
Hewlett Packard
Enterprise
HPE AI
COMPREHENSIVE HANDS-ON MACHINE LEARNING AND DEEP LEARNING TRAINING
HPE Official Training Information

HPE

Comprehensive Hands-on Machine Learning and Deep Learning

Topic Covered

01
Spark Essentials

02
Data Preparation

03
ML Task and Type

04
ML Model Training and Evaluation

05
Model optimization

06
Deep Neural Network

07
ANN and CNN

08
Distributed DNN with Determined.AI

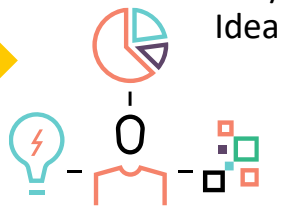
KEY TAKE AWAY FROM THE TRAINING

- Demystify AI
- Enable you to have intelligent conversation with Data Scientist, ML developer and IT
  - Open-up new opportunities for HPE
- Understand the open-source tools used for ML and DL
- Relate these AI tools to HPE's offering and strategy from the edge to the cloud
  - Development and Training at Scale
  - Inference and Deployment at Scale
  - Infrastructure and Accelerators

THE AI JOURNEY

Inspiration

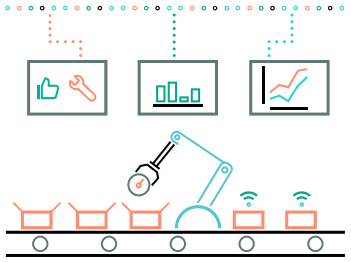
New AI/ML Idea

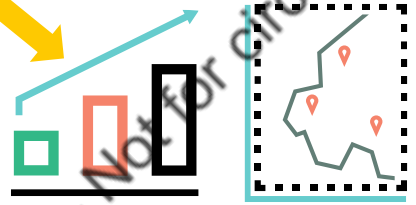


Workshop/Collaboration

R & D Lab PoC

Edge / Production



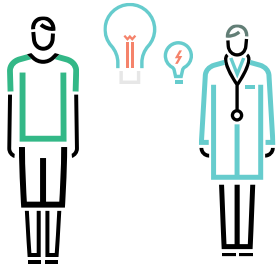


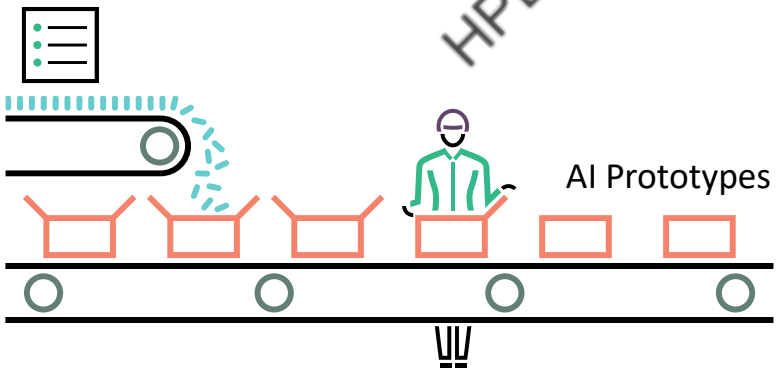
1. Outcomes in Business Value
2. Which usecases or workloads
3. How to get there and start

CXO makes a decision based business priorities

DATA

Data Scientists

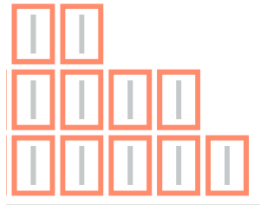




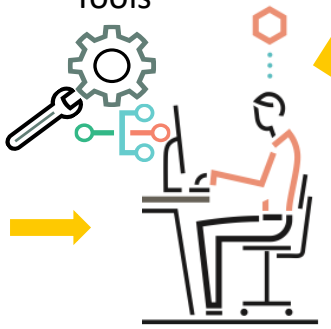
AI Prototypes

AI/ML Developer

Pile of AI Prototypes



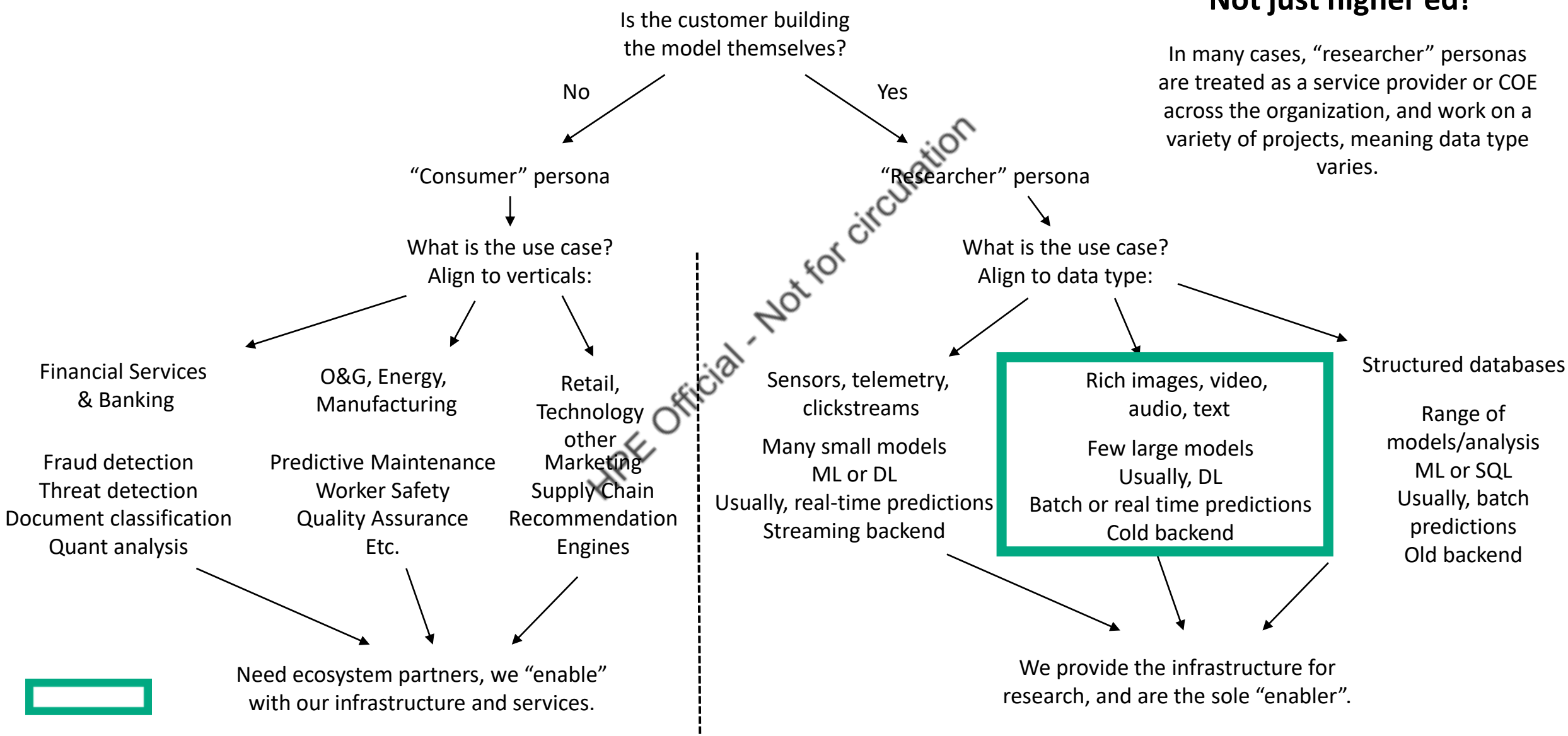
Tools



IT Manager/ Team

Data Center / Production

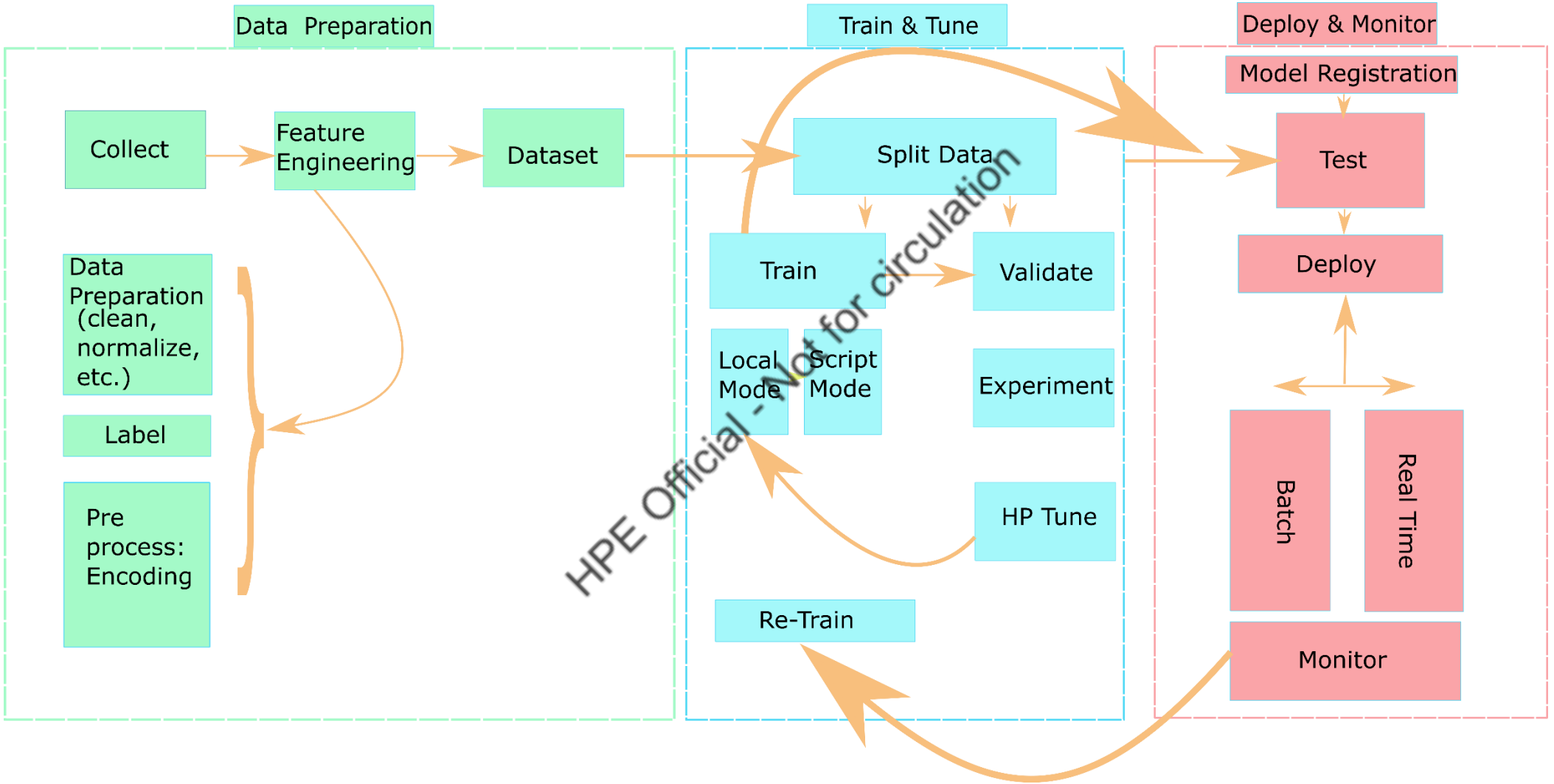
HOW TO QUALIFY AI CUSTOMERS WITH HPE



Not just higher ed!

In many cases, "researcher" personas are treated as a service provider or COE across the organization, and work on a variety of projects, meaning data type varies.

ML LIFE-CYCLE: HIGH LEVEL VIEW



THREE ASPECTS OF AN AI SOLUTION

Data

Preparation, Processing, ETL

Versioning

Labelling, Ground Truth

Storage, Database

Development

Model Training (Distributed, HPO)

Frameworks, Libraries

Collaboration, Workflow Management

SOTA Model Tuning, Evaluation

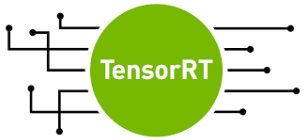
Deployment

Model Inference, Optimization

Testing, Drift/Bias Detection

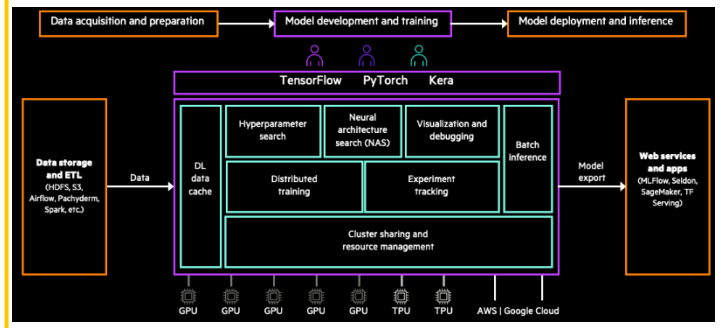
CI/CD, Interface

Visualization, Interchange



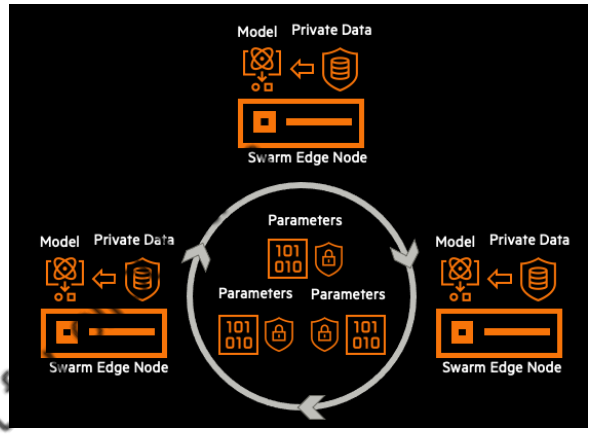
TRAINING WILL HELP POSITION HPE AI / ML SOFTWARE PLATFORMS

HPE Machine Learning Development Environment for at Scale Data Center and Cloud DL Models training



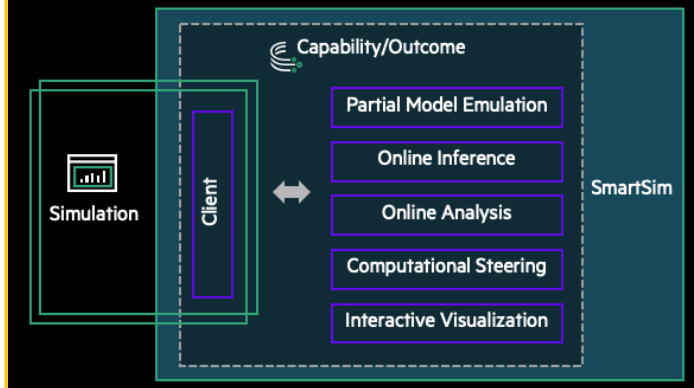
Build high-quality models in less time by simplifying the underlying infrastructure

HPE Swarm Learning for Edge DL and ML Models



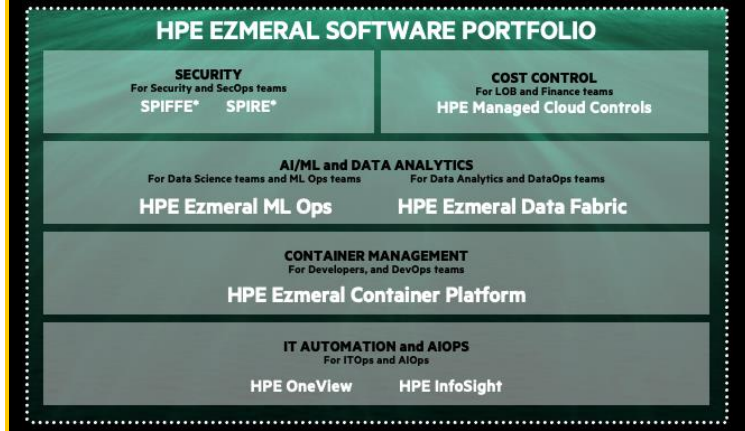
Integration of data from any owner worldwide while maintaining confidentiality without the need for a central coordinator

HPE SmartSIM for Training from Simulations



Accelerate the convergence of AI and data science with scientific simulations

HPE Ezmeral MLOPs & Unified Analytics for ML Models training



MLOPs

Unified Analytics

CloudThat: A Bird's Eye View of the Organization

10+
Years in Business

500K+
Professionals Trained

100+
Corporates Served

200+
Projects Delivered

350+
Cloud Certifications

Leader in Training and Consulting on Cloud, Security, IoT, AI/ML & DevOps

Trained over 500k+ professionals across technologies and geographies

Proven track record of training delivery for all stages of employee lifecycle

Strong team of 200+ certified cloud experts with consulting experience

Robust consulting division brings industry prospective to training delivery

Gold Microsoft Partner
Microsoft

aws partner network
Training Partner

aws partner network
Advanced Consulting Partner

Google Cloud Partner

Trainer Profile

Amit Suresh Sonawane
SME| CloudThat

I am a Microsoft Certified Trainer also certified as a HPE Instructor at **CloudThat**. Very enthusiastic and passionate trainer, empathic observer towards the trending technologies with demonstrated skill in Azure AI. I worked on different projects and various new technologies like Advance SQL, Apache Spark, Determined.ai, Python, seldon.io etc. Currently I am involved in planning, designing and executing various niche technologies trainings for various fortune 500 companies. I delivered several corporate and retail trainings on Azure and trained many people.

Skills: Azure Data Fundamentals, Azure Data Science , AI Fundamentals, Python, MySQL, Spark Programming, C#, JavaScript

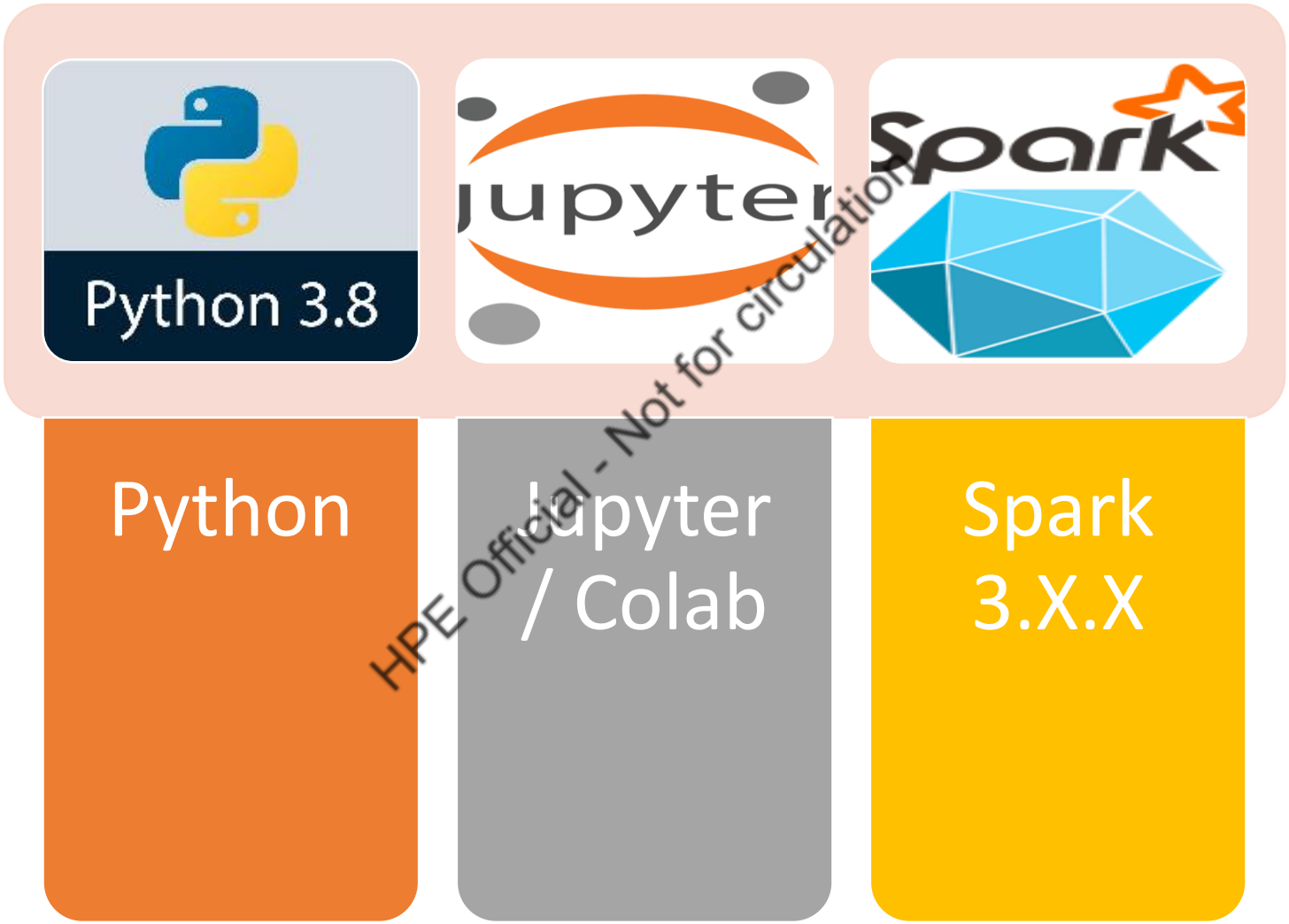
AGENDA

- **Module 1: Spark Overview - PySpark Essentials**
- **Module 2: Machine Learning - Train and Evaluate**
- **Module 3: Neural Network and Deep Learning**
- **Module 4: Distributed Deep Learning**

SCHEDULE (PROPOSED)

Time (min)	Day 1	Day 2	Day 3	Day 4
0 - 15	Intro	Review Day 1	Review Day 2	Review Day 1
15 - 105	Module 1.1	Module 2.1/2	Module 2. 6/7/8	Module 4. 1/2/3
105 - 120	Short Break	Short Break	Short Break	Short Break
120 - 210	Module 1.2	Lab	Lab	Lab
210 - 270	Lunch Break	Lunch Break	Lunch Break	Lunch Break
270 - 360	Module 1.3	Module 2.3/4/5	Module 3. 1/2/3	Module 4. 4/5/6
360 - 375	Coffee Break	Coffee Break	Coffee Break	Coffee Break
375 - 465	Lab Setup	Lab	Lab	Lab
465 - 480	QnA	QnA	QnA	QnA

Lab Environment



Hewlett Packard
Enterprise

MODULE 1: SPARK OVERVIEW

LESSONS:

Lesson 1: Python

- It's a programming language which is user-friendly in terms of coding and it has easy syntax.
- We have PySpark which is a python API which supports Apache Spark functionality
- It is possible because of Py4j library .
- With the help of PySpark we can install Apache Spark and Python on our machine.

Lesson 2: Spark

- open-source cluster computing framework.
- Its main purpose is to handle the real-time generated data.

Lesson 3: Data-Preprocessing

- It's the step that comes after getting the data.
- You try to convert the raw data into meaning data by applying some data pre-processing techniques.
- It helps to visualize data and also to get good prediction results when training a model.

LESSON-1

PYTHON ESSENTIALS

PYTHON ESSENTIALS:

ML Frameworks

- **TensorFlow :** An open-source library machine learning that is excellent for training and inference of deep neural network
- **XGBoost :** An open-source library that provides a scalable, portable and distributed Gradient Boosting (GBM, GBRT and GBDT) library
- **Scikit-learn :** a free machine learning library for python
- **PyTorch :** An open-source library used in applications such as computer vision and natural language processing
- **ONNX :** an open format built to represent AI focused machine learning models

LESSON-2

SPARK ESSENTIALS

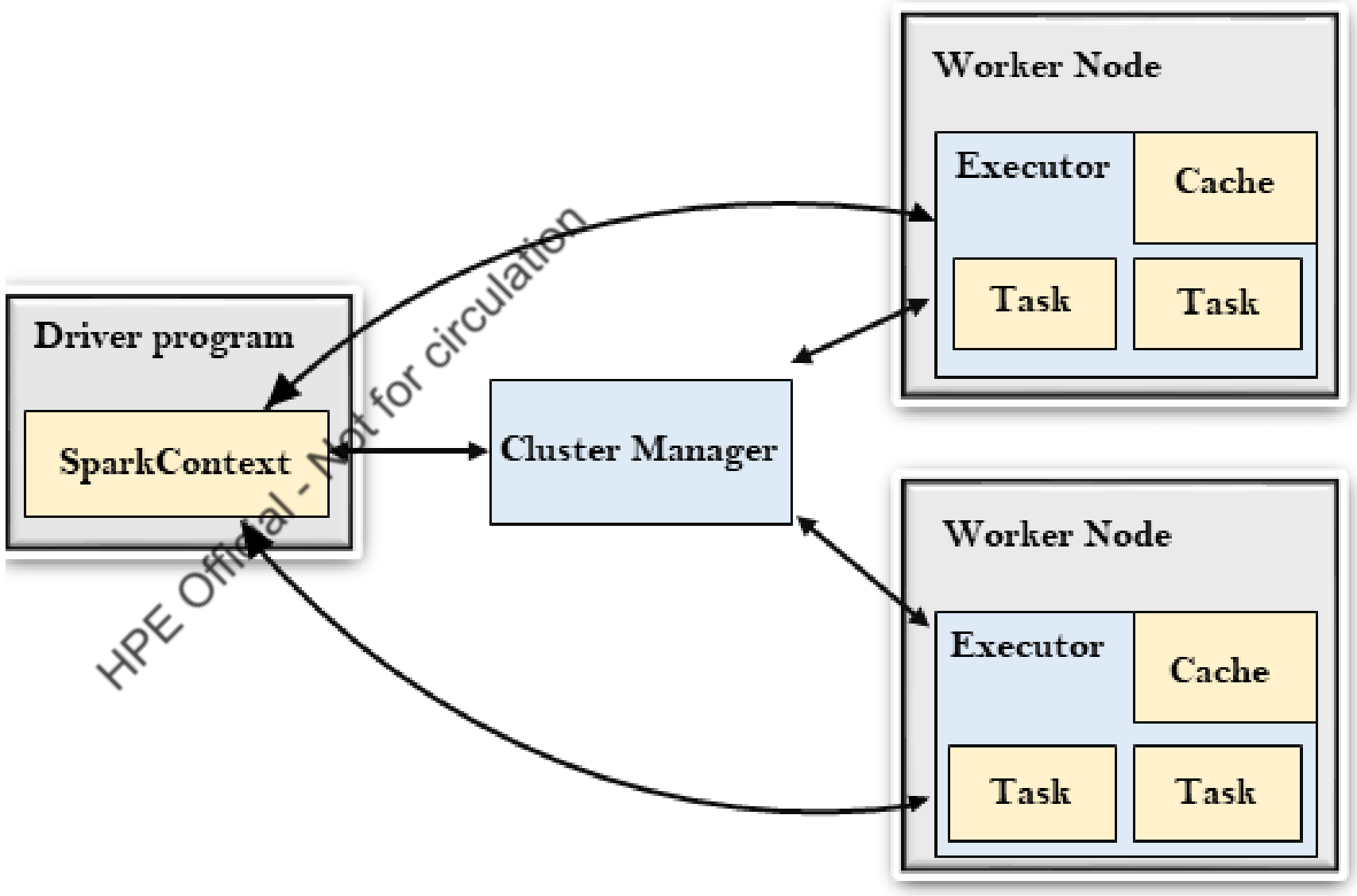
WHY SPARK?

- It performs In-memory computation.
- It's called as Lazy Executor but at the same time once action is called it does parallel processing.
- It supports batch as well as real-time processing whereas Map-Reduce lacked with real-time processing.
- It leverages Apache Hadoop for both storage and processing. It uses HDFS (Hadoop Distributed File system) for storage and it can run Spark applications on YARN as well.
- It has the capability to load data directly from memory, disk and other data storages like Amazon S3, HDFS, Cassandra.
- It’s read once and then till last execution it will not store to storage. This resulting faster than Hadoop Map/Reduce(caching).

BENEFITS OF USING SPARK?

- **Fast** - It provides high performance for both batch and streaming data, using a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine.
- **Easy to Use** - It facilitates to write the application in Java, Scala, Python, R, and SQL. It also provides more than 80 high-level operators.
- **Generality** - It provides a collection of libraries including SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming.
- **Lightweight** - It is a light unified analytics engine which is used for large scale data processing.
- **Deployment(Runs Everywhere)** - It can easily run on Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud.

SPARK ARCHITECTURE



SPARK TERMINOLOGIES

- Driver Program - It creates Spark Context( ) object which acts like an entry point to spark. It is responsible to connect to cluster manager and reserve executors on node in the cluster.
- Cluster Manager - It handles allocating/deallocating resources as it is having the capability to run large number of clusters. There are different types of cluster Managers named Hadoop YARN, Apache Mesos & Standalone Schedular.
- Worker Node - Also called as slave node. It's used to process your data.
- Executor - Controls worker node are working on the task mentioned in a spark job.
- Task – A unit of work that you want to be done which is sent to one of the executors.

SPARK TERMINOLOGIES

- Driver Program - It creates Spark Context( ) object which acts like an entry point to spark. It is responsible to connect to cluster manager and reserve executors on node in the cluster.
- Cluster Manager - It handles allocating/deallocating resources as it is having the capability to run large number of clusters. There are different types of cluster Managers named Hadoop YARN, Apache Mesos & Standalone Schedular.
- Worker Node - Also called as slave node. It's used to process your data.
- Executor - Controls worker node are working on the task mentioned in a spark job.
- Task – A unit of work that you want to be done which is sent to one of the executors.

DEMO

LESSON-3

DATA PREPROCESSING

DATA PREPROCESSING

- **Dataset generation and Import**
- **Data Preparation**
- **Data Cleaning**
- **Data visualization**

DEMO

THANK YOU

Hewlett Packard
Enterprise

MODULE 2:

MACHINE LEARNING LIFE CYCLE

LESSONS:

Lesson-1

Machine Learning Fundamentals

- ML Types
- ML Algorithms

Lesson-2

Machine Learning Life Cycle Steps

- Data ingestion
- Hypothesis Generation
- Testing and Validation
- Model Evaluation
- Model Optimization

Lesson-3

Model Evaluation

- Regression
- Classification
- Clustering

Lesson-4

Model Optimization

- Hyperparameter Tuning

- Classification
- Clustering

Hewlett Packard
Enterprise

Lesson 1:

MACHINE LEARNING FUNDAMENTALS

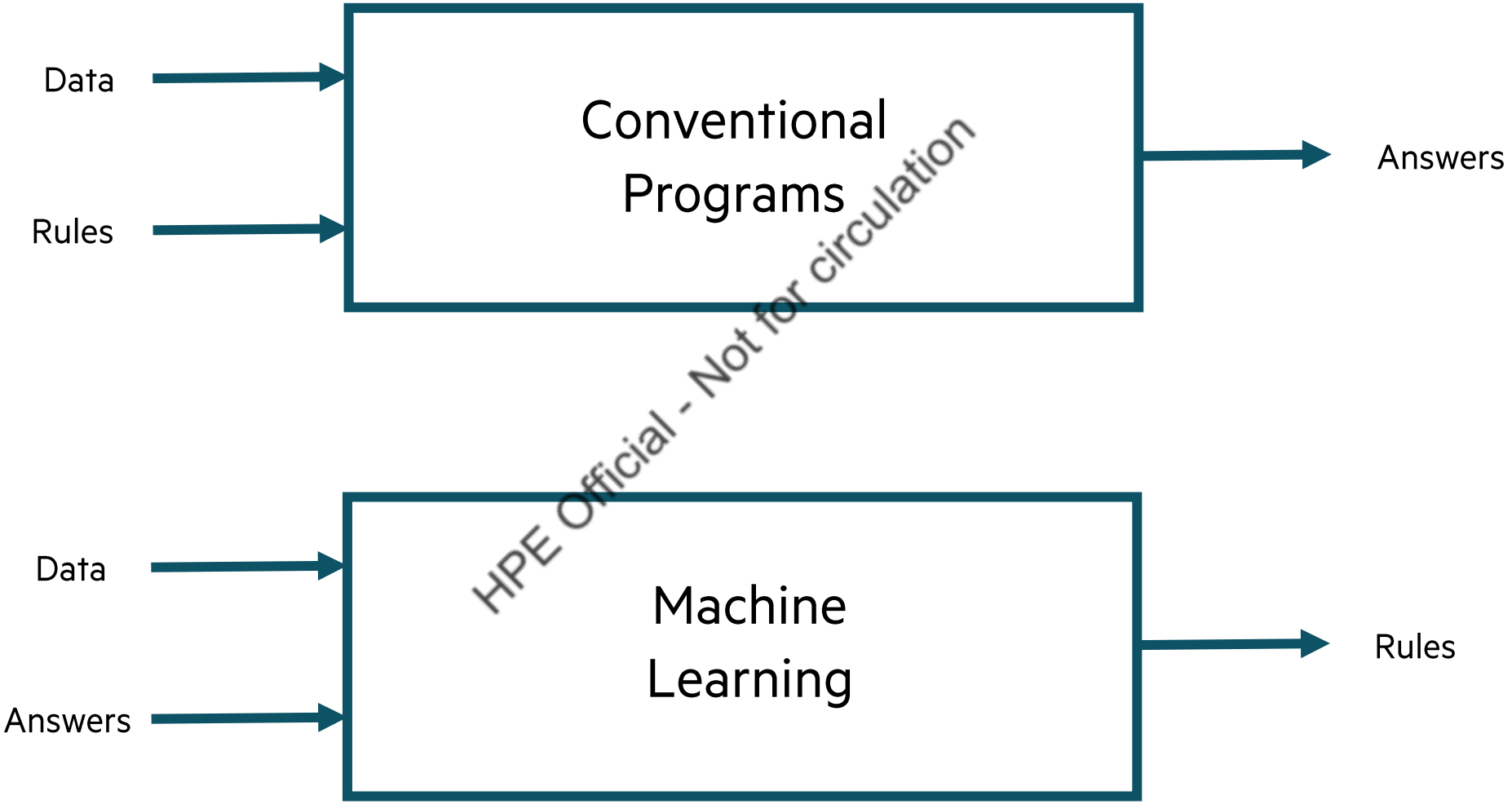
MACHINE LEARNING

- Machine learning technique uses predefined algorithms to analyze provided data and to predict the outcome of unanswered data.

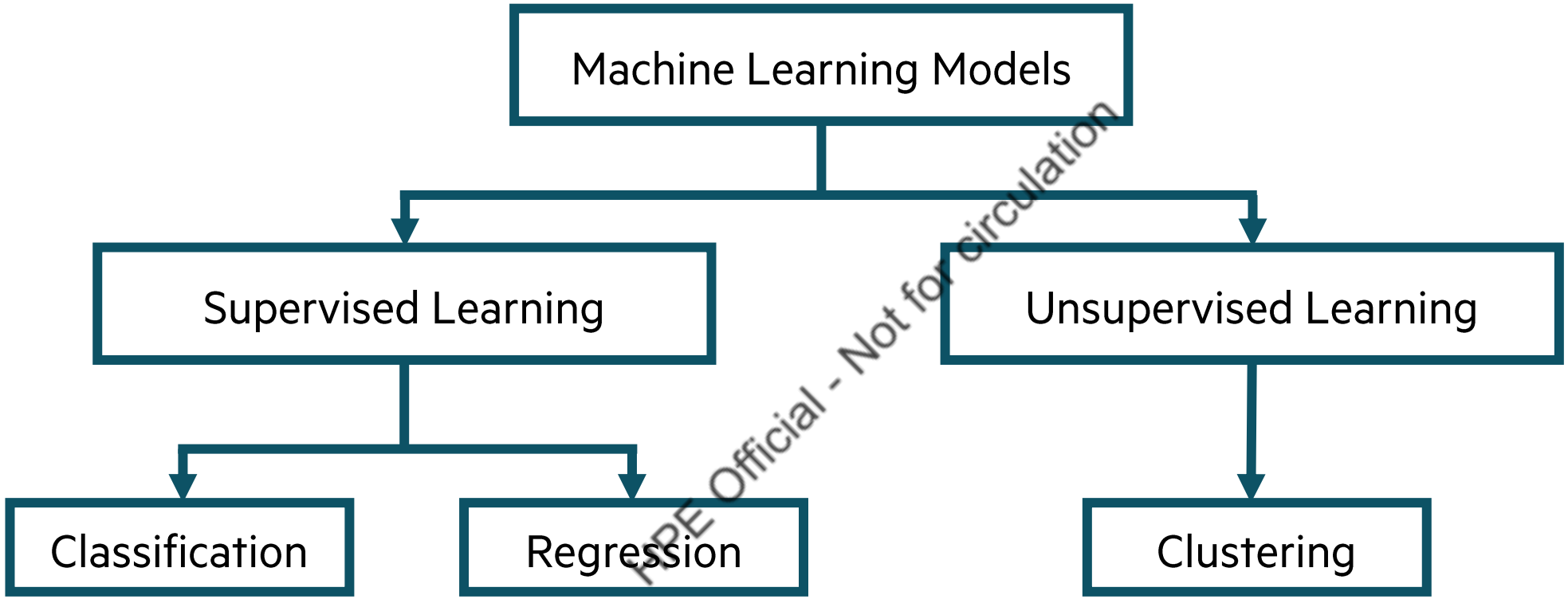
- In conventional programming we need to implement the logic to handle the incoming data based on the predefined rules.

- For ML, the programmer need not worry about rules, instead collect as many data as possible to make the machine to learn the rules.

MACHINE LEARNING



TYPES OF MACHINE LEARNING ALGORITHMS



TYPES OF MACHINE LEARNING ALGORITHMS

- Supervised learning uses a labelled dataset for training
- Unsupervised learning uses dataset without labels
- In classification problems, the model needs to predict a categorical value as a result.
  - For example, to predict incoming mail is ‘spam’ or 'not spam’.
- In regression problems, the model needs predict continuous values as a result.
  - The prediction average life expectancy of a person based on the key health parameters like blood pressure, blood lipid profile, etc. is an example for regression problem.

TYPES OF MACHINE LEARNING ALGORITHMS

- Clustering is used for categorizing the data into groups
- It's an unsupervised learning technique
  - Example grouping the data like customers with similar buying patterns

TYPES OF MACHINE LEARNING ALGORITHMS

- Based on the characteristics of the problem the machine learning models can be classified as

1. Classification
2. Regression
3. Clustering

- Classification models predict a categorical data based on the input features

- Regression models predicts a continuous data output

- Clustering model groups similar datapoints together

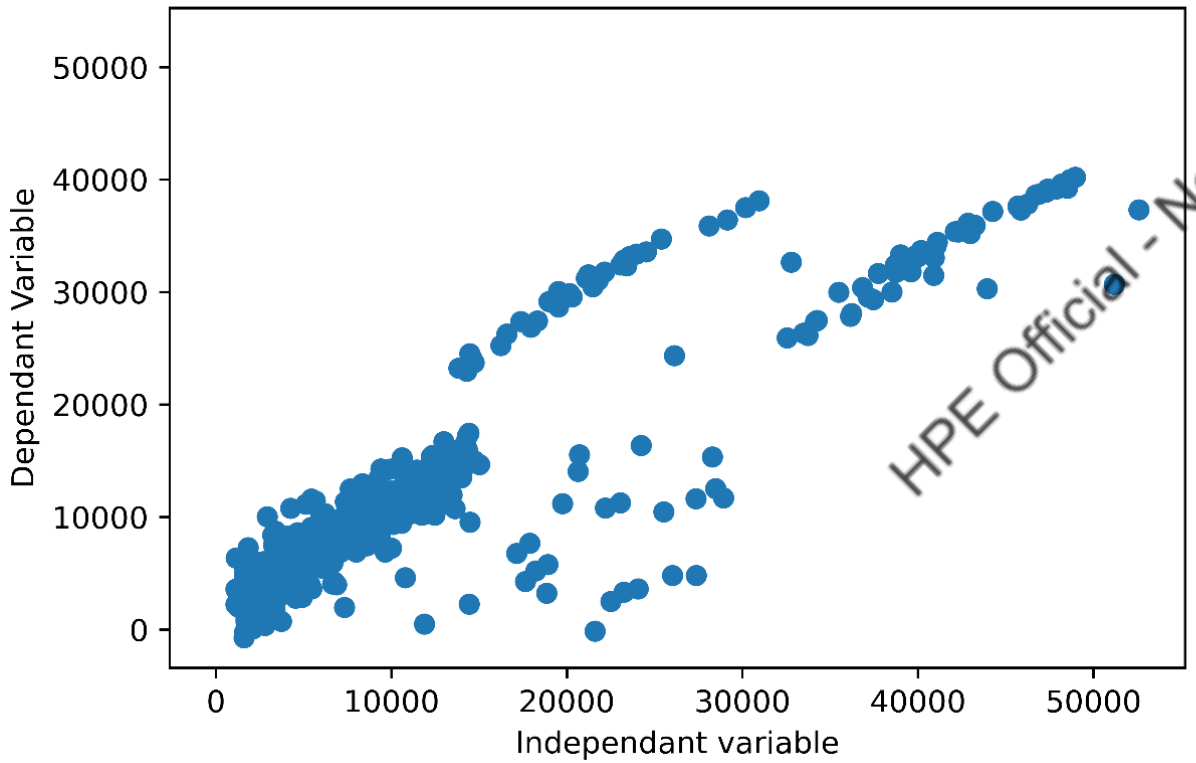
REGRESSION ALGORITHMS

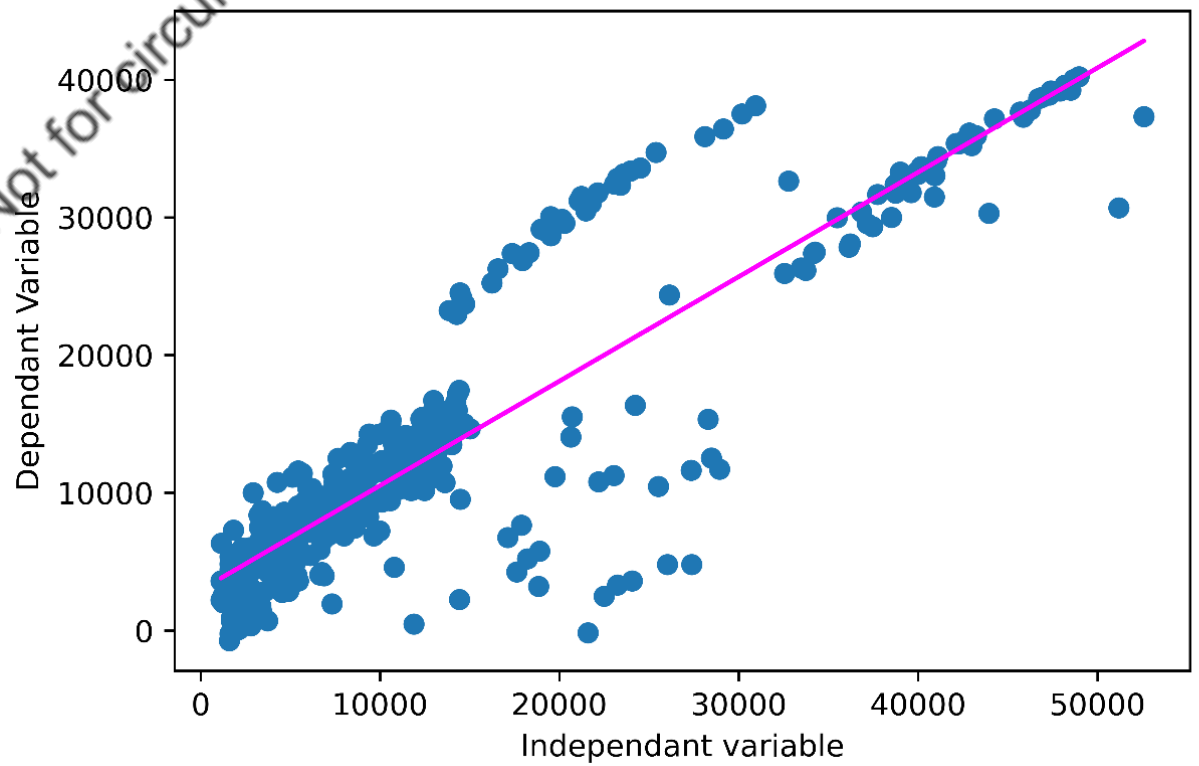
- Prediction of a continuous variable based on features.
- Regression models analyze the relationship between dependent and independent variables.
- Regression algorithms usually determine the possible curve fitting between variables.
- Predict the fuel efficiency based on the vehicle condition indicating a man's age based on his features, predicting the time required for a taxi trip, etc. are examples for regression analysis.

HOW REGRESSION ALGORITHMS WORKS

Linear Regression

- The linear regression algorithm is a process of fitting the data to a line.





LINEAR REGRESSION

- A general equation for linear regression is,

$$\hat{y} = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

- Consider a scenario of insurance premium prediction the equation will be

$$Premium = a_0 + a_1 * \text{ age} + a_2 * \text{weight}$$

- The first step is to choose a metric that tells us how well our model is performing, Mean Square Error (MSE) or Root Mean Square Error (RMSE) can be used

LINEAR REGRESSION

- The model's parameters (a_i) are initialized usually randomly and the error is calculated over the entire training data set.

- In order to minimize this error, these parameters are modified iteratively, Gradient descent is one of the algorithms used for this.

- Algorithm will continue to do this until the parameter value reach the optimal value at the bottom of the curve.

CLASSIFICATION PROBLEMS

- Supervised machine learning algorithms to predict the class of the output variable
- The classification algorithms are binary classification, multi-class classification
- Prediction of images into cat and dog, prediction of patient data into cancerous or not are examples for binary classification.
- Predicting the kind of wheat seed, predicting categories of food items are examples for multi class classification

LOGISTIC REGRESSION

- A Sigmoid curve, or S-shaped curve, is fitting to our observations in Logistic Regression rather than fitting a straight line like in linear regression.

- Logistic Regression models are binary classification models

- A probability of an observation belonging to one of the two categories can be calculated by computing the sigmoid function of X.

- Like linear regression, it is derived from the weighted sum of the input features.

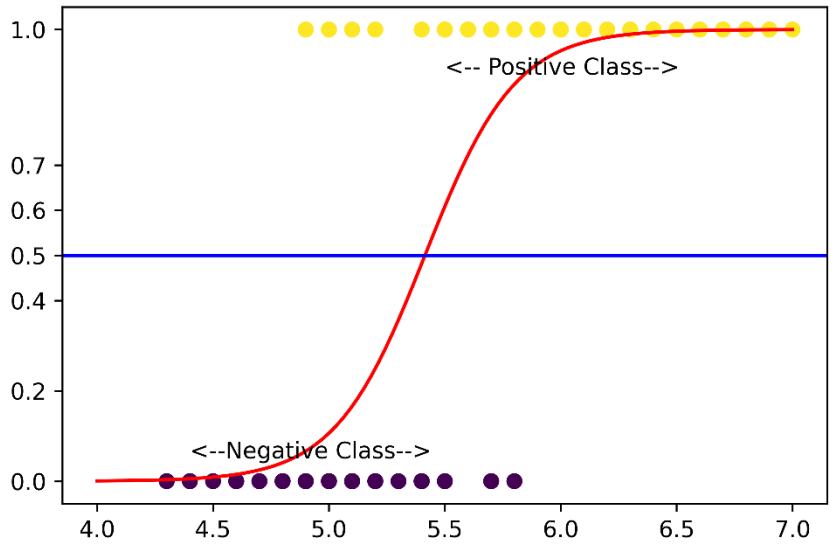
LOGISTIC REGRESSION

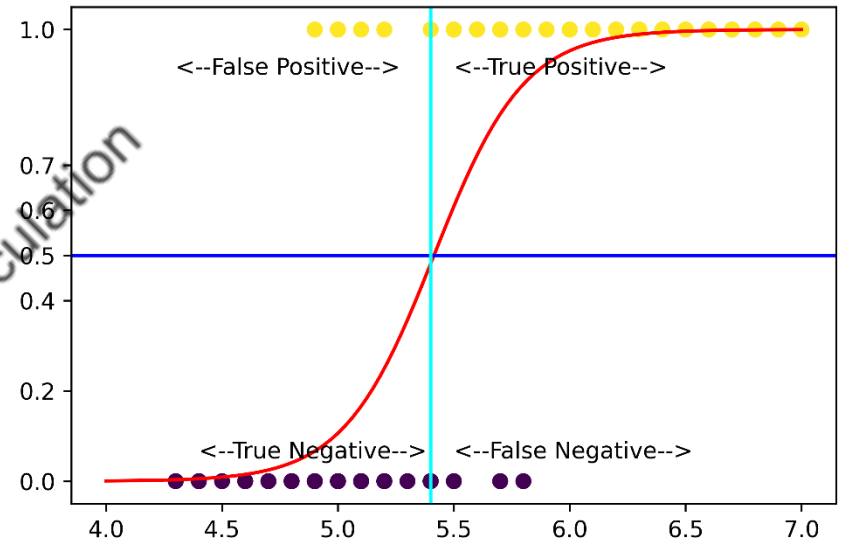
- The sigmoid function

$$sigmoid = \frac{1}{1 + e^{-x}}$$

- Consider the example where we are using logistic regression for predicting patient is diabetic or not.

LOGISTIC REGRESSION





- Figure represents how logistic regression works for binary classification
- Blue line represents the threshold probability

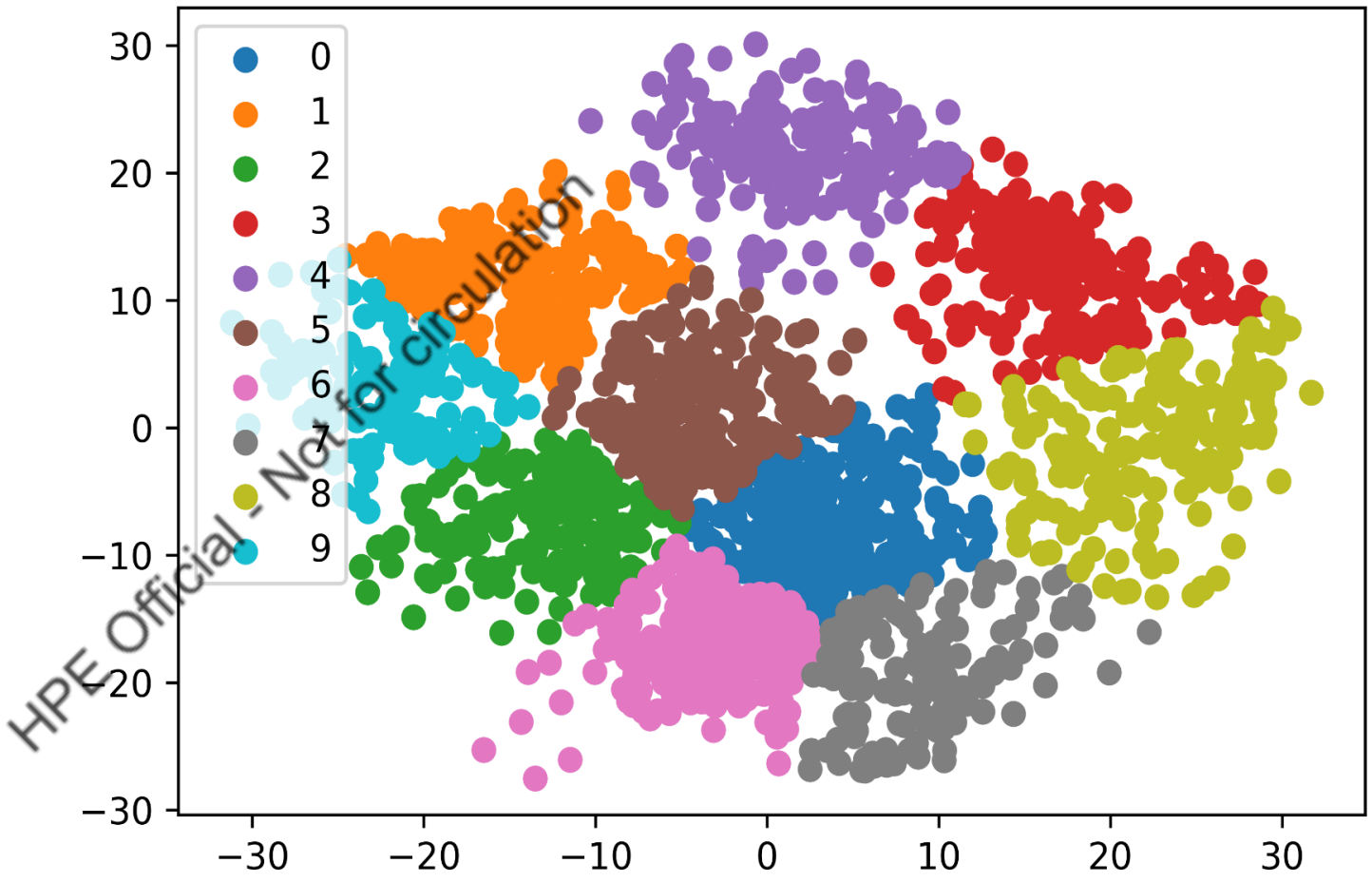
- The classification output can be classified into
  1. True Positive
  2. True Negative
  3. False Positive
  4. False Negative

CLUSTERING PROBLEMS

- An unsupervised machine learning algorithm used to group data points having similar characteristics.
- Hard decision clustering output will specify only one cluster.
- In soft decision clustering, the result provides the likelihood of data points to be in each set.
- The examples for clustering problems are image segmentation, recommendation systems, analysis of social networks, customer trend analysis, etc.

K MEANS CLUSTERING

- Based the Euclidian distance between datapoints from centroid the clusters are chosen



LESSON-2

MACHINE LEARNING LIFE CYCLE STEPS

MACHINE LEARNING LIFE CYCLE

- Understanding the life cycle of the ML model will enable you to know where you stand in the process and manage resources more effectively.
- The steps of a typical ML life cycle are

1. Data Ingestion and preprocessing
2. Hypothesis Generation
3. Model Selection
4. Testing and Validation
5. Model Evaluation
6. Model Optimization

DATA INGESTION

Realtime Data Ingestion : Data is sourced, processed, and loaded as soon as it is created

Batch Data Ingestion : The ingestion process regularly collects the data from the source and transfer it to the target system

Lambda Architecture : This method has the advantage of batch and real-time data ingestion techniques

DATASET SPLITTING

- The split ratio is the measure of the relative quantity of each of these datasets.
- For example, suppose the divided ratio is 0.7:0.1:0.2.
- In that case, 70 percent of the data will be split as training data set 10 percent as validation data set reminding 20% will be the testing data set.

HYPOTHESIS GENERATION

- Generation of educated guesses of various features affecting the problem
- The chances for machine learning projects' failure can be decreased
- Helps to face the problem in a structured way and increases the domain knowledge of the data scientist.

TESTING AND VALIDATION

- The data set for the machine learning need to be split into training data set, validation data set and testing data set.
- The significant chunk of the data will be split as the training data set.
- If we are using the training data set for supervised learning then it will be having labels
- The validation data set infuses new data during the training process to get some helpful information to optimize hyperparameters
- Test data set is usually unlabeled data set to check the generalization capability of our model.

MODEL EVALUATION

- Evaluation of model with the help of performance metrics.
- Evaluation metric for each kind problems will be different.
- Confusion matrix, accuracy, precision, recall, and AUC of ROC curve can be used for evaluation of classification models.
- MSE, RMSE, MAE, and Coefficient of Determination can be used for regression models
- Silhouette Score, Rand Index, Adjusted Rand Index, and Davies-Bouldin Index can work with clustering models.

MODEL EVALUATION (CNT ...)

The performance evaluation can provide following insights about the model

1. How well the model is performing?
2. Will the model performance be adequate for the business requirement?
3. Is the features selected for are relevant for the problem?
4. Do we need to add more features to improve the performance of the model?
5. Can more training improve the performance further?

MODEL OPTIMIZATION

- We do not know exactly what the best model architecture is for a given model.
- So, we would like to be able to experiment with a range of possibilities.
- Hyperparameters cannot be directly learned from the data and untrainable.
- Hyper Parameters are some variables that affect the characteristics of training process.
- Hyper parameter tuning choosing optimum values by looking at possible model architecture candidates, also known as "searching" the hyperparameter space for them.

HYPERPARAMETER TUNING

- A parameter external to the model whose value not estimated from data
- Its value need to set before training begins
- Process
  - Define a model
  - Define the range of possible values for all hyperparameters
  - Define a method for sampling hyperparameter values
  - Define an evaluative criteria to judge the model
  - Define a cross-validation method
- Example
  - Learning Rate, The C and Sigma for SVM, K in KNN, Regularization Constant, No of layers in NN

LESSON-3

MODEL EVALUATION

TOPIC -1

EVALUATION OF REGRESSION MODELS

EVALUATION OF REGRESSION MODELS

- For analysis of regression models, the metrics required will be different from that of the classification model.
- Here the emphasis goes to error in the predicted value
- Error is the difference between predicted value and actual value

RESIDUALS

- Residuals correspond to the distance vertically between a point and its regression line.

- It is the difference between a predicted value and the observed value.

$$Residual = Y - \hat{Y}$$

RESIDUALS

- Consider an example model for predicting the premium of the medical insurance based on the health features of an applicant.

- If the predicted premium value is 450$ and actual value is 500$. Then the residual value will be,

$$Residual = 500\$ - 450\$ = 50\$$$

MSE

- Mean Square Error (MSE) is the squared difference between the actual value and predicted value

$$MSE = \frac{1}{N}\sum\left(Y - \hat{Y}\right)^2$$

MSE

- For the example insurance premium prediction problem, if the summation of squared error is 1000 and 100 predictions are used, then MSE will be

$$MSE = \frac{1000}{100} = 10$$

RMSE

- Root Mean Squared Error (RMSE) is the square root of the mean squared error.

$$RMSE = \sqrt{\frac{1}{N}\sum(Y - \hat{Y})^2}$$

- For instance, if our target and is the insurance premium in dollars, RMSE shows the error in dollars.

- While MSE gives the error in dollars squared, which is more difficult to interpret.

MAE

- Mean Absolute Error (MAE) is defined as the absolute error between the actual and predicted values.

$$MAE = \frac{1}{N}\sum|Y - \hat{Y}|$$

MAE

- If we have 100 predictions and summation of residuals of these 100 predictions is 250$ in our example regression problem.

- Then MAE will be

$$MAE = \frac{250\$}{100} = 2.5\$.$$

R^2 SCORE

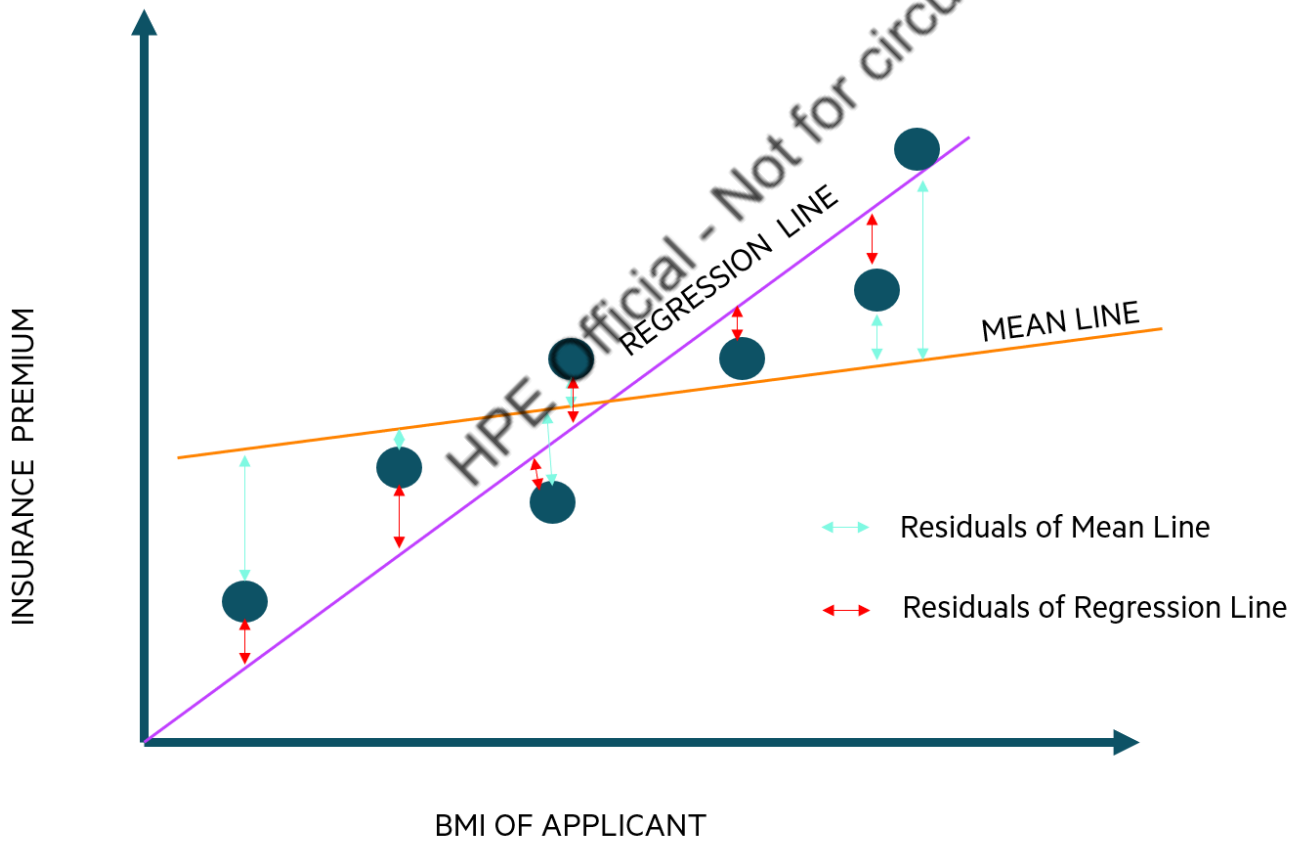
- R – Squared is the measure of how the regression line is better than the mean line.

$$R^2 Score = 1 - \frac{Squared\ Sum\ Error\ of\ Regression\ Line}{Squared\ Sum\ Error\ of\ Mean\ Line}$$

- R^2 Score indicates the square of the correlation (R) between the variables.
- R^2 gives more interpretable values than R.

R^2 SCORE

- The figure shows the relation between two variables, insurance premium and Body Mass Index (BMI), in our example regression problem.
- The blue dots are the data points, orange line indicates the mean line and the purple line indicates the regression line.



R^2 SCORE

- For example, if we choose two variables such that the R=0.7, and R=0.5,
- we can not quantify how much better the first as compared to the second.
- If we choose R^2 ,
- For R=0.7, R^2 =0.5=50%
- For R=0.5, R^2 =0.25=25%
- From R^2 value we can say the first one has double the correlation than second.

TOPIC-2

EVALUATION OF CLASSIFICATION MODELS

EVALUATION OF CLASSIFICATION MODELS

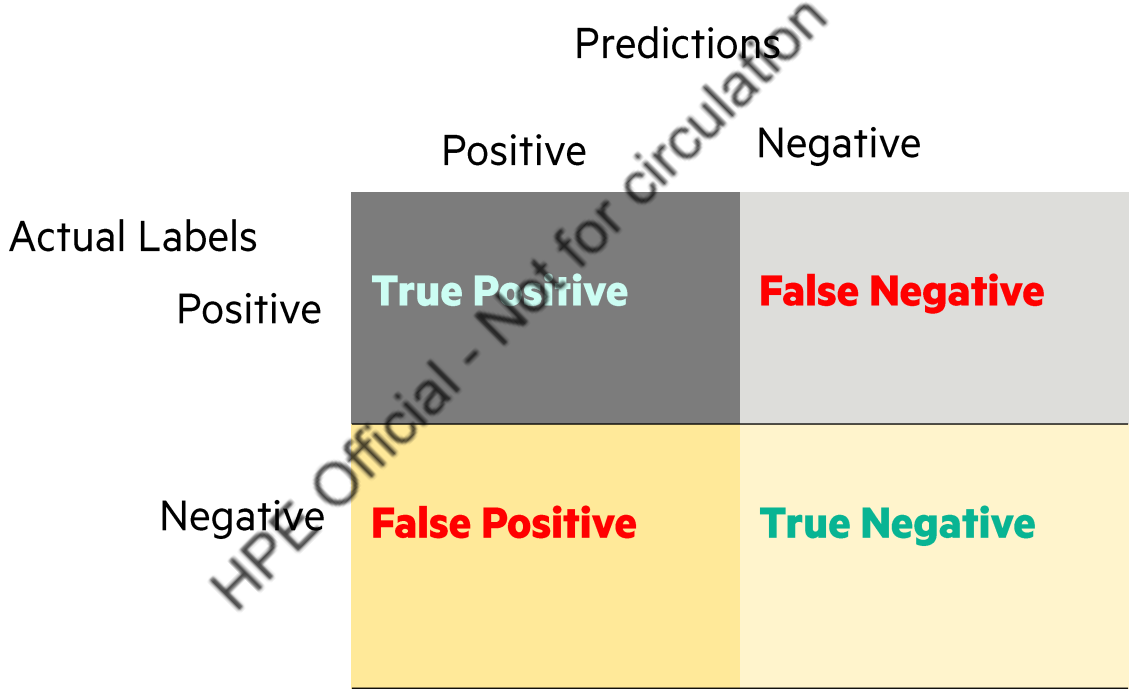
- For the machine learning problem, the model need to be evaluated based on its outputs.

- For classification problems the output of the model comes under,

1. True Positive
2. True Negative
3. False Positive
4. False Negative

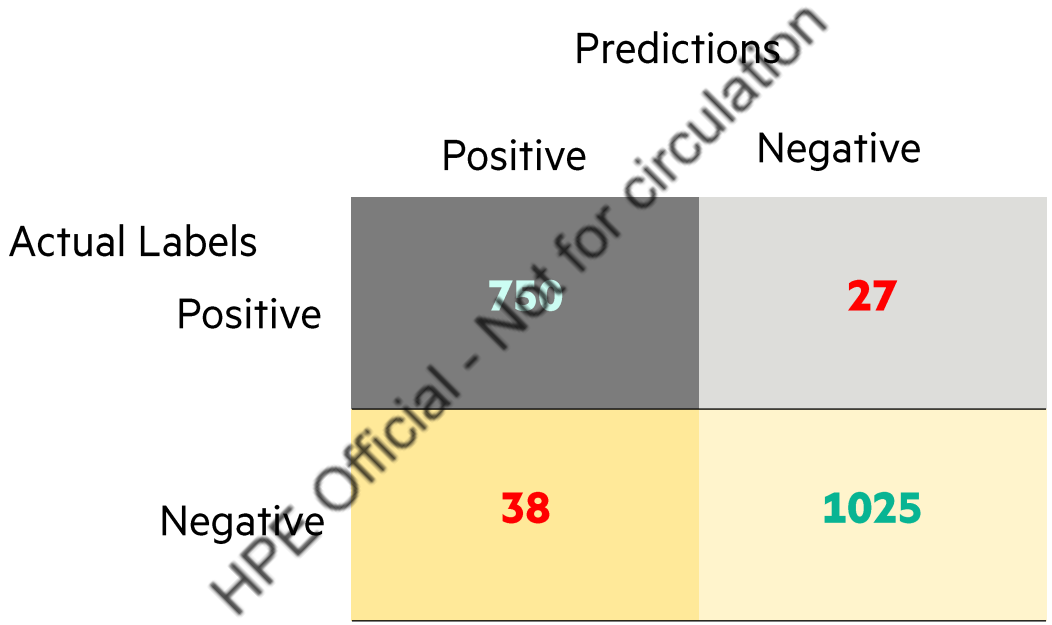
CONFUSION MATRIX

The confusion matrix depicts actual positive, true negative, false positive, false negative values graphically.



CONFUSION MATRIX

- In this example TP value is 750, TN is 1025, FP is 38, and FN is 27.



ACCURACY

- Accuracy is defined as the ratio of accurate prediction.
- It is suitable when classes are well balanced.

$$Accuracy = \frac{Correct\ Prediction}{All\ Predictions}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

ACCURACY

- In the example shown above the accuracy value will be

$$accuracy = \frac{750 + 1025}{750 + 38 + 1025 + 27} = \frac{1775}{1840} = 0.9647 = 96.47\%$$

PRECISION

- The fraction of actually positive predictions among the total positive predictions is known as precision.

$$Precision = \frac{Correct\ Positive\ Prediction}{All\ Positive\ Predictions}$$

$$Precision = \frac{TP}{TP + FP}$$

PRECISION

- For our example the precision value will be

$$Precision = \frac{750}{750+38} = \frac{750}{788} = 0.9517 = 95.17\%$$

RECALL

- The fraction of the positive class prediction, which is truly the positive class

$$Recall = \frac{Correct\ Positive\ Prediction}{All\ Positive\ Labels}$$

$$Recall = \frac{TP}{TP + FN}$$

RECALL

- In the diabetes example

$$Recall = \frac{750}{750 + 27} = 0.9652 = 96.52\%$$

F1-SCORE

- F1 score is the harmonic mean of precision and recall
- F1 score is a number between 0 and 1

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

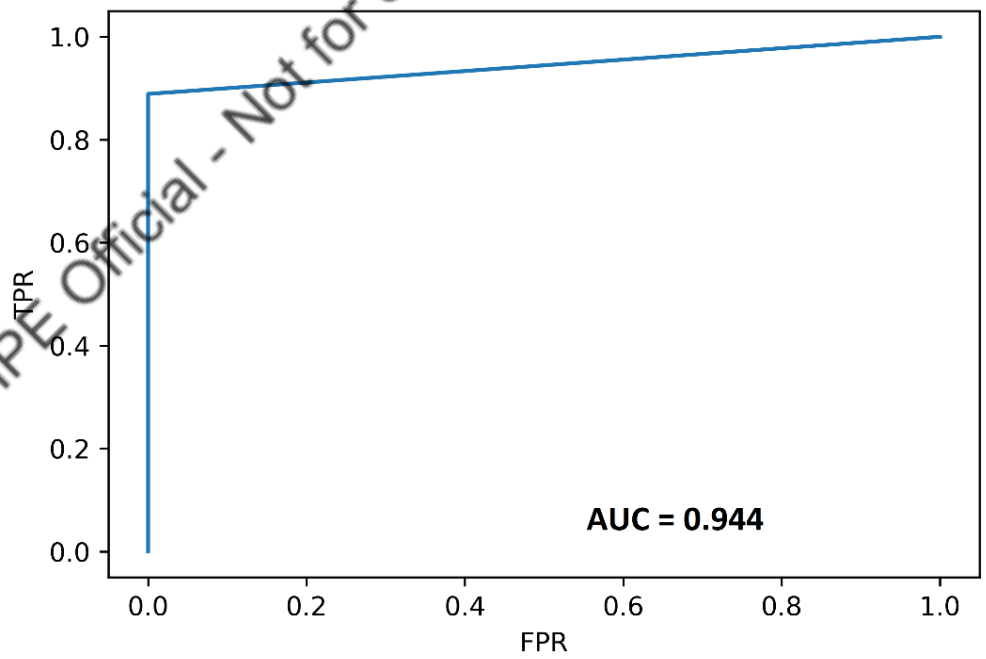
F1-SCORE

- In the diabetes example

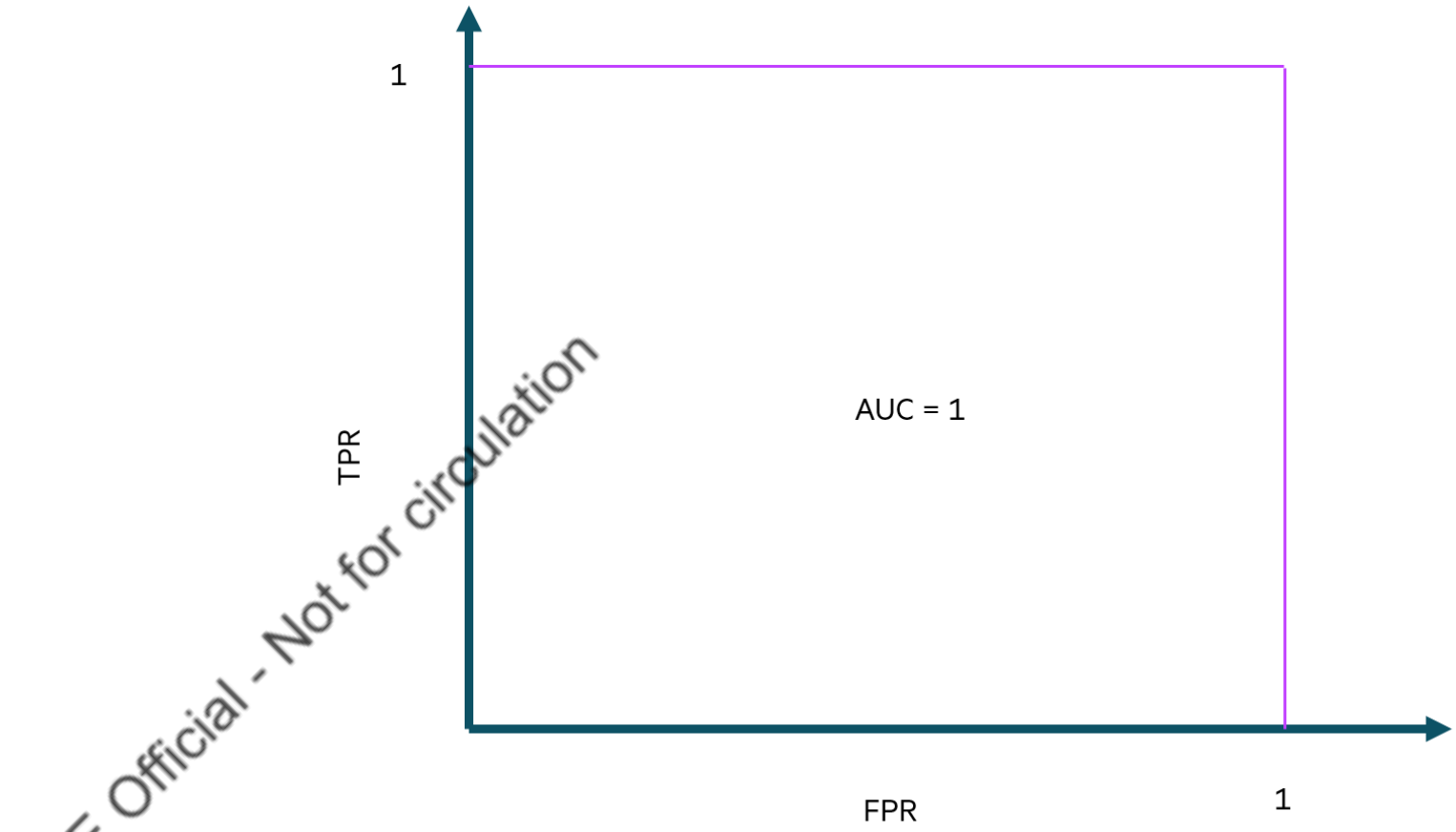
$$F1\ Score = \frac{2 * 0.9517 * 0.9652}{0.9517 + 0.9652} = \frac{1.8372}{1.9169} = 0.9584$$

AREA UNDER ROC CURVE

- It is measure of how well positive classes and negative classes are separated.
- The Receiver Operating Characteristics (ROC) is plotted TPR (Sensitivity) verses FPR (1 - Specificity)

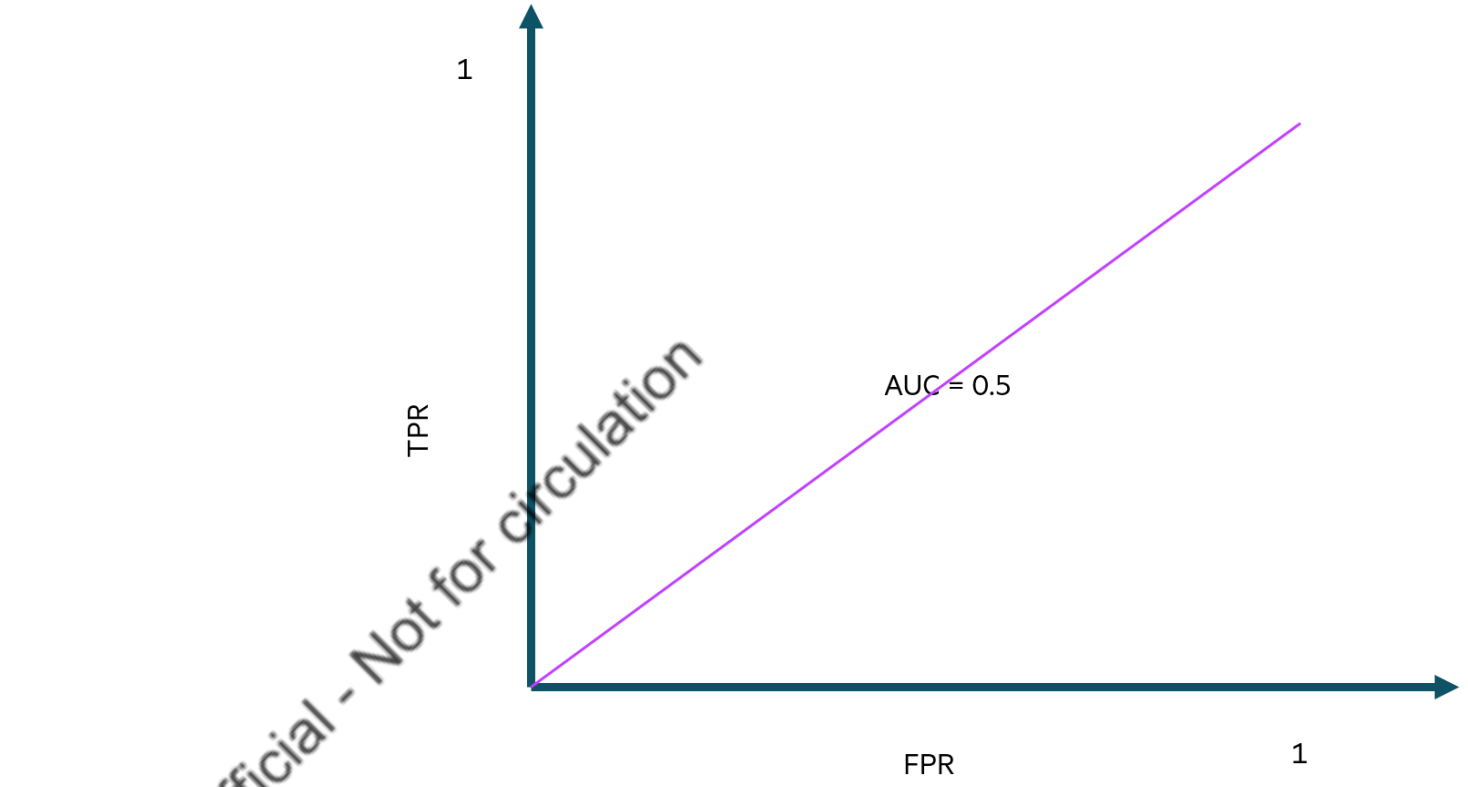


AREA UNDER ROC CURVE



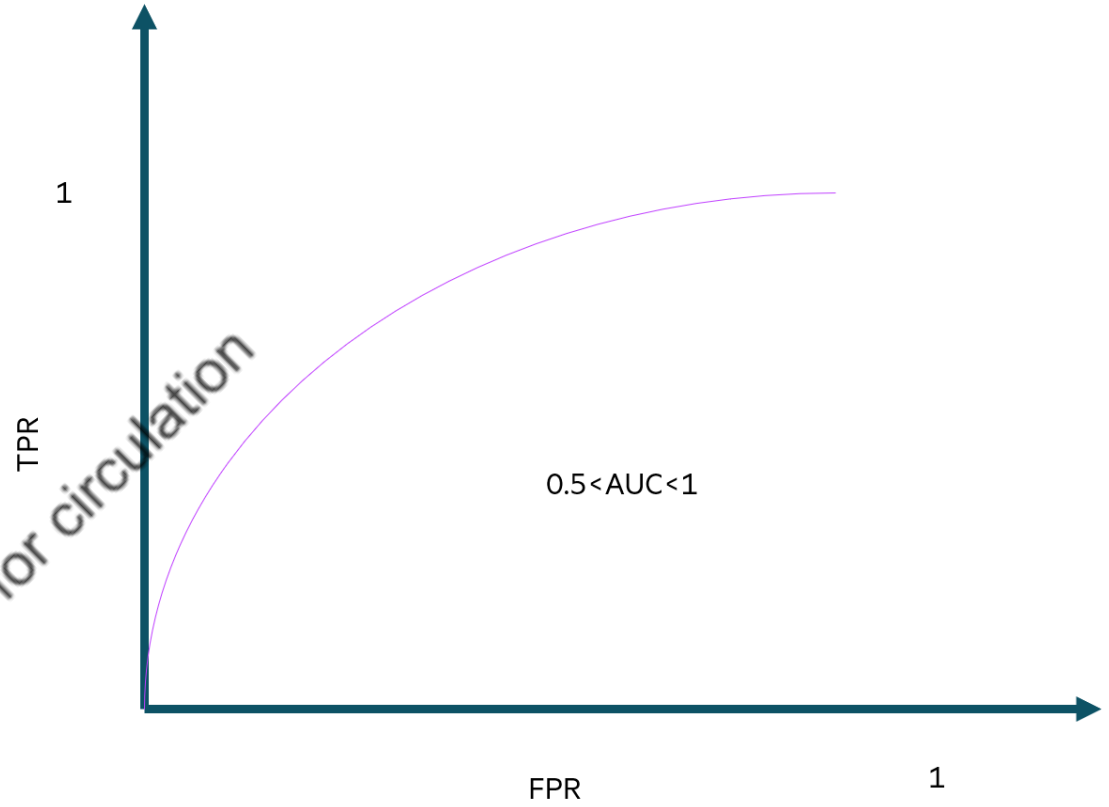
- If AUC = 1, then the classifier is capable of correctly separating all Positive and Negative class points.

AREA UNDER ROC CURVE



- If AUC = 0.5, either the classifier predicts a constant class or a random class for all the data points.

AREA UNDER ROC CURVE



- In the case of 0.5<AUC<1, a high probability exists that the classifier will be able to identify positive and negative class values.

TOPIC - 3

EVALUATION OF CLUSTERING MODELS

EVALUATION OF CLUSTERING MODELS

- In contrast to supervised machine learning, the clustering algorithm presents a slightly different challenge as it does not contain ground truth labels.

SILHOUETTE SCORE

- It is the measure of separateness between the clusters
- It is a value between -1 and +1.
- For mean intra cluster distance (i) and mean nearest cluster distance (n), the silhouette score is,

$$Silhouette\ Coefficient = \frac{(n-i)}{\max(n,i)}$$

RAND INDEX

- This is the measure of similarity for every pair in our model to each pair in optimal or ground truth cluster

- The value is between 0 and 1, for a perfect fit the RI will be 1.

$$Rand\ Index = \frac{Number\ of\ Agreeing\ Pair}{Total\ Number\ of\ Pairs}$$

ADJUSTED RAND INDEX

- It overcomes the unpredictability in the expected value of rand index between two random clusters.

$$Adjusted\ Rand\ Index\ = \frac{RI - Expected\ RI}{\max(RI) - Expected\ RI}$$

LESSON-4

MODEL OPTIMIZATION

TOPICS:

Topic-1

Hyperparameter Tuning

- Need For Hyperparameter Tuning
- Hyperparameter Tuning Process
- Types Of Splitting The Dataset
- Hyperparameter Search Space

Topic-2

Hyperparameter Sampling Techniques

- Grid Sampling
- Random Sampling
- Bayesian Sampling

Topic-3

Early Stopping Techniques

- Bandit Policy
- Median stopping policy
- Truncation selection policy

TOPIC-1

HYPERPARAMETER TUNING

INTRODUCTION – UNDERSTANDING THE TERMINOLOGIES

- **Model Parameters**

- The values of the model parameters are internal to the model.
- The values of the model parameters are determined by the training model.
- Therefore, the values of the model parameters are dependent on the training dataset.
- The model parameters are utilized in making the predictions for the new data during the inferencing

Weights of a neural network - W_0, W_1, … W_n.

Support vectors.

Coefficients of a regression model.

Centroids of a clustering model.

Filter values of a CNN model.

FIG. 1a Examples of model parameters

INTRODUCTION – UNDERSTANDING THE TERMINOLOGIES

- **Model Hyperparameters**

- The model hyperparameters are external to the machine learning model.
- The values of the model hyperparameters must be set using hyperparameter tuning or determined manually.
- The values of the model hyperparameters must be set before starting the training of the machine learning model.
- The model hyperparameters are in-turn used to determine values of the model parameters.

Learning rate.

Number of clusters.

Number of hidden layers in a neural network.

Choice of activation function, penalty function, loss function, etc.

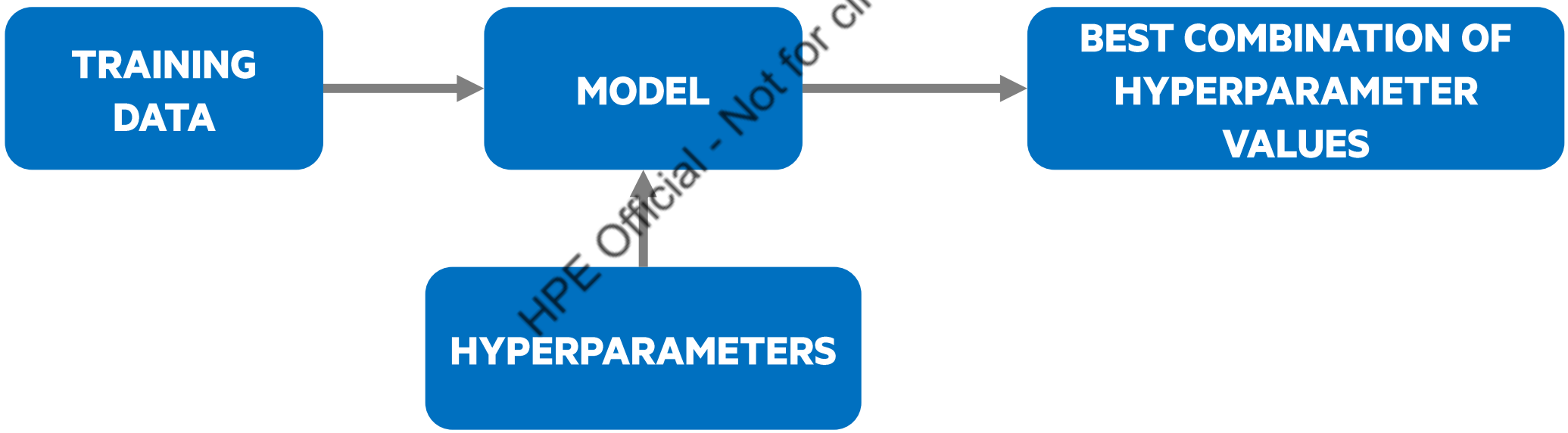
Value of regularization variable.

Choice of train test split.

FIG. 1b Examples of model hyperparameters

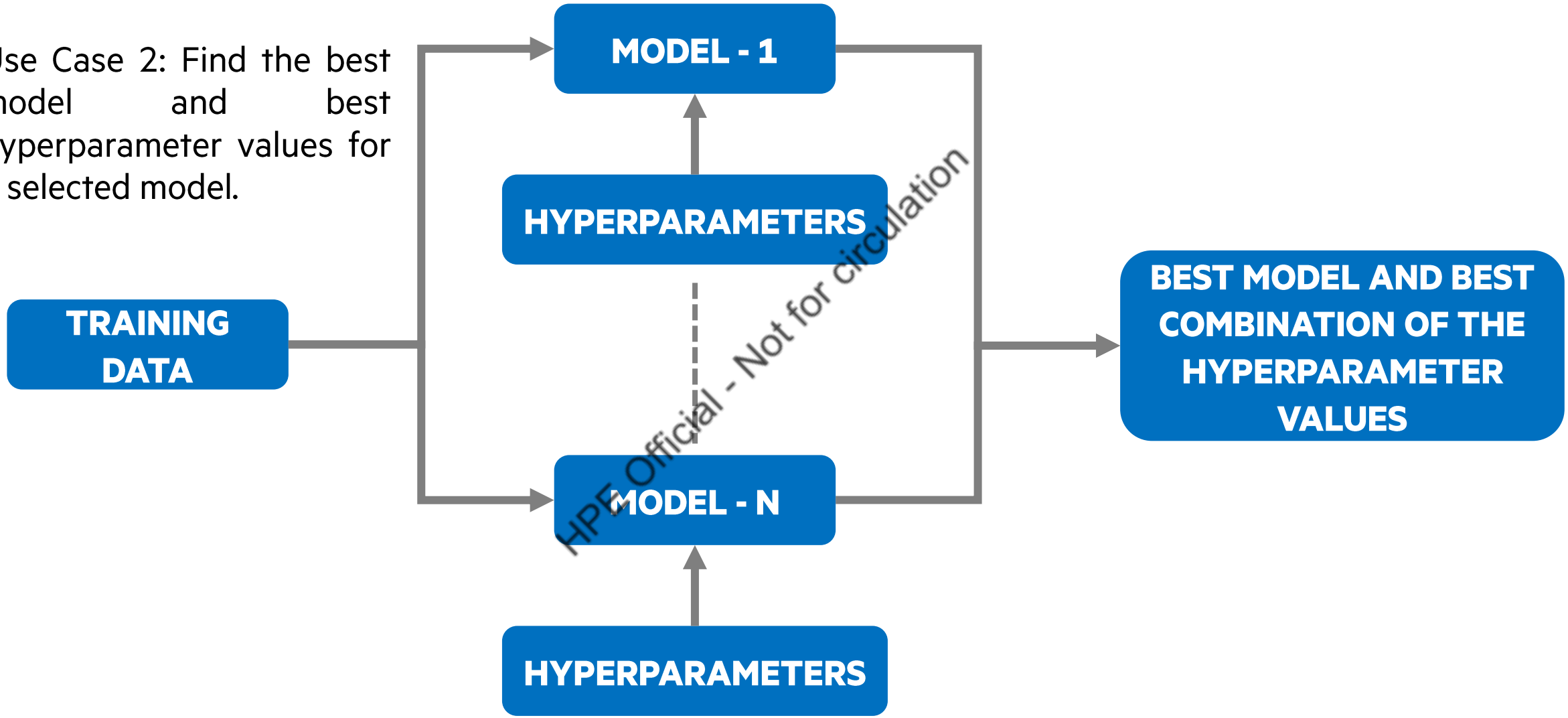
NEED FOR HYPERPARAMETER TUNING

- Use case 1: Find the best hyperparameter values for a selected model.

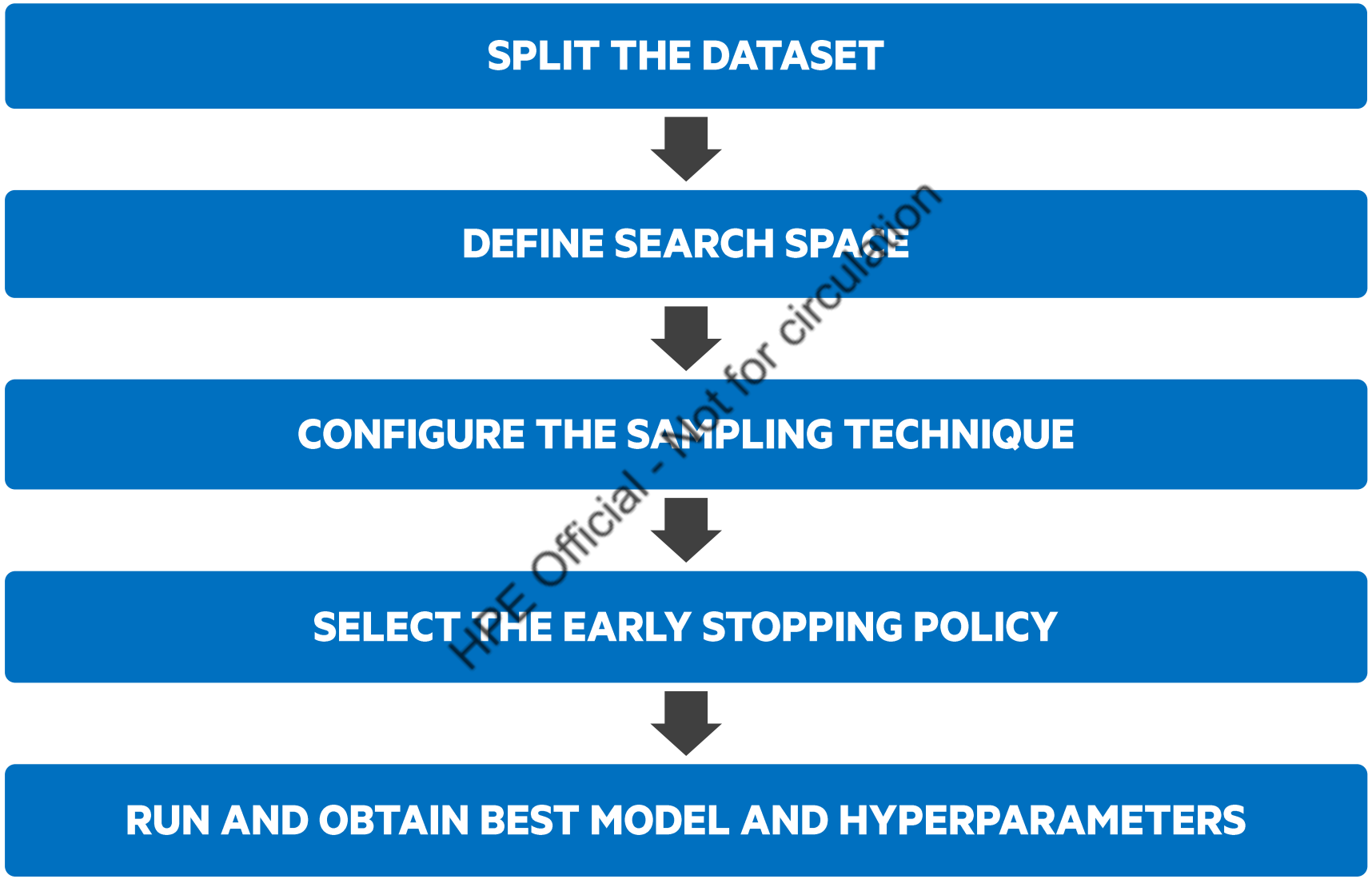


NEED FOR HYPERPARAMETER TUNING

- Use Case 2: Find the best model and best hyperparameter values for a selected model.



HYPERPARMETER TUNING PROCESS



TYPES OF SPLITTING THE DATASET

- The dataset is split into two parts namely, training dataset and test dataset.
- The models are trained using the training dataset for different combinations of hyperparameters and tested on the testing dataset.
- Generally, 70% of the data is used as training data and 30% is used as testing data

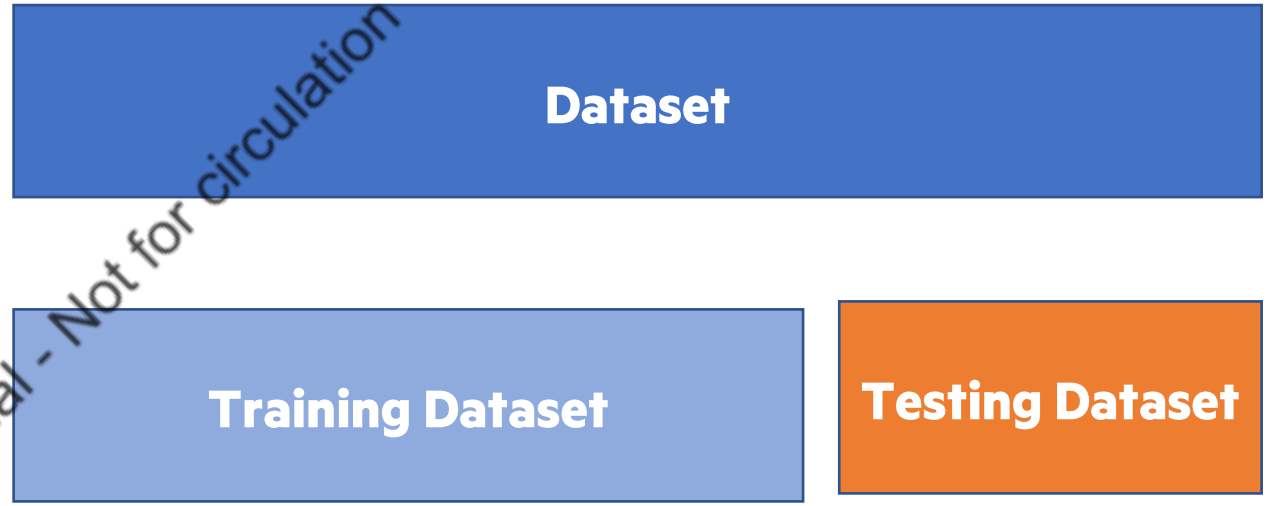


FIG. 2a Train Validation Split

TYPES OF SPLITTING THE DATASET

- Split the dataset into two parts i.e., training dataset and testing dataset.
- The training dataset is in-turn divided into “k” subsets.
- The machine learning model is trained “k” times.
- The performance of the machine learning model is an average of the performance over the k-folds.
- The final model is trained on the entire training dataset and tested on testing dataset.

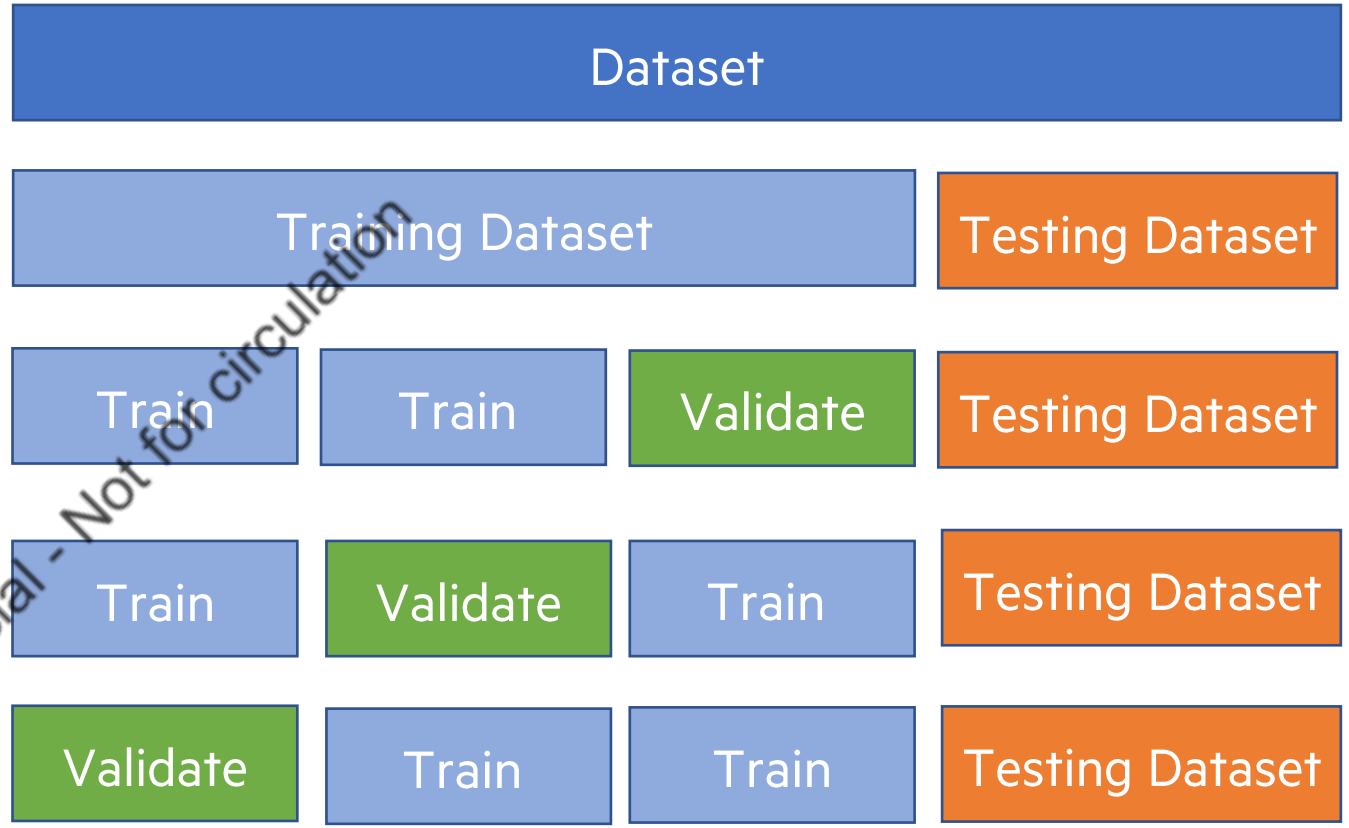


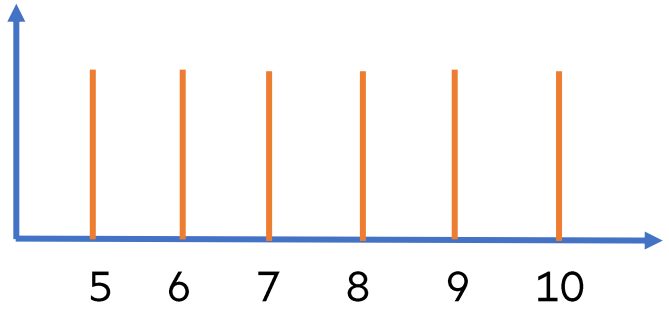
FIG. 2a Cross validation split

HYPERPARAMETER SEARCH SPACE

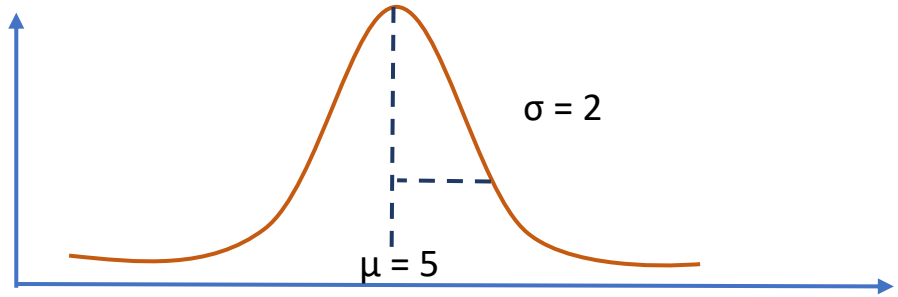
- The search space includes a set of values associated with a hyperparameter variable that should be tried during the training of the multiple models.

- There are two main types of hyperparameters namely,
  Discrete and Continuous.

a. [10, 12, 18, 24, 36, 45]
b. [Yes, No]
c. [True, False]
d. qUniform (5, 10)



a. normal(5, 2)



b. uniform(10, 50)

TOPIC-2

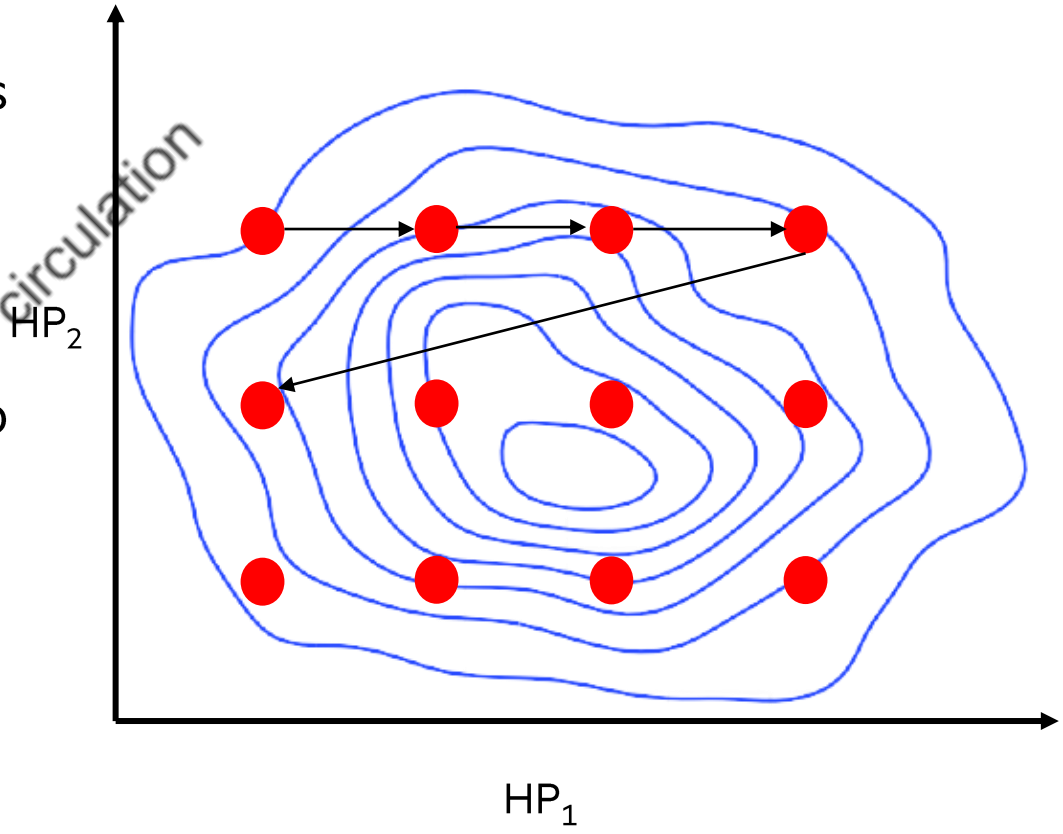
HYPERPARAMETER SAMPLING TECHNIQUES

HYPERPARAMETER SAMPLING TECHNIQUES

- **Grid Sampling**

- All the possible combinations of the hyperparameters are tried and the best combination is returned.

- Example:-
  - If HP1 = [2, 5], and HP2 = [60, 85, 90].
  - Then the 6 models trained using the two hyperparameters specified comprises
  - [2,60],
  - [2,85],
  - [2,90],
  - [5,60],
  - [5,85],
  - [5,90]



HYPERPARAMETER SAMPLING TECHNIQUES

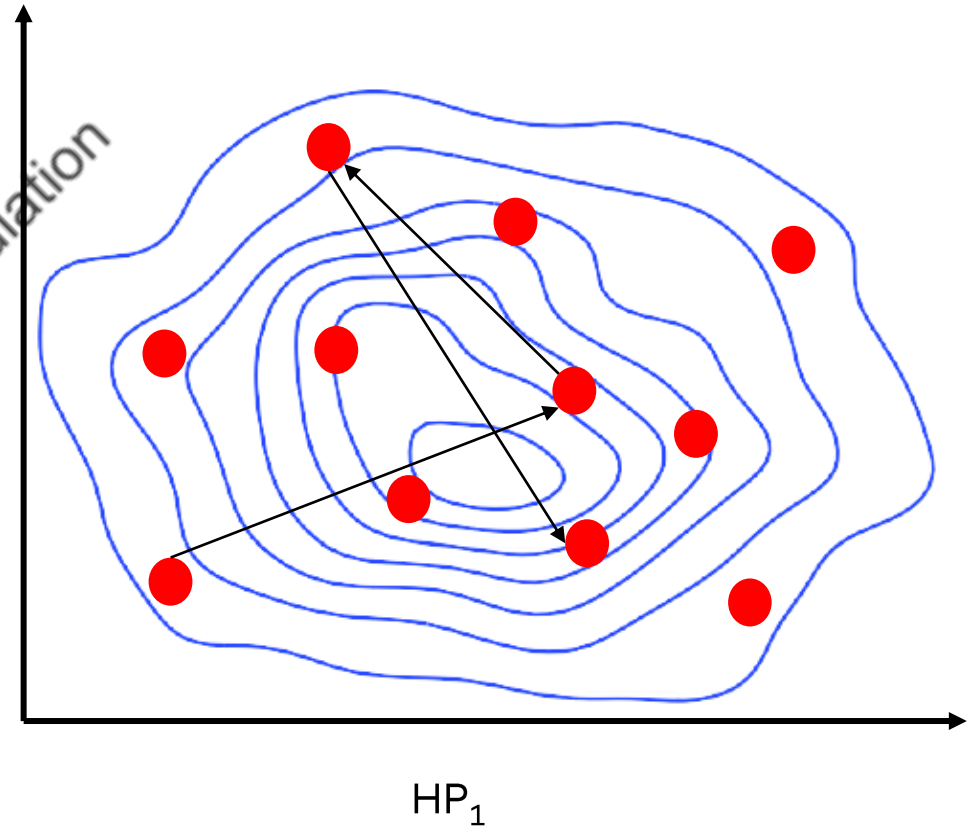
- **Random Sampling**

- The hyperparameter values are randomly selected from the specified values or range and models are trained.

- The best of values is provided based on the performance metric.

- Example:- If HP1 = [2, 5, 8, 10], and HP2 = uniform(150, 156).

- Then the models trained for the two hyperparameters specified may comprise {[8,155.6], [10,151.94], [8,154.82]}



HYPERPARAMETER SAMPLING TECHNIQUES

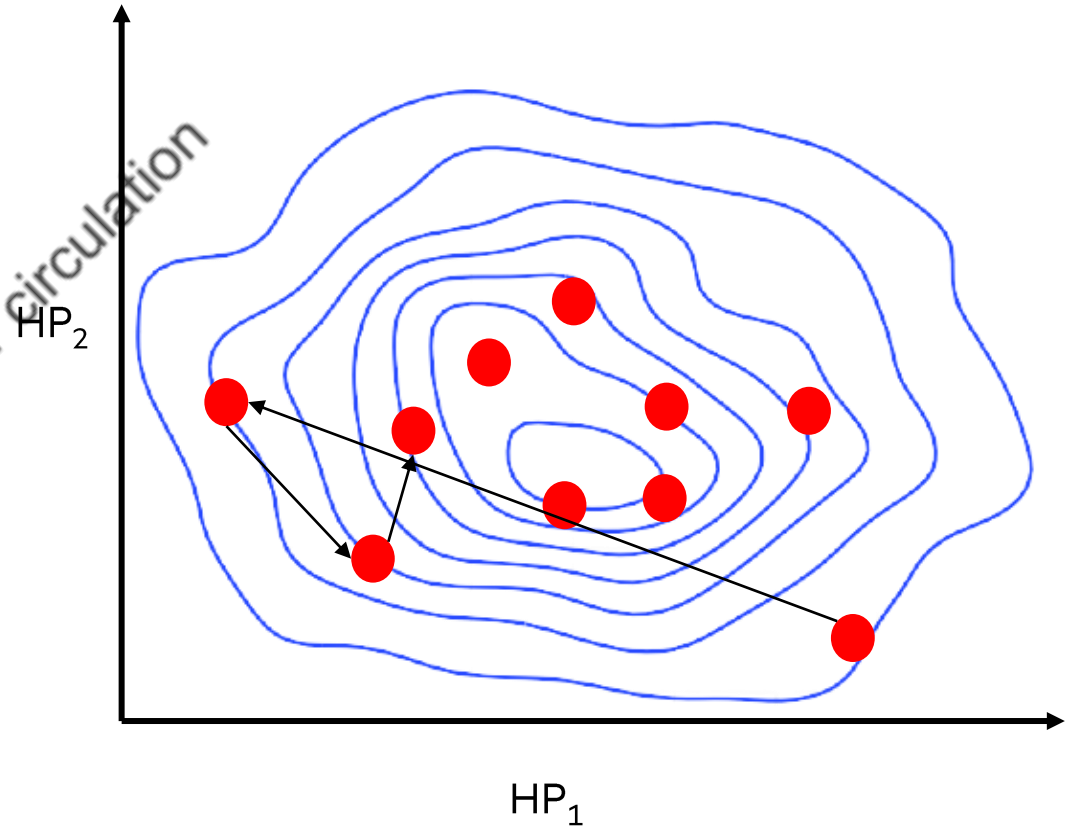
- **Bayesian Sampling**

- The value of the hyperparameters to be tried is decided using the Bayesian optimization algorithm.

- A probabilistic model is built for the function mapping between the hyperparameters and the performance metric.

- Example:- If HP_1 = [2, 5, 10], and HP_2 = normal(1, 7), then at iteration 'i' if the values [2, 1.9] resulted in a performance metric value of 'X'.

- For the iteration 'i+1' the algorithm provided the hyperparameter values (say [2, 1.1]) such that the model performance is better than 'X'.



TOPIC-3

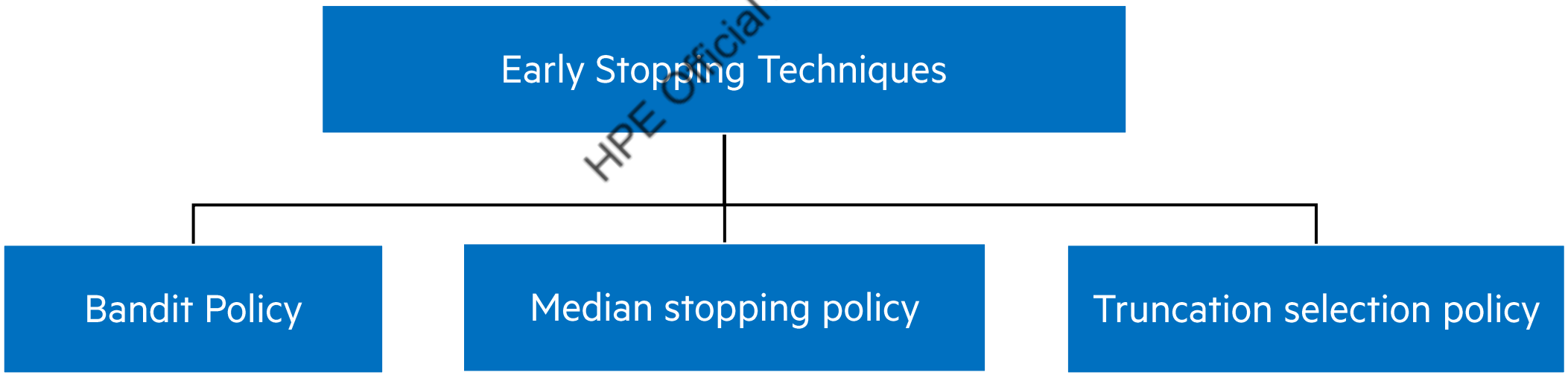
EARLY STOPPING TECHNIQUES

EARLY STOPPING TECHNIQUES

- The process of searching the best value of the hyperparameters is time consuming when the number of hyperparameters and/or the number of values in the search space is large.

- Therefore, we evaluate the trained models at specified intervals to check if the desired performance has been reached.

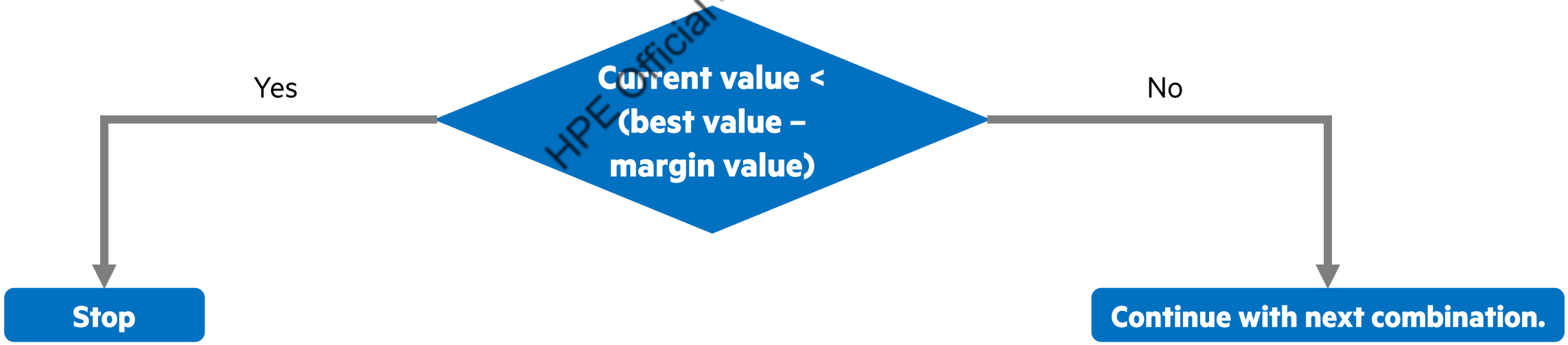
- If the performance metric has reached the desired value, then the process of hyperparameter can be terminated.



EARLY STOPPING TECHNIQUES

- **Bandit Policy**

- Models are evaluated by determining the variation with respect to the best performance metric value.
- First evaluation is done after a delay interval.
- Subsequent evaluations are performed at periodic intervals.

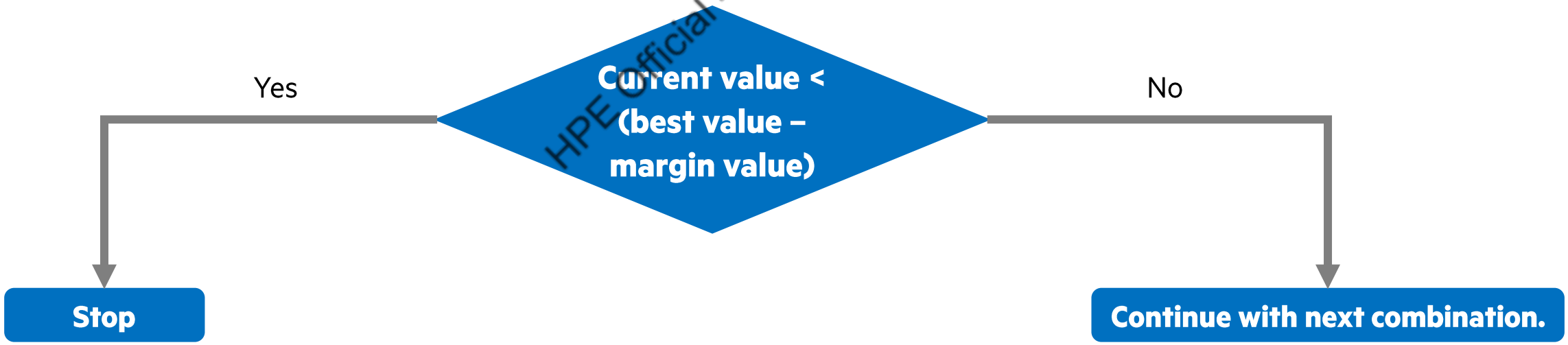


EARLY STOPPING TECHNIQUES

- **Bandit Policy - Example**

- Consider a margin value of 0.2 and evaluation interval is 5. Let the best performance value obtained be ‘Y’.

- The evaluation policy, after every 5 iterations compares the current performance metrics (say ‘X’) with (Y - 0.2). If the value of ‘X’ is lesser than (Y – 0.2), then the hyperparameter tuning is stopped.

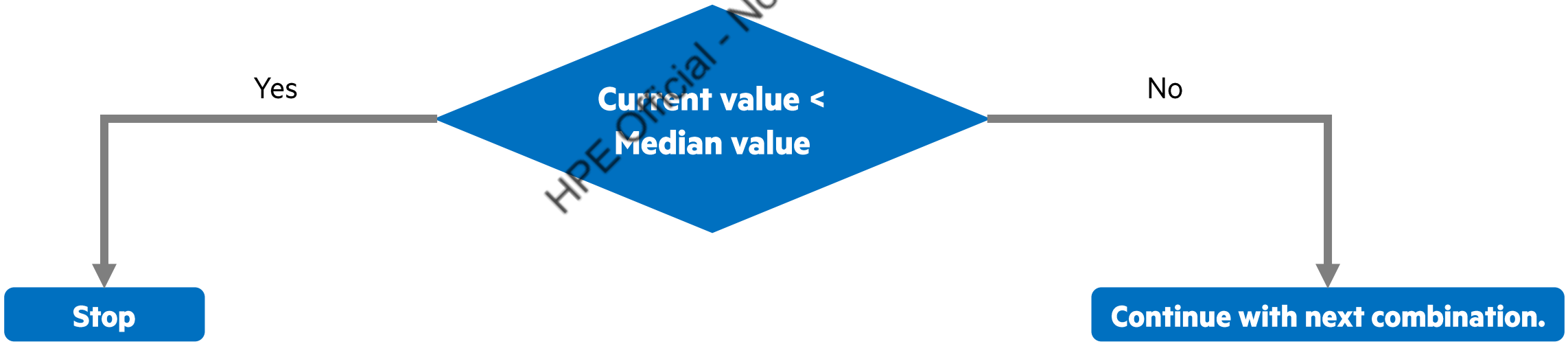


EARLY STOPPING TECHNIQUES

- **Median stopping policy**

- Median is the middle value in a set of sorted values.

- The hyperparameter tuning is stopped when the current value of the performance metric is worse than the median of the performance metrics obtained for the previous combinations of the hyperparameter values.

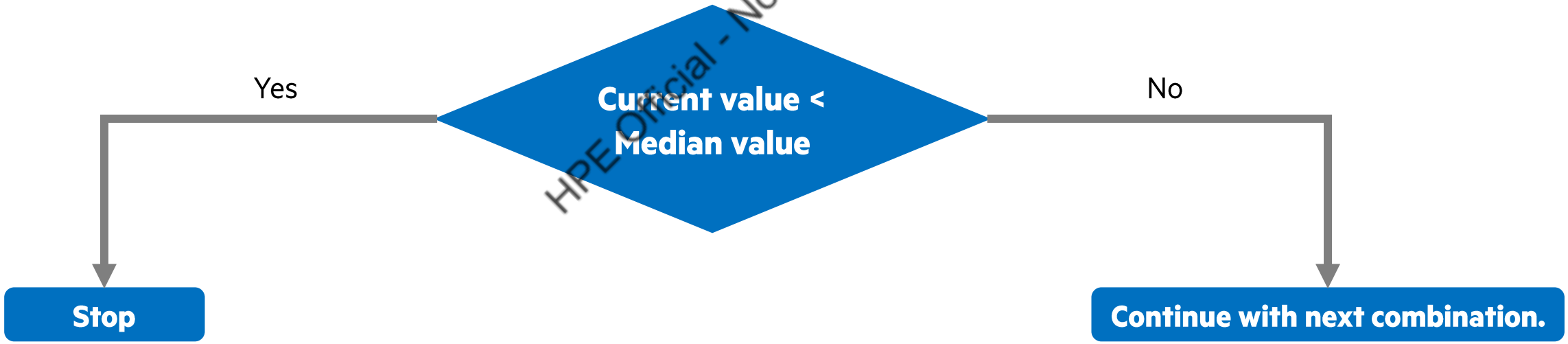


EARLY STOPPING TECHNIQUES

- **Median stopping policy - Example**

- Let the values of the performance metric (say accuracy) after training the model for 5 hyperparameter values be [0.62, 0.68, 0.71, 0.75, 0.78].

- At the 6th iteration if the performance metric value is less than the median value i.e., 0.71, then the hyperparameter tuning is stopped.

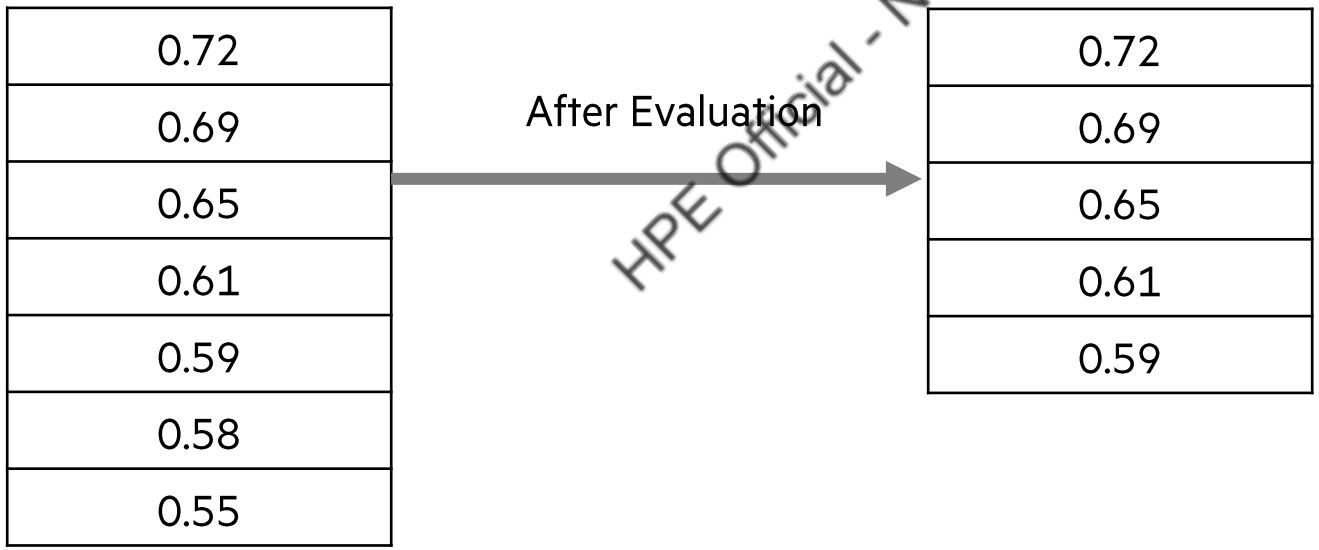


EARLY STOPPING TECHNIQUES

- Truncation selection policy

- In this approach, a truncation percentage value ('X') is provided. At the end of the evaluation interval, the least performing 'X' combinations of the hyperparameter values are removed.

- For example, if the value of 'X' = 20, then after the evaluation interval, two models with the least performance is removed.



THANK YOU

Hewlett Packard
Enterprise

MODULE 4: DEEP NEURAL NETWORKS

LESSONS:

Lesson-1

Artificial Neural Networks

- McCulloh Pitts model
- Perceptron model
- Multilayer Neural Networks
- Feed Forward Networks
- Backpropagation
- Gradient descent
- Cost Function

Lesson-2

Deep Neural Networks

- Introduction to DNN
- Convolutional Neural Networks
- Recurrent Neural Networks
- Long short-term Memory Networks
- Gated Recurrent Units.
- Transfer Learning.

Lesson-3

Real time Deployment

- Model Serving
- Use cases

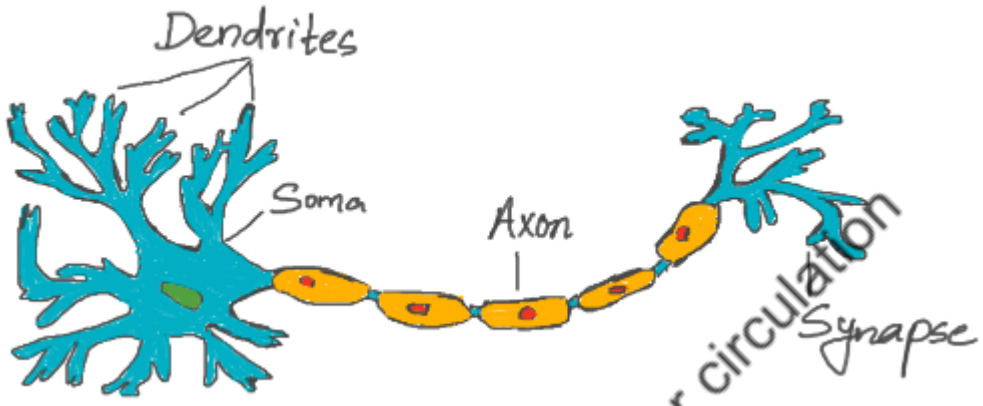
LESSON-1

ARTIFICIAL NEURAL NETWORKS

ARTIFICIAL NEURAL NETWORKS OVERVIEW

- The learning takes place based on a type of networks which simulates the functions of human brain.
- Neural Networks have layered structure to solve complex problems in logical fashion.
- It has also been widely used in
  - ➢credit card fraud detection.
  - ➢developing personalized treatment plans in medical industry.
  - ➢ Accurate Insurance provision based on customer needs.
  - ➢Self -driving cars.

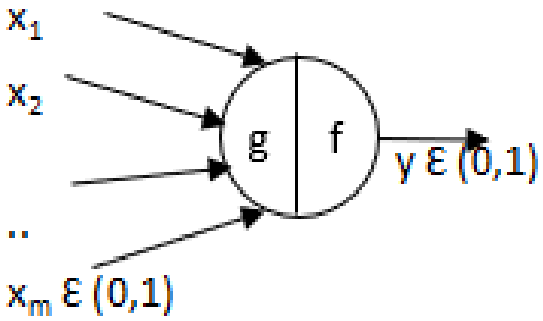
BIOLOGICAL NEURON



- **Dendrite**: Receives signals from other neurons.
- **Soma**: Processes the information received.
- **Axon**: Transmits the output of the processing.
- **Synapse**: Point of connection.

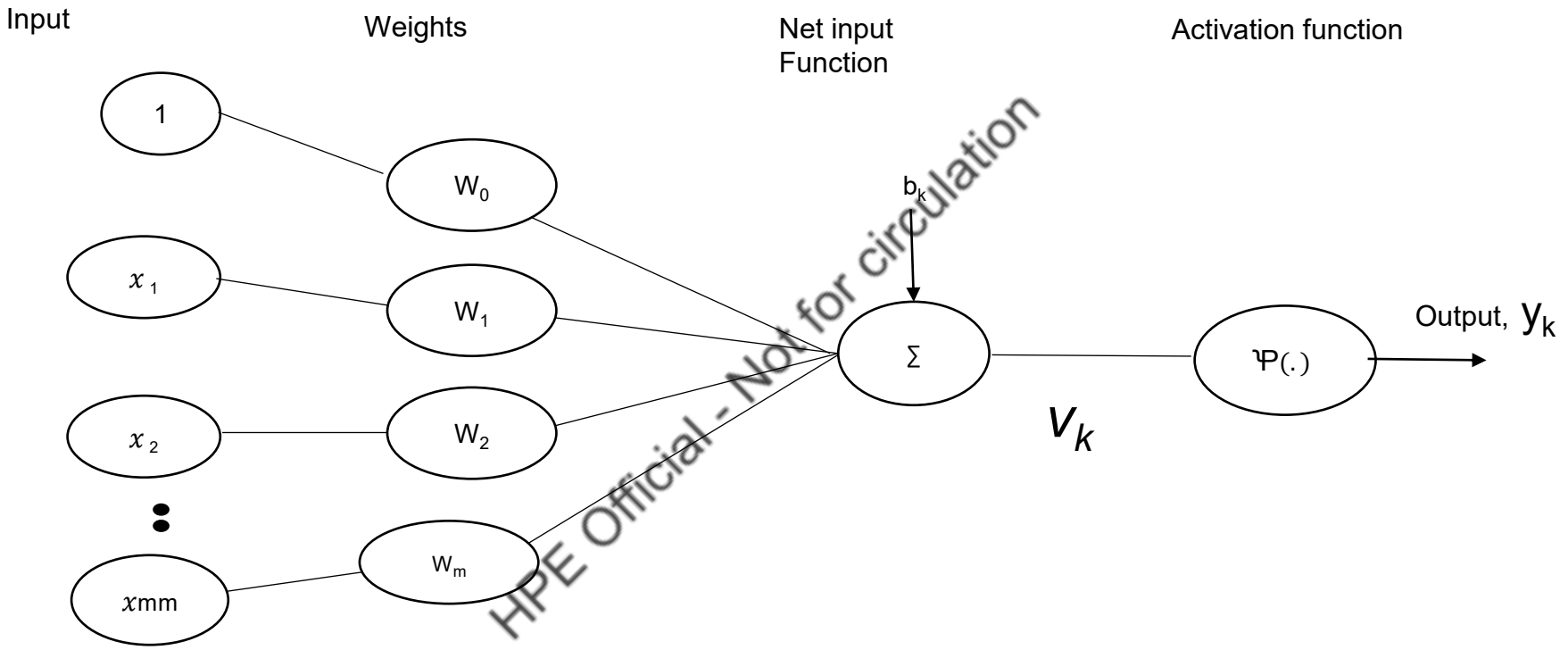
MCCULLOH PITTS MODEL –ARTIFICIAL NEURON

- McCulloch and Pitts created first artificial neural network by mimicking biological neuron functionality in 1943.
- The inputs are fed to ‘g’. It performs an aggregation (sum of binary inputs), and 'f' takes a decision based on the aggregation.
- If the sum exceeds a threshold value, the output is 1 else 0.
- This model works on linearly separable data.



PERCEPTRON MODEL

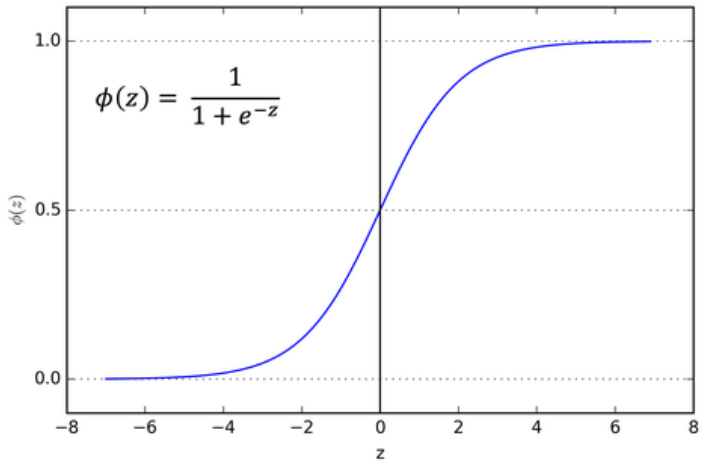
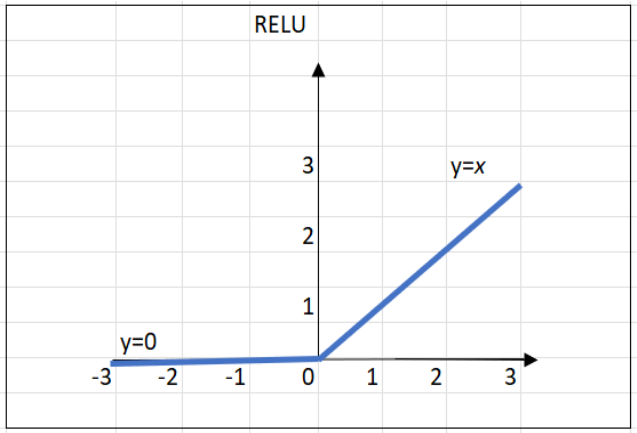
- In 1957, Frank Rosenblatt modified the above artificial neuron model known as perceptron.



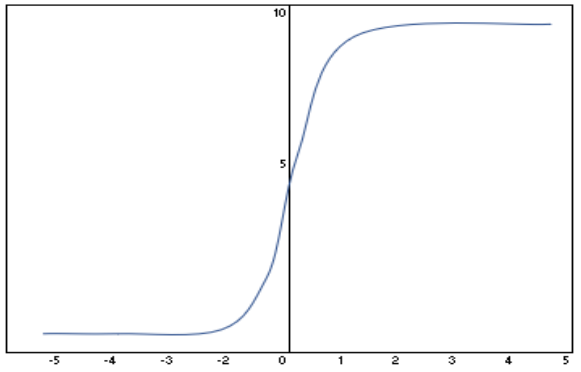
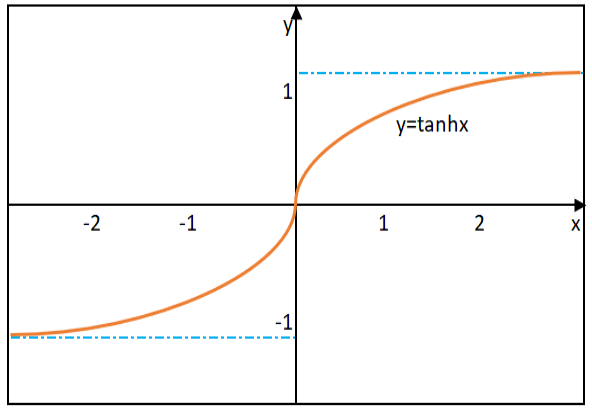
Linear Combiner Output, $$u_k = \sum_{j=0}^{m} \omega_{kj} x_j$$

$$Output, y_k = \Psi(u_k + b_k)$$

ACTIVATION FUNCTION

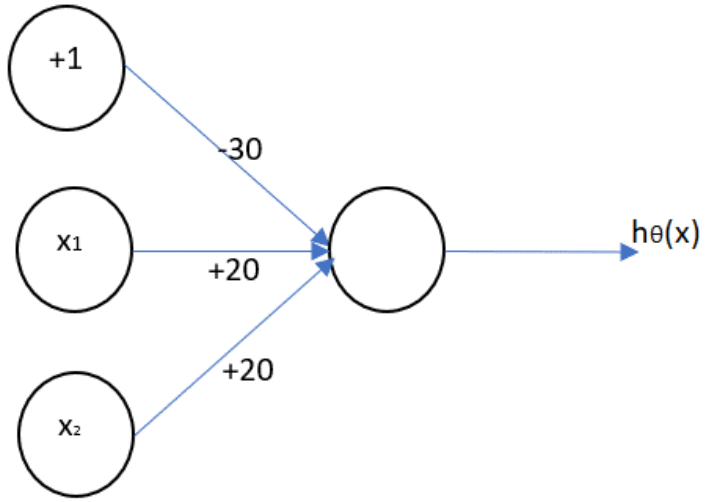
Activation function name	Equation	Plot of activation function
Sigmoid Activation Function	$f(x) = \frac{1}{1 + e^{-x}}$	
Rectified linear activation unit (ReLU)	$f(x) = max\ (0, x)$	

ACTIVATION FUNCTIONS

Activation function name	Equation	Plot of activation function
Softmax	$F(x)_i = \frac{e^{(x)_i}}{\sum_0^k e^{(x)i}}$	
Tanh function	$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$	

AND LOGICAL FUNCTION

- Consider x1, x2 Ɛ {0,1} and have y = x1 AND x2, w1=30, w2=20 and w3=20. The neural network can be implemented as shown

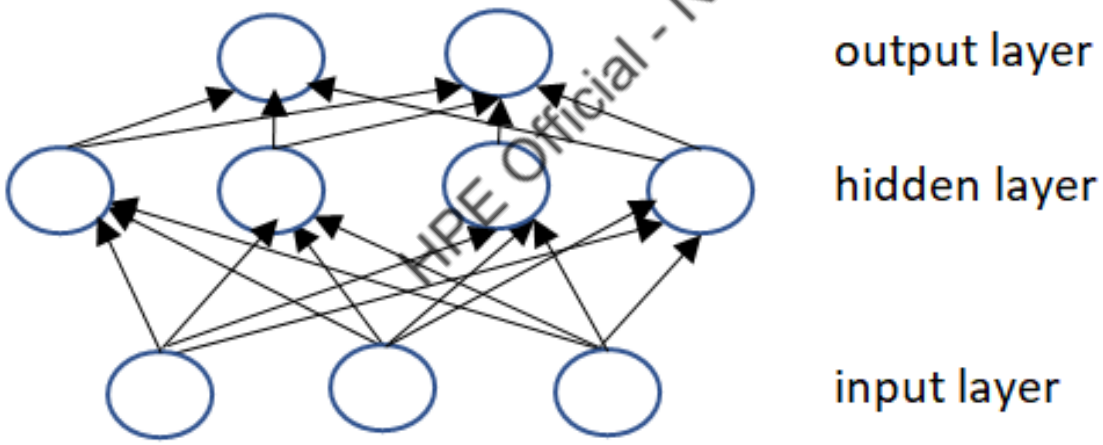


x_1	x_2	$h_\theta(x)$
0	0	f (-30) ≈ 0
0	1	f (-10) ≈ 0
1	0	f (-10) ≈ 0
1	1	f (10) ≈ 1

Computation, $h_\theta(x) = f(-30 + 20x_1 + 20x_2)$

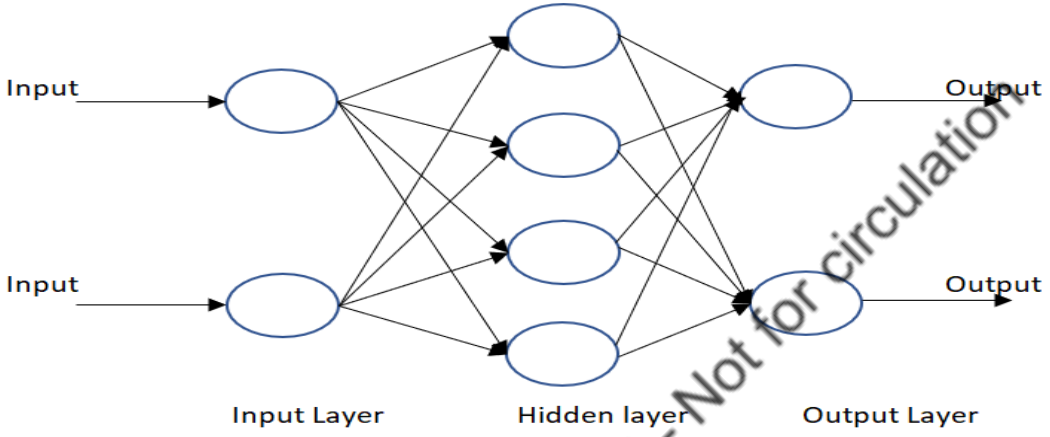
MULTILAYER NEURAL NETWORKS

- Multilayer network consists of input layer, hidden layer, and output layer.
- Output from each layer is connected to the inputs of next layer.
- Input layer receives the data that is considered for the algorithm



FEED FORWARD NETWORKS

- In case of feedforward networks, there is no feedback path present.



- The ground truth about the data is taken and is compared with the guesses

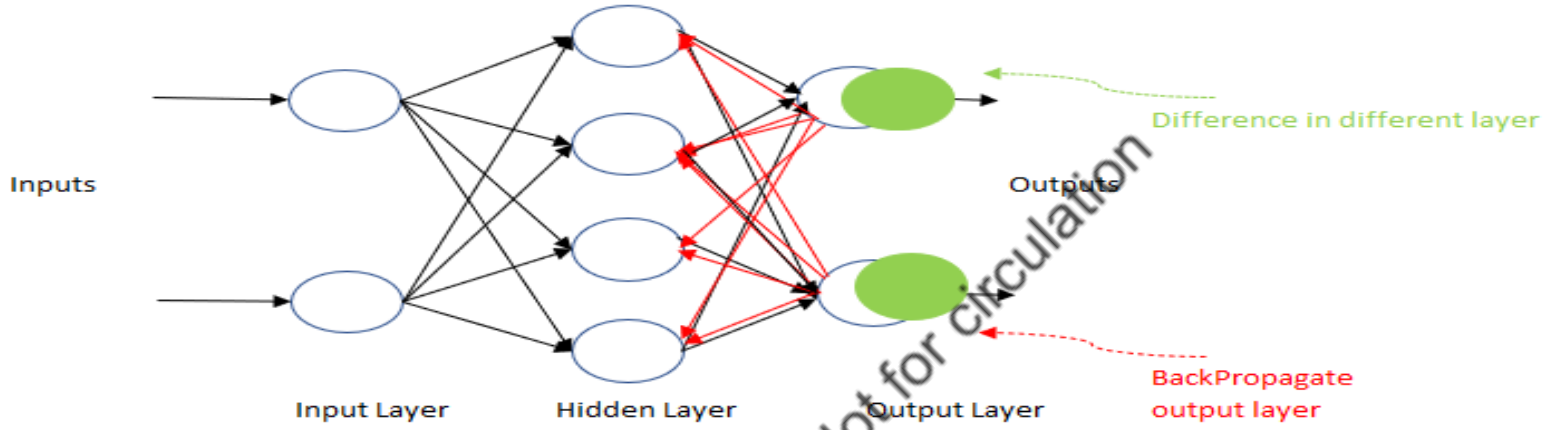
Ground truth - guess = error

- Depending upon the error, the network tries to adjust the weights to reduce the error.

Error * contribution of weights to error = adjustment.

BACKPROPAGATION ALGORITHM

- Backpropagation means “backward propagation of errors”.



The steps for backpropagation algorithm

1)Inputs enter through the input layer.

2)Weights are selected randomly and inputs are modeled using the weights.

3)Output is calculated for each neuron passing through the hidden layer.

4)Error is calculated which is the difference between actual output and desired output.

5)Again propagate in the backward direction to adjust the weights to reduce error.

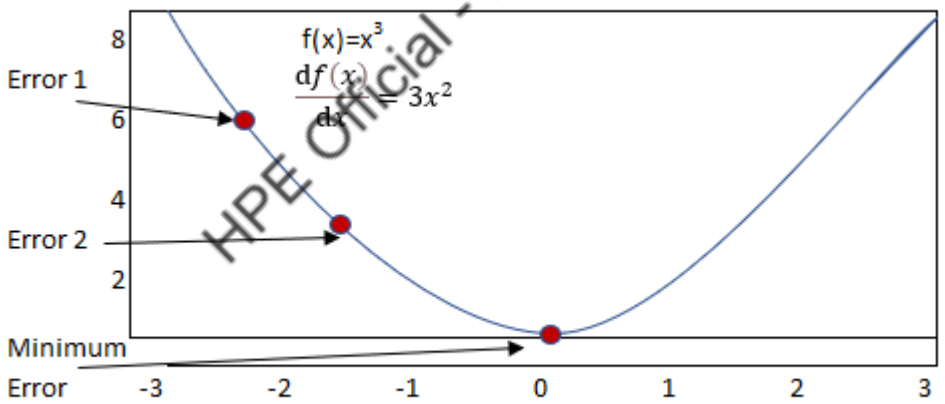
The processes are repeated until desired output is obtained.

GRADIENT DESCENT

- To optimize the cost function or the error of the model, gradient descent is used.

- **Gradient descent** (GD) is an iterative first-order optimization algorithm used to find a local minimum/maximum of a given function

- The error in the model must be reduced quickly without much resource wastage



TYPES OF GRADIENT DESCENT

- **Batch Gradient Descent**:
  - The error is calculated for each sample within the training dataset.
  - The model gets updated only after the entire training samples have been evaluated.
- **Stochastic Gradient Descent:**
  - The error is calculated for each training sample and parameters are updated then and there itself.
  - The model gets updated each time the error is evaluated.
- **Mini Batch Gradient Descent:**
  - Combination of the first two methods.
  - Training samples are divided into small batches and the model gets updated after each batch is updated on the error.

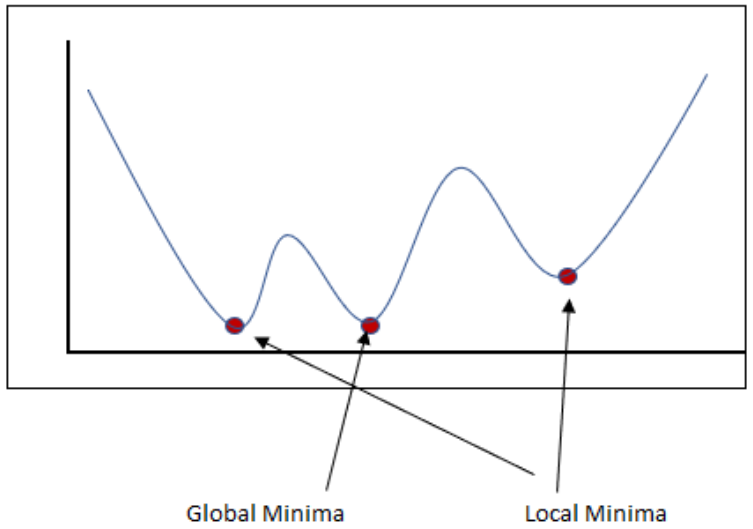
COST FUNCTION

- Cost function is a single value which gives a measure of performance of the neural network with respect to the prediction.
- It is the difference between the output predicted and the actual output.
- A cost function gives you an idea of how badly model is predicting.

$$\boldsymbol{Cost\ Function\ (J) = \frac{1}{n} \sum_{i=0}^{m} (h\theta\ (x)^{i} - y^{i})^{2}}$$

- In neural network each layer will have a cost function.
- Each layer's minima is known as local minima.
- The minimum value out of all values are considered

which is known as global minima.

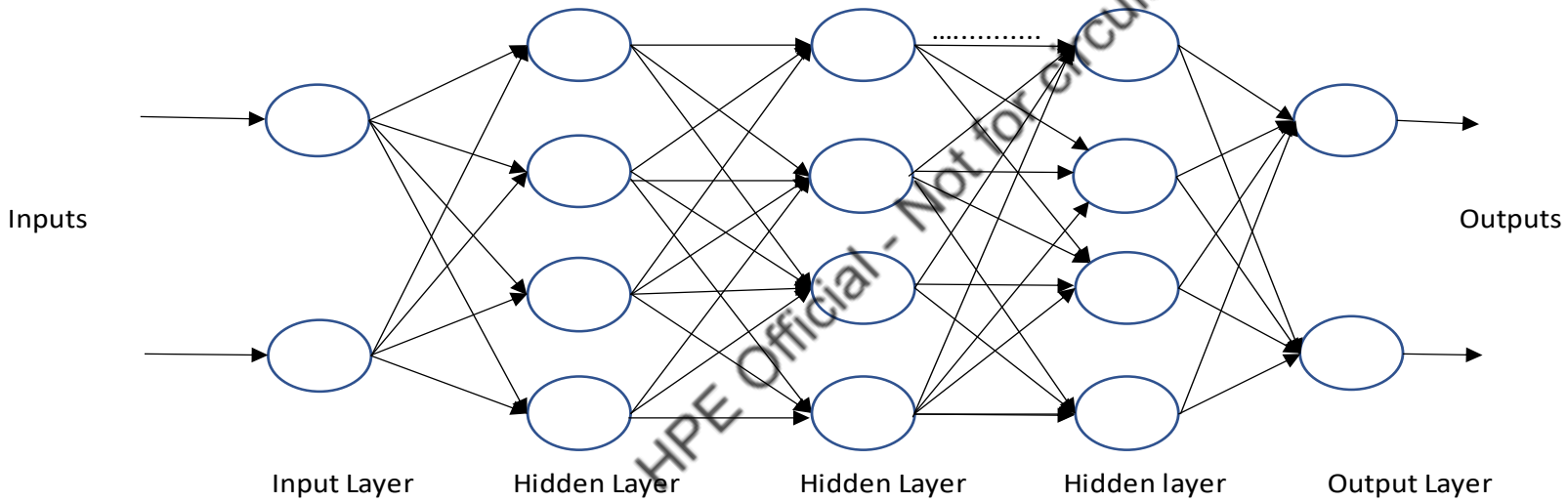


LESSON-2

DEEP NEURAL NETWORKS

INTRODUCTION TO DNN

- Deep neural networks (DNNs) are networks which has more than one hidden layer.
- Simple features in input data recombine from one layer to the next forming complex features.
- Hidden layers help to learn the features of the data in a form of feature hierarchy.



- Neural networks with one or two hidden layers are known as shallow networks and more than three are known as deep neural networks.

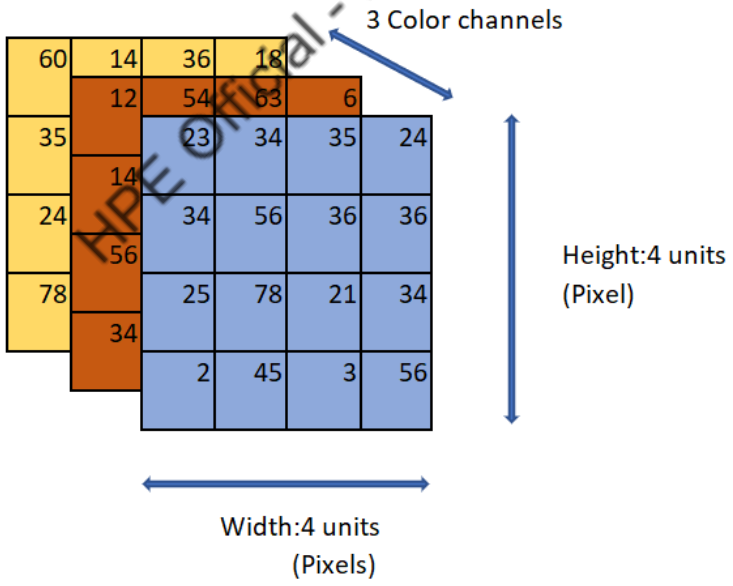
TRAINING A NEURAL NETWORK

- Multiple iterations known as *epochs* are carried out to train a deep neural network.
- Random initialization values are assigned to weights(w) and bias(b) values are assigned for the first epoch.
- Later the process is as follows.

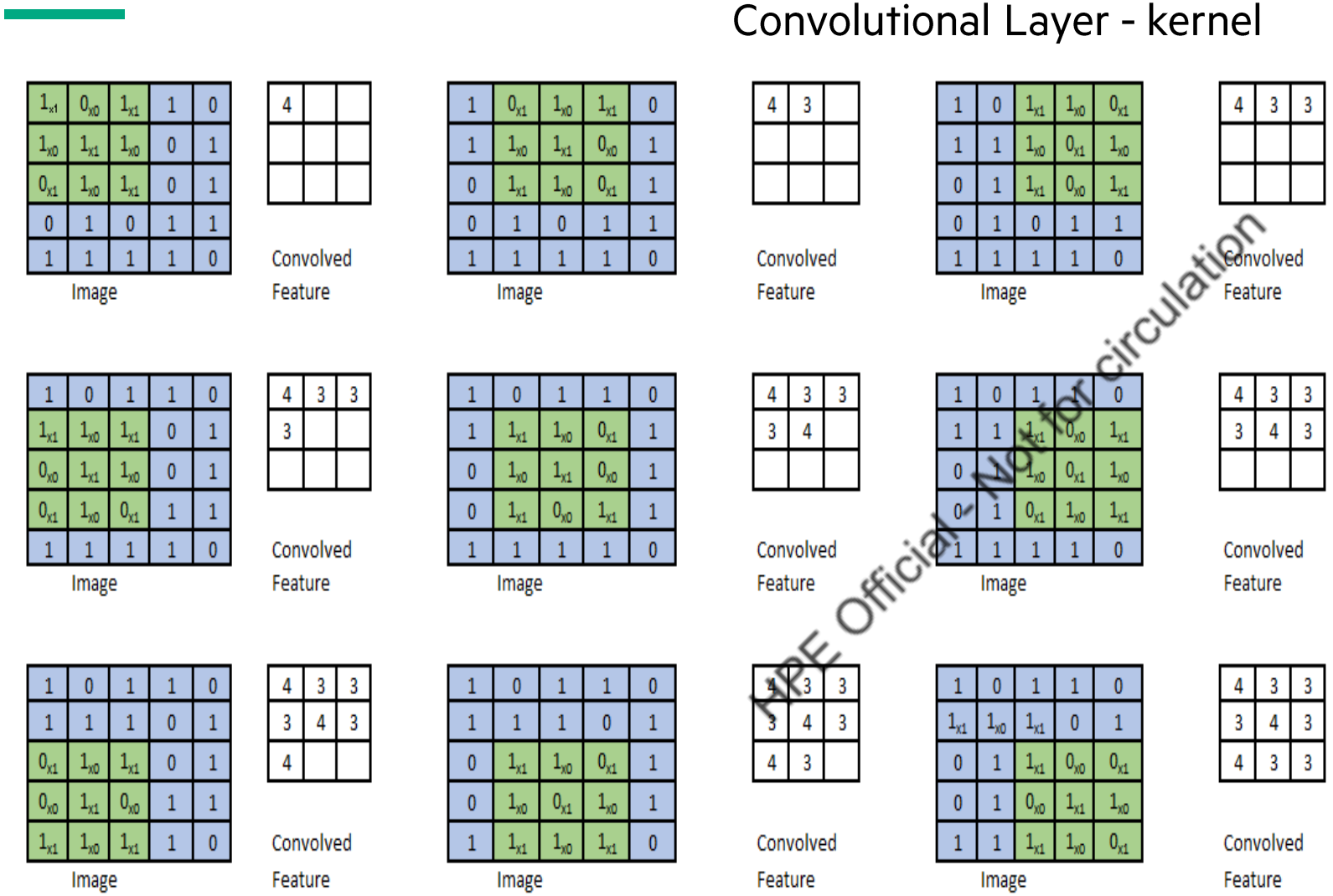
1) Features of data observations with known labels are generally grouped into batches. (also referred as mini-batches).
2) The neurons with activation functions applied are sent to the next layer until output are generated.
3) The error is calculated.
4) The revised weights are calculated, and adjustments are done to the neuron weights.
5) The next epoch repeats the forward pass with revised weights and bias values to improvise the model accuracy.

CONVOLUTIONAL NEURAL NETWORKS

- Convolutional Neural Networks (CNNs) are Deep Learning methodologies that take an input image and give importance (weights and biases) to distinct entities in the image, allowing them to be distinguished.
- Like the neuron connection pattern in the human brain.
- Capture the Spatial and Temporal correlations in a picture by applying filters.
- Consider an example of a three-color input image (red, green, and blue) (RGB).



CONVOLUTION NEURAL NETWORKS

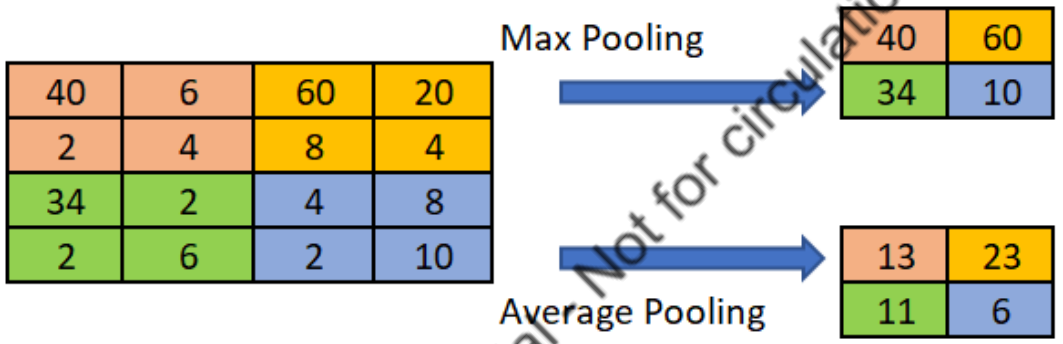


- Convolution is measuring how two functions overlap in an integral way as they pass over.
- One matrix is a kernel, and the other is the data portion overlapping with the kernel.
- A kernel helps to extract features from the image.

CONVOLUTIONAL NEURAL NETWORKS

Pooling/sub-sampling

- Limits the number of parameters and calculations in the network by steadily shrinking the spatial dimension of the representation.

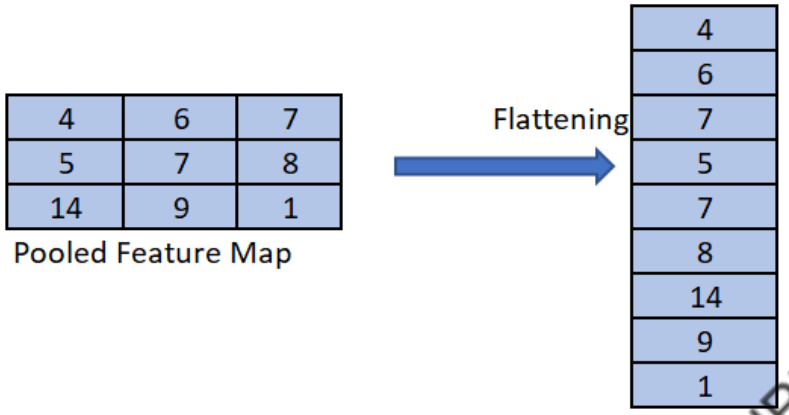


- There are three types of pooling: maximum pooling ,average pooling and minimum pooling
- Max Pooling extracts the maximum value from the portion of the image covered by the Kernel.
- Average Pooling returns the average of all the values from the image's Kernel section.
- Minimum pooling gives minimum value.

CONVOLUTIONAL NEURAL NETWORKS

Flattening

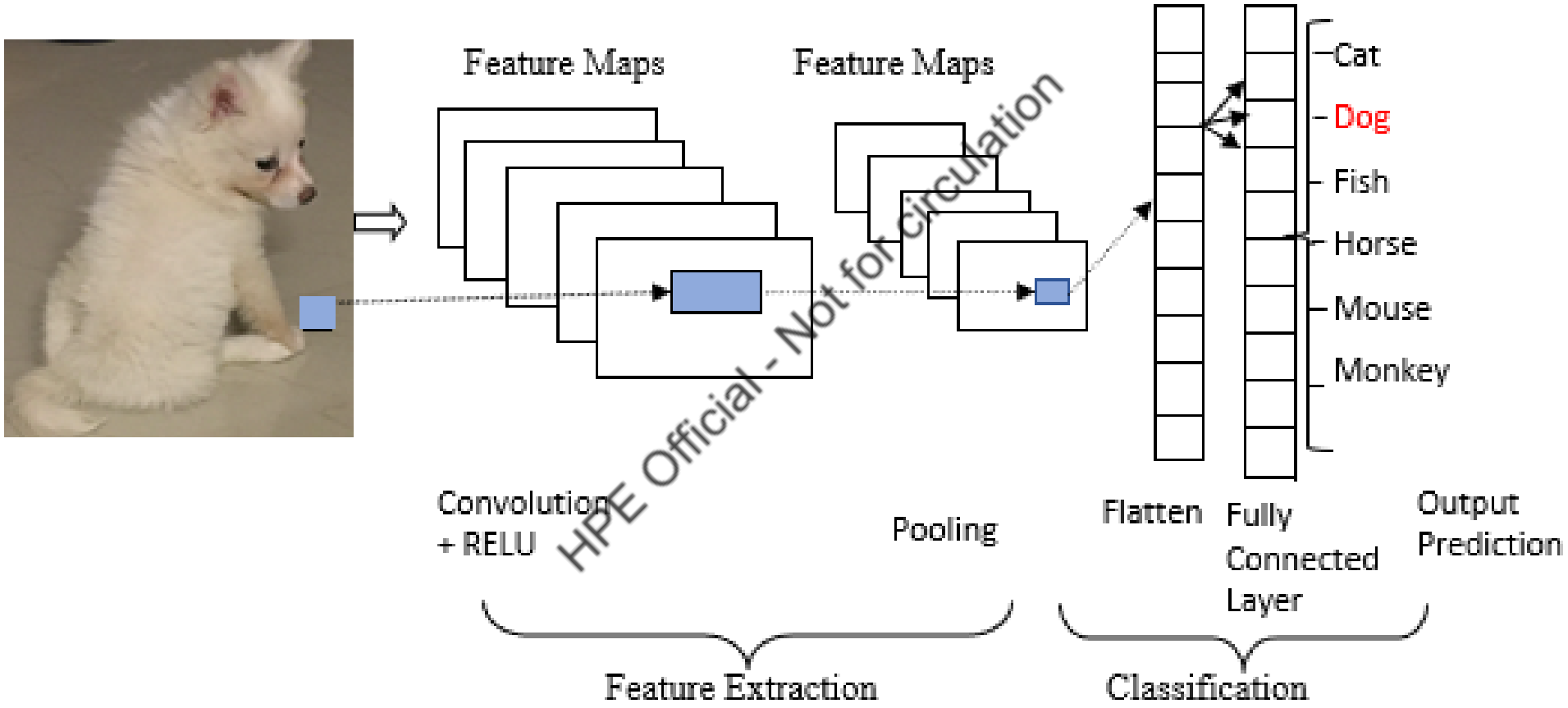
- Flattening is done to convert the matrix to a 1-D array.



Fully Connected Layer

- The fully connected layer is the layer that connects all the neurons with all other neurons in the next layer
- The fully connected layer (FC) has weights and biases used to connect the neurons between two layers.

CONVOLUTIONAL NEURAL NETWORKS –AN EXAMPLE

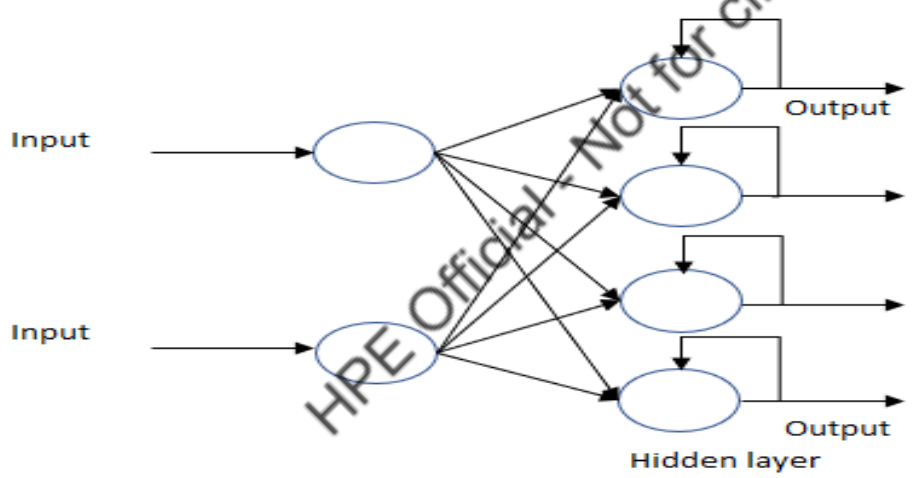


APPLICATIONS OF CONVOLUTION NEURAL NETWORKS

- Speech recognition
- Image classification
- Facial Recognition
- Climate predictions
- Natural Language Processing

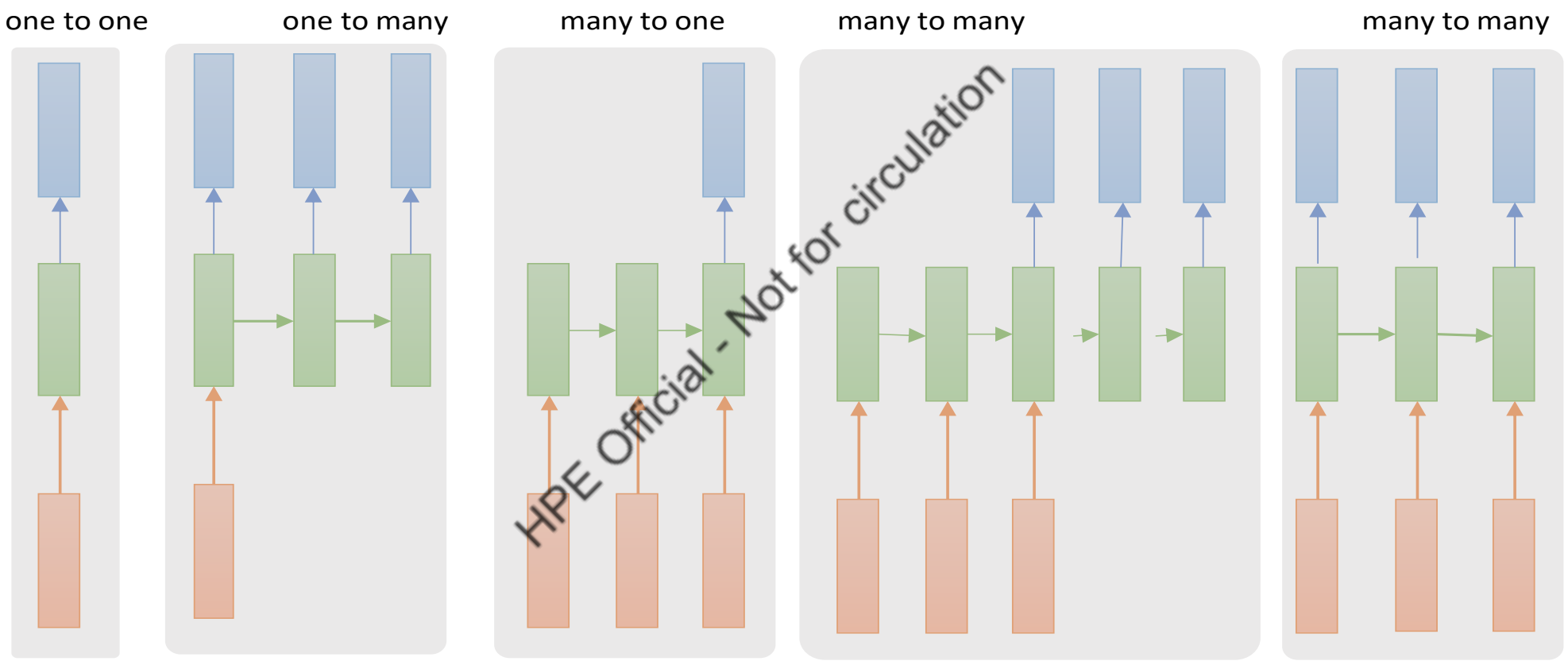
RECURSIVE NEURAL NETWORKS

- In RNN, there is a path from the output towards the input layer of the system.
- The input gets influenced partly by the feedback signal.
- The category of artificial neural network which uses feedback path are known as Feedback Neural Net.



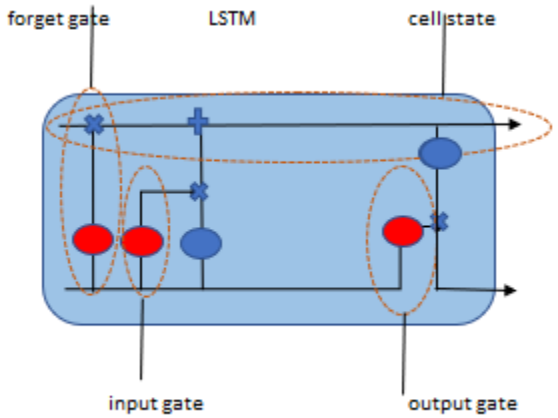
- As recurrent neural networks have memory, they can remember previous inputs.

TYPES OF RNN



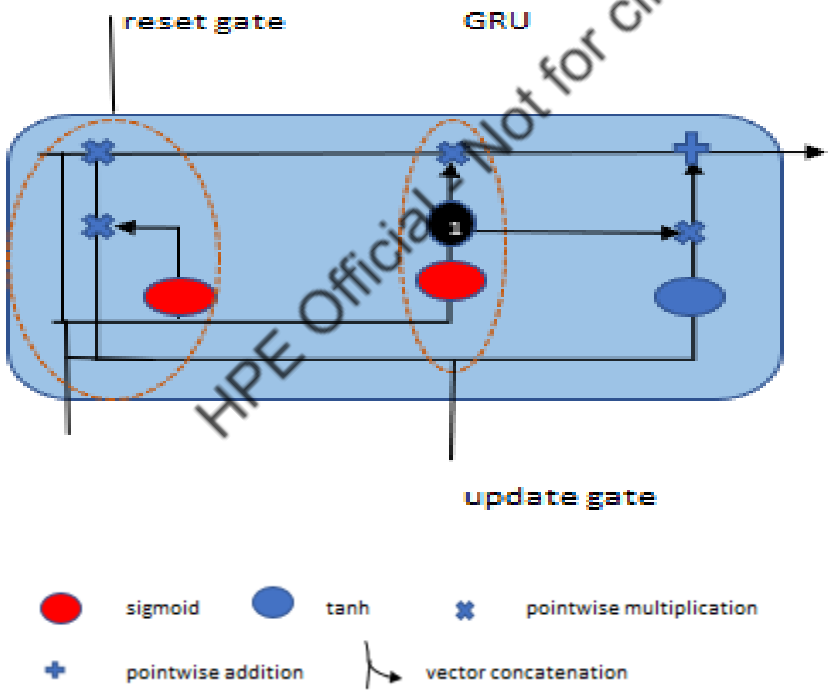
LSTM (LONG SHORT-TERM MEMORY)

- RNN can remember things for a short duration of time.
- In LSTM, a memory structure is introduced, which is controlled by a gate that decides the things to remember, things to forget and the things to output.
- LSTM is very productive for time series forecasting based problems.
- LSTM handles selective filtering of data based on gates.
- LSTM has three gates
- i) Forget gate ii) Input gate and iii) Output gate.
- Forget gate decides what information we are going to throw away from the cell.
- Input gate decides which information is allowed to pass inside the cell layer
- Output gate decides what we are going to output.



GATED RECURRENT NETWORK(GRU)

- The forget and input gates are replaced with an update gate, which combines both the types.
- GRU has only two gates known as reset gate and update gate.
- GRU's are computationally faster than normal LSTM with three gates.

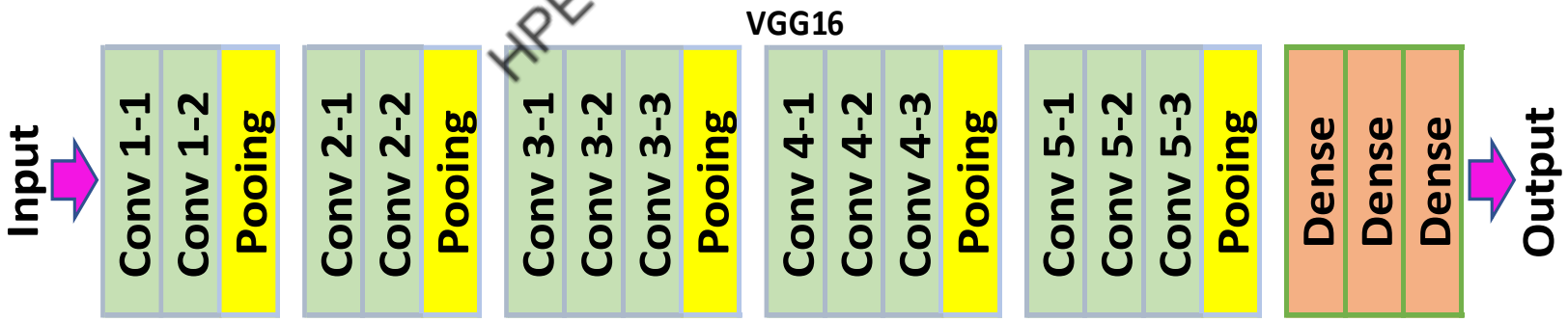


TRANSFER LEARNING

- In transfer learning, the knowledge acquired by the machine learning model training can be utilized for a different but related problem.
- A simple classifier model trained to detect sunglasses can be used for detecting face masks.
- Training a machine learning model means we're updating the weights and biases according to the data set.

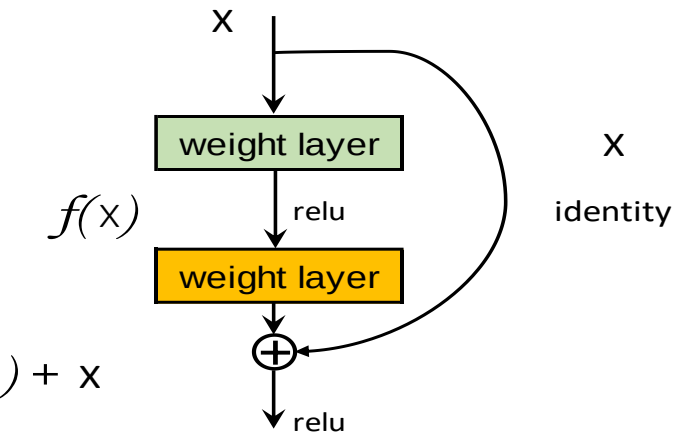
VGG16

- VGG16 is a 16-layer multilayer network of 13 convolutional layers, each with a 3x3 filter size and fully linked layers. It was developed by Visual Graphics group



RESNET 50

- ResNet is an abbreviation for residual networks.
- It uses residual learning instead of learning from features.
- Residual is the subtraction of feature learned from input of that layer. $f(x) + x$
- The future layer is formed by combining the preceding layers.
- Skipping one or more layers as shown in figure results in shortcut connections.
- Skip connections do identity mapping, and their outputs are added to stacked layers.
- ResNets optimizes easily compared to other networks where training error increases as depth increases.



DEMO

- Github link

LESSON-3

REAL TIME DEPLOYMENT

MODEL SERVING - REAL-TIME DEPLOYMENT

- **Data Wrangling**
  - Since the data scientist may have done many transformations to the data for training like one-hot encoding.
  - The same transformation needs to be provided to the real-time data from various sources
- **Storage**
  - Need to reduce the storage size of the model without much impact on the performance.
- **API**
  - According to the service structure of the ML model, the API structure needs to be carefully chosen.
- **Scheduling**
  - Scheduling can be done periodically, or it can be triggered when the performance goes beyond a threshold

MODEL SERVING : REAL-TIME DEPLOYMENT

- **Logging and Monitoring**

  - Proper logging of data and monitoring of health and conditions of deployment is crucial for making the model deployment fault-tolerant.

- **Security**

  - ML models are usually dealing with critical data, and ML models take many crucial decisions.
  - Hence the security breach of the deployment can cause serious harm.

- **Visualizations and User Interfaces**

  - The visualizations and UIs to be designed according to the target users

USE CASES:- MEDICAL IMAGE ANALYSIS

A convolutional neural network can identify edges, shapes, and even more sophisticated features over an image.

Steps involved in medical image analysis

1. Selection of data set, medical image data set is available from various sources.
2. Preparation of data, Image augmentation process, can be used to expand the size of the data set.
3. Selection of CNN model, the filter sizes, pooling layer, and stride lengths.
4. Model iteration
5. Evaluation of the model.

CNNs for Skin Cancer Diagnosis

- Skin cancer or Melanoma is a deadly disease.
- The diagnosis of skin cancer is carried out on the dermatoscopic images taken with a high-resolution magnifying camera.
- CNN can be utilized for skin cancer diagnosis.

USE CASES:- MEDICAL IMAGE ANALYSIS

Visual Data Processing for Tumor Detection

- The tumor images can be classified based on the visual features of the cancer images.
- The images and its masks can be visualized, and classification model can be trained using transfer learning modules such as ResNet-50
- This network has the capability to classify images into various categories, which can be utilized in identifying if the image has tumor or not.
- In addition, computer vision algorithms can also be used for the detection of tumor cells.

DEMO

- Github link

THANK YOU

Hewlett Packard
Enterprise

MODULE 4: DISTRIBUTED DEEP LEARNING

LESSONS:

Lesson-1

Architecture

- Introduction
- Need for distributed training
- Data Parallelism
- Model Parallelism

Lesson-2

Training

- Data distribution in the cluster
- Model distribution in the cluster
- Concurrent training

Lesson-3

Aggregation

- Synchronous update
- Asynchronous update
- Comparison between the two approaches

Lesson-4

Evaluation

- Additional Metrics to be considered in distributed training
- Examples for some performance metrics.

Lesson-5

Inference

- Use of ensemble learning approach
- Use of knowledge distillation techniques

Lesson-6

Conclusion

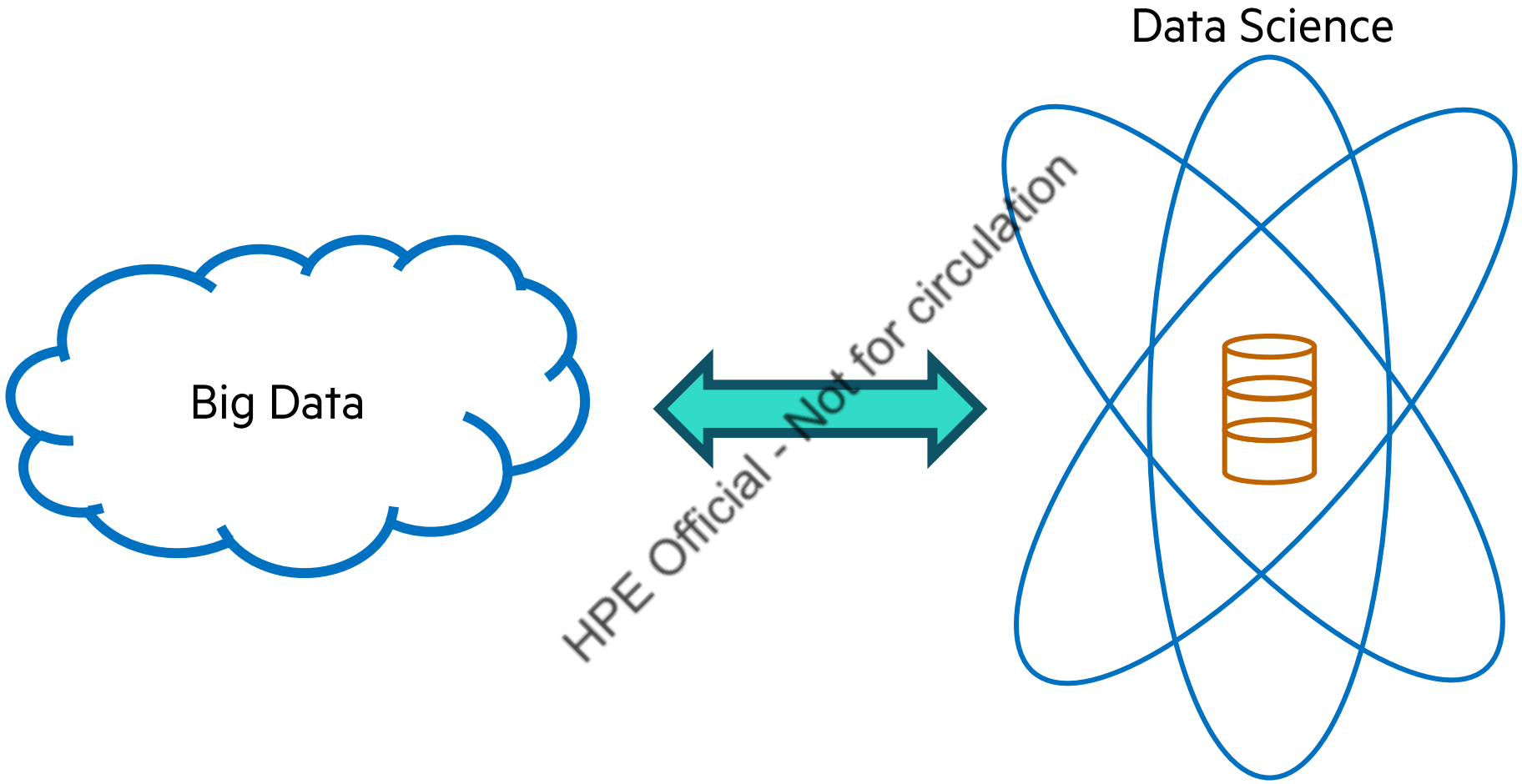
- Advantages
- Applications

LESSON-0

INTRODUCTION

LESSON 0:

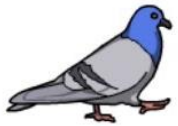
Introduction



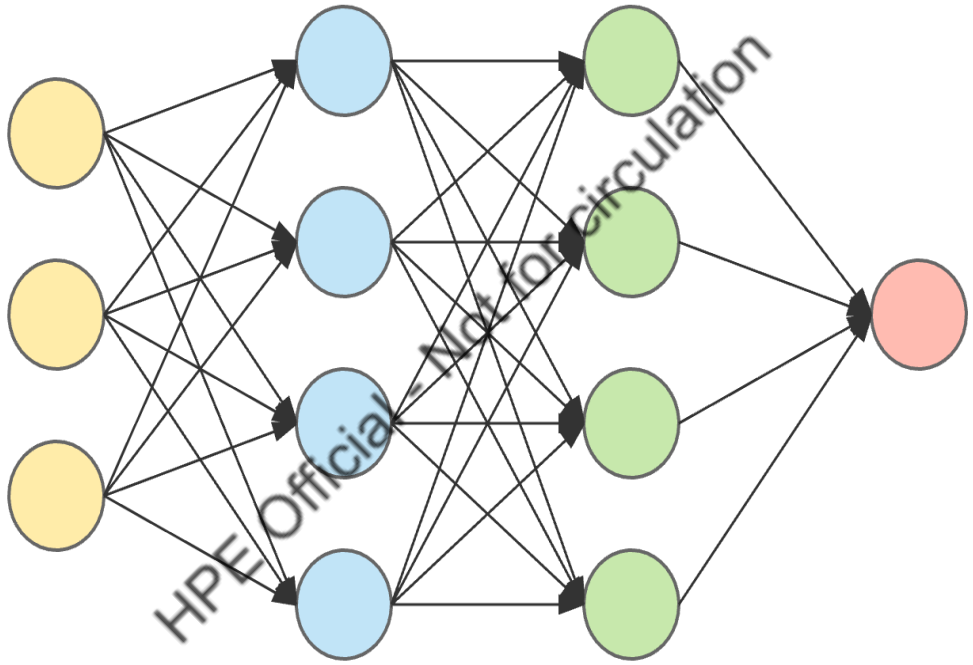
INTRODUCTION

- By 2021, the volume of data was predicted to reach 30 zettabytes (ZB) to 35ZB (zetta = 1021 bytes) according to a survey conducted by International Data Corporation (IDC).
- Bigdata includes performing operations on high volume of data. For example, in terabytes 1012 bytes.
- Deep learning models trained using the Bigdata have shown an exponential increase in the performance as compared to training with small datasets.
- Even with the usage of GPUs the deep learning models take days or months for completing the training.
- Therefore, there is a need to reduce the training duration of the deep learning models.

INTRODUCTION



Deep Learning Network



Frog

Pigeon

Fish

Cow

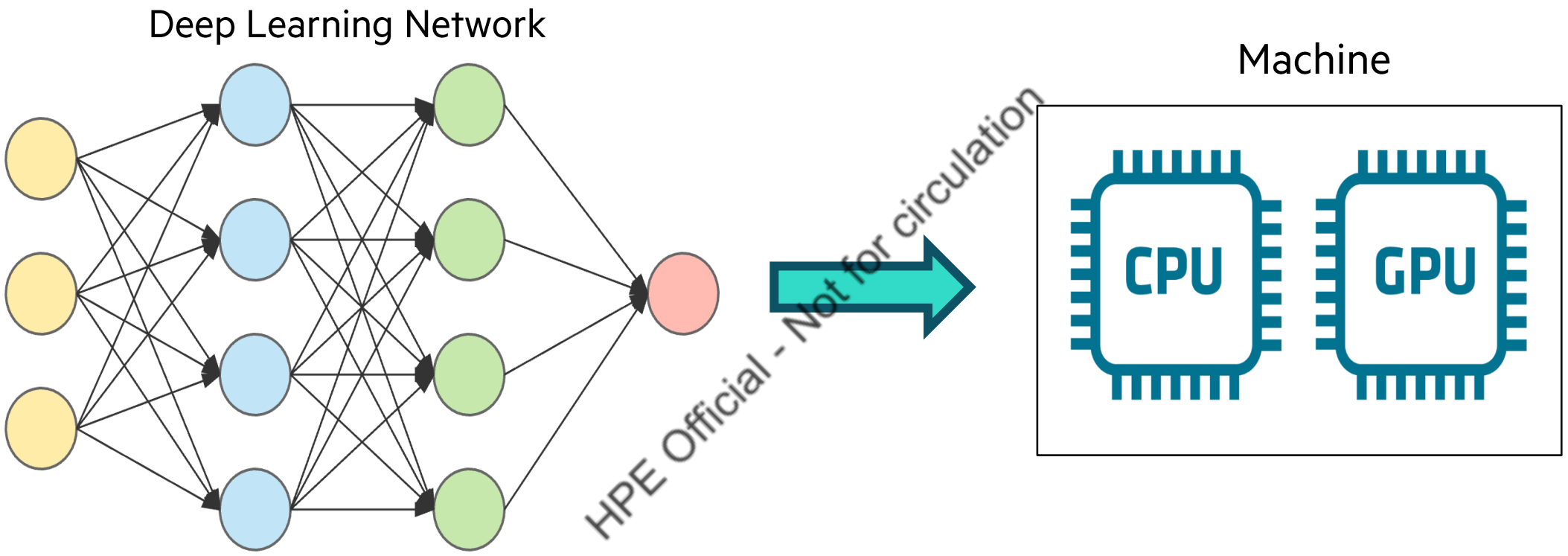
Training the network using the equation

$$w_{new} = w_{old} + \varepsilon * \nabla_L (w_{old}, D)$$

LOCAL TRAINING

- Used to share the computations involved in the deep learning models for reducing the training time.
- **Local (Parallel) Training**
- In this technique, the data and model is stored on a single machine with multiple cores.
- The multiple cores are used to train multiple mini batches of the model in parallel.
- Further, the GPUs present in machine can be shared for computationally intensive tasks such as matrix multiplication.

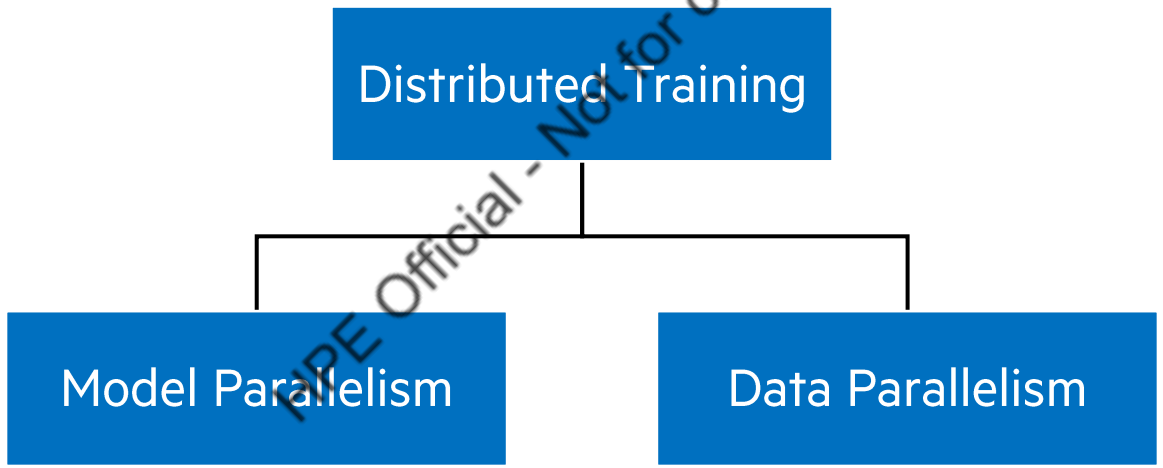
LOCAL TRAINING



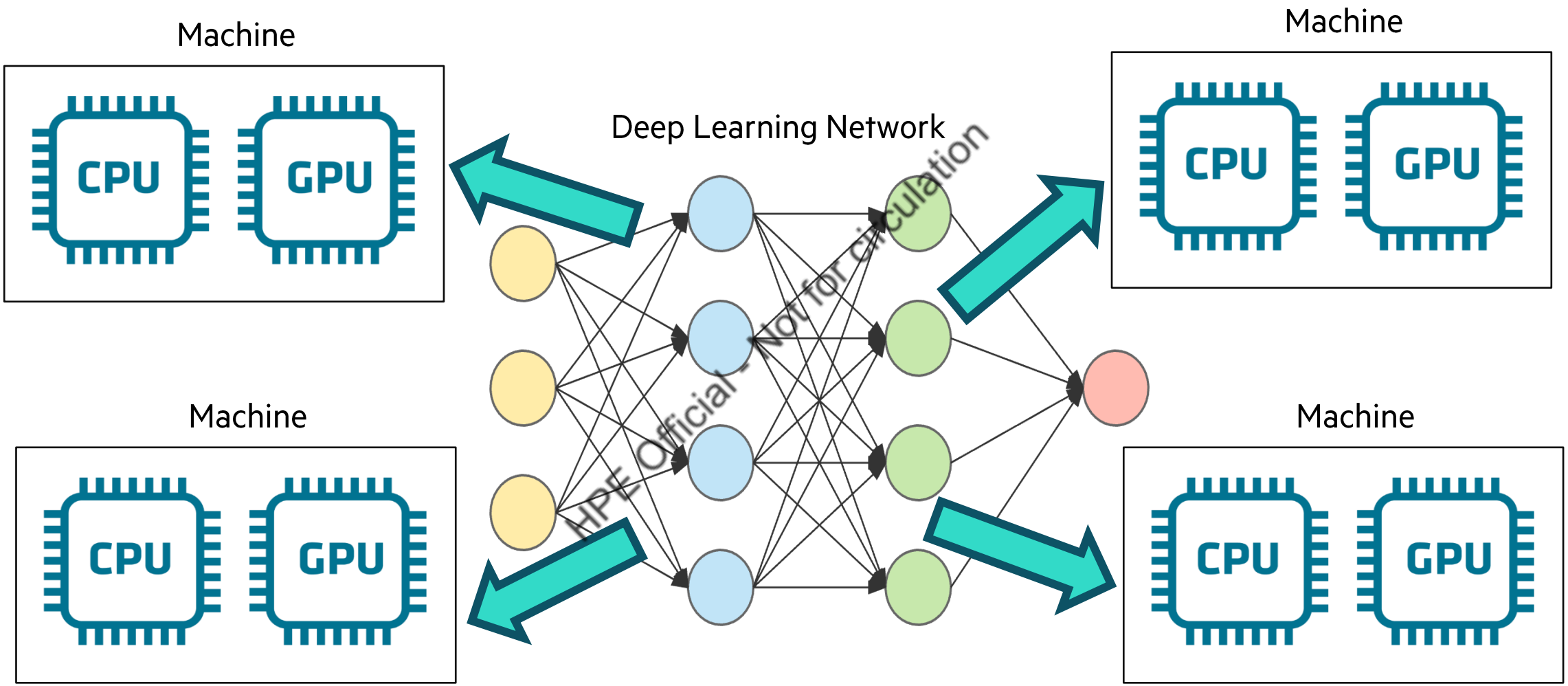
DISTRIBUTED TRAINING

- Used to share the computations involved in the deep learning models for reducing the training time.

- **Distributed Training:** In this technique the data and the model are stored or distributed across multiple machines.



DISTRIBUTED TRAINING

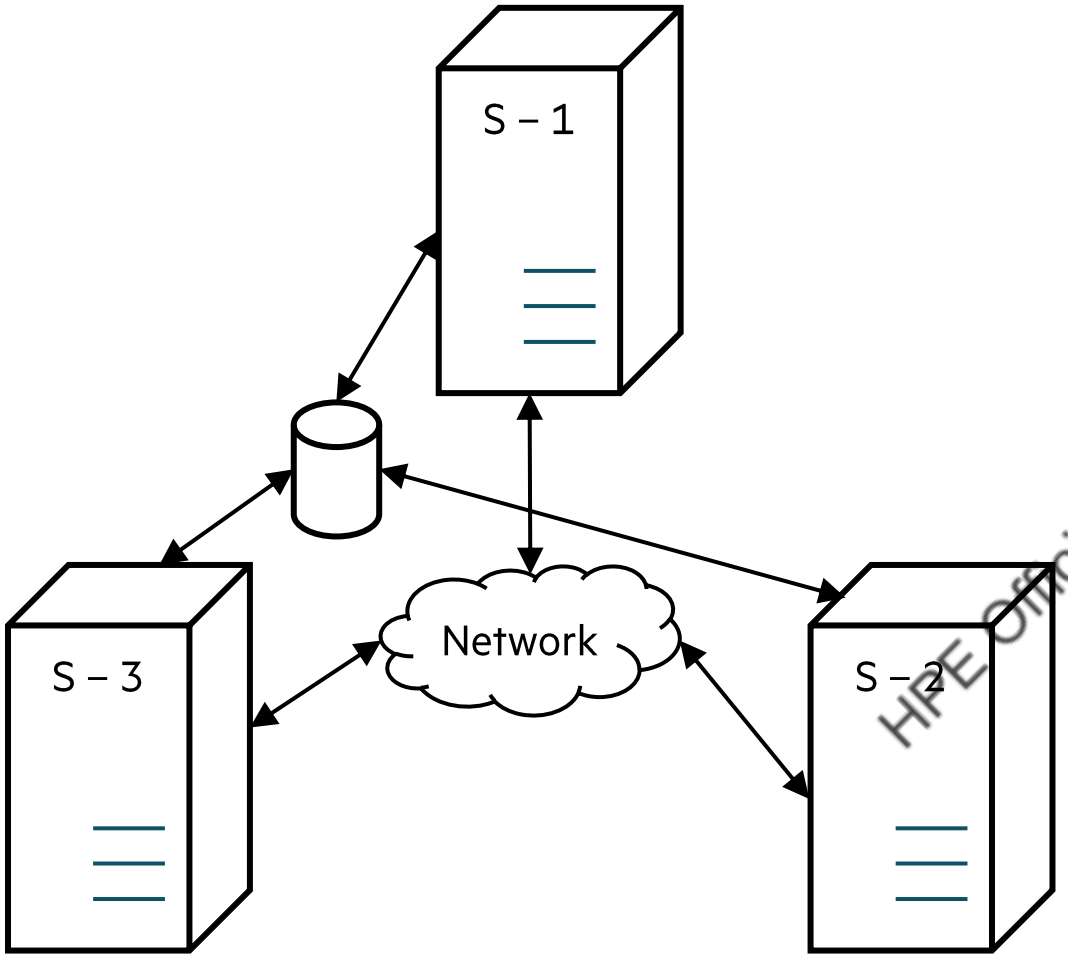


LESSON-1

ARCHITECTURE

ARCHITECTURE FOR DISTRIBUTED DEEP LEARNING

- **Model Parallelism**



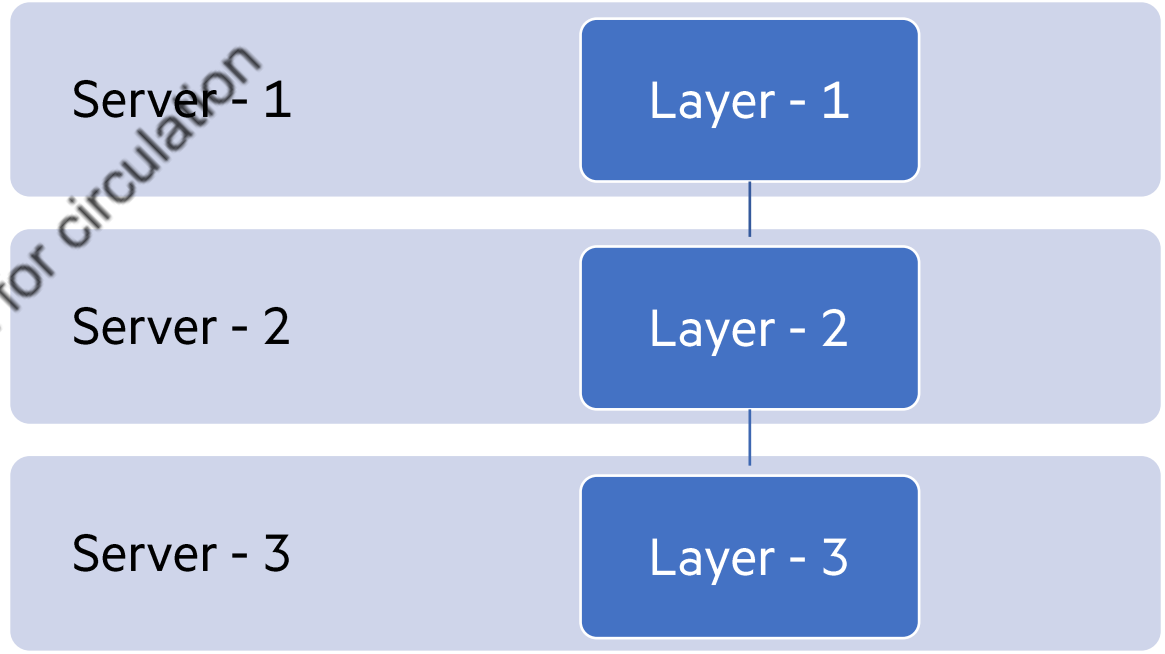


FIG.1 Model Parallelism

ARCHITECTURE FOR DISTRIBUTED DEEP LEARNING

- **Model Parallelism**

- Each machine can be treated as a sever comprising multiple CPUs and GPUs.
- The deep learning model to be trained is split and distributed across multiple machines.
- The model is segmented such that each portion of the model can run concurrently.
- Each portion of the model operates on the same data in different machines.

ARCHITECTURE FOR DISTRIBUTED DEEP LEARNING

Data Parallelism – Decentralized

- The deep learning model to be trained is replicated in each of the servers or machines.
- The training data set is divided to smaller subsets.
- Each machine or server trains the copy of the deep learning model stored in the memory with the subset of the training data.
- Trained Parameters are shared with other servers in the cluster.

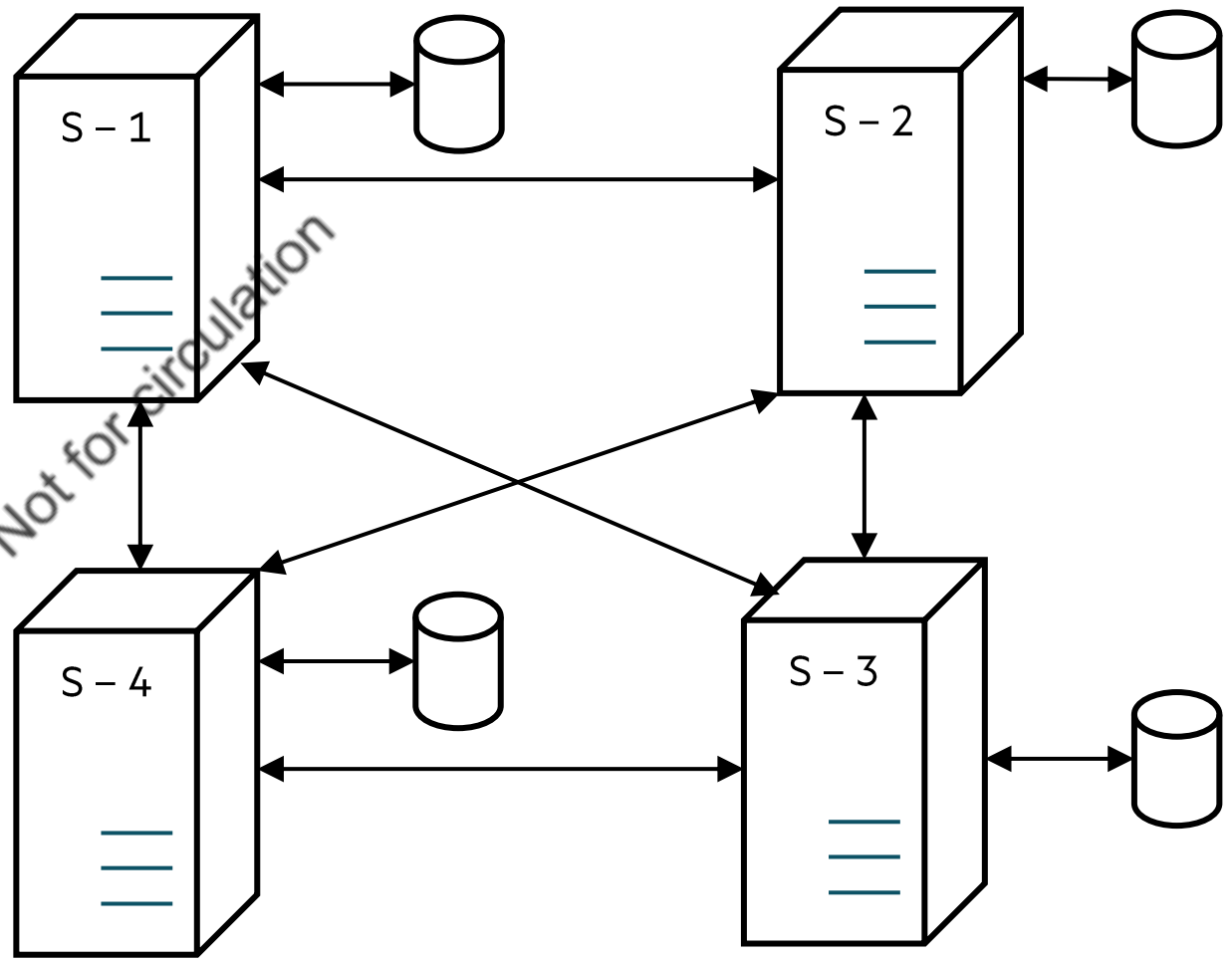


FIG. 2 Decentralized architecture in data parallelism

ARCHITECTURES FOR DISTRIBUTED DEEP LEARNING

- **Data Parallelism – Centralized**

- The deep learning model to be trained is replicated in each of the servers or machines.
- The training data set is divided to smaller subsets.
- Each machine or server trains the copy of the deep learning model stored in the memory with the subset of the training data.
- Trained parameters are shared with the parameter server (PS).
- The PS periodically shares the updated parameters with other severs.

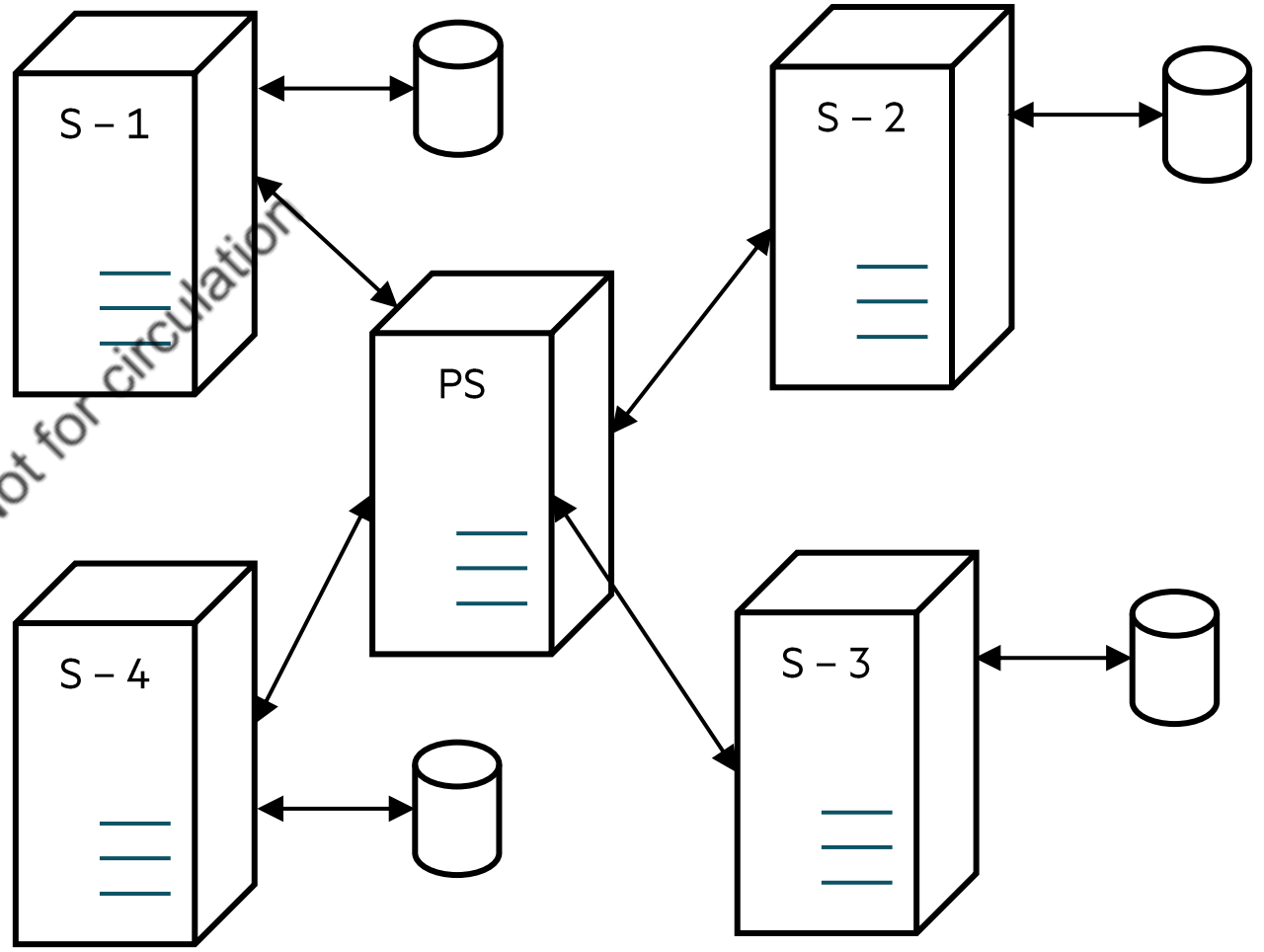
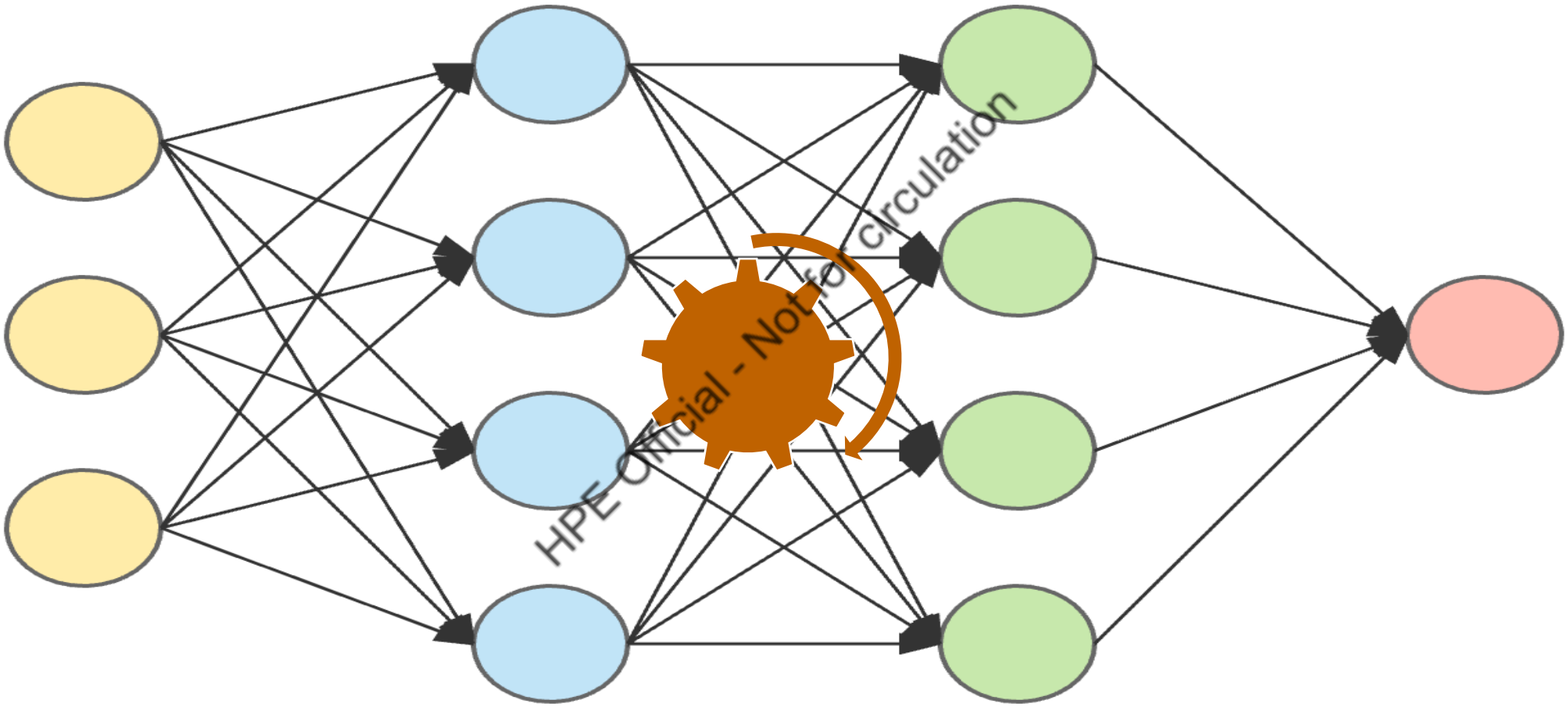


FIG. 3 Centralized architecture in data parallelism

LESSON-2

TRAINING

LESSON 2:
Training



TRAINING IN DISTRIBUTED DEEP LEARNING

- With Model Parallelism

- All the 3 machines execute a mini batch concurrently and update the trained parameters.

- Further, only the dependent parameters are shared among the machines.

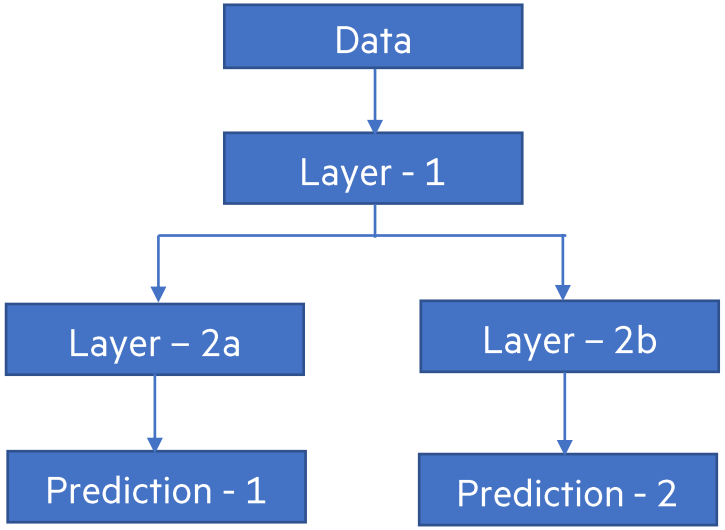


FIG. 4a

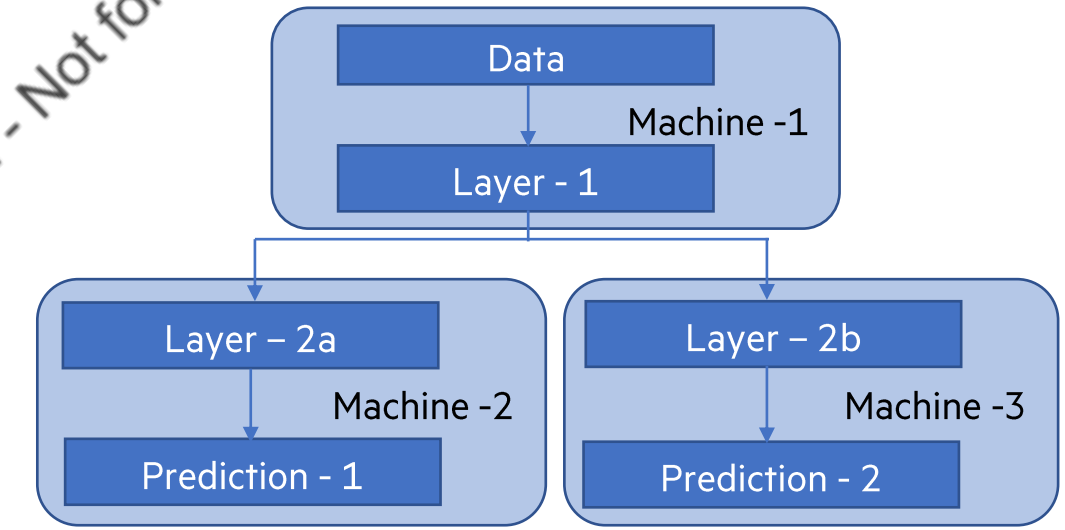


FIG.4b Example of training using model parallelism

TRAINING IN DISTRIBUTED DEEP LEARNING

- With Data Parallelism

- The deep learning model is replicated in all the 3 machines with random initializations.

- Each of the deep learning model is trained with the corresponding subsets concurrently for each mini batch.

- The trained parameters are combined and distributed to the 3 machines for training the subsequent mini batch.

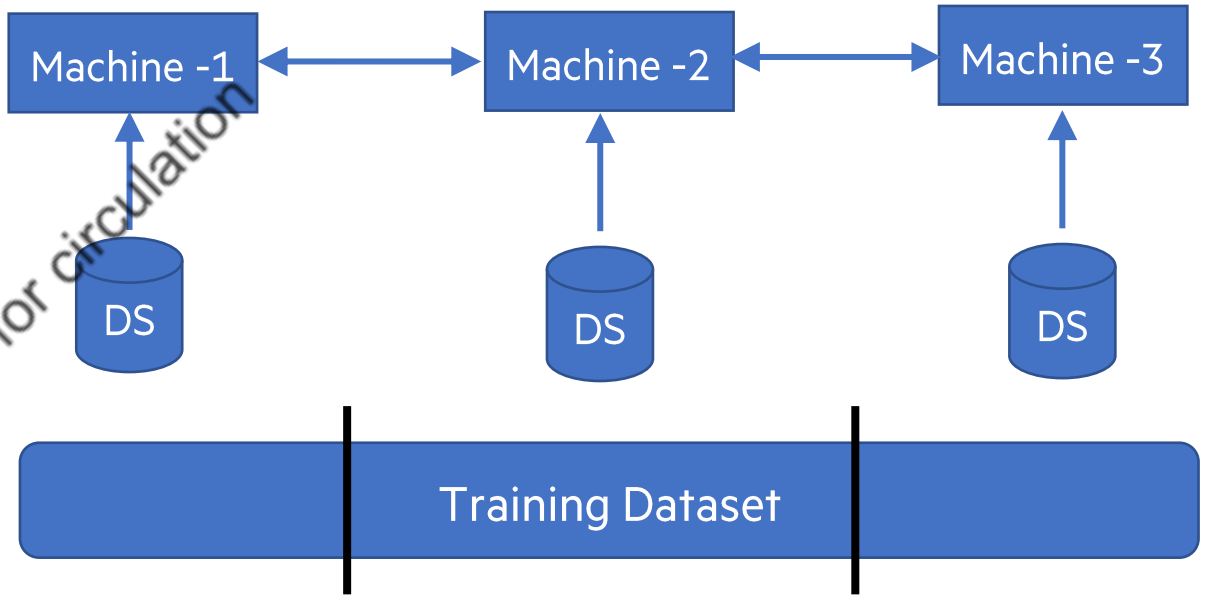
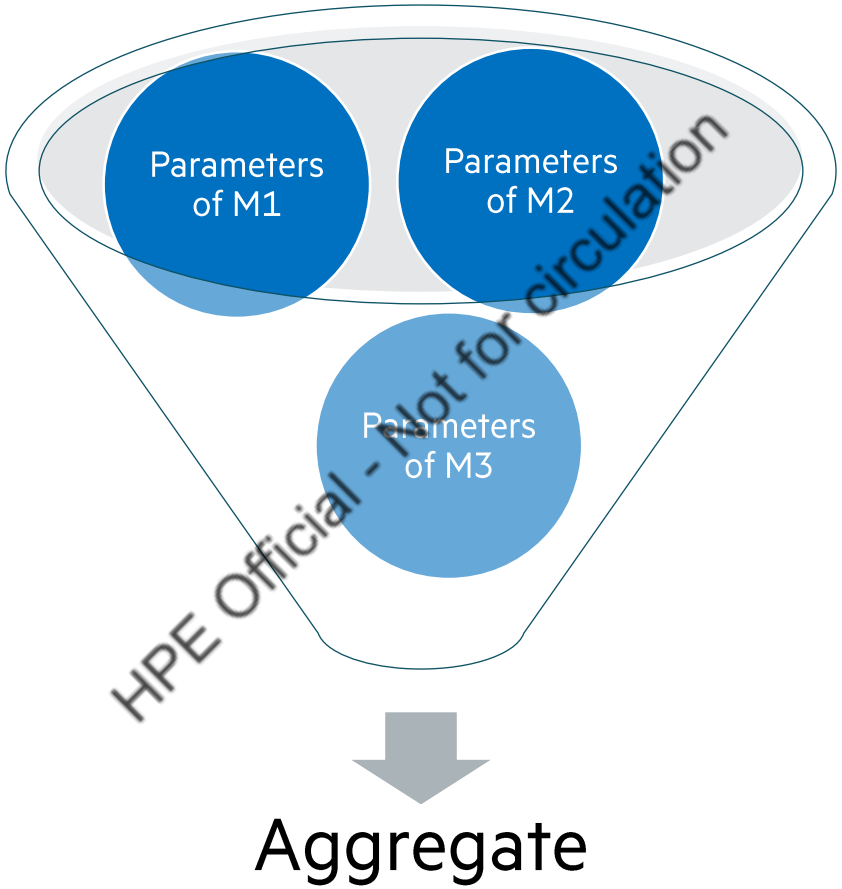


FIG. 5 Example of training using data parallelism

LESSON-3

AGGREGATION

LESSON 3:
Aggregation



AGGREGATION OF PARAMETERS IN DISTRIBUTED DEEP LEARNING

- **Synchronous Update**

- The parameter server distributes a copy of the parameters and the deep learning model to each of the machine in the cluster.
- The training dataset is divided into subsets and distributed to each of the machine in the cluster.
- The machines in the cluster, train on the subset of the data (mini batch) and share the trained parameters to the parameter server.
- The parameter server determines the average of the trained parameters obtained from all the machines in the cluster

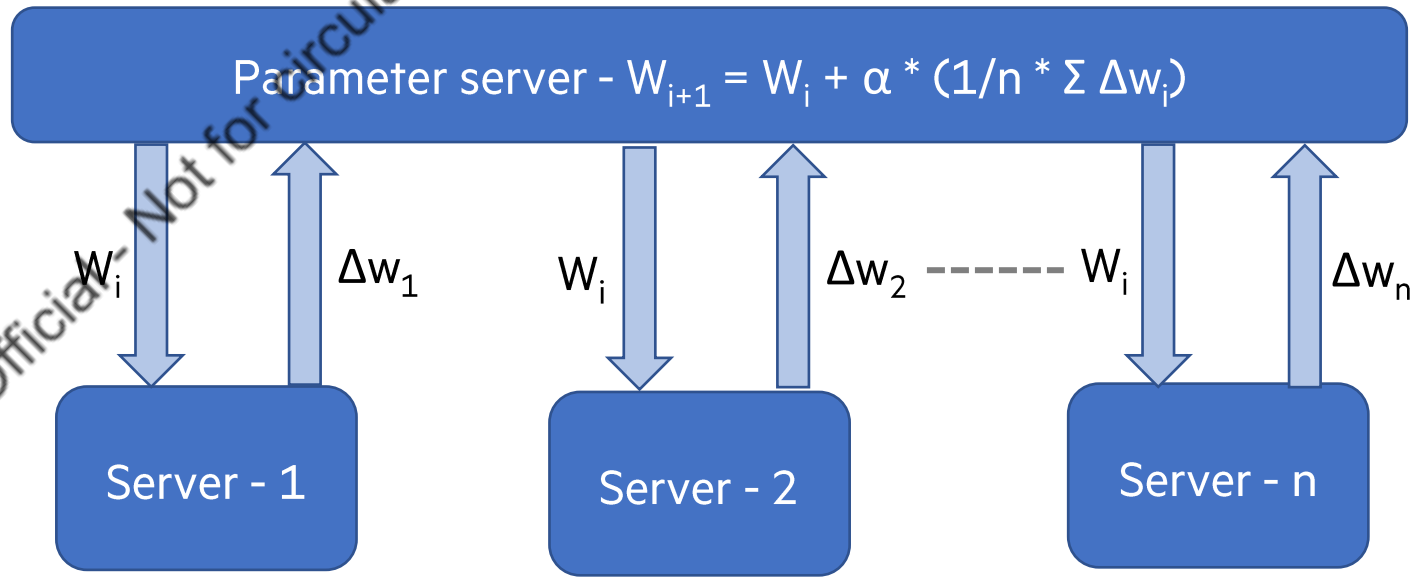


FIG. 6 Synchronous Update

AGGREGATION OF PARAMETERS IN DISTRIBUTED DEEP LEARNING

- **Asynchronous Update**

- The global set of parameters are stored in a shared memory.
- Each machine obtains a copy of the parameters from the shared memory and updates the parameters by training on a mini batch of data.
- The trained parameters aggregated to the global parameters asynchronously by each of the machines.

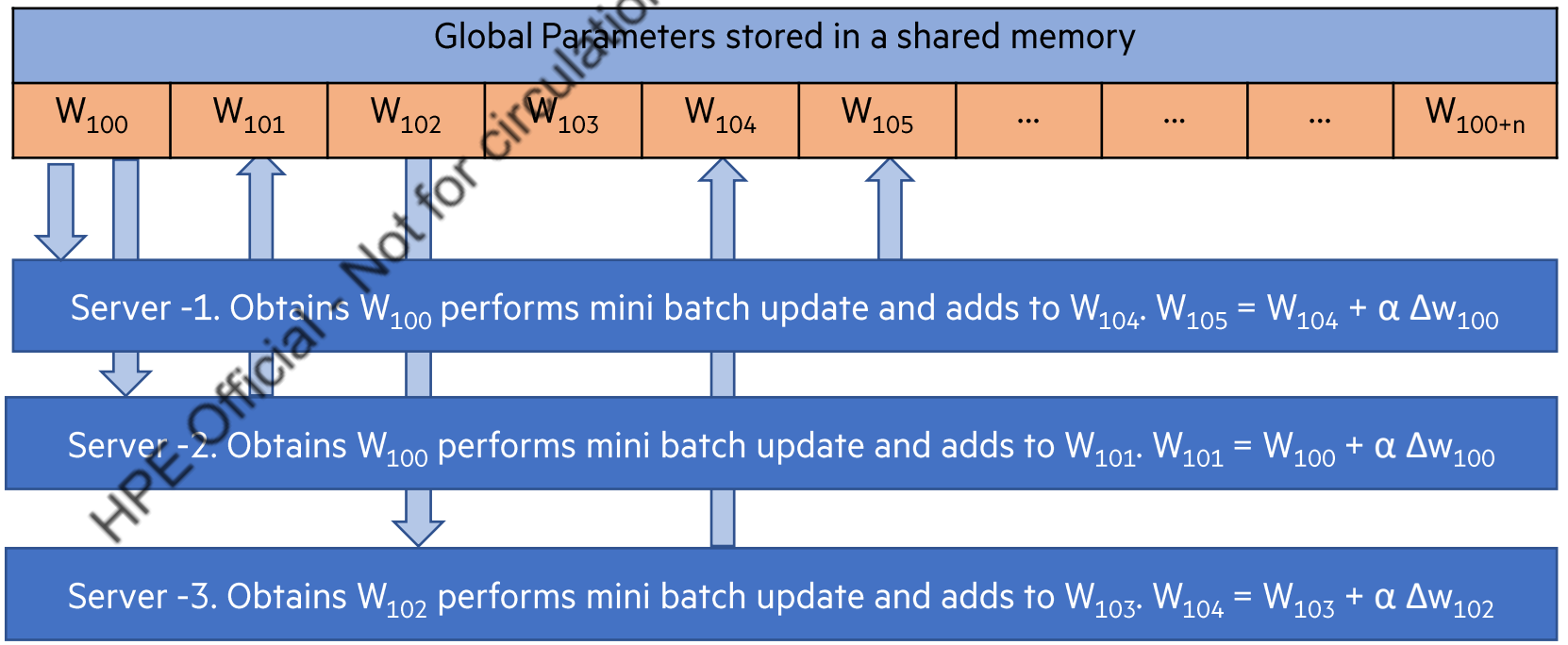
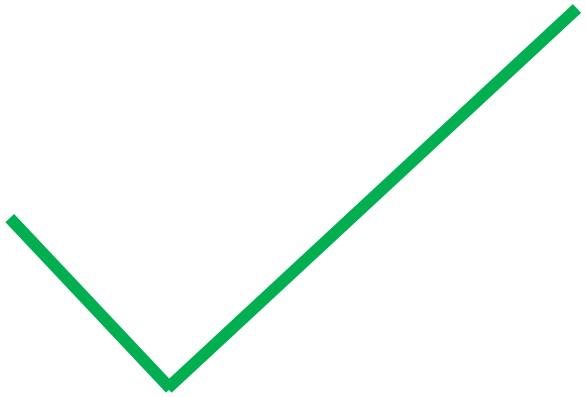


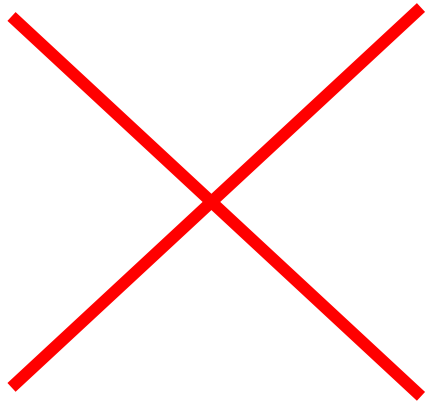
FIG. 7 Asynchronous Update

LESSON-4

EVALUATION

LESSON 4:
Evaluation





EVALUATION OF DISTRIBUTED DEEP LEARNING MODELS

- The classical evaluation metrics used to evaluate the deep learning models are unsuitable for the distributed deep learning models because of the new aspects such as data distribution, communication overhead between the machines, aggregation techniques, etc.

- Some of the evaluation criteria for distributed deep learning are given below:

a. Training Time vs Test Error
b. Training Time vs Training Error
c. Dataset size vs Training Time
d. Dataset size vs Test Error
e. Feature size vs Sample size vs Jaccard-index
f. Feature size vs Sample size vs Training time

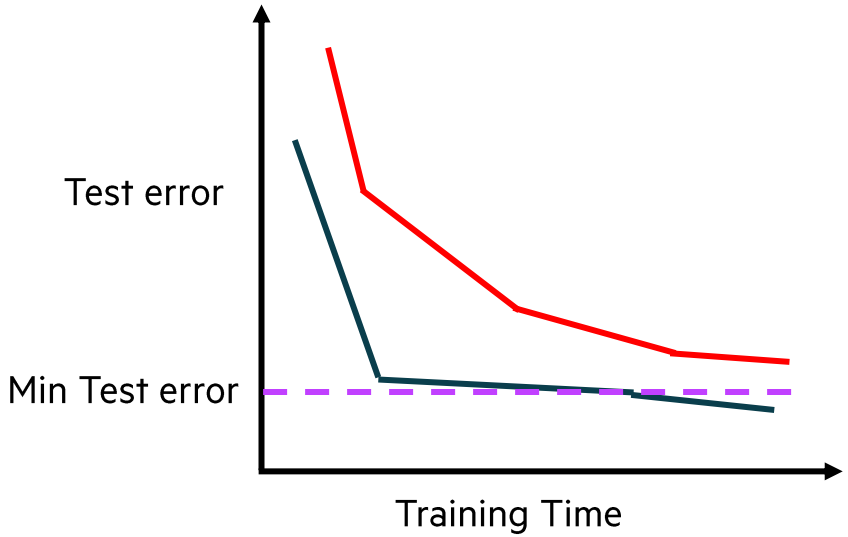
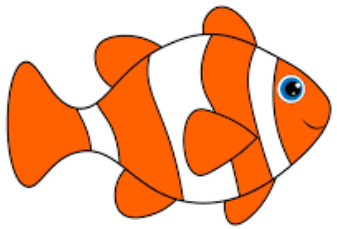


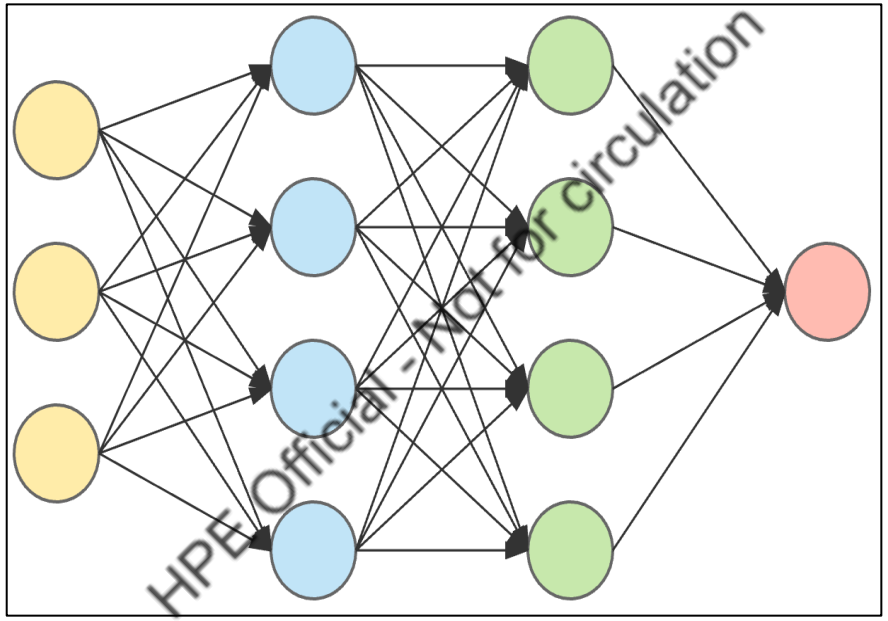
FIG. 8 Performance Evaluation

LESSON-5

INFERENCE

LESSON 5:
Inference





Fish

INFERENCING DISTRIBUTED DEEP LEARNING MODELS

- The models trained using Data parallelism can be combined during training before deploying the model into inferencing.
- Alternatively, the trained models can be combined post-training during inferencing with the help of ensemble learning approach.
- The size of the deep learning model trained using the model parallelism and/or data parallelism can be reduced using knowledge distillation techniques during inferencing.
- The trained models are used for inferencing by deploying the models as a web service.
- The REST API may be used for communication between the web service and the client applications.

LESSON-6

CONCLUSION

ADVANTAGES OF DISTRIBUTED DEEP LEARNING

- Distributed learning is scalable.
- Overcomes the problems related to centralized storage with respect to data and model.
- Enables large-scale learning with complex algorithms and memory limitations.
- Train several classifiers from distributed data sets, thereby achieving higher accuracy.
- Integration compensates the bias of the classifiers.
- Derive independent classifiers from different partitions of the data.

APPLICATIONS OF DISTRIBUTED DEEP LEARNING

- Used in identifying the best model architecture using the Sequential Model-Based Optimization (SMBO), Reinforcement Learning (RL), and Evolutionary Algorithms (EA).

- Used in searching the best hyper parameters for the model.

DEMO

- Github link

THANK YOU