tf.constant()

Creates a constant tensor with fixed values that cannot be modified during execution. Used for storing immutable data like hyperparameters or fixed input values in your model.

tf.Variable()

Creates a mutable tensor whose value can be changed during training. Commonly used for model weights and biases that need to be updated through gradient descent.

tf.keras.Sequential()

Builds a linear stack of neural network layers where data flows sequentially from input to output. Simplest way to create models for straightforward architectures.

tf.GradientTape()

Records operations for automatic differentiation, enabling gradient computation. Essential for implementing custom training loops and calculating gradients for backpropagation.

tf.matmul()

Performs matrix multiplication between two tensors. Fundamental operation used extensively in neural networks for computing layer outputs.

tf.nn.relu()

Applies the Rectified Linear Unit activation function (max(0, x)) to a tensor. Popular activation function that introduces non-linearity while being computationally efficient.

tf.reduce_mean()

Computes the mean of tensor elements across specified dimensions. Frequently used for calculating loss values and averaging metrics across batches.

tf.data.Dataset.from_tensor_slices()

Creates a dataset from tensor slices for efficient data pipeline construction. Enables batching, shuffling, and prefetching for optimized training performance.

tf.keras.layers.Dense()

Creates a fully connected neural network layer where every input is connected to every output. The most basic building block for feedforward neural networks.

tf.train.Checkpoint()

Saves and restores model variables and optimizer states to/from disk. Critical for model persistence, resuming training, and deployment.