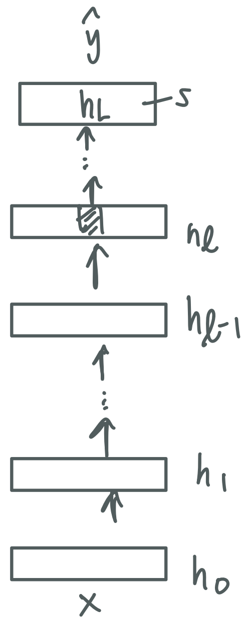
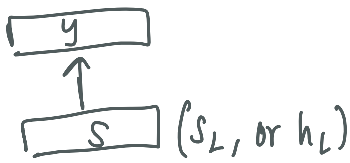
classification



$h_\ell = r_\ell(s_\ell)$

$s_\ell = W_\ell h_{\ell-1} + b_\ell$

Top layer



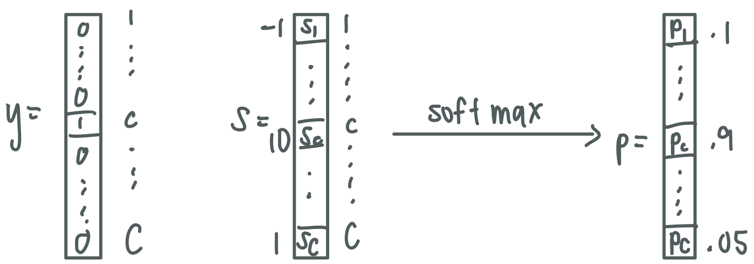
back-prop

$$x \rightarrow h_1 \rightarrow \cdots \rightarrow h_{\ell-1} \rightarrow h_\ell \rightarrow \cdots \rightarrow h_L \rightarrow y$$

$\frac{dLoss}{dh_{\ell-1}} \xleftarrow{①} \frac{dLoss}{dh_\ell}$ $\qquad (s) - e = \frac{dLoss}{ds}$

$W_1, b_1 \qquad W_\ell, b_\ell \overset{②}{\rightleftarrows} \frac{dLoss}{dW_\ell} \quad db_\ell \qquad W_L, b_L$

for each training example
w & b parameters
h hidden variables



as prediction of y

$$p_c = \frac{e^{s_c}}{\sum_{c'=1}^{C} e^{s_{c'}}}$$

summation over all categories

$$= \frac{e^{s_c}}{Z}$$ ← normalizing constant

$$\sum_{c=1}^{C} p_c = 1$$

$$\text{Likelihood} = \prod_{c=1}^{C} p_c^{\,y_c}$$

$$= \text{prob(observed category)}$$

$$\text{log-lik} = \sum_{c=1}^{C} y_c \log p_c$$

$$= \sum_{c=1}^{C} y_c (s_c - \log Z)$$

$$= \sum_{c=1}^{C} y_c s_c - \log Z \qquad \left(\text{b/c } \sum_{c=1}^{C} y_c = 1\right)$$

$$\text{Loss} = -(\text{log-lik}) = -\left(\sum_{c=1}^{C} y_c s_c - \log Z\right)$$

$$\underset{\substack{\text{(top layer}\\ \text{vector } h_L)}}{\frac{d\text{Loss}}{ds}} = \begin{array}{c} \left(\begin{array}{c} \\ \frac{d\text{Loss}}{ds_k} \\ \\ \end{array}\right) \end{array} \begin{array}{c} 1 \\ \vdots \\ k \\ \vdots \\ C \end{array} = \begin{array}{c} \left(\begin{array}{c} \\ y_k - \overbrace{\frac{d}{ds_k}\log Z}^{p_k} \\ \\ \end{array}\right) \end{array} \begin{array}{c} 1 \\ \vdots \\ k \\ \vdots \\ C \end{array} = -(\vec{y} - \vec{p}) = -\vec{e}$$

multinomial logistic regression

$$\frac{d}{ds_k}\log Z = \frac{\frac{d}{ds_k} Z}{Z} = \frac{\frac{d}{ds_k}\sum_{c=1}^{C} e^{s_c}}{Z} = \frac{e^{s_k}}{Z} = p_k$$

binary logistic

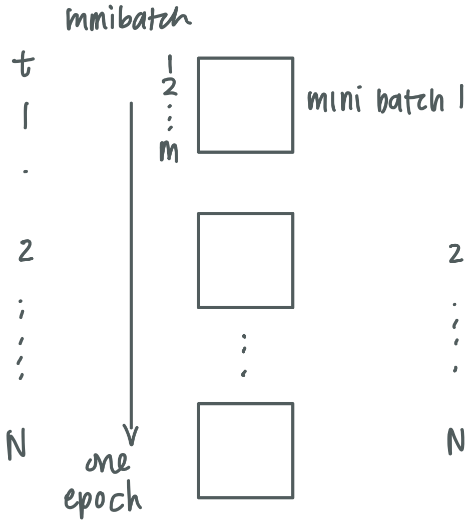
vector

scalar

sigmoid

$$y = \begin{array}{c} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{array} \begin{array}{c} + \\ - \end{array} \qquad s = \begin{bmatrix} s \\ 0 \end{bmatrix} \begin{array}{c} + \\ - \end{array} \qquad p = \begin{array}{c} p \\ 1-p \end{array} \begin{bmatrix} \frac{e^s}{e^0 + e^s} \\ \frac{e^0}{e^0 + e^s} \end{bmatrix} \begin{array}{c} + \\ - \end{array}$$

Stochastic gradient descent (SGD)



step size, learning rate $\eta_t \propto \frac{1}{t}$

$$\theta_{t+1} = \theta_t - \eta_t \frac{1}{m} \underbrace{\sum_{i=1}^{m}}_{\text{(mini batch)}} \frac{d\,Loss_i}{d\theta_t}$$

$$\theta = (W_\ell, b_\ell,\ \ell = 1, \dots, L)$$

Gradient

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ \vdots \\ x_d \end{pmatrix} \qquad \Delta X = \begin{pmatrix} \Delta x_1 \\ \vdots \\ \Delta x_k \\ \vdots \\ \Delta x_d \end{pmatrix}$$

$$f(x) = f(x_1, \dots, x_k, \dots, x_d)$$

(surface, hill, valley)

Taylor expansion

$$f(x+\Delta x) = f(x_1+\Delta x_1, \dots, x_k+\Delta x_k, \dots, x_d+\Delta x_d)$$

$$\doteq f(x_1 \dots x_k \dots x_d) + \frac{df}{dx_1}\Delta x_1 + \dots + \frac{df}{dx_k}\Delta x_k + \dots + \frac{df}{dx_d}\Delta x_d$$

1st order Taylor expansion approximation

$$\doteq f(x) + \left\langle \begin{array}{c} 1 \\ \vdots \\ k \\ \vdots \\ d \end{array} \begin{pmatrix} \\ \\ \frac{df}{dx_k} \\ \\ \\ \end{pmatrix}, \begin{array}{c} 1 \\ \vdots \\ k \\ \vdots \\ d \end{array} \begin{pmatrix} \\ \\ \Delta x_k \\ \\ \\ \end{pmatrix} \right\rangle$$

$$\doteq f(x) + \langle f'(x), \Delta x \rangle$$

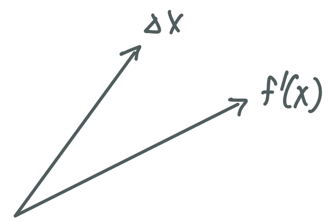
$$\doteq f(x) + |f'(x)|\,|\Delta x| \underbrace{\cos\theta}$$

max $\theta = 0$

$\Delta x \propto f'(x)$

steepest direction

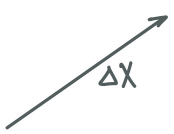
gradient

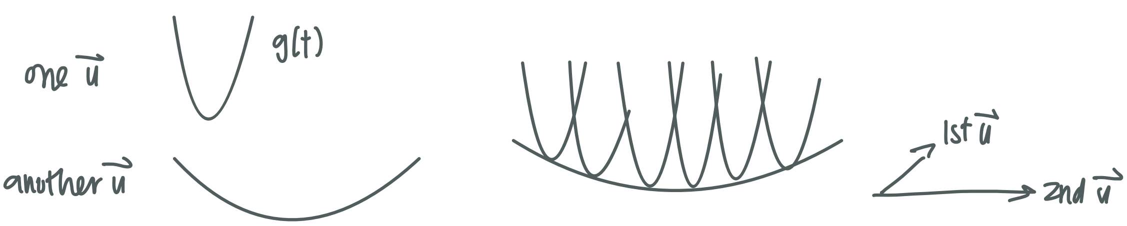


$$g(t) = f(x + t\cdot\vec{u}) \doteq g(0) + g'(0)t + \frac{1}{2}g''(0)t^2$$

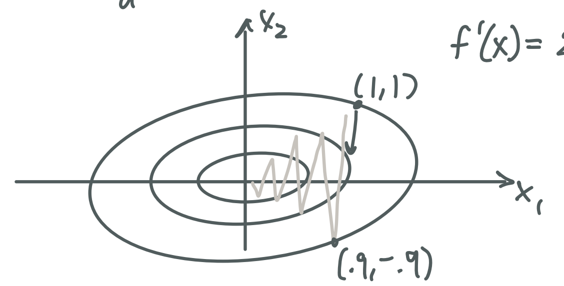
$$= f(x) + \langle f'(x), \Delta x \rangle + \frac{1}{2}\Delta x^T f''(x) \Delta x$$

$\Delta x = t \cdot \vec{u}$ $\quad (|\vec{u}| = 1)$





$$f(x) = \frac{x_1^2}{a^2} + x_2^2$$



$$f'(x) = 2\begin{pmatrix} \frac{x_1}{a^2} \\ x_2 \end{pmatrix}$$

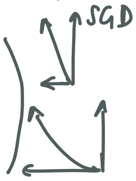
$$= 2\begin{pmatrix} \frac{1}{100} \\ 1 \end{pmatrix} \quad \text{for } (x_1, x_2) = (1,1)$$

$$\rightarrow 2\begin{pmatrix} \frac{.9}{100} \\ -.9 \end{pmatrix}$$

V_t :

momentum

G_t : $\begin{pmatrix} \left(\frac{.9}{100}\right)^2 \\ (-.9)^2 \end{pmatrix}$



Adam optimizer

$$g_t = \frac{1}{m}\sum_{i=1}^{n} \frac{d\text{Loss}_i}{d\theta t}$$

SGD: $\theta_{t+1} = \theta_t - \eta_t g_t$

Adam: $V_t = \gamma V_{t-1} + (1-\gamma) g_t$ momentum velocity

$G_t = \beta G_{t-1} + (1-\beta) g_t^2$ magnitude (elementwise g_t^2) $\quad$ β between 0 & 1

$\tilde{V}_t = V_t / (1-\gamma) \quad \tilde{G}_t = G_t / (1-\beta)$ adaptive gradient

$\theta_{t+1} = \theta_t - \eta_t \frac{\tilde{V}_t}{\sqrt{\tilde{G}_t + \varepsilon}}$ (element wise)

$G_0 = 0$

$G_1 = (1-\beta) g_1^2$

$G_2 = \beta(1-\beta) g_1^2 + (1-\beta) g_2^2$

$= (1-\beta)(\beta g_1^2 + g_2^2)$

$G_3 = \beta(1-\beta)(\beta g_1^2 + g_2^2) + (1-\beta) g_3^2$

$= (1-\beta)(\beta^2 g_1^2 + \beta g_2^2 + g_3^2)$

...

$G_t = (1-\beta)(\beta^{t-1} g_1^2 + \beta^{t-1} g_2^2 + \ldots + \beta g_{t-1}^2 + g_t^2)$

accumulating recent g_t^2
recursive formula allows accumulation of magnitude.

sparse features