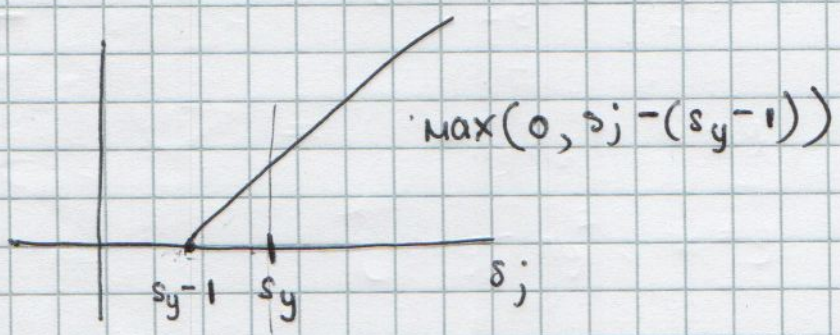
7.1 Hinge loss (the loss that is used soft margin SVM where data are not linearly separable)

$$L = \sum_{j \neq y} \max(0, s_j - s_y + 1)$$

suppose we have targets $t = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ i.e $y = 2$ and $s = \begin{pmatrix} -3 \\ 1.0 \\ 0.5 \\ -1.0 \\ -0.4 \end{pmatrix}$ — $s_y = 0.5$

$$s_j - s_y = \begin{pmatrix} -3.5 \\ 0.5 \\ 0 \\ -1.5 \\ -0.9 \end{pmatrix} \qquad s - s_y + 1 = \begin{pmatrix} -2.5 \\ 1.5 \\ \hline -0.5 \\ 0.1 \end{pmatrix} \quad (j \neq y)$$

$L = 1.5 + 0.1$
$= 1.6$

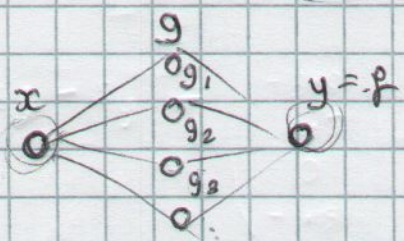


7.2 Chain rule for scalars/vectors

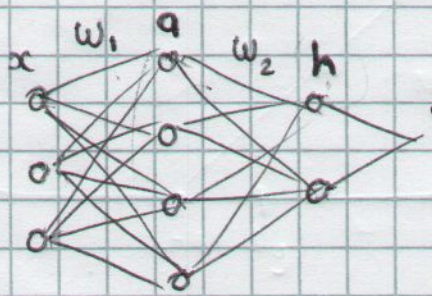
$$y = f(g(x))$$

all scalars: $\frac{\partial y}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x}$ $\quad f = y$

x, y scalars, g vector: $\frac{\partial y}{\partial x} = \sum_i \frac{\partial f}{\partial g_i}\frac{\partial g_i}{\partial x}$ dot product.



e.g. 2 layer NN



$$a = \max(0, w_1^T x)$$

$$h = w_2 a$$

$$e = |h - t|^2$$

$$\frac{\partial e}{\partial h_i} = 2(h_i - t_i)$$

$$\frac{\partial e}{\partial a_i} = \sum_j \frac{\partial e}{\partial h_j}\frac{\partial h_j}{\partial a_i} \qquad j = 0, 1$$

$$\frac{\partial e}{\partial w} = \sum_k \frac{\partial e}{\partial a_k}\frac{\partial a_k}{\partial w} = \sum_k \sum_j \frac{\partial e}{\partial h_j}\frac{\partial h_j}{\partial a_k}\frac{\partial a_k}{\partial w}$$

single scalar weight in w_1

7.3

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial x}$$

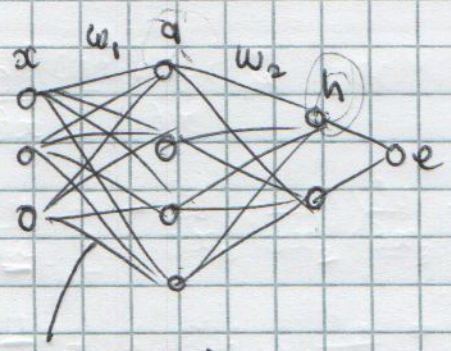
$$\frac{\partial y}{\partial \underline{x}} = \begin{pmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} & \frac{\partial y}{\partial x_3} & \cdots \end{pmatrix}$$

$$\frac{\partial \underline{y}}{\partial \underline{x}} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots \\ \vdots & & \end{pmatrix}$$

i^{th} $\frac{\partial y}{\partial x_i}$ vector gradient

i, j $\frac{\partial y_i}{\partial x_j}$ Jacobian J

7.4



$\frac{\partial e}{\partial w}$

① forward mode: start at input x

compute $\frac{\partial a}{\partial w} \rightarrow \frac{\partial h}{\partial w} \rightarrow \frac{\partial e}{\partial w}$

② reverse mode: start at output e

compute $\frac{\partial e}{\partial h} \rightarrow \frac{\partial e}{\partial a} \rightarrow \frac{\partial e}{\partial w}$

reverse more efficient if # outputs $<<$ # w_i

7.5

convolution backward pass

$$\underset{k}{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}} * \underset{I_i}{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}} = \underset{I_o}{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & i & h & g & 0 \\ 0 & f & e & d & 0 \\ 0 & c & b & a & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}$$

x, y

$$\frac{\partial e}{\partial I_i}(x,y) = \sum_{-1 \le dx \le 1} \sum_{-1 \le dy \le 1} \frac{\partial e}{\partial I_o}(x+dx, y+dy)\, k(-dx,-dy)$$

$$\underline{\nabla I_i} = \nabla I_o * k(-dx,-dy) = \underline{\text{corr}(\nabla I_o, k)}$$

backward pass of convolution is ~~correlation~~ correlation.