

## フーリエ密度モデルのパラメータ設定と画像圧縮への応用 Parameterization of Fourier Density Model and Its Application to Image Compression

大貫 創悟<sup>†</sup> 孫 鶴鳴<sup>†</sup>  
Sogo Ohnuki Heming Sun

### 1. はじめに

近年のデータ圧縮技術として深層学習を応用する非可逆圧縮が研究されている。深層学習による最適化対象に、主には入力信号の畳み込み圧縮を行うためのパラメータが挙げられるが、畳み込み後の潜在変数は未知のものであるから、値の出現確率分布がわからない。そこで、符号化器に渡す確率分布を同時に学習と最適化によって求める。実際の変数分布に適切にフィッティングされた確率密度関数を求めることができれば、それを符号化器に渡して、より圧縮率の高いモデルの学習が可能となる。推定される分布が実際の分布から離れるほど圧縮率の指標で損失となり、KL Divergence によって情報量損失の大きさを計算することができる。確率密度関数にも畳み込み層と同様に学習パラメータを設定する必要があるが、これは数理モデルや指定する事前分布によってきまる。学習によって数理モデル内のパラメータを決定するから、これが少ないほど学習に係るコストも小さい。

分布の種類が確定しない潜在変数に対して分布推定をしなければならぬ場合は、例えば正規分布やラプラス分布などのパラメトリックモデルとして定義したモデルでは適切な分布推定ができない。代わりに、事前分布のない数理モデルや、理論的にどのような関数にも近似できる数理モデルを使用する必要がある。事前分布のない数理モデルとしては、畳み込みニューラルネットワークを利用したノンパラメトリックモデルが挙げられる。

本研究では、パラメータ数を増加させることによって任意の関数への近似を可能とするフーリエ密度モデルの実装を行った。確率密度関数を複素フーリエ変換による近似によって表現し、入力変数を一周期内にマッピングをすることで左右端のある任意の関数を実現する。これを、特定の関数グラフと学習型画像圧縮の中間層出力の潜在変数分布を対象にした分布推定に用いることを考える。畳み込みニューラルネットワークにおいて多くの並列処理を行う場合、計算量が大きく増加する。中間層出力の圧縮可能性により記憶データを少なくし、その負荷を減少することが期待される。しかし、中間層出力は画像圧縮モデルの評価としては圧縮の対象とならず、分布が最適化されない。そのため、最終層出力よりも複雑でエネルギーの大きな分布を持つことになり[4]、適切な分布推定をすることの難しさが課題となる。

フーリエ密度モデルと同様に任意の関数にフィッティング可能な特徴を持つ、ニューラルネットワークを用いたノンパラメトリックモデルと比較して確率分布推定の性能を確かめる。

### 2. フーリエ密度モデル

本論文では、De la Fuente ら[3]の式によって、確率密度関数としてフーリエ密度モデルを定義する。フーリエ密度モデルは、ある一定の範囲の目的関数について、周期の異なる波形の合成によって近似を行う方法である。パラメータ数を増加させたとき、その無限大極限で任意の関数に対して近似可能となる。

#### 2.1 理論

近似によって確率分布関数 $p(x)$ を定義する。入力確率変数 $x$ は実数である。関数の近似範囲を $(-1, 1)$ としたとき、任意の関数 $f(x)$ は複素フーリエ変換によって以下の式で表される。

$$f(x) = \sum_{n=-N}^N c_n \exp(in\pi x) \quad (1)$$

ここで、複素の定数 $c_n$ について、

$$c_n = \frac{1}{2} \int_{-1}^1 f(x) \exp(-in\pi x) dx \quad (2)$$

$c_n = c_n^*$ であり、これによって関数 $f(x)$ は実数の範囲であることに注意する。 $c^*$ は複素数 $c$ の複素共役である。また、 $N$ の設定により近似制度を調整し、これを大きくすることによって詳細に近似を行うようにする。

確率分布関数 $p(x)$ は以上の定義のほかに複数の条件を満たしている必要がある。非負性、正規化、実数全体の入力である。

#### 2.1.1 非負性

関数が非負であるために、定数 $c_n$ に条件を設け定義する。ヘルグロッツの定理により、 $f(x)$ が正半定値であるとき、非負性が認められる。 $f(x)$ が正半定値であるためには、 $c_n$ も正半定値である必要がある。複素数列における自己相関数列は正半定値であるため、定数 $c_n$ を新たな複素の定数 $a_n$ を用いて次のように定義しなおす。

$$c_n = \sum_{k=0}^{N-n} a_k a_{k+n}^* \quad (3)$$

$c_n = c_n^*$ であることから、 $a_n$ を $n = 0, \dots, N$ について定義すればよい。学習可能パラメータは $a_n$ の実部と虚部にあたり、その数は $2(N+1)$ である。

<sup>†</sup> 横浜国立大学 Yokohama National University

2.1.2 正規化

確率分布関数 $p(x)$ は、全ての範囲での積分値が 1 である必要がある.  $(-1, 1)$ において、積分値すなわち正規化定数 $Z$ は次のようになる.

$$Z = \int_{-1}^1 f(x)dx = 2c_0 \quad (4)$$

よって、確率分布関数 $p(x)$ は、

$$p(x) = \frac{f(x)}{Z} = \frac{1}{2} + \sum_{n=1}^N \frac{c_n}{c_0} \exp(in\pi x) \quad (5)$$

これを積分することで累積密度関数 $P(x)$ を得ることができる.

$$P(x) = \frac{x}{2} + \sum_{n=1}^N \frac{c_n}{\pi i n c_0} \exp(in\pi x) + C \quad (6)$$

ただし、積分定数 $C$ はその他の定数により、 $P(-1) = 0$ かつ $P(1) = 1$ となるように設定される.

2.1.3 実数全体の入力

ここまでは入力変数の範囲として $[-1, 1]$ について考えている. 入力となる潜在変数は未知の分布であるから、実数全体を入力の範囲とし、マッピングすることを考える必要がある. まず、 $[-1, 1]$ と実数全体を対応させるための関数を定義する.

$$g(x) = \tanh^{-1}(x) \quad (7)$$

$$g^{-1}(x) = \tanh(x) \quad (8)$$

これを用いて新たに、入力が実数全体に拡張された累積密度関数 $Q(x)$ を次のように表す.

$$Q(x) = P(g^{-1}(x)) \quad (9)$$

これを微分して、入力が実数全体に拡張された確率分布関数 $q(x)$ を次のように表す.

$$q(x) = p(\tanh(x)) \cdot \text{sech}^2(x) \quad (10)$$

これらの関数は、入力として実数全体を期待し、出力として $[0, 1]$ の範囲を得る.  $Q(x)$ は単純増加関数であることと、 $\max(Q(x)) - \min(Q(x)) = 1$ であることが定義から保証されている.

3. 画像圧縮への応用

基本的な学習型画像圧縮モデルの学習は、畳み込み $g_a$ 、量子化 $Q$ 、変数分布推定(EntropyBottleneck)、逆畳み込み $g_s$ によって行われる[図 3-1]. 入力画像は畳み込みの過程において、複数の畳み込み層を持ち、それぞれの層の出力は複数のチャンネルに分解される. 最終層出力を量子化したものについて確率分布推定を行い、エントロピーが最小となるように学習された関数を用いて各チャンネルの Bit-stream を生成している. 圧縮モデル性能の評価としては畳み込み中

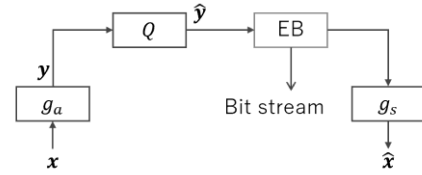


図 3-1 学習型画像圧縮モデル

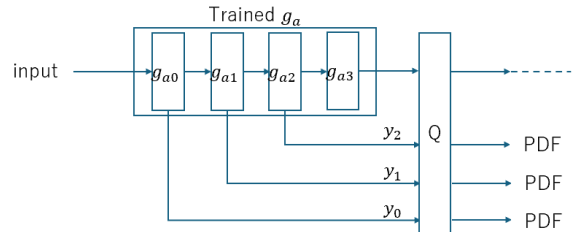


図 3-2 中間層出力の学習

CompressAI より Factorized モデルの中間層出力を学習の正解グラフに用いる. モデルの学習は、畳み込み $g_a$ 、量子化 $Q$ 、変数分布推定(EntropyBottleneck)、逆畳み込み $g_s$ によって行われ、出力画像はその過程で損失を受ける. Factorized モデルは $g_a$ の内容に畳み込み層を 4 層もち、本研究では中間層出力としてその 3 箇所について分布を調査する.

表 4-1 学習諸元

Iteration	10000
Learning rate	1(FBM), 1e-1(factorized)
Loss function	KL Divergence
Dataset	20 images from imagenet256

間層の出力は圧縮することを考えなくてよい. しかし、多くのチャンネルを利用するなど並行処理するタスクが増加すれば、一時的に記憶される中間層出力を圧縮するための数理モデルが役立つ.

高圧縮率のためには符号化器に渡す入力変数の正確な確率分布関数が必要である. 畳み込み中間層の出力は圧縮モデルの評価には影響しないことから、最終層出力よりも平均してエネルギーの高い分布となり、特に保持する情報が多くなるチャンネルが存在する. 数理モデルにはより複雑な分布に対しても適切に確率分布推定ができる能力が求められる[図 3-2].

4. 実験

実験は以下の 2 つの内容に従って行う.

- ① 任意に指定する確率分布関数へのフィッティングを行い、モデルの性能を確かめる.
- ② 画像圧縮への応用として PyTorch ライブラリより CompressAI から学習済みの Factorized モデルを活用し、畳み込みニューラルネットワーク内の中間層出力にあたる変数分布に対して、数理モデルまたはパラメータ設定の異なりによるフィッティング性能を調査し、評価する.

表 4-2 学習モデル

Model	Base	Option	Parameters
FBM64	FBM	N=64	130
FBM32	FBM	N=32	66
FBM16	FBM	N=16	34
Fac5	Factorized	5 layers	129
Fac4	Factorized	4 layers	58
Fac3	Factorized	3 layers	31

#### 4.1 モデルの提案

フーリエ密度モデル(FBM)を実装し、学習を行う。対照として学習済み画像圧縮モデルの確率分布推定に用いられるノンパラメトリックモデルである Factorized モデルをあげ、同様に実装して比較する。学習可能パラメータの数が同程度となるような FBM と Factorized モデルの組を複数定義し、3 セット合計 6 モデルを実装して学習を行う[表 4-2]。

#### 4.2 学習

実験①では学習対象として混合正規分布を用いた。混合数を 2, 3 として、任意のパラメータを設定して学習を行った。また、範囲を[-1, 1]として、これを 256 等分するものとした。

実験②では学習対象として、Quality 3 における学習済み Factorized モデルの中間層出力のうち、最もエネルギーの高いチャンネルの分布を選択した。Factorized モデルは 4 つの畳み込み層を持つため、その中間出力を  $y_0, y_1, y_2$  とし、それぞれの分布推定のための 3 つのモデルの学習を同時に行った。また、目的の分布は  $1/2^{16}$  スケールに量子化し、入力は範囲を 256 等分するものとした。

すべての実験で、損失関数に KL Divergence を用いた。正解の確率分布関数  $p$  と学習された確率分布関数  $q$  について次のように定義する。

$$\text{KLD} = \sum_i p_i \log \frac{p_i}{q_i} \quad (11)$$

学習対象は PDF となるから、学習モデルは式(10)とする。また、実験②におけるデータセットに imagenet256 より 20 枚のデータを使用し、それらを用いたときの学習済み中間層出力から正解グラフとなる分布を作成する。

### 5. 結果

#### 5.1 実験①：任意の関数へのフィッティング

正解グラフに混合正規分布(混合数 2, 3)を用いて出力される確率密度関数を比較した[図 5-1]。FBM は Factorized モデルに対して明らかに良いフィッティング精度を得た。Factorized モデルは起伏の大きな関数になればフィッティングが難しくなるが、FBM では正解グラフにおける混合数が増加したときも変わらない精度を保つことができた。このことから、実験①に使用したグラフ程度の複雑さであれば、今回定義した FBM はよく性能を発揮できていると考えられ、より複雑な分布に対してもよい結果を得ることが期待できる。

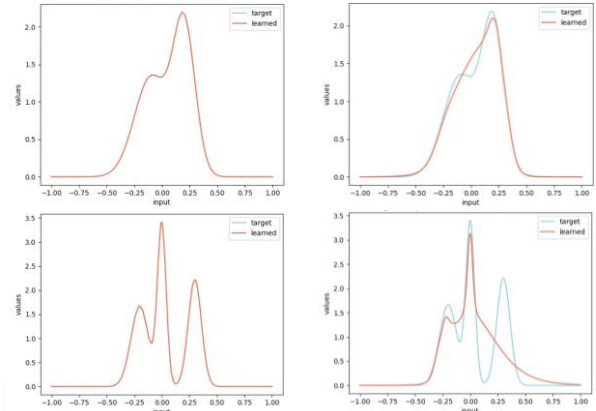


図 5-1 任意の分布の学習 FBM64(左) Fac5(右)

正解グラフは混合正規分布(混合数 2(上), 3(下))を任意のパラメータで設定し、横軸に 256 点をとった。FBM は正解グラフとほとんど重なるように学習され、適切な確率分布の学習ができたと言える。

表 5-1 任意の分布の学習損失

Model	Parameters	KLD	
		2 Gaussian	3 Gaussian
FBM64	128	5.33 e-5	6.24 e-5
FBM32	64	6.02 e-5	5.75 e-5
Fac5	126	3.42 e-3	1.27 e-1
Fac3	58	3.27 e-3	9.09 e-2

KL Divergence による結果の比較では、FBM と Factorized モデルで明らかな違いが確認された。また、FBM は正解グラフの混合数が増加し、複雑さが増したときでも、ほとんど結果を悪化させないことがわかる。

#### 5.2 実験②：画像圧縮中間層出力の分布推定

目的関数に学習済み Factorized モデルの中間層出力を用いた。中間層  $g_{a0}, g_{a1}, g_{a2}$  に対して、Quality 3 では出力が 128 (0,1,...,127)チャンネルの特徴マップに分解され、それぞれ異なる分布を持つ。これらのチャンネルのうち、最もエネルギーの高いチャンネルを求め、その分布を推定することとする。エネルギー  $E$  はチャンネル内の各要素  $p$  に対して次式により計算される。

$$E = \sum_{i,j} p_{i,j}^2 \quad (12)$$

数理モデルの学習に先んじて、データセットに対する中間層出力を収集しておく。これらの出力に対して上式を適用し、チャンネルごとに比較した。これにより、推定対象チャンネルを 103:  $g_{a0}$ , 7:  $g_{a1}$ , 30:  $g_{a2}$  とし、正解グラフを設定した。

学習結果は図の通り[図 5-2]。FBM は Factorized モデルよりもよい精度であると言える。特に FBM64 で、横軸に小さな幅を持つ入力データのそれぞれに対してグラフが振れていることが見て取れ、特にパラメータ数を増加させたときに急峻な変化に対応するようなグラフの形状を確認することができる。大きい N ではより周波数の高い波形の混合を試みることによるものと考えることができ、学習対象の

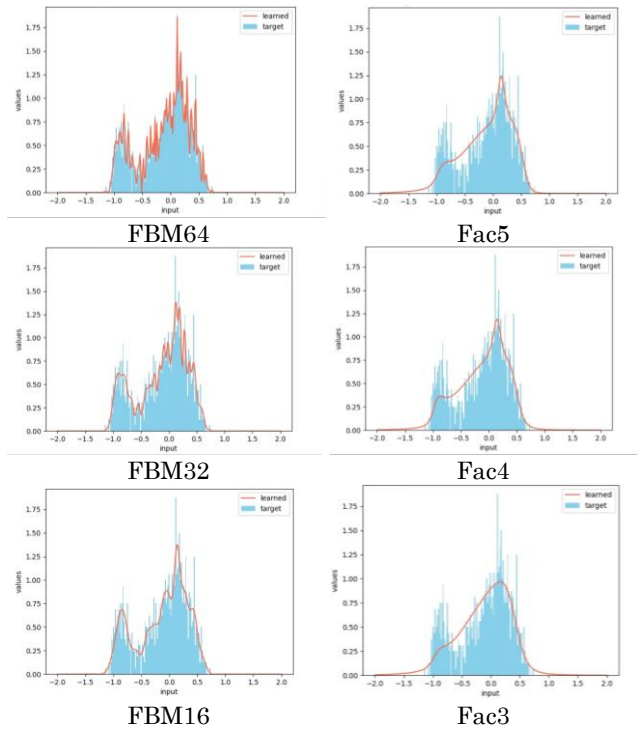


図 5-2 中間層出力の分布推定

学習済み Factorized モデルの中間層出力のうち、最もエネルギーの高いチャンネルの分布を取り出し、分布推定を行った。図は  $y_2$  出力のチャンネル 30 の分布についてモデルの学習である。入力範囲の 256 分割とし、値は  $1/2^{16}$  スケールに量子化した。FBM は Factorized モデルに対してよりよいフィッティング精度を得た。また、パラメータ数の増加に伴って FBM はより起伏の大きなグラフを生成することができるようになり、FBM64 では最も入力に対して細かくフィッティングしていることがわかる。

横軸の解像度により、効果を得る。対して Factorized モデルの Fac4 と Fac5 では学習パラメータ数に約 2 倍の差があるが、グラフの形状と KL Divergence の結果はほぼ同様である[表 5-2]。

FBM は学習パラメータ数を増加させることによる KL Divergence による損失の指標により、精度の向上が大きいことが確かめられた[図 5-3]。FBM の KLD Loss の結果は Factorized モデルと比較し、パラメータ数約 30 では 0.39 倍、パラメータ数約 60 では 0.33 倍、パラメータ数約 130 では 0.18 倍となった。このことから、パラメータ増加の効果は FBM で大きいといえる。FBM は N の無限大極限で理論的に任意の関数に近似可能であるから、確かに近似性能が向上している。

## 6. おわりに

本研究では、確率分布推定にフーリエ密度モデルを実装して学習を行い、その有効性を確かめた。また、そのパラメータ数を変更することによって Factorized モデルよりも KL Divergence を小さくし、複雑なモデルにフィッティング可能であることが確かめられた。FBM のパラメータ増加によるフィッティング精度の向上性から、さらなるパラメータ数増加やその設定と分析により、よりよい精度を得ることや、より複雑な分布推定を行うことへの効果が期待される。

表 5-2 中間層出力の KLD Loss

Model	$y_0$ KLD	$y_1$ KLD	$y_2$ KLD	KLD sum
FBM64	$4.08 \text{ e-}3$	$2.41 \text{ e-}3$	$4.61 \text{ e-}3$	$1.11 \text{ e-}2$
FBM32	$5.76 \text{ e-}3$	$4.21 \text{ e-}3$	$1.18 \text{ e-}2$	$2.13 \text{ e-}2$
FBM16	$1.92 \text{ e-}2$	$7.76 \text{ e-}3$	$1.42 \text{ e-}2$	$4.11 \text{ e-}2$
Fac5	$1.74 \text{ e-}2$	$1.39 \text{ e-}2$	$2.96 \text{ e-}2$	$6.09 \text{ e-}2$
Fac4	$2.28 \text{ e-}2$	$1.35 \text{ e-}2$	$2.74 \text{ e-}2$	$6.37 \text{ e-}2$
Fac3	$4.33 \text{ e-}2$	$2.55 \text{ e-}2$	$3.70 \text{ e-}2$	$1.06 \text{ e-}1$

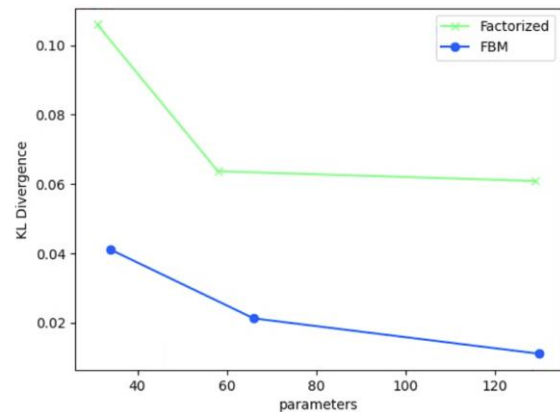


図 5-3 パラメータ数 - KLD Loss 比較

パラメータ数と中間層出力 3 つ[表 5-2]の KLD sum を比較している。特にパラメータ数が近くなるような 3 セット合計 6 モデルの学習結果である。FBM 少ないパラメータでも KL Divergence が小さくなるが、特に学習パラメータを増加したときに、フィッティング性能を向上したことがわかる。Factorized モデルはパラメータ数を大きく増加させたときの結果の改善が FBM よりも小さくなった。

今後の展開として、異なる Quality の分布を確認した確率分布推定の実施と、単一でないチャンネルにおける分布推定を考えるほか、定義の見直しによる学習パラメータの追加を考え、より正確な推定可能性を得ることを目指す。また、画像圧縮のための最終層出力分布推定としてモデルの実装を行い、既存研究における Factorized モデルと比較した画像圧縮モデルの性能評価を行うために学習を行っている。

## 謝辞

本研究は JSPS 科研費 JP23K16861, 電気通信普及財団研究調査助成, SCAT 研究費助成による支援を受けています。

## 参考文献

- [1] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, Nick Johnston: "Variational Image Compression with a Scale Hyperprior", ICLR 2018
- [2] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, Jiro Katto "Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules", CVPR 2020
- [3] Alfredo De la Fuente, Saurabh Singh, Johannes Ballé: "Fourier Basis Density Model", PCS 2024
- [4] Heming Sun, Lu Yu, Jiro Katto: "End-to-End Learned Image Compression with Quantized Weights and Activations", arXiv 2021