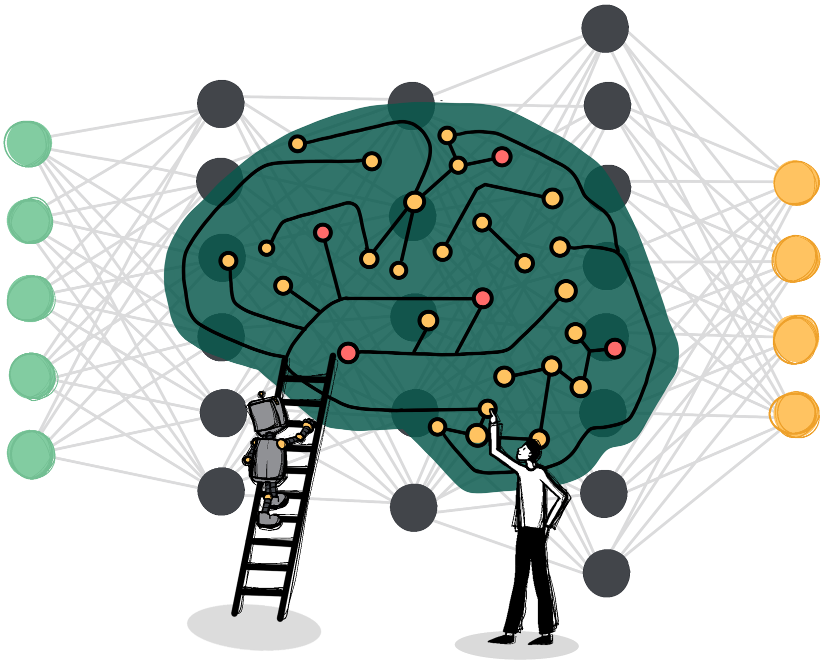
Module 7

Neural Networks

"Each neuron has a so-called ***receptive field****, which is a group of* ***neurons*** *that it is sensitized to* ***activate****."*



About the Module

This module serves as a basic introduction to the topic of neural networks (***NN***). The goal of this lesson is to introduce neural networks and not only show what they can do and how they are used, but also provide insight into how they work internally.

Objectives

Students will be able to…

…explain the basic structure of neural networks and list the basic elements of a neural network (part 1)
…explain how to feed the input data to the neural network. (Part 1)
…explain how data flows through a neural network. (Part 1)
…understand that the different layers correspond to processing steps. (Part 1)
…explain how to read the result of the processing operation. (Part 1)
…explain where the knowledge of a neural network is stored. (part 1)
…explain what changes happen during the learning process. (part 1)
…calculate the degree of freedom of a neural network. (Part 1)
…know that there are plenty of other network architectures for neural networks besides the feed-forward architecture.
…explain how neurons work. (Part 2)
…explain the operation of an activation function. (Part 2)
…explain why the activation function needs to be constrained and how that is implemented. (Part 1 and 2)
…know that different activation functions can be used in different layer types. (Part 2)
…explain the term receptive field. (part 2)
…understand that a neural network can only provide results that can be inferred from the data base. (Part 3)
…understand that smaller neural networks are relatively easy to implement with manageable effort and available technologies. (Part 3)
…understand that neural networks do not necessarily require the use of special libraries or other tools. (Part 3)
…explain how neurons can be used for graphical or numerical tasks. (part 1-3)

Agenda

Time	Content
50 min	Theory - Part 1 - Basics
30 min	Theory - Part 2 - Receptive fields
20 min	Practice - simulator in the manual and automatic variant
25 min	Theory - part 3 - practical minimal example
10 min	Practice - Simulator
15 min	Practice - practical exercise

Part 1 - Basics

This part explains the basic concepts and structure of neural networks and shows how they work using the example of character recognition using a simple feed-forward network.

Introduction and presentation of a feed-forward network

(Slides 2 - 5)

Neural networks as the presumed basis of all thought processes, artificial neural networks as an attempt to reproduce this process in principle but not in detail. Neural networks function fundamentally differently than conventional computers with the usual von Neumann architecture (Arithmetical Logical Unit, Control Unit, Memory, I/O Unit). Introduction of the term "Feed-Forward-Network".

Problem description digit and character recognition

(Slides 2 - 5)

Why is digit and character recognition challenging? What makes it so difficult for conventional programming?

Input data

(Slides 7 - 9)

How is the input data, i.e. in our case the graphically displayed digits and numbers, fed to the network? The pixel brightness levels are interpreted as activations of neurons of the first layer (= input layer).

Layered structure

(Slides 11 - 14)

In the feed-forward architecture, the first layer is the input layer, the last is the output layer, and the layers in between are called hidden layers.

Flow of data through the network

(Slides 19 - 22)

The slides show the flow of data through the network. The activations are passed through the connections and become the activations of the next layer.

Processing steps

(Slides 11 - 14)

The recognition of characters can be imagined step by step in such a way that first simple, more basic units of the character are recognized and then larger, more coherent ones. These units are highlighted in color in each case. One explanation of the way neural networks work is that in a successfully trained network there is a division of labor between the layers, so that the upstream layers (left side) respond to the more basic units and the downstream layers (right side) tend to respond to the larger units.

Details of the implementation and receptive field

(Slides 23 - 25)

The connection between activations and weights is explained here and the term receptive field is introduced. Each neuron has a so-called receptive field, which is a group of neurons that it is sensitized to activate. This is an important concept for understanding how neural networks work.

Restriction function

(Slides 26 - 27)

In order that values of neurons do not grow infinitely into the positive or negative (a small part of the net would be sufficient to make the total result useless) a restriction function is always used. This becomes an elementary component of the activation function.

Pre-excitation/bias and total activation formula

(Slides 28 - 31)

Explanation of which role the bias plays in the operation and indication of the complete activation formula consisting of the constraint function, the activations, the weights and the pre-excitations.

Tasks and solutions

(Slides 32 - 33)

Using three questions, we recapitulate some fundamental properties of a neural network.

Reference to other network architectures and links

(Slides 26 - 27)

The feed-forward network architecture has been presented here because it is a very simple and commonly used architecture. The graphic is only meant to indicate that a plethora of other architectures are being used and tested.

Material

- Neural_Networks_I_-_Basics.pdf

Part 2 - Receptive fields

This part deepens the notion of receptive field already introduced in part 1 Using an interactive example of object recognition with a 2x2 pixel camera. This part of the lesson allows an interactive understanding of the operation of a neural network with the simulator.

Camera

(Slides 2 - 5)

A simple neural network gives a camera equipped with a 2x2 pixel sensor the ability to recognize four different patterns. As with character recognition before, the problem is more challenging than it first sounds. Patterns of the same category may not have pixels in common. As before, the brightness of the pixels becomes the activation of the neurons in the input layer.

How neurons work

(Slides 6 - 13)

In these slides, we will again go into detail about how the neurons work: Forming the weighted sum of the inputs and constraining the values. The details of the sigmoid function form an extended content for more advanced students.

Completing the network

(Slides 14 - 16)

The started layer is completed, an additional layer of the same layer type is added and finally another layer with alternative activation function. For activations and weights only 0, 1, -1 are used in sequence. This is a drastic restriction and only works because this is a very simple use case. However, it makes the calculation much easier. It allows us to follow the computational process through the simple neural network or even perform it ourselves. In practice, it would not be a good idea to use only 0, 1, -1. This only works for this simple example.

Alternative activation function

(Slide 17)

The details of this activation function are again to be seen as extended content. Both require a firm understanding of the concept of functions and the graphical representation of functions. The most important property of the ReLU function in our example is that it does not pass negative values. This property is used in the last step to successfully calculate the result.

The application of the two activation functions is simplified considerably by the trick that only the values 0,1,-1 are used. Most important property of the mainly used activation function (Sigmoid) is that values larger than 1 are reduced to 1 and values small -1 are reduced to -1. Most important property of the alternative activation function (ReLU and leaky ReLU) is that negative values are reduced to 0 (or close to zero).

The network and the camera in use

(Slide 18)

This slide shows the finished system. A vertical pattern is added and also correctly recognized. For each neuron the receptive field is indicated. This is for each neuron the set of input neurons from which it can receive signals directly or indirectly. The receptive fields increase from left to right from size 1 to size 4. In each case it is also significant whether the neuron reacts to positive (white field) or negative activation (black field) in its receptive field.

Simulator

The simulator can be used to simulate the processing steps of the network. The simulator is available in a manual and an automatic version. In the manual version, all activations are determined and set by the user on the basis of input data (=activations 0,1,-1) and connections (0,1,-1). The automatic version of the simulator calculates all necessary activations and outputs from the input data and is of course much more comfortable to handle. The challenge for the operator is much higher with the manual simulator, but the learning effect is much greater. Talented or advanced students should be familiarized with the manual simulator first.

Material

- Neural_Networks_II_-_receptive_Fields.pdf
- NN_ii_simulator.zip

Part 3 - Practical Minimal Example

This part presents an interactive practical example of a neural network consisting of only 5 neurons. The special feature of this part of the lesson is that it is a minimal but in all phases self-contained small project. No external libraries or other tools are used. Training data, calculation process, training are manageable, clearly documented and immediately comprehensible in the Python source code. The project serves very well the purpose of demystifying the field of artificial intelligence. The project can also serve as a basis for further projects, for example one could use another data basis or extend the network also structurally: additional input and output possibilities, additional layers … . However, a structural change also requires a change in the training process.

Presentation of the neural network

(Slides 2 - 3)

Introduction, graphical representation and explanation of the structure of the neural network.

Data basis and output

(Slides 4 - 5)

Graphical representation of the relationship between body weight, height and gender. This is only to illustrate the relationship between these variables. However, this is not the training data. The network is only trained with 4 data samples (see Python program).
Due to the overlapping of the two groups, clear results are not always to be expected. Explanation of how to interpret the activation of the output layer as a result.

Calculation course

(Slide 6)

Here the exact data flow through the network and the calculation process is explained.

Restriction function

(Slide 7)

Here the necessity of the constraint function is explained.

Reference to training and simulator

(Slide 8)

Training represents advanced content. However, for advanced students iii_training.pptx provides a good overview to complete the example. The Python source code also provides information about the details of the training.

The simulator allows all students to interactively sample values and read the result.

The inputs are made via 2 sliders. The activation of the neurons is output numerically at the neuron and additionally visualized with gray scales. Black stands here for activation 0, white for activation 1. The strength of a weighted connection is symbolized with the line width. Positive connections are colored black, negative connections are colored red.

Practical exercise

The participants can now use the acquired knowledge to calculate their own example inputs. Due to the minimal, simple structure of the network, a manual calculation is easy to perform.

3 pairs of values are given for which the participants can calculate results. In the bottom line, own values can be used.

Sources and reference to the program

(Slide 9)

The underlying Python program is available in 2 versions, in which different activation functions are used.

Material

- Neural_Networks_III_-_practical_Example.pdf
- NN_iii_simulator.zip
- NN_exercise.zip
- NN_additional_content.zip

ENARIS
Education and Awareness
for Intelligent Systems

EUROPEAN UNION

Interreg
Austria-Hungary 2014-2020
European Union – European Regional Development Fund

TU
Graz

mobilis
interaktív kiállítási központ

OESTERREICHISCHE ®
COMPUTER GESELLSCHAFT
AUSTRIAN
COMPUTER SOCIETY