Train longer, generalize better: ニューラルネットワークの大規模なバッチトレーニングにおける汎化ギャップの解消

Elad Hoffer, Itay Hubara, Daniel Soudry テクニオン-イスラエル工科大学（ハイファ、イスラエル

{elad.hoffer, itayhubara, daniel.soudry}@gmail.com

# アブストラクト

背景深層学習モデルは、通常、確率的勾配降下法またはその亜種のいずれかを用いて学習されます。これらの手法では、学習データのごく一部から推定した勾配を用いて重みを更新します。しかし、大きなバッチサイズを使用すると、「汎化ギャップ」現象として知られる汎化性能の低下が持続することが観察されています。このギャップの原因を明らかにし、それを解消することは未解決の課題でした。

貢献しています。学習率の高い初期のトレーニング段階について検討した。その結果、初期設定からの重みの距離は、重みの更新回数に応じて対数的に成長することがわかった。そこで、同様の「超低速」な拡散挙動を示すことが知られている「ランダムな風景上のランダムウォーク」という統計モデルを提案する。この仮説に基づいて実験を行い、「汎化ギャップ」はバッチサイズではなく更新回数が比較的少ないことに起因しており、使用する学習レジームを適応することで完全に排除できることを実証的に示した。さらに、大規模なバッチ体制でモデルを学習するためのさまざまな手法を検討し、「Ghost Batch Normalization」という新しいアルゴリズムを発表した。このアルゴリズムは、更新回数を増やすことなく、一般化ギャップを大幅に減少させることができる。さらに、我々の発見を検証するために、MNIST、CIFAR-10、CIFAR-100、ImageNet を用いた実験を行った。最後に、深層モデルの学習に関する一般的な慣習や考え方を再評価し、良好な汎化を実現するためには最適ではない可能性を示唆している。

# はじめに

ここ数年、ディープニューラルネットワーク（DNN）は、画像からの物体認識（He et al., 2016）、音声認識（Amodei et al., 2015）、自然言語処理（Luong et al., 2015）、強化学習を用いたコンピュータゲームの制御（Silver et al., 2016; Mnih et al., 2015）など、多くの応用領域で大幅な改善を持続的に可能にしてきました。

非常に複雑で非凸の DNN を学習するために選択される最適化手法は、通常、stochastic gradient decent (SGD)またはその変形である。SGD は、せいぜい非凸の目的関数の局所的な最小値を求めるだけなので、DNN の画期的な結果を説明するために、かなりの研究努力が払われています。鞍点は回避できること（Ge et al., 2015）、学習誤差の「悪い」局所最小は指数関数的に消滅すること（Dauphin et al., 2014; Choromanska et al., 2015; Soudry & Hoffer, 2017）が主張されています。しかし、これらの複雑なモデルが、大きくオーバーパラメータ化されているにもかかわらず、見たことのないデータにうまく一般化する傾向がある理由はまだ不明である（Zhang et al.、2017）。

汎化の特定の側面は、最近多くの関心を集めています。Keskar ら（2017）は、長い間観

察されてきた現象（LeCun ら、1998a）に注目しました-大きなバッチサイズが使用される一方で

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

∗

均等な貢献

(

a)

トレーニングエラー

(

b)

バリデーションエラー

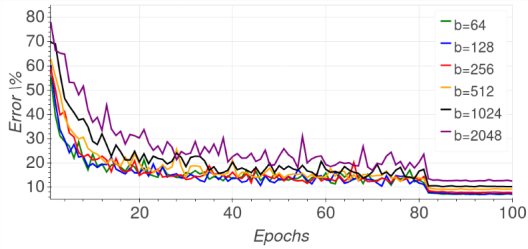
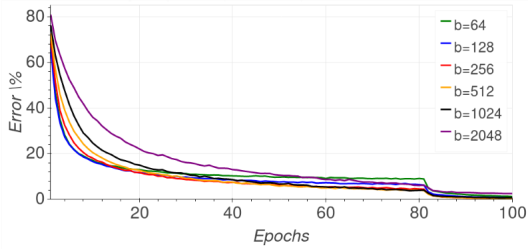


図 1：バッチサイズの分類誤差への影響

のトレーニング DNN では、トレーニングされたモデルの一般化がうまくいかないようです。これは、「予算や制限なしに、損失関数が改善しなくなるまで」モデルを学習しても変わらなかった（Keskar ら、2017）。この性能の低下は、「汎化ギャップ」と名付けられている。

汎化ギャップの原因を理解し、さらにそれを減少させる方法を見つけることは、実用上重要な意味を持つと考えられます。大きなバッチサイズでの学習は、すぐに並列化が進むため、学習時間を短縮できる可能性があります。Deep Learning のための SGD の並列化には多くの努力が払われてきたが（Dean et al., 2012; Das et al., 2016; Zhang et al., 2015）、それでも高速化とスケールアウトはバッチサイズによって制限されている。この研究では、この問題に取り組むための最初の試みを提案します。

最初に

* 我々は、初期の学習段階は、経験的に観察されるように、初期化からの重みの距離が「超低速」で対数的に増加する、高次元の「ランダムなポテンシャル上のランダムウォーク」プロセスを用いて記述できることを提案する。

この仮説に触発されて、次のことがわかります。

* 学習率とバッチの正規化を調整するだけで、一般化ギャップを大幅に減少させることができます（例えば、5％から 1～2％に）。
* 一般的な慣行（Montavon et al., 2012）や理論的な推奨（Hardt et al., 2016）とは対照的に、学習誤差や検証誤差に観察可能な変化がなくても、初期の高い学習率で長時間にわたって汎化が改善し続けます。ただし、この改善は、重みの初期化からの距離に関連しているようです。
* 固有の「汎化ギャップ」はありません。大規模なバッチトレーニングは、反復回数を調整することで、小バッチトレーニングと同様に汎化することができます。

# 大規模なバッチでのトレーニング

トレーニング方法。ディープニューラルネットワークを学習する一般的な方法は、固定の学習率とモメンタム項を持つ勾配ステップを用いて目的を最小化する最適化「レジーム」に従うことです（Sutskever et al.学習率は時間とともにアニールされ、通常は学習データの数エポックごとに指数関数的に減少していきます。このレジームに代わるものとして、Adam (Kingma & Ba, 2014)、Rmsprop (Dauphin et al.)、Adagrad (Duchi et al., 2011)などの適応的なパラメータごとの学習方法を使用することができる。これらの手法は、SGD ベースの最適化の収束率を高めることが知られています。しかし、現在の多くの研究では、最適化プロセスの全部または一部に、SGD の単純な変種（Ruder, 2016）を依然として使用しています（Wu et al.

そこで、He ら（2016）が採用したレジームと同様に、数回のエポックごとに指数関数的に減少する固定の学習率を持つ、モーメンタム SGD に着目しました。SGD の収束は、バッチサイズに影響されることも知られていますが（Li et al.、2014）、本研究では一般化に焦点を当てます。我々の結果のほとんどは、He ら（2016）によって導入された Resnet44 トポロジーで実施されました。セクション 6 では、追加の実証結果で我々の発見を強化します。

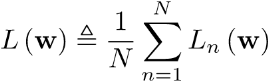
前作の経験的観察 Keskar ら（2017）による前作では、比較的大きなバッチで学習したモデルの性能と特性を調べ、以下のような観察結果を報告しています。

* バッチサイズが大きいモデルを学習すると、一般化誤差が大きくなります（図 1 参照）。
* この「汎化のギャップ」は、モデルを無制限に学習させても、損失関数が改善されなくなるまで残っているようでした。
* 汎化度が低い場合は「シャープな」ミニマム1（強い正の曲率）と相関があり、汎化度が高い場合は「フラットな」ミニマム（弱い正の曲率）と相関がありました。
* スモールバッチ体制では、ラージバッチ体制で作られたウェイトと比較して、初期点から遠く離れたウェイトが作られることが簡単に指摘された。

彼らの仮説は、小さなミニバッチにおける大きな推定ノイズ（フルバッチではなくミニバッチを使用したことに起因する）が、重みを鋭いミニマムの引力の流域から抜け出させ、より良い一般化をもたらす平坦なミニマムへと向かわせるというものでした。

# 理論的分析

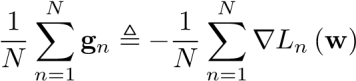
表記します。 本論文では、Stochastic Gradient Descent (SGD) ベースのディープニューラルネットワーク (DNN) の学習について検討する。DNN は、*N* 個のサンプルからなる有限のトレーニングセットで学習される。ニューラルネットワークのパラメータのベクトルを**w**、サンプル *n* に対する損失関数を *Ln* (**w**)と定義し、学習損失を最小化することで**w** を求める。

 *,*

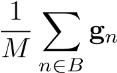
1後に指摘されたのは（Dinh et al., 2017）、損失に影響を与えずにパラメータを変更できるある種の「縮退」方向は、この説明から除外しなければならないということです。例えば、任意の *c>*

*0* と任意のニューロンに対して、すべての入力重みを c で乗算し、出力重みを c で除算することができます。これは損失には影響しませんが、任意に強い正の曲率を生成することができます。

を SGD を用いて行うことができます。*L*(**w**)を最小化するには、負の損失の勾配を推定する必要があります。

**g**, 

ここで**g**は真の勾配であり、*gn* はサンプルごとの勾配である。学習時には，あるミニバッチ *B*（無作為に選択された *M* 個のサンプルインデックスのセット）で計算された平均

勾配**g**ˆのみを用いて，パラメータベクトル**w**を増加させる。 **g**ˆ *, .*

この場合、更新ステップ *t* における重みは、ミニバッチ勾配 Δ*wt* = *η***g**ˆ*t* に従って増加します。SGD のランダム性に関しては

**Eg**ˆ*t*＝**g**＝-∇*L*（*wt*）。

であり，増分は異なるミニバッチ間で無相関である2．物理的に直感的に理解するためには、重みベクトル *wt* を、損失（「ポテンシャル」）ランドスケープ *L*(*wt*)上をランダムウォークする粒子と考えることができます。したがって、例えば、増分に運動量項を加えることは、粒子に慣性を加えることに似ています。

モチベーション損失の正確な形状がわからない複雑なシステム（DNN など）では、統計物理学モデルは一般的に、ランダムプロセスとしてのポテンシャルのより単純な記述を仮定しています。例えば、Dauphin ら(2014)は、DNN の損失面である *L*(**w**)と、Bray &

Dean(2007)で分析された高次元のガウスランダムフィールドの間のアナロジーを用いて、局所的な最小値が低誤差になる傾向があるという観測結果を説明していますが、このガウスランダムフィールドはゼロ平均と自己共分散を持つ

 (1) ここで、期待値は損失のランダム性に対するものである。このアナロジーは、DNN において、損失が大きいローカルミニマムは、Bray & Dean (2007)のように、実際に指数関数的に消滅するという仮説をもたらしました。最近になって、現実的なニューラルネット

ワークモデルでも同様の結果が証明され始めている（Soudry & Hoffer, 2017）。したがって、損失の同様の統計的モデルもまた、我々の経験的な観察に対して有用な洞察を与えるかもしれない。

モデルランダムポテンシャル上のランダムウォーク幸いなことに、粒子が「ランダムなポテンシャル上をランダムに歩く」という高次元のケースは、すでに数十年前に広く調査されています(Bouchaud & Georges, 1990)。その調査の主な結果は、ランダムポテンシャルの自己共分散の漸近的な振る舞いが [[1]](#footnote-1),

E(*L*(**w**1)*L*(**w**2)) ∼ **kw**1 - **w**2k*α , α > 0* (2) がある範囲にある場合、その範囲におけるランダムウォーカーの漸近的な振る舞いを決定します。

 *.* (3) これは "超低速拡散 "と呼ばれるもので、典型的には *kwt* - **w**0k ∼ (log√*t*)2*/α* であり、これに対して

2 正確に（置き換えあり）、またはおおよそ（置き換えなし）のいずれか：付録セクション A を参照。

このような振る舞いをする（任意の *α > 0* の場合）理由は、粒子が距離 d を移動するためには、高さ∼dα*/2* のポテンシャル障壁を通過しなければならないからです。(2).そして、それぞれの障壁を登る（または迂回する）ためには、障壁の高さに対して指数関数的に長い時間がかかります：*t* ∼ exp(*dα*/2)。この関係を逆にすると、*d* ∼ (log(*t*))2*/α* という式が得られます。高次元の場合、このような挙動は Marinari ら(1983)によって初めて数値的に示され、ヒューリスティックに説明された。その後、Durrett(1986)によって離散的な格子の場合に厳密に証明され、Bouchaud & Comtet(1987)によって連続的な場合に説明された。

3.1 実証実験結果との比較とその意味

この超低速拡散の予測を検証し、*α* の値を見つけるために、図 2a では、図 1 に示した実験の上で、初期の学習段階での *kwt* - **w**0k を調べています。その結果、初期化点からの重みの距離は、学習の反復回数（重みの更新回数）に応じて対数的に増加することがわかり、これは *α*＝2 のモデルと一致します。

*kwt* - **w**0k ∼ *logt.* (4) 興味深いことに、*α*=2 の値は、付録セクション B で推定された損失の統計と一致しています。

さらに、図 2a では、すべてのバッチサイズで非常に似た対数グラフが観測されていることがわかります。

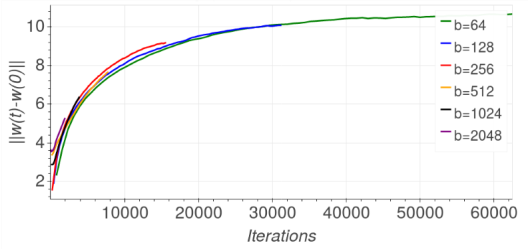
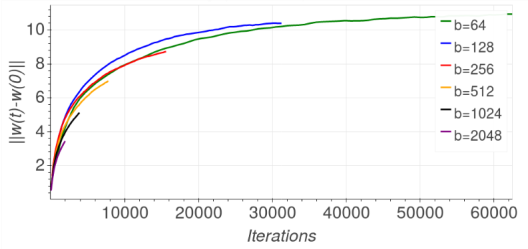
しかし、2 つの大きな違いがあります。まず、それぞれのグラフの傾きが多少異なるようです。

(つまり、異なる正の定数を乗じたもの）で、*M* = 128 でピークに達し、その後、ミニバッチサイズとともに減少しました。これは、バッチサイズが異なると、拡散速度が多少異なることを示しています。次に，すべてのモデルを一定のエポック数で学習したため，バッチサイズが小さいほど，学習の総反復回数が多くなります．そのため、初期学習段階の終了時に到達した反復回数とそれに対応する重みの距離には、大きな違いがあります。

このことから、次のような非公式な議論が導かれます（平坦なミニマムが実際に一般化に重要であると仮定しています）。最初の学習段階で、「幅」*d* のミニマムに到達するためには、重みベクトル *wt* は少なくとも距離 *d* を移動しなければならず、これには長い時間（約 exp(*d*)反復）がかかります。したがって、幅の広い（「フラット」な）ミニマムに到達するためには、可能な限り高い拡散率（数値的に不安定にならない）と、多くのトレーニング反復が必要となります。次のセクションでは、これらの結論を実際に実行してみます。

# 異なるミニバッチサイズでのマッチングウェイト増量統計

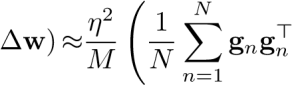
まず、異なるバッチサイズで観測された異なる拡散率を補正するために、重みの増分の統計を小さなバッチサイズのものと一致させることを目指します。



(a) 学習率調整と GBN の前 (b) 学習率調整と GBN の後図 2：初期化時の重みベクトルのユークリッド距離

学習率。 この論文では、推定された勾配に比例して重みの更新が行われる、おそらくモメンタムを伴う SGD を調査することを思い出してください。

Δ**w** ∝ *η***g**ˆ *,* (5) ここで、*η* は学習率であり、今はバッチの正規化の効果を無視しています。

付録のセクション A では、パラメータ更新ステップ Δ**w** の共分散行列が

!

cov(Δ**w**,(6))

ミニバッチインデックスを一様にサンプリングした場合（置き換えあり、または置き換えなし）、次のようになります。

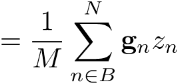
.したがって、共分散行列がすべての項目で同じになるようにする簡単な方法はミニバッチのサイズは選択する √

*η* ∝ *M ,* (7) つまり、ミニバッチサイズの平方根分だけ学習率を上げる必要があります。

Krizhevsky (2014) は、勾配期待値の分散を一定に保つために、同様の学習率スケーリングを提案したが、彼の設定ではより良い経験的結果に達したため、線形スケーリングのヒューリスティクスを使用することを選択したことに留意する。その後、Li (2017) も同じことを提案しています。

当然のことながら、このような学習率の増加は、平均ステップE[Δ**w**]も増加させます。しかし、E[Δ**w**]は通常、標準偏差よりも桁違いに小さいため、この効果は無視できることがわかりました。

さらに、次のように勾配の推定値に乗算ノイズを加えることで、1 次と 2 次の両方の統計量を一致させることができます。

**g**ˆ *,*

この方法では、バッチサイズを変更しても共分散は一定に保たれ、平均ステップ

E[Δ**w**]は変わりません。

どちらの場合も、最初の数回の反復において、発散を防ぐために勾配をクリップまたは正規化する必要がありました。どちらの方法も似たような性能を示したため（4 1 次の統計の影響が無視できるため）、より単純な学習率法を使うことにしました。

ここで重要なのは、他のタイプのノイズ（例えば、ドロップアウト（Srivastava et al,

1. 、ドロップコネクト

(Wan et al., 2013)、ラベルノイズ(Szegedy et al., 2016)）は共分散行列の構造を変化させ、そのスケールだけではないため、小バッチ増分の 2 次統計を正確に一致させることができません。したがって、これらのタイプのノイズが大きなバッチサイズの汎化ギャップを減らすのに役立つことはわかりませんでした。

最後に、上記の議論（および付録セクション A で提供される導出）では、各サンプルごとのグラジエント *gn* が選択されたミニバッチに依存しないと仮定したことに注意してください。しかし、これはバッチの正規化の影響を無視しています。次のサブセクションでは、この点を考慮します。

ゴースト・バッチ・ノーマライゼーション バッチ正規化（BN）（Ioffe & Szegedy,

1. は、学習を加速し、異なる初期化スキームに対するニューラルネットワークのロ

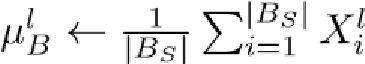
バスト性を高め、一般化を向上させることが知られている。しかし、BN はバッチ統計を

4 簡単な比較を付録（図 3）に示します。

使用するため、選択したバッチサイズに依存することがわかっています。私たちはこの依存性を研究し、実際の大きなバッチの代わりに小さな仮想（「ゴースト」）バッチの統計を取得することで、汎化エラーを低減できることを観察しました。実験では、(Ioffe & Szegedy, 2015)で提案されているように、完全なバッチ統計を推論段階で使用することが重要であることがわかりました。詳細は Algorithm 1 に記載されています。この修正は、それだけで一般化誤差を大幅に減少させます。

アルゴリズム 1：ゴーストバッチ正規化（GBN）、仮想ミニバッチ *BS* で大きなバッチ *BL* 上の活性化 *x* に適用。ここで *BS < BL*。

必要です。ラージバッチ上の *x* の値。*BL* = {x1*...m*} 仮想バッチのサイズ |*BS*|; 学習すべきパラメータ: *γ*, *β*, momentum *η* 訓練段階。

*BL* を{*X*1*,X*2*,...X*|*BL*|/|*BS*|}={x1*....*|*BS|,x*|*BS*|+1*...*2|*BS|...x*|*BL*|-|*BS*|...*m*} for  *l* =

1*,*2*,*3*...*{ゴーストミニバッチの平均値を計算する。}

 for *l* = 1*,*2*,*3*...* {ゴーストのミニバッチを計算します。}

*µrun* = (1 *- η*)|*BS|µrun* + *Pi*|*B*=1*L*|/|*BS*|(1 *- η*)*i* - *η - µBl σrun* = (1 *-*

*η*)|*BS*|*σrun* + *Pi*|*B*=1*L*|/|*BS*|(1 - *η*)*i* - *η* - *σBl* return  テスト段階。 *l*

return

{

scale and shift

}



マルチデバイスの分散環境では、「Ghost BN」の利点の一部がすでに発生している可能性があることに注意してください。というのも、追加の通信コストを回避するために、バッチノルム化が各デバイスで個別に事前に実行されることが多いからです。このように、各デバイスは、そのサンプル（すなわち、ミニバッチ全体の一部）のみを使用してバッチノルム統計を計算します。このようなバッチノルムの更新は、一般化に役立ち、バッチ全体でバッチノルム統計を計算するよりも良い結果が得られることは、著者の知る限りではまだ未発表の既知の事実である。なお、GBN は、デバイスごとのバッチ全体でバッチ統計を計算する商用フレームワーク（TensorFlow、PyTorch など）では提供されない、小さな（仮想）バッチサイズでの柔軟性を実現しています。さらに、それらの商用フレームワークでは、実行統計は通常、各更新部分を均等に重み付けすることで、

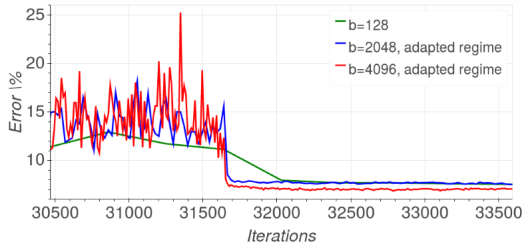
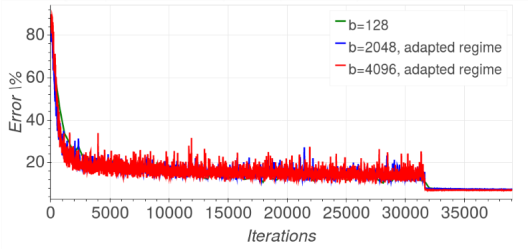
「Ghost BN」とは異なる方法で計算されます。我々の実験では，この方法では汎化性能が悪化することがわかりました．

学習率の調整と GBN の調整の両方を実施すると、第 6 章で説明するように、汎化性能が向上するようです。さらに、図 6 に見られるように、対数重み距離のグラフの傾きがより一致しているように見え、同様の拡散率を示しています。また、一定のシフトが見られますが、これはグラデーションのクリッピングに関係していると考えられます。このシフトはウェイトディスタンスを増加させるだけなので、性能を損なうものではないと考えられます。

# ウェイトの更新回数を調整することで、一般化のギャップを解消

セクション 3 の結論によると、初期の高学習率トレーニングフェーズでは、モデルがパラメータ空間のより遠くの位置に到達することができ、より広いローカルミニマムを見つけ、より良い一般化を行うために必要であると考えられます。図 2b を見ると、異なるバッチサイズのグラフを一致させるための次の明らかなステップは、初期の高学習率領域での学習反復回数を増やすことです。そして、現在の重みと初期化ポイントの間の距離が、学習率を下げるタイミングを決めるための良い指標になることに気づきました。これは一般的な方法とは異なることに注意してください。通常、実務者は検証誤差がプラトーに達したと思われる後に学習率を下げます。これは，オーバーフィットの恐れがあるため，検証誤差が「フラットライン」に達したときに最適化プロセスで学習誤差を

減少させてはならないという長年の信念によるものである（Girosi et al.，1995）．しかし，同じ条件で最適化を続けることで，最終的な精度を大幅に向上させることができることがわかりました．



(a) バリデーション・エラー (b) バリデーション・エラー - 拡大表示図 3：ラージバッチレジームの一般化を、スモールバッチトレーニングの性能に合わせて比較したもの。

学習率を下げていくと、検証誤差が急激に減少し、最終モデルの一般化が進みます。その後、学習率を下げていくと、検証誤差が急激に減少し、最終的なモデルの一般化が向上しました。

これらの観察結果から、「汎化ギャップ」現象は、バッチサイズではなく、更新回数が比較的少ないことに起因すると考えられました。具体的には，図 2 と我々のモデルから得られた知見をもとに，大規模なミニバッチの使用に適した学習方法を採用しました．最適化プロセスの時間枠を「ストレッチ」し、元の体制では *e* エポックの各期間が、使用するミニバッチサイズに応じたエポックに変換されます。この修正により、最適化のステップ数が、スモールバッチ領域で実行されたものと同じになります。図 3 に示すように、この修正と学習率の調整を組み合わせることで、先に見られた汎化ギャップが完全に解消されます5。

# 実験の様子

実験設定 一般的な画像分類タスクのセットを使って実験を行いました。

* MNIST (LeCun et al, 1998b) - 60*K* のトレーニングセットと，0 から 9 までの数字を表す 10*K* の28×28 グレースケール画像のテストセットで構成される．
* CIFAR-10 および CIFAR-100 (Krizhevsky, 2009) - それぞれ，サイズ 50K のトレーニングセットとサイズ 10*K* のテストセットで構成されています．インスタンスは，

10 または 100 のクラスを表す 32×32 のカラー画像である．

• ImageNet 分類タスク Deng et al. (2009) - 120 万サンプルのトレーニングセットと 5 万サンプルのテストセットから構成される。各インスタンスには，1000 個のカテゴリのうちの 1 つがラベル付けされている．

私たちの発見を検証するために、ニューラルネットワークモデルの代表的な選択を使用しました。完全連結モデルである F1 と、Keskar ら（2017）が提案した浅い畳み込みモデル C1 および C3 を使用しました。より最新のアーキテクチャのデモンストレーションと

5非適応レジームとの比較など、追加のグラフは付録（図 2）にあります。

して、モデルを使用しました。CIFAR10データセットにはVGG (Simonyan, 2014) とResnet44 (He et al., 2016)、CIFAR100 データセットには Wide-Resnet16-4 (Zagoruyko, 2016)、ImageNet データセットには Alexnet (Krizhevsky, 2014)を使用した。

各実験では、原著論文で提案されている学習レジームと、モーメンタム SGD オプティマイザを併用しました。4096 個のサンプルからなるバッチを「ラージバッチ」（LB）とし、

128 個（F1,C1,VGG,Resnet44,C3,Alexnet）または 256 個（WResnet）のスモールバッチ（SB）を使用しました。スモールバッチとラージバッチについて、オリジナルの学習ベースラインと、以下の方法を比較します[[2]](#footnote-2)。

* 学習率の調整（LB+LR）。*ηS* はスモールバッチに使用されたオリジナルの学習率、 *ηL* は適応された学習率、｜*BL*｜,｜*BS*｜はそれぞれラージバッチとスモールバッチ

のサイズである。

* ゴースト・バッチ・ノルム（LB+LR+GBN）。さらに、学習手順の中で「ゴースト・バッチ・ノーマライゼーション」法を使用しています。ゴーストバッチサイズ」は 128 を使用しています。
* レジームの適応。ゴースト・バッチノルムと同様に調整された学習率を使用しますが、適応された学習レジームを使用します。学習レジームは、使用される各バッチサイズの反復回数が同じになるように変更され、エポック数に大きなバッチの相対的なサイズを効果的に乗じます。

結果は？ 今回の実験では，我々の主張を実証的に裏付けることができました．表 1 に示された最終的な検証精度を見ると、過去の研究に従い、スモールバッチ(SB)からラージバッチ(LB)への移行には、確かにかなりの汎化ギャップがあることがわかる。しかし、ラージバッチ（+LR）の学習率を変更することで、このギャップは小さくなり、Ghost-BN 法

（+GBN）を用いることでさらに改善されます。最後に、学習方法を変更した場合

（+RA）には、一般化ギャップが完全になくなり、小ロットで得られたものと同等以上の検証精度が得られることがわかります。さらに、より難易度の高い ImageNet データセットで得られた結果を表 2 に示しますが、我々の手法は同様の効果を示しています。

表 1：検証精度の結果、SB/LB はそれぞれスモールバッチ、ラージバッチを表す。GBN は

Ghost-BN、RA は Regime adaptation の略です。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ネットワーク | データセ SB ット | | LB | +LR | +GBN | +RA |
| F1 (Keskar et al., 2017) | MNIST | 98.27% | 97.05% | 97.55% | 97.60% | 98.53% |
| C1 (Keskar et al., 2017) | Cifar10 | 87.80% | 83.95% | 86.15% | 86.4% | 88.20% |
| Resnet44 (He et al., 2016) | Cifar10 | 92.83% | 86.10% | 89.30% | 90.50% | 93.07% |
| VGG（Simonyan, 2014 | Cifar10 | 92.30% | 84.1% | 88.6% | 91.50% | 93.03% |
| C3(Keskar et al., 2017) | Cifar100 | 61.25% | 51.50% | 57.38% | 57.5% | 63.20% |
| WResnet16-4（Zagoruyko, 2016 | Cifar100 | 73.70% | 68.15% | 69.05% | 71.20% | 73.57% |

表 2：Alexnet トポロジーを用いた ImageNet top-1 の結果（Krizhevsky, 2014）、表記は表 1 と同じ。

4096 イメージ 57.10% 41.23% 53.25% 54.92% 59.5% スネット ネット

ネットワ

ーク

LB

サイ

ズ

データセ

ット

SB

LB

7

LR

+

8

GBN

+

+

RA

アレック

アレック 8192 イメージ 57.10% 41.23% 53.25% 53.93% 59.5% スネット ネット

# ディスカッション

大きなバッチサイズの使用に関して、2 つの重要な問題があります。1 つ目は，なぜバッチサイズを大きくすると汎化能力が低下するのか，どうすればこの現象を回避できるのか，という点です．2 つ目は，小バッチと同じ汎化性能を維持しつつ，大バッチを使用して（並列化を利用して）トレーニングの壁時計時間を短縮できるか？

本研究では、SGD のランダムウォーク挙動と、その拡散率とバッチサイズの関係を調べることで、最初の問題に取り組みます。この結果と経験的な観察に基づいて、小バッチトレーニング戦略と大バッチトレーニング戦略の間の汎化ギャップを縮めるための簡単な改善策を提案します。(1)勢いのある SGD、勾配クリッピング、学習率低下スケジュールを使用する、(2)学習率をバッチサイズに合わせる（平方根スケーリングを使用）、(3) 複数のパーティションでバッチノルム統計を計算する（「ゴーストバッチノルム」）、(4) 十分な数の高学習率トレーニング反復を使用する。

このように、今回の結果から得られた主なポイントは、これまでの考え方とは異なり、大きなミニバッチを使った学習には本質的な汎化の問題はないということです。つまり、大きなミニバッチを使ってモデルを学習しても、小さなミニバッチを使って学習したモデルと同じように一般化できるということです。これは最初の問題に対する答えですが、

2 つ目の問題は未解決です。

私たちの論文が初めて登場してから間もなく、この問題にも答えが出ました。Imagenet の Resnet モデルを使って、Goyal ら（2017）は、実際に、大きなバッチサイズを使ってトレーニングの大幅なスピードアップが達成できることを示しました。このことは、特に Goyal ら（2017）が我々が上述したものと同様のトレーニング方法を使用したことから、この作品でもたらされたアイデアと、将来のスケールアップにおける重要性をさらに強調しています。私たちの作品の主な違いは、Krizhevsky（2014）と同様に、また Bottou （2010）が提案したように、学習率の線形スケーリングを使用していることです。8しかし、我々は線形スケーリングが CIFAR10 ではあまりうまく機能しないことを発見し、後の研究では線形スケーリングルールが ImageNet の他のアーキテクチャではあまりうまく機能しないことを発見した(You et al., 2017)。

また、最適化の体制や明示的な学習率のアニーリングスケジュールに関する現在の「経験則」は誤っている可能性があることも指摘している。私たちは、一般的な慣習とは異なり、明らかな検証誤差の変化がなく、訓練誤差が下がり続けるような広範囲の勾配更新を行っても、良好な汎化が得られることを示しました。私たちの研究が発表された後、Soudry ら（2017）は、このことと、図 2 で観察された重み距離の対数増加に対する説明を提案しました。我々は、この挙動が分離可能なデータを持つ単純なロジスティック回帰問題でも起こることを示す。この場合、我々は漸近力学を正確に解き、**w**(*t*) =

log(*t*)**w**ˆ + *O*(1) (**w**ˆは *L*2 最大マージン分離器に対するもの)を証明する。したがって、学習誤差が非常に小さい場合でも、マージン（汎化に影響）はゆっくりと（*O*(*1/*log(*t*))とし

て）改善される。今後の課題としては、学習中にいつ、どのように学習率を下げればよいかを見つけることである。

結論本研究では、「汎化ギャップ」現象に取り組む最初の試みを行った。我々は、初期の学習段階は、経験的に観察されるように、重みの初期化からの距離が「超低速」で対数的に増加する、高次元の「ランダムなポテンシャル上のランダムウォーク」過程を用いて記述できることを主張する。この観察結果を受けて、我々は、パフォーマンスの低下

8 e.g. Goyal et al. (2017) も学習率に初期のウォームフェイズを使用していますが、このクリッピングは学習の初期段階でほとんどアクティブになっていたため、私たちが使用したグラデーションクリッピングと同様の効果があります。

に悩まされることなく、大規模なバッチでのトレーニングを可能にするいくつかのテクニックを提案します。これは、問題がバッチサイズではなく、むしろ更新の量に関係していることを意味する。さらに、シンプルかつ効率的なアルゴリズムである "Ghost-BN " を導入し、学習時間を維持したまま汎化性能を大幅に向上させることに成功しました。

謝辞最初の原稿に対するフィードバックをいただいたNir Ailon 氏、Dar Gilboa 氏、Kfir Levy 氏、 Igor Berman 氏に感謝いたします。本研究の一部は、Taub 財団、および Intelligence

Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center の支援を受けています。

(DoI/IBC）の契約番号 D16PC00003 です。米国政府は、政府の目的のために、著作権表示にかかわらず、複製および配布することを許可されています。免責事項：本書に記載された見解および結論は著者のものであり、IARPA、DoI/IBC、または米国政府の公式な方針または推奨を、明示的または黙示的に必ずしも表していると解釈されるべきではありません。

# リファレンス

Amodei, D., Anubhai, R., Battenberg, E., et al. Deep speech 2: End-to-end speech recognition in english and mandarin. *arXiv preprint arXiv:1512.02595*, 2015.

Bottou, L. Large-scale machine learning with stochastic gradient descent.*COMPSTAT'2010* の議事録、177-186 頁。Springer, 2010.

Bouchaud, J. P. and Georges, A. Anomalous diffusion in disordered media: statistical mechanism, models and physical applications.物理学レポート，195:127-293，1990．

Bouchaud, J. P. and Comtet, A. Anomalous diffusion in random media of any dimensionality.*J.Physique*, 48: 1445-1450, 1987.

Bray, A. J. and Dean, D. S., Statistics of critical points of Gaussian fields on large-dimensional spaces.*Physical Review Letters*, 98(15):1-5, 2007.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The Loss Surfaces of Multilayer Networks.*aistats15*, 38, 2015.

Das, D., Avancha, S., Mudigere, D., et al. Distributed deep learning using synchronous stochastic gradient descent. *arXiv preprint arXiv:1602.06709*, 2016.

Dauphin, Y. , de Vries, H., Chung, J., and Bengio, Y. Rmsprop and equilibrated adaptive learning rates for nonconvex optimization. corr abs/1502.04390 (2015).

Dauphin, Y., Pascanu, R., and Gulcehre, C. Identifying and attacking the saddle point problem in highdimensional non-convex optimization.In *NIPS*, pp.1-9, 2014.

Dean, J., Corrado, G., Monga, R., et al. 大規模分散型ディープネットワーク.In *NIPS*, pp.1223-1231, 2012.

Deng, J., Dong, W., Socher, R., et al. ImageNet:大規模な階層的画像データベース．In *CVPR09*, 2009.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. 機械学習研究会誌, 12(Jul):2121-2159, 2011.

Durrett, R. Subclassical limiting behavior を持つランダム環境における多次元ランダムウォーク.数理物理学の通信, 104(1):87-102, 1986.

Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping from saddle points-online stochastic gradient for tensor decomposition.In *COLT*, pp.797-842, 2015.

Girosi, F., Jones, M., and Poggio, T. Regularization theory and neural networks architectures.*Neural computation*, 7(2):219-269, 1995.

Goyal, P., Dollár, P., Girshick, R., et al. Accurate, large minibatch sgd:Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better:Stability of stochastic gradient descent.*ICML*, pp.1-24, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition.In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.770-778, 2016.

Ioffe, S. and Szegedy, C. Batch Normalization:Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning:Generalization gap and sharp minima.In *ICLR*, 2017.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A. Learning multiple layers of feature from tiny images.2009.

Krizhevsky, A. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

LeCun, Y., Bottou, L., and Orr, G. Efficient backprop in neural networks:トリック・オブ・ザ・トレード（Orr,

G. and Müller,

k.編著）。)*Lecture Notes in Computer Science*, 1524, 1998a.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition.*Proceedings of the IEEE*, 86(11):2278-2324, 1998b.

Li, M. *Scaling Distributed Machine Learning with System and Algorithm Co-design*（システムとアルゴリズムの共同設計による分散型機械学習の拡張）。PhD thesis, Intel, 2017.

Li, M., Zhang, T., Chen, Y., and Smola, A. J. 確率的最適化のための効率的なミニバッチ・トレーニング．

In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.661-670.ACM, 2014.

Luong, M.-T., Pham, H., and Manning, C. D. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Marinari, E., Parisi, G., Ruelle, D., and Windey, P. Random Walk in a Random Environment and 1f Noise.*Physical Review Letters*, 50(1):1223-1225, 1983.

Mnih, V., Kavukcuoglu, K., Silver, D., et al.深層強化学習による人間レベルの制御（Human-level control through deep reinforcement learning）。*Nature*, 518(7540):529-533, 2015.

モンタボン、G.、オア、G.、ミュラー、K.-R.*Neural Networks:trick of the trade*.2 版、2012 年。ISBN 978-

3-642-35288-1.

Ruder, S. An overview of gradient descent optimization algorithms.*CoRR*, abs/169.04747, 2016.

Silver, D., Huang, A., Maddison, C. J., et al. Mastering the game of go with deep neural networks and tree search.*Nature*, 529(7587):484-489, 2016.

Simonyan, K. e. a. Very deep convolutional networks for large image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Soudry, D., Hoffer, E., and Srebro, N. The Implicit Bias of Gradient Descent on Separable Data（分離可能なデータにおける勾配降下法の暗黙のバイアス）.*ArXiv e-prints*, October 2017.

Soudry, D. and Hoffer, E. Exponentially vanishing sub-optimal local minima in multilayer neural networks. *arXiv preprint arXiv:1702.05777*, 2017.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting.*Journal of Machine Learning Research*, 15(1):1929-1958, 2014.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning.In *International conference on machine learning*, pp.1139-1147, 2013.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision.In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.2818-2826, 2016.

Wan, L., Zeiler, M., Zhang, S., LeCun, Y., and Fergus, R. dropconnect を用いたニューラルネットワークの正則化。ICML'13, pp.III-1058-III-1066.JMLR.org, 2013.

Wu, Y., Schuster, M., Chen, Z., et al. Google のニューラル機械翻訳システム。人間と機械翻訳の間のギャップを埋める。*CoRR*, abs/1609.08144, 2016.

You, Y., Gitman, I., and Ginsburg, B. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.0388*, 2017.

Zagoruyko, K. ワイドな残差ネットワーク。In *BMVC*, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization.In *ICLR*, 2017.

Zhang, S., Choromanska, A. E., and LeCun, Y. Deep learning with elastic averaging sgd.In *NIPS*, pp.685-693, 2015.

1. *α*/2

   なお、この形式は eq.(1)と一致しており、*f* (*x*) = *x* . [↑](#footnote-ref-1)
2. コードは、[https://github.com/eladhoffer/bigBatch。](https://github.com/eladhoffer/bigBatch)

   7 メモリの制限のため、これらの実験は 2048 のバッチで行われました。 [↑](#footnote-ref-2)