AI CLUB

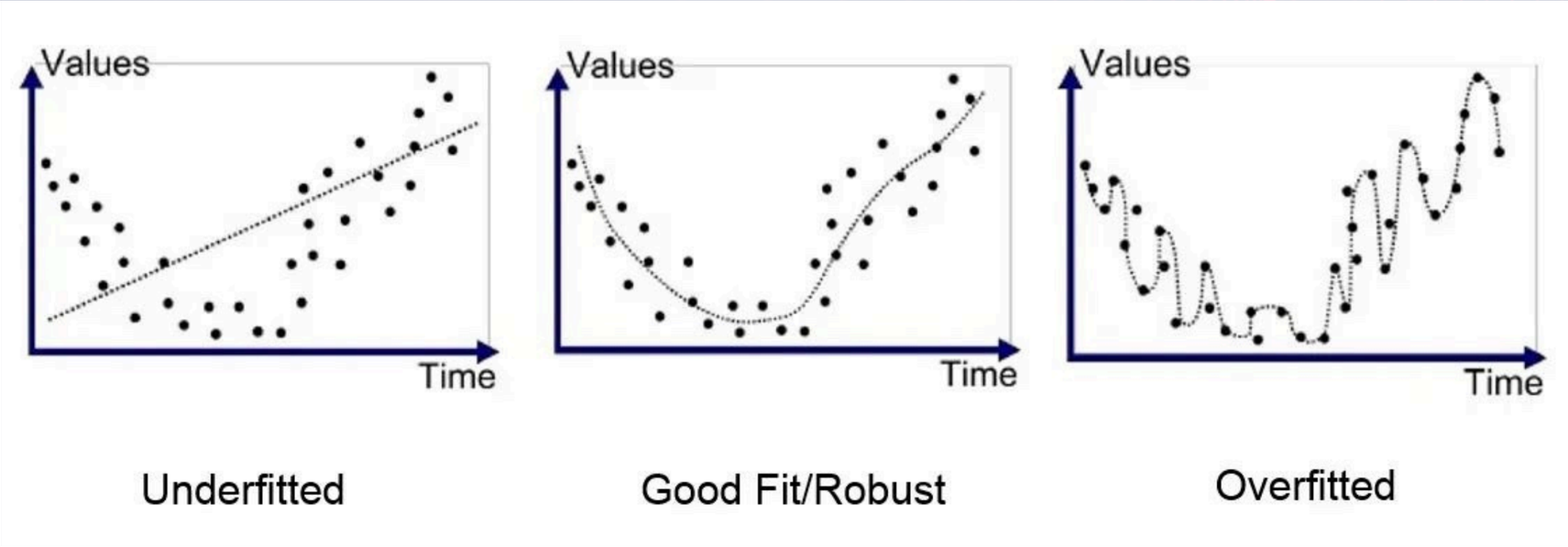
REGULARIZATION

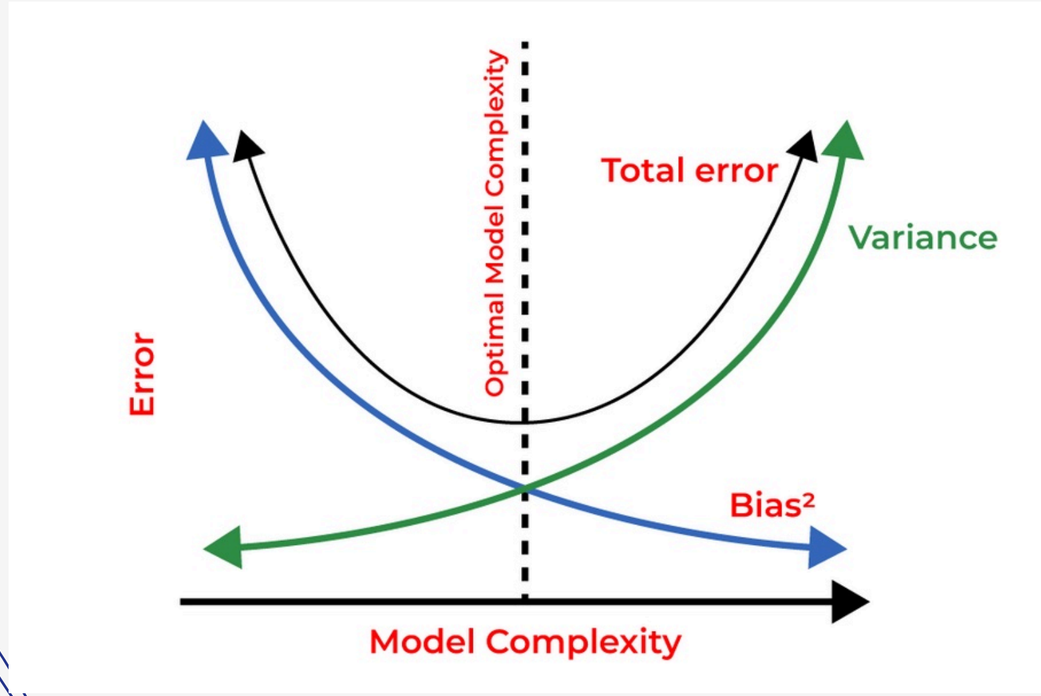
A universal problem in machine learning has been making an algorithm that performs equally well on training data and any new samples or test dataset.

While developing machine learning models you must have encountered a situation in which the training accuracy of the model is high but the validation accuracy or the testing accuracy is low. This is the case which is popularly known as overfitting in the domain of machine learning.

What is overfitting?

In machine learning, overfitting occurs when an algorithm fits too closely or even exactly to its training data, resulting in a model that can’t make accurate predictions or conclusions from any data other than the training data.





Bias-Variance Tradeoff

The bias-variance tradeoff is a fundamental concept in machine learning. It refers to the balance between bias and variance, which affect predictive model performance. Finding the right tradeoff is crucial for creating models that generalize well to new data

WHAT IS REGULARIZATION?

Regularization may be defined as any modification or change in the learning algorithm that helps reduce its error over a test dataset, commonly known as ***generalization*** error but not on the supplied or training dataset.

In simpler terms, Regularization is a method to prevent ***overfitting*** and improving the generalizability of a model.

Types of Regularization

Based on the approach used to overcome overfitting, we can classify the regularization techniques into three categories.

- Modifying the loss function
- Modifying sampling method
- Modifying the training algorithm

MODIFYING THE LOSS FUNCTION

L1 regularisation : Lasso Regression

At the core of L1 regularization, also known as Lasso (Least Absolute Shrinkage and Selection Operator) is a simple yet powerful modification to the loss function. The standard loss function is augmented by a penalty term

$$Loss_{L1} = Loss_{ori} + \lambda \sum |w_i|$$

‘Lambda’ here is a hyperparameter which allows us to increase of decrease the effect of lasso regression by increasing and decreasing its value respectively.

Effect on model

The primary consequence of L1 regularization is its tendency to drive some of the model coefficients to zero, effectively excluding certain features from the model. This phenomenon is particularly beneficial when dealing with high-dimensional data, where some features might be irrelevant or redundant. By pushing coefficients to zero, L1 regularization performs feature selection, simplifying the model and enhancing its interpretability.

It helps in avoiding overfitting by constructing a model that maintains a balance between complexity and performance

L2 regularisation : Ridge Regression

L2 regularization, commonly known as Ridge regression, introduces a different type of penalty to the loss function of a machine learning model compared to L1 regularization. In L2 regularization, the penalty term is the sum of the squares of the model coefficients

$$Loss_{L2} = Loss_{ori} + \lambda \sum |w_i^2|$$

The key aspect of the L2 penalty is the squaring of the coefficients, which tends to reduce their magnitude

Effect on model

The primary effect of L2 regularization is to shrink the coefficients towards zero, but unlike L1 regularization, it does not set them to zero. This shrinkage helps in reducing model complexity and preventing overfitting, particularly in situations where the dataset has highly correlated features. By penalizing the magnitude of the coefficients, L2 regularization ensures that the model does not become overly reliant on any single feature, thereby maintaining a balance in the contribution of all features.

Concept of *Sparcity*

One of the most significant differences between L1 and L2 regularization lies in their impact on model sparsity. Sparsity refers to the number of feature coefficients that are reduced to zero. L1 regularization, with its absolute value penalty term, inherently promotes sparsity. This property is particularly useful in high-dimensional datasets where feature selection is crucial. By driving certain coefficients to zero, L1 regularization simplifies the model and enhances interpretability by identifying the most relevant features.

In contrast, L2 regularization does not inherently lead to sparsity. Due to its squared penalty term, L2 regularization shrinks the coefficients towards zero but typically does not set them to zero. This results in a model where all features are retained, albeit with reduced influence. L2 regularization is more about controlling model complexity and preventing overfitting through a balanced contribution of all features, rather than feature selection.

QR code for Q1

MODIFYING SAMPLING METHOD

Data Augmentation

Data augmentation involves increasing the size of the available data set by augmenting them with more input created by random cropping, dilating, rotating, adding a small amount of noise, etc as shown in the example figure below. The idea is to artificially create more data in the hopes that the augmented dataset will be a better representation of the underlying hidden distribution.

Data Augmentation

Original Image

De-texturized

De-colorized

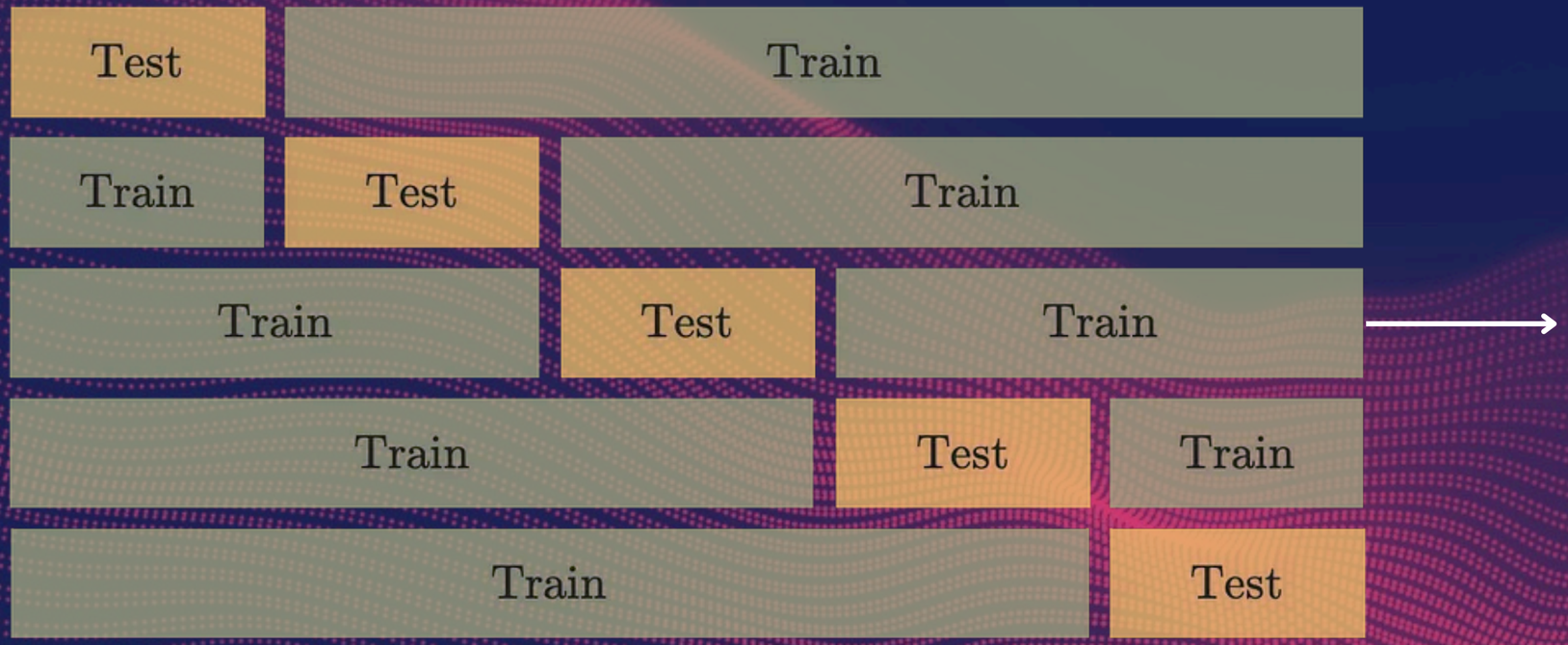
Edge Enhanced

Salient Edge Map

Flip/Rotate

K-Fold cross validation

In K-fold Cross-validation, the available training dataset is divided into k non-overlapping subset and K models are trained. For each model, one of the k subsets is used for validation while the rest of the (k-1) subsets are used for training.



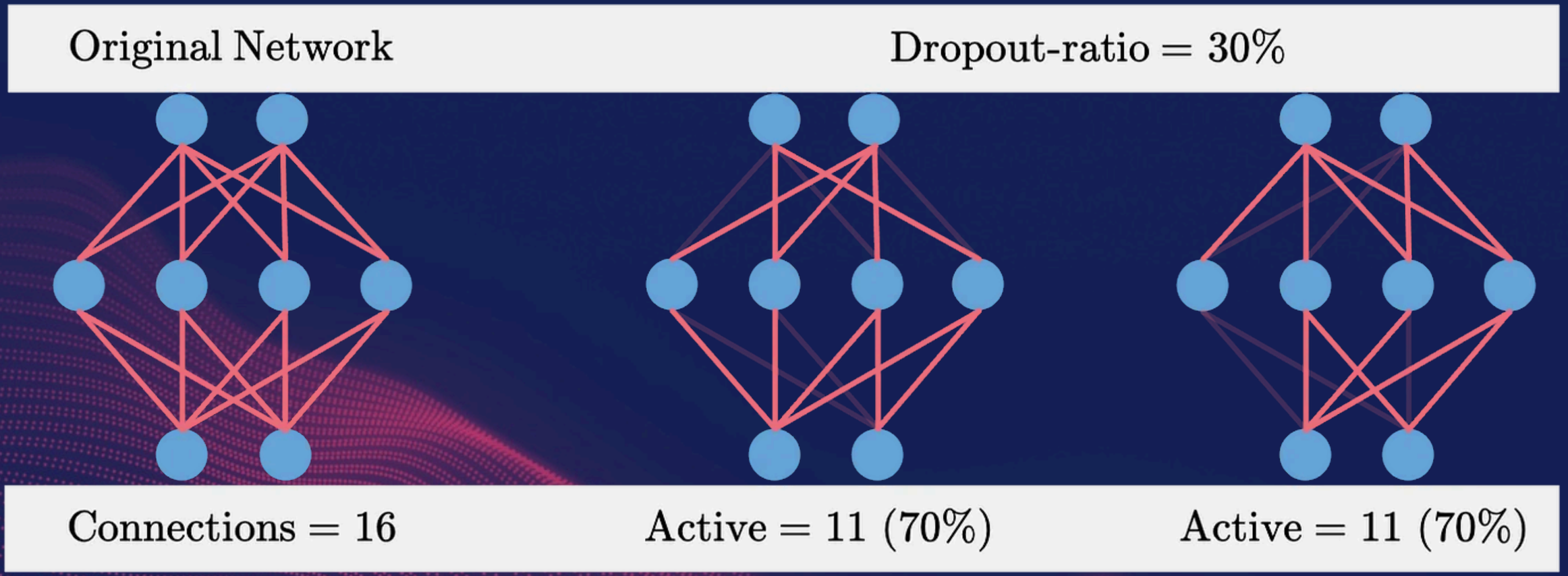
5-fold cross validation

- Shuffle the dataset in order to remove any kind of order
- Split the data into K number of folds. K= 5 or 10 will work for most of the cases.
- Now keep one fold for testing and remaining all the folds for training.
- Train(fit) the model on train set and test(evaluate) it on test set and note down the results for that split
- Now repeat this process for all the folds, every time choosing separate fold as test data
- So for every iteration our model gets trained and tested on different sets of data
- At the end sum up the scores from each split and get the mean score

MODIFYING TRAINING ALGORITHM

Dropout

Dropout is used when the training model is a neural network. A neural network consists of multiple hidden layers.
Each layer in the neural network consists of various nodes.
Nodes from the previous layer are connected to nodes of the subsequent layer. In the dropout method, connections between the nodes of consecutive layers are randomly dropped based on a dropout-ratio(p), which is the percentage of the total connection dropped, and the remaining network is trained in the current iteration.
In the next iteration, another set of random connections are dropped.



Dropout with ratio = 0.3 (30%)

The dropout method ensures that the neural network learns a more robust set of features that perform equally well with random subsets of the node selected. By randomly dropping connections, the network is able to learn a better-generalized mapping from input to output hence reducing the over-fitting.

The dropout ratio needs to be carefully selected and has a significant impact on the learned model. A good value of the dropout ratio is between 0.25 to 0.4

QR code for Q2

CODE IMPLEMENTATION