生体情報システム
2009 年 4 月 17 日

例題からの学習：
ニューラルネットワーク（神経回路モデル）入門

- 基本事項：
  - ・神経回路の数理モデル
  - ・二つのダイナミックス（ダイナミックスとは時々刻々変化する様）
    1. ニューロン活動 $\boldsymbol{x}$ のダイナミックス（速い）→ 思考・判断に対応
    2. 結合係数 $\boldsymbol{w}$ のダイナミックス（遅い）→ 学習に対応

- 神経回路の数理モデル

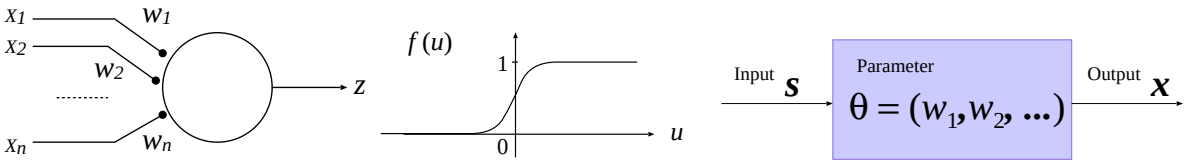

図 1: ニューロンの数理モデル（左）と出力関数（中），数理モデル一般に対する見方（右）

1 個のニューロンの入出力を考えよう（図 1 左）．入力を n 次元ベクトル $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$, 結合係数を $\boldsymbol{w} = (w_1, w_2, \cdots, w_n)$ とすると，出力 z は

$$z = f(\boldsymbol{w} \cdot \boldsymbol{x} - h) \tag{1}$$

と書ける．ここでニューロンの閾値（しきいち）を h，出力関数を f，$\cdot$ を内積とした．f としては例えば $f(u) = \dfrac{1}{1+e^{-u}}$ のシグモイド型関数や，$f(u) = 1$ if $u > 0$, otherwise 0 の階段関数などの単調増加関数がよく使われる．シグモイド型関数の場合，$f(u)$ は $u \to -\infty$ で 0，$u \to \infty$ で 1，$u = 0$ のとき 0.5 となる．入力 $\boldsymbol{x}$ が 2 次元，出力 z が 1 次元の簡単な例で考えよう．ニューロンは $\boldsymbol{x} = (x_1, x_2)$ が入力されると，それを結合係数で指定された重みを付けて受け取り，閾値を引き算した値

$$u = \sum_{i=1}^{2} w_i x_i - h \tag{2}$$

を元に $z = f(u)$ を出力する．u は内部状態と呼ばれる．閾値 h については，数学的に扱う場合には，常に $x_0 = 1$ という信号が入力されており，それを $w_0 = -h$ の重みで受け取っていると考えれば，

$$u = \sum_{i=0}^{2} w_i x_i \tag{3}$$

とすっきり書ける．

- ニューロンの学習

「りんご」と「みかん」から測定されたデータ（2 次元）が，それぞれ 10 個ある．このデータ例を参考に，りんごとみかんを分別できるような機械を学習により作ることを考えよう．データを観測し，りんごと判別する場合は 1，みかんと判別する場合は 0 を出力させたい．2 次元のデータを入力 $\boldsymbol{x} = (x_1, x_2)$ として受取り，正しい答え z を出す機械を作ることが目的である．機械は図 1 に示すニューロンを用いる．以下，20 個の例題（入力）を $\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^{20}$，それぞれの例題に対する望ましい出力を $z^1, z^2, \cdots$ と書く．まず学習の流れをつかんでおこう．

1. 初期設定： 各結合係数に適当な乱数を割り当てる．
2. 例題 $\boldsymbol{x}^\alpha$ を提示し（α は $1, \cdots, 20$ のどれか），出力 z を求める（$\boldsymbol{x}$ のダイナミックス）．
3. z が望ましい出力 z^α とどのくらいズレているか誤差 E を計算する．
4. 誤差が少なくなるよう，結合係数を修正する．これを学習と呼ぶ（$\boldsymbol{w}$ のダイナミックス）．
5. 誤差がなくなるまで 2. から 4. を繰り返す．

– 教師あり学習

初期状態では，機械は何も知らない状態なので，とんちんかんな答え $0, 1$ を出す（これは w_0, w_1, w_2 に適当な乱数が初期値として割り当てられているということ）．この機械に対して，正しい答えを教え，機械のパラメータ $\boldsymbol{w}$ を調整し，正しい答が出力されるようにする．機械に正しい答えを教えて機械の学習をおこなうので，これを「教師あり学習」という．

– 学習アルゴリズム

どのようにパラメータを修正していけば間違いが少なくなるだろうか．パラメータ修正の方法を記述したものを「学習アルゴリズム」という．いま α 番目の例題 $\boldsymbol{x}^\alpha$ が入力されたとき，機械の出力する答えと，望ましい答えの誤差を

$$E^\alpha = \left[z^\alpha - f\left(\sum_{i=0}^{2} w_i x_i^\alpha \right) \right]^2$$

で測ろう（他にも測り方はたくさんある）．例題は全部で 20 個あるので，全体としての誤差を

$$E = \sum_{\alpha=1}^{20} E^\alpha$$

と書く．使用できる例題は固定されているので，この誤差の値 E は現在の機械のパラメータ $\boldsymbol{w}$ に依存して決まる．問題は $\boldsymbol{w} := \boldsymbol{w} + \Delta\boldsymbol{w}$ と少しだけ変えて誤差 E をより小さくすることである．$\boldsymbol{w} = (w_0, w_1, w_2)$ であるから $\boldsymbol{w}$ も $\Delta\boldsymbol{w}$ も 3 次元である．誤差を少なくするにはどの方向に $\Delta\boldsymbol{w}$ を修正すればいいだろうか．それを求めるには最急降下法を用いればよい．具体的には，例えば w_1 については

$$\Delta w_1 \propto -\frac{\partial E}{\partial w_1}$$

に少しだけ変化させればいい（$\propto$ は比例するという意味の記号）．それはなぜか．$\dfrac{\partial E}{\partial w_1}$ を計算したところ，それが正の値だったとしよう．この場合は w_1 を正の方向に動かすと E が大きくなることを意味している．したがって E を減らすためには w_1 を負の方

向 $\left(-\dfrac{\partial E}{\partial w_1}\right)$ に動かす方が局所的にはよいことがわかる.「局所的には」という意味は, E を最も最小にする $\boldsymbol{w}$ の方向へ動かしている保証はないからである．この点は，この手法が「山下り法」とも呼ばれている理由である．同じように，$\dfrac{\partial E}{\partial w_1}$ の値が負だった場合も考えよう．やはり $-\dfrac{\partial E}{\partial w_1}$ の方向に w_1 を動かせば E が小さくなる方向に動く．したがって

$$w_1 := w_1 + \Delta w_1, \quad \Delta w_1 \propto -\frac{\partial E}{\partial w_1}$$

と動かすのがよい．E は具体的には

$$E = \left[z^1 - f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\right]^2 + \left[z^2 - f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\right]^2 + \cdots$$

と書けるので,

$$\Delta w_1 \propto -\left[\frac{\partial E^1}{\partial w_1} + \frac{\partial E^2}{\partial w_1} + \cdots\right]$$

である．見た目ほどこの計算は難しくない．α 番目の例に対する誤差 E^α の w_1 による偏微分 $\dfrac{\partial E^\alpha}{\partial w_1}$ について計算してみよう.

$$\begin{aligned}
\frac{\partial E^\alpha}{\partial w_1} &= 2\left[z^\alpha - f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\right] \frac{\partial}{\partial w_1}\left[z^\alpha - f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\right] \\
&= -2\left[z^\alpha - f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\right] \frac{\partial}{\partial w_1} f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right) \\
&= -2\left[z^\alpha - f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\right] f'\left(\sum_{i=0}^{2} w_i x_i^\alpha\right) \frac{\partial}{\partial w_1}\left(\sum_{i=0}^{2} w_i x_i^\alpha\right) \\
&= -2\left[z^\alpha - f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\right] f'\left(\sum_{i=0}^{2} w_i x_i^\alpha\right) x_1^\alpha
\end{aligned}$$

となる．ニューロンの出力関数 f が $f(u) = u$ という線形関数の場合，$f' = 1$ で

$$\frac{\partial E^\alpha}{\partial w_1} = -2\left[z^\alpha - \sum_{i=0}^{2} w_i x_i^\alpha\right] x_1^\alpha$$

となる．$f(u) = \dfrac{1}{1+e^{-\lambda u}}$ という非線形関数（λ はパラメータ）の場合，驚くべき事に,

$$f'(u) = \frac{\lambda e^{-\lambda u}}{(1+e^{-\lambda u})^2} = \lambda \cdot \frac{1}{1+e^{-\lambda u}} \cdot \frac{e^{-\lambda u}}{(1+e^{-\lambda u})} = \lambda f(u)(1-f(u))$$

となり，$f(u)$ をもとに微分値が計算できる．このことに気づけば,

$$\frac{\partial E^\alpha}{\partial w_1} = -2\lambda\left[z^\alpha - f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\right] f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\left(1 - f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)\right) x_1^\alpha$$

となる．これは仰々しく見えるが，入力信号に対するニューロンの出力 $z = z(\boldsymbol{x}^\alpha) = f\left(\sum_{i=0}^{2} w_i x_i^\alpha\right)$ を一度計算しておけば，

$$\frac{\partial E^\alpha}{\partial w_1} = -2\lambda(z^\alpha - z)z(1-z)x_1^\alpha$$

と計算できる．線形の出力関数を用いた場合と比較し，$z(1-z)$ という項を掛け算する点だけが違う．

$$\Delta w_1 \propto -\left[\frac{\partial E^1}{\partial w_1} + \frac{\partial E^2}{\partial w_1} + \cdots\right]$$

であったので結局，信号 $\boldsymbol{x}^\alpha$ が入力された場合，

$$\Delta w_1 \propto (z^\alpha - z)z(1-z)x_1^\alpha$$

と変更していけばよい．w_0, w_2 についても同じであり，

$$\Delta w_i \propto (z^\alpha - z)z(1-z)x_i^\alpha,\ i = 0, 1, 2$$

となる．出力 z が望ましい出力 z^α に近いほど Δw_i を修正する量が小さいことがわかる．

- 学習に成功する場合，失敗する場合

  上記では 1 個のニューロンにすべての処理をまかせていた（図 2 左）．これには，当然，能力に限界があり，できることとできないことがある．これを演習により，各自で確かめてほしい．学習できた場合も，たまたま偶然成功しただけかもしれないので，いろいろパラメータを変えて試してみよう．

- 学習に失敗する場合の対処の仕方

  実は，判別したいデータが 線形分離（この場合直線で分けられる）されていなければ，2 入力，1 出力 のネットワーク（図 2 左）を使っているかぎり，いくらがんばってもこの問題を解決することはできない．この問題に対する解決策はいくつかある．演習では誤差逆伝搬法（バックプロパゲーション）と呼ばれる手法を試してみよう（プログラムは用意されている）別の方法としては，入力信号の次元を上げる方法がある．例えば図 2 右のようなネットワークで，中間層に素子を 10 個置き，2 次元の入力を，なんらかの変換（例えばランダムな変換）で 10 次元にすることを考える．この場合，素子がたくさん必要になるが，判別したい線形分離されていないもとの 2 次元のデータが中間層の 10 次元空間では線形分離される可能性が高くなる．

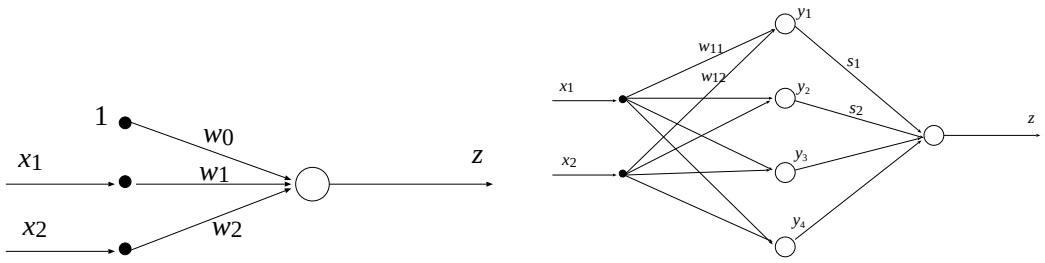

図 2: 1 個のニューロンからなる回路（左）と 3 層回路（右）

- バックプロパゲーションによる学習

  3 層の回路網の場合で，各ニューロンがアナログ型の出力関数 $f(x)$ を持つとしよう．出力 z は 1 次元とする．このとき，入出力関係は

$$z = f(\sum s_i y_i), \quad y_i = f(\sum w_{ij} x_j)$$

  のように書ける．一方，入力信号 $\boldsymbol{x}^\alpha$ に対する出力を $z = z(\boldsymbol{x}^\alpha)$，望ましい出力を z^α としよう．損失関数を

$$E^\alpha = (z^\alpha - z(\boldsymbol{x}^\alpha))^2$$

  とする．可変のパラメータは $\{s_i\}$ と $\{w_{ij}\}$ であり，$z(\boldsymbol{x}^\alpha)$ はこのパラメータに依存して決まる関数である．この問題に，最急降下法を適用してみよう．

$$\frac{\partial E}{\partial s_i} = -2(z^\alpha - z(\boldsymbol{x}^\alpha)) f'(\sum s_j y_j) y_i$$

  先と同様に，$v = \sum s_j y_j$ と書くとこれは

$$\begin{aligned} \frac{\partial E}{\partial s_i} &= -2(z^\alpha - z(\boldsymbol{x}^\alpha)) f'(v) y_i \\ &= -2\lambda(z^\alpha - z(\boldsymbol{x}^\alpha)) f(v)(1 - f(v)) y_i \\ &= r_0 y_i \end{aligned} \tag{4}$$

  となる（ここで $r_0 = -2(z^\alpha - z(\boldsymbol{x}^\alpha)) f'(v)$ とおいた）．損失 E を減らすためには

$$s_i := s_i - \eta \frac{\partial E}{\partial s_i}$$

  とすればいいから，

$$s_i := s_i + 2\eta\lambda(z^\alpha - z(\boldsymbol{x}^\alpha)) f(v)(1 - f(v)) y_i$$

  とすればいい．ここで η は学習定数と呼ばれる定数である．これを r_0 を使って書くと，

$$s_i := s_i - \eta r_0 y_i$$

  となる．中間層の学習 w_{ij} については，

$$\begin{aligned} \frac{\partial E^\alpha}{\partial w_{ij}} &= \frac{\partial E^\alpha}{\partial y_i} \frac{\partial y_i}{\partial w_{ij}} \\ &= -2(z^\alpha - z) f'(v) s_i f'(u_i) x_j \\ &= r_0 s_i f'(u_i) x_j \\ &= r_i x_j \end{aligned}$$

  ここでも $r_i = r_0 s_i f'(u_i)$ と定義した．$\{r_i\}$ は学習信号と呼ばれている．この学習の形は面白い．具体的に w_{42} （2 番目の入力と，4 番目の中間層の素子の間の結合係数）をどう修正したらいいか考えよう．この学習アルゴリズムは x_2 の値に r_4 を掛けた量（の定数倍）変更すればいいと主張している．r_4 というのは $r_0 s_4 f'(u_4)$ であり，最終層の素子の学習信号 r_0 が結合係数 s_4 でもとにもどされた格好をしている．これが由来で，この学習アルゴリズムはバックプロパゲーションと呼ばれている．