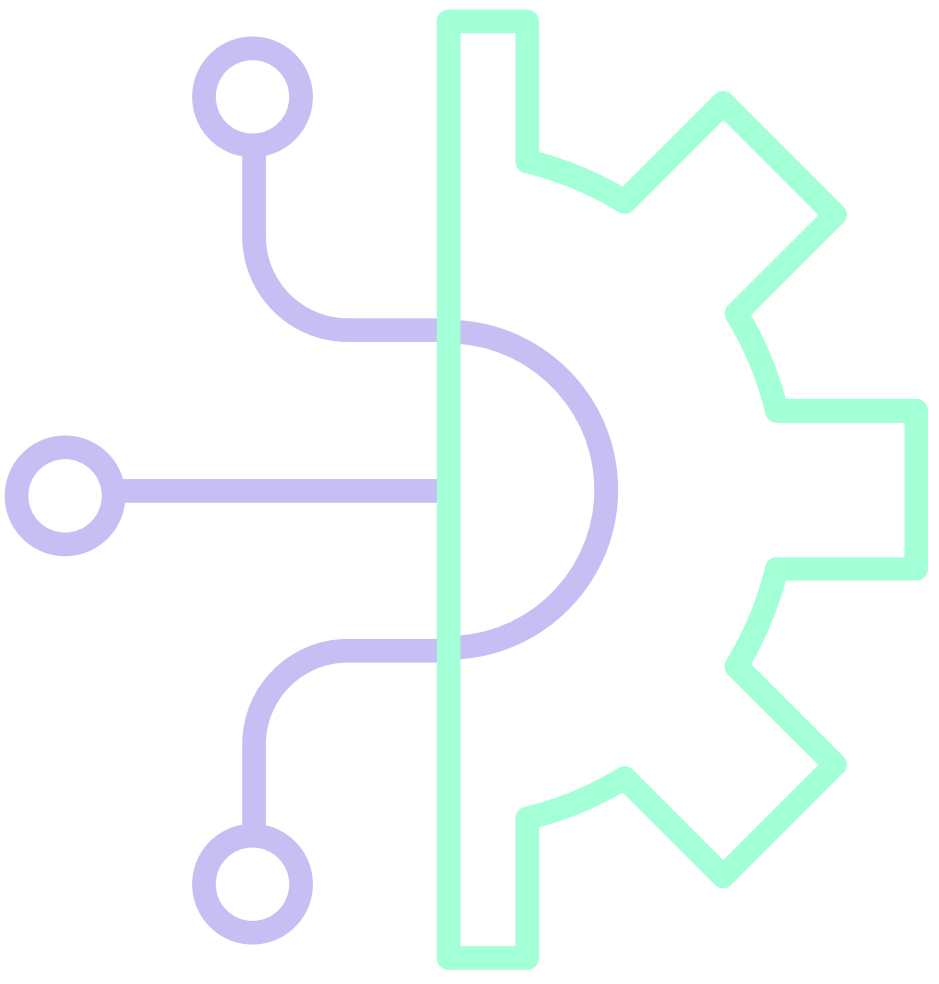
Learnbay

DEEP LEARNING INTERVIEW QUESTIONS



www.learnbay.co

1 What is deep learning, and how is it different from machine learning?

Answer:
Deep learning is a subset of machine learning that uses neural networks with multiple layers (deep architectures) to model complex patterns in data.

Feature	Machine Learning	Deep Learning
Feature engineering	Manual	Automatic (via hidden layers)
Model complexity	Limited	High
Data requirement	Works well with small datasets	Requires large datasets

2

Explain the concept of neural networks.

Answer:
A neural network is a computational model inspired by the human brain. It consists of layers of interconnected nodes (neurons) that process input data and generate output.

Code Sample

```
from keras.models import Sequential
from keras.layers import Dense

model = Sequential()
model.add(Dense(64, input_dim=10, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam',
loss='binary_crossentropy', metrics=['accuracy'])
model.summary()
```

3

What is the role of activation functions in a neural network?

Answer:
Activation functions introduce non-linearity into the network, allowing it to model complex relationships.

Common activation functions:

Function	Formula	Use Case
ReLU	f(x)=max(0,x)f(x)=max(0,x)	Hidden layers
Sigmoid	f(x)=11+e−xf(x)=1+e−x1	Binary classification
Tanh	f(x)=tanh(x)f(x)=tanh(x)	Outputs in range [-1, 1]

4 What are convolutional neural networks (CNNs), and where are they used?

Answer:
CNNs are specialized neural networks designed for processing grid-like data, such as images. They use convolutional layers to extract spatial features.

Code Sample

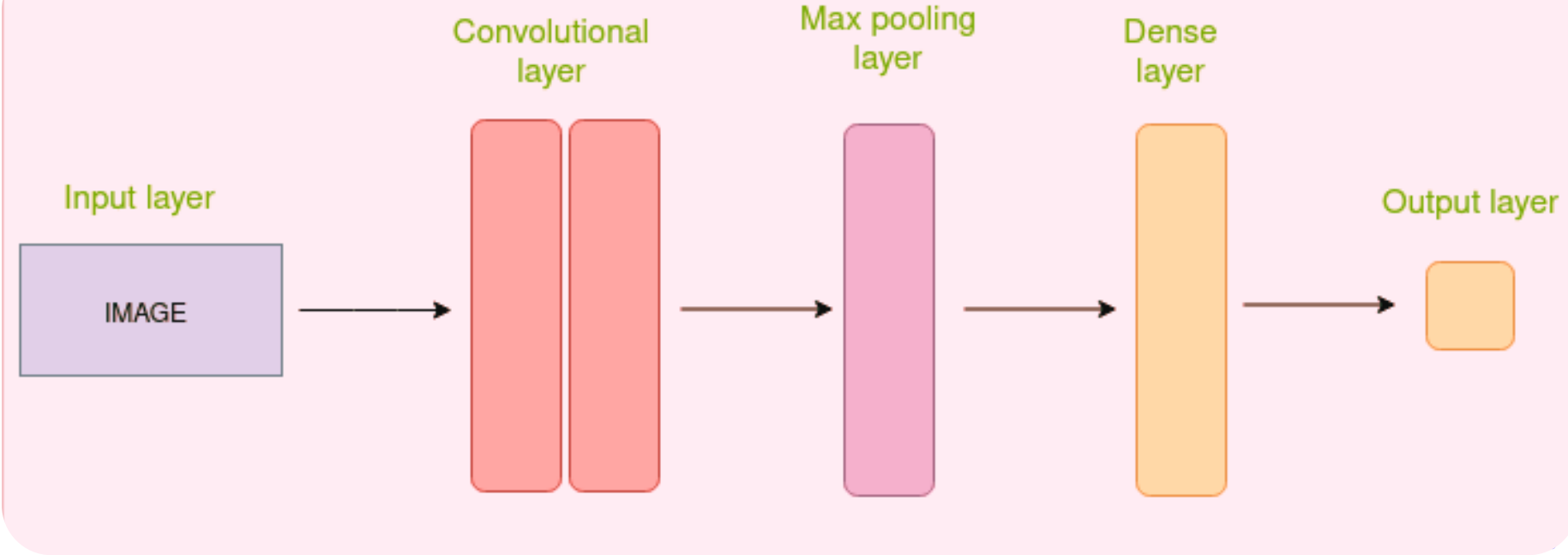
```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D,
Flatten, Dense

model = Sequential([
    Conv2D(32, (3, 3), activation='relu',
input_shape=(64, 64, 3)),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])
model.summary()
```

4 What are convolutional neural networks (CNNs), and where are they used?

Applications:

Field	Use Case
Healthcare	Medical image analysis
Retail	Object detection in stores
Security	Facial recognition

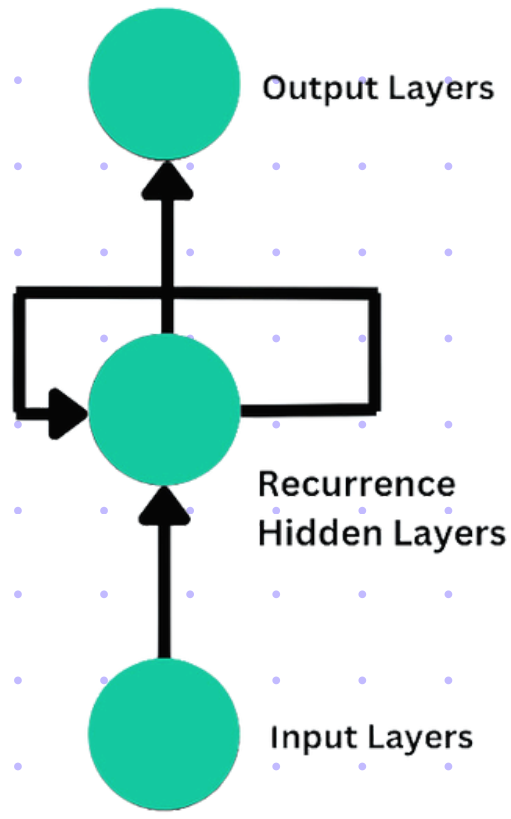


5

Explain the working of recurrent neural networks (RNNs)

Answer:

RNNs process sequential data by maintaining a memory of previous inputs through recurrent connections.



Code Sample

```
from keras.models import Sequential
from keras.layers import SimpleRNN, Dense

model = Sequential([
    SimpleRNN(50, activation='relu', input_shape=(10, 1)),
    Dense(1, activation='sigmoid')
])
model.compile(optimizer='adam',
loss='binary_crossentropy', metrics=['accuracy'])
model.summary()
```

6

What is gradient descent, and how does it work?

Answer:
Gradient descent is an optimization algorithm used to minimize the loss function by updating model parameters in the direction of the negative gradient.

Variants of gradient descent:

Type	Description
Batch gradient descent	**Uses the entire dataset for updates**
Stochastic gradient descent (SGD)	**Updates weights for each data point**
Mini-batch SGD	**Uses small batches for updates**

7

Explain the differences between L1 and L2 regularization.

Answer:
Gradient descent is an optimization algorithm used to minimize the loss function by updating model parameters in the direction of the negative gradient.

Variants of gradient descent:

Aspect	L1 Regularization	L2 Regularization
Definition	Adds (\lambda \sum	w
Weight penalty	Shrinks weights to 0	Reduces weight magnitude
Use case	Sparse models	General regularization

Key Insight:

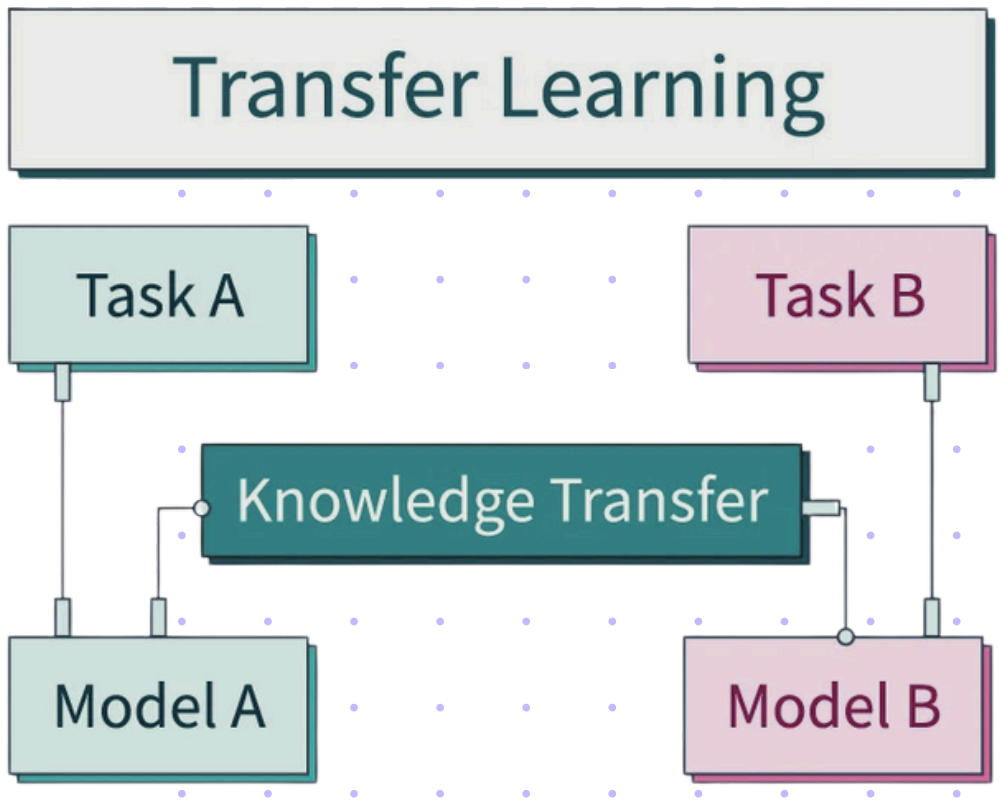
- **L1 Regularization** is ideal for creating sparse models by eliminating irrelevant features.
- **L2 Regularization** is best for stabilizing models by penalizing large weights without eliminating them.

ADVANCED CONCEPTS

8

What is transfer learning, and how is it implemented?

Answer:
Transfer learning reuses a pre-trained model on a new, similar task to save training time and improve performance.



Code Sample

```
from tensorflow.keras.applications import VGG16

base_model = VGG16(weights='imagenet',
include_top=False, input_shape=(224, 224, 3))
for layer in base_model.layers:
    layer.trainable = False
```

9

Explain the transformer architecture

Answer:
Transformers use self-attention mechanisms to process sequences without requiring sequential input, unlike RNNs.

They are widely used in natural language processing tasks like translation and summarization.

Applications:

Task	Example
Translation	Google Translate
Text summarization	News article summaries
Chatbots	GPT-based chatbots

SCENARIO-BASED

10

How would you handle imbalanced data in a classification problem?

Answer:

- Use oversampling (e.g., SMOTE) or undersampling.
- Apply class weights during training.
- Use evaluation metrics like precision-recall instead of accuracy.

Code Sample

```
from sklearn.utils.class_weight import
compute_class_weight

class_weights = compute_class_weight('balanced',
classes=[0, 1], y=train_labels)
model.fit(X_train, y_train, class_weight=class_weights)
```

Learnbay

India's Most Trusted Upskilling Platform for Professionals

Advance AI and ML Certification Program

- 1:1 Doubt Clearing
- Specialize in your domain
- Hybrid Learning Mode Available
- Job Assistance

"Be the mind behind the machines—**learn generative AI** to innovate smarter."

Get certified from:

IBM · Microsoft · IIT Guwahati · E&ICT Academy IIT Guwahati · WOOLF