Module - 5

LEARNING AND GENERALIZATION

Bias and Variance

Bias is a form of inaccuracy arising from erroneous assumptions about data, such as presuming linearity when the data really adheres to a complicated function. Transposed, variance is added with heightened sensitiveness to fluctuations in training data. This constitutes a form of error, as we aim to enhance our model's resilience to noise. Machine learning encompasses two categories of error. Irreducible error and reducible error. Bias and variance descent under the category of Reducible mistake.

Bias is described as the framework incapacity, resulting in a discrepancy or inaccuracy between the anticipated measure and the effective value. The discrepancies between existent or anticipated belief and expected values are referred to as error or bias error attributable to bias. Bias is a organized inaccuracy arising from erroneous presume in the machine learning procedure.

Let consider Y represent the genuine measure of a parameter, and Y^ denote an estimate of Y derived from a sample distribution of data. The bias of the estimate Y^ is expressed as follows:

$$\text{Bias}(\hat{Y}) = E(\hat{Y}) - Y$$

where $E(\hat{Y})$ is the expected value of the estimator $\hat{Y}$. It is the measurement of the model that how well it fits the data.

Low Bias

A minimal bias value indicates that fewer assumptions are made in constructing the target function. In this instance, the model will closely align with the training dataset.

High Bias

A high bias value indicates that numerous assumptions are used in constructing the target function. In this instance, the model will not closely align with the training dataset.

The high-bias model will fail to reflect the trend of the dataset. It is regarded as an under fitting model characterized by a high error rate. The cause is a highly simplified algorithm.

A linear regression model may exhibit considerable bias when the data demonstrates a non-linear connection.

Methods to reduce high bias in Machine Learning

Use a More Complex mode:

A primary cause of elevated bias is the overly simplistic model. It will fail to encapsulate the intricacy of the data. In such instances, we can enhance our model's complexity by augmenting the number of hidden layers in a deep neural network. Alternatively, we may employ a more sophisticated model such as Polynomial regression for non-linear datasets, Conventional Neural Networks (CNN) for image processing, and Recurrent Neural Networks (RNN) for sequence learning.

Increment the number of features:

Expanding the dataset with additional features will enhance the model's complexity. Enhance its capacity to discern the fundamental patterns within the information.

Reduce Regularisation of the model:

Regularisation methods like consider L1 or L2 can mitigate over fitting and enhance the model's generalization capacity. In cases with strong bias in the model, lowering the intensity of regularization or eliminating it entirely may enhance its operation.

Increment the size of the training data:

Augmenting the preparation data quantity can mitigate prepossess by offering the framework of a broader array of examples for learning from the dataset.

Variance

Variance specify the dispersion of information relational to its average. In Machine Learning Variance refers to the property to which the operation of a prediction framework fluctuates when disciplined on various sub sets of the preparation data. Variance refers to the models sensibility to variations in another subset of the preparation data set. That is, the extent to which it can adapt to the fresh sub set of the preparation sample.

Let us consider Y represent the actualized values of the target area variable and Y^ denote the anticipated belief of the target area variable. The variance of a framework is quantified as the expectable quantity of the squared deviation between predicted values and their expected value.

$$\text{Variance} = E[(\hat{Y} - E[\hat{Y}])^2]$$

where $E[\bar{Y}]$ is the expected value of the predicted values. Here expected value is averaged over all the training data.

Variance errors are either low or high-variance errors.

Variance mistakes can be classified as either low-level variance or higher variance errors.

Low variance:

Low variance indicates that the model exhibits reduced sensitivity to fluctuations in the training data, hence yielding consistent estimations of the target function across various subsets of data drawn from the same distribution. This exemplifies under fitting, when the model does not generalize effectively on either training or test data.

High variance:

High variance indicates that the model is very responsive to fluctuations in the training data, potentially leading to substantial alterations in the estimation of the target function when trained on various subsets of data from the same distribution. This exemplifies over fitting, wherein the model exhibits strong performance on training data yet falters on novel, unseen test data. It is overly fitted to the training data, resulting in poor performance on the new dataset.

Methods to Reduce the Variance in Machine Learning

Crosswise establishment involves partitioning the information into preparation and testing sets repeatedly, facilitating the detection of model over fitting or under fitting, and enabling hyper parameter tuning to mitigate variation.

Feature selection:

Selecting only the pertinent features will reduce the framework complexity. Moreover, it could diminish the variance erroneousness.

Regularization:

Let us consider L1 or L2 are Regularization can be employed to mitigate variance in machine learning methods.

Ensemble methods:

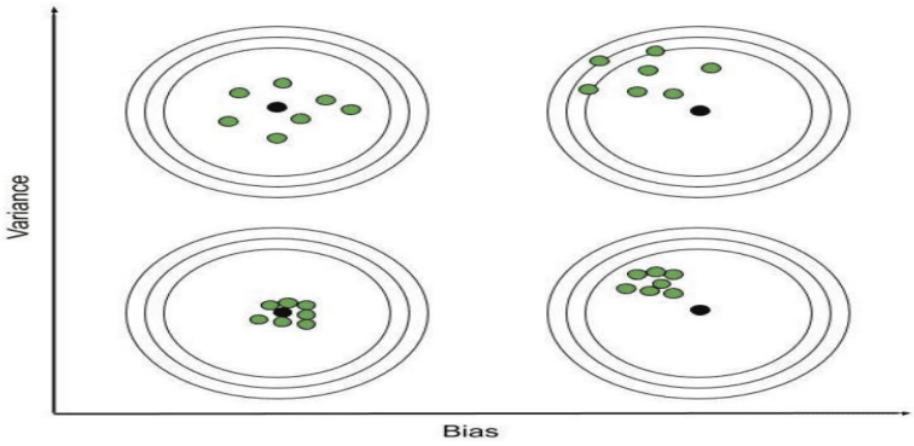
These will integrate different models to enhance generalization performance. Bagging, boosting, and stacking are prevalent ensemble techniques that can mitigate variation and enhance generalization performance.

Diminishing the intricacy of the framework, for instance, by lowering the amount of parametric quantity or layers in a ANN, might aid in mitigating variance and enhancing generalisation efficacy.

Early restricting is a method employed to avert over fitting by halting the preparation of the deep learning framework when operation on the establishment set ceases to enhance.

Different Combinations of Bias-Variance

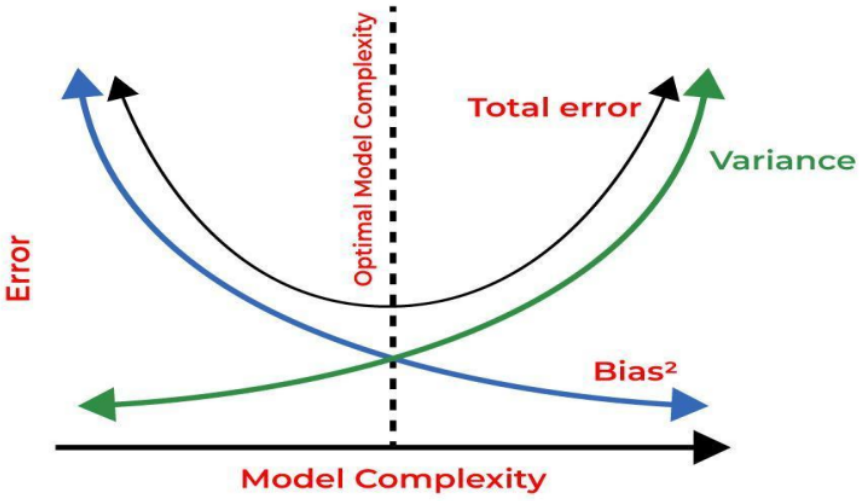
- **High Bias - Low Variance:** A model with high bias and low variance is said to be under fitting.
- **High Variance - Low Bias:** A model with high variance and low bias is said to be over fitting.
- **High-Bias - High-Variance:** A model has both high bias and high variance, which means that the model is not able to capture the underlying patterns in the data (high bias) and is also too sensitive to changes in the training data (high variance). As a result, the model will produce inconsistent and inaccurate predictions on average.
- **Low Bias - Low Variance:** A model that has low bias and low variance means that the model is able to capture the underlying patterns in the data (low bias) and is not too sensitive to changes in the training data (low variance). This is the ideal scenario for a machine learning model, as it is able to generalize well to new, unseen data and produce consistent and accurate predictions. But in practice, it's not possible.



Bias Variance Tradeoff

If the algorithm is overly simplistic (hypothesis represented by a linear equation), it may exhibit high bias and low variance, rendering it susceptible to errors. Conversely, if the algorithm is excessively complex (hypothesis represented by a high-degree equation), it may demonstrate high variance and low bias, resulting in poor performance on new data.

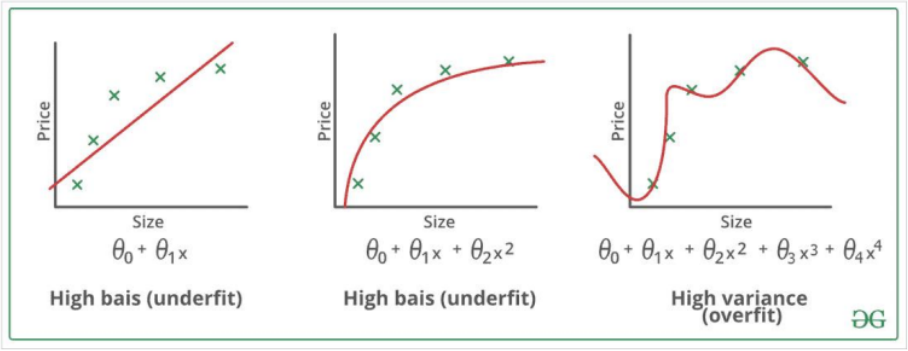
Between these extreme point lies the construct of the Bias & Variance Trade off, which elaborate the relationship between complexity of Bias & Variance. An algorithm cannot simultaneously possess both high complexity and low complexity. The optimal trade-off can be visually represented in a graph.



Bias and Variance

Bias denotes the inaccuracies that arise when attempting to apply a statistical framework to real world information that doesn't conform precisely to a mathematical model. Utilizing an excessively simplistic model to suit the data often results in high bias, which denotes the model's inability to discern structure in the available information, thereby guiding to sub part operation.

Variance denotes the error value that arises when predictions are made using data not previously encountered by the model. A phenomenon termed high variance arises when the model assimilates the interference inherent in the information.



Regularisation

Regularisation imposes a cost on more intricate models, thereby diminishing their complexity and promoting the acquisition of more universal patterns. This strategy achieves equilibrium between under fitting and over fitting, with under fitting arising when the framework is overly simplistic to discern the fundamental structure in the information resulting in low-level preparation and determination accuracy.

Role of Regularisation

Regularisation is a method employed to mitigate over fitting by incorporating a penalisation term into the failure mathematical function, so deterring the framework from attributing excessive significance to particular features or coefficients.

Regularization in Python

1. Complexity Control: Regularization mitigates model complexity by averting over fitting to training data, hence enhancing generalization to novel data.

2. Preventing Over fitting: One method to mitigate over fitting is by regularization, which imposes penalties on big coefficients and limits their magnitudes, thus preventing a model from becoming excessively complicated and memorizing the training data rather than discerning its fundamental patterns.

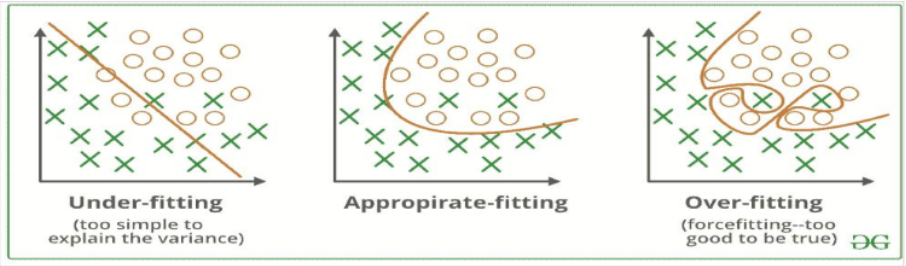
3. Balancing Bias and variance-off between model bias (underfitting) and model variance (overfitting) in machine learning, resulting in enhanced performance.

4. Feature Selection: Certain regularization techniques, such L1 regularization (Lasso), encourage sparse solutions that reduce some feature coefficients to zero.

This autonomously identifies significant traits while excluding those of lesser importance.

Addressing Multi col-linearity: When features exhibit high correlation (multi col-linearity), regularization can enhance model stability by diminishing constant sensibility to minor information fluctuations.

6. Generalisation: Regularised methods discern fundamental structure within information to enhance their applicability to novel data, rather than merely recalling individual instances.



Over fitting is a phenomena that arises when a Machine Learning model is overly tailored to the training set, resulting in poor performance on novel data. At that point, our model also assimilates the noise present in the training data. This occurs when our model retains the training data rather than discerning the underlying patterns.

Under fitting occurs when our model fails to recognize even the fundamental patterns present in the dataset. An under fitting model fails to perform adequately even on the training data, hence it cannot be anticipated to perform well on the validation data. This occurs when we are required to augment the model's complexity or expand the feature collection.

Train ANNs With Noise to Reduce Over fitting

ANNs have transformed AI, they frequently succumb to over fitting, which can diminish the models accuracy and reliability.

Training ANNs Utilizing Noise

With in the framework of ANNs, dissonance is characterized as stochastic or extraneous information that disrupts the framework capacity to identify target structure or state. Dissonance can negatively affect a model's learning efficiency, resulting in diminished performance and reduced accuracy.

Nevertheless, even a slight amount of noise can enhance neural network efficacy. The incorporation of unpredictability during training, referred to as noise injection, functions as a transformation element for the models.

Techniques for Noise Injection

Data augmentation is an effective approach employed to introduce noise into the data. Data augmentation may substantially diminish the generalization error frequently encountered in machine learning methodologies.

With sufficient preparation information, our machine learning framework can generalise more effectively. In certain real world scenarios, the restricted availability of information constrains the machine learning framework ability to generalise effectively. To address this issuance, we incorporate artificial data sometimes referred to as dissonance into the preparation set.

Gaussian dissonance is a widely employed method in data augmentation that introduces noise into input data, hence mitigating over fitting. It possesses a average of zero and an adjustable standardised deviation, enabling the modulation of noise intensity. It is generally incorporated into the input variables prior to their submission to the network.

The nature and quantity of noise introduced are critical hyper parameters. Insufficient noise has little effects, however excessive noise can hinder learning. It is necessary to do experimentation to determine the ideal parameters.

Dissonance Injection Timing: Dissonance is generally introduced solely during the preparation phase. The framework must be assessed and utilized for formulation on pristine information devoid of any dissonance interference.

Alternative Techniques for Dissonance Injection

Alternatively, Gaussian noise may be introduced into input variables, activation's, weights, gradients, and outputs.

Injecting noise into activation's: The introduction of noise in the activation layer allows the network to employ the injected noise at any moment during the forward pass through the layer. Introducing noise into an activation layer is beneficial for deep neural networks, since it aids in regularization and mitigates over fitting. The output layer can independently introduce noise through a noisy activation function.

Injecting noise into weights: In the realm of recurrent neural networks, including noise into the weights is an advantageous strategy for regularizing the model. The introduction of noise into the weights typically promotes stability in the function being learned by the neural network. This form of injection is efficient as it directly introduces noise into the weights, rather than either the input or output layers of the neural network.

Injecting noise into gradients: Rather than concentrating on the input domain's structure, injecting noise into the gradients primarily aims to improve the robustness of the optimization process. Similar to gradient descent, the initial level of noise during training may be elevated and typically diminishes over time. Injecting noise into a gradient is one of the most effective techniques for gaining attention in a deep neural network.

Advantages of Incorporating Random Noise

Mitigates over fitting: Introducing dissonance into the preparation process provides unpredictability to the information, resulting in data points being less distinct from one another, hence preventing the network from excessively conforming to each individual information point. This inhibits the communication system from excessively conforming to the preparation information, hence alleviating over fitting.

Minimized generalization error: The existence of noise inhibits the network's tendency to memorize specific training samples, promoting the acquisition of generalize characteristics from the data, hence resulting in minimized generalization error.

The use of noise during neural network training can markedly enhance the model's generalization ability. Moreover, noise injection during neural network training exerts a regularization impact that may enhance the framework characteristic.

Functions as information Augmentation: Noise injection implements a method that facilitates the addition of random dissonance to the stimulation variables during the preparation process. Its ability to uniquely modify input variables upon exposure to the model aids in preventing over fitting.

Execution: Educating Neural Network with Disturbance

The Neural Network model is trained on the MNIST dataset with noise injection for regularization, commencing with an input layer shaped 784, which corresponds to the flattened dimensions of MNIST images. Gaussian noise with a standard deviation of 0.1 is incorporated into the input data during training.

Training is conducted on the training dataset with noisy inputs.

Validation is performed on the testing dataset to assess model performance.

```
import tensor flow as tf
Import Input, Dense, and GaussianNoise from tensorflow.keras.layers.
Import Model from tensorflow.keras.models
Import Adam from tensorflow.keras.optimizers.
Import SparseCategoricalCrossentropy from tensorflow.keras.losses

define construct_model(input_dimensions, class_count):
inputs = Input(shape=input_shape)
noisy_inputs = GaussianNoise(0.1)(inputs)
x = Dense(128, activation='relu')(noisy_inputs)
x = Dense(64, activation='relu')(x)
outputs = Dense(num_classes, activation='softmax')(x)
model = Model(inputs=inputs, outputs=outputs)
return model
input_shape = (784,)
number_of_classes=10
model = construct_model(input_shape, num_classes)
model.compile(optimizer=Adam(),                loss=SparseCategoricalCrossentropy(),
metrics=['accuracy'])
Load the dataset.
tf. keras.datasets.mnist.load_data() returns (x_train, y_train) and (x_test, y_test).
Data Pre processing
x_train = x_train.reshape(-1, 784). astype('float32') / 255.0
x_test = x_test.reshape(-1, 784).astype('float32') / 255.0
history = model.fit(x_train, y_train, batch_size=32, epochs=10,
validation_data=(x_test, y_test))
```

Result:

Epoch 1 of 10

1875/1875 [==============================] - 13s 6ms/step - loss: 0.2555 - accuracy: 0.9247 - val_loss: 0.1313 - val_accuracy: 0.9601

Epoch 2 of 10

1875/1875 [==============================] - 8s 5ms/step - loss: 0.1173 - accuracy: 0.9643 - val_loss: 0.0953 - val_accuracy: 0.9702

Epoch 3 of 10

1875/1875 [==============================] - 10s 5ms/step - loss: 0.0847 - accuracy: 0.9740 - val_loss: 0.0919 - val_accuracy: 0.9728

Epoch 4 of 10

1875/1875 [==============================] - 9s 5ms/step - loss: 0.0688 - accuracy: 0.9780 - val_loss: 0.0803 - val_accuracy: 0.9745

Epoch 5 of 10

1875/1875 [==============================] - 9s 5ms/step - loss: 0.0563 - accuracy: 0.9825 - val_loss: 0.0771 - val_accuracy: 0.9768

Epoch 6 of 10

1875/1875 [==============================] - 9s 5ms/step - loss: 0.0483 - accuracy: 0.9844 - val_loss: 0.0843 - val_accuracy: 0.9746

Epoch 7 of 10

1875/1875 [==============================] - 8s 4ms/step - loss: 0.0423 - accuracy: 0.9859 - val_loss: 0.0796 - val_accuracy: 0.9756

Epoch 8 of 10

1875/1875 [==============================] - 9s 5ms/step - loss: 0.0363 - accuracy: 0.9875 - val_loss: 0.0860 - val_accuracy: 0.9766

Epoch 9 of 10

1875/1875 [==============================] - 9s 5ms/step - loss: 0.0353 - accuracy: 0.9884 - val_loss: 0.0740 - val_accuracy: 0.9790

Epoch 10 of 10

1875/1875 [==============================] - 8s 4ms/step - loss: 0.0302 - accuracy: 0.9900 - val_loss: 0.0715 - val_accuracy: 0.9811

Soft Weight Sharing

Each Soft Tone Weight is a high-density weight encased in soft Lycra fabric, providing exceptional comfort throughout wear. They are adjustable, suitable for either wrist or ankle, and remain secure during physical activity.

Soft Weight Sharing for ANN Compression

The occurrence of deep learning across various utilisation areas has generated the desire to execute and train these models on mobile devices. This, however, contradicts their computational, memory, and energy-intensive characteristics, resulting in an increasing interest in compression.

Algorithms for Growth and Pruning

Growing and pruning algorithms are employed to enhance neural networks and augment the efficacy of machine learning models.

Pruning Decision Trees

Decision tree pruning is an essential method in machine learning employed to enhance decision tree models by mitigating over fitting and augmenting generalization to novel data. This tutorial will examine the significance of decision tree pruning, its various forms, implementation methods, and its role in optimizing machine learning models.

Pruning of Decision Tree

Decision tree pruning is a method employed to avert decision trees from over fitting the preparation dataset. Pruning seeks to streamline the decision tree by eliminating components that lack substantial predictive value, hence enhancing its capacity to generalize to novel data.

Decision Tree Pruning eliminates superfluous nodes from the over fitted decision tree, thereby reducing its size and enhancing the speed, accuracy, and efficacy of forecasts.

Categories of Decision Tree Pruning

There are two primary forms of decision tree pruning:

1. PrePruning and 2. PostPruning.

PrePruning (Early Termination)

The growth of the decision tree can on occasion be halted prior to excessive complexity, a process known as pre-pruning. Preventing over fitting of the training data is crucial, as it leads to sub optimal performance when encountering new data.

Common Pre Pruning treatments encompass

1. **Maximum Depth:** It restricts the furthest level of depth in a decision tree.
2. **Minimum Samples per Leaf:** Establish a minimum criterion for the quantity of samples in each leaf node.
3. **Minimum Samples per Split:** Indicate the minimum quantity of samples required to partition a node.
4. **Maximum Features:** Limit the number of features evaluated for partitioning.

Post-Pruning (Node Reduction)

Post-pruning entails the removal of branches or nodes from a fully grown tree to enhance the model's generalization capability.

Common post-pruning treatments encompass:

Cost-Complexity Pruning (CCP): This technique evaluates each subtree by assigning a cost depending on its accuracy and complexity, then selecting the subtree with the minimal cost.

Reduced Error Pruning: Eliminates branches that do not substantially impact overall accuracy.

Minimum Impurity Decrease: Prunes nodes if the reducing in impurity (Gini impurity or Entropy) falls below a specified threshold.

Region Expansion

Region expanding is a fundamental approach for picture segmentation based on regions. This method is categorized as a pixel-based picture segmentation technique due to its reliance on the assortment of initial inspiration points.

This partition method analyzes adjacent constituent of initial inspiration points to ascertain if the neighboring pixels should be incorporated into the area. The procedure is repeated similarly to typical data clustering methods. A comprehensive overview of the region growth algorithm is presented below.

Region Eccentric segmentation

The primary objective of partition is to divide an representation into distinct areas. Certain partition techniques, like as thres hold, accomplish this objective by identifying borders between sections based on discontinuity in grayscale or colour attributes. Domain based partitioning is a method for immediately identifying the domain.

The fundamental formulation is

$(a) \bigcup_{i=1}^{n} R_i = R.$

(b) R_i is a connected region, $\mathrm{i} = 1,\ 2,\ \ldots, \mathrm{n}$

(c) $R_i \bigcap R_j = \varnothing, i \neq j$

(d) $P(R_i) = \mathrm{TRUE}$ for $i = 1, 2, \ldots, n.$

(e) $P(R_i \bigcup R_j) = \mathrm{FALSE}$ for any adjacent region R_i and R_j.

$P(R_i)$ is a logical predicate defined over the points in set R_i and $\varnothing$ is the null set.

Region Growing Algorithm

A fundamental domain growing algorithm predicated on Eight connectivity can be encapsulated as predate:

1. Identify all affiliated elements in the seed array S(a, b) and erode each component to a single constituent, designating all such constituents with the label 1. All remaining constituents in S are designated as 0.

2. Construct a function (f) so that, for a couple of Cartesian coordinate (a, b), f(a, b) = 1 if the input representation fulfills the specified assert Q at those Cartesian coordinate; otherwise, f(a, b)=0.

3. Let us consider ‘g’ denote an representation created by attaching to each inspiration point in S all the one valued points in ‘f’ that are Eight connected to the respective inspiration point.

4. Assign a distinct area designation (e.g., 1, 2, 3, ...) to each connected component in g. This is the segmented image acquired through region expanding.

Advantages

1. Can accurately delineate regions exhibiting same attributes as defined by us.

2. Capable of supplying original images characterized by distinct edges and superior segmentation outcomes.

3. Fundamental principle: require merely a limited quantity of seed points to encapsulate the desired feature, subsequently expanding the region.

4. Can identify the seed spots and the criteria we wish to establish.

5. Can select many criteria simultaneously.

6. Theoretically highly efficient as it visits each pixel a restricted number of times.

Disadvantages

1. Unless a threshold function has been practical to the representation, a never-ending path of color-related points may connect any two places inside the image.

2. Much stochastic memory approach impedes the algorithms efficiency, necessitating potential adaptation.

Committees and Networks

A committee machine is a form of artificial neural network that employs a divide and conquer strategy, wherein the outputs of several neural networks (experts) are amalgamated into a singular response. The resultant output of the committee machine is intended to surpass that of its individual experts.

Committee Apparatus

A. **Categories**

1. Immovable frameworks

In this category of committee machines, the outputs of several predictors (experts) are amalgamated by a method that excludes the input signal, thus earning the label static. This category encompasses the subsequent methods:

2. **Aggregate averaging**

In ensemble averaging, the outputs of various predictors are linearly aggregated to generate a comprehensive output.

3. Enhancing

In boosting, a weak algorithm is transformed into one that attains arbitrarily high precision.

B. **Dynamic structure**

In this second category of committee machines, the input signal directly facilitates the process that synthesizes the outputs of the different experts into a comprehensive result, thereby earning the name dynamic. There exist two categories of dynamic structures:

C. **Mixture of experts:**

In a mixture of experts, the distinct outputs of the experts are non-linearly integrated through a singular gating network.

Hierarchical mixture of experts:

In a hierarchical mixture of experts, the answers of individual experts are non-linearly integrated through many gating networks organized hierarchically.

D. Ensemble of specialists

The Mixture of Experts (MoE) is a machine learning methodology that use numerous expert networks (learners) to partition a problem space into homogeneous sections. MOE exemplifies a variant of ensemble learning.

Fundamental principles

The MOE consistently comprises the following components, albeit their implementation and integration vary based on the specific problem addressed:

- Experts $f_1, \ldots, f_n$, each taking the same input x, and producing outputs $f_1(x), \ldots, f_n(x)$.
- A weighting function (also known as a gating function) w, which takes input x and produces a vector of outputs $(w(x)_1, \ldots, w(x)_n)$.
- $\theta = (\theta_0, \theta_1, \ldots, \theta_n)$ is the set of parameters. The parameter θ_0 is for the weighting function.
- Given an input x, the mixture of experts produces a single output by combining $f_1(x), \ldots, f_n(x)$ according to the weights $w(x)_1, \ldots, w(x)_n$ in some way.

Both the specialists and the weighting function are trained by minimizing a certain loss function, typically using gradient descent. There is considerable autonomy in selecting the specific configuration of experts, the weighting system, and the loss function.

Meta pi network

The Meta pi network, as documented by Hampshire and Wai bel

$$f(x) = \sum_i w(x)_i f_i(x)$$

Result: The framework is disciplined using gradient descent on the average squared error loss.

$$L := \frac{1}{N} \sum_k \|y_k - f(x_k)\|^2$$

The experts may represent arbitrary functions.

ANN UPDATED NOTES 27-12-24.pdf

ORIGINALITY REPORT

4%	2%	0%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Indian Institute of Information Technology, Allahabad Student Paper	2%
2	en.wikipedia.org Internet Source	1%
3	polynoe.lib.uniwa.gr Internet Source	<1%
4	www.agriculturelore.com Internet Source	<1%
5	Artificial Neural Nets and Genetic Algorithms, 2001. Publication	<1%
6	core.ac.uk Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		