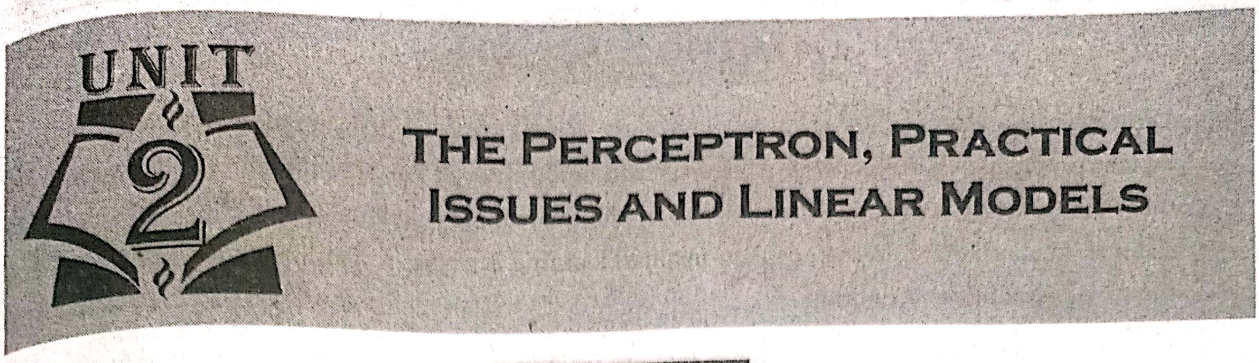


THE PERCEPTRON, PRACTICAL ISSUES AND LINEAR MODELS

SYLLABUS

The Perceptron: Bio-inspired Learning, The Perceptron Algorithm, Geometric Interpretation, Interpreting Perceptron Weights, Perceptron Convergence and Linear Separability, Improved Generalization, Limitations of the Perceptron.

Practical Issues: Importance of Good Features, Irrelevant and Redundant Features, Feature Pruning and Normalization, Combinatorial Feature Explosion, Evaluating Model Performance, Cross Validation, Hypothesis Testing and Statistical Significance, Debugging Learning Algorithms, Bias Variance Tradeoff.

Linear Models: The Optimization Framework for Linear Models, Convex Surrogate Loss Functions, Weight Regularization, Optimization and Gradient Descent, Support Vector Machine.

LEARNING OBJECTIVES

- Introduction to Bio-inspired Learning
- Perceptron and its Learning Algorithm
- Perceptron Convergence Theorem
- Limitations of Perceptron
- Importance of Good Features
- Irrelevant and Redundant Features
- Evaluating the Performance of a Model
- Cross Validation in Machine Learning
- Hypothesis Testing and Statistical Significance
- Debugging Learning Algorithm
- Optimization Framework for Linear Models
- Convex Surrogate Loss Functions in Machine Learning
- Brief Introduction to Support Vector Machines.

INTRODUCTION

Perceptron is a learning algorithm used in neural network model. It is simple and efficient in certain problems. This algorithm is error-driven and online. It has few limitations that it can only learn linearly separable functions and model only linear relationship between inputs and outputs. According to folk biology, Bio-inspired learning is a type of learning which works like the working of neurons in brain.

Features can be defined as the attributes that are used to represent the data and are used by the classification. Good features can improve the accuracy and generalization ability of a machine learning model, while poor features can lead to poor performance. Feature pruning can be defined as a process of eliminating irrelevant features and normalization can be defined as a process of making the data consistent.

For linear models, the optimized framework can be obtained using the perceptron. The main objective of perceptron is to determine a separating hyperplane for a given training dataset. Support Vector Machine (SVM) is a supervised machine learning algorithm that is used for classification and regression tasks. The main objective of SVM is to find a hyperplane in high dimensional space that separates the classes with maximum margin.

PART - A

SHORT QUESTIONS AND ANSWERS

Q1. Discuss about importance of good features.

Answer : (Model Paper-III, Q4 | July/Aug.-22, Q5 [MGU])

Features can be defined as the attributes that are used to represent the data and are used by the classification model to make predictions. Good features can significantly improve the accuracy and generalization ability of a machine learning model, while poor features can lead to poor performance. The importance of good features can be given as follows,

1. To Capture the Relevant Information in the Data
2. To Reduce Overfitting
3. To Improve Interpretability
4. To Reduce Computational Complexity.

Q2. Write about Redundant features.

Answer : Model Paper-II, Q4

Redundant Features

Redundant features can be defined as the features that are highly correlated for the prediction task. It consists of information more than required.

Example

In an image, having a dark red pixel at position (10, 20) is redundant while already having a dark red pixel at position (11, 20). Here, both the pixels are useful but one is unwanted depending on the situation.

Note that, it is ok to have one bad feature over 100 features. But, it creates difficulty only when the number of bad features exceeds good and correlated features.

In a shallow decision tree, the only correlated features are being selected. This model can kick out the uncorrelated features by limiting its depth.

Q3. Explain about pruning and normalization.

Answer : (Model Paper-I, Q4 | June/Aug.-22, Q4 [OU])

Feature Pruning

Feature pruning can be defined as a process of eliminating the number of unwanted/irrelevant input features in a model. It helps in removing the irrelevant features and concentrating on the required features.

Normalization

Normalization can be defined as a process of making the data consistent. It enhances the convergence rate of training algorithm and stops the domination of one feature over another. The main objective of normalization is to make the process of learning an algorithm easier. There are two types of normalization. They are as follows,

(a) Feature Normalization

(b) Example Normalization.

Q4. What is the purpose of cross validation?

Answer : Model Paper-II, Q5

Cross validation is used to evaluate the performance of a classification model. This technique splits the data into various sub-sets referred as folds. Then, one fold out or all folds can be considered as test and the remaining are considered as training data. It helps in training and testing all the data sets and estimates the performance of a model.

The k-fold crossvalidation can be considered as the general cross-validation technique. Here, the dataset is equally divided into k sub-parts. Thus, the model is trained k times, with each fold used as the validation set once, and the results are averaged over all the iterations to get the final estimate of the performance of a model.

Q5. Define statistical significance.

Model Paper-III, Q5

Answer :

Statistical significance refers to the likelihood that the results users obtained are not due to chance. In machine learning, the p-values are used to determine statistical significance. A p-value is the probability of observing a test statistic as extreme or more extreme than the one we observed, assuming the null hypothesis is true. If the p-value is small enough (typically less than 0.05), the null hypothesis is rejected and the user can conclude that the results are statistically significant.

Q6. Explain about bias-variance trade-off.

Answer :

(Model Paper-I, Q5 | June/Aug.-22, Q5 [OU])

The bias/variance trade-off can be defined as the trade-off between the estimation error and approximation error. Here, estimation error is variance and approximation error is bias.

Generally, the test errors can be decomposed into two terms called estimation error and approximation error. Consider that f is a learned chosen from a set of learned classifiers $\mathcal{F}$. Then,

$$\text{error}(f) = \underbrace{\left[\text{Error}(f) - \min_{f^* \in \mathcal{F}} \text{error}(f^*)\right]}_{\text{Estimation Error}} + \underbrace{\left[\min_{f^* \in \mathcal{F}} \text{Error}(f)\right]}_{\text{Approximation Error}}$$

Here,

- The first term i.e., estimation error defines the distance between the actual learned classifier f, and the optimal classifier f^*.
- The second term i.e., approximation error defines the quality (like depth of decision tree) of model family.

Note that, the estimation error and approximation error are calculated only in the constructed cases.

Q7. Write about convex surrogate loss function.

Answer :

(Model Paper-II, Q6 | June/Aug.-22, Q6 [OU])

In machine learning, the convex surrogate loss functions are used when the true loss function is not convex but a convex function is used as an alternative for optimization purpose. The main reason behind using convex function is that they contain unique global minimum that helps the optimization process simple and efficient.

The value of optimization depends on the values of w and b. A small changes in the values of these parameters create a large difference in the optimization function.

Q8. Write about weight regularization.

Answer :

In machine learning, weight regularization is used to prevent overfitting by adding the regularization term to the loss function that penalizes higher weights. The loss function can be given as,

$$\min_{w,b} \sum_{n} l(y_n, w.x_n + b) + \underbrace{\lambda R(w, b)}_{\text{Convex function}}$$

Here, the most important requirement is that the weight vector should be zero or closer to zero. This form is considered as "Inductive bias".

Q9. What is optimization and gradient descent in linear models?

Answer :

(Model Paper-I, Q6 | July/Aug.-22, Q6 [MGU])

The gradient methods of optimization is used when the user wants to find the maximum and minimum of a function $f(x)$. In this case, the gradient descent method that determines the minimum of a function is used. Here at each step, the gradient of a function that is trying to optimize the value is measured. This can be represented as,

$x \leftarrow x + \eta g$ [$\because$ Here, η is the size of each step]

Algorithm: GradientDescent($\mathcal{F}$, K, $\eta_1, \ldots$)

Input: $\mathcal{F}$ is a function needs to be minimized

K is the number of iterations

$\eta_1, \eta_2, \ldots \eta_k$ are the sequence of learning rates.

Step1: Initialize the variables that needs to be modified

$z^{(0)} \leftarrow <0, 0, \ldots 0>$

Step2: for $k = 1$ to K do:

$g^{(k)} \leftarrow \Delta_z \mathcal{F}|_{z^{(k-1)}}$ //Computing gradient at present location

$z^{(k)} \leftarrow z^{(k-1)} - \eta^{(k)} g^{(k)}$ //Next lower iteration to gradient

end for

Setp3: return $z^{(k)}$

Q10. Write about support vector machine in brief.

Answer : (Model Paper-III, Q6 | July/Aug.-22, Q2 [MGU])

Support Vector Machine (SVM) is a supervised machine learning algorithm that is used for classification and regression tasks. The main objective of SVM is to find a hyperplane in a high-dimensional space that separates the classes with maximum margin. Here, margin refers to the distance between the hyperplane and the closest data points in every class. This can be considered as a constrained optimization problem and written as,

$$\min_{w,b} \frac{1}{\gamma(w, b)}$$

The above equation subjects to $y_n(w.x_n + b) \geq 1$ (for all n)

Here, γ represents the margin

w, b represents the parameters weight, bias respectively.

PART - B

ESSAY QUESTIONS AND ANSWERS

2.1 THE PERCEPTRON

2.1.1 Bio-inspired Learning, The Perceptron Algorithm

Q11. Explain about bio-inspired learning.

Answer :

The type of learning which works like the working of neurons in brain according to folk biology is referred as bio-inspired learning. A human brain consists of neuron units that send and receive electrical signals among all these units. A neuron is said to be active based on its rate of sending signals. Typical connectivity of neurons is shown below,

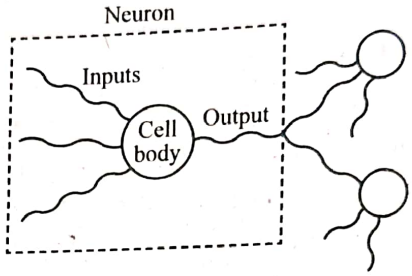


Figure: Connectivity Neuron

In the above figure, there are three neurons which are sending signals at different rates or activations. The strongness of firing depends on the strongness of connections and rate of incoming signals. Learning within the brain entirely depends on the inter neuron connectivity and connection strength.

Getting inspired from such working, bio-inspired learning algorithm is developed. Considering a single neuron, it receives D inputs from various neurons. The strength associated with these inputs are considered as feature values. The decision of firing or not firing depends on the sum of individual weights associated with incoming connections. If the neuron fires, it is considered as positive example and negative in case if it does not fire. The decision of firing depends on this negativity or positivity, i.e., firing is performed if the sum of weights is positive.

The sum can be computed by neuron as,

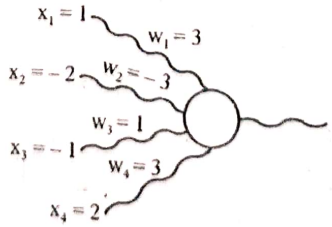
$$a = \sum_{d=1}^{D} w_d x_d$$

Here, w_d is the weight of input

x_d is the input feature vector.

Example

Consider the following figure,



The sum of weights and feature vectors can be calculated as,

$$a = 3 \times 1 + (-3) \times 2 + 1 \times (-1) + 3 \times 2$$
$$= 3 - 6 - 1 + 6$$
$$= 2.$$

Since, the result is positive, the neuron fires the signal.

Usually, the weight indicate whether the activation is increased or decreased. A positive weight indicates an increase whereas a negative weight indicates a decrease. Obtaining a positive threshold is always convenient which can be achieved by using a bias term 'b' in the neuron. So, the resultant summation with this term would be,

$$a = \left[\sum_{d=1}^{D} w_d x_d\right] + b$$

This is the resultant neural model of learning.

Q12. What is perceptron? Explain perceptron learning algorithm with example.

Answer : (Model Paper-I, Q14(a) | June/Aug.-22, Q14(a) [OU])

Perceptron

Perceptron is a learning algorithm used in neural network model. It is very simple and efficient in case of certain problems. It is different from k-nearest neighbours and decision tree algorithms in the following ways,

1. Online

This algorithm is online which means that only a single example is considered at a time rather than the entire data set. The next example is taken into consideration only after processing the existing one.

2. Error-driven

This algorithm is error-driven which means that it does not update or modify itself until an error or problem occurs.

Perceptron carries a guess or predicted value. While processing the example, the guess value is compared with actual one. If the predicted value is correct, no action is taken. However, in case of incorrect prediction, the value is set based on the next time around. After this, the next example is considered. When the last example is reached, it is looped back and iterated a specific times.

Perceptron Learning Algorithm

The training algorithm and its prediction algorithm is as follows,

Algorithm: PerceptronTrain(D, MaxIter)

$w_d \leftarrow 0$, for all $d = 1 \ldots D$ //initialization of weights

$b \leftarrow 0$ //initialization of bias

for iter = 1 . . . MaxIter do

  for all $(x,y) \in D$ do

    $a \leftarrow \sum_{d=1}^{D} w_d x_c + b$ // computation of activation for the given example

    if $ya \leq 0$ then

    $w_d \leftarrow w_d + yx_d$, for all $d = 1 \ldots D$ // updation of weights

    $b \leftarrow b + y$ // updation of bias

    end if

  end for

end for

return $w_0, w_1, \ldots, w_D, b$

Algorithm: PerceptronTest($w_0, w_1, \ldots, w_D, b, \hat{x}$)

$a \leftarrow \sum_{d=1}^{D} w_d \hat{x}_d + b$ // computation of activation for the test example

return SIGN(a)

In the above algorithm, the product ya is considered for checking whether the prediction is correct or not (at line 6). If it is found to be less than zero i.e., negative, update is performed. This update involves increasing of weight w_d to yx_d while bias is increased by y. With such an approach, the task gets improved in every iteration.

Example

Consider a positive example $y = +1$ which generates an error during computation of activation function a ($a < 0$). At this point, the weights are updated as $w'_1, w'_2, ..., w'_D$ and the bias is updated as b'. Now, the new activation function a' can be computed as,

$$a' = \sum_{d=1}^{D} w'_d x_d + b'$$

$$= \sum_{d=1}^{D} (w_d + x_d) x_d + (b + 1) \qquad \text{[As per the stated rule]}$$

$$= \sum_{d=1}^{D} w_d x_d + b + \sum_{d=1}^{D} x_d x_d + 1$$

$$a' = a + \sum_{d=1}^{D} x_d^2 + 1 > a$$

Since the obtained activation function a' greater than a and always positive [since x_d is squared and 1 is added to it]. For this, reason, the activation function is considered to be moved in correct direction.

The parameter MaxIter in the algorithm is considered as hyper parameter which conveys the number of iterations/passes to be done on the data set. This number should not be too larger or too smaller (1). A larger value makes the algorithm confusing while a too smaller value can lead it to underfit.

2.1.2 Geometric Interpretation, Interpreting Perceptron Weights

Q13. Explain the geometric interpretation of perceptron. Also discuss about interpreting perceptron weights.

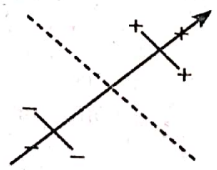
Answer :

Geometric Interpretation

The decision boundary of perceptron without bias term can be given as,

$$B = \left\{ x : \sum_d w_d x_d = 0 \right\}$$

The geometric interpretations can be done on this condition by considering the dot product i.e., w_d. If this product is zero for two vectors, they are considered as perpendicular to each other. For instance, by considering the weight vector, the data points and perpendicular plane can be drawn as follows,



It can be seen from the above figure that, a boundary is formed between positive and negative points. The vector is moving in the direction towards positive points or examples.

Making the geometric interpretations can be more helpful when projections are obtained from dot products. This means that the dot product $w.x$ can be considered as the distance from origin to x projected on vector w. This distance is considered as activation of the example.

When bias is considered, the threshold is directed towards negative or positive side based on the projection on w. This direction is directed by the bias term. When the production is computed, b is added and the result is then compared with zero. Therefore, the task of bias under geometric interpretation is to direct the direction boundary away from origin towards w. In case if b is negative, the boundary moves away from the w by $-b$ units. With this, positive classification can be obtained with more dimensional space.

Interpreting Perceptron Weights

The heuristic that help in interpretation of perceptron weights is to sort the weights from largest positive value to largest negative value and consider the top 10 and bottom ten among them. The features that positive predictions are made by perceptron are the top 10 while the features that negative predictions are made by perceptron are bottom 10. Such as heuristic is highly recommended when the input carries all binary values. However, it is not much useful in case of distinct individual features. This is because, when a set contains similar values say, $w_3 = 0$ or 1 and $w_9 = 0$ or 100, it can be easily predicted that w_9 is more important as it contains a larger set.

2.1.3 Perceptron Convergence and Linear Separability

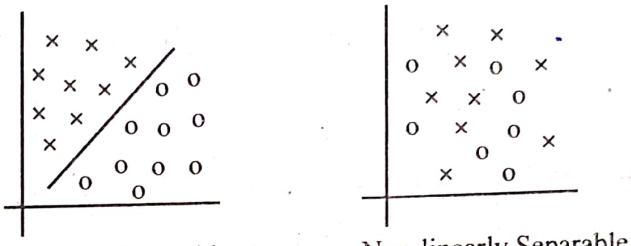
Q14. Explain about Perceptron Convergence and Linear Separability.

Answer :

Linear Separability

A data set is said to be linearly separable, if there exist a hyperplane (or) a line which separates the positive examples and negative examples. This simply means that the positive examples are kept at one side and negative examples on the other side of hyperplane.

If the data set cannot be separated, then they are said to be non-linearly separable data.



Linearly Separable Non-linearly Separable

Perceptron Convergence

Perceptron convergence refers to the procedure of making the pass through training data without involving any update or modification. In simple terms, it can be said that the training data has been classified successfully. The result of this is typically a linearly separable data set.

The perceptron learning algorithm works well with linearly separable data. This means that the algorithm stops at a weight vector w when the data is linearly separable. If dataset is not separable, then convergence does not happen.

The number of iterations depends upon the notation called margin. It is defined as the distance between the hyperplane and the nearest point to it. If the margin value is large, then it is known as easy learning problem and the perceptron can converge quickly on such models. If the margin value is less, then they are known as hard learning problems, where the perceptron takes more time to converge.

This margin defined on the separable data is computed as,

$$\text{Margin}(D, w, b) = \begin{cases} \min_{(x,y)\in D} y(wx + b); & \text{if } D \text{ is separable data} \\ -\infty \quad ; & \text{Otherwise} \end{cases}$$

Where, D = Dataset, w = Weight vector and b = Bias

In this case, the activation and label are multiplied and the minimum activation point is determined.

Similarly, margin of the dataset can also be calculated by considering every possible (w, b) pair. Among all these pairs, the largest one is selected as the overall margin of the dataset. It is generally denoted by the greek letter gamma (γ).

$$\therefore \quad \text{Margin } (D) = \max_{(w,b)} \text{Margin}(D, w, b)$$

Perceptron Convergence Theorem

Consider that a linearly separable data set 'D' on which the perceptron algorithm is run with margin (γ) > 0. Assume that $\|x\| \leq 1 \ \forall \ x \in D$. In this case, the algorithm will converge after atmost $\frac{1}{\gamma^2}$ updates.

Proof

Given that, the dataset is linearly separable with margin (γ). Then there exists a weight vector (w^*) which achieves this margin.

Consider a set of parameters (x^*) with margin > 0.

Let weight vector, $w = \{w^{(0)}, w^{(1)}, w^{(2)}, \ldots w^{(k)}\}$ for 0 to k updates.

Suppose that k^{th} update happens on example (x, y)

To show that $w^{(k)}$ is getting aligned with w^*,

Let $y\, w^{(k-1)}\, x < 0$ denote unclassified example.

From the first update, the result obtained is,

$$w^{(k)} = w^{(k-1)} + yx$$

$\Rightarrow$ $w^* w^{(k)} = w^*(w^{(k-1)} + yx)$ [definition]

$= w^* w^{(k-1)} + w^* yx$ [algebra notation]

$\geq w^* w^{(k-1)} + \gamma$ [w^* has margin]

This implies that for every update, its projection into w^* increases by atleast γ.

$\therefore$ $w^* w^{(k-1)} \geq k\gamma$ ... (1)

Now, compute the norm of $||w||$ and prove that the increase in marginal value is because, $w^{(k)}$ is getting closer to the w^*.

$\Rightarrow$ $||w^{(k)}||^2$

$= ||w^{(k-1)} + yx||^2$ [definition]

$= ||w^{(k-1)}||^2 + ||x||^2 y^2 + 2y\, w^{(k-1)} x$ [$a^2 + b^2 = a^2 + b^2 + 2ab$]

$= ||w^{(k-1)}||^2 + 1 + 0$ [Assumption]

$\therefore$ $||w^{(k)}||^2 \leq k$

$\sqrt{k} \geq ||w^{(k)}||^2$ ... (2)

Now solving the two statements [1 & 2]

$\Rightarrow$ $\sqrt{k} \geq ||w^{(k)}|| \geq w^* w^{(k)} \geq k\gamma$

$\Rightarrow$ $\sqrt{k} \geq k\gamma$ [divide both the sides by k]

$\Rightarrow$ $\frac{1}{\sqrt{k}} \geq \gamma$ [squaring on both sides]

$\Rightarrow$ $k \leq \frac{1}{\gamma^2}$

$\therefore$ Hence proved.

Q15. What is Geometric interpretation? Explain interpreting perceptron weights and linear separability.

Answer :

(Model Paper-II, Q14(a) | July/Aug.-22, Q9 [MGU])

Geometric Interpretation and Interpreting Perceptron Weights

For answer refer Unit-II, Page No. 32, Q.No. 13.

Linear Separability

For answer refer Unit-II, Page No. 33, Q.No. 14, Topic: Linear Separability.

2.1.4 Improved Generalization

Q16. Write about Improved Generalization: Voting and Averaging.

Answer :

Improved generalization can be defined as an approach that ensures the selected models work properly on the unseen data or real-time data. This can be done by using two methods. They are as follows,

1. Voting

In this method, various models are trained on the same data. The final prediction is calculated by aggregating the predictions of each individual model that is trained. In simple terms, the model prediction with higher votes will be considered as the final prediction.

Consider that $(w, b)^{(1)}$, $(w, b)^{(2)}$, ..., $(w, b)^{(k)}$ are different $(k+1)$ weight vectors evolved during training and $c^{(1)}, c^{(2)}, \ldots c^{(k)}$ are the survival times for each of the weight vectors. Therefore, the prediction on a specific test point can be given as,

$$\hat{y} = \text{sign}\left(\sum_{k=1}^{k} c^{(k)} \text{sign}\left(w^{(k)} . \hat{x} + b^{(k)}\right)\right)$$

This considered as voted perceptron that performs well compared to vanilla perceptron. But, the problem is that for each prediction of a model votings are required that eventually needs storage. Therefore, the prediction will be slower than vanilla prediction.

2. Averaging

In this method, the final prediction can be obtained by averaging all the outputs of multiple models. This method is mostly used when the individual models are very similar and can produce similar predictors.

Consider that $(w, b)^{(1)}$, $(w, b)^{(2)}$, ..., $(w, b)^{(k)}$ are different $(k+1)$ weight vectors evolved during training and $c^{(1)}, c^{(2)}, \ldots c^{(k)}$ are the survival times for each of the weight vectors. Therefore, the prediction on a specific test point can be given as,

$$\hat{y} = \text{sign}\left(\sum_{k=1}^{k} c^{(k)} \left(w^{(k)} . \hat{x} + b^{(k)}\right)\right)$$

This can be written as,

$$\hat{y} = \text{sign}\left(\left(\sum_{k=1}^{k} c^{(k)} w^{(k)}\right) . \hat{x} + \sum_{k=1}^{k} c^{(k)} b^{(k)}\right)$$

In averaged perceptron, one can also have the running sum and averaged bias. They are,

$$\text{Running sum} = \sum_{k=1}^{k} c^{(k)} w^{(k)}$$

$$\text{Averaged bias} = \sum_{k=1}^{k} c^{(k)} b^{(k)}$$

2.1.5 Limitations of the Perceptron

Q17. List the limitations of perceptron.

Answer :

The various limitations of perceptron are as follows,

1. The perceptron can only learn linearly separable functions. In simple terms, it can only classify the data points that are separated by a hyperplane on straight line.
2. It can model only the linear relationship between inputs and outputs.

Consider a XOR problem to explain the limitations of perceptron. The XOR problem is as shown in below figure,

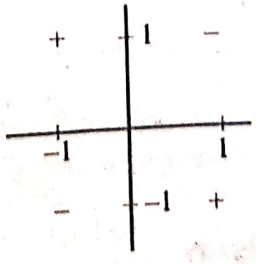


Figure: XOR Problem

The above figure consists of four data points at each corner where the diagonal points are same. Here, the user can try but cannot determine a linear decision boundary the exactly separates four data points.

Real-time Example

Consider real-time example similar to XOR problem. This problem consists of three features are used to find whether a review of a course consists of a spacific word or not. The features are,

(i) **EXCELLENT:** It represents positive reviews.

(ii) **TERRIBLE:** It represents negative reviews.

(iii) **NOT:** It is used to flip the reviews, considered as categorization flips.

This problem can be addressed by adding the features like EXCELLENT-AND-NOT, TERRIBLE-AND-NOT. The required effects can be achieved by assigning the following weights,

$w_{\text{EXCELLENT}} = +1$

$w_{\text{TERRIBLE}} = -1$

$w_{\text{NOT}} = 0$

$w_{\text{EXCELLENT-AND-NOT}} = -2$

$w_{\text{TERRIBLE-AND-NOT}} = +2$

Generally, inorder to address the problem with D-features. The feature mapping can be used, where the D-many features are combined as $\binom{D}{2} = O(D^2)$.

2.2 PRACTICAL ISSUES

2.2.1 Importance of Good Features, Irrelevant and Redundant Features

Q18. Explain the importance of good features.

Answer :

Features can be defined as the attributes that are used to represent the data and are used by the classification model to make predictions. Good features can significantly improve the accuracy and generalization ability of a machine learning model, while poor features can lead to poor performance. The importance of good features can be given as follows,

1. **To Capture the Relevant Information in the Data**

   Features that are informative and relevant to the task helps the model to capture the underlying patterns in the data and make accurate predictions. On the other hand, irrelevant or noisy features can confuse the model and reduce its performance.

2. **To Reduce Overfitting**

   Overfitting occurs when a model learns to fit the training data too well and fails to generalize to new data. Good features can help to reduce overfitting by providing the model with a more compact and meaningful representation of the data.

3. **To Improve Interpretability**

   In some applications, it is important to understand how the model is making its predictions. Good features can help to make the model more interpretable by providing meaningful and understandable input to the model.

4. **To Reduce Computational Complexity**

   Machine learning models can be computationally expensive to train and evaluate, especially when dealing with large datasets. Good features can help to reduce the computational complexity of the model by providing more compact and efficient representation of the data.

Q19. Discuss irrelevant and redundant features.

Answer :

Irrelevant Features

Irrelevant features can be defined as the features that are not relevant or uncorrelated for the prediction task. A feature having an expectation that is independent on the label is considered as irrelevant feature.

Example

The inclusion of the term "the" in a movie review to predict it as positive or negative.

Redundant Features

Redundant features can be defined as the features that are highly correlated for the prediction task. It consists of information more than required.

Example

In an image, having a dark red pixel at position (10, 20) is redundant while already having a dark red pixel at position (11, 20). Here, both the pixels are useful but one is unwanted depending on the situation.

Note that, it is ok to have one bad feature over 100 features. But, it creates difficulty only when the number of bad features exceeds good and correlated features.

In a shallow decision tree, the only correlated features are being selected. This model can kick out the uncorrelated features by limiting its depth.

Real-time Example (KNN)

The k-nearest neighbors (KNN) weights every features as much as another feature.

In turn, it messes up the KNN prediction. In a high-dimensional space, the data points are apart by small distances. If some randomly distributed features are added to the datasets, then also the distances will converge. For a perceptron, one can assign zero weight to irrelevant features. So that there would not be an effect on the performance.

2.2.2 Feature Pruning and Normalization, Combinatorial Feature Explosion

Q20. Discuss about feature pruning and normalization, combinatorial feature explosion.

(Model Paper-III, Q14(a) | July/Aug.-22, Q10 [MGU])

Answer :

Feature Pruning

Feature pruning can be defined as a process of eliminating the number of unwanted/irrelevant input features in a model. It helps in removing the irrelevant features and concentrating on the required features.

Normalization

Normalization can be defined as a process of making the data consistent. It enhances the convergence rate of training algorithm and stops the domination of one feature over another. The main objective of normalization is to make the process of learning an algorithm easier. There are two types of normalization. They are as follows,

1. Feature Normalization

Feature normalization involves in rescaling the each of features in a dataset.

The two standard things to do in this normalization are as follows,

- **Centering:** Moving the complete data set to make it centered around the origin. This makes sure that none of the feature is arbitrarily large.
- **Scaling:** It rescales each feature value so that all features contain equal values. Rescaling each of the feature to hold the following,

(i) In training data, each feature have variance 1

(ii) In training data, each feature has higher absolute value 1.

This can be mathematically represented as,

For centering, $x_{n,d} \leftarrow x_{n,d} - \mu_d$ $\left[\because \mu_d = \frac{1}{N}\sum_n x_{n,d}\right]$

For variance scaling, $x_{n,d} \leftarrow \frac{x_{n,d}}{\sigma_d}$ $\left[\because \sigma_d = \sqrt{\frac{1}{N-1}\sum_n (x_{n,d} - \mu_d)^2}\right]$

For absolute scaling, $x_{n,d} \leftarrow \frac{x_{n,d}}{r_d}$ $\left[\because r_d = \max_n |x_{n,d}|\right]$

Here $x_{n,d}$ represents the d^{th} feature of n^{th} sample.

Note that, if the scaling difference in a data point is significant. Then, the process of rescaling may remove some necessary information.

2. Example Normalization

Example normalization involves in rescaling each example independently. This method ensures that the length of each vector is one. That means, the sample lies in the unit hypersphere. This is considered as a simple transformation. This can be represented as,

$$x_n \leftarrow \frac{x_n}{\|x_n\|}$$

The main advantage of example normalization over feature normalization is that the example normalizations performs the comparisons clearly straight across all the data sets.

Combinatorial Features Explosion

In machine learning, combinatorial feature explosion can be considered as a problem that araise when the possible feature combinations increases rapidly and creates difficulties in analyzing the data. This leads to overfitting of data and reduces the performance of generalization.

Example

Consider a data set with n features, each having m different values then, the possible feature combinations will be m^n. For instance, consider a data set with 10 features each having 10 different values, then the possible combination of features is equal to 10 billions which is large.

This problem can be achieved by using the concept of feature pruning or selection.

2.2.3 Evaluating Model Performance

Q21. How to evaluate the performance of a model? Give an example.

Answer : Model Paper-I, Q14(b)

Evaluation Performance Model

The performance of a classification model can be evaluated by two metrics known as precision metric and recall metric.

The precision metric determines the fraction of true positive prediction a among all the positive predictions assumed by a classification model. In simple terms, it calculates the number of positive predictions made by the model that are correct. This can be mathematically represented as,

$$P = \frac{\text{True positives}}{(\text{True positives} + \text{False positives})}$$

The recall metric determines the fraction of true positives predictions among all the actual/correct positive predictions in the data. In simple terms, it calculates the number of positive predictions that can be correctly identified by the model. This can be mathematically represented as,

$$R = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

Note that, the two metrics precision and recall are inversly proportional to each other. The increase in the precisions leads to the decrease in recall and vice versa. Thus, one should manage the balance between precision and recall depending on the problem and its associated costing.

In some specific cases, the performance of a model can be evaluated by using the combination of precision and recall. This can be done by considering their harmonic mean. Mathematically, this can be represented as,

$$F(F\text{-score}) = \frac{2 \times P \times R}{P + R}$$

When the precision is more important than the recall, the weighted f-measure can be introduced. This can be represented as,

$$F_\beta = \frac{(1+\beta^2) \times P \times R}{\beta^2 \times P + R}$$

Example

Consider a binary classification problem where the user wants to predict whether a customer buy a specific product based on the demographic and behavioral data.

Assume that,

Number of instances in data set = 1000

Negatively labelled items = 700

Positively labelled items = 300

After the evaluation using the separate test data, the confusion matrix can be given as,

	Predicted Positive	Predicted Negative
Actual positive	70	30
Actual negative	50	200

From the above data,

$$\text{Precision, } P = \frac{\text{True positives}}{(\text{True positives}) + (\text{False positives})}$$

$$= \frac{70}{70+50} = \frac{70}{120} = 0.583$$

Precision, $P = 0.583$

$$\text{Recall, } R = \frac{\text{True positives}}{(\text{True positives}) + (\text{False negatives})}$$

$$= \frac{70}{70+30} = \frac{70}{100}$$

Recall, $R = 0.7$

Therefore,

- The recall of the model concludes that 70% of positive predictions can be correctly identified.
- The precision of the model concludes that 58.3% of positive predictions made by the model are correct.

On the other hand, there also exists another metric to evaluate the performance called as sensitivity/specificity metric. Here, the sensitivity metric finds every thing that it requires where as the specificity metric do not find the jobs that are not required.

2.2.4 Cross Validation, Hypothesis Testing and Statistical Significance

Q22. Explain about cross-validation in machine learning.

Answer :

Cross validation is used to evaluate the performance of a classification model. This technique splits the data into various sub-sets referred as folds. Then, one fold out or all folds can be considered as test and the remaining are considered as training data. It helps in training and testing all the data sets and estimates the performance of a model.

The k-fold crossvalidation can be considered as the general cross-validation technique. Here, the dataset is equally divided into k sub-parts. Thus, the model is trained k times, with each fold used as the validation set once, and the results are averaged over all the iterations to get the final estimate of the performance of a model.

Algorithm

The algorithm for k-fold cross validation can be given as,

Algorithm: CrossValidate(Learning Algorithm, Data, K)

$\hat{\epsilon} \leftarrow \infty$

$\hat{\alpha} \leftarrow$ unknown

for all hyperparameters settings α do;

err $\leftarrow$ [ ]

for k = 1 to K do

train $\leftarrow \{(x_n, y_n) \in$ Data : n mod K $\neq$ k $- 1\}$ //training test sample

test $\leftarrow \{(x_n, y_n) \in$ Data : n mod K = k $- 1\}$ //testing test sample

model $\leftarrow$ Run Learning Algorithm on "train"

err $\leftarrow$ err $\oplus$ error of model on "test" //add to the error list

end for

averageError $\leftarrow$ mean of set err

if averageError $\leftarrow \hat{\epsilon}$ then:

$\hat{\epsilon} \leftarrow$ averageError

$\hat{\alpha} \leftarrow \alpha$

end if

end for

Generally, the k value can be 8, 5, 10 and $N-1$. Here, the case where the value of $K = N-1$ can be considered as "Leave-one-out cross validation (100 cross validation)". The algorithm for KNN-training LOO cross validation can be given as,

Algorithm:KNN-TRAIN-LOO(D)

$\text{err}_k \leftarrow 0, \forall\ 1 \le k \le N-1$

for $n = 1$ to N

do

$S_m \leftarrow \langle \|x_n - x_m\|, m \rangle\ , \forall m \neq n$

$S \leftarrow$ SORT(S)

$\hat{y} \leftarrow 0$

for $k = 1$ to $N-1$ do

$\langle \text{dist}, m \rangle \leftarrow S_k$

$\hat{y} \leftarrow \hat{y} + y_m$

if $\hat{y} \neq y_m$ then

$\text{err}_k \leftarrow \text{err}_k + 1$

end if

end for

end for

return $\text{argmin}_k\ \text{err}_k$ //The minimum k

Q23. Write about hypothesis testing and statistical significance.

Answer :

In machine learning, hypothesis testing and statistical significance are used for evaluating the performance of a model and determining if the results are meaningful.

Hypothesis Testing

The hypothesis testing starts with a null hypothesis, which is a statement that there is no significant difference between two populations or that there is no relationship between two variables. Then, the user needs to collect data and perform statistical tests to determine if the data provides enough evidence to reject the null hypothesis in favor of an alternative hypothesis. The statistical test depends on the type of data being used in the model. The two different examples tests in hypothesis testing are as follows,

1. **Paired t-test**

Paired t-test is used to compare the means of two related samples. This test is commonly used in machine learning to compare the performance of two models on the same dataset.

2. **Bootstrapping**

Bootstrapping is a resampling method that is used to estimate the sampling distribution of a statistic. In machine learning, bootstrapping is often used to estimate the uncertainty of a model's performance.

Both paired t-test and bootstrapping are useful techniques for hypothesis testing in machine learning. They help to determine if there is a significant difference between two models or if the performance of a model is significantly different from the expected one.

Statistical Significance

Statistical significance refers to the likelihood that the results users obtained are not due to chance. In machine learning, the p-values are used to determine statistical significance. A p-value is the probability of observing a test statistic as extreme or more extreme than the one we observed, assuming the null hypothesis is true. If the p-value is small enough (typically less than 0.05), the null hypothesis is rejected and the user can conclude that the results are statistically significant.

2.2.5 Debugging Learning Algorithms, Bias Variance Tradeoff

Q24. Explain debugging learning algorithms.

Answer :

Debugging learning algorithm can be defined as a process of identifying and fixing errors/bugs that causes the algorithms to show case the poor performance and worst predictions.

Generally, the learning algorithms are hard to debug. But, there exists some steps in the design process of learning algorithm. These steps include data collection, feature selection, model family selection training data and model selection and the evaluation of test data. There is a possibility of errors while performing the above mentioned steps. These errors are isolated using the various strategies. They are as follows,

1. **Is the Problem with Generalization to Test Data?**

Generally, it is impossible and unrealistic to expect that the algorithm works well on test data over the training data. If the user thinks and assures that the specific algorithm works well on the training data then the problem is with the generalization (because of using complicated models). Otherwise, the problem is with the representation.

2. **Do you have Train/Test Data Mismatch?**

If the algorithms works well with the training data but not with the test data, then exchange the training and testing data. In that case, if the result is good then the problem is with the input test data. Otherwise, the model may has another generalization problem.

3. **Is your Learning Algorithm Implemented Correctly?**

The algorithm may result in error when the implementation is not done properly. In such cases, review the implementation of algorithm.

4. Do you have an Adequate Representation?

If the training data is not fit that means there is no accurate feature set. This kind of problem can be solved by adding the required features.

5. Do you have Enough Data?

Initially perform the test with 80% of training data. If the performance is poor try on increasing the test data and re-perform the test.

Q25. Explain about bias/variance trade-off.

Answer :

Bias/Variance Trade-off

The bias/variance trade-off can be defined as the trade-off between the estimation error and approximation error. Here, estimation error is variance and approximation error is bias.

Generally, the test errors can be decomposed into two terms called estimation error and approximation error. Consider that f is a learned chosen from a set of learned classifiers $\mathcal{F}$. Then,

$$\text{error}(f) = \underbrace{\left[\text{Error}(f) - \min_{f^* \in \mathcal{F}} \text{error}(f^*)\right]}_{\text{Estimation Error}} + \underbrace{\left[\min_{f^* \in \mathcal{F}} \text{Error}(f)\right]}_{\text{Approximation Error}} \quad \ldots (1)$$

Here,

- The first term i.e., estimation error defines the distance between the actual learned classifier f, and the optimal classifier f^*.
- The second term i.e., approximation error defines the quality (like depth of decision tree) of model family.

Note that, the estimation error and approximation error are calculated only in the constructed cases.

Additionally, there exists a basic trade-off between the estimation error and approximation error. As the representation is more complex, the value of $\mathcal{F}$ increases. Eventually, this leads to the decrease in approximation error as there are more functions. But, the estimation error increases because more parameters are added to the fit and leads to overfitting.

Example

Consider a hypothesis class $\mathcal{F}$ having two functions i.e., a positive classifier and a negative classifier. For instance, we have a data generating distribution function, D which is 60% positive and 40% negative examples. In case of drawing 41 examples, there is a 90% chance that we get positive. Therefore, there is 90% probability that we get "all positive" classifier. This is considered as low variance as a function of randomly drawing examples from training data.

Additionally, the learned classifier is very insensitive to the given input example. It is considered as strong biased towards predicting the +1 even the input is opposite.

2.3 LINEAR MODELS

2.3.1 The Optimization Framework for Linear Models

Q26. Write about optimization framework for linear models.

Answer :

The optimized framework for linear models can be obtained using the perceptron. The main objective of perceptron is to determine a separating hyperplane for a given training data set. But, not every training data set is linearly separable. Therefore, in such cases a hyperplane having lower probability of resulting errors on the data set need to be determined. This can be represented as mathematical optimization problem as given below,

$$\min_{w,b} \sum_{n} 1\left[y_n(w \cdot x_n + b) > 0\right] \quad \text{(Here, } w \text{ is the weight vector } b \text{ is the bias)} \quad \ldots (1)$$

In the above expression, the optimization is done among two variables w, b. The objective function defines the error rate (i.e., 0 or 1) in terms of w and b. Here, the expression $1[.]$ represents an indicator function. It results one when the [.] values is true, otherwise false.

There are two different cases in finding the optimized solution. They are as follows,

Case1: When the data set is linearly separable (i.e., the optimum of equation (1) is zero) then the efficient parameters can be easily determined for the model.

Case2: When the data set is not linearly separable, then the optimum can be determined by changing the satisfied problem into an optimization problem where the dataset is properly separable by the hyperplane. Note that, there is no alternate way to determine the small constant worst than the optimal solution (here, the optimal solution is, $418/415 \approx 1.007$).

In general, the equation (1) can be optimized by minimizing the error rate, or by obtaining minimum training error. This can be done by generalizing the testing data and by not over fitting the data. These all requirements can be achieved by introducing a regularizer 'R' over the model parameters i.e., w and b. This can be called as an arbitrary function and can be represented as $R(w, b)$. Therefore, the regularized objective can be given as,

$$\min_{w,b} \underbrace{\sum_{n} 1\,[y_n(w.x_n + b) > 0]}_{1^{st}\text{ term}} + \underbrace{\lambda R(w, b)}_{2^{nd}\text{ term}} \quad \begin{cases} R \rightarrow \text{Regularization for hyperplaner} \\ \lambda \rightarrow \text{Hyperparameter for optimization} \end{cases} \quad \text{... (2)}$$

The above equation is used to optimize the trade off between the solution that provides lower training error (1^{st} term) and the simple solution (2^{nd} term).

2.3.2 Convex Surrogate Loss Functions

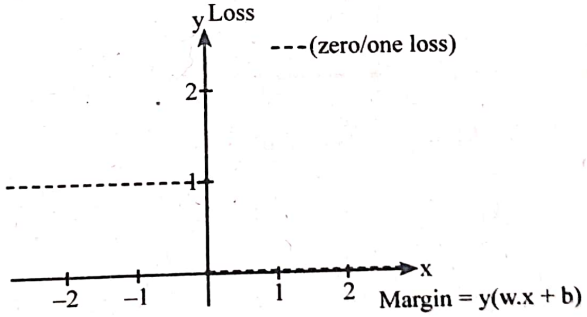
Q27. Discuss convex surrogate loss functions in machine learning.

Answer : Model Paper-II, Q14(b)

Convex Surrogate Loss Functions

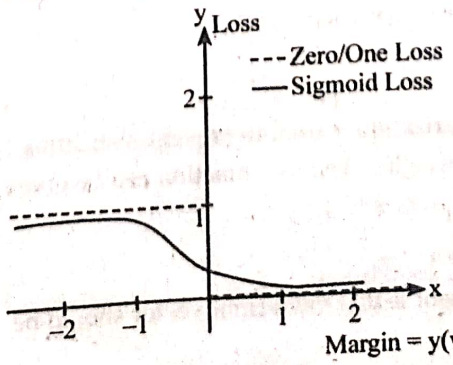
In machine learning, the convex surrogate loss functions are used when the true loss function is not convex but a convex function is used as an alternative for optimization purpose. The main reason behind using convex function is that they contain unique global minimum that helps the optimization process simple and efficient.

The value of optimization depends on the values of w and b. A small changes in the values of these parameters create a large difference in the optimization function. For instance, consider the below graph.



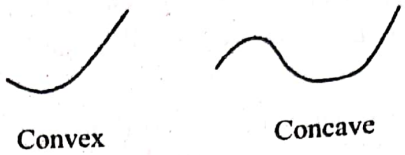
The above graph represents margin vs loss(zero/one loss). The x-axis represents the margin of a data point and y-axis represents the loss associated with the margin (zero/one loss). Here, if the margin $[y(w.x + b)] > 0$, then the loss is zero. Otherwise it is one. Thus, one can notice that the small change in the values of these parameters can have huge impact on the loss.

This specific problem can be resolved by considering a smooth approximation instead of non-smooth zero/one loss. This can be done by making a S-shaped function as shown in the below plot,



The main objective of considering S-function is that it is smooth and helps in optimizing process to make it simple. But, the complexity is that it is not convex.

The convex function looks like a happy face (⌣) and the concave function looks like a sad face (⌢). The examples of convex and concave functions are as shown in the below figure,



Convex Concave

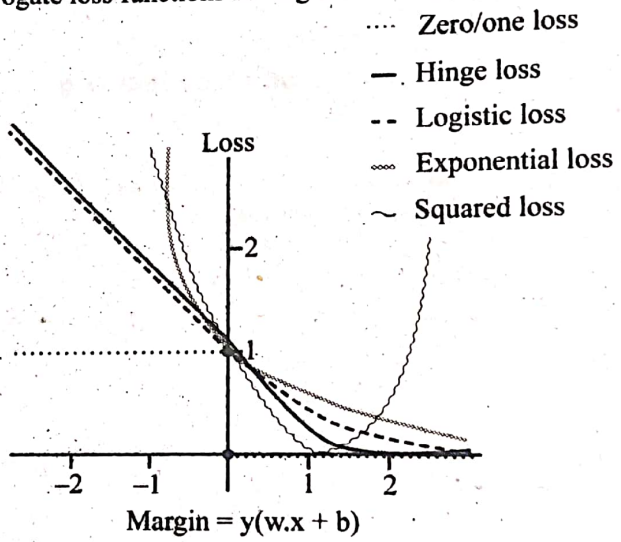
Figure: Convex and Concave Functions

Using the convex function, the optimization becomes easier. For instance, when a ball thrown into the convex, it reaches the minimum point only.

As the optimization of zero/one loss, one can consider the approximate zero/one loss with convex function and this convex function is referred to as "surrogate loss". It always ensures the minimum.

Common Surrogate Loss Functions

The four common surrogate loss functions having their own properties are as represented below,



Zero/one: $l^{(0/1)}(y, \hat{y}) = 1[y\hat{y} \le 0]$

Hinge: $l^{(hin)}(y, \hat{y}) = \max\{0, 1 - y\hat{y}\}$

Logistic: $l^{(log)}(y, \hat{y}) = \frac{1}{\log 2} \log(1 + \exp[-y\hat{y}])$

Exponential: $l^{(exp)}(y, \hat{y}) = \exp[-y\hat{y}]$

Squared: $l^{(sqr)}(y, \hat{y}) = (y - \hat{y})^2$

2.3.3 Weight Regularization, Optimization and Gradient Descent

Q28. Explain the concept of weight regularization.

Answer :

In machine learning, weight regularization is used to prevent overfitting by adding the regularization term to the loss function that penalizes higher weights. The loss function can be given as,

$$\min_{w,b} \sum_{n} l(y_n, w.x_n + b) + \underbrace{\lambda R(w, b)}_{\text{Convex function}}$$

Here, the most important requirement is that the weight vector should be zero or closer to zero. This form is considered as "Inductive bias".

Example

Consider a data sample x with label +1. Then, assume that, x' is a neighbor io x with label +1. Suppose a user get x' by considering x by only modifying the first component with ϵ, the user can think that it has the same classification. The difference between y and $\hat{y}$ is that ϵw_1, which is very small say almost equal to zero. Thus, it does not have any impact. If w_1 is large, then the classification may vary.

In another way, this can be said that derivating the function "$w.x + b$" with respect to w_1 can be given as,

$$= \frac{\partial[w.x+b]}{\partial w_1} = \frac{\partial[\Sigma_d w_d x_d + b]}{\partial w_1} = x_1$$

Here, one can observe that the rate of change of prediction function is directly proportional to independent weights.

All these requirements can be achieved by using the norm for weight vector i.e.,

$$R^{(\text{norm})}(w, b) = \|w\|$$
$$= \sqrt{\Sigma_d w_d^2}$$

The squared norm can be given as,

$$R^{(\text{sqr})}(w, b) = \|w\|^2$$
$$= \Sigma_d w_d^2$$

The approach for the above equation is to use the sum of absolute weights. This can be represented as,

$$R^{(\text{abs})}(w, b) = \Sigma_d |w_d|$$

Note that, all the norms are convex. Consider the weight w_d is equal to zero. Then, the alternative regularizer can be given as,

$$R^{(\text{cnt})}(w, b) = \Sigma_d 1[x_d \neq 0]$$

In general, this concept refers to p-norms. Then, the $\|w\|_p$ can be represented as the p-norm of w.

$$\|w\|_p = \left(\sum_d |w_d|^p\right)^{\frac{1}{p}}$$

The value of p may varies from one assumption to another. One can interpolate between a square (known as "max-norm") down to a circle(2-norm), diamond (1-norm) and pointy-star-shaped-thing ($p < 1$ norm) by modifying the value of p.

Q29. Write about the use of gradient methods of optimization.

Answer :

The gradient methods of optimization is used when the user wants to find the maximum and minimum of a function $f(x)$. In this case, the gradient descent method that determines the minimum of a function is used. Here at each step, the gradient of a function that is trying to optimize the value is measured. This can be represented as,

$x \leftarrow x + \eta g$ [$\because$ Here, η is the size of each step]

Algorithm: GradientDescent($\mathcal{F}$, K, $\eta_1, \ldots$)

Input: $\mathcal{F}$ is a function needs to be minimized

K is the number of iterations

$\eta_1, \eta_2, \ldots \eta_k$ are the sequence of learning rates.

Step1: Initialize the variables that needs to be modified

$z^{(0)} \leftarrow <0, 0, \ldots 0>$

Step2: for $k = 1$ to K do:

$g^{(k)} \leftarrow \Delta_z \mathcal{F}|_{z^{(k-1)}}$ //Computing gradient at present location

$z^{(k)} \leftarrow z^{(k-1)} - \eta^{(k)} g^{(k)}$ //Next lower iteration to gradient

end for

Setp3: return $z^{(k)}$

The gradient descent can be applied by computing the derivatives. Inorder to get the accuracy. Consider the loss function with 2-norms as regularizer. Then, the regularized object can be given as,

$$\mathcal{L}(w, b) = \sum_n \exp[-y_n(w.x_n + b)] + \frac{\lambda}{2} ||w||^2 \quad \left[\because \lambda \text{ is replaced by } \frac{\lambda}{2} \text{ to make the gradient clear}\right]$$

Now, compute derivative w.r.t b

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial}{\partial b}\sum_n \exp[-y_n(w \cdot x_n + b)] + \frac{\partial}{\partial b}\frac{\lambda}{2}||w||^2$$

$$= \sum_n \frac{\partial}{\partial b}\exp[-y_n(w \cdot x_n + b)] + 0$$

$$= \sum_n \left(\frac{\partial}{\partial b} - y_n(w \cdot x_n + b)\right)\exp[y_n(w \cdot x_n + b)]$$

$$= -\sum_n y_n \exp[-y_n(w \cdot x_n + b)]$$

In this case, the optimization can be operated by updating,

$$b \leftarrow b - \eta \frac{\partial \mathcal{L}}{\partial b}$$

If the positive examples are considered, then the prediction '$w.x_n + b$' will be large and tends to ∞. The infinity interms of exp[ ] results in 0. Therefore, derivation w.r.t bias cannot work. Now, derive the equation w.r.t w,

$$\Delta_w \mathcal{L} = \Delta_w \sum_n \exp[-y_n(w \cdot x_n + b)] + \Delta_w \frac{\lambda}{2}||w||^2$$

$$= \sum_n (\Delta_w - y_n(w \cdot x_n + b))\exp[-y_n(w \cdot x_n + b)] + \lambda w$$

$$= -\sum_n y_n x_n \exp[-y_n(w \cdot x_n + b)] + \lambda w$$

In this case, the optimization can be operated by updating,

$$w \leftarrow w - \eta \Delta_w \mathcal{L}.$$

The gradient descent value will be closer to zero or zero for properly classified points. Whereas for poorly classified points, the gradient points in $-y_n x_n$ direction and the updated form will be $w \leftarrow w + cy_n x_n$ [here, c is a constant].

By considering the gradient w.r.t. the regularizer, the updation is,

$$w \leftarrow w - \lambda w.$$

$$w \leftarrow w(1 - \lambda)$$

This has a huge effect of shrinking the weight value to 0.

Consider the below figure that shows good and bad step sizes.

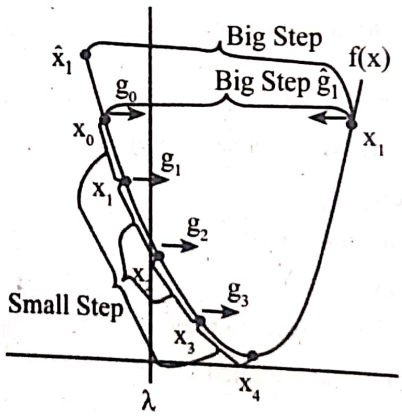


Figure: Good and Bad Step Sizes

Here,

- If the step size is big then the optimum value will be exceeded and ends up with oscillations.
- If the step size is small then the time consumed to find the optimum value becomes more.
- If the step size is properly chosen, then the optimum solution can be determined at fast service.

2.3.4 Support Vector Machines

Q30. Explain support vector machines.

Answer :

(Model Paper-III, Q14(b) | June/Aug.-22, Q14(b) [OU])

Support Vector Machine (SVM) is a supervised machine learning algorithm that is used for classification and regression tasks. The main objective of SVM is to find a hyperplane in a high-dimensional space that separates the classes with maximum margin. Here, margin refers to the distance between the hyperplane and the closest data points in every class. This can be considered as a constrained optimization problem and written as,

$$\min_{w,b} \frac{1}{\gamma(w,b)} \quad \ldots (1)$$

The above equation subjects to $y_n(w.x_n + b) \geq 1$ (for all n)

Here, γ represents the margin

w, b represents the parameters weight, bias respectively.

The maximum margin can be determined by using the minimum reciprocal of the margin.

The difficulty faced in the optimization problem is that the classification of every data point should be greater than one instead of greater than zero. This eventually makes the data as not linearly separable. Therefore, there won't be any feasible solution that means no set of parameters w, b be determined to satisfy all the required constraints. This case is referred as hard-margin SVM.

Inorder to change the optimization problem to manage the data that is not linearly separable, the slack parameters are used. The main objective of slack parameters is to allow the misclassification of data points while training process. They help in determining the optimal decision boundary that separates the classes of data points.

By using the slack parameters, a new objective function is created referred as soft-margin SVM. This can be represented as,

$$\min_{w,b,\xi} \underbrace{\frac{1}{\gamma(w,b)}}_{\text{Large margin}} + \underbrace{C\sum_n \xi_n}_{\text{small slack}} \quad \ldots (2)$$

The above equation subjects in,

$$y_n(w.x_n + b) \geq 1 - \xi_n \quad (\forall n)$$

$$\xi_n \geq 0 \quad (\forall n)$$

The equation (2) ensures that all the data points are properly classified. Incase, if 'n' cannot be classified properly, then set the slack value (ξ_n) to a value more than "zero" to classify the data property. The slack value should be set for all incorrect classifications. Here the hyperparameter $C > 0$ handles the overfitting and underfitting.

The advantage of soft margin SVM is that, it never produce a empty feasible region. That means, there always be a solution irrespective of linearly separable and inseparable data.

Calculating the Size of Margin

Consider a positive example that is 1 unit far from the hyperplane say x^+. Thus, $w.x^+ + b = 1$. Consider a negative example that lies on the opposite side of margin, say x^-. Thus, $w.x^- - b = -1$. Now, the distance can be measured as,

$$d^+ = \frac{1}{\|w\|} w.x^+ + b - 1$$

$$d^- = -\frac{1}{\|w\|} w.x^- - b + 1$$

Using the algebra,

$$\gamma = \frac{1}{2}[d^+ + d^-]$$

$$\gamma = \frac{1}{2}\left[\frac{1}{\|w\|} w.x^+ + b - 1 + \left(-\frac{1}{\|w\|} w.x^- - b + 1\right)\right]$$

$$= \frac{1}{2}\left[\frac{1}{\|w\|} w.x^+ + b - 1 - \frac{1}{\|w\|} w.x^- - b + 1\right]$$

$$= \frac{1}{2}\left[\frac{1}{\|w\|} w.x^+ - \frac{1}{\|w\|} w.x^-\right]$$

$$= \frac{1}{2}\left[\frac{1}{\|w\|}(+1) - \frac{1}{\|w\|}(-1)\right]$$

$$= \frac{1}{2}\left[\frac{2}{\|w\|}\right]$$

$$\gamma = \frac{1}{\|w\|}$$

Therefore, the size of the margin is inversely proportional to the norm of weight vector. Thus, the margin value gets maximized when the $\|w\|$ value is minimized.

Now, the value of slack parameter can be given as,

$$\xi_n = \begin{cases} 0 & \text{if } y_n(w.x_n + b) \geq 1 \\ 1 - y_n(w.x_n + b) & \text{Otherwise} \end{cases}$$

Alternatively, one can say that the slack values is same as hinge loss. The SVM can be written as unconstrained optimization problem. That is,

$$\min_{w,b} \underbrace{\frac{1}{2}\|w\|^2}_{\text{Large margin}} + \underbrace{C\sum_n l^{(hin)}(y_n, x_n + b)}_{\text{small slack}}$$

INTERNAL ASSESSMENT

I. Multiple Choice

1. A dataset is said to be _____ if there exists a hyperplane (or) a line which separates the positive and negative examples. [ ]

   (a) Non-linearly separable (b) Linearly separable

   (c) Converging (d) Non-converging

2. _____ refers to the procedure of making the pass through training data without involving any update or modification. [ ]

   (a) Perceptron convergence (b) Perceptron initialization

   (c) Separability (d) Converging

3. _____ feature is used to flip the reviews considered as categorization flips. [ ]

   (a) EXCELLENT (b) TERRIBLE

   (c) BAD (d) DOT

4. _____ can significantly improve the generalization ability of a machine learning model. [ ]

   (a) Real features (b) Good features

   (c) Poor features (d) Complex features

5. In _____ method, the final prediction can be obtained by averaging all the outputs of multiple models [ ]

   (a) Voting (b) Converging

   (c) Averaging (d) Capturing

6. _____ normalization involves in rescaling the each of features in a dataset. [ ]

   (a) Feature (b) Example

   (c) Pruning (d) Data scaling

7. _____ problem arises when possible feature combination increases rapidly and create difficulties in data analyzing. [ ]

   (a) Combinatorial feature explosion (b) Pruning feature explosion

   (c) Overfitting features (d) All the above

8. _____ test is used to compare the means of two related samples. [ ]

   (a) S-test (b) Bootstrapping

   (c) Paired t-test (d) None of the above

9. _____ defines the distance between the actual learned classifier and the optimal classifier. [ ]

   (a) Approximation error (b) Bias error

   (c) Data error (d) Estimation error

10. In SVM, _____ refers to distance between the hyperplane and the closest data points in class. [ ]

    (a) Slack (b) Margin

    (c) Region (d) None of the above

II. Fill in the Blanks

1. _____ algorithm is error driven which does not update or modify itself until an error or problem occurs.
2. In _____ method, various models are trained on the same data.
3. _____ normalization involves in rescaling each example independently.
4. _____ metric determines the fraction of true positive prediction among all the positive predictions assumed by classification model.
5. In bias/variance tradeoff, approximation error is _____.
6. _____ function helps the optimization process simple and efficient.
7. _____ helps in removing the irrelevant features and concentrating on the required features.
8. In _____ standard, the complete dataset is moved to make it centered around the origin.
9. In _____ cross validation technique, the dataset is equally divided into k sub-parts.
10. In training data, the variance value of each feature is _____.

KEY

I. Multiple Choice

1. (b)
2. (a)
3. (d)
4. (b)
5. (c)
6. (a)
7. (a)
8. (c)
9. (d)
10. (b)

II. Fill in the Blanks

1. Perceptron
2. Voting
3. Example
4. Precision
5. Bias
6. Convex
7. Feature pruning
8. Centering
9. k-fold
10. 1

III. Very Short Questions and Answers

Q1. Define bio-inspired learning.

Answer :

The type of learning which works like the working of neurons in brain according to folk biology is referred as bio-inspired learning. A human brain consists of neuron units that send and receive electrical signals among all these units.

Q2. What are irrelevant features?

Answer :

Irrelevant features can be defined as the features that are not relevant or uncorrelated for the prediction task. A feature having an expectation that is independent on the label is considered as irrelevant feature.

Q3. Define bootstrapping.

Answer :

Bootstrapping is a resampling method that is used to estimate the sampling distribution of a statistic. In machine learning, bootstrapping is often used to estimate the uncertainty of a model's performance.

Q4. Write about debugging learning algorithm.

Answer :

Debugging learning algorithm can be defined as a process of identifying and fixing errors/bugs that causes the algorithms to show case the poor performance and worst predictions.

Q5. Write about recall metric.

Answer :

The recall metric determines the fraction of true positives predictions among all the actual/correct positive predictions in the data. In simple terms, it calculates the number of positive predictions that can be correctly identified by the model. This can be mathematically represented as,

$$R = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

FREQUENTLY ASKED QUESTIONS & IMPORTANT QUESTIONS

SHORT QUESTIONS

Q1. Discuss about importance of good features.

Answer : (Important Question | July/Aug.-22, Q5 [MGU])

For answer refer Unit-II, Page No. 27, Q.No. 1.

Q2. Explain about pruning and normalization.

Answer : (Important Question | June/Aug.-22, Q4 [OU])

For answer refer Unit-II, Page No. 27, Q.No. 3.

Q3. Explain about bias-variance trade-off.

Answer : (Important Question | June/Aug.-22, Q5 [OU])

For answer refer Unit-II, Page No. 28, Q.No. 6.

Q4. Write about convex surrogate loss function.

Answer : (Important Question | June/Aug.-22, Q6 [OU])

For answer refer Unit-II, Page No. 28, Q.No. 7.

Q5. What is optimization and gradient descent in linear models?

Answer : (Important Question | July/Aug.-22, Q6 [MGU])

For answer refer Unit-II, Page No. 28, Q.No. 9.

Q6. Write about support vector machine in brief.

Answer : (Important Question | July/Aug.-22, Q2 [MGU])

For answer refer Unit-II, Page No. 29, Q.No. 10.

ESSAY QUESTIONS

Q7. What is perceptron? Explain perceptron learning algorithm with example.

Answer : (Important Question | June/Aug.-22, Q14(a) [OU])

For answer refer Unit-II, Page No. 31, Q.No. 12.

Q8. What is Geometric interpretation? Explain interpreting perceptron weights and linear separability.

Answer : (Important Question | July/Aug.-22, Q9 [MGU])

For answer refer Unit-II, Page No. 34, Q.No. 15.

Q9. Write about Improved Generalization: Voting and Averaging.

Answer : Important Question

For answer refer Unit-II, Page No. 34, Q.No. 16.

Q10. Explain the importance of good features.

Answer : Important Question

For answer refer Unit-II, Page No. 36, Q.No. 18.

Q11. Discuss about feature pruning and normalization, combinatorial feature explosion.

Answer : (Important Question | July/Aug.-22, Q10 [MGU])

For answer refer Unit-II, Page No. 37, Q.No. 20.

Q12. How to evaluate the performance of a model? Give an example.

Answer : Important Question

For answer refer Unit-II, Page No. 38, Q.No. 21.

Q13. Explain about cross-validation in machine learning.

Answer : Important Question

For answer refer Unit-II, Page No. 39, Q.No. 22.

Q14. Explain debugging learning algorithms.

Answer : Important Question

For answer refer Unit-II, Page No. 41, Q.No. 24.

Q15. Discuss convex surrogate loss functions in machine learning.

Answer : Important Question

For answer refer Unit-II, Page No. 43, Q.No. 27.

Q16. Write about the use of gradient methods of optimization.

Answer : Important Question

For answer refer Unit-II, Page No. 45, Q.No. 29.

Q17. Explain support vector machines.

Answer : (Important Question | June/Aug.-22, Q14(b) [OU])

For answer refer Unit-II, Page No. 47, Q.No. 30.