

---

# 3RD EXERCISE SESSION

---

## Exercise 1: Multi-Layer perceptron for Breast cancer Tumor classification

Following Exercise 2 from 1<sup>st</sup> exercise session, use a neural network to predict the class of the tumor.

- a) Use the pre-processed data from Exercise 2 from 1<sup>st</sup> exercise session.
- b) Use one of neural network libraries available in your chosen programming language to initiate a neural network for classification. (*In Python: Sklearn, Tensorflow, ...*)
- c) Try different architectures and compare their performances on both training and testing data.
  1. Start with one hidden layer with (5, 10, ...,30) nodes and evaluate the performance for each number of nodes.
  2. Increase the number of layers gradually from 1 to 10 layers and evaluate the performance for each number of layers. (You can choose the number of neurons per layer)
  3. Put the results in a table that you will return with this exercise.
- d) How does the performance related to training and test data improve when the number of nodes and layers increase?
- e) With 4 hidden layers 10 neurons each, implement a dropout at each layer (try different dropout rates from 0% to 40% and evaluate their performances). What is the best dropout rate?
- f) With 2 hidden layers 10 neurons each, use at least 3 different activation functions in the hidden layers and compare the results. What is the best activation function?

\*\*Suggested tutorial on neural networks with sklearn: [https://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html](https://scikit-learn.org/stable/modules/neural_networks_supervised.html)

\*\*Suggested tutorial on neural networks in TensorFlow: <https://pythonprogramminglanguage.com/tensorflow-neural-network/>

\*\*Suggested tutorial on neural networks in Matlab: <https://se.mathworks.com/help/thingspeak/create-and-train-a-feedforward-neural-network.html>