

**Introduction to Machine Learning with Neural Networks**

**Course Number:** PYTH-272
**Duration:** 1 day

**Overview**

This Machine Learning (ML) with Neural Networks training course introduces attendees to ML terminology and explores the architecture of neural networks and how they can be trained to recognize patterns. Attendees learn how to configure, train, and use neural networks.

**Prerequisites**

All students must have Python experience and a basic linear algebra and calculus understanding.

**Materials**

All ML with Neural Networks training students receive courseware covering the topics in the class.

**Software Needed on Each Student PC**

* Windows, Mac, or Linux
* A current version of Anaconda for Python 3.x
* Related lab files that Accelebrate will provide

**Objectives**

* Understand the concepts and terminology surrounding neural networks
* Navigate neural networks layers
* Evaluate the performance of a neural network
* Discuss AI's ethical concerns
* Implement a simple neural network in Keras
* Use the Keras functional API to build a more complex neural network

**Outline**

* Introduction
* Machine Learning with Neural Networks
	+ Arthur C. Clarke’s 3rd Law
	+ What is Machine Learning?
	+ Terminology: Features and Targets
	+ Terminology: Observations (Examples)
	+ Supervised and Unsupervised ML
	+ “Classical” ML and ML with Neural Networks
	+ The Shared Concepts and Principles
	+ AI and Data Science
	+ What is a Neural Network?
	+ Network vs. Model
	+ Positional Types of Layers
	+ Deep Learning
	+ How Does My Network Know Which Problem I Want It to Solve?
	+ The Desired Model Properties
	+ The Artificial Neuron
	+ Perceptron
	+ The Perceptron Symbol
	+ A Breakthrough in Neural Networks Design
	+ Perceptrons and MLPs
	+ A Basic Neural Network Example
	+ Popular Activation Functions
	+ Navigating Neural Networks Layers
	+ A Sample Neural Network Diagram
	+ Model Training
	+ Measuring the Error with the Loss (Cost) Function
	+ Loss Function Properties
	+ Mini-batches and Epochs
	+ Neural Networks Training Steps
	+ The Chain Rule in Calculus
	+ The Chain Rule in Neural Networks
	+ The Gradient Descent Formulation
	+ Applying Efficiencies with Autodiff
	+ Types of Neural Networks
	+ Convolutional Neural Networks (CNNs)
	+ Recurrent Neural Networks (RNNs)
	+ RNN Common and Unrolled Visual Representation
	+ Autoencoders
	+ Neural Network Libraries and Frameworks
	+ Ethical AI
* Machine Learning Concepts and Terminology
	+ Supervised and Unsupervised ML
	+ Self-Supervised Learning
	+ Terminology: Features and Targets
	+ Terminology: Observations (Examples)
	+ Notation for Observations
	+ Data Structures: Tensors
	+ Continuous and Categorical Features
	+ Continuous Features
	+ Categorical Features
	+ Feature Types Visually
	+ Common Distance Metrics
	+ The Euclidean Distance
	+ What is a Model?
	+ Model Parameters and Hyperparameters
	+ Model Accuracy
	+ Loss Functions
	+ Mean Squared Error (MSE)
	+ Mean Absolute Error (MAE)
	+ (Categorical) Cross Entropy Loss
	+ The Cross-Entropy Loss Visually
	+ The softmax Function
	+ Confusion Matrix
	+ The Binary Classification Confusion Matrix
	+ Multi-class Classification Confusion Matrix Example
	+ Feature Engineering
	+ Data Scaling and Normalization
	+ Bias-Variance (Underfitting vs. Overfitting) Trade-off
	+ Bias and Variance Visually
	+ Underfitting vs. Overfitting Visually
	+ Ways to Balance Off the Bias-Variance Ratio
	+ Regularization
	+ Dimensionality Reduction
	+ Model Validation and Avoiding Test Data Leakage
	+ Training Error vs. Validation Error Diagram
	+ Training/Validation/Test Data Split Ratios
	+ Online Glossaries
* Introduction to Keras
	+ What is Keras?
	+ Core Keras Data Structures
	+ Layers in Keras
	+ The Dense Layer
	+ Defining the Layer Activation Function
	+ Models in Keras
	+ Components of a Keras Model
	+ Creating Neural Networks in Keras
	+ The Sequential Model
	+ A Sequential Model Code Example
	+ The Strengths and Weaknesses of Sequential Models
	+ The Functional API
	+ A Functional API Example
	+ The Strengths and Weaknesses of the Functional API
	+ Making New Layers and Models via Subclassing
	+ The Strengths and Weaknesses of Subclassing
* Conclusion