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

Deep Learning with Keras, TF and PT

Learning Process in Neural Networks (Incl. Backpropagation)

In this chapter, we will delve into the core mechanisms that enable neural networks to learn from data and improve over time. By understanding how data flows forward through the network to generate predictions and how errors are then propagated backward to update the model's parameters, you will gain insights into the iterative process of learning that drives neural networks' ability to tackle complex problems.

between the predicted outputs and the actual correct values.

Cost/Loss Function II

In the case of `regression problems`, we can use MAE or MSE for this purpose:

However, in the case of `classification` – we will use log-loss, also known as categorical cross-entropy (or binary cross-entropy for two-class problems):

Error Backpropagation I

We know how to calculate the error based on the output and ground truth, but how relate it to each parameter between the layers of a neural network?

For many years, this problem has challenged the ability to effectively train multilayer neural networks.

Among the methods for teaching unidirectional multilayer networks, the `backward error propagation` algorithm dominates. The name of the method reflects the principle of its operation. It consists of `"transferring" the error made by the network` in the direction from the output layer to the input layer (this is done backward relative to the direction of information flow).

Error Backpropagation II

It involves a learning cycle that consists of the following steps:

1. `Evaluating the response` of neurons in the output and hidden layers based on a given input signal.

improve the network's performance.

Error Backpropagation III

The learning process in neural networks enables models to learn from data and improve over time. This process involves several key steps and principles, aiming to minimize the error between the predicted output, Predictions, y' and the actual output, True Values, y.

1. Randomly generate weight vectors.
2. Provide input data features to the input layer.
3. Find the predictions for all samples (forward propagation).
4. Calculate the errors.
5. Backpropagate the errors.
6. Adjust weights.
7. Move back to point 2.

Error backpropagation IV

The error backward propagation is very effective. Unfortunately, it typically requires a long learning time and many iterations. The learning process of a network strongly depends on the selected value of the `learning rate` coefficient η.

A value that is too high often leads to a discrepancy in the process. On the other hand, a value that is too low increases it significantly. Unfortunately, there are no rules that can precisely define its value. It is one of the hyperparameters of the network.

Now you are familiar with the intricacies of the learning process within neural networks, especially the pivotal role of backpropagation. By examining how the algorithm calculates gradients and adjusts the weights and biases to minimize the loss function, you will train NNs to learn from data and improve predictions.

- ◯ It is not effective for large datasets
- ◯ It typically requires a long learning time and many iterations
- ◯ It requires a high number of neurons
- ◯ It cannot handle non-linear problems

Submit

Completed

Next Lesson