Index

Symbols

- @ operator (matrix multiplication), The Normal Equation
- β (momentum), Momentum
- γ (gamma) value, Gaussian RBF Kernel
- ε (tolerance), Batch Gradient Descent, SVM Classes and Computational Complexity
- ε greedy policy, Exploration Policies
- ε neighborhood, DBSCAN
- ε sensitive, SVM Regression
- χ^2 test, Regularization Hyperparameters
- ℓ_0 norm, Select a Performance Measure
- ℓ_1 norm, Select a Performance Measure
- ℓ_2 norm, Select a Performance Measure
- ℓ_k norm, Select a Performance Measure

A

- A/B experiments, Training and Deploying TensorFlow Models at Scale
- accelerated k-means, Accelerated k-means and mini-batch k-means
- accelerated linear algebra (XLA), TensorFlow Functions and Graphs
- accuracy performance measure, What Is Machine Learning?, Measuring Accuracy Using Cross-Validation
- ACF (autocorrelation function), The ARMA Model Family
- action advantage, reinforcement learning, Evaluating Actions: The Credit Assignment Problem
- action potentials (APs), Biological Neurons
- actions, in reinforcement learning, Learning to Optimize Rewards, Evaluating Actions: The Credit Assignment Problem-Evaluating Actions: The Credit Assignment Problem
- activation functions, The Perceptron, The Multilayer Perceptron and Backpropagation-The Multilayer Perceptron and Backpropagation
  - for Conv2D layer, Implementing Convolutional Layers with Keras
  - custom models, Custom Activation Functions, Initializers, Regularizers, and Constraints

    - ELU, ELU and SELU-GELU, Swish, and Mish
    - GELU, GELU, Swish, and Mish-GELU, Swish, and Mish
    - hyperbolic tangent (htan), The Multilayer Perceptron and Backpropagation, Fighting the Unstable Gradients Problem
    - hyperparameters, Learning Rate, Batch Size, and Other Hyperparameters
    - initialization parameters, Glorot and He Initialization
    - leakyReLU, Leaky ReLU-Leaky ReLU
    - Mish, GELU, Swish, and Mish
    - PReLU, Leaky ReLU
    - ReLU (see ReLU)
    - RReLU, Leaky ReLU
    - SELU, ELU and SELU, Dropout
    - sigmoid, Estimating Probabilities, The Multilayer Perceptron and Backpropagation, The Vanishing/Exploding Gradients Problems, Sparse Autoencoders
    - SiLU, GELU, Swish, and Mish
    - softmax, Softmax Regression, Classification MLPs, Creating the model using the sequential API, An Encoder–Decoder Network for Neural Machine Translation
    - softplus, Regression MLPs
    - Swish, GELU, Swish, and Mish
- active learning, Using Clustering for Semi-Supervised Learning
- actor-critic, Overview of Some Popular RL Algorithms
- actual class, Confusion Matrices
- actual versus estimated probabilities, The ROC Curve
- AdaBoost, AdaBoost-AdaBoost
- AdaGrad, AdaGrad
- Adam optimization, Adam, ℓ1 and ℓ2 Regularization
- AdaMax, AdaMax
- AdamW, AdamW, ℓ1 and ℓ2 Regularization
- adaptive boosting (AdaBoost), Boosting-AdaBoost
- adaptive instance normalization (AdaIN), StyleGANs
- adaptive learning rate algorithms, AdaGrad-AdamW
- adaptive moment estimation (Adam), Adam
- additive attention, Attention Mechanisms
- advantage actor-critic (A2C), Overview of Some Popular RL Algorithms
- adversarial learning, Semantic Segmentation, Autoencoders, GANs, and Diffusion Models

- affine transformations, StyleGANs
- affinity function, Clustering Algorithms: k-means and DBSCAN
- affinity propagation, Other Clustering Algorithms
- agents, reinforcement learning, Reinforcement learning, Learning to Optimize Rewards, Policy Gradients, Q-Learning
- agglomerative clustering, Other Clustering Algorithms
- Akaike information criterion (AIC), Selecting the Number of Clusters
- AlexNet, AlexNet
- algorithms, preparing data for, Prepare the Data for Machine Learning Algorithms-Transformation Pipelines
- alignment model, Attention Mechanisms
- AllReduce algorithm, Data parallelism using the mirrored strategy
- alpha dropout, Dropout
- AlphaGo, Reinforcement learning, Reinforcement Learning, Overview of Some Popular RL Algorithms
- anchor priors, You Only Look Once
- ANNs (see artificial neural networks)
- anomaly detection, Examples of Applications, Unsupervised learning
  - clustering for, Clustering Algorithms: k-means and DBSCAN
  - GMM, Using Gaussian Mixtures for Anomaly Detection-Using Gaussian Mixtures for Anomaly Detection
  - isolation forest, Other Algorithms for Anomaly and Novelty Detection
- AP (average precision), You Only Look Once
- APs (action potentials), Biological Neurons
- area under the curve (AUC), The ROC Curve
- argmax(), Softmax Regression
- ARIMA model, The ARMA Model Family-The ARMA Model Family
- ARMA model family, The ARMA Model Family-The ARMA Model Family
- artificial neural networks (ANNs), Introduction to Artificial Neural Networks with Keras-Learning Rate, Batch Size, and Other Hyperparameters
  - autoencoders (see autoencoders)
  - backpropagation, The Multilayer Perceptron and Backpropagation-The Multilayer Perceptron and Backpropagation
  - biological neurons as background to, Biological Neurons-Biological Neurons

    - evolution of, From Biological to Artificial Neurons-From Biological to Artificial Neurons
    - hyperparameter fine-tuning, Fine-Tuning Neural Network Hyperparameters-Learning Rate, Batch Size, and Other Hyperparameters
    - implementing MLPs with Keras, Implementing MLPs with Keras-Using TensorBoard for Visualization
    - logical computations with neurons, Logical Computations with Neurons-Logical Computations with Neurons
    - perceptrons (see multilayer perceptrons)
    - reinforcement learning policies, Neural Network Policies
- artificial neuron, Logical Computations with Neurons
- association rule learning, Unsupervised learning
- assumptions, checking in model building, Data Mismatch, Check the Assumptions
- asynchronous advantage actor-critic (A3C), Overview of Some Popular RL Algorithms
- asynchronous gang scheduling, Bandwidth saturation
- asynchronous updates, with centralized parameters, Asynchronous updates
- à-trous convolutional layer, Semantic Segmentation
- attention mechanisms, Natural Language Processing with RNNs and Attention, Attention Mechanisms-Multi-head attention, Vision Transformers
    - (see also transformer models)
- attributes, Take a Quick Look at the Data Structure-Take a Quick Look at the Data Structure
    - categorical, Handling Text and Categorical Attributes, Handling Text and Categorical Attributes
    - combinations of, Experiment with Attribute Combinations-Experiment with Attribute Combinations
    - preprocessed, Take a Quick Look at the Data Structure
    - target, Take a Quick Look at the Data Structure
    - unsupervised learning, Supervised learning
- AUC (area under the curve), The ROC Curve
- autocorrelated time series, Forecasting a Time Series
- autocorrelation function (ACF), The ARMA Model Family
- autodiff (automatic differentiation), Autodiff-Reverse-Mode Autodiff

    - for computing gradients, Computing Gradients Using Autodiff-Computing Gradients Using Autodiff
    - finite difference approximation, Finite Difference Approximation
    - forward-mode, Forward-Mode Autodiff-Forward-Mode Autodiff
    - manual differentiation, Manual Differentiation
    - reverse-mode, Reverse-Mode Autodiff-Reverse-Mode Autodiff
- autoencoders, Autoencoders, GANs, and Diffusion Models-Generating Fashion MNIST Images
    - convolutional, Convolutional Autoencoders-Convolutional Autoencoders
    - denoising, Denoising Autoencoders-Denoising Autoencoders
    - efficient data representations, Efficient Data Representations-Efficient Data Representations
    - overcomplete, Convolutional Autoencoders
    - PCA with undercomplete linear autoencoder, Performing PCA with an Undercomplete Linear Autoencoder-Performing PCA with an Undercomplete Linear Autoencoder
    - sparse, Sparse Autoencoders-Sparse Autoencoders
    - stacked, Stacked Autoencoders-Training One Autoencoder at a Time
    - training one at a time, Training One Autoencoder at a Time-Training One Autoencoder at a Time
    - undercomplete, Efficient Data Representations
    - variational, Variational Autoencoders-Variational Autoencoders
- Autograph, AutoGraph and Tracing
- AutoGraph, Using AutoGraph to Capture Control Flow
- AutoML service, Fine-Tuning Neural Network Hyperparameters, Hyperparameter Tuning on Vertex AI
- autoregressive integrated moving average (ARIMA) model, The ARMA Model Family-The ARMA Model Family
- autoregressive model, The ARMA Model Family
- autoregressive moving average (ARMA) model family, The ARMA Model Family-The ARMA Model Family
- auxiliary task, pretraining on, Pretraining on an Auxiliary Task
- average absolute deviation, Select a Performance Measure
- average pooling layer, Implementing Pooling Layers with Keras
- average precision (AP), You Only Look Once

B

- backbone, model, GoogLeNet

- backpropagation, The Multilayer Perceptron and Backpropagation-The Multilayer Perceptron and Backpropagation, The Vanishing/Exploding Gradients Problems, Computing Gradients Using Autodiff, Generative Adversarial Networks
- backpropagation through time (BPTT), Training RNNs
- bagging (bootstrap aggregating), Bagging and Pasting-Random Patches and Random Subspaces
- Bahdanau attention, Attention Mechanisms
- balanced iterative reducing and clustering using hierarchies (BIRCH), Other Clustering Algorithms
- bandwidth saturation, Bandwidth saturation-Bandwidth saturation
- BaseEstimator, Custom Transformers
- batch gradient descent, Batch Gradient Descent-Batch Gradient Descent, Ridge Regression
- batch learning, Batch learning-Batch learning
- batch normalization (BN), Batch Normalization-Implementing batch normalization with Keras, Fighting the Unstable Gradients Problem
- batch predictions, Deploying a new model version, Creating a Prediction Service on Vertex AI, Running Batch Prediction Jobs on Vertex AI
- Bayesian Gaussian mixtures, Bayesian Gaussian Mixture Models
- Bayesian information criterion (BIC), Selecting the Number of Clusters
- beam search, Beam Search-Beam Search
- Bellman optimality equation, Markov Decision Processes
- bias, Learning Curves
- bias and fairness, NLP transformers, Hugging Face's Transformers Library
- bias term constant, Linear Regression, The Normal Equation
- bias/variance trade-off, Learning Curves
- BIC (Bayesian information criterion), Selecting the Number of Clusters
- bidirectional recurrent layer, Bidirectional RNNs
- binary classifiers, Training a Binary Classifier, Logistic Regression
- binary logarithm, Computational Complexity
- binary trees, Making Predictions, Other Clustering Algorithms
- biological neural networks (BNNs), Biological Neurons-Biological Neurons
- BIRCH (balanced iterative reducing and clustering using hierarchies), Other Clustering Algorithms
- black box models, Making Predictions

- blending, in stacking, Stacking
- BN (batch normalization), Batch Normalization, Fighting the Unstable Gradients Problem
- BNNs (biological neural networks), Biological Neurons-Biological Neurons
- boosting, Boosting-Histogram-Based Gradient Boosting
- bootstrap aggregating (bagging), Bagging and Pasting-Random Patches and Random Subspaces
- bootstrapping, Bagging and Pasting
- bottleneck layers, GoogLeNet
- bounding boxes, image identification, Classification and Localization-You Only Look Once
- BPTT (backpropagation through time), Training RNNs
- bucketizing a feature, Feature Scaling and Transformation
- byte pair encoding (BPE), Sentiment Analysis

C

- California Housing Prices dataset, Working with Real Data-Check the Assumptions
- callbacks, Using Callbacks-Using Callbacks
- CART (Classification and Regression Tree) algorithm, Making Predictions, The CART Training Algorithm, Regression
- catastrophic forgetting, Implementing Deep Q-Learning
- categorical attributes, Handling Text and Categorical Attributes, Handling Text and Categorical Attributes
- categorical features, encoding, The Hashing Layer-Encoding Categorical Features Using Embeddings
- CategoryEncoding layer, The CategoryEncoding Layer
- causal model, Forecasting Using a Sequence-to-Sequence Model, Bidirectional RNNs
- centralized parameters, Data parallelism with centralized parameters-Asynchronous updates
- centroid, cluster, Clustering Algorithms: k-means and DBSCAN, k-means, The k-means algorithm-Centroid initialization methods
- chain rule, The Multilayer Perceptron and Backpropagation
- chain-of-thought prompting, An Avalanche of Transformer Models
- ChainClassifier, Multilabel Classification
- chaining transformations, Chaining Transformations-Chaining Transformations

- char-RNN model, Generating Shakespearean Text Using a Character RNN-Stateful RNN
- chatbot or personal assistant, Examples of Applications
- check_estimator(), Custom Transformers
- chi-squared (χ^2) test, Regularization Hyperparameters
- classification, Classification-Multioutput Classification
  - application examples of, Examples of Applications-Examples of Applications
  - binary classifier, Training a Binary Classifier, Logistic Regression
  - CNNs, Classification and Localization-You Only Look Once
  - error analysis, Error Analysis-Error Analysis
  - hard margin, Soft Margin Classification, Under the Hood of Linear SVM Classifiers
  - hard voting classifiers, Voting Classifiers
  - image (see images)
  - logistic regression (see logistic regression)
  - MLPs for, Classification MLPs-Classification MLPs
  - MNIST dataset, MNIST-MNIST
  - multiclass, Multiclass Classification-Multiclass Classification, Classification MLPs-Classification MLPs
  - multilabel, Multilabel Classification-Multilabel Classification
  - multioutput, Multioutput Classification-Multioutput Classification
  - performance measures, Performance Measures-The ROC Curve
  - and regression, Supervised learning, Multioutput Classification
  - soft margin, Soft Margin Classification-Soft Margin Classification
  - softmax regression, Softmax Regression-Softmax Regression
  - SVMs (see support vector machines)
  - text, Sentiment Analysis-Reusing Pretrained Embeddings and Language Models
  - voting classifiers, Voting Classifiers-Voting Classifiers
- Classification and Regression Tree (CART) algorithm, Making Predictions, The CART Training Algorithm, Regression
- clone(), Early Stopping
- closed-form equation/solution, Training Models, The Normal Equation, Ridge Regression, Training and Cost Function
- cloud platform deployment with Vertex AI, Creating a Prediction Service on Vertex AI-Creating a Prediction Service on Vertex AI
- clustering algorithms, Examples of Applications, Unsupervised learning, Unsupervised Learning Techniques-Other Clustering Algorithms

  - affinity propagation, Other Clustering Algorithms
  - agglomerative clustering, Other Clustering Algorithms
  - applications for, Clustering Algorithms: k-means and DBSCAN-Clustering Algorithms: k-means and DBSCAN
  - BIRCH, Other Clustering Algorithms
  - DBSCAN, DBSCAN-DBSCAN
  - GMM, Gaussian Mixtures-Other Algorithms for Anomaly and Novelty Detection
  - image segmentation, Clustering Algorithms: k-means and DBSCAN, Using Clustering for Image Segmentation-Using Clustering for Image Segmentation
  - k-means (see k-means algorithm)
  - mean-shift, Other Clustering Algorithms
  - responsibilities of clusters for instances, Gaussian Mixtures
  - semi-supervised learning with, Using Clustering for Semi-Supervised Learning-Using Clustering for Semi-Supervised Learning
  - spectral clustering, Other Clustering Algorithms
- CNNs (see convolutional neural networks)
- Colab, Running the Code Examples Using Google Colab-Running the Code Examples Using Google Colab
- color channels, Stacking Multiple Feature Maps
- color segmentation, images, Using Clustering for Image Segmentation
- column vectors, Linear Regression
- ColumnTransformer, Transformation Pipelines
- complex models with functional API, Building Complex Models Using the Functional API-Building Complex Models Using the Functional API
- compound scaling, Other Noteworthy Architectures
- compressed TFRecord files, Compressed TFRecord Files
- compression and decompression, PCA, PCA for Compression-PCA for Compression
- computation graphs, A Quick Tour of TensorFlow
- computational complexity
  - DBSCAN, DBSCAN
  - decision trees, Computational Complexity
  - Gaussian mixture model, Gaussian Mixtures
  - histogram-based gradient boosting, Histogram-Based Gradient Boosting
  - k-means algorithm, The k-means algorithm

    - Normal equation, Computational Complexity
    - and SVM classes, SVM Classes and Computational Complexity
- concatenative attention, Attention Mechanisms
- concrete function, TF Functions and Concrete Functions-Exploring Function Definitions and Graphs
- conditional GAN, Deep Convolutional GANs
- conditional probability, Beam Search
- confidence interval, Evaluate Your System on the Test Set
- confusion matrix (CM), Confusion Matrices-Confusion Matrices, Error Analysis-Error Analysis
- ConfusionMatrixDisplay, Error Analysis
- connectionism, From Biological to Artificial Neurons
- constrained optimization, Under the Hood of Linear SVM Classifiers
- constraints, custom models, Custom Activation Functions, Initializers, Regularizers, and Constraints
- convergence rate, Batch Gradient Descent
- convex function, Gradient Descent
- convex quadratic optimization, Under the Hood of Linear SVM Classifiers
- convolution kernels (kernels), Filters, CNN Architectures
- convolutional neural networks (CNNs), Deep Computer Vision Using Convolutional Neural Networks-Semantic Segmentation
    - architectures, CNN Architectures-Choosing the Right CNN Architecture
    - autoencoders, Convolutional Autoencoders-Convolutional Autoencoders
    - classification and localization, Classification and Localization-You Only Look Once
    - convolutional layers, Convolutional Layers-Memory Requirements, Semantic Segmentation-Semantic Segmentation, Using 1D convolutional layers to process sequences, WaveNet-WaveNet, Masking
    - evolution of, Deep Computer Vision Using Convolutional Neural Networks
    - GANs, Deep Convolutional GANs-Deep Convolutional GANs
    - object detection, Object Detection-You Only Look Once
    - object tracking, Object Tracking
    - pooling layers, Pooling Layers-Implementing Pooling Layers with Keras

    - pretrained models from Keras, Using Pretrained Models from Keras-Using Pretrained Models from Keras
    - ResNet-34 CNN using Keras, Implementing a ResNet-34 CNN Using Keras
    - semantic segmentation, Examples of Applications, Using Clustering for Image Segmentation, Semantic Segmentation-Semantic Segmentation
    - splitting across devices, Model Parallelism
    - transfer learning pretrained models, Pretrained Models for Transfer Learning-Pretrained Models for Transfer Learning
    - U-Net, Diffusion Models
    - and vision transformers, Vision Transformers
    - visual cortex architecture, The Architecture of the Visual Cortex
    - WaveNet, WaveNet-WaveNet
- copy.deepcopy(), Early Stopping
- core instance, DBSCAN
- correlation coefficient, Look for Correlations-Look for Correlations
- cost function, Model-based learning and a typical machine learning workflow, Select a Performance Measure
    - in AdaBoost, AdaBoost
    - in autoencoders, Performing PCA with an Undercomplete Linear Autoencoder
    - in bounding box prediction model, Classification and Localization
    - in CART training algorithm, The CART Training Algorithm, Regression
    - in elastic net, Elastic Net Regression
    - in gradient descent, Training Models, Gradient Descent-Gradient Descent, Stochastic Gradient Descent-Stochastic Gradient Descent, The Vanishing/Exploding Gradients Problems
    - in lasso regression, Lasso Regression-Lasso Regression
    - in Nesterov accelerated gradient, Nesterov Accelerated Gradient
    - in linear regression, Linear Regression
    - in logistic regression, Training and Cost Function-Training and Cost Function
    - in momentum optimization, Momentum
    - in ridge regression, Ridge Regression
    - in variational autoencoders, Variational Autoencoders
- credit assignment problem, Evaluating Actions: The Credit Assignment Problem-Evaluating Actions: The Credit Assignment Problem

- cross entropy, Softmax Regression
- cross-validation, Hyperparameter Tuning and Model Selection, Better Evaluation Using Cross-Validation-Better Evaluation Using Cross-Validation, Evaluate Your System on the Test Set, Measuring Accuracy Using Cross-Validation-Measuring Accuracy Using Cross-Validation, Learning Curves-Learning Curves
- cross_val_predict(), Confusion Matrices, The Precision/Recall Trade-off, The ROC Curve, Error Analysis, Stacking
- cross_val_score(), Better Evaluation Using Cross-Validation, Measuring Accuracy Using Cross-Validation
- CUDA library, Getting Your Own GPU
- curiosity-based exploration, Overview of Some Popular RL Algorithms
- curriculum learning, Overview of Some Popular RL Algorithms
- custom models and training algorithms, Customizing Models and Training Algorithms-Custom Training Loops
  - activation functions, Custom Activation Functions, Initializers, Regularizers, and Constraints
  - autodiff for computing gradients, Computing Gradients Using Autodiff-Computing Gradients Using Autodiff
  - constraints, Custom Activation Functions, Initializers, Regularizers, and Constraints
  - initializers, Custom Activation Functions, Initializers, Regularizers, and Constraints
  - layers, Custom Layers-Custom Layers
  - loss functions, Custom Loss Functions, Losses and Metrics Based on Model Internals-Losses and Metrics Based on Model Internals
  - metrics, Custom Metrics-Custom Metrics, Losses and Metrics Based on Model Internals-Losses and Metrics Based on Model Internals
  - models, Custom Models-Custom Models
  - regularizers, Custom Activation Functions, Initializers, Regularizers, and Constraints
  - saving and loading models, Saving and Loading Models That Contain Custom Components-Saving and Loading Models That Contain Custom Components
  - training loops, Custom Training Loops-Custom Training Loops
- custom transformers, Custom Transformers-Custom Transformers
- customer segmentation, Clustering Algorithms: k-means and DBSCAN

D

- DALL-E, Vision Transformers
- data
  - downloading, Saving Your Code Changes and Your Data-Saving Your Code Changes and Your Data
  - efficient data representations, Efficient Data Representations-Efficient Data Representations
  - enqueuing and dequeuing, Queues
  - finding correlations in, Look for Correlations-Look for Correlations
  - making assumptions about, Data Mismatch
  - overfitting (see overfitting of data)
  - preparing for ML algorithms, Prepare the Data for Machine Learning Algorithms-Transformation Pipelines
  - preparing for ML models, Preparing the Data for Machine Learning Models-Preparing the Data for Machine Learning Models
  - preprocessing (see loading and preprocessing data)
  - shuffling of, Shuffling the Data
  - test data (see test set)
  - time series (see time series data)
  - training data (see training set)
  - underfitting of, Train and Evaluate on the Training Set, Learning Curves-Learning Curves, Polynomial Kernel
  - unreasonable effectiveness, Insufficient Quantity of Training Data
  - visualizing (see visualization)
  - working with real data, Working with Real Data-Working with Real Data
- data analysis, clustering for, Clustering Algorithms: k-means and DBSCAN
- data augmentation, Exercises, AlexNet, Pretrained Models for Transfer Learning
- data cleaning, Clean the Data-Clean the Data
- data drift, Batch learning
- data mining, Why Use Machine Learning?
- data mismatch, Data Mismatch
- data parallelism, Data Parallelism-Bandwidth saturation
- data pipeline, Frame the Problem
- data snooping bias, Create a Test Set
- data structure, Take a Quick Look at the Data Structure-Take a Quick Look at the Data Structure, Special Data Structures-Queues
- data-efficient image transformers (DeiT), Vision Transformers

- DataFrame, Clean the Data, Handling Text and Categorical Attributes, Feature Scaling and Transformation
- Dataquest, Other Resources
- Datasets library, The TensorFlow Datasets Project
- DBSCAN, DBSCAN-DBSCAN
- DCGANs (deep convolutional GANS), Deep Convolutional GANs-Deep Convolutional GANs
- DDPM (denoising diffusion probabilistic model), Autoencoders, GANs, and Diffusion Models, Diffusion Models-Diffusion Models
- DDQN (dueling DQN), Dueling DQN
- decision boundaries, Decision Boundaries-Decision Boundaries, Softmax Regression, Making Predictions, Sensitivity to Axis Orientation, k-means
- decision function, The Precision/Recall Trade-off, Under the Hood of Linear SVM Classifiers-Under the Hood of Linear SVM Classifiers
- decision stumps, AdaBoost
- decision threshold, The Precision/Recall Trade-off-The Precision/Recall Trade-off
- decision trees, Decision Trees-Decision Trees Have a High Variance, Ensemble Learning and Random Forests
  - (see also ensemble learning)
  - bagging and pasting, Bagging and Pasting in Scikit-Learn
  - CART training algorithm, The CART Training Algorithm
  - class probability estimates, Estimating Class Probabilities
  - computational complexity, Computational Complexity
  - and decision boundaries, Making Predictions
  - GINI impurity or entropy measures, Gini Impurity or Entropy?
  - high variance with, Decision Trees Have a High Variance
  - predictions, Making Predictions-The CART Training Algorithm
  - regression tasks, Regression-Regression
  - regularization hyperparameters, Regularization Hyperparameters-Regularization Hyperparameters
  - sensitivity to axis orientation, Sensitivity to Axis Orientation
  - training and visualizing, Training and Visualizing a Decision Tree-Training and Visualizing a Decision Tree
  - in training the model, Train and Evaluate on the Training Set-Better Evaluation Using Cross-Validation
- DecisionTreeClassifier, Training and Visualizing a Decision Tree, Gini Impurity or Entropy?, Regularization Hyperparameters, Sensitivity to

Axis Orientation, Random Forests
- DecisionTreeRegressor, Train and Evaluate on the Training Set, Decision Trees, Regression, Gradient Boosting
- decision_function(), The Precision/Recall Trade-off
- deconvolution layer, Semantic Segmentation
- deep autoencoders (see stacked autoencoders)
- deep convolutional GANS (DCGANs), Deep Convolutional GANs-Deep Convolutional GANs
- deep Gaussian process, Monte Carlo (MC) Dropout
- deep learning, The Machine Learning Tsunami
  - (see also deep neural networks; reinforcement learning)
- deep neural networks (DNNs), Training Deep Neural Networks-Summary and Practical Guidelines
  - CNNs (see convolutional neural networks)
  - default configuration, Summary and Practical Guidelines
  - faster optimizers for, Faster Optimizers-AdamW
  - learning rate scheduling, Learning Rate Scheduling-Learning Rate Scheduling
  - MLPs (see multilayer perceptrons)
  - regularization, Avoiding Overfitting Through Regularization-Max-Norm Regularization
  - reusing pretrained layers, Reusing Pretrained Layers-Pretraining on an Auxiliary Task
  - RNNs (see recurrent neural networks)
  - and transfer learning, Self-supervised learning
  - unstable gradients, The Vanishing/Exploding Gradients Problems
  - vanishing and exploding gradients, The Vanishing/Exploding Gradients Problems-Gradient Clipping
- deep neuroevolution, Fine-Tuning Neural Network Hyperparameters
- deep Q-learning, Approximate Q-Learning and Deep Q-Learning-Dueling DQN
- deep Q-networks (DQNs) (see Q-learning algorithm)
- deepcopy(), Early Stopping
- DeepMind, Reinforcement learning
- degrees of freedom, Overfitting the Training Data, Learning Curves
- DeiT (data-efficient image transformers), Vision Transformers
- denoising autoencoders, Denoising Autoencoders-Denoising Autoencoders

- denoising diffusion probabilistic model (DDPM), Autoencoders, GANs, and Diffusion Models, Diffusion Models-Diffusion Models
- Dense layer, The Perceptron, Creating the model using the sequential API, Creating the model using the sequential API, Building Complex Models Using the Functional API
- dense matrix, Feature Scaling and Transformation, Transformation Pipelines
- density estimation, Unsupervised Learning Techniques, DBSCAN-DBSCAN, Gaussian Mixtures
- density threshold, Using Gaussian Mixtures for Anomaly Detection
- depth concatenation layer, GoogLeNet
- depthwise separable convolution layer, Xception
- deque, Implementing Deep Q-Learning
- describe(), Take a Quick Look at the Data Structure
- development set (dev set), Hyperparameter Tuning and Model Selection
- differencing, time series forecasting, Forecasting a Time Series, Forecasting a Time Series, The ARMA Model Family
- diffusion models, Autoencoders, GANs, and Diffusion Models, Diffusion Models-Diffusion Models
- dilated filter, Semantic Segmentation
- dilation rate, Semantic Segmentation
- dimensionality reduction, Unsupervised learning, Dimensionality Reduction-Other Dimensionality Reduction Techniques
  - approaches to, Main Approaches for Dimensionality Reduction-Manifold Learning
  - autoencoders, Autoencoders, GANs, and Diffusion Models, Efficient Data Representations-Performing PCA with an Undercomplete Linear Autoencoder
  - choosing the right number of dimensions, Choosing the Right Number of Dimensions
  - clustering, Clustering Algorithms: k-means and DBSCAN
  - curse of dimensionality, The Curse of Dimensionality-The Curse of Dimensionality
  - for data visualization, Dimensionality Reduction
  - information loss from, Dimensionality Reduction
  - Isomap, Other Dimensionality Reduction Techniques
  - linear discriminant analysis, Other Dimensionality Reduction Techniques

  - LLE, LLE-LLE
  - multidimensional scaling, Other Dimensionality Reduction Techniques
  - PCA (see principal component analysis)
  - random projection algorithm, Random Projection-Random Projection
  - t-distributed stochastic neighbor embedding, Other Dimensionality Reduction Techniques, Visualizing the Fashion MNIST Dataset
- discount factor γ in reward system, Evaluating Actions: The Credit Assignment Problem
- discounted rewards, Evaluating Actions: The Credit Assignment Problem, Policy Gradients
- Discretization layer, The Discretization Layer
- discriminator, GAN, Autoencoders, GANs, and Diffusion Models, Generative Adversarial Networks-The Difficulties of Training GANs
- DistilBERT model, An Avalanche of Transformer Models, Hugging Face's Transformers Library-Hugging Face's Transformers Library
- distillation, An Avalanche of Transformer Models
- distribution strategies API, Training at Scale Using the Distribution Strategies API
- Docker container, Installing and starting TensorFlow Serving
- dot product, Attention Mechanisms
- Dota 2, Overview of Some Popular RL Algorithms
- Double DQN, Double DQN
- downloading data, Saving Your Code Changes and Your Data-Saving Your Code Changes and Your Data
- DQNs (deep Q-networks) (see Q-learning algorithm)
- drop(), Prepare the Data for Machine Learning Algorithms
- dropna(), Clean the Data
- Dropout, Denoising Autoencoders
- dropout rate, Dropout
- dropout regularization, Dropout-Dropout
- dual numbers, Forward-Mode Autodiff
- dual problem, The Dual Problem-Kernelized SVMs
- dueling DQN (DDQN), Dueling DQN
- dummy attributes, Handling Text and Categorical Attributes
- dying ReLU problem, Better Activation Functions
- dynamic models with subclassing API, Using the Subclassing API to Build Dynamic Models-Using the Subclassing API to Build Dynamic

Models
- dynamic programming, Markov Decision Processes

E

- eager execution (eager mode), AutoGraph and Tracing
- early stopping regularization, Early Stopping-Early Stopping, Gradient Boosting, Forecasting Using a Linear Model
- edge computing, Deploying a Model to a Mobile or Embedded Device
- efficient data representations, autoencoders, Efficient Data Representations-Efficient Data Representations
- elastic net, Elastic Net Regression
- EllipticEnvelope, Other Algorithms for Anomaly and Novelty Detection
- ELMo (Embeddings from Language Models), Reusing Pretrained Embeddings and Language Models
- ELU (exponential linear unit), ELU and SELU-GELU, Swish, and Mish
- EM (expectation-maximization), Gaussian Mixtures
- embedded device, deploying model to, Deploying a Model to a Mobile or Embedded Device-Deploying a Model to a Mobile or Embedded Device
- embedded Reber grammars, Exercises
- embedding matrix, Encoding Categorical Features Using Embeddings, An Encoder–Decoder Network for Neural Machine Translation
- embedding size, An Encoder–Decoder Network for Neural Machine Translation, Positional encodings
- embeddings, Handling Text and Categorical Attributes
  - encoding categorical features using, Encoding Categorical Features Using Embeddings-Encoding Categorical Features Using Embeddings
  - reusing pretrained, Reusing Pretrained Embeddings and Language Models-Reusing Pretrained Embeddings and Language Models
  - sentiment analysis, Sentiment Analysis
- Embeddings from Language Models (ELMo), Reusing Pretrained Embeddings and Language Models
- encoder, Efficient Data Representations
  - (see also autoencoders)
- encoder–decoder models, Input and Output Sequences, Natural Language Processing with RNNs and Attention, An Encoder–Decoder Network for Neural Machine Translation-Beam Search
  - (see also attention mechanisms)

- end-to-end ML project exercise, End-to-End Machine Learning Project-Try It Out!
  - building the model, Look at the Big Picture-Check the Assumptions
  - discovering and visualizing data, Explore and Visualize the Data to Gain Insights-Experiment with Attribute Combinations
  - fine-tuning your model, Fine-Tune Your Model-Evaluate Your System on the Test Set
  - getting the data, Get the Data-Create a Test Set
  - preparing data for ML algorithms, Prepare the Data for Machine Learning Algorithms-Transformation Pipelines
  - real data, advantages of working with, Working with Real Data-Working with Real Data
  - selecting and training a model, Train and Evaluate on the Training Set-Train and Evaluate on the Training Set
- endpoint, deploying model on GCP, Creating a Prediction Service on Vertex AI
- ensemble learning, Ensemble Learning and Random Forests-Stacking
  - bagging and pasting, Bagging and Pasting-Random Patches and Random Subspaces
  - boosting, Boosting-Histogram-Based Gradient Boosting
  - cross-validation, Better Evaluation Using Cross-Validation
  - fine-tuning the system, Ensemble Methods
  - random forests (see random forests)
  - stacking, Stacking-Stacking
  - voting classifiers, Voting Classifiers-Voting Classifiers
- entailment, An Avalanche of Transformer Models
- entropy impurity measure, Gini Impurity or Entropy?
- environments, reinforcement learning, Learning to Optimize Rewards, Introduction to OpenAI Gym-Introduction to OpenAI Gym
- epochs, Batch Gradient Descent
- equalized learning rate, GAN, Progressive Growing of GANs
- equivariance, Pooling Layers
- error analysis, classification, Error Analysis-Error Analysis
- estimated versus actual probabilities, The ROC Curve
- estimators, Clean the Data, Transformation Pipelines, Voting Classifiers
- Euclidean norm, Select a Performance Measure
- event files, TensorBoard, Using TensorBoard for Visualization
- Example protobuf, TensorFlow Protobufs

- exclusive or (XOR) problem, The Perceptron
- exemplars, affinity propagation, Other Clustering Algorithms
- expectation-maximization (EM), Gaussian Mixtures
- experience replay, The Difficulties of Training GANs
- explainability, attention mechanisms, Vision Transformers
- explained variance ratio, Explained Variance Ratio
- explained variance, plotting, Choosing the Right Number of Dimensions-Choosing the Right Number of Dimensions
- exploding gradients, The Vanishing/Exploding Gradients Problems
  - (see also vanishing and exploding gradients)
- exploration policies, Temporal Difference Learning, Exploration Policies
- exploration/exploitation dilemma, reinforcement learning, Neural Network Policies
- exponential linear unit (ELU), ELU and SELU-GELU, Swish, and Mish
- exponential scheduling, Learning Rate Scheduling, Learning Rate Scheduling
- export_graphviz(), Training and Visualizing a Decision Tree
- extra-trees, random forest, Extra-Trees
- extremely randomized trees ensemble (extra-trees), Extra-Trees

F

- face-recognition classifier, Multilabel Classification
- false negatives, confusion matrix, Confusion Matrices
- false positive rate (FPR) or fall-out, The ROC Curve
- false positives, confusion matrix, Confusion Matrices
- fan-in/fan-out, Glorot and He Initialization
- fast-MCD, Other Algorithms for Anomaly and Novelty Detection
- FCNs (fully convolutional networks), Fully Convolutional Networks-Fully Convolutional Networks, Semantic Segmentation
- feature engineering, Irrelevant Features, Clustering Algorithms: k-means and DBSCAN
- feature extraction, Unsupervised learning, Irrelevant Features
- feature maps, Stacking Multiple Feature Maps-Stacking Multiple Feature Maps, Implementing Convolutional Layers with Keras, ResNet, SENet
- feature scaling, Feature Scaling and Transformation-Feature Scaling and Transformation, Gradient Descent, Linear SVM Classification

- feature selection, Irrelevant Features, Analyzing the Best Models and Their Errors, Lasso Regression, Feature Importance
- feature vectors, Select a Performance Measure, Linear Regression, Under the Hood of Linear SVM Classifiers
- features, Supervised learning
- federated learning, Running a Model in a Web Page
- feedforward neural network (FNN), The Multilayer Perceptron and Backpropagation, Recurrent Neurons and Layers
- fetch_openml(), MNIST
- fillna(), Clean the Data
- filters, convolutional layers, Filters, Implementing Convolutional Layers with Keras, CNN Architectures, Xception
- first moment (mean of gradient), Adam
- first-order partial derivatives (Jacobians), AdamW
- fit()
  - and custom transformers, Custom Transformers, Transformation Pipelines
  - data cleaning, Clean the Data
  - versus partial_fit(), Stochastic Gradient Descent
  - using only with training set, Feature Scaling and Transformation
- fitness function, Model-based learning and a typical machine learning workflow
- fit_transform(), Clean the Data, Feature Scaling and Transformation, Custom Transformers, Transformation Pipelines
- fixed Q-value targets, Fixed Q-value Targets
- flat dataset, Preparing the Data for Machine Learning Models
- flowers dataset, Pretrained Models for Transfer Learning-Pretrained Models for Transfer Learning
- FNN (feedforward neural network), The Multilayer Perceptron and Backpropagation, Recurrent Neurons and Layers
- folds, Better Evaluation Using Cross-Validation, MNIST, Measuring Accuracy Using Cross-Validation, Measuring Accuracy Using Cross-Validation
- forecasting time series (see time series data)
- forget gate, LSTM, LSTM cells
- forward pass, in backpropagation, The Multilayer Perceptron and Backpropagation
- forward process, diffusion model, Diffusion Models-Diffusion Models
- FPR (false positive rate) or fall-out, The ROC Curve

- from_predictions(), Error Analysis
- full gradient descent, Batch Gradient Descent
- fully connected layer, The Perceptron, The Architecture of the Visual Cortex, Memory Requirements
- fully convolutional networks (FCNs), Fully Convolutional Networks-Fully Convolutional Networks, Semantic Segmentation
- function definition (FuncDef), TF Functions and Concrete Functions
- function graph (FuncGraph), TF Functions and Concrete Functions
- functional API, complex models with, Building Complex Models Using the Functional API-Building Complex Models Using the Functional API
- FunctionTransformer, Custom Transformers
- F_1 score, Precision and Recall

G

- game play (see reinforcement learning)
- gamma (γ) value, Gaussian RBF Kernel
- GANs (see generative adversarial networks)
- gate controllers, LSTM, LSTM cells
- gated activation units, WaveNet
- gated recurrent unit (GRU) cell, GRU cells-GRU cells, Masking
- Gaussian distribution, Feature Scaling and Transformation, Training and Cost Function, Variational Autoencoders-Variational Autoencoders
- Gaussian mixture model (GMM), Gaussian Mixtures-Other Algorithms for Anomaly and Novelty Detection
  - anomaly detection, Using Gaussian Mixtures for Anomaly Detection-Using Gaussian Mixtures for Anomaly Detection
  - Bayesian Gaussian mixtures, Bayesian Gaussian Mixture Models
  - fast-MCD, Other Algorithms for Anomaly and Novelty Detection
  - inverse_transform() with PCA, Other Algorithms for Anomaly and Novelty Detection
  - isolation forest, Other Algorithms for Anomaly and Novelty Detection
  - and k-means limitations, Limits of k-means
  - local outlier factor, Other Algorithms for Anomaly and Novelty Detection
  - one-class SVM, Other Algorithms for Anomaly and Novelty Detection
  - selecting number of clusters, Selecting the Number of Clusters-Selecting the Number of Clusters

- Gaussian process, Fine-Tuning Neural Network Hyperparameters
- Gaussian RBF kernel, Custom Transformers, Gaussian RBF Kernel-SVM Classes and Computational Complexity, Kernelized SVMs
- GBRT (gradient boosted regression trees), Gradient Boosting, Gradient Boosting-Histogram-Based Gradient Boosting
- GCP (Google Cloud Platform), Creating a Prediction Service on Vertex AI-Creating a Prediction Service on Vertex AI
- GCS (Google Cloud Storage), Creating a Prediction Service on Vertex AI
- GD (see gradient descent)
- GELU, GELU, Swish, and Mish-GELU, Swish, and Mish
- generalization error, Testing and Validating-Hyperparameter Tuning and Model Selection
- generative adversarial networks (GANs), Generative Adversarial Networks-StyleGANs
  - deep convolutional, Deep Convolutional GANs-Deep Convolutional GANs
  - progressive growing of, Progressive Growing of GANs-Progressive Growing of GANs
  - StyleGANs, StyleGANs-StyleGANs
  - training difficulties, The Difficulties of Training GANs-The Difficulties of Training GANs
- generative autoencoders, Variational Autoencoders
- generative models, Autoencoders, GANs, and Diffusion Models
  - (see also Gaussian mixture model)
- generator, GAN, Autoencoders, GANs, and Diffusion Models, Generative Adversarial Networks-The Difficulties of Training GANs
- genetic algorithm, Policy Search
- geodesic distance, Other Dimensionality Reduction Techniques
- geographic data, visualizing, Visualizing Geographical Data-Visualizing Geographical Data
- get_dummies(), Handling Text and Categorical Attributes
- get_feature_names_out(), Custom Transformers
- get_params(), Custom Transformers
- GINI impurity, Making Predictions, Gini Impurity or Entropy?
- global average pooling layer, Implementing Pooling Layers with Keras
- global versus local minimum, gradient descent, Gradient Descent
- Glorot initialization, Glorot and He Initialization-Glorot and He Initialization
- GMM (see Gaussian mixture model)

- Google Cloud Platform (GCP), Creating a Prediction Service on Vertex AI-Creating a Prediction Service on Vertex AI
- Google Cloud Storage (GCS), Creating a Prediction Service on Vertex AI
- Google Colab, Running the Code Examples Using Google Colab-Running the Code Examples Using Google Colab
- Google Vertex AI (see Vertex AI)
- GoogLeNet, GoogLeNet-GoogLeNet, Xception
- GPU implementation, Mini-Batch Gradient Descent, Using GPUs to Speed Up Computations-Parallel Execution Across Multiple Devices, Training at Scale Using the Distribution Strategies API
  - getting your own GPU, Getting Your Own GPU-Getting Your Own GPU
  - managing RAM, Managing the GPU RAM-Managing the GPU RAM
  - operations handling, Parallel Execution Across Multiple Devices
  - parallel execution across multiple devices, Parallel Execution Across Multiple Devices-Parallel Execution Across Multiple Devices
  - placing operations and variables on devices, Placing Operations and Variables on Devices-Placing Operations and Variables on Devices
- gradient ascent, Policy Search
- gradient boosted regression trees (GBRT), Gradient Boosting, Gradient Boosting-Histogram-Based Gradient Boosting
- gradient boosting, Gradient Boosting-Gradient Boosting
- gradient clipping, Gradient Clipping
- gradient descent (GD), Training Models, Gradient Descent-Mini-Batch Gradient Descent
  - algorithm comparisons, Mini-Batch Gradient Descent
  - batch gradient descent, Batch Gradient Descent-Batch Gradient Descent, Ridge Regression
  - local versus global minimum, Gradient Descent
  - mini-batch gradient descent, Mini-Batch Gradient Descent-Mini-Batch Gradient Descent
  - minimizing hinge loss, Under the Hood of Linear SVM Classifiers
  - versus momentum optimization, Momentum
  - with optimizers, Faster Optimizers-AdamW
  - shuffling data, Shuffling the Data
  - stochastic gradient descent, Stochastic Gradient Descent-Stochastic Gradient Descent
- gradient tree boosting, Gradient Boosting

- gradients
  - autodiff for computing, Computing Gradients Using Autodiff-Computing Gradients Using Autodiff
  - bandwidth saturation issue, Bandwidth saturation
  - PG algorithm, Policy Search, Policy Gradients-Policy Gradients
  - stale, Asynchronous updates
  - unstable (see vanishing and exploding gradients)
- graph mode, AutoGraph and Tracing
- graphical processing units (see GPU implementation)
- graphs and functions, TensorFlow, A Quick Tour of TensorFlow, TensorFlow Functions and Graphs-TF Function Rules, TensorFlow Graphs-Using TF Functions with Keras (or Not)
- Graphviz, Training and Visualizing a Decision Tree
- greedy algorithm, CART as, The CART Training Algorithm
- greedy decoding, Generating Fake Shakespearean Text
- greedy layer-wise pretraining, Unsupervised Pretraining, Training One Autoencoder at a Time, Progressive Growing of GANs
- grid search, Grid Search-Grid Search
- GridSearchCV, Grid Search-Grid Search
- gRPC API, querying through, Querying TF Serving through the gRPC API
- GRU (gated recurrent unit) cell, GRU cells-GRU cells, Masking

H

- hard clustering, k-means
- hard margin classification, Soft Margin Classification, Under the Hood of Linear SVM Classifiers
- hard voting classifiers, Voting Classifiers
- harmonic mean, Precision and Recall
- hashing collision, The StringLookup Layer
- Hashing layer, The Hashing Layer
- hashing trick, The StringLookup Layer
- HDBSCAN (hierarchical DBSCAN), DBSCAN
- He initialization, Glorot and He Initialization-Glorot and He Initialization
- Heaviside step function, The Perceptron
- heavy tail, feature distribution, Feature Scaling and Transformation
- Hebb's rule, The Perceptron
- Hebbian learning, The Perceptron

- Hessians, AdamW
- hidden layers
  - neurons per layer, Number of Neurons per Hidden Layer
  - number of, Number of Hidden Layers
  - stacked autoencoders, Stacked Autoencoders-Training One Autoencoder at a Time
- hierarchical clustering, Unsupervised learning
- hierarchical DBSCAN (HDBSCAN), DBSCAN
- high variance, with decision trees, Decision Trees Have a High Variance
- hinge loss function, Under the Hood of Linear SVM Classifiers
- histogram-based gradient boosting (HGB), Histogram-Based Gradient Boosting-Histogram-Based Gradient Boosting
- histograms, Take a Quick Look at the Data Structure
- hold-out sets, Testing and Validating
- holdout validation, Hyperparameter Tuning and Model Selection
- housing dataset, Working with Real Data-Check the Assumptions
- Huber loss, Regression MLPs, Custom Loss Functions, Custom Metrics, Forecasting Using a Linear Model
- Hugging Face, Hugging Face's Transformers Library-Hugging Face's Transformers Library
- Hungarian algorithm, Object Tracking
- Hyperband tuner, Fine-Tuning Neural Network Hyperparameters
- hyperbolic tangent (htan), The Multilayer Perceptron and Backpropagation, Fighting the Unstable Gradients Problem
- hyperparameters, Overfitting the Training Data, Fine-Tuning Neural Network Hyperparameters-Learning Rate, Batch Size, and Other Hyperparameters
  - activation function, Learning Rate, Batch Size, and Other Hyperparameters
  - batch size, Learning Rate, Batch Size, and Other Hyperparameters
  - CART algorithm, The CART Training Algorithm
  - convolutional layers, Implementing Convolutional Layers with Keras
  - in custom transformations, Custom Transformers
  - decision tree, Gradient Boosting
  - dimensionality reduction, Choosing the Right Number of Dimensions
  - gamma (γ) value, Gaussian RBF Kernel

  - GAN challenges, The Difficulties of Training GANs
  - Keras Tuner, Hyperparameter Tuning on Vertex AI
  - learning rate, Gradient Descent, Learning Rate, Batch Size, and Other Hyperparameters
  - momentum β, Momentum
  - Monte Carlo samples, Monte Carlo (MC) Dropout
  - neurons per hidden layer, Number of Neurons per Hidden Layer
  - and normalization, Feature Scaling and Transformation
  - number of hidden layers, Number of Hidden Layers
  - number of iterations, Learning Rate, Batch Size, and Other Hyperparameters
  - optimizer, Learning Rate, Batch Size, and Other Hyperparameters
  - PG algorithms, Policy Gradients
  - preprocessor and model interaction, Grid Search
  - randomized search, Fine-Tuning Neural Network Hyperparameters-Fine-Tuning Neural Network Hyperparameters
  - saving along with model, Custom Activation Functions, Initializers, Regularizers, and Constraints
  - SGDClassifier, SVM Classes and Computational Complexity
  - subsample, Gradient Boosting
  - SVM classifiers with polynomial kernel, Polynomial Kernel
  - tolerance (ε), SVM Classes and Computational Complexity
  - tuning of, Hyperparameter Tuning and Model Selection-Hyperparameter Tuning and Model Selection, Grid Search-Grid Search, Evaluate Your System on the Test Set, Training and evaluating the model, Hyperparameter Tuning on Vertex AI-Hyperparameter Tuning on Vertex AI
- hypothesis, Select a Performance Measure
- hypothesis boosting (see boosting)
- hypothesis function, Linear Regression

I

- identity matrix, Ridge Regression
- IID (see independent and identically distributed)
- image generation, Semantic Segmentation, StyleGANs, Diffusion Models
- image segmentation, Clustering Algorithms: k-means and DBSCAN, Using Clustering for Image Segmentation-Using Clustering for Image Segmentation

- images, classifying and generating, Examples of Applications
  - autoencoders (see autoencoders)
  - CNNs (see convolutional neural networks)
  - diffusion models, Diffusion Models-Diffusion Models
  - generating with GANs, Generative Adversarial Networks-StyleGANs
  - implementing MLPs, Building an Image Classifier Using the Sequential API-Using the model to make predictions
  - labels, Classification and Localization
  - loading and preprocessing data, Image Preprocessing Layers
  - representative images, Using Clustering for Semi-Supervised Learning
  - semantic segmentation, Using Clustering for Image Segmentation
  - tuning hyperparameters, Fine-Tuning Neural Network Hyperparameters
- importance sampling (IS), Prioritized Experience Replay
- impurity measures, Making Predictions, Gini Impurity or Entropy?
- imputation, Clean the Data
- incremental learning, Online learning
- incremental PCA (IPCA), Incremental PCA-Incremental PCA
- independent and identically distributed (IID), training instances as, Stochastic Gradient Descent
- inductive bias, Vision Transformers
- inertia, model, Centroid initialization methods, Accelerated k-means and mini-batch k-means
- inference, Model-based learning and a typical machine learning workflow
- info(), Take a Quick Look at the Data Structure
- information theory, Softmax Regression, Gini Impurity or Entropy?
- inliers, Unsupervised Learning Techniques
- input and output sequences, RNNs, Input and Output Sequences-Input and Output Sequences
- input gate, LSTM, LSTM cells
- input layer, neural network, The Perceptron, Creating the model using the sequential API, Building Complex Models Using the Functional API
  - (see also hidden layers)
- input signature, TF Functions and Concrete Functions
- instance segmentation, Using Clustering for Image Segmentation, Semantic Segmentation

- instance-based learning, Instance-based learning, Model-based learning and a typical machine learning workflow
- inter-op thread pool, Parallel Execution Across Multiple Devices
- intercept term constant, Linear Regression
- interleaving lines from multiple files, Interleaving Lines from Multiple Files-Interleaving Lines from Multiple Files
- interpretable ML, Making Predictions
- invariance, max pooling layer, Pooling Layers
- inverse_transform(), Feature Scaling and Transformation, Custom Transformers, PCA for Compression, Other Algorithms for Anomaly and Novelty Detection
- IPCA (incremental PCA), Incremental PCA-Incremental PCA
- iris dataset, Decision Boundaries
- irreducible error, Learning Curves
- IS (importance sampling), Prioritized Experience Replay
- isolation forest, Other Algorithms for Anomaly and Novelty Detection
- Isomap, Other Dimensionality Reduction Techniques
- isotropic noise, Diffusion Models
- IterativeImputer, Clean the Data

J

- Jacobians, AdamW
- joblib library, Launch, Monitor, and Maintain Your System-Launch, Monitor, and Maintain Your System
- JSON Lines, Running Batch Prediction Jobs on Vertex AI, Running Batch Prediction Jobs on Vertex AI
- Jupyter, Running the Code Examples Using Google Colab

K

- k-fold cross-validation, Better Evaluation Using Cross-Validation, Measuring Accuracy Using Cross-Validation, Measuring Accuracy Using Cross-Validation
- k-means algorithm, Custom Transformers, k-means-Limits of k-means
  - accelerated k-means, Accelerated k-means and mini-batch k-means
  - centroid initialization methods, Centroid initialization methods-Centroid initialization methods
  - finding optimal number of clusters, Finding the optimal number of clusters-Finding the optimal number of clusters

    - limitations of, Limits of k-means
    - mini-batch k-means, Accelerated k-means and mini-batch k-means
    - workings of, The k-means algorithm-The k-means algorithm
- k-means++, Centroid initialization methods
- k-nearest neighbors regression, Model-based learning and a typical machine learning workflow
- Kaiming initialization, Glorot and He Initialization
- Kalman Filters, Object Tracking
- Keras API, Objective and Approach, A Quick Tour of TensorFlow
    - (see also artificial neural networks)
    - and accessing TensorFlow API directly, Image Preprocessing Layers
    - activation function support, Leaky ReLU, ELU and SELU, GELU, Swish, and Mish
    - convolutional layer implementation, Implementing Convolutional Layers with Keras-Implementing Convolutional Layers with Keras
    - custom functions in, TensorFlow Functions and Graphs
    - gradient clipping, Gradient Clipping
    - image preprocessing layers, Image Preprocessing Layers
    - implementing MLPs with, Implementing MLPs with Keras-Using TensorBoard for Visualization
    - initialization handling, Glorot and He Initialization
    - initializers, Creating the model using the sequential API
    - layer preprocessing, Keras Preprocessing Layers-Using Pretrained Language Model Components
    - learning rate scheduling, Learning Rate Scheduling-Learning Rate Scheduling
    - loading a dataset, Building an Image Classifier Using the Sequential API
    - PG algorithm, Policy Gradients-Policy Gradients
    - pool layer implementation, Implementing Pooling Layers with Keras-Implementing Pooling Layers with Keras
    - pretrained CNN models, Using Pretrained Models from Keras-Using Pretrained Models from Keras
    - ResNet-34 CNN with, Implementing a ResNet-34 CNN Using Keras
    - saving models, Exporting SavedModels, Deploying a Model to a Mobile or Embedded Device
    - stacked encoder implementation, Implementing a Stacked Autoencoder Using Keras

- tf.data API dataset, Using the Dataset with Keras-Using the Dataset with Keras
- tf.keras library, Using Keras to load the dataset
- tf.keras.activations.get(), Custom Layers
- tf.keras.activations.relu(), Creating the model using the sequential API
- tf.keras.applications module, Using Pretrained Models from Keras
- tf.keras.applications.xception.preprocess_input(), Pretrained Models for Transfer Learning
- tf.keras.backend module, Tensors and Operations
- tf.keras.callbacks.EarlyStopping, Using Callbacks
- tf.keras.callbacks.LearningRateScheduler, Learning Rate Scheduling
- tf.keras.callbacks.ModelCheckpoint, Using Callbacks
- tf.keras.callbacks.TensorBoard, Using TensorBoard for Visualization, Masking
- tf.keras.datasets.imdb.load_data(), Sentiment Analysis
- tf.keras.initializers.VarianceScaling, Glorot and He Initialization
- tf.keras.layers.ActivityRegularization, Sparse Autoencoders
- tf.keras.layers.AdditiveAttention, Attention Mechanisms
- tf.keras.layers.Attention, Attention Mechanisms
- tf.keras.layers.AvgPool2D, Implementing Pooling Layers with Keras
- tf.keras.layers.BatchNormalization, Batch Normalization, Implementing batch normalization with Keras-Implementing batch normalization with Keras, Fighting the Unstable Gradients Problem
- tf.keras.layers.Bidirectional, Bidirectional RNNs
- tf.keras.layers.CategoryEncoding, The CategoryEncoding Layer
- tf.keras.layers.CenterCrop, Image Preprocessing Layers
- tf.keras.layers.Concatenate, Building Complex Models Using the Functional API, Building Complex Models Using the Functional API, GoogLeNet
- tf.keras.layers.Conv1D, Semantic Segmentation, Masking
- tf.keras.layers.Conv2D, Implementing Convolutional Layers with Keras
- tf.keras.layers.Conv2DTranspose, Semantic Segmentation
- tf.keras.layers.Conv3D, Semantic Segmentation
- tf.keras.layers.Dense, Building Complex Models Using the Functional API, Building Complex Models Using the Functional API,

Custom Layers-Custom Layers, Encoding Categorical Features Using Embeddings, Tying Weights, Variational Autoencoders

- tf.keras.layers.Discretization, The Discretization Layer
- tf.keras.layers.Dropout, Dropout
- tf.keras.layers.Embedding, Encoding Categorical Features Using Embeddings, Building and Training the Char-RNN Model, Sentiment Analysis, Positional encodings
- tf.keras.layers.GlobalAvgPool2D, Implementing Pooling Layers with Keras
- tf.keras.layers.GRU, GRU cells
- tf.keras.layers.GRUCell, GRU cells
- tf.keras.layers.Hashing, The Hashing Layer
- tf.keras.layers.Input, Building Complex Models Using the Functional API
- tf.keras.layers.Lambda, Custom Layers, Building and Training the Char-RNN Model
- tf.keras.layers.LayerNormalization, Fighting the Unstable Gradients Problem
- tf.keras.layers.LeakyReLU, Leaky ReLU
- tf.keras.layers.LSTM, LSTM cells
- tf.keras.layers.Masking, Masking
- tf.keras.layers.MaxPool2D, Implementing Pooling Layers with Keras
- tf.keras.layers.MultiHeadAttention, Multi-head attention
- tf.keras.layers.Normalization, Building a Regression MLP Using the Sequential API, Building Complex Models Using the Functional API, The Normalization Layer-The Normalization Layer
- tf.keras.layers.PReLU, Leaky ReLU
- tf.keras.layers.Rescaling, Image Preprocessing Layers
- tf.keras.layers.Resizing, Image Preprocessing Layers, Using Pretrained Models from Keras
- tf.keras.layers.RNN, LSTM cells
- tf.keras.layers.SeparableConv2D, Xception
- tf.keras.layers.StringLookup, The StringLookup Layer
- tf.keras.layers.TextVectorization, Text Preprocessing-Text Preprocessing, Creating the Training Dataset, Building and Training the Char-RNN Model, Sentiment Analysis, Sentiment Analysis-An Encoder–Decoder Network for Neural Machine Translation

- tf.keras.layers.TimeDistributed, Forecasting Using a Sequence-to-Sequence Model
- tf.keras.losses.Huber, Custom Loss Functions
- tf.keras.losses.kullback_leibler_divergence(), Sparse Autoencoders
- tf.keras.losses.Loss, Saving and Loading Models That Contain Custom Components
- tf.keras.losses.sparse_categorical_crossentropy(), Compiling the model, CNN Architectures, Building and Training the Char-RNN Model, An Encoder–Decoder Network for Neural Machine Translation, Hugging Face's Transformers Library
- tf.keras.metrics.MeanIoU, Classification and Localization
- tf.keras.metrics.Metric, Custom Metrics
- tf.keras.metrics.Precision, Custom Metrics
- tf.keras.Model, Building Complex Models Using the Functional API
- tf.keras.models.clone_model(), Using the Subclassing API to Build Dynamic Models, Transfer Learning with Keras
- tf.keras.models.load_model(), Saving and Restoring a Model, Saving and Loading Models That Contain Custom Components-Saving and Loading Models That Contain Custom Components, Custom Models, Training at Scale Using the Distribution Strategies API
- tf.keras.optimizers.Adam, Building a Regression MLP Using the Sequential API, AdamW
- tf.keras.optimizers.Adamax, AdamW
- tf.keras.optimizers.experimental.AdamW, AdamW
- tf.keras.optimizers.Nadam, AdamW
- tf.keras.optimizers.schedules, Learning Rate Scheduling
- tf.keras.optimizers.SGD, Momentum
- tf.keras.preprocessing.image.ImageDataGenerator, Pretrained Models for Transfer Learning
- tf.keras.regularizers.l1_l2(), ℓ1 and ℓ2 Regularization
- tf.keras.Sequential, Creating the model using the sequential API, Building a Regression MLP Using the Sequential API, CNN Architectures
- tf.keras.utils.get_file(), Creating the Training Dataset
- tf.keras.utils.set_random_seed(), Creating the model using the sequential API
- tf.keras.utils.timeseries_dataset_from_array(), Preparing the Data for Machine Learning Models, Preparing the Data for Machine Learning Models

    - tf.keras.utils.to_categorical(), Compiling the model
    - time series forecasting for RNN, Forecasting Using a Simple RNN-Forecasting Using a Deep RNN
    - transfer learning with, Transfer Learning with Keras-Transfer Learning with Keras
    - using TF functions (or not), Using TF Functions with Keras (or Not)
- Keras session, Creating the model using the sequential API
- Keras Tuner, Hyperparameter Tuning on Vertex AI
- kernel trick, Polynomial Kernel-SVM Classes and Computational Complexity, Kernelized SVMs-Kernelized SVMs
- kernelized SVMs, Kernelized SVMs-Kernelized SVMs
- kernels (convolution kernels), Filters, CNN Architectures
- kernels (runtimes), Running the Code Examples Using Google Colab
- KL (Kullback-Leibler) divergence, Softmax Regression, Sparse Autoencoders
- KLDivergenceRegularizer, Sparse Autoencoders
- KMeans, Custom Transformers
- KNeighborsClassifier, Multilabel Classification, Multioutput Classification
- KNNImputer, Clean the Data
- Kullback-Leibler (KL) divergence, Softmax Regression, Sparse Autoencoders

L

- label propagation, Using Clustering for Semi-Supervised Learning-Using Clustering for Semi-Supervised Learning
- labels, Frame the Problem
    - in clustering, k-means
    - image classification, Classification and Localization
    - supervised learning, Supervised learning
    - unlabeled data issue, Unsupervised Pretraining Using Stacked Autoencoders
- landmarks, Similarity Features
- language models, Generating Shakespearean Text Using a Character RNN
    - (see also natural language processing)
- large margin classification, Linear SVM Classification
- Lasso, Lasso Regression
- lasso regression, Lasso Regression-Lasso Regression

- latent diffusion models, Diffusion Models
- latent loss, Variational Autoencoders
- latent representation of inputs, Vision Transformers, Autoencoders, GANs, and Diffusion Models, Efficient Data Representations, Deep Convolutional GANs
- latent space, Variational Autoencoders
- law of large numbers, Voting Classifiers
- layer normalization, Exercises, Processing Sequences Using RNNs and CNNs, Fighting the Unstable Gradients Problem
- LDA (linear discriminant analysis), Other Dimensionality Reduction Techniques
- leaf node, decision tree, Making Predictions, Estimating Class Probabilities, Regularization Hyperparameters
- leakyReLU, Leaky ReLU-Leaky ReLU
- learning curves, overfit or underfit analysis, Learning Curves-Learning Curves
- learning rate, Online learning, Gradient Descent, Batch Gradient Descent, Stochastic Gradient Descent, Learning Rate, Batch Size, and Other Hyperparameters, Q-Learning
- learning rate schedules, Learning Rate Scheduling-Learning Rate Scheduling
- learning schedules, Stochastic Gradient Descent
- LeCun initialization, Glorot and He Initialization
- LeNet-5, The Architecture of the Visual Cortex, LeNet-5
- Levenshtein distance, Gaussian RBF Kernel
- liblinear library, SVM Classes and Computational Complexity
- libsvm library, SVM Classes and Computational Complexity
- life satisfaction dataset, Model-based learning and a typical machine learning workflow
- likelihood function, Selecting the Number of Clusters-Selecting the Number of Clusters
- linear discriminant analysis (LDA), Other Dimensionality Reduction Techniques
- linear models
  - forecasting time series, Forecasting Using a Linear Model
  - linear regression (see linear regression)
  - regularized, Regularized Linear Models-Early Stopping
  - SVM, Linear SVM Classification-Soft Margin Classification

    - training and running example, Model-based learning and a typical machine learning workflow
- linear regression, Model-based learning and a typical machine learning workflow-Model-based learning and a typical machine learning workflow, Linear Regression-Mini-Batch Gradient Descent
    - comparison of algorithms, Mini-Batch Gradient Descent
    - computational complexity, Computational Complexity
    - gradient descent in, Gradient Descent-Mini-Batch Gradient Descent
    - learning curves in, Learning Curves-Learning Curves
    - Normal equation, The Normal Equation-The Normal Equation
    - regularizing models (see regularization)
    - ridge regression, Ridge Regression-Ridge Regression, Elastic Net Regression
    - training set evaluation, Train and Evaluate on the Training Set
    - using stochastic gradient descent, Stochastic Gradient Descent
- linear SVM classification, Linear SVM Classification-Soft Margin Classification, Under the Hood of Linear SVM Classifiers-Kernelized SVMs
- linear threshold units (LTUs), The Perceptron
- linearly separable, SVM classes, Linear SVM Classification-Linear SVM Classification, Similarity Features
- LinearRegression, Clean the Data, Feature Scaling and Transformation, Train and Evaluate on the Training Set, The Normal Equation, Computational Complexity, Polynomial Regression
- LinearSVC, Soft Margin Classification, SVM Classes and Computational Complexity, Under the Hood of Linear SVM Classifiers
- LinearSVR, SVM Regression-SVM Regression
- Lipschitz continuous, derivative as, Gradient Descent
- LLE (locally linear embedding), LLE-LLE
- loading and preprocessing data, Loading and Preprocessing Data with TensorFlow-The TensorFlow Datasets Project
    - image preprocessing layers, Image Preprocessing Layers
    - layer preprocessing in Keras, Loading and Preprocessing Data with TensorFlow, Keras Preprocessing Layers-Using Pretrained Language Model Components
    - tf.data API, Loading and Preprocessing Data with TensorFlow-Using the Dataset with Keras
    - TFDS project, The TensorFlow Datasets Project-The TensorFlow Datasets Project

    - TFRecord format, Loading and Preprocessing Data with TensorFlow, The TFRecord Format-Handling Lists of Lists Using the SequenceExample Protobuf
- local outlier factor (LOF), Other Algorithms for Anomaly and Novelty Detection
- local receptive field, The Architecture of the Visual Cortex
- local response normalization (LRN), AlexNet
- local versus global minimum, gradient descent, Gradient Descent
- locality sensitive hashing (LSH), Random Projection
- localization, CNNs, Classification and Localization-Object Detection
- locally linear embedding (LLE), LLE-LLE
- LOF (local outlier factor), Other Algorithms for Anomaly and Novelty Detection
- log loss, Training and Cost Function
- log-odds function, Estimating Probabilities
- log-transformer, Custom Transformers
- logical GPU device, Managing the GPU RAM
- logistic function, Estimating Probabilities
- logistic regression, Supervised learning, Feature Scaling and Transformation, Logistic Regression-Softmax Regression
    - decision boundaries illustration, Decision Boundaries-Decision Boundaries
    - estimating probabilities, Estimating Probabilities-Estimating Probabilities
    - softmax regression model, Softmax Regression-Softmax Regression
    - training and cost function, Training and Cost Function-Training and Cost Function
- LogisticRegression, Decision Boundaries, Softmax Regression
- logit function, Estimating Probabilities
- long sequences, training RNN on, Handling Long Sequences-WaveNet
    - short-term memory problem, Tackling the Short-Term Memory Problem-WaveNet
    - unstable gradients problem, Fighting the Unstable Gradients Problem-Fighting the Unstable Gradients Problem
- long short-term memory (LSTM) cell, LSTM cells-LSTM cells, Masking, An Encoder–Decoder Network for Neural Machine Translation, Bidirectional RNNs
- loss functions

    - based on model internals, Losses and Metrics Based on Model Internals-Losses and Metrics Based on Model Internals
    - custom, Custom Loss Functions
    - versus metrics, Custom Metrics
    - output, Building Complex Models Using the Functional API
- LRN (local response normalization), AlexNet
- LSH (locality sensitive hashing), Random Projection
- LSTM (long short-term memory) cell, LSTM cells-LSTM cells, Masking, An Encoder–Decoder Network for Neural Machine Translation, Bidirectional RNNs
- LTUs (linear threshold units), The Perceptron
- Luong attention, Attention Mechanisms

M

- machine learning (ML), What Is Machine Learning?
    - application/technique examples, Examples of Applications-Examples of Applications
    - challenges of, Main Challenges of Machine Learning-Stepping Back
    - notations, Select a Performance Measure-Select a Performance Measure
    - project checklist, Look at the Big Picture
        - (see also end-to-end ML project exercise)
    - reasons for using, Why Use Machine Learning?-Why Use Machine Learning?
    - resources on, Other Resources-Other Resources
    - spam filter example, The Machine Learning Landscape-Why Use Machine Learning?
    - testing and validating, Testing and Validating-Data Mismatch
    - types of systems, Types of Machine Learning Systems-Model-based learning and a typical machine learning workflow
- MAE (mean absolute error), Select a Performance Measure
- majority-vote predictions, Exercises
- make_column_selector(), Transformation Pipelines
- make_column_transformer(), Transformation Pipelines
- make_pipeline(), Transformation Pipelines, Learning Curves
- Manhattan norm, Select a Performance Measure
- manifold hypothesis, Manifold Learning
- manifold learning, dimension reduction, Manifold Learning-Manifold Learning

- MAP (maximum a-posteriori) estimation, Selecting the Number of Clusters
- mAP (mean average precision), You Only Look Once
- MAPE (mean absolute percentage error), Forecasting a Time Series
- mapping network, StyleGANs, StyleGANs
- MapReduce, Frame the Problem
- margin violations, Soft Margin Classification, Under the Hood of Linear SVM Classifiers
- Markov chains, Markov Decision Processes
- Markov decision processes (MDPs), Markov Decision Processes-Markov Decision Processes, Exploration Policies
- mask R-CNN, Semantic Segmentation
- mask tensor, Masking
- masked language model (MLM), An Avalanche of Transformer Models
- masked multi-head attention layer, Attention Is All You Need: The Original Transformer Architecture
- masking, Masking-Masking, Multi-head attention
- Matching Engine service, Vertex AI, Creating a Prediction Service on Vertex AI
- Matplotlib, Running the Code Examples Using Google Colab
- max pooling layer, Pooling Layers, CNN Architectures
- max-norm regularization, Max-Norm Regularization
- maximization step, Gaussian mixtures, Gaussian Mixtures
- maximum a-posteriori (MAP) estimation, Selecting the Number of Clusters
- maximum likelihood estimate (MLE), Selecting the Number of Clusters
- MC (Monte Carlo) dropout regularization, Monte Carlo (MC) Dropout-Monte Carlo (MC) Dropout
- MCTS (Monte Carlo tree search), Overview of Some Popular RL Algorithms
- MDPs (Markov decision processes), Markov Decision Processes-Markov Decision Processes, Exploration Policies
- MDS (multidimensional scaling), Other Dimensionality Reduction Techniques
- mean absolute error (MAE), Select a Performance Measure
- mean absolute percentage error (MAPE), Forecasting a Time Series
- mean average precision (mAP), You Only Look Once
- mean squared error (MSE), Linear Regression, Variational Autoencoders

- mean-shift, clustering algorithms, Other Clustering Algorithms
- mean_squared_error(), Train and Evaluate on the Training Set
- measure of similarity, Instance-based learning
- memory bandwidth, GPU card, Prefetching
- memory cells (cells), RNNs, Memory Cells, Tackling the Short-Term Memory Problem-WaveNet
- memory requirements, convolutional layers, Memory Requirements
- Mercer's theorem, Kernelized SVMs
- meta learner, Stacking
- metagraphs, SavedModel, Exporting SavedModels
- min-max scaling, Feature Scaling and Transformation
- mini-batch discrimination, The Difficulties of Training GANs
- mini-batch gradient descent, Mini-Batch Gradient Descent-Mini-Batch Gradient Descent, Early Stopping
- mini-batch k-means, Accelerated k-means and mini-batch k-means
- mini-batches, Online learning, Progressive Growing of GANs
- MinMaxScaler, Feature Scaling and Transformation
- mirrored strategy, data parallelism, Data parallelism using the mirrored strategy, Training at Scale Using the Distribution Strategies API, Training a Model on a TensorFlow Cluster
- Mish activation function, GELU, Swish, and Mish
- mixing regularization, StyleGAN, StyleGANs
- ML (see machine learning)
- ML Operations (MLOps), Launch, Monitor, and Maintain Your System
- MLE (maximum likelihood estimate), Selecting the Number of Clusters
- MLM (masked language model), An Avalanche of Transformer Models
- MLPs (see multilayer perceptrons)
- MNIST dataset, MNIST-MNIST
- mobile device, deploying model to, Deploying a Model to a Mobile or Embedded Device-Deploying a Model to a Mobile or Embedded Device
- mode collapse, Vision Transformers, The Difficulties of Training GANs
- model parallelism, Model Parallelism-Model Parallelism
- model parameters, Model-based learning and a typical machine learning workflow
  - early stopping regularization, Training and Cost Function
  - in gradient descent, Gradient Descent, Batch Gradient Descent
  - linear SVM classifier mechanics, Under the Hood of Linear SVM Classifiers
  - and variable updating, Variables

    - weight matrix shape, Creating the model using the sequential API
- model rot, Batch learning
- model selection, Model-based learning and a typical machine learning workflow, Hyperparameter Tuning and Model Selection-Hyperparameter Tuning and Model Selection
- model server (see TensorFlow Serving)
- model warmup, Deploying a new model version
- model-based learning, Model-based learning and a typical machine learning workflow-Model-based learning and a typical machine learning workflow
- modes, Feature Scaling and Transformation
- momentum optimization, Momentum
- momentum β, Momentum
- Monte Carlo (MC) dropout, Monte Carlo (MC) Dropout-Monte Carlo (MC) Dropout
- Monte Carlo tree search (MCTS), Overview of Some Popular RL Algorithms
- MSE (mean squared error), Linear Regression, Variational Autoencoders
- multi-head attention layer, Attention Is All You Need: The Original Transformer Architecture, Multi-head attention-Multi-head attention
- multi-hot encoding, The CategoryEncoding Layer
- multiclass (multinomial) classification, Multiclass Classification-Multiclass Classification, Classification MLPs-Classification MLPs
- multidimensional scaling (MDS), Other Dimensionality Reduction Techniques
- multilabel classifiers, Multilabel Classification-Multilabel Classification
- multilayer perceptrons (MLPs), Introduction to Artificial Neural Networks with Keras, The Perceptron-Using TensorBoard for Visualization
    - and autoencoders, Efficient Data Representations, Performing PCA with an Undercomplete Linear Autoencoder
    - and backpropagation, The Multilayer Perceptron and Backpropagation-The Multilayer Perceptron and Backpropagation
    - callbacks, Using Callbacks-Using Callbacks
    - classification MLPs, Classification MLPs-Classification MLPs
    - complex models, Building Complex Models Using the Functional API-Building Complex Models Using the Functional API

    - dynamic models, Using the Subclassing API to Build Dynamic Models-Using the Subclassing API to Build Dynamic Models
    - image classifier, Building an Image Classifier Using the Sequential API-Using the model to make predictions
    - regression MLPs, Regression MLPs-Regression MLPs, Building a Regression MLP Using the Sequential API
    - saving and restoring a model, Saving and Restoring a Model-Saving and Restoring a Model
    - visualization with TensorBoard, Using TensorBoard for Visualization-Using TensorBoard for Visualization
- multimodal distribution, Feature Scaling and Transformation
- multimodal transformers, Vision Transformers
- multinomial logistic regression, Softmax Regression-Softmax Regression
- multioutput classifiers, Multioutput Classification-Multioutput Classification
- multiple regression, Frame the Problem
- multiplicative attention, Attention Mechanisms
- multitask classification, Building Complex Models Using the Functional API
- multivariate regression, Frame the Problem
- multivariate time series, Forecasting a Time Series, Forecasting Multivariate Time Series-Forecasting Multivariate Time Series

N

- Nadam, Nadam
- NAG (Nesterov accelerated gradient), Nesterov Accelerated Gradient, Nadam
- naive forecasting, Forecasting a Time Series
- Nash equilibrium, The Difficulties of Training GANs
- natural language processing (NLP), Natural Language Processing with RNNs and Attention-Hugging Face's Transformers Library
    - char-RNN model to generate text, Generating Shakespearean Text Using a Character RNN-Stateful RNN
    - encoder–decoder network for machine translation, An Encoder–Decoder Network for Neural Machine Translation-An Encoder–Decoder Network for Neural Machine Translation
    - machine learning examples, Examples of Applications

  - sentiment analysis, Sentiment Analysis-Reusing Pretrained Embeddings and Language Models
  - text classification, Sentiment Analysis-Reusing Pretrained Embeddings and Language Models
  - text encoding, Text Preprocessing-Using Pretrained Language Model Components
  - text summarization, Examples of Applications
  - transformer models (see transformer models)
  - word embeddings, Encoding Categorical Features Using Embeddings
- natural language understanding (NLU), Examples of Applications
- NCCL (Nvidia collective communications library), Training at Scale Using the Distribution Strategies API
- NEAT (neuroevolution of augmenting topologies), Policy Search
- negative class, Confusion Matrices, Logistic Regression
- nested dataset, Preparing the Data for Machine Learning Models
- Nesterov accelerated gradient (NAG), Nesterov Accelerated Gradient, Nadam
- Nesterov momentum optimization, Nesterov Accelerated Gradient, Nadam
- neural machine translation (NMT), Reusing Pretrained Embeddings and Language Models-Multi-head attention
  - and attention mechanisms, Attention Mechanisms-Multi-head attention
  - with transformers, An Avalanche of Transformer Models-An Avalanche of Transformer Models
- neural networks (see artificial neural networks)
- neuroevolution of augmenting topologies (NEAT), Policy Search
- next sentence prediction (NSP), An Avalanche of Transformer Models
- NLU (natural language understanding), Examples of Applications
- NMT (see neural machine translation)
- No Free Lunch theorem, Data Mismatch
- non-max suppression, bounding boxes, Object Detection
- nonlinear dimensionality reduction (NLDR), LLE-LLE
- nonlinear SVM classifiers, Nonlinear SVM Classification-SVM Classes and Computational Complexity
- nonparametric models, Regularization Hyperparameters
- nonrepresentative training data, Nonrepresentative Training Data
- nonresponse bias, Nonrepresentative Training Data

- normal distribution, The Vanishing/Exploding Gradients Problems, Glorot and He Initialization
- Normal equation, The Normal Equation-The Normal Equation
- normalization, Feature Scaling and Transformation, Batch Normalization-Implementing batch normalization with Keras, Fighting the Unstable Gradients Problem
- Normalization layer, The Normalization Layer-The Normalization Layer
- normalized exponential (softmax function), Softmax Regression
- notations, Select a Performance Measure-Select a Performance Measure, Linear Regression
- novelty detection, Unsupervised learning, Using Gaussian Mixtures for Anomaly Detection
- NP-Complete problem, CART as, The CART Training Algorithm
- NSP (next sentence prediction), An Avalanche of Transformer Models
- nucleus sampling, Generating Fake Shakespearean Text
- null hypothesis, Regularization Hyperparameters
- number of inputs, Creating the model using the sequential API, Glorot and He Initialization
- number of neurons per hidden layer, Number of Neurons per Hidden Layer
- NumPy, Running the Code Examples Using Google Colab
- NumPy arrays, Clean the Data, Custom Transformers, Using TensorFlow like NumPy-Other Data Structures
- Nvidia collective communications library (NCCL), Training at Scale Using the Distribution Strategies API
- Nvidia GPU card, Getting Your Own GPU-Getting Your Own GPU

O

- OAuth 2.0, Creating a Prediction Service on Vertex AI
- object detection, CNNs, Object Detection-You Only Look Once
- object tracking, CNNs, Object Tracking
- objectness score, Object Detection
- observation space, reinforcement learning, Learning to Optimize Rewards, Neural Network Policies
- observations, Introduction to OpenAI Gym-Introduction to OpenAI Gym
- OCR (optical character recognition), The Machine Learning Landscape
- OEL (open-ended learning), Overview of Some Popular RL Algorithms

- off-policy algorithm, Q-Learning
- offline learning, Batch learning
- on-policy algorithm, Q-Learning
- one-class SVM, Other Algorithms for Anomaly and Novelty Detection
- one-hot encoding, Handling Text and Categorical Attributes-Handling Text and Categorical Attributes, The CategoryEncoding Layer, Encoding Categorical Features Using Embeddings
- one-versus-all (OvA) strategy, Multiclass Classification-Multiclass Classification
- one-versus-one (OvO) strategy, Multiclass Classification-Multiclass Classification
- one-versus-the-rest (OvR) strategy, Multiclass Classification-Multiclass Classification
- 1cycle scheduling, Learning Rate Scheduling, Learning Rate Scheduling
- 1D convolutional layers, Semantic Segmentation, Using 1D convolutional layers to process sequences
- OneHotEncoder, Handling Text and Categorical Attributes-Handling Text and Categorical Attributes, Feature Scaling and Transformation, Transformation Pipelines
- online kernelized SVMs, Kernelized SVMs
- online learning, Online learning-Online learning
- online model, DQN, Fixed Q-value Targets
- OOB (out-of-bag) evaluation, Out-of-Bag Evaluation-Out-of-Bag Evaluation
- open-ended learning (OEL), Overview of Some Popular RL Algorithms
- OpenAI Gym, Introduction to OpenAI Gym-Introduction to OpenAI Gym
- operations (ops) and tensors, A Quick Tour of TensorFlow, Tensors and Operations-Tensors and Operations
- optical character recognition (OCR), The Machine Learning Landscape
- optimal state value, MDP, Markov Decision Processes
- optimizers, Faster Optimizers-AdamW
  - AdaGrad, AdaGrad
  - Adam optimization, Adam
  - AdaMax, AdaMax
  - AdamW, AdamW
  - hyperparameters, Learning Rate, Batch Size, and Other Hyperparameters

    - momentum optimization, Momentum
    - Nadam, Nadam
    - Nesterov accelerated gradient, Nesterov Accelerated Gradient
    - output layer, An Encoder–Decoder Network for Neural Machine Translation
    - RMSProp, RMSProp
- oracle, Fine-Tuning Neural Network Hyperparameters
- order of integration (d) hyperparameter, The ARMA Model Family
- OrdinalEncoder, Handling Text and Categorical Attributes
- orthogonal matrix, An Encoder–Decoder Network for Neural Machine Translation
- out-of-bag (OOB) evaluation, Out-of-Bag Evaluation-Out-of-Bag Evaluation
- out-of-core learning, Online learning
- out-of-sample error, Testing and Validating
- outlier detection (see anomaly detection)
- outliers, Unsupervised Learning Techniques
- output gate, LSTM, LSTM cells
- output layer, neural network, An Encoder–Decoder Network for Neural Machine Translation
- OvA (one-versus-all) strategy, Multiclass Classification-Multiclass Classification
- overcomplete autoencoder, Convolutional Autoencoders
- overfitting of data, Overfitting the Training Data-Overfitting the Training Data, Create a Test Set
    - avoiding through regularization, Avoiding Overfitting Through Regularization-Max-Norm Regularization
    - and decision trees, Regularization Hyperparameters, Regression
    - and dropout regularization, Dropout
    - gamma (γ) hyperparameter to adjust, Gaussian RBF Kernel
    - image classification, Training and evaluating the model
    - learning curves to assess, Learning Curves-Learning Curves
    - number of neurons per hidden layer, Number of Neurons per Hidden Layer
    - polynomial regression, Training Models
    - SVM model, Soft Margin Classification
- OvO (one-versus-one) strategy, Multiclass Classification-Multiclass Classification

- OvR (one-versus-the-rest) strategy, Multiclass Classification-Multiclass Classification

P

- p-value, Regularization Hyperparameters
- PACF (partial autocorrelation function), The ARMA Model Family
- padding options, convolutional layer, Implementing Convolutional Layers with Keras-Implementing Convolutional Layers with Keras
- PaLM (Pathways language model), An Avalanche of Transformer Models
- Pandas, Running the Code Examples Using Google Colab, Look for Correlations-Look for Correlations, Handling Text and Categorical Attributes, Feature Scaling and Transformation
- parallelism, training models across devices, Training Models Across Multiple Devices-Hyperparameter Tuning on Vertex AI
  - data parallelism, Data Parallelism-Bandwidth saturation
  - distribution strategies API, Training at Scale Using the Distribution Strategies API
  - with GPU, Parallel Execution Across Multiple Devices-Parallel Execution Across Multiple Devices
  - hyperparameter tuning, Hyperparameter Tuning on Vertex AI-Hyperparameter Tuning on Vertex AI
  - model parallelism, Model Parallelism-Model Parallelism
  - on TensorFlow cluster, Training a Model on a TensorFlow Cluster-Training a Model on a TensorFlow Cluster
  - Vertex AI for running large jobs, Running Large Training Jobs on Vertex AI-Running Large Training Jobs on Vertex AI
- parameter efficiency, Number of Hidden Layers
- parameter matrix, Softmax Regression
- parameter servers, Data parallelism with centralized parameters
- parameter space, Gradient Descent
- parameter vector, Linear Regression, Gradient Descent, Training and Cost Function, Softmax Regression
- parametric leaky ReLU (PReLU), Leaky ReLU
- parametric models, Regularization Hyperparameters
- partial autocorrelation function (PACF), The ARMA Model Family
- partial derivative, Batch Gradient Descent
- partial_fit(), Stochastic Gradient Descent

- Pathways language model (PaLM), An Avalanche of Transformer Models
- PCA (see principal component analysis)
- PDF (probability density function), Unsupervised Learning Techniques, Selecting the Number of Clusters
- Pearson's r, Look for Correlations
- penalties, reinforcement learning, Reinforcement learning
- PER (prioritized experience replay), Prioritized Experience Replay
- Perceiver, Vision Transformers
- percentiles, Take a Quick Look at the Data Structure
- perceptrons, Introduction to Artificial Neural Networks with Keras, The Perceptron-Classification MLPs
  - (see also multilayer perceptrons)
- performance measures, Performance Measures-The ROC Curve
  - confusion matrix, Confusion Matrices-Confusion Matrices
  - cross-validation to measure accuracy, Measuring Accuracy Using Cross-Validation-Measuring Accuracy Using Cross-Validation
  - precision and recall, Precision and Recall-The Precision/Recall Trade-off
  - ROC curve, The ROC Curve-The ROC Curve
  - selecting, Select a Performance Measure-Select a Performance Measure
- performance scheduling, Learning Rate Scheduling, Learning Rate Scheduling
- permutation(), Create a Test Set
- PG (policy gradients) algorithm, Policy Search, Policy Gradients-Policy Gradients
- piecewise constant scheduling, Learning Rate Scheduling, Learning Rate Scheduling
- PipeDream, Bandwidth saturation
- Pipeline class, Transformation Pipelines
- Pipeline constructor, Transformation Pipelines-Transformation Pipelines
- pipelines, Frame the Problem, Transformation Pipelines-Transformation Pipelines, Learning Curves, Soft Margin Classification
- pixelwise normalization layer, Progressive Growing of GANs
- placeholders, function definitions, Exploring Function Definitions and Graphs
- POET algorithm, Overview of Some Popular RL Algorithms

- policy gradients (PG) algorithm, Policy Search, Policy Gradients-Policy Gradients
- policy parameters, Policy Search
- policy space, Policy Search
- policy, reinforcement learning, Reinforcement learning, Policy Search, Neural Network Policies
- polynomial features, SVM classifiers, Polynomial Kernel
- polynomial kernel, Polynomial Kernel, Kernelized SVMs
- polynomial regression, Training Models, Polynomial Regression-Polynomial Regression
- polynomial time, The CART Training Algorithm
- PolynomialFeatures, Polynomial Regression, Nonlinear SVM Classification
- pooling kernel, Pooling Layers
- pooling layers, Pooling Layers-Implementing Pooling Layers with Keras
- positional encodings, Positional encodings-Positional encodings
- positive class, Confusion Matrices, Logistic Regression
- post-training quantization, Deploying a Model to a Mobile or Embedded Device
- posterior distribution, Variational Autoencoders
- power law distribution, Feature Scaling and Transformation
- power scheduling, Learning Rate Scheduling
- PPO (proximal policy optimization), Overview of Some Popular RL Algorithms
- precision and recall, classifier metrics, Precision and Recall-The Precision/Recall Trade-off
- precision/recall curve (PR), The Precision/Recall Trade-off, The ROC Curve
- precision/recall trade-off, The Precision/Recall Trade-off-The Precision/Recall Trade-off
- predict(), Clean the Data, Custom Transformers, Transformation Pipelines
- predicted class, Confusion Matrices
- prediction service, on Vertex AI, Creating a Prediction Service on Vertex AI-Running Batch Prediction Jobs on Vertex AI
- predictions
  - backpropagation, The Multilayer Perceptron and Backpropagation
  - confusion matrix, Confusion Matrices-Confusion Matrices

    - cross-validation to measure accuracy, Measuring Accuracy Using Cross-Validation-Measuring Accuracy Using Cross-Validation
    - decision trees, Making Predictions-The CART Training Algorithm
    - with linear SVM classifier, Under the Hood of Linear SVM Classifiers
- predictors, Supervised learning
    - (see also ensemble learning)
- predict_log_proba(), Soft Margin Classification
- predict_proba(), Soft Margin Classification
- prefetching of data, Prefetching-Prefetching
- PReLU (parametric leaky ReLU), Leaky ReLU
- preprocessed attributes, Take a Quick Look at the Data Structure
- preprocessing data (see loading and preprocessing data)
- preprocessing mismatch, The Normalization Layer
- pretraining and pretrained layers
    - on auxiliary task, Pretraining on an Auxiliary Task
    - CNNs, Using Pretrained Models from Keras-Pretrained Models for Transfer Learning
    - greedy layer-wise pretraining, Unsupervised Pretraining, Training One Autoencoder at a Time, Progressive Growing of GANs
    - language model components, Using Pretrained Language Model Components
    - reusing embeddings, Reusing Pretrained Embeddings and Language Models-Reusing Pretrained Embeddings and Language Models
    - reusing layers, Reusing Pretrained Layers-Pretraining on an Auxiliary Task
    - in unsupervised learning, Unsupervised Pretraining, Reusing Pretrained Embeddings and Language Models, An Avalanche of Transformer Models, Unsupervised Pretraining Using Stacked Autoencoders
- primal problem, The Dual Problem
- principal component (PC), Principal Components-Principal Components
- principal component analysis (PCA), PCA-Incremental PCA
    - choosing number of dimensions, Choosing the Right Number of Dimensions-Choosing the Right Number of Dimensions
    - for compression, PCA for Compression-PCA for Compression
    - explained variance ratio, Explained Variance Ratio

    - finding principal components, Principal Components
    - incremental PCA, Incremental PCA-Incremental PCA
    - preserving variance, Preserving the Variance
    - projecting down to d dimensions, Projecting Down to d Dimensions
    - randomized PCA, Randomized PCA
    - for scaling data in decision trees, Sensitivity to Axis Orientation
    - with undercomplete linear autoencoder, Performing PCA with an Undercomplete Linear Autoencoder-Performing PCA with an Undercomplete Linear Autoencoder
    - using Scikit_Learn for, Using Scikit-Learn
- prior distribution, Variational Autoencoders
- prioritized experience replay (PER), Prioritized Experience Replay
- probabilistic autoencoders, Variational Autoencoders
- probabilities, estimating, Estimating Probabilities-Estimating Probabilities, Estimating Class Probabilities, Voting Classifiers
- probability density function (PDF), Unsupervised Learning Techniques, Selecting the Number of Clusters
- probability versus likelihood, Selecting the Number of Clusters-Selecting the Number of Clusters
- profiling the network, with TensorBoard, Using TensorBoard for Visualization
- progressive web app (PWA), Running a Model in a Web Page
- projection, dimensionality reduction, Projection-Projection
- propositional logic, From Biological to Artificial Neurons
- protocol buffers (protobuf), A Brief Introduction to Protocol Buffers-TensorFlow Protobufs, Handling Lists of Lists Using the SequenceExample Protobuf, Querying TF Serving through the gRPC API
- proximal policy optimization (PPO), Overview of Some Popular RL Algorithms
- pruning of decision tree nodes, Regularization Hyperparameters
- pseudoinverse, The Normal Equation
- PWA (progressive web app), Running a Model in a Web Page
- Python API, Get the Data

Q

- Q-learning algorithm, Q-Learning-Dueling DQN
    - approximate Q-learning, Approximate Q-Learning and Deep Q-Learning

    - exploration policies, Exploration Policies
    - implementing deep Q-learning, Implementing Deep Q-Learning-Implementing Deep Q-Learning
    - variants in deep Q-learning, Deep Q-Learning Variants-Dueling DQN
- Q-value iteration algorithm, Markov Decision Processes-Markov Decision Processes
- Q-values, Markov Decision Processes-Markov Decision Processes
- quadratic equation, Polynomial Regression
- quadratic programming (QP) problems, Under the Hood of Linear SVM Classifiers
- quantization-aware training, Deploying a Model to a Mobile or Embedded Device
- quartiles, Take a Quick Look at the Data Structure
- queries per second (QPS), Training and Deploying TensorFlow Models at Scale, Creating a Prediction Service on Vertex AI
- question-answering modules, Examples of Applications
- queues, Other Data Structures, Queues

R

- radial basis function (RBF), Feature Scaling and Transformation
- ragged dimensions, Other Data Structures
- Rainbow agent, Dueling DQN
- random forests, Better Evaluation Using Cross-Validation, Ensemble Learning and Random Forests, Random Forests-Feature Importance
    - analysis of models and their errors, Analyzing the Best Models and Their Errors
    - decision trees (see decision trees)
    - extra-trees, Extra-Trees
    - feature importance measurement, Feature Importance
- random initialization, Gradient Descent, Gradient Descent
- random patches, Random Patches and Random Subspaces
- random projection algorithm, Random Projection-Random Projection
- random subspaces, Random Patches and Random Subspaces
- RandomForestClassifier, The ROC Curve-The ROC Curve
- RandomForestRegressor, Better Evaluation Using Cross-Validation
- randomized leaky ReLU (RReLU), Leaky ReLU
- randomized PCA, Randomized PCA

- randomized search, Randomized Search-Randomized Search, Fine-Tuning Neural Network Hyperparameters-Fine-Tuning Neural Network Hyperparameters
- RBF (radial basis function), Feature Scaling and Transformation
- rbf_kernel(), Feature Scaling and Transformation, Custom Transformers
- recall metric, Precision and Recall
- receiver operating characteristic (ROC) curve, The ROC Curve-The ROC Curve
- recognition network, Efficient Data Representations
  - (see also autoencoders)
- recommender system, Examples of Applications
- reconstruction error, PCA for Compression, Tying Weights
- reconstruction loss, Losses and Metrics Based on Model Internals, Efficient Data Representations
- rectified linear units (ReLU) (see ReLU)
- recurrent dropout, Processing Sequences Using RNNs and CNNs
- recurrent layer normalization, Processing Sequences Using RNNs and CNNs, Fighting the Unstable Gradients Problem
- recurrent neural networks (RNNs), Processing Sequences Using RNNs and CNNs-Beam Search
  - bidirectional, Bidirectional RNNs
  - deep RNN, Forecasting Using a Deep RNN
  - forecasting time series (see time series data)
  - gradient clipping, Gradient Clipping
  - handling long sequences, Handling Long Sequences-WaveNet
  - input and output sequences, Input and Output Sequences-Input and Output Sequences
  - memory cells, Memory Cells, Tackling the Short-Term Memory Problem-WaveNet
  - NLP (see natural language processing)
  - and Perceiver, Vision Transformers
  - splitting across devices, Model Parallelism
  - stateful, Natural Language Processing with RNNs and Attention, Stateful RNN-Stateful RNN
  - stateless, Natural Language Processing with RNNs and Attention, Stateful RNN
  - training, Training RNNs
  - and vision transformers, Vision Transformers

- recurrent neurons, Recurrent Neurons and Layers
- reduce operation, Data parallelism using the mirrored strategy
- region proposal network (RPN), You Only Look Once
- regression MLPs, Regression MLPs-Regression MLPs
- regression models
  - and classification, Supervised learning, Multioutput Classification
  - decision tree tasks, Regression-Regression
  - forecasting example, Examples of Applications
  - lasso regression, Lasso Regression-Lasso Regression
  - linear regression (see linear regression)
  - logistic regression (see logistic regression)
  - multiple regression, Frame the Problem
  - multivariate regression, Frame the Problem
  - polynomial regression, Training Models, Polynomial Regression-Polynomial Regression
  - regression MLPs, Building a Regression MLP Using the Sequential API
  - ridge regression, Ridge Regression-Ridge Regression, Elastic Net Regression
  - softmax regression, Softmax Regression-Softmax Regression
  - SVM, SVM Regression-SVM Regression
  - univariate regression, Frame the Problem
- regression to the mean, Supervised learning
- regularization, Overfitting the Training Data, Avoiding Overfitting Through Regularization-Max-Norm Regularization
  - custom regularizers, Custom Activation Functions, Initializers, Regularizers, and Constraints
  - decision trees, Regularization Hyperparameters
  - dropout, Dropout-Dropout
  - early stopping, Early Stopping-Early Stopping, Gradient Boosting, Forecasting Using a Linear Model
  - elastic net, Elastic Net Regression
  - hyperparameters, Regularization Hyperparameters-Regularization Hyperparameters
  - lasso regression, Lasso Regression-Lasso Regression
  - linear models, Regularized Linear Models-Early Stopping
  - max-norm, Max-Norm Regularization
  - MC dropout, Monte Carlo (MC) Dropout-Monte Carlo (MC) Dropout
  - ridge, Ridge Regression

    - shrinkage, Gradient Boosting
    - subword, Sentiment Analysis
    - Tikhonov, Ridge Regression-Ridge Regression
    - weight decay, AdamW
    - ℓ_1 and ℓ_2 regularization, $\ell 1$ and $\ell 2$ Regularization
- REINFORCE algorithms, Policy Gradients
- reinforcement learning (RL), Reinforcement learning, Reinforcement Learning-Overview of Some Popular RL Algorithms
    - actions, Evaluating Actions: The Credit Assignment Problem-Evaluating Actions: The Credit Assignment Problem
    - credit assignment problem, Evaluating Actions: The Credit Assignment Problem-Evaluating Actions: The Credit Assignment Problem
    - examples of, Examples of Applications, Learning to Optimize Rewards
    - learning in order to optimizing rewards, Learning to Optimize Rewards
    - Markov decision processes, Markov Decision Processes-Markov Decision Processes
    - neural network policies, Neural Network Policies
    - OpenAI Gym, Introduction to OpenAI Gym-Introduction to OpenAI Gym
    - PG algorithms, Policy Gradients-Policy Gradients
    - policy search, Policy Search
    - Q-learning, Q-Learning-Dueling DQN
    - TD learning, Temporal Difference Learning
- ReLU (rectified linear units)
    - and backpropagation, The Multilayer Perceptron and Backpropagation
    - in CNN architectures, CNN Architectures
    - as default for simple tasks, GELU, Swish, and Mish
    - leakyReLU, Leaky ReLU-Leaky ReLU
    - and MLPS, Regression MLPs
    - RNN unstable gradients problem, Fighting the Unstable Gradients Problem
    - RReLU, Leaky ReLU
    - tuning hyperparameters, Fine-Tuning Neural Network Hyperparameters
- replay buffer, Implementing Deep Q-Learning

- representation learning, Handling Text and Categorical Attributes
  - (see also autoencoders)
- residual block, Custom Models-Custom Models
- residual errors, Gradient Boosting-Gradient Boosting
- residual learning, ResNet
- residual network (ResNet), ResNet-ResNet
- residual units, ResNet
- ResNet-152, ResNet
- ResNet-34, ResNet, Implementing a ResNet-34 CNN Using Keras
- ResNet-50, Using Pretrained Models from Keras
- ResNeXt, Other Noteworthy Architectures
- REST API, querying through, Querying TF Serving through the REST API
- return, in reinforcement learning, Policy Gradients
- reverse process, diffusion model, Diffusion Models-Diffusion Models
- reverse-mode autodiff, The Multilayer Perceptron and Backpropagation, Computing Gradients Using Autodiff
- rewards, reinforcement learning, Reinforcement learning, Learning to Optimize Rewards
- Ridge, Ridge Regression
- ridge regression, Ridge Regression-Ridge Regression, Elastic Net Regression
- ridge regularization, Ridge Regression
- RidgeCV, Ridge Regression
- RL (see reinforcement learning)
- RMSProp, RMSProp
- ROC (receiver operating characteristic) curve, The ROC Curve-The ROC Curve
- root mean square error (RMSE), Select a Performance Measure-Select a Performance Measure, Train and Evaluate on the Training Set, Linear Regression, Early Stopping
- root node, decision tree, Making Predictions, Making Predictions
- RPN (region proposal network), You Only Look Once
- RReLU (randomized leaky ReLU), Leaky ReLU

S

- SAC (soft actor-critic), Overview of Some Popular RL Algorithms
- “same” padding, computer vision, Implementing Convolutional Layers with Keras

- SAMME, AdaBoost
- sample inefficiency, Policy Gradients
- sampled softmax, An Encoder–Decoder Network for Neural Machine Translation
- sampling bias, Nonrepresentative Training Data, Create a Test Set
- sampling noise, Nonrepresentative Training Data
- SARIMA model, The ARMA Model Family-The ARMA Model Family
- SavedModel, Exporting SavedModels-Exporting SavedModels
- saving, loading, and restoring models, Saving and Restoring a Model-Saving and Restoring a Model, Saving and Loading Models That Contain Custom Components-Saving and Loading Models That Contain Custom Components
- scaled dot-product attention layer, Multi-head attention
- scatter matrix, Look for Correlations-Look for Correlations
- Scikit-Learn, Objective and Approach
  - bagging and pasting in, Bagging and Pasting in Scikit-Learn
  - CART algorithm, The CART Training Algorithm, Regression
  - cross-validation, Better Evaluation Using Cross-Validation-Better Evaluation Using Cross-Validation
  - design principles, Clean the Data-Clean the Data
  - PCA implementation, Using Scikit-Learn
  - Pipeline constructor, Transformation Pipelines-Transformation Pipelines
  - sklearn.base.BaseEstimator, Custom Transformers
  - sklearn.base.clone(), Early Stopping
  - sklearn.base.TransformerMixin, Custom Transformers
  - sklearn.cluster.DBSCAN, DBSCAN
  - sklearn.cluster.KMeans, Custom Transformers, k-means
  - sklearn.cluster.MiniBatchKMeans, Accelerated k-means and mini-batch k-means
  - sklearn.compose.ColumnTransformer, Transformation Pipelines
  - sklearn.compose.TransformedTargetRegressor, Feature Scaling and Transformation
  - sklearn.datasets.load_iris(), Decision Boundaries
  - sklearn.datasets.make_moons(), Nonlinear SVM Classification
  - sklearn.decomposition.IncrementalPCA, Incremental PCA
  - sklearn.decomposition.PCA, Using Scikit-Learn
  - sklearn.ensemble.AdaBoostClassifier, AdaBoost

- sklearn.ensemble.BaggingClassifier, Bagging and Pasting in Scikit-Learn
- sklearn.ensemble.GradientBoostingRegressor, Gradient Boosting-Gradient Boosting
- sklearn.ensemble.HistGradientBoostingClassifier, Histogram-Based Gradient Boosting
- sklearn.ensemble.HistGradientBoostingRegressor, Histogram-Based Gradient Boosting
- sklearn.ensemble.RandomForestClassifier, The ROC Curve-The ROC Curve, Voting Classifiers, Random Forests, Feature Importance, Choosing the Right Number of Dimensions
- sklearn.ensemble.RandomForestRegressor, Better Evaluation Using Cross-Validation, Random Forests
- sklearn.ensemble.StackingClassifier, Stacking
- sklearn.ensemble.StackingRegressor, Stacking
- sklearn.ensemble.VotingClassifier, Voting Classifiers
- sklearn.externals.joblib, Launch, Monitor, and Maintain Your System-Launch, Monitor, and Maintain Your System
- sklearn.feature_selection.SelectFromModel, Analyzing the Best Models and Their Errors
- sklearn.impute.IterativeImputer, Clean the Data
- sklearn.impute.KNNImputer, Clean the Data
- sklearn.impute.SimpleImputer, Clean the Data
- sklearn.linear_model.ElasticNet, Elastic Net Regression
- sklearn.linear_model.Lasso, Lasso Regression
- sklearn.linear_model.LinearRegression, Model-based learning and a typical machine learning workflow, Clean the Data, Feature Scaling and Transformation, The Normal Equation, Computational Complexity, Polynomial Regression
- sklearn.linear_model.LogisticRegression, Decision Boundaries, Softmax Regression, Using Clustering for Semi-Supervised Learning
- sklearn.linear_model.Perceptron, The Perceptron
- sklearn.linear_model.Ridge, Ridge Regression
- sklearn.linear_model.RidgeCV, Ridge Regression
- sklearn.linear_model.SGDClassifier, Training a Binary Classifier, The Precision/Recall Trade-off, The Precision/Recall Trade-off, The ROC Curve-The ROC Curve, Multiclass Classification, SVM Classes and Computational Complexity, Under the Hood of Linear SVM Classifiers, The Perceptron

- sklearn.linear_model.SGDRegressor, Stochastic Gradient Descent, Early Stopping
- sklearn.manifold.LocallyLinearEmbedding, LLE
- sklearn.metrics.ConfusionMatrixDisplay, Error Analysis
- sklearn.metrics.confusion_matrix(), Confusion Matrices, Error Analysis
- sklearn.metrics.f1_score(), Precision and Recall, Multilabel Classification
- sklearn.metrics.mean_squared_error(), Train and Evaluate on the Training Set
- sklearn.metrics.precision_recall_curve(), The Precision/Recall Trade-off, The ROC Curve
- sklearn.metrics.precision_score(), Precision and Recall
- sklearn.metrics.recall_score(), Precision and Recall
- sklearn.metrics.roc_auc_score(), The ROC Curve
- sklearn.metrics.roc_curve(), The ROC Curve
- sklearn.metrics.silhouette_score(), Finding the optimal number of clusters
- sklearn.mixture.BayesianGaussianMixture, Bayesian Gaussian Mixture Models
- sklearn.mixture.GaussianMixture, Gaussian Mixtures
- sklearn.model_selection.cross_val_predict(), Confusion Matrices, The Precision/Recall Trade-off, The ROC Curve, Error Analysis, Stacking
- sklearn.model_selection.cross_val_score(), Better Evaluation Using Cross-Validation, Measuring Accuracy Using Cross-Validation
- sklearn.model_selection.GridSearchCV, Grid Search-Grid Search
- sklearn.model_selection.learning_curve(), Learning Curves
- sklearn.model_selection.RandomizedSearchCV, Choosing the Right Number of Dimensions
- sklearn.model_selection.StratifiedKFold, Measuring Accuracy Using Cross-Validation
- sklearn.model_selection.StratifiedShuffleSplit, Create a Test Set
- sklearn.model_selection.train_test_split(), Create a Test Set, Create a Test Set, Better Evaluation Using Cross-Validation
- sklearn.multiclass.OneVsOneClassifier, Multiclass Classification
- sklearn.multiclass.OneVsRestClassifier, Multiclass Classification
- sklearn.multioutput.ChainClassifier, Multilabel Classification

- sklearn.neighbors.KNeighborsClassifier, Multilabel Classification, Multioutput Classification, DBSCAN
- sklearn.neighbors.KNeighborsRegressor, Model-based learning and a typical machine learning workflow
- sklearn.neural_network.MLPClassifier, Classification MLPs
- sklearn.neural_network.MLPRegressor, Regression MLPs
- sklearn.pipeline.make_pipeline(), Learning Curves
- sklearn.pipeline.Pipeline, Transformation Pipelines
- sklearn.preprocessing.FunctionTransformer, Custom Transformers
- sklearn.preprocessing.MinMaxScaler, Feature Scaling and Transformation
- sklearn.preprocessing.OneHotEncoder, Handling Text and Categorical Attributes-Handling Text and Categorical Attributes, Feature Scaling and Transformation, Transformation Pipelines
- sklearn.preprocessing.OrdinalEncoder, Handling Text and Categorical Attributes, Histogram-Based Gradient Boosting
- sklearn.preprocessing.PolynomialFeatures, Polynomial Regression, Nonlinear SVM Classification
- sklearn.preprocessing.StandardScaler, Feature Scaling and Transformation, Gradient Descent, Ridge Regression, Nonlinear SVM Classification
- sklearn.random_projection.GaussianRandomProjection, Random Projection
- sklearn.random_projection.SparseRandomProjection, Random Projection
- sklearn.semi_supervised.LabelPropagation, Using Clustering for Semi-Supervised Learning
- sklearn.semi_supervised.LabelSpreading, Using Clustering for Semi-Supervised Learning
- sklearn.semi_supervised.SelfTrainingClassifier, Using Clustering for Semi-Supervised Learning
- sklearn.svm.LinearSVC, Soft Margin Classification, SVM Classes and Computational Complexity, Under the Hood of Linear SVM Classifiers
- sklearn.svm.SVC, Multiclass Classification, Polynomial Kernel, SVM Classes and Computational Complexity, Under the Hood of Linear SVM Classifiers
- sklearn.svm.SVR, SVM Regression

  - sklearn.tree.DecisionTreeClassifier, Training and Visualizing a Decision Tree, Gini Impurity or Entropy?, Regularization Hyperparameters, Sensitivity to Axis Orientation, Random Forests
  - sklearn.tree.DecisionTreeRegressor, Train and Evaluate on the Training Set, Decision Trees, Regression, Gradient Boosting
  - sklearn.tree.export_graphviz(), Training and Visualizing a Decision Tree
  - sklearn.tree.ExtraTreesClassifier, Extra-Trees
  - sklearn.utils.estimator_checks, Custom Transformers
  - sklearn.utils.validation module, Custom Transformers
  - SVM classification classes, SVM Classes and Computational Complexity
- score(), Clean the Data
- search engines, clustering for, Clustering Algorithms: k-means and DBSCAN
- search space, Randomized Search
- seasonality, time series modeling, Forecasting a Time Series
- second moment (variance of gradient), Adam
- second-order partial derivatives (Hessians), AdamW
- segment embedding, An Avalanche of Transformer Models
- SelectFromModel, Analyzing the Best Models and Their Errors
- self-attention layers, Attention Is All You Need: The Original Transformer Architecture, Multi-head attention, Vision Transformers
- self-distillation, Vision Transformers
- self-normalization, ELU and SELU, Dropout, Summary and Practical Guidelines
- self-supervised learning, Self-supervised learning-Self-supervised learning
- SELU (scaled ELU) activation function, ELU and SELU, Dropout
- semantic interpolation, Generating Fashion MNIST Images
- semantic segmentation, Examples of Applications, Using Clustering for Image Segmentation, Semantic Segmentation-Semantic Segmentation
- semi-supervised learning, Semi-supervised learning, Clustering Algorithms: k-means and DBSCAN, Using Clustering for Semi-Supervised Learning-Using Clustering for Semi-Supervised Learning
- SENet, SENet-SENet
- sensitivity (recall), ROC curve, The ROC Curve
- sensitivity metric, Confusion Matrices
- sensors, Learning to Optimize Rewards

- sentence encoder, Using Pretrained Language Model Components, Reusing Pretrained Embeddings and Language Models
- SentencePiece project, Sentiment Analysis
- sentiment analysis, Sentiment Analysis-Reusing Pretrained Embeddings and Language Models
- sentiment neuron, Stateful RNN
- separable convolution layer, Xception
- sequence length, Using 1D convolutional layers to process sequences, Positional encodings
- sequence-to-sequence (seq2seq) network, Input and Output Sequences, Forecasting Using a Sequence-to-Sequence Model-Forecasting Using a Sequence-to-Sequence Model
- sequence-to-vector network, Input and Output Sequences
- SequenceExample protobuf, Handling Lists of Lists Using the SequenceExample Protobuf
- sequential API, image classifier with, Building an Image Classifier Using the Sequential API-Using the model to make predictions
- service account, GCP, Creating a Prediction Service on Vertex AI
- service worker, Running a Model in a Web Page
- sets, Sets
- set_params(), Custom Transformers
- SGD (see stochastic gradient descent)
- SGDClassifier, Training a Binary Classifier, The Precision/Recall Trade-off, The Precision/Recall Trade-off, The ROC Curve-The ROC Curve, Multiclass Classification, SVM Classes and Computational Complexity, Under the Hood of Linear SVM Classifiers, Under the Hood of Linear SVM Classifiers
- SGDRegressor, Stochastic Gradient Descent, Early Stopping
- sharpening, NLP transformers, Vision Transformers
- shrinkage, Gradient Boosting
- shuffle_and_split_data(), Create a Test Set, Create a Test Set
- shuffling data, Stochastic Gradient Descent, Shuffling the Data-Shuffling the Data
- Siamese neural network, Parallel Execution Across Multiple Devices
- sigmoid activation function, Estimating Probabilities, The Multilayer Perceptron and Backpropagation, The Vanishing/Exploding Gradients Problems, Sparse Autoencoders
- signals, Biological Neurons
- silhouette coefficient, Finding the optimal number of clusters

- silhouette diagram, Finding the optimal number of clusters
- SiLU activation function, GELU, Swish, and Mish
- similarity features, SVM, Similarity Features
- SimpleImputer, Clean the Data
- simulated annealing, Stochastic Gradient Descent
- simulated environments, Introduction to OpenAI Gym-Introduction to OpenAI Gym
- single program, multiple data (SPMD), Data Parallelism-Bandwidth saturation
- single-shot learning, Semantic Segmentation
- singular value decomposition (SVD), The Normal Equation, Principal Components, Randomized PCA
- skewed datasets, Measuring Accuracy Using Cross-Validation
- skewed left/right, Take a Quick Look at the Data Structure
- skip connections, ELU and SELU, ResNet
- slack variable, Under the Hood of Linear SVM Classifiers
- smoothing terms, Batch Normalization, AdaGrad
- SMOTE (synthetic minority oversampling technique), AlexNet
- soft actor-critic (SAC), Overview of Some Popular RL Algorithms
- soft clustering, k-means
- soft margin classification, Soft Margin Classification-Soft Margin Classification, Under the Hood of Linear SVM Classifiers
- soft voting, Voting Classifiers
- softmax activation function, Softmax Regression, Classification MLPs, Creating the model using the sequential API, An Encoder–Decoder Network for Neural Machine Translation
- softmax regression, Softmax Regression-Softmax Regression
- softplus activation function, Regression MLPs
- spam filters, The Machine Learning Landscape-Why Use Machine Learning?, Types of Machine Learning Systems-Supervised learning
- sparse autoencoders, Sparse Autoencoders-Sparse Autoencoders
- sparse features, SVC class, SVM Classes and Computational Complexity
- sparse matrix, Handling Text and Categorical Attributes, Feature Scaling and Transformation, Transformation Pipelines
- sparse models, Lasso Regression, AdamW
- sparse tensors, Sparse Tensors
- sparsity loss, Sparse Autoencoders
- specificity, ROC curve, The ROC Curve
- spectral clustering, Other Clustering Algorithms

- speech recognition, Why Use Machine Learning?, Examples of Applications
- split node, decision tree, Making Predictions
- SPMD (single program, multiple data), Data Parallelism-Bandwidth saturation
- squared hinge loss, Under the Hood of Linear SVM Classifiers
- Stable Diffusion, Diffusion Models
- stacked autoencoders, Stacked Autoencoders-Training One Autoencoder at a Time
- stacking (stacked generalization), Stacking-Stacking
- stale gradients, Asynchronous updates
- standard correlation coefficient, Look for Correlations
- standard deviation, Take a Quick Look at the Data Structure
- standardization, Feature Scaling and Transformation
- StandardScaler, Feature Scaling and Transformation, Gradient Descent, Ridge Regression, Nonlinear SVM Classification
- state-action values (Q-Values), Markov Decision Processes-Markov Decision Processes
- stateful metric, Custom Metrics
- stateful RNN, Natural Language Processing with RNNs and Attention, Stateful RNN-Stateful RNN
- stateless RNN, Natural Language Processing with RNNs and Attention, Stateful RNN, Stateful RNN
- stationary time series, Forecasting a Time Series
- statistical mode, Bagging and Pasting
- statistical significance, Regularization Hyperparameters
- statsmodels library, The ARMA Model Family
- stemming, Exercises
- step functions, TLU, The Perceptron
- stochastic gradient boosting, Gradient Boosting
- stochastic gradient descent (SGD), Stochastic Gradient Descent-Stochastic Gradient Descent, SVM Classes and Computational Complexity
  - early stopping, Early Stopping
  - image classification, Compiling the model
  - and perceptron learning algorithm, The Perceptron
  - ridge regularization, Ridge Regression
  - and TD learning, Temporal Difference Learning
  - training binary classifier, Training a Binary Classifier

- stratified sampling, Create a Test Set-Create a Test Set, Measuring Accuracy Using Cross-Validation
- strat_train_set, Prepare the Data for Machine Learning Algorithms
- streaming metric, Custom Metrics
- strides, Convolutional Layers, Semantic Segmentation, Using 1D convolutional layers to process sequences
- string kernels, Gaussian RBF Kernel
- string tensors, Other Data Structures
- StringLookup layer, The StringLookup Layer
- strings, Strings-Strings
- strong learners, Voting Classifiers
- style mixing, StyleGAN, StyleGANs
- style transfer, GANs, StyleGANs
- StyleGANs, StyleGANs-StyleGANs
- subclassing API, Using the Subclassing API to Build Dynamic Models-Using the Subclassing API to Build Dynamic Models
- subgradient vector, Lasso Regression
- subsampling, pooling layer, Pooling Layers
- subword regularization, Sentiment Analysis
- super-convergence, Learning Rate Scheduling
- super-resolution, Semantic Segmentation
- supervised learning, Supervised learning
- support vector machines (SVMs), Support Vector Machines-Kernelized SVMs
  - decision function, Under the Hood of Linear SVM Classifiers-Under the Hood of Linear SVM Classifiers
  - dual problem, The Dual Problem-Kernelized SVMs
  - kernelized SVMs, Kernelized SVMs-Kernelized SVMs
  - linear classification, Linear SVM Classification-Soft Margin Classification
  - mechanics of, Under the Hood of Linear SVM Classifiers-Kernelized SVMs
  - in multiclass classification, Multiclass Classification
  - nonlinear classifiers, Nonlinear SVM Classification
  - one-class SVM, Other Algorithms for Anomaly and Novelty Detection
  - SVM regression, SVM Regression-SVM Regression
- support vectors, Linear SVM Classification

- SVC class, Polynomial Kernel, SVM Classes and Computational Complexity
- SVD (singular value decomposition), The Normal Equation, Principal Components, Randomized PCA
- SVMs (see support vector machines)
- SVR class, SVM Regression
- Swish activation function, GELU, Swish, and Mish
- Swiss roll dataset, Manifold Learning-Manifold Learning
- symbolic differentiation, Forward-Mode Autodiff
- symbolic tensors, AutoGraph and Tracing, TF Functions and Concrete Functions
- synchronous updates, with centralized parameters, Synchronous updates
- synthetic minority oversampling technique (SMOTE), AlexNet

T

- t-distributed stochastic neighbor embedding (t-SNE), Other Dimensionality Reduction Techniques, Visualizing the Fashion MNIST Dataset
- target attributes, Take a Quick Look at the Data Structure
- target distribution, transforming, Feature Scaling and Transformation
- target model, DQN, Fixed Q-value Targets
- TD error, Temporal Difference Learning
- TD target, Temporal Difference Learning
- TD-Gammon, Reinforcement Learning
- teacher forcing, An Encoder–Decoder Network for Neural Machine Translation
- temperature, Char-RNN model, Generating Fake Shakespearean Text
- temporal difference (TD) learning, Temporal Difference Learning, Prioritized Experience Replay
- tensor arrays, Tensor Arrays
- tensor processing units (TPUs), A Quick Tour of TensorFlow, Deploying a new model version, Deploying a Model to a Mobile or Embedded Device, Training a Model on a TensorFlow Cluster
- TensorBoard, Using TensorBoard for Visualization-Using TensorBoard for Visualization, A Quick Tour of TensorFlow, Masking, Running Large Training Jobs on Vertex AI
- TensorFlow, Objective and Approach, Objective and Approach, Custom Models and Training with TensorFlow-The TensorFlow Datasets

Project, Training and Deploying TensorFlow Models at Scale-Hyperparameter Tuning on Vertex AI
- (see also Keras API)
- architecture, A Quick Tour of TensorFlow
- creating training function, Using the Dataset with Keras
- custom models (see custom models and training algorithms)
- deploying model to a mobile device, Deploying a Model to a Mobile or Embedded Device-Deploying a Model to a Mobile or Embedded Device
- functions and graphs, TensorFlow Graphs-Using TF Functions with Keras (or Not)
- GPU management with, Managing the GPU RAM-Managing the GPU RAM, Parallel Execution Across Multiple Devices-Parallel Execution Across Multiple Devices
- graphs and functions, A Quick Tour of TensorFlow, TensorFlow Functions and Graphs-TF Function Rules, TensorFlow Graphs-Using TF Functions with Keras (or Not)
- hub.KerasLayer, Using Pretrained Language Model Components
- math operations, Tensors and Operations
- with NumPy, Using TensorFlow like NumPy-Other Data Structures
- operations (ops) and tensors, A Quick Tour of TensorFlow, Tensors and Operations-Tensors and NumPy
- parallelism to train models (see parallelism)
- platforms and APIs available, A Quick Tour of TensorFlow
- serving a model (see TensorFlow Serving)
- special data structures, Special Data Structures-Queues
- tf.add(), Tensors and Operations
- tf.autograph.to_code(), AutoGraph and Tracing
- tf.cast(), Type Conversions
- tf.config.set_soft_device_placement, Placing Operations and Variables on Devices
- tf.config.threading.set_inter_op_parallelism_threads(), Parallel Execution Across Multiple Devices
- tf.config.threading.set_intra_op_parallelism_threads(), Parallel Execution Across Multiple Devices
- tf.constant(), Tensors and Operations
- tf.data API (see tf.data API)
- tf.device(), Placing Operations and Variables on Devices

- tf.distribute.experimental.CentralStorageStrategy, Training at Scale Using the Distribution Strategies API
- tf.distribute.experimental.TPUStrategy, Training a Model on a TensorFlow Cluster
- tf.distribute.MirroredStrategy, Training at Scale Using the Distribution Strategies API, Hyperparameter Tuning on Vertex AI
- tf.distribute.MultiWorkerMirroredStrategy, Training a Model on a TensorFlow Cluster
- tf.float32, Tensors and NumPy
- tf.float32 data type, Custom Metrics, Preprocessing the Data
- tf.function(), TensorFlow Functions and Graphs, TF Function Rules
- tf.int32 data type, Other Data Structures
- tf.io.decode_base64(), Running Batch Prediction Jobs on Vertex AI
- tf.io.decode_csv(), Preprocessing the Data
- tf.io.decode_image(), Running Batch Prediction Jobs on Vertex AI
- tf.io.decode_png(), Running Batch Prediction Jobs on Vertex AI
- tf.io.decode_proto(), A Brief Introduction to Protocol Buffers
- tf.io.FixedLenFeature, Loading and Parsing Examples
- tf.io.parse_example(), Loading and Parsing Examples
- tf.io.parse_sequence_example(), Handling Lists of Lists Using the SequenceExample Protobuf
- tf.io.parse_single_example(), Loading and Parsing Examples
- tf.io.parse_single_sequence_example(), Handling Lists of Lists Using the SequenceExample Protobuf
- tf.io.parse_tensor(), Loading and Parsing Examples
- tf.io.serialize_tensor(), Loading and Parsing Examples
- tf.io.TFRecordOptions, Compressed TFRecord Files
- tf.io.TFRecordWriter, The TFRecord Format
- tf.io.VarLenFeature, Loading and Parsing Examples
- tf.linalg.band_part(), Multi-head attention
- tf.lite.TFLiteConverter.from_keras_model(), Deploying a Model to a Mobile or Embedded Device
- tf.make_tensor_proto(), Querying TF Serving through the gRPC API
- tf.matmul(), Tensors and Operations, Multi-head attention
- tf.nn.conv2d(), Implementing Convolutional Layers with Keras
- tf.nn.embedding_lookup(), Image Preprocessing Layers
- tf.nn.local_response_normalization(), AlexNet
- tf.nn.moments(), Image Preprocessing Layers

  - tf.nn.sampled_softmax_loss(), An Encoder–Decoder Network for Neural Machine Translation
  - tf.py_function(), TF Function Rules, Exporting SavedModels
  - tf.queue module, Other Data Structures
  - tf.queue.FIFOQueue, Queues
  - tf.RaggedTensor, Other Data Structures
  - tf.random.categorical(), Generating Fake Shakespearean Text
  - tf.reduce_max(), Implementing Pooling Layers with Keras
  - tf.reduce_mean(), Custom Training Loops
  - tf.reduce_sum(), TensorFlow Functions and Graphs
  - tf.saved_model_cli command, Exporting SavedModels
  - tf.sets module, Other Data Structures
  - tf.sort(), TF Function Rules
  - tf.SparseTensor, Other Data Structures
  - tf.stack(), Preprocessing the Data, Stateful RNN
  - tf.string data type, Other Data Structures
  - tf.strings module, Other Data Structures
  - tf.Tensor, Tensors and Operations, Variables
  - tf.TensorArray, Other Data Structures
  - tf.transpose(), Tensors and Operations
  - tf.Variable, Variables
  - tf.Variable.assign(), Variables
  - type conversions, Type Conversions
  - variables, Variables
  - web page, running a model in, Running a Model in a Web Page
- TensorFlow cluster, Training a Model on a TensorFlow Cluster-Training a Model on a TensorFlow Cluster
- TensorFlow Datasets (TFDS) project, The TensorFlow Datasets Project-The TensorFlow Datasets Project
- TensorFlow Extended (TFX), A Quick Tour of TensorFlow
- TensorFlow Hub, A Quick Tour of TensorFlow, Using Pretrained Language Model Components, Reusing Pretrained Embeddings and Language Models
- TensorFlow Lite, A Quick Tour of TensorFlow
- TensorFlow playground, Classification MLPs
- TensorFlow Serving (TF Serving), Serving a TensorFlow Model-Running Batch Prediction Jobs on Vertex AI
  - batch prediction jobs on Vertex AI, Running Batch Prediction Jobs on Vertex AI

    - creating prediction service, Creating a Prediction Service on Vertex AI-Creating a Prediction Service on Vertex AI
    - deploying new model version, Deploying a new model version-Deploying a new model version
    - Docker container, Installing and starting TensorFlow Serving
    - exporting SavedModels, Exporting SavedModels-Exporting SavedModels
    - gRPC API, querying through, Querying TF Serving through the gRPC API
    - installing and starting up, Installing and starting TensorFlow Serving-Installing and starting TensorFlow Serving
    - REST API, querying through, Querying TF Serving through the REST API
- TensorFlow Text, Text Preprocessing, Sentiment Analysis
- TensorFlow.js (TFJS) JavaScript library, Running a Model in a Web Page
- tensors, Tensors and Operations-Tensors and NumPy
- term-frequency x inverse-document-frequency (TF-IDF), Text Preprocessing
- terminal state, Markov chain, Markov Decision Processes
- test set, Testing and Validating, Create a Test Set-Create a Test Set
- text attributes, Handling Text and Categorical Attributes
- text processing (see natural language processing)
- TF Serving (see TensorFlow Serving)
- tf.data API, The tf.data API-Using the Dataset with Keras
    - chaining transformations, Chaining Transformations-Chaining Transformations
    - interleaving lines from multiple files, Interleaving Lines from Multiple Files-Interleaving Lines from Multiple Files
    - and Keras preprocessing layers, The Normalization Layer
    - prefetching, Prefetching-Prefetching
    - preprocessing the data, Preprocessing the Data-Preprocessing the Data
    - shuffling data, Shuffling the Data-Shuffling the Data
    - tf.data.AUTOTUNE, Chaining Transformations
    - tf.data.Dataset.from_tensor_slices(), The tf.data API-Chaining Transformations
    - tf.data.TFRecordDataset, The TFRecord Format, The TFRecord Format, Loading and Parsing Examples

    - using dataset with Keras, Using the Dataset with Keras-Using the Dataset with Keras
- TFDS (TensorFlow Datasets) project, The TensorFlow Datasets Project-The TensorFlow Datasets Project
- TFJS (TensorFlow.js) JavaScript library, Running a Model in a Web Page
- TFLite, Deploying a Model to a Mobile or Embedded Device-Deploying a Model to a Mobile or Embedded Device
- TFRecord format, Loading and Preprocessing Data with TensorFlow, The TFRecord Format-Handling Lists of Lists Using the SequenceExample Protobuf
- TFX (TensorFlow Extended), A Quick Tour of TensorFlow
- theoretical information criterion, Selecting the Number of Clusters
- 3D convolutional layers, Semantic Segmentation
- threshold logic units (TLUs), The Perceptron, The Multilayer Perceptron and Backpropagation
- Tikhonov regularization, Ridge Regression-Ridge Regression, Elastic Net Regression
- time series data, forecasting, Processing Sequences Using RNNs and CNNs, Forecasting a Time Series-Forecasting Using a Sequence-to-Sequence Model
    - ARMA model family, The ARMA Model Family-The ARMA Model Family
    - data preparation for ML models, Preparing the Data for Machine Learning Models-Preparing the Data for Machine Learning Models
    - with deep RNN, Forecasting Using a Deep RNN
    - with linear model, Forecasting Using a Linear Model
    - multivariate time series, Forecasting a Time Series, Forecasting Multivariate Time Series-Forecasting Multivariate Time Series
    - with sequence-to-sequence model, Input and Output Sequences, Forecasting Using a Sequence-to-Sequence Model-Forecasting Using a Sequence-to-Sequence Model
    - several time steps ahead, Forecasting Several Time Steps Ahead-Forecasting Several Time Steps Ahead
    - with simple RNN, Forecasting Using a Simple RNN-Forecasting Using a Simple RNN
- TLUs (threshold logic units), The Perceptron, The Multilayer Perceptron and Backpropagation
- TNR (true negative rate), The ROC Curve

- tokenizers library, Sentiment Analysis
- tolerance (ε), Batch Gradient Descent, SVM Classes and Computational Complexity
- TPR (true positive rate), Confusion Matrices, The ROC Curve
- TPUs (tensor processing units), A Quick Tour of TensorFlow, Deploying a new model version, Deploying a Model to a Mobile or Embedded Device, Training a Model on a TensorFlow Cluster
- train-dev set, Data Mismatch
- training, Model-based learning and a typical machine learning workflow
- training instance, What Is Machine Learning?
- training loops, Custom Training Loops-Custom Training Loops, Using the Dataset with Keras
- training models, Training Models-Softmax Regression
  - learning curves in, Learning Curves-Learning Curves
  - linear regression, Training Models, Linear Regression-Mini-Batch Gradient Descent
  - logistic regression, Logistic Regression-Softmax Regression
  - perceptrons, The Perceptron-The Perceptron
  - polynomial regression, Training Models, Polynomial Regression-Polynomial Regression
- training set, What Is Machine Learning?, Testing and Validating
  - cost function of, Training and Cost Function-Training and Cost Function
  - insufficient quantities, Insufficient Quantity of Training Data
  - irrelevant features, Irrelevant Features
  - min-max scaling, Feature Scaling and Transformation
  - nonrepresentative, Nonrepresentative Training Data
  - overfitting, Overfitting the Training Data-Overfitting the Training Data
  - preparing for ML algorithms, Prepare the Data for Machine Learning Algorithms
  - training and evaluating on, Train and Evaluate on the Training Set-Train and Evaluate on the Training Set
  - transforming data, Feature Scaling and Transformation
  - underfitting, Underfitting the Training Data
  - visualizing data, Explore and Visualize the Data to Gain Insights
- training set expansion, Exercises, AlexNet, Pretrained Models for Transfer Learning

- training/serving skew, Loading and Preprocessing Data with TensorFlow
- train_test_split(), Create a Test Set, Better Evaluation Using Cross-Validation
- transfer learning, Self-supervised learning, Reusing Pretrained Layers, Transfer Learning with Keras-Transfer Learning with Keras, Pretrained Models for Transfer Learning-Pretrained Models for Transfer Learning
- transform(), Clean the Data, Handling Text and Categorical Attributes, Feature Scaling and Transformation, Custom Transformers
- transformation of data
  - custom transformers, Custom Transformers-Custom Transformers
  - estimator transformers, Clean the Data
  - and feature scaling, Feature Scaling and Transformation-Feature Scaling and Transformation
  - transformer models (see transformer models)
- transformation pipelines, Transformation Pipelines-Transformation Pipelines
- TransformedTargetRegressor, Feature Scaling and Transformation
- transformer, Attention Is All You Need: The Original Transformer Architecture
- transformer models
  - attention mechanisms, Attention Is All You Need: The Original Transformer Architecture-Multi-head attention, Vision Transformers
  - BERT, An Avalanche of Transformer Models
  - DistilBERT, An Avalanche of Transformer Models, Hugging Face's Transformers Library-Hugging Face's Transformers Library
  - Hugging Face library, Hugging Face's Transformers Library-Hugging Face's Transformers Library
  - Pathways language model, An Avalanche of Transformer Models
  - vision transformers, Vision Transformers-Vision Transformers
- TransformerMixin, Custom Transformers
- transformers library, Hugging Face's Transformers Library-Hugging Face's Transformers Library
- translation, with RNNs, Natural Language Processing with RNNs and Attention, An Encoder–Decoder Network for Neural Machine Translation-Beam Search
  - (see also transformer models)

- transpose operator, Select a Performance Measure
- transposed convolutional layer, Semantic Segmentation-Semantic Segmentation
- true negative rate (TNR), The ROC Curve
- true negatives, confusion matrix, Confusion Matrices
- true positive rate (TPR), Confusion Matrices, The ROC Curve
- true positives, confusion matrix, Confusion Matrices
- trust region policy optimization (TRPO), Overview of Some Popular RL Algorithms
- 2D convolutional layers, Implementing Convolutional Layers with Keras
- tying weights, Tying Weights
- type I errors, confusion matrix, Confusion Matrices
- type II errors, confusion matrix, Confusion Matrices

U

- uncertainty sampling, Using Clustering for Semi-Supervised Learning
- undercomplete, autoencoder as, Efficient Data Representations-Performing PCA with an Undercomplete Linear Autoencoder
- underfitting of data, Underfitting the Training Data, Train and Evaluate on the Training Set, Learning Curves-Learning Curves, Gaussian RBF Kernel
- univariate regression, Frame the Problem
- univariate time series, Forecasting a Time Series
- Universal Sentence Encoder, Reusing Pretrained Embeddings and Language Models-Reusing Pretrained Embeddings and Language Models
- unreasonable effectiveness of data, Insufficient Quantity of Training Data
- unrolling the network through time, Recurrent Neurons and Layers
- unstable gradients problem, The Vanishing/Exploding Gradients Problems, Fighting the Unstable Gradients Problem
  - (see also vanishing and exploding gradients)
- unsupervised learning, Unsupervised learning-Unsupervised learning, Unsupervised Learning Techniques-Other Algorithms for Anomaly and Novelty Detection
  - anomaly detection, Unsupervised learning
  - association rule learning, Unsupervised learning
  - autoencoders (see autoencoders)

  - clustering (see clustering algorithms)
  - density estimation, Unsupervised Learning Techniques
  - diffusion models, Diffusion Models-Diffusion Models
  - dimensionality reduction (see dimensionality reduction)
  - GANs (see generative adversarial networks)
  - GMM, Gaussian Mixtures-Other Algorithms for Anomaly and Novelty Detection
  - k-means (see k-means algorithm)
  - novelty detection, Unsupervised learning
  - pretraining, Unsupervised Pretraining, Reusing Pretrained Embeddings and Language Models, An Avalanche of Transformer Models, Unsupervised Pretraining Using Stacked Autoencoders
  - stacked autoencoders, Unsupervised Pretraining Using Stacked Autoencoders
  - transformer models, An Avalanche of Transformer Models
  - visualization algorithms, Unsupervised learning-Unsupervised learning
- upsampling layer, Semantic Segmentation
- utility function, Model-based learning and a typical machine learning workflow

V

- VAEs (variational autoencoders), Variational Autoencoders-Variational Autoencoders
- “valid” padding, computer vision, Implementing Convolutional Layers with Keras
- validation set, Hyperparameter Tuning and Model Selection, Training and evaluating the model-Training and evaluating the model
- value_counts(), Take a Quick Look at the Data Structure
- vanishing and exploding gradients, The Vanishing/Exploding Gradients Problems-Gradient Clipping
  - activation function improvements, Better Activation Functions-GELU, Swish, and Mish
  - batch normalization, Batch Normalization-Implementing batch normalization with Keras
  - Glorot and He initialization, Glorot and He Initialization-Glorot and He Initialization
  - gradient clipping, Gradient Clipping

    - unstable gradients problem, Fighting the Unstable Gradients Problem-Fighting the Unstable Gradients Problem
- variables
    - handling in TF functions, Handling Variables and Other Resources in TF Functions-Handling Variables and Other Resources in TF Functions
    - persistence of, Custom Metrics
    - placing on GPUs, Placing Operations and Variables on Devices
    - in TensorFlow, Variables
- variance
    - bias/variance trade-off, Learning Curves
    - explained, Explained Variance Ratio-Choosing the Right Number of Dimensions
    - high variance with decision trees, Decision Trees Have a High Variance
    - preserving, Preserving the Variance
- variational autoencoders (VAEs), Variational Autoencoders-Variational Autoencoders
- vector-to-sequence network, Input and Output Sequences
- vectors, norms for measuring distance, Select a Performance Measure
- Vertex AI, Launch, Monitor, and Maintain Your System, Creating a Prediction Service on Vertex AI-Creating a Prediction Service on Vertex AI, Running Large Training Jobs on Vertex AI-Running Large Training Jobs on Vertex AI
- VGGNet, VGGNet
- virtual GPU device, Managing the GPU RAM
- vision transformers (ViTs), Vision Transformers-Vision Transformers
- visual cortex architecture, The Architecture of the Visual Cortex
- visualization of data, Examples of Applications, Unsupervised learning-Unsupervised learning, Explore and Visualize the Data to Gain Insights-Experiment with Attribute Combinations
    - decision trees, Training and Visualizing a Decision Tree-Training and Visualizing a Decision Tree
    - dimensionality reduction, Dimensionality Reduction, Choosing the Right Number of Dimensions
    - end-to-end exercise, Explore and Visualize the Data to Gain Insights-Experiment with Attribute Combinations
    - MLPs with TensorBoard, Using TensorBoard for Visualization-Using TensorBoard for Visualization

    - stacked autoencoders, Visualizing the Reconstructions-Visualizing the Reconstructions
    - t-SNE, Other Dimensionality Reduction Techniques
- ViTs (vision transformers), Vision Transformers-Vision Transformers
- voting classifiers, Voting Classifiers-Voting Classifiers

W

- wall time, Batch Normalization
- warmup phase, asynchronous model updates, Asynchronous updates
- WaveNet, Processing Sequences Using RNNs and CNNs, WaveNet-WaveNet
- weak learners, Voting Classifiers
- web page, running a model in, Running a Model in a Web Page
- weight decay, AdamW
- weight stashing, Bandwidth saturation
- weight-tying, Tying Weights
- weights
    - boosting, AdaBoost-AdaBoost
    - convolutional layers, Implementing Convolutional Layers with Keras
    - freezing reused layers, Reusing Pretrained Layers
    - of hidden layers, Creating the model using the sequential API
    - in prioritized experience replay, Prioritized Experience Replay
    - saving instead of whole model, Saving and Restoring a Model
- white box models, Making Predictions
- Wide & Deep neural network, Building Complex Models Using the Functional API-Building Complex Models Using the Functional API
- window length, Creating the Training Dataset
- wisdom of the crowd, Ensemble Learning and Random Forests
- word embeddings, Encoding Categorical Features Using Embeddings
    - neural machine translation, An Encoder–Decoder Network for Neural Machine Translation-Beam Search
    - sentiment analysis, Sentiment Analysis-Reusing Pretrained Embeddings and Language Models
- worker task type, Training a Model on a TensorFlow Cluster
- workers, Data parallelism with centralized parameters

X

- Xavier initialization, Glorot and He Initialization
- Xception (Extreme Inception), Xception-Xception, Pretrained Models for Transfer Learning-Pretrained Models for Transfer Learning
- XLA (accelerated linear algebra), TensorFlow Functions and Graphs
- XOR (exclusive or) problem, The Perceptron

Y

- You Only Look Once (YOLO), You Only Look Once-You Only Look Once

Z

- zero padding, Convolutional Layers, Implementing Convolutional Layers with Keras
- zero-shot learning (ZSL), An Avalanche of Transformer Models, Vision Transformers
- ZFNet, AlexNet