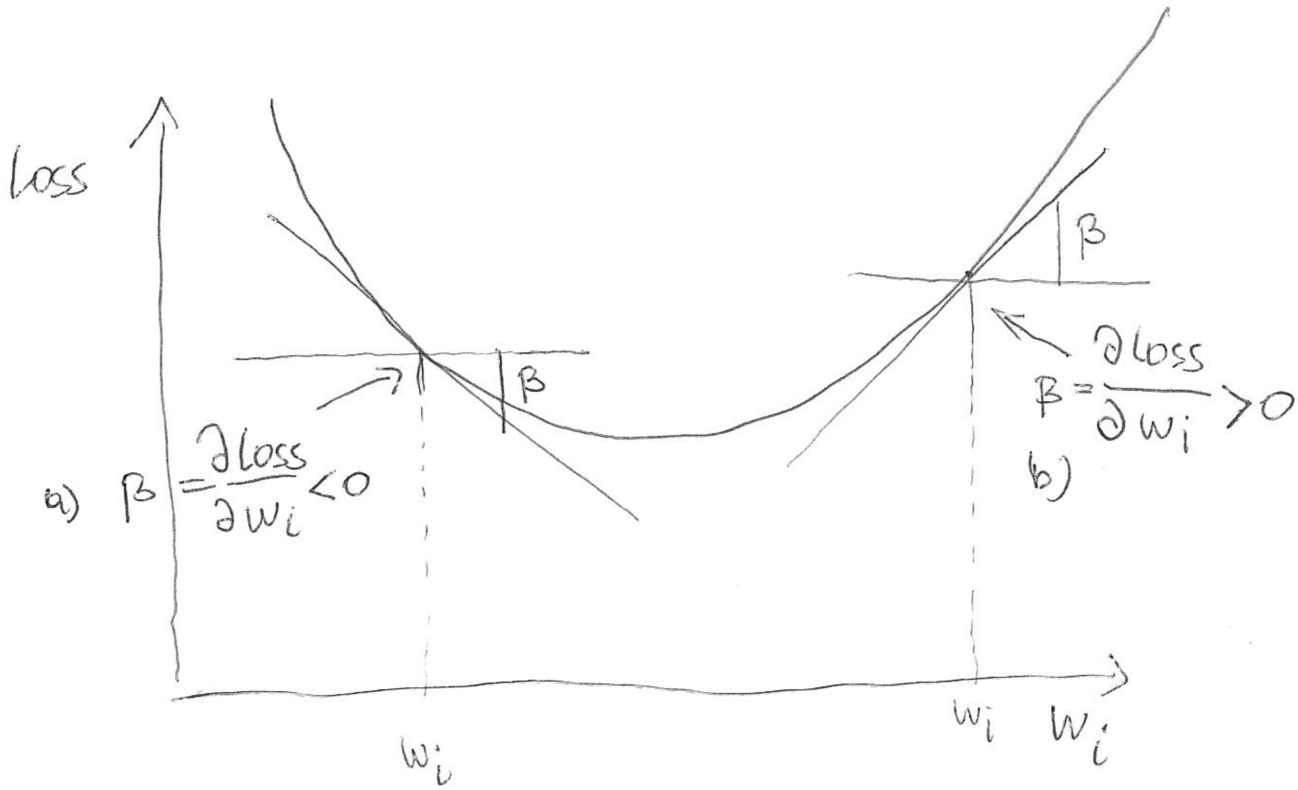
Perceptron Learning

$$h_w(\vec{x}) = g(in), \quad in = \sum_i w_i x_i$$

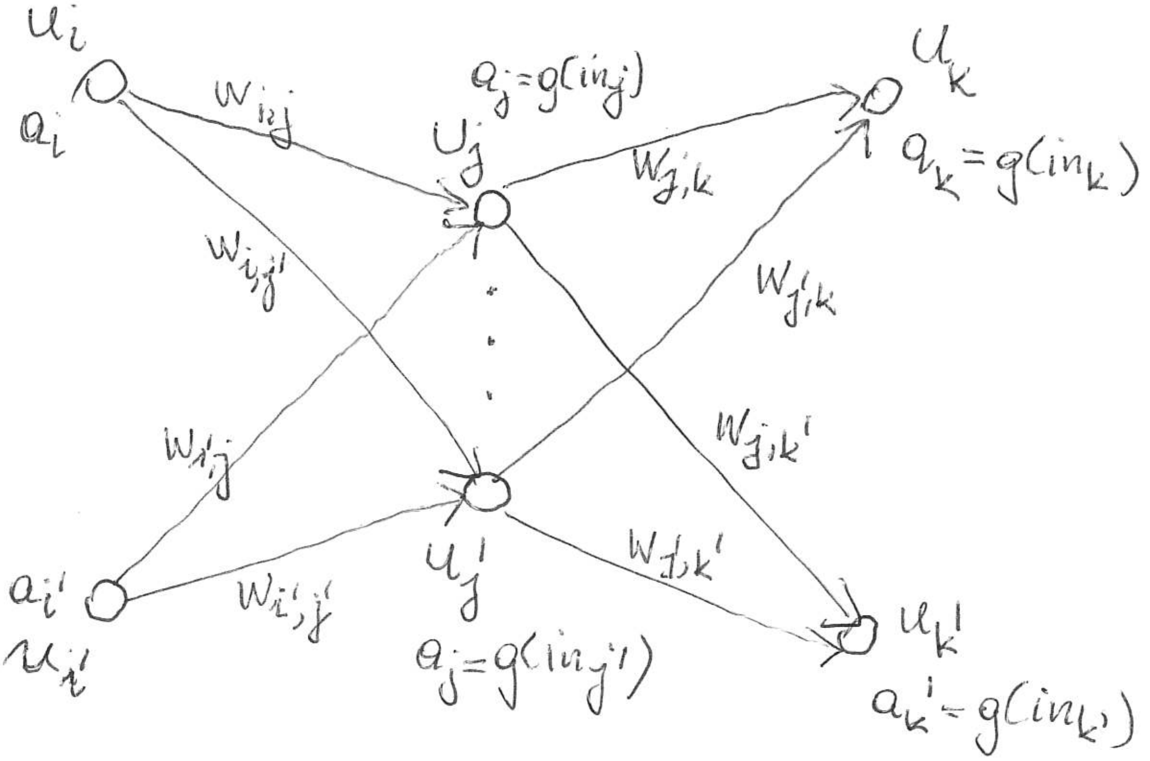
$$Loss(w) = Err^2 = (y - h_w(x))^2$$



a) increase w_i by β to affect $g(in)$

b) decrease w_i by β to affect $g(in)$

Multi-Layer Perceptrons



$$\frac{\partial a_k}{\partial w_{i,j}} = \frac{\partial g(in_k)}{\partial w_{i,j}} = g'(in_k) \cdot \frac{\partial in_k}{\partial w_{i,j}}$$

$= \sum_{j'} w_{j',k} \cdot a_{j'}$

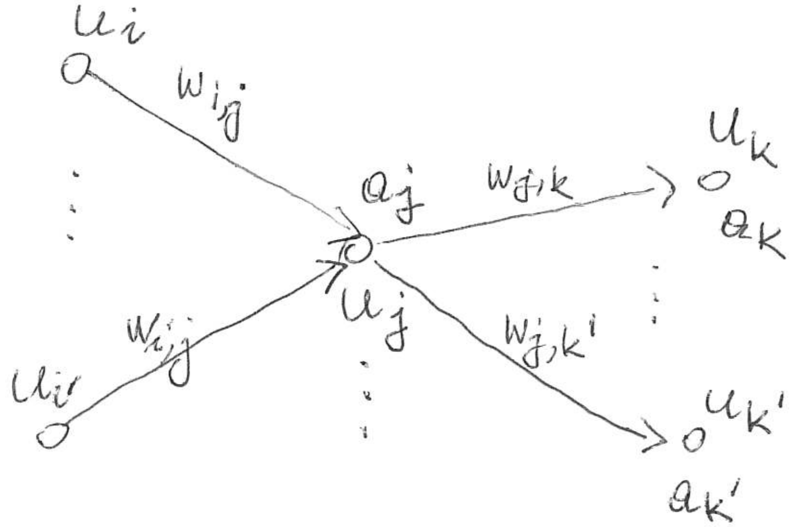
only $w_{j',k}$, $j'=j$ matters!

$$\frac{\partial a_j}{\partial w_{i,j}} = \frac{\partial g(in_j)}{\partial w_{i,j}} = g'(in_j) \cdot \frac{\partial in_j}{\partial w_{i,j}}$$

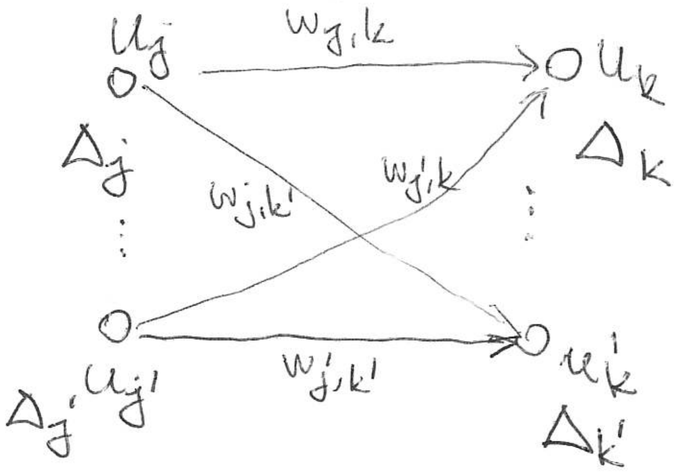
$= \sum_{i'} w_{i',j} \cdot a_{i'}$

only $w_{i',j}$, $i'=i$ matters

Multi-Layer Perceptron Learning



$a_j = g(in_j)$ $a_k = g(in_k)$



$\Delta_j = g'(in_j) \cdot \sum_k w_{j,k} \Delta_k$ $\Delta_k = Err_k \cdot g'(in_k)$

Push back weighted errors