

TRƯỜNG ĐẠI HỌC SÀI GÒN KHOA  
CÔNG NGHỆ THÔNG TIN



**BÀI TẬP QUÁ TRÌNH**

**MÔN: SEMINAR CHUYÊN ĐỀ LỚP:**

**DCT1211 (Nhóm 01)**

**NĂM HỌC: 2025- 2026, HỌC KỲ: 2**

**Họ và tên sinh viên**

3121410558 – Nguyễn Hoàng Tuấn

Giảng viên hướng dẫn: PGS. TS Nguyễn Tuấn Đăng

Thành phố Hồ Chí Minh, tháng 3 năm  
2025

## Mục Lục

<b>1. Lan truyền xuôi (Forward Propagation) :</b>	4
1.1. Công thức tính lan truyền xuôi :	4
a. Công thức tính $h_1$ :	4
b. Công thức tính $h_2$ :	4
c. Công thức tính $\hat{Y}$ :	4
d. Hàm sigmoid tính như sau :	4
1.2. Áp dụng vào bài tập :	5
<b>2. Tính hàm mất mát (Loss Calculation ) :</b>	7
2.1. Công thức tính cross-entropy cho từng mẫu :	7
2.2. Để tính mất mát trung bình áp dụng công thức :	8
2.3. kết quả cho áp dụng từng mẫu :	8
<b>3. Lan truyền ngược và cập nhật tham số (Backpropagation &amp; Parameter Update) :</b>	9
3.1. Công thức tính Gradient cho mẫu $i$ :	9
3.1.1. Lớp đầu ra :	9
3.1.2. Lớp ẩn thứ 2 :	9
3.1.3. Lớp ẩn thứ 3 :	10
3.2. Áp dụng vào Epoch 1 và Epoch 2 :	10
3.2.1. Epoch 1:	10
3.2.2. Cập nhật tham số Epoch 1:	11
3.2.3. Epoch 2 : ( dùng tham số mới của Epoch 1 để tính) :	12
3.2.4. Cập nhật tham số Epoch 2:	13
<b>4. Phân tích hiệu suất :</b>	14
<b>5. Đề xuất cải tiến :</b>	16

## Đề Bài :

- ❖ Mô tả bài toán : Thiết kế mạng Multilayer Perceptron (MLP) để dự đoán đầu ra nhị phân dựa trên vector đầu vào  $x = [x_1, x_2]$ . Sinh viên cần trình bày chi tiết

các tính toán thủ công, huấn luyện qua 2 epoch trên nhiều mẫu và phân tích hiệu suất của mô hình.

❖ Cấu trúc MLP :

Đầu vào : 2 (vector =  $[x_1, x_2]$ )

Lớp ẩn 1 : 1 neuron, hàm kích hoạt ReLU  $f(z) = \max(0, z)$

Lớp ẩn 2 : 1 neuron, hàm kích hoạt ReLU  $f(z) = \max(0, z)$

Lớp đầu ra : 1 neuron, hàm kích hoạt sigmoid

❖ Tham số ban đầu :

Lớp ẩn 1 :  $w_{11} = 0.5, w_{12} = -0.5, b_1 = 0.2$

Lớp ẩn 2 :  $w_{21} = 1.0, b_2 = -0.1$

Lớp đầu ra :  $w_{31} = 0.5, b_3 = 0.3$

Tốc độ học :  $\eta = 0.1$

❖ Tập dữ liệu huấn luyện :

Gồm 4 mẫu đại diện cho bài toán phân loại nhị phân đơn giản, có tính phi tuyến :

Mẫu	X1	X2	Y(đầu ra thực tế)
1	1.0	0.0	1
2	0.0	1.0	0
3	1.0	1.0	1
4	0.0	0.0	0

## 1. Lan truyền xuôi (Forward Propagation) :

### 1.1. Công thức tính lan truyền xuôi :

a. Công thức tính  $h_1$ :

$$\bullet \quad h_1 = \text{ReLU}(w_{11}x_1 + w_{12}x_2 + b_1)$$

-Ở đây, ReLU là hàm tính kích hoạt  $\max(0, z)$

Với các giá trị đã cho sau :

$$\text{Lớp ẩn 1: } w_{11} = 0.5, w_{12} = -0.5, b_1 = 0.2$$

b. Công thức tính  $h_2$ :

$$\bullet \quad h_2 = \text{ReLU}(w_{21}h_1 + b_2)$$

Với các giá trị đã cho sau :

$$\text{Lớp ẩn 2: } w_{21} = 1.0, b_2 = -0.1$$

c. Công thức tính  $\hat{Y}$  :

$$\bullet \quad \hat{y} = \sigma(w_{31}h_2 + b_3)$$

d. Hàm sigmoid tính như sau :

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

Với giá trị đã cho như sau :

$$\text{Lớp đầu ra: } w_{31} = 0.5, b_3 = 0.3$$

## 1.2. Áp dụng vào bài tập :

- **Mẫu 1 :**

$$h1 = \max(0, z)$$

$$z = w11*x1 + w12*x2 + b1 = 0.5*1 + (-0.5)*0 + 0.2 = 0.700$$

$$\Rightarrow h1 = \max(0, 0.7) = 0.700$$

$$h2 = \max(0, z)$$

$$z = w21*h1 + b2 = 1*0.700 + (-0.1) = 0.600$$

$$\Rightarrow h2 = \max(0, 0.600) = 0.600$$

$$\hat{Y} = 1/1 + e^{-z}$$

$$Z = w31*h2 + b3 = 0.5*0.6 + 0.3 = 1.4$$

$$\Rightarrow \hat{Y} = 1/1 + e^{-1.4} = 0.646$$

- **Mẫu 2 :**

$$h1 = \max(0, z)$$

$$z = w11*x1 + w12*x2 + b1 = 0.5*0 + (-0.5)*1 + 0.2 = -0.300$$

$$\Rightarrow h1 = \max(0, -0.300) = 0.000$$

$$h2 = \max(0, z)$$

$$z = w21*h1 + b2 = 1*0.000 + (-0.1) = -0.100$$

$$\Rightarrow h2 = \max(0, -0.100) = 0.000$$

$$\hat{Y} = 1/1 + e^{-z}$$

$$Z = w_{31} * h_2 + b_3 = 0.5 * 0.000 + 0.3 = 0.3$$

$$\Rightarrow \hat{Y} = 1 / (1 + e^{-0.3}) = 0.574$$

- **Mẫu 3:**

$$h_1 = \max(0, z)$$

$$z = w_{11} * x_1 + w_{12} * x_2 + b_1 = 0.5 * 0 + (-0.5) * 1 + 0.2 = 0.200$$

$$\Rightarrow h_1 = \max(0, 0.200) = 0.200$$

$$h_2 = \max(0, z)$$

$$z = w_{21} * h_1 + b_2 = 1 * 0.200 + (-0.1) = 0.100$$

$$\Rightarrow h_2 = \max(0, 0.100) = 0.100$$

$$\hat{Y} = 1 / (1 + e^{-z})$$

$$Z = w_{31} * h_2 + b_3 = 0.5 * 0.1 + 0.3 = 0.35$$

$$\Rightarrow \hat{Y} = 1 / (1 + e^{-0.35}) = 0.587$$

- **Mẫu 4:**

$$h_1 = \max(0, z)$$

$$z = w_{11} * x_1 + w_{12} * x_2 + b_1 = 0.5 * 0 + (-0.5) * 0 + 0.2 = 0.200$$

$$\Rightarrow h_1 = \max(0, 0.200) = 0.200$$

$$h_2 = \max(0, z)$$

$$z = w_{21} * h_1 + b_2 = 1 * 0.200 + (-0.1) = 0.100$$

$$\Rightarrow h_2 = \max(0, 0.100) = 0.100$$

$$\hat{Y} = 1 / (1 + e^{-Z})$$

$$Z = w_{31} * h_2 + b_3 = 0.5 * 0.1 + 0.3 = 0.35$$

$$\Rightarrow \hat{Y} = 1 / (1 + e^{-0.35}) = 0.587$$

➤ Kết quả của 4 mẫu với tham số ban đầu (lấy 3 chữ số thập phân) :

Mẫu	X1	X2	Y	H1	H2	$\hat{Y}$
1	1.0	0.0	1	0.700	0.600	0.646
2	0.0	1.0	0	0.000	0.000	0.574
3	1.0	1.0	1	0.200	0.100	0.587
4	0.0	0.0	0	0.200	0.100	0.587

## 2. Tính hàm mất mát (Loss Calculation) :

2.1. Công thức tính cross-entropy cho từng mẫu :

$$L_i = - [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

2.2. Để tính mất mát trung bình áp dụng công thức :

$$: L = \frac{1}{4} \sum_{i=1}^4 L_i$$

2.3. kết quả cho áp dụng từng mẫu :

**Mẫu 1:  $y_1=1$   $\hat{y}=0.646$**

$$L_1 = -[1 \cdot \log(0.646) + (1-1) \cdot \log(1-0.646)] = -\log(0.646) = 0.189$$

**Mẫu 2:  $y_2=0$   $\hat{y}=0.574$**

$$L_2 = -[0 \cdot \log(0.574) + (1-0) \cdot \log(1-0.574)] = -\log(0.4256) = 0.370$$

**Mẫu 3:  $y_3=1$   $\hat{y}=0.587$**

$$L_3 = -[1 \cdot \log(0.587) + (1-1) \cdot \log(1-0.587)] = -\log(0.587) = 0.231$$

**Mẫu 4:  $y_4=0$   $\hat{y}=0.587$**

$$L_4 = -[0 \cdot \log(0.587) + (1-0) \cdot \log(1-0.587)] = -\log(0.4134) = 0.384$$

$$L = \frac{1}{4} \sum_{i=1}^4 L_i = \frac{0.189+0.370+0.231+0.384}{4} = 0.294$$

Mẫu	y(đầu ra thực tế)	$\hat{y}$ (đầu ra dự đoán)	Hàm mất mát (L)
-----	-------------------	----------------------------	-----------------



1	1	0.646	0.190
2	0	0.574	0.371
3	1	0.587	0.231
4	0	0.587	0.384
AVG_L ( Mất mát trung bình)			0.294

### 3. Lan truyền ngược và cập nhật tham số (Backpropagation & Parameter Update) :

3.1. Công thức tính Gradient cho mẫu i : 3.1.1. Lớp đầu ra :

$$\frac{\partial L_i}{\partial w_{31}} = (\hat{y}_i - y_i) \cdot h_{2,i}$$

$$\frac{\partial L_i}{\partial b_3} = (\hat{y}_i - y_i)$$

3.1.2. Lớp ẩn thứ 2 :

$$\frac{\partial L_i}{\partial w_{21}} = (\hat{y}_i - y_i) \cdot w_{31} \cdot I(h_{2,i} > 0) \cdot h_{1,i}$$

$$\frac{\partial L_i}{\partial b_2} = (\hat{y}_i - y_i) \cdot w_{31} \cdot I(h_{2,i} > 0)$$

### 3.1.3. Lớp ẩn thứ 3 :

$$\frac{\partial L_i}{\partial w_{11}} = (\hat{y}_i - y_i) \cdot w_{31} \cdot I(h_{2,i} > 0) \cdot w_{21} \cdot I(h_{1,i} > 0) \cdot x_{1,i}$$

$$\frac{\partial L_i}{\partial w_{12}} = (\hat{y}_i - y_i) \cdot w_{31} \cdot I(h_{2,i} > 0) \cdot w_{21} \cdot I(h_{1,i} > 0) \cdot x_{2,i}$$

$$\frac{\partial L_i}{\partial b_1} = (\hat{y}_i - y_i) \cdot w_{31} \cdot I(h_{2,i} > 0) \cdot w_{21} \cdot I(h_{1,i} > 0)$$

### 3.2. Áp dụng vào Epoch 1 và Epoch 2 :

#### 3.2.1. Epoch 1:

Ví dụ : Mẫu 1 thực hiện tính gradient

$$\text{Gradient } b_3 = -0.354$$

$$\text{Gradient } w_{31} = b_3 \cdot h_2 = -0.354 \cdot 0.6 = -0.212$$

$$\text{Gradient } b_2 = b_3 \cdot w_{31} \cdot I(h_2 > 0) = -0.177$$

$$\text{Gradient } w_{21} = b_2 \cdot h_1 = -0.177 \cdot 0.7 = -0.124$$

$$\text{Gradient } b_1 = b_2 \cdot w_{21} \cdot I(h_1 > 0) = -0.177$$

$$\text{Gradient } w_{11} = b_1 \cdot x_1 = -0.177 \cdot 1 = -0.177$$

$$\text{Gradient } w_{12} = b_1 \cdot x_2 = -0.177 \cdot 0 = 0.000$$

Mẫu 2,3,4 tương tự

➤ Đây là bảng kết quả :

Mẫu	X1	X2	y	H1	H2	Y^	Y^ -Y	Gradien t w31	Gradien t b3	Gradien t w21	Gradien t b2	Gradien t w11	Gradien t w12	Gradien t b1
1	1	0	1	0.7	0.6	0.65	-0.35	-0.212	-0.354	-0.124	-0.177	-0.177	0	-0.177
2	0	1	0	0	0	0.55	0.574	0	0.574	0	0.287	0	0.287	0.287
3	1	1	1	0.2	0.1	0.59	-0.41	-0.041	-0.413	-0.041	-0.207	-0.207	-0.414	-0.207
4	0	1	0	0.2	0.1	0.59	0.587	0.059	0.587	0.059	0.294	0	0	0.294

### 3.2.2. Cập nhật tham số Epoch 1:

➤ Với công thức sau :

$$w_{ij} = w_{ij} - \eta \cdot \frac{\partial L_i}{\partial w_{ij}}, b_i = b_i - \eta \cdot \frac{\partial L_i}{\partial b_i}.$$

- Áp dụng mẫu 1:

$$W31 = w31 - n * \text{gradient } 31 = 0.5 - 0.1 * (-0.212) = 0.5212$$

$$B3 = b3 - n * \text{gradient } b3 = 0.3 - 0.1 * (-0.354) = 0.335$$

$$W21 = w21 - n * \text{gradient } 21 = 1.000 - 0.1 * (-0.124) = 1.0124$$

$$B2 = b2 - n * \text{gradient } b2 = -0.1 - 0.1 * (-0.177) = -0.082$$

$$W11 = w11 - n * \text{gradient } 11 = 0.5 - 0.1 * (-0.177) = 0.518$$

$$W12 = w12 - n * \text{gradient } 12 = -0.5 - 0.1 * 0 = -0.500$$

$$B1 = b1 - n * \text{gradient } b1 = 0.2 - 0.1 * (-0.177) = 0.218$$

Còn lại mẫu 2,3,4 tương tự

Ta có bảng tham số mới :

Mẫu	W31	B3	W21	B2	W11	W12	B1
1	0.521	0.335	1.012	-0.082	0.518	-0.500	0.218
2	0.500	0.243	1.000	-0.129	0.500	-0.529	0.171
3	0.504	0.341	1.004	-0.079	0.521	-0.479	0.221
4	0.494	0.241	0.994	-0.129	0.500	-0.500	0.171

### 3.2.3. Epoch 2 : ( dùng tham số mới của Epoch 1 để tính)

Áp dụng vào mẫu 1 :

$$\text{Gradient } b3 = 0.335 - 0.1 * (-0.336) = 0.369$$

$$\text{Gradient } w31 = b3 * h2 = -0.369 * 0.66 = -0.223$$

$$\text{Gradient } b2 = b3 * w31 * I(h2 > 0) = -0.336 * 0.521 * 1 = -0.175$$

$$\text{Gradient } w21 = b2 * h1 = -0.177 * 0.7 = -0.175 * 0.74 = -0.129$$

$$\text{Gradient } b1 = b2 * w21 * I(h1 > 0) = -0.175 * 1.012 * 1 = -0.177$$

$$\text{Gradient } w11 = b1 * x1 = -0.177 * 1 = -0.177 * 1 = -0.177$$

$$\text{Gradient } w12 = b1 * x2 = -0.177 * 0 = 0.000$$

Tương tự cho mẫu 2,3,4

➤ Bảng kết quả :

Mẫu	X1	X2	y	H1	H2	Y^	Y^ -Y	Gradien t w31	Gradien t b3	Gradien t w21	Gradien t b2	Gradien t w11	Gradien t w12	Gradien t b1
1	1	0	1	0.74	0.66	0.65	-0.34	-0.223	-0.336	-0.129	-0.175	-0.177	0	-0.177
2	0	1	0	0	0	0.56	0.56	0	0.56	0	0	0	0	0
3	1	1	1	0.26	0.18	0.61	-0.39	-0.072	-0.393	-0.052	-0.198	-0.199	-0.199	-0.199
4	0	1	0	0.17	0.04	0.57	0.565	0.023	0.565	0.048	0.279	0	0	0.278

### 3.2.4. Cập nhật tham số Epoch 2:

➤ Với công thức sau :

$$w_{ij} = w_{ij} - \eta \cdot \frac{\partial L_i}{\partial w_{ij}}, b_i = b_i - \eta \cdot \frac{\partial L_i}{\partial b_i}.$$

Áp dụng cho mẫu 1 :

$$W31 = w31 - n * \text{gradient } 31 = 0.521 - 0.1 * (-0.223) = 0.543$$

$$B3 = b3 - n * \text{gradient } b3 = 0.335 - 0.1 * (-0.336) = 0.369$$

$$W21 = w21 - n * \text{gradient } 21 = 1.012 - 0.1 * (-0.129) = 1.025$$

$$B2 = b2 - n * \text{gradient } b2 = -0.082 - 0.1 * (-0.175) = -0.065$$

$$W11 = w11 - n * \text{gradient } 11 = 0.518 - 0.1 * (-0.177) = 0.535$$

$$W12 = w12 - n * \text{gradient } 12 = -0.5 - 0.1 * 0 = -0.500$$

$$B1 = b1 - n * \text{gradient } b1 = 0.218 - 0.1 * (-0.177) = 0.235$$

Tương tự với mẫu 2,3,4

Mẫu	W31	B3	W21	B2	W11	W12	B1
1	0.543	0.369	1.025	-0.065	0.535	-0.500	0.235
2	0.500	0.187	1.000	-0.129	0.500	-0.529	0.171

3	0.511	0.381	1.009	-0.060	0.541	-0.459	0.241
4	0.492	0.185	0.989	-0.157	0.500	-0.500	0.143

#### 4. Phân tích hiệu suất :

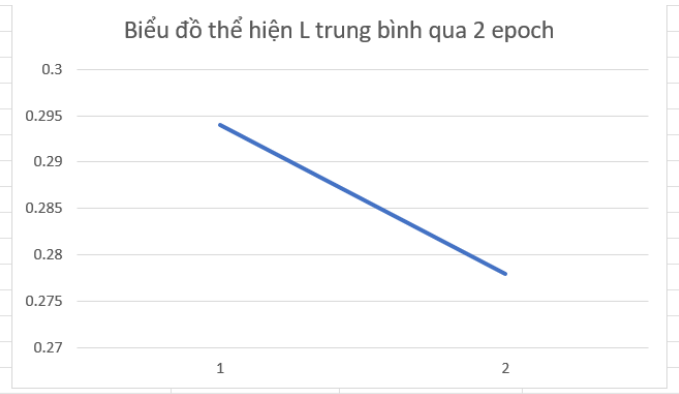
❖ Tính  $\hat{Y}$  và L :

Với tham số của Epoch mới cập nhật sau thì kết quả của  $\hat{Y}$  và L :

Mẫu	X1	X2	Y	H1	H2	$\hat{Y}$	L
1	1.0	0.0	1	0.735	0.662	0.664	0.178
2	0.0	1.0	0	0.000	0.000	0.560	0.357
3	1.0	1.0	1	0.262	0.184	0.607	0.217
4	0.0	0.0	0	0.171	0.040	0.565	0.362

➤ Trung Bình của L là :  $(0.178 + 0.357 + 0.217 + 0.362)/4 = 0.278$

❖ Vẽ sơ đồ hiện thị L trung bình của 2 Epoch :

[illegible]

 Nhận xét :

❖ **Độ lỗi trung bình (Loss) qua các Epoch:**

- Epoch 1: AVERAGE\_LossBinaryCrossEntropy = 0.294
- Epoch 2: AVERAGE\_LossBinaryCrossEntropy = 0.278
- Độ lỗi trung bình đã giảm từ 0.294 xuống 0.278 sau khi mô hình trải qua Epoch 2.

❖ **Nhận xét về mô hình:**

- Mô hình đã **cải thiện** sau Epoch 2 do độ lỗi giảm, chứng tỏ mô hình đang học tốt hơn từ dữ liệu.
- Việc giảm độ lỗi cho thấy quá trình tối ưu hóa đang hoạt động hiệu quả và mô hình đang dần tiến gần đến kết quả mong muốn.

❖ Nguyên nhân cải thiện:

- Trong quá trình huấn luyện, mô hình đã được cập nhật tham số thông qua quá trình tối ưu (ví dụ như Gradient Descent).
- Việc giảm độ lỗi là dấu hiệu tích cực, cho thấy mô hình đã điều chỉnh được các tham số theo hướng giảm thiểu sai số dự đoán.

## 5. Đề xuất cải tiến :

-Theo em một cải tiến có thể thực hiện là **tăng số lượng neuron trong các lớp ẩn**. Việc tăng số lượng neuron giúp mô hình có khả năng học được nhiều đặc trưng phức tạp hơn từ dữ liệu. Trong bài toán phân loại nhị phân, nếu số lượng neuron quá ít, mô hình có thể gặp hiện tượng **underfitting** (không đủ khả năng mô hình hóa dữ liệu phức tạp).

Ngoài ra, việc tăng số lượng neuron còn giúp **mở rộng không gian biểu diễn** của mô hình, từ đó có khả năng học tốt hơn đối với các mẫu có đặc trưng phức tạp hoặc phi tuyến tính. Tuy nhiên, cần lưu ý rằng việc tăng neuron cũng có thể làm tăng nguy cơ **overfitting** (quá khớp), đặc biệt khi dữ liệu không đủ lớn.

Để tránh overfitting, có thể áp dụng các kỹ thuật như **regularization** (L2, dropout) hoặc tăng cường dữ liệu (data augmentation). Đồng thời, cần theo dõi độ lỗi trên tập kiểm tra và tập huấn luyện để đánh giá hiệu quả của việc tăng neuron.

Với việc điều chỉnh số lượng neuron một cách hợp lý, mô hình có thể cải thiện khả năng tổng quát hóa và đạt được hiệu quả cao hơn.