

Machine Learning and More

O2A Data Flow Solutions

Giorgio Busatto

Computing and Data Centre

27th May 2020

Overview



- 1 Definitions and Background
- 2 Popular Python Libraries for ML and DL
- 3 ML in the Data Centre Infrastructure
- 4 Literature

Artificial Intelligence



Definition (attempts) by Russell and Norvig (see [RN10]):

Acting Humanly “The study of how to make computers do things at which, at the moment, people are better.” (Rich and Knight, 1991)

Thinking Humanly “[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . .” (Bellman, 1978)

Thinking Rationally “The study of the computations that make it possible to perceive, reason, and act.” (Winston, 1992)

Acting Rationally “AI . . . is concerned with intelligent behavior in artifacts.” (Nilsson, 1998)

Branches of Artificial Intelligence (1)



Classification from [RN10]:

- *Problem-solving*: search, constraint satisfaction, . . .
“... agent [that] can find a sequence of actions that achieves its goals when no single action will do.”
- *Knowledge, reasoning, and planning*: logic-based knowledge representation and inference.
“... agents that can form representations of a complex world, use a process of inference to derive new representations about the world, and use these new representations to deduce what to do.”
- *Uncertain knowledge and reasoning*: probabilistic reasoning.
Construct “models to reason under uncertainty according to the laws of probability theory.”

Branches of Artificial Intelligence (2)



Classification from [RN10] (continued):

- *Communicating, perceiving, and acting*: natural language processing, (visual) perception, robotics.
- *Learning*: logic-based learning (deductive), probabilistic model learning, learning from examples (inductive, machine learning).
“An agent is learning if it improves its performance on future tasks after making observations about the world.”

Machine Learning



Definition by Müller and Guido in [MG17]:

- *Machine learning* is about extracting knowledge from data.
- It is a research field at the intersection of *statistics*, *artificial intelligence*, and *computer science* . . .
- and is also known as *predictive analytics* or *statistical learning*.

Machine Learning Concepts



“... from a collection of input-output pairs, learn a function that predicts the output for new inputs.” [RN10].

- *Supervised learning*:
 - *Features* (inputs): vector of numeric values,
 - *Target* (output): continuous or discrete numeric values,
 - *Classification*: target is a discrete value,
 - *Regression*: target is a continuous value.
- *Unsupervised learning*:
 - *Features*: vector of numeric values,
 - *Clustering*: grouping objects according to similarities,
 - *Anomaly detection*: detect samples which differ from the majority of the data,
 - *Dimensionality reduction*: reducing the number of random variables under consideration.

Supervised Learning Models



- Linear models:
 - *logistic regression*,
 - *linear SVC*,
 - ...
- *Decision trees and random forests*.
- *Neural networks: multilayer perceptron, ...*
- *Deep neural networks (deep learning, see [PG17]):*
 - *convolutional neural networks (CNN)*,
 - *recurrent neural networks (RNN)*,
 - ...

Note: For [MG17], *all* neural networks are deep learning.

Python Data Science Libraries



- Python data science stack:
 - numpy: <https://numpy.org/>,
 - pandas: <https://pandas.pydata.org/>,
 - scipy: <https://www.scipy.org/>.
- Scikit-Learn (ML toolkit): <https://scikit-learn.org>.
- Python DL libraries / frameworks:
 - TensorFlow (TF, <https://www.tensorflow.org>): ML and DL framework,
 - Keras (<https://keras.io>): unified API for ML backends (supports Tensorflow).
- Examples in Jupyter notebooks.

Resources at the Data and Computing Centre



- Computing:
 - Own PC: for fast prototyping on small / reduced datasets.
 - JupyterHub (<https://jupyterhub.awi.de>) or virtual machine on the VMWare test environment.
Training on larger datasets, interactive work possible: No waiting time, no timebox.
 - HPC: Cray (ollie). Very large datasets, hyperparameter tuning.
Batch jobs are the preferred way to work, time-boxed.
- Storage:
 - Online storage (Isilon): accessible from VMs and PC.
See eResources at <https://cloud.awi.de> (My projects).
 - Cray has its own storage: data must be copied.
- <https://intranet.awi.de/infrastruktur/rechenzentrum/platforms.html>.

A use case



- Classification of LOKI (see below) images using image descriptors.
- Classification of LOKI images using DL methods (CNN).
- Scope:
 - Experience with a concrete, scientifically relevant example.
 - Comparing different methods, hardware requirements, running times, and so on.
- Not in scope (yet):
 - To produce a model that can compete with state-of-the-art models.
 - To produce a complete system that can be used by the scientists.

Lightframe On-sight Keyspecies Investig. (LOKI)

**a**

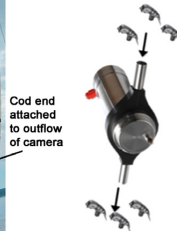
1) Plankton concentration net



2) LOKI computer with SSD drive and sensors:
- Temperature
- Conductivity
- Oxygen
- Fluorescence
- Pressure

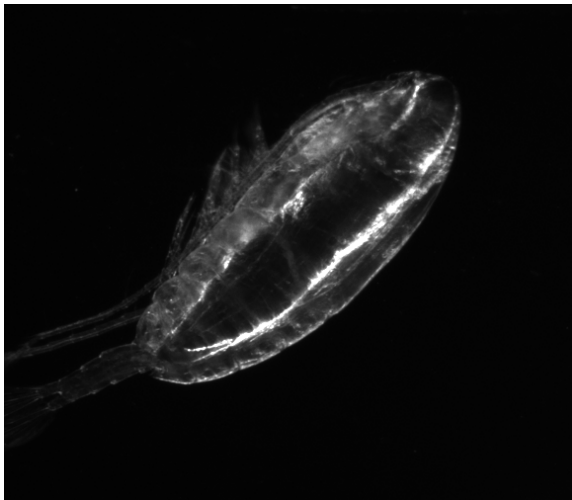
3) LOKI camera

4) Battery

b**c**

Cod end attached to outflow of camera

Example of LOKI image



Applications and open issues



- Used by biologists to study the abundance of plankton species.
- The classification of the images is performed by the scientists.
- Software-aided, partially automatic classification is supported, e.g. by Ecotaxa (<https://ecotaxa.obs-vlfr.fr/>).
- Time-consuming due to large datasets (many thousands of images).
- Better (semi-)automatic classification techniques would be welcome.

Classifying LOKI Images with Descriptors



- Use pre-computed image descriptors exported from Eco-Taxa, e.g. area, Fourier descriptors, . . .
- Several sklearn models: K-neighbors, decision trees, random forests, MLP.
- Training was performed on a laptop with about 2000 samples.
- Results:
 - Some classes achieve quite high accuracy (0.90) with some models, whereas others are still too poor (less than 0.50).
 - Training time is relatively short: about 45 seconds for all the models.

Classifying LOKI Images with CNNs



- Bachelor thesis of S. Mahler, University of Applied Sciences Bremen, 2019.
- Training was performed on VMs using Python / Keras.
- Results:
 - Standard CNN models (e.g. ResNet101, InceptionV3) give promising results for the classification of LOKI images.
 - Models must be trained from scratch: pre-trained models (*transfer learning*) have poor performance.
 - Best results with self-trained InceptionV3 model: 46 epochs, about 4h/*epoch* (avg.), 0.82 F1-score.

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$




Summary for both Methods



- Descriptor-based models are faster and provide comparable precision for certain classes.
- Directions for improving both kinds of models:
 - better (larger and more balanced) training datasets,
 - parameter tuning,
 - better features, more descriptors, hybrid model (descriptors + DL).
- Jupyter notebooks are available for both methods, see documentation at <https://spaces.awi.de/display/DM/ANALYTICS> or github: <https://github.com/o2a-data/o2a-data-ml>.

References I



-  Andreas C. Müller and Sarah Guido, *Introduction to machine learning with python*, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2017.
-  Josh Patterson and Adam Gibson, *Deep learning*, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2017.
-  Stuart J. Russell and Peter Norvig, *Artificial intelligence*, Pearson Education, Inc., Upper Saddle River, New Jersey 07458., 2010.