CS 4100: Foundations of Artificial Intelligence (Spring 2020) Roi Yehoshua
Student name: (Due) April 9, 2020

Machine Learning: Regression and Neural Nets

1 Regularized Linear Regression

Recall the problem of linear regression and the derivation of its various learning algorithms. We are given a dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$, where $x^{(i)} \in \mathbb{R}^p$ (i.e., each data point has p features) and $y^{(i)} \in \mathbb{R}$. The hypothesis class we consider in linear regression is, for $w \in \mathbb{R}^{p+1}$:

$$h_w(x) = w_0 + w_1 x_1 + \ldots + w_p x_p = w_0 + \sum_{j=1}^{p} w_j x_j$$

Consider the following alternative error function to *minimize* (while using squared-error loss):

$$J(w) = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - h_w(x^{(i)})\right)^2 + \lambda \sum_{j=1}^{p} w_j^2$$

1. Compared to standard linear regression, there is an extra *regularization* term: $\lambda \sum_{j=1}^{p} w_j^2$. From an optimization perspective, what is the role of this term, i.e, which weights (parameters) w does it prefer, and which does it penalize, assuming $\lambda > 0$? Regularization helps to avoid the overfitting
L2 likes the small value, since it penalize the large one.

2. λ is known as a *regularization constant*; it is a *hyperparameter* that is chosen by the algorithm designer and fixed during learning. What do you expect to happen to the optimal w when $\lambda = 0$? $\lambda \to +\infty$? $\lambda \to -\infty$?
$\lambda = 0$, no regularization applied | $\lambda \to -\infty$ can get optimal w | $\lambda \to \infty$, penalty grow all $w \to 0$.

3. Find the partial derivatives $\frac{\partial J}{\partial w_0}$ and $\frac{\partial J}{\partial w_j}, j \in \{1, \ldots, p\}$. (These two cases are different; for the second one, only derive once for an arbitrary index j.)

4. Write out the update rule for gradient descent applied to the error function $J(w)$ above. Compare with the gradient descent update rule for standard linear regression; what is the difference? How does this difference affect gradient descent, assuming $\lambda > 0$?
Consider what happens both when w_j is positive and when it is negative.

5. In order for the problem to be well-defined, an appropriate regularization constant λ must be chosen for the given problem (dataset). How should the designer choose λ?

3. $$\frac{\partial J}{\partial w_0} = \frac{1}{n} \cdot \sum_{i=1}^{n} \cdot \frac{\partial}{\partial w_0} (y^{i} - h_w(x^{i})^2) + \lambda \frac{\partial}{\partial w_0} (w_1^2 + w_2^2 + w_3^2)$$

$$= -\frac{1}{n} \sum_{i=1}^{n} 2 \cdot (y^{(i)} - h_w(x^{(i)}))$$

$$\frac{\partial J}{\partial w_i} = -\frac{1}{n} \sum_{i=1}^{n} \cdot 2(y^{(i)} - h_w(x^{(i)})) \cdot x_i + \lambda 2 w_i$$

4. $$w_0 = w_0 + \alpha \left[\frac{1}{n} \cdot \sum_{i=1}^{n} (y^{(i)} - h_w(x^{(i)}))\right]$$

$$w_j := w_j + \alpha \left[\frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - h_w(x^{(i)})) x_j^{(i)} - \lambda \cdot 2w\right]$$

5. The regularization constant can't be too scalable since it doesn't avoid the overfit, it has too much bias. The designer can use the brute force method, to attempt some parameter and plot the result to find a great range for it

⑦ $w_0 = -1 + 0.5 = -1$

$\Delta w_1 = 0.5 \cdot 0 = 0$

$w_1 = 0$

$\Delta w_2 = 0$ $w_2 = 0.5$

$\Delta w_3 = 0$ $w_3 = 0.5$

⑧ ↑ same

2 Perceptron

You are given the following training set:

Sample	x_1	x_2	x_3	y
1	1	4	1	1
2	1	2	3	1
3	0	0	1	1
4	-1	4	0	1
5	1	0	-2	0
6	-1	-1	1	0
7	0	-4	0	0
8	1	0	-3	0

$I = 2$

$f(x) = -1 x_1 + 0 \times 2 + 0.5 x_3 + 0.5 x_4$

1) $f(x) = 1 = y = 1$

2)

Run two iterations of the perceptron learning algorithm on this data set. Start with initial weights of 0, and use a learning rate of $\alpha = 0.5$. Don't forget to add a bias neuron $x_0 = 1$.

5) $\Delta w_0 = 0.5$ $\Delta w_1 = 0$

$w_0 = -0.5$ $\Delta w_2 = 0.5$

$w_3 = 0.5 + 0.5 = 1$

$i = 1$

$f(x) = x_{01} + w_1 x_1 + w_2 x_2 + w_3 x_3$

$1 = 1 + 0 + 0 + 0$

$w_i = w_i + 0.5(1 - 1)$

$1 = w_i + \dots$

6) $w_0 = -0.5$ $\Delta w_0 = -0.5$

$w_0 = -0.5 + (-0.5) = -1$

$w_1 = 0 + 0.5 = 0.5$

$w_2 = 1$

$\Delta w_3 = 0.5$ $w_3 = 0.5$

7)

8) $f(x)$: practice = label, so no change

$o = (1 + 0 + 0 + 0) = -1$

⑤ $\Delta w_1 = 0.5 \cdot (-1) \cdot 1 = -.5$

$w_1 = 0 - .5 = -0.5$

$\Delta w_3 = 0.5 \cdot (-1) \cdot -2 = 1$

$w_3 = 0 + 1 = 1$

$f(x) = -1 x_1 + 0.5 x_1 + x_2 + 0.5 x_3$

⑥ $1 + (-1) + 0.5) + 0 + 1.1 = 2.5 > 1 = 1$

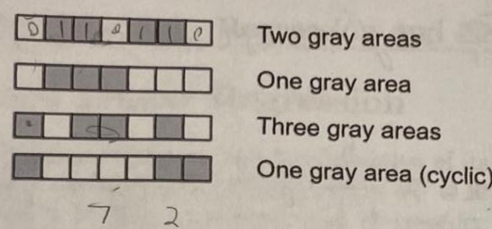
$\Delta w_1 = -0.5 + 0.5 = 0$

$\Delta w_2 = 0.5 = 0.5$

$\Delta w_3 = 1 + 0.5 \times (0 - 1) \times 1 = 0.5$

3 Neural Networks

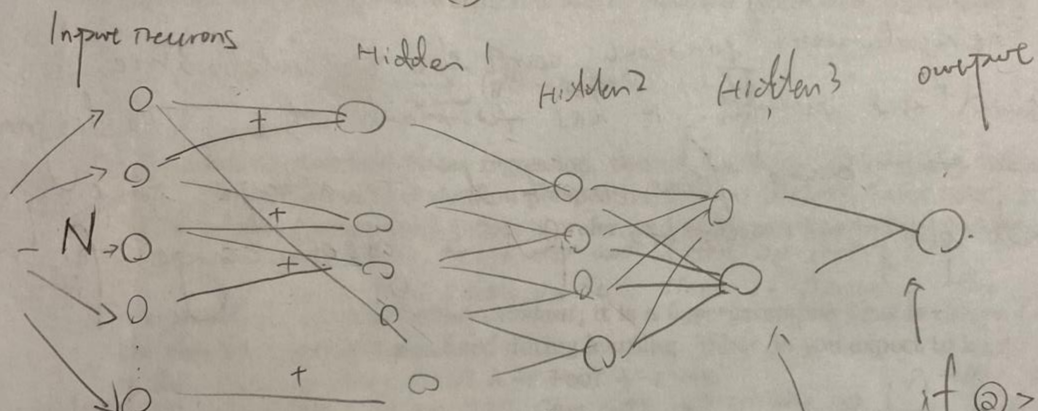
You are given the following patterns:



Each pattern consists of N cells, that can be white or gray. Treat the patterns as cyclic (i.e., the first and the last cell are considered to be adjacent).

Design a neural network that identifies if there are more than G gray areas in the pattern. Show the architecture of your network and its weights for $N = 7$ and $G = 2$.

MORE than N → 1



Sum up each adjacency input node.
there are 2 possibilities
0+1 = 1   1+1 = 2
0+0 = 0

Activate function if the adjacency are 2. combine them

if ② > G, return 1, otherwise 0

☆ count how many 2 in the last layer.