

Image classification and intro to neural networks

Vlad Shakhuro



Outline

- I. Image classification task and datasets
2. Linear classification and MLPs
3. Convolutional neural networks
4. Milestone: AlexNet

Binary classification

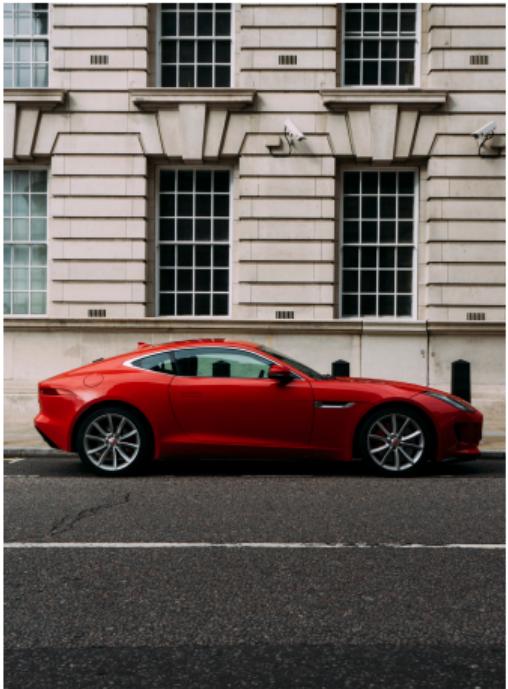


Does this image contain a pedestrian?

Binary answer $y \in \begin{cases} 0, & \text{no} \\ 1, & \text{yes} \end{cases}$

Alternatively, the estimated probability
of the positive answer $p_{\text{yes}} \in [0; 1]$

Multiclass classification



Which object is shown on this image?

The set of *allowed* object classes is determined in advance

Integer answer $y \in \left\{ \begin{matrix} 1 & , & 2 & , & \dots & , & S \\ \text{car} & , & \text{sign} & , & & & \text{bike} \end{matrix} \right\}$

Alternatively, a list of estimated probabilities:

$$p_i \in [0; 1] \quad i \in 1, \dots, S \quad \sum_{i=1}^S p_i = 1$$

Attribute recognition



Male
Asian
Bearded
Smiling

Attributes are properties or characteristics that are commonly expressed by some object

Human attributes may include race, sex, age, color of hair, current emotional state or the presence of wearable accessories such as masks, glasses and hats

Attribute recognition can often be reduced to one or more classification tasks, for example:

- *sex* → binary
- *race* → multiclass
- *age* → multiclass (over discrete age groups)

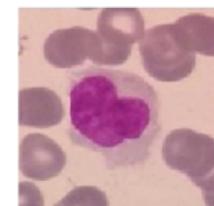
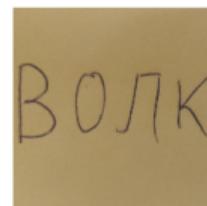
Metrics

Accuracy — percentage of correctly classified samples

Dataset	CNN	Original	BP[23]	CBP[11]	KP	Others
CUB [43]	VGG-16 [38]	73.1*	84.1	84.3	86.2	82.0 84.1
	ResNet-50 [15]	78.4	N/A	81.6	84.7	[18] [16]
Stanford Car [19]	VGG-16	79.8*	91.3	91.2	92.4	92.6 82.7
	ResNet-50	84.7	N/A	88.6	91.1	[18] [14]
Aircraft [27]	VGG-16	74.1*	84.1	84.1	86.9	80.7
	ResNet-50	79.2	N/A	81.6	85.7	[14]
Food-101 [4]	VGG-16	81.2	82.4	82.4	84.2	50.76
	ResNet-50	82.1	N/A	83.2	85.5	[4]

Top-K Accuracy (Rank K) — percentage of sample for which the correct class is within K most likely predicted classes (often K=5)

Data domains and modalities

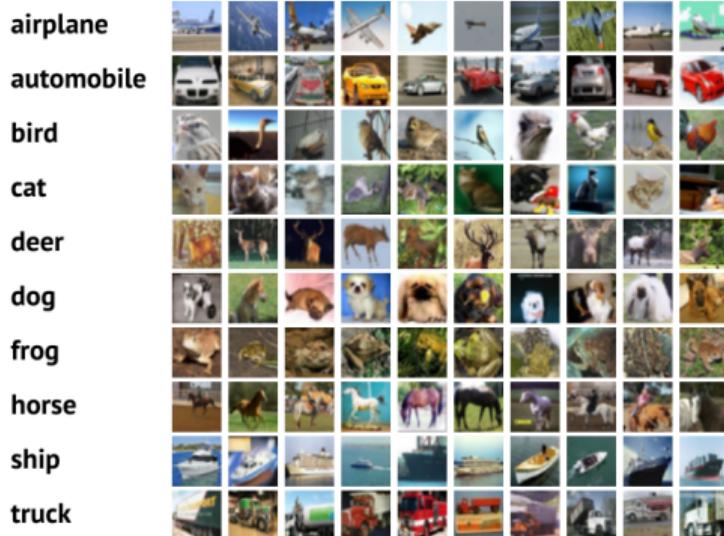


Every computer vision algorithm is designed to operate on images sampled from some *statistical population*. This population is described by an empirical distribution over the set of all “valid” (for that algorithm) images:

$$img \sim P(\mathbb{I}) \quad \mathbb{I} \subseteq \mathbb{R}^{H \times W \times C}$$

These algorithms work by exploiting the inherent properties and invariants of the *statistical population* they support

CIFAR-10 and CIFAR-100



Subset of the TinyImages collection
60000 images total

CIFAR-10: 10 classes

- 5000 training images per class
- 1000 testing images per class

CIFAR-100: 100 classes

- 500 training images per class
- 100 testing images per class

ImageNet

Goal: create a dataset with at least 1000 images for each of the original 117000 synsets/classes

~14 000 000 images

(~1 000 000 images with bounding box annotations)

~22 000 non-empty classes (~10 000 classes with at least 1000 examples)

prisoner housing animal weight
offspring teacher computer drop headquarters television
register, insurance gallery court key structure light date spread
king fireplace church press market lighter
hotel road Paper cup concert pack
sport screen tree file tower camp fish, salmon
sky plant wall means fan hill can railcar
bread table top man car study stock film
cloud cover range leafy net menu ball button
spring range leafy net menu ball button
bed shop top man car study stock film
kitchen train camera box memory sieve cell kid bar watch
engine box center step goal
chain dinner stone child case student stand
apple girl hat home room office club
flag bank cross chair minicard rule hall
radio support level line street golf
beach library stage video food building
tool material player machine security call clock
football hospital match equipment cell phone mountain telephone
short circuit bridge scale gas point microphone recording crowd



ImageNet: annotation problems



mite

container ship

motor scooter

leopard

mite	container ship	motor scooter	leopard
black widow	lifeboat	go-kart	jaguar
cockroach	amphibian	moped	cheetah
tick	fireboat	bumper car	snow leopard
starfish	drilling platform	golfcart	Egyptian cat



grille

mushroom

cherry

Madagascar cat

convertible	agaric	dalmatian	squirrel monkey
grille	mushroom	grape	spider monkey
pickup	jelly fungus	elderberry	titi
beach wagon	gill fungus	ffordshire bulterrier	indri
fire engine	dead-man's-fingers	currant	howler monkey

OpenImages



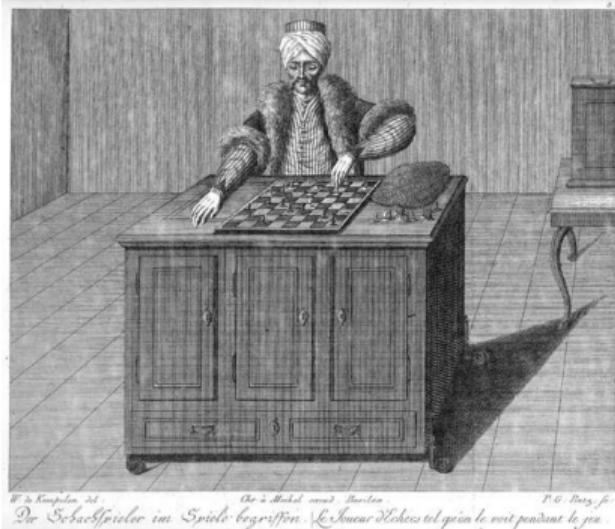
Goal: create the largest **open** dataset of real-life photographs with diverse annotations

- ~9 000 000 images
licensed under CC BY 2.0
- ~60 000 000 annotations for
~20 000 categories
- Various supplementary
annotations are also available
(for example, localized text descriptions)

Fine-grained classification



Mechanical Turk



"Mechanical Turk, Automaton Chess Player" was a robot created **in 1770** that could play chess (and even beat competent players). In 1820 it was revealed that the robot couldn't actually play chess by itself and that it was instead **controlled by a human sitting in a hidden compartment**

Galaxy Zoo



GALAXY ZOO galaxyzoo.org

- Classification of galaxy images
- The first large scale project of this kind
- More than 150 000 volunteers created over 60 000 000 annotations in a single year **for free**

Annotation as a service

amazon mechanical turk
Artificial Artificial Intelligence

Your Account HITs Qualifications

Xiaodan Zhou | Account Settings | Sign Out | Help

Introduction | Dashboard | Status | Account Settings

Mechanical Turk is a marketplace for work.

We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it's convenient.

264,053 HITs available. [View them now.](#)

Make Money
by working on HITs

HITs - Human Intelligence Tasks - are individual tasks that you work on. [Find HITs now.](#)

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work

Find an interesting task → Work → Earn money

[Find HITs Now](#)

or [learn more about being a Worker](#)

Get Results
from Mechanical Turk Workers

Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. [Register Now](#)

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

Find your account → Load your tasks → Get results

[Get Started](#)

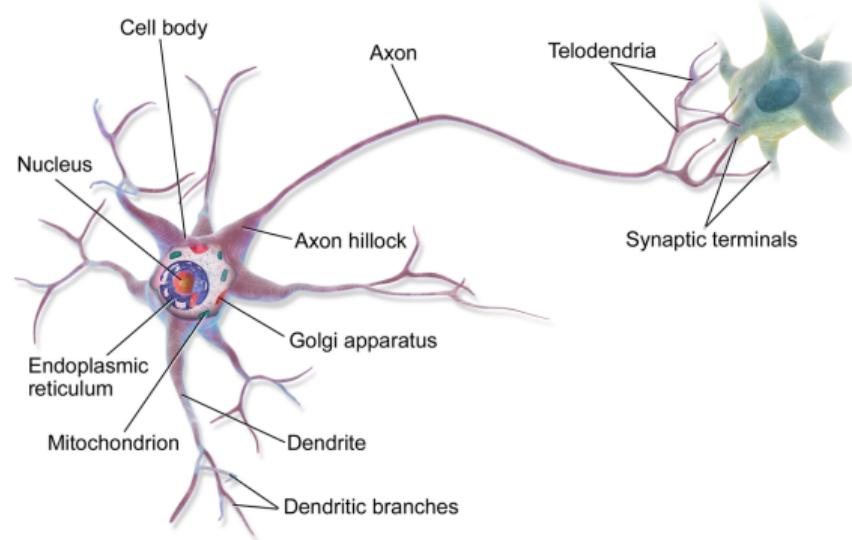
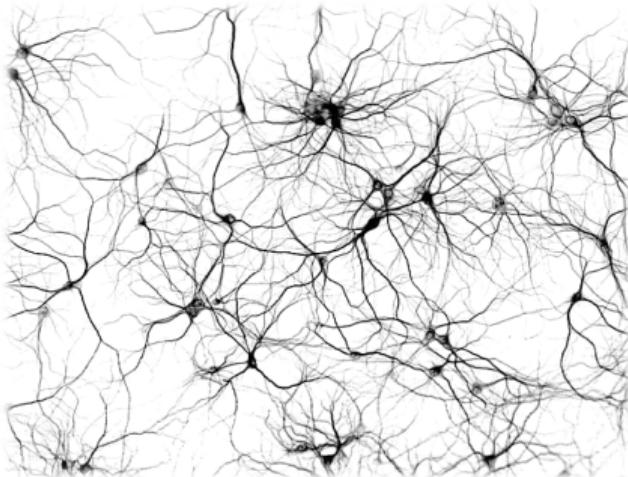
FAQ | Contact Us | Careers at Amazon | Developers | Press | Policies | Blog
©2005-2012 Amazon.com, Inc. or its Affiliates

An amazon.com company

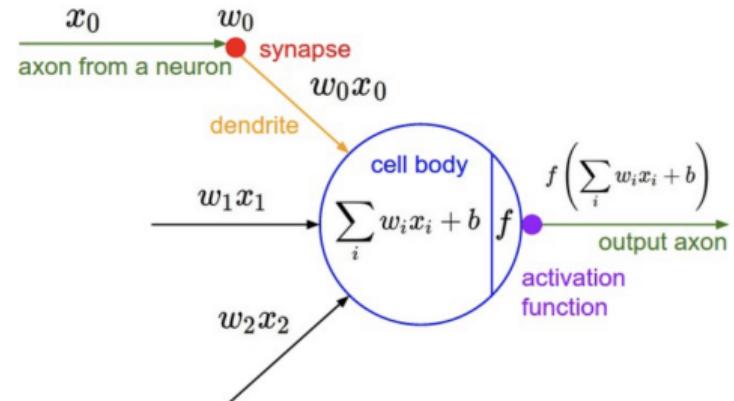
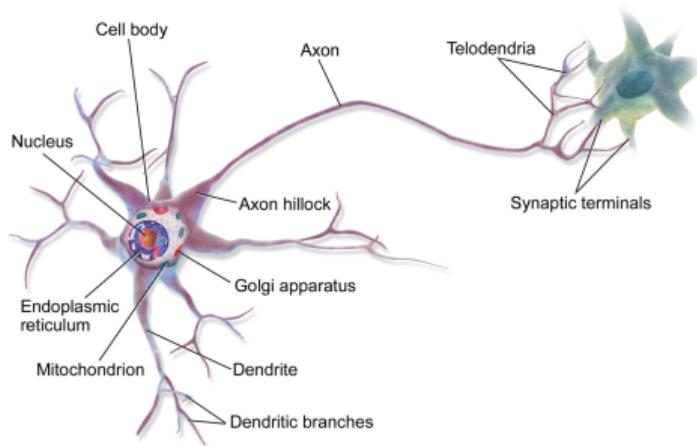
Outline

- I. Image classification task and datasets
2. Linear classification and MLPs
3. Convolutional neural networks
4. Milestone: AlexNet

Biological neurons

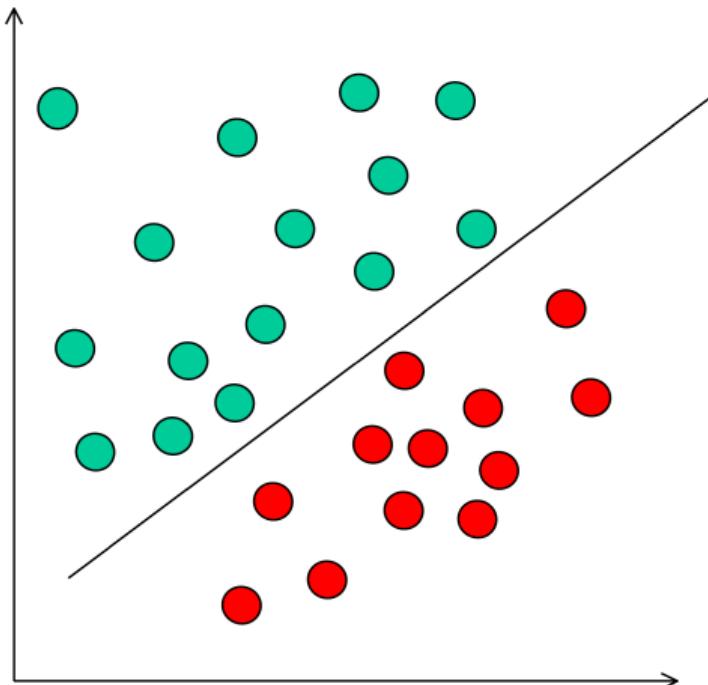


McCulloch-Pitts neuron model



$$a(x, w) = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

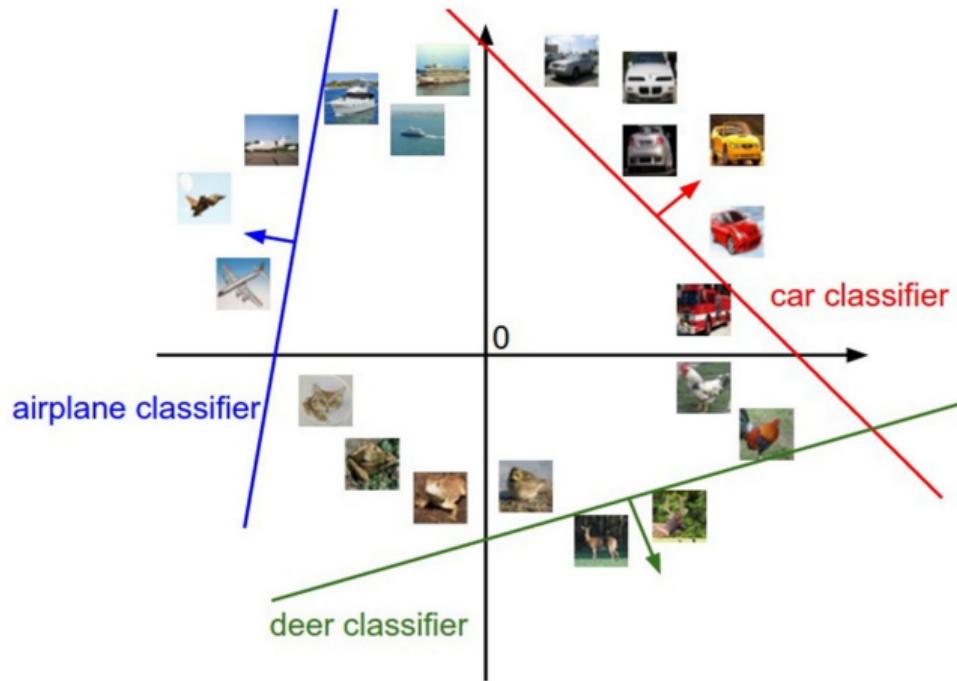
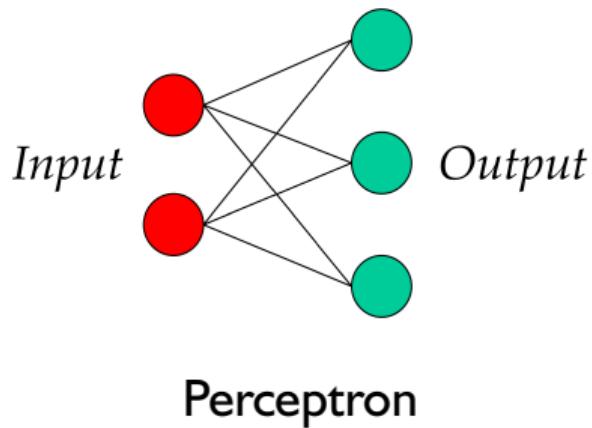
Neuron as linear classifier



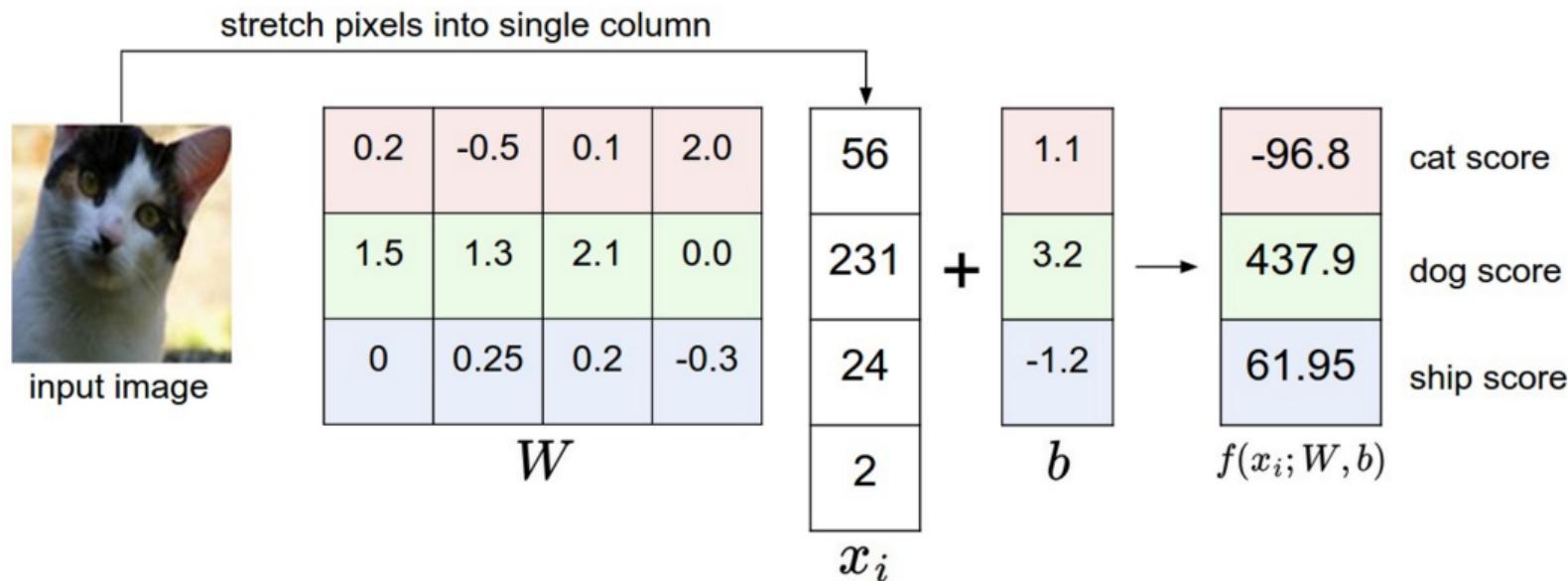
$$a(x, w) = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

Optimal parameters w_i can be found using classical iterative methods

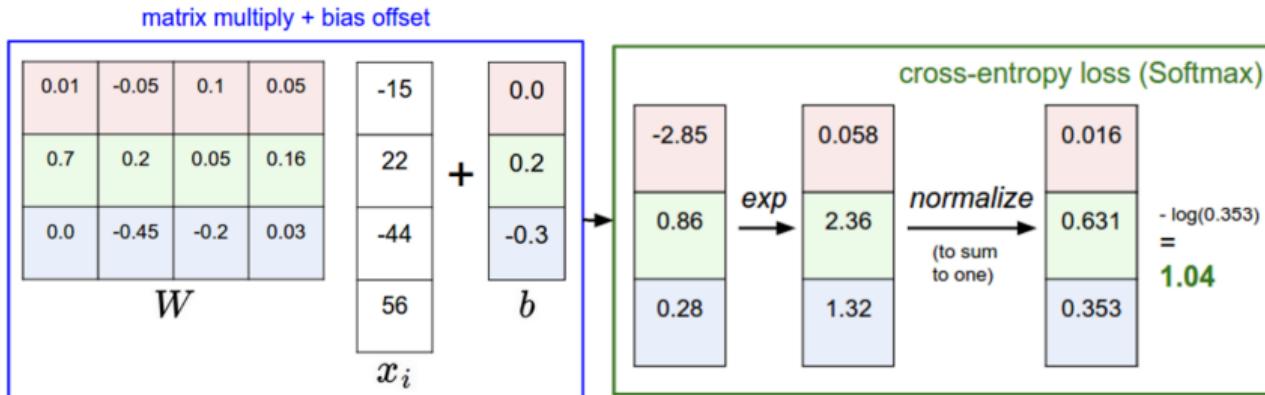
Multiclass classification



Multiclass classification for images



Loss function



Normalize scores with softmax activation:

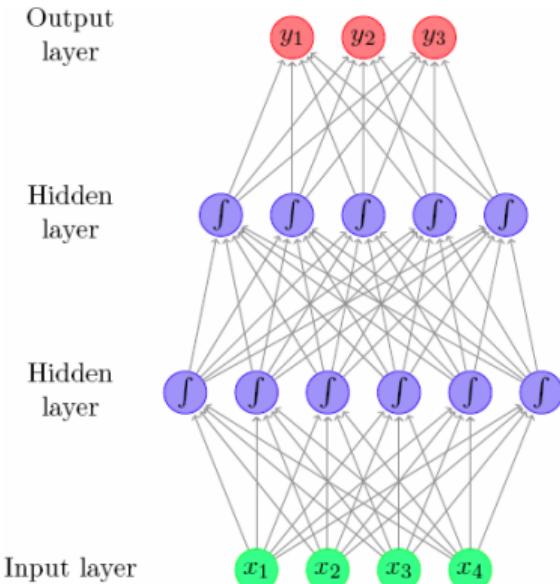
$$p_i^{\text{pr}} = \frac{e^{y_i}}{\sum_{j=1}^N e^{y_j}}$$

and compute categorical cross-entropy:

$$L(p^{\text{pr}}, p^{\text{gt}}) = - \sum_{i=1}^N p_i^{\text{gt}} \cdot \log(p_i^{\text{pr}})$$

Then we can train neuron using SGD with minibatches

Multilayer perceptron (MLP)



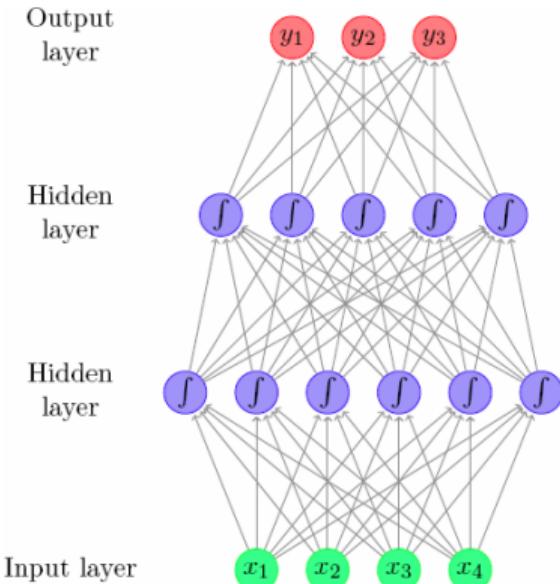
Chained perceptrons may be called **deep neural networks**

Layers in NN may have two meanings: a set of neuron activations (also called representations) and a set of connections with weights

Hidden layer neurons usually have nonlinear activation function (sigmoid, ReLU)

Number of outputs depends on task

Multilayer perceptron (MLP)



Chained perceptrons may be called **deep neural networks**

Layers in NN may have two meanings: a set of neuron activations (also called representations) and a set of connections with weights

Hidden layer neurons usually have nonlinear activation function (sigmoid, ReLU)

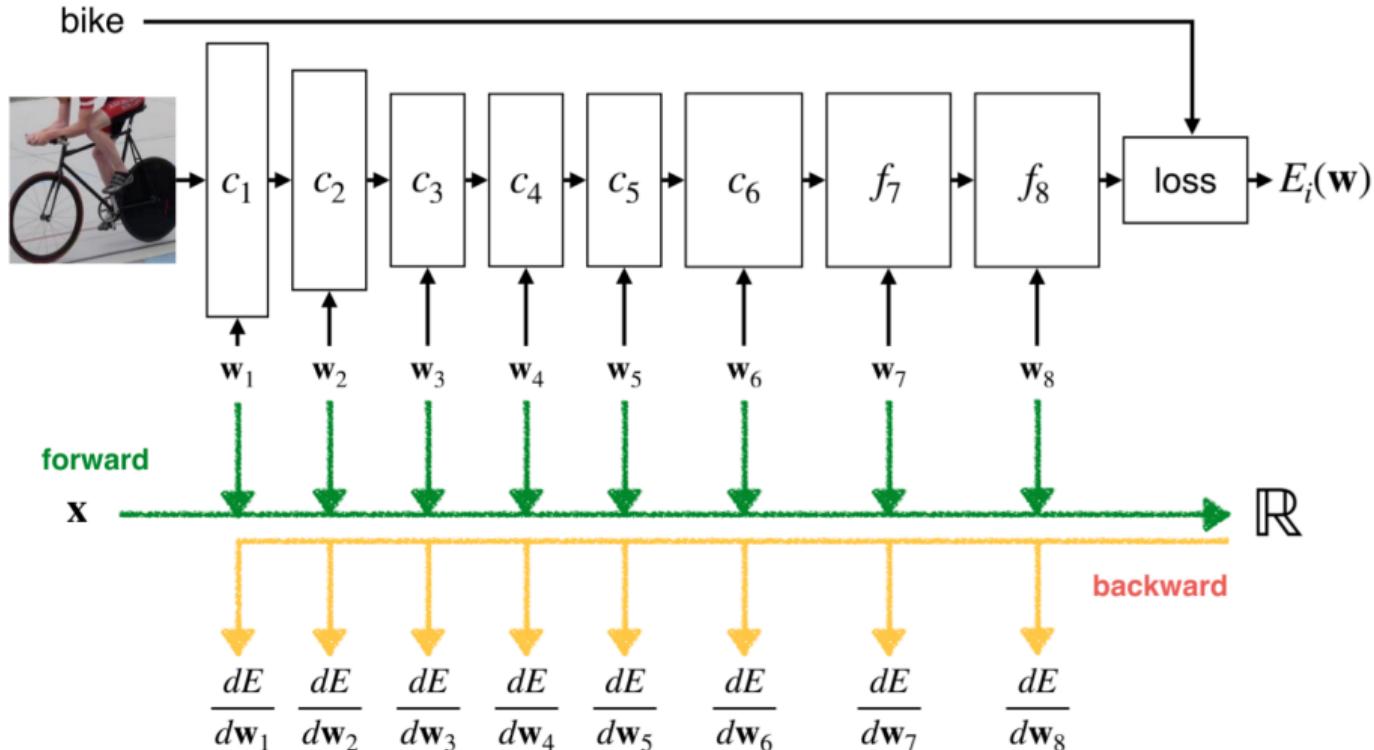
Number of outputs depends on task

How can we define architecture?

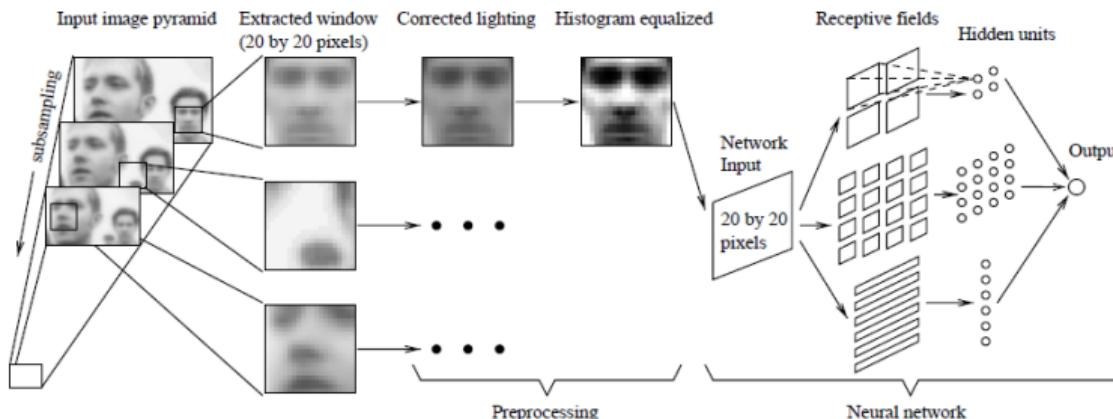
Backpropagation

class c_i

image \mathbf{x}_i



Rowley face detector

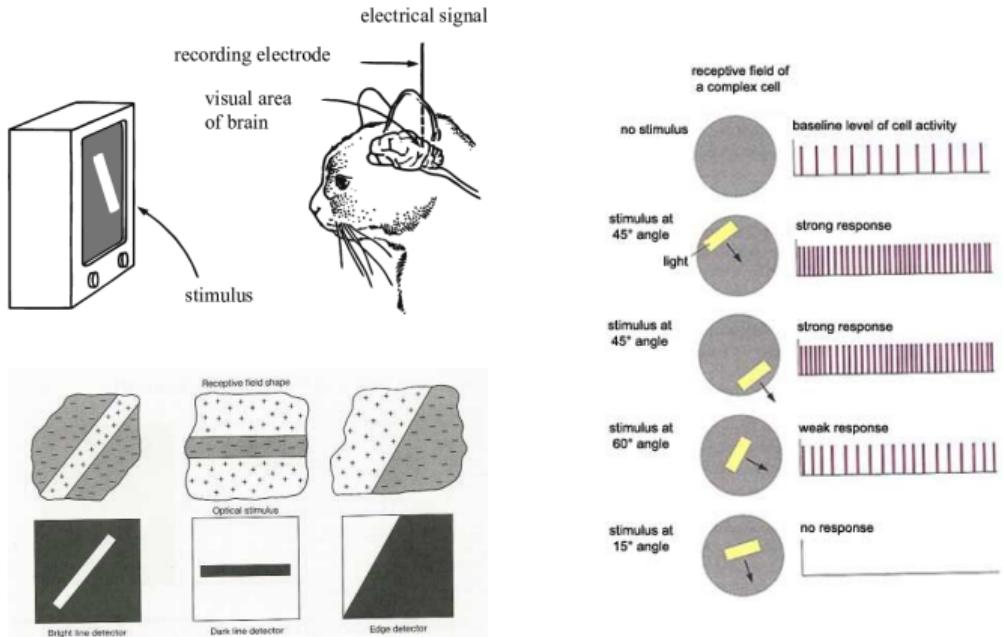
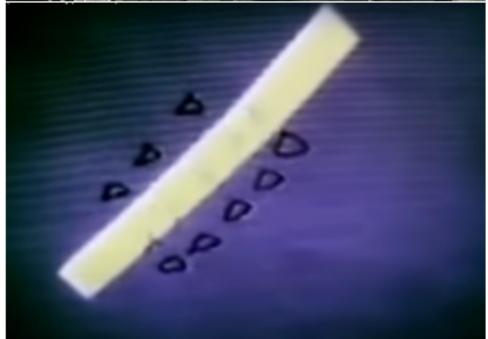


Rowley, Kanade. Neural Network-Based Face Detection. PAMI 1998

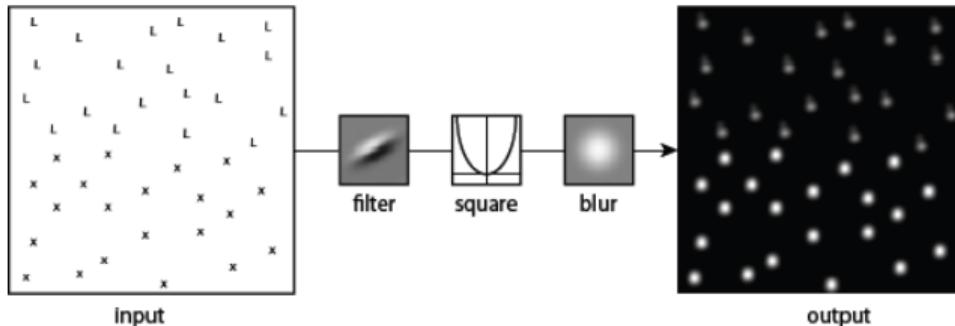
Outline

- I. Image classification task and datasets
2. Linear classification and MLPs
3. Convolutional neural networks
4. Milestone: AlexNet

Hubel and Wiesel visual cortex experiments



Modelling texture



Texture may be described using a bank of filters. Every pixel convolved with filters will give vector of features

Gabor filter as a model for simple cells

Bank of filters may be obtained using gabor filters for different orientations:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right)$$

$$x' = x \cos \theta + y \sin \theta$$

$$y' = -x \sin \theta + y \cos \theta$$

Parameters:

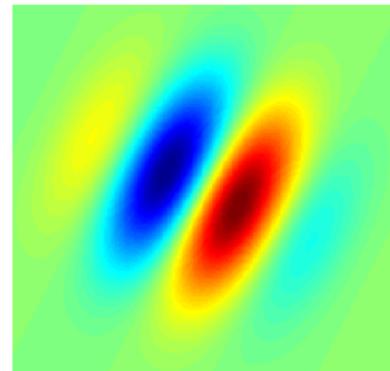
σ — gaussian stdev

γ — aspect ratio

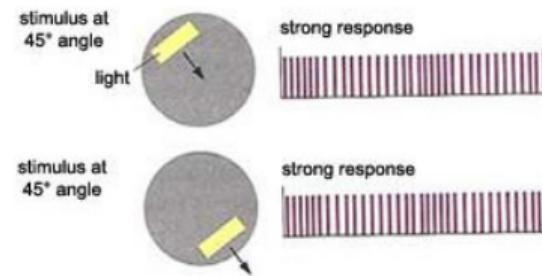
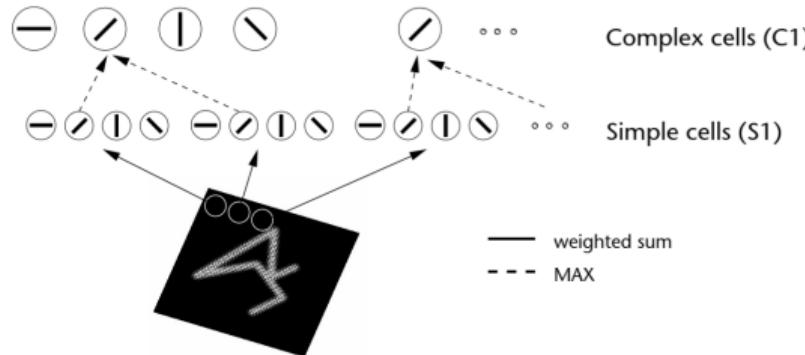
θ — orientation

λ — wave length

ψ — phase shift

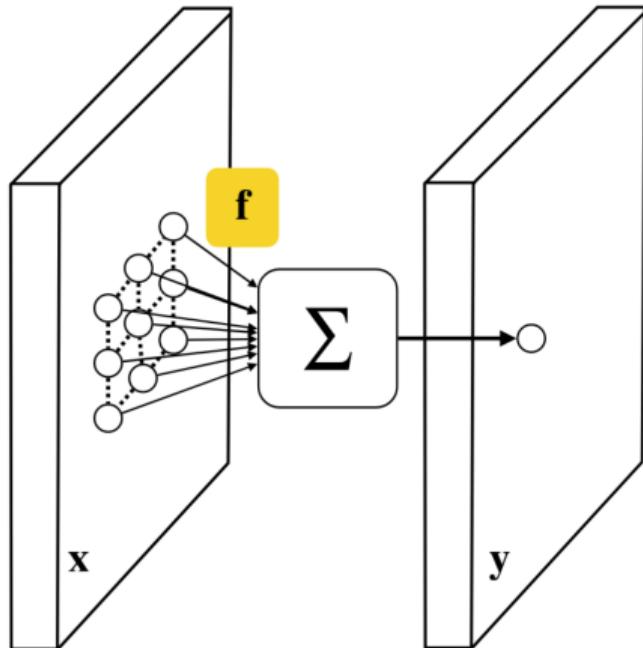


Max operation as a model for complex cells



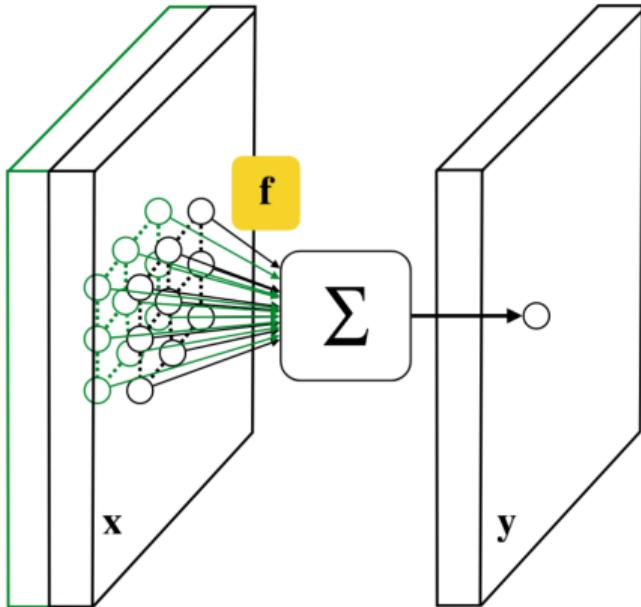
Position invariance (complex cells) may be obtained using MAX operation on top of simple convolutional cells

Convolutional layer



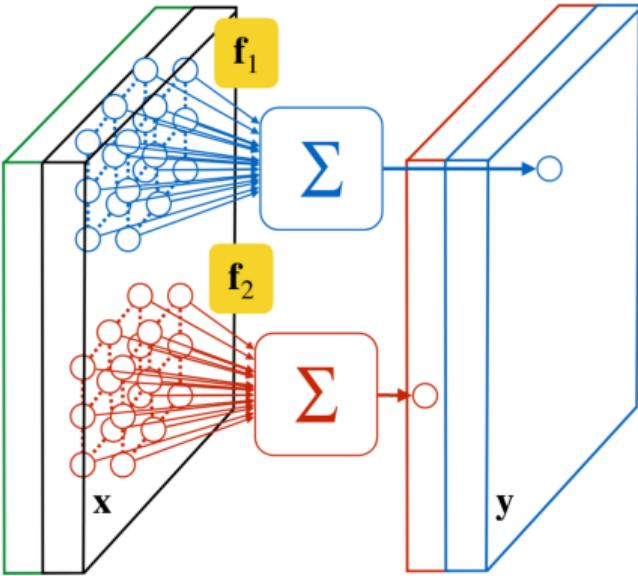
Convolution (linear filtering) for whole image may be modelled using a layer of neurons with shared weights.

Convolutional layer



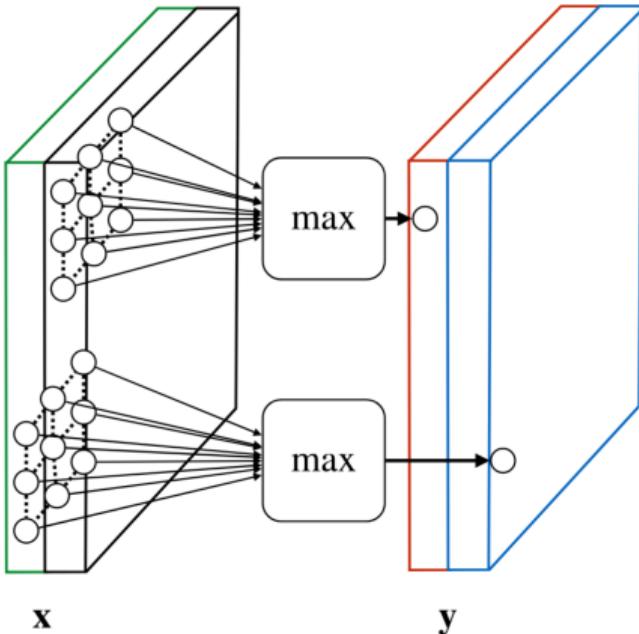
Convolution (linear filtering) for whole image may be modelled using a layer of neurons with shared weights.

Convolutional layer

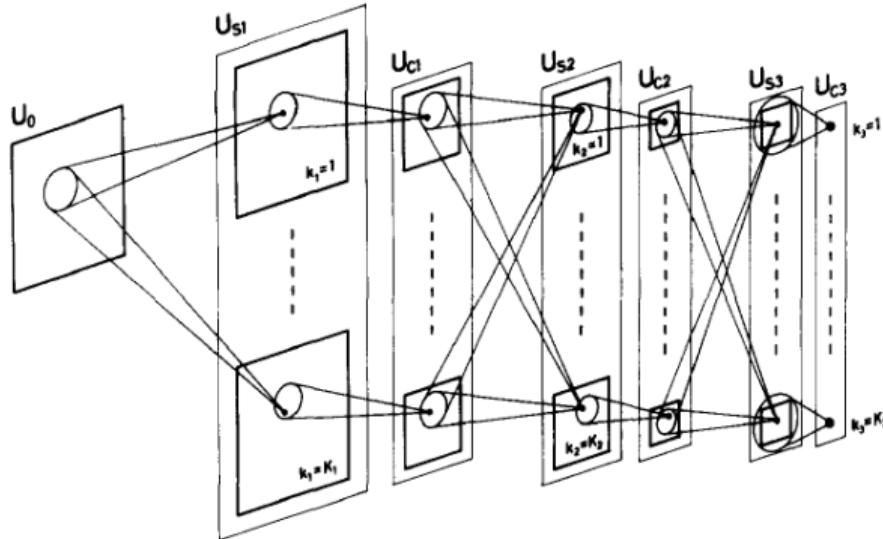


Convolution (linear filtering) for whole image may be modelled using a layer of neurons with shared weights. Convolutional layer is a set of convolutions over the same input

Max pooling layer



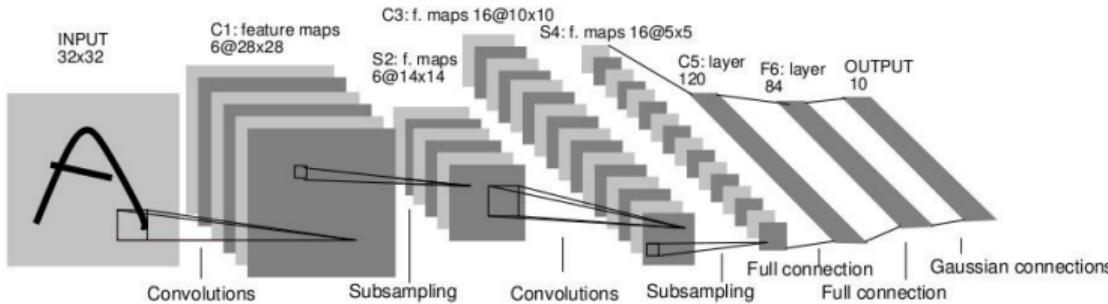
Neocognitron



Multilayer network with interleaved S and C layers.
Last layer neurons are invariant to shifts in image

Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 1980

LeNet

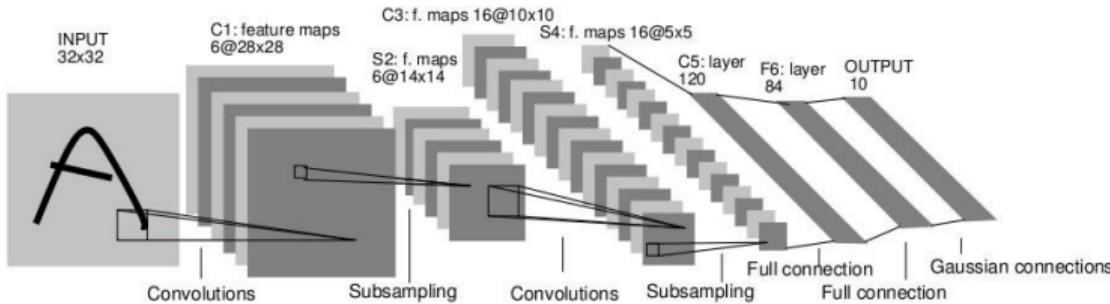


Neocognitron idea + error backpropagation method
→ Convolutional Neural Network (CNN)

Since convolutional neurons share parameters and look at a small neighbourhood, convolutional networks are very effective

LeCun et al. Gradient-based learning applied to document recognition. 1998

LeNet



Neocognitron idea + error backpropagation method
→ Convolutional Neural Network (CNN)

Since convolutional neurons share parameters and look at a small neighbourhood, convolutional networks are very effective

How many trained weights are there in different layers? (C1, S2, ..., F6, Output)?

LeCun et al. Gradient-based learning applied to document recognition. 1998

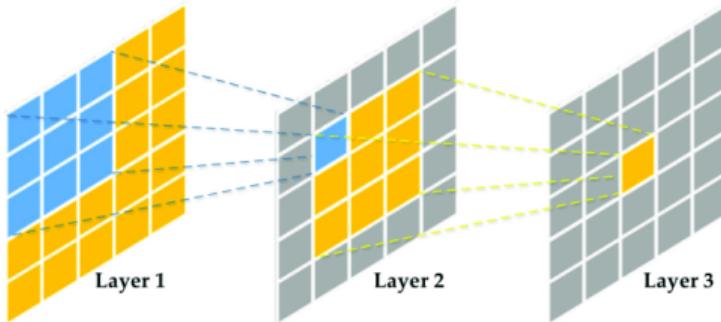
Convolutional filters for RGB images



Neural networks trained on RGB image classification task have first layers very similar to Gabor filter

Some layers may duplicate each other

Receptive field



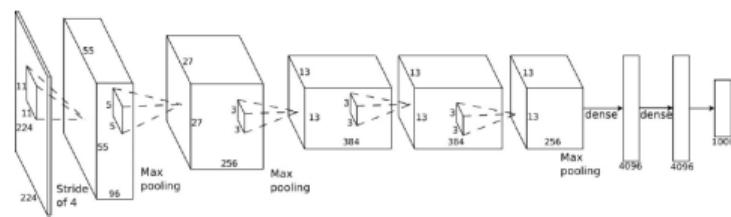
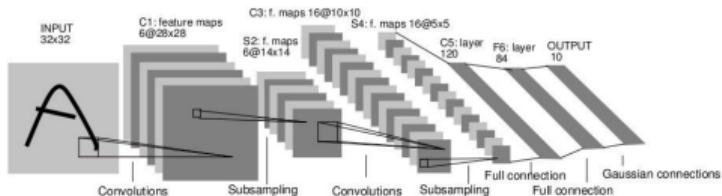
Receptive field is an area of image that *may* influence neuron output.
Depends on network architecture

Effective receptive field is an area that depends on trained weights

Outline

- I. Image classification task and datasets
2. Linear classification and MLPs
3. Convolutional neural networks
4. Milestone: AlexNet

LeNet and AlexNet comparison



1998:

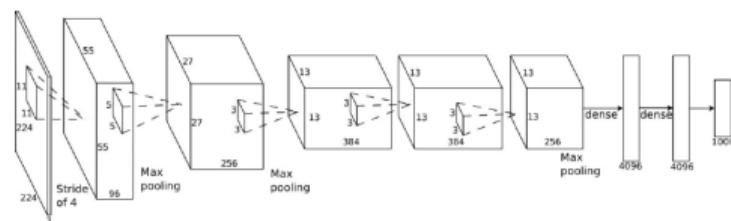
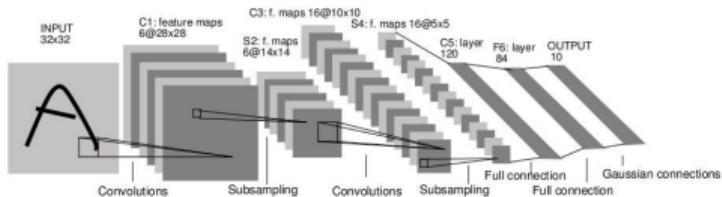
- 2 conv layers (6, 16 filters)
- 2 fully connected layers (120, 84 neurons)

2012:

- 5 conv layers (96, 256, 384, 384, 256 filters)
- 2 fully connected layers (4096, 4096 neurons)

Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks. NIPS 2012

LeNet and AlexNet comparison



1998:

- 2 conv layers (6, 16 filters)
- 2 fully connected layers (120, 84 neurons)

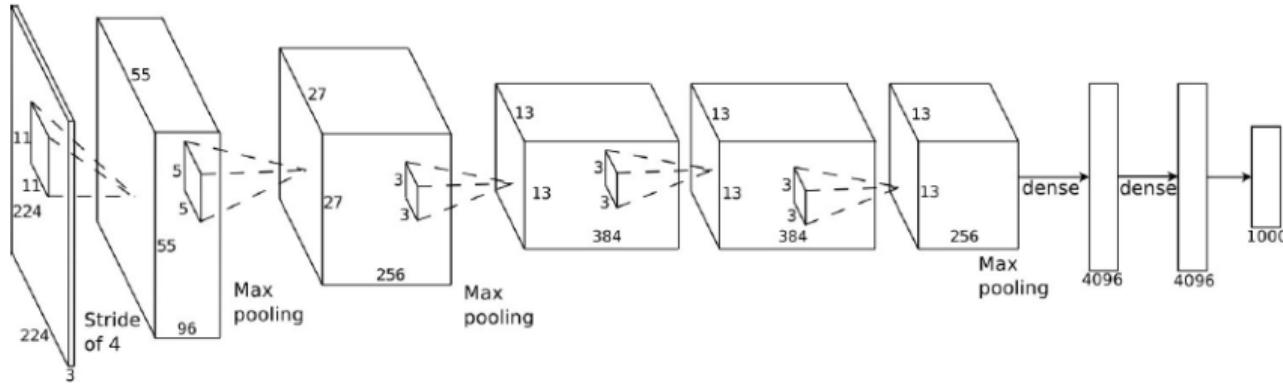
2012:

- 5 conv layers (96, 256, 384, 384, 256 filters)
- 2 fully connected layers (4096, 4096 neurons)

What else has changed?

Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks. NIPS 2012

AlexNet

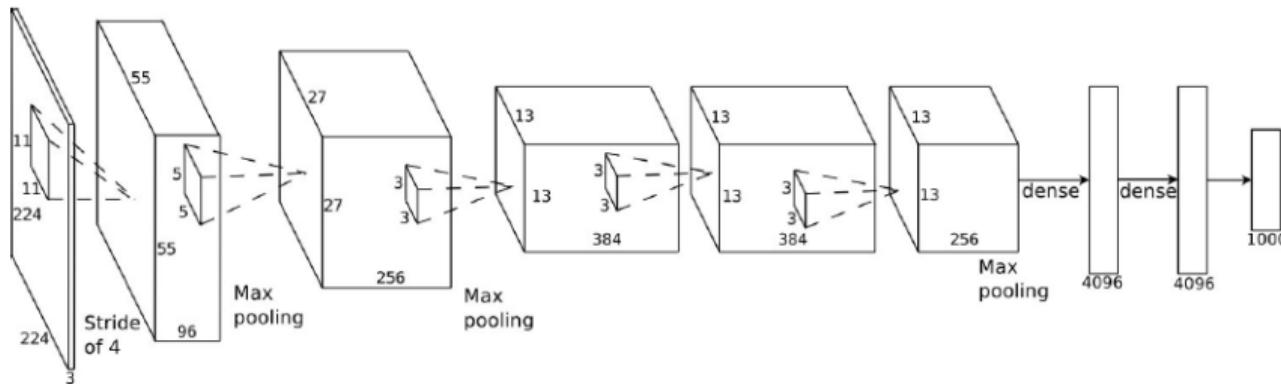


- 60M parameters
- 2GPU × 3GB, 5GB RAM, 27GB HDD
- 1 week to train

Key ideas:

- ReLU activation
- image augmentations
- dropout

AlexNet



- 60M parameters
- 2GPU \times 3GB, 5GB RAM, 27GB HDD
- 1 week to train

Key ideas:

- ReLU activation
- image augmentations
- dropout

HW: compute *manually* number of parameters for AlexNet

Conclusion

We reviewed following topics:

- image classification tasks
- how to obtain and label data
- classification with single neuron and MLP
- main biological principles behind convolutional neural networks