COMPUTATIONAL INTELLIGENCE

AUTOENCODERS
for feature extraction

AGH

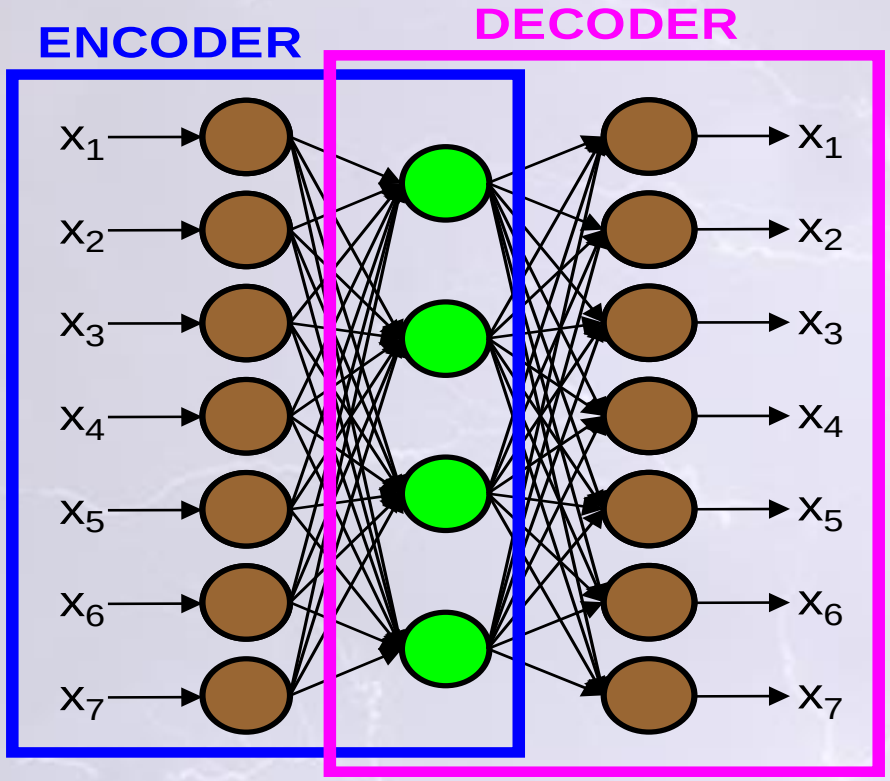
Adrian Horzyk

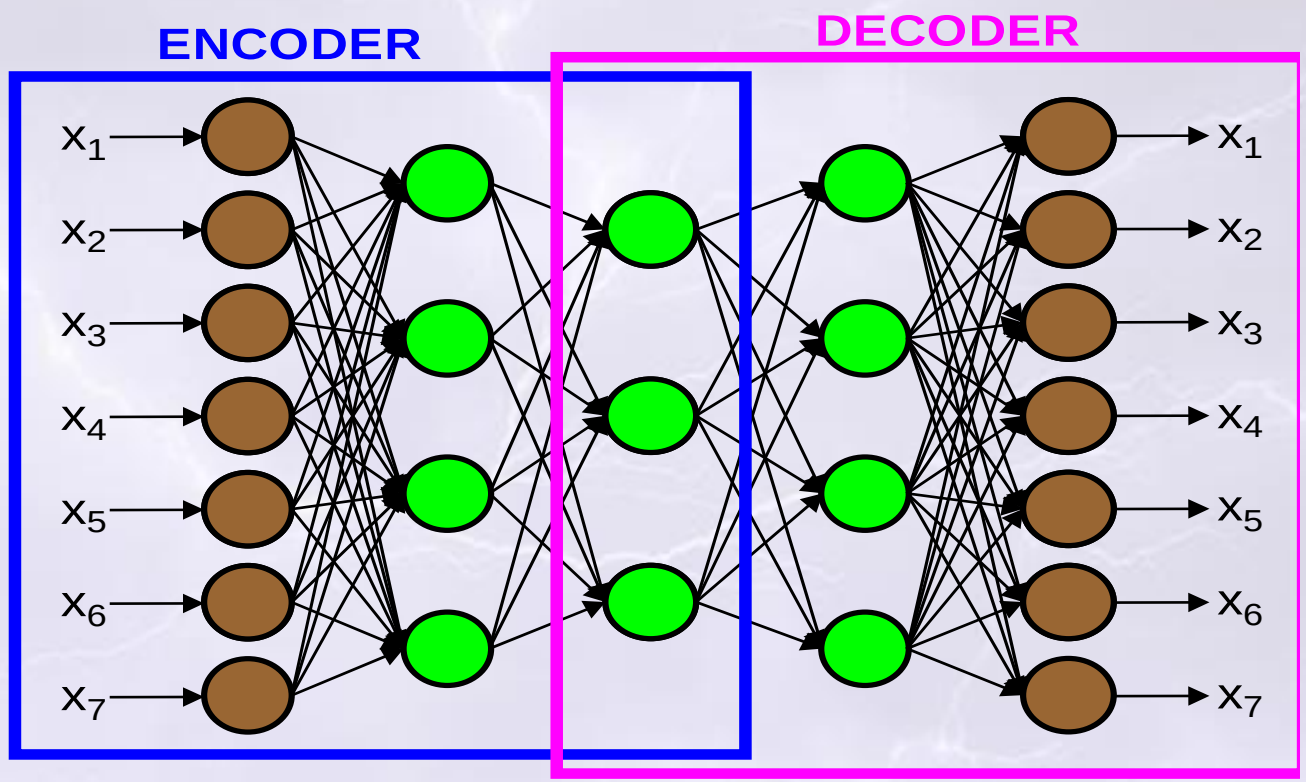
Autoencoders

Autoencoder is a kind of artificial neural networks which is trained to represent a set of training data in an unsupervised manner using a reduced dimensionality and gets the same output data as input ones.

The reduced dimensionality is used to find out frequent combinations which constitute complex data features which can be used in various classifiers.

Autoencoders consist of encoders and decoders:





Types of Autoencoders

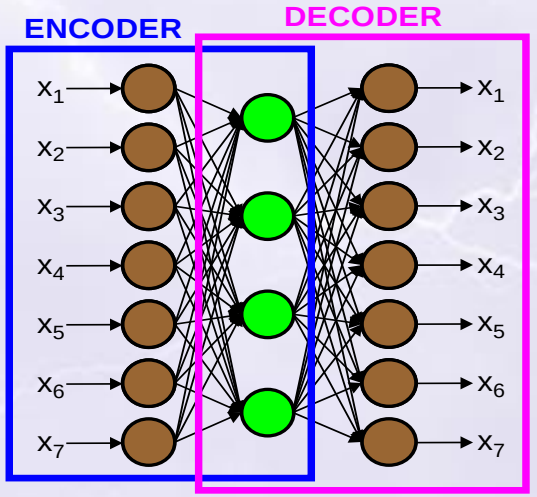
We can distinguish a few types of autoencoders:

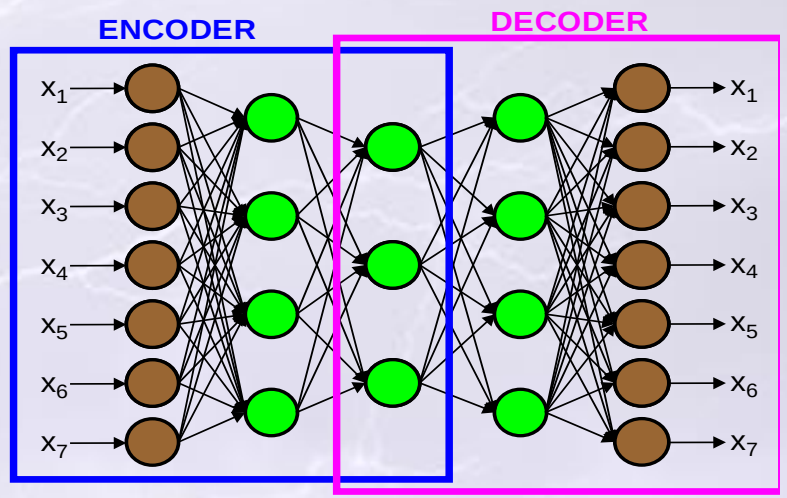
- ✓ **Undercomplete Autoencoders** – are defined to represent data in undercomplete way, i.e. the outputs do not reproduce inputs precisely in order to allow for generalization, feature extraction, data distribution, and correction of outliers. Training of such autoencoders aims to minimize the **loss function** defining the differences between outputs and inputs. When the autoencoders are linear, they work similarly to PCA (Principal Components Analysis), so they can replace such kind of preprocessing algorithms (PCA or ICA).
- ✓ **Autoencoders with Regularization** – use the complexity of the modeled distribution of the data to select an adequate dimension and capacity of encoders and decoders. They use a loss function to be resistant to noise and missing data, and learn correct data distribution. These autoencoders can be non-linear and overcomplete as well.
- ✓ **Sparse Autoencoders** – are autoencoders which are used for other computational tasks, e.g. for classification, where we need to represent frequent features more than find a perfect identity function. In this approach, representation of rare features is penalized. This leads to a sparse representation of inputs and useful feature extraction as a preparation phase for classification.
- ✓ **Anomaly Detection Autoencoders** – are autoencoders which are used to detect rare features that stand for various anomalies in data and can identify outliers.
- ✓ **Denoising Autoencoders (DAE)** – try to find a function which returns correct output for noised, corrupted or incomplete inputs. They have to recover the original undistorted inputs on their outputs.

Training of Autoencoders

Autoencoders are trained in an unsupervised way using the algorithm typically used for supervised learning, e.g. backpropagation. This is because we use the outputs which are the same as the inputs.

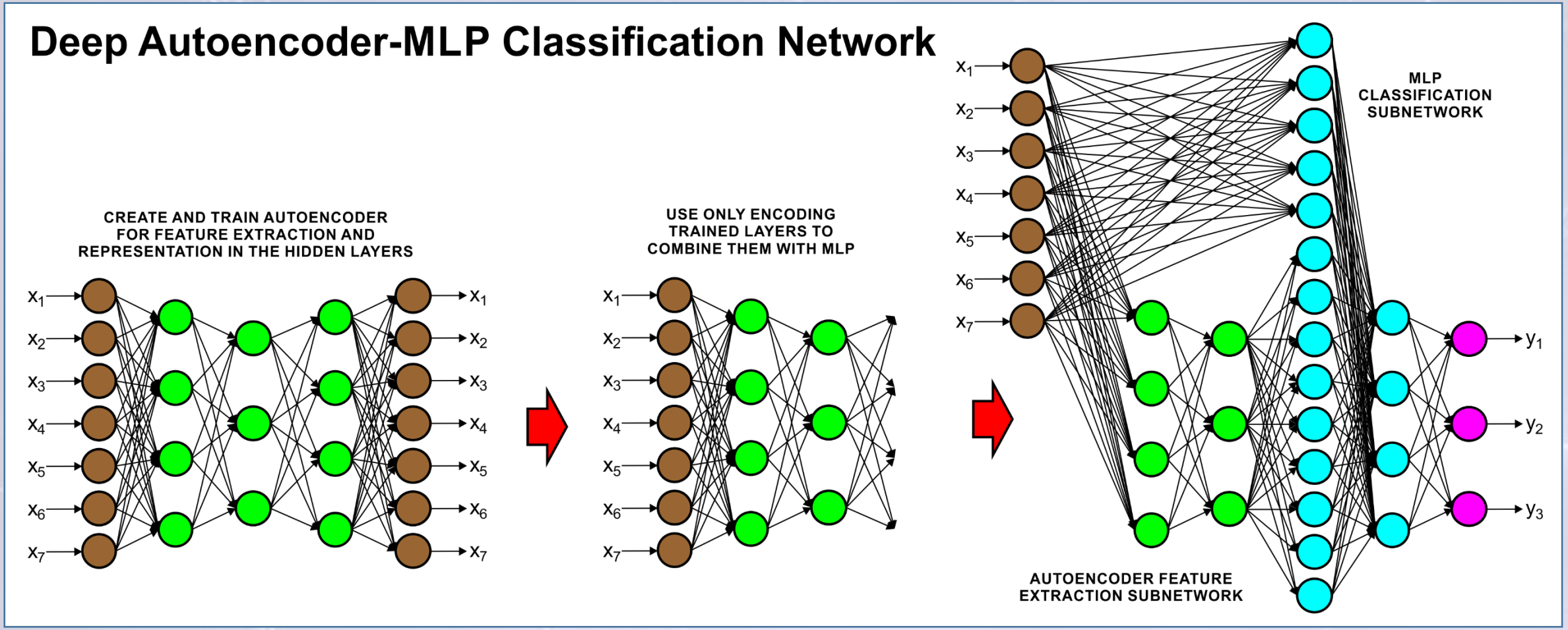
- ✓ **Assume that we have a set of unlabeled training examples $\{x_1, x_2, x_3, \ldots\}$, where $x_i \in R^n$.**
- ✓ **An autoencoder uses outputs defined as $y_i = x_i$ where y_i is an expected output value.**
- ✓ **Autoencoders can learn to extract features similarly as Convolutional Neural Networks (CNN) do.**
- ✓ **The training capabilities of autoencoders are associated with the number of encoding and decoding layers. When autoencoders have more than single encoding and decoding layers, we call them deep autoencoders. Deep autoencoders usually have a better compression ratio than flat autoencoders.**
- ✓ **Deep autoencoders can be constructed from flat autoencoders trained subsequently and separately.**
- ✓ **Autoencoders are usually trained using the backpropagation algorithm, however, we can also use other algorithms, e.g. the recirculation algorithm.**





Combining Autoencoders with MLPs

Sparse Autoencoders are often trained to be combined with other types of artificial neural networks, e.g. MLPs. This is because they can preprocess raw input data and extract useful features for other networks:

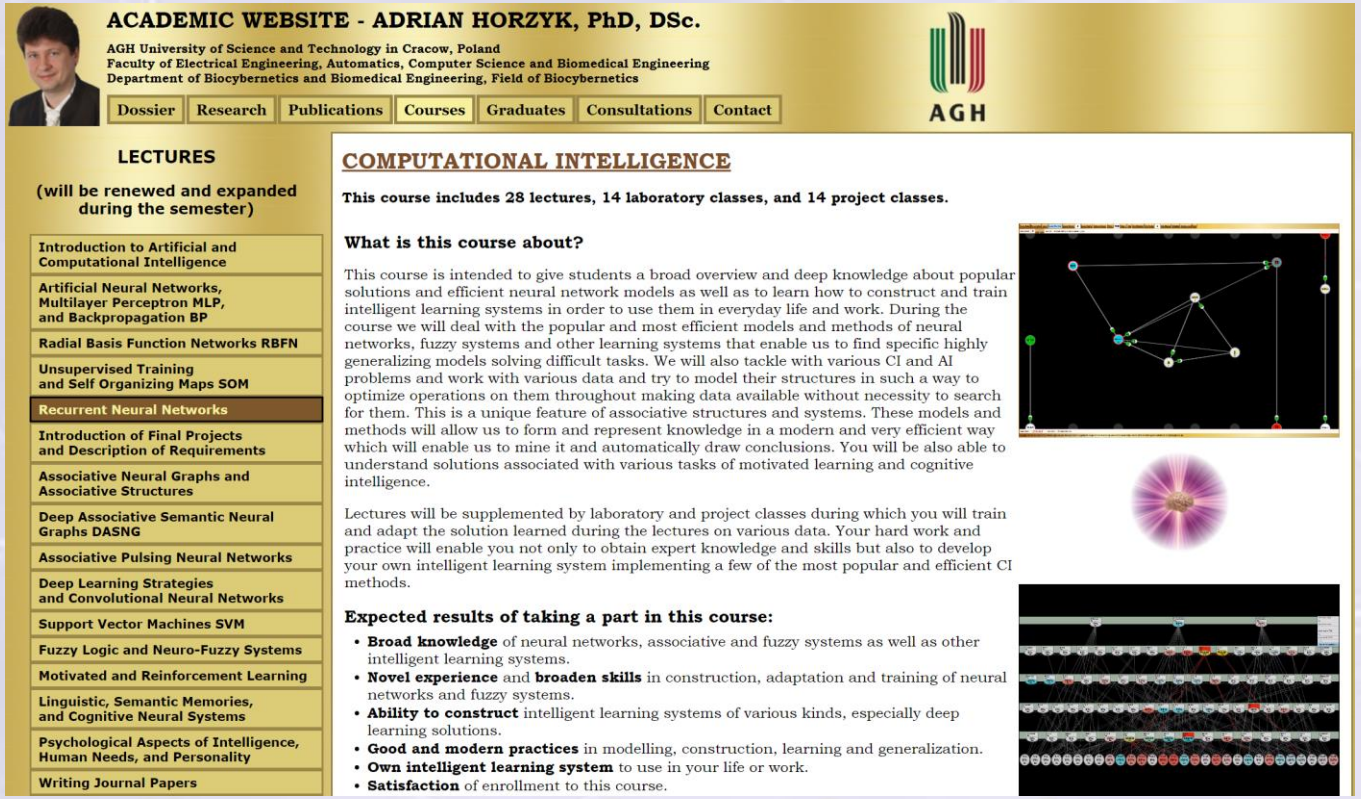


One of our goals during laboratory classes will be to implement such a combination of an autoencoder and MLP.

BIBLIOGRAPHY AND REFERENCES

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016, ISBN 978-1-59327-741-3 or PWN 2018.

2. Stanford University Tutorial of Unsupervised Learning used to Autoencoders: http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/



http://home.agh.edu.pl/~horzyk/lectures/ahdydci.php