Learning → Giving a machine the ability to learn, adapt, organize or repair itself

Ability to adaptive to new surroundings and solve problems → one of the distinguishing characteristics of intelligent beings

Adaptive → same task in similar circumstance more effectively + efficiently.

Broadly → Learning has two major aspects related to it
→ Skill refinement
→ Knowledge acquistion

Learning mechanism should be such that it can learn in an unknown environment as well and evolve itself.

→ Learning one of the fundamental building blocks of artificial intelligence (AI)

From mathematical point :- AI learning processes focus on processing a collection of input - output pairs for a specific function + predict the output for new input data

Knowledge - Based Learning

AI learning models can be classified into 2 major types based on their representation of knowledge.

1) Inductive learning :- AI learning model is based on inference of general rule from datasets of input - output pairs. Process of learning general patterns or rules from data points. Common algorithms Decision trees; Support Vector Machine (SVM), Neural Network + K- Nearest Neighbours (K-NN).

2) Deductive learning :-

AI learning model begins with a series of rules + infers new rules that are more efficient in the context of a specific AI algorithm.

In contrast to inductive learning (which start with specific data and generalize to create a model), deductive learning begins with established rules or knowledge + uses them to draw specific inferences.

→ Relatively less common than inductive learning

Some deductive learning machine algorithm are

1) Prolog → programming language designed for symbolic + deductive reasoning. One define set of rules + then query the system to derive specific conclusions

1c

Prolog - created in 1970.

Define problem rather than how to solve it.

Uses facts, rules + queries to describe problems + infer soln.

popular in application like expert systems, natural language processing + tasks that require logical reasoning

Not straightforward concept

sentence --> noun-phrase, verb-phrase.

noun-phrase --> determiner, noun.

verb-phrase --> verb, noun-phrase.

determiner --> [the].

noun --> [cat]; [dog]; [fish].

verb --> [eats]; [chases].

check if a sentence is valid sentence or no

?- phrase(sentence, [the, cat, eats, the, fish]).

output is true.

If there were multiple possible interpretations.

Prolog's back tracking would explore each option to find matches.

→ PySWIP → one of the most popular Prolog implementation.

→ Kanren → great for constraint-solving
→ useful for ~~creating~~ certain NLP tasks

→ MiniKaren (via logpy).

→ Clorm (Constraint logic programming)

Scanned with OKEN Scanner

c)

CLIPS

→ uses an if-then rule-based approach where rules are defined to evaluate conditions and execute specific action if conditions are met.

→ Knowledge in CLIPs is represented as facts. Rules act on these facts, allowing for complex decision-making processes.

→ Inference engine processes facts & rules to derive conclusion.

→ It allows developers to define what to do rather ~~what to~~ how to do (Declarative Paradigm)

→ suitable for complex rule based logic

Medical diagnosis → Expert systems to diagnose diseases based on symptoms & medical history.

Financial Decision → based on predefined rule.

customer support, Manufacturing

Py CLIPs

↳ help to use both rule based reasoning (CLIPS) + other AI features (Tensorflow, scikit-learn) in single application,

-2a

CLIPS(C Languag Integrated Production System)

→ forward-chaining rule-based approach.

rules are applied to data to derive specific conclusions. Developed in 1980s by NASA for building expert system

Rete algorithm → specialized algorithm used in expert system for efficient pattern matching + inference. Optimizes rule execution by storing + matching facts + rules in network structure

Cyc, MYCIN, DataLog and so on..

Python based)

Durable rules, Pyke; Experta, PyClips.

more closest.

L2

Feedback - Based Learning

Based on feedback characteristic, AI learning mod[els] can be classified as supervised, unsupervised or reinforcement learning model.

1) Unsupervised learning - Unsupervised models focus on learning a pattern in input data without any external feedback. e.g. PCA, K-means

2) Supervised learning - models use external feedback to learn function that map input data to output observation. External environment acts as a trainer of the algorithm. e.g. Linear regression, Logistic regression, Decision Tree, SVM, K-Nearest Neighbours, Linear Discriminant Analysis, RNN, CNN

3) Reinforcement learning - models use opposite dynamics such as rewards & punishment to emphasize on different types of knowledge obtained. e.g. Q-Learning; Deep Q-Network, AlphaZero.

Scanned with OKEN Scanner

3 a

Supervised Learning

Concept behind this form of learning is to create a model such that it can be supervised by a set of examples. These examples are such that the out come is known so that in future, if any unknown example whose out come is not known is ~~applie~~ supplied to the algorithm; then it can infer and predict the out come.

Two phases in supervised learning

1) Training phase
↓
model is to be learned based on training data.

2) Testing phase
↓
model created in training phase is tested against the test data. Testing phase contributes in calculating the accuracy of model.

$$\text{Accuracy measure} = \frac{\text{Number of correct classification}}{\text{Total number of test cases}}$$

Unsupervised Learning

Given a set of features, aim of unsupervised learning is to gather essential information about given data and its features. Here there is no response variable to predict and give certain value.

Evaluating the outcome in this kind of problem becomes difficult as there is no procedure to perform cross-validation on the results of independent set. Mostly statistical learning tasks can be performed using Unsupervised learning technique e.g. Amazon group identical customers with similar purchase histories + browsing patterns.

Useful for Exploring data

K-Means, PCA, ...

i-4a

Reinforcement learning

goal oriented learning. Reinforcement learning is done on the basis of information gained by different kind of interaction with the environment.

Interacts with an environment & learns to make a sequence of decisions or action to maximize a cumulative reward. Over time, through trial & error, agent improves its policy & becomes better at ~~playing~~ doing the job

Real world application

Autonomous driving, robotics, game playing & so on.

Chess, Go, dota 2 and so on.

Q-learning → popular model-free algorithm that learns Q-value for each state-action pair

Q-value represents expected future rewards of taking a certain action in a particular state

$$Q(s,a) = Q(s,a) + \alpha\left(\text{reward} + \gamma \times \max_{a'} Q(s',a') - Q(s,a)\right)$$

$\alpha \rightarrow$ learning rate & $\gamma \rightarrow$ discount factor

0.1 $\qquad$ 0.9

Classifier

We have data set. In inductive learning one tries to learn a function f. We have an hypothesis $h \rightarrow$ it is a function that approximate f.

In real, we have more than one hypothesis. So there are set of function one can have which can approximate f.

Hypothesis Space H

Input : Training set X

Output : A hypothesis $h \in H$

One can constraint h by using some bias or preferences

The hypothesis consistent with all training examples then such a hypothesis is called consistence hypothesis.

If one comes with a hypothesis h which has a low training error over a sufficiently large training set then one expect that the h will approximate the target function well on other unseen examples

So; one search ~~for~~ hypothesis space of possible representation for h ~~we~~ which best fit the data; given the bias.

The tendency to prefer one hypothesis over another is called a bias.

There are different type of bias. One classical bias is a bias called Occam's Razor.

Occam's razor state that one should prefer the simplest hypothesis over a more complex hypothesis. E.g. power plug of an appliance!

As we are using a function and trying to do a kind of generalization. This can lead to some discrepancy or errors.

The errors are of two types

→ Bias errors

→ Variance errors

Bias error → As we restrict the hypothesis space or prefer a particular hypothesis. ~~space~~. By deciding a particular hypothesis, one applies a bias. This error introduced due to the restriction on h is called "bias error". Due to incorrect or overly simplistic assumptions in learning algorithm.

Variance error → Introduced when one has small or limited test set. If one change data set and get different models. The difference between them is variance error. It represents the model's sensitivity to the specific data used for training. Model's sensitivity to small fluctuations in training set

Underfitting → When the machine learning model is too simple to capture the underlying patterns in the data. Model consistently makes prediction that are far from the actual values

→ High Bias & Low Variance.

→ High training error & high test error

5a

Overfitting → When Machine learning model is too complex & fits the training data too closely; capturing not only the underlying pattern but also the noise in the data.

→ Low Bias & high Variance

→ Low training error & high test error.

Hypothesis space

→ $\mathcal{H}$, X Training set

$h \in \mathcal{H}$ → we get h → hypothesis

Using h; we get $\hat{y}$ a predicted value of target feature on test data x.

while y is the observed value on test data x.

then

One how can define error

Absolute error = $\frac{1}{n} \sum |h(x) - y|$

Sum of square error $\frac{1}{n} \sum (h(x)-y)^2$

No. of miss classification $\frac{1}{n} \sum \delta(h(x), y)$

$\delta(h(x), y)$ → delta function

L2

$\delta(h(x), y)$ → return 1 if the classification is wrong

$\delta(h(x) y) = 0$ if correct classification.

Confusion mt we already talked in previous class.

$error_S(h)$

* Sample error $\frac{1}{n} \sum \delta(f(x), c(x))$ ← target function

~~error~~ No. of miss classification divided by n.

$error_D(h)$

* True error

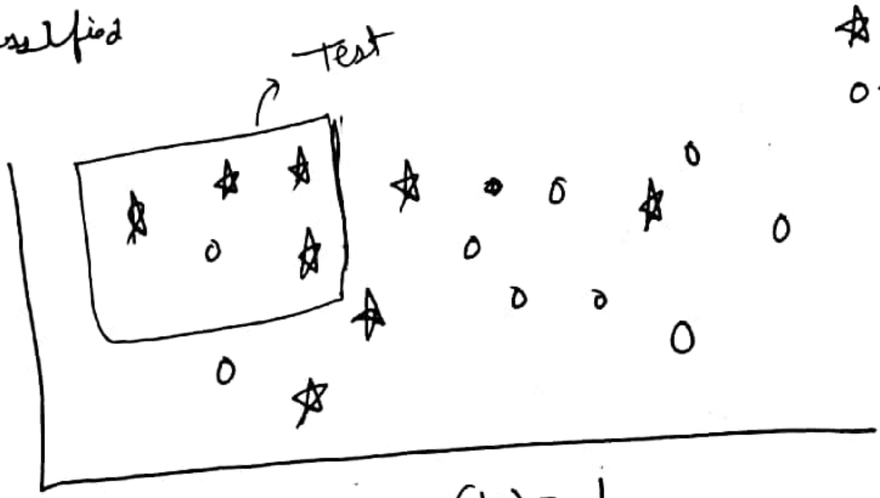
is the probability that on this distribution $f(x)$ + $C(x)$ will not agree. (on entire population)

Randomly drawn instance from entire distribution

miss classified

★ correct class

o - False classify

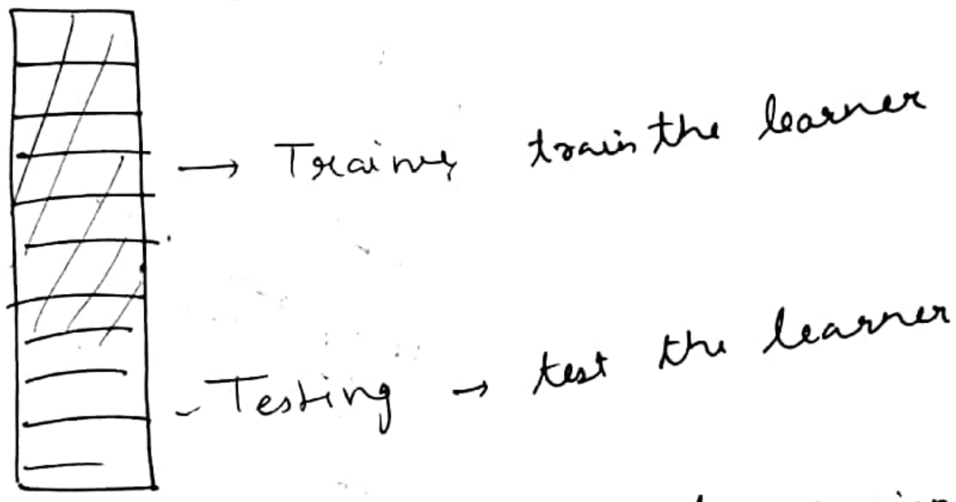


Using test, we get $error_S(h) = \frac{1}{5}$

$error_D(h) = \frac{10}{18}$

7a)

Data



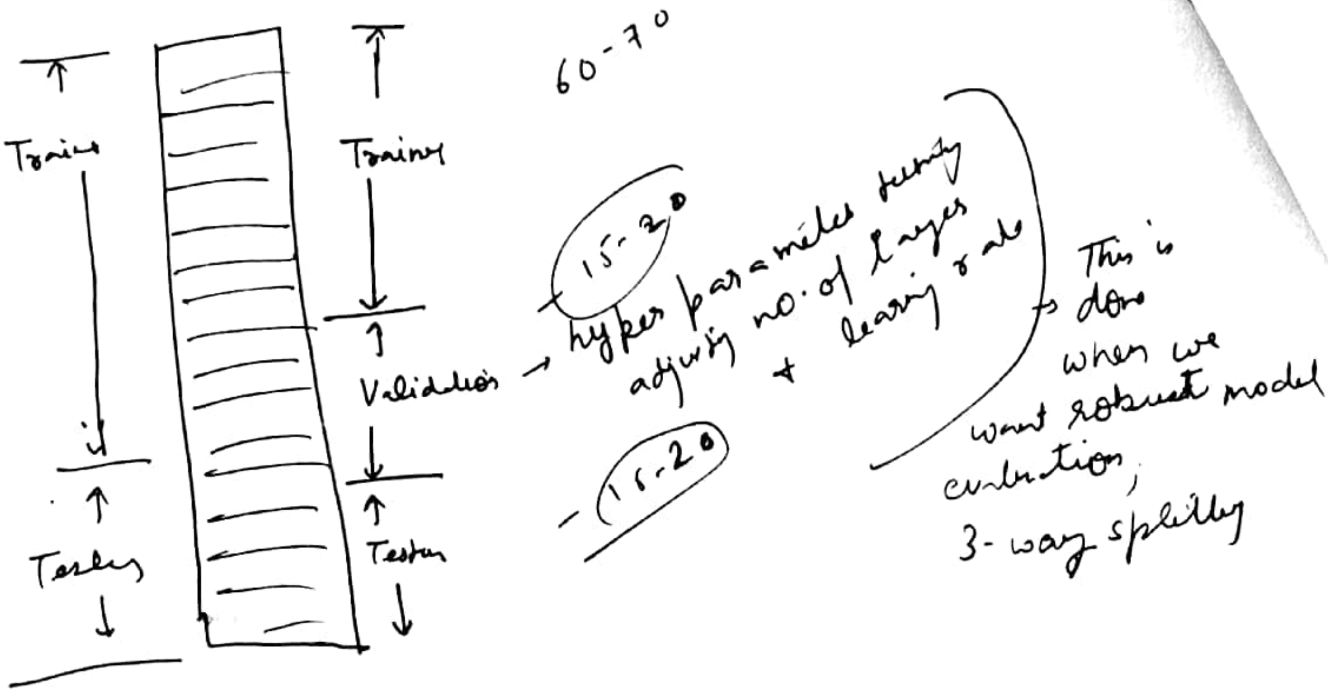
If test set is small → we will have large variance

If test set is large → variance will be smaller.

If we use all data for training we might not be able to really get a good estimate of the error as one require an independent set.

But if we get a good chunk of test set then training set is small. If training set is too small it will give rise to bias error (over fitting)

∴ ; one want to use as much of the training example as possible and also have sufficient test size.



Split the data into training set & testing set.

During training when we are tuning the model parameters, one can use some part of training set for validation.

After training is over one check the accuracy of hypothesis on the test set.

As we do validation, ~~we~~ if we are splitting data into these parts validation fails to use all the available data.

There is another scheme called cross validation

Cross validation

is a technique used in machine learning to assess the performance and generalization ability of predictive model. It involves partitioning dataset into multiple subset, training the model on some of these subsets & evaluating it on remaining data.

The most common form of cross-validation is called "k-fold cross-validation", where dataset is divided into k roughly equal sized subsets or "folds".

Steps are

1) Data splitting - dataset is divided into k subset of approximately equal size. If k = 6; then data is divided into six folds.

2) One of k fold is used as test + remaining as training

3) Evaluation metrics (Accuracy, mean squared error) are computed

c) Results from each fold is recorded; models performance is averaged across all folds.

- Provide better estimate of the model's performance on unseen data.
- Helps in identifying potential issues like underfitting or overfitting.
- More efficient use of data

Common choice of value are 5-fold & 10-fold cross-validation.

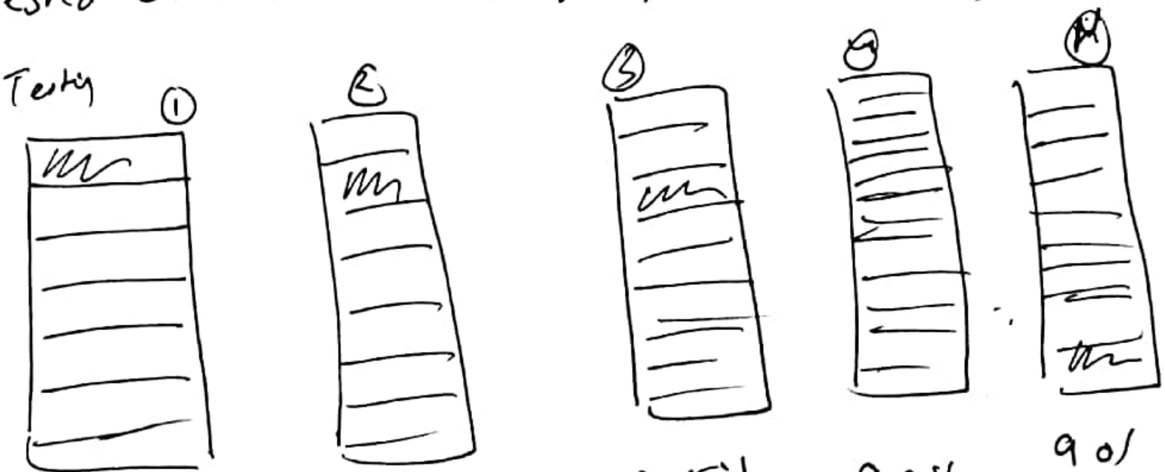
There are other variation

→ Stratified k-fold.

→ leave-one-out cross-validation (k equals no. of data point)

Nested cross-validation (hyper parameter tuning)

Ex Testing



Accuracy 91% 92% 95% 92% 90%

Find Accuracy = ~~Average (...)~~ Average(1 attempt, 2 attempt, ... N attempt)

→ Stratified K-fold cross validation

→ maintains the proportions of classes (for classification tasks) in each fold, ensuring each fold is representative of whole dataset.

→ Useful for imbalanced datasets

→ Leave-one-out cross validation (LOOCV)

→ dataset divided so that each data-point form a single fold.

Model is trained on all but one data point & validated on single data point.

Useful for small datasets but computationally expensive for large datasets.

→ Leave P-out cross validation

Similar to LOOCV, but instead of leaving out a single point, $\underline{\underline{p}}$ data points are left out in each iteration.

Also computationally expensive as there are $\frac{n!}{p!(n-p)!}$ possible ways to select p data points from n.

→ Time Series cross validation

Specially designed for time series data.

where data points depend on previous points in time. Instead of randomly shuffling & splitting data, time series cross validation uses rolling validation window

Train model on first portion of data + validate it on next portion.

Keep increasing training data + moving forward in time