29/10/18

Lec-21

- Multi-class classification

One Vs rest

Eg → MNIST

0 vs 9 else

1 vs else

:

↓ If one of them, says 1, then no problem

↓ If 2 say, use some confidence measure

- Pairwise-classifiers:-

$^nC_2$ classifiers

↳ Majority voting

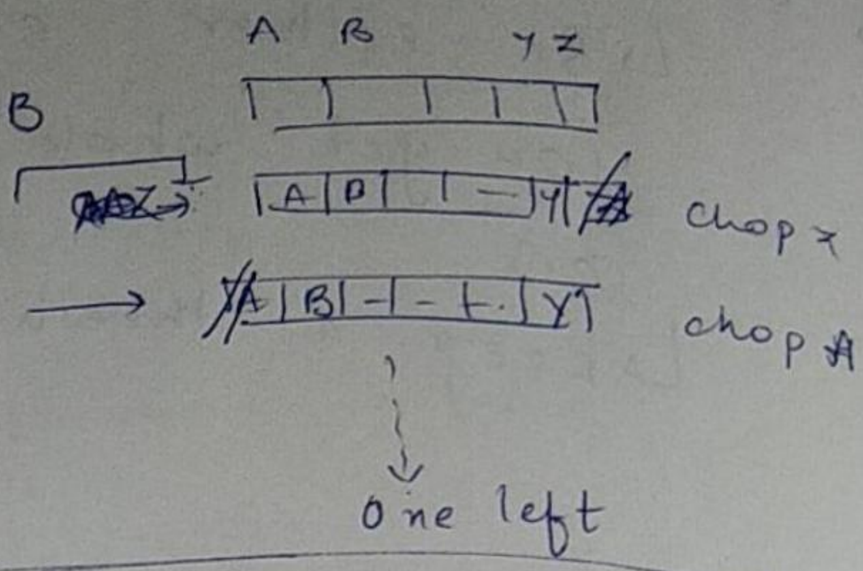
AB → A
AC → C
BC → C

⇒ Winner = C

Tree - Based

AB → classifer → output A

⇒ The point is definitely not B



Current story

Classifiers :

Many are there

can inputs be combined to give output?

$h_1(X)$, $h_2(X)$, $h_3(X)$ → Different answers

Bayesian → 92%

Multi SVM → 90%

Decision tree → 88%

One soln

⇒ Find the best classifier, by finding error-rates and choose the best

Can we find anything good out of the not so good classifiers ???

- Multiple classifiers → find the best answer

① Majority Vote

↳ can cause ties too

eg: 3 { A, B, C } so tie

② weighted voting

y_k // class output by each classifier , ξ_k // Error rate

$y_{final} = f(y_i, \xi_i)$

$= \begin{cases} \frac{\sum(1-\xi_i) y_i}{\sum(1-\xi_i)} > 0.5 \Rightarrow \text{class } 1 \\ < 0.5 \Rightarrow \text{class } 0 \end{cases}$

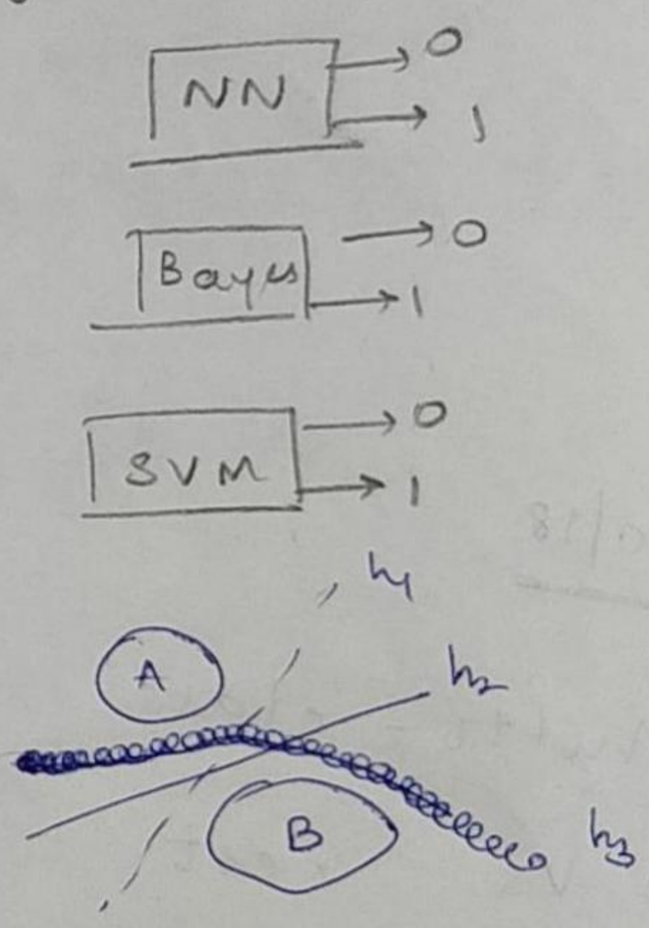
$\sum(1-\xi_i) y_i - \frac{1}{2}\sum(1-\xi_i) > 0 \Rightarrow c1$

$< 0 \quad c2$

$= \boxed{\text{floor}\left(\frac{\quad}{\quad}\right) + 1}$

↓

a round fn instead of if.



- Errors that h_2 makes $\neq$ errors h_1 makes, (eventho' $\varepsilon_1 > \varepsilon_2$)
  Because, they are made different types of errors //.

- Sample specific confidence measure:
  NN — directly give

$$y_k \rightarrow \underset{\text{confidence}}{c_k}$$

$$\frac{y_k c_k}{\Sigma c_k}$$

---

② Feature fusion :-
Assuming the classifiers use different features
~~Each~~ ~~positions~~
find a combined vec and classify - -

No use of this //

- We can learn, the fn $f \rightarrow$ called STACKING

$$y = f(y_k, c_k)$$

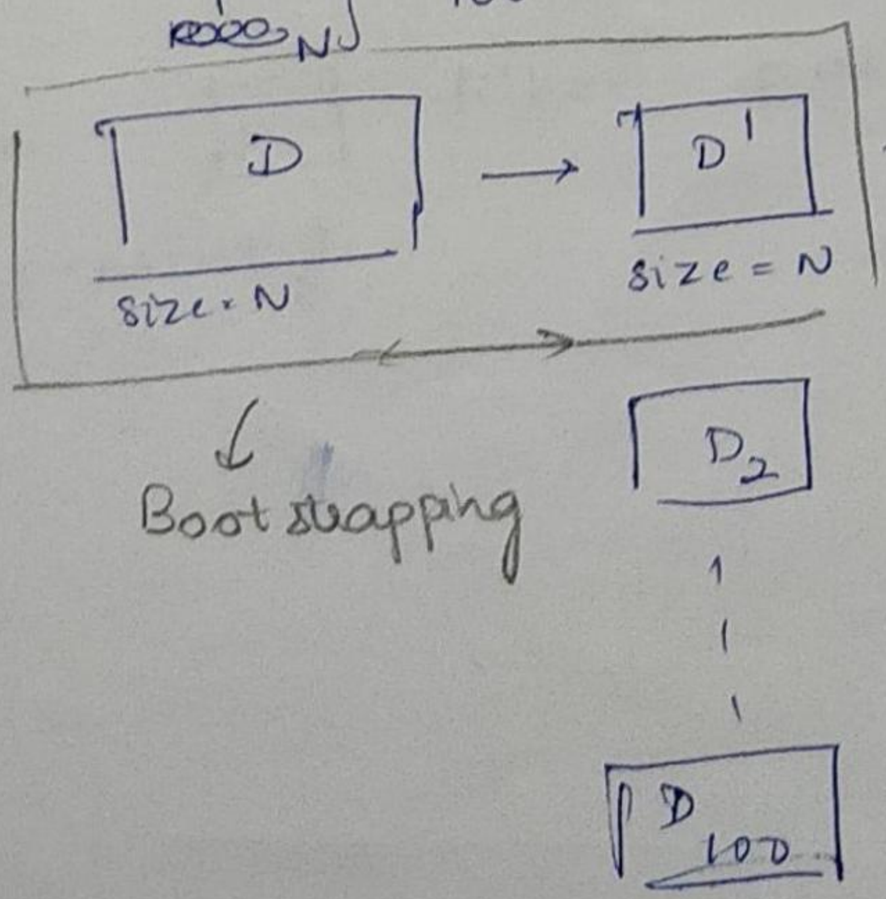
- Tweaking classifier params for better advantage

Strong $\rightarrow$ ~90% accuracy
weak $\rightarrow$ ~50% accuracy
If all classifiers $\frac{1}{2}$ → everything, no improv
If $\frac{1}{2} + \delta$ → Improvement

Ensemble methods

---

i> Bootstrap aggregation
- Sample $\{N\}$ → 100 times → with replacement



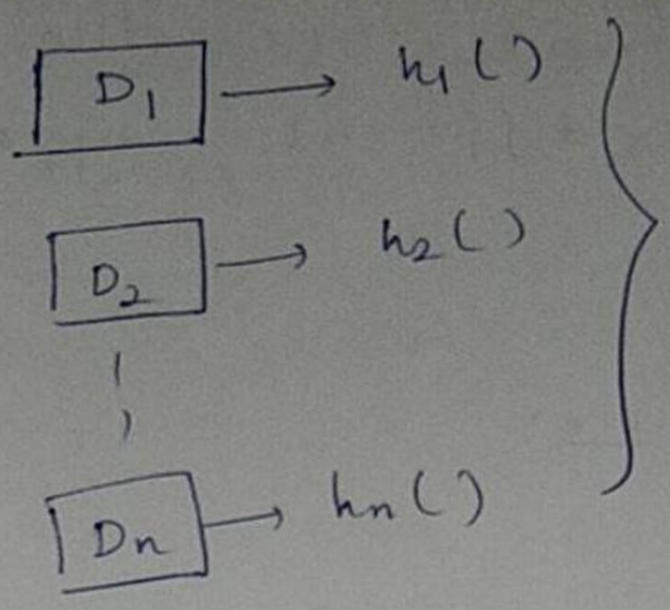
→ Some eles $\in D$ and repeat
Some are not in D' which belong to D.

Bootstrapping

[It was found that when $N' < N \Rightarrow$ very high variability in D_1, D_2, D_3 - -]

and

$N' (100N) >> N \Rightarrow$ very $D' \approx D$ distros //



→ all this is to create variation in datas

all hi's are generally same.

⇒ Boosting:

Non-uniform sampling

~~$\delta_i(x)$~~

$D_1(x_i) = \frac{1}{n}$ for all x_i

$\delta_1(x_i)$ → associate a weight to each x_i

$\delta_1(x) = \frac{1}{n}$

$\delta_2(x) = \begin{cases} e^{-d_i} \longrightarrow \text{correctly classified} \\ e^{\alpha_i} \longrightarrow \text{Incorrectly classified} \end{cases}$

Ada(ptive) - Boost //

D_1

D_2

⋮

D_{100} → Badly classified probs [An expert]

↳ Highly specialized in a small region

Now, each classifier is given weight

$\frac{1-\varepsilon_i}{\varepsilon_i}$ If $\varepsilon_i \sim 0$ (D_{100})

$\frac{1-0}{0} \sim \infty$

$\log(\infty) \rightarrow$ High wt

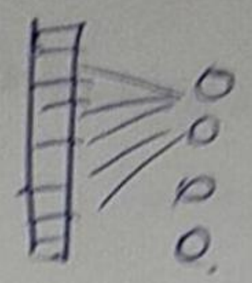
$out = \sum \alpha_t h_t(x)$

$h_t(x) = \pm 1$, not (0, 1)

If $\begin{cases} out > 0 \rightarrow \text{class } 1 \\ out < 0 \rightarrow \text{class } 0 \end{cases}$

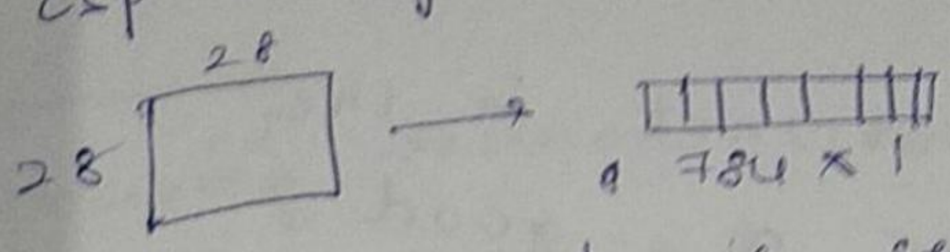
Cross entropy — Softmax + NLL

- SimpleMLP
  Adam optimizer



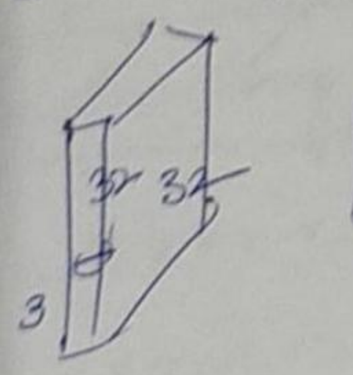
- CNN

Esp. Image info



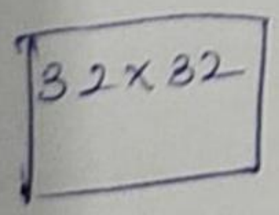
Each input is sent to all neurons.
So spatial info is lost //

CNN operates over Volume

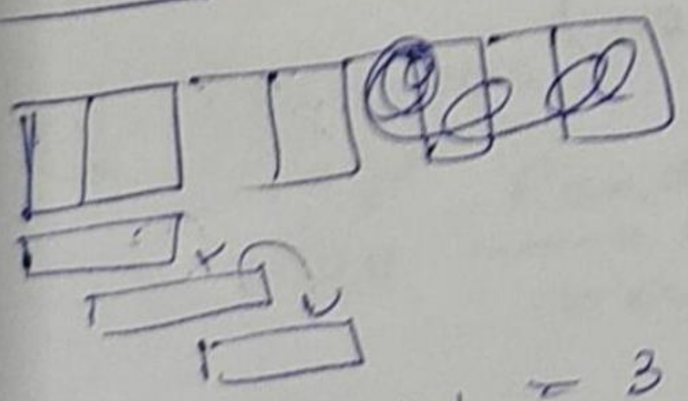


→ Neurons × filters.

- Less number of params.



So ⇒ $\frac{N-F}{Stride} + 1$

- 

⇒ padding is done

when filter passes, it sees how much line is present //