

第27回数理科学コンクール課題解説

令和6年11月3日 千葉大学先進科学センター

目次

1	はじめに	2
2	優秀者氏名	4
3	課題 1	7
	解説	8
	講評	8
4	課題 2	10
	解説	11
	講評	11
5	課題 3	12
	解説	14
	講評	15
6	課題 4	16
	解説	17
	講評	17
7	課題 5	18
	解説	19
	講評	19
	課題	21
	講評	22
	昨年度の課題	22
8	課題	24
	解説	25
	講評	31

1 はじめに

明治の文明開化以来、我が国は欧米先進国の科学技術を効率よく吸収して発展してきました。戦後もこの傾向は基本的には変わっていません。現在、我が国は大量の自動車や電子機器を輸出して経済大国となっていますが、これらの工業製品の基本原理はほとんど外国で考えられたものです。欧米諸国との間に経済摩擦や文化摩擦が生じている現状を考えると、これからの我が国で大切なことは独創性のある個性的な人材を育成して、新しい科学技術のフロンティアを切り開き、世界に貢献することであると考えられます。

千葉大学では、日本のみならず、世界の科学技術の先端を担う若者を発掘し、育成するための一助として、本年度も、第27回数理科学コンクールを開催しました。

昨年度から、対面開催でのコンクールを復活しました。それに合わせて、「ロボットの部」も復活しました。過去の経験から、課題の部は、遠隔開催も引き続き実施することとしました。そして、昨年度に引き続き、本年度も「人工知能の部」を開催しました。人工知能の部は完全に遠隔開催としました。

AI、データ・サイエンスに関する報道は落ち着きを見せてきています。AI、データ・サイエンスの手法がいろいろな現場で普通の考え方になってきました。また、AI、データ・サイエンスの課題を取り扱い、解決するためのプログラムも種々公開されています。そこで、千葉大学数理科学コンクールでは、昨年度より「人工知能の部」を開催しました。

数理科学コンクールの主題である、「自ら実験をして現象を考察する。」を引き続き実施するために、遠隔開催参加者には実験機材を送ることにしました。配送できる機材の大きさや種類には制限があるため、皆さんが受け取る機材の中には自分の使いたいものが入っていないかもしれませんでした。しかし、与えられた機材だけを使って、実験の方法を考察して工夫することも科学者にとって重要な訓練です。今回から、対面開催の参加者と遠隔開催の参加者と共通の課題対面参加者のための課題、遠隔参加者のための課題、3種類の課題を用意しました。それぞれ、4つの課題に取り組んでももらいました。対面開催への参加者は従来通りに、コンクール当日の提出締切時間までに、答案を提出してもらいました。また、遠隔開催への参加者は、開催案内に書かれている期日までに解答を千葉大学に返送してもらいました。

ロボットの部は、第1日目にロボットの動作に関するプログラムの講習を受けてもらいました。第2日目(対面開催課題の部実施日)にプログラムを作成し、第2日目の午後3時以降に動作評価のためのコンテストを実施しました。

人工知能の部は、課題を登録したメールアドレスに送信し、解答をメールに添付して提出してもらいました。

このコンクールの特色は次の通りです。

1. 自由にゆったり考える

会場での試験時間は1日、自宅で時間を自由に使い、解答を導く。また、インターネットの検索も条件を付けて可としました。

2. たのしい物理・数学の発見

物理や数学のカリキュラムにとらわれず、物理や数学の本質に根ざした、考えて楽しい問題を提供する。

3. 多彩な才能の評価

様々な参加者の優秀な能力やユニークな発想を多面的に評価するため、問題をたくさん解いたものだけでなく、1題に集中してすばらしい発想を出したのも表彰の対象にする。

4. 人材の育成

コンクール参加者の物理や数学の能力をさらに高めるため、ホームページ上に表彰者の名前と講評を掲載する。

過去26回のコンクールに引き続き、多くの中高生の参加者があり、いろいろユニークなアイデアが生まれました。中学生も、高校生に負けず優秀でありました。そして、答案を見ると、それぞれの問題に興味を持ちながら解答していることが読んでとれました。

第27回数理科学コンクールの課題の解説と提出された答案の評価を以下にまとめます。解説に述べてあるように、各課題は課題出題者の周りにある基本的な問題や最先端の問題、さらには歴史的に意味のある問題を元にして作成しました。課題提出者一同、みなさんの素晴らしい洞察力と表現力を前にして、大変感心いたしました。

参加者の皆さんが今後、科学する心を磨き続け、我国の科学の発展に貢献することを課題作成者一同希望します。今後も諸君と共に科学することを楽しみたいと考えています。千葉大学では今後も引き続きこのコンクールを実施する予定です。物理・数学に興味がある中高生の積極的な参加を期待しています。課題作成者もさらに研鑽をかさね、おもしろく、しかも科学の本質に迫る課題を考えます。

課題作成者

千葉大学教授	石井久夫
千葉大学名誉教授	井宮 淳
東京慈恵会医科大学教授	植田 毅
	(五十音順)

令和6年11月3日

2 優秀者氏名

令和6年7月30日に開催した第27回数理科学コンクール課題の部, 7月29日と30日に開催した第27回数理科学コンクールロボットの部, 8月24日に開催した人工知能の部, それぞれの参加者の皆さんのすばらしい答案の中から以下の参加者諸君を表彰することを決定しました.

第27回数理科学コンクール優秀者

課題の部

金櫛賞	今枝駿斗			
	高野 遥	高澤 静	長山 茉生	
銀櫛賞	萩原叶翔			
	大月真愛			
	井ノ上 要			
	河原挑武			
	加藤結羽	落合 謠恵		
	板橋 祥	谷岡 耀		
	ヘワティラカ ヒルキ	東風平萌絵		
	小島星司	関弥建伸		
	三代川 彪	平澤駿汰		
	福田琥太郎	鈴木翔太	竹味真俐	
	都築凜斗	江口隼平	本山松明	
	坂井ひかり	佐藤瑠泉		
	野田清士郎	高梨吉平		
学長賞	小林杏珠			

ロボットの部

機巧賞	該当なし
敢闘賞	瀧澤 煌

人工知能の部

知巧賞	森川大幹
	伊藤尚太

課題	参加者氏名		
1	高野 遥	高澤 静	長山 茉生
2	今枝駿斗 萩原叶翔 加藤結羽	落合 謠恵	
	ヘワティラカ ヒルキ	東風平萌絵	
	福田琥太郎	鈴木 翔太	竹味真俐
3	今枝駿斗 井ノ上 要 大月真愛		
	都築凜斗	江口隼平	本山松明
	坂井ひかり	佐藤瑠泉	
	野田清士郎	高梨吉平	
4	高野 遥	高澤 静	長山 茉生
	板橋 祥	谷岡 耀	
	小島星司	関弥建伸	
	三代川 彪	平澤駿汰	
5	今枝駿斗 河原挑武 小林杏珠		

事前に了承を得た方の氏名のみ掲載しております。

千葉大学先進科学センター

センター長 教授

千葉大学情報・データサイエンス学部

学部長

千葉大学大学院融合理工学府数学情報科学専攻情報科学コース

コース長

加納博文

教授 塩田茂雄

教授 大澤範高

課題の部

3 課題 1

(対面・遠隔共通課題)

水生生物や草むらに住む昆虫などは触角をもっています。触角で餌を探したり、周りの環境の必要な情報を得ています。触角は、アンテナ状の突起にものが接触したときに電気信号が発生する仕組みになっています。しかし、触角には常に水流や空気、ごみなどが衝突し、必要ではない刺激が入力されます。これらの生物は必要な信号と不必要な信号をどのように区別しているのか数理科学的に考察しなさい。

虫を観察しながら考えたい人は中庭などキャンパス内で昆虫を捕獲して観察してもかまいません。ただし、捕獲の際、熱中症には十分気を付けてください。

解説

食虫植物のハエトリグサは二枚貝のように重なる双葉のような枚の葉があり、その葉の内側には数本ずつの感覚毛が生えています。その感覚毛に獲物が触れると葉を閉じる仕組みになっていますが、風やごみなどが触れたときに反応してしまうとエネルギーの無駄になってしまうので、昆虫などの獲物が2回または2本以上の感覚毛に同時に触れると、約0.5秒で葉を閉じます。これはノイズと信号を区別する仕組みになっています。

この課題では、さらにノイズがひどいときに、どのようにすればひどいノイズの中から信号を取り出せるのか考えてもらいました。実際、ザリガニの尾の部分には有毛細胞があり、その毛が揺れることで水の流れて感じています。しかし、ザリガニは通常、泥の中やひどく濁った水の中に住んでいるので、さまざまな雑音にさらされているはずですが、そのセンサーはしっかり機能しています。それにはどのように工夫、秘密があるのか考えてもらいました。

この問題は、1995年に発表された論文

Wiesenfeld, K., Moss, F. Stochastic resonance and the benefits of noise: from ice ages to crayfish and SQUIDS, Nature 373, 33–36 (1995).

<https://doi.org/10.1038/373033a0>

で説明されています。この論文の主旨はノイズがある方がセンサーの感度がよくなるというもので、確率共鳴と呼ばれます。この論文の解説記事が

日経サイエンス 1995年10月号 p.126-131, F. モス/K. ビーゼンフェルト「役に立つノイズ」

に掲載されているのでこれを参考にするとわかりやすいでしょう。

ノイズの原因もセンサーの反応ですから、信号とノイズの大きさは同程度のはずで、そのレベルの信号に反応してはノイズばかりに反応してしまうことになります。したがって、センサーから来た信号やノイズには通常反応しない設定になっているはずですが、では、そのような設定でどうやって必要な信号を感知するのでしょうか？

触覚からの信号を検知するレベルを1として、ノイズだけの場合（紺色）、獲物の周期的な運動による信号だけの場合（オレンジ色）には反応しないとすると、触覚からの電気信号は図1のような状態（イメージ図）であると考えられます。

ノイズが存在している状態で周期的な獲物の信号を受け取ると、触覚からの電気信号は図2のようになり（イメージ図）、獲物の信号の周期を反映した信号を感知できることが分かります。

講評

課題文に「電気信号が発生する」とあるのですが、雑音のある電気信号の中から意味のある信号をどのように取り出すかという方向性で考えた答えは皆無でした。そもそも、電気信号となると難しかったのか、この課題にチャレンジしてくれた答えは僅かでした。少ないながらも、ゴミなど雑多な無意味な信号から意味のある信号を取り出せそうな方法を考えていれば評価しました。

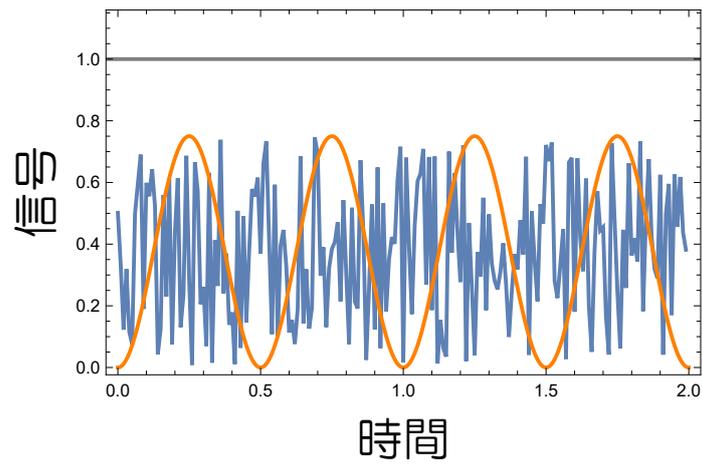


図 1: ノイズ (紺色), 獲物の周期的な信号 (オレンジ色) のイメージ図

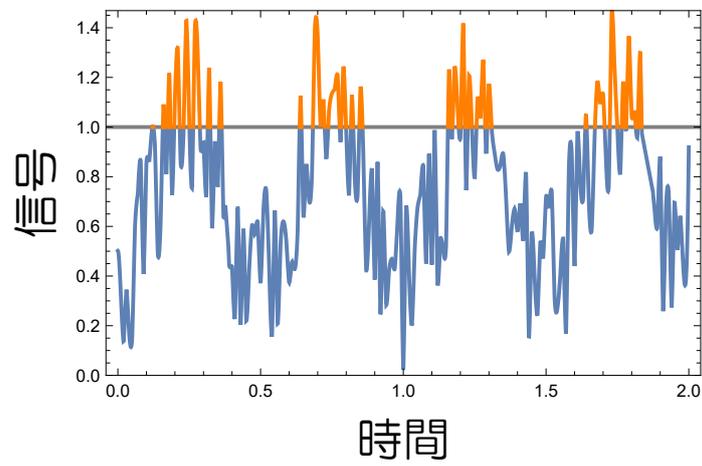


図 2: ノイズが存在している状態での周期的信号の周期を反映した信号を感知

4 課題 2

(対面・遠隔共通課題)

同じ大きさの正6角形を平面上に並べた構造をハニカム (Honeycomb) 構造 (蜂の巣構造) と言います。その名の通り, 自然界では蜂の巣の構造に観察されます。また, 幾つかの植物の花弁の構造に観察されます。この構造は工学的に, 軽量で丈夫な構造のため, 航空機の内壁, 金属扉の中充などに利用されています。

実際の蜂の巣では, 蜂の巣構造は, 繭の収容場所に成っています。蜜蜂の場合, 女王蜂候補の繭の入る6角単位は, 働蜂の区画よりも大きな6角形の区画になります。しかし, 蜜蜂の巣の構造は全体として殆ど崩れていません。そこで, 辺の長さの異なる6角形が含まれる場合に, 平面上に蜂の巣構造を作る方法と, その数学的性質を考察してください。

解説

蜜蜂の巣は6角形を敷詰めた構造をしています。この構造をHoneycomb構造と呼んでいます。Honeycomb構造はどうして起こるのでしょうか？ある種の最小化がなされているのではないかと、長い間、未解決の問題でした。そのため”Honeycomb Conjecture”(ハニカム予想)と呼ばれて来ました。1999年に出版されたThomas C. Halesの論文によって「平面を同じ大きさの正六角形で区切ると、辺の総延長が最小になる。」ことを証明しました。このことを工学的に言い直すと「平面を最小限の仕切りで均等に小部屋に分割するとき、各部屋を正六角形にすると、壁の総延長(分割の材料の長さ)が最短になる。」と成ります。蜜蜂は最小の手間で巣を構築していることに成ります。

課題は、Honeycomb構造に大きさの異なる6角形を入れるときに規則正しい分割がどのようになるかを問うています。5角形と7角形を対にして割込ませれば、辺の総数を変わりません。辺の長さが異なるHoneycomb構造を接続させる場合には接続部分に5角形を入れれば、2つのHoneycomb構造を接続できます。

講評

インターネットの中に、この問題をVoronoi分割を利用して説明した記事があります。その記事を引用した解答がありました。しかし、記事にも解答にも、どうしてVoronoi分割が利用されるのかの説明がありません。

平面のVoronoi分割を計算するためには、あらかじめ母点の位置を決めておく必要があります。大きさの異なる6角形を挿入する場合、どのように母点を決めるのかを、解答には付加える必要があります。母点の位置を区画に入る幼虫の代表点とすれば、代表点の勢力圏(縄張り)としてVoronoi分割が現れます。Voronoi分割が決める領域は、隣の点と結ぶ直線を、垂直2等分線を辺とする凸多角形になります。この点に関する解答はありませんでした。

5角形と7角形と対がHoneycomb構造に挿入されると、辺の長さの総延長がどのように変動するのか等を解析する必要があります。

5 課題 3

(対面・遠隔共通課題)

変数 x, y を 0 又は 1 とする. 0 と 1 とに 3 種類の演算

$$\begin{aligned}0 + 0 &= 0, & 0 \times 0 &= 0, \\0 + 1 &= 1, & 0 \times 1 &= 0, & \bar{x} &= 1 - x \\1 + 0 &= 1, & 1 \times 0 &= 0, \\1 + 1 &= 1, & 1 \times 1 &= 1,\end{aligned}$$

を定義します. この演算をブール演算と言います.

ブール演算によって構成される変換に関する以下の問に答えてください.

問 1

関数

$$e = x \oplus y = x\bar{y} + \bar{x}y$$

の入出力関係を表を利用して表現しなさい. 例えば, 3 変数関数

$$m = x \times y + y \times z + z \times x$$

の入出力関係は

x	y	z	$u = x \times y$	$v = y \times z$	$w = z \times x$	$m = u + v + w$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

なる表を完成することで得られます. また, 出力が複数ある関数

$$\begin{aligned}m &= x \times y + y \times z + z \times x \\n &= x \oplus y + y \oplus z + z \oplus x\end{aligned}$$

の入出力関係も同様に表を利用して記述できます.

問 2

0 又は 1 を値とする 3 変数 x_1, x_2, x_3 から 3 変数 y_1, y_2, y_3 への変換

$$\begin{aligned}y_1 &= \bar{x}_3 \times x_1 + x_3 \times x_2 \\y_2 &= x_3 \times x_1 + \bar{x}_3 \times x_2 \\y_3 &= x_3\end{aligned}$$

の入出力関係を表に示さない。

また、 x_3 を 0 にすると、出力 y_2 の表現はどうなりますか。さらに、 x_3 を 0 にすると、出力 y_1 の表現はどうなりますか。加えて、出力 y_2 を \bar{x}_3 とするにはどうすれば良いでしょうか。

問 3

0 又は 1 を値とする 3 変数 x_1, x_2, x_3 から 3 変数 y_1, y_2, y_3 への変換

$$y_1 = x_1$$

$$y_2 = x_2$$

$$y_3 = (x_1 \times x_2) \oplus x_3$$

の入出力関係を表に示さない。

問 4

問 2, 問 3 の変換は共に出力 y_1, y_2, y_3 が与えられれば、入力 x_1, x_2, x_3 を復元できることを示さない。

問 5

0 を否 1 を可、 \bar{x} を否定 (否が可となり、可が否となる。) 表しています。また、 \times は且つ (and) を、 $+$ は又は (or) を表しています。このことから、問 4 で示した性質は何を示しているか数学的に論じてください。

解説

論理演算の基本は、否定 (not), 論理 (or), 論理積 (and) です. 3つの演算中で、否定の否定は肯定ですから、否定演算は可逆演算です. しかも、否定自身が逆演算子です. 他の2つの演算は、2入力1出力の関数と考えれば、逆を計算できない演算です.

しかし、論理和、論理積を3入力3出力の関数として表現すると、可逆な関数と成り、しかも全ての論理演算が構成できることが知られています. そのような関数対応として Fredkin-gate と Toffoli-gate があります. これらを可逆ゲートと言います. その動作原理はそれぞれ、

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \overline{x_3} & x_3 & 0 \\ x_3 & \overline{x_3} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$
$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & x_1 \overline{x_3} & \overline{x_1 x_2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

であり、 3×3 行列で表すことができます. この中から、適切な出力を1つ選べば、通常の論理演算の入出関係を記述できます.

論理が可逆であることが、示されます. 可逆論理、可逆ゲートは量子計算と密接な関係があることが分かっています. しかし、可逆ゲートは量子計算機の出現以前から提案され、その性質について研究されてきました.

可逆素子は論理素子の組合せで実現できます. そして、論理素子は閾値論理素子によって実現できます. 従って、可逆素子は閾値論理素子によって実現できます. ここで、閾値論理素子とは、 $w_i \geq 0$ に対して、入出関係が

$$y = u \left(\sum_{i=1}^n (w_i x_i - w_{n+1}) \right), \quad u(s) = \begin{cases} 1, & s \geq 0 \\ 0, & s < 0 \end{cases}$$

で構成される素子です. 確かに $x_1, x_2 \in \{0, 1\}$ に対して、

$$u(x_1 + x_2 - 1) \Rightarrow x_1 + x_2$$

$$u(x_1 + x_2 - 2) \Rightarrow x_1 \times x_2$$

$$u(2x_1 - 1) \Rightarrow \overline{x_1}$$

です. ここで、左辺の四則演算は通常の実数に関する四則演算である. 一方、右辺の演算は課題で定義した演算である.

McCulloch-Pitt は、この素子の組合せによってニューラルネットワークが構成できると主張しました. さらに、 $u(s)$ を微分できる関数例えば、

$$u(s) = \frac{1}{2} \left(\frac{e^s - e^{-s}}{e^s + e^{-s}} + 1 \right) = \frac{e^s}{e^s + e^{-s}} = \frac{1}{1 + e^{-2s}}$$

に置換えることを Hopfield や Rumelhart-Hinton-Williams は提唱してきました。現在の主流は

$$u(s) = \frac{1}{2}(x + |x|)$$

と、条件に合わせてこの関数を適切に変形した関数です。

講評

論理関数を計算する問いには解答していました。しかし、論理の可逆性を明確に説明した答えはありませんでした。

6 課題 4

(対面課題)

車体に動力装置を搭載しない搬送車を考え、その動作原理を科学的に解析し説明してください。また、考えた装置を実現する上で、物理的、工学的に問題点がある場合は、その問題点を指摘し、解決策を考えてください。例えば、リニアモーター鉄道には移動用の動力装置を載せない形式も存在します。

解説

この課題の参考にした事実は Isambard Kingdom Brunel が考案した大気圧鉄道です。これは、管の中を空気圧で、動力部分を動かし、その上に付けた搬送車を引っ張るものです。実用化されましたが、管の中の部分と搬送車との連動のために、管の搬送車側に切れ込みを入れ、その切れ込みをシールドして管の中の空気が漏れないようにしていました。この方法は効率が悪いのですたれていきました。

現在、最先端の方法は、管に切れ込みを入れる代わりに、搬送車と管の中の運動分とを磁力で結びつけるものです。空気圧で、荷物を輸送する手段として気送管 (pneumatic tube) と呼ばれる装置がしばらく前まで、病院内で処方箋の伝達に使われていました。

搬送車に動力を載せないのであれば、搬送車をロープで引っ張ることも考えられます。エレベータを横にした搬送車です。かつて、成田空港で運用されていました。

坂を登攀する場合だけ、搬送車外の錘を使うのであれば、水を容器にためて錘として利用することもできます。ドイツやスイスに 100 年以上運用されている水力ケーブルカーがあります。

また、風を利用することもできます。現在 SailGP と呼ばれる高速ヨットレースで利用される F50 ヨットのメインセールは飛行機の主翼のような構造をしています。風上に向かって進むことができない範囲が、普通の帆だけのヨットよりも狭くなっています。

講評

解説で述べた手法をいろいろ作成して解答をしています。重力、電磁気に着目した解答はありました。しかし、大気圧鉄道と結び付けた解答はありませんでした。

数学、物理、化学、最近では生物学が実社会にどのように利用され、問題解決のための新しい学問分野が作られたかを眺めることができます。熱力学の体系化は、産業革命時の蒸気機関の効率化から要求された学問分野です。インターネットに続く通信技術は、その始まりは古代戦争での狼煙に拠る合図に、暗号は軍事司令の秘匿化に、個人識別は英国によるインド亜大陸植民地支配に端を発しています。

インターネットの中の情報は全てが正しいとは限りません。しかし、インターネット上で検索できる科学史や歴史の項目は、図書館、博物館に蓄えられた人類の膨大な知恵や知識に辿り着く入り口には成ります。また、海外のドキュメンタリー番組も図書館、博物館に蓄えられた情報に辿り着く手助けに成ります。教科の勉強だけでなく、幅広く勉強してください。

7 課題 5

(遠隔課題)

分厚いスポンジを綺麗に切るにはどうすればよいか科学的に考えてください。考えた方法で実際にスポンジブロックから3角柱, 6角柱のスポンジを切り出して, 写真(切り口などの十分な解像度があれば普通紙にプリンターで印刷したもので可.)を提出してください。資料のスポンジブロックは厚さ7cmです。

刃物などを用いるときには怪我をしないよう十分に注意してください。

解説

ルパン三世に出てくる石川五右衛門の斬鉄剣は何でも切れてしましますが、コンニャクは切れません。コンニャクのように粘弾性が切りにくいのは確かですが... スポンジはさらに大小の穴の開いたポーラス構造をしているので余計厄介です。

スポンジは押さえて、潰せば、横方向に膨らむので、この状態で切ると、力を抜いたときに、どのように戻るのかが問題になります。膨らんだまま、切ると、厚みの中ほどがえぐれたような形になったりします。

私が実際に、学生の「応力とひずみの実験」の試料となるスポンジを用意しているときに直面した問題でした。

スポンジ構造のものでも、スポンジケーキであれば、ワイヤー等で簡単に切ることができます。しかし、繊維の弾性が大きく強靱なウレタンスポンジを切るのは難しい。大掛かりな装置を用いるなら、レーザーカッター、ウォータージェット切断機を用いれば様々な形にカットできると思います。が、もっと、身近なものできれいに切ることはできないのか？考えてもらいました。

まず、工場でどのように切っているのか？大きなものについては次のような動画があり、ベルト式ののこぎりで切っていることが分かります。ただ、この切り方ではサイズの小さなものを正確に切るのは難しいかもしれません。

<https://www.youtube.com/watch?v=nppZkl7zZ2o>

Precision Cutting of 40mm EVA Foam with JWEI Digital Cutter

<https://www.youtube.com/watch?v=xVDE3oWb0VE>

Flexible Foam Machine, Vertical Cutting Machine

<https://www.youtube.com/watch?v=kybyI8Y4Fv0>

Foam cutting machine vertical — PU Foam cutting machine vertical

<https://www.youtube.com/watch?v=RLui8x4K-Yc>

Foam Cutting machine Circular Cutting Machine by AS Enterprises

講評

様々な方法を考えてくれていました。まず、パン切包丁で切るというもの。この方法も波打った刃が切りつけがよく、カッターなどよりもきれいに切れます。スポンジを押しつぶして切る試みをした答案が複数ありましたが、あまりうまくはいっていませんでした。変形しないように水を含ませ凍らせて切ろうという試みもありました。氷を切れる電気ノコギリがあればうまくいったかもしれません。それでは、簡単に切れるとは言い難いので、より切り易くするために、一旦、水を含ませたスポンジを絞り、水分を少なくして(繊維にまとわせる程度にして)、凍らせ、切ろうとした試みもありました。氷は固いということで、凍らせても比較的柔らかいポリマー(樹脂)を染み込ませる方法を考えた答案もありました。更に考えを一步進めて、切ったスポンジを利用することも考え、染み込ませるポリマーの取り除き易さも考慮した解答もありました。切断面の平滑度、正確さに加え、考え方の論理的正しさも評価しました。

ロボットの部

第1日目の実習に続いて、第2日目に後に以下の課題を課しました。第2日目 15:00 から競技を開始しました。

課題

1階の自習室に、ロボット (RAPIRO) が用意してあります。各自でロボットの動作課題を考え、その動作を実現しなさい。どのような動作課題を考えたかレポートを作成してください。動作の独創性、面白さを評価します。動作実演は7月30日の午後3時に始まります。それまでに、動作プログラムを完成させてください。

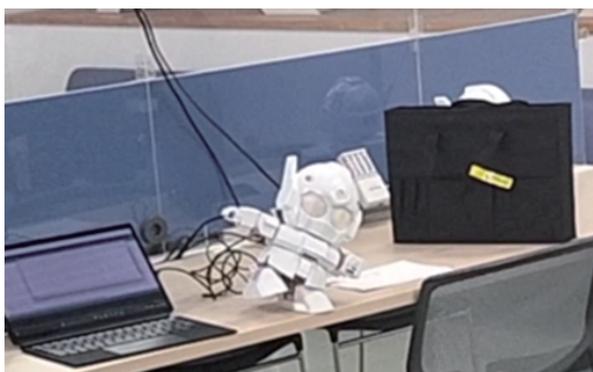


図 3: Rapiro が負けた時の残念ダンスの様子

講評

今年度の参加者が2名しかいなかったのは残念でした。人数が少ないので昨年のような共通課題でのコンテストではなく、自由演技で競ってもらいました。1名は、昨年度の課題だったミニ迷路を抜ける問題に取り組んでももらいました。距離センサーで壁を認識しながら迷路を抜ける課題でしたが、Rapiroをそもそも真っすぐに歩かせること自体が意外と難しく、いろいろと工夫が凝らされていました。途中少しだけやり直しもありましたが、ミニコースを無事に1週することができました。もう一人の参加者は、Rapiroとジャンケンをするプログラムを作ってくれました。人間側が出すグー・チョキ・パーを直接Rapiroが認識することはできないので、Rapiroの前にもものをおくとか、Rapiroを持ち上げるなどの動作で、人間の出した手をRapiroに伝えました。それを受けて、人間が出した手を正しく認識していることをRapiroの眼の色を変えることで確認し、そのあと、Rapiroがジェスチャーでグー・チョキ・パーを示します。Rapiroが勝つと勝利のダンスを披露してくれました。Rapiroの手は指が2本しかありませんので、体全体を使ったジェスチャーでジャンケンを表すなどの工夫がされており、Rapiroの容姿とマッチしてかわいらしいパフォーマンスでした。なんどやってもRapiroが勝ち続けるのでプログラムの不備が危惧されましたが、最後に無事(?)負けて、残念ダンスも見せてくれました。図3にRapiroが負けた時の残念ダンスの様子を示します。今回は、このジャンケンRapiroに敢闘賞を出しました。

昨年度の課題

課題 1 1.5mの直線走路をできるだけ早く走行できるように、プログラムしてください。但し、テープで示されたレーンからRapiroの両足がはみ出ると失格と成ります。

課題 2 床にテープで描いた迷路を通してできるだけ短時間でゴールに到達するようにプログラムしてください。但し、テープで示されたレーンからRapiroの両足がはみ出ると失格と成ります。

課題 3 Rapiroをプログラムしてダンスや体操などのパフォーマンスをさせてください。但し、Rapiroを制御する技術面と、パフォーマンスの印象面を総合して判定します。

人工知能の部

8 課題

眩暈 (めまい) や 頭部の打撲, 捻挫骨折の疑いがある場合には X 線 CT 検査が行われます. X 線 CT 検査は日本人の我々にとって, 非常に身近な検査となっています. それは, X 線 CT の装置は全世界 (OECD) の約 30% が日本にあり, 日本の 100 万人あたりの台数は 107 台で 2 位のオーストラリアの 56 台の倍近く (OECD health care activities 2019) あり, 実際に世界で一番身近になっています. ただし, 1000 人あたりの撮影回数は 176 回で OECD の中で 6 番目であり (OECD の平均は 140 回), 無暗に撮影しているわけではありません. とは言え, CT 検査による被曝量は, 1 回あたり 5~30mSv 程度で, 胸部 X 線撮影の 0.06mSv 程度に比べると被曝量は格段に大きいと言わざるを得ません. 画像を高精細化するためには多方向から数多くのビームを照射する必要があります. さらに, 一般に線量が増えると画質はよくなります. 一方で, 2021 年に通常の MRI の 1/100 程度の磁場でシールドが不要な MRI が香港の放射線技師によって開発されました [1]. この装置では, AI を用いて画像を構成しています. そこで, X 線 CT の被曝量を減らすため, 照射するビームの本数を減らして, AI を用いてどこまで画像を再現できるのかを考えます. Wolfram 言語 の機械学習機能を用いて, ビームの照射方向を減らしたデータすなわち, 斜めの照射ビームのデータをなくした, 縦横のビーム [2, 3] の透過率データのみから内部の吸収率の分布をできるだけ高精細に再現してください.

[1] Liu, Y., Leong, A.T.L., Zhao, Y. et al.: A low-cost and shielding-free ultra-low-field brain MRI scanner, Nature Commun 12, 7238 (2021)

<https://doi.org/10.1038/s41467-021-27317-1>

[2] Herman, G.T., A. Kuba eds.: Discrete Tomography: Foundations, Algorithms, and Applications. Birkhäuser Boston, MA (1999)

<https://doi.org/10.1007/978-1-4612-1568-4>

[3] Herman, G.T., A. Kuba eds.: Advances in Discrete Tomography and Its Applications, Birkhäuser Boston, MA (2007)

<https://doi.org/10.1007/978-0-8176-4543-4>

元画像は $n \times n$ の 2 次元の正方形の画像とします. その画像から縦 n 本, 横 n 本のビームの透過率を計算し, 元の画像を推定させるための元データを $n = 4, 6, 8, 10, 12, 14, 16, 18, 20, 30$ の場合について用意してあります. データは図の $\{h_1, v_1, h_2, v_2, \dots, h_{n-1}, v_{n-1}, h_n, v_n\}$ の順に並んでいます. このデータから元画像をできるだけ正しく推定してください. どのように教師データを発生させたか, 教師データの数による変化についても調べてください. 複数の方法による比較を推奨します.

解説

この課題に似た問題を 2021 年度 第 24 回数理科学コンクールの課題 1

https://www.cfs.chiba-u.ac.jp/events/math/2021/R03_kadai.pdf

に出題していました。過去の課題を見ていれば、この課題が数学的にかなり厳しいものであるイメージが湧いたのではないかと思います。それを AI が何とかしてくれるかということです。

まず、どのような画像の再現に挑んでもらったかという点、元の画像は別紙 1 にある通りです。今年の課題は Mathematica を用いてもいくつか解く方法が考えられます。

この課題を解くためには、まず、教師データを自分で用意しないとけません。そのためには、いろいろな図を生成し、それに対する縦方向、横方向の透過率のデータを与えられたデータの順序で作成します。透過率のデータは図 4 の $\{h_1, v_1, h_2, v_2, \dots, h_{n-1}, v_{n-1}, h_n, v_n\}$ の順に並んでいます。

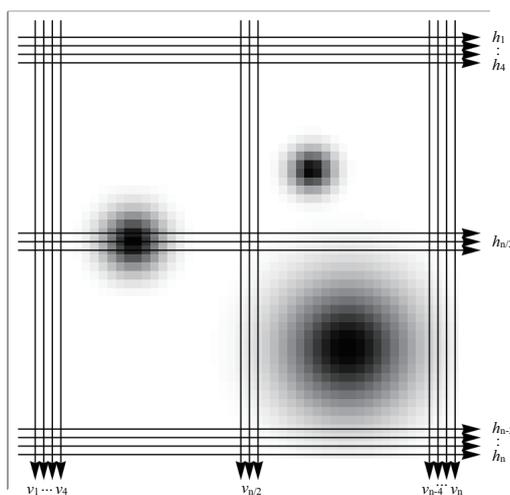


図 4: 縦横方向の透過率の定義

図と縦横の透過率を対応させた学習用データセットを用意します。

さて、推定させるための方法の例を 2 つ紹介します（あくまで例です）。

まず、簡単な方法として、Predict 関数を用いる方法を説明します。ただ、Predict 関数は入力としてはデータの配列（ベクトル）や画像を入力できますが、推定できるのは 1 つの数値のみなので工夫が必要です。画像のデータは $n \times n$ の各点の吸収率で表現できるので、同じ縦横の透過率データに対して、2次元の座標 (i, j) それぞれの吸収率を対応させて学習させます。その意味で、同じ吸収率のデータについて、 $n \times n$ 回学習させる必要があります、非常に効率が悪い。この学習したデータに、推定用に与えられた縦横の透過率のデータを入力し、それに対応した 2次元の座標 (i, j) それぞれの吸収率を推定させ、画像として表示します。

教師データの画像を発生させるとき、各座標の吸収率を乱数で発生させると、推定する画像も砂嵐のような画像になってしまうので、教師データの元画像の生成には工夫が必要です。この計算例では、2次元の正規分布（釣鐘状の関数）の数、中心位置、広がり具合（標準偏差）を乱数で発生させたものを数だけ加え合わせて図を生成させています。

また、Predict 関数では Method を指定でき、Method には LinearRegression, NearestNeighbors,

RandomForest, NeuralNetwork, GaussianProcess, GradientBoostedTrees, DecisionTree, Automatic を指定できます。どの Method が最も良い元画像の推定をしてくれるのか、比較してみなければなりません。最も簡単には、何も指定しなければ Automatic でその都度最適と判断されるものが採用されます。

$n = 4$ の場合のプログラムの例を別紙 2 に示します。

この方法でどの程度正解に近い画像を推定で来たのか、Mathematica に自己採点させてみましょう。教師データの数が異なる場合に推定された画像を特徴量空間にプロットしてみると図 5 のようになります。

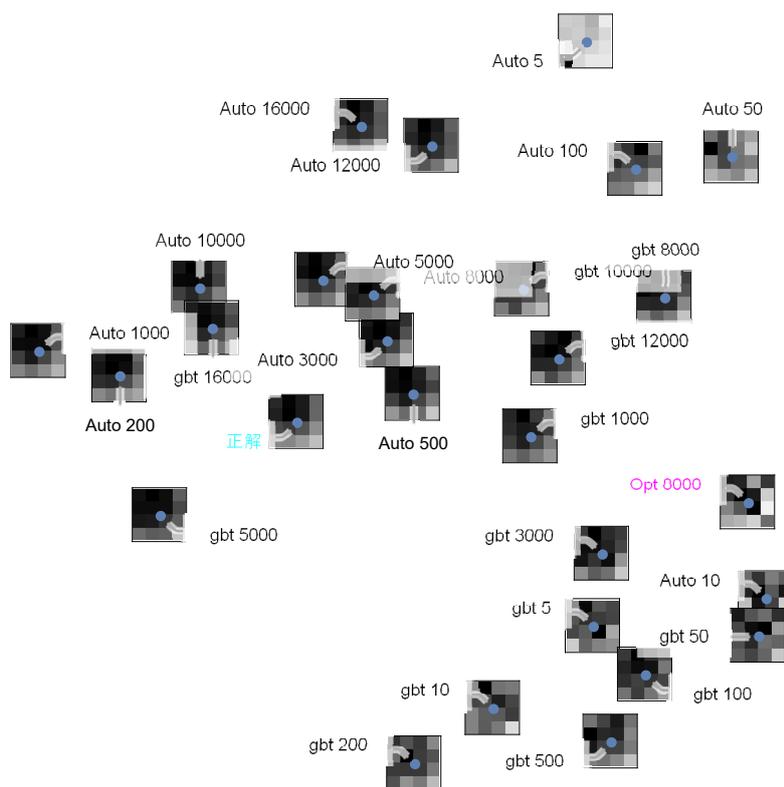


図 5: $n = 4$ の場合に様々な教師データ数について推定された画像の特徴量空間プロット。

図には、Method が Automatic (Auto) , GradientBoostedTrees (gbt) の場合の推定画像を正解とともに示してあります。Method の横の数字が教師データ数です。Opt 8000 はモンテカルロ法を用いた通常の最適化法によって求めた推定画像です。

Auto の場合、200 ほど教師データがあればかなりよい推定ができていることが分かります。しかし、教師データの数が触れるとよい推定画像が得られるかというところはなっていないことが分かります。しかし、Method が GradientBoostedTrees の場合には教師データ数が多い方がよい推定になっています。このような性質を理解して機械学習機能、それぞれのオプションを利用する必要があります。

この問題は、画像の分割数 n が増加するに従い、未知変数の数（推定する画素数）に対する与えられる方程式の数の割合が小さくなり、より推定が困難になります。

分割数が $n = 30$ の場合、どのような推定結果になるかを図 6 に示します。

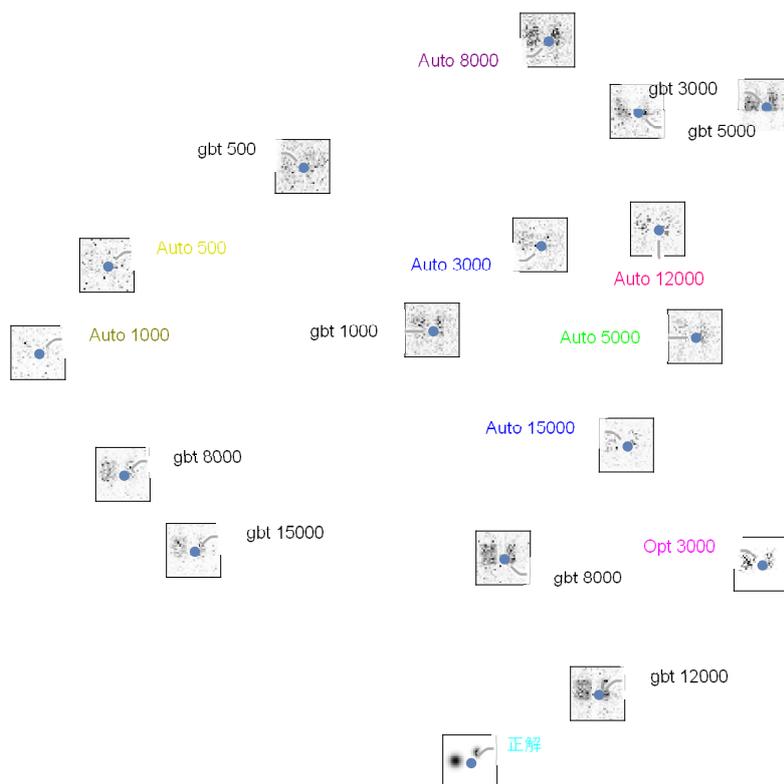


図 6: $n = 30$ の場合に様々な教師データ数について推定された画像の特徴量空間プロット。

この場合、Automatic よりも GradientBoostedTrees を用いて教師データを増やすとよりよい推定画像が得られることが分かります。しかし、この FeatureSpacePlot の結果そのものにも疑問があり、人間の目で判断すると Opt 3000、つまり、従来の最適化法で推定した画像が最も正解に近いように思えます。機械学習よりも、従来の最適化法の方がこの問題に関してはまだまだ実力は上なのかもしれません。

ここまでは、FeatureSpacePlot を用いて Mathematica に自己評価させましたが、より客観的な評価基準が必要です。この場合、画像が似ているかどうか、どのような数値で評価したらよいでしょうか？このようなことを考えることも重要な研究の要素です。

ここでは、画像は吸収率の 2 次元配列で表現されています。そこで、2 次元配列の数値を順番に一行に並べ、 $n \times n$ 次元ベクトルで表現し、正解との差のノルム（それぞれの位置での吸収率の差の 2 乗を全て足して平方根を取ったもの、同じであれば 0 になる）で評価してみましょう。ただ、分割数を変えた場合にも同じように評価できるように、ノルムを分割数で割ったもの（規格化画像残差ノルム）で評価します。

$n = 4$ のとき、それぞれの Method で推定した画像の規格化画像残差ノルムが教師データ数に対してどのように変化するかを図 7 に示します。

図 7 を見ると、どの Method でも教師データ数が増えると規格化画像残差ノルムは減少し、 $n \sim 500$ で収束していることが分かります。それ以上に教師データ数を増やしても、0.25 近傍を中心に揺らい

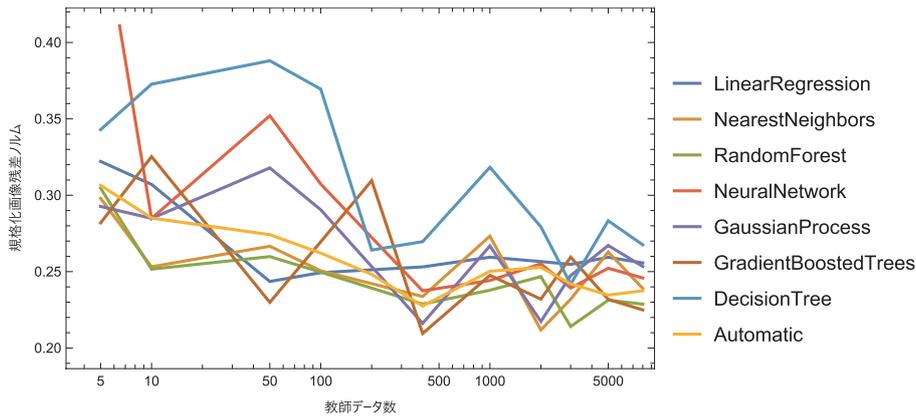


図 7: 規格化画像残差ノルムの教師データ数依存性 ($n = 4$)

で、規格化画像残差ノルムの値が減少することはありません。これが FeatureSpacePlot の結果に表れていると思われます。

また、この推定では推定に用いた透過率データを必ず満たすという条件は課されていないので、推定した画像から得られる透過率データがどの程度元の透過率データに一致しているのかが重要です。透過率データは 1 次元配列 ($2n$ 次元ベクトル) で与えられているので、推定した画像に対しても同様に透過率データを計算し、推定用に与えられた透過率データベクトルとの差をとり、そのノルムを計算し、 \sqrt{n} で割ったものを規格化透過率残差ノルムとして定義します。それぞれの Method で推定した画像の規格化透過率残差ノルムが教師データ数に対してどのように変化するかを図 8 に示します。

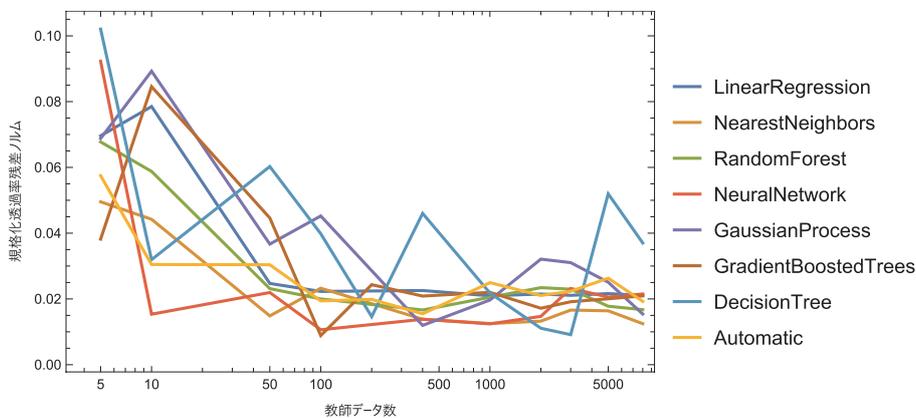


図 8: 規格化透過率残差ノルムの教師データ数依存性 ($n = 4$)

規格化透過率残差ノルムの教師データ数依存性も規格化画像残差ノルムと同様で、 $n \sim 500$ で収束しています。教師データ数をそれ以上に増やしても 0.02 近傍で揺らいで、更に減少することはないことが分かります。

また、どの Method がよい結果となるかは、場合により、特定のものがよい結果を与えるというわけではないことが分かります（少なくとも、LinearRegression は全体により結果を与えない傾向にはあります）。

分割数が増えた場合の例として、 $n = 30$ の結果を図 9, 10 に示します。 $n = 30$ となると、教師データの数が 5000 以上ないと収束しないことがわかります。

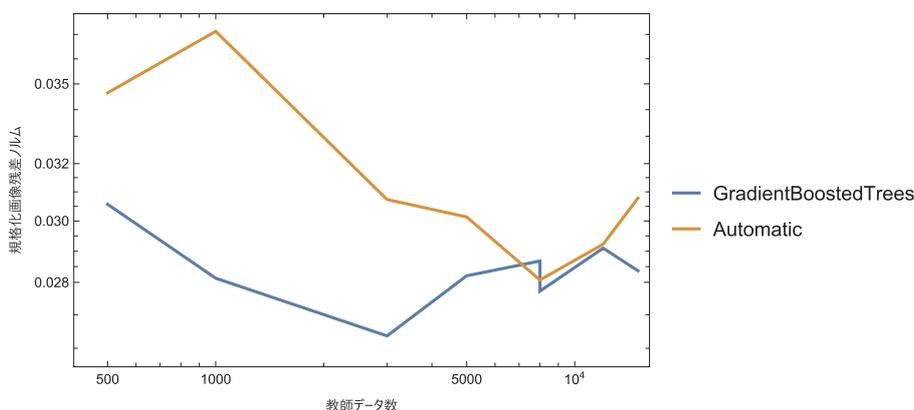


図 9: 規格化画像残差ノルムの教師データ数依存性 ($n = 30$)

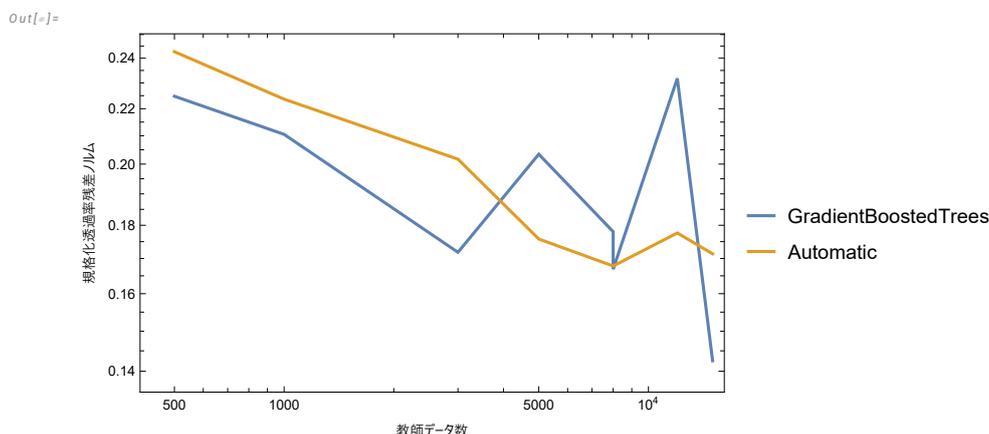


図 10: 規格化透過率残差ノルムの教師データ数依存性 ($n = 30$)

Automatic を用いた場合に、規格化画像残差ノルムおよび規格化透過率残差ノルムの教師データ数依存性の分割数に対する変化を図 11, 12 に示します。

規格化透過率残差ノルムは分割数 n が小さいほど値が小さいが、規格化画像残差ノルムは必ずしもそのようになってはいません。規格化画像残差ノルム、規格化透過率残差ノルムともに n が増えるに伴って収束に必要な教師データの数が指数関数的に増えることもわかります。

Predict 関数を用いた方法では、教師データの学習を n^2 回行う必要があります。また、画像を再現する場合も、 n^2 回 Predict する必要があります。これを回避するために、NetChain, NetTrain という関数を用いる方法があります。NetChain はニューラルネットワークを構成する層の数、各層の入力の数、出力の数を指定し、ニューラルネットワークを定義する関数です。NetTrain は用意したニューラルネットワークを指定した教師データで訓練する関数です。これを用いれば、ベクトルの透過率データから推定した各位置の透過率をベクトルとして出力が可能になり、学習を 1 回で終わることができます。ただし、ネットワークの自由度を上げると、その分多くの教師データが必要にな

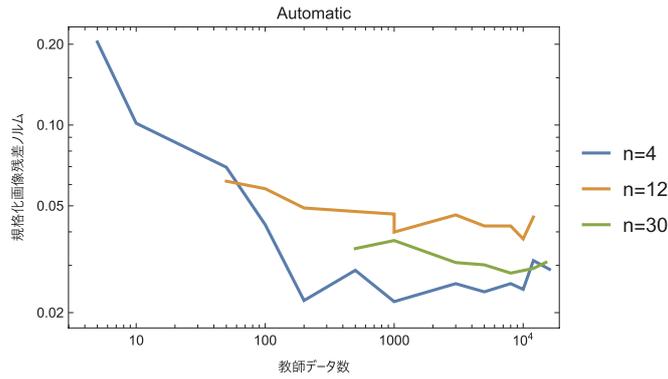


図 11: 規格化画像残差ノルムの教師データ数依存性の分割数による変化 ($n = 4, 12, 30$)

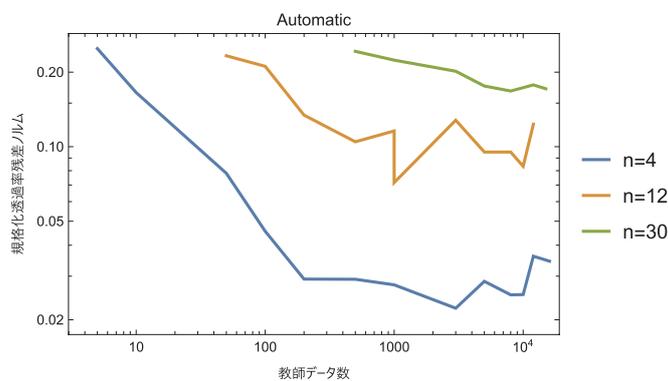


図 12: 規格化透過率残差ノルムの教師データ数依存性の分割数による変化 ($n = 4, 12, 30$)

ります。これを用いた notebook の例を別紙 3 に示します。

分割数 $n = 30$ の場合の規格化画像残差ノルムおよび規格化透過率残差ノルムの教師データ数依存性を Predict 関数を用いた結果（つまり，図 9，10）とともに図 13，14 に示します。

これから，NetChain，NetTrain を用いても，規格化画像残差ノルム，規格化透過率残差ノルムともに Predict を用いた場合と同様の収束をしていることが分かります。しかし，どちらも Predict 関数で GradientBoostedTrees を用いる方がより元画像に近い画像を再現できていると推定できます。

また，NetChain，NetTrain を用いた場合の規格化画像残差ノルムの教師データ数依存性は分割数に対して図 15 のように変化します。

教師データ数が同じ場合には，分割数 n が多いと規格化画像残差ノルムの値は大きくなる傾向があるが，Predict を用いた場合同様， n の順になっているわけではありません。また， $n = 100$ の場合には 2 通りの計算を行っているが，規格化画像残差ノルムの教師データ数依存性は大きく異なります。これは，教師データのサンプル依存性が非常に大きいことを意味しており，統計的性質を調べる必要があります。

NetChain，NetTrain を用いて，様々な教師データ数で推定された画像の特徴量空間プロットは図 16 のようになります。

元画像（図中「正解」）とモンテカルロ法を用いた通常の最適化法による結果とともにプロットし

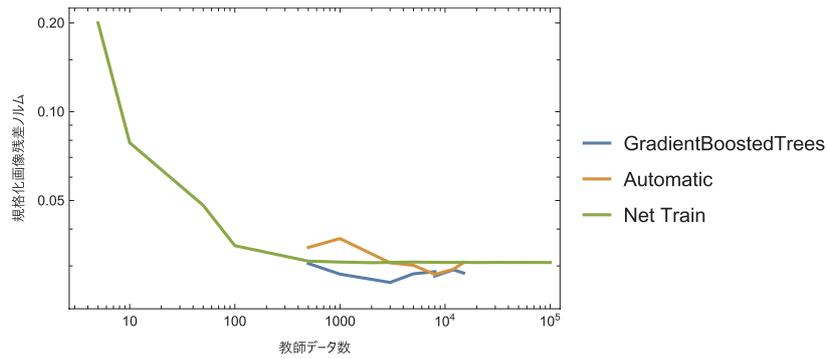


図 13: 規格化画像残差ノルムの教師データ数依存性 ($n = 30$)

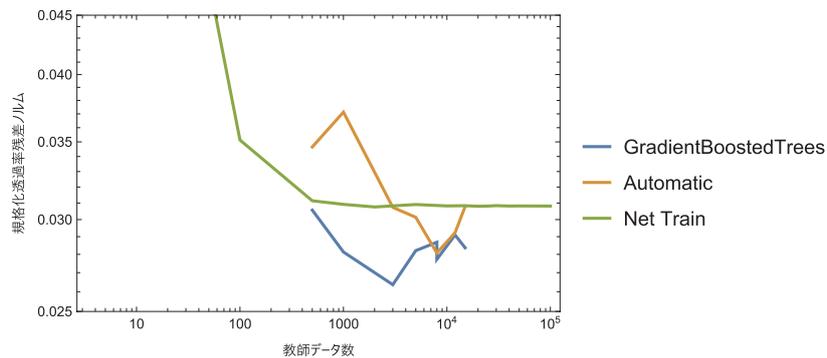


図 14: 規格化透過率残差ノルムの教師データ数依存性 ($n = 30$)

であるが、似ても似つかない画像を推定していることが分かります。しかしながら、NetChain のネットワーク構成はかなり自由度が高いのでそれらを工夫する余地があります。

ここで示した結果は、あるサンプル教師データを用いた場合の結果であって、サンプル依存性がかなり大きいと言えます。本来、同じ条件（分割数、教師データ数）について多くのサンプル教師データセットを用意して、その統計性を確認する必要があります。機械学習で得られる結果の特性をよく理解した上で利用しましょう。

講評

透過率のデータは Mathematica notebook で提供していて、それぞれの分割数ごとにリスト形式で与えてあるので、セルを選んでコピーすればいいだけだったのですが、なんだか、ファイルを読み込もうとしていた例が非常に多かったですね。ファイルの中身を見て、何が最も簡単かをよく考えましょう。プログラミングの技法を競っているわけではありません。如何に手早く、良い結果を得るかの問題なので。

notebook フロントエンドはあーでもない、こーでもない、これならうまくいくか？と試行錯誤するのに最適な入力フロントエンドなのでその機能を有効に利用しましょう。

さて、最も簡単には、昨年同様、Predict 関数を用いれば画像を得られるという解法だったのですが、画像を（正解でないにしても）再現できた解答は 2 例しかありませんでした。分割数 n が大きく

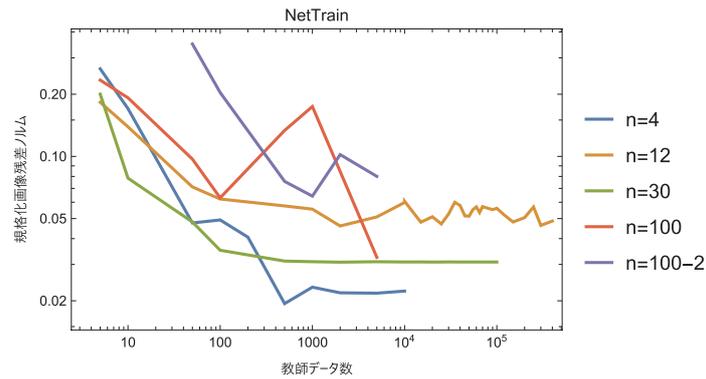


図 15: 規格化画像残差ノルムの教師データ数依存性の分割数による変化 ($n = 4, 12, 30, 100$)

なると計算時間がかかるので、 n が大きな場合については時間的に無理かとは思っていましたが、惨憺たる状況で驚きました。昨年の様子から今年ももっと頑張ってくれるだろうと期待していたので、残念でした。

教師データを Mathematica 上で作成し、それを学習させ、与えられたサンプルデータから画像を推定させなければならぬので、教師データを作成する段階で挫折したのかもしれませんが。

表彰された答案でも、学習データの生成法の記述、どのような方法で推定したかの方法の記述が極めて不十分なものでした。今後、報告書の書き方の訓練を積んでもらいたいと思います。

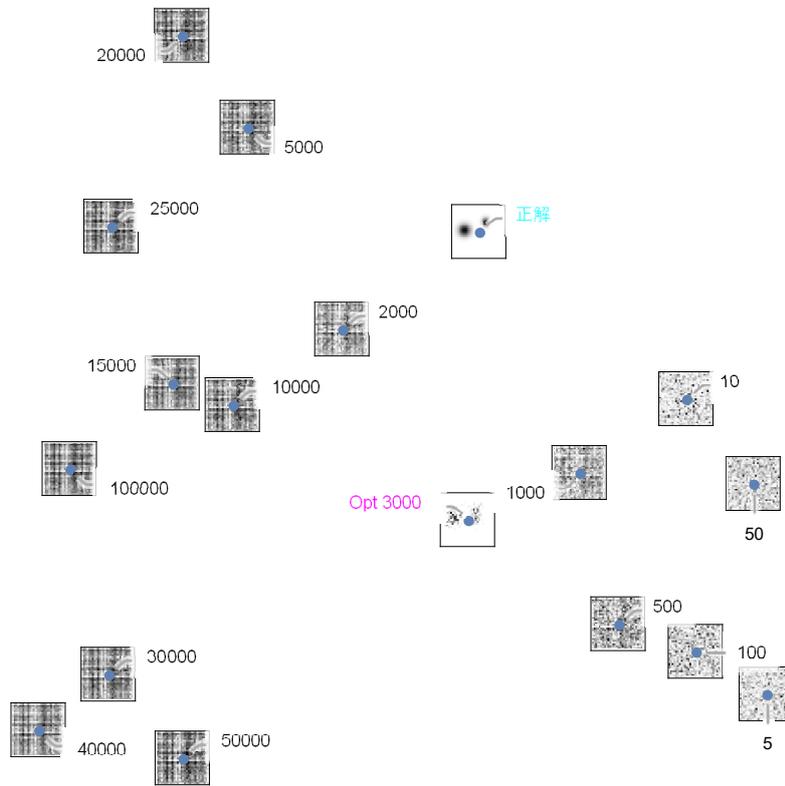
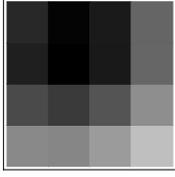


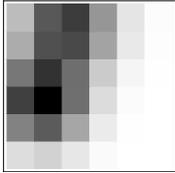
図 16: $n = 30$ の場合に様々な教師データ数について NetChain, NetTrain を用いて推定された画像の特微量空間プロット.

元画像

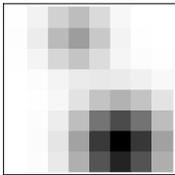
n = 4



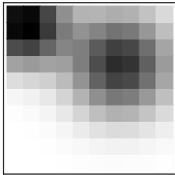
n = 6



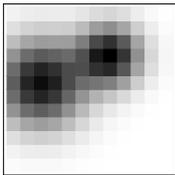
n = 8



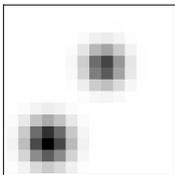
n = 10



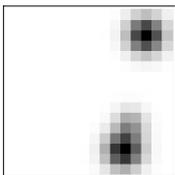
n = 12



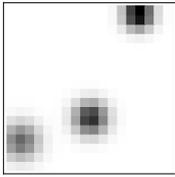
n = 14



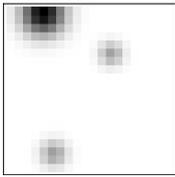
n = 16



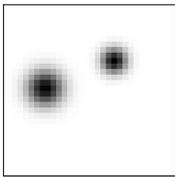
n = 18



n = 20



n = 30



```

In[*]:= n = 4;
        nsmp = 16000;

In[*]:= testm = Table[
        0.2 (Exp[-0.2 ((i - n/3)^2 + (j - 2n/3)^2)] + Exp[-0.1 ((i - n/2)^2 + (j - n/4)^2)]),
        {i, 1, n}, {j, 1, n}];
testsol = Flatten[Table[{Product[(1 - testm[[i, j]]), {j, 1, n}],
        Product[(1 - Transpose[testm][[i, j]]), {j, 1, n}], {i, 1, n}]]

Out[*]:=
{0.252071, 0.310152, 0.245916, 0.26138, 0.357147, 0.312335, 0.550396, 0.481244}

```

```

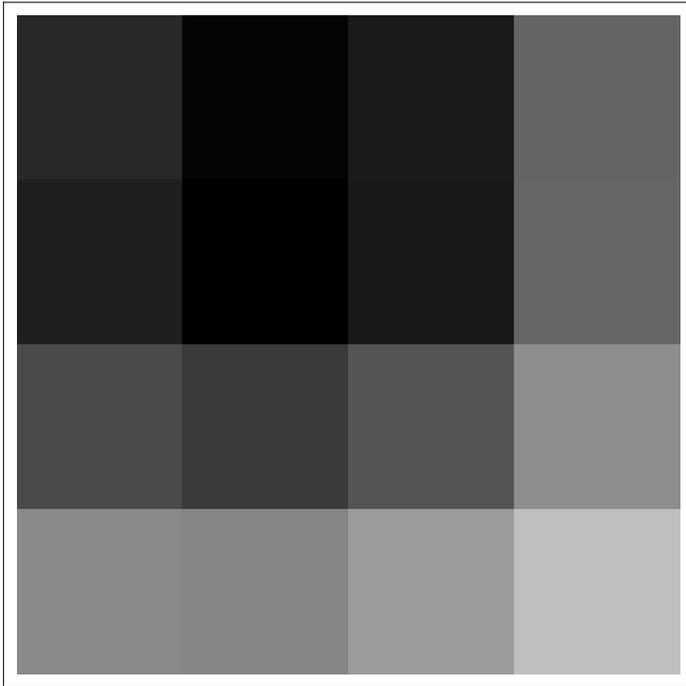
In[*]:= ArrayPlot[testm]

```

```

Out[*]:=

```



```

In[*]:= Do[kt[i] = {}, {i, 1, n n}]

In[*]:= Do[abdis =
        Table[0.2 (Exp[-RandomReal[{0.1, 0.5}] ((i - n Random[])^2 + (j - n Random[])^2)] +
        Exp[-RandomReal[{0.1, 0.5}]
        ((i - n Random[])^2 + (j - n Random[])^2)]), {i, 1, n}, {j, 1, n}];;
sol = Flatten[Table[{Product[(1 - abdis[[i, j]]), {j, 1, n}],
        Product[(1 - Transpose[abdis][[i, j]]), {j, 1, n}], {i, 1, n}]];
Do[kt[(i - 1) n + j] = AppendTo[kt[(i - 1) n + j], sol -> abdis[[i, j]],
        {i, 1, n}, {j, 1, n}], {1, 1, nsmp}]

In[*]:= p = Table[Predict[kt[(i - 1) n + j]], {i, 1, n}, {j, 1, n}]

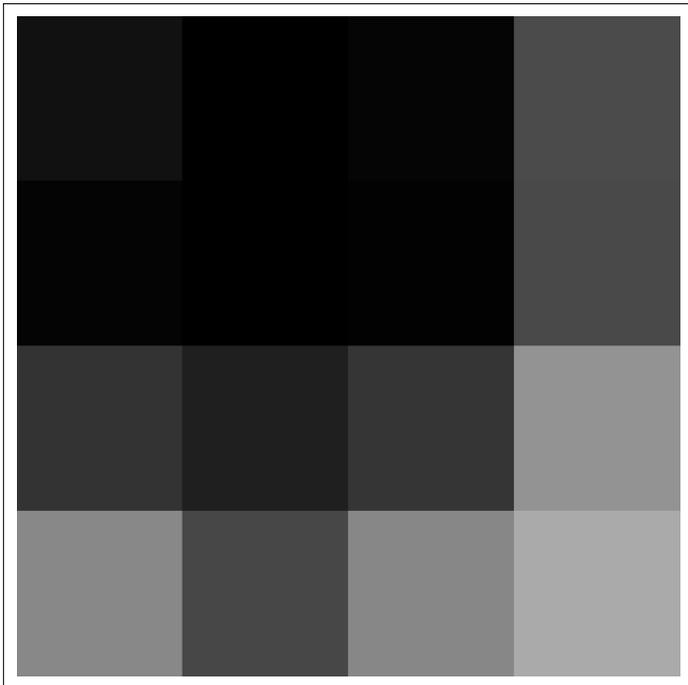
In[*]:= solm = Table[p[[i, j]][testsol], {i, 1, n}, {j, 1, n}]

Out[*]:=
{{0.268942, 0.288139, 0.282142, 0.203517}, {0.283186, 0.288464, 0.28542, 0.205924},
        {0.23025, 0.25321, 0.227663, 0.122801}, {0.13507, 0.207538, 0.135819, 0.0961372}}

```

```
In[*]:= ArrayPlot[solm]
```

```
Out[*]=
```



```
In[*]:= sol = Flatten[Table[{Product[(1 - solm[[i, j]]), {j, 1, n}],
    Product[(1 - Transpose[solm] [[i, j]]), {j, 1, n}]}, {i, 1, n}]]
```

```
Out[*]=
```

```
{0.297551, 0.348891, 0.289412, 0.299757, 0.389452, 0.342374, 0.535386, 0.501463}
```

```
In[*]:= (sol - testsol).(sol - testsol)
```

```
Out[*]=
```

```
0.00951377
```

```
In[*]:= Flatten[(solm - testm)].Flatten[(solm - testm)]
```

```
Out[*]=
```

```
0.0134575
```

```

In[ ]:= n = 30;
        nsmp = 100000;

In[ ]:= testm = Table[
        0.2 (Exp[-0.2 ((i - n/3)^2 + (j - 2n/3)^2)] + Exp[-0.1 ((i - n/2)^2 + (j - n/4)^2))],
        {i, 1, n}, {j, 1, n}];
testsol = Flatten[Table[{Product[(1 - testm[[i, j]]), {j, 1, n}],
        Product[(1 - Transpose[testm][[i, j]]), {j, 1, n}], {i, 1, n}]]

Out[ ]:=
{1., 0.983722, 0.999998, 0.946844, 0.999955, 0.861254, 0.999402, 0.714359, 0.99462,
0.535711, 0.967786, 0.386514, 0.874199, 0.307922, 0.686316, 0.307922, 0.486281,
0.386514, 0.387427, 0.535711, 0.398617, 0.714359, 0.432926, 0.861252, 0.39777,
0.946802, 0.32691, 0.98314, 0.296813, 0.990656, 0.337477, 0.967324, 0.454145,
0.875709, 0.625554, 0.692055, 0.794971, 0.501423, 0.911665, 0.424967, 0.969801,
0.501424, 0.991689, 0.692067, 0.99814, 0.875827, 0.99966, 0.968115, 0.999949,
0.994671, 0.999994, 0.999408, 0.999999, 0.999956, 1., 0.999998, 1., 1., 1., 1.}

```

```

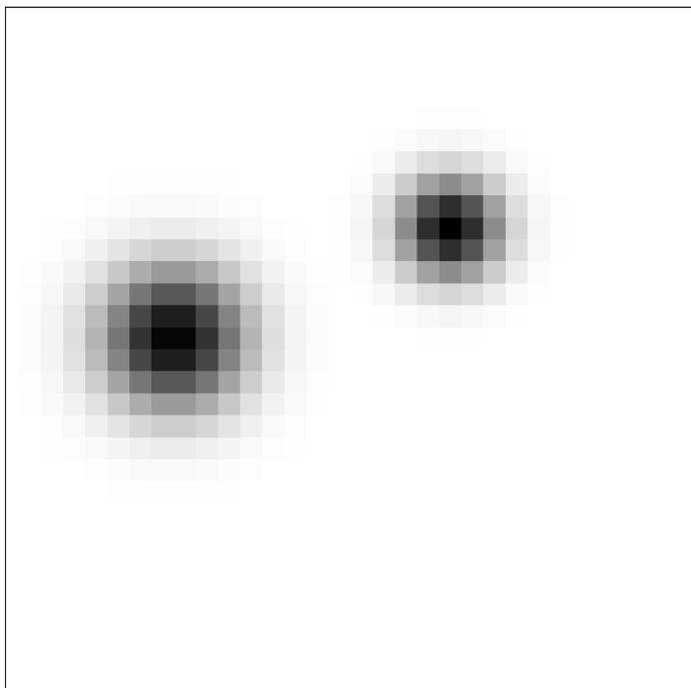
In[ ]:= ArrayPlot[testm]

```

```

Out[ ]:=

```



```

In[ ]:= kt = {};

```

```

In[ ]:= Do[abdis =
        Table[0.2 (Exp[-RandomReal[{0.1, 0.5}] ((i - n Random[])^2 + (j - n Random[])^2)] +
        Exp[-RandomReal[{0.1, 0.5}]
        ((i - n Random[])^2 + (j - n Random[])^2)]), {i, 1, n}, {j, 1, n}];;
sol = Flatten[Table[{Product[(1 - abdis[[i, j]]), {j, 1, n}],
        Product[(1 - Transpose[abdis][[i, j]]), {j, 1, n}], {i, 1, n}]];
kt = AppendTo[kt, sol -> Flatten[abdis]], {1, 1, nsmp}]

```

```
In[*]:= net = NetChain[{LinearLayer[10], LinearLayer[20], LinearLayer[n n]}]
```

```
Out[*]=
```



```
In[*]:= trained = NetTrain[net, kt]
```

```
Out[*]=
```

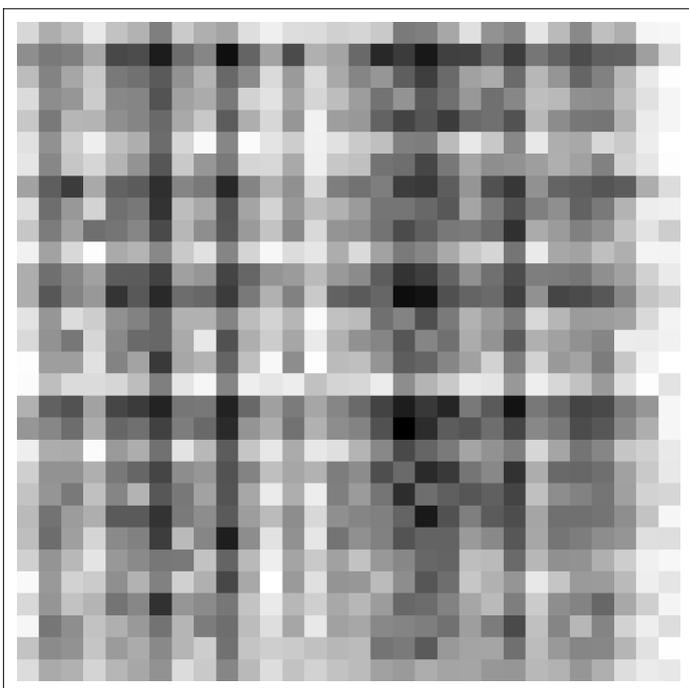


```
In[*]:= solv = trained[testsol];
```

```
In[*]:= solm = Partition[solv, n]
```

```
In[*]:= ArrayPlot[solm]
```

```
Out[*]=
```



```
In[*]:= Flatten[(solm - testm)].Flatten[(solm - testm)]
```

```
Out[*]=
```

```
0.85459
```