数理脳科学
2018 年 4 月 26 日
更新 2018 年 5 月 27 日

課題 1: 教師あり学習

中間発表（30 秒/人）: 5 月 17 日（木）

提出締切 5 月 31 日（木）

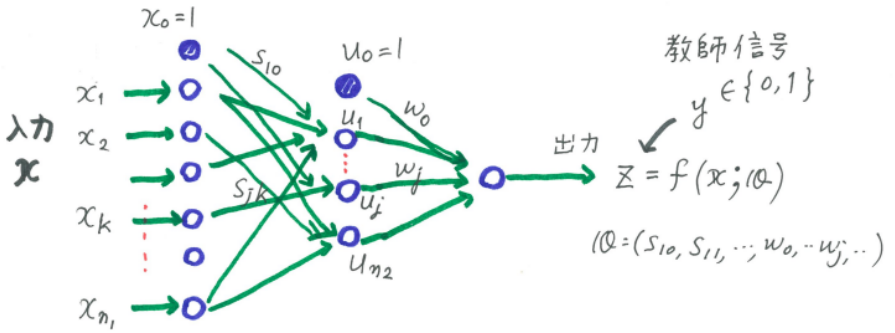

図 1: 3 層神経回路モデル（実質は 2 層）

ニューラルネットには，いくつかの典型的な要素モデルがある．その一つが，パーセプトロンを代表とする教師あり学習である．これは，入力とそれに対する望ましい出力のペアが多数与えられることから,「例からの学習」[1] とよばれている．Deep Learning の問題設定もこれにあたると考えてよい．

ニューラルネットについては，次のようなイメージを持っておけばよいだろう．

1. 活動のダイナミクス．これは「思考」に対応．個々の素子（ニューロン）は，他の素子の活動を重み付きの和で受け取り，自分が興奮するか否かを決定する．

2. 結合のダイナミクス．これは「学習」に対応.「重み」（結合荷重，結合係数）は，活動のダイナミクスとは比較にならないくらい，ゆっくり変化する．学習のダイナミクスとも言う．学習がローカル（局所的）とは，j 番目から i 番目の素子への結合荷重 w_{ij} の変化分が，j 番目の素子の活動と i 番目の素子の活動にのみ依存する学習法則のこと．誤差逆伝搬法は，ローカルな学習とは考えられていないが,「活動」をそれぞれの素子がなんらかの形で持っている「信号」と思えば，誤差逆伝搬法 [1, 2, 3] もローカルな学習と解釈できる．

3. 単純な Yes，No の意思決定をする素子（人間）が多数存在する回路（世界・社会）．本来は，大量の素子が並列に動作する．結合係数（人と人との信頼関係）は時々刻々ゆっくり変化して形成される，

1　例からの学習（教師あり学習，パーセプトロンなど）[1, 2, 3]

問題設定

いま入力とそれに対する望ましい出力（教師信号）のペア $(\boldsymbol{x}^1, \boldsymbol{y}^1), (\boldsymbol{x}^2, \boldsymbol{y}^2), \ldots (\boldsymbol{x}^m, \boldsymbol{y}^m)$ が m 個，与えられている．各入出力ペア $(\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha), \alpha = 1, \cdots, m$ は，ある未知の確率分布 $\Pr(\boldsymbol{x}, \boldsymbol{y})$ にしたがっているとする．目的は，$\boldsymbol{x}^\alpha$ を入力すると，$\boldsymbol{y}^\alpha$ を出力するような回路（ネットワーク）$f(\boldsymbol{x}; \boldsymbol{\theta})$ を作ることである．ただし，与えられた例題に対してだけではなく，未知の入力 $\boldsymbol{x}$ に対しても，正しい答え $\boldsymbol{y}$ を出力できる回路を作ることである．

- given $\{(\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha)\}, \alpha = 1, \cdots, m$
- find $\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathrm{E}\Big[\frac{1}{2}\Big(f(\boldsymbol{x}^\alpha; \boldsymbol{\theta}) - \boldsymbol{y}^\alpha\Big)^2\Big]$

回路の性質は結合係数の値で決まる．すべての結合係数をまとめてパラメータ $\boldsymbol{\theta} = (w_{12}, w_{13}, \cdots)$ で表そう．回路を設計することは，関数 $f(\boldsymbol{x}; \boldsymbol{\theta})$ を設計することとも言える．したがって回路 $\approx$ 関数である．学習とは，初期値として与えられた不完全なパラメータ $\boldsymbol{\theta}$ の値を，例題（訓練データ，training set）$\mathcal{D} = \{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots (\boldsymbol{x}^m, \boldsymbol{y}^m)\}$ に合うように変えていくことである．学習は，パラメータ推定ともいう．パラメータ $\boldsymbol{\theta}$ は，例題 $\mathcal{D}$ に依存して決まるので，回路 $f(\boldsymbol{x}; \boldsymbol{\theta})$ のことを $f(\boldsymbol{x}; \mathcal{D})$ とも記述する．

最近出版された機械学習の書籍では，Tensorflow, Chainer など機械学習のフレームワークを使い，コンピュータシミュレーションをする方法が記述されている．そこを入口にしても悪くはないが，まず小さい問題を解く回路のコンピュータプログラムをゼロから書いてみよう．大規模な回路がどのように動いているか，動作原理の理解が深まるはずである．

1.1　教師あり学習の原理

以下，図 1 に記述した記号を用いて説明する．入力層に信号 $\boldsymbol{x}$ が与えられると，第 2 層の j 番目の出力 u_j が

$$u_j \quad = \quad g\left(\sum_{k=1}^{n_1} s_{jk} x_k - \theta_j\right) \tag{1}$$

と計算できる（$j = 1, \cdots, n_2$）．ここで，s_{jk} は，第 1 層 k 番目の素子から第 2 層 j 番目の素子への結合の重み，θ_j は第 2 層 j 番目の素子のしきい値である．$g(u)$ は，ニューロンの出力関数で

$$g(u) \quad = \quad \frac{1}{1 + \mathrm{e}^{-u}} \tag{2}$$

の場合を考える．ここで $x_0 = 1$ という常に興奮する素子を考えると

$$u_j \quad = \quad g\left(\sum_{k=0}^{n_1} s_{jk} x_k\right) \tag{3}$$

と簡潔に記述できる．$s_{j0} = -\theta_j$ のことを素子のバイアスとよぶ．第 2 層の各素子の出力が決まると，出力層の素子の出力が

$$z \quad = \quad g\left(\sum_{j=0}^{n_2} w_j u_j\right) \tag{4}$$

$$= \quad g\left(\sum_{j=0}^{n_2} w_j g\left(\sum_{k=0}^{n_1} s_{jk} x_k\right)\right) \tag{5}$$

と計算できる（w_0 はバイアス項）．ここで，w_j は，第 2 層 j 番目の素子から出力層の素子への結合の重みである．

例題（訓練データ，training set）$\mathcal{D} = \{(\boldsymbol{x}^1, y^1), \ldots (\boldsymbol{x}^m, y^m)\}$ が m 個，与えられているとする．入力 $\boldsymbol{x}^\alpha$ に対する望ましい出力が $y^\alpha \in \{0, 1\}$ である（現実には $\lim_{u\to\infty} g(u) = 1$ なので，$y^\alpha \in \{0.1, 0.9\}$ としたほうがいいかもしれない）．いま，目的は，ある入力 $\boldsymbol{x}$ が与えられた時,

$$E \quad = \quad \frac{1}{2}\sum_{\alpha=1}^{m} (z(\boldsymbol{x}^\alpha) - y^\alpha)^2 \tag{6}$$

を最小化することだと考えよう，E を損失関数という．もちろん別の損失関数を定義し，それを最小化してもよい.

パラメータ $\{s_{jk}\}, \{w_j\}$ は，初期値として，でたらめな値が設定されている．したがって，たとえば信号 $\boldsymbol{x}^5$ を回路に入力しても，出力 $z = z(\boldsymbol{x}^5)$ として y^5 と同じ値が出力されることは，まずない．回路が望ましい出力により近い値を出すように，各パラメータを，少し大きくするか，小さくするか，変えればよい．どちらに動かせばよいかは，$\dfrac{\partial E}{\partial w_j}$ を計算してみるとよい．$\dfrac{\partial E}{\partial w_j}$ が正の値の場合，w_j を大きくするると，損失 E が大きくなる．E は小さくしたいので,

$$w_j \quad := \quad w_j + \Delta w_j = w_j - \mu \frac{\partial E}{\partial w_{ij}} \tag{7}$$

と更新すればよいだろう（大きくするか小さくするかが問題で，動かす大きさを $\dfrac{\partial E}{\partial w_j}$ に比例させるのがいいか，ということは議論の余地がある）．ここで，μ は $\mu = 0.05$ などの，小さい正の値で，学習係数とよばれている．同様に，$\dfrac{\partial E}{\partial w_j}$ が負の値の場合，w_j を大きくすると E が小さくなることを意味している．損失 E は小さくしたいのでやはり

$$w_j \quad := \quad w_j - \mu \frac{\partial E}{\partial w_{ij}} \tag{8}$$

と，更新すれば，損失 E は小さくなる．具体的に，偏微分の項を計算しよう．

$$E = \frac{1}{2}\sum_{\alpha=1}^{m}(z(\boldsymbol{x}^{\alpha})-y^{\alpha})^2 \tag{9}$$

$$z = g\left(\sum_{j=0}^{n_2} w_j u_j(\boldsymbol{x}^{\alpha})\right) \tag{10}$$

であるので，

$$\Delta w_j = -\mu\frac{\partial E}{\partial w_j} = -\frac{\mu}{m}\sum_{\alpha=1}^{m}\left\{\left(z(\boldsymbol{x}^{\alpha})-y^{\alpha}\right)g'\left(\sum_{j=0}^{n_2} w_j u_j(\boldsymbol{x}^{\alpha})\right)u_j(\boldsymbol{x}^{\alpha})\right\} \tag{11}$$

と書ける．実は，この更新式を

$$\Delta w_j = \mu r u_j \tag{12}$$

と，すっきりと解釈する見方がある．ここで r は出力素子が持っているなにかである（あとで学習信号とよぶ）．この更新式は出力層の素子と中間層の j 番目の素子との結合を，r との掛け算に比例して更新，という形になっている．こういう形を Hebb 学習という．信号 $\boldsymbol{x}^5$ が回路に入力された場合で学習信号 r を具体的に考えよう．

1. 信号 $\boldsymbol{x}^5$ を回路に入力
2. 出力層の素子の内部状態 $v(\boldsymbol{x}^5)=\sum_{j=0}^{n_2} w_j u_j(\boldsymbol{x}^5)$ を計算.
3. 出力 $z=g(v(\boldsymbol{x}^5))$ と望ましい出力との誤差 $d=z(\boldsymbol{x}^{\alpha})-y^{\alpha}$ を計算.
4. $r=-dg'(v(\boldsymbol{x}^5))$ と定義すると（これは出力素子の学習信号），入力 $\boldsymbol{x}^5$ に対してだけの変化分は $\Delta w_j=\mu r u_j(\boldsymbol{x}^5)$ と書ける．$g'(v)$ は素子の出力関数 $g(v)$ の微分で，

$$g(v) = \frac{1}{1+\mathrm{e}^{-v/T}} \tag{13}$$

の場合（T はパラメータ）

$$g'(v) = \frac{1}{T}g(v)(1-g(v)) \tag{14}$$

であり（実際に微分し確認してみよ），$T=1$ の場合，出力値 $z=g(v)$ であるので，$g'=z(1-z)$ と簡単に計算できる．したがって

$$r = -(z-y)z(1-z) = (y-z)z(1-z) \tag{15}$$

が学習信号である．

考察：この学習信号の式は面白いので，少しの間，にらめっこして式の意味を考えてみよう．自分が出力素子（ボス）になった気分になればよくわかる．$\Delta w_j = \mu r u_j$ であるので，r の正負が重要になる．$z(1-z)$ の項は常に正（もしくは 0）なので，とりあえずは気にしないでよいだろう．いま回路に入力が与えられ，順に計算し，出力を計算（ボスが意思決定）したとする．回路の出力 z と望ましい出力 y が一致している場合，$r = 0$ となる．更新式は，結合係数の値は更新しなくて良い，ということを意味している．一方 $y = 1$ を出力すべきなのに $z = 0$ だった場合は，$\Delta w_j = \mu r u_j$ であるので，出力が正の素子（部下）とは結合を強め（もっと君の意見を尊重すべきだった），出力が負の素子とは（おまえのせいで間違えた！，ということで）結合を弱めることになる．逆に，$y = 0$ を出力すべきなのに $z = 1$ を出力してしまった場合は，出力が正の素子とは結合弱め（おまえのせいで間違えた！），出力が負の素子とは結合を強めることになる．r がの正（負）であるということは，いまよりも，もっと正（負）の値を出力せよ，という教師信号であると思ってよい．

ここまで出力素子が $0, 1$ の間の値をとる場合を考えてきたが，$-1, 1$ の間をとるモデル

$$g(v) \quad = \quad \tanh(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} \tag{16}$$

を考える場合もよくある [3]．ここで整理しておこう．これを微分すると

$$g'(v) \quad = \quad (1 - g(v)^2) = (1 - z^2) \tag{17}$$

と簡単に書ける（やはり実際に微分し確認しておくこと）．$\{0, 1\}$ モデルの場合 $z(1-z)$ であった項が，$(1-z^2)$ となるが，この項も常に 0 以上 1 以下の値をとる（負にはならない）ことは同じである．

演習課題：Δs_{jk} については，自分で計算してみよ（s_{jk} が関係している項に○をつけると計算しやすい）．

$$E = \frac{1}{2}(z - y)^2$$
$$z = g(v)$$
$$v = \sum w_j u_j$$
$$u_j = g(a_j)$$
$$a_j = \sum_k s_{jk} x_k$$

$$\frac{\partial E}{\partial s_{jk}} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial u_j} \cdot \frac{\partial u_j}{\partial a_j} \cdot \frac{\partial a_j}{\partial s_{jk}}$$

出力層の素子が複数個になった場合について，バックプロパゲーションの学習式を書いておこう．Hebb 学習を一般化したものであると考えれば，わかりやすい（図 2）．

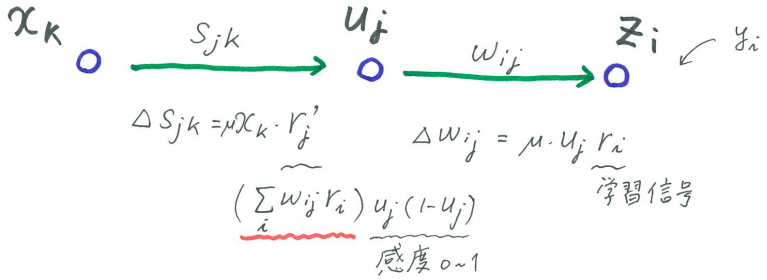

図 2: 入力側の素子の出力と出力側の素子の学習信号との Hebb 学習

第 2 層の j 番目の素子と，第 3 層の i 番目の素子間の結合の学習は

$$\Delta w_{ij} = -\mu \frac{\partial E}{\partial w_{ij}} = -\mu \Big(z_i(\boldsymbol{x}) - y_i(\boldsymbol{x}) \Big) g' \Big(\sum_{j=0}^{n_2} w_{ij} u_j(\boldsymbol{x}) \Big) u_j(\boldsymbol{x}) \tag{18}$$

$$= -\mu (z_i - y_i) z_i (1 - z_i) u_j \tag{19}$$

$$= \mu r_i u_j \tag{20}$$

と書けるので，これは第 3 層 i 番目の素子の学習信号 r_i と第 2 層 j 番目の素子の出力との Hebb 学習である（$i = 1, \cdots, n_3, j = 0, \cdots, n_2$）．第 1 層 k 番目の素子と第 2 層 j 番目の素子の結合係数を s_{jk} とすると

$$\Delta s_{jk} = -\mu \frac{\partial E}{\partial s_{jk}} = \mu \left(\sum_{i=1}^{n_3} w_{ij} r_i \right) g' \left(\sum_{k=0}^{n_1} s_{jk} x_k \right) x_k \tag{21}$$

$$= \mu \left(\sum_{i=1}^{n_3} w_{ij} r_i \right) u_j (1 - u_j) x_k \tag{22}$$

$$= \mu r_j' x_k \tag{23}$$

と，やはり Hebb 学習の形で記述できる（$k = 0, \cdots, n_1, j = 1, \cdots, n_2$）．中間層の j 番目の素子の学習信号 r_j' が，出力層の学習信号 r_i の重み付けの和 $\sum_{i=1}^{n_3} w_{ij} r_i$ で計算できる点が，バックプロパゲーションという名前の由来である．

ここまで 3 層からなる回路について説明してきたが，層を積み重ねた 100 層回路の場合も，この学習の仕方は変わらない．

ここまで，損失関数

$$E = \frac{1}{2} \sum_{\alpha=1}^{m} (z(\boldsymbol{x}^\alpha) - y^\alpha)^2 \tag{24}$$

を最小化することを考えてきた．ここで，別の損失関数 E を考えてみる．最終層の素子の出力 $z = f(\boldsymbol{\theta}|\boldsymbol{x})$ の値は 0 から 1 の値をとる．この値を用いると，「入力 $\boldsymbol{x}$ が与えられたときに，

回路が正解を出力する条件付き確率」は $z^y(1-z)^{1-y}$ と書ける．したがって，学習を尤度

$$L \;=\; \prod_{\alpha=1}^{m} z(\boldsymbol{x}^\alpha)^{y^\alpha}(1-z(\boldsymbol{x}^\alpha))^{1-y^\alpha} \tag{25}$$

を最大化にするパラメータを求める問題としても定式化できる．先の 2 乗誤差最小化の学習と比較するため，最大化ではなく最小化として定式化すると

$$E \;=\; -\sum_{\alpha=1}^{m}\Big\{ y^\alpha \log z(\boldsymbol{x}^\alpha) + (1-y^\alpha)\log\big(1-z(\boldsymbol{x}^\alpha)\big)\Big\} \tag{26}$$

となる（対数関数は単調増加関数なので，対数をとっても最小化すること自体は変わらない）．これを各結合係数で偏微分した値を計算し，2 乗誤差誤差最小の場合と結果を比較してみよう．例題のインデックス α は，手計算に余計なので，記述を省略し

$$E \;=\; -\Big\{ y\log z + (1-y)\log(1-z)\Big\} \tag{27}$$

の偏微分を考えよう（各自，一度手を動かして計算すること）．

$$\Delta w_j \;=\; -\mu\frac{\partial E}{\partial w_j} = \mu\Big\{ y\frac{1}{z}z(1-z)u_j + (1-y)(-1)\frac{1}{1-z}z(1-z)u_j\Big\} \tag{28}$$

$$\;=\; \mu\Big\{ y(1-z)u_j - (1-y)zu_j\Big\} = \mu\Big\{ y(1-z) - (1-y)z\Big\}u_j \tag{29}$$

$$\;=\; \mu(y-z)u_j = \mu r u_j \tag{30}$$

と，2 乗誤差最小の場合と比べ，最終層の素子の学習信号 r に感度項 $z(1-z)$ がなくなったものになっている．このように簡潔に書けるのは，出力関数としてシグモイド関数 $g(u) = 1/(1+\exp(-u))$ を用いたおかげである．この E は，交差エントロピー誤差関数とよばれている．昔（25 年前のニューロブーム）のときは，そういう言葉はなかった．単なる尤度最大化である．

2　補遺（ほい）

1. 学習誤差，汎化誤差

   与えられた例題に対して，回路の出力と望ましい出力の誤差を学習誤差とよぶ．例題を丸暗記していれば，これはゼロにすることができる．目的は，まだ経験していない例題に対して，正しい答えを推論することである．未知の入力（学習に用いていないデータ）に対しての，回路の出力と望ましい出力の誤差を汎化誤差（generalization error）とよぶ．つまり目的は汎化誤差をゼロにすることである．しかも，少ない数の例題が与えられただけで，本質を見抜く力を身につけることができる知性・知能のある機械を作りたい．

2. Bias/Variance ジレンマ

   これが本課題のメインテーマ．二乗誤差は，Bias 項と Variance 項に分解できる（文献 [3] p.10）．$k-$NN やニューラルネットの実験で確かめ，考察する．

3. 回帰（regression）

   これは $\mathrm{E}[y|\boldsymbol{x}]$ のこと．二乗誤差を最小にするという意味では，これよりよい推定量はない（証明は文献 [3] p.4）．もちろん，データからは $\mathrm{E}[y|\boldsymbol{x}]$ を推定することはできるが，本当のところは神様しかわからない（今回の例題では分かっているが，推論にはもちろん使わない）．

4. Nearest-Neighbor Regression

   $k-$Nearest-Neighbor とよばれる．k には 1 とか，5 などの整数が入る．この方法は単純で分かりやすい．まず例題を全部記憶しておく．未知の例題 $\boldsymbol{x}$ が与えられたときには，与えられている例題の中に含まれている，$\boldsymbol{x}$ に近いものから順に k 個の例題を取り出し推論に使う．それらの例題の望ましい出力の k 個の平均を，推定値として用いればよい．極端な場合を考えれば分かりやすい．$k = 1$ では，入力にもっとも近い例題と同じ出力を，k が全例題数であれば，全例題の望ましい出力の平均，つまり，どんな入力が与えられても同じ出力を，推定値として用いる．

5. Parzen-Window Regression

   例題にガウス分布のような重みを付けて考える．考え方は $k-$NN と同じ（といえば，語弊があるかもしれない）．

6. 具体的な実験の仕方（評価の方法）

   ここでは，$\mathrm{E}[y|\boldsymbol{x}]$ がわかっている場合で実験するので，学習誤差について，横軸に時間 t，縦軸に誤差 $E(t)$ をプロットした図で評価すれば良い

7. ニューロンの出力を決める関数について．

   $g(u) = 1/\{1 + \exp(-u)\}$ など（活動度関数，活性化関数とよばれたりする）は，0 から 1 の値を出力する．$-1, 1$ の値を出力させたければ，$g(u) = \tanh(u)$ を用いればよい．出力層の素子はともかく，中間層の素子には，$g(u) = \tanh(u)$ を用いたほうが，学習が速く進む場合が多いようだ（←理論的根拠を調べる必要あり）．出力値の望ましい信号も $0, 1$ より，$y = \pm 1$ のときのほうが学習がうまくいくという話もある．どちらの場合も，望ましい出力を $0.1, 0.9$ とか，$-0.9, 0.9$ とすると，学習が速く進むは

ずである．$g(u) = 1/\{1 + \exp(-u)\}$ では，実際には 0, 1 を出力できないので，結合係数の値が発散していく可能性がある．

8. ReLU

   最近，よく用いられる活性化関数．どうしてよく用いられるかは，また別の機会に解説．

9. 結合係数の初期値について

   初期値が，以前に考えられていたよりも，重要であることがわかっている．平均 0，分散 0.1 の正規分布にしたがう乱数を用いるなど，かなり小さい値をもつのがよいようだ．あまりに小さすぎると，学習に時間がかかるようではある．学習係数との兼ね合いなど，職人芸が必要．

10. 最適化（E の最小化）の方法

    これも職人芸が必要．おおくのフレームワークでは，最適化のオプション，たとえば adam，を選べばよい．フレームワークを使わず，自分で，一度書いてみること．

11. Tensorflow, Chainer などの利用

    （今後，説明する予定）

3　コンピュータ実験の方法

基本的な手順は，次のようになる．

1. 例題データの作成
2. 結合係数の初期値を適当な乱数に設定
3. 回路に入力を与え，順に素子の出力を計算し，最終出力を計算．
4. 結合係数の値を更新
5. 学習がある程度完了するまで，2. に戻って繰り返す．学習が進んでいるかどうかは，E の値の時間変化をみるとよい．振動している場合，学習係数が大きすぎることに原因があると考えよう．

100 台の回路を学習した結果が必要な場合，あらかじめ，1 台づつの学習結果（各場所への入力に対する出力）をファイルに保存しておく．そのあと，作成した 100 ファイルを読み込み，結果を表示するプログラムを別に書いたほうが，一つのプログラムで 100 台を学習するコードを書くより，効率が良いと思われる．

4　課題

課題は，著名な論文 [1, 3] に掲載されている結果（図）の再現を試み，その結果を『考察』することとする．本質的でない点まで再現する必要はない.「系統的に」解析する，とはどういうことかを，これらの論文から感じとって欲しい．なにを考察するかは具体的には指定しないが，以下，考察価値のある項目の例をいくつか示す.

どの課題も出力層は 1 個の出力素子から構成される．入力層の素子数 n_1 や，中間層（第 2 層，隠れ層）の素子数 n_2 は，課題により異なる.

期日までにできなかった場合，できたところまでを提出すればよい．pdf ファイルをメールに添付して送ればよい（宛先は `date@cs.miyazaki-u.ac.jp` ．紙での提出でもよい）．件名は「数理脳科学レポート課題の提出」，ファイル名は 「`date20180531bpnn.pdf`」（名前 + 日付 + bpnn）などとしてもらえると助かる.

1. XOR ゲートの実現 （入力 2 次元，$n_1 = 2, n_2 = 2, 3, \cdots$）
2. mirror symmetry detection（$n_1 = 6$.　文献 [1] の Fig. 1）
   - 論文通りの結果が得られない可能性が高い．それはどこに原因があるか．隠れ層の素子数 2 で本当にできるのか．考察する価値あり.
   - 回路がすべての例題に正解できた場合，その回路の構造を考察する．論文に書かれている構造とは別の解決方法がいくつもあるかもしれない.
   - 64 個すべての例題を学習に用いなくても学習できるはずである（人間の場合 5 例くらいをじっくり見れば，規則性はわかるのではないか）．どの少数の例を用いるか，何例くらい用いるか.
   - 結合係数の初期値の分布.
   - バッチ学習（64 例，提示したあとで，結合係数を平均の方向に変える）と逐次学習（一例一例提示するごとに，結合係数を変える）での実験結果の違い.
3. サインカーブで区切られた領域の分類（文献 [3] の Figure 3）
   (a) Fig. 3 (sinusoid-within-rectangle problem, SWR)
   (b) Fig. 4 (k-nearest-neighbor regression, $k = 1, \cdots, 10$, SWR)
   (c) Fig. 5 (1 and 2-nearest-neighbor regression, SWR)
   (d) Fig. 6 (average of 100 5-nearest-neighbor machines, SWR)
   (e) Fig. 7 (Bias and variance of 1 and 10-nearest-neighbor estimators, SWR)

(f) Fig. 8 (Bias and variance of feedforward neural networks, #(hidden units)=$1, \cdots, 15$, SWR)

(g) Fig. 9 (two examples of the output of feedforward neural networks, #(hidden units)=5, SWR)

参考文献

[1] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning representations by back-propagating errors," Nature, vol.323, pp.533–536, 1986.

[2] 甘利俊一, "ニューロコンピューティングの数学的基礎," コンピュートロール, no.24, pp.2–14, 1988.

[3] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," Neural Computation, vol.4, pp.1–58, 1992.

学籍番号：＿＿＿＿ ＿＿＿＿

2018 年 4 月 26 日
数理脳科学　　名前：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿　得点：＿＿＿＿

小テスト 【バックプロパゲーション】

1. ニューロンの出力関数が

$$f(u) = \frac{1}{1+\mathrm{e}^{-u/T}} \tag{31}$$

の場合を考える (T は正の定数).

$$f'(u) = \frac{1}{T} f(u)(1-f(u)) \tag{32}$$

であることを以下の手順で確かめよ.

(a) $1-f(u)$ を求めよ.

(b) $f'(u)$ を求めよ. 微分の公式：$\left\{\frac{1}{g(u)}\right\}' = -\frac{g'(u)}{\{g(u)\}^2}$

学籍番号：＿＿＿＿ ＿＿＿＿

2018 年 4 月 26 日

数理脳科学　　名前：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿　　得点：＿＿＿＿

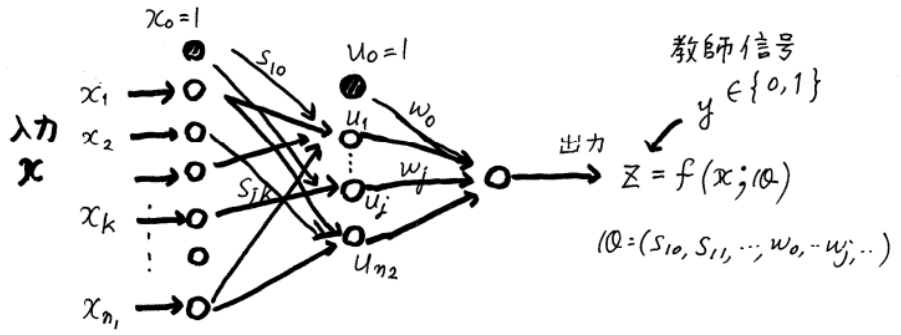

図 3: 3 層神経回路モデル（実質は 2 層）

2. $\boldsymbol{x}$ が入力されたときの $\Delta s_{jk} = -\mu \dfrac{\partial E}{\partial s_{jk}}$ を計算せよ．$E = \dfrac{1}{2}\Big(z(\boldsymbol{x}) - y\Big)^2$ とする．
以下，余力がある場合： 回路の出力 $z = f(\boldsymbol{\theta}|\boldsymbol{x})$ を「入力 $\boldsymbol{x}$ に対し回路が正解を出力する確率」と考え，学習の目的を尤度 $z^y(1-z)^{1-y}$ を最大化することと定式化できる．最小化する関数を $E = -\Big\{ y \log z(\boldsymbol{x}) + (1-y)\log\big(1 - z(\boldsymbol{x})\big) \Big\}$ （対数尤度の符号を反転したもの）と定義した場合についても，各結合係数で偏微分した値を計算し，2 乗誤差誤差最小の場合と結果を比較してみよ．

学籍番号：＿＿＿＿ ＿＿＿＿

2018 年 4 月 26 日
数理脳科学　　名前：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿　得点：＿＿＿＿

3. $E\left[\left(y - f(\boldsymbol{x};\boldsymbol{\theta})\right)^2 \middle| \boldsymbol{x}\right] \geqq E\left[\left(y - E[y|\boldsymbol{x}]\right)^2 \middle| \boldsymbol{x}\right]$ を証明せよ.

ここで，$f(\boldsymbol{x})$ は回路の出力，y は望ましい出力（一次元，$\boldsymbol{y} = y$），E は入力 $\boldsymbol{x}$ に対する期待値．この不等式の意味：パラメータ $\boldsymbol{\theta}$ を調節して，どんなによい回路 $f(\boldsymbol{x};\boldsymbol{\theta})$ を作れたとしても，$E[y|\boldsymbol{x}]$ より適切な回路は作れない（二乗誤差を最小にするという意味で．Among all functions of $\boldsymbol{x}$, the regression is the *best* predictor of y given $\boldsymbol{x}$, in the mean-squared-error sense. ）．※ $E[y|\boldsymbol{x}]$ は $\boldsymbol{x}$ の決定的な関数，回帰（regression）という.

学籍番号：＿＿＿＿＿ ＿＿＿＿＿

2018 年 4 月 26 日
数理脳科学　　名前：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿　得点：＿＿＿＿＿

4. $E_{\mathcal{D}}\left[\left(f(\boldsymbol{x};\mathcal{D})-E[y|\boldsymbol{x}]\right)^2\right] = \left(E_{\mathcal{D}}\left[f(\boldsymbol{x};\mathcal{D})\right]-E[y|\boldsymbol{x}]\right)^2+E_{\mathcal{D}}\left[\left(f(\boldsymbol{x};\mathcal{D})-E_{\mathcal{D}}\left[f(\boldsymbol{x};\mathcal{D})\right]\right)^2\right]$ を証明せよ．右辺第一項をバイアス項，第二項をバリアンス項という．$E_{\mathcal{D}}$ は，訓練データ集合 $\mathcal{D}$ に関して期待値をとる操作.