Birla Institute of Technology and Science Pilani
Dept. of Computer Science & Information Systems
Deep Learning (CS F425)

Sat Mar 18, 2023
9:00–10:30 AM

Time: $1\frac{1}{2}$ Hour | *Mid-Sem (Close Book)* | Maximum Marks: 50

INSTRUCTIONS:

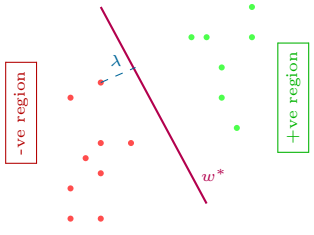
1. There are **7** questions in this paper, spread over **5** pages, make sure you have all the printed pages. Attempt **ALL** questions in any order you like.
2. You are free to make reasonable assumptions that you may be needed to logically answer the question (mention clearly them in the beginning of the answer). Answers without proper justification would not have any value.
3. Use of calculator is **allowed** in this exam.

1. What happens if we model a linearly separable data using a linear equation having random weights $(w_0, w_1, .., w_i, ..., w_n)$; and then use misclassified data points (say $x^{(k)}$) to update weights using [2+6]

$$w_i = w_i + \alpha y^{(k)} x_i^{(k)}$$

Prove or disprove that this strategy converges.

Solution: Since data is linearly separable, proposed modeling strategy would finally terminate and then the weights would be such that the line represented by $(w_0, w_1, .., w_i, ..., w_n)$ will work as separating hyperplane.
Convergence proof.



- Let w^* be separating hyperplane
- Make $||w^*|| = 1$, $\forall_i ||x_i|| \leq 1$
- Let nearest data point be λ distance apart from w^*
- Consider $yw^{*T}x$ it is > 0 for all $x \in D$ with minimum value λ
- Update increases $w^T w^*$ least by $\alpha\lambda$
  $(w + \alpha yx)^T w^* = w^T w^* + \alpha y w^{*T} x \geq w^T w^* + \alpha\lambda$
- $w.w^T$ increases less then α^2

$$w.w^T \rightsquigarrow (w + \alpha yx).(w + \alpha yx)^T$$

$$= w.w^T + 2\alpha y.w^T x + \alpha^2 y^2 x.x^T$$

Update happens only when x is mis classified so $y.w^T x < 0$. Also $x.x^T \leq 1$ due to scaling and $y^2 = 1$. So, $(w + \alpha yx) \times (w + \alpha yx)^T \leq w.w^T + \alpha^2$

- After k updates (when it converges)

$$k.\alpha\lambda \leq w^T.w^* = ||w^T.w^*|| \leq ||w^T||.||w^*|| = ||w|| = \sqrt{W^T W} < \sqrt{k\alpha^2}$$

$k \leq 1/\lambda^2$

- Number of steps of update is bounded by $1/\lambda^2$

2. Optimize the equation [8]

$$f(x, y, z) = x^2 + x + 3y^2 + 3z^2$$

provided the constraint being $x^2 + y^2 + z^2 = 2$

Solution:

$$L = x^2 + x + 3y^2 + 3z^2 - \lambda(x^2 + y^2 + z^2 - 2)$$

$$\frac{dL}{dx} = 2x + 1 - \lambda(2x) \quad (1)$$

$$\frac{dL}{dy} = 6y - \lambda(2y) \quad (2)$$

$$\frac{dL}{dz} = 6z - \lambda(2z) \quad (3)$$

$$\frac{dL}{d\lambda} = -(x^2 + y^2 + z^2 - 2) \quad (4)$$

Solve equation $\frac{dL}{dy}$ and $\frac{dL}{dz}$ to get values and apply same to $\frac{dL}{d\lambda}$ to obtain other coordinated of optimal points by equating to zero.

x	y	z	λ
$\pm\sqrt{2}$	0	0	
$\frac{1}{4}$	0	$\pm\frac{\sqrt{31}}{4}$	3
$\frac{1}{4}$	$\pm\frac{\sqrt{31}}{4}$	0	3

Points comes out to be

Coordinates	Value
$(\sqrt{2}, 0, 0)$	$2 + \sqrt{2} = 3.44$
$(-\sqrt{2}, 0, 0)$	$2 - \sqrt{2} = 0.56$
$(1/4, 0, \frac{\sqrt{31}}{4})$	$\frac{1}{16} + \frac{4}{16} + 3\frac{31}{16} = \frac{98}{16} = 6.125$
$(1/4, 0, -\frac{\sqrt{31}}{4})$	$\frac{1}{16} + \frac{4}{16} + 3\frac{31}{16} = \frac{98}{16} = 6.125$
$(1/4, \frac{\sqrt{31}}{4}, 0)$	$\frac{1}{16} + \frac{4}{16} + 3\frac{31}{16} = \frac{98}{16} = 6.125$
$(1/4, -\frac{\sqrt{31}}{4}, 0)$	$\frac{1}{16} + \frac{4}{16} + 3\frac{31}{16} = \frac{98}{16} = 6.125$

So optimum (minima) is at $(-\sqrt{2}, 0, 0)$

3. While training a single perceptron on data, explain *perceptron training rule.* for the following [1+2+7] data

x_1	x_2	y
10	9	green
4	7	green
2	5	red
7	1	red
2	10	green
8	5	green
1	2	red
4	5	red
5	3	red
8	9	green
4	2	red

If the current weights are $w_0 = -0.5, w_1 = -0.5, w_2 = 0.5$, determine the loss. Also determine values of weights in the immediate next step. (Assume $\eta = 0.01$ and show calculations.)

Solution:

- **Perceptron Training Rule**
  For each misclassified example: $w_i = w_i + \eta(t - o)x_i$

x_1	x_2	y	y	$sw_0 + w_1x_1 + w_2x_2$	$\hat{y}$	Correct?
10	9	green	+1	-1	-1	0
4	7	green	+1	+1	+1	1
2	5	red	-1	+1	+1	0
7	1	red	-1	-3.5	-1	1
2	10	green	+1	3.5	+1	1
8	5	green	+1	-2	-1	0
1	2	red	-1	0	+1	0
4	5	red	-1	0	+1	0
5	3	red	-1	-1.5	-1	1
8	9	green	+1	0	+1	1
4	2	red	-1	-1.5	-1	1

Accuracy: $7/11 = 63.63\%$
Loss = 100 -63.63 = 36.37%

Weight update: $w_0 = -0.5, w_1 = -0.5, w_2 = 0.5, \eta = 0.01$

x_1	x_2	t	$\hat{o}$	t-o	w_0	w_1	w_2
10	9	+1	-1	2	-0.48	-0.3	0.68
4	7	+1	+1	0	-0.48	-0.3	0.68
2	5	-1	+1	-2	-0.50	-0.32	0.63
7	1	-1	-1	0	-0.50	-0.32	0.63
2	10	+1	+1	0	-0.50	-0.32	0.63
8	5	+1	-1	2	-0.48	-0.16	0.73
1	2	-1	+1	-2	-0.50	-0.18	0.69
4	5	-1	+1	-2	-0.52	-0.62	0.59
5	3	-1	-1	0	-0.52	-0.62	0.59
8	9	+1	+1	0	-0.52	-0.62	0.59
4	2	-1	-1	0	-0.52	-0.62	0.59

Final weights in next step are $w_0 = -0.52, w_1 = -0.62, w_2 = 0.59$

4. What is Xavier initialization? Derive the formula for appropriate variance in initialized weights. [1+4]

Solution:

1. xavier initialization
   - Standard is to make weights centered around 0 with small variance[1] k/S_{l-1} where previous layer has S_{l-1} neurons[2]
   - Weights are sampled on normal distribution with varience k/S_{l-1}
     $k = 2$ for RelU and $k = 1$ for sigmoid/tanh
   - Another possibility is to use variance as
     $k/(S_{l-1} + S_l)$ or $k/(S_{l-1}S_l)$

[1] variance is average of the squared differences from the mean
[2] Standard daviation becomes $\sqrt{k/S_{l-1}}$

2. Assume data is i.i.d. from normal distribution with $\mu = 0, \sigma = 1$

   - As $var[w_i x_i] = E[x_i]^2 var[w_i] + E[w_i]^2 var[x_i] + var[x_i]var[w_i]$ since $E[x_i] = \mu = 0$ therefore $var[w_i x_i] = var[x_i]var[w_i]$
   - Consider a neuron $y = w_1 x_1 + w_2 x_2 + ... + w_n x_n + b$

$$\begin{aligned} var[y] &= var[w_1 x_1 + w_2 x_2 + ... + w_n x_n + b] \\ &= var[w_1]var[x_1] + var[w_2]var[x_2] + ... + var[w_n]var[x_n] + var[b] \\ &= var[w_i]var[x_i] + var[w_i]var[x_i] + ... + var[w_i]var[x_i] + 0 \\ &= S_{l-1}.var[w_i]var[x_i] \qquad n = S_{l-1} \end{aligned}$$

3. To have same varince in input and output $S_{l-1}.var[w_i] = 1$ or

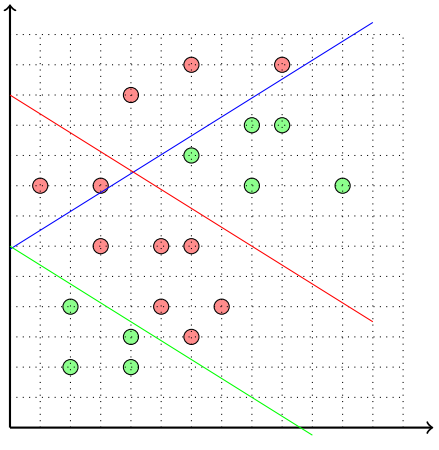
$$var[w_i] = 1/S_{l-1}$$

5. Consider following training data points. [6]

Class Green	(2,2), (2,4), (4,2), (4,3), (6,9), (8,8), (11,8), (8,10), (9,10)
Class Red	(1,8), (3,6), (3,8), (4,11), (5,4), (5,6), (6,3), (6,6), (6,12), (7,4), (9,12)

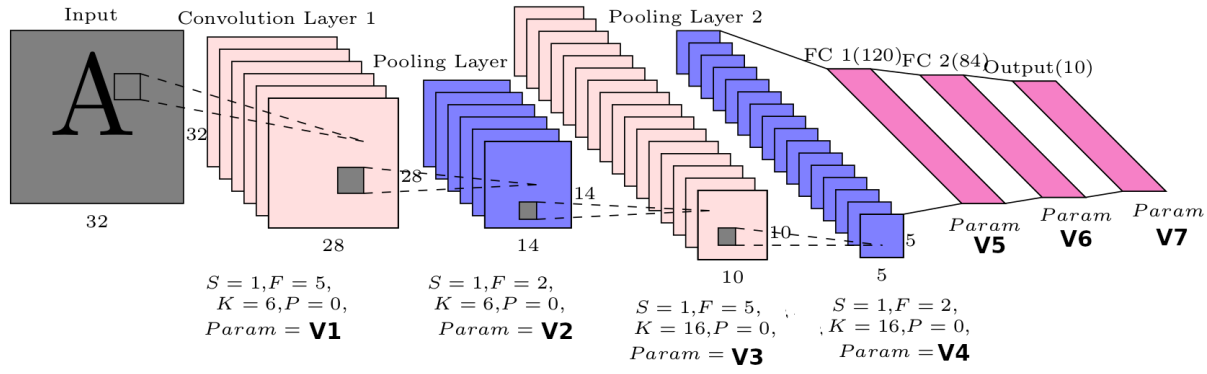
Suggest a suitable **neural network** architecture to obtain 100% training accuracy. Justify your answer.

Solution:



Three lines are needed to divide the region so
Input: 2
First: 3
Second: 5
Final: 1

6. Identify the architecture in the image given below and determine the values of the number of parameters shown as $v1, v2, v3, v4, v5, v6, v7$ [1+5]



Solution: Assuming no bias

$v1 = 6(5 \times 5) = 150$
$v2 = nil$
$v3 = 16(5 \times 5 \times 6) = 2400$
$v4 = nil$
$v5 = 5 \times 5 \times 16(120) = 48000$
$v6 = 120 \times 84 = 10080$
$v7 = 84 \times 10 = 840$

7. Comment on the following (in less than 300 words)
   (a) What happens in training when we increase relevent training data for a fixed model. [1]
   (b) What is early stopping? How it is done and what is achieved by this. [2]
   (c) What limitation of a single perceptron was highlighted by the Minskey? [2]
   (d) What is ChatGPT? Explain techinique used and importence of the components. [2]

What happens in training when we increase relevent training data for a fixed model.

Solution:

(a) Model gets regularized.
(b) Stopping training before model overfits.
(c) Dealing with XOR is not possible with single perceptron (layers are needed)
(d) Large language model with human in loop for reinforcement.
