📚 Summary Tables

- Optimizers
  - 📚 Optimizers for Regression
  - 📚 Optimizers for Classification
- Loss
  - 📚 Regression Loss Functions in Keras
  - 📚 Classification Loss Functions in Keras

Optimizers

📚 Optimizers for Regression

Title	Description	Use Cases	Example of Use Case	Major Parameters	Pros	Cons
SGD (Stochastic Gradient Descent)	A simple yet efficient optimization algorithm.	Used in linear regression, logistic regression, and any case where the loss function is differentiable.	Linear regression model for house price prediction.	Learning Rate (0.01): Step size. Momentum (0.0): Accelerates SGD in the relevant direction.	Simple and easy to understand. Effective in a wide range of problems.	Requires careful tuning of the learning rate. Slow convergence on complex landscapes.
Adam	Adaptive Moment Estimation; combines ideas from RMSprop and SGD with momentum.	Works well for large datasets and high-dimensional parameter spaces.	Neural network for predicting continuous variables like temperature.	Learning Rate (0.001): Step size. Beta1 (0.9): First moment estimates. Beta2 (0.999): Second moment estimates.	Good default settings for the parameters. Efficient for large datasets and parameters.	May not converge to the global optimum on non-convex problems.

Title	Description	Use Cases	Example of Use Case	Major Parameters	Pros	Cons
RMSprop	Divides the gradient by a running average of its recent magnitude.	Suitable for recurrent neural networks and other cases where the gradient can be very different for different parameters.	Time-series prediction using RNNs.	Learning Rate (0.001): Step size. Decay (0.0): Learning rate decay.	Good in online and non-stationary settings. Adapts the learning rate for each parameter.	May require more tuning of the learning rate and other parameters.

📚 Optimizers for Classification

Title	Description	Use Cases	Example of Use Case	Major Parameters	Pros	Cons
Adam	Adaptive optimizer that combines momentum and scaling of the gradient's variance.	Commonly used in deep learning for classification tasks.	Image classification using CNNs.	Learning Rate (0.001): Step size. Beta1 (0.9): First moment. Beta2 (0.999): Second moment.	Efficient for large datasets and high-dimensional spaces. Robust and versatile.	Might not converge to the optimal solution in certain cases.
Adagrad	Adapts the learning rate to the parameters, performing larger updates for infrequent parameters.	Well-suited for sparse data like text classification.	Text classification using a simple neural network.	Learning Rate (0.01): Step size. Initial Accumulator Value (0.1): Controls past gradient discounting.	Good for sparse data. Reduces the learning rate effectively over time.	Accumulated squared gradients can lead to early and excessive decrease in learning rate.
Nadam	Combines Adam and Nesterov	Useful for deep learning models	Multi-class classification in	Learning Rate (0.002): Step size.	Faster convergence than	Can be more computationally

Title	Description	Use Cases	Example of Use Case	Major Parameters	Pros	Cons
	momentum.	where faster convergence is desired.	deep neural networks.	Beta1 (0.9): Momentum decay. Beta2 (0.999): Second moment decay.	Adam. Maintains a per-parameter learning rate.	intensive. Requires careful tuning of hyperparameters.

Loss

📚 Regression Loss Functions in Keras

Function Name	Description	Use Cases	Example of Use Case
Mean Squared Error (mse)	Calculates the mean of the squares of the differences between the predicted and actual values.	Regression problems.	Predicting house prices based on various features.
Mean Absolute Error (mae)	Calculates the mean of the absolute differences between the predicted and actual values.	Regression problems where outliers are expected.	Predicting the age of a person from their photograph.
Mean Squared Logarithmic Error (msle)	Computes the mean of the squares of the logarithmic differences between the predicted and actual values.	Regression problems where the target values are positive.	Forecasting sales in economics where numbers are large.
Huber Loss	Less sensitive to outliers than the mse. Combines mse and mae.	Regression problems, particularly with noisy datasets.	Predicting stock prices with noisy data.
Log Cosh Loss	Calculates the logarithm of the hyperbolic cosine of the prediction error.	Regression tasks, especially when handling outliers.	Estimating the load on a network.

📚 Classification Loss Functions in Keras

Function Name	Description	Use Cases	Example of Use Case
Binary Crossentropy	Measures the loss when there are only two label classes (labels are supposed to be 0 or 1).	Binary classification problems.	Classifying email as spam or not spam.

Function Name	Description	Use Cases	Example of Use Case
Categorical Crossentropy	Measures the loss when there are multiple label classes. Assumes that each sample is assigned to exactly one label.	Multi-class classification problems.	Recognizing digits from 0 to 9 in images.
Sparse Categorical Crossentropy	Similar to Categorical Crossentropy but accepts integer targets instead of one-hot encoded targets.	Multi-class classification problems with many classes.	Classifying news articles into various categories.
Kullback-Leibler Divergence (kld)	Measures how one probability distribution diverges from a second, expected probability distribution.	When the model needs to learn complex, multi-modal distributions or for comparison of two probability distributions.	Generative models or probabilistic model comparison.
Poisson Loss	Computes the loss between the predicted and actual values using the Poisson distribution.	Problems where the target variables are count data.	Predicting the number of customers visiting a store.