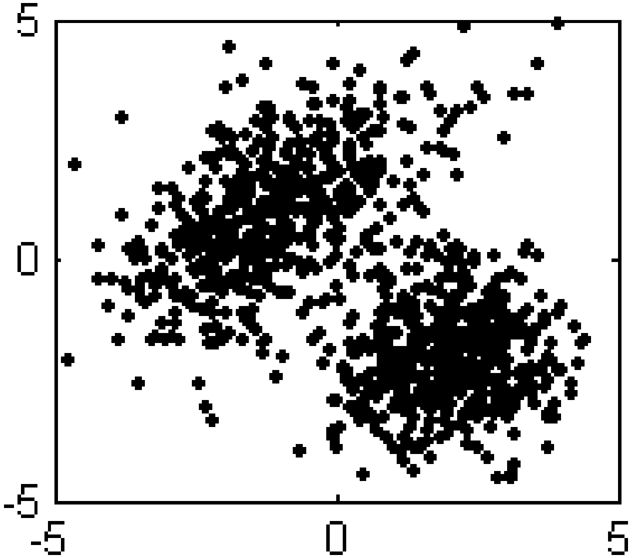
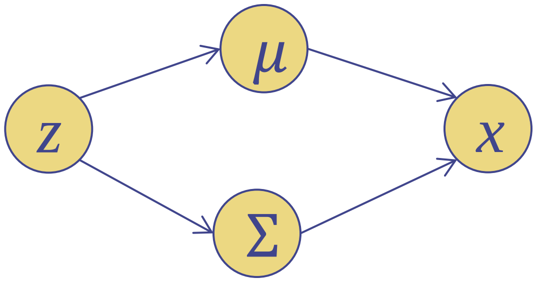
Machine Learning

4771

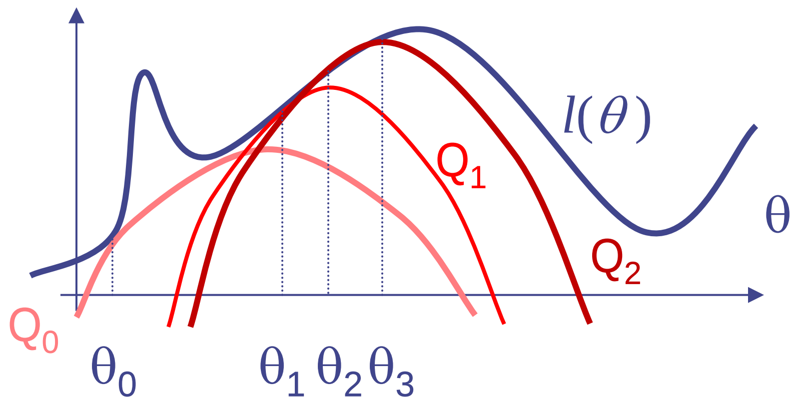
Instructor: Itsik Pe’er

Reminder: EM for Gauss. Mix.





Expectation-Maximization:
Iteratively improve
Expected-log-likelihood



Class 16

- Multi-Layer Neural Networks
- Back-Propagation

Multi-Layer Neural Networks

- Perceptron/linear/logistic/threshold neurons

Multi-Layer Neural Networks

- Perceptron/linear/logistic/threshold

- Different functions of the linear combination of inputs

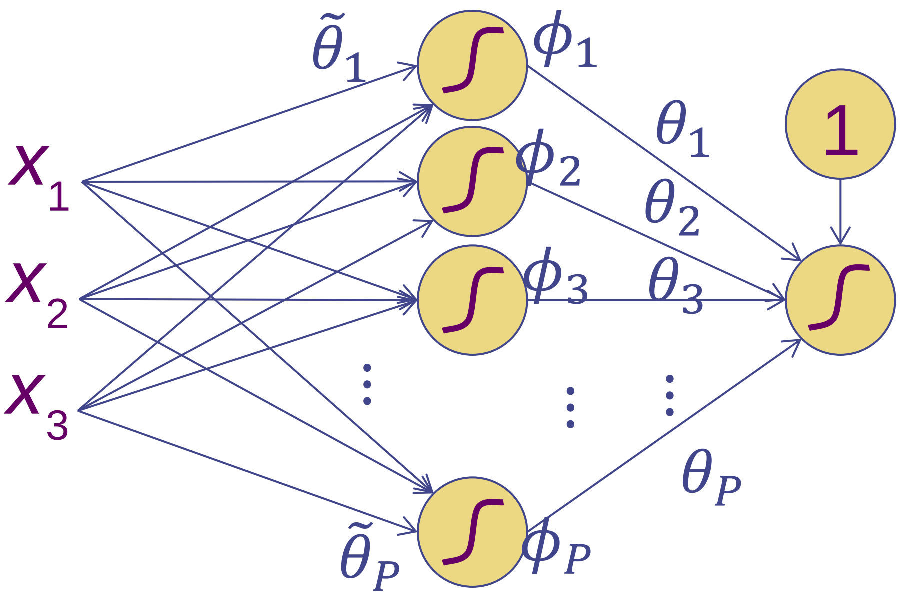
$$f(\boldsymbol{x}) = g(\theta^T \boldsymbol{x})$$

- Different loss functions
- Different strategies for minimizing empirical risk

Multi-Layer Neural Networks

- Need to introduce non-linearities between layers

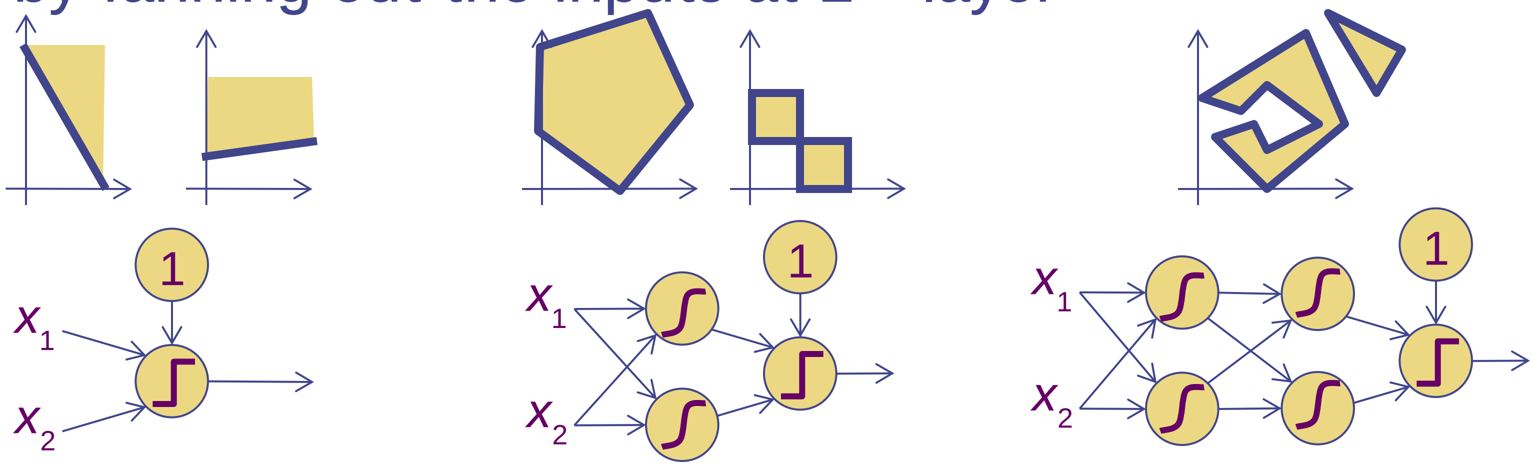
- Neural network can adjust the basis functions themselves...



$$f(\boldsymbol{x}) = g\left(\sum_{i=1}^{P} \theta_i g\left(\tilde{\theta}_i^{\,T} \boldsymbol{x}\right)\right)$$

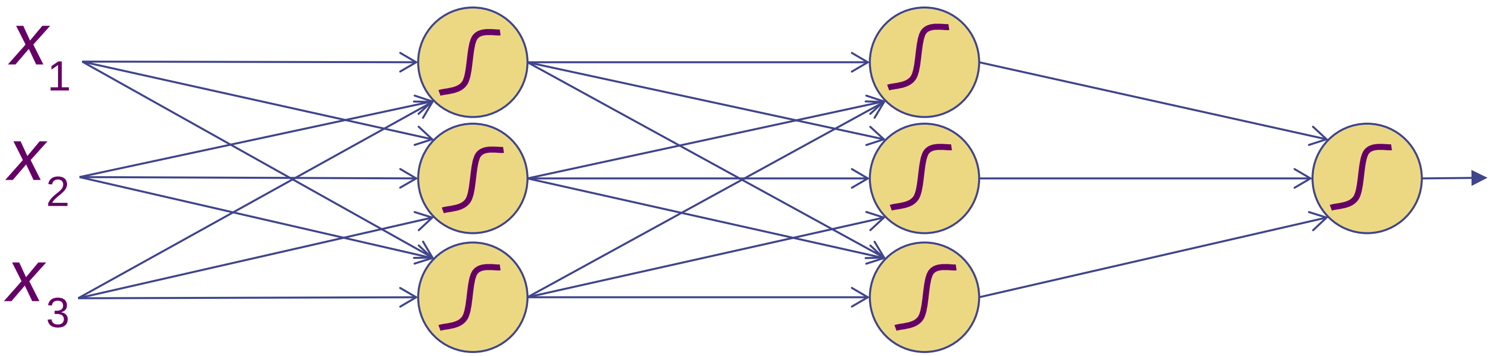
Multi-Layer Neural Networks

- Multi-Layer Network can handle more complex decisions
- 1-layer: is linear, can't handle XOR
- Each layer adds more flexibility (but more parameters!)
- Each node splits its input space with linear hyperplane
- 2-layer: if last layer is AND operation, get convex hull
- 3-layer: can do almost anything multi-layer can
  by fanning out the inputs at 2nd layer



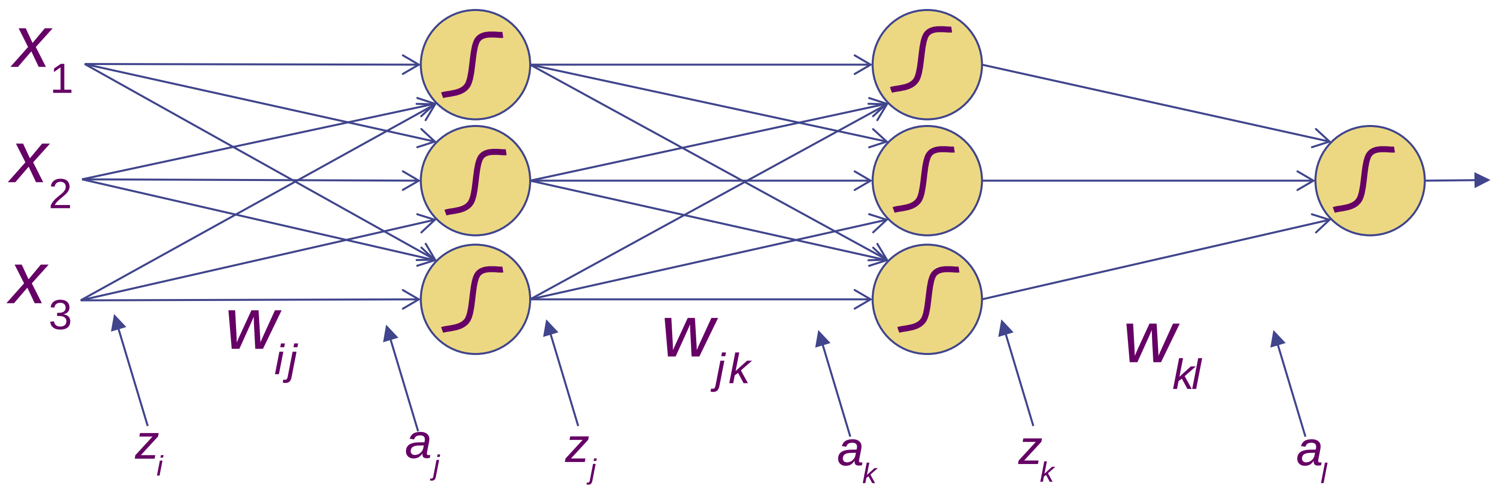
- Note: Without loss of generality, we can omit the 1 and θ_0

Parameterizing Neural Networks



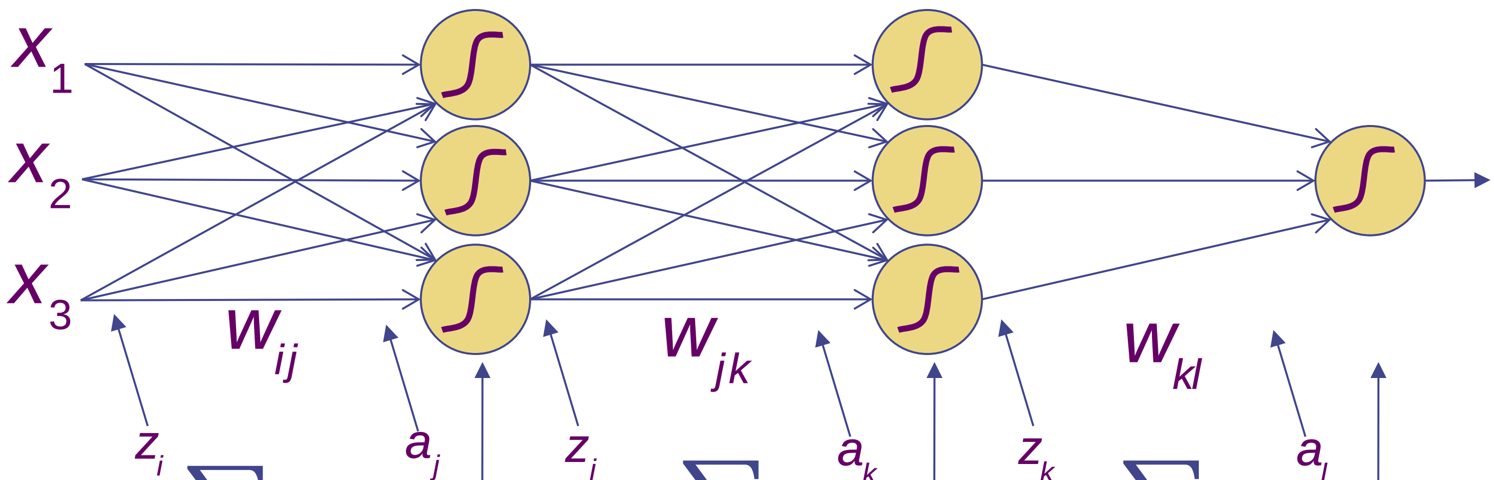
Parameterizing Neural Networks

- Parameters are *weights* $\theta = \{w_{ij}, w_{jk}, w_{kl}\}$
- Weights define linear combinations of inputs...



Parameterizing Neural Networks

- Parameters are *weights* $\theta = \{w_{ij}, w_{jk}, w_{kl}\}$
- Weights define linear combinations of inputs...



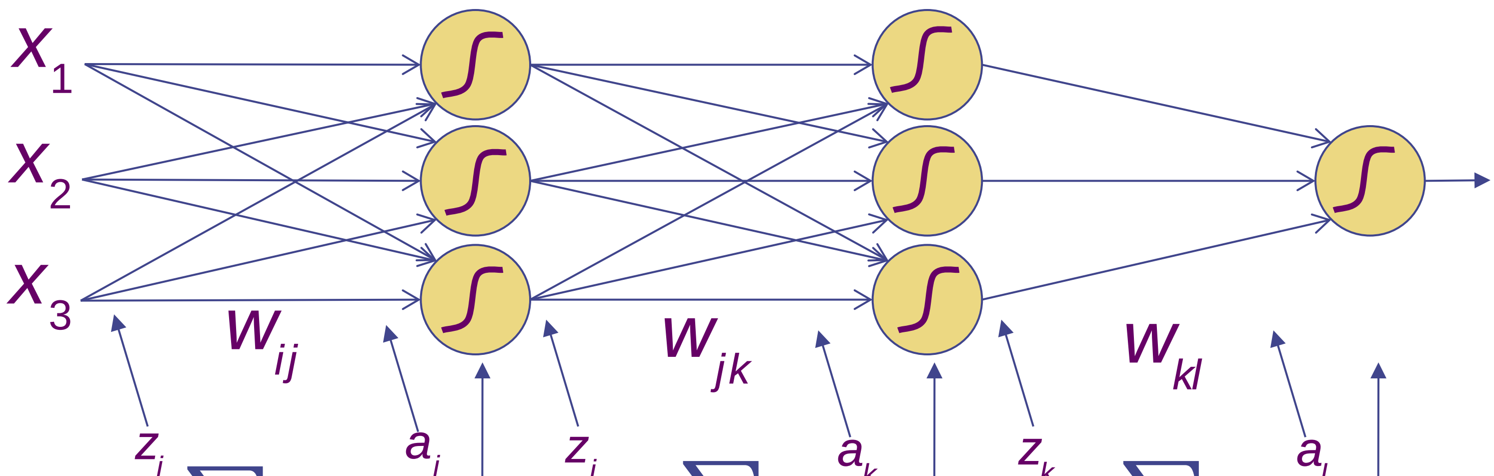
$$a_j = \sum_i w_{ij} z_i \qquad a_k = \sum_j w_{jk} z_j \qquad a_l = \sum_k w_{kl} z_k$$

$$z_j = g(a_j) \qquad z_k = g(a_k) \qquad z_l = g(a_l)$$

...that activate neurons...
...that linearly combine...
...to activate neurons...
...that linearly combine to produce output

Back-Propagation

- Gradient descent on squared loss is done layer by layer



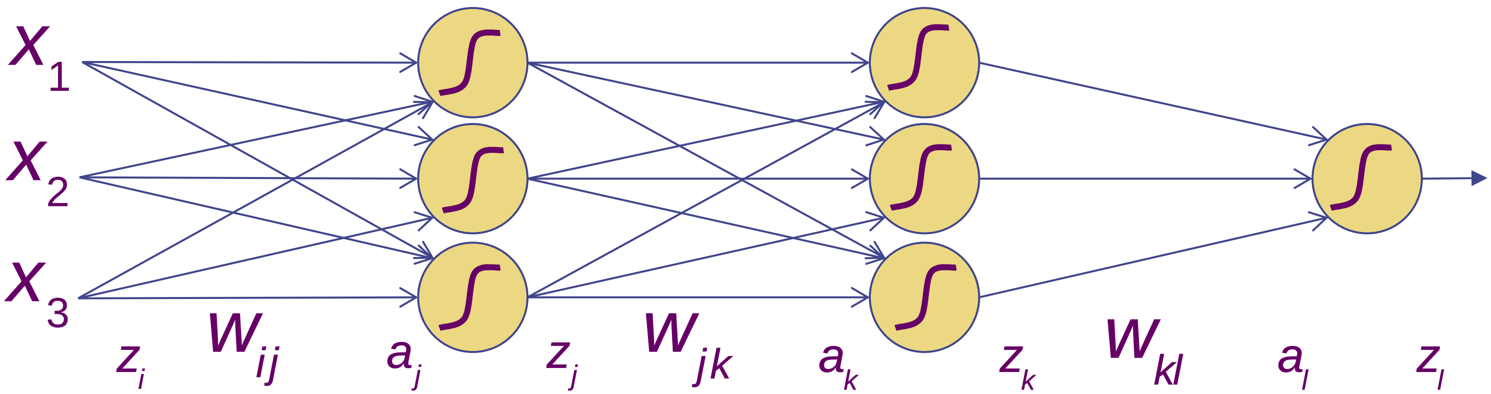
$$a_j = \sum_i w_{ij} z_i \qquad a_k = \sum_j w_{jk} z_j \qquad a_l = \sum_k w_{kl} z_k$$

$$z_j = g(a_j) \qquad z_k = g(a_k) \qquad z_l = g(a_l)$$

- Back-Propagation: Splits layer into its inputs & outputs
- Get gradient on output…back-track chain rule until input

Back-Propagation

- Cost function: $R(\theta) = \frac{1}{N}\sum_{n=1}^{N} L(y^n - f(\boldsymbol{x}^n))$

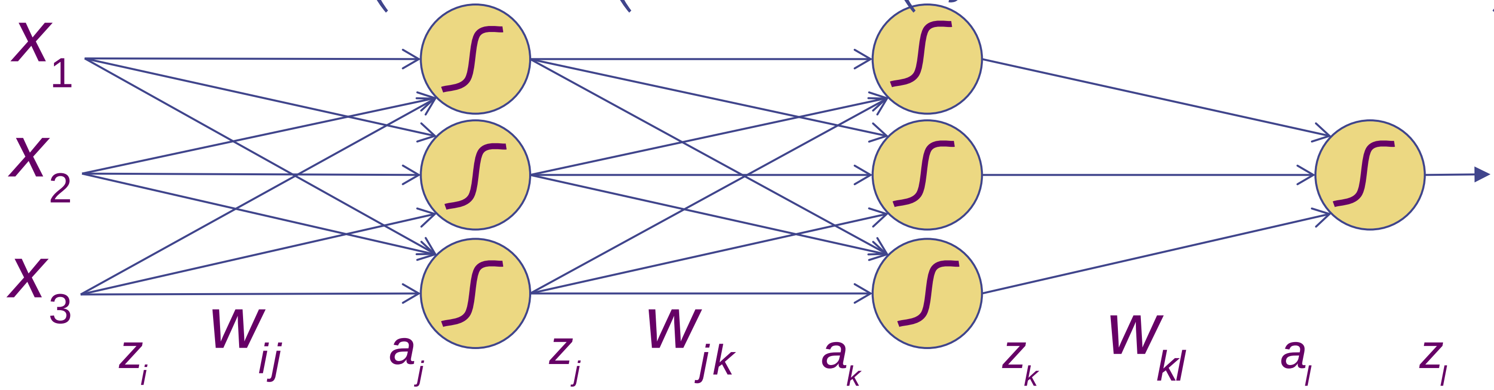


Back-Propagation

- Cost function:

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N} L(y^n - f(\boldsymbol{x}^n))$$

$$= \frac{1}{N}\sum_{n=1}^{N} \tfrac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



- First compute output layer derivative:

$$L^n \stackrel{\text{def}}{=} \tfrac{1}{2}(y^n - f(\boldsymbol{x}^n))^2$$

$$\frac{\partial R}{\partial w_{kl}} =$$

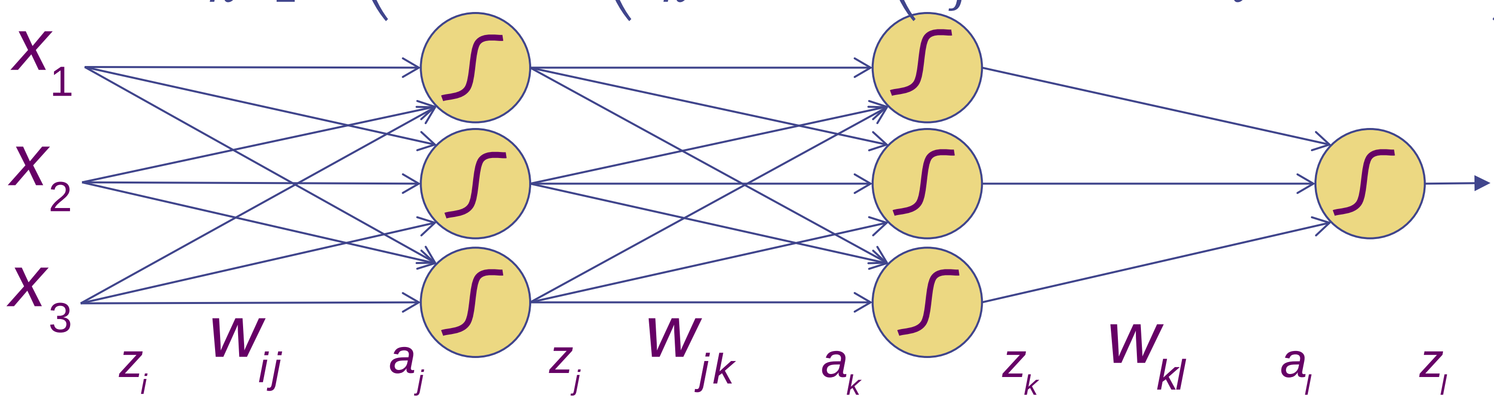
Itsik Pe'er, Columbia University

Back-Propagation

- Cost function:

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N} L(y^n - f(x^n))$$

$$= \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



- First compute output layer derivative:

$$L^n \stackrel{\text{def}}{=} \frac{1}{2}(y^n - f(x^n))^2$$

$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right)$$

Chain Rule

$$= \frac{1}{N}\sum_n \left[\frac{\partial \frac{1}{2}(y^n - g(a_l^n))^2}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right)$$

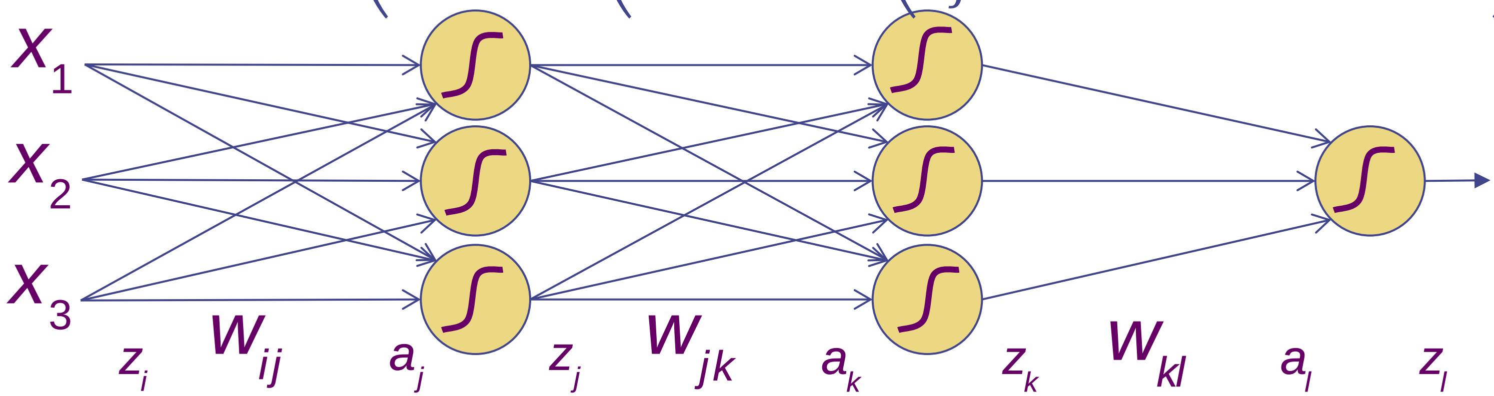
Itsik Pe'er, Columbia University

Back-Propagation

- Cost function:

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N} L(y^n - f(x^n))$$

$$= \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



- First compute output layer derivative:

$$L^n \stackrel{\text{def}}{=} \frac{1}{2}(y^n - f(x^n))^2$$

$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right)$$

Chain Rule

$$= \frac{1}{N}\sum\nolimits_n \left[\frac{\partial \frac{1}{2}(y^n - g(a_l^n))^2}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right) = \frac{1}{N}\sum\nolimits_n [-(y^n - z_l^n)g'(a_l^n)](z_k^n)$$

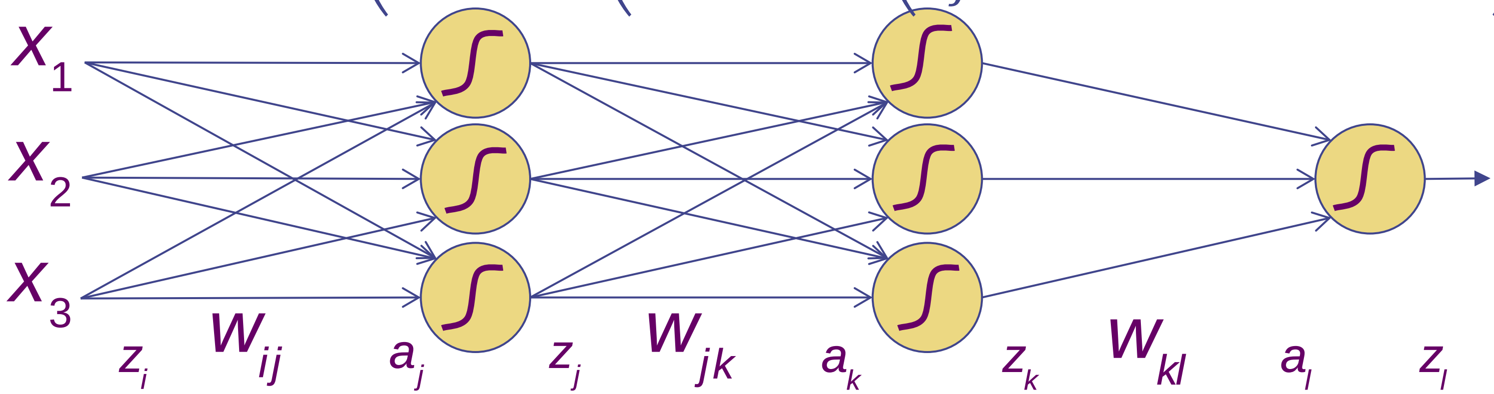
Itsik Pe'er, Columbia University

Back-Propagation

- Cost function:

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N} L(y^n - f(x^n))$$

$$= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



- First compute output layer derivative:

$$L^n \stackrel{\text{def}}{=} \frac{1}{2}(y^n - f(x^n))^2$$

$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right)$$

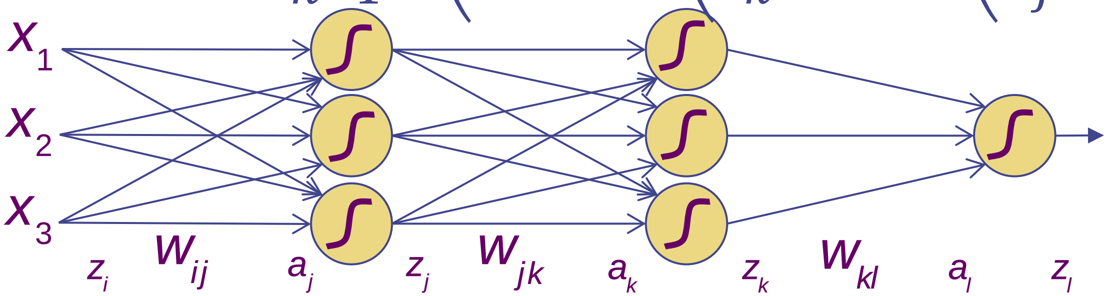
Chain Rule

$$= \frac{1}{N}\sum_n \left[\frac{\partial \frac{1}{2}(y^n - g(a_l^n))^2}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right) = \frac{1}{N}\sum_n \underbrace{[-(y^n - z_l^n)\, g'(a_l^n)]}_{\text{Define as } \delta}(z_k^n) = \frac{\sum_n \delta_l^n z_k^n}{N}$$

Itsik Pe'er, Columbia University

Back-Propagation

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



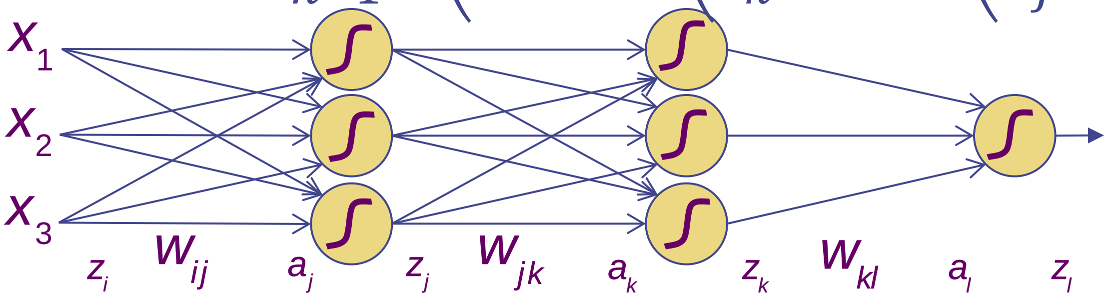
$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right) = \frac{1}{N}\sum_n [-(y^n - z_l^n) g'(a_l^n)] z_k^n = \frac{\sum_n \delta_l^n z_k^n}{N}$$

- Next, hidden layer derivative:

$$\frac{\partial R}{\partial w_{jk}} =$$

Back-Propagation

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N} \tfrac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right) = \frac{1}{N}\sum_n [-(y^n - z_l^n) g'(a_l^n)] z_k^n = \frac{\sum_n \delta_l^n z_k^n}{N}$$

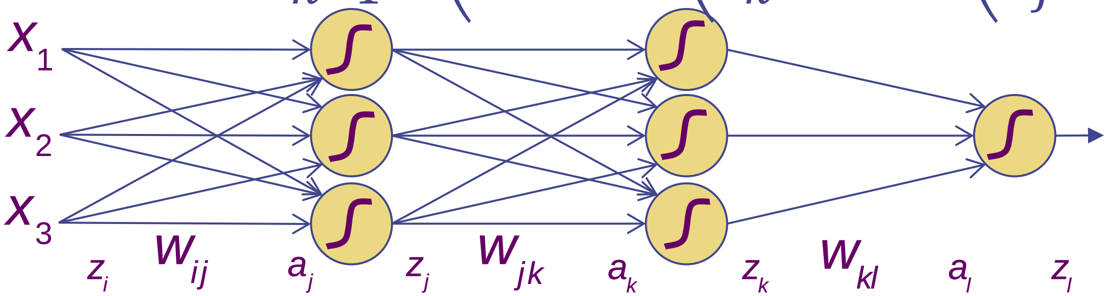
- Next, hidden layer derivative:

$$\frac{\partial R}{\partial w_{jk}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right)$$

Itsik Pe'er, Columbia University

Back-Propagation

$$R(\theta)=\frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\left(y^n-g\left(\sum_k w_{kl}\,g\left(\sum_j w_{jk}\,g\left(\sum_i w_{ij}x_i^n\right)\right)\right)\right)^2$$



$$\frac{\partial R}{\partial w_{kl}}=\frac{1}{N}\sum_n\left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right)=\frac{1}{N}\sum_n[-(y^n-z_l^n)g'(a_l^n)]z_k^n=\frac{\sum_n\delta_l^n z_k^n}{N}$$

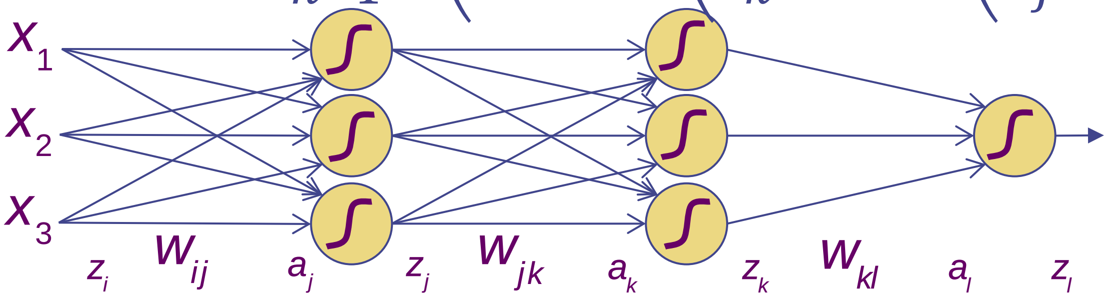
- Next, hidden layer derivative:

$$\frac{\partial R}{\partial w_{jk}}=\frac{1}{N}\sum_n\left[\frac{\partial L^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right)=\frac{1}{N}\sum_n\left[\sum_l\frac{\partial L^n}{\partial a_l^n}\frac{\partial a_l^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right)$$

Multivariate Chain Rule

Back-Propagation

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right) = \frac{1}{N}\sum_n [-(y^n - z_l^n)g'(a_l^n)]z_k^n = \frac{\sum_n \delta_l^n z_k^n}{N}$$

- Next, hidden layer derivative:

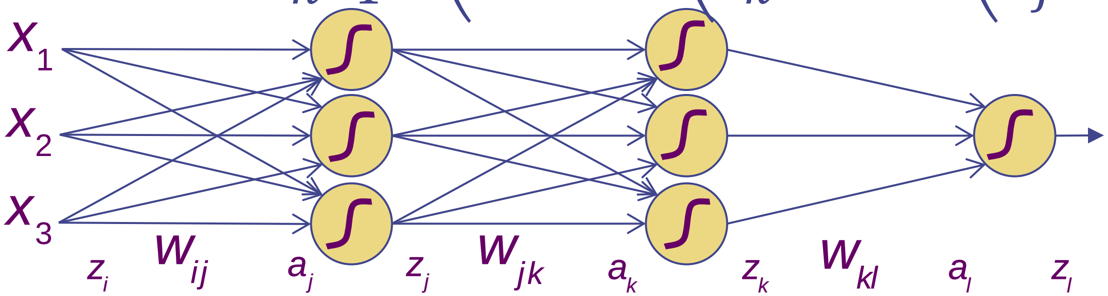
$$\frac{\partial R}{\partial w_{jk}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right) = \frac{1}{N}\sum_n \left[\sum_l \frac{\partial L^n}{\partial a_l^n}\frac{\partial a_l^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right)$$

$$= \frac{1}{N}\sum_n \left[\sum_l \delta_l^n \frac{\partial a_l^n}{\partial a_k^n}\right]\left(z_j^n\right)$$

recall $a_l = \sum_k w_{kl}\, g(a_k)$

Back-Propagation

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N}\tfrac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right) = \frac{1}{N}\sum_n \left[-(y^n - z_l^n)\, g'(a_l^n)\right] z_k^n = \frac{\sum_n \delta_l^n z_k^n}{N}$$

- Next, hidden layer derivative:

$$\frac{\partial R}{\partial w_{jk}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right) = \frac{1}{N}\sum_n \left[\sum_l \frac{\partial L^n}{\partial a_l^n}\frac{\partial a_l^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right)$$

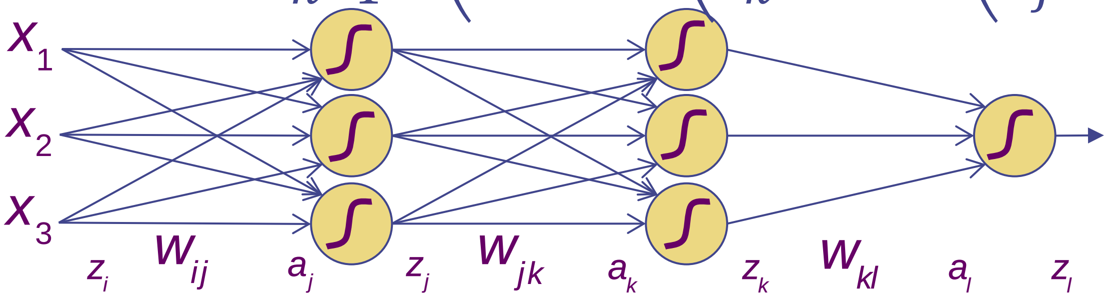
$$= \frac{1}{N}\sum_n \left[\sum_l \delta_l^n \frac{\partial a_l^n}{\partial a_k^n}\right](z_j^n) = \frac{1}{N}\sum_n \left[\sum_l \delta_l^n w_{kl}\, g'(a_k^n)\right] z_j^n$$

recall $a_l = \sum_k w_{kl}\, g(a_k)$

Itsik Pe'er, Columbia University

Back-Propagation

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right) = \frac{1}{N}\sum_n [-(y^n - z_l^n)g'(a_l^n)]z_k^n = \frac{\sum_n \delta_l^n z_k^n}{N}$$

- Next, hidden layer derivative:

$$\frac{\partial R}{\partial w_{jk}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right) = \frac{1}{N}\sum_n \left[\sum_l \frac{\partial L^n}{\partial a_l^n}\frac{\partial a_l^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right)$$

$$= \frac{1}{N}\sum_n \left[\sum_l \delta_l^n \frac{\partial a_l^n}{\partial a_k^n}\right](z_j^n) = \frac{1}{N}\sum_n \left[\sum_l \delta_l^n w_{kl}\, g'(a_k^n)\right] z_j^n = \frac{1}{N}\sum_n \delta_k^n z_j^n$$

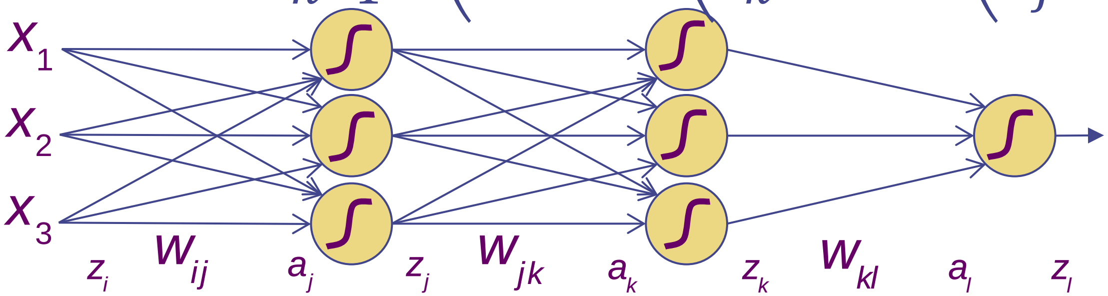
recall $a_l = \sum_k w_{kl}\, g(a_k)$

Define as δ

Itsik Pe'er, Columbia University

Back-Propagation

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N}\tfrac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right) = \frac{1}{N}\sum_n [-(y^n - z_l^n)\, g'(a_l^n)]\, z_k^n = \frac{\sum_n \delta_l^n z_k^n}{N}$$

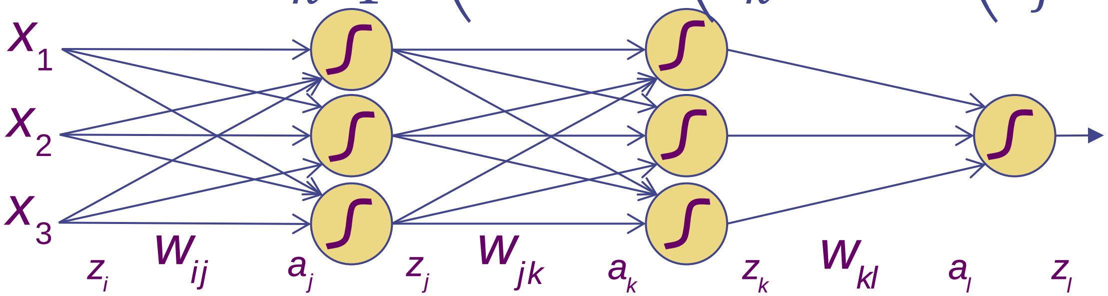
$$\frac{\partial R}{\partial w_{jk}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right) = \frac{1}{N}\sum_n \left[\sum_l \delta_l^n w_{kl}\, g'(a_k^n)\right] z_j^n = \frac{\sum_n \delta_k^n z_j^n}{N}$$

- Any previous (input) layer derivative: repeat the formula!

$$\frac{\partial R}{\partial w_{ij}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_j^n}\right]\left(\frac{\partial a_j^n}{\partial w_{ij}}\right) = \frac{1}{N}\sum_n \left[\sum_k \frac{\partial L^n}{\partial a_k^n}\frac{\partial a_k^n}{\partial a_j^n}\right]\left(\frac{\partial a_j^n}{\partial w_{ij}}\right) = \frac{\sum_n \delta_j^n z_i^n}{N}$$

Back-Propagation

$$R(\theta) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\left(y^n - g\left(\sum_k w_{kl}\, g\left(\sum_j w_{jk}\, g\left(\sum_i w_{ij} x_i^n\right)\right)\right)\right)^2$$



$$\frac{\partial R}{\partial w_{kl}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_l^n}\right]\left(\frac{\partial a_l^n}{\partial w_{kl}}\right) = \frac{1}{N}\sum_n \left[-(y^n - z_l^n)\, g \ldots\right] z_k^n = \frac{\sum_n \delta_l^n z_k^n}{N}$$

$$\frac{\partial R}{\partial w_{jk}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_k^n}\right]\left(\frac{\partial a_k^n}{\partial w_{jk}}\right) = \frac{1}{N}\sum_n \left[\sum_l \delta_l^n w_{kl}\, g'(a_k^n)\right] z_j \ldots \frac{\sum_n \delta_k^n z_j^n}{N}$$

What is this last z?

- Any previous (input) layer derivative: repeat the formula!

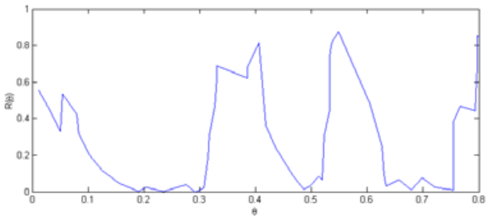
$$\frac{\partial R}{\partial w_{ij}} = \frac{1}{N}\sum_n \left[\frac{\partial L^n}{\partial a_j^n}\right]\left(\frac{\partial a_j^n}{\partial w_{ij}}\right) = \frac{1}{N}\sum_n \left[\sum_j \frac{\partial L^n}{\partial a_k^n}\frac{\partial a_k^n}{\partial a_j^n}\right]\left(\frac{\partial a_j^n}{\partial w_{ij}}\right) = \frac{\sum_n \delta_j^n z_i^n}{N}$$

Back-Propagation

- Again, take small step in direction opposite to gradient

$$w_{ij}^{t+1} = w_{ij}^{t} - \eta \frac{\partial R}{\partial w_{ij}} \qquad w_{jk}^{t+1} = w_{jk}^{t} - \eta \frac{\partial R}{\partial w_{jk}} \qquad w_{kl}^{t+1} = w_{kl}^{t} - \eta \frac{\partial R}{\partial w_{kl}}$$

- Problems with back-prop
  is that MLP over-fits...

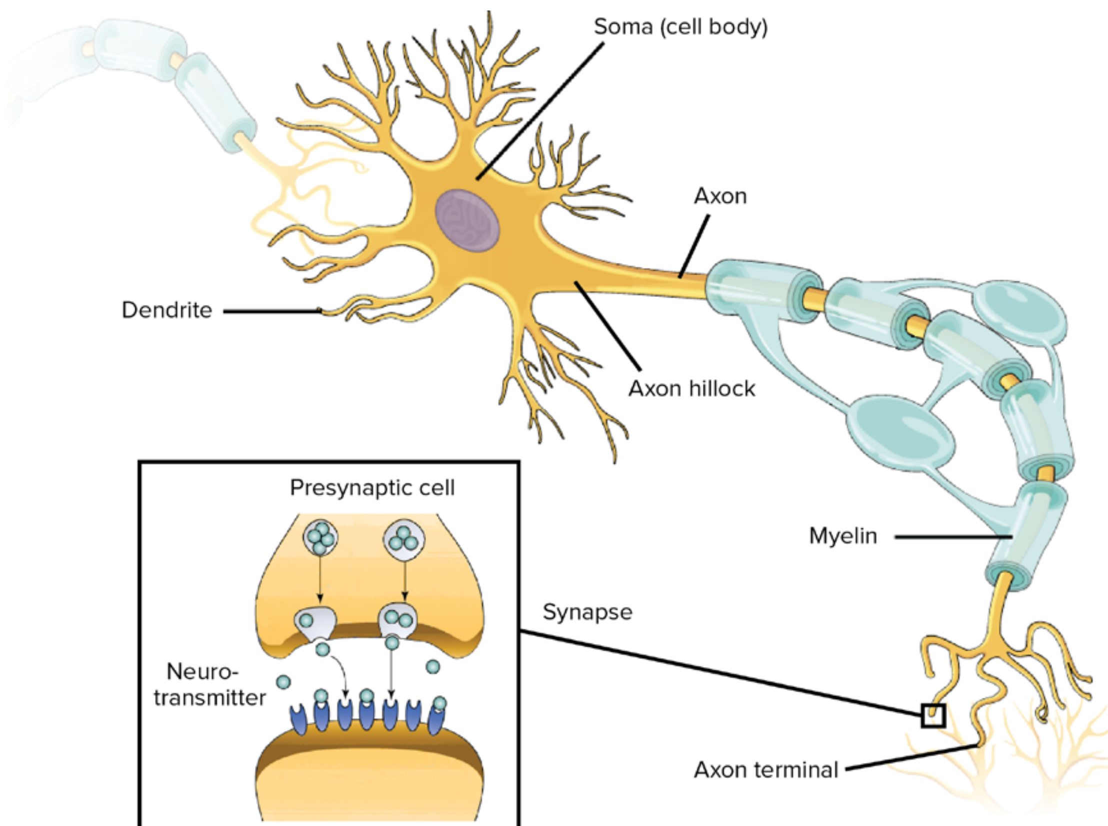


- Other problems: hard to interpret, black-box
- What are the hidden inner layers doing?

- Other main problem: minimum training error
  not minimum testing error...

Neural Networks - Upside

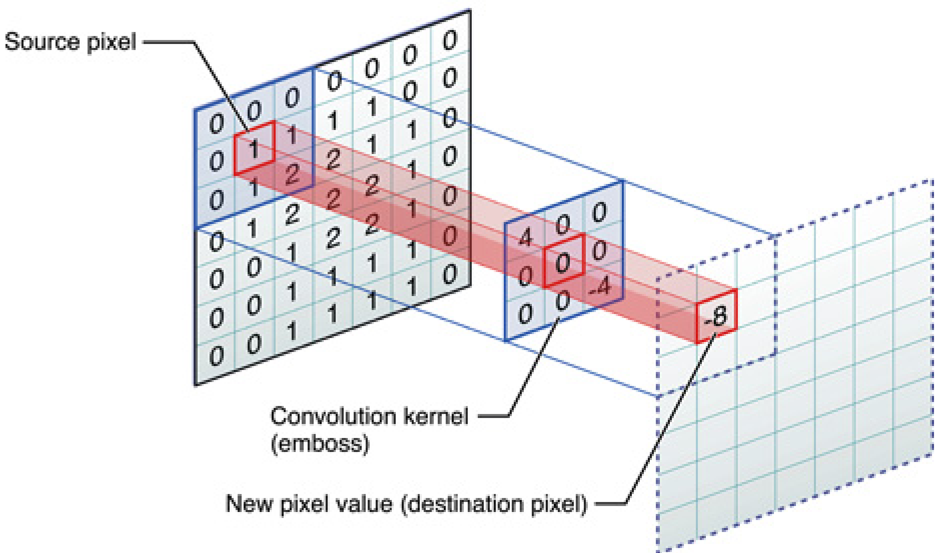
- Live neurons inspiration



Neural Networks - Upside

- Live neurons inspiration

- Flexibility, parameter efficiency, modularity



Neural Networks - Upside

- Live neurons inspiration
- Flexibility, parameter efficiency, modularity
- Success across data-rich domains, tasks
  - Vision, robotics, security, language, genomics...
  - Classification, dimensionality reduction...