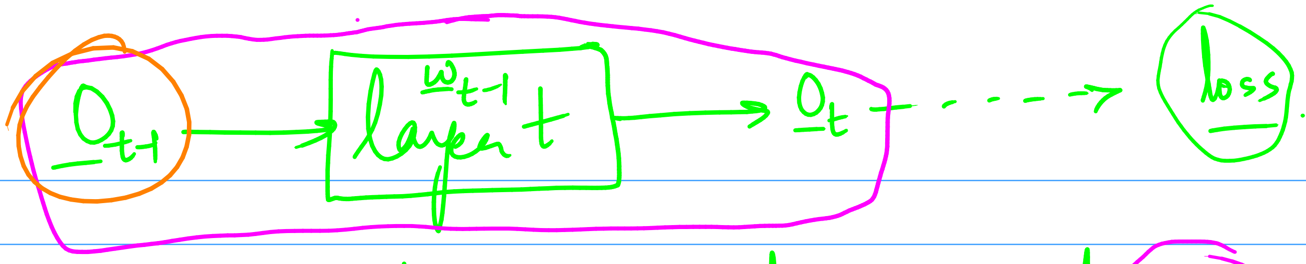
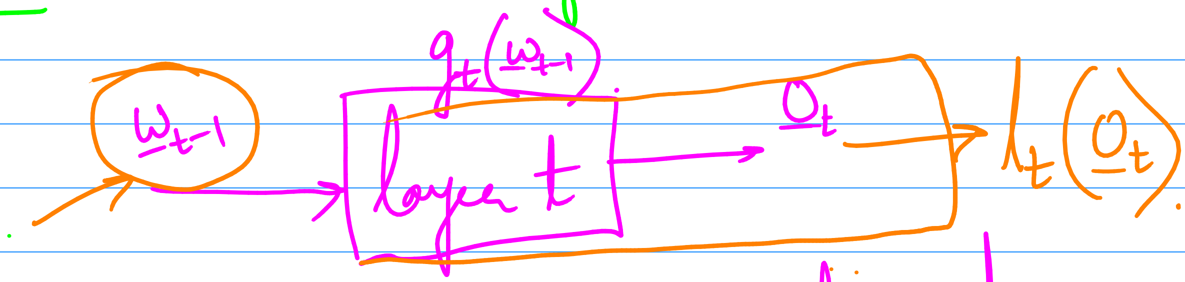


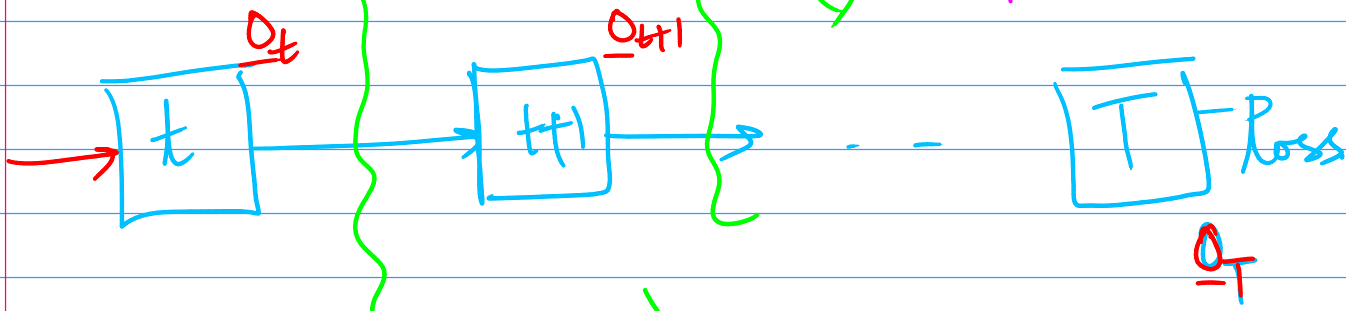
SGD

derivatives of the loss w.r.t. w_{t-1}



Allows to compute derivatives

$l_t(O_t)$ $\quad$ $l_{t+1}(O_{t+1})$ $\quad$ $J_{w_{t-1}}(O_t)$



$$l_t(O_t) \equiv \text{Final}(O_T)$$
$$\equiv l_{t+1}(O_{t+1})$$

used in SGD weight update

Composition

$w_{t-1} \to$ layer g_t t $\to O_t$

$\text{Aff} \to \sigma \to l_t$

Jacobian of $g_t(w_{t-1})$ w.r.t. input w_{t-1}

$$J_{w_{t-1}}(O_t) \quad J_{w_{t-1}}(g_t) = J_{O_t}(l_t) \cdot J_{a_t}(\sigma)\; J_{w_{t-1}}(\text{Affine})$$

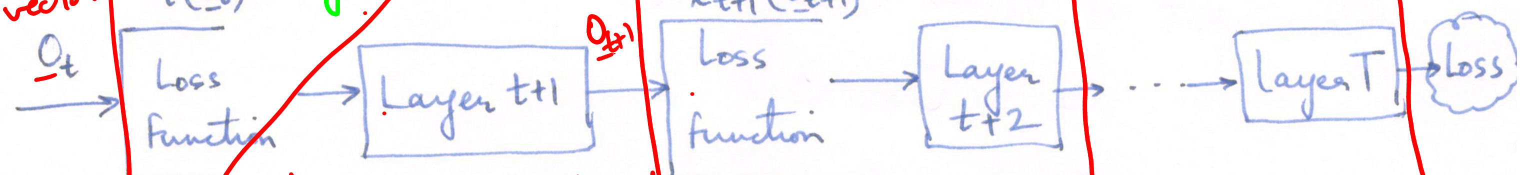
?? $\;l_t(O_t)$ $\quad \sigma(a_t) \equiv O_t$ $\quad a_t \equiv B w_{t-1}$

depends on further network

W matrix

w vector

$g_t(w_{t-1}) \equiv O_t$



subsequent network

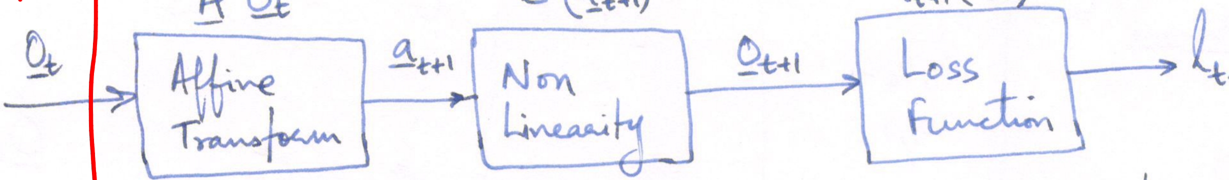
$$l_t(O_t) = \underbrace{l_{t+1}(O_{t+1})} = l_{t+1}(\sigma(\underbrace{W_t O_t}_{a_{t+1}}))$$

O_{t+1}

Back propagated

$$J_{O_t}(l_t) = J_{O_{t+1}}(l_{t+1})\; J_{a_{t+1}}(O_{t+1})\; J_{O_t}(a_{t+1})$$

$l_t(O_t)$ $\quad l_{t+1}(O_{t+1})$ $\quad \sigma(a_{t+1})$ $\quad \text{Affine}(O_t)$



$l_t(O_t) \to$ Loss from further network $\to$

Computing Jacobian for Affine Transform

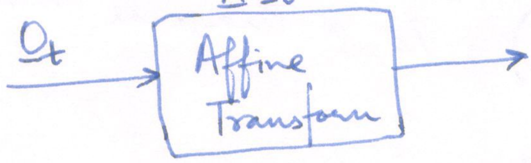
General Rule

if $f(w) = A w$ then $\boxed{J_w(f) = A}$

$$\begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \frac{\partial f_1}{\partial w_2} & \frac{\partial f_1}{\partial w_3} & \cdots \\ \frac{\partial f_2}{\partial w_1} & \frac{\partial f_2}{\partial w_2} & \frac{\partial f_2}{\partial w_3} & \cdots \end{bmatrix}$$

$y = cx$

$\frac{dy}{dx} = c$



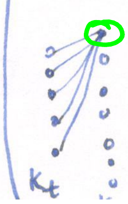
$a_{t+1} = W_t O_t$

$$\boxed{J_{O_t}(a_{t+1}) = W_t}$$

matrix

$$W_t = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,k_t} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,k_t} \\ \vdots & & & \\ w_{k_{t+1},1} & w_{k_{t+1},2} & & w_{k_{t+1},k_t} \end{bmatrix}$$

layer t to t+1



Computing Jacobian for Affine Transform

If $f(\underline{w}) = B\,\underline{w}$ then

$$J_{\underline{w}}(\underline{f}) = \underline{B}$$

When input is vector $\underline{w}_{t-1}$ then

$$\underline{B} \equiv \underline{O}_{t-1}$$

$$\underline{a}_t = \underline{O}_{t-1}\,\underline{w}_{t-1}$$

$$J_{\underline{w}_{t-1}}(\underline{a}_t) = \underline{O}_{t-1}$$

Here $\underline{O}_{t-1}$ is the output matrix constructed in a special way

vector

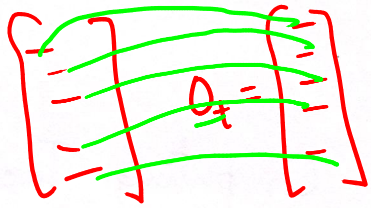
$$\underline{w}_{t-1} = \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,k_t-1} \\ w_{2,1} \\ \vdots \\ w_{k_t,k_{t-1}} \end{bmatrix}$$

$k_{t-1}\ k_t$

$k_{t-1} \times k_t$

$$\underline{O}_{t-1} = \begin{array}{c} \\ k_t \end{array} \begin{bmatrix} \underline{O}_{t-1}^T & 0_z & \cdots & 0^z \\ 0^z & \underline{O}_{t-1}^T & \cdots & 0^z \\ \vdots & \vdots & & \vdots \\ 0^z & 0^z & \cdots & \underline{O}_{t-1} \end{bmatrix}$$

$\underline{a}_t$

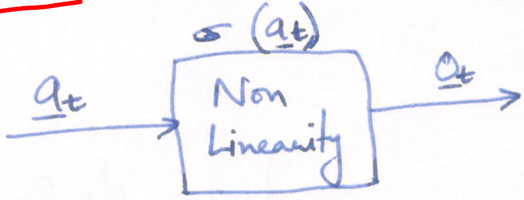


Sigmoid Function (Non-Linearity)

element-wise

$$\underline{O}_{t_i} = \frac{1}{1+\exp(-a_{t_i})}$$

$J_{\underline{a}_t}(\underline{O}_t)$ is a diagonal matrix because $\underline{O}_{t_i}$ depends only on $\underline{a}_{t_i}$



Each diagonal entry is $\sigma'(\underline{a}_{t_i})$

$\underline{a}^{m\times 1}$ $\quad$ $\underline{b}^{n\times 1}$

$$\underset{m\times n}{J} = \begin{array}{c} \\ m \end{array} \overset{\xrightarrow{\quad\quad n \quad\quad}}{\begin{bmatrix} \frac{\partial a_1}{\partial b_1} & \frac{\partial a_1}{\partial b_2} & \cdots & \frac{\partial a_1}{\partial b_n} \\ \vdots & & & \\ \frac{\partial a_m}{\partial b_1} & \cdots & \cdots & \frac{\partial a_m}{\partial b_n} \end{bmatrix}}$$

$\underset{m\times 1}{\underline{a}} \rightarrow \underline{b}_{m\times 1}$ element-wise operations.

$$\begin{bmatrix} a_1 \\ a_2 \\ a_i \\ \end{bmatrix} \longrightarrow \begin{bmatrix} b_1 \\ b_2 \\ b_i \\ \end{bmatrix}$$

$\frac{\partial b_1}{\partial a_1} \neq 0 \qquad \frac{\partial b_1}{\partial a_{2/3/4..}} = 0$

$\frac{\partial b_2}{\partial a_2} \neq 0 \qquad$ others zero

$$J_{\underline{a}}(\underline{b}) = \begin{bmatrix} \ddots & & 0 \\ & \ddots & \\ 0 & & \ddots \end{bmatrix}$$ diagonal.