Introduction

This three-day course gives an introduction to the fundamentals of Deep Neural Networks using the KNIME® environment, specifically applied to QSAR. The training is intended for people already familiar with KNIME® and who have previous experience in data analysis. The main theme that will be addressed throughout the course is the implementation of Deep Neural Networks applied to the fields of classification, regression and unsupervised learning in the KNIME® environment. This training is based on the Keras library nodes, which allow the user to interact with the Keras Deep Neural Networks and Machine Learning algorithms through KNIME.

Who is this course for?

The training is intended for scientists who want to learn the basis of deep neural networks for the first time, with generic practical examples in the KNIME® environment to illustrate the theory.

You will learn how to . . .

•Implement Deep Neural Networks from scratch based on the Keras library.

•Differentiate between the various DL architectures (Dense, Convolutional, Recursive, Generative, etc.) and choose the most appropriate for your type of data.

•Evaluate and compare the different DL architectures with different parameter settings and structures.

To follow this course you should . . .

•Have followed the Pikaïros training courses “Introduction to Data Mining through KNIME®” (IDMK), “Introduction to Data Analysis with KNIME®” (IDAK) and “Advanced Data Mining through KNIME®” (ADMK) or have equivalent in-depth experience in Data Mining, Data Analysis and in the use of the KNIME® environment.

•Have a minimum experience in programming with a structured programming language such as C or Python, in order to understand the code used in the Python scripting nodes of KNIME.

•Understand basic computer English (the KNIME® interface is in English).

Terms, access time and adaptation to people with disabilities

This course is entirely based on the KNIME educational software with the implementation of workflows as a learning method.

A continuous evaluation by MCQ is carried out on the exercises proposed during the training.

This course is only offered face-to-face at the client's premises.

Possibility of adapting the time and mode of training to people with disabilities.

This course should be booked at least 1 month in advance of the desired training date.

Full programme

First day

Introduction to the theory and practice of Deep Neural Networks (DNN)

What is Deep Learning (DL)?

- Relationship between Artificial Intelligence (AI), Machine Learning (ML) and Deep Neural Networks (DNL)
- Short history of DL up to the present day

Practical basics of Deep Learning Tools in KNIME

- Choice of Keras and Python among KNIME® libraries for Deep Learning
- Installing Anaconda, Python and Keras in a Windows environment
- Setting up KNIME® to access Keras and Python
- Setting up KNIME® to access RDKit
- Testing KNIME® Deep Learning environment with an example workflow

Theoretical basis for understanding Deep Learning

- Definition of Neuron, Layer of Neurons and Deep Neural Network
- Importance and significance of Layers in a DNN
- Introduction to Basic Mathematical notions to understand DNN
- How to build a DNN in KNIME® using Keras nodes
- How to build the same DNN using Python nodes

Exercises: Study and implementation of a DL Neural Network for 1D & 2D data

- Implementation of a single hidden layer DNN for 1D data classification with Keras nodes
- Improvement of DNN classification performance by adapting its structure and parameters: number of neurons and layers
- Implementation and improvement of the same DNN with Python nodes

First day

Introduction to the theory and practice of Deep Neural Networks (DNN)

Basics for creating and using a DNN

- Definition of Loss function & Optimizer
- Definition of Learning, Under-Fitting, Over-fitting and Regularization
- Definition of Gradient Descent, Learning Rate and Backpropagation
- Definition of Normalization, Epoch, Batch Learning and Stochastic Learning
- Regularization of a DNN by Early-Stop and Drop-Out

Generic Dense multi-layer Neural Networks

- Basic neural layers, their function and use in Keras and Python in KNIME
  - Input, Reshape, Dense, Activation and Output
- Useful neural layers, their function and use in Keras and Python in KNIME
  - Activation with Regularization, Augmentation, DropOut, Repetition

Exercises: Improvement of the first DNN for 1D and 2D classification

- Influence of the number of Epochs on learning and over-fitting
- Application of Early Stopping, Normalization, Drop Out and Regularization functions to control over-fitting
- Application and comparison of different optimization functions
- Evaluation of the performance of DNNs in different configurations
- Discovering the training of a DNN in multitasking

Second day

Discovery of Keras and Python nodes for the implementation of Convolutional Neural Networks in KNIME®

Convolutional Deep Neural Networks (CNN)

- Definition and architecture of a CNN
- Implementation of a CNN with KNIME® and Python nodes
- Neural layers used in CNN, their function and implementation in KNIME®
- Improvement of the results of a 2D DNN by the Augmentation technique

Exercises: Study and construction of a CNN for 2D classification

- Construction of a CNN with convolutional layers
- Improve DNN classification performance by stacking Convolutional layers
- Construction of an autoencoder with and without noisy data
- Improve DNN classification performance by augmenting data
- Evaluation and comparison of different CDNN.

Third day

Discovery of Keras and Python nodes for the implementation of Recursive Neural Networks in KNIME®

Recursive Deep Neural Networks (RNN)

- Definition and Architecture of an RNN
- Definition of RNN with KNIME® and Python nodes
- Definition of the Long-Short-Term Memory (LSTM) neural network
- Structure of an RNN based on LSTM neurons

Exercises: Study and construction of an RNN in temporal data prediction

- Construction of an RNN with the Keras Simple RNN Layer node
- Construction of an RNN with the Keras LSTM Layer node

Discovery of Keras and Python nodes for the implementation of Generative Adversarial Networks (GAN) in KNIME®

Generative Adversarial Networks (GAN)

- Definition and architecture of a GAN
- Definition of a GAN with Jupyter and Python
- Structure of a simple GAN network

Exercises: Study and construction of a GAN

- Construction of a GAN with Jupyter Notebook & Python