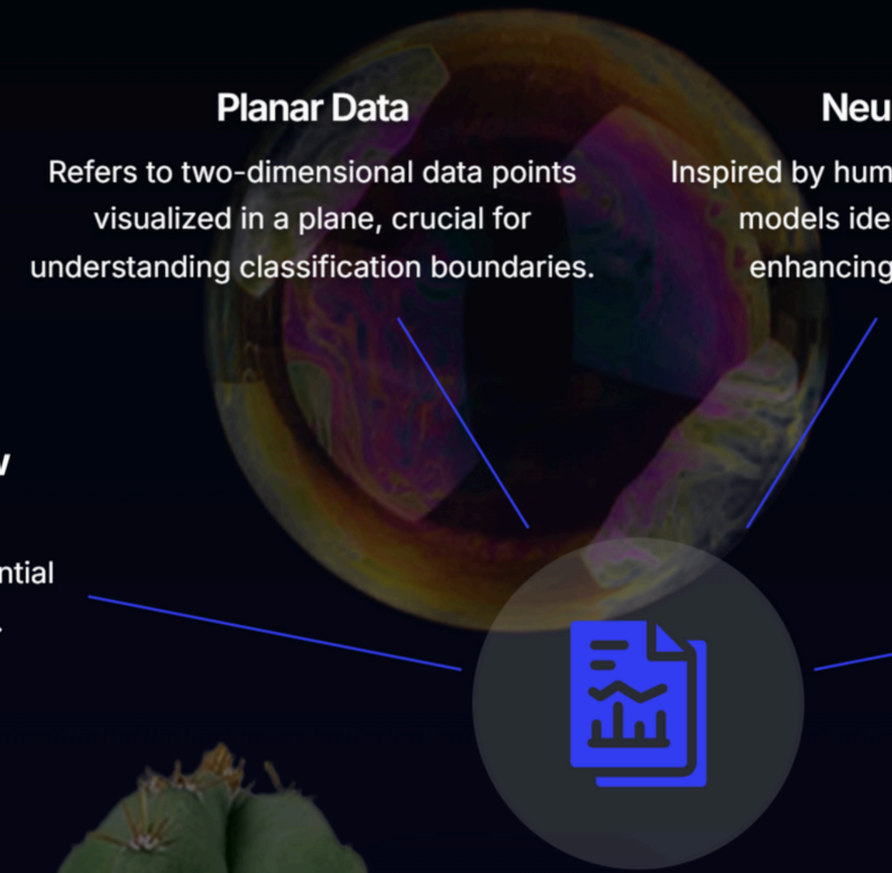
DATA CLASSIFICATION OVERVIEW

Introduction to Data Classification

Understanding the Basics of Data Categorization in Machine Learning



Planar Data

Refers to two-dimensional data points visualized in a plane, crucial for understanding classification boundaries.

Neural Networks

Inspired by human neural networks, these models identify patterns in data, enhancing classification tasks.

Data Classification Overview

This process organizes data into categories for effective use. It's essential for machine learning applications.

Importance of Hidden Layers

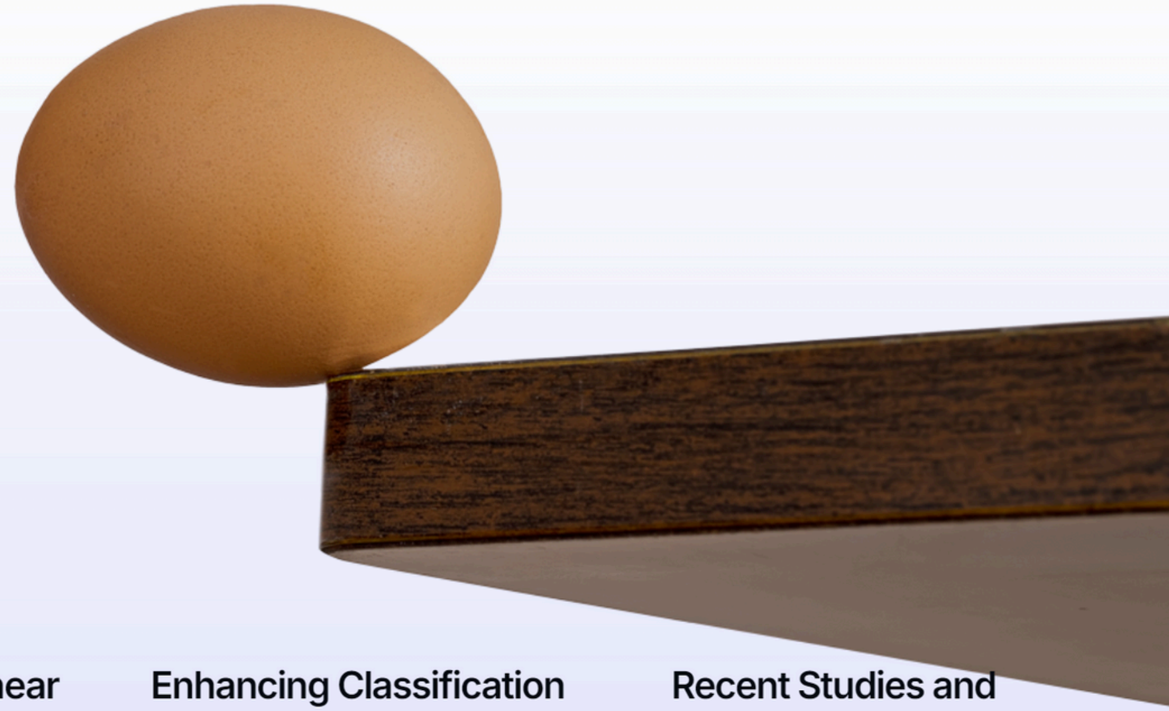
Hidden layers in neural networks refine data classification, enabling the model to learn complex patterns.

BY - TALHA ANSARI
AIML A[47]

HIDDEN LAYER INSIGHTS

Overview of Hidden Layers

Understanding the Role of Hidden Layers in Neural Networks



Definition of Hidden Layers

Intermediate layers that process input data between input and output layers.

Functionality in Transformation

They convert raw input data into a usable format for classification tasks.

Capturing Non-linear Relationships

Enable the network to learn complex patterns, enhancing accuracy in predictions.

Enhancing Classification Accuracy

Their presence significantly improves the model's ability to classify data correctly.

Recent Studies and Validation

Research like Kulbear's (2019) demonstrates their effectiveness in classification tasks.

BY - TALHA ANSARI
AIML A[47]

CLASSIFICATION TECHNIQUES

Techniques for Planar Classification

Exploring various methods for effective data classification and their advantages

Linear Classification

Utilizes a straight line to distinguish between classes, ideal for simple linear separability.

Non-Linear Classification

Incorporates curves to manage complex data distributions, often requiring hidden layers.

Support Vector Machines (SVM)

Employs hyperplanes for effective separation in high-dimensional spaces, maximizing margin.

Neural Networks

Leverages multiple hidden layers to tackle complex, non-linear classification tasks with deep learning.

BY - TALHA ANSARI
AIML A[47]

IMAGE RECOGNITION & NLP

Case Studies: Applications in Image Recognition and NLP

Exploring the Impact of Hidden Layers in Data Classification

High accuracy in image classification.

85%

Neural networks with hidden layers can achieve up to 85% accuracy in classifying images, showcasing their ability to learn complex patterns and features effectively.

Investment in deep learning education.

$40,000

The Coursera Deep Learning Specialization, which includes practical assignments on planar data classification, has been valued at $40,000, reflecting its importance in training professionals in this field.

Effectiveness in NLP tasks.

90%

Hidden layers in models like RNNs and Transformers can result in 90% effectiveness in tasks like named-entity recognition, demonstrating their crucial role in processing natural language.

BY - TALHA ANSARI
AIML A[47]

KEY INSIGHTS

Conclusion and Key Takeaways

Summarizing Key Insights on Planar Data Classification and Hidden Layers

Significance of Data Classification

Organizes data effectively, facilitating easier access and analysis.

01

Role of Hidden Layers

Enhances the capabilities of neural networks, enabling complex feature learning.

02

Techniques for Planar Classification

Includes linear methods and deep learning approaches for more accurate results.

03

Real-World Applications

Utilized in image recognition and NLP, showcasing practical benefits of hidden layers.

04

BY - TALHA ANSARI
AIML A[47]

CODE FOR PLANNER CLASSIFICATION

```
pip install tensorflow
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
# Load and preprocess
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()
# Normalize pixel values to be between 0 and 1
train_images, test_images = train_images / 255.0, test_images / 255.0
# Define the class names
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer',
               'dog', 'frog', 'horse', 'ship', 'truck']
# Build the CNN model
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10, activation='softmax')
])
# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
# Train the model
history = model.fit(train_images, train_labels, epochs=10,
                    validation_data=(test_images, test_labels))
# Evaluate the model
test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=2)
print(f'\nTest accuracy: {test_acc}')
# Plot accuracy and loss over epochs
plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0, 1])
plt.legend(loc='lower right')
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='loss')
plt.plot(history.history['val_loss'], label = 'val_loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(loc='upper right')
plt.show()
```

BY - TALHA ANSARI
AIML A[47]

Refrenced from Understanding the difficulty of training deep feedforward neural networks by Xavier Glorot & Yoshua Bengio