

Experiment 01

Tensorflow :

TensorFlow, a popular open-source machine learning framework developed by Google. TensorFlow provides a wide range of functions and APIs for building and training machine learning models.

- **tf.constant:** This function is used to create constant tensors. Tensors are the primary data structure in TensorFlow, similar to multi-dimensional arrays.
- **tf.Variable:** Unlike constants, variables are mutable tensors that can be modified during training. They are often used to store model parameters.
- **tf.placeholder (Deprecated):** In earlier versions of TensorFlow, placeholders were used to feed data into a computation graph during training. However, they are now mostly replaced by the `tf.data` API and the `tf.function` decorator for performance improvements.
- **tf.data:** TensorFlow's `tf.data` API is used to efficiently load and preprocess data for training. It offers various functions to create input pipelines using different data sources.
- **tf.keras.layers:** TensorFlow's high-level API, Keras, provides an easy way to build and define neural network layers. Various layers like Dense, Conv2D, LSTM, etc., can be used to construct complex models.
- **tf.keras.models:** This module provides tools for building and training complete neural network models. It includes the Sequential model for simple linear stacks of layers and the Functional API for more complex architectures.
- **tf.optimizers:** This module provides various optimization algorithms to update model parameters during training, such as SGD, Adam, RMSProp, etc.
- **tf.reduce_XXX:** TensorFlow provides various functions for reducing dimensions in tensors, such as `tf.reduce_sum`, `tf.reduce_mean`, and `tf.reduce_max`.

Keras :

Keras which is a popular deep learning framework for building and training neural networks. Keras was integrated into TensorFlow as its official high-level API

- **keras.models.Sequential:** A class used to create a linear stack of layers, which is the simplest way to build a neural network in Keras.
- **keras.layers:** Module containing various types of layers, such as Dense (fully connected), Conv2D (convolutional), LSTM (long short-term memory), etc., that you can stack together to create your neural network architecture.
- **keras.optimizers:** Module containing optimizers like Adam, SGD, and RMSprop that are used to update the weights of the network during training.
- **keras.losses:** Module containing loss functions such as mean_squared_error, categorical_crossentropy, and binary_crossentropy.
- **keras.metrics:** Module containing evaluation metrics like accuracy, precision, recall, etc.
- **model.compile():** Method to compile the model, where you specify the optimizer, loss function, and metrics.
- **model.fit():** Method to train the model on your training data.
- **model.evaluate():** Method to evaluate the trained model on a validation or test dataset.
- **model.predict():** Method to make predictions using the trained model.