CG-003

深層学習の逆伝播における全結合層ニューロンの準同期式更新
Semi-Synchronous updating of neurons in full-connection layers through backpropagation of deep learning

堀哲也 † 関谷侑希 † 竹中要一 †‡
Tetsuya Hori Yuki Sekiya Yoichi Takenaka

1. はじめに

人工知能を実現する機械学習手法の 1 つとして，Hinton らが実現を可能とした深層学習が注目を集めている．これは，大脳皮質の神経回路を模した多層の人工ニューラルネットワークである．その応用例は多岐にわたり，画像認識 [1] や自然言語処理 [2]，音声処理 [3] などが存在する．

画像認識の例としては，物体検出が挙げられる．物体検出とは，いくつかの物体を含む画像に対し，物体の種類とその位置を推定するタスクである [4]．この技術は監視カメラや，特にリアルタイム処理を実現する事により自動運転が可能となるなど幅広い応用が存在する．リアルタイム処理を実現する手法としては，Girshick が提案した Fast R-CNN[5] や，Redmon らによる YOLO(You Only Look Once)[6] が存在する．また，Goodfellow らの提案した GAN(Generative Adversarial Networks)[7] と，Radford らが提案した DCGAN(Deep Convolutional Generative Adversarial Networks)[8] では，訓練セットに存在しない画像を生成することができる．これについては，Ledig らによる超解像への応用 [9] や，Zhu らによる，画風の変換 [10] がある．これらの技術や応用手法により深層学習は画像認識に不可欠な存在となっている．

一方で，深層学習のより基本的な部分に対する研究も存在する．深層学習の学習は，順伝播，逆伝播と重み更新の 3 ステップを繰り返して行われる．訓練セットの全てを使って学習するバッチ学習に対して，訓練セットの一部を無作為に選んで学習するミニバッチ学習がある．Ioffe と Szegedy による Batch Normalization では，層への入力データの分布を揃えることで，重みの初期値に頑健になり，過学習が抑制される [11]．Srivastava らの Dropout は，ニューロンを確率的に順伝播の計算式から除去することで，過学習を抑制する手法である [12]．重み更新におけるモメンタムは，慣性の項を加えることで，学習の進行を速くすることができる．

本研究では，重み更新ステップにおいて，全結合層で重みが更新されるニューロン数を $\sqrt{N}$ に制限する方法を提案する．ここで N は全結合層に含まれるニューロンの数とする．重み更新が実施されるニューロン数を制限する本提案手法は，ニューロン重みの過度な学習を緩和し，最終的な正答率を向上することが期待される．重み更新を実施するニューロン数を制限する方法は，ホップフィールドネットワークモデルに対する適用例があり [13]，準同期式更新と命名されている．しかしながら深層学習で用いられた例は存在しない．そこで我々は，深層学習の全結合層に準同期式更新を適用する方法を提案する．

2. 深層学習

2.1. アーキテクチャ

深層学習モデルは，いくつかの層が連なった形で構成される．層は，いくつかのニューロンから成る．データを入力されたとき，それを変換して出力する機能を持つ．全結合層は，重みを持ち，入力と重みの行列積を計算して，非線形変換したものを出力する．非線形変換には，標準シグモイド関数や，$\max(x, 0)$ が用いられる．ただし，最終層では，タスクに合わせた特殊な変換を行う．分類問題では，層の出力の合計が 1 になるように変換を行う．これにより，出力を確率として解釈できるようになる．畳み込み層もまた，重みを持ち，重みのウィンドウをスライドさせながら行列積を計算し，非線形変換する．プーリング層は，重みを持たない層である．平均値または最大値を取りながら，入力をダウンサンプリングして出力する．

深層学習の有効性を証明したモデルが，Krizhevsky らが考案した AlexNet である [1]．これは，畳み込み層とプーリング層をいくつか連ね，最後に全結合層を 3 つ加えた構造である．AlexNet より性能が高く，かつシンプルな構造のものとして VGG-16 と VGG-19 が挙げられる [14]．これは，2 から 4 個の畳み込み層と 1 つのプーリング層のセットをいくつか連ねたものである．VGG-16 は 16 個，VGG-19 は 19 個の，重みのある層を持つ．さらに良いパフォーマンスを発揮するモデルとして，ResNet が存在する [15]．ResNet の特徴は，連続する 2 つの畳み込み層に対し，それらを迂回するショートカット構造である．ショートカット構造を含むことにより，層を多くしても正しく学習できるようになった．ResNet は VGG よりさらに性能が高く，152 層まで層を連ねても性能が向上し続けた．

2.2. 学習過程

深層学習の学習は順伝播，逆伝播と重み更新の 3 ステップの繰り返しである．各ステップの説明を行う．

2.2.1. 順伝播

順伝播では，まず，データが最初の層に入力される．最初の層の出力は，2 番目の層の入力となる．そして，2 番目の層の出力は 3 番目の層の入力になる．このようにして，最後の層の出力まで繰り返し計算する．

†関西大学, Kansai University
‡大阪大学, Osaka University

2.2.2. 逆伝播

順伝播の結果と教師データの差を求め，損失とする．全ての層の重みについて，勾配を計算する．勾配は，損失に対する偏微分で求められる．ただし，ある層の重みの勾配を求めるには，その後ろの層の入力に対する勾配を求める必要がある．そこで，誤差逆伝播を用いる．これは，動的計画法の 1 種で，順伝播とは逆に，最後の層から順に勾配を求める方法である．

2.2.3. 重み更新

全ての重みに対する勾配を求めた後，勾配を用いて重みを更新する．勾配 $\frac{\partial L}{\partial W}$ に対し，学習率 η を用いて，重み W を次のように変更する．

$$W \leftarrow W - \eta \frac{\partial L}{\partial W} \tag{1}$$

モメンタムを用いる場合，前回の更新量 ΔW，慣性パラメータ α で，次のように更新する．

$$W \leftarrow W - \eta \frac{\partial L}{\partial W} + \alpha \Delta W \tag{2}$$

2.3. 学習におけるテクニック

2.3.1. ミニバッチ学習

全訓練セットを一度に全て用いる学習方法をバッチ学習と呼ぶ．逆に，訓練セットをランダムにサブセット（以下，ミニバッチと呼ぶ）へ分割し，それを用いて学習する方法をミニバッチ学習と呼ぶ． ミニバッチ学習は，毎回異なるミニバッチに対して最適化を行うため，局所解に陥りにくくなるという利点を持つ．

ミニバッチ学習の手順を示す．事前に，ミニバッチの大きさ N を決めておく．学習を開始すると，訓練セットを要素数 N のミニバッチへ分割する．ミニバッチを順に，重複なく用いながら，順伝播，逆伝播，重み更新の 3 ステップを繰り返す．全てのミニバッチを用いたら，再び訓練セットをランダムにミニバッチへ分割する．そのようにして学習を繰り返す．

2.3.2.Dropout

深層学習のような，多量の重みを持つモデルでは，過学習が問題になる場合がある．これは，訓練セットに過剰に適合した結果，一般のデータに対して正しく推論できない状態である．そこで，層内のニューロンを部分的に無効化し，過学習に陥りにくくする手法として，Dropout が提案された．

Dropout では，パラメータとして，有効率 p $(0 < p < 1)$ を設定する．m 個のニューロンからなる層に Dropout を適用し，n 個のデータが入力されたとする．順伝播で，層が積和計算と非線形変換した結果を Y とする．Y は $n \times m$ 型の行列になる．同じ形の行列 $mask$ を，確率 p のベルヌーイ分布に従う確率変数で作成する．層は，式 4 の $\tilde{Y}$ に示す通り，Y と $mask$ の要素ごとの積を出力する．

$$mask_{n \times m} \sim \text{Bernoulli}(p) \tag{3}$$

$$\tilde{Y} = Y \circ mask \tag{4}$$

Dropout を用いると，層の重みは，ニューロン数 $\times$ p 個のニューロンを用いて正しい出力を得るよう学習される．一方，推論時には Dropout を用いずに，全てのニューロンを用いて計算を行う必要がある．そのため，推論する場合，層は計算を行ったあと，結果を p 倍したものを出力する．

2.3.3.Batch Normalization

深層学習には，内部共変量シフトという問題がある．これは，ある層の入力の分布と，非線形変換する際のデータの分布が変わってしまう問題である．層が多くなるほど，分布の乖離が大きくなり，正しく学習することが難しくなる．この問題に対処するために，Batch Normalization が考案された．

Batch Normalization では，各層に，パラメータ $\gamma = 1, \beta = 0$ を用意する．順伝播では，積和計算後の値について，その平均と分散を用いて正規化を行う．正規化したデータ $\hat{u}$ を，$\gamma\hat{u} + \beta$ としてから非線形変換を行い，層の出力とする．逆伝播では，層の重みだけでなく，γ, β の勾配を求める．そして，重み更新のタイミングで，γ, β も更新する．

3. 提案手法

本研究では重み更新ステップにおいて，全結合層で重みが更新されるニューロン数を $\sqrt{N}$ に制限する方法を提案する．ニューラルネットワークモデルのある全結合層（ニューロン数 N）に対する準同期式更新は次の通りである．

N 個のニューロンを $\sqrt{N}$ 個のグループ $G = \{g_1, \ldots g_{\sqrt{N}}\}$ に分割する．ここで各グループ g_i は $\sqrt{N}$ 個のニューロンから構成されるものとし，複数のグループに属するニューロンは存在しない，すなわち $g_i \cap g_j = \phi$ とする．

準同期式更新とは，ニューロンの重みを更新するタイミングにおいて，一つのグループ g_i に属するニューロンの重みだけが更新されるものと定義する．ただし，グループは g_1 から $g_{\sqrt{N}}$ まで順に選択される．また $g_{\sqrt{N}}$ の次に選ばれるグループは g_1 とする．ここで，ニューロンの重みを更新するタイミングとは，バッチ学習の重み更新や，ミニバッチ学習における 1 ミニバッチに対する重み更新のタイミングとする．

モメンタムへの対応法

深層学習を効率的に行う一手法として 2.2.3 節で述べたモメンタムがある．準同期式はモメンタムとの併用

が可能である．併用する場合，慣性項に基づく重み更新を全ニューロンに対して実行する．

更新対象となるニューロンの重み更新式

$$W \leftarrow W - \eta \frac{\partial L}{\partial W} + \alpha \Delta W \tag{5}$$

更新非対象となるニューロンの重み更新式

$$W \leftarrow W + \alpha \Delta W \tag{6}$$

複数の全結合層に対する適用法

提案する準同期式更新は，ある全結合層に対して定義される．そのため，深層学習に存在する複数の全結合層にそれぞれ独立して適用することが可能である．

Dropout との相違点と併用

提案手法と混同しやすい類似手法として Dropout が挙げられる．Dropout と提案手法は，更新するニューロン数を制限するという点が同じである．Dropout は制限されたニューロンだけで順伝播，逆伝播を行う．一方，準同期式更新は，重み更新のみを制限する．すなわち，順伝播および逆伝播は必ず全てのニューロンで行い，重み更新は一部のニューロンだけで行う．

提案手法は Dropout と併用可能である．Dropout は一定の確率でニューロンが存在しないものと見做す手法である．そのため準同期式更新と併用が可能である．

1. Dropout の適用に先立ちニューロングループ G への分割を行う
2. Dropout するニューロンの選定を G とは独立して，全ニューロンに対して行う

4. 実験

4.1. 既存モデルへの導入

4.1.1. 実験方法

準同期式更新の有効性を確かめるため，画像分類問題による実験を行う．データセットは，CIFAR10[16] を用いる． CIFAR10 は，カラー画像のデータセットで，解像度は縦横共に 32 である．10 種類のクラスが存在し，訓練セット 5 万枚（5000 枚/クラス）とテストセット 1 万枚により構成される．モデルは，VGG-16 を用いる．その構造を図 1 に示す．これは， 224×224 の解像度の画像を，1000 種類に分類することを想定して作られたモデルである．この実験では 10 クラス分類問題を行うため，最終層のニューロンは 10 個に変更する．

重み更新における学習率 η は 0.001，モメンタムの慣性 α は 0.9 に設定する．また，ミニバッチの大きさを 32 として，ミニバッチ学習を行う．データは，バイキュービック法を用いて 224×224 に拡大し，平均・標準偏差で標準化する．学習は 100 エポックまで行う．ただし，本実験におけるエポックとは，分割されたミニバッチ全てを使い切った時点を指す．

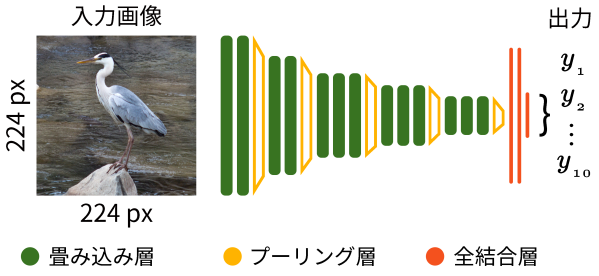

図 1: VGG-16 の構造

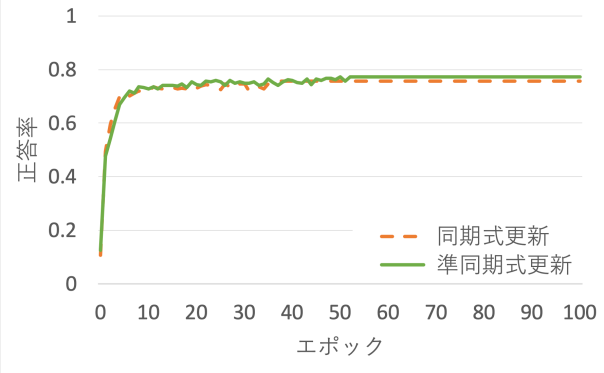

図 2: 正答率の推移

準同期式更新は，最終層以外の全結合層に対して適用する．この 2 つの層は，いずれも 4096 個のニューロンをもつ．よって，これらのニューロンは，64 個のグループに分割される．

評価指標には正答率を用いる．実験開始前および，1 エポックごとに，テストセットに対する正答率を求める．学習後，正答率の推移を比較する．

4.1.2. 結果

正答率の推移を図 2 に示す．学習開始から 10 エポック程度までは，同期式更新がより高い正答率である．しかし，それ以後は，準同期式更新で学習した場合がより高い正答率を記録している．

4.2. 既存類似技法との比較

提案手法と類似した手法としては Dropout(2.3.2 節), Batch Normalization (2.3.3 節) が挙げられる．本節では前節で用いた VGG-16 に類似手法を導入し，提案手法と正答率の比較を行う．

4.2.1. 実験条件

本実験では 6 種類のニューラルネットワークモデルの比較を行う．4.1.1 節で用いた，VGG-16 の全結合層

を同期式更新したモデル及び，準同期式更新したモデル．そして，VGG-16 に類似手法を導入した 4 種類のニューラルネットワークモデルである．

1. Dropout を全結合層に導入したモデル
2. Batch Normalization を全結合層に導入したモデル
3. Batch Normalization を畳み込み層に導入したモデル
4. Batch Normalization を全結合層と畳み込み層に導入したモデル

正答率を比較するエポックは，4.1.1 節の図 2 において正答率が概ね収束した 50 エポックとし，それ以外の実験条件は 4.1.1 節と同じとした．

4.2.2. 結果

4 種類の比較対象の結果を表 1 に示す．提案する準同期式更新手法の結果を再掲している．表中の BN は，Batch Normalization を用いたことを表す．

表より Batch Normalization を畳み込み層に導入したモデルの正答率が最も高いことがわかる．一方，提案する準同期式更新は最も正答率が悪い結果となった．

4.3. 既存類似技法との組合せ

準同期式更新は，Dropout や Batch Normalization と組みわせて利用可能である事が挙げられる．そこで，Dropout や Batch Normalization と組み合わせて利用した場合，準同期式更新が性能に与える影響を検証する実験を行う．

4.3.1. 実験条件

表 1 で最も正解率が高かった畳み込み層に Batch Normalization を適用したモデルに対し，準同期式更新，Dropout を独立して導入する．準同期式更新の有無，Dropout の有無の合計 4 種類に対して 4.2.1 節と同じ条件で実験を 10 回行った．

表 1: 提案手法と 4 つのモデルの正答率

	正答率
同期式更新	0.757
準同期式更新	0.772
1) Dropout	0.778
2) BN (全結合層)	0.822
3) BN (畳み込み層)	0.866
4) BN (全結合層と畳み込み層)	0.819

表 2: 既存類似技法と組み合せた場合の正答率

	Dropout	
	なし	あり
同期式更新	0.8616 ± 0.0029	0.8645 ± 0.0086
準同期式更新	0.8865 ± 0.0023	0.8483 ± 0.0127

4.3.2. 結果

表 2 に 4 種類のニューラルネットワークモデルの正解率及び，その右に 10 回試行時の標本標準偏差を ± で記す．

正解率が最も高かったのは準同期式更新のみを導入したモデルであり，正解率が最も低かったのは，準同期式更新と Dropout の両者を導入したモデルである．Dropout と準同期式更新のいずれかを導入する事によって正解率は向上するが，両者の導入により正解率が低下している．

4.4. 考察

3 種類の実験より明らかとなった準同期式更新の性質について考察する．

4.1.1 節の実験により，準同期式更新で重み更新を行う事によって学習速度が低下するものの，十分なエポック数を経た後の最終正解率が向上する事がわかった．準同期式更新では，1 つのミニバッチで重みが更新されるニューロン数が少ないため，学習速度が低下するものと考えられる．一方，最終正解率が向上するのは，全ニューロンの同時更新が解空間において過度の局所最適化に陥っている可能性が示唆される．この検証には，学習速度が同程度になるように学習率を調節する事で調べられると考えている．

4.2.1 節の実験により類似手法である Dropout 及び Batch Normalization との性能比較を行った．その結果，準同期式更新単体では，類似手法よりも正解率向上能力が低いことが示された．ただし，準同期式更新は，Dropout や Batch Normalization との併用が可能である．4.3.1 節において両類似手法との併用の有効性を検証した．その結果，正解率向上に最も良い組み合わせが Batch Normalization と準同期式更新の併用であることが判明した．また，準同期式更新と Dropout は，それぞれ単独利用は効果的であるものの，併用によって正解率が低下することが明らかとなった．以上より準同期式更新は Dropout を代替することでニューラルネットワークの性能向上に資する有用な手法であると考えている．

まとめ

本研究では，深層学習の重み更新に注目した．全ての重みを更新する従来手法に対し，一部の重みだけを更新する準同期式更新を定義した．そして，ベーシックなモデルへ導入し，準同期式更新が，従来の更新手法よ

り優れていることを確かめた．また，Dropout，Batch Normalization と比較して，準同期式更新自体は，劣っていることがわかった．一方，畳み込み層に Batch Normalization を適用した状態で，Dropout の代わりに準同期式更新を併用することで，さらに性能が向上することが明らかになった．これらの結果より，提案手法の有用性が認められた．

今後の課題は 2 つ挙げられる．1 つは，準同期式更新が性能向上に寄与する原因を追求することである．4.4 節で述べた，従来手法が局所最適化に陥っている可能性について検証することがこれにあたる．もう 1 つは，他のモデルへの導入を検討することである．他のモデルでも準同期式更新の有用性が明らかになれば，より多くの場面で性能向上が期待できるからである．

参考文献

[1] Krizhevsky, Alex, Ilya Sutskever, Geoffrey E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems 25, 編集者：F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, 10971105, Curran Associates, Inc. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf, (2012).

[2] Sutskever, Ilya, Oriol Vinyals, Quoc V, Le, "Sequence to Sequence Learning with Neural Networks", arXiv:1409.3215 [cs], http://arxiv.org/abs/1409.3215, (2014).

[3] Dai, Wei, Chia Dai, Shuhui Qu, Juncheng Li, Samarjit Das, "Very Deep Convolutional Neural Networks for Raw Waveforms", arXiv:1610.00087 [cs], http://arxiv.org/abs/1610.00087, (2016).

[4] Liu, Li, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, Matti Pietikinen, "Deep Learning for Generic Object Detection: A Survey", arXiv:1809.02165 [cs], http://arxiv.org/abs/1809.02165, (2019).

[5] Girshick, Ross, "Fast R-CNN", arXiv:1504.08083 [cs], http://arxiv.org/abs/1504.08083, (2015).

[6] Redmon, Joseph, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", arXiv:1506.02640 [cs], 5 月. http://arxiv.org/abs/1506.02640, (2016).

[7] Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Networks", arXiv:1406.2661 [cs, stat], http://arxiv.org/abs/1406.2661, (2014).

[8] Radford, Alec, Luke Metz, Soumith Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", arXiv:1511.06434 [cs], http://arxiv.org/abs/1511.06434, (2016).

[9] Ledig, Christian, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, et al. , "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", arXiv:1609.04802 [cs, stat], http://arxiv.org/abs/1609.04802, (2017).

[10] Zhu, Jun-Yan, Taesung Park, Phillip Isola, Alexei A, Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks", 2017 IEEE International Conference on Computer Vision (ICCV), 224251, Venice: IEEE. https://doi.org/10.1109/ICCV.2017.244, (2017).

[11] Ioffe, Sergey, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", arXiv:1502.03167 [cs], http://arxiv.org/abs/1502.03167, (2015).

[12] Srivastava, Nitish, Georey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overtting", Journal of Machine Learning Research 15 (2014) 1929-1958, 30, (2014).

[13] 竹中要一, "組合せ最適化問題に対するニューラルネットワーク解法のニューロンフィルタに関する研究", https://doi.org/info:doi/10.11501/3169484, (2000).

[14] Simonyan, Karen, Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv:1409.1556 [cs], http://arxiv.org/abs/1409.1556, (2015).

[15] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", arXiv:1512.03385 [cs], http://arxiv.org/abs/1512.03385, (2015).

[16] "Learning Multiple Layers of Features from Tiny Images", Alex Krizhevsky, (2009).