

レポートについて

コード

- CNN_CIFAR-10_rep.ipynb
 - CNNによるCIFAR-10の分類(クラス数とデータ数可変Ver.)
 - CNNでCIFAR-10を分類するコード
- MLP_CIFAR-10_rep.ipynb
 - MLPによるCIFAR-10の分類(クラス数とデータ数可変Ver.)
 - MNISTでCIFAR-10を分類するコード

チェックリスト

レポートチェックリスト.xlsx

レポート用コード:

CNNによるCIFAR-10の分類(クラス数とデータ数可変Ver.)

CNN_CIFAR-10.ipynbと同様にCNNによりCIFAR-10の分類を行うコード

変更点:

CIFAR-10データセットから, 任意のクラスを選ぶことができる.

さらに, 各クラスの学習データ数を調整することができる.

CIFAR-10の各クラスの詳細

	class	count_train	count_test
0	0:airplane	5000	1000
1	1:automobile	5000	1000
2	2:bird	5000	1000
3	3:cat	5000	1000
4	4:deer	5000	1000
5	5:dog	5000	1000
6	6:frog	5000	1000
7	7:horse	5000	1000
8	8:ship	5000	1000
9	9:truck	5000	1000



抜粋したデータの内訳

	class	count_train	count_test
0	0:airplane	2500	1000
1	2:bird	2500	1000
2	6:frog	5000	1000

データの抜粋の方法

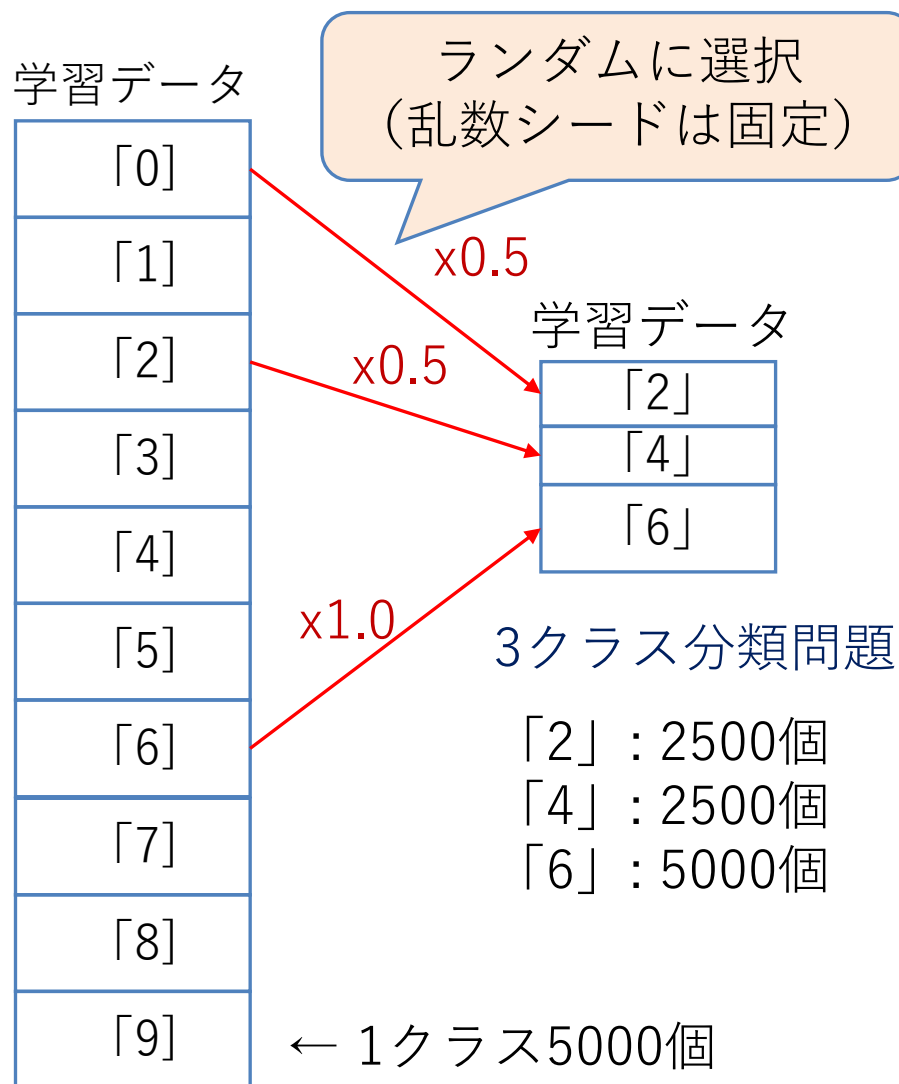
Cifar-10の全体のデータから、クラスとデータ数を指定して抜粋

問題の設定

```
1 # 使用するクラスを指定する
2 use_num = [0, 2, 6]
3
4 # 学習データの使用率
5 # (use_numと同じ要素数のリスト)
6 use_rate = [0.5, 0.5, 1.0]
7
8 print("use_num =", use_num)
9 print("use_rate =", use_rate)
```

```
use_num = [0, 2, 6]
use_rate = [0.5, 0.5, 1.0]
```

```
use_num=range(10)
use_rate=[1]*len(use_num)
とすると、元の10クラス分類問題
```



実行例

▼ 学習結果

▼ 学習結果のグラフを表示

問題設定

num_classes = 3

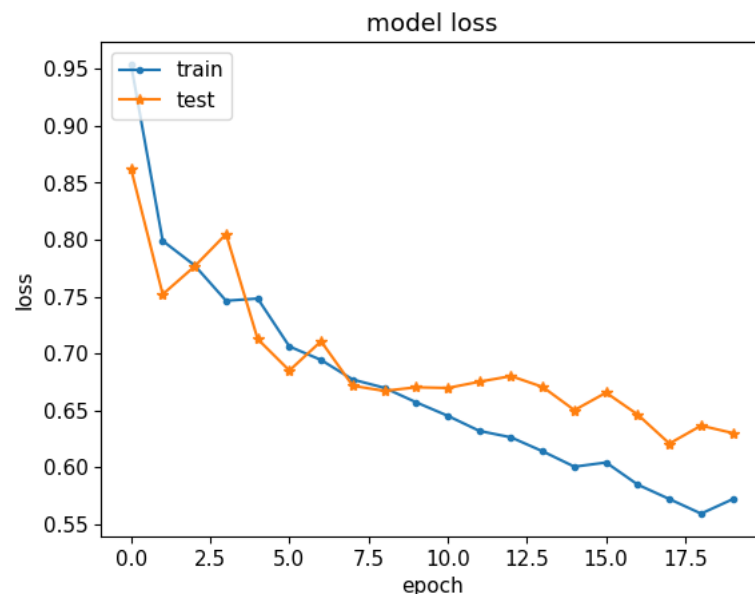
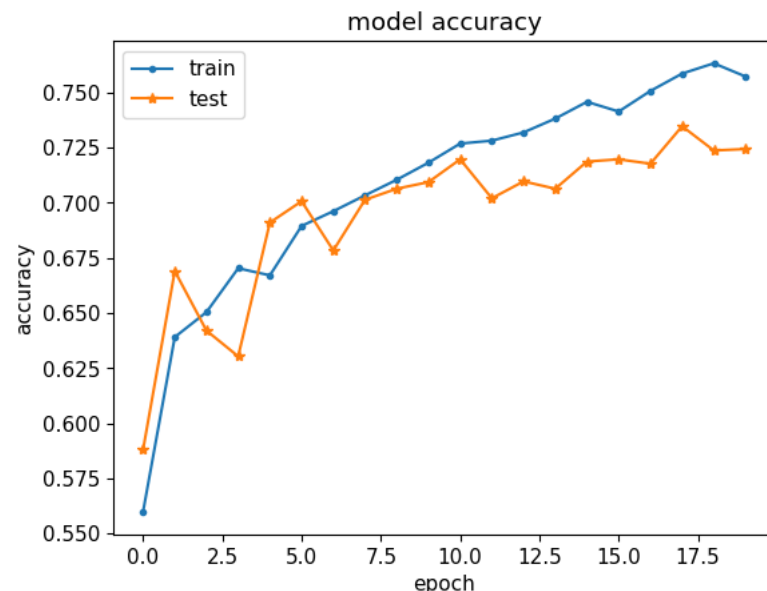
use_num = [0, 2, 6]

use_rate = [0.5, 0.5, 0.5]

学習終了時の結果

Train: loss = 0.526, accuracy=0.781

Test: loss = 0.63, accuracy=0.724



レポート用コード: MLPによるCIFAR-10の分類(クラス数とデータ数可変Ver.)

CNN版とほぼ同様, ただしカラー画像を平坦化してモデルに入力

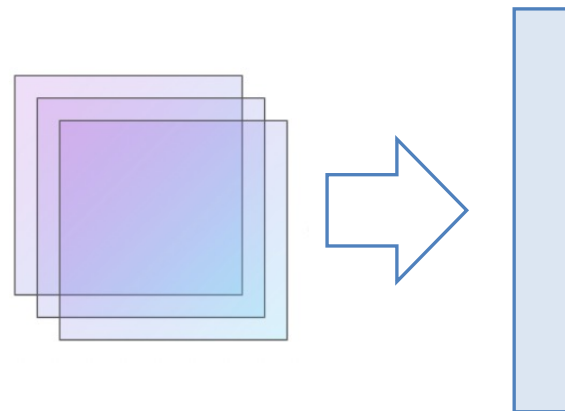
モデルの定義

CIFAR10(RGB画像)の `image_shape = (32, 32, 3)` を平坦化してMLPに入力する.

R, G, Bを
まとめて平坦化

- 最初の層は, `model.add(Flatten(input_shape=image_shape))`
- 最後の層は, `Dense(num_classes, activation='softmax')`

`image_shape = (32, 32, 3) # RGB画像`



```
8 # モデルを構築
9 model = Sequential()
10 model.add(Flatten(input_shape=image_shape)) # RGB画像を平坦化
```

平坦化する関数

レポートのテーマ

大きなテーマ：CIFAR-10に対するMLPとCNNの性能比較

各自で詳細な(独自の)テーマを設定してレポートを作成する

テーマの例

- データ数に関する検証
- クラスの違いに関する検証
- MLP, CNNのモデルの構成に関する検証
- 問題の設定(用いるデータセットについて):
 - クラス数 n は 3 以上
 - 各クラスのデータ数は任意として, n クラス分類問題を設定する.

クラス数やデータ数が多いと学習に時間がかかる(GPUリソースを多く消費する)ので, 実験はある程度少なめのクラスで(3~5クラス程度)で行ってください.

=> あとで余裕があれば, もう少し大規模なデータセットで試す

データ数に関する実験

- データ数を大きくしていくと、学習結果は？

use_rate =
[0.1, 0.1, 0.1]

	class	count_train	count_test
0	0:airplane	500	1000
1	2:bird	500	1000
2	6:frog	500	1000



use_rate =
[0.3, 0.3, 0.3]

	class	count_train	count_test
0	0:airplane	1500	1000
1	2:bird	1500	1000
2	6:frog	1500	1000



use_rate =
[0.5, 0.5, 0.5]

	class	count_train	count_test
0	0:airplane	2500	1000
1	2:bird	2500	1000
2	6:frog	2500	1000



- クラス間のデータ数の非均一度合いが大きくなると、学習結果は？

use_rate =
[0.1, 0.1, 0.1]

	class	count_train	count_test
0	0:airplane	500	1000
1	2:bird	500	1000
2	6:frog	500	1000



use_rate =
[0.1, 0.3, 0.1]

	class	count_train	count_test
0	0:airplane	500	1000
1	2:bird	1500	1000
2	6:frog	500	1000



use_rate =
[0.1, 0.5, 0.1]

	class	count_train	count_test
0	0:airplane	500	1000
1	2:bird	2500	1000
2	6:frog	500	1000



- データ数に応じてバッチサイズも変えた方が良い？
- 最適なバッチサイズは存在する？

バッチサイズ小 ⇔ バッチサイズ大

抜粋するクラスの違いに関する実験

CIFAR10のクラス

0: airplane	機械
1: automobile	機械
2: bird	生物
3: cat	生物
4: deer	生物
5: dog	生物
6: frog	生物
7: horse	生物
8: ship	機械
9: truck	機械

- 動物同士は誤分類しやすい？
 - 例) 猫を犬と分類, 犬を猫と分類
- 誤分類を減らすためには？
- 同じ3クラス分類問題でも, 結果はどう異なるか？

設定 1

3: cat	生物
4: deer	生物
5: dog	生物

設定 2

0: airplane	機械
1: automobile	機械
8: ship	機械

設定 3

3: cat	生物
4: deer	生物
9: truck	機械

モデルの構成に関する実験

モデルの規模・構成と性能の関係は？

パラメータ数：小

	Layer (type)	Output Shape	Param #
0	Conv2D	(30, 30, 32)	896
1	MaxPooling2D	(15, 15, 32)	0
2	Conv2D	(13, 13, 16)	4624
3	MaxPooling2D	(6, 6, 16)	0
4	Flatten	(576,)	0
5	Dense	(16,)	9232
6	Dense	(3,)	51

モデルのパラメータ総数: 14803

パラメータ数：中

	Layer (type)	Output Shape	Param #
0	Conv2D	(30, 30, 64)	1792
1	MaxPooling2D	(15, 15, 64)	0
2	Conv2D	(13, 13, 32)	18464
3	MaxPooling2D	(6, 6, 32)	0
4	Conv2D	(4, 4, 16)	4624
5	MaxPooling2D	(2, 2, 16)	0
6	Flatten	(64,)	0
7	Dense	(16,)	1040
8	Dense	(3,)	51

モデルのパラメータ総数: 25971

パラメータ数：大

	Layer (type)	Output Shape	Param #
0	Conv2D	(30, 30, 128)	3584
1	MaxPooling2D	(15, 15, 128)	0
2	Conv2D	(13, 13, 64)	73792
3	MaxPooling2D	(6, 6, 64)	0
4	Conv2D	(4, 4, 32)	18464
5	MaxPooling2D	(2, 2, 32)	0
6	Flatten	(128,)	0
7	Dense	(64,)	8256
8	Dense	(32,)	2080
9	Dense	(3,)	99

モデルのパラメータ総数: 106275



- 大規模なモデル（パラメータ数が多い or 層が多い）ほど性能は良い？
- 学習に必要なエポック数は？（学習時間は？）
- 効果的な層の組み合わせ(層の数, 種類)は？
- Dropoutの効果は？
- なるべく小さな規模のモデルで性能を向上させるには？

データフレームの変換(1)

- データフレームをエクセルに貼り付け

```
1 # モデルの情報をDataFrameで表示する
2 # model_summary_to_dataframe関数は上のセルで定義
3 model_df = model_summary_to_dataframe(model)
4 model_df
```

	Layer (type)	Output Shape	Param #
0	Conv2D	(30, 30, 64)	1792
1	MaxPooling2D	(15, 15, 64)	0
2	Conv2D	(13, 13, 32)	18464
3	MaxPooling2D	(6, 6, 32)	0
4	Flatten	(1152,)	0
5	Dense	(16,)	18448
6	Dropout	(16,)	0
7	Dense	(3,)	51

	Layer (type)	Output Shape	Param #
0	Conv2D	(30, 30, 64)	1792
1	MaxPooling2D	(15, 15, 64)	0
2	Conv2D	(13, 13, 32)	18464
3	MaxPooling2D	(6, 6, 32)	0
4	Flatten	(1152,)	0
5	Dense	(16,)	18448
6	Dropout	(16,)	0
7	Dense	(3,)	51

データフレームをマウスで
範囲指定して右クリック
(or 2本指タップ)
でメニューからコピー

エクセルを開く



形式を選択してペーストから
Unicodeテキストを選択

Layer (type)	Output Shape	Param #	
0	Conv2D	(30, 30, 64)	1792
1	MaxPooling2D	(15, 15, 64)	0
2	Conv2D	(13, 13, 32)	18464
3	MaxPooling2D	(6, 6, 32)	0
4	Flatten	(1152,)	0
5	Dense	(16,)	18448
6	Dropout	(16,)	0
7	Dense	(3,)	51

[LaTeX Table Generator](#)
(LaTeXへ変換)

データフレームの変換(2)

データフレームの内容をLatexの表として貼り付け

```
1 # DataFrame を LaTeX 形式の文字列に変換する
2 latex = model_df.to_latex(index=False)
3
4 # LaTeX コードに横線を追加する
5 styled_latex = latex.replace('\\begin{tabular}', '\\begin{tblr}')
6 styled_latex = styled_latex.replace('\\hline', '\\midrule')
7 styled_latex = styled_latex.replace('\\end{tabular}', '\\end{tblr}')
8
9 # LaTeX コードを出力する
10 print(styled_latex)
```

```
\begin{tblr}{llr}
\toprule
Layer (type) & Output Shape & Param \# \\
\midrule
Conv2D & (30, 30, 64) & 1792 \\
MaxPooling2D & (15, 15, 64) & 0 \\
Conv2D & (13, 13, 32) & 18464 \\
MaxPooling2D & (6, 6, 32) & 0 \\
Flatten & (1152,) & 0 \\
Dense & (16,) & 18448 \\
Dropout & (16,) & 0 \\
Dense & (3,) & 51 \\
\bottomrule
\end{tblr}
```

<ipython-input-315-31c3882ac0cd>:2: FutureWarning: In
latex = model_df.to_latex(index=False)

```
\begin{tabular}{llr}
:
\end{tabular}
```

をLaTeXの表のサンプル
に貼り付ける

```
36 % 表の作成
37 \begin{table}[htbp]
38 \begin{center}
39 \caption{表のサンプル}
40 \label{tbl:sample_table}
41
42 \begin{tblr}{llr}
43 \hline
44 & 平均 & 標準偏差 \\
45 手法1 & 10.5 & 4.5 \\
46 手法2 & 11.4 & 2.1 \\
47 手法3 & 21.5 & 3.8 \\
48 \end{tblr}
49
50 \end{center}
51 \end{table}
52
53
```

表 1: 表のサンプル

	平均	標準偏差
手法 1	10.5	4.5
手法 2	11.4	2.1
手法 3	21.5	3.8

コピー

```
36 % 表の作成
37 \begin{table}[htbp]
38 \begin{center}
39 \caption{表のサンプル}
40 \label{tbl:sample_table}
41
42 \begin{tblr}{llr}
43 \toprule
44 Layer (type) & Output Shape & Param \# \\
45 \midrule
46 Conv2D & (30, 30, 64) & 1792 \\
47 MaxPooling2D & (15, 15, 64) & 0 \\
48 Conv2D & (13, 13, 32) & 18464 \\
49 MaxPooling2D & (6, 6, 32) & 0 \\
50 Flatten & (1152,) & 0 \\
51 Dense & (16,) & 18448 \\
52 Dropout & (16,) & 0 \\
53 Dense & (3,) & 51 \\
54 \bottomrule
55 \end{tblr}
56 \end{center}
57 \end{table}
```

表 1: 表のサンプル

Layer (type)	Output Shape	Param #
Conv2D	(30, 30, 64)	1792
MaxPooling2D	(15, 15, 64)	0
Conv2D	(13, 13, 32)	18464
MaxPooling2D	(6, 6, 32)	0
Flatten	(1152,)	0
Dense	(16,)	18448
Dropout	(16,)	0
Dense	(3,)	51

レポートの構成の例

1. 目的

✓レポートの目的を書く.

2. CIFAR-10データセット

✓データセットの説明を書く.

3. 問題の設定

✓各自の問題の設定を説明する.

4. モデルの設定

✓実験で用いるモデルの種類や構成を説明する.

5. 実験

✓実験の手順や結果を説明する.

6. 考察 (or まとめ)

参考文献

実験結果のまとめや、モデルの性能差が付いた理由など、考えられることを書く.

参考文献について

Google scholarで「Cifar10 研究報告」や「畳み込み 研究報告」など
「キーワード + 研究報告」とすると、**研究会などの原稿**がヒットしやすい

情報処理学会研究報告

- ページ数は少なめ
- (基本的に)査読なし



記事 約 174 件 (0.05 秒)  プロフィール

期間指定なし

2023 年以降

2022 年以降

2019 年以降

期間を指定...

GAN を用いたデータ拡張

河野曜平, 川本一彦 - **研究報告**コンピュータビジョンとイメージ ..., 2017 - ipsj.ixsq.nii.ac.jp

... 本研究では,近年盛んに**研究**されている機械学習を用いた画像の生成方法を用いて,データ拡張することを提案する.そして,実際に Convolutional Neural Network(CNN)で学習を行い,幾何変形を用...

☆ 保存 引用 被引用数: 3 関連記事 全 2 バージョン

[PDF] nii.ac.jp

関連性と並び替え

日付順に並び替え

深層学習を用いた物体の位置姿勢推定に関する研究

河野本柝 - **研究報告**ソフトウェア工学 (SE), 2020 - ipsj.ixsq.nii.ac.jp

... 概要:本研究では 3DCG 空間上で撮影した画像から物体の位置と姿勢を推定するシステムを構築した.作成した学習モデルは画像解析に特化し,扱いやすい VGG16 モデルを参考にした.結果とし物体...

☆ 保存 引用 関連記事 全 2 バージョン

[PDF] nii.ac.jp

すべての言語

英語 と 日本語のペ

pdfが
閲覧可能

文献情報について

関連性で並べ替え
日付順に並べ替え

すべての言語
英語 と 日本語のペ

深層学習を用いた物体の位置姿勢推定に関する研究

河野本祈 - 研究報告ソフトウェア工学 (SE), 2020 - ipsj.ixsq.nii.ac.jp

... 概要:本研究では 3DCG 空間上で撮影した画像から物体の位置と姿勢を推定するシステムを構築した.作成した学習 モデルは画像解析に特化し,扱いやすい VGG16 モデルを参考にした.結果とし物体...

☆ 保存 ㊦ 引用 関連記事 全 2 バージョン ㊦

[PDF] nii.ac.jp

引用 -> BibTeX

文献情報

```
@article{河野本祈2020深層学習を用いた物体の位置姿勢推定に関する研究,  
  title={深層学習を用いた物体の位置姿勢推定に関する研究},  
  author={河野本祈 and others},  
  journal={研究報告ソフトウェア工学 (SE)},  
  volume={2020},  
  number={5},  
  pages={1--8},  
  year={2020}  
}
```

参考文献

- [1] Xiaodong Li, “Efficient differential evolution using speciation for multimodal function optimization”, Proc.of GECCO2005, pp.873-880, 2005.
- [2] 木村孝作, 原章, 市村匠, 高濱徹行, “近傍グラフに基づく種分化を導入した Differential Evolution による複数解の探索”, 第 26 回ファジィシステムシンポジウム (FSS2010), pp.566-571, 2010.

著者, タイトル, 雑誌名, ページ数, 年

提出について

- レポートはLatexで作成すること [Overleaf](#)
- レポートのフォーマットは,
- レポートチェックリスト
レポートチェックリストの項目を, すべて確認(チェック)した上で提出すること
- レポートの締め切り