

MODE: Automated Neural Network Model Debugging via State Differential Analysis and Input Selection

MODE: 状態差分分析と入力選択による ニューラルネットワークモデルの自動デバッグ

訓練データを賢く選択することでNNモデルの精度を上げる

著者

Shiqing Ma, Yingqi Liu, Wen-Chuan Lee, Xiangyu Zhang, Ananth Grama
Purdue University, USA

論文情報

ESEC/FSE '18, November 4-9, 2018, Lake Buena Vista, FL, USA

紹介者: 堀口日向

概要



ニューラルネットワーク（以降NN）で分類問題を解く
NNモデルが本来の性能を発揮できない



モデルに“**バグ**”がある。しかし、直接修正できない

既存のデバッグ手法

追加で訓練データを学習する

問題点

モデルの誤動作についての理解不足、精度向上に限界

提案手法（MODE）

モデル状態の差分分析で欠陥のあるニューロンを特定
欠陥を修正できるような訓練データを入力

背景 & 動機 (1 / 2)

ソフトウェアには必然的にバグがあり
デバッグは、ソフトウェア開発プロセスでの重要なステップ

AI/MLでも同様に、望ましくない動作（モデルバグ）をすることがある
モデルバグのデバッグは、AI開発プロセスでの重要なステップ

モデルの本来の性能（本来可能であるはずの精度）は不明だが
デバッグによって精度を向上させることができるのであれば
それはモデルバグがあるということになる

AIの使われどころによっては
モデルバグは壊滅的である可能性
（例：自動運転、ドローン）



背景 & 動機 (2 / 2)

NNモデルの精度低下の要因

- ・ モデル構造（隠れ層の数、各層のニューロンの数） ⇒ 構造的バグ
- ・ 誤った訓練プロセス（偏った訓練データの入力） ⇒ 訓練バグ

モデル構造に関する研究はたくさんある

本研究では訓練バグをデバッグする手法を提案する

以降では**訓練バグ**のことを**モデルバグ**と呼ぶ

モデルバグの分類

モデルバグには2つの分類があり、それぞれ検出・修正方法が若干異なる

未学習バグ

訓練時の精度が θ 以下かつ
テスト時の精度が θ 以下の
ラベル g が存在する

訓練時の精度も
テスト時の精度も
低い！

過学習バグ

テスト時の精度よりも
訓練時の精度の方が γ 高い
ラベル g が存在する

訓練時の精度よりも
テスト時の精度の方が
かなり低い！

提案手法 | 手順

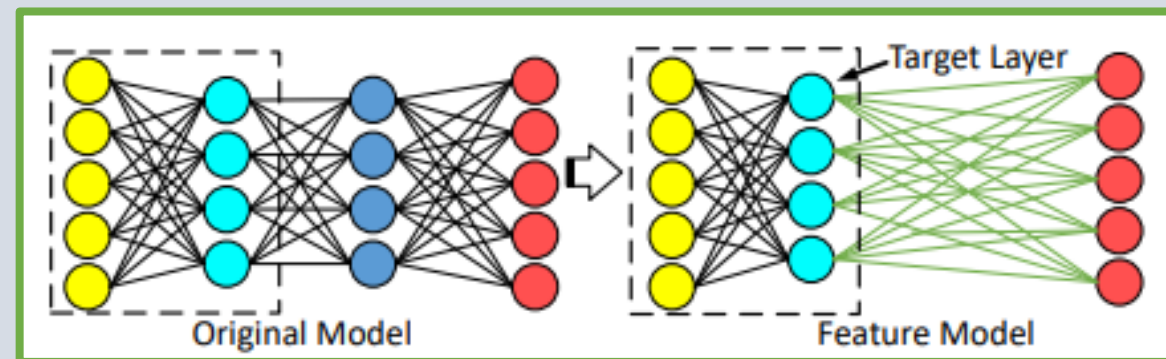
- ① ラベルの誤分類に最も影響が高い層（ターゲットレイヤ）を特定する
- ② ターゲットレイヤの出力を特徴として正しく分類できる入力の特徴と誤分類する入力の特徴の差分（差分ヒートマップ）を求める
- ③ 差分ヒートマップを使って、訓練データからデータを選択しモデルを再学習する

モデルの
状態差分分析

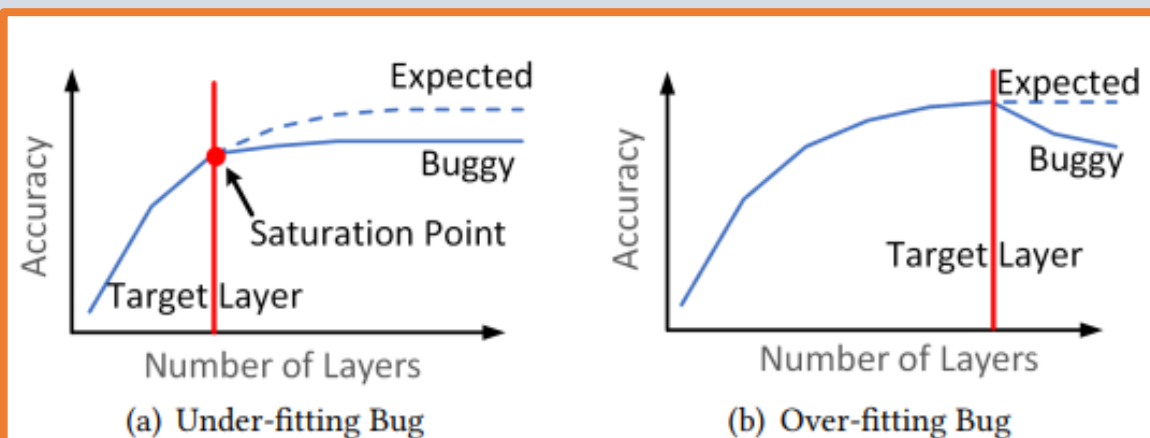
提案手法 | モデルの状態差分分析 (1/2)

① ラベルの誤分類に最も影響が高い層（ターゲットレイヤ）を特定する

未学習 過学習
入力層（出力層）側から*i*番目までの
中間層と出力層（入力層）を直接結合
訓練データセットを使って学習
→特徴モデル*i*
これを中間層の数分繰り返す



未学習 過学習
*i*と*i*+1 (*i*-1) の特徴モデルの
テスト精度の差分が一定以下なら
レイヤ*i*をターゲット層にする



提案手法 | モデルの状態差分分析 (2/2)

- ② ターゲットレイヤの出力を特徴として、正しく分類できる入力の特徴と誤分類する入力の特徴の差分（差分ヒートマップ）を求める

具体例：手書き数字の分類

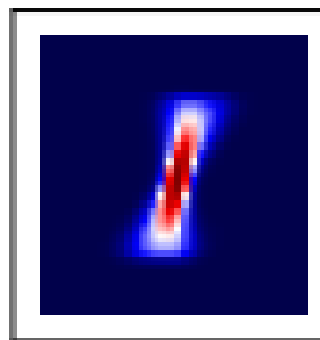
(a) 数字の1を入力したときの特徴

(b) 数字の2を入力したときの特徴

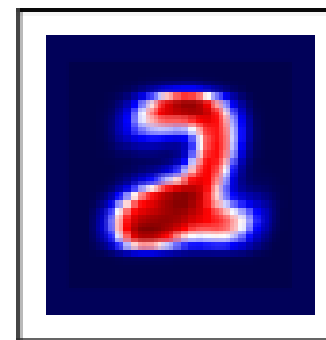
(c) 差分ヒートマップ

白：差分なし

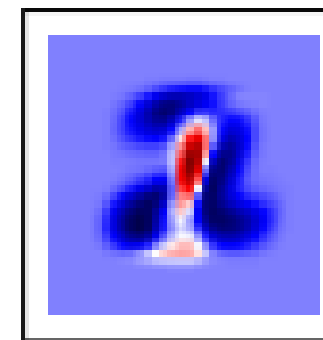
赤/青：正/負の差分あり



(a) H_1



(b) H_2



(c) $DHCI_{1,2}$

元のモデルでは(a)も(b)も1に分類されている

赤の部分の成分を持つ訓練データが欲しい！

青の部分の成分を持つ訓練データはいらない！

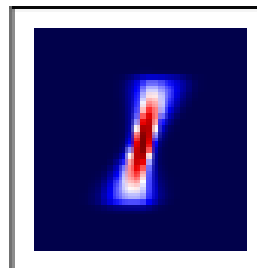
白の部分の成分を持つ訓練データはあってもなくてもいい！

提案手法 | 訓練データの選択

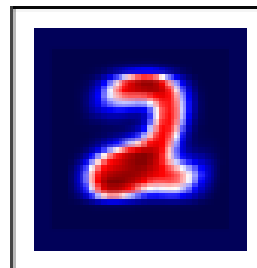
③差分ヒートマップを使って、訓練データからデータを選択し、モデルを再学習する

具体例：

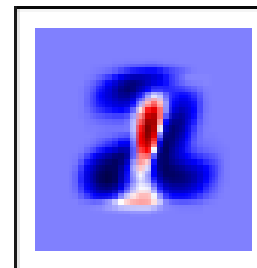
(d)と(e)どちらを選択するか
→ (d)を選択するべき
(e)のせいで精度が落ちる



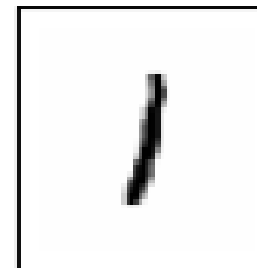
(a) H_1



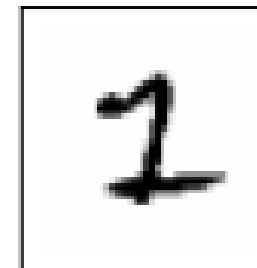
(b) H_2



(c) $DHCH_{1,2}$



(d) Good



(e) Bad

差分ヒートマップはバグ修正に対して
最も効果的な方向を持つベクトル

差分ヒートマップと入力データのcos類似度をとる
→ 類似度が高い入力データを訓練データとして選択

評価方法

以下の各データセットに対するTensorFlowのチュートリアルで実装されていたモデル（全20個）を使う



データセットを分割

トレーニングセット：30%

検証セット：10%

テストセット：10%

バグ修正セット：50%

トレーニングセットで学習した後
バグ修正セットを使ってバグ修正を試みる

3つのバグ修正方法で結果を比較

MODE, GANで生成したデータをランダム選択, 誤分類したデータを選択

結果

3つのバグ修正方法で結果を比較

Not Fixed	MODE	Random Selecting GAN	Failing
0.84	0.91	0.84	0.84

既存手法に比べ
12倍も速くバグを修正し
精度向上ができた！

既存手法は
どれだけ時間をかけても
効果なし！