Introduction to Neural Networks

Objectives

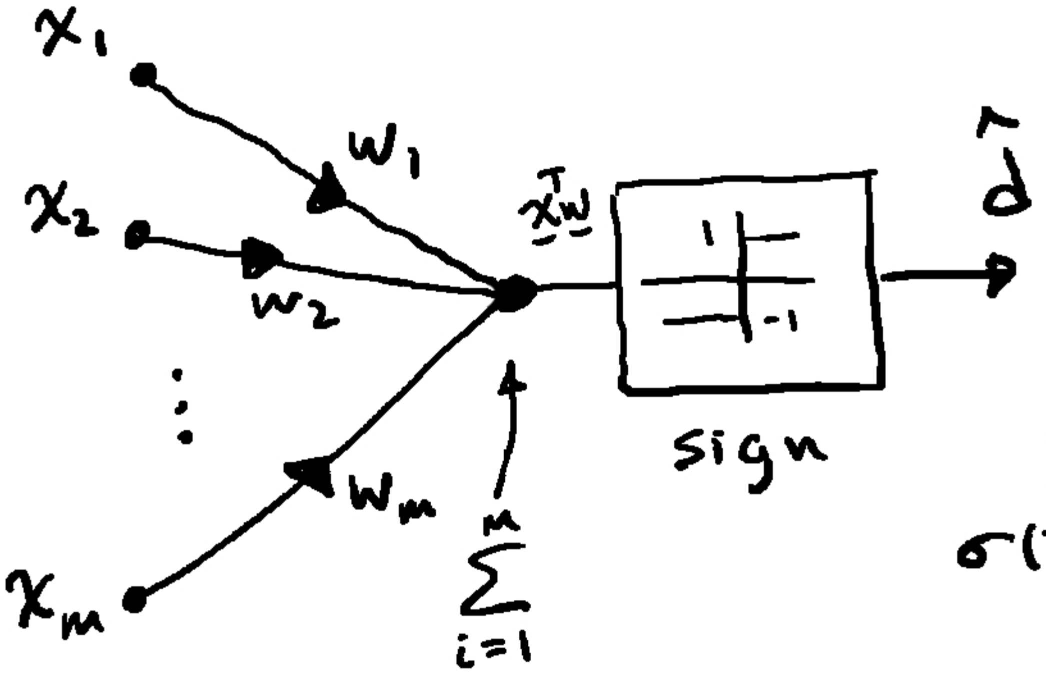
- relate neural networks to linear classifiers
- define structure of multilayer neural network
- overview procedure for training neural networks

The "neuron" generalizes a linear classifier

Feature: $\underline{x}^T = [\, x_1 \;\; x_2 \;\; \cdots \;\; x_m \,]$ Weights: $\underline{w}^T = [\, w_1 \;\; w_2 \cdots \; w_m \,]$

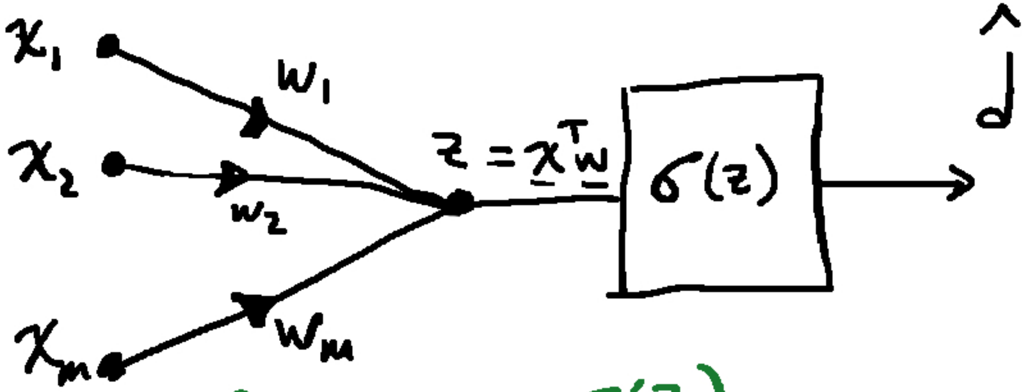
Linear classifier

$$\hat{d} = \operatorname{sign}(\underline{x}^T \underline{w})$$



Neuron

$$\hat{d} = \sigma(\underline{x}^T \underline{w})$$



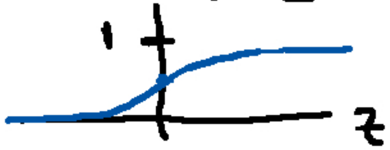
Common $\sigma(z)$

ReLU

$\sigma(z) = \max\{0, z\}$

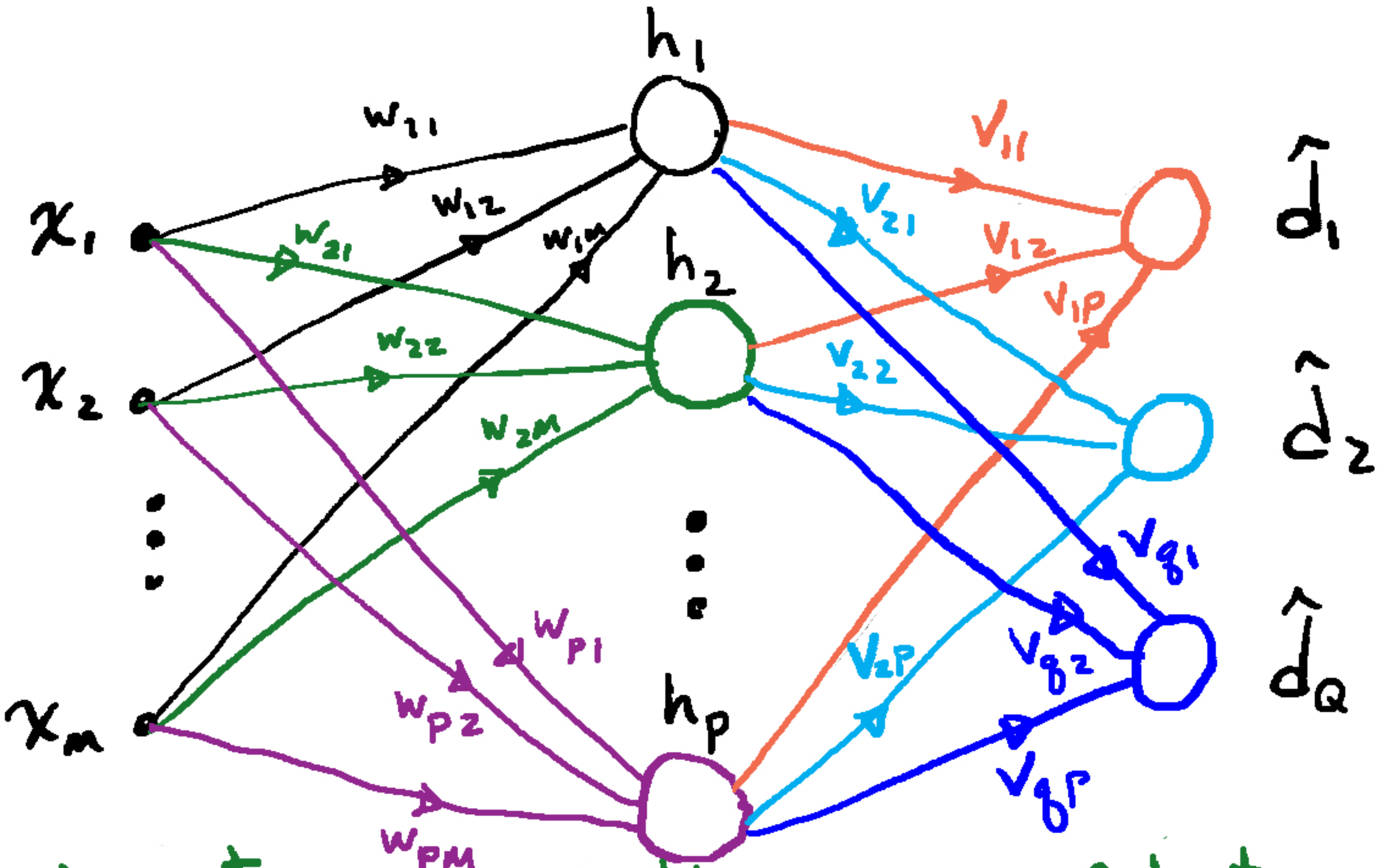
Logistic

$\sigma(z) = \dfrac{1}{1+e^{-z}}$



$\sigma(z) = \operatorname{sign}(z)$

A neural network is a network of neurons



input layer

hidden layer

output layer

shallow ⟶ deep

multiple outputs → solve multiple problems

$$h_\ell = \sigma\left(\sum_{j=1}^{M} w_{\ell j}\, x_j\right)$$

$$\hat{d}_m = \sigma\left(\sum_{n=1}^{P} v_{mn}\, h_n\right)$$

$$= \sigma\left(\sum_{n=1}^{P} v_{m,n}\,\sigma\left(\sum_{j=1}^{M} w_{nj}\, x_j\right)\right)$$

"deep" learning → many hidden layers

Two issues must be addressed to use NNs

1) Network structure: number of layers, number of hidden nodes in each layer

Open question. Universal approximation theorem (1991): Three layer network can approximate any function abitrorily well given enough hidden nodes and the right weights

2) Choosing the weights

Stochastic gradient descent and backpropagation

Nonconvex $\rightarrow$ local minima

Backpropagation updates each layer in sequence

- work back from deep (output) to shallow (input)

$$\min_{w_{\ell j}, v_{\ell j}, \ldots} \sum_{i=1}^{N} \sum_{q=1}^{Q} \frac{1}{2}\left(\hat{d}_{i,q} - d_{i,q}\right)^2$$

N training samples, Q outputs

1) Initial guess on $w_{\ell j}, v_{\ell j}$ (etc)
2) Randomly choose training sample i
3) Calculate $h_{i,p}$, $\hat{d}_{i,q}$
4) Gradient descent update $v_{\ell j}$, then $w_{\ell j}$ (deep to shallow)

Chain rule is key for deriving gradients