Machine Learning

CS 539

Worcester Polytechnic Institute
Department of Computer Science
Instructor: Prof. Kyumin Lee

HW1 grading

HW3

- https://canvas.wpi.edu/courses/57384/assignments/340697?module_item_id=1073181

- Due date is Feb 20.

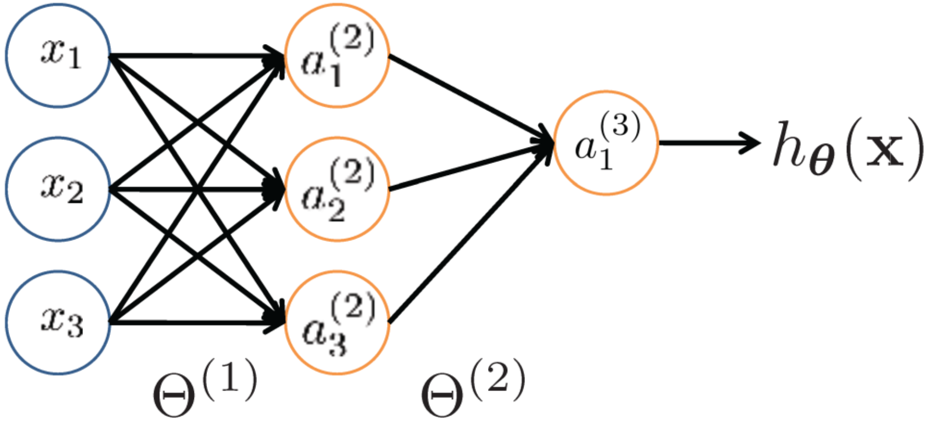
Vectorization

$$a_1^{(2)} = g\left(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3\right) = g\left(z_1^{(2)}\right)$$

$$a_2^{(2)} = g\left(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3\right) = g\left(z_2^{(2)}\right)$$

$$a_3^{(2)} = g\left(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3\right) = g\left(z_3^{(2)}\right)$$

$$h_\Theta(\mathbf{x}) = g\left(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)}\right) = g\left(z_1^{(3)}\right)$$



Feed-Forward Steps:

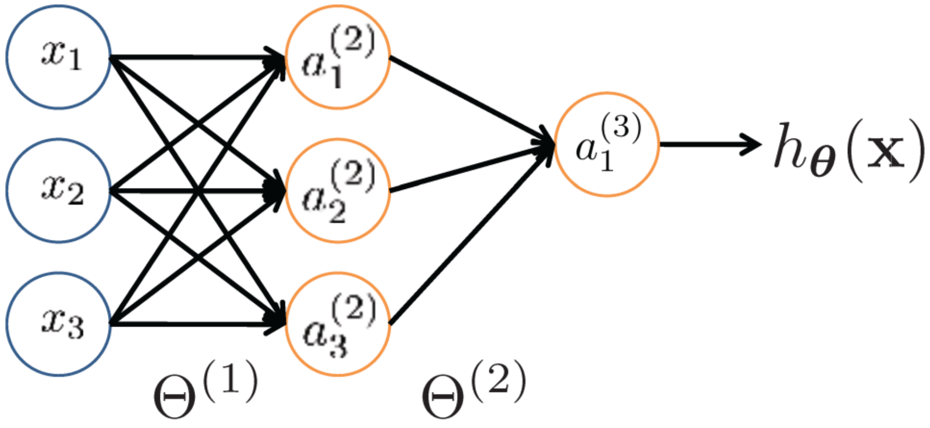
$$\mathbf{z}^{(2)} = \Theta^{(1)}\mathbf{x}$$

$$\mathbf{a}^{(2)} = g(\mathbf{z}^{(2)})$$

$$\text{Add } a_0^{(2)} = 1$$

$$\mathbf{z}^{(3)} = \Theta^{(2)}\mathbf{a}^{(2)}$$

$$h_\Theta(\mathbf{x}) = \mathbf{a}^{(3)} = g(\mathbf{z}^{(3)})$$



Feed-Forward Steps:

$$\mathbf{z}^{(2)} = \Theta^{(1)}\mathbf{x}$$

$$\mathbf{a}^{(2)} = g(\mathbf{z}^{(2)})$$

$$\text{Add } a_0^{(2)} = 1$$

$$\mathbf{z}^{(3)} = \Theta^{(2)}\mathbf{a}^{(2)}$$

$$h_\Theta(\mathbf{x}) = \mathbf{a}^{(3)} = g(\mathbf{z}^{(3)})$$

where $\text{cost}(\mathbf{x}_i) = -y_i \log h_\Theta(\mathbf{x}_i) - (1-y_i)\log(1-h_\Theta(\mathbf{x}_i))$

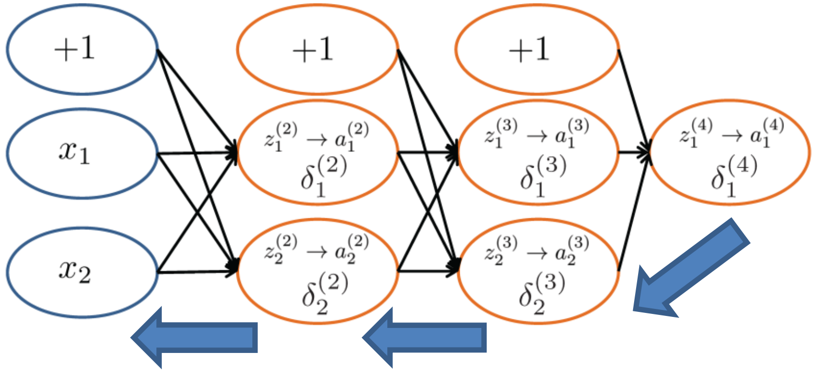
$$\frac{\partial\, \text{cost}(x_i)}{\partial \Theta^{(2)}} = \frac{\partial\, \text{cost}(x_i)}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial \Theta^{(2)}} = \frac{a^{(3)}-y}{a^{(3)}(1-a^{(3)})} \cdot a^{(3)}(1-a^{(3)}) \cdot a^{(2)} = a^{(2)} \cdot (a^{(3)}-y)$$

$$\frac{\partial\, \text{cost}(x_i)}{\partial \Theta^{(1)}} = \frac{\partial\, \text{cost}(x_i)}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial \Theta^{(1)}} = (a^{(3)}-y) \cdot \Theta^{(2)} \cdot a^{(2)}(1-a^{(2)}) \cdot x_i$$

Backpropagation: Gradient Computation (binary classification)

Let $\delta_j^{(l)}$ = "error" of node j in layer l

(#layers L = 4)



Element-wise product .*

Backpropagation

$\frac{\partial cost(x_i)}{\partial a^{(4)}} \times \frac{\partial a^{(4)}}{\partial z^{(4)}} =$

- $\boldsymbol{\delta}^{(4)} = \boldsymbol{a}^{(4)} - \mathbf{y}$
- $\boldsymbol{\delta}^{(3)} = (\Theta^{(3)})^{\mathsf{T}}\boldsymbol{\delta}^{(4)} \;.*\; g'(\mathbf{z}^{(3)})$
- $\boldsymbol{\delta}^{(2)} = (\Theta^{(2)})^{\mathsf{T}}\boldsymbol{\delta}^{(3)} \;.*\; g'(\mathbf{z}^{(2)})$
- (No $\boldsymbol{\delta}^{(1)}$)

$\boxed{\frac{\partial cost(x_i)}{\partial a^{(4)}} \cdot \frac{\partial a^{(4)}}{\partial z^{(4)}} \cdot \frac{\partial z^{(4)}}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}}}$

$\text{same} \times \frac{\partial z^{(3)}}{\partial a^{(2)}} \times \frac{\partial a^{(2)}}{\partial z^{(2)}}$

$g'(\mathbf{z}^{(3)}) = \mathbf{a}^{(3)} \;.*\; (1 - \mathbf{a}^{(3)})$

$g'(\mathbf{z}^{(2)}) = \mathbf{a}^{(2)} \;.*\; (1 - \mathbf{a}^{(2)})$

Assume g() is sigmoid

$\boxed{\frac{\partial a}{\partial z} = a(1-a)}$ sigmoid

$$\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) = a_j^{(l)} \delta_i^{(l+1)} = \frac{\partial cost(x_i)}{\partial a_i^{(l+1)}} \cdot \frac{\partial a_i^{(l+1)}}{\partial z_i^{(l+1)}} \cdot \frac{\partial z_i^{(l+1)}}{\partial \theta_{ij}^{(l)}}$$

(ignoring λ; if λ = 0)

Mathematical Proof: https://web.cs.wpi.edu/~kmlee/cs539/cs229-notes-deep_learning.pdf

Backpropagation

Set $\Delta_{ij}^{(l)} = 0 \quad \forall l, i, j$ (Used to accumulate gradient)

For each training instance $(\mathbf{x}_i, y_i)$:

Set $\mathbf{a}^{(1)} = \mathbf{x}_i$

Compute $\{\mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(L)}\}$ via forward propagation

Compute $\boldsymbol{\delta}^{(L)} = \mathbf{a}^{(L)} - y_i$

Compute errors $\{\boldsymbol{\delta}^{(L-1)}, \ldots, \boldsymbol{\delta}^{(2)}\}$

Compute gradients $\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$

Compute avg regularized gradient $D_{ij}^{(l)} = \begin{cases} \frac{1}{n}\Delta_{ij}^{(l)} + \lambda\Theta_{ij}^{(l)} & \text{if } j \neq 0 \\ \frac{1}{n}\Delta_{ij}^{(l)} & \text{otherwise} \end{cases}$

$\boldsymbol{D}^{(l)}$ is the matrix of partial derivatives of $J(\Theta)$

Note: Can vectorize $\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$ as $\boldsymbol{\Delta}^{(l)} = \boldsymbol{\Delta}^{(l)} + \boldsymbol{\delta}^{(l+1)} \mathbf{a}^{(l)\mathsf{T}}$

Training a Neural Network via Gradient Descent with Backprop

Given: training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
Initialize all $\Theta^{(l)}$ randomly (NOT to 0!)
Loop // each iteration is called an epoch

    Set $\Delta_{ij}^{(l)} = 0 \quad \forall l, i, j$ (Used to accumulate gradient)

    For each training instance $(\mathbf{x}_i, y_i)$:

        Set $\mathbf{a}^{(1)} = \mathbf{x}_i$

        Compute $\{\mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(L)}\}$ via forward propagation

        Compute $\boldsymbol{\delta}^{(L)} = \mathbf{a}^{(L)} - y_i$

        Compute errors $\{\boldsymbol{\delta}^{(L-1)}, \ldots, \boldsymbol{\delta}^{(2)}\}$

        Compute gradients $\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$

    Compute avg regularized gradient $D_{ij}^{(l)} = \begin{cases} \frac{1}{n}\Delta_{ij}^{(l)} + \lambda\Theta_{ij}^{(l)} & \text{if } j \neq 0 \\ \frac{1}{n}\Delta_{ij}^{(l)} & \text{otherwise} \end{cases}$

    Update weights via gradient step $\Theta_{ij}^{(l)} = \Theta_{ij}^{(l)} - \alpha D_{ij}^{(l)}$

Until weights converge or max #epochs is reached

Backpropagation

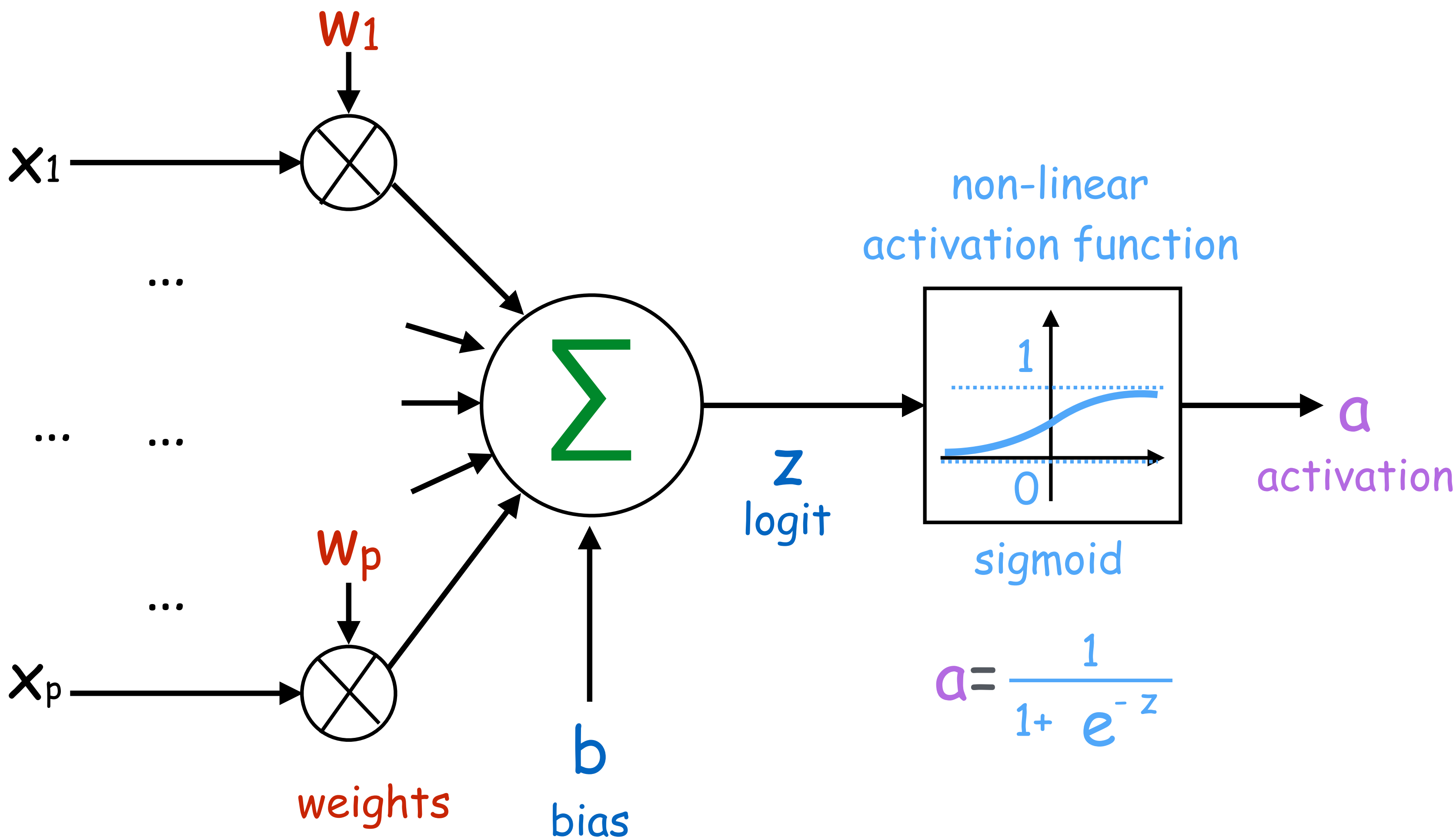
Multi-Class Classification

(Softmax regression and Fully Connected Neural Network)

Softmax Regression

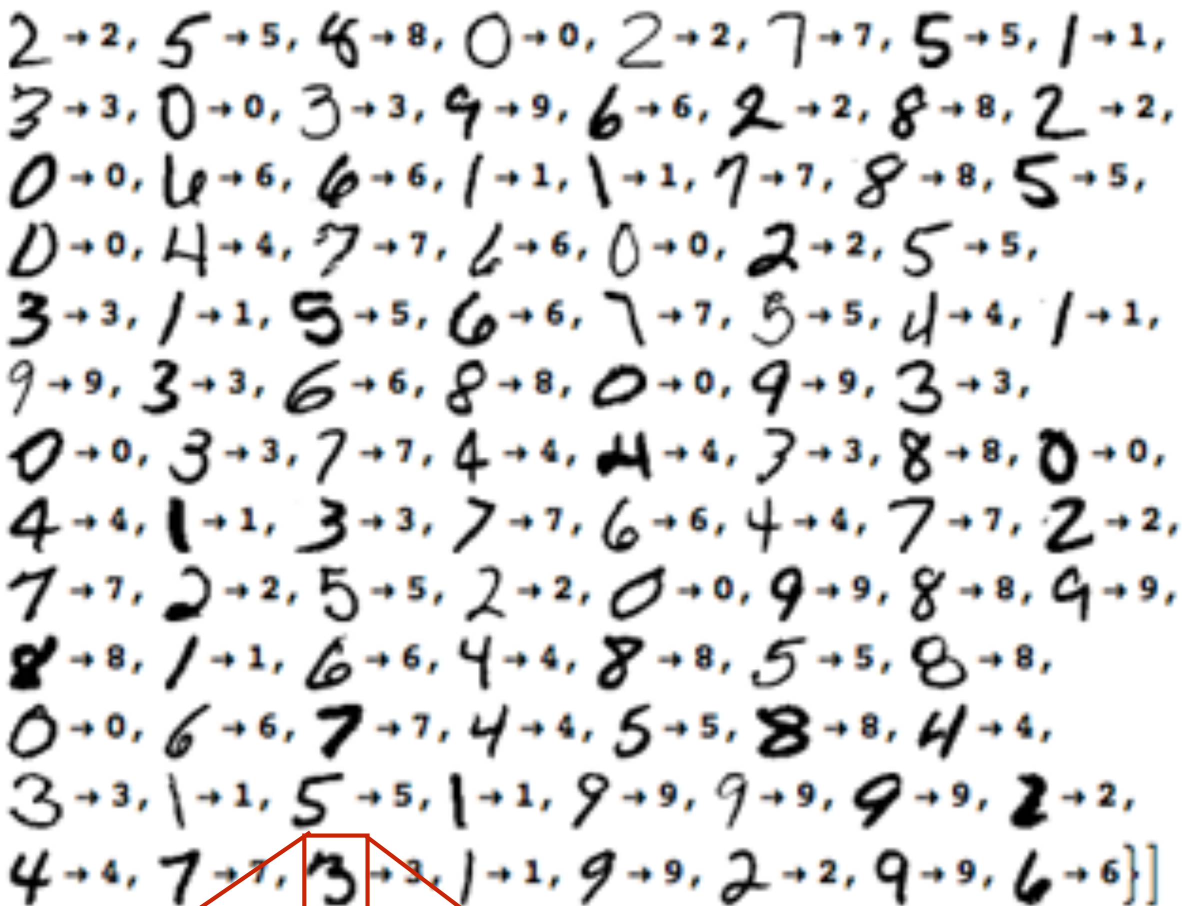
:a generalization of logistic regression to the case where we want to handle multiple classes

Logistic Regression



Multi-class Classification Problem

data set



candidate labels (classes)

0
1
2
⋮
9

input (instance)

output (label)

0 or 1 or … or 9

Feature Matrix X

(n by p)

Feature (pixel)

Instance

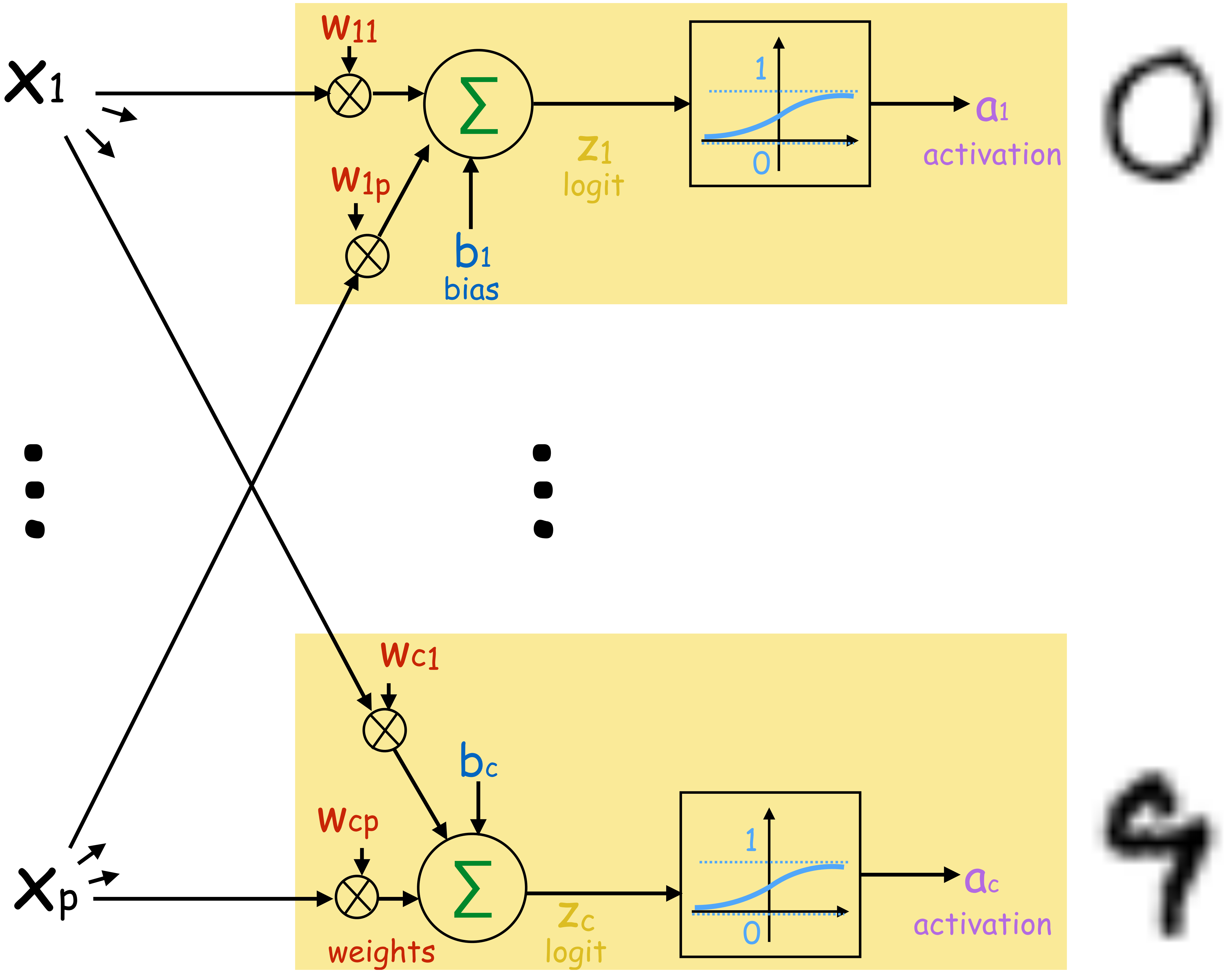
		1	2				…						p
x_1	2	1	3	4	3	8	3	5	7	9	7	3	4
x_2	5	3	3	5	7	7	0	4	1	2	1	9	7
	0	7	0	4	1	1	4	3	7	8	6	2	7
	3	1	4	3	7	9	7	3	2	7	0	4	1
	4	7	7	3	2	2	1	9	8	1	4	3	7
	0	2	1	9	8	8	6	2	0	7	7	3	2
⋮	9	8	6	2	0	0	4	1	1	4	1	9	8
	6	0	2	1	4	1	3	7	9	7	6	2	0
	7	3	5	3	3	7	3	2	2	1	2	1	3
x_n	6	1	7	2	3	2	2	1	2	3	5	3	1

Label Vector y

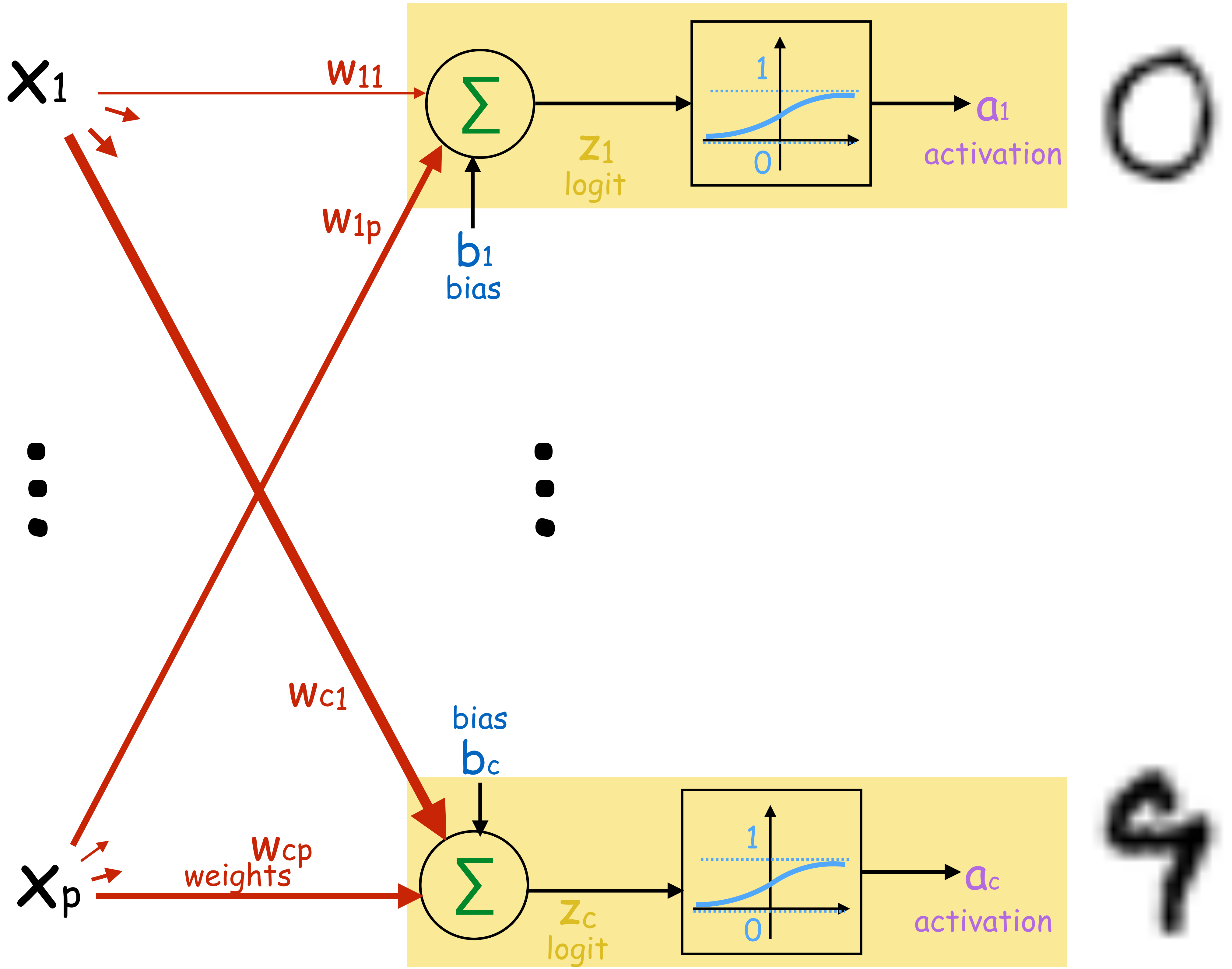
(length n)

y_1	2
y_2	5
	0
	3
	8
	0
⋮	9
	6
	7
y_n	6

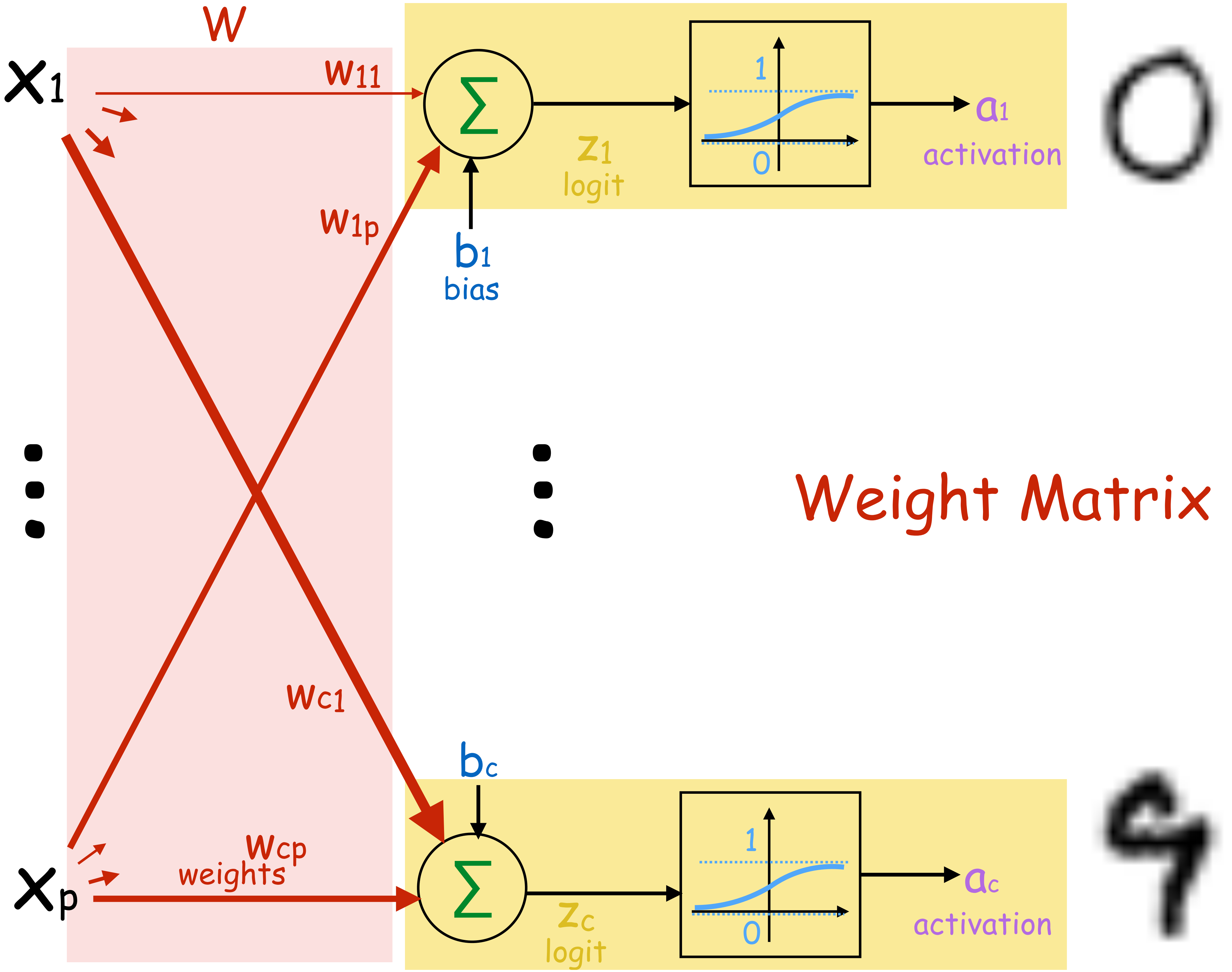
n - # instances p - # features



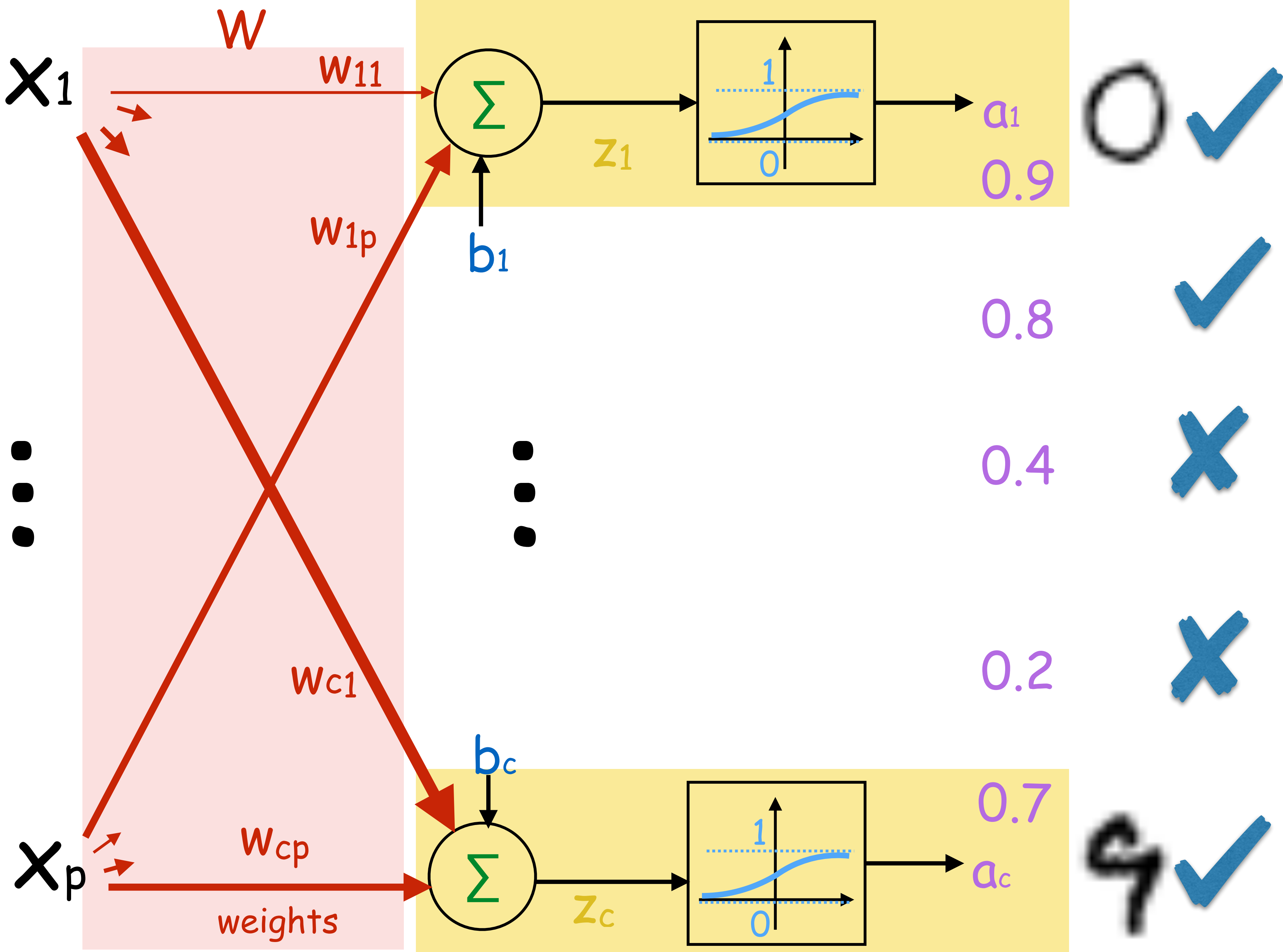
x1
w11
w1p
Σ
b1
bias
z1
logit
1
0
a1
activation
xp
wc1
wcp
weights
bc
Σ
zc
logit
1
0
ac
activation
0
9



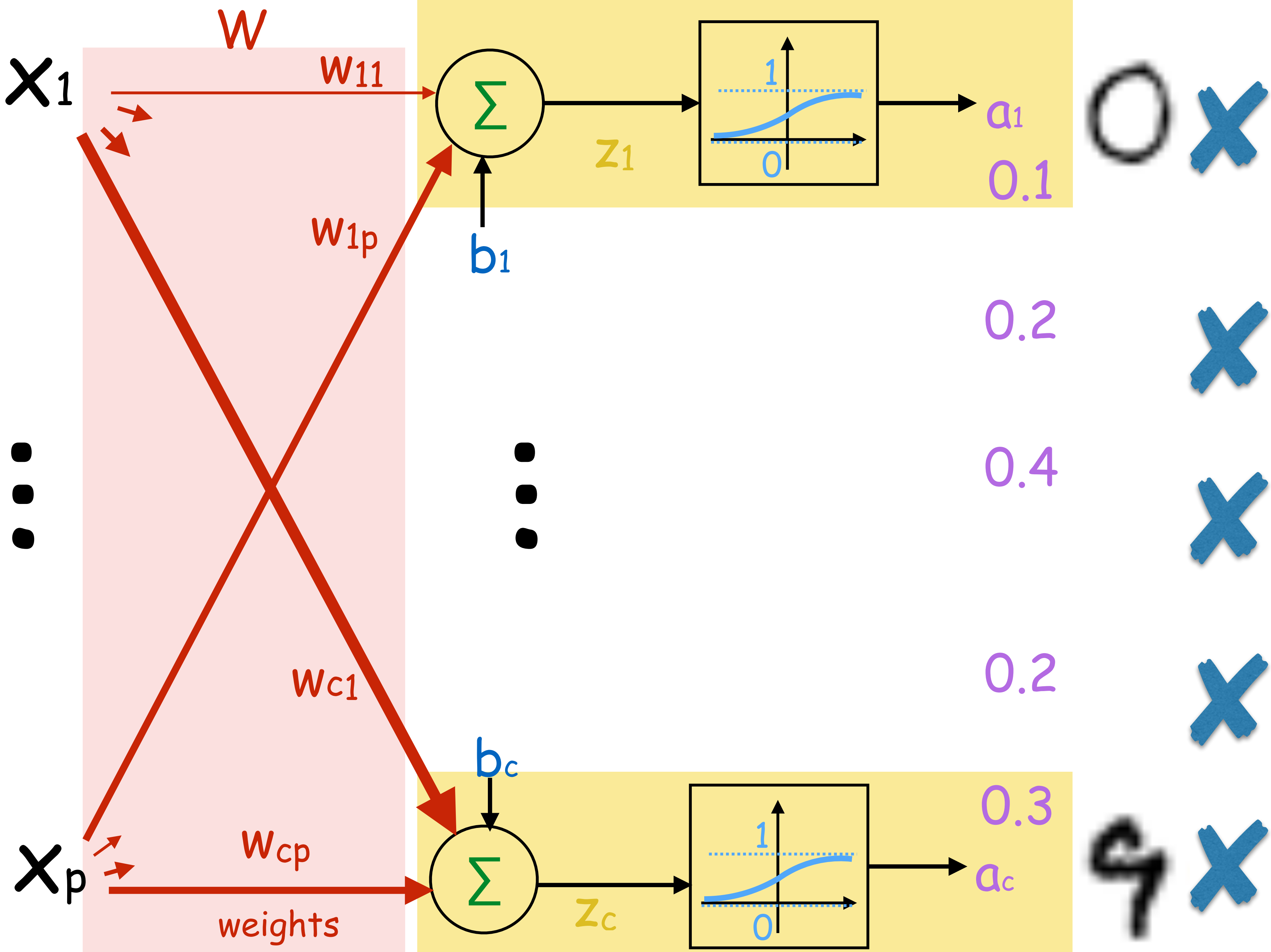
x1
w11
a1
activation
z1
logit
1
0
b1
bias
w1p
wc1
bias
bc
xp
wcp
weights
zc
logit
ac
activation


W
x1
xp
w11
w1p
wc1
wcp
weights
z1
logit
zc
logit
b1
bias
bc
1
0
a1
activation
ac
activation
Weight Matrix

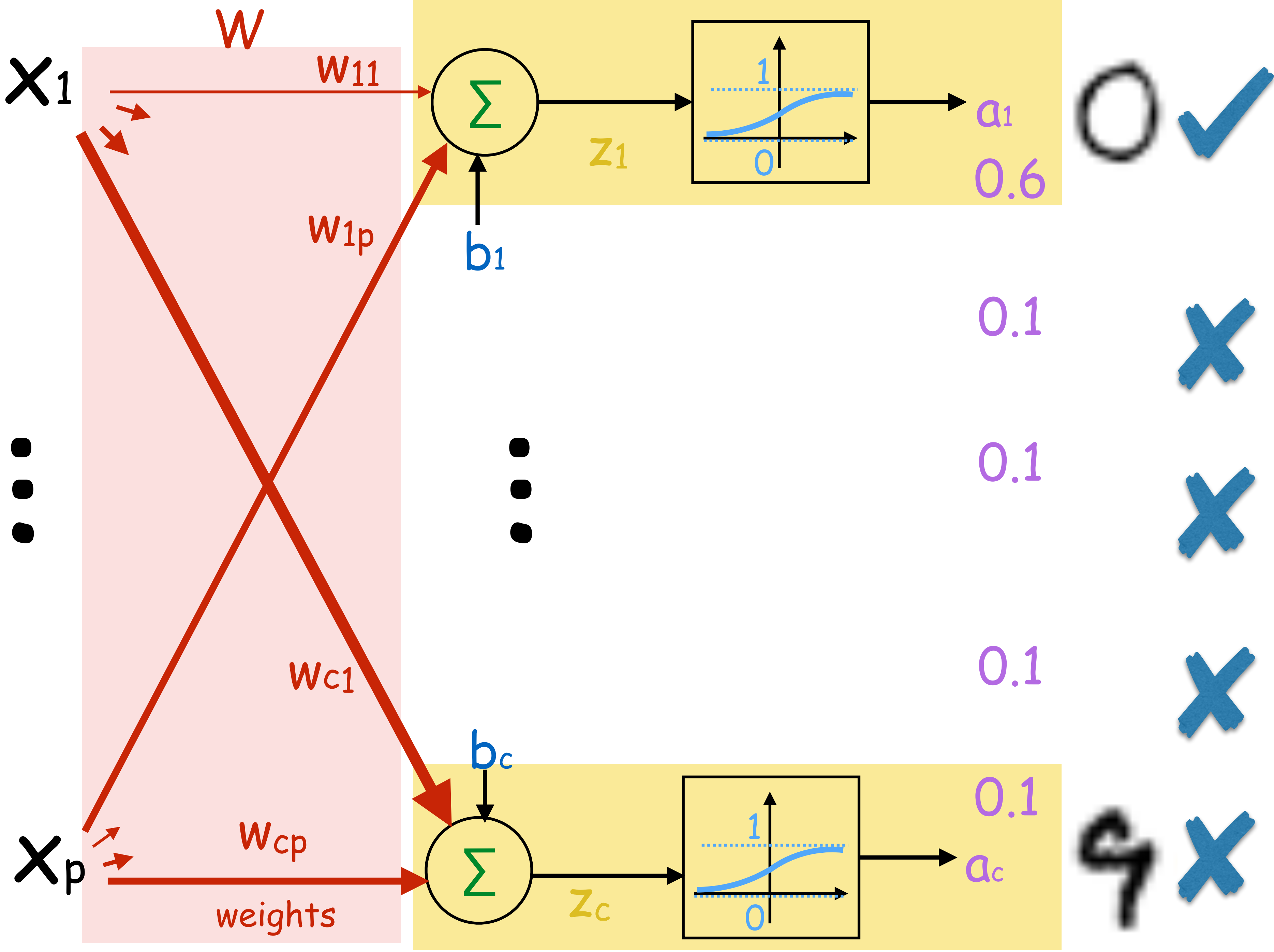
Independent Outputs



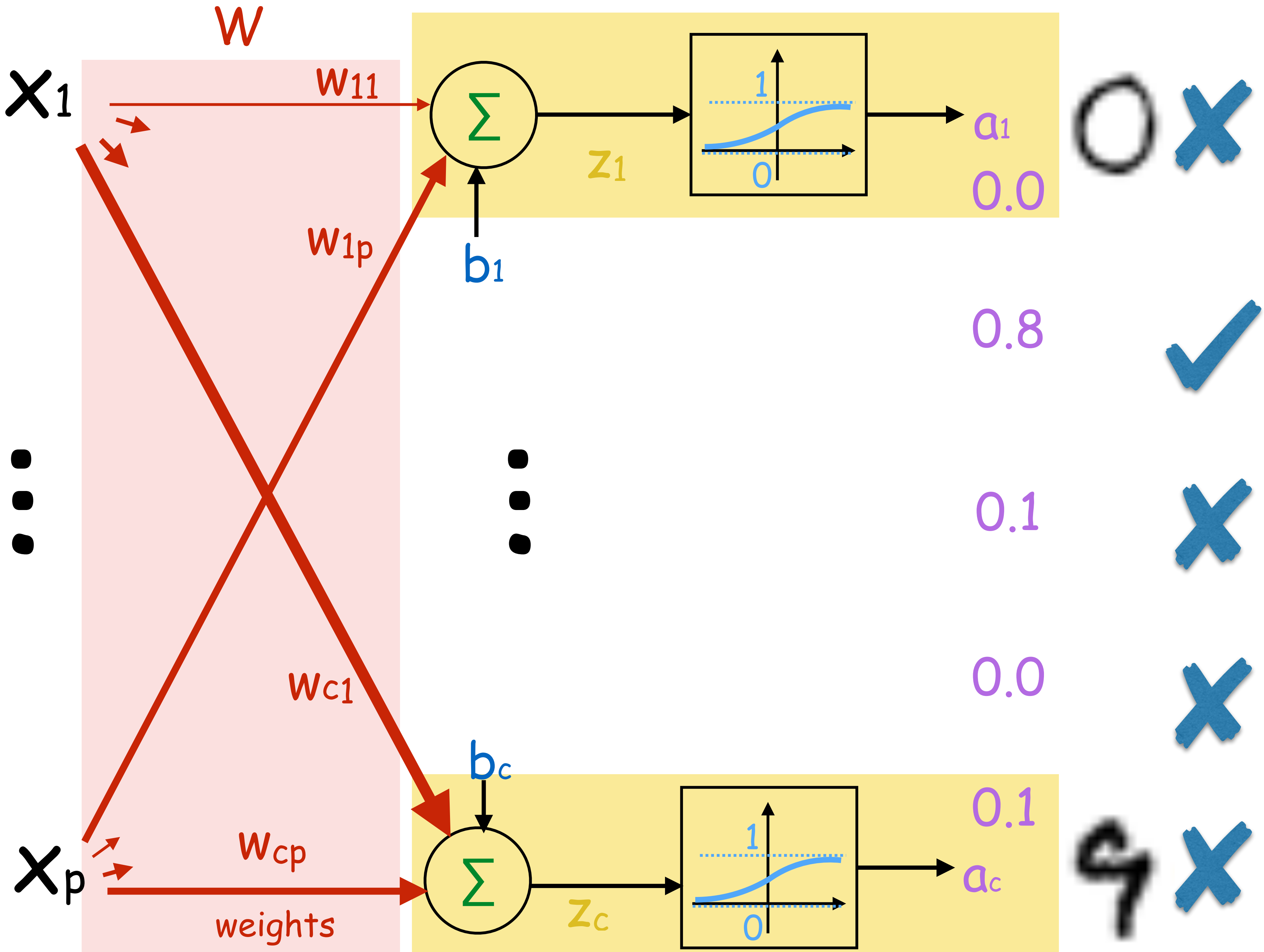
Independent Outputs



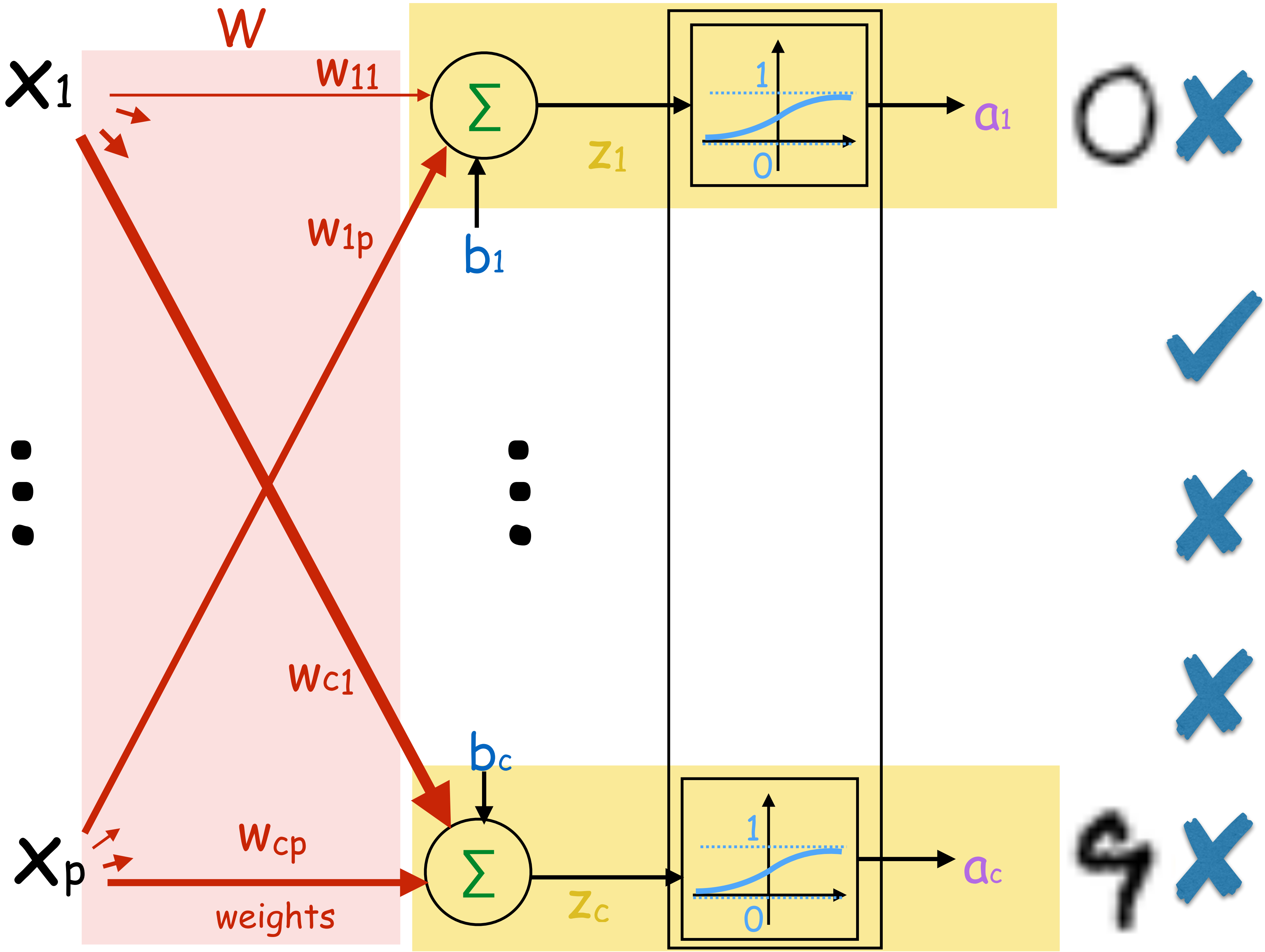
The outputs we need



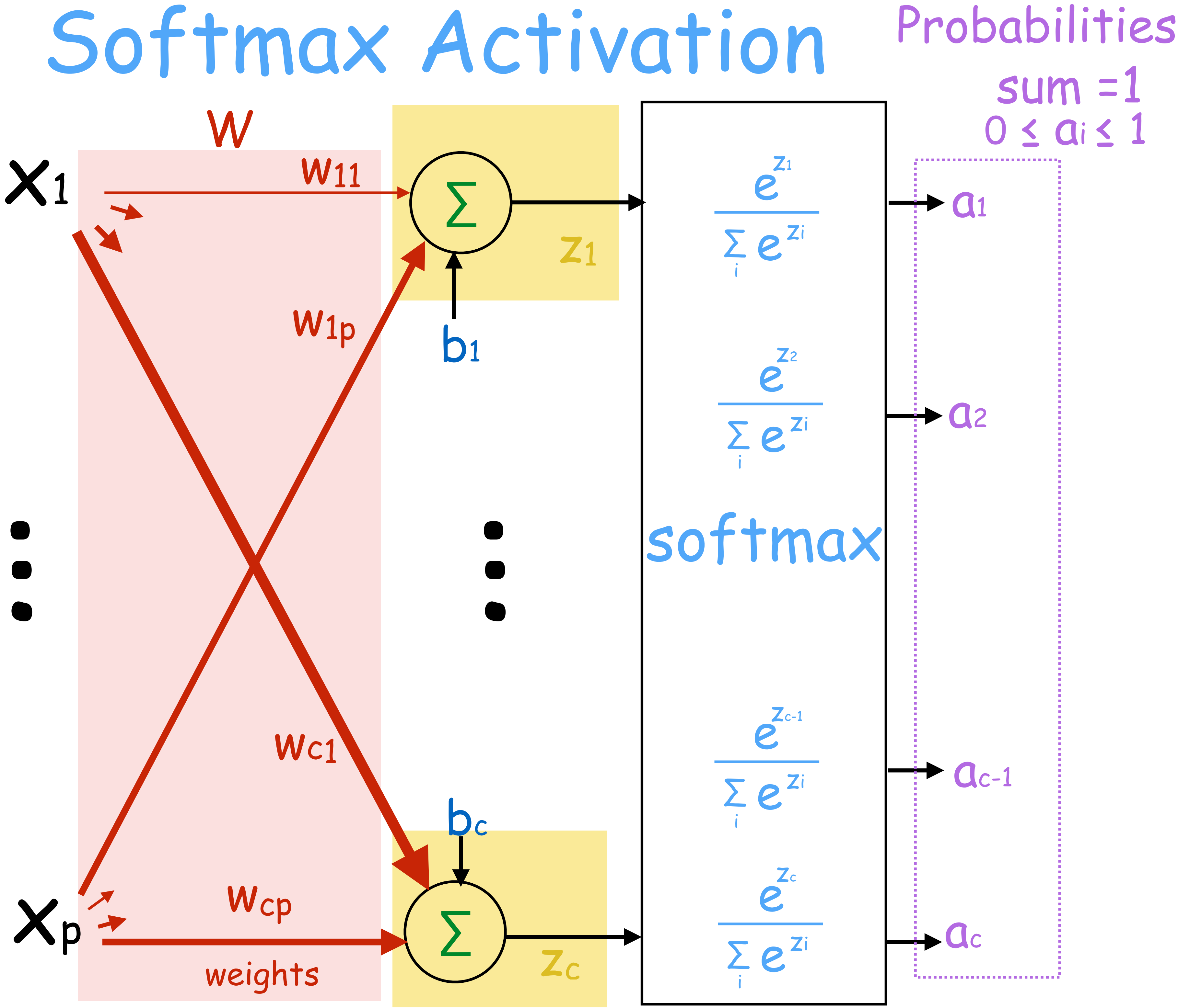
The outputs we need



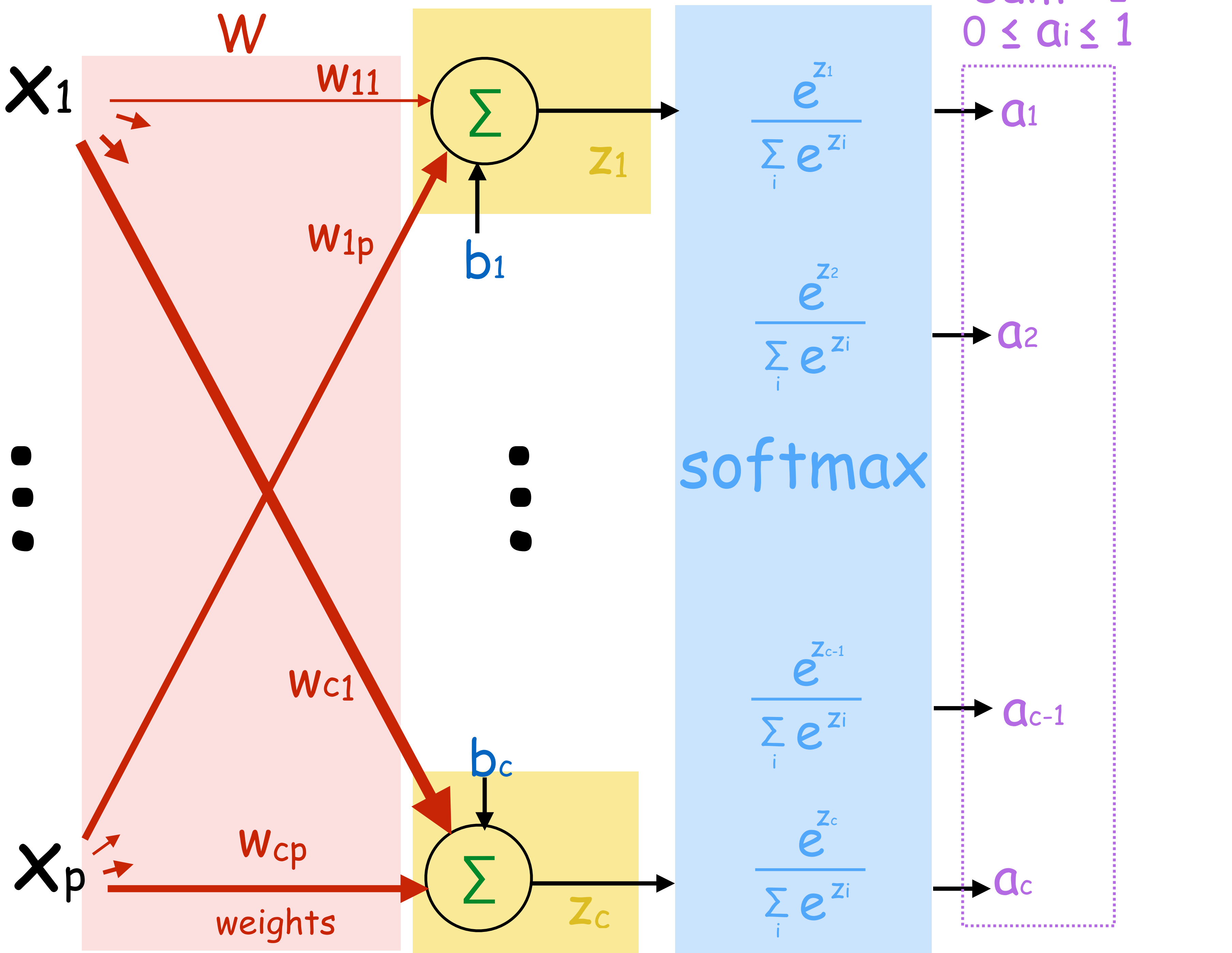
Coordinated Outputs



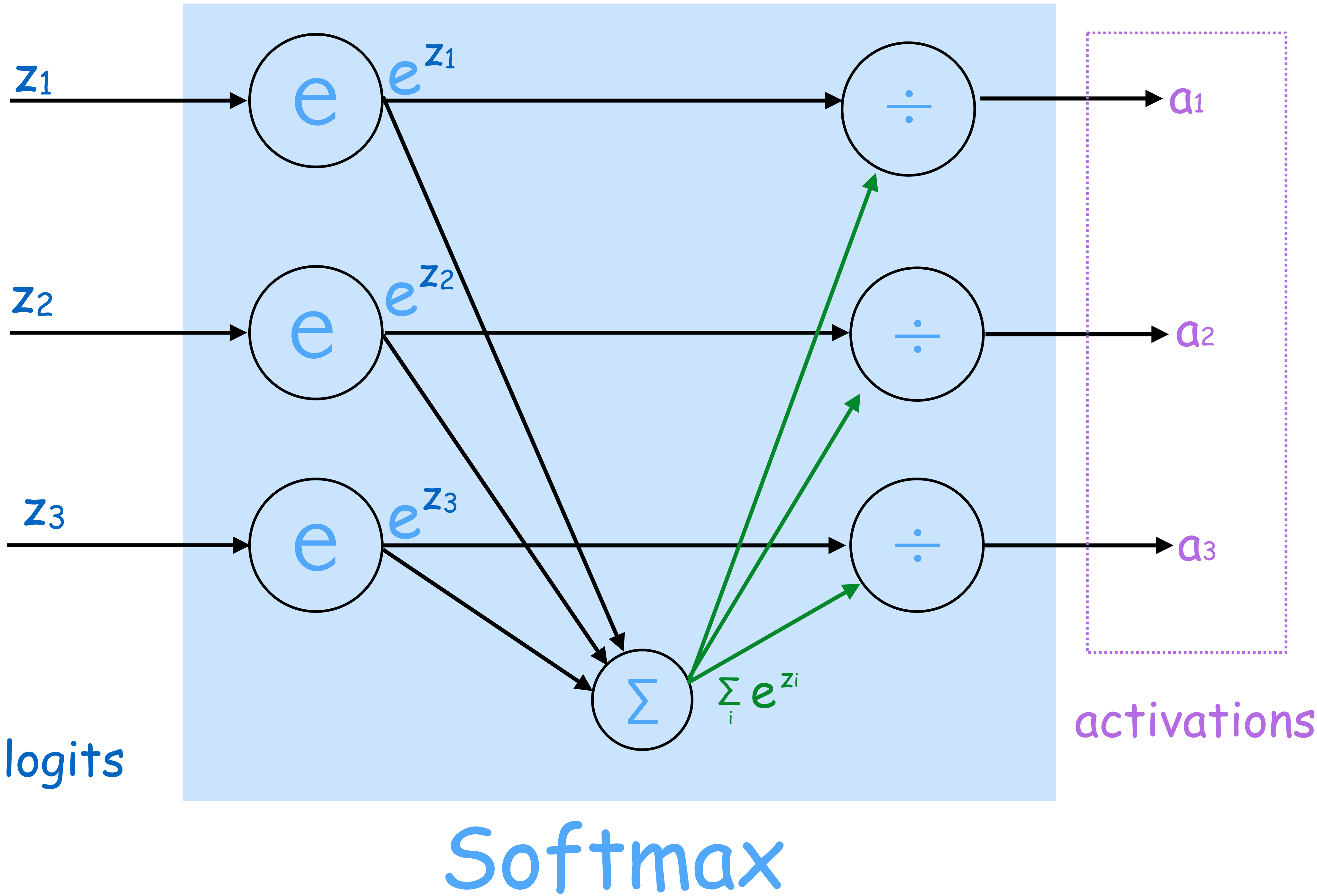
Softmax Activation

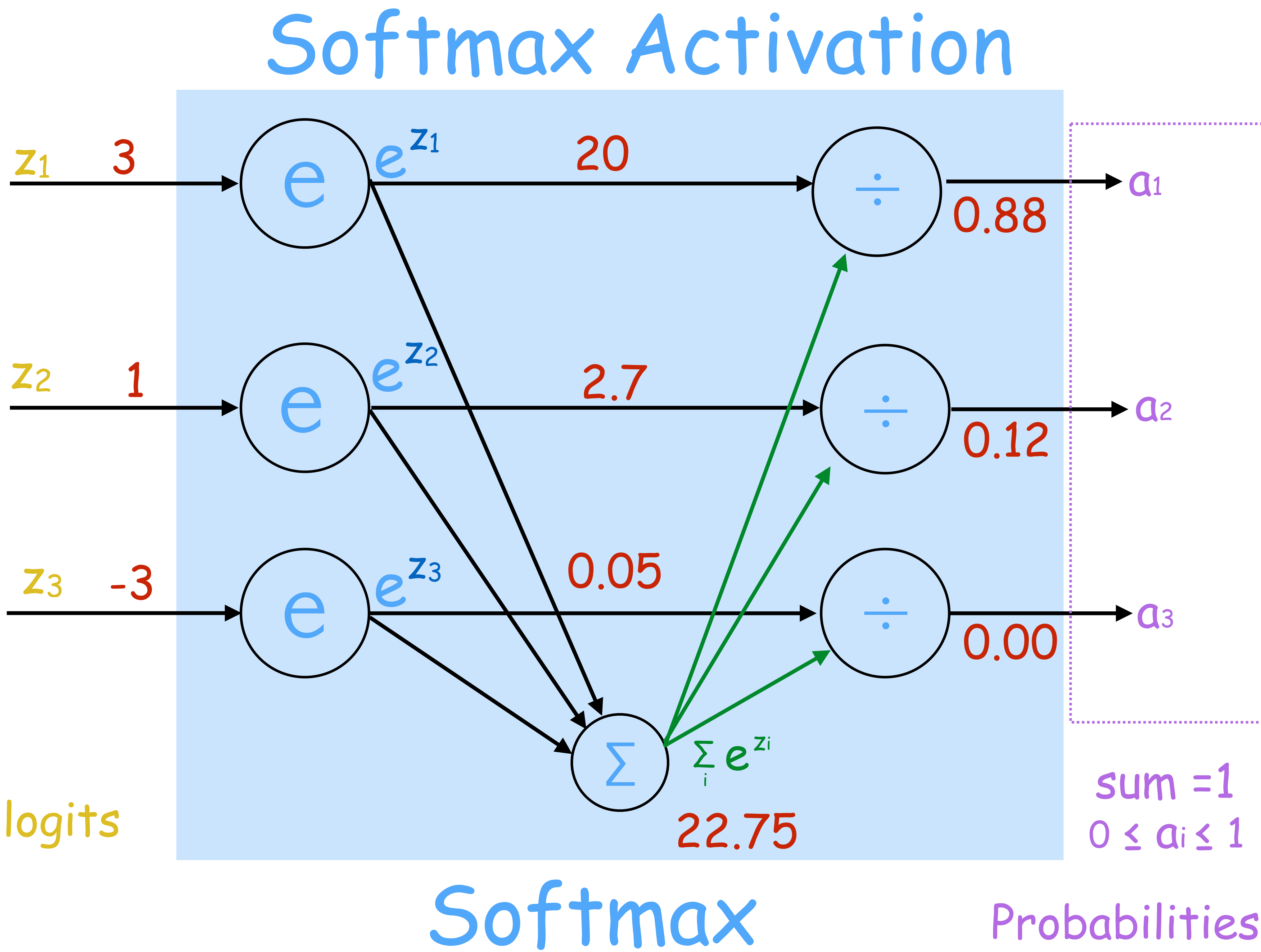


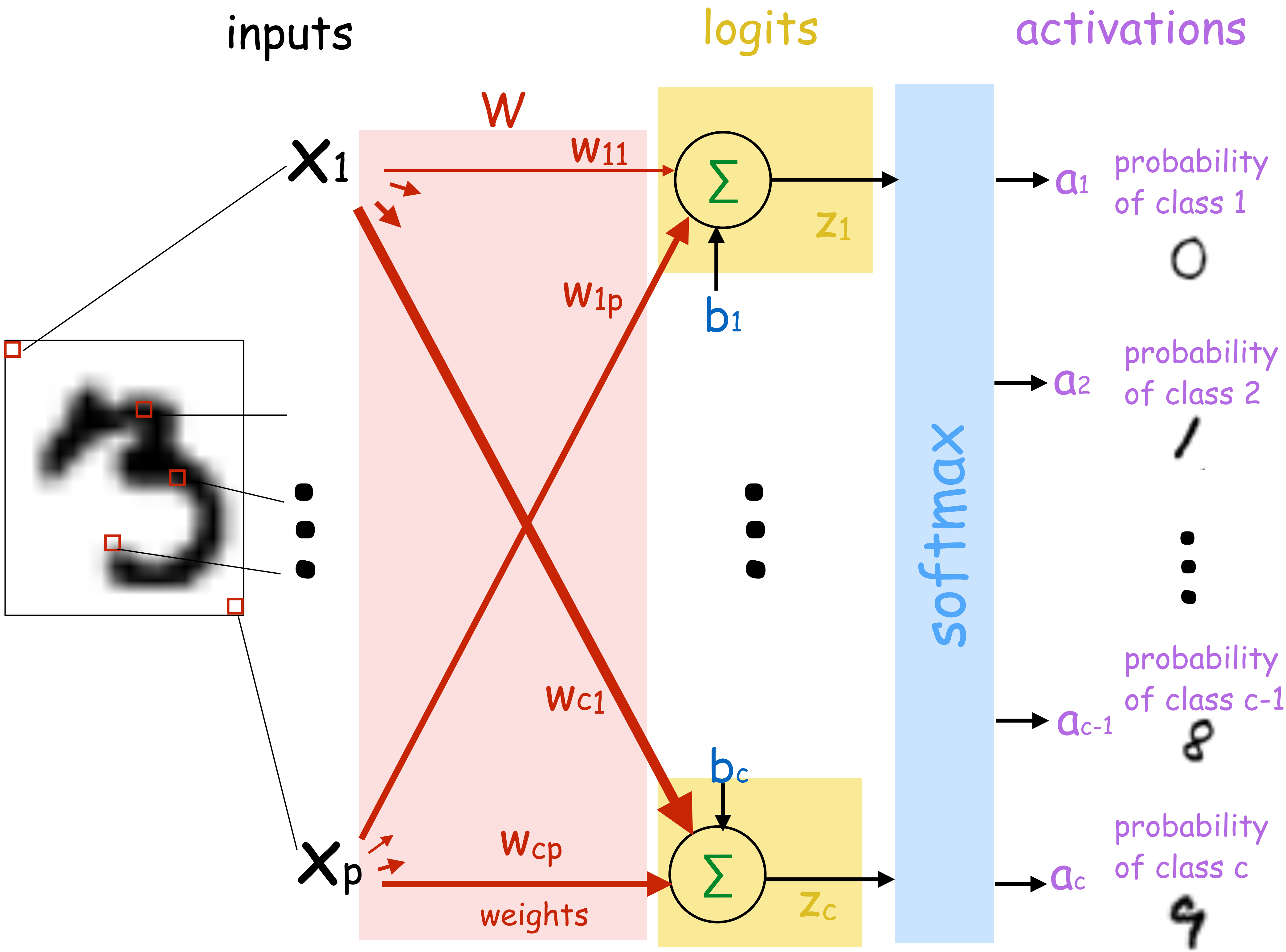
Softmax Activation



Softmax Activation

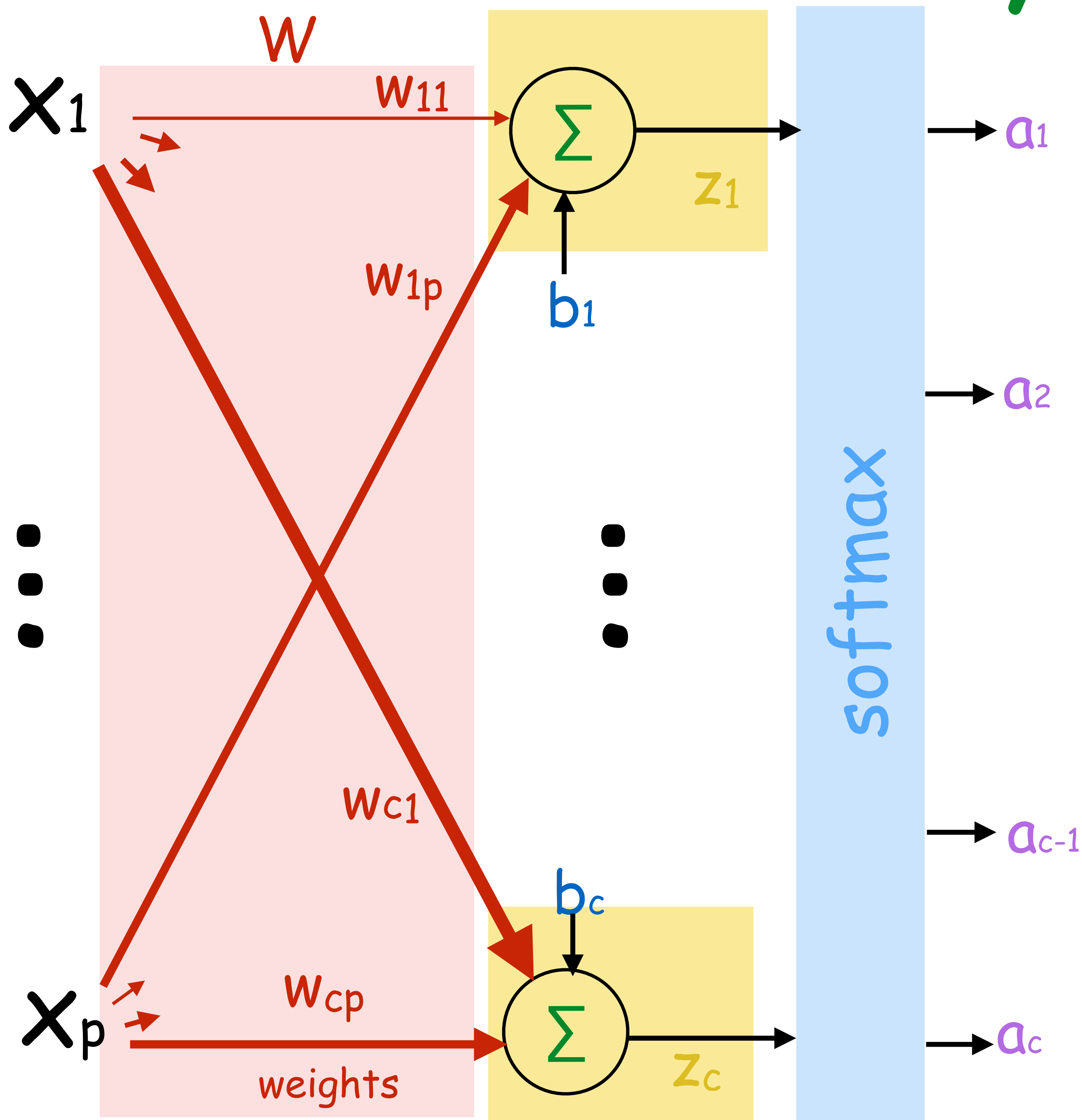



Softmax Activation
z_1
3
z_2
1
z_3
-3
e
e
e
e^{z_1}
e^{z_2}
e^{z_3}
20
2.7
0.05
÷
÷
÷
0.88
0.12
0.00
a_1
a_2
a_3
Σ
$\sum_i e^{z_i}$
22.75
logits
sum =1
$0 \leq a_i \leq 1$
Softmax
Probabilities



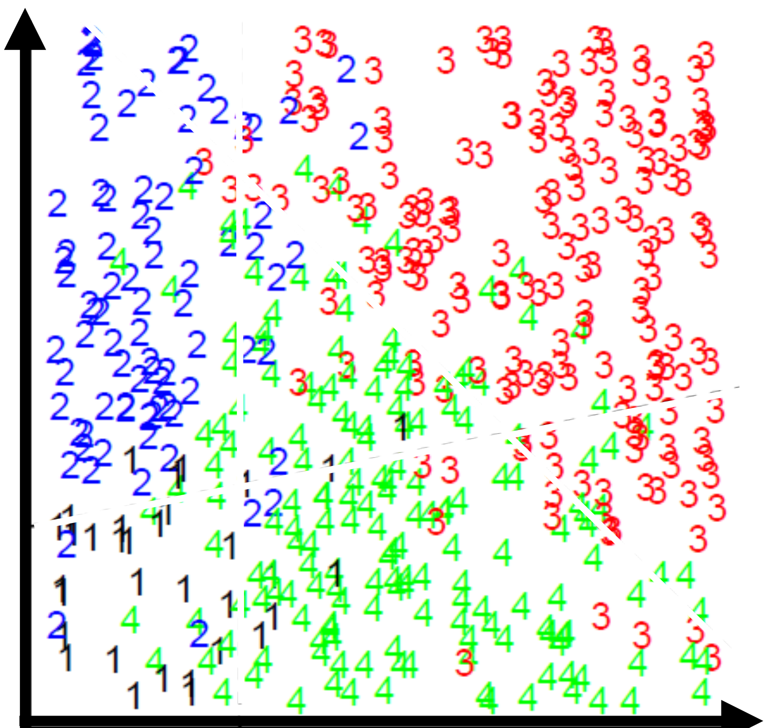
inputs
logits
activations
W
x_1
w_{11}
w_{1p}
z_1
b_1
w_{c1}
b_c
x_p
w_{cp}
weights
z_c
softmax
a_1
probability of class 1
0
a_2
probability of class 2
1
a_{c-1}
probability of class c-1
8
a_c
probability of class c
9

Parameters W, b



Training Model

training set

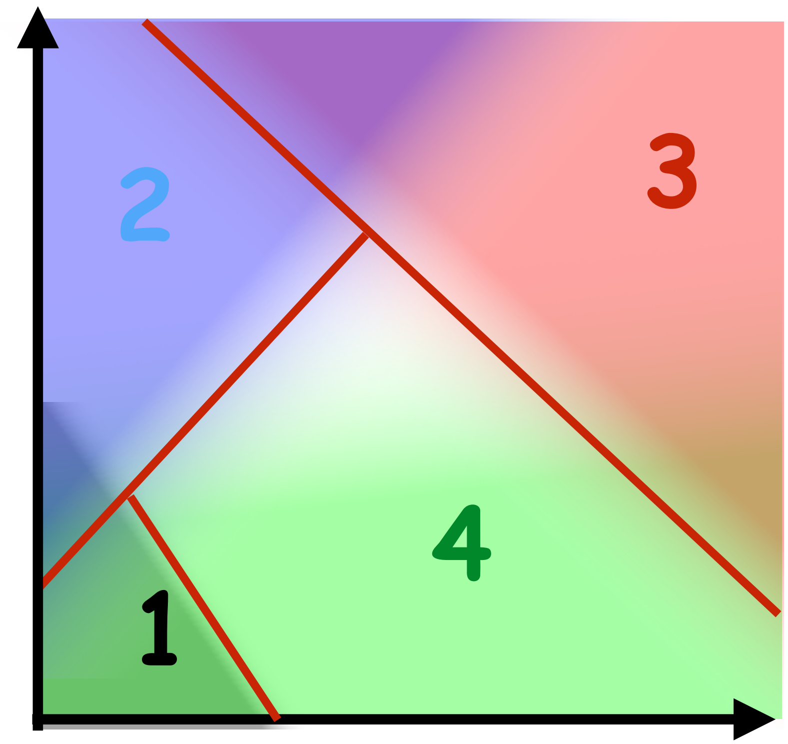


Feature Matrix $X_{(n \text{ by } p)}$

Label Vector Y

(length n) (0, 1, 2,... value)

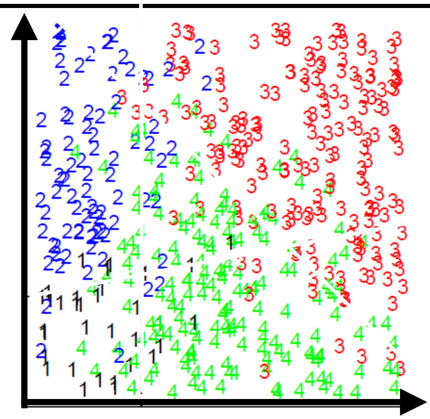
learn parameters



Weight Matrix $\mathbf{W}_{(\text{shape } c \text{ by } p)}$

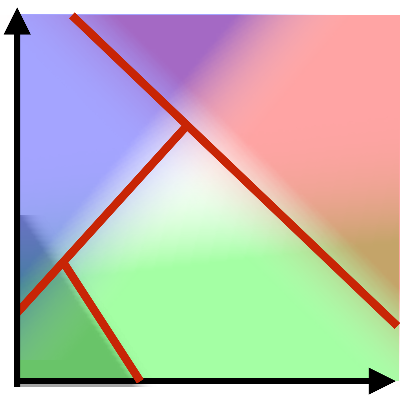
bias vector b (length c)

negative log Likelihood



Label: y 3 y 1

observed data



Weights **w**

biases b

Outputs:

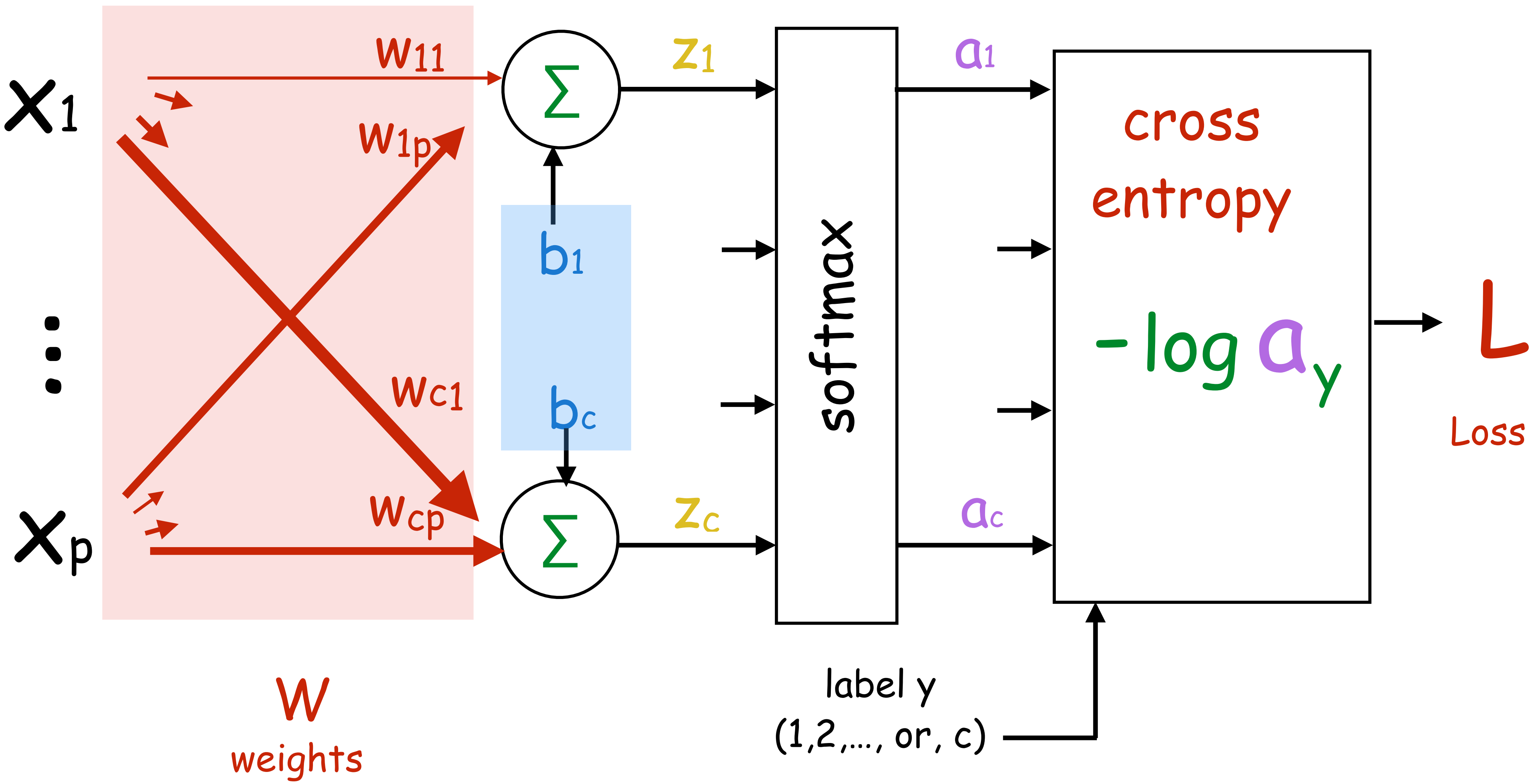
probabilities

a_1	.4	$\boxed{a_1 \quad .4}$	
a_2	.2	a_2	.2
$\boxed{a_3 \quad .1}$		a_3	.1
a_c	.1	a_c	.1

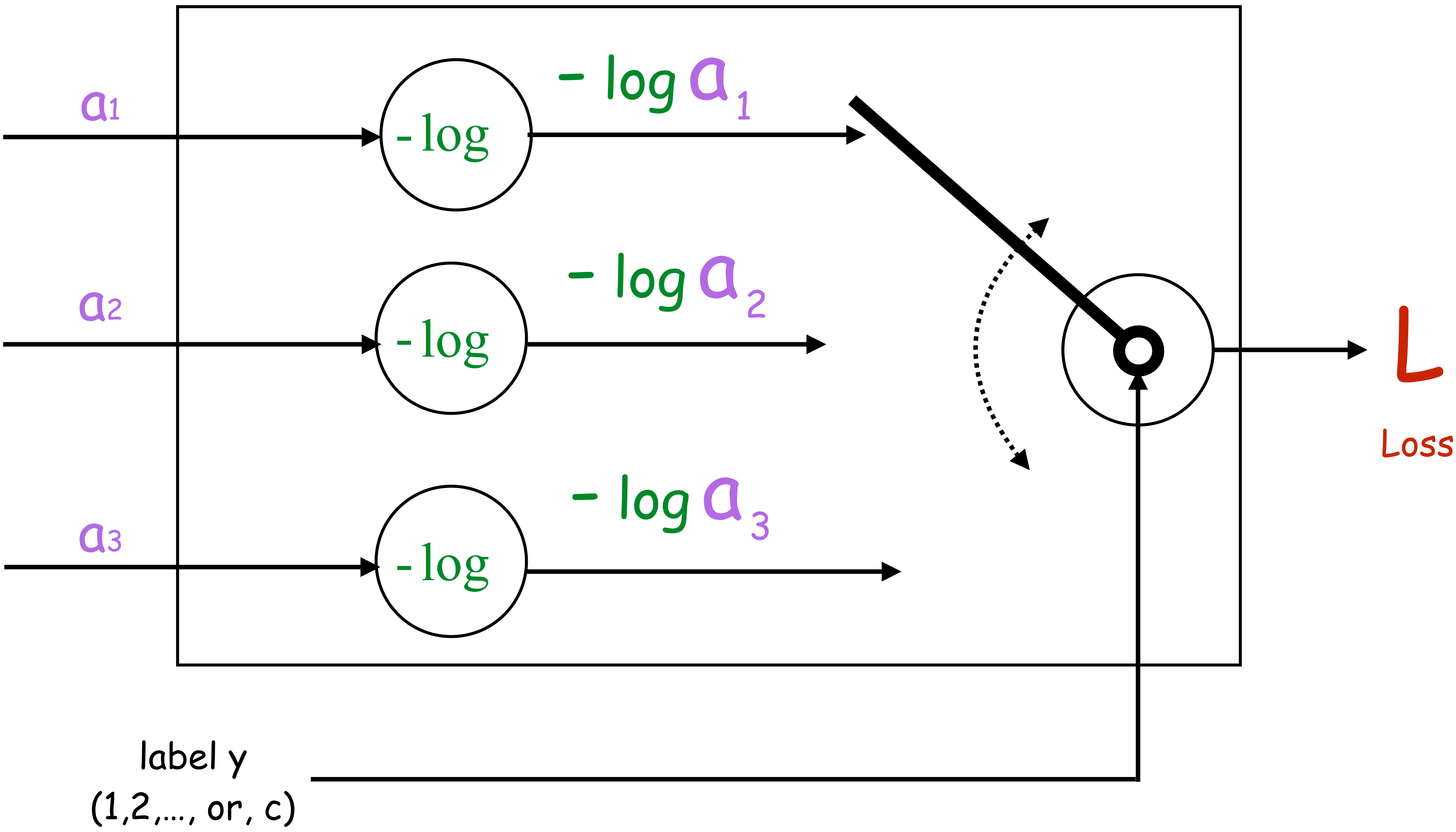
$$- \log \text{Likelihood} = -\log a_y$$

Multi-class cross entropy loss

Multi-Class Cross Entropy Loss



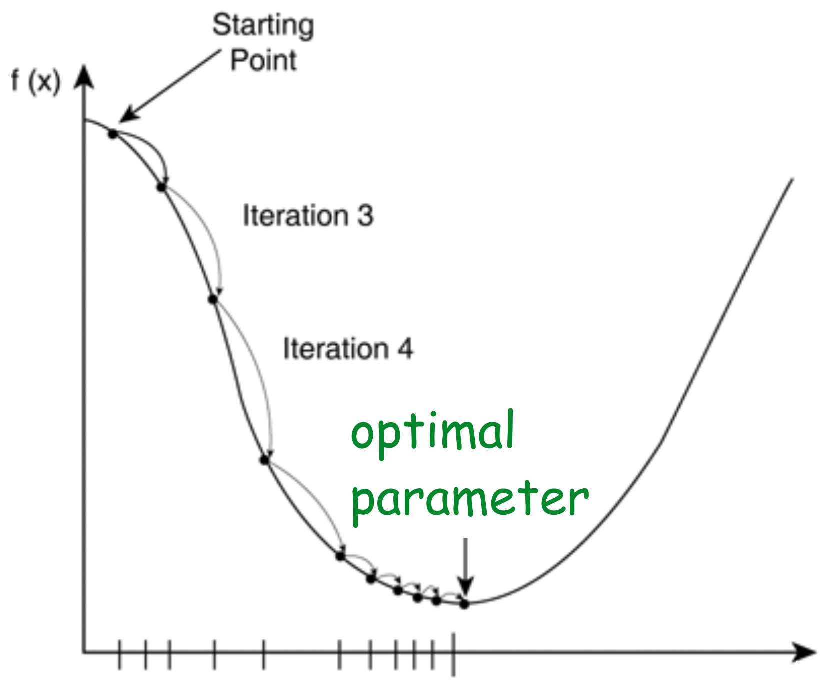
Multi-Class Cross Entropy Loss



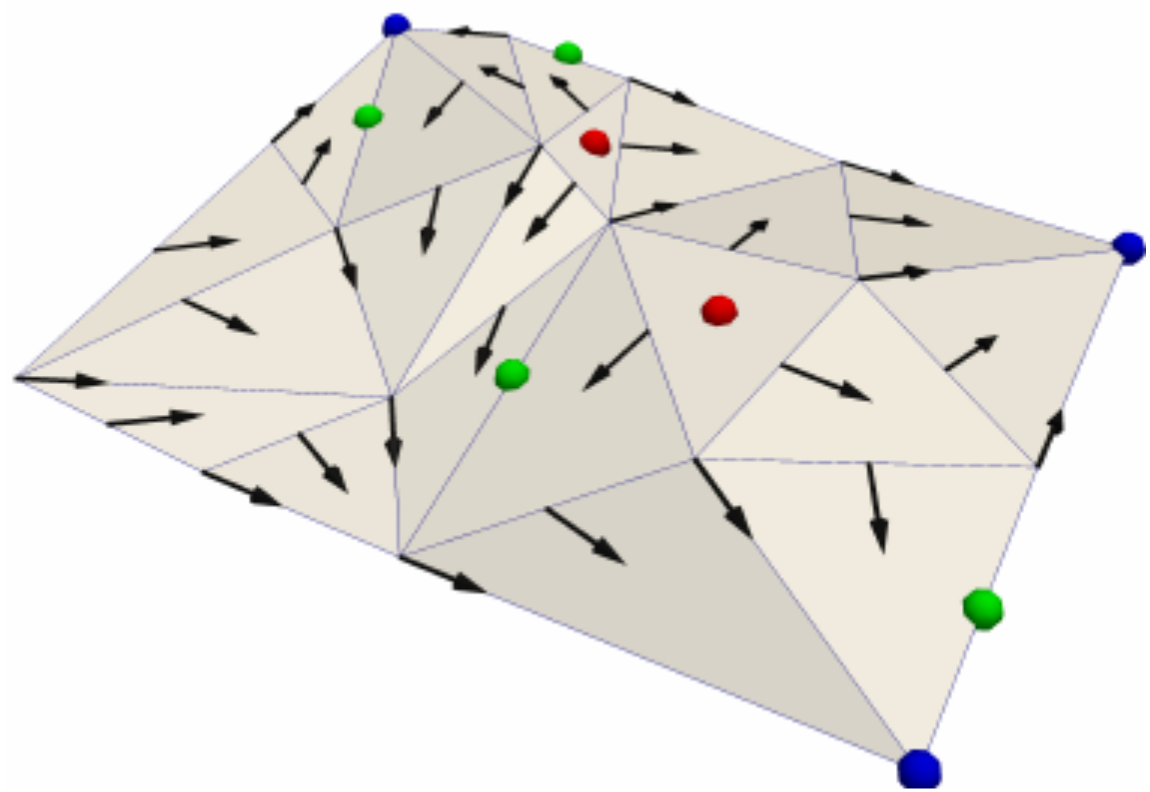
$$-\log a_y$$

Gradient Descent

L= loss (w)



parameter value



$$W \longleftarrow W - a \frac{\partial L}{\partial W}$$

$$b \longleftarrow b - a \frac{\partial L}{\partial b}$$

a step size (a constant scalar)

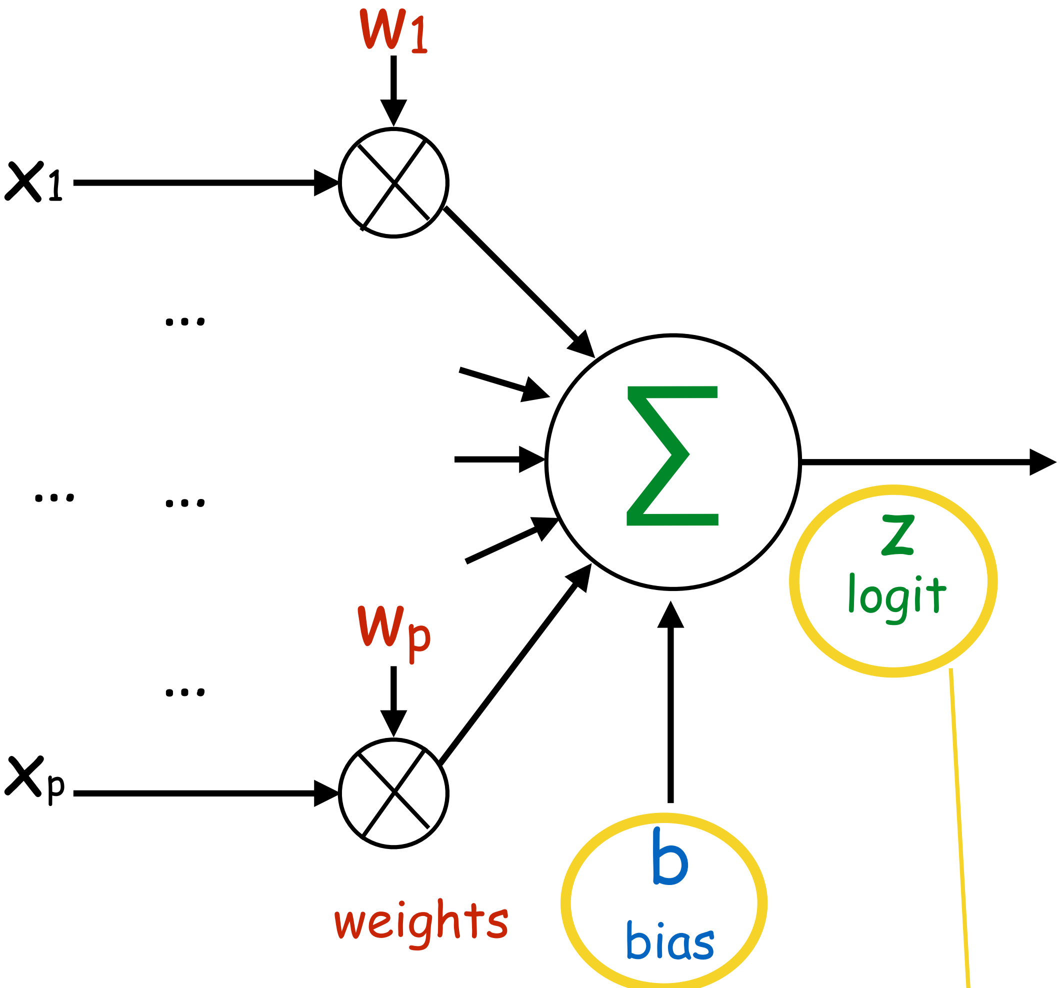
Gradient of L w.r.t. a **vector**?

$$\frac{\partial L}{\partial b}$$

Gradient of L w.r.t. a **matrix**?

$$\frac{\partial L}{\partial \mathbf{W}}$$

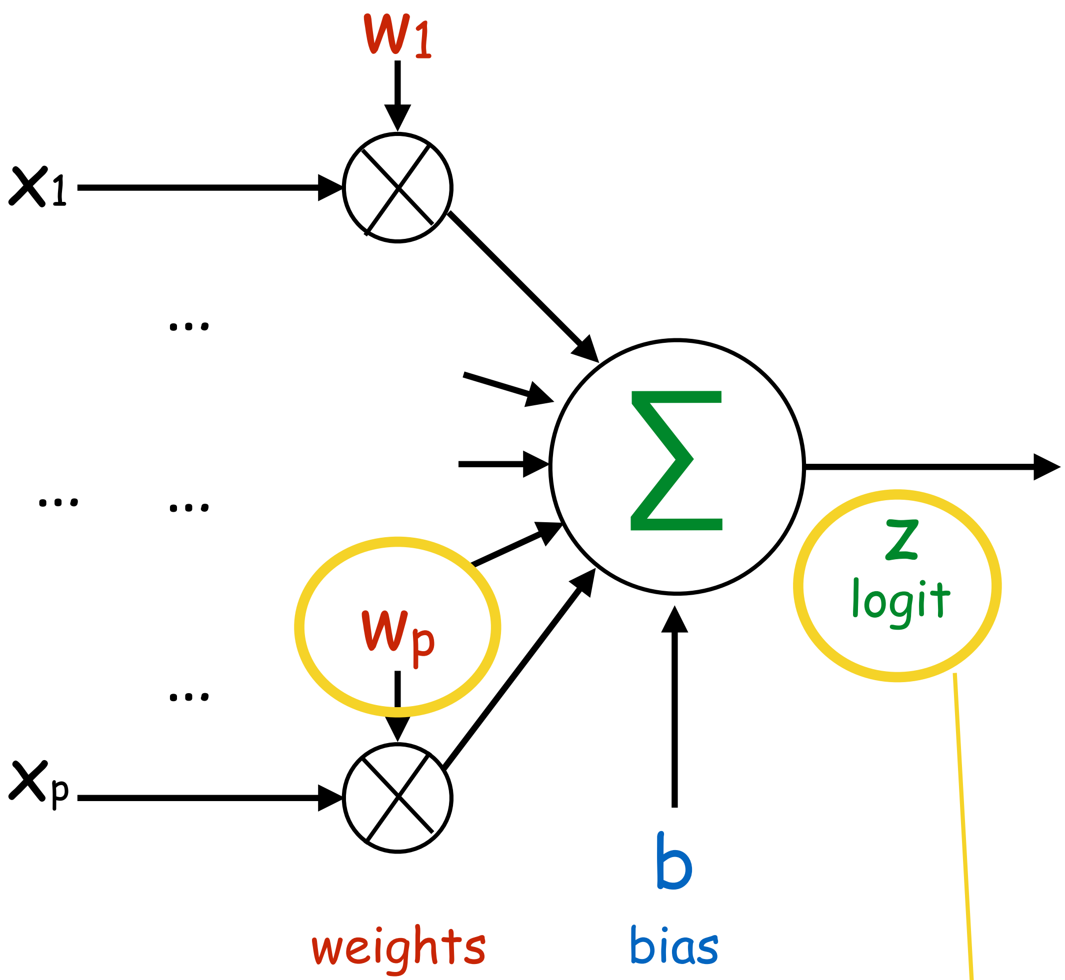
Example



gradient

$$\frac{\partial z}{\partial b} = 1$$

Example

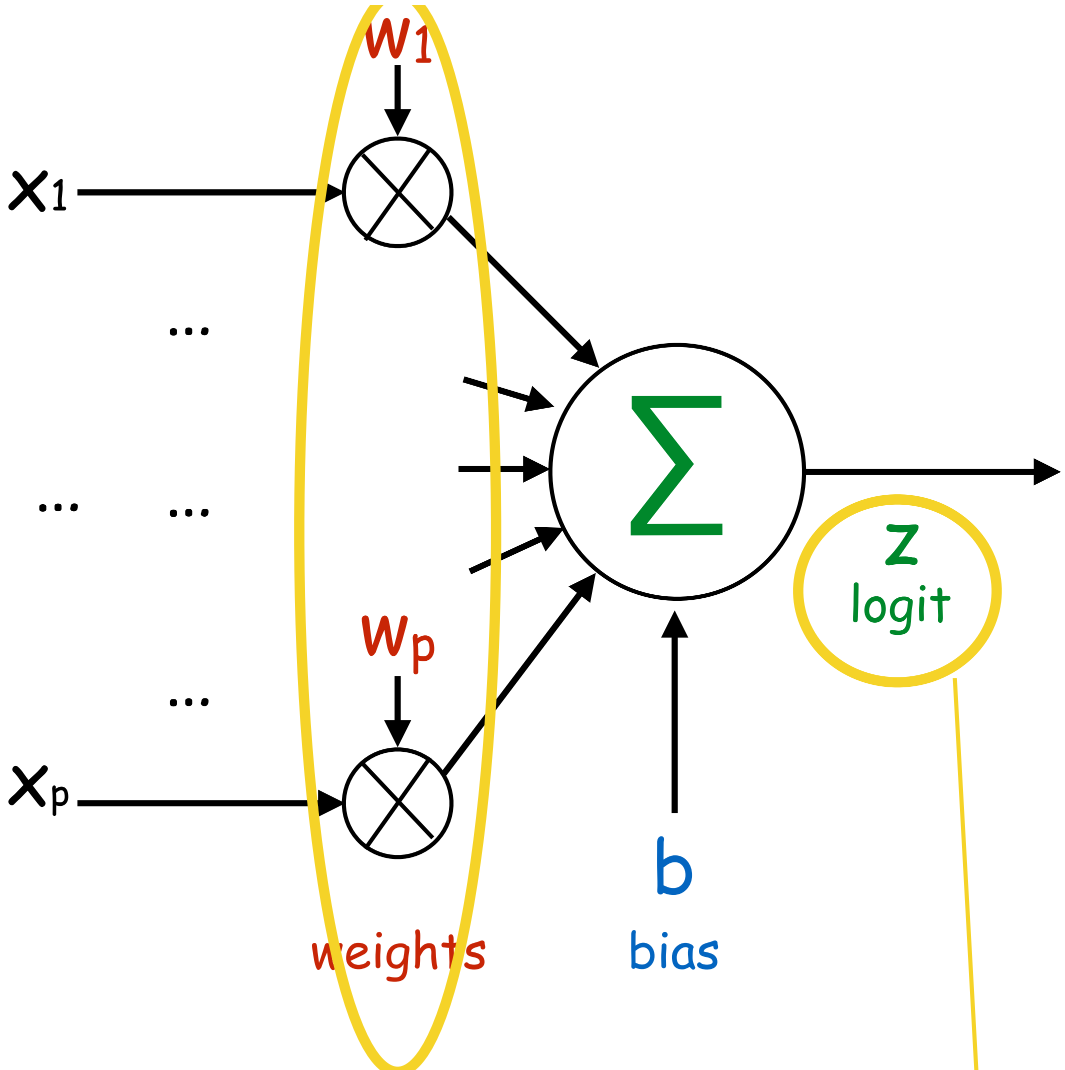


gradient

$$\frac{\partial z}{\partial w_p} = x_p$$

is a function of x_p

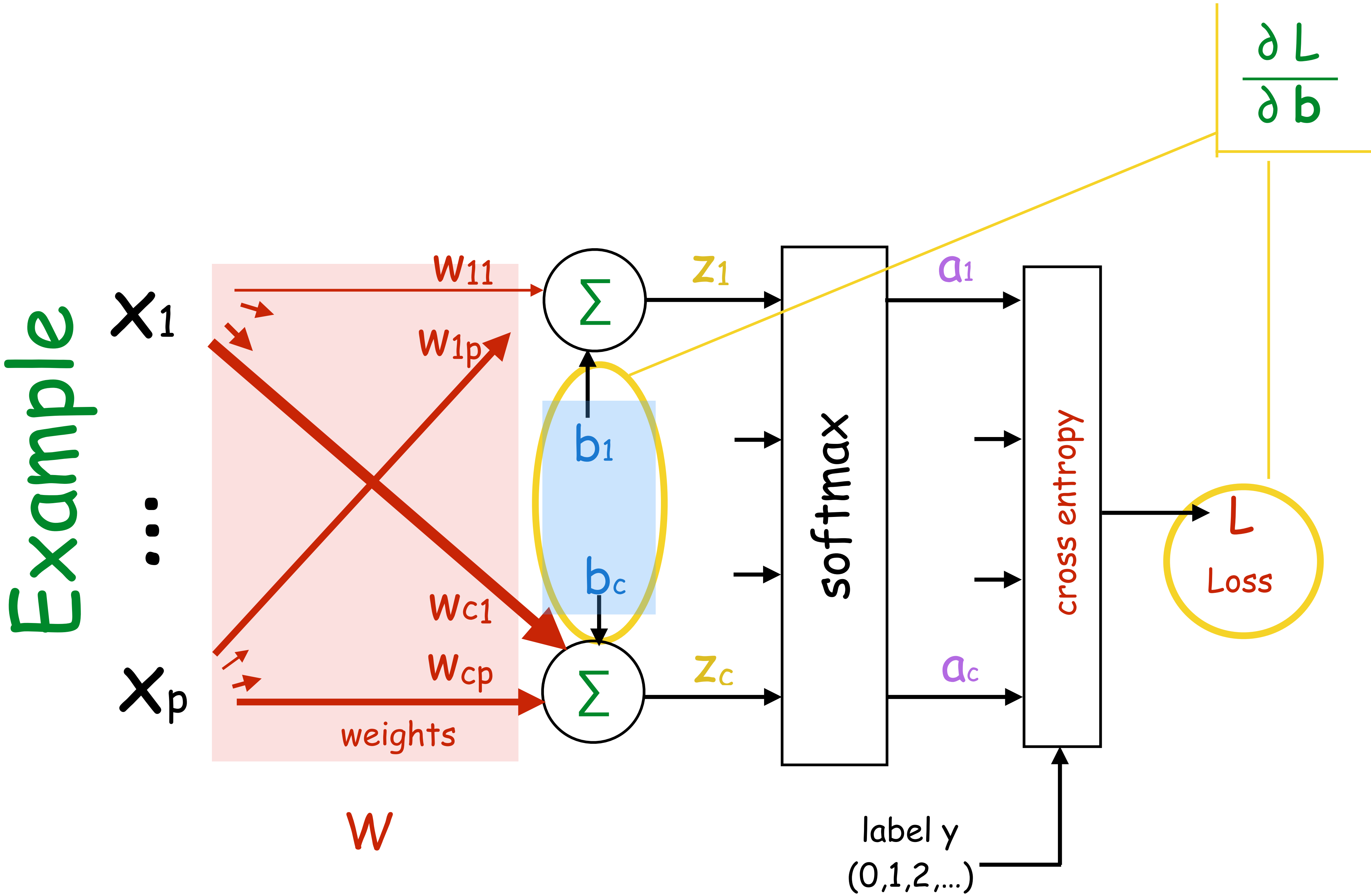
Example



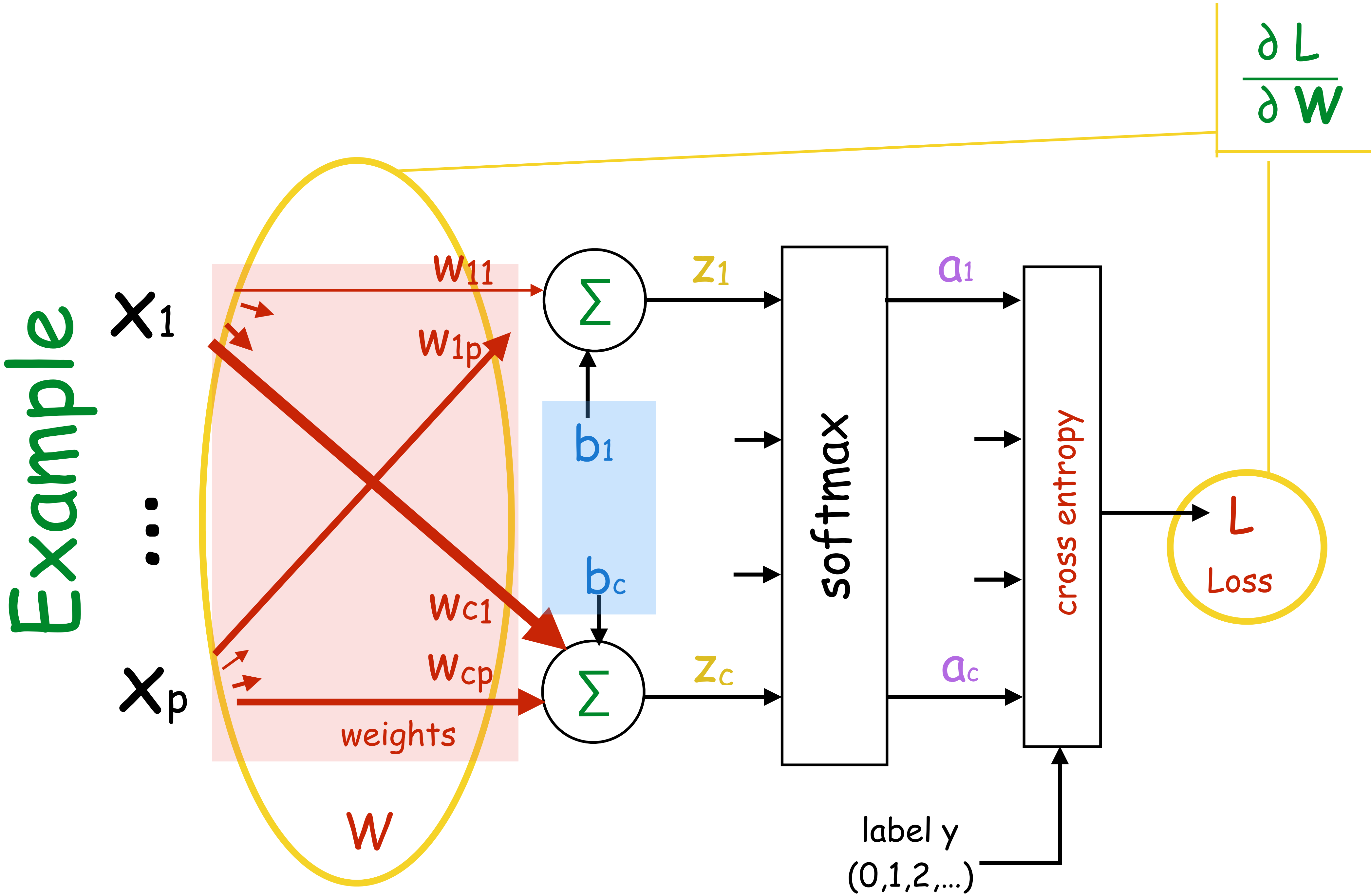
gradient

$$\frac{\partial z}{\partial \mathbf{w}} = \left(\frac{\partial z}{\partial w_1}, \frac{\partial z}{\partial w_2}, \ldots, \frac{\partial z}{\partial w_p}\right)$$

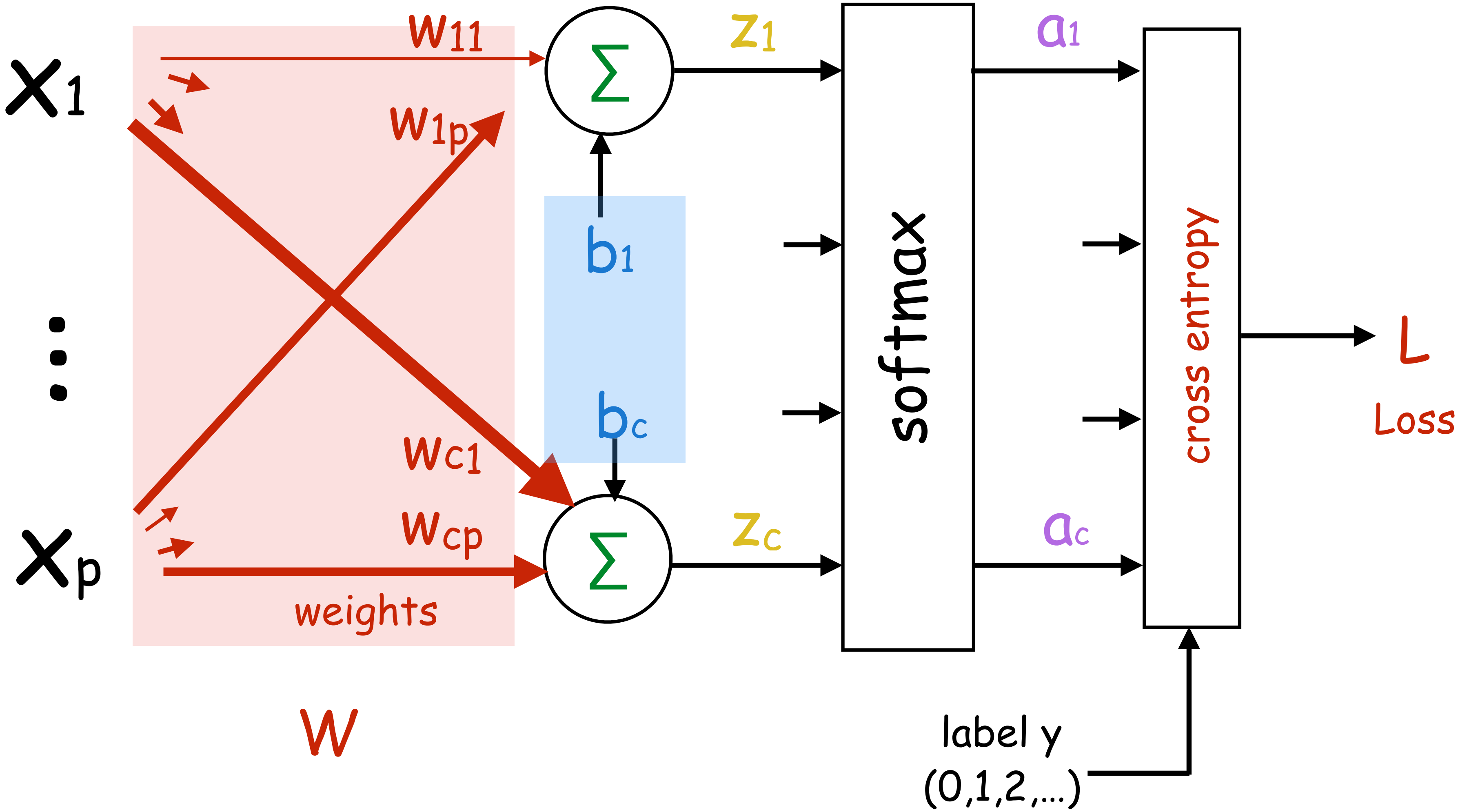
$$= (x_1, x_2, \ldots, x_p) = \mathbf{x}$$



∂L/∂b
Example
x1
xp
w11
w1p
wc1
wcp
weights
W
b1
bc
z1
zc
softmax
a1
ac
cross entropy
L
Loss
label y
(0,1,2,...)



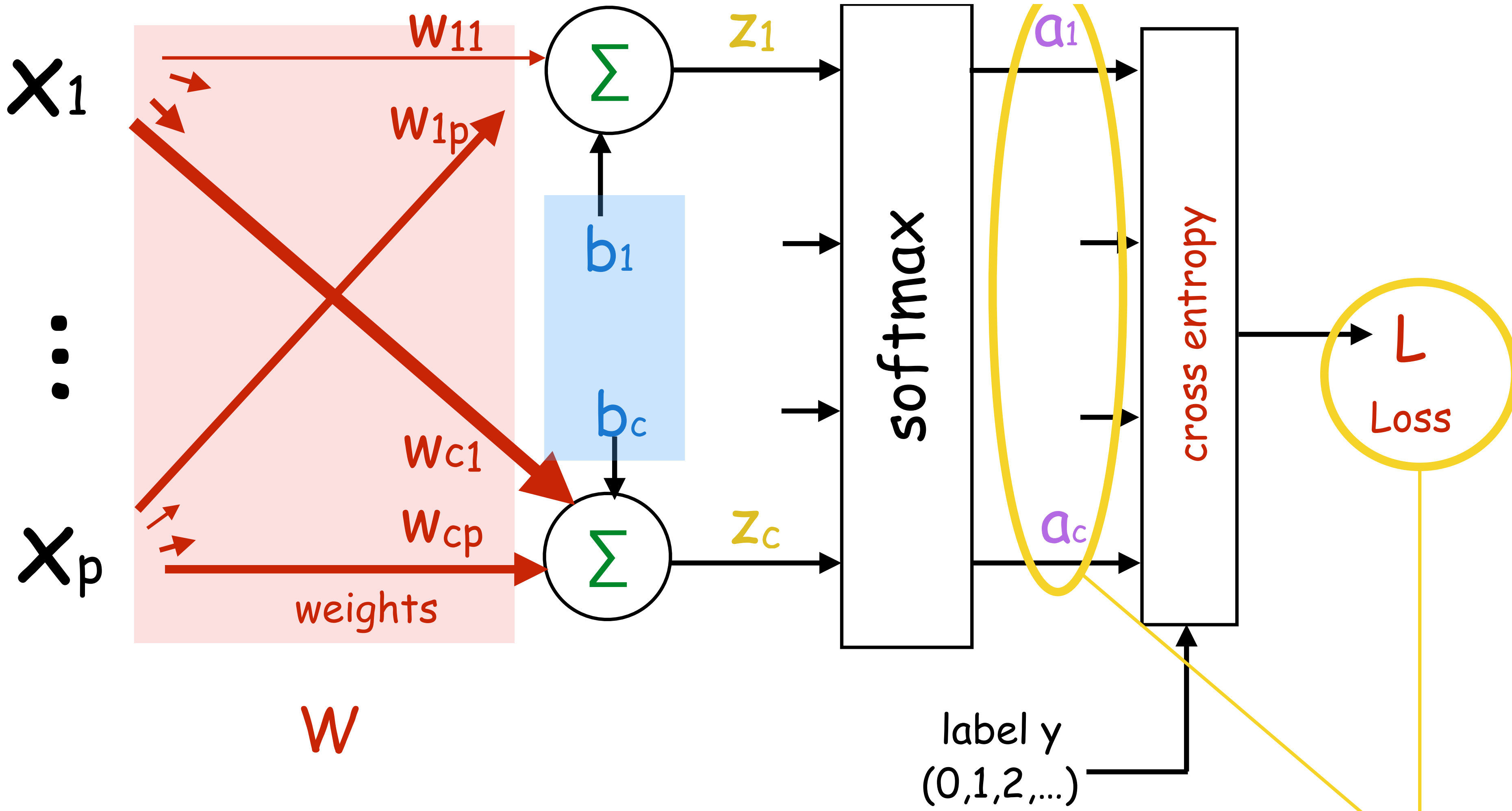
Example
x1
xp
w11
w1p
wc1
wcp
weights
W
b1
bc
z1
zc
softmax
a1
ac
cross entropy
L
Loss
label y
(0,1,2,...)
∂L
∂W



$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial W}$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial b}$$

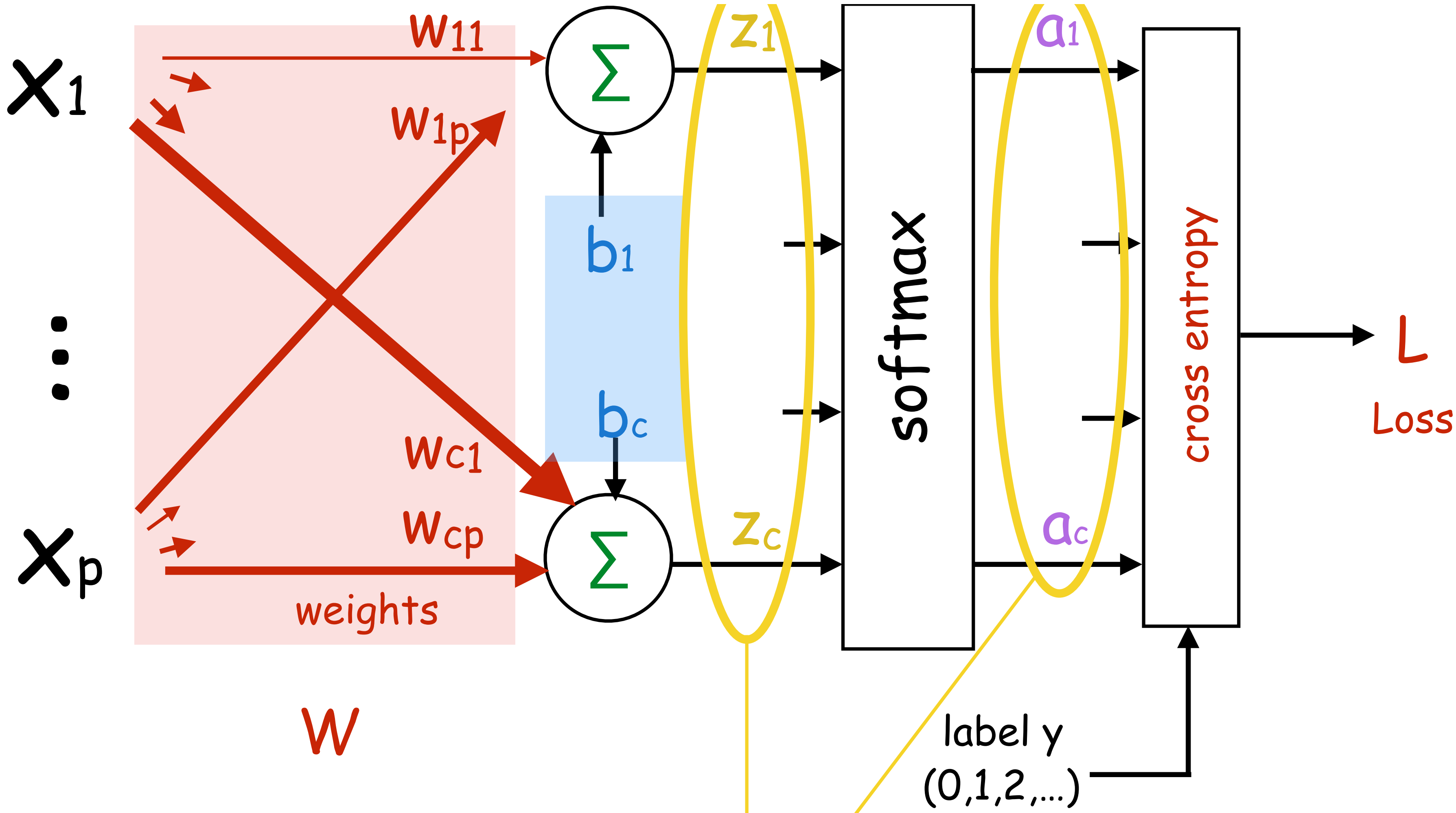
Chain Rule



$$\frac{\partial L}{\partial a} = \left(\frac{\partial L}{\partial a_1}, \ldots, \frac{\partial L}{\partial a_y}, \ldots, \frac{\partial L}{\partial a_c}\right)$$

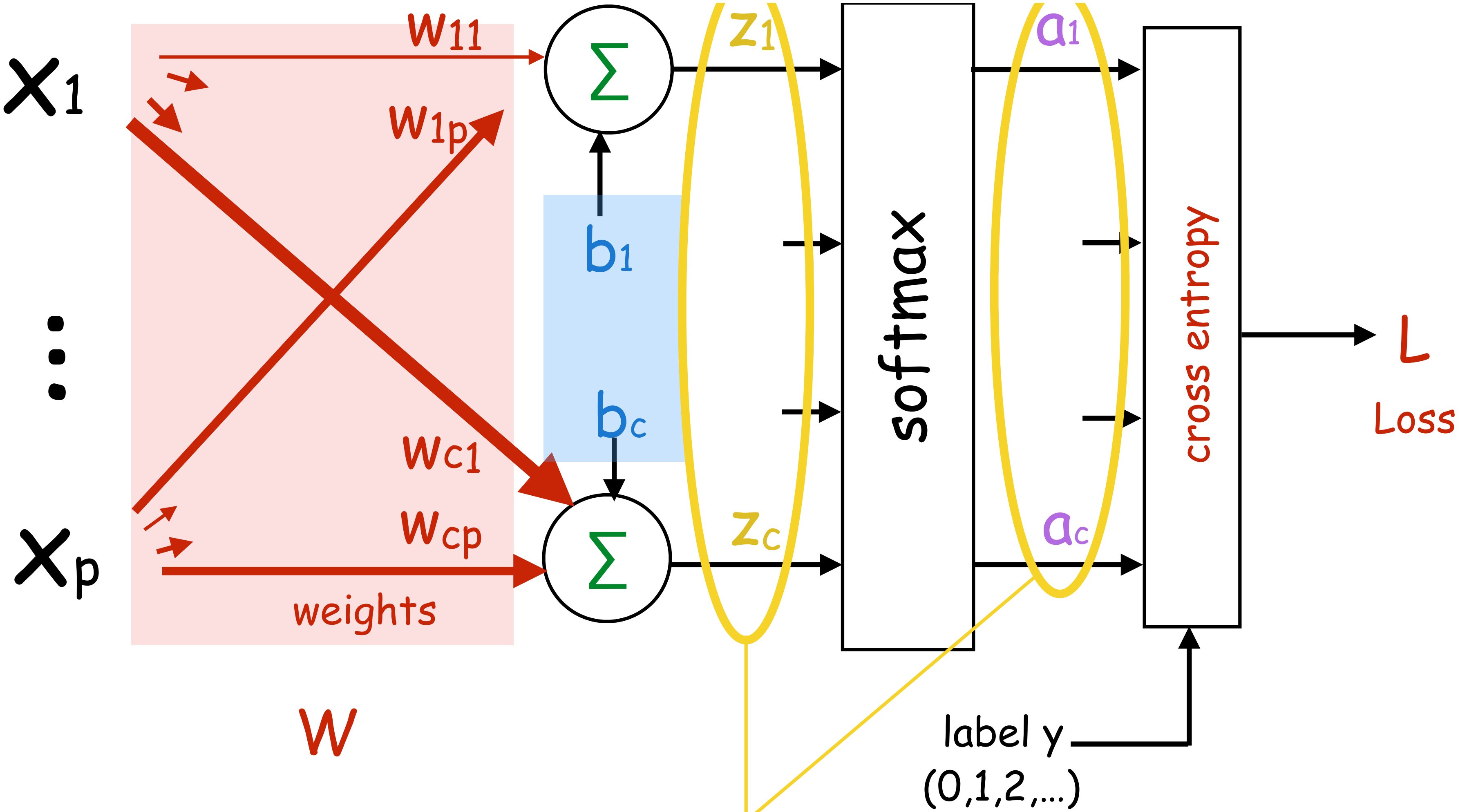
$$= \left(0, \ldots, -\frac{1}{a_y}, \ldots, 0\right)$$

y



$$\frac{\partial \mathbf{a}}{\partial \mathbf{z}} = ?$$

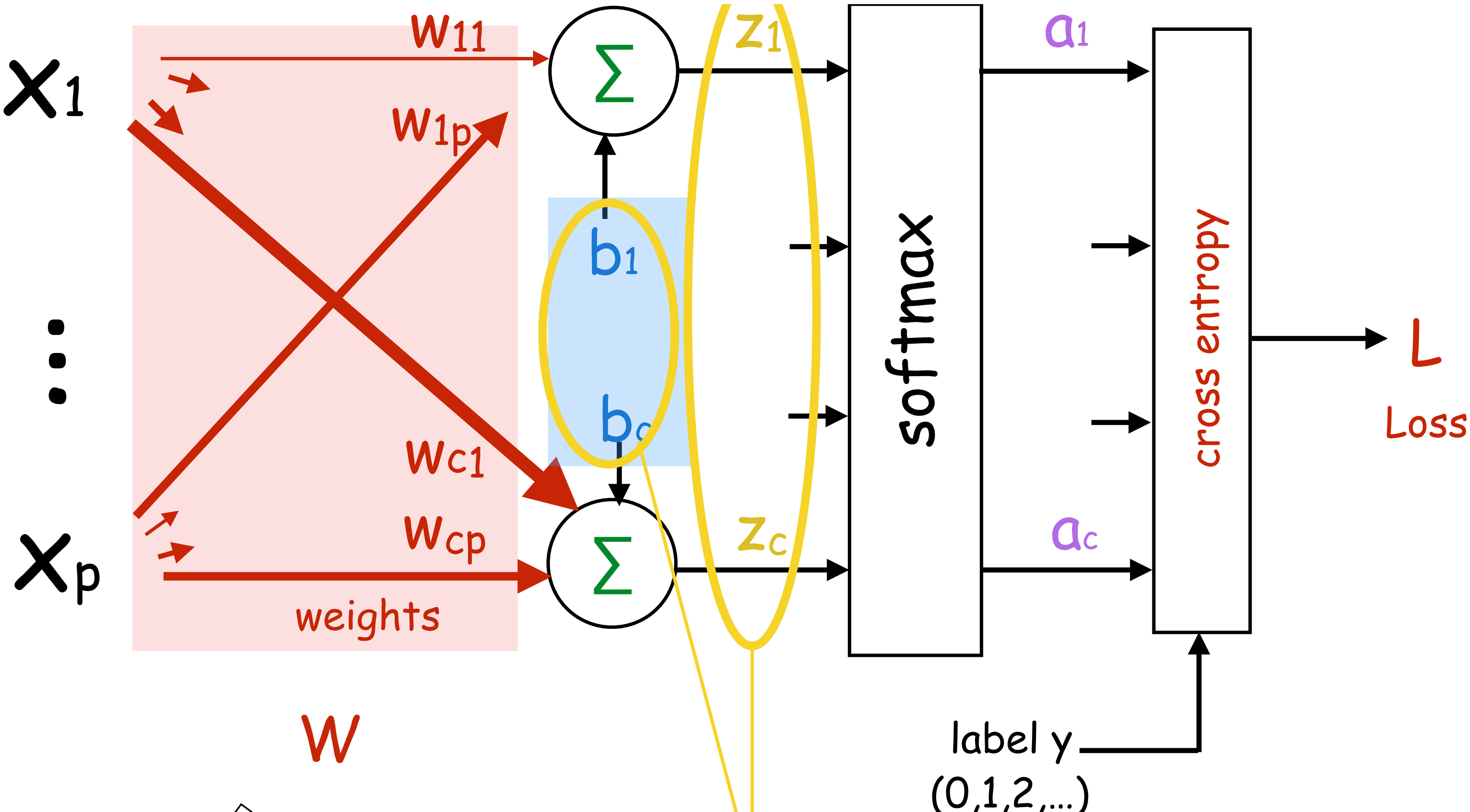
Gradient of a vector w.r.t. a vector !!!



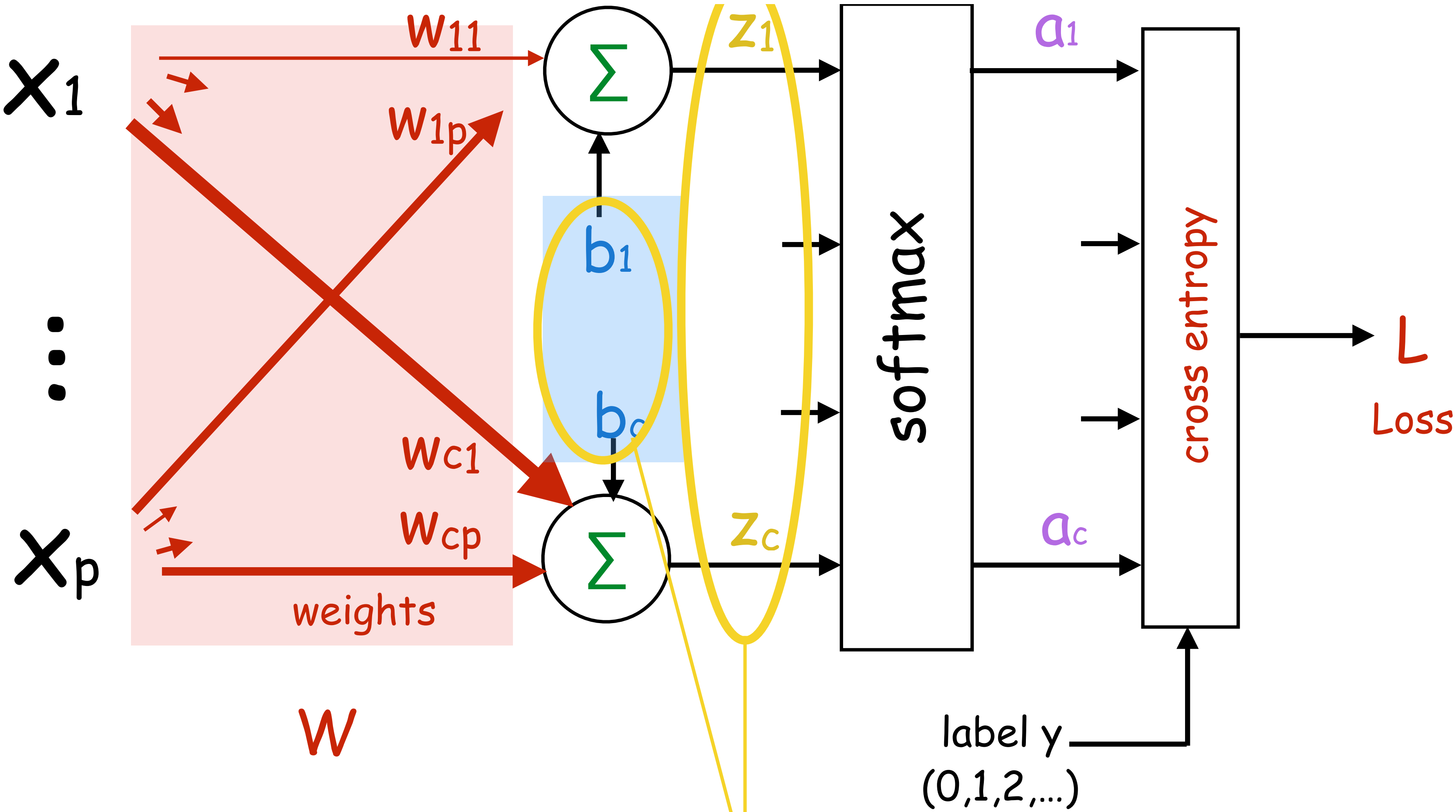
$$\frac{\partial a}{\partial z} = \begin{bmatrix} \frac{\partial a_1}{\partial z_1} & \cdots & \frac{\partial a_1}{\partial z_c} \\ \vdots & & \vdots \\ \frac{\partial a_c}{\partial z_1} & & \frac{\partial a_c}{\partial z_c} \end{bmatrix}$$

$$\frac{\partial a_i}{\partial z_j} = \begin{cases} a_i(1 - a_i) & \text{if } i=j \\ -a_i a_j & \text{if } i \neq j \end{cases}$$

https://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/



$$\frac{\partial z}{\partial b} = \begin{bmatrix} 1 & & \cdots & 0 \\ & 1 & & \\ \vdots & & \ddots & \vdots \\ & & 1 & \\ 0 & \cdots & & 1 \end{bmatrix}$$

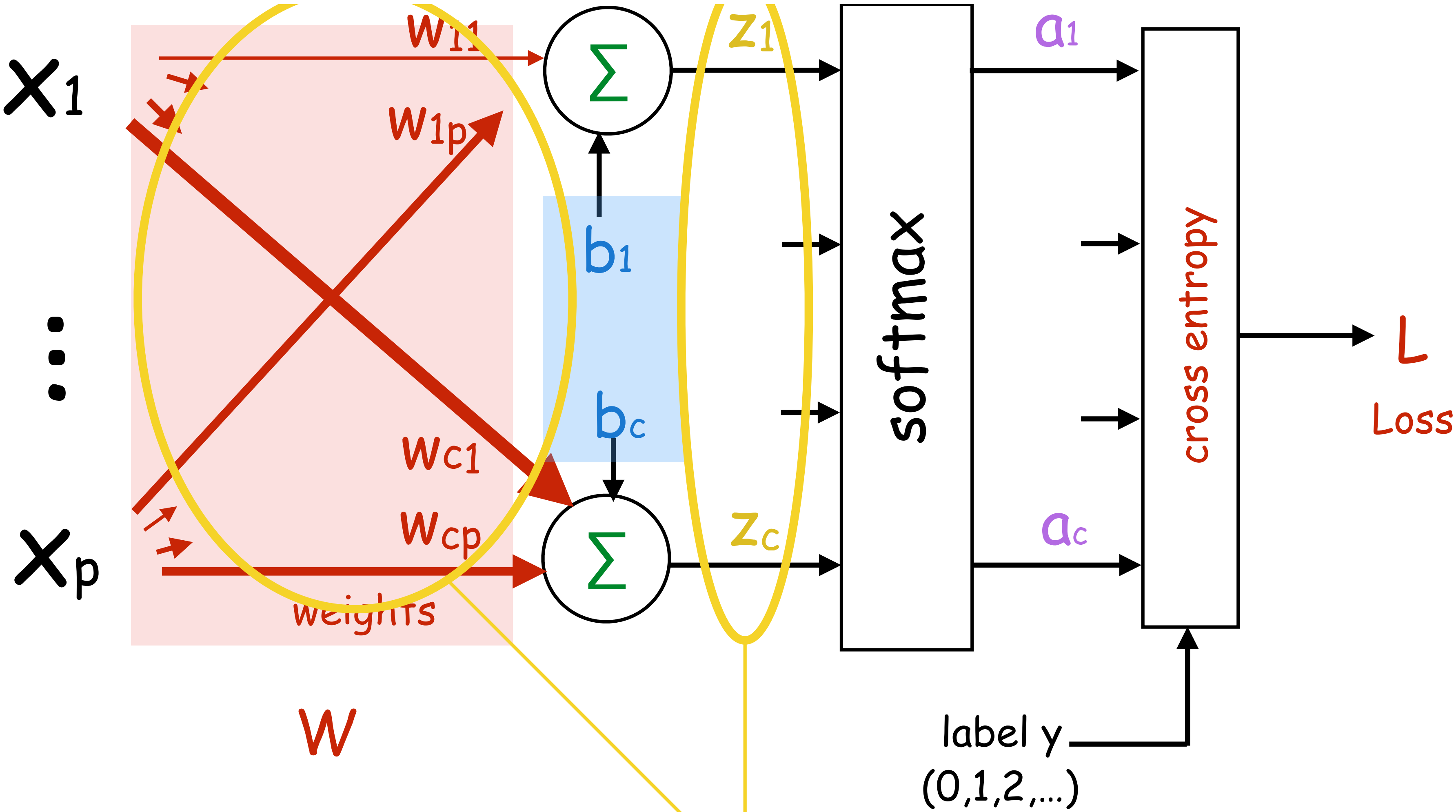


$$\frac{\partial z}{\partial b}$$

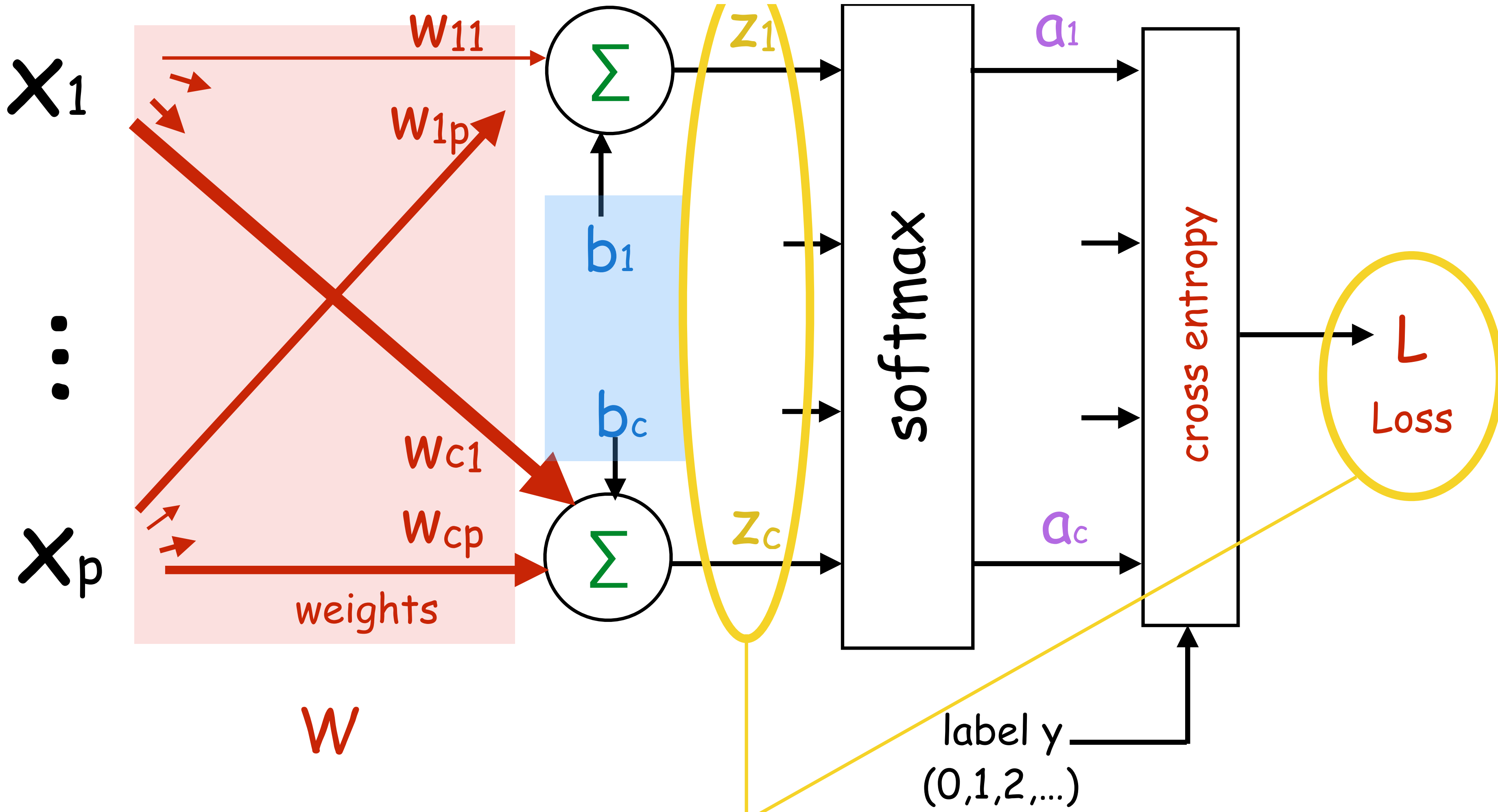
$$\frac{\partial z_1}{\partial b_1} = 1$$

$$\frac{\partial z_i}{\partial b_i} = 1$$

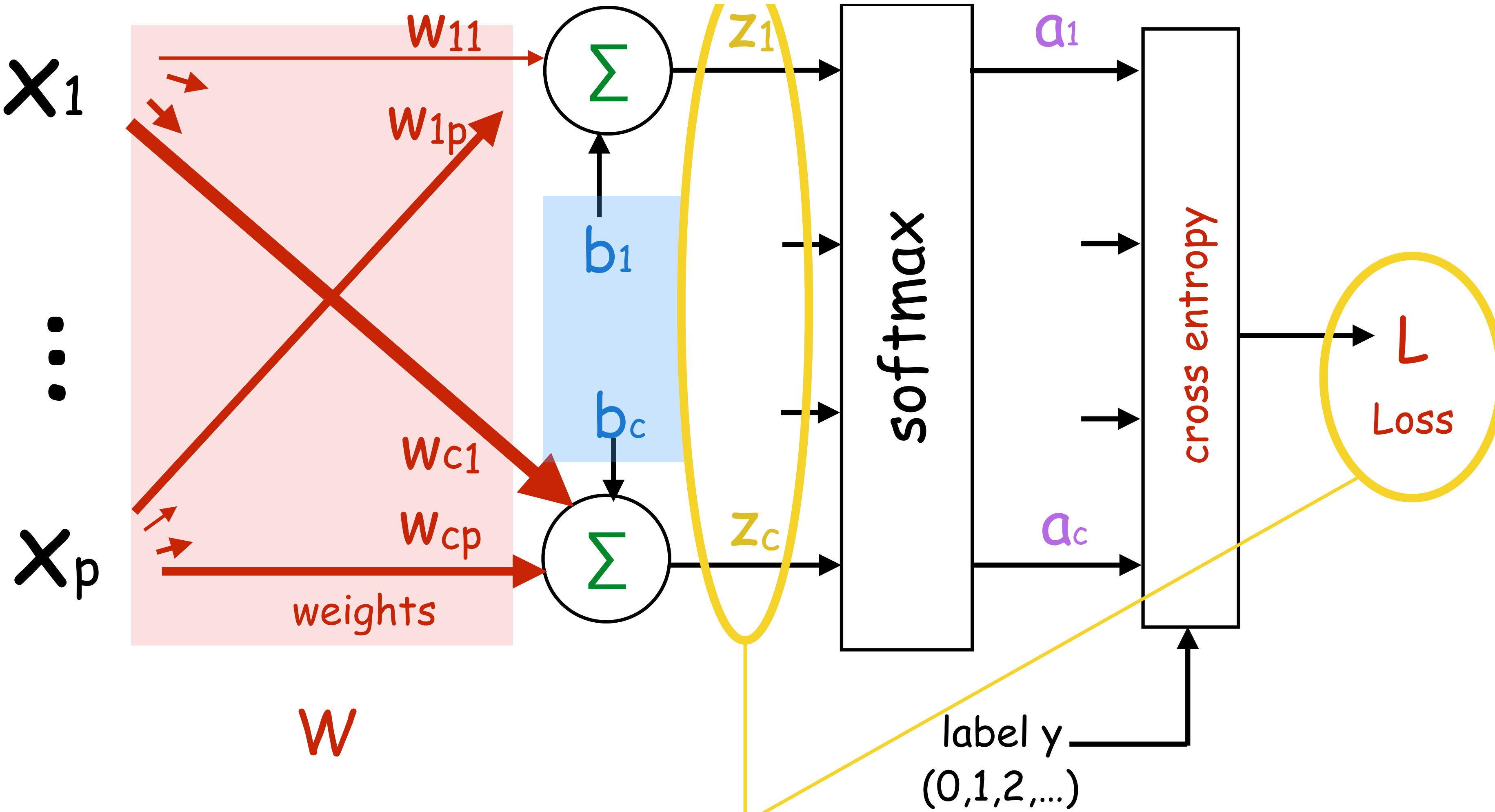
$$\frac{\partial z_c}{\partial b_c} = 1$$



$$\frac{\partial z}{\partial W} \quad \begin{array}{l} \dfrac{\partial z_1}{\partial w_{1*}} = \\[2ex] \dfrac{\partial z_i}{\partial w_{i*}} = \\[2ex] \dfrac{\partial z_c}{\partial w_{c*}} = \end{array} \boxed{\begin{array}{ccc} \dfrac{\partial z_1}{\partial w_{11}}, & \cdots & , \dfrac{\partial z_1}{\partial w_{1p}} \\[2ex] \dfrac{\partial z_i}{\partial w_{i1}}, & \cdots & , \dfrac{\partial z_i}{\partial w_{ip}} \\[2ex] \dfrac{\partial z_c}{\partial w_{c1}}, & \cdots & , \dfrac{\partial z_c}{\partial w_{cp}} \end{array}} = \boxed{\begin{array}{cccc} x_1 & x_2 & \cdots & x_p \\[2ex] x_1 & x_2 & \cdots & x_p \\[2ex] x_1 & x_2 & \cdots & x_p \end{array}}$$



$$\frac{\partial L}{\partial z} = \left(\frac{\partial L}{\partial z_1}, \ldots, \frac{\partial L}{\partial z_i}, \ldots, \frac{\partial L}{\partial z_c}\right) = ?$$



Chain Rule

$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z}$$

vector (1 by c) vector (1 by c) matrix (c by c)

$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z}$$

vector (1 by c) vector (1 by c) matrix (c by c)

$$\left(\frac{\partial L}{\partial z_1}, \ldots, \frac{\partial L}{\partial z_i}, \ldots, \frac{\partial L}{\partial z_c}\right) =$$

$$\left(\frac{\partial L}{\partial a_1}, \ldots, \frac{\partial L}{\partial a_i}, \ldots, \frac{\partial L}{\partial a_c}\right) \times \boxed{\begin{matrix} \frac{\partial a_1}{\partial z_1} & \cdots & \frac{\partial a_1}{\partial z_c} \\ \vdots & & \vdots \\ \frac{\partial a_c}{\partial z_1} & & \frac{\partial a_c}{\partial z_c} \end{matrix}}$$

$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z}$$

vector (1 by c) vector (1 by c) matrix (c by c)

$$\left(\frac{\partial L}{\partial z_1}, \ldots, \frac{\partial L}{\partial z_i}, \ldots, \frac{\partial L}{\partial z_c}\right) =$$

$$\left(\frac{\partial L}{\partial a_1}, \ldots, \frac{\partial L}{\partial a_i}, \ldots, \frac{\partial L}{\partial a_c}\right) \times \boxed{\begin{matrix} \frac{\partial a_1}{\partial z_1} & \cdots & \frac{\partial a_1}{\partial z_c} \\ \vdots & & \vdots \\ \frac{\partial a_c}{\partial z_1} & & \frac{\partial a_c}{\partial z_c} \end{matrix}}$$

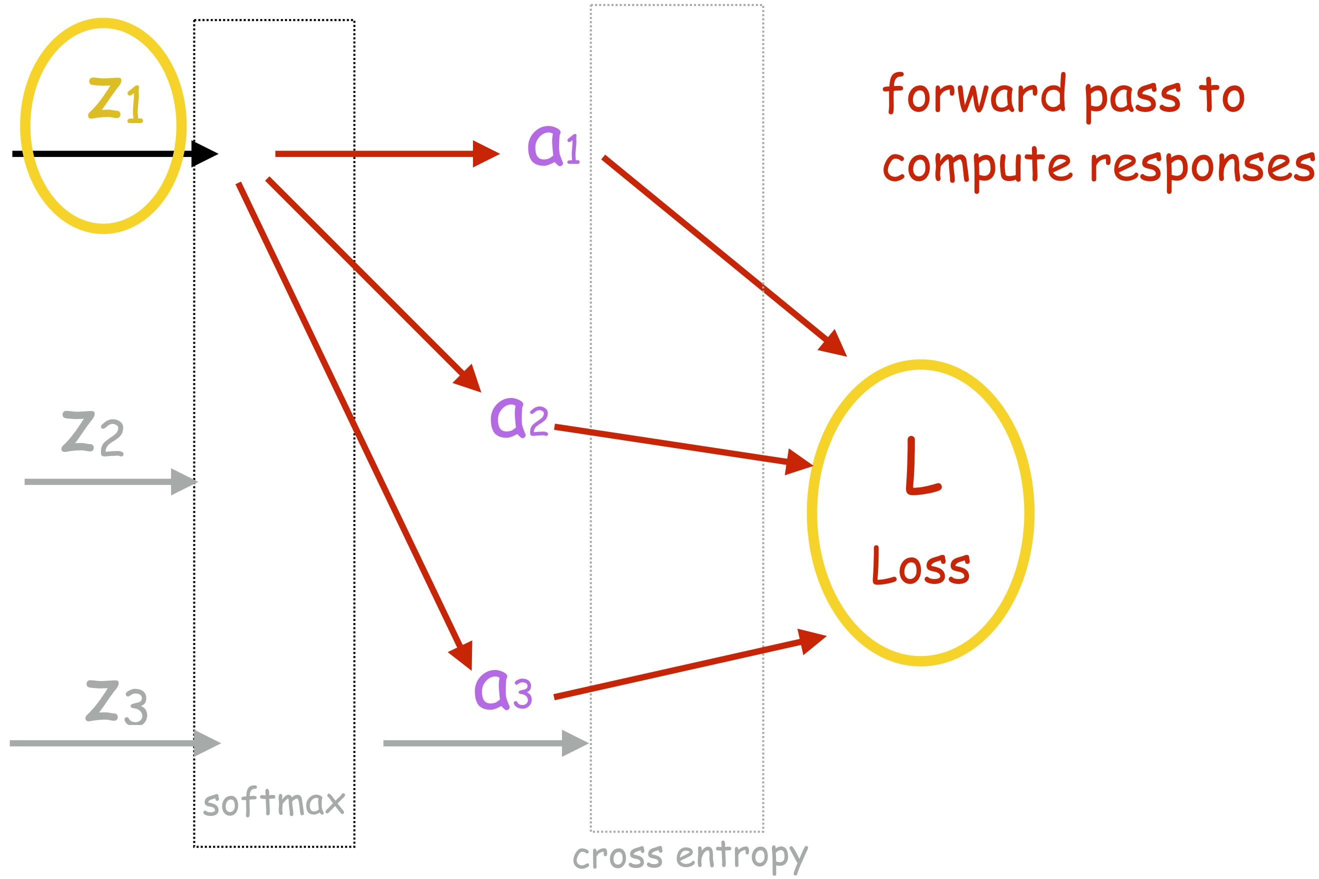
$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial a} \times \frac{\partial a}{\partial z}$$

vector (1 by c) vector (1 by c) matrix (c by c)

$$\left(\frac{\partial L}{\partial z_1}, \ldots, \frac{\partial L}{\partial z_i}, \ldots, \frac{\partial L}{\partial z_c}\right) =$$

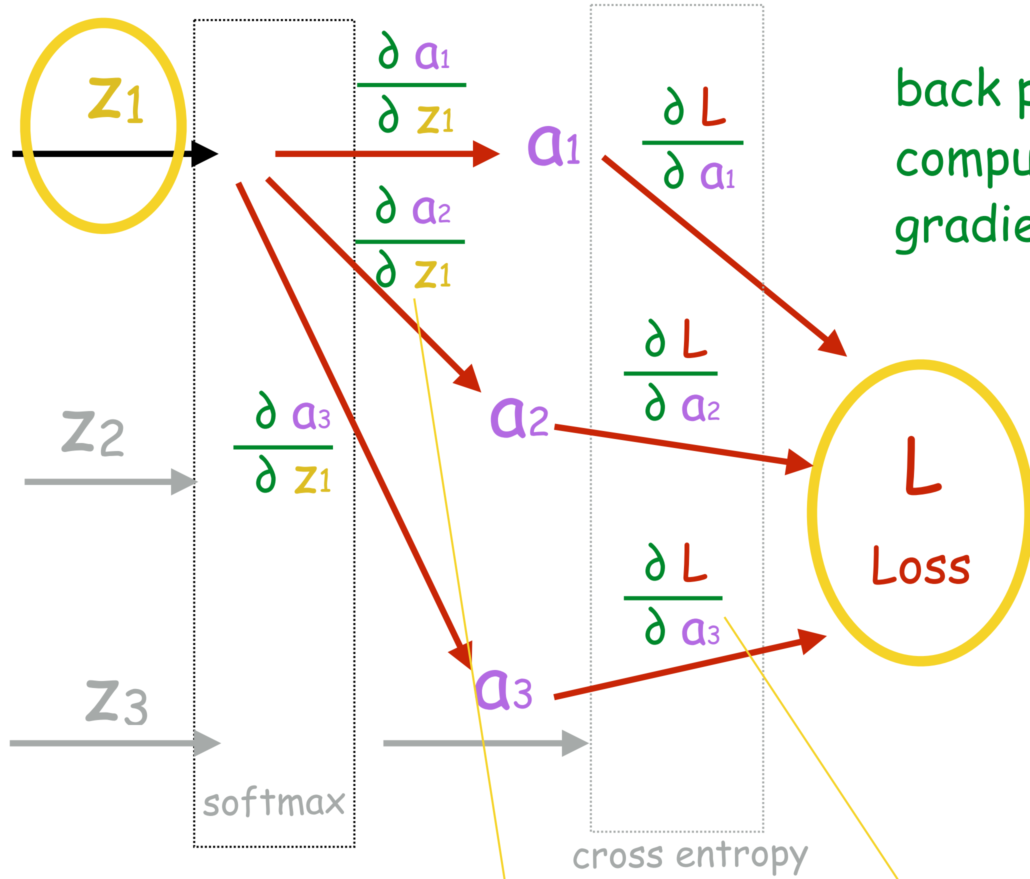
$$\left(\frac{\partial L}{\partial a_1}, \ldots, \frac{\partial L}{\partial a_i}, \ldots, \frac{\partial L}{\partial a_c}\right) \times \boxed{\begin{matrix} \frac{\partial a_1}{\partial z_1} & \cdots & \frac{\partial a_1}{\partial z_c} \\ \vdots & & \vdots \\ \frac{\partial a_c}{\partial z_1} & & \frac{\partial a_c}{\partial z_c} \end{matrix}}$$

$$\frac{\partial L}{\partial z_1}$$



$$\left(\frac{\partial L}{\partial a_1}, \ldots, \frac{\partial L}{\partial a_i}, \ldots, \frac{\partial L}{\partial a_c} \right) \times \begin{bmatrix} \frac{\partial a_1}{\partial z_1} & \cdots & \frac{\partial a_1}{\partial z_c} \\ \vdots & & \vdots \\ \frac{\partial a_c}{\partial z_1} & & \frac{\partial a_c}{\partial z_c} \end{bmatrix}$$

$$\frac{\partial L}{\partial z_1}$$

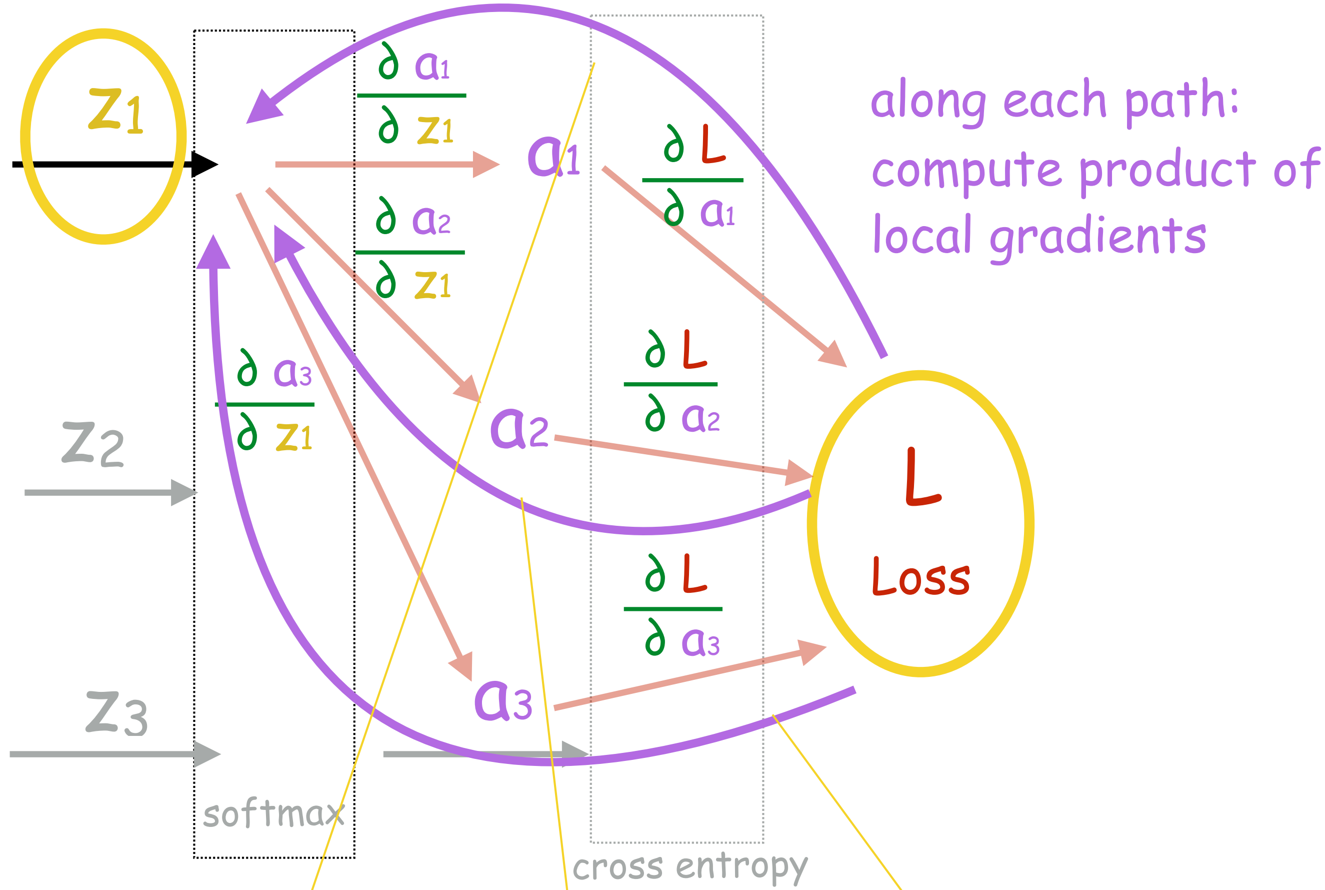


back propagation to compute local gradients

$$\frac{\partial L}{\partial a} = \left(\frac{\partial L}{\partial a_1}, \frac{\partial L}{\partial a_2}, \frac{\partial L}{\partial a_3}\right)$$

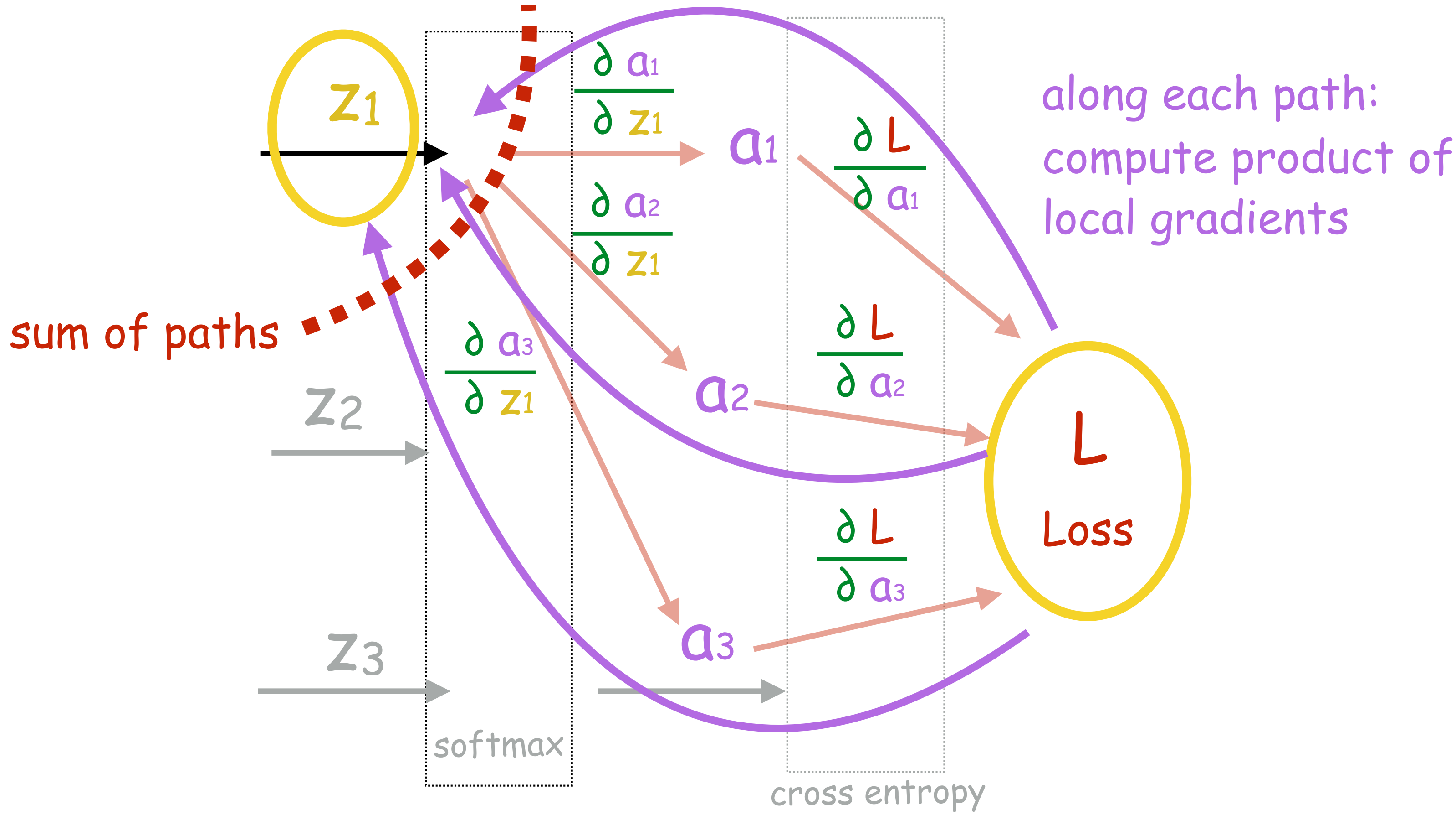
$$\frac{\partial a}{\partial z_1} = \left(\frac{\partial a_1}{\partial z_1}, \frac{\partial a_2}{\partial z_1}, \frac{\partial a_3}{\partial z_1}\right)$$

$$\frac{\partial L}{\partial z_1}$$



$$\begin{array}{ccccccc}
\frac{\partial L}{\partial a} & = & \Big(\frac{\partial L}{\partial a_1} & , & \frac{\partial L}{\partial a_2} & , & \frac{\partial L}{\partial a_3}\Big) \\
 & & \times & & \times & & \times \\
\frac{\partial a}{\partial z_1} & = & \Big(\frac{\partial a_1}{\partial z_1} & , & \frac{\partial a_2}{\partial z_1} & , & \frac{\partial a_3}{\partial z_1}\Big)
\end{array}$$

$$\frac{\partial L}{\partial z_1}$$



product along each path

sum of paths

$$\frac{\partial L}{\partial z_1} = \frac{\partial L}{\partial a_1} \times \frac{\partial a_1}{\partial z_1} + \frac{\partial L}{\partial a_2} \times \frac{\partial a_2}{\partial z_1} + \frac{\partial L}{\partial a_3} \times \frac{\partial a_3}{\partial z_1}$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial z}^{T} \times \frac{\partial z}{\partial W}$$

matrix (c by p) vector (c by 1) matrix (c by p)

$$\left(\frac{\partial L}{\partial z_1}, \ldots, \frac{\partial L}{\partial z_i}, \ldots, \frac{\partial L}{\partial z_c}\right)^{T} \times \boxed{\begin{matrix} \frac{\partial z_1}{\partial W_{11}}, & \ldots & , \frac{\partial z_1}{\partial W_{1p}} \\ \frac{\partial z_i}{\partial W_{i1}}, & \ldots & , \frac{\partial z_i}{\partial W_{ip}} \\ \frac{\partial z_c}{\partial W_{c1}}, & \ldots & , \frac{\partial z_c}{\partial W_{cp}} \end{matrix}}$$

transpose

element-wise product

$$= \boxed{\begin{matrix} \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial W_{11}}, & \ldots & , \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial W_{1p}} \\ \frac{\partial L}{\partial z_i}\frac{\partial z_i}{\partial W_{i1}}, & \ldots & , \frac{\partial L}{\partial z_i}\frac{\partial z_i}{\partial W_{ip}} \\ \frac{\partial L}{\partial z_c}\frac{\partial z_c}{\partial W_{c1}}, & \ldots & , \frac{\partial L}{\partial z_c}\frac{\partial z_c}{\partial W_{cp}} \end{matrix}}$$

Softmax Regression (train)

initialize **W** and **b**

Loop for n_epoch iterations:

Loop for each training instance (**x**, y) in training set

forward pass to compute **z**, **a** and L for the instance

backward pass to compute local gradients

$$\frac{\partial L}{\partial \mathbf{a}} \quad \frac{\partial \mathbf{a}}{\partial \mathbf{z}} \quad \frac{\partial \mathbf{z}}{\partial \mathbf{b}} \quad \frac{\partial \mathbf{z}}{\partial \mathbf{W}}$$

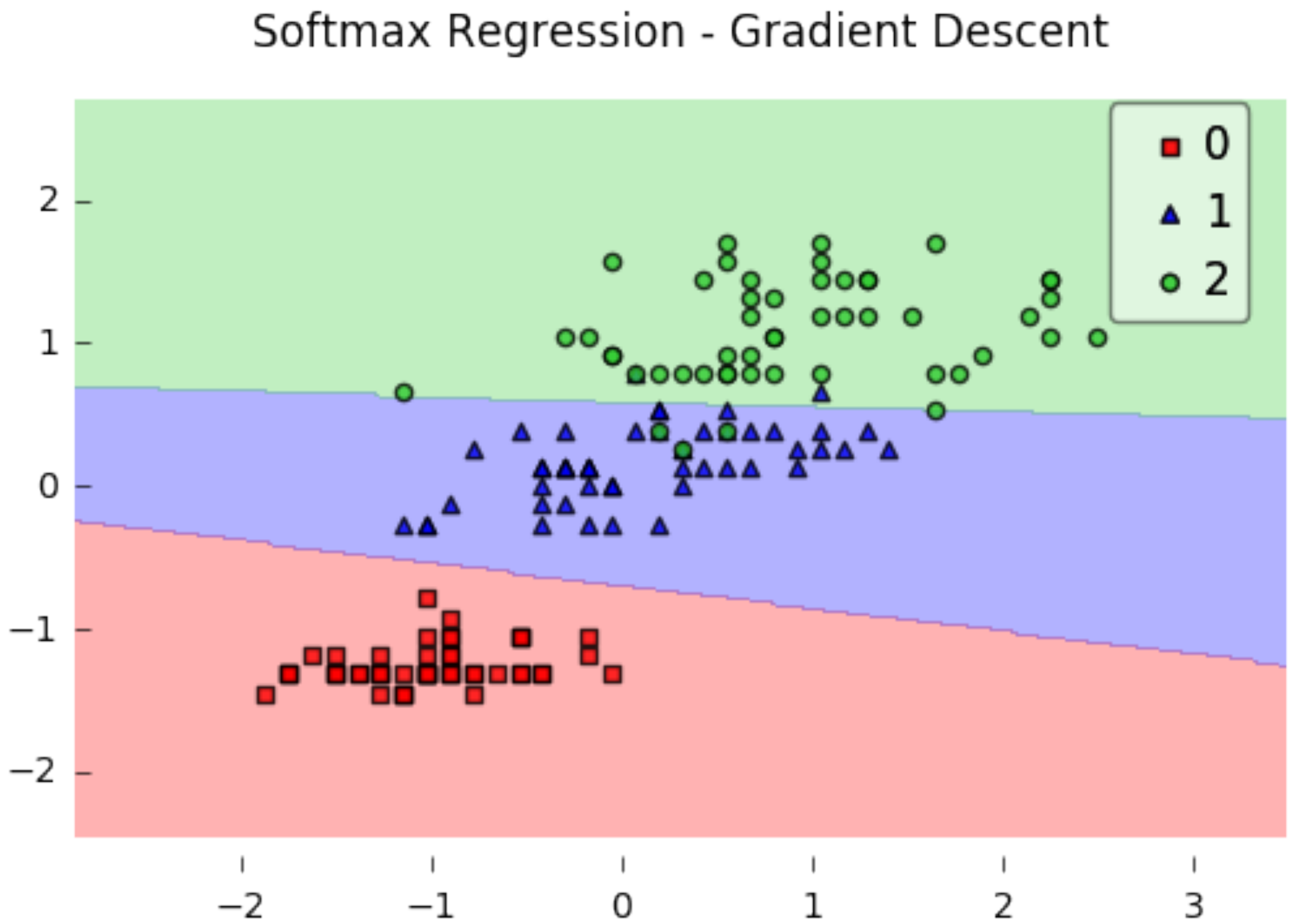
compute global gradients using chain rule

$$\frac{\partial L}{\partial \mathbf{W}} \quad \frac{\partial L}{\partial \mathbf{b}}$$

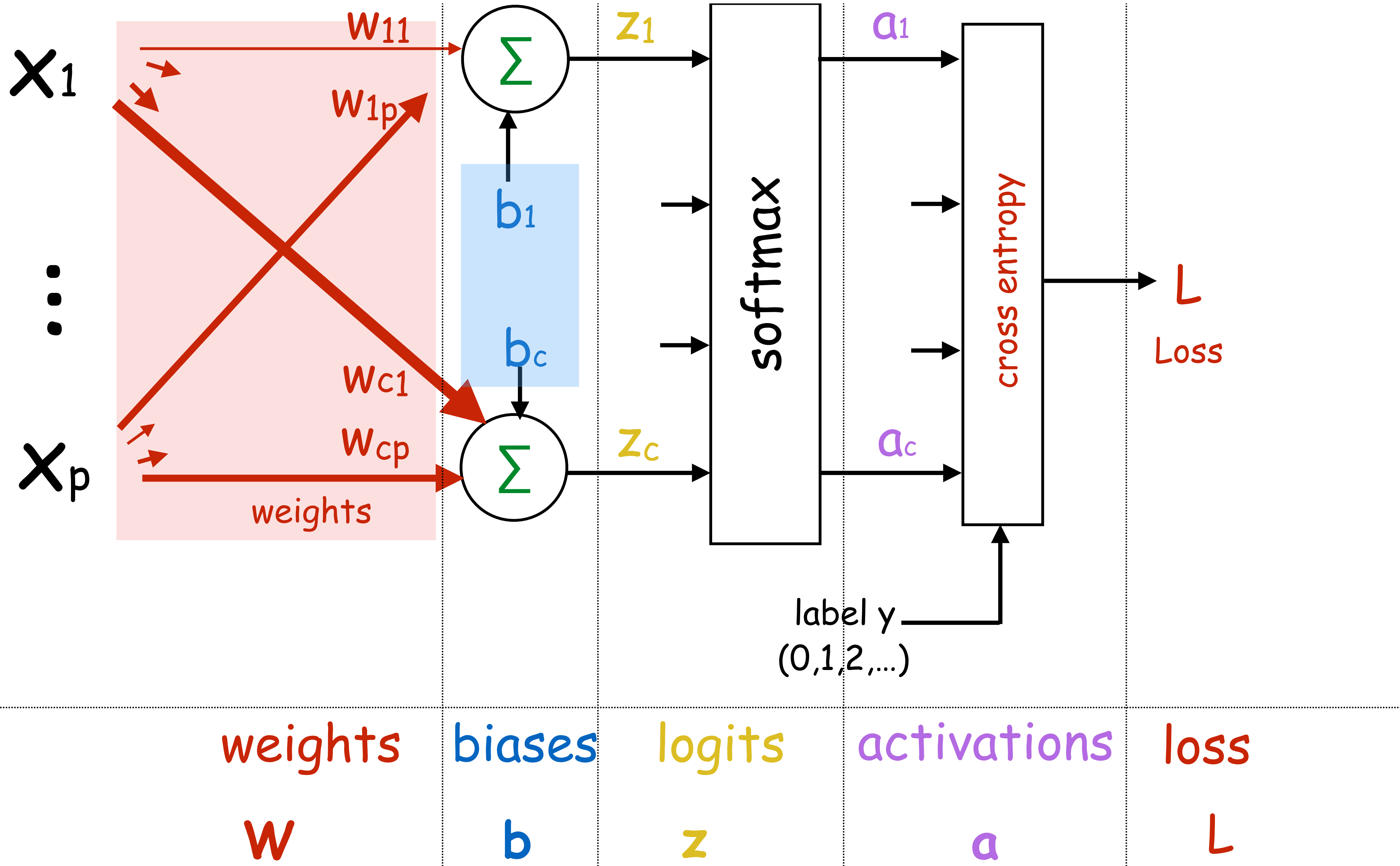
update the parameters **W** and **b**

$$\mathbf{W} \longleftarrow \mathbf{W} - a\frac{\partial L}{\partial W}$$

$$b \longleftarrow b - a\frac{\partial L}{\partial b}$$

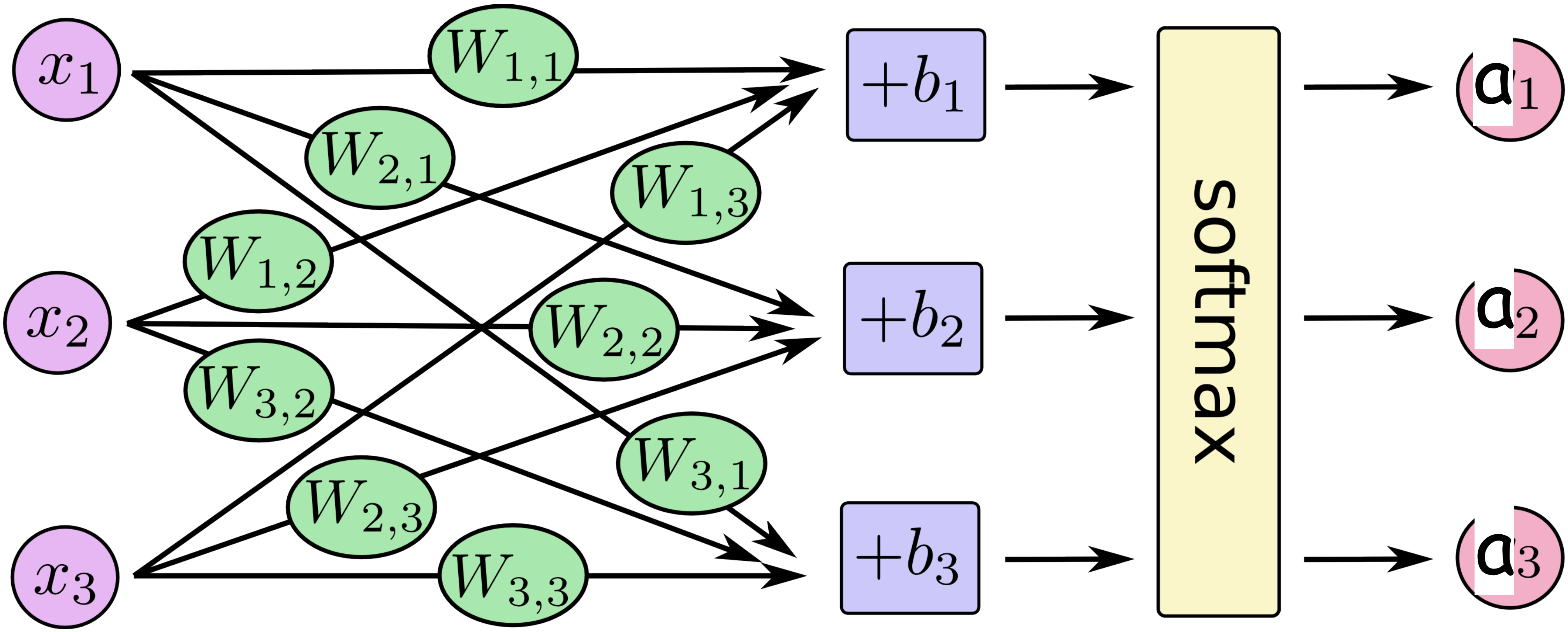


Softmax Regression - Gradient Descent
0
1
2
2
1
0
−1
−2
−2
−1
0
1
2
3



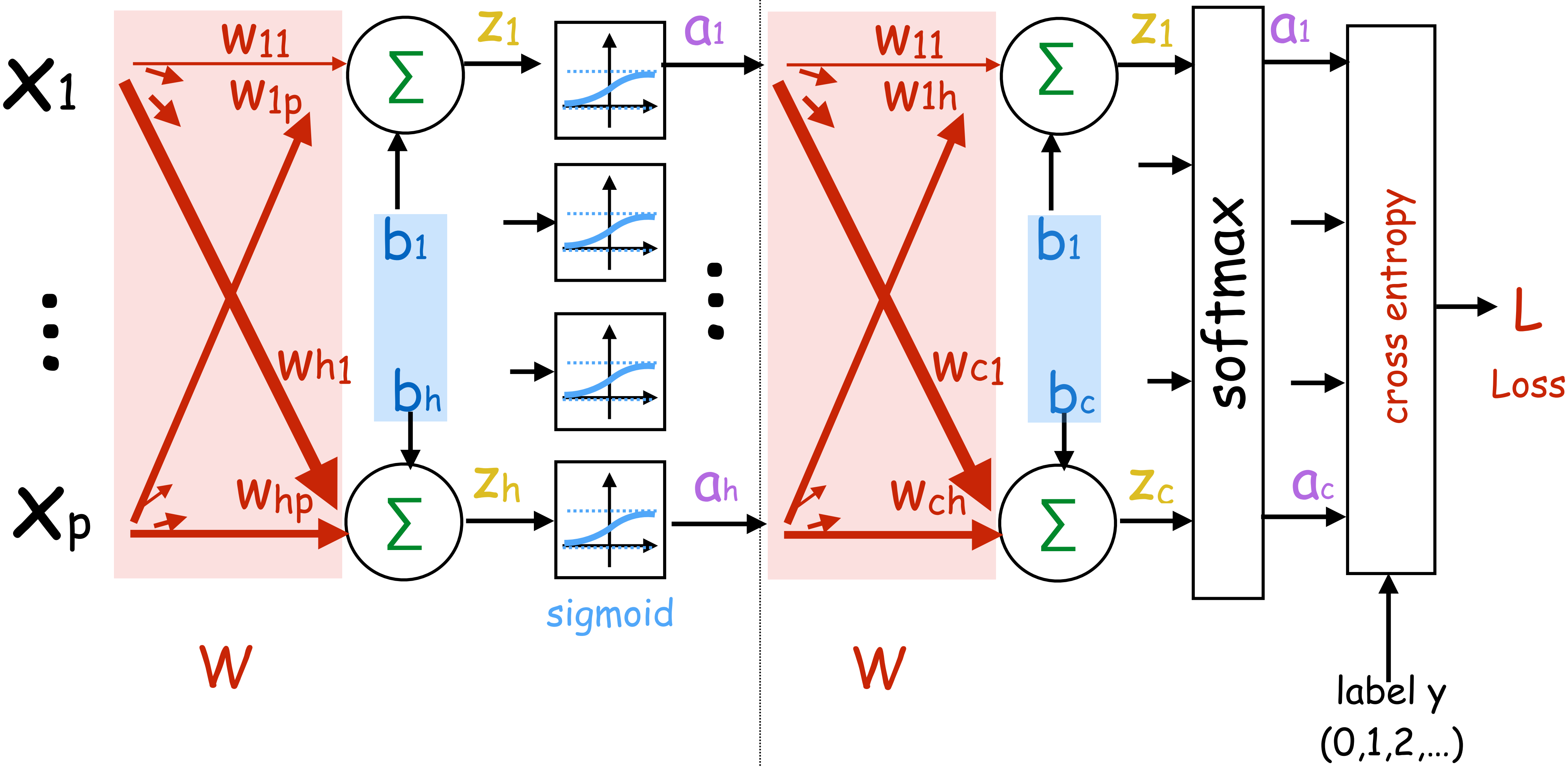
Softmax Regression

Prediction

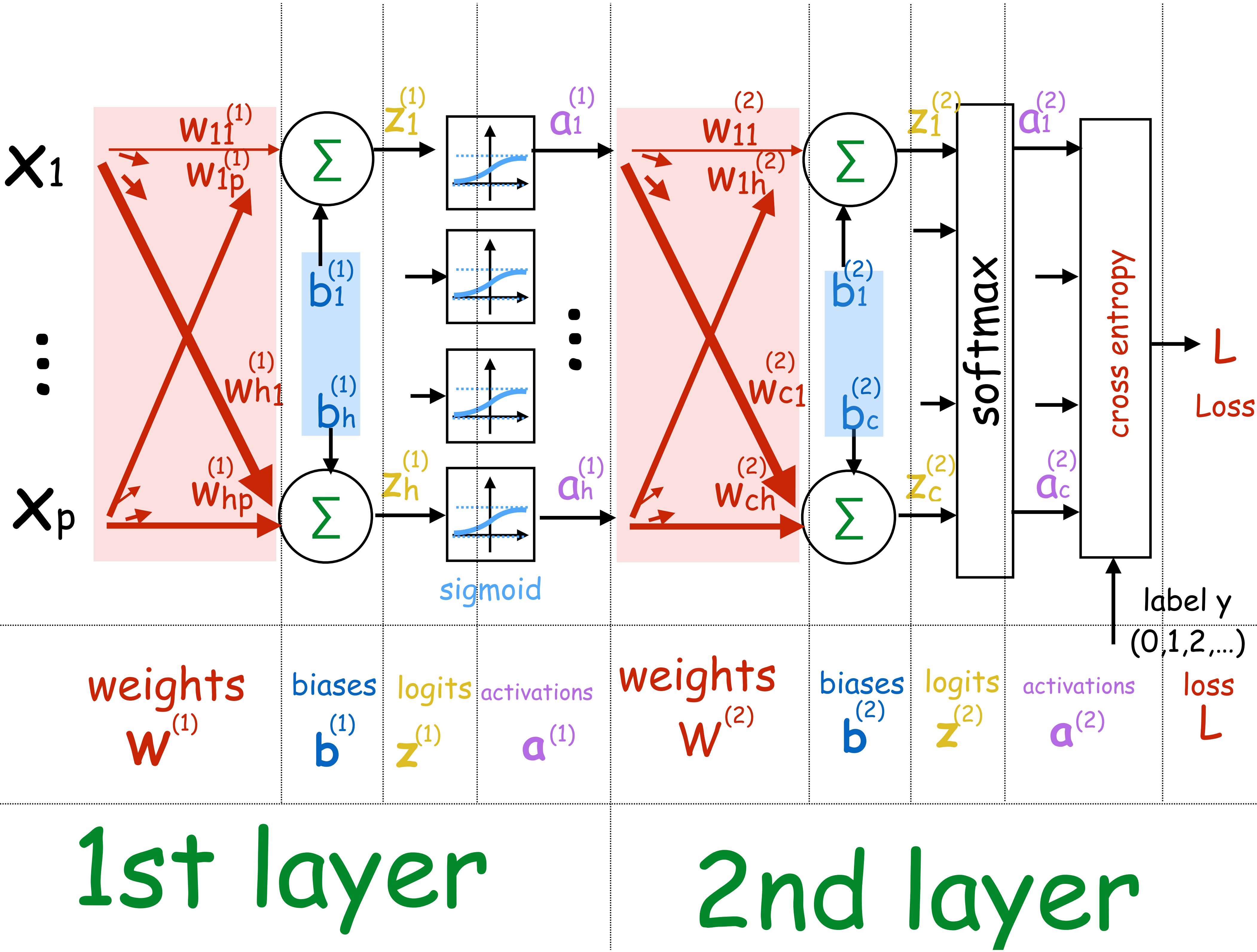


$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \text{softmax}\left(\begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

Fully Connected Neural Network



1st layer 2nd layer



x_1
x_p
$w_{11}^{(1)}$
$w_{1p}^{(1)}$
$w_{h1}^{(1)}$
$w_{hp}^{(1)}$
$b_1^{(1)}$
$b_h^{(1)}$
$z_1^{(1)}$
$z_h^{(1)}$
sigmoid
$a_1^{(1)}$
$a_h^{(1)}$
$w_{11}^{(2)}$
$w_{1h}^{(2)}$
$w_{c1}^{(2)}$
$w_{ch}^{(2)}$
$b_1^{(2)}$
$b_c^{(2)}$
$z_1^{(2)}$
$z_c^{(2)}$
softmax
$a_1^{(2)}$
$a_c^{(2)}$
cross entropy
L
Loss
label y
(0,1,2,...)
weights
$W^{(1)}$
biases
$b^{(1)}$
logits
$z^{(1)}$
activations
$a^{(1)}$
weights
$W^{(2)}$
biases
$b^{(2)}$
logits
$z^{(2)}$
activations
$a^{(2)}$
loss
L
1st layer
2nd layer

$$\frac{\partial L}{\partial a^{(1)}} = \frac{\partial L}{\partial z^{(2)}} \times \frac{\partial z^{(2)}}{\partial a^{(1)}}$$

vector length h　　　vector of length c　　　matrix (c by h)

$$\left(\frac{\partial L}{\partial a_1^{(1)}}, \ldots, \frac{\partial L}{\partial a_c^{(1)}}\right)$$

$$= \left(\frac{\partial L}{\partial z_1^{(2)}}, \ldots, \frac{\partial L}{\partial z_c^{(2)}}\right) \times \boxed{\begin{matrix} \dfrac{\partial z_1^{(2)}}{\partial a_1^{(1)}}, & \cdots & , \dfrac{\partial z_1^{(2)}}{\partial a_h^{(1)}} \\ & \cdots & \\ \dfrac{\partial z_c^{(2)}}{\partial a_1^{(1)}}, & \cdots & , \dfrac{\partial z_c^{(2)}}{\partial a_h^{(1)}} \end{matrix}}$$