

MLP Full Flow (Review)

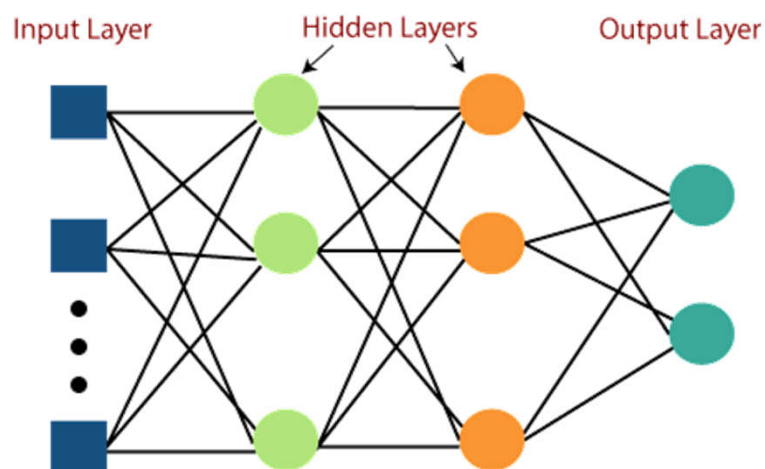
Table of content

- **1. Khái niệm của các tính chất trong MLP**
 - 1.1 Cấu trúc tổng quan
 - 1.2 Layer (lớp)
 - 1.3 Pass (Loại chạy)
 - 1.4 Loss (Giá trị mất mát)
- **2. Giới thiệu bài toán**
 - 2.1 Giới thiệu bài toán
 - 2.2 Giới thiệu mô hình
 - 2.3 Công thức toán học
- **3. Hệ thống hóa lại bài toán theo dạng coding**
 - 3.1 MLP Flow: khởi tạo object
 - 3.2 MLP Flow: forward pass, loss, backward pass
 - 3.3 MLP Flow: kết luận

1. Khái niệm của các tính chất trong MLP

- **1.1 Cấu trúc tổng quan**

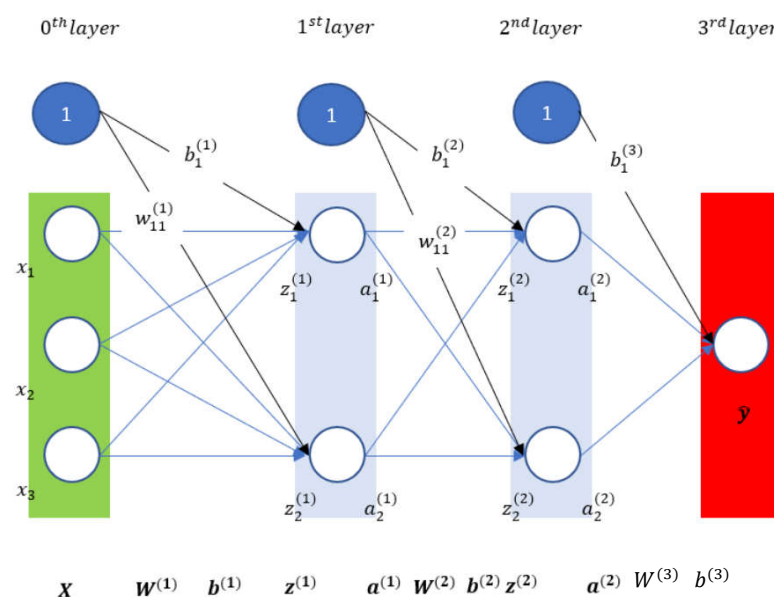
- Mạng MLP (Multi-layer perceptron): Là mạng gồm nhiều layer, và layer thì được phân ra thành 3 dạng chính: Input, Hidden, Output
- Mạng MLP sẽ tính giá trị từ input layer đi từng bước tới output layer để tìm giá trị dự đoán, và đi từ output layer trở về input layer để cập nhật trọng số



1. Khái niệm của các tính chất trong MLP

1.2. Layer:

- Input Layer ($0^{th} layer$): Là lớp đầu vào, dùng để truyền các thuộc tính của dataset vào.
- Hidden Layer ($1^{st}, 2^{nd} layer$): sẽ có trọng số W và b được dùng để tính phương trình tuyến tính với giá trị được trả từ lớp trước đó và kết quả tuyến tính sẽ cho qua hàm Activation (RELU)
- Output Layer ($3^{rd} layer$): Sẽ có trọng số W và b , dùng để tính phương trình tuyến tính với giá trị được trả về từ lớp trước đó và kết quả sẽ là kết quả dự đoán



1. Khái niệm của các tính chất trong MLP

• 1.3. Pass

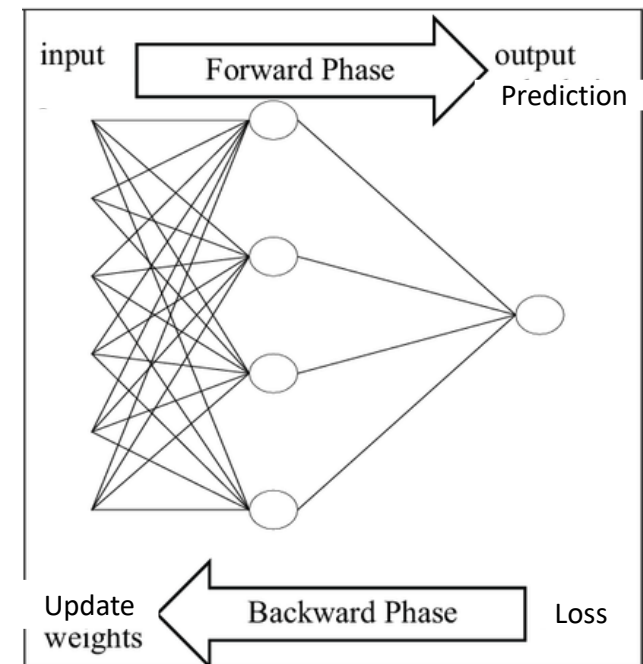
- Trong mạng MLP sẽ có 2 loại chạy: forward pass và backward pass

- 3.1 Forward pass:

- Truyền từ layer input đến layer output.
- **Mục đích là dùng để tìm giá trị dự đoán.**

- 3.2 Backward pass:

- Truyền từ layer output đến layer input
- **Mục đích là để cập nhật trọng số W và b của các layer**



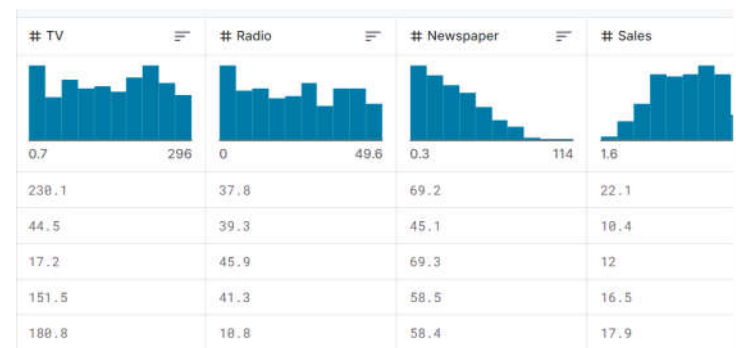
1. Khái niệm của các tính chất trong MLP

- **1.4. Loss**

- Là công thức tính độ sai số của giá trị dự đoán (kết quả của forward pass) với giá trị thực tế
- Giá trị Loss càng lớn, thì giá trị sai số càng nhiều

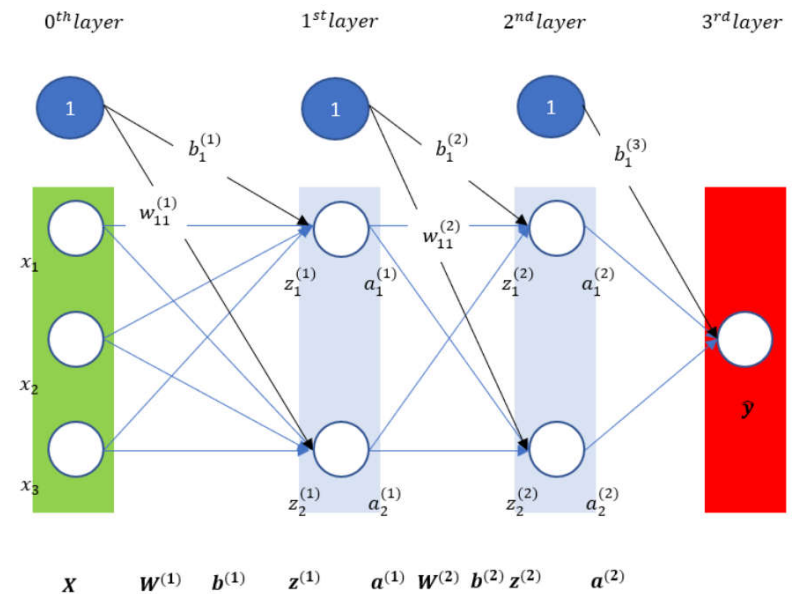
2.1 Giới thiệu bài toán

- Hiện tại ta đang học tới bài toán dự đoán (regression), là bài toán dự đoán một giá trị output từ một hoặc nhiều feature đầu vào.
- Ví dụ:
 - Dự đoán giá nhà (house pricing) : feature là diện tích (m^2) , output là giá nhà
 - **Dự đoán doanh thu từ quảng cáo (advertising):** features là số tiền bỏ vào cho TV, Radio, Newspaper, output là doanh thu
- Ta sẽ dùng ví dụ bài toán doanh thu từ quảng cáo cho những slide tiếp theo.



2.2 Giới thiệu mô hình sử dụng

- Ta dùng tổng cộng là 4 lớp:
 - Input (0^{th} layer): Gồm 3 features là TV, Radio, Newspaper, tương ứng với 3 node trong lớp input
 - Hidden ($1^{st}, 2^{nd}$ layer): đều cùng có 2 node trong từng lớp
 - Output (3^{rd} layer): có 1 output nên sẽ có 1 node ở lớp output
- Cùng đi qua công thức để giải bài toán



2.3 Công thức toán học

- **1. Forward pass:** Bước tìm giá trị doanh thu dự đoán từ 3 features: TV, Radio, Newspaper.

- Ta sẽ có những tham số và hàm sau:

- X : ma trận của $[x_1 \ x_2 \ x_3]$
- $W^{(1)}, b^{(1)}$: ma trận gồm các trọng số giữa 2 lớp 0^{th} và 1^{st} được khởi tạo ngẫu nhiên lúc đầu
- $z^{(1)}$: ma trận gồm các hàm $[z_1^{(1)} \ z_2^{(1)}]$
- $a^{(1)}$: ma trận gồm các hàm $[a_1^{(1)} \ a_2^{(1)}]$ ($f(x)$: hàm *ReLU*)
-
- \hat{y} : giá trị dự đoán

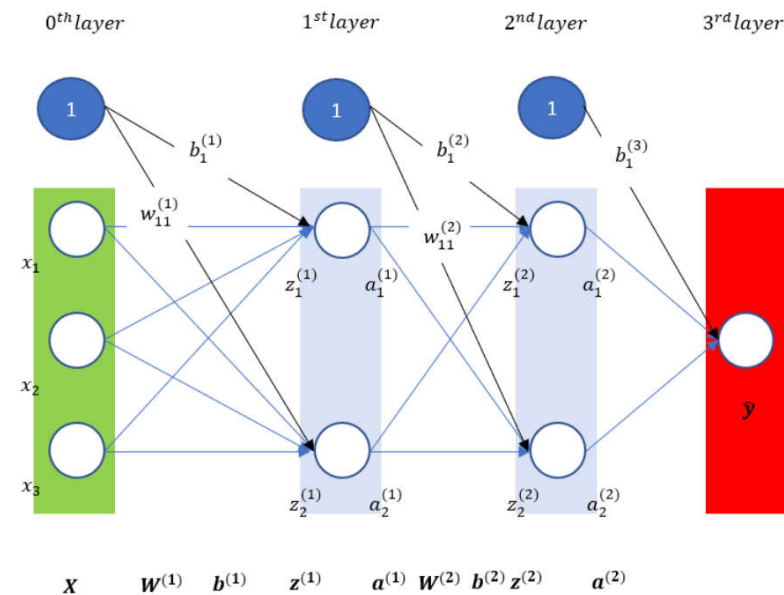
$$z^{(1)} = XW^{(1)} + b^{(1)}$$

$$a^{(1)} = f(z^{(1)})$$

$$z^{(2)} = a^{(1)}W^{(2)} + b^{(2)}$$

$$a^{(2)} = f(z^{(2)})$$

$$\hat{y} = a^{(2)}W^{(3)} + b^{(3)}$$



2.3 Công thức toán học

- **2. Loss:** Tính ra giá trị sai số của giá trị dự đoán so với giá trị thực tế
 - Khi huấn luyện mô hình thì ta cần có thêm giá trị thực tế để so sánh với giá trị dự đoán.
Ta cho giá trị thực tế là: y

$$loss = (\hat{y} - y)^2$$

2.3 Công thức toán học

- **3. Backward pass:** Bước cập nhật các trọng số
 - Ta cần cập nhật lại các trọng số W và b của các layer để phù hợp với bài toán

$$\hat{y} = a^{(2)}W^{(3)} + b^{(3)}$$

$$loss = (\hat{y} - y)^2$$

Công thức forward và loss

$$W^{(3)} = W^{(3)} - loss'(W^{(3)}) * lr$$

$$loss'(W^{(3)}) = \frac{dloss}{dW^{(3)}} = \frac{dloss}{d\hat{y}} * \frac{d\hat{y}}{dW^{(3)}} = 2 * (\hat{y} - y) * a^{(2)}$$

$$b^{(3)} = b^{(3)} - loss'(b^{(3)}) * lr$$

$$loss'(b^{(3)}) = \frac{dloss}{db^{(3)}} = \frac{dloss}{d\hat{y}} * \frac{d\hat{y}}{db^{(3)}} = 2 * (\hat{y} - y) * 1$$

2.3 Công thức toán học

- **3. Backward pass:** Bước cập nhật các trọng số
 - Ta cần cập nhật lại các trọng số W và b của các layer để phù hợp với bài toán

$$\begin{array}{l} z^{(2)} = a^{(1)}W^{(2)} + b^{(2)} \\ a^{(2)} = f(z^{(2)}) \end{array} \quad \hat{y} = a^{(2)}W^{(3)} + b^{(3)} \quad loss = (\hat{y} - y)^2$$

Công thức forward và loss

$$W^{(2)} = W^{(2)} - loss'(W^{(2)}) * lr$$

$$loss'(W^{(2)}) = \frac{dloss}{dW^{(2)}} = \frac{dloss}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} * \frac{dz^{(2)}}{dW^{(2)}} = 2 * (\hat{y} - y) * W^{(3)} * \begin{cases} 0, & z^{(2)} < 0 \\ 1, & z^{(2)} \geq 0 \end{cases} * a^{(1)}$$

$$b^{(2)} = b^{(2)} - loss'(b^{(2)}) * lr$$

$$loss'(b^{(2)}) = \frac{dloss}{db^{(2)}} = \frac{dloss}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} * \frac{dz^{(2)}}{db^{(2)}} = 2 * (\hat{y} - y) * W^{(3)} * \begin{cases} 0, & z^{(2)} < 0 \\ 1, & z^{(2)} \geq 0 \end{cases} * 1$$

2.3 Công thức toán học

- **3 Backward pass:** Bước cập nhật các trọng số
 - Ta cần cập nhật lại các trọng số W và b của các layer để phù hợp với bài toán

$$z^{(1)} = XW^{(1)} + b^{(1)}$$

$$a^{(1)} = f(z^{(1)})$$

$$z^{(2)} = a^{(1)}W^{(2)} + b^{(2)}$$

$$a^{(2)} = f(z^{(2)})$$

$$\hat{y} = a^{(2)}W^{(3)} + b^{(3)}$$

$$loss = (\hat{y} - y)^2$$

$$W^{(1)} = W^{(1)} - loss'(W^{(1)}) * lr$$

$$loss'(W^{(1)}) = \frac{dloss}{dW^{(1)}} = \frac{dloss}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} * \frac{dz^{(2)}}{da^{(1)}} * \frac{da^{(1)}}{dz^{(1)}} * \frac{dz^{(1)}}{dW^{(1)}}$$

$$= 2 * (\hat{y} - y) * W^{(3)} * \begin{cases} 0, & z^{(2)} < 0 \\ 1, & z^{(2)} \geq 0 \end{cases} * W^{(2)} * \begin{cases} 0, & z^{(1)} < 0 \\ 1, & z^{(1)} \geq 0 \end{cases} * X$$

$$b^{(1)} = b^{(1)} - loss'(b^{(1)}) * lr$$

$$loss'(b^{(1)}) = \frac{dloss}{db^{(1)}} = \frac{dloss}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} * \frac{dz^{(2)}}{da^{(1)}} * \frac{da^{(1)}}{dz^{(1)}} * \frac{dz^{(1)}}{db^{(1)}}$$

$$= 2 * (\hat{y} - y) * W^{(3)} * \begin{cases} 0, & z^{(2)} < 0 \\ 1, & z^{(2)} \geq 0 \end{cases} * W^{(2)} * \begin{cases} 0, & z^{(1)} < 0 \\ 1, & z^{(1)} \geq 0 \end{cases} * 1$$

3. Hệ thống hóa lại theo dạng coding

- Với số lượng càng nhiều layer, thì công thức sẽ càng dài
- Việc hard code công thức sẽ tốn nhiều effort và dễ bị nhầm, cho nên ta sẽ đi tìm công thức tổng quát ở tất cả layer.

3. Hệ thống hóa lại theo dạng coding

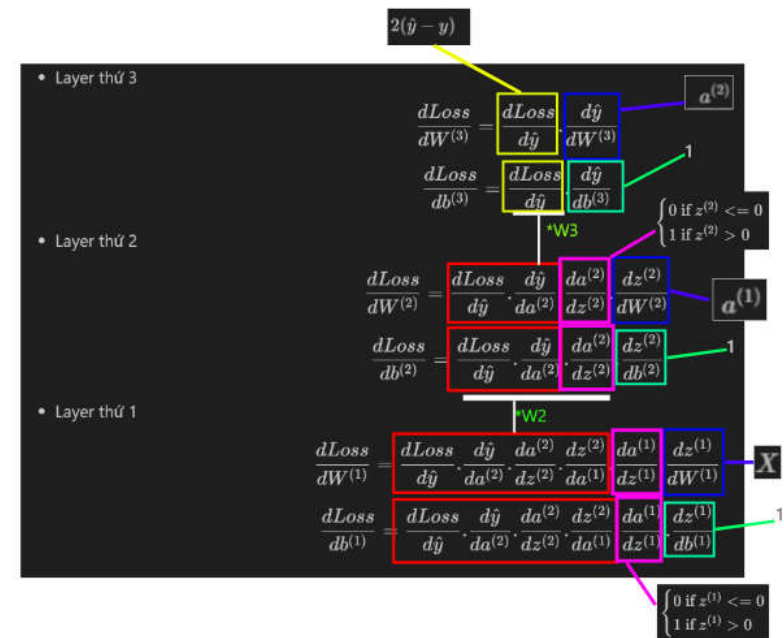
1. Từ layer thứ 2 dùng lại giá trị từ layer thứ 3 và tương tự với layer thứ 1.
2. Các giá trị đạo hàm của $\frac{da^{(l)}}{dz^{(l)}}$ (l là thứ tự layer) cụ thể là $z^{(l)}$ được dùng tại chính layer đó.
3. Các giá trị đạo hàm của $\frac{dz^{(l)}}{dW^{(l)}}$ là giá trị từ lớp trước đó.

Sẽ có một tham số là output của lớp trước đó truyền vào layer để tính backward

Giá trị khi tính forward sẽ được lưu lại để tính backward

$$\hat{y} = a^{(2)} W^{(3)} + b^{(3)}$$

Giá trị khi tính forward sẽ được lưu lại để tính backward



Công thức toán học

3. Hệ thống hóa lại theo dạng coding

* Trong code:

- Ta sẽ tạo ra một class `NetworkLayer`
- Và sau đó sẽ tạo ra 3 objects `NetworkLayer` tương ứng với layer 1st 2nd 3rd.
- Hàm khởi tạo `init` sẽ dựa vào `n_input_node` và `n_output_node` để tạo ra ma trận trọng số ngẫu nhiên

NetworkLayer
+ weights: [float] + bias: [float] + activation: str + input: [float] + linear: [float]
+ init(n_input_node, n_output_node) + forward(prior_forward) + backward(prior_backward)

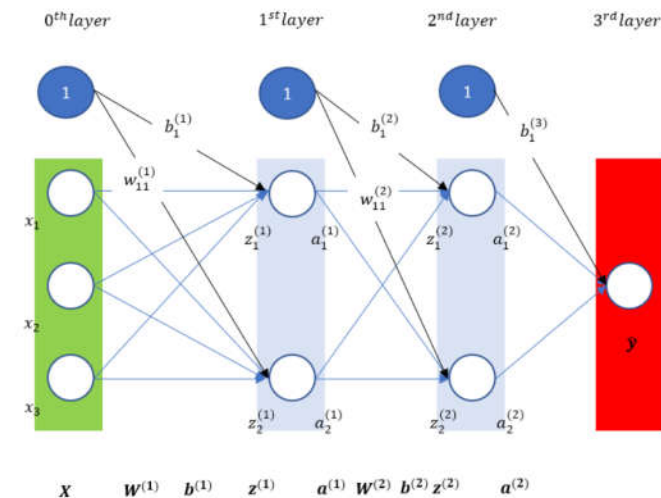
3.1 MLP Flow: Khởi tạo các object

```
NetworkLayer(3,2, activation="RELU")  
NetworkLayer(2,2, activation="RELU")  
NetworkLayer(2,1)
```

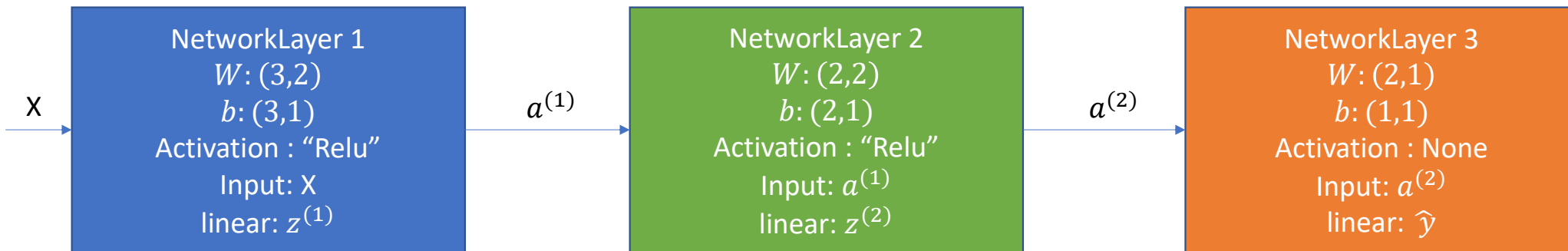
NetworkLayer 1
 $W: (3,2)$
 $b: (3,1)$
Activation : "Relu"

NetworkLayer 2
 $W: (2,2)$
 $b: (2,1)$
Activation : "Relu"

NetworkLayer 3
 $W: (2,1)$
 $b: (1,1)$
Activation : None



3.2 MLP Flow: Forward Pass

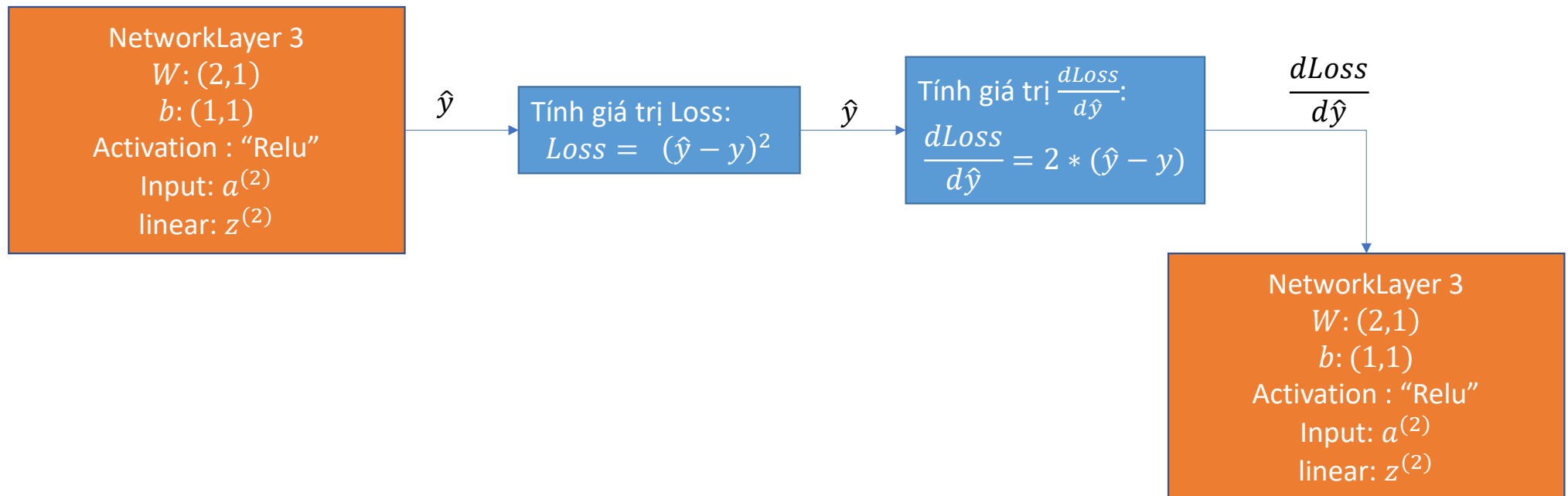


1. Lưu giá trị X vào biến "input" của "NetworkLayer 1"
2. Sẽ tính $z^{(1)} = XW + b$ và lưu $z^{(1)}$ vào biến "linear"
3. Xét điều kiện nếu:
 1. **Activation là "Relu" thì $a^{(1)} = \max(0, z^{(1)})$ và truyền $a^{(1)}$ tới layer tiếp theo**
 2. Activation là None thì truyền $z^{(1)}$ tới layer tiếp theo

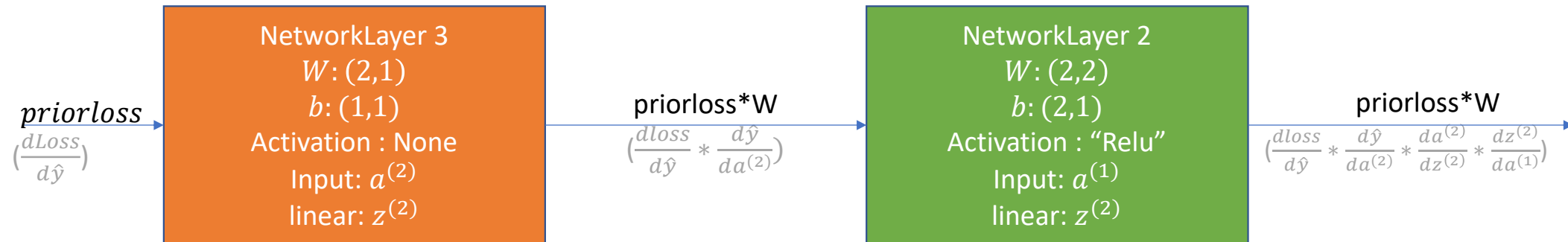
1. Lưu giá trị $a^{(1)}$ vào biến "input" của "NetworkLayer 2"
2. Sẽ tính $z^{(2)} = a^{(1)}W + b$ và lưu $z^{(2)}$ vào biến "linear"
3. Xét điều kiện nếu:
 1. **Activation là "Relu" thì $a^{(2)} = \max(0, z^{(2)})$ và truyền $a^{(2)}$ tới layer tiếp theo**
 2. Activation là None thì truyền $z^{(2)}$ tới layer tiếp theo

1. Lưu giá trị $a^{(2)}$ vào biến "input" của "NetworkLayer 3"
2. Sẽ tính $\hat{y} = a^{(2)}W + b$ và lưu \hat{y} vào biến "linear"
3. Xét điều kiện nếu:
 1. Activation là "Relu" thì $a^{(3)} = \max(0, \hat{y})$ và truyền \hat{y} tới layer tiếp theo
 2. **Activation là None thì truyền \hat{y} tới tính loss**

3.2 MLP Flow: Loss



3.2 MLP Flow: Backward Pass (1)



- Xét điều kiện nếu:
 - Activation là ReLU thì ta tính

$$\text{priorloss} = \text{priorloss} * \begin{cases} 0, & \text{linear} < 0 \\ 1, & \text{linear} \geq 0 \end{cases}$$
- $W -= \text{priorloss} * \text{input} * lr$

$$\left(\frac{d\text{loss}}{d\hat{y}} * \frac{d\hat{y}}{dW^{(3)}} * lr \right)$$

$\hat{y} = a^{(2)}W^{(3)} + b^{(3)}$
- $b -= \text{priorloss} * lr$

$$\left(\frac{d\text{loss}}{d\hat{y}} * lr \right)$$
- Trả về cho layer trước đó: $\text{priorloss} * W$

$$\left(\frac{d\text{loss}}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} \right)$$

- Xét điều kiện nếu:
 - Activation là ReLU thì ta tính

$$\text{priorloss} = \text{priorloss} * \begin{cases} 0, & \text{linear} < 0 \\ 1, & \text{linear} \geq 0 \end{cases}$$
- $W -= \text{priorloss} * \text{input} * lr$

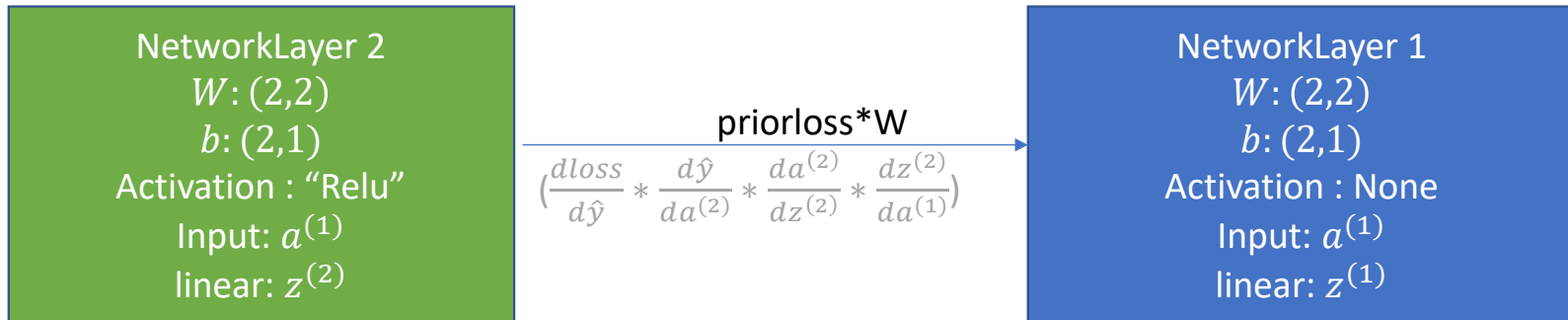
$$\left(\frac{d\text{loss}}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} \right)$$
- $b -= \text{priorloss} * lr$

$$\left(\frac{d\text{loss}}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} * lr \right)$$
- Trả về cho layer trước đó: $\text{priorloss} * W$

$$\left(\frac{d\text{loss}}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} * \frac{dz^{(2)}}{da^{(1)}} \right)$$

$z^{(2)} = a^{(1)}W^{(2)} + b^{(2)}$
 $a^{(2)} = f(z^{(2)})$

3.2 MLP Flow: Backward Pass (2)



1. Xét điều kiện nếu:

1. Activation là ReLU thì ta tính

$$\text{priorloss} = \text{priorloss} * \begin{cases} 0, & \text{linear} < 0 \\ 1, & \text{linear} \geq 0 \end{cases}$$

$$(\frac{d\text{loss}}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} * \frac{dz^{(2)}}{da^{(1)}} * \frac{da^{(1)}}{dz^{(1)}})$$

2. $W \mathrel{-=} \text{input} * \text{priorloss} * lr$

$$(\frac{d\text{loss}}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} * \frac{dz^{(2)}}{da^{(1)}} * \frac{da^{(1)}}{dz^{(1)}} * \frac{dz^{(1)}}{dW^{(1)}} * lr)$$

3. $b \mathrel{-=} \text{priorloss} * lr$

$$(\frac{d\text{loss}}{d\hat{y}} * \frac{d\hat{y}}{da^{(2)}} * \frac{da^{(2)}}{dz^{(2)}} * \frac{dz^{(2)}}{da^{(1)}} * \frac{da^{(1)}}{dz^{(1)}} * lr)$$

$$z^{(1)} = XW^{(1)} + b^{(1)}$$

$$a^{(1)} = f(z^{(1)})$$

3.3 MLP Flow: Kết luận

- Mỗi lớp layer sẽ sở hữu:
 - Trọng số W và b của nó
 - Activation của nó
 - Biến linear và input dùng được tạo từ bước forward, và được dung lại tại bước backward
 - Hàm forward và backward nhận giá trị từ lớp trước và sau nó
- -> Code sẽ logic và mang hướng đối tượng hơn.
- -> Người dùng (người không cần quá hiểu về cấu tạo code bên trong) có thể dễ dàng gọi và tùy chỉnh thông số của các layer.

