

**Machine Learning with TensorFlow & Keras**

**Course Number:** PYTH-150  
**Duration:** 2 days

**Overview**

Accelebrate's  Machine Learning with TensorFlow & Keras training teaches attendees how to develop real-world machine learning applications powered by TensorFlow and Keras, popular technologies for building production-grade deep learning models. Students also learn how to use an HTTP API to retrieve model predictions.

**Prerequisites**

All students should have programming experience, familiarity with Python 3, and basic knowledge of web-applications (i.e., how the HTTP protocol works). Knowledge of the following topics are useful, but not required:

* Machine learning concepts such as the bias and variance tradeoff, and accuracy estimation methods
* Familiarity with other machine learning algorithms such as Support Vector Machines, Adaptive Boosting (a.k.a. "AdaBoost"), Naïve Bayes, and Logistic Regression
* Knowledge of statistics, probability, linear algebra, and calculus
* Experience with data-provisioning systems, including file systems (local and remote) and databases (SQL and NoSQL)

**Materials**

All TensorFlow & Keras training students receive comprehensive courseware covering all topics in the class.

**Software Needed on Each Student PC**

* Windows, Mac, or Linux
* Anaconda or comparable Python distribution with TensorFlow and Keras installed (Accelebrate will provide instructions)
* PyCharm, Spyder, or comparable Python IDE
* Related lab files that Accelebrate will provide

**Objectives**

* Understand the necessary technical requirements for developing a TensorFlow-powered application
* Gain knowledge on different deep learning architectures and start to develop an intuition on what to chose for your problem
* Experiment with different deep learning algorithms
* Train and evaluate the performance of deep learning models
* Combine your model with Flask to create an HTTP API that returns model predictions

**Outline**

* Introduction
  + Overview neural networks and deep learning
  + Organizing a project directory
  + Setting-up programming environment
    - Installing Python 3
    - Creating a Python virtual environment
    - Installing application dependencies
    - Configuring TensorFlow, Keras, and TensorBoard
* Model Architecture
  + Load and prepare Bitcoin dataset
    - Load CSV file into memory
    - Extracting and exploring dataset features
    - Split dataset into a train and a test set
  + Overview of a TensorFlow computation graph (i.e. a "model")
  + Overview of the Keras interface
    - Instantiating a model
    - Adding components
    - Creating a queue and parallelize computations
    - Using the debugger
  + Deploying TensorBoard
* Model Training and Evaluation
  + cl Training a model
    - Epochs
    - Weights and Biases
    - Hidden layers
    - Learning rate and optimization (i.e. activation functions)
  + Testing a model
    - Measuring accuracy and error rates
  + Determining when to stop
  + Model deployment
* Productization
  + Overview of an (extremely) simple Flask application
  + Creating an HTTP endpoint
  + Retrieving JSON results
* Conclusion