修 士 論 文

題　目　　バージョン管理の支援を目的とした
差分の時系列変化可視化ツールDeepDiffViewer

主任指導教員　　水野　修　教授

指導教員　　崔　恩瀞　助教

京都工芸繊維大学大学院 工芸科学研究科

情報工学専攻

学生番号　　21622010

氏　　名　　大橋　幸奈

令和5年2月9日提出

様式 2 号

学位論文内容の要旨（和文）

令和 5 年 2 月 9 日

京都工芸繊維大学大学院
工芸科学研究科長　殿

工芸科学研究科　情報工学専攻
令和 3 年入学
学生番号　21622010
氏　名　大橋　幸奈　㊞

（主任指導教員　水野　修　㊞ ）

本学学位規則第 4 条に基づき、下記のとおり学位論文内容の要旨を提出いたします。

1. 論文題目

バージョン管理の支援を目的とした差分の時系列変化可視化ツール DeepDiffViewer

2. 論文内容の要旨（400 字程度）

深層学習モデルとは，人間の脳の構造に着目した多層パーセプトロンを用いたプロセスである．深層学習モデルは，多様なパターンを学習することができるため，自動運転や画像認識など様々な分野で応用されている．最近，深層学習モデルの性能を向上させるために行うハイパーパラメータのチューニングに関心が高まっている．そこで，本研究では，差分の時系列変化可視化ツール DeepDiffViewer を開発した．本研究では，まず，実際の開発者がハイパーパラメータのチューニングにおいてどんな課題に直面しているかを Stack Overflow に投稿された質問を通して調査した．その後，その調査結果に基づいてハイパーパラメータの値とモデル精度の関係などの実験結果をグラフで可視化することで記録し管理する DeepDiffViewer を開発した．DeepDiffViewer は特定プログラミング言語やフレームワークなどに依存しないため，モデル開発を終始サポートできる．また，Git と連携することで training anomaly を引き起こすコードへの変更を特定も支援できる．最後に，Stack Overflow の実際の質問をに基づいて DeepDiffViewer の 3 つの利用シナリオを議論した．

DeepDiffViewer: Tool for visualizing time-series changes in differences to support version control

2023 21622010 *OHASHI Yukina*

Abstract

Deep learning models are based on multilayer perceptrons and are designed to mimic the structure of the human brain. They have successfully applied in various fields, including autonomous driving and image recognition. Recently, there has been a growing interest in tuning hyperparameters to improve the performance of deep learning models. This paper presents DeepDiffViewer, a tool for visualizing time-series changes in different settings. Firstly, I investigated the questions posted on Stack Overflow to find out what challenges developers face in tuning hyperparameters. Based on the investigation results, I then developed DeepDiffViewer, which saves and manages the results of experiments, such as the relationship between hyperparameter values and model accuracy, by visualizing them in graph form. Because DeepDiffViewer does not depend on any specific programming language or framework, it can effectively support model development. In addition, DeepDiffViewer can help identify changes to the code that cause training anomalies by integrating with Git. Finally, the paper presents three real-world usage scenarios for DeepDiffViewer, based on questions asked on Stack Overflow.

目　次

1. 緒言

深層学習モデル（以降モデル）とは，人間の脳の構造に着目した多層パーセプトロンを用いたプロセスである [1]．モデルは，多様なパターンを学習することができるため，近年，自動運転 [2] や画像認識 [3] など様々な分野で応用されている．モデル開発プロセスは，データの収集，データの前処理，モデルの構築，学習，性能評価，運用など，6 つに分けられる．そのうち，モデルの構築，学習のステップでは誤差が最小になるようにモデルを学習させる．最近では，モデルの性能を向上させるために行うハイパーパラメータのチューニングに関心が高まっている [4]．ハイパーパラメータとは深層学習アルゴリズムの挙動を設定するパラメータ（活性化関数，隠れ層の数，隠れ層のユニット数活性化関数，ドロップアウトする割合，学習率，最適化関数，誤差関数，バッチサイズ，エポック数など）をさす．しかし，ハイパーパラメータをチューニングする際にはパラメータの値を変更し，それによって出力したメトリクスを確認した上で，必要に応じてパラメータの値を変更する作業が繰り返し行われる．そして開発者は，実験の再現性を確保するため，実験の試行錯誤のハイパーパラメータの条件と結果がどうだったのかを記録しておく必要がある．また，これらのハイパーパラメータの設定の誤りは，モデルの精度が極端に低い，または高い，損失値が下がらない，オーバーフィット（モデルが学習データに適合しすぎてしまうこと），反復間で精度値が不連続，損失値が不安定など，様々な学習異常の原因となり得るが，これらの学習異常は，モデルの学習プログラムの実行を止めないし，エラー表示もない．従って開発者は，損失値や精度などの値の変化を観察し続ける必要がある．本研究ではこの問題における解決策として，差分の時系列変化可視化ツール DeepDiffViewer を開発した．本研究では，まず，実際の開発者たちがハイパーパラメータのチューニングにおいてどんな課題を抱えているのかを Stack Overflow に投稿された質問を通して調査した．そして調査の結果に基づきながら開発者たちのモデル開発におけるハイパーパラメータのチューニングを支援するツール DeepDiffViewer を開発した．モデルのハイパーパラメータの値やメトリクスの値を管理し，見やすくグラフで可視化する．またある時点のモデルに至るまでの履歴を it へのコミット情報などを含めてフロー図として表示し，モデルの改善を支援する．

最後に本論文の以降の構成を示す．まず，第 2 章で本研究の必要となる背景知識について言及しながら，関連研究を紹介する．第 3 章では RQ に関する調査方法と調査結果，そしてどんなツールが必要とされているのかを議論する．第 4 章では本研究で開発した DeepDiffViewer の仕様を述べるため，可視化の対象，フロー，利用シナリオなどを追って説明する．第 5 章で本研究の結言を述べる．

2. 背景

2.1 深層学習モデル開発プロセス

モデルとは，人間の脳の構造に着目した多層パーセプトロンを用いたプロセスである [1]．モデルは，多様なパターンを学習することができるため，近年，自動運転 [2] や画像認識 [3] など様々な分野で応用されている．また，モデルの実装は，Pytorch (注 1)，Keras (注 2)，tensorflow (注 3)など様々なライブラリのおかげで，多くの開発者が比較的簡単に手を出すことができる [5].

図 2.1 は深層学習モデル開発プロセス [4] の 6 つのステップを示す.

ステップ 1: **データの収集**

このステップでは，モデルで使用するデータセットを収集し，収集されたデータセットの統計量が妥当かどうかを検証するプロセスである．深層学習では，データセットのパターンから学習し，その結果を汎化しようとする．つまりデータセットが深層学習ワークフローの中心であり，その品質が深層学習プロジェクト成功の礎になる．また，学習時に使用されたデータセットがその後変更されたかどうかを確認することが必要になる．モデル開発プロセスでは，生成されたモデルを学習時のデータセットと紐づけて追跡するためバージョン管理する必要がある．

ステップ 2: **データの前処理**

このステップでは，収集したデータセットをモデルで学習できるように数値表現へ変換する．モデルの学習に使用するデータは，多くの場合，モデルに入力できない形式で与えられる．例えばモデルの学習に使用したい特徴は Yes と No のタグとして与えられるが，モデルにはこれらを数値表現（1 や 0 など）として当てる必要がある．テキストであれば，単語をインデックスや単語ベクトルに変換できる [6]．また，学習するデータに人工的にさまざまな変換を加えデータを増やすこともある [7]．この時，ステップ 1 で作成したデータセットの特徴に関する統計量を考慮し，前処理の形式はどうあるべきかを決定する．

(注 1)：https://pytorch.org/
(注 2)：https://www.tensorflow.org/guide/keras?hl=ja
(注 3)：https://www.tensorflow.org/?hl=ja

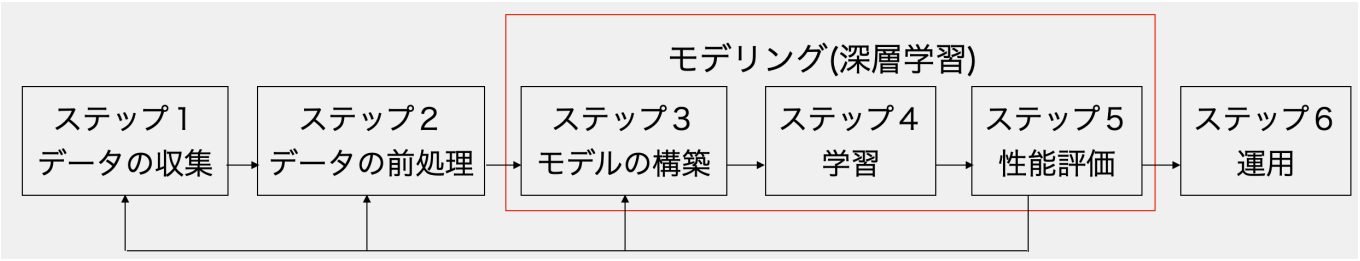


図 2.1　深層学習モデルの適用プロセス

また，前処理ステップの変更と前処理済みデータが対応付けし管理することが大切で，前処理が変更された場合にはモデル開発プロセス全体を再度実行する必要がある．

ステップ 3: **モデルの構築**

このステップでは，データセットの特徴を捉えるためのモデルの構築をする．学習を行うモデルにどんなモデルを使用するかで大枠が決まるため，開発したい問題を踏まえて，データセットにあったモデルや選択することが重要になる [8].

ステップ 4: **学習**

この学習ステップでは，誤差が最小になるようモデルを学習する．モデルや学習データセットが大規模になる程，学習ステップの管理は難しくなる．一般的に使えるメモリは有限なので，学習は分散して効率的に行うことが重要になる．また，ハイパーパラメータの設定は．予測性能の高いモデルを実現する上で非常な重要な要素である [9][10]．初期段階の実験中に行うこともあれば，パイプラインに組み込むこともある．設定するハイパーパラメータの中でも，最適化とネットワークアーキテクチャを制御するタイプのパラメータが重要になる．より小さなモデルでは，過学習を避けるために，早期停止（early stopping）を使用し，指定したエポック数を経過しても検証データの損失値が改善しない場合，モデルの学習を停止する必要がある [11][12]．学習時の性能は向上するが過学習の原因となる上，モデルの学習時間が長くなるため，層数と層のサイズはネットワークアーキテクチャを設定する上で最も重要なパラメータである．

ハイパーパラメータ探索のアプローチとしては，グリッドサーチとランダムサーチがよく使われている [13]．グリッドサーチは，パラメータのすべての組み合わせを網羅的に試行する．一方，ランダムサーチでは，パラメータはあり得る選択肢からサンプリングされため，すべての組み合わせを試行するとは限らない．グリッドサーチは，ハイパーパラメータの組み合わせが多い場合，非常に時間がかかってしまう．様々な値を試した後，最も性能の良いハイパーパラメータを取り出して，それらを中心に新しい探索を開始することで，微調整できる．

ステップ 5: 性能評価

このステップでは，モデルの性能を詳細に分析する．一般的に，最適なモデルパラメータは正解率か損失値を用いて決定する．ただし，モデルの最終版が決まったらモデル性能の分析をより深く行うことで，モデルの改善を行うことができる．分析には，適合率，再現率，AUC（曲線下面積）などの評価指標の計算や，学習中に用いる検証データセットより大規模なデータセットを用いた性能の測定が含まれる．また，モデルの予測が公平であることを確認する必要がある．データセットをスライスし，スライスしたデータセットごとに性能を計測しない限り，多様なユーザーグループに対してモデルがどのような予測するか分からない．さらに，学習時に使用した特徴に対するモデルの依存性も調査し，ある学習データの特徴を変更した場合に，モデルの予測がどのように変化するかを調べる必要がある．

ステップ 6: 運用

このステップでは，学習済みのモデルを本番環境にデプロイして運用する．新しくデプロイされたモデルの有効性と性能を測定するため，モデルのフィードバックが必要である [14]．また，必要な場合，新しい学習データをキャプチャしてデータセットを増やし，モデルを更新することもある．モデルサーバーを使用すると，再デプロイすることなくモデルバージョンを更新させることができる．パイプラインを自動化することで，データサイエンティストは，現行のモデルの更新や保守ではなく，新しいモデルの開発に集中することができる [15][16].

2.2 深層学習モデルの開発支援ツール

モデルのメトリクスなどをプロットし可視化するツール TensorBoard (注 4) や Visdom (注 5) は，モデルを学習するプロセスの全体を把握しやすくすることで管理することができる．しかし，これらのツールはただメトリクスをプロットするため，開発者が学習データの設定の誤りや，学習異常が起きた時にその原因となった変更コード

(注 4)：https://www.tensorflow.org/tensorboard?hl=en
(注 5)：https://github.com/fossasia/visdom

がどこなのかという追跡が困難である．

DVC（Data Version Control）[注 6]はは学習や評価の試行の繰り返しの中で大量に生み出されるデータのバージョン管理を支援するツールである．DVC は，Git と連携することで，データの格納場所として各種ストレージサービスが利用でき、大量の画像などの大きなサイズのデータを扱うことができる，また，機械学習パイプライン処理の記述と実行を行い，再現性を確保する．しかし，DVC の操作は基本的に Git ライクなコマンドラインインタフェースで行い，GUI 画面はないため操作が難しい．また，多くの Github プロジェクトは，DVC の機械学習パイプラインはとても複雑のため，まだ DVC を模索し，実験している段階である [17]．従って，深層学習パイプラインをサブコンポーネントに分解することを考慮することが必要である．

MLflow [注 7]は深層学習の開発を行う上で複雑になりがちな実行環境，モデル，パラメータ，評価指標などの実験管理を支援するツールである．MLflow は，実験管理というモデル学習・評価・スコアリングの実行記録を行うことで，過去の実行結果の参照や再現が可能になる．また，標準で管理 UI があり，Web ブラウザで実験管理の結果を視覚的に確認できる．そしてレポジトリでのバージョン管理と、ラベルによる運用対象バージョン管理が可能である．しかし，MLflow API を使用して実行記録を行うため，学習や評価の Python コードを MLflow に適用するため書き換える必要がある．

著者が開発した DeepDiffViewer[18] は，モデルにおける差分の時系列変化可視化ツールである．開発者が，ハイパーパラメータの設定を行うとき，モデルの学習や精度評価など何度も条件を変えて繰り返す試行を記録する必要がある．ハイパーパラメータの値とモデル精度の関係などの実験結果のグラフで可視化することで記録し，管理する際に DeepDiffViewer を用いると実験結果を比較をしやすくすることができる．まずツールを立ち上げると，精度とコミット日時を軸として，コミットされたファイルをグラフ上に表示する．次に，プロットされたノードをクリックするとフロー図が表示される．モデル学習や精度評価など何度も条件を変えて繰り返す試行を記録し，条件とモデル精度の関係などの実験結果の比較をしやすくすることができる．

（注 6）: https://dvc.org
（注 7）: https://mlflow.org

2.3 Stack Overflow

Stack Overflow(以降 SO)[注 8]はプログラミングに関する題材を扱うオンライン Q&A サイトである．開発者はソフトウェア開発中に起こる問題を，SO へ質問として投稿し，他の開発者の回答により解決することができる．SO では，質問者は投稿する質問にタグ付けをすることにより，質問をカテゴリに分類でき，解答できるエキスパートと質問を結びつける機能や，関係のある質問を簡単に検索できる機能がある．SO は他のオンライン Q&A サイトより質問の量が圧倒的の多く，解答の質も高いため多くの開発者に利用されている．

図 2.2 は，実際に SO に投稿された質問の例を示す．[注 9] この質問の質問者は，xgboost というアルゴリズムを使用していて，モデルがオーバーフィットを起こしている理由や解決策を探している．質問には，[python],[machine-learning],[classifiction],[xgboost],[hyperparameters] というタグが付けられている．

(注 8)：https://stackoverflow.com
(注 9)：https://stackoverflow.com/questions/62332861/how-to-deal-with-overfitting-of-xgboost-classifier

How to deal with overfitting of xgboost classifier?

Ask Question

Asked 2 years, 7 months ago Modified 9 months ago Viewed 1k times

0

I use xgboost to do a multi-class classification of spectrogram images(data link: automotive target classification). The class number is 5, training data includes 20000 samples(each class 5000 samples), test data includes 5000 samples(each class 1000 samples), the original image size is 144*400. This is my code snippet:

```
train_data, train_label, test_data, test_label = load_data(data_dir, resampleX=4, resamp
scaler = StandardScaler()
train_data = scaler.fit_transform(train_data)
test_data = scaler.transform(test_data)
cv_params = {'n_estimators': [100,200,300,400,500], 'learning_rate': [0.01, 0.1]}
other_params = {'learning_rate': 0.1,  'n_estimators': 100,
                'max_depth': 5, 'min_child_weight': 1, 'seed': 27, 'nthread': 6,
                'subsample': 0.8, 'colsample_bytree': 0.8, 'gamma': 0,
                'reg_alpha': 0, 'reg_lambda': 1,
                'objective': 'multi:softmax', 'num_class': 5}
model = XGBClassifier(**other_params)
classifier = GridSearchCV(estimator=model, param_grid=cv_params, cv=3, verbose=1, n_jobs
classifier.fit(train_data, train_label)
print("The best parameters are %s with a score of %0.2f" % (classifier.best_params_, cla
```

During hyperparameter tunning, according to https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/, I tuned `n_estimators` at first with `GridSearchCV(n_estimators=[100,200,300,400,500])` using training data, then test with test data. Then I tried GridSearchCV with both 'n_estimators' and 'learning_rate' also.

The best hyperparameter is `n_estimators=500` + 'learning_rate=0.1' with best_score_=0.83, when I use this best estimator to classify, the training data I get 100% correct result, but the test data only gets precison of `[0.864 0.777 0.895 0.856 0.882]` and recall of `[0.941 0.919 0.764 0.874 0.753]`. I guess with `n_estimators=500` is overfitting, but I don't know how to choose this n_estimator and learning_rate at this step.

For reducing dimensionality, I tried PCA but more than n_components>3500 is needed to achieve 95% variance, so I use downsampling instead as shown in code.

Sorry for the incomplete info, hope this time is clear. Many thanks!

python machine-learning classification xgboost hyperparameters

図 2.2　Stack Overflow の質問例

3. 予備調査

本章では，ツール開発のための予備調査の目的，リサーチクエスチョン (以降 RQ) を示し，次に扱うデータセット，調査方法，調査結果について説明する．

3.1 調査目的

本研究では，2.2 節で説明した DeepDiffViewer を改善するため，実際の開発者の深層学習におけるハイパーパラメータに関する共通の課題を明らかにする予備調査を行なっている．

この予備調査では，以下の RQ を調査している．

- RQ: 深層学習のハイパーパラメータについて，SO でどんな質問が多いのか？
  この質問は，モデルを含むソフトウェア開発のハイパーパラメータ調整プロセスにおいて，どんな課題があるのかを調査することを目的としている．

3.2 調査方法

3.2.1 データの前処理

3.1 節で説明した RQ に答えるために，データセットとして SO の質問を集めた．この調査では 2023 年 1 月に取得した SO の質問のうち，[hyperparameters] でタグ付きされた 784 件の質問を抽出した．

まず，Stackexchange API (注 10) を使って,SO から抽出した質問に付けられたタグ 3,290 件を取得した．タグのデータの前処理として，全ての質問に共通して付けられている [hyperparameters] タグを削除した．深層学習とほぼ同じ意味であり，調査結果に影響しないと考えられる [machine-learning]，[deep-learning]，[neural-network]，[conv-neural-network] のタグも削除した．また，[python] とほぼ同じ意味であり，まとめても調査結果に影響しないと考えられる [python-2.7]，[python-3.x] のタグは [python] タグに置き換えた．以上の前処理を行なった結果，タグ 2,127 件を集計する．

(注 10)：https://api.stackexchange.com/

3.2.2 アソシエーション分析

[hyperparameters] タグと関連の深いタグを見つけるため，前処理を行ったタグを昇順に並べ，度数分布表とした．また，アソシエーション分析を用いて，より深く，タグ同士の関係を調査する．アソシエーション分析とは，主にマーケティングの分野におよって活用されていて，購買データを基に消費者の購買行動の中にある関連性を見つけ出す分析手法である．本研究では，タグにアソシエーション分析を適応し，どのタグのペアに相関関係があるのかを調査する．支持度 (support) は，どのタグに対してどのタグがどのくらい一緒に質問に付けられているかという指標であり，全体の中から両方のタグがまとめて付けられている確率を指す．支持度 (support)，信頼度 (confidence)，リフト値 (lift) を算出した数式を以下に示す．

$$support = \frac{antecedents\ タグと\ consequents\ タグが両方ついた質問数}{全質問数}$$

$$confidence = \frac{antecedents\ タグと\ consequents\ タグが両方ついた質問数}{antecedents\ タグがついた質問数}$$

$$lift = \frac{confidence}{consequents\ タグがついた質問数}$$

3.2.3 質問内容の調査

モデルがパターンを見つけられるようにする学習アソシエーション分析によって，[hyperparameters] タグと関連性が高いタグのペアがついた質問が明らかになった．その中でも主要な質問を抽出するため，アソシエーション分析によって抽出されたルール 80 個のうち Support（支持度）の高い上位 25% である 20 個のペアを抽出する．また，[scikit-learn],[keras],[tensorflow] のように，フレームワークやツールに関しての質問につくタグのルールを除く，その他の関連性が高いタグのペアがついた質問について，内容を調査し分類した．Zhang と Gao らが特定した，Stackoverflow の質問のカテゴリに基づき，質問の内容を分類した [19]．以下で各カテゴリについて説明する．

implementation この実装カテゴリの質問は，質問者が希望する機能の実装方法．または API の使用方法に関するものである．また，質問者は異なるタスクや異なるデータセットに対して希望するモデルの実装をどのように適応させるかについて質問している．

program crash このカテゴリは，プログラムを停止させるものに関しての質問である．モデルでは，層の間の多次元配列の入力と出力の不一致によってプログラム実行時にエラーを生じる可能性がある．

training anomaly このカテゴリは，モデルの学習時，信頼性のないメトリクスが出力されるものに関しての質問である．モデルの学習は，基本的に損失関数によって測定した過去の予測誤差に基づいてハイパーパラメータを継続的に調整する．これらの設定の誤りは，学習時の異常の原因となる．

performance このカテゴリは，プラットフォームやフレームワーク，GPU の違いによる性能差についての質問である．モデルの学習にかかる時間や，GPU の使用率などを改善する質問である．

comprehension このカテゴリの質問は，概念やアルゴリズム，フレームワークなどの知識の理解に関するものである．

3.3 結果

本章では，前の章の方法によって得られた RQ の分析結果を示す．

図 3.1 は，SO から抽出した質問のタグに前処理を行い，昇順に並べた度数分布を示している．

上位には，[python],[r] という言語や，[keras],[tensorflow] というフレームワーク，[scikit-learn] というツールなどがある．よって，開発者はさまざまな言語やフレームワーク，ツールなどを使って，ハイパーパラメータの調整を行なっていて，それぞれに課題を抱えていることがわかった．したがって，全ての開発者を支援するツールは，言語やフレームワーク，他のツールなどから独立し，使用できることが必要である．

表 3.1 は，アソシエーション分析の結果，抽出した 20 個のルールである．主要な質問を抽出するため，80 個のルールのうち支持度 0.03 以上で上位 25%である 20 個のルールを抽出し，それ以下は削除した．

[python][scikit-learn],[keras],[tensorflow] などのペアは，一般的なフレームワークやツールに関しての質問であると考えられる．本研究ではパラメータの最適化などに関係のあるタグだけに着目した．ハイパーパラメータをチューニングする上での

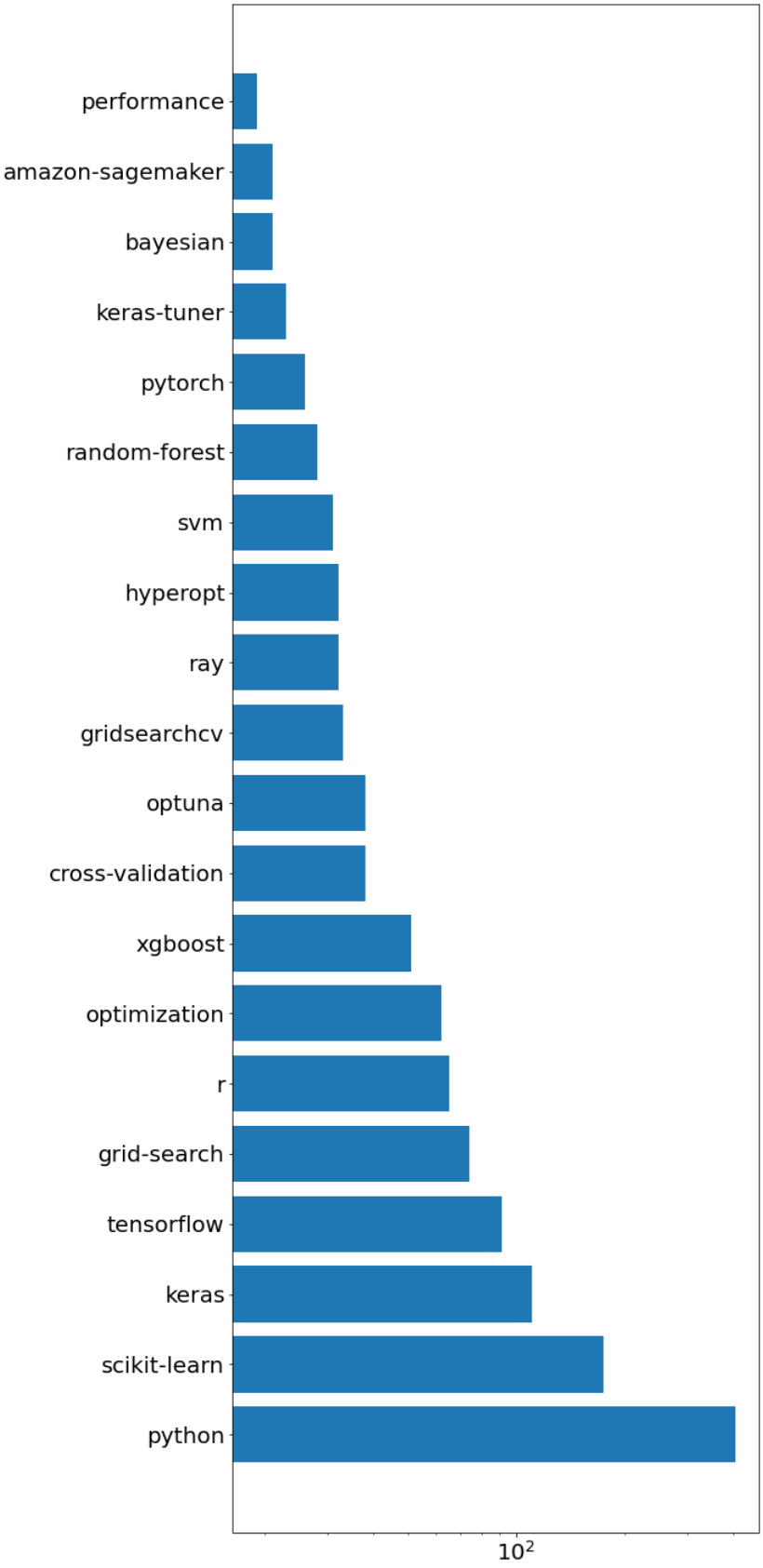


図 3.1　質問のタグ度数分布

表 3.1　アソシエーションルール

	antecedents	consequents	support	confidence	lift
0	python	scikit-learn	0.1637055	0.3200992	1.4413612
1	scikit-learn	python	0.1637055	0.7371428	1.4413612
2	keras	python	0.0761421	0.5405405	1.0569378
3	python	keras	0.0761421	0.1488833	1.0569378
4	python	tensorflow	0.0609137	0.1191067	1.0201747
5	tensorflow	python	0.0609137	0.5217391	1.0201747
6	grid-search	python	0.0532994	0.5675675	1.1097847
7	python	grid-search	0.0532994	0.1042183	1.1097847
8	keras	tensorflow	0.0532994	0.3783783	3.2408930
9	tensorflow	keras	0.0532994	0.4565217	3.2408930
10	optimization	python	0.0368020	0.4603174	0.9000748
11	xgboost	python	0.0342639	0.5192307	1.0152700
12	hyperopt	python	0.0329949	0.7647058	1.4952561
13	grid-search	scikit-learn	0.0329949	0.3513513	1.5820849
14	scikit-learn	grid-search	0.0329949	0.1485714	1.5820849
15	optuna	python	0.0329949	0.7027027	1.3740191
16	keras,python	tensorflow	0.0329949	0.4333333	3.7115942
17	keras,tensorflow	python	0.0329949	0.6190476	1.2104454
18	python,tensorflow	keras	0.0329949	0.5416666	3.8453453
19	keras	python,tensorflow	0.0329949	0.2342342	3.8453453
20	tensorflow	keras,python	0.0329949	0.2826086	3.7115942

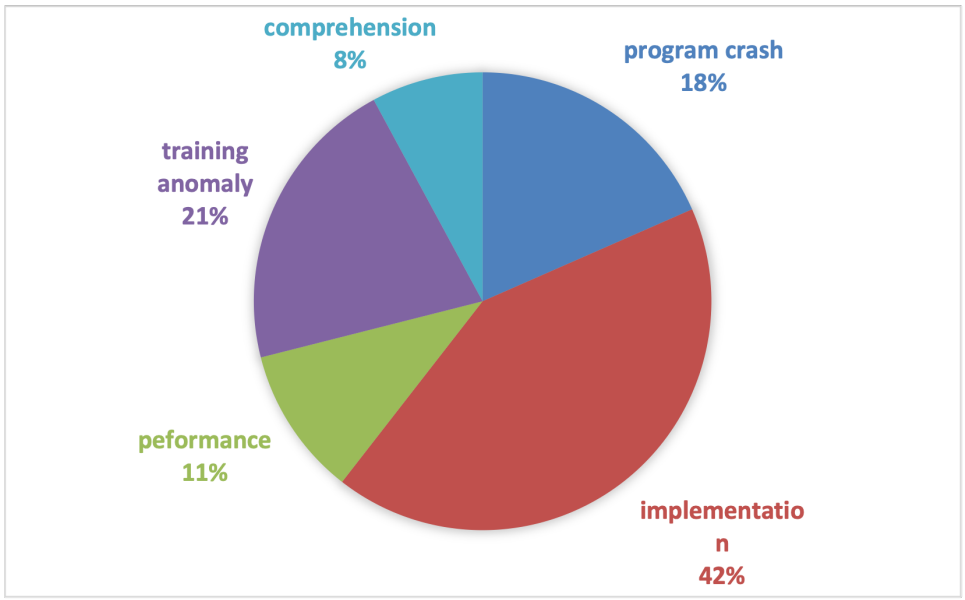


図 3.2　[python][grid-search] タグの質問カテゴリ分け

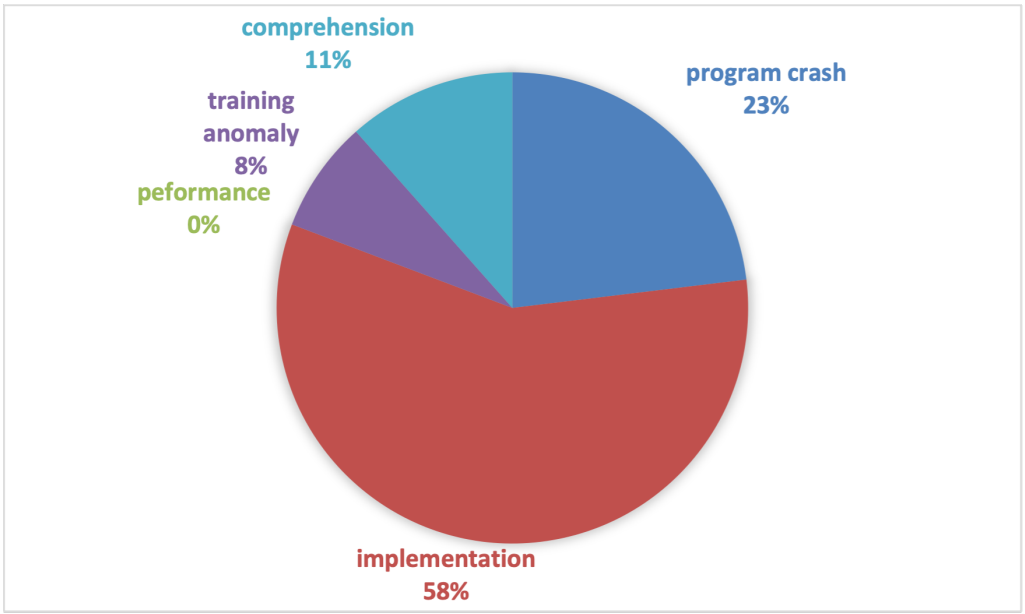


図 3.3　[optimization][python] タグの質問カテゴリ分け

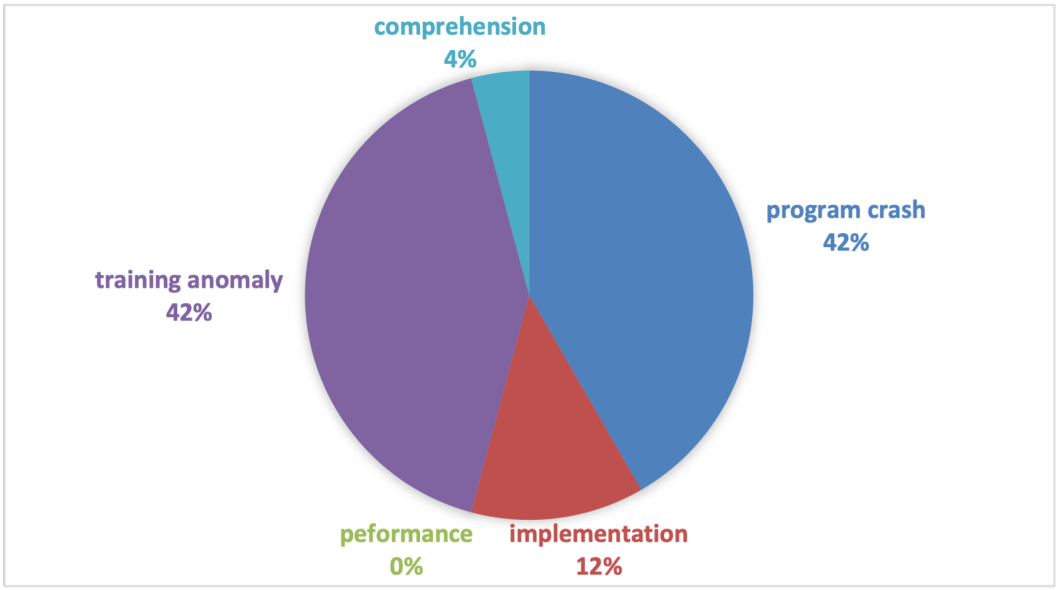


図 3.4　[xgboost][grid-python] タグの質問カテゴリ分け

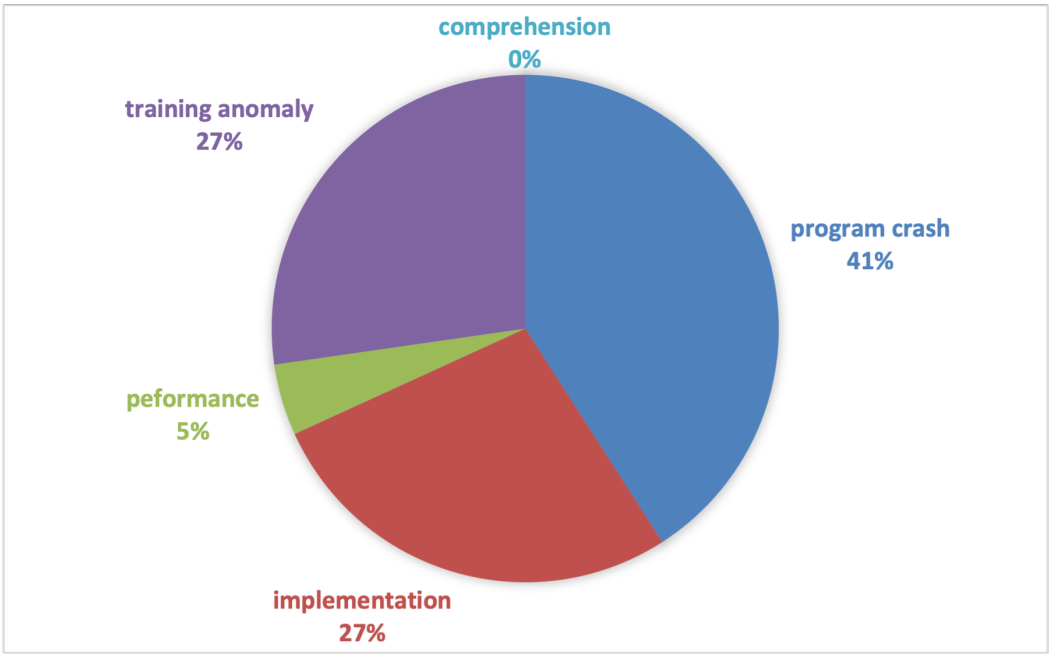


図 3.5　[hyperopt][python] タグの質問カテゴリ分け

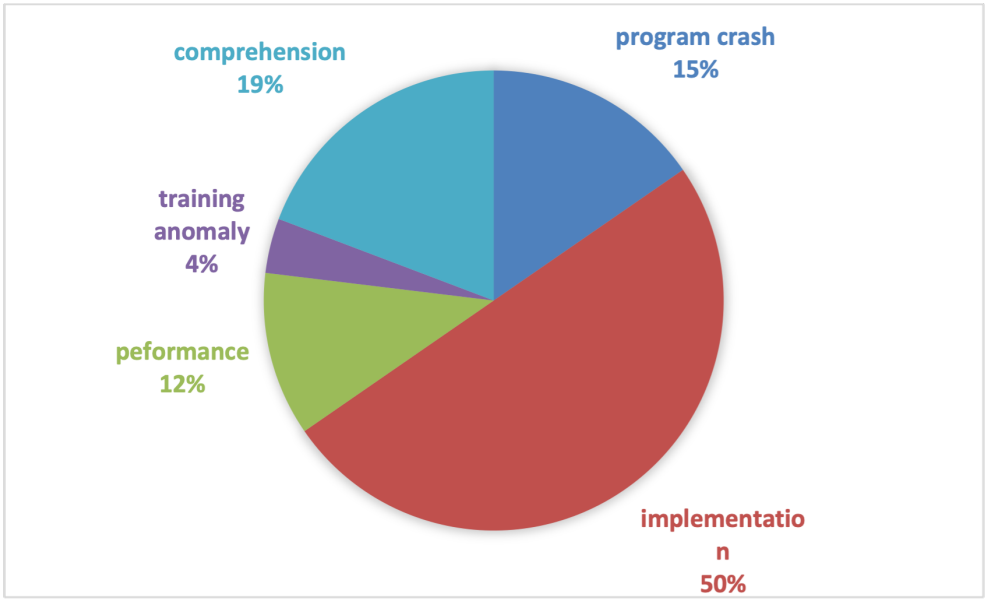


図 3.6　[grid-search][scikit-learn] タグの質問カテゴリ分け

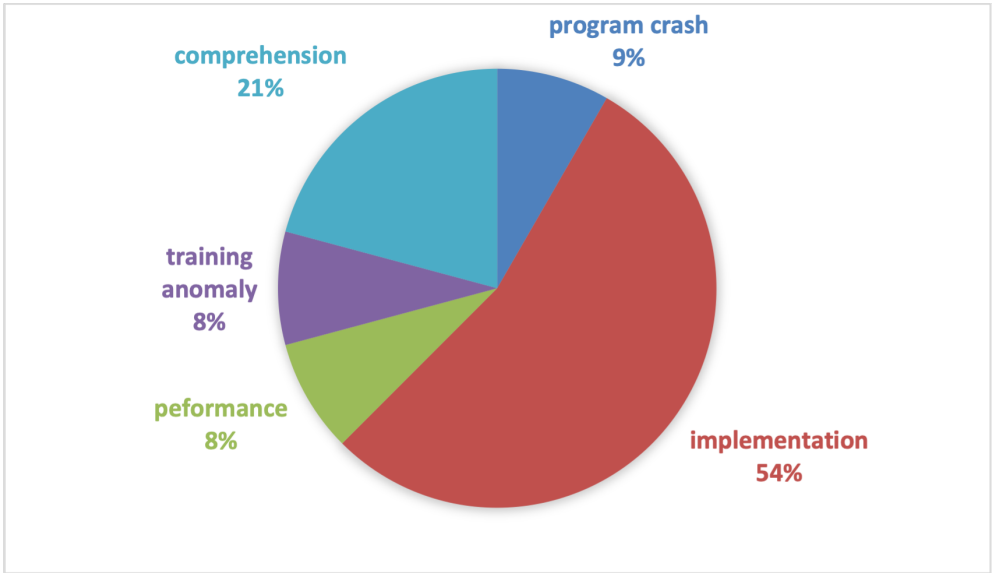


図 3.7　[optuna][python] タグの質問カテゴリ分け

課題に関する質問を抽出するため，[python][scikit-learn],[keras],[tensorflow] のペアを除く．

[python][grid-search],[optimization][python],[xgboost][python], [hyperopt][python],[grid-search][scikit-learn],[optuna][python] の 6 つのペアのタグがついた質問について内容を調査し分類した．(図 3.2 図 3.3 図 3.4 図 3.5 図 3.6 図 3.7)

支持度が大きいほどよく起こる事象を表すため，[python][grid-search] の意味は python に対する質問は，grid-search に対しての質問にもなりやすい．図 3.2 から分かるように，質問の 42%が implementation に関するものである．同様に，[optiomization][python], [grid-search][scikit-learn], [optuna][python] は全て implementation に関する割合が一番高い．これらの時には implementation に困っている可能性が高い．グリッドサーチを使用する際，ハイパーパラメータと検証する範囲を事前に設定する必要がある．また，optuna はハイパーパラメータを自動最適化してくれるツールですが，同じように検証する範囲を事前に設定する必要あり，開発者は深層学習コードを書き換え繰り返し実行している．

図 3.5 図 3.6 から分かるように，[xboost][python], [hyperopt][python] は program crash や training anomaly に関する質問が多い．ハイパーパラメータをチューニングする際に，誤った値を設定すると，program crash や training anomaly が起きる可能性がある．これらの学習データの設定の誤りは，モデルの精度が極端に低い，または高い，損失値が下がらない，モデルが学習データに適合しすぎてしまうオーバーフィット，反復間で精度値が不連続，損失値が不安定など，様々な training anomaly の原因となり得る．これらの training anomaly は，program crash よりも悪質で，モデルの学習プログラムの実行を止めない上に，エラー表示もない．よって開発者は，損失値や精度などの値の変化を観察し続ける必要がある．しかし，大規模なシステムでは，パラメータ設定のコード行数は，実際に深層学習を行うコード行数より多い場合があり，これらを修正することは膨大な時間やコストがかかる可能性がある [20].

図 3.8 は，SO に投稿された質問例である．(注 11)．この図から分かるように，この質問の質問者は，hyperopt を使用して XGBoost のハイパーパラメータをチューニン

(注 11)：https://stackoverflow.com/questions/69521240/different-result-metric-from-evaluation-and-prediction-with-hyperopt

Different result metric from evaluation and prediction with hyperopt

Ask Question

Asked 1 year, 3 months ago Modified 1 year, 3 months ago Viewed 573 times

4

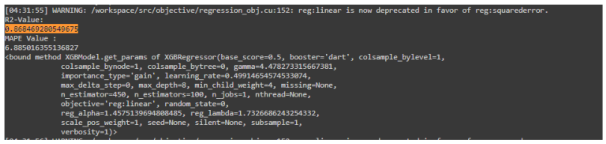
This is my first experience with tuning XGBoost's hyperparameter. My plan is finding the optimal hyperparameter by using hyperopt.

```
def obj (params):
  xgb_model=xgb.XGBRegressor(
      n_estimator=params['n_estimator'],
      learning_rate=params['learning_rate'],
      booster=params['booster'],
      gamma=params['gamma'],
      max_depth=int(params['max_depth']),
      min_child_weight=int(params['min_child_weight']),
      colsample_bytree=int(params['colsample_bytree']),
      reg_lambda=params['reg_lambda'],reg_alpha=params['reg_al
  )
  evaluation=[(X_train,Y_train),(X_test,Y_test)]
  xgb_model.fit(X_train, Y_train,
            eval_set=evaluation,
            verbose=False)
  pred = xgb_model.predict(X_test)
  r2_value=r2_score(y_true=Y_test,y_pred=pred)
  mape=MAPE(pred,Y_test)
  print('R2-Value:',r2_value)
  print('MAPE Value :',mape)
  print(xgb_model.get_params)
  return {'loss': -r2_value, 'status': STATUS_OK ,'model':xgb_

params={'n_estimator':450,
        'learning_rate':hp.loguniform('learning_rate',np.log(0
        'booster':hp.choice('booster',['gbtree','dart','gblinea
        'reg_lambda':hp.uniform('reg_lambda',0,2.5),
        'reg_alpha':hp.uniform('reg_alpha',0,2.5),
        'colsample_bytree':hp.uniform('colsample_bytree',0,1),
        'gamma':hp.uniform('gamma',0,10),
        'max_depth':hp.quniform('max_depth',3,10,1),
        'min_child_weight':hp.quniform('min_child_weight',0,10,

trials = Trials()
best_hyperparams = fmin(fn = obj,
```

I display loss value based on the R2 Score and MAPE. I caught the best loss value after running the code.



When I use that hyperparameter, I got different MAPE and R2 results than before.

```
model=xgb.XGBRegressor(base_score=0.5, booster='gbtree', colsar
              colsample_bynode=1, colsample_bytree=0, gamma=4.47
              importance_type='gain', learning_rate=0.499146545
              max_delta_step=0, max_depth=8, min_child_weight=4,
              n_estimator=450, n_estimators=100, n_jobs=1, nthre
              objective='reg:linear', random_state=0,
              reg_alpha=1.4575139694808485, reg_lambda=1.732668
              scale_pos_weight=1, seed=None, silent=None, subsar
              verbosity=1)

model.fit(X_train,Y_train)
model.predict(X_test)
```

```
R2 Score: 0.45169292085598933
MAPE:  16.35871224743466 %
```

Can you give me some explanation, why could it happen?

python machine-learning xgboost hyperparameters hyperopt

The Overflow Blog

- Comparing tag trends with our Most Loved programming languages
- The less JavaScript, the better (Ep. 532)

Featured on Meta

- Accessibility Update: Colors
- Introducing a new close reason specifically for non-English questions
- We're bringing advertisements for technology courses to Stack Overflow
- 2022: a year in moderation
- Temporary policy: ChatGPT is banned

Linked

3 Does XGBoost produce the same results if I use different number of cores?

Related

1668 How can I make a dictionary (dict) from separate lists of keys and values?

1297 Why does comparing strings using either '==' or 'is' sometimes produce a different result?

1573 Use different Python version with virtualenv

1681 Random string generation with upper case letters and digits

2317 Importing files from different folder

886 How are iloc and loc different?

1 How to use XGBoost softprob multi class classification such that I do not get the error for num_class?

1 HyperOpt multi metric evalution

Hot Network Questions

- I want to nerf my character for roleplay purposes, but I fear I might have the "My Guy Syndrome"
- MySQL 8 - Access denied when dropping procedures
- Got accepted to top-choice PhD program. Drop other interviews?
- Is there a standardized way to classify languages according to how much the order of the words is tied to the words themselves?
- What ways would you recommend for paying for the transportation services in Hong Kong?
- How flat is water?
- Did Apple drop/ reduce support for Chromecast?
- How to achieve a 200 Nm tightening torque when installing a freewheel body?

図 3.8　Stack Overflow に投稿された質問例

グしている．複数回学習を実行しているとR2とMAPEなどのメトリクスの値が低い原因を探している．回答ではランダムシードなどを修正する方法が提案されている．このように，開発者はモデルの学習コードを修正し，ハイパーパラメータを何度も変更している．また変更ごとにメトリクスの値を確認しながらモデルの改善をおこなっている．

3.4 議論

3.4.1 妥当性への脅威

(1) 内部妥当性

予備調査では，Stack Overflowに投稿された質問に対して目視で内容を確認し，各カテゴリに分類した．しかし，分類したカテゴリのセットは先行研究において2人の著者がStack Overflowに投稿された715個の質問を精察したものであり，前例にならって分類を行なっている．

また，Stack Overflowに投稿された質問に，回答者は質問に対してのコメントで質問者の疑問に答えているケースや，質問者もまた質問に対してのコメントや，回答に対してのコメントなどで回答者の回答を指示している場合がある．本研究ではこれらを目視で確認し，質問者の一番役に立った回答を選択して考察をしている．

(2) 外部妥当性

予備調査では，Stack Overflowに投稿された質問のうち特定のタグについての質問のみについて調査している．よって他の質問についての調査を含まない．また，Stack Overflowに投稿された質問のみを調査しているため，他のソースで他の開発者についての調査を含まない．

3.4.2 考察

実際の開発者のハイパーパラメータのチューニングに関する課題，またその開発環境を調査した結果，どのような支援が必要かを議論する．

SO の質問には [python],[r] という言語や，[keras],[tensorflow] というフレームワーク，[scikit-learn] というツールなどのタグが多くつけられていることから，開発者はさまざまな言語やフレームワーク，ツールなどを使って，ハイパーパラメータの調整を行なっていて，それぞれに課題を抱えていることがわかった．従って，深層学習の言語やフレームワーク，ツールなどに依存せず，独立した支援が必要がことがわかった．

また，ハイパーパラメータのチューニング中 training anomaly などの問題が起こることから，モデルの差分とその精度などの分かりやすい可視化や，どのコードへの変更が training anomaly を引き起こすのか特定できるようなツールが必要となる．どの文，操作，ハイパーパラメータが異常を引き起こすかをピンポイントで特定することができれば，これらを修正する上で時間やコストを減らすことが期待できる．

4. DeepDiffViewer

本章では，予備調査の結果に基づいて改善したモデルにおける差分の時系列変化可視化ツールである DeepDiffViewer を説明する．

4.1 目的

第 2 章で述べたように，モデルの学習では，基本的に損失関数や評価指標に基づいてモデルのパラメータを継続的に調整するプロセスとなるため，頻繁に膨大な量の学習データと、ミニバッチや特徴スケーリングなどのパラメータと精度の組み合わせの多くのデータを扱う．DeepDiifViewer の目的は，このデータを可視化することでデータ管理の支援することである．3.4 節で説明した通り，深層学習のプログラミング言語やフレームワーク，ツールに依存していないツールであり，training anomaly を引き起こすコードへの変更を特定を支援するツールが必要である．従って，DeepDiffViewer は，プログラミング言語やフレームワーク，他のツールから独立していて，training anomaly が起きた時にその原因となった変更コードがどこなのかという追跡するため Git と連携している．

4.2 DeepDiffViewer

DeepDiffViewer の仕様について述べる．3.4 節で議論した内容に基づいて改善した点を以下に説明する．主な改善点は，表示するグラフの自由度が上がったことである．これまでツールが表示できるグラフは，メトリクスが時間ごとにどう改善したかを示す，x 軸をコミット日時，y 軸をメトリクスの値とするグラフだけでなく，メトリクスの値，ハイバーパラメータの値，コミット日時などの中から，x 軸 y 軸を開発者が自由に選択しグラフを表示することができる．それによって，例えば x 軸をコミット日時，y 軸をハイパーパラメータの値にして，開発者がハイパーパラメータを時間ごとにどう変更したかを知れたり，x 軸をハイパーパラメータの値，y 軸をメトリクスの値にすることによって二つの値の相関関係などを知ることができる．

また，ローカルで Git の管理下にあるプロジェクトだけでなく，GitHub の URL を入力することによってリポジトリをクローンしツールで可視化することができるよ

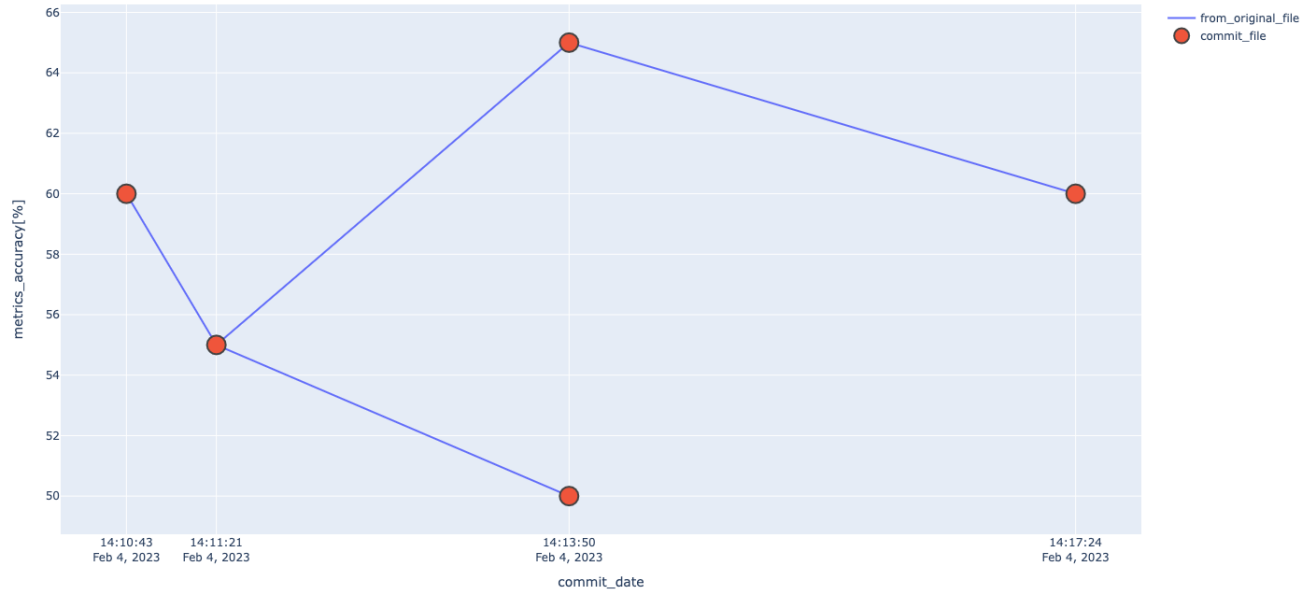


図 4.1　精度と時系列グラフの例


Change Log
File_name cifar10_width.py
author_date 2023-02-04 14:17:24+09:00
author_name yukinaohashi
command metrics accuracy 60% in cifar10_width.py change param width_shift_range 0.3 to 0.4 from cifar10_width.py to cifar10_width.py
committer_date 2023-02-04 14:23:52+09:00
committer_name yukinaohashi
diff file:////Users/yukinaohashi/Desktop/cifar10_/diff_ddf73bce20f92cc031d4ad951a57ddb29fd504b7.txt
metrics accuracy
metrics_value 60%
original_file cifar10_width.py
parameter width_shift_range
parameter_change 0.3 to 0.4
commit diff diff.txt
^
File_name cifar10_width.py
author_date 2023-02-04 14:13:50+09:00
author_name yukinaohashi
command metrics accuracy 50% in cifar10.py change param height_shift_range 0.2 to 0.3 from cifar10.py to cifar10.py
metrics accuracy 65% in cifar10_width.py change param width_shift_range 0.2 to 0.3 from cifar10.py to cifar10_width.py
committer_date 2023-02-04 14:13:50+09:00
committer_name yukinaohashi
diff file:////Users/yukinaohashi/Desktop/cifar10_/diff_1e09eff3433e51b0d031009e2c7ea1afd54861b2.txt
metrics accuracy
metrics_value 65%
original_file cifar10.py
parameter width_shift_range
parameter_change 0.2 to 0.3
commit hash diff.txt
^
File_name cifar10.py
author_date 2023-02-04 14:11:21+09:00
author_name yukinaohashi
command metrics accuracy 55% in cifar10.py change param height_shift_range 0.1 to 0.2 from cifar10.py to cifar10.py
committer_date 2023-02-04 14:11:21+09:00
committer_name yukinaohashi
diff file:////Users/yukinaohashi/Desktop/cifar10_/diff_5cadad318a4bec19a3c8e246b629f47801dee173.txt
metrics accuracy
metrics_value 55%
original_file cifar10.py
parameter height_shift_range
parameter_change 0.1 to 0.2
commit hash diff.txt
^
File_name cifar10.py
author_date 2023-02-04 14:10:43+09:00
author_name yukinaohashi
command metrics accuracy 60% in cifar10.py
committer_date 2023-02-04 14:10:43+09:00
committer_name yukinaohashi
diff file:////Users/yukinaohashi/Desktop/cifar10_/diff_72196185644f7b7c9479b25c107b1caf7f8810ea.txt
metrics accuracy
metrics_value 60%
original_file
parameter
parameter_change
commit hash diff.txt

図 4.2　フロー図の例

うになった．これによって複数の開発者で開発されているプロジェクトのデータ管理がより容易になる．

また，フロー図から，コミット間のコードの変更箇所を表示する．これによってメトリクスの値に異常が起きた時，その原因となり得るコード変更の箇所を確認することができる．以下では DeepDiffViewer の使用を説明する．

4.2.1 可視化の対象

本研究では，モデルにおける差分の時系列変化可視化ツールを開発する．ここでの差分を取るデータとして以下を定義する．

1. モデル（例：RNN(Recurrent Neural Network) や CNN (Convolutional Neural Network)）の構造
2. データの前処理部分におけるパラメータ
3. ハイパーパラメータ（活性化関数，隠れ層の数，隠れ層のユニット数活性化関数，ドロップアウトする割合，学習率，最適化関数，誤差関数，バッチサイズ，エポック数）

以上のモデルとその学習に関するデータのコミット間の差を本研究の差分とする．

4.2.2 可視化までのフロー

モデルにおける差分の時系列変化可視化ツールを利用するまでのフローを説明する．

1. 開発者が，差分を取るデータを指定する．
2. git commit 時に，開発者が差分と結果の精度部分を取り出しコミットメッセージにコマンドを書き込む．
3. Git のローカルリポジトリの内容をリモートリポジトリに送信する．
4. ツールを立ち上げ Git の管理下にあるプロジェクト，または GitHub の URL を入力する．
5. ツール内で表示したいグラフの軸を選択後，グラフが可視化される．

コミットメッセージにコマンドが書かれていない場合には，そのコミットは無視されるので，コミットごとに学習をし精度を出す必要はない．

4.3 利用シナリオ

利用シナリオ 1

図 4.3 は，SO に投稿された質問例である．(注 12) この図から分かるように，この質問の質問者は，Keras で作成した MLP のモデルを，RandomizedSearchCV を使ってパラメータチューニングを行なっている．RandomizedSearchCV とは，パラメータの設定範囲や試行回数を指定し，指定値範囲内から無作為に抽出したパラメータにより学習を試行することにより最適なパラメータを走査するランダムサーチを行うツールである．質問者は，適合度（precision）や再現度（recall）などのメトリクスが良くない値である原因を探している．回答では，モデルのオーバーフィットを疑い，早期終了 (early stopping) や，ドロップアウト (dropout) を試す提案が投稿された．質問者は，回答に従いモデルのコードに，早期終了とドロップアウトのパラメータをそれぞれ追加し，メトリクスの値の変化を観察することになる．

その際，質問者は追加したコード，それに伴い出力されたメトリクスの変化を管理し，どのコードの変化がメトリクスの値に影響を与えたのかを記録する必要がある．

ここで DeepDiffViewer を用いると，メトリクスの値をプロットしその変化をわかりやすく可視化することができる．また，DeepDiffViewer は Git と連携していてコミット情報に基づき値をプロットしているため，値の変化の原因がどのコードの追加によるものなのかを確認することができる．従って，DeepDiffViewer を用いると，質問者は逐一，コードの変更点とそれによるメトリクスの値の変化を確認し，より良いモデルの改良を効率よく進めることができる．

さらに，もし質問者が他の開発者にモデルの開発を引き継ぐことになった場合や，モデルの運用者への引き継ぎが必要になった場合，DeepDiffViewer の可視

(注 12)：https://stackoverflow.com/questions/55666937/hyperparameter-tuning-in-keras-mlp-via-randomizedsearchcv

Hyperparameter tuning in Keras (MLP) via RandomizedSearchCV

Ask Question

Asked 3 years, 9 months ago Modified 3 years, 9 months ago Viewed 2k times

1

I have been trying to tune a neural net for some time now but unfortunately, I cannot get a good performance out of it. I have a time-series dataset and I am using RandomizedSearchCV for binary classification. My code is below. Any suggestions or help will be appreciated. One thing is that I am still trying to figure out how to incorporate is early stopping.

EDIT: Forgot to add that I am measuring the performance based on F1-macro metric and I cannot get a scoring higher that 0.68. Another thing that I noticed is that the more parameters I try to estimate at once (increase my grid), the worse my scoring is.

```
train_size = int(0.70*X.shape[0])
X_train, X_test, y_train, y_test = X[0:train_size], X[train_si

from numpy.random import seed
seed(3)
from tensorflow import set_random_seed
set_random_seed(4)

from imblearn.pipeline import Pipeline

def create_model(activation_1='relu', activation_2='relu',
                 neurons_input = 1, neurons_hidden_1=1,
                 optimizer='adam',
                 input_shape=(X_train.shape[1],)):

  model = Sequential()
  model.add(Dense(neurons_input, activation=activation_1, input

  model.add(Dense(neurons_hidden_1, activation=activation_2, k

  model.add(Dense(2, activation='sigmoid'))

  model.compile (loss = 'sparse_categorical_crossentropy', opti
  return model


clf=KerasClassifier(build_fn=create_model, verbose=0)

param_grid = {
    'clf__neurons_input':[5, 10, 15, 20, 25, 30, 35],
    'clf__neurons_hidden_1':[5, 10, 15, 20, 25, 30, 35],
    'clf__optimizer': ['Adam', 'Adamax','Adadelta'],
    'clf__activation_1': ['softmax', 'softplus', 'softsign',
```

My scores

```
[Parallel(n_jobs=1)]: Done  50 out of  50 | elapsed:  1.9min finished
Best: 0.639051 using {'clf__activation_1': 'softmax', 'clf__activation_2': 'softsign',
'clf__batch_size': 80, 'clf__learning_rate': 0.49986097171525545,
'clf__neurons_hidden_1': 20, 'clf__neurons_input': 20}
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     15587
           1       0.24      0.79      0.37       181

   micro avg       0.97      0.97      0.97     15768
   macro avg       0.62      0.88      0.68     15768
weighted avg       0.99      0.97      0.98     15768

[[15134   453]
 [   38   143]]
Testing: 0.676059780224415
```

python keras time-series grid-search hyperparameters

Share Improve this question Follow

edited Apr 14, 2019 at 17:39

The Overflow Blog

- Comparing tag trends with our Most Loved programming languages
- The less JavaScript, the better (Ep. 532)

Featured on Meta

- Accessibility Update: Colors
- Introducing a new close reason specifically for non-English questions
- We're bringing advertisements for technology courses to Stack Overflow
- 2022: a year in moderation
- Temporary policy: ChatGPT is banned

Linked

3 Grid search and KerasClassifier using class weights

-1 How to Customize Metric for GridSearchCV in Scikit Learn to tune for specific class?

Related

401 Understanding Keras LSTMs

6 Create model using one - hot encoding in Keras

8 'Sequential' object has no attribute 'loss' - When I used GridSearchCV to tuning my Keras model

4 LightGBM hyperparameter tuning RandomizedSearchCV

0 Approximating a smooth multidimensional function using Keras to an error of 1e-4

Hot Network Questions

- What ways would you recommend for paying for the transportation services in Hong Kong?
- Command-line Tower of Hanoi game
- Why did merely Canada retain the rank of Brigadier General? But not Australia, New Zealand, or UK?
- Cell-based vs face-based finite element methods
- Do courts generally run at a loss, run at a profit, or generally break even?
- Film or series where bacteria were the protagonists
- History of High Availability in the mainframe and minicomputer eras?
- Multiple alignments of equations

図 4.3　Stack Overflow に投稿された質問例

化されたグラフによって，これまでのモデルの変更を一覧することができる．

利用シナリオ２

2.3 節で説明した図 2.2 は，SO に投稿された質問例である．この図から分かるように，この質問の質問者は，xgboost というアルゴリズムを使用していて，モデルがオーバーフィットを起こしている理由や解決策を探している．その際，ハイパーパラメータチューニングの GridSearchCV を利用している．xgboost という勾配ブースティング回帰木では，浅い決定木を複数作成し，ブースティングを行うことで性能を向上させる．性能の高さや使い勝手の良さの一方で，XGBoost は多くのハイパーパラメータを持つため，その性能を十分に発揮するためにはパラメータチューニングが重要となる [21][22]．特にオーバーフィットは最も陥りやすい学習異常であり，質問者もその対処に困っている．ここでは回答者は，ハイパーパラメータチューニングの Optuna を使い，Early stopping rounds パラメータを追加することを提案している．質問者は，回答に従いモデルのコードに，Oputuna を利用し Early stopping ounds パラメータを追加し，メトリクスの値の変化を観察することになる．ここで DeepDiffViewer を用いると，追加したハイパーパラメータによるメトリクスの変化を可視化することができる．このように，質問者は自身のモデルやそのアルゴリズムの状況によって，GridSearchCV や Oputuna など様々なツールを試すことがある．その際に，DeepDiffViewer はプログラミング言語やフレームワーク，ハイパーパラメータのチューニングツールなどに依存せず独立したものであるため，質問者のモデル開発を終始サポートすることができる．

利用シナリオ３

3.3 節で説明した図 3.8 は，SO に投稿された質問例である．この図から分かるように，この質問の質問者は，hyperopt を使用して XGBoost のハイパーパラメータをチューニングしている．複数回学習を実行していると R2 と MAPE などのメトリクスの値が低い原因を探している．回答ではランダムシードなどを修正する方法が提案されている．このように，質問者は様々なメトリクスの値を個別に確認し，ハイパーパラメータとの関係を確認する必要がある．ここで DeepDiffViewer を用いると，メトリクスの値や，ハイパーパラメータの値，コ

ミットした時間などからグラフ軸を自由に選択できるため，一回の実験に対して詳しく分析することができる．

5. 結言

本研究では，ハイパーパラメータの値とモデル精度の関係などの実験結果をグラフで可視化することで記録し管理する DeepDiffViewer を開発した．予備調査では，SO に投稿された質問を調査し，実際の開発者の深層学習におけるハイパーパラメータに関する共通の課題を明らかにした．また，予備調査の結果に基づき DeepDiffViewer を開発した．DeepDiffViewer は，プログラミング言語やフレームワーク，ハイパーパラメータのチューニングツールなどに依存せず独立したものであるため，開発者のモデル開発を終始サポートすることができる．また，Git と連携することで training anomaly が起きた時にその原因となった変更コードがどこなのかという追跡を支援する．最後に，Stack Overflow の実際の質問をに基づいて DeepDiffViewer の 3 つの利用シナリオを議論した．

謝辞

本研究を行うにあたり，研究課題の設定や研究に対する姿勢，本報告書の作成に至るまで，全ての面で丁寧なご指導を頂きました，本学情報工学・水野修教授，崔恩瀞助教，立命館大学 情報理工学部 吉田則裕教授，九州大学 大学院システム情報科学研究院 近藤 将成助教に厚く御礼申し上げます．本報告書執筆にあたり貴重な助言を多数頂きました，本学情報工学専攻　ソフトウェア工学研究室の皆さん，学生生活を通じて著者の支えとなった家族や友人に深く感謝致します．

参考文献

[1] X.-W. Chen and X. Lin, “Big data deep learning: Challenges and perspectives,” IEEE Access, vol.2, pp.514–525, 2014.

[2] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A.Y. Ng, “An empirical evaluation of deep learning on highway driving,” 2015.

[3] H. Greenspan, B. vanGinneken, and R.M. Summers, “Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique,” IEEE Transactions on Medical Imaging, vol.35, no.5, pp.1153–1159, 2016.

[4] C.N. Hannes Hapke，入門　機械学習パイプライン，中山光樹（編），（株）オライリー・ジャパン，東京，2021.

[5] X. Tan, K. Gao, M. Zhou, and L. Zhang, “An exploratory study of deep learning supply chain,” Proceedings of the 44th International Conference on Software Engineering, p.86–98, ICSE ’22, Association for Computing Machinery, New York, NY, USA, 2022.

[6] U. The Robotics Institute, Carnegie Mellon University, “Text sentiment analysis based on depth learning model,” 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB), pp.1–3, 2018.

[7] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” 2017.

[8] N. Li, O. Kähler, and N. Pfeifer, “A comparison of deep learning methods for airborne lidar point clouds classification,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol.14, pp.6467–6486, 2021.

[9] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, and M. Young, “Machine learning: The high interest credit card of technical debt,” SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop), pp.1–9, 2014.

[10] S.R. Young, D.C. Rose, T.P. Karnowski, S.-H. Lim, and R.M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, pp.1–5, MLHPC '15, Association for Computing Machinery, New York, NY, USA, 2015.

[11] I. Bilbao and J. Bilbao, "Overfitting problem and the over-training in the era of data: Particularly for artificial neural networks," 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), pp.173–177, 2017.

[12] H. Li, J. Li, X. Guan, B. Liang, Y. Lai, and X. Luo, "Research on overfitting of deep learning," 2019 15th International Conference on Computational Intelligence and Security (CIS), pp.78–81, 2019.

[13] L. Dong, H. Du, F. Mao, N. Han, X. Li, G. Zhou, D. Zhu, J. Zheng, M. Zhang, L. Xing, and T. Liu, "Very high resolution remote sensing imagery classification using a fusion of random forest and deep learning technique—subtropical area for example," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol.13, pp.113–128, 2020.

[14] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, "Software engineering for machine learning: A case study," 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), pp.291–300, 2019.

[15] M. Kim, T. Zimmermann, R. DeLine, and A. Begel, "The emerging role of data scientists on software development teams," Proceedings of the 38th International Conference on Software Engineering, p.96–107, ICSE '16, Association for Computing Machinery, New York, NY, USA, 2016.

[16] H. Liu, S. Eksmo, J. Risberg, and R. Hebig, "Emerging and changing tasks in the development process for machine learning systems," Proceedings of the International Conference on Software and System Processes, p.125–134, ICSSP '20, Association for Computing Machinery, New York, NY, USA, 2020.

[17] A. Barrak, E.E. Eghan, and B. Adams, “On the co-evolution of ml pipelines and source code - empirical study of dvc projects,” 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), pp.422–433, 2021.

[18] 大橋幸奈，崔恩瀞，吉田則裕，近藤将成，水野修，深層学習モデルにおける差分の時系列変化可視化ツール，京都工芸繊維大学情報工学課程卒業論文（未刊行），京都工芸繊維大学，2020.

[19] T. Zhang, C. Gao, L. Ma, M. Lyu, and M. Kim, “An empirical study of common challenges in developing deep learning applications,” 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), pp.104–115, 2019.

[20] M. Dilhara, A. Ketkar, N. Sannidhi, and D. Dig, “Discovering repetitive code changes in python ml systems,” 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE), pp.736–748, 2022.

[21] L. Sun, “Application and improvement of xgboost algorithm based on multiple parameter optimization strategy,” 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), pp.1822–1825, 2020.

[22] M.S. Oughali, M. Bahloul, and S.A. El Rahman, “Analysis of nba players and shot prediction using random forest and xgboost models,” 2019 International Conference on Computer and Information Sciences (ICCIS), pp.1–5, 2019.