数理解析研究所講究録
938巻 1996年 33-46

Maximum Evidence Nonlinear Time Series Prediction and Applications

早稲田大学 理工学部 電気工学科
松本 隆　　長南 吉正　　浜田 雅之
Takashi Matsumoto　Yoshimasa Chonan　Masayuki Hamada

chonan@matsumoto.elec.waseda.ac.jp

1 はじめに

背後に非線形性がひそむ時系列予測は、極めて実用的なもの [9],[10],[11] から理論的なものまでそのニーズは多岐多様である。当然のことながら既存の線形手法には限界があり、何らかの形で非線形性を取り込んだ手法が望まれる。ここでは次の2つのクラスの問題を考える。

問題1 時系列 $\{\boldsymbol{x}_t, \boldsymbol{y}_t\}_{t=1}^N$, $\boldsymbol{x}_t \in \mathbb{R}^n$, $\boldsymbol{y}_t \in \mathbb{R}^m$ が与えられた時 $\boldsymbol{y}_t$, $t > N$ をもとに $\boldsymbol{x}_t$, $t > N$ を予測せよ

問題2 時系列 $\{\boldsymbol{x}_t\}_{t=1}^N$, $\boldsymbol{x}_t \in \mathbb{R}^n$ が与えられた時 $\boldsymbol{x}_t$, $t > N$ を予測せよ

前者は、例えば $\boldsymbol{y}_t$ が気温、湿度、太陽光線量、風速等の気象データで、$\boldsymbol{x}_t$ がビルディングやあるいは都市全体の電力や冷水、温水等の消費量を予測する問題であり、後者は自律系の過去のトラジェクトリーから未来値を予測する問題である。いずれも $\{\boldsymbol{x}_t, \boldsymbol{y}_t\}_{t=1}^N$ 又は $\{\boldsymbol{x}_t\}_{t=1}^N$ が与えられているだけであって、その背後のダイナミクスあるいは関数関係は全く与えられていないのが要点である。この様な問題にアプローチするため考えられる一つの手法は適当なパラメータ付けされた非線形予測子を用意し、"学習データ" $\{\boldsymbol{x}_t, \boldsymbol{y}_t\}_{t=1}^N$ 又は $\{\boldsymbol{x}_t\}_{t=1}^N$ を用いて予測子のパラメータを調整する方法である。この場合まずどのような予測子を選ぶかが問題となり、次にパラメータをどの様に調整するかが問題となる。前者については、例えば feedforward neural net (perceptron)、RBF (radial basis function) 等よく知られたいくつかの手法があり、パラメータの調整が上手く行なわれればこれらの予測子間の能力差はそれ程ないというのが一般的認識と思われる。従って問題はいかにしてパラメータ調整を行なうかであり、これに関しては膨大な量の仕事がある。

小文の目的は前者として 3層 feedforward neural net (3層 perceptron) を用い、後者に Bayes 統計的枠組みから、パラメータの周辺尤度 ("Evidence") を最大化する手法を用い、次の時系列予測を行なう事である：

(i) 問題1に関しては、ASHRAE 時系列予測問題

(ii) 問題2に関しては、Lorenz 系のカオス的な時系列予測問題

Feedforward neural net の Bayes 的枠組みは MacKay による [5],[6]。カオス的時系列予測を neural net で行なう試みは多数あり [1],[2],[4],[7],[8],[15],[21]、時系列データ学習から分岐図の再構成を行なう試み [14],[19] もある。

2 非線形予測子

ここでは良く知られた 3層 feedforward neural net （3層 perceptron）を用いる:

$$u_i = \boldsymbol{w}_i^T \boldsymbol{x} + p_i\ ,\quad z_i = \sigma(u_i)\ ,\quad y = \sum_{i=1}^{H} a_i z_i + q \tag{2.1}$$

$$\sigma(v) = 1/(1+e^{-v})\ ,\quad \boldsymbol{x}, \boldsymbol{w}_i \in \mathbb{R}^n,\ p_i, q \in \mathbb{R}$$

素子と素子の間には各々、結合の度合を示す weight が与えられている。1 つの中間層 u_i にはその weight を結合係数とする入力の線形結合 $\boldsymbol{w}_i^T\boldsymbol{x}$ とバイアス p_i の和が入ってくる。この入力に sigmoid 関数 $\sigma(v) = 1/(1+e^{-v})$ を施したものが出力として与えられ、最終的にネットワークの出力 y は、中間層の出力の線形結合 $\sum_{i=1}^{H} a_i z_i$ とバイアス q の和で与えられる。勿論これ以外の予測子、例えば RBF (radial basis function) 等が考えられるが、ここで perceptron を用いる理由は身近に code があったこと、広く用いられていること、そして近似能力が理論的に示されていることによる。即ち任意の連続関数

$$f : \text{compact} \subset \mathbb{R}^n \longrightarrow \mathbb{R}$$

は、3 層 perceptron により任意の精度で近似可能であることが知られている [3]。

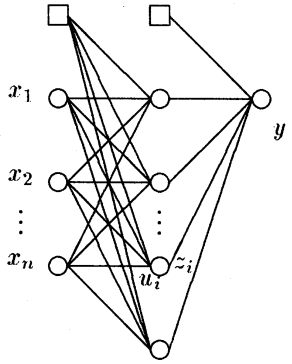


図 2.1 feedforward neural net

一般に入力 $\boldsymbol{x}_t$ と出力 y_t の "学習データ" $\{\boldsymbol{x}_t, y_t\}_{t=1}^N$ が与えられたとき、3 層 perceptron のパラメータを調整し、

$$E_D := \frac{1}{2}\sum_{t=1}^{N}[\, y_t - \{\sum_{i=1}^{H} a_i\, \sigma(\boldsymbol{w}_i^T\boldsymbol{x}_m + p_i) + q\}]^2 \tag{2.2}$$

を最小化する手法が広く用いられている。これは単純で良い方法ではあるが、しばしば overfit が起こる。例えば図 2.1 のような 3 次関数

$$y_t = \frac{3}{8}x_t(2x_t - 3)(2x_t + 3) \tag{2.3}$$

に Gaussian noise $N(0,1)$ を加えてつくった 4 0 点に対して上記学習を行なったところ、図 2.2 の様な overfit が起きた。この様な傾向は多項式近似の場合に更に深刻である。図 2.3 は式 (2.3) の次数を知らず、15 次の多項式で

$$\frac{1}{2}\sum_{t=1}^{N}[\, y_t - \sum_{i=0}^{15} a_i x_t^i]^2$$

の最小化により得た解であり、極端な overfit が起きている。

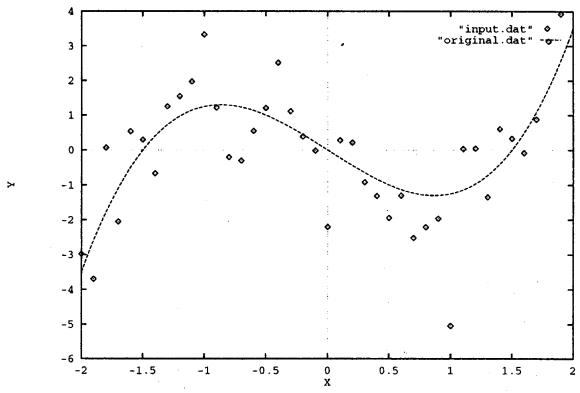


図 2.1 3 次関数 $y = \frac{3}{8}x(2x-3)(2x+3)$ に $N(0,1)$ の Gaussian noise を加えて作ったデータ 4 0 点

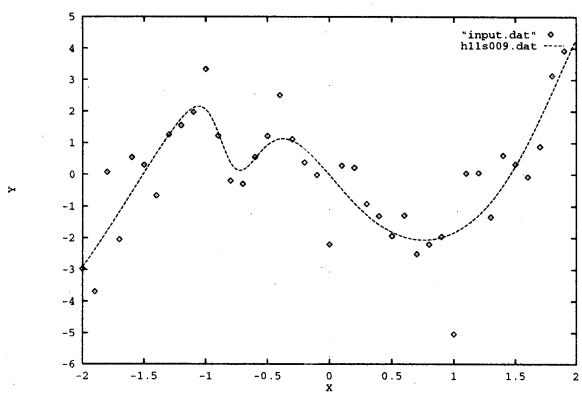


図 2.2 典型的な overfit. $H = 11$

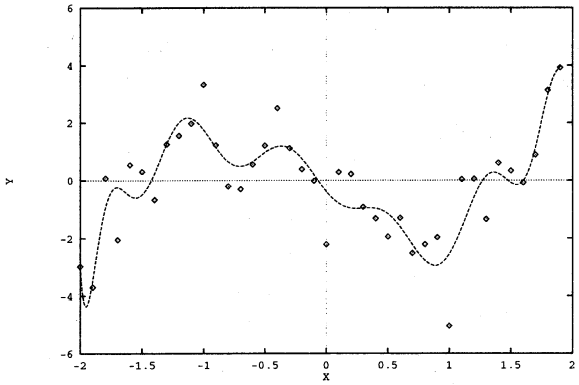


図 2.3 多項式による予測

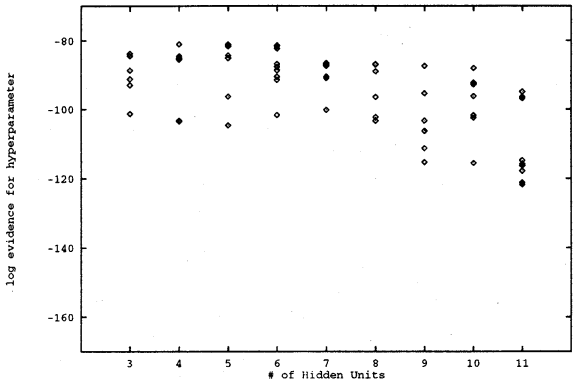


図 2.4 log Evidence v.s. Hidden Units のグラフ

このような問題点に対し、何らかの統計的手法が必要と思われる。以下に述べる Bayesian Backpropagation では Evidence（周辺尤度）を最大化することで overfit を防ぐ。また feedforward neural net では hidden unit 数 H の決定も重要問題のひとつであり、Evidence による評価も可能である。図 2.5 に H と Evidence の対数のグラフを示す。複数プロットがある理由は weight の初期値を複数用意したためである。Evidence が最大 (Maximum Evidence) のモデルを選択することにより（この場合 $H = 6$）、この近似問題に対して図 2.5 の予測結果を得ることができた。

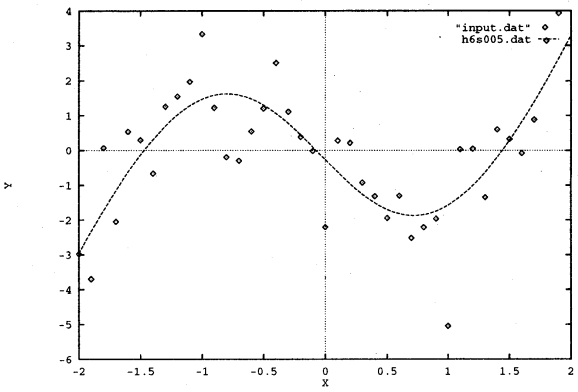


図 2.5 Maximum Evidence 予測

3 Bayesian Backpropagation

1. 前節で述べた 3 層 feedforward neural net の hidden 素子数 H を fix し、この neural net の構造を $\mathcal{H}$ (Hypothesis) と書く。モデルとも呼ばれる。

2. 関数 $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ から データ $D = \{\boldsymbol{x}_t, y_t\}_{t=1}^N$ が生成されるプロセスを

$$y_t = f(\boldsymbol{x}_t) + \nu_t \tag{3.1}$$

と仮定する。ν は Gaussian noise $N(0, \frac{1}{\beta})$、β は noise 分散の逆数で勿論 未知 でありノイズレベルと呼ぶ。$\{(a_i, \boldsymbol{w}_i, p_i)_{i=1}^H, q\}$ をまとめて $\boldsymbol{w}$ と書き、$\boldsymbol{w}$ の **Likelihood**、即ちデータ $\{y_t\}_{t=1}^N$ の条件付確率を次で定義する：

$$\begin{aligned} P(D \mid \boldsymbol{w}, \beta, \mathcal{H}) &:= P(\{y_t\}_{t=1}^N \mid \{\boldsymbol{x}_t\}_{t=1}^N, \boldsymbol{w}, \beta, \mathcal{H}) \\ &= \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \exp\left(-\beta E_D\right) \end{aligned} \tag{3.2}$$

$$E_D := \frac{1}{2} \sum_{t=1}^{N} \mid y_t - f(\boldsymbol{w}; \boldsymbol{x}_t)|^2$$

但し、$f(\boldsymbol{w};\boldsymbol{x}_t)$ はパラメータ $\boldsymbol{w}$ を与えたときの neural net の出力である。

3. $\boldsymbol{w}$ を G 個のグループに分割し、$\boldsymbol{w}=(\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_G)$, $\boldsymbol{w}_g\in\mathbb{R}^{k_g}$ とする。$\boldsymbol{w}$ の **Prior**(先験的情報) として

$$P(\boldsymbol{w}\mid\boldsymbol{\alpha},\mathcal{H})=\prod_{g=1}^{G}\left(\frac{\alpha_g}{2\pi}\right)^{\frac{k_g}{2}}\exp\left\{-\sum_{g=1}^{G}\alpha_g E_{W_g}\right\} \tag{3.3}$$

を考える。但し、

$$\boldsymbol{\alpha}=(\alpha_1,\cdots,\alpha_G)\ ,\quad E_{W_g}:=\frac{1}{2}\|\boldsymbol{w}_g\|^2 \tag{3.4}$$

である。即ち、$\boldsymbol{w}_g$ の prior :

$$N(0,\tfrac{1}{\alpha_g})\ ,\quad g=1,\cdots,G,\ \ \alpha_g\text{: 互いに独立}$$

勿論 α_g, $g=1,\cdots,G$ は 未知 である。

Remarks

(1). 式 (3.2) を $\boldsymbol{w}$ に関して最大化すると Maximum Likelihood 解が得られ、これは前節で述べた様にしばしば overfit を起こす。式 (3.4) の様ないわゆる weight decay 項あるいは regularizer を考慮すると overfit を防ぐことができ、これはかなり広く用いられている。問題は regularizer をどの程度強くするか、即ち $\alpha_1,\cdots,\alpha_G$ をどの様に決定するか であって、これはしばしば *ad hoc* である。以下で述べる手法では、$\alpha_1,\cdots,\alpha_G$ 及び β の最適値が客観的評価基準のもとで決定可能である。

(2). $\boldsymbol{w}$ をグループに分ける理由は、入力から中間層への weight に対する weight decay の度合と、中間層から出力への weight decay の度合は異なり得るし、また $\boldsymbol{x}_t$ がベクトルの場合、その成分ごとにも weight decay の度合が異なり得るからである。

(3). 式 (3.3) で与えられる prior は、weight の大きさが小さいものが多く大きいものが少ないことを意味しており、比較的 reasonable な場合が多い。

以上から、Bayes 公式を用いると次を得る：

Level 1: $\boldsymbol{w}$ の Posterior:

$$P(\boldsymbol{w}\mid D,\boldsymbol{\alpha},\beta,\mathcal{H})=\frac{P(D\mid\boldsymbol{w},\beta,\mathcal{H})P(\boldsymbol{w}\mid\boldsymbol{\alpha},\mathcal{H})}{P(D\mid\boldsymbol{\alpha},\beta,\mathcal{H})}=\frac{exp(-M)}{\int exp(-M)d\boldsymbol{w}}$$

$$M(\boldsymbol{w}):=\beta E_D+\sum_g\alpha_g E_{W_g}$$

Level 2: $\boldsymbol{\alpha},\beta$ の Posterior:

$$P(\boldsymbol{\alpha},\beta\mid D,\mathcal{H})=\frac{P(D\mid\boldsymbol{\alpha},\beta,\mathcal{H})P(\boldsymbol{\alpha},\beta\mid\mathcal{H})}{P(D\mid\mathcal{H})}$$

Level 3: Model 比較:

$$P(\mathcal{H}\mid D)\propto P(D\mid\mathcal{H})P(\mathcal{H})$$

Remarks

(1). Level 2 の (marginal) likelihood は Level 1 の 規格化定数に対応し、Level 3 の (marginal) likelihood は Level 2 の規格化定数に対応する階層構造をしている：

Level 2 の likelihood:

$P(D \mid \boldsymbol{\alpha}, \beta, \mathcal{H})$ = Level 1 の規格化定数 $\longrightarrow$ "Evidence for $\boldsymbol{\alpha}, \beta$"

Level 3 の likelihood:

$P(D \mid \mathcal{H})$ = Level 2 の規格化定数 $\longrightarrow$ "Evidence for model"

これらを図式的に書くと図 3.1 の様になる。

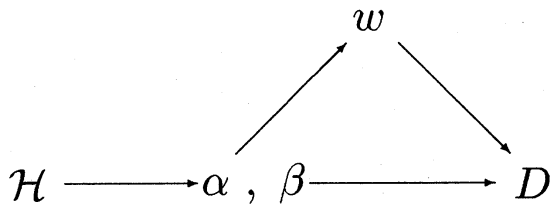


図 3.1 Bayesian Backprop. の階層構造

(2). 各 level ごとに次の形をしている：

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

各 level でパラメータ、あるいは H を変えて最大化を行なうことにより、最適解が得られる：

Level 1. $\quad \boldsymbol{w}_{MP} := arg\ min\ M(\boldsymbol{w})$

Level 2. $\quad (\boldsymbol{\alpha}_{MP}, \beta_{MP}) := arg\ max\ P(D \mid \boldsymbol{\alpha}, \beta, \mathcal{H})$

Level 3. $\quad \mathcal{H} := arg\ max\ P(D \mid \mathcal{H})$

Level 1 の最適化は適当なアルゴリズム、例えば Conjugate Gradient でそのまま計算可能であるが Level 2 の最適化には近似計算が必要とされる。比較的うまくいく手法は $\log P(D \mid \boldsymbol{\alpha}, \beta, \mathcal{H})$ の 2 次近似である：

$$\log P(D \mid \boldsymbol{\alpha}, \beta, \mathcal{H}) \approx \frac{N}{2} \log \frac{\beta}{2\pi} - \beta E_D^{MP} - \sum_g \alpha_g E_{W_g}^{MP} - \frac{1}{2} \log \det \boldsymbol{A} + \sum_g \frac{k_g}{2} \log \alpha_g \tag{3.5}$$

但し MP は (1) で得られた $\boldsymbol{w}_{MP}$ での評価、$\boldsymbol{A}$ は M の Hessian：

$$\boldsymbol{A} := \nabla\nabla M \ = \ \nabla\nabla \left\{ \beta E_D + \sum_g \alpha_g E_{W_g} \right\}$$

の $\boldsymbol{w}_{MP}$ における値である。即ち

$$P(D \mid \boldsymbol{\alpha}, \beta, \mathcal{H}) \approx \frac{1}{(\frac{2\pi}{\beta})^{\frac{N}{2}}} \exp(-\beta E_D^{MP}) \ \frac{1}{\prod_g (\frac{2\pi}{\alpha_g})^{\frac{k_g}{2}}} \exp(-\sum_g \alpha_g E_{W\,g}) \frac{(2\pi)^{\frac{k}{2}}}{\sqrt{\det \boldsymbol{A}}}$$

であり、第一 factor：

$$\frac{1}{(\frac{2\pi}{\beta})^{\frac{N}{2}}} \exp(-\beta E_D^{MP}) = \text{best fit likelihood}$$

一方、第二 factor：

$$\frac{1}{\prod_g (\frac{2\pi}{\alpha_g})^{\frac{k_g}{2}}} \exp(-\sum_g \alpha_g E_{W\,g}) \frac{(2\pi)^{\frac{k}{2}}}{\sqrt{\det \boldsymbol{A}}} = \text{Occam factor（補正項）}$$

であって、これはデータ $\{y_t\}_{t=1}^N$ を観測した時に得られる information gain である。prior の鋭さ $\{\alpha_g\}$ と、posterior の緩やかさ $\sqrt{\det \boldsymbol{A}}$ の兼ね合いになっており、後者の緩やかさは大きな Occam factor に対応することに注意すべきである。

式 (3.5) で、例えば β に関する振る舞いを直観的に説明すると、第 1 項と第 3 項に負符号がついている一方、第 4 項は正符号なので丁度よい β_{MP} がある。図 3.2 は前節の 3 次関数の場合の $\log P(D \mid \boldsymbol{\alpha}_{MP}, \beta, \mathcal{H})$ のプロットであり、β_{MP} はノイズレベルをほぼ正しく推定していることが分かる。このような山は **Occam Hill** と呼ばれている。α_g に関しても同様である。

hidden unit 数 H については、必ずしも美しい Occam Hill が見えるか否か明らかではないが、H を決定する目安になることは例えば前節図 2.4 から分かる。

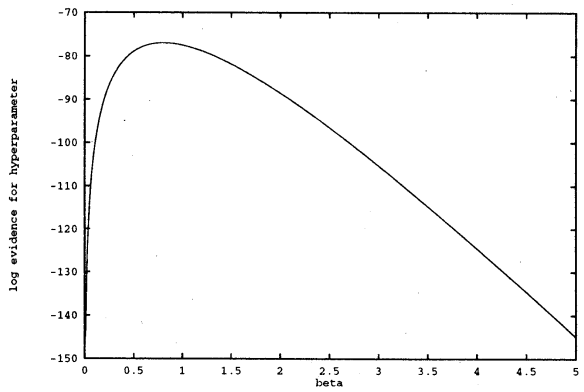


図 3.2 $\log P(D \mid \boldsymbol{\alpha}_{MP}, \beta, \mathcal{H})$ v.s. β のグラフ

式 (3.5) の $\boldsymbol{\alpha}, \beta$ に関する最大化を gradient 型のアルゴリズムで行なうためには微系数が必要になる：

$$\frac{\partial}{\partial \alpha_g} \log P(D \mid \boldsymbol{\alpha}, \beta, \mathcal{H}) \approx -E_{W_g} + \frac{1}{2\alpha_g}\left(k_g - \alpha_g \mathrm{Tr}\boldsymbol{A}^{-1}\boldsymbol{I}_g\right) \tag{3.6}$$

$$\frac{\partial}{\partial \beta} \log P(D \mid \boldsymbol{\alpha}, \beta, \mathcal{H}) \approx -E_D - \frac{1}{2\beta}\sum_g \left(k_g - \alpha_g \mathrm{Tr}\boldsymbol{A}^{-1}\boldsymbol{I}_g\right) + \frac{N}{2\beta} \tag{3.7}$$

但し Tr は行列の Trace を意味し、

$$(\boldsymbol{I}_g)_{ij} = \begin{cases} 1, & i = j,\ w_i \in \text{Group } g \\ 0, & \text{otherwise} \end{cases}$$

である。

Level 3 でも 2 次近似を用いると次を得る：

$$\begin{aligned} \log P(D|\mathcal{H}) \approx\ & \frac{N}{2}\log\frac{\beta^{MP}}{2\pi} - \beta^{MP}E_D^{MP} - \sum_g \alpha_g^{MP}E_{W_g}^{MP} - \frac{1}{2}\log\det \boldsymbol{A} + \sum_g \frac{k_g}{2}\log\alpha_g^{MP} \\ & + \frac{G+1}{2}\log 2\pi + \frac{1}{2}\log\frac{2}{N-\gamma} + \frac{1}{2}\sum_g \log\frac{2}{\gamma_g} \end{aligned} \tag{3.8}$$

但し

$$\gamma = \sum_g \gamma_g\ ,\quad \gamma_g = k_g - \alpha_g \mathrm{Tr}\boldsymbol{A}^{-1}\boldsymbol{I}_g$$

である。

公式 (3.5)~(3.8) が如何に導出されるかを簡単に述べる。まず

$$P(D \mid \boldsymbol{\alpha}, \beta, \mathcal{H}) = \frac{1}{(\frac{2\pi}{\beta})^{\frac{N}{2}}\prod_{g=1}^{G}(\frac{2\pi}{\alpha_g})^{\frac{k_g}{2}}}\int \exp(-M(\boldsymbol{w}))d\boldsymbol{w}$$

第 2 factor の積分は一般に解析的表現は困難なので $M(\boldsymbol{w})$ の 2 次近似を考える：

$$M(\boldsymbol{w}) \approx M(\boldsymbol{w}_{MP}) + \frac{1}{2}(\boldsymbol{w}-\boldsymbol{w}_{MP})^T \boldsymbol{A}\,(\boldsymbol{w}-\boldsymbol{w}_{MP})$$

この時

$$\int \exp(-M(\boldsymbol{w}))d\boldsymbol{w} \approx \exp(-M(\boldsymbol{w}_{MP}))(2\pi)^{\frac{k}{2}}\det{}^{-\frac{1}{2}}\boldsymbol{A}$$

なので、式 (3.5) を得る。

式 (3.6),(3.7) は、式 (3.5) を α_g 及び β に関して微分することにより得られる。式 (3.6) 第 1 項は式 (3.5) 第 2 項の、式 (3.6) 第 2 項は式 (3.5) 第 5 項の α_g に関する微分である。式 (3.5) 第 3 項の微分は

$$\frac{\partial}{\partial\alpha_g}(-\frac{1}{2}\log\det\boldsymbol{A}) = -\frac{1}{2}\frac{\partial}{\partial\alpha_g}\mathrm{Tr}\log\boldsymbol{A}$$

$$= -\frac{1}{2}\mathrm{Tr}(\boldsymbol{A}^{-1}\frac{\partial\boldsymbol{A}}{\partial\alpha_g}) = -\frac{1}{2}\mathrm{Tr}(\boldsymbol{A}^{-1}\boldsymbol{I}_g)$$

となり、これが式 (3.6) 第 3 項になっている。式 (3.7) も同様である。式 (3.8) は、$\boldsymbol{\alpha},\beta$ に関する Hessian を計算することにより求まる。

4 応用

4.1 ASHRAE Time Series Prediction

昨秋、米国 ASHRAE は電力消費予測コンテストを開催した。我々も参加したので、その概略を述べる。参加者には、一定期間の 1 時間毎の米国のある場所（未知）における気象データ（温度、湿度、太陽光線量、風速）と、そこに建っているビルの 1 時間毎の電力使用量（総電力、モーター制御室、照明）、冷水使用量、温水使用量のデータが与えられる。与えられたデータには人為的に電力、冷・温水使用量が隠された部分が含まれており、参加者は自由にアルゴリズムを考えその隠された部分を予測し精度を競う。

我々は

$$\boldsymbol{x}_t := (\text{総電力使用量、モーター制御電力量、照明用電力量、冷水使用量、温水使用量})_t \in \mathbb{R}^5$$

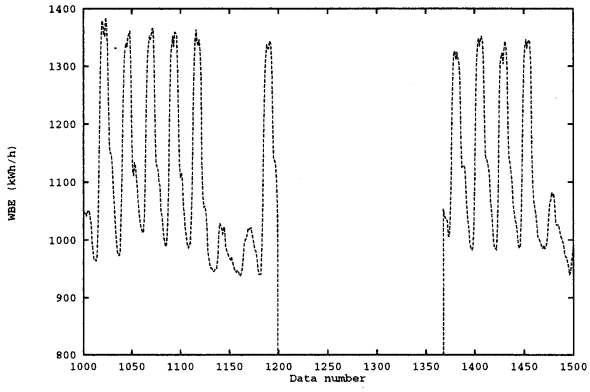
$$\boldsymbol{u}_t := (\text{温度、太陽光線量、湿度、風速、}\ g_t\ )_t \in \mathbb{R}^n$$

但し g_t は日付、時間、季節などから定義される（ベクトル）関数とし、data $D := \{\boldsymbol{x}_t;\boldsymbol{u}_t\}_t$ を用いて学習を行なって $\boldsymbol{f}(\boldsymbol{x}_t;\boldsymbol{u}_t)$ を構成し $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t;\boldsymbol{u}_t)$ で予測を行なうことを考えた。今回は時間的制約が厳しく残念ながら $\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{u}_t)$ とした。

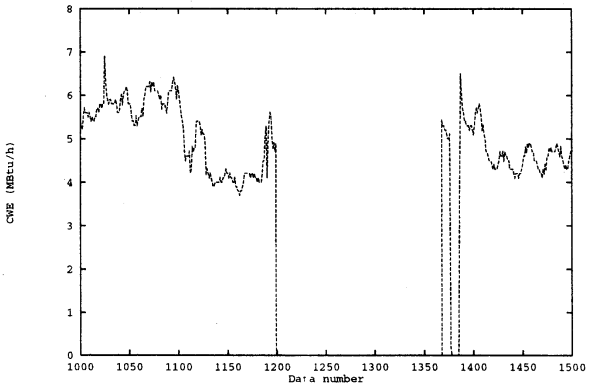
図 4.1 (a), (b) は WBE（総電力使用量）、CWE（冷水使用量）、そして HWE（温水使用量）の学習データの一部である。まず全ての変数を $[0,1]$ 区間に規格化し、電力データについては moving average をとった上で trend を除去し残差を 3 層 feedforward neural net に入力した。各入力にはひとつの α_g が対応し、それ以外は中間層から出力への weight、及び bias 項 $\{p_i\}_{i=1}^H, q$ に対して各々 α_g を割り当てた。結果は図 4.1 (c), (d) に示す。コンテストは中央部の予測精度で競われた。具体的には Coefficient of Variation of the Root Mean Square Error (CV-RMSE) 及び Mean Bias Error により評価された：

$$\mathrm{CV-RMSE}[\%] = \frac{\sqrt{\frac{\sum_{i=1}^n (y_{pred,i}-y_{data,i})^2}{n-1}}}{\bar{y}_{data}} \times 100\,, \qquad \mathrm{MBE}[\%] = \frac{\frac{\sum_{i=1}^n (y_{pred,i}-y_{data,i})}{n-1}}{\bar{y}_{data}} \times 100$$

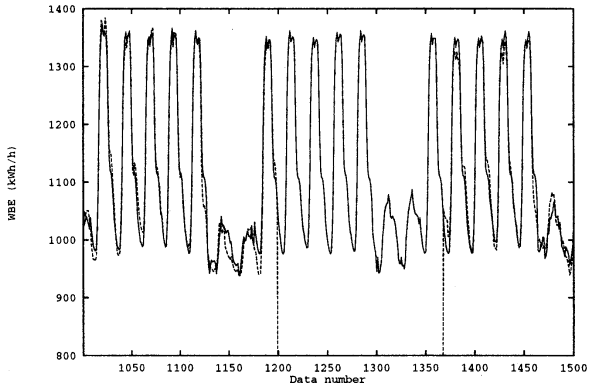
ただし $y_{data,i}$ は隠された部分の真の電力、冷・温水消費量、$y_{pred,i}$ はその部分に対する予測値、$\bar{y}_{data}$ は data set の $y_{data,i}$ の平均値、n は data set に含まれる data の数である。詳細は [11] に述べられている。主催者は CV のみで結果を評価し、我々の結果は参加 4 7 グループ中 3 位であった。MBE では我々が 1 位であった。



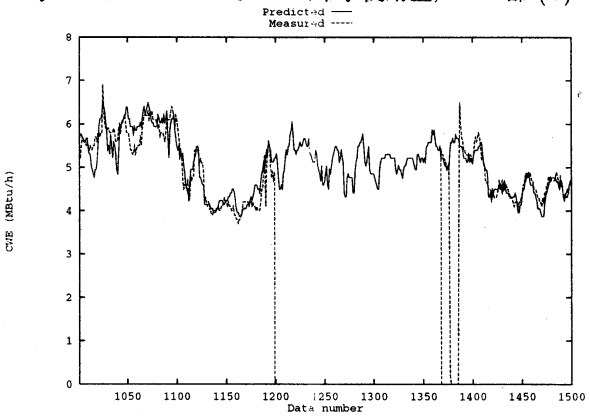
与えられたデータ WBE (総電力使用量) の一部 (a)



与えられたデータ CBE (冷水使用量) の一部 (b)



上図 (a) に対する neural net の予測波形 (c)



上図 (b) に対する neural net の予測波形 (d)

図 4.1 与えられたデータとその予測波形

破線で示されているのが、与えられたデータ。実線で表されているのが neural network の予測波形。中間層の素子数は 5 から 1 2 までの中で、Evidence の高いモデルを選んでいる。

4.2 Lorenz 系

Lorenz 系

$$\begin{aligned} \frac{dx}{dt} &= \sigma(x-y) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= -bz + xy \end{aligned}$$

でよく知られたパラメータ値 $(r, \sigma, b) = (28, 10, 8/3)$ とし、4 次 Runge-Kutta 法によりきざみ幅 0.001 で解いて x 成分を 0.02 毎に 1000 点取り出して 学習データとした (図 4.2)。これから delay coordinate system

$$(x_t, x_{t-\tau}, \cdots, x_{t-(m+1)\tau})$$

をつくり、これに基づいて学習した 3 層 feedforward neural net により

$$x_{t+\tau} = f(\boldsymbol{w}; x_t, x_{t-\tau}, \cdots, x_{t-(m+1)\tau})$$

を用いて予測を行なう。$(r, \sigma, b) = (28, 10, 8/3)$ で Lorenz 系はカオス的振る舞いを示しており、長期予測は不可能であるので短期予測を検討する。

この種の予測問題では

(i). 1-step 予測：$x_t, x_{t-\tau}, \cdots, x_{t-(m+1)\tau}$ が与えられたとき $x_{t+\tau}$ を予測する

(ii). K-step 予測：$x_t, x_{t-\tau}, \cdots, x_{t-(m+1)\tau}$ が与えられたとき $x_{t+\tau}, \cdots, x_{t+K\tau}$ を予測する

のふたつの場合があるが [7]、ここで扱うのは後者である。

τ, m の決定は重要課題で多くの手法があるが、今回はこれを決定することが主目的ではないので $\tau = 0.06$, $m = 3$ とした。Takens の定理の $2d+1$ は十分条件であり、m がそれ未満であっても元の系の性質が保存されることはいくらでもある。我々の知る限り大半の著者は $m < 2d+1$ で delay coordinate system を用いている。

図 4.3 に予測子の構造を示す。ハイパーパラメータ $\alpha_1, \alpha_2, \alpha_3$ は各々の入力 $x_t, x_{t-\tau}, x_{t-2\tau}$ に対応し、α_4 は中間層から出力へ、α_5 は bias のハイパーパラメータである。

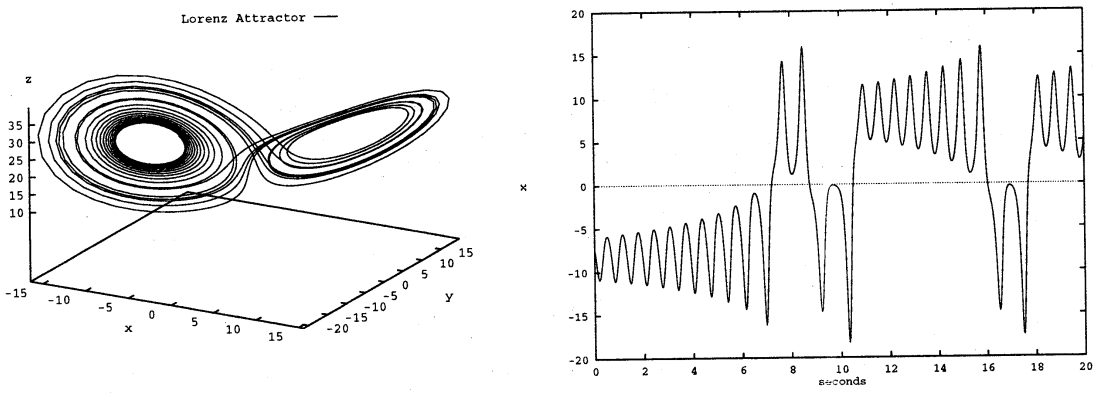


図 4.2 Lorenz Attractor (a) と学習に用いた x 時系列波形 (b)

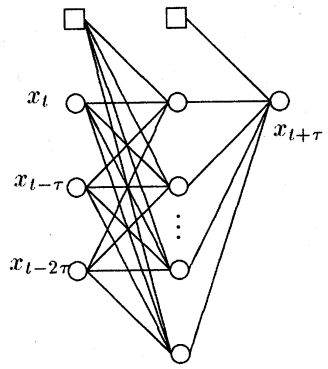


図 4.3 Lorenz 系の予測に用いた予測子の構造

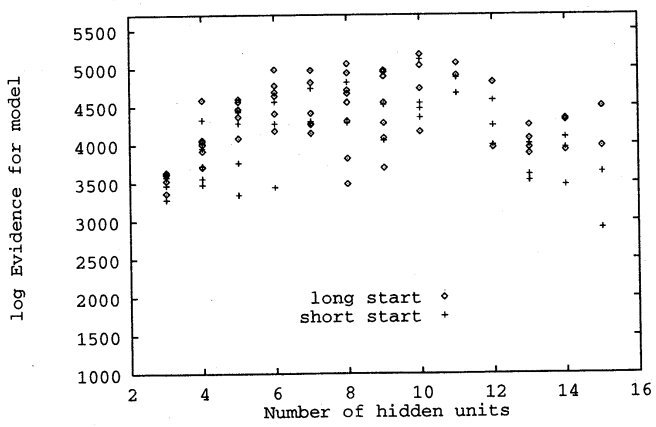
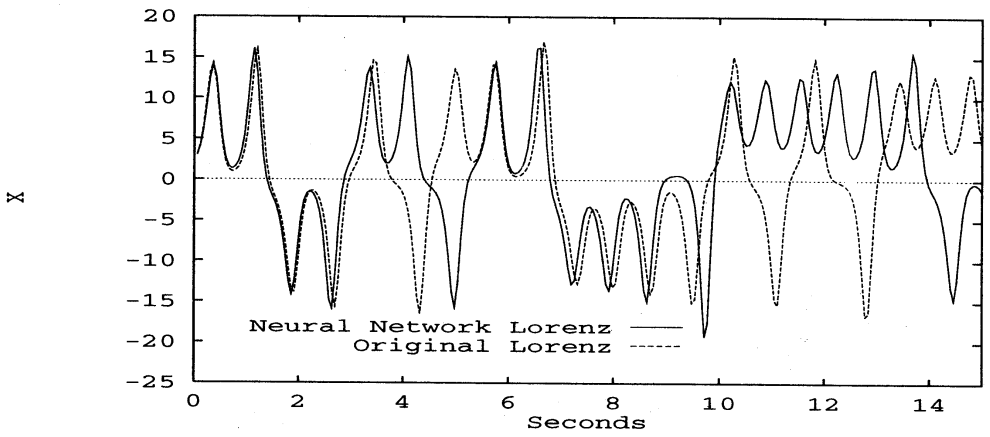
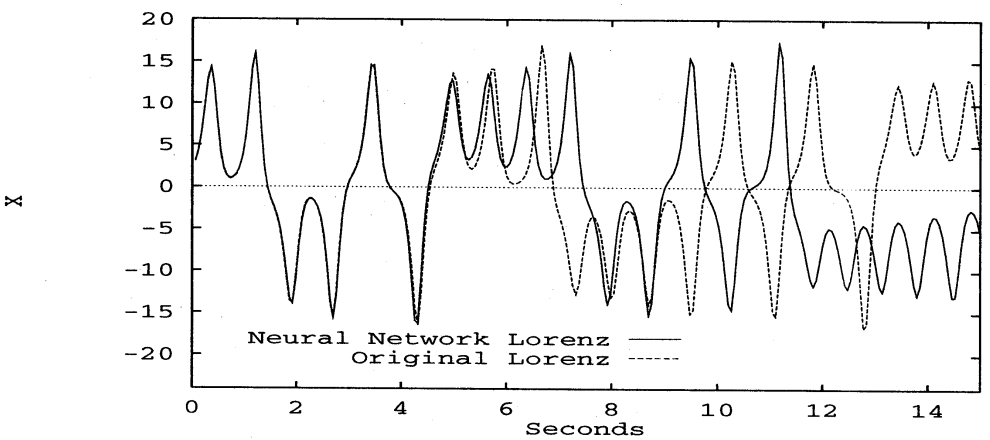


図 4.4 中間層の素子数 H に対する Evidence for model

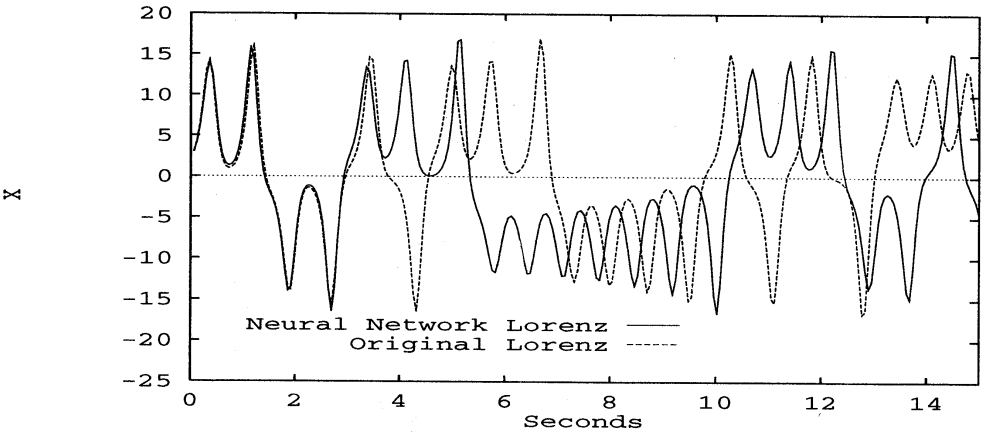
図 4.4 に $\log P(D \mid \mathcal{H})$ を H の関数としてプロットした。long start は weight の初期値の分散を 0.1 にして計算を行なったもの、short start は weight の初期値の分散を 0.05 にして計算したもので、この図では中間層の素子数が１０の時に Evidence が最大になり most probable なモデルと考えられる。図 4.5 は学習データにない初期値から出発した 3 つの "neural net Lorenz" 時系列（実線）と、Runge-Kutta によるもの（破線）との比較 (K=250 step) を示したものである。Evidence の高いものが優れた予測能力をもつことが分かる。



(a) 250 step 予測 (Hidden 4, Evidence = 4585)



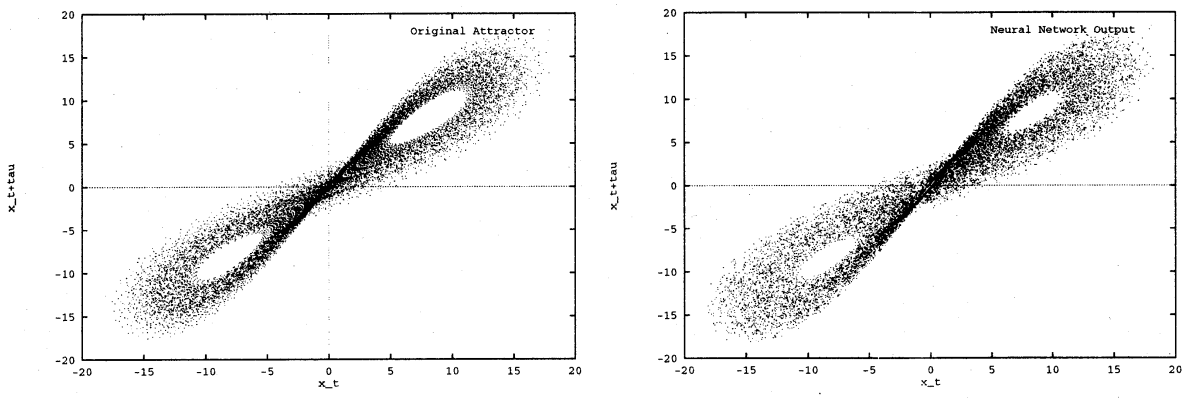
(b) 250 step 予測 (Hidden 10, Evidence = 5179)



(c) 250 step 予測 (Hidden 14, Evidence = 4337)

図 4.5 予測時系列 (250 step) の比較

この予測時系列 8192 点を $(x_t, x_{t+\tau})$ 空間にプロットし、original (Runge-Kutta 法で解いた x 時系列) と比べたのが図 4.6 である。



(a) Original Lorenz　　(b) Neural Net Lorenz

図 4.6 $(x_t, x_{t+\tau})$ 空間のアトラクタ比較

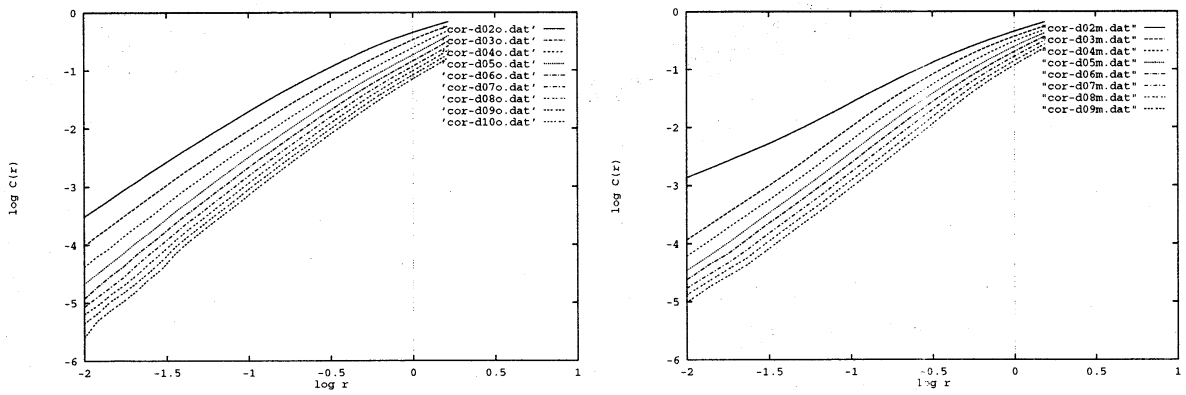
この neural net Lorenz と、元の力学系とのいくつかの定量的性質を比較してみる。

4.2.1 最大 Lyapunov 指数

表 1: 最大 Lyapunov 指数の比較

	最大 Lyapunov 指数
Original Lorenz	1.02
Neural Network Lorenz	1.05

4.2.2 相関次元



(a) Original Lorenz $(\sigma, r, b) = (10, 28, \frac{8}{3})$, $d \approx 2.05$　　(b) Neural Network Lorenz (H = 10), $d \approx 2.07$

図 4.7 相関次元の比較

4.2.3 不動点

表 2: 平衡点、不動点の比較

	Original Lorenz	Neural Network Lorenz
平衡点・不動点	(0, 0, 0)	(0.00612, 0.00612, 0.00612)
	(8.1649, 8.1649, 25)	(8.5345, 8.5345, 8.5345)
	(−8.1649, −8.1649 25)	(−8.5234, −8.5234, −8.5234)

neural net Lorenz の不動点は原点付近と原点対象の 2 点であり、original Lorenz の z 軸対象平衡点と異なっている。また

$$\begin{bmatrix} x_{t+\tau} \\ x_t \\ x_{t-\tau} \end{bmatrix} = \begin{bmatrix} f(\boldsymbol{w}; x_t, x_{t-\tau}, x_{t-2\tau}) \\ x_\tau \\ x_{t-\tau} \end{bmatrix}$$

$$:= F(\boldsymbol{w}; x_t, x_{t-\tau}, x_{t-2\tau}) \tag{4.1}$$

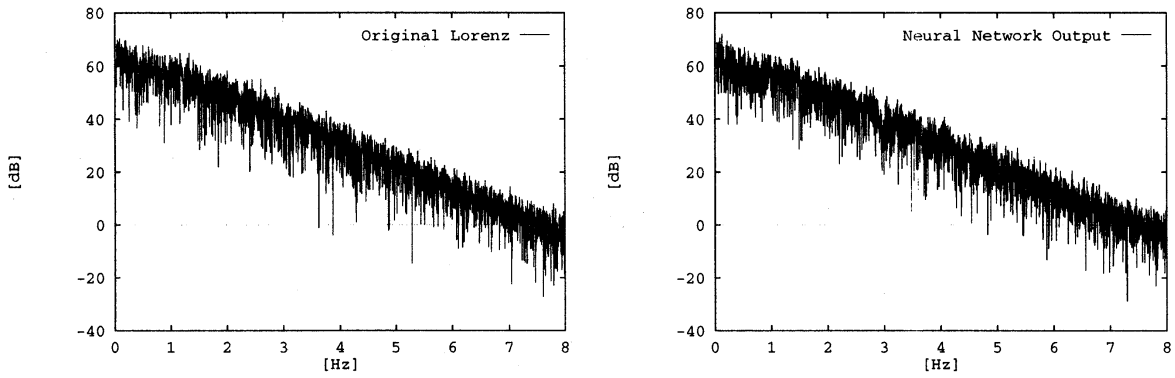
なので、不動点は必然的に

$$(x^*, x^*, x^*)\ ,\quad x^* = f(\boldsymbol{w}; x^*, x^*, x^*)$$

の形になる。

4.2.4 Power Spectrum

予測時系列 8192 点に関しての FFT をかけたものを図 4.8 に示す。



(a) Original Lorenz から Runge-Kutta で得た x 成分 8192 点の FFT

(b) Neural Network の予測値 8192 点の FFT

図 4.8 Power Spectrum の比較

4.2.5 Symmetry

Lorenz 系では z-symmetric となっているが、Neural Network Lorenz は式 (4.1) で与えられるので

$$F(\boldsymbol{w}; -x_t, -x_{t-\tau}, -x_{t-2\tau}) \neq F(\boldsymbol{w}; x_t, x_{t-\tau}, x_{t-2\tau})$$

となり、z-symmetric ではあり得ない。学習結果 $F(\boldsymbol{w}; x_t, x_{t-\tau}, x_{t-2\tau})$ は原点対称性を備えているか調べるため、いくつかの $(x_t, x_{t-\tau}, x_{t-2\tau})$ の組に対して $f(\boldsymbol{w}; x_t, x_{t-\tau}, x_{t-2\tau})$ を計算してみたのが表3である。学習された F はほぼ原点対称性を備えていると思われる。

表 3: 対称性についての計算結果

$x_{t-2\tau}$	$x_{t-\tau}$	x_t	$f(\boldsymbol{w}; \boldsymbol{X}_t)$
2.00000	3.00000	4.00000	5.33619
-2.00000	-3.00000	-4.0000	-5.33804
6.00000	9.00000	13.0000	15.4681
-6.00000	-9.00000	-13.0000	-15.4612
12.0000	7.00000	4.00000	2.97528
-12.0000	-7.00000	-4.00000	-2.97289

5 おわりに

非線形時系列予測の手法として、Maximum Evidence 法の枠組み述べ、この手法を使って ASHRAE time series data と Lorenz 系への適用を考えた。

これからの課題として

1. Nonautonomous の場合も dynamics を入れる
2. $\boldsymbol{w}, \boldsymbol{\alpha}, \beta$ の prior の検討
3. noisy Lorenz data
4. delayed coordinate system の次元 m を変えた時の evidence の変化を眺める

などが考えられる。この手法を用いて様々な問題へ適用していきたい。

謝辞　ディスカッションして頂いた、戸川 美郎氏 (東京理科大)、宇敷 重広氏 (京大)、D.J.C.MacKay 氏 (ケンブリッジ大)、竹内 亮氏、浜岸 広明氏 (早大) に感謝します。

References

[1] A.Lapedes and R.Farber (1987), "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modelling", LA-UR-87-2662, Los Alamos National Laboratory.

[2] J.P.Crutchfield and B.S.McNamara (1987), "Equations of Motion From A Data Series", Complex Systems 1 417.

[3] H.White, K.Hornik and M.Stinchcombe (1989), "Multilayer Feedforward Neural Networks Are Universal Approximators", *Neural Computation* **2**, 359-366.

[4] M.Sato et al. (1990), Learning chaotic dynamics by recurrent neural networks, Proc. Inter. Conf. on Fuzzy Logic & N.N. 601-605.

[5] D.J.C.MacKay (1992), "Bayesian interpolation", *Neural Computation* **4** 3, 415-447.

[6] D.J.C.MacKay (1992), "A practical Bayesian framework for backprop networks", *Neural Computation* **4** 3, 448-472.

[7] 合原一幸 (1993), "ニューラルシステムにおけるカオス", 東京電機大学出版, 92-124, 246-272.

[8] 小室元政 (1993), "力学系の埋め込みによる時系列予測", 平成 4 年度科研費総合 (A) 位相幾何学の総合研究集会.

[9] D.J.C.MacKay (1994), "Bayesian non-linear modelling for the prediction competition", *ASHRAE Transactions* 1994 Vol.100 PART 2.

[10] M.Iijima, K.Takagi, R.Takeuchi and T.Matsumoto (1994), "A piecewise-linear regression on the ASHRAE time-series data", *ASHRAE Transactions* 1994 Vol.100 PART 2.

[11] Y.Chonan, K.Nishida and T.Matsumoto, "A Bayesian Nonlinear Regression with Multiple Hyperparameters on the ASHRAE II Time Series Data", preprint.

[12] 高木健次, 松本隆 (1994), "ニューラルネットにおける ARD について", 電子情報通信学会秋期大会 A-32.

[13] 高木健次, 松本隆 (1994), "ARD (Automatic Relevance Determination) の M.L.P. への適用について", 日本神経回路学会第 5 回全国大会 P305.

[14] R.Tokunaga et. al. (1994), Reconstructing bifurcation diagrams only from time-waveforms, Physica D 79 348-360.

[15] T.Miyano and F.Girosi (1994), "Forecasting Global Temperature Variations by Neural Network", MIT AI Lab. Memo No.1447.

[16] 池口徹, 合原一幸, 的崎健 (1994), "時系列信号からのアトラクタ再構成について", 電気学会情報処理研究会 IP94-18.

[17] 高木健次, 松本隆 (1995), "ニューラルネットへの ARD (Automatic Relevance Determination) 適用の有効性について", 電子情報通信学会技術研究報告 NC94-61.

[18] 松本隆, 長南吉正, 浜田雅之 (1995), "Maximum Evidence 時系列予測について", 電気学会情報処理研究会 IP95-31.

[19] 梶原志保子, 徳田功, 徳永隆治, 松本隆 (1995), "時間連続系に対する分岐図再構成", 電気学会情報処理研究会 IP95-32.

[20] 吉村尚郎, 松本隆 (1995), "Evidence による Feedforward Neural Network の評価", 日本神経回路学会第 6 回全国大会 P1-5.

[21] 宮野尚哉, 柴富田浩, 中島研, 池永泰治, "動径基底関数ネットワークによる高炉状態", 日本神経回路学会第 6 回全国大会 P1-32.

[22] 長南吉正, 浜田雅之, 松本隆, "Feedforward Neural Net による Maximum Evidence 時系列予測", 日本神経回路学会第 6 回全国大会 P2-29.