SAPT
第5回知能化分科会

一色 浩
（SAPT知能化分科会長）

2018.06.16

SAPT電子会議室C

目　　次

1. 瀧 雅人著「これならわかる深層学習」

2. 講義担当者のニューロとの関わり
3. 飲み会の代わり…自由討論

瀧 雅人著「これならわかる深層学習」

2　機械学習と深層学習

2.4　機械学習の基礎

2.4.1　教師あり学習

深層学習の基礎になるのは教師あり学習である.

学習を実現するのは，学習アルゴリズムである.

入力データ x に対する出力 y が学習により望ましい出力をするようなものを学習機械という.

具体的には，繰り返し学習により，入力データ x に対する出力 y と教師データの差を小さくして行く.

統計学的には，説明変数 x と目標変数 y との関係を学習させて，目標変数の適切な推定量 $\hat{y}$ を求めること

学習には2つのタイプがある.

（1） 質的変数 … コインの裏と表 … 分類

（2） 量的変数 … 連続変数 …… 回帰

2.4.2 最小二乗法による線形回帰

出力 y を入力 x の関数として，その関数を関数補間（あてはめ）で推定するもの.

$$\hat{y} = f(x; w)$$

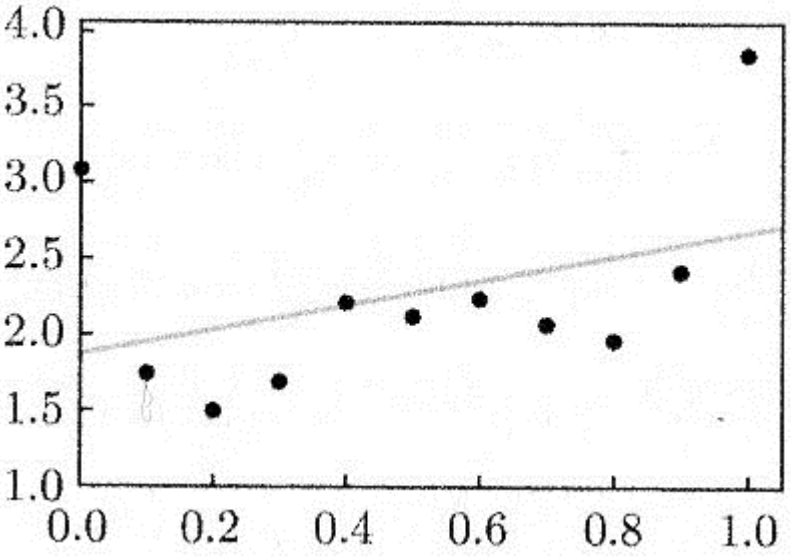
w はパラメータ．観測データにはエラーが ϵ があるので

$$y = f(x; w) + \varepsilon \qquad (2.30)$$

f がパラメータ w の1次関数であると仮定する線形回帰モデルでは

$$f(x,w)=\sum_j w_j h_j(x) \qquad (2.31)$$

学習データ: $(x_n, w_n), \quad n=1,2,\cdots,N$



学習データ

誤差関数: $E_D(w)=\frac{1}{N}\sum_{n=1}^{N}(\hat{y}(x_n;w)-y_n)^2$ (2.33)

式(2.33)を最小にする方法を**最小二乗法**と呼ぶ.

誤差を最小にする学習をする:

$$w^* = \arg\min_w E_D(w) \qquad (2.34)$$

注意: 機械学習の本来の目的は，訓練データ以外のデータも含めて，ありとあらゆるデータに対してよい推定をするモデルをつくること. 未知のデータに対してまでよく働く状況が実現することを汎化（generalizationn）という. この場合の誤差を汎化誤差という:

$$E_{gen.}(w) = \mathrm{E}_{(x,y)\sim P_{data}}\left[(\hat{y}(x_n;w) - y_n)^2\right] \qquad (2.35)$$

式（2.33）のような訓練データの平均で作った訓練誤差を汎化誤差の近似とする. それで多くの成功が得られた.

2.4.3 線形回帰の確率的アプローチ

関数的ではなくて，確率的なアプローチで回帰分析を定式化する.

説明変数 x も目標変数 y も確率変数と考え，**条件付き分布 *P*(y|x) をモデル化することで推定量 $\hat{y}$ を作るアプローチを考える.** この条件付き分布は，説明変数がどのような値を取り易いかを表す．誤差関数も確率変数：

$$E(\hat{\mathrm{y}}(\mathrm{x})-\mathrm{y}) = \tfrac{1}{2}(\hat{\mathrm{y}}(\mathrm{x})-\mathrm{y})^2 \qquad (2.39)$$

パーフォーマンスを測る数値（期待誤差という）：

$$\mathrm{E}_{P_{data}}\left[E(\hat{\mathrm{y}}(\mathrm{x})-\mathrm{y})\right] = \sum_x \sum_y \tfrac{1}{2}(\hat{\mathrm{y}}(\mathrm{x})-\mathrm{y})^2 P_{data}(x,y) \qquad (2.40)$$

期待誤差を最小化する推定量 $\hat{y}$ は，ある説明変数 x を固定して変分を取ると

$$0=\delta \mathrm{E}_{P_{data}}\left[E(\hat{\mathrm{y}}(\mathrm{x})-\mathrm{y})\right]=\sum_{y}(\hat{\mathrm{y}}(\mathrm{x})-\mathrm{y})P_{data}(x,y) \quad (2.41)$$

この式から

$$0=\hat{\mathrm{y}}(\mathrm{x})P_{data}(x)-\sum_{y}\mathrm{y}P_{data}(y\mid x)P_{data}(x) \quad (2.42)$$

が得られるので

$$\hat{\mathrm{y}}(\mathrm{x})=\sum_{y}\mathrm{y}P_{data}(y\mid x)=\mathrm{E}_{P_{data}(y|x)}[y\mid x] \quad (2.43)$$

y 予測値としてもっとも良いのは，生成分布に関する期待値である.

(1) 生成モデル（generative model）

データの背後にある生成メカニズムを直接，同時分布P(x,y)としてモデル化する．つまりP(x,y)を推定した後に，$P(x)=\sum_y P(x,y)$ とベイズの公式：

$$P(y\,|\,x)=P(x,y)/P(x) \qquad (2.44)$$

で $P(y|x)$ を計算し，それより期待値（2.43）を計算．

(2) 識別モデル（discriminative model）

生成モデルは間接的に$P(y|x)$ を計算するので，精度が落ちる可能性がある．そこで$P(y|x)$ を直接モデル化し，データからこの値を推定することで，期待値（2.43）を計算する．クラス分類の呼び方にならって，識別モデルと呼ぶ．

2.4.4 最小二乗法と最尤法

平均二乗誤差の確率論的な由来

最小自乗法の出発点： $y = f(x; w) + \varepsilon$ (2.30)

ϵ が平均0，分散σ^2のガウス分布に従うと仮定すると，y 自体が平均 $\hat{y}(x; w) = f(x; w)$ ，分散σ^2のガウス分布に従う．

$$\varepsilon \sim N(0, \sigma^2) \quad \rightarrow \quad y \sim P(y \mid x; w) = N(\hat{y}(x; w), \sigma^2) \quad (2.45)$$

最尤法を用いれば尤度関数は $L(w) = \prod_n P(y_n \mid x_n; w)$ なので，対数尤度は

$$\log P(y_n \mid x_n; w) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (\hat{y}(x_n; w) - y_n)^2 + \text{const.} \quad (2.46)$$

となる．対数尤度最大は $\frac{1}{2\sigma^2} \sum_{n=1}^{N} (\hat{y}(x_n; w) - y_n)^2$ が最小．

2.4.5 過適合と汎化

図2.3に示されるような1次元データに多項式モデルを当てはめる.

$$\hat{y} = \sum_{j=0}^{M} w_j x^j \tag{2.47}$$

$M = 1$ に対する図2.3(a)の場合には，データの豊かさに対して，モデルの自由度が小さ過ぎる.

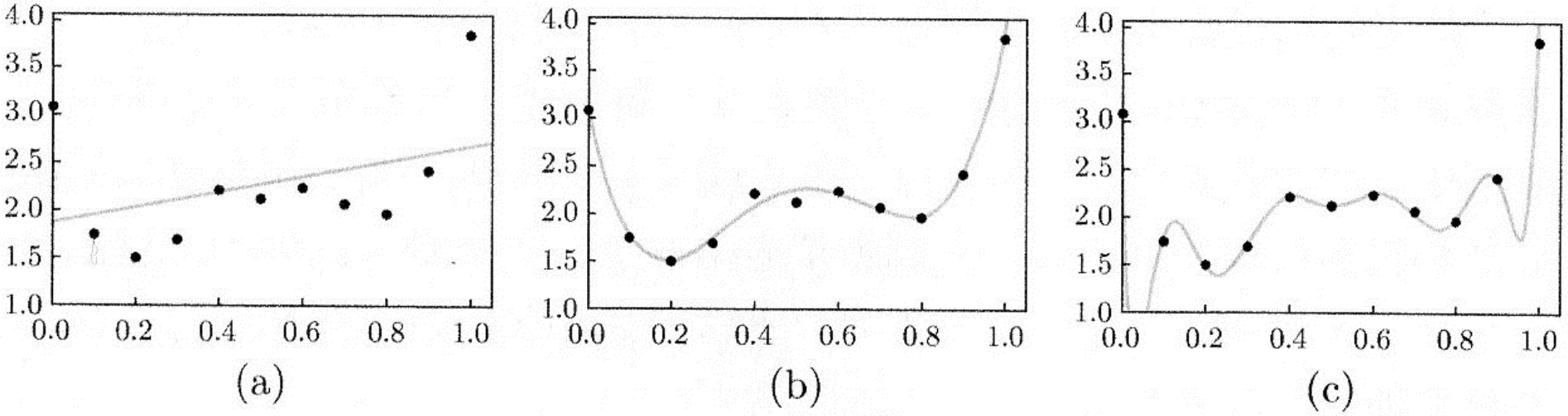


図2.3 データの線形回帰の一例

図2.3(c)の場合には，モデルの自由度が大き過ぎて学習データのノイズまで含めて学習してしまう．訓練データに適応しすぎて未知データに対する予測が望ましくない．いわば「丸暗記で勉強しすぎたせいで融通が利かなくなり，見たことがない問題にはまったく手が出なくなっている受験生」の状況である．過適合，過学習と呼ばれている状況である．

モデルの自由度を減らすと過学習は避けられるが，どんどん訓練誤差が大きくなる．逆に自由度を大きくすると訓練誤差をいくらでも小さくできる反面，汎化誤差が増えてしまう．

実務的には，学習用のデータと別に検証用のデータを用意し，検証用データに関する誤差が増え始めたら，訓練を打ち切るなどする．

2.4.6 正則化

過学習を避ける方法：自由度の多くないモデルを選ぶ

自由度を実質的に減らす手法：正則化

重み減衰

$$E_{wd}(w) = E(w) + \lambda w^T w$$
$$= E(w) + \lambda \left(w_1^2 + w_2^2 + \cdots + w_N^2 \right) \qquad (2.49)$$

この式を最小化することは，できる限り右辺第2項のwの二乗和を小さくすること．可能な限りのwの成分がほぼゼロになる．これをRidge回帰と呼ぶ．正規方程式は $w^* = (XX^T + 2N\lambda I)^{-1} Xy$ になる．

自由度減で過学習を避けるのは機械学習で極く重要

2.4.7 クラス分類

(1) 2値分類: 分類先が2つしかない状況
1つ目のクラスに属する → $y = 1$
2つ目のクラスに属する → $y = 0$

(2) 多クラス分類: 0〜9の手書き数字判別など
K個のクラスに応じて $y = 1, 2, \ldots, K$とする
one-hot 表現: (1,0,0,0,…,K) ← y=1に相当

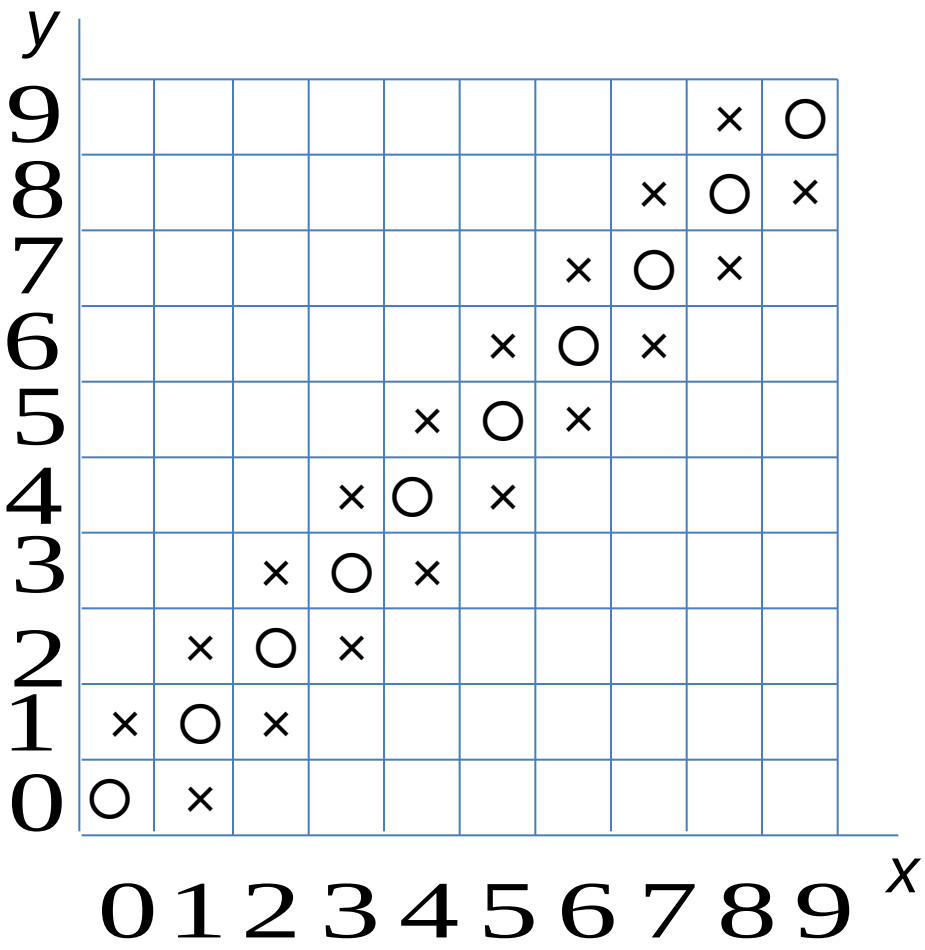
2.4.8 クラス分類へのアプローチ

(1) 関数モデル: 入力と出力の関係を関数とする

$$\hat{y} = y(x; w) \tag{2.55}$$

(2) 生成モデル: データを確率的に取り扱う

生成モデルはまずデータに潜むランダム性を仮定して同時分布$P(x,y)$をモデル化する. xについて周辺化して$P(x)$を求め, ベイズの公式により, $P(y=C_k|x)$を求める. $P(C_k|x)$が最大になることからyの予測値を求める.



頻度または確率分布

(3)　識別モデル: $P(C_k|x)$を直接，計算する

生成モデルはまず同時分布を得て，そこから$P(C_k|x)$|を計算している．識別モデルでは，$P(C_k|x)$を直接，計算する．

2.4.9 ロジステック回帰

まず，2クラス分類から考える. $P(C_1 \mid x) + P(C_2 \mid x) = 1$

シグモイド関数: $\sigma(u) = \dfrac{1}{1+e^{-u}} = \dfrac{e^{u}}{1+e^{u}}$ (2.57)

シグモイド関数は $0 < u < 1$ であるので

$$P(C_1 \mid x) = \sigma(u) \quad (2.58)$$

とできる. uは対数オッズ:

$$u = \log \frac{P(C_1 \mid x)}{1 - P(C_1 \mid x)} = \log \frac{P(C_1 \mid x)}{P(C_2 \mid x)} \quad (2.59)$$

$$P(C_1 \mid x) > P(C_2 \mid x) \rightarrow P(C_1 \mid x) / P(C_2 \mid x) > 1 \rightarrow u > 0$$

ロジスティック回帰: 一般線形化モデル

$$u = w^T x + b = w_1 x_1 + w_2 x_2 + b \qquad (2.60)$$

訓練データの学習には最尤法を用いる. この場合

$$P(y=1|x) = P(C_1|x),\ P(y=0|x) = 1 - P(C_1|x) \qquad (2.61)$$

$P(y|x)$ はベルヌーイ分布なので

$$P(y|x) = \left(P(C_1|x)\right)^y \left(1 - P(C_1|x)\right)^{1-y} \qquad (2.62)$$

データ $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ に対する尤度関数は

$$L(w) = \prod_{n=1}^{N} \left(P(C_1|x_n)\right)^{y_n} \left(1 - P(C_1|x_n)\right)^{1-y_n} \qquad (2.63)$$

w と u の関係は式(2.60)で与えられる.

機械学習の実装で用いられるのは対数尤度であるので，負の対数尤度で誤差関数を定義する.

$$E(w) = -\sum_{n=1}^{N} \left(\begin{array}{l} y_n \log P(C_1 \mid x_n) \\ +(1-y_n)\log(1-P(C_1 \mid x_n)) \end{array} \right) \quad (2.64)$$

これはデータの経験分布とモデル分布の間の交差エントロピー（cross entropy）と呼ばれ，深層学習でもよく登場する重要な誤差関数の一つである.

エントロピー:
情報理論の概念で、あるできごと（事象）が起きた際、それがどれほど起こりにくいかを表す尺度である。珍しいできごとが起これば、それはより多くの「情報」を含んでいると考えられる。情報量はそのできごとが本質的にどの程度の情報を持つかの尺度であるとみなすこともできる。

$$I(E) = \log(1/P(E)) = -\log P(E)$$

（ウィキペディア情報量:
https://ja.wikipedia.org/wiki/%E6%83%85%E5%A0%B1%E9%87%8F）

2.4.10 ソフトマックス回帰

次に多クラス分類の識別モデルを議論する．ロジステック回帰の一般化で取り扱える．まず

$$P(y \mid x) = \prod_{k=1}^{K} (P(C_k \mid x))^{t(y)_k} \tag{2.65}$$

ここで $t(y)_k = \delta_{y,k} = \begin{cases} 1 & (y = k) \\ 0 & (y \neq k) \end{cases}$ (2.54)

式(2.65)はベルヌーイ分布の自然な拡張であり，マルチヌーイ分布またはカテゴリカル分布と呼ばれる．

$\sum_{k=1}^{K} t(y)_k = 1$ が常に成り立っているので，

$t(y)_K = 1 - \sum_{k=1}^{K-1} t(y)_k$ であるので

$$P(y \mid x) = \prod_{k=1}^{K-1} (P(C_k \mid x))^{t(y)_k} \cdot (P(C_K \mid x))^{t(y)_K}$$

$$= \prod_{k=1}^{K-1} (P(C_k \mid x))^{t(y)_k} (P(C_K \mid x))^{1-\sum_{k=1}^{K-1} t(y)_k}$$

$$= \prod_{k=1}^{K-1} \left(\frac{P(C_k \mid x)}{P(C_K \mid x)} \right)^{t(y)_k} P(C_K \mid x)$$

$$= P(C_K \mid x) \prod_{k=1}^{K-1} \left(e^{u_k} \right)^{t(y)_k} = P(C_K \mid x) e^{\sum_{k=1}^{K-1} t(y)_k u_k} \quad (2.66)$$

が得られる. 対数オッズを多クラスに一般化した

$$u_k = \log \frac{P(C_k \mid x)}{P(C_K \mid x)} \quad (2.67)$$

を用いた.

対数オッズの定義から $P(C_K \mid x)e^{u_k} = P(C_k \mid x)$

$\sum_{k=1}^{K} P(C_k \mid x) = 1$ であるので

$$P(C_K \mid x) = \frac{1}{\sum_{k=1}^{K} e^{u_k}} \qquad (2.68)$$

元の式に代入すると $P(C_k \mid x) = \frac{e^{u_k}}{\sum_{k=1}^{K} e^{u_k}}$

書き直すと $P(C_k \mid x) = \mathrm{soft\,max}_k(u_1, u_2, \ldots, u_K)$ (2.69)

ソフトマックス関数とは

$$\mathrm{soft\,max}_k(u_1, u_2, \ldots, u_K) = \frac{e^{u_k}}{\sum_{k'=1}^{K} e^{u_{k'}}} \qquad (2.70)$$

この関数は $\mathrm{soft\,max}_k(u_1, u_2, \ldots, u_K) = 1 \Big/ \sum_{k'=1}^{K} e^{-(u_k - u_{k'})}$

と書き換えられるので，変数のうち ℓ 番目が多よりもはるかに大きな値をとる場合 ($u_l >> u_k$)

この関数は

$$\mathrm{soft\,max}_k(u_1, u_2, \ldots, u_K) \approx \begin{cases} 1 & (k = l) \\ 0 & (k \neq l) \end{cases} \qquad (2.71)$$

この対数オッズを線形関数でモデル化すると

$$\begin{aligned} u_k &= w_k{}^T x + b_k \\ &= w_{k1} x_1 + w_{k2} x_2 + \cdots + w_{kN} x_N + b_k \end{aligned} \qquad (2.72)$$

この線形の対数オッズを用いた識別モデルをソフトマックス回帰（softmax regression）と呼ぶ.

ソフトマックス回帰を最尤法で取り扱う方法もロジスティック回帰の場合と全く同じである. つまり負の対数尤度関数（交差エントロピー）を最小化することで, パラメータを最適化する.

$$E(w_1,\ldots,w_{K-1}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t(y_n)_k \log P(C_k \mid x_n) \quad (2.73)$$

ただし $\sum_{k=1}^{K} P(C_k \mid x_n) = 1, \quad \sum_{k=1}^{K} t(y_n)_k = 1$

という条件がついていることに注意したい. そのため余分な1自由度が消去されて, 式（2.67）により, $u_k = 0$ となっている. K番目のクラスには w_k は付与されない.

2.5 表現学習と深層学習の進展

2.5.1 表現学習

画像データを例にして考える.

原画像そのものではなくて，原画像から特徴量を抽出し，特徴量から画像判別すると，精度が高い.

原画像が複雑になると，特徴量を見つけること自体が難しくなる.

問題ごとに人力で特徴量を見つけることは極めて煩雑で普遍性に欠ける.

特徴量を見つけることを機械学習でやれないかとの発想で出てきたのが，表現学習である.

表現学習は目的に適した特徴量を，学習を通じて自発的に獲得しようというアプローチ．

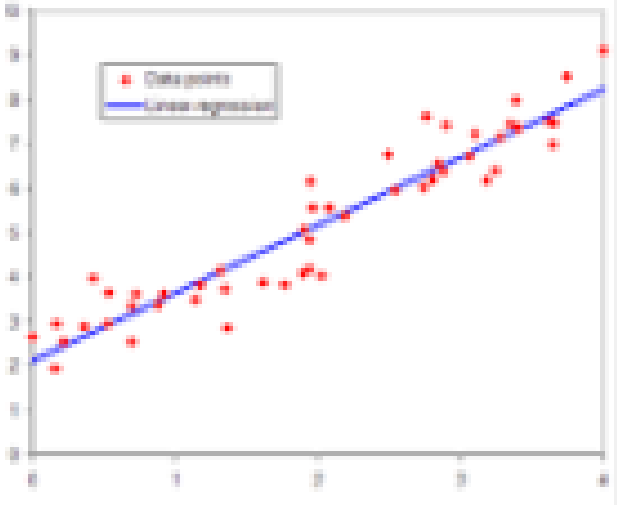
したがってデータは特徴量の推定と実際の回帰という2つの目的のために同時に利用される，

表現学習を取り入れた機械学習では，入力をタスクに最適な表現 $h(x)$ に変換する方法を学習しつつ，その表現を回帰分析する方法である．

回帰の意味？

そもそも回帰とは，とは元の位置，状態に戻ること．

回帰分析は，ちらばったデータ点を生み出した元の直線を求めることなのでこの名がある？



$$x \to h(x) \to \hat{y}(h(x)) \qquad (2.74)$$

データ→タスクに最適な表現→推定量

真相学習がとても重要視されているのは，さまざまなシチュエーションでこのような非自明なデータの表現 $h(x)$ を構築･学習するのに，汎用的で強力な手法を提供しているからである.

深層学習から得られる表現を特に**深層表現**と呼ぶ.

2.5.2 深層学習の登場

深層学習の急激な発展の歴史

現代的な意味での深層学習の起源は，2006年にヒントンが発展させた深層ボルツマンマシンの研究まで遡る.

ヒントンらの研究者は，ニューラルネット冬の時代に，ボルツマンマシンの深層化の困難を解消するためのさまざまなアイデアの開発を推し進めた.

さらには計算機の性能向上にも後押しされて，一昔前では不可能であったであろう深層ボルツマンマシンの実装を成功させた.

アーキテクチュアの深層化がより良い表現を獲得するための際立って優秀な手法であることを立証.

ボルツマシンの成功からしばらく経つと，ニューラルネットでも深層化ができることが分かった．さらにボルツマンマシンで培われたさまざまな技術がニューラルネットに流用できるため，その後はニューラルネットを中心に発展している．

深層学習の威力を広く知らしめた画期的な研究は，2012年のヒントンのグループによるAlexNetでのIRSVRC優勝と，同年のルらによる「Googleの猫」の報告ではないか．

The ImageNet Large Scale Visual Recognition Challenge（ILSVRC）は2010年から始まったImageNet データセットの一部を用いた画像認識のコンペティションです．毎年さまざまな大学や企業が機械学習の先端技術を競う．

100万枚ほどの訓練用自然画像により，画像を1000カテゴリほどに分類する画像認識アーキテクチュアを訓練してその性能を競う.

2010年と2011年は、物体カテゴリ認識に参加したアーキテクチュアは従来型のパターン認識手法に基づいていた.

2010年優勝：NEC-UIUC…トップ5エラー率28%

2011年優勝：Xerox Rec.Cen. Eur…エラー率26%
（人間のエラー率5%）

2012年優勝：ヒントンのグループ…エラー率16%
（深層学習）

2015年優勝：深層学習…エラー率4.8%

ILSVRC2012が話題になった同じ年の6月，ルのスタンフォード大学・Googleのチームが国際会議International Conference on Machine Learning（ICML）でいわゆる「Googleの猫」を発表.

彼らはYoutube動画からランダムに切り出した1000万枚の画像を用いて，9層構造のニューラルネットワークを教師なし学習させた.

その結果は驚くべきもので，ニューラルネットの層構造の内部に，さまざまな「概念」に特異的に反応する「細胞」が自発的に生まれた.

神経科学では，脳にはさまざまな概念に反応する個別の細胞があるという「おばあさん細胞」仮説があります．おばあさんを見たときに発火する細胞です.

生物学では「おばあさん細胞」仮説は必ずしも広く支持されていないが，同じものが，脳を模した深層学習で自動的に実現されたことは興味深い．

図2.4は猫細胞の反応を入力画像に引き戻して再現した，深層学習が学び取った「一般的な猫」の顔である．

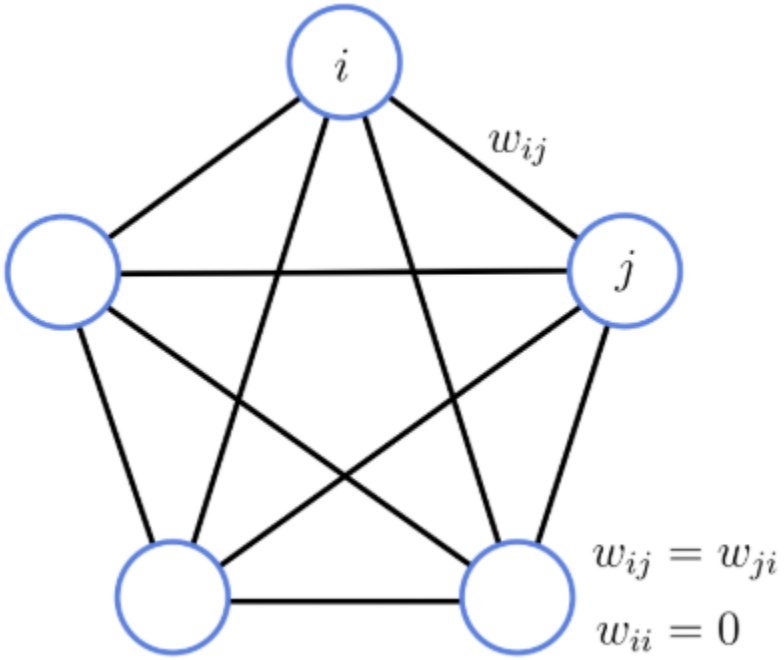
図2.4　ニューラルネットが獲得した「猫の概念」を画像化した「Googleの猫」

深層学習は機械翻訳，対話プログラム，文章からの画像生成，コンピュータ囲碁などで、高い性能を示している．「人工知能の大半が深層学習になる」という専門家も少なくない．

講義担当者のニューロとの関わり

ホップフィールド･マシン(アメリカの物理学者が提唱)

相互結合型のネットワークは非線形の回路網なので，ある初期状態から活動させると，いくつかの平衡状態のどれかに収束する．どれに収束するかは初期状態による．



ホップフィールド･マシン

$w_{ij} = w_{ji}$ は神経回路では正しくない

ネットワークの状態遷移は非同期的である時刻 t にランダムに選ばれたニューロン j が

$$v_j(t+1) = \phi(u_j(t)) \qquad u_j(t) = \sum_{i \neq j} w_{ji} u_i(t) + \theta_j$$

で遷移し，j 以外の i は遷移しない．$v_i(t+1) = v_i(t)$

$\phi(u)$ はユニットの出力関数：$\phi(u) = \begin{cases} 1 & \text{when } u > 0 \\ 0 & \text{when } u \leq 0 \end{cases}$

スピンモデルとのアナロジからネットワークのエネルギ $E(t)$:

$$E(t) = -\frac{1}{2} \sum_{j \neq i} \sum w_{ji} v_j(t) v_i(t) - \sum_j \theta_j v_j(t)$$

上述の状態遷移で

$$
\begin{aligned}
&E(t+1)-E(t)\\
&=-\sum_{j\neq i} w_{ji}v_j(t+1)v_i(t)-\theta_j v_j(t+1)+\sum_{j\neq i} w_{ji}v_j(t)v_i(t)+\theta_j v_j(t)\\
&=-\sum_{j\neq i} w_{ji}\left[v_j(t+1)-v_j(t)\right]v_i(t)-\theta_j\left[v_j(t+1)-v_j(t)\right]\\
&=-\left[v_j(t+1)-v_j(t)\right]\left[\sum_{j\neq i} w_{ji}v_i(t)-\theta_j\right]\\
&=-\left[v_j(t+1)-v_j(t)\right]u_j(t)\leq 0
\end{aligned}
$$

となる. 最後の不等式は下記による.

$$
\begin{aligned}
&\text{If}\quad u_j(t)>0,\quad v_j(t+1)=1\rightarrow v_j(t+1)-v_j(t)\geq 0\\
&\text{If}\quad u_j(t)\leq 0,\quad v_j(t+1)=0\rightarrow v_j(t+1)-v_j(t)\leq 0
\end{aligned}
$$

したがって，常に　$E(t+1) \leq E(t)$

なので $v_j(t)$ はエネルギ極小の状態に収束する.

この性質を利用すると，巡回セールスマン問題（TSP）が解ける.

（http://leo.ec.t.kanazawa-u.ac.jp/~nakayama/edu/file/inct_ai_neural_hf-tsp.pdf）

ホップフィールドニューラルネットワークによる巡回セールスマン問題

◆巡回セールスマン問題（TSP）とは

N個ある都市全てを最小コストで訪問する経路を求める問題であり，一つの都市は1回のみ訪問する．

◆TSPの問題設定

都市数： N(＝4で説明する)

都市名： A, B, C, D

都市間距離(旅費)： $d_{XY},\ X,Y = A,B,C,D$

◆ニューラルネットワークの構成

訪問順　1　2　3　4　　(ニューロンの記号)

都市名								
	○	○	○	○	A1	A2	A3	A4
	○	○	○	○	B1	B2	B3	B4
	○	○	○	○	C1	C2	C3	C4
	○	○	○	○	D1	D2	D3	D4

ニューロンの状態＝1または0

ニューロンXnの状態＝1　→　都市Xをn番目に訪問することを意味する.

「全ての都市を1回のみ訪問する」

◆ネットワークの状態

①全ての行において状態＝1となるニューロンは1個のみ.
②全ての列において状態＝1となるニューロンは1個のみ.
③ネットワーク全体では状態＝1となるニューロンはN個.

◆結合重みに対する条件

①同じ行にあるニューロンは相互に抑制する.
　負の結合重みで結合する$(-\alpha)$.
②同じ列にあるニューロンは相互に抑制する.
　負の結合重みで結合する$(-\beta)$.
③正のバイアスを加える.
　①, ②は抑制のみであるから, 状態＝1となるニューロンの数が零になる可能性があるので, 全ての結合重み（自己ループは除く）に正の値を加える$(+\delta)$.

「最小コストで訪問する」

◆都市間移動の制約条件
都市間の距離（旅費）が大きいほど，移動困難とする．

◆結合重みに対する条件
都市Xをn番目に訪問し，都市Yをn＋1番目に訪問する場合を考える（X≠Y）．
ニューロンXnとYn＋1を結合重みで結ぶ（対称結合）．
重みを$-\gamma d_{XY}$とする．（d_{XY}＝都市X，Y間の距離（旅費））

この結合重みにより，もし，d_{XY}が大きいとニューロンXnの状態＝1となったとき，ニューロンYn＋1は活性化しにくくなる．逆にd_{XY}が小さいときはニューロンYn＋1は活性化しやすくなる．

◆結合重みの設計（まとめ）

ニューラルネットワーク

A1 A2 A3 A4
B1 B2 B3 B4
C1 C2 C3 C4
D1 D2 D3 D4

●XmとXn（m≠n）の結合重み

$-\alpha$

●XnとYn（X≠Y）の結合重み

$-\beta$

●XnとYn＋1（X≠Y）の結合重み

$-\gamma d_{XY}$

（X1とY4の結合重みも含まれる）

●バイアス

全ての結合重み（自己ループ除く）に正数を加算

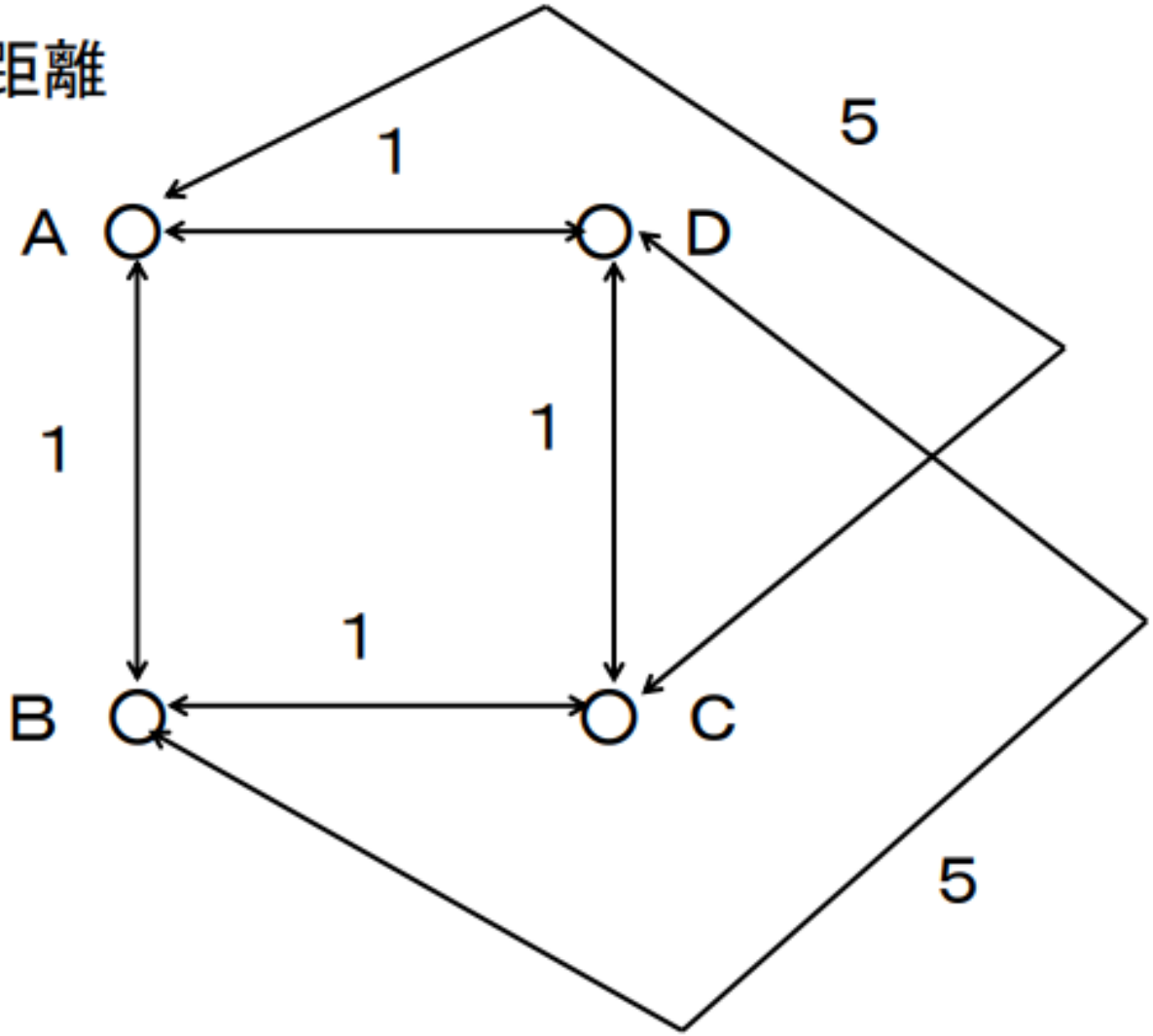
δ

Excelによるシミュレーション

- 都市数＝4
- 都市間の距離（旅費）　　ユーザが決める
- 結合重みに対するスケーリング係数　ユーザが決める
  $(\alpha,\ \beta,\ \gamma,\ \delta)$
- ニューラルネットワークの初期状態　ユーザが決める
- ニューロン数＝4×4＝16個
- 結合重み：　自動生成
- 1ラウンド：　16個のニューロンの状態遷移
- 状態変化させるニューロンの順番（収束結果に影響）
  A1→A2→A3→A4→B1→B2→B3→B4→
  C1→C2→C3→C4→D1→D2→D3→D4（固定）
- 4ラウンドまでシミュレーション
- エネルギー関数を表示

例題

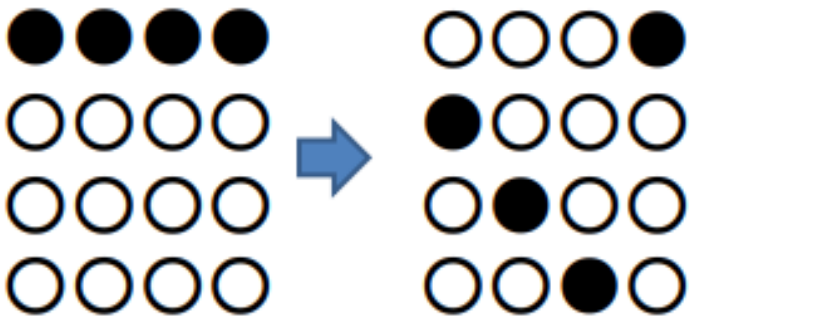
都市間の距離



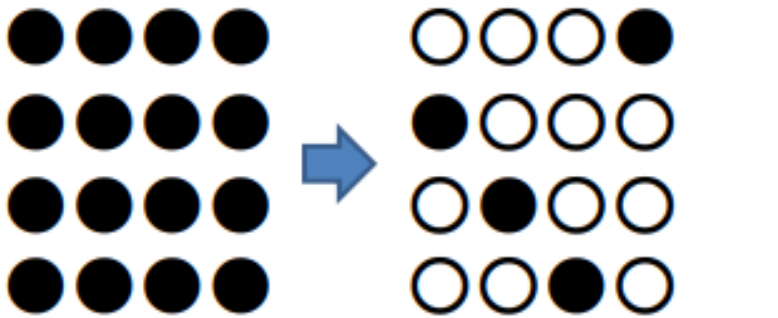
スケーリング係数

$\alpha = 8, \quad \beta = 8, \quad \gamma = 1, \quad \delta = 3$

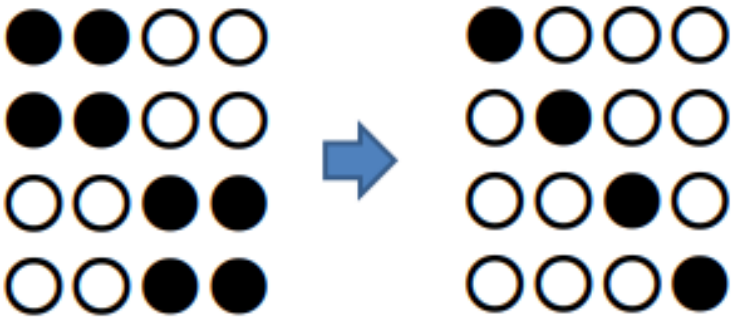
シミュレーション結果（1）成功例



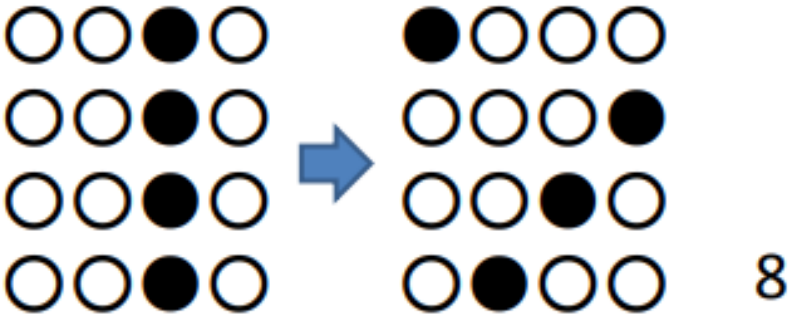
B→C→D→A→B



B→C→D→A→B

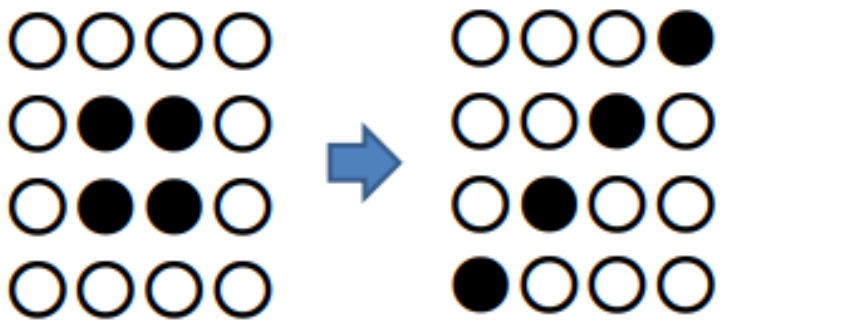


A→B→C→D→A



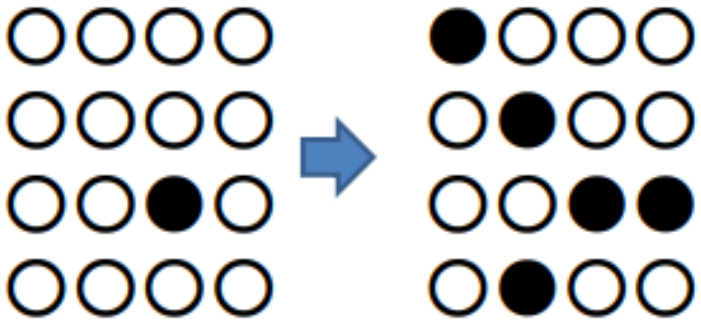
A→D→C→B→A

シミュレーション結果（１－続き－）

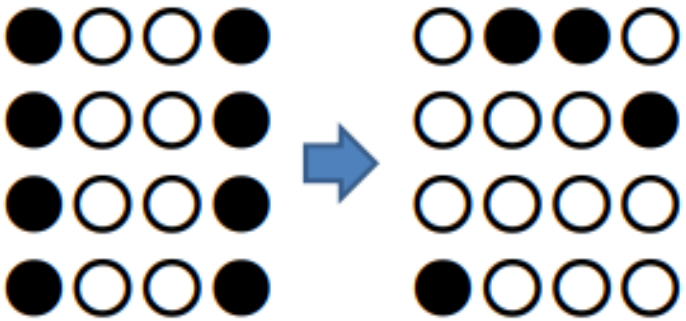


D→C→B→A→D

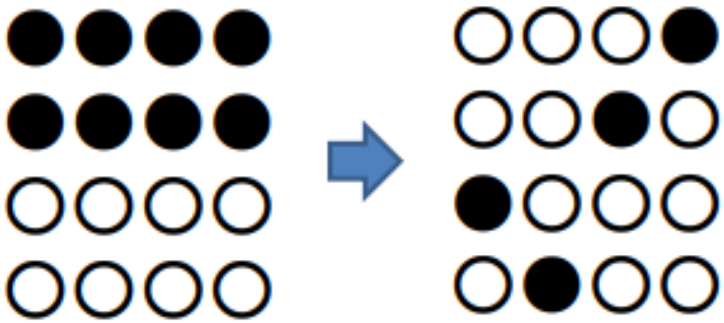
シミュレーション結果（2）不成功例



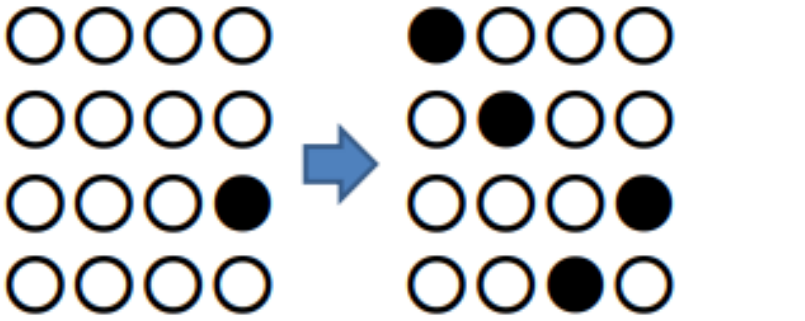
不成功



不成功



C→D→B→A→C
（最短距離ではない）



A→B→D→C→A
（最短距離ではない）

演習問題

巡回セールスマン問題についてExcelのプログラムを用いて以下の問に答えよ.

1. 都市数＝4（A，B，C，D）として都市間距離を決めよ.
   図（スライド参照）と行列（プログラム参照）で示せ.
2. スケーリング係数（$\alpha, \beta, \gamma, \delta$）を求めよ.
   スライドの例題における「結合重みの分布」や「平均」を参考にする.
3. 初期状態を変えてプログラムを実行し，成功例（3例）と不成功例（3例）を求めて図（本スライド参考）で示せ.
4. ホップフィールドネットワークで巡回セールスマン問題を解く際の問題点（難しさ）について考察せよ.

飲み会の代わり…自由討論

分科会運営方法に関するコメント

最近の世相，時代の潮流

夢：遠隔地音楽アンサンブル
　時間遅れをどう克服するか？
　音楽合奏に関する鍋島さんの経験
　ユニソンからアンサンブルへ
　ノイズキャンセラーが悪さをする？
　　A, Bの2信号があると、一方が信号，他方がノイズになり、抑えられる。

末吉先生に紹介していただいたVisual Studio 2012をダウンロードしてみました。C，C++，Basic などが統合されていますので，大変便利と思います.

夢：遠隔地音楽アンサンブル

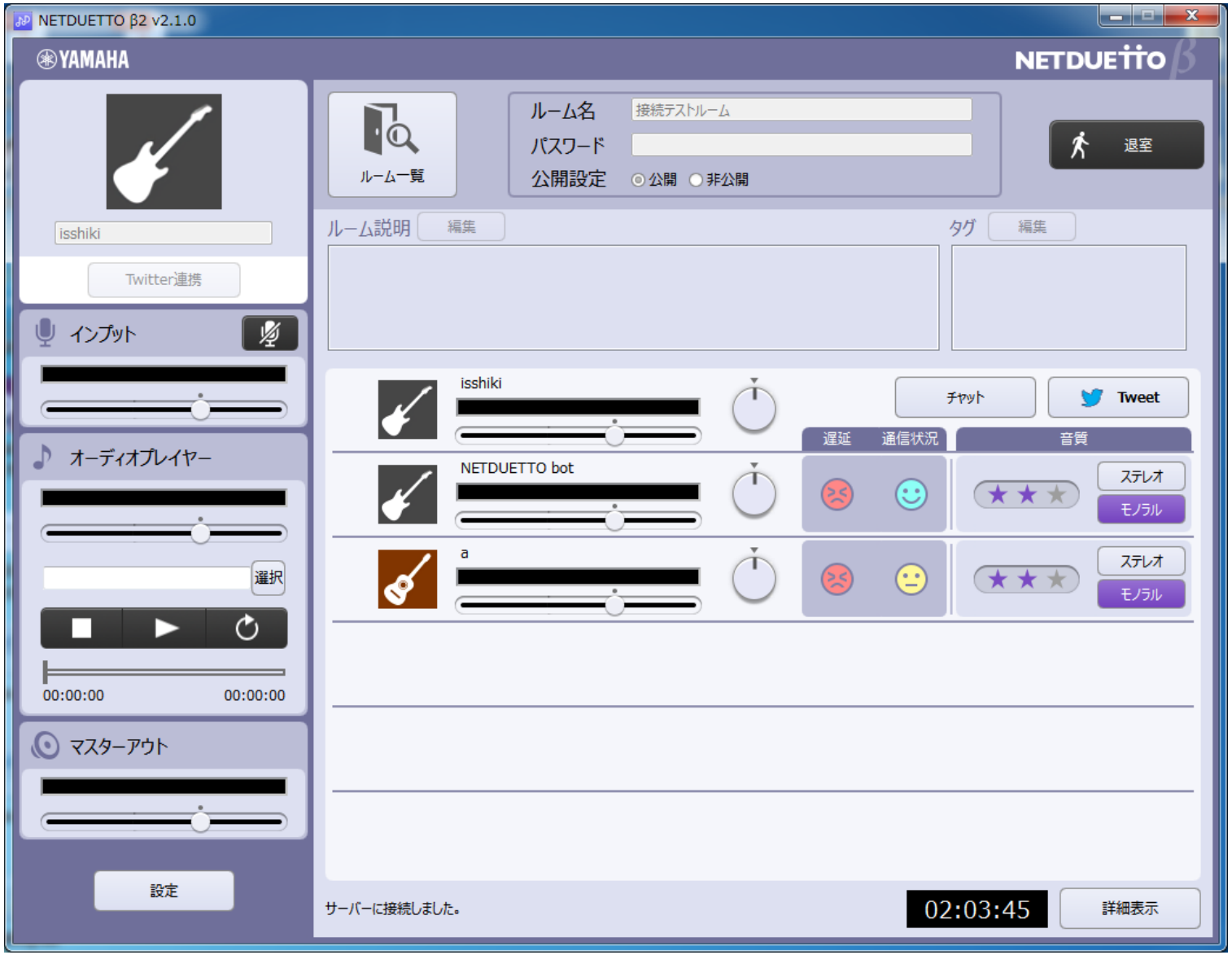
ヤマハNETDUETTO（宮前さんからの情報）

現行の通信回線でよい.

各パーツの時間遅れを極力小さくする.

ダウンロードの仕方
NETDUETTOラボ http://netduetto.net/manual/

ヤマハNETDUETTOの画面



Q.音楽セッションをするのに目処となる遅延時間を教えてください。

A.あくまでも目処となる数値ですが以下を目安にしてください。

35msec以下　ある程度の音楽セッションが実現できます。

45msec以下　オケに合わせるような形式なら音楽セッションが可能です。

70msec以下　カラオケでデュエットを楽しむ程度のことが可能です。

音速は約 340m/s ですから 10msec とは 3.4m に相当します。すなわち 30msec とは約 10m 離れて演奏するのと同じ時間差になります。

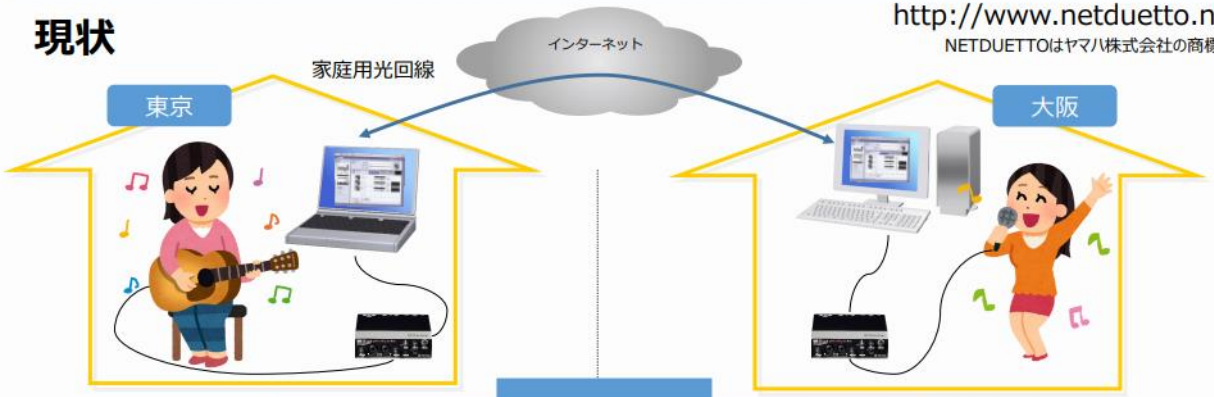
ただし私達が通常 10m 離れている場合はアイコンタクトをして音楽セッションを成立させています。インターネット越しに行う場合はアイコンタクトが使えないのでセッションは難しくなります。

■将来的には、モバイル端末を活用し「いつでもどこでも」音楽が出来る世界を目指します。

NETDUETTO ヤマハの開発した**インターネットを介してリアルタイムな音楽セッション**を行う技術です。

http://www.netduetto.net/
NETDUETTOはヤマハ株式会社の商標です。

現状



５Ｇでは、**場所や機材の制約から自由**に

将来