ニューラルネットワークにおける新しい活性化関数の構成

日大生産工(院)	○榎本	知幸
日大生産工	角田	和彦
日大生産工	三浦 慎	〔一郎

1 はじめに

現在、CNNにおける畳み込み層の活性化関数として、現在、ReLU関数^[1]が幅広く使われているが、そのほかにもPReLU関数^[2]やELU 関数^[3]など新しい活性化関数が提案されており、どの活性化関数がどのネットワークにおいて最適であるかは分かっていないことが多い。

本研究では、流体力学的立場から得られる非 線形活性化関数をReLU関数に組み込み、負領 域が非線形のパラメータを持つ新しい活性化 関数を提案する。また、その活性化関数を、画 像認識を行うCNNに適用することで、活性化 関数の負領域が非線形である時の特性を明ら かにすることを目的としている。

2 畳み込み層における活性化関数

CNNにおける畳み込み層の活性化関数とし て、現在ではReLU関数が幅広く使われている。 この関数は入力値vが入力されると、max(0,v) を出力値として返す。ReLU関数は勾配消失問 題を軽減することが出来るという特徴があり、 畳み込み層の活性化関数に適用することでシ グモイド関数やtanh関数よりも良い結果が得 られている。

また、ReLU関数の負の領域を拡張した、 PReLU関数が提案されている。PReLU関数は 次のように示される。

$$f(v) = \begin{cases} v \ (v > \mathbf{0}) \\ av \ (v \le \mathbf{0}) \end{cases}$$
(1)

ただし、パラメータaは逆伝搬により学習し、 活性化関数は最適化される。PReLU 関数は、 層ごとに最適な関数に学習することで、ReLU 関数よりも優れた結果を残した例も報告され ている^[2]。また、負の値をとることで、ReLU 関数におけるユニットが死んでしまう問題点 も改善されている。しかし、非活性化領域で飽 和しない為、ノイズに弱いという問題点も指摘 されている[4]。

また、ReLU関数の負領域を非線形的に拡張 したELU関数も提案されており、次のように 示される。

$$f(v) = \begin{cases} v & (v > 0) \\ \alpha(e^{v} - 1) & (v \le 0) \end{cases}$$
(2)

ただし、αは定数。この関数の負領域は一定量 で飽和している為ノイズに強く、さらにユニッ トが死んでしまう問題も改善されている。

3 新しい活性化関数の提案

文献[5]で以下の活性化関数が示されている。

$$f(\boldsymbol{v}) = \begin{cases} \frac{1}{2} \left(2 - \frac{1}{1 + |\boldsymbol{\gamma}|} \right) & (\boldsymbol{v} > \mathbf{0}) \\ \frac{1}{2} \left(\frac{1}{1 + |\boldsymbol{\gamma}|} \right) & (\boldsymbol{v} \le \mathbf{0}) \end{cases}$$
(3)

ただし、 γ = v/2k, k>0。

この活性化関数では、ネットワーク温度kを 用いて、ステップ関数からシグモイド関数を近 似することができる。

そこで、負の入力に対する式に注目し、入力 値vをかけ、係数を除去することで次の式が得 られる。

$$g(v) = \frac{kv}{k-v} \quad (v < \mathbf{0}) \tag{4}$$

この式は $k \rightarrow 0$ の極限で傾き $0, k \rightarrow +\infty$ の極限で傾き1の線形関数に収束する。これを ReLU関数の負の入力における式に適用し、ネットワーク温度kを、学習パラメータaと置く ことで次の活性化関数が得られる。

$$h^{\wedge}(v) = \begin{cases} v & (v \ge \mathbf{0}) \\ \frac{av}{a-v} & (v < \mathbf{0}) \end{cases}$$
(5)

Construction of New Activation Functions in Neural Network

Tomoyuki ENOMOTO, Kazuhiko KAKUDA and Shinichiro MIURA

6-4

-601 -

また、式(4)のaをexp(a)に置換することで次の 活性化関数が得られる。

$$h(v) = \begin{cases} v & (v \ge 0) \\ \frac{e^a v}{e^a - v} & (v < 0) \end{cases}$$
(6)

この活性化関数は、 $a \rightarrow -\infty$ の極限で ReLU関数に収束し、 $a \rightarrow +\infty$ の極限で傾き1 の線形関数に収束する(図1)。



図1 本研究の活性化関数(Exponential型)

逆伝搬学習のために使われるvの偏微分は次 式で示される。

$$\frac{\partial h(v)}{\partial v} = \begin{cases} 1 & (v \ge 0) \\ \left(\frac{e^a}{e^a - v}\right)^2 & (v < 0) \end{cases}$$
(7)

また、活性化関数におけるaによる偏微分は 次のように示される。

$$\frac{\partial h(v)}{\partial a} = \begin{cases} 0 & (v \ge 0) \\ -\frac{e^a v^2}{(e^a - v)^2} & (v < 0) \end{cases}$$
(8)

CNN への実装時は、各層、もしくは各カーネ ルで共有するパラメータ*a*_iを導入する。 それに従い、確率的勾配降下法では誤差関数 E によりパラメータ*a*_iは、次のように更新される。

$$\Delta a_i := \varepsilon \frac{\partial \mathbf{E}}{\partial a_i} \tag{9}$$

$$\frac{\partial E}{\partial a_i} = \sum_{v_i} \frac{\partial E}{\partial h(v_i)} \frac{\partial h(v_i)}{\partial a_i}$$
(10)

ただし、 ε は学習率、 $\sum v_i$ は a_i に対応する全ての v_i における合計を表す。

本研究では、式(5)の活性化関数をRational 型、式(6)の活性化関数をExponential型と し、この二つの活性化関数と既存の活性化関 数を比較していく。

4 CIFAR-10での数値計算

ディープラーニングフレームワーク Tensorflowを使用し、ReLU関数とPReLU関数、本研究の活性化関数(Rational型と Exponential型)をCNNの畳み込み層に組み込み、一般物体データセットCIFAR-10を用いて それぞれ学習させた。実験に用いる畳み込み層 の構造を図2に示す。



図2 数値計算に用いるネットワークの構造

3つの畳み込み層の後にそれぞれの活性化関数を入れ、計算、比較を行う。損失は交差エントロピーを使って算出し、それを用いて学習を行う。また、入力画像はクリッピングや左右反転、明るさ、コントラストをランダムに変更することでデータの水増しを行っている。

ミニバッチサイズは100、学習アルゴリズム はAdam^[6]、epochは200、活性化関数のパラメ ータaは層ごとに共有し、初期値としてPReLU 関数では0.25、本研究の活性化関数では Rational型は1.0、Exponential型は0.0と設定 した。また、ELU関数の定数αは1.0とした。 のそれぞれの関数の訓練データにおける損失 の推移を図3に、検証データにおける損失を図 4に示す。次に、それぞれの関数の訓練データ における正解率の推移を図5に、検証データに おける正解率の推移を図6に示す。また、学習 後における各関数のパラメータαの値を表1に、 学習後のテストデータにおける損失と正解率 を表2に示す。



図3 訓練データにおける損失(CIFAR-10)



図4 検証データにおける損失(CIFAR-10)



図5 訓練データにおける正解率(CIFAR-10)



図6 検証データにおける正解率(CIFAR-10)

表1 CIFAR-10における学習後のパラメータa

	PReLU	Present	Present
		(Rational)	(Exponential)
init	0.250	1.00	0.00
conv1	0.056	0.118	-2.11
conv2	0.027	0.103	-2.03
conv3	0.030	0.206	-1.66

表 2 CIFAR-10 における損失と正解率

	Loss	Accuracy[%]
ReLU	0.591	77.7
ELU	0.504	80.4
PReLU	0.547	79.8
Present(Rational)	0.389	80.2
Present(Exponential)	0.399	80.8

4 CIFAR-100での数値計算

また、図2に示されるネットワークの出力層 を10から100に変え、畳み込み層のチャネル数 を32から64に変更し、一般物体データセット CIFAR-100を用いてそれぞれ学習させた。た だし、出力層、チャネル数以外のハイパーパラ メータはCIFAR-10と同じとした。

それぞれの関数の訓練データにおける損失 の推移を図7に、検証データにおける損失の推 移を図8に示す。また、学習後におけるテスト データにおける損失と正解率を表3に示す。



図7 訓練データにおける損失(CIFAR-100)



図8	検証アー	-9	におけ	る損矢(CIFAR-	100)
----	------	----	-----	------	--------	------

表3	CIFAR-100における損失と正解率

	Loss	Accuracy[%]
ReLU	1.83	55.0
ELU	1.72	56.1
PReLU	1.81	56.6
Present(Rational)	1.66	56.9
Present(Exponential)	1.55	56.5

5 おわりに

本研究では、ネットワーク温度による関数の 分布の変化といった特徴を持つ、流体力学的立 場から構成される活性化関数を、ReLU関数に 組み込み、新しい活性化関数を提示した。

今回の実験結果ではCIFAR-10、CIFAR-100 ともに本研究の活性化関数においてELU関数 やPReLU関数よりも低い損失が確認できた。 今後はCNNだけではなく様々なニューラルネ ットワークを用いて、本研究の活性化関数の負 領域が非線形であることから生じる、PReLU 関数やELU関数を用いたネットワークとの学 習の違いについて、さらに詳しく調べていきた い。

「参考文献」

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", Proc. of the IEEE, (1998), pp. 2278-2324. [2] V. Nair and G. E. Hinton, "Rectified Linear Units Improverestricted Boltzmann Machines", In ICML, (2010), pp. 807-814. [3] K. He, X. Zhang, S. Ren, J. Sun. "Delving Deep into Rectifiers", arXiv:1502.01852v1 [cs.CV] 6 Feb 2015. [4]D. Clevert, T. Unterthiner, S. Hochreiter, "Fast And Accurate Deep Network Learning by Exponential Linear Units (ELUs)", arXiv:1511.07289v5 [cs.LG] 22 Feb 2016. [5]角田和彦, "ニューラルネットの特性関数 とその近似関数",情報処理学会第67回全国 大会, (2005), pp. 239-240. [6] Diederik P. Kingma, Jimmy Lei Ba "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION", arXiv:1412.6980v9 [cs.LG] 30 Jan 2017