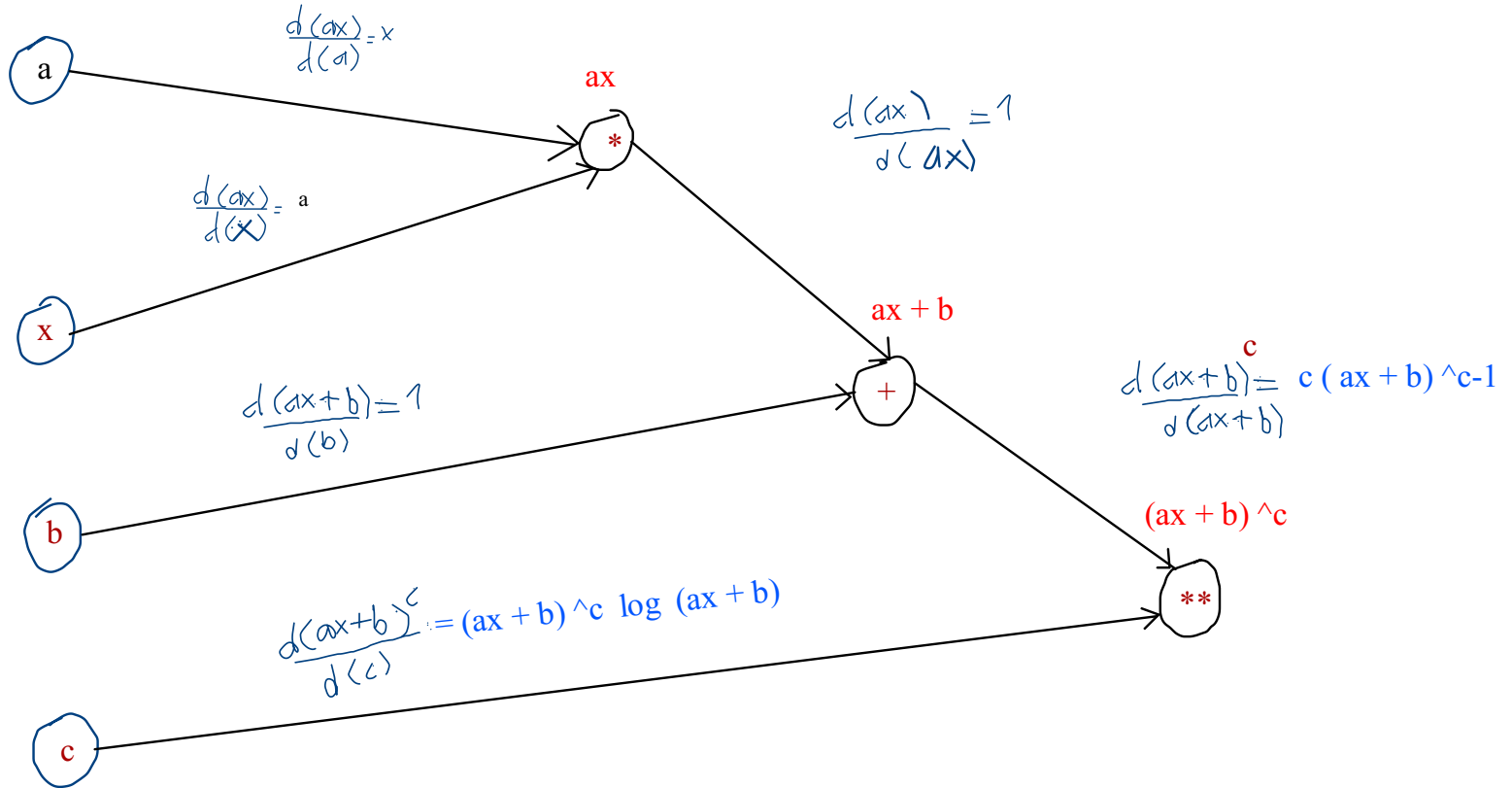


## Auto Diff

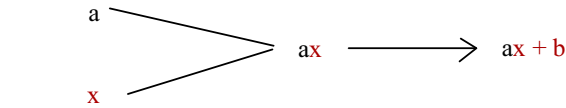
การปรับ weight ด้วย Gradient Descen --> ไข Diff --> Auto Diff

EX -->  $y = (ax + b)^c$



EX -->  $y = (ax + b) ** c$

```
[{'id': 0,
  'data': 1,
  'created_from': 'assigned',
  'send_to': [4],
  'diff': [2],
  'destination': 'wait'},
 {'id': 1,
  'data': 2,
  'created_from': 'assigned',
  'send_to': [4],
  'diff': [1],
  'destination': 'wait'},
 {'id': 2,
  'data': 3,
  'created_from': 'assigned',
  'send_to': [5],
  'diff': [1],
  'destination': 'wait'},
 {'id': 3,
  'data': 4,
  'created_from': 'assigned',
  'send_to': [6],
  'diff': [1005.8986952713127],
  'destination': 'wait'},
 {'id': 4,
  'data': 2,
  'created_from': ('*', 0, 1),
  'send_to': [5],
  'diff': [1],
  'destination': 'wait'},
 {'id': 5,
  'data': 5,
  'created_from': ('+', 4, 2),
  'send_to': [6],
  'diff': [500],
  'destination': 'wait'},
 {'id': 6,
  'data': 625,
  'created_from': ('**', 5, 3),
  'send_to': [],
  'diff': [],
  'destination': 'wait'}]
```



$$\frac{d(ax)}{d(a)} = x$$

$$\frac{d(ax)}{d(x)} = a$$

$$\frac{d(ax+b)}{d(b)} = 1$$

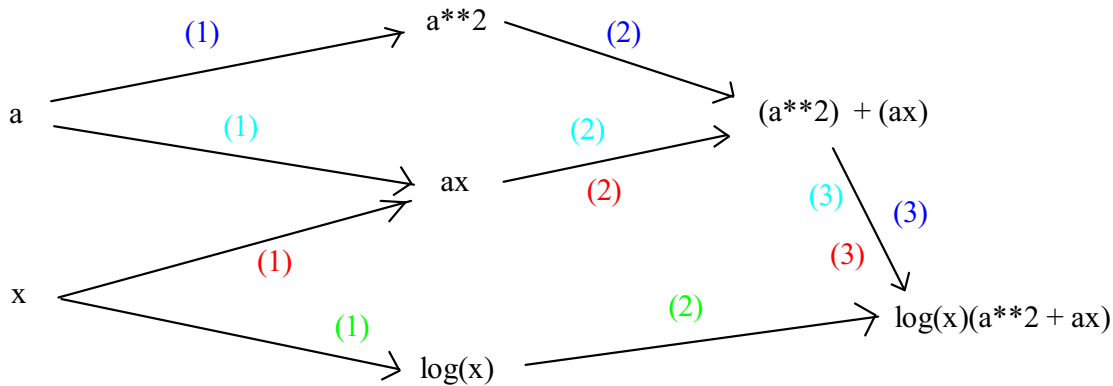
$$\frac{d(ax+b)^c}{d(c)} = (ax+b)^c \log(ax+b)$$

$$\frac{d(ax)}{d(ax)} = 1$$

$$\frac{d(ax+b)^c}{d(ax+b)} = c(ax+b)^{c-1} = 4((1 \times 2) + 3) ** 3$$

a = tensor(a)  
x = tensor(x)  
b = tensor(b)  
c = tensor(c)

a = tensor(1)  
x = tensor(2)  
b = tensor(3)  
c = tensor(4)



คือ การหา Chain Rule ที่มากกว่า 1  
สาย เพื่อไล่ลำดับ การคำนวณ

$\leq (\pi)$  ผลบวกรวม ของผลคูณ

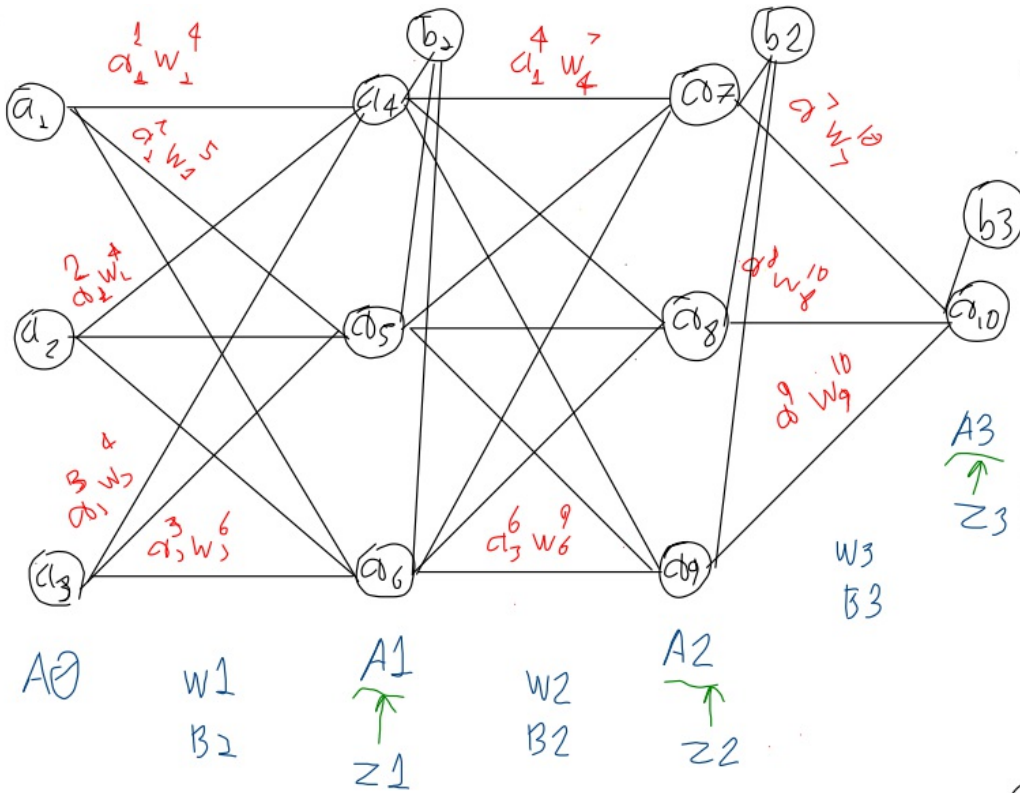
ถ้าต้องการ

$$\frac{d}{da} \text{Log}[(x)((a^{**2}) + ax)] = (1 \times 2 \times x \times 3) + (1 \times x \times 2 \times 3)$$

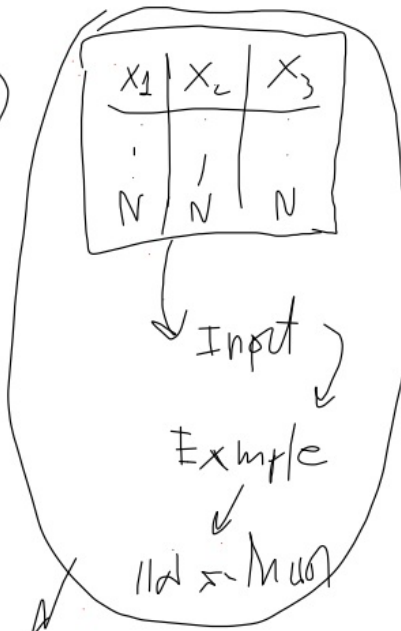
$$\frac{d}{dx} \text{Log}[(x)((a^{**2}) + ax)] = (1 \times x \times 2 \times 3) + (1 \times x \times 2)$$

เวลาทำ Blackward (Y) ปลายทาง  
หาเวลาทางคำนวณที่ได้เฉพาะ ค่า Y เท่านั้น

ทำให้เป็นการกำจัด Scope เฉพาะสาย และ  
ที่อยากจะ Diff เทียบเท่านั้น



$a$  }  $d$  = Feature  $\leftarrow \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}$   
 $a_p$  }  $P$  = Example



$a_1, a_2, a_3$

$$A0 \rightarrow A0W1 + B1 \rightarrow Z1 \rightarrow A1 \rightarrow A1W2 + B2 \rightarrow Z2 \rightarrow A2 \rightarrow A2W3 + B3 \rightarrow Z3 \rightarrow \underline{\underline{A3}}$$

$$Z1 = A0W1 + B1$$

$$Z2 = X1W2 + B2$$

$$Z3 = X2W3 + B3$$

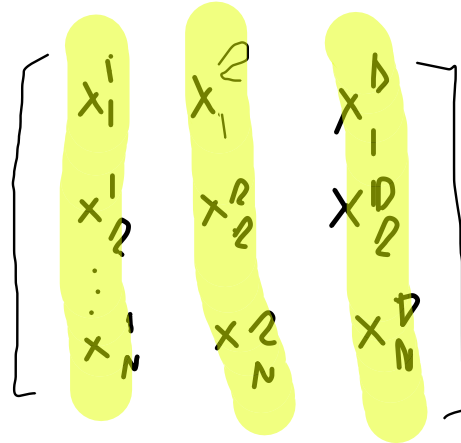
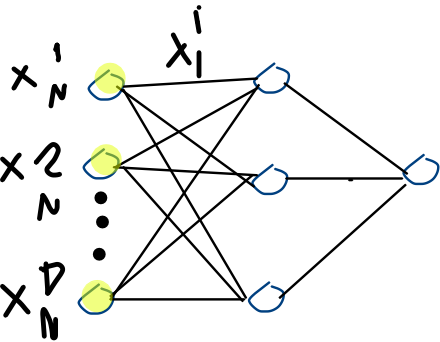
$$X1 = af(Z1)$$

$$X2 = af(Z2)$$

$$X3 = af(Z3)$$

## Auto diff (Numpy)

ช่วยให้ทำหลายๆ สมการพร้อมกันได้



ตัวแปรหลายๆ กันเป็น 1 สมการ และ matrix รวมๆ กันเป็น N สมการ

เราจะไม่ใช่ข้อมูลใน Matrix ที่ละตัว ทั้งหมด เพราะ ข้อมูล Sample มากเกินไป

เลือกเฉพาะ Sample ที่ 1 ก็จะรู้ตัวที่เหลือ ถูกทำอะไรบ้าง (ใช้ architecture ร่วมกัน)

$$\begin{bmatrix} x_1^1 & x_1^2 & x_1^D \\ x_2^1 & x_2^2 & x_2^D \\ \vdots & \vdots & \vdots \\ x_N^1 & x_N^2 & x_N^D \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

1) Sample --> แถวแรกเป็น tensor

2) Weight และ Bias ใช้ทุกตัว(เพราะค่าต่างกัน และเล็กกว่าค่า Sample) --> เป็น Matrix แบบ tensor

