

Neural Network の判定に対する Bayesian Network を利用した判定原因推定

中里克久^{†1}

概要：一般に Deep Learning と呼ばれる Deep Neural Network (DNN)を用いた機械学習は、画像認識などのタスクで高い性能を示すことが知られている。一方で、人間はある判定結果を得た際、その判定原因も知りたいと感じる場合が多いが、Neural Network の出力は多数のパラメタが複雑に組み合わさった式により計算されており、それを人間が解釈可能な形で表現するのは困難である。ひとつの解決策は、原因まで含めた教師データを用いて Neural Network を学習させ、認識結果と原因を同時に出力可能にすることであるが、学習ベースで高精度な原因推定を行うには、細かくラベル付けされた教師データが大量に必要となり、実施コストが非常に高いことが課題である。これを解決するため、Neural Network の内部で最終出力の前に計算される中間出力と Bayesian Network を組み合わせ、Neural Network の学習状態の変更を伴わずに、少ない教師データで判定原因を出力可能にする技術を開発し、その検証実験を行った。結果、2つの判定原因を想定した画像セットに対して、45枚程度の少量の教師データを用いて、正解率約83%とランダム出力より有意に精度の高い原因推定が可能だった。また、同量の教師データで Neural Network を学習して原因推定する場合と比較して、同等以上の推定精度を省資源で実現できることも確認した。

キーワード：Neural Network, Bayesian Network, Deep Learning, CNN, 画像認識, 原因推定

1. はじめに

近年、Deep Learning は画像認識、画像生成、翻訳など、様々な分野で活用が進んでいる。その活用先のひとつとして、ソフトウェアエンジニアリング分野への適用も試みられてきており、コードクローン検出[13][14]、ソフトウェア欠陥予測[15]、プログラム理解[16]、などの例がある。これらは、従来から静的解析、動的解析などによって自動化が試みられてきたタスクであるが、人間の作業を完全に代替できてはいなかった。その課題の解決アプローチのひとつが、Deep Learning などの機械学習技術の適用である。従来技術では、アルゴリズムの設計と実装が主であったのに対し、機械学習利用アプローチでは、重要なのは学習用のデータである。データ上、ある出力、例えばソフトウェア欠陥の判定と、入力の特定のパターンとの間に強い相関があれば、たとえ両者の関係を論理的に説明するのが難しい場合であったとしても、そのパターンに対応するソースコードに欠陥があると判定する。学習用データの質と量を十分に確保できる場合、従来技術が人間作業に及ばなかった部分、すなわち既存のアルゴリズムや論理の不完全性をデータの活用によって補完し、より高度な判定を実現できる可能性がある。なお、機械学習を用いる場合でも、ソースコードを入力として解析のすべてを機械学習ベースで行う、従来技術の置換を目指すアプローチもある一方、Abstract Syntax Tree(AST)などの、従来技術の解析結果またはその一部を入力として、従来技術の補完的な位置付けで利用するアプローチもあり、必ずしも従来技術と排他的に対立するものではないことに留意されたい。

Deep Learning は画像認識などでの実績から、他の機械学習技術よりもパターン認識などの点で優位性があると考え

られるが、Deep Learning のソフトウェアエンジニアリング分野への適用については、研究論文が増え始めたのが2016年頃であり[17]、現時点では初期段階にあると言える。その適用手法は画像認識に近いものから自然言語処理に近いものまで様々であり、確立された手法はまだ無い。しかし、データの面では、OSSの普及やその開発・公開プラットフォームであるGitHubの一般化により、分析や学習のために利用可能なソフトウェア関連の資産の量は年々増大しており、ソフトウェアエンジニアリング分野でのDeep Learningの活用も更に進んでいくと予想される。

Deep Learning の幅広い活用のきっかけとなったのは、2012年の画像認識コンテストでDeep Learningを利用したチームが従来技術に対して大幅に高い認識率を達成して優勝したことである[1]。この時用いられたConvolutional Neural Network (CNN)は、Deep Learningの代表的な手法として、その後も活用し続けられている。

画像認識の従来技術と比較した場合のCNNの利点のひとつは、人間があらかじめ特徴量を定義する必要が無い点である。従来技術では、画像から特徴量を抽出する手法を規定しておき、学習時には特徴量と画像の分類の関係を教師データから学んで画像の分類方法を学習した。この場合、有効な特徴量を定義できていないと、質の高い教師データが大量に存在しても学習はうまくいかない。一方、CNNでは、特徴量に相当する畳み込みフィルタの内容は学習の際に自動的に調整され、最終的に画像の分類に効果があるフィルタが教師データに応じて自動的に決定される。すなわち、従来は人間が定めていた画像分類用の特徴量を、大量の教師データに整合するように様々な特徴量を試して効果が高いものを残すことで高い分類性能を実現する特徴量を

^{†1} 株式会社富士通研究所
Fujitsu Laboratories Ltd.

自動的に決定する挙動となる。CNN の学習の概念図を図 1 に示す。

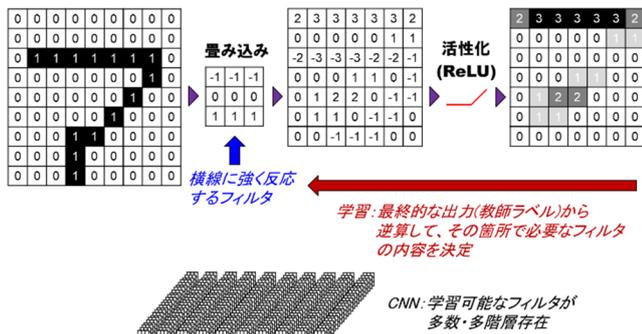


図 1 CNN の学習の概念図

CNN の学習は、効果的な特徴量を人間が決定するという従来技術の困難な課題を解決しているという点で大きな成果だが、反面、どのような判断基準を用いて画像を分類したのか、人間が理解するのが難しくなっているという欠点もある。CNN の畳み込みの個々のフィルタは単純な形状であるが、入力から出力までに多階層かつ多次元で畳み込み処理が行われ、全体としては非常に複雑で、人間が解釈可能な形で処理内容を表現するのが困難なためである。このため、CNN による画像認識は、確かに高い性能を示すものの、人間から見るとブラックボックス的な挙動であるために、判定の原因を人間が理解できず、挙動に対する不信感を抱いたり、判定結果の活用が難しくなるケースも生じ得る。例えば、画像認識を目的とした学習後、ある対象についての認識率が低い場合、再学習によって認識率を改善する必要が生じるが、認識率が低い原因がわからないと再学習に効果的なデータを判断できず、様々な種類の大量のデータを追加した再学習が必要になってしまう。一方で、誤った判定を行った原因が判れば、それを補正するようなデータに絞って追加することで効率的に再学習できる。

判定原因の推定は、今後 CNN をソフトウェアエンジニアリング分野で活用する場合、より重要となると考えられる。例えばソフトウェア欠陥予測に適用した場合、欠陥の有無を高い精度で判定できたとしても、欠陥の修正のためには、何故欠陥があると判定したか、その原因を知る必要がある。

このように、判定原因の推定は、人間の心理的な納得性の向上だけでなく、画像認識システムの学習効率改善や、判定結果を受けた後の人間の行動方針の決定支援など、実用面での効果も大きいと期待できる。

2. 関連研究

CNN を含む DNN や、その他の機械学習のモデルの挙動を説明可能にする技術には、様々な先行研究が存在する。

ひとつのアプローチは、学習モデル自体に、判定の出力に加えて判定の原因を出力する機能も持たせる方法である[2]。原因説明の機能を備えたモデルは有用であるが、既存のモデルやデータを活用できない点で課題がある。

もうひとつのアプローチは、特にモデルが複雑化する DNN において、その挙動を評価する指標を用いたり[3]、可視化するなどして[4][5]、モデルとその挙動を説明する方法である。このアプローチは既存のモデルやデータを活用しつつ、説明の機能を付加するものであって、適用が容易な点が利点である。一方で、専用モデルを構築する前者のアプローチと比べて間接的な説明に留まる点は課題であり、また、判定の原因を知りたいという要望に直接には応えない出力になる点も課題である。

3. Neural Network と Bayesian Network の組み合わせによる判定原因推定

説明可能な機械学習の技術には様々なものがあるが、その目的やアプローチも様々に異なっている。実用場面での効果を重視する場合、その場面で何が求められているのか、要件を明確化し、その要件を少ない実施コストで満たすアプローチを採用することが重要である。

ここでは、画像に写った物体や場面を判定することを主機能とする CNN を用いた画像認識システムにおいて、判定結果が何を決め手として決定されたのかについての情報を追加出力することを目的とする。本稿では、その追加出力を判定原因と呼ぶ。

画像認識における判定原因は単一とは限らず、複数の原因によって判定していることもあり得る。よって、要件に応じて任意の数の判定原因を出力可能であることが必要である。また、判定原因の出力形態として、判定原因を示唆する図表や数値などを出力することもあり得るが、ここでは、人間が明確に認識できる出力形態として、あらかじめ定義した任意の判定原因群のそれぞれについて、該当するかしないか、2 値で判定結果を出力する形態とする。

この条件において、判定原因の出力は期待される機能ではあるものの、主となる機能は画像認識であるので、判定原因の出力のために画像認識の学習をやり直すことは好ましくない。また、必ずしも CNN の内部の挙動を網羅的に説明する必要は無く、判定原因をより低コストで出力可能にすることが求められる。

3.1 判定原因推定のための仮説

CNN をはじめとする多層の Neural Network では、最終的な出力以外に、途中の計算結果、すなわち中間出力を取得することができる。中間出力は、通常の DNN であれば各層を構成するニューロン(ノード)ごとに取得することができ、CNN であれば各層で畳み込むフィルタごとに取得する

ことができる。学習済みの Neural Network では、内部に状態を持つ一部の再帰型 Neural Network を除き、同一の入力に対しては途中のノードの出力も同一となる。各ノードの出力がどのような意味を持つかは、学習の度に変化するのので一般的には判断することができない。ただし、CNN については、入力に近い層では色や直線・曲線など、単純な低次の図形的特徴が学習され、出力に近い層になるほど、犬や猫、人間や自動車など、より複雑な高次の特徴が学習される傾向があることが知られている[6]。

CNN を大量の画像群からなる教師データで学習する場合、ある分類に含まれる画像群が単一の共通した特徴を持つ可能性は低く、様々な特徴を学習して複合的に判定結果を出力するように学習されると考えられる。一例として、ある画像に写っているものが特定の対象物、例えば自動車であるかないか、その一種の分類のみを真偽値や確率で出力する単純な画像認識タスクを考える。自動車と他の物を区別し得る特徴として、四輪のタイヤやボディ形状、ヘッドライトなど、様々な候補があるが、写真の撮影方法を限定していなければ、それらの特徴の画像への写り方は統一されず、写らない場合さえあり得る。また、同じ種類の特徴であっても、写っている大きさが異なれば、同じ畳み込みフィルタには反応せず、別の層の畳み込みフィルタに反応させる必要が生じる。よって、様々な画像の自動車を高精度に認識するには、少数の特徴に強く依存してはならず、様々な写り方があり得る多くの特徴に反応できるように、多数の畳み込みフィルタを学習によって構成する必要がある。従って、CNN の中間出力、すなわち、画像の入力から出力に至るまで何回も行われる畳み込み処理のうちの途中の 1 回分の出力になるが、その多数の中間出力の中には、画像の分類に寄与する様々な特徴に対応する出力が、分散する形で含まれているとみなせる。同時に、ある分類の画像認識において、その主要な判定原因になり得る特徴として人間が認識するいくつかの画像的特徴のうち、Neural Network 上にまったく学習されておらず、中間出力に何も影響しない特徴が存在する場合、その特徴が主要な構成要素となっている画像については認識が失敗する可能性が高く、その分類全体としても認識精度が低くなるはずである。つまり、高い認識精度が実現されている場合、人間が主要な判定原因とみなす特徴は、何らかの形で Neural Network 上に学習され、中間出力にも影響を与えている可能性が高いと言える。これらから、どの特徴を用いて画像が認識されたか、すなわち、画像認識の判定原因を、Neural Network の中間出力を用いて推定できると考えられる。

3.2 判定原因推定のための Neural Network の中間出力の利用

Neural Network の中間出力に画像認識に利用する特徴が反映されているとしても、人間が認識可能な粒度の特徴と

Neural Network の各中間出力の傾向との対応関係は 1 対 1 とは限らず、複数の中間出力と 1 つの特徴が対応付いたり、その逆の関係になっていたりする可能性もある。その関係を直接推定するのは容易ではないので、間接的な方法を用いて推定することが必要になる。具体的には、判定原因を含む画像群を用いる方法が考えられる。

ある判定において、考えられる判定原因を列挙し、それらの判定原因に対応する画像上の特徴を含む画像群と含まない画像群を用意する。学習済みの Neural Network にそれらの画像群を入力した際の中間出力のうち、画像群の分類に有効だとみなせる中間出力を、判定原因となる特徴に対応した中間出力だと判断する。

この方法は、特徴を含む画像群を用いる点で、Neural Network の学習と共通している。学習と比較した場合の利点として想定されるのは、必要な画像枚数を少なく済ませられる点である。学習によって判定原因推定を実現するアプローチでは、大量の学習用データで学習済みの Neural Network を、特徴を含む画像群を用いて再度学習することになる。その実現には様々な方法が考えられるが、代表的な方法としては、学習済みの Neural Network の一部を新たな学習用データを用いて再学習する転移学習が挙げられる[7]。転移学習では、初期状態からの学習と比べて少量の学習用データでも高い判定精度が達成できるとされている。

通常の転移学習では、あるドメインで学習した Neural Network を、別ドメインに転用する用途が想定されている。一方、判定原因推定の場合は、画像認識の判定のみを行っていた Neural Network を、判定原因を追加して出力するように変更する、同じドメインで出力を拡張する目的の学習となるので、少量の学習用データを用いた出力層近傍の再学習によって、画像認識の精度を維持したまま、判定原因も高い精度で追加出力する要求仕様を実現できる可能性は低いと考えられる。

それに対し、中間出力を用いる方法は、中間出力に判定原因に対応する特徴が反映されているという仮定に基づいており、仮定が正しければ、学習済みの Neural Network の状態を変更することなく、その能力の一部を活用するだけで機能を実現できる。画像群は妥当な中間出力を選択するためだけに用いるため、学習アプローチよりも少量で済むと考えられる。

一方で、人間が認識する粒度の特徴と中間出力の関係は 1 対 1 でなく、認識しにくい他の特徴と合わせて中間出力が変動することもあり得る。加えて、入力データとなる画像も、人間が目視した場合には同じ特徴を含んでいるように認識する場合でも、その特徴の位置や色合いに相違があることが一般的である。すなわち、Neural Network の中間出力は入力データに対して決定的に定まるものであるが、人間が認識する画像的特徴に対しては、決定的でなくむしろ確率的に見える挙動を示すと解釈できる。この確率的な

現象から判定原因となる特徴の有無を判断するために、Bayesian Network を用いる。

3.3 Neural Network の中間出力と Bayesian Network を利用した判定原因推定

Bayesian Network は、事象を示す複数のノードと、ノード間の因果関係を示すエッジからなる非循環の有向グラフである。この因果関係は近似的には原因と結果の関係とみなすことができ、原因と結果のノードがネットワーク状に組み合わせられたグラフとなる。ネットワーク内の原因と結果について条件付き確率表(Conditional Probability Table, CPT)をあらかじめ設定しておくことで、各ノードが示す事象の発生確率を計算することができ、観察可能なノードの状態について、できるだけ多く確定させることで、観察不可能な原因ノードの各状態の発生確率を高い精度で計算することができる[8]。

Neural Network の判定原因推定に Bayesian Network を利用する場合、有向グラフの原因側の末端は、想定される複数の判定原因群となる。一方、結果側の末端は、Neural Network の最終出力である画像認識の判定結果となる。これらの中間のノードとして、Neural Network の中間出力群を用いる。なお、Bayesian Network の前提として、接続されていないノード同士は独立している必要がある。ここで、Neural Network の中間出力はひとつの Network 内の部分出力であるから、厳密に独立しているとは言えない。しかし、一般に Neural Network 内の中間出力は数十万から数百万にも及び、個々の相関関係には強弱の幅があると考えられる。例えば、CNN の場合、遠く離れた層の中間出力同士には、極めて弱い相関しかないと考えられるし、隣り合った層であっても、画像上で近い部分でなければ直接的な関係は生じないので、任意に選択した中間出力同士が強い相関を持つ可能性はさほど高くない。迷惑メール分類で Naive Bayes を用いる場合にメール文中の各単語の出現確率が独立であるとみなす場合が多いが[9]、その例と同様、個々の中間出力は独立とみなすこととする。

3.4 判定原因推定に効果的な中間出力の選択

理論的には、Neural Network の全ての中間出力を組み込んだ Bayesian Network を構築することは可能である。しかし、CNN は Neural Network の中では総パラメータ数を抑制可能なアーキテクチャではあるものの、それでも数百万以上のパラメータを持ち[10]、中間出力として利用可能なものも数万以上存在するので、全てを Bayesian Network に組み込むと推論の処理時間が膨大になることが懸念される。また、判定原因推定に寄与しない中間出力を組み込むことにより、推定精度がかえって低くなる可能性もある。よって、判定原因推定に有効な中間出力を選別する必要が生じる。

CNN のひとつの特徴として、入力画像であり、中間出

力も縦×横×畳み込みフィルタの次元数の構造を持っている点が挙げられる。すなわち、ある特徴に反応する畳み込みフィルタがある時に、その特徴が入力画像の左上に存在している場合は左上に対応する中間出力のみが影響を受け、右下に対応する中間出力は影響を受けない。推定しようとしている判定原因は、その内容によって位置に影響を受ける場合と受けない場合があり得るが、一般的にはその判定原因に対応する画像的要素が存在するか否かが重要で、位置の影響は小さいか、まったく受けない場合の方が多いと予想される。よって、ある畳み込みフィルタに対応する中間出力は、縦×横だけ存在する出力値のうち、値が最も大きなものを代表値として使用することとする。この処理は、CNN で時々用いられる処理層である Global Max Pooling 層と同じ処理内容となる。類似の処理として、全体の平均を取る Global Average Pooling もあるが、対象の縦×横のサイズが大きく、かつ一部のみ特徴が存在するようなケースでは、平均を取ると特徴の存在を検出できなくなってしまうので、全体の最大値を取る Global Max Pooling の方が妥当と判断し、これを採用する。

これらの施策により、中間出力として利用可能な出力の数は、畳み込み層×各層の畳み込みフィルタ次元数となる。更に絞り込むため、中間出力の値の大小に着目する。Bayesian Network に中間出力を組み込みやすくするためには、中間出力を連続値のまま使用せず、値を離散化して 2 つ以上のクラスに分類することが望ましい。中間出力の値の基準は出力ごとに異なるため、ある枚数の画像を入力した時の各出力の平均値を各出力の基準値とし、それを用いてクラス化する。ここでは、中間出力を 3 クラス化することとし、大きい方から Large(L), Medium(M), Small(S)と名付ける。この時、判定原因推定に寄与しやすい中間出力は、判定原因となる画像的特徴を含む画像を入力した場合と含まない画像を入力した場合とで、一方が L になり、他方が S になる傾向が強い出力である。この傾向を、実際に Neural Network に画像を入力した際の中間出力の値を用いて評価し、判定原因推定に有効な中間出力を選別して Bayesian Network を構築する。

4. 実験

Bayesian Network を用いた判定原因推定の有効性を確かめるため、既存の学習済み Neural Network と、2 種類の判定原因を想定した実験用の画像セットを用いた実験を行った。使用する Neural Network は、ImageNet [11]で学習された Inception V3 モデル[12]である。この Inception V3 モデルには、判定原因推定のための追加学習などの変更は一切行わず、入力層から出力層に至るまでそのまま使用する。なお、ここでは Inception V3 モデルを用いているが、Bayesian Network を用いた判定原因推定の方式は特定のモデルに依

存しておらず、CNN 一般に適用可能である。

実験の設定としては、Inception V3 モデルでタクシー(cab)である確率が高いと判定される画像群を対象とし、各画像がタクシーであると判定された原因を推定するタスクを想定する。実験で考慮する判定原因は、以下の2種とする。

- 原因“Yellow”：車体が黄色いことを原因としてタクシーと判定
- 原因“Sign”：車体にタクシーサインを備えていることを原因としてタクシーと判定

この2種の判定原因それぞれについて判定原因推定用の Bayesian Network を構築する。ある画像を Neural Network に入力し、タクシーであるとの判定結果を得た際、判定時の中間出力を取得して、それぞれの Bayesian Network に入力する。それぞれの Bayesian Network では、その画像がタクシーと判定された原因として、原因 Yellow が該当するか、原因 Sign が該当するかを独立に推定し、その判定原因が該当すれば True を、該当しなければ False を出力する。判定原因は重複することもあるので、両方 True になることも両方 False になることもあり得る。その出力結果と、結果確認用に人間が事前に画像を分類した結果との整合性から判定原因推定の精度を評価する。

ここで挙げた2種の判定原因は、あくまで人間が想定した代表的と考えられる判定原因であり、必ずしも Neural Network が実際に判定原因としているとは限らない。この2種以外の原因でタクシーと判定されている画像については、2種の原因に関する出力が共に False になることが期待される挙動となる。また、2種としていることに関しては技術的な制約は無く、3種以上に増やすことも可能である。その場合の影響としては、Bayesian Network 構築に必要な画像数が2種の場合より増えることが予想される。

実験用の画像セットについては、人間の目視判断で Yellow の原因のみを含む画像群、Sign の原因のみを含む画像群、Yellow と Sign の両方を含む画像群の3グループに分類した画像セットを用いた。画像分類の例を図2に示す。



図2 実験用画像セットの分類の例

実験に使用する画像の枚数は、Bayesian Network の作成に用いるのが3グループ15枚ずつの計45枚、作成した Bayesian Network の検証に用いるのも同じ構成で同数の45枚である。単純に Bayesian Network の判定原因推定精度の向上だけを目的とする場合、Bayesian Network 作成に用い

る画像が多いほど精度が向上することが期待される。また、Yellow も Sign も含まないがタクシーである画像群や、タクシーでない画像群などを用いることで、Yellow と Sign のみに強く反応する中間出力を選別できる可能性が高まると考えられる。しかし、この Bayesian Network を使用する方式で想定している利点は、Neural Network の再学習よりも必要な画像枚数を削減することであるので、容易に収集可能と考えられる45枚の画像セットに留めて実験を行った。

4.1 Bayesian Network の作成

実験用の画像セットのうち、Bayesian Network 作成用の45枚の画像群を学習済みの Neural Network、ここでは Inception V3 モデルに入力し、それぞれの画像の処理時の中間出力の値を採取する。なお、Inception V3 では処理が分岐し、複数の層が並列に処理される箇所が存在するが、中間出力としてはそのような箇所は用いず、処理が集約される箇所のみを対象とした。これにより、中間出力の採取対象は16層で、二次元の平面構造を持つ場合は最大値を採用する処理により、中間出力の数は10448個となる。

それぞれの中間出力について、総入力画像数と同数の45個の値が採取できるので、平均値を計算して中間出力の3クラス分類の基準値とする。3クラス分類は、平均値の125%以上の値の場合はL、平均値の75%未満の値の場合はS、それ以外の場合はMとする。この操作により、想定している各原因、この例では Yellow と Sign について、Yellow が含まれる画像30枚、Yellow が含まれない画像15枚、Sign が含まれる画像30枚、Sign が含まれない画像15枚、それぞれについての3クラス分類結果を得ることができる。

ここで、判定原因推定に寄与しやすい中間出力は、判定原因の有無に応じて値の傾向が明確に変化するものと考えられるので、中間のMのクラスを除き、LとSのクラスに分類された画像枚数で判定への寄与の度合いを判断する。Yellow の判定では、Yellow を含む画像と含まない画像とでLとSの大小関係が逆になっていることが望ましく、そのような中間出力を選別する。その中で、LとSの比が大きいものが Yellow の推定に寄与する中間出力であるとみなせ、この寄与の度合いを判定寄与指数(Distinguishing Index)と呼ぶこととすると、ある中間出力についての Yellow に関する判定寄与指数 D_Y は以下のように定義される。

$$D_Y = \frac{\max(Y_L, Y_S)}{\min(Y_L, Y_S)} \times \frac{\max(\bar{Y}_L, \bar{Y}_S)}{\min(\bar{Y}_L, \bar{Y}_S)} \\ = \left(\frac{|Y_L + Y_S| + |Y_L - Y_S|}{|Y_L + Y_S| - |Y_L - Y_S|} \right) \times \left(\frac{|\bar{Y}_L + \bar{Y}_S| + |\bar{Y}_L - \bar{Y}_S|}{|\bar{Y}_L + \bar{Y}_S| - |\bar{Y}_L - \bar{Y}_S|} \right)$$

ここで、 Y_L は原因 Yellow を含む画像群を入力した時に当該中間出力がLとなった回数で、 Y_S はSとなった回数であ

る。また、 \bar{Y}_L は原因 Yellow を含まない画像群を入力した時に当該中間出力が L となった回数で、 \bar{Y}_S は S となった回数である。 D_Y は常に正となる。

各中間出力について判定寄与指数 D_Y を計算して値の大きい順に並べ、上位の中間出力をノードとして組み込んだ Bayesian Network を作成する。組み込むノード数は可変であり、その決定については後述する。ここでは、上位 15 ノードを組み込むこととし、各ノードを $N_0 \sim N_{14}$ と呼び、ノードを総称する場合は N_i と呼ぶ。各ノードの遷移元は判定原因であり、この場合は Yellow に対応するノードとなる。遷移先は Neural Network の出力である画像認識の判定結果とする。ただし、多くの場合、同一の判定結果に対して判定原因を推定する用途で用いるため、遷移先を設定しなくても良い。この場合は、Naive Bayes と同等の構造となる。実験では判定結果のノードは設定しなかった。この時の Bayesian Network の構成を図 3 に示す。

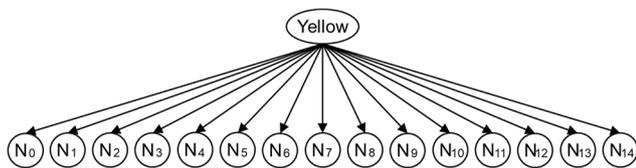


図 3 実験時の Bayesian Network

CPT については、Yellow の原因を含む確率である $P(\text{Yellow})$ と、ノード N_i についての $P(N_i|\text{Yellow})$ の L, M, S それぞれの確率を設定する。 $P(\text{Yellow})$ は計算の目的となる確率であって、最初に設定するのは単なる初期値であるので、True/False それぞれ 0.5 で良い。 $P(N_i|\text{Yellow})$ は、L, M, S それぞれについて設定する必要があるが、Bayesian Network 作成用の 45 枚の画像についての、 N_i での中間出力の L, M, S の分類数を、総枚数で割った値を確率とする。

以上により Yellow の原因を含む確率である $P(\text{Yellow})$ を計算するための Bayesian Network が構築できる。Neural Network でタクシー(cab)と判定された任意の画像について、Yellow の原因を含むか含まないかを推定するには、その画像の Neural Network の中間出力のうち、Bayesian Network で利用しているノード N_i の中間出力を取得し、Bayesian Network 作成時と同じ平均値を用いて L, M, S のクラス分類を行う。求めた分類結果によって Bayesian Network の $P(N_i)$ が確定するので、Yellow の原因を含む確率である $P(\text{Yellow})$ を計算することができる。

同様の手順で、Sign の原因を含む確率である $P(\text{Sign})$ を計算する Bayesian Network も構築することができる。Yellow の推定に寄与する中間出力と Sign の推定に寄与する中間出力は重複することもあるので同一の Bayesian Network として構築しても良いが、それぞれの計算は独立に行うことができ、一般に Bayesian Network の規模が大きくなると確

率計算が遅くなるため、別々に構築しても良い。今回の実験では、原因ごとに別々の Bayesian Network を構築した。

4.2 未知画像に対する判定原因推定

作成した Bayesian Network の性能を確認するため、作成に利用した画像とは別の 45 枚の画像群を用いて、Bayesian Network の出力を検証した。45 枚の画像群は事前を目視で Yellow と Sign の原因を含むか否かを評価しており、その内訳は Bayesian Network 作成に用いた画像群と同様、Yellow のみを含む画像が 15 枚、Sign のみを含む画像が 15 枚、Yellow と Sign の両方を含む画像が 15 枚である。これらの画像群に対し、Bayesian Network による Yellow と Sign の判定原因推定モデルを適用し、推定モデルの出力と目視判断の画像分類との一致率によって正解率を求めた。表 1 に得られた正解率を示す。表中、「原因 Yellow に関する推定」の列と「原因 Sign に関する推定」の列は、それぞれの原因が含まれていると推定した回数と推定機会の母数を示しており、例えば Yellow のみの画像セットに対しては、Yellow の推定は 15/15 に、Sign の推定は 0/15 になるのが理想的な推定結果となる。実際の結果を見ると、Yellow のみの画像種別に対する Sign の誤推定が 6 回と比較的多く記録されており、他の画像種別と比較して正解率が低くなっているが、3つの画像セット全体の推定結果は、全 90 回の推定機会のうち正解が 75、不正解が 15 で、正解率は約 0.83 となった。また、この 45 枚の画像に対する 90 回の推定機会での正解率の有意性を検定すると、2 値での推定なので正解率 0.50 とした場合や、この画像セットのすべての画像に対し Yellow と Sign が共に含まれていると推定した場合の正解率 0.67 と比較して、有意水準=0.05 で有意に正解率が高いと確認できた。

表 1 判定原因推定の正解率

画像種別	原因 Yellow に関する推定	原因 Sign に関する推定	正解率
Yellow のみ	14 / 15	6 / 15	0.77
Sign のみ	2 / 15	14 / 15	0.90
Yellow と Sign の両方	11 / 15	14 / 15	0.83
合計	--	--	0.83

4.3 効果的な Bayesian Network の構築手法の検討

Bayesian Network の構築に利用するノード数によって、判定原因推定の正解率は変化する。タクシー画像セットにおけるノード数と正解率の関係を図 4 に示す。横軸は Bayesian Network に含めるノード数で、判定寄与指数順に上位 1~3000 ノードの範囲で各ノード数を用いた Bayesian Network による Yellow および Sign の原因推定の正解率を測定し、片対数で表示している。

図 4 から、原因 Yellow と原因 Sign のどちらも、ノード数によって正解率が 0.6~0.9 の範囲で変化しているが、変化の傾向や、正解率が最高となるノード数は両者で異なっている。また、図中には、各ノード数での平均および最小の判定寄与指数も示しているが、正解率との間に明確な相関関係は確認できなかった。

これらから、Bayesian Network の構築に利用するノード数によって判定原因推定の正解率は変化するため推定に効果的なノード数を求めることが望ましいものの、判定原因ごとにノード数に応じた正解率の変化の傾向が異なるため、最適なノード数を決定することが難しいという課題があることがわかる。判定原因ごとに分類済みの画像群を用いて図 4 のように正解率を比較してノード数を決定する方法はひとつの選択肢となるが、理想的には、画像群を用いることなく最適なノード数を決定する手法が求められる。

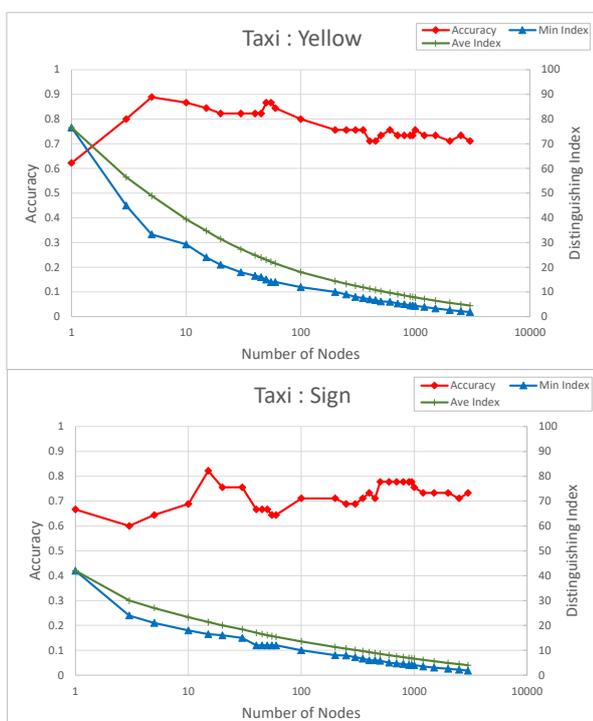


図 4 ノード数と判定原因推定の正解率の関係

効果的なノード数を画像データ非依存で導出する汎用的な手法を検討するため、タクシー画像セットに加えて、同じように 2 つの判定原因を想定して人手で分類したデータセットを 2 セット(野球の画像セットと火山の画像セット)用意し、また、その 3 セットそれぞれについて、Bayesian Network 構築用の 45 枚と検証用の 45 枚を逆にしたデータセットも使い、合計 6 つのデータセットと 12 の判定原因について、図 4 と同様のノード数と判定原因推定の正解率の関係調べた。そのグラフを図 5 に示す。なお、これらの図の横軸および縦軸の範囲は統一しており、ノード数は 1~10000 の対数表示、正解率は 0~1、判定寄与指数は 0~

100 である。判定寄与指数は数か所で 100 を超えており、最大値は 192 となっているが、グラフの視認性のために 100 までの範囲で表示している。

図 5 を見ると、ノード数の増加に応じて正解率は高くなる例も低くなる例も存在し、すべての例で共通している明確な規則性は無いと考えられる。また、正解率と平均および最小の判定寄与指数の関係についても、明確な相関関係は確認できなかった。一方、図中で左右に並べたグラフは、同種の判定原因について、Bayesian Network 構築用と結果検証用の画像群を逆転させたデータセット同士の正解率を示しているが、これらを比較すると、ノード数に応じた正解率の変動の傾向は異なる例が散見されるものの、正解率の平均的な水準は左右でほぼ同等になっていることがわかる。

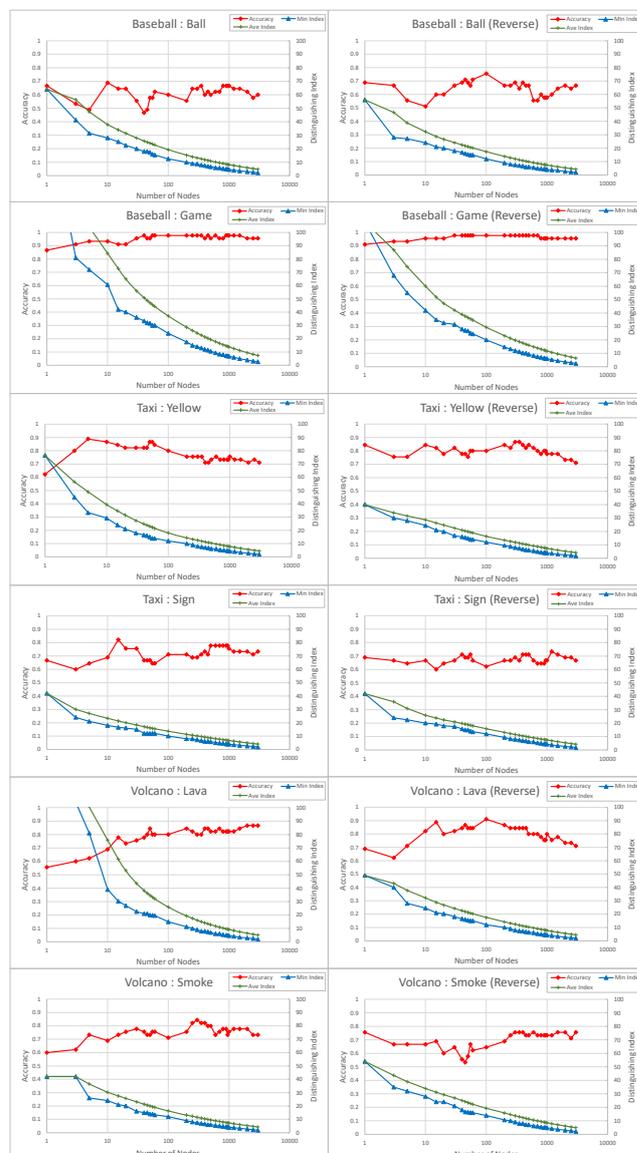


図 5 6 データセット・12 判定原因についてのノード数と判定原因推定の正解率の関係

各判定原因についての、通常データセットと逆転データセットの正解率の平均値の比較を表 2 に示す。平均正解率は、判定原因間で比較すると最大で 0.3 以上の差が生じているが、通常データセットと逆転データセットとの間の比較では、最大でも 0.06 程度の差となっており、判定原因の相違による差は、同一判定原因で画像群を入れ替えた場合の差より明らかに大きいことが確認できる。前述の通り、ここで設定している判定原因は人間が想定したもので、実際の Neural Network による画像認識の際にこれらの判定原因がどの程度影響しているか、その影響度は判定原因ごとに異なっていると考えられる。また、判定原因自体を画像的に認識できなければ画像認識の判定原因にはならないと考えられるが、その画像的な認識の難易度も、判定原因ごとに異なっていると考えられる。これらのことが、表 2 のように判定原因ごとに平均の正解率に相違があることの原因になっている可能性が高い。

以上より、正解率の水準は概ね判定原因の内容によって決まるものの、使用するノード数によって若干の高低の変動が生じる。変動の傾向は判定原因ごとに異なり、事前に予測するのは困難であると考えられる。しかし、図 5 から、同じ内容の判定原因で、Bayesian Network 構築用と検証用の画像群を逆転させたデータセット間においてもノード数に応じた正解率の変動の傾向は異なっていることから、これは判定原因の内容だけでは決定されない確率的な事象であると解釈できる。従って、表 2 の各判定原因のノード数間の平均の正解率と比較して、相対的に高い正解率を多くの判定原因で期待できるノード数を選択することで、ある程度最適に近いノード数となると考えられる。

表 2 判定原因ごとの平均正解率

判定原因	通常データセットの平均正解率	逆転データセットの平均正解率
Baseball: Ball	0.6093	0.6409
Baseball: Game	0.9577	0.9642
Taxi: Yellow	0.7720	0.7986
Taxi: Sign	0.7176	0.6746
Volcano: Lava	0.7921	0.8000
Volcano: Smoke	0.7534	0.6961
平均	0.7670	0.7624

図 5 に示している正解率を、表 2 の判定原因ごとの平均正解率で割った正解率の平均値比は、判定原因ごとに異なっている正解率の水準の影響を除外し、ノード数が正解率に与える影響のみを示す値として、判定原因間での比較や計算が可能と考えられる。図 6 に、6 つのデータセットと 12 の判定原因についての正解率の平均値比を、ノード数ごとに平均を取った結果を示す。この結果から、10 ノード未

満では明らかに正解率の期待値が低くなり、300 ノードで最も期待値が高くなるのがわかる。前述の通り、最高の正解率を示す最適なノード数はデータ依存で決まるので、ノード数決定用の画像セットが用意できる場合はそれを用いて決定するのが望ましいが、それが用意できない場合、300 ノードを用いると、他のノード数と比較して高い正解率を得られる可能性が相対的に高くなると考えられる。

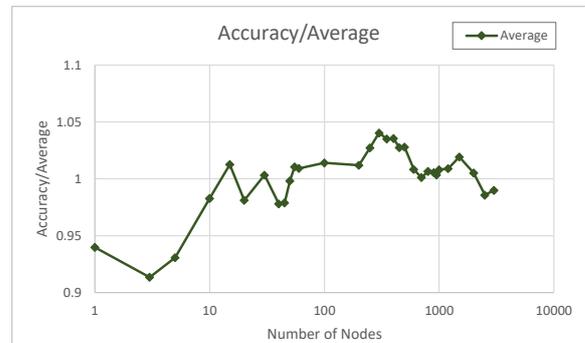


図 6 ノード数と正解率の平均値比のデータセット間での平均との関係

4.4 学習アプローチによる判定原因推定との比較

Bayesian Network を利用した判定原因推定方式(以下, BN 方式)は、原因ごとに分類済みの画像セットを用いる点で、前述の Neural Network の再学習によって原因推定を実現する学習アプローチ(以下, 学習方式)と共通している。BN 方式の利点は学習方式よりも必要な資源を削減できる点だと想定していたが、実際に比較して検証した。

表 1 と同様の条件での学習方式の判定原因推定結果を求めるため、実験で使用したタクシー画像セットを用い、Inception V3 モデルの出力層近傍のみを原因推定用に再学習した。原因 Yellow の推定と原因 Sign の推定は 2 つの異なるモデルとして学習し、ある程度学習の収束傾向が認められた 50 回繰り返して学習した。その推定モデルを用いた学習方式の判定原因推定結果を表 3 に示す。BN 方式の表 1 と比較すると、BN 方式の方がやや正解率が高いが、サンプルの少なさから有意差までは認められない。BN 方式は同じデータを用いる限り、何度 Bayesian Network を構築しても推定結果は同一になるが、学習方式では学習時のバッチ構成の差異などによってモデルの内容も推定結果も試行毎に変わる。3 回試行時の平均正解率は 0.79 であった。

図 7 に、学習方式と BN 方式の正解率と処理時間の比較を示す。学習方式は繰り返し回数が 10 回~100 回の結果を示しているが、明らかに学習不足の 10 回を除くと正解率は同程度で、BN 方式の正解率の方が高い。

処理時間は準備フェーズ(Training)と推定フェーズ(Prediction)に分けて表示しており、それぞれ 3 回試行時の平均値である。学習方式では Neural Network の学習が Training に、学習済み Neural Network を用いた原因推定が

Prediction に含まれる。BN 方式では Neural Network の中間出力の選択までが Training に、判定原因推定対象の画像の Neural Network による判定とその際の中間出力の取得、およびそれを反映した Bayesian Network の作成と判定原因の推定処理が Prediction に含まれる。図 7 の学習方式の各繰り返し回数での処理時間の相違から明らかなように、絶対的な処理時間は設定次第で大きく増減するため比較しにくい。同程度の正解率を実現するための処理時間という観点で見ると、BN 方式が学習方式より優れているとみなせる。また、BN 方式の Prediction が長くなっている点については、方式の相違から BN 方式の方が Prediction に分類している処理が多くなっている点と、学習方式が CNN の一般的な処理のみで構成され、Deep Learning 用ミドルウェアの活用によって GPU 対応などの処理の最適化が行われているのに対し、BN 方式では独自実装部分が多く、処理の最適化が不十分な点が原因と考えられ、後者に関しては実装の改善による処理時間短縮の余地が大きいと考えられる。

また、モデルのデータ容量の観点でも、学習方式では Yellow と Sign を判定するそれぞれのモデルが必要となり、Inception V3 ベースでは、圧縮ありのバイナリ形式でも 2 つ合計で 192MB のサイズとなるが、BN 方式では、非圧縮の JSON 形式で必要なデータを保存した場合でも 8.2MB のサイズで済み、処理時間を含め、学習方式よりも必要な資源が少なく済むとの想定は正しかったと評価できる。

表 3 学習方式による判定原因推定の正解率

画像種別	原因 Yellow に関する推定	原因 Sign に関する推定	正解率
Yellow のみ	15 / 15	6 / 15	0.80
Sign のみ	7 / 15	13 / 15	0.70
Yellow と Sign の両方	15 / 15	9 / 15	0.80
合計	--	--	0.77

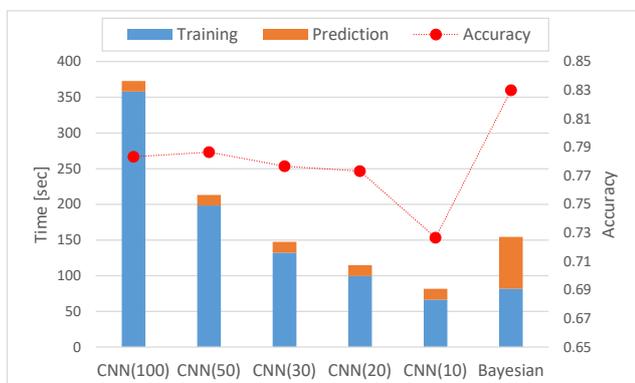


図 7 学習方式(CNN)と BN 方式(Bayesian)の判定原因推定
 正解率と処理時間の比較

5. まとめと今後の課題

CNN による画像認識結果の判定原因を推定するため、CNN の中間出力を抽出し、ノードとして組み込んだ Bayesian Network を構築し、未知の画像を CNN で認識する際、その中間出力と Bayesian Network を用いて判定原因も同時に推定する手法を考案した。この手法では、大量の画像で学習済みの Neural Network を変更せずにそのまま用いるために高い画像認識精度を維持することが可能であり、また、判定原因推定に有効な中間出力を選別するために用いる、判定原因を含んだ少数の画像群があれば実現可能な点が利点であると考えられ、実験により、45 枚程度の少量の画像群を用いた場合でも、他の 45 枚の画像群に対して 2 つの判定原因を約 83% の正解率で推定することができた。これは、同じ画像群を用いて Neural Network の学習によって判定原因推定を行った場合と比較して、同等以上の推定精度を、より省資源で実現できていると評価できる。

ソフトウェアエンジニアリング分野への CNN の適用では、学習用の教師データの不足が課題になると考えられる。例えばレビューや分析に用いる場合、一般的な画像認識に比べて教師ラベルの作成難易度が高く、高コストとなるためである。従って、CNN の学習には粗い評価のみを行った大量の低コストな教師データを用い、細かい評価を行った少量の高コストな教師データと本稿の技術によって詳細な評価も出力可能にするなど、工夫が必要になると考えられる。今後は、そのようなユースケースでの有効性の検証が課題となる。

参考文献

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks", In NIPS, 2012
- [2] D. H. Park, L. A. Hendricks, Z. Akata, B. Schiele, T Darrell, and M. Rohrbach. "Attentive Explanations: Justifying Decisions and Pointing to the Evidence", arXiv:1612.04757, 2016
- [3] P. W. Koh, and P. Liang. "Understanding Black-box Predictions via Influence Functions", In ICML, 2017
- [4] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", arXiv:1610.02391, 2016
- [5] A. Shrikumar, P. Greenside, and A. Kundaje. "Learning Important Features Through Propagating Activation Differences", arXiv:1704.02685, 2017
- [6] M. D. Zeiler, and R. Fergus. "Visualizing and Understanding Convolutional Neural Networks", In ECCV, 2014
- [7] P. Agrawal, R. Girshick, and J. Malik. "Analyzing the Performance of Multilayer Neural Networks for Object Recognition", In ECCV, 2014
- [8] J. Pearl. "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference", Morgan Kaufmann, 1988
- [9] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. "A

- Bayesian Approach to Filtering Junk E-Mail”, In AAAI, 1998
- [10] A. Canziani, E. Culurciello, and A. Paszke. “An Analysis of Deep Neural Network Models for Practical Applications”, arXiv:1605.07678, 2017
- [11] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. “ImageNet: A Large-Scale Hierarchical Image Database”, In CVPR, 2009
- [12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. “Rethinking the Inception Architecture for Computer Vision”, In CVPR, 2016
- [13] M. White, M. Tufano, C. Vendome, and D. Poshyvanyk. “Deep Learning Code Fragments for Code Clone Detection”, In ASE, 2016
- [14] L. Li, H. Feng, W. Zhuang, N. Meng, and B. Ryder. “CCLearner: A Deep Learning-Based Clone Detection Approach”, In ICSME, 2017
- [15] J. Li, P. He, J. Zhu, and M. R. Lyu. “Software Defect Prediction via Convolutional Neural Network”, In QRS, 2017
- [16] L. Mou, G. Li, L. Zhang, T. Wang, and Z. Jin. “Convolutional Neural Networks over Tree Structures for Programming Language Processing”, In AAAI, 2016
- [17] X. Li, H. Jiang, Z. Ren, G. Li, and J. Zhang. “Deep Learning in Software Engineering”, arXiv: 1805.04825, 2018