CAP5415 Computer Vision

Yogesh S Rawat

yogesh@ucf.edu

HEC-241

Happy Halloween!



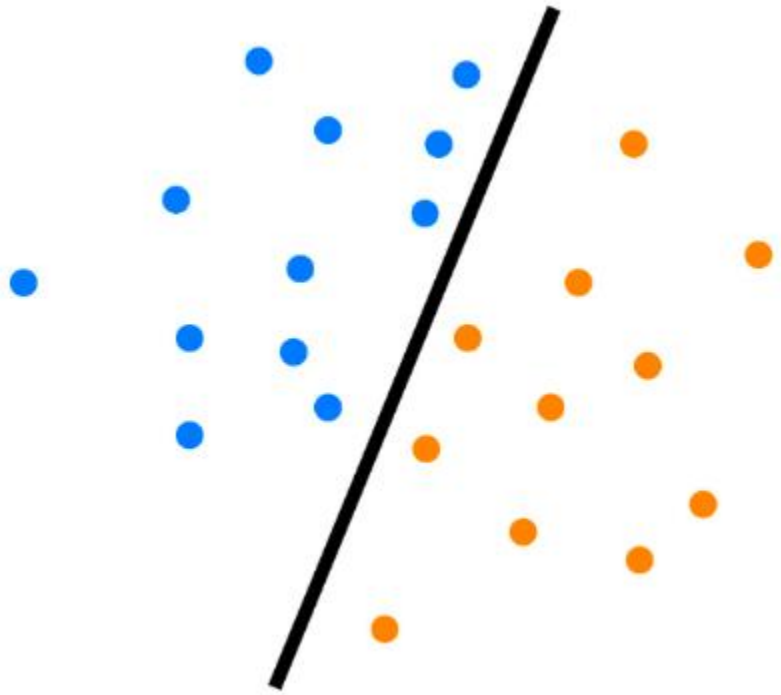
https://huggingface.co/spaces/stabilityai/stable-diffusion

Classification - II

Lecture 12

Classification

- Categorizing given set of data into classes



The machine learning framework

$$\mathbf{f(x) = y}$$

Prediction function or *classifier*

Image feature

Output (label)

Training: Given a *training set* of labeled examples:

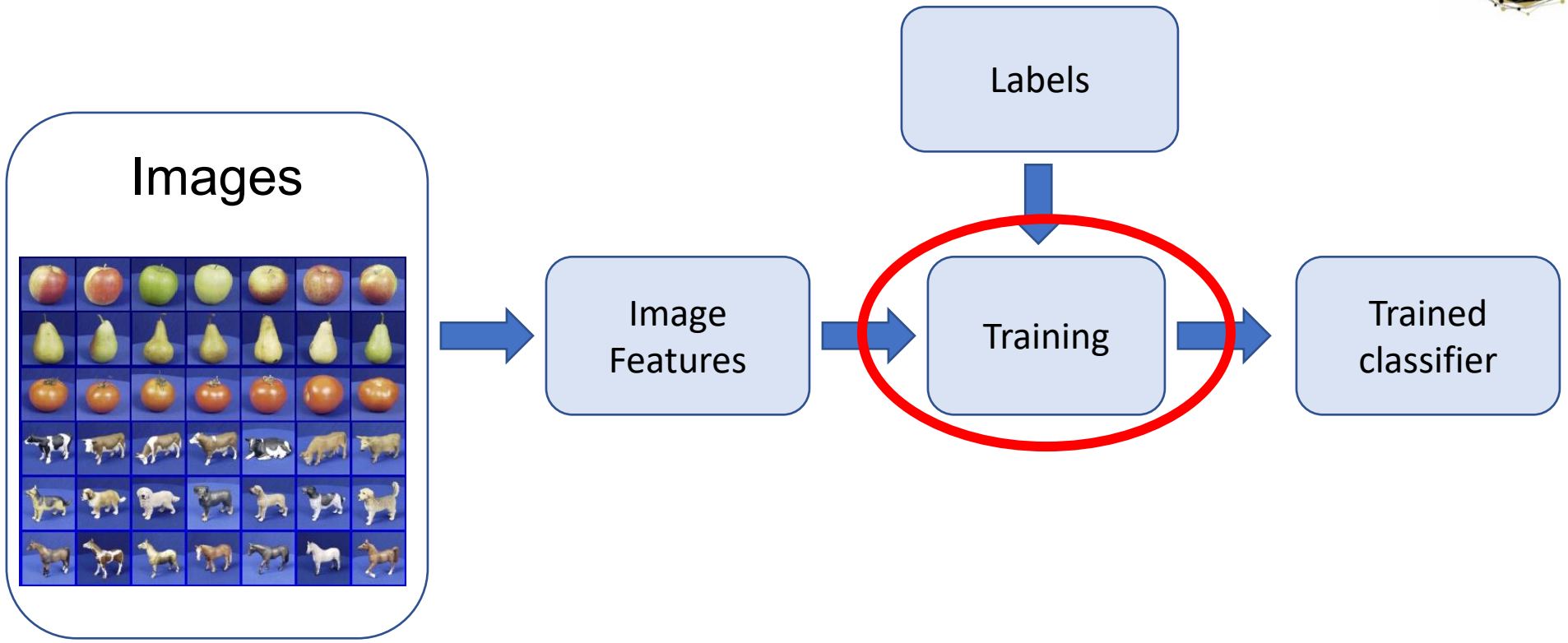
$$\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$$

Estimate the prediction function f by minimizing the prediction error on the training set.

Testing: Apply f to an unseen *test example* $\mathbf{x}_u$ and output the predicted value $y_u = f(\mathbf{x}_u)$ to *classify* $\mathbf{x}_u$.

Slide credit: L. Lazebnik

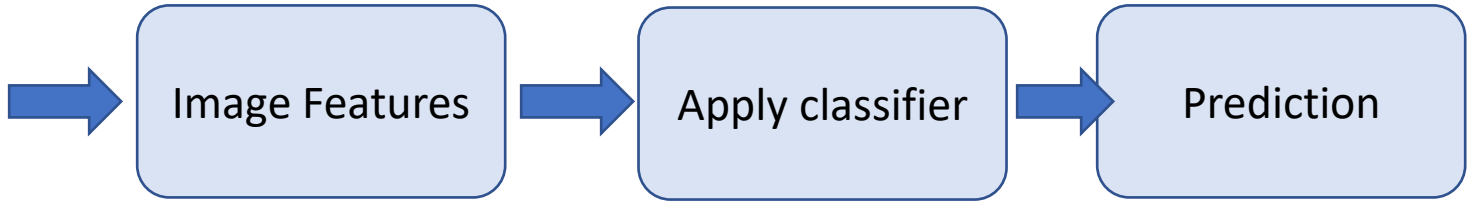
Training



Testing

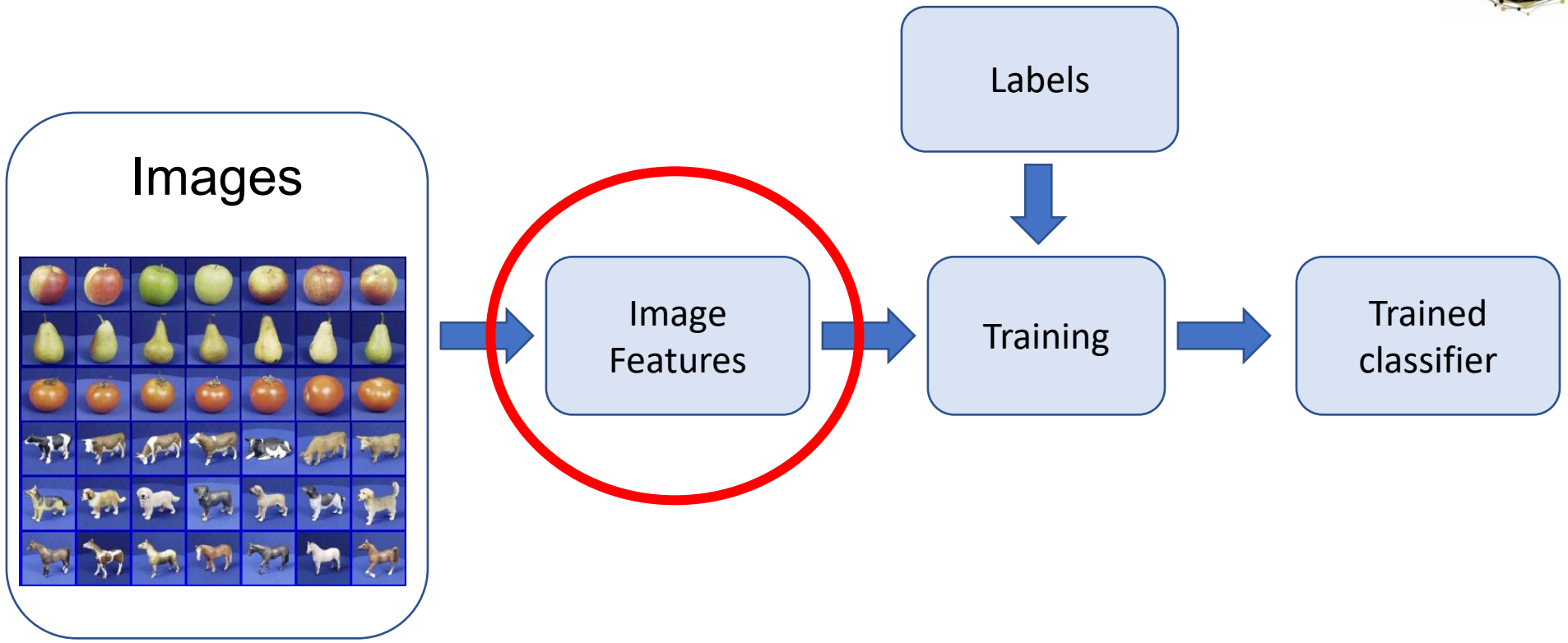
Image *not in training set*



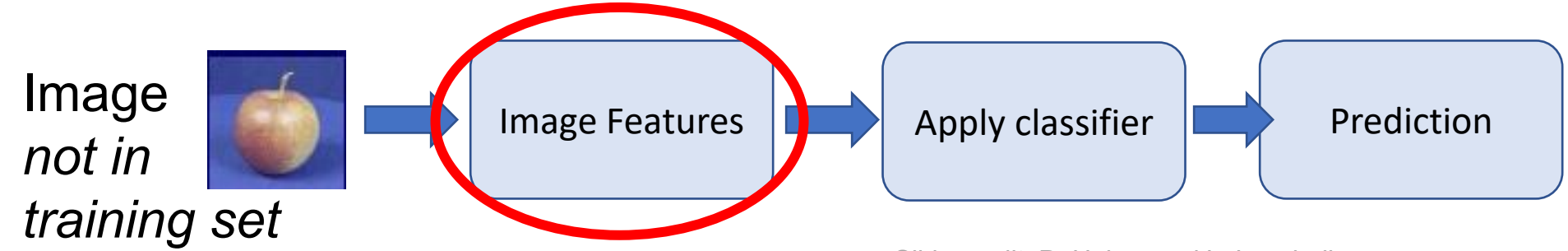


Slide credit: D. Hoiem and L. Lazebnik

Training



Testing



Slide credit: D. Hoiem and L. Lazebnik

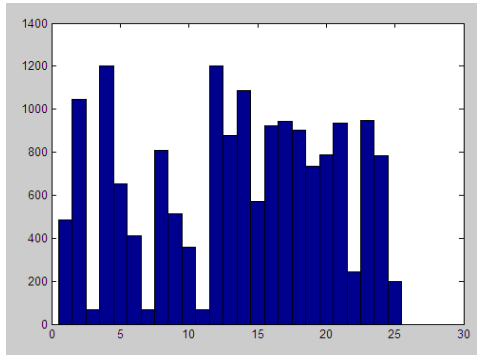
Features

- Raw pixels
- Histograms
- Templates
- SIFT
- HOG
- SURF

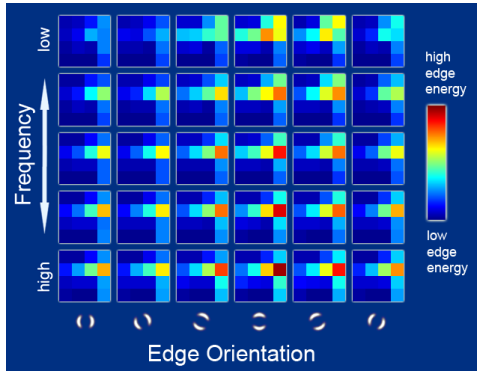
OR

- Learn features



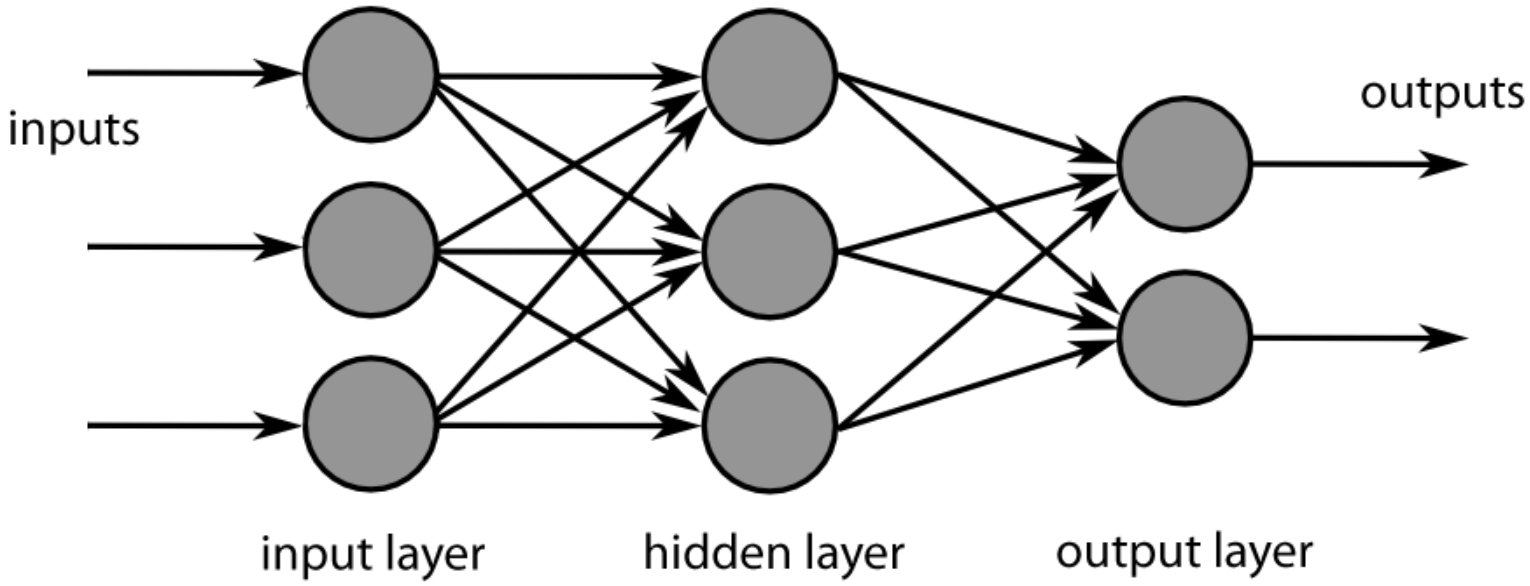




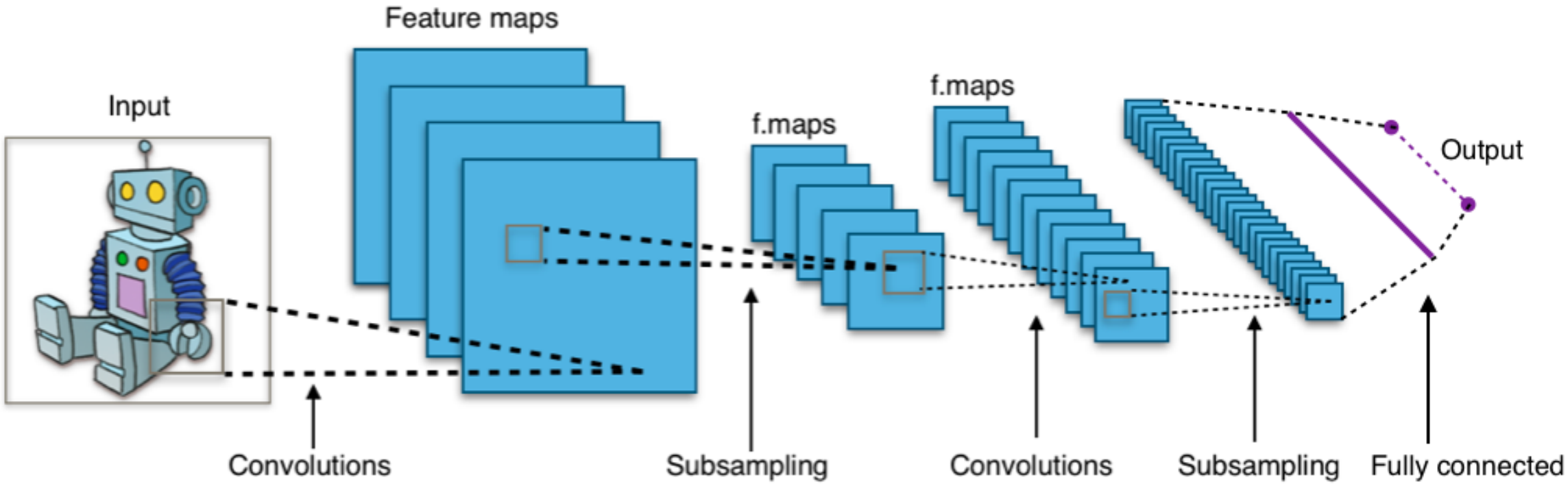


L. Lazebnik

Neural Network



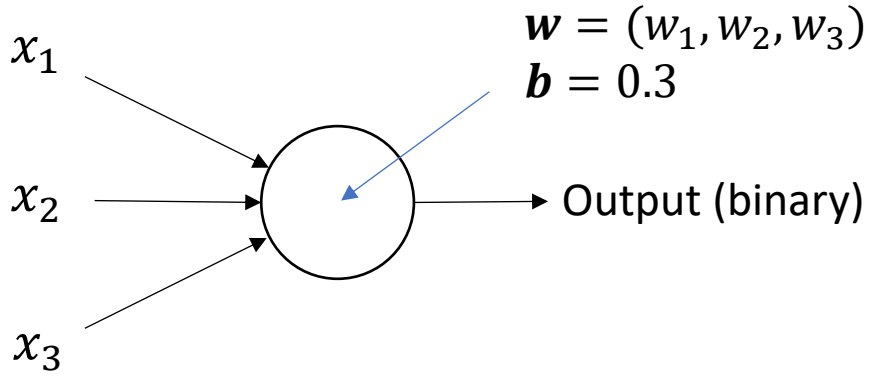
CNN architecture



Neural Networks

Basic building block for composition is a *perceptron* (Rosenblatt c.1960)

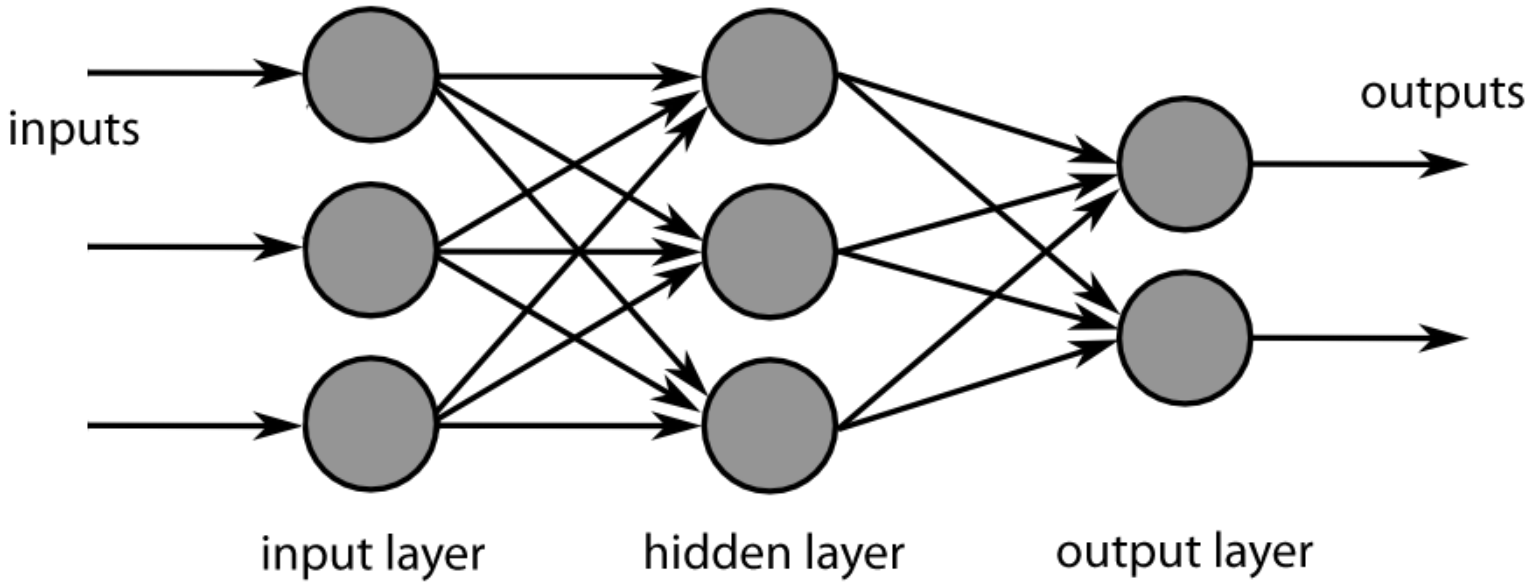
Linear classifier – vector of weights *w* and a 'bias' *b*



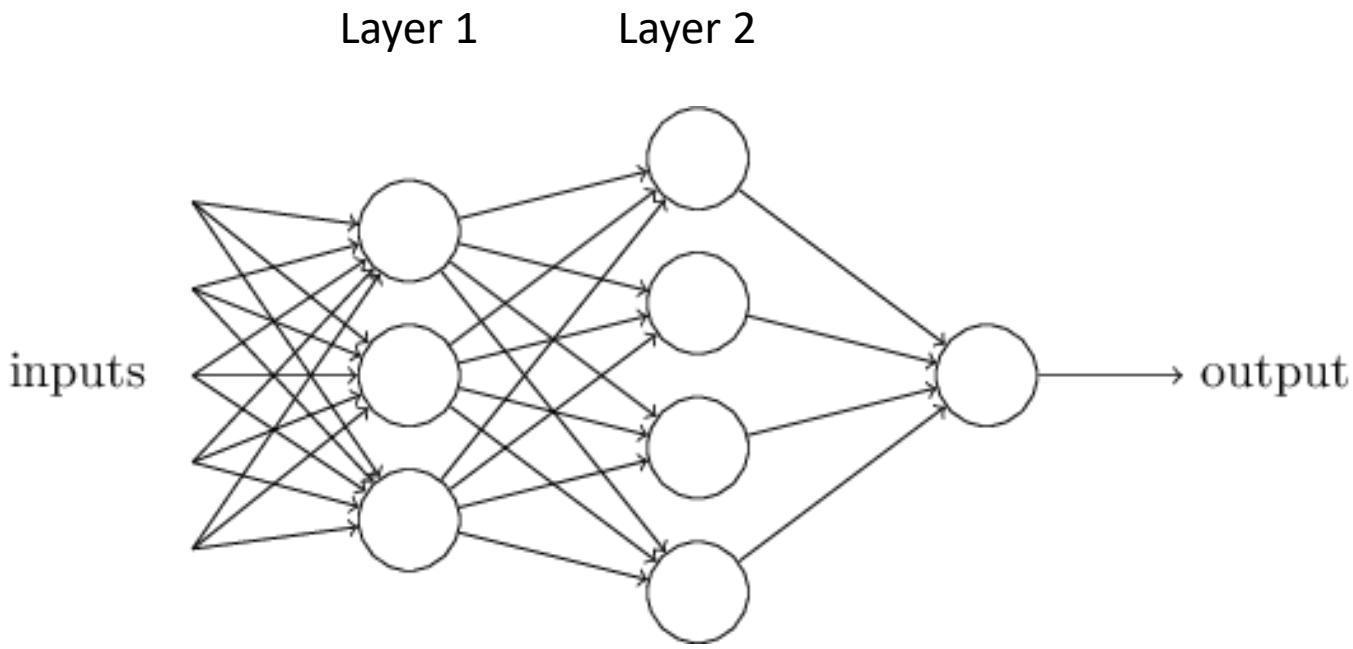
$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases} \qquad w \cdot x \equiv \sum_j w_j x_j$$

Neural Network

- Multiple classes

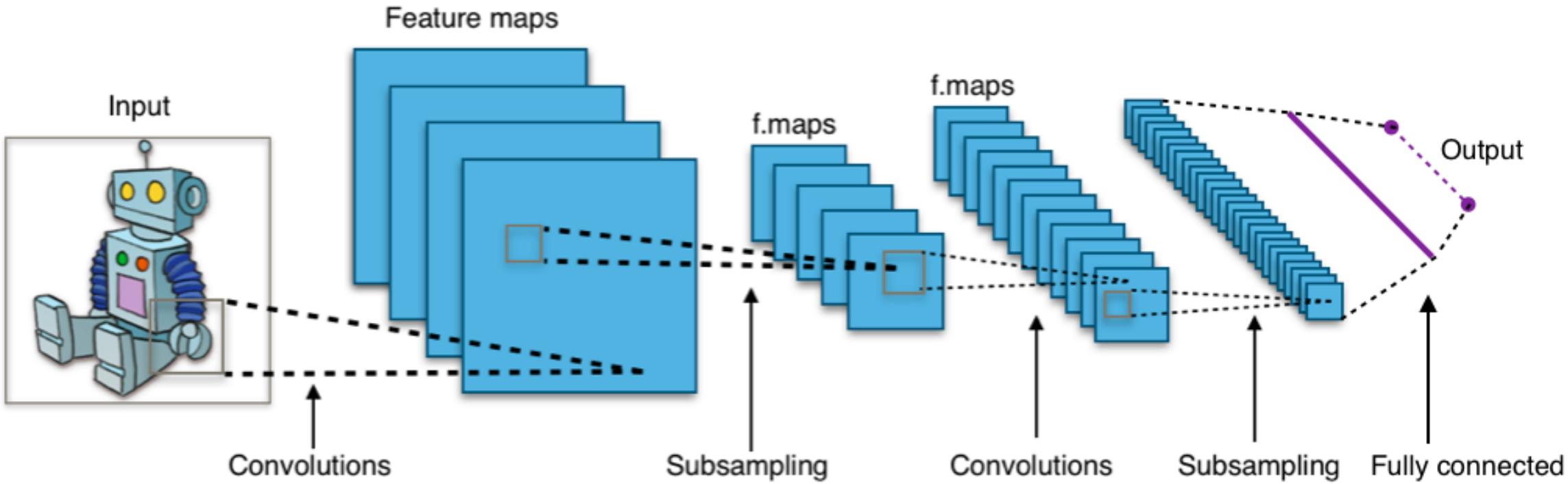


Composition

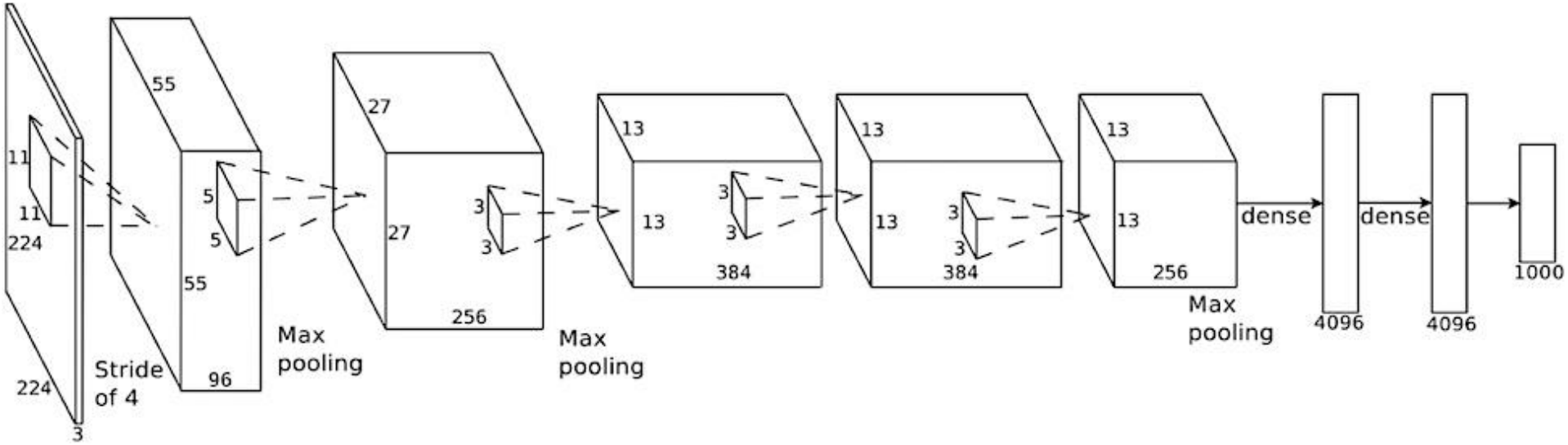


Sets of layers and the connections (weights) between them define the *network architecture.*

CNN architecture



AlexNet : Network Size



- Input 227x227x3
- 5 convolution layers
- 3 dense layers
- Output 1000-D vector

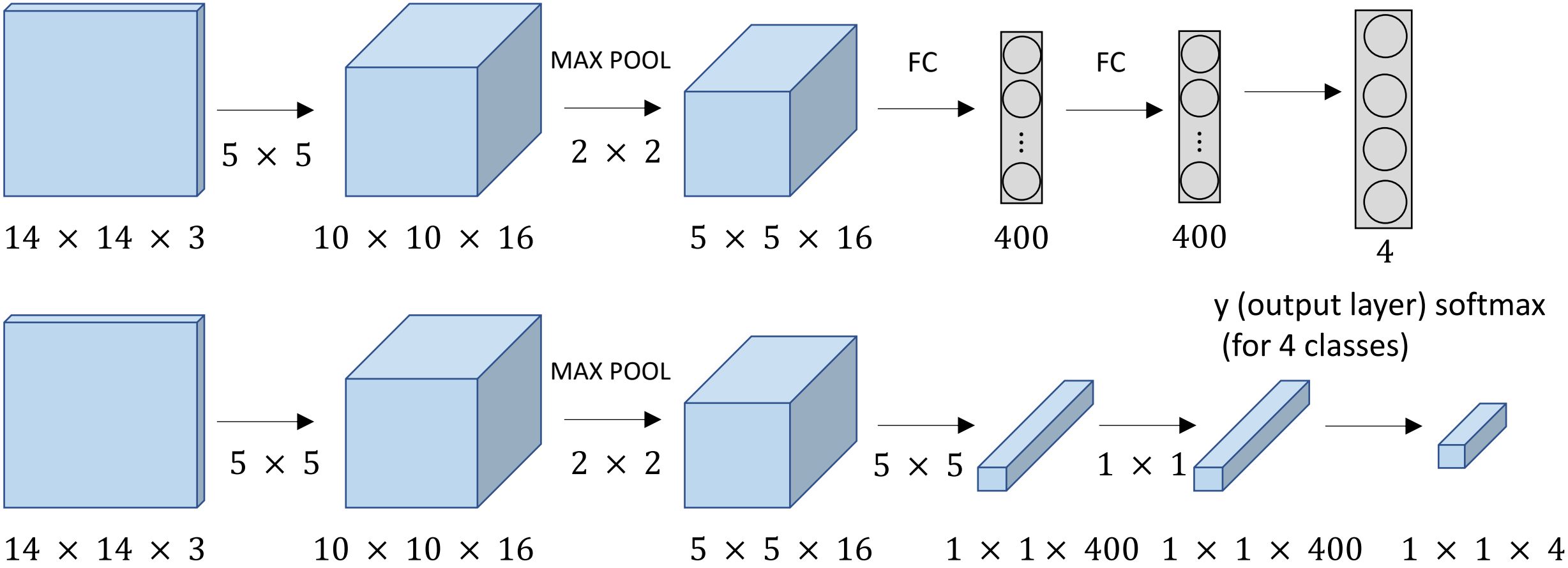
CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3
FC6
FC7
FC8

Fully convolutional network

- No feature flattening
- No fully connected layers

- Trick
  - Reduce the spatial dimension to 1x1
  - Use 1x1 convolution for prediction
  - In a way 1x1 convolution is a fully connected layer

Converting FC into conv

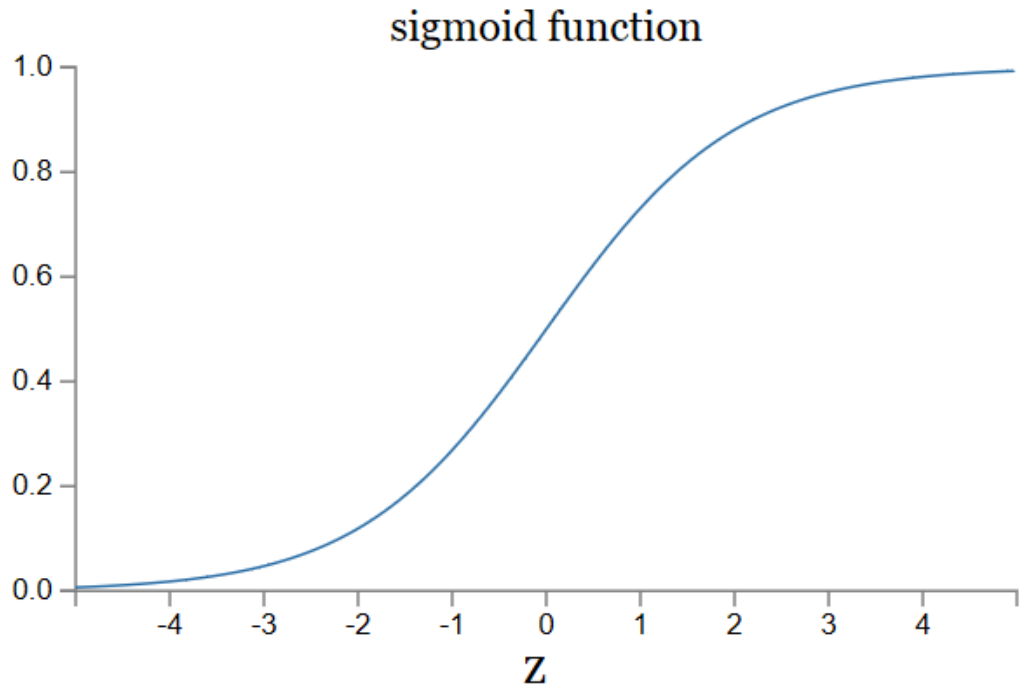


Credit: Sedat Ozer

Let's introduce non-linearities

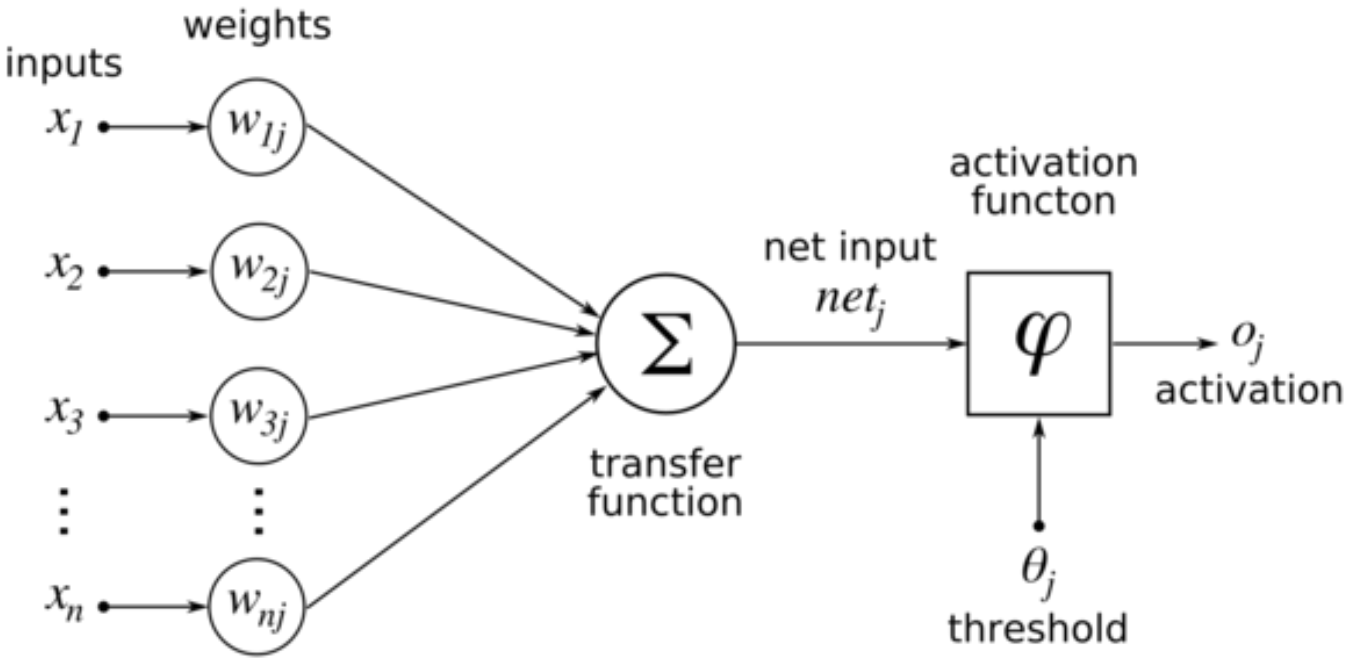
We're going to introduce non-linear functions to transform the features.

$$\sigma(z) \equiv \frac{1}{1+e^{-z}}.$$



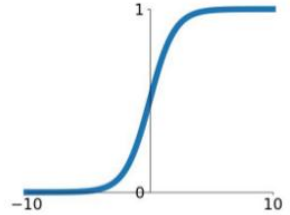
Nielsen

Activation Functions



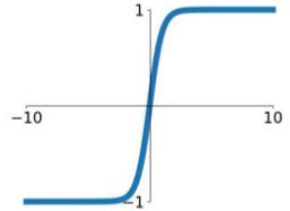
Sigmoid

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



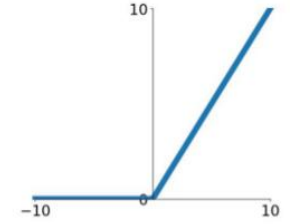
tanh

$$\tanh(x)$$



ReLU

$$\max(0, x)$$

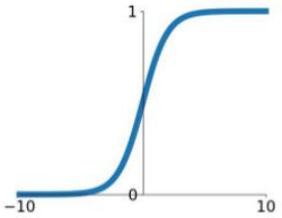


Activation Functions

- Uses in the final prediction
- Target range?
  - Sigmoid: (0, 1)
  - Tanh: (-1, 1)
- Normalized image
  - (-1, 1)
  - (0, 1)
  - (0, INF)

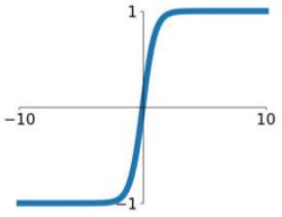
Sigmoid

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



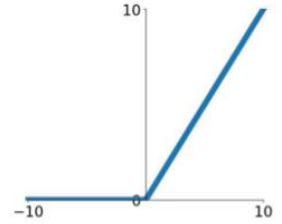
tanh

$$\tanh(x)$$



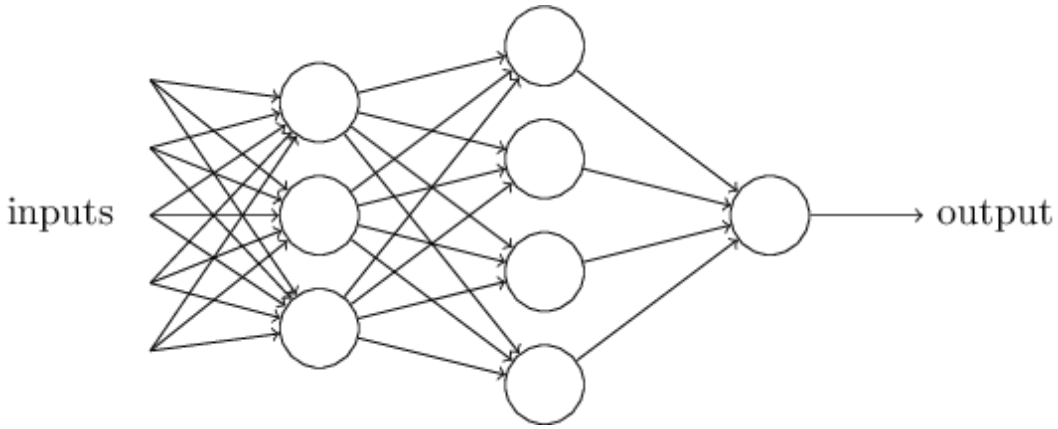
ReLU

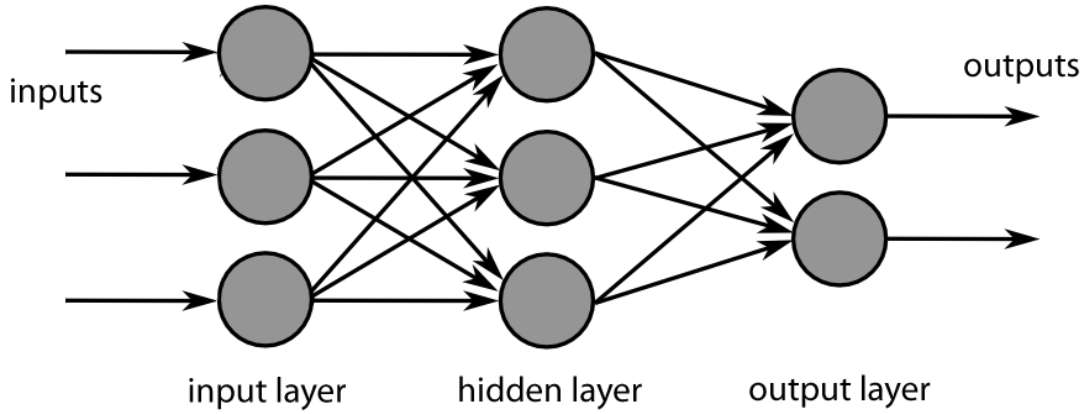
$$\max(0, x)$$



Binary classification

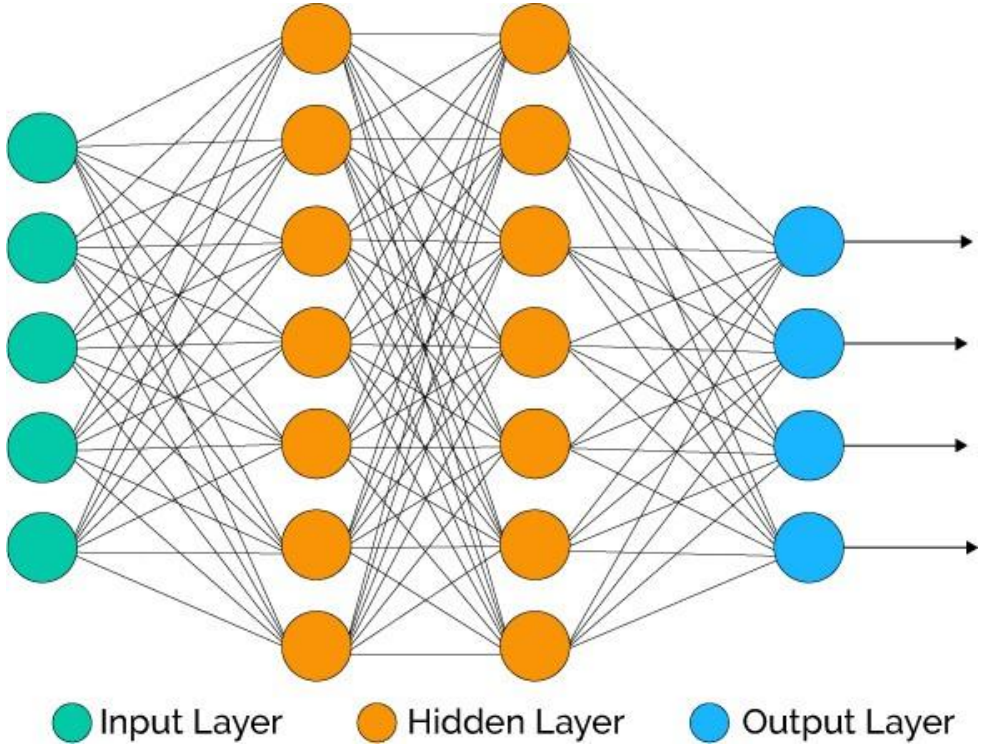
- Target class present or not?
  - Single output
  - Two outputs





Multi-class

- One prediction for each class



Softmax activation



scores = unnormalized log probabilities of the classes.

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

where $s = f(x_i; W)$

cat	**3.2**
car	5.1
frog	-1.7

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax activation



scores = unnormalized log probabilities of the classes.

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

where $s = f(x_i; W)$

cat	**3.2**
car	5.1
frog	-1.7

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax activation



scores = unnormalized log probabilities of the classes.

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

where $s = f(x_i; W)$

cat	**3.2**	exp →	**24.5**
car	5.1		164.0
frog	-1.7		0.18

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

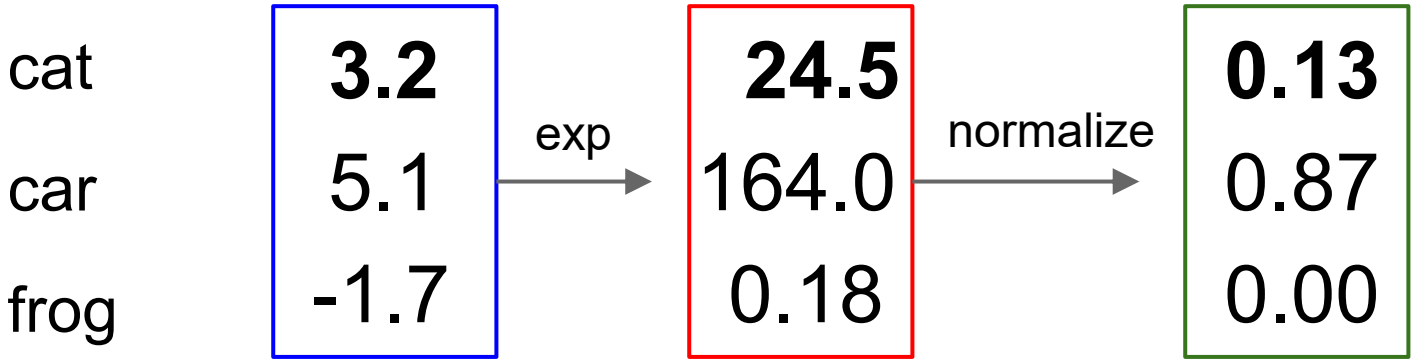
Softmax activation

scores = unnormalized log probabilities of the classes.

$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

where $s = f(x_i; W)$

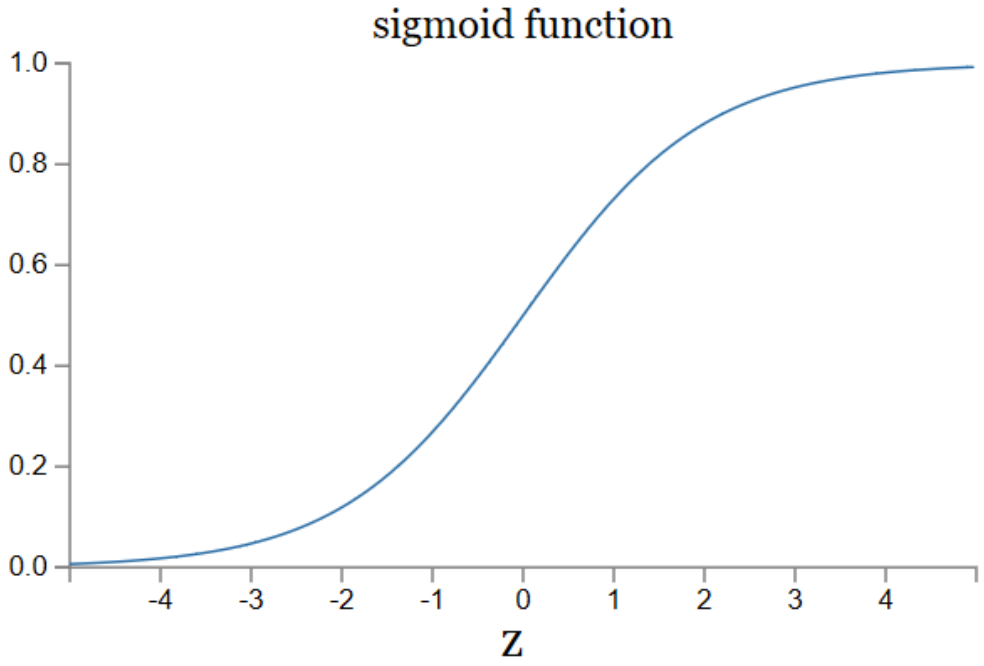




Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Multi-label

- Multiple classes can be active
- Softmax will not work
- Use sigmoid activation



Loss Function

- Way to define how good the network is performing
  - In terms of prediction
- Network training (Optimization)
  - Find the best network parameters to minimize the loss

$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), yi)$$

input

Ground truth

Loss function

network

Network parameters

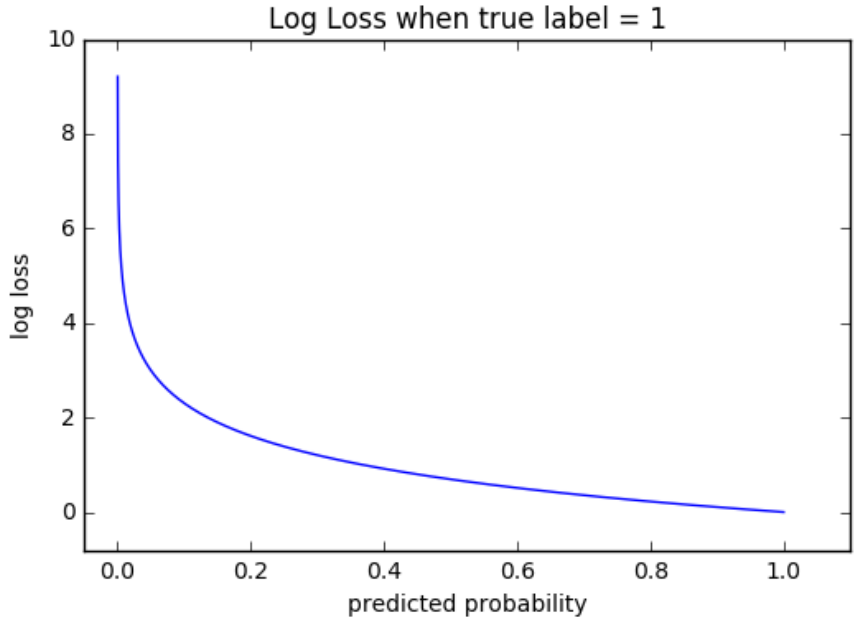
Loss Function

- Cross entropy

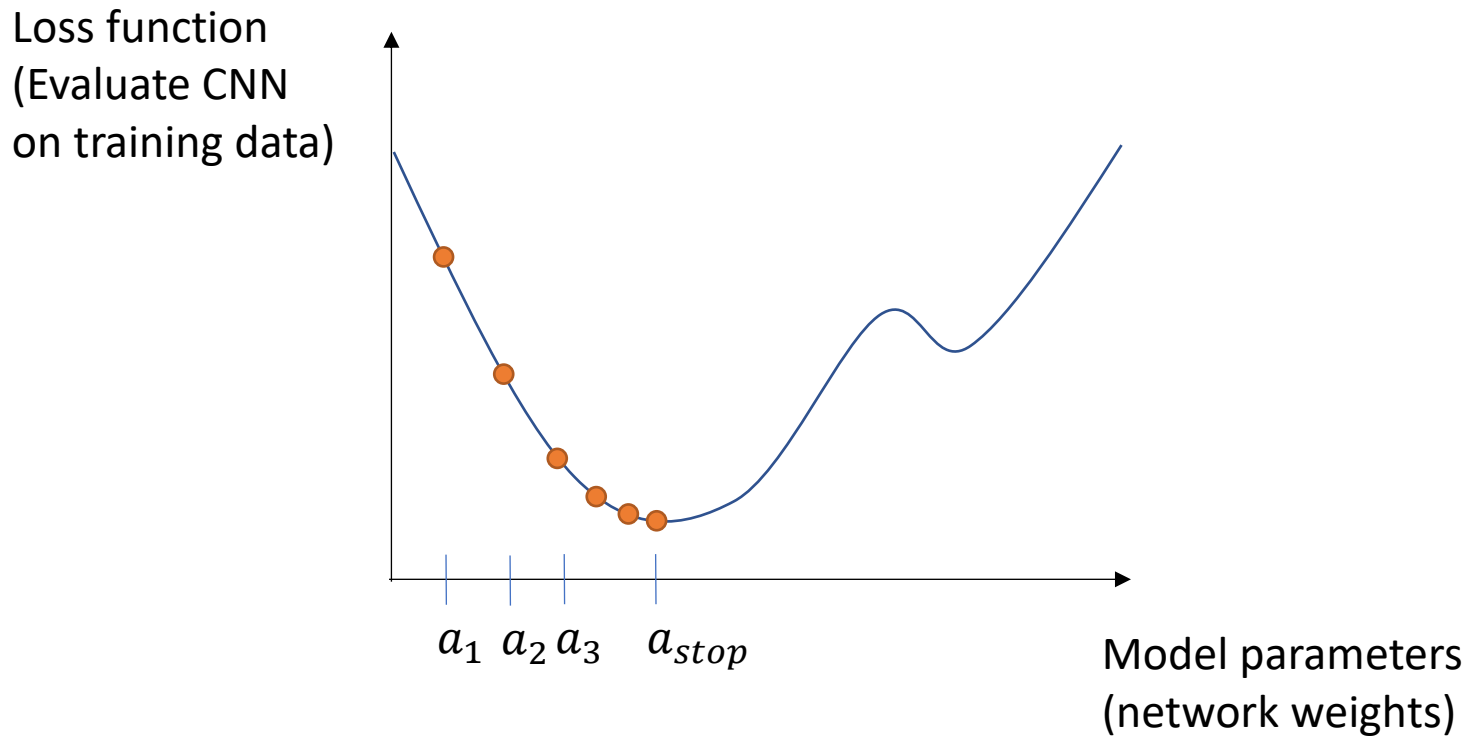
Ground-truth

Predicted value

$$-\frac{1}{N}\sum_{i=1}^{N}(y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i))$$



Train with Gradient Descent

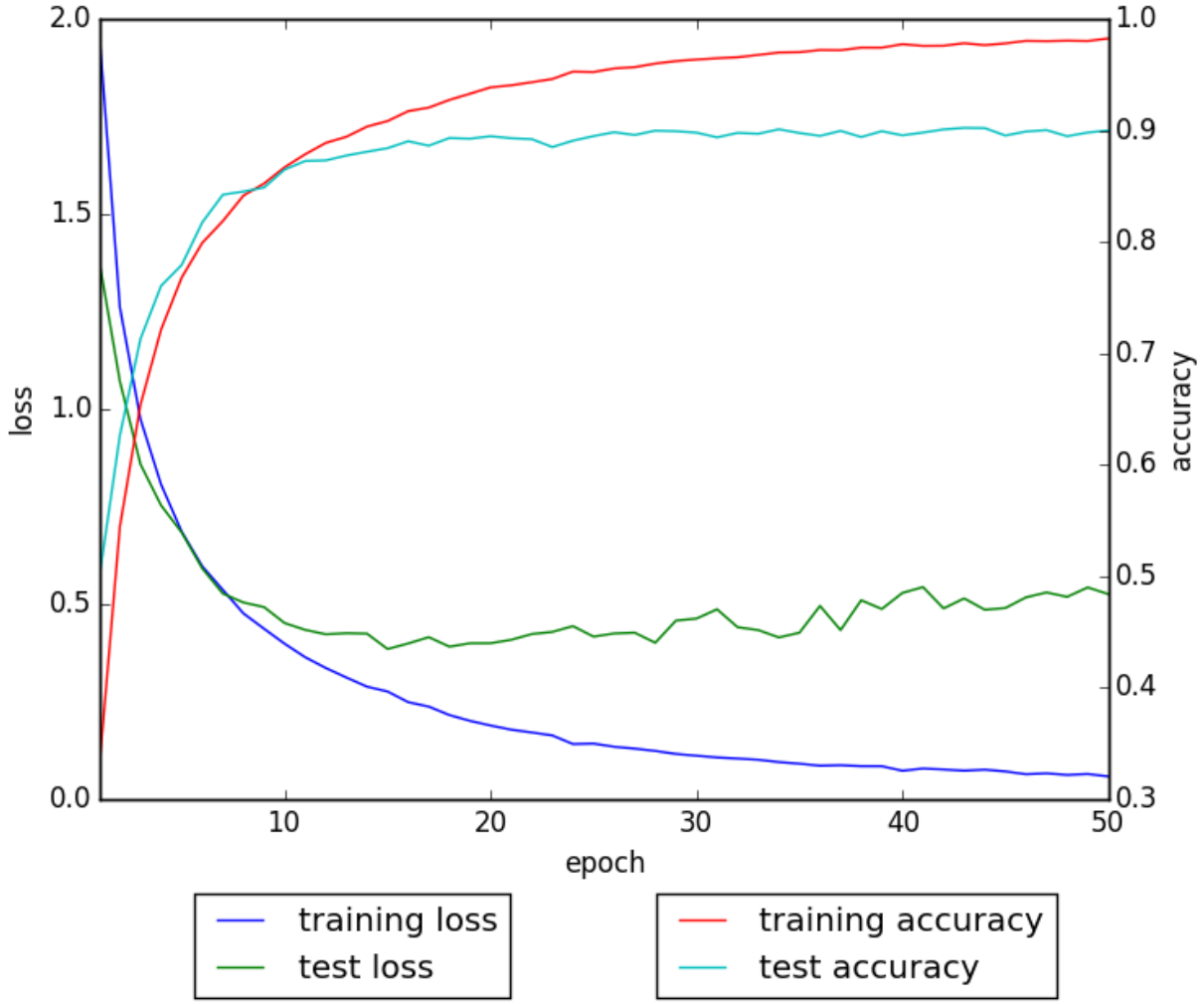


Optimization

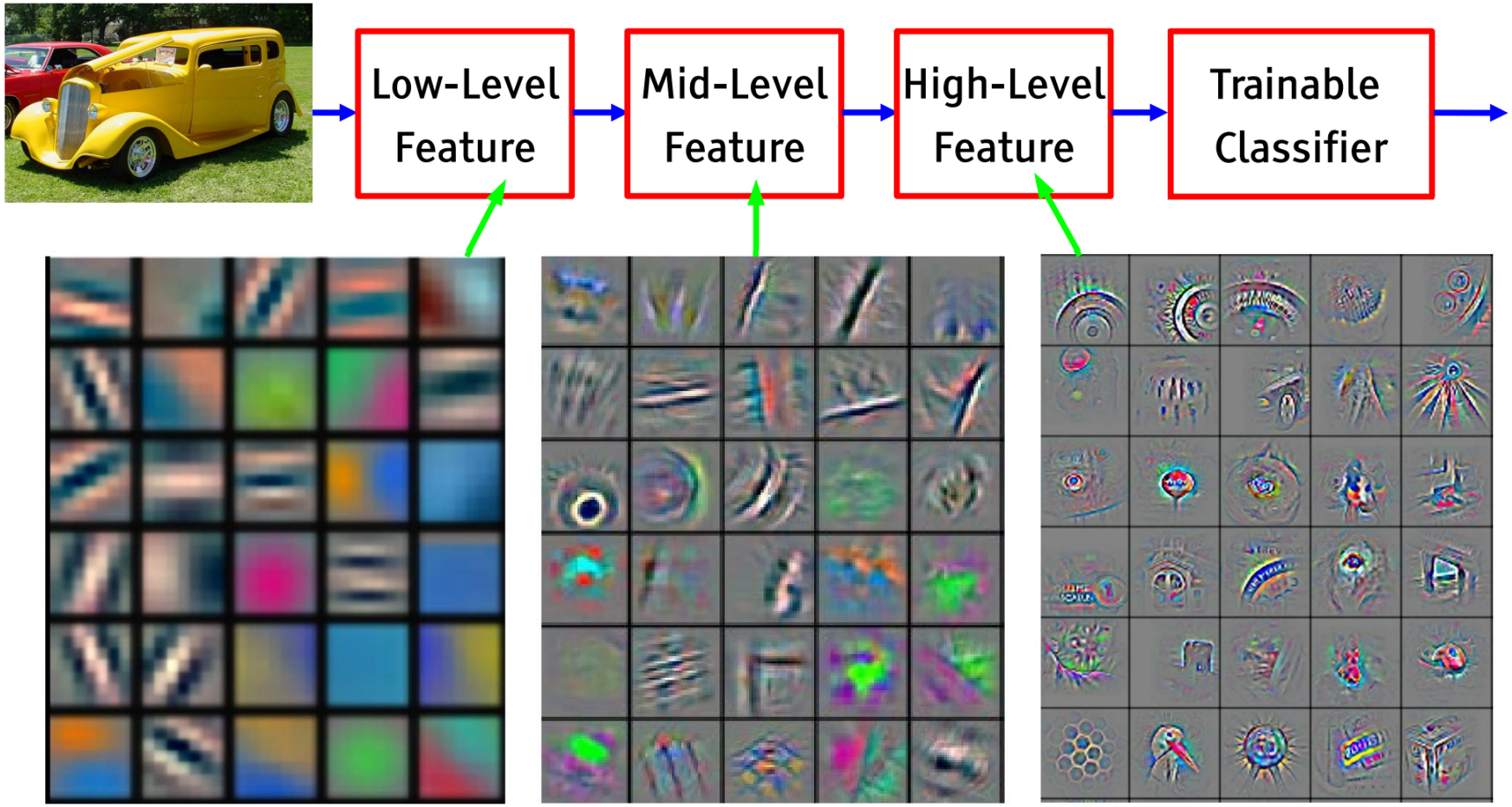
- SGD
- Adam
- AdaDelta
- …

- Ongoing research
- You can define your own loss function

Network training



Visualizing Convolution



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Pop Quiz!

Questions?

Sources for this lecture include materials from works by Abhijit Mahalanobis, James Tompkin, and Fei Fei Li