Deep Learning

TENSORFLOW:

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks

Tensors and Constants:

tf.constant: Creates a constant tensor.

tf.Variable: Creates a mutable variable tensor.

tf.zeros: Creates a tensor filled with zeros.

tf.ones: Creates a tensor filled with ones.

tf.fill: Creates a tensor filled with a specified value.

tf.random: Generates random tensors.

Operations and Math Functions:

tf.add: Adds two tensors element-wise.

tf.subtract: Subtracts two tensors element-wise.

tf.multiply: Multiplies two tensors element-wise.

tf.divide: Divides two tensors element-wise.

tf.matmul: Performs matrix multiplication.

tf.reduce_sum: Computes the sum of elements along specified axes.

tf.reduce_mean: Computes the mean of elements along specified axes.

tf.exp: Computes the exponential of a tensor element-wise.

tf.log: Computes the natural logarithm of a tensor element-wise.

tf.square: Computes the square of a tensor element-wise.

Neural Network Functions:

tf.nn.relu: Applies the ReLU activation function.

tf.nn.softmax: Computes the softmax activation function.

tf.nn.conv2d: Performs 2D convolution.

tf.nn.max_pool: Performs max pooling.

tf.nn.dropout: Applies dropout regularization.

Layers and Model Building:

tf.keras.layers.Dense: Creates a fully connected (dense) layer.

tf.keras.layers.Conv2D: Creates a 2D convolutional layer.

tf.keras.layers.MaxPooling2D: Creates a 2D max pooling layer.

tf.keras.layers.Dropout: Creates a dropout layer.

tf.keras.Sequential: Creates a sequential model.

Optimization and Training:

tf.train.Optimizer: Base class for gradient-based optimizers.

tf.train.GradientDescentOptimizer: Optimizer that implements the gradient descent algorithm.

tf.train.AdamOptimizer: Optimizer that implements the Adam algorithm.

tf.train.RMSPropOptimizer: Optimizer that implements the RMSProp algorithm.

tf.train.saver: Saves and restores variables.

Input and Output:

tf.data.Dataset: Represents a collection of elements.

tf.data.TextLineDataset: Reads text from one or more files.

tf.data.TFRecordDataset: Reads data from one or more TFRecord files.

tf.io.read_file: Reads the entire contents of a file.

tf.io.decode_csv: Decodes a CSV file into a tensor.

Evaluation and Metrics:

tf.metrics.accuracy: Computes the accuracy between predicted and true labels.

tf.metrics.precision: Computes the precision between predicted and true labels.

tf.metrics.recall: Computes the recall between predicted and true labels.

tf.metrics.mean_squared_error: Computes the mean squared error between two tensors.

Saving and Loading Models:

tf.saved_model.save: Saves a model to disk.

tf.saved_model.load: Loads a saved model from disk.

tf.keras.models.save_model: Saves a Keras model to disk.

tf.keras.models.load_model: Loads a saved Keras model from disk.

KERAS:

Keras is an open-source deep learning framework that provides a high-level API for building and training neural networks. It is designed to be user-friendly, modular, and extensible. Keras was initially developed as a standalone library but was later integrated into TensorFlow as the official high-level API.

Methods in Keras:

Model Creation:

keras.models.Sequential: Creates a sequential model where layers are stacked sequentially.

keras.models.Model: Allows the creation of complex models with shared layers or multiple inputs/outputs.

Layers:

keras.layers.Dense: Fully connected (dense) layer.

keras.layers.Conv2D: 2D convolutional layer.

keras.layers.MaxPooling2D: 2D max pooling layer.

keras.layers.Dropout: Dropout layer for regularization.

keras.layers.Embedding: Embedding layer for handling text or categorical data.

Activation Functions:

keras.activations.relu: Rectified Linear Unit (ReLU) activation function.

keras.activations.sigmoid: Sigmoid activation function.

keras.activations.softmax: Softmax activation function.

keras.activations.tanh: Hyperbolic tangent activation function.

Optimizers:

keras.optimizers.SGD: Stochastic Gradient Descent optimizer.

keras.optimizers.Adam: Adam optimizer.

keras.optimizers.RMSprop: RMSprop optimizer.

Loss Functions:

keras.losses.mean_squared_error: Mean Squared Error (MSE) loss.

keras.losses.categorical_crossentropy: Categorical Crossentropy loss.

keras.losses.binary_crossentropy: Binary Crossentropy loss.

Metrics:

keras.metrics.accuracy: Accuracy metric.

keras.metrics.precision: Precision metric.

keras.metrics.recall: Recall metric.

keras.metrics.mean_squared_error: Mean Squared Error metric.

Training:

model.compile: Configures the model for training, specifying the optimizer, loss function, and metrics.

model.fit: Trains the model on training data.

model.evaluate: Evaluates the model on test data.

model.predict: Generates predictions for new data.

Callbacks:

keras.callbacks.ModelCheckpoint: Saves the model during training based on specific conditions.

keras.callbacks.EarlyStopping: Stops training early based on a monitored metric.

keras.callbacks.TensorBoard: Enables visualization and monitoring of training progress using TensorBoard.