

# MIKAN: MLP Integrated Kolmogorov-Arnold Networks

日大生産工 ○ 籠原 佳希      日大生産工 山内 ゆかり

## 1. まえがき

多層パーセプトロン (Multi-Layer Perceptron: MLP) は今日の機械学習において必要不可欠なものとなっており、普遍性定理により、任意の連続関数を任意の精度で近似可能であることが知られている。一方、Ziming Liuらはコルモゴロフ・アーノルド表現定理[1]に触発され、MLPの有望な代替としてKolmogorov-Arnold Networks (KAN) [2]を提案した。KANは各エッジに学習可能な単変数関数を持ち、解釈可能性に優れていることが特徴である。また、MLPと比較して精度に対するパラメータ効率が高いことも示されている。KANは学習可能な単変数関数としてBスプライン曲線を使用しており、微分可能であるため誤差逆伝播法によって学習可能である。しかし、Bスプラインは計算コストが高く、KANはMLPと比較して約10倍の訓練時間を必要とするという問題がある。

訓練時間の高速化のため、Bスプラインの基底関数をより効率的なもので置き換える研究が既にいくつか行われている[3]。しかし、その多くはBスプラインの区分的多項式の形を継承しており、異なる形の近似手法についてはあまり検討されていない。本研究では、Bスプラインを小規模な一入力一出力のMLPで置き換えたモデルを提案する。これにより、KANが持つ解釈可能性やパラメータ効率を維持しつつ、訓練時間をMLPと同等まで高速化することを目指す。

## 2. 従来研究

### 2.1. Kolmogorov-Arnold Networks (KAN)

#### 2.1.1. コルモゴロフ・アーノルド表現定理

コルモゴロフ・アーノルド表現定理[1]は、任意の多変数の連続関数がある単変数関数の和によって表現可能であることを示す定理であり、Vladimir ArnoldとAndrey Kolmogorovによって提唱された。

$$f(\mathbf{x}) = f(x_1, \dots, x_n) \\ = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^n \phi_{q,p}(x_p) \right) \quad (1)$$

式(1)において、 $f$ は多変数の連続関数であり、 $[0, 1]^n$ から $\mathbb{R}$ への写像である。また、 $\phi_{q,p}$ 、 $\Phi_q$ は一次元の連続関数であり、それぞれ $[0, 1]^n$ から $\mathbb{R}$

のへ写像、 $\mathbb{R}$ から $\mathbb{R}$ への写像である。つまり、この式は高次元の連続関数である $f$ を、 $\phi_{q,p}$ 、 $\Phi_q$ という一次元関数の和として落とし込むことができることを示している。

#### 2.1.2. Bスプライン曲線

Bスプライン曲線は、複数の制御点とノット列によって定義される滑らかな曲線であり、以下の式によって定義される。

$$C(u) = \sum_{i=1}^n P_i B_{i,k}(u) \quad (2)$$

ここで、 $P_i$ は曲線の形状を決定する制御点であり、 $n$ は制御点の個数である。また、 $k$ は曲線の滑らかさを決定する次数である。 $B_{i,k}$ はBスプライン基底関数であり、対応する制御点を重みづけして線形結合することで、Bスプライン曲線が生成される。Figure 1にBスプライン曲線と基底関数の図を示す。

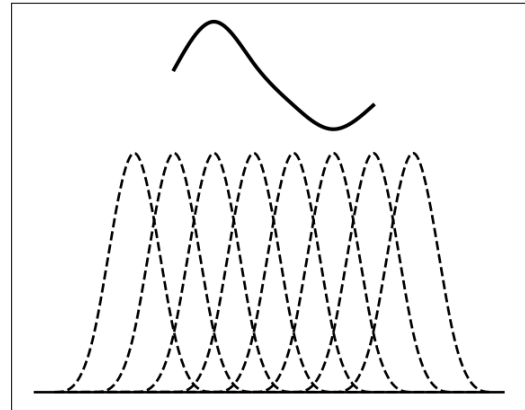


Figure 1 B スプライン曲線と基底関数

実線がBスプライン曲線であり、破線で示される釣り鐘型の山がBスプライン基底関数である。Bスプライン基底関数は以下のde Boor Cox漸化式によって再帰的に求まる。

$$B_{i,0}(u) = \begin{cases} 1 & \text{if } t_i \leq u \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ B_{i,k}(u) \\ = \frac{t - t_i}{t_{i+k} - t_i} B_{i,k-1}(u) \\ + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(u) \quad (3)$$

ここで、 $\{t_i\}$ はノット列であり、基底関数が値を持つ範囲を定めている。制御点の影響を受ける範囲はノット列によって制限されているため、1つの制御点を動かしても、曲線全体でなく、制御点付近のみが変化する。この局所的な特性は、Bスプラインの大きな特徴である。

### 2.1.3. KANのアーキテクチャ

KAN[2]はコルモゴロフ・アーノルド表現定理をネットワーク化したモデルである。しかし、式(1)通りにネットワーク化しようとする、その構造は深さ2、幅 $2n+1$ となり、モデルのサイズが固定されてしまう。それでは表現力が限定されるため、KANでは以下のように任意の幅、深さとして一般化する。

$$f(x) = \sum_{i_{L-1}=1}^{n_{L-1}} \phi_{L-1,i_{L-1}} \left( \sum_{i_{L-2}=1}^{n_{L-2}} \dots \left( \sum_{i_1=1}^{n_1} \phi_{1,i_1} \left( \sum_{i_0=1}^{n_0} \phi_{0,i_0} i_0(x_{i_0}) \right) \right) \dots \right) \quad (4)$$

また、 $\phi_{l,j,i}$ として示される一次元関数はBスプライン曲線によって表現し、Bスプラインの制御点を学習可能なパラメータとして設定する。

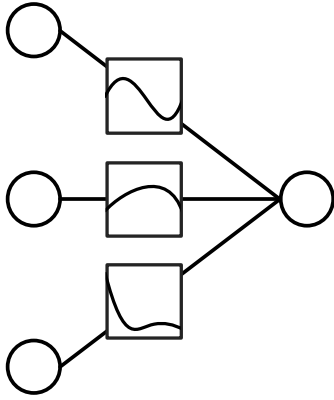


Figure 2 KANのエッジの概略

以上の議論から、Figure 2に示すように、KANは各エッジに一次元関数を持つネットワークモデルであるとみなせる。KANでは、各エッジの関数 $\phi(x)$ を以下のように設定する。

$$\phi(x) = w_b b(x) + w_s spline(x) \quad (5)$$

ここで $b(x)$ は残差接続に似た役割を持ち、以下のようにSiLU関数を設定している。

$$b(x) = silu(x) = \frac{x}{1 + e^{-x}} \quad (6)$$

また、 $spline(x)$ はBスプラインであり、以下の式で定義される。

$$spline(x) = \sum_i c_i B_i(x) \quad (7)$$

$B_i$ はBスプライン基底関数であり、式(3)によって求まる。KANが学習するパラメータは

$w_b, w_s, c_i$ であり、MLPと同様に逆伝播によってモデル全体の学習を行う。

KANではエッジに持つ関数を可視化することができるため、MLPと比較して解釈可能性が高いことが大きな特徴である。またBスプラインの局所的な特性により、破滅的忘却への耐性が高いこともKANの優れた特徴の一つである。

### 2.2. FastKAN

Bスプライン基底関数は、式(3)に示されるように再帰的な計算を必要とするため、計算コストが高いという問題があった。FastKAN[3]では、Bスプライン基底関数をガウシアン放射基底関数で置き換えることで計算効率の向上を達成した。ガウシアン放射基底関数は、以下のように再帰を必要としない形で、3次のBスプライン基底関数を近似することができる。

$$B_i(x) = \exp\left(-\left(\frac{x - c_i}{h}\right)^2\right) \quad (8)$$

ここで $\{c_i\}$ は中心点列であり、Bスプラインにおけるノット列と似た役割を持つ。 $h$ は基底関数の広がり制御する任意のパラメータである。

## 3. 提案手法

Bスプライン基底関数を置き換える形でのKANの高速化手法が多く提案されてきたが、本研究ではMLPにより一次元関数 $\phi(x)$ そのものを置き換えたモデルを提案する。このモデルを、MLP-Integrated KAN (MLPを統合したKAN) より、MIKANと呼ぶ。

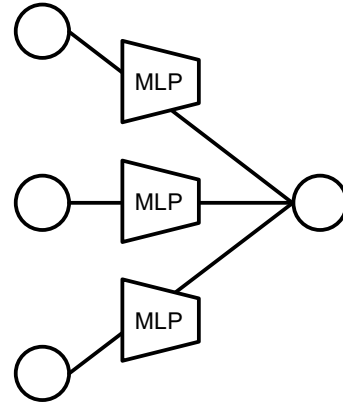


Figure 3 MIKANのエッジの概略

Figure 3に示すように、各エッジの関数を入力が次元である小規模なMLPによって表現する。このMLPをEdgeMLPとし、その構成をFigure 4に示す。EdgeMLPは隠れ層が1層のMLPであり、中間次元をパラメータDとする。

MLPは普遍性定理により、任意の連続関数を任意の精度で近似可能であるため、KANのエッジ上の関数もまたMLPにより近似可能であると考えられる。また、KANはMLP同様に逆伝播

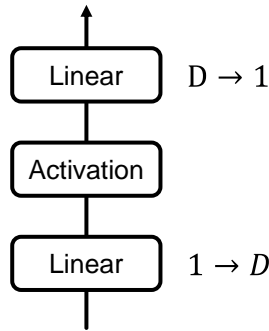


Figure 4 EdgeMLP の構成

により学習するため、モデル全体を一般的な深層学習の枠組みで学習を進めることができる。更に、パラメータの初期化や正則化手法等、MLP向けの拡張を取り入れることが可能となり、既存の知見を活かしてモデルを設計できるようになることが期待される。

#### 4. 実験および検討

データセットにMNISTを用いた画像分類タスクにおいて、MIKANとFastKANを比較し、訓練時間と精度を評価する。実験では、FastKANの制御点数を13、EdgeMLPの中間次元Dを4、活性化関数をSiLUとする。このとき、EdgeMLP1つのパラメータ数は重みが8、バイアスが5で合わせて13となり、FastKANと等しくなる。またFastKAN、MIKAN共に、各層のノード数は(784, 64, 10)であり、学習率は0.005である。加えて、中間層ではレイヤー正則化を適用している。

各モデルを20エポック学習し、10回平均を行った結果をTable 1に示す。

Table 1 MNISTでの実験結果

| Model   | Train Acc | Test Acc | Time  |
|---------|-----------|----------|-------|
| FastKAN | 1.000     | 0.970    | 12964 |
| MIKAN   | 0.882     | 0.883    | 1209  |

提案手法であるMIKANはFastKANに対し、訓練時間において約10倍の高速化を達成した。しかし、MIKANはFastKANと比較して、テスト精度が約9%低下してしまった。これは、入力が1次元である小規模なMLPでは表現力でBスプラインに劣っていることが原因であると考えられる。

Figure 5にFastKANとMIKANのテスト精度の推移を示す。

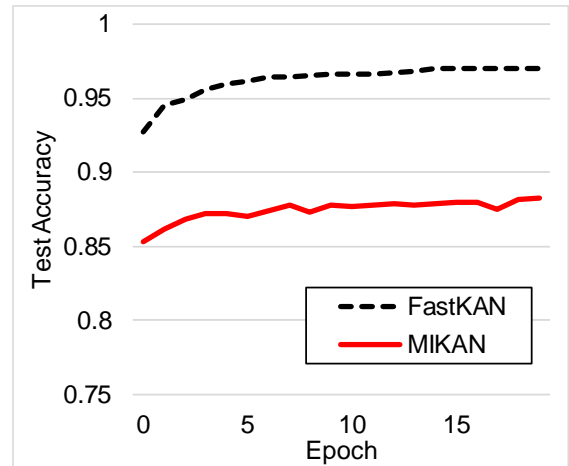


Figure 5 テスト精度の推移

MIKANは低い精度を示すだけでなく、精度の推移が上下しており、FastKANと比較して学習が不安定である。

#### 5. まとめ

本研究では、KANにおけるエッジ上の一次元関数をMLPにより表現することを提案した。実験より、提案手法は従来のFastKANから大幅な高速化を達成した。一方で精度が大きく減少し、学習の不安定さも示された。

本研究により、一次元関数の近似において、単純なMLPではBスプラインに表現力で劣ることが推察できた。更に、Bスプラインの持つ局所性もMLPでは失われてしまうため、現状のEdgeMLPではBスプラインの代替とするには不十分であると考えられる。今後、MLPとしての形を崩さず、訓練時間の優位性を保ったまま、従来のモデルに対し精度で劣らないようなEdgeMLPの構成を検討したい。

#### 参考文献

- 1) Andrei Nikolaevich Kolmogorov, “On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition”, In Doklady Akademii Nauk, volume 114, pages 953–956. Russian Academy of Sciences (1957).
- 2) Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou and Max Tegmark, “KAN: Kolmogorov-Arnold Networks”, arXiv: 2404.19756 (2024).
- 3) Ziyao Li, “Kolmogorov-Arnold Networks are Radial Basis Function Networks”, arXiv: 2405.06721 (2024).