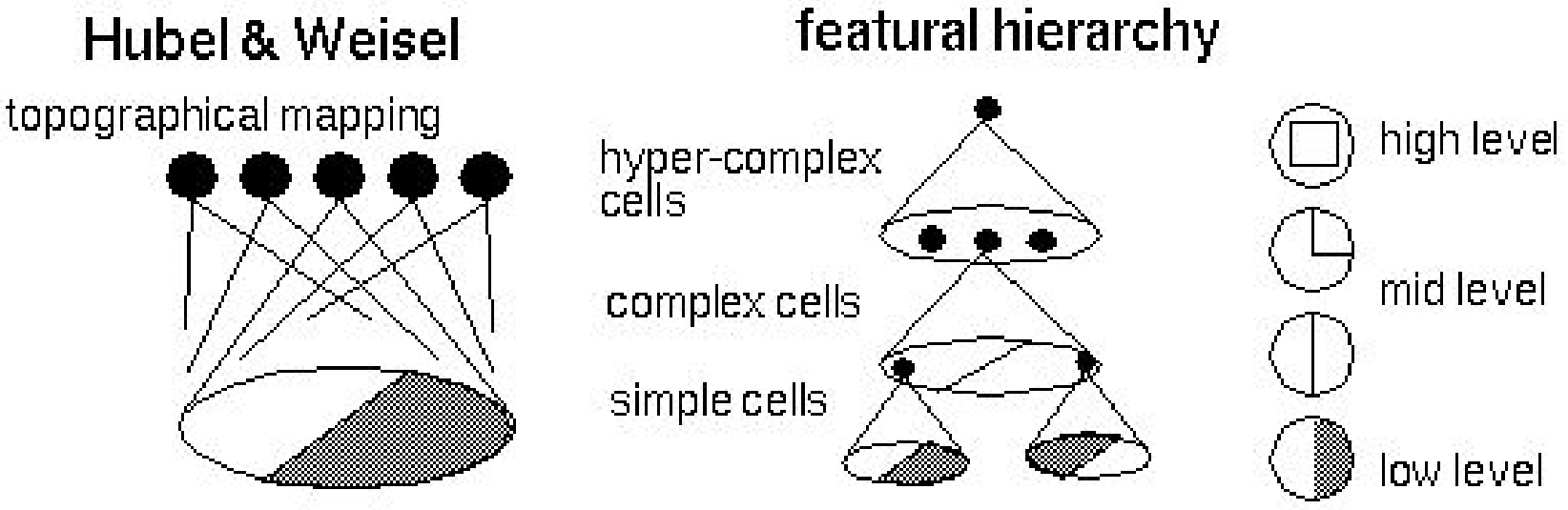
tanh: $f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$

Content

- Introduction to Computer Vision
- Traditional Neural Networks and Their Limitations
- CNN and Recent Progress
  - CNN
  - Rectified Linear Units (ReLU)
  - Dropout / Batch Normalization
  - Representative Network Architectures
- Vision Transformers and Recent Progress

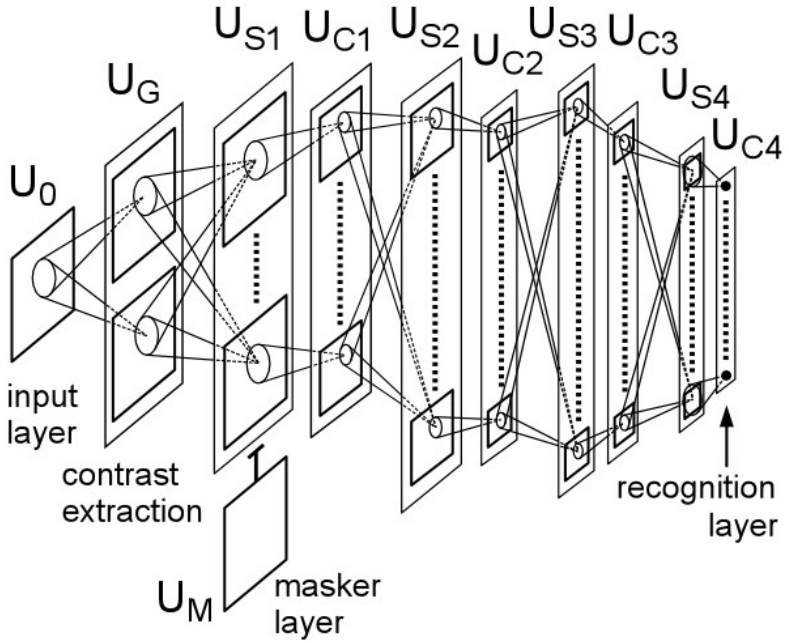
Hubel/Wiesel Architecture

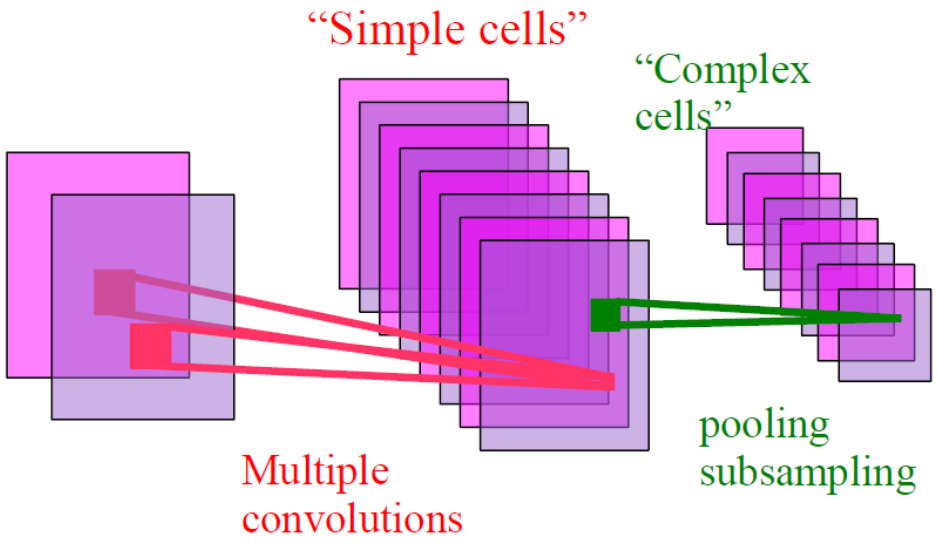
- D. Hubel and T. Wiesel (1959, 1962, Nobel Prize 1981)
  - Visual cortex consists of a hierarchy of *simple*, *complex*, and *hyper-complex* cells



Early Hierarchical Feature Models for Vision

- **[Hubel & Wiesel 1962]:**
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.

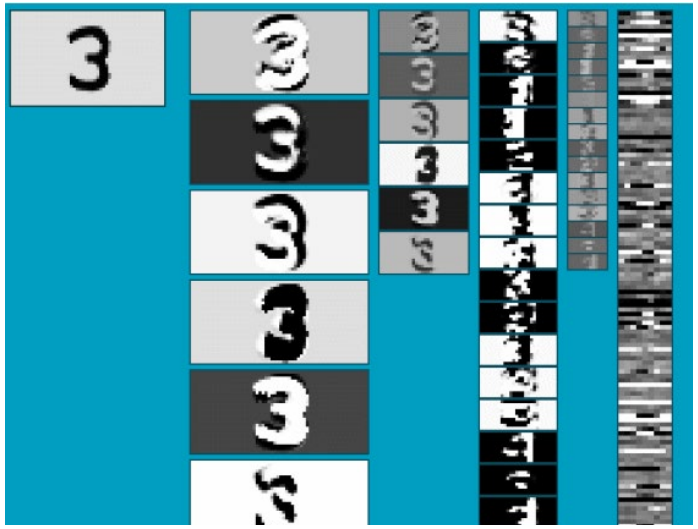


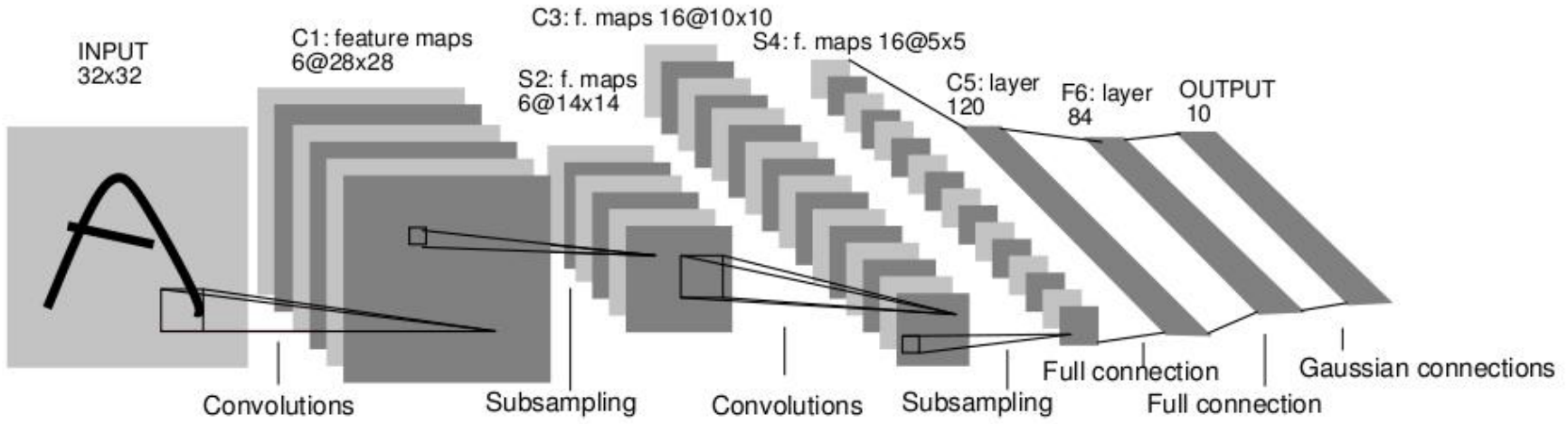


Cognitron & Neocognitron [Fukushima 1974-1982]

Convolutional Neural Networks (CNN, Convnet)

- Neural network with specialized connectivity structure
- Stacking multiple stages of feature extractors
- Higher stages compute more global, more invariant features
- Classification layer at the end

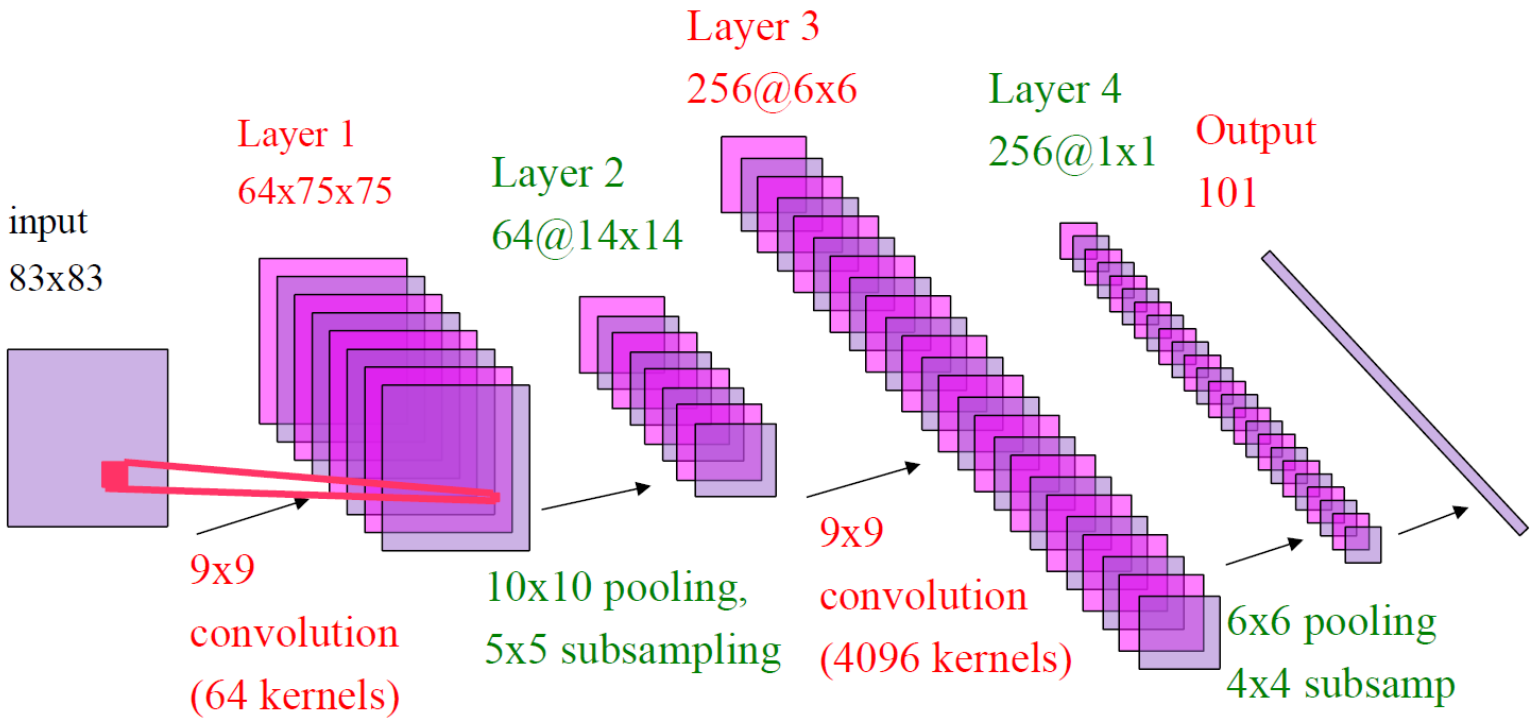




Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11): 2278–2324, 1998.

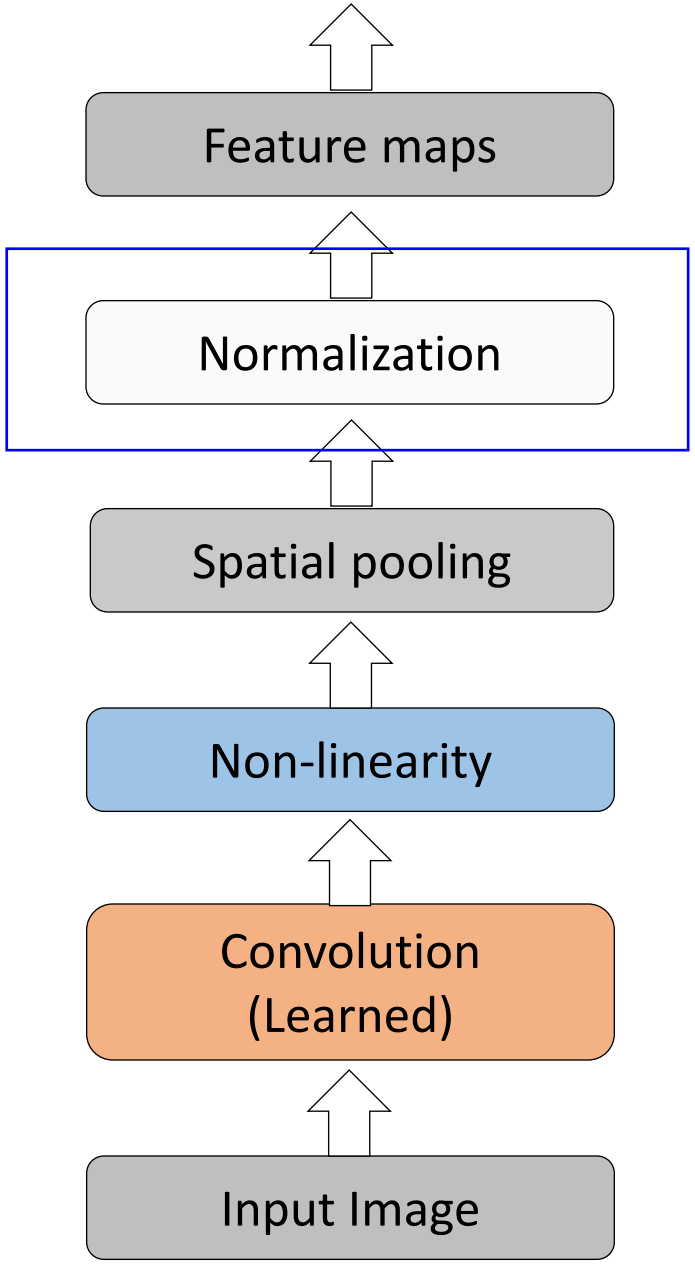
Convolutional Neural Network (LeCun)

- LeCun et al., NIPS 1989



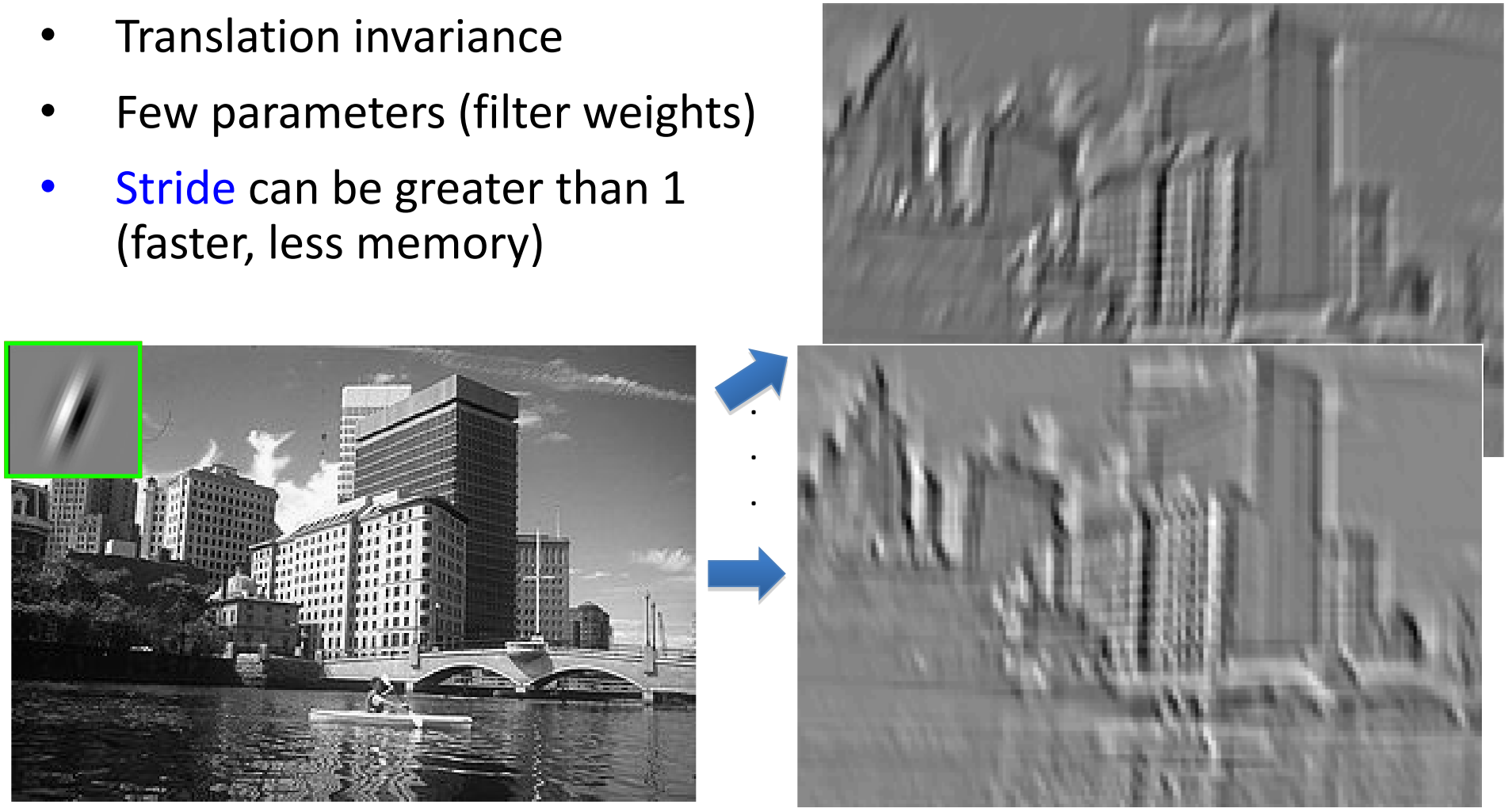
Convolutional Neural Networks (CNN, Convnet)

- Feed-forward feature extraction:
  1. Convolve input with learned filters
  2. Non-linearity
  3. Spatial pooling
  4. Normalization
- Supervised training of convolutional filters by back-propagating classification error



1. Convolution

- Dependencies are local
- Translation invariance
- Few parameters (filter weights)
- Stride can be greater than 1 (faster, less memory)

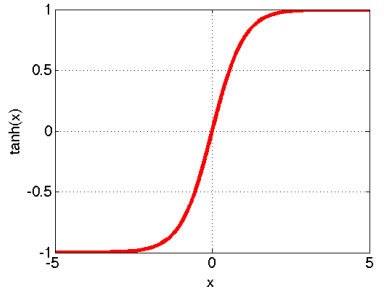


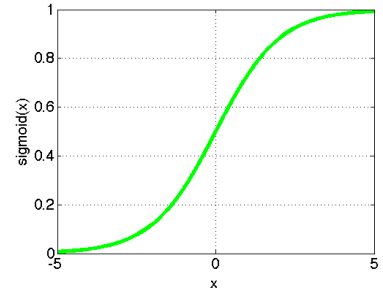
Input

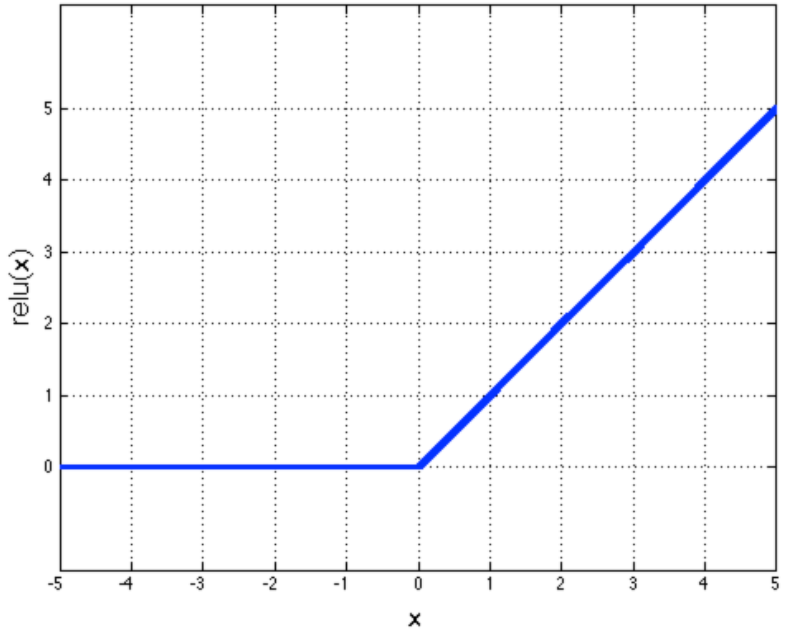
Feature Map

2. Non-Linearity

- Per-element (independent)
- Options:
  - Tanh
  - Sigmoid: 1/(1+exp(-x))
  - Rectified linear unit (ReLU)
    - Simplifies back-propagation
    - Makes learning faster
    - Avoids saturation issues
      → Preferred option
  - More adaptability
    - Parametric ReLU
    - Weighted combination of RBFs

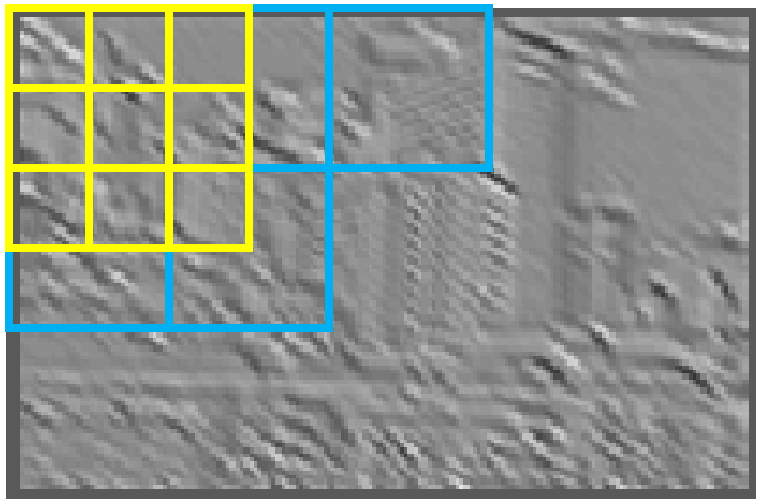




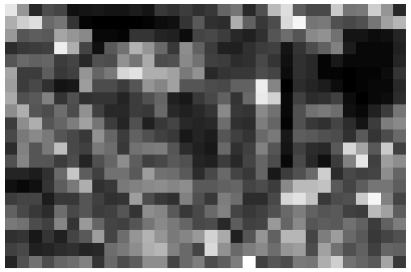


3. Spatial Pooling

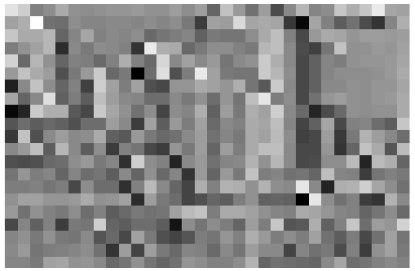
- Average or max
- Non-overlapping / overlapping regions
- Role of pooling:
  - Invariance to small transformations
  - Larger receptive fields (see more of input)



Max

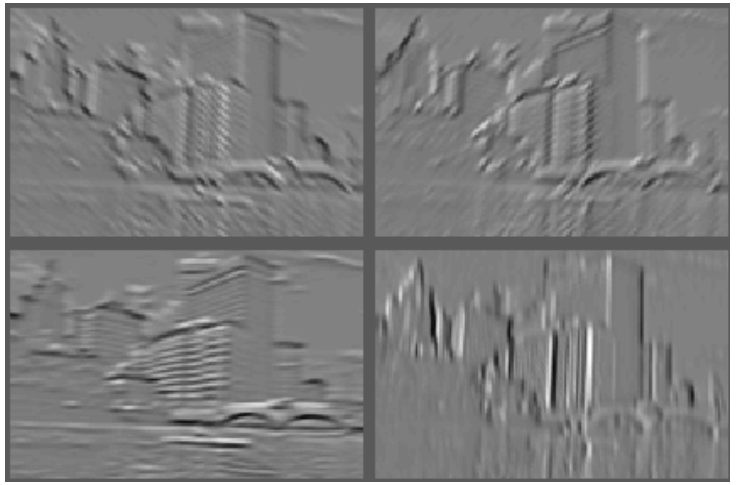


Sum

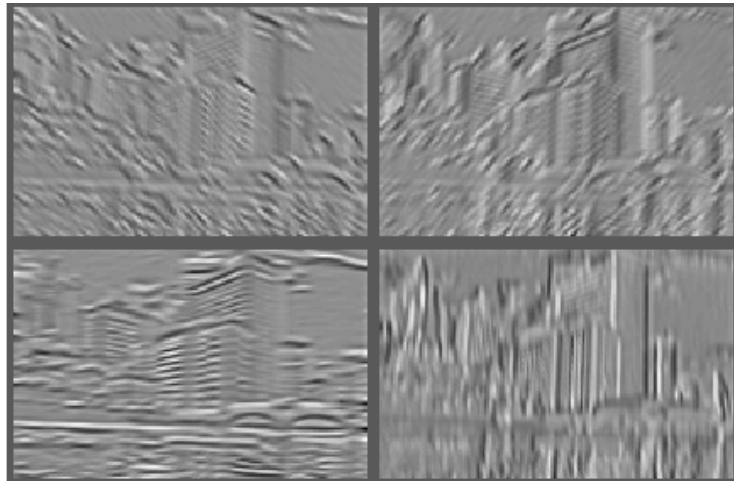


4. Normalization

- Within or across feature maps
- Before or after spatial pooling

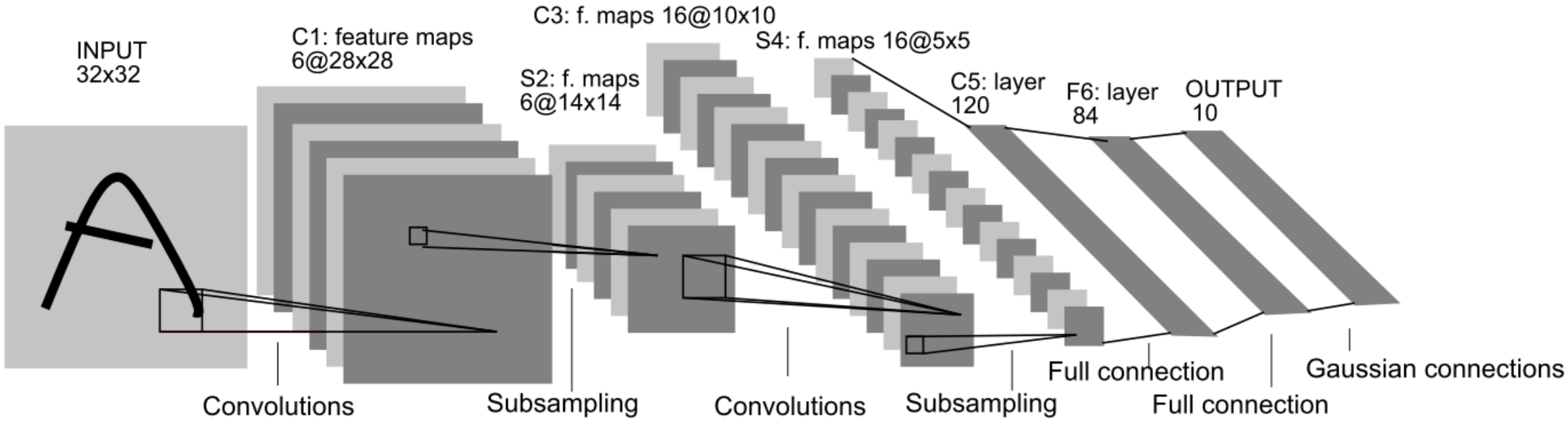


Feature Maps



Feature Maps
After Contrast Normalization

LeNet 5, LeCun 1998



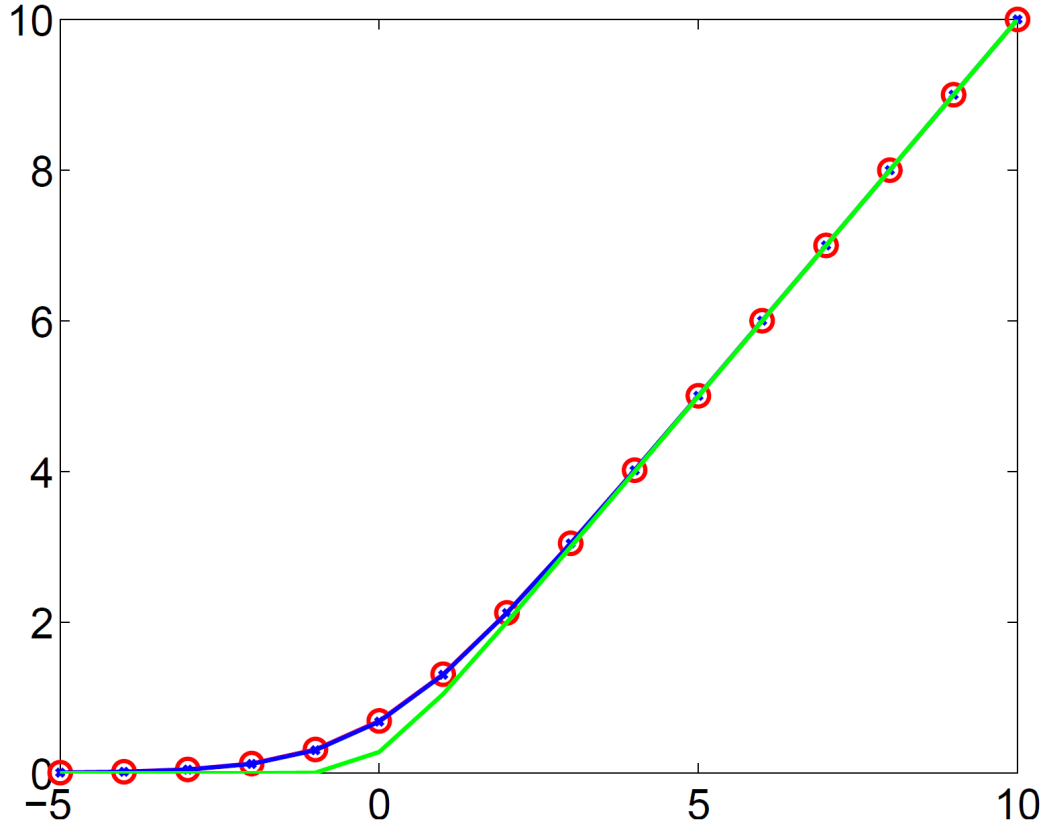
- Input: 32x32 pixel image. Largest character is 20x20
- Cx: Convolutional layer
- Sx: Subsample layer
- Fx: Fully connected layer
- Black and White pixel values are normalized

Content

- Traditional Neural Networks and Their Limitations
- CNN and Recent Progress
  - CNN
  - Rectified Linear Units (ReLU)
  - Dropout / Batch Normalization
  - Representative Network Architectures
- Vision Transformers and Recent Progress

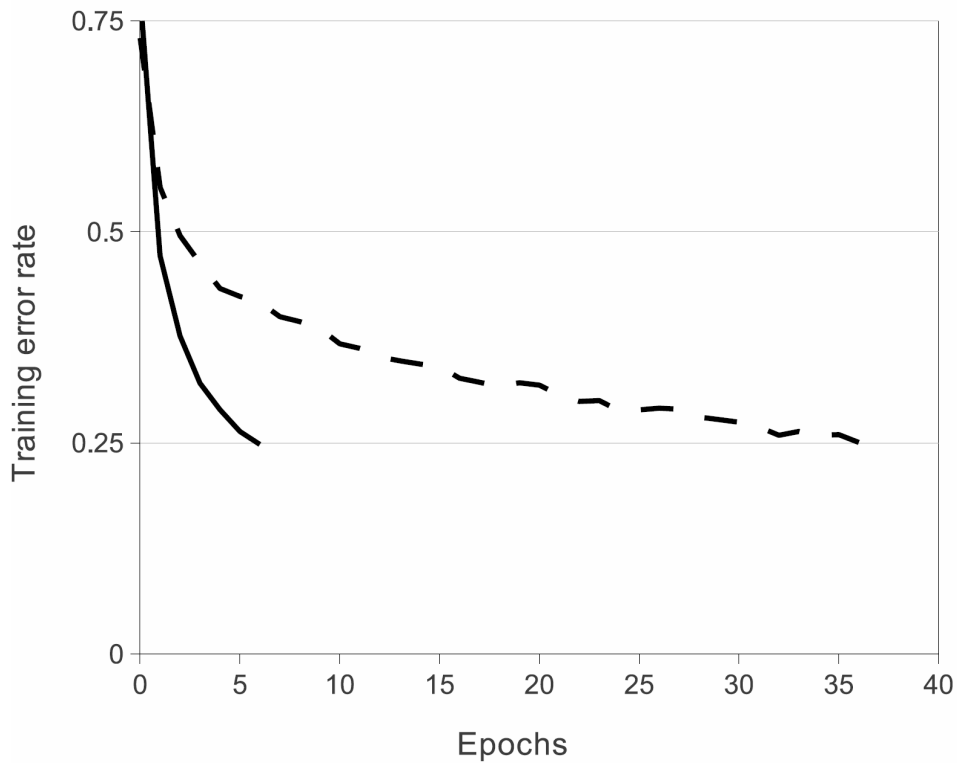
ReLU (Rectified Linear Units)

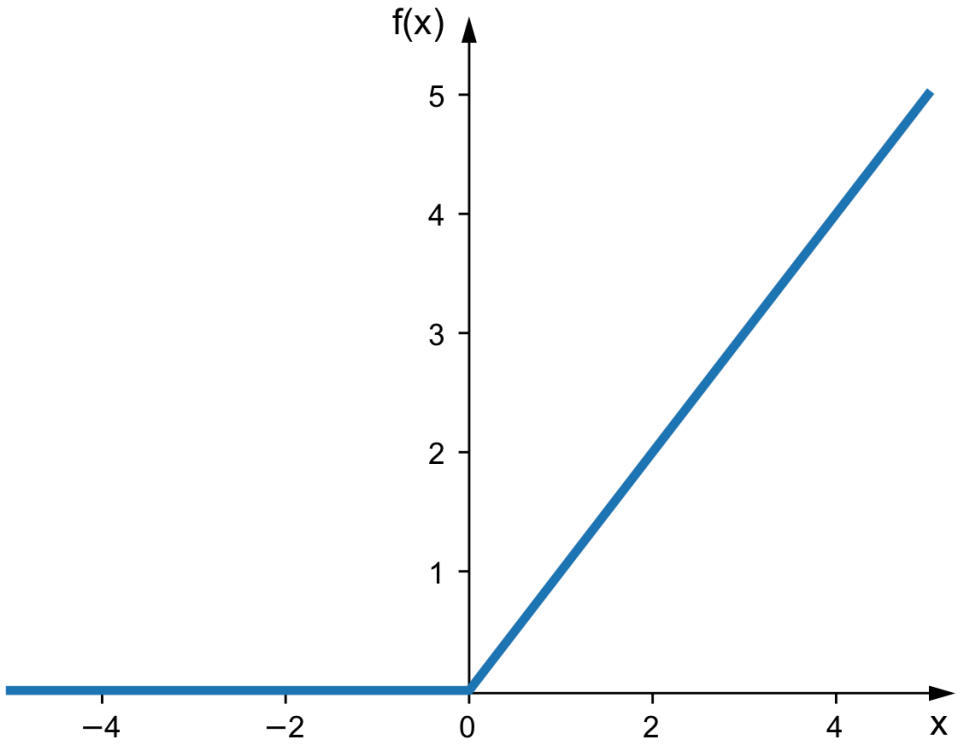
$$\sum_{i=1}^{N} \sigma(x - i + 0.5) \approx \log(1 + e^x)$$



Vinod Nair, Geoffrey E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, ICML 2010.

ReLU (Rectified Linear Units)

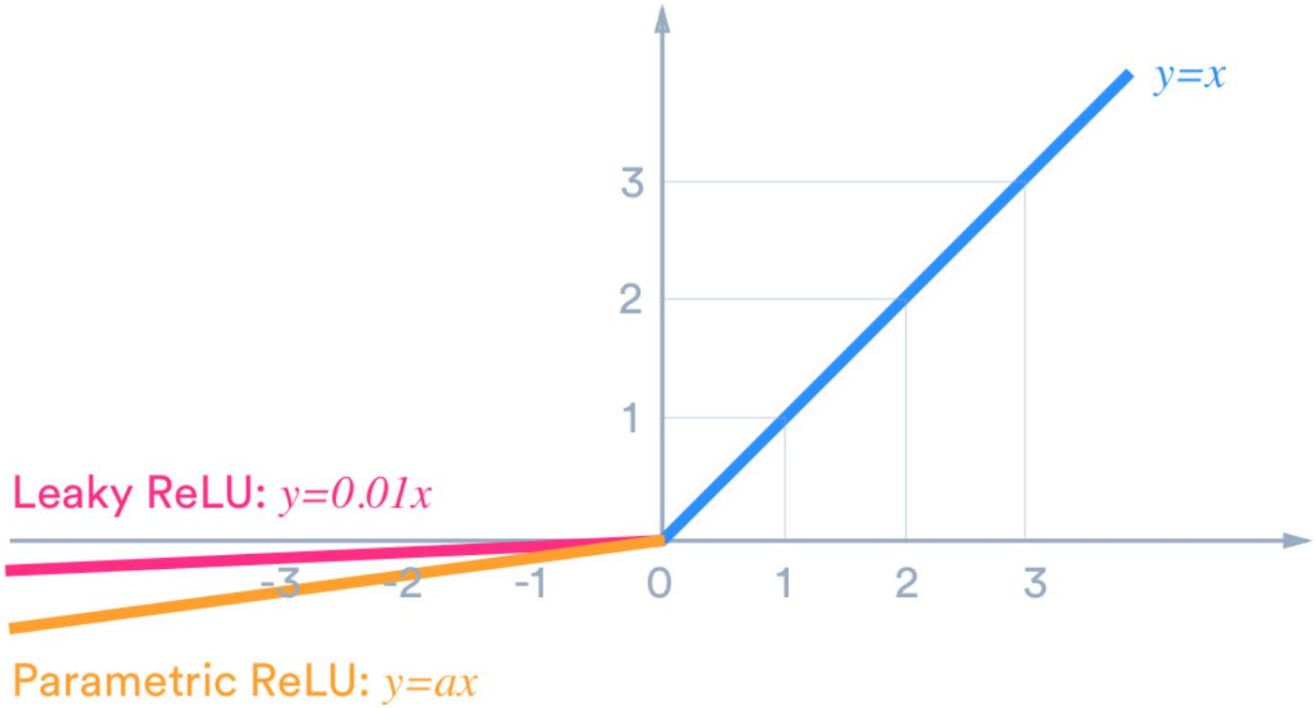




A Krizhevsky, I Sutskever, GE Hinton, Imagenet classification with deep convolutional neural networks, NIPS 2012

ReLU Variants

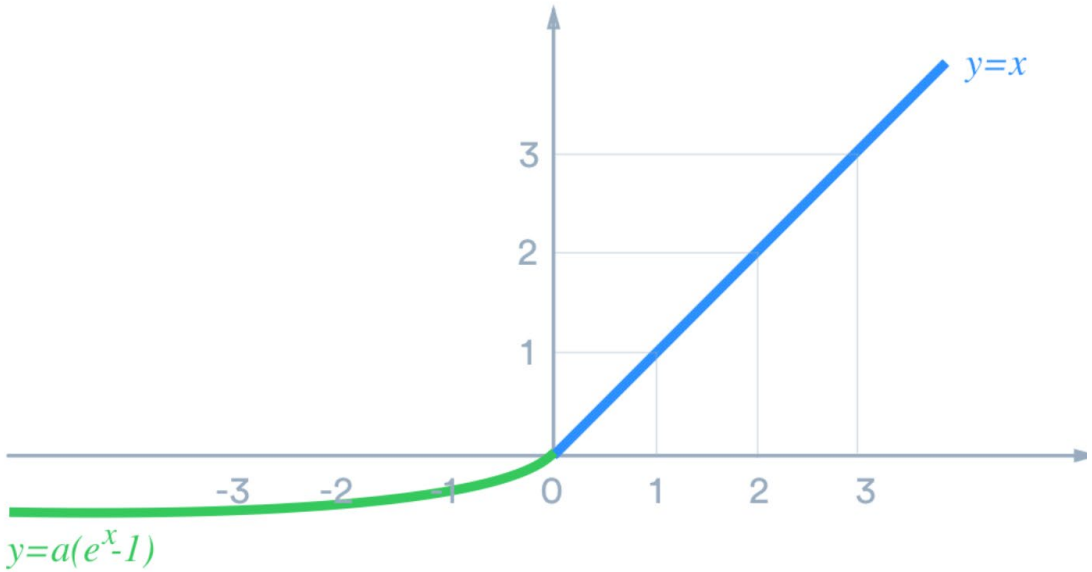
- Leaky ReLU & PReLU



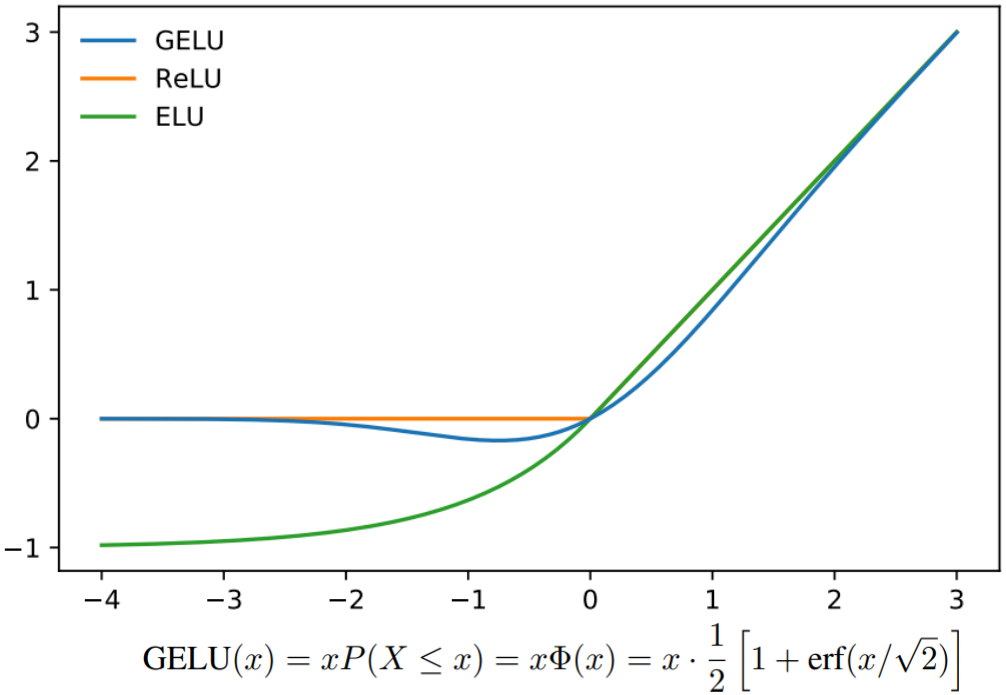
K He, X Zhang, S Ren, J Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, ICCV 2015

ReLU Variants

- ELU: Exponential Linear Units



- GELU



D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), ICLR 2016

Content

- Traditional Neural Networks and Their Limitations
- CNN and Recent Progress
  - CNN
  - Rectified Linear Units (ReLU)
  - Dropout / Batch Normalization
  - Representative Network Architectures
- Vision Transformers and Recent Progress

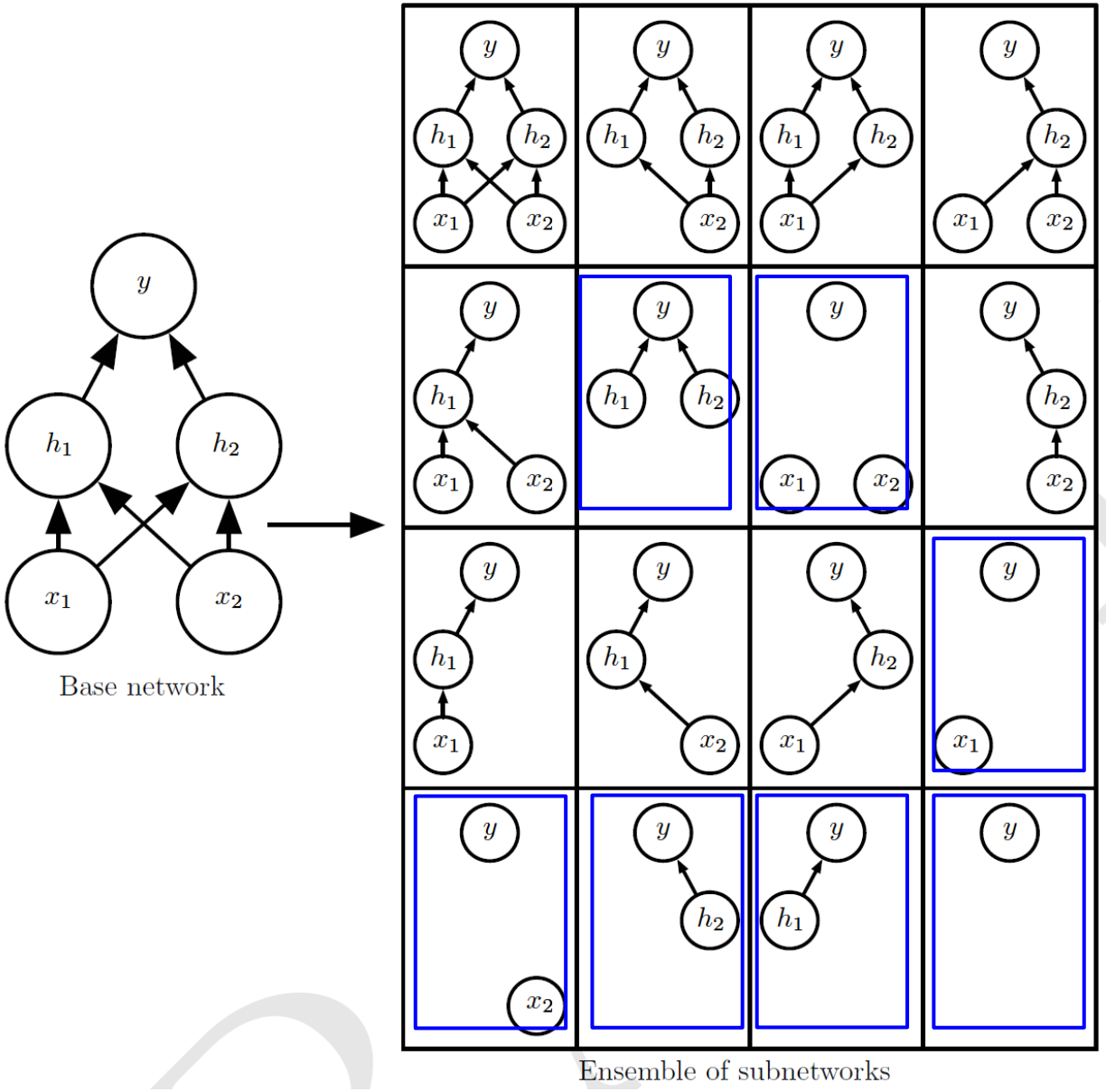
Regularization

- Principled:
  - Bayes: Likelihood + Prior
  - Regularization

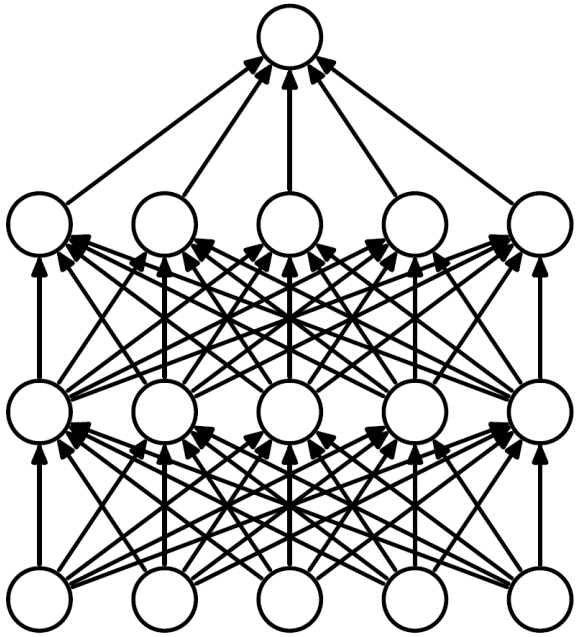
- Engineered:
  - Early stopping
  - Model averaging
  - Mini-Batch
  - …

Christopher M. Bishop, Neural Networks for Pattern Recognition, Oxford, 1995

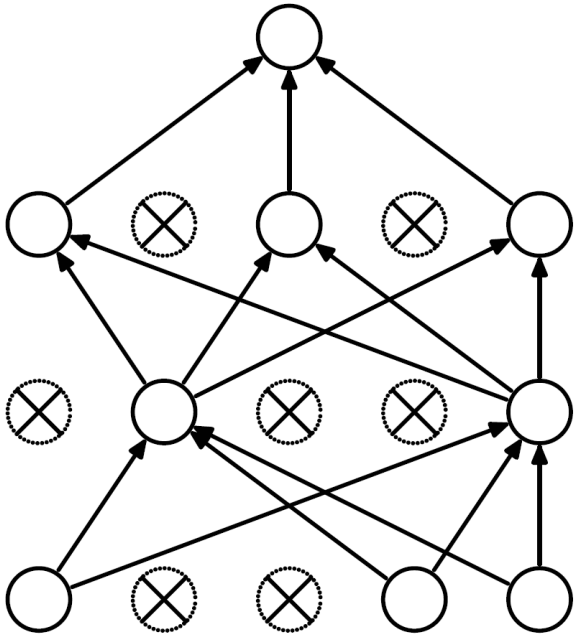
Dropout



Dropout训练



(a) Standard Neural Net



(b) After applying dropout.

- 在小批量中加载一个样本，然后随机抽样应用于网络中所有输入和隐藏单元的不同二值掩码
- 超参数：掩码值为1 的采样概率

Dropout训练

$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)}\mathbf{y}^l + b_i^{(l+1)}$$
$$y_i^{(l+1)} = f(z_i^{(l+1)})$$

Dropout

$$r_j^{(l)} \sim \text{Bernoulli}\ (p)$$
$$\tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} * \mathbf{y}^{(l)}$$
$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)}\mathbf{y}^l + b_i^{(l+1)}$$
$$y_i^{(l+1)} = f(z_i^{(l+1)})$$

Dropout

- 训练阶段所有模型共享参数，测试阶段直接组装成一个整体的大网络
- 有效避免过拟合
- 可用于前馈神经网络、概率模型，如受限玻尔兹曼机，以及循环神经网络等
- 会需要较多的迭代次数和训练时间
- 理论解释

Batch Normalization (Ioffe & Szegedy, Arxiv 2015)

- Internal covariate shift: the change in the distributions of internal nodes of a deep network, in the course of training
- Batch normalization: Fix the means and variances of layer inputs
- Batch Normalizing Transform

$$\mathrm{z} = g(\mathrm{BN}(W\mathrm{u}))$$

Recover the original activations

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Batch Normalization

- Gradient for Back-Propagation

$$\frac{\partial \ell}{\partial \widehat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \gamma$$

$$\frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2}(\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2}$$

$$\frac{\partial \ell}{\partial \mu_{\mathcal{B}}} = \left(\sum_{i=1}^{m} \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \right) + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_{\mathcal{B}})}{m}$$

$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}$$

$$\frac{\partial \ell}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i} \cdot \widehat{x}_i$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i}$$

- Advantages:
  - Regularization
  - Higher learning rate
  - Internal covariate shift
  - Remove dropout
  - Remove L2 regularizer
- Note: not always work

Going beyond (1)

- Batch Normalization on Feature Maps

- Orthogonal regularization on convolution filters

$$\|W^{\mathrm{T}}W - I\|_F^2$$

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift.
In *ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015.
K. Jia, D. Tao, S. Gao, and X. Xu, Improving training of deep neural networks via Singular Value Bounding, CVPR 2017

Going beyond (2)

- Weight normalization

$$\sigma_1(\bar{W}_{\mathrm{WN}})^2 + \sigma_2(\bar{W}_{\mathrm{WN}})^2 + \cdots + \sigma_T(\bar{W}_{\mathrm{WN}})^2 = d_o$$

- Spectral normalization

$$\bar{W}_{\mathrm{SN}}(W) := W/\sigma(W)$$

T. Salimans and D.P. Kingma. Weight normalization: A simple reparameterization to accelerate training
of deep neural networks. In NIPS, pp. 901–909, 2016.
T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, ICLR 2018

Content

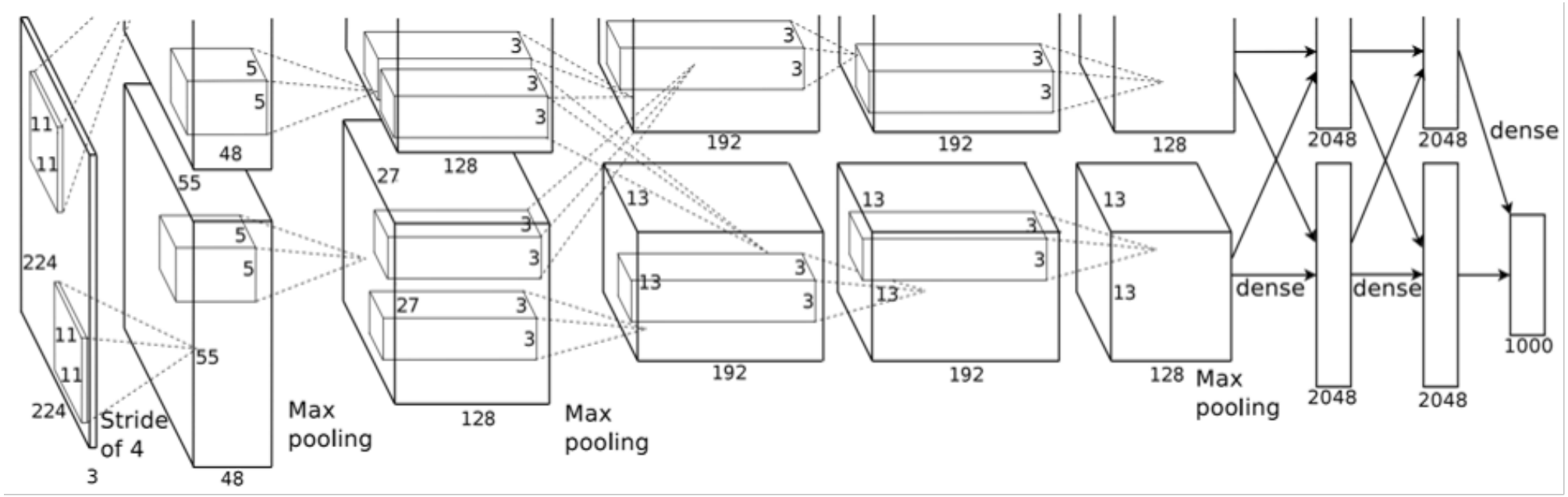
- Traditional Neural Networks and Their Limitations
- CNN and Recent Progress
  - CNN
  - Rectified Linear Units (ReLU)
  - Dropout / Batch Normalization
  - Representative Network Architectures
- Vision Transformers and Recent Progress

Representative Deep CNNs

- AlexNet (Krizhevsky et al. NIPS 2012)
- ~~ZeilerNet (Zeiler & Fergus, ECCV 2014)~~
- VGGNet (Simonyan & Zisserman, ICLR 2015)
- GoogLeNet (Szegedy et al., CVPR 2015)
- U-Net (MICCAI, 2015)
- Deep Residual Network (He et al., CVPR 2016)
- DenseNet
- SENet

Convolutional Neural Network (Alex)

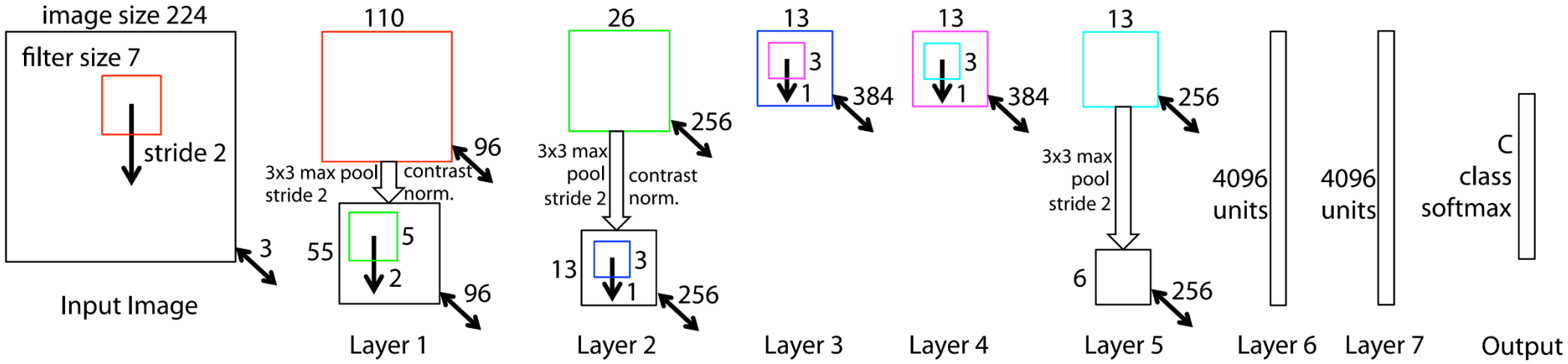
- Krizhevsky et al. NIPS 2012



- Five convolutional and three fully-connected layers
- 60 million parameters
- +ReLU+Dropout
- Group Conv.

ZeilerNet

- Remedy:
  - Reduced the 1st layer filter size from 11x11 to 7x7
  - Made the stride of the convolution 2, rather than 4.



VGGNet (Simonyan & Zisserman, ICLR 2015)

- VGG16 & VGG19
  - Small receptive field: 3×3
  - More layers
  - Max pooling with stride 2
  - +ReLU+Dropout

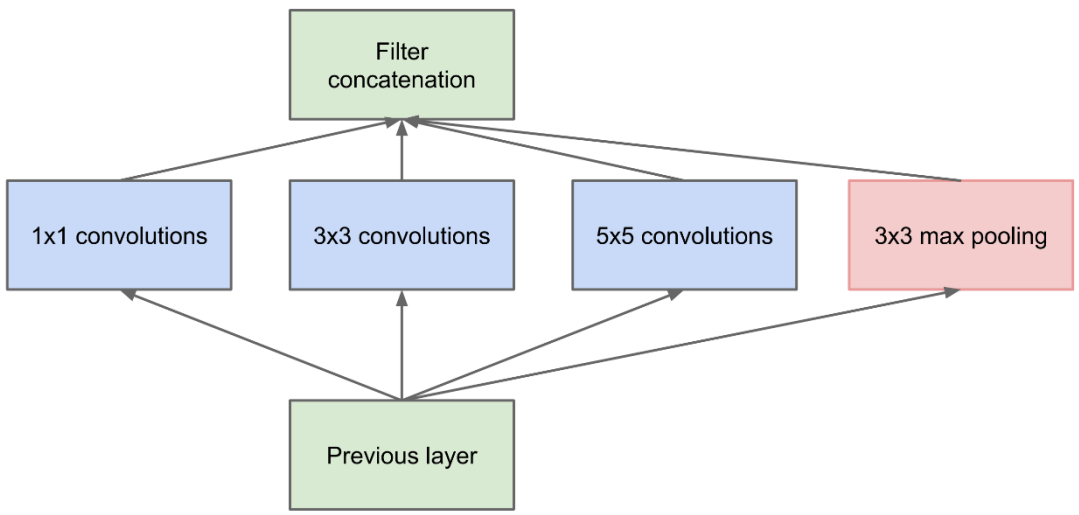
Table 2: **Number of parameters** (in millions).

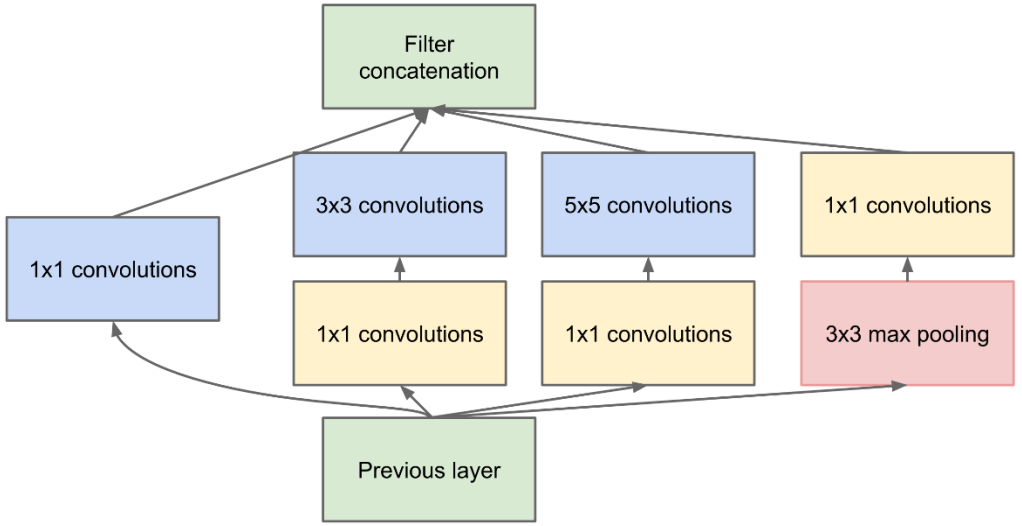
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

GoogLeNet (Szegedy et al., CVPR 2015)

- Inception: Basic structure



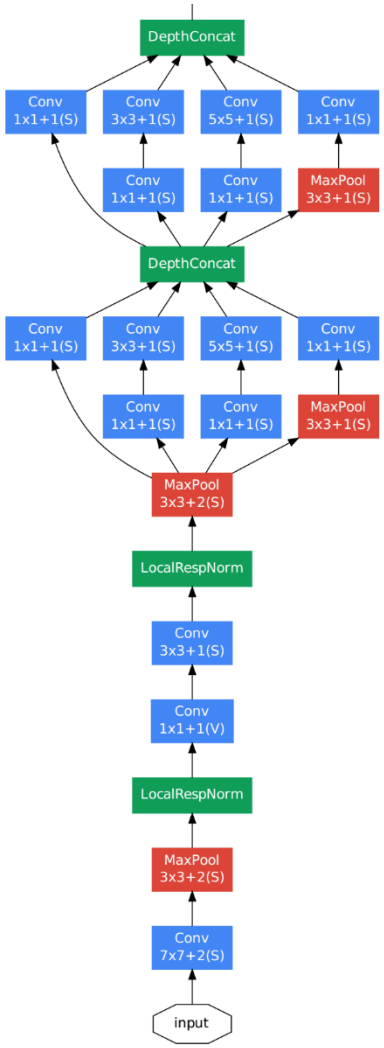


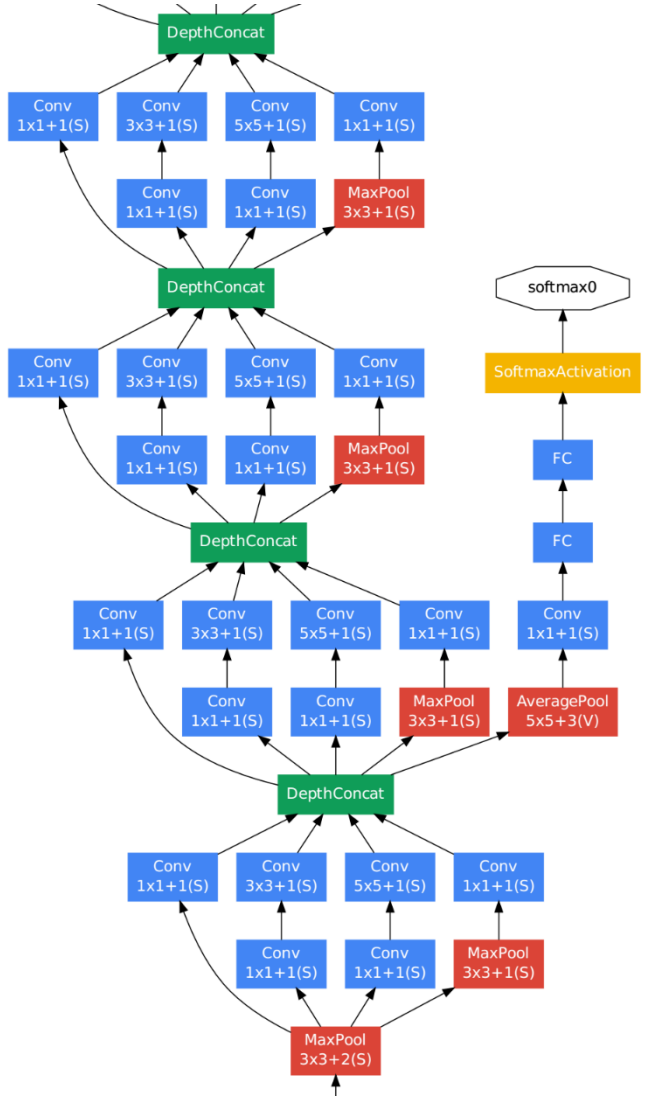
  - (Arora et al., Arxiv 2013): Layer-by layer construction to analyze the correlation statistics of the last layer and cluster them into groups of units with high correlation.

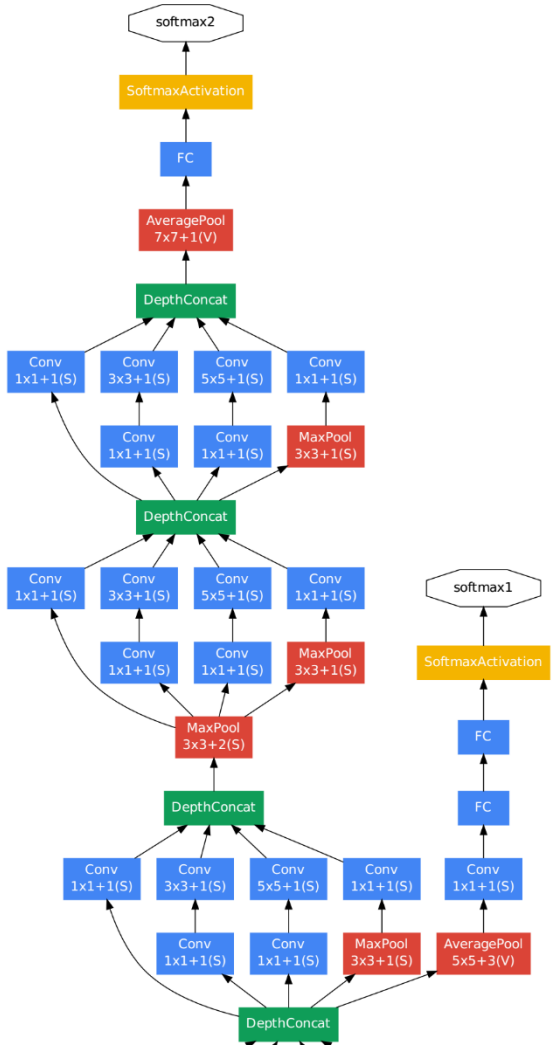
GoogLeNet (Szegedy et al., CVPR 2015)

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

GoogLeNet


input
Conv 7x7+2(S)
MaxPool 3x3+2(S)
Conv 1x1+1(V)
Conv 3x3+1(S)
LocalRespNorm
LocalRespNorm
MaxPool 3x3+2(S)
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat


softmax0
FC
FC
AveragePool 5x5+3(V)
SoftmaxActivation
DepthConcat
MaxPool 3x3+2(S)
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
MaxPool 3x3+1(S)
DepthConcat

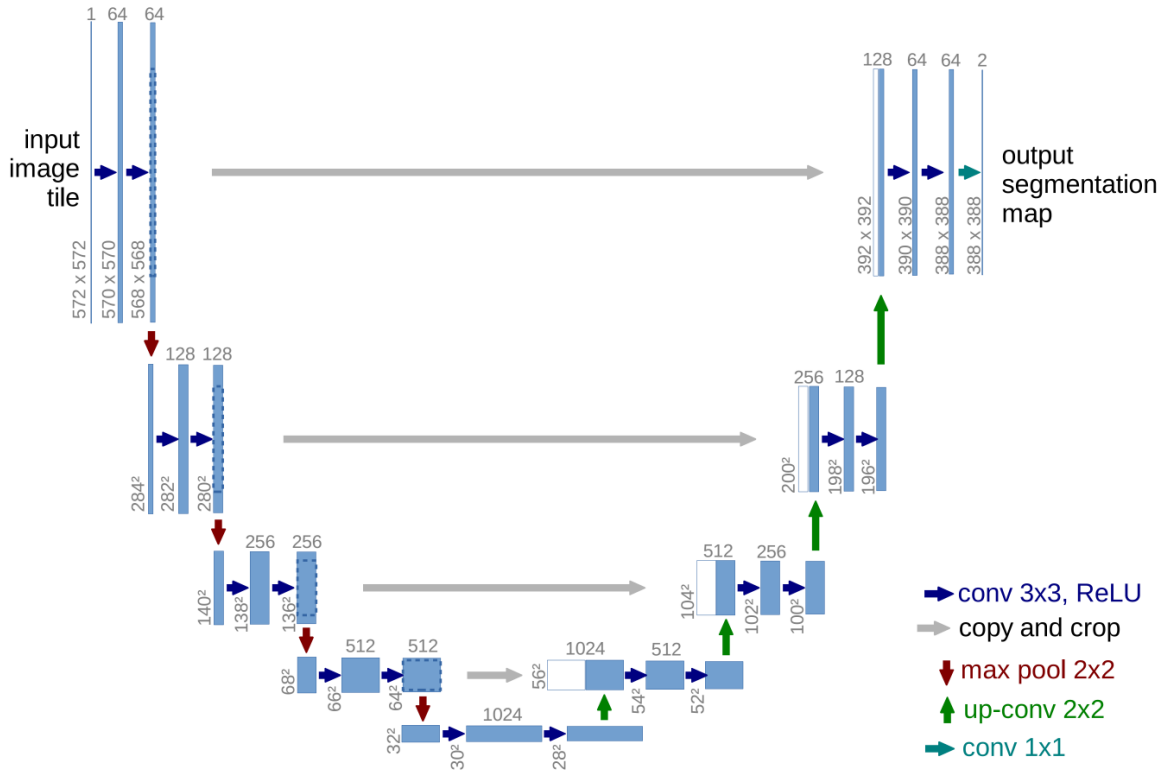

softmax1
SoftmaxActivation
FC
FC
AveragePool 5x5+3(V)
softmax2
SoftmaxActivation
FC
AveragePool 7x7+1(V)
DepthConcat
MaxPool 3x3+2(S)
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
MaxPool 3x3+1(S)
DepthConcat

Go deeper: ImageNet 2014

Team name	Entry description	**Classification error**	Localization error
GoogLeNet	No localization. Top5 val score is 6.66% error.	0.06656	0.606257
VGG	a combination of multiple ConvNets, including a net trained on images of different size (fusion weights learnt on the validation set); detected boxes were not updated	0.07325	0.256167
VGG	a combination of multiple ConvNets, including a net trained on images of different size (fusion done by averaging); detected boxes were not updated	0.07337	0.255431
VGG	a combination of multiple ConvNets (by averaging)	0.07405	0.253231
VGG	a combination of multiple ConvNets (fusion weights learnt on the validation set)	0.07407	0.253501
MSRA Visual Computing	Multiple SPP-nets further tuned on validation set (B)	0.0806	0.354924

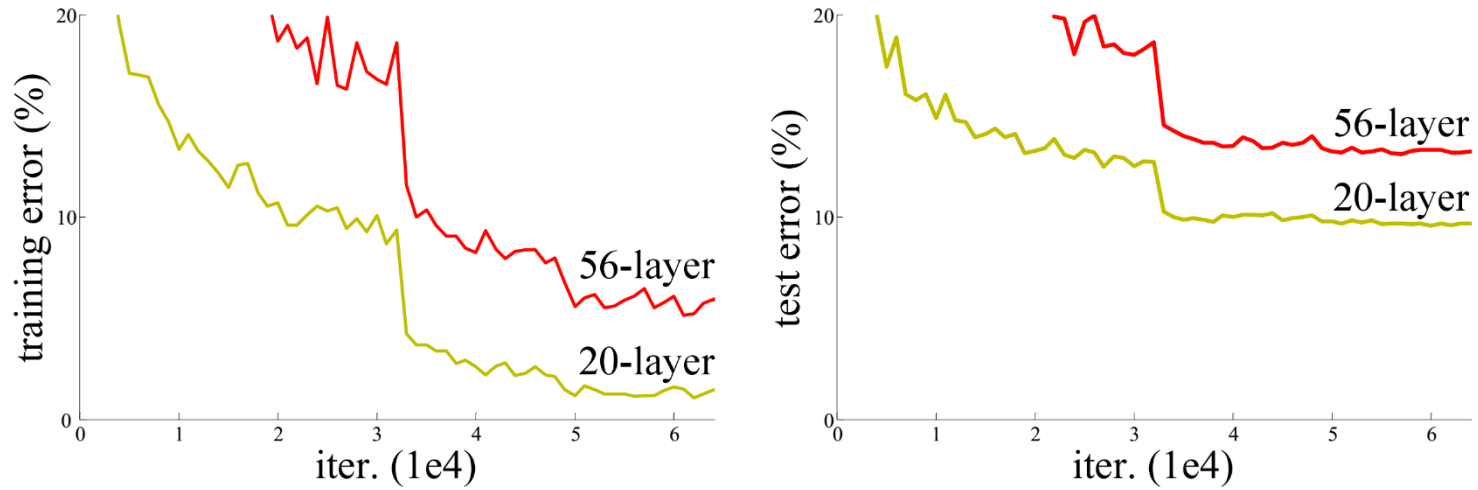
U-Net (Ronneberger et al., MICCAI 2015)

- Fine-details
- Skip connection



Residual Network (He et al., CVPR 2016)

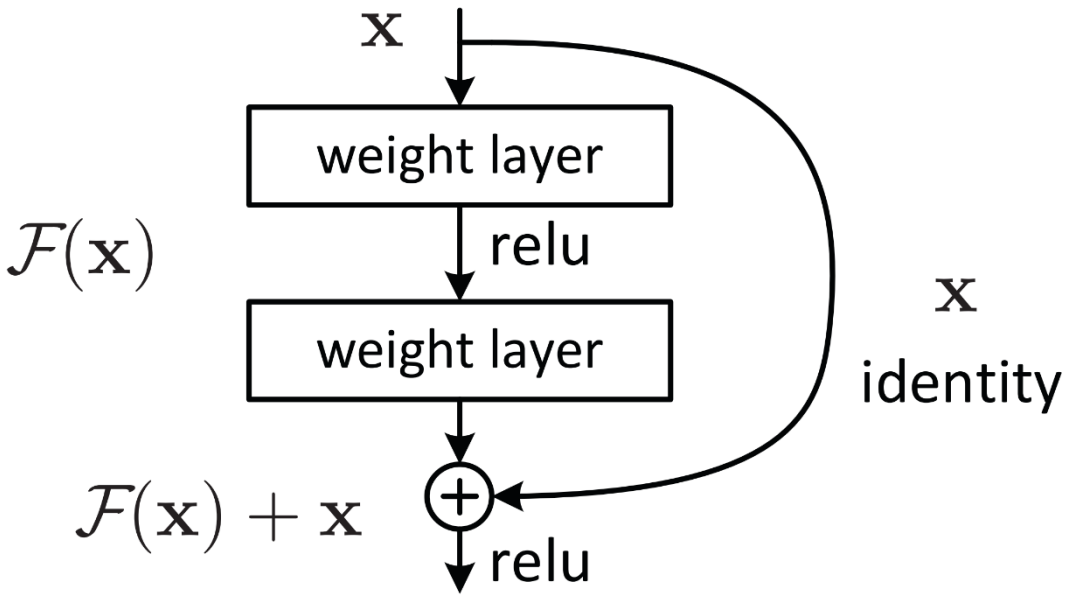
- Degradation
- How to increase depth without degradation?



Training error (left) and test error (right) on CIFAR-10

Residual Network (He et al., CVPR 2016)

- Desired mapping: $\mathcal{H}(\mathbf{x})$
- Residual mapping: $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$
- Residual learning: $\mathcal{F}(\mathbf{x}) + \mathbf{x}$



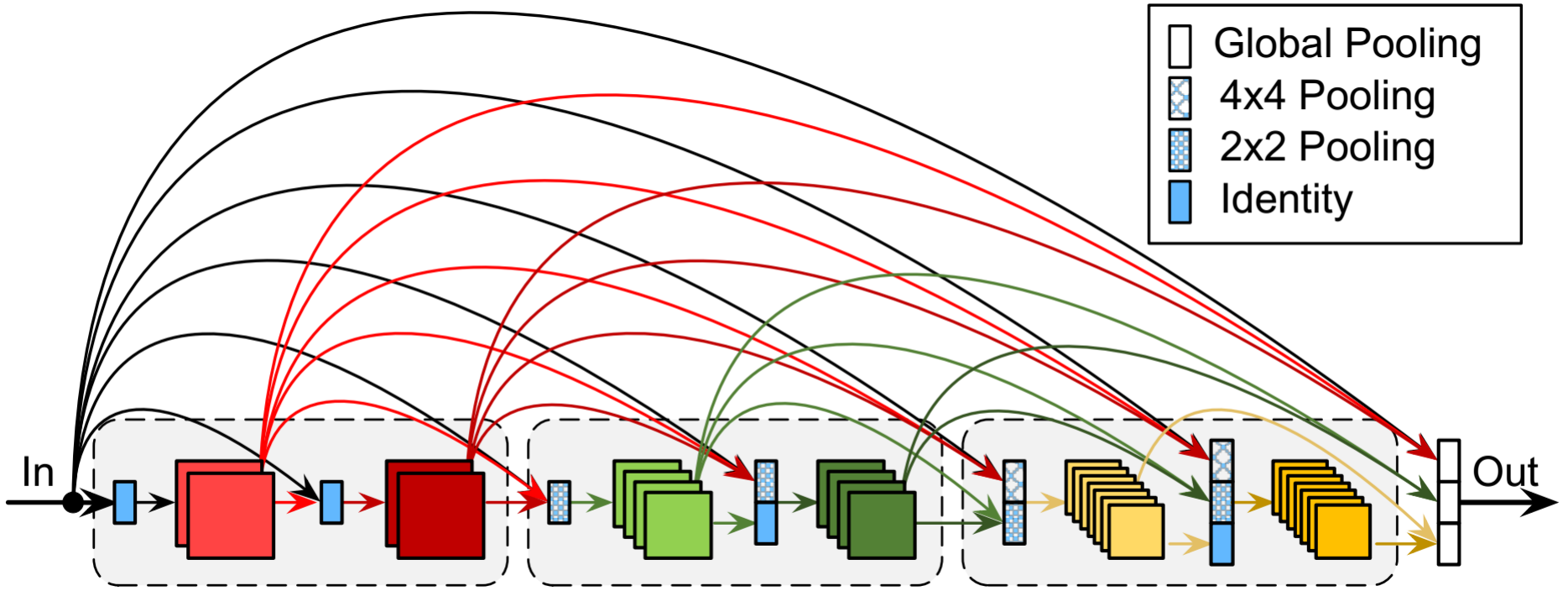
Residual learning: a building block.

Residual Network (He et al., CVPR 2016)

- Architecture

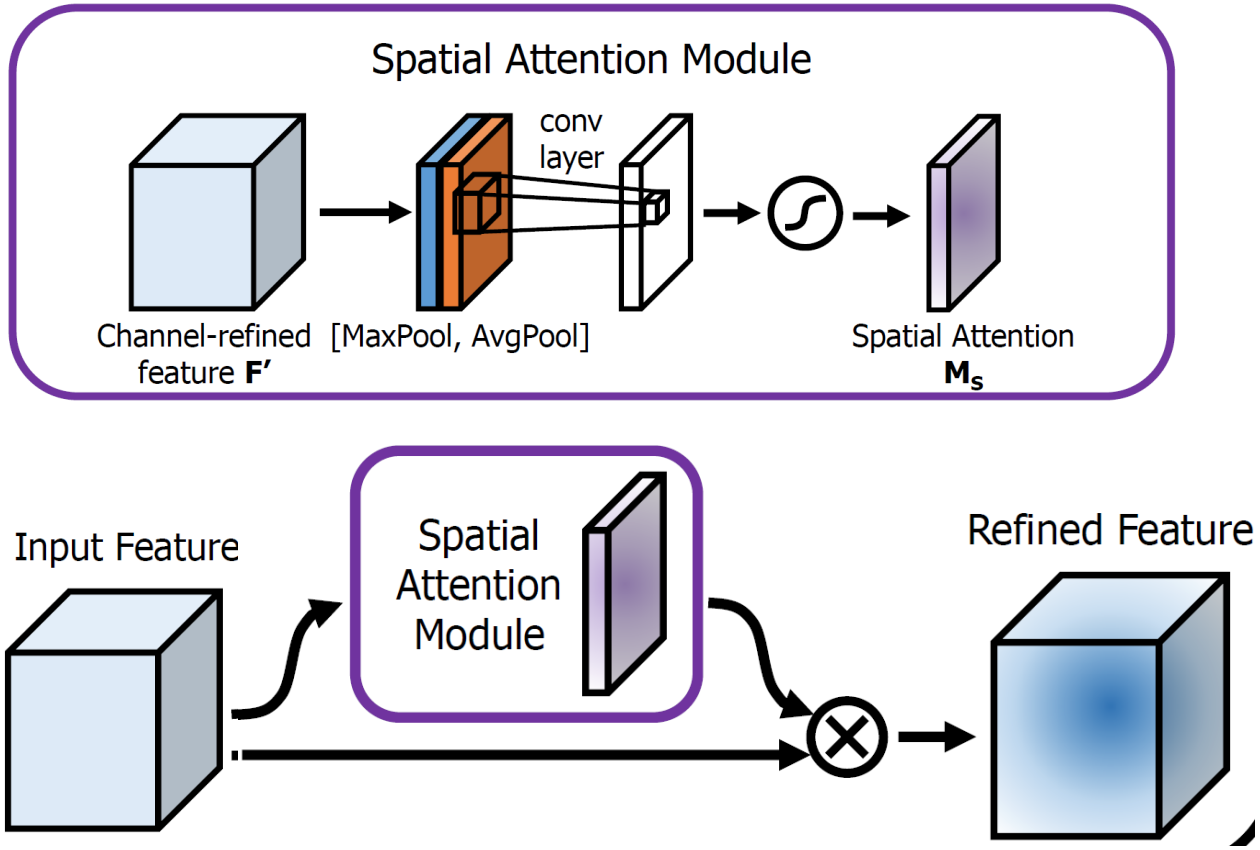
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

DenseNet (Huang et al., CVPR 2017)



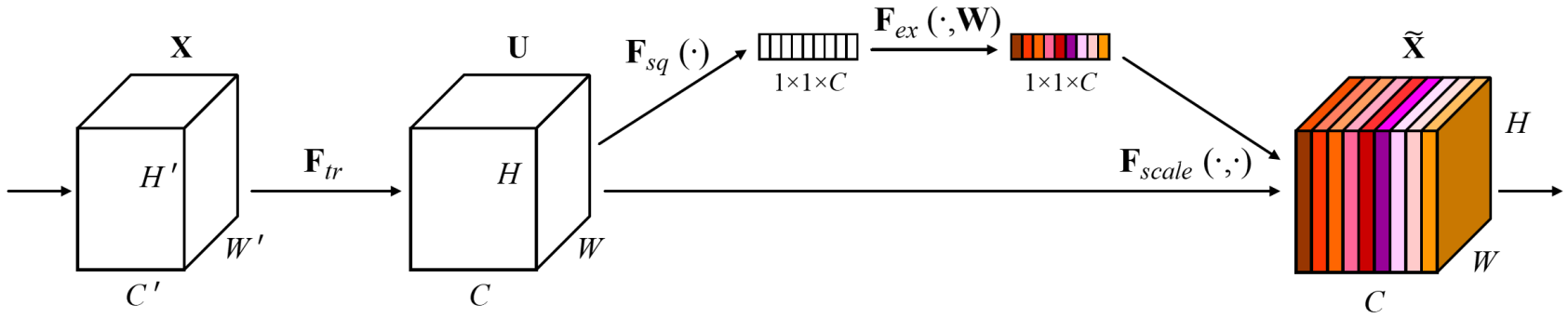
空域注意力模型

$$\alpha = \text{softmax}\left(\mathbf{W}_i\mathbf{a} + b_i\right)$$



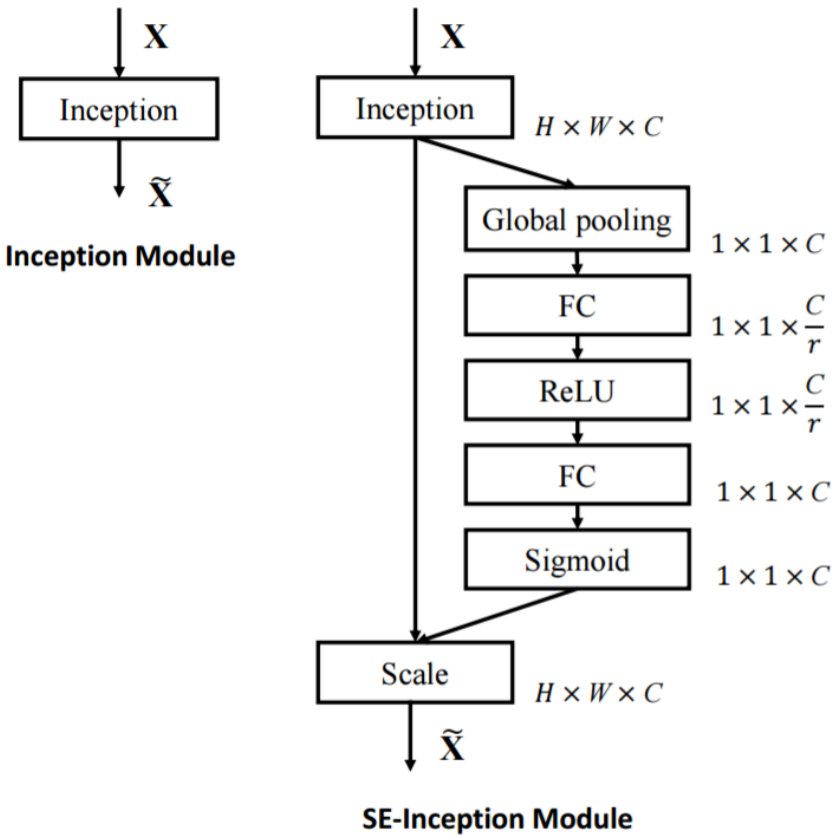
SENet：通道注意力模型

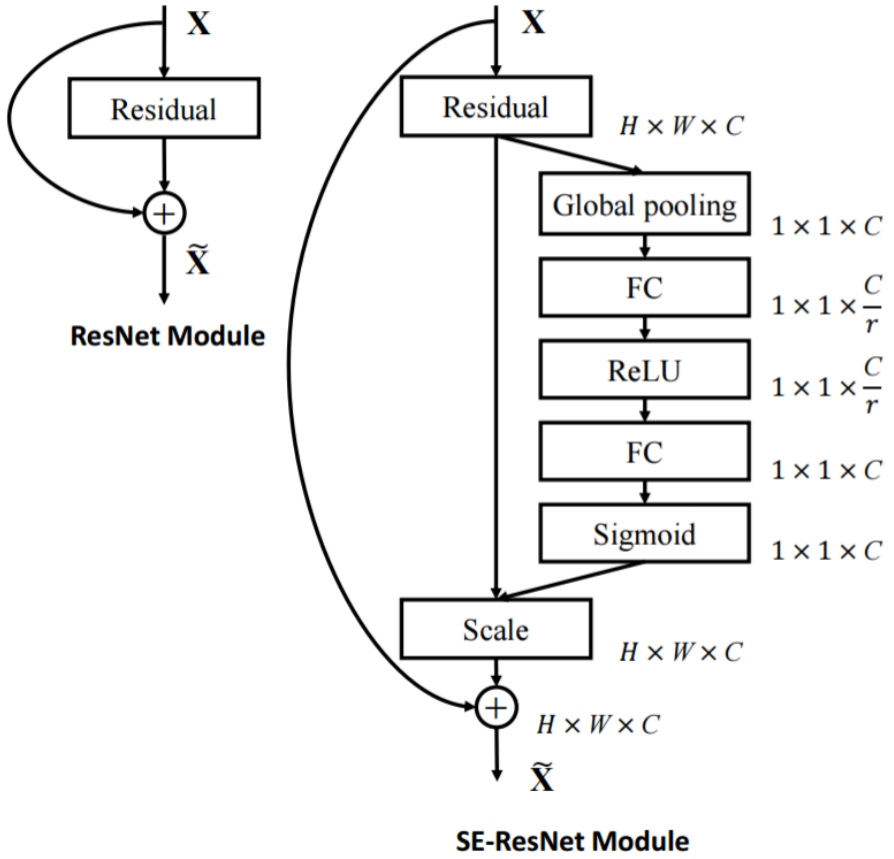
- SENet (Squeeze-and-Excitation Networks)（最后一年ImageNet竞赛冠军）



Squeeze-and-Excitation Networks. CVPR, 2018.

与主流网络的结合


X
Inception
$\tilde{X}$
Inception Module
X
Inception
$H \times W \times C$
Global pooling
$1 \times 1 \times C$
FC
$1 \times 1 \times \frac{C}{r}$
ReLU
$1 \times 1 \times \frac{C}{r}$
FC
$1 \times 1 \times C$
Sigmoid
$1 \times 1 \times C$
Scale
$H \times W \times C$
$\tilde{X}$
SE-Inception Module


X
Residual
$\tilde{X}$
ResNet Module
X
Residual
$H \times W \times C$
Global pooling
$1 \times 1 \times C$
FC
$1 \times 1 \times \frac{C}{r}$
ReLU
$1 \times 1 \times \frac{C}{r}$
FC
$1 \times 1 \times C$
Sigmoid
$1 \times 1 \times C$
Scale
$H \times W \times C$
$H \times W \times C$
$\tilde{X}$
SE-ResNet Module

Summary

- 掌握近年来的主流卷积操作和网络结构
- 结合实际问题和任务灵活使用
- 持续了解和跟进研究进展
- 自己能有一些思考、拓展和突破
- 深度学习的可解释性 (可信赖性)