

# ニューラルネットワークによる微分方程式の解法 (計算経済学の研究その26)

## Solving Differential Equations Using Neural Networks

釜 国男<sup>\*</sup>  
Kunio KAMA

これまで解析解のない微分方程式にはルンゲ・クッタ法など数値解法を適用してきた。数値的な方法によってかなり正確な解が得られるが、問題によっては初期条件や境界条件を考慮するのは簡単ではない。また離散的な解しか得られないので、サンプル点以外は内挿を行う必要がある。このような問題は以前からわかっていたが、最近、機械学習の一環としてニューラルネットワーク（以後、NN と略する）を用いた新しいアプローチが提案されている<sup>1)</sup>。Cybenko (1989), Funahashi (1989), Hornik et al. (1989) の関数近似の定理によると、簡単な構造の NN はあらゆる関数を任意の精度で近似可能である。この性質を利用すれば、NN を用いて微分方程式の解を数値的に求めることができる。本稿ではいくつかの常微分方程式と偏微分方程式を取りあげて、このアプローチはきわめて有効であることを示す。

### 1. 1 階常微分方程式

つぎの 1 階常微分方程式について考える。

$$\frac{d\Psi(x)}{dx} = f(x, \Psi) \quad (1)$$

ここで  $\Psi(x)$  は方程式の解であり  $x \in [0, 1]$  とする。初期条件として、 $\Psi(0) = A$  が与えられているものとする。(1) の解をつぎの形の関数で近似する。

$$\Psi_i(x, p) = A + xN(x, p) \quad (2)$$

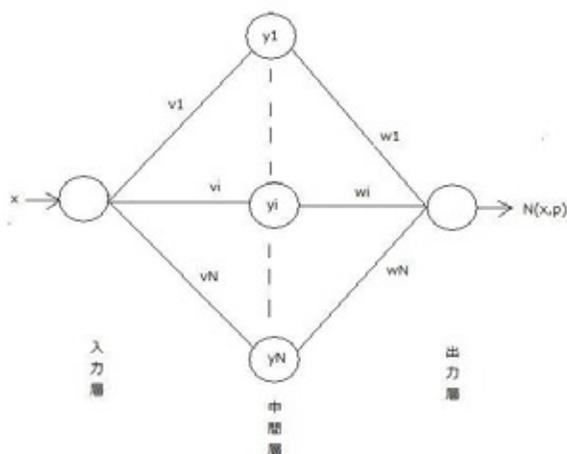
ここで  $N(x, p)$  はパラメータ  $p$  をもつフィードフォワードネットワークである。 $\Psi_i(0, p) = A$  であり、試行関数は初期条件を満たしている。(2) の形で初期条件を組み込む方法は形状因子法とよばれる。(1) を離散化して

$$\frac{d\Psi(x_i)}{dx} = f(x_i, \Psi(x_i)), \quad i=1, 2, \dots, n \quad (3)$$

---

<sup>\*</sup> 創価大学名誉教授

図1 ニューラルネットワーク



誤差関数を

$$E(p) = \sum_{i=1}^n \left\{ \frac{d\Psi_i(x_i)}{dx} - f(x_i, \Psi_i(x_i)) \right\}^2 \quad (4)$$

として最小化する。ここで、 $d\Psi_i(x)/dx = N(x, p) + x dN(x, p)/dx$  である。(1)の数值解は  $\Psi_i(x, p^*)$ ,  $p^* = \arg \min_p E(p)$  で与えられる。誤差関数の最小化には準ニュートン法を用いた。

図1はネットワークの構造を示している。入力層と出力層は1個のニューロンからなり、中間層にはN個のニューロンを配した。入力層にインプットされた  $x$  は中間層で処理されて出力層に入り最終的な処理を受ける。入力と出力の関係はつぎのように表される。

$$N(x, p) = \sum_{i=1}^N w_i y_i \quad (5)$$

$$y_i = f(v_i x + \theta_i) \quad (6)$$

ここで  $v_i$  と  $w_i$  は入力-中間層、および中間-出力層を結ぶ結合係数であり、 $\theta_i$  は中間層の閾値である。係数ベクトルを、 $p = (v_1, \dots, v_N, \theta_1, \dots, \theta_N, w_1, \dots, w_N)$  と記す。(6)の関数  $f(\cdot)$  は活性化関数である。様々な関数が提案されているが、ここではシグモイド関数

$$f(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

を用いる。これは  $y=0$  と  $y=1$  を漸近線とする単調増加関数であり、 $\lim_{x \rightarrow \infty} f(x) = 1$ ,  $\lim_{x \rightarrow -\infty} f(x) = 0$ ,  $f'(x) = f(x)\{1 - f(x)\}$  となる<sup>2)</sup>。中間層のニューロンは10個として、結合係数は全部で30となる。中間層を2層としたりニューロン数を増やすと近似精度は高くなるが、誤差関数の収束解を得るのは難しくなる。

以上の方法でつぎの常微分方程式を解いてみよう。

$$\frac{d\Psi}{dx} = e^{-2x} - 2\Psi \tag{8}$$

$$0 \leq x \leq 2, \Psi(0) = 0.1$$

解析解があり、 $\Psi_a(x) = 0.1e^{-2x} + xe^{-2x}$  となる。数値計算では  $x$  の値域を 50 等分して、試行解を  $\Psi_t(x) = 0.1 + xN(x, p)$  とする。なお、 $\Psi_t(0) = 0.1$  であり試行解は初期条件を満たしている。計算の結果、結合係数と閾値はつぎのようになる。

$$\begin{aligned} y_1 &= f(0.1099x - 0.1930), y_2 = f(-4.2244x - 1.5336), y_3 = f(-0.0547x + 0.0148), \\ y_4 &= f(-0.0936x - 0.0903), y_5 = f(-0.5312x + 0.2153), y_6 = f(0.0802x + 0.2784), \\ y_7 &= f(0.1848x + 0.3098), y_8 = f(-2.2013x - 0.3327), y_9 = f(-0.1715x - 0.3663), \\ y_{10} &= f(0.0047x + 0.0339) \end{aligned}$$

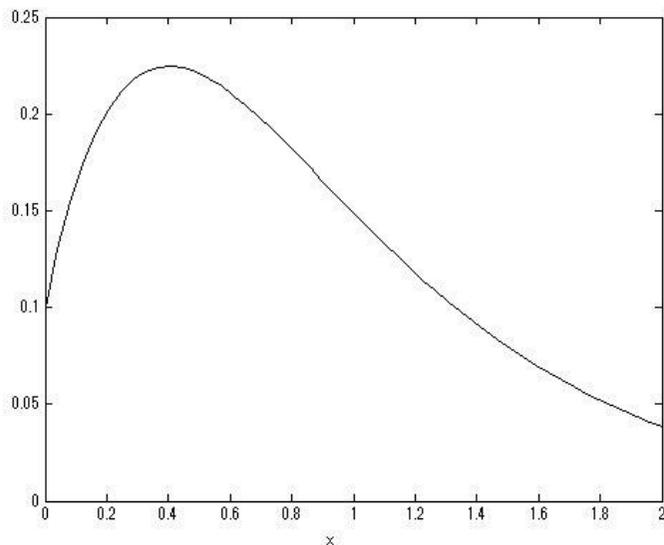
$$\begin{aligned} N(x, p) &= 0.0323y_1 + 1.6961y_2 + 0.0053y_3 + 0.0706y_4 - 0.2490y_5 \\ &\quad - 0.0932y_6 - 0.1001y_7 + 1.3673y_8 + 0.2381y_9 + 0.0534y_{10} \end{aligned}$$

$N(x, p)$  は  $x$  の減少関数となる。解析解と数値解から誤差を求めると、RMSE は  $3.8940e-05$  となり高精度の解が得られる。図 2 は  $\Psi_t(x, p)$  を示している。4 次のルンゲ・クッタ法による RMSE は 0.0070 となり、NN の方がより精度は高い。しかも陽表的な解が得られる。境界ペナルティ法を適用することもできる。この場合、(4) 式のかわりに

$$E(p) = \sum_{i=1}^n \left\{ \frac{d\Psi_t(x_i)}{dx} - f(x_i, \Psi_t(x_i)) \right\}^2 + \lambda \left( \sum_{i=1}^N w_i f(\theta_i) - 0.1 \right)^2$$

を最小化する。ここで  $\lambda > 0$  はペナルティ係数である。この方法を用いると誤差の RMSE は  $4.8427e-04$  となり、形状因子法とほとんど差はない。

図 2 1 階常微分方程式の数値解



## 2. 2階常微分方程式

消費のライフサイクルモデルによると、消費者は予算制約のもとで生涯の効用を最大化する。いくつかの仮定をおくと、資産はつぎの2階常微分方程式に従う。

$$\dot{A} = (2r - \rho)\dot{A} + r(\rho - r)A + (\rho - r)w$$

利率は  $r = 0.1$  で、割引率は  $\rho = 0.05$ 、賃金は  $w = 2$  とすると

$$\dot{A} = 0.15\dot{A} - 0.005A - 0.1 \quad (9)$$

となる。初期条件と境界条件は  $A(0) = 0, A(T) = 0$  とする。計画期間は  $T = 3$  とすると、(9) の解は

$$A(t) = 37.2142e^{0.05t} - 17.2142e^{0.1t} - 20$$

で与えられる。数値解法ではつぎの試行関数を用いる。

$$A(t, p) = t(3 - t)N(t, p) \quad (10)$$

簡単な計算により

$$\begin{aligned} \frac{dA(t, p)}{dt} &= (3 - 2t)N(t, p) + t(3 - t)\frac{dN(t, p)}{dt} \\ \frac{d^2A(t, p)}{dt^2} &= -2N(t, p) + 2(3 - 2t)\frac{dN(t, p)}{dt} + t(3 - t)\frac{d^2N(t, p)}{dt^2} \end{aligned}$$

となる。ここで

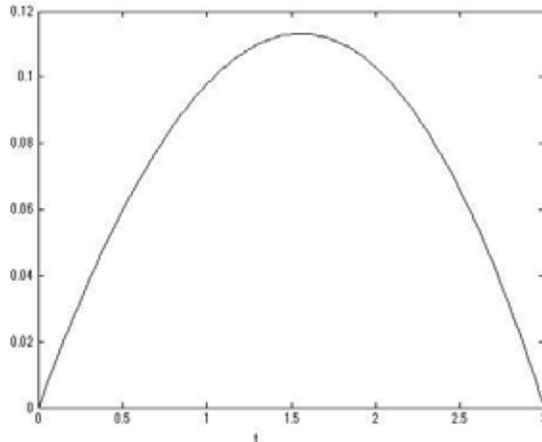
$$\begin{aligned} \frac{dN(t, p)}{dt} &= \sum_{i=1}^N v_i w_i y_i (1 - y_i) \\ \frac{d^2N(t, p)}{dt^2} &= \sum_{i=1}^N v_i^2 w_i y_i (1 - y_i)(1 - 2y_i) \end{aligned}$$

である。区間  $[0, 3]$  に 100 の分点をとる

$$E(p) = \sum_{i=1}^{100} \left\{ \frac{d^2A(t_i, p)}{dt^2} - (0.15 \frac{dA(t_i, p)}{dt} - 0.005A(t_i, p) - 0.1) \right\}^2 \quad (11)$$

を最小化して結合係数を求めた。図3は資産の最適経路を示している。資産は計画期間の前半は増加し、後半になると減少して最終的に0となる。平均二乗誤差は  $1.0751e-05$  とほとんど0となり正確な解が得られる。消費は、 $c = w + rA(t, p) - dA(t, p)/dt$  から計算する。従来の方法ではシューティング法を適用するが、計算は簡単ではない。対照的にニューラルネットワークを用いた方法は簡単な計算で済む。

図3 資産の変動



### 3. 連立微分方程式

つぎに連立 1 階微分方程式

$$\frac{d\Psi_i}{dx} = f_i(x, \Psi_1, \Psi_2, \dots, \Psi_K) \quad (12)$$

$$\Psi_i(0) = A_i, \quad (i=1, 2, \dots, K)$$

の解を求める。この場合、つぎの関数に 1 つのネットワークを充てる。

$$\Psi_{i_k}(x) = A_i + xN_{i_k}(x, p_i) \quad (13)$$

$K=2$  であれば 2 つのネットワークが必要となる。誤差関数を

$$E(p) = \sum_{k=1}^K \sum_{i=1}^n \left\{ \frac{d\Psi_{i_k}(x_i)}{dx} - f_k(x_i, \Psi_{i_1}, \Psi_{i_2}, \dots, \Psi_{i_K}) \right\}^2 \quad (14)$$

と定義する。

この方法でつぎの連立微分方程式を解いてみよう。

$$\frac{d\Psi_1}{dx} = \Psi_1 - 4\Psi_2 \quad (15)$$

$$\frac{d\Psi_2}{dx} = \Psi_1 + \Psi_2$$

$$0 \leq x \leq 2, \quad \Psi_1(0) = 2, \quad \Psi_2(0) = 3$$

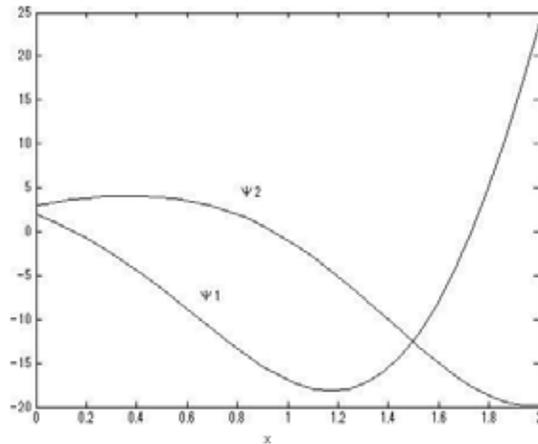
解析解は

$$\Psi_{a1}(x) = -2e^x + 4e^x \cos^2(x) - 12e^x \sin(x) \cos(x)$$

$$\Psi_{a2}(x) = -3e^x + 6e^x \cos^2(x) + 2e^x \sin(x) \cos(x)$$

となる。(13) 式によりつぎの関数を試みる。

図4 連立微分方程式の数値解



$$\Psi_{1_1}(x) = 2 + xN_1(x, p_1)$$

$$\Psi_{1_2}(x) = 3 + xN_2(x, p_2)$$

変数  $x$  について 30 の分点を取って計算すると、(15) の RMSE はそれぞれ 0.1739、70.1204 となる。これまで検討したのと比べると、誤差は大きくなる。その一つの理由として考えられるのは、NN のパラメータが 2 倍に増えたことである。図4 は  $\Psi_{1_1}(x)$  と  $\Psi_{1_2}(x)$  をプロットしている。

#### 4. 偏微分方程式

ニューラルネットは偏微分方程式にも適用可能である。一例として、つぎのポアソン方程式に適用しよう。

$$\frac{\partial^2 \Psi(x, y)}{\partial x^2} + \frac{\partial^2 \Psi(x, y)}{\partial y^2} = h(x, y) \quad (16)$$

計算領域は  $x, y \in [0, 1]$  で、4 つの境界条件:  $\Psi(0, y) = g_1(y)$ ,  $\Psi(1, y) = g_2(y)$ ,  $\Psi(x, 0) = g_3(x)$ ,  $\Psi(x, 1) = g_4(x)$  を課す。解候補としてつぎの関数を試みる。

$$\Psi_i(x, y) = A(x, y) + x(1-x)y(1-y)N(x, y, p) \quad (17)$$

NN の入力層は  $x$  と  $y$  に対応する 2 つのニューロンを含み、 $A(x, y)$  は

$$\begin{aligned} A(x, y) = & (1-x)g_1(y) + xg_2(y) + (1-y)\{g_3(x) - [(1-x)g_3(0) + xg_3(1)]\} \\ & + y\{g_4(x) - [(1-x)g_4(0) + xg_4(1)]\} \end{aligned} \quad (18)$$

とする。誤差関数は次式で定義する。

$$E(p) = \sum_{i=1}^n \left\{ \frac{\partial^2 \Psi(x_i, y_i)}{\partial x^2} + \frac{\partial^2 \Psi(x_i, y_i)}{\partial y^2} - h(x_i, y_i) \right\}^2 \quad (19)$$

つぎの偏微分方程式について考える。

$$\frac{\partial^2 \Psi(x, y)}{\partial x^2} + \frac{\partial^2 \Psi(x, y)}{\partial y^2} = 6x - 2x^2 + 12y - 2y^2 \quad (20)$$

(16) 式で  $h(x, y) = 6x - 2x^2 + 12y - 2y^2$  であり、境界条件は

$$\begin{cases} g_1(y) = 2y^3 \\ g_2(y) = 1 - y^2 + 2y^3 \\ g_3(x) = x^3 \\ g_4(x) = 2 - x^2 + x^3 \end{cases} \quad (21)$$

である。この場合、(18) 式の  $A(x, y)$  は

$$A(x, y) = xy - xy^2 - x^2y + x^3 + 2y^3$$

となる。ネットワークに  $x$  と  $y$  の値をインプットし、入出力関係は

$$z = \sum_{i=1}^N w_i f(u_i x + v_i y + \theta_i) \text{ と表される。これより}$$

$$\Psi_t(x, y) = xy - xy^2 - x^2y + x^3 + 2y^3 + x(1-x)y(1-y) \sum_{i=1}^N w_i f(u_i x + v_i y + \theta_i)$$

と表される。変数  $x$  と  $y$  で2回微分すると

$$\frac{\partial^2 \Psi_t(x, y)}{\partial x^2} = 6x - 2y + y(1-y)[-2z + 2(1-2x) \frac{dz}{dx} + x(1-x) \frac{d^2 z}{dx^2}]$$

$$\frac{\partial^2 \Psi_t(x, y)}{\partial y^2} = -2x - 12y + x(1-x)[-2z + 2(1-2y) \frac{dz}{dy} + y(1-y) \frac{d^2 z}{dy^2}]$$

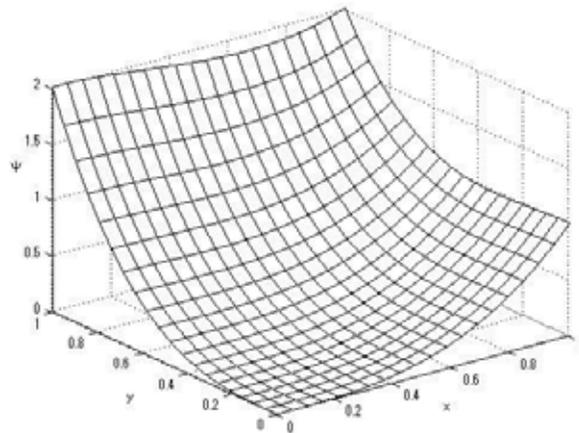
となる。20 の分点をとり、(19) を最小化して結合係数を求めた。形状因子法による解は図5のようになる。誤差の RMSE は  $1.799e-006$  となり、厳密解とほとんど区別できない。

境界ペナルティ法では誤差関数は次式で与えられる。

$$\begin{aligned} E(p) = & \sum_{i=1}^n \left\{ \frac{\partial^2 \Psi(x_i, y_i)}{\partial x^2} + \frac{\partial^2 \Psi(x_i, y_i)}{\partial y^2} - h(x_i, y_i) \right\}^2 + \lambda_1 \left( \sum_{i=1}^N w_i f(v_i y_i + \theta_i) - 2y^3 \right)^2 \\ & + \lambda_2 \left( \sum_{i=1}^N w_i f(u_i + v_i y_i + \theta_i) - 1 + y^2 - 2y^3 \right)^2 + \lambda_3 \left( \sum_{i=1}^N w_i f(u_i x_i + \theta_i) - x^3 \right)^2 \\ & + \lambda_4 \left( \sum_{i=1}^N w_i f(u_i x_i + v_i + \theta_i) - 2 + x^2 - x^3 \right)^2 \end{aligned}$$

$\lambda_i = 8(i=1, \dots, 4)$  とすると、RMSE は 0.0207 となり点 (1,1) 付近でやや大きな誤差が生じる。

図5 形状因子法による数値解



## 5. ブラック・ショールズ方程式

最後に取り上げるのはオプション価格に関するブラック・ショールズ方程式である。ヨーロピアン型オプションの原資産を株式とし、株価を  $S$  とすると  $t$  時点におけるコール・オプションの価格はつぎの偏微分方程式で表される。

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0 \quad (22)$$

ただし  $r$  は無リスク金利であり、 $\sigma^2$  は株式収益率の分散である。つぎの2つの条件を満たす必要がある。

$$V(S, T) = \max(S - K, 0)$$

$$V(0, t) = 0$$

$T$  は満期日で  $K$  は権利行使価格である。

いま  $\tau = T - t$  とおくと、(22) 式はつぎのように書き換えられる。

$$\frac{\partial V}{\partial \tau} - rS \frac{\partial V}{\partial S} - \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rV = 0 \quad (23)$$

$$V(S, 0) = \max(S - K, 0)$$

$$V(0, \tau) = 0$$

この方程式の解析解は

$$V(S, \tau) = S\Phi(d_1) - Ke^{-r\tau}\Phi(d_2) \quad (24)$$

$$d_1 = \frac{\log(S/K) + (r + \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}$$

$$d_2 = d_1 - \sigma\sqrt{\tau}$$

である。ただし、 $\Phi$  は標準正規分布の累積分布関数である。 $\tau \rightarrow 0$  のとき

$$V(S, \tau) \rightarrow \begin{cases} S - K, & S > K \\ 0, & S \leq K \end{cases}$$

となる。

つぎに NN を用いて数値的に解を求める<sup>3)</sup>。形状因子法では

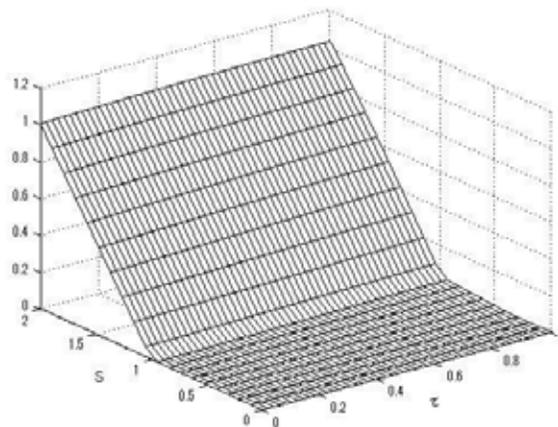
$$\Psi_t(S, \tau) = \max(S - K, 0) + \tau SN(S, \tau, p) \quad (25)$$

として、誤差関数

$$E(p) = \sum_{i=1}^n \left\{ \frac{\partial \Psi_t}{\partial \tau} - rS \frac{\partial \Psi_t}{\partial S} - \frac{1}{2} \sigma^2 \frac{\partial^2 \Psi_t}{\partial S^2} + r\Psi_t \right\}^2 \quad (26)$$

を最小化する。 $T = 1, K = 1, 0 \leq S \leq 2, 0 \leq \tau \leq 1, r = 0.05, \rho = 0.3$  として計算した。使用したのは、入力層は 2 個、出力層は 1 個のニューロンからなるネットワークである。図 6 はオプションの価格を示している。 $\tau = 0$  は満期日に対応し、価格は初期条件で与えられている。 $\tau = 1$  から  $\tau = 0$  へ移動するにつれて、満期のペイオフに近づくことが見てとれる。(24) の解析解と比べると、誤差は最大でも 0.0956 にとどまる。一方、境界ペナルティ法では最大誤差は 0.0760 となり若干精度は良くなる。ヨーロピアン型オプションについては差分法も使えるが、より複雑なタイプのオプションにはニューラルネットワークが適している。

図 6 コール・オプションの価格



## 6. 結語

ニューラルネットワークの関数近似能力に基づく微分方程式の解法をいくつかの例を用いて説明した。計算結果から見て、この方法は極めて有効で高精度の微分可能な解が得られる。境界条件を満たすように試行解を設定するので、制約付の最小化問題は無制約の問題に変換される。新しい解法は汎用性があり、常微分方程式だけでなく偏微分方程式にも適用可能である。また中間層が1層だけの簡単なネットワークでも良好な結果が得られた。ここでは最も簡単なネットワークを使用したが、出力層でも非線形の変換をしたり、中間層を増やすと複雑な微分方程式にも適用可能であろう。差分法に代表される従来の方法では、変数が増えると計算機のメモリーと計算時間が爆発的に増加する。一方、ニューラル微分方程式ではそうした問題は生じない。入力層のニューロンを増やすことで対処できる。また自動微分を使えば煩雑な微分の計算は必要でなくなる。今後の改善点は、サンプル点を一様に取り替わりに誤差の大きな領域では密にして近似精度を高めることである。また学習に際してサンプル点をランダムに選ぶ方法についても検討したい。

## 注

- 1) 基本的な文献は Lagaris, et al. (1998) と Chiamonte and Kiener (2013) である。ニューラルネットについては Bishop (2006) の第5章が詳しい。
- 2) 釜 (2001) の第11章～第13章を参照。甘利 (1978) はニューラルネットの数理モデルについて詳しい。
- 3) 釜 (2018) は差分法で数値解を求めている。

## 参考文献

- 甘利俊一 (1978) 『神経回路網の数理』産業図書。  
釜国男 (2001) 『経済行動の数量分析』多賀出版。  
———, (2018) 『コンピューテーショナル・エコノミクス』多賀出版。  
Bishop, C. M. (2006) *Pattern Recognition and Machine Learning*, Springer.  
『パターン認識と機械学習 上・下』元田浩他訳、丸善出版、2012。  
Chiamonte, M. and Kiener, M. (2013) “Solving differential equations using neural networks”, *Machine Learning Project*, Vol.1.  
Cybenko, G. (1989) “Approximations by superpositions of sigmoidal functions”, *Mathematics of Control, Signals, and Systems*, Vol.2, pp.303-314.  
Funahashi, K. (1989) “On the approximate realization of continuous mappings by neural networks”, *Neural Networks*, Vol.2, pp.183-192.  
Hornik, K., Stinchcombe, M. and White, H. (1989) “Multilayer feedforward networks are universal approximators”, *Neural Networks*, Vol.2, pp.359-366.  
Lagaris, E., Likas, A. and Fotiadis, D. (1998) “Artificial neural networks for solving Ordinary and partial differential equations”, *IEEE Transactions on Neural Networks*, Vol.9, pp.987-1000.