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

Hyperparameters in Neural Networks

Last but not least... model hyperparameters. Unlike parameters that are learned during training, hyperparameters are set before the training process begins and have a significant impact on the model's learning efficiency and performance. In this chapter, we explore various hyperparameters, specific to neural networks and deep learning algorithms.

- learning rate,
- the number of hidden layers,
- the number of neurons in each layer,
- the number of epochs,
- batch size.

Often we have so much data that we can't upload it immediately to the network. This would be too `memory-intensive` and put a heavy burden on the computer. To deal with this situation, we `divide the data into smaller pieces` and send them to the network one at a time. This updates the network weights after each step to match the model to the data provided.

Each such piece of data is called a `batch`.

Network Hyperparameters II

An `epoch` is a back-and-forth movement of all available data on the network. In one epoch, the model "sees" all possible examples in the set.

Why do we need more than one epoch? The network learning process is iterative, and the more epochs we add, the more times we update the weights. As a result, the model learns patterns in the data with a better approach. Unfortunately, there is no clear answer to the question of how many such epochs there should be. It depends on the complexity of the task or the degree of variability in the data.

Network Hyperparameters III

Consider an example: we have a dataset containing 2000 items. We set the batch size to 50 and the epoch to 100.

As a result, the dataset will be divided into 40 batches. Each of them will contain 50 examples. The weights of the model will be updated after each batch. We will process 40 batches in each epoch. Then we will update the model weights the same number of times.

Module Summary

Learning about neural networks involves understanding the architecture and computational workflow of such AI algorithms, including how they process input data through layers of neurons to make predictions.

Key concepts include:

- `neurons` (basic computational units),
- `activation functions` (e.g., ReLU, Sigmoid, Softmax) that introduce non-linearity,
- `cost/loss functions` (e.g., Cross-Entropy, MSE, MSA) that measure prediction accuracy,
- and `backpropagation`, an algorithm for adjusting network parameters to reduce prediction error.

Neural networks are structured into `input, hidden, and output layers`, where each layer's neurons are connected to the next layer's neurons through weights and biases. The network learns by adjusting these weights and biases using optimization algorithms (e.g., SGD, Adam) based on the gradients calculated during backpropagation.

What is the purpose of dividing data into batches during neural network training?

- ◯ To reduce the number of epochs needed
- ◯ To avoid overfitting
- ◯ To make the training process less memory-intensive and more manageable
- ◯ To ensure all data is seen in every epoch

Submit