신경망 구현하기

$$(h_1, h_2, h_3, h_4) = (x_1, x_2)\begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \end{pmatrix} + (b_1, b_2, b_3, b_4)$$

출력 (추정값) (1×4) → (N×4)

특징벡터 (입력) (1×2) → (N×2)

가중치행렬 (2×4) → (2×4) (input d, output d)

bias (1×4) → (N×4)

⇒ $\underline{h} = \underline{x} \cdot w + \underline{b}$ (MLP → perceptron의 집합) (각 층이라는 layer로 둠)

$\underline{H} = \underline{X} \cdot W + \underline{B}$ ← Affine Layer

• Sigmoid function (활성화함수)

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

* $\underline{X}$ → Affine → Sigmoid → Affine → $\underline{O}$ (순전파, propagation) → 순차적으로 계산을 진행함.

오차역전파 (Error Back-propagation)

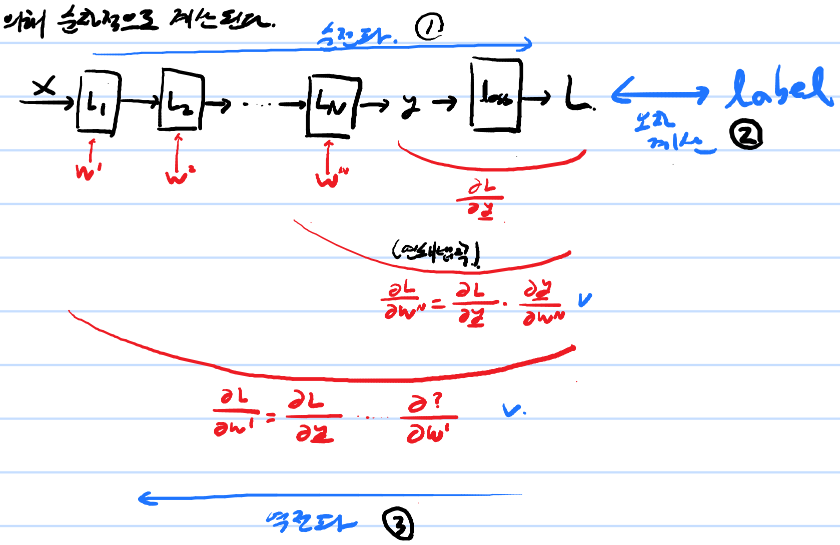
• 신경망의 기계학습과정에서, 주어진 학습데이터 (특징벡터)에 대해 손실값을 계산해야 한다.

• 신경망의 매개변수를 업데이트하기 위해서는 손실함수의 그래디언트를 계산해야 한다.

• 손실함수 L, 매개변수 w에 대하여: (매개변수 w에 대한 기울기)

$$\frac{\partial L}{\partial W} = \begin{pmatrix} \frac{\partial L}{\partial w_{11}} & \cdots & \frac{\partial L}{\partial w_{1n}} \\ \vdots & & \vdots \\ \frac{\partial L}{\partial w_{m1}} & \cdots & \frac{\partial L}{\partial w_{mn}} \end{pmatrix} \in \mathbb{R}^{m \times n}, \quad W \in \mathbb{R}^{m \times n}$$

• 신경망은 여러 개의 perceptron이 순차적으로 연결된 형태이므로, Gradient는 연쇄법칙에 의해 순차적으로 계산된다.



• 신경망의 출력에서 입력 방향으로 거꾸로 진행되는 것처럼 보여서 역전파라는 표현을 씀

연쇄법칙 (chain rule)

• 합성함수에 대한 미분법칙.

• 오차역전파법에서 사용한다.

ex) $y = f(x)$, $z = g(y)$에 대해

→ $z = g(f(x))$

→ $\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x}$