

Relay BP における中間出力モジュールの活用

日大生産工 ○外山 慶祐 日大生産工 山内 ゆかり

1. まえがき

ハードウェアの急速な発展に伴い、国内外ではビッグデータを用いた機械学習の活用が積極的に行われている。その機械学習の1種であるニューラルネットワーク (Neural Network: NN)[1]は、このハードウェアの発展の恩恵を特に大きく受けており、コンピュータの処理能力向上から短時間で高精度の出力を出すことができるようになり、利活用を盛んに行われている。また、NNの1種である畳み込みニューラルネットワーク (Convolutional Neural Network: CNN)[2]は、画像や音声等といった高次元データの入力でも短時間で高精度の出力を出すことができることから、近年大きな注目を集めている。CNNはネットワークの深さを深くするほど精度が向上するとされている[3,4]が、深くするに伴いパラメータサイズが増加しモデルが複雑になる。また、誤差逆伝播 (Back Propagation: BP)[5]学習時に、目標とする出力に関する情報量が層を遡るにつれて減少していくことから、発散や収束が遅く、過剰適合のリスクが生じるとされている。

この問題に対しLi Shenらは、学習時に関連性の高い有効な情報の伝播を促す新しい手法であるリレー誤差逆伝播法 (Relay Backpropagation: Relay BP)[6]を提案した。多層CNNにおいて標準的なBP時では、下位層 (入力層に近い層) のフィルターやバイアスは、効果的に学習されていなかったが、Li Shenらの手法ではネットワークモデルの中間で中間出力モジュールを導入することで、下位層のフィルターやバイアスが効果的に学習され出力精度の向上につながった。

しかしLi Shenら手法では、中間出力モジュールをネットワークモデルの学習時にのみ使用し、推定時は使用していない。そこで本研究では、中間出力モジュールを推定時にも使用し、出力の候補を増やすことで、計算コストはLi Shenらの手法と変わらず、出力の精度向上を図る。

2. 従来研究

2.1. 畳み込みニューラルネットワーク

畳み込みニューラルネットワーク (Convolutional Neural Network: CNN)[2]は主に画像や音声等といった高次元データの入力を元に、解析・予測を行うニューラルネットワーク

(Neural Network: NN)[1]の1種である。NNは入力が1次元に対し、CNNでは入力を高次元とすることができることから、各次元のデータだけではなく、各次元のつながりをも学習させることができ、高次元データの入力でも精度の高い出力を行うことができる。CNNは畳み込み層、プーリング層、パディング層、全結合層から構成される。Fig.1は畳み込み層、プーリング層、全結合層から構成されるCNNモデルの一例である。

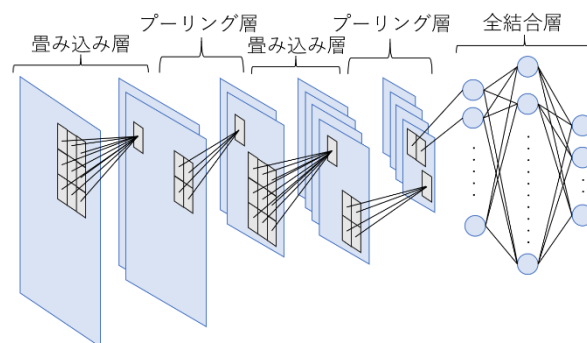


Fig.1 CNN モデルの一例

畳み込み層では、層への入力データとフィルターの積を求め、その値とバイアスの和を活性化関数に通すことで、出力データの中の1つの数値として出力する。この処理を入力データ全体に沿って計算することで、出力データ全体が生成される。式(1)は畳み込み層で行う計算を表したものである。入力データ u とフィルター h の積を求め、その合計値とバイアス b の和を Net' に代入している。式(2)では、式(1)で求めた Net' を活性化関数に通すことで、出力データ u' が出力している。

$$Net'_{(x,y)} = \left(\sum_i \sum_j u_{(x+i,y+j)} * h_{(i,j)} \right) + b \quad (1)$$

$$u'_{(x,y)} = f(Net'_{(x,y)}) \quad (2)$$

プーリング層では、層への入力データに対し、ある範囲内で規則に則って数値を計算し、それを出力データの1つの数値として出力する。プーリング層では、主に特定の範囲内で最大値を求めるマックスプーリングや、平均値を求めるアベレージプーリング等が採用される。式(3)は、マックスプーリングを式として表したものである。入力データ u に対し、 $l \times l$ の範囲に分割し、その範囲内の最大値が出力データ u' として出力されるということである。

$$u'(x,y) = \max(u_{(lx+s,ly+t)}) \quad (3)$$

where $s \in [0, l]$ $t \in [0, l]$

全結合層では、畳み込み層やプーリング層と異なり、入力データが高次元ではなく、1次元の形でニューロンに格納され、複数のニューロンが連なる列が複数並ぶ構成になっている。各ニューロンの列の間には重み W とバイアス b が存在し、この重み W とバイアス b がBP等の学習法で更新を繰り返すことで、学習が進み入力データの特徴を掴む。式(4)では、ニューロンの列に入力されたデータ $Input_i$ と重み W_{ji} の積の合計値にバイアス b_j を加えている。式(5)では、式(4)で計算された値 Net_j を活性化関数 f に通し、出力データとして $Output_j$ を出力している。

$$Net_j = \left(\sum_i Input_i * W_{ji} \right) + b_j \quad (4)$$

$$Output_j = f(Net_j) \quad (5)$$

Fig.1のCNNモデルには含まれていなかったが、他のCNNモデルによってはパディング層が含まれている場合も存在する。そこでパディング層についても説明していく。パディング層では、層への入力データに対し、その入力データの周りに固定の数値を埋める処理を行う。入力データの周りに0を格納するゼロパディングが主に用いられている。畳み込み層やプーリング層の処理を行うと、データサイズは小さくなってしまいが、パディング層を入れることでサイズを大きくすることができることから、データサイズを調整する際にパディング層が用いられる。Fig.2はパディング層で行われている処理を図示したものである。

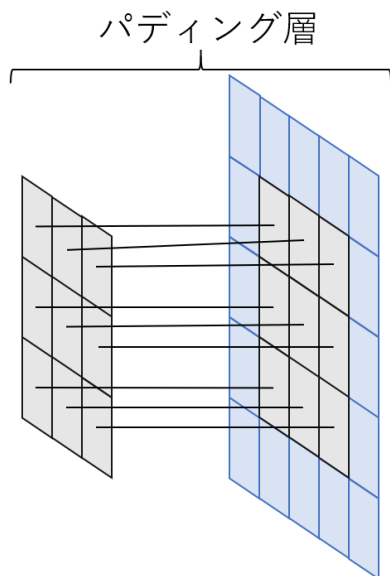


Fig.2 パディング層

2.2. 誤差逆伝播法

誤差逆伝播法(Back Propagation: BP)[5]は、NNやCNN等においてネットワークモデルを学

習させる際に用いられる学習アルゴリズムの1種である。Fig.3のNNモデルを元に具体的な式を用いて、BPについて説明していく。NNモデルにデータが入力されると必ず出力がされる。そのときの出力が、本来出力されるべき出力になるようにモデルを学習させるアルゴリズムがBPである。NNモデルによって出力される結果と、本来出力されるべき結果の誤差を元に、重み W やバイアス b の変化量を計算し、それを反映させることで出力に変化が生まれ、モデル全体が学習される。

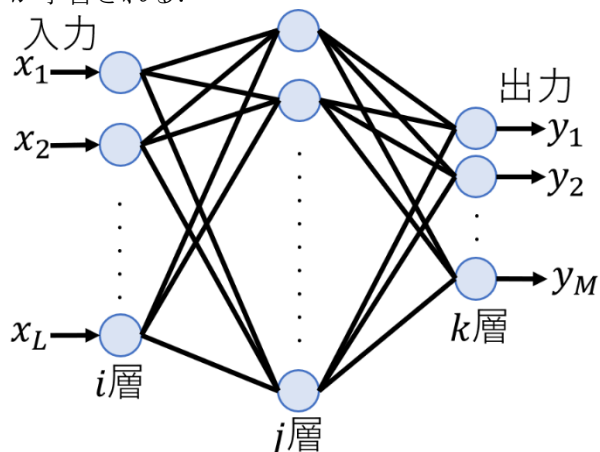


Fig.3 NNモデルの一例

式(6)は j 層と k 層の間に存在する重み W_{kj} の変化量 $\Delta \frac{\partial E}{\partial W_{kj}}$ を元の重み W_{kj} に反映させることで、適切な出力がされるように重み W_{kj} を変化させている式である。式(7)では、重み W_{ji} の更新を行っている式である。重み W_{ji} の更新の際も式(6)と同様に、計算で求めた変化量 $\Delta \frac{\partial E}{\partial W_{ji}}$ を重み W_{ji} に反映させることで学習させている。

$$W_{kj} \leftarrow W_{kj} + \Delta \frac{\partial E}{\partial W_{kj}} \quad (6)$$

$$W_{ji} \leftarrow W_{ji} + \Delta \frac{\partial E}{\partial W_{ji}} \quad (7)$$

上式の $\Delta \frac{\partial E}{\partial W_{kj}}$ や $\Delta \frac{\partial E}{\partial W_{ji}}$ は、以下の式(8)(9)で計算される。 α は学習率を表している。

$$\Delta \frac{\partial E}{\partial W_{kj}} = -\alpha * \frac{\partial E}{\partial W_{kj}} \quad (8)$$

$$\Delta \frac{\partial E}{\partial W_{ji}} = -\alpha * \frac{\partial E}{\partial W_{ji}} \quad (9)$$

式(8)(9)中の $\frac{\partial E}{\partial W_{kj}}$ や $\frac{\partial E}{\partial W_{ji}}$ を求めるにはNNの前向き演算時の計算式が必要となる。そこで、Fig.3のNNモデルの前向き演算時の計算式を式(10)~(14)として記した。

$$net_j = \sum_i x_i * W_{ji} \quad (10)$$

$$x_j = f(net_j) \quad (11)$$

$$net_k = \sum_j x_j * W_{kj} \quad (12)$$

$$x_k = f(net_k) \quad (13)$$

$$E = \frac{1}{2}(t - x_k)^2 \quad (14)$$

式(10)~(14)を用いて、 $\frac{\partial E}{\partial W_{kj}}$ や $\frac{\partial E}{\partial W_{ji}}$ を求めてい

く、 $\frac{\partial E}{\partial W_{kj}}$ や $\frac{\partial E}{\partial W_{ji}}$ は文字通り、誤差 E に対して重み

W_{kj} や W_{ji} で偏微分したものである。式(15)(16)

は、連鎖律を用いて $\frac{\partial E}{\partial W_{kj}}$ や $\frac{\partial E}{\partial W_{ji}}$ を求めている式で

ある。

$$\frac{\partial E}{\partial W_{kj}} = \frac{\partial E}{\partial x_k} * \frac{\partial x_k}{\partial net_k} * \frac{\partial net_k}{\partial W_{kj}} \quad (15)$$

$$\frac{\partial E}{\partial W_{ji}} = \frac{\partial E}{\partial x_k} * \frac{\partial x_k}{\partial net_k} * \frac{\partial net_k}{\partial x_j} * \frac{\partial x_j}{\partial net_j} * \frac{\partial net_j}{\partial W_{ji}} \quad (16)$$

式(15)(16)を見比べると、計算式の一部で同じ計算をしている箇所がある。その箇所をデルタ δ に置き換えることで同じ計算を複数回繰り返し計算しないで済む。ゆえに、無駄な計算を削減することができる。この規則をデルタ則と呼んでいる。NNやCNNの全結合層が多層化した場合は、このデルタ則を用いると計算量を削減できるだけでなく、複雑さを解消できることから広く用いられている。式(17)は、式(15)の一部をデルタ δ_k に置き換えた式であり、式(18)は、式(17)を用いて式(16)を書き換えた式である。

$$\delta_k = \frac{\partial E}{\partial x_k} * \frac{\partial x_k}{\partial net_k} \quad (17)$$

$$\frac{\partial E}{\partial W_{ji}} = \delta_k * \frac{\partial net_k}{\partial x_j} * \frac{\partial x_j}{\partial net_j} * \frac{\partial net_j}{\partial W_{ji}} \quad (18)$$

2.3. リレー誤差逆伝播法

リレー誤差逆伝播法 (Relay Backpropagation: Relay BP)[6]は、Li Shenらによって提案された目標とする出力に関する情報量が層を遡るにつれて減少していくというBPの問題を解消した新たなBPの1種である。Relay BPが標準的なBPとどのように異なるのかCNNモデルを用いて説明していく。まずCNNモデルについて、Fig.4で3層の畳み込み(conv)層と1層のプーリング(pooling)層をconvモジュールとし、3層の全結合(fc)層と1層の出力層をfcモジュールとする。Fig.4で説明されたモジュールを用いたRelay BPで学習するCNNモデルの一例がFig.5である。Fig.5の左側が前向き演算時の情報が流れる様子を表した図であり、右側はBP時の情報が流れる様子を表した図である。CNNモデルの中盤でfcモジュールを導入し、その箇所を中間出力モジュールとすると、学習段階でネットワークを介した有効な情報の伝

播を促し、下位層の重み W やフィルター F 、バイアス b にも有効な情報を流れることから、効率よく学習される。また、情報が下位層に伝播すればするほど有効な情報が少なくなってしまうことから、fcモジュールからやってきた情報が他のfcモジュールからやってきた情報と被った際は、より上位(出力層に近い層)のfcモジュールからやってきた情報は打ち切りにする。2つの情報は特定の割合でフィルター F やバイアス b の更新に用いられる。

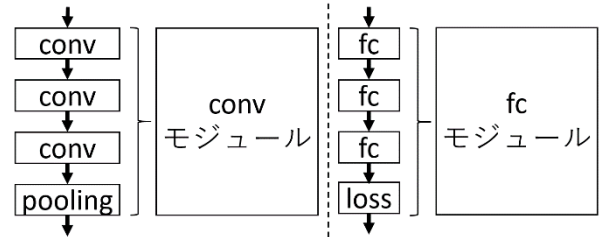


Fig.4 モジュール化

そしてFig.5のCNNモデルの例では、fcモジュールが3つあることから教師信号との誤差が3つ出力される。それら全体の誤差の最小化を目指すことで、モデル全体の学習を目指す。

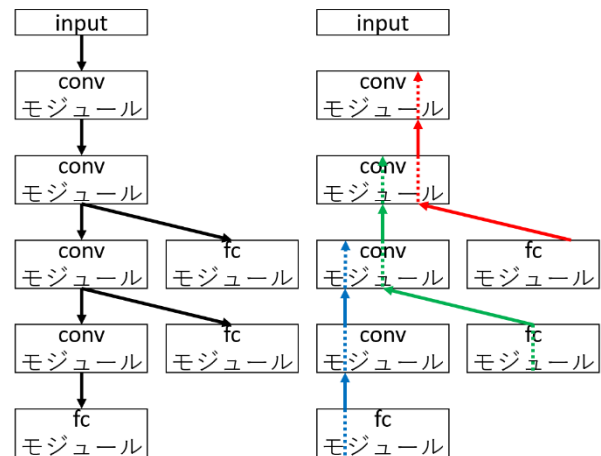


Fig.5 Relay BPを用いたCNNの一例

3. 提案手法

Relay BPでは中間出力モジュールを学習時のみに使用していたが、推定時にも使用することで、計算コストは変わらずさらなる精度向上を目指す新たなCNNモデルのアーキテクチャを提案する。Relay BPで学習するCNNモデルでは、出力は1つしかされていないが、中間出力モジュールを推定時にも使用することで、出力を複数個にすることができる。つまり、最上位層での出力を中間出力モジュールでの出力が手助けするかたちになるのである。Fig.6は新たに提案する構造をしたCNNモデルの一例である。複数の出力はそれぞれ特定の割合が割り振られており、その割合に応じてCNNモデル全体の出力が決定するかたちである。

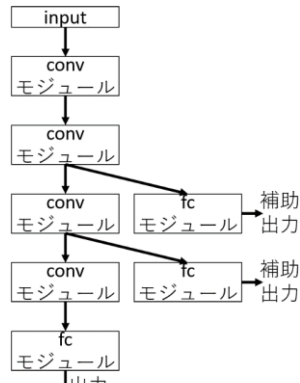


Fig. 6 新たに提案するCNNの一例

4. 実験および検討

提案手法の有効性を検証するため、学習時のみに中間出力モジュールを使用するCNNモデルと、学習時と推定時に中間出力モジュールを使用するCNNモデルで比較する。使用するデータセットはMNISTとし、60000枚の学習データで学習し、60000枚の訓練データと10000枚のテストデータのそれぞれでテストを行う。また、使用するCNNモデルはFig.7とFig.8で示したCNNモデルとする。

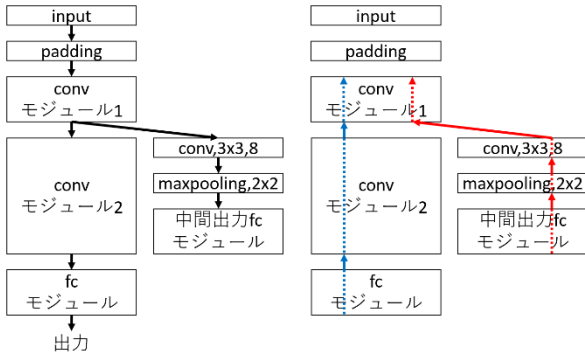


Fig.7 学習時のみに中間出力モジュールを使用するCNNモデル

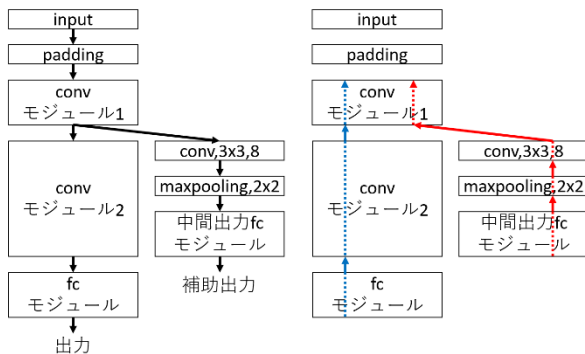


Fig.8 学習時推定時に中間出力モジュールを使用するCNNモデル

Fig.7とFig.8はCNNモデルの一部で畳み込み層やパディング層、プーリング層、全結合層をモジュール化している。Fig.9は、Fig.7とFig.8内に出てくるconvモジュール1とconvモジュール2の構成を表している。また、Fig.10ではfcモジュールと中間出力fcモジュール内の構成を表している。

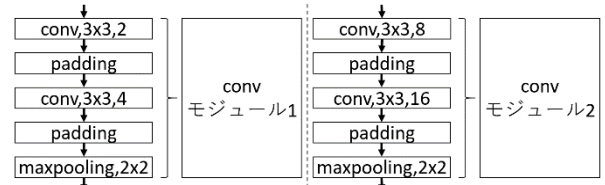


Fig.9 convモジュール1とconvモジュール2

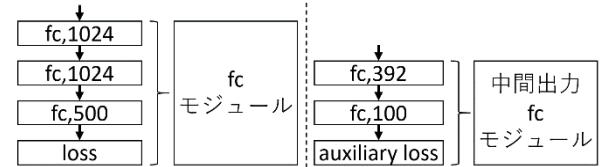


Fig.10 fcモジュールと

中間出力fcモジュール

それぞれのモデルで、訓練データとテストデータでテストした際の識別率を比較することで提案手法の有効性を検証する。

5. まとめ

今回、中間出力モジュールを学習時のみに使用しCNNモデルを学習させ、識別率向上を図るという従来手法に対し、推定時にも中間出力モジュールを活用することを新たなCNNモデルのアーキテクチャを提案し、さらなる識別率向上を試みた。

参考文献

- [1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", The bulletin of mathematical biophysics volume 5, (1943).
- [2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition.", Neural Computation 1, (1989).
- [3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks.", NIPS, (2012).
- [4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition.", ICLR, (2015).
- [5] David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams, "Learning representations by back-propagating errors", NATURE, (1986).
- [6] Li Shen, Zhouchen Lin and Qingming Huang, "Relay Backpropagation for Effective Learning of Deep Convolutional Neural Networks", European Conference on Computer Vision. Computer Vision – ECCV 2016, (2016).