Introduction to data science & artificial intelligence (INF7100)

Arthur Charpentier

#371 Pictures

été 2020

Neural Nets for Pictures

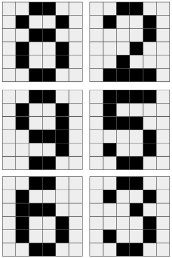
Very popular for pictures...

Picture $\boldsymbol{x}_i$ is

- a $n \times n$ matrix in $\{0,1\}^{n^2}$ for black & white
- a $n \times n$ matrix in $[0,1]^{n^2}$ for grey-scale
- a $3 \times n \times n$ array in $([0,1]^3)^{n^2}$ for color
- a $T \times 3 \times n \times n$ tensor in $(([0,1]^3)^T)^{n^2}$ for video

y here is the label ("8", "9", "6", etc)

Suppose we want to recognize a "6" on a picture



$$m(\boldsymbol{x}) = \begin{cases} +1 \text{ if } \boldsymbol{x} \text{ is a "6"} \\ -1 \text{ otherwise} \end{cases}$$

Handwritting Classification

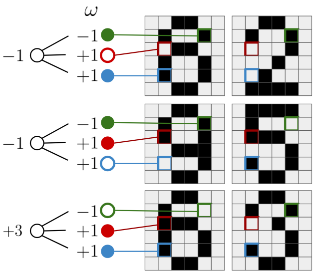
Consider some specifics pixels, and associate weights ω such that

$$\widehat{m}(\boldsymbol{x}) = \text{sign}\left(\sum_{i,j} \omega_{i,j} x_{i,j}\right)$$

where

$$x_{i,j} = \begin{cases} +1 \text{ if pixel } x_{i,j} \text{ is black} & \blacksquare \\ -1 \text{ if pixel } x_{i,j} \text{ is white} & \square \end{cases}$$

for some weights $\omega_{i,j}$
(that can be negative...)

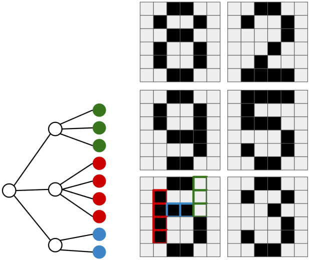


Handwritting Classification

A deep network is a network with a lot of layers

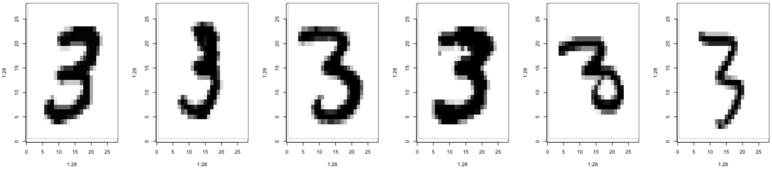
$$\widehat{m}(\boldsymbol{x}) = \text{sign}\left(\sum_i \omega_i \widehat{m}_i(\boldsymbol{x})\right)$$

$\widehat{m}_i$'s are outputs of previous neural nets
layers can capture shapes in some areas
nonlinearities, cross-dependence, etc



Classification... on Pictures

Here $\{(y_i, \boldsymbol{x}_i)\}$ with $y_i =$ "3" and $\boldsymbol{x}_i \in [0,1]^{28\times 28}$



Can it recognize '1' ?

```
> reg=glm((y==1)~.,data=df,family=binomial)
Warning messages:
1: glm.fit: |\aftergroup\redcolor|algorithm did not
    converge |\aftergroup\blackcolor|
2: glm.fit: fitted probabilities numerically 0 or 1
    occurred
```

Numerical issues..!

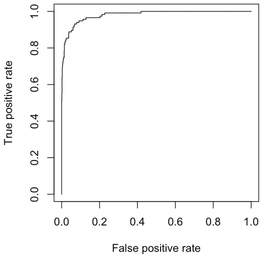
We have $k = 784$ features... it is necessary to reduce dimension !

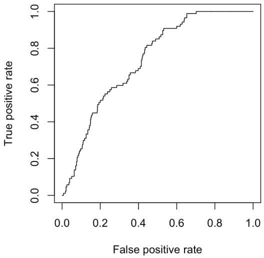
Classification... on Pictures

Let us try to recognize ‘1’ and ‘8’

```
> library(factoextra)
> pca=prcomp(MV)
> res.ind = get_pca_ind(pca)
> PTS = res.ind$coord
> k=3
> dfpca = data.frame(y=mnist$train$y
    [1:n],x=PTS[,1:k])
> reg1 = glm((y==1)~.,data=dfpca,
    family=binomial)
> reg8 = glm((y==8)~.,data=dfpca,
    family=binomial)
```

use only the first three components





Alzeihmer Detection

See Ewen Gallic's tutorial