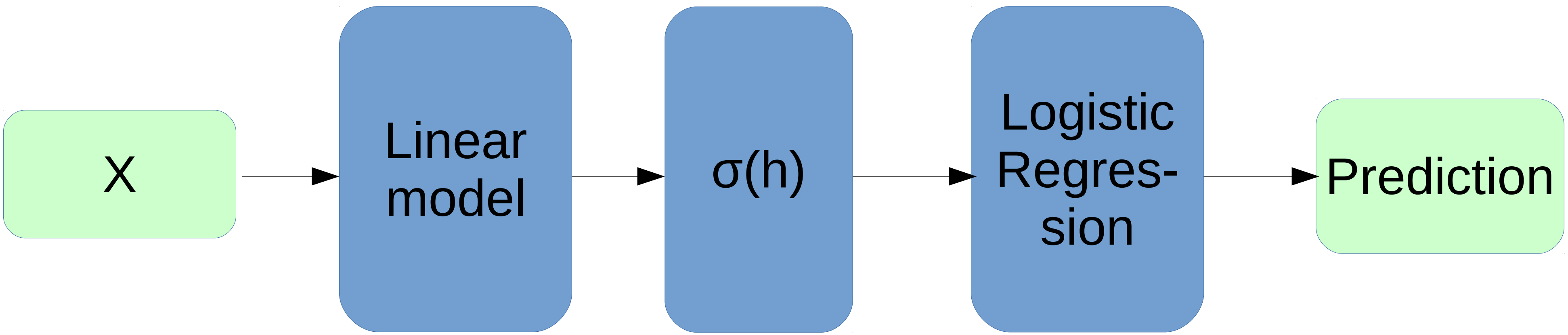
seminar slides
2018.02.14

LAMBDA

British Hedgehog
Preservation Society

TL;DR deep learning

Model:



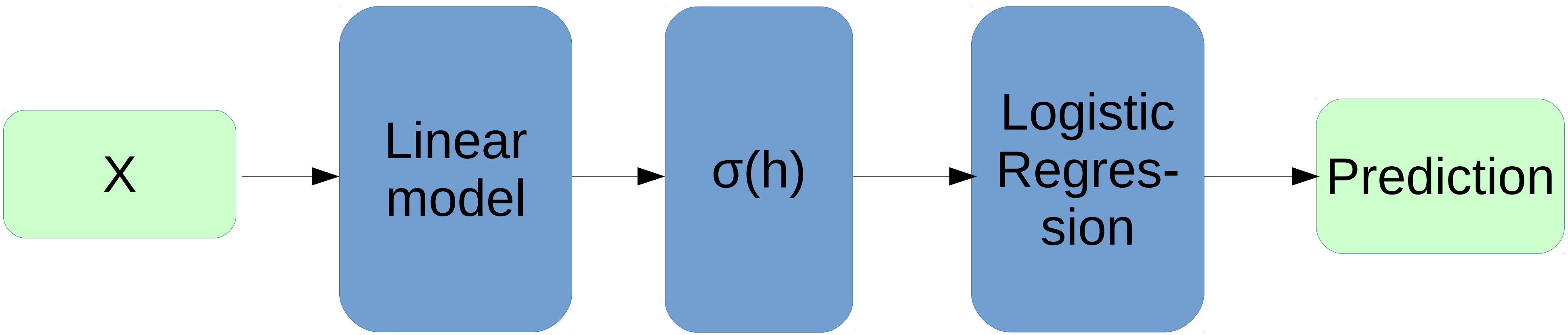
Output:

$$P(y|x)=\sigma\left(\sum_j w_j^o \sigma\left(\sum_i w_{ij}^h x_i + b_j^h\right) + b^o\right)$$

Training: **???**

TL;DR deep learning

Model:

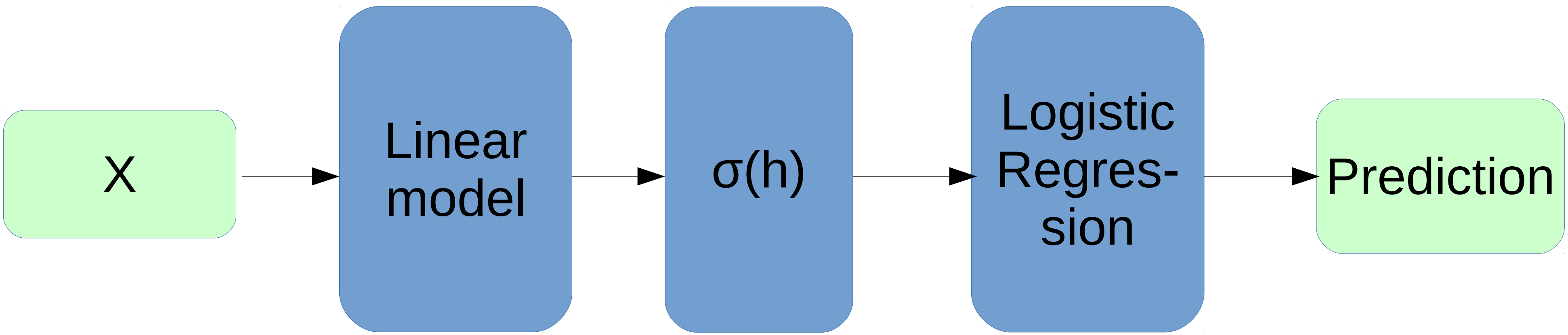


Output:

$$P(y|x)=\sigma\left(\sum_j w_j^o \sigma\left(\sum_i w_{ij}^h x_i + b_j^h\right) + b^o\right)$$

TL;DR deep learning

Model:



Output:

$$P(y|x)=\sigma\left(\sum_j w_j^o \sigma\left(\sum_i w_{ij}^h x_i + b_j^h\right) + b^o\right)$$

Training:

$$w := w - \alpha \frac{\partial \underset{x_i,y_i}{E} - \log P_w(y_i|x_i)}{\partial w}$$

Backpropagation

TL;DR: backprop = chain rule*

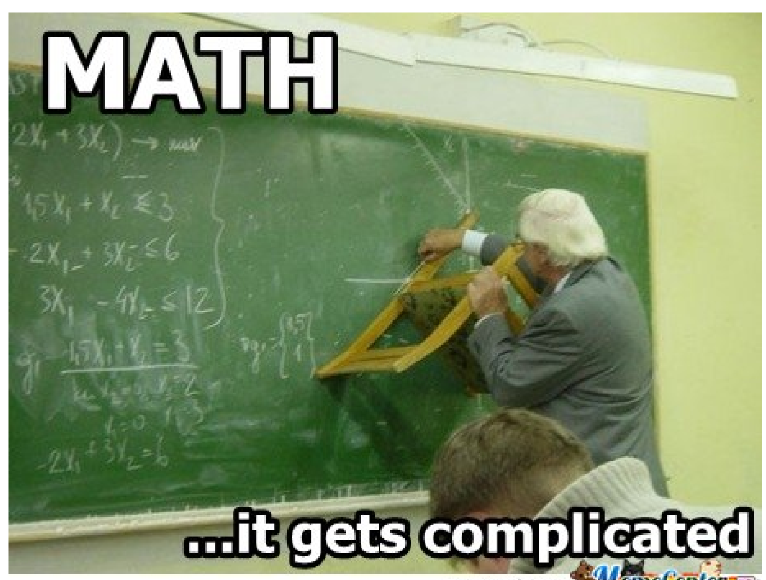
$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g(x)} \cdot \frac{\partial g(x)}{\partial x}$$

Backpropagation

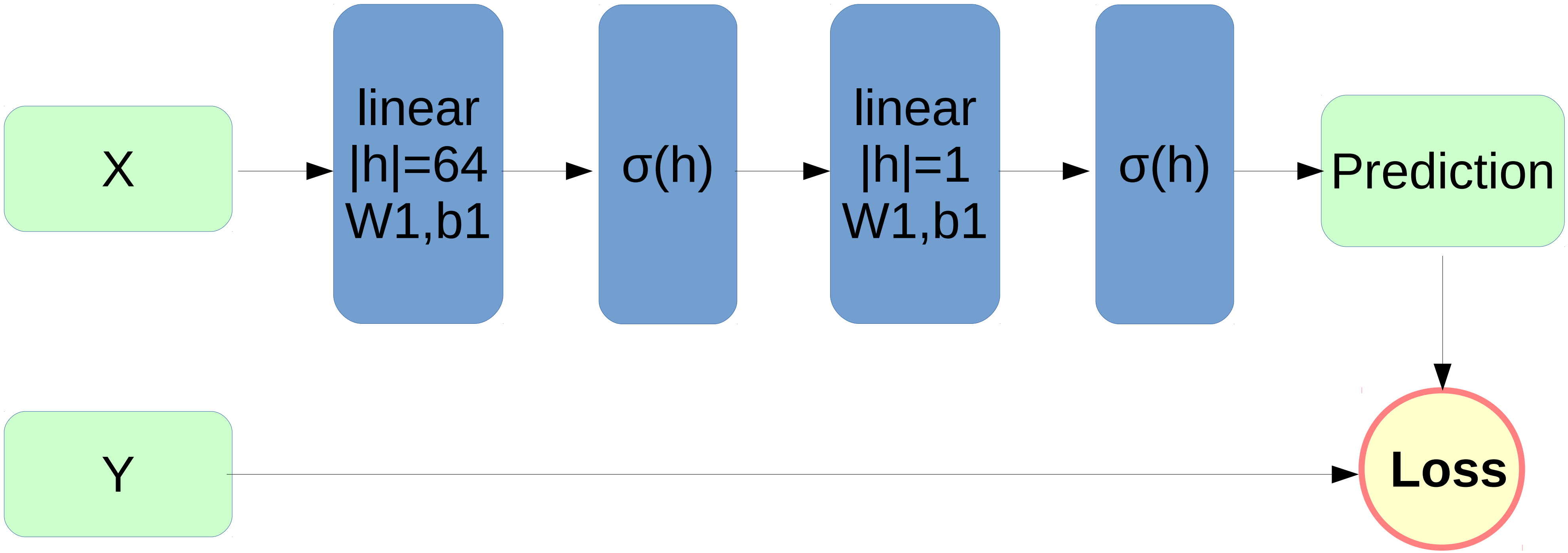
TL;DR: backprop = chain rule*

$$\frac{\partial f\left(g\left(x\right)\right)}{\partial x} = \frac{\partial f\left(g\left(x\right)\right)}{\partial g\left(x\right)} \cdot \frac{\partial g\left(x\right)}{\partial x}$$

* g and x can be vectors/vectors/tensors

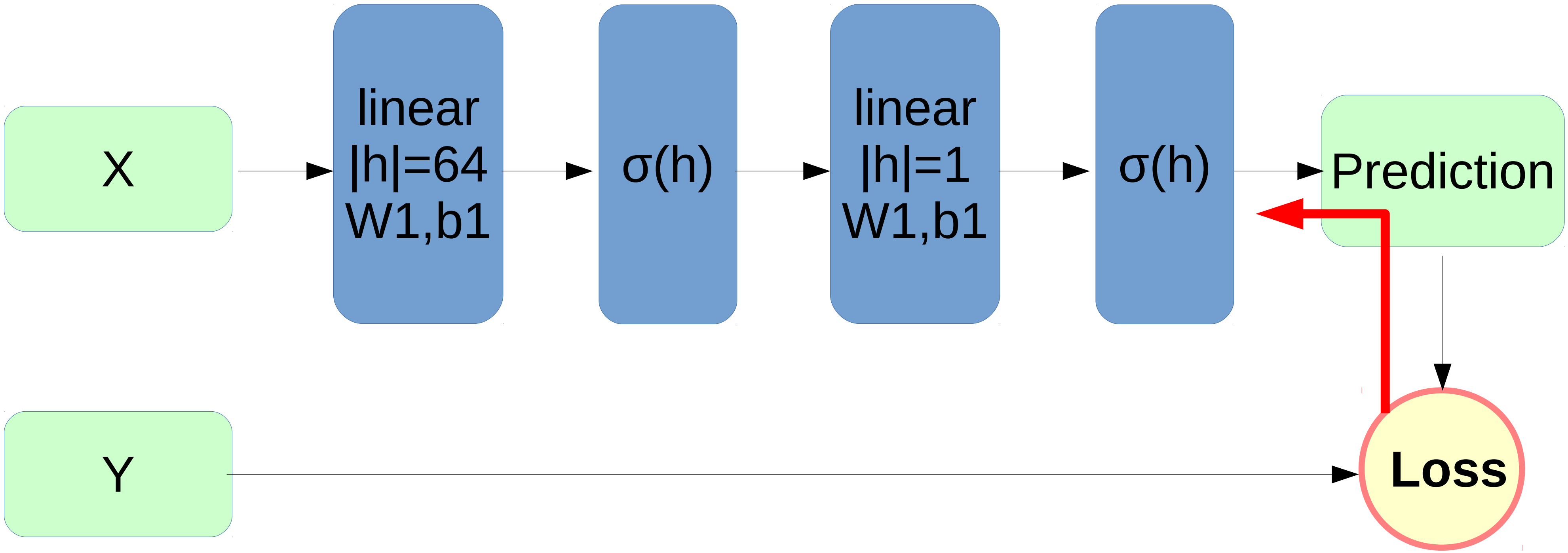


Backpropagation



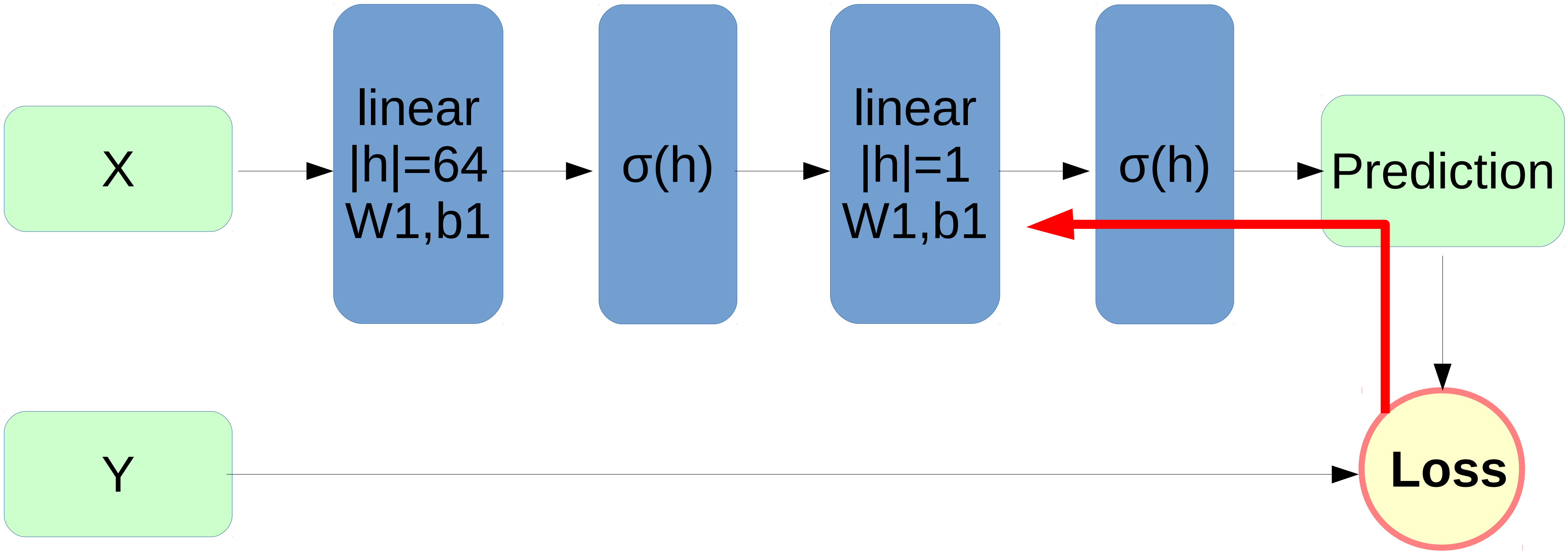
$$\frac{\partial L(\sigma(linear_{w2,b2}(\sigma(linear_{w1,b1}(x)))))}{\partial w1} = \ldots$$

Backpropagation



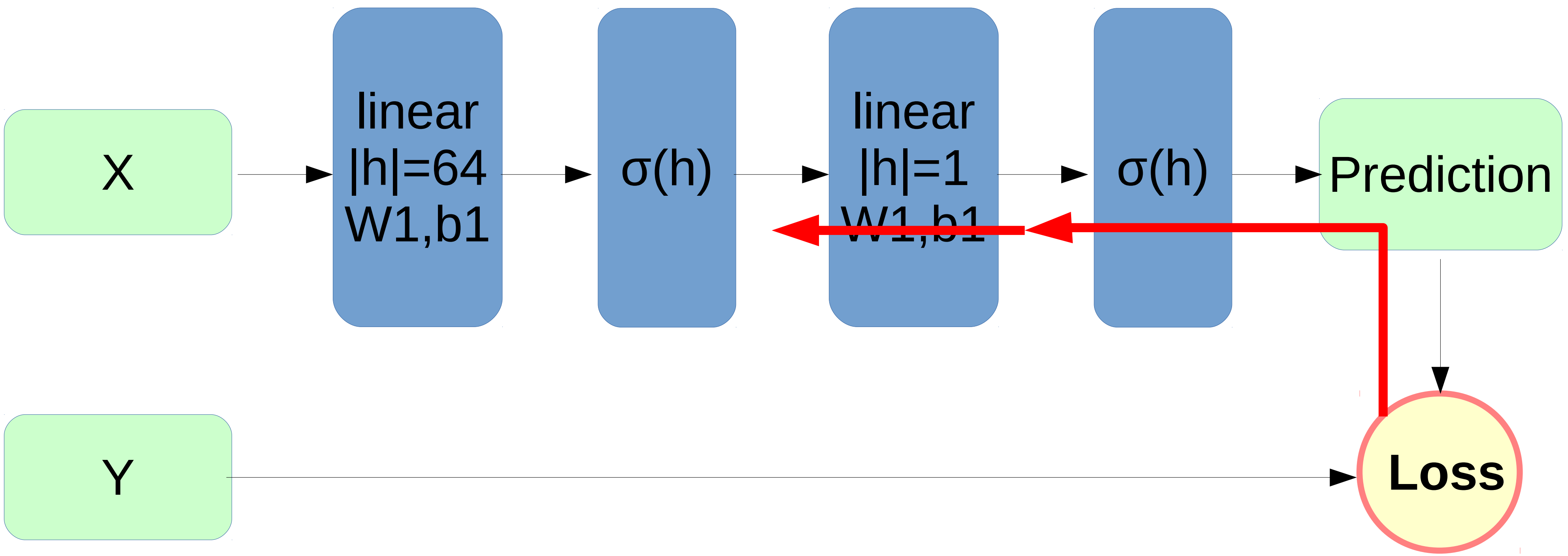
$$\frac{\partial L}{\partial w1} = \frac{\partial L}{\partial \sigma} \cdot$$

Backpropagation



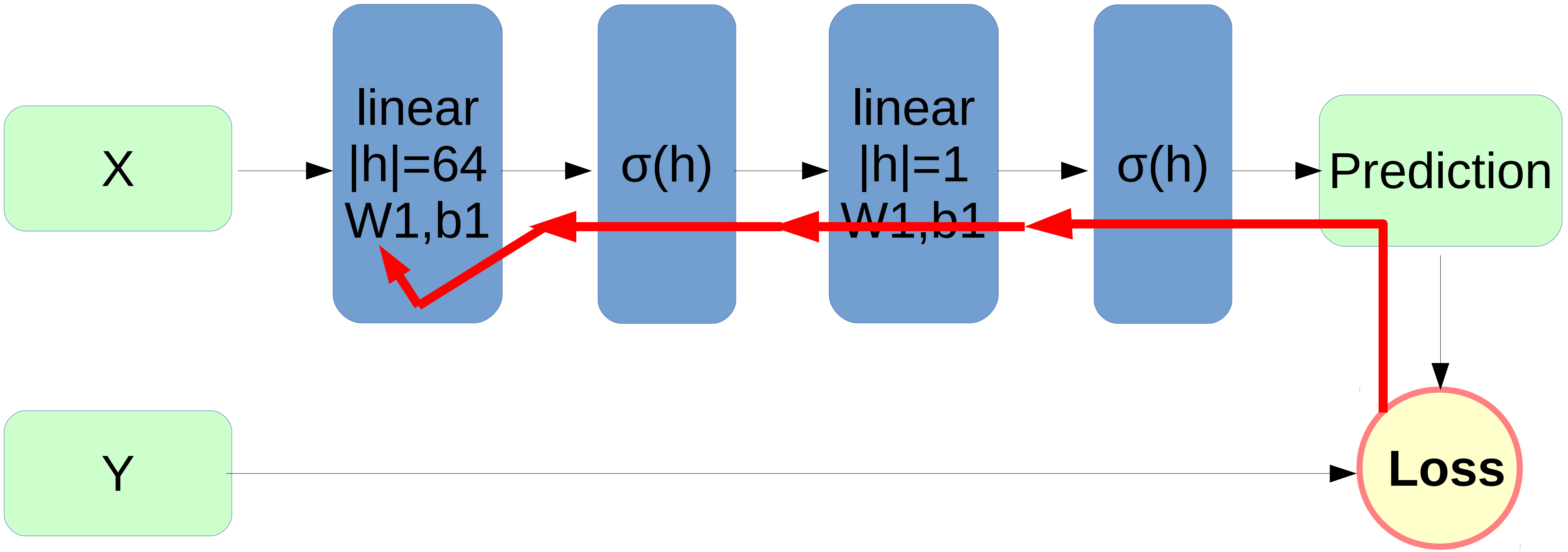
$$\frac{\partial L}{\partial w1} = \frac{\partial L}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial\, linear_{w2,b2}} \cdot$$

Backpropagation



$$\frac{\partial L}{\partial w1} = \frac{\partial L}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial\, linear_{w2,b2}} \cdot \frac{\partial\, linear_{w2,b2}}{\partial \sigma} \cdot$$

Backpropagation



$$\frac{\partial L}{\partial w1} = \frac{\partial L}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial\, linear_{w2,b2}} \cdot \frac{\partial\, linear_{w2,b2}}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial\, linear_{w1,b1}} \cdot \frac{\partial\, linear_{w1,b1}}{\partial w1}$$

Matrix derivatives

Let's compute:

$$\frac{\partial L(X \times W + b)}{\partial X} = \frac{\partial L(X \times W + b)}{\partial [X \times W + b]} \times \boxed{\textbf{What?}}$$

Variable shapes:

X [batch size, features]

W [features, outputs]

b [outputs]

$\frac{\partial L(X \times W + b)}{\partial X}$ [batch size, features]

$\frac{\partial L(X \times W + b)}{X \times W + b}$ [batch size, outputs]

Matrix derivatives

Let's compute:

$$\frac{\partial L(X \times W + b)}{\partial X} = \frac{\partial L(X \times W + b)}{\partial [X \times W + b]} \times W^T$$

Variable shapes:

X	W	b
[batch size, features]	[features, outputs]	[outputs]

$\frac{\partial L(X \times W + b)}{\partial X}$	$\frac{\partial L(X \times W + b)}{X \times W + b}$
[batch size, features]	[batch size, outputs]

Matrix derivatives (words)

$$\text{Gradient of } \sum_i \log p(y_i | x_i, w) = \sum_i \text{gradient} \log p(y_i | x_i, w)$$

$$\text{linear over X}: \frac{\partial L}{\partial [X \times W + b]} \times W^T$$

$$\text{linear over W}: \frac{1}{\|X\|} \cdot X^T \times \frac{\partial L}{\partial [X \times W + b]}$$

$$\text{sigmoid}: \frac{\partial L}{\partial \sigma(x)} \cdot [\sigma(x) \cdot (1 - \sigma(x))]$$

Works for any kind of x
(scalar, vector, matrix, tensor)

Matrix derivatives (formulae)

$$\frac{\partial \sum_i \log p(y_i|x_i,w)}{\partial w} = \frac{\sum_i \partial \log p(y_i|x_i,w)}{\partial w}$$

$$\frac{\partial L(X \times W + b)}{\partial X} = \frac{\partial L}{\partial [X \times W + b]} \times W^T$$

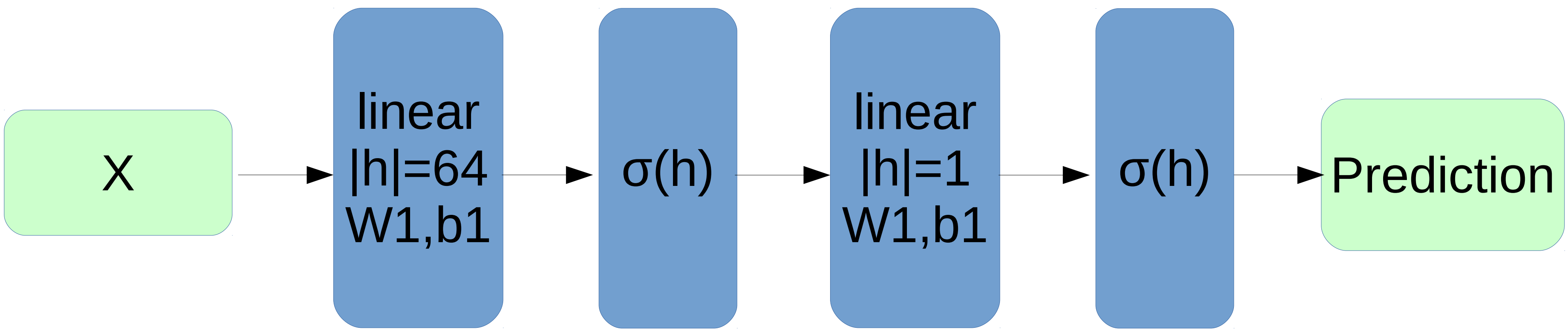
$$\frac{\partial L(X \times W + b)}{\partial W} = X^T \times \frac{\partial L}{\partial [X \times W + b]}$$

$$\frac{\partial L(\sigma(x))}{\partial x} = \frac{\partial L}{\partial \sigma(x)} \cdot [\sigma(x) \cdot (1 - \sigma(x))]$$

Works for any kind of x
(scalar, vector, matrix, tensor)

Back to neural networks

Model:



Training: