WP-GEB-Estimator ユーザーマニュアル

～ WP-GEB : Weight-Perturbed Generalization Error Bounds ～

磯部祥尚
産業技術総合研究所
y-isobe@aist.go.jp

2024 年 4 月 30 日

目次

1　サマリ

ツール名 WP-GEB-Estimator の “WP-GEB” は**重み摂動付加汎化誤差上界**（Weight-Perturbated Generalization Error Bounds）を表しており、WP-GEB-Estimator は順伝播型の**ニューラル分類器**[*1]（以下，**分類器**とよぶ）の**無作為重み摂動**付加汎化誤差と**最悪重み摂動**付加汎化誤差の上界を見積もるツールである。重み摂動付加汎化誤差とは、任意の入力に対する推論中に、指定範囲内の任意の摂動をニューロン間の重みに付加した場合の不正解率の期待値である。**無作為重み摂動**は一様分布から無作為に選択した摂動であり、**最悪重み摂動**は可能な限り不正解になるように選択した摂動である。指定範囲内では最悪摂動でも不正解になるとは限らないが、実際に不正解になる摂動を**敵対的摂動**とよぶ。

機械学習の本格的な社会実装に向けて、機械学習を利用したシステムの品質に関する**機械学習品質マネジメント（MLQM）ガイドライン** [1] が策定されいる。このガイドラインには、機械学習要素の内部品質特性の一つとして、**機械学習モデルの安定性**「データセット以外の未見の入力に対しても安定して推論できること」が定義されている。図 1 は MLQM ガイドライン [1] の図 14「各フェーズとレベルに対する安定性の評価・向上技術」である。WP-GEB-Estimator は以下の機能を有してり、評価フェーズでの訓練済み分類器の安定性評価に有用なツールである。

- テストデータセットと無作為重み摂動サンプルに対する分類器の不正解率を計測できる（**ノイズ耐性**）
- テストデータセットの各入力に対する敵対的重み摂動を探索できる（**敵対的攻撃**）
- 任意の入力に対する敵対的重み摂動の存在確率上界をある信頼度で確率的に保証できる（**敵対的検証**）
- 任意の入力に対する重み摂動付加汎化誤差上界をある信頼度で確率的に保証できる（**汎化誤差**）

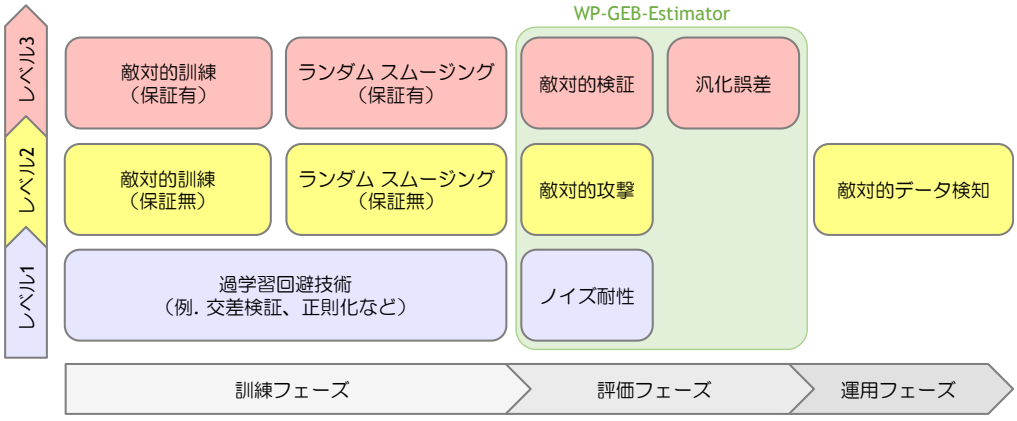

図 1　安定性の評価・向上に関する技術（MLQM ガイドライン [1] の図 14 ）

[*1] 分類目的で訓練したニューラルネットワーク

2　WP-GEB の紹介

本節では、2.1 小節で WP-GEB（重み摂動付加汎化誤差上界）の定義を与え、2.2 小節で WP-GEB の見積法について述べる。

2.1　WP-GEB の定義

無作為重み摂動付加汎化誤差 $\mathbf{R}^{\alpha}(f_w)$ は任意のデータ $(x,y) \sim \mathcal{D}$ に対する分類器 f_w の重み摂動付加個別誤差 $\mathbf{r}^{\alpha}_{(x,y)}(f_w)$ の期待値である（w は重みパラメータ、$\mathcal{D}$ は入力 x と正解出力 y の組 (x,y) の分布）。

$$\mathbf{R}^{\alpha}(f_w) := \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\mathbf{r}^{\alpha}_{(x,y)}(f_w)\right]$$

ここで、**重み摂動付加個別誤差** $\mathbf{r}^{\alpha}_{(x,y)}(f_w)$ は、任意の摂動 $u \sim \mathcal{U}_{w,\alpha}$ を重み w に付加したときの、データの一組 (x,y) に対する不正解率の期待値であり、重み摂動の分布 $\mathcal{U}_{w,\alpha}$ は重み摂動集合 $U_{w,\alpha}$ から無作為に一つの重み摂動を選択するための多次元一様分布である。

$$\mathbf{r}^{\alpha}_{(x,y)}(f_w) := \mathop{\mathbb{E}}_{u\sim\mathcal{U}_{w,\alpha}}\left[\ell(f_{w+u}(x),y)\right]$$
$$U_{w,\alpha} := \{(u_1,\ldots,u_{|w|}) \mid \forall i.\ |u_i| \le \alpha|w_i|\}$$

ここで、$\ell(y,y')$ は $y=y'$ ならば 0、$y\neq y'$ ならば 1 を返す 0-1 損失関数（i.e. $\ell(y,y') := \mathbb{1}[y\neq y']$）である。重み摂動集合 $U_{w,\alpha}$ は、各重み w_i に付加される重み摂動 u_i の絶対値が $\alpha|w_i|$ 以下であることを表している（α は**重み摂動幅対重み比**）。

最悪重み摂動付加汎化誤差 $\mathbf{W}^{\alpha}_{\theta}(f_w)$ は、任意のデータに対して重み摂動付加個別誤差 $\mathbf{r}^{\alpha}_{(x,y)}(f_w)$ が閾値以上となる 0/1 真偽値（閾値より大きいならば 1）の期待値である。

$$\mathbf{W}^{\alpha}_{\theta}(f_w) := \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\mathbb{1}\left[\mathbf{r}^{\alpha}_{(x,y)}(f_w) > \theta_{(x,y)}\right]\right]$$
$$\Theta := \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\theta_{(x,y)}\right]$$

ここで、$\theta_{(x,y)}$ はデータ (x,y) ごとに設定可能な閾値であり、Θ はその期待値である。この閾値は敵対的重み摂動の存在を許容できる割合である。摂動の影響を全く受けないことを保証するための厳しい閾値はゼロ（$\Theta=0$）であるが、現実的には計算コスト等の観点から、閾値は適切な微小な閾値に設定することが妥当である。妥当な閾値設定法（固定閾値、適応閾値）については次の 2.2 小節で説明する。

一般に、データ (x,y) と重み摂動 u は無数に存在するため、重み摂動付加汎化誤差 GE を正確に計算することは難しい。そこで、WP-GEB-Estimator では、任意の確率 $\delta\in(0,1)$ について、信頼度 $(1-\delta)$ 以上で次の不等式が成り立つような、重み摂動付加汎化誤差 GE の上界 B を見積もる。

$$\mathbb{P}[GE \le B] \ge 1-\delta$$

2.2　WP-GEB の見積法

無作為重み摂動付加汎化誤差 $\mathbf{R}^{\alpha}(f_w)$ の上界の見積法については多くの研究がある。例えば、Pérez-Ortiz 等 [4] は、有限なデータセットと有限な重み摂動サンプルに対する分類器の正解/不正解の計測結果から、

Maurer の定理 [3] とサンプル収束上界の定理 [2] によって、無作為重み摂動付加汎化誤差上界の見積法を与えた。WP-GEB-Estimator は、訓練データセットの代わりにテストデータセットを用いるが、基本的には Pérez-Ortiz 等 [4] の無作為重み摂動付加汎化誤差上界見積法を実装している。

一方、最悪重み摂動付加汎化差誤 $\mathbf{W}^{\alpha}_{\theta}(f_w)$ の見積法についての研究は少ない。その見積法は閾値 $\theta_{(x,y)}$ に依存しており、実用的で合理的な閾値 $\theta_{(x,y)}$ の決め方として、以下に説明する固定閾値と適応閾値がある [5]。**固定閾値**では、データ非依存の閾値 $\theta^{fix}(m,n,\delta_0)$ を次のように定義する[*2]。

$$\theta^{fix}(m,n,\delta_0) := 1 - \left(\frac{\delta_0}{2n}\right)^{1/m}$$

ここで、確率 $\delta_0 \in (0,\delta)$ は、任意の重み摂動の代わりにサイズ m の無作為重み摂動サンプルを用いることによって生じる信頼性低下の許容上限であり、n はテストデータセット T のサイズである。一般には、δ の半分程度が妥当である（$\delta_0 = 0.5 \times \delta$）。**適応閾値**では、データに依存して閾値 $\theta^{adapt}_{(x,y)}(m,n,\delta_0)$ を次のように定義する。

$$\theta^{adapt}_{(x,y)}(m,n_0,\delta_0) := \begin{cases} 0 & \text{if } (x,y) \in T_1 \\ \theta^{fix}(m,n_0,\delta_0) & \text{otherwise} \end{cases}$$

ここで、T_1 は探索や無作為サンプルで敵対的な重み摂動の存在を確認できたデータの集合（テストデータセット T の部分集合）、n_0 は $(T - T_1)$ のサイズである。WP-GEB-Estimator は、固定閾値 $\theta^{fix}(m,n,\delta_0)$ の最悪重み摂動付加汎化誤差 $\mathbf{W}^{\alpha}_{\theta^{fix}}(f_w)$ の上界と適応閾値 $\theta^{adapt}_{(x,y)}(m,n,\delta_0)$ の最悪重み摂動付加汎化誤差 $\mathbf{W}^{\alpha}_{\theta^{adapt}}(f_w)$ の上界を見積もるために、文献 [5] の見積式を実装している。

3 WP-GEB-Estimator の紹介

本節では、3.1 小節で WP-GEB-Estimator を構成する 4 つのツールを概説し、3.2 小節と 3.3 小節で WP-GEB-Estimator の実行に必要な入力ファイルと実行時に生成される出力ファイルについて説明する。

3.1 ツール構成

WP-GEB-Estimator は評価用の分類器の訓練から評価までを行う次の 4 つのツールから構成されている。このツールは Python 言語の Tensorflow/Keras を用いて記述されている。

- `train`: 評価用分類器訓練ツール
  - 入力：訓練データセット、ネットワークアーキテクチャ（CSV 形式）等
  - 出力：訓練済み分類器（TensorFlow-SavedModel 形式）
- `measure`: 訓練済み分類器の無作為重み摂動付加テスト誤差計測ツール
  - 入力：テストデータセット、訓練済み分類器、重み摂動対重み比リスト等
  - 出力：無作為重み摂動付加誤差計測結果（入力情報含む、CSV 形式）
- `search`: 訓練済み分類器の敵対的重み摂動探索ツール
  - 入力：テストデータセット、訓練済み分類器、無作為重み摂動付加誤差計測結果（CSV 形式）
  - 出力：無作為重み摂動付加誤差計測結果・敵対的重み摂動探索結果（CSV 形式）

[*2] 無作為摂動サンプルで敵対的摂動が一つも発見できない場合、重み摂動付加個別誤差 $\mathbf{r}^{\alpha}_{(x,y)}(f_w)$ の期待値の上界はその固定閾値 $\theta^{fix}(m,n,\delta_0)$ 以下であることを、確率 $(1 - \delta_0/n)$ 以上で保証できる。

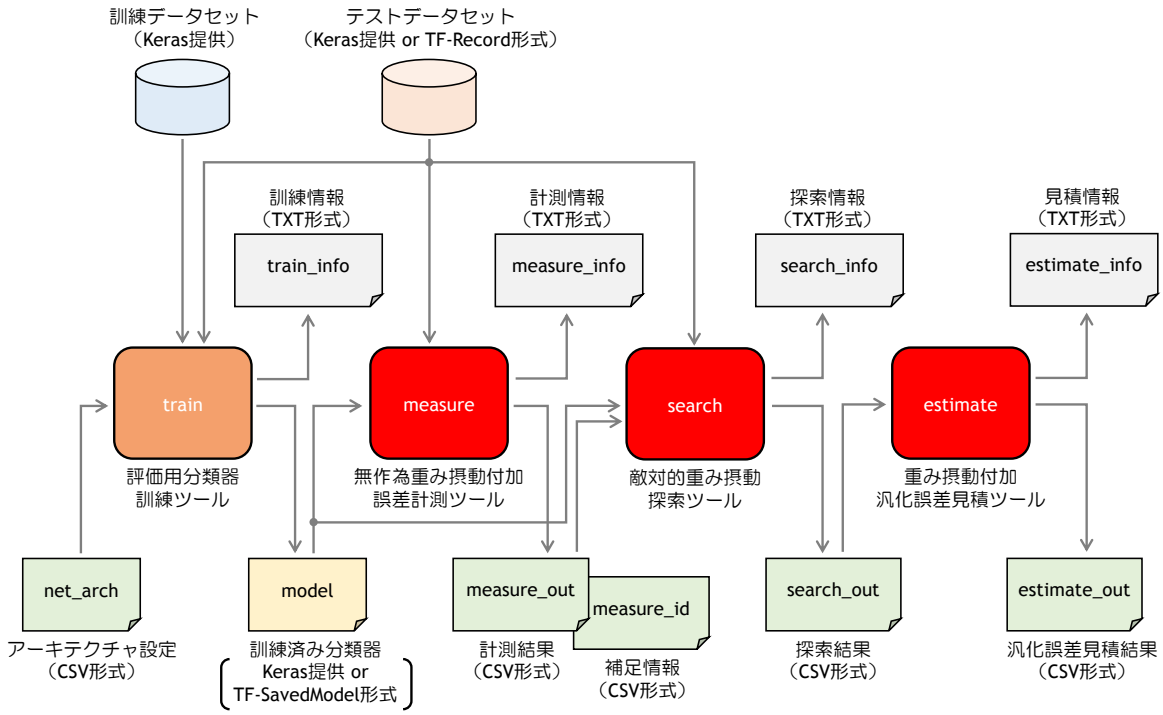

図 2　WP-GEB-Estimator を構成する四つのツールの入出力ファイルの関係（デフォルトのファイル名）

- `estimate`: 訓練済み分類器の重み摂動付加汎化誤差上界見積ツール
  - 入力：無作為重み摂動付加誤差計測結果・敵対的重み摂動探索結果（CSV 形式）
  - 出力：無作為/最悪重み摂動付加汎化誤差上界見積結果等（入力情報含む、CSV 形式）

上記の 4 つのツールを順番に実行することによって重み摂動付加汎化誤差上界 WP-GEB を見積もることができる。各ツールの実行に必要な入力ファイルと実行によって生成される出力ファイルの関係を図 2 に示す。既存の訓練済み分類器を評価する場合は、訓練ツール `train` を実行する必要はない。ただし、適応閾値の最悪摂動付加汎化誤差上界を見積りが不要な場合、すなわち、敵対的重み摂動の探索が不要な場合でも、探索ツール `search` を実行する必要はある（`search_out.csv` を生成するため）。この場合はオプション（`--skip_search`）で実際の探索をスキップできる。図 2 の入力ファイルと出力ファイルについては、各々、次の 3.2 小節と 3.3 小節で説明する。

3.2　入力ファイル

評価対象となる訓練済み分類器とテストデータセットは、各々、計測ツール `measure` のオプション`--model_dir` と`--dataset_file` で指定することができる。現在サポートされている分類器のフォーマットは TensorFlow-SavedModel 形式、データセットのフォーマットは TF-Record 形式のみである。なお、Keras が提供している分類器（例. Inception v3）やデータセット（例. MNIST）は名前を指定するだけで使用できる。指定方法は 4 節のオプションを参照してほしい。

分類器訓練ツール `train` によってシンプルな分類器を訓練することができる。訓練には分類器のアーキテクチャの情報が必要であり、アーキテクチャ情報を記述した CSV ファイルはツール `train` 実行時に読み込ま

表 1　指定可能な層の種類とパラメータ（$a \in \{\texttt{relu} \mid \texttt{linear} \mid \texttt{softmax}\}$、$n, n_i \in \mathbb{I}$、$r \in \mathbb{R}$）

type	activation	units	filters	int_tuple	regular_l2	rate
Activation	a					
Dense	a	n			r	
Conv2D	a		n	(n_1, n_2)		
MaxPooling2D				(n_1, n_2)		
Dropout						r
BatchNormalization						
Flatten						

れる。この CSV ファイルで指定可能な層の種類とパラメータ（CSV ファイルの列）を表 1 に示す。ここで、$\mathbb{I}$ は 0 以上の整数集合、$\mathbb{R}$ は 0 以上の実数集合である。表 1 に示すように、層の種類に応じてパラメータを指定する必要があるが、L2 正則化係数（regular_l2）とドロップアウト率（rate）は省略可能であり、省略した場合は 4 節で説明するオプションで指定することができる。表 1 の層を入力側から順番に記述することで、簡単な多層パーセプトロンや畳み込みニューラルネットワークを表現することができる。ディレクトリ net_arch にアーキテクチャの CSV ファイルのサンプルが保存されているので参考にしてほしい。

3.3　出力ファイル

各ツール train, measure, search, estimate で使用した入出力情報は、各々、テキストファイル train_info, measure_info, search_info, estimate_info に保存される（ファイル名はオプションで変更可能）。また、重み摂動付加汎化誤差上界のグラフ化など、見積結果の処理に適した表形式のデータは CSV ファイル estimate_out.csv に保存される（途中の見積結果 measure_out.csv, search_out.csv も含む）。measure で複数の重み摂動比を指定した場合、重み摂動比ごとに見積結果がこの CSV ファイルの 1 行に保存され、measure → search → estimate を実行するごとにその見積結果が下の行に追加される。以下、CSV ファイル estimate_out.csv の各列（A〜AL）の意味について説明する。

- measure の入出力値（measure_out.csv の列 A〜Q と同じ）
  - A（rnd_seed_measure）：ランダムシード
  - B（dataset_name）：テストデータセット名
  - C（dataset_size）：テストデータセットサイズ
  - D（dataset_offset）：テストデータセットの開始インデックス
  - E（dataset_file）：テストデータセットのファイル名
  - F（dataset_fmt）：テストデータセットのファイルのフォーマット
  - G（image_width）：テストデータセット画像の幅
  - H（image_height）：テストデータセット画像の高さ
  - I（batch_size_measure）：一度に推論するデータ数
  - J（model_dir）：分類器の名前/ディレクトリ名
  - K（perturb_bn）：バッチ正規化のパラメータへの摂動付加（0：無、1：有）

L（perturb_params_size）：重み摂動を付加する重みパラメータ数
M（perturb_ratio）：最大重み摂動幅対重み比
N（perturb_sample_size）：無作為重み摂動サンプルサイズ（無作為重み摂動付加推論回数）
O（err_num_random）：無作為重み摂動サンプルで敵対的重み摂動の存在が確認されたデータ数
P（test_err_wst）：無作為重み摂動サンプルで敵対的重み摂動の存在が確認されたデータの割合
Q（test_err_avr）：無作為重み摂動サンプルとテストデータセットに対する不正解率の平均

- search の入出力値（search_out.csv の列 R〜W と同じ）
  - R（rnd_seed_search）：ランダムシード
  - S（batch_size_search）：同時に勾配を計算するデータ数
  - T（search_mode）：敵対的重み摂動探索モード（0：FGSM、1：I-FGSM）
  - U（max_iteration）：I-FGSM の場合の最大繰返し回数
  - V（err_num_search）：探索で敵対的重み摂動の存在が確認されたデータ数
  - W（err_num）：探索または無作為サンプルで敵対的重み摂動の存在が確認されたデータ数
- estimate の入出力値
  - 最悪重み摂動（適応閾値）
    - X（gen_err_wst_adapt_ub）：摂動付加汎化誤差上界
    - Y（test_err_wst_adapt_ub）：摂動付加テスト誤差上界
    - Z（err_thr_adapt_ub）：適応閾値（期待値上界）
    - AA（err_thr_adapt）：適応閾値（テスト平均）
    - AB（conf_wst_adapt）：摂動付加汎化誤差上界信頼度
    - AC（conf0_wst_adapt）：摂動付加テスト誤差上界信頼度
  - 最悪重み摂動（固定閾値）
    - AD（gen_err_wst_fix_ub）：摂動付加汎化誤差上界
    - AE（test_err_wst_fix_ub）：摂動付加テスト誤差上界
    - AF（err_thr_fix）：固定閾値
    - AG（conf_wst_fix）：摂動付加汎化誤差上界信頼度
    - AH（conf0_wst_fix）：摂動付加テスト誤差上界信頼度
  - 無作為重み摂動
    - AI（gen_err_rnd_ub）：摂動付加汎化誤差上界
    - AJ（test_err_rnd_ub）：摂動付加テスト誤差上界
    - AK（conf_rnd）：摂動付加汎化誤差上界信頼度
    - AL（conf0_rnd）：摂動付加テスト誤差上界信頼度

4 WP-GEB-Estimator の実行

本節では、WP-GEB-Estimator の各ツールの実行コマンドと実行時オプションについて説明する。以下、引数の集合として、0 以上の整数の集合 $\mathbb{I}$、0 以上の実数の集合 $\mathbb{R}$、長さ 1 以上の文字列の集合 $\mathbb{S}$ の他、次に示す実数のリスト（区切り記号はスペース）の文字列の集合 $\mathbb{SR}^*$ も用いる。

$$\mathbb{SR}^* = \{\text{"}r_1 \;\; r_2 \;\; \cdots \;\; r_m\text{"} \mid \exists m \in \mathbb{I}.\, \forall i \in \{1, \ldots, m\}.\, r_i \in \mathbb{R}\} \subset \mathbb{S}$$

また、Tensorflow/Keras が提供するデータセットと分類器を利用するため、次の集合も用いる。

$$\texttt{KerasDataSets} = \{\texttt{"mnist"}, \texttt{"fashion\_mnist"}, \texttt{"cifar10"}\} \subset \mathbb{S}$$
$$\texttt{KerasModels} = \{\texttt{"inception\_v3"}, \texttt{"inception\_resnet\_v2"}, \texttt{"resnet50"}, \texttt{"xception"}, \texttt{"densenet121"}, \texttt{"densenet169"}, \texttt{"densenet201"}, \texttt{"vgg16"}, \texttt{"vgg19"}, \texttt{"nasnetlarge"}, \texttt{"nasnetmobile"}\} \subset \mathbb{S}$$

4.1 分類器訓練

評価用分類器訓練ツールを実行するには、ディレクトリ src で次のコマンドを入力する。

```
python train_main.py [options]
```

以下、各オプションについて説明する[*3]。

--random_seed n （$n \in \mathbb{I}$, default $n = 1$）
ランダムシードを n に設定する（ただし、$n = 0$ ならばランダムシードを設定しない）

--net_arch_file s （$s \in \mathbb{S}$, default $s =$ "net_arch/cnn_s"）
分類器のアーキテクチャを読み込むファイル名を s にする

--result_dir s （$s \in \mathbb{S}$, default $s =$ "result"）
訓練結果を保存するディレクトリ名を s にする

--model_dir s （$s \in \mathbb{S}$, default $s =$ "model"）
訓練した分類器を保存するディレクトリ名を s にする

--dataset_name s （$s \in$ KerasDataSets, default $s =$ "mnist"）
訓練/テストに用いるデータセット名を s にする

--train_dataset_size n （$n \in \mathbb{I}$, default $n = 50000$）
訓練データセットのサイズを n にする

--train_dataset_offset n （$n \in \mathbb{I}$, default $n = 0$）
訓練データセットの開始インデックスを n にする

--test_dataset_size n （$n \in \mathbb{I}$, default $n = 5000$）
テストデータセットのサイズを n にする

--test_dataset_offset n （$n \in \mathbb{I}$, default $n = 0$）
テストデータセットの開始インデックスを n にする

--validation_ratio r （$r \in [0, 1) \subset \mathbb{R}$, default $r = 0.1$）
訓練データセットのうちバリデーションに使用する割合を r にする

--sigma r （$r \in \mathbb{R}$, default $r = 0.1$）
訓練パラメータ（重みとバイアス）を初期化する正規分布の標準偏差を r にする

--batch_size n （$n \in \mathbb{I}$, default $n = 100$）
訓練データのバッチサイズを n にする

--epochs n （$n \in \mathbb{I}$, default$n = 50$）
訓練のエポック数を n にする

--dropout_rate r （$r \in [0, 1) \subset \mathbb{R}$, default $r = 0.0$）

[*3] ファイル名を指定する場合、拡張子（.txt, .csv）は不要

分類器のアーキテクチャに明記されていない場合、ドロップアウト率を r にする

`--regular_l2` r　（$r \in \mathbb{R}$, default $r = 0.0$）

分類器のアーキテクチャに明記されていない場合、L2 正則化係数を r にする

`--learning_rate` r　（$r \in \mathbb{R}$, default $r = 0.01$）

訓練の（初期）学習率を r にする

`--decay_rate` r　（$r \in \mathbb{R}$, default $r = 1.0$）

学習率の指数関数的減衰率を r にする（$r = 1.0$ ならば減衰なし）

`--decay_steps` n　（$n \in \mathbb{I}$, default $n = 0$）

学習率が減衰率倍になるステップ幅を n にする（$n = 0$ ならば減衰なし）

`--early_stop` b　（$b \in \{0, 1\}$, default $b = 0$）

$b = 1$ ならば早期終了を有効にする（$b = 0$ ならば無効にする）

`--early_stop_delta` r　（$r \in \mathbb{R}$, default $r = 0.0$）

損失値の減少が r 以下の場合に改善なしと判断する

`--early_stop_patience` n　（$n \in \mathbb{I}$, default $n = 3$）

早期終了が有効な場合、損失値の改善なしが n 回連続したときに早期終了する

`--verbose` b　（$b \in \{0, 1, 2\}$, default $b = 1$）

$b = 1$ or 2 ならば訓練中の進捗状況を表示する（$b = 0$ ならば表示しない）

4.2 無作為重み摂動付加誤差計測

無作為重み摂動付加誤差計測ツールを実行するには、ディレクトリ `src` で次のコマンドを入力する。

```
python measure_main.py [options]
```

以下、各オプションについて説明する。

`--random_seed` n　（$n \in \mathbb{I}$, default $n = 1$）

ランダムシードを n に設定する（ただし、$n = 0$ ならばランダムシードを設定しない）

`--result_dir` s　（$s \in \mathbb{S}$, default $s =$ `"result"`）

訓練結果の読込みや計測結果を保存をするディレクトリ名を s にする

`--measure_file` s　（$s \in \mathbb{S}$, default $s =$ `"measure"`）

無作為重み摂動付加誤差計測結果を保存するファイル名を s にする

`--model_dir` s　（$s \in \mathbb{S}$, default $s =$ `"model"`）

訓練済み分類器を読込むディレクトリ名を s にする

（$s \in$ `KerasModels` の場合は Keras が提供する訓練済み分類器 s を読込む）

`--dataset_name` s　（$s \in \mathbb{S}$, default $s =$ `"mnist"`）

テストデータセット名を s にする（$s \in$ `KerasDataSets` ならば Keras のデータセット s を読込む）

`--dataset_file` s　（$s \in \mathbb{S}$, default $s =$ `"~/imagenet/1k-tfrecords/validation-*-of-00128"`）

テストデータセットを読込むファイル名を s にする（$s \in$ `KerasDataSets` の場合は不要）

`--dataset_fmt` s　（$s \in \mathbb{S}$, default $s =$ `"tfrecord"`）

テストデータセットのファイルのフォーマットを s にする

（現在サポートしているフォーマットは TF-Record（"tfrecord"）のみ）

`--image_width` n　（$n \in \mathbb{I}$, default $n = 0$）

テストデータ画像の幅を n にする（Keras 提供のモデルやデータセットで指定不要の場合は 0）

`--image_height` n　（$n \in \mathbb{I}$, default $n = 0$）

テストデータ画像の高さを n にする（Keras 提供のモデルやデータセットで指定不要の場合は 0）

`--dataset_size` n　（$n \in \mathbb{I}$, default $n = 5000$）

テストデータセットのサイズを n にする

`--dataset_offset` n　（$n \in \mathbb{I}$, default $n = 0$）

テストデータセットの開始インデックスを n にする

`--batch_size` n　（$n \in \mathbb{I}$, default $n = 0$）

一度に誤差を計測するデータセットサイズを n にする（$n = 0$ ならばデータセットサイズと同じ）

`--perturb_ratios` s　（$s \in \mathbb{SR}^*$, default $s =$ "0.01 0.1 1"）

重み摂動幅対重み比（各重みの大きさに対するその重み摂動最大幅の比率）リストを s にする

（リストの先頭の重み摂動幅から順番に重み摂動付加汎化誤差上界を見積もる）

`--perturb_bn` b　（$b \in \{0, 1\}$, default $b = 0$）

$b = 1$ ならば、バッチ正規化の訓練パラメータ（スケール、シフト）にも摂動を付加する

（$b = 0$ ならば、バッチ正規化の訓練パラメータには摂動を付加しない）

`--perturb_sample_size` n　（$n \in \mathbb{I}$, default $n = 1215$）

無作為に選択する重み摂動サンプルサイズを n にする

`--verbose_measure` b　（$b \in \{0, 1\}$, default $b = 1$）

$b = 1$ ならば無作為重み摂動付加誤差計測進捗状況を表示する（$b = 0$ ならば表示しない）

4.3 敵対的重み摂動探索

敵対的重み摂動探索ツールを実行するには、ディレクトリ src で次のコマンドを入力する。

```
python search_main.py [options]
```

以下、各オプションについて説明する。

`--random_seed` n　（$n \in \mathbb{I}$, default $n = 1$）

ランダムシードを n に設定する（ただし、$n = 0$ ならばランダムシードを設定しない）

`--result_dir` s　（$s \in \mathbb{S}$, default $s =$ "result"）

計測結果の読込みや探索結果を保存をするディレクトリ名を s にする

`--measure_file` s　（$s \in \mathbb{S}$, default $s =$ "measure"）

無作為重み摂動付加誤差計測結果を読み込むファイル名を s にする

`--search_file` s　（$s \in \mathbb{S}$, default $s =$ "search"）

敵対的重み摂動探索結果を保存するファイル名を s にする

`--skip_serach` b　（$b \in \{0, 1\}$, default $b = 0$）

$b = 1$ ならば敵対的探索をスキップする（$b = 0$ ならば探索する）

（無作為重み摂動付加汎化誤差上界のみ見積もる場合は探索をスキップできる）

`--search_mode` n　（$n \in \{0, 1\}$, default $n = 0$）
　敵対的な重み摂動の探索法を mode-n にする
　・ mode-0：FGSM（勾配で損失が最大になる方向に最大の摂動を重みに付加する）
　・ mode-1：I-FGSM（不正解になるか損失低下しなくなるまで FGSM を繰り返す）
`--batch_size` n　（$n \in \mathbb{I}$, default $n = 10$）
　FGSM（mode-0）の場合、並列に勾配計算するデータセットサイズを n にする
`--max_iteration` n　（$n \in \mathbb{I}$, default $n = 20$）
　I-FGSM（mode-1）の場合の探索の最大繰返し回数（探索打切り回数）を n にする
`--verbose_search` b　（$b \in \{0, 1\}$, default $b = 1$）
　$b = 1$ ならば敵対的重み摂動探索中の進捗状況を表示する（$b = 0$ ならば表示しない）

4.4 重み摂動付加汎化誤差上界（WP-GEB）の見積り

重み摂動付加汎化誤差上界見積ツールを実行するには、ディレクトリ `src` で次のコマンドを入力する。

```
python estimate_main.py [options]
```

以下、各オプションについて説明する。

`--result_dir` s　（$s \in \mathbb{S}$, default $s =$ `"result"`）
　探索結果（計測結果含む）の読込みや見積結果を保存するディレクトリ名を s にする
`--search_file` s　（$s \in \mathbb{S}$, default $s =$ `"search"`）
　敵対的重み摂動探索結果（無作為摂動付加誤差計測結果含む）を読み込むファイル名を s にする
`--estimate_file` s　（$s \in \mathbb{S}$, default $s =$ `"estmate"`）
　重み摂動付加汎化誤差上界見積結果を保存するファイル名を s にする
`--delta` δ　（$\delta \in (0, 1) \subset \mathbb{R}$, default $\delta = 0.1$）
　重み摂動付加汎化誤差がその上界の見積結果を超える確率を δ（2.2 小節の δ）まで許容する
　（i.e. 上界の信頼度は $1 - \delta$ になる）
`--delta0_ratio` r　（$r_0 \in (0, 1) \subset \mathbb{R}$, default $r = 0.5$）
　重み摂動付加テスト誤差がその上界見積結果を超える確率を δr_0（2.2 小節の δ_0）まで許容する
　（i.e. 上界の信頼度は $1 - \delta r_0$ になる）
`--max_nm` n　（$n \in \mathbb{I}$, default $r = 10$）
　汎化誤差の見積りに用いるニュートン法の最大繰返し回数を n にする
`--eps_nm` r　（$r \in \mathbb{R}$, default $r = 0.0001$）
　汎化誤差の見積りに用いるニュートン法の許容誤差を r にする

5 WP-GEB-Estimator の実行例

本節では、WP-GEB-Estimator の実行コマンド（実行スクリプト）の例と、その実行結果の例を紹介する。なお、WP-GEB-Estimator の実行には TensorFlow と NumPy ライブラリをインストールした Python 環境が必要である。WP-GEB-Estimator 開発時に使用したソフトウェアのバージョンは Python 3.10.10,

```
python train_main.py  --net_arch_file "net_arch/mlp_s_bn"
python measure_main.py
python search_main.py
python estimate_main.py
```

図 3　実行スクリプト run.sh（簡単な分類器の訓練と評価）

TensorFlow 2.13.0, NumPy 1.23.2 である。

5.1　分類器訓練〜汎化誤差見積り

デフォルトパラメータを用いた手書き数字（MNIST）による 3 層パーセプトロンの訓練、無作為重み摂動サンプルによる誤差の計測、敵対的重み摂動の探索、汎化誤差上界の見積りは、図 3 に示すスクリプト run.sh（ディレクトリ examples に保存されている）により実行でき、その実行結果はディレクトリ results/result に保存されている。その実行時間については、MacBook Pro（CPU: Apple M2 Max, Mem: 96GB）で実行した場合、訓練時間は約 90 秒、一つの摂動幅対重み比あたりの無作為重み摂動付加誤差計測時間は約 5 分（データセットサイズ 5000、重み摂動サンプルサイズ 1215（固定閾値 1% に相当））、敵対的重み摂動探索時間は約 1 分（無作為摂動サンプルで発見した敵対的摂動数に依存）、汎化誤差見積時間は 0.1 秒以下であった。

図 4 に、実行スクリプト run.sh の実行結果（estimate_info.txt の一部）を示す。重み摂動幅対重み比が 0.01 と 0.1 の最悪重み摂動（適応閾値/固定閾値）と無作為重み摂動を付加した場合の汎化誤差上界、テスト誤差上界、閾値の見積結果が出力されている。例えば、重み摂動幅対重み比 0.01 の最悪重み摂動付加汎化誤差（適応閾値上界 0.75%））は 25% 以下（信頼度 90% 以上）である。

5.2　訓練済み分類器（Inception-v3）の評価

図 5 は、TensorFlow の Keras で提供されている訓練済み分類器 Inception-v3 の汎化誤差を見積るためのスクリプトの例（examples/run_imagenet.sh に保存されている）であり、その実行結果はディレクトリ results/result_imagenet に保存されている。なお、この実行スクリプトでは、実行前にデータセット ImageNet のファイル（TF-Record 形式）がディレクトリ ImageNet_DIR に保存されていることを仮定している。TF-Record 形式の ImageNet のファイル（1k-tfrecords-0.zip, 1k-tfrecords-1.zip）は下記の Kaggle のサイトからダウンロードできる。

```
ImageNet 1K TFrecords ILSVRC2012 - part 0
https://www.kaggle.com/datasets/hmendonca/imagenet-1k-tfrecords-ilsvrc2012-part-0
```

このスクリプト run_imagenet.sh の実行時間については、MacBook Pro（CPU: Apple M2 Max, Mem: 96GB）で実行した場合、無作為重み摂動付加誤差計測時間が約 29 分（データセットサイズ 1000、重み摂動サンプルサイズ 525（固定閾値 2% に相当）、敵対的重み摂動探索時間が約 7 分、汎化誤差計算時間が 0.1 秒以下であった。見積結果は、重み摂動幅対重み比 0.0001 の最悪重み摂動付加汎化誤差（適応閾値上界 1.4%）が 36.1% 以下（信頼度 90% 以上）、無作為重み摂動付加汎化誤差が 33.2% 以下（信頼度 90% 以上）であった。

```
Perturbation ratio = 0.01
  Random perturbation sample size: 1215
  Worst weight-perturbation (adaptive threshold):
    Perturbed generalization error bound: 25.01% (Conf: 90.00%)
    Perturbed Test error bound: 23.36% (Conf: 95.00%)
    Adaptive threshold bound (expected): 0.7648% (Conf: 90.00%)
    Adaptive threshold (average): 0.7495%
  Worst weight-perturbation (fixed threshold):
    Perturbed generalization error bound: 4.32% (Conf: 90.00%)
    Perturbed Test error bound: 3.56% (Conf: 95.00%)
    Fixed threshold: 0.9996%
  Random weight-perturbation:
    Perturbed generalization error bound: 6.23% (Conf: 90.00%)
    Perturbed Test error bound: 4.91% (Conf: 95.00%)
  (Elapsed Time: 0.0 [sec])

Perturbation ratio = 0.1
  Random perturbation sample size: 1215
  Worst weight-perturbation (adaptive threshold):
    Perturbed generalization error bound: 100.00% (Conf: 90.00%)
    Perturbed Test error bound: 99.90% (Conf: 95.00%)
    Adaptive threshold bound (expected): 0.0013% (Conf: 90.00%)
    Adaptive threshold (average): 0.0004%
  Worst weight-perturbation (fixed threshold):
    Perturbed generalization error bound: 7.34% (Conf: 90.00%)
    Perturbed Test error bound: 6.36% (Conf: 95.00%)
    Fixed threshold: 0.9996%
  Random weight-perturbation:
    Perturbed generalization error bound: 6.29% (Conf: 90.00%)
    Perturbed Test error bound: 4.96% (Conf: 95.00%)
  (Elapsed Time: 0.0 [sec])
```

図 4　実行スクリプト run.sh の実行結果（estimate_info.txt の一部）

5.3　分類器の WP-GEB の比較

ディレクトリ examples に保存されているファイル run_main.sh は、正則化無しと有りの二つの分類器を訓練し、重み摂動付加汎化誤差上界を見積もり比較すつための実行スクリプトである。アーキテクチャファイ

```
ImageNet_DIR="$HOME/datasets/imagenet/1k-tfrecords/validation"
python measure_main.py \
    --result_dir "result_imagenet" --dataset_name "imagenet" \
    --dataset_file "$ImageNet_DIR/validation-*-of-00128" \
    --dataset_size 1000 --batch_size 50 \
    --model_dir "inception_v3" \
    --perturb_ratios "0.0001" --perturb_sample_size 525
python search_main.py --result_dir "result_imagenet"
python estimate_main.py --result_dir "result_imagenet"
```

図 5　実行スクリプト `run_imagenet.sh`（訓練済み分類器 Inception-v3 の評価）

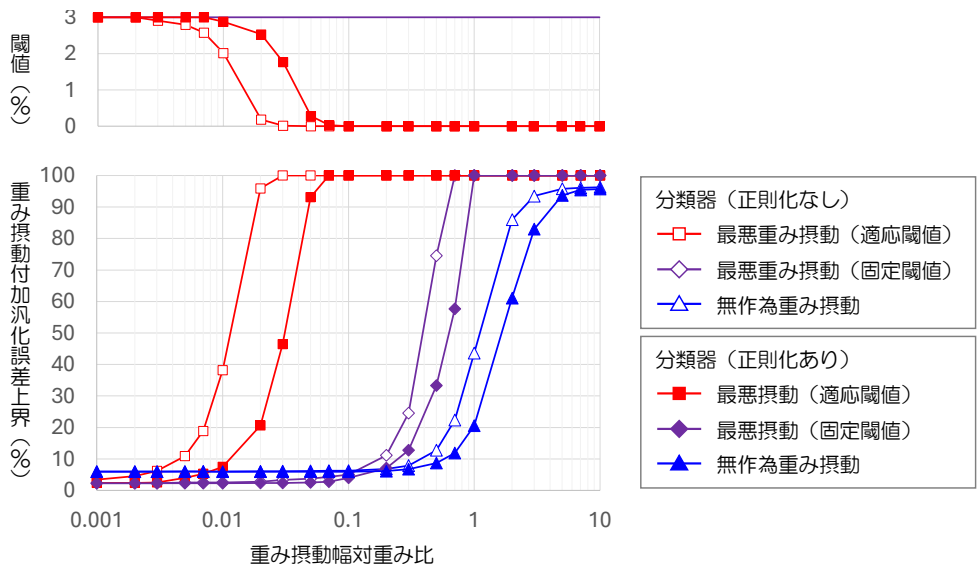

図 6　重み摂動付加汎化誤差上界と最悪重み摂動の場合の閾値上界の見積結果（信頼度 90%）

ルの正則化係数が省略されている場合はオプションで指定できるため、様々な正則化係数で簡単に試すことができる。正則化無しで訓練した分類器と正則化有り（L2 係数 0.001）で訓練した分類器の重み摂動付加汎化誤差上界（WP-GEB）と最悪重み摂動の場合の閾値上界の見積結果（信頼度 90%）を図 6 に示す。このグラフは `estimate_out.csv` を表計算ソフト（Microsoft Excel）で読込み、グラフ描画機能で作成した。図 6 の横軸は重み摂動幅対重み比であり、重み摂動を付加しない場合の二つの分類器のテスト誤差はほぼ同じであるが、重み摂動を付加することによって、正則化有りで訓練した分類器の方が重み摂動に対する耐性が高いことが明確になっている。また、最悪重み摂動（適応閾値）の場合は無作為重み摂動よりも 2 桁程度小さい重み摂動幅で性能の低下（誤差の増加）が見られる。

参考文献

[1] 大岩 寛他, 機械学習品質マネジメントガイドライン 第 4 版, Digiarc-TR-2023-03, CPSEC-TR-2023003, 2023. https://www.digiarc.aist.go.jp/publication/aiqm/

[2] J. Langford and R. Caruana, (Not) Bounding the True Error, 14th International Conference on Neural Information Processing Systems (NIPS 2001), pp.809–816, 2001.

[3] A. Maurer, A Note on the PAC Bayesian Theorem, arXiv:cs/0411099, 2004.

[4] M. Pérez-Ortiz, O. Rivasplata, J. Shawe-Taylor, and C. Szepesvári, Tighter risk certificates for neural networks, Journal of Machine Learning Research (JMLR), 2021.

[5] 磯部, 最悪重み摂動付加ニューラル分類器の汎化誤差上界の見積法, 第 38 回人工知能学会全国大会（JSAI2024）, 2024（5 月発表予定）

謝辞

本ツール WP-GEB-Estimator は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務（JPNP20006）にて開発されたものです.