

レポート課題1-1

- レポート課題の 1-2 は後日出します。
- レポートの提出は、keio.jp を使って下さい。
 - TeX, MsWord 等で作成して下さい。提出は pdf 形式でも結構です。
- 書くべき内容に関しては特には述べません。すべて常識で判断してください。
- 締め切りは、11/27(月曜日)一杯とします。
 - ただし、12/4(月)一杯まで待ちます。

課題 1-1

1. 講義のスライド中で、誤差逆伝播法の概念を図を用いて口頭で説明しました。これを式に書き直して下さい。通常の数式を用いるのを標準としますが、プログラミング言語を用いてもよいとします。
 - プログラミング言語を使って書くときには、使用している言語名と、各文・式・変数の意味を分かりやすく書いて下さい。
2. 多層ニューラルネットワークの学習を、誤差逆伝播法で行うプログラムを書いて下さい。言語は、(原則として) R を用いて下さい。事情があれば(嫌いだ、でもよい)、それを説明の上、使用して下さい。
 - なお、中間層(隠れ層)の層数は、パラメータ化する必要はなく、1層、2層、3層のプログラムを別個に作ればよいとします。
 - 学習データ・テストデータは、csvファイルから読み込むようにして下さい。

プログラミングが非常に苦手だという方は、別途相談に来て下さい。

課題 1-1 (続き)

3. そのプログラムを用いて、簡単な手書き数字識別の実験を行って下さい。
 - データは、HPにリンクのある HWO1-1-data.zip 中の optdigits.tra.csv (学習用) と optdigits.tes.csv (テスト用) を用いて下さい。管理工学実験演習(データマイニング)で用いたものと同じです。
 - 実験条件を変更して、どの程度の精度が得られるかを調べて下さい。**ただし、下に記すように10出力としたので、そのように書き換えた optdigits2.tra.csv (学習用) と optdigits2.tes.csv (テスト用) を用いてもよい。**
 - なお、以下の点に注意して下さい。
 - プログラム中で、各層に定数値(1)を出力する素子が必要です。
 - 最終段の出力素子は、10個用いて下さい。数字の種類0~9に対応するものです。
 - 誤差関数(コスト関数)は、2乗誤差で構いません。こうした問題には、多くの場合、クロスエントロピーの方が、よいとされますので、クロスエントロピーを用いても結構です。
 - 活性化関数は シグモイド関数を用いて下さい。
 - 隠れ層は、1層、2層、3層で実験を行って下さい。
 - 学習の加速は行っても行わなくても結構です。

R の neuralnet パッケージを用いた例を記します。

```
# サンプルプログラム
library(neuralnet)
setwd("E:/kadai/")

# 列の名称は、x1,...,x64, y0,...,y9 for dtrain2 and dtest2
# V1,...,V64, V65 for dtrain and dtest
dtrain2 <- read.csv("optdigits2.tra.csv", header=T)
dtrain <- read.csv("optdigits.tra.csv", header=F)
dtest2 <- read.csv("optdigits2.tes.csv", header=T)
dtest <- read.csv("optdigits.tes.csv", header=F)

# 各行、最大値を調べその場所(列番号)を返す関数
rowWhichmax <- function(x) apply(x, 1, which.max)

# f1 = "y0+y1+ ... +y8+y9 ~ x1+x2+ ... +x63+x64"
f1 <- paste(paste(paste("y", 0:9, sep=""), collapse="+"), "~",
            paste(paste("x", 1:64, sep=""), collapse="+"))

# 学習データを全部(3823個)用いると速い(しかし精度はよい)
# m <- neuralnet(formula=f1, data=dtrain2, hidden=20, err.fct="ce", threshold=0.1, linear.output=F)
# 最初の1000個のみ使って学習
m <- neuralnet(formula=f1, data=dtrain2[1:1000,], hidden=20, err.fct="ce", threshold=0.1, linear.output=F)

# network図。文字や線が重なってしまい、残念ながら、読めない
plot(m)

# performance on training dataset
# 多くの機械学習関数では predict を用意するが neuralnet は compute
predValues <- compute(m, dtrain2[,1:64])
# 最大値を身える出力素子を知る
pred <- rowWhichmax(predValues$net.result)
tbl <- table(dtrain[,65], pred)
print(tbl)
# accuracy
sum(diag(tbl))/sum(tbl)

# performance on test dataset
predValues <- compute(m, dtest[,1:64])
pred <- rowWhichmax(predValues$net.result)
tbl <- table(dtest[,65], pred)
print(tbl)
# accuracy
sum(diag(tbl))/sum(tbl)
```