Layer Normalization

- Jimmy Lei Ba (University of Toronto)
- Jamie Ryan Kiros (University of Toronto)
- Geoffrey E. Hinton (University of Toronto and Google Inc.)

Abstract

最先端の深いニューラルネットワークのトレーニングは、計算コストがかかります。トレーニング時間を短縮する1つの方法は、ニューロンの活動を正常化することである。最近紹介されたバッチ正規化と呼ばれる手法では、トレーニングケースのミニバッチでニューロンに入力された入力の分布を使用して平均と分散を計算し、各トレーニングケースでそのニューロンへの入力を正規化します。これにより、フィードフォワードニューラルネットワークのトレーニング時間が大幅に短縮されます。しかし、バッチ正規化の効果はミニバッチサイズに依存し、リカレントニューラルネットワークにどのように適用するかは明らかではない。この論文では、バッチ正規化をレイヤー正規化に転置します。これは、単一のトレーニングケースでレイヤー内のすべての入力からニューロンへの正規化に使用される平均と分散を計算します。バッチ正規化と同様に、各ニューロンには、正規化後で非線形性の前に適用される独自の適応バイアスおよびゲインも与えます。バッチの正規化とは異なり、レイヤの正規化はトレーニングとテスト時に全く同じ計算を実行します。各時間ステップで正規化統計を別々に計算することによってリカレントニューラルネットワークに適用することも簡単である。レイヤの正規化は、再帰的ネットワークにおける隠れた状態のダイナミクスを安定化する上で非常に有効です。経験的には、層の正規化が以前に公表された技法と比較してトレーニング時間を大幅に短縮できることを示す。

1 Introduction

確率的勾配降下のいくつかのバージョンで訓練された深い神経ネットワークは、コンピュータビジョン（Krizhevsky et al。、2012）および音声処理[Hinton et al。、2012]における様々な教師付き学習タスクに対する従来のアプローチを大幅に上回ることが示されている。しかし、最先端の深いニューラルネットワークは、しばしば、何日もの訓練を必要とする。異なるマシン上のトレーニングケースの異なるサブセットのグラジエントを計算するか、またはニューラルネットワーク自体を多くのマシンに分割することで学習をスピードアップすることは可能ですが（Dean et al。、2012）、これは多くの通信を必要とする可能性があります複雑なソフトウェアです。また、並列化の度合いが高まるにつれて、リターンが急速に減少する傾向があります。直交アプローチは、学習を容易にするためにニューラルネットの順方向パスで実行される計算を修正することである。近年、ディープニューラルネットワークに正規化段階を追加することで、トレーニング時間を短縮するバッチ正規化[Ioffe and Szegedy、2015]が提案されている。正規化は、訓練データにわたるその平均値および標準偏差を用いて、各合計入力を標準化する。バッチ正規化を使用してトレーニングされたフィードフォワードニューラルネットワークは、単純なSGDであってもより早く収束します。訓練時間の改善に加えて、バッチ統計からの確率は、訓練中に正規化因子として役立つ。

単純化されているにもかかわらず、バッチの正規化では、合計入力統計の実行平均が必要です。固定深度のフィードフォワードネットワークでは、各隠れた層ごとに統計を個別に保存するのは簡単です。しかし、反復ニューラルネットワーク（RNN）における再帰ニューロンへの入力の合計は、シーケンスの長さによって変化することが多いため、RNNにバッチ正規化を適用すると、異なる時間ステップに対して異なる統計が必要になるようです。さらに、バッチの正規化は、オンライン学習タスク、またはミニバッチを小さくしなければならない非常に大きな分散モデルには適用できません。

本稿では、様々なニューラルネットワークモデルの学習速度を向上させるための単純正規化法である層正規化を紹介する。バッチ正規化とは異なり、提案された方法は、隠れ層内のニューロンへの合計入力から正規化統計を直接推定し、正規化はトレーニングケース間に新たな依存性を導入しない。レイヤの正規化がRNNでうまく機能し、トレーニング時間といくつかの既存のRNNモデルの汎化性能の両方が向上することを示します。

2 Background

フィードフォワードニューラルネットワークは、入力パターンxから出力ベクトルyへの非線形マッピングである。深いフィードフォワード、ニューラルネットワークにおけるl番目の隠れ層を考え、その層のニューロンへの入力の合計をベクトル表現するとする。合計された入力は、以下のように与えられた重み行列W lとボトムアップ入力hlとの線形射影を介して計算される。

$$a_i^l = {w_i^l}^{\mathrm{T}} h^l \quad h_i^{l+1} = f(a_i^l + b_i^l) \qquad (1)$$

ここで、f（・）は要素ごとの非線形関数であり、wilはi番目の隠れユニットへの入力重みであり、bliはスカラバイアスパラメータである。ニューラルネットワークのパラメータは、グラジエントベースの最適化アルゴリズムを使用して学習され、グラジエントはバックプロパゲーションによって計算されます。

深い学習の課題の1つは、1つの層の重みに関する勾配が、前の層のニューロンの出力に大きく依存することであり、特に、これらの出力が高度に相関した方法で変化する場合である。そのような望ましくない「共変量シフト（covariate shift）」を減らすために、バッチ標準化[Ioffe and Szegedy、2015]が提案された。この方法は、トレーニングケースにわたって各隠れユニットへの合計入力を正規化する。具体的には、i番目のレイヤーで合計されたi番目の入力に対して、バッチ正規化方法は、データの分布の下でその差異に従って合計入力を再スケーリングする

$$\bar{a}_i^l = \frac{g_i^l}{\sigma_i^l}(a_i^l - \mu_i^l) \quad \mu_i^l = \mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x})}[a_i^l] \quad \sigma_i^l = \sqrt{\mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x})}\left[(a_i^l - \mu_i^l)^2\right]} \quad (2)$$

ここで,- 1は、l番目の層のi番目の隠れユニットへの正規化された合計入力であり、giは、非線形活性化関数の前の正規化された賦活をスケーリングする利得パラメータである。期待されるのは、トレーニングデータの配布全体の下にあることに注意してください。一般に、式（5）の期待値を計算することは実用的ではない。 （2）は、現在の重みのセットでトレーニングデータセット全体を順方向に通過する必要があるためです。代わりに、μおよびσは、現在のミニバッチからの経験的サンプルを使用して推定される。これは、ミニバッチのサイズに制約を課し、リカレントニューラルネットワークに適用することは困難です。

3 Layer normalization

バッチ正規化の欠点を克服するために設計されたレイヤ正規化方法について検討します。

1つのレイヤーの出力の変化は、特に多くの出力が変わることがあるReLUユニットでは、次のレイヤーへの合計入力の相関の高い変化を引き起こす傾向があることに注意してください。これは、各層内の合計入力の平均と分散を固定することによって、「共変量シフト」問題を軽減できることを示唆しています。したがって、我々は、次のように、同じ層内のすべての隠れユニットに対する層正規化統計量を計算する。

$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l \quad \sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_i^l - \mu^l)^2} \tag{3}$$

Hはレイヤ内の隠れユニットの数を表す。式（2）と式（3）は、レイヤ下の正規化では、レイヤ内の全ての隠れユニットが同じ正規化項μとσを共有するが、異なるトレーニングケースは異なる正規化項を有する。バッチ正規化とは異なり、レイヤ正規化はミニバッチのサイズに制約を課すものではなく、バッチサイズ1の純粋なオンラインレジームで使用できます。

3.1 Layer normalized recurrent neural networks

最近のシーケンス・シーケンス・シーケンス[Sutskever et al。、2014]は、自然言語処理における逐次予測問題を解決するためにコンパクトなリカレント・ニューラル・ネットワークを利用している。 NLPタスク間では、異なる訓練の場合に異なる文章長を持つことが一般的です。これは、同じ重みがすべての時間ステップで使用されるため、RNNで対処するのは簡単です。しかし、バッチ正規化をRNNに明白な方法で適用する場合は、シーケンス内の各タイムステップごとに個別の統計を計算して格納する必要があります。これは、テストシーケンスがいずれのトレーニングシーケンスよりも長い場合には問題となる。レイヤ正規化は、正規化項が現在のタイムステップでレイヤへの入力の合計にのみ依存するため、このような問題はありません。また、すべてのタイムステップで共有されるゲインとバイアスのパラメータは1セットしかありません。

標準RNNでは、反復層の合計入力は、at = Whhht-1 + Wxhxtとして計算される隠れ状態ht-1の現在の入力xtおよび前のベクトルから計算される。層正規化反復層は、式（5）と同様の余分な正規化項を使用して、その活性化を再センタリングし、再スケーリングする。 （3）：

$$\mathbf{h}^t = f\left[\frac{\mathbf{g}}{\sigma^t}\odot(\mathbf{a}^t - \mu^t) + \mathbf{b}\right] \quad \mu^t = \frac{1}{H}\sum_{i=1}^{H} a_i^t \quad \sigma^t = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_i^t - \mu^t)^2} \tag{4}$$

ここで、Whhは隠れた隠れた重みに反復され、Wxhは隠れた重みの下の上の入力です。 ⊙は2つのベクトルの要素ごとの乗算です。 bおよびgは、htと同じ次元のバイアスおよび利得パラメータとして定義される。

標準的なRNNでは、再帰ユニットへの合計入力の平均的な大きさが、時間ステップごとに増加または減少する傾向があり、爆発または勾配が崩壊する。レイヤ正規化されたRNNでは、正規化の項がレイヤに合計された入力のすべてを再スケーリングすることを不変にし、より安定した隠れから隠れたダイナミクスをもたらします。

4 Related work

バッチ正規化は以前はリカレントニューラルネットワークに拡張されてきた[Laurentら、2015、Amodeiら、2015、Cooijmansら、2016]。前回の研究[Cooijmans et al。、2016]は、各時間ステップごとに独立した正規化統計を保持することによって、反復バッチ正規化の最良の性能が得られることを示唆している。著者らは、反復バッチ正規化レイヤーでゲインパラメータを0.1に初期化すると、モデルの最終的な性能に大きな違いがあることが示されています。私たちの研究は、体重の正常化[Salimans and Kingma、2016]にも関連しています。重みの正規化では、分散の代わりに、入力ウェイトのL2ノルムを使用して、ニューロンへの合計入力を正規化する。予想される統計値を使用して重み正規化またはバッチ正規化を適用することは、元のフィードフォワードニューラルネットワークの異なるパラメータ化を行うことと等価です。 ReLUネットワークの再パラメータ化は、パス正規化SGD [Neyshabur et al。、2015]で研究されました。しかしながら、我々の提案する層正規化法は、元のニューラルネットワークの再パラメータ化ではない。このように、層正規化モデルは、他の方法とは異なる不変性を持っており、次のセクションで検討します。

5 Analysis

この節では、異なる正規化スキームの不変性特性を調べる。

5.1 Invariance under weights and data transformations

提案された層正規化は、バッチ正規化および重量正規化に関連する。しかし、これらの方法は、2つのスカラーμとσを介して、入力された入力aiをニューロンに正規化するように要約することができます。また、正規化後の各ニューロンの適応バイアスbとゲインgを学習します。

$$h_i = f(\frac{g_i}{\sigma_i}(a_i - \mu_i) + b_i) \qquad (5)$$

層の正規化とバッチの正規化では、μとσは式重み正規化において、μは0であり、σ=‖w‖2である。

	Weight matrix re-scaling	Weight matrix re-centering	Weight vector re-scaling	Dataset re-scaling	Dataset re-centering	Single training case re-scaling
Batch norm	Invariant	No	Invariant	Invariant	Invariant	No
Weight norm	Invariant	No	Invariant	No	No	No
Layer norm	Invariant	Invariant	No	Invariant	No	Invariant

表1：正規化法による不変性

表1は、3つの正規化方法に対する以下の不変性結果を強調している。

重みの再スケーリングと再センタリング：まず、バッチの正規化と重みの正規化では、単一のニューロンの入力ウェイトw iに対する再スケーリングは、ニューロンへの正規化された合計入力に影響を与えないことに注意してください。正確には、バッチおよび重みの正規化の下で、重みベクトルがδによってスケーリングされる場合、2つのスカラーμおよびσもδだけスケーリングされる。正規化された合計入力は、スケーリングの前後で同じままです。したがって、バッチおよび重量の正規化は、重みの再スケーリングに対して不変である。一方、レイヤ正規化は、単一の重みベクトルの個々のスケーリングに対して不変ではない。代わりに、層の正規化は、重み行列全体のスケーリングに対して不変であり、重み行列におけるすべての入力重みへのシフトに不変である。重み行列WおよびW 'がスケーリングファクタδだけ異なる2組のモデルパラメータθ、θ'が存在し、W '内のすべての入力重みも定数ベクトルγだけシフトされる、すなわちW' =δW+ 1γ⊤。レイヤーの正規化の下では、2つのモデルが同じ出力を効果的に計算します。

$$\begin{aligned} \mathbf{h}' &= f(\frac{\mathbf{g}}{\sigma'}(W'\mathbf{x} - \mu') + \mathbf{b}) = f(\frac{\mathbf{g}}{\sigma'}((\delta W + \mathbf{1}\gamma^{\mathrm{T}})\mathbf{x} - \mu') + \mathbf{b}) \\ &= f(\frac{\mathbf{g}}{\sigma}(W\mathbf{x} - \mu) + \mathbf{b}) = \mathbf{h} \end{aligned} \quad (6)$$

正規化が重みの前に入力にのみ適用される場合、モデルは重みの再スケーリングと再センタリングに不変ではないことに注意してください。

データの再スケーリングと再センタリング：すべての正規化方法は、ニューロンの合計入力が変化のもとで一定であることを検証することによってデータセットを再スケーリングすることに不変であることを示すことができる。さらに、層正規化は、個々の訓練事例の再スケーリングに不変である。正規化スカラμおよびσは式（3）は、現在の入力データのみに依存する。 x 'は、xをδで再スケーリングした新しいデータ点とする。それから、

$$h_i' = f(\frac{g_i}{\sigma'}(w_i^{\mathrm{T}}\mathbf{x}' - \mu') + b_i) = f(\frac{g_i}{\delta\sigma}(\delta w_i^{\mathrm{T}}\mathbf{x} - \delta\mu) + b_i) = h_i \quad (7)$$

個々のデータ点を再スケーリングしても、層の正規化の下でモデルの予測は変化しないことは容易にわかります。層正規化における重み行列の再センタリングと同様に、バッチ正規化がデータセットの再センタリングに対して不変であることを示すこともできる。

5.2 Geometry of parameter space during learning

我々は、パラメータの再センタリングおよび再スケーリングのもとでのモデルの予測の不変性を調べた。しかし、モデルが同じ基本的な機能を表現していても、ラーニングは異なるパラメータ化の下で非常に異なる動作をすることができます。このセクションでは、ジオメトリとパラメータ空間の多様体を通じた学習行動を分析します。我々は、正規化スカラσが暗黙的に学習率を低下させ、学習をより安定にすることができることを示す。

5.2.1 Riemannian metric

統計的モデルにおける学習可能パラメータは、モデルの全ての可能な入出力関係からなる滑らかな多様体を形成する。出力が確率分布であるモデルの場合、このマニホールド上の2点の分離を測定する自然な方法は、モデル出力分布間のKullback-Leiblerの相違である。 KLダイバージェンスメトリックの下では、パラメータ空間はリーマン多様体である。

リーマン多様体の曲率はリーマンメトリックによって完全に捕捉され、その二次形式はds2と表される。これは、パラメータ空間内のある点における接平面空間の微小距離です。直観的には、パラメータ空間からの接線方向に沿ったモデル出力の変化を測定します。 KLに基づくリーマンメトリックは以前に研究された[Amari、1998]、フィッシャー情報行列を使用して2次テイラー展開の下でよく近似されることが示された：

$$ds^2 = \mathrm{D}_{\mathrm{KL}}\left[\mathrm{P}(\mathbf{y} \mid \mathbf{x};\, \theta) \,\|\, \mathrm{P}(\mathbf{y} \mid \mathbf{x};\, \theta + \delta)\right] \approx \frac{1}{2}\delta^{\mathrm{T}}\mathbf{F}(\theta)\delta \quad (8)$$

$$F(\theta) = \mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x}), y\sim P(y|\mathbf{x})}\left[\frac{\partial \log P(y \mid \mathbf{x};\, \theta)}{\partial \theta}\frac{\partial \log P(y \mid \mathbf{x};\, \theta)^{\mathrm{T}}}{\partial \theta}\right] \quad (9)$$

ここで、δはパラメータに対する小さな変化である。上のリーマンメトリックは、パラメータ空間の幾何学的なビューを示します。リーマンメトリックの以下の分析は、いくつかの洞察を提供する 正規化法がニューラルネットワークのトレーニングにどのように役立つかについて説明します。

5.2.2 The geometry of normalized generalized linear models

我々は、一般化された線形モデルに我々の幾何学的解析を集中させる。以下の分析の結果は、フィッシャー情報行列に対してブロック対角近似を持つディープニューラルネットワークを理解するために簡単に適用できます。フィッシャー情報行列は、各ブロックが単一のニューロンのパラメータに対応します。

一般化線形モデル（GLM）は、重みベクトルwおよびバイアススカラーbを用いて指数族からの出力分布をパラメータ化するものとみなすことができる。前のセクションと一致するように、GLMの対数尤度は、以下のように合計入力aを使用して書き込むことができます。

$$\log P(y \mid \mathbf{x};\, w, b) = \frac{(a+b)y - \eta(a+b)}{\phi} + c(y, \phi) \qquad (10)$$

$$\mathbb{E}[y \mid \mathbf{x}] = f(a+b) = f(w^{\mathrm{T}}\mathbf{x} + b),\ \mathbb{V}\mathrm{ar}[y \mid \mathbf{x}] = \phi f'(a+b) \quad (11)$$

ここで、f（・）はニューラルネットワークの非線形性のアナログである伝達関数、f '（・）は伝達関数の導関数、η（・）は実数値関数、c（・）ログの分割機能です。 φは出力分散をスケーリングする定数です。 H個の独立したGLMを用いてH次元の出力ベクトルy = [y1、y2、...、yH]をモデル化し、logP（y | x; W、b）= Hi = 1 logP（yi | x; wi、bi）。 Wを個々のGLMの重みベクトルとなる重み行列とし、bを長さHのバイアスベクトルとし、vec（・）をクロネッカーベクトル演算子とする。そのパラメータθ= [w1⊤、b1、...、wH⊤、bH]⊤= vec（[W、b]⊤）に対する多次元GLMのフィッシャー情報行列は、予測されたクロネッカー積のデータ特徴と出力共分散行列：

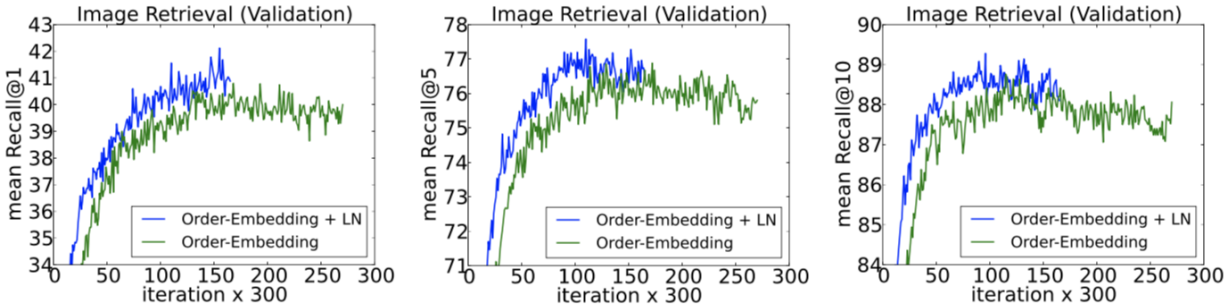
$$F(\theta) = \mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x})}\left[\frac{\mathrm{Cov}[\mathbf{y} \mid \mathbf{x}]}{\phi^2} \otimes \begin{bmatrix} \mathbf{x}\mathbf{x}^{\mathrm{T}} & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & 1 \end{bmatrix}\right] \qquad (12)$$

正規化されたGLMは、正規化法をμとσによって元のモデルの入力aに加算することによって得られる。一般性を失うことなく、正規化された多次元GLMの下でフィッシャー情報行列としてF denを、付加的なゲインパラメータθ= vec（[W、b、g]⊤）で表す。

$$\bar{F}(\theta) = \begin{bmatrix} \bar{F}_{11} & \dots & \bar{F}_{1H} \\ \vdots & \ddots & \vdots \\ \bar{F}_{H1} & \dots & \bar{F}_{HH} \end{bmatrix},\ \bar{F}_{ij} = \mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x})}\left[\frac{\mathrm{Cov}[\mathbf{y_i}, \mathbf{y_j} \mid \mathbf{x}]}{\phi^2} \begin{bmatrix} \frac{g_i g_j}{\sigma_i \sigma j}\chi_i \chi_j^{\mathrm{T}} & \chi_i \frac{g_i}{\sigma_i} & \chi_i \frac{g_i(a_j - \mu_j)}{\sigma_i \sigma_j} \\ \chi_j^{\mathrm{T}} \frac{g_j}{\sigma_j} & 1 & \frac{a_j - \mu_j}{\sigma_j} \\ \chi_j^{\mathrm{T}} \frac{g_j(a_i - \mu_i)}{\sigma_i \sigma_j} & \frac{a_i - \mu_i}{\sigma_i} & \frac{(a_i - \mu_i)(a_j - \mu_j)}{\sigma_i \sigma_j} \end{bmatrix}\right] \quad (13)$$

$$\chi_i = \mathbf{x} - \frac{\partial \mu_i}{\partial w_i} - \frac{a_i - \mu_i}{\sigma_i}\frac{\partial \sigma_i}{\partial w_i} \qquad (14)$$

重みベクトルの成長による暗黙の学習率の低下：標準的なGLMと比較すると、重みベクトルwi方向に沿ったブロックF ijijは、ゲインパラメータおよび正規化スカラーσiを含む。重みベクトルwiのノルムが2倍大きくなると、モデルの出力が変わっていなくても、フィッシャー情報マトリックスは異なるでしょう。 wi方向に沿った曲率は、σiも2の2倍になるので、1の係数で変化します 大きい。その結果、正規化されたモデルにおける同じパラメータの更新に関して、重みベクトルのノルムは、重みベクトルの学習速度を効果的に制御する。学習中、大きなノルムを持つウェイトベクトルの向きを変更するのは難しいです。したがって、正規化方法は、重みベクトルに暗黙の「早期停止」効果をもたらし、収束に向けた学習を安定させるのに役立つ。



(a) Recall@1 (b) Recall@5 (c) Recall@10

図1：層の正規化の有無にかかわらずオーダ埋め込みを使用したリコールカーブ。

	MSCOCO							
	Caption Retrieval				Image Retrieval			
Model	**R@1**	**R@5**	**R@10**	**Mean *r***	**R@1**	**R@5**	**R@10**	**Mean *r***
Sym [Vendrov et al., 2016]	45.4		88.7	5.8	36.3		85.8	9.0
OE [Vendrov et al., 2016]	46.7		88.9	5.7	37.9		85.9	8.1
OE (ours)	46.6	79.3	89.1	5.2	37.8	73.6	85.7	7.9
OE + LN	**48.5**	**80.6**	**89.8**	**5.1**	**38.9**	**74.3**	**86.3**	**7.6**

表2：キャプションと画像検索のための5つのテスト分割にまたがる平均結果。 R @ Kはリコール@K（ハイが良い）です。平均rは平均ランク（低いものは良い）です。 Symは対称ベースラインに対応し、OEは順序埋め込みを示す。

入力ウェイトの大きさの学習：正規化モデルでは、入力ウェイトの大きさがゲインパラメータによって明示的にパラメータ化されます。正規化されたGLMのゲインパラメータの更新と学習中の元のパラメータ化のもとでの等価ウェイトの大きさの更新との間で、モデル出力がどのように変化するかを比較する。 F inのゲインパラメータに沿った方向は、入力ウェイトの大きさのジオメトリを取得します。標準GLMの入力ウェイトの大きさに沿ったリーマンメトリックは、その入力のノルムによってスケーリングされるが、バッチ正規化モデルおよびレイヤ正規化モデルのゲインパラメータの学習は、予測誤差の大きさのみに依存することを示す。したがって、正規化されたモデルにおける入力ウェイトの大きさの学習は、標準モデルよりも入力およびそのパラメータのスケーリングに対してより堅牢です。詳細な派生については、付録を参照してください。

6 Experimental results

画像文ランキング、質問応答、文脈言語モデリング、生成モデリング、手書きシーケンス生成、MNIST分類の6つのタスクで、レイヤー正規化を用いた実験を繰り返します。特に明記しない限り、層正規化のデフォルトの初期化は、実験で適応ゲインを1に、バイアスを0に設定することです。

6.1 Order embeddings of images and language

この実験では、Vendrov et al。の最近提案された順序埋め込みモデルに層正規化を適用する。 [2016]は画像と文の結合空間を学習する。 Vendrov et al。と同じ実験プロトコールに従う。 Theano [Team et al。、2016]を利用する層正規化1を組み込むために公開されているコードを修正する。 Microsoft COCOデータセット[Lin et al。、2014]の画像とセンテンスは共通のベクトル空間に埋め込まれており、GRU [Cho et al。、2014]は文章をエンコードし、事前訓練されたVGG ConvNet [Simonyan and Zisserman、2015]（10作物）は、画像をエンコードするために使用されます。順序埋め込みモデルは、画像と文を2レベル部分順序として表し、Kiros et al。で使用されているコサイン類似度スコアリング関数を置き換えます。 [2014]は非対称なものである。

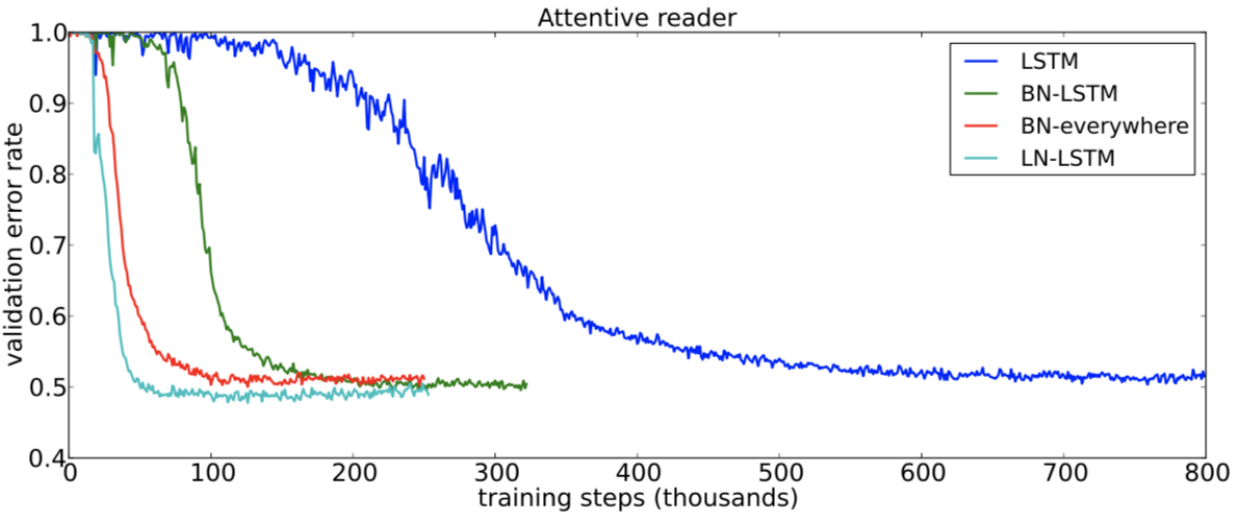


図2：注意深いリーダーモデルの検証曲線BNの結果は[Cooijmans et al。、2016]から取られている。

我々は2つのモデルを訓練した：基線順序埋め込みモデルと、GRUに適用された層正規化を伴う同じモデルとを訓練した。 300回の反復ごとに、保留された検証セットのRecall @ K（R @ K）値を計算し、R @ Kが改善するたびにモデルを保存します。最高のパフォーマンスを発揮するモデルは、5つの別々のテストセットで評価され、それぞれに1000のイメージと5,000のキャプションが含まれており、平均結果が報告されます。どちらのモデルもAdam [Kingma and Ba、2014]を使用して同じ初期ハイパーパラメータを使用し、両方のモデルはVendrov et al。 [2016]。レイヤ正規化がGRUにどのように適用されるかについては、付録を参照してください。

図1は、層の正規化の有無にかかわらず、モデルの検証曲線を示しています。画像検索タスクのためにR1、R5及びR10をプロットする。レイヤーの正規化は、すべてのメトリックにわたって反復ごとのスピードアップを提供し、ベースラインモデルにかかる時間の60％で最良の検証モデルに収束します。表2では、テストセットの結果が報告されており、レイヤーの正規化によって元のモデルよりも一般化が改善されていることがわかります。我々が報告する結果は、RNN埋め込みモデルの最先端技術であり、Wangらの構造保存モデルのみである。 [2016]この作業についてより良い結果を報告する。しかし、それらは異なる条件（5以上の平均の代わりに1つのテストセット）で評価されるため、直接比較できません。

6.2 Teaching machines to read and comprehend

最近提案された反復バッチ正規化[Cooijmans et al。、2016]に対する層正規化を比較するために、我々はHermannらによって導入されたCNNコーパス上の一方向の注意深いリーダーモデルを訓練する。 [2015]。これは質問に答える作業で、通路に関する質問の説明に空白を記入する必要があります。データは匿名化され、エンティティには無作為化されたトークンが与えられ、訓練と評価中に一貫して置換される縮退解を防ぎます。我々はCooijmansらと同じ実験プロトコルに従う。 [2016]、Theano [Team et al。、2016]を使用する層の正規化2を組み込むために公開コードを変更する。我々は、Cooijmansらによって使用される前処理されたデータセットを得た。 Hermannらの最初の実験とは異なる[2016]。 [2015]では、各節は4文に制限されている。 Cooijmans et al。 [2016]では、BNがLSTMにのみ適用され、もう1つはモデル全体にBNが適用される反復バッチ正規化の2つの変種が使用されています。実験では、LSTM内のレイヤ正規化のみを適用します。

この実験の結果を図2に示します。レイヤの正規化は、列車の速度が速いだけでなく、ベースラインとBNの両方のバリエーションよりも優れた検証結果に収束することがわかります。 Cooijmans et al。 [2016]、BNのスケールパラメータは慎重に選択しなければならず、実験では0.1に設定する必要があると主張されている。我々は、1.0と0.1の両方のスケール初期化について層正規化を実験し、前者のモデルが著しく良好に実行されることを見出した。これは、層の正規化が、再発BNと同じ方法で初期スケールに対して敏感ではないことを実証する。 3

6.3 Skip-throught vectors

Skip-thoughts [Kiros et al。、2015]は、教師なし分散文の学習のためのスキップグラムモデル[Mikolov et al。、2013]の一般化である。連続したテキストが与えられると、文はエンコーダRNNで符号化され、復号器RNNは周囲の文を予測するために使用される。 Kiros et al。 [2015]は、このモデルが微調整されずにいくつかのタスクでうまく機能する一般的な文表現を生成できることを示した。しかし、このモデルをトレーニングするには時間がかかり、意味のある結果を出すために数日間のトレーニングが必要です。

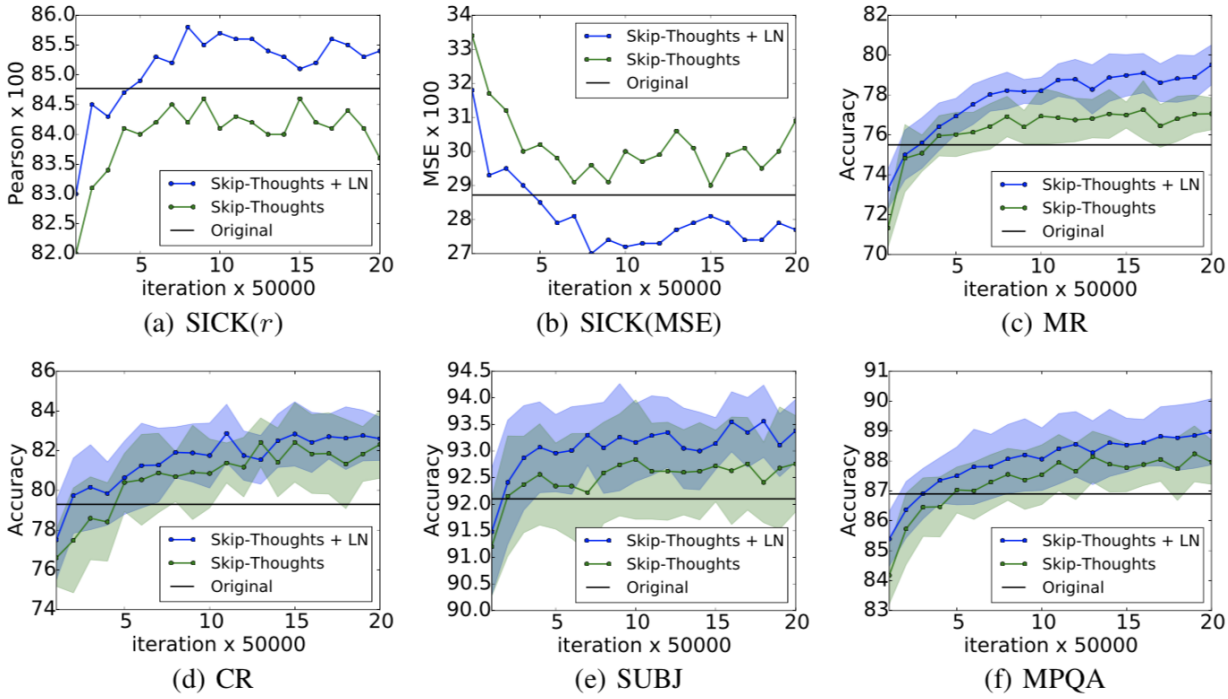


図3：訓練反復の関数として下流のタスクで層正規化を行った場合としない場合のskip-thoughtベクトルのパフォーマンス元の行は[Kiros et al。、2015]で報告された結果である。誤差のあるプロットは10倍のクロスバリデーションを使用します。最高の色で見られる。

Method	SICK(r)	SICK(ρ)	SICK(MSE)	MR	CR	SUBJ	MPQA
Original [Kiros et al., 2015]	0.848	0.778	0.287	75.5	79.3	92.1	86.9
Ours	0.842	0.767	0.298	77.3	81.8	92.6	87.9
Ours + LN	0.854	0.785	0.277	**79.5**	82.6	93.4	89.0
Ours + LN †	**0.858**	**0.788**	**0.270**	79.4	**83.1**	**93.7**	**89.3**

表3：スキップ・アイデアの結果最初の2つの評価列はPearsonとSpearmanの相関を示し、3つ目は平均二乗誤差で、残りは分類精度を示します。 MSEを除くすべての評価で高い方が良い。我々のモデルは、1ヶ月間訓練された（†）を除いて1M反復のために訓練された（約1.7M反復）

この実験では、層の正規化がトレーニングをスピードアップできる効果を判断します。公開されているKiros et al。 4、BookCorpusデータセット[Zhu et al。、2015]上で2つのモデルを訓練する.1つは層正規化なしで、もう1つは層正規化なしである。これらの実験は、Theano [Team et al。、2016]を用いて行われる。我々は、Kiros et al。 [2015]、同じハイパーパラメータを有する2400次元文エンコーダを訓練する。使用される状態のサイズが与えられると、層の正規化は反復ごとの更新を伴わない場合よりも遅くなると考えられる。しかし、提供されたCNMeM 5が使用されていることがわかりました.2つのモデルの間に大きな違いはありませんでした。セマンティック関連性（SICK）[Marelli et al。、2014]、映画レビューセンチメント（MR）[Pang and Lee、2005]、顧客製品レビュー（CR） ）[Hu and Liu、2004]、主観/客観的分類（SUBJ）[Pang and Lee、2004]、および意見極性（MPQA）[Wiebe et al。すべてのタスクで各チェックポイントの両方のモデルのパフォーマンスをプロットし、LNでパフォーマンス・レートを改善できるかどうかを判断します。

実験結果は図3に示されています。レイヤ正規化を適用すると、表3に示すように、1M反復が実行された後に、ベースラインを上回るスピードアップとより良い最終結果の両方が得られることがわかります。合計で1か月間実行されるため、1つのタスクを除くすべてのタスクでパフォーマンスが向上します。オリジナルの報告された結果と私たちのパフォーマンスの違いは、元のモデルが行っているデコーダのタイムステップごとに公開されているコードが条件付けられていないという事実に起因すると考えられます。

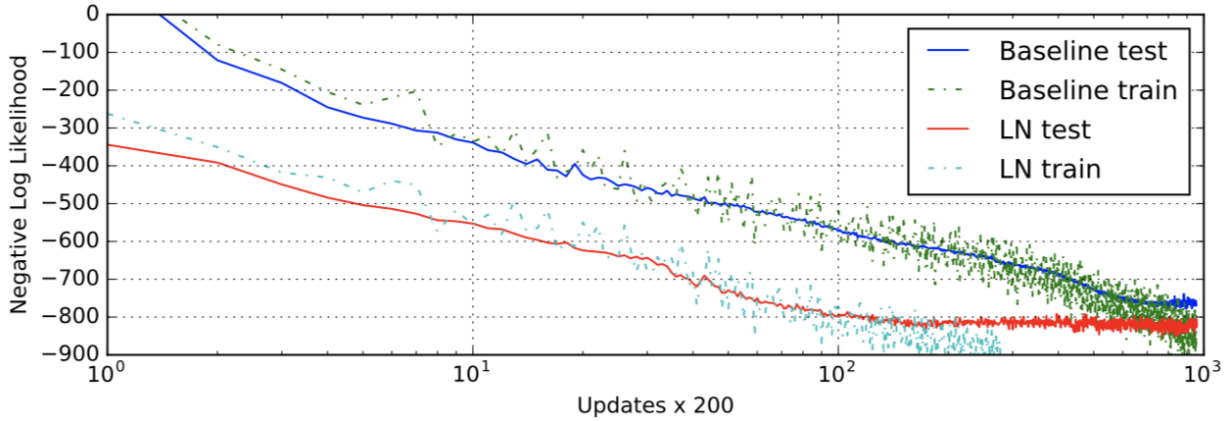


図5：手書きのシーケンス生成モデルネガティブ対数尤度、層正規化有り無し。モデルは、ミニバッチサイズ8および配列長500で訓練されている

6.4 Modeling binarized MNIST using DRAW

我々はまた、MNISTデータセットの生成モデルを実験した。ディープ・リカレント・アテンション・ライター（DRAW）[Gregor et al。、2015]は、以前はMNISTのディジットの分布をモデリングする際の最先端のパフォーマンスを達成しています。このモデルでは、微分アテンション機構とリカレントニューラルネットワークを用いて画像の断片を順次生成する。我々は、64個の一瞥と256個のLSTM隠れた単位を用いて、DRAWモデルに対する層正規化の効果を評価する。このモデルはAdam [Kingma and Ba、2014]オプティマイザのデフォルト設定と128のミニバッチサイズで訓練されています。バイナリ化されたMNISTの以前の出版物では、さまざまなトレーニングプロトコルを使用してデータセットを生成しています。この実験では、LarochelleとMurray [2011]の固定二値化を使用しました。データセットは、50,000トレーニング、10,000検証、10,000テスト・イメージに分割されています。

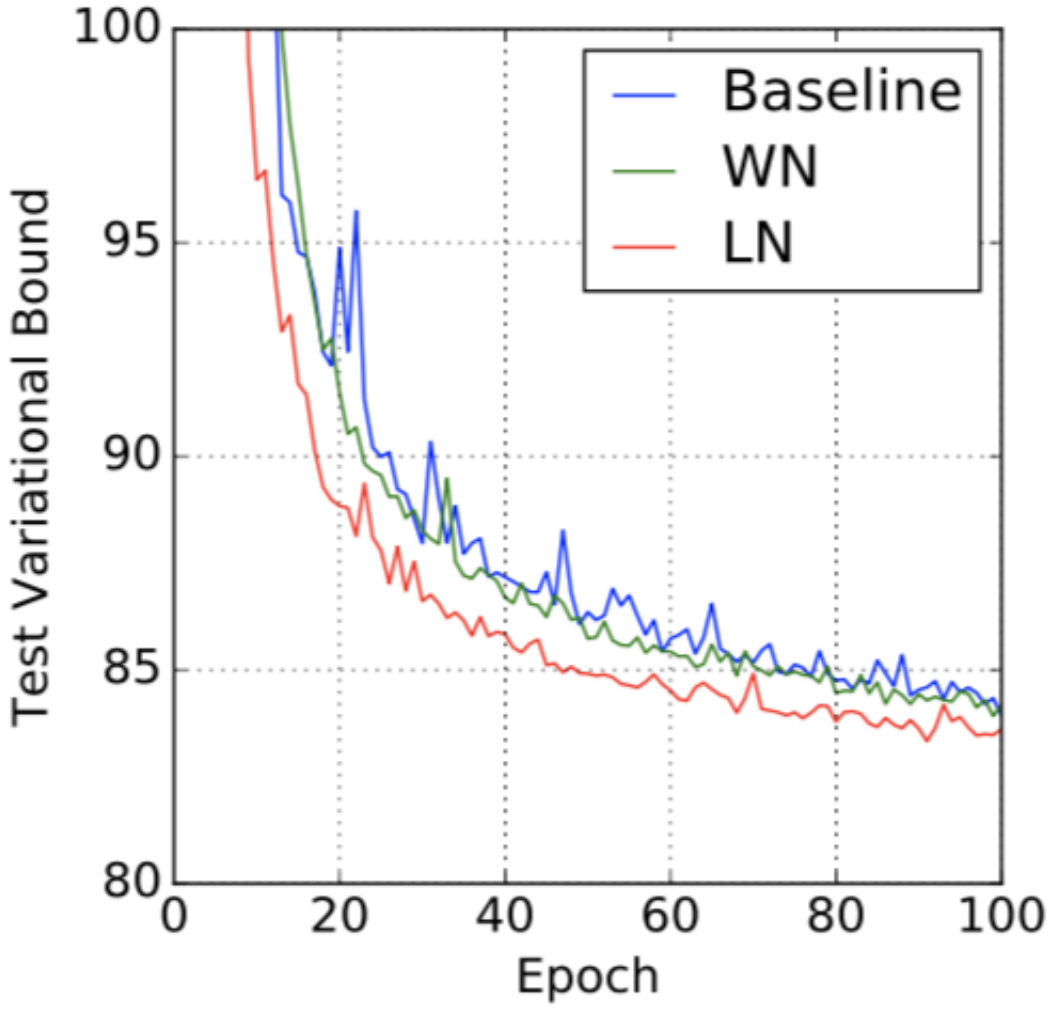


図4：層正規化の有無にかかわらず、負の対数尤度を持つDRAWモデルテスト。

図4は、最初の100エポック。これは、レイヤーノーマライズを適用することによるスピードアップの利点を強調しています。 層正規化されたDRAWがベースラインモデルよりもほぼ2倍の速さで収束することを示している。 200回のエポッシュの後、ベースラインモデルは、試験データ上の82.36ナットの変量対数尤度に収束し、層正規化モデルは82.09ナットを得る。

6.5 Handwriting sequence generation

これまでの実験では、長さが10から40までのNLPタスクでRNNを調べていました。長いシーケンスに対するレイヤ正規化の有効性を示すために、IAMオンライン手書きデータベースを使用して手書き生成タスクを実行しました[Liwicki and Bunke、2005]。 IAM-OnDBは221人の異なる作家から収集された手書きの行で構成されています。入力文字列が与えられると、目標はホワイトボード上の対応する手書き線のxおよびyペン座標のシーケンスを予測することである。合計で12179の手書き線シーケンスが存在する。入力文字列は通常25文字以上で、平均手書き文字列の長さは約700です。

Graves [2013]のセクション（5.2）と同じモデルアーキテクチャを使用しました。モデルアーキテクチャは、出力層に2つの二変量ガウス混合成分を生成する400個のLSTMセルの3つの隠れ層と、サイズ3の入力層とからなる。文字列はワンホットベクトルで符号化されていたため、ウィンドウベクトルはサイズ57でした。ウィンドウパラメータには10個のガウス関数の混合が使用され、サイズ30のパラメータベクトルが必要でした。総重量は約3.7Mに増加しました。このモデルは、サイズ8のミニバッチとAdam [Kingma and Ba、2014]オプティマイザを使用して訓練されています。

小さなミニバッチサイズと非常に長いシーケンスを組み合わせることで、非常に安定した隠れダイナミクスが重要になります。図5は、レイヤー正規化がベースラインモデルと同等の対数尤度に収束するが、はるかに高速であることを示している。

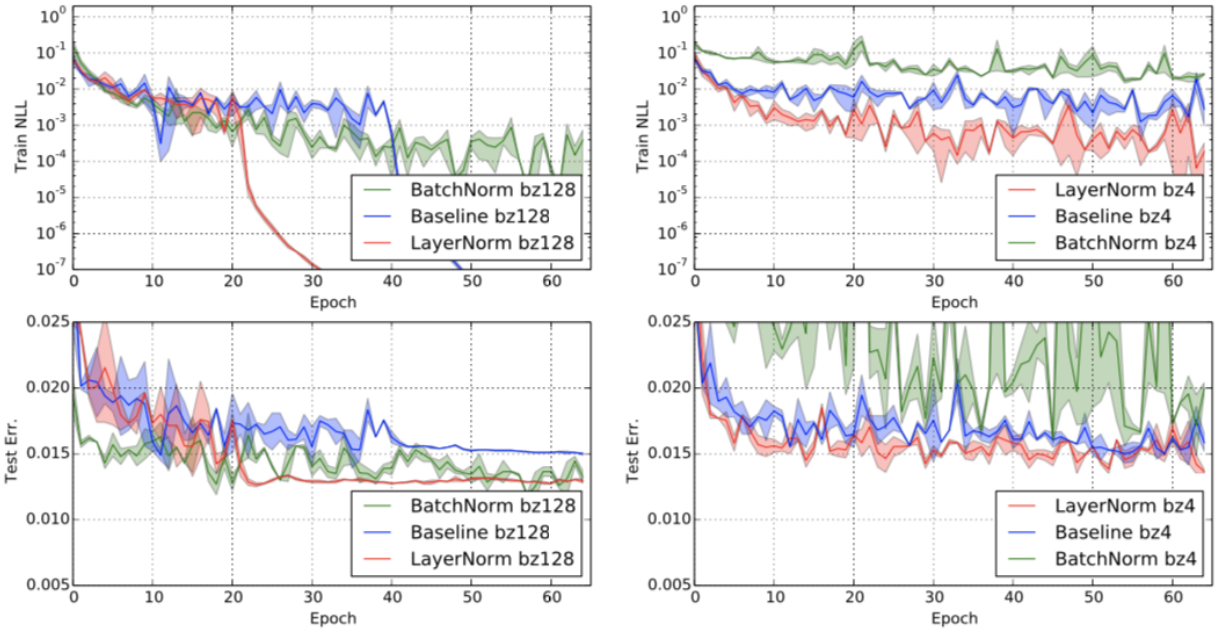


図6：パーミュテーション不変MNIST 784-1000-1000-10モデルネガティブ対数尤度と層正規化とバッチ正規化によるテスト誤差（左）モデルはバッチサイズ128で訓練されています。（右）モデルはバッチサイズ4で訓練されています。

6.6 Permutation invariant MNIST

RNNに加えて、我々はフィードフォワードネットワークにおける層正規化を調べた。我々は、よく研究された順列不変MNIST分類問題の層正規化とバッチ正規化との比較を示す。前の分析から、層の正規化は入力の再スケーリングに対して不変であり、内部隠れ層にとって望ましい。しかし、これは、予測信頼度がロジットのスケールによって決定されるロジット出力には不要です。最後のsoftmaxレイヤーを除外した、完全に接続された隠れレイヤーにのみレイヤー正規化を適用します。 すべてのモデルは、55000のトレーニングデータポイントとAdam [Kingma and Ba、2014]オプティマイザを使用して訓練されました。より小さなバッチサイズの場合、バッチ正規化の分散項はバイアスのない推定器を使用して計算されます。図6の実験結果は、レイヤの正規化がバッチサイズに対して堅牢であり、すべてのレイヤに適用されるバッチ正規化と比較して、より速いトレーニングコンバージェンスを示すことを強調しています。

6.7 Convolutional Networks

また、畳み込みニューラルネットワークを実験しました。我々の予備実験では、層の正規化は、正規化なしでベースラインモデルよりもスピードアップを提供するが、バッチ正規化は他の方法よりも優れていることがわかった。完全に接続されたレイヤーでは、レイヤー内のすべての隠しユニットが最終的な予測に同様の寄与をし、レイヤーへの合計入力を再センタリングして再スケーリングする傾向があります。しかし、畳み込みニューラルネットワークでは、類似の寄与の仮定はもはや真ではない。受像フィールドが画像の境界近くにある多数の隠れユニットは、めったにオンにならず、したがって、同じ層内の隠れユニットの残りのユニットとは非常に異なる統計を有する。ConvNetsではレイヤの正規化をうまく機能させるためにはさらなる研究が必要だと考えています。

7 Conclusion

本稿では、ニューラルネットワークの学習を高速化するために層正規化を導入した。我々は、層正規化の不変性をバッチ正規化および重量正規化と比較する理論的分析を提供した。レイヤの正規化は、トレーニングケースのフィーチャシフトとスケーリングごとに不変であることを示しました。

経験的に、我々は、長いシーケンスと小さなミニバッチの場合、提案された方法からリカレントニューラルネットワークが最も有益であることを示した。

Acknowledgments

この研究は、NSERC、CFI、およびGoogleからの助成金によって資金提供されました。

References

1. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neuralnetworks. In NIPS, 2012.
2. Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior,Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling inspeech recognition: The shared views of four research groups. IEEE, 2012.
3. Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker,Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In NIPS, 2012.
4. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducinginternal covariate shift. ICML, 2015.
5. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. InAdvances in neural information processing systems, pages 3104–3112, 2014.
6. Ce´sar Laurent, Gabriel Pereyra, Phile´mon Brakel, Ying Zhang, and Yoshua Bengio. Batch normalized recurrentneural networks. arXiv preprint arXiv:1510.01378, 2015.
7. Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen,Mike Chrzanowski, Adam Coates, Greg Diamos, et al. Deep speech 2: End-to-end speech recognition inenglish and mandarin. arXiv preprint arXiv:1512.02595, 2015.
8. Tim Cooijmans, Nicolas Ballas, Ce´sar Laurent, and Aaron Courville. Recurrent batch normalization. arXivpreprint arXiv:1603.09025, 2016.
9. Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate train-ing of deep neural networks. arXiv preprint arXiv:1602.07868, 2016.
10. Behnam Neyshabur, Ruslan R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deepneural networks. In Advances in Neural Information Processing Systems, pages 2413–2421, 2015.
11. Shun-Ichi Amari. Natural gradient works efficiently in learning. Neural computation, 1998.
12. Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. ICLR, 2016.
13. The Theano Development Team, Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller,Dzmitry Bahdanau, Nicolas Ballas, Fre´de´ric Bastien, Justin Bayer, Anatoly Belikov, et al. Theano: A pythonframework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688, 2016.

14. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dolla´r, andC Lawrence Zitnick. Microsoft coco: Common objects in context. ECCV, 2014.
15. Kyunghyun Cho, Bart Van Merrie¨nboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, HolgerSchwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statisticalmachine translation. EMNLP, 2014.
16. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition.ICLR, 2015.
17. Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multi-modal neural language models. arXiv preprint arXiv:1411.2539, 2014.
18. D. Kingma and J. L. Ba. Adam: a method for stochastic optimization. ICLR, 2014. arXiv:1412.6980.
19. Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. CVPR, 2016.
20. Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In NIPS, 2015.
21. Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In NIPS, 2015.
22. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
23. Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and SanjaFidler. Aligning books and movies: Towards story-like visual explanations by watching movies and readingbooks. In ICCV, 2015.
24. Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli.Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences throughsemantic relatedness and textual entailment. SemEval-2014, 2014.
25. Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respectto rating scales. In ACL, pages 115–124, 2005.
26. Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In Proceedings of the tenth ACMSIGKDD international conference on Knowledge discovery and data mining, 2004.
27. Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization basedon minimum cuts. In ACL, 2004.
28. Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in lan-guage. Language resources and evaluation, 2005.
29. K. Gregor, I. Danihelka, A. Graves, and D. Wierstra. DRAW: a recurrent neural network for image generation.arXiv:1502.04623, 2015.
30. Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In AISTATS, volume 6,page 622, 2011.
31. Marcus Liwicki and Horst Bunke. Iam-ondb-an on-line english sentence database acquired from handwrittentext on a whiteboard. In ICDAR, 2005.
32. Alex Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.