

知能システム学特論 最終発表

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 30 日

人工ニューラルネットワーク

誤差逆伝搬法

80 年から 90 年代，誤差逆伝搬法（back propagation）が多層ニューラルネットワークの学習方法として確立

しかし，誤差逆伝搬法で学習可能なのは 2 層程度

問題点

多い層をもつニューラルネットワークの学習では様々な問題により上手く学習することが出来ない

- 単層ネットワークに分解し，入力層に近い側から順番に教師なしで学習する
- 目的とするニューラルネットワークの学習前に層ごとに学習を行うことで良い初期値パラメータを求めておくやり方を事前学習（pretraining）という

人工ニューラルネットワーク

ディープビリーフネットワーク (deep belief network, DBN) を制約ボルツマンマシンに単層ごとに分解後, それぞれ学習する手法が提案される

自己符号化器 (autoencoder)

入力に対し, 計算される出力が入力になるべく近くなる用に訓練されるニューラルネットワーク.

自己符号化器を用いて単層毎に教師なしで学習.

単層ごとに学習を行うことで初期パラメータを決定後, ニューラルネットワーク全体を学習すると上手く学習できる.

出力層の設計と誤差関数

順伝播型ネットワークが表現する関数 $\vec{y}(\vec{x}; \vec{w})$ をネットワークのパラメータ \vec{w} を変えることで変化させ、望みの関数を与える。

$$\{(\vec{x}_1, d_1), (\vec{x}_1, d_1), \dots, (\vec{x}_N, d_N)\} \quad (1)$$

これらのペア (\vec{x}, d) 1つ1つを訓練サンプル (training samples) といい、その集合を訓練データ (training data) という。学習とはネットワーク \vec{w} を調整することで訓練データの入出力ペアをできるだけ再現すること。

ネットワークが表す関数と訓練データとの近さ $(\vec{y}(\vec{x}_n; \vec{w}))$ を誤差関数 (error function) で定義する。誤差関数は問題の種別や活性化関数によって異なる。

Table: 問題の種別ごとの活性化関数と誤差関数

問題の種別	出力層の活性化関数
回帰	正接双曲線関数や恒等写像
二値分類	ロジスティック関数
多クラス分類	ソフトマックス関数

畳込みニューラルネットワークの構造

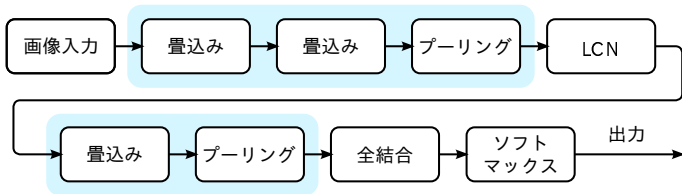
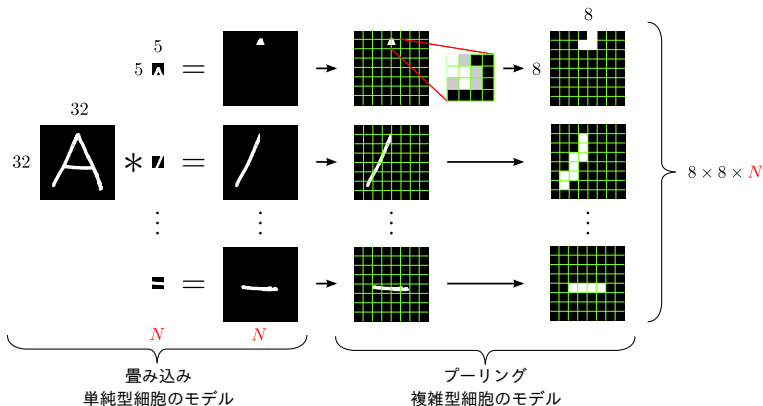


Figure: 畳込みニューラルネットワークの構造

- 畳込み層（convolution layer）とプーリング層（pooling layer）がペアでこの順に並び、このペアが複数回繰り返される
- 畳込み層とプーリング層の後に、局所コントラスト正規化（local contrast normalization, LCN）層を挿入する場合がある
- 畳込み層とプーリング層の繰り返しの後には、隣接層間ユニットが全結合した層が配置される
- 目的がクラス分類であれば最終的な出力はソフトマックスを適用する

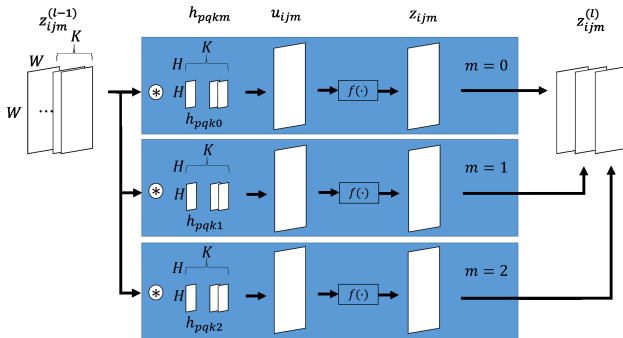
畳込みニューラルネットワークの構造



二層の働き

- 畳込み層はフィルタが表す特徴を入力から抽出する
- プーリング層は抽出された特徴の位置感度を低下させる

畳み込み層



- 各フィルタについて並列に計算され、 u_{ijm} が出力される。各チャンネルについて並列に画像とフィルタの畳み込みを行い全チャンネルにわたり加算する。

$$u_{ijm} = \sum_{p=0}^{K-1} \sum_{q=0}^{H-1} \sum_{k=0}^{H-1} z_{i+p, j+q, k}^{(l-1)} h_{pqkm} + b_{ijm} \quad (2)$$

プーリング

プーリング

プーリング層は通常、畳み込み層の直後に設置され、プーリング層は畳み込み層で抽出された特徴の位置感度を低下させる働きがある。

サイズ $W \times W \times K$ の入力画像上で画素 (i, j) を中心とする $H \times H$ の正方領域 P_{ij} をとり、 P_{ij} 内の画素についてチャネル k ごとに独立に、 H^2 個ある画素値を使って1つの画素値 u_{ijk} を求める。最大プーリングや平均プーリングなどがある。

	62	71	72	69	65	71	79	107
	73	79	80	81	79	79	98	128
	76	82	81	79	75	81	114	132
	77	85	83	77	72	99	144	145
	74	79	77	77	79	112	155	142
	74	73	71	73	89	142	162	137
	69	73	73	77	110	160	166	134
	60	67	68	78	124	154	148	116

79	81	128
85	83	155
74	124	166

Figure: プーリングの例. プーリングサイズ 3×3 , スライド $s = 3$, ゼロパディングで最大プーリングを行った場合

勾配の計算

畳み込み層の計算式

$$\mathbf{u}^{(l)} = \mathbf{W}^{(l)} \mathbf{u}^{(l-1)} + \mathbf{b}^{(l)} \quad (3)$$

$$\mathbf{z}^{(l)} = f^{(l)}(\mathbf{u}) \quad (4)$$

- 逆伝播の計算も多層パーセプトロン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

$$\mathbf{W}_{ji} = \mathbf{t}_{ji}^{\top} \mathbf{h} \quad (5)$$

- 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

$$\partial \mathbf{W} = \delta^{(l)} \mathbf{z}^{(l-1)\top} \quad (6)$$

- $\delta^{(l)}$: 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

$$(\partial \mathbf{h})_r = \sum_{i,j} (\mathbf{T}_r \odot \partial \mathbf{W})_{ji} \quad (7)$$

- \odot, \sum : 成分ごとの積, 行列の全成分の和
- \mathbf{T} : \mathbf{t}_{ji} の成分を t_{ijr} とし, r を固定したときの t_{ijr} を (j, i) 成分に持つ行列

caffe による学習と認識

- アニメキャラクターの認識（ラブライブ！... 9人のキャラクター +etc）
- アイドルグループの顔認識（少女時代のメンバー... 9人 +etc）
- 手書きひらがな文字認識（ひらがな 48 文字 + 濁音 20 文字 + 半濁音 5 文字 = 73 文字）



ラブライブ！



少女時代

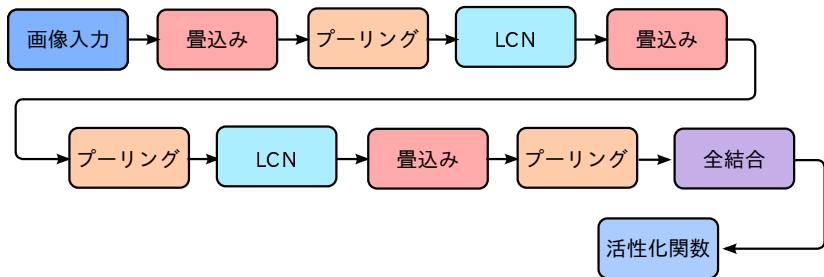


手書き文字

データセットと学習器

- アニメキャラクターの認識
各キャラクターに 2500 枚 ~4000 枚と負例約 9000 枚
- アイドルグループの顔認識（少女時代のメンバー... 9 人 +etc）
各メンバーに 1000 枚 ~2500 枚と負例約 3500 枚
- 手書きひらがな文字認識（ひらがな 48 文字 + 濁音 20 文字 + 半濁音 5 文字 = 73 文字）
各文字に 180 枚

学習器には cifar10 のフルソルバーを使用した。



学習結果（ラブライブ！）

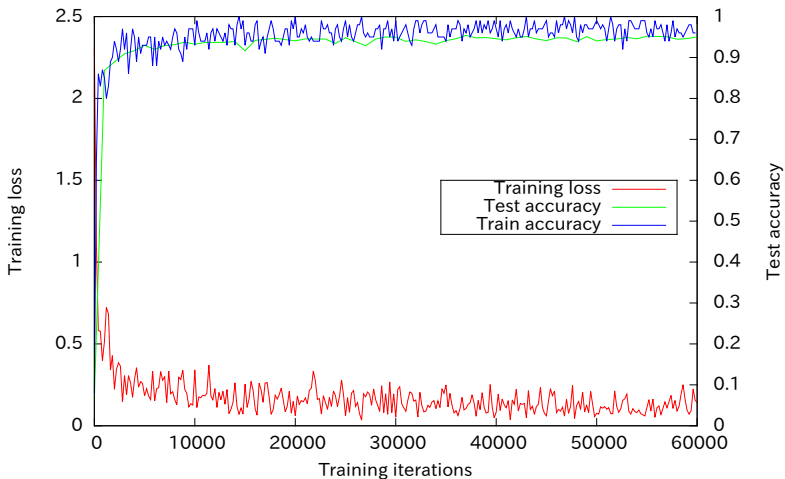


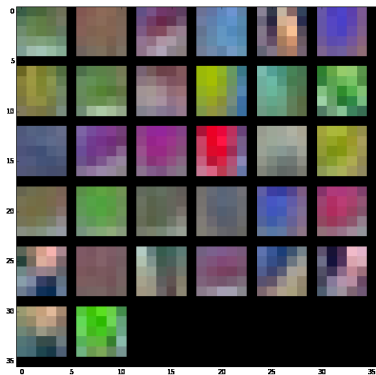
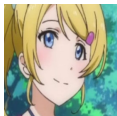
Figure: 損失関数の値と精度

識別結果

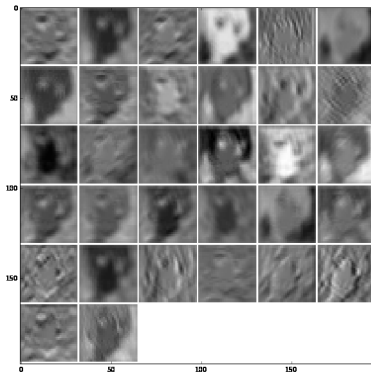


識別器を用いた識別結果

アニメキャラクター認識時の中間層出力



フィルタ



第一層目の出力

学習結果（少女時代）

ドロップアウトのユニットを Caffe の学習に取り込んだ結果

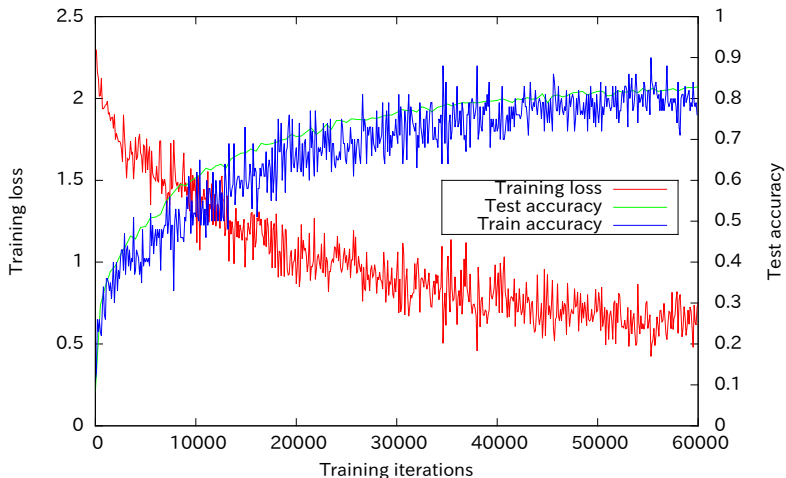
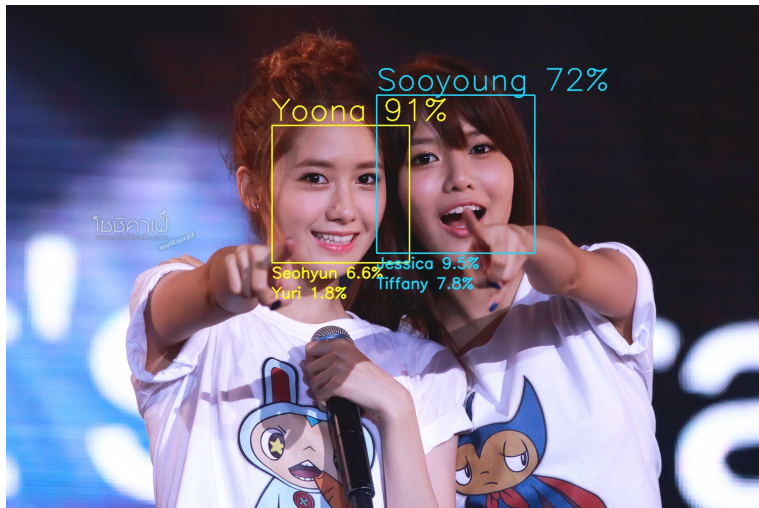


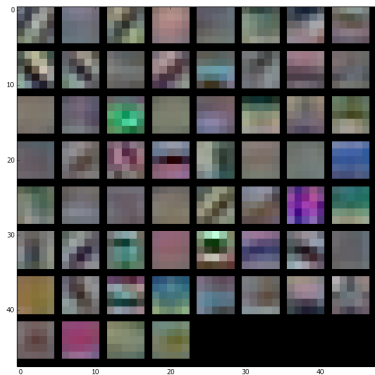
Figure: 損失関数の値と精度

識別結果

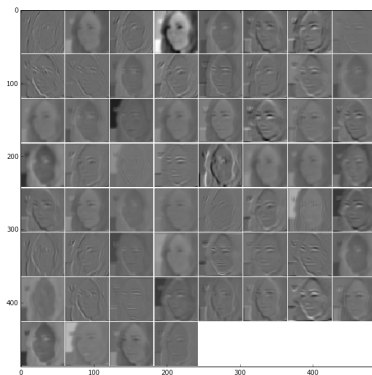


識別器を用いた識別結果

認識時の中間層出力



フィルタ



第一層目の出力

学習結果（手書き文字認識）

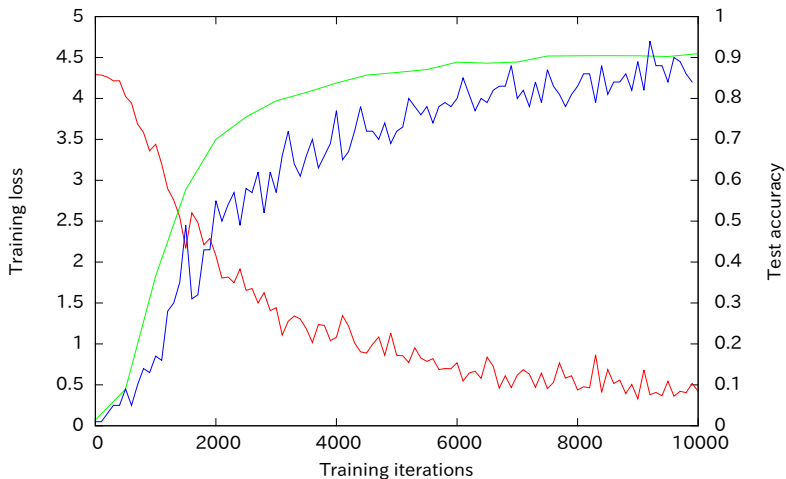


Figure: 損失関数の値と精度

識別結果

手書き文字認識用の簡単なアプリケーションの作成

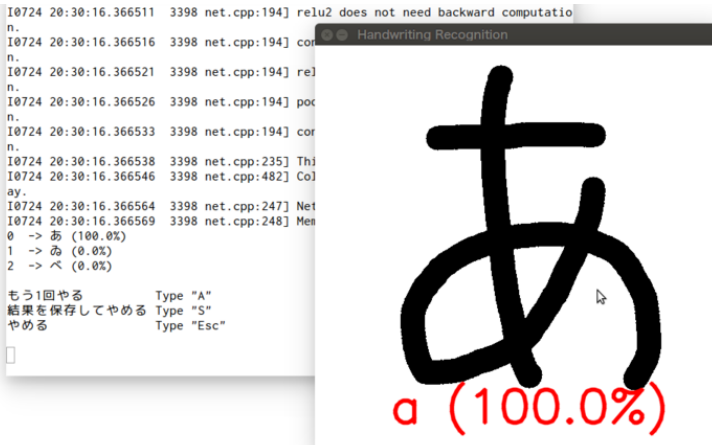


Figure: Caffe を使った手書き文字認識