

# 変分原理とニューラルネットワーク

佐藤 哲†

パーソルキャリア株式会社†

## 1. はじめに

変分原理は、変分法により汎関数の極値を求めることで方程式を導く考え方であり、関数の最適化手法と関係が深い。一方で、ニューラルネットワークは、パラメータを調整することで必要な出力を得る手法であり、やはり関数の最適化が重要である。従って変分原理とニューラルネットワークには様々な共通したテクニックがあり、変分原理の問題を解くためにニューラルネットワークを利用することは自然な考えである。そこで本研究では、変分原理の問題に対しニューラルネットワークを適用し、変分原理の目的である方程式の解軌道を求めるための一連のアルゴリズムを提案する。

## 2. 変分原理と変分原理により導かれる方程式

変分原理は、変分法により汎関数の極値を求めることで方程式を導く考え方であり、数学や物理学、工学などの分野で広く使用されている。変分法は境界条件や近似の精度により様々な定式化が考えられるが、もっとも基本的な考え方は次のようなものである。

パラメータ  $t$ ,  $t$  による関数  $x(t)$ ,  $x(t)$  の  $t$  による微分  $x'(t)$  を引数とする関数  $F(t, x, x')$  に対し定義される作用積分と呼ばれる積分を考える：

$$S = \int_{t_0}^{t_1} F(t, x, x') dt. \quad (1)$$

式 (1) に対する変分法とは、関数  $x(t)$  に対する変分  $\delta x(t)$  を考えたときの作用積分の変化

$$\delta S = \delta \int_{t_0}^{t_1} F(t, x, x') dt \quad (2)$$

$$= \int_{t_0}^{t_1} F(t, x + \delta x, x' + \delta x') dt - \int_{t_0}^{t_1} F(t, x, x') dt \quad (3)$$

がゼロになるような条件

$$\delta S = 0 \quad (4)$$

を満たす関数  $x(t)$  を求める手法を指す。式 (4) は作用積分が停留値を持つことの必要条件であり、これにより多くの問題を解くことができる。例えば  $F$  としてラグランジアン

$$L = T - U \quad (5)$$

を取ればラグランジュの運動方程式

$$\frac{\partial L}{\partial x} - \frac{d}{dt} \frac{\partial L}{\partial x'} = 0 \quad (6)$$

を導くことができ、 $F$  としてラグランジアンをルジャンドル変換した

$$H = pq' - L \quad (7)$$

を取ればハミルトンの正準方程式

$$\frac{dq}{dt} = \frac{\partial H}{\partial p} \quad (8)$$

$$\frac{dp}{dt} = -\frac{\partial H}{\partial q} \quad (9)$$

が、 $F$  としてリーマン計量  $ds = \sqrt{g_{ij} dx^i dx^j}$  を取れば測地線の微分方程式を導くことができる。ここで、 $T$  は運動エネルギー、 $U$  はポテンシャルエネルギー、 $p$  は運動量ベクトル、 $q$  は座標ベクトル、 $g_{ij}$  は計量テンソルと呼ばれる座標ベクトルの関数、 $dx^i$  はリーマン幾何学で言う反変ベクトルであるが、大雑把に言って座標ベクトル  $x_i$  の微小な変化  $\Delta x_i$  と考えて良い。ただし、方程式を導くことができたからといって、解を求めることができるとは限らない。解析的な解を求めることが難しい場合や、解析的な形が分かっても計算が困難な場合もある。さらに、現実世界の問題としては  $F$  の形が不明であることもあり得る。そこで、データから学習することで具体的な  $F$  の形を用いずに  $F$  の値を得られるようにし、またニューラルネットワークを微分することで方程式を導き、数値的に解を求める一連のアルゴリズムを提案する。

Variational Principle and Neural Network  
†Tetsu R. Satoh, PERSOL CAREER CO., LTD.

### 3. 変分原理に対するニューラルネットワークの利用

変分問題をに対するニューラルネットワークを用いたアプローチとしては、 $L$ として式(7)を採用した手法[1][2]、式(5)を採用した手法[3][4]、式(7)や式(5)に対しあえて一般化座標系を用いることをやめることで精度向上を図る手法[5]、作用積分の変分を構成して具体的に作用積分の値を計算し、式(1)の値を最小にする関数を求める方法[6]などがあげられる。前者の手法は変分原理に用いられる作用積分がスカラー関数であり、スカラー関数を学習によって構成することにより、自動微分を用いることで時間発展の運動方程式を導出可能であることを利用している。また後者の手法は、変分原理に用いられる作用積分がスカラー関数であり、最適化手法を容易に適用可能であることを利用している。本研究では、ハミルトンの正準方程式やラグランジュの運動方程式を含むオイラー・ラグランジュ方程式を考える。オイラー・ラグランジュ方程式は、式(3)を部分積分によって変形することにより得られ、次式で表される：

$$\frac{\partial F}{\partial x} - \frac{d}{dt} \frac{\partial F}{\partial x'} = 0. \quad (10)$$

理論的には、 $F$ の形が分かれば式(10)より偏微分方程式を構成することが可能である。しかし実際には、(1) $F$ の形によっては非常に複雑な偏微分方程式が導出され、解析が難しい、(2)方程式が導出されたとしても、解を求めることが難しい、(3) $F$ の形が分からない場合がある、などの数理科学上の一般的な問題がある。それらの問題に対するアプローチの一つが、 $F$ をニューラルネットワークを用いて近似する手法である。ニューラルネットワークを微分可能なように構成しておけば、(1)偏微分方程式を導出することができ、(2)解を求めるために数値解法を適用することができ、(3)形が不明でもデータからの学習により $F$ を構築できる、という可能性がある。

データから $L$ を得るためには、損失関数として以下のものを使うことが考えられる：

$$loss_t = |F - F^*|_p \quad (11)$$

ここで、 $F^*$ は学習データ、記号 $||_p$ は $p$ -ノルムを表す。しかし、式(11)によりネットワークを訓練した場合、何の条件も無いために単なる $t, x, x' \rightarrow F(t, x, x')$

という写像が構成されるだけで、 $F$ の微分についての情報は得られない。そこで式(10)を考慮した項

$$loss_c = \left| \frac{\partial F}{\partial x} - \frac{d}{dt} \frac{\partial F}{\partial x'} \right|_p \quad (12)$$

を付け加えることは自然である：

$$loss_{total} = loss_t + loss_c. \quad (13)$$

ここで、式(10)が2階の微分方程式であることに注意が必要である。数値的に解を定めるためには自由度の2倍の数の関係式が必要であり、十分な数の独立なデータが必要となる。例えば $F$ を観測することが困難な場合、 $loss_t$ を用いない方法が考えられるが、その場合は独立なデータ数を増やす工夫が必要である。例えば式(7)を用いる場合は、座標と座標の微分による速度のパラメータに加え、運動量と運動量の微分による加速度の情報を加えることにより、 $loss_c$ のみで解を定める工夫がされている。また、 $F$ をニューラルネットワークにより構成するというアプローチを取る場合、2階微分が必要なことから活性化関数にReLUなどの2階微分がゼロとなる関数を使用することはできないことにも留意が必要である。

### 4. 変分原理の適用例：測地線

2点間の最短経路を求める問題を考える。2点間の経路の長さは、局所的な距離 $ds$ を積分することで計算できるが、変分問題としては距離 $ds$ の極値を考えると距離の2乗 $ds^2$ を考えることは同値であるので、簡単のために $F = ds^2$ を考えることとする。すなわち作用積分は以下である：

$$S = \int (ds^2) dt \quad (14)$$

一般的には $ds^2$ は以下のように表され、リーマン計量と呼ばれる：

$$ds^2 = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} g_{ij} dx^i dx^j \quad (15)$$

ここで $g_{ij}$ は計量テンソル、 $dx^i$ は座標成分 $x_i$ に対応する接ベクトル、 $n$ は空間の次元である。オイラー・ラグランジュ方程式により $x_i$ が満たす方程式として次の測地線の微分方程式が導かれる：

$$\frac{d^2 x^i}{ds^2} + \Gamma_{jk}^i \frac{dx^j}{ds} \frac{dx^k}{ds} = 0 \quad (16)$$

ここで $\Gamma_{jk}^i$ はクリストッフエル記号と呼ばれる計量テンソルとその微分により構成される関数である。式

(16) を解くことができれば  $x^i(t)$  が求まり測地線の軌道を描くことができるが、一般的には解けるわけではない。そこで本研究では式 (16) は用いず、学習により  $F = ds^2$  を構成することで  $x^i(t)$  を求める手法を検討する。  $g_{ij}$  は一般的には  $(t, x^i, x'^i)$  の関数であるが、通常幾何的な空間を考える時は時間的には変化しない静的な空間を考えるので  $x^i$  のみの関数とするのが普通である。また、多くの場合は空間に対称性があり、式 (15) は次の形を取る：

$$ds^2 = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \delta_{ij} g_{ij} dx^i dx^j \quad (17)$$

$$= \sum_{i=0}^{n-1} g_{ii} dx^i{}^2 \quad (18)$$

例えば 2 次元直交座標系であれば変数を  $(x, y)$  として

$$ds^2 = dx^2 + dy^2 \quad (19)$$

となり、2 次元極座標系であれば変数を  $(r, \theta)$  として

$$ds^2 = dr^2 + r^2 d\theta^2 \quad (20)$$

となる。式 (18) を  $F$  と置き、パラメータとして  $t$  を取ると

$$F = \sum_{i=0}^{n-1} g_{ii} \left( \frac{dx^i}{dt} \right)^2 \quad (21)$$

$$= \sum_{i=0}^{n-1} g_{ii} (x'_i)^2 \quad (22)$$

となることから、両辺を  $x'_i$  で偏微分することで次式が得られる：

$$\frac{dx_i}{dt} = \frac{1}{2g_{ii}} \frac{\partial F}{\partial x'_i} \quad (23)$$

式 (23) は独立した  $n$  個の  $F$  の 1 次微分の条件を定める式であり、損失関数

$$loss_d = \left| \frac{1}{2g_{ii}} \frac{\partial F}{\partial x'_i} - \frac{dx_i}{dt} \right|_p \quad (24)$$

を式 (13) に加えた以下を新たな損失関数とすることができる：

$$loss_{total} = loss_t + loss_c + loss_d. \quad (25)$$

ここで、 $dx_i/dt^*$  は学習データを表す。  $F$  が学習によって構成できれば、式 (23) を自動微分した値を用いて数値的に積分することにより、変分法の目的

である  $x(t)$  の軌道を求めることができる。ただし式 (23) は

$$\frac{dx_i}{dt} = f(t, x_i, x'_i) \quad (26)$$

という  $x'_i$  について陰的な形をしているため、数値スキームに反復法を組み合わせるなどの工夫が必要である。

## 5. 実験例

直交座標系での測地線は直線になることは明らかであるが、変分原理の問題として解を求める実験を行う。前提としては、観測問題として時系列の座標と速度が観測可能とする。すなわち  $F(t, x, x')$  の学習のためのデータが得られるものとする。2 次元直交座標系では測地線は直線となることから、学習フェイズでは

$$\mathbf{x} = \mathbf{x}_0 + t\mathbf{a} \quad (27)$$

となる直線のデータより、

$$F = (t, \mathbf{x}, \mathbf{x}') \rightarrow ds^2 \quad (28)$$

という写像を学習する。比較のため、損失関数として式 (25) のうち、 $loss_t$  のみを用いて学習した場合を case 1、 $loss_t + loss_c$  を用いて学習した場合を case 2、 $loss_t + loss_c + loss_d$  を用いて学習した場合を case 3 とし、学習時の各 case での式 (11) の値を図 1、図 5、図 9 に、式 (12) の値を図 2、図 6、図 10 に、式 (24) の値を図 3、図 7、図 11 に示す。横軸はエポック数である。学習に用いたニューラルネットワークは、入力層、2 層からなる隠れ層、出力層から構成される全結合型ネットワークで、活性化関数は  $\tanh$ 、最適化アルゴリズムは Adam を使用した。ノルムは  $p = 2$ 、学習率は  $10^{-4}$ 、エポック数は 3000、隠れ層のユニット数は 32 とした。学習により構成された  $F$  を用いて変分原理の解を検証するために、推論フェイズでは解軌道が直線となる情報は使用せずに学習によって得られたスカラー関数  $F$  を利用し、式 (23) から数値解を求めた結果を図 4、図 8、図 12 に示す。オレンジで示されているのが正解データで、青で示されているのが式 (23) の数値積分により得られた解軌道である。図 4 では正解データの下方方向に解軌道が存在し、図 8 では正解データの上方方向に解軌道が存在している。図 12 では、直線が得られていることがわかる。数値積分の手法は適応的の刻み幅型の 4 次の Runge-Kutta 法を用いた。

以上の実験は、python3.7, torch1.9.0 及び周辺のライブラリを用いて実装した。

## 6. おわりに

本研究では、変分原理の問題に対しニューラルネットワークを利用することで、変分原理の目的である解軌道を求める事に対して必要な損失関数や数値積分の適用方法を提案し、いくつかの仮定のもとで実験により検証した。実験では測地線を求める問題に対し精度良く解軌道が得られることが示されたが、最も単純な2次元平面上でも問題であるため、さらに多くの問題に対し実装を進めることが必要である。今後の課題としては、様々な変分原理の問題に対する実験の実施や、ニューラルネットワークを用いずに変分原理の問題を解く手法との実用上の比較などがあげられる。

### 参考文献

- [1] S. Greydanus, M. Dzamba and J. Yosinski, Hamiltonian Neural Networks, *arXiv:1906.01563*, 2019.
- [2] T. Matsubara, A. Ishikawa and T. Yaguchi, Deep Energy-Based Modeling of Discrete-Time Physics, *arXiv:1905.08604*, 2020.
- [3] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel and S. Ho, Lagrangian Neural Networks, *arXiv:2003.04630*, 2020.
- [4] T. Aoshima, T. Matsubara and T. Yaguchi, Deep Discrete-Time Lagrangian Mechanics, ICLR2021 Workshop on Deep Learning for Simulation, 2021.
- [5] M. Finzi, K. A. Wang and A. G. Wilson, Simplifying Hamiltonian and Lagrangian Neural Networks via Explicit Constraints, *arXiv:2010.13581*, 2020.
- [6] Z. Jin, J. Y.-Y. Lin and S.-F. Li, Learning Principle of Least Action with Reinforcement Learning, *arXiv:2011.11891*, 2020.

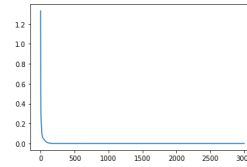


図 1:  $loss_t$  (case 1)

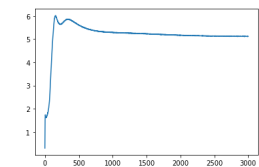


図 2:  $loss_c$  (case 1)

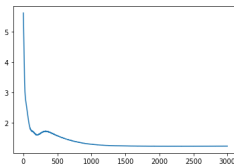


図 3:  $loss_d$  (case 1)

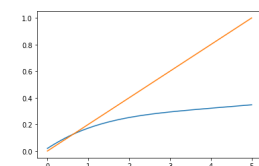


図 4: 解軌道 (case 1)

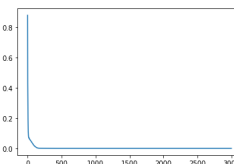


図 5:  $loss_t$  (case 2)

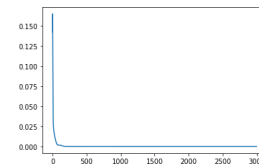


図 6:  $loss_c$  (case 2)

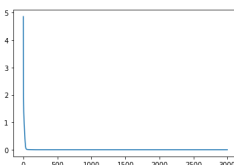


図 7:  $loss_d$  (case 2)

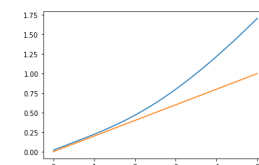


図 8: 解軌道 (case 2)

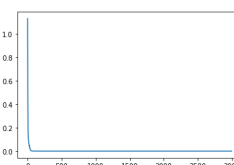


図 9:  $loss_t$  (case 3)

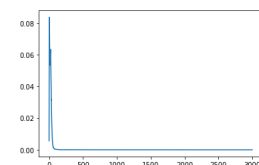


図 10:  $loss_c$  (case 3)

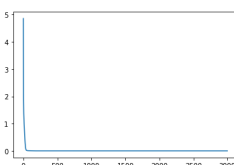


図 11:  $loss_d$  (case 3)

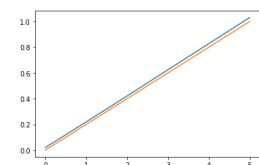


図 12: 解軌道 (case 3)