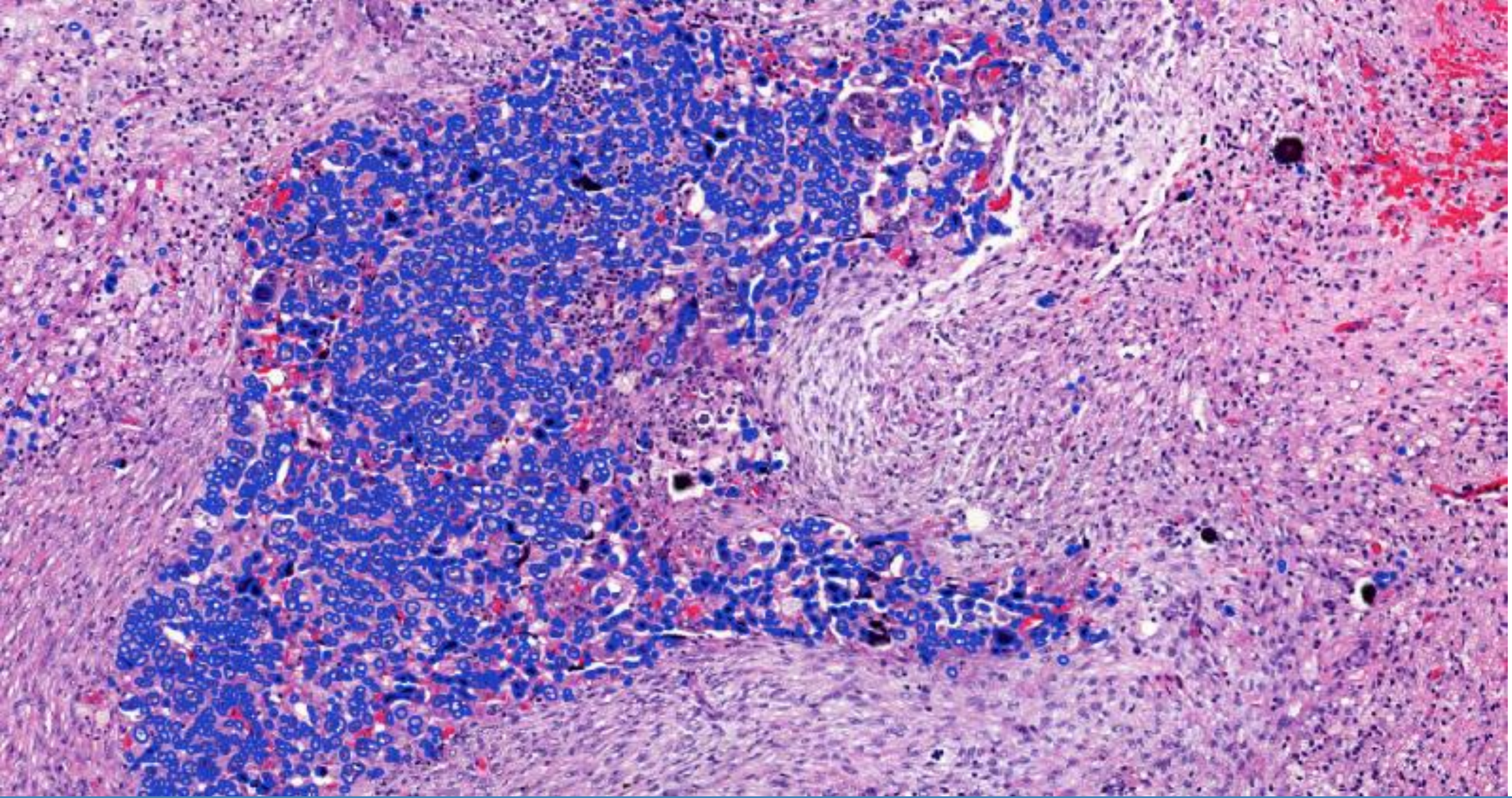


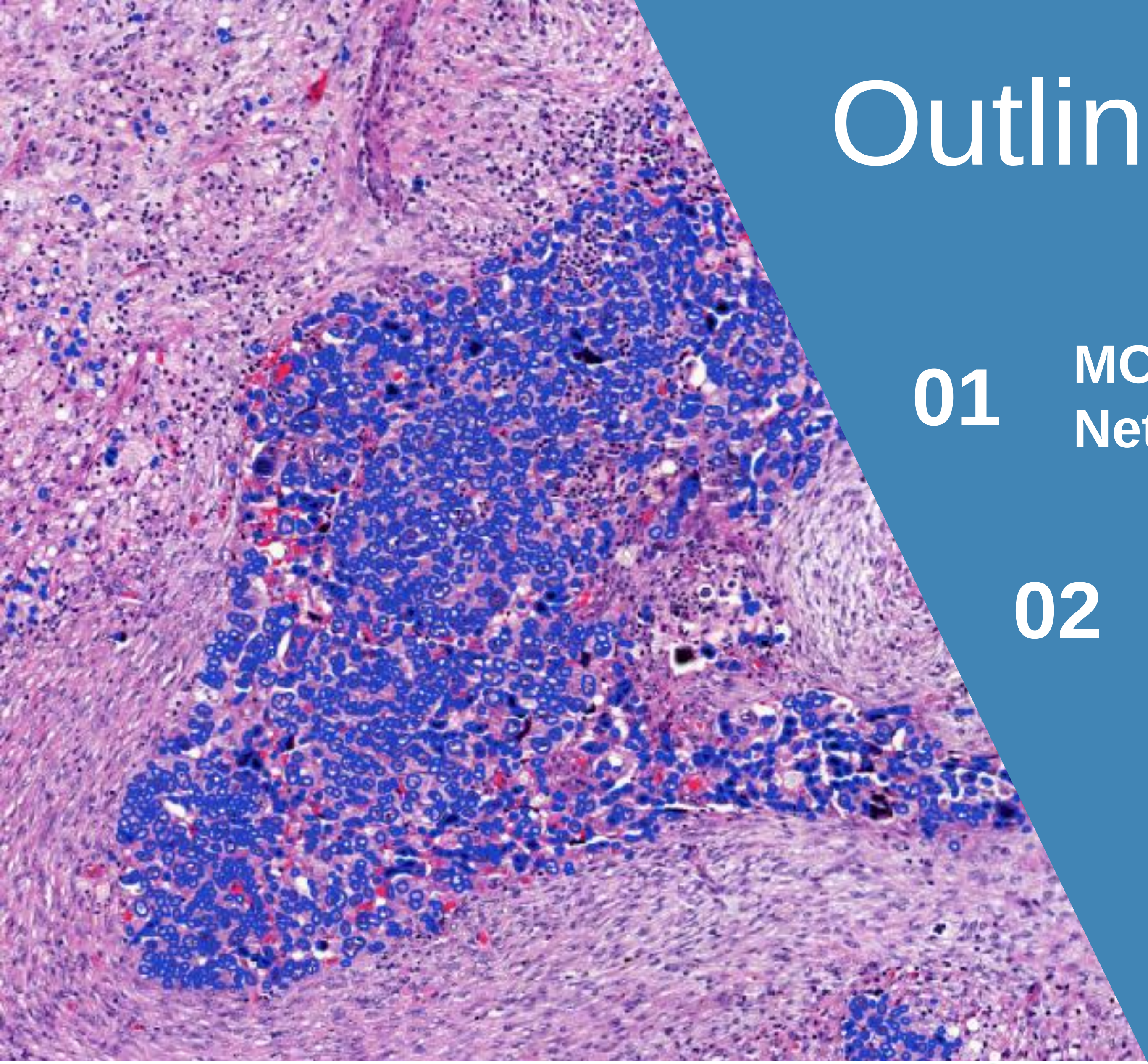
Bayesian NN approximation LAB

Marta Lovino, PhD

2023/2024

marta.lovino@unimore.it

October 5th, 2023



Outline

01 MCDropout as Bayesian Neural Network approximation, a synthesis

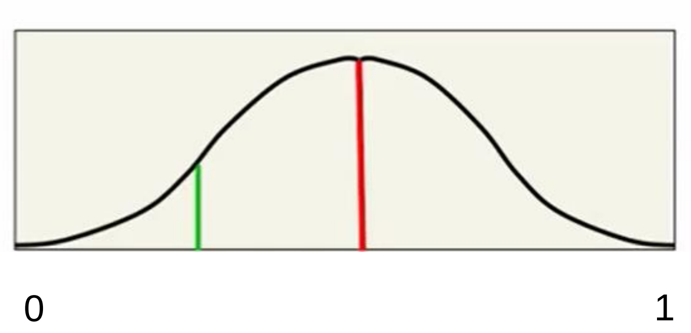
02 Lab assignment

A recap

Is it reasonable to give a single answer?

- If we don't have much data, we are unsure about *p*.
- The computation works better if we take this uncertainty into account

For each parameter *p* we use a probability density function to take uncertainty into account.
In this way, we no longer have a unique parameter but the probability density for that parameter.



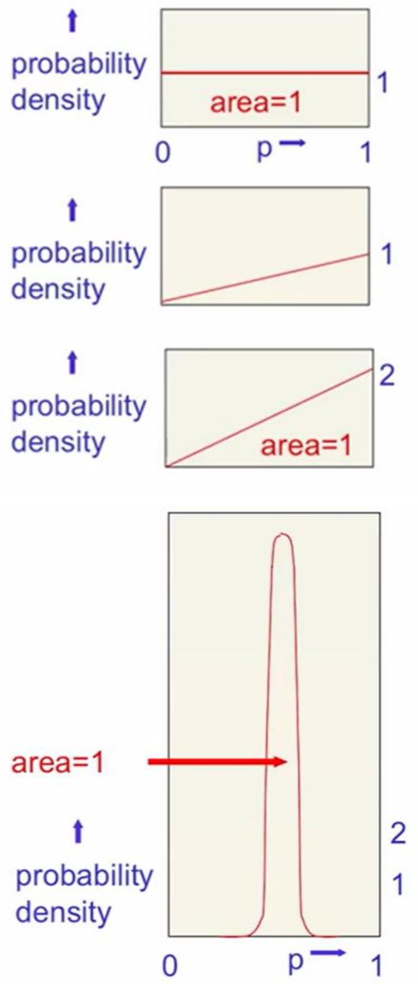
Non Bayesian way: *p = 0.5*

Bayesian way: *p*=0.5 with probability 70%
p=0 with probability 0%
p=1 with probability 0%
p=0.25 with probability 10%

A recap

The parameter *p* can be learnt from data, using Bayes theorem (see the coin example).

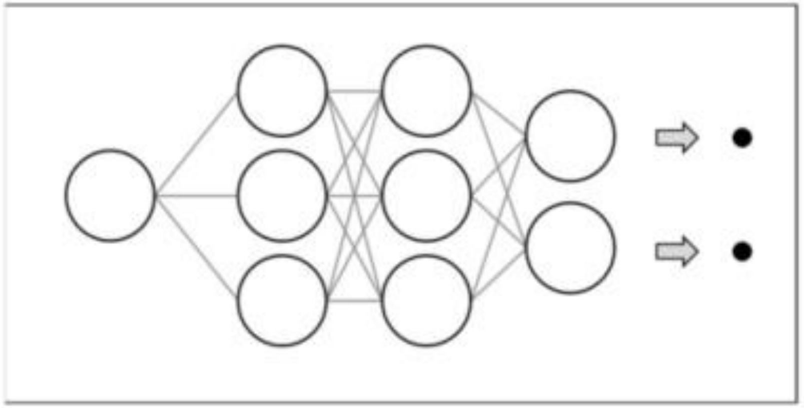
1. **Start a prior** distribution over p (e.g., a uniform distribution)
2. **Multiply the prior probability** of each parameter value **by the probability of observing a head/tail** given that value.
3. Then, **scale up** all the probability densities so that their integral comes to 1. This gives the posterior distribution.
4. Repeat from point 2.



A recap

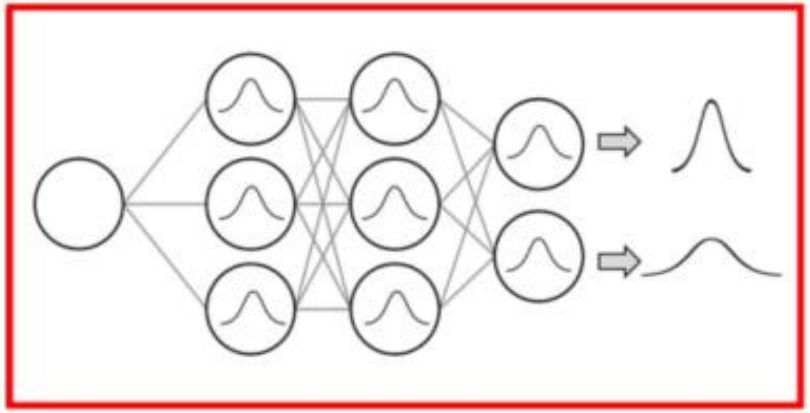
In the Bayesian way, each weight w_{ij} of a neural network is drawn by a probability density function.

- It is a complex problem, but we have a solution to approximate the real Bayesian posterior for our weights.



Dropout!

- Consists in applying dropout before each trainable layer in a deep network, **also at inference time**.
- This has been shown to be equivalent to gaussian distributions for the weights [1]

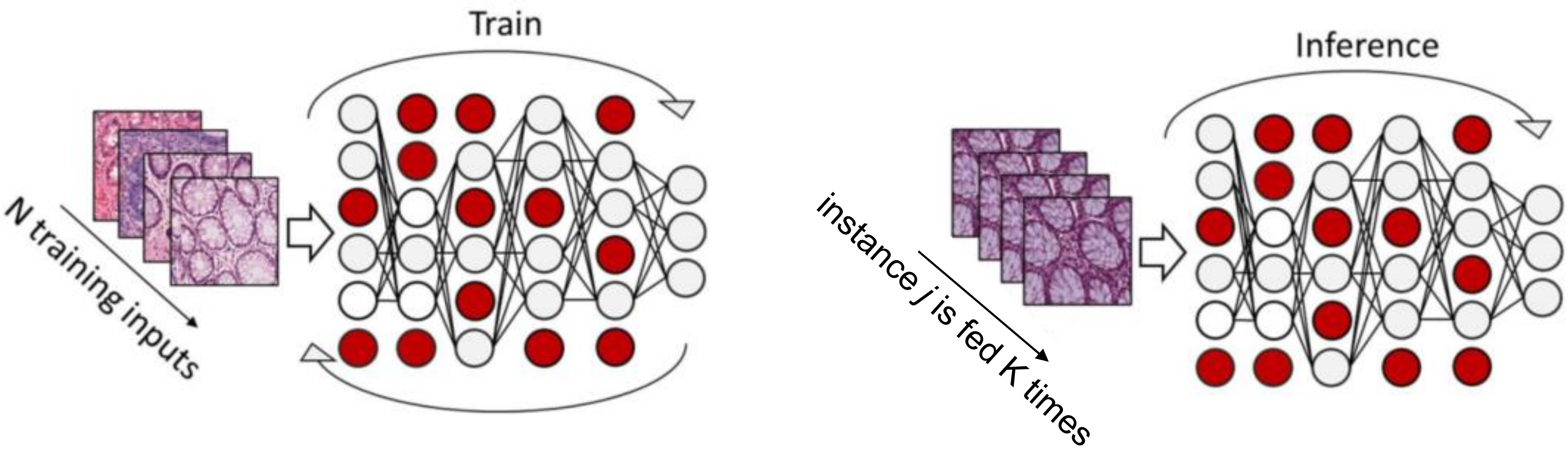


[1] Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning (Gal et al., 2016)

A recap

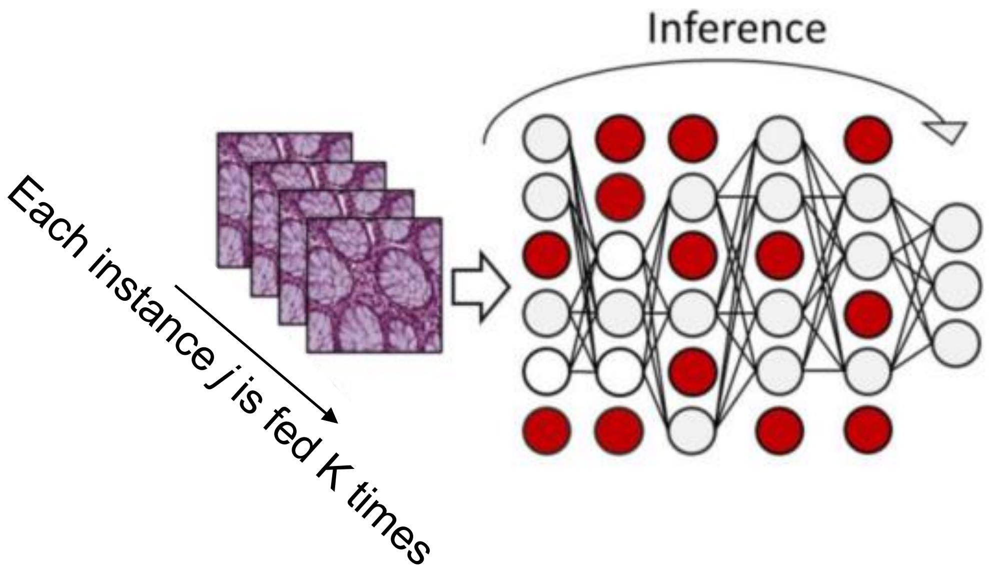
Dropout!

- Consists in applying dropout before each trainable layer in a deep network, also at inference time.
- This has been shown to be equivalent to gaussian distributions for the weights[1].



[1] Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning (Gal et al., 2016)

A recap



At test time:

- **Each instance** is fed K times to the NN

  k=1, $y_1 = 0.25$
  k=2, $y_2 = 0.2$
  k=3, $y_3 = 0.28$
  k=4, $y_4 = 0.20$
  k=5, $y_5 = 0.3$
  …
  k=K-1, $y_{K-1} = 0.24$
  k=K, $y_K = 0.3$

- The probability density function of *y* is obtained from the $y_{\{1..K\}}$ values.

A recap

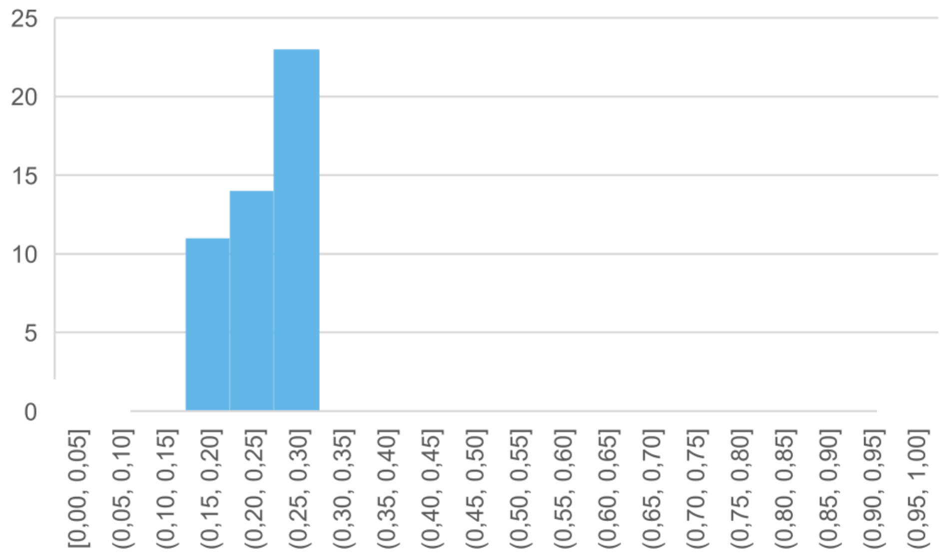
At test time:

- **Each instance** is fed K times to the NN

  k=1, $y_1 = 0.25$
  k=2, $y_2 = 0.2$
  k=3, $y_3 = 0.28$
  k=4, $y_4 = 0.20$
  k=5, $y_5 = 0.3$
  …
  k=K-1, $y_{K-1} = 0.24$
  k=K, $y_K = 0.3$

- The probability density function of *y* is obtained from the $y_{\{1..K\}}$ values.

- **Output *y* for each instance, with its probability value.**



Lab assignment

Download dataset_LUMINAL_A_B.csv file from Teams and implement a Bayesian Breast Cancer classifier using MCDropout Bayesian approximation.

The input consists of gene expression levels of a patient (vector of numbers), the label consists of the patient breast cancer subtype: LUMINAL A or LUMINAL B.

Implement a simple MLP classifier with MCDropout approximation to get for each test patient the class label and the class probability.

Lab assignment

Some cautions:

- Divide dataset_LUMINAL_A_B.csv in train and test sets based on your preference (e.g., 80-20 split).
- Standardize features by removing the mean and scaling to unit variance
- If you want, perform some dimensionality reduction (e.g., with PCA , 80 features).
- Train the MLP classifier on the train set.
- Test the classifier on the test set to get for each test patient the class label and the class probability.
- Test patients are usually classified with a high or low probability?

Useful LINK:
https://xuwd11.github.io/Dropout_Tutorial_in_PyTorch/#51-dropout-as-bayesian-approximation-in-classification-task

Course Folder

You are encouraged to share your scripts on the course folder, to receive comments and feedback from colleagues and instructors.

https://drive.google.com/drive/folders/1ynFYoc3xicaYhSi1X62w9k_JAj2fUrox?usp=sharing

Please upload your solutions with the proper naming:

e.g., LAB1_SURNAMEName

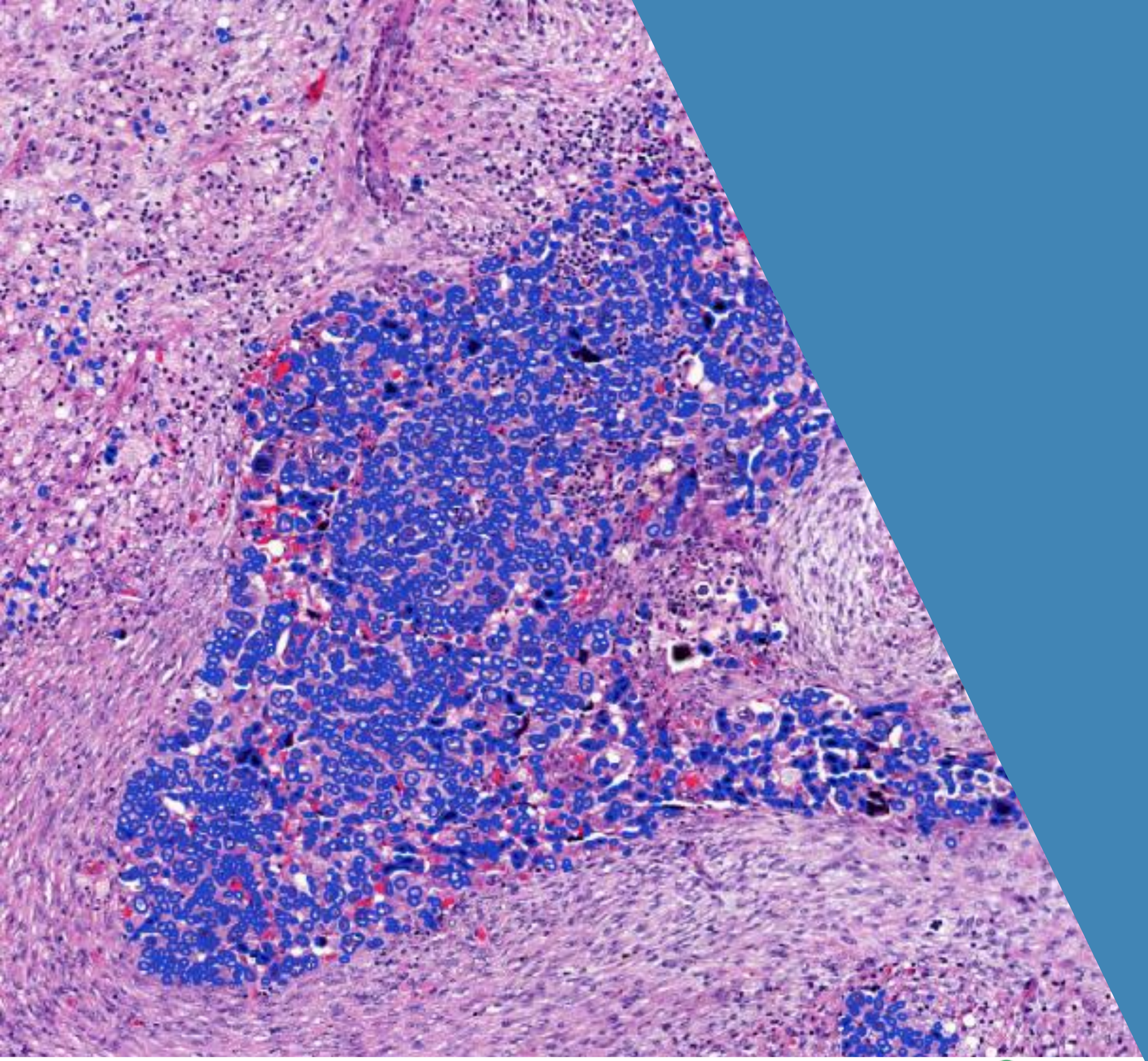
BNN SOLUTION

TRAIN (MLP)
STANDARD.
DROPOUT = 0,5 AFTER OGNI
TRAINABLE LAYER

TEST

- DROPOUT ACTIVE
(togliere model.evaluate.

- Va trovata la distribuzione di y
- alla fine prendo il picco come la predizione vera.



Questions?

Better a stupid question in class than a stupid answer in the exam

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

AImage Lab

DECIDER
Improving clinical decisions in cancer