Thuật toán để chuyển các pixel về giá trị [-1,1]

+ Ban đầu mỗi pixel mang giá trị [0:255]

Với a = train_images/255

Pixel	Sau khi chia cho 255
0	0
127	0.498 xấp xỉ 0.5
255	1

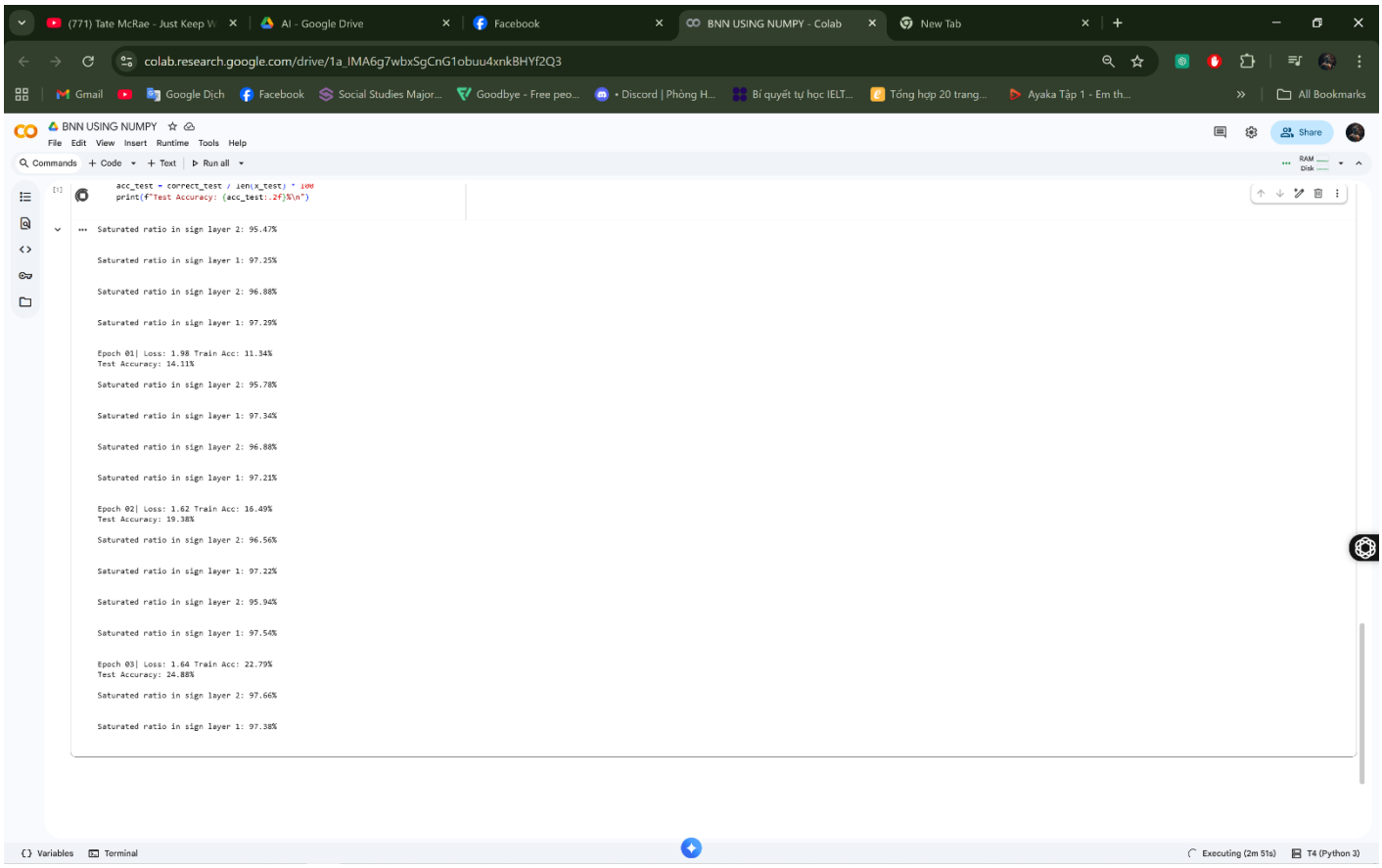
Với b = a – 0.5

a	b
0	-0.5
0.5	0
1	0.5

Với c = b / 0.5

b	b
-0.5	-1
0	0
0.5	1

Kết quả khi để ngưỡng STE_backward là 1, thì các neuron đều đã bị bão hòa. Vì thế gradient không thể lan truyền ngược cho các layer phía sau. Kết quả thu được là mạng không thể học được.



```
Saturated ratio in sign layer 2: 97.97%

Saturated ratio in sign layer 1: 97.36%

Saturated ratio in sign layer 2: 96.25%

Saturated ratio in sign layer 1: 97.30%

Epoch 16| Loss: 1.46 Train Acc: 31.98%
Test Accuracy: 31.29%

Saturated ratio in sign layer 2: 98.28%

Saturated ratio in sign layer 1: 97.48%

Saturated ratio in sign layer 2: 98.75%

Saturated ratio in sign layer 1: 97.63%

Epoch 17| Loss: 1.52 Train Acc: 31.61%
Test Accuracy: 31.46%

Saturated ratio in sign layer 2: 97.50%

Saturated ratio in sign layer 1: 97.42%
```

Độ accuracy thu được cũng rất thấp, vì hầu hết các neuron ở lớp sign 2 đã bão hòa => Gradient không thể lan truyền cho các layer phía sau, dẫn đến các neuron phía sau đã chết vì không được cập nhật.

```
Saturated ratio in sign layer 2: 95.78%
Input: [-22.   4.  20.   2.  -6.   8. -36. -24. -32. -20.  -4.  18.   6.  12.
 -24. -18. -50.  14.  -6.  14.  22.  -4.   0.  18.   2.   4.   8.  -4.
  16.  24. -30.   0.  24.  26. -14.   0. -32.   8.  16.  20.  12.  18.
 -14.  12.   0.  -6. -30. -42.  22. -26. -16.  10.  26.   8.  -8. -14.
   2.  -6. -14.  30.  20. -34.  42.   8.  28.  -2. -30. -34. -30.  30.
 -12. -14.  50.  36.   4. -22. -38. -14. -22.  34.  40.  -6.  18.  -4.
  20.  -2. -34. -18. -18.  22.  24.  -6.  -2.  -4. -20.  -2. -22.   2.
   2.   2.  16.  -6.  10. -32.  28.  26.  10. -22.  34. -26.   8.   6.
  22.   8.  -4.   6. -14.  10.  26. -30.   4.   2.  14.  16.  12.  -6.
 -22. -10.   2. -10.  22.   8. -12. -14.  22.  -4.  -8.  -8.  36.  -8.
   4.  26.  10.   0.  24.   6.  -2.   6.   2. -34. -10.  28.  28.  -6.
 -18.   4. -16.  -4.  28. -12.   6. -28. -16.  14.  14.   4. -32. -76.
 -16.  12.  22. -20.  16.  -2.  10. -16. -48. -20.   4.  -8.  -4. -10.
  18.   8. -20. -10. -14.  -6.  10.  18.  26.  12.   0.   2.   2.   8.
 -20. -20.  12.  12.  16. -18. -14. -28.  -8.   2.   6.  -2.  22. -10.
  12.  26.  22.  -8.  -4.   6. -30.  14.   6.  -2.  -6.  12.   0.  18.
   6. -60. -56.  16.  -8. -16. -10. -12.  28.  26. -14. -28. -44. -44.
   4.   8.  20.   2.   2. -16. -12. -10.   2. -30.  18.  14.  14.  24.
  -4.  10.  -2. -20. -36. -32. -24.  -8.  18. -12.  32.  42. -14. -40.
 -24. -28.  -4.   8.  32.  -6.   2. -28.  16.  14. -18. -26.   6. -10.
   4.   2. -10.  12. -40. -18. -22. -26. -38.  22.  16. -10.  26.  -8.
 -12.  18.  -2. -14.  -2. -18.  24.  14.  26. -12.  20. -10. -10.  -2.
   6.  18.  50.  16.  16. -22.   2. -16.   8.  -4.   0.  -4.  14. -20.
   0.  18.   2.   8. -52. -40.   0.   8.  10. -20. -20.   6.   2.  -4.
 -28.  -8.   4.  24.  22.  28.  -4. -38.  -2.   0.   4. -32.  16.  16.
   8.  18.  -6.  16.   4. -18. -10.   6. -18.  -6.  36. -10.  42.  16.
  12. -22.  -6.  -6.  -2.  30.  16.  26.  14.  16.   0.  -2. -18.   2.
   6.  14.  24.  18.   6.  12. -16. -38.  -2. -54. -14.   6.  -4.  -6.
  18. -24. -28. -14. -30. -18.   6.   2.   0.   2.  14.  48.  16. -18.
 -26.  22.  -6.  18. -18. -24.  28. -22.  10.   0. -12. -36.   0.  16.
  12.  14. -26.  12. -12.  22. -30.   2.   2. -14.  18.   0.   4. -18.
  22.   0. -24. -44.   8. -20.  18.   4.   4.   2. -22. -12.  -4. -32.
  32. -20.  12.  -6.  -6.   4. -28.   6. -10. -14.  38. -18.  22.  16.
  48.  14.  -2. -28. -36. -28. -24.   4.  20. -14.  42.  32.  20.  46.
 -38.   6. -14.  30. -10.  -8.  24.  10. -10.  -8. -16. -28. -12.  16.
  22.   0.  -4. -18. -18.  -4. -12.   4. -16.   4.  -2.  20.  12.   2.
  30.   0. -36. -28. -12.  12.  22.  12.   0.  14.  18. -16. -16. -32.
   0.  -4.  16.   6.  22. -20.   4.  -2. -14. -18.   2.  22.  14.  12.
  36.   6.  14. -12. -16.  -8.  -4.   0.  14. -16.  32.   6.  22. -16.
 -36.  -8.  -4.   8.  -6.  20.  20.   6.  -2. -12. -24.  -4.   0.  36.
  24.  18.  26.  12.  20.  -6.  -6. -22.   6.  -2.  -4.  -6.  22.   0.
   8. -18. -14.  -2. -18.  22.  30.  -4.  28. -18. -22. -24. -20. -32.
   8. -20.  14.  28.  -4.  -6. -10.  -4. -40.  -8. -40.  12. -16.  30.
  26.  24. -16.  -2. -30. -34. -22.  26. -14.  44.   0.   6. -30. -16.
 -24. -20.  20. -16.  10.  16.  32.  10. -14.   0. -36.   0.   8.   4.
 -24.   2.  22.  16.   8.   6. -14. -14. -22.  10.]
```

Thay đổi khoảng cắt gradient là 25

```
        grad_input = grad_output * (np.abs(self.x) <= 25)  # Ở ĐÂY, TA CHỌN NGƯỠNG (THRESHOLD LÀ 25, ĐƯỢC XEM NHƯ LÀ "VÙNG CHO PHÉP GRADIENT ĐI QUA")
        saturated = np.sum(np.abs(self.x) > 25)
```

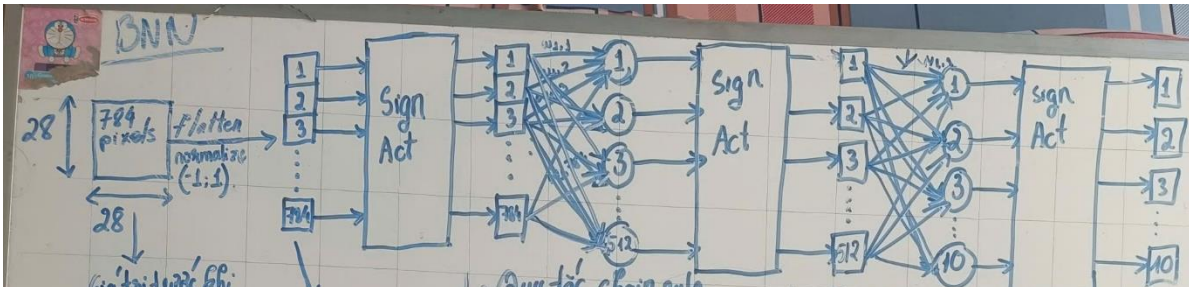
```
Saturated ratio in sign layer 2: 22.34%
Input: [ 12.  -2.  32. -16. -30.  30.  -6. -12. -16. -36. -18.  12. -22.   6.
 -40.  32. -12. -22.  22. -18.  14.  28.  22. -34. -32. -12.  12.  26.
   6.  22. -10.  12.   2. -18. -12.  16. -16.  -6. -22.   6.  28.  10.
   8. -20. -22.  26.  22.  28.   8. -16.   4.  -6.   8. -16. -10.  30.
 -10. -20. -24. -20.  -4.  38. -12. -12. -26.  14.  30.   0.  -8.  20.
 -22.  28.  -2.  10. -24.  28. -12. -30. -18.  -2.   4.   2.   4. -12.
 -22.  26. -26.   4. -28.   0. -18.  12.  30. -22. -36.   4.   0.  10.
  -6. -26.  18.  12.  -6.  -6. -12.   8.  -4. -42. -26. -18.  10.   4.
   2.   2. -36.  16. -20. -10.  -2.  14.  -6.  -4. -10.   6.  -4.   8.
 -32.  -2.  10. -14.  18.  28.  26. -26. -32. -12.   0.   6. -42. -14.
 -14.  36.   2.   2.  -8.   8.   4.  -6. -22. -10.  -2.  40.  -2. -14.
 -32.  -8. -12.   2. -26. -22.  12.  18.   4. -24. -38.  14.  18. -20.
 -16.  -8.  10.   0.  14. -34. -20.  12.  12. -26.  -6. -18.  28.   2.
  44. -20. -34.  22.  10.   0.  16.  -8.  28.  30.   0.   0.  -2.  34.
 -18.  32.   0.  24.  -2.   4.   2. -46. -36.  16.   8.  -2.  -6.  10.
   4.  14.   4. -28. -14.  50.   2.  16.   0. -28.   0.   6. -12. -16.
 -38.  18.   6. -16. -24. -20.  12.  30.  -8. -20. -30.  18.   6. -24.
 -20.  -8. -10.  12.  26.   6.  -4.  40.   4.   2.   6. -30.  24.  22.
  24.   0. -22.   6.  14.  28.   8. -12.   0.  30.  28.   0. -26.  10.
  10.  32.  -8. -36.   0.  30.  -4.   0. -14. -14.  -6.  -4. -20.  16.
  36.  22.   0.  -8. -18.  38. -30. -16.   0.   0.  14.  32.  22. -30.
 -20.  20.  12.   2.   6.   2.   0.   6. -16. -12. -22.  38. -14.  20.
   4. -16.  14.  36.  10. -18. -44.   4.  -8.  22.   2.  -2.  22.  16.
  10. -22. -32.  -8.  12.  26.   6.  -2.   4.  26. -16. -24. -18.  26.
 -30. -16.  24. -16. -42.  16. -14.  14. -36.  36. -16. -18. -38.  10.
   0.  14.  24. -24. -22.  -2. -14.   4.  16.  -4.  42.  16.  10.  -6.
 -36.   0.  16.  -2.  18.  -6.  -4.   2.   8. -24. -22.   2.   2.  24.
 -24.  -4.   4. -30.  -8.  -8. -22.  22.  -6. -12.  28. -44.   0.  -6.
  -4. -20. -10.  30. -14.   0. -28.  36.  10.  12.  -6.  -6. -32.  16.
  20.  10. -14. -18.   6.  16. -30. -22. -16.  48.   4. -14.  10. -18.
  30.  16.  14. -26.   0.  28. -16.   6. -14. -18.   4.  18.   0. -20.
 -22.  10.  14.   0. -28.  36.  16.  22.  28. -16. -30.  -2. -18. -12.
   4. -16.  10.  24.  26.  10. -32.  -4.  16.   6. -10. -46. -16.  26.
  32. -24. -26.  22. -10.   8.   4. -32.  -8. -14.  20.  -8. -18.  30.
 -26.  24.  20.   8. -10.  20.  26. -18. -12.  -4. -16. -26.  -2.  14.
  12.  10.  -4. -20. -10.  46. -30. -12.  -8.  -4.  28.  22.  12.   4.
 -10.  34.  10.   8.  -8. -16.  10.  36.   2. -18. -20.  12.   4.   2.
   6. -14.  18.  12.  -2. -30. -32.  20.   8.  18.   6. -30.  -4.  14.
  -8. -16. -22.  14. -10.   0. -40.  -8.  -4.  10.   0. -12. -14.  22.
  30.   4. -12. -32.  12.  14.  -4.  -4. -26.  18.  -6. -12. -16.   0.
 -10.  16.  -6.  14. -40.  24.   8.  30.   2.  -2.  -4.  -6.  28. -40.
 -30.  14. -26.   8.   0.  -8.  -2.  12.   2. -22. -24.  20.  20.   6.
  -6.  -2.  -4.   2.  16.   0. -26.  22.  22.  -4.  16. -56.  -8.  26.
   8.   4. -18.   2.  18.   0. -24. -12. -16.   2.  -8.  -4. -46.  14.
 -26.   4.  32. -12.  16.  18.   4. -48. -50.  22.  22.  12.   8. -40.
   8.  34.   8.  -8. -62.  22.  10.   8.   4. -36.]


Saturated ratio in sign layer 1: 38.96%
```

Mức độ bão hòa trong quá trình training đã được giảm xuống, gradient có thể lan truyền cho các layer phía sau.

```
Epoch 04| Loss: 0.19 Train Acc: 84.04%
Test Accuracy: 83.03%
```

Tại epoch 04 mạng đã đạt được yêu cầu

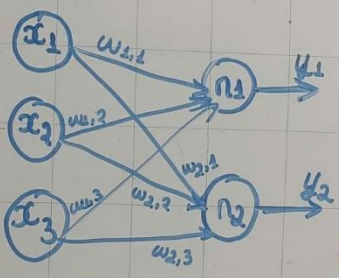
BNN



Giá trị trước khi chuẩn hoá là (0;255) cho mỗi pixel

[-1;1] cho mỗi pixel

• Backpropagation



$$\begin{cases} y_1 = w_{1,1}\cdot x_1 + w_{1,2}\cdot x_2 + w_{1,3}\cdot x_3 + b_1 \ (1) \\ y_2 = w_{2,1}\cdot x_1 + w_{2,2}\cdot x_2 + w_{2,3}\cdot x_3 + b_2 \ (2) \end{cases}$$

3] $L(y; y_L)$ $\frac{\partial L}{\partial y_1}, \frac{\partial L}{y_2}$

Hỏi trọng số $w_{i,j}$ cần thay đổi bao nhiêu để giảm Loss?

• Quy tắc chain rule

$$\frac{\partial L}{\partial w_{i,j}} = \frac{\partial L}{\partial y_j} \cdot \frac{\partial y_j}{\partial w_{i,j}} \ (*)$$

Từ pt Forward, ta có

$$y_j = \sum_i x_i \cdot w_{j,i} + b_j$$

$$\Rightarrow \frac{\partial y_j}{\partial w_{i,j}} = \sum_i \cdot x_i = X_{\text{ma trận}}$$

$$(*) \Rightarrow \frac{\partial L}{\partial w_{i,j}} = \frac{\partial L}{\partial y_j} \cdot X \Rightarrow \frac{\partial L}{\partial w} = \begin{bmatrix} \frac{\partial L}{\partial y_1}\cdot x_1 & \frac{\partial L}{\partial y_1}\cdot x_2 & \frac{\partial L}{\partial y_1}\cdot x_2 \\ \frac{\partial L}{y_2}\cdot x_1 & \frac{\partial L}{\partial y_2}\cdot x_2 & \frac{\partial L}{\partial y_2}\cdot x_2 \end{bmatrix}$$

$$\Rightarrow \frac{\partial L}{\partial W} = \left(\frac{\partial L}{\partial y}\right)^T \cdot x = np.dot(grad\_output.T, self.x)$$

↳ gradient ở lớp phía trước.

• Gradient theo bias

Từ pt forward, ta có:

$$\frac{\partial L}{\partial b_j} = \frac{\partial L}{\partial y_j} \cdot \frac{\partial y_j}{\partial b_j}$$

Với $\frac{\partial y_j}{\partial b_j} = 1 \Rightarrow \frac{\partial L}{\partial b_j} = \frac{\partial L}{\partial y_j} = np.sum(grad\_output, axis=0, keepdims=True)$

• return np.dot(x, self.binary_weight.T) + self.bias.

→ $y = x \cdot W_b^T + bias$

c. dan truyền ngược - back propagation

① Gradient theo bias

• Từ pt forward ta có:

$$y_j = \sum_i w_{j,i} \cdot x_i + b \Rightarrow \frac{\partial y}{\partial b} = 1$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial b} = \frac{\partial L}{\partial y}$$

* Khi xử lý nhiều ảnh, ví dụ batch_size = 64.

• Đại ngõ ra: $\frac{\partial L}{\partial y}$ có shape là (64 x 10), tức mỗi ảnh đưa vào sẽ có 1 vector lỗi [redacted] tương ứng với mỗi neuron ngõ ra.

HONGHA

Mo Tu We Th Fr Sa Su

• Những bias thì chỉ có 1 vector duy nhất cho cả Fc layer (tức chung mọi ảnh).

→ Vậy ta sẽ cộng tất cả lỗi từ các mẫu trong batch lại.

$$\frac{\partial L}{\partial b_j} = \sum_{i=1}^{batch} \frac{\partial L_i}{\partial y_{ij}}$$

Giả sử ta có:

• Batch size = 3

• Output layer = 2 neuron => out_features = 2.

• grad_output = np.array([

→ [0.2, -0.1] # ảnh 1

[0.5, 0.3] # ảnh 2

[-0.4, 0.6] # ảnh 3

])

→ Shape của grad_output = (3x2), 3 mẫu trong đó mỗi mới chứa 2 giá trị lỗi của 2 neuron).

• Như đã chứng minh ở trên bias là chung cho mọi ảnh

→ ta cộng lỗi theo batch.

→ axis = 0 → cộng theo chiều "batch" (3 ảnh)

→ keepdims = True → giữ shape (1x2) = shape của mỗi mẫu

HONG HA

Mo Tu We Th Fr Sa Su

- Câu hỏi: Tại sao lại là cộng mà k° phải lấy 'giá trị trung bình.

Ảnh	grad_output[0]		grad_output[1]
1	+0.2	grad_b =	-0.1
2	+0.5	0.2 + 0.5 - 0.4.	+0.3
3	-0.4		+0.6

So sánh. [1] Cập nhật theo mỗi ảnh.

+ Ảnh 1 (neuron 1): $b_{new_1} = b_{old} - 0.1 . 0.2$ (b1)

+ Ảnh 2: $b_{new_2} = b_{old} - 0.1 . 02 - 0.1 . 0.5$

+ Ảnh 3: $b_{new_3} = b_{old} - 0.1 [0.2 + 0.5 - 0.4]$

Cuối dù được tính theo từng ảnh thì giá trị grad_b ~~vẫn b~~ chính là cộng lại.

② gradient theo input → để lan truyền ngược về cho layer sau.

Gọi pt forward: $y = W^T . x + b$

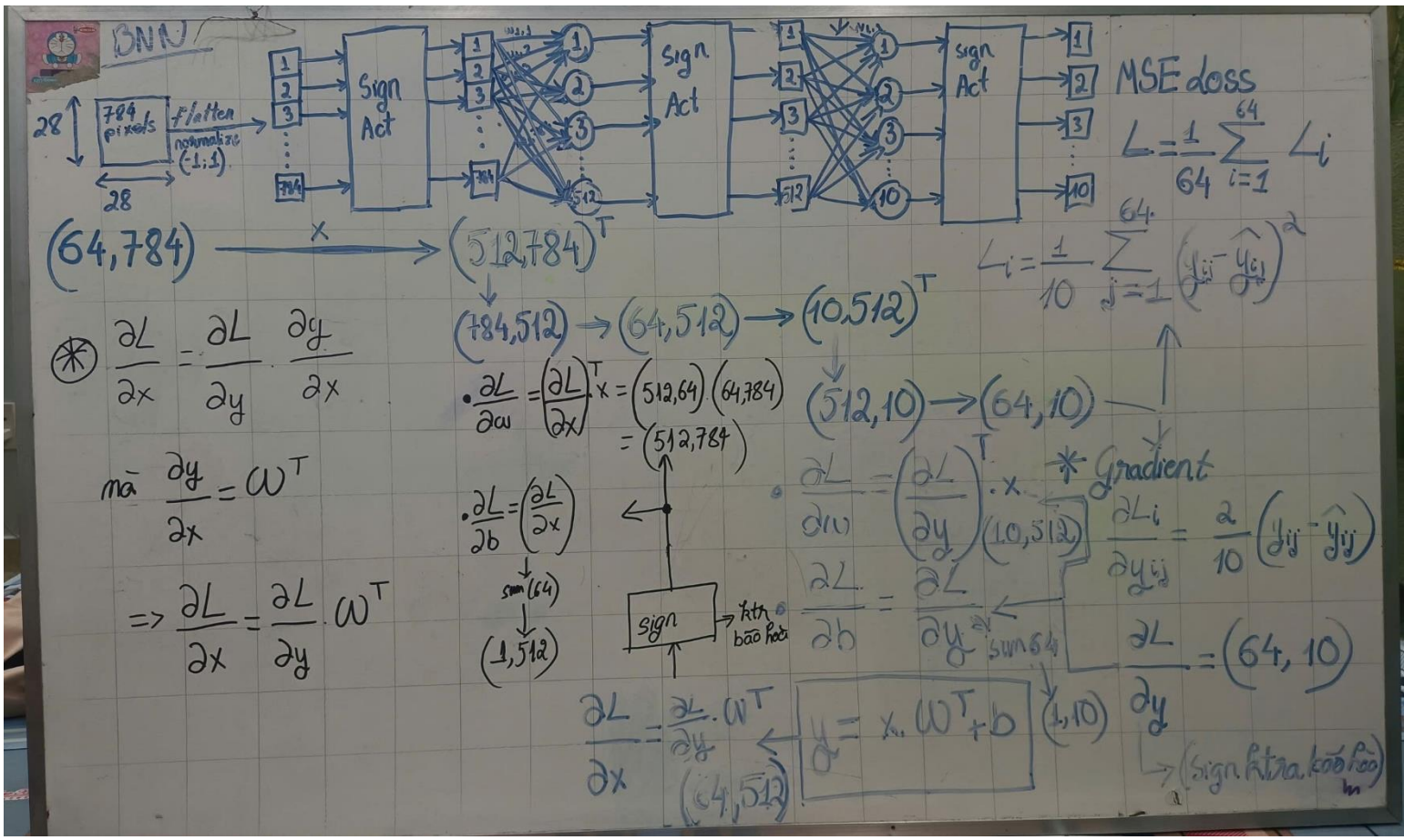
$$\Rightarrow \frac{\partial y}{\partial x} = W^T$$

kq trả về shape = 512 ↑

$$\Rightarrow \frac{\partial L}{\partial x} = \frac{\partial y}{\partial x} . \frac{\partial L}{\partial y} = \frac{\partial L}{\partial y} . W^T = np.dot(grad\_output, W^T)$$

$\frac{\partial L}{\partial y}$ → gradient ở layer

HONGHA


BNN
28
784 pixels
Flatten
normalize (-1;1)
28
Sign Act
Sign Act
sign Act
MSE loss
$L = \frac{1}{64}\sum_{i=1}^{64} L_i$
$L_i = \frac{1}{10}\sum_{j=1}^{64.} (y_{ij} - \hat{y}_{ij})^2$
(64,784) x (512,784)^T
(784,512) → (64,512) → (10,512)^T
(512,10) → (64,10)
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x}$
mà $\frac{\partial y}{\partial x} = w^T$
$\Rightarrow \frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot w^T$
$\frac{\partial L}{\partial w} = \left(\frac{\partial L}{\partial x}\right)^T \cdot x = (512,64)(64,784) = (512,784)$
$\frac{\partial L}{\partial b} = \left(\frac{\partial L}{\partial x}\right)$
sum(64)
(1,512)
sign
Xấp bão hoà
$\frac{\partial L}{\partial w} = \left(\frac{\partial L}{\partial y}\right)^T \cdot x$ (10,512)
$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial y}$ sum 64
(1,10)
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot w^T$
(64,512)
$y = x \cdot w^T + b$
* Gradient
$\frac{\partial L_i}{\partial y_{ij}} = \frac{2}{10}(y_{ij} - \hat{y}_{ij})$
$\frac{\partial L}{\partial y} = (64,10)$
(Sign. Xấp. bão hoà)