Title: Learning rate

URL: https://en.wikipedia.org/wiki/Learning_rate

PageID: 59969558

Categories: Category:Machine learning, Category:Model selection, Category:Optimization algorithms and methods

-----

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 and statistics , the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function . [1] Since it influences to what extent newly acquired information overrides old information, it metaphorically represents the speed at which a machine learning model "learns". In the adaptive control literature, the learning rate is commonly referred to as gain . [2]

In setting a learning rate, there is a trade-off between the rate of convergence and overshooting . While the descent direction is usually determined from the gradient of the loss function, the learning rate determines how big a step is taken in that direction. A too high learning rate will make the learning jump over minima but a too low learning rate will either take too long to converge or get stuck in an undesirable local minimum. [3]

In order to achieve faster convergence, prevent oscillations and getting stuck in undesirable local minima the learning rate is often varied during training either in accordance to a learning rate schedule or by using an adaptive learning rate. [4] The learning rate and its adjustments may also differ per parameter, in which case it is a diagonal matrix that can be interpreted as an approximation to the inverse of the Hessian matrix in Newton's method . [5] The learning rate is related to the step length determined by inexact line search in quasi-Newton methods and related optimization algorithms. [6] [7]

Learning rate schedule

Initial rate can be left as system default or can be selected using a range of techniques. [8] A learning rate schedule changes the learning rate during learning and is most often changed between epochs/iterations. This is mainly done with two parameters: decay and momentum . There are many different learning rate schedules but the most common are time-based, step-based and exponential . [4]

Decay serves to settle the learning in a nice place and avoid oscillations, a situation that may arise when a too high constant learning rate makes the learning jump back and forth over a minimum, and is controlled by a hyperparameter.

Momentum is analogous to a ball rolling down a hill; we want the ball to settle at the lowest point of the hill (corresponding to the lowest error). Momentum both speeds up the learning (increasing the learning rate) when the error cost gradient is heading in the same direction for a long time and also avoids local minima by 'rolling over' small bumps. Momentum is controlled by a hyperparameter analogous to a ball's mass which must be chosen manually—too high and the ball will roll over minima which we wish to find, too low and it will not fulfil its purpose. The formula for factoring in the momentum is more complex than for decay but is most often built in with deep learning libraries such as Keras .

Time-based learning schedules alter the learning rate depending on the learning rate of the previous time iteration. Factoring in the decay the mathematical formula for the learning rate is:

η n + 1 = η n 1 + d n {\displaystyle \eta _{n+1}={\frac {\eta _{n}}{1+dn}}}

where η {\displaystyle \eta } is the learning rate, d {\displaystyle d} is a decay parameter and n {\displaystyle n} is the iteration step.

Step-based learning schedules changes the learning rate according to some predefined steps. The decay application formula is here defined as:

η n = η 0 d ■ 1 + n r ■ {\displaystyle \eta _{n}=\eta _{0}d^{\left\lfloor {\frac {1+n}{r}}\right\rfloor }}

where η n {\displaystyle \eta _{n}} is the learning rate at iteration n {\displaystyle n} , η 0 {\displaystyle \eta _{0}} is the initial learning rate, d {\displaystyle d} is how much the learning rate should change at each drop (0.5 corresponds to a halving) and r {\displaystyle r} corresponds to the drop rate , or how often the rate should be dropped (10 corresponds to a drop every 10 iterations). The floor function (■ … ■ {\displaystyle \lfloor \dots \rfloor }) here drops the value of its input to 0 for all values smaller than 1.

Exponential learning schedules are similar to step-based, but instead of steps, a decreasing exponential function is used. The mathematical formula for factoring in the decay is:

η n = η 0 e − d n {\displaystyle \eta _{n}=\eta _{0}e^{-dn}}

where d {\displaystyle d} is a decay parameter.

Adaptive learning rate

The issue with learning rate schedules is that they all depend on hyperparameters that must be manually chosen for each given learning session and may vary greatly depending on the problem at hand or the model used. To combat this, there are many different types of adaptive gradient descent algorithms such as Adagrad , Adadelta, RMSprop , and Adam [9] which are generally built into deep learning libraries such as Keras . [10]

See also

Hyperparameter (machine learning)

Hyperparameter optimization

Stochastic gradient descent

Variable metric methods

Overfitting

Backpropagation

AutoML

Model selection

Self-tuning

References

Further reading

Géron, Aurélien (2017). "Gradient Descent" . Hands-On Machine Learning with Scikit-Learn and TensorFlow . O'Reilly. pp. 113– 124. ISBN 978-1-4919-6229-9 .

Plagianakos, V. P.; Magoulas, G. D.; Vrahatis, M. N. (2001). "Learning Rate Adaptation in Stochastic Gradient Descent" . Advances in Convex Analysis and Global Optimization . Kluwer. pp. 433– 444. ISBN 0-7923-6942-4 .

External links

de Freitas, Nando (February 12, 2015). "Optimization" . Deep Learning Lecture 6 . University of Oxford – via YouTube .