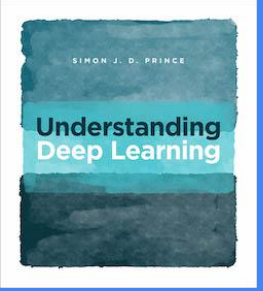
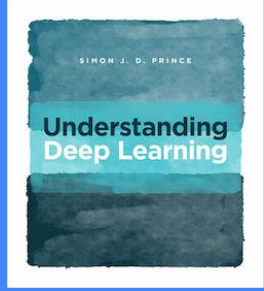
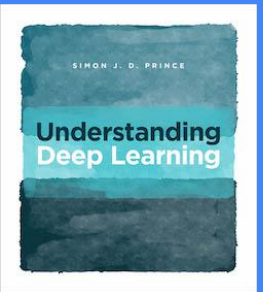
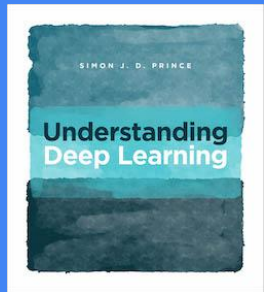




Fitting Models





Gradient Descent: Step 1

Compute the derivatives of the loss with respect to the parameters

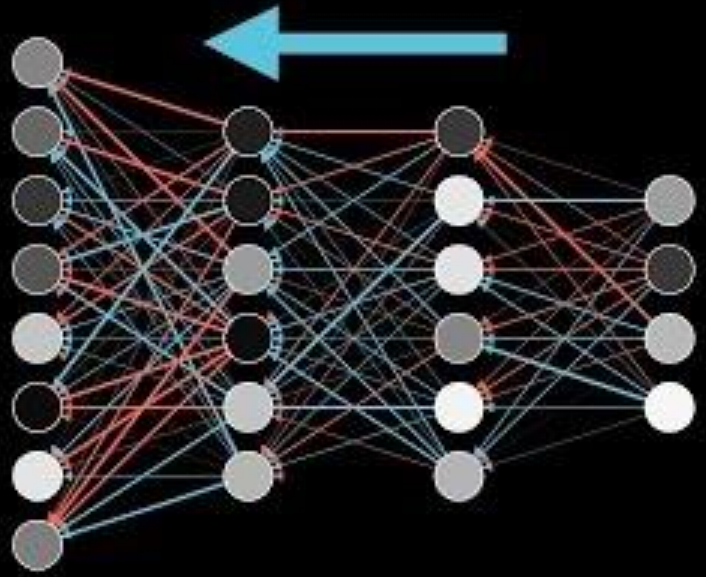
$$\frac{\partial L}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \frac{\partial L}{\partial \phi_0} \\ \frac{\partial L}{\partial \phi_1} \\ \vdots \\ \frac{\partial L}{\partial \phi_N} \end{bmatrix}$$

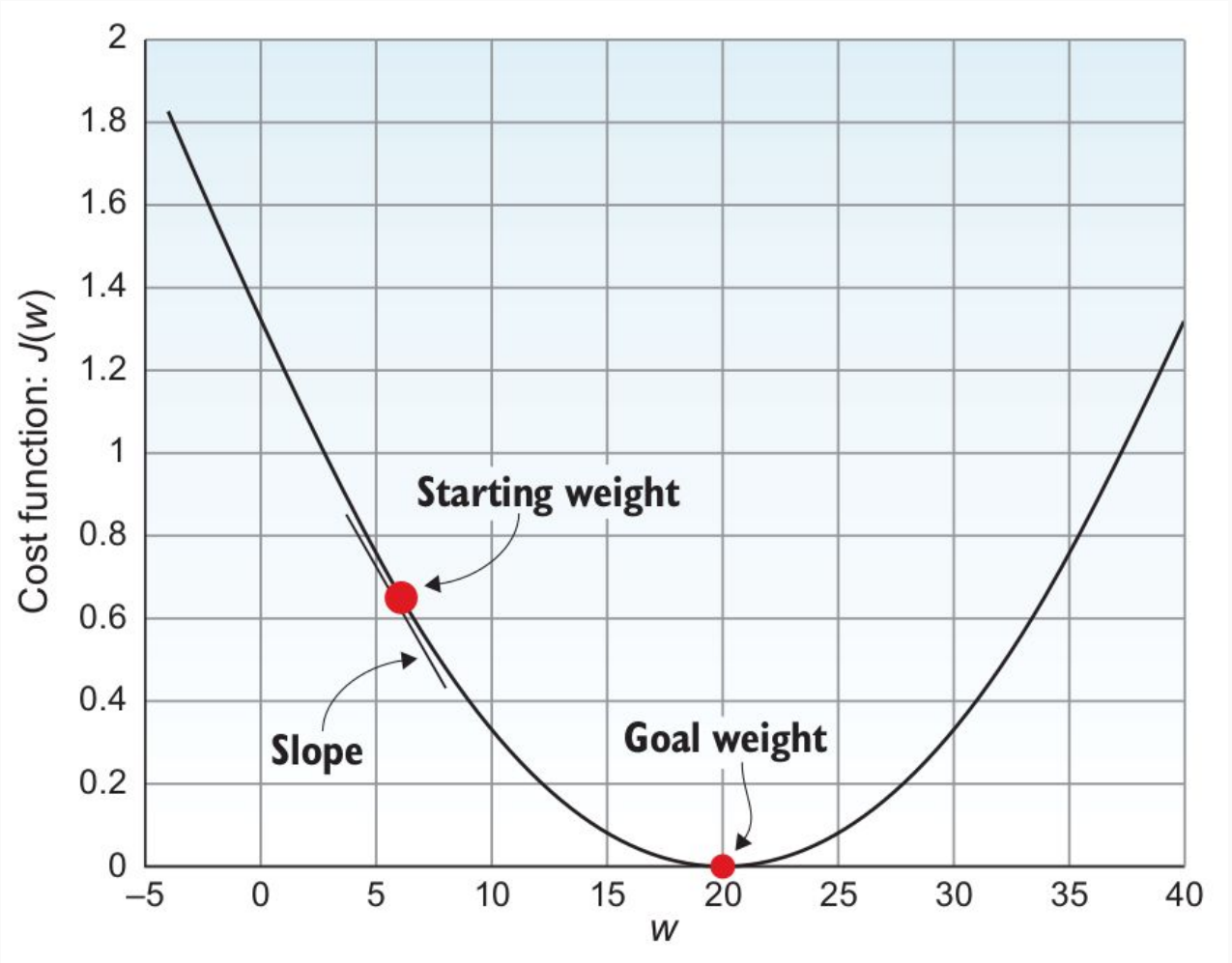
Gradient Descent: Step 1

Update the parameters according to the rule

$$\phi \longleftarrow \phi - \alpha \cdot \frac{\partial L}{\partial \phi}$$

Backpropagation





Elgendy, M., 2020. Deep learning for vision systems. Simon and Schuster.

Partial derivative

Constant Rule: $\frac{d}{dx}c = 0$

Constant Multiple Rule: $\frac{d}{dx}[cf(x)] = cf'(x)$

Difference Rule: $\frac{d}{dx}[f(x) - g(x)] = f'(x) - g'(x)$

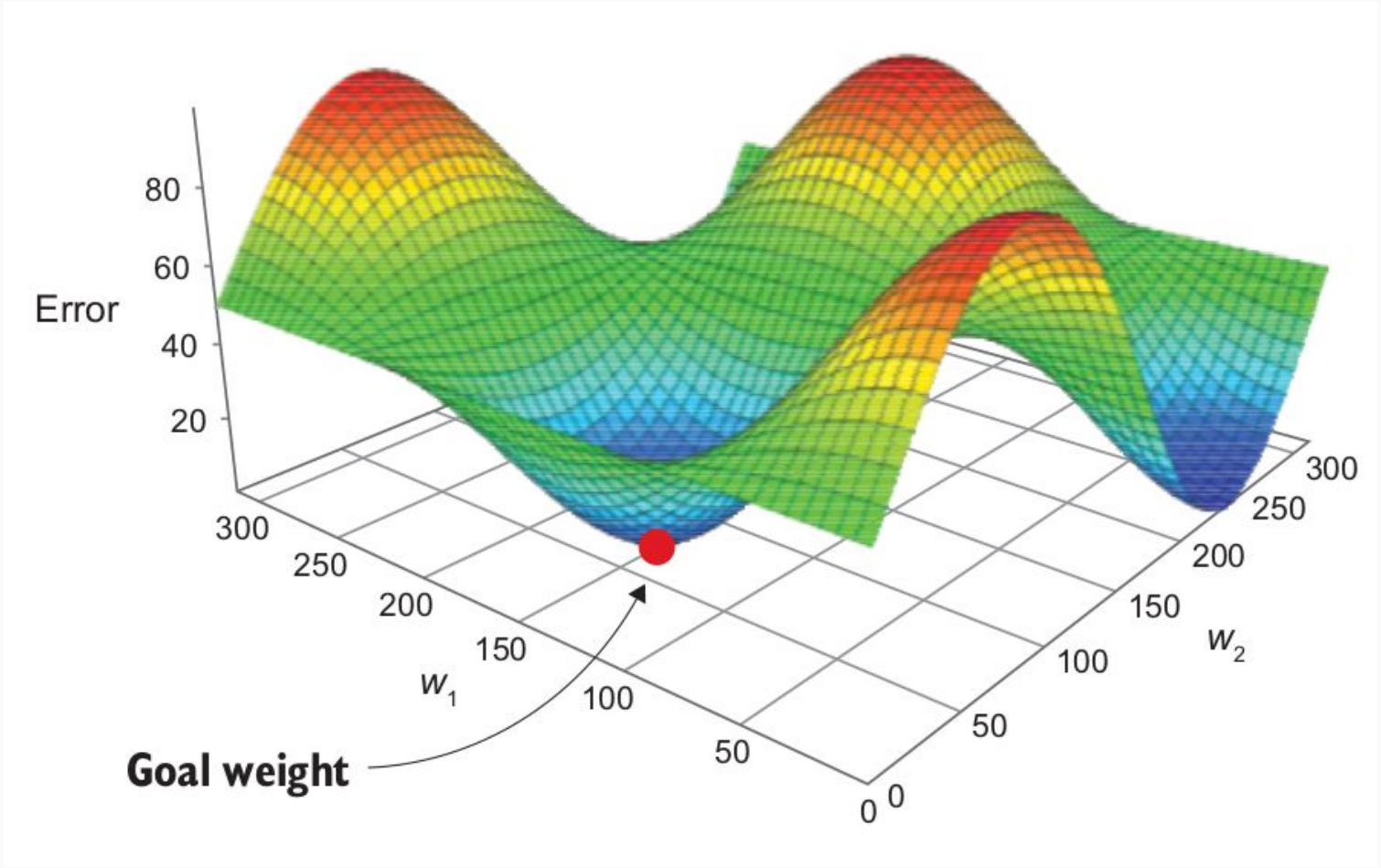
Sum Rule: $\frac{d}{dx}[f(x) + g(x)] = f'(x) + g'(x)$

Product Rule: $\frac{d}{dx}[f(x)g(x)] = f(x)g'(x) + g(x)f'(x)$

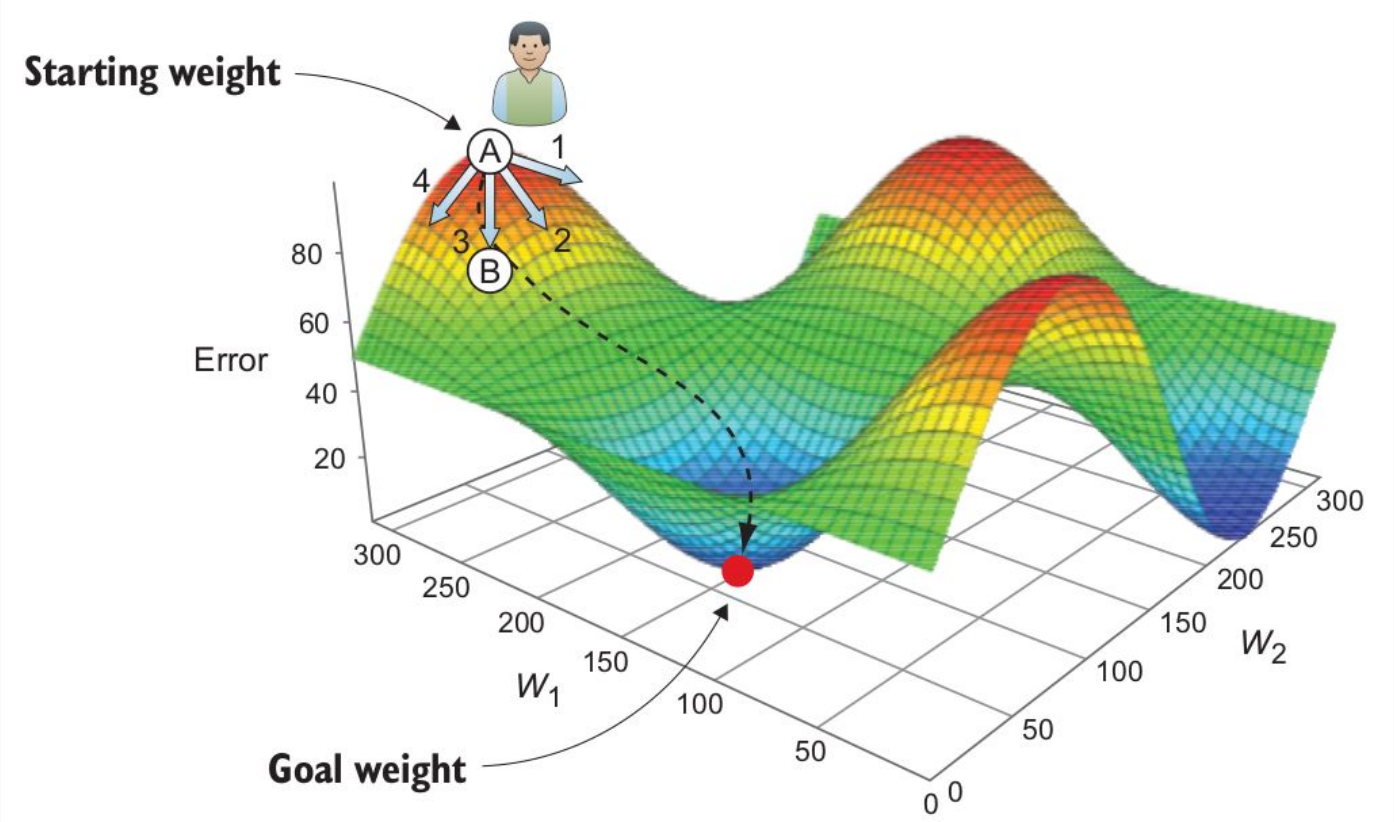
Quotient Rule: $\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = \frac{g(x)f'(x) - f(x)g'(x)}{[g(x)]^2}$

Power Rule: $\frac{d}{dx}x^n = nx^{n-1}$

Chain Rule: $\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$



Elgendy, M., 2020. Deep learning for vision systems. Simon and Schuster.

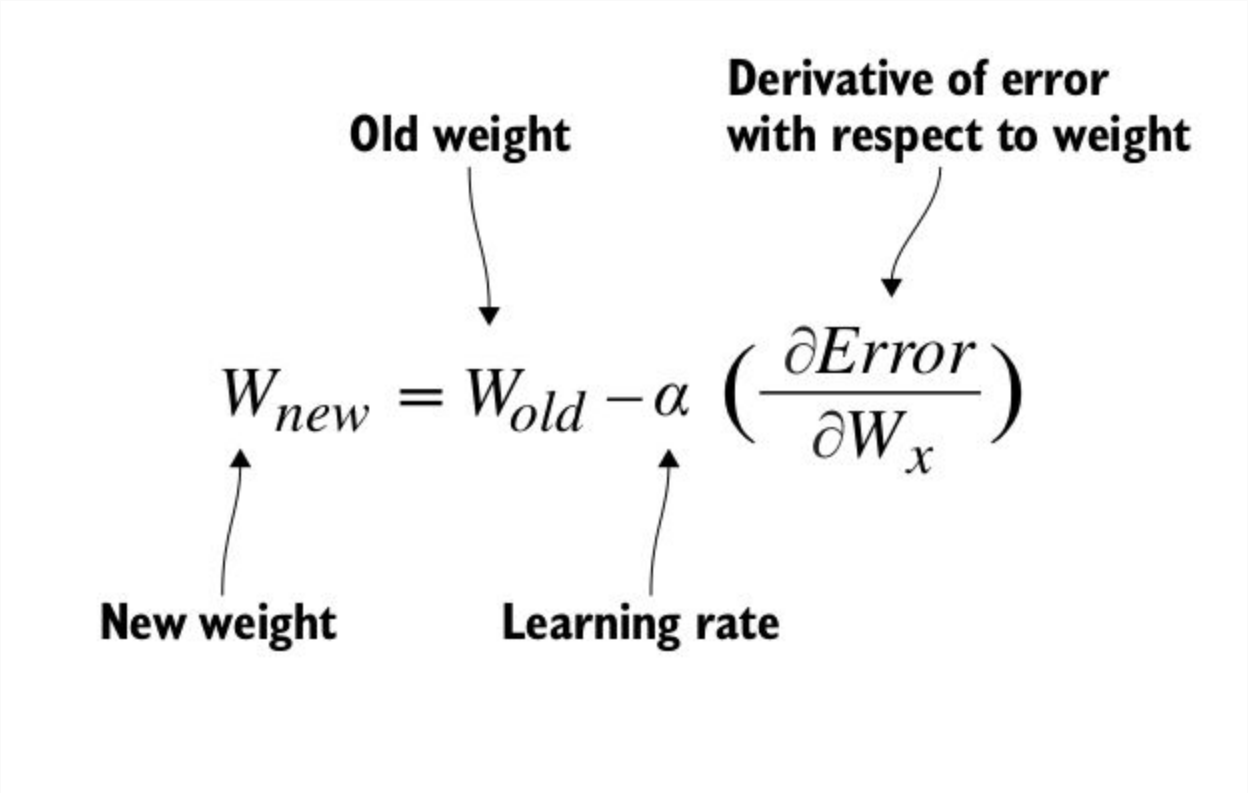


Elgendy, M., 2020. Deep learning for vision systems. Simon and Schuster.

$$\Delta w_i = -\alpha \frac{dE}{dw_i}$$

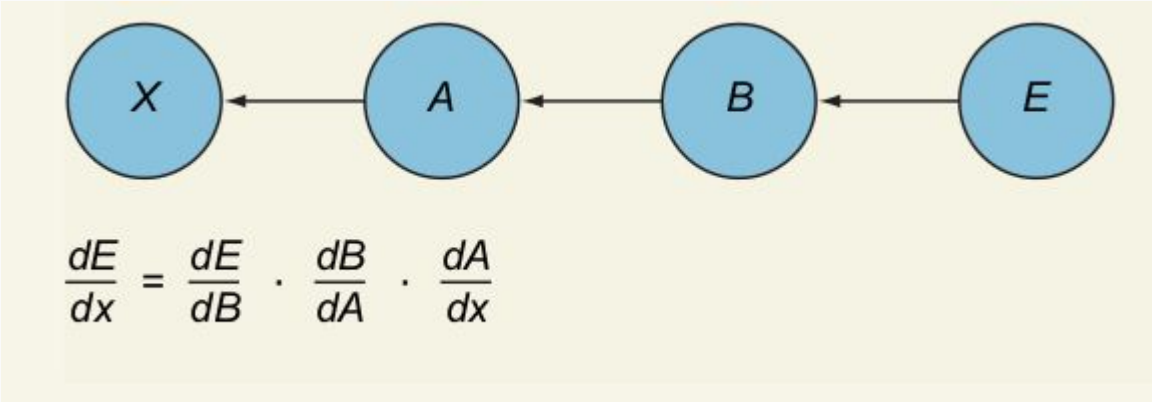
$$w_{\text{next–step}} = w_{\text{current}} + \Delta w$$

Elgendy, M., 2020. Deep learning for vision systems. Simon and Schuster.



Elgendy, M., 2020. Deep learning for vision systems. Simon and Schuster.

Chain Rule: $\frac{d}{dx} f(g(x)) = f'(g(x))g'(x)$



Elgendy, M., 2020. Deep learning for vision systems. Simon and Schuster.

Reading for Next Class

Chapter 7 from UDL