HW5 ECE498 Spring 2020

Netid :- GOWTHAM4

Problem1 :- Jupyter Notebook has been submitted online.

It contains codes for

SVM, DecisionTree, Randomforest classifier and 5-fold Cross validation results (sklearn)

1) (Mean Accuracy, S.D Accuracy)

SVM	DT	RF
~~C=0.1~~	d=3	N=5
C=1	d=4	N=11
~~C=10~~	d=6	N=13

(mean Precision, S.D Precision)

SVM	DT	RF
C=0.1	d=3	N=5
C=1	d=4	N=11
C=10	d=6	N=13

(mean Recall, s.D Recall)

SVM	DT	RF
C=0.1	d=3	N=5
C=1	d=4	N=11
C=10	d=6	N=13

2) Label imbalances cause accuracy to be high. Even if most are predicted to of class 0 (not pulsar), accuracy will be > 90%. Recall, Precision take TP into account

See Jupyter Notebook

3) Best classifier : SVM C=10, we used F1_score weighted → takes care of label imbalance

---

Problem 2 :- Activation function

$$g(x) = \frac{1}{1+e^{-x}} \quad \text{(sigmoid function)}$$

$$g'(x) = g(x)(1-g(x))$$

1) (vector dot products)

$$z_1 = w_1 \cdot x_1 + b_1$$

$$z_4 = w_4 a_1 + w_5 \cdot a_2 + w_6 \cdot a_3 + b_4$$

$$a_1 = \frac{1}{1+e^{-z_1}}$$

$$a_4 = \frac{1}{1+e^{-z_4}}$$

$$\text{Output} = \hat{y} = a_4$$

$$a_4 = \frac{1}{1+e^{-z_4}} = \frac{1}{1+e^{-\sum_{i,j} w_i \cdot a_j + b_4}}$$

$$\Rightarrow \hat{y} = a_4 = \frac{1}{1+e^{-\left[w_4 \cdot \left(\frac{1}{1+e^{-w_1 x_1 + b_1}}\right) + w_5 \cdot \left(\frac{1}{1+e^{-w_2 x_1 + b_2}}\right) + w_6 \cdot \left(\frac{1}{1+e^{-w_3 x_1 + b_3}}\right) + b_4\right]}}$$

Note :- $g(x) = \frac{1}{1+e^{-x}}$:- Sigmoid function

2) Loss, $L_i = (y_i - \hat{y}_i)^2$ (True label, Predicted label)

$$\frac{\partial L_i}{\partial b_4} = 2(y_i - \hat{y}_i) \frac{\partial \hat{y}_i}{\partial b_4}$$

$$\hat{y}_i = a_4 = g(z_4)$$

$$\Rightarrow \frac{\partial \hat{y}_i}{\partial b_4} = g(z_4)(1-g(z_4)) \frac{\partial z_4}{\partial b_4}^{1}$$

$$\Rightarrow \boxed{\frac{\partial L_i}{\partial b_4} = 2(y_i - \hat{y}_i)\, g(z_4)(1-g(z_4))}$$

Similarly

$$\frac{\partial L_i}{\partial w_4} = \frac{\partial \hat{y}_i}{\partial w_4} 2(y_i - \hat{y}_i)$$

also, $$\frac{\partial \hat{y}_i}{\partial w_4} = g(z_4)(1-g(z_4)) \frac{\partial z_4}{\partial w_4}^{a_1}$$

$$\Rightarrow \boxed{\frac{\partial L_i}{\partial w_4} = 2(y_i - \hat{y}_i)\, g(z_4)(1-g(z_4))\, a_1}$$

$$\frac{\partial L_i}{\partial b_1} = \frac{\partial L_i}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial z_4} \frac{\partial z_4}{\partial a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial b_1}, \quad \frac{\partial z_1}{\partial b_1} = 1$$

$$\Rightarrow \boxed{\frac{\partial L_i}{\partial b_1} = 2(y_i - \hat{y}_i)\left[g(z_4)(1-g(z_4))\right] w_4 \left[g(z_1)(1-g(z_1))\right] * 1}$$

$$\frac{\partial L_i}{\partial w_1} = \frac{\partial L_i}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial z_4} \frac{\partial z_4}{\partial a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial w_1}$$

$$\Rightarrow \boxed{\frac{\partial L_i}{\partial w_1} = 2(y_i - \hat{y}_i)\left[g(z_4)(1-g(z_4))\right] w_4 \left[g(z_1)(1-g(z_1))\right] x_1}$$

3) Gradient descent step :-

$$w_1' = w_1 - \eta \frac{\partial L_i}{\partial w_1} \quad \text{(Single } x_i\text{)}$$

$$\Rightarrow \boxed{w_1' = w_1 - \eta \, x_i \, \delta}$$

4) Gradient descent multiple samples (n)

$$L = \frac{1}{n} \sum_{i=1}^{n} L_i = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$\frac{\partial L}{\partial w_1} = \sum_{i=1}^{n} \frac{\partial L_i}{\partial w_1}$$

$$\Rightarrow \boxed{w_1' = w_1 - \frac{\eta}{n} \sum_{i=1}^{n} \frac{\partial L_i}{\partial w_1}} \rightarrow \text{given by Q2.2.}$$

5) Sigmoid vs ReLU Activation functions

Ref:- stackexchange, 126238
wikipedia entries

Advantages of ReLU over Sigmoid

1) Gradient is easier to calculate and is 1 when $a > 1$
if $ReLU(a) = max(0, a)$

2) Sparsity, when $a \leq 0$
the resulting computation is sparse because ~~gradient~~ function outputs zero.

⭐ Sparsity functions better in many cases than dense representations (common in sigmoid)

Advantages of Sigmoid over ReLU

Sigmoid

1) Activation produces output in the range (0, 1) → easy to store. Doesn't blow up output of neurons.

2) It is better than ReLU in some cases where large number of neurons die (ex:- ~~to~~ multiple -ve values in input)