AIが学習に失敗する例を用いた教材の提案

中田 豊久[1,a)]　三浦 元喜[2,b)]

概要：従来の人工知能 (AI) を学ぶための教材は，チュートリアルのように AI をブラックボックス化して，まず AI は何をできるのかを理解するといったものが多い．この場合の課題は，例えば既存の AI を拡張して新たな問題に適用させようとし，しかし上手くいかない場合に，その対処の方法が分からないことである．その原因は，AI の中の仕組みをやはり十分に理解していないことであると考える．そこで本研究では，従来の教材に加え，主に仕組みを理解することを主眼とした教材について提案する．その基本的な考え方は，(1) あえて答えの分かっているシンプルな問題を AI に学習させる，また (2)AI が学習に失敗する例を用い，AI に学ばせるために人間が試行錯誤を行うことにより AI の仕組みを理解する，である．本教材の対象者は，チュートリアルのような AI が何ができるのかを概ね理解していて，かつ，ある程度のプログラミング技術をすでに修得した大学生 3，4 年生程度を想定している．

キーワード：人工知能, 学習教材, 失敗例

Proposal of teaching materials using examples of AI learning failures

1. はじめに

内閣府の AI 戦略 2019[1] では，文理を問わずにすべての大学，高専生がリテラシーレベルの数理・データサイエンス・AI 教育を修得するという目標が掲げられている．また高等学校で新たに必履修化された情報科目の教科書では，例えば「情報 II」教員研修用教材 第 3 章 情報とデータサイエンス [2] では，機械学習モデルとしてニューラルネットワークが取り上げらえている．そして文部科学省 高等学校情報科に関する特設ページ 授業・研修用コンテンツ [3] でも AI に関する教材が提示されていて，音声認識による機械とのインターフェースをプログラミングでカスタマイズする例などが示されている．

これらの AI を学ぶための教材は，文理を問わずにすべての学生が学ぶことを想定しているため，(1) 数式を多用しない，(2) プログラミング言語を多用しない，(3) サンプルを動かしてまずは AI の挙動を見る，(4) 入出力データを変えることによって AI をカスタマイズする，といった視点が用いられていると考える．しかし一方，AI の仕組みについてはその構造の複雑さから教科書等では深く立ち入っていないのが現状である．AI の仕組みを理解できないことの問題点は，例えば現時点の AI では何ができないのかといった限界を想像することが困難なことが挙げられる．そこで本論文では，これまでの教材開発の視点を踏襲しつつ新たに AI の仕組みについて理解を深める手法について提案する．具体的には次の 2 点を主に重視する．

図 1: 学生は，この Google Colaboratory のような開発環境で与えらえたサンプルプログラムをカスタマイズしながら動作させ，AI の挙動や仕組みを学んでいく．

1 新潟国際情報大学
2 千葉工業大学
a) nakada@nuis.ac.jp
b) motoki.miura@p.chibakoudai.jp

©2023 Information Processing Society of Japan

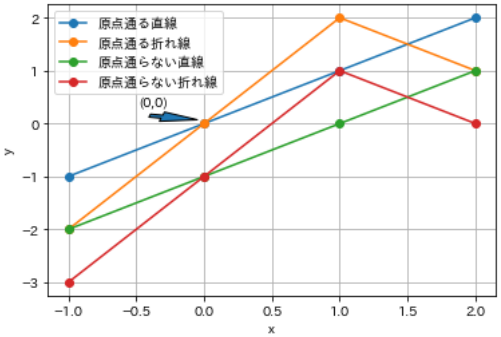

図 2: ニューラルネットワークの計算方法を学ぶためのデータ例

	原点通る直線	原点通る折れ線	原点通らない直線	原点通らない折れ線
neuron=1,バイアスなし	0.0022	0.17	0.76	1.4
neuron=1,バイアスあり	0.0033	0.17	0.0023	1.1
neuron=2,バイアスなし	0.0014	8.3e-10	0.76	0.92
neuron=2,バイアスあり	4.2e-05	1.9e-09	8.7e-12	2.8e-08

図 3: ニューラルネットワークのモデルごとの学習結果: それぞれの数値は平均二乗誤差を示し，青は学習成功, 赤は失敗を表している．

(1) **答えの分かっているシンプルな問題を AI に学習させる．**

(2) **AI が学習に失敗する例を用いて，人間が AI に学習を成功させることを試行錯誤することにより，AI の仕組みを学ぶ．**

従来は，例えば畳み込みニューラルネットワークではその起源であり効果が最も現れやすい画像認識を題材とすることが多い．AI の挙動を理解する上ではこのような題材は効果的であると考えられるが，構造が複雑になりすぎて機械学習モデルの個々の機能が画像認識にどのように影響しているのかなどの仕組みを把握することは困難である．そこで仕組みの理解を促進するためにあえてシンプルな問題を AI に学習させ，また機械学習モデルの機能を操作して AI が学習に失敗する例をも用いて AI の機能や仕組みを学ぶものである．

本教材の対象者は，チュートリアルのような AI が何ができるのかを概ね理解していて，かつ，ある程度のプログラミング技術をすでに修得した大学生 3，4 年生程度を想定している．またすべての教材例について学生は，図 1 の Google Colaboratory のような開発環境で講師から提供されたプログラムをカスタマイズしながら動かし，そして講師のからの説明を聞いて学ぶことを想定している．また機械学習のライブラリとしては，本教材では tensorflow ライブラリを使用している．

2. ニューラルネットワークの計算方法を学ぶ

近年の AI は，ニューラルネットワーク [4] をベースとした機械学習が非常に流行っている．そこでまずはニューラルネットワークについて何を計算しているのかを理解するためにあえてシンプルな回帰問題を提案する．

2.1 問題設定

図 2 には 4 種類のデータを示している．ここでの問題は，ニューラルネットワークの入力を横軸の x としたとき，図 2 と同じ縦軸 y の数値を出力するニューラルネットワークを作成することである．ただしニューラルネットワークの構造は 1 層でできていて，ニューロン数が 1 のもの，2 のもの，そしてそれぞれにバイアスがあるもの，ないものの 4 種類とする．

2.2 学生が学ぶこと

ニューラルネットワークは，データからモデルを作成する学習過程と，学習済みモデルで実際に計算を実施する推論過程とがある．ここではこのうち推論過程についてニューラルネットワークがどのような仕組みで計算しているのかを学生が理解することを目的とする．具体的には，ニューラルネットワークのニューロンと呼ばれる個々の素子は，活性化関数と呼ばれる関数の一部を切り出して拡大，縮小するような処理を行っていて，ニューロンが複数になる場合は，それぞれのニューロンが切り出した関数を足し合わせて 1 つの関数を作るようなことを行う．またバイアスは，この切り出した一部を左右に移動する機能を持つ．これらのことを学生が理解することを目的とする．

2.3 学生の学習手順 1: 挙動の理解

まず学生は，図 2 として与えられたデータからニューラルネットワークをベースとした機械学習モデルをニューロン数を 1 つ，2 つとしたものと，それぞれにバイアスあり，なしとしたものの 4 種類を作成する．次にそれぞれのモデルにデータを学習させて損失を求める．損失とは，AI に学んでほしい正解データから実際に AI が出力したデータとの差分である．その結果を図にしてまとめると，図 3 のようなものを学生は自分自身で作成することになる．図 3 は 4 種類のモデルと 4 種類のデータで構成されるもので，それぞれのセルには損失の数値が記載されている．またここではデータを再現できた AI については青，できなかったところは赤で示されている．できた／できないの具体的な判定基準は，データの 5%の誤差を許容するとし，その際の平均二乗誤差である 0.01 未満を AI が学習できた，としている．

ここで学生に学んで欲しいことは，次の 2 点である．

- 折れ線のデータを再現するには，ニューロンが少なく

©2023 Information Processing Society of Japan

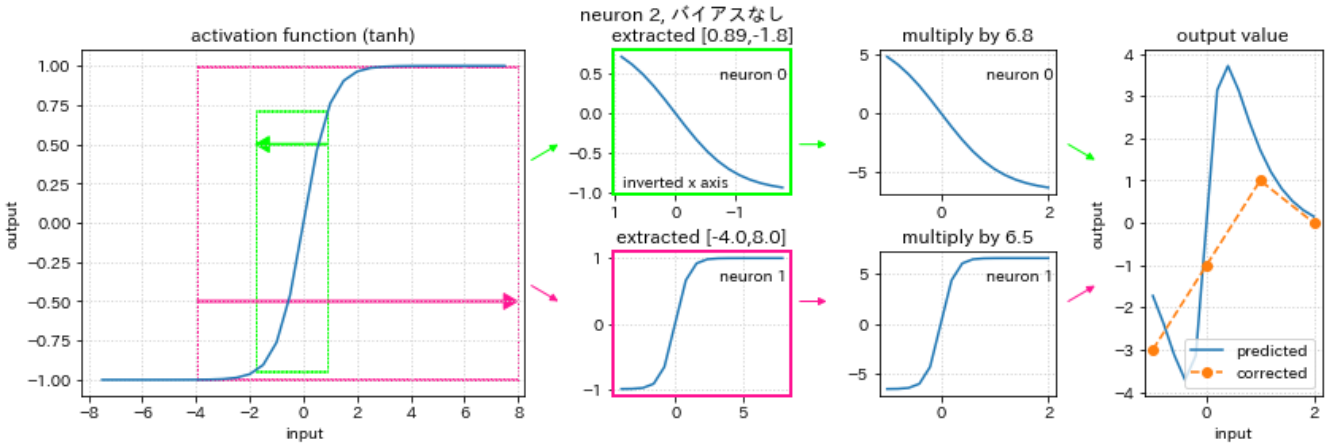

図 4: ニューラルネットワークの計算フロー図 (ニューロン数 2, バイアスなしの場合, データは原点を通らない折れ線グラフ): 4 列になっているグラフの左から 1 列目は活性化関数 (tanh)，2 列目はニューロンによって 1 列目から切り出したもので 1 列目の緑とピンクの枠に上下の図が対応している．ニューロン数が 2 であるため 2 つの枠が切り出されている．3 列目は 2 列目に数値をかけて拡大縮小反転したもの，そして 4 列目は 3 列目の 2 つのグラフを足し合わせた青線のグラフと教師データのオレンジのグラフである．このオレンジの，点線ではなく丸の 4 点をニューラルネットワークの出力である青線が通過することが求められている．

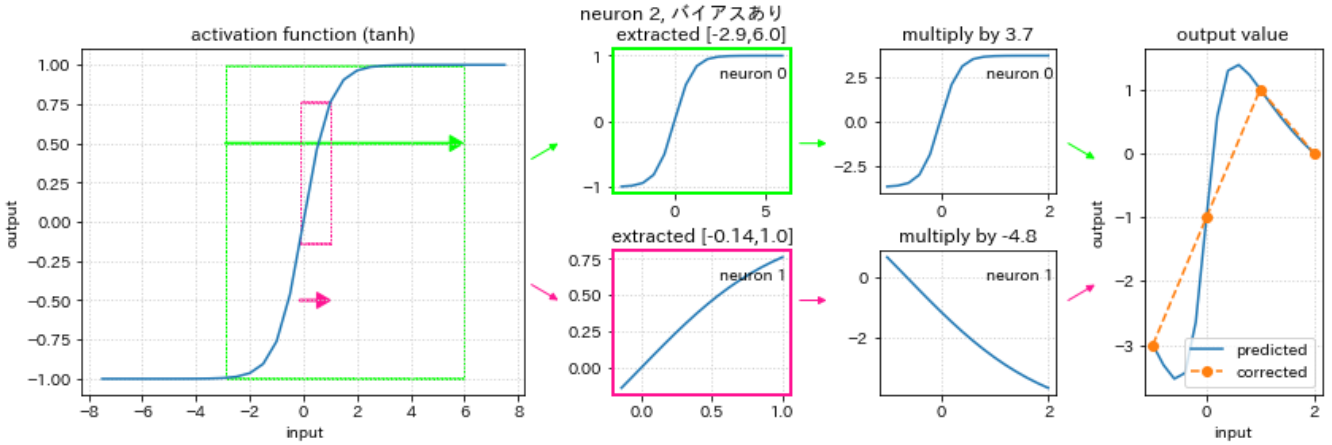

図 5: ニューラルネットワークの計算フロー図 (図 4 との違いは，モデルにバイアスがある場合の例)

とも 2 つは必要になる．

- 原点を通らないデータを再現するには，バイアスが必要になる．

図 3 において損失が大きく AI が学習できなかったとされる赤いセルは，2 列目の折れ線の場合はニューロン数 1 のときに AI は学習できていない．また同図の 3 列目によると原点を通らないデータではバイアスがなしだと AI は学習できない．これらを統合して，4 列目の原点を通らず，かつ，折れ線のデータの場合はニューロン数 2，バイアスありのものしか学習できていないことが明らかである．この理由については次節以降で学生はさらに深く学んでいくことになる．

2.4 学生の学習手順 2: 仕組みの理解

学生は与えられたプログラムを動作させて，図 4 のようなニューラルネットワークの計算フロー図を作成する．計算フロー図とは，本研究で名付けた名前であり，ニューラルネットワークがどのように入力データから出力データを作り出しているのかを図示したものである．

ここで前節で示したニューロン数とバイアスに関するニューラルネットワークの挙動は，次のように学生に理解される．

2.4.1 ニューロン数の理解

ニューロン数は，活性化関数から切り出せるグラフの数になる．例えばニューロンが 1 つの場合は，図 4 の 2,3 列目が上下 2 つではなく 1 つになり，拡大縮小した後の切り出したグラフはそのままニューラルネットワークの出力になる．ここで，図 4 の左側の活性化関数には数値が上がって下がるのような山やその逆の谷の部分が無いことに注目すると，ニューロンによって切り出す部分がどこであって

©2023 Information Processing Society of Japan

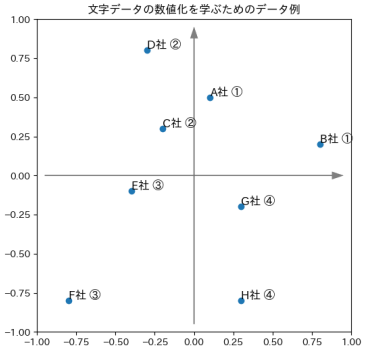

図 6: 文字データの数値化を学ぶためのデータ: ポジションマップの例: 横軸，縦軸は何かしらの基準で作成され，例えば企業分析などで各社の特徴を可視化するときに使用される．

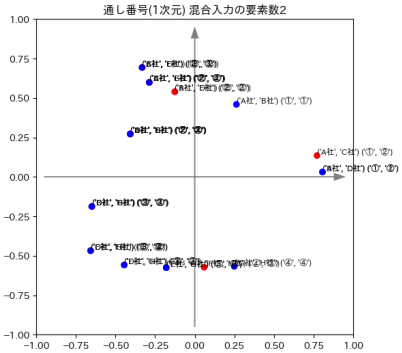

図 7: 文字を 1 次元の数値に変換した AI の学習結果例

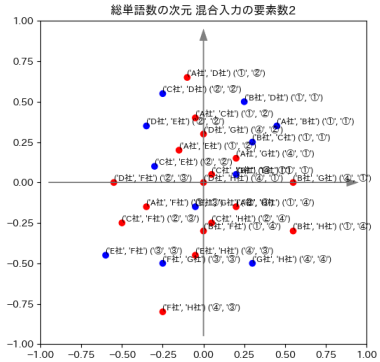

図 8: 文字を総単語数の次元に変換した AI の学習結果例

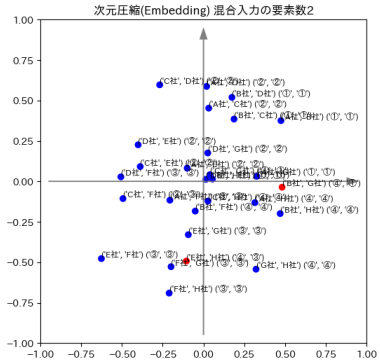

図 9: 文字を Embedding によって 2 次元に変換した AI の学習結果例

もニューロンが 1 つしかない場合はグラフの山や谷を表すことができないことがわかる．よって教師データが折れ線場合には，1 つのニューロンだけではその山になる部分を再現することができない．

一方ニューロン数が 2 の場合，図 4 の 4 列目は 3 列目の 2 つのグラフを足し合わせて作られているため，グラフの山や谷を作り出すことができる．よって今回の教師データでは，折れ線のデータの場合にはニューロン数 2 のモデルでなければ学習できないということを，学生は理解する．

2.4.2 バイアスの理解

バイアスは，活性化関数から切り出したグラフを左右に移動するという機能を持つ．例えば図 4 の 3 列目は，上下のどちらのグラフも x=0,y=0 の原点を通る．一方バイアスのある図 5 では，原点を通らないグラフも作れている．よって図 4 の 4 列目のように，ニューラルネットワークの出力が原点を通らないグラフを作り出すことができている．

今回の教師データでは，単純な直線であっても原点を通らないグラフの場合には，バイアスが無いと AI の学習に失敗してしまうことが，学生は理解できると考える．

3. ポジショニングマップを用いて文字データの数値化を学ぶ

ポジショニングマップとは図 6 のように，2 つの軸で評価された各項目を 2 次元マップ上に配置するものである．マーケティングなどの分野で用いられることがある．

3.1 問題設定

図 6 の社名からマップ上の位置と所属する第 1〜4 象限までの領域を学習する AI を作成し，A 社と B 社を足して 2 で割るのような入力を混合した例にどれだけ対応できるかを調べる．

3.2 学生が学ぶこと

文字データは，(“A”+“B”)/2 のような計算をすることはできない．しかし近年に開発された Embedding と呼ばれる AI の手法を使用することにより，ある程度は文字を数値のように計算するようなことが実現することができる．ここでは文字を数値化するときに，単純に通し番号のように順番に文字に番号をつけていく方法，one-hot と呼ばれる単語数の次元を持つベクトルを作り，該当する箇所の要素だけを 1 にする方法，そして Embedding[5] と呼ばれる数値化する部分も教師データから AI が学習する方法の 3

©2023 Information Processing Society of Japan

種類を試し，ポジショニングマップの領域を推定するような問題では Embedding の手法が良い性能を示すことを学生が理解することを目的とする．

3.3 学生の学習手順

講師から渡されるサンプルプログラムを参考に，社名から数値に変換する関数を学生は作成する．その関数を使用した入力データの変換を持つ機械学習モデルによって，図 6 のマップを学習する．ここで言うマップの学習とは，例えば “A 社” が入力だった場合に，x=?, y=?, 領域=第?象限といった出力をする機械学習モデルを作成することである．

学生が作成する関数は 3 種類で，文字に順番に数値を割り当てる通し番号のような変換，one-hot と呼ばれる総単語数の次元ベクトルの該当する要素だけを 1 にしたベクトルへの変換，そして Embedding と呼ばれる数値化の部分も教師データから学習するものである．この例では Embedding は tensorflow ライブラリの該当するモジュールを使用している．

機械学習モデルのトレーニングが終わると，学生はそれぞれの関数を使用して “A 社+B 社” の数値を作成し学習済みモデルの入力として渡す．このようにして A 社から H 社までの 28 通りの組み合わせについて AI に計算させた出力と教師データから得られる正解データとを図示する．図は図 7〜図 9 のようになる．それぞれの図の赤い点は，AI が領域を間違えているものである．図 7 の通し番号による数値化の結果では，複数の点が重なっている．

この結果から学生は，図 7 および図 8 の結果から単純に通し番号でつけた番号や多次元ベクトルでは四則演算することはできないことを学び，図 9 から Embedding によって社名のマップ上の位置であり，ある意味では社の特徴を AI が学習できることを理解する．

4. 五目並べの勝利判定を用いて CNN の計算方法を学ぶ

2012 年の画像認識の精度を争う大会である ILSVRC で二位のチームに大差をつけて優勝した畳み込みニューラルネットワーク (CNN) を用いた画像認識 [6] が CNN のブームを起こしたといわれている．CNN はフィルタのような要素をデータにスライドしながら適用し，それぞれの位置から得られた値で構成された特徴マップを作成する．そしてその特徴マップから認識等を行う．

4.1 問題設定

本論文では，あえて CNN の起源である画像認識ではなく五目並べの勝利判定に CNN を使うことを提案する．その理由は，画像を認識するというタスクよりもよりシンプルな例を用いた方が CNN の仕組みを理解できると考えたからである．ここで五目並べの勝利判定とは，盤面に駒が縦横斜めのどこかに連続して 5 個並んでいるとその駒のプレイヤが勝ちとするものとする．ただし本論文では実装の都合上，連続する駒は 5 個ではなく 4 個としている．

4.2 学生が学ぶこと

CNN のフィルターが盤面を走査し，個々の位置から得られる値で構成された特徴マップを作るという CNN の仕組みを学生は理解する．この問題設定の場合，フィルターは縦横斜めという連続する駒の形を自動的に AI はデータから学ぶ．学生は，五目並べのルールを AI に教えていなくても，多くの盤面とその盤面におけるプレイヤの勝利状態というデータから結果的に五目並べの勝利判定ルールを AI が導き出すことを体験する．また CNN のモデルをカスタマイズすることによって，勝利判定を学習できない AI を作成することによって CNN の仕組みを理解する．

4.3 学生の学習手順 1: 挙動の理解

講師より与えられたサンプルプログラムを実行し，ランダムに生成された盤面情報と，その盤面における勝利状態を作成し，AI に対するトレーニングデータを学生は作成する．そして予め用意された図 10 の機械学習モデルにデータを与えて AI を学習させる．学習の結果は，損失を学生が確認するだけでなく，新しくランダムに生成された盤面情報を与えて正しく勝利判定ができているかどうかを学生自身による勝利判定と見比べて確認する．

4.4 学生の学習手順 2: 仕組みの理解

サンプルプログラムの AI は学習に成功するので，そのモデルが作ったフィルターを可視化して学生は確認する．その可視化例を図 11 に示す．黄色でハイライトされている部分は，フィルタを通って情報が利用される部分を示している．ここで分かることは，縦や横，そして斜めのラインが形成されているということは，そこに駒がある場合にフィルタを通過する量が多くなるということである．つまり AI は結果的に，連続した駒があると勝利する，という知識を獲得しているともいえる．

4.4.1 フィルタ数が少なくて AI が失敗する例

学生は CNN のフィルタ数を 12 だったものから 1 に変更して AI の学習を再実行する．図 11 から明らかなようにフィルタは縦横斜めの駒を数える機能を有することになる．この数を数えるという機能は，その仕組みから 1 つのフィルタで縦横斜めを同時にカウントすることはできない．よってフィルタ数を 1 に変更した AI は学習に失敗する．ここから学生には，最低でいくつのフィルタが必要なのかを検討してもらう．ここで CNN のパラメータの 1 つである padding というものがその答えに関係してくる．padding とはフィルタを適用するときに盤面の枠を広げて適用する

©2023 Information Processing Society of Japan

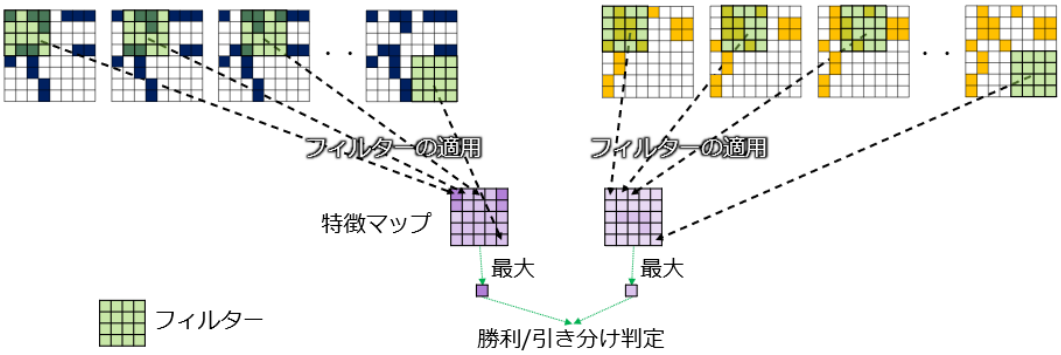

図 10: 五目並べの勝利判定をする CNN を用いた機械学習モデルの構造

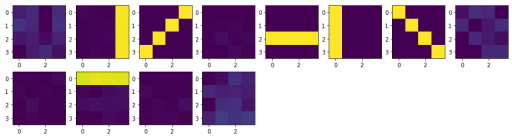

図 11: CNN のフィルターの可視化

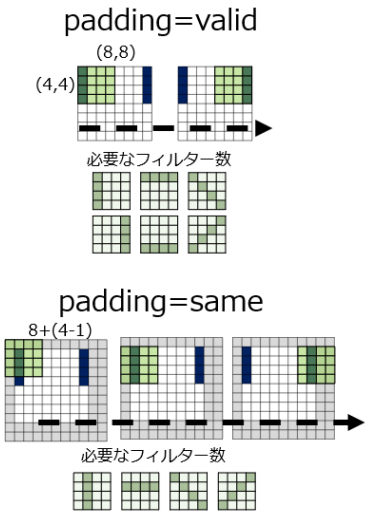

図 12: CNN のパディングと最小フィルタ数

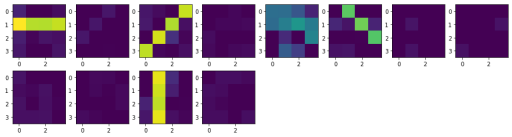

図 13: バイアスがある場合の失敗例

かどうかのパラメータである．padding を valid(枠を広げない) と，same(枠を広げる) にしたときのそれぞれのフィルタの適用の様子を図 12 に示す．この図から盤面の端にある連続した駒をカウントするには padding の設定によって影響を受けることが明白であり，学生は最小のフィルター数と，padding という CNN の仕組みについて理解を得ることができる．

4.4.2 バイアスや重みに制限を与えずに AI が失敗する例

最初に学生に渡す CNN のモデルでは，バイアスなし，重みには負の数を使用しない，という制限がかかっている．これを外すと AI は勝利判定を学習できないことが多い．このことを学生自身が CNN のモデルをカスタマイズして確認する．例えばバイアスありとして AI を学習させたときの結果は，図 13 のようになる．ここには右下がりのフィルタが現れていなく，つまり学習できていないことが分かる．また重みに負の数を許容しても同様の結果を得る．この AI が学習に失敗する理由は，1 つはフィルタによって駒の引き算ができてしまうことにある．図 10 に示す CNN のモデルは，2 つの特徴マップの最大値を比較してどちらが勝利しているのかを判定する．ここで想定していることは，連続する駒があるほど特徴マップの値が大きくなることである．大きな値を持つ特徴マップのプレイヤが勝利という判定をモデルの最後で行っている．しかしフィルタが負の数を出力し特徴マップの最大値が例えば負の数になると，相対的に対戦プレイヤの特徴マップの値が大きな数値と評価されて実際には勝利していないプレイヤを勝利していると判定してしまうことがある．

このような AI が学習に失敗する現象は，トレーニングデータではあまり起こらずテストデータで発生する．学生は，AI の学習はトレーニングデータの損失を確認すると成功しているように見えるのだが，そのあとで新しい盤面情報を作成して AI をテストするときに上手くいかないことを経験する．そしてそこから過学習であるとか，CNN の仕組みについて理解していく．

4.4.3 プレイヤごとの盤面を分けずに AI が失敗する例

画像に CNN を適用する場合，画像の RGB3 要素のそれぞれにフィルタを適用して 1 つの特徴マップを作成する．その様子を図 14 の左側に示す．学生には五目並べの盤面を 2 要素の画像データとして取り扱って AI を学習させることを実行させる．しかしこのような適用を五目並べの勝利判定に行うと AI の学習は失敗する．その理由は図 14 に示す通りで，左側の 2 要素とした場合には特徴マップが 1 つになってしまっているためである．これは例えばプレイヤ 1 の連続する駒を数えているときにプレイヤ 2 の駒も計算に入れているようなものである．そこで図 14 の右側のように盤面を分離してそれぞれカウントするような処理が

©2023 Information Processing Society of Japan

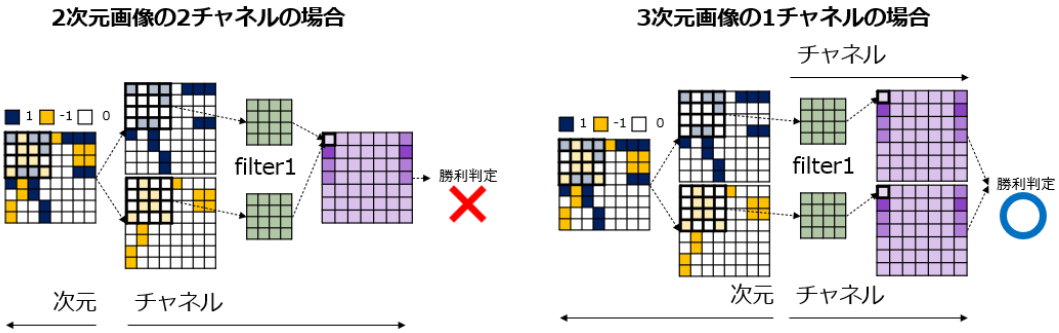

図 14: 次元とチャネルの設計による失敗例

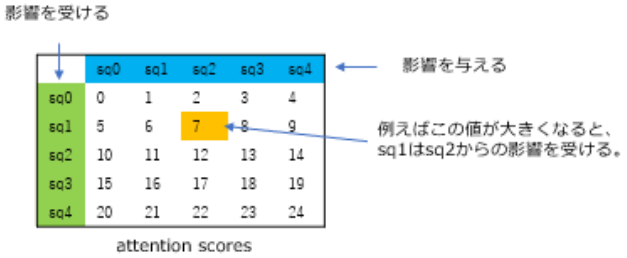

図 15: Attention score の見方

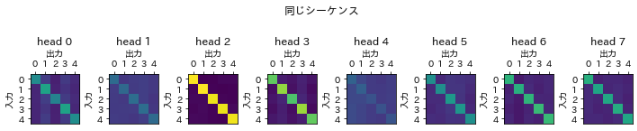

図 16: シーケンスごとの合計を求める AI の attention scores

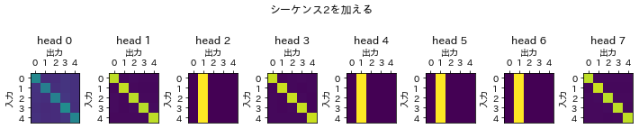

図 17: シーケンス 1 が他のシーケンスに加算される AI の attention scores

必要になることを，学生は理解する．

5. 入力データの関係性を見つける Attention 機構について学ぶ

Attention[7] とは，ChatGPT, GPT-4[8] のベースになっている Transformer と呼ばれる機械学習モデルの 1 要素であり，入力データのどこに着目するかを AI が学習することができるようにするものである．例えば入力データが画像データの場合，Attention は画素間の関係性を広く調べることができる．一方例えば CNN では，隣り合う画素の関係しか考慮しないところに違いがある．Attention が提案されたときの課題である自然言語処理は，単語間の係り受けのように少し離れた位置にある単語間の関係を見出すことが重要であり，そこで高性能な結果を示して注目されるようになった．また近年では，自然言語処理だけでなく画像認識等でも性能を上げていることが報告されている [9]．

本論文で扱う Attention は，tensorflow ライブラリの tf.keras.layers.MultiHeadAttention を使用する．ただしオリジナルの実装では MultiHeadAttention の実行時の引数である query, value, key のうち，key に適用するバイアスが常に 1 次元であることに対して，本論文では key の次元に拡張したものを使用している．これは，入力データの個々のデータ間の関係をデータ数が少なくてもより顕著に AI に学習させるためである．また MultiHeadAttention への 3 つの引数である query, value, key には同じ値を入れるといういわゆる self attention という手法を取り扱う．

5.1 問題設定

本問題ではデータ列を “シーケンス” と呼ぶことにする．そして 1 つのシーケンスはさらにいくつかの要素でできているとする．ここでの問題設定は，5 シーケンス，1 つのデータが 7 要素からできているランダムなデータから，(1)1 つのシーケンス内の全要素の合計を求める，(2)2 番目のシーケンスの値を他のシーケンスに加える，の 2 つの教師データを用いて AI が “合計する”,“2 番目を加算する” ということをデータのみから知識獲得できるかを問う．

5.2 学生が学ぶこと

Attention は，入力データ間の関係を学習する機構である．問題設定の (1) のシーケンス内の全要素の合計は，1 つのシーケンスで処理が完結していて他のシーケンスデータとの関係はない．一方 (2) の「2 番目のシーケンスの要素を他のシーケンスに加える」は，シーケンス間の関係ができることになる．具体的には 2 番目のシーケンスの値が他のすべてのシーケンスに対して影響を与える．この様子を Attention という機構はデータのみから導き出すことを学生は体験し，Attention の機能について理解していく．

5.3 学生の学習手順

学生は，AI を学習させるためのトレーニングデータを作成する．そして学習済みモデルから AI の挙動を，モデルをカスタマイズして AI が学習に失敗することから Attention

©2023 Information Processing Society of Japan

図 18: ヘッド数 1 による例

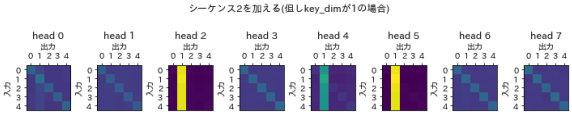

図 19: key dimension 1 による失敗例

の仕組みを理解していく．

5.3.1 シーケンスごとの合計と，他のシーケンスデータを加算することを AI が自動で学ぶことを確認

ランダムに生成された入力データと機械学習モデルを学生は受け取る．ここで学生は，入力データを法則に従って変換した教師データを作成する．この法則は，(1)1 つのシーケンスの全要素を合計するもの，そして (2)2 番目のシーケンスデータの要素を他のすべてのシーケンスデータに加えるもの，の 2 つである．そして AI を学習させ，学習した内容を Attention Score と呼ばれる方法で可視化する．Attention Score とは，図 15 のような形状をしていて，入力データ間の影響を与えている箇所をハイライトするものである．Attention Score を表示するプログラムは予め講師から提供されているとする．

シーケンスごとの合計を求める AI の Attention Score は図 16 のようになる．斜めラインは，自分自身のシーケンスデータからのみ影響を受けていることを表している．一方，2 番目のシーケンスデータを他のシーケンスに加算する AI の attention score は図 17 となる．2 番目のシーケンスを加算しているので，自分自身から影響を受ける斜めラインと，シーケンス 2 から影響を受ける縦ラインの 2 種類が観測されている．学生はまず，自分自身が作った入力データの法則を AI が見つけ出していることを体験する．

5.3.2 ヘッド数，key dimension 数が少なくて AI が失敗する例

学生は MultiHeadAttention のヘッド数を 1 にして機械学習モデルの学習を再度実施する．その結果例を図 18 に示す．1 つのヘッドで縦ラインと斜めラインの関係を AI が学習していることは確認できるがその効果は薄い．AI の学習結果からヘッドが 8 個の時に比べて AI の学習時の損失も約 0.0012 から 0.02 程度に増えていることを学生は確認できる．また図 19 は，MultiHeadAttention の内部状態の次元数である key dimension を 7 から 1 に縮小したときの attention score である．縦ラインは特徴を掴んでいるようだが，斜めラインは薄いものしか観測できていない．損失も約 0.0012 だったものが 0.077 程度の増えていることを学生は確認する．

これらの AI が学習できない例を学生がモデルをカスタマイズして作ることにより，MultiHeadAttention はヘッド数がデータが持つ特徴の数に関係し，またその関係の複雑さは key dimension で表されることを学生は理解する．

6. おわりに

本論文では，AI を作るという観点から AI の仕組みを理解する学習教材について提案した．具体的には，よりシンプルなデータによる例と，AI が学習に失敗する例を用いた．本論文における提案には，次の課題がある．1. 本提案は主に機械学習の推論機構に着目していて，機械学習のもう 1 つの重要な部分であるバックプロパゲーションなどの学習機構については取り扱っていない，2. 現時点において広く利用される高度な AI と本教材には隔たりがあり本教材を学んで最新の AI を理解したとは言い難い，3. 実地による検証ができてない，である．これらの点について今後は検証および検討を進めていきたい．

謝辞 本研究の一部は JSPS 科研費 JP22K12319 の支援によるものです．

参考文献

[1] 内閣府：AI 戦略 2019，https://www8.cao.go.jp/cstp/ai/index.html （2023 年 6 月 15 日確認）（2019）．

[2] 文部科学省：高等学校情報科「情報 II」教員研修用教材 (本編) 第 3 章 情報とデータサイエンス 後半，https://www.mext.go.jp/content/20200609-mxt_jogai01-000007843_007.pdf （2023 年 6 月 15 日確認）(2020)．

[3] 文部科学省：高等学校情報科に関する特設ページ 授業・研修用コンテンツ，https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/mext_01832.html （2023 年 6 月 15 日確認）．

[4] Rumelhart, D. E., Hinton, G. E. and Williams, R. J.: Learning representations by back-propagating errors, *Nature*, Vol. 323, pp. 533–536 (1986).

[5] Gal, Y. and Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks, *Advances in neural information processing systems*, Vol. 29 (2016).

[6] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, Red Hook, NY, USA, Curran Associates Inc., p. 1097–1105 (2012).

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Advances in neural information processing systems*, Vol. 30 (2017).

[8] OpenAI: GPT-4 Technical Report, (online), DOI: 10.48550/arXiv.2303.08774 (2023).

[9] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, *International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/forum?id=YicbFdNTTy⟩ (2021).

©2023 Information Processing Society of Japan