UNCERTAINTY IN BCNN

Project 9- Bioinformatics

SIMONA MARIA BORRELLO S277789
GIULIO CERRUTO S277335

PROF. ELISA FICARRA
PROF. FRANCESCO PONZIO

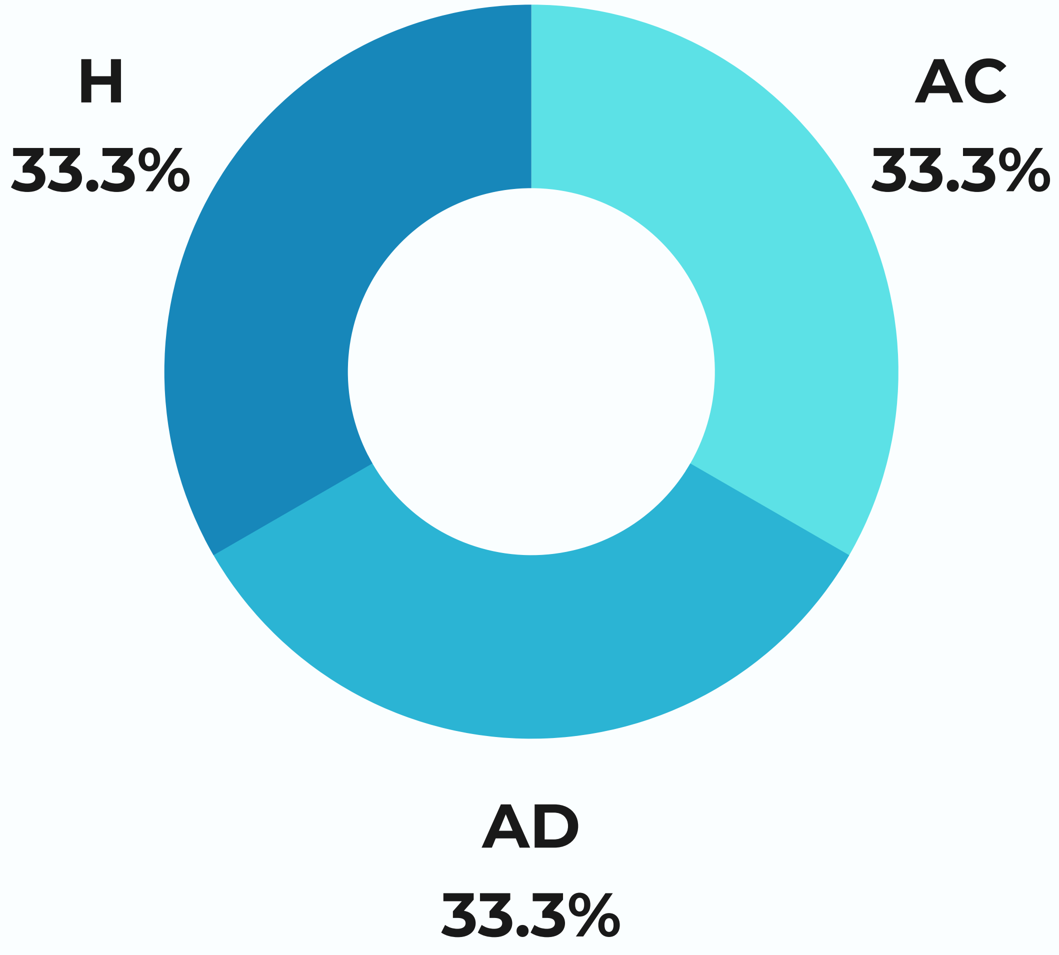
OBJECTIVES

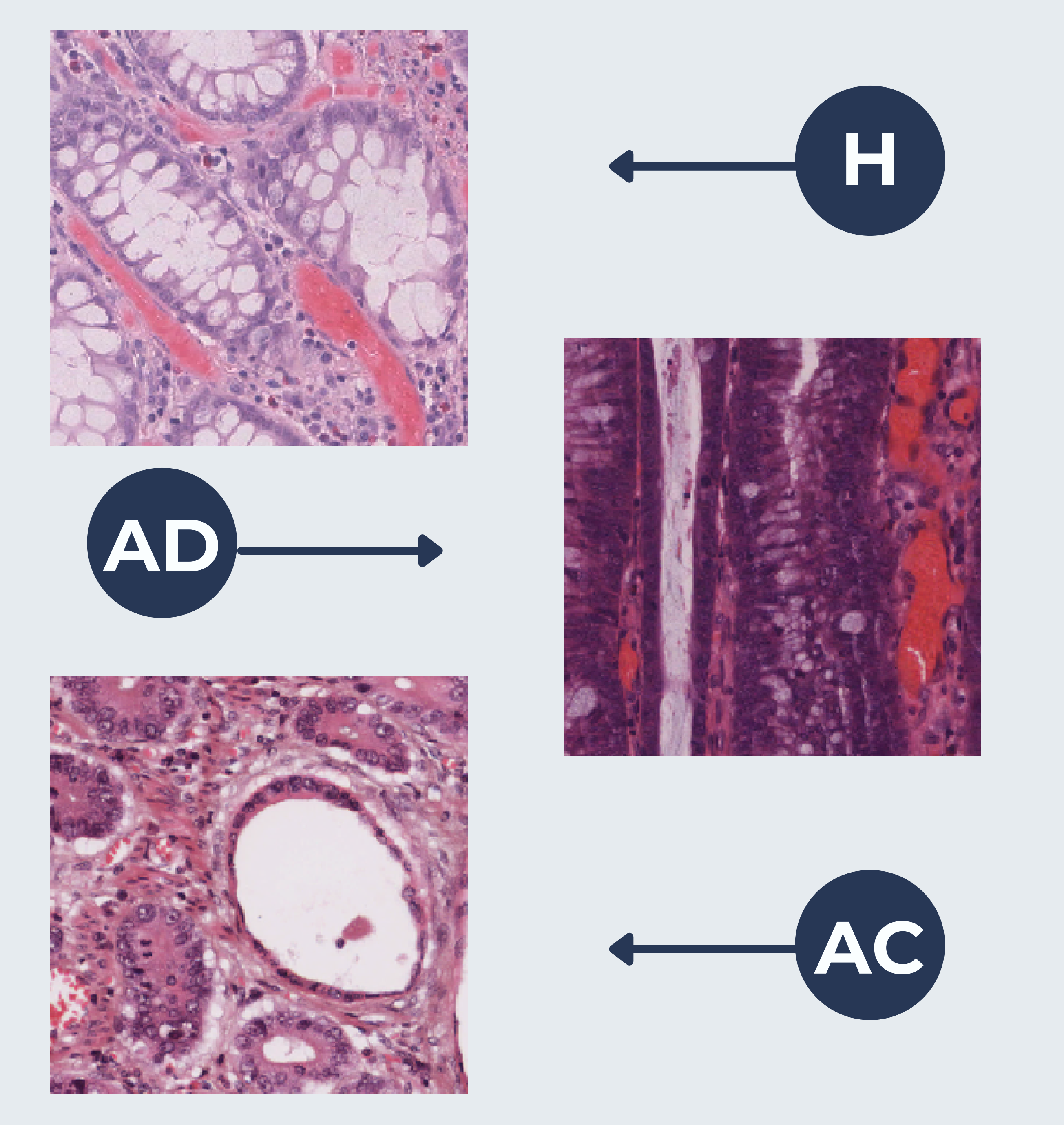
EMBEDDING THE MONTE CARLO DROPOUT UNCERTAINTY INTO THE LEARNING LOSS OF A CONVOLUTIONAL NEURAL NETWORK

- Reduction of weight updates coming from images recognized as spurious
+ Amplification of weight updates coming from clear images .

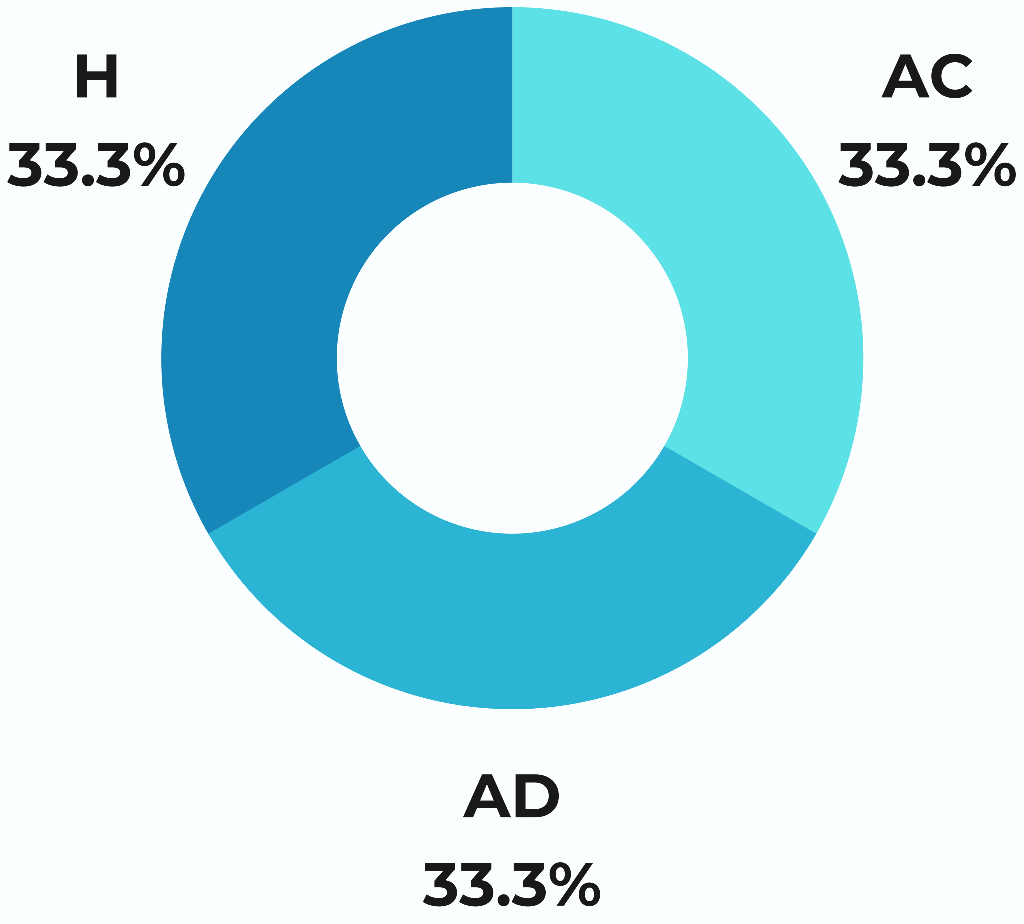
DATASET DESCRIPTION

9000 TRAIN SAMPLES



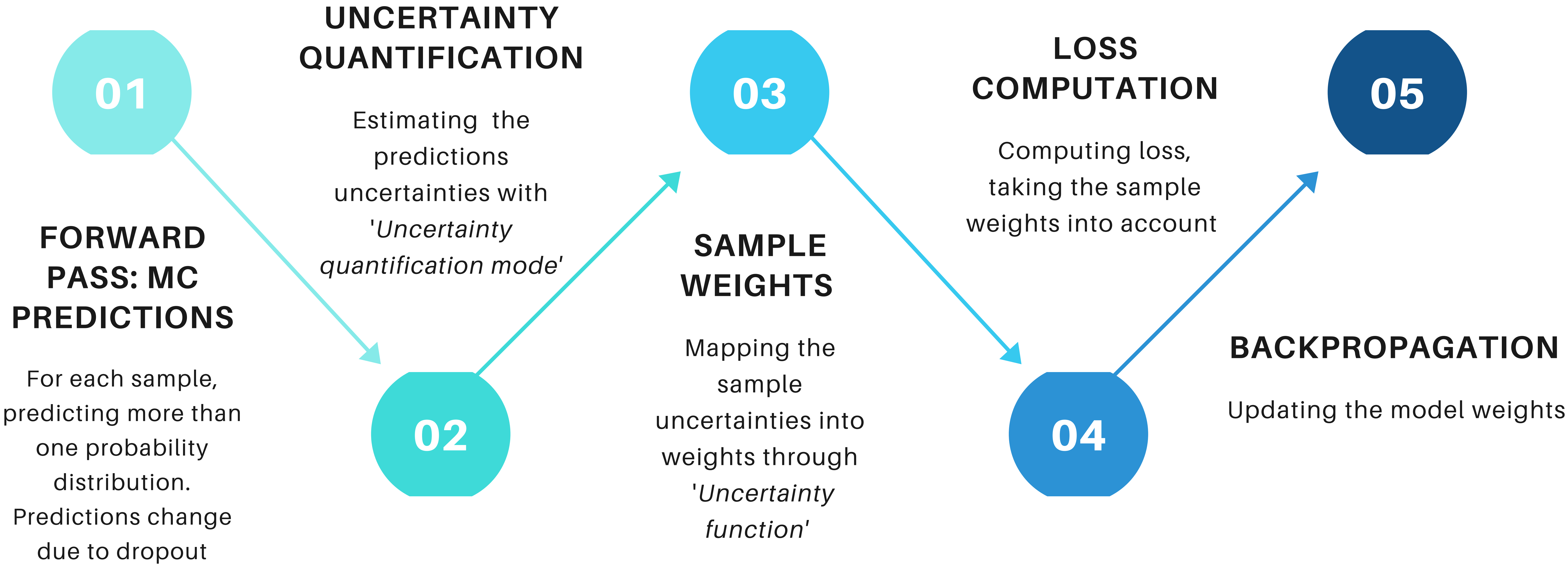


4500 TEST SAMPLES

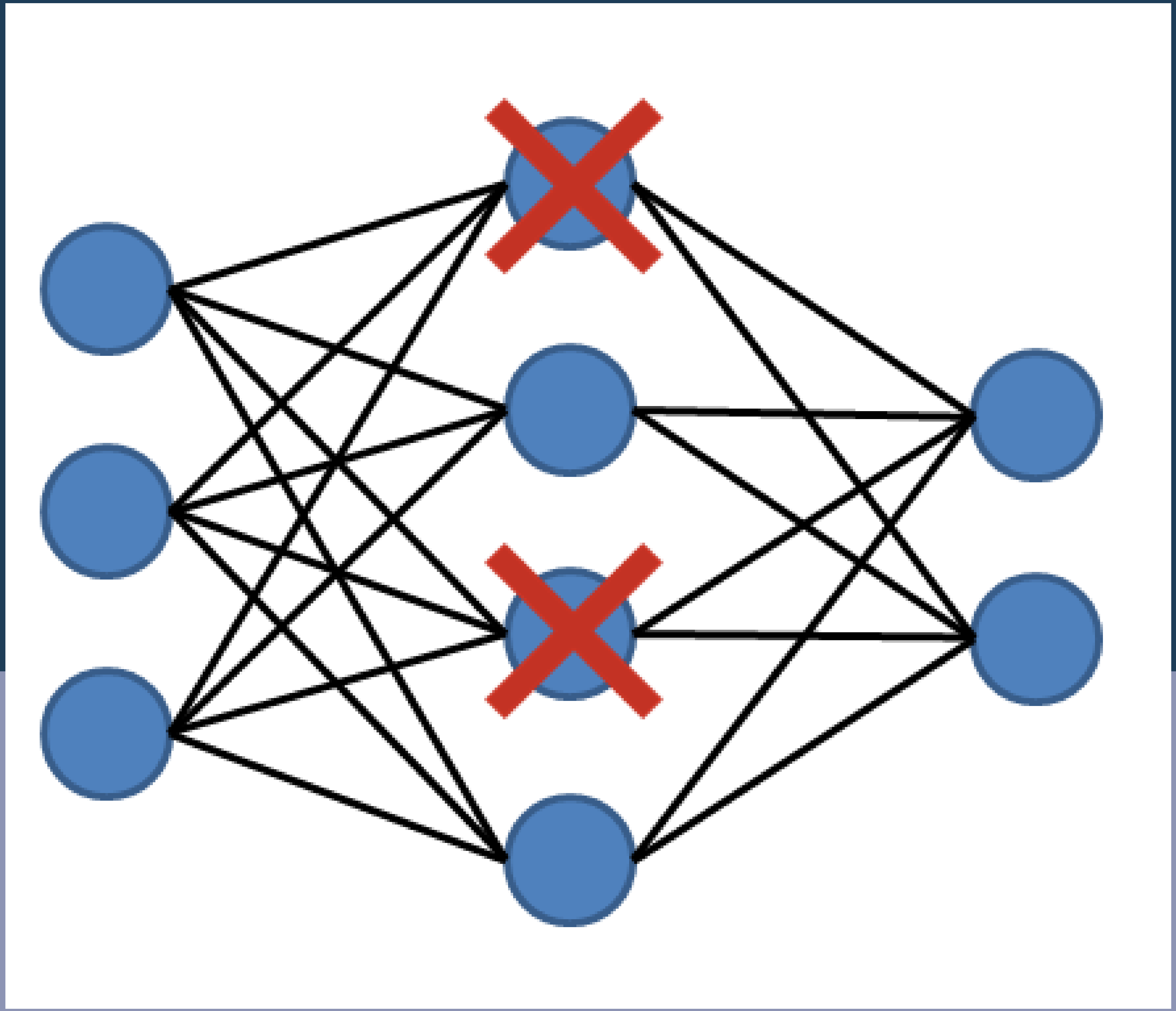


Histological images of Colorectal Cancer belonging to 3 classes:
AC (*adenocarcinoma*), AD (*adenoma*) and H (*healthy*) tissue.

The Stages of Training



DROPOUT LAYERS



DROPOUT SETTING

Where to place dropout layers in the neural network.

Our choices:

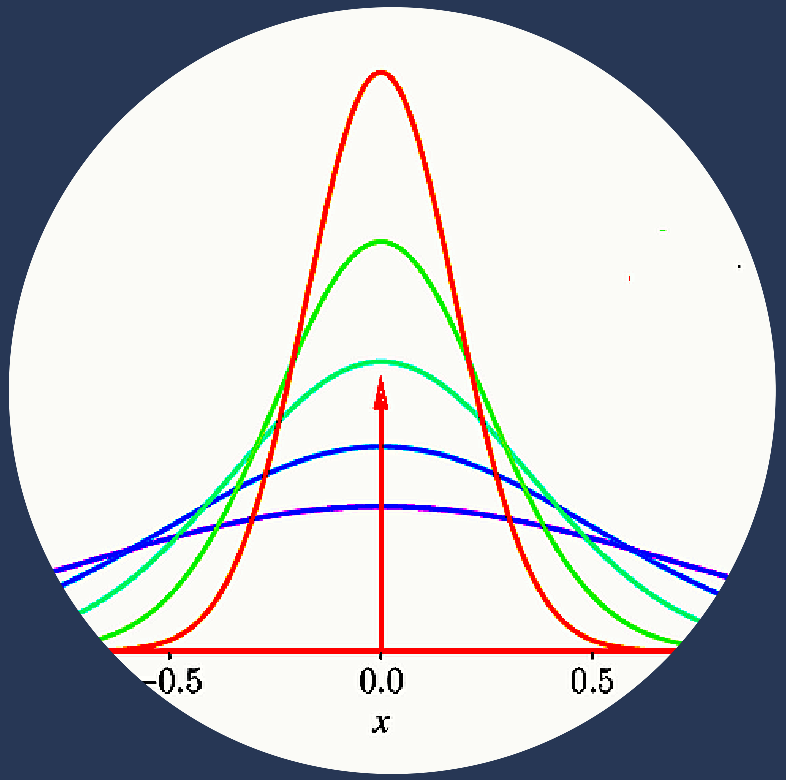
- Classification layers [3]
- Convolutional layers [4]

DROPOUT RATE

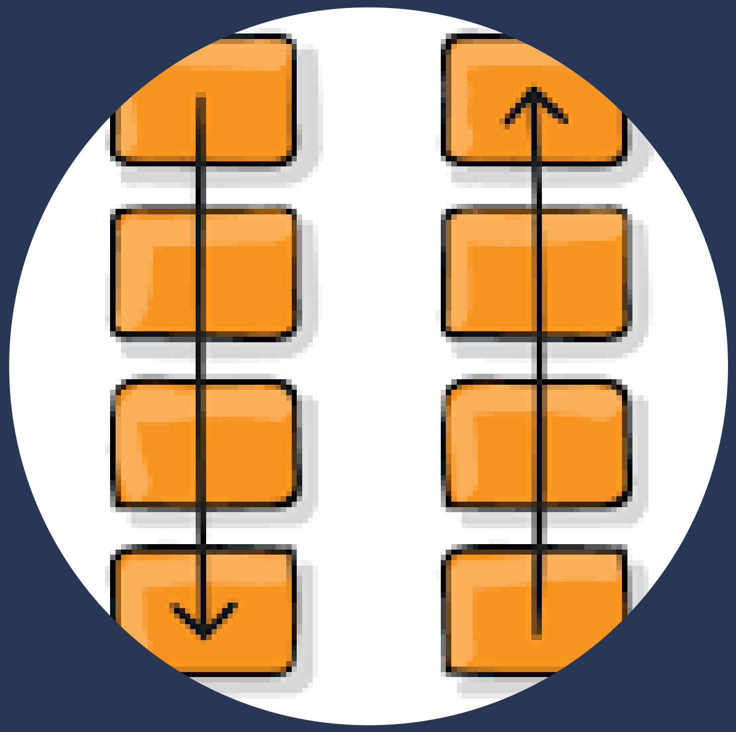
Fraction of the input units to drop.

UNCERTAINTY QUANTIFICATION

how to quantify the uncertainty, after the *mc_replications* times repeated forward pass.

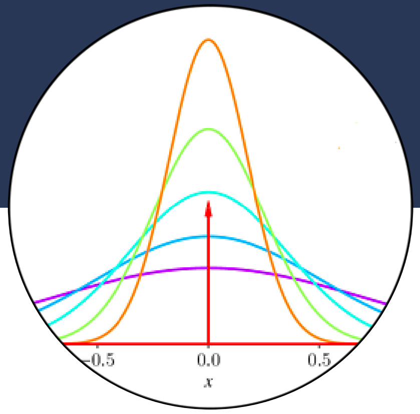


predicted class variances [1]



vertical uncertainties

entropy uncertainties [1]



UNCERTAINTY QUANTIFICATION: PREDICTED CLASS VARIANCES

01

MONTE CARLO SAMPLING

Mean of the predicted probabilities

02

ALEATORIC UNCERTAINTY

Irreducible variability

03

EPISTEMIC UNCERTAINTY

Variability estimation

04

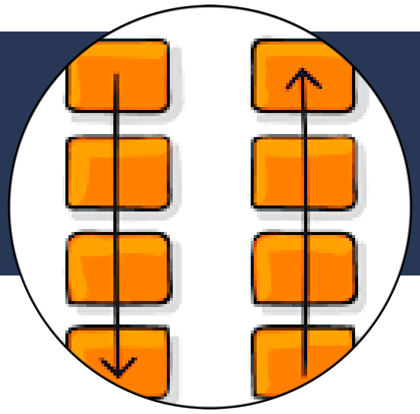
SUM UNCERTAINTIES

Overall uncertainty

05

UNCERTAINTY OF PREDICTED CLASS

Discard the other values



UNCERTAINTY QUANTIFICATION: VERTICAL UNCERTAINTIES

01	MONTE CARLO SAMPLING	Mean of the predicted probabilities
02	MAXIMUM	Maximum between the probabilites
03	DIFFERENCE	Difference between the maximum probability and the others
04	MIN_DIFFERENCE	Minimum between the above calculated differences
05	UNCERTAINTY	Return (1 - minimum)

What is the idea behind it?

Making the samples less uncert
if the maximum probability
is much bigger than the others.

NUMERICAL EXAMPLE:

P=[0.6 0.1 0.3]

Maximum=0.6

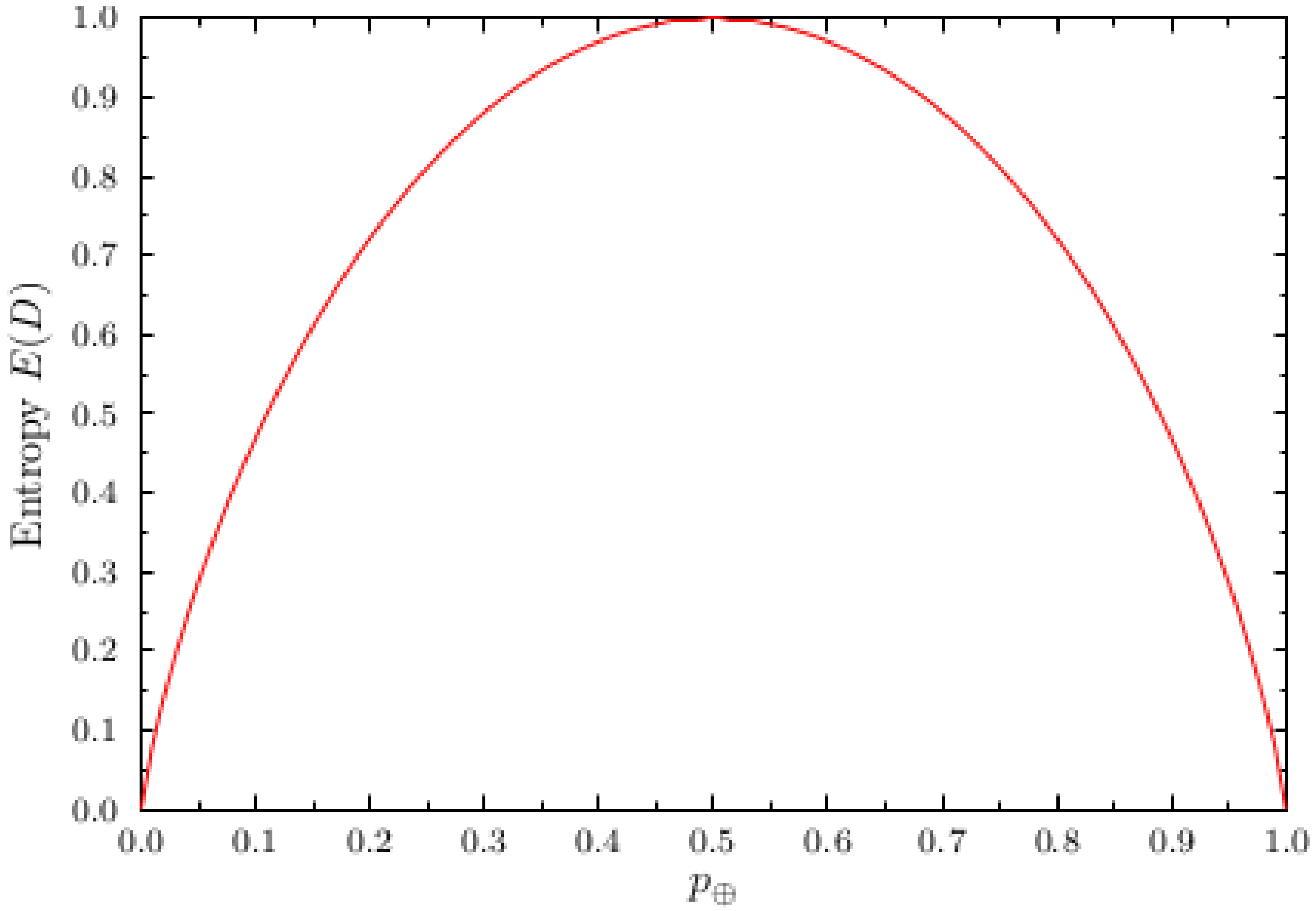
Difference= [0.5 0.3]

Min_difference= 0.3

Uncertainty= 0.7

UNCERTAINTY QUANTIFICATION: ENTROPY UNCERTAINTIES

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$$



NUMERICAL EXAMPLE:

P=[0.6 0.1 0.3]

Entropy= 1.29

P=[0.33 0.33 0.33]

MAX ENTROPY = 1.58

P=[0 0 1]

MIN ENTROPY = 0

UNCERTAINTY FUNCTION

how to map the uncertainty value of a sample to the weight it will have in the loss minimization step.

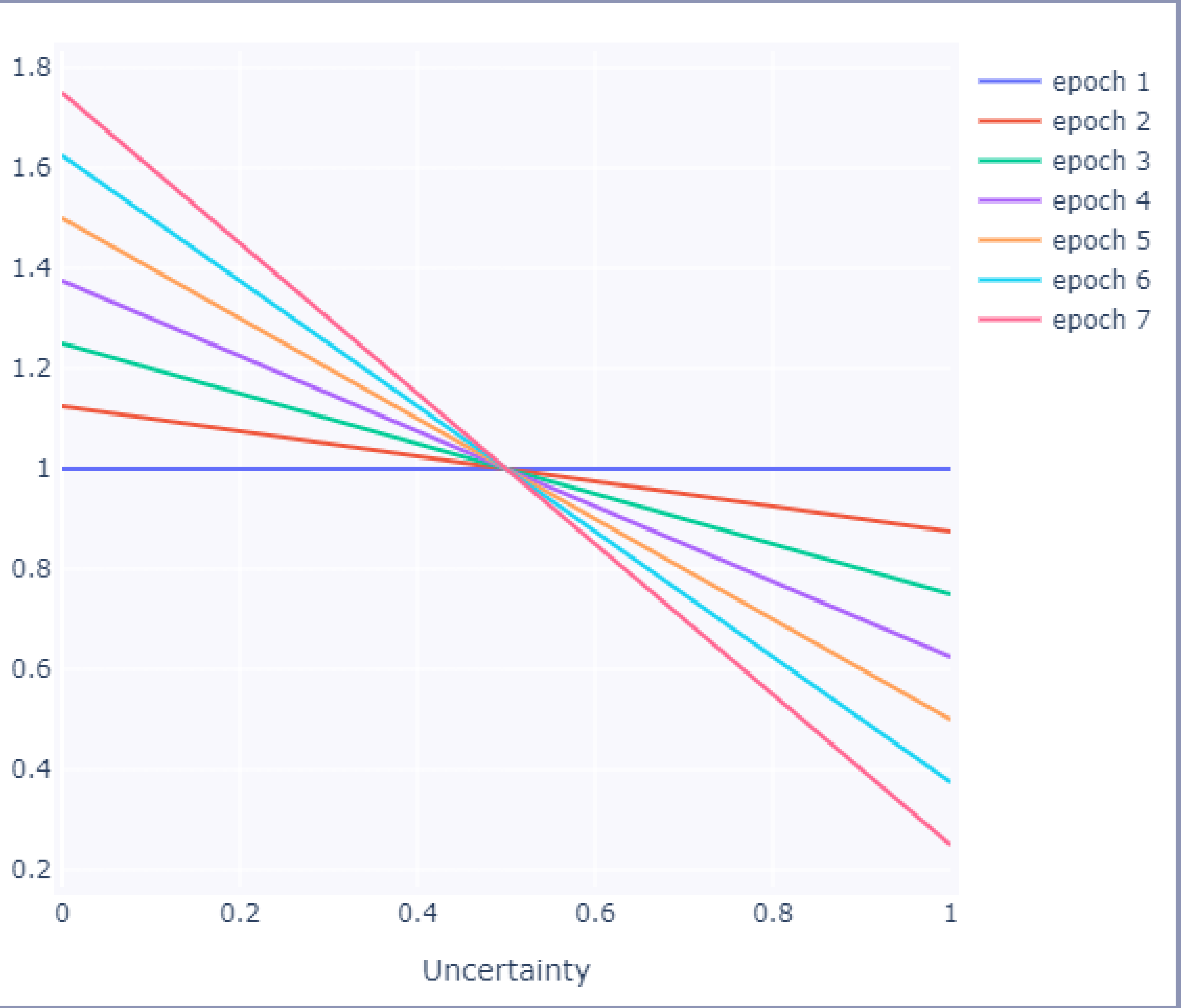
Normalization: **The uncertainty belongs to the interval [0,1].**

The function can take

1 argument : the uncertainty of the sample

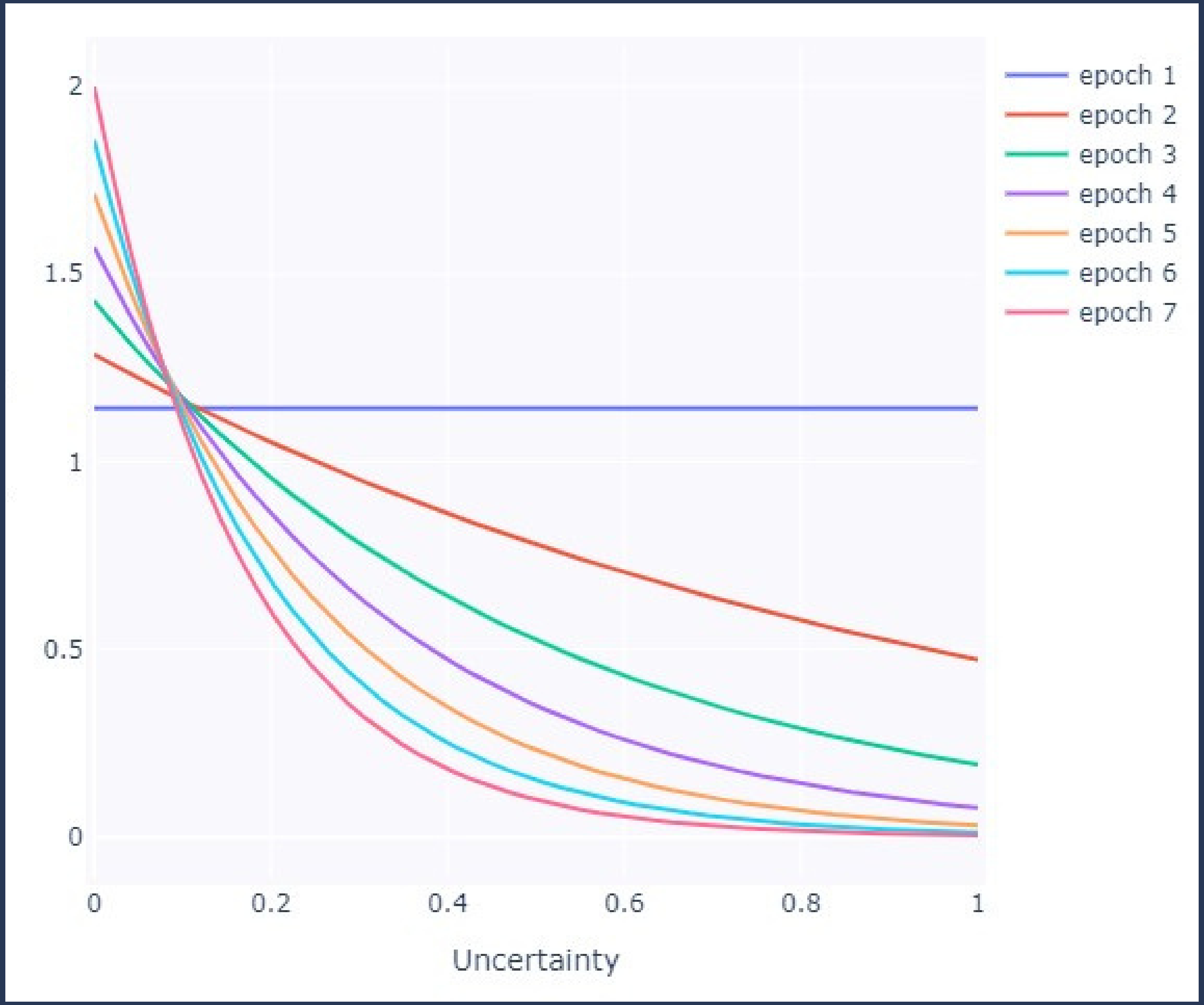
2 arguments: the epoch number and the uncertainty of the sample

UNCERTAINTY FUNCTION



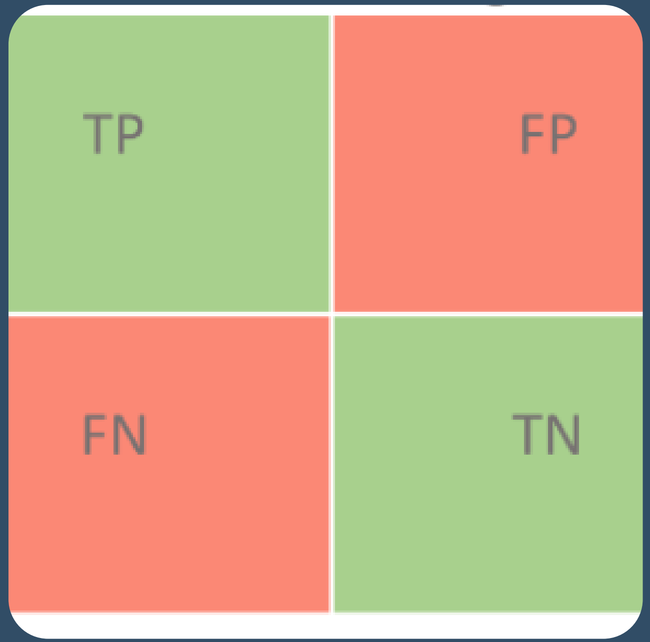
LINEAR

EXPONENTIAL



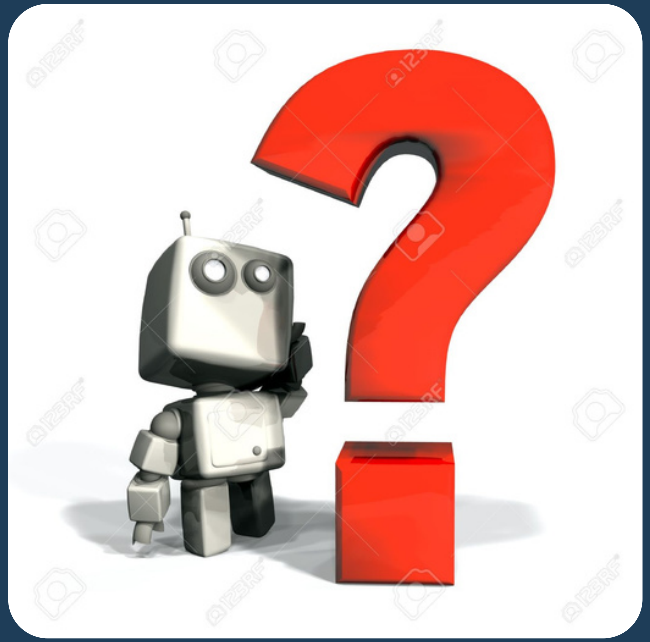
MODEL EVALUATION

ACCURACY



- All classes are thought to be equally important. Nonetheless, in a real-world context, the AC class is definitely more important to be detected.
- The dataset is perfectly balanced, so this choice doesn't bring about a big bias in the results.

NO UNCERTAINTY ACCURACY



Accuracy calculated on the subset of samples whose value of uncertainty is below a threshold (i.e. the model is fairly certain on them)

Uncertain samples → Analysis by the doctor

Fixed Hyperparameters

BATCH SIZE

Fixed to 20

LEARNING RATE

Fixed to 1e-4

OPTIMIZER

Adam optimizer

LOSS

Sparse categorical cross entropy

Tuned Hyperparameters

UNDERLYING MODEL

Deterministic model starting point for our dropout model

EPOCH NUMBER

UNCERTAINTY FUNCTION

UNCERTAINTY QUANTIFICATION

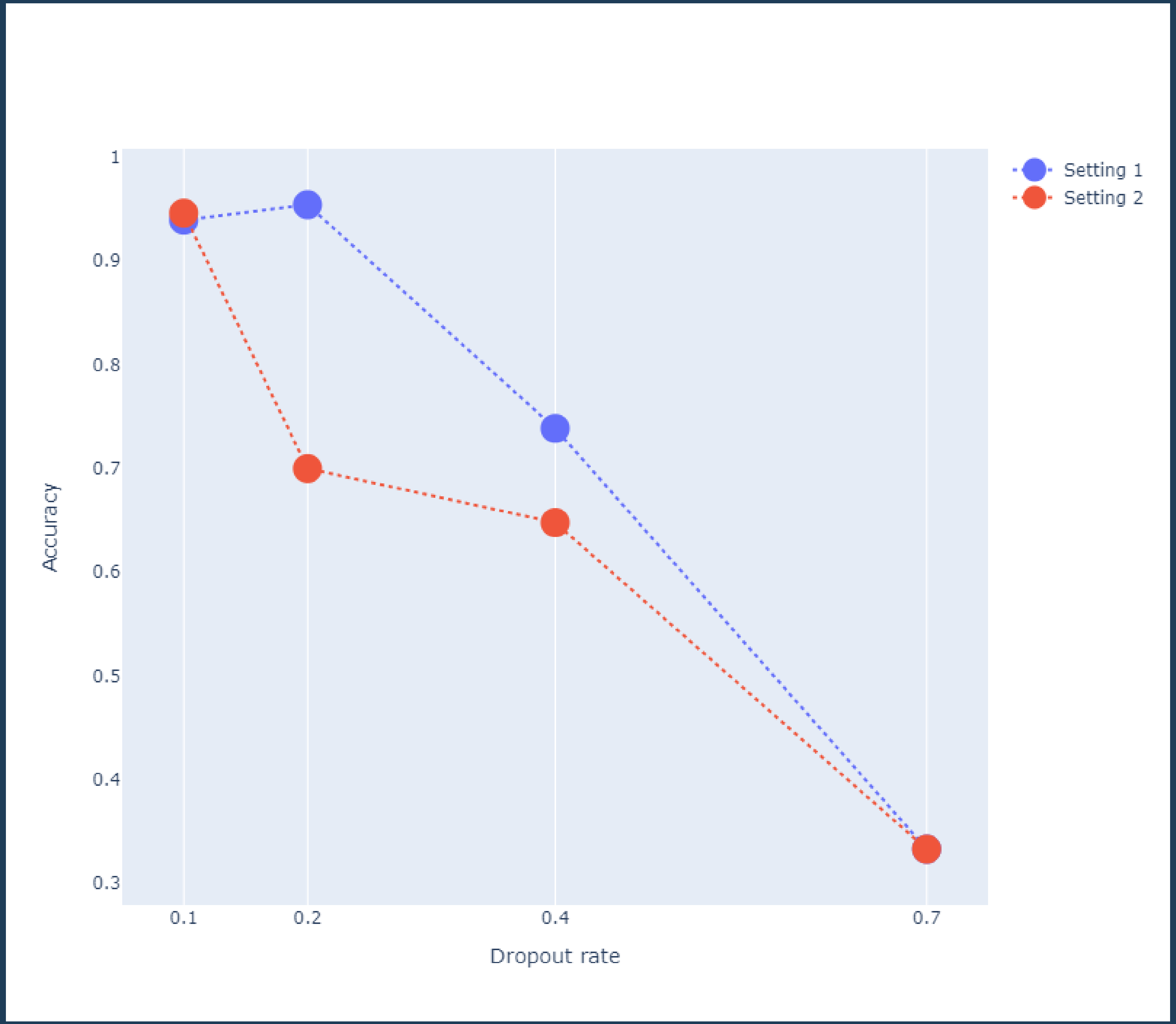
MC REPLICATIONS

Number of times of each forward pass

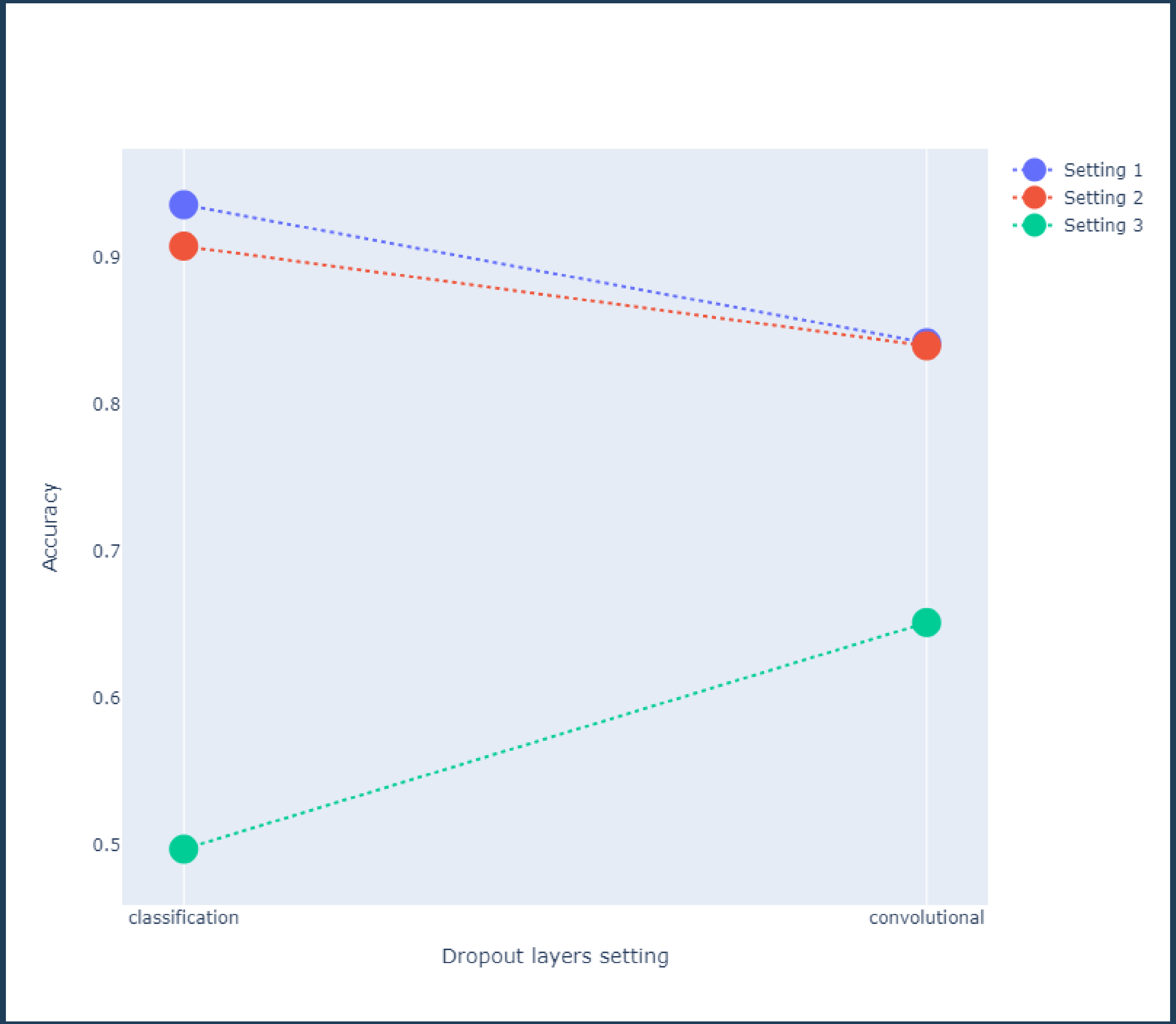
DROPOUT RATE

DROPOUT LAYERS SETTING

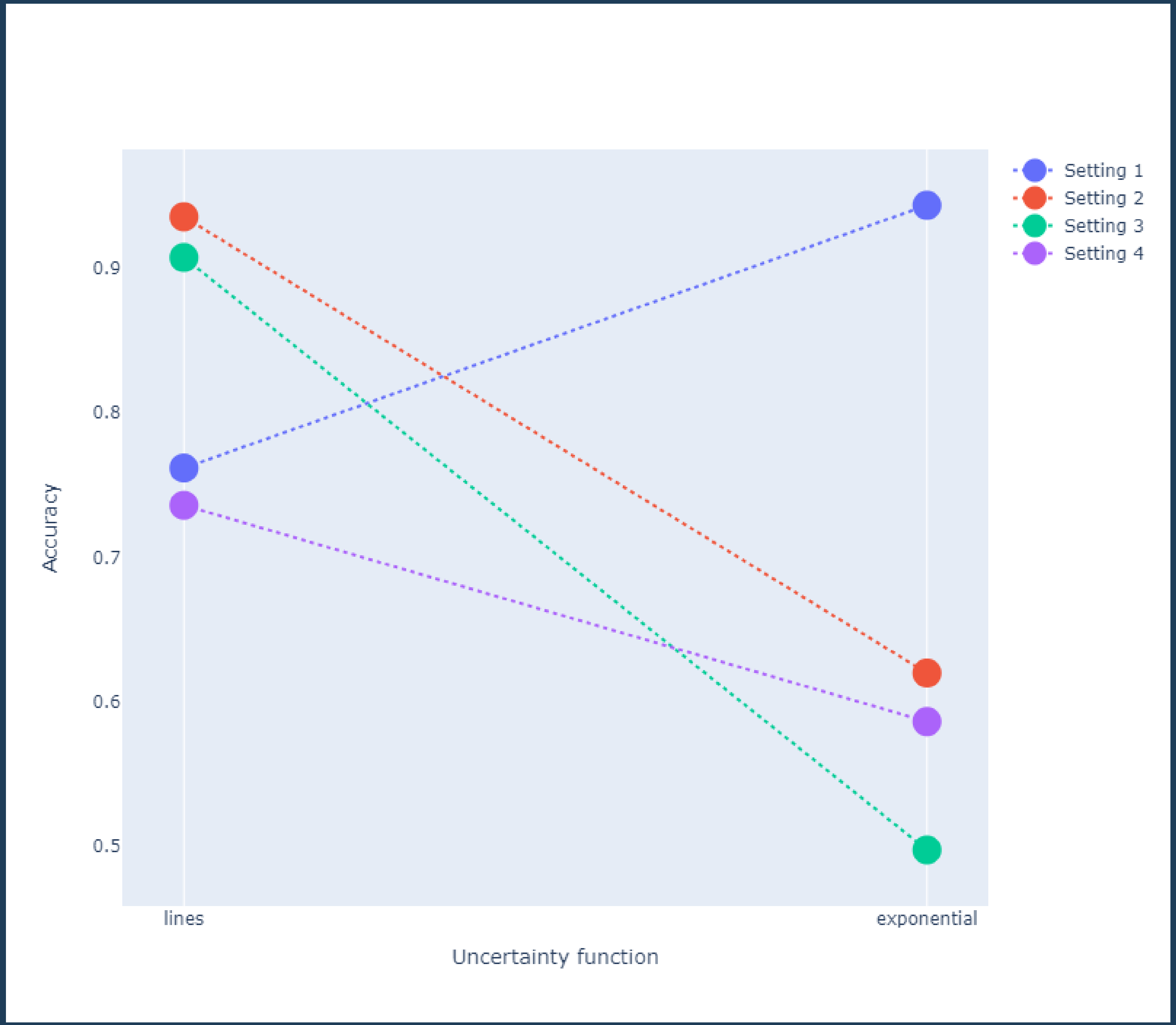
DROPOUT RATE



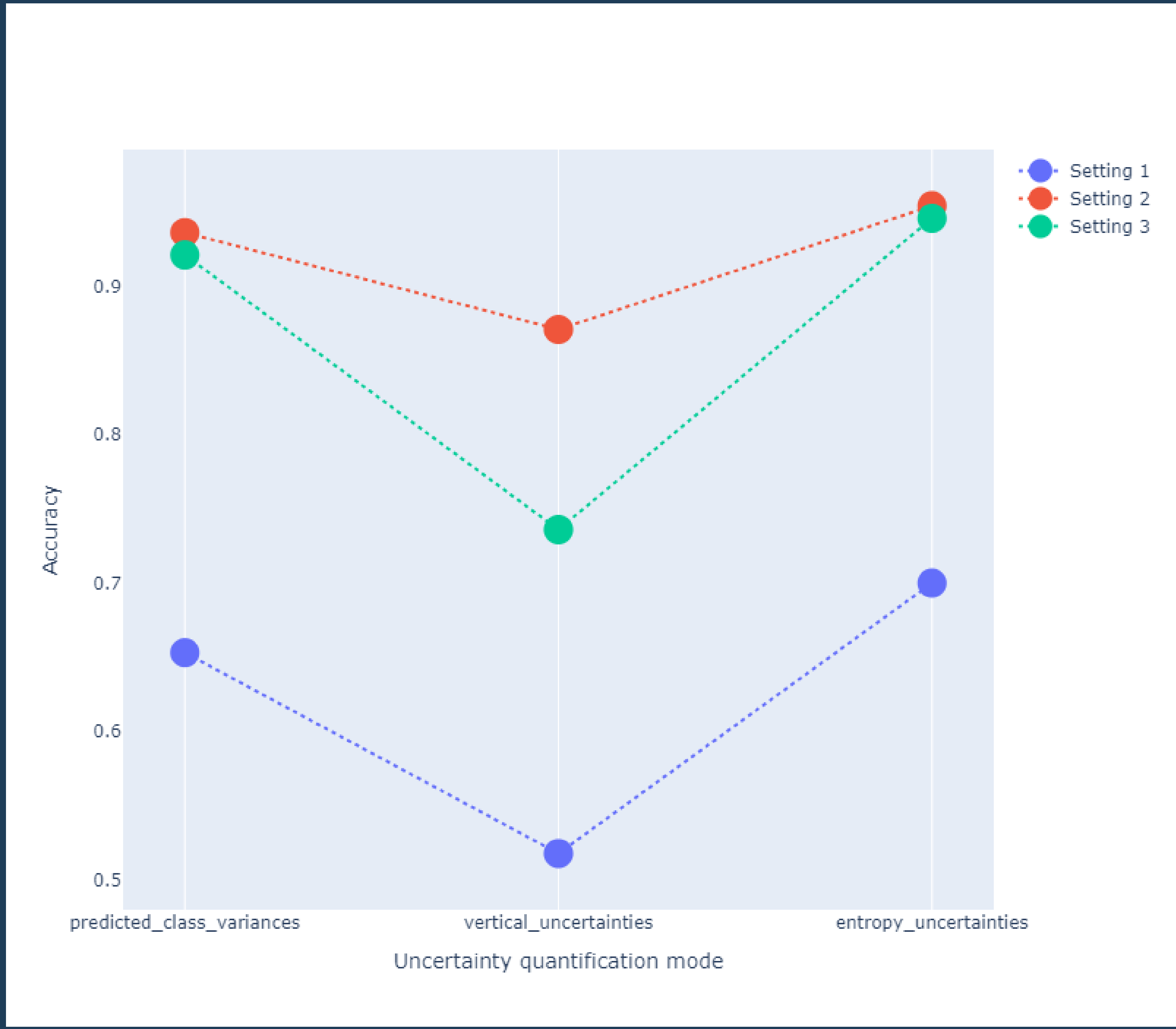
DROPOUT LAYERS SETTING



UNCERTAINTY FUNCTION



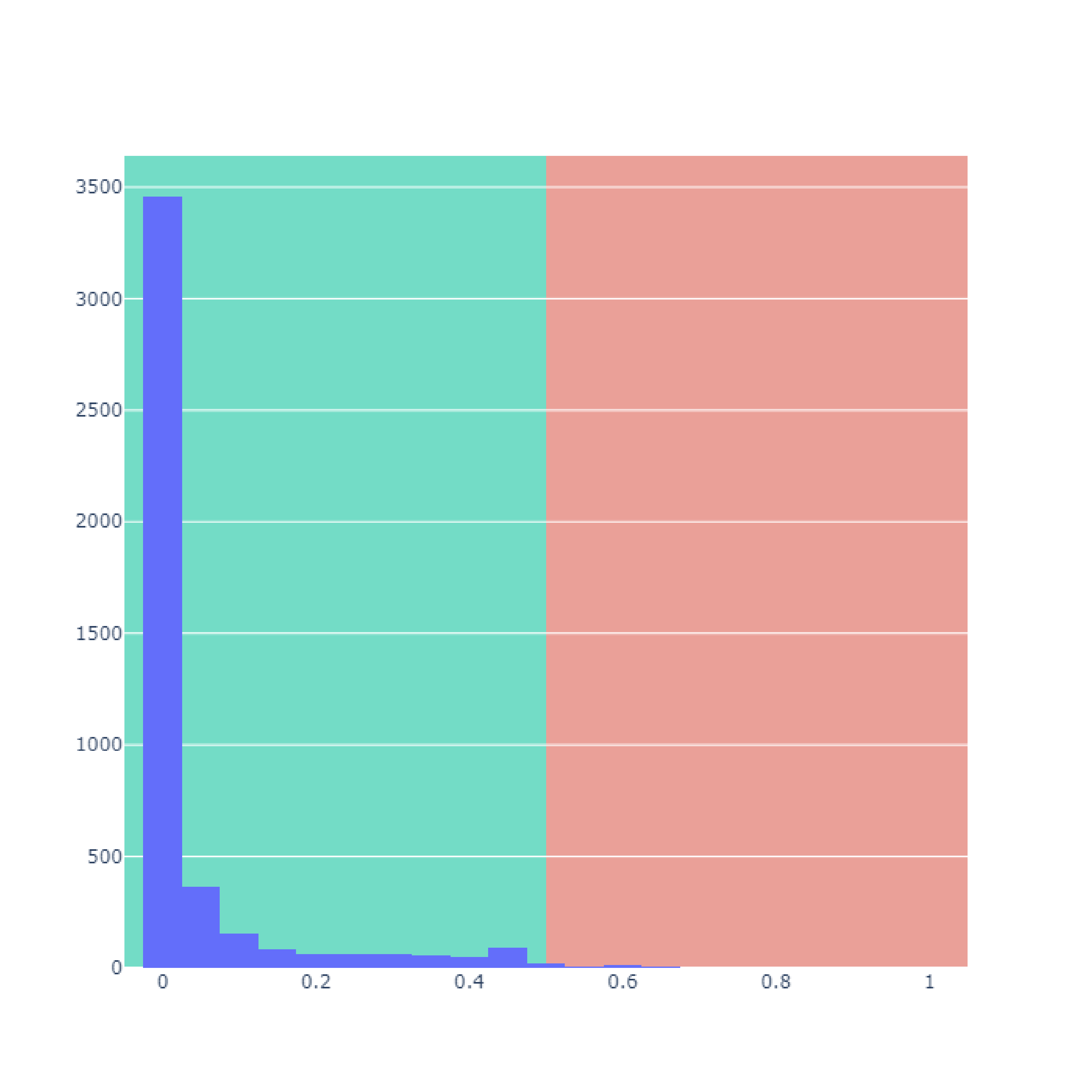
UNCERTAINTY QUANTIFICATION



BEST CONFIGURATION

- UNDERLYING MODEL= **VGG19**
- EPOCH NUMBER = **10**
- UNCERTAINTY FUNCTION = '**LINEAR**'
- UNCERTAINTY QUANTIFICATION = '**ENTROPY_UNCERTAINTIES**'
- MC REPLICATIONS=**6**
- DROPOUT RATE=**0.2**
- DROPOUT LAYERS SETTING= '**CONVOLUTIONAL**'

Accuracy= 96.09 %



No_uncert_Accuracy= 96.46 %

Disregarding samples with uncertainty > 0.5

Ignoring 0.87% of test samples

Little accuracy improvement

Bibliography

[1] Yongchan Kwona, Joong-Ho Wona, Beom Joon Kimb, Myunghee Cho Paik. **Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation.** 2019.

[2] Yarin Gal, Zoubin Ghahramani. **Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.** 2016.

[3] G. E. Hinton , N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. 2012.

[4] Sungheon Park and Nojun Kwak. **Analysis on the Dropout Effect in Convolutional Neural Networks.** 2016.