พื้นฐานการแก้สมการเบื้องต้น

Function → Linear → $Y = ax + b$

Function → Non - linear → $Y = ax^2 + bx + c \,, y = e^2$

แก้สมการเบื้องต้น → กำจัดตัวแปร

แก้สมการเบื้องต้น → จัดรูปย้ายข้าง

กำจัดตัวแปร

สัมประสิทธิ์

$$4x = 5y + 3 \quad \ldots ๑$$
$$2x = 3y - 3 \quad \ldots ๒$$

๒ * 2; $4x = 6y - 6 \quad \ldots ๓$

๓ - ๑; $0 = y - 9$

$y = 9$ **แทนค่า

แทนค่า

$$2x = 3y - 3$$
$$2x = 3(9) - 3$$
$$x = \frac{24}{2} = 12$$

Check: $4(12) = 5(9) + 3$

$$48 = 48$$

จัดรูปย้ายข้าง

$$4x = 5y + 3 \quad \ldots ๑$$
$$2x = 3y - 3 \quad \ldots ๒$$

๒; $x = \frac{3y - 3}{2}$

แทนค่าเข้าไป ๑; $4\left(\frac{3y-3}{2}\right) = 5\,y + 3$

$$4\left(\frac{3y-3}{2}\right) = 5y + 3$$
$$2(3y - 3) = 5y + 3$$
$$6y - 6 = 5y + 3$$
$$6y - 5y = 6 + 3$$
$$y = 9$$
$$x = \frac{3(9) - 3}{2} = 12$$

สูตรการหาอนุพันธ์ของฟังก์พีชคณิต

(1) $\boxed{\dfrac{dc}{dx} = 0}$ $\quad \dfrac{d5}{dx} = 0 \quad \dfrac{d(-\frac{8}{3})}{dx} = 0 \quad \dfrac{d\sqrt{5}}{dx} = 0$

(2) $\boxed{\dfrac{dx}{dx} = 1}$ $\quad \dfrac{d(x+5)}{d(x+5)} = 1$

(3) $\boxed{\dfrac{d}{dx}cu = c\dfrac{du}{dx}}$ $\quad \dfrac{d(5x)}{dx} = 5\underbrace{\dfrac{dx}{dx}}_{=1} = 5(1) = 5$

(4) $\boxed{\dfrac{d}{dx}(u + v - w) = \dfrac{du}{dx} + \dfrac{dv}{dx} - \dfrac{dw}{dx}}$ $\quad \dfrac{d}{dx}\left(\dfrac{5}{6}x^4 + 2x^3 - 5x^2 + \underbrace{8}_{=0}\right) = \dfrac{d}{dx}\dfrac{5}{6}x^4 + \dfrac{d2x^3}{dx} - \dfrac{d5x^2}{dx}$

(5.1) $\boxed{\dfrac{du^2}{dx} = nu^{n-1}\dfrac{du}{dx}}$ $u =$ กลุ่มของ x

(5.2) $\boxed{\dfrac{dx^n}{dx} = nu^{n-1}}$

(5.1) $\boxed{\dfrac{d}{dx}\left(\dfrac{5}{6}x^4 + 2x^3 - 5x^2 + 8\right)^5}$

(3),(4)

$\boxed{5\left(\frac{5}{6}x^4 + 2x^3 - 5x^2 + 8\right)^4 \frac{d}{dx}\left(\frac{5}{6}x^4 + 2x^3 - 5x^2 + 8\right)}$

(5.2)

$\boxed{5\left(\dfrac{5}{6}x^4 + 2x^3 - 5x^2 + 8\right)^4\left(\dfrac{5}{6}\dfrac{dx^4}{dx} + 2\dfrac{dx^3}{dx} - 5\dfrac{dx^2}{dx}\right)}$

$\boxed{5\left(\dfrac{5}{6}x^4 + 2x^3 - 5x^2 + 8\right)^4\left(\dfrac{10}{3}x^3 + 6x^2 - 10x\right)}$

สูตรการหาอนุพันธ์ของฟังก์พีชคณิต

กฎลูกโซ่

(6) $$\frac{d}{dx}(uvw) = vw\frac{dv}{dx} + uw\frac{dv}{dx} + uv\frac{dw}{dx}$$

$$y = (4x^5 + 2x^3 - 8x)(\frac{5}{4}x^3 - 6)$$

$$\frac{dy}{dx} = \frac{d}{dx}(4x^5 + 2x^3 - 8x)(\frac{5}{4}x^3 - 6)$$

$$(\frac{5}{4}x^3 - 6)\frac{d}{dx}(4x^5 + 2x^3 - 8x) + (4x^5 + 2x^3 - 8x)\frac{d}{dx}(\frac{5}{4}x^3 - 6)$$

$$(\frac{5}{4}x^3 - 6)(20x^4 + 6x^2 - 8) + (4x^5 + 2x^3 - 8x)(\frac{15}{4}x^2)$$

4,3,2,1,5.2 คิดในใจ

(7) $$\frac{d}{dx}\left(\frac{u}{v}\right) = \frac{1}{v^2}\left[v\frac{du}{dx} - u\frac{dv}{dx}\right] = \frac{v\frac{du}{dx} - u\frac{dv}{dx}}{v^2}$$

6

4,3,2,1,5.2

$$y = \frac{(2x+3)^3}{(4x^2-1)^8}$$

$$y = \frac{(4x^2-1)^8\frac{d}{dx}(2x+3)^3 - (2x+3)^3\frac{d}{dx}(4x^2-1)^8}{(4x^2-1)^{16}}$$

$$y = \frac{(4x^2-1)^8 3(2x+3)^2(2) - (2x+3)^3 8(4x^2-1)^7(8x)}{(4x^2-1)^{16}}$$

v^2

อนุพันธ์(derivative)

หาจุดสูงสุดจุดต่ำสุด

$$\frac{dx^n}{dx} = \mathrm{n}x^{n-1}$$

$$\mathrm{d}\left(\frac{x^n + x^m}{d_x}\right) = \frac{\mathrm{d}x^n}{\mathrm{d}x} + \frac{\mathrm{d}x^m}{\mathrm{d}x}$$
$$= \mathrm{n}x^{n-1} + mx^{m-1}$$

$$\frac{\mathrm{d}c}{\mathrm{d}x} = 0$$

$$\frac{de^x}{dx} = e^x$$

$$\frac{\mathrm{d}\ln x}{\mathrm{d}x} = \frac{1}{x}$$

$x^3+3x^2 + 2x + 5$

$$\frac{\mathrm{d}(x^3 + 3x^2 + 2x + 5)}{\mathrm{d}x}$$

$3x^2 + 3(2) + 2$

$x^5+4x^3 + 3x^2 + x + 7$

$$\frac{\mathrm{d}(x^5 + 4x^3 + 3x^2 + x + 7)}{\mathrm{d}x}$$

$5x^4 + 12x^2 + 6x + 1$

เปรียบเทียบ x1 ; x2,x3 จึงเท่ากับค่าคงที่ มีค่าเท่ากับ 0

อนุพันธ์ย่อย (partial derivative)

- กำหนดให้ตัวแปรอื่นเป็นค่าคงที่ → $\frac{\partial(x_1^2+x2+x3)}{\partial x1} = 2x1 + 0 + 0$
- ∂

5*2

2-1

$x^3 + 5y^2 + 5\text{x}\,y$	$x^3 + 5y^2 + 5\text{x}\,y$	$x^{0.6}y^{0.4}$	$x^{0.6}y^{0.4}$
$\frac{\partial(X^3 + 5y^2 + 5XY)}{\partial X}$	$\frac{\partial(X^3 + 5y^2 + 5XY)}{\partial y}$	$\frac{\partial(x^{0.6}y^{0.4})}{\partial y}$	$\frac{\partial(x^{0.6}y^{0.4})}{\partial x}$
$3x^2 + 0 + 5y(1)$	$0 + 10\text{y} + 5X(1)$	$x^{0.6}(0.4)y^{-0.6}$	$y^{0.4}(0.6)x^{-0.4}$

$$\frac{\partial(x+y)}{\partial x} = 1$$

$$\frac{\partial(x+y)}{\partial y} = 1$$

$$\frac{\partial(x-y)}{\partial x} = 1$$

$$\frac{\partial(x-y)}{\partial y} = -1$$

$$\frac{\partial(xy)}{\partial x} = y$$

$$\frac{\partial(xy)}{\partial y} = x$$

$$\frac{\partial(x/y)}{\partial x} = 1/y$$

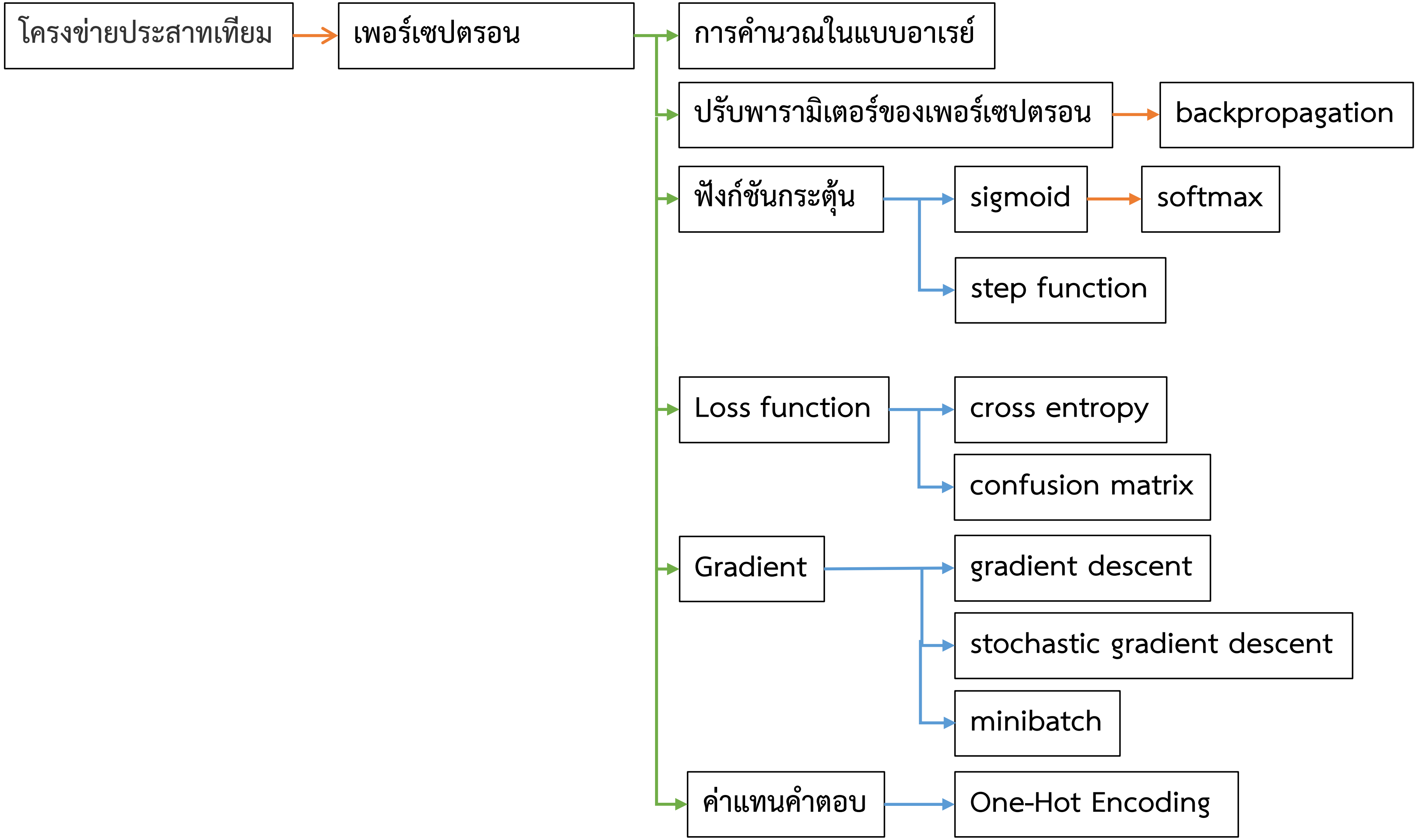
$$\frac{\partial(x/y)}{\partial y} = -x/y^2$$

```
class Buak:
    def pai(self,x,y): # ไปข้างหน้า
        return x+y
    def yon(self,g): # ย้อนกลับ
        return g, g
```

```
class Lop:
    def pai(self,x,y):
        return x-y
    def yon(self,g):
        return g,-g
```

```
class Khun:
    def pai(self,x,y): self.x = x self.y = y
        return x*y
    def yon(self,g):
        return g*self.y, g*self.x
```

```
class Han:
    def pai(self,x,y): self.x = x self.y = y
        return x/y
    def yon(self,g):
        return g/self.y,-g*self.x/self.y**2
```



โครงข่ายประสาทเทียม
เพอร์เซปตรอน
การคำนวณในแบบอาเรย์
ปรับพารามิเตอร์ของเพอร์เซปตรอน
backpropagation
ฟังก์ชันกระตุ้น
sigmoid
softmax
step function
Loss function
cross entropy
confusion matrix
Gradient
gradient descent
stochastic gradient descent
minibatch
ค่าแทนคำตอบ
One-Hot Encoding

โครงข่ายประสาทเทียม

เพอร์เซปตรอน (perceptron)

(Weight, bias)

พารามิเตอร์ (parameter)

ค่าที่จะต้องปรับให้เหมาะสมกับปัญหาที่พิจารณา

ข้อมูล

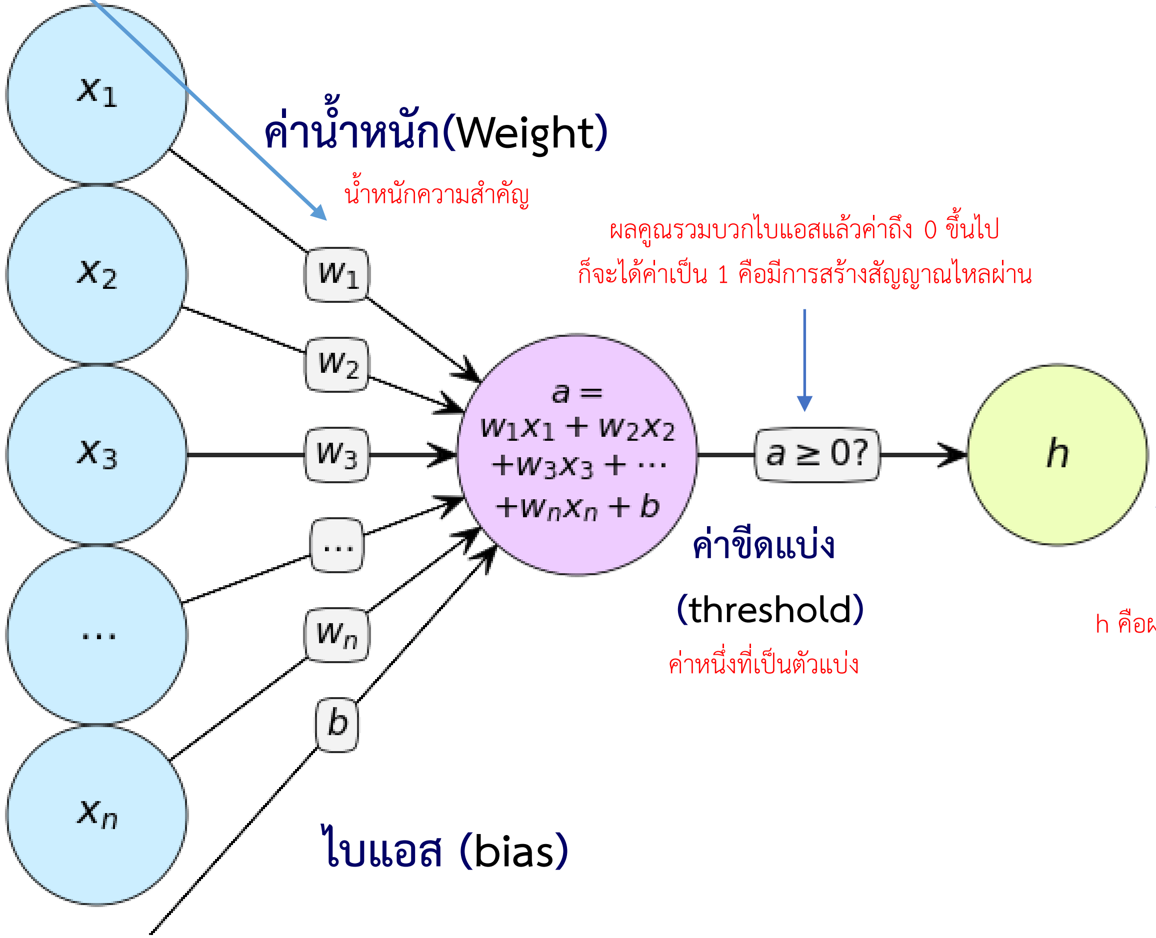
ข้อมูลป้อนเข้า w_1, w_2

ค่าน้ำหนัก(Weight)

น้ำหนักความสำคัญ

ผลคูณรวมบวกไบแอสแล้วค่าถึง 0 ขึ้นไป

ก็จะได้ค่าเป็น 1 คือมีการสร้างสัญญาณไหลผ่าน



ค่าขีดแบ่ง

(threshold)

ค่าหนึ่งที่เป็นตัวแบ่ง

h คือผลลัพธ์ที่ได้ ซึ่งจะมีค่า 1 หรือ 0

ไบแอส (bias)

เอาค่า x มาคูณกับ w แล้ว บวก b จากนั้นก็ได้ค่า a แล้วนำ a ไปเข้าฟังก์ชันกระตุ้น

การคำนวณในแบบอาเรย์

อาเรย์ ก็คือ เมทริกซ์

$$a_i = \sum_{j=0}^{m-1} w_j x_{i,j} + b$$

a ของแต่ละตัว

n คือจำนวนข้อมูล

m คือจำนวนตัวแปร

$$\mathbf{x} = \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,m-1} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1,0} & x_{n-1,1} & \cdots & x_{n-1,m-1} \end{bmatrix}$$

$$\vec{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{bmatrix}$$

อาเรย์หนึ่งมิติ (เวกเตอร์)

การคูณเมทริกซ์ โดยใน numpy

จะใช้คำสั่ง np.dot()

$$\vec{a} = \mathbf{x} \cdot \vec{w} + b$$

ตัวเอียงธรรมดาจะหมายถึงค่าเลขตัวเดียว

อาเรย์สองมิติ (เมทริกซ์)

$$\vec{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^{m-1} w_j x_{0,j} + b \\ \sum_{j=0}^{m-1} w_j x_{1,j} + b \\ \vdots \\ \sum_{j=0}^{m-1} w_j x_{n-1,j} + b \end{bmatrix}$$

ฟังก์ชันของเพอร์เซปตรอน

```
def h(X):
    w = np.array([20,50])
    b = -1000
    a = np.dot(X,w) + b
    return (a>=0).astype(int)
```

```
X = np.array([ [10,15], [10,18], [22,15] ]) print(h(X)) # ได้ [0 1 1]
```

กำหนด
ค่าพารามิเตอร์ตั้งต้น

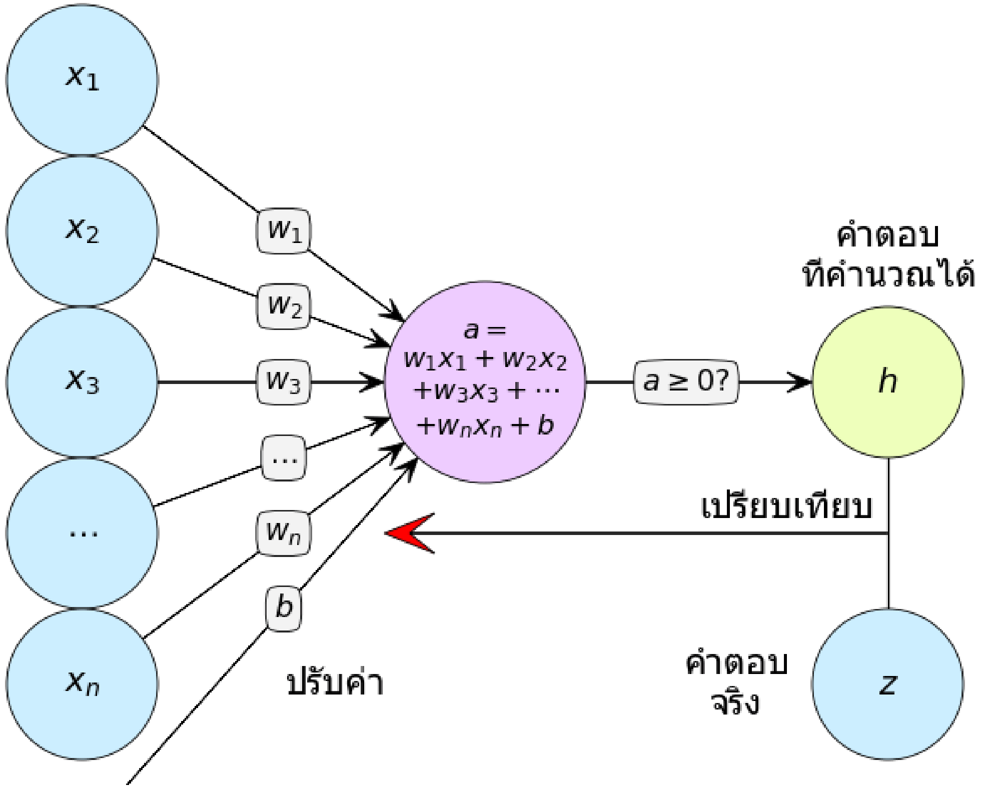
คำนวณผลลัพธ์แล้ว
ปรับค่า

ทำซ้ำจนครบ
หรือ ทายถูกทั้งหมด

```
eta = 0.2 # อัตราการเรียนรู้
print('เริ่มต้น: h(x)=%s, w=%s, b=%s'%(h(X),w,b))
for j in range(100): # ให้ทำซ้ำสูงสุด 100 ครั้ง
    for i in range(4):
        z_h = z[i] - h(X[i])
        dw = eta*z_h*X[i]
        db = eta*z_h w += dw b += db
        print('รอบ %d.%d: h(x)=%s, w=%s, b=%s, Δw=%s, Δb=%s'%(j+1,i+1,h(X),w,b,dw,db))
    if(np.all(h(X)==z)): # ถ้าผลลัพธ์ถูกต้องทั้งหมดก็ให้เสร็จสิ้นการเรียนรู้ break
```

η อัตราการเรียนรู้ (learning rate)

ค่าที่กำหนดจำนวนการปรับพารามิเตอร์



คำตอบ 1 ถ้าคำนวณได้ 0 แปลว่าค่าต่ำไป
เพิ่มค่า Weight ให้สูงขึ้น
(ตัวแปรไหนเป็นบวก น้ำหนักที่คูณตัวนั้นจะต้องเพิ่มค่า,
ตัวแปรเป็นลบ น้ำหนักจะต้องลดค่า)
bias = ปรับเพิ่มขึ้น

คำตอบ 0 ถ้าคำนวณได้ 1 แปลว่าค่าสูงไป
ลดค่า Weight น้อยลง
(ตัวแปรไหนเป็นบวกก็ลดค่า,
ตัวแปรไหนเป็นลบต้องเพิ่มค่าน้ำหนัก)
bias = ปรับลดค่า

z_i คือคำตอบจริงตัวที่ i

$$\Delta \mathbf{w} = \eta(z_i - h(\vec{x}_i))\vec{x}_i$$
$$\mathbf{w}_{\text{ใหม่}} = \mathbf{w} + \Delta \mathbf{w}$$

$$\Delta b = \eta(z_i - h(\vec{x}_i))$$
$$b_{\text{ใหม่}} = b + \Delta b$$

$h(x_i)$ คือคำตอบที่คำนวณได้จากตัวแปรต้น x ตัวที่ i

z_i เป็น 1 แต่ $h(x_i)$ เป็น 0 จะได้ z_i-$h(x_i)$=1 แบบนั้น b จะถูกปรับเพิ่ม ส่วน w_j คือค่าน้ำหนักของตัวแปรตัวที่ j จะถูกปรับโดยขึ้นกับว่าค่า $x_{i,j}$ เป็นเท่าไหร่ ถ้าเป็นบวก w_j ก็ถูกปรับเพิ่ม ถ้าเป็นลบก็ลด

z_i เป็น 0 แต่ $h(x_i)$ เป็น 1 ก็จะได้ z_i-$h(x_i)$=-1 แล้วการเปลี่ยนแปลงก็เป็นไปในทางตรงกันข้าม

z_i และ $h(x_i)$ เป็น 0 หรือ 1 ทั้งคู่ z_i-$h(x_i)$=0 ก็จะไม่เกิดการเปลี่ยนแปลงใดๆ

ปรับพารามิเตอร์ของเพอร์เซปตรอนอย่างง่าย

ฟังก์ชันกระตุ้น (activation function)

(มีค่ามากเมื่อ x มาก และน้อยเมื่อ x น้อย x ที่ค่ามากกว่าจะต้องได้ค่าฟังก์ชันที่มากกว่าหรือเท่ากับค่า x ที่น้อยกว่าเสมอ)

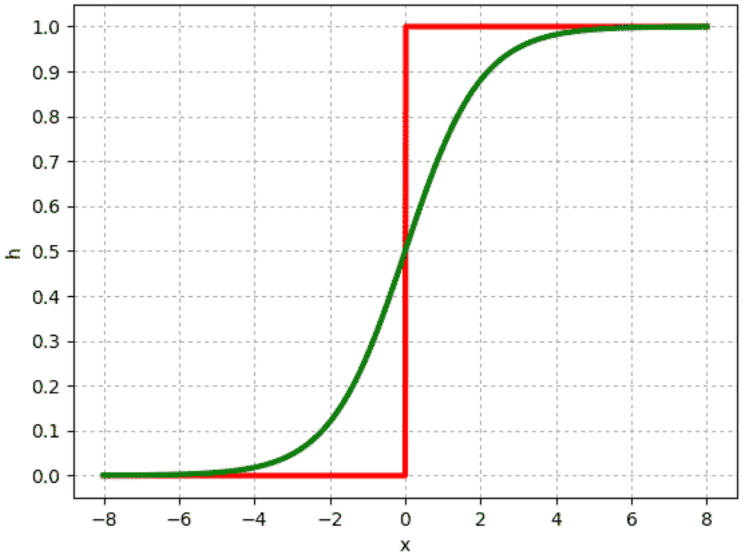
$$a_i = a(\vec{x}_i) = \sum_{j=0}^{m-1} w_i x_{i,j} + b$$

step function

$$h_i = h(\vec{x}_i) = \begin{cases} 1 \text{ ถ้า } a_i & \geq 0 \\ 0 \text{ ถ้า } a_i & < 0 \end{cases}$$

sigmoid

$$h_i = \mathbf{sigmoid}(a_i) = \frac{1}{1+\exp(-a_i)}$$



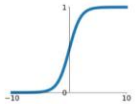

```
def sigmoid(x):
  return 1/(1+np.exp(-x))
```

```
def bandai(x):
  return x>0
```

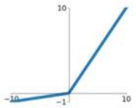
sigmoid

step function

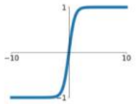
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



Leaky ReLU
$\max(0.1x, x)$

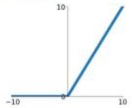


tanh
$\tanh(x)$

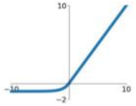


Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ReLU
$\max(0, x)$



ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$



(activation function)

ฟังก์ชันค่าเสียหาย (loss function)

(ค่าที่เป็นตัวบ่งบอกถึงความผิดพลาดของผลการทำนาย ยิ่งมีค่าน้อยก็ยิ่งดี)

J ขึ้นอยู่กับ h
h ขึ้นอยู่กับ a
a ก็ขึ้นอยู่กับ w
w ก็ขึ้นอยู่กับ b

```
def ha_entropy(z,h):
    return -(z*np.log(h)+(1-z)*np.log(1-h))
```

$$J_i = -z_i \ln h_i - (1-z_i)\ln(1-h_i)$$

เอนโทรปีไขว้ (cross entropy)

ความควรจะเป็น (likelihood)

(loss function)

หาอนุพันธ์

$$\frac{\partial J_i}{\partial h_i} = \frac{z_i}{h_i} - \frac{1-z_i}{1-h_i} = \frac{h_i - z_i}{h_i(1-h_i)}$$

แทนค่า

$$a_i = a(\vec{x}_i) = \sum_{j=0}^{m-1} w_i x_{i,j} + b$$

หาอนุพันธ์

$$\frac{\partial a_i}{\partial w_j} = x_{j,i}$$

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

$$g_{a_i} = \frac{\partial J_i}{\partial a_i} = \frac{\partial J_i}{\partial h_i}\frac{\partial h_i}{\partial a_i} = h_i - z_i$$

ความชันของค่าเสียหายเทียบกับ ai

แทนค่า

(พิจารณาปรับค่าพารามิเตอร์)

sigmoid (activation function)

$$h_i = \mathbf{sigmoid}(a_i) = \frac{1}{1+\exp(-a_i)}$$

หาอนุพันธ์

$$\frac{\partial h_i}{\partial a_i} = \frac{\exp(-x)}{(1+\exp(-x))^2} = \frac{1}{1+\exp(-a_i)}\,\frac{1+\exp(-a_i)-1}{1+\exp(-a_i)} = h_i(1-h_i)$$

อนุพันธ์ย่อย (partial derivative)

หรือ

ค่าความชัน (gradient)

พิจารณาปรับค่าพารามิเตอร์ตัวไหนค่าเสียหายจะมีค่าเปลี่ยนแปลงไปยังไง ด้วยการหา อนุพันธ์ย่อย ของ J เทียบกับพารามิเตอร์ตัวนั้น

J เทียบกับ w

$$\frac{\partial J_i}{\partial w_j} = \frac{\partial J_i}{\partial h_i}\frac{\partial h_i}{\partial a_i}\frac{\partial a_i}{\partial w_j}$$

$$\frac{\partial J_i}{\partial b} = \frac{\partial J_i}{\partial h_i}\frac{\partial h_i}{\partial a_i}\frac{\partial a_i}{\partial b}$$

J เทียบกับ B

(หาความชัน)

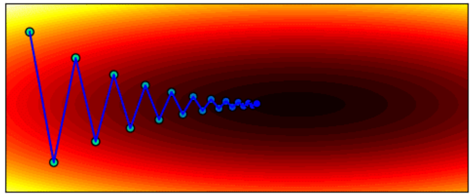
$$\frac{\partial a_i}{\partial w_j} = x_{j,i}$$

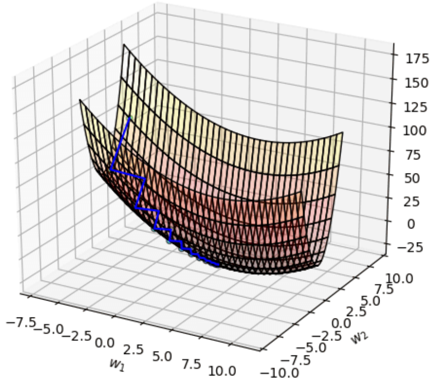
$$\frac{\partial a_i}{\partial b} = 1$$

ค่าได้จากการหาอนุพันธ์

$$g_{a_i} = \frac{\partial J_i}{\partial a_i} = \frac{\partial J_i}{\partial h_i}\frac{\partial h_i}{\partial a_i} = h_i - z_i$$

ความชันของค่าเสียหายเทียบกับ ai





การเคลื่อนลงตามความชัน (gradient descent)

(วิธีการที่พิจารณาหาค่าความชันแล้วปรับพารามิเตอร์ไปในทิศทางตามความชัน)

แทนค่าลงในสูตร

J เทียบกับ w

$$\frac{\partial J_i}{\partial w_j} = \frac{\partial J_i}{\partial h_i}\frac{\partial h_i}{\partial a_i}\frac{\partial a_i}{\partial w_j}$$

$$\frac{\partial J_i}{\partial b} = \frac{\partial J_i}{\partial h_i}\frac{\partial h_i}{\partial a_i}\frac{\partial a_i}{\partial b}$$

J เทียบกับ B

(หาความชัน)

$$g_{w_j} = \frac{\partial J_i}{\partial w_j} = g_{a_i} x_{i,j}$$

$$g_b = \frac{\partial J_i}{\partial b} = g_{a_i}$$

ได้ค่าอนุพันธ์ของค่าเสียหายเทียบกับ w และ b

(gradient descent)

ปรับค่าพารามิเตอร์

$$\Delta\vec{w} = -\eta\vec{g}_w$$
$$\vec{w}_{\text{ใหม่}} = \vec{w} + \Delta\vec{w}$$

$$\Delta b = -\eta g_b$$
$$b_{\text{ใหม่}} = b + \Delta b$$

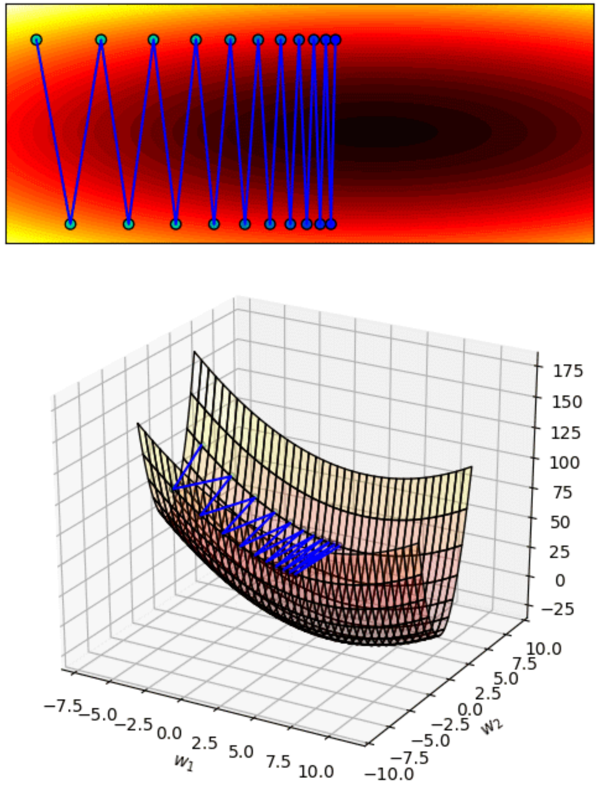
ปรับค่าพารามิเตอร์ตัวนั้นไปในทิศทางที่ทำให้ J ลดลง

gradient descent

*ϕ ในครั้งนี้คำนวณด้วยฟังก์ชันซิกมอยด์

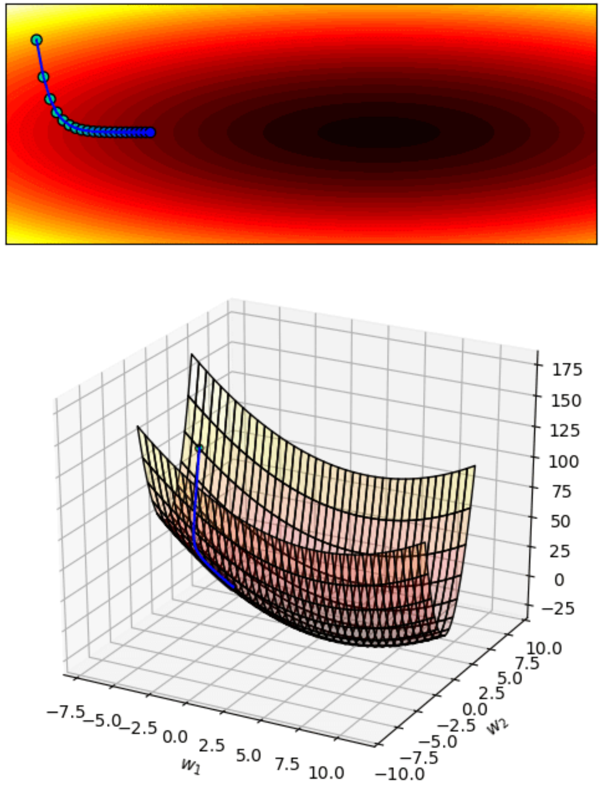
การเคลื่อนลงตามความชัน (gradient descent)

(วิธีการที่พิจารณาหาค่าความชันแล้วปรับพารามิเตอร์ไปในทิศทางตามความชัน)



(η) ถ้ากำหนดให้สูงเกินไป

- อาจจะโดนข้ามค่าที่ต้องการ



(η) ถ้ากำหนดต่ำเกินไป

- เสียเวลามาก

(activation function)

** ฟังก์ชันกระตุ้น จำเป็นต้องเป็นฟังก์ชันต่อเนื่องจึงจะหาอนุพันธ์ของค่าเสียหายได้

การวิเคราะห์การถดถอยโลจิสติก (logistic regression)

(เพอร์เซปตรอนที่มีการปรับพารามิเตอร์โดยใช้ฟังก์ชันซิกมอยด์เป็นฟังก์ชันกระตุ้น)

การเคลื่อนลงตามความชัน (gradient descent)

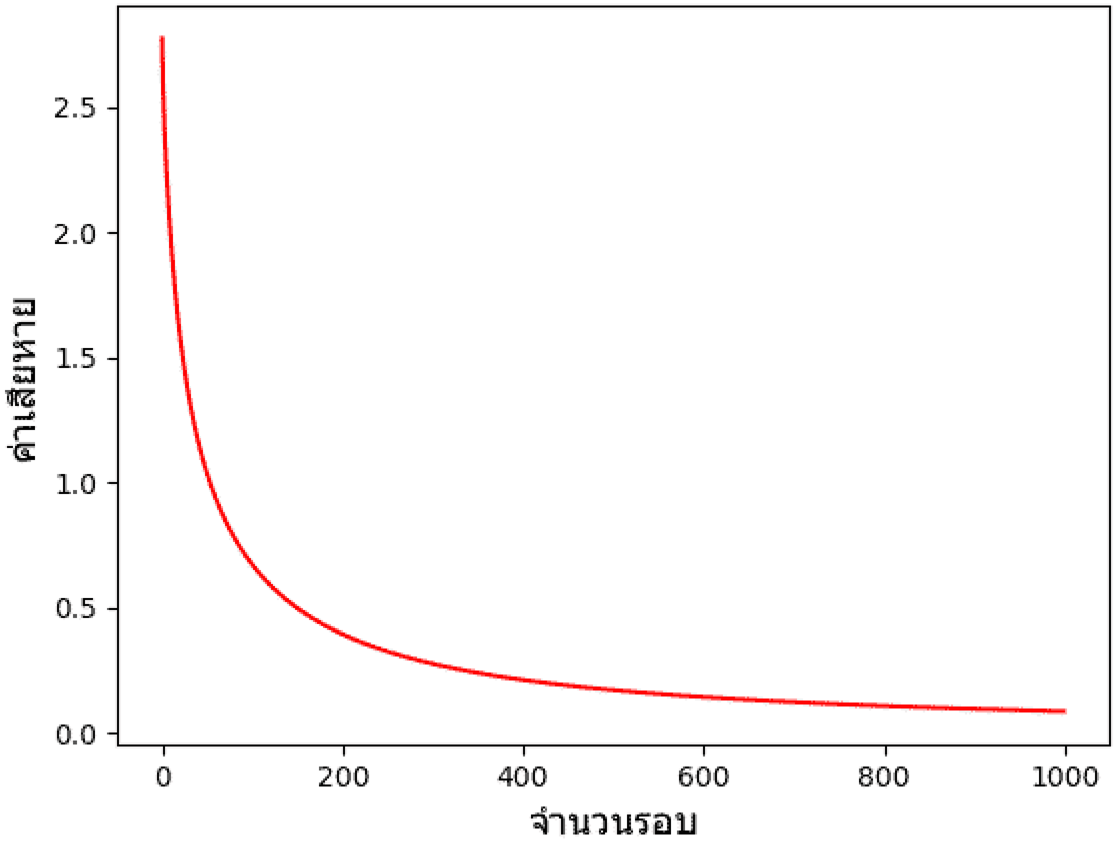
(คำนวณค่าเสียหายรวมของทุกตัวแล้วจึงนำมาเฉลี่ยแล้วค่อยปรับพารามิเตอร์ทีเดียว)

การเคลื่อนลงตามความชันแบบสุ่ม (stochastic gradient descent)

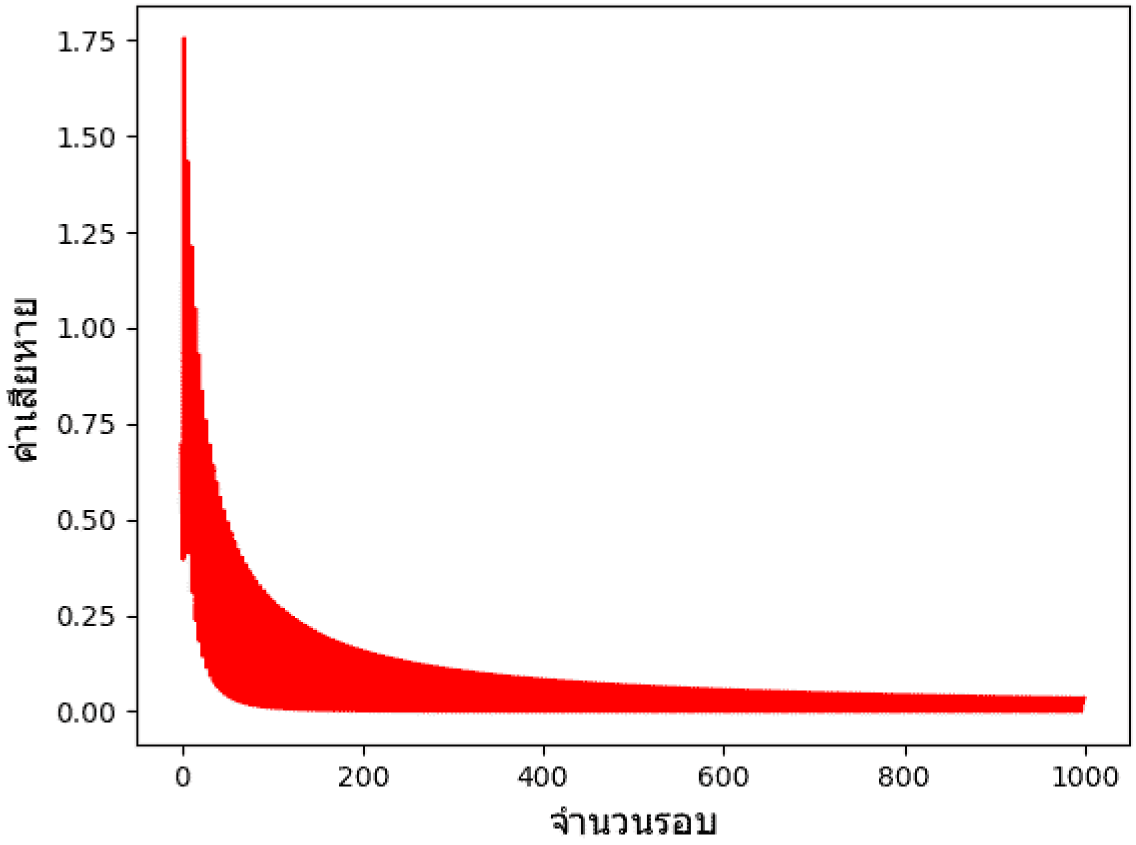
(การคำนวณค่าเสียหายทีละจุดแล้วปรับพารามิเตอร์ทันที ** ข้อดีคือเร็ว)

มินิแบตช์ (minibatch)

(แบ่งข้อมูลเป็นกลุ่มเล็กๆแล้วคำนวณทีละกลุ่ม **นิยมใช้)



การเคลื่อนลงตามความชัน (gradient descent)
(คำนวณค่าเสียหายรวมของทุกตัวแล้วจึงนำมาเฉลี่ยแล้วค่อยปรับพารามิเตอร์ทีเดียว)



การเคลื่อนลงตามความชันแบบสุ่ม (stochastic gradient descent)
(การคำนวณค่าเสียหายทีละจุดแล้วปรับพารามิเตอร์ทันที ** ข้อดีคือเร็ว)

อนุพันธ์และกฎลูกโซ่ของอาเรย์

(อนุพันธ์ของค่าเสียหายเทียบกับน้ำหนักรูปของอาเรย์)

ความสัมพันธ์ของขนาดเมทริกซ์เวลาคูณกันเป็น [m×n][n×1] = [m×1]
ขนาดมิติหลังของตัวแรกกับมิติแรกของตัวหลังจะต้องเท่ากัน
อาเรย์ขนาดเท่ากับมิติแรกของตัวแรกและมิติหลังของตัวหลัง

$$\frac{\partial J}{\partial \vec{w}} = \begin{bmatrix} \frac{\partial J}{\partial w_0} \\ \frac{\partial J}{\partial w_1} \\ \vdots \\ \frac{\partial J}{\partial w_{m-1}} \end{bmatrix}$$

m คือจำนวนตัวแปรต้น

$\partial J/\partial a_i$ เป็นอาเรย์หนึ่งมิติขนาด n
$\partial a_i/\partial w_j$ เป็นอาเรย์สองมิติขนาด n×m
แต่ผลลัพธ์ที่ได้คือ $\partial J/\partial w_j$
นั้นจะต้องมีขนาดเป็น m จึงคูณกันปกติไม่ได้
จึงต้องควรจะอยู่ในรูปของการคูณเมทริกซ์

$$\vec{g}_w = \frac{\partial J}{\partial \vec{w}} = \left(\frac{\partial \vec{a}}{\partial \vec{w}}\right)^T \cdot \left(\frac{\partial J}{\partial \vec{h}}\frac{\partial \vec{h}}{\partial \vec{a}}\right) = \mathbf{x}^T \cdot \vec{g}_a$$

```
gw = np.dot(X.T,ga)
```

$$\frac{\partial J}{\partial \vec{w}} = \frac{\partial J}{\partial \vec{h}}\frac{\partial \vec{h}}{\partial \vec{a}}\frac{\partial \vec{a}}{\partial \vec{w}}$$

$$\Delta\vec{w} = -\eta\vec{g}_w$$
$$\Delta b = -\eta g_b$$

นำค่าอนุพันธ์มาใช้ปรับค่า

a และ h เป็นปริมาณที่มีจำนวนตัวแปรเท่ากัน
จึงนำทั้งอาเรย์มาคำนวณกันได้โดยตรง

$$\vec{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

$$\vec{h} = \frac{1}{1+\exp(-\vec{a})}$$

$$\frac{\partial \vec{h}}{\partial \vec{a}} = \vec{h}(1-\vec{h})$$

a คำนวณจาก w และ b

$$\vec{a} = \mathbf{x}\cdot\vec{w} + b$$

อาเรย์มาคูณกันแบบเมทริกซ์ np.dot()

a ขนาดข้อมูลเป็น n
w ขนาดข้อมูลเป็น m

จึงแจกแจงเป็นอาเรย์ 2 มิติ

a แนวตั้ง
w แนวนอน

$$\frac{\partial \vec{a}}{\partial \vec{w}} = \begin{bmatrix} \frac{\partial a_0}{\partial w_0} & \frac{\partial a_0}{\partial w_1} & \cdots & \frac{\partial a_0}{\partial w_{m-1}} \\ \frac{\partial a_1}{\partial w_0} & \frac{\partial a_1}{\partial w_1} & \cdots & \frac{\partial a_1}{\partial w_{m-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial a_{n-1}}{\partial w_0} & \frac{\partial a_{n-1}}{\partial w_1} & \cdots & \frac{\partial a_{n-1}}{\partial w_{m-1}} \end{bmatrix} = \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,m-1} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1,0} & x_{n-1,1} & \cdots & x_{n-1,m-1} \end{bmatrix}$$

ค่าเท่ากับอาเรย์ X

$$\frac{\partial \vec{a}}{\partial \vec{w}} = \mathbf{x}$$

$$\vec{g}_a = \frac{\partial J}{\partial \vec{a}} = \frac{\partial J}{\partial \vec{h}}\frac{\partial \vec{h}}{\partial \vec{a}} = \frac{1}{n}(\vec{h}-\vec{z})$$

```
ga = (h-z)/len(z)
```

$$\frac{\partial \vec{a}}{\partial b} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \vec{1}_n$$

b เป็นเลขตัวเดียว
a เป็นอาเรย์
ทุกตัวเป็น 1

```
gb = ga.sum()
```

$$g_b = \frac{\partial J}{\partial b} = \vec{g}_a^T \cdot \vec{1}_n = \mathbf{sum}(\vec{g}_a)$$

อนุพันธ์ของค่าเสียหายเทียบกับ b

อาเรย์

$$\vec{h} = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{n-1} \end{bmatrix}$$

J คำนวณจาก h แต่ละค่า

$$J = \frac{1}{n}\sum_{j=0}^{m-1}(z_i \ln h_i + (1-z_i)\ln(1-h_i))$$

$$\frac{\partial J}{\partial h_i} = \frac{h_i - z_i}{nh_i(1-h_i)}$$

$$\frac{\partial J}{\partial \vec{h}} = \begin{bmatrix} \frac{\partial J}{\partial h_0} \\ \frac{\partial J}{\partial h_1} \\ \vdots \\ \frac{\partial J}{\partial h_{n-1}} \end{bmatrix} = \begin{bmatrix} \frac{h_0-z_0}{nh_0(1-h_0)} \\ \frac{h_1-z_1}{nh_1(1-h_1)} \\ \vdots \\ \frac{h_{n-1}-z_{n-1}}{nh_{n-1}(1-h_{n-1})} \end{bmatrix}$$

$$a_i = a(\vec{x}_i) = \sum_{j=0}^{m-1} w_i x_{i,j} + b$$

```
a = np.dot(X,w) + b
```

sigmoid

$$h_i = \textbf{sigmoid}(a_i) = \frac{1}{1+\exp(-a_i)}$$

```
h = sigmoid(a)
```

```
def sigmoid(x):
    return 1/(1+np.exp(-x))
```

(activation function)

z = คำตอบที่ถูกต้อง

```
z = np.arange(2).repeat(900) # คำตอบ เลข 0 และ 1
```

w = ค่าน้ำหนักที่สุ่มหรือคำนวณ

```
w = np.array([0,0.])
```

```
w -= eta*gw
```

b = ค่า bias ที่สุ่มหรือคำนวณ

```
b = 0
```

```
b -= eta*gb
```

$$J_i = -z_i \ln h_i - (1-z_i)\ln(1-h_i)$$

```
J = ha_entropy(z,h)
```

```
def ha_entropy(z,h):
    return -(z*np.log(h)+(1-z)*np.log(1-h)).mean()
```

(loss function)

$$\vec{g}_w = \frac{\partial J}{\partial \vec{w}} = \left(\frac{\partial \vec{a}}{\partial \vec{w}}\right)^T \cdot \left(\frac{\partial J}{\partial \vec{h}}\frac{\partial \vec{h}}{\partial \vec{a}}\right)$$

$$= \mathbf{x}^T \cdot \vec{g}_a$$

```
gw = np.dot(X.T,ga)
```

$$\vec{g}_a = \frac{\partial J}{\partial \vec{a}} = \frac{\partial J}{\partial \vec{h}}\frac{\partial \vec{h}}{\partial \vec{a}} = \frac{1}{n}(\vec{h} - \vec{z})$$

```
ga = (h-z)/len(z)
```

$$g_b = \frac{\partial J}{\partial b} = \vec{g}_a^T \cdot \vec{1}_n = \textbf{sum}(\vec{g}_a)$$

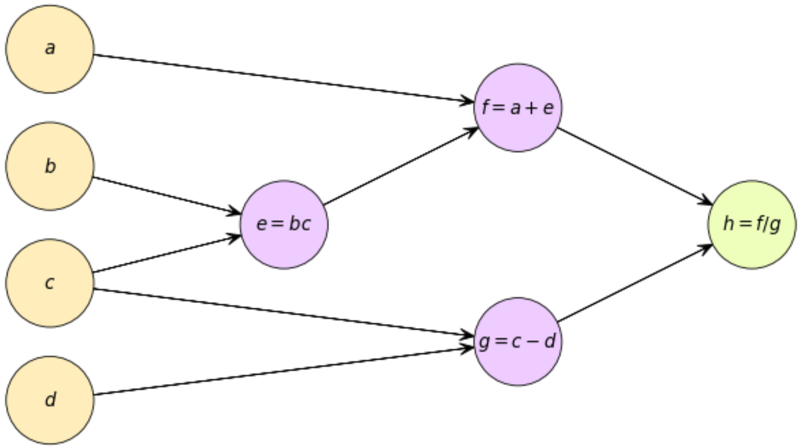
```
gb = ga.sum()
```

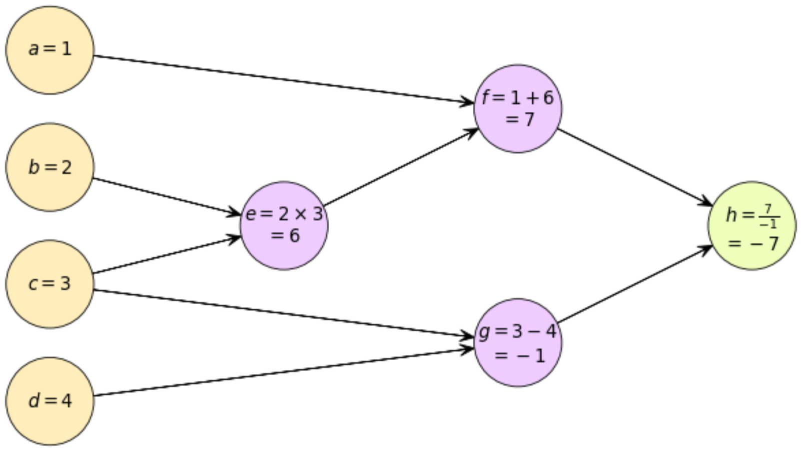
กราฟคำนวณ(computational graph)

(กราฟคำนวณก็ทำให้เรามองเห็นความสัมพันธ์ต่างๆเพื่อคำนวณอนุพันธ์ตามกฎลูกโซ่ได้ง่าย)

$$h = \frac{a+bc}{c-d}$$

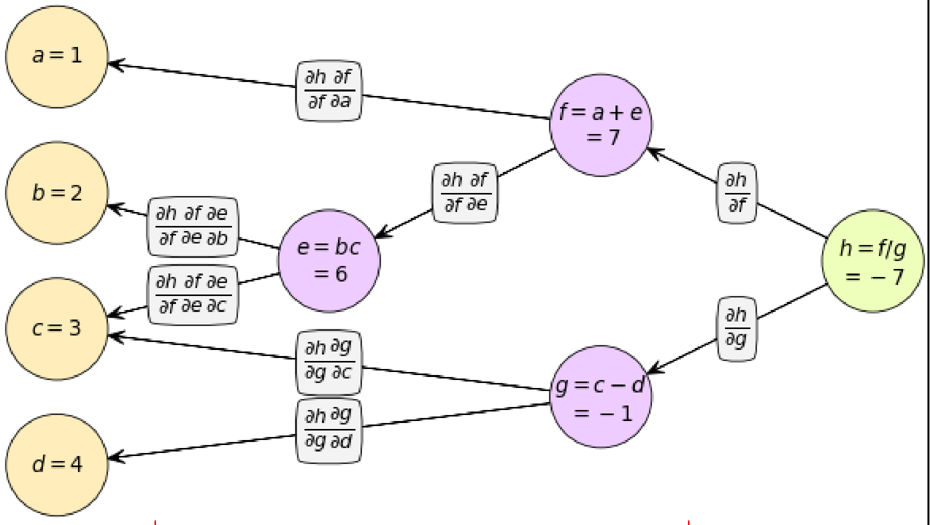
$$e = bc$$
$$f = a + e$$
$$g = c - d$$
$$h = \frac{f}{g}$$





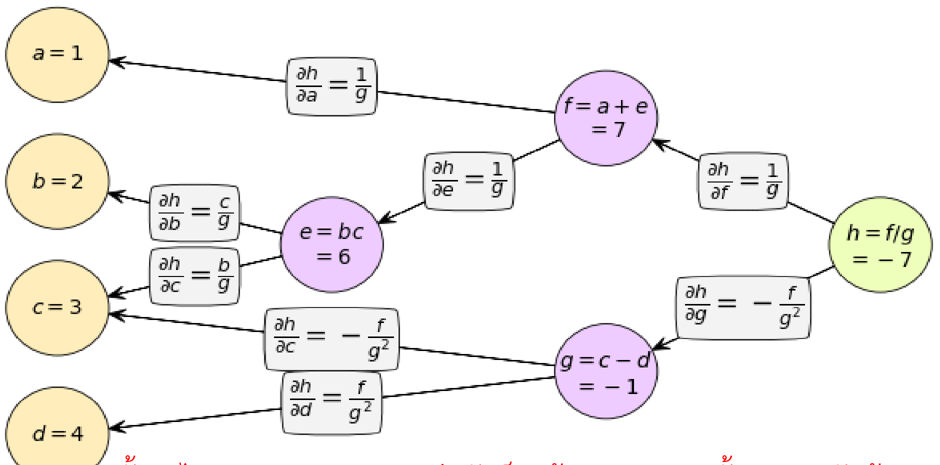
การแพร่ย้อนกลับ(backpropagation)

(การไล่คำนวณค่าอนุพันธ์เป็นลำดับขั้นจากกราฟ)

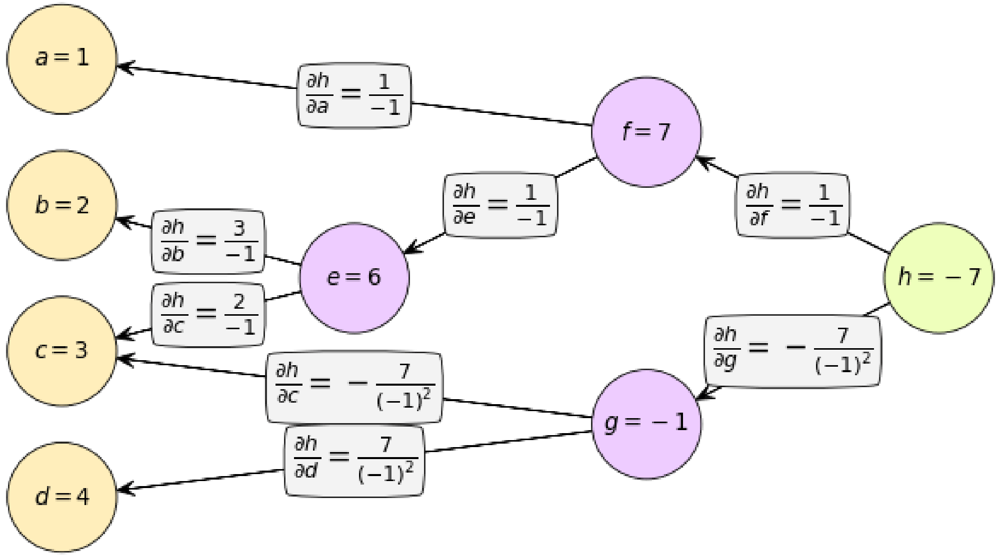


h เป็นจุดเริ่มต้น หาอนุพันธ์เทียบ f และ g แล้วก็โยงต่อไปเรื่อยๆจนถึง a,b,c,d

$$
\begin{aligned}
\frac{\partial h}{\partial g} &= -\frac{f}{g^2} \\
\frac{\partial h}{\partial f} &= \frac{1}{g} \\
\frac{\partial h}{\partial e} &= \frac{\partial h}{\partial f}\frac{\partial f}{\partial e} = \frac{1}{g}\times 1 \\
\frac{\partial h}{\partial d} &= \frac{\partial h}{\partial g}\frac{\partial g}{\partial d} = -\frac{f}{g^2}\times(-1) \\
\frac{\partial h}{\partial c} &= \frac{\partial h}{\partial f}\frac{\partial f}{\partial e}\frac{\partial e}{\partial c} + \frac{\partial h}{\partial g}\frac{\partial g}{\partial c} = \frac{1}{g}\times 1\times b - \frac{f}{g^2}\times 1 \\
\frac{\partial h}{\partial b} &= \frac{\partial h}{\partial f}\frac{\partial f}{\partial e}\frac{\partial e}{\partial b} = \frac{1}{g}\times 1\times c \\
\frac{\partial h}{\partial a} &= \frac{\partial h}{\partial f}\frac{\partial f}{\partial a} = \frac{1}{g}\times 1
\end{aligned}
$$

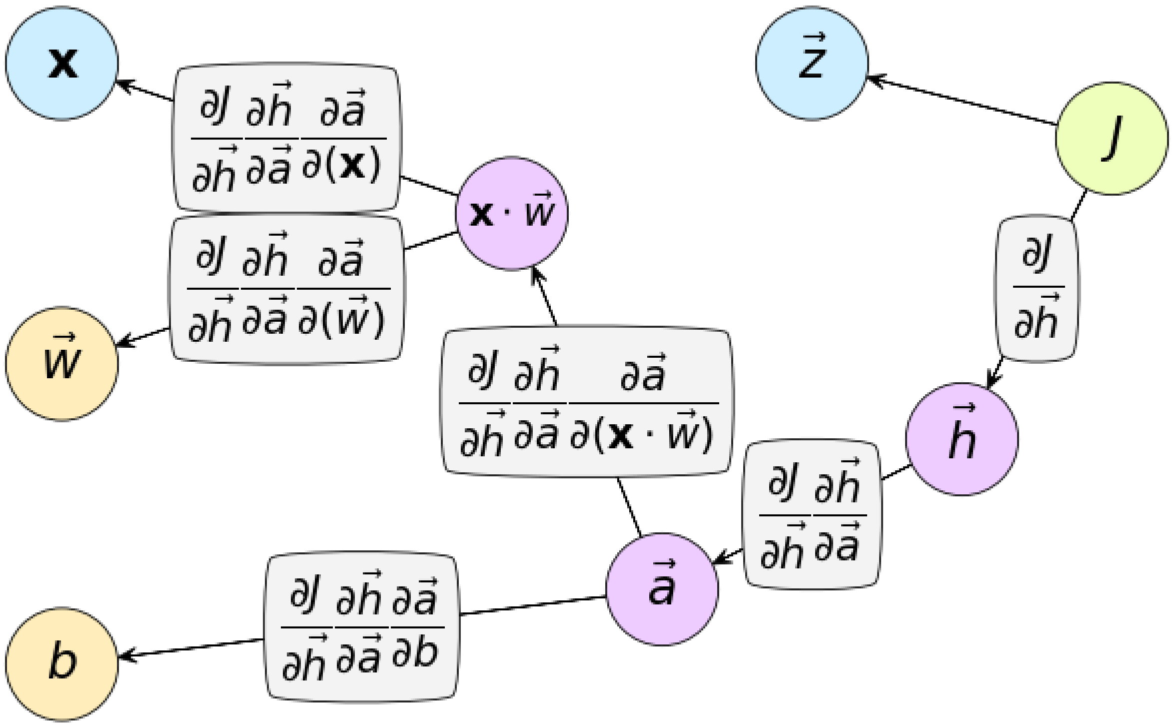


c นั้นขาไปเดิน ๒ สาย ตอนแพร่กลับก็จะต้องเอา ๒ สายนั้นมาบวกกันด้วย

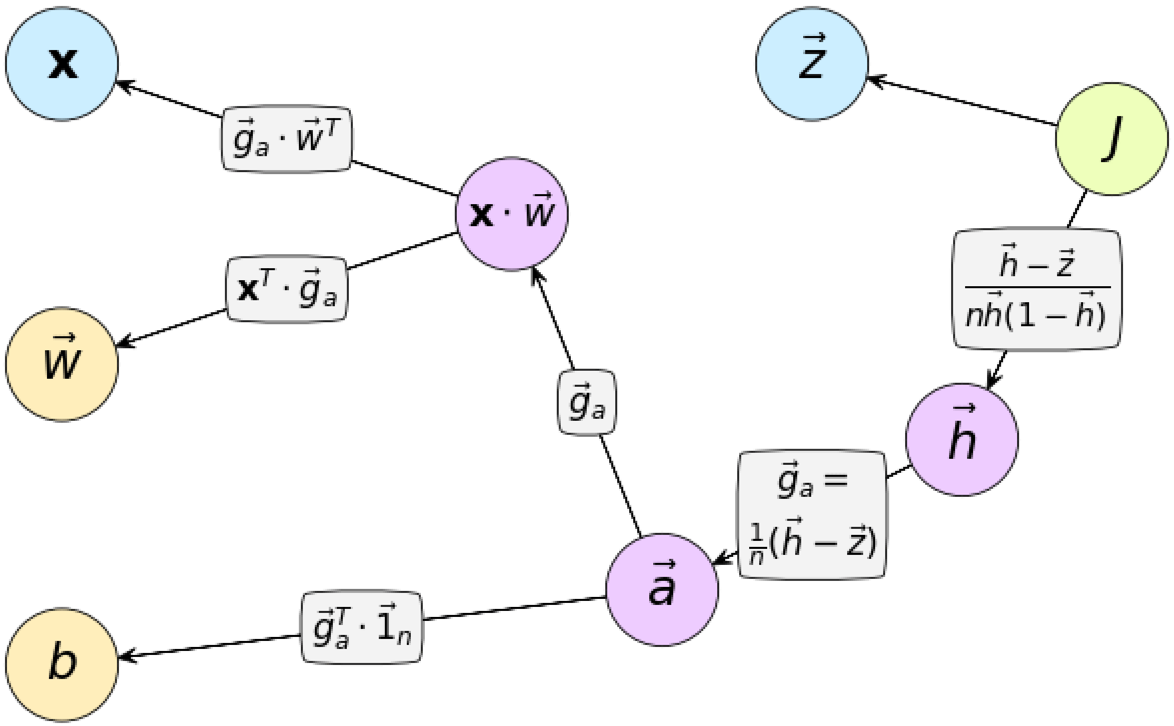


กราฟคำนวณของเพอร์เซปตรอนชั้นเดียว

กราฟคำนวณของเพอร์เซปตรอนที่ใช้ฟังก์ชันซิกมอยด์



แทนค่าอนุพันธ์ต่างๆ



ตัวแปรต้นของเราคือ x,w,b ส่วนตัวแปรปลายทางคือค่าเสียหาย J

h เป็นค่าที่คำนวณได้ระหว่างทาง และมันก็ถูกใช้ตอนที่หาอนุพันธ์ของ J เทียบกับ w และ b ด้วย

การวิเคราะห์จำแนกประเภทหลายกลุ่ม

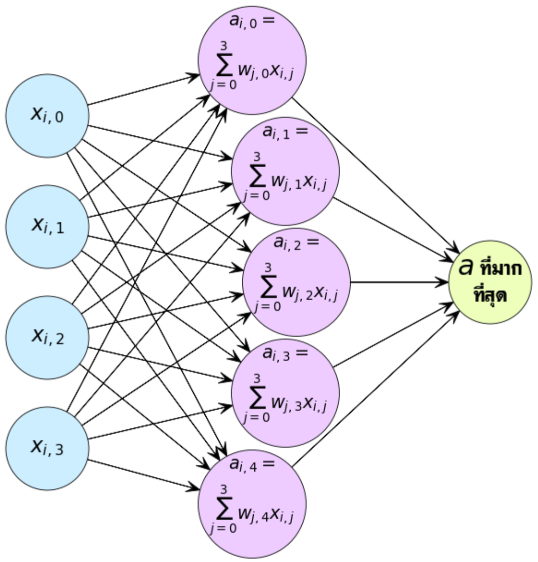
การคำนวณหาค่า a อาจเขียนในรูปของการคูณเมทริกซ์

m_0 เป็นจำนวนตัวแปรต้น
จำนวนมิติข้อมูลขาเข้า

m_1 เป็นจำนวนกลุ่มที่แบ่ง
จำนวนมิติของค่าขาออก

อาเรย์ค่าน้ำหนัก w
มีขนาดเป็น $m_0 \times m_1$

ทำหน้าที่เป็นตัวเปลี่ยน
ถ่ายระหว่างข้อมูลขาเข้ากับขาออก



$$\vec{a}_i = \mathbf{w}^T \cdot \vec{x}_i + \vec{b}$$

$$\begin{bmatrix} a_{i,0} \\ a_{i,1} \\ \vdots \\ a_{i,m_1-1} \end{bmatrix} = \begin{bmatrix} w_{0,0} & w_{1,0} & \cdots & w_{m_0-1,0} \\ w_{0,1} & w_{1,1} & \cdots & w_{m_0-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{0,m_1-1} & w_{1,m_1-1} & \cdots & w_{m_0-1,m_1-1} \end{bmatrix} \cdot \begin{bmatrix} x_{i,0} \\ x_{i,1} \\ \vdots \\ x_{i,m_0-1} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m_1-1} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j=0}^{m_0-1} w_{j,0} x_{i,j} + b_0 \\ \sum_{j=0}^{m_0-1} w_{j,1} x_{i,j} + b_1 \\ \vdots \\ \sum_{j=0}^{m_0-1} w_{j,m_1-1} x_{i,j} + b_{m_1-1} \end{bmatrix}$$

เมื่อได้ค่า a แล้วก็ดูว่าค่าไหนหลักไหนในแต่ละแถวมีค่ามากสุด
คำตอบก็คือกลุ่มนั้นสำหรับในแถวนั้น

ไพธอนใช้ .argmax เพื่อหาว่าแถวไหนมีค่าสูงสุดได้
โดยต้องระบุแกนเป็น axis=1 จะได้ค่าสูงสุดในแต่ละแถว

a และ x เป็นข้อมูลหลายค่าหลายตัวแปร
ดังนั้นจึงอยู่ในรูปของอาเรย์สองมิติ

$$\boldsymbol{a} = \mathbf{x} \cdot \mathbf{w} + \vec{b}^T$$

$$\boldsymbol{a} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,m_0-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,m_1-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,m_1-1} \end{bmatrix}$$

ขนาดของ a, x, w สอดคล้องกันดี $[n,m_1] = [n,m_0][m_0,m_1]$

$$\mathbf{x} \cdot \mathbf{w} + \vec{b}^T = \mathbf{x} \cdot \mathbf{w} + \begin{bmatrix} b_0 & b_1 & \cdots & b_{m_1-1} \end{bmatrix}$$

$$= \mathbf{x} \cdot \mathbf{w} + \begin{bmatrix} b_0 & b_1 & \cdots & b_{m_1-1} \\ b_0 & b_1 & \cdots & b_{m_1-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_0 & b_1 & \cdots & b_{m_1-1} \end{bmatrix}$$

b เป็นอาเรย์หนึ่งมิติ
เวลานำมาบวกกันจะเท่ากับเป็นการบวกทุกแถว
ตามคุณสมบัติของอาเรย์

ซอฟต์แม็กซ์ (softmax)

(ใช้แทนค่าความน่าจะเป็นแบบหลายกลุ่ม)

$$\mathbf{h} = \mathbf{softmax}(\mathbf{a}) = \frac{\exp(\boldsymbol{a})}{\sum_{k=0}^{m_1-1} \exp(\vec{a}_{:,k})}$$

เขียนในรูปของแต่ละค่าได้เป็น

$$h_{i,k} = \frac{\exp(a_{i,k})}{\sum_{l=0}^{m_i-1} \exp(a_{i,l})}$$

ฟังก์ชันซอฟต์แม็กซ์แต่ละแถวจะบวกกันแล้วได้ 1
ใช้แทนค่าความน่าจะเป็นของแต่ละตัว

ในทำนองเดียวกับซิกมอยด์

```
def softmax(x):
    exp_x = np.exp(x.T-x.max(1))
    return (exp_x/exp_x.sum(0)).T
```

เอนโทรปีไขว้ (แบบกลุ่ม)

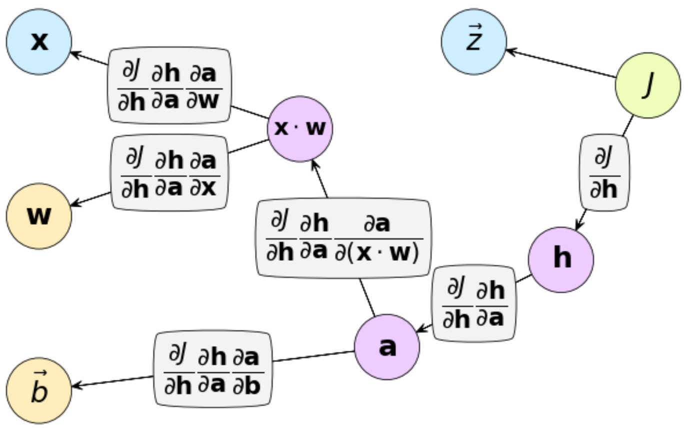
คำนวณค่าเสียหาย

$$J = -\frac{1}{n}\sum_{i=0}^{n-1}\sum_{k=0}^{m_1-1} z_{i,k} \ln h_{i,k}$$
$$= -\frac{1}{n}\mathbf{sum}(\mathbf{z}\ln\mathbf{h})$$

z = ค่าคำตอบในรูปของ**วันฮ็อต** (**one-hot**)

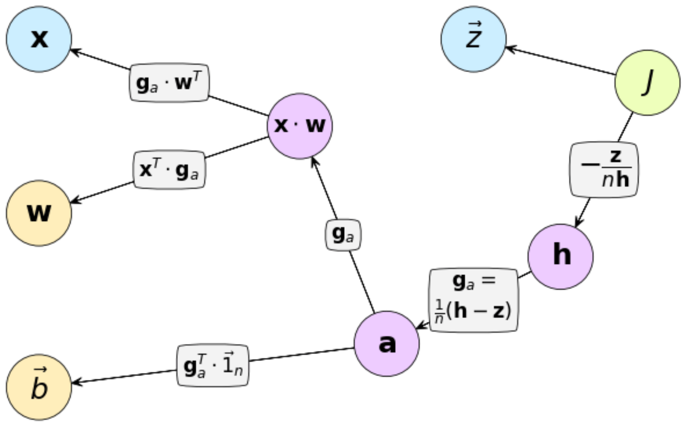
```
def ha_1h(z,n):
    return (z[:,None]==range(n))
```

การแพร่ย้อนกลับเพื่อหาอนุพันธ์ของค่าน้ำหนักและไบแอส



$$\frac{\partial J}{\partial \mathbf{h}} = -\frac{\mathbf{z}}{n\mathbf{h}}$$
$$\frac{\partial \mathbf{h}}{\partial \boldsymbol{a}} = \frac{\mathbf{h}(\mathbf{z}-\mathbf{h})}{\mathbf{z}}$$
$$\mathbf{g}_a = \frac{\partial J}{\partial \boldsymbol{a}} = \frac{1}{n}(\mathbf{h}-\mathbf{z})$$
$$\frac{\partial \boldsymbol{a}}{\partial \mathbf{w}} = \mathbf{x}$$
$$\frac{\partial \boldsymbol{a}}{\partial \vec{b}^T} = \vec{1}_n = \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}$$
$$\mathbf{g}_w = \frac{\partial J}{\partial \mathbf{w}} = \mathbf{x}^T\cdot\mathbf{g}_a$$
$$\vec{g}_b = \frac{\partial J}{\partial \vec{b}} = \mathbf{g}_a^T\cdot\vec{1}_n = \sum_{i=0}^{n-1}\mathbf{g}_{a,i}$$

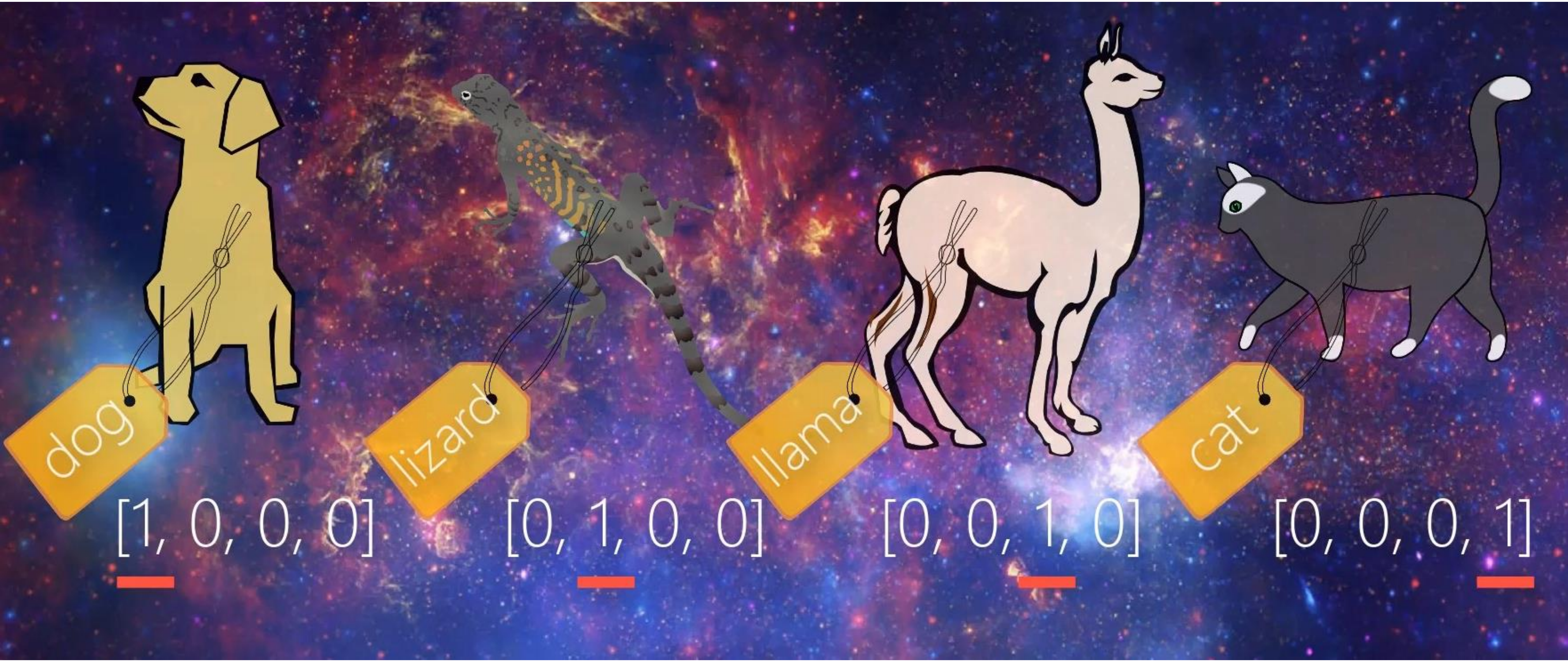
แทนลงในกราฟคำนวณ



$$\Delta\mathbf{w} = -\eta\mathbf{g}_w$$
$$\Delta\vec{b} = -\eta\vec{g}_b$$

นำค่าอนุพันธ์มาใช้ปรับค่าน้ำหนักและไบแอส

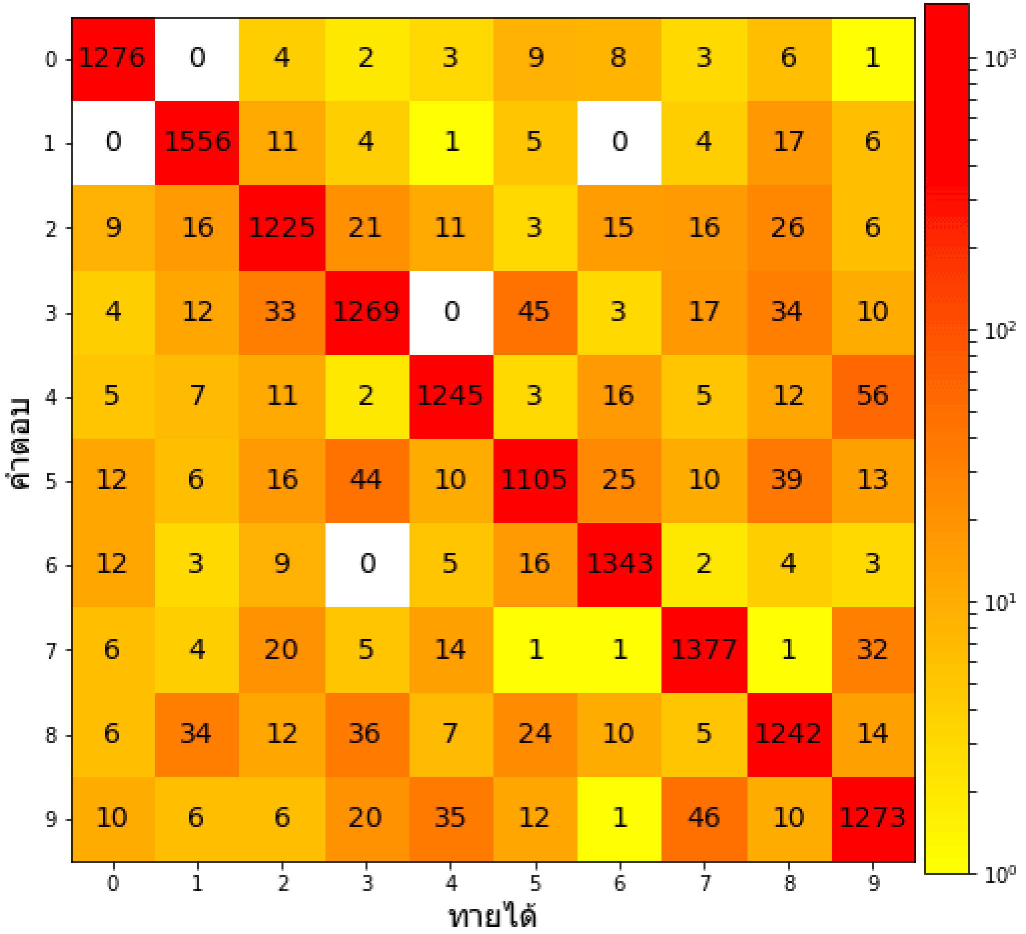
วันฮ็อต (one-hot)



เมทริกซ์ความสับสน (confusion matrix)

(ตารางที่สร้างขึ้นมาเพื่อสำหรับไว้ดูเทียบว่ามีการทายสับสนระหว่างตัวไหนกับตัวไหน)

```
[1276    0    4    2    3    9    8    3    6    1]
[   0 1556   11    4    1    5    0    4   17    6]
[   9   16 1225   21   11    3   15   16   26    6]
[   4   12   33 1269    0   45    3   17   34   10]
[   5    7   11    2 1245    3   16    5   12   56]
[  12    6   16   44   10 1105   25   10   39   13]
[  12    3    9    0    5   16 1343    2    4    3]
[   6    4   20    5   14    1    1 1377    1   32]
[   6   34   12   36    7   24   10    5 1242   14]
[  10    6    6   20   35   12    1   46   10 1273]
```



แนวตั้งคือคำตอบจริง แนวนอนคือผลที่ทาย ค่าที่อยู่ในแนวทแยงคือที่ทายถูก ไล่จากบนลงล่าง และซ้ายไปขวา จาก 0 ถึง 9 ตามลำดับ

```python
def confusion_matrix(z1,z2):
    n = max(z1.max(),z2.max())+1
    return np.dot((z1==np.arange(n)[:,None]).astype(int),(z2[:,None]==np.arange(n)).astype(int))
```

```python
from sklearn.metrics import confusion_matrix
conma = confusion_matrix(z_truat,z_thamnai)
[print(c) for c in conma]
```

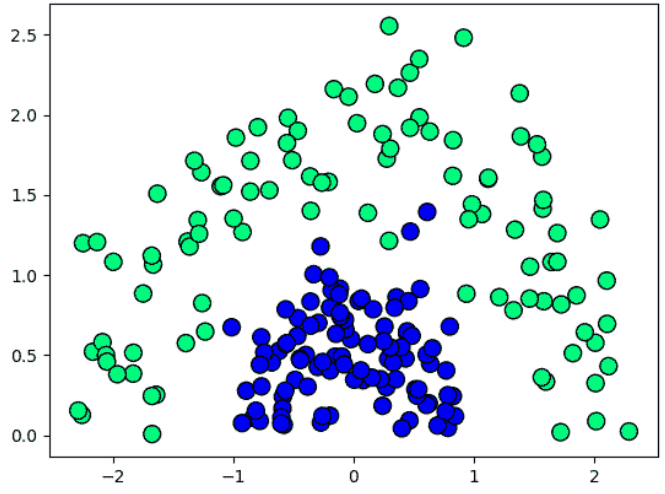
เพอร์เซปตรอนหลายชั้น (multi-layer neural network)

เพอร์เซปตรอนสองชั้น

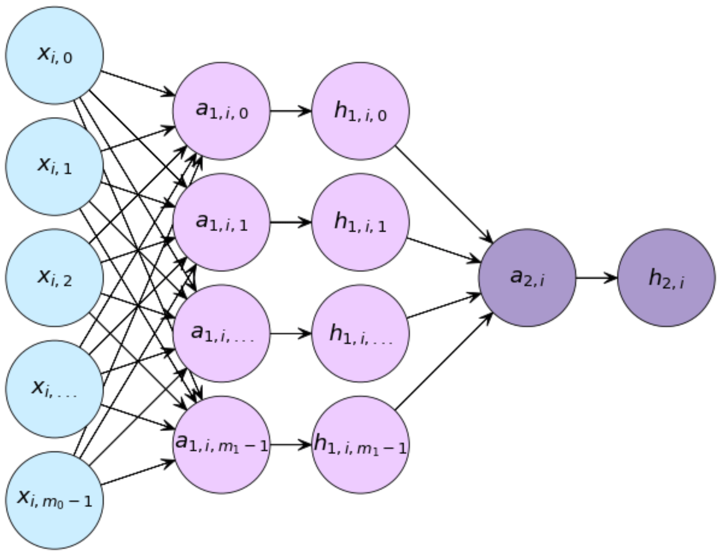
นำเซลล์ประสาทมาต่อกันเป็น ๒ ชั้นจะทำให้สามารถคำนวณ

แบบไม่เป็นเชิงเส้นได้

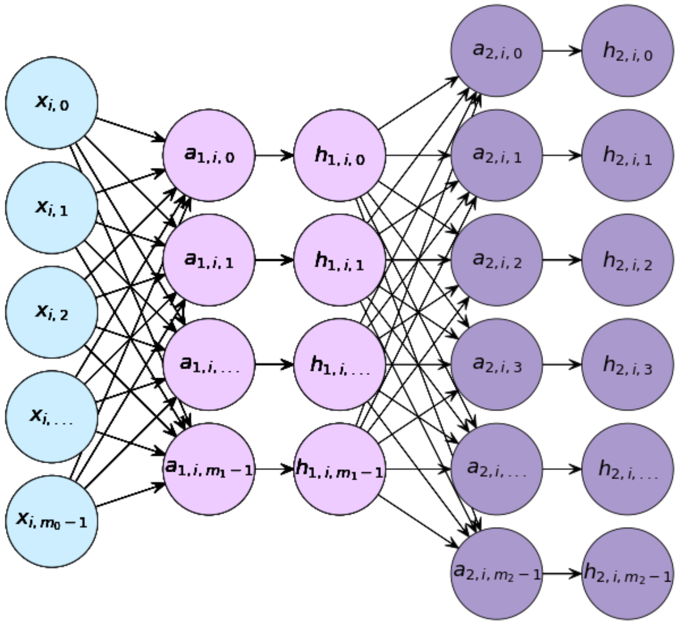
ต้องใช้ฟังก์ชันกระตุ้นระหว่างชั้น



ไม่เป็นเชิงเส้น (nonlinear)



กรณีปัญหาการจำแนกประเภท ๒ กลุ่ม

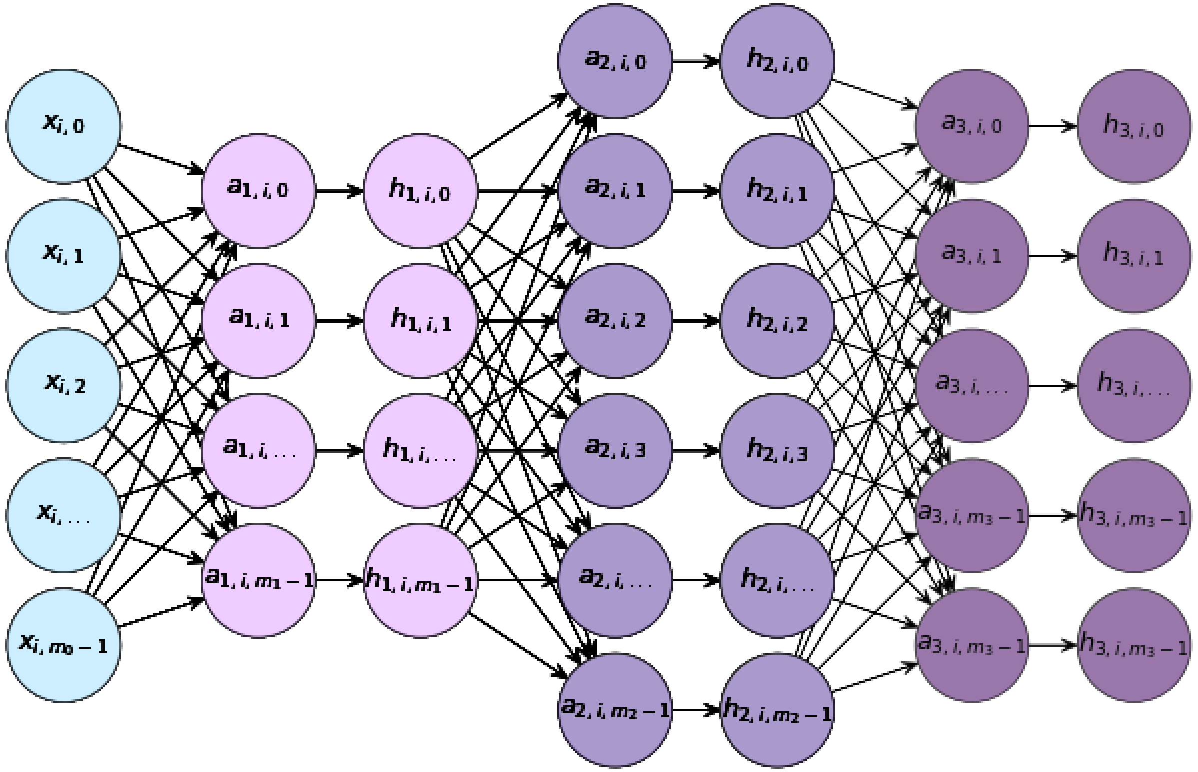


กรณีจำแนกประเภทหลายกลุ่ม

แต่ละชั้นก็มีค่า w, b, a, h ของตัวเอง เลข 1 และ 2 ที่ห้อยอยู่แสดงถึงว่าเป็นชั้นที่เท่าไหร่

แต่ละชั้นจะประกอบไปด้วยส่วนคำนวณเชิงเส้น และตามด้วยฟังก์ชันกระตุ้น ชั้นแรกมี x เป็นค่าขาเข้า และ h1 เป็นค่าขาออก ชั้นสองมี h1 เป็นค่าขาเข้าและ h2 เป็นค่าขาออก

เพอร์เซปตรอนสามชั้น



$$\boldsymbol{a}_1 = \mathbf{x} \cdot \mathbf{w}_1 + \vec{b}_1^T \qquad \boldsymbol{a}_2 = \mathbf{h}_1 \cdot \mathbf{w}_2 + \vec{b}_2^T \qquad \boldsymbol{a}_3 = \mathbf{h}_2 \cdot \mathbf{w}_3 + \vec{b}_3^T$$

$$\mathbf{h}_1 = \phi(\boldsymbol{a}_1) \quad \rightarrow \quad \mathbf{h}_2 = \phi(\boldsymbol{a}_2) \quad \rightarrow \quad \mathbf{h}_3 = \phi(\boldsymbol{a}_3)$$

ϕ หมายถึงฟังก์ชันกระตุ้น

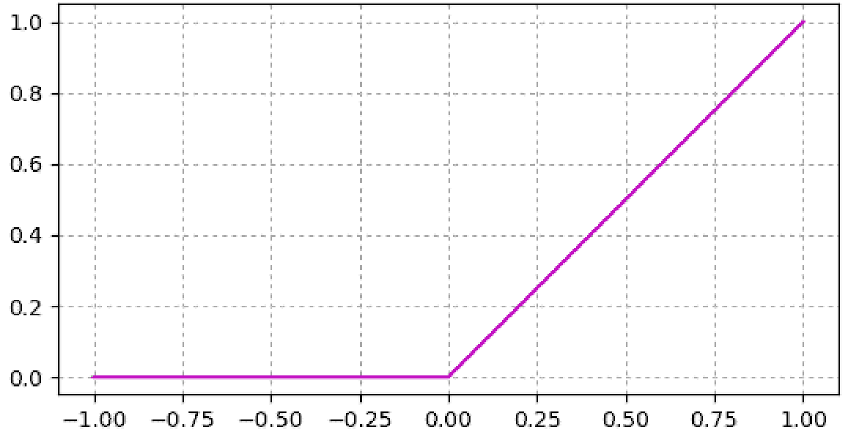
จะกี่ชั้นการคำนวณก็เป็นแบบนี้ต่อไปเรื่อยๆ

ฟังก์ชันกระตุ้นระหว่างชั้น

ReLU (Rectified Linear Unit)

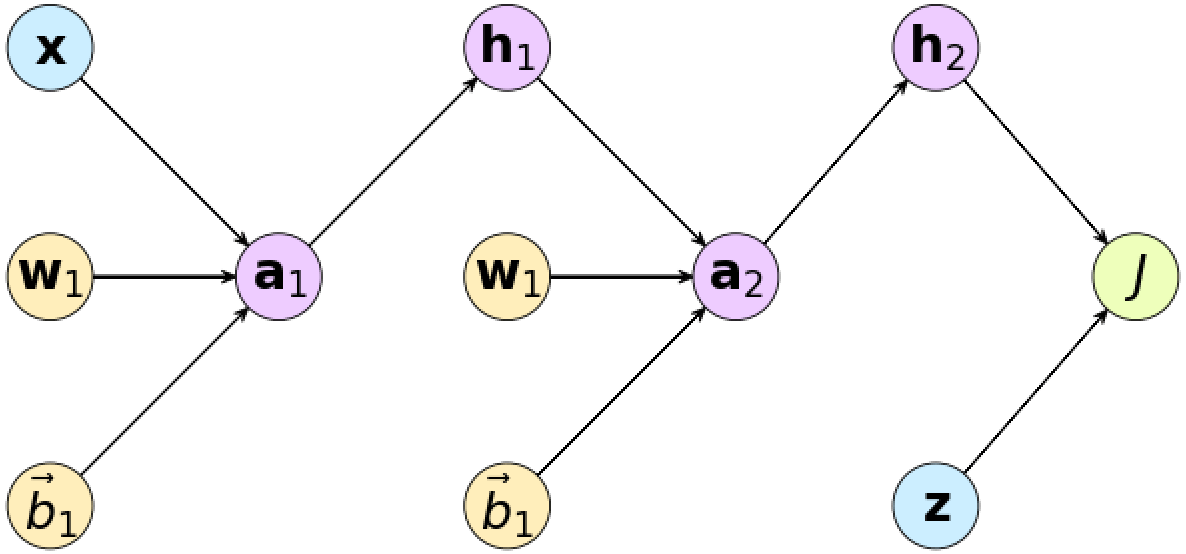
$$\mathbf{ReLU}(a) = \begin{cases} a & \text{ถ้า } a \geq 0 \\ 0 & \text{ถ้า } a \leq 0 \end{cases}$$

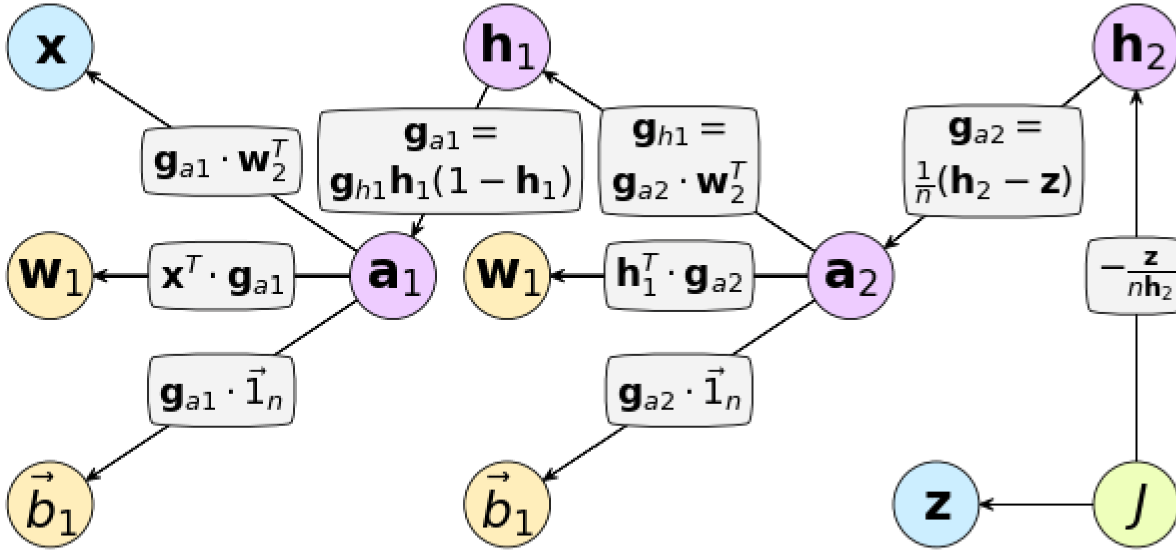
```python
def relu(X):
    return np.maximum(0,X)
```



เพอร์เซปตรอนหลายชั้น (multi-layer neural network)

การเรียนรู้ของเพอร์เซปตรอนหลายชั้น





แทนลงกราฟคำนวณ

$$\boldsymbol{a}_1 = \mathbf{x}\cdot\mathbf{w}_1 + \vec{b}_1^T$$
$$\mathbf{h}_1 = \frac{1}{1+\exp(-\boldsymbol{a}_1)}$$
$$\boldsymbol{a}_2 = \boldsymbol{h}_1\cdot\mathbf{w}_2 + \vec{b}_2^T$$
$$\mathbf{h}_2 = \frac{1}{1+\exp(-\boldsymbol{a}_2)}$$
$$J = \frac{1}{n}\mathbf{sum}(\mathbf{z}\ln\mathbf{h})$$

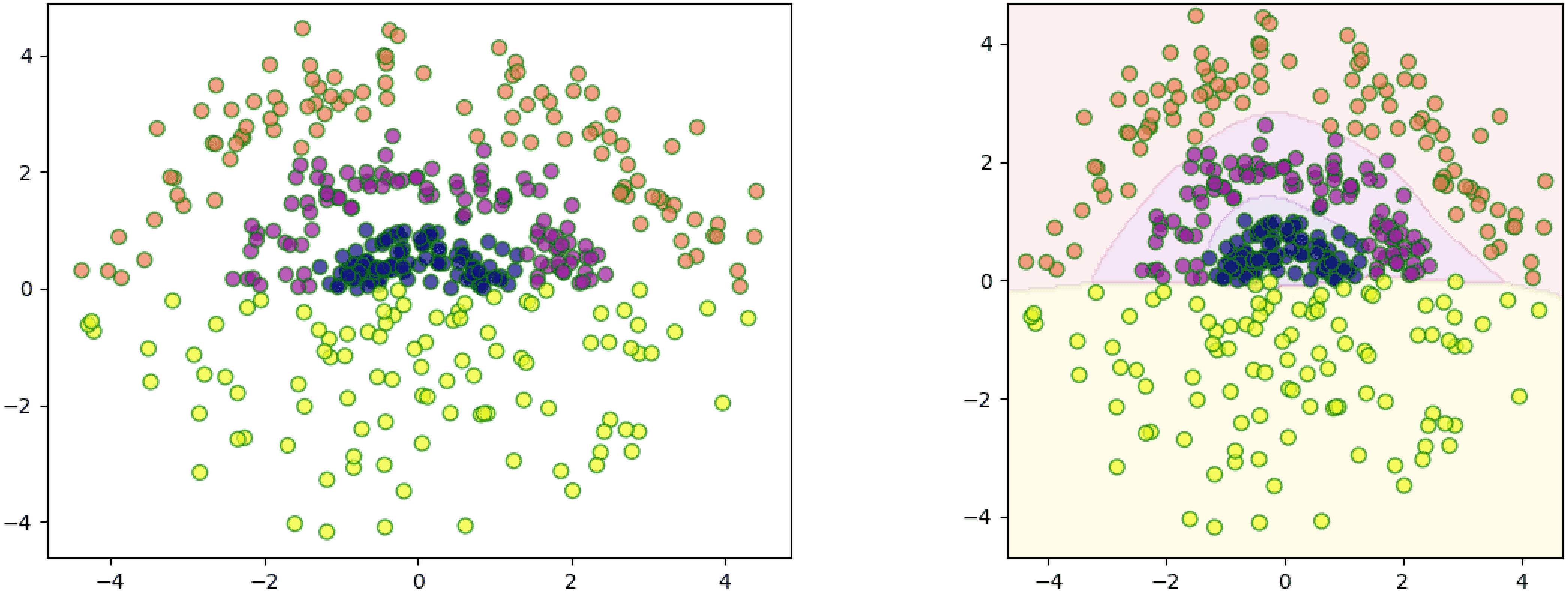
หาอนุพันธ์ →

$$\mathbf{g}_{h_2} = \frac{\partial J}{\partial \mathbf{h}_2} = \frac{\mathbf{z}}{n\mathbf{h}_2}$$
$$\mathbf{g}_{a_2} = \frac{\partial J}{\partial \boldsymbol{a}_2} = \frac{1}{n}(\mathbf{h}_2 - \mathbf{z})$$
$$\mathbf{g}_{w_2} = \frac{\partial J}{\partial \mathbf{w}_2} = \mathbf{h}_1^T\cdot\mathbf{g}_{a_2}$$
$$\vec{g}_{b_2} = \frac{\partial J}{\partial \vec{b}_2} = \mathbf{g}_{a_2}\cdot\vec{1}_n$$
$$\mathbf{g}_{h_1} = \frac{\partial J}{\partial \mathbf{h}_1} = \mathbf{g}_{a_2}\cdot\mathbf{w}_2^T$$
$$\mathbf{g}_{a_1} = \frac{\partial J}{\partial \boldsymbol{a}_1} = \mathbf{g}_{h_1}\mathbf{h}_1(1-\mathbf{h}_1)$$
$$\mathbf{g}_{w_1} = \frac{\partial J}{\partial \mathbf{w}_1} = \mathbf{x}^T\cdot\mathbf{g}_{a_1}$$
$$\vec{g}_{b_1} = \frac{\partial J}{\partial \vec{b}_1} = \mathbf{g}_{a_2}\cdot\vec{1}_n$$

→

$$\Delta\mathbf{w}_2 = -\eta\mathbf{g}_{w_2}$$
$$\Delta\vec{b}_2 = -\eta\vec{g}_{b_2}$$
$$\Delta\mathbf{w}_1 = -\eta\mathbf{g}_{w_1}$$
$$\Delta\vec{b}_1 = -\eta\vec{g}_{b_1}$$

พารามิเตอร์ที่ต้องปรับมีทั้งหมด ๔ ตัว



4
2
0
−2
−4
−4
−2
0
2
4
4
2
0
−2
−4
−4
−2
0
2
4

affine layer

ชั้นที่มีการคูณแบบเมทริกซ์ คำนวณเชิงเส้น

```
class Affin:
  def _init_(self, w, b):
    self.w  = w
    self.b  = b
    self.gw = 0
    self.gb = 0

  def pai(self, X):
    self.X  = X
    return np.dot(X, self.w) + self.b

  def yon(self, g):
    self.gw += np.dot(self.X.T, g)
    self.gb += g.sum(0)
    return np.dot(g, self.w.T)
```

```
class Affin:
  # ขาเข้า (m0) ขาออก (m1) ความกว้างการแจกแจงค่า
(sigma)
   def __init__(self, m0, m1, sigma):
     self.m  = m0, m1
     self.w  = np.random.normal(0,
sigma, self.m)
     self.b  = np.zeros(m1)
     self.gw  = 0
     self.gb  = 0

  def pai(self, X):
     self.X  = X
     return  np.dot(X, self.w) +
self.b

  def yom(self, g):
     self.gw  += np.dot(self.X.T, g)
     self.gb  += g.sum(0)
     return  np.dot(g, self.w.T)
```

การนิยามขณะวิ่ง (define by run)

เส้นทางการคำนวณได้ถูกนิยามขึ้นทันทีในขณะวิ่งคำนวณไปข้างหน้า

การปรับค่าพารามิเตอร์

Adam(Adaptive Moment**)**

$$\vec{m}_t = \beta_1 \vec{m}_{t-1} + (1-\beta_1)\vec{g}(w_t)$$
$$\vec{v}_t = \beta_2 \vec{v}_{t-1} + (1-\beta_2)\vec{g}(w_t)^2$$

$$\Delta \vec{w}_{t+1} = -\eta \frac{\sqrt{1-\beta_2^i}}{1-\beta_1^i} \frac{\vec{m}_t}{\sqrt{\vec{v}_t}}$$

β_1=0.9 และ β_2=0.999

การกำหนดค่าพารามิเตอร์ตั้งต้น

ค่าเริ่มต้นของซาวีเย โกลโร และ เหอ ไข่หมิง

ซาวีเย โกลโร (Xavier Glorot)

ระบุว่าเมื่อใช้ซอฟต์แม็กซ์เป็นฟังก์ชันกระตุ้นแล้ว

ค่า σ ที่เหมาะสมที่สุดคือ

$$\sigma = \frac{1}{\sqrt{m}}$$

ค่าตั้งต้นแบบซาวีเย (Xavier Initialization)

เมื่อใช้ ReLU เป็นฟังก์ชันกระตุ้น

ค่า σ ที่เหมาะสมที่สุดคือ

$$\sigma = \sqrt{\frac{2}{m}}$$

ค่าตั้งต้นแบบเหอ (Hé Initialization)

m คือจำนวนมิติขาเข้าของข้อมูล

การวิเคราะห์การถดถอยเชิงเส้น (linear regression)

z = xw+b

การวิเคราะห์การถดถอยจะไม่มีการใช้ฟังก์ชันกระตุ้น

ค่าความเสียหายเป็นค่าความต่างกำลังสองเฉลี่ย

$$J = \frac{1}{n} \sum_{i=0}^{i=n-1} (h_i - z_i)^2$$

z คือคำตอบจริง ส่วน h คือคำตอบที่คำนวณได้

h ก็มาจากการคำนวณจาก x และ w โดยตรง

$$\vec{h} = \mathbf{x} \cdot \vec{w} + b$$

อนุพันธ์ของค่าเสียหายเทียบกับ w และ b

$$\frac{\partial \vec{h}}{\partial \vec{w}} = \mathbf{x}$$

$$\frac{\partial J}{\partial \vec{h}} = \frac{2}{n}(\vec{h} - \vec{z})$$

$$\frac{\partial J}{\partial \vec{w}} = \frac{\partial J}{\partial \vec{h}} \frac{\partial \vec{h}}{\partial \vec{w}} = \frac{2}{n}\mathbf{x}^T \cdot (\vec{h} - \vec{z})$$