

Deep Learning におけるコストを考慮した Dropout 率制御に関する検証

玉城 翔† 當間 愛晃‡ 赤嶺 有平‡ 山田 孝治‡ 遠藤 聡志‡
 † 琉球大学大学院理工学研究科情報工学専攻 ‡ 琉球大学工学部情報工学科

1 はじめに

1.1 研究背景

2012 年頃から機械学習の分野で、深層学習という言葉が流行りだしてきた。それは、各分野のコンテストにおいて深層学習を使った技術が、これまでの技術を圧倒する精度を誇って優勝してきたことにある。深層学習とは、機械学習の技術であるニューラルネットを基本とした機械学習アルゴリズムである。ニューラルネット自体の歴史は古く、始まりは 1940 年代からであり、様々な提案・改良が成されてきた。その為、ニューラルネットには様々な技術・手法が存在する。深層学習に限った話でも、それは例外ではない。深層学習の中で主に目立った手法としては、事前学習、畳み込みニューラルネット、Rectified Linear Unit (ReLU)、Dropout といったものが挙げられる。

この中で我々は、Dropout に注目した。ニューラルネットの学習における問題点の 1 つに過学習というものがある。Dropout ではこの過学習を解消することによる、ニューラルネットの汎化能力の向上が期待できる。この Dropout において経験的観点でのパラメータ設定が未だに通例となっているため、最適な Dropout 率のパラメータ設定方法が確立されていない。パラメータ設定によっては、学習コストが高くなることも想定されるが、問題の複雑さ、用意したニューロン数、活性化関数の種類等の様々なモデルに応じて適切な Dropout 率のパラメータ設定方法があると考えられる。そこで、我々はニューラルネットにおける評価関数の値を用い、最適な Dropout 率の設定が可能かどうかの検証をする。

1.2 研究内容

図 1(左) に示されるのが代表的な階層型ニューラルネットのモデルである。Dropout[1] は、ニューラルネットの学習時に行われる。学習時に Dropout 率を指定し、Dropout 率に従ってニューラルネットモデルの中間層のユニットを確率的に落とす。図 1(右) の場合は、中間層のユニット総数が 10、Dropout 率 0.5 のモデルの一例である。また、テスト時はすべてのユニットが稼働するが、その際、中間層すべての重みに Dropout 率を掛けることによって、ユニット同士の繋がりを抑制する。この Dropout に関する研究に、Li ら [2] の DropConnect と、Lei ら [3] の Adaptive dropout がある。Li らの研究では、Dropout 対象をユニットではなく、重みを掛ける部分としたものである。Lei らの研究では、Dropout

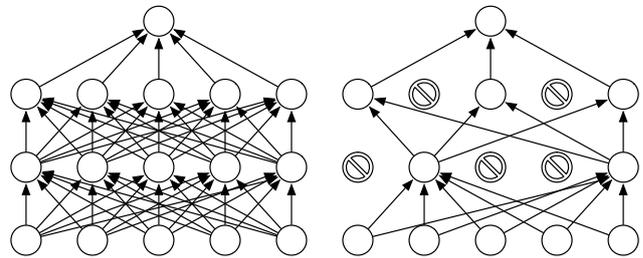


図 1: (左) 中間層 2 のニューラルネットモデル (右) Dropout 使用時のニューラルネットモデル。斜線ユニットは Dropout 適用ユニット

を行うニューラルネットモデルの他に、Dropout 率を求めるためのニューラルネットモデルを用意し、入力情報に依存した Dropout 率の算出を行うものである。我々は、適切な Dropout 率の設定方法の研究ということで、Lei らの研究に近いものとなるが、Lei らの研究では Dropout 率を求めるためにニューラルネットモデルを他に用意しているため、学習コストが高くなっている。我々の研究では、学習毎に得られる評価関数の値を利用するパターンと、学習毎に得られるエラー率の値を利用するパターンの 2 パターンで、学習毎の最適な Dropout 率の算出を試みる。この場合、既存のパラメータを利用しているため、Lei らの研究と比べて学習コストは低くなっている。

評価関数の値を利用するパターンでは、学習始めの段階で Dropout 率を高く設定することにより、始めは弱い学習を行う。学習が進む毎に Dropout 率を徐々に低く設定することにより、後半は強い学習を行う。エラー率の値を利用するパターンでは、逆に学習始めの段階では Dropout 率を低く設定することにより、始めは強い学習を行う。学習が進む毎に Dropout 率を徐々に高く設定することにより、後半は弱い学習を行う。この 2 パターンの Dropout 率変更の際に、それぞれ評価関数、エラー率の値を利用する。評価関数、エラー率の収束具合が小さくなってきた場合にそれぞれの Dropout 率を変更する。

2 実験

2.1 実験目的

今回の実験では、従来の Dropout と提案手法とを比べて、エラー率が下がるかどうかを見る。従来手法の場合、Dropout 率は学習始めから終わりまで 0.5[4] で固定である。そこで、我々の提案手法と比べることにより、学習の進行具合による Dropout 率の制御が有効かどうかを検証した。

A Parameter Control Simulation on Dropout Rate using the Cost in Deep Learning

† Kakeru TAMASHIRO · Information Engineering, Graduate School of Science and Engineering, University of the Ryukyus

‡ Naruaki TOMA · Yuhei AKAMINE · Koji YAMADA · Satoshi ENDO · Department of Information Engineering, Faculty of Engineering, University of the Ryukyus

2.2 実験設計

今回の実験では、Omidら [5] が提供している YouTube Multiview Video Games Dataset を使用した。

表 1: Youtube Multiview Video Games Dataset

データセット	データ数
学習用	72300 件
テスト用	8900 件
特徴ベクトル	次元数
画像	647
クラスラベル	ラベル数
ゲームタイトル	30 件

モデルは中間層 3 つのニューラルネットとし、入力層のユニット数は、入力画像の次元数が 647 次元なので 647 とし、出力層のユニット数は、ゲームタイトル 30 件を分類するので 30 とした。中間層のユニット数は、すべての層で 500 とした。学習には SGD (stochastic gradient descent method) 法を使い、学習回数は 1 万回とした。バッチサイズは 100、初期学習率は 1、学習スケール率は 0.9998、活性化関数は tanh (ハイパボリックタンジェント) とした。今回は 4 つの Dropout パターンを用意し比較した。1 つ目は Dropout なしパターン。2 つ目は Dropout 率を 0.5 に固定したパターン。3 つ目は Dropout 率を徐々に下げていくパターン。初期 Dropout 率を 0.9 に設定し、評価関数の収束具合によって Dropout 率を下げていく。Dropout 率下限は 0.5。4 つ目は Dropout 率を徐々に上げていくパターン。初期 Dropout 率を 0.1 に設定し、エラー率の収束具合によって Dropout 率を上げていく。Dropout 率上限は 0.5。

評価関数, エラー率の収束具合の算出方法は、window 幅を設定し、window 幅分の評価関数, エラー率の平均を取る。window 幅毎に Dropout 率変更点を取り、現在学習時点での window 幅平均と、window 幅 1 つ前分の平均の 2 つの比率を出す。この比率が閾値を下回った場合を収束が停滞していると見て、Dropout 率を変更した。3 つ目のパターンでは、window 幅を 1000 とし、閾値を 1% とした。4 つ目のパターンでは、window 幅を 50 とし、閾値を 3% とした。

評価方法はテストデータを使い、学習したモデルで予測した結果で、間違った件数をテストデータ数で割った値をエラー率とした。

2.3 実験結果

図 2(左) は横軸を学習回数、縦軸をエラー率としたグラフである。一番エラー率が低かったのが Dropout 率を徐々に上げていくパターンである。Dropout 率を徐々に下げていくパターンが一番エラー率が高かったが、学習を進めていくと更にエラー率が下がっていく可能性が見られる。

図 2(右) は横軸を学習回数、縦軸を Dropout 率としたグラフである。この図から、Dropout 率を下げていくパターンでは、Dropout の特性である学習の遅さから、

学習回数を更に伸ばしてエラー率の変化具合を見ていく必要がある。Dropout 率を上げていくパターンでは、学習効率を上げるという点で、Dropout 固定のパターンよりも早くエラー率が低くなっている。しかしながら、後半になると学習効率が下がってしまっているのも見て分かる。

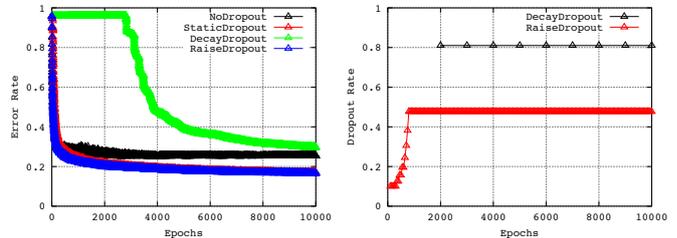


図 2: (左) 各パターンにおけるエラー率。NoDropout : Dropout なしパターン, StaticDropout : Dropout 率を 0.5 に固定したパターン, DecayDropout : Dropout 率を徐々に下げるパターン, RaiseDropout : Dropout 率を徐々に上げるパターン (右) Dropout 率の変化

3 現状と今後の課題

Dropout 率を下げていくパターンでは、学習回数を更に増やした実験をする必要がある。学習を続けていくことによって、最終的に Dropout 固定のパターンのエラー率よりも下回るかを検証していく。また、Dropout 率を上げていくパターンでは、Dropout の特性の学習の遅さを改善できる可能性を示唆した。しかしながら、学習後半においては、Dropout 率上限に達してしまい、Dropout 固定のパターンと似た学習を行ってしまったために、エラー率の差が縮まってしまった可能性がある。

参考文献

- [1] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", Journal of Machine Learning Research 15 (2014) 1929-1958.
- [2] Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, Rob Fergus, "Regularization of Neural Networks using DropConnect", JMLR W&CP 28 (3) : 1058-1066, 2013.
- [3] Lei Jimmy Ba, Brendan Frey, "Adaptive dropout for training deep neural networks", Advances in Neural Information Processing Systems 26 (NIPS 2013)
- [4] Pierre Baldi, Peter J Sadowski, "Understanding Dropout", NIPS Conference, 2013
- [5] Omid Madani, Manfred Georg, David A. Ross, "On Using Nearly-Independent Feature Families for High Precision and Confidence ", Machine Learning 92 457- 477, 30 May 2013