1A3-4

DNN 圧縮時のパラメータと圧縮後の精度, 大きさの関係

Relationship between Parameters for compressing DNNs and their Accuracy and Size

西上 良祐 Ryosuke Nishigami
岸田 脩平 Syuhei Kishida
村田 剛志 Tsuyoshi Murata

東京工業大学 情報理工学院 情報工学系
Department of Computer Science, School of Computing, Tokyo Institute of Technology

Deep Compression is one of the method for compressing DNNs. It requires parameters for compression, and relationship between the parameters and accuracy and size of compressed DNNs is not clear. In this paper, we compress six DNNs by simplified Deep Compression which we implemented. We investigate the relationship between parameters and accuracy and size of compressed DNN, and we found some criteria of setting parameters. We proposed a method for determining the parameter of weight sharing automatically.

1. はじめに

DNN(ディープニューラルネットワーク)とはディープラーニングに用いられる多層構造のニューラルネットワークである. モバイル端末や組み込みシステムなどのストレージが限られた環境でディープラーニングで得られた学習器を使おうとすると, その大きさがストレージに収まらなかったり, ストレージを圧迫してしまう. そこで精度を大幅に下げることなく DNN を圧縮する必要がある.

DNN の圧縮手法として Deep Compression [Song 16] が Song Han らによって提案された. この手法は DNN の枝刈り, DNN の重みの量子化と共有, ハフマン符号の 3 つの手法を組み合わせて DNN を圧縮する手法である. Deep Compression を用いる際には 2 つのパラメータが必要となる. 1 つ目は枝刈りを行う際の重みの閾値である . 2 つ目は重みの量子化と共有の際に用いる K-means 法のクラスタ数である. DNN を圧縮する際には「DNN の精度を落とすことなくできるだけ圧縮したい」などの圧縮の目的が存在すると考えられる. Deep Compression を用いてこの目的を満たすためには, パラメータを変えて圧縮を繰り返す必要がある. これらのパラメータと圧縮後の DNN の精度, 大きさとの関係についてはまだ十分な研究がなされていない.

そこで本稿では次の 2 つのことを行った. 1 つ目はパラメータを設定する基準を見つけるために, 6 つの DNN を圧縮し圧縮後の精度, 大きさとパラメータの関係に共通点がないか調べた. Deep Compression のプログラムで公開されているものはないので Deep Compression を簡略化した手法を実装し, 圧縮に用いた. 2 つ目は X-means 法 [Dan 00] を用いて重み共有のパラメータを自動で決定する方法を提案した. 提案手法を用いて DNN を圧縮した場合, Deep Compression を簡略化した手法を用いた場合と比べて圧縮後の精度, 大きさがどう異なるかを実験で調べた.

Deep Compression を簡略化した手法による圧縮実験から, パラメータを設定する際の基準を得ることができた. また提案手法による圧縮では, 圧縮後の精度, 大きさの両方で良い結果が得られないことがわかった.

2. 関連研究

連絡先: 西上良祐,東京工業大学 情報理工学院 情報工学系,
東京都目黒区大岡山 2-12-1 W8-59,
nishigami.r.aa@m.titech.ac.jp

本節では Deep Compression, および X-means 法について述べる.

2.1 Deep Compression

Deep Compression は DNN の枝刈り, DNN の重みの量子化と共有, ハフマン符号の 3 つの手法を組み合わせて DNN を圧縮する手法である.

(1) 枝刈り

DNN の枝刈りは DNN から一部の辺を取り除くことで DNN を圧縮する手法である. Deep Compression では辺の重みの絶対値が閾値よりも小さい辺を取り除いている. 辺を取り除いた後, 残った辺のみで誤差逆伝播法を行い DNN を再学習させている.

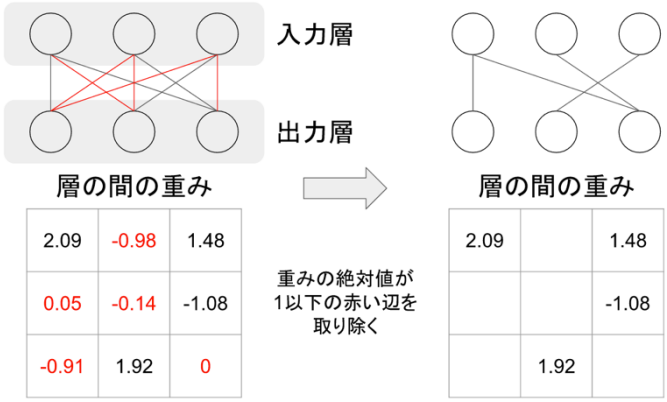

図 1: 枝刈りの例. この例では重みの絶対値が 1 以下の辺を取り除いている.

(2) 重みの量子化と共有

DNN の重みの量子化と共有は DNN の重みを表すのに必要なビット数を小さくすることで DNN を圧縮する手法である. 入力層と出力層の大きさをそれぞれ 3 とすると, 全結合の層の間の辺の重みは 3 行 3 列の行列で表現される. 重みを K-means 法を用いてクラスタリングし, 同じクラスタに属する全ての重みは共有の重みを持つものとする. 共有の重みはクラスタの重心である. 元の重みの代わりに, 重みの属するクラスタの番号と共有の重みを保持することで重みを表すのに必要なビット数を小さくしている. DNN の学習時には, 勾配降下法により共有の重みが調

整される. その際同じクラスタに属する重みの勾配をクラスタごとに合計し, それに学習率をかけた値をそれぞれの共有重みから引く.

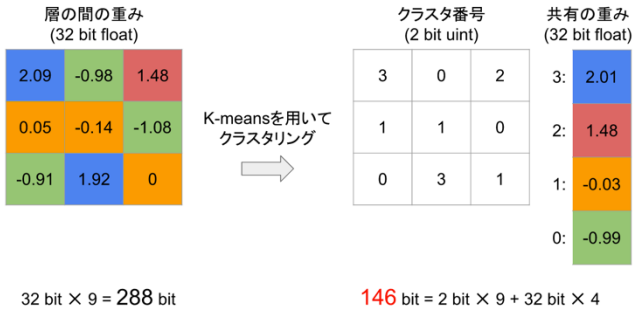

図 2: 重みの量子化と共有の例. この例では K-means 法を用いてクラスタ数 4 でクラスタリングを行っている. 圧縮前の重みのビット数が 288 ビットであり, 圧縮後の重みを表すのに必要なビット数は 146 ビットである.

(3) ハフマン符号

ハフマン符号 [David 52] はデータの可逆圧縮手法の一つである. この手法では出現頻度の高いデータに短いビット列を, 出現頻度の低いデータに長いビット列を割 り当てる. そうすることでデータ全体の符号化に使われるビット数を削減する手法である.

2.2 X-means 法

X-means 法 [Dan 00] は Dan Pelleg と Andrew Moore によって提案されたクラスタリング手法の一つである. 後に石岡により改良が加えられた手法 [石岡 06] が提案された. X-means 法はクラスタリング手法の一つである K-means 法を拡張したものである. K-means 法の入力と出力は以下の通りである.

- 入力: クラスタ数 k, 解析する n 個の p 次元データ$x_i(i = 1,2,\dots,n)$
- 出力: 入力された n 個のデータの属するクラスタ番号, 各クラスタの重心, 各クラスタに含まれるデータ数

K-means 法では初期状態として各データx_iを k 個のクラスタに振り分ける. この初期状態の決め方には K-means++法 [David 07] など様々な方法があるが, ここでは紹介にとどめておく. 初期状態のクラスタの重心 G_j (j = 1,2,…, k) を計算する. 重心は通常各データの算術平均で計算される. その後各 x_i と各 G_j の距離を計算しx_iを最も重心の近いクラスタに割り当て直す. この処理を全てのx_iのクラスタの割り当てが変化しなくなるまで繰り返す.

X-means 法の入出力は以下のとおりである.

- 入力: 解析する n 個の p 次元データ
- 出力: 入力された n 個のデータの属するクラスタ番号, 各クラスタの重心, 各クラスタに含まれるデータ数

K-means 法では入力としてクラスタ数が必要になるが, X-means 法ではその必要がない. X-means 法ではクラスタを, 2 分割が適当であると判断される限り, K-means 法を (k = 2 で) 用いて 2 分割を繰り返す. そうすること でクラスタ数を事前に決めることなくクラスタリングを行う. 二分割が適当であるかの判断基準は BIC [Schwarz 78] を用いる.

Deep Compression では重みの量子化と共有を行う際に, K-means 法によるクラスタ リングを行っている. しかし精度を落とさずに DNN を圧縮したい場合の最適なクラスタ数というのは各 DNN の重みの分布によって異なるはずである. そこでクラスタリングを行う際に K-means 法の代わりに X-means 法を用いれば, DNN の重みの分布によって自動で最適なクラスタ数を決定できるのではと考えた.

本稿では石岡により改良が加えられた手法を用いる. Pelleg と Moore による手法ではなく 石岡の手法を選んだ理由は以下のとおりである. Pelleg と Moore の手法では逐次分割されるクラスタの重心からの距離分散を一定と仮定している. 逐次分割のネストが深くなると分割対象のデータが少なくなり, それに伴い分散も一 般に小さくなる. 石岡の手法はこの分散の違いを考慮しているため Pelleg と Moore の手法より一般的な手法だと言えるからである.

3. 実験に用いる圧縮手法

本節では本実験で DNN を圧縮する際に用いる Deep Compression を簡略化した手法, および提案手法について述べる.

3.1 Deep Compression を簡略化した手法

Deep Compression を簡略化した手法では DNN の枝刈りと, DNN の重みの量子化と共有を用いて DNN を圧縮する. Deep Compression との違いは以下の通りである.

- ハフマン符号を用いない.
- 圧縮後の DNN を再学習させない.

Deep Compression を簡略化した手法の入出力は以下のとおりである. 圧縮時のパラメータである, 枝刈りの割合 r とクラスタ数 k については 4.1 節で述べる.

- 入力: DNN, 枝刈りの割合 r, クラスタ数 k
- 出力: 圧縮後の DNN を保持したバイナリファイル

3.2 提案手法

提案手法では DNN の重みの量子化と共有の際に行うクラスタリングに K-means 法の代わりに X-means 法を用いる. それに伴い入力としてクラスタ数 k が必要ではなくなる. その他の点で Deep Compression を簡略化した手法と異なる点はない. 提案手法の入出力は以下のとおりである.

- 入力: DNN, 枝刈りの割合 r
- 出力: 圧縮後の DNN を保持したバイナリファイル

4. 実験

本節では Deep Compression を簡略化した手法または提案手法を用いて DNN を圧縮する実験とその結果, および考察について述べる.

4.1 圧縮時のパラメータ

本実験では以下の 2 つのパラメータの値を変えて DNN を圧縮し, 圧縮後の DNN の精度, 大きさを計測した.

- 枝刈りの割合: r = {10, 20, …, 90} [%]
- K-means 法を用いる際のクラスタ数: k = $2^n - 1$ (提案手法では省略される)

本実験では, DNN の各層の間の辺から重みの絶対値の小さい順に r %の辺を取り除く. 取り除く辺の割合はどの層の間の辺においても r %である. Deep Compression では重みの絶対値の閾値がパラメータであるが, 重みの絶対値の閾値は DNN 間で共通の尺度とならないので枝刈りの割合をパラメータとした. Deep

Compression を簡略化した手法による圧縮では, どの層の間の辺の重みも, クラスタ数 k で K-means 法を用いてクラスタリングを行う. 提案手法では層の間の辺の重みごとにクラスタ数が自動で決定されるので, 各層の間ごとでクラスタ数が異なる可能性がある.

4.2 圧縮する DNN

本実験で圧縮した DNN は表 1 の 6 つである. LeNet は手書き数字の識別に用いられる DNN である. その他の 5 つは飛行機や猫などの画像を識別するための DNN である. DNN の精度は式(1)で定義される.

精度 = 正しく識別された画像の枚数/識別した画像の枚数 (1)

圧縮前の DNN の大きさは, 全ての重みとバイアスのビット数の合計である. バイアスとは層間で信号が伝わる際に信号に加えられる値のことである. 圧縮後の DNN の大きさは Deep Compression を簡略化した手法または提案手法により出力されるバイナリファイルの大きさである.

表 1: 圧縮した DNN と精度の計測に用いた画像の種類

DNN	圧縮前の大きさ	精度の計測に用いた画像
LeNet	1.7MB	MNIST
AlexNet	233MB	Image Net
NIN	29MB	Image Net
VGG-16	528MB	Image Net
GoogLeNet	27MB	Image Net
ResNet-50	97MB	Image Net

4.3 圧縮後の DNN の精度, 大きさ

本節ではパラメータと圧縮後の DNN の精度, 大きさの関係について説明する.

(1) DNN 間で共通した, パラメータと圧縮後の精度の関係

図 3 は ResNet-50 の圧縮後の精度のグラフを表している. まず Deep Compression を簡略化した手法による圧縮後の精度について述べる. 6 つの DNN では, クラスタ数を大きくしていっても, ある程度以上からは圧縮後の精度にほとんど影響はなくなった. ResNet-50 ではクラスタ数を 63 以上としても精度のグラフ間の差は最大でも 1 %だった. ResNet-50 ではクラスタ数を精度に影響がなくなる値まで大きくした時, 40 %の辺を取り除いても圧縮前と後で精度が 2%しか変わらなかった. 6 つの DNN では, クラスタ数を精度に影響がなくなる値まで大きくした時, 30 %の辺を取り除いても圧縮前と後で精度が最大でも 3 %しか変わらなかった. ResNet-50 では 50 %以上の辺を取り除くと圧縮後の精度の下がり方が大きくなり始めた. 6 つの DNN では, 圧縮後の精度の下がり方が大きくなり始める, 取り除く辺の割合の最小値は 40 %だった.

次に提案手法による圧縮後の精度について述べる. ResNet-50 では提案手法による圧縮後の精度は Deep Compression を簡略化した手法のグラフと比べて最大で 19 %精度が下がった. 6 つの DNN でも提案手法による圧縮後の精度は, Deep Compression を簡略化した手法と比べるとほとんどの場合で上回ることはなかった. この原因の一つとして次のことが考えられる. ResNet-50 において, 提案手法を用いて r=20 で圧縮した際のクラスタ数を, クラスタリング対象となった層間の重みごとで比べると最小の場合で 14 となった. 一部の層の間の辺の重みをクラスタリングする際にクラスタ数が小さくなることで, 圧縮前後の重みの値の差大きくなり, 圧縮後の精度も下がってしまうと考えられる.

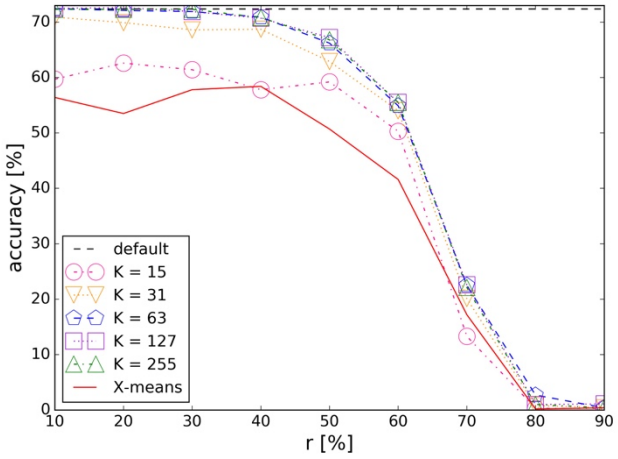

図 3: ResNet-50 の圧縮後の精度のグラフ. 横軸が取り除いた辺の割合. 縦軸が精度である. 各グラフが K-means 法のクラスタ数に対応し, X-means というグラフが提案手法による圧縮後の精度を表している.

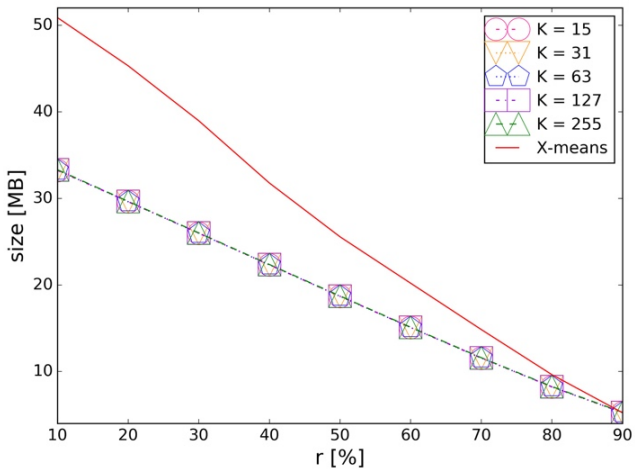

図 4: ResNet-50 の圧縮後の大きさのグラフ. 横軸が取り除いた辺の割合. 縦軸が大きさである. 各グラフが K-means 法のクラスタ数に対応し, X-means というグラフが提案手法による圧縮後の大きさを表している.

(2) DNN 間で共通した, パラメータと圧縮後の大きさの関係

図 4 は ResNet-50 の圧縮後の大きさのグラフを表している. まず Deep Compression を簡略化した手法による圧縮後の大きさについて述べる. 6 つの DNN では, クラスタ数を変えても圧縮後の大きさのグラフは, 最大でも圧縮前の DNN の 0.5%分の大きさしか変わらなかった. これは今回実装したプログラムでは, クラスタ番号を保持する際に必要最小のバイト数で保持したためである. クラスタ数を必要最小のビット数で保持していればクラスタ数が小さくなるほど圧縮後の大きさも小さくなる. 6 つの DNN では取り除く辺の割合を大きくするとそれに伴い線形的に圧縮後の大きさは小さくなった.

次に提案手法による圧縮後の精度について述べる. Resnet-50 では提案手法による圧縮後の大きさは, Deep Compression を簡略化した手法による圧縮後の大きさの最大で 1.53 倍になった. 6 つの DNN では提案手法による圧縮後の大きさが, Deep Compression を簡略化した手法による圧縮後の大きさより小さくなることはなかった. この原因は提案手法を用いた場合, クラスタリングの際のクラスタ数が極端に多くなる場合があったからだと考えられる. Resnet-50 では, 提案手法を用いて r=20 で圧縮した際の層の間の辺の重みにおける最大のクラスタ数は 1125 であった. クラスタ数が極端に大きくなることで, クラスタ番号を保持するのに必要なバイト数が大きくなり圧縮後の大きさも大きくなる.

(3) DNN の特徴との関連

図 5 は NIN の圧縮後の精度を表している. NIN では 40%以上の辺を取り除くと圧縮後の精度の下がり方が大きくなり始めた. ResNet-50 では 50%以上の辺を取り除くと圧縮後の精度の下がり方が大きくなり始めたので, それと比べると ResNet-50 の方が 10%多くの辺を取り除いても精度が下がりにくい結果となった. この精度の下がり方が大きくなり始める, 取り除く辺の割合は各 DNN では表 2 のようになった. LeNet のみ手書き数字の識別に用いられる DNN なので, その他の 5 つの DNN で層の数に注目すると, 層の数が多い DNN の方が多くの辺を取り除いても精度が下がりにくいことがわかる.

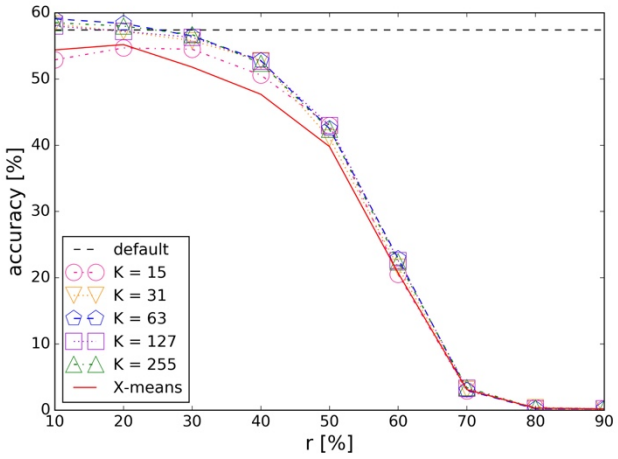

図 5: NIN の圧縮後の精度のグラフ. 横軸が取り除いた辺の割合. 縦軸が精度である. 各グラフが K-means 法のクラスタ数に対応し, X-means というグラフが提案手法による圧縮後の精度を表している.

表 2: 各 DNN で精度の下がり方が大きくなり始める時の取り除く辺の割合. 層の数は畳み込み層と全結合層の合計数.

DNN	辺の割合	層の数
LeNet	50%	4
AlexNet	40%	8
NIN	40%	12
VGG-16	40%	16
GoogLeNet	50%	58
ResNet-50	50%	54

5. おわりに

本稿では, Deep Compression を簡略化した手法を実装し, 6 つの DNN を圧縮した. Deep Compression を簡略化した手法では各層の間の辺から同じ割合で辺を取り除き, 各層の間の辺の重みを同じクラスタ数でクラスタリングを行った. Deep Compression とは異なり, DNN を圧縮後に再び学習させることはしなかった. また提案手法では X-means 法を用いて重みの量子化と共有を行う際のパラメータのクラスタ数を層の間の辺の重みごとに自動で決定する方法を試みた.

Deep Compression を簡略化した手法による圧縮実験の結果から, 圧縮時のパラメータと圧縮後の精度, 大きさの関係で次のことがわかった.

- クラスタ数を大きくしても, ある程度からは圧縮後の精度に影響がほとんどなくなった. その時の値は 6 つの DNN では 63 であった.
- クラスタ数を圧縮後の精度に影響がなくなる値まで大きくした時, 最低でも 30%の辺を取り除いても圧縮後の精度が下がらないことがわかった.
- 40%以上の辺を取り除くと精度の下がり方が大きく下がる可能性があることがわかった.
- 畳み込み層と全結合数の合計数が多い DNN の方が, より多くの辺を取り除いても圧縮後の DNN の精度がさがりにくい.

また提案手法ではクラスタ数が小さくなりすぎることで圧縮後の精度が下がり, クラスタ数が大きくなりすぎることで圧縮後の大きさが大きくなった. 提案手法では圧縮後の精度, 大きさの両方で良い結果が得られないことがわかった.

今後の課題としては取り除く辺の割合とクラスタリングのクラスタ数を各層の間ごとで変えて圧縮後の精度を調べ, 層単位でパラメータを設定する基準を探すことなどがある.

参考文献

[Song 16] Song Han, Huizi Mao, William J. Dally: Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, ICLR 2016.

[David 52] David A. Huffman: A Method for the Construction of Minimum-Redundancy Codes, Proceedings of the IRE (Volume: 40, Issue: 9, Sept. 1952), IEEE ,1952.

[Dan 00] Dan Pelleg, Andrew Moore: X-means: Extending K-means with Efficient Es- timation of the Number of Clusters, ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning Pages 727-734, June 29 - July 02, 2000.

[石岡 06] 石岡 恒憲: x-means 法改良の一提案 : k-means 法の逐次繰り返しと クラスターの再併合, 計算機統計学 18(1), 3-13, 2006-06-30, 日本計算機統計学会.

[David 07] David Arthur, Sergei Vassilvitskii: k-means++: the advantages of careful seed- ing, SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Pages 1027-1035, 2007.

[Schwarz 78] Schwarz, Gideon E.: Estimating the dimension of a model, Annals of Statistics, 6 (2): 461464, 1978.