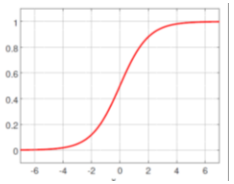
Neural Computation

The Chain Rule II

Summary

- Soft perceptron



- Chain rule

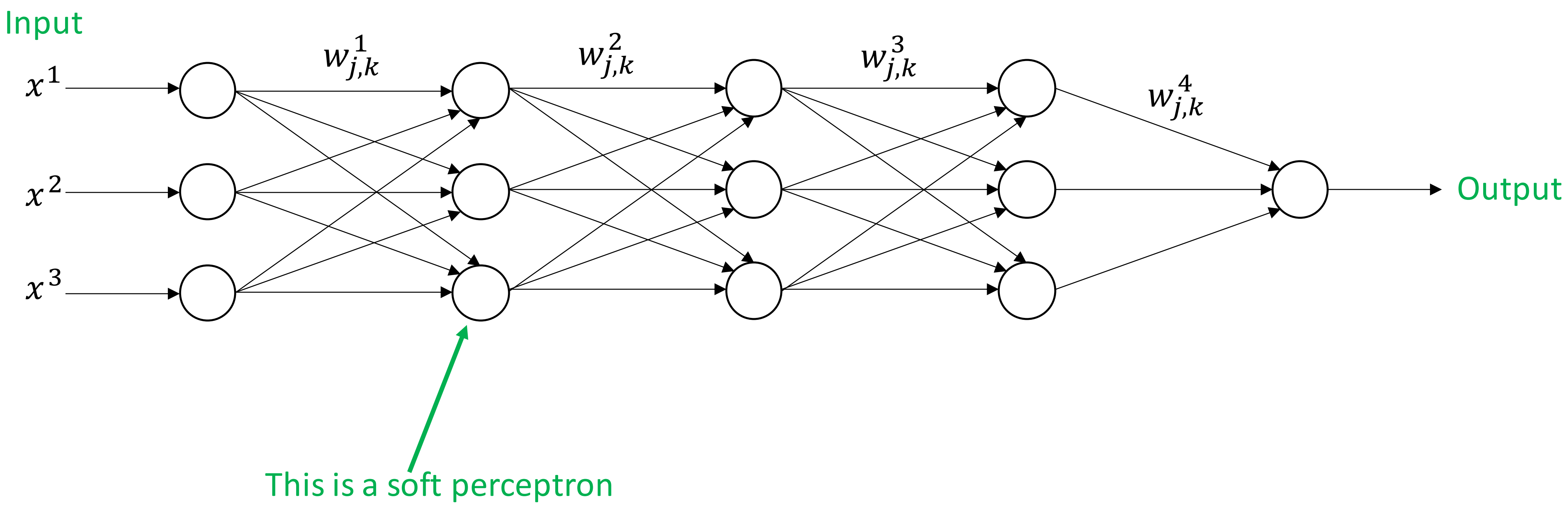
$$\boxed{\frac{\partial}{\partial t} f = \frac{\partial f}{\partial g} \frac{\partial g}{\partial t}}$$

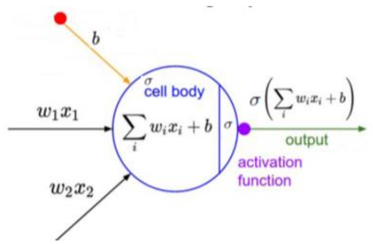
$$t \longrightarrow g \longrightarrow f$$

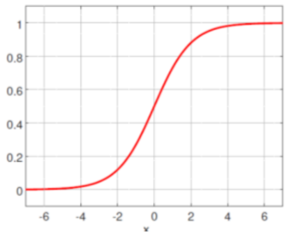
- Gradient descent

$$\frac{\partial}{\partial w_i} C \quad = \left(\sigma\left(\sum_j^m w_j x_j + b \right) - y \right) \quad \sigma'\left(\sum_j^m w_j x_j + b \right) \quad x$$

Multi-Layer-Perceptron (MLP) aka Feed-Forward-Net



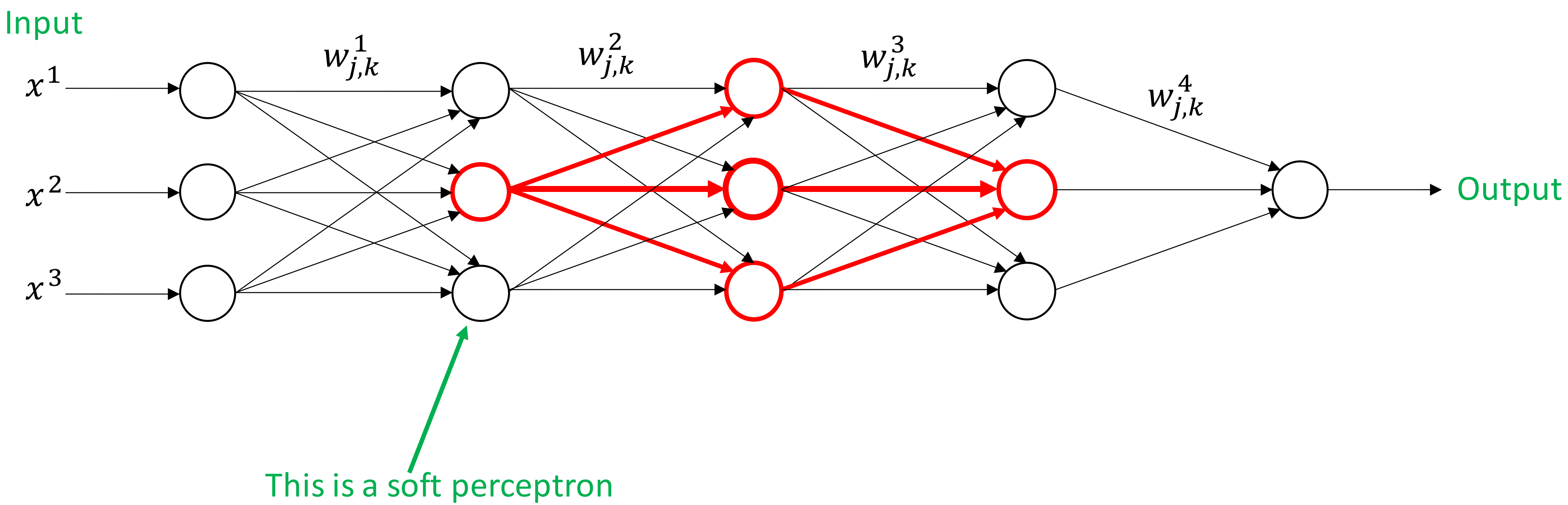


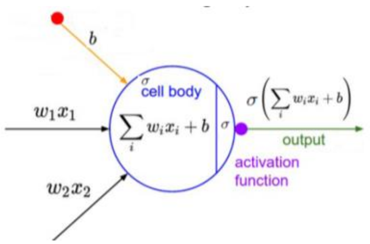


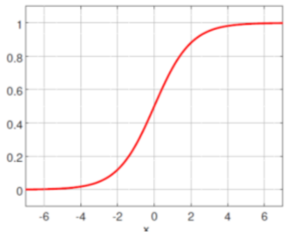
Can we use the chain rule to compute gradients?

- Problem: Multiple dependencies on variable.

Multi-Layer-Perceptron (MLP) aka Feed-Forward-Net







Can we use the chain rule to compute gradients?

- Problem: Multiple dependencies on variable.

Multivariate Chain Rule

- Consider

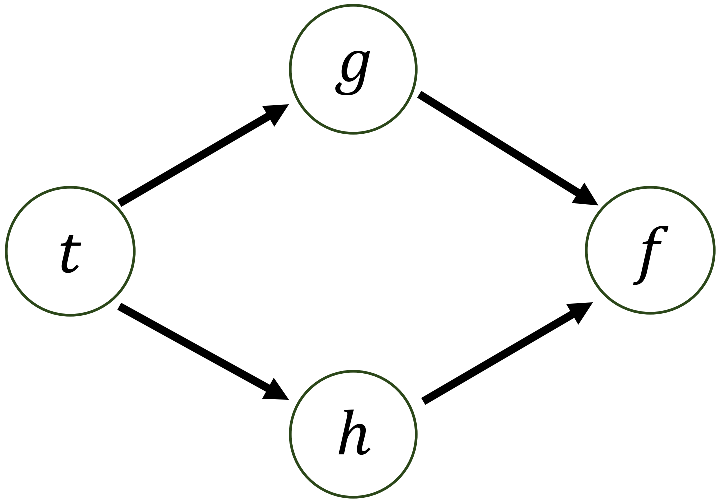
$$f = f(g, h),$$

where

$g = g(t)$ and $h = h(t)$ are functions of t

- We can compute the derivative as

$$\boxed{\frac{\partial}{\partial t} f = \frac{\partial f}{\partial g}\frac{\partial g}{\partial t} + \frac{\partial f}{\partial h}\frac{\partial h}{\partial t}}$$



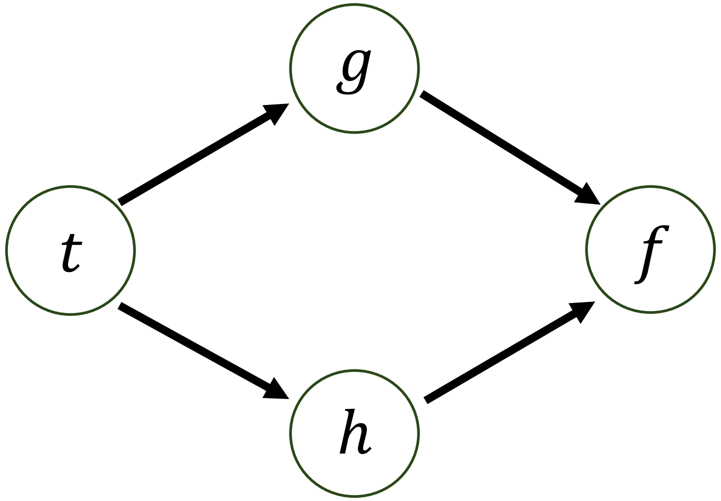
Example

$$\frac{\partial}{\partial t}f = \frac{\partial f}{\partial g}\frac{\partial g}{\partial t} + \frac{\partial f}{\partial h}\frac{\partial h}{\partial t}$$

$$f = f(g,h) = h + e^{gh}$$

$$g = g(t) = \cos t$$

$$h = h(t) = t^2$$



$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial t} + \frac{\partial f}{\partial g}\;\frac{\partial g}{\partial t}$$

$$= (he^{gh})(-\sin t) + (1 + ge^{gh})\,2t$$

Multivariate Chain Rule (general)

- Conder

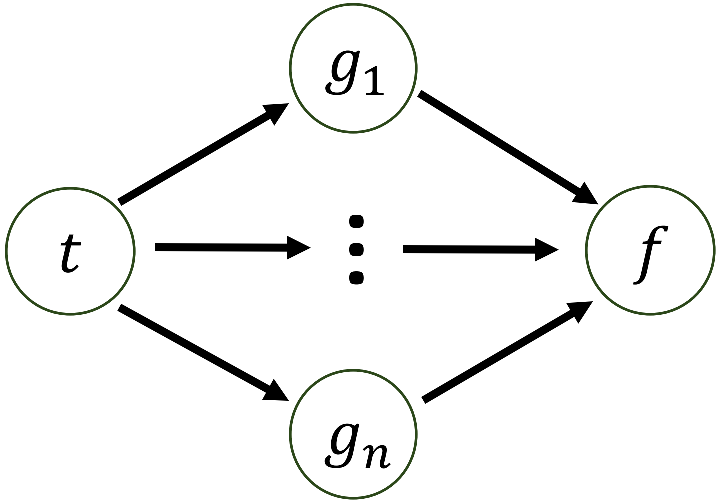
$$f = f(g_1, \dots, g_n),$$

where all

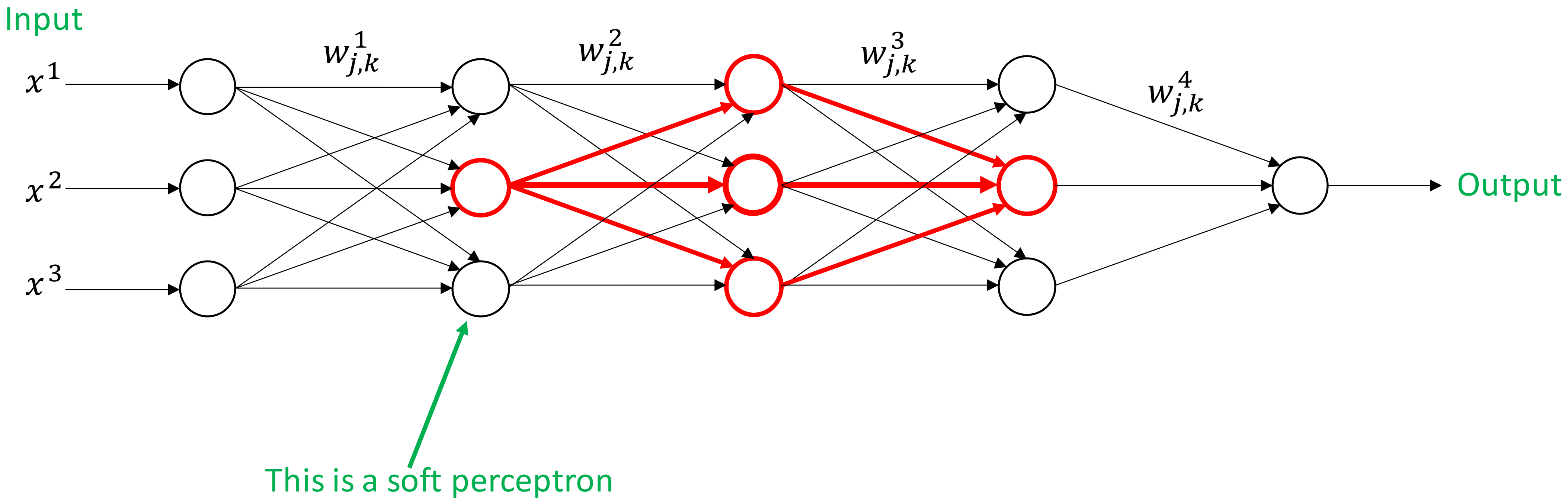
$g_i = g_i(t)$ are functions of t

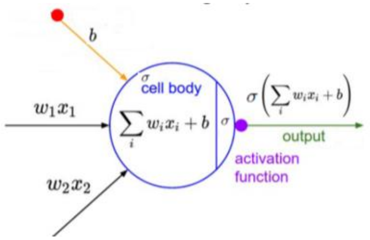
- We can compute the derivative as

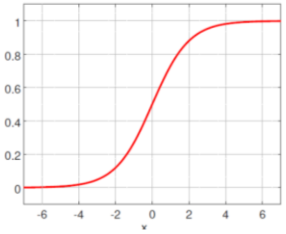
$$\frac{\partial}{\partial t} f = \sum_{i=1}^{n} \frac{\partial f}{\partial g_i} \frac{\partial g_i}{\partial t}$$



Multi-Layer-Perceptron (MLP) aka Feed-Forward-Net







In the next video:

- Compute derivatives for MLP
- Efficient algorithm (error back propagation)