計算工学講演会論文集　Vol.28 (2023年5月)　　計算工学会

# 支配方程式による制約を加えた機械学習PINNに対する動的重み付け法

Balancing Multiple Back-Propagated Gradients with Dynamic Weight Tuning for Accurate Training and Inference of Physics-Informed Neural Network

出口翔大[1)]，浅井光輝[2)]
Shota Deguchi and Mitsuteru Asai

1) 九州大学大学院　工学府土木工学専攻（〒819-0023 福岡県福岡市西区元岡744, E-mail: deguchi@doc.kyushu-u.ac.jp）
2) 九州大学 工学研究院社会基盤部門（〒819-0023 福岡県福岡市西区元岡744, E-mail: asai@doc.kyushu-u.ac.jp）

Data-driven technology has been widely applied to various problems in science and engineering. In recent years, the community has focused on developing machine learning models that can integrate data and prior knowledge rather than purely data-driven approaches, and in particular, PINN (Physics-Informed Neural Network) has attracted much attention due to its applicability to both forward and inverse analysis. Since PINN trains based on both physical laws and data, the weights need to be chosen appropriately. In this study, we present a dynamic weight tuning method for PINN training. Numerical experiments show that the presented method enables accurate inverse analysis while keeping the additional computational cost within a reasonable range.

***Key Words:*** *Machine Learning, Physics-Informed Neural Network, Dynamic Weight Tuning*

## 1.　緒言

データ駆動型科学の技術は，幅広い分野で活用・応用が展開されており，画像認識などの分野では機械学習技術が成熟しつつある．同技術は工学分野にも拡張されつつあり，物理現象を評価するための新たな方法論としての研究が盛んである．近年では，既知の物理的支配原理とデータとの融合に基づく機械学習モデルの開発が展開されており[1,2]，中でもPINN（Physics-Informed Neural Network）[3]は，要素離散化を必要としない点や，順解析・逆解析の両者に柔軟に適用可能であるという特性から，幅広い応用がなされている[1,4,5,6].

PINNは支配方程式とデータの情報を組みわせた学習を行うため，それぞれにどの程度の重みを与えるべきかは慎重に検討する必要がある．この重みは，多くの問題でハイパーパラメータとして取り扱われているものの[4,6]，PINNの学習は重みの定め方により大きく影響されるため，適切にチューニングすることが重要である[7,8,9].

そこで著者らは，PINNの学習過程にネットワーク中を逆伝播する損失関数のパラメータ勾配を用いた動的重み付け法（DN: Dynamic Normalization）を提案している[10]．DNは，新たなハイパーパラメータを必要とするものの，上記の重みを試行錯誤的に定める必要性を取り除きつつPINNによる近似解の高精度化を可能にする手法であり，導入による計算コストの増分も数10%以内に抑えることができる．本研究では，先行研究[10]で順解析への応用に留められていたDNを，PINNを用いた逆解析へと適用し，その有用性を確認した．

## 2.　機械学習手法

### (1)　PINN: Physics-Informed Neural Network

以下のように記述される初期値境界値問題を考える．

$$\frac{\partial}{\partial t}u(t,\boldsymbol{x}) = \mathcal{F}[u(t,\boldsymbol{x});\boldsymbol{\mu}] \tag{1}$$

$$u(0,\boldsymbol{x}) = g(\boldsymbol{x}) \tag{2}$$

$$u(t,\boldsymbol{x}) = h(t,\boldsymbol{x}) \tag{3}$$

ここで，式(1)は支配原理である偏微分方程式，式(2)，(3)はそれぞれ初期条件，境界条件である．また，式(1)の$\mathcal{F}[\cdot;\boldsymbol{\mu}]$は$\boldsymbol{\mu}$をパラメータに持つ偏微分作用素である．PINNは，まず解$\boldsymbol{u}$をニューラルネットワーク$\hat{\boldsymbol{u}}$により近似する．入力$\mathbf{x} = \{t,\boldsymbol{x}\}\left(\in \mathbb{R}^{f_{\text{in}}}\right)$，出力$\hat{\mathbf{y}} = \{\hat{u}\}\left(\in \mathbb{R}^{f_{\text{out}}}\right)$を持つ$L$層のニューラルネットワークにおいて，第$l(=1,2,\ldots,L)$層における順伝播$\mathbf{z}^{(l)}\left(\in \mathbb{R}^{f_{\text{hidden}}^{(l)}}\right)$は，以下の通りである．

$$\mathbf{z}^{(l)} = \sigma^{(l)}\left(\mathbf{W}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}\right) \tag{4}$$

ここで，$\mathbf{z}^{(0)} = \mathbf{x}$，$\mathbf{z}^{(L)} = \hat{\mathbf{y}}$であり，$\sigma^{(l)}(\cdot)$は活性化関数と呼ばれる，要素ごとに作用する非線形写像（出力層でのみ恒等写像）である．また，$\mathbf{W}^{(l)}$, $\mathbf{b}^{(l)}$は第$l$層における重み，バイアスであり，これらは，以下の損失関数を最小化することで学習する．

$$\begin{aligned}\mathcal{L} &= \lambda_{\text{PDE}}\mathcal{L}_{\text{PDE}} + \lambda_{\text{IC}}\mathcal{L}_{\text{IC}} + \lambda_{\text{BC}}\mathcal{L}_{\text{BC}} + \lambda_{\text{Data}}\mathcal{L}_{\text{Data}} \\ &= \sum_j \lambda_j \mathcal{L}_j\end{aligned} \tag{5}$$

$$\mathcal{L}_{\mathrm{PDE}} = \sum_i \left| \frac{\partial}{\partial t}\hat{u}_i - \mathcal{F}[\hat{u}_i; \boldsymbol{\mu}] \right|^2 \tag{6}$$

$$\mathcal{L}_{\mathrm{IC}} = \sum_i |\hat{u}_i - g(\boldsymbol{x}_i)|^2 \tag{7}$$

$$\mathcal{L}_{\mathrm{BC}} = \sum_i |\hat{u}_i - h(t_i, \boldsymbol{x}_i)|^2 \tag{8}$$

$$\mathcal{L}_{\mathrm{Data}} = \sum_i |\hat{u}_i - u(t_i, \boldsymbol{x}_i)|^2 \tag{9}$$

ここで，$\hat{u}_i = \hat{u}(t_i, \boldsymbol{x}_i; \boldsymbol{\theta})$，$\boldsymbol{\theta}$はパラメータベクトルである（$\boldsymbol{\theta} = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L}$）．また，$\lambda_j$は各損失項に与える重みであり，それぞれの相対的な重要度を示す．

学習には，勾配降下法を用いることとする．勾配降下法による学習は，以下の通りである．

$$\begin{aligned}\boldsymbol{\theta}^{(n+1)} &= \boldsymbol{\theta}^{(n)} - \eta^{(n)} \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}^{(n)}\right) \\ &= \boldsymbol{\theta}^{(n)} - \eta^{(n)} \sum_j \lambda_j \nabla_{\boldsymbol{\theta}} \mathcal{L}_j\left(\boldsymbol{\theta}^{(n)}\right)\end{aligned} \tag{10}$$

ここで，$\boldsymbol{\theta}^{(n)}$, $\eta^{(n)}$はそれぞれ$n$エポック目でのパラメータ，学習率である．

(2) **動的重み付け法**（DN: Dynamic Normalization）

損失関数$\mathcal{L}(\boldsymbol{\theta})$の，$\boldsymbol{\theta}^{(n)}$まわりでの2次までのTaylor展開を考える．

$$\begin{aligned}\mathcal{L}^{(n+1)} = \mathcal{L}^{(n)} &+ \Delta\boldsymbol{\theta}^{(n)\mathsf{T}} \mathbf{G}^{(n)} \\ &+ \frac{1}{2} \Delta\boldsymbol{\theta}^{(n)\mathsf{T}} \mathbf{H}^{(n)} \Delta\boldsymbol{\theta}^{(n)}\end{aligned} \tag{11}$$

表記の簡略化のため，$\mathcal{L}^{(n)} = \mathcal{L}\left(\boldsymbol{\theta}^{(n+1)}\right)$, $\mathbf{G}^{(n)} = \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{\theta}^{(n)}\right)$, $\mathbf{H}^{(n)} = \nabla_{\boldsymbol{\theta}}^2 \mathcal{L}\left(\boldsymbol{\theta}^{(n)} + \alpha\Delta\boldsymbol{\theta}^{(n)}\right)$とした．なお，$\alpha \in (0,1)$である．式(10)，および式(11)から，以下を得る．

$$\begin{aligned}\mathcal{L}^{(n+1)} - \mathcal{L}^{(n)} = &-\eta^{(n)} \left\| \mathbf{G}^{(n)} \right\|_2^2 \\ &+ \frac{1}{2} \left(\eta^{(n)}\right)^2 \mathbf{G}^{(n)\mathsf{T}} \mathbf{H}^{(n)} \mathbf{G}^{(n)}\end{aligned} \tag{12}$$

ここで，右辺第2項はPINNの学習過程を洞察する上で有用であるものの[8]，$\mathbf{H}^{(n)}$の計算が高負荷であるため，実用上，これを利用して重みを定めることは非現実的である．したがって，動的重み付け法（DN: Dynamic Normalization）[10]では第2項を無視し，第1項の勾配に関する項に基づき重みを定める．まず，式(12)を以下のように変形する．

$$\sum_j \Delta\mathcal{L}_j^{(n)} = -\eta^{(n)} \left\| \sum_j \lambda_j^{(n)} \mathbf{G}_j^{(n)} \right\|_2^2 \tag{13}$$

ここで，$j$番目の損失項$\mathcal{L}_j^{(n)}$が，対応する勾配$\mathbf{G}_j^{(n)}$によってのみ減少されると仮定すれば，$\Delta\mathcal{L}_j^{(n)} = -\eta^{(n)} \left\| \lambda_j^{(n)} \mathbf{G}_j^{(n)} \right\|_2^2$である．なお，PINNの学習過程においては，各損失項に応じて勾配の分布に散付きが現れやすいことが知られており[8,11]．特に他の損失項と比較して，支配方程式からの残差を評価する項$\mathcal{L}_{\mathrm{PDE}}$を急速に減少させる傾向が存在する（すなわち，$\|\mathbf{G}_{\mathrm{PDE}}\|_2 \gg \|\mathbf{G}_{\mathrm{BC}}\|_2$など）[9]．物理的な問題を評価する上では支配方程式だけでなく，初期条件・境界条件も考慮する必要があることから，複数の損失項を同等の速度で減少させることが重要である．以上より，複数の損失項を同等の速度で減少するには，以下の関係が満足されるべきである．

$$\left\| \lambda_j^{(n)} \mathbf{G}_j^{(n)} \right\|_2^2 = -\eta^{(n)} \left\| \lambda_k^{(n)} \mathbf{G}_k^{(n)} \right\|_2^2 \tag{14}$$

ここで，相対的な重み（重要度）を考えるため，$\lambda_k^{(n)} = 1.0$とすると，以下の重みを得る．

$$\lambda_j^{(n)} = \frac{\left\| \mathbf{G}_k^{(n)} \right\|_2}{\left\| \mathbf{G}_j^{(n)} \right\|_2} \tag{15}$$

ここで，$\lambda_j^{(n)}$は学習過程で計算される勾配に基づいて計算されるため，学習中に動的に変化する．また，計算負荷を抑えるため，$\tau(\geq 1)$エポックごとに更新することとする（例えば，$\tau = 5, 10$など[8,10,11]）．ただし，式(15)の定義を直接適用すると，$\lambda_j^{(n)}$が大きく振動することから，指数減衰率$\beta\left(\in [0,1)\right)$を用いて以下の平滑化を施す．

$$\hat{\lambda}_j^{(n)} = \frac{\left\| \mathbf{G}_k^{(n)} \right\|_2}{\left\| \mathbf{G}_j^{(n)} \right\|_2} \tag{16}$$

$$\lambda_j^{(n)} = \beta\lambda_j^{(n-1)} + (1-\beta)\hat{\lambda}_j^{(n)} \tag{17}$$

ここで，$\beta = 0$とすると，式(15)に一致する．$\beta$はハイパーパラメータであり，経験的に$\beta \to 1$で$\lambda_j^{(n)}$が安定することが知られている（$\beta = 0.9, 0.99$など[10]）．

## 3. 数値実験

本章では，上述の動的重み付け法（DN: Dynamic Normalization）を適用し，PINNによる逆解析を行った数値実験を示す．本研究で用いたネットワーク構造は，$\left\{f_{\mathrm{hidden}}^{(l)}\right\}_{l=1}^{L-1} = f_{\mathrm{hidden}} = 50$，$L = 5$とし[9,12]，活性化関数には$\left\{\sigma^{(l)}(\cdot)\right\}_{l=1}^{L-1} = \sigma(\cdot) = \tanh(\cdot)$を選択した[13]．活性化関数の選択に伴い，重みの初期値にはGlorotの初期値[14]を採用した．最適化手法にはAdam[15]を採用し，学習率$\eta$は$\eta = 0.001$，1次・2次モーメント推定の指数減衰率$\beta_1$, $\beta_2$はそれぞれ$\beta_1 = 0.9$，$\beta_2 = 0.999$とした．なお，本論文におけるPINNの開発にはTensorFlow[16]を，学習にはNVIDIA RTX A6000を用いた．

(1) **ハイパーパラメータの検討**

まず，動的重み付け法（DN: Dynamic Normalization）で導入されるハイパーパラメータに関して，先行研究[10]，および類似の研究[8,11]から，一定の指針は示されているものの，未だ議論の余地が残される．本節では，動的重みの指数減衰率$\beta$，および更新間隔$\tau$に関する検討を行う．

検証問題として，先行研究[9]に倣い熱拡散方程式を取り上げる．支配方程式，および初期条件，境界条件は以下のように与えられる．

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \tag{18}$$

$$u(0,x) = \sin(\pi x) \tag{19}$$

$$u(t,0) = u(t,1) = 0 \tag{20}$$

ここで，$t \in [0, 0.5]$，$x \in [0, 1]$，$\nu$は拡散係数であり，$\nu = 1.0$とした．上記には以下の厳密解が存在する（**図-1**参照）．

$$u = \sin(\pi x)\exp(-\nu\pi^2 t) \tag{21}$$

**図-1**には，マゼンタ色の四角形で示す点が観測点であり，式(9)の$u(t_i, \boldsymbol{x}_i)$に相当する．ここでは，$x = \{0.25, 0.5, 0.75\}$の3点において$t = \{0.1, 0.2, 0.3, 0.4, 0.5\}$に観測される温度の情報から拡散係数$\nu$を逆解析する．10,000エポックだけ学習を行うこととし，テストエラーの推移，学習コストの増加に基づき$\beta$，$\tau$の設定を検討する．それぞれに対して$\beta = \{0, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99, 0.999\}$，$\tau = \{1, 10\}$の範囲でグリッドサーチを行った結果，いずれの$\tau$に対しても$\beta < 0.9$ではテストエラーがほとんど減少しなかったため，$\beta = \{0.9, 0.99, 0.999\}$の結果のみ示す．

10,000エポック間のテストエラーの推移を**図-2**に示す．また，$\tau$と学習コストとの関係を**表-1**に示す．**表-1**には，比較のため，動的重みを適用しない場合（w/o DN）のコストを記し，括弧内には増分を示している．**図-2**より，$\tau = 1$の場合，$\beta = 0.9, 0.99$ではテストエラーが減少せず，$\beta = 0.999$でのみその減少が確認できる．$\tau = 10$では，$\beta = 0.9$ではテストエラーが減少しないものの，$\beta = 0.99, 0.999$で適切な学習が実行できている．以上より，$\beta$を1,0に近い値に定めておき，式(16)，(17)で定める動的重みを緩やかに変化させることで安定した学習が可能になると考える．なお，**表-1**より，$\tau$の選択によって学習コストに隔たりがある．上記の$\beta$に関する議論より，安定した学習のためには，動的重みは緩やかに変化させることが好ましい．また，主に式(16)の計算に起因する学習コストの増分は小さく抑えるべきである．以上から，本研究では$\beta = 0.99, 0.999$，$\tau = 10$を選択する．

(2) 1次元熱拡散方程式

前節と同じ問題において，動的重みを適用しない場合（w/o DN）と適用する場合（w/ DN）での結果を比較する．なお，ニューラルネットワークの重みの初期値[14]は確率分布から抽出されるため，学習・推論には一定のランダム性が含まれる[17]．したがって，ここでは異なる乱数シードを用いて10回の独立な試行を行った結果を示す．

30,000エポックのフルバッチ学習を行い，拡散係数$\nu$の逆解析を行った．学習過程での推定値の推移を**図-3**に示す．**図-3**では，10試行の平均値を実線で，標準偏差の2倍の区間を網掛け線で示している．また，学習終了時点での

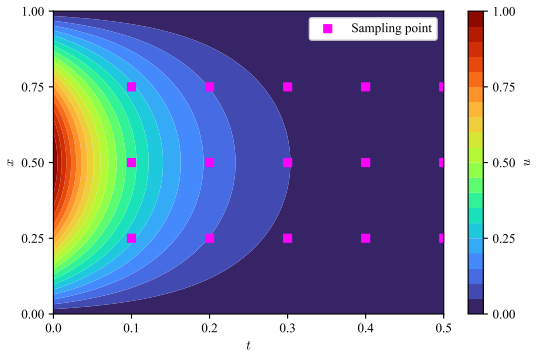


**図-1 熱拡散方程式の参照解**

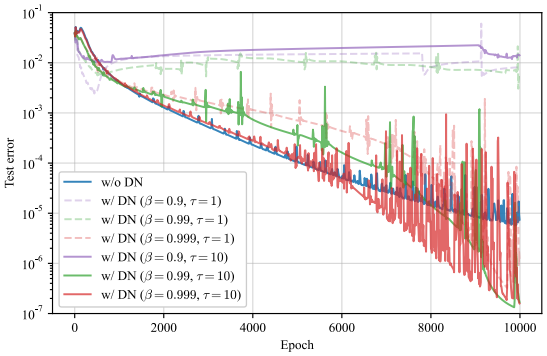


**図-2 熱拡散方程式：テストエラーの推移**

**表-1 熱拡散方程式：更新間隔$\tau$と学習コスト**

| Method | $\tau$ | Training time (sec / 10,000 epochs) |
|---|---|---|
| w/o DN | - | 182.69 (× 1.00) |
| w/ DN | 1 | 231.80 (× 1.27) |
| w/ DN | 10 | 189.96 (× 1.04) |

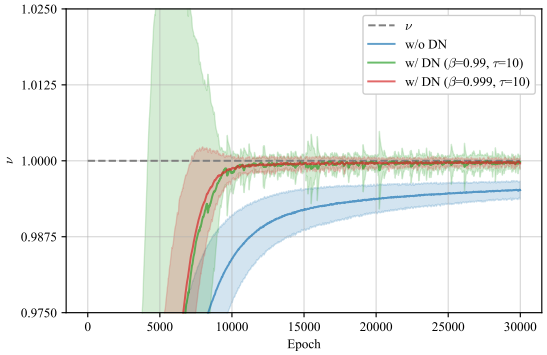


**図-3 熱拡散方程式：拡散係数の推定値の推移**

**表-2 熱拡散方程式：拡散係数の推定結果**

| Method | $\beta$ | $\tau$ | Identified $\nu$ ($\pm$ std) |
|---|---|---|---|
| w/o DN | - | - | 0.9952 ($\pm 7.5 \times 10^{-4}$) |
| w/ DN | 0.99 | 10 | 0.9998 ($\pm 4.9 \times 10^{-4}$) |
| w/ DN | 0.999 | 10 | 0.9996 ($\pm 4.8 \times 10^{-4}$) |

拡散係数$\nu$の推定値を**表-2**に示す．**表-2**でも**図-3**と同様に10試行の平均値と標準偏差を示している．**図-3**より，動的重みを適用しない場合のPINNは，30,000エポックの学習では未だ真値へ収束しきれておらず，高精度な逆解析のためには，より長時間の学習が必要であると予想される．

一方，動的重みを適用した場合には10,000エポックの学習を終えた時点で比較的良好な推定値を得ており，その後，徐々に真値へと漸近している．上記より，本問題においては，式(16)，(17)で定める動的重みはPINNを用いた逆解析において高速化・高精度化に寄与していると考える．

(3) 1次元移流拡散方程式

続いて先行研究[18]に倣い，移流拡散方程式を取り上げる．支配方程式，初期条件，境界条件は以下の通りである．

$$\frac{\partial u}{\partial t}+\mu\frac{\partial u}{\partial x}=\nu\frac{\partial^2 u}{\partial x^2} \tag{22}$$

$$u(0,x)=-\sin(\pi x) \tag{23}$$

$$u(t,-1)=u(t,1)=0 \tag{24}$$

ここで，$t\in[0,1]$，$x\in[-1,1]$である．$\mu$は移流速度，$\nu$は拡散係数であり，$\mu=1.0$，$\nu=0.1/\pi\cong 0.0318$とした[18]．上記の問題にはFourier無限級数による厳密解が知られているが[19]，ここでは800項までの有限級数を参照解とする[18]（**図-4**参照）．ただし，級数展開が煩雑であるため，参照解の式の記述は省略する（[19]参照）．本問題では，$x=\{-0.5,0.0,0.5\}$の3点において$t=\{0.2,0.4,0.6,0.8,1.0\}$に観測される濃度の情報（**図-4**内のマゼンタ色の四角形）から移流速度$\mu$と拡散係数$\nu$を逆解析する．また，初期化による結果の散付きを考慮するため，異なる乱数シードを用いて10回の独立な試行を行った結果を報告する．

60,000エポックのフルバッチ学習を行った結果を示す．学習過程での移流速度$\mu$の推定値，拡散係数$\nu$の推定値の推移を**図-5**，**図-6**にそれぞれ示す．前節と同様に10試行の平均値を実線で，標準偏差の2倍の区間を網掛け線で示す．また，学習終了時点でのそれぞれの推定値（平均値・標準偏差）を**表-2**，**表-3**に示す．**図-5**，および**表-2**より，60,000エポックの学習では，動的重みを適用しない場合と適用する場合の両者で，高精度に移流速度の推定が実行できている．一方，**図-5**，**表-3**より，動的重みを適用しない場合と$(\beta,\tau)=(0.99,10)$として適用した場合とでは，両者に明確な差がほとんど見られないが，$(\beta,\tau)=(0.999,10)$として動的重みを適用した場合には，拡散係数の逆解析が僅かに高精度化している．以上より，本問題においては，動的重みを適用することで僅かな高精度化が期待できるものの，結果には顕著な影響を与えるものではなかった．著者らが提案するDN以外の動的重み付け手法に関する研究でも，重みが結果に大きく影響する問題と影響し辛い問題があることが報告されており[8,11]，今後どのような問題に対して重みを考慮すべきか整理する必要がある．

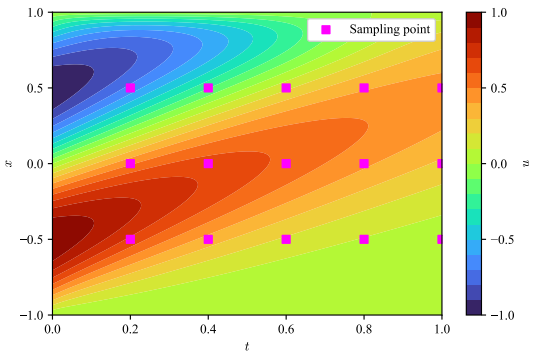


**図-4 移流拡散方程式の参照解**

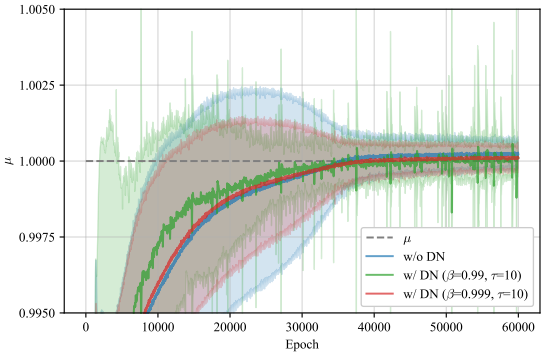


**図-5 移流拡散方程式：移流速度の推定値の推移**

**表-2 移流拡散方程式：移流速度の推定結果**

| Method | $\beta$ | $\tau$ | Identified $\mu$ ($\pm$ std) |
|---|---|---|---|
| w/o DN | - | - | 1.0002 ($\pm 2.1\times10^{-4}$) |
| w/ DN | 0.99 | 10 | 1.0001 ($\pm 2.5\times10^{-4}$) |
| w/ DN | 0.999 | 10 | 1.0001 ($\pm 2.2\times10^{-4}$) |

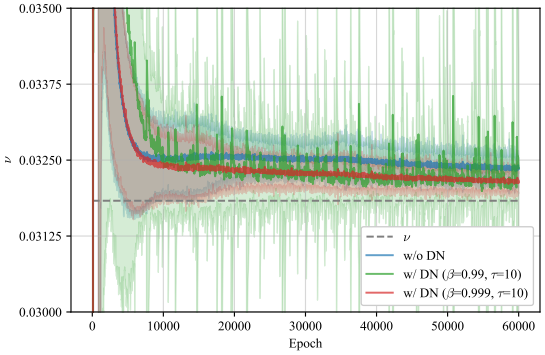


**図-6 移流拡散方程式：拡散係数の推定値の推移**

**表-3 移流拡散方程式：拡散係数の推定結果**

| Method | $\beta$ | $\tau$ | Identified $\nu$ ($\pm$ std) |
|---|---|---|---|
| w/o DN | - | - | 0.0324 ($\pm 9.3\times10^{-5}$) |
| w/ DN | 0.99 | 10 | 0.0324 ($\pm 9.1\times10^{-4}$) |
| w/ DN | 0.999 | 10 | 0.0321 ($\pm 9.2\times10^{-5}$) |

## 4. 結言

本研究では，PINNの学習においてハイパーパラメータとして設定されてきた各損失項に与える重みの動的な定め方を議論した．著者らの先行研究[10]では，式(16)，(17)で定める動的重みの適用は順解析に留まっていた．本研究では，同様に定めた重みをPINNを用いた逆解析に適用した．特定の損失項が対応する勾配のみによって減少するという仮定が導入されているものの，数値実験から，その一定の有用性を確認した．熱拡散方程式に関する数値実験では，逆解析の高速化・高精度化において一定の成果があることを確認したものの，移流拡散方程式においては，重みを適用しない場合と比較してほぼ同等の結果であった．これまでに，重みを調整しなくとも一定の精度で順解析・逆解析が可能である例も複数示されていることから[3,9,20]，今回数値実験例として取り上げた移流拡散方程式（また，考えた初期条件，境界条件）も重みの効果が顕著に表れなかったものと考える．今後は，[8]に倣い，式(11)，(12)のヘッシアン$\mathbf{H}^{(n)}$の情報を活用することで，問題に応じ，重み調整の必要性を検討する予定である．また，著者らが提案する動的重みの性質として，指数減衰を施しているため，学習の初期段階では重みが初期値向きにバイアスされている．特に，$\beta$が 1.0 に近い値をとるとき，長時間の学習をしなければ重みが変動し辛いため，動的重み付け法にも改善の余地がある．例えば，Adam[14]に倣い，バイアス補正を施すことなどで，学習初期段階から適切に重みが変動することが期待できる．上記の課題に関しても，今後検討する予定である．

**謝辞**: 本研究は，JSPS科研費JP20H02418，JST次世代研究者挑戦的研究プログラムJPMJSP2136，および九州大学数理・データサイエンス教育研究センターの支援を受けた．ここに記して謝意を表する．

### 参考文献

[1] Karniadakis, G.E., Kevrekidis, I.G., Lu, L. *et al.*: Physics-informed machine learning. *Nature Reviews Physics*, Vol. 3, pp. 422-440, 2021.

[2] von Rueden, L., Mayer, S., Beckh, K. et al.: Informed Machine Learning – A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 1, pp. 614-633, 2023.

[3] Raissi, M., Perdikaris, P. and Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, Vol. 378, pp. 686-707, 2019.

[4] Sahli Costabal, F., Yang, Y., Perdikaris, P. et al.: Physics-informed neural networks for cardiac activation mapping, *Frontiers in Physics*, Vol. 8, pp. 42, 2020.

[5] Cai, S., Wang, Z., Fuest, F. et al.: Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks, *Journal of Fluid Mechanics*, Vol. 915, pp. A102, 2021.

[6] Buhendwa, A.A., Adami, S. and Adams, N.A.: Inferring incompressible two-phase flow fields from the interface motion using physics-informed neural networks, *Machine Learning with Applications*, Vol. 4, pp. 100029, 2021.

[7] 出口翔大，柴田洋佑，浅井光輝: 予測に物理的意味を付与した機械学習 PINNs による誤差を含む教師データからのパラメータ推定，土木学会論文集A2（応用力学）, Vol. 77, No. 2, pp. I_35-I_45, 2021.

[8] Wang, S, Teng, Y. and Perdikaris, P.: Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks, *SIAM Journal on Scientific Computing*, Vol. 43, No. 5, pp. A3055-A3081, 2021.

[9] Rohrhofer, F.M., Posch, S. and Geiger, B.C.: On the Pareto Front of Physics-Informed Neural Networks, *arXiv*: 2105.00862, 2021.

[10] Deguchi, S. and Asai, M.: Dynamic & norm-based weights to normalize imbalance in back-propagated gradients of physics-informed neural networks, *Journal of Physics Communications* (submitted).

[11] Maddu, S., Sturm, D., Müller, C.L. and Sbalzarini, I.F.: Inverse Dirichlet weighting enables reliable training of physics informed neural networks, *Machine Learning: Science and Technology*, Vol. 3, No. 1, pp. 015026, 2022.

[12] Jin, X., Cai, S., Li, H. and Karniadakis, G.E.: NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations, *Journal of Computational Physics*, Vol. 426, pp. 109951, 2021.

[13] 出口翔大，柴田洋佑，浅井光輝: 空間特徴量抽出を援用した PINNs によるパラメータ逆解析の効率化，土木学会論文集, Vol. 79, No. 15, pp. 22-15011, 2023.

[14] Glorot, X. and Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9, pp. 249-256, 2010.

[15] Kingma, D.P. and Ba, J.: Adam: A Method for Stochastic Optimization, *3rd International Conference on Learning Representations (ICLR)*, 2015.

[16] Abadi, M., Agarwal, A. Barham, P. et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015.

[17] Tartakovsky, A.M., Marrero, C.O., Perdikaris, P. et al.: Physics-Informed Deep Neural Networks for Learning Parameters and Constitutive Relationships in Subsurface Flow Problems, *Water Resources Research*, Vol. 56, No.

5, pp. e2019WR026731, 2020.

[18] Kharazmi, E., Zhang, Z. and Karniadakis, G.E.M.: hp-VPINNs: Variational physics-informed neural networks with domain decomposition, *Computer Methods in Applied Mechanics and Engineering*, Vol. 374, pp. 113547, 2021.

[19] Mojtabi, A. and Deville, M.O.: One-dimensional linear advection–diffusion equation: Analytical and finite element solutions, *Computers & Fluids*, Vol. 107, pp. 189-195, 2015.

[20] Lu, L., Meng, X., Mao, Z. et al.: DeepXDE: A deep learning library for solving differential equations, *SIAM Review*, Vol. 63, No. 1, pp. 208-228, 2021.