

深層学習における正則化へのドロップアウトデザインの適用

熊澤 努
株式会社 SRA

kumazawa@sra.co.jp

地寄 頌子
大阪工業大学

shoko.chisaki@oit.ac.jp

中川 智之
東京理科大学

t_nakagawa@rs.tus.ac.jp

室井 浩明
エヂリウム株式会社
muroi@edirium.co.jp

渡邊 卓也
エヂリウム株式会社
sodium@edirium.co.jp

要旨

深層ニューラルネットワークの汎化性能を向上させる正則化法に、学習時に一部のノードだけを活性化するドロップアウト法が知られている。ドロップアウト法が活性化するノードを無作為に選択する技法なのに対して、ノードと重みを1エポックの訓練で均一に選択する、ドロップアウトデザインと呼ばれる組合せ構造が提案されている。しかしながら、正則化の効果については不明点が多く、実験を通じて解明されることが望まれてきた。本論文では、ドロップアウトデザインの性質を明らかにするために、ドロップアウトデザインを用いた正則化法を多層パーセプトロンに適用した。4種類のネットワーク構成について、画像分類問題の精度と損失を評価する実験を行い、すべてのネットワークについてドロップアウト法と同程度の汎化性能を示すことを確認した。一方で、ドロップアウトデザインに固有の特徴的な結果は観察されなかった。以上の成果は、ドロップアウト法とドロップアウトデザインをより深く理解するための手掛かりを今後の研究に提供することが期待される。

1. はじめに

近年、深層学習に関連した研究開発が非常に活発に行われており、画像認識や自然言語処理をはじめとして様々な分野への応用が図られている。深層学習は訓練データに対する最適化を実行するため、訓練データに対しては高い評価性能を示す一方、未知のデータに対する汎化性

能は向上しなくなる問題が指摘されている。この過学習の問題を緩和する方法が正則化である。

ドロップアウト法 [1, 2] は最も有力な正則化法の一つとして広く知られている。ドロップアウト法は、ネットワークの訓練の間に無作為にノードを不活性にすることで、一部の重みパラメータの学習のみを進める方法である。不活性にするノードはミニバッチごとに切り替える。訓練を終了した後の、未知データに対する出力値の推定の際には、全てのノードを活性化させる。ノードの不活性化は、部分ネットワークで重みパラメータの値を学習することと解釈できる。そのため、ドロップアウト法は複数のモデル間でパラメータを共有するアンサンブル学習の一種としても理解されている。ドロップアウト法を使用する際には、ノードを不活性にする確率をハイパーパラメータで与える必要がある。

地寄らは、ドロップアウト法に代わる正則化技法を想定した組合せ構造であるドロップアウトデザイン [3] を提案した。ドロップアウトデザインは、ミニバッチごとの訓練で使用する部分ネットワーク構成の集合である。この部分ネットワークの構成をスーパーブロックという。ドロップアウトデザインは、1エポックあたりのノードの活性化回数と重みパラメータの更新回数が所与の値で釣合うように、スーパーブロックを構成する。したがって、訓練時に特定のノードだけが活性化されて学習が進む現象を緩和することが期待できる。文献 [3] では、特定のドロップアウトデザインの構成法が示されており、計算機を用いたデザインの構成が可能となっている。その一方で、正則化の効果をはじめ、ドロップアウトデザインの性質については明らかになっていない点が多い。

特に、ドロップアウトデザインを機械学習に実際に適用してその性質を調べる研究はこれまでなされておらず、評価実験に基づいた研究の成果が望まれる状況にある。

本論文の目標は、ドロップアウトデザインを用いた正則化を深層学習に対して適用して、デザインの性質を実験的に明らかにすることにある。実験では、画像分類問題を取りあげた。ニューラルネットワークは、多層パーセプトロン（MLP）を対象とした。全結合層（隠れ層）の層数とノード数が異なる計4種類のネットワークに対して、正則化を実行しない場合、従来のドロップアウト法を適用した場合、ドロップアウトデザインを適用した場合についてそれぞれ汎化性能を評価した。その結果、次の二点を確認した。

- すべてのネットワークについて、ドロップアウト法とドロップアウトデザインによる正則化法は同程度の精度と損失を示した。
- ドロップアウト法にはない特徴的な結果は、ドロップアウトデザインを用いた結果には見られなかった。

以上の結果は、ノードを無作為に選ぶドロップアウト法と、ブロックによって均一にノードを活性化させるドロップアウトデザインには、性能上大きな差がないことを示唆している。このことは、組合せ構造であるドロップアウトデザインと無作為性を持つドロップアウト法の関連性を示しており、今後の研究において、組合せ論を使ったことによりドロップアウト法を分析するアプローチが期待できる。本論文のもう一つの貢献は、ドロップアウトデザインをニューラルネットワークに適用する具体的な方法を示したことである。ドロップアウトデザインの適用に際しては、デザインの構造とニューラルネットワークの構成との対応づけ、複数エポックの訓練におけるスーパーブロックの割り当てを決定する必要がある。本論文では、多層パーセプトロンの全結合層に対して正則化を行う場合のドロップアウトデザインの適用法を示し、実験により汎化性能の違いを調べた。

本論文の構成は次の通りである。2節で研究の背景となるドロップアウト法とドロップアウトデザインについて説明する。3節では、ドロップアウトデザインの深層学習への適用法を説明する。4節で、評価実験の結果を報告する。5節で関連する研究を論じ、6節で結論と今後の課題を述べる。

2. 背景

ここでは、本論文が着目するドロップアウト法とドロップアウトデザインの概要を説明する。

2.1. ドロップアウト法

ドロップアウト法 [1, 2] は、ニューラルネットワークのノードを無作為に不活性にして訓練を行うことで、汎化性能を高める技術である。ノードを不活性にする確率であるドロップアウト率 p はハイパーパラメータとして与えられる、訓練時には、層ドロップアウトマスクと呼ばれるベクトル $\mathbf{m}$ で層の出力をマスキングすることでノードの活性化と不活性化を行う [4]。ただし、 $\mathbf{m}$ の各成分 m_i はベルヌーイ分布に従って確率 p で0を、確率 $1-p$ で1をとる。すなわち、 $m_i \sim \text{Bernoulli}(1-p)$ とする。ここで、 $\mathbf{m}$ の各成分はノードを表し、1であれば活性化することを、0であれば不活性化することを意味する。ドロップアウト法を適用する層への入力 $\mathbf{x}$ と出力 $\mathbf{y}$ との間には以下の関係が成り立つ [4]。

$$\mathbf{y} = f(\mathbf{W}\mathbf{x} + \mathbf{b}) \circ \mathbf{m} \quad (1)$$

ここで、 $\mathbf{W}$ は層の重み、 $\mathbf{b}$ はバイアス、 f は活性化関数、 $\circ$ は成分ごとの積を表す。未知データに対する推定時には全てのノードを活性化し、以下のように平均化を行う。

$$\mathbf{y} = (1-p)f(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (2)$$

ドロップアウト法の異なる定式化として、訓練時には (1) 式の右辺に $1/(1-p)$ を乗じた値を出力する一方、推定時には (2) 式の $(1-p)$ の乗算を実行しない方式があり、どちらの定式化を用いても、推定時の出力が等しいことが知られている [4]。簡単のため、本論文では両者を区別せず、(1) 式と (2) 式を用いる。

ドロップアウト法で訓練に使われるネットワークは、元のネットワークの部分ネットワークである。したがって、ドロップアウト法を、重みを共有する複数のネットワークによるアンサンブル学習の一種とみなすことができる [1]。

2.2. ドロップアウトデザイン

ドロップアウトデザインは、ニューラルネットワークの重みを更新する回数を均一にすることを目的とした組

合せ構造の一種である．ドロップアウトデザインは複数の点集合から構成される．ネットワークの各層のノードの集合を異なる点集合とみなし，層を横断した部分集合をとることによって活性化するノードを選択するモデルを与える． $V_1, V_2, \dots, V_n$ を互いに異なる点集合とし，それぞれの部分集合の集合 (スーパーブロックと呼ぶ) を以下のように定める．

$$\mathcal{B} = \{ \{C_1 | C_2 | \dots | C_n\} \mid C_i \subset V_i, C_i \neq \emptyset, 1 \leq i \leq n \}.$$

各部分集合 C_i をサブブロックと呼ぶ．任意の $0 \leq i \leq n-s$ について，連続した s 個の点集合 $V_{i+1}, V_{i+2}, \dots, V_{i+s}$ それぞれから選出した任意の $d_1, d_2, \dots, d_s$ 個の点を同時に含むスーパーブロックが $\mathcal{B}$ の中に λ_i 個存在するとき， $(V_1, V_2, \dots, V_n; \mathcal{B})$ を $(d_1, d_2, \dots, d_s)$ 型-ドロップアウトデザインと呼ぶ [3]． λ_i ($0 \leq i \leq n-s$) はドロップアウトデザインの会合数と呼ばれる．会合数を一定にすることで，ノードの活性化回数及びエッジの使用回数の均一化を実現する．2層のネットワークでの例を以下に示す．

例 1. 点集合を $V_1 = \{0, 1, 2, 3\}, V_2 = \{0, 1, 2, 3, 4, 5\}$ とする．このとき， $(V_1, V_2; \mathcal{B})$ は会合数 $\lambda_1 = 1$ をもつ (2, 1) 型-ドロップアウトデザインをなす．ただし，

$$\begin{aligned} \mathcal{B} = \{ & \{0, 1 \mid \mathbf{0}, 1, 2\}, \{0, 1 \mid \mathbf{3}, 4, 5\}, \{0, 2 \mid \mathbf{1}, 2, 5\}, \\ & \{0, 2 \mid \mathbf{0}, 3, 4\}, \{0, 3 \mid \mathbf{1}, 2, 4\}, \{0, 3 \mid \mathbf{0}, 3, 5\}, \\ & \{1, 2 \mid \mathbf{0}, 2, 3\}, \{1, 2 \mid \mathbf{1}, 4, 5\}, \{1, 3 \mid \mathbf{0}, 4, 5\}, \\ & \{1, 3 \mid \mathbf{1}, 2, 3\}, \{2, 3 \mid \mathbf{0}, 1, 3\}, \{2, 3 \mid \mathbf{2}, 4, 5\} \}. \end{aligned}$$

このデザインでは，例えば， V_1 の2個の点0, 1と V_2 の点0を同時に含むスーパーブロックは1個だけである．

先行研究において，直交配列を用いた構成法や，アフィン空間や射影空間の有限幾何を用いた構成法が提案されている．本論文の実験に使用したアフィン空間による構成法は，(2, 1) 型かつ (1, 2) 型-ドロップアウトデザインを構成することができる (Theorem 5.13 [3])．ここで，デザインから構成されるモデルは，素数ベキ q ，整数 t ， $d \geq 3$ について，層数は q^{d-t} ，各層のノード数は q^t であり，会合数は各 i に対して $\lambda_i = (q^{d-2} - q^{d-t-1})/(q-1)$ をとる．また，各層のノードを活性化する割合は $1/q$ である．したがって，不活性にするノードの割合，すなわちドロップアウト率は $1 - 1/q$ となる．

3. ドロップアウトデザインの深層学習への適用

本節では，ドロップアウトデザインのニューラルネットワークへの適用方法について述べる．

本論文では，訓練に使用するドロップアウトデザインをネットワークごとに固定する．ドロップアウトデザインは，1エポックの学習に必要な部分ネットワークの集合を定めたものとする．2.2節で述べたように，ドロップアウトデザインの点集合は，ノードの集合と1対1に対応させる．スーパーブロックは，1つのミニバッチで学習を進める部分ネットワークであり，サブブロックは各層において活性化するノードの集合とする．(1)式において，サブブロックに含まれる点に対応づけられたノードを1，含まれない点に対応づけられたノードを0とするように層ドロップアウトマスク $\mathbf{m}$ を構成することで，訓練時のノードの活性化と不活性化を実現する．

ドロップアウトデザインの各スーパーブロックを，1エポックにつき1回，重みの更新に使用する．ただし，複数エポックの訓練におけるスーパーブロックの使用法については，ドロップアウトデザインの定義には定められていない．本論文では，第1エポックで使用するスーパーブロックの順序を任意に一つ固定するものとし，複数エポックの訓練について，次の二通りの使用法を考える．

- 第1エポックで定めたスーパーブロックの使用順序を巡回的に保つ．第2エポック以降では，各エポックの開始時にスーパーブロックを1だけシフト移動して巡回したデザインを構成することで，スーパーブロックの使用順序を定める．以降では，この使用方法をブロックシフトと呼ぶことにする．ブロックシフトにより構成されるドロップアウトデザインは互いに同値である点に注意する．
- 第1エポックで定めたスーパーブロックの使用順序を第2エポック以降も引き続き使用する．

例 2. 例1のドロップアウトデザイン $\mathcal{B}$ を再び考える．表1にブロックシフトの適用例を示す．まず， $\mathcal{B}$ のスーパーブロックの集合を順列と読み替えることで，第1エポックでのスーパーブロックの使用順序を表1aのように定める．このデザインにブロックシフトを適用すると，第2エポックでのスーパーブロックの使用順序が表1bのように定まる．ブロックシフトを適用しない場合には，表1aの順序を第2エポック以降も使用する．

表 1: ブロックシフトの例

(a) 第 1 エポック			(b) 第 2 エポック		
順序	1 層	2 層	順序	1 層	2 層
1	0, 1	0, 1, 2	1	2, 3	2, 4, 5
2	0, 1	3, 4, 5	2	0, 1	0, 1, 2
3	0, 2	1, 2, 5	3	0, 1	3, 4, 5
4	0, 2	0, 3, 4	4	0, 2	1, 2, 5
5	0, 3	1, 2, 4	5	0, 2	0, 3, 4
6	0, 3	0, 3, 5	6	0, 3	1, 2, 4
7	1, 2	0, 2, 3	7	0, 3	0, 3, 5
8	1, 2	1, 4, 5	8	1, 2	0, 2, 3
9	1, 3	0, 4, 5	9	1, 2	1, 4, 5
10	1, 3	1, 2, 3	10	1, 3	0, 4, 5
11	2, 3	0, 1, 3	11	1, 3	1, 2, 3
12	2, 3	2, 4, 5	12	2, 3	0, 1, 3

文献 [3] ではドロップアウトデザインを適用するネットワーク構成について特に仮定は設けられていない。本論文では、MLP を対象として、これらのネットワークの全結合層にドロップアウトデザインを適用する。そのためには、各層の出力ベクトルの各成分とデザインの各点とを一对一に対応づければよい。

4. 評価実験

本節では、計算機による評価実験の結果を報告する。本実験の目的はドロップアウト法に対する優位性を評価することではなく、ドロップアウトデザインによる正則化とドロップアウト法による正則化の効果に違いがあるか比較検討することで、ドロップアウトデザインの性質を調べることである。

4.1. 実験設定

本実験では、深層ニューラルネットワークの全結合層に対して、訓練時の性能と、テストデータを用いた時の汎化性能の評価を行った。性能評価は、ドロップアウトデザインを用いた正則化を施した場合、ドロップアウト法を用いた正則化を施した場合、正則化を行わない場合を対象とした。

本実験で扱う問題は画像分類問題とした。データセットには CIFAR-10 [5] を使用した。CIFAR-10 は、50,000 個

の訓練データと 10,000 個のテストデータからなる 10 クラス画像分類用データセットである。各データは 32×32 ピクセルのカラー画像である。3 節で述べたように、実験を行うニューラルネットワーク構成は MLP とした。ドロップアウトデザインとドロップアウト法はネットワークの全結合層だけに適用した。使用するドロップアウトデザインは、4 種類の (2, 1) かつ (1, 2) 型-ドロップアウトデザインとした。全結合層の構成は、実験で使用するドロップアウトデザインに基づいて定めた。表 2 に、実験で扱うドロップアウトデザインと、対応する MLP の全結合層の構成を示す。4 種類の各ネットワークに対してドロップアウト法を適用する場合には、各層のドロップアウト率を表 2 にある値に設定した。この値は、ドロップアウトデザインによって定まるノードの不活性率と同一の値である。ドロップアウトデザインを適用する際には、ブロックシフトを実行する場合と、ブロックシフトを行わず、どのエポックでも同一順序でスーパーブロックを適用する場合の二通りを評価した。スーパーブロックの使用順序の性能への影響を調べるためである。

各ネットワークの損失関数には交差エントロピーを、パラメータの更新には確率的勾配降下法 (SGD) を使用した。また、入力層と全結合層の活性化関数は ReLU 関数とし、出力層の活性化関数は Softmax 関数とした。

実験では、各ネットワーク構成について、500 エポックの学習を行った。その後、テストデータに対する精度 (正解率) と、交差エントロピーに基づく損失を評価した。精度と損失はテストデータの正解クラスについて求めた。この実験をそれぞれのネットワーク構成について 10 回ずつ行った結果を評価した。ドロップアウト法とドロップアウトデザインの正則化効果だけを識別しやすくするため、データ拡張などの他の汎化性能を向上させる技法は使用しなかった。

実験プログラムは Python 3 で実装した。ニューラルネットワークの訓練ならびにテストの実装には、Keras ライブラリを用いた。ドロップアウト法には、Keras が提供するドロップアウト層の実装を使用した。一方、ドロップアウトデザインに関しては、アフィン空間を利用したドロップアウトデザインの構成法 (2.2 節) を実装した。また、Keras のドロップアウト層と同じ実装方法でデザインを用いたノードの活性化を実現するために、従来のドロップアウト層を拡張したデザイン適用層を新たに実装した。このデザイン適用層を用いて、実験プログラムは、構成したデザインから層ドロップアウトマス

表 2: 実験で使用したドロップアウトデザインとネットワーク構成の対応

アフィン空間			ドロップアウトデザイン					ニューラルネットワーク（全結合層）			
次元 d	位数 q	flat の 次元 t	サブ ブロック 数	点 集合 サイズ	サブ ブロック サイズ	スーパー ブロック 数	会 合 数	層 数	1 層の ノード 数	1 層の 活性化 ノード数	ドロップ アウト 率
8	2	7	2	128	64	508	63	2	128	64	0.5
7	3	6	3	729	243	3276	121	3	729	243	0.67
9	2	7	4	128	64	1016	126	4	128	64	0.5
10	2	8	4	256	128	2040	254	4	256	128	0.5

クを生成して、(1) 式に従って訓練を実行する。テスト時には、ドロップアウト法と同様に (2) 式により出力を計算することとした。

実験には、CPU に Intel Xeon W-2133 (3.60GHz), GPU に NVIDIA TITAN RTX を備えたマシン (Keras 2.3.1, Tensorflow 2.1.0) と、CPU に Intel Xeon W-2245 (3.90GHz), GPU に NVIDIA TITAN RTX を備えたマシン (Keras 2.4.3, Tensorflow 2.3.0) を使用した。

4.2. 実験結果

MLP の全結合層を 2 層, 3 層, 4 層にした場合の実験結果をそれぞれ 図 1, 図 2, 図 3, 図 4 に示す。測定した値は、10 回の実験に対する精度の平均と標準偏差、損失の平均と標準偏差である。加えて、第 51 エポックから第 500 エポックまでの各エポックについて、そのエポックを含む過去 50 エポックの区間に対する精度と損失の分散を算出した。これは、訓練の進行に伴う、実験ごとの汎化性能の収束の違いを評価する指標である。各グラフは、正則化を行わなかった結果 (**w/o dropout**), ドロップアウト法による正則化を行った結果 (**dropout**), ブロックシフトを適用した場合 (**design(bs)**) と適用しない場合 (**design**) のドロップアウトデザインでの正則化の結果をそれぞれ示している。グラフの横軸はエポック数である。

図 1 から図 4 から、以下の三点が観察された。

1. 本実験では、ドロップアウトデザインを用いることで汎化性能が高まるかどうかを調べるため、精度と損失の平均、標準偏差、分散を測定した。それらすべての項目について、ドロップアウトデザインを用いた正則化は、従来のドロップアウト法と同程度の汎化性能を達成した。

2. 無作為にノードを不活性にする従来のドロップアウト法と異なり、ドロップアウトデザインは活性化するノードの回数に制約を設けた決定論的な組合せ構造である。そのため、ドロップアウトデザインの固有の特徴を示す結果が得られることが期待される。実験の結果からは、精度と損失に関して、ドロップアウトデザインとその適用法に特有の性質や挙動は確認できなかった。

3. ドロップアウトデザインの適用において、ブロックシフトを実行した場合としない場合を比較したとき、精度と損失の平均、標準偏差に大きな違いは確認されなかった。この結果は、スーパーブロックの使用順序は汎化性能に影響しないことを示していると考えられる。

ドロップアウト法との正則化の違いを評価するという点から、まず、上で述べた結果 1 と 2 を詳しく検討する。本実験の結果から、表 2 に示したすべてのネットワーク構成について、ドロップアウト法とドロップアウトデザインを用いた正則化は平均的に同程度の精度と損失であり、各曲線の収束の仕方にも大きな違いは見られなかった。ただし、精度と損失の平均値に関しては、全結合層が 4 層、各層のノード数が 128 の場合には、両者の挙動にわずかながら違いを確認した。訓練データ、テストデータ共に平均精度はドロップアウト法が上回り (図 3a, 3b), ドロップアウト法が低損失になるという傾向が観察された (図 3c, 3d)。また、同じ構成について、精度の標準偏差の結果 (図 3e, 3f) から、ドロップアウト法を用いた場合は、ドロップアウトデザインを用いる場合よりもわずかに低い標準偏差を示した。このような結果が得られた原因として、中間層数が最も大きいネットワークを使用していることが考えられる。しかし、各

層を 256 ノードとした結果 (図 4e, 4f) では同様の傾向は見られないことから、ドロップアウトデザインの固有の特徴を示す結果とはいえないと思われる。ニューラルネットワークの訓練とドロップアウト法における確率的な挙動が影響している可能性もあり、本論文で行った実験だけからでは結論を得ることは難しい。今後、5 層以上の深いネットワークに対する評価実験が必要である。50 エポック区間の精度分散、損失分散に関しては、全結合層 2 層、3 層、4 層のいずれの場合も、ドロップアウト法とドロップアウトデザインを用いた正則化と顕著な違いは見られなかった。以上の考察から、訓練の進行に伴う精度や損失の収束の仕方には大きな違いはないと結論付けられる。

次に、ドロップアウトデザインの結果に注目して、ブロックシフトを実行した場合としない場合を比較した場合を議論する。上の 3 で述べたように、すべての実験を通じて、精度と損失の平均、標準偏差に大きな違いは確認されなかった。なお、全結合層 4 層、各層 128 ノードに対する 50 エポック区間の精度と損失の分散 (図 3i, 3j) において、ブロックシフトの方が訓練の初期段階での分散値が低くなる傾向が見られるが、その差は微小である。

最後に、実験結果への影響を与える要因として、過学習の可能性と収束速度について検討する。正則化を行わない場合については、全結合層 3 層 (図 2a) と 4 層かつ各層 256 ノード (図 4a) のそれぞれにおいて、短いエポック数の訓練で精度が 1.00 に達して収束し、また、標準偏差が 0 になった。そのため、訓練の進行と共に過学習を引き起こしている可能性が高い。全結合層 4 層、各層 128 ノードの場合は、図 3e においても、エポック数の増加に伴いテストデータに対する精度が低下する傾向があり、過学習となっている恐れがある。これらのネットワーク構成では、正則化技法を用いない場合に最も短い収束速度で高い汎化性能が得られた、という結果を示している。以上の議論から、この現象は過学習が原因である可能性がある。ドロップアウト法とドロップアウトデザインを用いた場合には、過学習の傾向は見られなかった。全結合層 3 層と 4 層については、訓練時の平均精度が収束まで達しておらず、正則化技法を用いない場合と比較して、収束に遅れが見られる。訓練するエポック数をさらに増やすと、精度や損失の結果に違いが現れる可能性がある。

5. 関連研究

ドロップアウト法は後に続く正則化法の研究に大きな影響を与えた。代表的な正則化法の一つにドロップコネクト法 [6] がある。ドロップコネクト法は、学習時にノード間の接続を無作為に切断する方法である。一方、高速ドロップアウト法 [7] は、ガウス近似によりドロップアウト法と同様の正則化効果を発生させることで、学習時の収束速度を高める方法である。より最近では、畳み込み層の出力の周波数成分に対してドロップアウト法を適用するスペクトラルドロップアウト法 [8]、ドロップアウト法を用いた訓練をベイズ推定で近似する MC ドロップアウト法 [9, 10]、学習効果の高い部分ネットワークを進化計算を用いて絞り込む EDropout [11]、強化学習でドロップアウトマスクを学習する AutoDropout [12] などの多くの方法が提案されている。また、R-Drop は、ドロップアウト法で部分ネットワークを二つ構成し、両者が推定した分布の差異を低減するように訓練を行う正則化法である [13]。ドロップアウト法と深く関係する主要な技法は文献 [4] で詳しく論じられている。

組合せデザインをはじめとする組合せ構造の計算機科学への応用には、数多くの研究がある [14]。デザインによるバギング [15, 16] は、バギングにおいて組合せデザインの考え方にに基づきデータをサブサンプリングすることで、汎化性能を改善する技法である。バギングは複数の弱分類器で学習を行うアンサンブル学習の一種である。ドロップアウト法もアンサンブル学習と考えることもできるので、デザインによるバギングは、組合せデザインを機械学習に適用する先駆的な試みの一つである。また、組合せデザインと関連が深い分野に実験計画法や品質工学 (タグチメソッド) があり、ニューラルネットワークの最適な構成を求める技法 [17, 18] をはじめ、機械学習での活用が研究されている。

6. おわりに

深層学習における正則化に関しては、数多くの研究がなされてきた。本論文で扱ったドロップアウトデザインを用いた正則化技法は、部分ネットワークの組合せ構造に注目した方法である。無作為にノードを不活性にするドロップアウト法とは異なり、ドロップアウトデザインは、ブロックによってノードの活性化回数を釣り合わせるという特徴を持つ。本論文では、ドロップアウトデ

ザインによる正則化の持つ性質を明らかにするために、MLP の全結合層を正則化する技術を実装した。加えて、画像分類問題での評価実験により、全結合層が 2, 3, 4 層の時の汎化性能を分析した。その結果、いずれの場合についても、ドロップアウト法と同程度の汎化性能が得られた一方で、ドロップアウトデザイン固有の特徴的な結果は得られなかった。これらのことは、訓練時に使用したドロップアウトデザインが持つ組合せ論に基づいたネットワーク構造と、従来のドロップアウト法で得られる無作為な構造との間に、関係性があることを示唆していると考えられる。ドロップアウトデザインの分析を通じて、組合せ構造を用いてドロップアウト法を理解する研究が進むものと期待される。ドロップアウトデザインのニューラルネットワークへの適用法の違いは、実験結果に大きな影響を与えることはなかった。

今後、ドロップアウトデザインとドロップアウト法についてさらに深く理解するために、以下の課題に取り組む必要がある。まず、引き続きより深いニューラルネットワークや、画像分類以外の問題についても実験を進めていく必要がある。また、本論文での評価実験は (2, 1) かつ (1, 2) 型-ドロップアウトデザインに限った結果であり、ドロップアウトデザインの型が汎化性能に与える影響を分析する必要がある。文献 [19] のように、ドロップアウトデザインの構成法の研究は進展しているが、様々な型のドロップアウトデザインを理論的に構成することは現時点では困難である。そのため、従来のドロップアウト法と異なり、ドロップアウト率を自由に設定して訓練を行うことができないという制約がある。そこで、計算機によりドロップアウトデザインを構成する技術を開発することが必要である。計算機を利用して組合せ構造を探索する研究 [20, 21, 22] が参考になると考えられる。最後に、本論文の実験では、ニューラルネットワークのドロップアウトデザインへの適用法の有効性を確認できなかった。今後も有効な適用法を検討する必要がある。

謝辞

本研究は JSPS 科研費 基盤研究 (C) 19K11866 及び、若手研究 21K13845 の助成を受けたものです。

参考文献

[1] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhut-

dinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.

- [2] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [3] Shoko Chisaki, Ryoh Fuji-Hara, and Nobuko Miyamoto. Combinatorial designs for deep learning. *Journal of Combinatorial Designs*, 28(9):633–657, 2020.
- [4] Alex Labach, Hojjat Salehinejad, and Shahrokh Valaee. Survey of dropout methods for deep neural networks. *arXiv:1904.13310*, 2019.
- [5] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- [6] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using DropConnect. In *30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1058–1066. PMLR, 2013.
- [7] Sida Wang and Christopher Manning. Fast dropout training. In *30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 118–126. PMLR, 2013.
- [8] Salman H. Khan, Munawar Hayat, and Fatih Porikli. Regularization of deep neural networks with spectral dropout. *arXiv:1711.08591*, 2017.
- [9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059. PMLR, 2016.
- [10] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 29, pages 1027–1035, 2016.
- [11] Hojjat Salehinejad and Shahrokh Valaee. EDropout: Energy-based dropout and pruning of deep neural networks. *arXiv:2006.04270*, 2020.
- [12] Hieu Pham and Quoc Le. AutoDropout: Learning dropout patterns to regularize deep networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(11):9351–9359, 2021.
- [13] Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. R-drop: Regularized dropout for neural networks. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

- [14] Charles J. Colbourn and Paul C. van Oorschot. Applications of combinatorial designs in computer science. *ACM Comput. Surv.*, 21(2):223–250, 1989.
- [15] Periklis Papakonstantinou, Jia Xu, and Zhu Cao. Bagging by design (on the suboptimality of bagging). *AAAI Conference on Artificial Intelligence*, 28(1), 2014.
- [16] Mahdi Hamdani, Patrick Doetsch, and Hermann Ney. Bagging by design for continuous handwriting recognition using multi-objective particle swarm optimization. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 256–260, 2015.
- [17] F. Sánchez Lasheras, J. A. Vilán Vilán, P. J. García Nieto, and J. J. del Coz Díaz. The use of design of experiments to improve a neural network model in order to predict the thickness of the chromium layer in a hard chromium plating process. *Mathematical and Computer Modelling*, 52(7–8):1169–1176, 2010.
- [18] Ahmad M. Karim, Mehmet S. Güzel, Mehmet R. Tolun, Hilal Kaya, and Fatih V. Çelebi. A new generalized deep learning framework combining sparse autoencoder and Taguchi method for novel data classification and processing. *Mathematical Problems in Engineering*, 2018:3145947, 2018.
- [19] Shoko Chisaki, Ryoh Fuji-Hara, and Nobuko Miyamoto. A construction for circulant type dropout designs. *Designs, Codes and Cryptography*, 89(8):1839–1852, 2021.
- [20] 松中春樹; 丹生智也; 番原睦則; 田村直之. SAT 符号化を用いた釣合い型不完備ブロック計画の構成. *人工知能学会論文誌*, 27(2):10–15, 2012.
- [21] 青柳卓; 宮本暢子; 篠原聡. Optical orthogonal codes の探索 II. *明星大学研究紀要 (情報学部)*, 22:15–22, 2014.
- [22] B. N. Mandal, Rajender Parsad, and Sukanta Dash. Construction of a-optimal balanced treatment incomplete block designs: An algorithmic approach. *Communications in Statistics - Simulation and Computation*, 49(6):1653–1664, 2020.

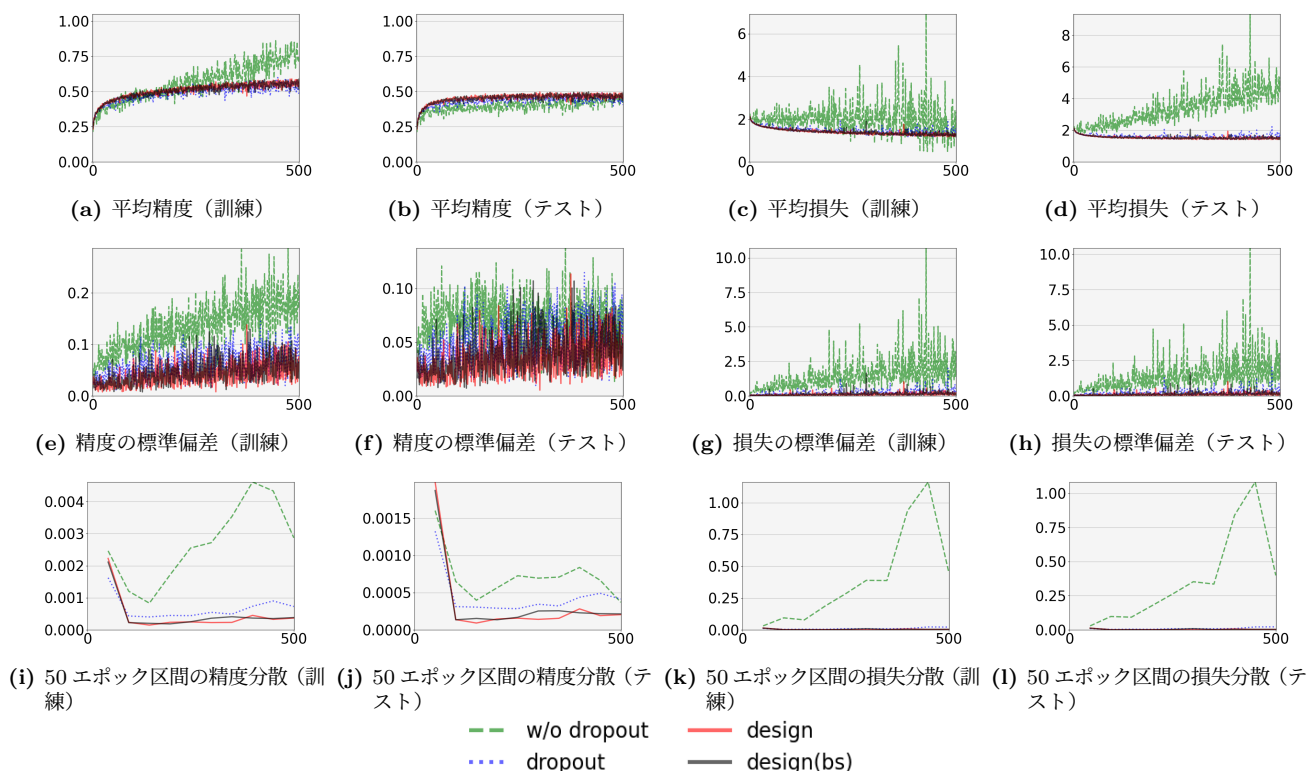


図 1: 全結合層 2 層, 1 層当たり 128 ノード, ドロップアウト率 0.5 の実験結果

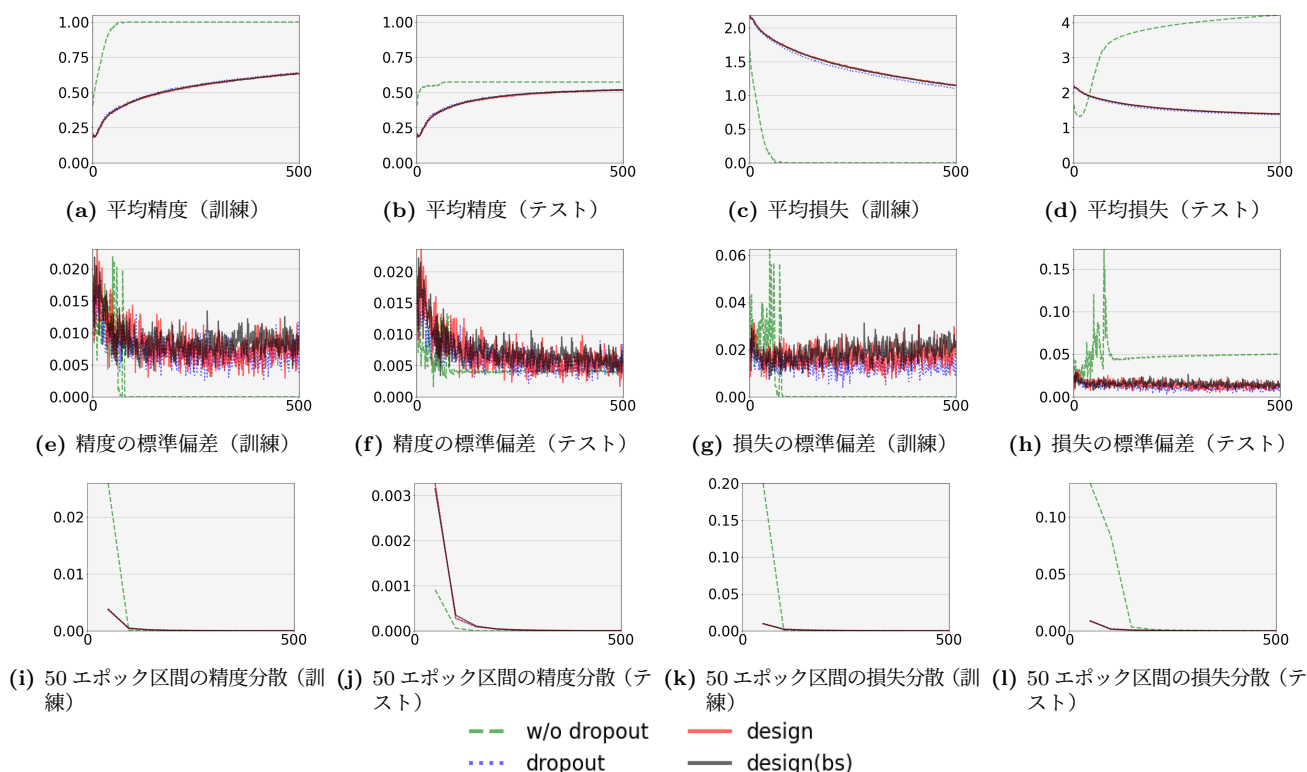


図 2: 全結合層 3 層, 1 層当たり 729 ノード, ドロップアウト率 0.67 の実験結果

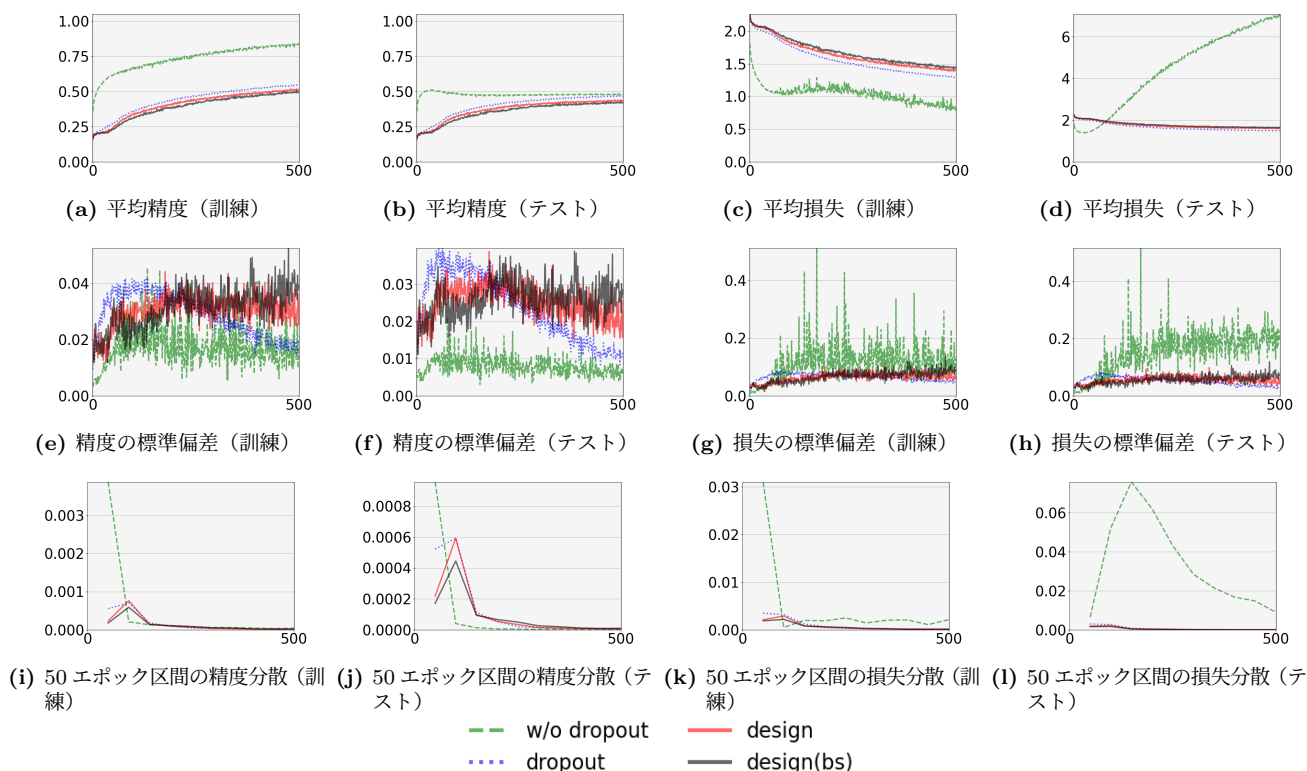


図 3: 全結合層 4 層, 1 層当たり 128 ノード, ドロップアウト率 0.5 の実験結果

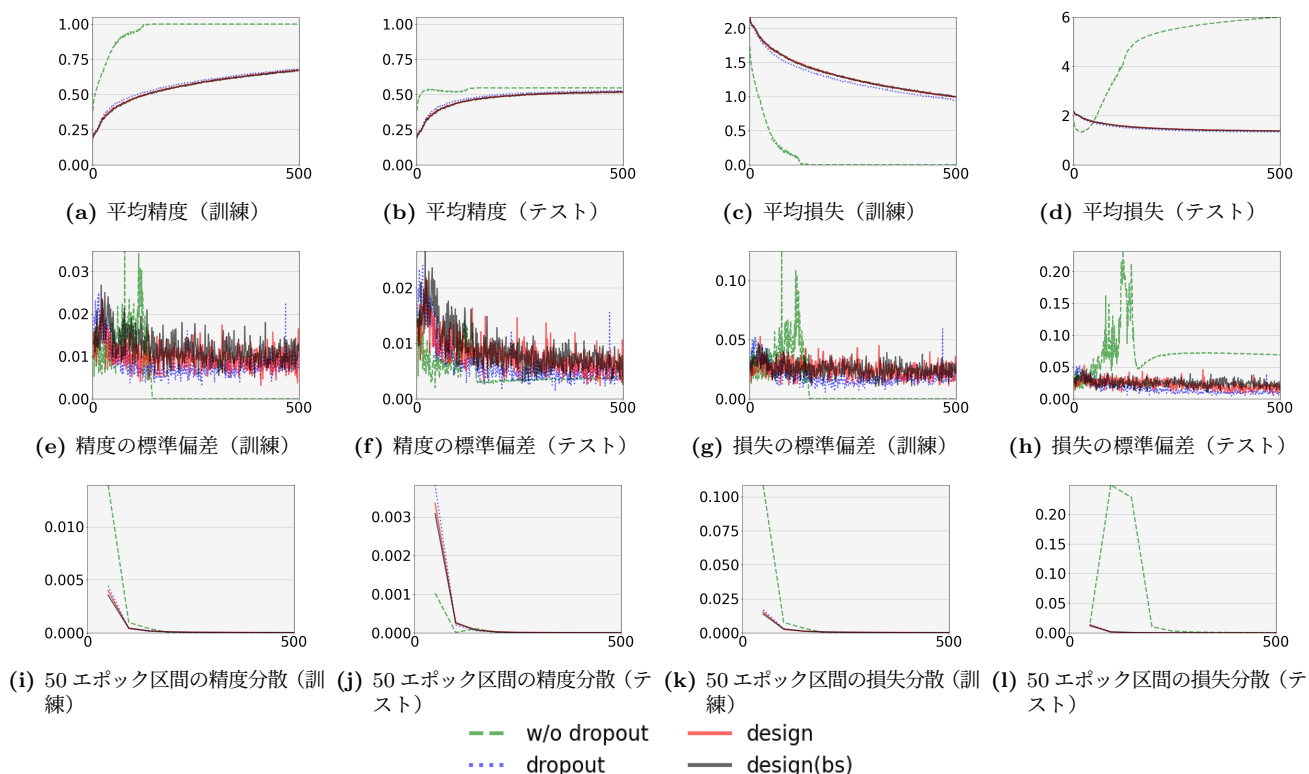


図 4: 全結合層 4 層, 1 層当たり 256 ノード, ドロップアウト率 0.5 の実験結果