COMP90051

Workshop Week 05

About the Workshops

- 7 sessions in total
  - Tue 12:00-13:00 AH211
  - Tue 12:00-13:00 AH108 *
  - Tue 13:00-14:00 AH210
  - Tue 16:15-17:15 AH109
  - Tue 17:15-18:15 AH236 *
  - Tue 18:15-19:15 AH236 *
  - Fri 14:15-15:15 AH211

About the Workshops

- Homepage
  - https://trevorcohn.github.io/comp90051-2017/workshops

- Solutions will be released on next Friday (a week later).

Syllabus

1	Introduction; Probability theory	Probabilistic models; Parameter fitting	
2	Linear regression; Intro to regularization	Logistic regression; Basis expansion	
3	Optimization; Regularization	Perceptron	
4	Backpropagation	CNNs; Auto-encoders	←
5	Hard-margin SVMs	Soft-margin SVMs	
6	Additional topics	Kernel methods	
7	Unsupervised learning	Unsupervised learning	
8	Dimensionality reduction; Principal component analysis	Multidimensional scaling; Spectral clustering	
9	Bayesian fundamentals	Bayesian inference with conjugate priors	
10	PGMs, fundamentals	Conditional independence	
11	PGMs, inference	Belief propagation	
12	Statistical inference; Apps	Subject review	

Outline

- Review the lecture, background knowledge, etc.
  - Gradient descent & stochastic gradient descent (SGD)
  - Gradient and backpropagation
    - Logistic regression
    - Neural networks with one hidden layer
- Notebook tasks
  - Task 1: Multi-layer perceptron, SGD

Outline

- Review the lecture, background knowledge, etc.
  - Gradient descent & stochastic gradient descent (SGD)
  - Gradient and backpropagation
    - Logistic regression
    - Neural networks with one hidden layer
- Notebook tasks
  - Task 1: Multi-layer perceptron, SGD

Gradient descent & Stochastic GD (SGD)

- To minimize an objective function $obj(w)$
- Usually, obj is the average loss plus a regularization term

$$\min_{\boldsymbol{w}} obj(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} L(f(x_i;\boldsymbol{w}), y_i) + \lambda R(\boldsymbol{w})$$

Gradient descent

- To minimize an objective function $obj(w)$
- Usually, obj is the average loss plus a regularization term

$$\min_{\boldsymbol{w}} obj(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} L(f(x_i; \boldsymbol{w}), y_i) + \lambda R(\boldsymbol{w})$$

- Loop until $\boldsymbol{w}$ doesn't change

$$\boldsymbol{w} = \boldsymbol{w} - \eta \frac{\partial obj(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

Gradient descent

- To minimize an objective function $obj(w)$
- Usually, obj is the average loss plus a regularization term

$$\min_{\boldsymbol{w}} obj(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} L(f(x_i;\boldsymbol{w}), y_i) + \lambda R(\boldsymbol{w})$$

- Loop until $\boldsymbol{w}$ doesn't change

$$\mathrm{grad}_{\boldsymbol{w}} = \frac{\partial obj(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial L(f(x_i;\boldsymbol{w}), y_i)}{\partial \boldsymbol{w}} + \lambda \frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

$$\boldsymbol{w} = \boldsymbol{w} - \eta\,\mathrm{grad}_{\boldsymbol{w}}$$

Stochastic gradient descent (SGD)

- To minimize an objective function $obj(w)$
- Usually, obj is the average loss plus a regularization term

$$\min_{\boldsymbol{w}} obj(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} L(f(x_i;\boldsymbol{w}), y_i) + \lambda R(\boldsymbol{w})$$

- Loop until $\boldsymbol{w}$ doesn't change
  - Sample i from $\{1,2,\dots,N\}$ or For $i = 1,2,\dots N$

$$\mathrm{grad}_{\boldsymbol{w}} = \frac{\partial L(f(x_i;\boldsymbol{w}), y_i)}{\partial \boldsymbol{w}} + \lambda \frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

$$\boldsymbol{w} = \boldsymbol{w} - \eta\,\mathrm{grad}_{\boldsymbol{w}}$$

Stochastic gradient descent (SGD)

- Note: SGD has other variants
- Loop until $\boldsymbol{w}$ doesn't change ← online learning
  - Sample i from $\{1,2,\ldots,N\}$ or For $i = 1,2,\ldots N$

$$\text{grad}_{\boldsymbol{w}} = \frac{\partial L(f(x_i;\boldsymbol{w}), y_i)}{\partial \boldsymbol{w}} + \lambda \frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

$$\boldsymbol{w} = \boldsymbol{w} - \eta\,\text{grad}_{\boldsymbol{w}}$$

- Loop until $\boldsymbol{w}$ doesn't change ← mini-batch
  - Sample a subset S from $\{1,2,\ldots,N\}$

$$\text{grad}_{\boldsymbol{w}} = \frac{1}{|S|}\sum_{i\in S} \frac{\partial L(f(x_i;\boldsymbol{w}), y_i)}{\partial \boldsymbol{w}} + \lambda \frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

$$\boldsymbol{w} = \boldsymbol{w} - \eta\,\text{grad}_{\boldsymbol{w}}$$

Gradient descent & Stochastic GD (SGD)

- Gradient descent
- Loop until $\boldsymbol{w}$ doesn't change

$$\text{grad}_{\boldsymbol{w}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial L(f(x_i;\boldsymbol{w}), y_i)}{\partial \boldsymbol{w}} + \lambda\frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

$$\boldsymbol{w} = \boldsymbol{w} - \eta\,\text{grad}_{\boldsymbol{w}}$$

- Stochastic GD (SGD)
- Loop until $\boldsymbol{w}$ doesn't change
  - Sample i from $\{1,2,\dots,N\}$ or For $i = 1,2,\dots N$

$$\text{grad}_{\boldsymbol{w}} = \frac{\partial L(f(x_i;\boldsymbol{w}), y_i)}{\partial \boldsymbol{w}} + \lambda\frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

$$\boldsymbol{w} = \boldsymbol{w} - \eta\,\text{grad}_{\boldsymbol{w}}$$

Outline

- Review the lecture, background knowledge, etc.
  - Gradient descent & stochastic gradient descent (SGD)
  - Gradient and backpropagation
    - Logistic regression
    - Neural networks with one hidden layer
- Notebook tasks
  - Task 1: Multi-layer perceptron, SGD

Formulas you need to know

- Logistic function

$$y = \sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\frac{dy}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = y(1 - y)$$

- Hyperbolic tangent function

$$y = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\frac{dy}{dx} = \frac{4}{(e^x + e^{-x})^2} = 1 - y^2$$

Formulas you need to know

- Log-loss

$$L(\boldsymbol{x}_i, y_i; \boldsymbol{W}) = -\log p(y = y_i | \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{W})$$

- Log-loss for binary classification

$$\hat{y}_i = p(y = 1 | \boldsymbol{x} = \boldsymbol{x}_i; \boldsymbol{W})$$

$$L(\boldsymbol{x}_i, y_i; \boldsymbol{W}) = -(1 - y_i)\log(1 - \hat{y}_i) - y_i \log \hat{y}_i$$

$$L(\boldsymbol{x}_i, y_i; \boldsymbol{W}) = \begin{cases} -\log(1 - \hat{y}_i) & y_i = 0 \\ -\log \hat{y}_i & y_i = 1 \end{cases}$$

Formulas you need to know

- Logistic regression (2-D points, 2 classes)
- $\boldsymbol{x} = [x_1 \quad x_2] \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$
- Decision function

$$s = f(\boldsymbol{x}; \boldsymbol{w}, b) = x_1 w_1 + x_2 w_2 + b$$

- Probability output

$$\hat{y} = \sigma(s) = \frac{1}{1 + e^{-s}}$$

- Log-loss

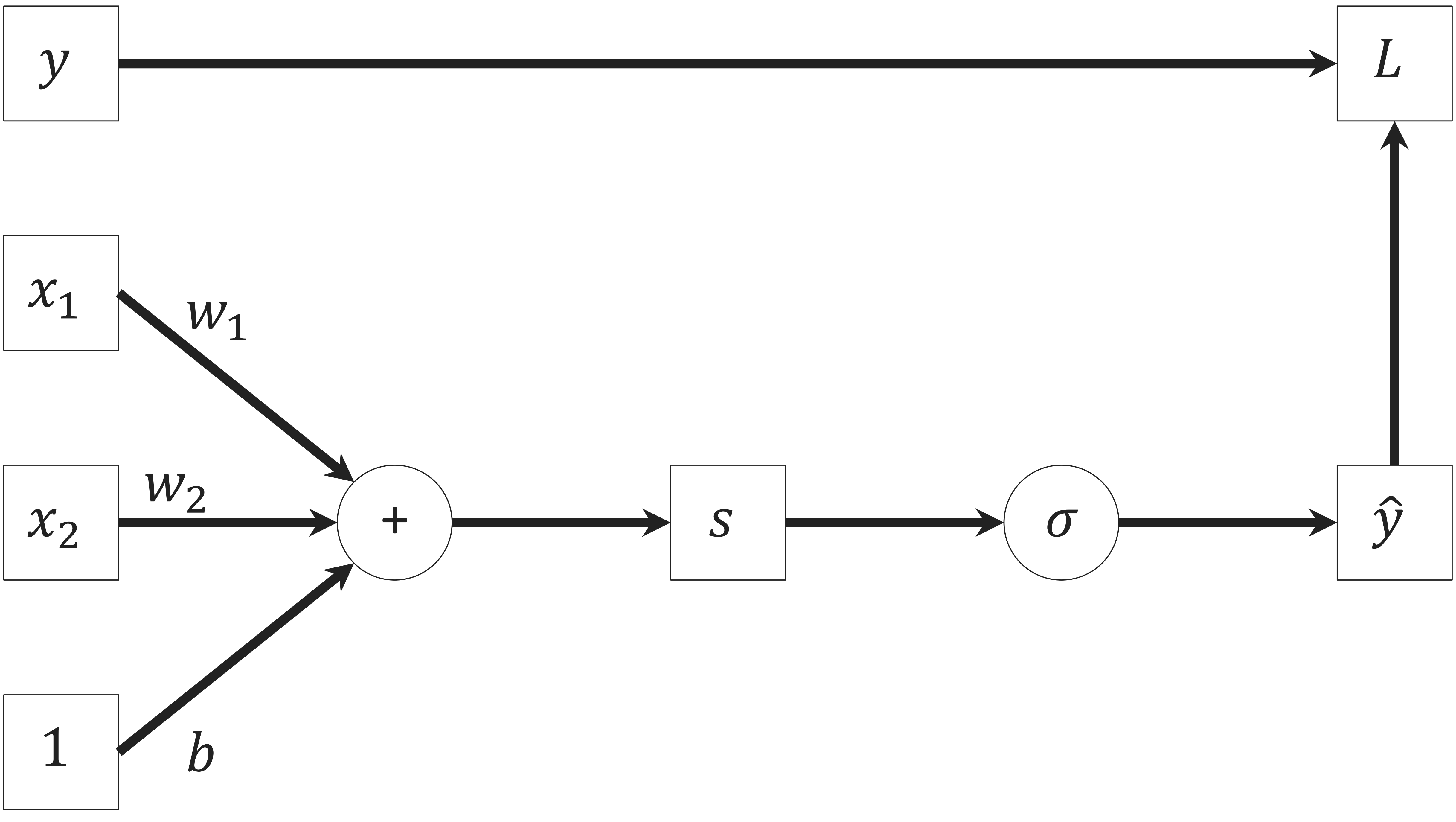
$$L(\boldsymbol{x}, y; \boldsymbol{w}, b) = -(1 - y)\log(1 - \hat{y}) - y \log \hat{y}$$

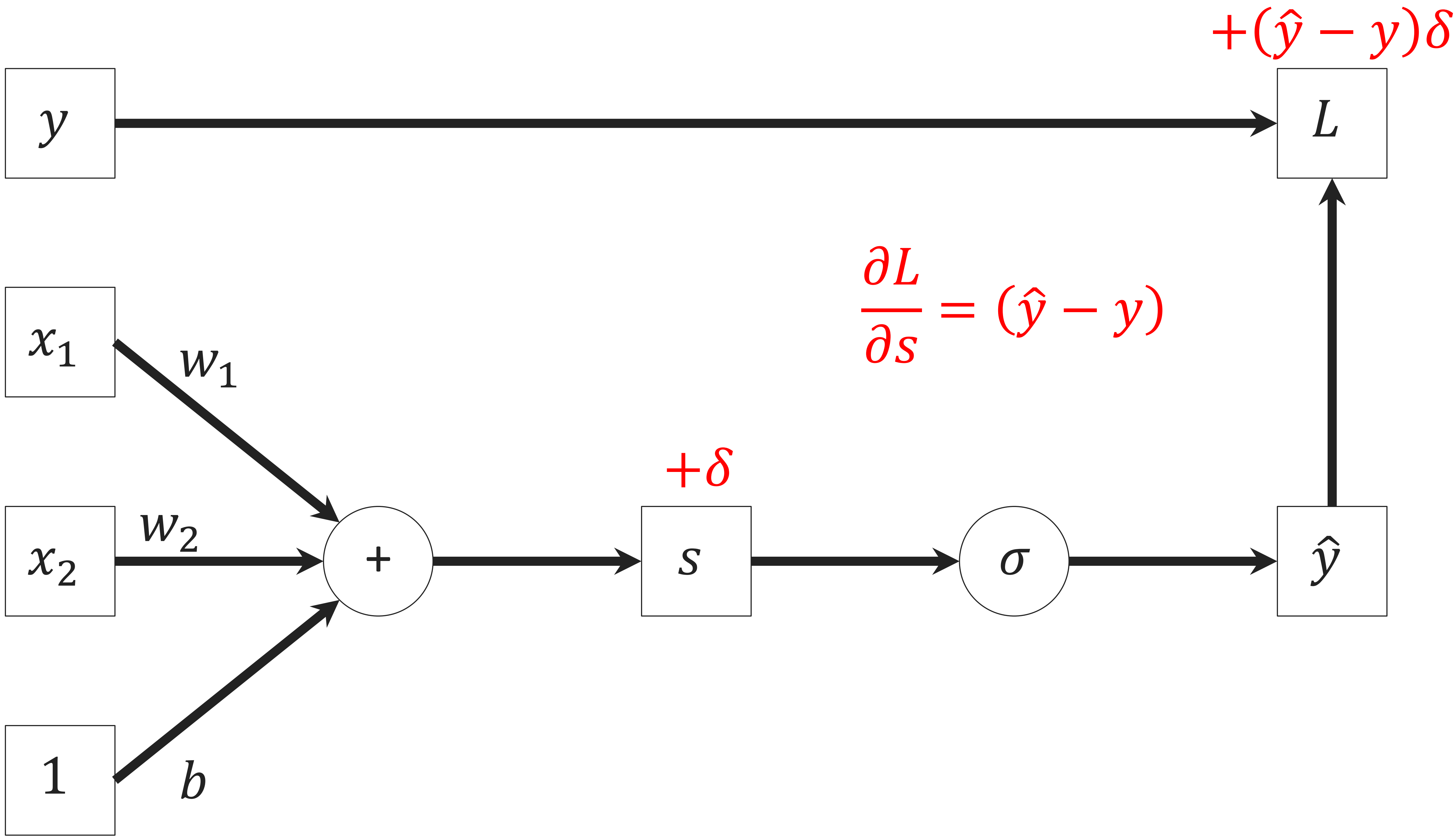
$$\frac{\partial L}{\partial s} = \frac{1}{1 + e^{-s}} - y = \hat{y} - y$$

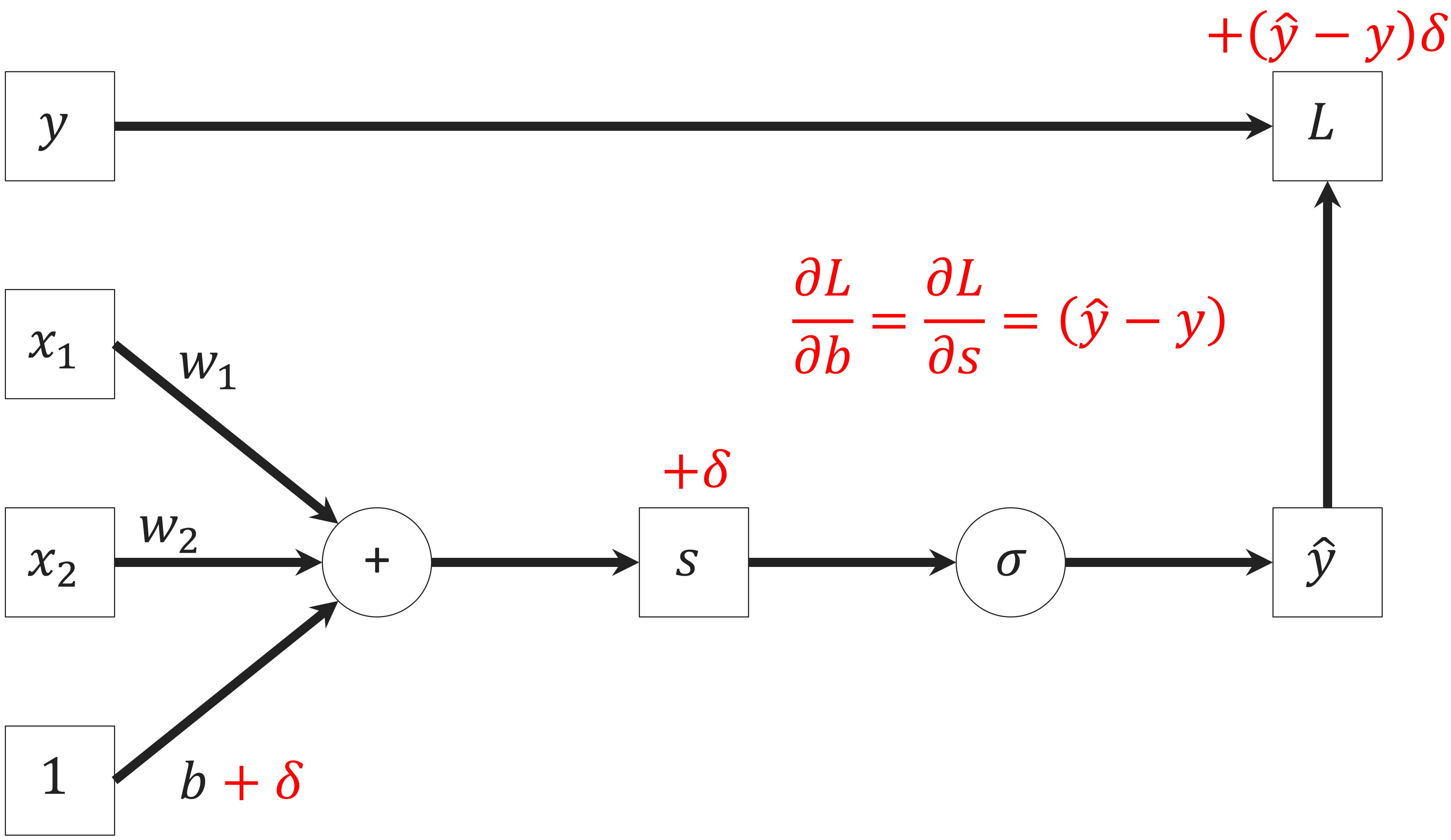
Outline

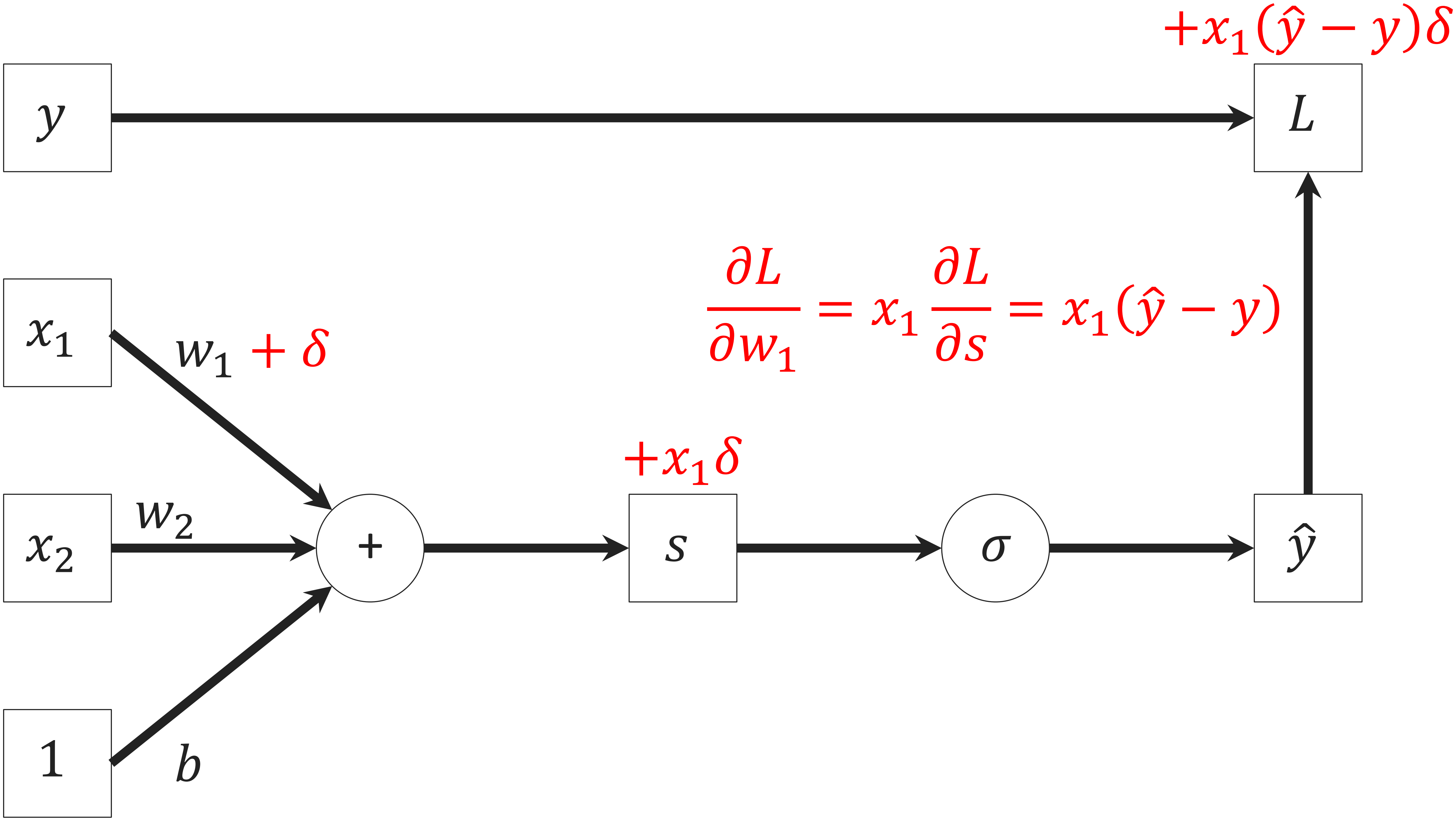
- Review the lecture, background knowledge, etc.
  - Gradient descent & stochastic gradient descent (SGD)
  - Gradient and backpropagation
    - Logistic regression
    - Neural networks with one hidden layer
- Notebook tasks
  - Task 1: Multi-layer perceptron, SGD

Forward pass

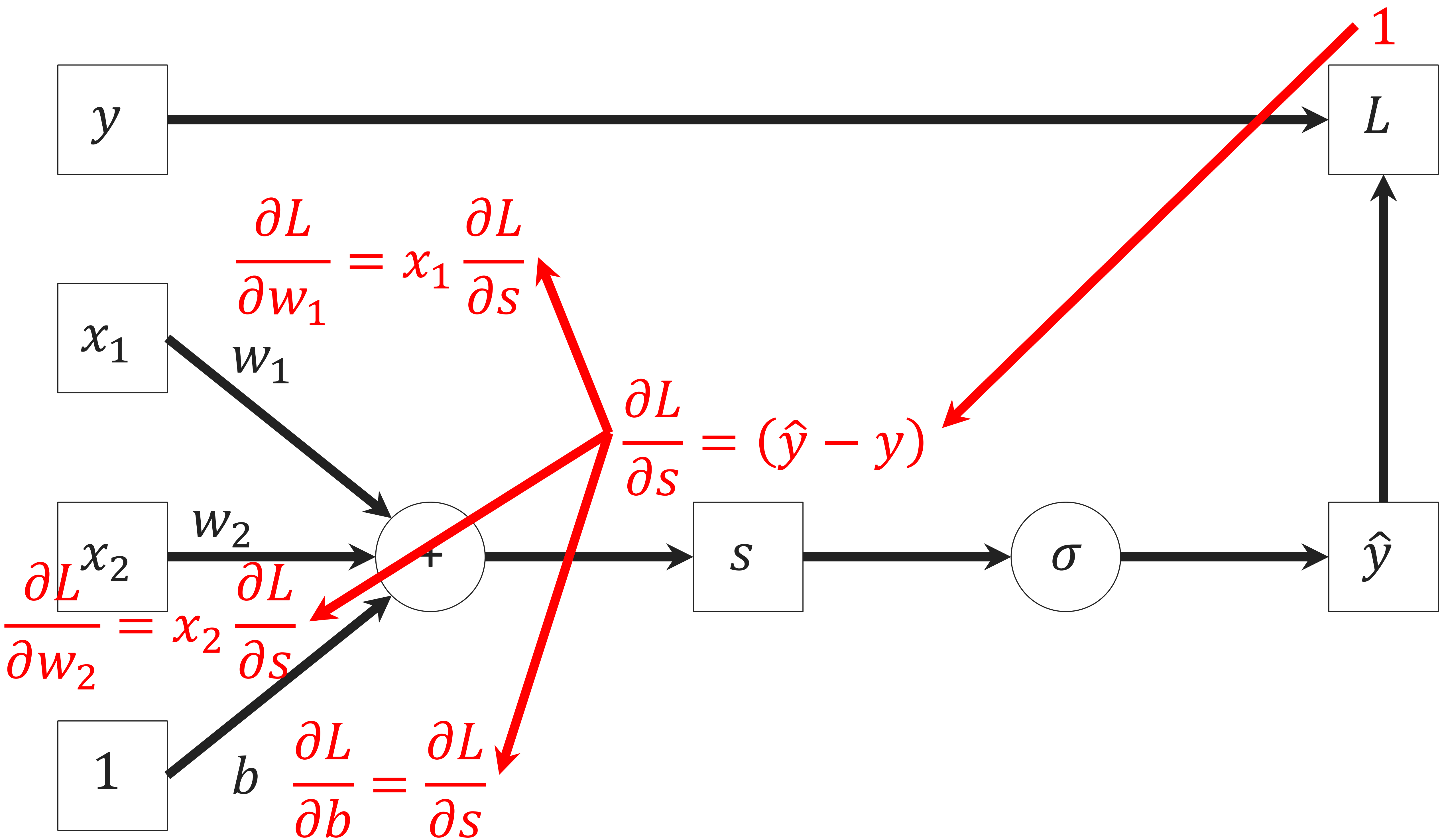



$+(\hat{y}-y)\delta$
y
L
$\frac{\partial L}{\partial s} = (\hat{y}-y)$
x_1
w_1
$+\delta$
x_2
w_2
$+$
s
σ
$\hat{y}$
1
b

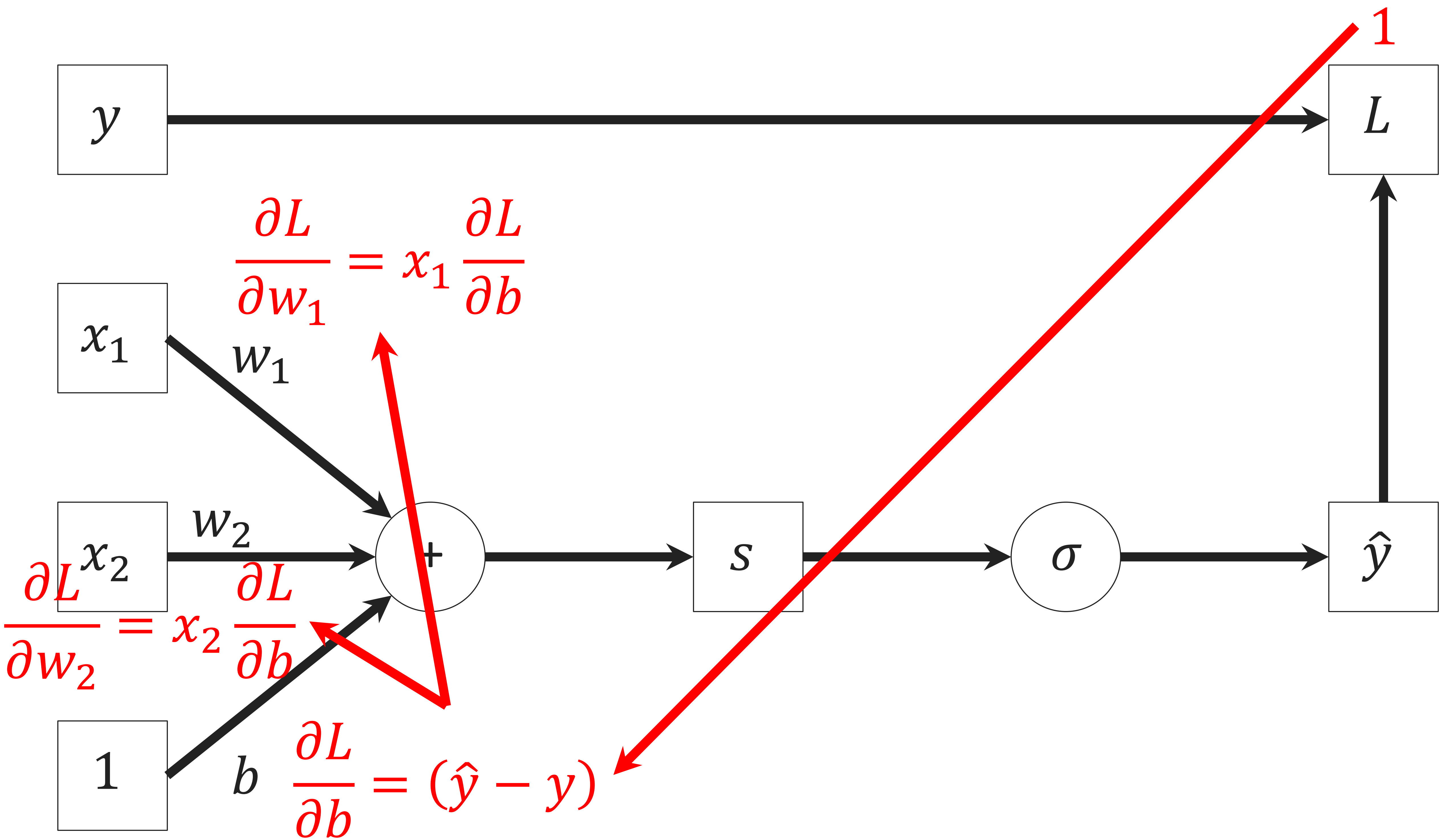

$+(\hat{y} - y)\delta$
y
L
$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial s} = (\hat{y} - y)$
x_1
w_1
$+\delta$
x_2
w_2
$+$
s
σ
$\hat{y}$
1
$b + \delta$



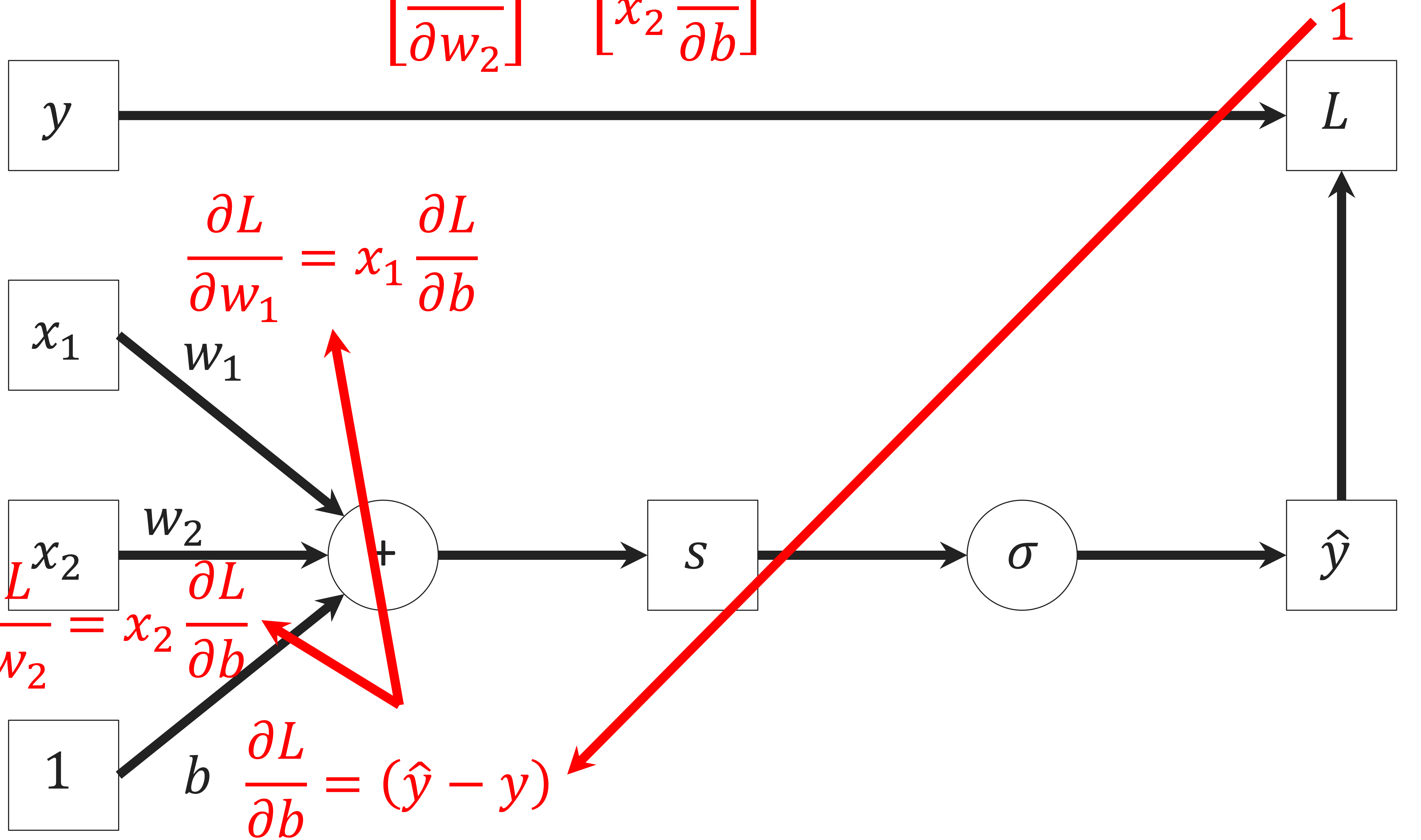
$+x_1(\hat{y}-y)\delta$
y
L
$\frac{\partial L}{\partial w_1} = x_1 \frac{\partial L}{\partial s} = x_1(\hat{y}-y)$
x_1
$w_1 + \delta$
$+x_1\delta$
x_2
w_2
$+$
s
σ
$\hat{y}$
1
b



y
L
1
$\frac{\partial L}{\partial w_1} = x_1 \frac{\partial L}{\partial s}$
x_1
w_1
$\frac{\partial L}{\partial s} = (\hat{y} - y)$
x_2
w_2
$+$
s
σ
$\hat{y}$
$\frac{\partial L}{\partial w_2} = x_2 \frac{\partial L}{\partial s}$
1
b
$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial s}$


y
L
x_1
w_1
x_2
w_2
$+$
s
σ
$\hat{y}$
1
b
1
$\frac{\partial L}{\partial w_1} = x_1 \frac{\partial L}{\partial b}$
$\frac{\partial L}{\partial w_2} = x_2 \frac{\partial L}{\partial b}$
$\frac{\partial L}{\partial b} = (\hat{y} - y)$

$$\frac{\partial L}{\partial \boldsymbol{w}} = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \end{bmatrix} = \begin{bmatrix} x_1 \frac{\partial L}{\partial b} \\ x_2 \frac{\partial L}{\partial b} \end{bmatrix} = \frac{\partial L}{\partial b} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{\partial L}{\partial b} \boldsymbol{x}^T$$

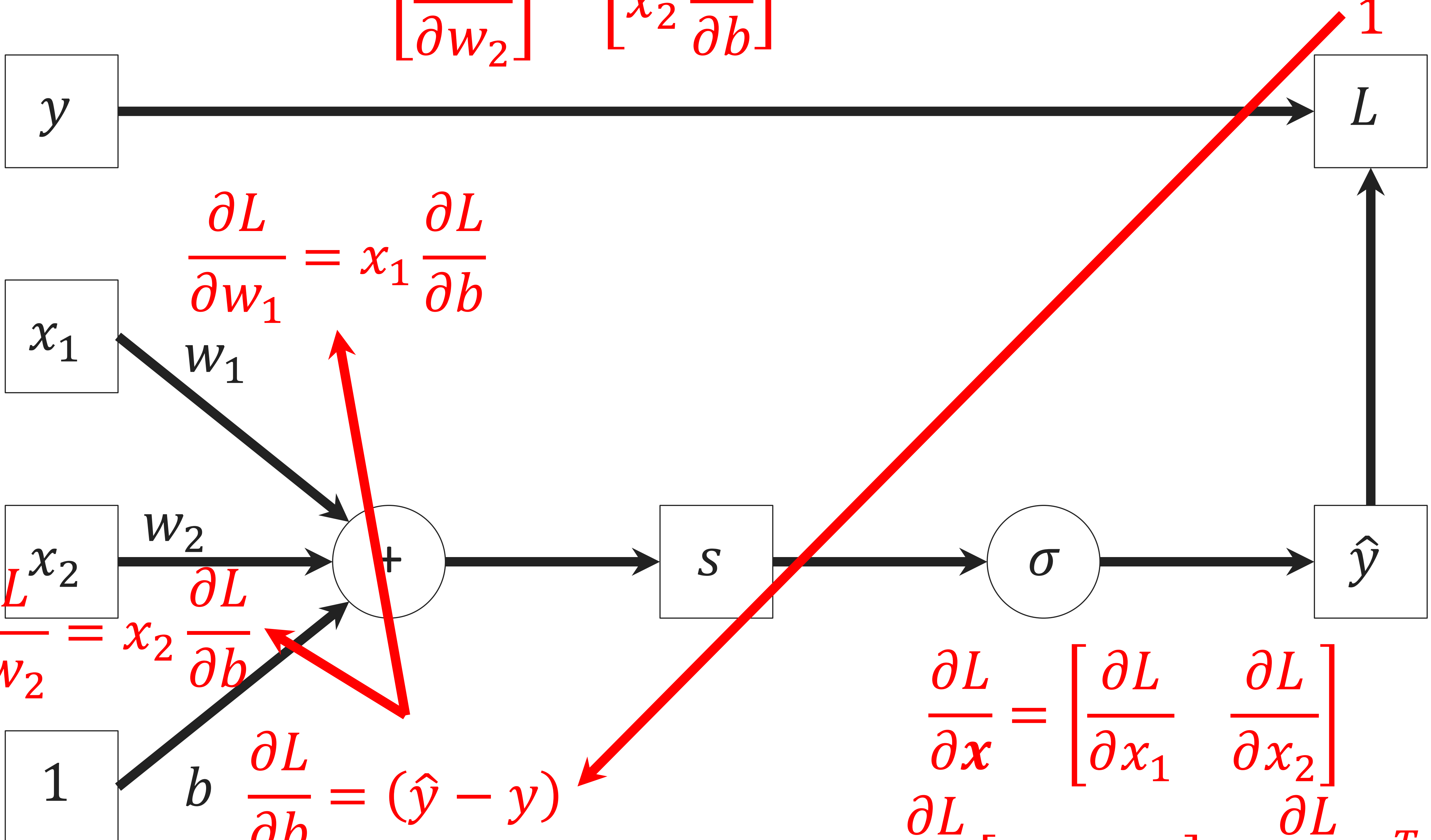


$$\frac{\partial L}{\partial w_1} = x_1 \frac{\partial L}{\partial b}$$

$$\frac{\partial L}{\partial w_2} = x_2 \frac{\partial L}{\partial b}$$

$$\frac{\partial L}{\partial b} = (\hat{y} - y)$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \end{bmatrix} = \begin{bmatrix} x_1 \frac{\partial L}{\partial b} \\ x_2 \frac{\partial L}{\partial b} \end{bmatrix} = \frac{\partial L}{\partial b} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{\partial L}{\partial b} \boldsymbol{x}^T$$



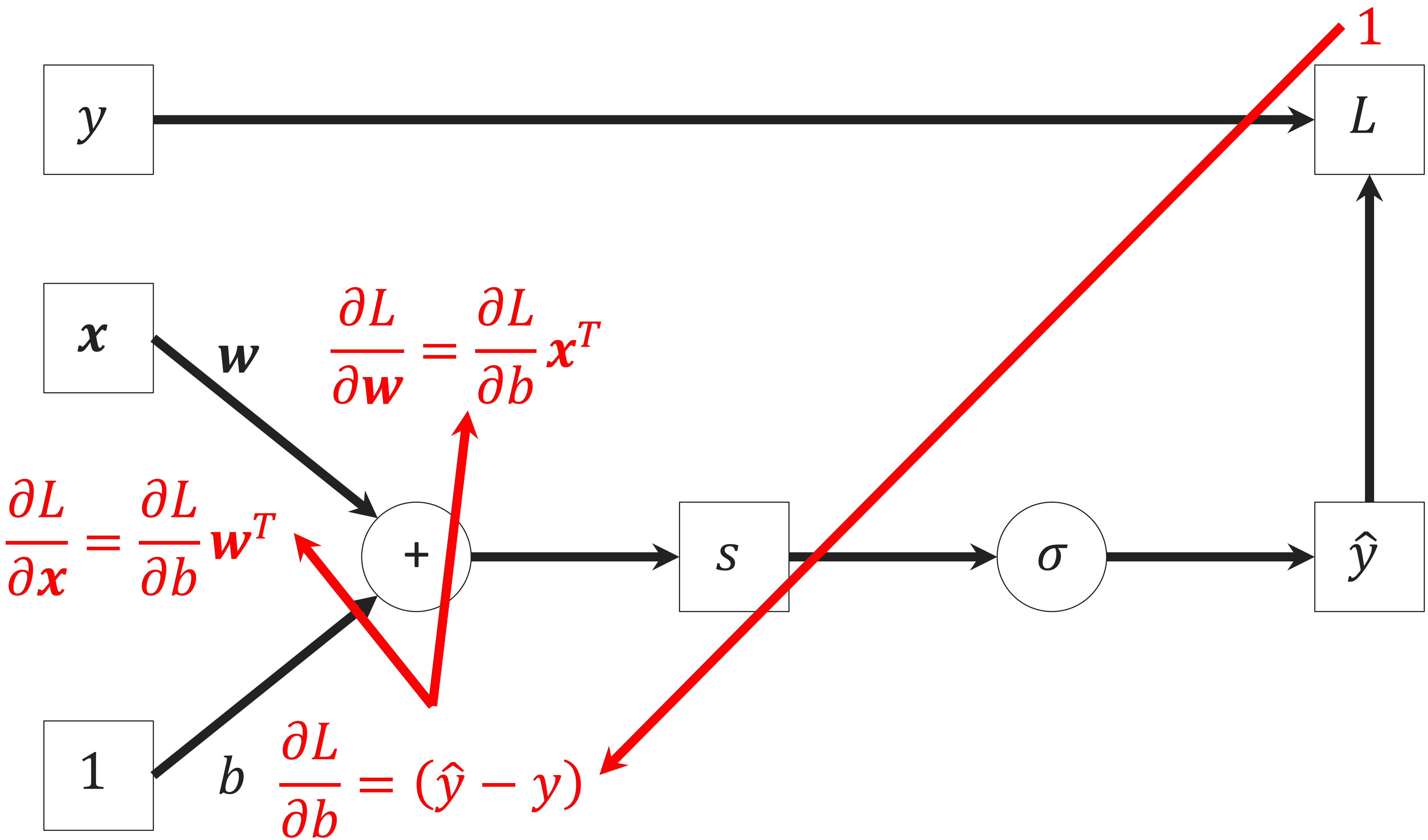
$$\frac{\partial L}{\partial w_1} = x_1 \frac{\partial L}{\partial b}$$

$$\frac{\partial L}{\partial w_2} = x_2 \frac{\partial L}{\partial b}$$

$$\frac{\partial L}{\partial b} = (\hat{y} - y)$$

$$\frac{\partial L}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial L}{\partial x_1} & \frac{\partial L}{\partial x_2} \end{bmatrix} = \frac{\partial L}{\partial b} \begin{bmatrix} w_1 & w_2 \end{bmatrix} = \frac{\partial L}{\partial b} \boldsymbol{w}^T$$

Backpropagation



Outline

- Review the lecture, background knowledge, etc.
  - Gradient descent & stochastic gradient descent (SGD)
  - Gradient and backpropagation
    - Logistic regression
    - Neural networks with one hidden layer
- Notebook tasks
  - Task 1: Multi-layer perceptron, SGD

Neural networks with one hidden layer

- Input: 2-D points $\boldsymbol{x} = [x_1 \quad x_2]$
- Hidden layer: 2 units $\boldsymbol{h} = [h_1 \quad h_2]$

$$\boldsymbol{t} = [t_1 \quad t_2] = \boldsymbol{xV} + \boldsymbol{c} = [x_1 \quad x_2]\begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} + [c_1 \quad c_2]$$

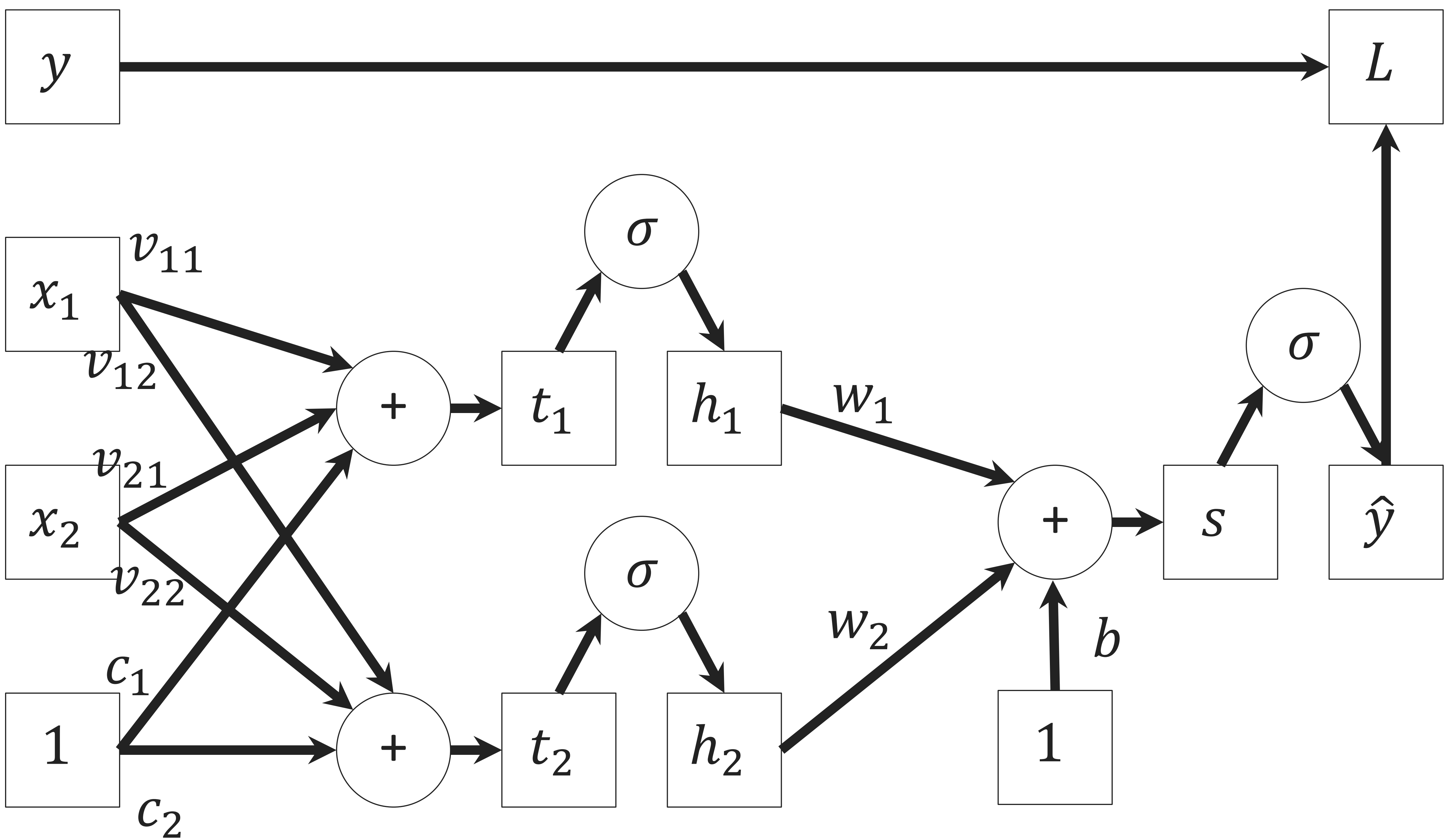
$$\boldsymbol{h} = [h_1 \quad h_2] = \sigma(\boldsymbol{t}) = \sigma([t_1 \quad t_2]) = [\sigma(t_1) \quad \sigma(t_2)]$$

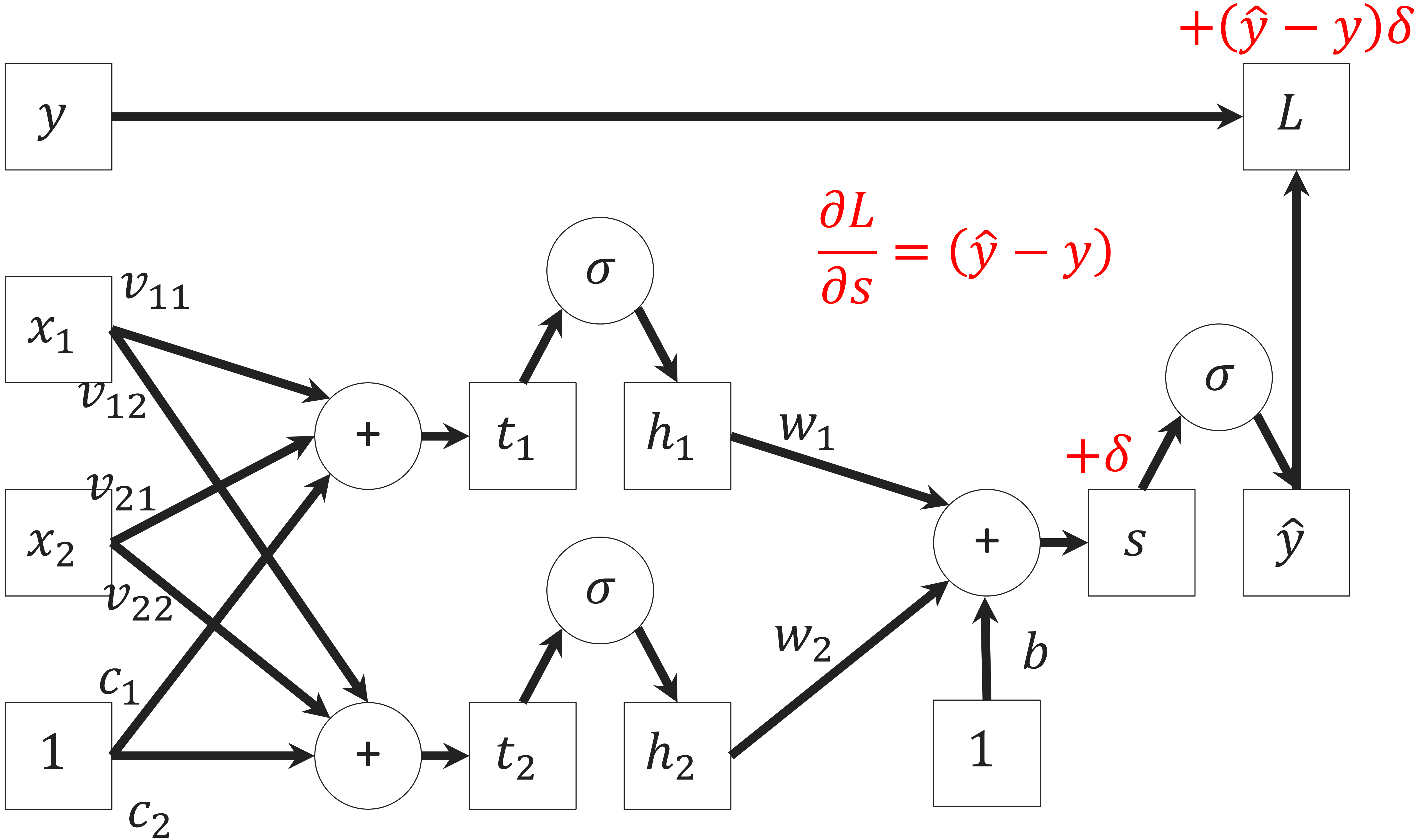
- Output: $\hat{y}$

$$s = \boldsymbol{hw} + b = [h_1 \quad h_2]\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + b$$

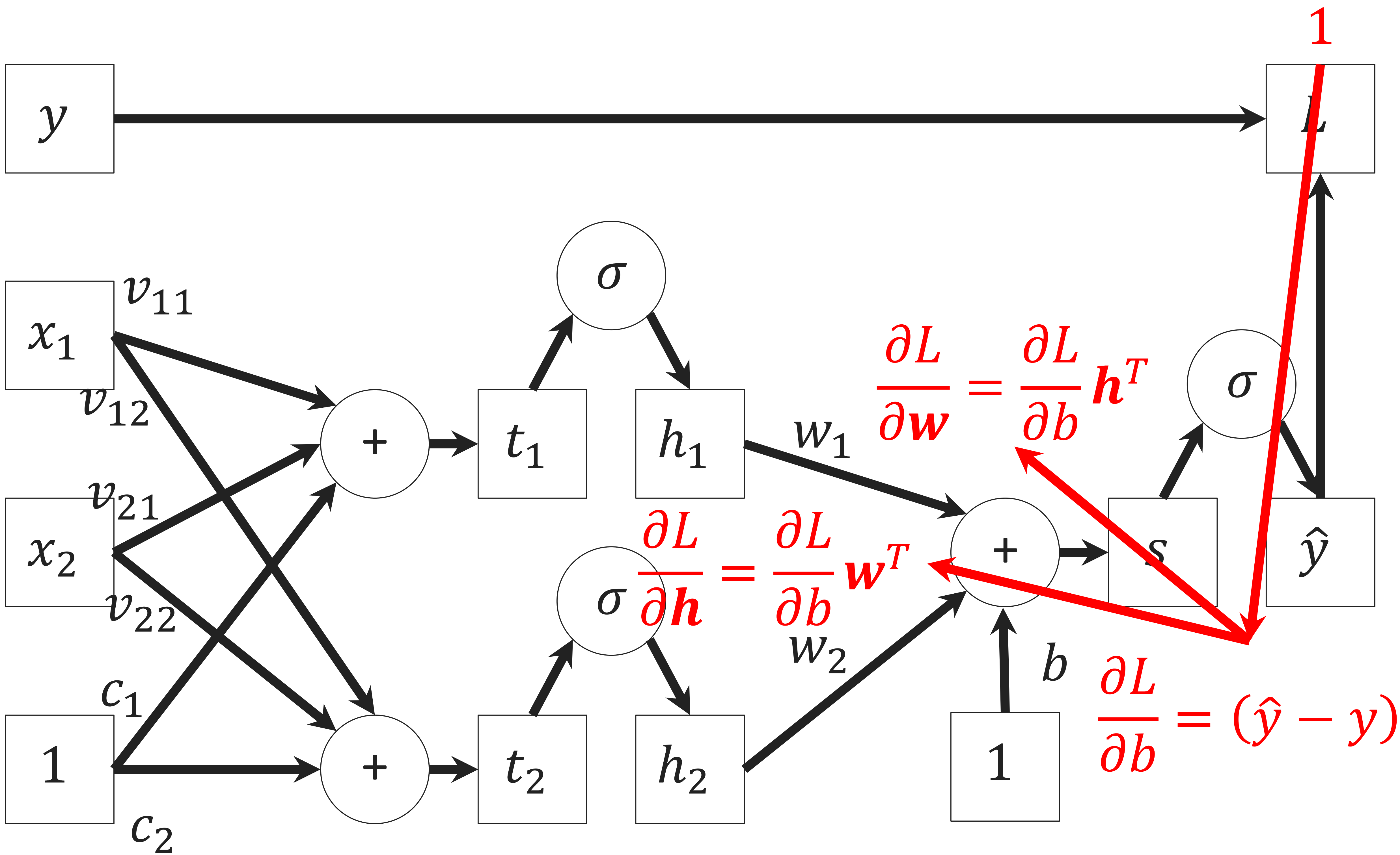
$$\hat{y} = \sigma(s)$$

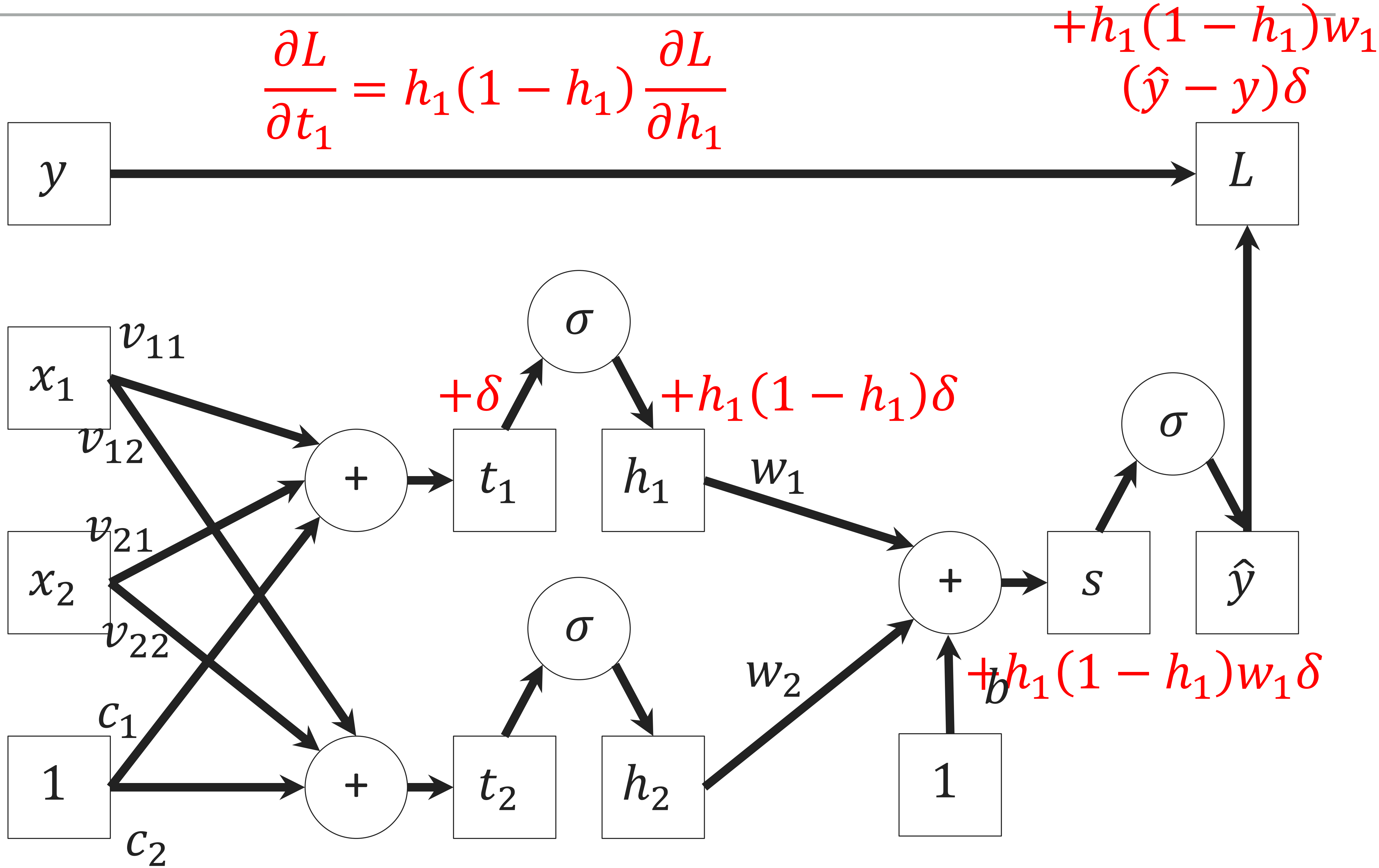
Forward pass



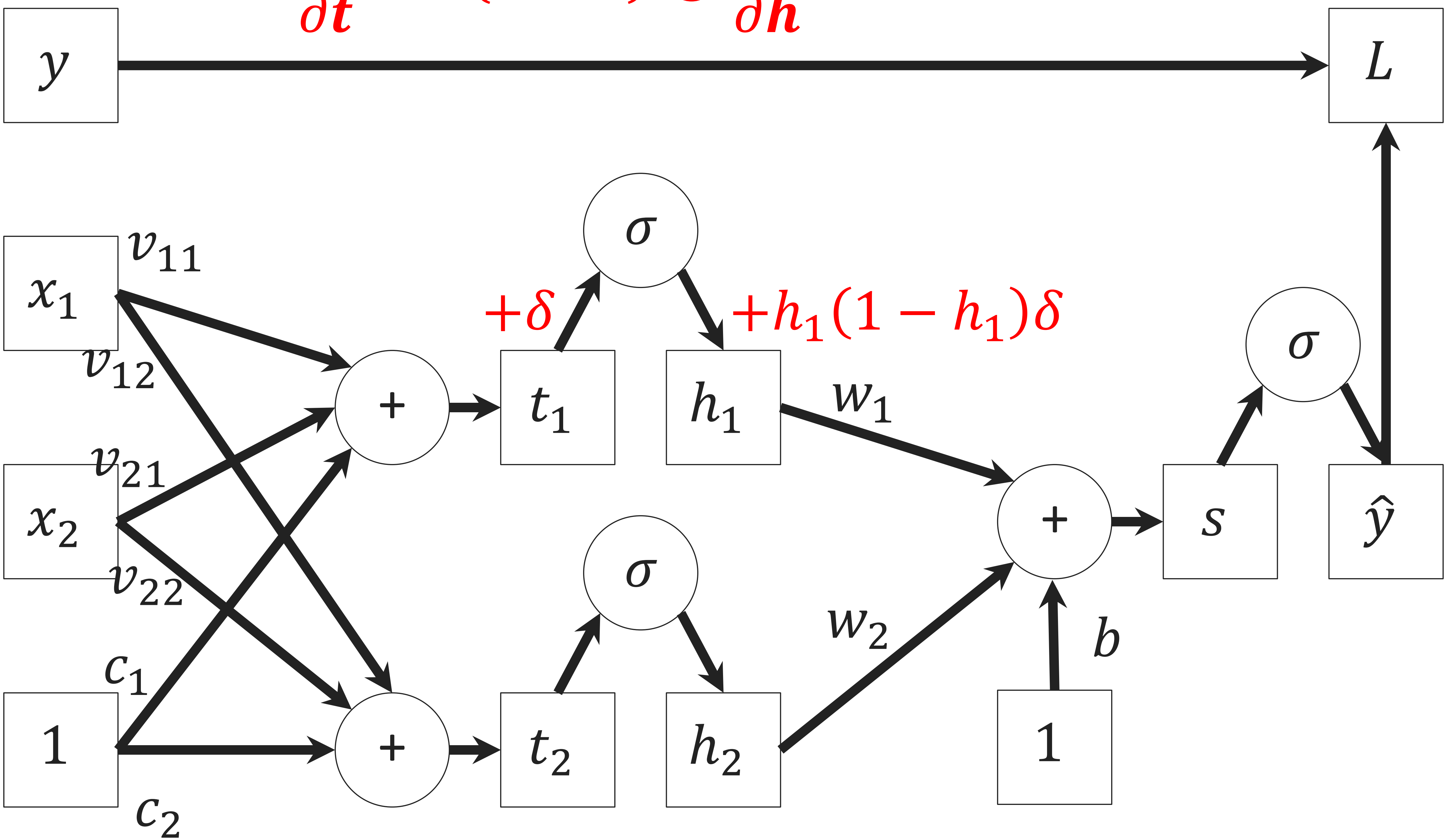


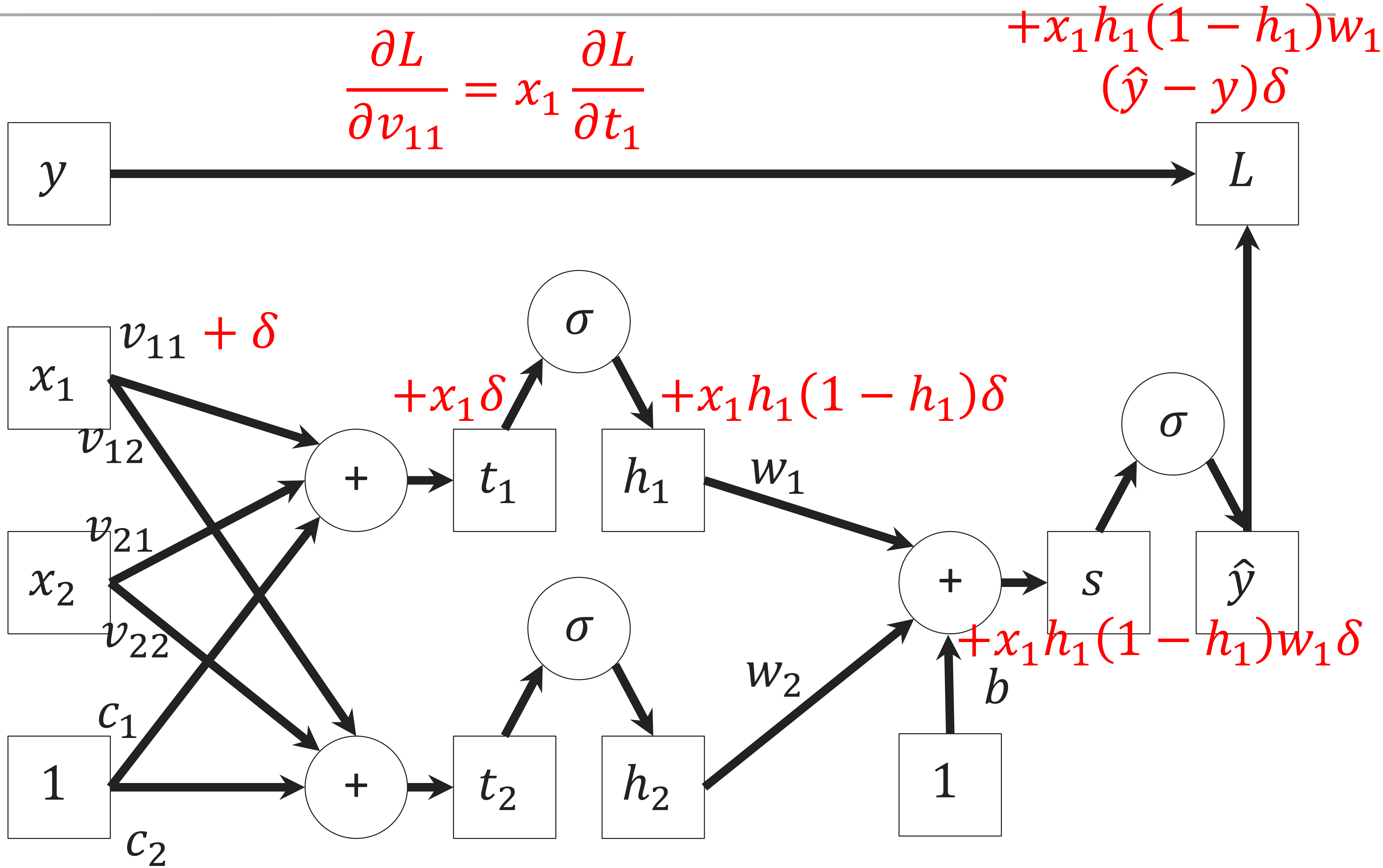
$+(\hat{y}-y)\delta$
y
L
$\frac{\partial L}{\partial s} = (\hat{y}-y)$
x_1
v_{11}
v_{12}
σ
σ
$+$
t_1
h_1
w_1
$+\delta$
v_{21}
x_2
$+$
s
$\hat{y}$
v_{22}
σ
w_2
b
c_1
1
$+$
t_2
h_2
1
c_2


y
L
1
x_1
x_2
1
v_{11}
v_{12}
v_{21}
v_{22}
c_1
c_2
$+$
$+$
t_1
t_2
σ
σ
h_1
h_2
w_1
w_2
$+$
b
1
s
σ
$\hat{y}$
$\frac{\partial L}{\partial \boldsymbol{w}} = \frac{\partial L}{\partial b} \boldsymbol{h}^T$
$\frac{\partial L}{\partial \boldsymbol{h}} = \frac{\partial L}{\partial b} \boldsymbol{w}^T$
$\frac{\partial L}{\partial b} = (\hat{y} - y)$


$\frac{\partial L}{\partial t_1} = h_1(1 - h_1)\frac{\partial L}{\partial h_1}$
$+h_1(1 - h_1)w_1$
$(\hat{y} - y)\delta$
y
L
x_1
x_2
1
v_{11}
v_{12}
v_{21}
v_{22}
c_1
c_2
$+$
$+$
t_1
t_2
$+\delta$
σ
σ
$+h_1(1 - h_1)\delta$
h_1
h_2
w_1
w_2
$+$
b
1
$+h_1(1 - h_1)w_1\delta$
s
σ
$\hat{y}$

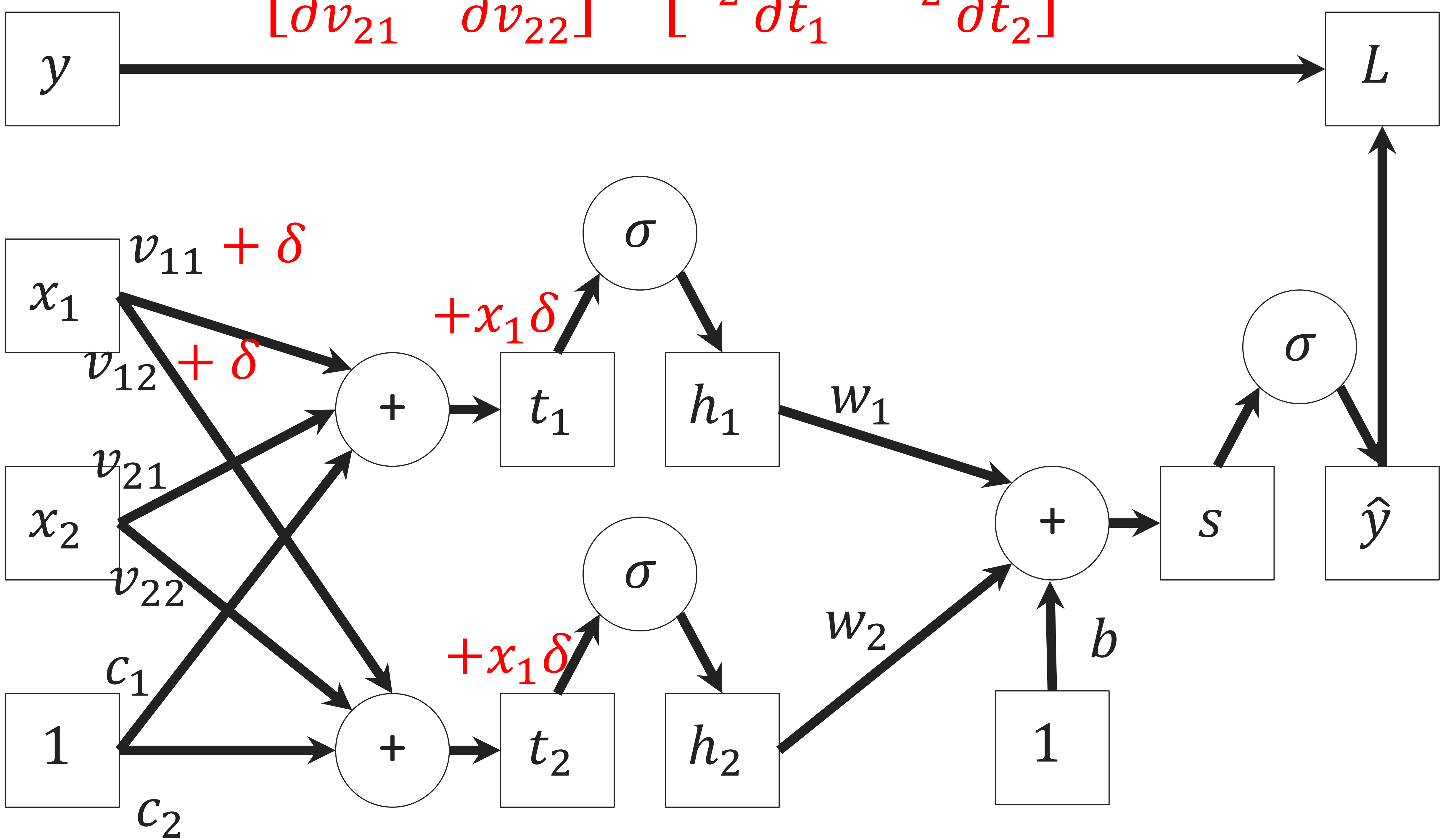
$$\frac{\partial L}{\partial \boldsymbol{t}} = \boldsymbol{h}(1-\boldsymbol{h}) \odot \frac{\partial L}{\partial \boldsymbol{h}}$$



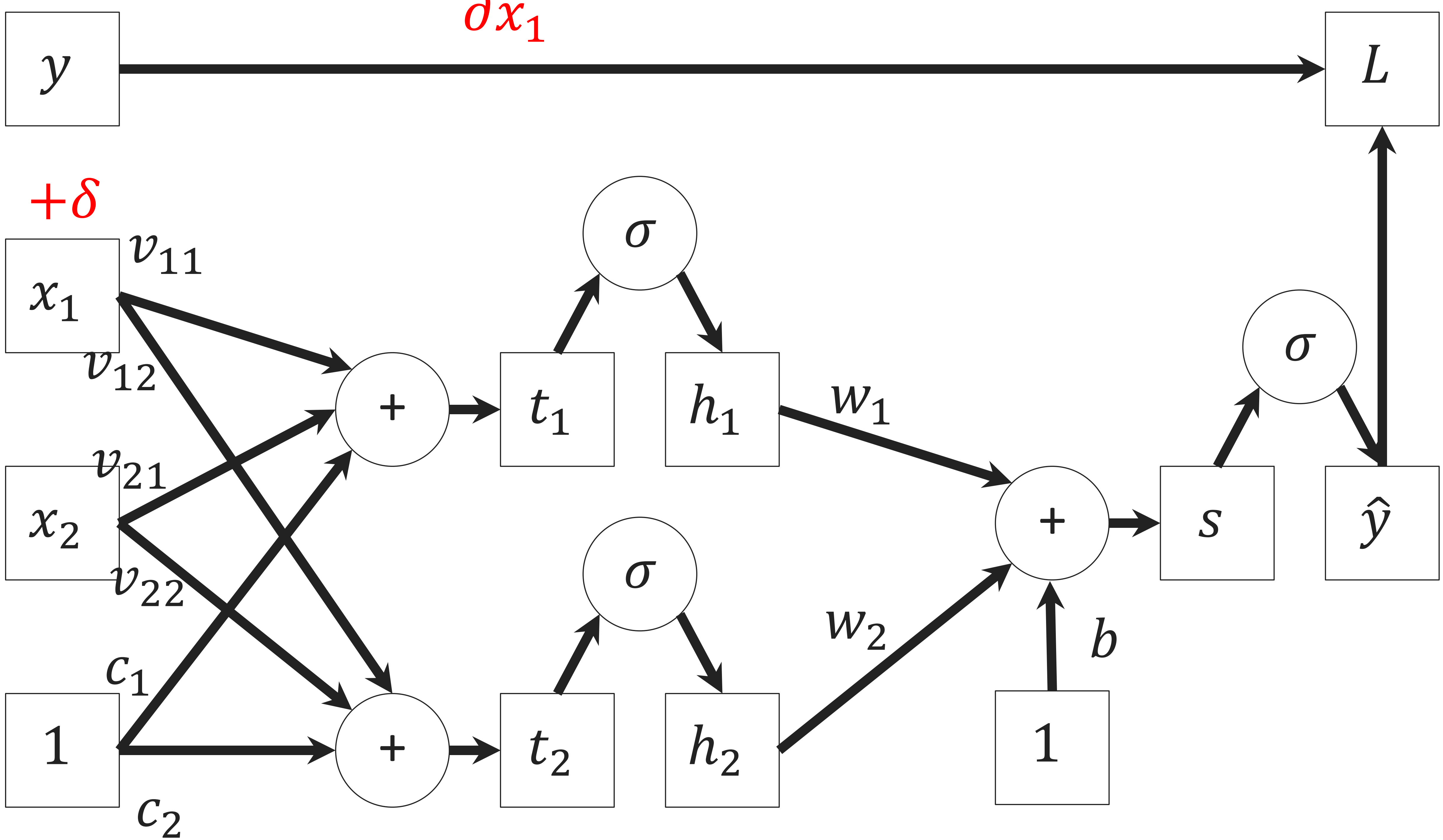


$+x_1 h_1(1-h_1)w_1$
$(\hat{y}-y)\delta$
$\frac{\partial L}{\partial v_{11}} = x_1 \frac{\partial L}{\partial t_1}$
y
L
x_1
$v_{11} + \delta$
v_{12}
$+x_1\delta$
σ
$+x_1 h_1(1-h_1)\delta$
t_1
h_1
w_1
σ
v_{21}
x_2
$+$
s
$\hat{y}$
v_{22}
$+x_1 h_1(1-h_1)w_1\delta$
σ
w_2
b
c_1
1
$+$
t_2
h_2
1
c_2

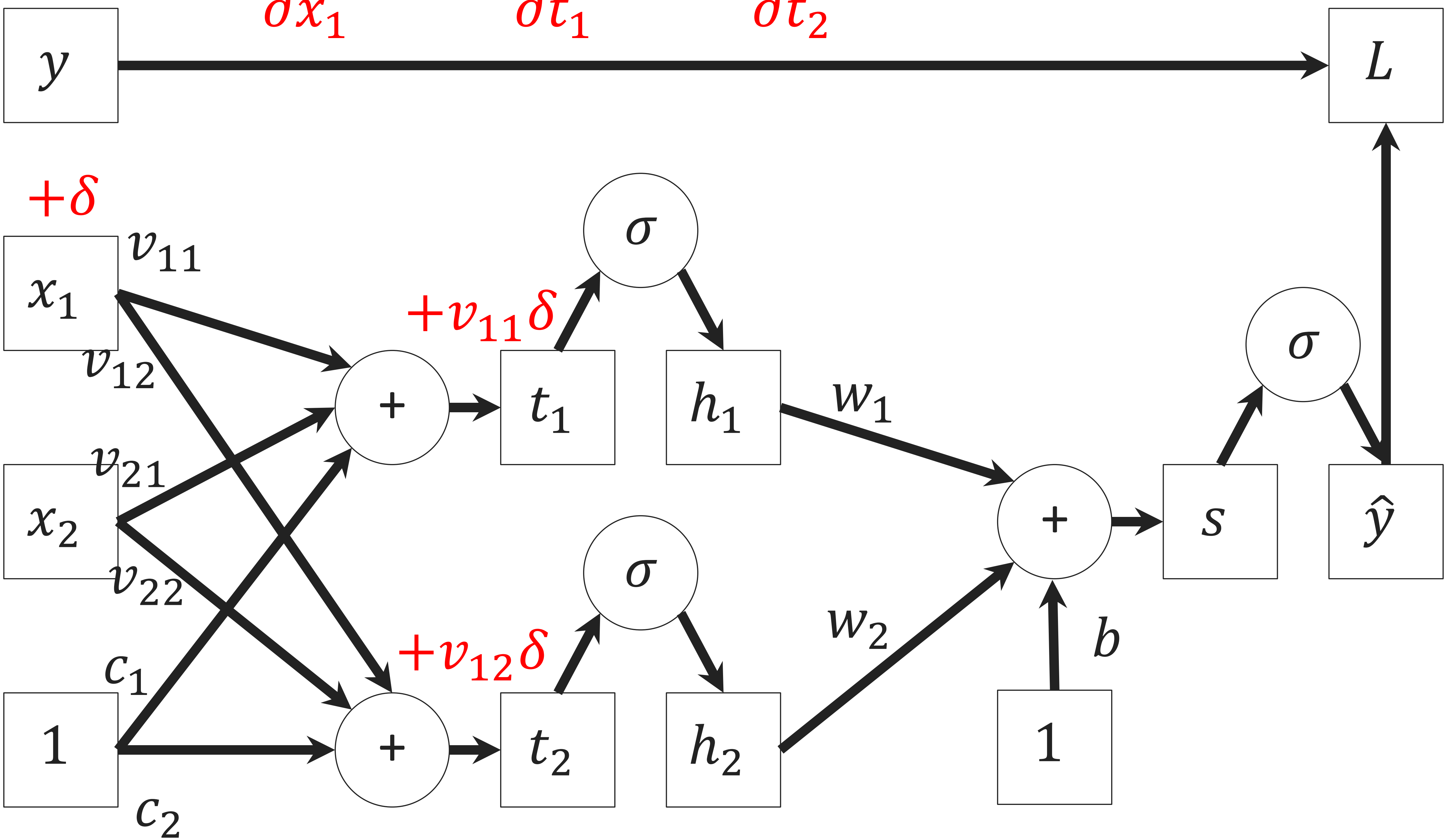
$$\frac{\partial L}{\partial \boldsymbol{V}} = \begin{bmatrix} \frac{\partial L}{\partial v_{11}} & \frac{\partial L}{\partial v_{12}} \\ \frac{\partial L}{\partial v_{21}} & \frac{\partial L}{\partial v_{22}} \end{bmatrix} = \begin{bmatrix} x_1 \frac{\partial L}{\partial t_1} & x_1 \frac{\partial L}{\partial t_2} \\ x_2 \frac{\partial L}{\partial t_1} & x_2 \frac{\partial L}{\partial t_2} \end{bmatrix} = \frac{\partial L}{\partial \boldsymbol{t}} \boldsymbol{x}^T$$



$$\frac{\partial L}{\partial x_1} = ?$$

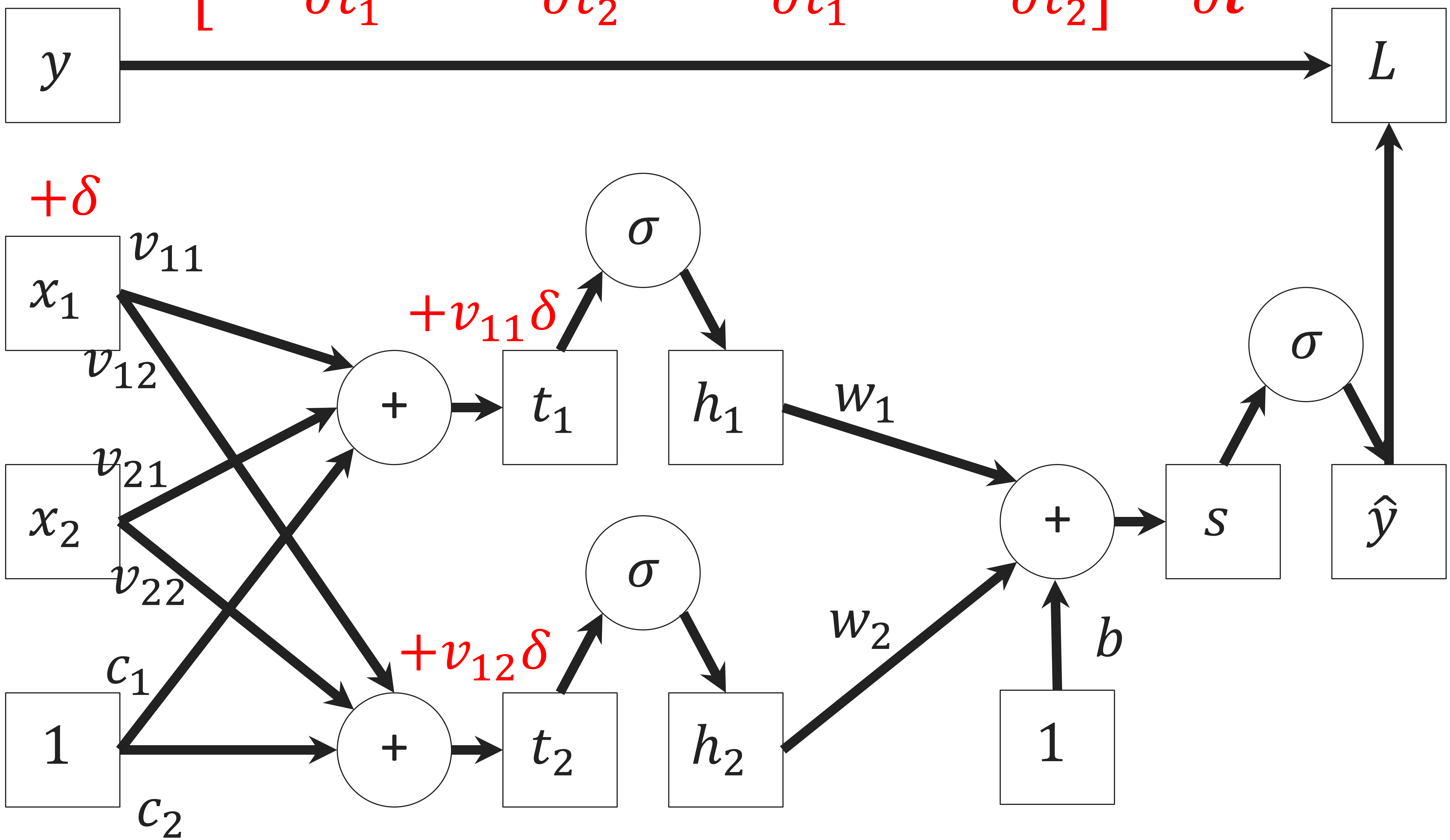


$$\frac{\partial L}{\partial x_1} = v_{11}\frac{\partial L}{\partial t_1} + v_{12}\frac{\partial L}{\partial t_2}$$

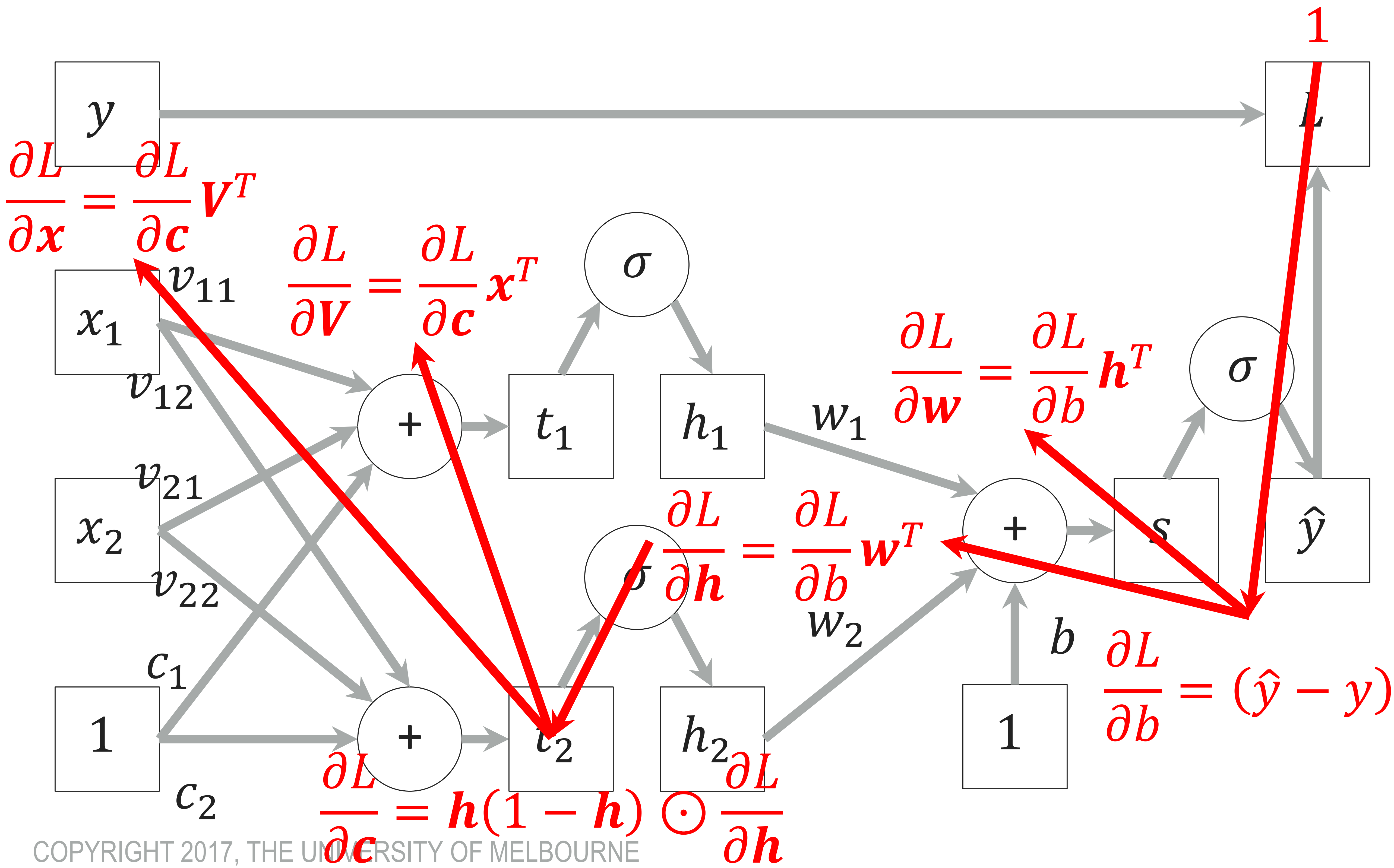


$$\frac{\partial L}{\partial \boldsymbol{x}} = \left[\frac{\partial L}{\partial x_1} \quad \frac{\partial L}{\partial x_2}\right]$$

$$= \left[v_{11}\frac{\partial L}{\partial t_1} + v_{12}\frac{\partial L}{\partial t_2} \quad v_{21}\frac{\partial L}{\partial t_1} + v_{22}\frac{\partial L}{\partial t_2}\right] = \frac{\partial L}{\partial \boldsymbol{t}}\boldsymbol{V}^T$$



Backpropagation



Outline

- Review the lecture, background knowledge, etc.
  - Gradient descent & stochastic gradient descent (SGD)
  - Gradient and backpropagation
    - Logistic regression
    - Neural networks with one hidden layer
- Notebook tasks
  - Task 1: Multi-layer perceptron, SGD