物理学とディープラーニング (ゼミ)

B4 柳瀬調知

2023 年 4 月 19 日

第 4 章

・そもそもニューロンとは？:図 4.1 参照

人間の脳を作っている神経細胞で脳機能の構成要素のこと。あるニューロンからのびた軸索が別のニューロンの樹状突起と呼ばれる部分に結合しており、電気信号が伝達される。シナプスに来た信号はある閾値を超えて初めて伝わるが、このことを「発火」と呼ぶ。

上で述べた脳の構造を図 4.2 のような人工的なネットワークに落とし込み、非線形変換を効率よく実現する。「発火」をコントロールするのが、後に述べる活性化関数である。なお、図 4.2 のようなネットワークの最も単純な構成単位をパーセプロトンと呼ぶ。

・活性化関数

(i) ニューロンが離散値の入出力をもつ場合

$H(x)$ を用いたニューロンの数理モデルは「形式ニューロン」と呼ばれている。形式ニューロンを複数組み合わせて作った回路を「パーセプロトン」と呼ぶ。

図 4.2 に似た形式ニューロンを考えると、入力信号が $x_i \quad (i=1,2,3)$ のとき、合成信号は $x' = \sum_{i=1}^{3} w_i x_i$ で、出力は $y = H(x'+b) = H(\sum_{i=1}^{3} w_i x_i + b)$ である。ここで、$x_0 = 1$ と固定されたニューロンを考えることで、バイアスを重みに含められる。$w_0 := b$ とすると、$y = H(\sum_{i=0}^{3} w_i x_i)$ である。これが p105 の説明で $x_0 = 1$ に固定する理由である。

(ii) ニューロンが実数値の入出力をもつ場合（活性化関数として微分可能な増加関数も採用する）

活性化関数は微分可能でかつ微分がゼロになる領域が小さくないと学習が進まない (後に説明がある)。そこで、$H(x)$ よりも適しているのがシグモイド関数。しかし、シグモイド関数も縮尺が変わればステップ関数と同じ問題に直面する。

これらの弱点を克服したのが ReLU であるが、ReLU は発火していない領域で微分がゼロという問題があるため、

さらに改良されたのが、Leaky ReLU。(ハイパーパラメータ: モデルの選択時に決めなければならないパラメーターで、学習アルゴリズムでは一切修正されない。学習されるパラメーターとは区別される。)

ソフトマックス関数はこれまで述べたものと性格の違う活性化関数で分類問題に多用される。

・ソフトマックス関数の（簡単な）説明

(i)2 クラス分類 C_1, C_2

C_1 である確率とそうではない確率の比、$u = \log \frac{P(C_1|x)}{P(C_2|x)}$ を対数オッズ比と呼ぶ。これを $P(C_1|x)$ について解くと、

$$P(C_1|x) = \sigma(u) = \frac{1}{1+e^{-u}} \tag{1}$$

である。これが 2 クラス分類では重要なパラメータである。

(ii) 多クラス分類 $C_1, \ldots, C_K$

(i) の対数オッズ比を多クラス分類に一般化すると、

$$u_k = \log \frac{P(C_k|x)}{P(C_K|x)} \quad (k = 1, \ldots, K) \tag{2}$$

となる。(2) 式から、

$$P(C_K|x)e^{u_k} = P(C_k|x) \tag{3}$$

全確率の法則から、$P(C_K|x) = \frac{1}{\sum_{k=1}^{K} e^{u_k}}$ となるので、これを (3) 式に代入して、

$$P(C_k|x) = \frac{e^{u_k}}{\sum_{k'=1}^{K} e^{u_{k'}}} \tag{4}$$

となる。多クラス分類の分布は対数オッズ比を引数とするソフトマックス関数で表せる。

・普遍性定理

ディープラーニングの最も基礎となる定理。「discriminatory な 活性化関数を組み合わせたネットワークを使えば、どんなに複雑な関数でも表現できる」

活性化関数として、シグモイド関数をより一般化した拡張シグモイド関数 (4-6) を選ぶ。さらに図 4.4 のように右側のニューロンを N 個に増やし、最終的な出力 $f_N(x_0, x_1, \ldots)$ はこれら活性化関数の線形結合で与えられるものとする。つまり、

$$f_N(x_0, x_1, \ldots) = \sum_{i=1}^{N} \alpha_i \zeta(x'_i) \quad (x' = \omega_0 x_0 + \omega_1 x_1 + \cdots)$$
$$f_N(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \zeta(x'_i) \quad (x'_i = b_i + \boldsymbol{\omega_i} \cdot \boldsymbol{x})$$

である。この $f_N(\boldsymbol{x})$ は任意の連続関数 $f(\boldsymbol{x})$ に対応している。十分大きな N を選ぶ、つまり右側のニューロンの数を十分多くすれば $f_N(\boldsymbol{x})$ で $f(\boldsymbol{x})$ をいくらでも正確に近似できるというのがこの定理の主張である。(4-8) の式を参照。