Assignment3

Foundations of Machine Learning

IIT-Hyderabad

Aug-Dec 2021

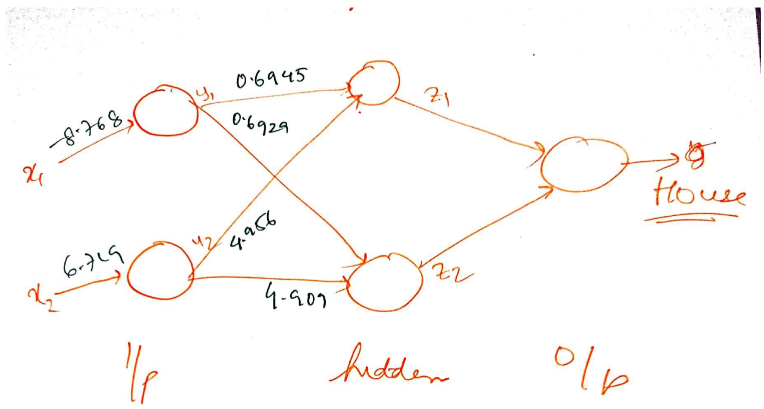
Submitted by:

Ankita Jain

BM21MTECH14001

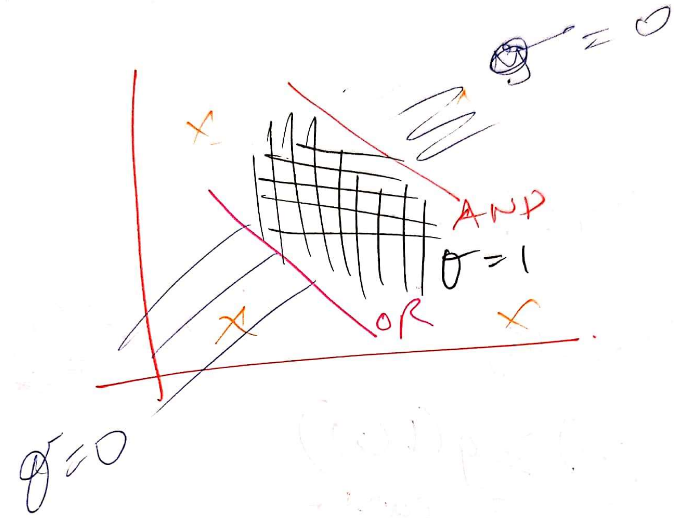
Questions: Theory

1a) Neural Networks (XOR)



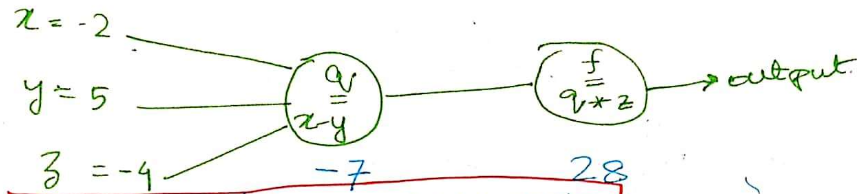
Working :

x_1	x_2	y_1 (And)	y_2 (OR)	z_1 (XOR Add)	z_2	$\hat{y}$
0	0	0	0	0	0	0
0	1	0	1	0	1	1
1	0	0	1	1	0	1
1	1	1	1	1	1	0



1b)

Graphical representation



Gradient of 'f' w.r.t. x, y and z

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial x} = z * 1 = \boxed{z = \frac{\partial f}{\partial x}}$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} * \frac{\partial q}{\partial y} = z * (-1) = \boxed{-z = \frac{\partial f}{\partial y}}$$

$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial z} = q = \boxed{x - y = \frac{\partial f}{\partial z}}$$

3.

Given: $E_{AV} = \frac{1}{M}\sum_m E_x\left[(y_m(x) - f(x))^2\right]$

$E_{ENS} = E_x\left[\left(\frac{1}{M}\sum_m y_m(x) - f(x)\right)^2\right]$

$f(x) = x^2$

To prove: $E_{ENS} \leq E_{AV}$

Proof

here: $x \to f(x) = x^2$

Jensen's Inequality

$$E(g(x)) \geq g(E(x))$$

~~Using above~~ E & Σ can be

$$E_{AV} = \frac{1}{M}\sum_m E_x\left[(y_m(x) - f(x))^2\right]$$

$\because E$ & Σ are inter-changeable

$$= \frac{1}{M}\sum_m E_x (y_m(x))^2 - \frac{2}{M}\sum_m E_x\, y_m(x) f(x)$$

$$+ \underbrace{\frac{1}{M}\sum_m E_x (f(x))^2}_{\text{No dependence on } m}$$

$$= \frac{1}{M}\sum_m E_x (y_m(x))^2 - \frac{2}{M}\sum_m E_x\, y_m(x) f(x)$$

$$+ E_x (f(x))^2$$

~~$E_{ENS} - E_{AV} \leq 0$~~

$$E_{ENS} = E_x\left[\left(\frac{1}{M} y_m(x)\right)^2\right] - 2E_x\left(\frac{1}{M} y_m(x) f(x)\right) + E_x (f(x))^2$$

$E_{ENS} - E_{AV} \leq 0$ Hence proven

$\underline{\frac{1}{M^2} - \frac{1}{M} \leq 0}$ Given $M > 0$

Let another kind of error be: RMS error

$$E_{RMS} = E\left[\frac{1}{M}\left(\sum_i (y_i(x) - f(x))^2\right)\right]^{1/2}$$

$$= \frac{1}{M} E\left(\sum y_i(x)\right)^2 + \frac{1}{M} E (f(x))^2$$

$$\leq E \frac{1}{M}\left[\sum y_i x\right]$$

On checking with E_{ENS}

$$\boxed{\frac{1}{M} - \frac{1}{M^2} > 0}$$

Hence $E_{RMS} < E_R$

Let another kind of error be: RMS error

$$E_{RMS} = E\left[\frac{1}{M}\left(\sum_i \left(y_i(x_i) - f(x)\right)^2\right)\right]^{1/2}$$

$$= \frac{1}{M} E\left(\sum y_i(x)\right)^2 + \frac{1}{M} E\left(f(x)\right)^2$$

$$\le E\frac{1}{M}\left[\sum y_i x - \right]$$

On checking with E_{ENS}

$$\boxed{\frac{1}{M} - \frac{1}{M^2} > 0}$$

Hence $E_{RMS} < E_R$

Given: Extension of the cross entropy error function for a multi-class classification problem:

$$E(\omega) = -\sum_{n=1}^{N}\sum_{k=1}^{N} t_{kn} \ln y_k(x_n, \omega) \quad (1)$$

where, k = no of classes ; N = no. of data samples

t_n = OHE for a data sample 'x_n'

y_k = n/w o/p = $P(t_k = 1 \mid x)$

$$0 \leq y_k(X, W) = \frac{\exp(a_k(X, W))}{\sum_j \exp(a_j(X_j, W))} \leq 1 \quad (2)$$

$a_k \rightarrow$ pre softmax activation error function

To prove: $\frac{\partial E}{\partial a_k} = y_k - t_k$

Proof: wkt,

$$a_k = \frac{1}{1+e^{-(X^T W + b)}}$$

$$\frac{\partial E}{\partial a_k} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial a_k}$$

Case(1) for $i=k$

$$\frac{\partial y_i}{\partial a_k} = \frac{e^{a_i}\sum_j e^{a_j} - e^{a_i a_k}}{\left(\sum_j e^{a_j}\right)^2}$$

$$= \frac{e^{a_i}\left(\sum_j e^{a_j} - e^{a_k}\right)}{\sum_j (e^{a_j})^2 \cdot \cancel{\quad}}$$

$$= \left(\frac{e^{a_i}}{\sum_j e^{a_j}}\right) \cdot \left(1 - \frac{e^{a_i}}{\sum e^{a_j}}\right) \quad k=i$$

$$= y_i (1-y_i)$$

Case(2) for $i \neq k$

$$\frac{\partial y_i}{\partial a_k} = 0 - \frac{e^{a_i} e^{a_k}}{\sum_j e^{a_j} \sum_j e^{a_j}}$$

$$= -y_i y_k$$

Hence:

$$\frac{\partial y_i}{\partial a_k} = \left\{ \begin{array}{ll} y_i (1-y_i) & \text{if } i=k \\ y_i y_k & \text{if } i \neq k \end{array} \right\} \quad \text{--- } (1)$$

$$\text{Now } \frac{\partial E_g}{\partial y_k} = \sum_k \sum_n \frac{t_{kn}}{y_k(x,w)} \quad \text{--- } (2)$$

Now multiplying Eq(1) & Eq(2)

$$\frac{\partial E}{\partial a_k} = -\sum_k \sum_n \frac{t_{kn}}{y_k(x,w)} \Big[\underbrace{-y_i y_k}_{i \neq k} - \underbrace{y_i(1-y_i)}_{i=k} \Big]$$

$$= -\sum_{\substack{k,n \\ \neq i}} \frac{t_{kn}}{\cancel{y_k}} (-y_i \cancel{y_k}) - \sum_{\substack{k,n \\ =i}} \frac{t_{kn}}{y_k} (y_i - (1-y_i))$$

$$= + y_i t_{kn} - \sum t_{kn} \frac{\cancel{y_i}}{\cancel{y_i}} + \sum \frac{t_{kn}}{\cancel{y_i}} y_i \cancel{y_i}$$

$$= y_i \Big(\underbrace{\sum_{k \neq i} t_{kn} + t_{in}}_{1} \Big) - t_i$$

Summed over all values

$$\boxed{\frac{\partial E}{\partial a_k} = y_i - t_i}$$

Hence proven

1. Ensemble Methods

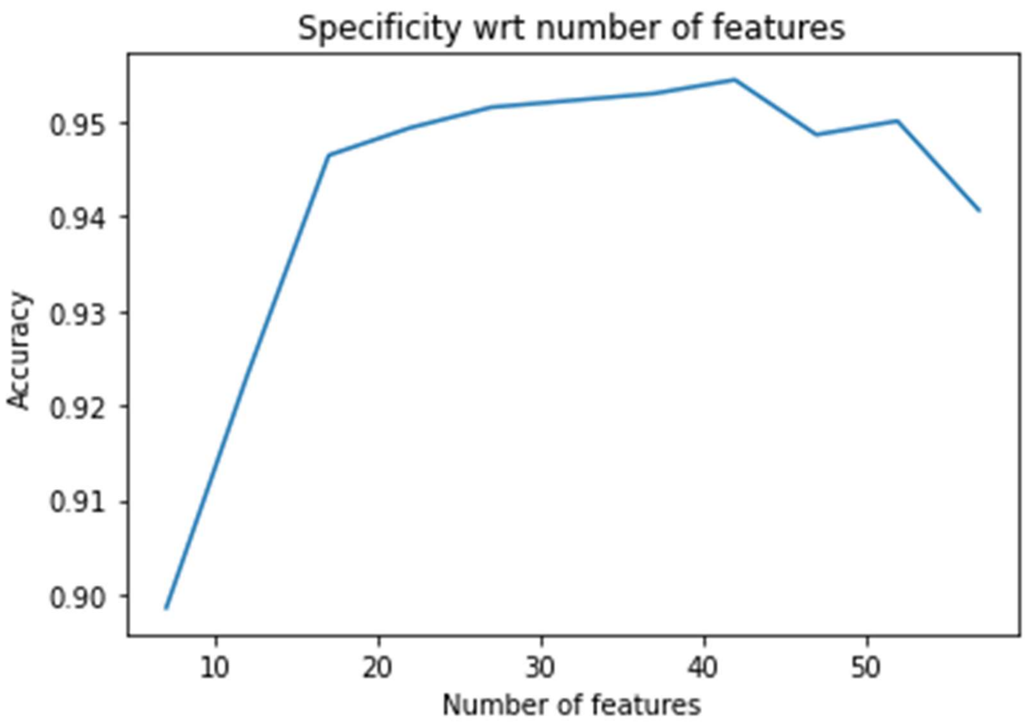
Question: Practical

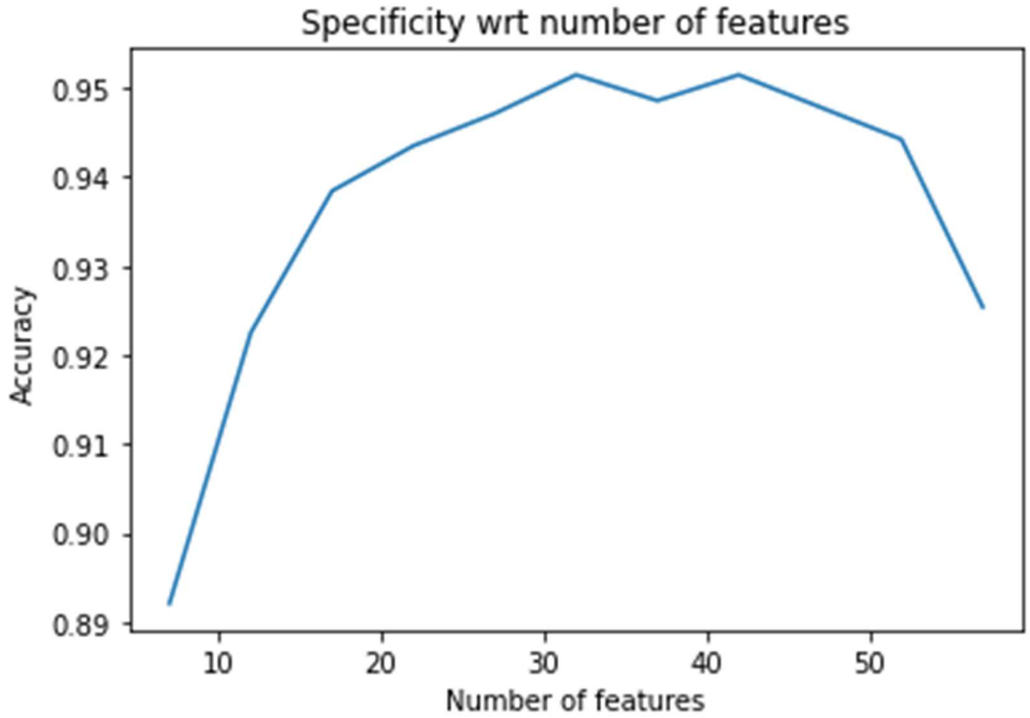
4a) The results have been computed below for a from the scratch Random Forest Classification

Code from scratch
The accuracy was 94.4967414916727% on the test data with number of features = 42
Time taken = Time: 02:51:17
Built-in scikit-learn library
The accuracy was 93.9174511223751% on the test data with number of features =
Time taken = 0.34009859599973424 seconds

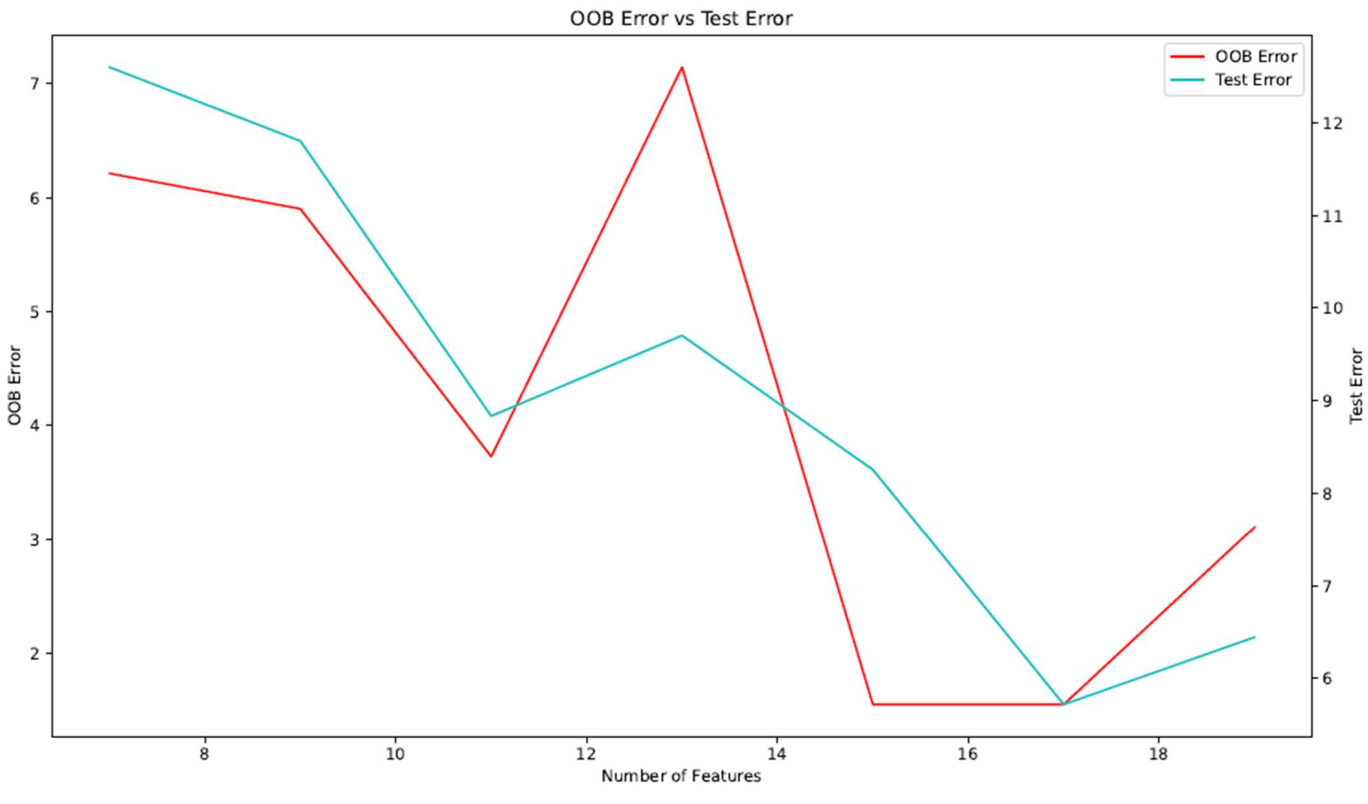
4b) b} Sensitivity of Random Forests to the parameter “no. of features”

```
The accuracy was89.86241853729182% on the test data with number of features = 7
The accuracy was92.32440260680667% on the test data with number of features = 12
The accuracy was94.6415640839971% on the test data with number of features = 17
The accuracy was94.93120926864592% on the test data with number of features = 22
The accuracy was95.1484431571325% on the test data with number of features = 27
The accuracy was95.2208544532947% on the test data with number of features = 32
The accuracy was95.2932657494569% on the test data with number of features = 37
The accuracy was95.43808834178131% on the test data with number of features = 42
The accuracy was94.85879797248371% on the test data with number of features = 47
The accuracy was95.00362056480812% on the test data with number of features = 52
The accuracy was94.0622737146995% on the test data with number of features = 57
```





4c) The relation between oob accuracy and test accuracy



5. The preprocessing and machine learning classification of the given loan dataset has been performed and attached as a google co5.llab IPython file:

https://colab.reseRarch.google.com/github/JainAnki/SiameseNetworks/blob/main/5.ipynb