Mạng nơ-ron nhân tạo

(Tài liệu nội bộ)

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

UIT
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Nội dung trình bày

Outline

1. Bài toán XOR

Hàm số Exclusive OR (XOR) Boolean

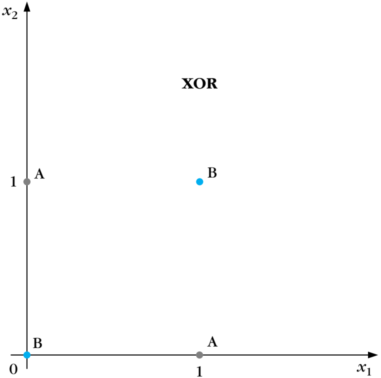


Figure 1: Các lớp A và B cho bài toán XOR.

x_1	x_2	XOR	Class
0	0	0	B
0	1	1	A
1	0	1	A
1	1	0	B

Figure 2: Bảng chân trị cho bài toán XOR.

- Các hàm số Boolean có thể được biểu diễn bởi các bài toán phân loại.
- Cụ thể, đầu vào gồm các giá trị nhị phân $\mathbf{x} = [x_1, x_2, ..., x_l]^T$, đầu ra là 0 hoặc 1, và $\mathbf{x}$ được phân loại thành một trong hai lớp A(1) và B(0).

Hàm số AND và OR Boolean

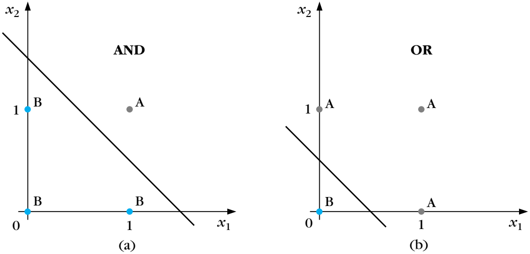


Figure 3: Các lớp A và B cho các bài toán (a) AND và (b) OR.

x_1	x_2	AND	Class	OR	Class
0	0	0	B	0	B
0	1	0	B	1	A
1	0	0	B	1	A
1	1	1	A	1	A

Figure 4: Bảng chân trị cho các bài toán AND và OR.

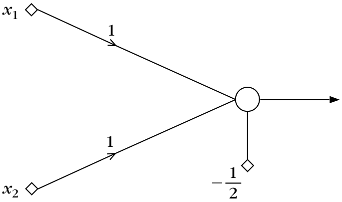


Figure 5: Một "perceptron" biểu diễn bài toán OR.

Outline

Perceptron hai lớp cho bài toán XOR

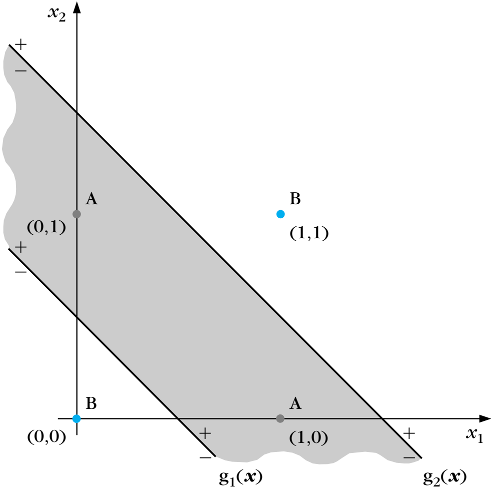


Figure 6: Các lớp A và B cho bài toán XOR.

1st Phase				
x_1	x_2	y_1	y_2	2nd Phase
0	0	0 (−)	0 (−)	B (0)
0	1	1 (+)	0 (−)	A (1)
1	0	1 (+)	0 (−)	A (1)
1	1	1 (+)	1 (+)	B (0)

Figure 7: Bảng chân trị cho bài toán XOR.

Perceptron hai lớp cho bài toán XOR (tiếp theo)

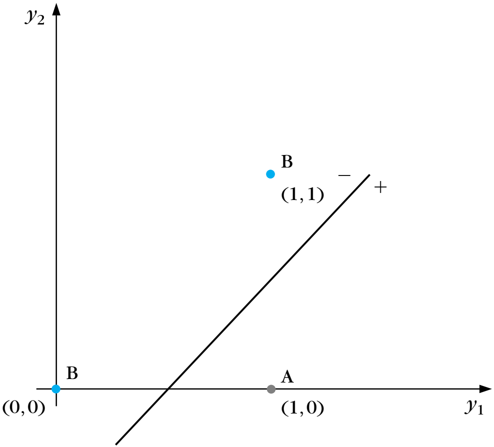


Figure 8: Đường thẳng phân loại hình thành bởi nơ-ron tại lớp thứ hai cho bài toán XOR.

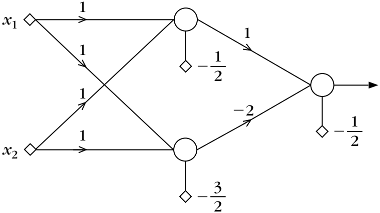


Figure 9: Perceptron hai lớp giải quyết bài toán XOR.

Khả năng phân loại của perceptron hai lớp

- Đối với trường hợp tổng quát hơn, ta giả sử các véc-tơ đầu vào $\mathbf{x} \in \mathcal{R}^l$, và lớp ẩn có p nơ-ron.
- Sử dụng hàm kích hoạt bậc thang, lớp ẩn ánh xạ không gian đầu vào lên trên các đỉnh của siêu khối H_p có độ dài cạnh đơn vị trong không gian p chiều. Các đỉnh của siêu khối là tất cả các điểm $[y_1, ..., y_p]^T$ của H_p với $y_i \in \{0, 1\}$, và $1 \leq i \leq p$, cụ thể H_p được định nghĩa như sau:

$$H_p = \left\{[y_1, ..., y_p]^T \in \mathcal{R}^p, y_i \in \{0, 1\}, 1 \leq i \leq p\right\} \quad (1)$$

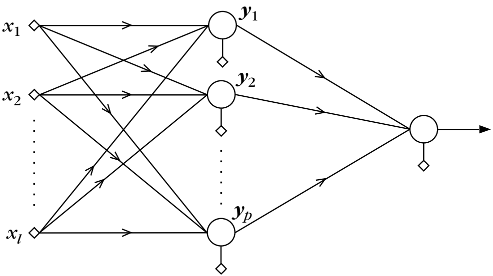


Figure 10: Kiến trúc của một perceptron hai lớp.

Khả năng phân loại của perceptron hai lớp (tiếp theo)

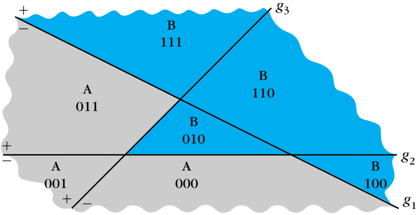


Figure 11: Các khối đa diện hình thành bởi các nơ-ron tại lớp ẩn thứ nhất của một perceptron hai lớp.

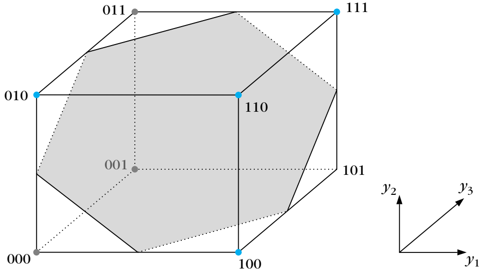


Figure 12: Các nơ-ron tại lớp ẩn thứ nhất ánh xạ véc-tơ đầu vào lên trên các đỉnh của một khối lập phương có độ dài cạnh đơn vị. Nơ-ron tại lớp đầu ra đóng vai trò của một mặt phẳng phân loại các đỉnh tương ứng với lớp của chúng.

[0]Một khối đa diện là phần giao nhau hữu hạn của các nửa không gian đóng thuộc $\mathcal{R}^l$, được định nghĩa bởi một số siêu phẳng.

Outline

3. Perceptron ba lớp

Perceptron ba lớp

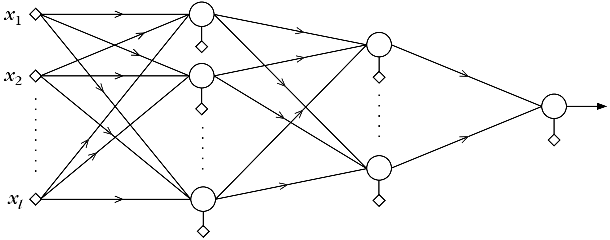


Figure 13: Kiến trúc của một perceptron đa lớp gồm hai lớp ẩn và một lớp đầu ra.

Outline

4. Thuật toán lan truyền ngược

Hàm kích hoạt (Activation function)

- Kiến trúc perceptron trong phần trước của bài học được phát triển dựa vào nơ-ron McCulloch–Pitts, sử dụng hàm bậc thang như là một hàm kích hoạt:

$$\mathrm{f}(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases} \tag{2}$$

- Hàm logistic, là một đại diện điển hình trong các hàm sigmoid, là một hàm liên tục có thể lấy đạo hàm, được coi là một hàm xấp xỉ của hàm bậc thang, cụ thể:

$$\mathrm{f}(x) = \frac{1}{1 + exp(-ax)} \tag{3}$$

trong đó a là hệ số góc.

Hàm kích hoạt (Activation function) (tiếp theo)

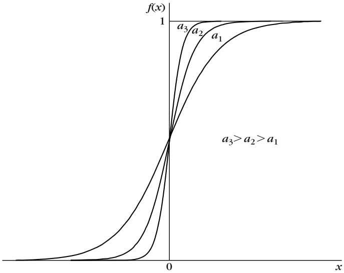


Figure 14: Hàm logistic. Giá trị hệ số góc a càng lớn thì hàm logistic càng gần với hàm bậc thang.

- Đôi khi, một biến thể của hàm logistic được sử dụng là một hàm phản xứng đối với nó, $\mathrm{f}(-x) = -\mathrm{f}(x)$, hàm này được định nghĩa như sau:

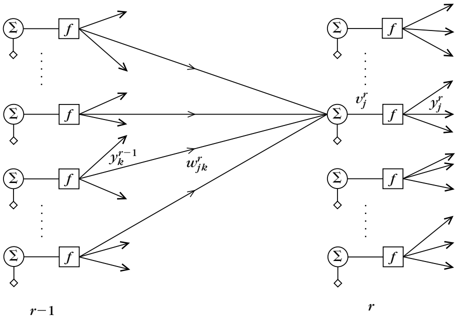
$$\mathrm{f}(x) = \frac{2}{1 + exp(-ax)} - 1 \qquad (4)$$

Giá trị của hàm này nằm trong đoạn -1 đến 1, và nó thuộc lớp các hàm tangent:

$$\mathrm{f}(x) = c\frac{1 - exp(-ax)}{1 + exp(-ax)} = c \tanh\left(\frac{ax}{2}\right) \qquad (5)$$

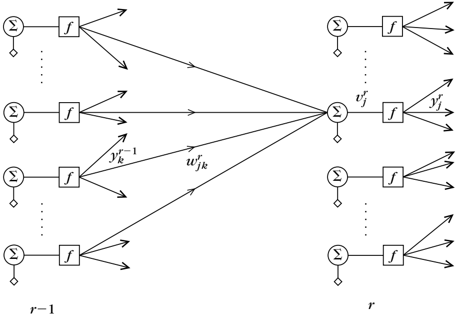
- Các hàm số vừa được giới thiệu có tên gọi phổ biến là hàm kích hoạt bởi vì đầu ra của chúng nằm trong phạm vi giá trị hữu hạn.

Một số quy ước về ký hiệu cho mạng perceptron



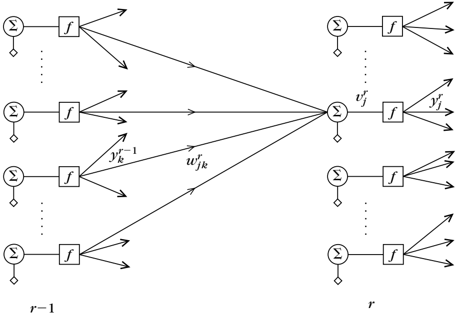
- Giả sử mạng perceptron đa lớp chứa L lớp cố định, với k_0 nút tại lớp đầu vào và k_r nơ-ron tại lớp thứ r, với $r = 1, 2, ..., L$.
- Giá trị của k_0 là l (l là số chiều của dữ liệu đầu vào như đã mô tả ở phần trước).
- Tất cả các nơ-ron sử dụng hàm kích hoạt sigmoid.

Một số quy ước về ký hiệu cho mạng perceptron (tiếp theo)



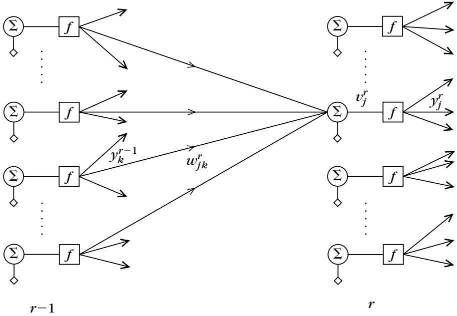
- Giả sử rằng có N cặp dữ liệu huấn luyện $(\mathbf{y}(i), \mathbf{x}(i))$, với $i = 1, 2, ..., N$.
- Véc-tơ đặc trưng đầu vào là véc-tơ có số chiều là k_0, $\mathbf{x}(i) = [x_1(i), ..., x_{k_0}(i)]^T$.
- Bởi vì có k_L nơ-ron tại lớp đầu ra, đầu ra là một véc-tơ k_L chiều, $\mathbf{y}(i) = [y_1(i), ..., y_{k_L}(i)]^T$.

Một số quy ước về ký hiệu cho mạng perceptron (tiếp theo)



- Khi huấn luyện, khi lớp đầu vào nhận véc-tơ $\mathbf{x}(i)$, đầu ra của mạng sẽ là $\hat{\mathbf{y}}(i)$, khác với giá trị mong muốn $\mathbf{y}(i)$.
- Các trọng số của mạng được tính toán sao cho một hàm chi phí J, dựa vào giá trị của $\mathbf{y}(i)$ và $\hat{\mathbf{y}}(i)$, $i = 1, 2, ..., N$ được tối thiểu hoá.
- Bởi vì hàm chi phí J phụ thuộc phi tuyến dựa vào $\hat{\mathbf{y}}(i)$ và các trọng số của mạng. Cho nên, việc tối thiểu hoá hàm chi phí J có thể thực hiện thông qua phương pháp lặp (gradient descent).

Một số quy ước về ký hiệu cho mạng perceptron (tiếp theo)

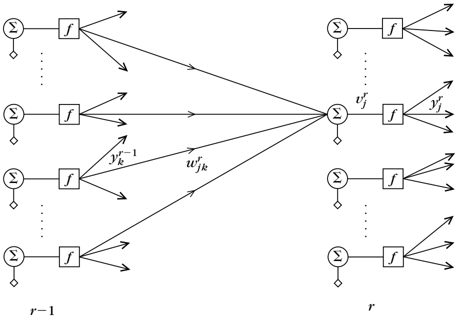


- Đặt $\mathbf{w}_j^r$ là véc-tơ trọng số của nơ-ron thứ j tại lớp thứ r, có số chiều là $k_{r-1}+1$ và được định nghĩa là $\mathbf{w}_j^r = [w_{j0}^r, w_{j1}^r, ..., w_{jk_{r-1}}^r]^T$. Một vòng lặp cơ bản sẽ có dạng như sau:

$$\mathbf{w}_j^r(\text{new}) = \mathbf{w}_j^r(\text{old}) + \Delta\mathbf{w}_j^r = \mathbf{w}_j^r(\text{old}) - \mu\frac{\partial J}{\partial \mathbf{w}_j^r} \quad (6)$$

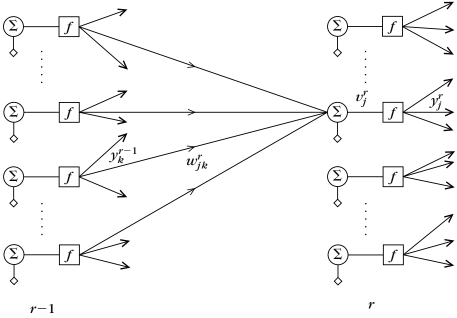
trong đó, $\mathbf{w}_j^r(\text{old})$ là ước lượng hiện tại của các trọng số chưa biết, $\Delta\mathbf{w}_j^r$ là sự điều chỉnh trọng số để đạt được ước lượng tiếp theo $\mathbf{w}_j^r(\text{new})$.

Một số quy ước về ký hiệu cho mạng perceptron (tiếp theo)



- Giá trị v_j^r là tổng có trọng số của các giá trị đầu vào của nơ-ron thứ j của lớp thứ r và giá trị y_j^r là đầu ra tương ứng sau khi áp dụng hàm kích hoạt.

Một số quy ước về ký hiệu cho mạng perceptron (tiếp theo)

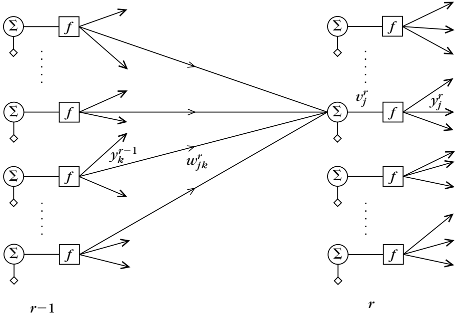


- Hàm chi phí J có dạng như sau:

$$J = \sum_{i=1}^{N} \varepsilon(i) \tag{7}$$

trong đó, ε là một hàm xác định thích hợp dựa trên $\hat{\mathbf{y}}(i)$ và $\mathbf{y}(i)$, với $i = 1, 2, ..., N$. Nói cách khác, J được biểu diễn bởi tổng của N giá trị của hàm ε cho các cặp huấn luyện $(\mathbf{y}(i), \mathbf{x}(i))$.

Một số quy ước về ký hiệu cho mạng perceptron (tiếp theo)

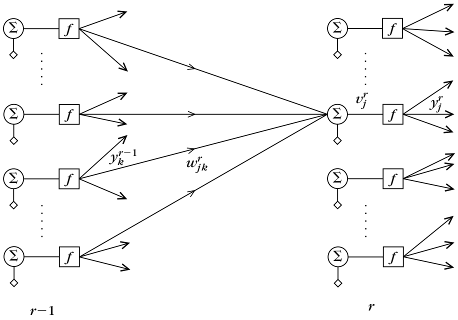


- Ví dụ, ta có thể chọn $\varepsilon(i)$ bởi tổng của các bình phương độ lỗi tại các nơ-ron đầu ra:

$$\varepsilon(i) = \frac{1}{2}\sum_{m=1}^{k_L} e_m^2(i) \equiv \frac{1}{2}\sum_{m=1}^{k_L}(y_m(i) - \hat{y}_m(i))^2, i = 1, 2, ..., N \qquad (8)$$

Đối với thành phần điều chỉnh trọng số $\Delta\mathbf{w}_j^r$ trong biểu thức (6), đạo hàm của hàm chi phí đối với các trọng số bây giờ trở thành $\partial\varepsilon(i)/\partial\mathbf{w}_j^r$.

Tính toán các biểu thức đạo hàm

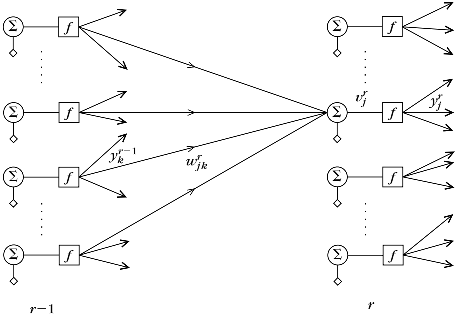


- Đặt y_k^{r-1} là đầu ra của nơ-ron thứ k, $k = 1, 2, ..., k_{r-1}$, tại lớp thứ $(r-1)$ cho cặp huấn luyện thứ i và w_{jk}^r là ước lượng hiện tại của trọng số dẫn đến nơ-ron thứ j tại lớp r, với $j = 1, 2, ..., k_r$. Vì vậy, tham số của hàm kích hoạt $f(\cdot)$ của phần sau nơ-ron sẽ là:

$$v_j^r(i) = \sum_{k=1}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i) + w_{j0}^r \equiv \sum_{k=0}^{k_{r-1}} w_{jk}^r y_k^{r-1}(i) \tag{9}$$

bằng cách định nghĩa $y_0^r \equiv +1, \forall r, i$.

Tính toán các biểu thức đạo hàm



- Đối với lớp đầu ra, ta có $r = L$, $y_k^r(i) = \hat{y}_k(i), k = 1, 2, ..., k_L$ là các đầu ra của mạng nơ-ron.
- Đối với lớp đầu vào, ta có $r = 1$, $y_k^{r-1}(i) = x_k(i), k = 1, 2, ..., k_0$ là các đầu vào của mạng nơ-ron.

Tính toán các biểu thức đạo hàm

- Từ biểu thức (8) và (9), bởi vì $\varepsilon(i)$ phụ thuộc vào $\mathbf{w}_j^r$ thông qua $v_j^r(i)$, bằng cách áp dụng quy tắc đạo hàm cho hàm hợp, ta có:

$$\frac{\partial \varepsilon(i)}{\partial \mathbf{w}_j^r} = \frac{\partial \varepsilon(i)}{\partial v_j^r(i)} \frac{\partial v_j^r(i)}{\partial \mathbf{w}_j^r} \tag{10}$$

- Từ biểu thức (9), ta được:

$$\frac{\partial}{\partial \mathbf{w}_j^r} v_j^r(i) \equiv \begin{bmatrix} \frac{\partial}{\partial w_{j0}^r} v_j^r(i) \\ \vdots \\ \frac{\partial}{\partial w_{jk_{r-1}}^r} v_j^r(i) \end{bmatrix} = \mathbf{y}^{r-1}(i) \tag{11}$$

trong đó

$$\mathbf{y}^{r-1}(i) = \begin{bmatrix} +1 \\ y_1^{r-1}(i) \\ \vdots \\ y_{k_{r-1}}^{r-1}(i) \end{bmatrix} \tag{12}$$

Tính toán các biểu thức đạo hàm (tiếp theo)

- Từ biểu thức (10), ta đặt:

$$\frac{\partial \varepsilon(i)}{\partial v_j^r(i)} \equiv \delta_j^r(i) \tag{13}$$

thì thành phần điều chỉnh trọng số $\Delta \mathbf{w}_j^r$ trong biểu thức biểu thức (6) trở thành:

$$\Delta \mathbf{w}_j^r = -\mu \sum_{i=1}^{N} \delta_j^r(i) \mathbf{y}^{r-1}(i) \tag{14}$$

Biểu thức (10) là tổng quát cho bất kỳ hàm chi phí nào có thể lấy đạo hàm trong biểu thức (7). Trong phần tiếp theo, biểu thức của $\delta_j^r(i)$ được khai triển cụ thể ứng với trường hợp bình phương nhỏ nhất. *Cách thức biến đổi sẽ tương tự đối với các hàm chi phí khác.*

Tính toán thành phần $\delta_j^r(i)$ cho hàm chi phí trong biểu thức (8)

Quá trình tính toán bắt đầu từ lớp $r = L$ và lan truyền ngược đến lớp $r = L - 1$, $L - 2$, ..., 1. Đây là lý do tại sao thuật toán chúng ta đang tìm hiểu được biết đến với tên gọi "lan truyền ngược".

- Tại lớp $r = L$, ta có:

$$\delta_j^L(i) = \frac{\partial \varepsilon(i)}{\partial v_j^L(i)} \tag{15}$$

$$\varepsilon(i) \equiv \frac{1}{2}\sum_{m=1}^{k_L} e_m^2(i) \equiv \frac{1}{2}\sum_{m=1}^{k_L}(\mathrm{f}(v_m^L(i)) - y_m(i))^2 \tag{16}$$

Vì vậy

$$\delta_j^L(i) = e_j(i)\mathrm{f}^{'}(v_j^L(i)) \tag{17}$$

trong đó $\mathrm{f}^{'}$ là đạo hàm của hàm $\mathrm{f}(\cdot)$. Bởi vì hàm $\mathrm{f}(\cdot)$ là sigmoid, ta có:

$$\mathrm{f}^{'}(x) = a\mathrm{f}(x)(1 - \mathrm{f}(x)) \tag{18}$$

Tính toán thành phần $\delta_j^r(i)$ cho hàm chi phí trong biểu thức (8) (tiếp theo)

Tại lớp $r < L$, ta có:

$$\frac{\partial \varepsilon(i)}{\partial v_j^{r-1}(i)} = \sum_{k=1}^{k_r} \frac{\partial \varepsilon(i)}{\partial v_k^r(i)} \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} \quad (19)$$

nếu theo quy ước trong biểu thức (13), biểu thức (19) trở thành:

$$\delta_j^{r-1}(i) = \sum_{k=1}^{k_r} \delta_k^r(i) \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} \quad (20)$$

Ngoài ra, ta có:

$$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = \frac{\partial [\sum_{m=0}^{k_{r-1}} w_{km}^r y_m^{r-1}(i)]}{\partial v_j^{r-1}(i)} \quad (21)$$

trong đó $y_m^{r-1}(i) = \mathrm{f}(v_m^{r-1}(i))$, nên ta có:

$$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = w_{kj}^r \mathrm{f}'(v_j^{r-1}(i)) \quad (22)$$

Từ biểu thức (20) và (22), ta có kết quả sau:

$$\delta_j^{r-1}(i) = \left[\sum_{k=1}^{k_r} \delta_k^r(i) w_{kj}^r\right] \mathrm{f}'(v_j^{r-1}(i)) \quad (23)$$

Nếu viết theo quy ước trong biểu thức (17), biểu thức (23) trở thành:

$$\delta_j^{r-1}(i) = \left[e_j^{r-1}(i)\right] \mathrm{f}'(v_j^{r-1}(i)) \quad (24)$$

Thuật toán lan truyền ngược cho mạng perceptron đa lớp

- Khởi tạo: Khởi tạo tất cả các trọng số với các giá trị nhỏ ngẫu nhiên.
- Tính toán chuyển tiếp: Với mỗi véc-tơ đặc trưng $\mathbf{x}(i)$, $i = 1, 2, ..., N$, tính tất cả giá trị $v_j^r(i)$, $y_j^r(i) = \mathrm{f}(v_j^r(i))$, $j = 1, 2, ..., k_r$, $r = 1, 2, ..., L$ từ biểu thức (8). Tính hàm chi phí cho các tham số ước lượng hiện tại từ biểu thức (7) và (16).
- Tính toán truyền ngược: Với mỗi $i = 1, 2, ..., N$ và $j = 1, 2, ..., k_L$ tính các giá trị $\delta_j^L(i)$ từ biểu thức (17). Tiếp theo, tính các giá trị $\delta_j^{r-1}(i)$ từ biểu thức (24) với $r = L$, $L-1$, ..., 2, và $j = 1, 2, ..., k_r$.
- Cập nhật giá trị các trọng số: Với $r = 1, 2, ..., L$ và $j = 1, 2, ..., k_r$, thực hiện:

$$\mathbf{w}_j^r(\text{new}) = \mathbf{w}_j^r(\text{old}) + \Delta\mathbf{w}_j^r \tag{25}$$

$$\Delta\mathbf{w}_j^r = -\mu \sum_{i=1}^{N} \delta_j^r(i)\mathbf{y}^{r-1}(i) \tag{26}$$

Outline

Bài tập

- Bài tập 1: Viết chương trình thực hiện thuật toán lan truyền ngược cho bài toán XOR với kiến trúc mạng perceptron gồm một lớp ẩn có **hai** nơ-ron và lớp đầu ra có **một** nơ-ron. Hàm kích hoạt tại mọi nơ-ron là sigmoid. Trong đó hàm mất mát là tổng bình phương.
- Bài tập 2: Viết chương trình thực hiện thuật toán lan truyền ngược cho bài toán XOR với kiến trúc mạng perceptron gồm một lớp ẩn có **hai** nơ-ron và lớp đầu ra có **hai** nơ-ron. Trong đó hàm chi phí là cross-entropy:

$$J = -\sum_{i=1}^{N}\sum_{k=1}^{k_L} y_k(i) \ln\left(\frac{\hat{y}_k(i)}{y_k(i)}\right) \tag{27}$$

Tất các các nơ-ron trong mạng sử dụng hàm kích hoạt sigmoid, ngoại lệ, hàm kích hoạt là softmax cho các nơ-ron tại lớp đầu ra:

$$\hat{y}_k \equiv \frac{\exp(v_k^L)}{\sum_{k'} \exp(v_{k'}^L)} \tag{28}$$

Tài liệu tham khảo

- Sách tham khảo Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow, 2nd Edition của tác giả Aurélien Géron.
- Sách tham khảo Pattern Recognition, 4th Edition của nhóm tác giả Sergios Theodoridis và Konstantinos Koutroumbas.