

知能システム学特論レポート

(DL2 班) Caffe on Ubuntu

2015 年 7 月 21 日

1 報告者

15344203	有田 裕太
15344206	緒形 裕太
15344209	株丹 亮
12104125	宮本 和

2 進行状況

- CNN の評価関数について
- 独自の訓練データにより作成した識別器を用いた識別結果

3 理論研究

3.1 誤差関数

本班の caffe を用いた画像分類では、第 4 回レポートで記したような多クラス分類を行っており、出力関数はソフトマックス関数である。多クラス分類ではネットワークが実現する関数を各クラスの事後確率のモデルであると見なし、そのモデルのもとで訓練データに対するネットワークパラメータの尤度を評価し、これを最大化する。いま訓練データとして、入力 \mathbf{x} とその正解クラスの組 C_k が与えられたとする。このときの目標出力の 2 値の値を K 個並べたベクトル \mathbf{d}_n によって表現すると、事後分布は次式のようなになる。

$$p(\mathbf{d}|\mathbf{x}) = \prod_{k=1}^K p(C_k|\mathbf{x})^{d_k} \quad (1)$$

これより、訓練データ $(\mathbf{x}_n|\mathbf{d}_n)(n = 1, \dots, N)$ に対する \mathbf{w} の尤度は

$$L(\mathbf{w}) = \prod_{n=1}^N p(\mathbf{x}_n|\mathbf{x}_n; \mathbf{w}) = \prod_{n=1}^N \prod_{k=1}^K p(C_k|\mathbf{x})^{d_{nk}} = \prod_{n=1}^N \prod_{k=1}^K (y_k(\mathbf{x}; \mathbf{w}))^{d_{nk}} \quad (2)$$

と導ける。この尤度の対数として符号を反転した次の式を誤差関数として用いる。この関数は交差エントロ

ピー (cross entropy) と呼ばれる.

$$E(\mathbf{w}) = - \sum_{n=1}^N \sum_{k=1}^N d_{nk} \log y_k(\mathbf{x}_n; \mathbf{w}) \quad (3)$$

4 プログラミング

4.1 Python を用いたキャラクター識別用プログラムの作成

前回までで独自のデータセットを用意して学習を行い、モデルを作成した. Caffe では.caffemodel という形式で保存され, これと CNN がどのように構成されているのかを表す.prototxt ファイルを用いてキャラクター識別用プログラムを作成した. これらを扱うための Caffe が提供している API と, 画像を読み込んだり入力した画像に識別結果を書き込むために OpenCV を用いた.

まず.caffemodel と.prototxt を読み込む部分のプログラムを示す. mean_array は平均画像等が格納されているリストである. raw_scale は読み込まれた入力画像をどのスケールにするかを指定する.

```
1 classifier = caffe.Classifier(
2     'cifar10_full.prototxt',
3     'cifar10_full_150717_iter_60000.caffemodel',
4     mean=mean_array,
5     raw_scale=255)
```

そして, 入力画像から OpenCV で実装されている Haar-like 特徴量から人の顔を抽出し, これを Caffe の識別器にかける. この部分のプログラムを示す.

```
1 predictions = classifier.predict([image], oversample=False)
2 pred = np.argmax(predictions)
```

classifier.predict() で抽出された顔が各ラベルの人の顔である確率がリターンされる. そして numpy の関数を用いて np.argmax() によって確率がリストとして代入された predictions から最も高い確率のラベルを取得する. この値と関連付けられたキャラクター名と確率の値を OpenCV の API を用いて入力画像に書き込む. これを Haar-like 特徴量から取得された顔の数だけ繰り返す.

4.2 識別に用意した画像

作成した識別器の性能を確認するために用意した画像を Fig.1 と Fig.2 に示す. Fig.1 は学習したキャラクターしか含まれていないが, Fig.2 では学習していないキャラクターしか含まれていない. 従って, Fig.1 ではキャラクター名それぞれに適切な名前が推定され, Fig.2 では全て “etc” と推定されることが望ましい.

4.3 識別結果

Fig.1 と Fig.2 を識別した結果をそれぞれ Fig.3 と Fig.4 に示す. Fig.3 ではキャラクター名が全て適切に推定された. しかし, Fig.4 では全て “etc” とならなければならないが一部のキャラクターで誤認識が発生した. 誤認識が起こったキャラクターとそのラベルのキャラクターを比較すると髪の色が似ているなどの特徴があることが分かった.

5 今後の課題

- 理論研究を進める.
- 作成した識別器でどのような特徴量が利用されているのかを調査する.



Fig.1 学習したキャラクターのみが含まれる画像



Fig.2 学習したキャラクターが含まれていない画像

