

$$\begin{bmatrix} w_{11} & \dots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \dots & w_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} + \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

*n × m matrix*  
*↑ rows      ↑ cols*

let  $\left\{ \begin{array}{l} V_1 = w_{11}x_1 + w_{12}x_2 + \dots + w_{1m}x_m + b_1 \\ V_2 = w_{21}x_1 + w_{22}x_2 + \dots + w_{2m}x_m + b_2 \\ \vdots \\ V_n = w_{n1}x_1 + w_{n2}x_2 + \dots + w_{nm}x_m + b_n \end{array} \right\}$  be the results of each row

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax} \begin{bmatrix} W_{1,1}x_1 + W_{1,2}x_2 + W_{1,3}x_3 + b_1 \\ W_{2,1}x_1 + W_{2,2}x_2 + W_{2,3}x_3 + b_2 \\ W_{3,1}x_1 + W_{3,2}x_2 + W_{3,3}x_3 + b_3 \end{bmatrix} \begin{array}{l} V_1 \\ V_2 \\ V_3 \end{array}$$

softmax formula:  $y_i = \text{softmax}(V_i) \quad \sum_j e^{V_j} = V = e^{V_1} + \dots + e^{V_n}$

$$= \frac{e^{V_i}}{\sum_j e^{V_j}} = \frac{e^{V_i}}{V}$$

↓ *actual*      ↓ *prediction*

Equation To Minimize:  $J(y) = - \sum_i y'_i \ln(y_i)$

$$= - \left[ y'_1 \ln \left( \frac{e^{V_1}}{V} \right) + \dots + y'_n \ln \left( \frac{e^{V_n}}{V} \right) \right]$$

$$= - \left[ y'_1 (\ln e^{V_1} - \ln V) + \dots + y'_n (\ln e^{V_n} - \ln V) \right]$$

$$= - \left[ y'_1 (V_1 - \ln V) + \dots + y'_n (V_n - \ln V) \right]$$

$$= - \left[ (y'_1 V_1 + \dots + y'_n V_n) - \ln V (y'_1 + \dots + y'_n) \right]$$

$$= -L(y_i'v_1 + \dots + y_n'v_n) - \ln V (y_1' + \dots + y_n')$$

recall  $v_i = w_{i1}x_1 + w_{i2}x_2 + \dots + w_{im}x_m + b_i$

$$\frac{\partial J}{\partial w_{ij}} = - \underbrace{\left[ y_i \frac{\partial v_i}{\partial w_{ij}} - \left( \frac{\partial}{\partial w_{ij}} \ln V \right) (y_1' + \dots + y_n') \right]}_{\text{call it } Y = 1 \text{ b/c "one hot" encoding}}$$

all  $v$  except  
 $v_i$  become 0 when taking  $\frac{\partial}{\partial w_{ij}}$

$$\Rightarrow \frac{\partial v_i}{\partial w_{ij}} = x_j$$

$$\begin{aligned} \frac{\partial}{\partial w_{ij}} \ln V &= \frac{\partial}{\partial w_{ij}} \ln(e^{v_1} + \dots + e^{v_n}) \quad \rightarrow \text{only } e^{v_i} \text{ remains} \\ &= \frac{1}{e^{v_1} + \dots + e^{v_n}} \frac{\partial}{\partial w_{ij}} [e^{v_1} + \dots + e^{v_n}] \\ &= \frac{1}{e^{v_1} + \dots + e^{v_n}} \left( \frac{\partial}{\partial w_{ij}} e^{v_i} \right) \\ &= \frac{1}{e^{v_1} + \dots + e^{v_n}} e^{v_i} \left( \frac{\partial}{\partial w_{ij}} v_i \right) \quad \text{by chain rule} \\ &= \frac{e^{v_i} x_j}{e^{v_1} + \dots + e^{v_n}} = \frac{e^{v_i} x_j}{V} \end{aligned}$$

$$\begin{aligned} \Rightarrow \frac{\partial J}{\partial w_{ij}} &= - \left[ y_i x_j - \frac{e^{v_i} x_j Y}{V} \right] \\ &= \boxed{-x_j \left[ y_i - \frac{e^{v_i} Y}{V} \right]} \end{aligned}$$

Gradient Descent:  $w_{ij} := w_{ij} - \alpha \frac{\partial}{\partial w_{ij}} J(\omega)$  formula

for each  $\omega$  update

where  $\alpha$  = learning rate

$$\Rightarrow \boxed{w_{ij} := w_{ij} + \alpha x_j \left[ y_i - \frac{e^{v_i} Y}{V} \right]}$$

$$\Rightarrow \boxed{\omega_{ij} := \omega_{ij} + \alpha x_j [y_i - \frac{e^{v_i y_i}}{V}]} \quad \checkmark$$

Note:  $V = 1$

$$\Rightarrow \omega_{ij} := \omega_{ij} + \alpha x_j [y_i - \frac{e^{v_i}}{V}]$$

$\uparrow$  actual       $\uparrow$  prediction

for multiple training examples,

$$\omega_{ij} := \omega_{ij} + \alpha \sum_k x_j^k [y_i^k - \frac{e^{v_i(k)}}{V(k)}]$$

Stochastic Method can be used

↳ faster by using estimation

for ( $k=0$ ;  $k < \# \text{training examples}$ ;  $k++$ ) { }  
 for (each  $\omega_{ij}$ ) { }  
 $\omega_{ij} := \omega_{ij} + \alpha x_j^k [y_i^k - \frac{e^{v_i(k)}}{V(k)}]$   
 }

repeat until convergence.

⇒ only looks @ 1 training example per update