

Deep Learning in Python



Community
Consulting
Academy

Deep Learning Basics

Neural Networks –

Composed of input, hidden, and output layers

Activation Functions –

Sigmoid, ReLU, Tanh, Softmax

Loss Functions – MSE, Cross-Entropy

Optimizers – SGD, Adam, RMSprop



Community
Consulting
Academy



Simple Neural Network

Feedforward Neural Network (FNN)



```
import tensorflow as tf
from tensorflow.keras.models import
Sequential
from tensorflow.keras.layers import Dense

model = Sequential([
    Dense(64, activation='relu', input_shape=
(10,)),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam',
loss='binary_crossentropy', metrics=
['accuracy'])
model.summary()
```



Community
Consulting
Academy



Activation Functions

- **ReLU** (Rectified Linear Unit) – $\max(0, x)$
- **Sigmoid** – Converts to probability (0 to 1)
- **Tanh** – Scales between -1 and 1
- **Softmax** – Used in multi-class classification



```
import tensorflow.keras.backend  
as K  
def relu(x):  
    return K.maximum(0, x)  
def sigmoid(x):  
    return 1 / (1 + K.exp(-x))
```



Community
Consulting
Academy



Training a DL Model

Train & Evaluate a Neural Network



```
history = model.fit(X_train,  
y_train, epochs=50,  
batch_size=32, validation_data=  
(X_test, y_test))  
loss, accuracy =  
model.evaluate(X_test, y_test)  
print(f"Test Accuracy: {accuracy  
* 100:.2f}%")
```



Community
Consulting
Academy



Convolutional Neural Networks (CNNs)

Best for Image Classification & Object Detection



```
from tensorflow.keras.layers import Conv2D,  
MaxPooling2D, Flatten  
  
cnn_model = Sequential([  
    Conv2D(32, (3,3), activation='relu', input_shape=  
(128,128,3)),  
    MaxPooling2D(2,2),  
    Conv2D(64, (3,3), activation='relu'),  
    MaxPooling2D(2,2),  
    Flatten(),  
    Dense(128, activation='relu'),  
    Dense(10, activation='softmax')  
)  
  
cnn_model.compile(optimizer='adam',  
loss='categorical_crossentropy', metrics=['accuracy'])
```



Community
Consulting
Academy

DECODING Events
DATA SCIENCE .com



RNNs & LSTMs

Best for Sequential Data like
Time Series & NLP



```
from tensorflow.keras.layers import  
SimpleRNN, LSTM
```

```
rnn_model = Sequential([  
    LSTM(50, return_sequences=True,  
input_shape=(100, 1)),  
    LSTM(50),  
    Dense(1)  
])
```

```
rnn_model.compile(optimizer='adam',  
loss='mse')
```



Community
Consulting
Academy



Transformers (BERT, GPT, ViTs)

Best for NLP & Vision Tasks



```
from transformers import  
TFAutoModel
```

```
bert_model =  
TFAutoModel.from_pretraine  
d("bert-base-uncased")
```



Community
Consulting
Academy

DECODING Events
DATA SCIENCE .com



Model Regularization

Prevent Overfitting – Dropout,
L1/L2 Regularization



```
from  
tensorflow.keras.layers  
import Dropout  
  
model.add(Dropout(0.3))  
# Dropout 30%
```



Community
Consulting
Academy



Hyperparameter Tuning

Optimize Model Performance



```
from keras_tuner import  
RandomSearch
```

```
tuner = RandomSearch(model,  
objective='val_accuracy',  
max_trials=5)  
tuner.search(X_train, y_train,  
epochs=10, validation_data=  
(X_test, y_test))
```



Community
Consulting
Academy

DECODING Events
DATA SCIENCE .com



Deploying DL Model

Save & Load a Model



```
model.save("my_model.h5")
```

```
loaded_model =  
tf.keras.models.load_model("my_model.h5")
```



Community
Consulting
Academy

DECODING Events
DATA SCIENCE .com

