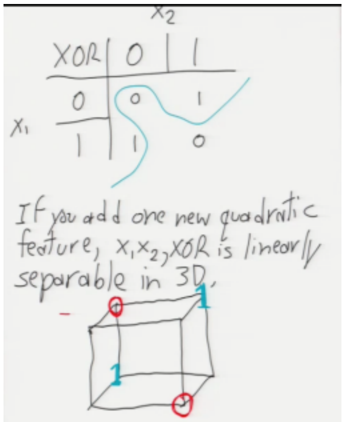
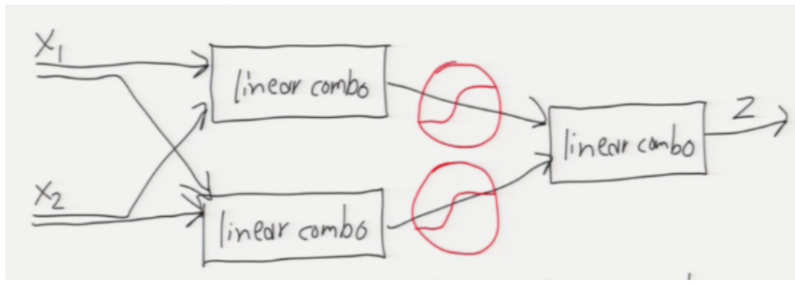
03/30/2016

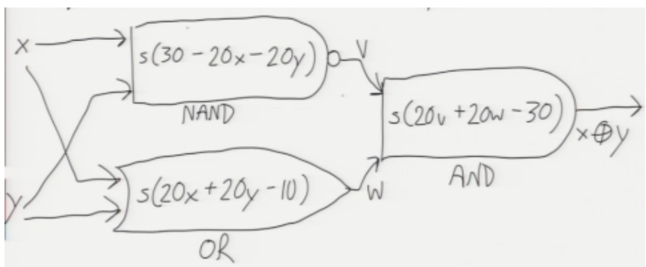
Neural Networks

- Can do both classification and regression.



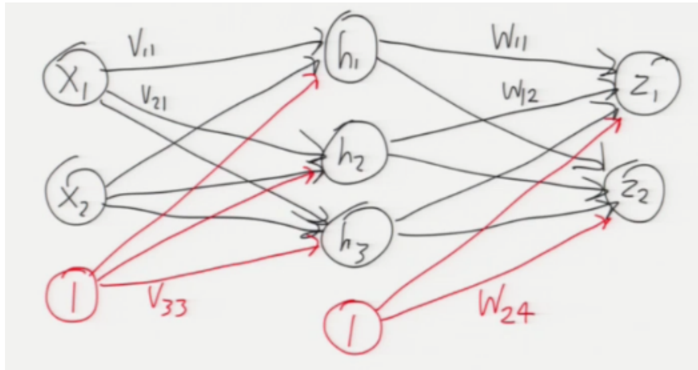


- A linear combination of a linear combination is a linear combination... only works for linearly separable samples.



Network with 1 Hidden Layer

- Input layer: $x_1, \ldots, x_d$; $x_{d+1} = 1$
- Hidden units: $h_1, \ldots, h_m$; $h_{m+1} = 1$
- Output layer: $z_1, \ldots, z_k$
- Layer 1 weights: m x $(d+1)$ matrix V V_i is row i
- Layer 2 weights: k x $(m+1)$ matrix W W_i is row i
- Recall logistic function $s(\gamma) = \frac{1}{1+e^{-\gamma}}$. Other nonlinear functions can be used.
- For vector v, $s(v) = \begin{bmatrix} s(v_1) \\ s(v_2) \\ \vdots \\ s(v_n) \end{bmatrix}$



- $h_i = s(\sum_{j=1}^{n} V_{ij} x_j)$, in short, $h = s(Vx)$.
- $z = s(Wh) = s(W s_1(Vx))$ the one on the s means you have to add a 1 to end of vector before multiplication.

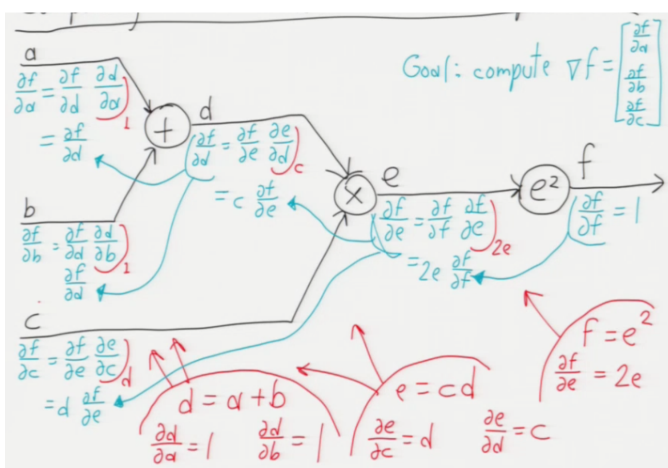
Training

- Usually stochastic or batch gradient descent.
- Pick loss function $L(z, y)$, $z =$ predictions, $y =$ true (values often a vector) e.g. $L(z, y) = |z - y|^2$.
- Cost function is $J(h) = \sum_{i=1}^{n} L(h(X_i, Y_i))$. Start with random weights.
- Usually there are many local minima!
- Rewrite all the weights in V and W as a vector w.

Batch gradient descent:
    $w \leftarrow$ vector of (small) random weights
    repeat:
        $w \leftarrow w - \epsilon \nabla J(w)$

- Hard part is computing $\nabla J(w)$.
- Naive gradient computation: $\mathcal{O}$(units x edges) time.
- Back-propagation: $\mathcal{O}$(edges) time.

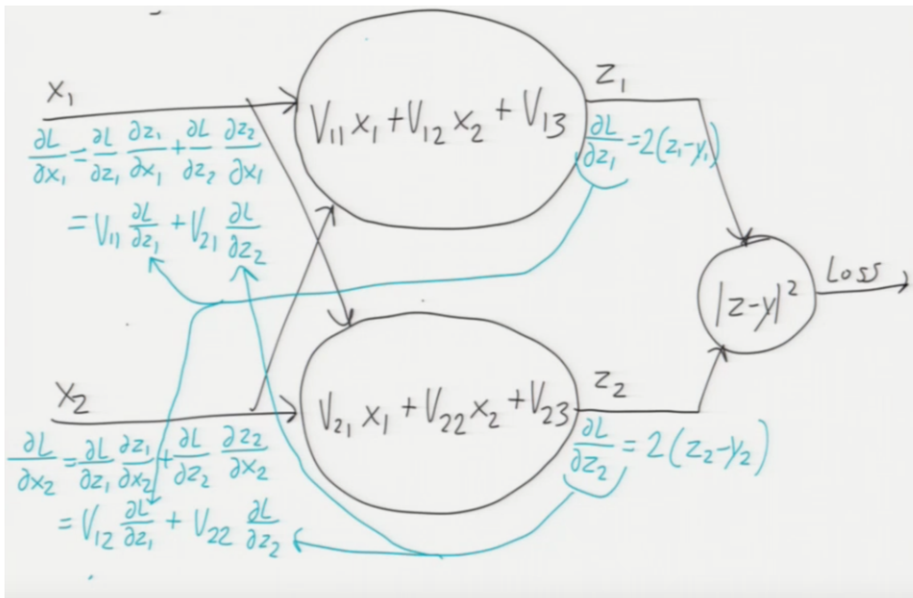
Computing Gradients for Arithmetic Expressions



- Each value z gives partial derivative of the form,

$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial n}\frac{\partial n}{\partial z}$$

- Can always compute $\frac{\partial n}{\partial z}$ in forward pass.
- Compute $\frac{\partial f}{\partial n}$ during backward pass <u>after</u> forward pass.
- This is "back-propagation."



- Algorithm doesn't work if there's cycles.

The Back-propagation Algorithm

- Recall $s'(\gamma) = s(\gamma)(1 - s(\gamma))$
- $h_i = s(V_i \cdot x)$, so $\nabla_{v_i} h_i = s'(V_i \cdot x)x$