Neural Networks Learning Differential Data

(微分情報を学習するニューラルネットワーク)

益岡　竜介

発表の概要 (1)

- はじめに
- ニューラルネットワーク
  - 多層パーセプトロンとバックプロパゲーション (BP)
- ニューラルネットワークによる微分の学習
  - アルゴリズムとアーキテクチャ

発表の概要 (2)

- 実装と実験
- 解析
  - BP との比較を中心に
- 強化学習での問題への適用
- Radial Basis Functions (RBF)
- 課題とまとめ

はじめに

学習するシステム

- いかに行動するかではなく、なにをすればいいかを示すだけでよい
- 学習アルゴリズム
  - ニューラルネットワーク、ファジーシステム、遺伝学習アルゴリズム、クラスタリング、強化学習、論理ベースシステム、
- 教師つき学習／教師なし学習
  - ニューラルネットワークなどの「教師つき学習」では与えられた例から学習する

教師つき学習への知識の反映

- 入出力関係の例だけではなく、世界に関するその他の形の知識を学習に反映する
  - 知識としては微分に関する制約、論理で記述された知識、事前確率などがある
  - それらの知識を学習にいっしょに使うことにより効率的かつ正確な学習を可能にする

本研究での結果

- 一般の階数の微分情報を学習するニューラルネットワークの
  - アルゴリズムとアーキテクチャ
  - C++ による実装と実験
  - 解析
  - 強化学習での問題への適用、実験

  - Radial Basis Function (RBF) の高階の誤差項がある場合への拡張

アプリケーション

- 微分情報を学習するニューラルネットワークには以下のようなアプリケーションがある
  - 微分方程式
    - Linear Poisson equation
    - Thermal conduction with non-linear heat generation
    - Plasma equilibrium problem
  - パターン認識 [Simard]
  - Robot learning, deterministic chaos, economics, sensitivity analysis [Hornik]
  - 人間の腕のシミュレーション
    - Minimum torque change model

歴史

- 微分情報を学習するニューラルネットワークについての理論的な研究
  - 存在定理 [Funahashi, Hornik et. al.]
  - 収束定理 [Gallant and White]
- 1 階の微分情報を学習するアルゴリズム (tangent prop) と実装
  - パターン認識に適用 [Simard]

一般階数の微分を学習するアルゴリズムの必要性

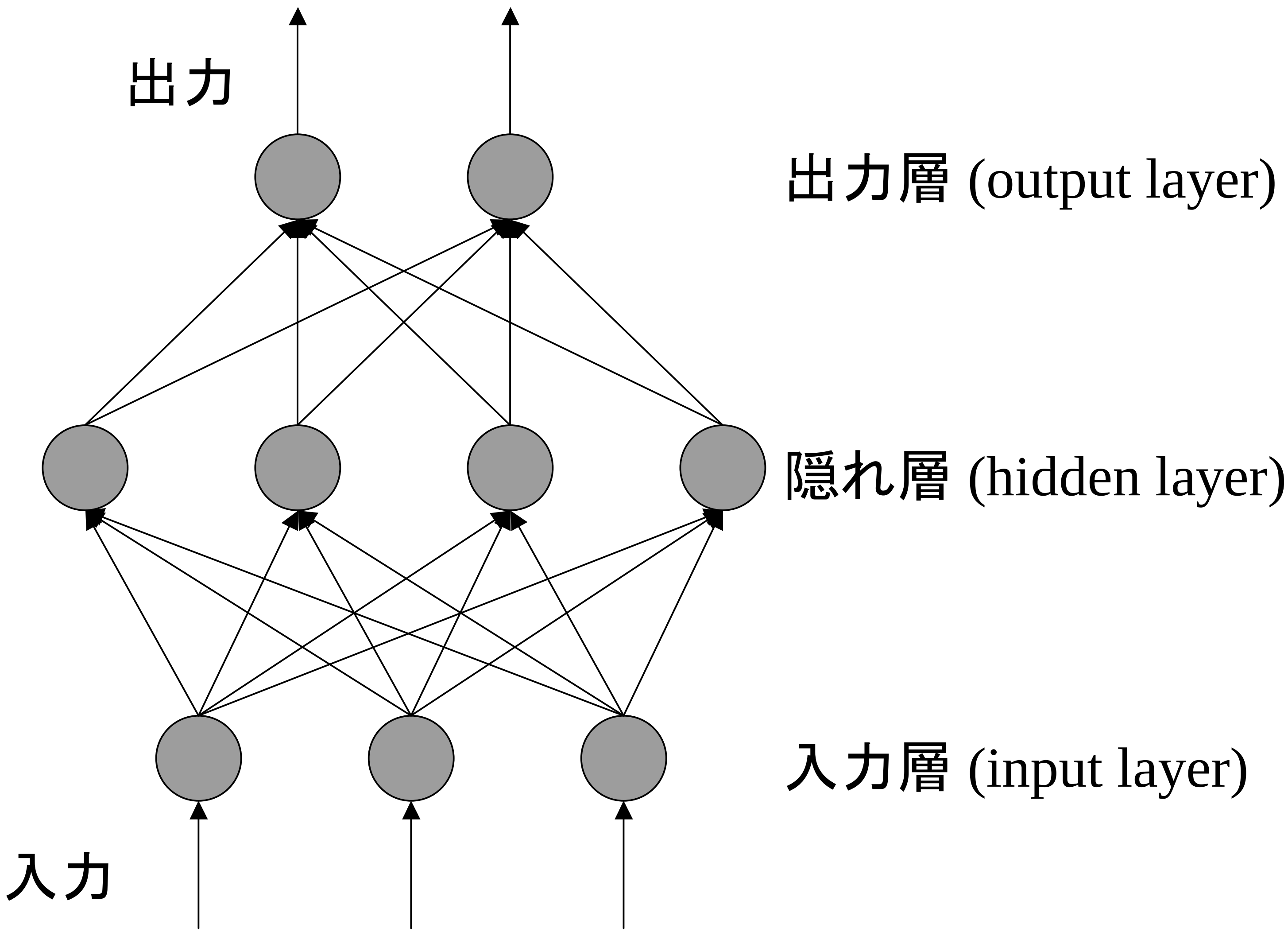
- 必ず最小誤差を与える関数を見つけられる実用的な学習アルゴリズムはない
- 無限にユニットを増やしていくのは現実的でない
- 観測点などが制限されている問題では無限にデータを取ることは不可能である
- 二階以上の微分の情報を用いるアプリケーションがある

ニューラルネットワーク

ニューラルネットワークとは

- たくさんの比較的単純な情報処理要素が、相互に結合して簡単な信号をやりとりするような型のネットワーク状のメカニズム (高等な動物の神経回路網にヒントを得たものです) を使って「情報処理」と呼ばれる仕事をさせる [from 「ニューラルネットワーク情報処理」 麻生 英樹 著]
- Layered network / Mutually connected network
- Back propagation (BP) / Hopfield network / Boltzmann machine / Radial Basis Function (RBF)

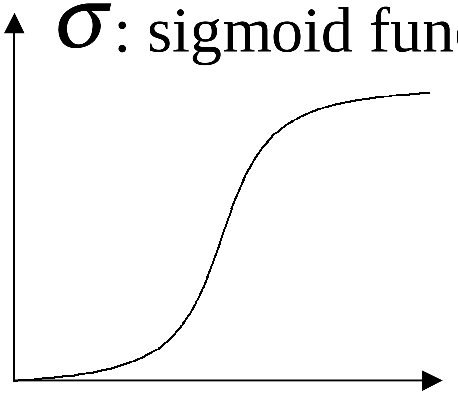
多層パーセプトロン

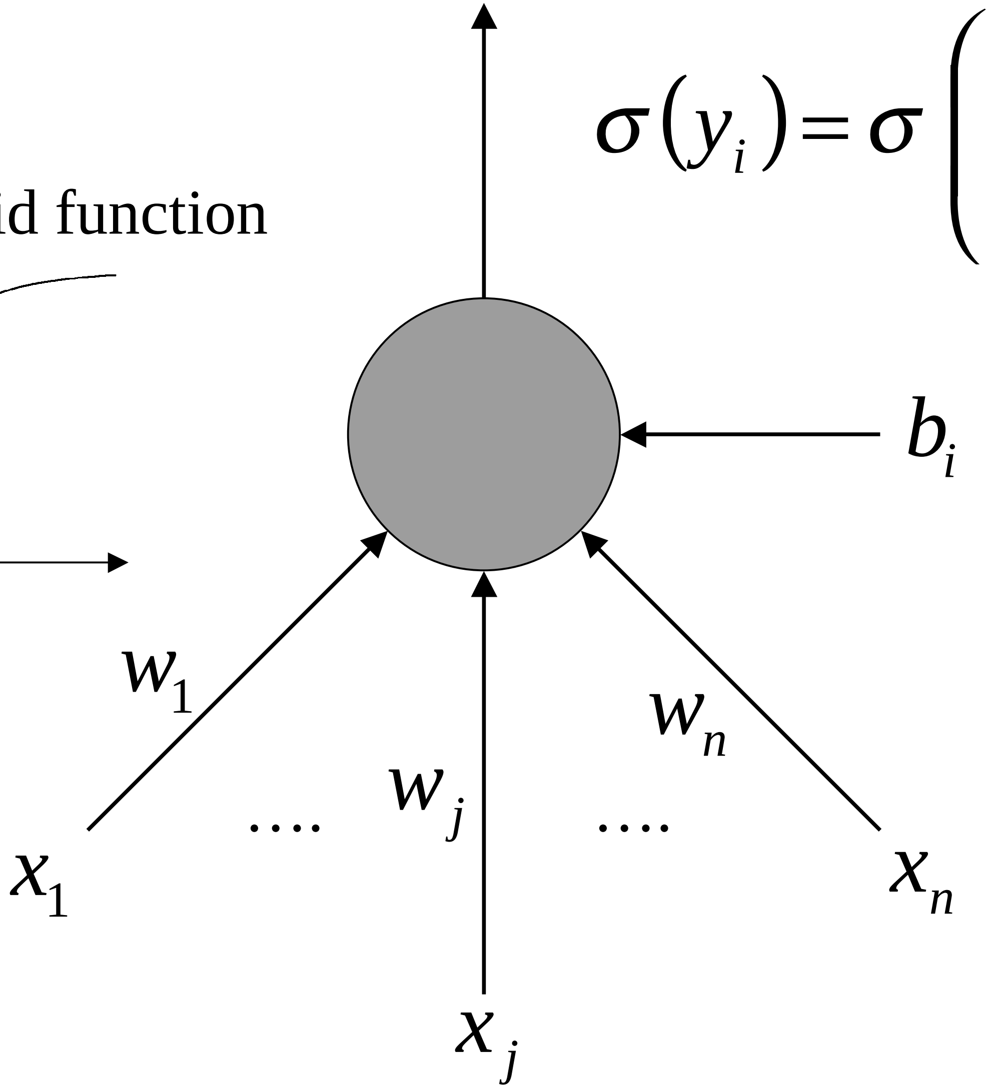


ユニット (ニューロン)

$$\sigma(y_i) = \sigma\left(\sum_{j=1}^{n} w_j\, x_j + b_i\right)$$

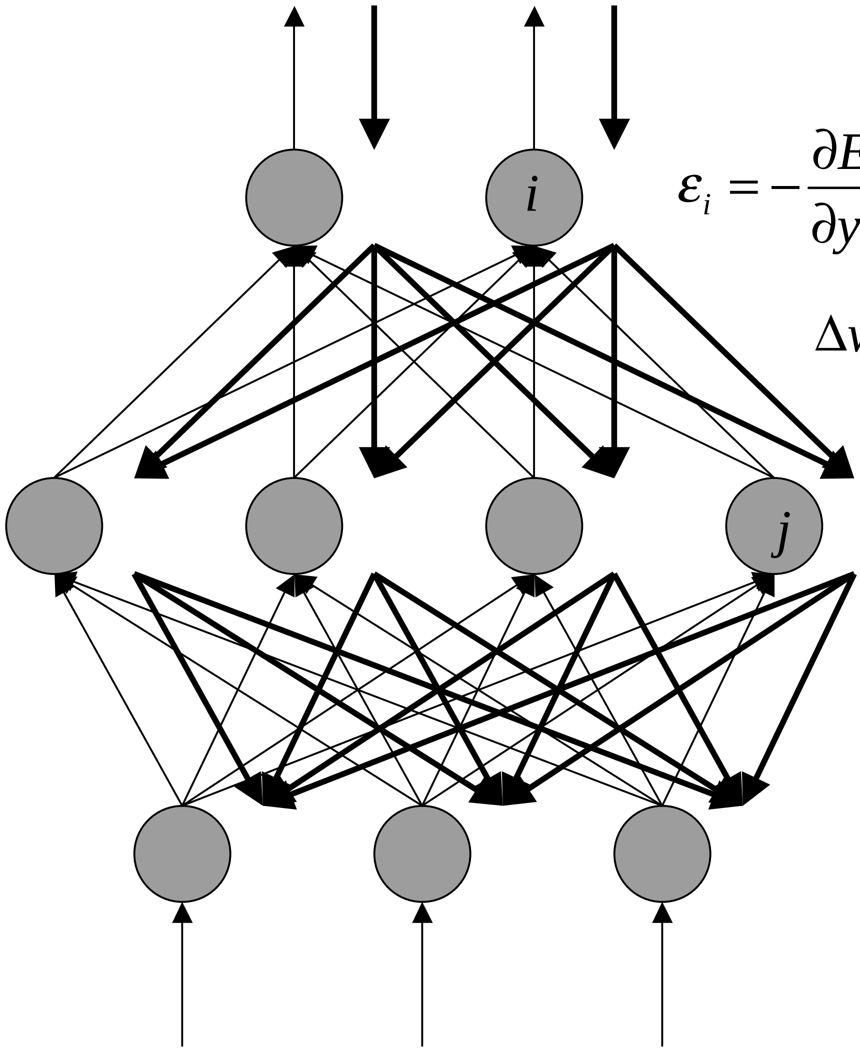
σ: sigmoid function





バックプロパゲーション

$$E = \frac{1}{2}\sum \left(O_i - x_i\right)^2$$



$$\varepsilon_i = -\frac{\partial E}{\partial y_i} = \left(O_i - x_i\right)\sigma'\left(y_i\right)$$

$$\Delta w_{i,j} = -\alpha\,\frac{\partial E}{\partial w_{i,j}} = \alpha\,\varepsilon_i\;x_j$$

$$\varepsilon_j = -\frac{\partial E}{\partial y_j}$$

$$= \sum_k \varepsilon_k w_{k,j} \sigma_k\left(y_j\right)$$

バックプロパゲーションの特徴

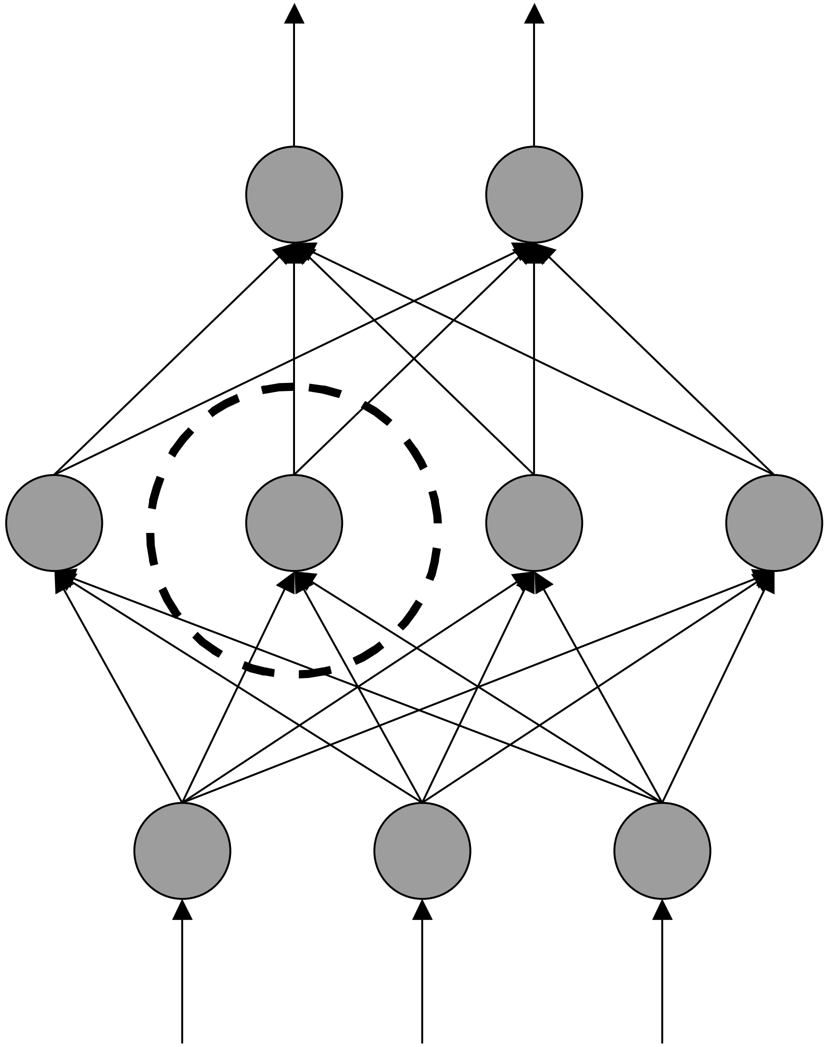
- 値の二乗誤差によるエラー関数を定義
- 重み (やバイアス) をエラー関数に関して gradient descent を行う
- Chain rule によりユニット間にある値を逆向きに伝播 (back propagate) させる
- その値により重みに関する gradient が局所的に計算される

ニューラルネットワークによる微分の学習

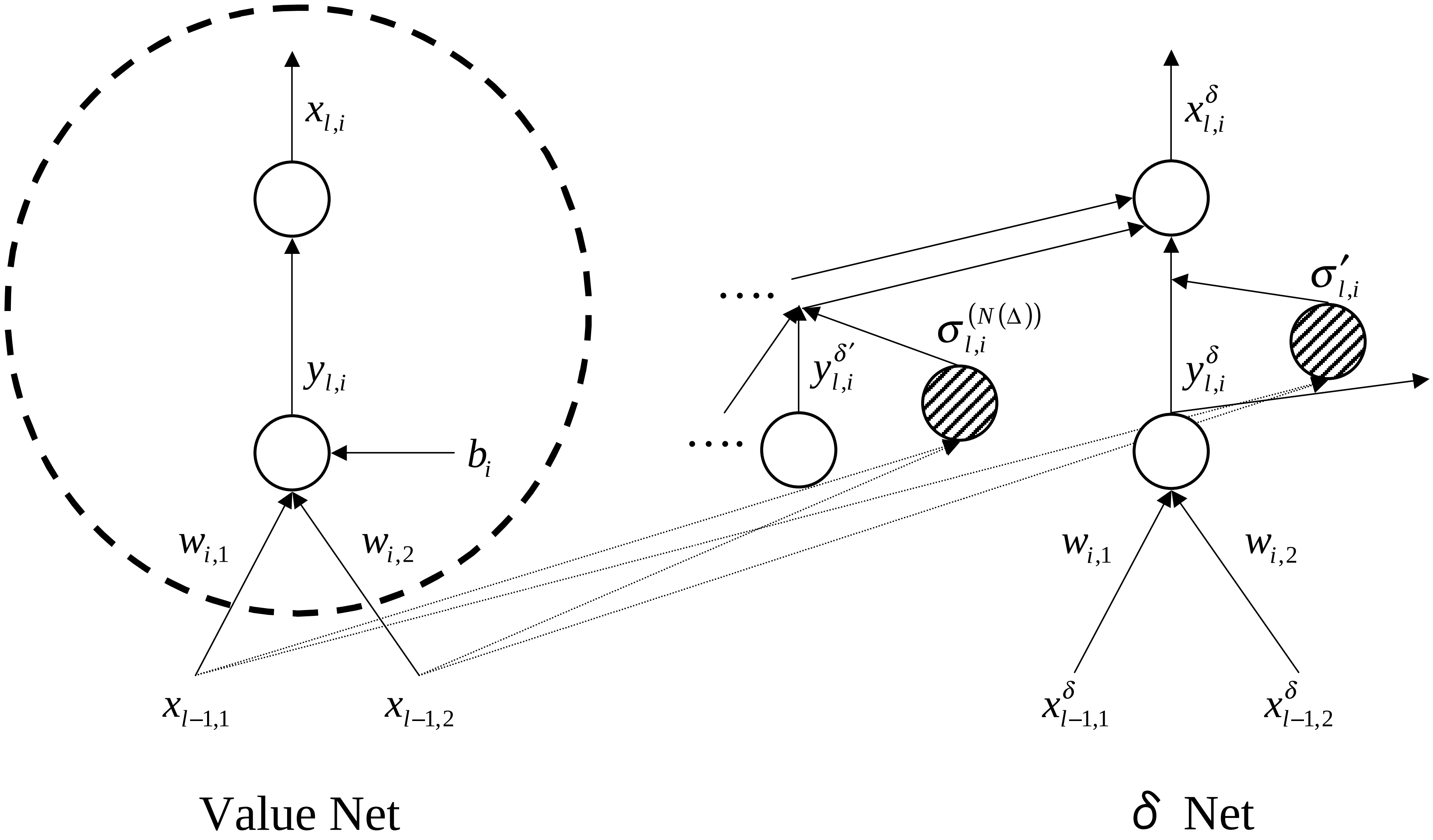
アルゴリズムの特徴

- 入力による微分を forward propagate するネットワークアーキテクチャを構成
- 微分の値の二乗誤差をエラーとして定義
- 上記の二乗誤差に対して BP と同様に重みに関する gradient を、chain rule により逆伝播させる値により局所的な形で計算するアルゴリズム

アーキテクチャ 1



アーキテクチャ 2

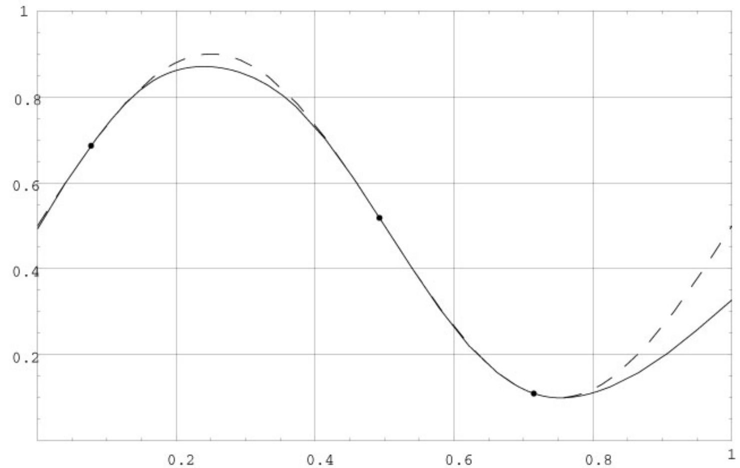

$x_{l,i}$
$y_{l,i}$
b_i
$w_{i,1}$
$w_{i,2}$
$x_{l-1,1}$
$x_{l-1,2}$
....
....
$y_{l,i}^{\delta'}$
$\sigma_{l,i}^{(N(\Delta))}$
$x_{l,i}^{\delta}$
$\sigma'_{l,i}$
$y_{l,i}^{\delta}$
$w_{i,1}$
$w_{i,2}$
$x_{l-1,1}^{\delta}$
$x_{l-1,2}^{\delta}$
Value Net
δ Net

実装と実験

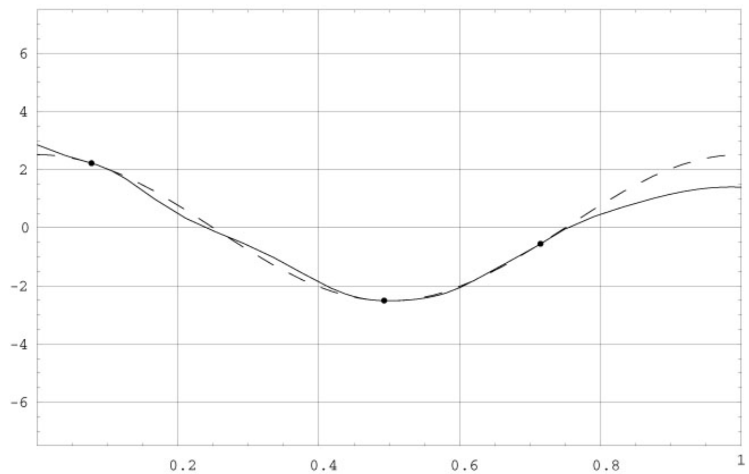
実装の概要

- 全体規模は約 5,000 行
- メインモジュールを C++ で実装
  - その他、Java, Perl, Yacc, Lex などを使用
  - モジュール間はファイルおよび CORBA でやりとり
- Mathematica を使って実装を確認

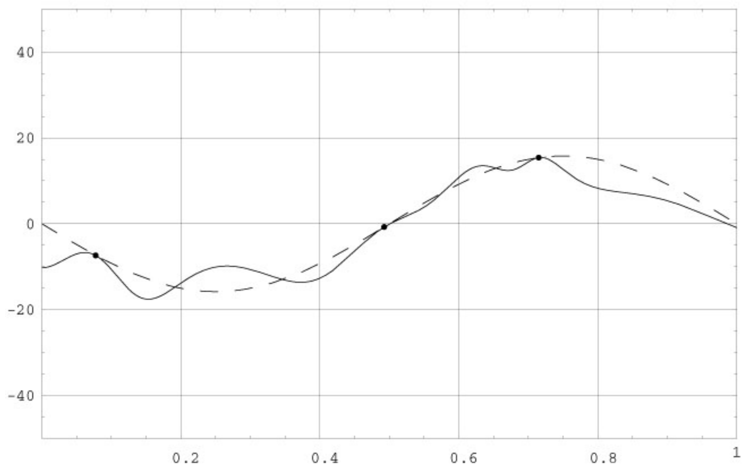
Sine 関数の学習



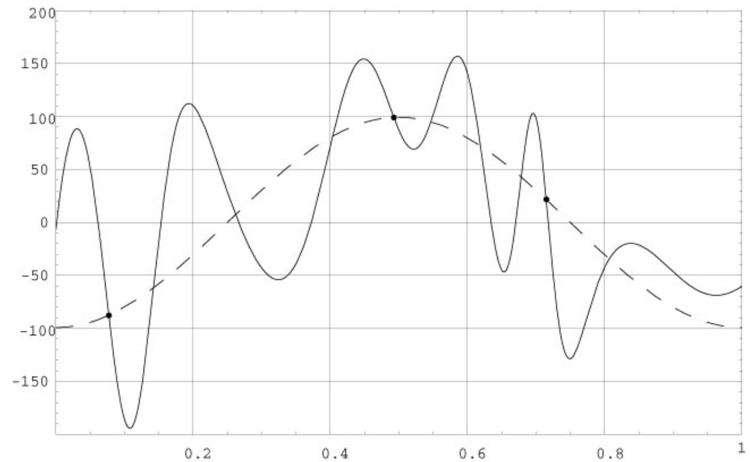
0 階



1 階

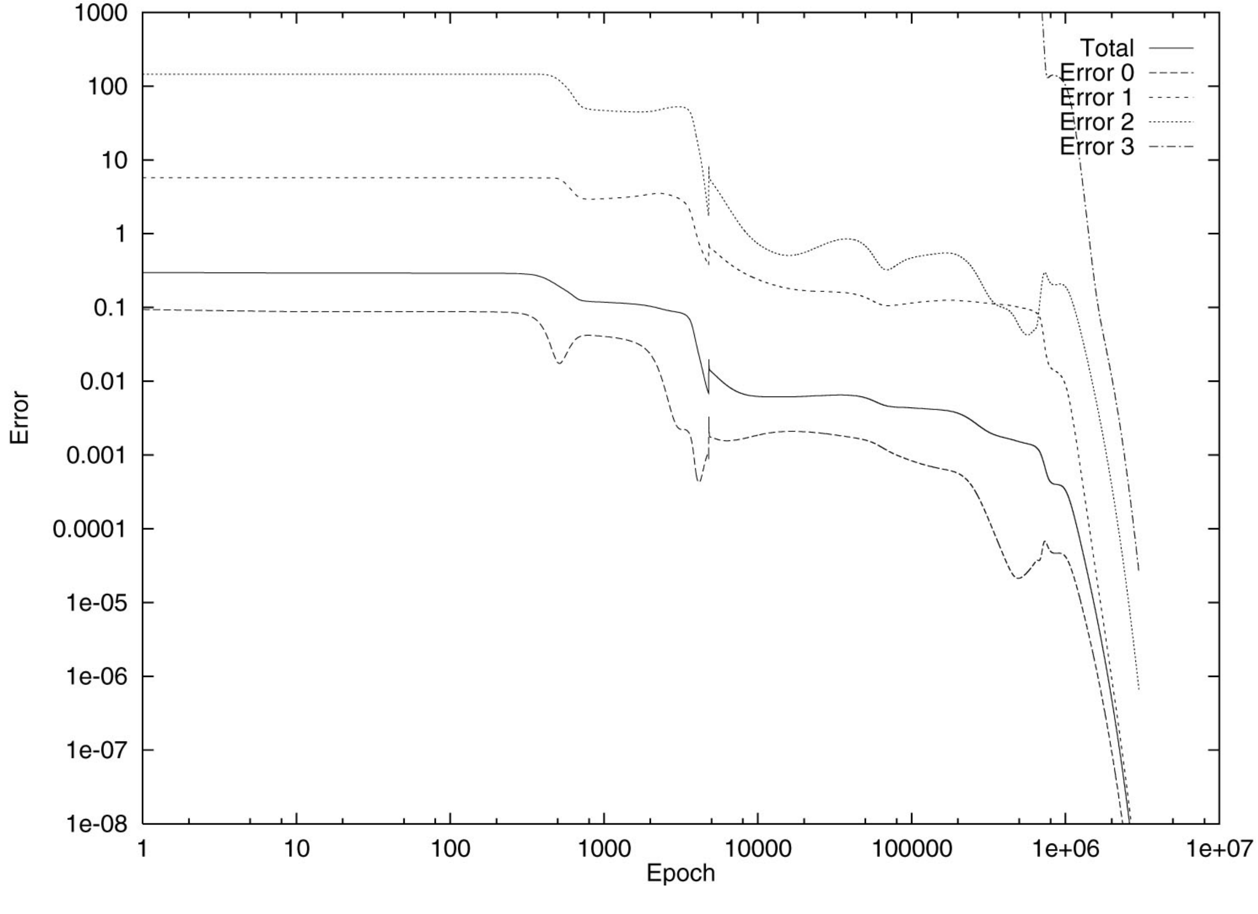


2 階



3 階

Sine 関数の学習 (Learning curve)

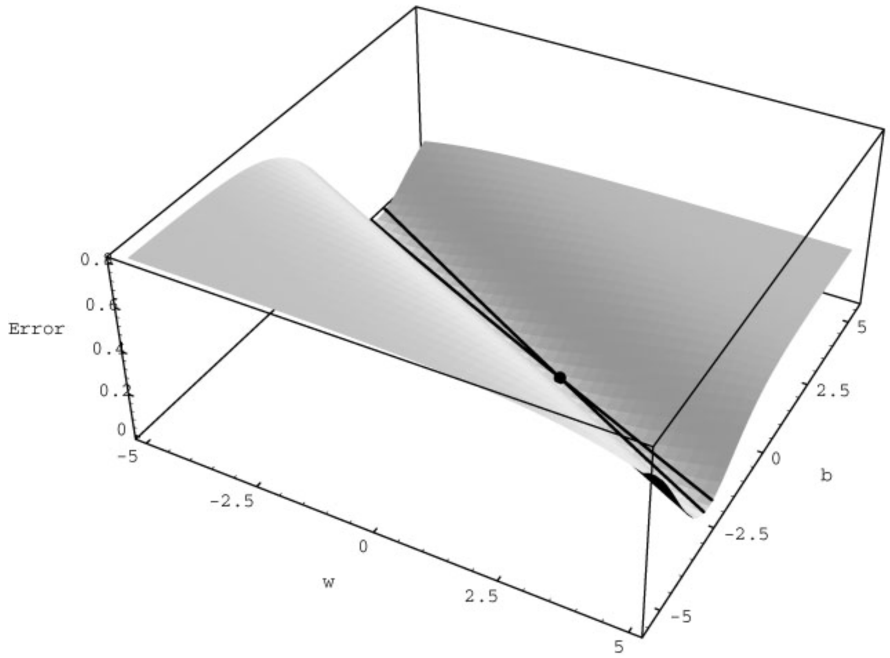


解析

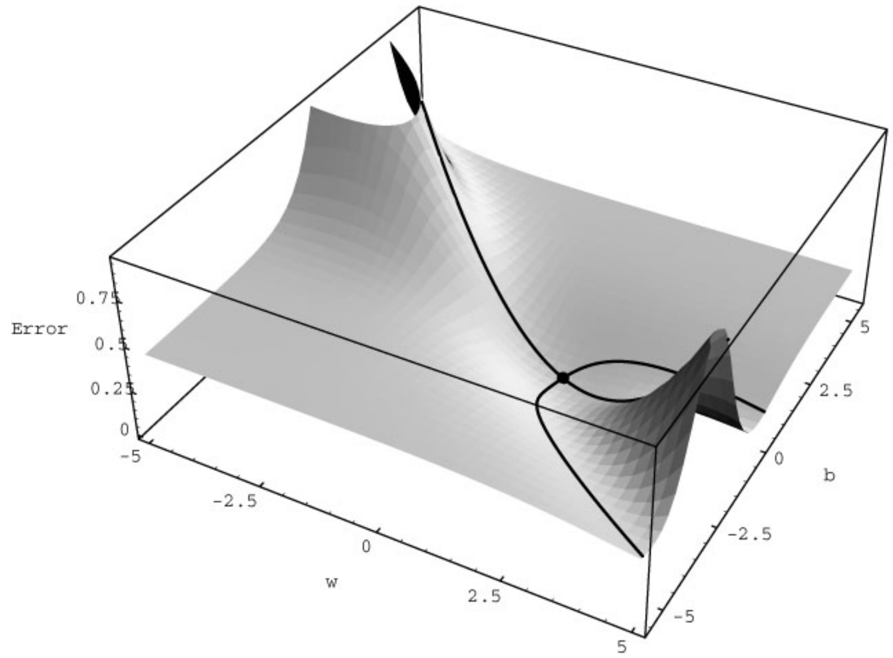
解析で行ったこと

- <u>より多く点をとった時の場合との比較</u>
  - Extra pattern scheme (Simple case)
  - Extra pattern scheme (General case)
- <u>2 次元の関数の学習例</u>
- Sample complexity
- Effect of irrelevant features
- <u>ノイズ耐性</u>

Extra Pattern Scheme (Simple Case)

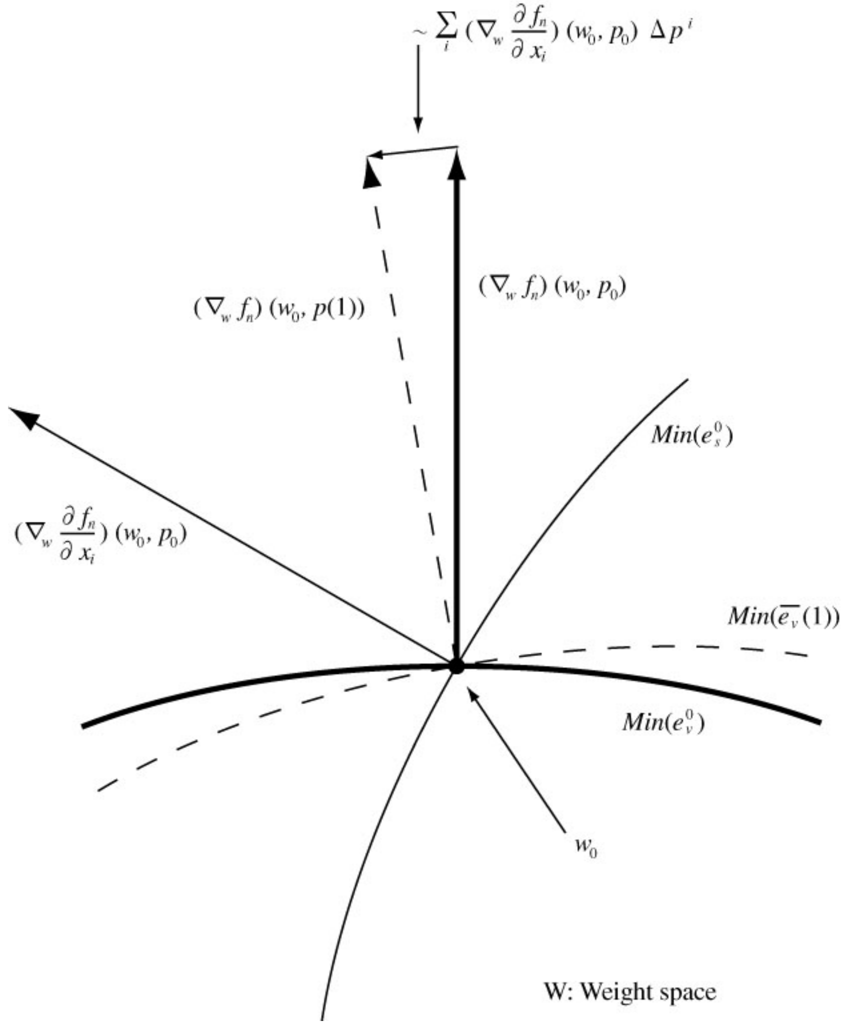


近接した 2 点での値のエラー

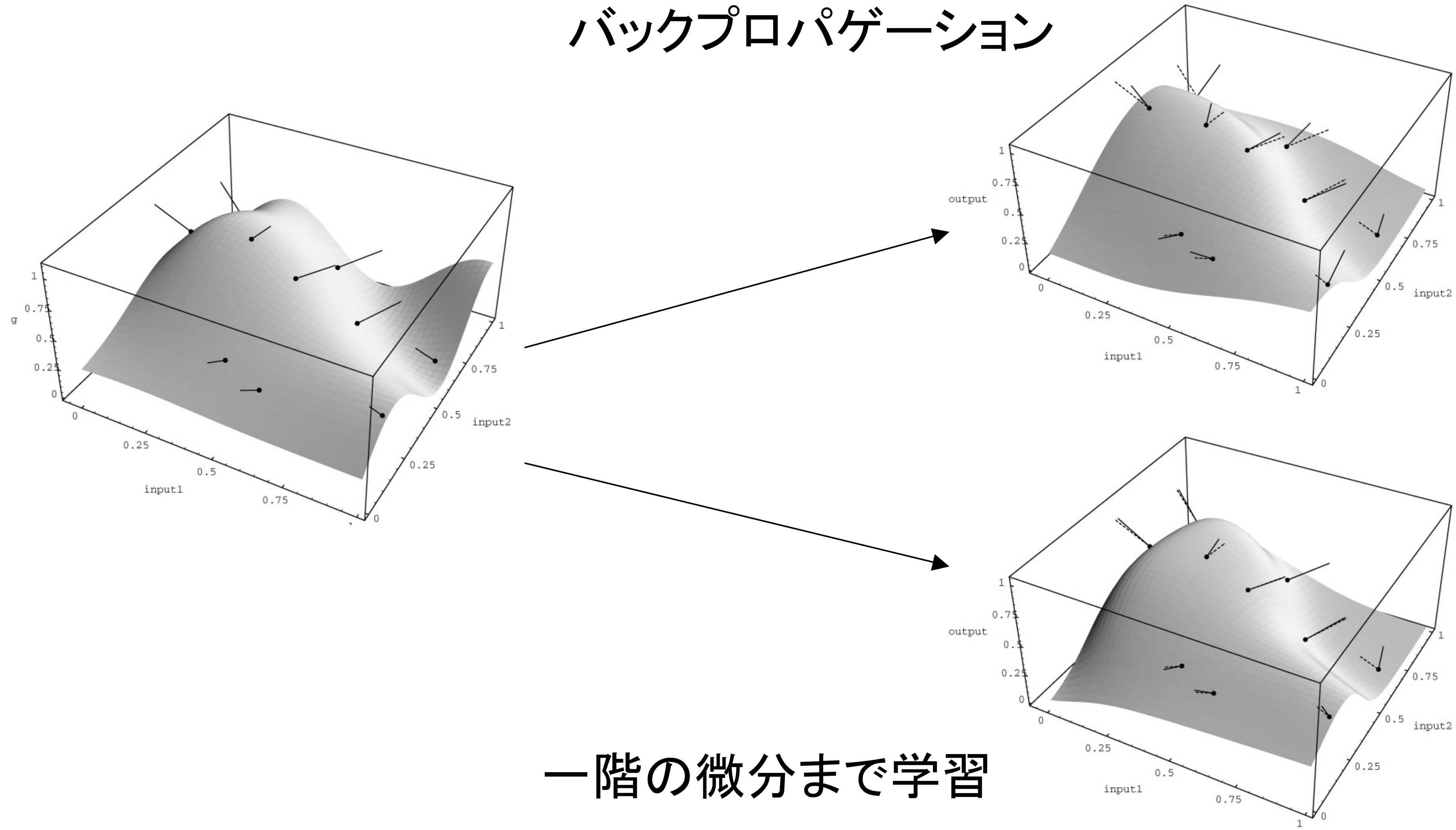


1 点における値と 1 階微分のエラー

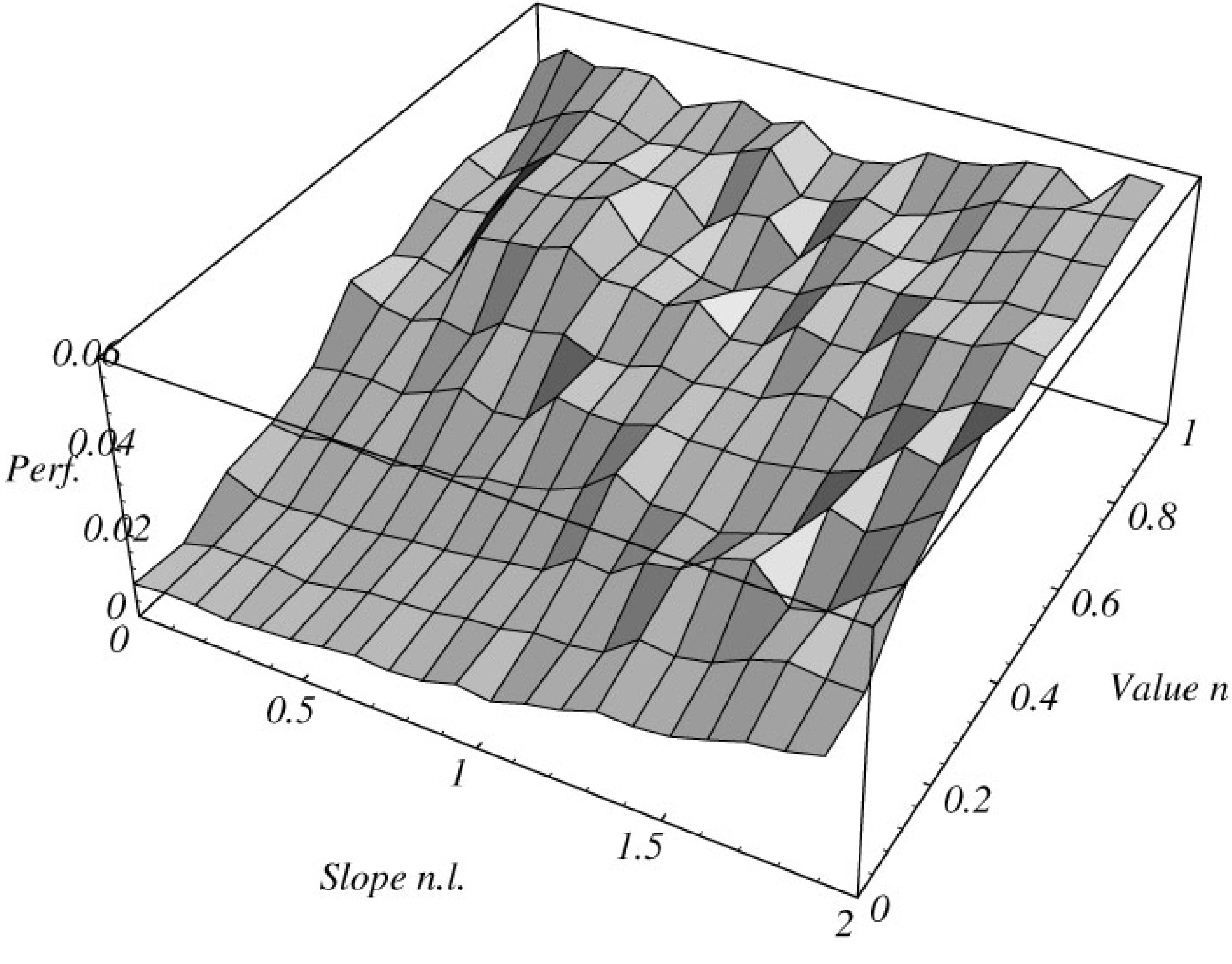
Extra Pattern Scheme (General Case)



2 次元の関数の学習例



ノイズ耐性



強化学習での問題への適用

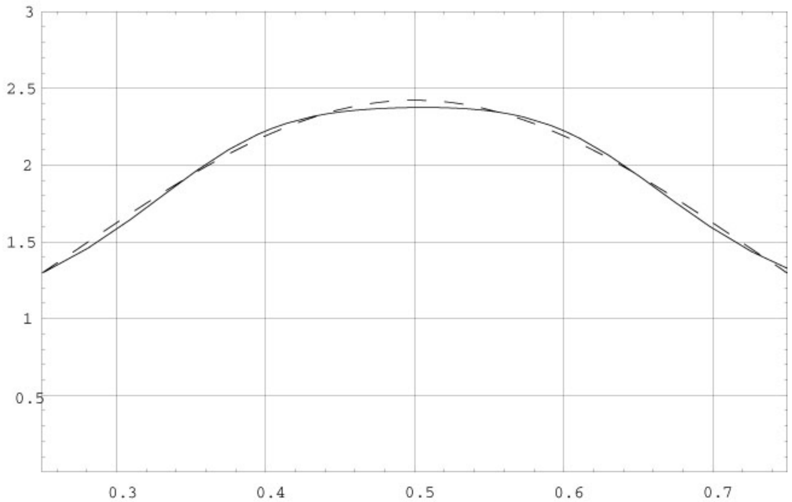
連続的な行動出力

- 強化学習で得られた連続的な行動空間の確率分布に対して、乱数発生器を実現する
- 乱数発生には以下のような方法がある
  - 変換法 → 特定の関数のみ
  - 棄却法 → 時間がかかる
- ニューラルネットワークによって一階の微分方程式を解くことにより、一般の確率分布に対して変換法による乱数発生器を実現する

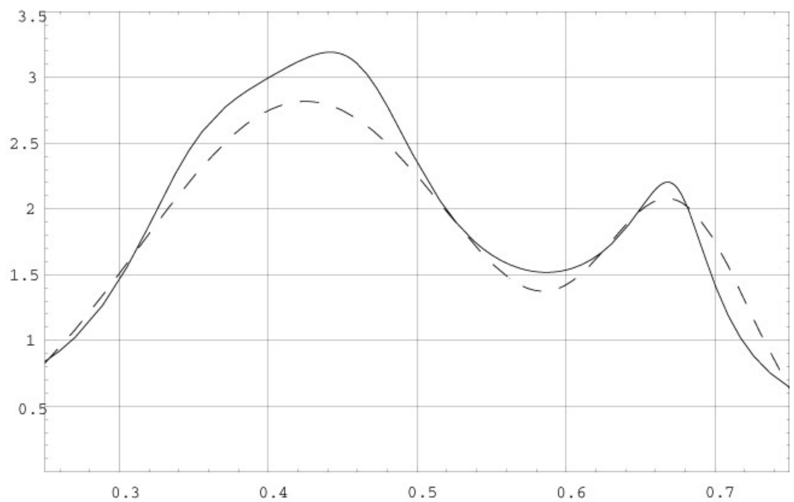
難しい点

- 決まった入力に対しても、学習するデータがネットワークの出力によって動的に変わっていく

実験結果



ガウス確率分布



多峰性確率分布

Radial Basis Functions (RBF)

RBF について

- 歴史
  - Poggio と Girosi が値の二乗誤差と smoothing term を持つ functional の minimizing solution として導入
  - 点からの距離の関数の和であらわされる
  - 一般の kernel を持つ RBF が L^p, C^n で稠密 [Park, Chen, Li]
- 得られた結果
  - 微分のエラータームがある時の minimizing solution の導出
  - C^n 稠密性の別証明

課題とまとめ

今後の課題

- 適切なパラメタ設定方法や収束の性質などの明確化
- ニューラルネットワーク以外の微分情報で与えられる知識を使う学習方法との比較
- 高階の微分情報の場合の解析、統計的な解析
- 任意に与えられた微分方程式に対応できる実装
- その他のアプリケーション

まとめ

- 微分を学習するニューラルネットワークのアルゴリズムとアーキテクチャを構成した
- 実際にインプリメントし、収束することを示した
- 解析や強化学習の問題への適用を行った
- 以上により、微分情報を学習するニューラルネットワークの実用化に道をつけた

Neural Networks Learning Differential Data

(微分情報を学習するニューラルネットワーク)

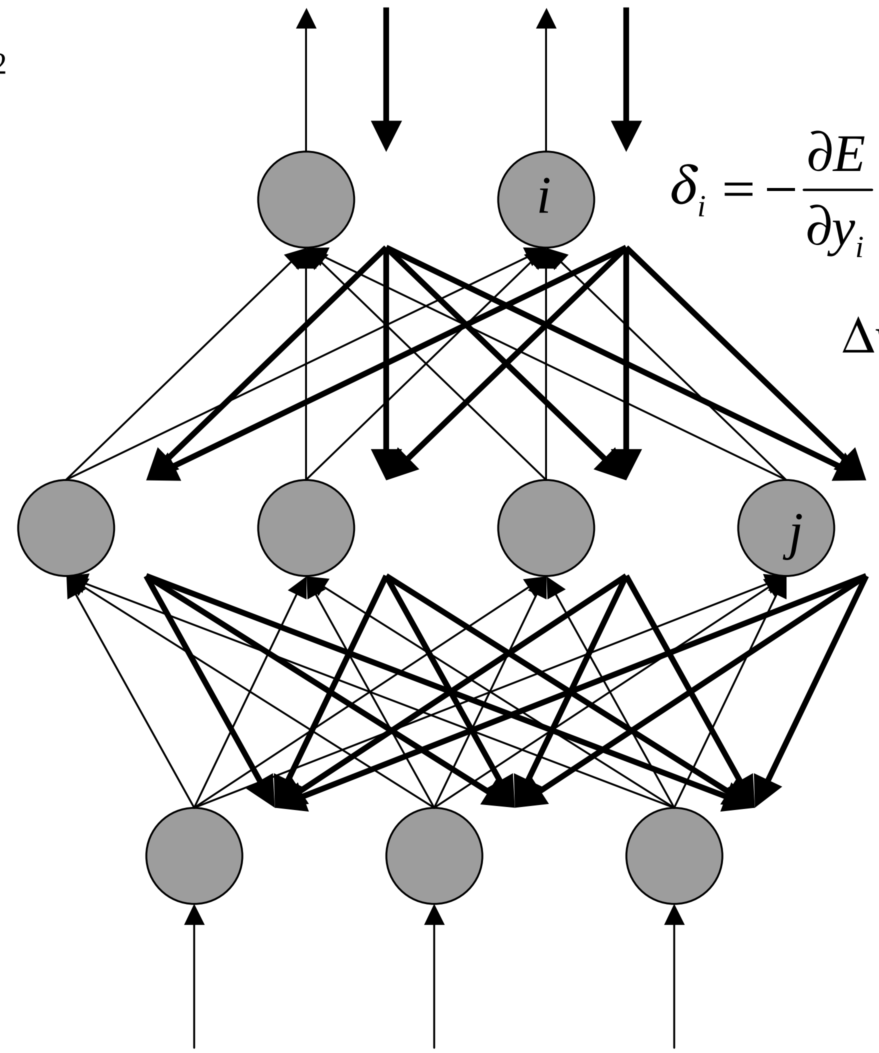
Ryusuke Masuoka

Department of Mathematical Sciences

The University of Tokyo

バックプロパゲーション

$$E = \frac{1}{2}\sum (O_i - x_i)^2$$



$$\delta_i = -\frac{\partial E}{\partial y_i} = (O_i - x_i)\sigma'(net_i)$$

$$\Delta w_{i,j} = -\alpha \frac{\partial E}{\partial w_{i,j}} = \alpha \delta_i x_j$$

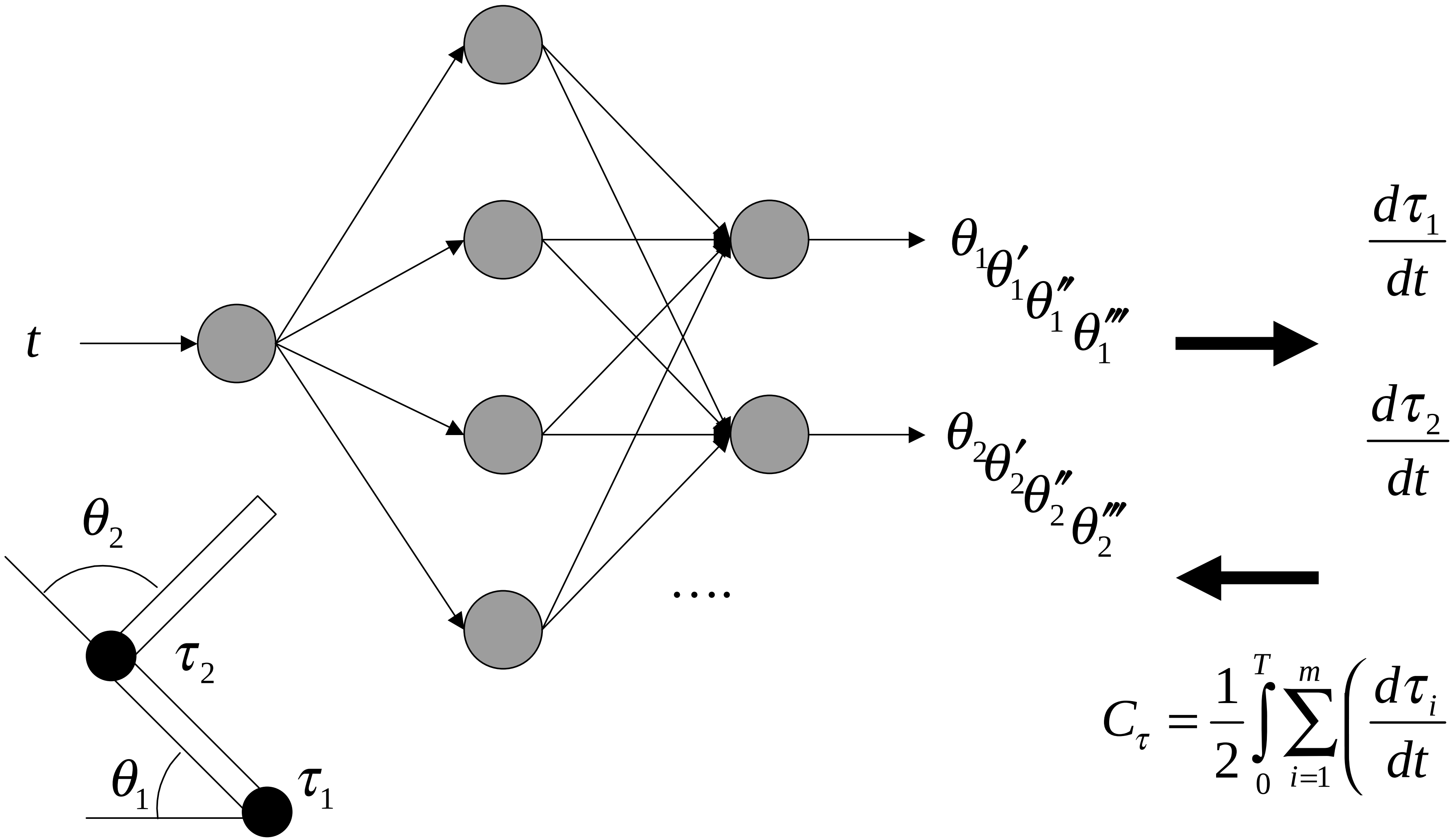
$$\delta_j = -\frac{\partial E}{\partial y_j}$$

$$= \sum_k \delta_k w_{k,j} \sigma_k (net_j)$$

微分方程式への適用

- Linear Poisson equation $\nabla^2 u = \sin(\pi x)\sin(\pi y)$
- Thermal conduction with non-linear heat generation $\nabla^2 T + f(T) = 0$
- Plasma equilibrium problem $\nabla^2 \psi - A - Cr\cos(\theta) = 0$
- Temperature and wind velocity $\frac{\partial T}{\partial t} + (u \cdot \nabla)T = 0$
- MASCON $\nabla f(x) = 0$

Minimum-torque-change Model



$$C_\tau = \frac{1}{2}\int_0^T \sum_{i=1}^{m} \left(\frac{d\tau_i}{dt} \right)^2 dt$$