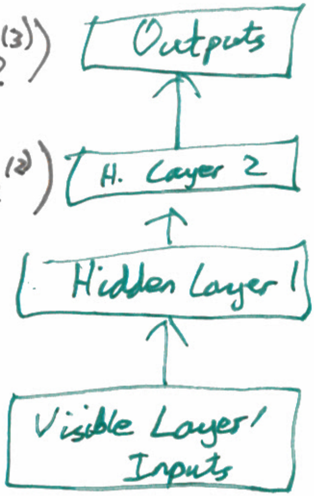
Feed-forward Neural Networks

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

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

$$\underline{h}^{(1)} = g^{(1)}(W^{(1)}\underline{x} + \underline{b}^{(1)})$$



- When f is a scalar, $W^{(3)}\underline{h}^{(2)} = \underline{w}^{(3)T}\underline{h}^{(2)}$

- Other architectures possible:
  "skip connections"
  parameterize the g's non-linearities

- Special layers for images/audio
  ConvNets ... and others.

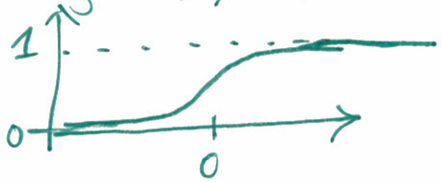
Initialization

Don't set all weights the same.

Naive code $W_{ij}^{(\ell)} \sim N(0,1)$ "use randn()"

$$h_k^{(\ell)} = g\big(\underbrace{W_{k:}^{(\ell)}\, \underline{h}^{(\ell-1)}} + b_k^{(\ell)}\big)$$

$g = \sigma(a)$



Typically how big is this sum over $K^{(\ell-1)}$

Background on expectations:

Typically sum $\sim \pm\sqrt{K^{(\ell-1)}}$

Example Initialization $w \sim N\left(0, \left(1/\sqrt{K}\right)^2\right)$

MLP: More sophisticated suggestions.

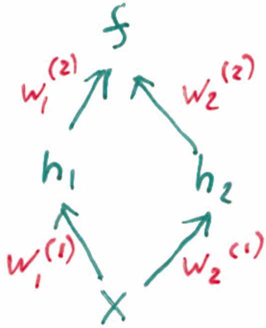
Showed MLPR 2017 L12 3
Again

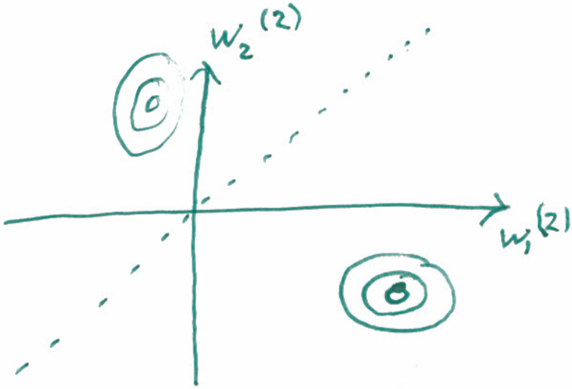
MLPR 2017 L13 1

NN don't ~~con~~ have a convex cost

Convex

$\Rightarrow$ Unique optimum





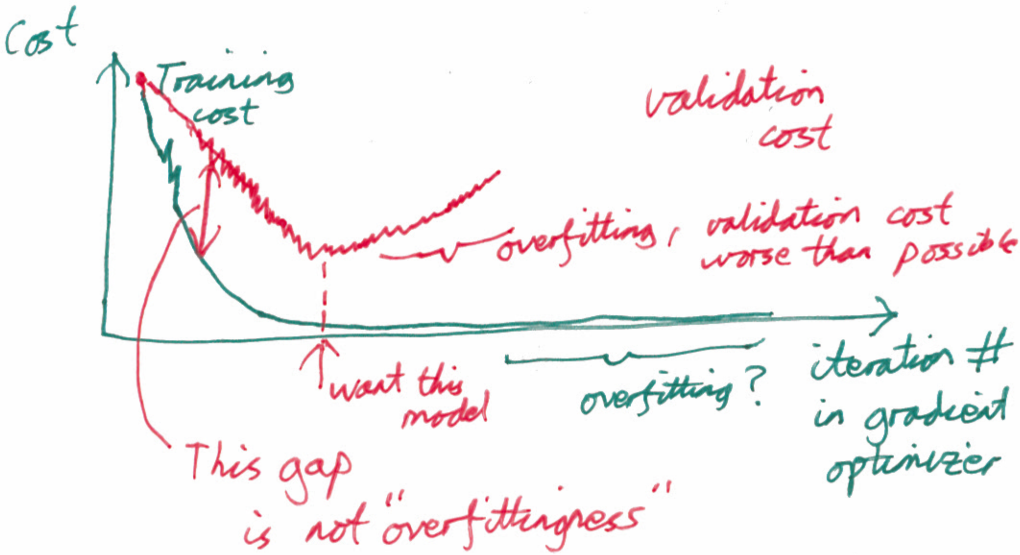
Set $\underline{w}^{(1)}$ to their best values.

- These local optima don't matter
  - → because the function (predictions) are the same.
- Not all optima are equivalent.
- Use heuristics to get good fits.
- Random restarts. — not if network large/expensive

Regularization

Could do L2 regularization

Set λ ... cross-validate.



Every K updates:

If val. cost the smallest I've seen:

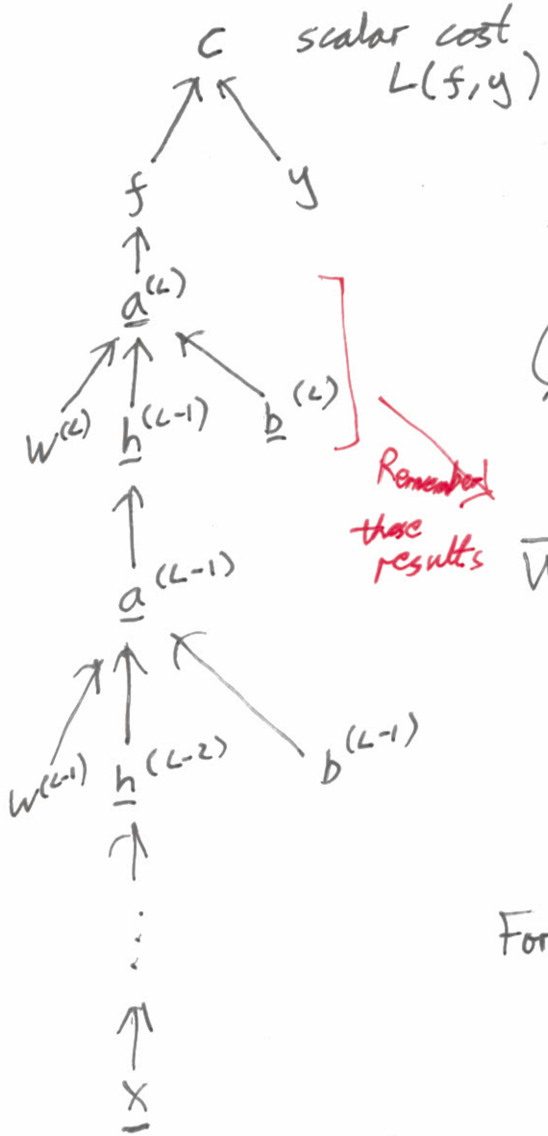
store the weights & val. cost

If val. cost hasn't improved in 20 updates

Stop. Return the weights we stored from best val. score

Getting gradients – Reverse-mode differentiation

Backpropagation



Strategy:

$$\bar{f} = \frac{\partial c}{\partial f}$$

$$\bar{a}_i^{(L)} = \frac{\partial c}{\partial a_i^{(L)}}$$

$\bar{W}^{(L)}$ $\quad\bar{\underline{h}}^{(L-1)}$ $\quad\bar{\underline{b}}^{(L)}$

$$\bar{W}_{ij}^{(L)} = \frac{\partial c}{\partial W_{ij}^{(L)}}$$

For every intermediate θ

get $\bar{\theta} = \dfrac{\partial c}{\partial \theta}$

Derivative Propagation

Local function

Assume we have

$$\bar{w} = \frac{\partial c}{\partial w}$$

Want:

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

Chain rule:

$$\frac{\partial c}{\partial u} = \frac{\partial c}{\partial w}\frac{\partial w}{\partial u}$$

$$= \underbrace{\bar{w}}_{\text{Number which is propagated to us.}} \; \underbrace{\frac{\partial w}{\partial u}}_{\text{Derivative of small local function.}}$$

Number which is propagated to us.

Derivative of small local function.
Look up expression
of u, v (and/or w) to use.