1. **Core Layers**

These are the fundamental layers used in almost every neural network.

- **Dense**: Fully connected layer.
- **Activation**: Applies an activation function.
- **Dropout**: Randomly sets input units to 0 during training to prevent overfitting.
- **Flatten**: Flattens the input without affecting the batch size.
- **Reshape**: Reshapes an output to a certain shape.

Example:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Dropout, Flatten, Reshape

model = Sequential([
    Dense(128, input_shape=(784,)),
    Activation('relu'),
    Dropout(0.2),
    Dense(64),
    Activation('relu'),
    Flatten(),
    Reshape((8, 8, 1)),
    Dense(10, activation='softmax')
])
```

2. **Convolutional Layers**

Used primarily for processing grid-like data such as images.

- **Conv1D, Conv2D, Conv3D**: Convolution layers for 1D, 2D, and 3D inputs.
- **SeparableConv2D**: Depthwise separable convolution.
- **ConvLSTM2D**: Convolutional LSTM.
- **DepthwiseConv2D**: Depthwise convolution only.
- **Conv2DTranspose**: Transposed convolution (deconvolution).

Example:

```
from tensorflow.keras.layers import Conv2D, SeparableConv2D, ConvLSTM2D, Conv1D,
Conv3D, Conv2DTranspose, DepthwiseConv2D, LocallyConnected2D
from tensorflow.keras.models import Sequential

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    SeparableConv2D(64, (3, 3), activation='relu'),
    ConvLSTM2D(64, (3, 3), activation='relu', return_sequences=True),
    Conv1D(32, 3, activation='relu', input_shape=(64, 64)),
    Conv3D(16, (3, 3, 3), activation='relu', input_shape=(16, 64, 64, 3)),
```

```
    Conv2DTranspose(32, (3, 3), activation='relu'),
    DepthwiseConv2D((3, 3), activation='relu'),
    LocallyConnected2D(32, (3, 3), activation='relu')
])
```

3. **Normalization Layers**

Helps in stabilizing and speeding up the training process.

- **BatchNormalization**: Normalizes the activations of the previous layer at each batch.
- **LayerNormalization**: Normalizes across the features instead of the batch.

Example:

```
from tensorflow.keras.layers import BatchNormalization, LayerNormalization, Dense,
Activation
from tensorflow.keras.models import Sequential

model = Sequential([
    Dense(64, input_shape=(100,)),
    BatchNormalization(),
    Activation('relu'),
    LayerNormalization(),
    Dense(10, activation='softmax')
])
```

4. **Pooling Layers**

Used to reduce the spatial dimensions (width, height) of the input.

- **MaxPooling1D, MaxPooling2D, MaxPooling3D**: Max pooling operations.
- **AveragePooling1D, AveragePooling2D, AveragePooling3D**: Average pooling operations.
- **GlobalMaxPooling2D, GlobalAveragePooling2D**: Global pooling operations.

Example:

```
from tensorflow.keras.layers import MaxPooling2D, GlobalAveragePooling2D, Conv2D
from tensorflow.keras.models import Sequential

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    GlobalAveragePooling2D()
])
```