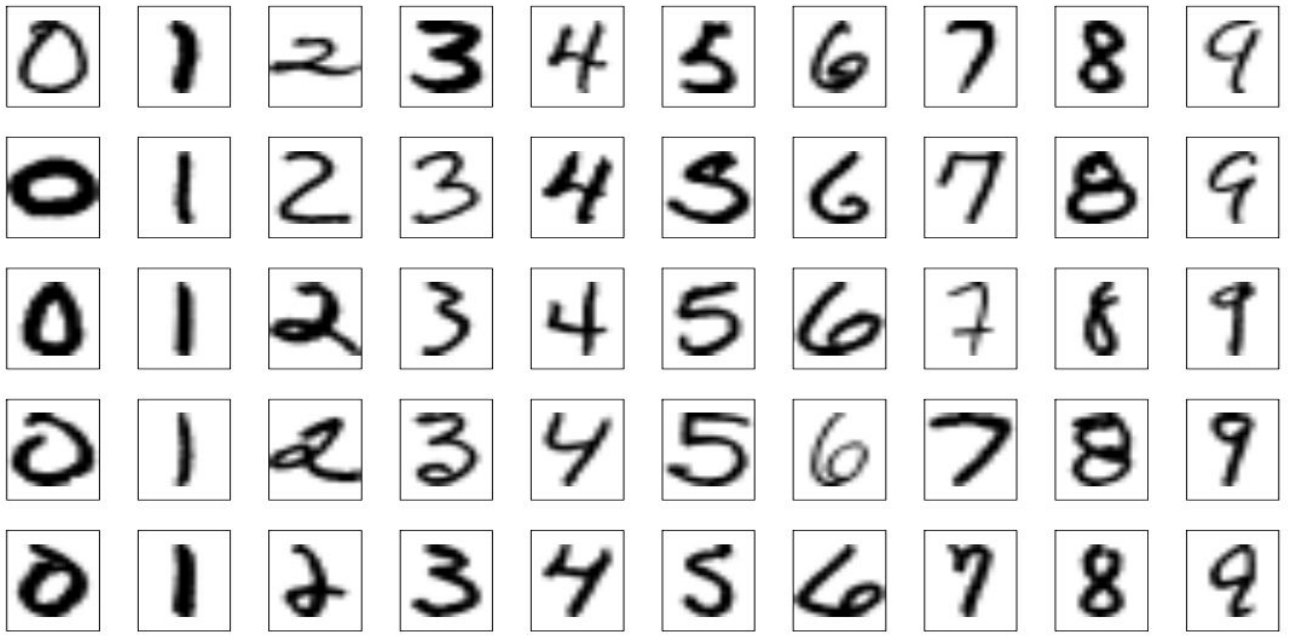
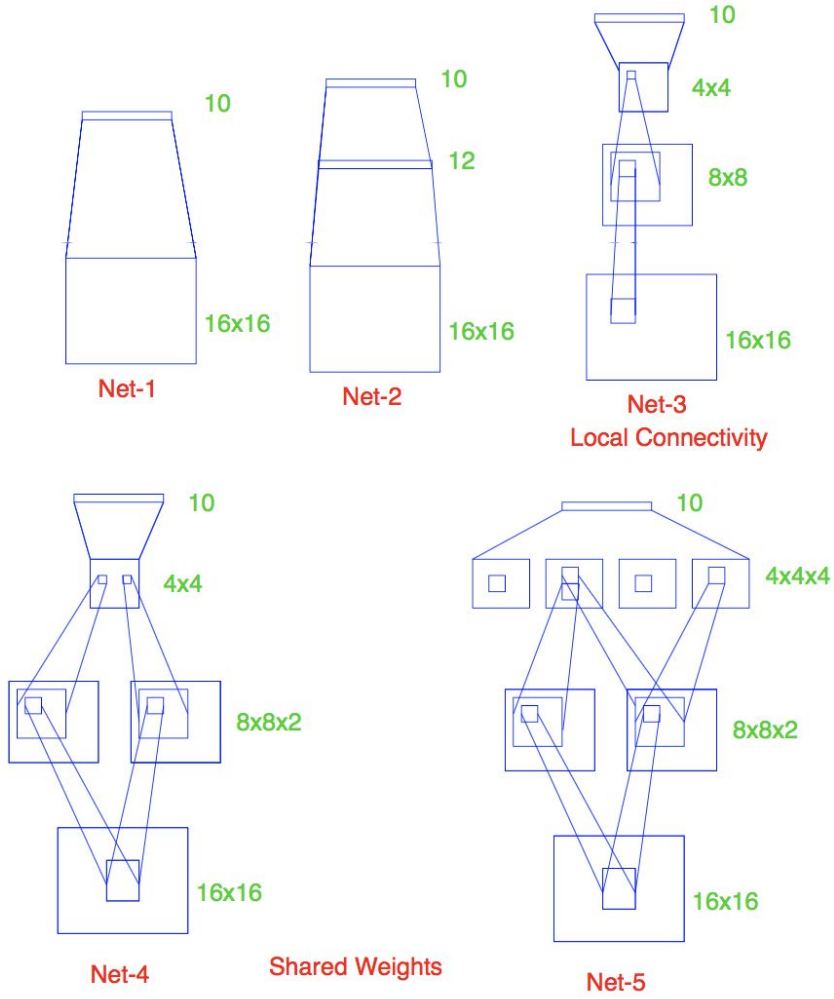
Deep Learning

CNN





Net-1: No hidden layer, equivalent to multinomial logistic regression.

Net-2: One hidden layer, 12 hidden units fully connected.

Net-3: Two hidden layers locally connected.

Net-4: Two hidden layers, locally connected with weight sharing.

Net-5: Two hidden layers, locally connected, two levels of weight sharing.

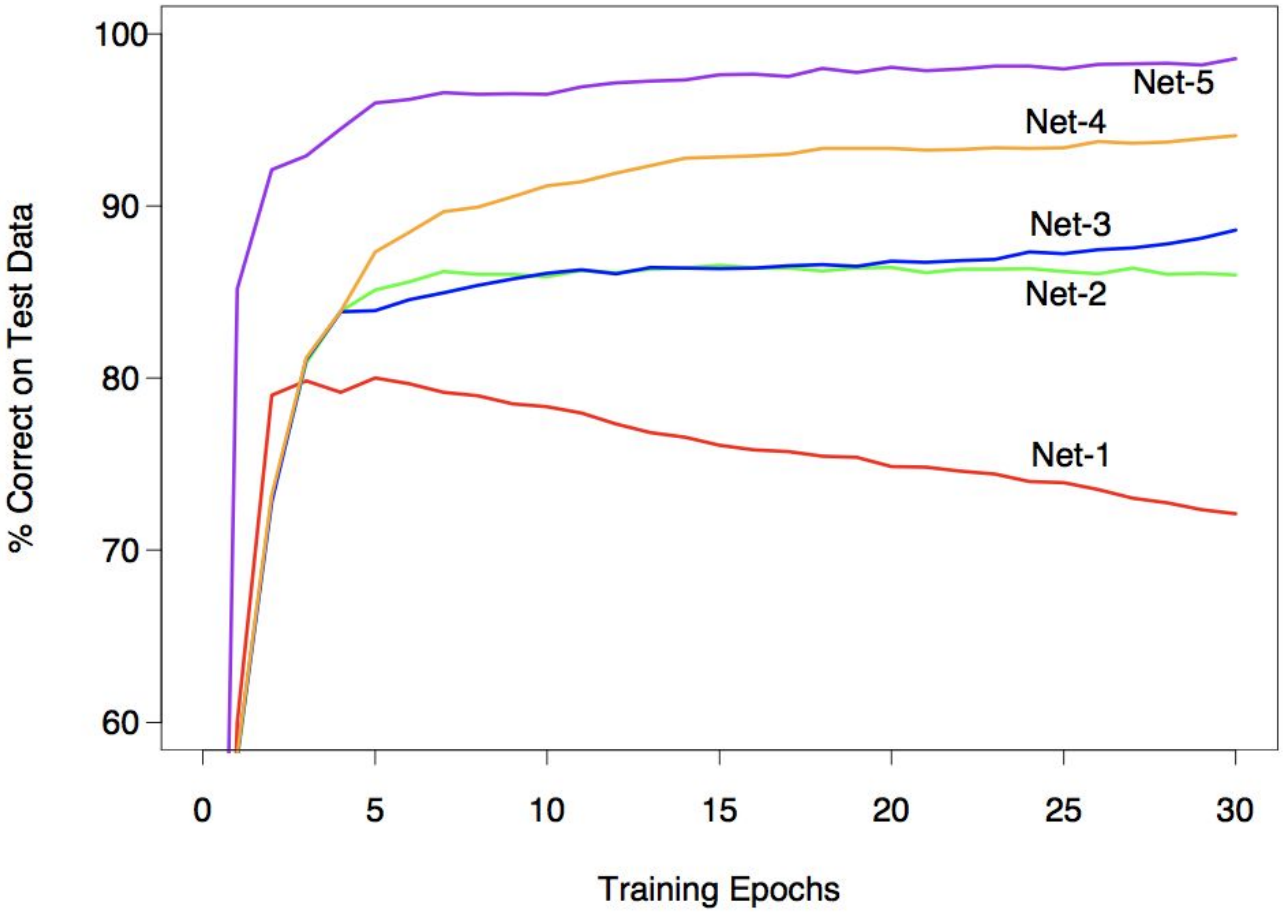
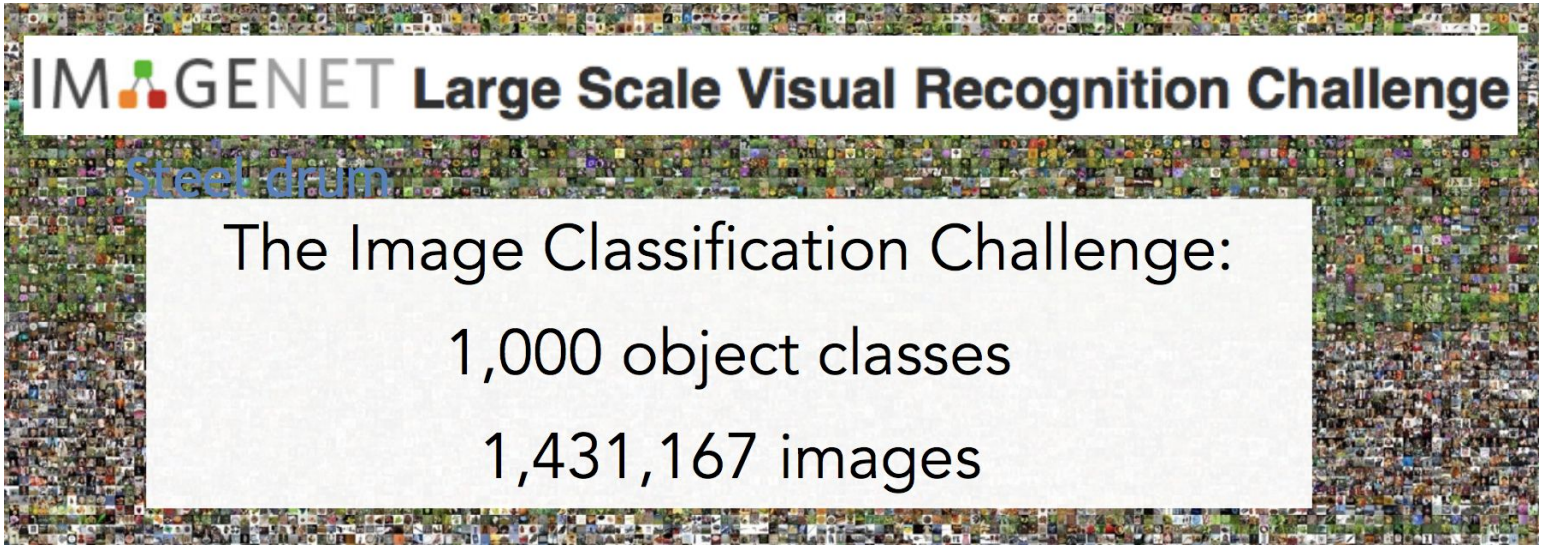


FIGURE 11.11. *Test performance curves, as a function of the number of training epochs, for the five networks of Table 11.1 applied to the ZIP code data. (Le Cun, 1989)*


IMAGENET Large Scale Visual Recognition Challenge
Steel drum
The Image Classification Challenge:
1,000 object classes
1,431,167 images

IMAGENET Large Scale Visual Recognition Challenge

Steel drum

The Image Classification Challenge:

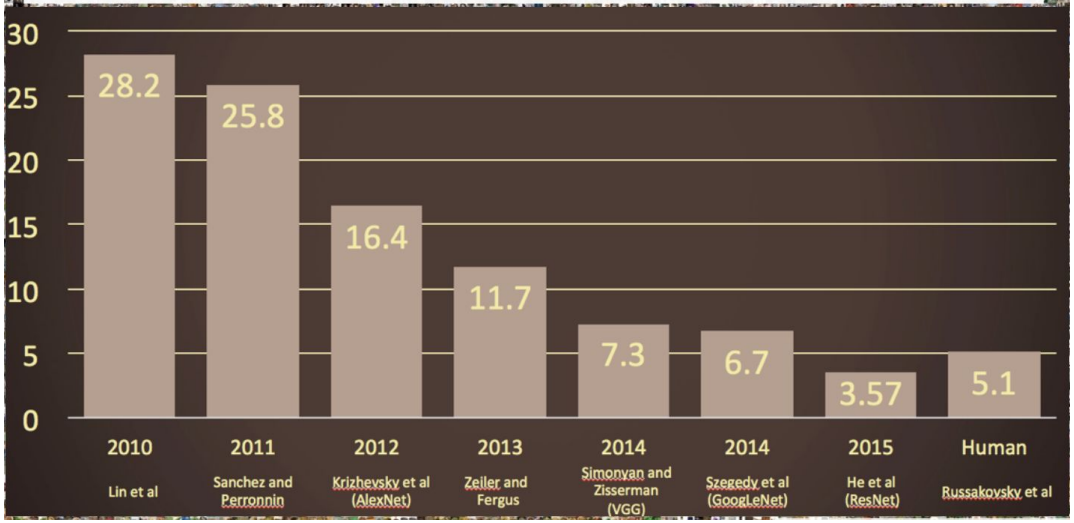
1,000 object classes

1,431,167 images

The Image Classification Challenge:

1,000 object classes

1,431,167 images



IMAGENET Large Scale Visual Recognition Challenge

Year 2010

NEC-UIUC

Dense descriptor grid: HOG, LBP

Coding: local coordinate, super-vector

Pooling, SPM

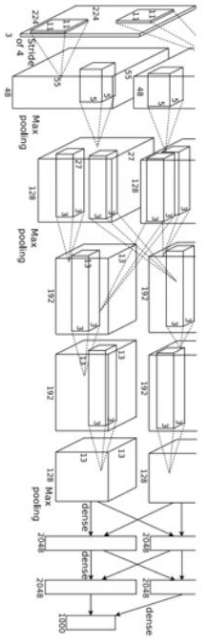
Linear SVM

[Lin CVPR 2011]

Lion image by Swissfrog is

Year 2012

SuperVision

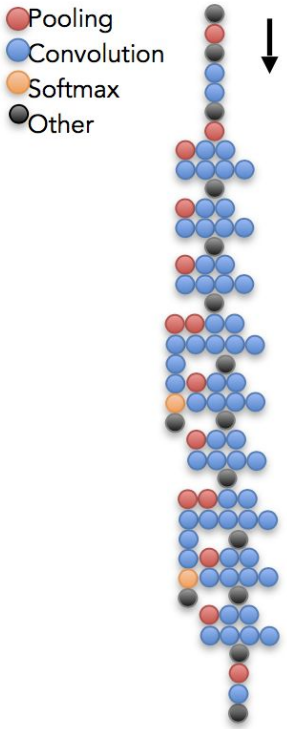


[Krizhevsky NIPS 2012]

Figure copyright Alex Krizhevsky, Ilya

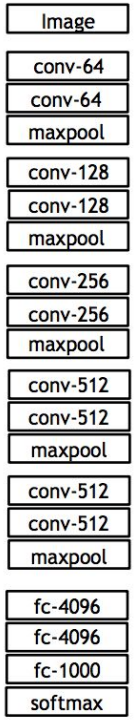
Year 2014

GoogLeNet



[Szegedy arxiv 2014]

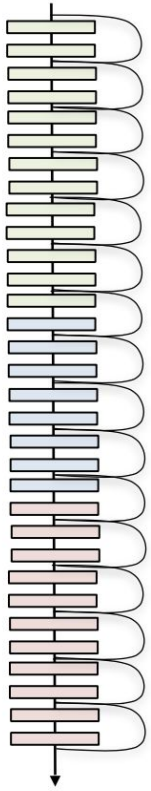
VGG



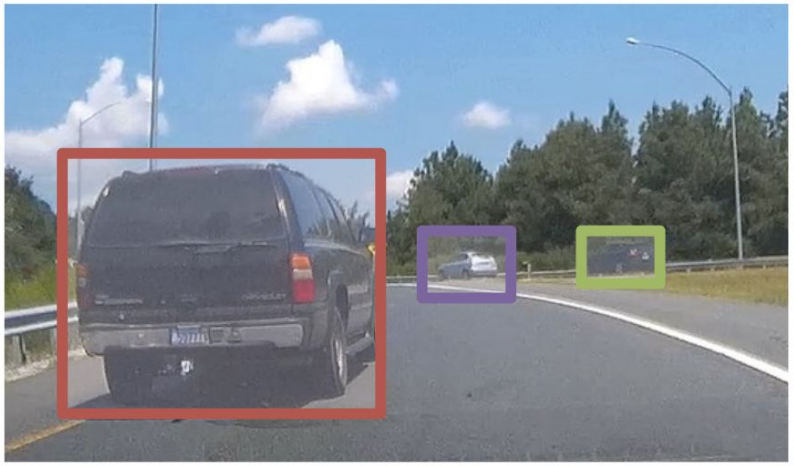
[Simonyan arxiv 2014]

Year 2015

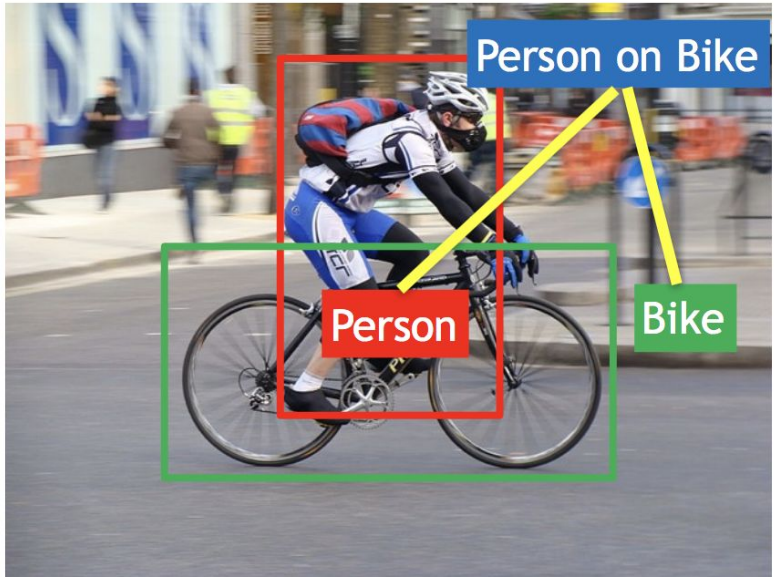
MSRA



[He ICCV 2015]

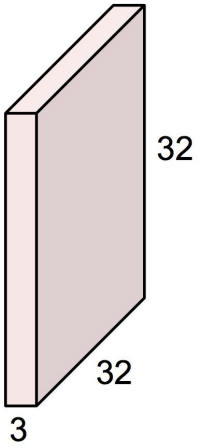


- Object detection
- Action classification
- Image captioning
- …



Convolution Layer

32x32x3 image

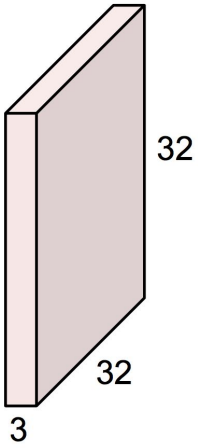


5x5x3 filter

Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Convolution Layer

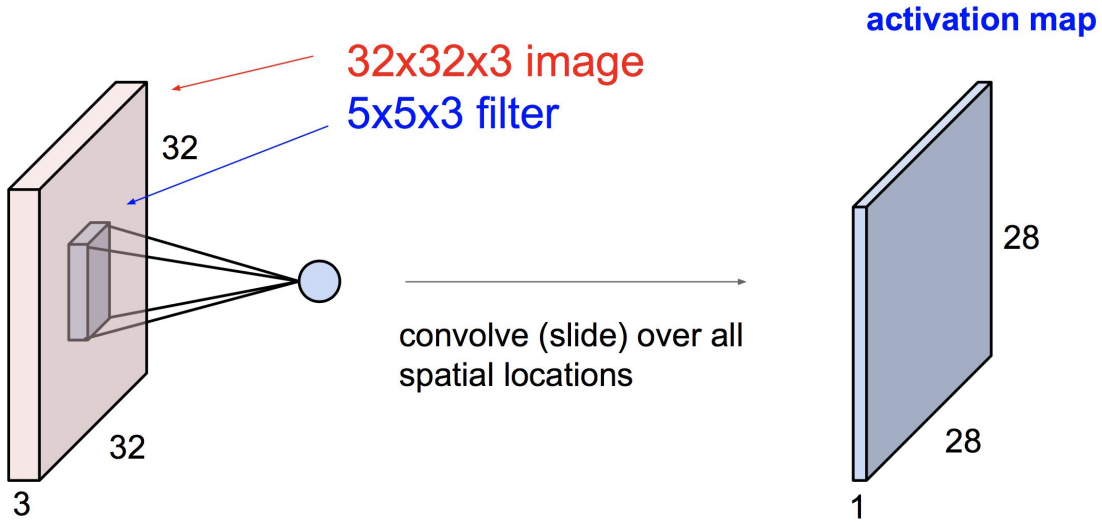
32x32x3 image

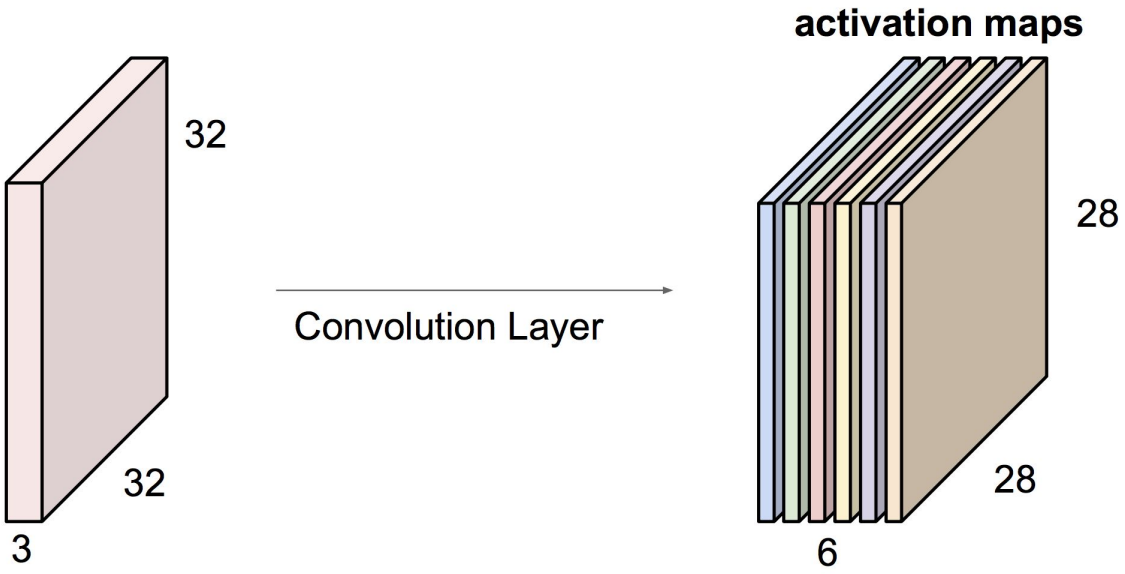


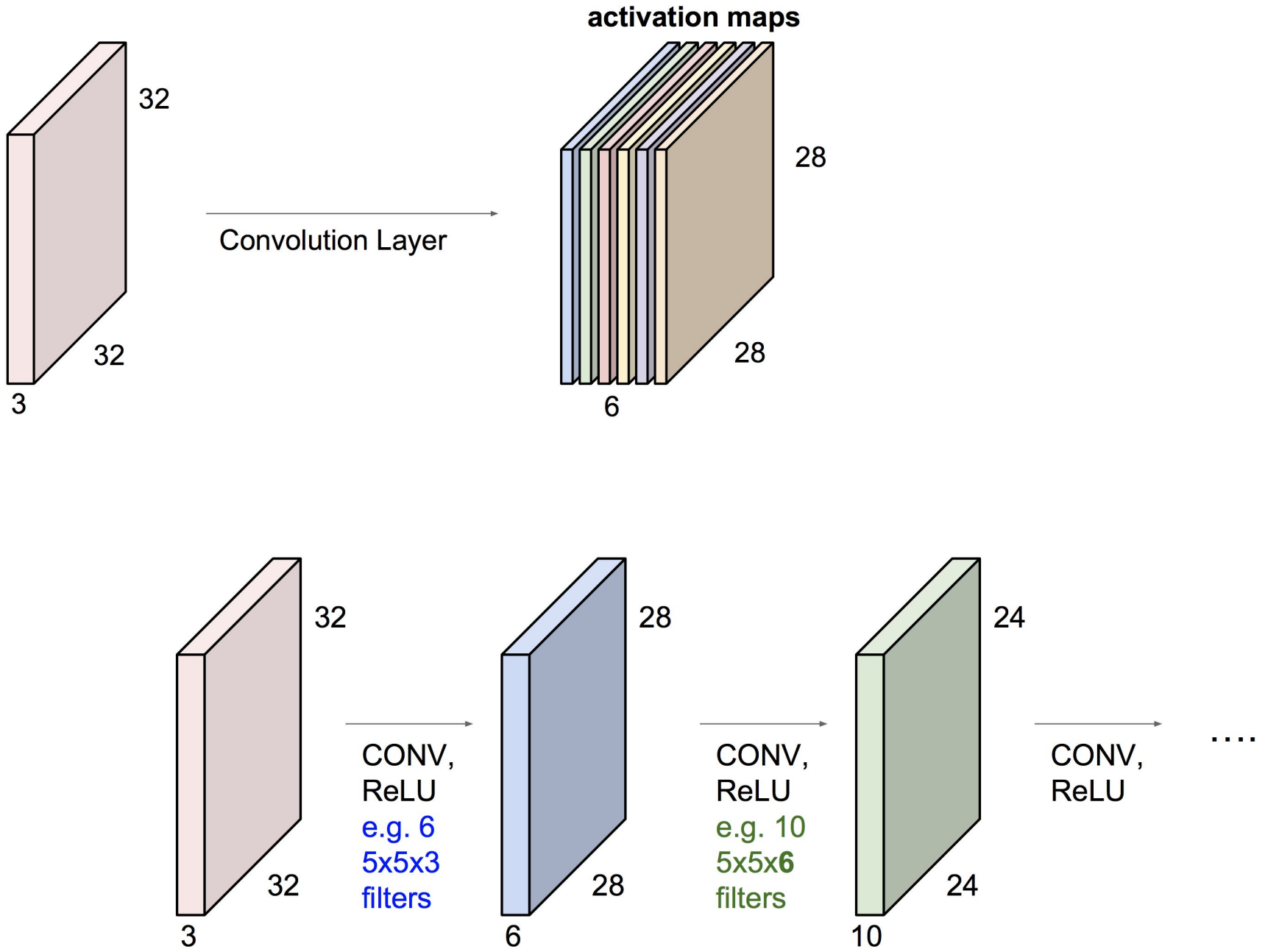
5x5x3 filter

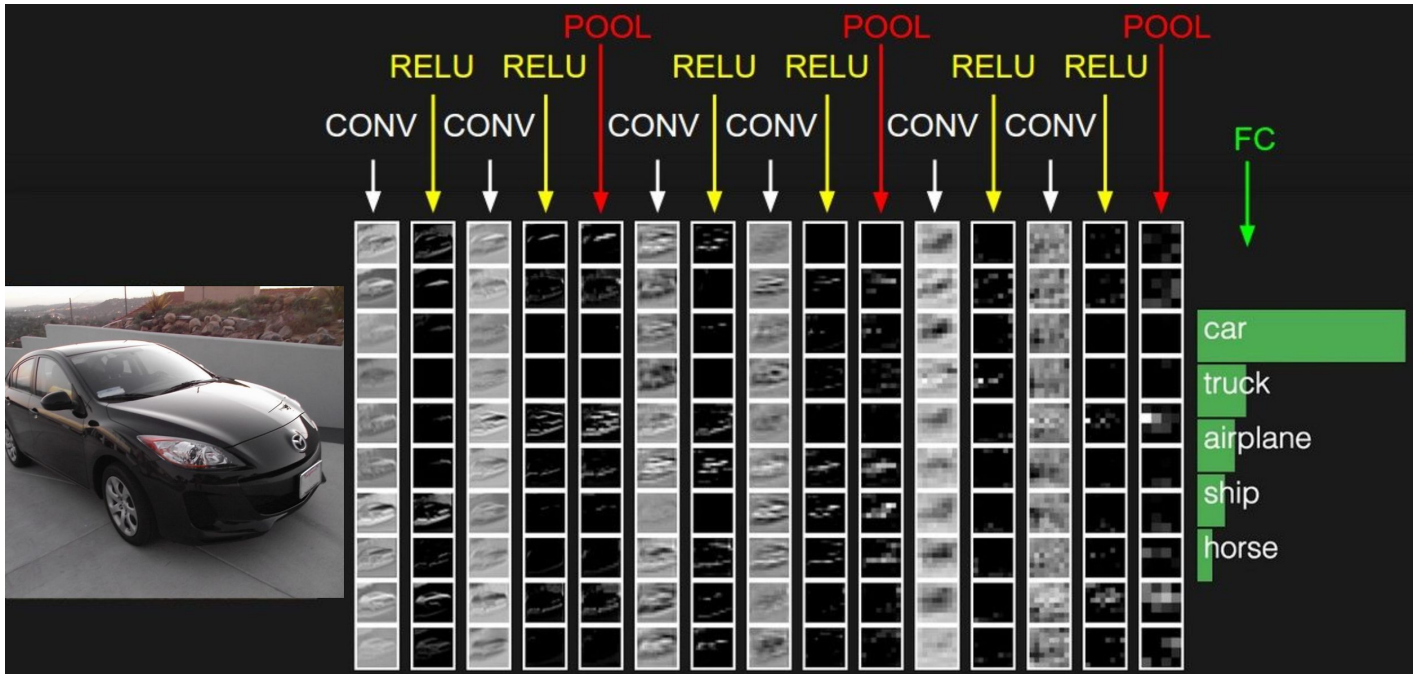
Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Convolution Layer



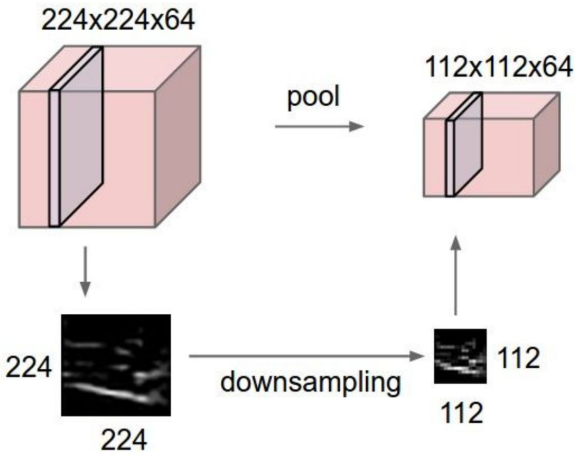

activation maps
32
32
3
Convolution Layer
28
28
6

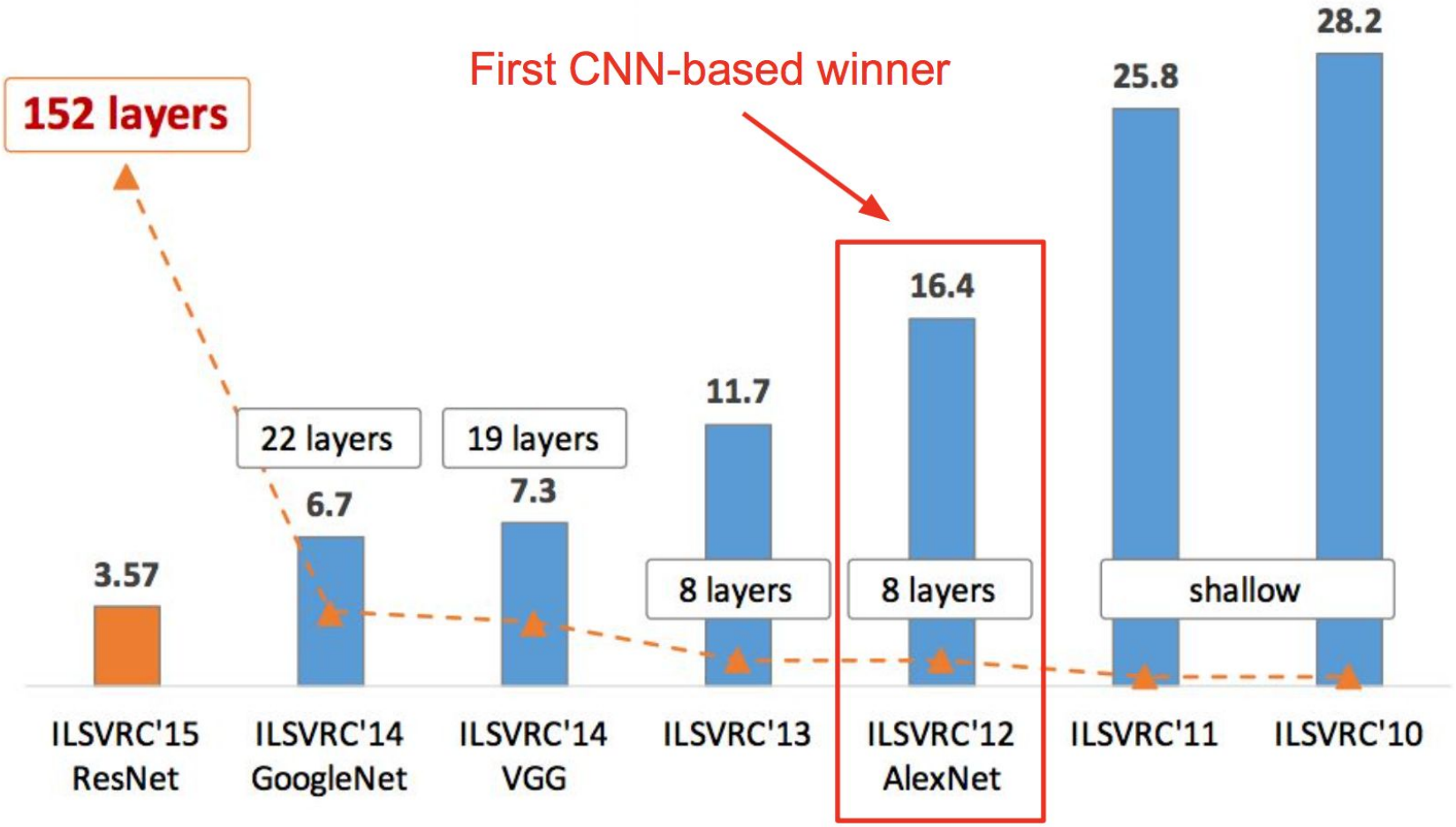

activation maps
32
32
3
Convolution Layer
28
28
6
32
32
3
CONV,
ReLU
e.g. 6
5x5x3
filters
28
28
6
CONV,
ReLU
e.g. 10
5x5x6
filters
24
24
10
CONV,
ReLU
....



Pooling layer

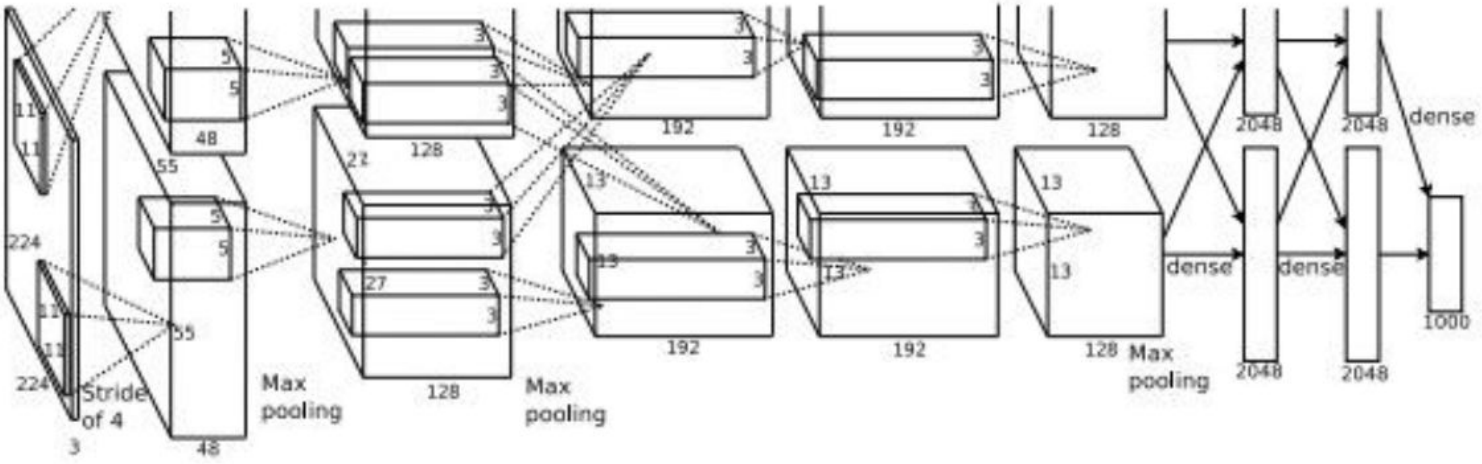
- makes the representations smaller and more manageable
- operates over each activation map independently:

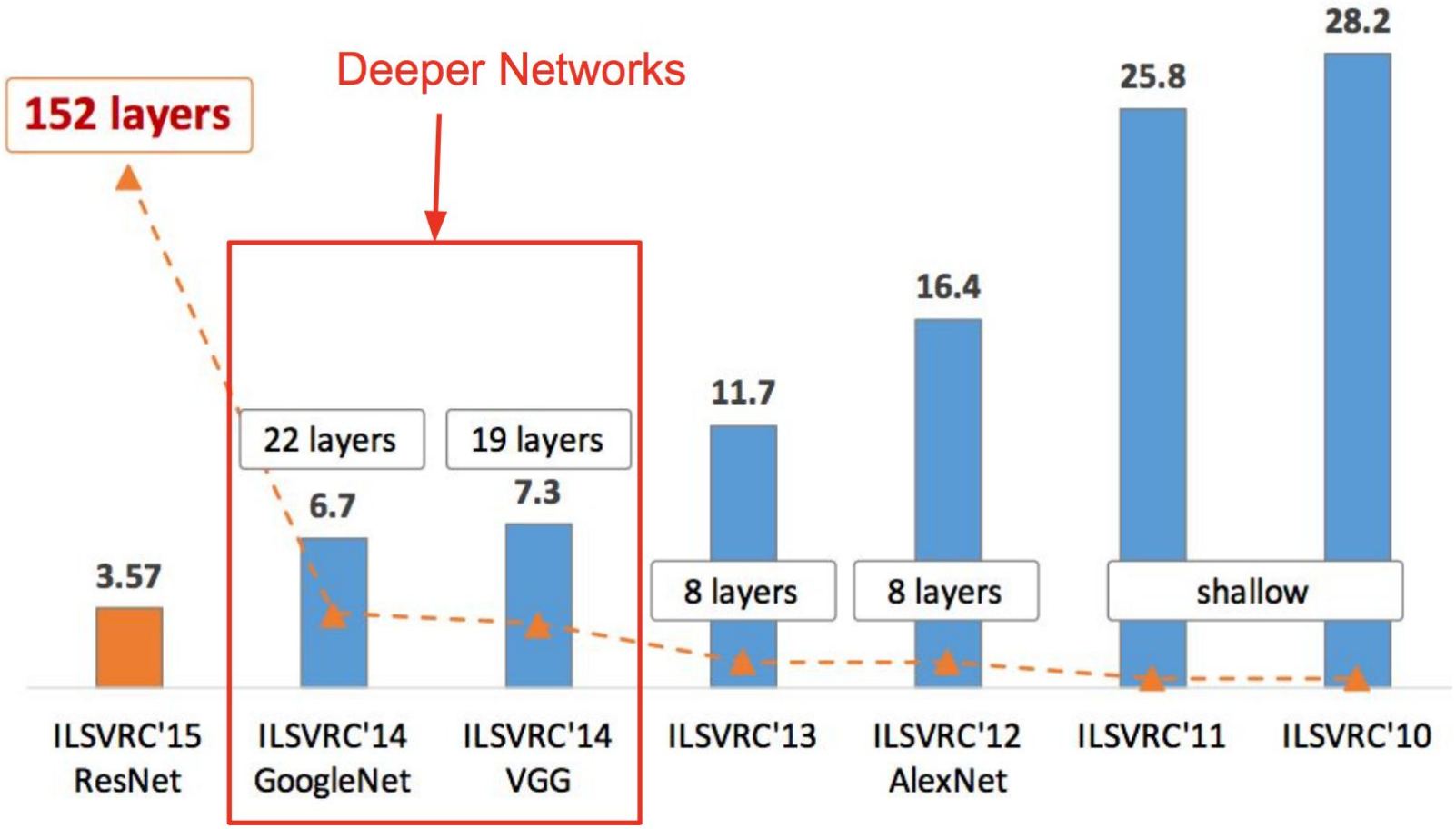



First CNN-based winner
152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15
ResNet
ILSVRC'14
GoogleNet
ILSVRC'14
VGG
ILSVRC'13
ILSVRC'12
AlexNet
ILSVRC'11
ILSVRC'10

Imagenet classification with deep convolutional neural networks

Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, 2012

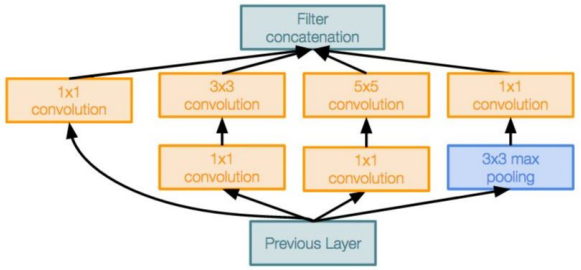



Deeper Networks
152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15
ResNet
ILSVRC'14
GoogleNet
ILSVRC'14
VGG
ILSVRC'13
ILSVRC'12
AlexNet
ILSVRC'11
ILSVRC'10

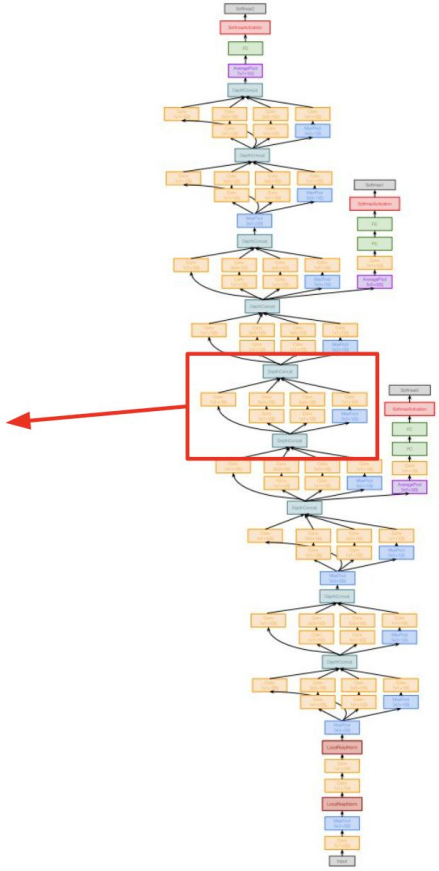
Case Study: GoogLeNet

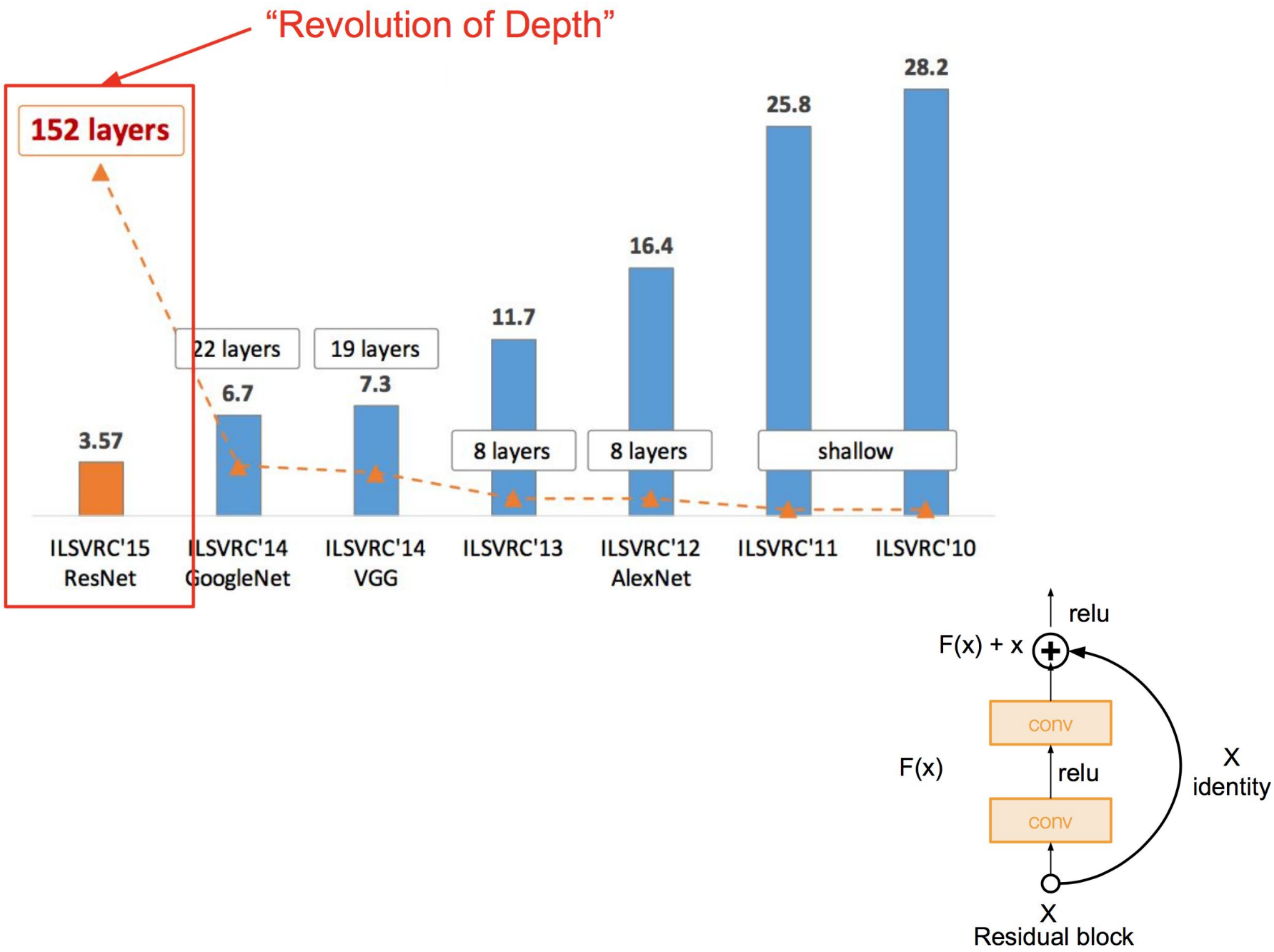
[Szegedy et al., 2014]

"Inception module": design a good local network topology (network within a network) and then stack these modules on top of each other



Inception module




"Revolution of Depth"
152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15 ResNet
ILSVRC'14 GoogleNet
ILSVRC'14 VGG
ILSVRC'13
ILSVRC'12 AlexNet
ILSVRC'11
ILSVRC'10
relu
F(x) + x
conv
F(x)
relu
conv
X
identity
X
Residual block

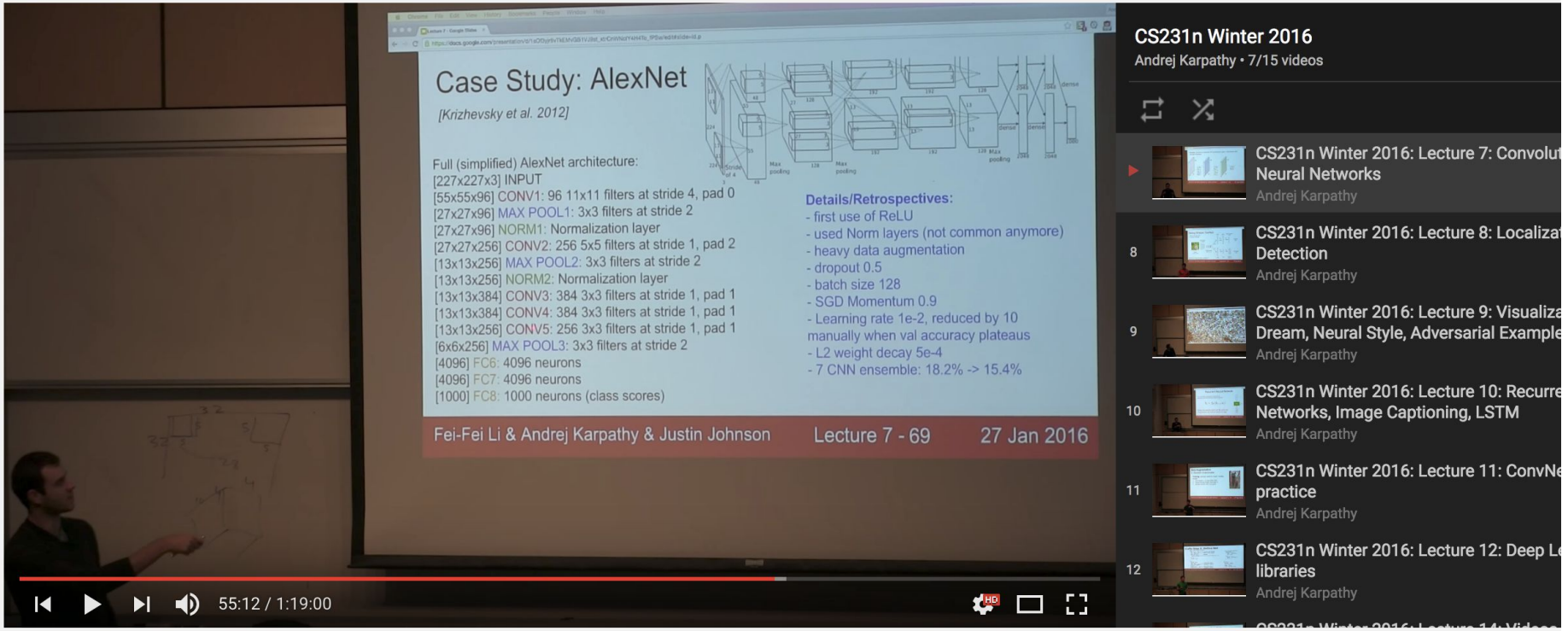

REPUBLIC OF GAMERS
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
EVGA
SLI

cs231n.stanford.edu

Apps bm

CS231n: Convolutional Neural Networks for Visual Recognition

Spring 2017



CS231n Winter 2016: Lecture 7: Convolutional Neural Networks

Andrej Karpathy

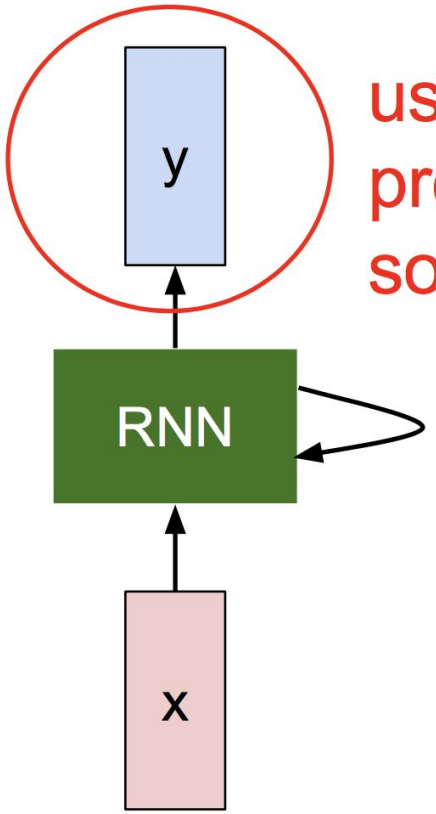
CS231n Winter 2016: L Localization and Detec

Andrej Karpathy

Deep Learning

RNN / LSTM

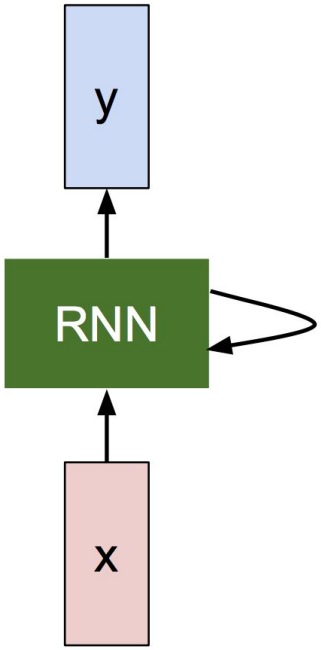
Recurrent Neural Network



usually want to predict a vector at some time steps

(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



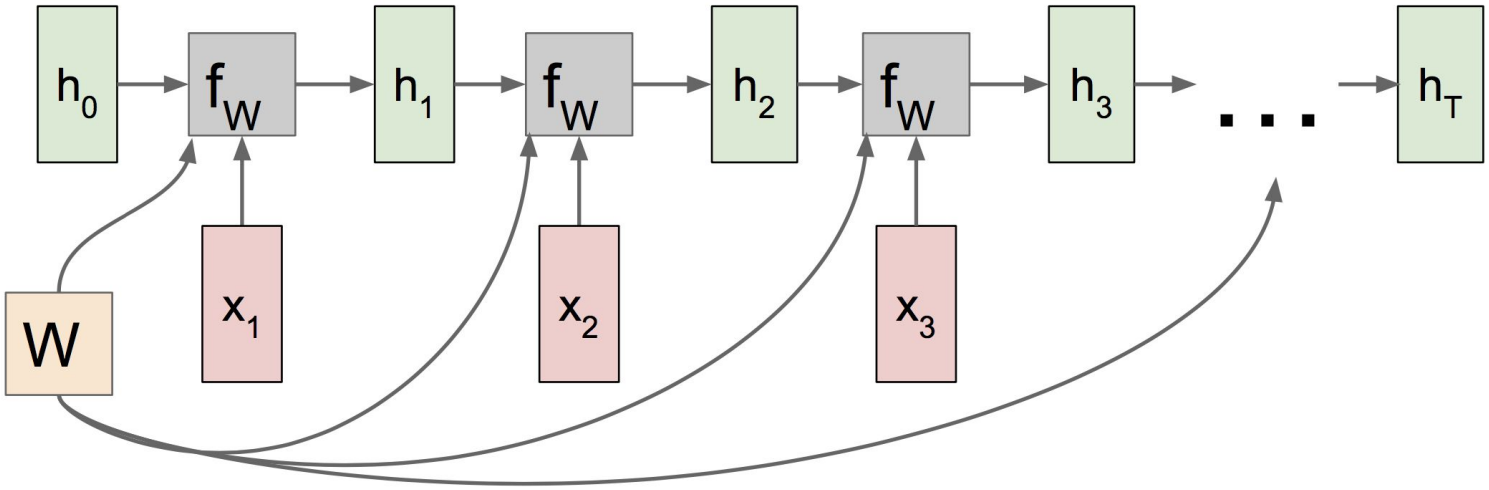
$$h_t = f_W(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

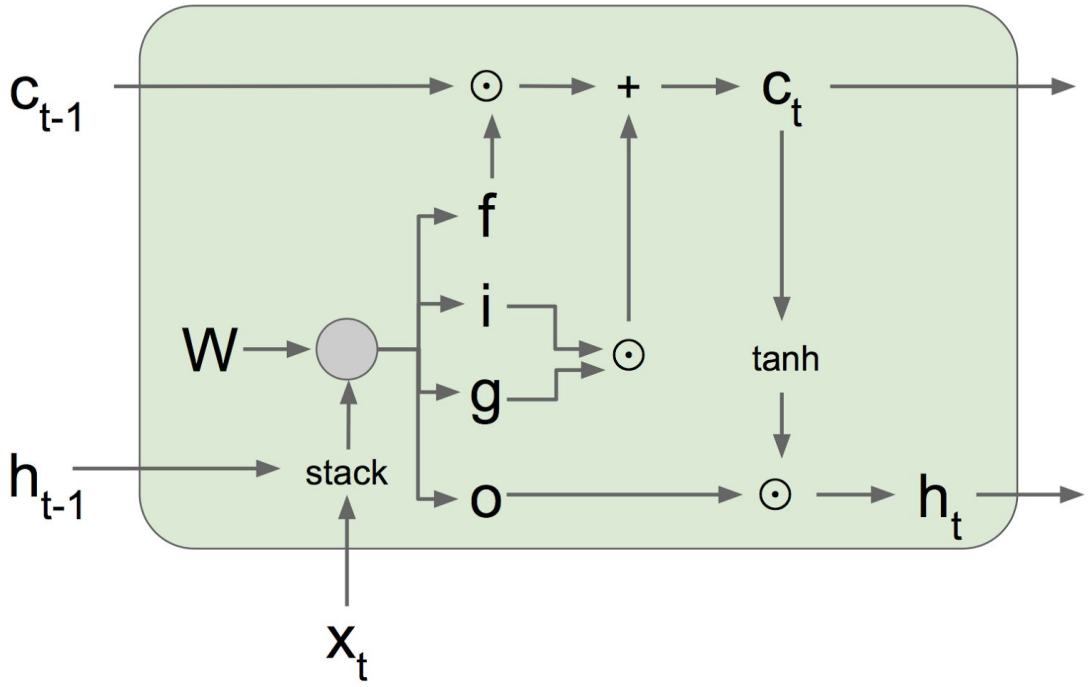
RNN: Computational Graph

Re-use the same weight matrix at every time-step



Long Short Term Memory (LSTM)

[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

Unsupervised Learning

Clustering

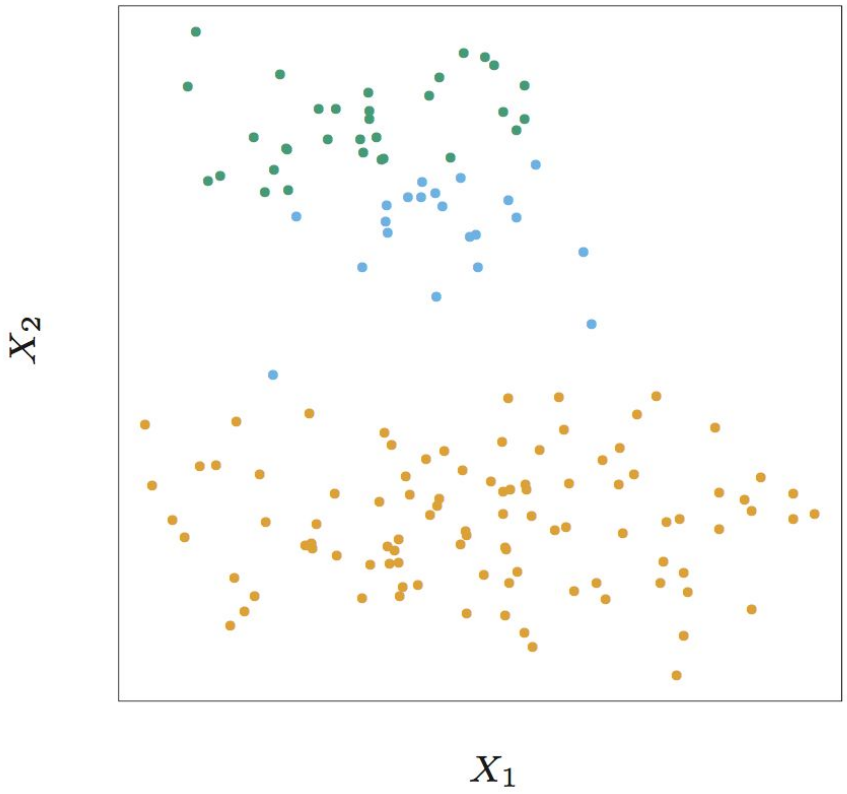


FIGURE 14.4. *Simulated data in the plane, clustered into three classes (represented by orange, blue and green) by the K-means clustering algorithm*

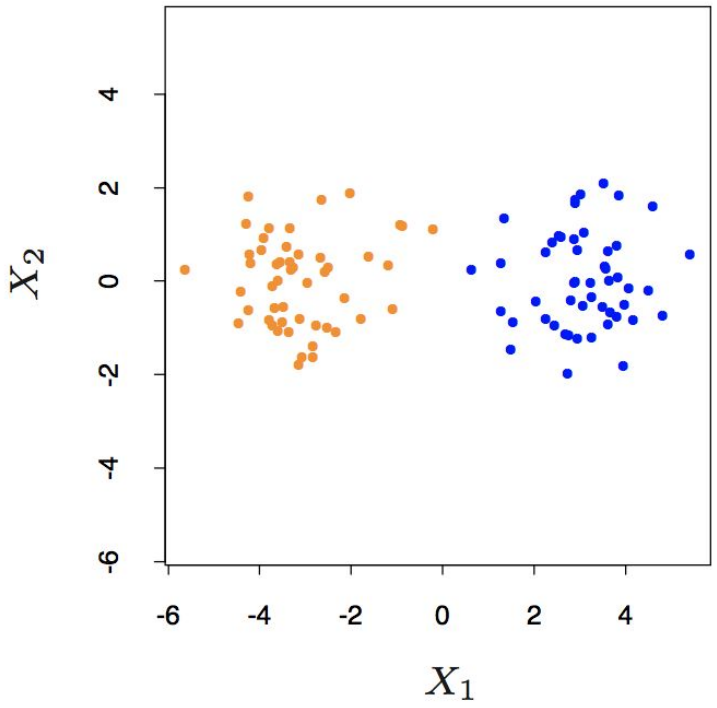
$$D(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot d_j(x_{ij}, x_{i'j});$$

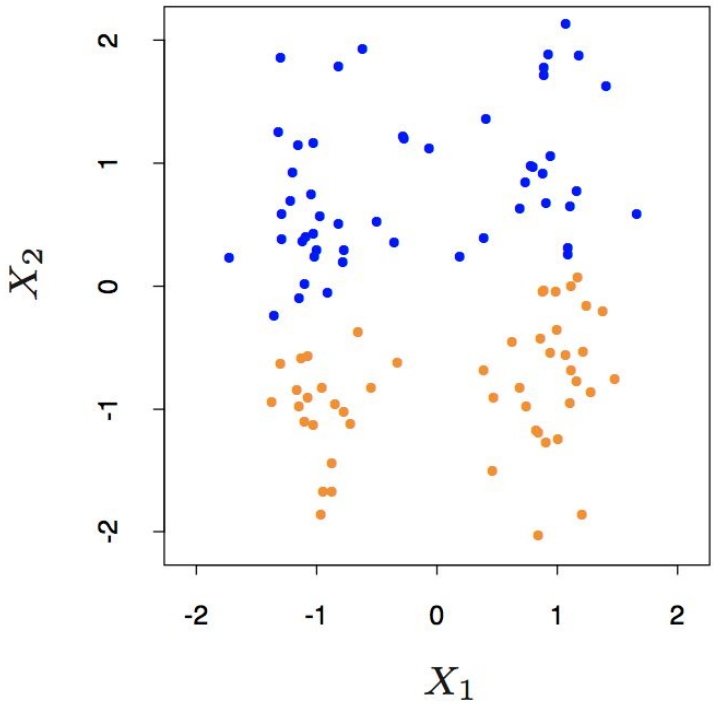
$$\bar{D} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} D(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot \bar{d}_j,$$

$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_j(x_{ij}, x_{i'j})$$

jects. For example, with p quantitative variables and squared-error distance used for each coordinate, then (14.24) becomes the (weighted) squared Euclidean distance

$$D_I(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot (x_{ij} - x_{i'j})^2 \tag{14.26}$$


X_2
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
X_1


X_2
2
1
0
-1
-2
-2
-1
0
1
2
X_1

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k} d(x_i, x_{i'}).$$

$$T = \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N} d_{ii'} = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\left(\sum_{C(i')=k} d_{ii'} + \sum_{C(i')\neq k} d_{ii'}\right)$$

or

$$T = W(C) + B(C),$$

$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')\neq k} d_{ii'}$$

$$W(C) = T - B(C)$$

and minimizing $W(C)$ is equivalent to *maximizing* $B(C)$.

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = ||x_i - x_{i'}||^2$$

$$\begin{aligned} W(C) &= \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} ||x_i - x_{i'}||^2 \\ &= \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \bar{x}_k||^2, \end{aligned}$$

$$C^* = \min_{C} \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \bar{x}_k||^2$$

can be obtained by noting that for any set of observations S

$$\bar{x}_S = \operatorname*{argmin}_{m} \sum_{i \in S} ||x_i - m||^2. \tag{14.32}$$

Hence we can obtain C^* by solving the enlarged optimization problem

$$\min_{C, \{m_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - m_k||^2. \tag{14.33}$$

Algorithm 14.1 *K-means Clustering.*

1. For a given cluster assignment C, the total cluster variance (14.33) is minimized with respect to $\{m_1, \ldots, m_K\}$ yielding the means of the currently assigned clusters (14.32).

2. Given a current set of means $\{m_1, \ldots, m_K\}$, (14.33) is minimized by assigning each observation to the closest (current) cluster mean. That is,

$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} ||x_i - m_k||^2. \tag{14.34}$$

3. Steps 1 and 2 are iterated until the assignments do not change.

NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set

	Index
1	CH (Calinski and Harabasz 1974)
2	CCC (Sarle 1983)
3	Pseudot2 (Duda and Hart 1973)
4	KL (Krzanowski and Lai 1988)
5	Gamma (Baker and Hubert 1975)
6	Gap (Tibshirani *et al.* 2001)
7	Silhouette (Rousseeuw 1987)
8	Hartigan (Hartigan 1975)
9	Cindex (Hubert and Levin 1976)
10	DB (Davies and Bouldin 1979)
11	Ratkowsky (Ratkowsky and Lance 1978)
12	Scott (Scott and Symons 1971)
13	Marriot (Marriot 1971)
14	Ball (Ball and Hall 1965)
15	Trcovw (Milligan and Cooper 1985)
16	Tracew (Milligan and Cooper 1985)
17	Friedman (Friedman and Rubin 1967)
18	Rubin (Friedman and Rubin 1967)
19	Dunn (Dunn 1974)

2.4. Cindex

The C-Index was reviewed in Hubert and Levin (1976). It is calculated using Equation 6.

$$\text{Cindex} = \frac{S_w - S_{\min}}{S_{\max} - S_{\min}},\ S_{\min} \neq S_{\max},\ \text{Cindex} \in (0, 1), \tag{6}$$

where

- $S_{\min}$ = is the sum of the N_w smallest distances between all the pairs of points in the entire data set (there are N_t such pairs);
- $S_{\max}$ = is the sum of the N_w largest distances between all the pairs of points in the entire data set.

2.10. DB index

The Davies and Bouldin (1979) index is a function of the sum ratio of within-cluster scatter to between-cluster separation. It is calculated using Equation 12.

$$\mathrm{DB}(q) = \frac{1}{q}\sum_{k=1}^{q}\max_{k\neq l}\left(\frac{\delta_k + \delta_l}{d_{kl}}\right), \tag{12}$$

where

- $k, l = 1, \ldots, q$ = cluster number,
- $d_{kl} = \sqrt[v]{\sum_{j=1}^{p} |c_{kj} - c_{lj}|^v}$ = distance between centroids of clusters C_k and C_l (for $v = 2$, d_{kl} is the Euclidean distance),
- $\delta_k = \sqrt[u]{\frac{1}{n_k}\sum_{i\in C_k}\sum_{j=1}^{p} |x_{ij} - c_{kj}|^u}$ = dispersion measure of a cluster C_k (for $u = 2$, δ_k is the standard deviation of the distance of objects in cluster C_k to the centroid of this cluster).

Hierarchical Clustering

Agglomerative Clustering

Agglomerative clustering algorithms begin with every observation representing a singleton cluster. At each of the $N-1$ steps the closest two (least dissimilar) clusters are merged into a single cluster, producing one less cluster at the next higher level. Therefore, a measure of dissimilarity between two clusters (groups of observations) must be defined.

in H. *Single linkage* (SL) agglomerative clustering takes the intergroup dissimilarity to be that of the closest (least dissimilar) pair

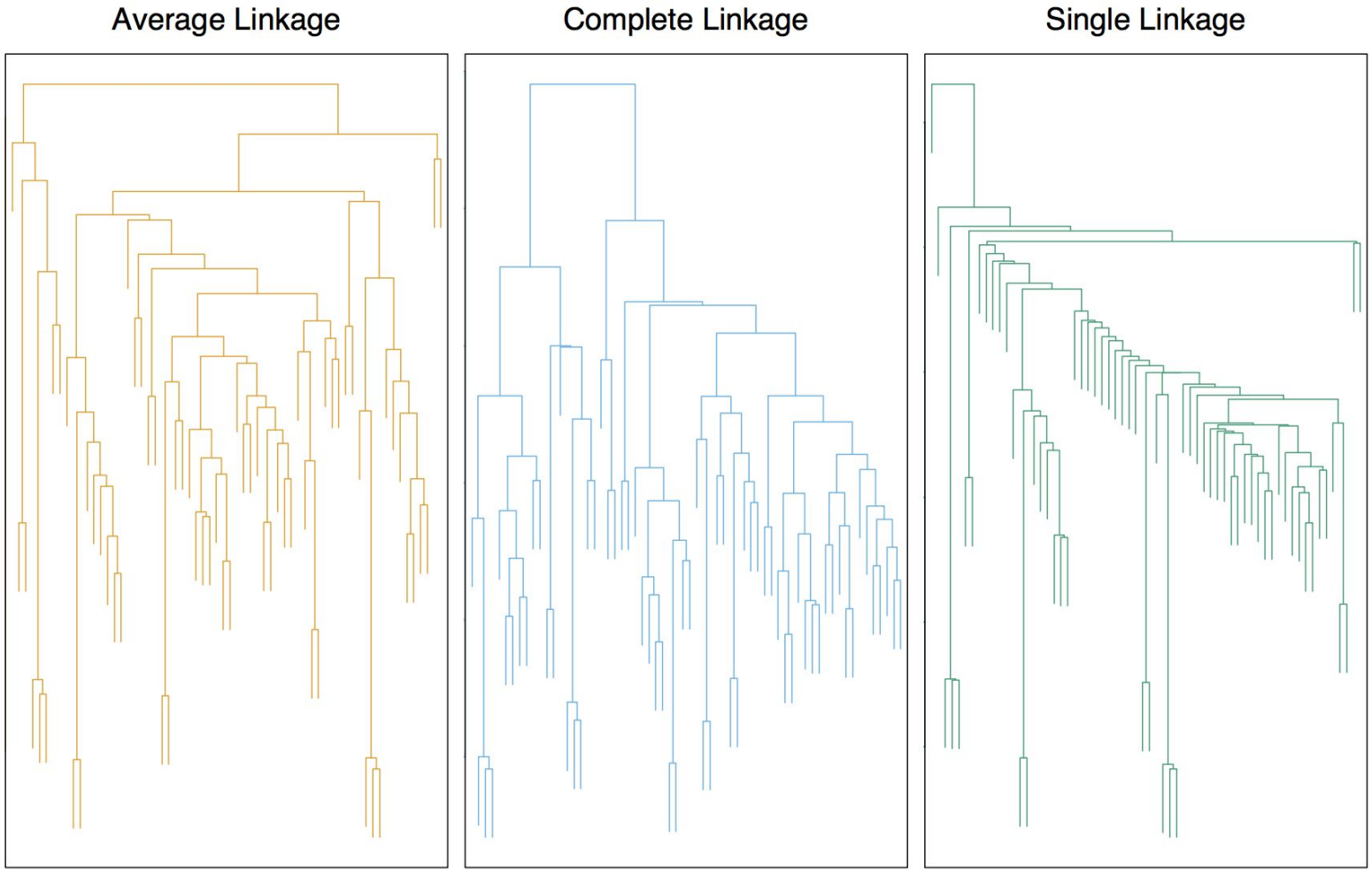
$$d_{SL}(G,H) = \min_{\substack{i\in G \\ i'\in H}} d_{ii'}. \tag{14.41}$$

This is also often called the *nearest-neighbor* technique. *Complete linkage* (CL) agglomerative clustering (*furthest-neighbor* technique) takes the intergroup dissimilarity to be that of the furthest (most dissimilar) pair

$$d_{CL}(G, H) = \max_{\substack{i \in G \\ i' \in H}} d_{ii'}. \tag{14.42}$$

Group average (GA) clustering uses the average dissimilarity between the groups

$$d_{GA}(G, H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{i' \in H} d_{ii'} \tag{14.43}$$


Average Linkage
Complete Linkage
Single Linkage

Reinforcement Learning