Title: Training, validation, and test data sets
URL: https://en.wikipedia.org/wiki/Training,_validation,_and_test_data_sets
PageID: 1514392
Categories: Category:Datasets in machine learning, Category:Validity (statistics)


---

Supervised learning
Unsupervised learning
Semi-supervised learning
Self-supervised learning
Reinforcement learning
Meta-learning
Online learning
Batch learning
Curriculum learning
Rule-based learning
Neuro-symbolic AI
Neuromorphic engineering
Quantum machine learning
Classification
Generative modeling
Regression
Clustering
Dimensionality reduction
Density estimation
Anomaly detection
Data cleaning
AutoML
Association rules
Semantic analysis
Structured prediction
Feature engineering
Feature learning
Learning to rank
Grammar induction
Ontology learning
Multimodal learning

Apprenticeship learning

Decision trees

Ensembles Bagging Boosting Random forest

Bagging

Boosting

Random forest

k -NN

Linear regression

Naive Bayes

Artificial neural networks

Logistic regression

Perceptron

Relevance vector machine (RVM)

Support vector machine (SVM)

BIRCH

CURE

Hierarchical

k -means

Fuzzy

Expectation–maximization (EM)

DBSCAN

OPTICS

Mean shift

Factor analysis

CCA

ICA

LDA

NMF

PCA

PGD

t-SNE

SDL

Graphical models Bayes net Conditional random field Hidden Markov

Bayes net

Conditional random field

Hidden Markov

RANSAC

k -NN

Local outlier factor

Isolation forest

Autoencoder

Deep learning

Feedforward neural network

Recurrent neural network LSTM GRU ESN reservoir computing

LSTM

GRU

ESN

reservoir computing

Boltzmann machine Restricted

Restricted

GAN

Diffusion model

SOM

Convolutional neural network U-Net LeNet AlexNet DeepDream

U-Net

LeNet

AlexNet

DeepDream

Neural field Neural radiance field Physics-informed neural networks

Neural radiance field

Physics-informed neural networks

Transformer Vision

Vision

Mamba

Spiking neural network

Memtransistor

Electrochemical RAM (ECRAM)

Q-learning

Policy gradient

SARSA

Temporal difference (TD)

Multi-agent Self-play

Self-play

Active learning

Crowdsourcing

Human-in-the-loop

Mechanistic interpretability

RLHF

Coefficient of determination

Confusion matrix

Learning curve

ROC curve

Kernel machines

Bias–variance tradeoff

Computational learning theory

Empirical risk minimization

Occam learning

PAC learning

Statistical learning

VC theory

Topological deep learning

AAAI

ECML PKDD

NeurIPS

ICML

ICLR

IJCAI

ML

JMLR

Glossary of artificial intelligence

List of datasets for machine-learning research List of datasets in computer vision and image processing

List of datasets in computer vision and image processing

Outline of machine learning

v

t

e

In machine learning , a common task is the study and construction of algorithms that can learn from and make predictions on data . [ 1 ] Such algorithms function by making data-driven predictions or decisions, [ 2 ] through building a mathematical model from input data. These input data used to build the model are usually divided into multiple data sets . In particular, three data sets are commonly used in different stages of the creation of the model: training, validation, and test sets.

The model is initially fit on a training data set , [ 3 ] which is a set of examples used to fit the parameters (e.g. weights of connections between neurons in artificial neural networks ) of the model. [ 4 ] The model (e.g. a naive Bayes classifier ) is trained on the training data set using a supervised learning method, for example using optimization methods such as gradient descent or stochastic gradient descent . In practice, the training data set often consists of pairs of an input

vector (or scalar) and the corresponding output vector (or scalar), where the answer key is commonly denoted as the target (or label ). The current model is run with the training data set and produces a result, which is then compared with the target , for each input vector in the training data set. Based on the result of the comparison and the specific learning algorithm being used, the parameters of the model are adjusted. The model fitting can include both variable selection and parameter estimation .

Successively, the fitted model is used to predict the responses for the observations in a second data set called the validation data set . [ 3 ] The validation data set provides an unbiased evaluation of a model fit on the training data set while tuning the model's hyperparameters [ 5 ] (e.g. the number of hidden units—layers and layer widths—in a neural network [ 4 ] ). Validation data sets can be used for regularization by early stopping (stopping training when the error on the validation data set increases, as this is a sign of over-fitting to the training data set). [ 6 ] This simple procedure is complicated in practice by the fact that the validation data set's error may fluctuate during training, producing multiple local minima. This complication has led to the creation of many ad-hoc rules for deciding when over-fitting has truly begun. [ 6 ]

Finally, the test data set is a data set used to provide an unbiased evaluation of a final model fit on the training data set. [ 5 ] If the data in the test data set has never been used in training (for example in cross-validation ), the test data set is also called a holdout data set . The term "validation set" is sometimes used instead of "test set" in some literature (e.g., if the original data set was partitioned into only two subsets, the test set might be referred to as the validation set). [ 5 ]

Deciding the sizes and strategies for data set division in training, test and validation sets is very dependent on the problem and data available. [ 7 ]

Training data set

A training data set is a data set of examples used during the learning process and is used to fit the parameters (e.g., weights) of, for example, a classifier . [ 9 ] [ 10 ]

For classification tasks, a supervised learning algorithm looks at the training data set to determine, or learn, the optimal combinations of variables that will generate a good predictive model . [ 11 ] The goal is to produce a trained (fitted) model that generalizes well to new, unknown data. [ 12 ] The fitted model is evaluated using “new” examples from the held-out data sets (validation and test data sets) to estimate the model’s accuracy in classifying new data. [ 5 ] To reduce the risk of issues such as over-fitting, the examples in the validation and test data sets should not be used to train the model. [ 5 ]

Most approaches that search through training data for empirical relationships tend to overfit the data, meaning that they can identify and exploit apparent relationships in the training data that do not hold in general.

When a training set is continuously expanded with new data, then this is incremental learning .

Validation data set

A validation data set is a data set of examples used to tune the hyperparameters (i.e. the architecture) of a model. It is sometimes also called the development set or the "dev set". [ 13 ] An example of a hyperparameter for artificial neural networks includes the number of hidden units in each layer. [ 9 ] [ 10 ] It, as well as the testing set (as mentioned below), should follow the same probability distribution as the training data set.

In order to avoid overfitting, when any classification parameter needs to be adjusted, it is necessary to have a validation data set in addition to the training and test data sets. For example, if the most suitable classifier for the problem is sought, the training data set is used to train the different candidate classifiers, the validation data set is used to compare their performances and decide which one to take and, finally, the test data set is used to obtain the performance characteristics such as accuracy , sensitivity , specificity , F-measure , and so on. The validation data set functions as a hybrid: it is training data used for testing, but neither as part of the low-level training nor as part of the final testing.

The basic process of using a validation data set for model selection (as part of training data set, validation data set, and test data set) is: [ 10 ] [ 14 ]

Since our goal is to find the network having the best performance on new data, the simplest approach to the comparison of different networks is to evaluate the error function using data which is independent of that used for training. Various networks are trained by minimization of an appropriate error function defined with respect to a training data set. The performance of the networks is then compared by evaluating the error function using an independent validation set, and the network having the smallest error with respect to the validation set is selected. This approach is called the hold out method. Since this procedure can itself lead to some overfitting to the validation set, the performance of the selected network should be confirmed by measuring its performance on a third independent set of data called a test set.

An application of this process is in early stopping , where the candidate models are successive iterations of the same network, and training stops when the error on the validation set grows, choosing the previous model (the one with minimum error).

Test data set

A test data set is a data set that is independent of the training data set, but that follows the same probability distribution as the training data set. If a model fit to the training data set also fits the test data set well, minimal overfitting has taken place (see figure below). A better fitting of the training data set as opposed to the test data set usually points to over-fitting.

A test set is therefore a set of examples used only to assess the performance (i.e. generalization) of a fully specified classifier. [ 9 ] [ 10 ] To do this, the final model is used to predict classifications of examples in the test set. Those predictions are compared to the examples' true classifications to assess the model's accuracy. [ 11 ]

In a scenario where both validation and test data sets are used, the test data set is typically used to assess the final model that is selected during the validation process. In the case where the original data set is partitioned into two subsets (training and test data sets), the test data set might assess the model only once (e.g., in the holdout method ). [ 15 ] Note that some sources advise against such a method. [ 12 ] However, when using a method such as cross-validation , two partitions can be sufficient and effective since results are averaged after repeated rounds of model training and testing to help reduce bias and variability. [ 5 ] [ 12 ]

Confusion in terminology

Testing is trying something to find out about it ("To put to the proof; to prove the truth, genuineness, or quality of by experiment" according to the Collaborative International Dictionary of English) and to validate is to prove that something is valid ("To confirm; to render valid" Collaborative International Dictionary of English). With this perspective, the most common use of the terms test set and validation set is the one here described. However, in both industry and academia, they are sometimes used interchanged, by considering that the internal process is testing different models to improve (test set as a development set) and the final model is the one that needs to be validated before real use with an unseen data (validation set). "The literature on machine learning often reverses the meaning of 'validation' and 'test' sets. This is the most blatant example of the terminological confusion that pervades artificial intelligence research." [ 16 ] Nevertheless, the important concept that must be kept is that the final set, whether called test or validation, should only be used in the final experiment.

Cross-validation

In order to get more stable results and use all valuable data for training, a data set can be repeatedly split into several training and a validation data sets. This is known as cross-validation . To confirm the model's performance, an additional test data set held out from cross-validation is normally used.

It is possible to use cross-validation on training and validation sets, and within each training set have further cross-validation for a test set for hyperparameter tuning. This is known as nested cross-validation .

Causes of error

Omissions in the training of algorithms are a major cause of erroneous outputs. [ 17 ] Types of such omissions include: [ 17 ]

Particular circumstances or variations were not included.

Obsolete data

Ambiguous input information

Inability to change to new environments

Inability to request help from a human or another AI system when needed

An example of an omission of particular circumstances is a case where a boy was able to unlock the phone because his mother registered her face under indoor, nighttime lighting, a condition which was not appropriately included in the training of the system. [ 17 ] [ 18 ]

Usage of relatively irrelevant input can include situations where algorithms use the background rather than the object of interest for object detection , such as being trained by pictures of sheep on grasslands, leading to a risk that a different object will be interpreted as a sheep if located on a grassland. [ 17 ]

See also

Statistical classification

List of datasets for machine learning research

Hierarchical classification

References

v

t

e

History timeline

timeline

Companies

Projects

Parameter Hyperparameter

Hyperparameter

Loss functions

Regression Bias–variance tradeoff Double descent Overfitting

Bias–variance tradeoff

Double descent

Overfitting

Clustering

Gradient descent SGD Quasi-Newton method Conjugate gradient method

SGD

Quasi-Newton method

Conjugate gradient method

Backpropagation

Attention

Convolution

Normalization Batchnorm

Batchnorm

Activation Softmax Sigmoid Rectifier

Softmax

Sigmoid

Rectifier

Gating

Weight initialization

Regularization

Datasets Augmentation

Augmentation

Prompt engineering

Reinforcement learning Q-learning SARSA Imitation Policy gradient

Q-learning

SARSA

Imitation

Policy gradient

Diffusion

Latent diffusion model

Autoregression

Adversary

RAG

Uncanny valley

RLHF

Self-supervised learning

Reflection

Recursive self-improvement

Hallucination

Word embedding

Vibe coding

Machine learning In-context learning

In-context learning

Artificial neural network Deep learning

Deep learning

Language model Large language model NMT

Large language model

NMT

Reasoning language model

Model Context Protocol

Intelligent agent

Artificial human companion

Humanity's Last Exam

Artificial general intelligence (AGI)

AlexNet

WaveNet

Human image synthesis

HWR

OCR

Computer vision

Speech synthesis 15.ai ElevenLabs

15.ai

ElevenLabs

Speech recognition Whisper

Whisper

Facial recognition

AlphaFold

Text-to-image models Aurora DALL-E Firefly Flux Ideogram Imagen Midjourney Recraft Stable Diffusion

Aurora

DALL-E

Firefly

Flux

Ideogram

Imagen

Midjourney

Recraft

Stable Diffusion

Text-to-video models Dream Machine Runway Gen Hailuo AI Kling Sora Veo

Dream Machine

Runway Gen

Hailuo AI

Kling

Sora

Veo

Music generation Riffusion Suno AI Udio

Riffusion

Suno AI

Udio

Word2vec

Seq2seq

GloVe

BERT

T5

Llama

Chinchilla AI

PaLM

GPT 1 2 3 J ChatGPT 4 4o o1 o3 4.5 4.1 o4-mini 5

1

2

3

J

ChatGPT

4

4o

o1

o3

4.5

4.1

o4-mini

5

Claude

Gemini Gemini (language model) Gemma

Gemini (language model)

Gemma

Grok

LaMDA

BLOOM

DBRX

Project Debater

IBM Watson

IBM Watsonx

Granite

PanGu-Σ

DeepSeek

Qwen

AlphaGo

AlphaZero

OpenAI Five

Self-driving car

MuZero

Action selection AutoGPT

AutoGPT

Robot control

Alan Turing

Warren Sturgis McCulloch

Walter Pitts

John von Neumann

Claude Shannon

Shun'ichi Amari

Kunihiko Fukushima

Takeo Kanade

Marvin Minsky

John McCarthy

Nathaniel Rochester

Allen Newell

Cliff Shaw

Herbert A. Simon

Oliver Selfridge

Frank Rosenblatt

Bernard Widrow

Joseph Weizenbaum

Seymour Papert

Seppo Linnainmaa

Paul Werbos

Geoffrey Hinton

John Hopfield

Jürgen Schmidhuber

Yann LeCun

Yoshua Bengio

Lotfi A. Zadeh

Stephen Grossberg
Alex Graves
James Goodnight
Andrew Ng
Fei-Fei Li
Alex Krizhevsky
Ilya Sutskever
Oriol Vinyals
Quoc V. Le
Ian Goodfellow
Demis Hassabis
David Silver
Andrej Karpathy
Ashish Vaswani
Noam Shazeer
Aidan Gomez
John Schulman
Mustafa Suleyman
Jan Leike
Daniel Kokotajlo
François Chollet
Neural Turing machine
Differentiable neural computer
Transformer Vision transformer (ViT)
Vision transformer (ViT)
Recurrent neural network (RNN)
Long short-term memory (LSTM)
Gated recurrent unit (GRU)
Echo state network
Multilayer perceptron (MLP)
Convolutional neural network (CNN)
Residual neural network (RNN)
Highway network
Mamba
Autoencoder
Variational autoencoder (VAE)
Generative adversarial network (GAN)
Graph neural network (GNN)

Category