2019.1のブログ：「ＡＩシステム検証へのニューロンカバレッジの有用性について」
（→　http://www.1968start.com/M/blog/index.html#1901　）

「ＡＩシステム検証へのニューロンカバレッジの有用性について」

■ニューロンカバレッジの文献調査のきっかけ

情報処理学会誌「情報処理」の 2019 年１月号の以下の解説論文について：
「機械学習工学：3．機械学習応用システムのテストと検証」
（情報処理，Vol.60，No.1，pp.25-33）

本解説の研究動向に以下の記述があった：
『深層学習で得たモデル（ニューラルネットワーク）も一種のプログラムコードであるが、
コードの分岐カバレッジを 100％にする（一式のテストによってすべての分岐が
少なくとも１回ずつは実行されるようにする）ことなどは容易であり、
それをもって十分にさまざまな可能性をテストしたとは言いがたい。
このため、ニューラルネットワーク専用のさまざまなカバレッジ指標が提案されている
（ニューロンカバレッジ〔参考文献１〕等)』

これに対し、以下の疑問を持った：
★ニューロンカバレッジを 100％にすることは容易だろうか？

そこで、以下の参考文献１を読んでみた。
Kexin Pei，Yinzhi Cao，Junfeng Yang，Suman Jana ：
DeepXplore: Automated Whitebox Testing of Deep Learning Systems,
The 26th ACM Symposium on Operating Systems Principles，ACM，pp.1-18 (Oct. 2017)

■疑問の理由

訓練用データセットで学習終了後に、テスト用データセットでカバレッジ測定し、一度も活性化しなかったｎ段目のノードを活性化させるテストデータをどのように作成するか？
→たとえば、画像認識の応用において、以下の方法は容易だろうか？
＊そのノードへの入力エッジの重みの大きいものからいくつか選択し、
そのノードを活性化させるｎ-1 段目のノードセットを求める。
＊この方法を繰り返し、１段目のノードセットをもとめる。
＊最後に、この１段目のノードセットを同時に活性化させるテストデータを作成する。
（アプリケーションが画像認識の場合、画像を生成する必要がある）

■文献の読後感

・本論文での新たな入力の生成方法は、既存の入力データを少しずつ変化させて、
出力結果が元の入力データの出力結果と異なるデータを発見するというもの。

・この新たに生成された入力データは、ＡＩアプリケーションの精度の改善に有用で、
ニューロン網羅率は少し向上するが、試行錯誤的な方法なので、100％網羅が目標ではない。

・結果の異なる類似のテストデータを用いる方法は、
従来のプログラムのテストにおける限界値／境界値テストに似ている。
（拙著「ソフトウェア工学」の7.2節「機能テスト（ブラックボックステスト）」参照）
http://www.1968start.com/M/lecture/SE3index.html

・測定したニューロン網羅率をどのように利用しているのかは不明。
ホワイトボックステストというよりブラックボックステストに近いのでは？

■文献の内容と断片的コメント

Kexin Pei, Yinzhi Cao, Junfeng Yang, Suman Jana ：
DeepXplore: Automated Whitebox Testing of Deep Learning Systems,
The 26th ACM Symposium on Operating Systems Principles, ACM, pp.1-18 (Oct. 2017)

【ABSTRACT】
・実世界の深層学習（ＤＬ）システムのシステマチックなテストのための、
初めてのホワイトボックスフレームワーク、DeepXplore、を設計、開発、評価する。
・最初に、テスト入力で訓練したシステムの各部分をシステマチックに測定するための
ニューロン網羅率を導入し、その網羅率を高める入力を見出す方法を提示する。
・自動運転システムなどの訓練済みシステムの多くの誤りを効率よく見つける。
・生成された入力は、訓練済みのシステムの再訓練に用いて、正確性を３％まで向上する。

【1 INTRODUCTION】
・キーアイデアは、人間の目に同じに見えるが、深層学習モデルが異なる識別をするような、
元の画像とは異なる合成画像を生成すること
・図１：本システムを用いて、自動運転システムへの入力画面を少し暗くすると、
左カーブの道路で、間違って右カーブの操作をした例
・ランダムに抽出したテスト入力を用いて、ニューロン網羅率が10％以下であっても、
コード網羅率は100％達成可能なものがある。
・本システムが生成した入力は、ランダムに抽出した入力よりも 34.4%および 33.2%高い
ニューロン網羅率を達成した。

【2 BACKGROUND】
【2.1 DL Systems】
・図２：従来のソフトウェア開発と機械学習（ＭＬ）システムの開発プロセスの違い
【2.2 DNN Architecture】
・図３：深層ニューラルネットワークの説明
・図４：従来のプログラムでの分岐処理による識別と
深層ニューラルネットワークＤＮＮでの（車と顔の）識別方法との違い
【2.3 Limitations of Existing DNN Testing】
＜Expensive labeling effort＞
・従来のＤＮＮテスト技術は、実行して確認するテストしかないので大変

＜Low test coverage＞
・従来のＤＮＮのテストでは、ＤＮＮの異なるルールを網羅する試みが含まれていないので、
たとえば、データを訓練用と訓練後のテスト用にランダムに分割して利用する場合、
テストでは、訓練用データで学習したルールの一部分のみが対象の恐れがある。
・これらの入力データセットは網羅率を最大化するようには考えられていない。

＜Problems with low-coverage DNN tests＞
・ランダムなテストを多く実施した後も、多くの間違った振る舞いが検出されずに残る。
・もし、鼻＆赤に特徴づけられたニューロンが高い確率で「車」と誤って識別するならば、
ピエロのような赤い鼻の画像を正しく識別する可能性は小さいにもかかわらず、
通常のテストでは見逃されるだろう。

【3 OVERVIEW】
・図５：本システムの処理フロー
元になる入力 →【テスト対象のＤＮＮ】→ 出力＆中間層のニューロンの傾斜
→【差異化とニューロン網羅率の最大化を目標に、傾斜が上昇するように最適化】
→ 差異を誘導する入力を出力

・図６：類似の複数のＤＮＮが異なる識別をするような入力
(a) 複数のＤＮＮが共に「車」と識別する例　(b)「車」と「顔」と異なる識別する例

・システムは、一方のＤＮＮが車と認識する可能性を最大化し、他方のＤＮＮが可能性を
最小化するように入力を変更して、二つのＤＮＮが異なる結果を出すように試みる。
・システムは、中間層の不活性ニューロンを活性化して、
できるだけ多くのニューロンを網羅することを試みる

・図７：元となる入力から開始して、差異をもたらすテスト入力を見出すような傾きの上昇

【4 METHODOLOGY】
【4.1 Definitions】
＜Neuron coverage＞
・ニューロン網羅率とは、ある入力ｘに関して
閾値を超える出力を示したニューロンの数を、全ニューロン数で割った値

＜Gradient＞
・ｙは、任意のニューロンに関して、パラメータθの条件下での入力ｘに対する出力
・内部のｆの計算は、直前の階層からの入力を用いて計算し、直後の階層への出力とする
・傾きＧは、ｙの微分値。入力変動に伴う出力変動の度合い。
ｙを出力するニューロンの階層（中間層、出力層）から
入力ｘを受け取った入力階層に至る「階層単位の傾き」から計算する（？）

【4.2 DeepXplore algorithm】
・テスト生成の目的は、「微分値？」と「ニューロン網羅率」の最大化

＜Maximizing differential behaviors＞
・複数のテスト対象のＤＮＮに対して、異なる分類になる入力を導く
・ｎ種類のＤＮＮを対象として、ｋ番目のＤＮＮは、Ｆｋの機能、すなわち、
　入力ｘに対して、あるクラスへと識別する確率を示すベクトルを出力する、とする
・テスト入力生成アルゴリズム：
　λ1は、k番目とｊ番目のＤＮＮが元のｘで同じ出力、変形後は別の出力とするパラメータ
＜Maximizing neuron coverage＞
・活性化していないニューロンを活性化するように入力を変形して、網羅率を最大化する
★コメント：変形後のｘが元のｘと異なるクラス｛車か顔か｝と認識するものを選択すれば、
　統計的・確率的に網羅率ＵＰが期待できるということか？
　一度も活性化していないニューロンを活性化させる入力を生成するアルゴリズムではない。
＜Joint optimization＞
＜Domain-specific constraints＞
＜Hyperparameters in Algorithm 1＞

【5 IMPLEMENTATION】

【6 EXPERIMENTAL SETUP】
【6.1 Test datasets and DNN】
・表１：本システムの評価に用いたＤＮＮとデータセットの詳細
【6.2 Domain-specific constraints】
＜Image constraints (MNIST, ImageNet, and Driving)＞
・明るさ、コントラストなどの変化で、異なる認識のデータになった
・アプリの精度を改善するテストとしては有効
・図８：
　奇数行は元のテスト入力、偶数行は、結果が異なるように生成されたテスト入力。
　各行の最初の3件は自動運転用、次の3件は手書き数字用，最後の3件は画像用のテスト入力。
　1,2行は明るさ変更、3,4行は小さな長方形を混入、5.6行は複数の微小の長方形を混入
＜Other constraints (Drebin and Contagio/VirusTotal)＞

【7 RESULTS】
＜Summary＞
・本システムは、複数のテスト済みのＤＮＮに関して、多くの間違った動作を見つけ出した
・表２：元データから無作為に選択した2000件から生成された差異結果をもたらす入力数。
　　ニューロン網羅率と差異結果をもたらす入力の割合を最大化するようパラメータ値設定
・表３と表４：テストされたＤＮＮの中で誤判断を引き起こすように、
　　本システムによって生成された差異結果をもたらす入力の例を示す。
【7.1 Benefits of neuron coverage】
・コード網羅率100％でも、ニューロン網羅率は34％以下なので、後者が重要
・同じクラスよりも異なるクラスからのテスト入力のほうが、
　ユニークなニューロンを活性化する傾向アリ
・ニューロン網羅率は、入力を用いて訓練したＤＮＮルール群の数とタイプの良い評価に使える

＜Neuron coverage vs. code coverage＞
・表６：コード網羅率100%でも、ニューロン網羅率は34%以下なので、
　　ニューロン網羅率のほうが、より良い評価基準である。
・ニューロン活性化の閾値を.75とした。
★コメント：閾値0.75が高いか低いか、判断できず。
　　比較するならコート網羅ではなく、分岐網羅でやるべきだが．比較は有用かな？

＜Effect of neuron coverage on the difference-inducing inputs found by DeepXplore＞
・ニューロン網羅率最大化の初期の目標は、異なる結果となる入力の生成
　★コメント：これは有意義
・表５：
　ニューロン網羅率を考慮したほうが（その網羅率は高く）、
生成された差異をもたらす入力の距離が大きくなる（差異が大きくなる）ことを示す。
　ニューロン網羅率の増加の割合は1～2%と小さいが、
生成された差異をもたらす入力の距離の増加の効果は大きい。
　ニューロン網羅率の高いところでの増加は難しいが、小さな増加でも大きな効果をもたらす。
生成される差異入力の数は少なくなるが、距離は大きくなっている。

＜Activation of neurons for different classes of inputs＞
・表７：
　同じクラスに識別される入力では、より多くの共通の活性化されるニューロンを有する
　★コメント：当たり前

【7.2 Performance】
＜Neuron coverage＞
・図９：本システムのニューロン網羅率は高い
・閾値を高くするとニューロン網羅率は下がる
　★コメント：当たり前
＜Execution time and number of seed inputs＞
＜Different choices of hyperparameters＞
＜Testing very similar models with DeepXplore＞

【7.3 Improving DNNs with DeepXplore】
＜Augmenting training data to improve accuracy＞
・本システムが生成した誤認識をもたらす入力を再訓練に使用して精度を改善できる
＜Detecting training data pollution attack＞

【8 DISCUSSION】
＜Causes of differences between DNNs＞
・本システムは、テスト済みのDNNのすべてに対して、効率よく誤認識のケースを発見できる。

＜Overhead of training vs. testing DNNs＞

＜Limitations＞
・本システムは、ソフトウェア分析とは異なるテスト技術なので、限界がある。
・本システムの利用には、同じ機能を有する少なくとも二つ以上のＤＮＮが必要。
　ＤＮＮの違いがわずかの場合、異なる認識をもたらす入力を見つけるのに時間がかかる。
・少なくともひとつのＤＮＮがほかと異なる結果を出す場合に誤認識を検出できる。
　もし、すべてのＤＮＮが同じ間違いをしたら誤認識するテストケースを生成できないが、
　すべてのＤＮＮが同じ間違いする可能性は小さい。

【9 RELATED WORK】
＜Adversarial deep learning＞
・従来の文献は、正しく識別している画像を少し変えて適用することを馬鹿にしている。
　変形画像が人間の目で見分けられない場合すらも。
・従来の文献では、ニューロン網羅率が低いので、異なるタイプの誤認識を見つけにくい
・目に見える変形は人手による検査を必要とするので、自動生成プロセスでは、
　小さな、目ではわからない変形に限られる。本システムは自動化できている。

＜Testing and verification of DNNs＞
・これまでの機械学習システムの評価方法は、ランダムに選んだ入力で精度を測っていた
・本システムでは、手作業なしの全自動で、システマチックなテストができる

＜Other applications of DNN gradients＞
＜Differential testing of traditional software＞
・重要な特長は、ニューロン網羅率を最大化しながら、
　多くの差異をもたらす入力を発見する問題は、
　明確に定義されたジョイント最適化問題として表現されることである。

【10 CONCLUSION】

以上