Convolutional Neural Networks

Jupyter + Python + scikit-learn (sklearn) demo

Due Date Reminder

GCP credit claimed? Any problem?

Software installation & use, jupyter + google colab

Deadline for choosing project groups (also support each other), with tentative project title, for paper review and presentation: 6 pm, 04/18/2023 (DONE?)

Topic for paper review + presentation can be the same as your final project.

TO-DO: choose 1 task, select 2-5 classes, ~2 weeks. Any report? Due next week

- CIFAR-10
- ImageNet

Python demo with sklearn

Topics Taken

- If you choose your presentation papers outside the reading list, please send me or everyone the papers (at least two) one week before your presentation date so that every else will have time to look at the papers.

- Presentations start 5/9 & 5/16.

- Paper review due on the week (in the class 6 pm) when the presentation will be given.

Google Colab GPU
g.co/colab

The GPU used in the backend is **Tesla K80, P4, T4, P100**
The 12-hour limit is for a continuous assignment of VM

Colab Pro ($9.99/month, cancel anytime), announced 2/2020

- Faster GPU: T4 or P100, even V100
- Longer runtimes: < 24-hour
- More memory: High-memory VMs (2x regular)

colab vm info

python v=3.6.3 (python3)
tensorflow v=1.6.0rc1
2 CPU VM with 1 optional GPU for free
tf device=/device:GPU:0
model name : Intel(R) Xeon(R) CPU @ 2.30GHz
model name : Intel(R) Xeon(R) CPU @ 2.30GHz
MemTotal: 13341960 kB (13GB ram)
MemFree: 4275200 kB
MemAvailable: 11659380 kB

Which version of python you are using?
(first select ' Python 2' in Notebook Settings;
for 'Python 3' – no output)

```
!pip show python
```

Which version of tensorflow you are using?

```
!pip show tensorflow
```

Is GPU Working? (first select 'GPU' in Notebook Settings)

```
import tensorflow as tf
tf.test.gpu_device_name()
```

Which GPU Am I Using?

```
import torch
torch.cuda.get_device_name()
gpu_info = !nvidia-smi

from tensorflow.python.client import device_lib
device_lib.list_local_devices()
```

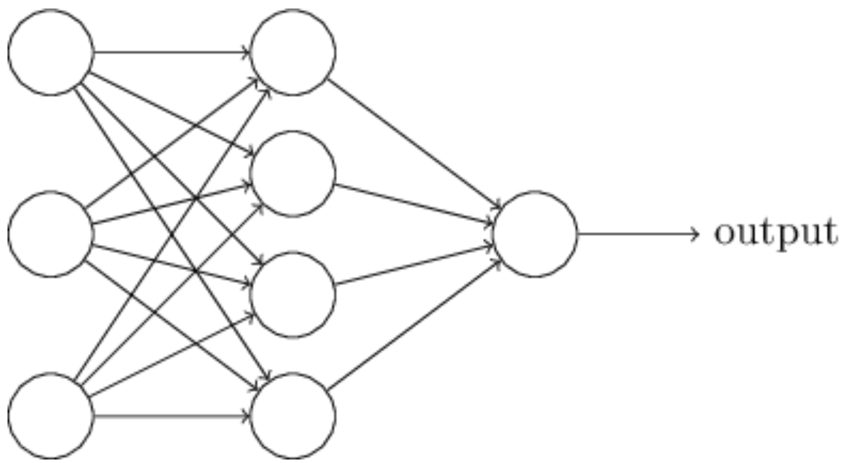
What about RAM?

```
!cat /proc/meminfo
```

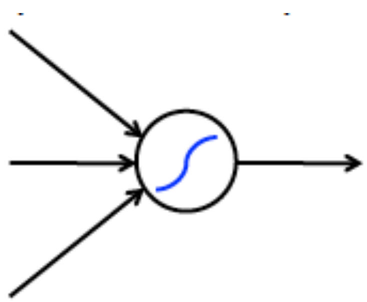
What about CPU?

```
!cat /proc/cpuinfo
```

Last week: Neural Networks

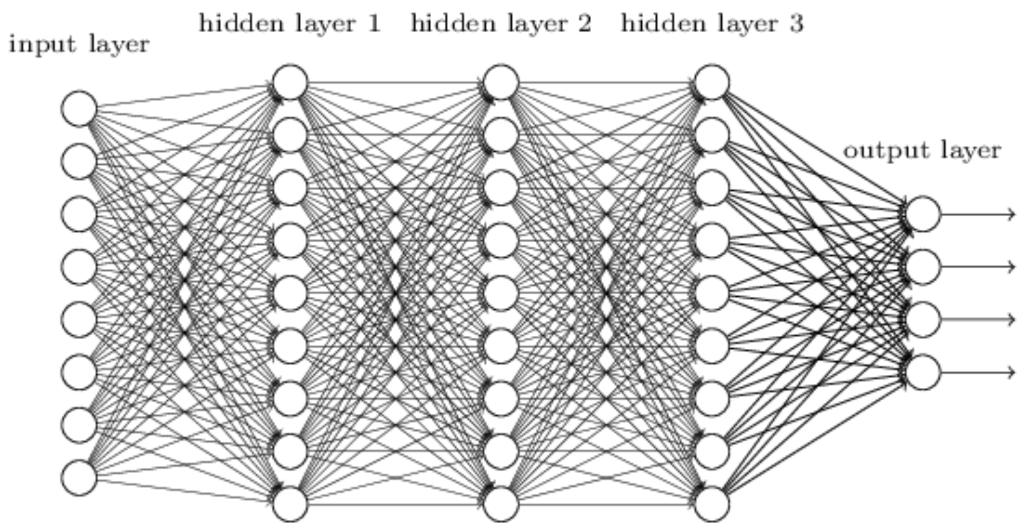


Can learn nonlinear functions provided each perceptron has a differentiable nonlinearity



Sigmoid: $g(t) = \frac{1}{1+e^{-t}}$

Multi-layer Neural Networks



Training Multi-layer Neural Networks

- Find network weights to minimize the *training error* between true and estimated labels of training examples, e.g.:

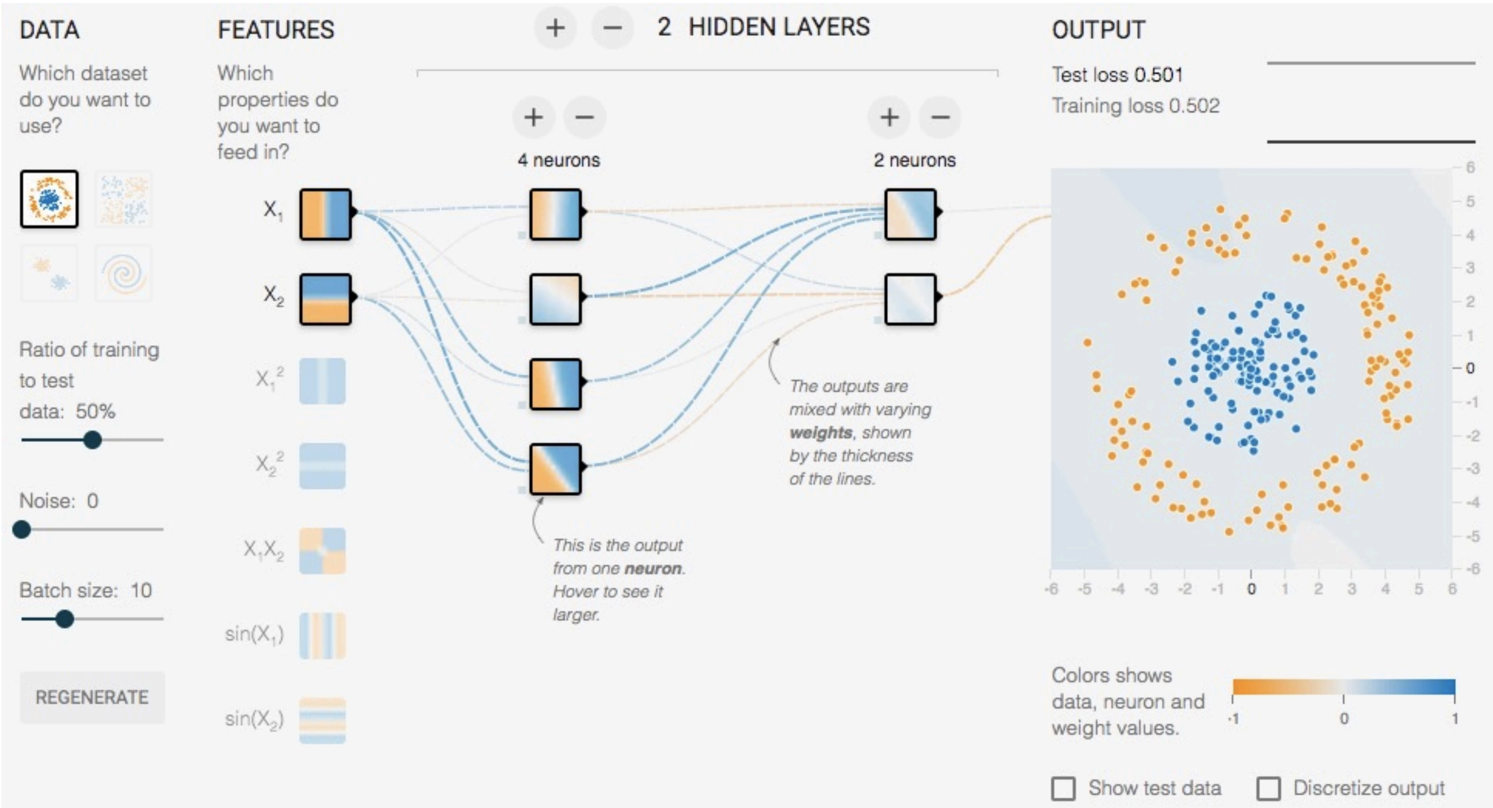
$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

- Update weights by **gradient descent:** $\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial E}{\partial \mathbf{w}}$

- **Back-propagation:** gradients are computed in the direction from output to input layers and combined using chain rule
- **Stochastic gradient descent:** compute the weight update w.r.t. one training example (or a small batch of examples) at a time, cycle through training examples in random order in multiple epochs

Demo: Multi-layer Neural Networks

http://playground.tensorflow.org/

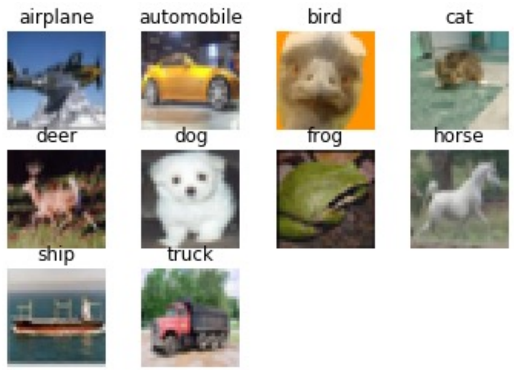


Datasets for Project Ideas

- The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.
    - The MNIST database contains 60,000 training images and 10,000 testing images. Each image is 28x28 black-white.
- The CIFAR-10 dataset (Canadian Institute For Advanced Research) is a collection of images that are commonly used to train machine learning and computer vision algorithms.
    - The CIFAR-10 dataset contains 60,000 32x32 color (RGB) images in 10 different classes. There are 6000 images of each class.
- The ImageNet project is a large visual database designed for use in visual object recognition software research.

ImageNet contains over 20 thousand ambiguous categories; a typical category, such as "balloon" or "strawberry", contains several hundred images.

- YouTube-8M Dataset
- ISIC Archive - Skin Cancer Prediction

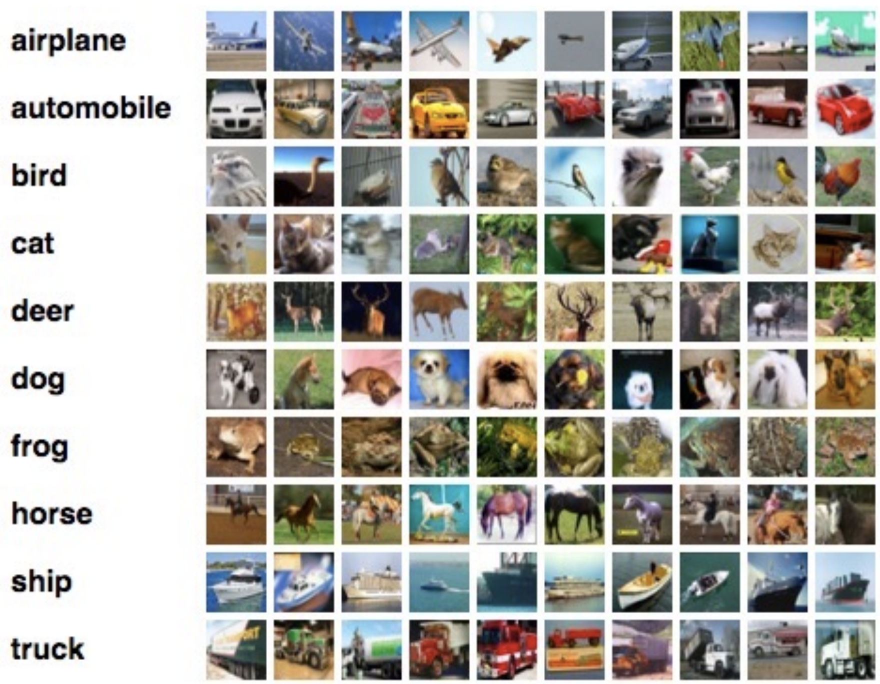


MNIST - Introduction

- MNIST (Mixed National Institute of Standards and Technology) is a database for handwritten digits, distributed by Yann Lecun.
- 60,000 examples, and a test set of 10,000 examples.
- 28x28 pixels each.
- Widely used for research and educational purposes.

(Wikipedia)

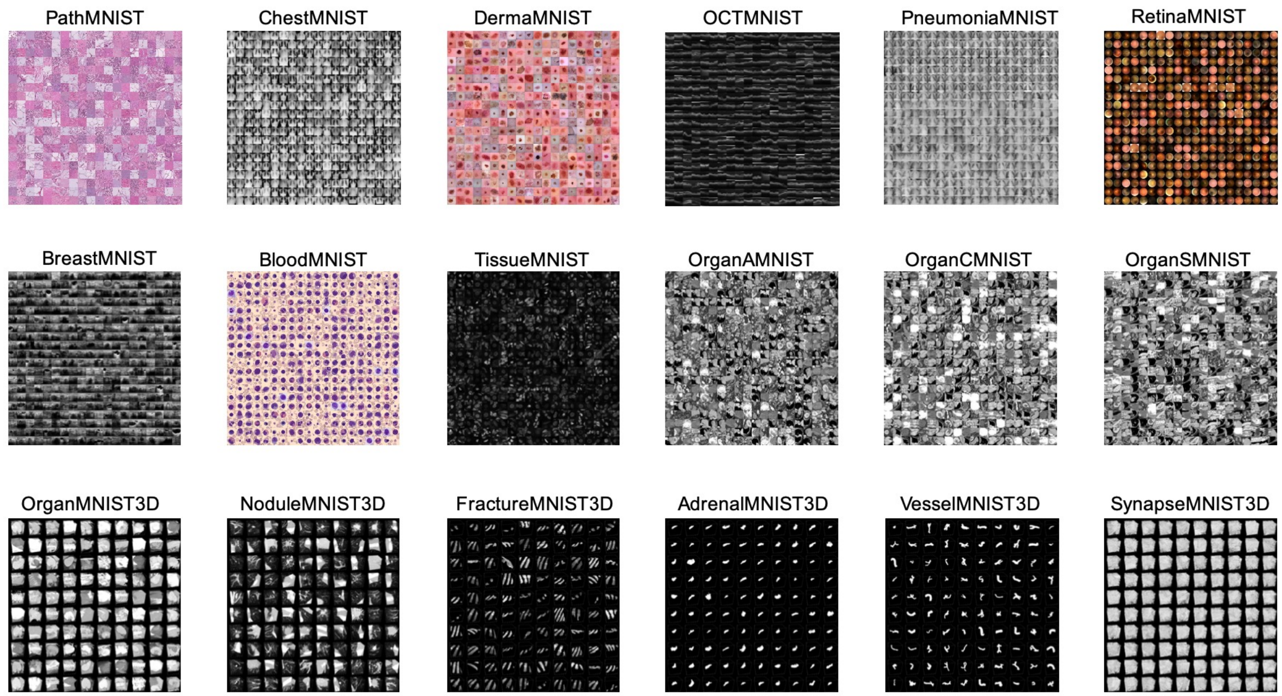
CIFAR-10 Dataset



- The CIFAR-10 dataset (Canadian Institute For Advanced Research) is a collection of images that are commonly used to train machine learning and computer vision algorithms.
- The CIFAR-10 dataset contains 60,000 32x32 color (RGB) images in 10 different classes. There are 6000 images of each class.

(https://www.cs.toronto.edu/~kriz/cifar.html)

MedMNIST



- The MedMNIST v2: A Large-Scale Lightweight Benchmark for 2D and 3D Biomedical Image Classification

(https://medmnist.com)

Data Summary of MedMNIST v2 Dataset

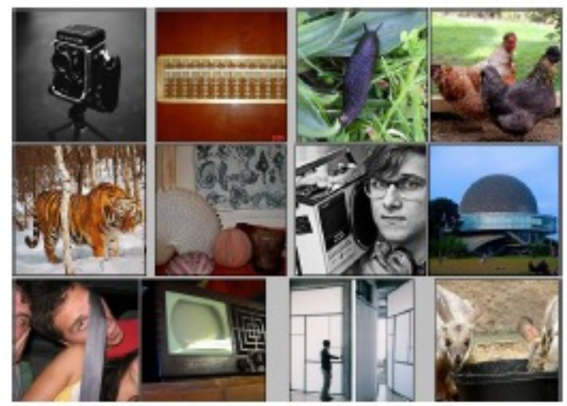
Name	Source	Data Modality	Task (# Classes / Labels)	# Samples	# Training / Validation / Test	License
MedMNIST2D						
PathMNIST	Kather et al.[16]	Colon Pathology	MC (9)	107,180	89,996 / 10,004 / 7,180	CC BY 4.0
ChestMNIST	Wang et al.[17]	Chest X-Ray	ML (14) BC (2)	112,120	78,468 / 11,219 / 22,433	CC0 1.0
DermaMNIST	Tschandl et al.[18], Codella et al.[19]	Dermatoscope	MC (7)	10,015	7,007 / 1,003 / 2,005	CC BY-NC 4.0
OCTMNIST	Kermany et al.[20]	Retinal OCT	MC (4)	109,309	97,477 / 10,832 / 1,000	CC BY 4.0
PneumoniaMNIST	Kermany et al.[20]	Chest X-Ray	BC (2)	5,856	4,708 / 524 / 624	CC BY 4.0
RetinaMNIST	DeepDRiD Team[21]	Fundus Camera	OR (5)	1,600	1,080 / 120 / 400	CC BY 4.0
BreastMNIST	Al-Dhabyani et al.[22]	Breast Ultrasound	BC (2)	780	546 / 78 / 156	CC BY 4.0
BloodMNIST	Acevedo et al.[23]	Blood Cell Microscope	MC (8)	17,092	11,959 / 1,712 / 3,421	CC BY 4.0
TissueMNIST	Ljosa et al.[24]	Kidney Cortex Microscope	MC (8)	236,386	165,466 / 23,640 / 47,280	CC BY 3.0
OrganAMNIST	Bilic et al.[25], Xu et al.[26]	Abdominal CT	MC (11)	58,850	34,581 / 6,491 / 17,778	CC BY 4.0
OrganCMNIST	Bilic et al.[25], Xu et al.[26]	Abdominal CT	MC (11)	23,660	13,000 / 2,392 / 8,268	CC BY 4.0
OrganSMNIST	Bilic et al.[25], Xu et al.[26]	Abdominal CT	MC (11)	25,221	13,940 / 2,452 / 8,829	CC BY 4.0
MedMNIST3D						
OrganMNIST3D	Bilic et al.[25], Xu et al.[26]	Abdominal CT	MC (11)	1,743	972 / 161 / 610	CC BY 4.0
NoduleMNIST3D	Armato et al.[27]	Chest CT	BC (2)	1,849	1,158 / 165 / 526	CC BY 3.0
AdrenalMNIST3D	New	Shape from Abdominal CT	BC (2)	1,584	1,188 / 98 / 298	CC BY 4.0
FractureMNIST3D	Jin et al.[28]	Chest CT	MC (3)	1,370	1,027 / 103 / 240	CC BY-NC 4.0
VesselMNIST3D	Yang et al.[29]	Shape from Brain MRA	BC (2)	1,909	1,335 / 192 / 382	CC0 1.0
SynapseMNIST3D	New	Electron Microscope	BC (2)	1,759	1,230 / 177 / 352	CC BY 4.0

- Upper: MedMNIST2D, 12 datasets of 2D images. Lower: MedMNIST3D, 6 datasets of 3D images. MC: Multi-Class. BC: Binary-Class. ML: Multi-Label. OR: Ordinal Regression

(https://medmnist.com)

What is ImageNet

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon MTurk
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC): 1.2 million training images, 1000 classes



http://www.image-net.org/

www.image-net.org/challenges/LSVRC/

ImageNet

- 14 million+ labeled images,
- 21k+ classes
- Images gathered from Internet
- Human labels via Amazon MTurk
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC): 1.2 million training images, 1000 classes

http://www.image-net.org/

www.image-net.org/challenges/LSVRC/

(Russakovsky etc. ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015)

What is ImageNet (http://www.image-net.org/)

A project which aims to provide a large image database for research purposes. It contains more than 14 million images which belong to more than 20,000 classes (or synsets). They also provide bounding box annotations for around 1 million images, which can be used in Object Localization tasks. It should be noted that they only provide urls of images and you need to download those images.

Since 2010, the ImageNet project runs an annual software contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where software programs compete to correctly classify and detect objects and scenes. The training data is a subset of ImageNet with 1.2 million images belonging to 1000 classes. Deep Learning came to limelight in 2012 when Alex Krizhevsky and his team won the competition by a margin of a whooping 11%. ILSVRC and Imagenet are sometimes used interchangeably.

Stanford Tiny-Imagenet-200

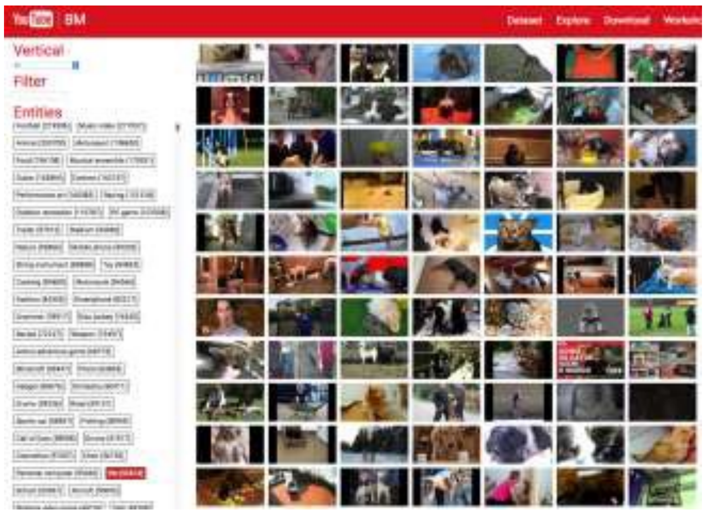
- fastAI: Imagenette and Imagewoof datasets, https://github.com/fastai/imagenette/
  1. Imagenette is a subset of 10 easily classified classes from Imagenet (tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute)
  2. Imagewoof is a subset of 10 classes from Imagenet that aren't so easy to classify, since they're all dog breeds. The breeds are: Australian terrier, Border terrier, Samoyed, Beagle, Shih-Tzu, English foxhound, Rhodesian ridgeback, Dingo, Golden retriever, Old English sheepdog.

- https://github.com/rmccorm4/Tiny-Imagenet-200

- Downlaod: http://cs231n.stanford.edu/tiny-imagenet-200.zip
  - Tiny Imagenet has 200 classes.
    - a training dataset of 100,000 images,
    - a validation dataset of 10,000 images, and
    - a test dataset of 10,000 images.
    - All images are of size 64×64.
  - Each class has 500 training images, 50 validation images, and 50 test images.

6/29/2020: MIT takes down 80 Million Tiny Images data set due to racist and offensive content!
https://groups.csail.mit.edu/vision/TinyImages/

YouTube-8M Dataset



- 7 Million Video URLs
- 450K+ Hours
- 4700+ Classes
- 3.2 Billion Audio / Visual Features
- 3.4 Avg. Labels / Video

(https://research.google.com/youtube8m/index.html)

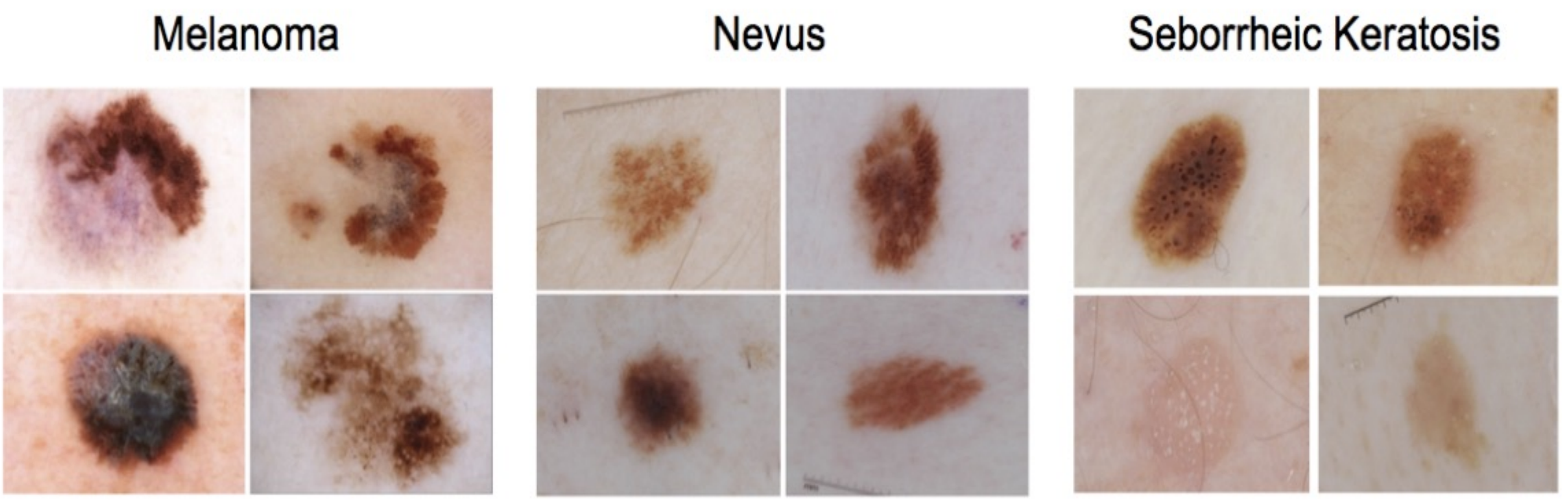
Abu-El-Haija, Sami, et al. "YouTube-8M: A large-scale video classification benchmark." arXiv preprint arXiv:1609.08675 (2016).

ISIC Archive - Skin Cancer Prediction

- ISIC (The International Skin Imaging Collaboration) is a Skin Cancer Research dataset. https://isic-archive.com/
- The ISIC Archive contains over 13,000 dermoscopic images.
- ISIC 2017: Skin Lesion Analysis Towards Melanoma Detection

https://challenge.kitware.com/#phase/5840f53ccad3a51cc66c8dab

- 2000 images are provided as training data, including 374 "melanoma", 254 "seborrheic keratosis", and the remainder as benign nevi (1372)



COVID-19 Data Science Resources

- **COVID-19 Data Science Resources** [comprehensive]: datasets, analytic tools, articles collection, challenges, funding, opportunities, data visualizations, computing resources, supports
  https://www.academicdatascience.org/covid

- **Image Data: CT and X-Ray**
  COVID-CT-Dataset: https://github.com/UCSD-AI4H/COVID-CT
  COVID-19 cases with chest X-ray or CT images
  https://github.com/ieee8023/covid-chestxray-dataset

- **Time Series Data:** daily information on the number of cases, deaths, and recoveries from across different regions, including time-stamps
  JHU CSSE
  https://github.com/CSSEGISandData/COVID-19
  COVID-19 in US and CANADA
  https://coronavirus.1point3acres.com/?from=groupmessage&isappinstalled=0
- **NLP**
  COVID-19 Open Research Dataset
  https://pages.semanticscholar.org/coronavirus-research

Case Studies

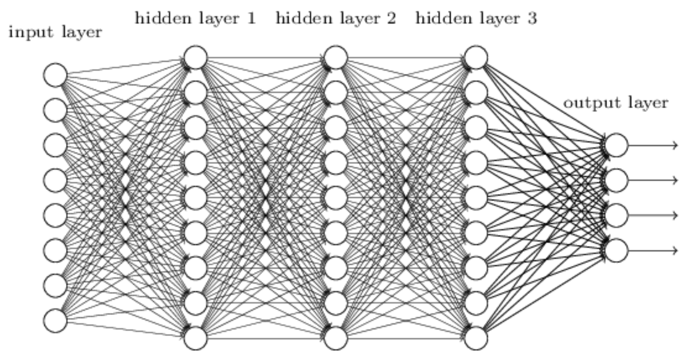
Classic network architectures

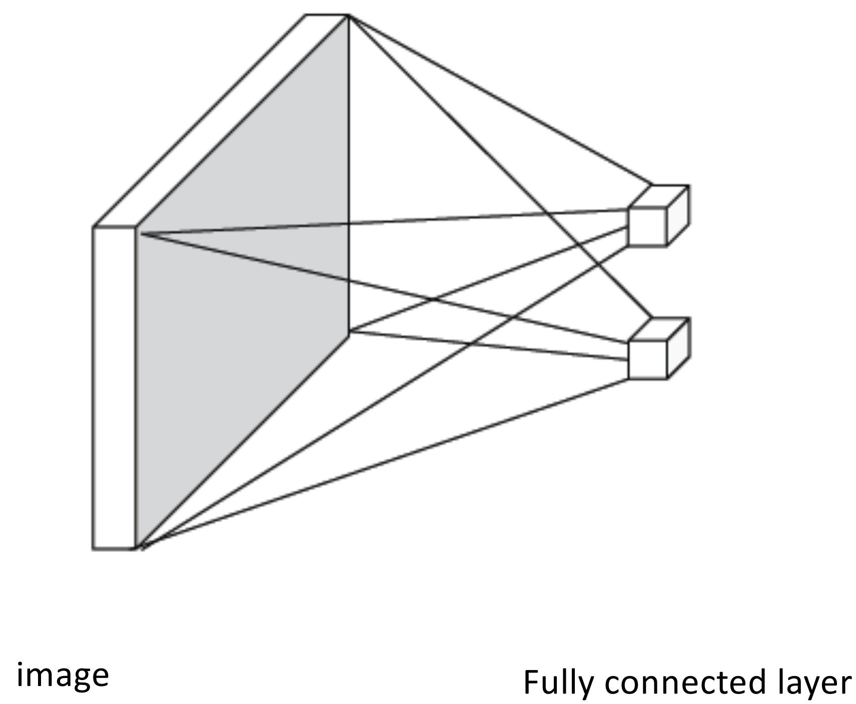
- LeNet. The first successful applications of Convolutional Networks were developed by Yann LeCun in 1990's.
- AlexNet. The first work that popularized Convolutional Networks in Computer Vision was the AlexNet, developed by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton. The AlexNet was submitted to the ImageNet ILSVRC challenge in 2012 and significantly outperformed the second runner-up (top 5 error of 16% compared to runner-up with 26% error). The Network had a very similar architecture to LeNet, but was deeper, bigger, and featured Convolutional Layers stacked on top of each other.
- ZF Net. The ILSVRC 2013 winner was a Convolutional Network from Matthew Zeiler and Rob Fergus. It was an improvement on AlexNet by tweaking the architecture hyperparameters.
- VGGNet. The runner-up in ILSVRC 2014 was the network from Karen Simonyan and Andrew Zisserman that became known as the VGGNet. Its main contribution was in showing that the depth of the network is a critical component for good performance. Their final best network contains 16 CONV/FC layers and, appealingly, features an extremely homogeneous architecture that only performs 3x3 convolutions and 2x2 pooling from the beginning to the end.

Modern network architectures

- GoogLeNet (Inception). The ILSVRC 2014 winner was a Convolutional Network from Szegedy et al. from Google. Its main contribution was the development of an Inception Module that dramatically reduced the number of parameters in the network (4M, compared to AlexNet with 60M). Additionally, this paper uses Average Pooling instead of Fully Connected layers at the top of the ConvNet, eliminating a large amount of parameters that do not seem to matter much.
- ResNet. Residual Network developed by Kaiming He et al. was the winner of ILSVRC 2015. It features special skip connections and a heavy use of batch normalization. ResNets are currently by far state of the art Convolutional Neural Network models and are a common choice for using ConvNets in practice.
- DenseNet. Dense Convolutional Network by Huang et al (CVPR 2017, Best Paper Award). Each layer is receiving a "collective knowledge" from all preceding layers.
- OctConv. Octave Convolution is a simple replacement for the traditional convolution operation that gets better accuracy with fewer FLOPs (4/10/2019)

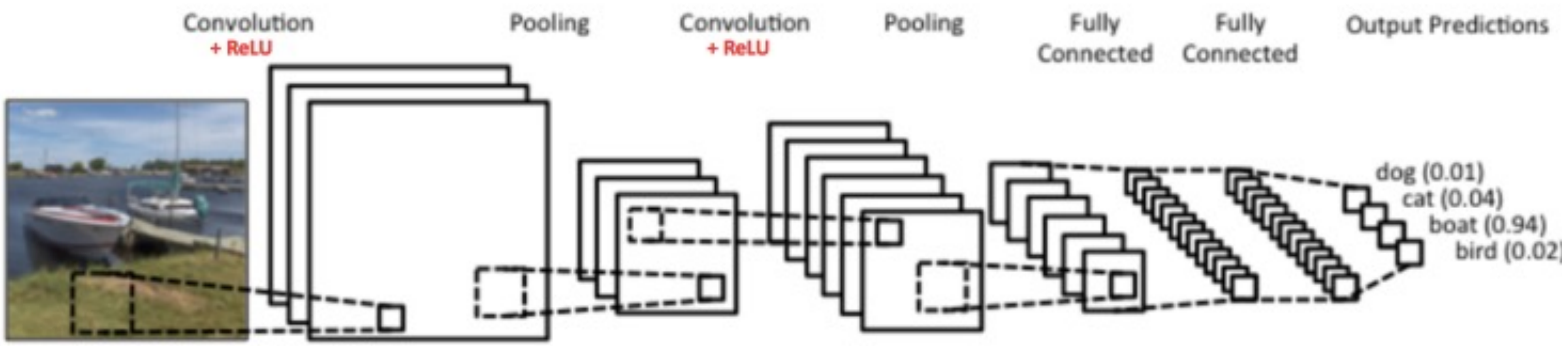
From fully connected to convolutional networks





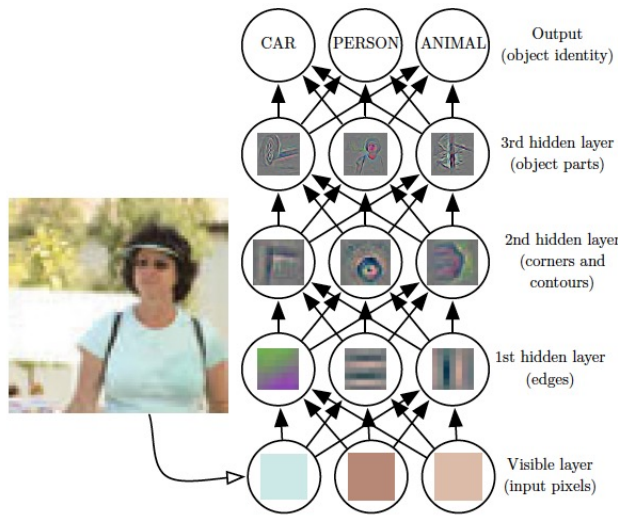
What Are Convolutional Neural Networks (CNN)

A convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that explicitly assumes that the inputs are images, which allows us to encode certain properties into the architecture.



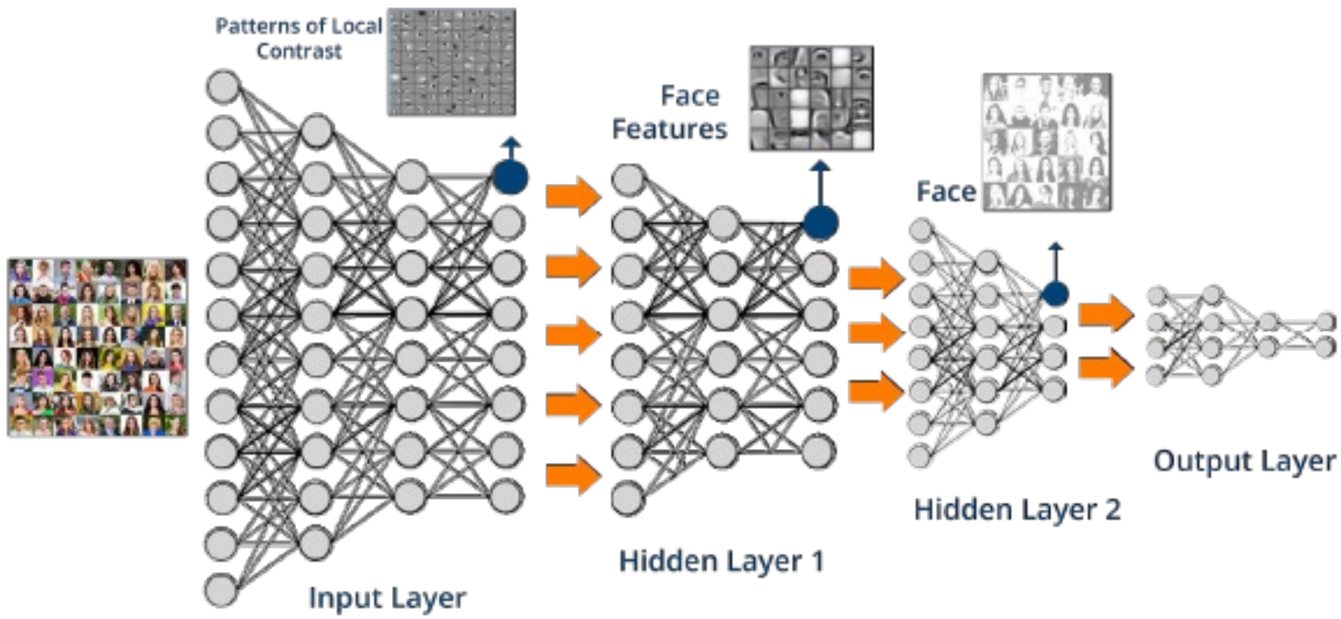
- Convolution
- Non Linearity (ReLU)
- Pooling or Sub Sampling
- Classification (Fully Connected Layer)

Deep Learning: Repeated Composition along Depth



(Goodfellow et al. 2016)

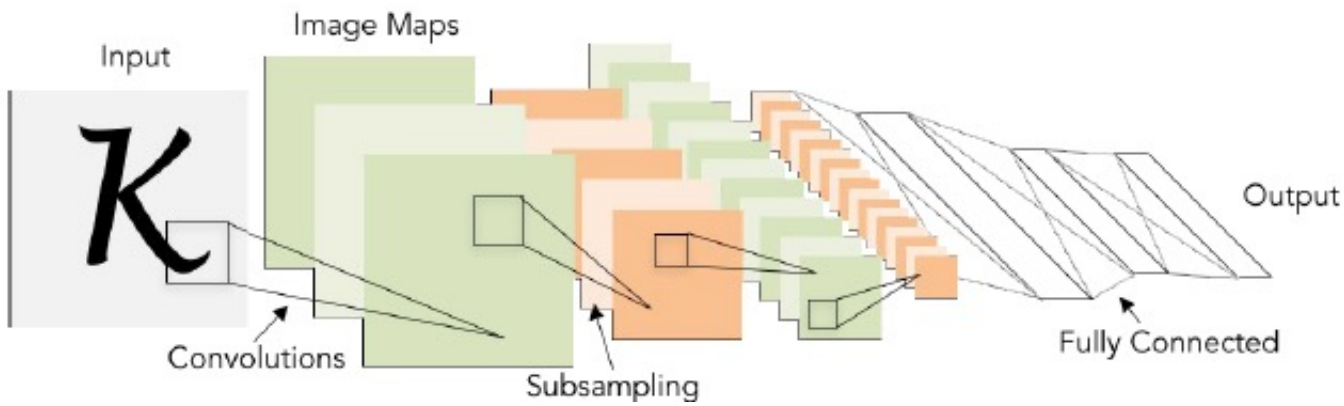
Deep Learning for Facial Recognition



(www.edureka.co)

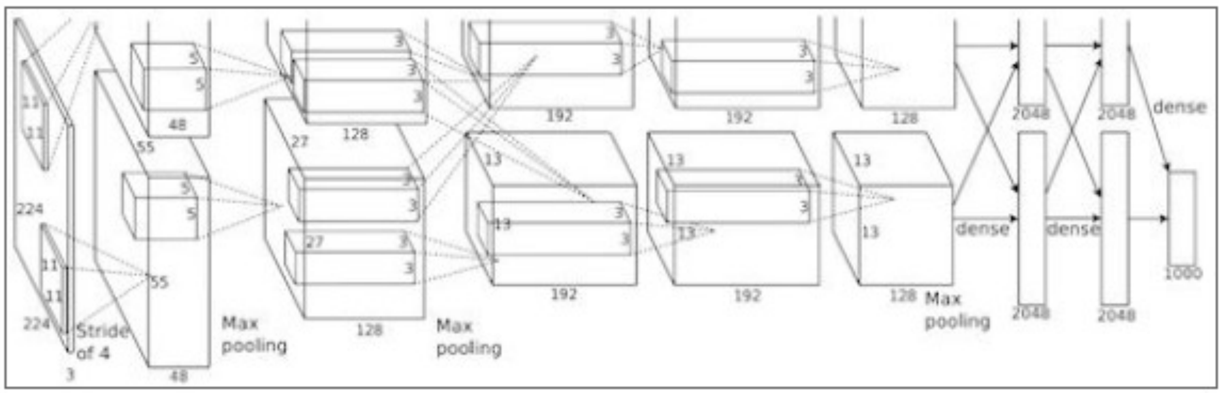
Convolutional Neural Networks (Convnets): the early day

LeNet-5



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

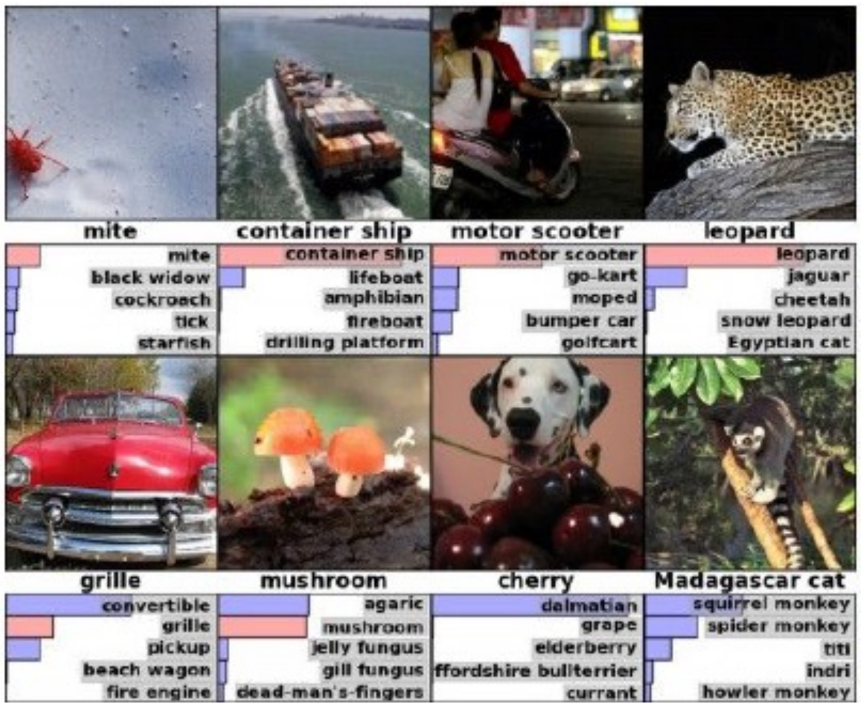
AlexNet



ImageNet Classification with Deep Convolutional Neural Networks [Krizhevsky, Sutskever, Hinton, 2012]

Fast-forward to today: ConvNets are everywhere

Classification



Retrieval

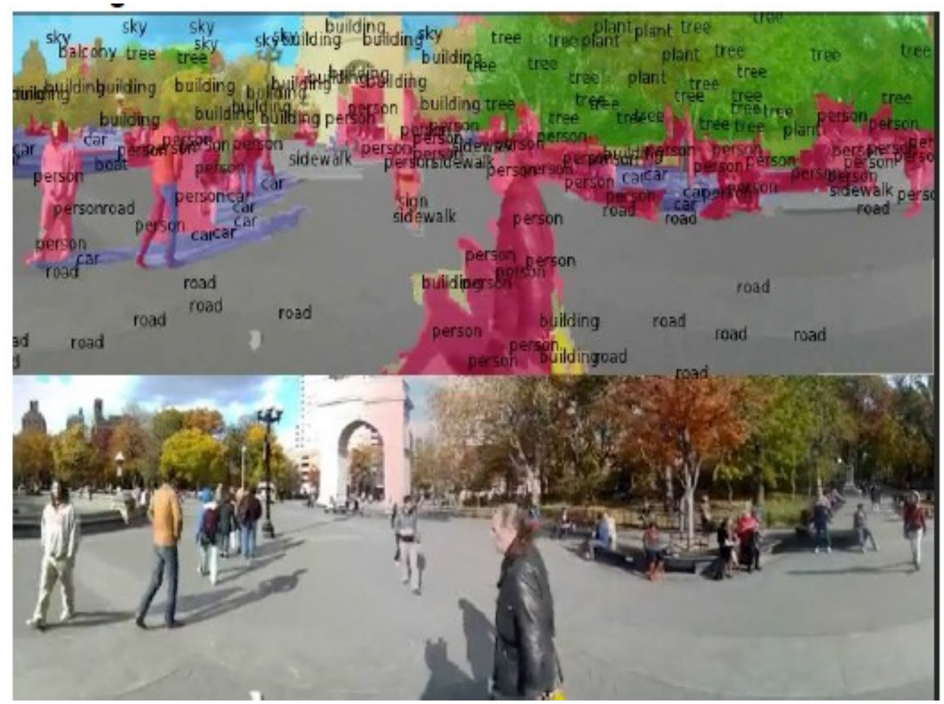
Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012.

Fast-forward to today: ConvNets are everywhere

Detection

[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



[Farabet et al., 2012]

What happed to the field?

Classification: ImageNet Challenge top-5 error

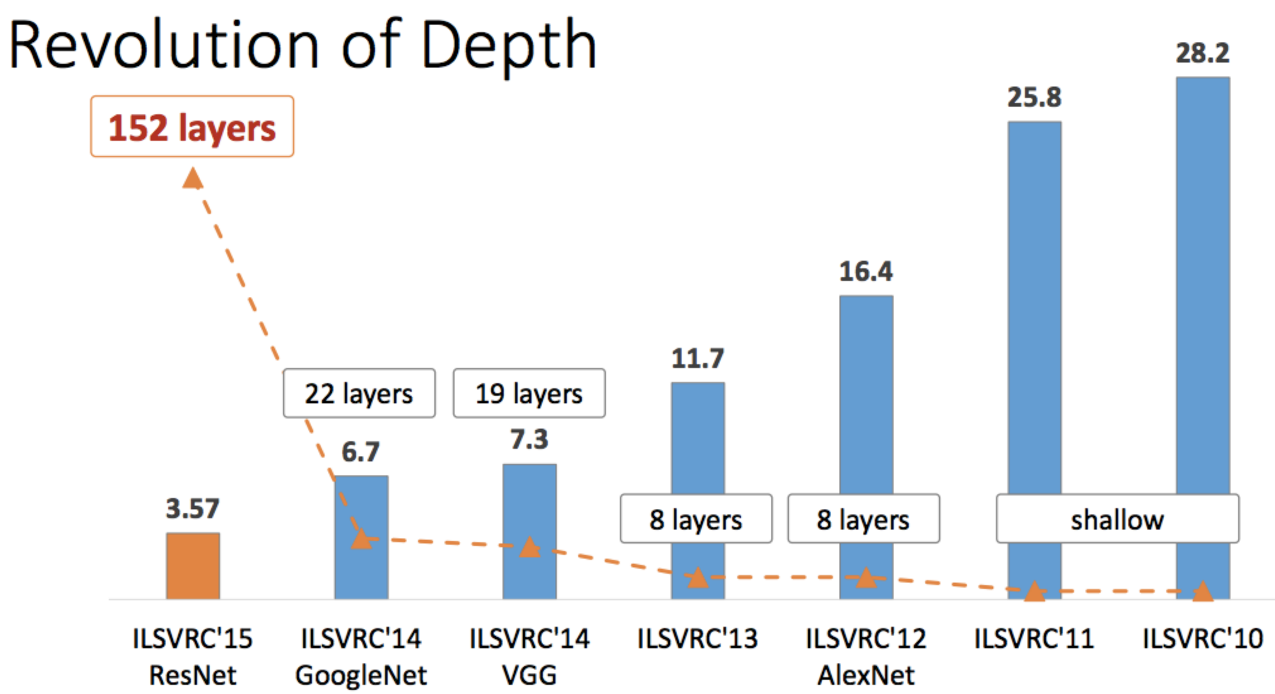
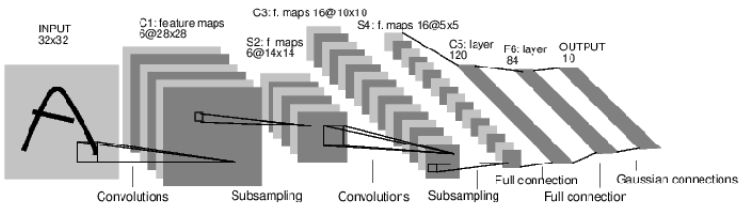
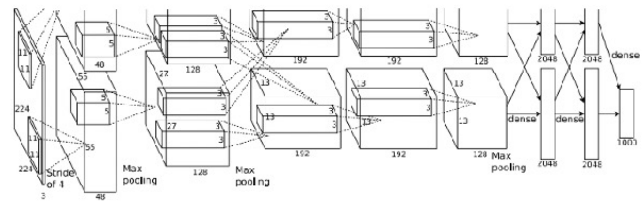


Figure source: Kaiming He

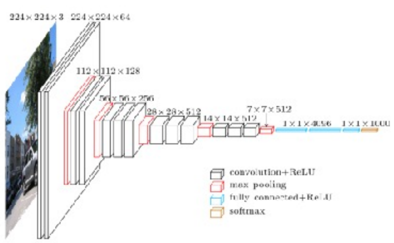
Convolutional Neural Networks



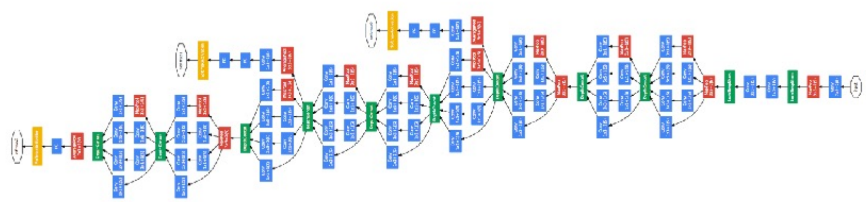
LeNet



AlexNet

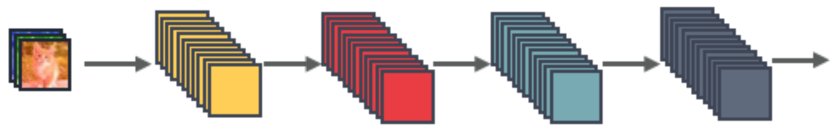


VGG

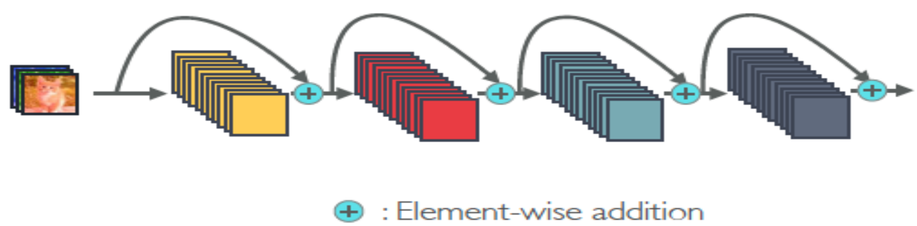


Inception

ResNet



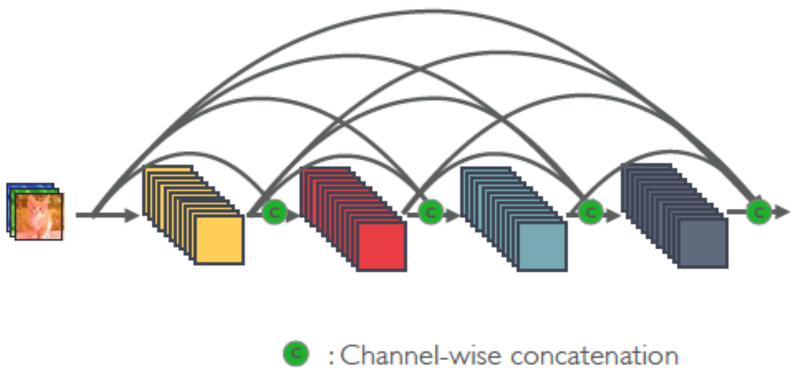
Standard Connectivity



ResNet Connectivity: identity mappings promote gradient propagation.

He, Zhang, Ren, Sun, Deep Residual Learning for Image Recognition, CVPR 2015

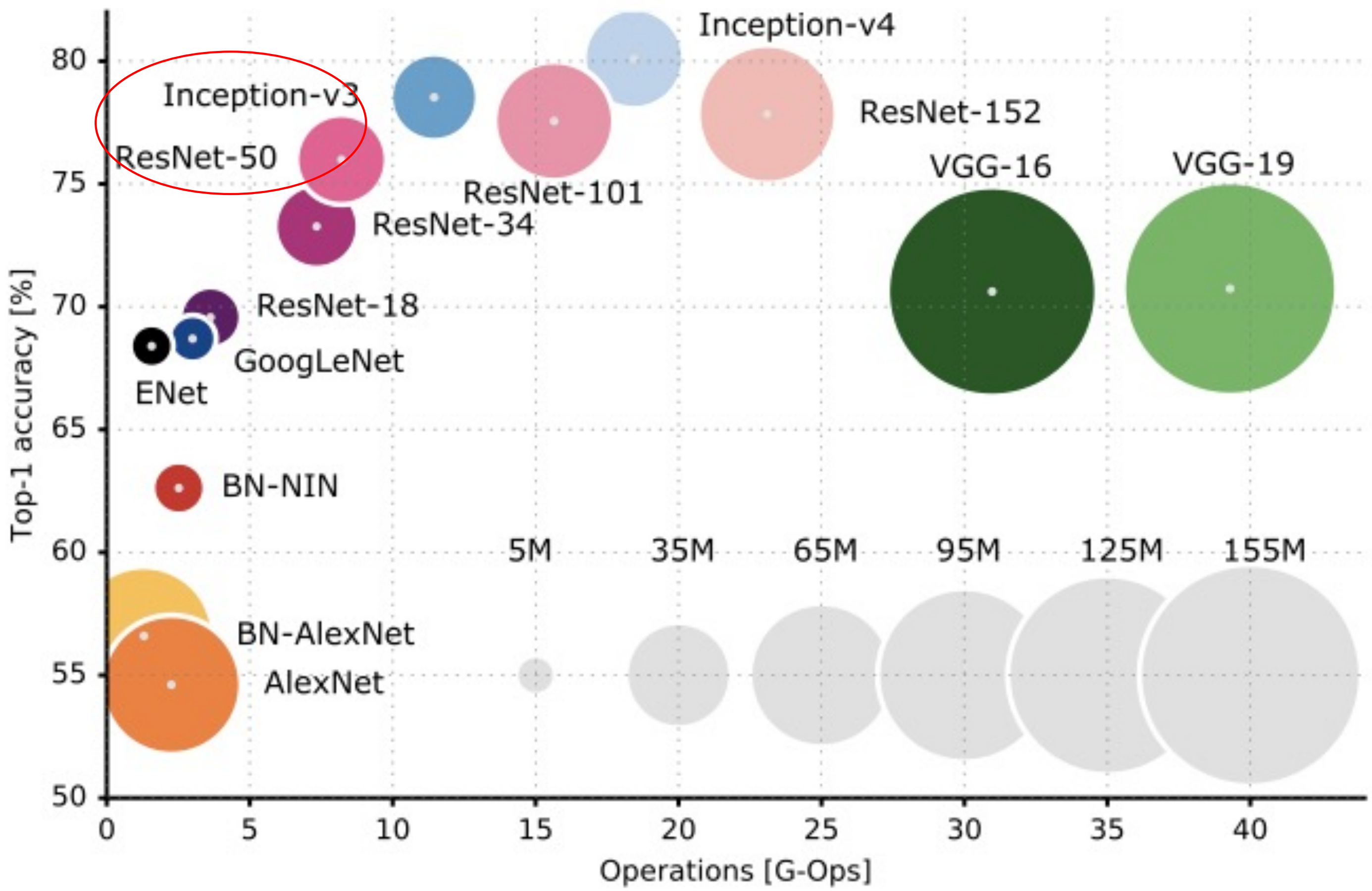
Dense Connectivity (DenseNet)



Each layer is receiving a “collective knowledge” from all preceding layers.

Huang et al, Densely Connected Convolutional Networks, *CVPR 2017*

Accuracy vs. efficiency



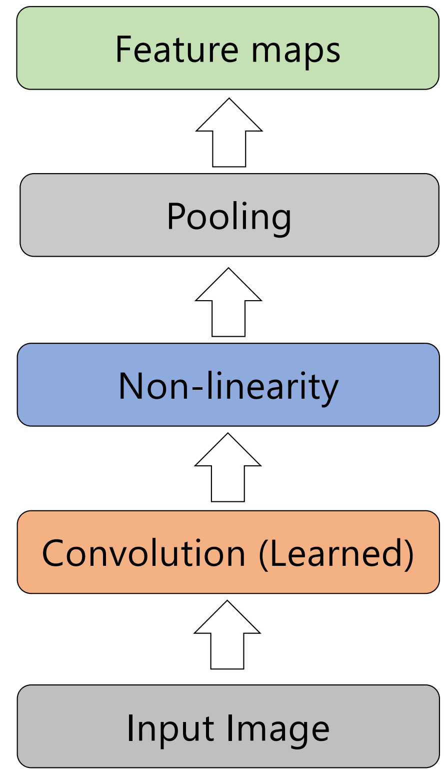
https://towardsdatascience.com/neural-network-architectures-156e5bad51ba

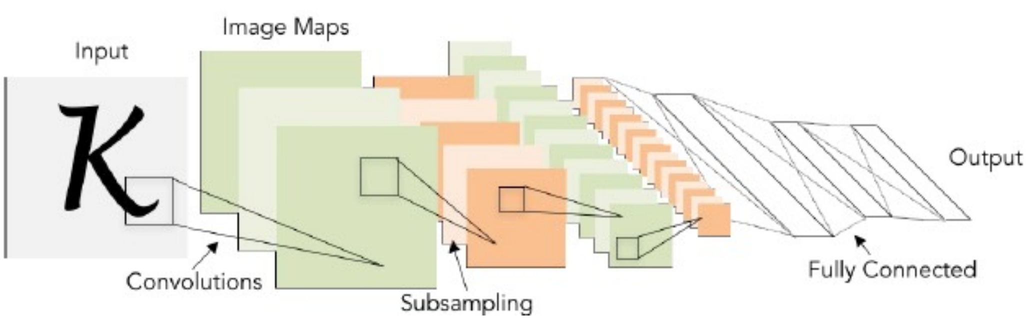
What Are Convolutional Neural Networks?
(4:45 min, matlab)

https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html

Overview of Convnets

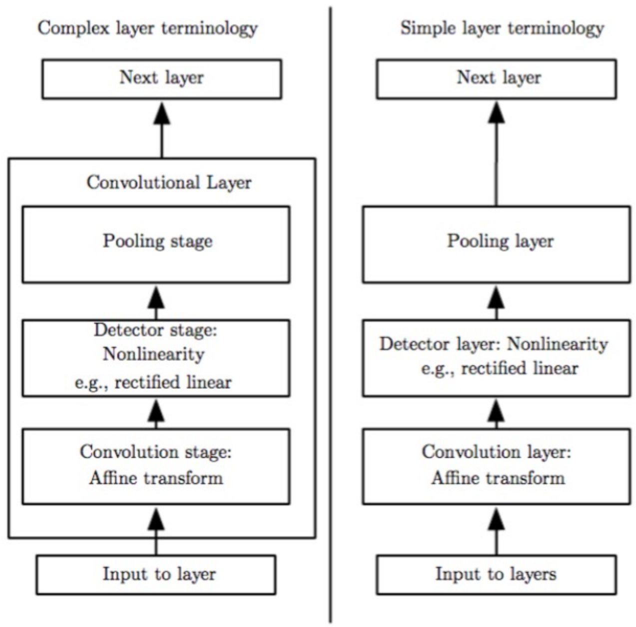
- Feed-forward:
  - Convolve input
  - Non-linearity (rectified linear)
  - Pooling (local max)
- Supervised
- Train convolutional filters by back-propagating classification error





LeCun et al. 1998

Three Stages of a Convolutional Layer



1. Convolution stage
2. Nonlinearity: a nonlinear transform such as rectified linear or tanh
3. Pooling: output a summary statistics of local input, such as max pooling and average pooling

The problem: an image is a matrix of pixel values

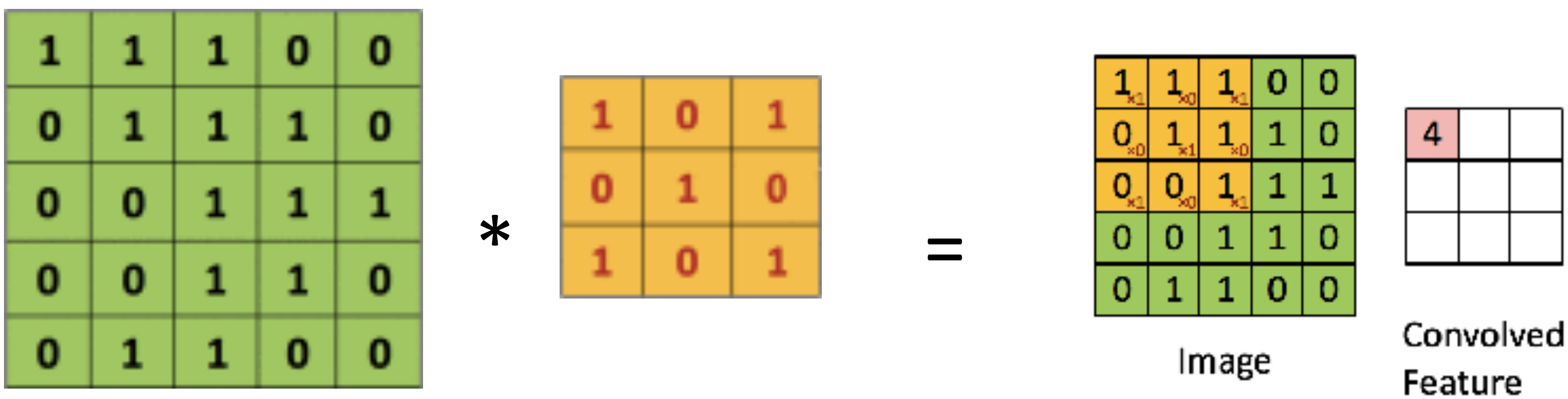
What We See

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

What Computers See

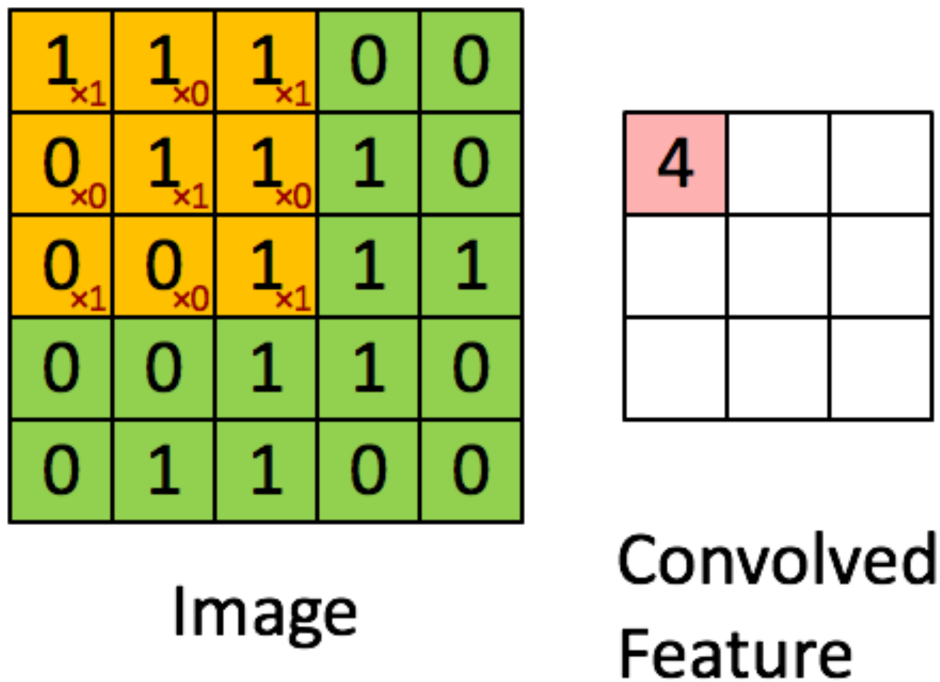
What is Convolution?

Convolution operation



5x5 image

3x3 kernal

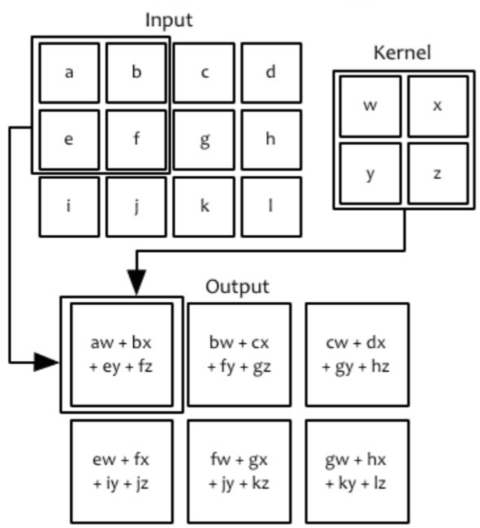


http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

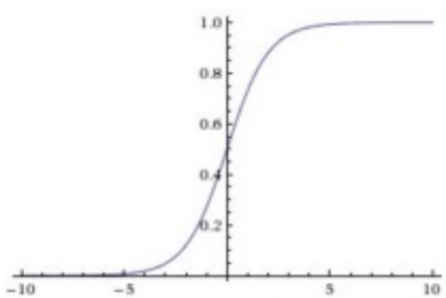
Convolution Operation in CNN

- Input: an image (2-D array) x
- Convolution kernel/operator(2-D array of learnable parameters): w
- Feature map (2-D array of processed data): s
- Convolution operation in 2-D domains:

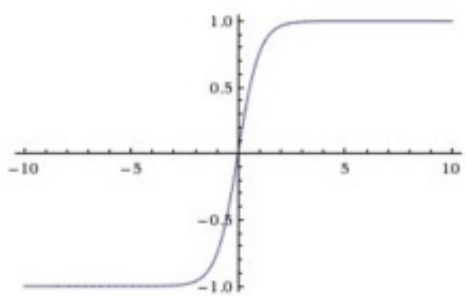
$$s[i,j] = (x * w)[i,j] = \sum_{m=-M}^{M} \sum_{n=-N}^{N} x[i+m, j+n]\, w[m,n]$$



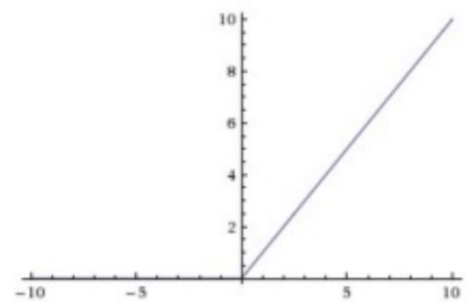
Non Linearity: ReLU - Rectified Linear Unit.



Sigmoid



tanh



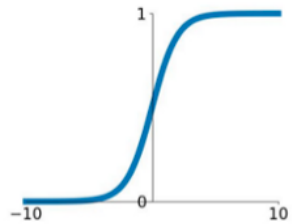
ReLU

- **Sigmoid:** $\sigma(x) = 1 / (1 + \exp(-x))$, [0 1]
- **Tanh:** $\tanh(x) = 2\sigma(2x) - 1$, [-1 1]
- **ReLU:** $f(x) = \max(0, x)$

Activation Functions

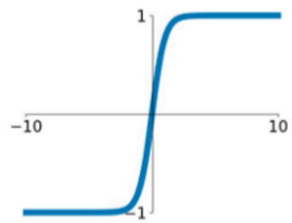
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



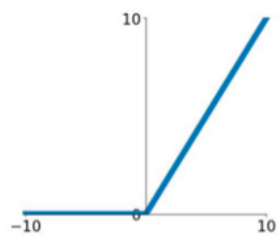
tanh

$\tanh(x)$



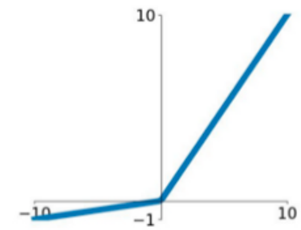
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

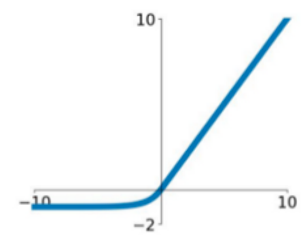


Maxout

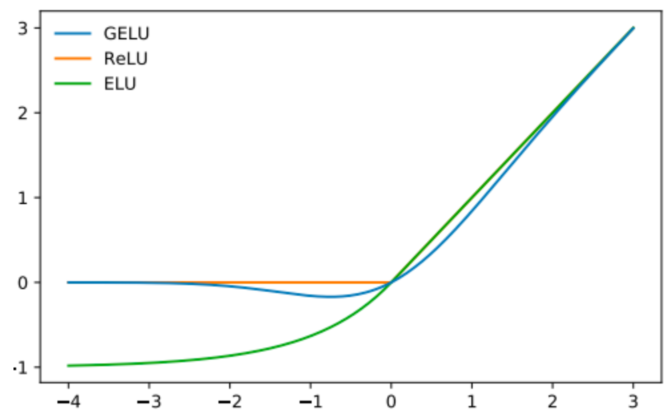
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Gaussian Error Linear Units (GELUs)



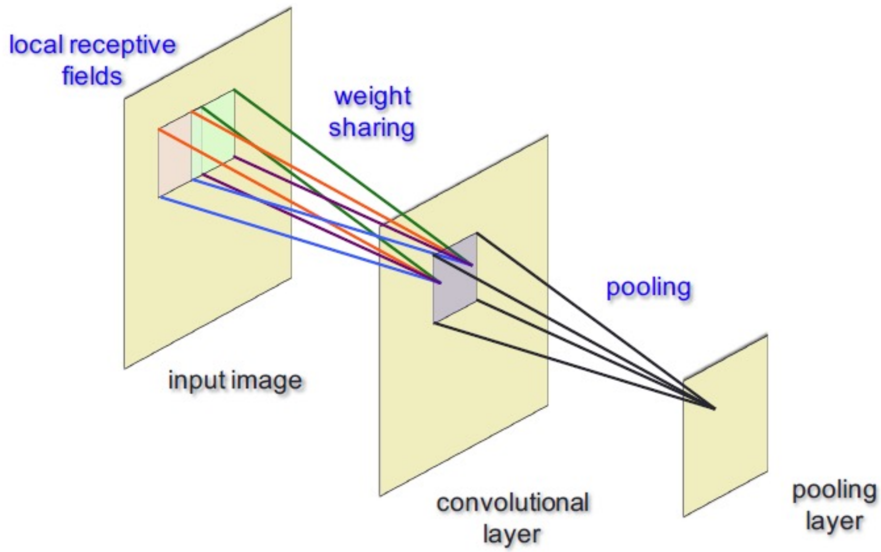
- Activations like ReLU, ELU and PReLU have enabled faster and better convergence of NN than sigmoids.
- Dropout regularizes the model by randomly multiplying a few activations by 0
- GELU combines both
- Used in the most recent Transformers – Google's BERT and OpenAI's GPT-2

$$\mathrm{GELU}(x) = xP(X \leq x) = x\Phi(x)$$
$$\approx 0.5x\left(1 + \tanh\left[\sqrt{2/\pi}\left(x + 0.044715x^3\right)\right]\right)$$

https://arxiv.org/pdf/1606.08415v3.pdf

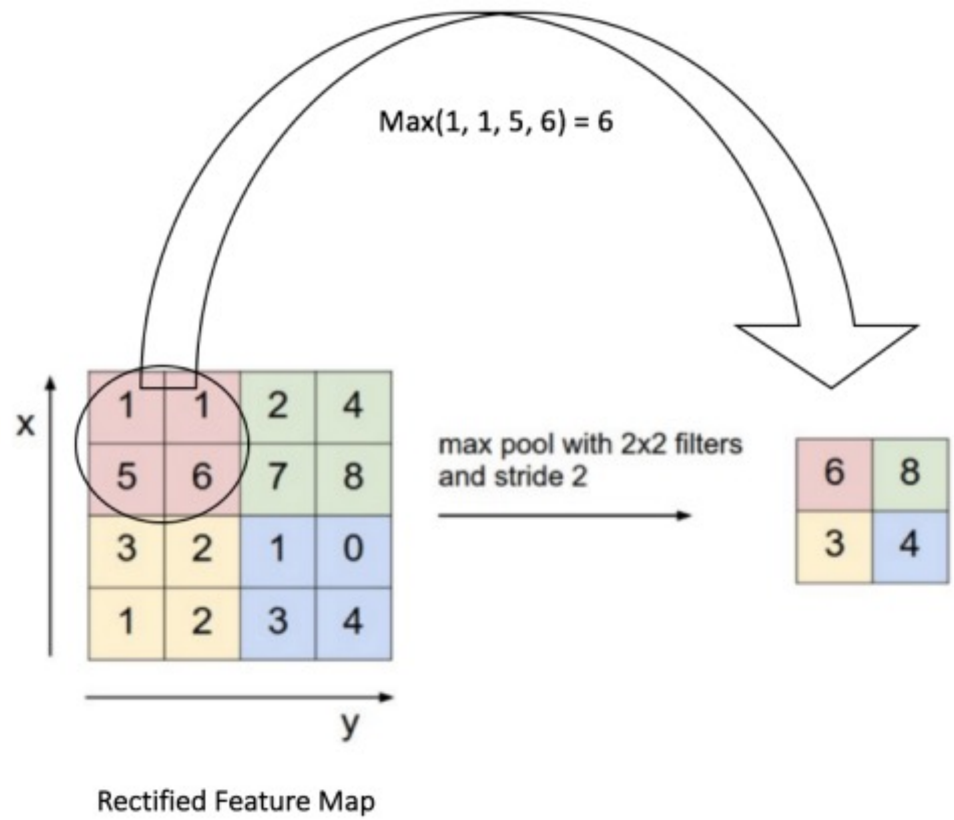
Pooling

- Common pooling operations:
  - Max pooling: reports the maximum output within a rectangular neighborhood.
  - Average pooling: reports the average output of a rectangular neighborhood (possibly weighted by the distance from the central pixel).



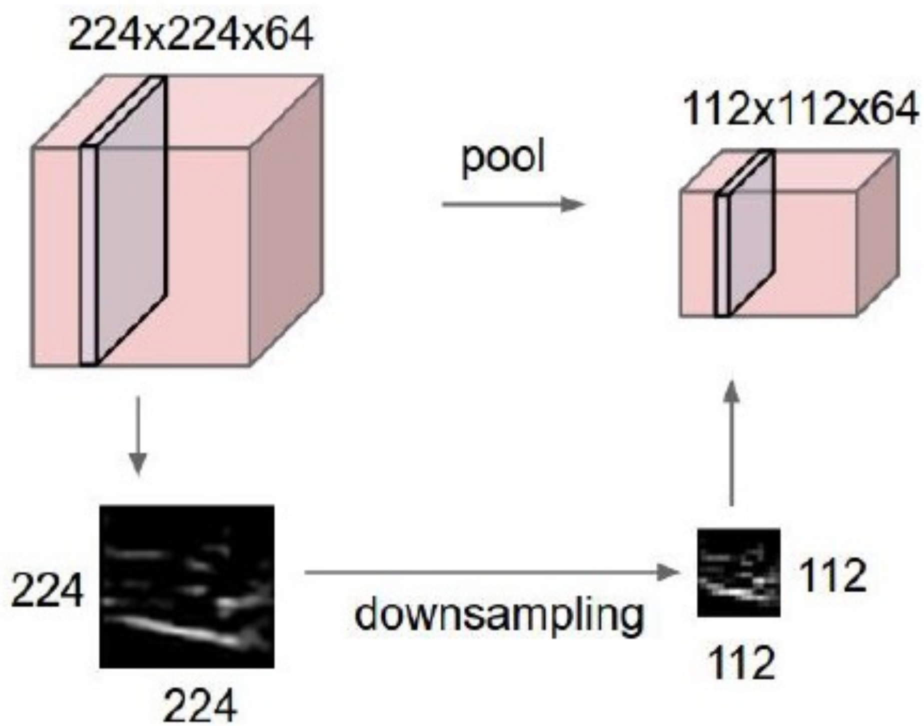
control overfitting
Using pooling to counteract *co-adapting* — depending too much on a few inputs

Pooling or Sub Sampling



Pooling or Sub Sampling

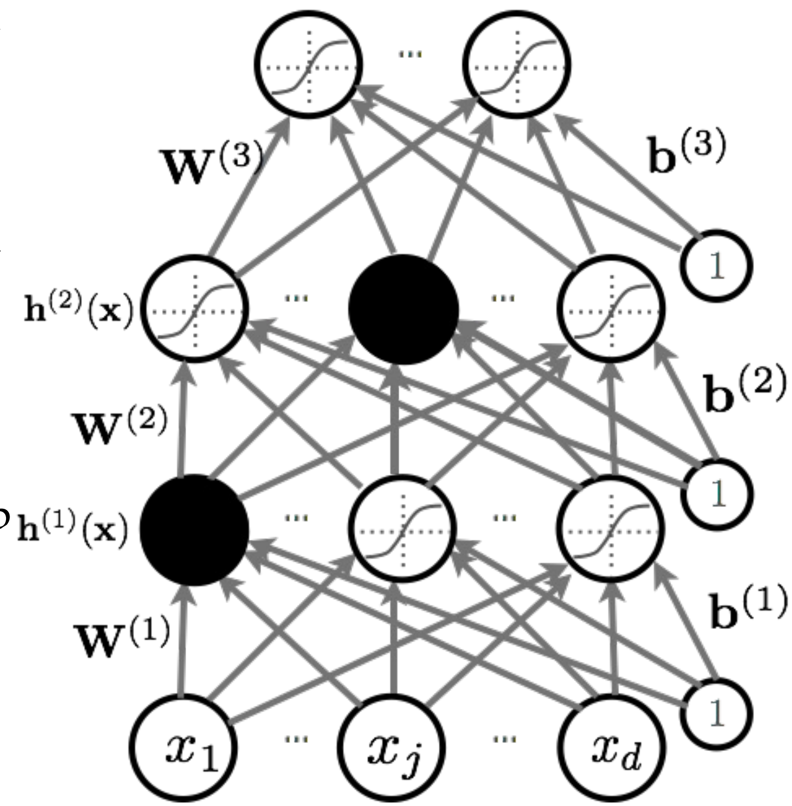
- makes the representations smaller and more manageable
- operates over each activation map independently:



Dropout: control overfitting

The idea: Cripple neural network by removing hidden units stochastically

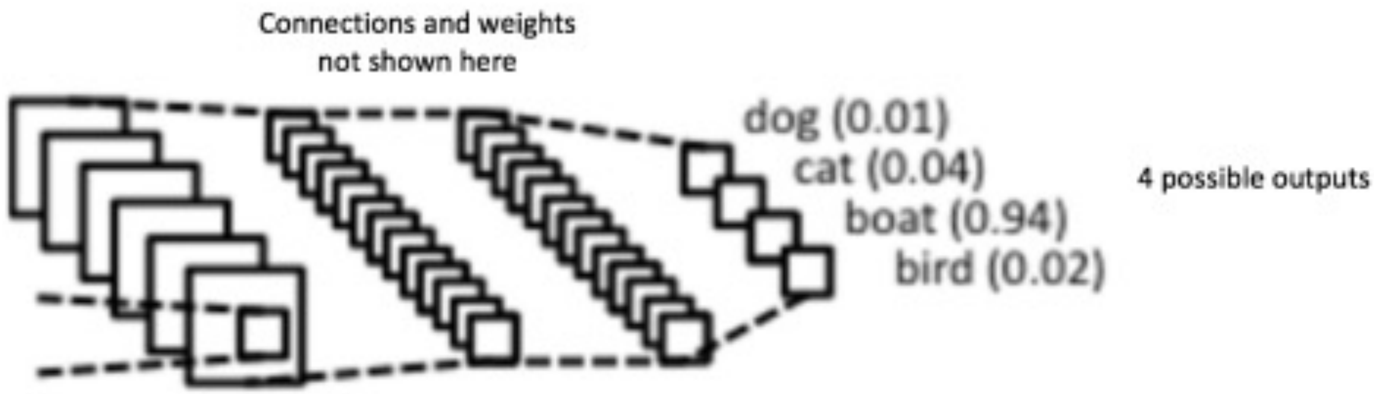
- a dropout layer is often paired with a pooling layer.
- each hidden unit is randomly set to 0 with probability 0.5; Could use a different dropout probability, but 0.5 usually works well
- hidden units cannot co-adapt to other units; neurons are less likely to be influenced too much by neighboring neurons, because any of them might drop out of the network at random.
- This makes the network less sensitive to small variations in the input, so more likely to generalize to new inputs.



Flattened

- The weights from the convolutional layers must be made 1-dimensional — flattened — before passing them to the fully connected Dense layer.
- The output shape of the previous layer is (2, 2, 128), so the output of Flatten() is an array with 512 elements.

Fully Connected Layer

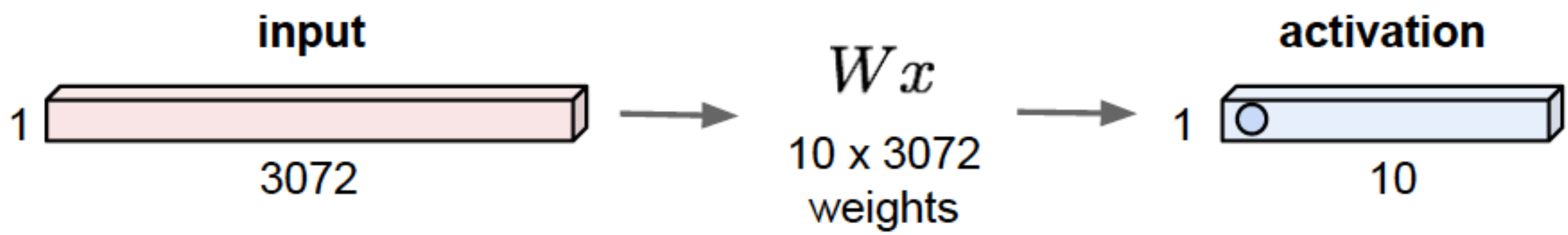


The sum of output probabilities from the Fully Connected Layer is 1.

Put things together

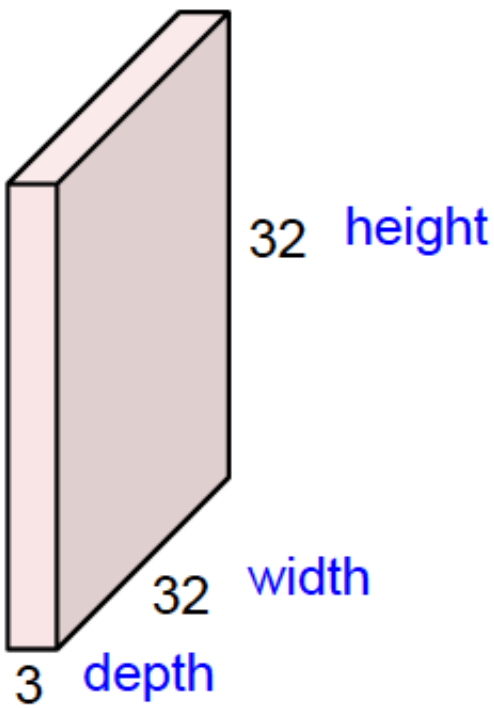
Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1

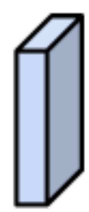


Convolutional Layer

32x32x3 image -> preserve spatial structure



5x5x3 filter

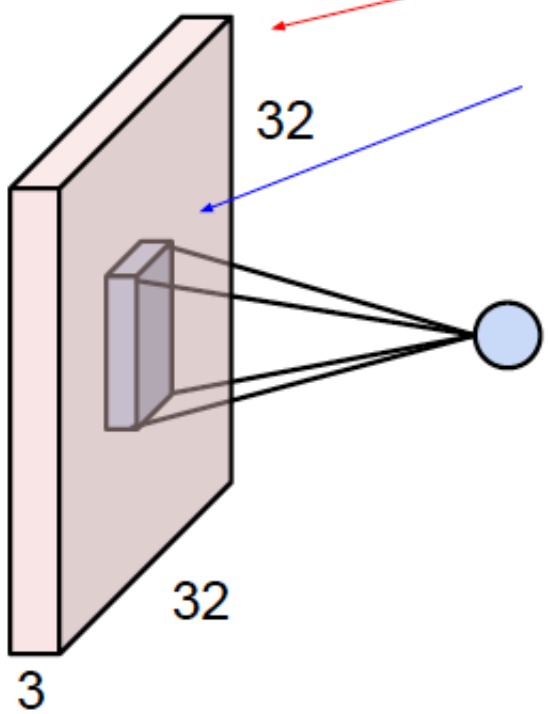


Convolve the filter with the image
i.e. "slide over the image spatially,
computing dot products"

Convolutional Layer

32x32x3 image

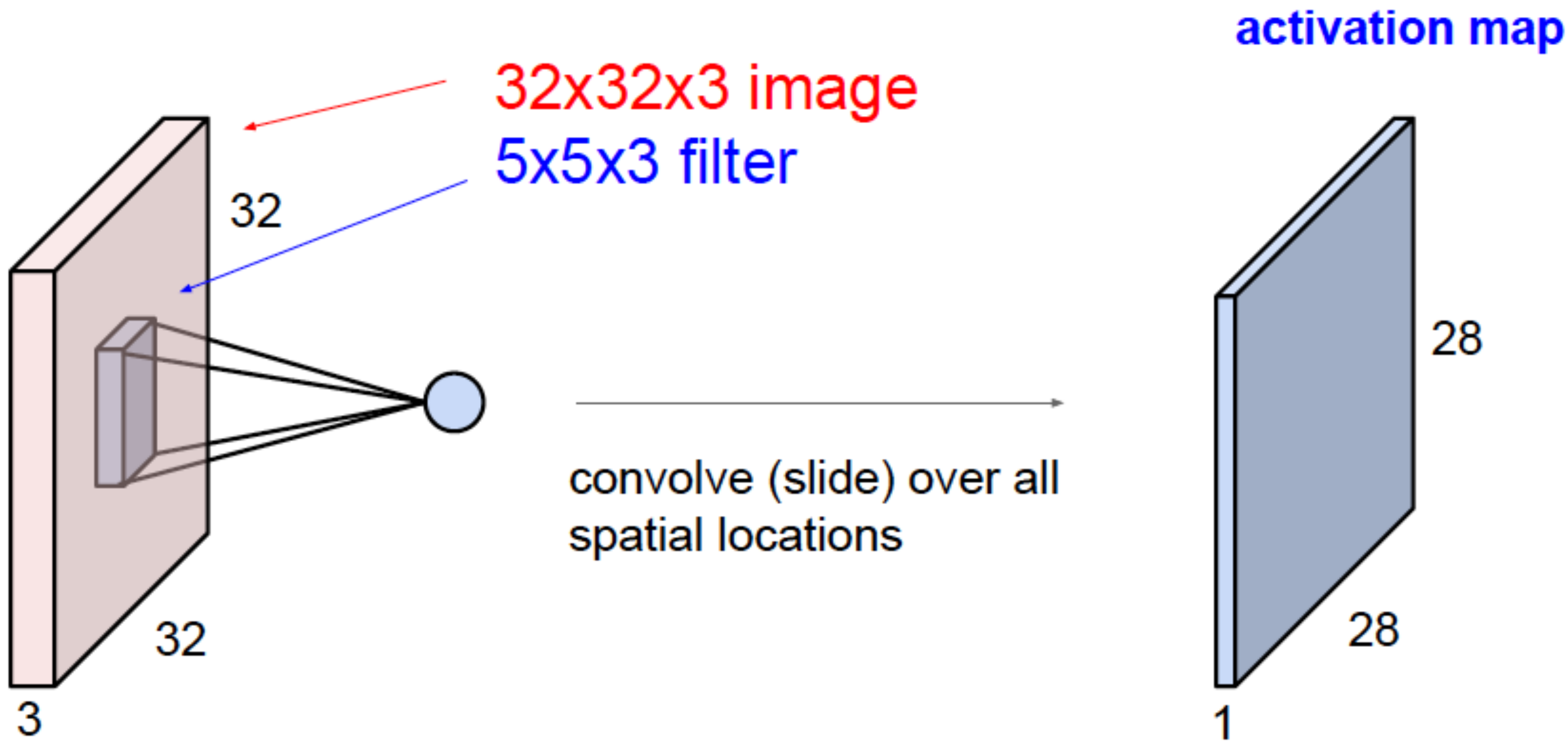
5x5x3 filter w



1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

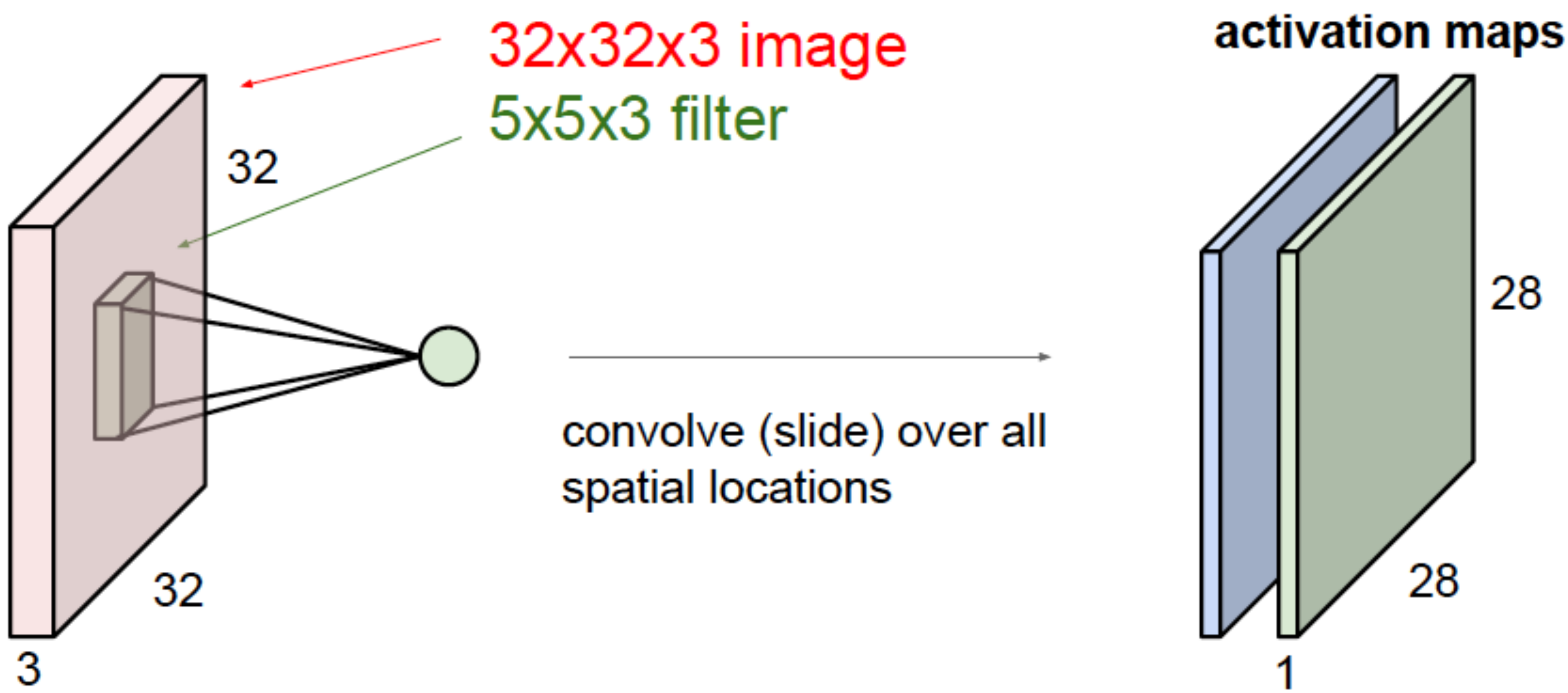
$$w^T x + b$$

Convolutional Layer

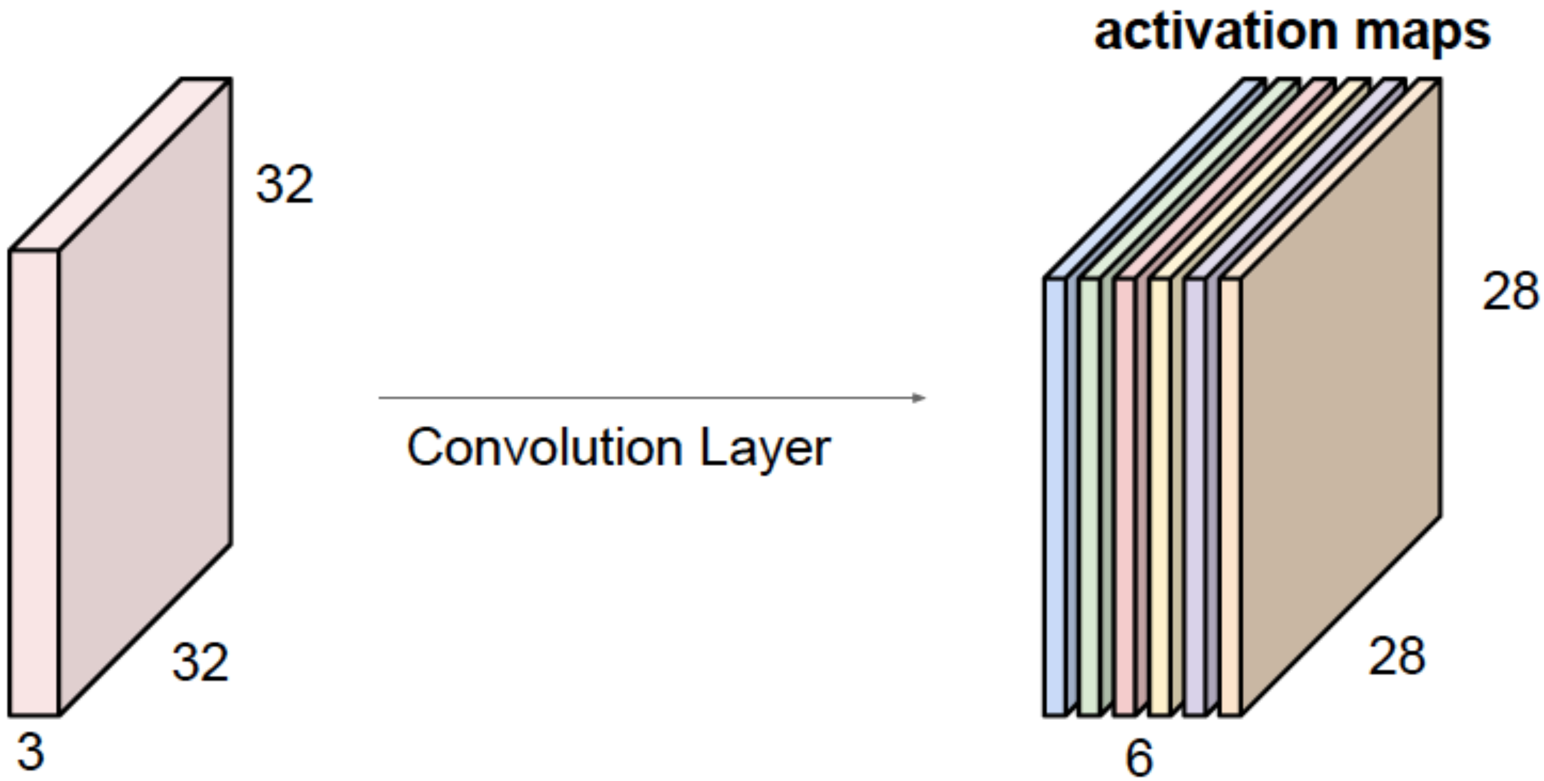


Convolutional Layer

consider a second, green filter

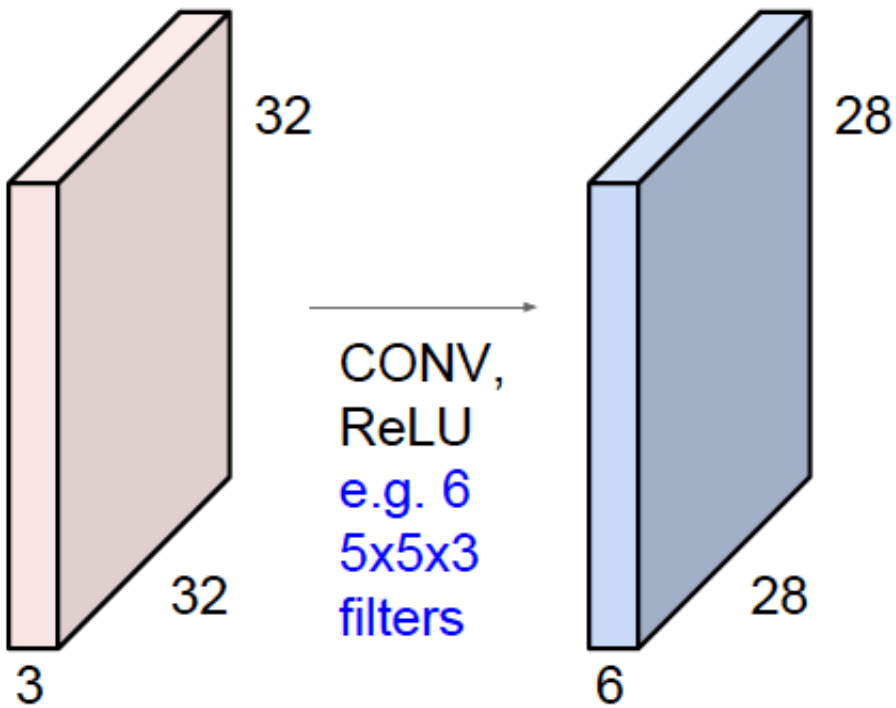


For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

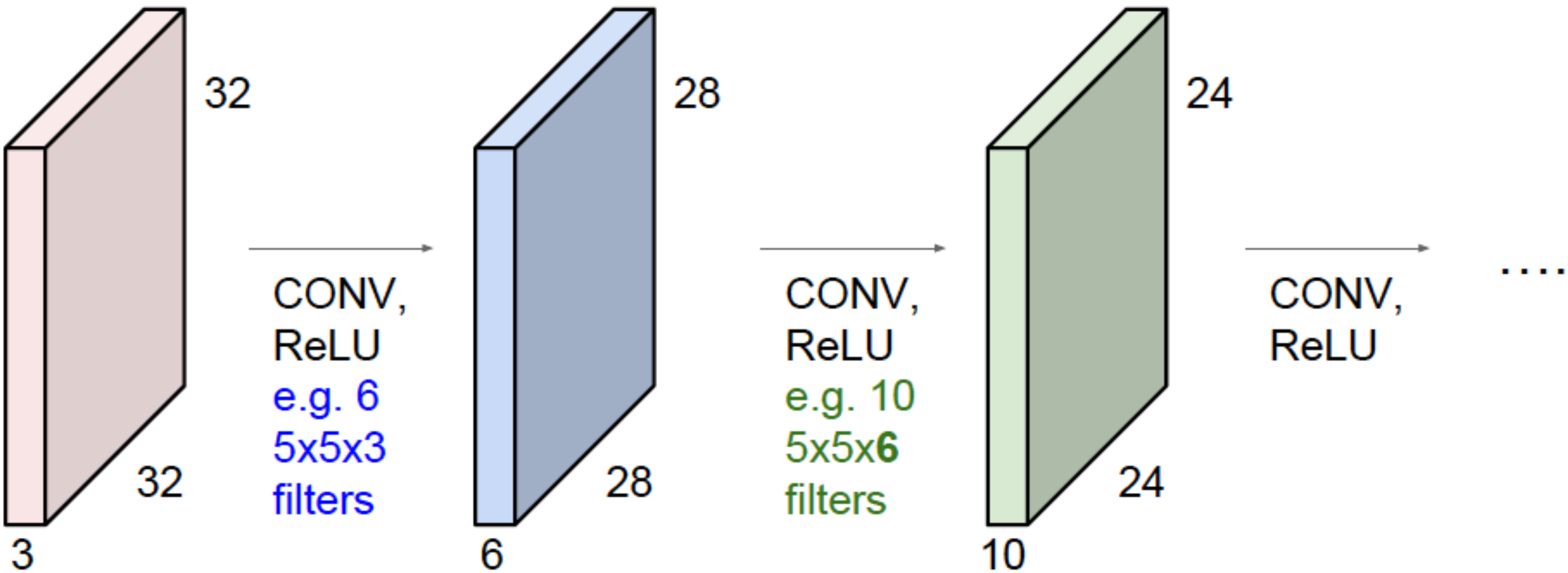


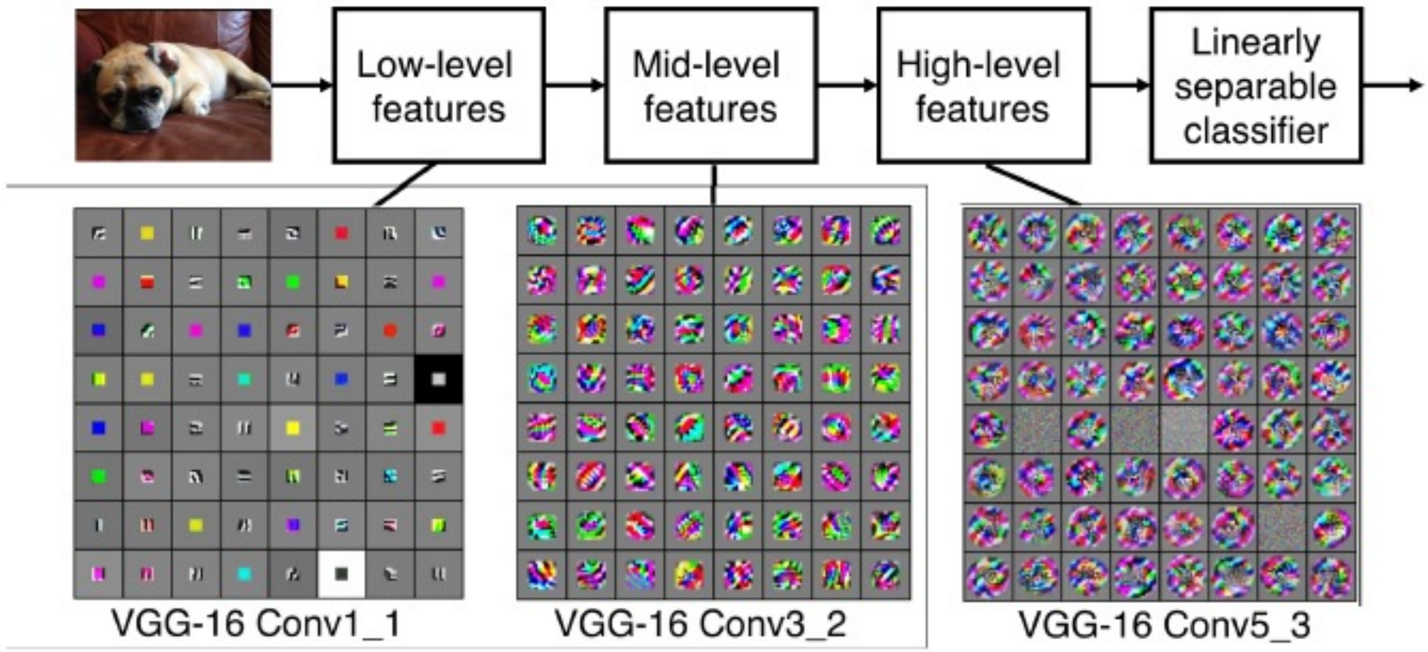
We stack these up to get a "new image" of size 28x28x6!

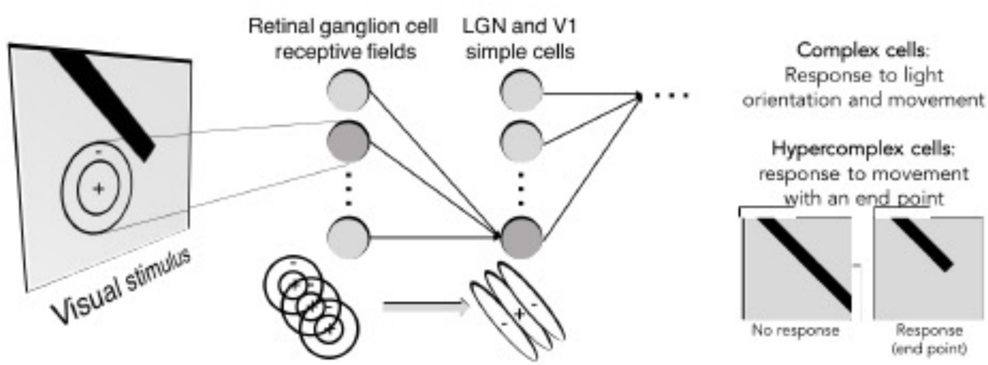
ConvNet is a sequence of Convolution Layers, interspersed with activation functions



ConvNet is a sequence of Convolution Layers, interspersed with




Low-level features
Mid-level features
High-level features
Linearly separable classifier
VGG-16 Conv1_1
VGG-16 Conv3_2
VGG-16 Conv5_3


Retinal ganglion cell receptive fields
LGN and V1 simple cells
Complex cells:
Response to light orientation and movement
Hypercomplex cells:
response to movement with an end point
Visual stimulus
No response
Response (end point)

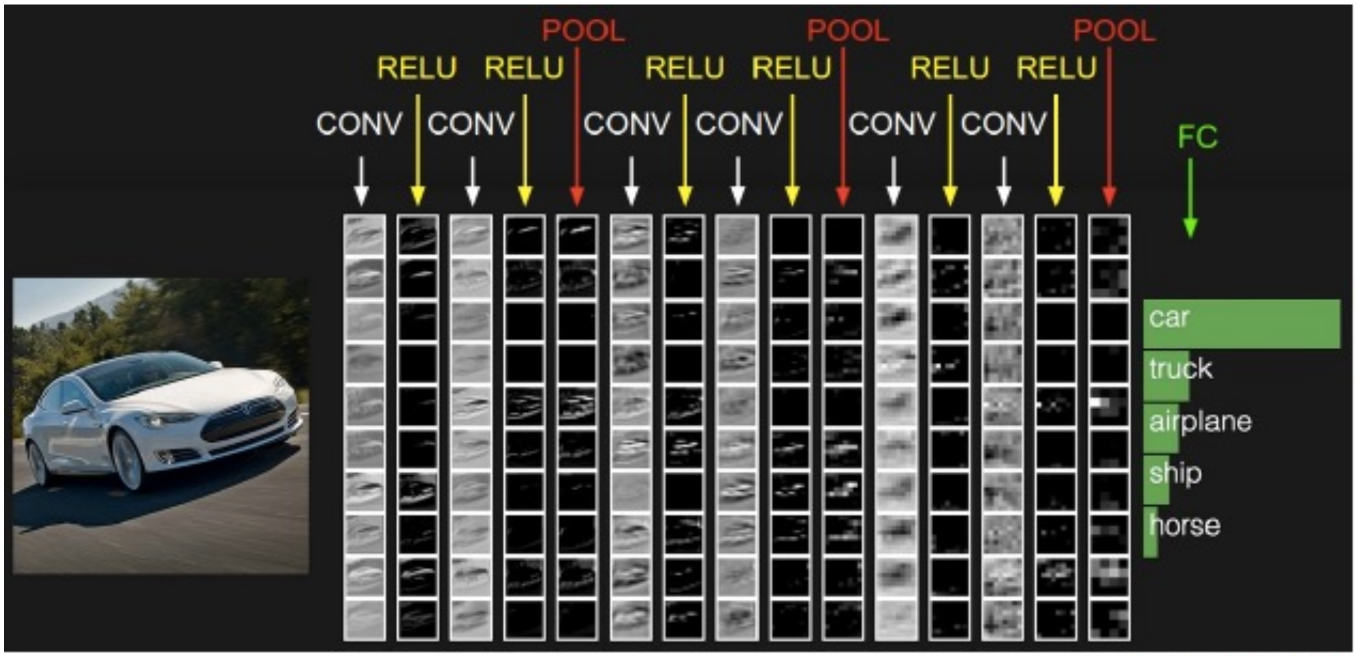


Figure source: A. Karpathy

Convolutional Neural Networks (CNN): Summary

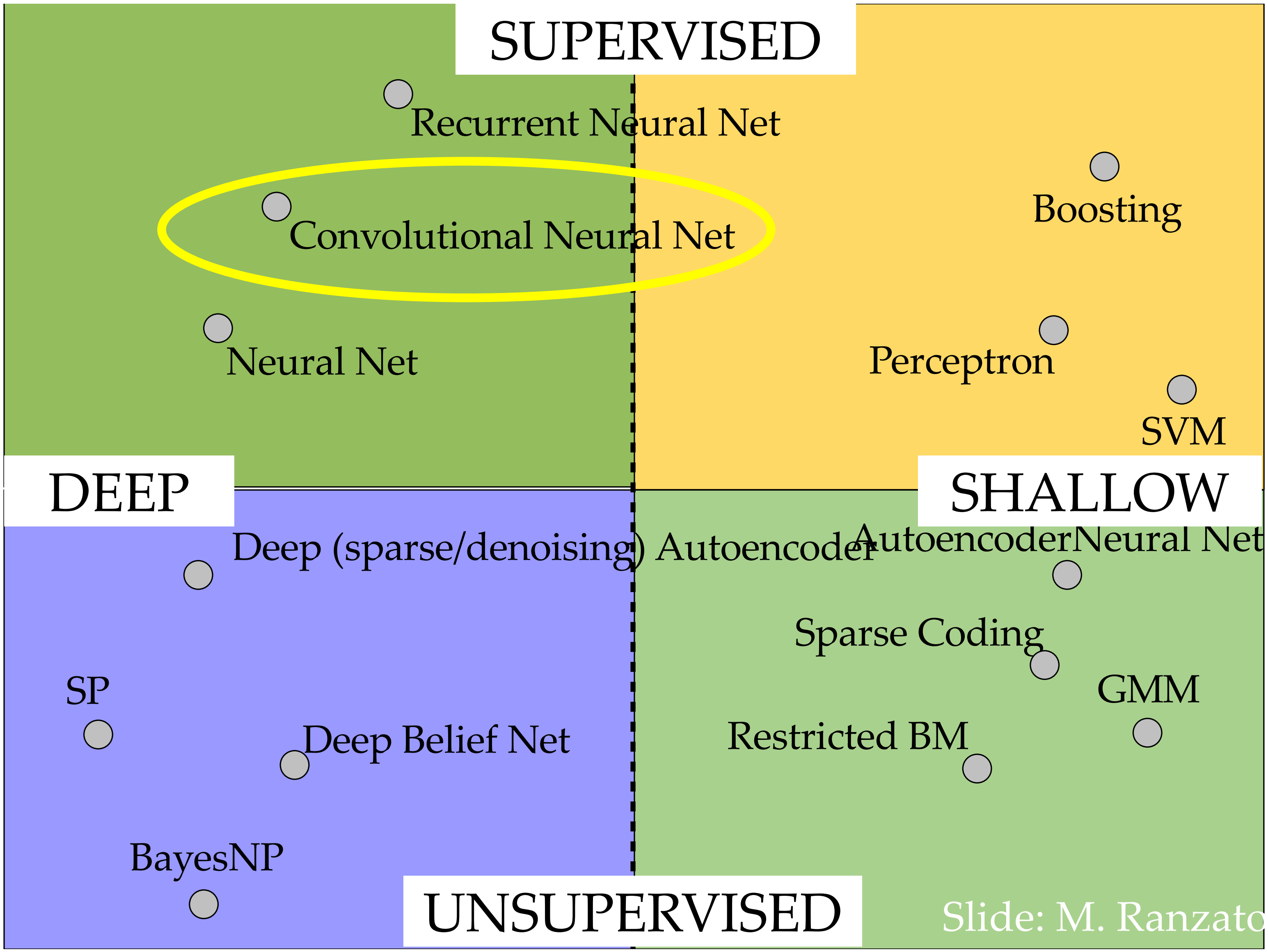
- A special kind of multi-layer neural networks.
- Implicitly extract relevant features.
- A feed-forward network that can extract topological properties from an image.
- Like almost every other neural networks CNNs are trained with a version of the back-propagation algorithm.
- ConvNets stack CONV,POOL,FC layers
- Trend towards smaller filters and deeper architectures
- Trend towards getting rid of POOL/FC layers (just CONV)

What's missing from the picture?

- Training tricks and details: initialization, regularization, normalization
- Training data augmentation
- Averaging classifier outputs over multiple crops/flips
- Ensembles of networks

What about ILSVRC 2016?

- No more ImageNet classification
- No breakthroughs comparable to ResNet



SUPERVISED
Recurrent Neural Net
Boosting
Convolutional Neural Net
Neural Net
Perceptron
SVM
DEEP
SHALLOW
Deep (sparse/denoising) Autoencoder
Autoencoder
Neural Net
Sparse Coding
SP
GMM
Deep Belief Net
Restricted BM
BayesNP
UNSUPERVISED
Slide: M. Ranzato

Convolutional Neural Networks