ラージバッチ・トレーニングのパフォーマンスに及ぼすネットワーク幅の影響

Lingjiao Chen1 , Hongyi Wang1 , Jinman Zhao1,

Paraschos Koutris, 1 Dimitris Papailiopoulos2

1 コンピュータサイエンス学科 2 電気・情報工学専攻ウィスコンシン大学マディソン校

# アブストラクト

ミニバッチ方式の確率的勾配降下法（SGD）の分散実装では，通信のオーバーヘッドに悩まされる．大規模なバッチで学習することで、このようなオーバーヘッドを軽減することができるが、アルゴリズムの収束性や汎化性能が低下する。本研究では、ニューラルネットワークの構造（幅と深さ）がラージバッチトレーニングの性能にどのように影響するかを分析するための第一歩を踏み出しました。本研究では，ニューラルネットワークの構造（幅と深さ）が大規模一括学習の性能にどのように影響するかを分析するための第一歩を踏み出しました。また、残差ニューラルネットワークと完全結合ニューラルネットワークを用いた大規模な実験により、深いニューラルネットワークとは異なり、広いニューラルネットワークは収束の遅れを生じることなく大規模なバッチを用いて学習できることが示唆された。

# はじめに

確率的最適化アルゴリズムの分散実装は，大規模なモデル学習の標準となっています [1, 2, 3, 4, 5, 6, 7]．Tensorflow [1]、MxNet [4]、Caffe2 [7]など、ほとんどの機械学習フレームワークは、デフォルトの分散学習アルゴリズムとしてミニバッチ SGD のバリエーションを実装しています。ミニバッチ SGD の分散型反復処理では、パラメータサーバ(PS) がグローバルモデルを保存し、*P* 個の計算ノードが合計 *B* 個のグラデーションを評価します。B は一般的にバッチサイズと呼ばれています。PS はすべての計算ノードからこれらの *B* 個のグラデーションの合計を受け取ると、それをグローバルモデルに適用し、モデルを計算ノードに送り返して、新たな分散反復を開始します。

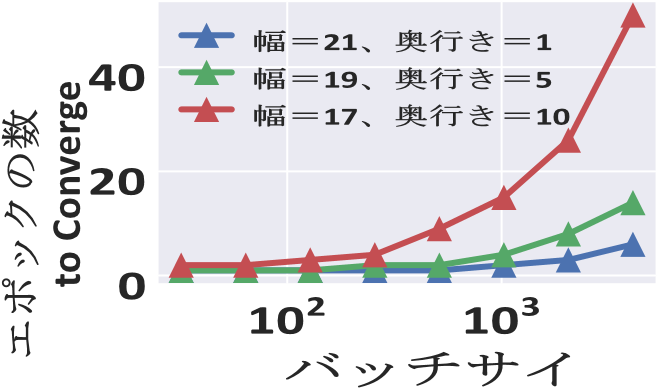
分散実装の大前提はスピードアップであり，すなわち，*P* 個の計算ノードと 1 個の計算ノードの間でどれだけトレーニングが速くなるかである．しかし，実際には，数十台の計算ノードを超えたあたりで，この高速化は飽和してしまいます[6, 8, 9]．これは，通信がボトルネックになるためです．すなわち，固定の *B* 個の例のバッチに対して，計算ノードの数が増えると，これらのノードはグラデーションの計算よりも PS への通信に多くの時間を費やすことになります．このボトルネックを軽減するために，最近では，低精度のトレーニングと勾配のスパース化に関する研究が数多く行われています（例：[10,

11, 12]）．

これらのオーバーヘッドを軽減する別のアプローチは，バッチサイズ*B* を大きくすることです．最近では，最先端のモデルやデータセットを用いた大規模なバッチトレーニングを可能にする洗練された手法が開発されています[13, 14, 15]．一方で，大規模なバッチトレーニングは，モデルの汎化性に影響を与えたり[16]，収束が遅くなったり[17, 18,

19]，ハイパーパラメータの調整ミスに敏感になったりすることが示唆されている[20]．いくつかの理論的な結果[21, 18, 22, 19, 17]によると、バッチサイズ *B* が問題に依存した閾値 *B*∗よりも大きくなると、収束までの総反復回数が大幅に減少することが示唆されています。

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

が増加すると、より大きな *B* を使用することができなくなります。これらの研究の中には、暗黙的または明示的に、閾値 *B*∗ がバッチ内の勾配の類似性に よって制御されることを示しているものがあります。特に [19] は、勾配の多様性という尺度が、*B* とミニバッチ SGD の収束速度の関係を直接制御することを示しています。勾配の多様性は、同時に処理される勾配の類似性を測定するもので、[19]は、多様性が高いほど、問題が高速な大規模バッチトレーニングに適していること、また、分散環境でのスピードアップの程度が大きいことを理論的および実験的に示しています。

ニューラルネットワークの構造が、モデルの複雑

さや容量[23, 24, 25]、表現効率にどのような影響を与えるかについては、多くの研究がなされています。

[26]、予測精度[27, 28]などがあります。しかし、 図 1：ニューラルネットワークの構造がニューラルネットワークの構造が、分散型高速化 大規模なバッチトレーニングへの適応性への適応性にどのように影響するかを理解するた に与える影響。MNIST における ReLU を用めの研究はほとんどありません。 いた完全連結モデルを対象としている．本研究では、さまざまなネットワークアーキテク 各完全連結ネットワークについて、バッチャの勾配の多様性を分析することで、以下の疑 チサイズを変化させ、MNIST 上で96%の問を解決するための一歩を踏み出しました。ニュ精度に収束するまでのエポック数を測定ーラルネットワークの構造は、高速なラージバッした。広くて浅いネットワークは、狭くチトレーニングへの適応性にどのように影響するて深いネットワークよりも収束までのエか？ ポック数が少なくて済むことから、前者私たちの貢献 私たちは、理論的な接続を確立する の方がより多くの計算ノードにスケールアウトするのに適していることがわかる。

ニューラルネットワークの構造（深さと幅）と、収束速度を落とさずにどれだけバッチサイズを大きくできるかの指標である勾配の多様性との間の関係を明らかにした[19]。特に、2 層の完全連結線形および非線形ニューラルネットワークと、多層の完全連結線形ニューラルネットワークの 2 種類のネットワークについて、勾配の多様性が幅と深さの関数としてどのように変化するかを証明します。理論的な分析によると、意外にも、幅が大きく、深さが小さくなるにつれて、勾配の多様性が単調に増加することがわかった。高いレベルでは、ネットワークの幅が広いほど、グラデーションが多様化するためのスペースが広くなる。この結果は、広くて浅いネットワークは、深いネットワークに比べて、高速な大規模バッチ学習に適していることを示唆している。図 1 は、この現象の例を示している。

我々の理論的発見を裏付ける広範な実験結果を提供する。CIFAR10，MNIST，EMNIST， Gisette，および合成データセットを用いて，完全連結ネットワークと残差ネットワークの実験を行った．実験では，ネットワークのパラメータ数を固定し，深さと幅を変化させ，（ステップサイズを調整した後に）バッチサイズ *B* の精度に達するまでにデータを何回通過させるかを測定した．この閾値 *B*∗は、ネットワークが深くなるほど小さくなり、ネットワークが深くなると高速な大規模バッチ学習ができなくなるという理論的な結果が検証されました。

本研究の主要なメッセージを要約すると、分散型ミニバッチ SGD の通信ボトルネックは、通信効率の高いアルゴリズムを設計するだけでなく、ラージバッチトレーニングを可能にするためにニューラルネットワークのアーキテクチャを最適化することによっても、部分的に克服することができます。

# 関連する仕事

ミニバッチ 最適なバッチサイズの選択は，非強収束モデル[21]，最小二乗回帰[22]， SVM[29]などで研究されている．また，バッチサイズをその場で自動的に選択する手法も提案されている[30, 31]．ミニバッチアルゴリズムは、加速勾配降下アルゴリズム[32]と組み合わせたり、デュアル座標降下を用いたりすることができる[33, 34]。ミニバッチプロキシマルアルゴリズムは[35]で紹介されています。これまでの研究では、主に（強く）凸なモデルや特定のモデル（最小二乗回帰や SVM など）に焦点を当てていましたが、本研究では、ニューラルネットワークの構造が最適なバッチサイズにどのように影響するかを研究しています。

勾配の多様性 これまでの研究では、ミニバッチは、勾配バッチの多様性を高めることで、

より良い収束率を達成できることが示されています。例えば、層化サンプリング[36]、 Determinantal Point Processes[37]、アクティブサンプリング[38]などを用いています。勾配間の類似性の概念と、それが収束性能にどのように影響するかについては、いくつかの論文で研究されています[17, 18, 19]。勾配多様性の正式な定義と分析は[19]に記載されており、凸型および非凸型モデルの勾配多様性と最大バッチサイズの関係を確立しています。我々の知る限り、既存の研究では、勾配多様性（したがって、最適なバッチサイズ）をニューラルネットワークの構造と関連付けるものはない。

人工ニューラルネットワークにおける幅と深さ 深層ニューラルネットワークと幅広ニューラルネットワークの品質に関する関心と議論が高まっています。[23]は，ディープネットワークはワイドネットワークよりも複雑さが大きいため，より優れたモデルを得ることができる可能性を示唆しています．また，[26]は，ディープネットワークがワイドネットワークよりも効率的に和積を近似できることを証明している．一方、[39]は、ワイドネットワークのクラスが、少なくともディープネットワークと同じくらいの精度を達成できることを示している。また，[40]では，浅いネットワークと深いネットワークの 2 つのクラスを提示し，それらが顕著な予測誤差を達成することを示しています．実際、[41]では、うまく設計されたシャローニューラルネットワークは、多くのディープニューラルネットワークを凌駕できることを示しています。さらに最近では、[27]が、密な構造を用いて、広くて浅いネットワークが、深いネットワークに比べて、精度を大幅に向上させることを示しています。また，[42]では，幅を大きくすることで，より良い最適化ランドスケープが得られることを示している．これまでの研究では、主にネットワーク構造が予測精度に与える影響を研究してきましたが、本研究では、分散計算におけるバッチサイズの最適な選択に与える影響に焦点を当てます。

# セットアップと前置き

このセクションでは、必要な背景と問題設定を紹介します。

ミニバッチ SGD データからモデルを学習するプロセスは、経験的リスク最小化（ERM）と呼ばれる最適化問題として考えることができます。

min `(**w**;(*xi,yi*)) **w**

ここで、*xi*∈*Rm* は i 番目のデータポイントを表し、*n* はデータポイントの総数、**w**∈*Rd* はパラメータベクトルまたはモデル、`(-;-)は各データポイントに対するモデルの予測精度を測定する損失関数である。上記 ERM を近似的に解く一つの方法として、ミニバッチ式の確率的勾配降下法（SGD）があり、以下のように動作する。

(*k*+1)*B*-1

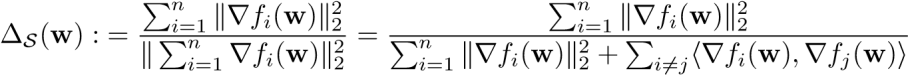
**w**(*k*+1)*B* = *wkB* - *γ* X ∇*fs*`(*wkB)*。(3.1)

`=*kB*

ここで、各インデックス *s`*は、[*n*]から一様にランダムに置き換えて描画されます。*k* 回の分散反復の後に得られるモデル、すなわち*kB* 回の勾配更新の総数を示すために、**w** に添え字をつけて *kB* とします。関連する研究では、バッチ計算に正規化係数が含まれていることが多いが、ここではステップサイズ *γ* に含まれている。

勾配の多様性と収束の速さ 勾配の多様性は、損失関数の個々の勾配が互いに異なる度合いを測定します。

定義 1 (Gradient Diversity [19]).以下の比率を*Gradient Diversity* と呼ぶことにする。

*.*

勾配の多様性 ΔS(**w**)は、異なるデータポイントに関して取られた勾配の間の内積が小さいときに大きくなる。勾配の多様性の概念を用いて、各データセット S と各 **w** のバッチサイズ境界 BS(**w**)を以下のように定義する。

BS(**w**) := *n* - ∆S(**w***).*

次の結果[19]は、勾配の多様性という概念を使用して、ミニバッチ SGD の収束率を捉えています。

Lemma 1*. [Theorem 3 in [19],Informal]* 各反復において *B ≤ δ* - *n*∆S(**w**) + 1,∀**w** であるとする。シリアル *SGD* が *T* 回の勾配更新後に「準最適解」を達成する場合、シリアル *SGD* と同じステップサイズを使用して、バッチサイズ *B* のミニバッチ *SGD* を使用すれば、同じ回数の勾配更新*/*データパス（すなわち *T/B* 反復）後に「準最適解」を達成することができます。

上記の結果は凸型問題と非凸型問題の両方に当てはまり、その主なメッセージは、ミニバッチ SGD はバッチサイズが *n* - ∆S(**w**)よりも小さい限り、スピードアップの飽和に悩まされないということです (定数まで)。さらに、[19]では、これが最悪の場合の最適境界であることも示しています。つまり、（大まかには）バッチサイズが勾配の多様性の *n* 倍よりも大きい場合、ミニバッチ SGD の収束速度がシリアル SGD の収束速度よりも遅くなるようなモデルが存在するということになります。

## この作品で研究している主な理論的課題は、「ニューラルネットワークの構造（深さや幅）が変化すると、グラデーションの多様性はどのように変化するのか？

完全連結型ニューラルネットワーク *L*≧2 層の線形および非線形の完全連結型ネットワークを考える。ここで，``∈{0*,...,L*}の場合，*``*番目の層の幅（ノード数）を *K`*で表す．最初の層は次元 *d* の入力に対応するので，*K*0 = *d* である．最後の層はニューラルネットワークの単一の出力に対応するので，*KL* = 1 である．層と` - 1 層を結ぶエッジの重みは，*l*∈{1*,...,L*} の場合，行列 *W*`∈*RK`*×*K*`-1 で表されます．簡略化のために、重みの集まり（すなわちモデル）を **w** = (*W*1*,W*2*,...,WL*)と表現する。

*L*≧2 層の一般的なニューラルネットワーク（NN）は、*W*`∈*RK`*×*K*`-1 の行列 *W*1*,...,WL* と、（一般的には非線形の）活性化関数 *σ*(-)の集合体として記述することができます。入力データ点 *xi* に対する NN（または LNN）の出力は、*y*ˆ*i* = *WL* - *σ*(---*σ*(*W*2 - *σ*(*W*1 - *xi*)))と定義されます。

活性化には様々な種類があり、tanh(*x*)、ソフティーン関数

と、ReLU 関数 max{0*,x*}を用いています。線形ニューラルネットワーク

(LNN)の場合、以下のように表します。

.また、*W`,p,q* と書くことで、*p* 番目の行列 *W`*の行と *q* 番目の列です。

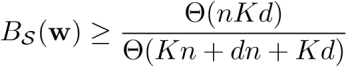
入力 *xi* を持つニューラルネットワークの出力を *y*ˆ*i* と定義する。本論文の理論部分では、誤差を測定するために二乗損失関数を使用し、*i* 番目のデータポイントについては、とさらに、データが達成可能であると仮定します。すなわち、*W`,p,q* = *W`,p,q*∗ のとき、各データポイントで損失関数が 0 になるような *W`,p,q*∗ が存在すると仮定します。

# 主な結果

このセクションでは、ニューラルネットワークの構造的特性、特に深さと幅が、勾配の多様性、ひいてはバッチサイズ *B* を変化させた場合のミニバッチ SGD の収束率にどのように影響するかについての理論的分析を示します。 すべての証明は付録に譲ります。以下の導出では、*n* 個のデータポイントのラベル{*y*1*,...,yn*}が実現可能であること、すなわち、入力 *xi* に対して *yi* を出力する *L* 層のネットワークが存在することを仮定します。我々の結果は、確率論的な記述として示され、ほとんどすべての重み行列に対して適用されます。

ウォームアップ：2 層線形ニューラルネットワーク 最初の結果は、1 つの隠れ層を持つ単純な 2 層線形ニューラルネットワークの場合です。表記を簡単にするために、隠れ層の幅を *K* = *K*1 とします。また、Θ(-)と Ω(-)は標準的な意味で使用します。主な結果は次のようになります。

Theorem 1.*2 LNN* を考える。重みと*xi* を独立に描かれた確率変数とし、*k*≦4 の*k* 次モーメントが後続区間に束縛されるようにする。このとき、任意の一定確率で以下が成立する。



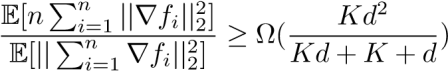
十分に大きな *n* に対して、バッチサイズに対する上記の比率は、 .この比率は，隠れ層の幅の関数として常に増加しており，幅が大きいほどバッチサイズが大きくなることを意味しています．

2 層非線形ニューラルネットワーク 理論解析の次のステップとして，非線形活性化関数 *σ* を持つ一般的な 2 層 NN を解析します。

定理 2.単調な活性化関数*σ* を持つ*2* 層*NN* を考えると、すべての*x* に対して次のようになる。*σ*(*x*)=*σ*(-*x*)であり、｜*σ*(*x*)｜と *supx*{*xσ*0(*x*)}はともに有界である。*l*∈{1*,*2}の重み *Wl,p,q,Wl,p,q*∗と、*xi* をN(0*,*1)からの*i.i.d.*確率変数とする。すると、高い確率で次のことが成り立つ。

*.*

*.*



## ここでは、期待値は

上の定理は、2 層 LNN の場合に得られた定理よりも弱いことを指摘しておきます。なぜなら、この定理は期待値の比を制限するものであり、比の期待値を制限するものではないからです（バッチサイズ制限）。それにもかかわらず、我々はバッチサイズ境界が集中していると推測しており、したがって、上記の定理はバッチサイズ境界をよく近似することができます。

また，NN でよく使われる活性化関数である tanh，arctan，softsign 関数などは，上記定理の仮定を満たしています。ここでは、2 層 LNN の場合と同じ傾向が見られます。(*i*) 幅が大きいほど勾配の多様性が大きくなり、分散ミニバッチ SGD の収束が速くなる、(*ii*) 比率が Ω(*d*)を超えることはない。

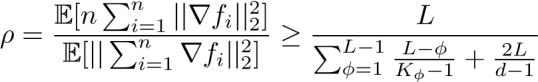
多層リニアニューラルネットワーク ここでは、2 層の LNN に関する結果を、任意の深さ

*L*≧2 の一般的な多層 LNN に一般化します。以下に主な結果を示します。

## 定理 3 です。*l*∈{1*,...,L*}の重み値*Wl,p,q* と*xi* をN(0*,*1)から独立に引いた確率変数とする。多層 *LNN* を考えると



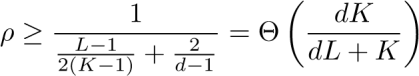
*.*すべての``∈{0*,...,L* - 1}に対して*K*`≧2 であるとし、*n* はが十分に大きい場合、次のようになります。

*.* (4.1)

上記の境界は、期待値の比ではなく、期待値の比を制限するため、2 層 LNN の場合に得られた境界よりも弱いことに注意してください。これは、分母と分子がそれぞれの期待値の周りに集中することで、期待値の比が期待値の比を反映すると考えられています。

このことが証明できるかどうかは興味深い未解決問題です。

次に、定理 3 がミニバッチ SGD の収束率に与える影響について説明します。この場合、式(4.1)の比は次のようになります。



上の境界から得られるものは 3 つあります。まず、LNN の幅 *K* を大きくすると、比率も大きくなり、収束率が高くなることがわかります。第二に、深さ *L* の効果は逆で、深さを増すと比率は減少します。第 3 に、比率は Θ(*d*)を超えることはないが、任意に小さくすることができる。ここで、LNN の重みの総数を固定して、各層の幅を増やしていく

（つまり深さが減っていく）とします。この場合、比率も大きくなります。

最後に、すべての層の幅が同じであるという単純化された仮定を捨てれば、幅と深さに対する境界の挙動は同じであることを指摘しておきます。

# 実験

本節では，様々なデータセットとネットワークアーキテクチャを用いて，ニューラルネットワークの構造（幅と深さ）が，大規模なバッチトレーニングへの適応性にどのような影響を与えるかについて，実証的な結果を示します。我々の主な発見は 3 つある。

1. 今回使用した全てのニューラルネットワークには、閾値 *B*∗が存在し、この閾値よりも大きなバッチサイズを使用すると収束が遅くなります。
2. 広いニューラルネットワークの閾値は、深いニューラルネットワークの閾値よりも大きいことが多い。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| データセット | 合成 | MNIST | Cifar10 | EMNIST | ジゼット |
| データポイント | 10,000 | 70,000 | 60,000 | 131,600 | 6,000 |
| モデル | リニア FC | FC/LeNet | ResNet-18/34 | FC | FC |
| ♪ クラス | +∞ | 10 | 10 | 47 | 2 |
| # パラメータ | 16k | 16K /  431K | 11m / 21m | 16k | 262k |
| コンバージド・アキュラシー | 10-12（損失 | 96% / 98% | 95% | 65% | 95% |

表 1：使用したデータセットとそれに関連する学習モデルおよびハイパーパラメータ

1. 同じ大きなバッチサイズを使用した場合、ほとんどすべてのワイド・ニューラル・ネットワークは、ディープ・ニューラル・ネットワークと比較して、収束に必要なエポック数が非常に少なくなります。

これらの結果は、我々の理論的な分析を検証するものであり、幅の広いニューラルネットワークは実際に大規模なバッチトレーニングに適しており、したがってスケールアウトに適していることを示唆しています。

Keras [43]を用いて実験用パイプラインを実装し，Amazon EC2 上の p2.xlarge インスタンスですべての実験を行いました．報告されたすべての結果は，5 回の独立した実行の平均値である．

データセットとネットワーク 実験に使用したデータセットとネットワークの概要を表 1 に示します。合成データセットでは，理論的な結果と同様に，すべてのデータポイントを N(0*,*1)から独立して抽出した．真のラベルを生成するために，重みが N(0*,*1)から独立して生成された深層線形完全連結ニューラルネットワーク(FC)が使用された．合成データ上のタスクは、回帰タスクです。合成データセットで線形 FC を学習します。我々が使用した実世界のデータセットは、MNIST[44]、EMNIST[45]、Gisette[46]、CIFAR-10[47] であり、適切なネットワークは、線形から非線形の完全連結のもの、そして LeNet[48]と

ResNet[28]までである。

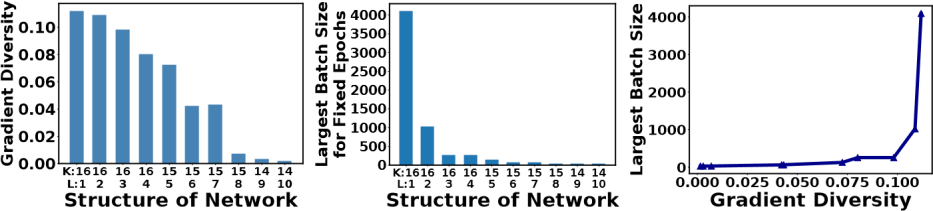
完全連結ネットワークと LeNet では，深さ *L* を1 から 10 まで変化させ，それに応じて *K* を変化させて，パラメータの総数がほぼ固定されるようにしています。より正確には、パラメータの総数 *p* を固定し、以下の式を解く。

din × *K* + (*L* - 1) × *K*2 + *K* × dout = *p.* となります。

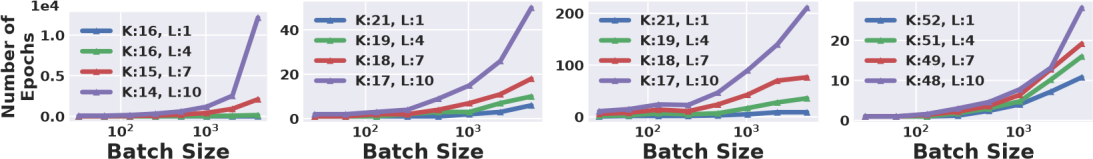
ここで、*din* はデータの次元、*dout* は出力のサイズです。ResNet では，2 つのパラメータを別々に変化させます．まず，完全連結層の幅と深さを変化させますが，残余ブロックは変化させません．次に、完全連結層を固定して、各チャンクのブロック数と畳み込みフィルタを変更します。ここでは、[28]で説明した残差関数のビルディングブロックを chunk と呼びます。ResNet-18/34 アーキテクチャでは、特定の構造を表すために[*s*1*,s*2*,s*3*,s*4] を使用し、*s*1 は第 1 のチャンクに積み上げられたブロックの数、*s*2 は第 2 のチャンクに積み上げられたブロックの数、などを表します。深さを変化させるために、各チャンクのブロックを 1 つずつ増やしたり減らしたりして、各ブロックの畳み込みフィルタの数を調整し、固定のパラメータ数の要件を満たします。

NN アーキテクチャの深さと幅の組み合わせごとに，分類タスクでは学習精度に一定の閾値を設定し，回帰タスクでは損失を設定してモデルを学習します．次に、様々なバッチサイズ（以下、ステップサイズ）で NN を学習しステップサイズの調整は、以下の方法で行います。(i）すべての学習率 *η* について、データを 2 回通過させた後、（ii）2 回のエポック後の学習損失が最も小さくなる *η*ˆを選択します。エポックは、データのフルパスを表します。

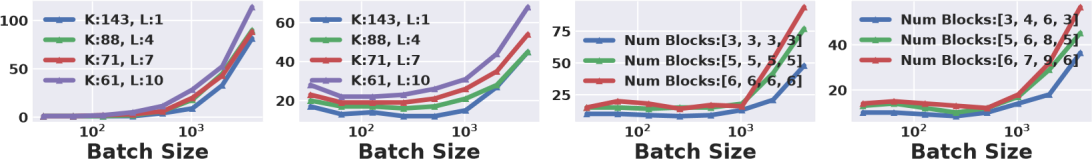
実験結果 最初に，勾配の多様性が大規模なバッチトレーニングへの適応性を反映しているかどうかを検証する．幅と深さを固定した各線形 FC ネットワークについて、10 エポックごとに勾配の多様性を測定し、その平均値を算出する。図 2(a)は深さと幅の変化に伴う平均的な勾配多様性の変化を示し、図 2(b)は各ネットワークがあらかじめ設定したエポック数内で収束する最大のバッチを示している。いずれも、ネットワークの幅 *K* が大きくなるにつれて増加しています。実際、図 2(c)に示すように、収束率に影響を与えない最大のバッチサイズは、勾配の多様性に応じて単調に成長します。これにより、勾配の多様性を利用して、大規模なバッチトレーニングへの適応性を捉えることができるという、我々の理論的分析が検証された。



(a) 勾配のあるダイバーシティ (b) 最大のバッチサイズ(c) ダイバーシティ対バッチサイズ図 2：回帰タスクの合成データセットで学習した線形 FC に対する勾配多様性の効果。(a) 異なる幅 /深さに対する勾配の多様性 (b) あらかじめ設定された数(すなわち14)のエポック内で損失10-12 に収束するための最大のバッチサイズ。(c) 勾配の多様性に対する最大のバッチサイズ。



(a) 合成、リニア FC (b) MNIST、FC (c) EMNIST、FC (d) ジゼット、FC



(e) MNIST, LeNet (f) Cifar10, ResNet18, FC(g) Cifar10, ResNet18, Res(h) Cifar10, ResNet34, Res

図 3：表 1 で示された同じ損失／精度に収束するのに必要なエポックの数。*K*は幅、*L*は深さを表す。(f)では、ResNet 18 の残差ブロックを固定し、完全連結部分のみを変化させた。(g)と(h)では、完全連結層を固定し、ResNet18 と ResNet34 の残差ブロックを変化させています。

次に，実世界のデータセットに対して異なるバッチサイズを使用した場合に，収束に必要なエポック数を調べました．まず，ほぼすべてのネットワークアーキテクチャにおいて，バッチサイズの閾値が存在し，これよりも大きなバッチサイズを使用すると，収束に必要なエポック数が多くなることが分かりました．これは，[19]の観察結果と一致します．例えば，図 3(b)では，バッチサイズが 256 より小さい場合，幅 *K*=17，深さ *L*=10 の

FC ネットワークでは，収束に必要なエポック数が 2～3 と少ない．しかし、バッチサイズが 256 よりも大きくなると、収束に必要なエポック数が大幅に増加し、例えば、バッチサイズが 4096 の場合、収束までに 50 回のエポックが必要となります。さらに、幅が大きくなるにつれて、閾値が大きくなることも確認できました。図 3(b)に示すように，FC ネットワークの *L* = 10 の場合，バッチサイズの閾値は 256 ですが，*L* = 1 の場合は 1024 まで上がります．さらに、同じ大きなバッチサイズを使用した場合、幅の広いネットワークは深いネットワークよりも収束までのエポック数が少なくなる傾向があります。例えば、図 3(c)では、4096 のバッチサイズを使用した場合、幅 *K* が 17 から 21 に増加すると、収束に必要なエポック数は 211 から 9 に減少します。これらの傾向は、実験で使用したすべての FC ネットワークで同様です。

ResNets と LeNet に関しては、傾向は必ずしもシャープではありません。これは、我々の理論的な分析がこのようなケースをカバーしていないことから予想されることですが、それでも主要な傾向は観察することができます。例えば、図 3（e）と（f）に示すように、固定のバッチサイズの場合、幅を大きくすると収束までのエポック数が減少することがほとんどです。図 4 は、各ネットワークアーキテクチャの収束までの正確なエポック数をヒートマップで表したものです。興味深いことに、ResNet では、深さを増やすことでも収束までのエポック数が減少するケースがわずかながら存在することがわかります。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1 1 1 1 1 1 1 1 1 1**  **1 1** **1** **1** **1** **1** **1** **1** **1** **1**  **1 1** **1** **1** **1** **1** **1** **1** **2** **2** | | | | | | | |  |
|  | | | | | | |  |
| 1. **2** **1** **2** **2** **2** **2** **2** **2** **3** 2. **2** **3** **3** **3** **3** **4** **4** **4** **5** | | | | | | | |
| **4 4** **4** **5** **6** **7** **6** **7** **7** **8** | | | | | | | |
| **7 8** **9** **10** **11** **13** **13** **14** **16** **13** | | | | | | | |
|  |  |  |  |  |  |  |  |

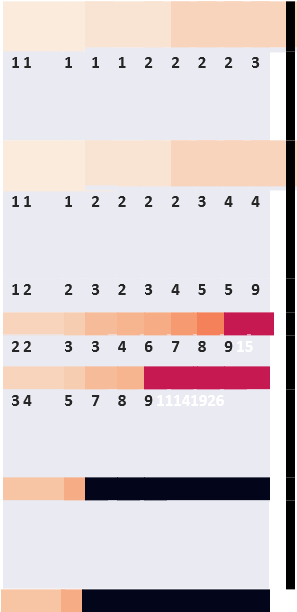
|  |
| --- |
| **0.80.91.01.21.51.71.92.12.2**  **1.01.11.21.61.72.02.22.42.5** |
| **1.01.61.51.82.12.32.52.72.8**  **1.31.81.62.12.42.62.83.03.1**  **1.52.01.92.42.73.03.13.43.4** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  | **5 20 24** | |
|  | **2 4 5 7 9 9 14** |
|  | |  | |  |
| **2 2 3 5 7 8 11** | | **5 18 23** | | |
| **3 4 5 7 12 13 24 29 27 47**  **5 8 13 17 22 26 43 51 62 90**  **9 13 23 28 44 56 70 92115140** | |  | | |

多くの実用的なアプリケーションでは、時間やリソースの制約から、合理的で限られた数のデータパスしか実行できません。そこで、ネットワークの構造が、一定のエポック数/データパス数であらかじめ指定された精度に収束させるための最大のバッチサイズにどのような影響を与えるかについても研究しました。図 5 に示すように、幅 *K* が大きいニューラルネットワークでは、事前に設定した少ない総エポック数で収束させるために、より大きなバッチサイズが可能になります。これは、次のような場合に特に有効です。



**32 0.50.60.70.70.81.01.11.31.51.9 4. 032 1 1 1 1 1 1 1 1 1 2 5032 1 1 2 2 3 4 6 9 9 11  2003225**

 **64**

**0.60.70.80.81.01.21.41.61.81.964 1 1 1 1 1 1 1 1 2 2**  **4064 2 3 2 3 5 6 8**

**11 14 15  1606420**

**0.63. 2128128128**

**128**

**256 0.62. 42563025612025615**

**512 0. 75122051280102451210**

**1024 0.81.6 10241024 2048 1.0204810 20484020485**

**18 18 19 21 21 409628**  **1.11.72.22.22.82.83.33.43.64.1**

**1 2 3 4 5 6 7 8 910** 隠れた層の深さ

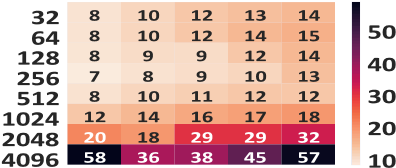
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1 1 1 1 1 1 1 1 1 1**  **1 1** **1** **1** **1** **1** **1** **1** **1** **1** | | | | | |  |
| **1 1** **1** | **1** | **1** | **1** | **2** **2** **2** **2** | |
|  |  |  |  |  | |
| **1 2 2** | **2** | **2** | **3** | **3 3 4 5** | |
|  |  |  |  |  | |
| **4 4** **4** **6** **7** **8** **6** **9** **8** **12** | | | | | |
| **9 14** **16** **18** **15** **21** **20** **18** **28** **28** | | | | | |
| **33 41** **36** **45** **41** **48** **43** **46** **44** **52** | | | | | |
|  |  |  |  |  |  |

(a) 合成、リニア FC

**1 2 3 4 5 6 7 8 910**

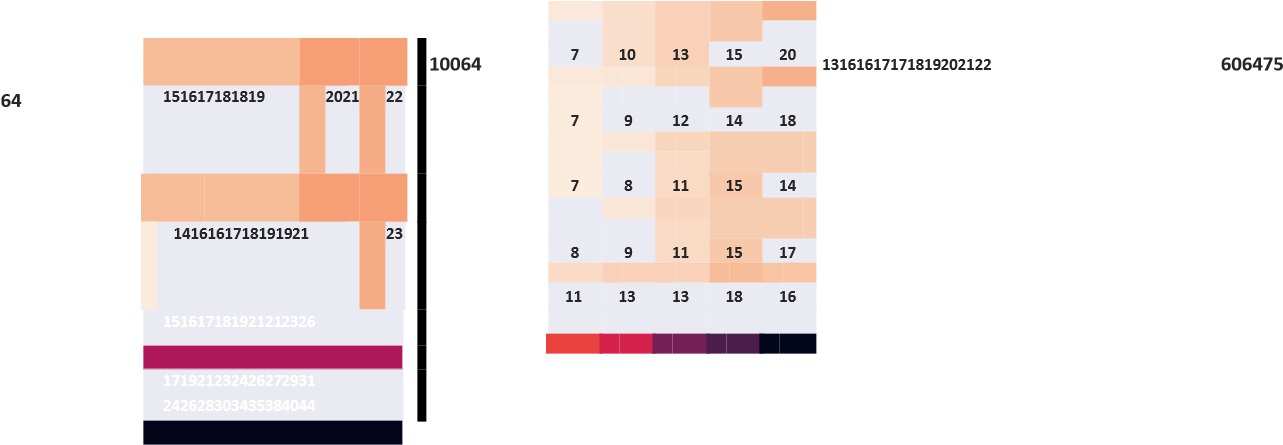
**0.8 4096 6 6 9 101415182425504096 9 15 24 36 48 63 761131302114096 11 12 12 16**





|  |  |  |
| --- | --- | --- |
| **1 2 3 4 5 6 7 8 910**  隠れた層の深さ  (b) MNIST, リニア FC | **1 2 3 4 5 6 7 8 910** 隠れた層の深さ  (c) EMNIST, FC | **1 2 3 4 5 6 7 8 910** 隠れた層の深さ  (d) ジゼット、FC |

|  |  |  |
| --- | --- | --- |
| **512512 102440**  **1024 1530 2048**  **2048202048 2720 4096 41 48**  **68 77 9415** |  |  |
| **4096 82 81 83 90 82 83 88 961041144096 45394245515254596468** | **1 2 3 4** | **51 2 3 4 5** |

**3232 17182020212323242728 32 810 13 15 1590**

**24** **21** **22** **42** **53**

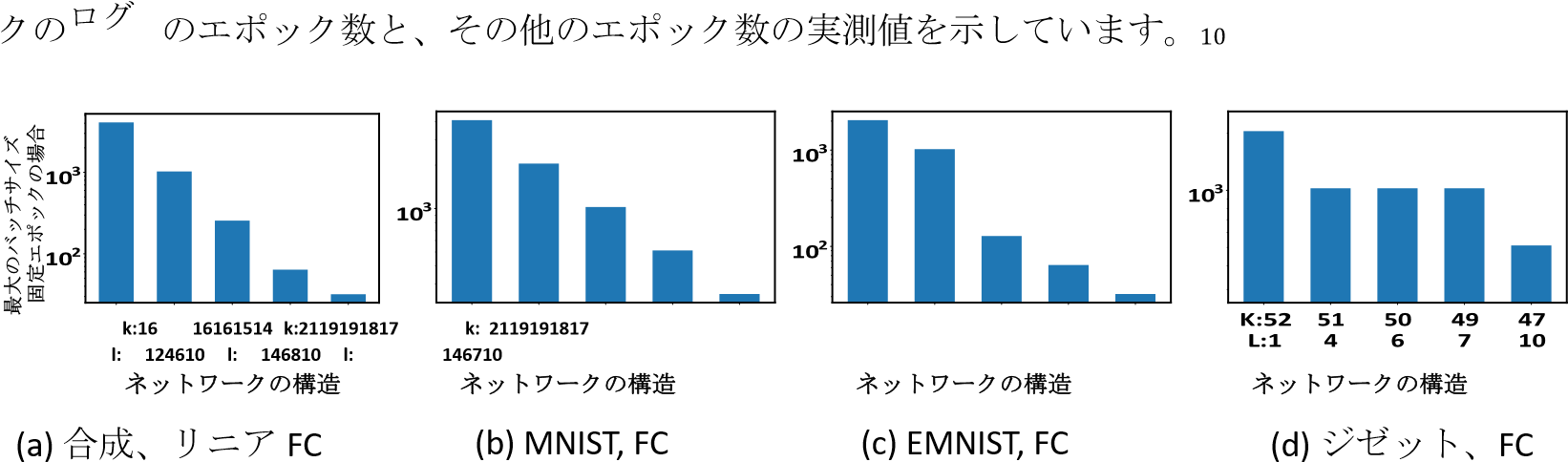
**12880128 1450 12825660**

**25660256 121240 10245124530**

**1 2 3 4 5 6 7 8 910** ブロック数ブロック数各チャンクの隠れた層の深さ各チャンクの隠れた層の深さ

隠れた層の深さ(e) MNIST,

LeNet (f) Cifar10, ResNet18, FC (g) Cifar10, ResNet18, Res (h) Cifar10, ResNet34, Res 図 4：表 1 で定義した損失/精度に収束するのに必要なエポック数のヒートマップ。(a)ではエポッ



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | | | |
|  |  |  |  | | | | | | | | |
|  |  |  | | | | | | |
|  |  |  |  |  |  |  |



造

(e) LeNet 上の MNIST (f) Cifar10, ResNet18, FC (g) Cifar10, ResNet18, Res (h) Cifar10, ResNet34, Res 図 5：一定のエポック数で収束させるための最大のバッチサイズ。

大規模な分散学習のシナリオでは、バッチサイズを大きくすると、通信量が減るため、より多くのスピードアップが可能になるからです。最後に、最大のバッチサイズは、ネットワークやデータセットによって異なることに留意する必要があります。これは、勾配の多様性がデータにもモデルにも依存するためである。

# おわりに

本論文では、ニューラルネットワークの構造が、大規模なバッチトレーニングのパフォーマンスにどのように影響するかを研究しています。勾配の多様性の観点から、ネットワークの深さと幅が、大規模なバッチ学習に適しているかどうかを定量的に示した。広範な実験結果と理論的な分析により、大規模なクラスのニューラルネットワークでは、幅を大きくすると勾配の多様性が大きくなり、分散計算に常に有益な大規模なバッチトレーニングが可能になることが示された。

将来的には、畳み込みフィルターや残差ブロックなどの特定の構造が、勾配の多様性にどのように影響するかを調べる予定です。実用的な観点からは、ネットワークのアーキテクチャを考慮することが、分散環境での高速化のために重要であると考えています。そのため、大規模なバッチトレーニングを可能にし、通信のボトルネックを最小限に抑えるために、ネットワークをどのように微調整するかを検討する予定です。もう一つの方向性は、汎化誤差がどのように影響するかを定量的に研究することです。