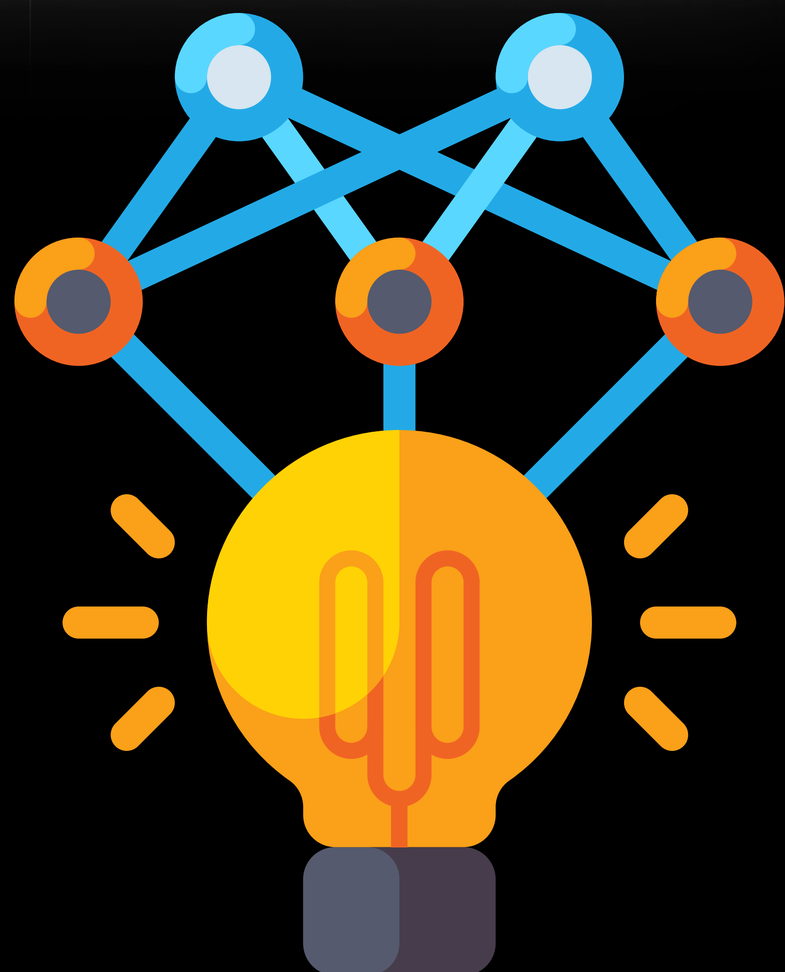
@LEARNEVERYTHINGAI

# DEEP LEARNING GUIDE FOR DATA SCIENCE



**SHIVAM MODI**
@learneverythingai

**Deep Learning** is a subset of **Machine Learning** that uses neural **networks** to learn patterns from data. It's revolutionizing **data science!**

## NEURAL NETWORK ARCHITECTURE

Neural networks consist of interconnected layers of nodes (neurons) that transform input data, passing it through each layer to produce an output. The network learns by adjusting weights during training.

```
# Sample code for a basic neural network
import tensorflow as tf

model = tf.keras.Sequential([
   tf.keras.layers.Dense(units=64, activation='relu', input_shape=
(input_dim,)),
   tf.keras.layers.Dense(units=32, activation='relu'),
   tf.keras.layers.Dense(units=output_dim, activation='softmax')
])
```

# ACTIVATION FUNCTIONS

Activation functions introduce non-linearity, enabling neural networks to learn complex patterns and make predictions. ReLU is widely used for hidden layers due to its simplicity and effectiveness.

```
# Example code for ReLU activation function
def relu(x):
    return max(0, x)
```

# BACKPROPAGATION

Backpropagation is the core algorithm behind training neural networks. It calculates gradients to adjust weights, minimizing errors and improving model accuracy during training.

```
# Formula for updating weights with backpropagation
new_weight = old_weight - learning_rate * gradient
```

# CONVOLUTIONAL NEURAL NETWORKS (CNN)

**CNNs are designed to process grid-like data, like images. They employ convolutional layers to detect features and pooling layers to reduce spatial dimensions, making them ideal for image recognition tasks.**

```
# Code for a simple CNN
import tensorflow as tf

model = tf.keras.Sequential([
   tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(width, height, channels)),
   tf.keras.layers.MaxPooling2D((2, 2)),
   tf.keras.layers.Flatten(),
   tf.keras.layers.Dense(64, activation='relu'),
   tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

# RECURRENT NEURAL NETWORKS (RNN)

RNNs excel at processing sequential data, like time series or natural language. Their recurrent connections allow them to maintain context and remember information from past inputs.

```
# Code for a basic RNN
import tensorflow as tf

model = tf.keras.Sequential([
    tf.keras.layers.SimpleRNN(64, activation='relu', input_shape=
(timesteps, features)),
    tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

# REGULARIZATION TECHNIQUES

Regularization methods like Dropout and L2 Regularization prevent neural networks from overfitting by reducing the impact of certain neurons during training and adding penalties to large weights.

```
# Example code for applying Dropout
import tensorflow as tf

model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=
(input_dim,)),
    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.Dense(output_dim, activation='softmax')
])
```

SHIVAM MODI
@learneverythingai

# OPTIMIZERS

Optimizers control how neural networks update their weights during training. Algorithms like Adam and RMSprop help speed up convergence and improve training efficiency.

```
# Example code using the Adam optimizer
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=optimizer,
loss='categorical_crossentropy', metrics=['accuracy'])
```

**SHIVAM MODI**
@learneverythingai

# LOSS FUNCTIONS

Loss functions measure the difference between predicted and actual values during training. Choosing the right loss function is crucial, as it affects how the model learns from data.

```
# Example code for using categorical cross-entropy loss
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
```

**SHIVAM MODI**
@learneverythingai

@learneverythingai

# Like this Post?

- Follow Me
- Share with your friends
- Check out my previous posts

SAVE THIS

SHIVAM MODI
@learneverythingai

Follow

SHARE

www.learneverythingai.com