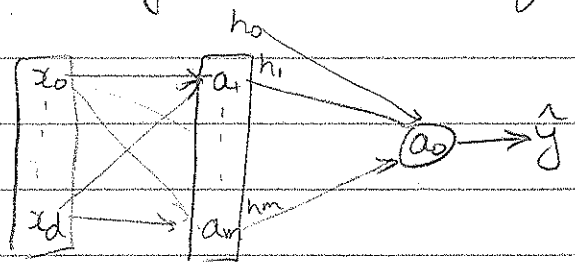
4/16/15

NEURAL NETS : BACKPROP

1. Forward Propagation or "Forward Prop" or Forward Pass
→ Making predictions from a network



Matrix Mult

(a) $\vec{a} = \begin{bmatrix} \vec{w}_1^T \vec{x} \\ \vdots \\ \vec{w}_m^T \vec{x} \end{bmatrix} = W_h \vec{x}$ $\quad W_h = \begin{bmatrix} \leftarrow w_1^T \rightarrow \\ \vdots \\ \leftarrow w_m^T \rightarrow \end{bmatrix}$

Non-linearity

(b) $\vec{h} = g(\vec{a}) = \begin{bmatrix} h_0 = 1 \\ g(a_1) \\ \vdots \\ g(a_m) \end{bmatrix}$

Matrix multiplication

(c) $a_0 = \vec{w}_0 \vec{h}$

Non-linearity

(d) $\hat{y} = g(a_0)$

(2) Parameter learning in Neural Nets

[Assume fixed structure / architecture: 3-layer NN]

Weights to learn: $\{\vec{w}_1 \dots \vec{w}_m\} \in \mathbb{R}^{d+1}$, $\vec{w}_0 \in \mathbb{R}^{m+1}$ $\Big\} \vec{\theta}$

Dataset: $\{(\vec{x}_1, y_1) \dots (\vec{x}_N, y_N)\}$

Objective / Loss $L(\vec{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \text{Loss}(y_i, \hat{y}(\theta)) + \text{Reg}(\vec{\theta})$

many choices based on problem type

Say Regression

$L(\vec{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2}(y_i - \hat{y}(\theta))^2 + \lambda \|\theta\|_2^2$

L_2-loss "least squares" ; L_2-norm

Say Classification

$L(\vec{\theta}) = \frac{1}{N} \sum_{i=1}^{N} -\log P(y_i \mid \vec{x}_i, \vec{\theta}) + \lambda \|\theta\|^2$

maximizing log-likelihood

$-\log\left[\frac{1}{1+e^{-w_0^T \vec{x}}}\right]$ ($y_i = -1$)

Algorithm: Gradient Descent:

$$\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)} - \eta \underbrace{\frac{\partial L}{\partial \vec{\theta}}}$$

How do we compute gradient?
Backprop!

But first, let's study a simpler case:

(2.5) Detour: Assume no hidden units
$g(a) = a$ [Linear Regression]
$\hat{y} = \vec{w}_0^T \vec{x}$

$$L(\vec{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} (y_i - \vec{w}_0^T \vec{x}_i)^2$$ [Assume no Reg for simplicity]

$$\frac{\partial L}{\partial \vec{w}_0} = \frac{1}{N} \sum_{i=1}^{N} -[\underbrace{y_i - w_0^T x_i}_{\text{error}}) \cdot \underbrace{x_i^T}_{\text{pattern/feature}}$$

Batch-Gradient Descent
$$\vec{w}_0^{(t+1)} = w_0^{(t)} + \eta \frac{1}{N} \sum_{i=1}^{N} (y_i - \vec{w}_0^T \vec{x}_i) \cdot \vec{x}_i$$

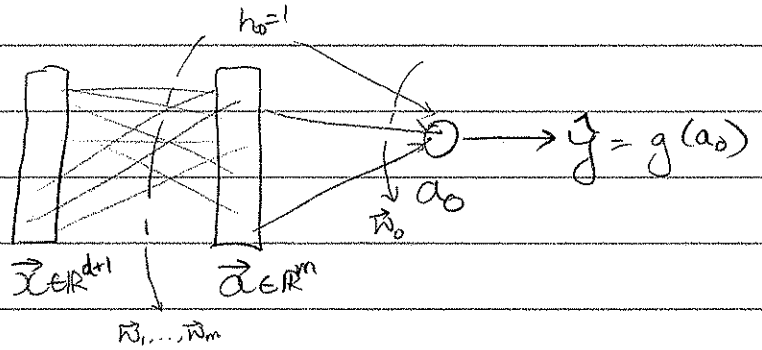
Stochastic Gradient Descent
→ Sample $(\vec{x}, y)$
→ $w_0^{(t+1)} = w_0^{(t)} + \eta (y - \vec{w}_0^T x) \cdot \vec{x}$

→ Similar expression as MLE in LR!

(3) Backward Propagation or BackProp

→ algorithm for computing gradients in NN with Chain Rule

BackProp ≡ Chain Rule + Caching



→ Assume (for simplicity of math/notation)
- → Regression : $(y-\hat{y})^2$
- → No regularization
- → Single data-point $(\vec{x}, y)$

→ $L(\vec{\theta}) = L(\vec{w}_1, \ldots \vec{w}_m, \vec{w}_0) = \frac{1}{2}(y - \hat{y}(\theta))^2$

→ Goal: Compute $\frac{\partial L}{\partial \vec{w}_0}, \left\{\frac{\partial L}{\partial \vec{w}_i}\right\}$

→ Approach: we will compute gradients backwards w.r.t $a_0, \vec{w}_0, a_i, \vec{w}_i$

1) $$\frac{\partial L}{\partial a_0} = \frac{\partial}{\partial a_0}\left[\frac{1}{2}(y - g(a_0))^2\right] = -(y - g(a_0)) \cdot g'(a_0)$$

2) $$\frac{\partial L}{\partial \vec{w}_0} = \underbrace{\frac{\partial L}{\partial a_0} \cdot \frac{\partial a_0}{\partial \vec{w}_0}}_{\text{[Chain Rule]}} = \underbrace{\frac{\partial L}{\partial a_0}}_{\text{already computed above}} \cdot \frac{\partial (\vec{w}_0^T \vec{h})}{\partial \vec{w}_0} = \underbrace{\frac{\partial L}{\partial a_0}}_{\text{error}} \cdot \underbrace{\vec{h}^T}_{\text{pattern}}$$

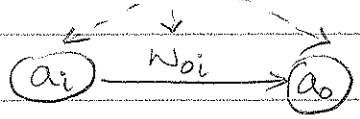
3) $\frac{\partial L}{\partial a_i} = \frac{\partial L}{\partial a_o} \cdot \frac{\partial a_o}{\partial a_i}$ [Chain Rule]

already computed

$$a_o = \vec{w}_o^T h$$
$$= \sum_{i=0}^{m} w_{oi} h_i$$
$$= \sum_{i=0}^{m} w_{oi} g(a_i)$$

$$\frac{\partial a_o}{\partial a_i} = w_{oi} \cdot g'(a_i)$$

$$\Rightarrow \frac{\partial L}{\partial a_i} = \frac{\partial L}{\partial a_o} \cdot w_{oi} \cdot g'(a_i)$$



4) $\frac{\partial L}{\partial \vec{w}_i} = \frac{\partial L}{\partial a_i} \cdot \frac{\partial a_i}{\partial \vec{w}_i}$

already computed

$$a_i = \vec{w}_i^T X$$
$$\frac{\partial a_i}{\partial \vec{w}_i} = \vec{X}^T$$

$$\Rightarrow \frac{\partial L}{\partial \vec{w}_i} = \frac{\partial L}{\partial a_i} \cdot \vec{X}^T$$

error · pattern