

画像認識のための Optimally Wired Neural Networks Optimally Wired Neural Networks for Image Recognition

河野 隆太¹⁾ 松谷 宏紀²⁾ 鯉淵 道紘¹⁾ 天野 英晴²⁾
Ryuta Kawano Hiroki Matsutani Michihiro Koibuchi Hideharu Amano

1 はじめに

コンピュータビジョンの画像認識技術において、ニューラルネットワーク・アーキテクチャに基づくディープラーニングが有効とされている。近年のニューラルネットワークでは、ResNet [1] などの洗練されたネットワークモデルが確立されている。こうしたネットワークにおいて、特徴量の演算操作と伝播はそれぞれノードとエッジとしてモデル化されている。

さらに、ノード内の演算操作とその伝播のパターンを同時に探索するための Neural Architecture Search (NAS) 技術が広く研究されているが、その探索空間の膨大さのため、大規模なモデルにおいては依然として手動でのモデル設計が主流となっている。

こうした中、近年の研究において、ランダムグラフを基にネットワークモデルを構築する Randomly Wired Neural Network [2] が有望であることが示されている。この提案では、グラフ理論における古典的なランダムグラフモデルが採用されている。巡回を含むランダムグラフは、各ノードが同じ種類の畳み込み演算に対応した有向非巡回グラフ (Directed Acyclic Graph; DAG) に変換される。

一方、Order/Degree Problem [3] と呼ばれる問題が、特定のネットワークサイズに対する直径や平均距離といったグラフの評価指標を最適化するために取り組まれている。この問題に対して提案された手法のほとんどは、ヒューリスティックなアプローチを使用している。こうした手法の中で最も有望なものは、グローバルな対称性とローカルな不規則性を併せ持つグラフを、焼きなまし法 (Simulated Annealing; SA) を用いて最適化する手法 [4] である。この手法の結果として得られるグラフは、従来のランダムグラフよりも大幅に優れた直径と平均距離を実現できる。

本研究では、上記の最適化された対称トポロジをニューラルネットワークに適用する Optimally Wired Neural Network を提案する。すなわち、Randomly Wired Neural Network のフレームワークを活用し、基となるグラフとして最適化された対称トポロジを利用する。

2 Randomly Wired Neural Network

ネットワークジェネレータは、特定のパラメータの計算グラフを決定するマッピング関数として定義できる。例として、ResNet のネットワークジェネレータは、 $x + \mathcal{F}(x)$ を計算する残差ブロックのスタックを生成する。ResNet のパラメータは、ステージ数、各ステージのブロック数、深さ/幅/フィルタサイズ、活性化関数などを指定できる。

従来の NAS では、ネットワークジェネレータ自体は手動で設計される。こうした設計手法では、計算グラフ

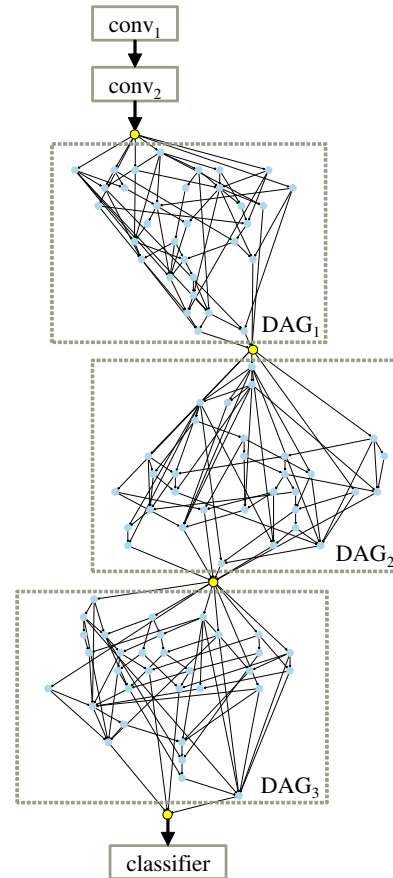


図 1 Watts-Strogatz (WS) ランダムグラフに基づく Randomly Wired Neural Network

を作成するための探索空間が厳しく制限される。この制限を緩和するために、ランダムに配線されたニューラルネットワーク [2] が提案されている。

画像認識用のニューラルネットワークでは、ネットワークをいくつかのステージに分割して、特徴マップを段階的にダウンサンプリングするのが一般的である。こうした手法と同様に、Randomly Wired Neural Network では、ネットワーク全体が複数のステージで構成され、その一部は、図 1 に示すようにランダムグラフから生成された有向非巡回グラフ (DAG) によって定義される。各 DAG は 1 つのステージに対応し、図 1 に黄色のノードとして示されている固有の入力/出力ノードによって、ステージ間が接続されている。

2.1 ランダムグラフの生成と DAG へのマッピング

Randomly Wired Neural Network のフレームワークでは、ネットワークジェネレータは、ノードとエッジのセットで構成されるランダムな無向グラフを生成する。この論文では、他の 2 つのランダムグラフモデル (Eldős-Rényi [5] および Barabasi-Albert モデル [6]) よりも

1) 国立情報学研究所 National Institute of Informatics

2) 慶應義塾大学 Keio University

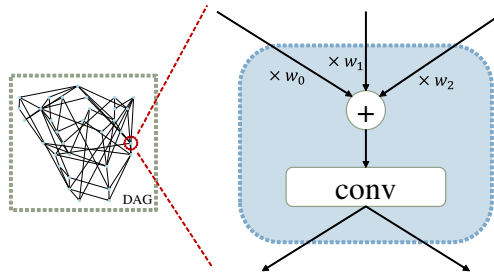


図 2 DAG レイヤ内のノード操作。3 入力, 2 出力の例。入力値は学習可能な 0 以上 1 以下の重み w_0, w_1, w_2 の加重和として集約される。conv と示された層で ReLU-convolution-BN による変換が行われ、値のコピーが 2 出力に渡される。

優れた性能を達成すると報告されている Watts-Strogatz ランダムグラフモデル [7] をベースラインとする。

Watts-Strogatz ランダムグラフ $WS(N, K, P)$ では、最初に N ノードがリング状に配置される。次に、リング内の各ノードは、近傍の K ノードとエッジにより接続される。さらに、各エッジ接続を確率 P でランダムな 2 ノード間の接続に置き換える。こうして循環を持つランダムグラフが生成された後、各ノードにインデックスを割り振り、順序付けを行う。このようにして、ニューラルネットワークとして値の伝播・演算が可能な有向非巡回グラフ (DAG) にマッピングされる。Watts-Strogatz ランダムグラフの場合、ノードのインデックスは時計回りの順序で順番に割り当てられる。

2.2 ノード操作

DAG のノード操作を図 2 に示す。DAG のノードには、複数の入力エッジ・出力エッジが存在する。(1 つ以上のエッジからの) 入力値は学習可能な重み変数 (例: w_0, w_1, w_2) の加重和として集約される。集約された値は、図 2 で ReLU-Conv-BN (Conv: 3×3 の分離可能な畳み込み) として定義される変換によって処理される。変換されたデータの同じコピーが、複数の出力エッジを介して送信される。

表 1 ニューラルネットワークアーキテクチャ。conv* は ReLU-Conv-BN 変換を表す (conv₁ では Conv-BN を用いる。)

stage	output size	layer (output channel count)
conv ₁	32×32	3×3 conv* ($C/2$)
conv ₂	16×16	3×3 conv* (C)
conv ₃	8×8	DAG wiring (C)
conv ₄	4×4	DAG wiring ($2C$)
conv ₅	2×2	DAG wiring ($4C$)
		1×1 conv*,
classifier	1×1	1280-d global average pool, 1000-d fc, softmax

2.3 ステージの構築

DAG レイヤには複数の入力ノードと出力ノードが存在しうするため、図 1 で黄色のノードとして示されている追加のノードが、DAG レイヤを含む 2 レイヤ間に挿入される。この追加ノードは、入力ノードからの値の重み付けされていない平均を計算し、その値を全ての出力ノードにコピーして伝播させる。DAG は 1 つのステージを表し、固有の入力/出力ノードによって前後の

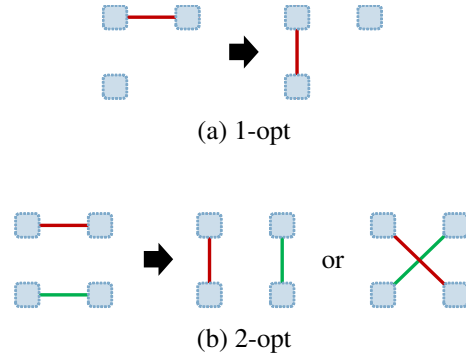


図 3 エッジ交換操作

ステージに接続される。入力ノードに直接接続されているすべてのノードについて、それらの変換はストライドが 2 になるように変更される。ランダムグラフのチャンネル数は、あるステージから次のステージに移動するときに 2 倍される。

図 1 は、この論文で使用されている Randomly Wired Neural Network の全体の実装をまとめたものである。conv1 と conv2 では、単一の畳み込み層を用いる。ネットワークは classifier の出力で終了する (表 1、最後の行)。

3 Optimally Wired Neural Network

この研究では、ランダムグラフよりも洗練されたグラフジェネレータを使用する Optimally Wired Neural Network を提案する。フレームワーク内の DAG レイヤ (2.1 節) へのネットワークジェネレータとして、ランダムグラフではなく、最適化された点対称グラフを利用する。

3.1 点対称性を持つ低直径グラフ

本節では、点対称を持つ低直径グラフ SymSA [4] を生成するための先行手法について説明する。点対称トポロジ SymSA は、大規模な低直径ネットワークを構成するための Order/Degree Problem の最適解法の一つである。

3.1.1 エッジの入れ替えによる新たなグラフの生成

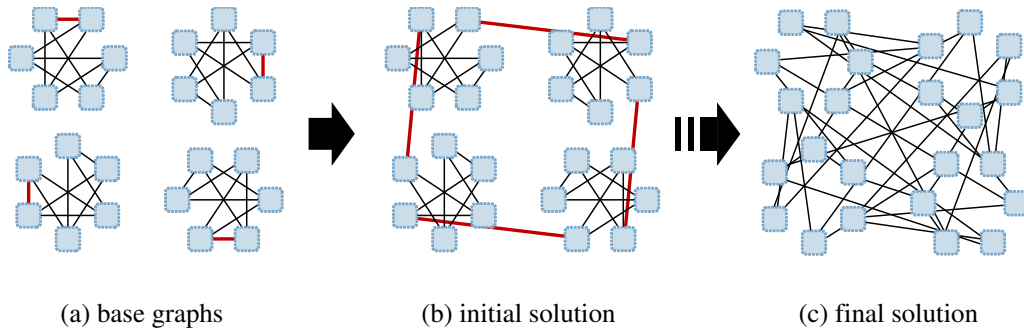
SymSA グラフの最適化手法の中で、現在の最良解であるグラフに対し、エッジの入れ替え操作を行うことにより新たな解を生成する。図 3 に入れ替え操作手法の概観を示す。現在の解のグラフに対し、グラフ内からランダムに選択したエッジ 1 本ないしは 2 本について入れ替え操作を行い、入れ替え後のグラフを新たな解とする。

図 3(a) に示すエッジ 1 本の入れ替えを行う 1-opt 操作では、1 本のエッジの一端を別のノードに付け替える。また、図 3(b) に示すエッジ 2 本の入れ替えを行う 2-opt 操作では、2 通りの入れ替え方のうちいずれか 1 つをランダムに選択する。

3.1.2 グラフの点対称性を維持した解の探索

Order/Degree Problem の解となるグラフを高速に最適化するためには、その解の評価関数となるグラフの直径・平均距離の算出に必要な計算量を減らすことが重要である。そこで、解のグラフに点対称性を持たせることにより、探索空間と必要計算量を同時に削減している。

この手法では、与えられたノード数 N 、次数 d に対し、 N の約数となる整数 g を導入する。この変数 g はグループ数を表し、グラフを 2 次元平面に展開した際に $360/g$ 度回転すると、ノードとエッジの接続関係が全く同じで

図 4 焼きなまし法 (SA) を用いた点対称トポロジ SymSA の最適化 ($N = 24, d = 3, g = 4$)

あるグラフが得られる。 $g = 1$ の場合、点対称ではない通常のグラフが得られる。

グラフの点対称性を維持しつつ焼きなまし法を用いて新たな解を生成する例を図 4 に示す。この例ではノード数・次数・グループ数をそれぞれ $N = 24, d = 3, g = 4$ と設定している。

(1) 初めにノード数 N/g , 次数 d のランダム正則グラフを生成し、それを図 4(a) に示すように点対称に g 個並べる。(2) 次に、図 4(a) の太線 (赤線) で示すように点対称なエッジを各グラフから 1 本ずつ、合計 g 本ランダムに選択し、それらに対して図 3(a) に示した 1-opt 操作によるエッジの入れ替えを行う。これにより図 4(b) に示すような点対称かつ 1 つのグラフとして連結された初期解を得る。(3) さらに、この初期解を入力とし、図 3(b) に示した 2-opt 操作 (もしくは図 3(a) に示した 1-opt 操作) を繰り返し適用して新たな解を生成する。この解が直径または平均距離をより小さくするか、または焼きなまし法において確率的に遷移条件を満たした場合、現在の解として保存する。

これらの手順により、図 4(c) に示すような最終的な解となるグラフを得る。

3.2 SymSA(N, d, g) のニューラルネットワークへの適用

本報告の提案である Optimally Wired Neural Network では、Randomly Wired Neural Network のフレームワーク内でランダムグラフの代わりに、最適化された点対称グラフ SymSA(N, d, g) をネットワークジェネレータとして使用する (2.1 節)。SymSA(N, d, g) の生成後、ノードの順序としてインデックスを付けることにより、計算可能なニューラルネットワークの有向非巡回グラフ (DAG) にマッピングされる。SymSA(N, d, g) において、ノードのインデックスは対称性に沿って割り当てられる。

4 ネットワークジェネレータの比較

本報告の評価において、Randomly Wired Neural Network では、32 ノード、64 エッジのランダムグラフ WS(32, 4, 0.75) が各 DAG レイヤにおいてネットワークジェネレータとして使用される。一方で、Optimally Wired Neural Network では、各 DAG レイヤにおいて 32 ノード、次数 4 の点対称グラフ SymSA(32, 4, 4) を使用する。

WS ランダムグラフと SymSA 最適化グラフの性能比較を表 2 に示す。ここで評価されるホップ数は DAG に変換する前の無向グラフのものである。WS ランダムグラフ、SymSA 最適化グラフの両方において、異なるラン

表 2 トポロジ比較

	WS(32, 4, 0.75)	SymSA(32, 4, 4)
# of Nodes	32	32
# of Edges	64	64
Diameter	4.533 ± 0.507	3.000 ± 0.000
ASPL	2.532 ± 0.054	2.355 ± 0.001

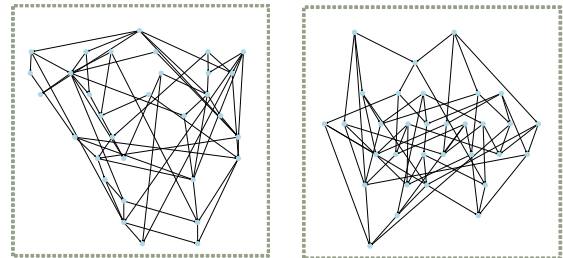


図 5 Watts-Strogatz ランダムグラフに基づく DAG (WS(32, 4, 0.75)) 図 6 点対称トポロジ SymSA に基づく DAG (SymSA(32, 4, 4))

ダムシード値から 10 個のグラフを生成し、直径・平均距離の平均・標準偏差を評価している。

本報告の提案手法で用いる最適化グラフ SymSA(32, 4, 4) は、ランダムシード値に関わらず理論下界のホップ数 3 を達成している。さらに、SymSA は既存手法で用いる WS ランダムグラフと比べ、直径と平均距離をそれぞれ 33.8%, 7.0% 削減している。

DAG 変換後の WS ランダムグラフ及び SymSA 最適化グラフを図 5 と図 6 にそれぞれ示す。これらのグラフを Randomly Wired Neural Network 及び Optimally Wired Neural Network の DAG レイヤにそれぞれ用いる。

5 評価

CIFAR-10 [8] 分類タスクをアプリケーションとして用いる。各 DAG レイヤ内のグラフのノード数を $N = 32$ とし、チャンネル数のハイパーパラメータを $C = 78$ とする。これらの設定値は Randomly Wired Neural Network のものと同様である。異なるランダムシードから 10 組の DAG を生成し、テスト誤差と top-1 精度を評価する。誤差・精度の評価値は "平均 \pm 標準誤差" として示す。ネットワークの訓練エポック数は 100 とし、学習率 0.001 の Adam optimizer を用いる。

比較手法の Randomly Wired Neural Network と提案手法の Optimally Wired Neural Network において、4 節に示したランダムグラフ WS(32, 4, 0.75) と最適化された点対称グラフ SymSA(32, 4, 4) をグラフジェネレータとしてそれぞれ用いる。

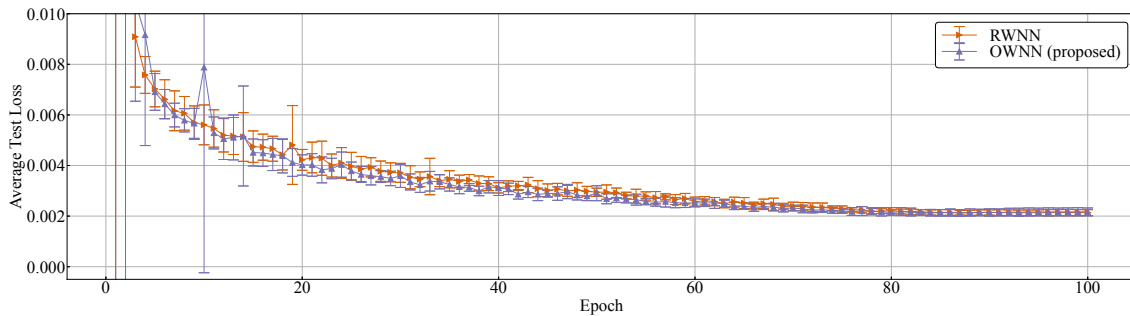


図 7 RWNN と OWNN (提案手法) における CIFAR-10 平均テスト誤差

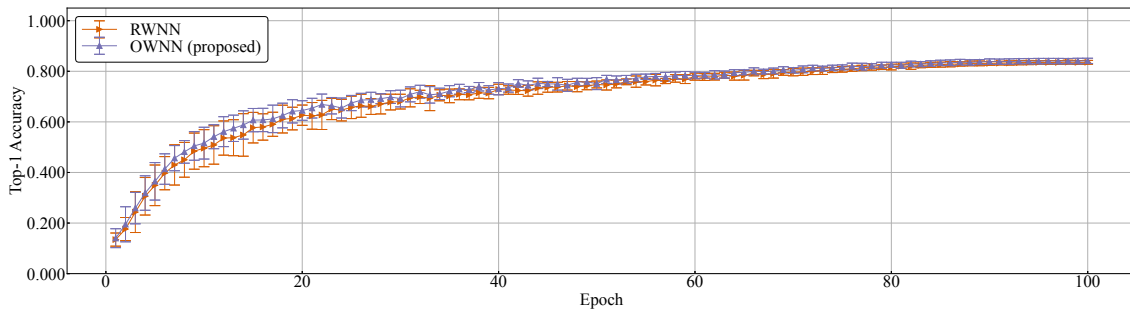


図 8 RWNN と OWNN (提案手法) における CIFAR-10 Top-1 分類精度

各エポックにおけるテスト誤差と top-1 精度の評価結果を図 7, 8 にそれぞれ示す。この評価結果より、グラフジェネレータである WS ランダムグラフと SymSA 最適化グラフ (図中ではそれぞれ RWNN, OWNN と表記する) は、100 エポックの訓練中に、DAG を生成するためのランダムシードに関わらず、精度の改善を達成している。テスト誤差と top-1 精度の標準誤差は、RWNN と OWNN の両方において、それぞれ 0.1, 0.001 以下の値を、ほぼすべてのエポックにおいて達成している。

本報告で提案した Optimally Wired Neural Network (OWNN) は、最適化された点対称グラフを利用して、ランダムグラフを用いる Randomly Wired Neural Network に比べ、訓練中の誤差・精度を大幅に改善している。訓練中において、epoch=14 で 7.2 %改善する効果が見られた。これは、提案フレームワークの点対称グラフがよりクラスタ性の高い接続を持つことにより、入出力間の低ホップ数な複数経路におけるアンサンブル学習がより効率的に行われたものと考えられる。

その一方で、提案手法である Optimally Wired Neural Network の最終的な精度は、既存の Randomly Wired Neural Network に比べ、0.67 %の改善に留まっている。これは、CIFAR-10 のアプリケーション規模においては、よりホップ数の大きい複数経路によるアンサンブル学習を用いても、十分に準最適な精度まで最適化可能であるためと考えられる。

6 おわりに

本研究では、画像分類タスクのための Optimally Wired Neural Network を提案した。このモデルでは、Randomly Wired Neural Network のフレームワークにおける DAG レイヤ内で、ホップ数最適な点対称グラフを用いる。このグラフのノード間の密な接続及び接続の均一性により、生成されたニューラルネットワークが従来のモデルに比べ分類精度を向上させることが分かった。

今後の課題として、本提案をより大規模なニューラルネットワークモデルに適用することや、より一般的な深層学習タスクに適用することが挙げられる。

謝辞

本研究の一部は JSPS 科研費 JP20K19788 の助成を受けたものである。

参考文献

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016.
- [2] Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. Exploring Randomly Wired Neural Networks for Image Recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1284–1293, October 2019.
- [3] Graph Golf: The Order/degree Problem Competition. <http://research.nii.ac.jp/graphgolf/>.
- [4] Masahiro Nakao, Hitoshi Murai, and Mitsuhsisa Sato. A Method for Order/Degree Problem Based on Graph Symmetry and Simulated Annealing with MPI/OpenMP Parallelization. In *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, HPC Asia 2019*, pp. 128–137, New York, NY, USA, 2019. ACM.
- [5] P. Erdős and A. Rényi. On the Evolution of Random Graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pp. 17–61, 1960.
- [6] Reka Albert and Albert-Laszlo Barabasi. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, Vol. 74, No. 1, pp. 47–97, January 2002.
- [7] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. Vol. 393, p. 3, 1998.
- [8] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.