Mrunmayee Naik - 37 DL

EXPERIMENT - 01

TENSORFLOW:

TensorFlow is an open-source machine learning framework developed by Google. It is designed to facilitate the development and deployment of machine learning models, particularly deep neural networks. TensorFlow provides a wide range of tools and libraries that allow researchers and developers to build, train, and deploy machine learning models efficiently.

Methods:

- tf.constant(): Creates a tensor with constant values.
- tf.Variable(): Creates a mutable tensor variable for optimization.
- tf.placeholder(): Defines a placeholder for input data in TensorFlow 1.x (superseded by regular Python variables in TensorFlow 2.x).
- tf.constant_initializer(): An initializer for creating constant tensors with specific values.
- tf.keras.layers: A collection of layer classes for building neural network architectures.
- tf.nn: Module with neural network functions, including activation and loss functions.
- tf.train.Optimizer: Base class for optimization algorithms during model training.
- tf.Session(): Executes computational graphs and operations in TensorFlow 1.x.

- tf.GradientTape(): Automatic differentiation context for computing gradients in TensorFlow 2.x.

- tf.saved_model.save(): Saves trained models for deployment using TensorFlow Serving.

Operations:

- tf.add(x, y): Adds two tensors element-wise.

- tf.subtract(x, y): Subtracts one tensor from another element-wise.

- tf.multiply(x, y): Multiplies two tensors element-wise.

- tf.divide(x, y): Divides one tensor by another element-wise.

- tf.matmul(x, y): Performs matrix multiplication between two tensors.

- tf.reduce_sum(x): Computes the sum of all elements in a tensor.

- tf.reduce_mean(x): Computes the mean of all elements in a tensor.

- tf.exp(x): Computes element-wise exponentiation.

- tf.log(x): Computes element-wise natural logarithm.

- tf.square(x): Computes element-wise square.

KERAS

Keras is an open-source high-level neural networks API written in Python. It provides a user-friendly interface for designing, training, and deploying deep learning models. Keras acts as a front-end library that interfaces with popular deep learning frameworks like TensorFlow, Microsoft Cognitive Toolkit (CNTK), and Theano.

Methods:

- Sequential(): Creates a linear stack of layers, used for building a sequential neural network model.
- add(layer): Adds a layer to the model's architecture.
- compile(optimizer, loss, metrics): Configures the model for training by specifying optimizer, loss function, and evaluation metrics.
- fit(x, y, epochs, batch_size): Trains the model on training data (x, y) for a specified number of epochs with given batch size.
- evaluate(x, y): Evaluates the model's performance on validation or test data (x, y).
- predict(x): Generates predictions based on input data x using the trained model.
- summary(): Prints a summary of the model's architecture, including layer types, output shapes, and parameters.
- layers: Provides access to the list of layers in the model.
- get_layer(name): Retrieves a layer by its name from the model.
- save(filepath): Saves the model's architecture, weights, and optimizer state to a file.

Functions:

- keras.models.Sequential(): Creates a linear stack of layers for building a sequential neural network model.
- keras.layers: A module containing various layer classes used to construct neural network architectures.
- keras.layers.Dense(units, activation): Fully connected layer with specified number of units and activation function.
- keras.layers.Conv2D(filters, kernel_size, activation): 2D convolutional layer for image data with specified filters and kernel size.
- keras.layers.MaxPooling2D(pool_size): 2D max pooling layer to downsample feature maps.
- keras.layers.LSTM(units, activation): Long Short-Term Memory (LSTM) layer for sequence data.
- keras.optimizers: A module containing various optimization algorithms for model training.
- keras.optimizers.Adam(learning_rate): Adam optimizer with adjustable learning rate.
- keras.losses: A module containing various loss functions used to compute the model's error during training.
- keras.losses.mean_squared_error(y_true, y_pred): Mean squared error loss between true and predicted values.