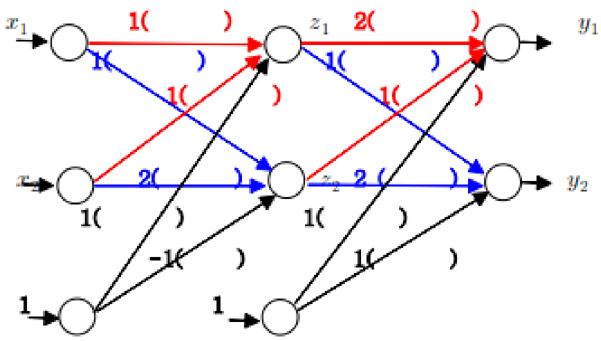
2023 Spring Artificial Intelligence Final Exam.

ID (2021720639) Name (이찬우)

1. Following 2 layer neural network with initial weights is given. Weight colors are matched with colors of connections. The activation function of each node is <u>sigmoid function</u>. Input data X=(1, -1), desired output value d=(1, 2) is given. The computed output values for hidden nodes and output nodes in forward pass are as follows. (It is assumed that these values are correct. You <u>need not</u> compute these values) z_1= 0.5, z_2= -0.7 , y_1=0.2, y_2= -0.8. (refer lecture notes for notation)

Apply error backpropagation algorithm, and



① **compute $\boldsymbol{\Delta_1, \Delta_2, \delta_1, \delta_2}$**

$$\Delta_1 = (d_1 - y_1) \times y_1 \times (1 - y_1)$$
$$= (1 - 0.2) \times 0.2 \times (1 - 0.2)$$
$$= 0.128$$

$$\Delta_2 = (d_2 - y_2) \times y_2 \times (1 - y_2)$$
$$= \left(2 - -(0.8)\right) \times -(0.8) \times \left(1 - -(0.8)\right)$$
$$= -1.024$$

$$\delta_1 = z_1 \times (1 - z_1) \times \left(w_{13} \times \Delta_1 + w_{14} \times \Delta_2\right)$$
$$= 0.5 \times (1 - 0.5) \times (2 \times 0.128 - 1.024) = -0.12$$

$$\delta_2 = z_2 \times (1 - z_2) \times \left(w_{23} \times \Delta_1 + w_{24} \times \Delta_2\right)$$
$$= (-0.7) \times \left(1 - (-0.7)\right) \times (0.128 - 2.048) = 0.275$$

② write updated weights in (   )'s

학습율은 $\alpha = 0.1$ 이라고 하고 가중치와 편향을 구하면 다음과 같다.

$\Delta w_{11} = \alpha \times \delta_1 \times x_1 = 0.1 \times (-0.12) \times 1 = -0.012$이므로

$w_{11} = w_{11} + \Delta w_{11} = 1 - 0.012 = 0.988$이다.

$\Delta w_{12} = \alpha \times \delta_1 \times x_2 = 0.1 \times (-0.12) \times (-1) = 0.012$이므로

$w_{12} = w_{12} + \Delta w_{12} = 1 + 0.012 = 1.012$이다.

$\Delta w_{21} = \alpha \times \delta_2 \times x_1 = 0.1 \times 0.275 \times 1 = 0.0275$이므로

$w_{21} = w_{21} + \Delta w_{21} = 1 + 0.0275 = 1.0275$이다.

$\Delta w_{22} = \alpha \times \delta_2 \times x_2 = 0.1 \times 0.275 \times (-1) = -0.0275$이므로

$w_{22} = w_{22} + \Delta w_{22} = 2 - 0.0275 = 1.9725$이다.

$\Delta w_{13} = \alpha \times \Delta_1 \times z_1 = 0.1 \times 0.128 \times 0.5 = 0.0064$이므로

$w_{13} = w_{13} + \Delta w_{13} = 2 + 0.0064 = 2.0064$이다.

$\Delta w_{14} = \alpha \times \Delta_2 \times z_1 = 0.1 \times (-1.024) \times 0.5 = -0.0512$이므로

$w_{14} = w_{14} + \Delta w_{14} = 2 - 0.0512 = 0.9488$이다.

$\Delta w_{23} = \alpha \times \Delta_1 \times z_2 = 0.1 \times 0.128 \times (-0.7) = -0.0896$이므로

$w_{23} = w_{23} + \Delta w_{23} = 1 - 0.00896 = 0.99104$이다.

$\Delta w_{24} = \alpha \times \Delta_2 \times z_2 = 0.1 \times (-1.024) \times (-0.7) = -0.07168$이므로

$w_{24} = w_{24} + \Delta w_{24} = 2 + 0.007168 = 2.07168$이다.

$\Delta b_1 = \alpha \times \delta_1 = 0.1 \times -0.12 = -0.012$이므로

$b_1 = b_1 + \Delta b_1 = 1 - 0.012 = 0.988$이다

$\Delta b_2 = \alpha \times \delta_2 = 0.1 \times 0.275 = 0.0275$이므로

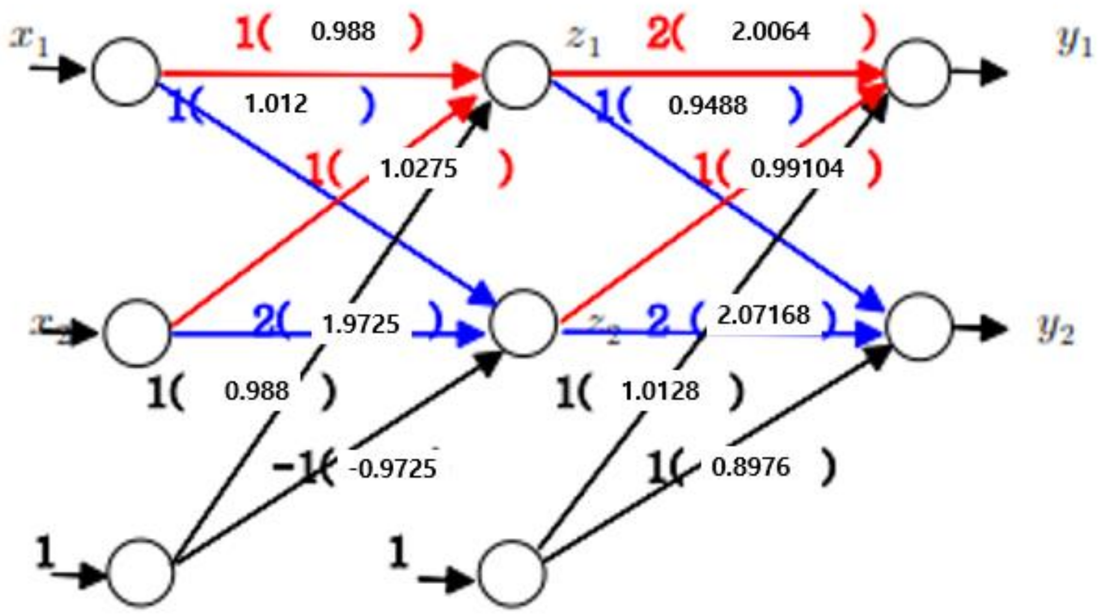
$b_2 = b_2 + \Delta b_2 = -1 + 0.0275 = -0.9725$이다.

$\Delta b_3 = \alpha \times \Delta_1 = 0.1 \times 0.128 = 0.0128$이므로

$b_3 = b_3 + \Delta b_3 = 1 + 0.0128 = 1.0128$이다.

$\Delta b_4 = \alpha \times \Delta_2 = 0.1 \times -1.024 = -0.1024$이므로

$b_4 = b_4 + \Delta b_4 = 1 - 0.1024 = 0.8976$이다.



2. Following training data set S is given for class1 and class2. For each class, five data are collected. Each data has three attributes, (size, number, color). The first attribute "size" has value set (large, middle, small), the second attribute "number" has value set (2,3,5), and the third attribute "color" has value set (black, white). We will apply ID-3 algorithm.

S: class1 {(middle, 2, black), (small, 5, white), (middle, 5, black), (large, 5, black), (large, 5, white)}

class2 {(small, 3, black), (small, 3, white), (middle, 3, white), (small, 2, black), (middle, 3, black)}

① Compute entropy of initial training set s, i.e. H(S)

엔트로피를 구하는 공식은 다음과 같다.

$$H(S) = -p \times log_2 p - q \times log_2 q$$

class1과 class2의 크기는 각각 5로 총 크기는 10이다. 따라서 S의 엔트로피는 다음과 같다.

$$H(S) = -0.5 \times log_2 0.5 - 0.5 \times log_2 0.5 = 1.0$$

② Compute information gain Gain(S, color) when set S is classified using attribute color.

$$Gain(S, A) = H(S) - \sum [(|S_v|/|S|) \times H(S_v)]$$

S_v는 속성 A가 값 v를 갖는 S의 부분 집합이다. $|S_v|$는 부분집합 S_v의 크기이고 $|S|$는 집합 S의 크기이다. $H(S_v)$는 S_v의 엔트로피이다.

"색상" 속성을 사용하여 집합 S를 분류하면 S_{black}, S_{white} 두 부분집합을 얻는다.

S_{black}의 엔트로피는 $p = 3/7 \approx 0.429$이므로

$$H(S_{black}) = -0.429 \times log_2 0.429 - 0.571 \times log_2 0.571 \approx 0.985$$

S_{white}의 엔트로피는 $p = 2/3 \approx 0.667$이므로

$$H(S_{white}) = -0.667 \times log_2 0.667 - 0.333 \times log_2 0.333 \approx 0.918$$

따라서 $Gain(S, color)$을 계산하면 다음과 같다.

$$Gain(S, color) = H(S) - [(|S_{black}|/|S|) \times H(S_{black}) + (|S_{white}|/|S|) \times H(S_{white})]$$

$$Gain(S, color) = 1.0 - [(7/10) \times 0.985 + (3/10) \times 0.918] \approx 0.021$$

③ What will be the attribute selected as root node of decision tree? Show the derivation steps, too.

"color" 속성에 대한 information gain을 이미 계산했으므로 "size"와 "number"에 대해 계산해야 한다.

먼저 "size" 속성을 사용하여 집합 S을 $S_{small}, S_{middle}, S_{large}$로 분류한다.

각 부분 집합에 대한 엔트로피를 계산하면 다음과 같다.

$$H(S_{small}) = -1/3 \times log_2 1/3 - 2/3 \times log_2\{2/3\} \approx 0.918$$

$$H(S_{middle}) = -2/4 \times log_2 2/4 - 2/4 \times log_2\{2/4\} \approx 1.0$$

$$H\left(S_{large}\right) = -2/3 \times log_2 2/3 - 1/3 \times log_2\{1/3\} \approx 0.918$$

따라서 $Gain(S, size)$을 계산하면 다음과 같다.

$$Gain(S, size) = H(S) - [(|S_{small}|/|S|) \times H(S_{small})$$

$$+(|S_{middle}|/|S|) \times H(S_{middle}) + (|S_{large}|/|S|) \times H(S_{large})]$$

$$Gain(S, size) = 1.0 - [(3/10\ \times 0.918)\ + (4/10\ \times 1.0)\ + (3/10 \times 0.918)] \approx 0.048$$

다음으로 "number" 속성을 사용하여 집합 S을 S_2, S_3, S_5로 분류한다.

각 부분 집합에 대한 엔트로피를 계산하면 2는 class1과 class2의 수가 같으므로 $H(S_2) = 1.0$이다. 3은 모두 class2에서 온 것임으로 $H(S_3) = 0$이다. 5는 모두 class1에서 온 것임으로 $H(S_5) = 0$이다.

따라서 $Gain(S, number)$을 계산하면 다음과 같다.

$$Gain(S, number) = H(S) - [(S_2|/|S|) \times H(S_2)$$

$$+(|S_3|/|S|) \times H(S_3) + (|S_5|/|S|) \times H(S_5)]$$

$$Gain(S, number) = 1.0 - [(2/10\ \times 1.0)\ + (4/10\ \times 0)\ + (4/10 \times 0)] \approx 0.8$$

모든 속성의 information gain을 비교하면

$$Gain(S, color) = 0.021$$

$$Gain(S, size) = 0.048$$

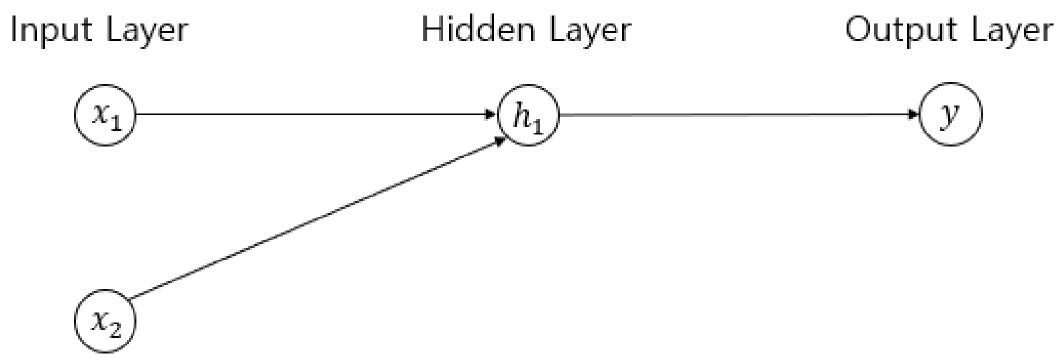
$$Gain(S, number) = 0.8$$

이므로, information gain이 가장 높은 속성은 "number"이다. 따라서 의사 결정 트리의 루트 노드로 선택되는 것은 "number"이다.

3. Following data set S of 4 training data is given. In each pair, the first component is input data and the second data is the desired output. For example, the first training data has input data (0, 0) and the desired output is 1. Input data has two attributes, x_1 and x_2.

S = {<(0, 0), 1>, <(1, 1), 1>, <(1, 0), 0>, <(0, 1), 0>}

① Design 2-layer perceptron, i.e. show the architecture.



② Compute weights for connections. Use hard limit function as activation function.

Activation function을 기반으로 가중치를 계산할 수 있다. Hard limit function을 사용하고 있으므로 activation function은 다음과 같다.

$$y = \mathrm{hardlimit}(\sum(\mathrm{w_i} \times \mathrm{x_i}) + \mathrm{b}) = \begin{cases} 1, \text{if } \sum(\mathrm{w_i} \times \mathrm{x_i}) + \mathrm{b} \geq 0 \\ 0, \text{if } \sum(\mathrm{w_i} \times \mathrm{x_i}) + \mathrm{b} < 0 \end{cases}$$

$w_1 = w_2 = b = 0$ 으로 초기화 하고 데이터 세트를 반복했을 때, <(0, 0), 1>는 $y = hardlimit(0 \times 0 + 0 \times 0 + 0) = 0$이고 $w_1 = 0 + (1 + 0) \times 0 = 0, w_2 = 0 + (1 - 0) \times 0 = 0, b = 0 + (1 - 0) = 1$ 이다. <(1, 1), 1>는 $y = hardlimit(0 \times 1 + 0 \times 1 + 1) = 1$ 이고 가중치와 편향의 수정은 없다. <(1, 0), 0>는 $y = hardlimit(0 \times 1 + 0 \times 0 + 1) = 1$ 이고 $w_1 = 0 + (0 - 1) \times 1 = -1, w_2 = 0 + (0 - 1) \times 0 = 0, b = 1 + (0 - 1) = 0$ 이다. <(0, 1), 1>는 $y = hardlimit(0 \times -1 + 1 \times 0 + 0) = 1$이고 가중치와 편향의 수정은 없다.

이러한 단계를 적용하여 수렴될 때까지 가중치와 편향을 반복적으로 업데이트할 수 있다.

4. Digital image is shown as fig. (a). Perform mask(template matching) operation with a mask given in fig. (b) for a center pixel(i.e. a pixel with a grey level of 15) of fig (a).

5	8	9
8	15	7
7	9	12

fig (a) image

-1	-1	-1
-1	8	-1
-1	-1	-1

fig (b) mask(template)

이미지의 중앙 픽셀은 15의 회색 레벨을 가지며 마스크의 다른 모든 픽셀은 중앙의 8을 제외하고 모두 -1이다. 작업은 다음과 같이 진행된다.

$(-1 \times 5) + (-1 \times 8) + (-1 \times 9) + (-1 \times 8) + (8 \times 15)$

$+(-1 \times 7) + (-1 \times 7) + (-1 \times 9) + (-1 \times 12)$

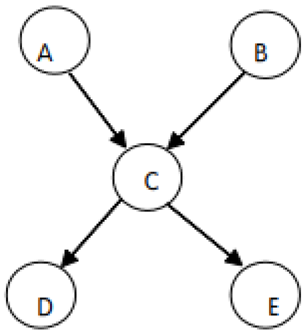
$= -5 - 8 - 9 - 8 + 120 - 7 - 7 - 9 - 12$

$= 55$

즉, 이미지 중앙에 (b)를 적용하면 값이 55가 된다. Template matching의 해당 위치의 이미지와 얼마나 잘 일치하는지를 나타낼 수 있다. 절대값이 높을수록 반응이 강함을 나타내며, 이 경우 상대적으로 높은 점수를 생성하여 잘 일치하는 것을 알 수 있다.

5.

① For the following Bayesian network, specify the necessary probability table for each node.



Node A:

A	**P(A)**
True	P(A=True)
False	P(A=False)

Node B:

B	**P(B)**
True	P(B=True)
False	P(B=False)

Node C:

A	**B**	**P(C=True)**	**P(C=False)**
True	True	P(C=True | A=True, B=True)	P(C=False | A=True, B=True)
True	False	P(C=True | A=True, B=False)	P(C=False | A=True, B=False)
False	True	P(C=True | A=False, B=True)	P(C=False | A=False, B=True)
False	False	P(C=True | A=False, B=False)	P(C=False | A=False, B=False)

Node D:

C	**P(D=True)**	**P(D=False)**
True	P(D=True | C=True)	P(D=False | C=True)
False	P(D=True | C=False)	P(D=False | C=False)

Node E:

C	**P(E=True)**	**P(E=False)**
True	P(E=True | C=True)	P(E=False | C=True)
False	P(E=True | C=False)	P(E=False | C=False)

② Using the probability in the above table,

compute P(A=false, B=false, C=true)=P(~A, ~B, C)

P(A=False, B=False, C=True)=P(~A, ~B, C)

= P(C=True | A=False, B=False) × P(A=False) × P(B=False)

= P(C | ~A, ~B) × P(~A) × P(~B)

③ Compute P(A=true, b=true, C=true, D=true, E=false)=P(A, B, C, D, ~E).

P(A=True, b=True, C=True, D=True, E=False) = P(A, B, C, D, ~E)

= P(A=True) × P(B=True) × P(C=True | A=True, B=True) × P(D=True | C=True) × P(E=False | C=True)

= P(A) × P(B) × P(C | A, B) × P(D | C) × P(~E | C)