**1. Q: How do you create a simple feedforward neural network in Keras?**

from keras.models import Sequential

from keras.layers import Dense

# Define the model

model = Sequential()

model.add(Dense(64, input\_dim=10, activation='relu')) # Input layer

model.add(Dense(32, activation='relu')) # Hidden layer

model.add(Dense(1, activation='sigmoid')) # Output layer

# Compile the model

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

**2. Q: How do you initialize weights in a Keras model?**

from keras.initializers import RandomNormal

model = Sequential()

model.add(Dense(64, input\_dim=10, activation='relu', kernel\_initializer=RandomNormal(mean=0.0, stddev=0.05)))

**3. Q: How do you add dropout to prevent overfitting?**

from keras.layers import Dropout

model = Sequential()

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.5)) # Dropout layer with a 50% rate

model.add(Dense(1, activation='sigmoid'))

**4. Q: How do you implement early stopping during training?**

from keras.callbacks import EarlyStopping

early\_stopping = EarlyStopping(monitor='val\_loss', patience=3)

history = model.fit(X\_train, y\_train, validation\_data=(X\_val, y\_val), epochs=100, callbacks=[early\_stopping])

**5. Q: How do you save and load a trained neural network model in Keras?**

# Save model

model.save('model.h5')

# Load model

from keras.models import load\_model

model = load\_model('model.h5')

**6. Q: How do you normalize input data before feeding it into a neural network?**

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

**7. Q: How do you implement a neural network with multiple hidden layers?**

model = Sequential()

model.add(Dense(128, input\_dim=10, activation='relu'))

model.add(Dense(64, activation='relu'))

model.add(Dense(32, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

**8. Q: How do you change the learning rate of the optimizer in Keras?**

from keras.optimizers import Adam

optimizer = Adam(learning\_rate=0.001)

model.compile(optimizer=optimizer, loss='binary\_crossentropy', metrics=['accuracy'])

**9. Q: How do you visualize the architecture of a neural network in Keras?**

from keras.utils.vis\_utils import plot\_model

plot\_model(model, to\_file='model.png', show\_shapes=True, show\_layer\_names=True)

**10. Q: How do you implement a custom loss function in Keras?**

from keras import backend as K

def custom\_loss(y\_true, y\_pred):

return K.mean(K.square(y\_pred - y\_true), axis=-1)

model.compile(optimizer='adam', loss=custom\_loss)

**11. Q: How do you use the BatchNormalization layer to stabilize training?**

from keras.layers import BatchNormalization

model = Sequential()

model.add(Dense(64, input\_dim=10, activation='relu'))

model.add(BatchNormalization())

model.add(Dense(1, activation='sigmoid'))

**12. Q: How do you use the LeakyReLU activation function in a neural network?**

from keras.layers import LeakyReLU

model = Sequential()

model.add(Dense(64, input\_dim=10))

model.add(LeakyReLU(alpha=0.1)) # LeakyReLU with alpha parameter

model.add(Dense(1, activation='sigmoid'))

**13. Q: How do you implement a neural network with L2 regularization?**

from keras.regularizers import l2

model = Sequential()

model.add(Dense(64, input\_dim=10, activation='relu', kernel\_regularizer=l2(0.01)))

model.add(Dense(1, activation='sigmoid'))

**14. Q: How do you implement a convolutional neural network (CNN) in Keras?**

from keras.layers import Conv2D, MaxPooling2D, Flatten

model = Sequential()

model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 3)))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Flatten())

model.add(Dense(64, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

**15. Q: How do you implement a recurrent neural network (RNN) using LSTM in Keras?**

from keras.layers import LSTM

model = Sequential()

model.add(LSTM(100, input\_shape=(10, 64)))

model.add(Dense(1, activation='sigmoid'))

**16. Q: How do you create a neural network with multiple inputs in Keras?**

from keras.layers import Input

from keras.models import Model

input\_1 = Input(shape=(10,))

input\_2 = Input(shape=(5,))

x1 = Dense(64, activation='relu')(input\_1)

x2 = Dense(64, activation='relu')(input\_2)

merged = concatenate([x1, x2])

output = Dense(1, activation='sigmoid')(merged)

model = Model(inputs=[input\_1, input\_2], outputs=output)

**17. Q: How do you freeze the layers of a pre-trained model in Keras?**

from keras.applications import VGG16

base\_model = VGG16(weights='imagenet', include\_top=False)

for layer in base\_model.layers:

layer.trainable = False

**18. Q: How do you implement a neural network that outputs probabilities for multi-class classification?**

model = Sequential()

model.add(Dense(64, input\_dim=10, activation='relu'))

model.add(Dense(3, activation='softmax')) # Softmax for multi-class classification

**19. Q: How do you add a callback to reduce the learning rate when the validation loss plateaus?**

from keras.callbacks import ReduceLROnPlateau

reduce\_lr = ReduceLROnPlateau(monitor='val\_loss', factor=0.2, patience=3, min\_lr=0.001)

model.fit(X\_train, y\_train, validation\_data=(X\_val, y\_val), epochs=100, callbacks=[reduce\_lr])

**20. Q: How do you load pre-trained weights into a neural network model in Keras?**

model.load\_weights('model\_weights.h5')