greatlearning
Learning for Life

TensorFlow

Introduction

Agenda

- What is tensorflow?
- Why we are using tensorflow?
- TensorFlow 2.x
- TensorFlow 2.x - features and changes
- Changes with respect to TensorFlow 1.x
- Keras and it's advantages
- Keras vs tf.keras
- Tutorials and guides
- Notes

TensorFlow

- TensorFlow is a machine learning library by Google
- It is open sourced
- It is mainly used for implementing neural networks
- It is an end-to-end platform, which means you can use it for building your models from scratch to deploying them into a production environment

Why TensorFlow for this course?

- Most used library for deep learning
- Easy transition from research to production
- Extensive industry support
- Easy deployment around servers, mobile devices, web platforms etc
- You can easily sort out the issues using GitHub, StackOverflow etc
- Used by world's top AI companies
- Consistently updated with cutting edge changes

Some technical reasons

- Easy and readable syntax
- Being a low-level library it provide more flexibility to developers to implement their own functionalities and services
- Provides high level API for implementing advanced neural net architectures
- Distributed training

OS Support

TensorFlow is supported on following 64-bit systems:

- Ubuntu 16.04 or later
- macOS 10.12.6 (Sierra) or later (no GPU support)
- Windows 7 or later
- Raspbian 9.0 or later

Language support

- Python
- C++
- JavaScript
- Java
- Go
- Swift

New version- TensorFlow 2.x

- Much easier development
- Tight integration with Keras
- More familiar for Python developers
- Deploy anywhere across servers, mobile and edge devices, browser and Node.js with TensorFlow Extended, TensorFlow Lite and TensorFlow .JS
- Multi GPU support

What's new in TF2?

- tf.function(): Creates a callable TensorFlow graph from a Python function
- tf.GradientTape(): Records operations for automatic differentiation
- tf.data(): helps in building complex input pipelines from simple, and reusable pieces

Changes with respect to TensorFlow 1.x

- Cleaning up of APIs: many APIs are either gone or moved
- Eager execution (default): no need manually compile the abstract syntax tree
- No more globals: no need to track variables
- Introduction of function and elimination of session

Model building: from simple to flexible

greatlearning
Learning for Life

1. Sequential API + built in layers -----> New users
2. Functional API + built in layers -----> Engineers with standard use case
3. Functional API + custom layers -----> Engineers requiring control
   + custom metrics
   + custom losses
4. Subclassing: write everything -----> Researchers
   yourself from
   scratch

greatlearning
Learning for Life

Keras

Keras

- Keras is a high-level neural network API
- Written and implemented in Python
- Can run on top of TensorFlow
- It was designed keeping fast experimentation in mind.

Keras advantages

- User-friendly
  - Simple and user friendly interface. Actionable feedbacks are also provided.
- Modular and composable
  - Modules are there for every step, you can combine them to build solutions.
- Easy to extend
  - Gives freedom to add custom blocks to build on new ideas.
  - Cusrom layers, metrics, loss functions etc. can be defined easily.

Keras vs tf.keras

- In TF2 instead of writing “import keras” you will write “from tensorflow import keras”
- In colab, if you see the message “Using TensorFlow Backend”, you are not using tensorflow 2.x implementation of keras
- tf.keras is the TensorFlow’s implementation of keras so it supports all the latest changes in the TensorFlow version
- tf.keras is also better in support and maintenance

Latest tutorials and guides

- https://www.tensorflow.org/tutorials
- https://www.tensorflow.org/guide

Notes

- TensorFlow 2.0 announcement video: https://www.youtube.com/watch?v=EqWsPO8DVXk
- Guide to convert your code from TF 1 to TF 2: https://www.tensorflow.org/guide/migrate
- Detailed introduction video: https://www.youtube.com/watch?v=5ECD8J3dvDQ

greatlearning
Learning for Life

Thank you!

Happy Learning :)