1.4.1 誤り訂正学習

手書き文字(digit)認識をします。データは下記からダウンロードしてください。
http://www.kaggle.com/c/digit-recognizer (http://www.kaggle.com/c/digit-recognizer)

```
setwd("C:/Users/k-harada/Desktop/kdd/digit/ORG")
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.1.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.1.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
traindata <- fread("train.csv")
```

```
##
Read 0.0% of 42000 rows
Read 47.6% of 42000 rows
Read 95.2% of 42000 rows
Read 42000 rows and 785 (of 785) columns from 0.072 GB file in 00:00:07
```

```
testdata <- fread("test.csv")
```

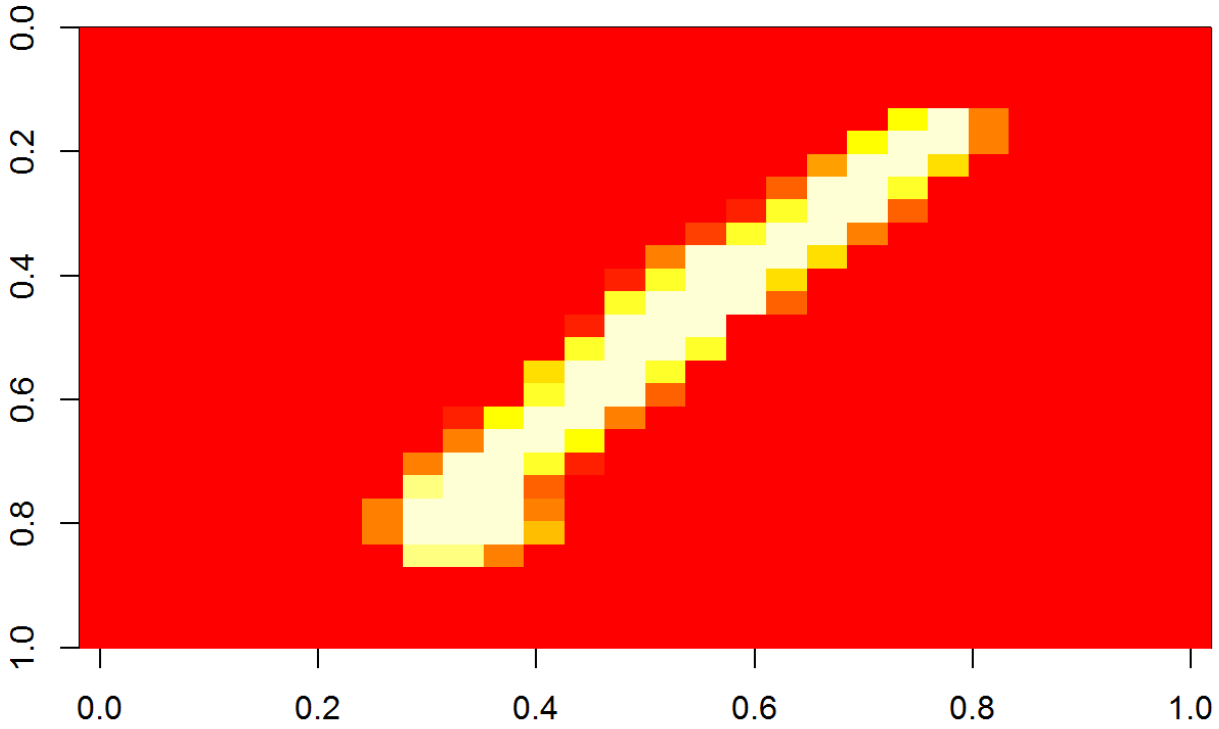
```
##
Read 71.4% of 28000 rows
Read 28000 rows and 784 (of 784) columns from 0.048 GB file in 00:00:03
```

```
train_label <- traindata$label[1:28000]
valid_label <- traindata$label[28001:42000]

# transform into array
train_array <- array(as.matrix(traindata[1:28000, ])[,-1], dim=c(28000,28,28))
valid_array <- array(as.matrix(traindata[28001:42000, ])[,-1], dim=c(14000,28,28))
```

image(行列)で画像っぽくPlotに出力されます。適宜チェックしてください。

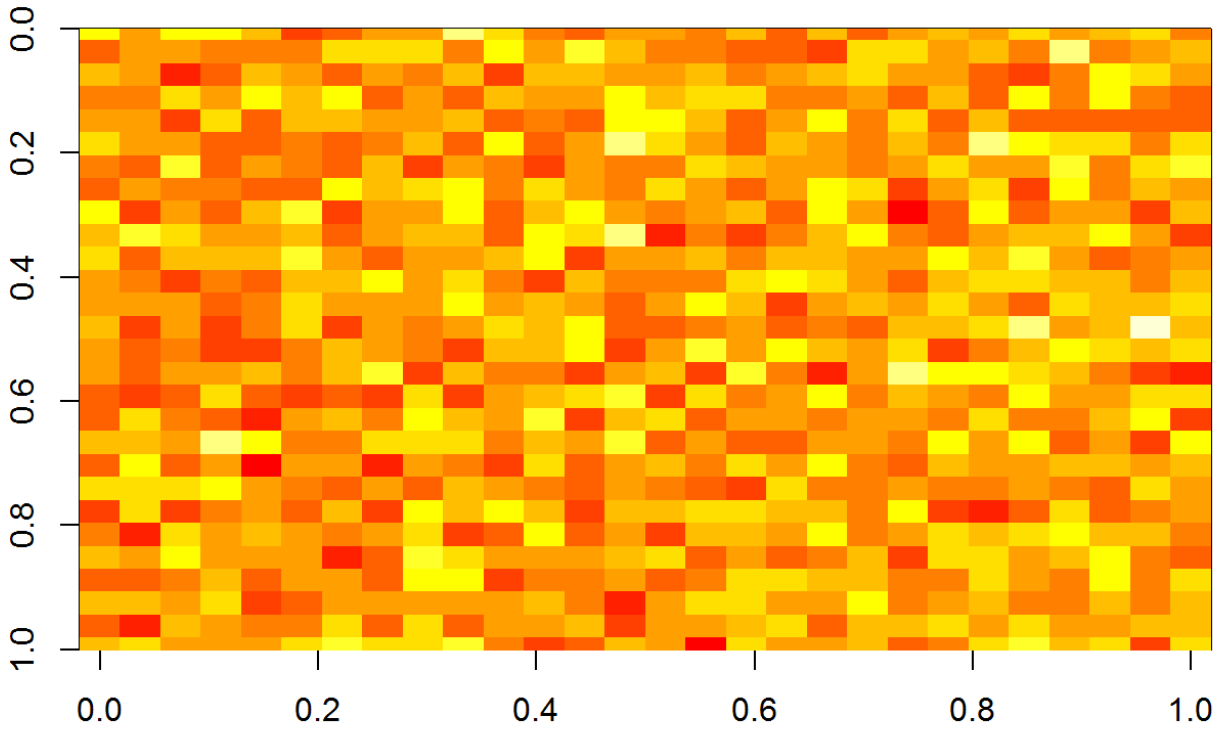
```
# check
image(train_array[1,,], ylim=c(1, 0))
```




```
# image(train_array[2,,], ylim=c(1, 0))
# image(train_array[3,,], ylim=c(1, 0))
# image(train_array[4,,], ylim=c(1, 0))
# image(train_array[5,,], ylim=c(1, 0))
# image(train_array[6,,], ylim=c(1, 0))
# image(train_array[7,,], ylim=c(1, 0))
# image(train_array[8,,], ylim=c(1, 0))
# image(train_array[9,,], ylim=c(1, 0))
```

0-9のそれぞれに対応する学習器を作ります。
まず0をやってみます。

```
# 初期条件
set.seed(0)
w0 <- matrix(rnorm(28*28), nrow=28)
b0 <- 0
image(w0, ylim=c(1, 0))
```

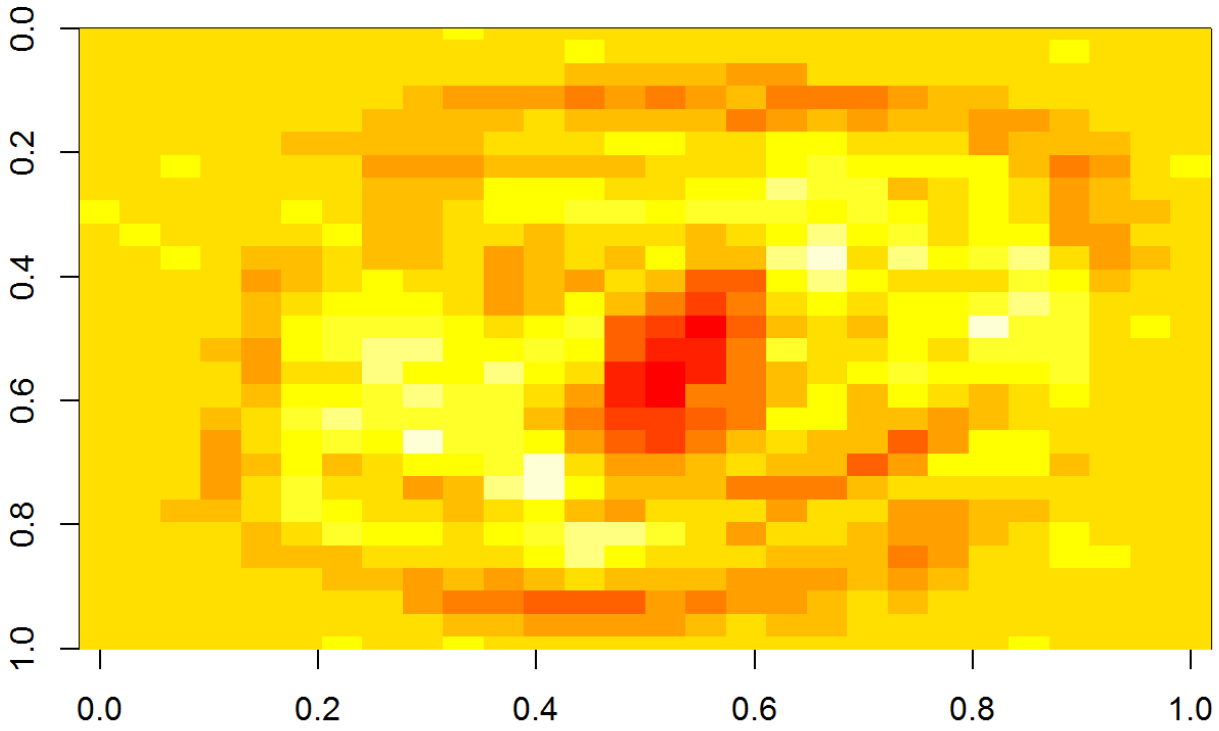


学習率を設定して、誤り訂正学習をします。

```
eta <- 0.01
for (i in seq(28000)){
  y <- ifelse(sum(w0 * train_array[i,,]) + b0 >= 0, 1, 0)
  t <- ifelse(train_label[i] == 0, 1, 0)
  if (y == t){
    next
  } else {
    w0 <- w0 + eta * (t - y) * train_array[i,,]
    b0 <- b0 + eta * (t - y) * 1
  }
}
```

結果を見てみます。

```
y0 <- rep(0, 28000)
t0 <- rep(0, 28000)
for (i in seq(28000)){
  y0[i] <- ifelse(sum(w0 * train_array[i,]) + b0 >= 0, 1, 0)
  t0[i] <- ifelse(train_label[i] == 0, 1, 0)
}
image(w0, ylim=c(1, 0))
```

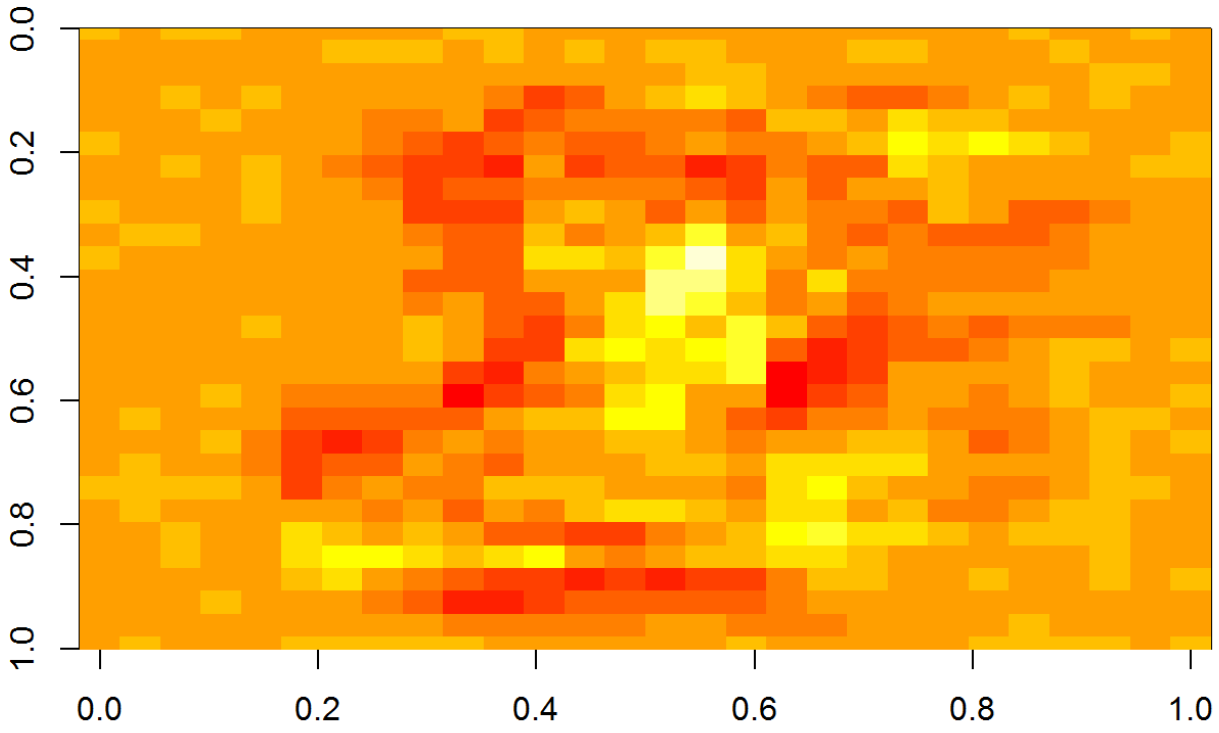



```
# y0:出力
# t0:こたえ
table(y0, t0)
```

```
##    t0
## y0      0     1
##   0 25207   296
##   1    72  2425
```

1-9についてもそれぞれやってみます

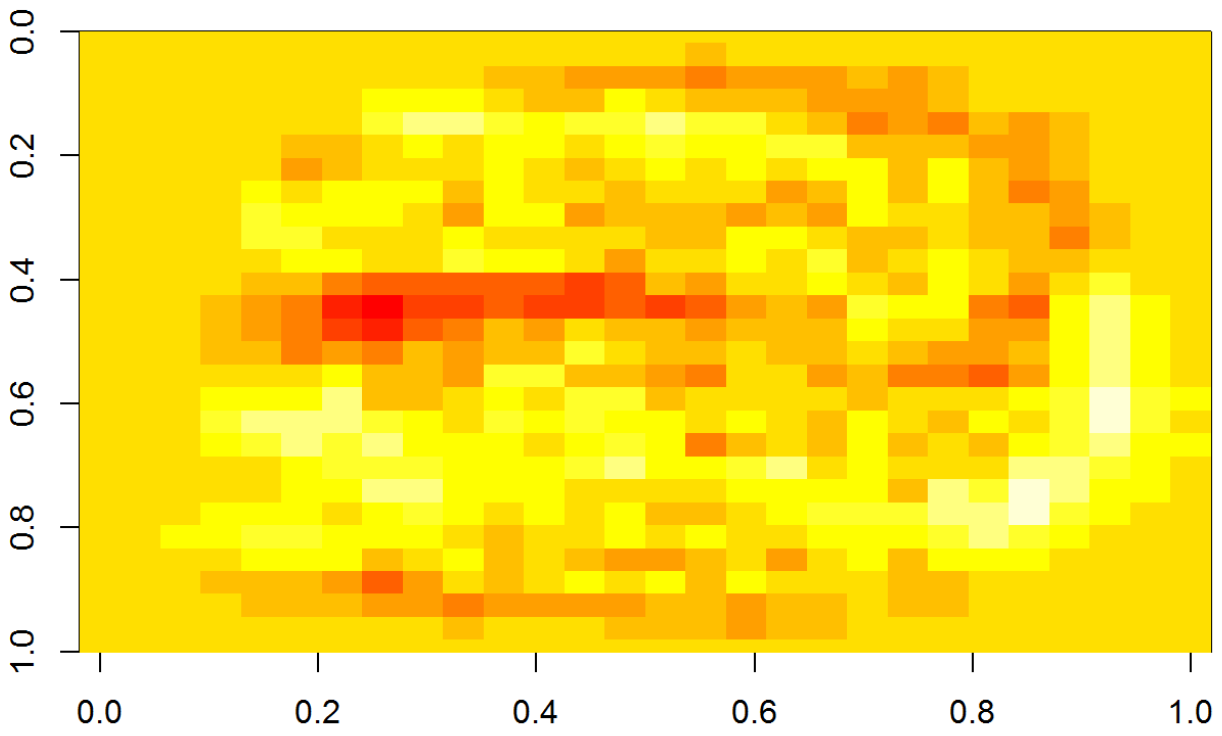
```
image(w1, ylim=c(1, 0))
```




```
# y1:出力
# t1:こたえ
table(y1, t1)
```

```
##    t1
## y1      0     1
##   0 24766   101
##   1   127  3006
```

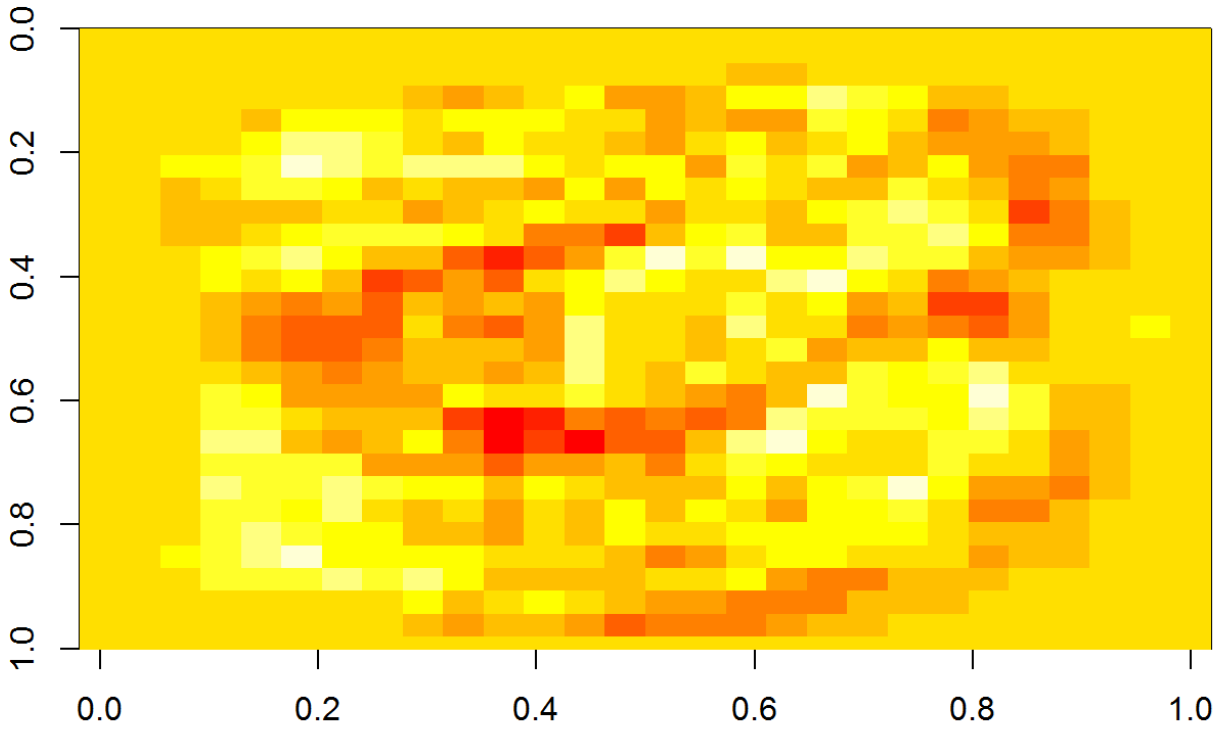
```
image(w2, ylim=c(1, 0))
```




```
# y2:出力
# t2:こたえ
table(y2, t2)
```

```
##    t2
## y2      0     1
##   0 25009   523
##   1   167  2301
```

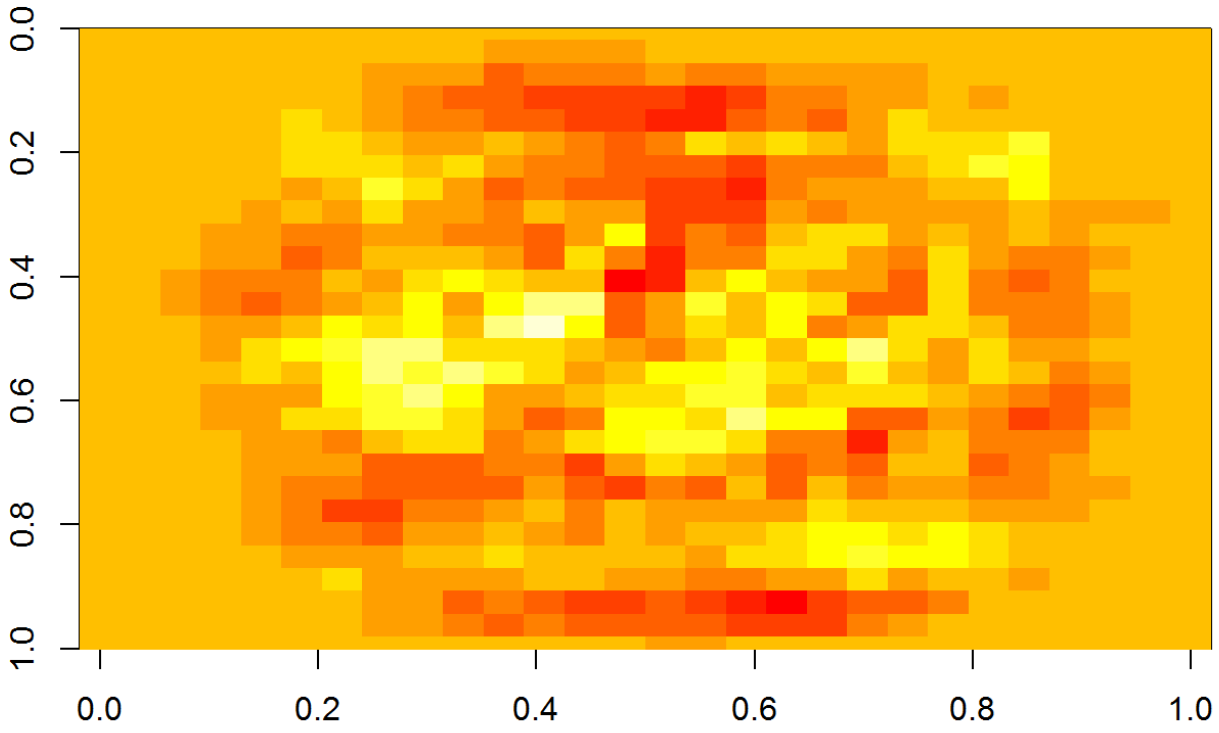
```
image(w3, ylim=c(1, 0))
```




```
# y3:出力
# t3:こたえ
table(y3, t3)
```

```
##    t3
## y3     0    1
##   0 24797  764
##   1   294 2145
```

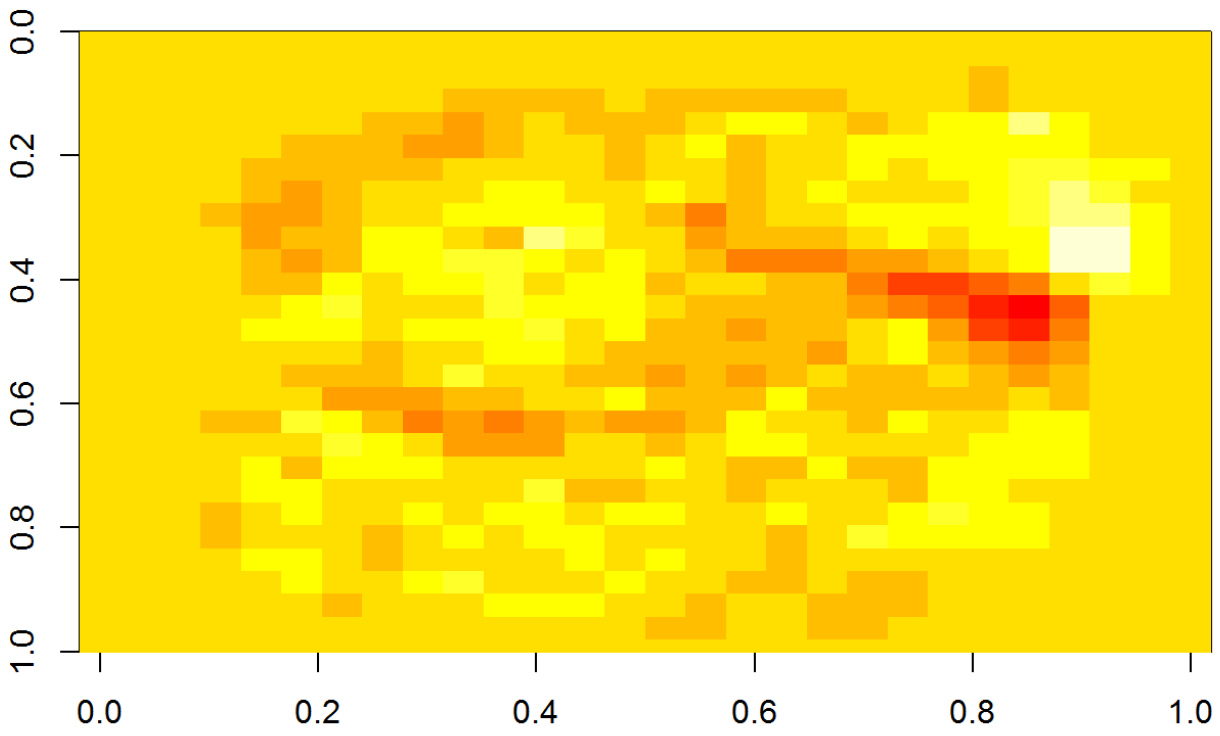
```
image(w4, ylim=c(1, 0))
```




```
# y4:出力
# t4:こたえ
table(y4, t4)
```

```
##    t4
## y4      0     1
##   0 25126   590
##   1   118  2166
```

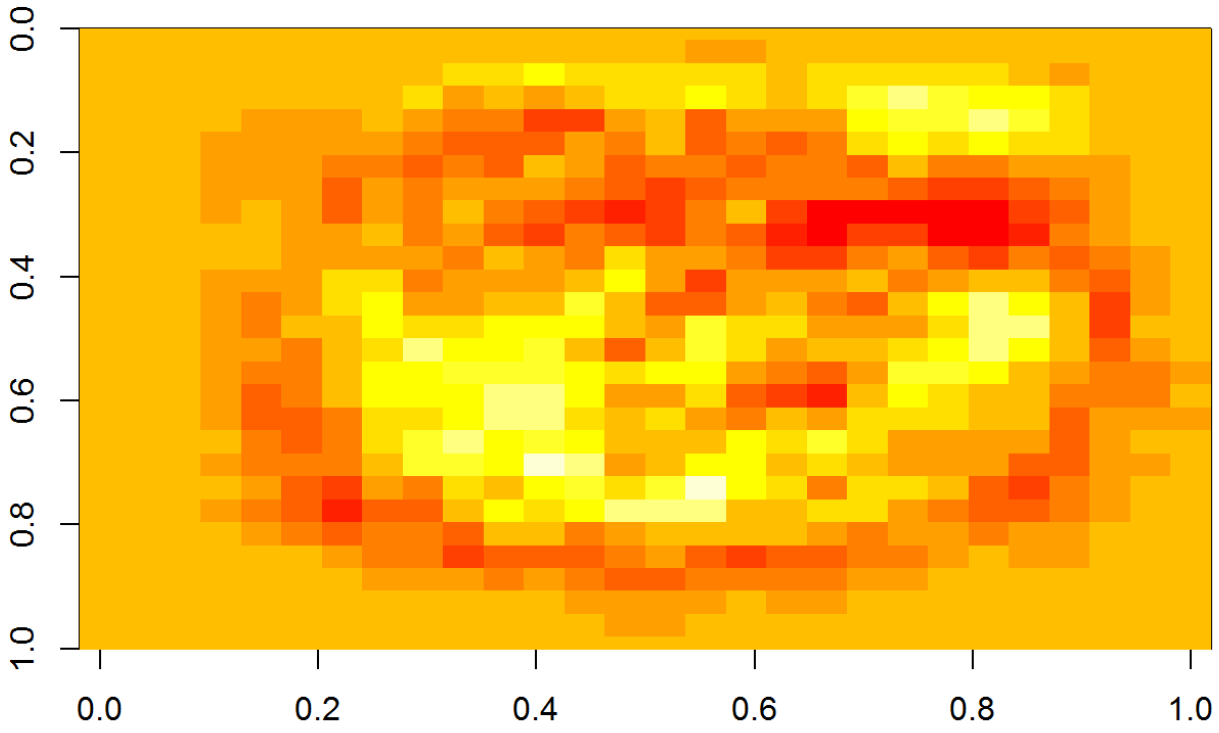
```
image(w5, ylim=c(1, 0))
```




```
# y5:出力
# t5:こたえ
table(y5, t5)
```

```
##    t5
## y5      0     1
##   0 25443  1187
##   1    36  1334
```

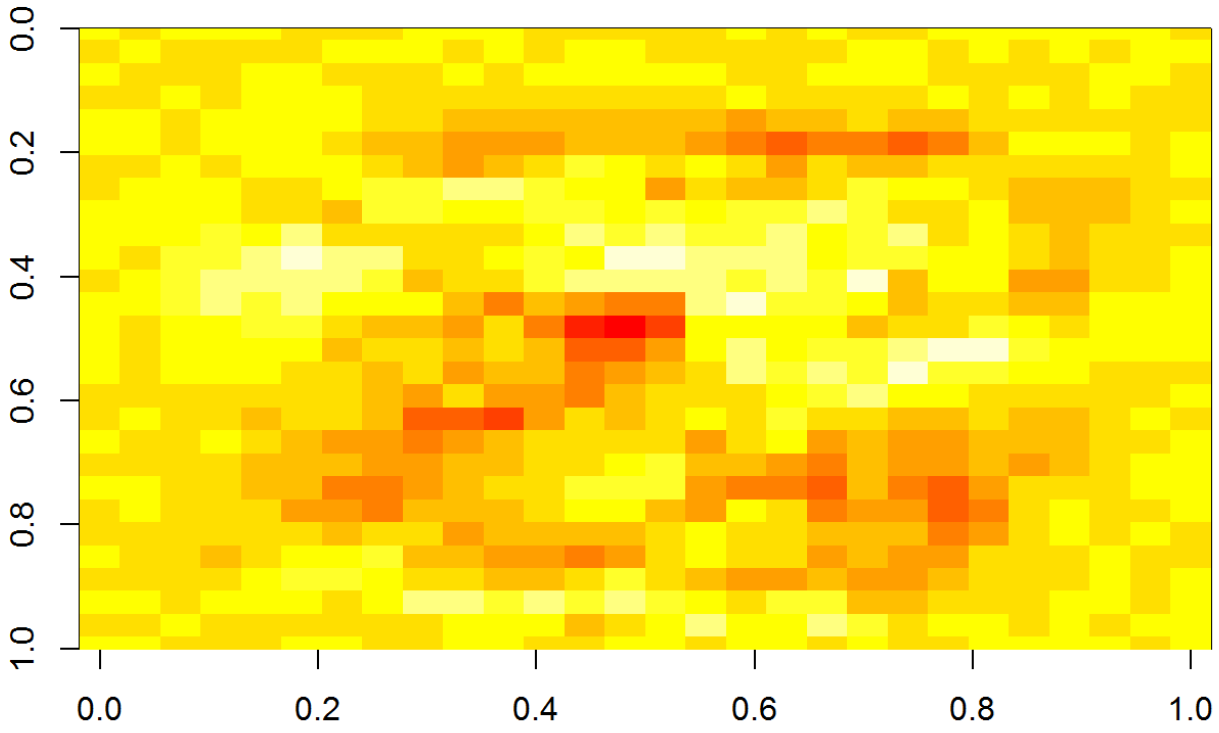
```
image(w6, ylim=c(1, 0))
```




```
# y6:出力
# t6:こたえ
table(y6, t6)
```

```
##    t6
## y6      0     1
##   0 24702   131
##   1   519  2648
```

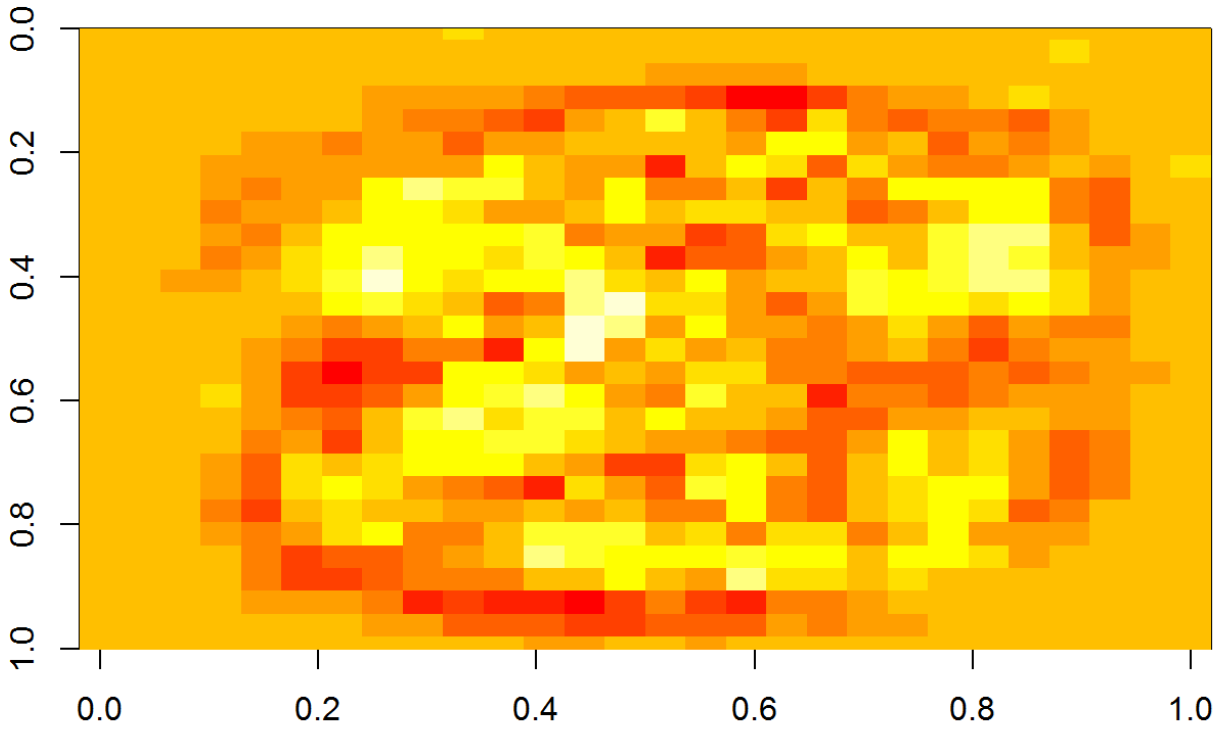
```
image(w7, ylim=c(1, 0))
```




```
# y7:出力
# t7:こたえ
table(y7, t7)
```

```
##    t7
## y7      0     1
##   0 24793   273
##   1   283  2651
```

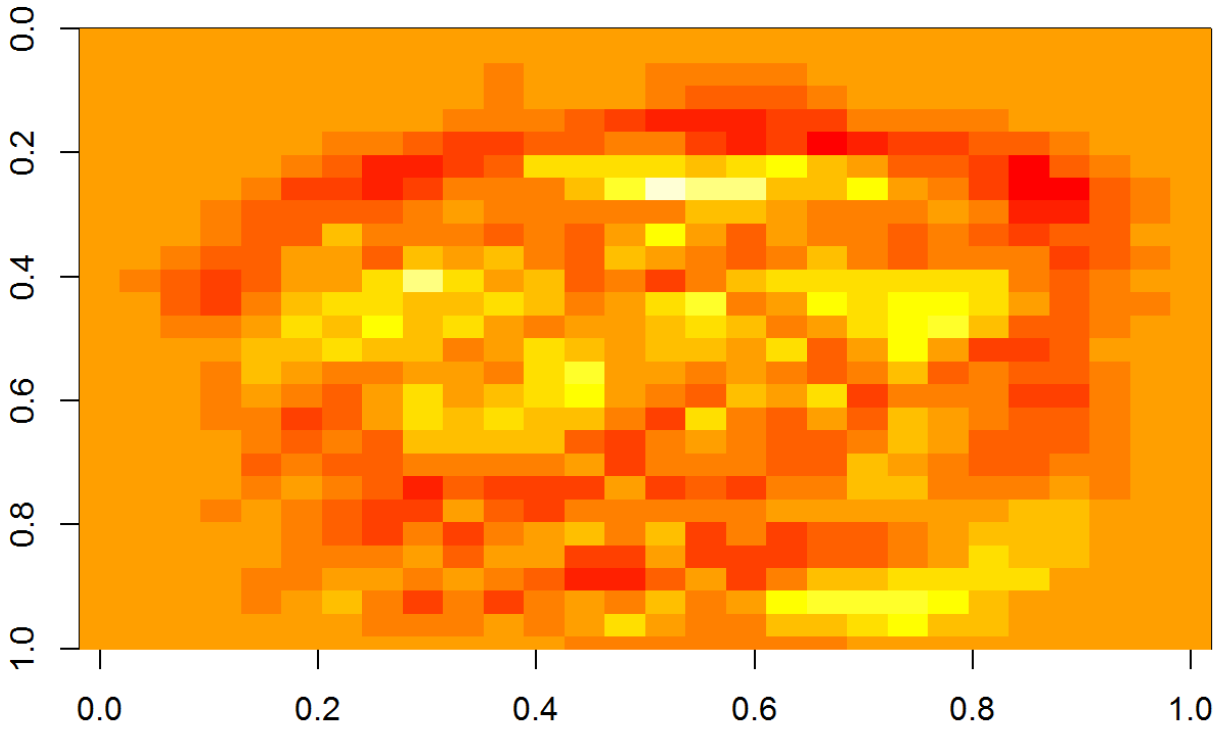
```
image(w8, ylim=c(1, 0))
```




```
# y8:出力
# t8:こたえ
table(y8, t8)
```

```
##    t8
## y8      0     1
##   0 23508   795
##   1  1798  1899
```

```
image(w9, ylim=c(1, 0))
```




```
# y9:出力
# t9:こたえ
table(y9, t9)
```

```
##    t9
## y9      0     1
##   0 24367   709
##   1   868  2056
```

合計の精度を確認

ここでは一番強く反応したものを正解とする

学習用データ

```
yall <- matrix(rep(0, 280000), ncol=10)
for (i in seq(28000)){
  yall[i, 1] <- sum(w0 * train_array[i,,]) + b0
  yall[i, 2] <- sum(w1 * train_array[i,,]) + b1
  yall[i, 3] <- sum(w2 * train_array[i,,]) + b2
  yall[i, 4] <- sum(w3 * train_array[i,,]) + b3
  yall[i, 5] <- sum(w4 * train_array[i,,]) + b4
  yall[i, 6] <- sum(w5 * train_array[i,,]) + b5
  yall[i, 7] <- sum(w6 * train_array[i,,]) + b6
  yall[i, 8] <- sum(w7 * train_array[i,,]) + b7
  yall[i, 9] <- sum(w8 * train_array[i,,]) + b8
  yall[i,10] <- sum(w9 * train_array[i,,]) + b9
}
yall_res <- max.col(yall) - 1

table(yall_res, train_label)
```

```
##         train_label
## yall_res    0    1    2    3    4    5    6    7    8    9
##        0 2551    0    5    2    6   13    9   11   11   11
##        1    1 3001   38   14   11   22    3   14   43   19
##        2   16   12 2393   88   12   27    6   24   23    4
##        3   10    2   52 2457   12  175    4   18   51   28
##        4    2    1   26    4 2208   30    5    5    4   33
##        5    8    2    5   25    0 1625   11    1   13    9
##        6   44    7   95   30   53   80 2701    7   29    2
##        7    4    8   38   22   17   17    4 2716   14  200
##        8   73   71  123  207  186  448   34   40 2415  107
##        9   12    3   49   60  251   84    2   88   91 2352
```

```
mean(yall_res == train_label)
```

```
## [1] 0.8721071
```

検証用データ

```
yall_valid <- matrix(rep(0, 140000), ncol=10)
for (i in seq(14000)){
  yall_valid[i, 1] <- sum(w0 * valid_array[i,,]) + b0
  yall_valid[i, 2] <- sum(w1 * valid_array[i,,]) + b1
  yall_valid[i, 3] <- sum(w2 * valid_array[i,,]) + b2
  yall_valid[i, 4] <- sum(w3 * valid_array[i,,]) + b3
  yall_valid[i, 5] <- sum(w4 * valid_array[i,,]) + b4
  yall_valid[i, 6] <- sum(w5 * valid_array[i,,]) + b5
  yall_valid[i, 7] <- sum(w6 * valid_array[i,,]) + b6
  yall_valid[i, 8] <- sum(w7 * valid_array[i,,]) + b7
  yall_valid[i, 9] <- sum(w8 * valid_array[i,,]) + b8
  yall_valid[i,10] <- sum(w9 * valid_array[i,,]) + b9
}
yall_valid_res <- max.col(yall_valid) - 1

table(yall_valid_res, valid_label)
```

```
##              valid_label
## yall_valid_res    0    1    2    3    4    5    6    7    8    9
##              0 1317    0    6    1    1    8    7    8    1    6
##              1    0 1518   12   10    8   11    2    2   39    7
##              2    5    5 1139   44    8   16    3   16   12    4
##              3    6    3   31 1205    6  103    2    7   18   17
##              4    2    1    7    0 1018   14    3    2    3   20
##              5    2    0    3   13    2  773    7    0    5    6
##              6   28    4   45   17   39   40 1313    2   13    2
##              7    4    3   21   17    9    8    1 1362    7  108
##              8   42   41   65   91  110  255   18   31 1223   48
##              9    5    2   24   44  115   46    2   47   48 1205
```

```
mean(yall_valid_res == valid_label)
```

```
## [1] 0.8623571
```