KERAS

A Presentation On Artificial Intelligence Tools

PRESENTED BY :
V. Jayanth Kumar (20A51A1260)

Agenda

- Abstract
- Introduction to Keras
- Architecture of Keras
- Key modules in Keras
- Applications
- Uses
- conclusion

ABSTRACT

Keras is a Python-based high-level neural network API widely cherished by researchers and practitioners in AI. It simplifies the complex task of building neural networks with a modular, user-friendly approach. Its seamless integration with TensorFlow and Theano enhances performance. Keras provides an abstraction layer for quick experimentation with various neural network designs, catering to both beginners and experts. It offers a rich library of pre-defined layers, optimizers, and loss functions, streamlining model creation. Additionally, Keras supports transfer learning and model deployment on cloud platforms and mobile devices, making it a pivotal tool in deep learning research and practical applications. Keras has made deep learning more accessible, efficient, and scalable.

Introduction to keras

Keras is a user-friendly high-level neural networks API in Python, known for its simplicity and modularity. It abstracts complexities, making deep learning accessible to all skill levels. It's compatible with popular frameworks like TensorFlow and offers pre-defined layers and functions, which simplifies model development. Keras supports transfer learning and model deployment, making it valuable for practical applications. With a strong community and ample documentation, it's a go-to choice for building and deploying neural networks.

Architecture of keras

The API of Keras is distributed into three primary categories:

- Model
- Layer
- Core Modules
- In Keras, each ANN is indicated using Keras Models.

Key modules in keras

1. `keras.layers`: Contains various neural network layers (e.g., Dense, Convolutional, Recurrent).
2. `keras.models`: Includes classes for defining and building neural network architectures (Sequential, Model).
3. `keras.optimizers`: Provides optimization algorithms (e.g., SGD, Adam) for training models.
4. `keras.losses`: Contains loss functions for training neural networks (e.g., MSE, cross-entropy).
5. `keras.metrics`: Includes predefined evaluation metrics (e.g., accuracy, precision, recall).
6. `keras.utils`: Provides utility functions for data preprocessing and manipulation.
7. `keras.callbacks`: Customizable callbacks for controlling the training process.
8. `keras.datasets`: Built-in datasets for tasks like image and text classification.
9. `keras.preprocessing`: Tools for data preprocessing and augmentation.

10. `keras.applications`: Pre-trained deep learning models for transfer learning.

Applications

- Image Classification: Keras is frequently used for building convolutional neural networks (CNNs) to classify images. Applications include identifying objects in photos, medical image analysis, and more.
- Object Detection: You can implement object detection models using Keras, such as YOLO (You Only Look Once) or Faster R-CNN, for tasks like self-driving cars, surveillance, and more.
- Recommendation Systems: Building collaborative filtering or deep recommendation models for personalized content recommendations in e-commerce or content platforms.
- Time Series Forecasting: Keras can be used for forecasting in applications such as stock price prediction, weather forecasting, and demand forecasting.
- Speech Recognition: Implementing deep learning models for automatic speech recognition (ASR) and voice assistants using Keras.

- Healthcare: Keras is used in AI applications for medical image analysis, disease diagnosis, drug discovery, and personalized medicine.

Conclusion

Keras provides a high level API for creating deep neural network. In this tutorial, you learned to create a deep neural network that was trained for finding the digits in handwritten text. A multi-layer network was created for this purpose. Keras allows you to define an activation function of your choice at each layer. Using gradient descent, the network was trained on the training data. The accuracy of the trained network in predicting the unseen data was tested on the test data. You learned to plot the accuracy and error metrics. After the network is fully trained, you saved the network model for future use.

Any questions

Thank You