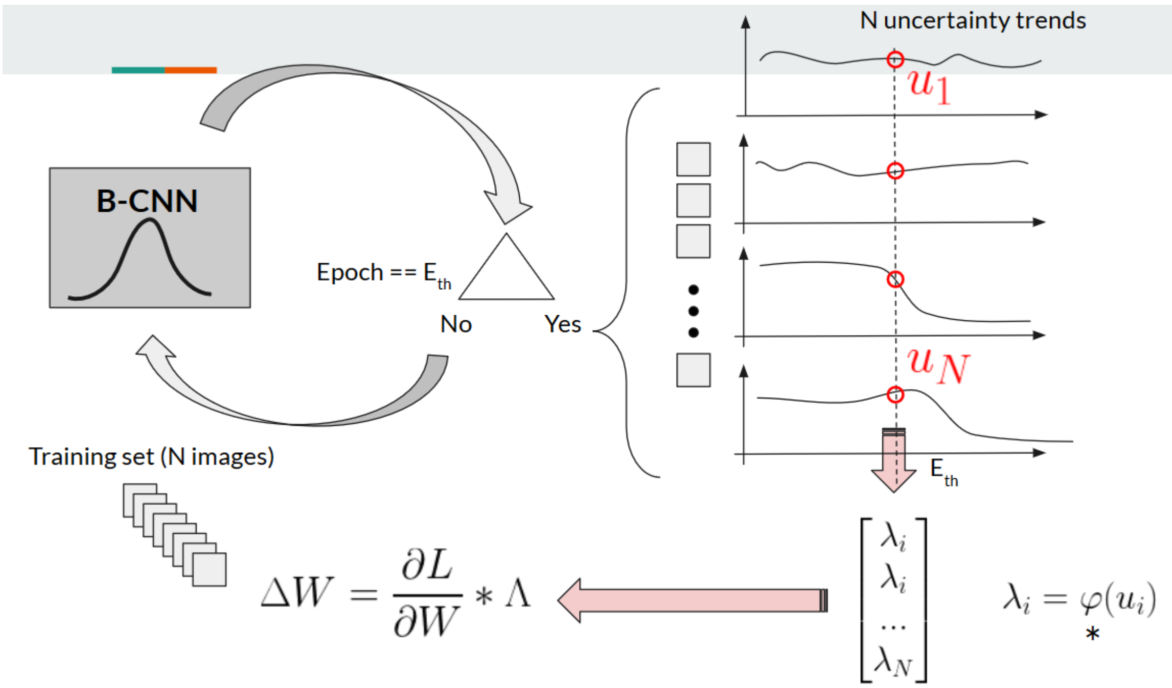
Project #7: Uncertainty-weighted loss for B-CNNs

Introduction

Convolutional Neural Networks (CNNs) are the state-of-the-art methodologies for many computer vision tasks, ranging from image segmentation to image classification. Despite their success, CNNs are not able to provide a reliable measure of uncertainty, able to state if the net is confident or not regarding a given prediction. To fill this technological gap, Bayesian CNNs (BNNs) have been developed and investigated. A common way to implement a BCNN leverages the so-called Montecarlo Dropout [1] (MC dropout). Using such method, implementing a BCNN is as simple as keeping dropout layers active also in inference phase. Nevertheless, the uncertainty computation is nowadays a phase which is done downstream of the training. An interest research focus may be tailoring the training of the network upon the uncertainty related information, using such knowledge to modify, for example, the loss function.

Scope

<u>The aim of the project is the embedding of the uncertainty related information, exploited by a BCNN leveraging MC dropout, into the loss metric of the network itself.</u>



For instance, in the above figure, the uncertainty information, retrieved during training, is multiplied to the loss of the model. How to retrieve uncertainty information? As discussed in Machine Learning and Deep Learning LAB4, by iterating the prediction on a given image, let us say K times, it is possible to get an approximation of the predictive distribution, sampled K times indeed. Using the uncertainty implementation of [2] over such predictive distribution, we can get a measure of the confidence of the net regarding the given input. Implementing the uncertainty computation during the training phase, at a cost of a longer training time, we can get values that in some way (multiplied, summed, subtracted) may be integrated into the model's loss.

Dataset

The application is Colorectal cancer classification from histological images. Such images are digitalization of tissue samples, studied by pathologists looking for cancer.

The dataset is made up of three classes of interest:

1. Healthy tissue (H)
2. Cancer (AC)
3. Adenoma (AD), a lesion that may turn into cancer.

And four classes of spurious images, associated with a fake class from AC, AD, H:

a) Blood (BLOOD)
b) Fat (FAT)
c) Glass (GLASS)
d) Stroma (STROMA)

The images are in form of numpy array:

- *X{train, test}.npy*
  shape ({12336, 7308}, 32, 32, 3)
  training images
- *patients{train, test}.npy*
  shape ({12336, 7308},)
  patients corresponding to {train, test} images
- *real_classes{train, test}.npy*
  shape ({12336, 7308},)
  real class corresponding to {train, test} images
- *Y{train, test}.npy*
  shape ({12336, 7308},)
  labels corresponding to {train, test} images. In case of spurious image this is a "fake" class belonging to [AC, AD, H]

References

[1] Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." international conference on machine learning. 2016.

[2] Kwon, Yongchan, et al. "Uncertainty quantification using bayesian neural networks in classification: Application to ischemic stroke lesion segmentation." (2018).ù