2024.11.30

PRMLゼミ 5.1. Feed-forward Network Functions.

[1]

$$y(x, w) = f\left(\underbrace{\sum_{j=1}^{M} w_j \phi_j(x)}_{\text{線型和}}\right)$$

f：非線型活性化関数　$\phi_j(x)$：非線型基底関数

NNでも同様

activation(活性化、ポテンシャル)

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}$$

$(x_1, \cdots, x_D \rightarrow D$次元入力$)$

$j = 1, \cdots, M \mapsto$ 入力レイヤー.

非線型活性化関数.

$$z_j = h(a_j)$$

h : sigmoid, tanh とか.（微分可能）

→ DeepならReLUもよく使う

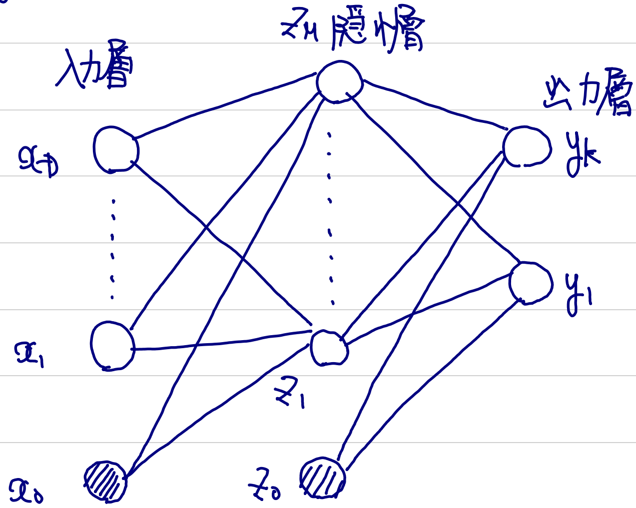
$$a_k = \sum_{i=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}$$

$$y_k = \sigma(a_k)$$

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$

σ : sigmoid (2値分類)（微分可能）

→ 回帰なら恒等写像



← 3層ネットワーク
入力層も含む

◎2層ネットワーク
適応層の数.

◎単一隠れ層ネットワーク

の3つの呼び方があり得る.

(p.229

まとめると、

$$y_k(x, w) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right)$$

→順伝播の式

また、$x_0 = 1$とすると、バイアス項 $w_{j0}^{(1)}$, $w_{k0}^{(2)}$ もまとめられて.

$$y_k(x, w) = \sigma\left(\sum_{j=0}^{M} w_{kj}^{(2)} h\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i\right)\right)$$

[2] NNとパーセプトロンの違い

微分可能

NN → σ (sigmoid ; continuous, differentiable)

key role in NN

↕

パーセプトロン → (step ; 微分不可!)

・もし、全ての隠れ層の活性化関数が「線型」なら… ☆

↓

線型な関数(行列)の積としてまとめられ. 隠れ層がないのと同等.

↓

☆のような状況には関心がない。

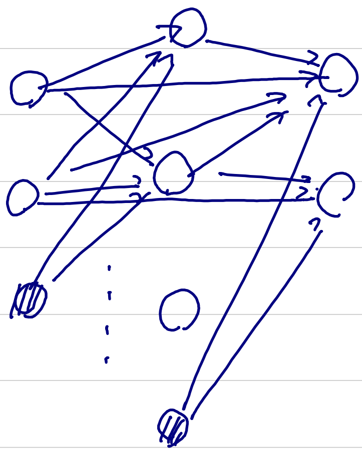
・もし隠れ層の次元が入出力の次元より小さいなら…

↓

次元削減 = 一般的な「線型写像にならない」

「主成分分析」を得る (意味はある).

[3] 層を飛び超えた結合.



sigmoid を隠れ層に持つ.

↓

○ ⟶ ○ ⟶ ○ とする.

小 << 大

(実際には左図をプログラムするわけない…)

[4] サイクルがあると数学表現できない (戻ると定義できない)

⇓

Feed-forward でないとダメ.

└→ 決定論 ⟷ ~~確率論~~

F.F　　　　リカレント (← カオス).

[5] NN = 万能近似器 (図 5.3)

[6] <u>x → 隠</u>

$\boxed{\text{重み空間対称性}}$

$$\overset{-a}{\tanh(-a)} = \overset{a}{\tanh(a)}$$

↓

for each weight (num = M)

2^M 個の等価なベクトル

<u>隠 → 出</u>

→ 対称 : $M!$

$M!\,2^M$ の等価な重み

⟶ どーでもいい.