# Fitting Neural Networks

$c$, cost $\quad (\underline{f}-\underline{y})^\top(\underline{f}-\underline{y})$, or $-\log \underbrace{P(\underline{y}\,|\,\underline{f})}$

Prob. model with parameters $\underline{f}$

$\underline{y}$ train label $\qquad \underline{f} = g^{(3)}(\underline{a}^{(3)})$

$$\underline{a}^{(3)} = W^{(3)}\underline{h}^{(2)} + \underline{b}^{(3)}$$

$W^{(3)}$ $\quad \underline{b}^{(3)}$

$$\underline{h}^{(2)} = g^{(2)}(\underline{a}^{(2)})$$

$$\underline{a}^{(2)} = W^{(2)}\underline{h}^{(1)} + \underline{b}^{(2)}$$

$W^{(2)}$ $\quad \underline{b}^{(2)}$

$\underline{h}^{(1)}$

$\vdots$

Training input $\underline{x}$

Neural net params:

$$\underline{\theta} = \begin{bmatrix} \mathrm{vec}(W^{(1)}) \\ \mathrm{vec}(\underline{b}^{(1)}) \\ \mathrm{vec}(W^{(2)}) \\ \vdots \end{bmatrix}$$

SGD fitting:

Initialize $\underline{\theta} \neq \underline{0}$

Loop over examples:

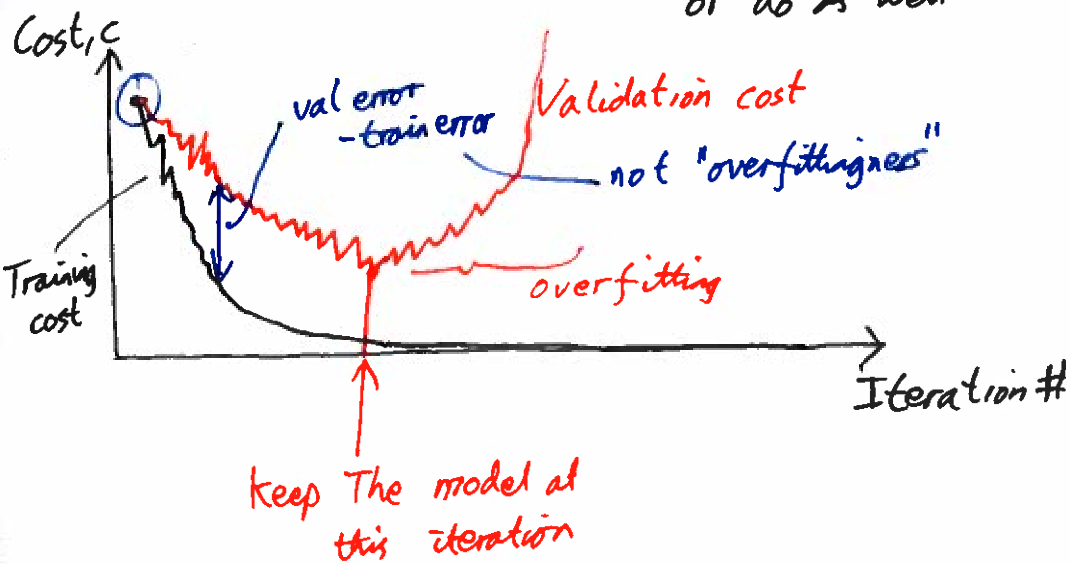
$$\underline{\theta} \leftarrow \underline{\theta} - \eta \nabla_{\underline{\theta}}\, c$$

Today: Termination (early stopping)

Computing $\nabla_{\underline{\theta}}\, c$ (reverse-mode diff.)

# Early Stopping

Form regularization, alternative to L2
or do as well



Every sweep through training set "an epoch": ← or some # updates

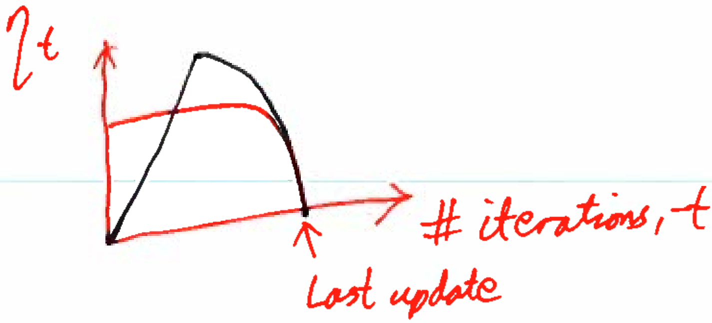
If val cost is the smallest we've seen

Store weights $\theta$ & val cost

If val. cost hasn't improved in 20 evaluations of val set:

Stop return best $\theta$

reduce learning rate $\eta$



# Reverse-mode differentiation "Back-progation"

Works on a compute graph (DAG)

Directed acyclic

## Strategy

For every intermediate $Z$

get $\bar{Z} = \frac{\partial c}{\partial Z}$

For a matrix $\bar{Z}_{ij} = \frac{\partial c}{\partial Z_{ij}}$

## Start at end of computation

Example: $c = (f-y)^2$

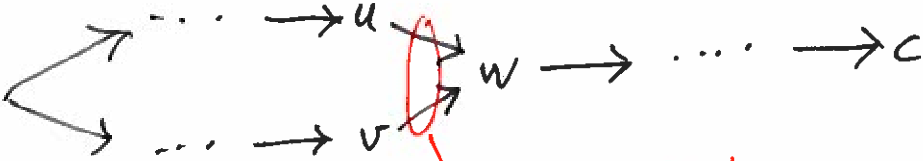
$$\bar{f} = \frac{\partial c}{\partial f} = 2(f-y)$$

If have $\underline{f}$ and $\underline{y}$ ($K \times 1$, $K \times 1$) $\quad c = \sum_k (f_k - y_k)^2$

$$\bar{f}_j = \frac{\partial c}{\partial f_j} = 2(f_j - y_j)$$

$$\bar{\underline{f}} = 2(\underline{f} - \underline{y})$$

# We combine local propagation rules




Local function: eg $w = \frac{u}{v}$

Assume we have $\bar{w} = \frac{\partial c}{\partial w}$

Want $\bar{u} = \frac{\partial c}{\partial u}$, $\bar{v} = \frac{\partial c}{\partial v}$

## Chain rule

$$\bar{u} = \frac{\partial c}{\partial u} = \underbrace{\frac{\partial c}{\partial w}}_{\bar{w}} \underbrace{\frac{\partial w}{\partial u}}$$

Derivative of local function

Depends on u and v and/or w

In example: $\frac{1}{v}$

$$\bar{v} = \underbrace{\frac{\partial c}{\partial w}}_{\bar{w}} \underbrace{\frac{\partial w}{\partial v}}$$

In example $\frac{-u}{v^2}$

# Example: Matrix Multiplication

N×N matrix $N^2$ numbers in it

$O(\cdot)$ cost of matrix-matrix

$\underline{N^3}$ ~~$N^2\log N$? $N\log N$?~~ $\underline{N^{2.373}}$

$$\begin{tikzcd} {} \arrow[r] & X \arrow[dr] & & & & \\ & & \boxed{@} \arrow[r] & Z \arrow[r] & \cdots \arrow[r] & c \\ \cdots & Y \arrow[ur] & & & & \end{tikzcd}$$

$$Z = XY$$

$$\begin{tikzcd} \cdots & \bar{X} \arrow[l] & & & & \\ & & \bar{Z} \arrow[ul] \arrow[dl] & \cdots \arrow[l] & \bar{\xi} \arrow[l] & c \arrow[l] \\ \cdots & \bar{Y} \arrow[l] & & & & \end{tikzcd}$$

$$\bar{X}_{ij} = \frac{\partial c}{\partial X_{ij}} = \sum_{m,n} \underbrace{\frac{\partial c}{\partial Z_{mn}}}_{\bar{Z}_{mn}} \underbrace{\frac{\partial Z_{mn}}{\partial X_{ij}}}_{\delta_{im} Y_{jn}} = \sum_{n} \bar{Z}_{in} \underbrace{Y_{jn}}_{(Y^T)_{nj}}$$

$$\boxed{\bar{X} = \bar{Z} Y^T} \qquad \boxed{\bar{Y} = X^T \bar{Z}}$$

$$Z_{mn} = \sum_p X_{mp} Y_{pn}$$

$$= X_{m1} Y_{1n} + X_{m2} Y_{2n} + \cdots$$

$$\cdots \underbrace{X_{mj} Y_{jn}} + \cdots$$

$$\underset{M \times N}{Z} = \underset{M \times P}{X} \; \underset{P \times N}{Y}$$

$$\underset{P \times N}{\bar{Y}} = \underset{P \times M}{X^T} \; \underset{M \times N}{\bar{Z}}$$