Get Ready to Become AI Engineer

56 lessons

Python Programming Basics

46 lessons

Data Processing and AI/ML Models

72 lessons

Deep Learning with Keras, TF and PT

- Introduction
- **Foundations of Artificial Neural Networks (ANNs)**
- Learning Process in Neural Networks (Incl. Backpropagation)
- Hyperparameters in Neural Networks
- Architecture and Mechanics of Neural Networks
- Custom Models in Keras
- From Neural Networks to Deep Learning
- Training DNNs I-IV (part I)
- Training DNNs V-VII (part II)
- Keras Toolkit: Advanced Techniques
- Deep Learning with IMDB Reviews
- Time to Practice: Custom Models in Keras

- Computer Vision | Applications and Raster Images
- Computer Vision | Deep Learning
- Transfer Learning
- Computer Vision with the CelebA Dataset
- Time to Practice: Pre-trained Models
- TF | TensorFlow
- TensorFlow by Google
- Programming Basics and Operations in TensorFlow
- Building a Simple Neural Network with TensorFlow
- Advanced TensorFlow | Keras API, TensorBoard, and Graph Execution
- TF | Custom Training Loops
- Time to Practice: TF | TensorFlow
- PT | PyTorch
- PyTorch by Facebook
- OOP | Object-Oriented Programming
- OOP | Example Employee Class
- [OPTIONAL] More on OOP | Intermediate
- Iterators & Generators
- Practical Applications in PyTorch
- DataBox Class Project | OOP
- PT | Custom Models
- Time to Practice: PT | PyTorch OOP

Managing AI/ML Pipelines & Systems Deployment

47 lessons

- Guide for My Top 4 Movies (Pyhton)
- Guide for What If (Python)
- Guide for Logic inside us (Python)
- Guide for Organising Operations (Python)
- mastering-git
- git-merged–general
- Python Strings
- Python IO (Input/Output)
- file-handling
- Sets
- Supervised, Unsupervised, and Reinforcement Learning
  Deep Learning with Keras, TF and PT

Foundations of Artificial Neural Networks (ANNs)

Dive into the foundations of Artificial Neural Networks (ANNs), uncovering the inspiration drawn from the human brain and the basic computational units known as artificial neurons. The chapter aims to demystify the structure and functioning of ANNs, illustrating how these models emulate biological processes to perform complex computations.

neurons and **transmits electrical impulses**. With this information at hand, it can develop appropriate responses to stimuli.

Each neuron receives electrical signals through dendrites, which then pass through the body and travel by the axon to other neurons.

Artificial Neurons I

An artificial neuron, also known as a **perceptron**, serves as the fundamental unit in artificial neural networks (ANNs). They emulate the behavior of biological neurons found in the human brain, providing a mathematical model for computational tasks.

In this context, **input data** with n features x_1, x_2, ..., x_n is fed into an artificial neuron. Through **various transformations**, typically achieved with an activation function denoted as f, the neuron processes the input and **generates an output**. This output is then either forwarded to another neuron for further processing or utilized as the final prediction.

> Often, an additional x_0 feature is introduced that allows to include bias/intercept.

Artificial Neurons II

Each **input feature x_i** is associated with its **weight, typically denoted as w_i or $_i$**, which acts as a multiplication coefficient in the **weighted sum z**. In other words, all products of input features and their corresponding weights are summed up.

Subsequently, the resulting value undergoes a transformation through an **activation function ϕ**, and the **output** of this function is then passed further.

A perceptron is a simple `model based on a single neuron` or a set of independent neurons.

The capabilities of perceptrons are very limited – note, that if you skip the activation function (or equivalently use a linear function) you end up with `linear regression` or binary `logistic regression` when sigmoid function is introduced.

Only by `combining artificial neurons into a network`, their capabilities rapidly increase, and the true power of ANN computing is revealed. It is the result of the simultaneous (ie. parallel) operation of many neurons connected in a network.

It creates various structures (also called `architectures`) with perceptron being the simplest neural network.

Having completed this chapter, you now have a solid understanding of the biological inspirations behind ANNs, the basic building blocks that constitute these networks, and some historical context of their development. It is necessary to equip you with the background to understand the nuances of how these models learn and adapt. With this foundation, learners are well-prepared to examine the detailed architecture and mechanics that enable neural networks to tackle a wide array of complex tasks.

What happens if you skip the activation function or use a linear function in a perceptron model?

- ◯ The model becomes a decision tree
- ◯ The model transforms into a k-nearest neighbor classifier
- ◯ The model introduces non-linearities only to binary problems
- ◯ You end up with a linear regression model

Completed

Next Lesson