Title: Bias–variance tradeoff

URL: https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

PageID: 40678189

Categories: Category:Dilemmas, Category:Machine learning, Category:Model selection, Category:Statistical classification

---

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 statistics and machine learning , the bias–variance tradeoff describes the relationship between a model's complexity, the accuracy of its predictions, and how well it can make predictions on previously unseen data that were not used to train the model. In general, as the number of tunable parameters in a model increase, it becomes more flexible, and can better fit a training data set. That is, the model has lower error or lower bias . However, for more flexible models, there will tend to be greater variance to the model fit each time we take a set of samples to create a new training data set. It is said that there is greater variance in the model's estimated parameters .

The bias–variance dilemma or bias–variance problem is the conflict in trying to simultaneously minimize these two sources of error that prevent supervised learning algorithms from generalizing beyond their training set :

The bias error is an error from erroneous assumptions in the learning algorithm . High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

The variance is an error from sensitivity to small fluctuations in the training set. High variance may result from an algorithm modeling the random noise in the training data (overfitting).

The bias–variance decomposition is a way of analyzing a learning algorithm's expected generalization error with respect to a particular problem as a sum of three terms, the bias, variance, and a quantity called the irreducible error , resulting from noise in the problem itself.

Motivation

High bias, low variance

High bias, high variance

Low bias, low variance

Low bias, high variance

The bias–variance tradeoff is a central problem in supervised learning. Ideally, one wants to choose a model that both accurately captures the regularities in its training data, but also generalizes well to unseen data. Unfortunately, it is typically impossible to do both simultaneously. High-variance learning methods may be able to represent their training set well but are at risk of overfitting to noisy or unrepresentative training data. In contrast, algorithms with high bias typically produce simpler models that may fail to capture important regularities (i.e. underfit) in the data.

It is an often made fallacy to assume that complex models must have high variance. High variance models are "complex" in some sense, but the reverse needs not be true. In addition, one has to be careful how to define complexity. In particular, the number of parameters used to describe the model is a poor measure of complexity. This is illustrated by an example adapted from: The model f a , b (x) = a sin ■ (b x) {\displaystyle f_{a,b}(x)=a\sin(bx)} has only two parameters (a , b {\displaystyle a,b}) but it can interpolate any number of points by oscillating with a high enough frequency, resulting in both a high bias and high variance.

An analogy can be made to the relationship between accuracy and precision . Accuracy is one way of quantifying bias and can intuitively be improved by selecting from only local information. Consequently, a sample will appear accurate (i.e. have low bias) under the aforementioned selection conditions, but may result in underfitting. In other words, test data may not agree as closely with training data, which would indicate imprecision and therefore inflated variance. A graphical example would be a straight line fit to data exhibiting quadratic behavior overall. Precision is a description of variance and generally can only be improved by selecting information from a comparatively larger space. The option to select many data points over a broad sample space is the ideal condition for any analysis. However, intrinsic constraints (whether physical, theoretical, computational, etc.) will always play a limiting role. The limiting case where only a finite number of data points are selected over a broad sample space may result in improved precision and lower variance overall, but may also result in an overreliance on the training data (overfitting). This means that test data would also not agree as closely with the training data, but in this case the reason is inaccuracy or high bias. To borrow from the previous example, the graphical representation would appear as a high-order polynomial fit to the same data exhibiting quadratic behavior. Note that error in each case is measured the same way, but the reason ascribed to the error is different depending on the balance between bias and variance. To mitigate how much information is used from neighboring observations, a model can be smoothed via explicit regularization , such as shrinkage .

Bias–variance decomposition of mean squared error

Suppose that we have a training set consisting of a set of points x 1 , … , x n {\displaystyle x_{1},\dots ,x_{n}} and real-valued labels y i {\displaystyle y_{i}} associated with the points x i {\displaystyle x_{i}} . We assume that the data is generated by a function f (x) {\displaystyle f(x)} such as y = f (x) + ε {\displaystyle y=f(x)+\varepsilon } , where the noise, ε {\displaystyle \varepsilon } , has zero mean and variance σ 2 {\displaystyle \sigma ^{2}} . That is, y i = f (x i) + ε i

{\displaystyle y_{i}=f(x_{i})+\varepsilon _{i}} , where ε i {\displaystyle \varepsilon _{i}} is a noise sample.

We want to find a function f ^ (x ; D) {\displaystyle {\hat {f}}(x;D)} , that approximates the true function f (x) {\displaystyle f(x)} as well as possible, by means of some learning algorithm based on a training dataset (sample) D = { (x 1 , y 1) … , (x n , y n) } {\displaystyle D=\{(x_{1},y_{1})\dots ,(x_{n},y_{n})\}} . We make "as well as possible" precise by measuring the mean squared error between y {\displaystyle y} and f ^ (x ; D) {\displaystyle {\hat {f}}(x;D)} : we want (y – f ^ (x ; D)) 2 {\displaystyle (y-{\hat {f}}(x;D))^{2}} to be minimal, both for x 1 , … , x n {\displaystyle x_{1},\dots ,x_{n}} and for points outside of our sample . Of course, we cannot hope to do so perfectly, since the y i {\displaystyle y_{i}} contain noise ε {\displaystyle \varepsilon } ; this means we must be prepared to accept an irreducible error in any function we come up with.

Finding an f ^ {\displaystyle {\hat {f}}} that generalizes to points outside of the training set can be done with any of the countless algorithms used for supervised learning. It turns out that whichever function f ^ {\displaystyle {\hat {f}}} we select, we can decompose its expected error on an unseen sample x {\displaystyle x} (i.e. conditional to x) as follows:

E D , ε [(y – f ^ (x ; D)) 2] = (Bias D ■ [f ^ (x ; D)]) 2 + Var D ■ [f ^ (x ; D)] + σ 2 {\displaystyle \mathbb {E} _{D,\varepsilon }{\Big [}{\big (}y-{\hat {f}}(x;D){\big)}^{2}{\Big]}={\Big (}\operatorname {Bias} _{D}{\big [}{\hat {f}}(x;D){\big]}{\Big)}^{2}+\operatorname {Var} _{D}{\big [}{\hat {f}}(x;D){\big]}+\sigma ^{2}}

where Bias D ■ [f ^ (x ; D)] ■ E D [f ^ (x ; D) – f (x)] = E D [f ^ (x ; D)] – f (x) = E D [f ^ (x ; D)] – E y | x [y (x)] {\displaystyle {\begin{aligned}\operatorname {Bias} _{D}{\big [}{\hat {f}}(x;D){\big]}&\triangleq \mathbb {E} _{D}{\big [}{\hat {f}}(x;D)-f(x){\big]}\\&=\mathbb {E} _{D}{\big [}{\hat {f}}(x;D){\big]}\,-\,f(x)\\&=\mathbb {E} _{D}{\big [}{\hat {f}}(x;D){\big]}\,-\,\mathbb {E} _{y|x}{\big [}y(x){\big]}\end{aligned}}}

and

Var D ■ [f ^ (x ; D)] ■ E D [(E D [f ^ (x ; D)] – f ^ (x ; D)) 2] {\displaystyle \operatorname {Var} _{D}{\big [}{\hat {f}}(x;D){\big]}\triangleq \mathbb {E} _{D}\left[\left(\mathbb {E} _{D}[{\hat {f}}(x;D)]-{\hat {f}}(x;D)\right)^{2}\right]}

and

σ 2 = E y ■ [(y – f (x) ■ E y | x [y]) 2] {\displaystyle \sigma ^{2}=\operatorname {E} _{y}{\Big [}{\big (}y-\underbrace {f(x)} _{E_{y|x}[y]}{\big)}^{2}{\Big]}}

The expectation ranges over different choices of the training set D = { (x 1 , y 1) … , (x n , y n) } {\displaystyle D=\{(x_{1},y_{1})\dots ,(x_{n},y_{n})\}} , all sampled from the same joint distribution P (x , y) {\displaystyle P(x,y)} which can for example be done via bootstrapping .

The three terms represent:

the square of the bias of the learning method, which can be thought of as the error caused by the simplifying assumptions built into the method. E.g., when approximating a non-linear function f (x) {\displaystyle f(x)} using a learning method for linear models , there will be error in the estimates f ^ (x) {\displaystyle {\hat {f}}(x)} due to this assumption;

the variance of the learning method, or, intuitively, how much the learning method f ^ (x) {\displaystyle {\hat {f}}(x)} will move around its mean;

the irreducible error σ 2 {\displaystyle \sigma ^{2}} .

Since all three terms are non-negative, the irreducible error forms a lower bound on the expected error on unseen samples.

The more complex the model f ^ (x) {\displaystyle {\hat {f}}(x)} is, the more data points it will capture, and the lower the bias will be. However, complexity will make the model "move" more to capture the data points, and hence its variance will be larger.

Derivation

The derivation of the bias–variance decomposition for squared error proceeds as follows. For convenience, we drop the D {\displaystyle D} subscript in the following lines, such that f ^ (x ; D) = f ^ (x) {\displaystyle {\hat {f}}(x;D)={\hat {f}}(x)} .

Let us write the mean-squared error of our model:

MSE ■ E [(y – f ^ (x)) 2] = E [(f (x) + ε – f ^ (x)) 2] since y ■ f (x) + ε = E [(f (x) – f ^ (x)) 2] + 2 E [(f (x) – f ^ (x)) ε] + E [ε 2] {\displaystyle {\begin{aligned}{\text{MSE}}&\triangleq \mathbb {E} {\Big [}{\big (}y-{\hat {f}}(x){\big)}^{2}{\Big]}\\&=\mathbb {E} {\Big [}{\big (}f(x)+\varepsilon -{\hat {f}}(x){\big)}^{2}{\Big]}&&{\text{since }}y\triangleq f(x)+\varepsilon \\&=\mathbb {E} {\Big [}{\big (}f(x)-{\hat {f}}(x){\big)}^{2}{\Big]}\,+\,2\ \mathbb {E} {\Big [}{\big (}f(x)-{\hat {f}}(x){\big)}\varepsilon {\Big]}\,+\,\mathbb {E} [\varepsilon ^{2}]\end{aligned}}}

We can show that the second term of this equation is null:

E [(f (x) – f ^ (x)) ε] = E [f (x) – f ^ (x)] E [ε] since ε is independent from x = 0 since E [ε] = 0 {\displaystyle {\begin{aligned}\mathbb {E} {\Big [}{\big (}f(x)-{\hat {f}}(x){\big)}\varepsilon {\Big]}&=\mathbb {E} {\big [}f(x)-{\hat {f}}(x){\big]}\ \mathbb {E} {\big [}\varepsilon {\big]}&&{\text{since }}\varepsilon {\text{ is independent from }}x\\&=0&&{\text{since }}\mathbb {E} {\big [}\varepsilon {\big]}=0\end{aligned}}}

Moreover, the third term of this equation is nothing but σ 2 {\displaystyle \sigma ^{2}} , the variance of ε {\displaystyle \varepsilon } .

Let us now expand the remaining term:

E [(f (x) – f ^ (x)) 2] = E [(f (x) – E [f ^ (x)] + E [f ^ (x)] – f ^ (x)) 2] = E [(f (x) – E [f ^ (x)]) 2] + 2 E [(f (x) – E [f ^ (x)]) (E [f ^ (x)] – f ^ (x))] + E [(E [f ^ (x)] – f ^ (x)) 2] {\displaystyle {\begin{aligned}&\mathbb {E} {\Big [}{\big (}f(x)-{\hat {f}}(x){\big)}^{2}{\Big]}\\&=\mathbb {E} {\Big [}{\big (}f(x)-\mathbb {E} {\big [}{\hat {f}}(x){\big]}+\mathbb {E} {\big [}{\hat {f}}(x){\big]}-{\hat {f}}(x){\big)}^{2}{\Big]}\\&={\color {Blue}\mathbb {E} {\Big [}{\big (}f(x)-\mathbb {E} {\big [}{\hat {f}}(x){\big]}{\big)}^{2}{\Big]}}\,+\,2\ {\color {PineGreen}\mathbb {E} {\Big [}{\big (}f(x)-\mathbb {E} {\big [}{\hat {f}}(x){\big]}{\big)}{\big (}\mathbb {E} {\big [}{\hat {f}}(x){\big]}-{\hat {f}}(x){\big)}{\Big]}}\,+\,\mathbb {E} {\Big [}{\big (}\mathbb {E} {\big [}{\hat {f}}(x){\big]}-{\hat {f}}(x){\big)}^{2}{\Big]}\end{aligned}}}

We show that:

E [(f (x) – E [f ^ (x)]) 2] = E [f (x) 2] – 2 E [f (x) E [f ^ (x)]] + E [E [f ^ (x)] 2] = f (x) 2 – 2 f (x) E [f ^ (x)] + E [f ^ (x)] 2 = (f (x) – E [f ^ (x)]) 2 {\displaystyle {\begin{aligned}{\color {Blue}\mathbb {E} {\Big [}{\big (}f(x)-\mathbb {E} {\big [}{\hat {f}}(x){\big]}{\big)}^{2}{\Big]}}&=\mathbb {E} {\big [}f(x)^{2}{\big]}\,-\,2\ \mathbb {E} {\Big [}f(x)\ \mathbb {E} {\big [}{\hat {f}}(x){\big]}{\Big]}\,+\,\mathbb {E} {\Big [}\mathbb {E} {\big [}{\hat {f}}(x){\big]}^{2}{\Big]}\\&=f(x)^{2}\,-\,2\ f(x)\ \mathbb {E} {\big [}{\hat {f}}(x){\big]}\,+\,\mathbb {E} {\big [}{\hat {f}}(x){\big]}^{2}\\&={\Big (}f(x)-\mathbb {E} {\big [}{\hat {f}}(x){\big]}{\Big)}^{2}\end{aligned}}}

This last series of equalities comes from the fact that f (x) {\displaystyle f(x)} is not a random variable, but a fixed, deterministic function of x {\displaystyle x} . Therefore, E [f (x)] = f (x) {\displaystyle \mathbb {E} {\big [}f(x){\big]}=f(x)} . Similarly E [f (x) 2] = f (x) 2 {\displaystyle \mathbb {E} {\big [}f(x)^{2}{\big]}=f(x)^{2}} , and E [f (x) E [f ^ (x)]] = f (x) E [E [f ^ (x)]] = f (x) E [f ^ (x)] {\displaystyle \mathbb {E} {\Big [}f(x)\ \mathbb {E} {\big [}{\hat {f}}(x){\big]}{\Big]}=f(x)\ \mathbb {E} {\Big [}\ \mathbb {E} {\big [}{\hat {f}}(x){\big]}{\Big]}=f(x)\ \mathbb {E} {\big [}{\hat {f}}(x){\big]}} . Using the same reasoning, we can expand the second term and show that it is null:

E [(f (x) – E [f ^ (x)]) (E [f ^ (x)] – f ^ (x))] = E [f (x) E [f ^ (x)] – f (x) f ^ (x) – E [f ^ (x)] 2 + E [f ^ (x)] f ^ (x)] = f (x) E [f ^ (x)] – f (x) E [f ^ (x)] – E [f ^ (x)] 2 + E [f ^ (x)] 2 = 0 {\displaystyle {\begin{aligned}&{\color {PineGreen}\mathbb {E} {\Big [}{\big (}f(x)-\mathbb {E} {\big [}{\hat {f}}(x){\big]}{\big)}{\big (}\mathbb {E} {\big [}{\hat {f}}(x){\big]}-{\hat {f}}(x){\big)}{\Big]}}\\&=\mathbb {E} {\Big [}f(x)\ \mathbb {E} {\big [}{\hat {f}}(x){\big]}\,-\,f(x){\hat {f}}(x)\,-\,\mathbb {E} {\big [}{\hat {f}}(x){\big]}^{2}+\mathbb {E} {\big [}{\hat {f}}(x){\big]}\ {\hat {f}}(x){\Big]}\\&=f(x)\ \mathbb {E} {\big [}{\hat {f}}(x){\big]}\,-\,f(x)\ \mathbb {E} {\big [}{\hat {f}}(x){\big]}\,-\,\mathbb {E} {\big [}{\hat {f}}(x){\big]}^{2}\,+\,\mathbb {E} {\big [}{\hat {f}}(x){\big]}^{2}\\&=0\end{aligned}}}

Eventually, we plug our derivations back into the original equation, and identify each term:

MSE = (f (x) – E [f ^ (x)]) 2 + E [(E [f ^ (x)] – f ^ (x)) 2] + σ 2 = Bias ■ (f ^ (x)) 2 + Var ■ [f ^ (x)] + σ 2 {\displaystyle {\begin{aligned}{\text{MSE}}}&={\Big (}f(x)-\mathbb {E} {\big [}{\hat {f}}(x){\big]}{\Big)}^{2}+\mathbb {E} {\Big [}{\big (}\mathbb {E} {\big [}{\hat {f}}(x){\big]}-{\hat {f}}(x){\big)}^{2}{\Big]}+\sigma ^{2}\\&=\operatorname {Bias} {\big (}{\hat {f}}(x){\big)}^{2}\,+\,\operatorname {Var} {\big [}{\hat {f}}(x){\big]}\,+\,\sigma ^{2}\end{aligned}}}

Finally, the MSE loss function (or negative log-likelihood) is obtained by taking the expectation value over x ∼ P {\displaystyle x\sim P} : MSE = E x { Bias D ■ [f ^ (x ; D)] 2 + Var D ■ [f ^ (x ; D)] } + σ 2 . {\displaystyle {\text{MSE}}=\mathbb {E} _{x}{\bigg \{}\operatorname {Bias} _{D}[{\hat {f}}(x;D)]^{2}+\operatorname {Var} _{D}{\big [}{\hat {f}}(x;D){\big]}{\bigg \}}+\sigma ^{2}.}

Approaches

Dimensionality reduction and feature selection can decrease variance by simplifying models. Similarly, a larger training set tends to decrease variance. Adding features (predictors) tends to decrease bias, at the expense of introducing additional variance. Learning algorithms typically have some tunable parameters that control bias and variance; for example,

linear and Generalized linear models can be regularized to decrease their variance at the cost of increasing their bias.

In artificial neural networks , the variance increases and the bias decreases as the number of hidden units increase, although this classical assumption has been the subject of recent debate. Like in GLMs, regularization is typically applied.

In k -nearest neighbor models, a high value of k leads to high bias and low variance (see below).

In instance-based learning , regularization can be achieved varying the mixture of prototypes and exemplars.

In decision trees , the depth of the tree determines the variance. Decision trees are commonly pruned to control variance.

One way of resolving the trade-off is to use mixture models and ensemble learning . For example, boosting combines many "weak" (high bias) models in an ensemble that has lower bias than the individual models, while bagging combines "strong" learners in a way that reduces their variance.

Model validation methods such as cross-validation (statistics) can be used to tune models so as to optimize the trade-off.

k -nearest neighbors

In the case of k -nearest neighbors regression , when the expectation is taken over the possible labeling of a fixed training set, a closed-form expression exists that relates the bias–variance decomposition to the parameter k :

E [(y – f ^ (x)) 2 ■ X = x] = (f (x) – 1 k ∑ i = 1 k f (N i (x))) 2 + σ 2 k + σ 2 {\displaystyle \mathbb {E} \left[(y-{\hat {f}}(x))^{2}\mid X=x\right]=\left(f(x)-{\frac {1}{k}}\sum _{i=1}^{k}f(N_{i}(x))\right)^{2}+{\frac {\sigma ^{2}}{k}}+\sigma ^{2}}

where N 1 (x) , … , N k (x) {\displaystyle N_{1}(x),\dots ,N_{k}(x)} are the k nearest neighbors of x in the training set. The bias (first term) is a monotone rising function of k , while the variance (second term) drops off as k is increased. In fact, under "reasonable assumptions" the bias of the first-nearest neighbor (1-NN) estimator vanishes entirely as the size of the training set approaches infinity.

Applications

In regression

The bias–variance decomposition forms the conceptual basis for regression regularization methods such as LASSO and ridge regression . Regularization methods introduce bias into the regression solution that can reduce variance considerably relative to the ordinary least squares (OLS) solution. Although the OLS solution provides non-biased regression estimates, the lower variance solutions

produced by regularization techniques provide superior MSE performance.

In classification

The bias–variance decomposition was originally formulated for least-squares regression. For the case of classification under the 0-1 loss (misclassification rate), it is possible to find a similar decomposition, with the caveat that the variance term becomes dependent on the target label. Alternatively, if the classification problem can be phrased as probabilistic classification , then the expected cross-entropy can instead be decomposed to give bias and variance terms with the same semantics but taking a different form.

It has been argued that as training data increases, the variance of learned models will tend to decrease, and hence that as training data quantity increases, error is minimised by methods that learn models with lesser bias, and that conversely, for smaller training data quantities it is ever more important to minimise variance.

In reinforcement learning

Even though the bias–variance decomposition does not directly apply in reinforcement learning , a similar tradeoff can also characterize generalization. When an agent has limited information on its environment, the suboptimality of an RL algorithm can be decomposed into the sum of two terms: a term related to an asymptotic bias and a term due to overfitting. The asymptotic bias is directly related to the learning algorithm (independently of the quantity of data) while the overfitting term comes from the fact that the amount of data is limited.

In Monte Carlo methods

While in traditional Monte Carlo methods the bias is typically zero, modern approaches, such as Markov chain Monte Carlo are only asymptotically unbiased, at best. Convergence diagnostics can be used to control bias via burn-in removal, but due to a limited computational budget, a bias–variance trade-off arises, leading to a wide-range of approaches, in which a controlled bias is accepted, if this allows to dramatically reduce the variance, and hence the overall estimation error.

In human learning

While widely discussed in the context of machine learning, the bias–variance dilemma has been examined in the context of human cognition , most notably by Gerd Gigerenzer and co-workers in the context of learned heuristics. They have argued (see references below) that the human brain resolves the dilemma in the case of the typically sparse, poorly-characterized training-sets provided by experience by adopting high-bias/low variance heuristics. This reflects the fact that a zero-bias approach has poor generalizability to new situations, and also unreasonably presumes precise knowledge of the true state of the world. The resulting heuristics are relatively simple, but produce better inferences in a wider variety of situations.

Geman et al. argue that the bias–variance dilemma implies that abilities such as generic object recognition cannot be learned from scratch, but require a certain degree of "hard wiring" that is later tuned by experience. This is because model-free approaches to inference require impractically large training sets if they are to avoid high variance.

See also

Accuracy and precision

Bias of an estimator

Double descent

Gauss–Markov theorem

Hyperparameter optimization

Law of total variance

Minimum-variance unbiased estimator

Model selection

Regression model validation

Supervised learning

Cramér–Rao bound

Prediction interval

References

External links

MLU-Explain: The Bias Variance Tradeoff — An interactive visualization of the bias–variance tradeoff in LOESS Regression and K-Nearest Neighbors.

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