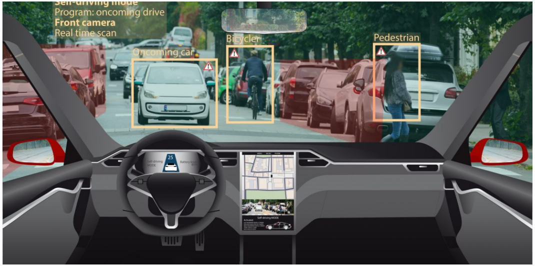
Introduction to Deep Learning

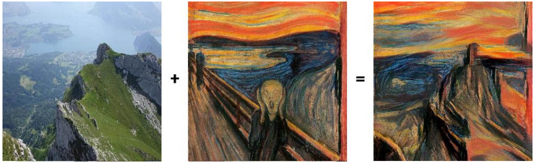
Deep Learning Workshop at MBL

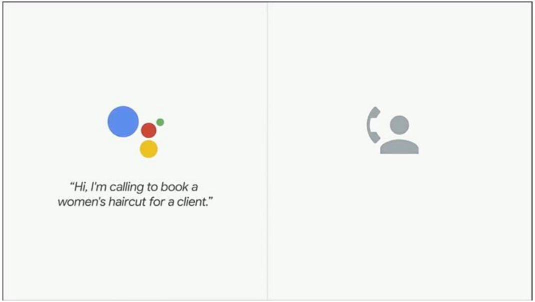
http://funkey.science/deep_learning_mbl.pdf

May 11, 2019


KE JIE
00:46:57

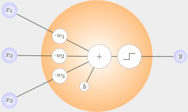

Program: oncoming drive
Front camera
Real time scan
Oncoming car
Bicycler
Pedestrian



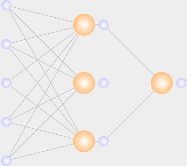

“Hi, I'm calling to book a women's haircut for a client.”

Outline

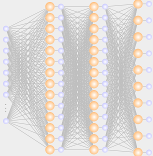
1. Perceptrons



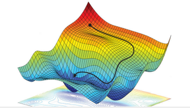
2. Networks



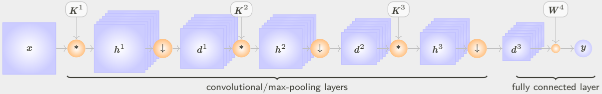
3. Deep Networks



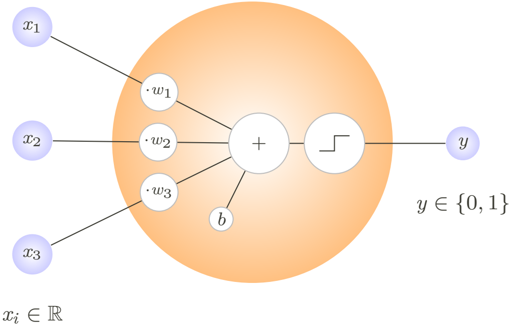
4. Learning



5. Convolutional Networks

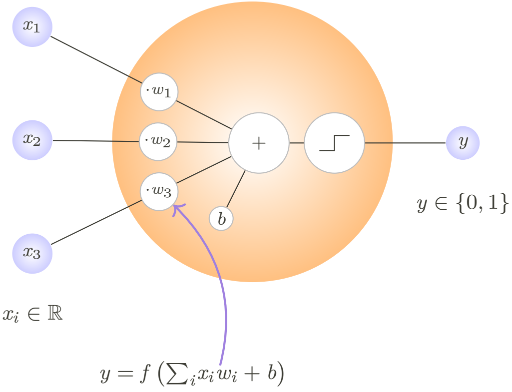


The Building Block of Neural Networks



Perceptron

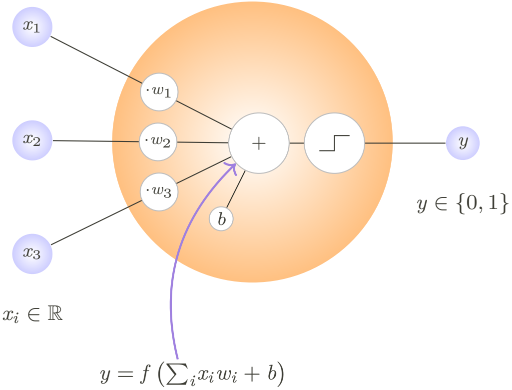
- invented by McCulloch and Pitts in 1943
- termed "perceptron" by Rosenblatt in 1958
- also known as: Threshold Logic Unit, Linear Threshold Unit, McCulloch-Pitts model



Perceptron

- invented by McCulloch and Pitts in 1943
- termed "perceptron" by Rosenblatt in 1958
- also known as: Threshold Logic Unit, Linear Threshold Unit, McCulloch-Pitts model

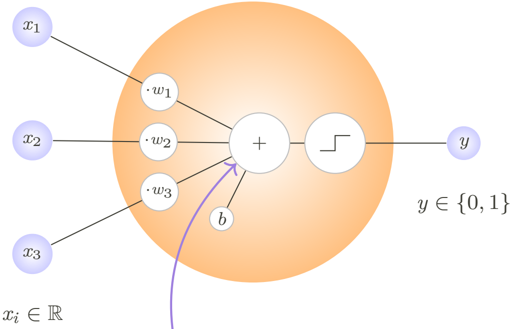
The Building Block of Neural Networks



Perceptron

- invented by McCulloch and Pitts in 1943
- termed "perceptron" by Rosenblatt in 1958
- also known as: Threshold Logic Unit, Linear Threshold Unit, McCulloch-Pitts model

The Building Block of Neural Networks

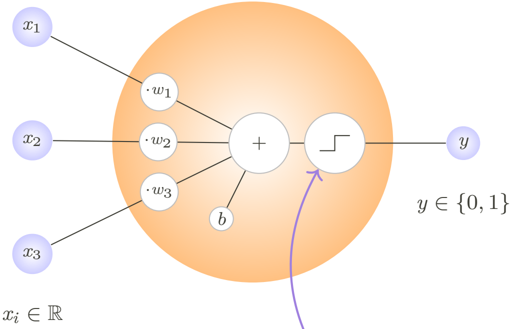


$$y = f\left(\sum_i x_i w_i + b\right) = f(\boldsymbol{x}^\intercal \boldsymbol{w} + b)$$

Perceptron

- invented by McCulloch and Pitts in 1943
- termed "perceptron" by Rosenblatt in 1958
- also known as: Threshold Logic Unit, Linear Threshold Unit, McCulloch-Pitts model

The Building Block of Neural Networks

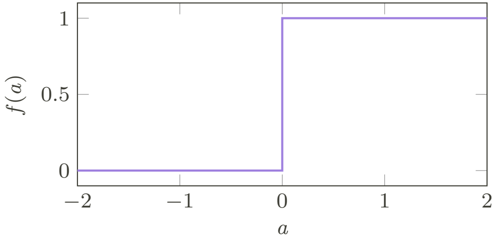


$$y = f\left(\sum_i x_i w_i + b\right) = f(\boldsymbol{x}^\intercal \boldsymbol{w} + b)$$

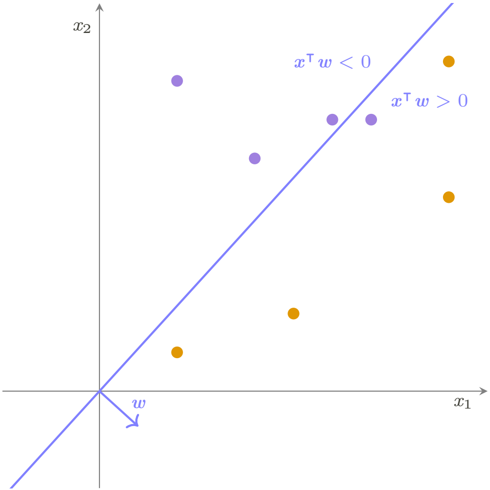
$$f(a) = \begin{cases} 0 & \text{if } a \leq 0 \\ 1 & \text{otherwise} \end{cases}$$

Perceptron

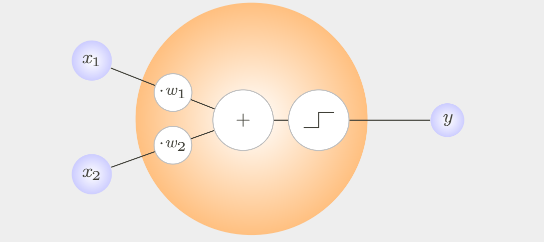
- invented by McCulloch and Pitts in 1943
- termed "perceptron" by Rosenblatt in 1958
- also known as: Threshold Logic Unit, Linear Threshold Unit, McCulloch-Pitts model



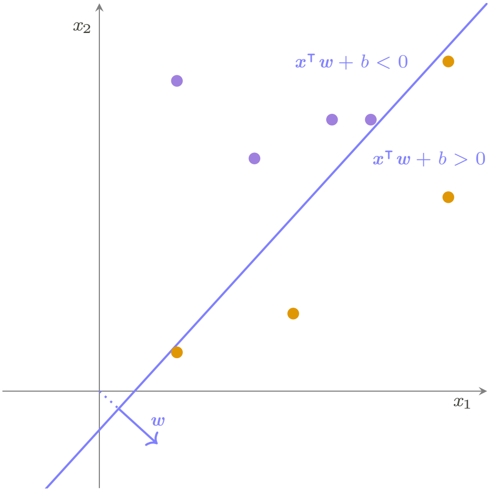
Just a Linear Classifier



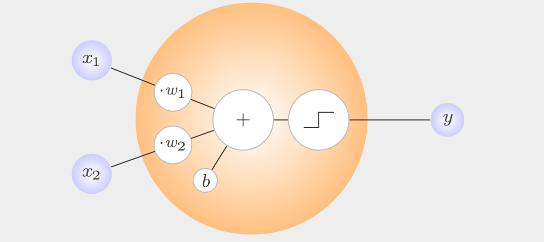
Example with 2 inputs



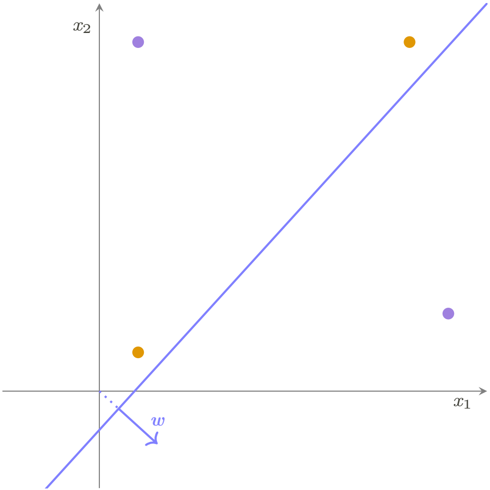
Just a Linear Classifier



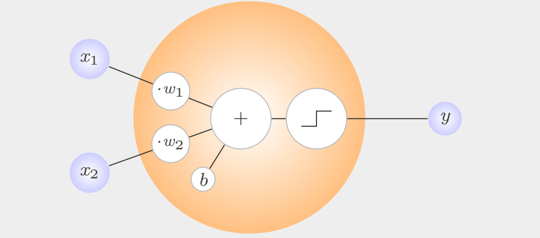
Example with 2 inputs



Just a Linear Classifier



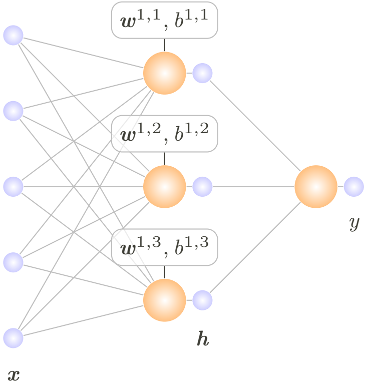
Example with 2 inputs



The XOR Affair and AI Winter

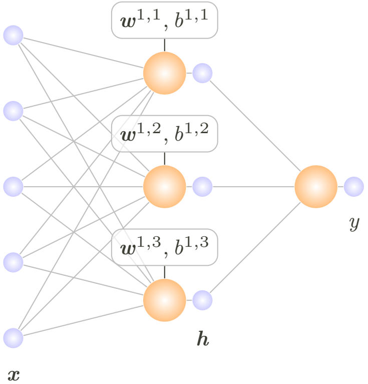
- "The perceptron may eventually be able to learn, make decisions, and translate languages." –Rosenblatt, 1958
- "Some functions are tricky." –Minsky/Papert, 1969
- interest in neural networks declined until the 80s

[1]



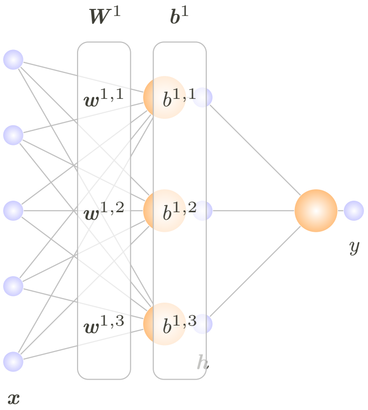
Output Function with one Hidden Layer

$$h = \left(f(w^{\mathbf{1,1}}x + b^{1,1}), f(w^{\mathbf{1,2}}x + b^{1,2}), f(w^{\mathbf{1,3}}x + b^{1,3})\right)$$



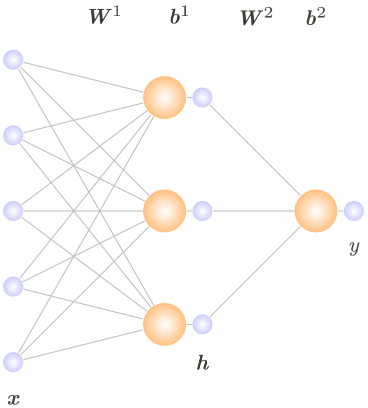
Output Function with one Hidden Layer

$$
\begin{aligned}
\boldsymbol{h} &= \left(f(\boldsymbol{w}^{\mathbf{1,1}}\boldsymbol{x} + b^{1,1}), f(\boldsymbol{w}^{\mathbf{1,2}}\boldsymbol{x} + b^{1,2}), f(\boldsymbol{w}^{\mathbf{1,3}}\boldsymbol{x} + b^{1,3})\right) \\
&= f\left((\boldsymbol{w}^{\mathbf{1,1}}\boldsymbol{x} + b^{1,1}, \boldsymbol{w}^{\mathbf{1,2}}\boldsymbol{x} + b^{1,2}, \boldsymbol{w}^{\mathbf{1,3}}\boldsymbol{x} + b^{1,3})\right)
\end{aligned}
$$



Output Function with one Hidden Layer

$$\begin{aligned}\boldsymbol{h} &= \left(f(\boldsymbol{w^{1,1}x} + b^{1,1}), f(\boldsymbol{w^{1,2}x} + b^{1,2}), f(\boldsymbol{w^{1,3}x} + b^{1,3})\right) \\ &= f\left((\boldsymbol{w^{1,1}x} + b^{1,1}, \boldsymbol{w^{1,2}x} + b^{1,2}, \boldsymbol{w^{1,3}x} + b^{1,3})\right) \\ &= f(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1)\end{aligned}$$

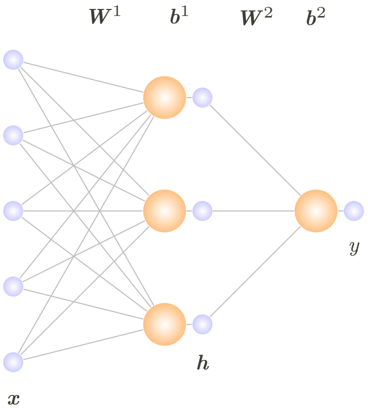


Output Function with one Hidden Layer

$$\begin{aligned}\boldsymbol{h} &= \left(f(\boldsymbol{w}^{\mathbf{1,1}}\boldsymbol{x} + b^{1,1}), f(\boldsymbol{w}^{\mathbf{1,2}}\boldsymbol{x} + b^{1,2}), f(\boldsymbol{w}^{\mathbf{1,3}}\boldsymbol{x} + b^{1,3})\right) \\ &= f\left((\boldsymbol{w}^{\mathbf{1,1}}\boldsymbol{x} + b^{1,1}, \boldsymbol{w}^{\mathbf{1,2}}\boldsymbol{x} + b^{1,2}, \boldsymbol{w}^{\mathbf{1,3}}\boldsymbol{x} + b^{1,3})\right) \\ &= f(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1)\end{aligned}$$

$$y = f(\boldsymbol{W}^2\boldsymbol{h} + \boldsymbol{b}^2) = f(\boldsymbol{W}^2 f(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1) + \boldsymbol{b}^2)$$

Neural Networks



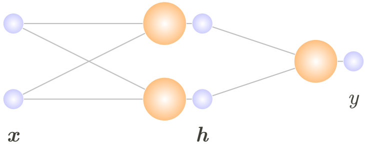
Output Function with one Hidden Layer

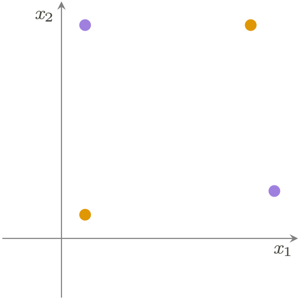
$$\begin{aligned}\boldsymbol{h} &= \left(f(\boldsymbol{w}^{\mathbf{1,1}}\boldsymbol{x} + b^{1,1}), f(\boldsymbol{w}^{\mathbf{1,2}}\boldsymbol{x} + b^{1,2}), f(\boldsymbol{w}^{\mathbf{1,3}}\boldsymbol{x} + b^{1,3})\right) \\ &= f\left((\boldsymbol{w}^{\mathbf{1,1}}\boldsymbol{x} + b^{1,1}, \boldsymbol{w}^{\mathbf{1,2}}\boldsymbol{x} + b^{1,2}, \boldsymbol{w}^{\mathbf{1,3}}\boldsymbol{x} + b^{1,3})\right) \\ &= f(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1)\end{aligned}$$

$$y = f(\boldsymbol{W}^2\boldsymbol{h} + \boldsymbol{b}^2) = f(\boldsymbol{W}^2 f(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1) + \boldsymbol{b}^2)$$

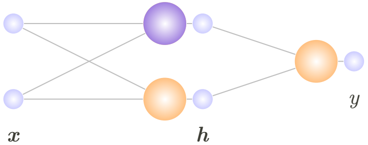
- just matrix multiplications and element-wise non-linearities

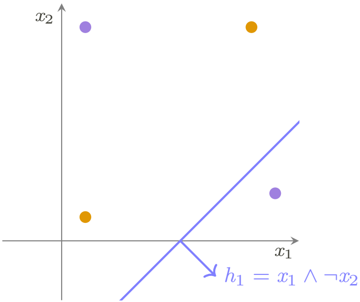
XOR Revisited



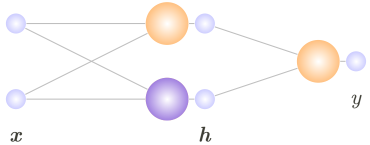


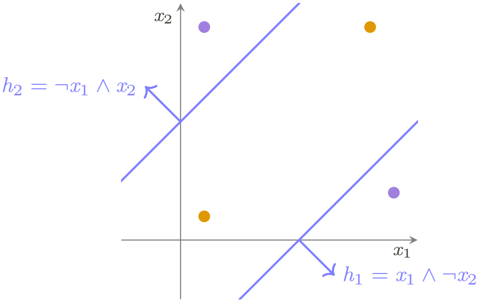
XOR Revisited



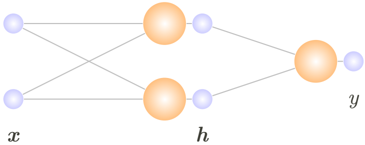


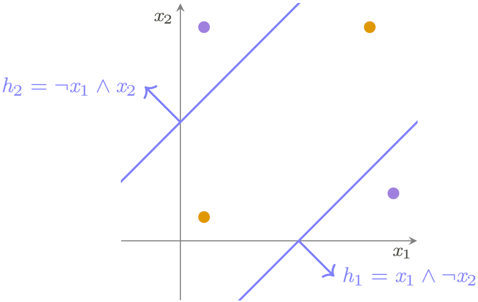
XOR Revisited

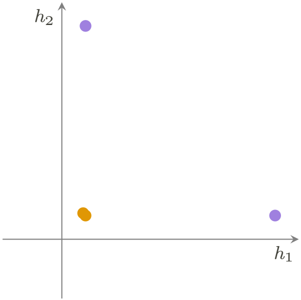




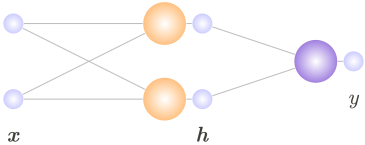
XOR Revisited

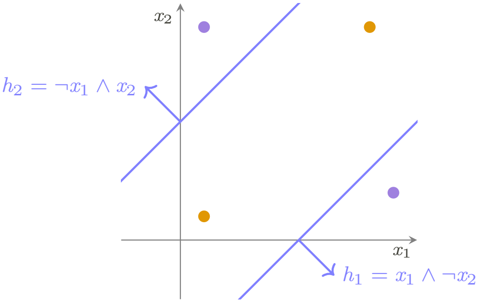


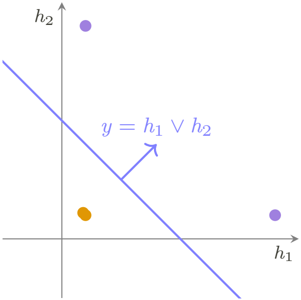




XOR Revisited







How many layers do we need?

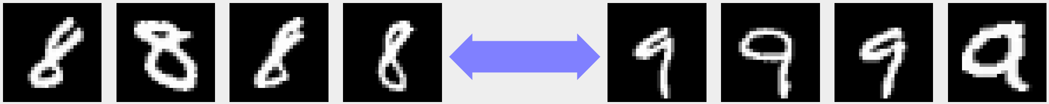
Theoretically...

...one hidden layer is sufficient to model any function:

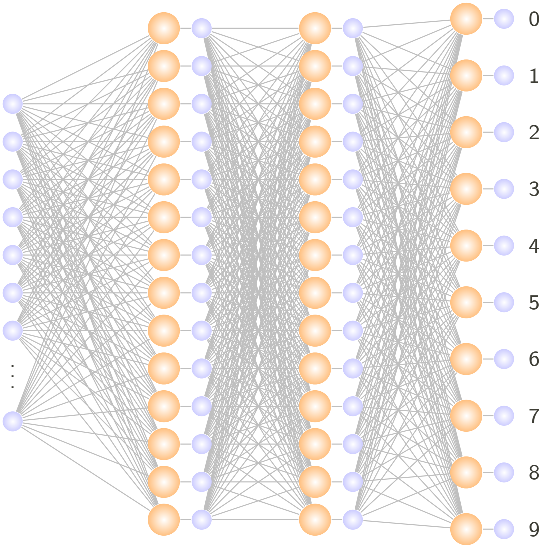
- every boolean function can be transformed into disjunctive normal form [2]
- every continuous function can be approximated with one hidden layer [3]

Practically...

...more than one layer will be more efficient:

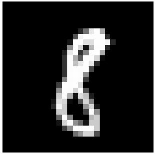


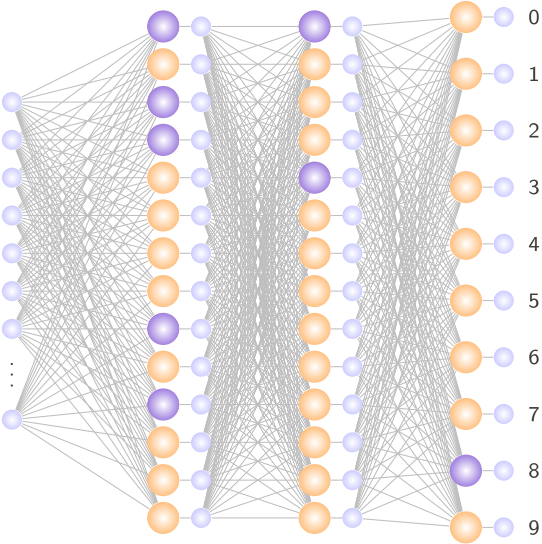
Example: Digit Recognition



[4]

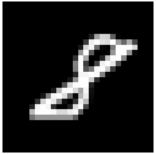
Example: Digit Recognition

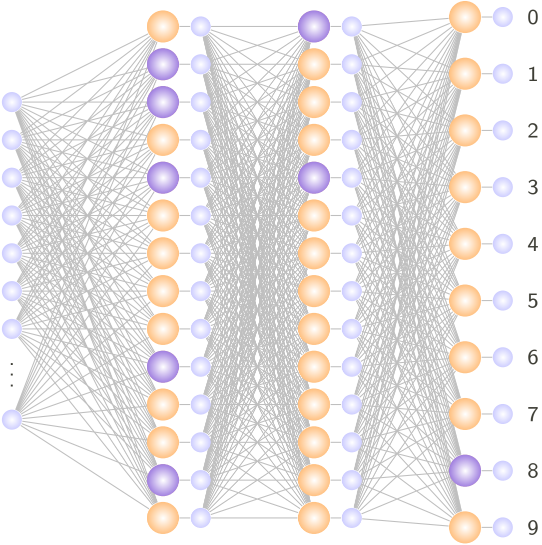




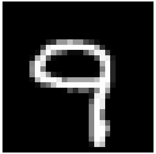
[4]

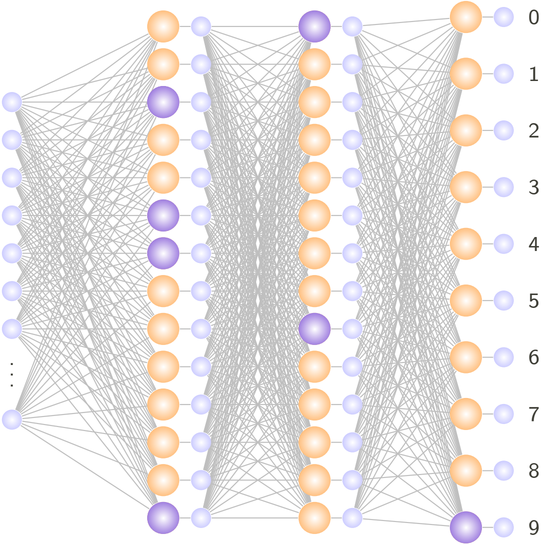
Example: Digit Recognition





[4]





[4]

Supervised Learning

Model

- function $M_{\boldsymbol{\theta}} : \mathbb{R}^n \mapsto \mathbb{R}^m$, $\hat{\boldsymbol{y}} = M_{\boldsymbol{\theta}}(x)$
- parameterized by $\boldsymbol{\theta}$

Data

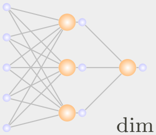
- $T = \left\{\boldsymbol{x}^i, \boldsymbol{y}^i\right\}_{i=1,\dots,k}$
- input $\boldsymbol{x}$
- desired output $\boldsymbol{y}$

Objective

- $l : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$, e.g., $l(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{2}|\hat{\boldsymbol{y}} - \boldsymbol{y}|^2$
- $L(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} l(\hat{\boldsymbol{y}}^i, \boldsymbol{y}^i)$

Model

- function $M_{\boldsymbol{\theta}} : \mathbb{R}^n \mapsto \mathbb{R}^m$, $\hat{\boldsymbol{y}} = M_{\boldsymbol{\theta}}(x)$
- parameterized by $\boldsymbol{\theta}$



$\dim(\boldsymbol{\theta}) \gg 1$

Data

- $T = \{\boldsymbol{x}^i, \boldsymbol{y}^i\}_{i=1,\dots,k}$
- input $\boldsymbol{x}$
- desired output $\boldsymbol{y}$

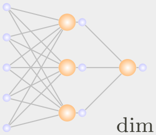
Objective

- $l : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$, e.g., $l(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{2}|\hat{\boldsymbol{y}} - \boldsymbol{y}|^2$
- $L(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} l(\hat{\boldsymbol{y}}^i, \boldsymbol{y}^i)$

Supervised Learning

Model

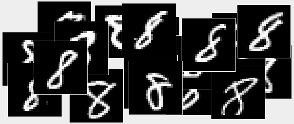
- function $M_{\boldsymbol{\theta}} : \mathbb{R}^n \mapsto \mathbb{R}^m$, $\hat{\boldsymbol{y}} = M_{\boldsymbol{\theta}}(x)$
- parameterized by $\boldsymbol{\theta}$



$\dim(\boldsymbol{\theta}) \gg 1$

Data

- $T = \{\boldsymbol{x}^i, \boldsymbol{y}^i\}_{i=1,\dots,k}$
- input $\boldsymbol{x}$
- desired output $\boldsymbol{y}$



lots of it

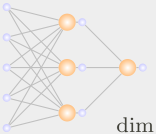
Objective

- $l : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$, e.g., $l(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{2}|\hat{\boldsymbol{y}} - \boldsymbol{y}|^2$
- $L(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} l(\hat{\boldsymbol{y}}^i, \boldsymbol{y}^i)$

Supervised Learning

Model

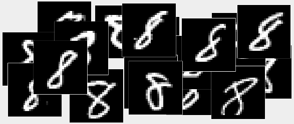
- function $M_{\boldsymbol{\theta}} : \mathbb{R}^n \mapsto \mathbb{R}^m$, $\hat{\boldsymbol{y}} = M_{\boldsymbol{\theta}}(x)$
- parameterized by $\boldsymbol{\theta}$



$\dim(\boldsymbol{\theta}) \gg 1$

Data

- $T = \{\boldsymbol{x}^i, \boldsymbol{y}^i\}_{i=1,\ldots,k}$
- input $\boldsymbol{x}$
- desired output $\boldsymbol{y}$



lots of it

Objective

- $l : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$, e.g., $l(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{2}|\hat{\boldsymbol{y}} - \boldsymbol{y}|^2$
- $L(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} l(\hat{\boldsymbol{y}}^i, \boldsymbol{y}^i)$

smooth, differentiable

Learning by Gradient Descent

Optimization Problem

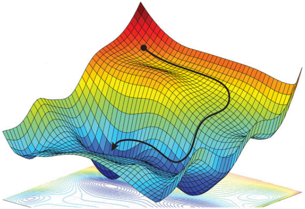
Learning objective:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; L(\theta)$$

- in general no closed-from solution
- probably not even wanted

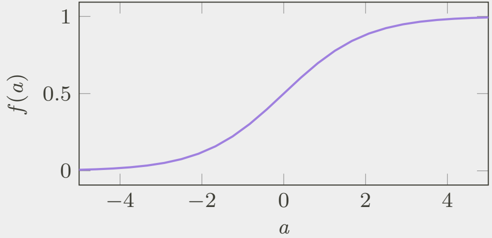
$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

- fallback to gradient descent

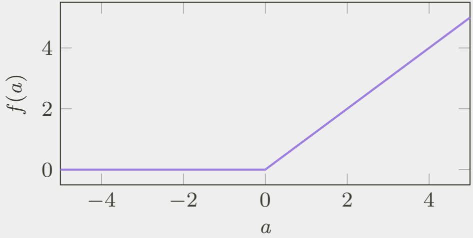


Differentiable Non-linearities

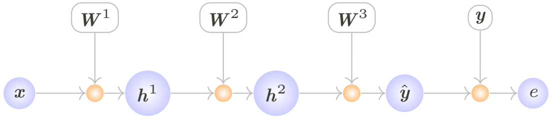
Sigmoid activation function:



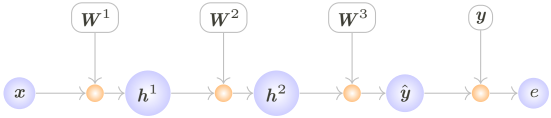
Rectified linear unit (ReLU) activation function:



Error Back Propagation



Error Back Propagation

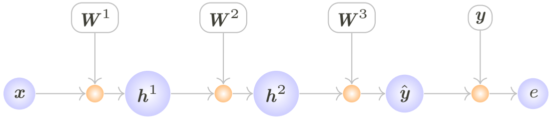


$$\begin{aligned}\hat{\boldsymbol{y}} &= M_{\boldsymbol{\theta}}(\boldsymbol{x}) \;\; (\text{with } \boldsymbol{\theta} = \boldsymbol{W}^1, \boldsymbol{W}^2, \boldsymbol{W}^3) \\ &= f(\boldsymbol{W}^3 \boldsymbol{h}^2) \\ &= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 \boldsymbol{h}^1)) \\ &= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 f(\boldsymbol{W}^1 \boldsymbol{x})))\end{aligned}$$

$$e = L(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} l(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

How to compute $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{\theta}}$?

Error Back Propagation



$$
\begin{aligned}
\hat{\boldsymbol{y}} &= M_{\boldsymbol{\theta}}(\boldsymbol{x}) \quad (\text{with } \boldsymbol{\theta} = \boldsymbol{W}^1, \boldsymbol{W}^2, \boldsymbol{W}^3) \\
&= f(\boldsymbol{W}^3 \boldsymbol{h}^2) \\
&= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 \boldsymbol{h}^1)) \\
&= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 f(\boldsymbol{W}^1 \boldsymbol{x})))
\end{aligned}
$$

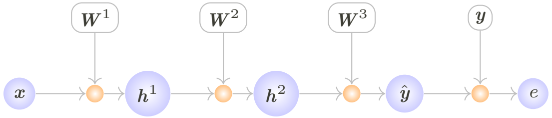
$$e = L(\boldsymbol{\theta}) = \frac{1}{k} \sum_{i=1}^{k} l(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

How to compute $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{\theta}}$?

Chain Rule

$$\frac{\mathrm{d}f(g(x))}{\mathrm{d}x} = \frac{\mathrm{d}f(g(x))}{\mathrm{d}g(x)} \frac{\mathrm{d}g(x)}{\mathrm{d}x}$$

Error Back Propagation



$$\begin{aligned}\hat{\boldsymbol{y}} &= M_{\boldsymbol{\theta}}(\boldsymbol{x}) \ \ (\text{with } \boldsymbol{\theta} = \boldsymbol{W}^1, \boldsymbol{W}^2, \boldsymbol{W}^3) \\ &= f(\boldsymbol{W}^3\boldsymbol{h}^2) \\ &= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2\boldsymbol{h}^1)) \\ &= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 f(\boldsymbol{W}^1\boldsymbol{x})))\end{aligned}$$

$$e = L(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} l(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

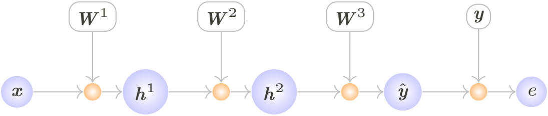
How to compute $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{\theta}}$?

Chain Rule

$$\frac{\mathrm{d}f(g(x))}{\mathrm{d}x} = \frac{\mathrm{d}f(g(x))}{\mathrm{d}g(x)}\frac{\mathrm{d}g(x)}{\mathrm{d}x}$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}}\frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{W}^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}} f'(\boldsymbol{W}^3\boldsymbol{h}^2)\boldsymbol{h}^2$$

Error Back Propagation



$$\hat{y} = M_{\theta}(x) \ \ (\text{with } \theta = W^1, W^2, W^3)$$
$$= f(W^3 h^2)$$
$$= f(W^3 f(W^2 h^1))$$
$$= f(W^3 f(W^2 f(W^1 x)))$$

$$e = L(\theta) = \frac{1}{k} \sum_{i=1}^{k} l(\hat{y}, y)$$

How to compute $\nabla_{\theta} L(\theta) = \frac{\mathrm{d}e}{\mathrm{d}\theta}$?

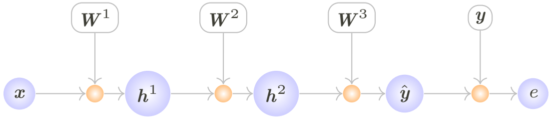
Chain Rule

$$\frac{\mathrm{d}f(g(x))}{\mathrm{d}x} = \frac{\mathrm{d}f(g(x))}{\mathrm{d}g(x)} \frac{\mathrm{d}g(x)}{\mathrm{d}x}$$

$$\frac{\mathrm{d}e}{\mathrm{d}W^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{y}} \frac{\mathrm{d}\hat{y}}{\mathrm{d}W^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{y}} f'(W^3 h^2) h^2$$

$$\frac{\mathrm{d}e}{\mathrm{d}W^2} = \frac{\mathrm{d}e}{\mathrm{d}\hat{y}} \frac{\mathrm{d}\hat{y}}{\mathrm{d}h^2} \frac{\mathrm{d}h^2}{\mathrm{d}W^2}$$

Error Back Propagation



$$
\begin{aligned}
\hat{\boldsymbol{y}} &= M_{\boldsymbol{\theta}}(\boldsymbol{x}) \quad \text{(with } \boldsymbol{\theta} = \boldsymbol{W}^1, \boldsymbol{W}^2, \boldsymbol{W}^3\text{)} \\
&= f(\boldsymbol{W}^3 \boldsymbol{h}^2) \\
&= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 \boldsymbol{h}^1)) \\
&= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 f(\boldsymbol{W}^1 \boldsymbol{x})))
\end{aligned}
$$

$$e = L(\boldsymbol{\theta}) = \frac{1}{k} \sum_{i=1}^{k} l(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

How to compute $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{\theta}}$?

Chain Rule

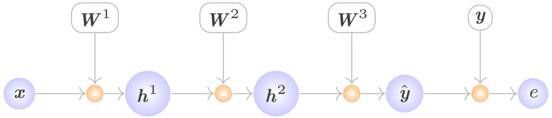
$$\frac{\mathrm{d}f(g(x))}{\mathrm{d}x} = \frac{\mathrm{d}f(g(x))}{\mathrm{d}g(x)} \frac{\mathrm{d}g(x)}{\mathrm{d}x}$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}} \frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{W}^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}} f'(\boldsymbol{W}^3 \boldsymbol{h}^2) \boldsymbol{h}^2$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^2} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}} \frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{h}^2} \frac{\mathrm{d}\boldsymbol{h}^2}{\mathrm{d}\boldsymbol{W}^2}$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^1} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}} \frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{h}^2} \frac{\mathrm{d}\boldsymbol{h}^2}{\mathrm{d}\boldsymbol{h}^1} \frac{\mathrm{d}\boldsymbol{h}^1}{\mathrm{d}\boldsymbol{W}^1}$$

Error Back Propagation



$$\hat{\boldsymbol{y}} = M_{\boldsymbol{\theta}}(\boldsymbol{x}) \quad \text{(with } \boldsymbol{\theta} = \boldsymbol{W}^1, \boldsymbol{W}^2, \boldsymbol{W}^3\text{)}$$
$$= f(\boldsymbol{W}^3 \boldsymbol{h}^2)$$
$$= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 \boldsymbol{h}^1))$$
$$= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 f(\boldsymbol{W}^1 \boldsymbol{x})))$$

$$e = L(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} l(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

How to compute $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{\theta}}$?

Chain Rule

$$\frac{\mathrm{d}f(g(x))}{\mathrm{d}x} = \frac{\mathrm{d}f(g(x))}{\mathrm{d}g(x)}\frac{\mathrm{d}g(x)}{\mathrm{d}x}$$

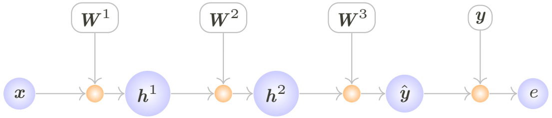
$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}}\frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{W}^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}} f'(\boldsymbol{W}^3\boldsymbol{h}^2)\boldsymbol{h}^2$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^2} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}}\frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{h}^2}\frac{\mathrm{d}\boldsymbol{h}^2}{\mathrm{d}\boldsymbol{W}^2}$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^1} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}}\frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{h}^2}\frac{\mathrm{d}\boldsymbol{h}^2}{\mathrm{d}\boldsymbol{h}^1}\frac{\mathrm{d}\boldsymbol{h}^1}{\mathrm{d}\boldsymbol{W}^1}$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^i} = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{h}^i} f'(\boldsymbol{W}^i\boldsymbol{h}^{i-1})\boldsymbol{h}^{i-1}$$

Error Back Propagation



$$\begin{aligned}\hat{\boldsymbol{y}} &= M_{\boldsymbol{\theta}}(\boldsymbol{x}) \quad (\text{with } \boldsymbol{\theta} = \boldsymbol{W}^1, \boldsymbol{W}^2, \boldsymbol{W}^3) \\ &= f(\boldsymbol{W}^3 \boldsymbol{h}^2) \\ &= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 \boldsymbol{h}^1)) \\ &= f(\boldsymbol{W}^3 f(\boldsymbol{W}^2 f(\boldsymbol{W}^1 \boldsymbol{x})))\end{aligned}$$

$$e = L(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} l(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

How to compute $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{\theta}}$?

Chain Rule

$$\frac{\mathrm{d}f(g(x))}{\mathrm{d}x} = \frac{\mathrm{d}f(g(x))}{\mathrm{d}g(x)}\frac{\mathrm{d}g(x)}{\mathrm{d}x}$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}}\frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{W}^3} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}} f'(\boldsymbol{W}^3 \boldsymbol{h}^2)\boldsymbol{h}^2$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^2} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}}\frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{h}^2}\frac{\mathrm{d}\boldsymbol{h}^2}{\mathrm{d}\boldsymbol{W}^2}$$

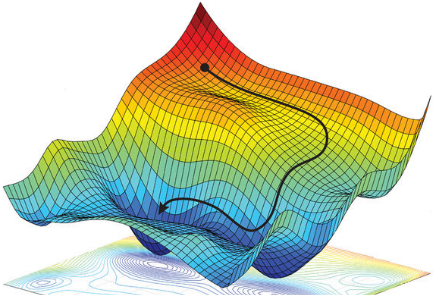
$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^1} = \frac{\mathrm{d}e}{\mathrm{d}\hat{\boldsymbol{y}}}\frac{\mathrm{d}\hat{\boldsymbol{y}}}{\mathrm{d}\boldsymbol{h}^2}\frac{\mathrm{d}\boldsymbol{h}^2}{\mathrm{d}\boldsymbol{h}^1}\frac{\mathrm{d}\boldsymbol{h}^1}{\mathrm{d}\boldsymbol{W}^1}$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{W}^i} = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{h}^i} f'(\boldsymbol{W}^i \boldsymbol{h}^{i-1})\boldsymbol{h}^{i-1}$$

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{h}^i} = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{h}^{i+1}}\frac{\mathrm{d}\boldsymbol{h}^{i+1}}{\mathrm{d}\boldsymbol{h}^i} = \frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{h}^{i+1}}\left(f'(\boldsymbol{W}^{i+1}\boldsymbol{h}^i)\boldsymbol{W}^{i+1}\right)$$

Sampling Strategies

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$



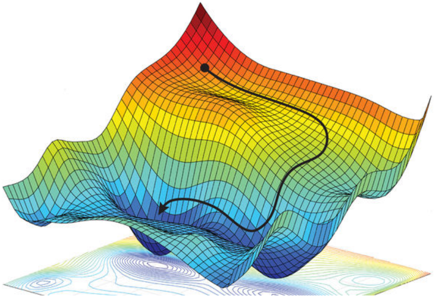
Gradient Descent

$$L(\boldsymbol{\theta}) = \frac{1}{k} \sum_{i=1}^{k} l(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

- impractical for large k

Sampling Strategies

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$



Gradient Descent

$$L(\boldsymbol{\theta}) = \frac{1}{k} \sum_{i=1}^{k} l(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

- impractical for large k

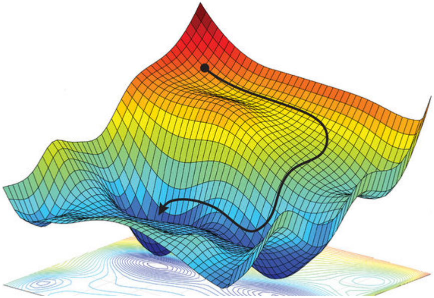
Stochastic Gradient Descent

$$L(\boldsymbol{\theta}) = l(\hat{\boldsymbol{y}}^{i}, \boldsymbol{y}^{i}) \quad \text{with } i \text{ random in } [1, \dots, k]$$

- randomly sampled training example

Sampling Strategies

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$



Gradient Descent

$$L(\boldsymbol{\theta}) = \frac{1}{k} \sum_{i=1}^{k} l(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

- impractical for large k

Stochastic Gradient Descent

$$L(\boldsymbol{\theta}) = l(\hat{\boldsymbol{y}}^{i}, \boldsymbol{y}^{i}) \quad \text{with } i \text{ random in } [1, \ldots, k]$$

- randomly sampled training example

Batch Gradient Descent

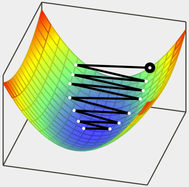
$$L(\boldsymbol{\theta}) = \frac{1}{b} \sum_{i=1}^{b} l(\hat{\boldsymbol{y}}^{p_i}, \boldsymbol{y}^{p_i}) \quad \text{with } p_i \text{ random in } [1, \ldots, k]$$

- randomly sampled set of training examples

Optimization Algorithms

Plain Gradient Descent

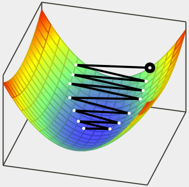
$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^{t})$$



Optimization Algorithms

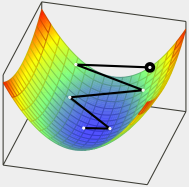
Plain Gradient Descent

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^{t})$$



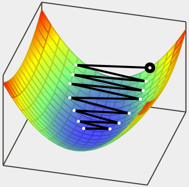
Gradient Descent with Momemtum

$$\boldsymbol{\delta}^{t} = \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^{t}) + \beta \boldsymbol{\delta}^{t-1}$$
$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \boldsymbol{\delta}^{t}$$



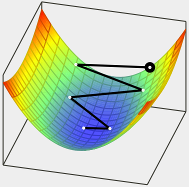
Plain Gradient Descent

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^t)$$



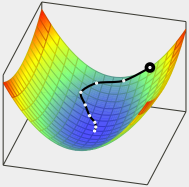
Gradient Descent with Momemtum

$$\boldsymbol{\delta}^t = \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^t) + \beta \boldsymbol{\delta}^{t-1}$$
$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \boldsymbol{\delta}^t$$



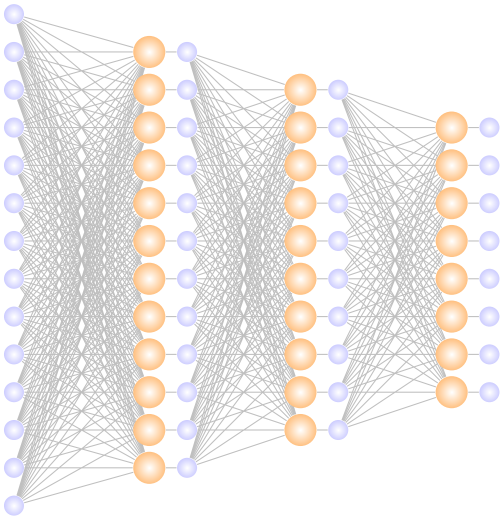
Adaptive Moment Estimation (Adam)

$$\begin{aligned}
\boldsymbol{g}^t &= \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^t) \\
\boldsymbol{m}^t &= \beta_1 \boldsymbol{m}^{t-1} + (1 - \beta_1) \boldsymbol{g}^t \\
\boldsymbol{v}^t &= \beta_2 \boldsymbol{v}^{t-1} + (1 - \beta_2) \left(\boldsymbol{g}^t \odot \boldsymbol{g}^t\right) \\
\boldsymbol{\theta}^{t+1} &= \boldsymbol{\theta}^t - \alpha \frac{\boldsymbol{m}^t}{\sqrt{\boldsymbol{v}^t} + \epsilon}
\end{aligned}$$

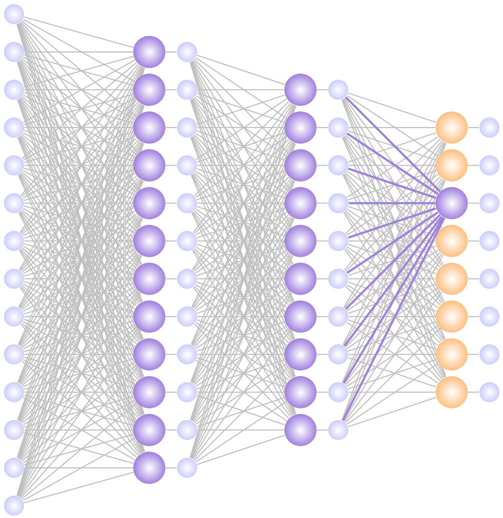


[5]

Receptive Fields

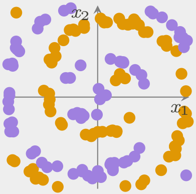


Receptive Fields

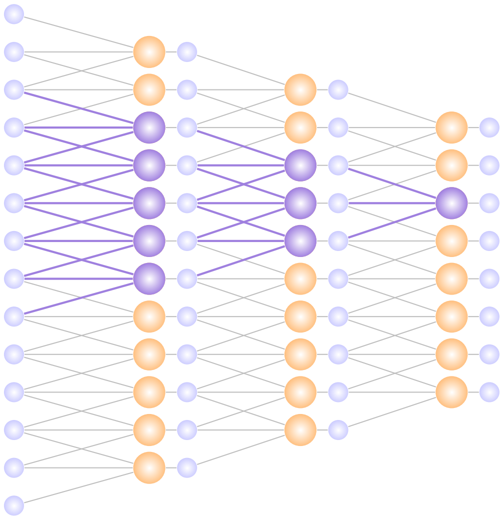


Fully Connected

- if inputs are unstructured



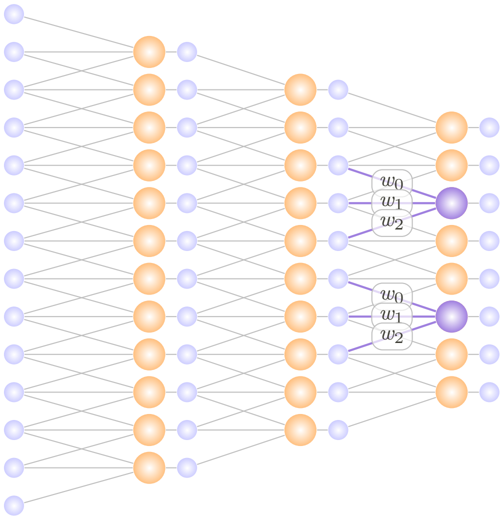
Receptive Fields



Locally Connected and Shared Weights

- if inputs have spatial/temporal structure

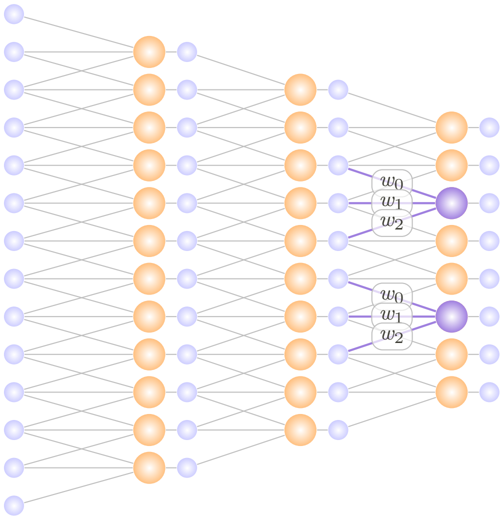
Receptive Fields



Locally Connected and Shared Weights

- if inputs have spatial/temporal structure
- if outputs are equivariant to inputs

Receptive Fields



Locally Connected and Shared Weights

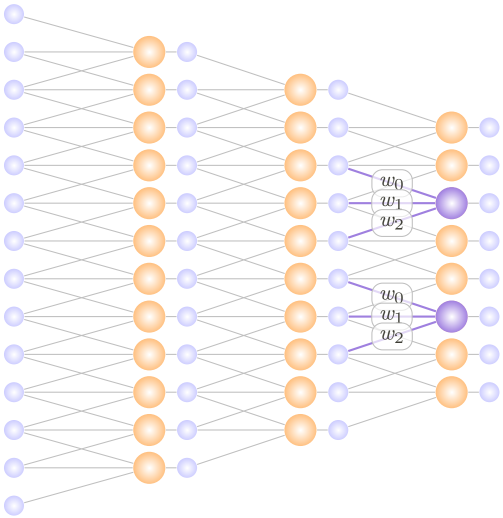
- if inputs have spatial/temporal structure
- if outputs are equivariant to inputs

$$y_i = f(\boldsymbol{W}\boldsymbol{h} + \boldsymbol{b})_i$$

$$= f(\sum_{j=0}^{2} h_{i+j} w_j + b_i)$$

$$\boldsymbol{y} = f(\boldsymbol{h} * \boldsymbol{k} + \boldsymbol{b}) \text{ with } \boldsymbol{k} = (w_j)_{j=0,\dots,2}$$

Receptive Fields



Locally Connected and Shared Weights

- if inputs have spatial/temporal structure
- if outputs are equivariant to inputs

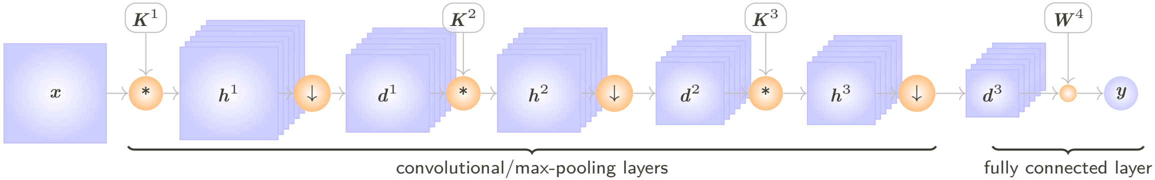
$$y_i = f(\boldsymbol{W}\boldsymbol{h} + \boldsymbol{b})_i$$

$$= f(\sum_{j=0}^{2} h_{i+j} w_j + b_i)$$

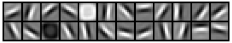
$$\boldsymbol{y} = f(\boldsymbol{h} * \boldsymbol{k} + \boldsymbol{b}) \text{ with } \boldsymbol{k} = (w_j)_{j=0,\dots,2}$$

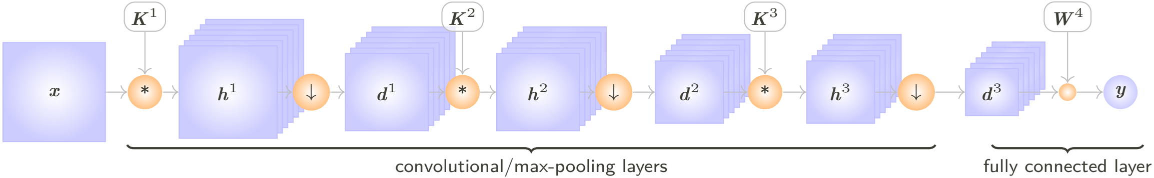
$\rightarrow$ convolutional neural networks (CNNs)

Convolutional Neural Networks

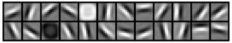


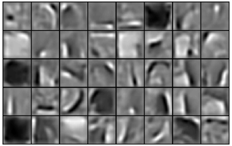
Convolutional Neural Networks

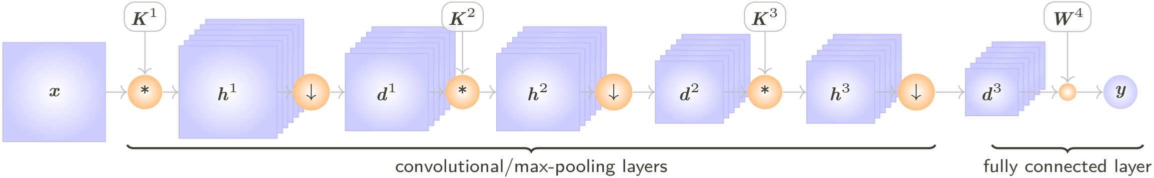




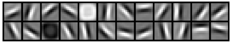
Convolutional Neural Networks

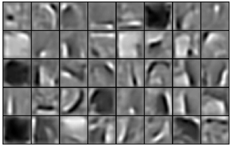


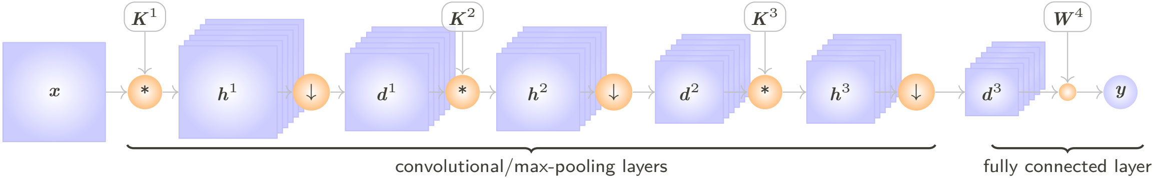




Convolutional Neural Networks

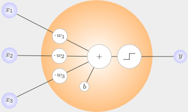




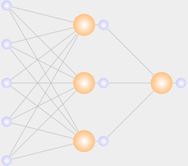


Summary

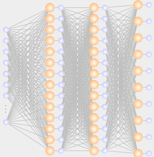
1. Perceptrons



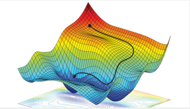
2. Networks



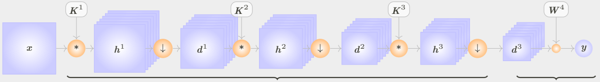
3. Deep Networks



4. Learning



5. Convolutional Networks



References

[1] Pollack, J. B.
No harm intended: A review of the perceptrons expanded edition.
Journal of Mathematical Psychology (1989).

[2] Davey, B. & Priestley, H.
Introduction to Lattices and Order (Cambridge Mathematical Textbooks, 1990).

[3] Cybenko, G.
Approximations by superpositions of sigmoidal functions.
Mathematics of Control, Signals, and Systems (1989).

[4] URL `https://www.youtube.com/watch?v=aircAruvnKk`.

[5] Kingma, D. P. & Ba, J.
Adam: A method for stochastic optimization.
CoRR **abs/1412.6980** (2014).
URL `http://arxiv.org/abs/1412.6980`.
1412.6980.