→ Algorithm that performs update for each example

$$initialize\ \theta \Rightarrow \theta = \{W_0, W_1, W_2 \ldots, W_L\}$$

→ For N epochs, for each training example X_i, Y_i

- Pre-activation layer

$$g(x)^{i+1} = \sum_i^n (w_i * x_i) \Rightarrow W^i \times X^i$$

- hidden layer activation

$$L^{i+1} = sigmoid(g(x)^{i+1})$$

→ Back propagation

$$\nabla_{n-1} = \nabla_n * W_{n-1}^T \qquad \nabla = \frac{\partial E}{\partial w}$$

→ Use the chain rule to efficiently compute gradients, top to bottom

$$Error = \frac{1}{2}\sum_i^n (y - \hat{y})^2 \qquad \hat{y} = Sigmoid(x_i \times w_i)$$

$$\frac{\partial E}{\partial w} = \frac{\partial}{\partial w}\frac{1}{2}\sum_i^n (y - \hat{y})^2$$

$$\frac{\partial E}{\partial w} = \sum_i^n (y - \hat{y})(-\frac{\partial E}{\partial w}\hat{y}) \qquad \Rightarrow (-\frac{\partial E}{\partial w}\hat{y}) = \hat{y}(1 - \hat{y})$$