

第 9 回 知能システム学特論レポート

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

西田研究室, 計算力学研究室

2015 年 7 月 16 日

進捗状況

理論研究の進捗

畳込みニューラルネットワークの理論について

プログラミングの進捗

学習器のパラメータ設定について

データセットを作成し、学習を行った結果

勾配の計算

畳み込み層の計算式

$$u^{(l)} = W^{(l)}u^{(l-1)} + b^{(l)} \quad (1)$$

$$z^{(l)} = f^{(l)}(u) \quad (2)$$

- 逆伝播の計算も多層パーセプトロンるときと基本的に同じ
- 同じフィルタの係数が何度も現れてしまう

重みの計算式

$$W_{ji} = t_{ji}^T h \quad (3)$$

- h : フィルタの係数 h_{pqkm} を適当な順に並べたベクトル
- t_{ji} : h と同じ長さで, h との内積が重み w_{ji} を与えるベクトル

勾配の計算

層 l の重みの勾配

$$\partial \mathbf{W} = \delta^{(l)} \mathbf{z}^{(l-1)\top} \quad (4)$$

- $\delta^{(l)}$: 層 l のデルタ
- $\mathbf{z}^{(l-1)}$: $l-1$ 層からの出力

\mathbf{W} の多くの成分はもともと 0 であり、そうでない成分も重み共有により同じ変数 (フィルタの係数) に対応する。そこでフィルタの係数 \mathbf{h} についての勾配 $\partial \mathbf{h}$ に変形する必要がある。

フィルタ係数についての勾配成分

$$(\partial \mathbf{h})_r = \sum_{i,j} (\mathbf{T}_r \odot \partial \mathbf{W})_{ji} \quad (5)$$

- \odot : 成分ごとの積
- \sum : 行列の全成分の和

勾配の計算

プーリング層には学習の対象となるパラメータはないので、勾配計算は必要ないが、さらに下層に伝えるデルタの逆伝播計算は必要である。

平均プーリング層の逆伝播計算

$$w_{ji}^{(l+1)} = \begin{cases} \frac{1}{H^2} & (\text{if } i \in P_j) \\ 0 & (\text{otherwise}) \end{cases} \quad (6)$$

- P_j : 層 $l+1$ のユニット j のサイズ $H \times H$ の受容野
- 順伝播時のプーリングの結果により重みが変わる
- 逆伝播計算では最大値を返したユニットにデルタがそのまま伝えられる

学習パラメータの設定

学習を行う上で必要なパラメータについて説明する.

この設定が記述されているファイルは `cifar10_quick_solver.prototxt` である.

```
1 net: "examples/cifar10/cifar10_quick_train_test.prototxt"
2 test_iter: 100
3 test_interval: 500
4 base_lr: 0.0001
5 momentum: 0.9
6 weight_decay: 0.004
7 lr_policy: "fixed"
8 display: 100
9 max_iter: 4000
10 snapshot: 4000
11 snapshot_prefix: "examples/cifar10/cifar10_quick"
12 solver_mode: GPU
```

単位 [batch] の定義

教師データを一度にいくつ処理するか（バッチサイズ）を決定し、これを `1[batch]` とする.

学習パラメータの設定

各項目の意味を以下に示す.

`net` : 学習用ネットワーク定義ファイルを指定する.

`test_iter` : 学習中の正答率評価を 1 回行うのに使う評価セットのデータ数をバッチ数で指定.

`test_interval` : テストデータから正答率評価を行う間隔をバッチ数で指定.

`base_lr`, `momentum`, `weight_decay`, `lr_policy` : 学習率に関する設定.

`display` : 学習中のステータスを出力する回数をバッチ数で指定.

`max_iter` : 学習の計算を最大どれだけ続けるかを訓練データのバッチ数で指定.

`snapshot`, `snapshot_prefix` : 学習の途中経過を保存する間隔と場所を指定.

`solver_mode` : 学習を CPU のみ, あるいは GPU を用いるかを指定.

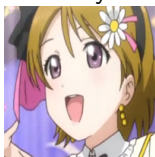
学習データの用意

eri



1908 枚

hanayo



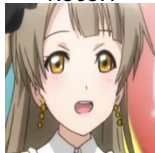
2220 枚

honoka



2733 枚

kotori



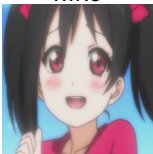
1722 枚

maki



1874 枚

niko



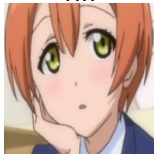
1910 枚

nozomi



1489 枚

rin



2425 枚

umi



1703 枚

他に負例 (etc) として 7001 枚の画像 ($200 \times 200 \times 3$) を用意

今回はネットワークモデルに cifar10 のモデルを用いて学習を行った.

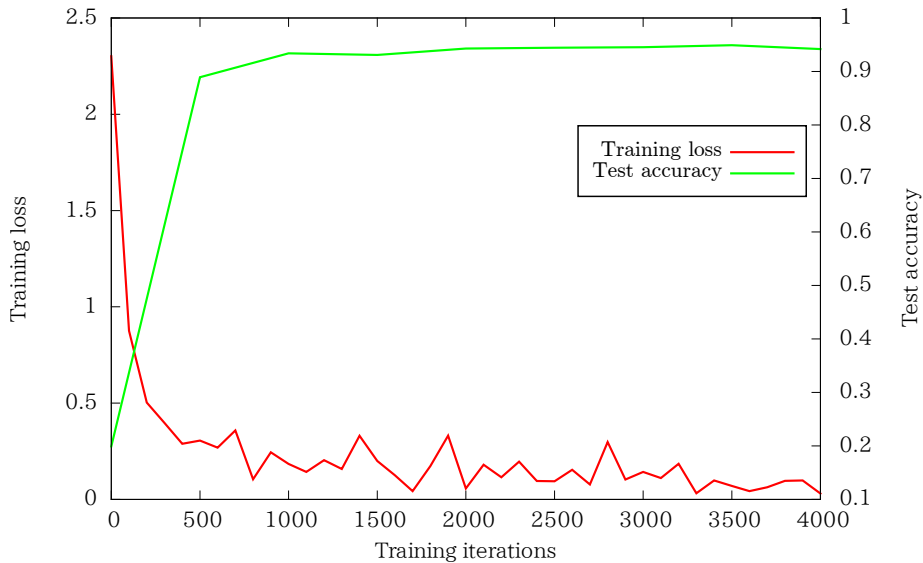
```
1 build/tools/caffe train --solver examples/cifar10/  
  cifar10_quick_solver.prototxt
```

cifar10_quick.prototxt

畳込み層 3 層, プーリング層 3 層, 全結合層 2 層, 活性化関数は Softmax 関数を用いている.

- test_iter : 100
- max_iter : 4000
- batch_size : 100
- solver_mode : CPU

學習結果



今後の課題

理論研究

CNN の詳細な調査

プログラミング

学習された識別器を用いて実際に識別を試みる