

NOPROP:ニューラルネットワークのトレーニング バックプロパゲーションまたはフォワードプロパゲーション

Qinyu Liオ
ックスフォード大学統計学部
qinyu.li@stats.ox.ac.uk

Yee Whye Tehオ
ックスフォード大学統計学部
ywteh@stats.ox.ac.uk

ラスヴァン・パスカヌ・ミラ
r.pascanu@gmail.com

抽象的な

標準的な深層学習の学習アプローチでは、出力からの誤差信号を各学習可能パラメータへと逆伝播させることで、各層で勾配項を計算する必要があります。ニューラルネットワークの積層構造では、各層が前の層の表現に基づいて構築されるため、このアプローチは階層的な表現につながります。より抽象的な特徴はモデルの最上位層に存在し、下位層の特徴はより抽象度が低いことが期待されます。これとは対照的に、我々はNoPropという新しい学習手法を導入します。これは順伝播にも逆伝播にも依存しません。NoPropは拡散法とフローマッチング法から着想を得ており、各層は独立してノイズの多いターゲットのノイズ除去を学習します。この研究は、少なくとも通常の意味では階層的な表現を学習しない、新しい勾配フリー学習手法ファミリーの導入に向けた第一歩であると考えています。NoPropでは、各層の表現を事前にノイズのかかったターゲットに固定し、推論時に利用できる局所的なノイズ除去プロセスを学習する必要があります。MNIST、CIFAR-10、CIFAR-100の画像分類ベンチマークにおいて、本手法の有効性を実証しました。その結果、NoPropは、既存のバックプロパゲーションを必要としない手法と比較して、優れた精度、使いやすさ、そして計算効率を実現する実用的な学習アルゴリズムであることがわかりました。NoPropは、従来の勾配学習パラダイムから脱却することで、ネットワーク内でのクレジット割り当て方法を変更し、より効率的な分散学習を可能にするだけでなく、学習プロセスの他の特性にも影響を与える可能性があります。

1はじめに

バックプロパゲーション(Rumelhart 他、1986年)は長い間ディープラーニングの基礎となっており、その応用によりディープラーニング技術は科学から産業まで幅広い分野で目覚ましい成功を収めることができました。

簡単に言えば、これは反復的なアルゴリズムであり、多層ニューラルネットワークの各ステップでパラメータを適応させ、出力が目的のターゲットにより適合するように調整します。バックプロパゲーションの各ステップでは、まず入力信号の順方向伝播を実行して予測値を生成し、次にその予測値を目的のターゲットと比較し、最後に誤差信号をネットワークに逆伝播させて、誤差を低減するために各層の重みをどのように調整すべきかを決定します。

このように、各層は、下位層から受け取った表現を、後続層がより良い予測を行うために使用できる表現へと変更することを学習していると考えられます。後方に伝播される誤差信号は、クレジットの割り当て、つまり誤差を最小化するために各パラメータをどの程度変更する必要があるかを決定するために使用されます。

バックプロパゲーションは単純であるため、ニューラルネットワークのトレーニングにおける事実上の標準手法となっています。しかしながら、長年にわたり、バックプロパゲーションに依存しない代替手法の開発への関心が高まっています。この関心にはいくつかの要因があります。第一に、バックプロパゲーションは、フォワードパスとバックワードパスを同期させて交互に実行する必要があるため、生物学的に実現不可能です(例: Lee et al., 2015)。第二に、バックプロパゲーションでは、バックワードパスでの勾配計算を容易にするために、フォワードパス中に中間活性化を保存する必要があり、これが大きなメモリオーバーヘッドを引き起こす可能性があります(Rumelhart et al., 1986)。最後に、勾配の順次伝播は、並列計算を妨げる依存関係を導入し、大規模な機械学習に複数のデバイスとサーバーを効果的に活用することを困難にします(Carreira-Perpinan & Wang, 2014)。クレジットの割り当てが計算されるこの順次的な性質は、学習にも追加的な影響を与え、干渉(Schau et al., 2019)につながり、壊滅的な忘却に影響を与える(Hadsell et al., 2020)とされています。

逆伝播法の代替最適化手法としては、勾配フリー法(例えば、直接探索法(Fermi, 1952; Torczon, 1991)およびモデルベース法(Bortz & Kelley, 1997; Conn et al., 2000))、ゼロ次勾配法(Flaxman et al., 2004; Duchi et al., 2015; Nesterov & Spokoyny, 2015; Liu et al., 2020; Ren et al.,

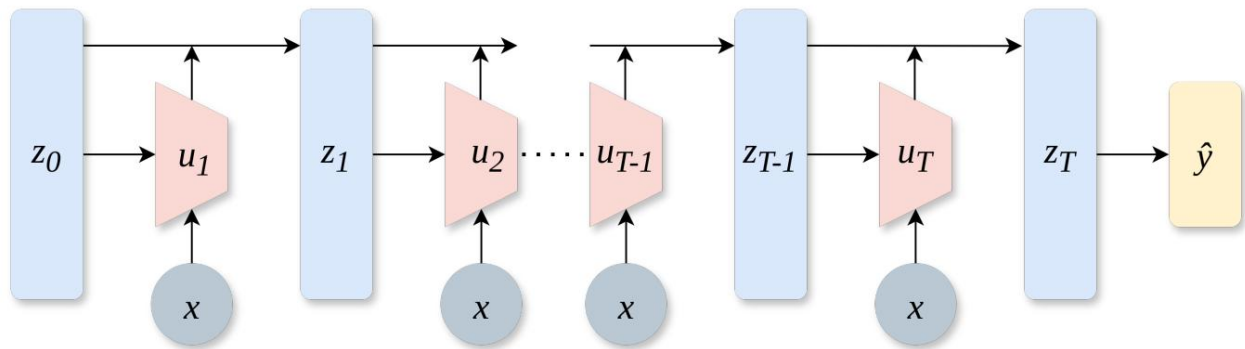


図 1: NoProp のアーキテクチャ。\$z_0\$ はガウスノイズを表し、\$z_1, \dots, z_T\$ は学習したダイナミクス \$u_1, \dots, u_T\$ を介した連続的な変換であり、各レイヤーは画像 \$x\$ に基づいて条件付けられ、最終的にクラス \$u_T\$、予測 \$y\$ を生成します。

勾配を近似するために有限差分を用いる手法 (Wierstra et al., 2014; Salimans et al., 2017; Such et al., 2017; Khadka & Tumer, 2018)、差分ターゲット伝播法 (Lee et al., 2015) や順方向アルゴリズム (Hinton, 2022) などの局所損失を利用する手法などが挙げられる。しかし、これらの手法は、精度、計算効率、信頼性、スケーラビリティの面で限界があるため、ニューラルネットワークの学習には広く採用されていない。

本稿では、バックプロパゲーションを必要としない学習手法を提案する。この手法は、拡散モデル (Sohl-Dickstein et al., 2015; Chen et al., 2018; Ho et al., 2020; Song et al., 2021) の基盤となるノイズ除去スコアマッチング手法をベースとしており、ニューラルネットワークの各層を独立に学習させることができる。簡単に説明すると、学習時には各層がノイズラベルと学習入力を与えられた場合にターゲットラベルを予測するように学習される。一方、推論時には各層が前の層で生成されたノイズラベルを受け取り、予測したラベルに近づくようにノイズ除去を行う。特に注目すべきは、学習時にはフォワードパスさえ必要としないという点である。そのため、本手法は NoProp と呼んでいる。MNIST、CIFAR-10、CIFAR-100 ベンチマークで実験的に、NoProp は従来のバックプロパゲーションを使用しない方法よりも大幅にパフォーマンスが優れていると同時に、よりシンプルで堅牢かつ計算効率が高いことを示しています。

2 方法論

このセクションでは、バックプロパゲーションフリー学習のための NoProp 法について説明します。技術的な考え方は変分拡散モデル (Sohl-Dickstein et al., 2015; Kingma et al., 2021; Gulrajani & Hashimoto, 2024) の考え方にほぼ沿っていますが、本研究では異なる文脈に適用し、異なる解釈を行います。

2.1 ノープロパティ

\$x\$ と \$y\$ を分類データセット内のサンプルの入力ラベルペアとし、データ分布 \$q_0(x, y)\$ から抽出したものと仮定します。また、\$z_0, z_1, \dots, z_T \in \mathbb{R}^T\$ を \$T\$ ブロックのニューラルネットワークの対応する確率的中間活性化とし、\$q_0(y|x)\$ を推定するためにトレーニングすることを目指します。

次のように分解された 2 つの分布 \$p\$ と \$q\$ を定義します。

$$p((z_t)_{t=0}^T, y|x) = p(z_0) \prod_{t=1}^T p(z_t|z_{t-1}, x) p(y|z_T), \quad (1)$$

$$q((z_t)_{t=0}^T|y, x) = q(z_T|y) \prod_{t=1}^T q(z_{t-1}|z_t) \text{ です。} \quad (2)$$

分布 \$p\$ は、前の活性化 \$z_{t-1}\$ と入力 \$x\$ から次の活性化 \$z_t\$ を反復的に計算する確率的順伝播過程として解釈できます。実際には、活性化にガウスノイズを加えた残差ネットワークとして明示的に与えることができます。

$$z_t = a_t u_{\theta_t}(z_{t-1}, x) + b_t z_{t-1} + \sqrt{c_t} \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(\epsilon_t | 0, 1) \text{ ここで、} \quad (3)$$

\$\mathcal{N}(\cdot | 0, 1)\$ は平均ベクトルが 0 で共分散行列が恒等である \$d\$ 次元ガウス密度、\$a_t, b_t, c_t\$ はスカラー (下記参照)、\$b_t z_{t-1}\$ は重み付きスキップ接続、\$u_{\theta_t}(z_{t-1}, x)\$ は残差ブロックで、次式でパラメータ化される。

3

完全な導出については付録A.4を参照してください。ここでは、時間 t は $[0, 1]$ の連続変数として扱われます。連続ノイズスケジュールを使用し、 $\text{SNR}'(t)$ は信号対雑音比の導関数を表します。単一のブロック $u_{\theta}(z_t, x, t)$ は、 t を追加入力として、すべてのタイムステップ t に対してトレーニングされます。潜在変数 z_t の進化は、確率微分方程式（SDE）で記述される連続時間拡散プロセスに従います。SDEと同じ周辺分布を共有する対応する常微分方程式（ODE） $\dot{z}_t = f(z_t | x, t)$ が存在することに注意してください（Song et al., 2021）。関数 $f(z_t | x, t)$ は、 z_t の進化を生成する決定論的ベクトル場を表します。このようにパラメータ化されたニューラルネットワークは、ニューラルODEとして知られています（Chen et al., 2018）。

ニューラルODEの学習では、通常、ODEソルバーを介したバックプロパゲーションを用いてタスク固有の損失関数（例：分類におけるクロスエントロピー）を最適化するか、随伴感度法（Chen et al., 2018）を用いて別のODEを時間的に逆方向に解くことで勾配を推定します。一方、連続時間拡散モデルは、事前定義されたノイズ処理を反転するODEダイナミクスを学習します。学習は時間ステップを独立にサンプリングすることで行われ、時間的に順方向または逆方向のパスを必要とせずに行われます。これにより、学習効率が向上し、表現力豊かなODEダイナミクスを実現できます。

2.2.3 フローマッチング

NoPropの連続定式化の代替アプローチはフロー マッチング(Lipman et al., 2022; Tong et al., 2023)です。これは、ODE を介して予測ラベル埋め込みに向かってノイズを転送するベクトル フィールド $f(z_t | x, t)$ を直接学習します。

$f(z_t | x, t)$ は一般に未知であるため、フローマッチングでは代わりに条件付きベクトル場 $f(z_t | z_0, z_1, x, t)$ を学習します。ここで、 z_0 は初期ノイズ、 $z_1 = u_y$ はラベル埋め込みです。条件付きベクトル場はユーザーが指定します。単純な方法としては、ノイズ z_0 とラベル埋め込み $z_1 = u_y$ の間のガウス確率パスを定義することが挙げられます。

$$p_t(z_t | z_0, z_1, x) = \mathcal{N}(z_t | tz_1 + (1 - t)z_0, \sigma^2), \quad (10)$$

対応するベクトル場 $f(z_t | z_0, z_1, x, t) = z_1 - z_0$ となる。フローマッチングの目的は、

$$\mathbb{E}_t \left[U[0, 1], q_0(x, y), p(z_0), p_t(z_t | z_0, z_1, x) \right] \propto \int_0^1 \mathbb{E} \left[\left\| \frac{dz_t}{dt} - (z_1 - z_0) \right\|^2 \right] dt, \quad (11)$$

ここで、 $v_{\theta}(z_t, x, t)$ はパラメータ θ を持つニューラルネットワークである。ラベル埋め込みを共同学習する場合、異なるクラスの埋め込みが崩れるのを防ぐため、追加のアンカー損失を導入する。先行研究（Gao et al., 2022; Hu et al., 2024）に従い、クロスエントロピー項 $-\log p_{\theta}(\text{out}(y | z_1(z_t, x, t)))$ を組み込む。ここで、 $z_1(z_t, x, t)$ は外挿線形推定値である。

$$z_1(z_t, x, t) = z_t + (1 - t)v_{\theta}(z_t, x, t) \text{ です。} \quad (12)$$

修正されたフローマッチングの目的は

$$\mathbb{E}_t \left[U[0, 1], q_0(x, y), p(z_0), p_t(z_t | z_0, z_1, x) \left[\int_0^1 \mathbb{E} \left[\left\| \frac{dz_t}{dt} - (z_1 - z_0) \right\|^2 \right] dt - \log p_{\theta}(\text{out}(y | z_1(z_t, x, t))) \right] \right]. \quad (13)$$

拡散と同様に、線形層とそれに続くソフトマックスを用いて $p_{\theta}(\text{out}(y | z_1(z_t, x, t)))$ をパラメータ化します。この定式化により、クラス埋め込みが適切に分離されます。

2.3 実装の詳細

2.3.1 アーキテクチャ

NoPropアーキテクチャを図1に示します。推論中、ガウスノイズ z_0 は拡散過程を通じて一連の変換を受けます。各ステップにおいて、潜在変数 z_t は拡散動的ブロック u_t を介して進化し、 z_T に達するまで z_1, z_2, \dots, z_T というシーケンスを生成します。

各 u_t は、前の潜在状態 z_{t-1} と入力画像 x に基づいて条件付けられます。最後に、線形層とそれに続くソフトマックス関数によって、 z_T が予測ラベル y にマッピングされます。

各 u_t ブロックをパラメータ化するために使用されるモデルについては、セクション2.3.2で説明します。

このアーキテクチャは推論専用設計されています。学習中は時間ステップがサンプリングされ、各拡散ダイナミックブロック u_t は独立して学習されます。線形層と埋め込み行列は拡散ブロックと共同で学習されます。線形層はクラス埋め込みの崩壊を防ぐのに役立ちます。

フローマッチング法では、図1の u_t は常微分方程式のダイナミクスを表します。ラベル予測 y は、ユークリッド距離で z_T に最も近いクラス埋め込みを求めることで、 z_T から直接得られます。

2.3.2 トレーニング手順

モデル学習に使用したモデルは付録の図6に示されている。離散時間拡散の場合、ニューラルネットワーク $u_{\theta}(t)$ を用いて拡散ダイナミクス u_t をモデル化する。このモデルは入力画像 x と潜在変数 x を取り、

ラベル埋め込み空間内の z_{t-1} を、連結する前に別々の埋め込み経路で処理します。

画像 x は畳み込み埋め込みモジュールに渡され、その後全結合層が続きます。埋め込み次元が画像次元と一致する場合、 z_{t-1} は画像として扱われ、同様に畳み込みモジュールを用いて埋め込まれます。そうでない場合、 z_{t-1} はスキップ接続を含む全結合ネットワークに渡されます。融合された表現は、追加の全結合層によって処理され、出力としてロジットが生成されます。拡散ダイナミクス $u_{\theta t}$ は、ロジットにソフトマックス関数を適用することで得られ、クラス埋め込みの確率分布を生成します。最終出力 $u_{\theta t}$ は、この確率分布を用いてWEmbedにおけるクラス埋め込みの加重和として計算されます。

連続時間拡散の場合、 u_{θ} を学習します。 u_{θ} は、追加のタイムスタンプ t を入力として受け取ります。タイムスタンプは位置埋め込みを用いて符号化され、全結合層で処理された後、 x および z_t のタイムスタンプと連結されます。モデルの残りの部分は、離散時間の場合とほぼ同じ構造です。

フローマッチングのために、ニューラルネットワーク v_{θ} を学習します。アーキテクチャは連続時間拡散の場合と同じですが、ソフトマックスを適用する代わりに、ロジットを用いてクラス埋め込みの重み付き和を直接計算し、ロジットを無制限の重みとして扱うことで v_{θ} を得ます。これにより、クラス埋め込みの凸結合に制約される u_{θ} とは異なり、 v_{θ} は埋め込み空間内の任意の方向を表すことができます。

モデル全体の構造は設定間で同様ですが、連続時間拡散およびフローマッチングの場合に t が組み込まれることで、離散時間拡散の場合と比較して、追加の入力および処理ステップが導入されます。式8の $p_{\theta out}(y|z_T)$ 、式9の $p_{\theta out}(y|z_1)$ 、および式13の $p_{\theta out}(y|z_1(z_t, x, t))$ を、線形層とそれに続くソフトマックスを用いてパラメータ化することを選択します。

ノイズスケジュール離散時間拡散の場合、固定のコサインノイズスケジュールを使用します。連続時間拡散の場合、ノイズスケジュールはモデルと連携して学習します。学習可能なノイズスケジュールの詳細については、付録Bを参照してください。

関連作品3件

3.1バックプロパゲーションフリー法

フォワードフォワードアルゴリズム(Hinton, 2022)は、画像分類において、従来のフォワードパスとバックワードパスを2つのフォワードパスに置き換えます。1つは正しいラベルとペアになった画像を使用し、もう1つは誤ったラベルとペアになった画像を使用します。

ターゲット伝播とその変種 (例: Lee et al., 2015, Lillicrap et al., 2016)は、出力層から隠し層への逆マッピングを学習することで機能し、バックプロパゲーションなしで重み更新のためのローカルターゲットを生成します。

ただし、そのパフォーマンスは学習した逆マッピングの品質に大きく依存し、オートエンコーダの不正確さにより更新が最適ではなくなる可能性があり、ディープネットワークでの有効性が制限されます。

ゼロ次法(Flaxman 他, 2004 年, Duchi 他, 2015 年, Ren 他, 2022 年)は、明示的な微分なしで勾配を推定する広範なアルゴリズムのファミリーであり、ブラックボックスおよび微分不可能な関数に役立ちます。

自然進化に着想を得た進化戦略 (Wierstra et al., 2014; Salimans et al., 2017; Khadka & Tumer, 2018)は、ランダムノイズによる摂動と集団ベース最適化アプローチを用いた報酬増加の方向への更新によってパラメータを最適化します。しかし、これらの戦略は多数の関数評価を必要とし、サンプル複雑度が高くなるため、高次元パラメータ空間における効果的な探索は依然として重要な課題となっています。

3.2拡散とフローマッチング

拡散とフローマッチングに関するいくつかの研究は、本手法と密接に関連しています。Han et al. (2022)は分類と回帰拡散モデルを導入し、Kim et al. (2025) はペアデータに対するフローマッチングを提案し、Hu et al. (2024) はフローマッチングを条件付きテキスト生成に適用しました。一方、本論文では、これらのアイデアの含意をバックプロパゲーションフリー学習の枠組みの中で探求します。

3.3表現学習

一般的に言えば、バックプロパゲーションに代わる手法のほとんどは、単に勾配を異なる方法で近似する場合でも、進化戦略のように異なる探索スキームを利用する場合でも、依然として、互いに積み重なる中間表現の学習に依存しています (Bengio et al., 2013)。これにより、モデルのより深い層を見るにつれてより抽象的な表現を学習することができ、深層学習とニューラルネットワークにとって根本的に重要であると考えられています。

認知プロセスを表現する能力 (Markman & Dietrich, 2000)。実際、深層学習の初期の成功は、深層アーキテクチャを階層的表現を学習させる能力に起因するとされ (Hinton et al., 2006; Bengio et al., 2013)、初期の解釈可能性に関する研究は、ますます複雑化するこれらの特徴を視覚化することに重点を置いていました (Zeiler & Fergus, 2014; Lee et al., 2009)。

各層にラベルのノイズ除去を学習させ、ラベルのノイズ分布をユーザーが選択するようにすれば、NoProp は表現を学習しないと言えるでしょう。むしろ、ユーザーが設計した表現に依存します (具体的には、中間層の表現は、拡散の場合はターゲット ラベルのガウス ノイズ付き埋め込み、フロー マッチングの場合はガウス ノイズとターゲット ラベルの埋め込み間の補間です)。これは意外ではないかもしれませんが、順方向伝播と逆方向伝播は、各層の表現が隣接する層の表現と「適合」し、最終層の表現からターゲット ラベルが簡単に予測できるようにするために、ニューラル ネットワークの層全体に情報が伝播されていると理解できます。そのため、順方向伝播や逆方向伝播なしで NoProp が機能するには、これらの層の表現を事前に固定する (つまり、ユーザーが設計する) 必要があります。

NoPropが表現学習なしで優れた性能を達成しているという事実は、表現学習が実際に深層学習に必要なかどうか、そして表現を設計することで、異なる特性を持つ深層学習への代替アプローチが可能になるかどうかという疑問を生じさせます。特に、NoPropで固定されている表現は、後続層でより抽象化されると考えられる表現とは異なる点に注意してください。これは、複雑な行動をモデル化する際の階層的表現の役割を再検討する道を開きます (Markman & Dietrich, 2000)。これらの疑問は、バックプロパゲーションに基づく学習の中核となる仮定、例えばiid仮定や、ネットワークを通じた情報と誤差信号の順次伝播が限界であることが明らかになっているため、ますます重要になる可能性があります。

4つの実験

画像分類タスクにおいて、離散時間モデルではNoPropをバックプロパゲーションと比較し、連続時間モデルでは随伴感度法 (Chen et al., 2018)と比較します。ハイパーパラメータの詳細は付録の表3をご覧ください。

データセット。MNIST、CIFAR-10、CIFAR-100の3つのベンチマークデータセットを使用します。MNISTデータセットは、手書き数字 (28×28ピクセル)のグレースケール画像70,000枚で構成され、10クラス (数字0~9)にまたがります。CIFAR-10には、10の異なるオブジェクトクラスにまたがるカラー画像 (32×32ピクセル)60,000枚が含まれています。CIFAR-100はCIFAR-10と構造が似ており、60,000枚のカラー画像を含みますが、100の細分化されたクラスに分割されています。トレーニングセットとテストセットの分割は標準の分割を使用します。私たちの実験では、これらのデータセットにデータ拡張技術を適用しません。

NoProp (離散時間)この手法をNoProp with Discrete-Time Diffusion (NoProp-DT)と呼びます。T = 10 を固定し、各エポックにおいて、アルゴリズム1で概説されているように、各タイムステップのパラメータを順次更新します。逐次更新はアルゴリズムの動作に厳密には必要ではない (時間ステップもサンプリングできる)が、他のバックプロパゲーションフリー手法と整合させ、文献の既存の手法との一貫性を確保するために、このアプローチを選択した。式8のクラス確率 $p_{\theta_{out}(y|zT)}$ のパラメータ化に関しては、2つのアプローチを検討する。1つ目は、前述のように、パラメータ θ_{out} を持つ完全結合層の出力にソフトマックスを適用するものであり、

$$p_{\theta_{out}(y|zT)} = \text{ソフトマックス}(f_{\theta_{out}}(zT)y) = \frac{\exp(f_{\theta_{out}}(zT)y)}{\sum_{y'=1}^{m_l} \exp(f_{\theta_{out}}(zT)y')}, \tag{14}$$

ここで、 m_l はクラス数である。さらなる検討として、クラス確率をパラメータ化するために放射状距離を利用する代替アプローチを検討する。このアプローチでは、 zT から再構成されたラベル y を推定する。これは、同じ全結合層 θ_{out} の出力にソフトマックスを適用し、クラス埋め込み行列 W_{Embed} に投影することで行われる。この場合、 y は重み付き埋め込みとなる。 zT が与えられた場合のクラス y の確率は、 y と真のクラス埋め込み $u_y = (W_{Embed})y$ との間のユークリッド距離の2乗に基づいて算出される。

$$y = \text{ソフトマックス}(f_{\theta_{out}}(zT))W_{Embed}, \tag{15}$$

$$p_{\theta_{out}(y|zT)} = \frac{\exp(-\frac{\|y - u_y\|^2}{2\sigma^2})}{\sum_{y'=1}^{m_l} \exp(-\frac{\|y - u_{y'}\|^2}{2\sigma^2})}. \tag{16}$$

この定式化は、等しい事前クラス確率と正規尤度 $p(y|y) = Nd(y|u_y, \sigma^2)$ を仮定して事後クラス確率を計算するものとして解釈できます。

プレプリント。

方法	MNIST		CIFAR-10		CIFAR-100	
	電車	テスト	電車	テスト	電車	テスト
離散時間						
バックプロパゲーション（ワンホット）	100.0±0.0	99.46±0.06	99.98±0.01	79.92±0.14	98.63±1.34	45.85±2.07
バックプロパゲーション（次元=20）	99.99±0.0	99.43±0.03	99.96±0.02	79.3±0.52	94.28±7.43	46.57±0.87
バックプロパゲーション（プロトタイプ）	99.99±0.01	99.44±0.05	99.97±0.01	79.58±0.44	99.19±0.71	47.8±0.19
NoProp-DT（ワンホット）	99.92±0.01	99.47±0.05	95.02±0.19	79.25±0.28	84.97±0.67	45.93±0.46
NoProp-DT（dim=20）	99.93±0.01	99.49±0.04	94.95±0.09	79.12±0.37	83.25±0.39	45.19±0.22
NoProp-DT（プロトタイプ）	99.97±0.0	99.54±0.04	97.23±0.11	80.54±0.2	90.7±0.14	46.06±0.25
連続時間						
随伴（ワンホット）	98.7±0.13	98.62±0.14	70.64±0.49	66.78±0.76	26.72±0.81	25.03±0.7
NoProp-CT（ワンホット）	97.88±0.61	97.84±0.71	97.31±0.84	73.35±0.55	75.1±3.43	33.66±0.5
NoProp-CT（dim=20）	97.7±0.42	97.7±0.51	94.88±3.08	71.77±2.47	74.22±2.33	33.99±1.08
NoProp-CT（プロトタイプ）	97.18±1.02	97.17±0.94	86.2±7.34	66.54±3.63	40.88±10.72	21.31±4.17
NoProp-FM（ワンホット）	99.97±0.0	99.21±0.09	98.46±0.4	73.14±0.9	12.69±10.4	6.38±4.9
NoProp-FM（dim=20）	99.99±0.0	99.29±0.05	99.49±0.15	73.5±0.28	83.49±4.62	31.14±0.52
NoProp-FM（プロトタイプ）	99.27±0.09	98.52±0.16	99.8±0.03	75.18±0.57	96.37±1.09	37.57±0.32
従来のバックプロパゲーションフリー法						
前進前進	-	98.63	-	-	-	-
順方向勾配	100.00	97.45	80.61	69.32	-	-
差異ターゲットプロパティ	-	98.06	-	50.71	-	-

表1 :MNIST,CIFAR-10,CIFAR-100における分類精度（％）。平均値と標準誤差は、手法ごとに3つのシードとシードあたり5回の推論実行から得られた15個の値から計算されています。順方向勾配とは、Ren et al. (2022)の Local Greedy Forward Gradient Activity-Perturbed (LG-FG-A) 法。one-hot: 固定 one-hot ラベル埋め込み; dim=20: 次元 20 の学習済みラベル埋め込み; プロトタイプ: 画像サイズに等しい次元の学習済みラベル埋め込み。

バックプロパゲーション.T = 10 を固定し、フォワードパスは次のように与えられる。

$$z_0 \quad N_d(z_0|0, 1), z_t = (1 \tag{17}$$

$$- \alpha t) z_{t-1} + \alpha t \quad \theta_t(z_{t-1}, x), t = 1, \dots, T, y = \operatorname{argmax}_{i \in \{1, \dots, m\}} \tag{18}$$

$$p \quad \theta_{out}(y_i | z_T), \alpha T \text{は範囲 } (-1, 1) \text{ に制約された} \tag{19}$$

ここで α_1, \dots , この 学習可能なパラメータであり、 $\alpha t = \tanh(wt)$ として定義され、学習可能な w_1, \dots, w_T . バック設計はNoProp-DTの順方向パスに非常に似ていますが、ネットワークは標準の ロパゲーション。 $\theta_t(z_{t-1}, x)$ と $p \quad \theta_{out}(y_i | z_T)$ はNoProp-DTと同一のモデル構造を持つ。

NoProp（連続時間）連続時間バリエーションを連続時間拡散付きNoPropと呼ぶ。
（NoProp-CT）とフローマッチング付きNoProp（NoProp-FM）。学習中、時間変数tはランダムにサンプリングされる。
[0, 1]から始まる。推論中は、NoProp-CTの拡散過程をシミュレートするためにT = 1000ステップに設定し、
NoProp-FMの確率フロー常微分方程式。詳細な学習手順はアルゴリズム2と3に示されている。

随伴感度ニューラルODEを訓練する随伴感度法（Chen et al., 2018）も評価する。
関連する随伴方程式を時間的に遡って解くことで推定された勾配を用いる。この手法を用いることで、
連続時間の場合のアプローチの効率と精度を評価するための有意義な基準となる。公平な
比較すると、トレーニングと推論の両方で T = 1000 を固定し、最終的な分類には線形レイヤーを使用します。

主な結果表1にまとめた結果は、NoProp-DTが同等の性能を達成することを示しています。
離散時間設定において、MNIST,CIFAR-10,CIFAR-100のバックプロパゲーションと同等かそれ以上の結果が得られました。さらに、
NoProp-DTは、フォワードフォワードアルゴリズム、差分法など、従来のバックプロパゲーションフリー法よりも優れた性能を発揮します。
ターゲットプロパゲーション（Lee et al., 2015）、およびローカルグリーディフォワードグラディエントと呼ばれるフォワードグラディエント法
アクティビティ摂動（Ren et al., 2022）。これらの手法は異なるアーキテクチャを使用し、層を調整しないが、
NoPropのように画像入力に明示的に依存するため直接比較は困難ですが、私たちの手法は明確な
フォワードパスを必要としないという利点があります。さらに、NoPropはGPUメモリの消費量を削減します。

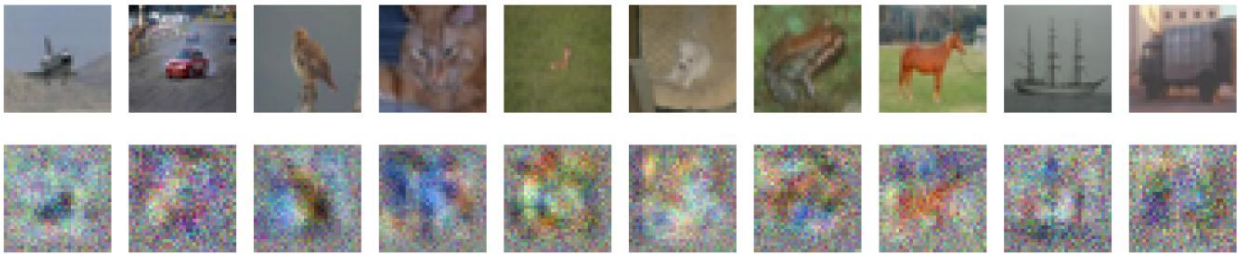


図2 :最初の行は、CIFAR-10において、同一クラス内の他の全ての画像との距離の中央値が最小となる画像を用いて初期化されたクラス埋め込みを示しています。2行目は、NoProp-DTから学習されたクラス埋め込みを示しており、これは各クラスの学習済み画像プロトタイプとして解釈できます。

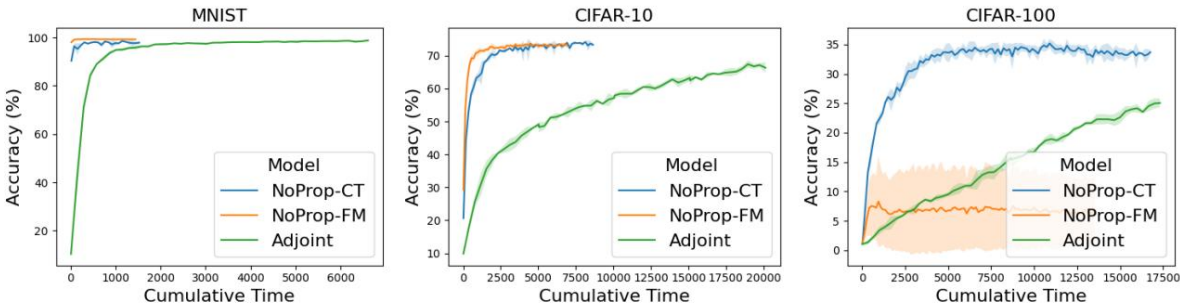


図3 :連続時間設定において、ワンホットラベル埋め込みを用いたモデルのテスト精度 (%)を累積学習時間 (秒)に対してプロットしたもの。各プロット内のすべてのモデルは、公平な比較のため、同じ種類のGPUで学習された。NoProp-CTは、アジョイント感度と比較して、精度と速度の両方で優れたパフォーマンスを達成している。CIFAR-100では、NoProp-FMはワンホットラベル埋め込みでは効果的に学習しない。

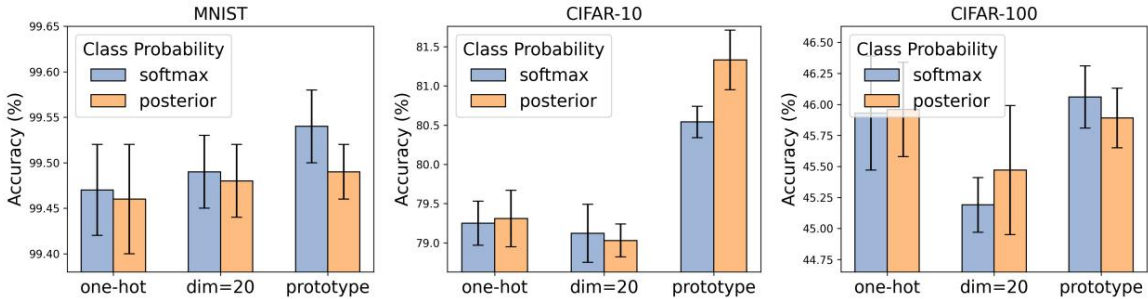


図4: 式8のクラス確率 $p_{\theta \text{out}(y|zT)}$ の2つのパラメータ化に対するソフトマックスまたは事後確率を使用したテスト精度(%)の比較。

方法	MNIST	CIFAR-10	CIFAR-100
分散時間			
バックプロップ	0.87 GB	1.17 GB	1.73 GB
NoProp-DT	0.49 GB	0.64 GB	1.23 GB
連続時間			
アジョイント	2.32 GB	6.23 GB	6.45 GB
NoProp-CT	1.05 GB	0.45 GB	0.50 GB
NoProp-FM	1.06 GB	0.44 GB	0.49 GB

表 2: ワンホット ラベル埋め込みを使用するモデルに割り当てられたプロセス GPU メモリ (GB 単位)。

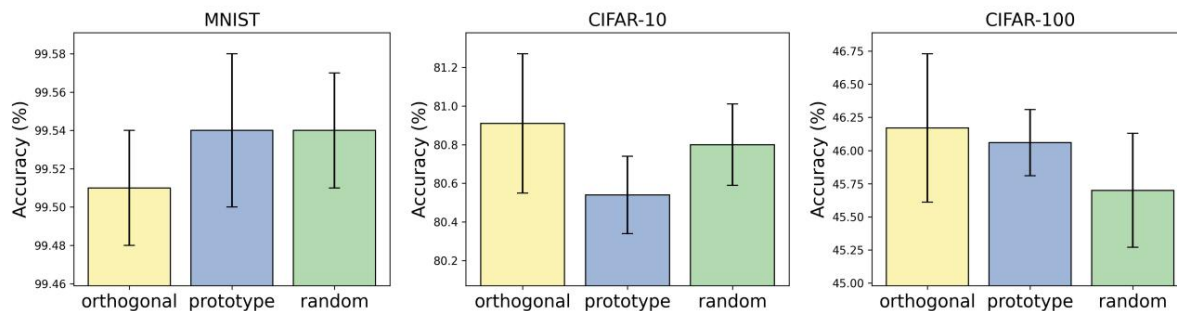


図5: クラス埋め込み行列WEmbedの異なる初期化におけるテスト精度 (%)の比較 (クラス埋め込み次元が画像次元と一致する場合)。検討対象とした初期化は、ランダム行列、直交行列、プロトタイプ画像です。

学習したクラス埋め込みを説明するために、図2は、埋め込み次元が画像次元と一致しているNoProp-DTによって学習されたCIFAR-10の初期化クラス埋め込みと最終クラス埋め込みの両方を視覚化しています。

連続設定では、NoProp-CTとNoProp-FMはNoProp-DTよりも精度が低くなります。これはおそらく、時刻に関する追加の条件付けによるものと考えられます。しかしながら、CIFAR-10およびCIFAR-100においては、精度と計算効率の両面で、アジョイント感度法を概ね上回っています。アジョイント法はMNISTにおいてNoProp-CTおよびNoProp-FMと同等の精度を達成しますが、図3に示すように、その速度ははるかに遅くなります。

CIFAR-100において、ワンホット埋め込みを用いた場合、NoProp-FMは効果的に学習できず、精度向上が非常に遅くなります。一方、NoProp-CTは依然として随伴法よりも優れた性能を示します。しかし、ラベル埋め込みを共同学習すると、NoProp-FMの性能は大幅に向上します。

また、クラス確率のパラメータ化 $p_{\theta}(\mathbf{y}|\mathbf{z})$ とクラス埋め込み行列の初期化WEmbed についてもアブレーション研究を実施し、それぞれ図4と図5に結果を示しました。アブレーションの結果、クラス確率のパラメータ化間に一貫した優位性は見られず、データセット間でパフォーマンスにばらつきが見られました。クラス埋め込みの初期化に関しては、直交初期化とプロトタイプ初期化はどちらも、一般的にランダム初期化と同等か、それを上回るパフォーマンスを示しました。

5結論

拡散モデルの基盤となるノイズ除去スコアマッチング手法を用いて、我々はニューラルネットワークを訓練するための順方向および逆伝播を必要としない手法であるNoPropを提案した。この手法により、ニューラルネットワークの各層は、ノイズを含むラベルと訓練入力を与えられた場合にターゲットラベルを予測するように独立して訓練することができる。一方、推論時には、各層は前の層によって生成されたノイズを含むラベルを受け取り、予測したラベルに近づくようにノイズを除去する。我々は実験的に、NoPropは従来の逆伝播を必要としない手法よりも大幅に性能が高く、同時により単純で、より堅牢で、より計算効率が高いことを示した。ノイズ除去スコアマッチングを介してニューラルネットワークを訓練するというこの視点は、逆伝播なしでディープラーニングモデルを訓練する新たな可能性を切り開くと信じており、我々の研究がこの方向へのさらなる研究を促すことを期待している。

参考文献

Yoshua Bengio, Aaron Courville, Pascal Vincent. 「表現学習 : レビューと新たな視点」IEEE Trans. Pattern Anal. Mach. Intell., 35(8):1798–1828, 2013年8月. ISSN 0162-8828.

デイヴィッド・ボルツとカール・ケリー。単体勾配とノイズ最適化問題。計算手法最適設計と制御, 1997年2月。

ミゲル・カレイラ＝ペルピナンとウェイラン・ワン。「深くネストされたシステムの分散最適化」。サミュエル・カスキとユッカ・コランダー編著、『Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics』、機械学習研究論文集第33巻, pp. 10–19, レイキャビク（アイランド）、2014年4月22日～25日。PMLR。

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, David K Duvenaud. ニューラル常微分方程式。神経情報処理システムの進歩, 31, 2018年。

アンドリュー・コン、ニコラス・グールド、フィリップ・トイン 『信頼領域法』SIAM, 2000年。

John C Duchi, Michael I Jordan, Martin J Wainwright, Andre Wibisono. ゼロ階凸最適化における最適レート : 2つの関数評価のべき乗. IEEE Transactions on Information Theory, 61(5):2788–2806, 2015.

E. フェルミ 「最小問題の数値解法」技術報告書、ロスアラモス科学研究所、ロスアラモス、NM、1952年11月。

アブラハム・D・フラックスマン、アダム・タウマン・カライ、H・ブレンダン・マクマハン。バンディット法におけるオンライン凸最適化設定: 勾配なしの勾配降下法。arXiv プレプリント cs/0408007, 2004 年。

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, Linli Xu. Diffformer: テキスト生成のための埋め込み空間での拡散モデルを強化します。 arXiv プレプリント arXiv:2212.09412, 2022年。

Ishaan GulrajaniとTatsunori B Hashimoto. 尤度ベース拡散言語モデル. Advances in Neural Information Processing Systems, 36, 2024年。

ライア・ハドセル、ドゥシャント・ラオ、アンドレイ・A・ルス、ラズヴァン・パスカヌ。変化を受け入れる : ディープニューラルネットワークにおける継続学習. 認知科学の動向, 24(12):1028–1040, 2020.

ハン・シゼウエン、鄭黄傑、周明源。カード: 分類および回帰拡散モデル。進歩神経情報処理システム, 35:18100–18115, 2022年。

ジェフリー・ヒントン 「フォワード・フォワードアルゴリズム : 予備的研究」arXivプレプリントarXiv:2212.13345 2022年。

ジェフリー・E・ヒントン、サイモン・オシンドロ、イエー・ワイ・テ。ディープ・ピリーフ・ネットのための高速学習アルゴリズム。ニューラルネットワーク計算, 18(7):1527–1554, 2006年。

ジョナサン・ホー、アジェイ・ジェイン、ピーター・アビール。拡散確率モデルのノイズ除去。神経情報科学の進歩処理システム, 33:6840–6851, 2020年。

Vincent Hu, Di Wu, Yuki Asano, Pascal Mettes, Basura Fernando, Bjorn Ommer, Cees Snoek. 「フロアマッチングを用いた条件付きテキスト生成のための、わずかなサンプリングステップで実現」。計算言語学協会ヨーロッパ支部第18回会議論文集（第2巻 : 短報）、pp. 380–392, 2024年。

Shauharda KhadkaとKagan Tumer. 強化学習における進化誘導方策勾配。Advances in Neural Information Processing Systems, 31, 2018年。

Semin Kim, Jaehoon Yoo, Jinwoo Kim, Yeonwoo Cha, Saehoon Kim, Seunghoon Hong. ペアデータを用いたニューラルコードのシミュレーションフリー学習. Advances in Neural Information Processing Systems, 37:60212–60236, 2025.

Diederik Kingma, Tim Salimans, Ben Poole, Jonathan Ho. 変分拡散モデル. ニューラル情報処理システムの進歩, 34:21696–21707, 2021.

Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, Yoshua Bengio. 差異ターゲット伝播. 機械学習とデータベースにおける知識発見 : ヨーロッパ会議、ECML PKDD 2015、ポルトガル、ポルト、2015年9月7日～11日、議事録、パート 15, pp. 498–515. Springer, 2015年。

Honglak Lee, Roger Grosse, Rajesh Ranganath, Andrew Y. Ng. 階層的表現のスケラブルな教師なし学習のための畳み込みディープヒープネットワーク. 第26回国際機械学習会議ICML '09の議事録, pp. 609–616, ニューヨーク, ニューヨーク, 米国, 2009年. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553453. URL <https://doi.org/10.1145/1553374.1553453>.

Timothy P. Lillicrap, Daniel Counden, Douglas Blair Tweed, Colin J. Akerman. ランダムシナプスフィードバック重みは、深層学習における誤差逆伝播をサポートする。Nature Communications, 7, 2016年。URL <https://api.semanticscholar.org/CorpusID:10050777>。

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, Matt Le. 生成モデリングのためのフローマッチング。arXivプレプリント arXiv:2210.02747, 2022年。

Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O. Hero III, Pramod K. Varshney. 信号処理と機械学習におけるゼロ次最適化入門: 原理、最近の進歩、そして応用。IEEE信号処理マガジン, 37(5):43–54, 2020年。

アーサー・B・マークマンとエリック・ディートリッヒ「表象の擁護」認知心理学, 40(2):138–171, 2000年。

Yurii Nesterov and Vladimir G. Spokoiny. 凸関数のランダム勾配フリー最小化。計算数学, 17:527 – 566, 2015年。

Mengye Ren, Simon Kornblith, Renjie Liao, Geoffrey Hinton. 局所損失を考慮した順方向勾配のスケリング。arXivプレプリント arXiv:2210.03310, 2022年。

David E Rumelhart, Geoffrey E Hinton, Ronald J Williams. 誤差逆伝播による表現の学習. Nature, 323(6088):533–536, 1986.

ティム・サリマンス、ジョナサン・ホー、シー・チェン、シモン・シンドール、イリヤ・スツケヴァー。スケラブルな代替手段としての進化戦略強化学習へ。arXivプレプリント arXiv:1703.03864, 2017。

トム・ショール、ダイアナ・ボルサ、ジョセフ・モダイル、ラズヴァン・パスカヌ。光線干渉: 深宇宙におけるプラトンの発生源強化学習, 2019年。

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, Surya Ganguli. 非平衡熱力学を用いた深層教師なし学習。Francis Bach, David Blei (編), Proceedings of the 32nd International Conference on Machine Learning, Volume 37 of Proceedings of Machine Learning Research, pp. 2256–2265, Lille, France, 2015年7月7日–9日. PMLR.

Yang Song, Conor Durkan, Jain Murray, Stefano Ermon. スコアベース拡散モデルの最大尤度学習。M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, J. Wortman Vaughan (編), 『Advances in Neural Information Processing Systems』第34巻. Curran Associates, Inc., 2021年。

Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, Jeff Clune. 「ディープニューロエボリューション: 遺伝的アルゴリズムは、強化学習のためのディープニューラルネットワークのトレーニングにおける競争力のある代替手段である」ArXiv, abs/1712.06567, 2017年。

Alexander Tong, Nikolay Malkin, Guillaume Hugué, Yanlei Zhang, Jarrod Rector-Brooks, Kilian Fatras, Guy Wolf, Yoshua Bengio. ミニバッチ最適輸送を用いたフローベース生成モデルの改良と一般化。arXivプレプリント arXiv:2302.00482, 2023。

ヴァージニア・トルツォン. 多方向探索アルゴリズムの収束について. SIAM Journal on Optimization, 1(1): 123–145, 1991年。

ダーン・ヴィアストラ、トム・ショール、トビアス・グラスマツハース、イー・スン、ヤン・ペータース、ユルゲン・シュミットフーバー。自然進化戦略. 機械学習研究ジャーナル, 15(1):949–980, 2014。

Matthew D. ZeilerとRob Fergus. 畳み込みネットワークの可視化と理解. David Fleet, Tomas Pajdla, Bernt Schiele, Tinne Tuytelaars (編), Computer Vision – ECCV 2014, pp. 818–833, Cham, 2014. シュプリンガー・インターナショナル・パブリッシング。ISBN 978-3-319-10590-1。

NOPROPの訓練目標の導出

完全性を期すために、[Sohl-Dickstein et al. \(2015\)](#)および[Kingma et al. \(2021\)](#)に厳密に従って、NoProp-DT および NoProp-CT のトレーニング目標の導出を含めます。

A.1式4の導出

$$\text{対数} p(y|x) = \text{対数} p((z_t)_{t=0}^T, y|x) d(z_t)_{t=0}^T \quad (20)$$

$$= \text{対数} \frac{p((z_t)_{t=0}^T, y|x) q((z_t)_{t=0}^T | y, x)}{q((z_t)_{t=0}^T | y, x)} d(z_t)_{t=0}^T \quad (21)$$

$$= \log \mathbb{E}_{q((z_t)_{t=0}^T | y, x)} \frac{p((z_t)_{t=0}^T, y|x)}{q((z_t)_{t=0}^T | y, x)} \quad (22)$$

$$\geq \mathbb{E}_{q((z_t)_{t=0}^T | y, x)} \log p((z_t)_{t=0}^T, y|x) - \log q((z_t)_{t=0}^T | y, x). \quad (23)$$

最後のステップはジェンセンの不等式から導かれ、 $\log p(y|x)$ の下限値が得られます。この下限値は一般に証拠下限値 (ELBO) と呼ばれます。

A.2 $q(z_{t-1}|z_t)$ から $q(z_t|y)$

$q(z_t|y)$ と $q(z_t|z_s)$ の式を導くために、再パラメータ化のトリックを使います。再パラメータ化のトリックを使うと、分布からサンプリングされた確率変数を、ノイズ変数の決定論的関数で書き直すことができます。具体的には、ガウス分布の確率変数 $z \sim \text{Nd}(z|\mu, \sigma^2)$ の場合、以下のように再パラメータ化できます。

$$z = \mu + \sigma \epsilon, \quad \epsilon \sim \text{Nd}(\epsilon|0, 1). \quad (24)$$

{ t , $t+1$ } を選びます。 $(z_{t=0}^T, y, y) \sim \text{Nd}(0, 1)$ である。すると、任意の $0 \leq t < s \leq T$ に対して、任意のサンプル $z_t \sim q(z_t|z_s)$ に対して、以下のように表される。

$$z_t = \sqrt{a_t} z_{t+1} + \sqrt{1 - a_t} \epsilon_{t+1} \quad (25)$$

$$\sqrt{a_{t+1} z_{t+2} + 1 - a_{t+1}} + \sqrt{1 - a_{t+1}} \epsilon_{t+1} = \sqrt{a_t} \left(\sqrt{a_{t+1} z_{t+2} + 1 - a_{t+1}} + \sqrt{1 - a_{t+1}} \epsilon_{t+1} \right) \quad (26)$$

$$= \sqrt{a_t a_{t+1} z_{t+2} + a_t - a_t a_{t+1}} = \sqrt{a_{t+2}} z_{t+2} + \sqrt{1 - a_{t+2}} \epsilon_{t+1} \quad (27)$$

$$a_{t+1} z_{t+2} + (a_t - a_t a_{t+1}) + (1 - a_t) z_{t+2} = \sqrt{a_{t+2}} a_{t+1} z_{t+2} + 1 \quad (28)$$

$$- a_t a_{t+1} z_{t+2} \quad (29)$$

$$= \frac{a_{t+2} z_{t+2} + 1 - a_{t+2}}{a_{t+2}} = \frac{a_{t+2} z_{t+2} + 1 - a_{t+2}}{a_{t+2}} \quad (30)$$

$$= \frac{a_{t+2} z_{t+2} + 1 - a_{t+2}}{a_{t+2}} = \frac{a_{t+2} z_{t+2} + 1 - a_{t+2}}{a_{t+2}} \quad (31)$$

したがって、

$$q(z_t|z_s) = \text{Nd}(z_t | \frac{a_t}{a_s} z_s, 1 - \frac{a_t}{a_s}). \quad (32)$$

特に、

$$q(z_{t-1}|z_t) = \text{Nd}(z_{t-1} | \sqrt{a_{t-1} z_t}, 1 - a_{t-1}) \text{ です。} \quad (33)$$

同様に、

$$q(z_t|y) = \text{Nd}(z_t | \sqrt{a_t} y, 1 - a_t). \quad (34)$$

A.3 $q(z_t|z_{t-1}, y)$

ベイズの定理を適用すると、事後密度が得られる。

$$q(z_t|z_{t-1}, y) \propto q(z_t|y)q(z_{t-1}|z_t) \quad (35)$$

$$\propto \text{Nd}(z_t | \sqrt{\alpha_t} \text{tuy}, 1 - \alpha_t) \text{Nd}(z_{t-1} | \sqrt{\alpha_{t-1}} \text{tuy}, 1 - \alpha_{t-1}) \quad (36)$$

$$\propto \exp - \frac{1}{2(1 - \alpha_t)} (z_t - \sqrt{\alpha_t} \text{tuy})^T (z_t - \sqrt{\alpha_t} \text{tuy}) \quad (37)$$

$$- \frac{1}{2(1 - \alpha_{t-1})} (z_{t-1} - \sqrt{\alpha_{t-1}} \text{tuy})^T (z_{t-1} - \sqrt{\alpha_{t-1}} \text{tuy}) \quad (38)$$

$$\propto \exp - \frac{1}{2} (z_t - \mu_t(z_{t-1}, y))^T (z_t - \mu_t(z_{t-1}, y)) \cdot 2ct \quad (39)$$

したがって $q(z_t|z_{t-1}, y) = \text{Nd}(z_t | \mu_t(z_{t-1}, y), ct)$ となる。ここで

$$\mu_t = \frac{\sqrt{\alpha_t} \text{tuy} (1 - \alpha_{t-1}) + \sqrt{\alpha_{t-1}} \text{tuy} (1 - \alpha_t)}{\sqrt{\alpha_t} (1 - \alpha_{t-1}) + \sqrt{\alpha_{t-1}} (1 - \alpha_t)} z_{t-1}, \quad (40)$$

$$ct = \frac{1}{2} \left(\frac{1}{1 - \alpha_t} + \frac{1}{1 - \alpha_{t-1}} \right). \quad (41)$$

A.4 NOPROP-DTとNOPROP-CTの目的

付録A.1で導出したELBOから出発して、

$$\log p(y|x) \geq \mathbb{E}_q \left(\sum_{t=0}^T \log \frac{p(z_t | z_{t-1}, y, x)}{q(z_t | z_{t-1}, y, x)} \right) \quad (42)$$

$$\stackrel{\text{ログ}}{=} \mathbb{E}_q \left(\sum_{t=0}^T \log \frac{p(z_0) \prod_{t=1}^T p(z_t | z_{t-1}, x) p(y | z_T)}{q(z_T | y) \prod_{t=1}^T q(z_t | z_{t-1}, y)} \right) \quad (43)$$

$$= \mathbb{E}_q \left(\sum_{t=0}^T \log \frac{p(z_0) p(y | z_T)}{q(z_T | y)} + \sum_{t=1}^T \log \frac{p(z_t | z_{t-1}, x)}{q(z_t | z_{t-1}, y)} \right) \quad (44)$$

$$\stackrel{\text{ログ}}{=} \mathbb{E}_q \left(\sum_{t=0}^T \log \frac{p(z_0) p(y | z_T)}{q(z_T | y)} + \sum_{t=1}^T \log \frac{p(z_t | z_{t-1}, x) q(z_t | z_{t-1}, y)}{q(z_t | z_{t-1}, y) q(z_t | z_{t-1}, y)} \right) \quad (45)$$

$$= \mathbb{E}_q \left(\sum_{t=0}^T \log \frac{p(z_0) p(y | z_T) q(z_T | y)}{q(z_T | y) q(z_0 | y)} + \sum_{t=1}^T \log \frac{p(z_t | z_{t-1}, x)}{q(z_t | z_{t-1}, y)} \right) \quad (46)$$

$$= \mathbb{E}_q \left(\sum_{t=0}^T \log \frac{p(z_0) p(y | z_T)}{q(z_0 | y)} + \sum_{t=1}^T \log \frac{p(z_t | z_{t-1}, x)}{q(z_t | z_{t-1}, y)} \right) \quad (47)$$

$$= \mathbb{E}_q \left(\sum_{t=0}^T [\log p(y | z_T)] + \mathbb{E}_q \left(\sum_{t=0}^T \log q(z_0 | y) \right) + \sum_{t=1}^T \log \frac{p(z_t | z_{t-1}, x)}{q(z_t | z_{t-1}, y)} \right) \quad (48)$$

$$= \mathbb{E}_q \left(\sum_{t=0}^T [\log p(y | z_T)] + \mathbb{E}_q \left(\sum_{t=0}^T \log q(z_0 | y) \right) + \sum_{t=1}^T \log \frac{p(z_t | z_{t-1}, x)}{q(z_t | z_{t-1}, y)} \right) \quad (49)$$

$$= \mathbb{E}_q \left(\sum_{t=0}^T [\log p(y | z_T)] - \text{DKL}(q(z_0 | y) \parallel p(z_0)) - \sum_{t=1}^T \mathbb{E}_q \left(\sum_{t=1}^T [\text{DKL}(q(z_t | z_{t-1}, y) \parallel p(z_t | z_{t-1}, x))] \right) \right) \quad (50)$$

$q(z_t|z_{t-1}, y)$ と $p(z_t|z_{t-1}, x)$ は両方ともガウス分布であり、 $q(z_t|z_{t-1}, y) = \text{Nd}(z_t|\mu_t(z_{t-1}, uy), \sigma_t^2)$ と $p(z_t|z_{t-1}, x) = \text{Nd}(z_t|\mu_t(z_{t-1}, u - \theta_t(z_{t-1}, x)), \sigma_t^2)$ であるため、KL ダイバージェンスは閉じた形式で利用できます。

$$\text{DKL}(q(z_t|z_{t-1}, y)p(z_t|z_{t-1}, x)) = \mu_t(z_{t-1}, u - \theta_t(z_{t-1}, x)) - \mu_t(z_{t-1}, uy) \quad (51)$$

$$= \frac{\sigma_t^2(1 - \alpha_t - 1)2(1 - \alpha_t)}{\sigma_t^2(1 - \alpha_t - 1)(1 - \alpha_t)} \parallel u - \theta_t(z_{t-1}, x) - uy \parallel^2 \quad (52)$$

$$= \frac{\sigma_t^2(1 - \alpha_t)(1 - \alpha_t - 1)u - \theta_t(z_{t-1}, x) - uy}{\sigma_t^2(1 - \alpha_t - 1)(1 - \alpha_t - 1)} \parallel u - \theta_t(z_{t-1}, x) - uy \parallel^2 \quad (53)$$

$$= \frac{1}{2} \parallel u - \theta_t(z_{t-1}, x) - uy \parallel^2 \quad (54)$$

$$= \frac{1}{2} \parallel u - \theta_t(z_{t-1}, x) - uy \parallel^2 \quad (55)$$

$$= \frac{1}{2} (\text{SNR}(t) - \text{SNR}(t-1)) \parallel u - \theta_t(z_{t-1}, x) - uy \parallel^2 \quad (56)$$

$p_{\theta_{\text{out}}}(y|z_T)$ を使って $p(y|z_T)$ を推定すると、ELBO は次のようになります。

$$\mathbb{E}_q(z_T|y) [\log p_{\theta_{\text{out}}}(y|z_T)] - \text{DKL}(q(z_0|y) \parallel p(z_0)) - \frac{1}{2} \sum_{t=1}^T \mathbb{E}_q(z_t|y) (\text{SNR}(t) - \text{SNR}(t-1)) \parallel u - \theta_t(z_{t-1}, x) - uy \parallel^2 \quad (57)$$

合計の T 項すべてを計算する代わりに、不偏推定値で置き換えます。これは

$$\mathbb{E}_q(z_T|y) [\log p_{\theta_{\text{out}}}(y|z_T)] - \text{DKL}(q(z_0|y) \parallel p(z_0)) - \frac{T}{2} \mathbb{E}_t \mathbb{U}(1, T, q(z_{t-1}|y)) (\text{SNR}(t) - \text{SNR}(t-1)) \parallel u - \theta_t(z_{t-1}, x) - uy \parallel^2 \quad (58)$$

式8にハイパーパラメータ η を追加すると、離散時間モデルにおいて NoProp 目的関数が得られます。ただし、実験ではランダムな値をサンプリングするのではなく、 t の値を1から T まで反復処理することを選択していることに注意してください。詳細はアルゴリズム1を参照してください。

連続時間の場合、 $t \in (0, 1)$ の範囲にスケールされているので、 $\tau = 1/T$ とする。NoProp 目的関数における SNR に関する項を次のように書き直すと、

$$\frac{1}{2} \mathbb{E}_t \mathbb{U}[0, 1] \frac{\text{SNR}(t+\tau) - \text{SNR}(t)}{\tau} \parallel u - \theta(z_t, x, t) - uy \parallel^2 \quad (59)$$

$T \rightarrow \infty$ のとき、これは

$$\frac{1}{2} \mathbb{E}_t \mathbb{U}[0, 1] \text{SNR}'(t) \parallel u - \theta(z_t, x, t) - uy \parallel^2 \quad (60)$$

ここでも、追加のハイパーパラメータ η を使用すると、式9の連続時間の場合の NoProp 目的関数が得られます。

B NOPROP-CTの訓練可能なノイズスケジュール

Kingma et al. (2021) および Gulrajani & Hashimoto (2024) に従って、信号対雑音比 (SNR) をパラメータ化する。

として

$$\text{SNR}(t) = \exp(-\gamma(t)), \quad (65)$$

ここで $\gamma(t)$ はノイズ減衰率を決定する学習可能な関数である。我々の定式化との整合性を確保するため、 $\gamma(t)$ は t について単調減少でなければならない。

$\gamma(t)$ はニューラルネットワークベースのパラメータ化を用いて実装します。具体的には、単位区間に正規化された中間関数 $\gamma^-(t)$ を定義します。

$$\gamma^-(t) = \frac{\gamma_-(t) - \gamma_-(0)}{\gamma_-(1) - \gamma_-(0)}, \quad (66)$$

ここで $\gamma_-(t)$ は重みが正に制限された2層ニューラルネットワークとしてモデル化されます。

$\gamma(t)$ が t とともに減少するという本論文の定式化に沿うように、以下のように定義する: $\gamma(t) = \gamma_0 + (\gamma_1 - \gamma_0)(1 - \gamma^-(t))$

$$(67) \quad \text{ここで、} \gamma_0 \text{ と } \gamma_1 \text{ はノイズスケジュールの学習可能な端点である。最終的に、ノイズスケジュール } \alpha^{-t} =$$

$\sigma(-\gamma(t))$ を得る。

プレプリント。

データセット	方法	バッチサイズ	エポック	最適化	学習率	重み	減衰	タイムステップ η
MNIST	バックプロパゲーション 128	100	アダム W	0.001	100	アダム W	0.001	10
	ノードロップ-DT 128	0.001	2 アダム W	0.001	100	アダム	0.001	10
	副次128	0.001	100 アダム	0.001			0.001	1000
	ノードロップ-CT 128			0.001			0.001	1000
	ノードロップ-FM 128			0.001			0.001	1000
CIFAR-10	バックプロパゲーション 128	150	アダムW	0.001	150	アダムW	0.001	10
	ノードロップ-DT 128	0.001	4 アダムW	0.001	500	アダム	0.001	10
	副次128	0.001	500 アダム	0.001			0.001	1000
	ノードロップ-CT 128			0.001			0.001	1000
	ノードロップ-FM 128			0.001			0.001	1000
CIFAR-100	バックプロパゲーション 128	150	アダムW	0.001	150	アダムW	0.001	10
	ノードロップ-DT 128	0.001	4 アダムW	0.001			0.001	10
	副次128			0.001			0.001	1000
	ノードロップ-CT 128	1000	アダム			0.001	0.001	1000
	ノードロップ-FM 128	1000	アダム			0.001	0.001	1000

表3:実験の詳細。 η は式8と式9のハイパーパラメータです。

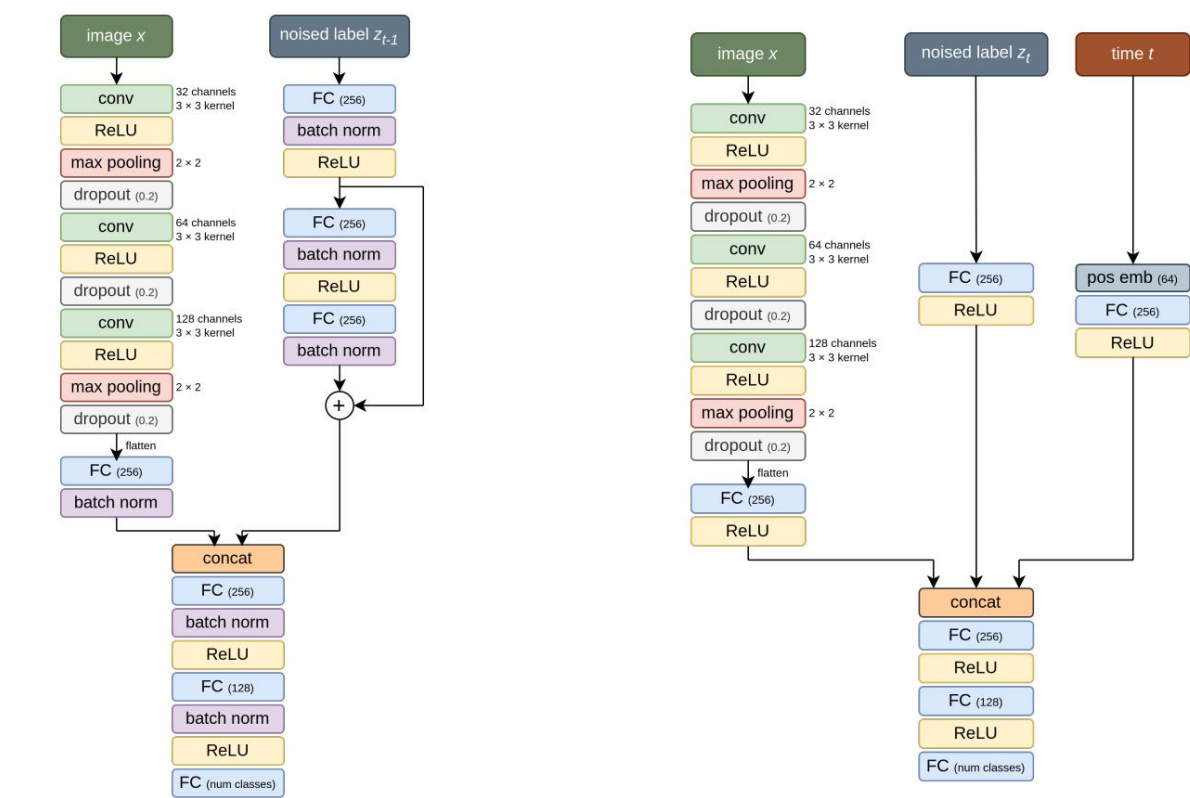


図6: クラスの埋め込み次元が画像の次元と異なる場合の学習に使用されるモデル。左:

離散時間の場合のモデル。右: 連続時間の場合のモデル。conv: 畳み込み層。FC: 完全

接続層 (括弧内の数字は単位を示す)。concat: 連結。pos emb: 位置埋め込み

(括弧内の数字は時間埋め込み次元を示す)。クラス埋め込み次元が

画像次元では、ノイズラベルと画像は各モデルで連結される前に同じ方法で処理されます。

バッチ正規化は連続時間モデルには含まれていないことに注意してください。

プレプリント。

アルゴリズム1 NoProp-DT (トレーニング)

要件: T拡散ステップ、データセット{(xi、 yi)}、θout、ノイズスケ北 i=1, バッチサイズB、ハイパーパラメータη、埋め込み行列WEmbed、パラメータ

t=1 から T ま t=1, ジュール{αt} t=0

での各ミニバッチ B C

{(xi , yi)}に対して、各 (xi , yi) ∈ Bに対して、 {θt}を北 サイズBのi=1

計算し、ラベル埋め込みuyi =

{WEmbed}yiを取得します。

zt,i Nd(zt,i|√ α^-tuy,i, 1 - α^-t)をサンプルします。

終わりのために

損失関数を計算します。

$$\begin{aligned} \text{左} = & \frac{1}{B} \sum_{i \in B} [-\log p_{\theta_{\text{out}}}(y_i | z_{T,i})] \\ & + \frac{1}{B} \sum_{i \in B} \text{DKL}(q(z_0 | y_i) \parallel p(z_0)) \\ & + \eta \frac{T}{2B} \sum_{i \in B} (\text{SNR}(t) - \text{SNR}(t-1)) \parallel u^{\theta_t}(z_{t-1,i}, x_i) - u_{y_i} \parallel^2. \end{aligned} \tag{61}$$

勾配ベースの最適化を使用してθt、 θout、 WEmbedを更新します。

アルゴリズム2 NoProp-CT (トレーニング)

要件:データセット {(xi , yi)}北 i=1, バッチサイズB、ハイパーパラメータη、埋め込み行列WEmbed、パラメータθ、 θout、ノイズ

各ミニバッチ B C {(xi , yi)}に対し

てα^-t=α(-γψ(t))をスケジュールし、各 (xi , yi)北 サイズBの場合、 i=1

∈ Bに対してラベル埋め込みuyi

= {WEmbed}yiを取得します。

ti U(0, 1)のサンプル。

zti,i Nd(zti,i|√ α^-tiuy,i, 1 - α^-ti)をサンプルします。

終わりのために

損失関数を計算します。

$$\begin{aligned} L = & \frac{1}{B} \sum_{i \in B} [-\log p_{\theta_{\text{out}}}(y_i | z_{1,i})] \\ & + \frac{1}{B} \sum_{i \in B} \text{DKL}(q(z_0 | y_i) \parallel p(z_0)) \\ & + \eta \frac{1}{2B} \sum_{i \in B} \text{SNR}'(t_i) \parallel u^{\theta}(z_{t_i,i}, x_i) - u_{y_i} \parallel^2. \end{aligned} \tag{62}$$

勾配ベースの最適化を使用してθ、 θout、 ψ、 WEmbedを更新します。

プレプリント。

アルゴリズム3 NoProp-FM (トレーニング)

要件: データセット $\{(x_i, y_i)\}_{i=1}^N$, バッチサイズ B , 埋め込み行列 W , パラメータ θ, θ_{out}
各ミニバッチ $B \subset \{(x_i, y_i)\}$ について、各 (x_i, y_i) (サイズ B の場合、 $i=1$)
 $\in B$ に対してラベル埋め込み $u_{y_i} =$
 $\{W \text{Embed}\} y_i$ を取得し、 $z_{1,i} = u_{y_i}$ に設定します。
 $z_{0,i} \sim \text{Nd}(z_{0,i}|0, 1)$ をサンプルします。
 $t_i \sim U(0, 1)$ のサンプル。
 $z_{t_i,i} \sim \text{Nd}(z_{t_i,i} | t_i z_{1,i} + (1 - t_i) z_{0,i}, \sigma^2)$ をサンプルします。
 終わりのために
 損失関数を計算します。

$$L = \frac{1}{B} \sum_{i \in B} \|v_\theta(z_{t_i,i}, x_i, t_i) - (z_{1,i} - z_{0,i})\|^2. \tag{63}$$

W Embed に学習可能なパラメータがある場合、各 (x_i, y_i)
 $\in B$ に対して外挿線形推定値 $z_{-1,i}$
 $= z_{t_i,i} + (1 - t_i)v_\theta(z_{t_i,i}, x_i, t_i)$ を計算します。
 終わりのために
 損失関数を変更します。

$$L \leftarrow L - \frac{1}{B} \sum_{i \in B} \log \frac{p_{out}(y_i | z_{-1,i})}{p_{out}(y_i | z_{t_i,i})} \tag{64}$$

end if 勾配ベースの最適化を使用して θ, θ_{out}, W Embed を更新します。end for
