

知能システムレポート 4

三浦夢生

2021 年 1 月 3 日

1 目的

ディープラーニングに用いられる手法やライブラリ、モデルの構築や評価の方法を実際にプログラムを動かすことを通して学ぶ。

2 前提知識

今回用いたライブラリについて簡単な解説を行う。

2.1 TensorFlow

Google が提供するオープンソースの機械学習ライブラリだが、機械学習に限らずテンソル計算も行える。また、大量の画像などのデータセットも提供している。クラウドやブラウザ、モバイルや IoT デバイス上などでも構築でき、様々なサービスを支える技術となっている。

データフローグラフを用いることで複雑なネットワークを記述でき、また柔軟にネットワークを構築できる。後述する Keras も含まれているが、データフローを考慮して構築する必要があるため、初心者には少々難易度が高い。

2.2 Keras

Python で書かれた高水準の機械学習ライブラリのこと。内部的には TensorFlow で計算を行っており、Keras は多少の柔軟性を引き換えに機械学習の難易度を下げている。TensorFlow と同様にデータセットの提供もしている。

数行のコードだけでモデルの構築、トレーニング、評価が可能であり、機械学習の入門ハードルが大幅に下げられたとも言われている。

2.3 CIFAR-10

Keras から呼び出して利用ができるデータセットで、学習に利用できる。

6 万枚もの乗り物や動物のラベル付きカラー画像で、学習やベンチマークに利用されている。

3 方法

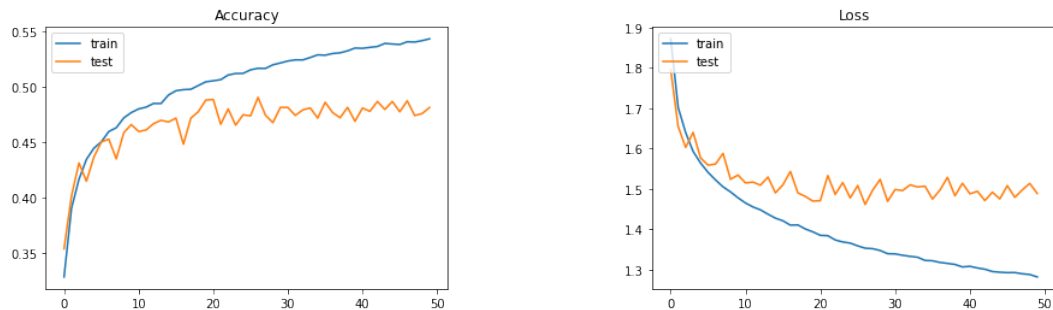
Colab 上で付録に示すソースコードを実行し、結果を得、考察する。

4 結果

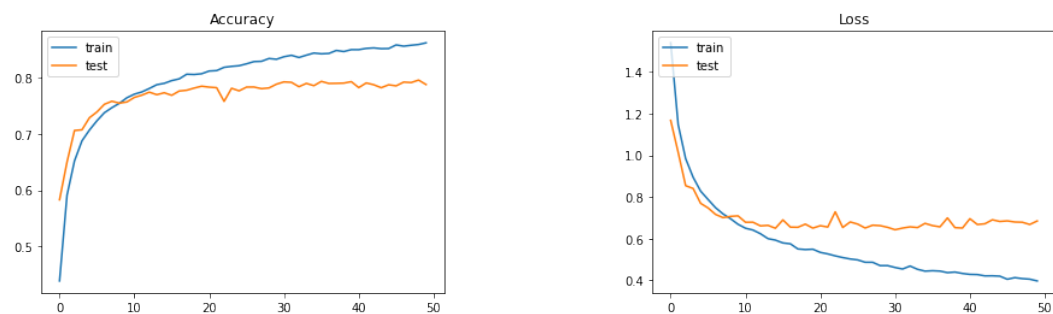
より画像の学習を得意とする CNN のほうが高い正解率を出せるモデルの作成に成功した。しかし、ネットワークの複雑さからか CNN のほうが時間がかかった。

以下、正解率及びロスのグラフを示す。

4.1 MLP による学習の結果



4.2 CNN による学習の結果



5 付録

今回用いたソースコードを以下に示す.

5.1 MLP による学習

MLP(Multi Layer Perceptron) とは, 入力層・中間層・出力層の少なくとも 3 層からなる順伝播型ニューラルネットワークである.

学習にはバックプロパゲーションを用いる. 今回のソースコードでは Adam を最適化アルゴリズムとし, 多クラス交差エントロピーを損失関数としている. Adam とは慣性項を追加するモーメンタム法と学習率を調整する RMSProp を組み合わせた, 現在デファクトスタンダードとなっている最適化アルゴリズムである. また, 多クラス交差エントロピー (Categorical Cross Entropy) とはモデルの出力の log 値と正解をかけたものの総和を損失とする手法である. 損失関数の値が大きいときに学習幅が大きいために学習スピードが速い.

```
1 import matplotlib.pyplot as plt
2 import keras
3 from keras.datasets import cifar10
4 from keras.models import Sequential
```

```

5 from keras.layers import Dense, Dropout
6
7 num_classes = 10
8 im_rows = 32
9 im_cols = 32
10 im_size = im_rows * im_cols * 3
11
12 (X_train, y_train), (X_test, y_test) = cifar10.load_data()
13
14 X_train = X_train.reshape(-1, im_size).astype('float32') / 255
15 X_test = X_test.reshape(-1, im_size).astype('float32') / 255
16
17 y_train = keras.utils.to_categorical(y_train, num_classes)
18 y_test = keras.utils.to_categorical(y_test, num_classes)
19
20 model = Sequential()
21 model.add(Dense(512, activation='relu', input_shape=(im_size,)))
22 model.add(Dense(num_classes, activation='softmax'))
23
24 model.compile(
25     loss='categorical_crossentropy',
26     optimizer='adam',
27     metrics=['accuracy']
28 )
29
30 hist = model.fit(X_train, y_train, batch_size=32, epochs=50, verbose=1, validation_data
    =(X_test, y_test))
31
32 score = model.evaluate(X_test, y_test, verbose=1)
33 print('seikai =', score[1], 'loss = ', score[0])
34
35 model.save('./gdrive/My Drive/Colab Notebooks/cifar10-mlp.h5')
36
37 plt.plot(hist.history['accuracy'])
38 plt.plot(hist.history['val_accuracy'])
39 plt.title('Accuracy')
40 plt.legend(['train', 'test'], loc='upper left')
41 plt.show()
42 plt.plot(hist.history['loss'])
43 plt.plot(hist.history['val_loss'])
44 plt.title('Loss')
45 plt.legend(['train', 'test'], loc='upper left')
46 plt.show()

```

5.2 CNN による学習

CNN(Convolutional Neural Network) とは、通常のニューラルネットワークに「畳み込み」や「プーリング」といった処理を追加したものであり、画像の深層学習においてメジャーな手法である。

畳み込みとは、カーネル (フィルターとも言う) という格子状の数値データと、それと同サイズの元画像の一部の数値データに関して各要素の積の和を計算することで一つの数値を得 (これをテンソルという)、これを繰り返すことで特徴マップを作成することである。またプーリングとは、ウィンドウと呼ばれるサイズで画像の一部にフォーカスを当てて、数値を作り出すことで、ウィンドウのうち最大値を選ぶ手法を最大値プーリングといい、ウィンドウ内の平均値をとる手法を平均値プーリングという。

また以下のモデルは画像認識コンテストで優秀な成績を収めたチームが作成したモデルに似ていることから VGG like と呼ばれている。

```
1 import matplotlib.pyplot as plt
2 import keras
3 from keras.datasets import cifar10
4 from keras.models import Sequential
5 from keras.layers import Dense, Dropout, Activation, Flatten
6 from keras.layers import Conv2D, MaxPooling2D
7
8 num_classes = 10
9 im_rows = 32
10 im_cols = 32
11 in_shape = (im_rows, im_cols, 3)
12
13 (X_train, y_train), (X_test, y_test) = cifar10.load_data()
14
15 X_train = X_train.astype('float32') / 255
16 X_test = X_test.astype('float32') / 255
17
18 y_train = keras.utils.to_categorical(y_train, num_classes)
19 y_test = keras.utils.to_categorical(y_test, num_classes)
20
21 model = Sequential()
22 model.add(Conv2D(32, (3, 3), padding='same', input_shape=in_shape))
23 model.add(Activation('relu'))
24 model.add(Conv2D(32, (3, 3)))
25 model.add(Activation('relu'))
26 model.add(MaxPooling2D(pool_size=(2, 2)))
27 model.add(Dropout(0.25))
28
29 model.add(Conv2D(64, (3, 3), padding='same'))
30 model.add(Activation('relu'))
31 model.add(Conv2D(64, (3, 3)))
32 model.add(Activation('relu'))
```

```

33 model.add(MaxPooling2D(pool_size=(2, 2)))
34 model.add(Dropout(0.25))
35
36 model.add(Flatten())
37 model.add(Dense(512))
38 model.add(Activation('relu'))
39 model.add(Dropout(0.5))
40 model.add(Dense(num_classes))
41 model.add(Activation('softmax'))
42
43 model.compile(
44     loss='categorical_crossentropy',
45     optimizer='adam',
46     metrics=['accuracy']
47 )
48
49 hist = model.fit(X_train, y_train, batch_size=32, epochs=50, verbose=1, validation_data
    =(X_test, y_test))
50
51 score = model.evaluate(X_test, y_test, verbose=1)
52 print('seikai =', score[1], 'loss = ', score[0])
53
54 model.save('./gdrive/My Drive/Colab Notebooks/cifar10-cnn.h5')
55
56 plt.plot(hist.history['accuracy'])
57 plt.plot(hist.history['val_accuracy'])
58 plt.title('Accuracy')
59 plt.legend(['train', 'test'], loc='upper left')
60 plt.show()
61 plt.plot(hist.history['loss'])
62 plt.plot(hist.history['val_loss'])
63 plt.title('Loss')
64 plt.legend(['train', 'test'], loc='upper left')
65 plt.show()

```
