Deep Learning II: Neural Networks

Hinrich Schütze

Center for Information and Language Processing, LMU Munich

2017-07-21

Overview

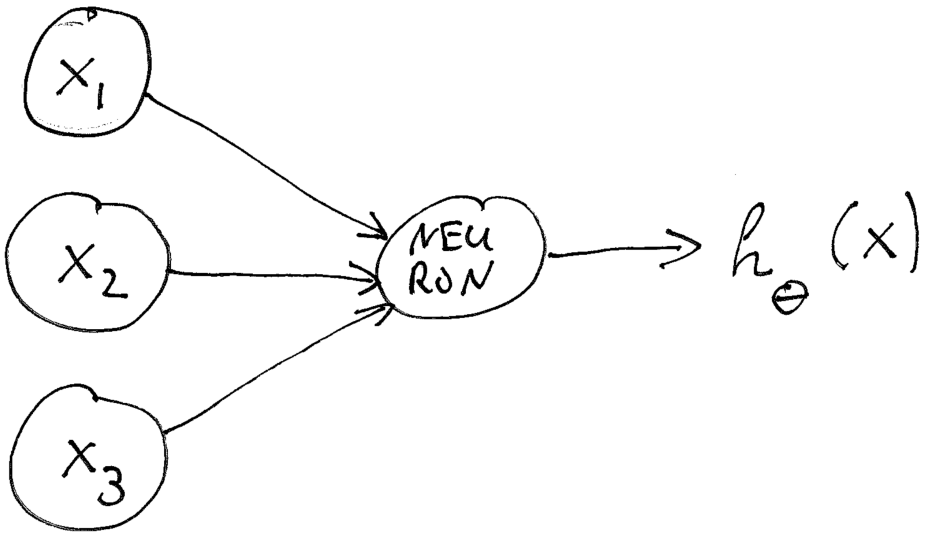
Outline

Inverted Classroom
Andrew Ng: "Machine Learning"
http://coursera.org

Neural networks: Andrew Ng videos

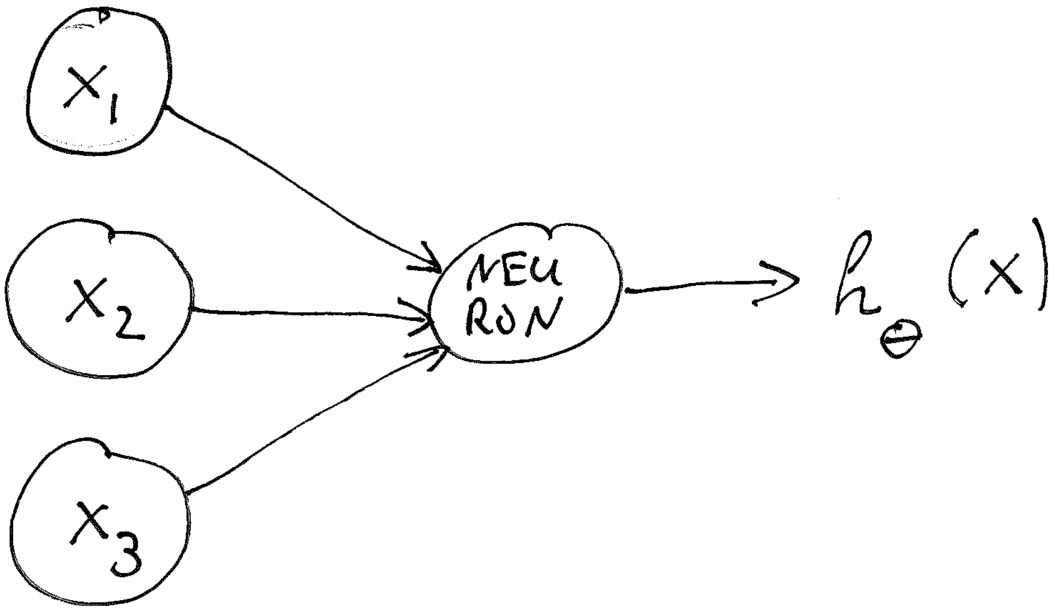
- Model representation I
- Model representation II

A single neuron



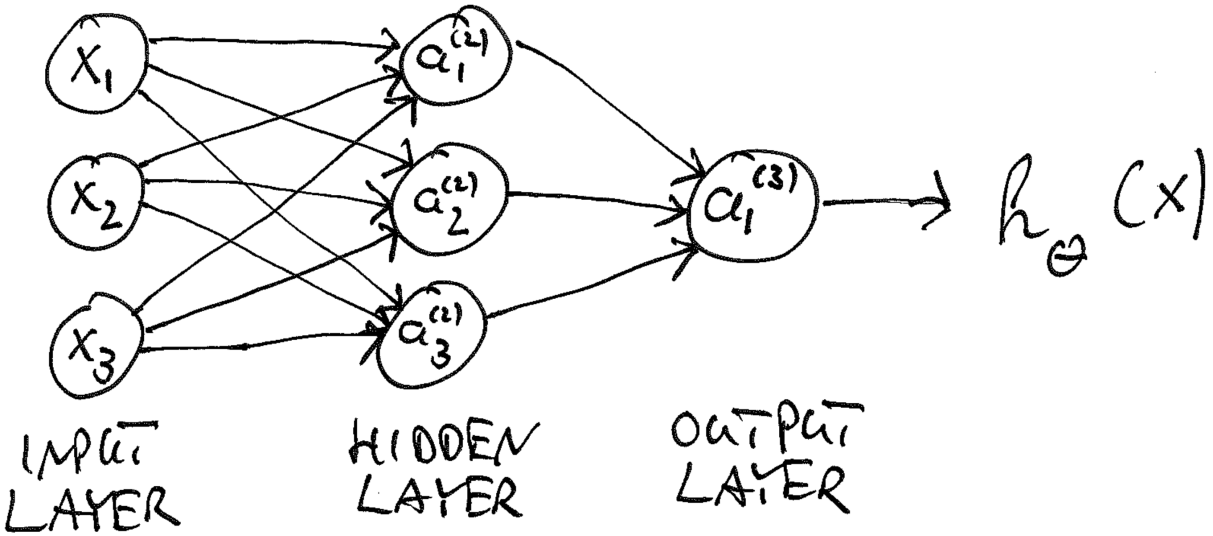
- Input nodes: x_1, x_2, x_3
- Parameters/weights: lines connecting nodes
- Raw input to neuron: weighted sum $\Theta^T \vec{x} = \sum_{i=1}^{3} \theta_i x_i$
- Nonlinear activation function (e.g., sigmoid): $g(\Theta^T \vec{x}) = 1/(1 + \exp(-\Theta^T \vec{x}))$
- Output of neuron: $g(\Theta^T \vec{x})$

A neuron



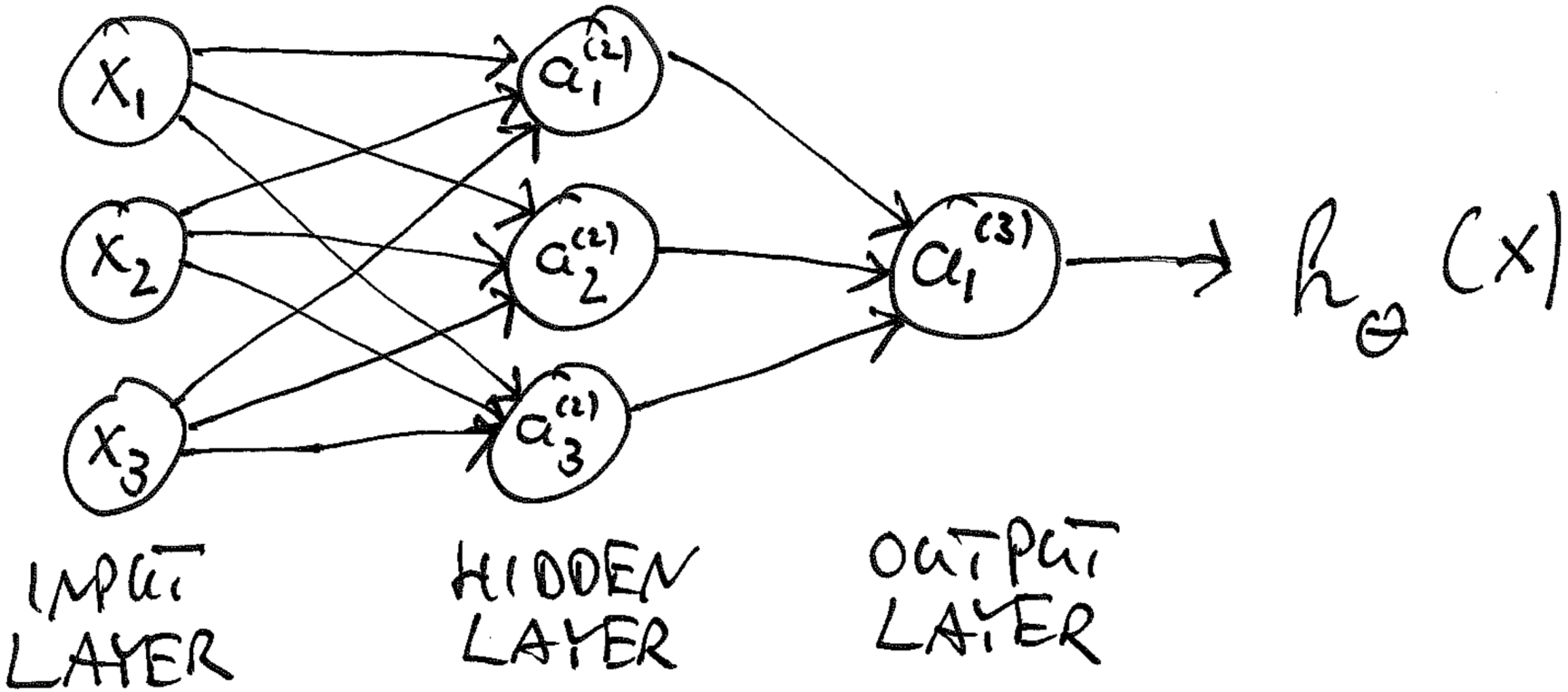
- Inputs: x_1, x_2, x_3
- Parameters (= weights = lines): θ_1, θ_2, θ_3
- Activation function (e.g., sigmoid / logistic)
- Hypothesis: $h_\Theta(\vec{x}) = (\Theta^T \vec{x})$

A neural network

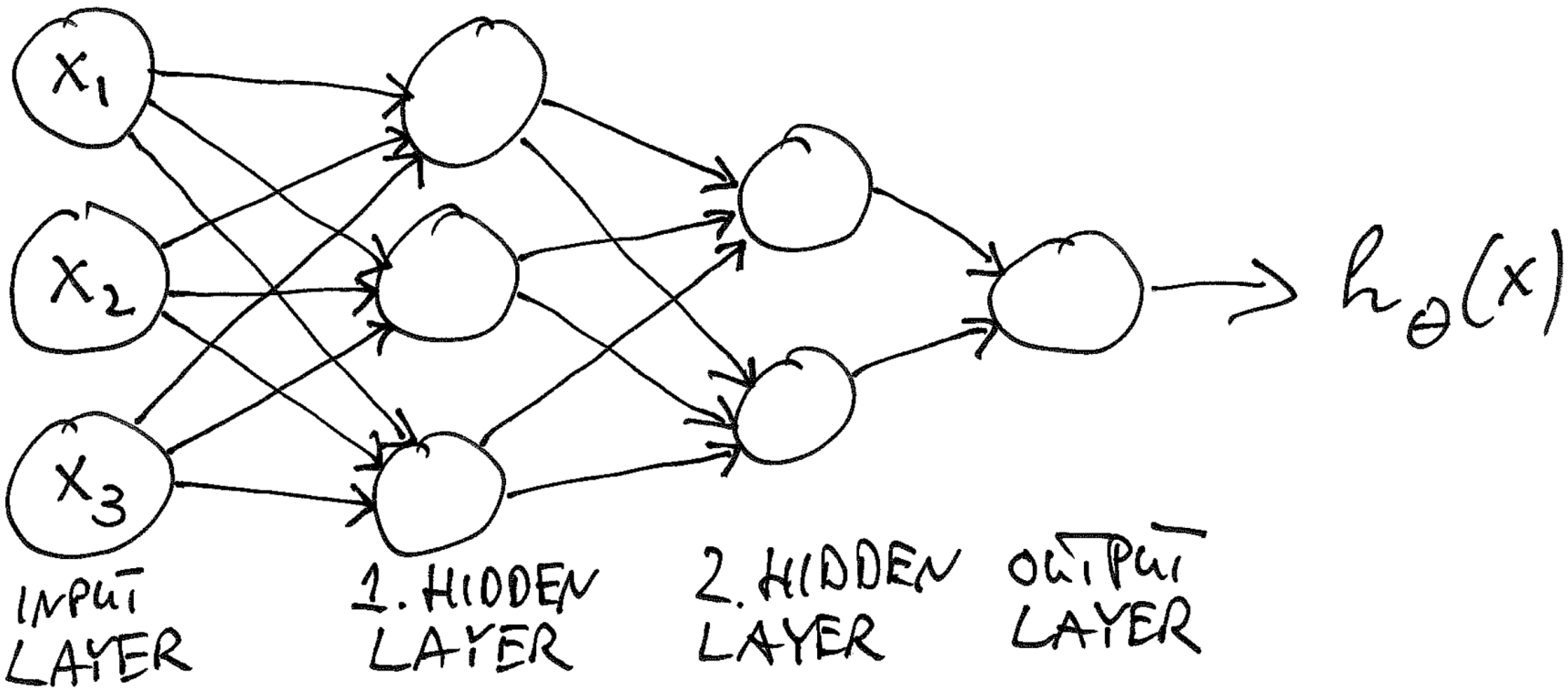


- Input layer (same as before): x_i
- Hidden layer, here: three neurons
- Output layer, here: single neuron
- Activations $a_i^{(k)}$, $k =$ layer
- Full connectivity
- Same or different activation functions

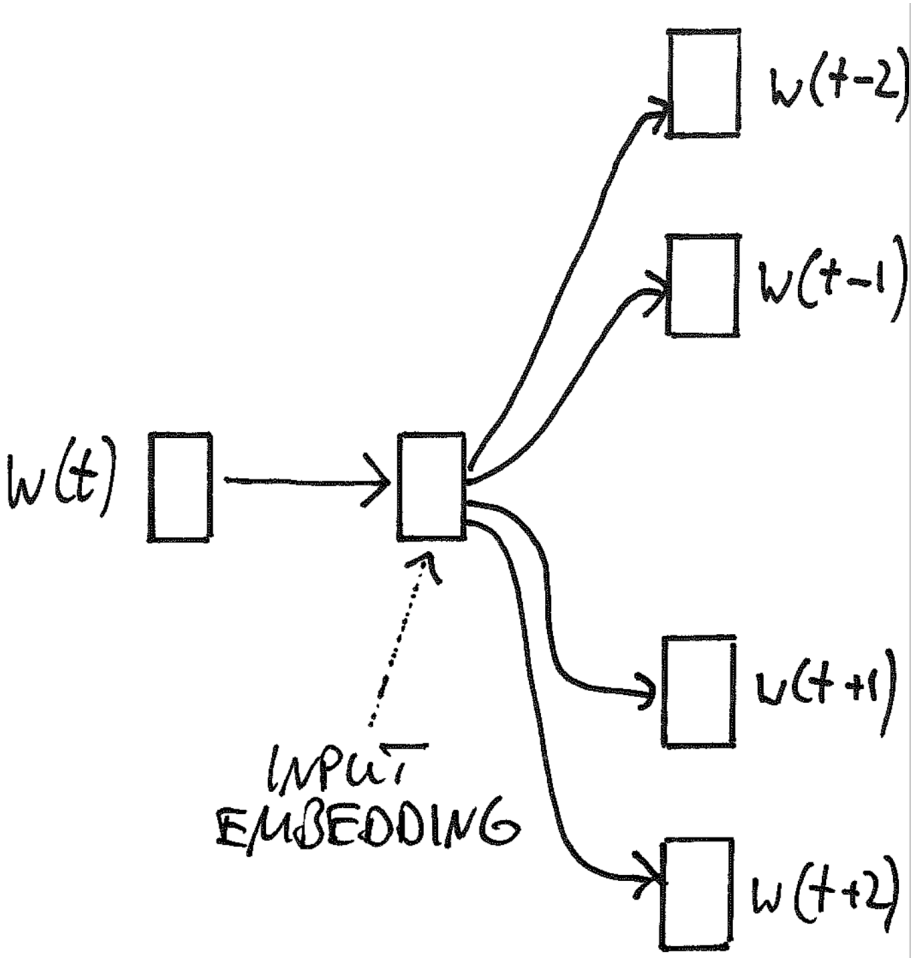
A neural network



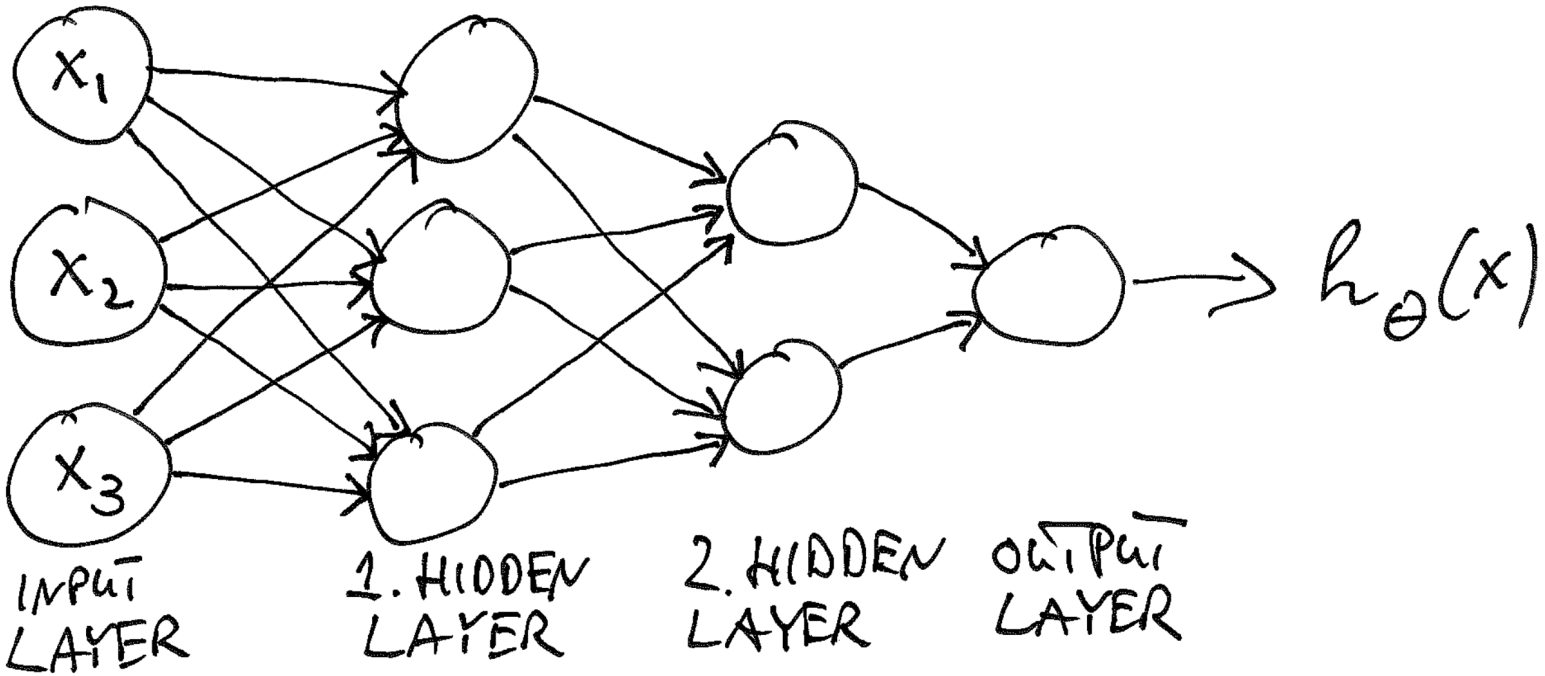
Another neural network architecture



Another neural network architecture



Forward propagation of activity

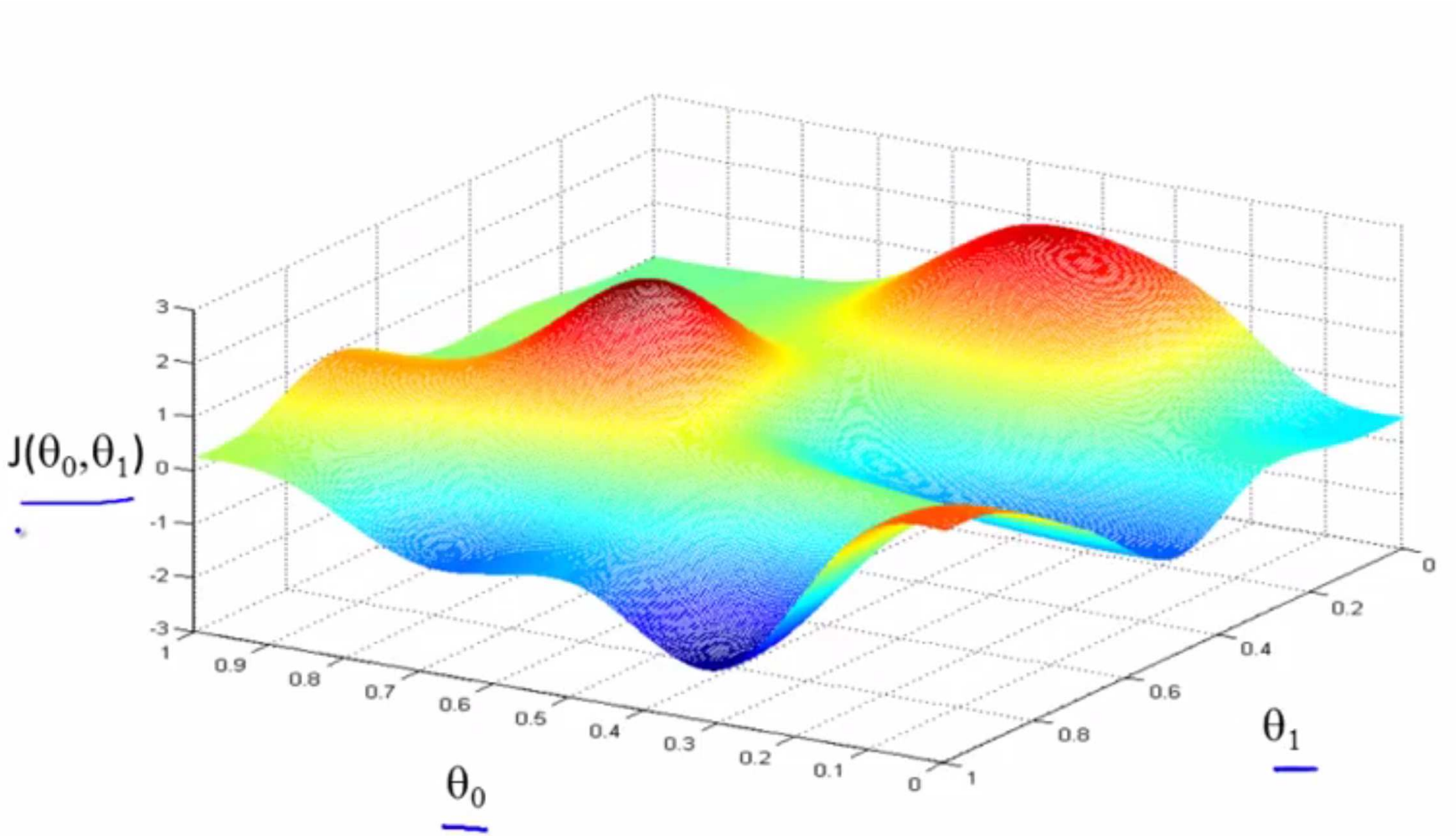


$$a_j^{(i)} = g_i(\Theta_{ij}^T \vec{a}^{\,(i-1)}) = 1/(1 + \exp(-\Theta_{ij}^T \vec{a}^{\,(i-1)}))$$

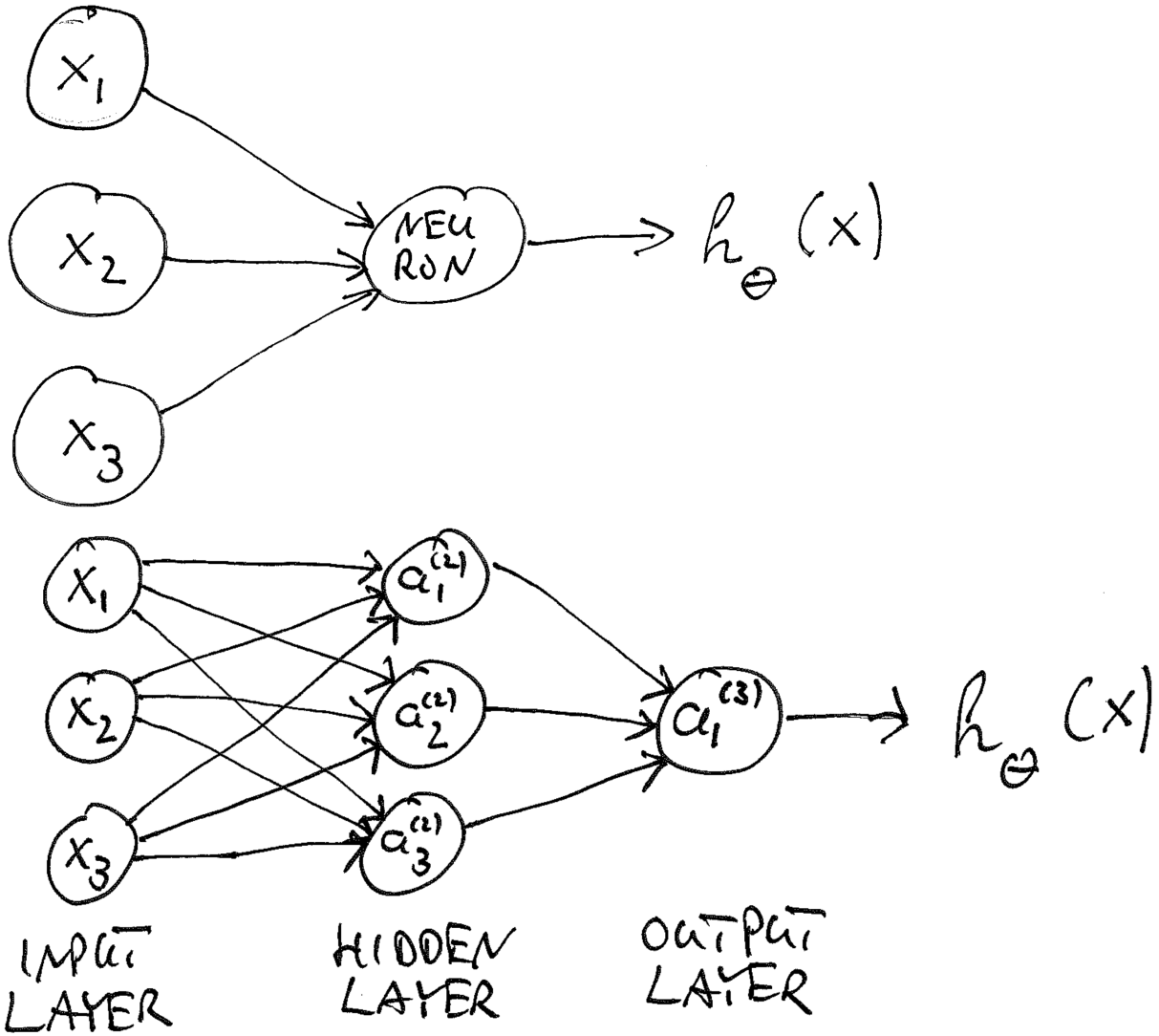
Learning/Training: Backpropagation

- As before: cost function
- As before: objective
  (find parameters that minimize cost)
- As before: gradient descent
- That is: compute gradient and move along gradient
- What's new:
  We use backpropagation to compute the gradient.

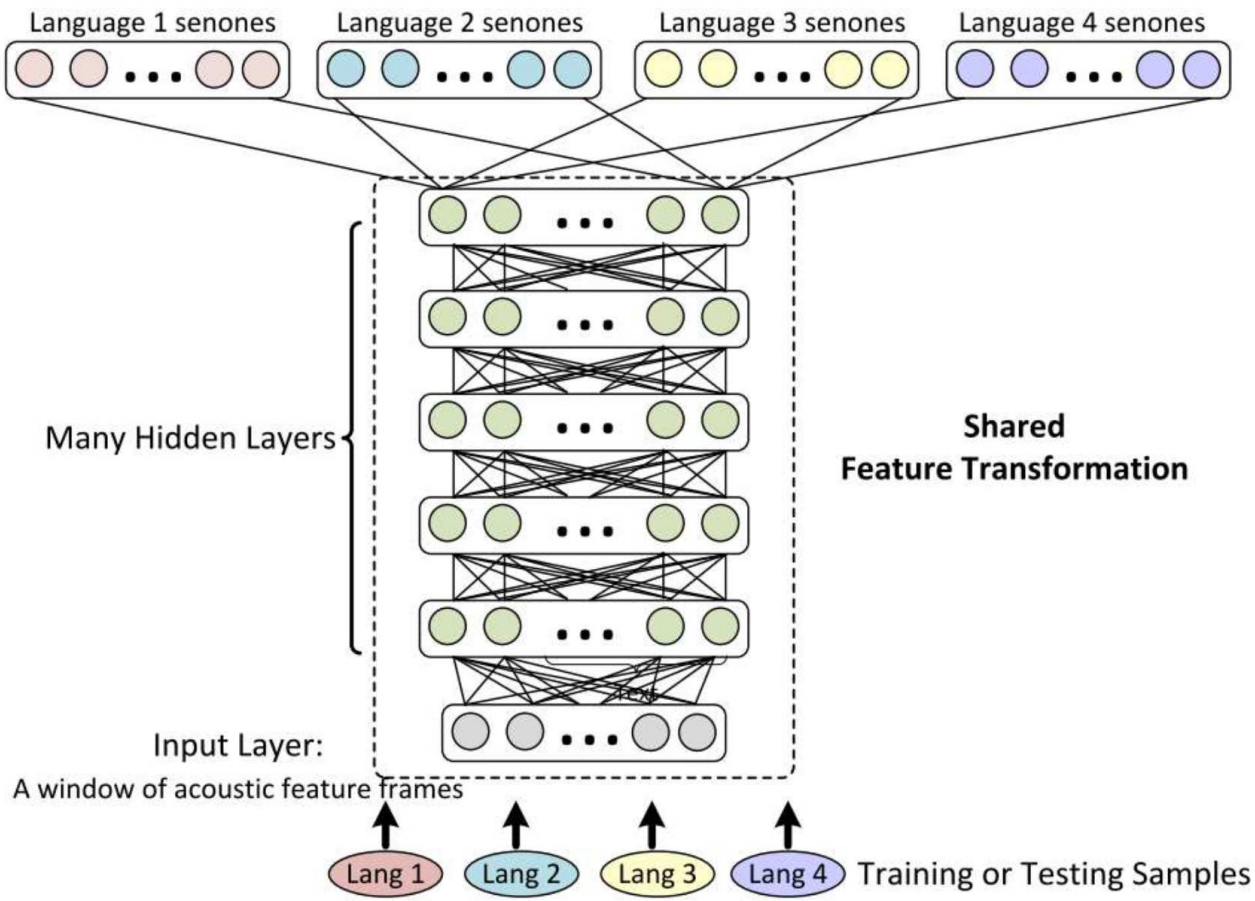
Gradient descent



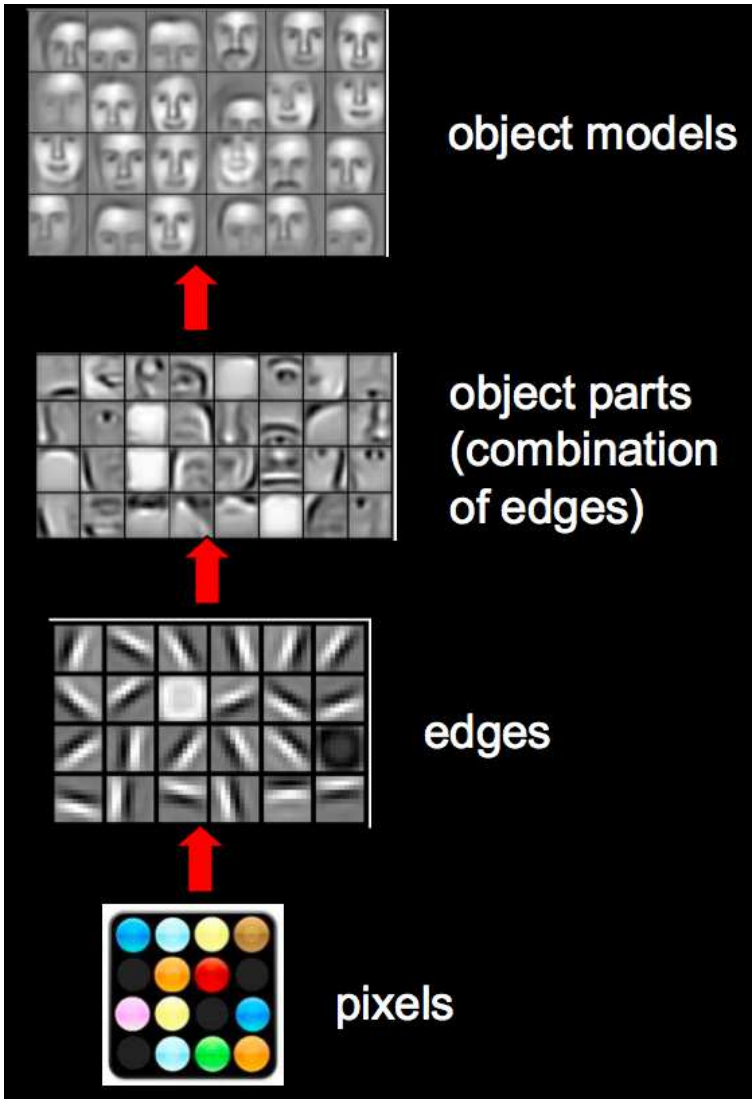
Neurons can be trained to detect features.



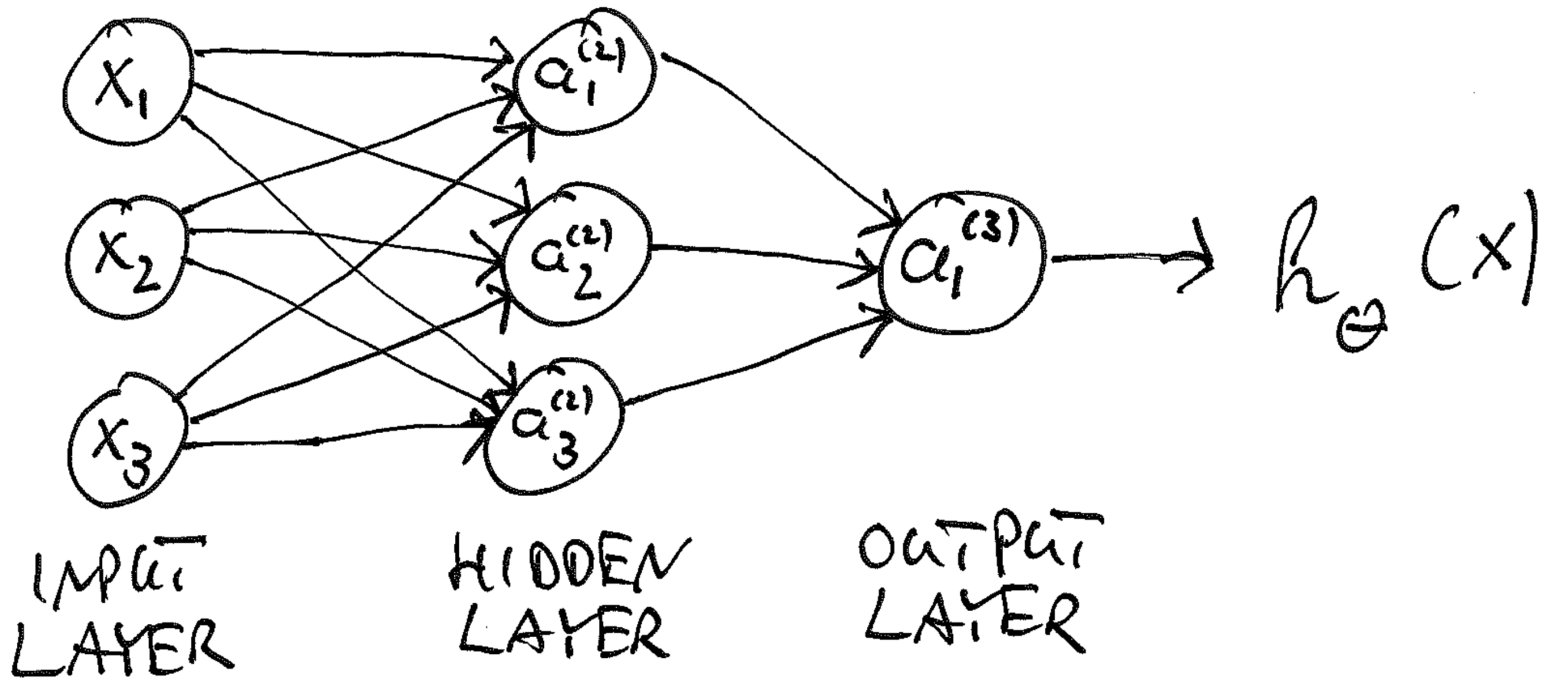
Deep learning:
Each layer learns more powerful/abstract features.



Increasingly abstract features in vision

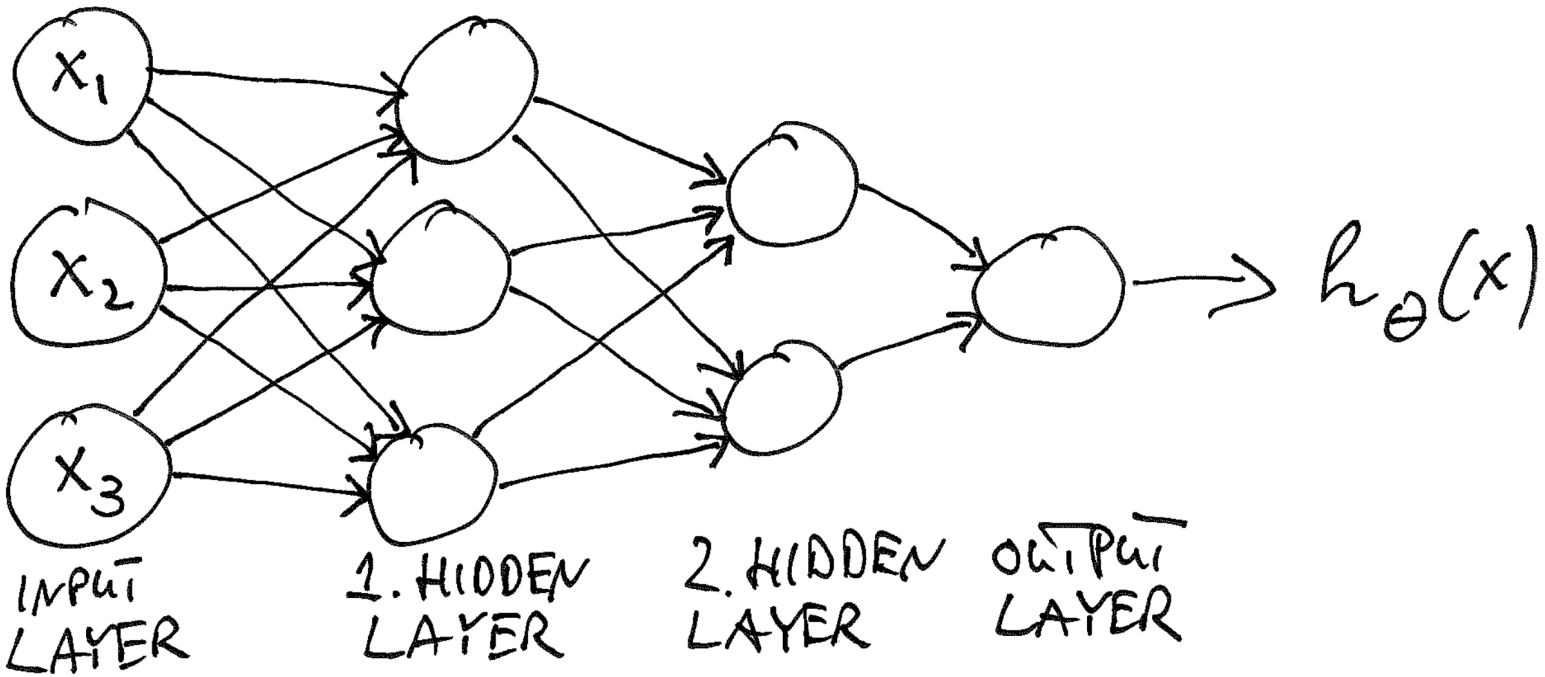


Number of weights/parameters



$$|L_1| * |L_2| + |L_2| * |L_3| = 3 \cdot 3 + 3 \cdot 1 = 12$$

Exercise: Number of weights/parameters



Task: Does this sentence mention an officer leaving?

Given: A sentence

Workforce Solutions Alamo fired CEO John Hathaway yesterday.

Binary classification task

Class "yes": This sentence contains information about an officer leaving a company (so a financial analyst should look at it).
Class "no": This sentence does not contain information about an officer leaving a company (so nobody has to look at it).

Correct class in this case?

Class "yes": This sentence contains information about an officer leaving a company.
Class "no": This sentence does not contain information about an officer leaving a company.

Task: Does this sentence mention an officer leaving?

Given: A sentence

CEO John Hathaway fired his gardener yesterday.

Correct class in this case?

Class "yes": This sentence contains information about an officer leaving a company.

Class "no": This sentence does not contain information about an officer leaving a company.

Task: Does this sentence mention an officer leaving?

Given: A sentence

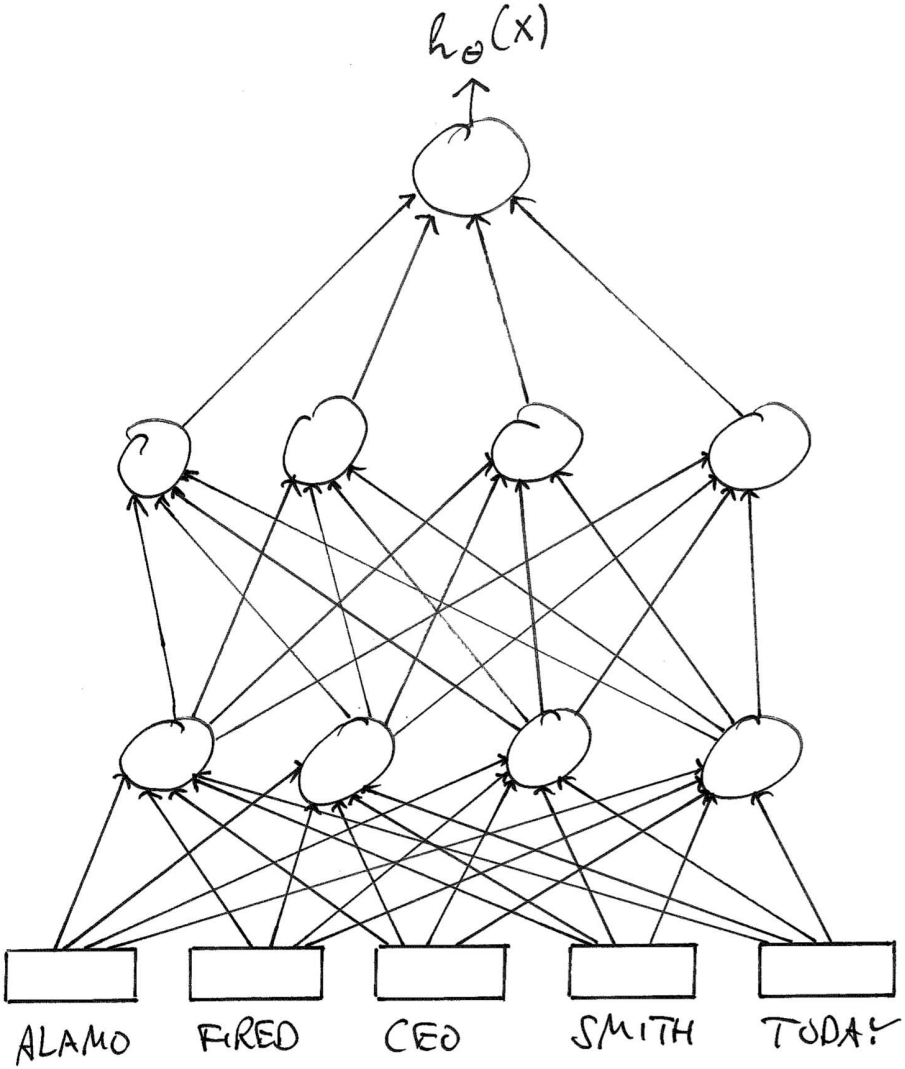
This picture shows parting CEO Cook talking with ex-CFO Dyer.

Correct class in this case?

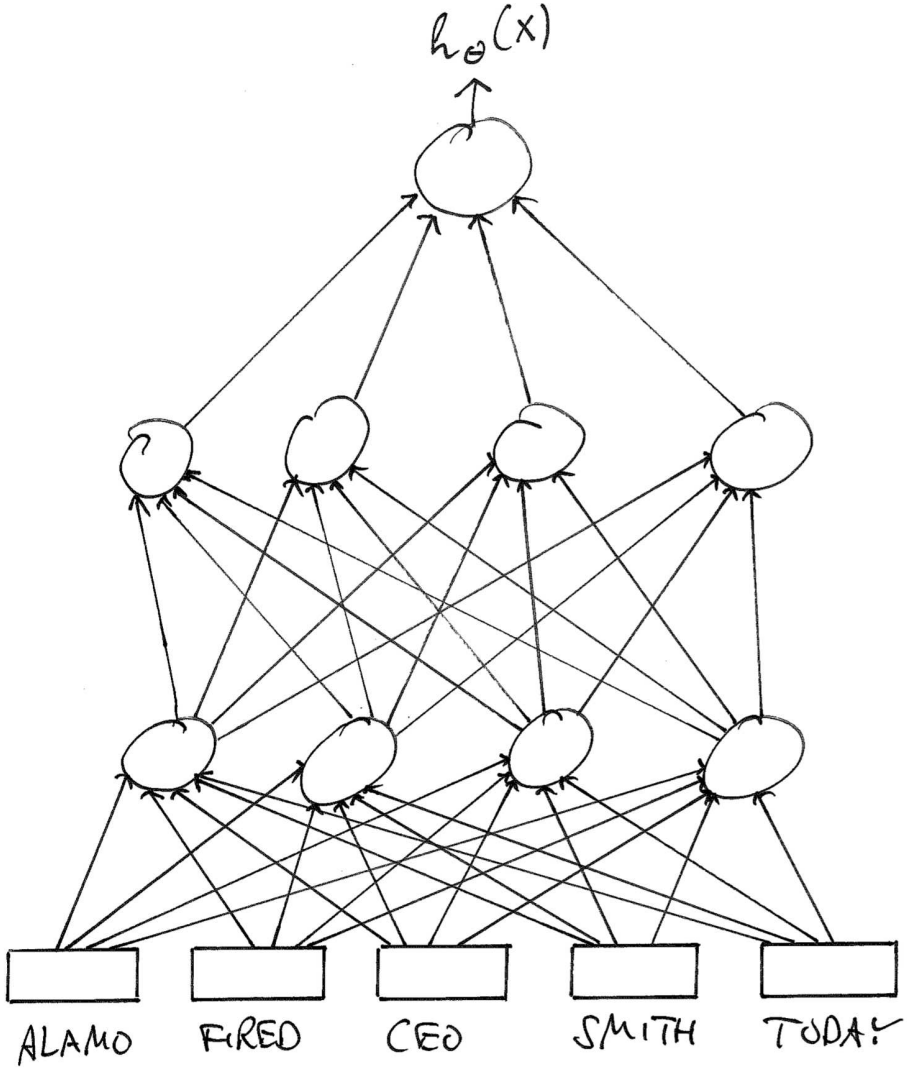
Class "yes": This sentence contains information about an officer leaving a company.

Class "no": This sentence does not contain information about an officer leaving a company.

Simple architecture for detecting leaving events



Hypothesis? Parameters? Cost? Objective?



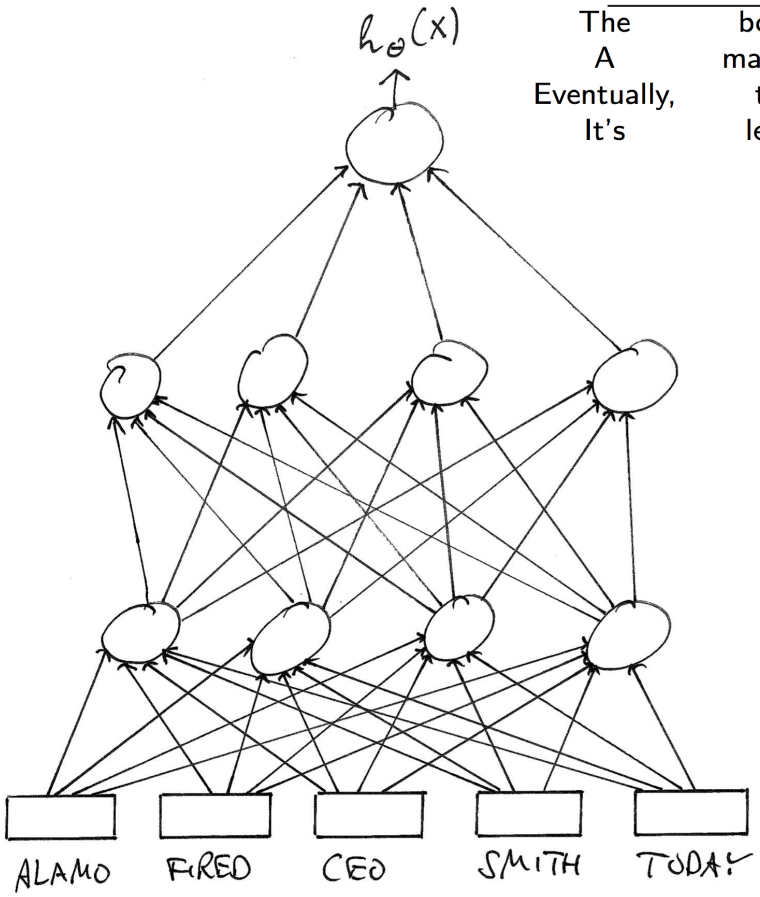
Simplest architecture: Fixed-length input → Padding

1	2	3	4	5	6	7	8	9
The	board	forced	him	to	resign	PAD	PAD	PAD
A	majority	of	the	board	forced	him	to	quit
Eventually	the	escalation	led	him	to	resign	PAD	PAD
It's	legal	threats	that	compelled	him	to	leave	PAD

key idea of convolution:
learn a filter for the pattern
"[force] [pronoun] to [leave]"
filter = feature detector

Exercise

If you use this architecture, why is it hard to learn the filter "[force] [pronoun] to [leave]"?



1	2	3	4	5	6	7	8	9
The	board	forced	him	to	resign	PAD	PAD	PAD
A	majority	of	the	board	forced	him	to	quit
Eventually,	the	escalation	led	him	to	resign	PAD	PAD
It's	legal	threats	that	compelled	him	to	leave	PAD

Outline

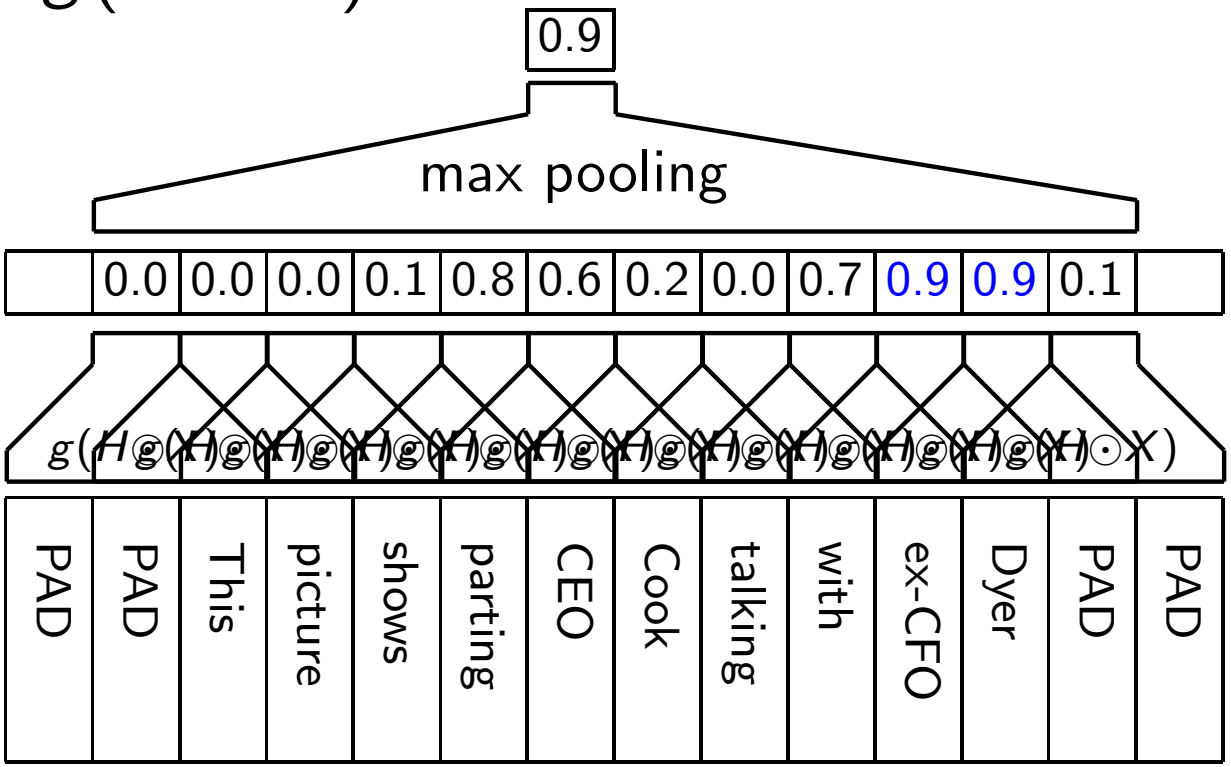
Use convolution&pooling architecture Input layer

selects max feature

pooling layer

computes features

convolution layer

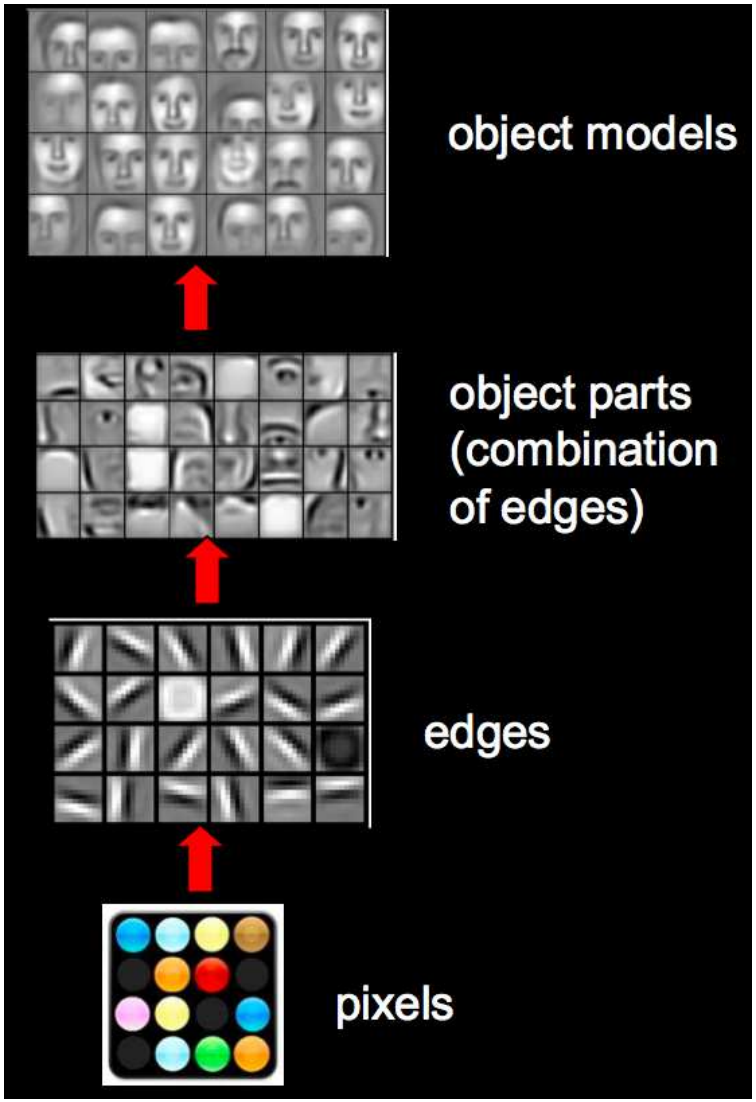


input layer

Convolution & pooling

- Widely used in vision
- Recent development: widely used in NLP
- Best example of successful transfer from vision to NLP

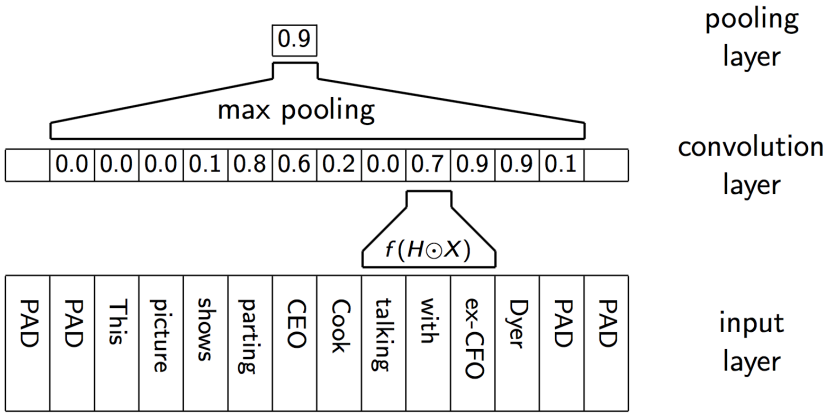
Convolution and max pooling in vision



Exercise

Try to find a good example of a typical NLP task for which max pooling (i.e., detecting whether or not a particular type of thing occurs in a sentence) is the wrong approach.

(Alternatively, try to find a good example of a typical vision task for which max pooling (i.e., detecting whether or not a particular type of thing occurs in a scene) is the wrong approach.)



Convolutional filter H

- $a = g(H \odot X)$
- Kernel size k: length of subsequence
- H is applied to every subsequence of length k.
- X is the representation of the subsequence, of dimensionality $D \times k$.
- D is the dimensionality of the embeddings.
- H also has dimensionality $D \times k$.
- $\odot$ is the (Frobenius) inner product: $H \odot X = \sum_{(i,j)} H_{ij} X_{ij}$
- g: nonlinearity (e.g., sigmoid)

Notation

V	vocabulary size
D	embedding dimensionality
C	number of classes
C_{i}	number of input channels
C_{o}	number of output channels
Ks	kernel sizes
N	minibatch size
W	padded sequence length