Proof of gradient update rule

prove that δ_{ij} is equivalent to $\frac{\partial L}{\partial Z_{ij}}$ for $0 \le i \le L$

$$L = \frac{1}{2}(y-\hat{y})^2; \quad \frac{\partial L}{\partial Z_{L0}} = \frac{\partial L}{\partial \hat{y}} = \hat{y} - y$$

$$Z_{L0} = \sum_{p=0}^{h-1} W_{\hat{y}p}\phi_{Lp} + b_{\hat{y}} = \hat{y}$$

$$\frac{\partial L}{\partial Z_{(L-1)j}} = \frac{\partial L}{\partial Z_{L0}} \cdot \frac{\partial Z_{L0}}{\partial \phi_{Lj}} \cdot \frac{\partial \phi_{Lj}}{\partial Z_{(L-1)j}} = \frac{\partial L}{\partial Z_{L0}} \cdot W_{\hat{y}j} \cdot A'(Z_{(L-1)j})$$

$$\frac{\partial L}{\partial Z_{(L-2)j}} = \sum_{p=0}^{h-1} \frac{\partial L}{\partial Z_{(L-1)p}} \cdot \frac{\partial Z_{(L-1)p}}{\partial \phi_{(L-1)j}} \cdot \frac{\partial \phi_{(L-1)j}}{\partial Z_{(L-2)j}}$$

$$= \sum_{p=0}^{h-1} \frac{\partial L}{\partial Z_{(L-1)p}} \cdot W_{\phi_{(L-1)p}j} \cdot A'(Z_{(L-2)j})$$

Statement S_i: $\frac{\partial L}{\partial Z_{ij}} = \sum_{p=0}^{h-1} \frac{\partial L}{\partial Z_{(i+1)p}} \cdot \frac{\partial Z_{(i+1)p}}{\partial \phi_{(i+1)j}} \cdot \frac{\partial \phi_{(i+1)j}}{\partial Z_{ij}}$

S_{L-2} is true, show $S_{i+1} \to S_i$

$$\frac{\partial L}{\partial Z_{ij}} = \sum_{p=0}^{h-1} \left(\sum_{k=0}^{h-1} \frac{\partial L}{\partial Z_{(i+2)k}} \cdot \frac{\partial Z_{(i+2)k}}{\partial \phi_{(i+2)p}} \cdot \frac{\partial \phi_{(i+2)p}}{\partial Z_{(i+1)p}} \right) \cdot \frac{\partial Z_{(i+1)p}}{\partial \phi_{(i+1)j}} \cdot \frac{\partial \phi_{(i+1)j}}{\partial Z_{ij}}$$

$$= \sum_{p=0}^{h-1} \frac{\partial L}{\partial Z_{(i+1)p}} \cdot \frac{\partial Z_{(i+1)p}}{\partial \phi_{(i+1)j}} \cdot \frac{\partial \phi_{(i+1)j}}{\partial Z_{ij}}$$

$$\frac{\partial L}{\partial Z_{ij}} = \sum_{p=0}^{h-1} \frac{\partial L}{\partial Z_{(i+1)p}} \cdot W_{\phi_{(i+1)p}j} \cdot A'(Z_{ij}) \quad \text{for } 0 \le i \le L-2$$

So we can represent $\frac{\partial L}{\partial Z_{ij}}$ as δ_{ij}

$$\delta_{(L-1)j} = (\hat{y} - y) \cdot W_{\hat{y}j} \cdot A'(Z_{(L-1)j})$$

$$\delta_{L0} = \hat{y} - y \qquad \delta_{(L-2)j} = \delta_{ij} = \sum_{p=0}^{h-1} \delta_{(i+1)p} \cdot W_{\phi_{(i+1)p}j} \cdot A'(Z_{ij})$$

$\forall\, 0 \le i \le L-2$

backward pass for some x_i, y_i
error backpropogation

$\delta_{Lo} = \hat{y} - y$; $T(\delta_L) = \theta(1)$

error term for neuron ij (Lo is output neuron)

$\delta_{(L-1)j} = \delta_{Lo} \cdot w_{\hat{y}j} \cdot A'(Z_{(L-1)j})$; $T(\delta_{(L-1)}) = \theta(h)$
$T(A') = \theta(1)$

$$\delta_{(L-2)j} = \sum_{p=0}^{h-1} \delta_{(L-1)p} \cdot w_{\phi_{(L-1)pj}} \cdot A'(Z_{(L-2)j}); \quad T(\delta_{(L-2)}) = \theta(h^2)$$

$$\delta_{ij} = \sum_{p=0}^{h-1} \delta_{(i+1)p} \cdot w_{\phi_{(i+1)pj}} \cdot A'(Z_{ij}); \quad T(\delta_{ij}) = \theta(h^2)$$

$T(\delta) = \theta(1) + \theta(h) + \theta((L-2)h^2) \rightarrow T(\delta) = O(Lh^2)$

gradient update

$w_{\phi_{ij}p} = w_{\phi_{ij}p} - \eta \delta_{ij} \phi_{ip}$ $\qquad$ $w_{\hat{y}p} = w_{\hat{y}p} - \eta \delta_{Lo} \phi_{Lp}$

$b_{\phi_{ij}} = b_{\phi_{ij}} - \eta \delta_{ij}$ $\qquad$ $b_{\hat{y}} = b_{\hat{y}} - \eta \delta_{Lo}$

$T(\text{update}) = \theta(h) + \theta((L-2)h^2) + \theta(nh)$

$$\rightarrow T(\text{update}) = \begin{cases} O(Lh^2) \text{ if } Lh > n \\ \theta(nh) \text{ else} \end{cases}$$

gradient clipping (done before update)

$$\delta_{ij} = \begin{cases} \psi \cdot \text{sign}(\delta_{ij}) \text{ if } |\delta_{ij}| > \psi \\ \delta_{ij} \text{ else} \end{cases}$$ for some clipping threshold ψ

$T(\text{clipping}) = \theta(1) + \theta((L-1) \cdot h) = O(Lh)$

$T(\text{backward}) = T(\delta) + T(\text{clipping}) + T(\text{update})$

$$\rightarrow T(\text{backward}) = \begin{cases} O(Lh^2) \text{ if } Lh > n \\ \theta(nh) \text{ else} \end{cases}$$

$T(\text{pass}) = T(\text{forward}) + T(\text{backward})$ $\qquad$ total time complexity

$$\rightarrow T(\text{pass}) = \begin{cases} O(Lh^2) \text{ if } Lh > n \\ \theta(nh) \text{ else} \end{cases}$$

$$T(\text{pass}_{\text{total}}) = \begin{cases} O(MLh^2) \text{ if } Lh > n \\ \theta(Mnh) \text{ else} \end{cases}$$
for M samples

$$T(\text{train}) = \begin{cases} O(KMLh^2) \text{ if } Lh > n \\ \theta(KMnh) \text{ else} \end{cases}$$
for K training iterations

Multilayer perceptron

Input/target

- $\underset{\substack{M\times n \\ \text{input}}}{\mathcal{X}} = \left[\left[x_{cj}\right]_{j=0}^{n-1}\right]_{c=0}^{M-1}$; $\underset{\substack{M\times 1 \\ \text{target}}}{y} = \left[y_c\right]_{c=0}^{M-1}$, M = # samples in dataset, n = # input features

$$T(\text{read}) = \Theta(Mn) + \Theta(M) = \Theta(Mn)$$

other parameters: h = hidden layer size, L = hidden layer dim (# HLs)

η = learning rate

Weight/bias initialization

$$SD_i = \begin{cases} \sqrt{\frac{2}{n}} & i=0 \\ \sqrt{\frac{2}{h}} & i>0 \end{cases} \qquad ND_i = N(0, SD_i) \qquad \text{gen = random \# generator}$$

$$\underset{\substack{\text{weights on hidden} \\ \text{layer neurons}}}{W_\phi} = \left[\left[\begin{matrix} \left[ND_i(\text{gen})\right]_{p=0}^{n-1} & i=0 \\ \left[ND_i(\text{gen})\right]_{p=0}^{h-1} & i>0 \end{matrix}\right]_{j=0}^{h-1}\right]_{i=0}^{L-1}$$

for $0 \le i \le L$

- $\underset{\substack{\text{weights on output} \\ \text{neuron}}}{W_{\hat{y}}} = \left[ND_i(\text{gen})\right]_{j=0}^{h-1}$ $\quad \underset{\substack{\text{bias on output} \\ \text{neuron}}}{b_{\hat{y}}} = ND_i(\text{gen})$ $\quad \underset{\substack{\text{biases on} \\ \text{hidden layer} \\ \text{neurons}}}{b_\phi} = \left[\left[ND_i(\text{gen})\right]_{j=0}^{h-1}\right]_{i=0}^{L-1}$

$$T(\text{init}) = \Theta(1) + \Theta(h) + \Theta(Lh) + \Theta((L-1)h^2) + \Theta(nh)$$

$$\rightarrow T(\text{init}) = \begin{cases} O(Lh^2) & \text{if } Lh > n \\ \Theta(nh) & \text{else} \end{cases}$$

Forward pass

for some x_c

$$\underset{\substack{\text{hidden layers} \\ L\times h}}{\phi} = \left[\left[\phi_{ij}\right]_{j=0}^{h-1}\right]_{i=0}^{L-1}, \quad \phi_{ij} = A(z_{ij}) \text{ for some activation function } A, \quad T(A) = \Theta(1)$$

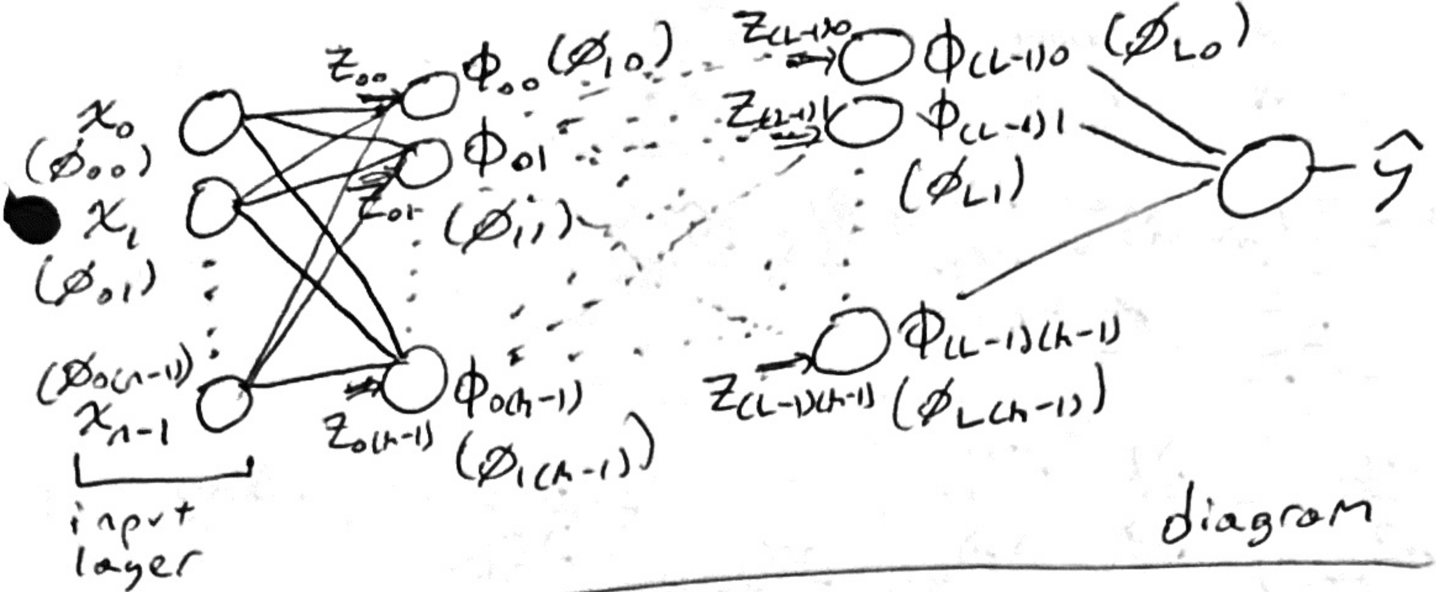
$$z_{ij} = \begin{cases} \sum_{p=0}^{n-1} W_{\phi_{ijp}} x_p + b_{\phi_{ij}} & i=0 \\ \sum_{p=0}^{h-1} W_{\phi_{ijp}} \varnothing_{ip} + b_{\phi_{ij}} & i>0 \end{cases}$$

$$T(\phi) = \Theta((L-1)\cdot h^2) + \Theta(nh) \quad \Big| \quad T(\phi_{ij}) = T(z_{ij}) \quad \Big| \quad T(z_{ij}) = \begin{cases} \Theta(n) & \text{if } i=0 \\ \Theta(h) & \text{if } i>0 \end{cases}$$

$$\rightarrow T(\phi) = \begin{cases} O(Lh^2) & \text{if } Lh>n \\ \Theta(nh) & \text{else} \end{cases} \qquad \underset{\text{output neuron}}{\hat{y}} = \sum_{p=0}^{h-1} W_{\hat{y}p} \varnothing_{(L-1)p} + b_{\hat{y}}, \quad T(\hat{y}) = \Theta(h)$$

- $$\underset{\substack{\text{hidden/output} \\ \text{layer inputs}}}{\varnothing_i} = \begin{bmatrix} \left[x_j\right]_{j=0}^{n-1} & i=0 \\ \left[\phi_{(i-1)j}\right]_{j=0}^{h-1} & i>0 \end{bmatrix} ; \quad \varnothing = \left[\varnothing_c\right]_{c=0}^{M-1}$$

$$T(\text{forward}) = T(\phi) + T(\hat{y}) = \begin{cases} O(Lh^2) & Lh>n \\ \Theta(nh) & \text{else} \end{cases}$$



diagram

---

activation functions

$\text{reLu}(z) = \begin{cases} z & \text{if } z>0 \\ 0 & \text{else} \end{cases}$; $\text{reLu}'(z) = \begin{cases} 1 & \text{if } z>0 \\ 0 & \text{else} \end{cases}$

$\text{lR}(z) = \begin{cases} z & \text{if } z>0 \\ \alpha z & \text{else} \end{cases}$; $\text{lR}'(z) = \begin{cases} 1 & \text{if } z>0 \\ \alpha & \text{else} \end{cases}$

leaky relu — α is a small constant

$\sigma(z) = \dfrac{1}{1+e^{-z}}$; $\sigma'(z) = \sigma(z)(1-\sigma(z))$

$\tanh(z) = \dfrac{e^z - e^{-z}}{e^z + e^{-z}}$; $\tanh'(z) = 1 - \tanh^2(z)$

$\forall A$ above, $T(A) = \theta(1)$, $T(A') = \theta(1)$

---

time recordings

$m = 15{,}000 = 15 \cdot 10^3$

$n = 6{,}528$ $\quad n > Lh$

$h = 5$ $\quad \to T(\tanh) = \theta(kmnh)$

$L = 2$ $\quad = c \cdot 3 \cdot 6528 \cdot 5 \cdot 15000 = c \cdot 97920 \cdot 15 \cdot 10^3$

$k = 3$ $\quad = c \cdot 1{,}468{,}800 \cdot 10^3$

running time: 194 seconds $\quad = c \cdot 14{,}688 \cdot 10^5$

$= 194 \cdot 10^9 \, nS$ $\quad = c \cdot 1.4688 \cdot 10^9$

running time 2: 189 seconds

$c = \dfrac{189}{1.4688} \approx 129$ $\quad c = \dfrac{194 \cdot 10^9}{1.4688 \cdot 10^9} \approx 132.08$