

Deep learning is a subset of machine learning that involves training artificial neural networks to recognize patterns in data. Python is one of the most popular programming languages for deep learning because it has many powerful libraries and frameworks that make it easy to build and train neural networks.

1. **TensorFlow:** TensorFlow is a popular open-source platform for deep learning developed by Google. Some of the parameters classes used in TensorFlow include `tf.keras.layers`, `tf.keras.optimizers`, and `tf.keras.losses`.

- `tf.keras.layers`: This module provides a way to create layers for neural networks. It includes classes for common types of layers such as dense layers, convolutional layers, and recurrent layers.

- `tf.keras.optimizers`: This module provides a way to optimize the parameters of a neural network during training. It includes classes for common optimization algorithms such as stochastic gradient descent (SGD), Adam, and Adagrad.

- `tf.keras.losses`: This module provides a way to define the loss function that is used during training. It includes classes for common loss functions such as mean squared error (MSE), binary cross-entropy, and categorical cross-entropy.

2. **Keras:** Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks.

Keras has several classes that can be used to build deep learning models. Some of the most commonly used classes are:

- **Sequential:** This is a linear stack of layers where you can use the large variety of available layers in Keras. You can add layers to the model using the `add()` method. For example, `model.add(Dense(64, activation='relu'))` adds a dense layer with 64 units and ReLU activation function to the model.

- **Model:** This is the most general class that can be used to define any kind of model. It is typically used when you have multiple inputs or outputs in your model.

- **Layers:** These are the building blocks of neural networks in Keras. They are used to define the input layer, hidden layers, and output layer of your model. Some commonly used layers are:

- **Dense:** This is a fully connected layer where each neuron is connected to every neuron in the previous layer.
- **Conv2D:** This is a 2D convolutional layer that is typically used for image processing.
- **MaxPooling2D:** This is a 2D pooling layer that is typically used for image processing.
- **LSTM:** This is a long short-term memory layer that is typically used for sequence processing.

- **Activation:** These are functions that are applied to the output of a layer. They are used to introduce non-linearity into your model. Some commonly used activation functions are:

- **ReLU:** Rectified Linear Unit.
- **Sigmoid:** Sigmoid function.
- **Softmax:** Softmax function.

- **Optimizers:** These are algorithms that are used to optimize the weights of your model during training. Some commonly used optimizers are:

- **SGD:** Stochastic Gradient Descent.
- **Adam:** Adaptive Moment Estimation.
- **RMSprop:** Root Mean Square Propagation.

- **Losses:** These are functions that are used to compute the error between the predicted output and the actual output during training. Some commonly used losses are:

- **Mean Squared Error (MSE):** This is typically used for regression problems.
- **Binary Crossentropy:** This is typically used for binary classification problems.
- **Categorical Crossentropy:** This is typically used for multi-class classification problems.

3. **Theano:** Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. Some of the parameters classes used in Theano include `theano.tensor.nnet.conv2d` and `theano.tensor.nnet.relu`.

- `theano.tensor.nnet.conv2d`: This class provides a way to perform 2D convolution on multi-dimensional arrays. It is commonly used in convolutional neural networks (CNNs) for image recognition tasks.

- `theano.tensor.nnet.relu`: This class provides a way to apply the rectified linear unit (ReLU) activation function to multi-dimensional arrays. ReLU is commonly used as an activation function in neural networks because it is computationally efficient and has been shown to work well in practice.

4. **Blocks:** Blocks is a Theano framework for building and training neural networks. Some of the parameters classes used in Blocks include `blocks.bricks.conv.ConvolutionalLayer` and `blocks.bricks.recurrent.LSTM`.

- `blocks.bricks.conv.ConvolutionalLayer`: This class provides a way to perform convolution on multi-dimensional arrays. It is commonly used in CNNs for image recognition tasks.

- `blocks.bricks.recurrent.LSTM`: This class provides a way to create long short-term memory (LSTM) cells for recurrent neural networks (RNNs). LSTM cells are commonly used in RNNs because they are able to remember information over long periods of time.

5. **PyTorch:** PyTorch is another popular open-source machine learning library based on Torch library. Some of the parameters classes used in PyTorch include `torch.nn.Conv2d` and `torch.nn.ReLU`.

- `torch.nn.Conv2d`: This class provides a way to perform 2D convolution on multi-dimensional arrays. It is commonly used in CNNs for image recognition tasks.

- `torch.nn.ReLU`: This class provides a way to apply the ReLU activation function to multi-dimensional arrays.