ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

APAI Lab01: DNN Definition and Training

Davide Nadalini, Alberto Dequino, Luca Bompani, Lorenzo Lamberti, Francesco Conti.
(University of Bologna)

d.nadalini@unibo.it

In this Hands-on session:

A first-time user of Pytorch framework will learn how to :

- define a Neural Network in PyTorch;
- train a NN;
- test a NN.

Tasks:

1. PyTorch definition of a NN model;
2. Count network's parameters and MAC operations;
3. Data loader for Fashion-MNIST
4. Code for testing a neural network on Fashion MNIST dataset;
5. Code for training a neural network on Fashion MNIST;
6. Save and load model's trained weights;

All the details about the tasks are explained in the pdf document attached.

RULES of the labs

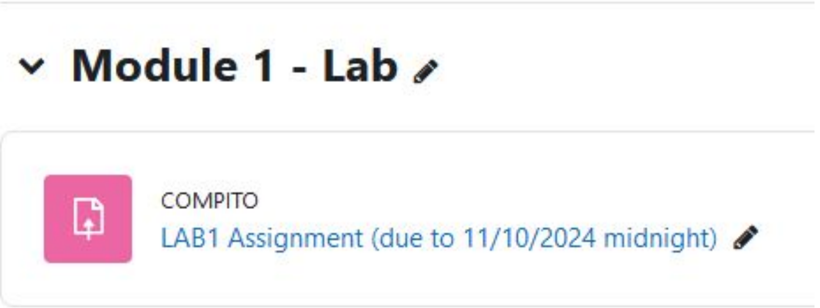
- 4 LABS in total (first module)
- +1 point at the exam (only Module I - Prof. Garofalo part) for each lab assignment delivered and correct (up to +4 points).
- I week of deadline for the solution (2 for today's lab only)
- 2-people group work is encouraged (1 solution for the group, make sure of writing your names in the solution!!)

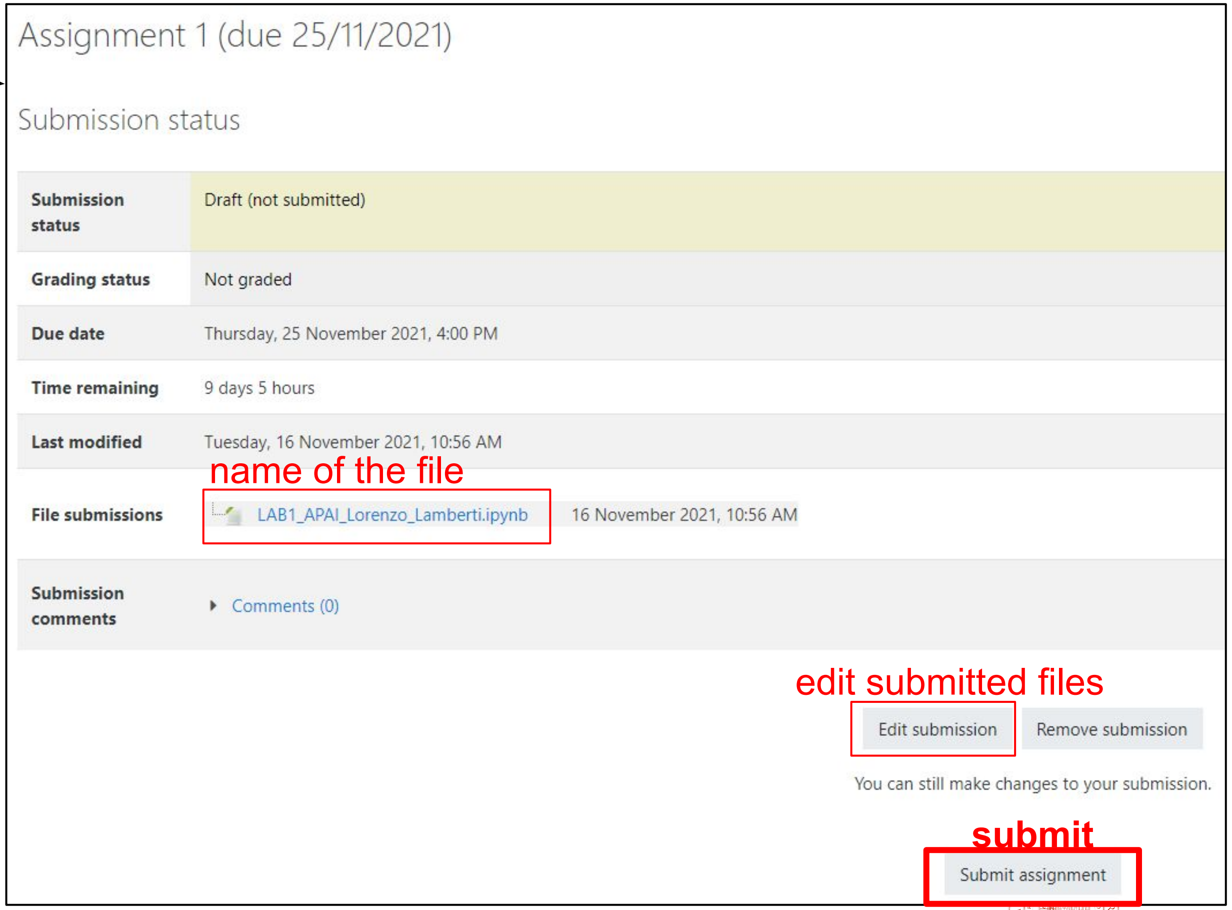
How to deliver the assignment

- Use Virtuale platform to load your file
- update only the .ipynb file, **<u>named as follows</u>**:

  LAB1_APAI_yourname.ipynb

Important: the **notebook must be pre-run by you**. Outputs must be correct and visible when you download it.



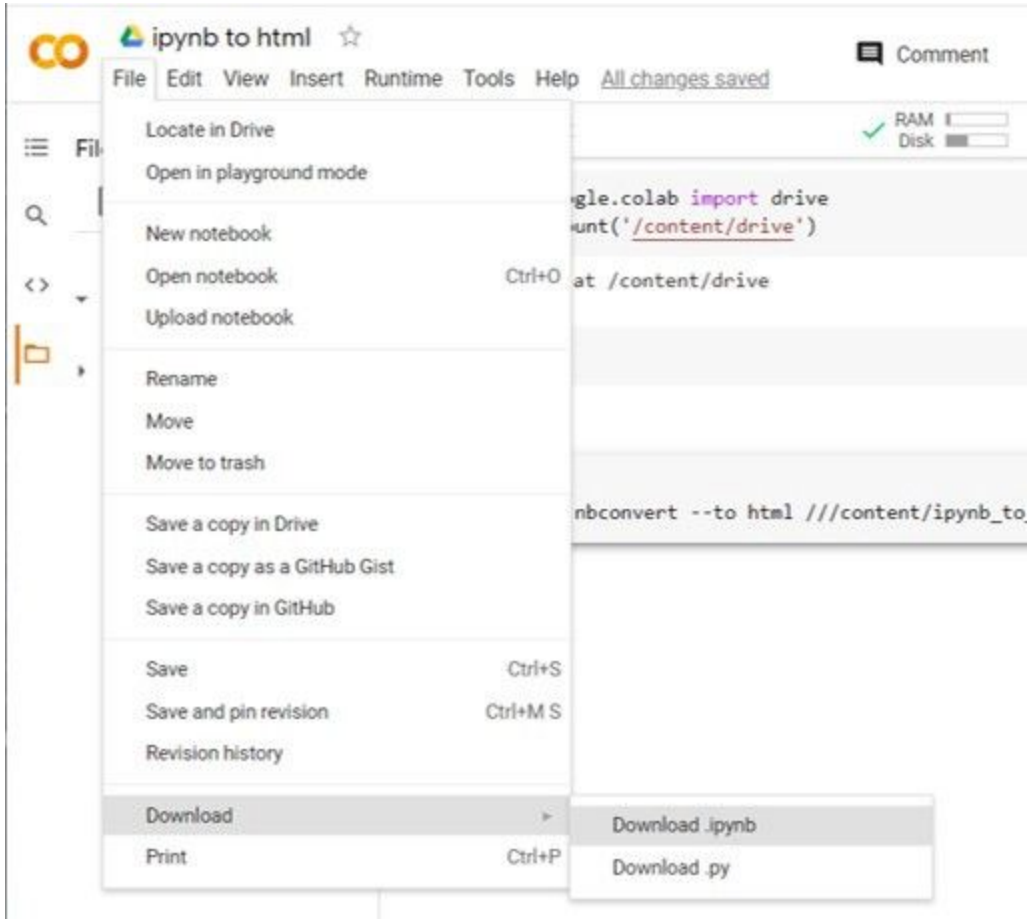


<u>LAB1 DEADLINE</u>:

11/10/2024

(midnight)

How to download the .ipynb file



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Setup

1

Links to COLAB exercise:

GitHub
Solution is coming after the deadline

Open github link (assignment file)

2

README

APAI24-LAB1-DNN-definition-and-training

Guidelines:

1. Start by reading the slides;
2. Then read the assignment;
3. Now complete the assignment: colab.

Open Jupyter notebook

3

Open in Colab

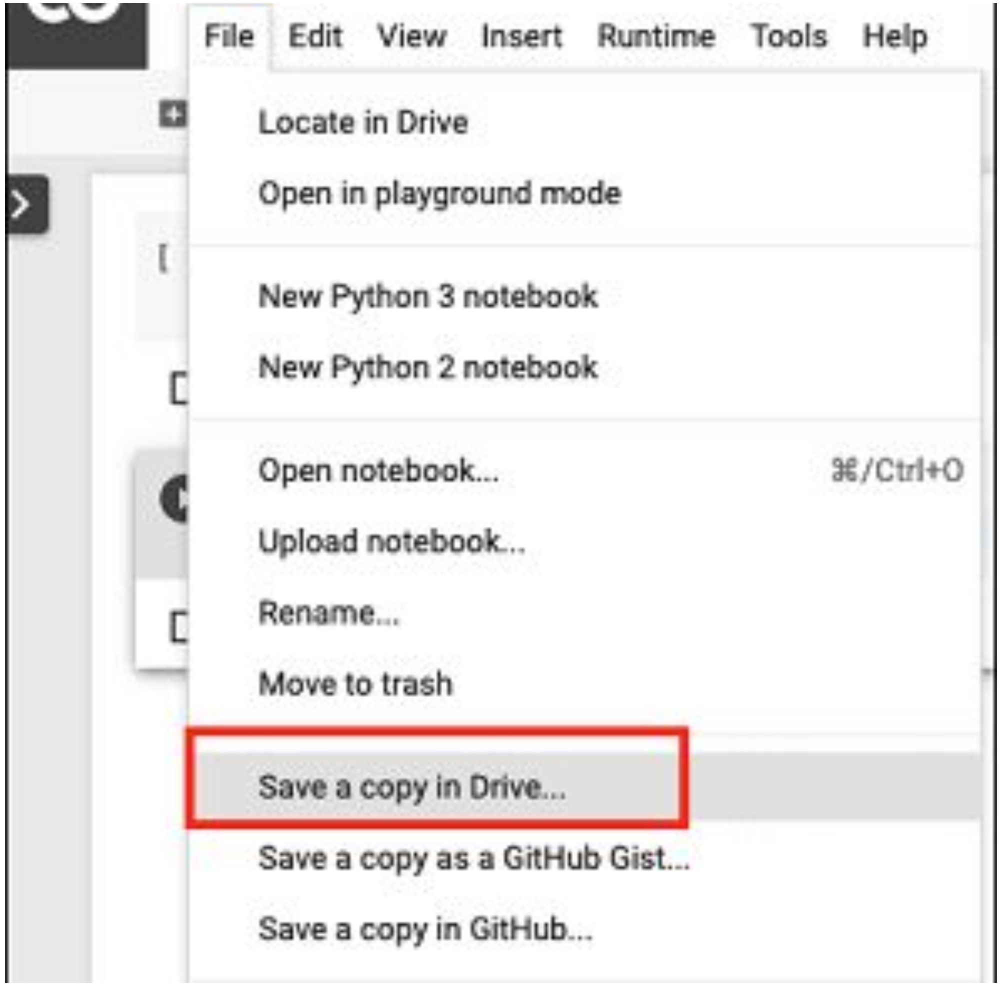
Open the .ipynb file in COLAB to modify it !

LAB1 APAI: DNN Definition & Training

Setup

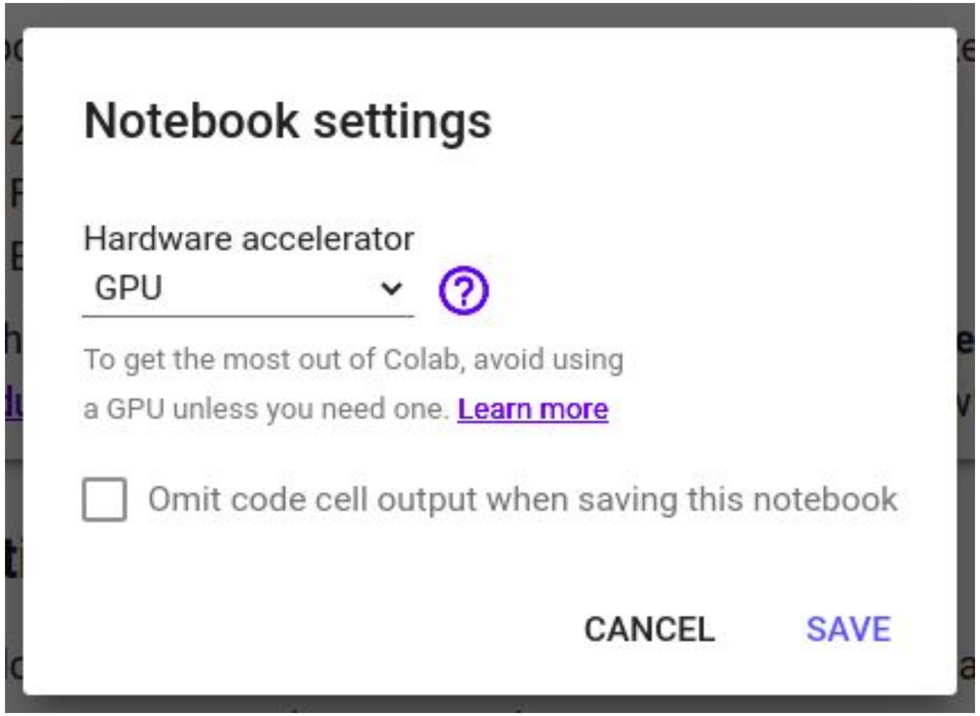
IMPORTANT:

Create your own copy of the COLAB notebook!



Others:

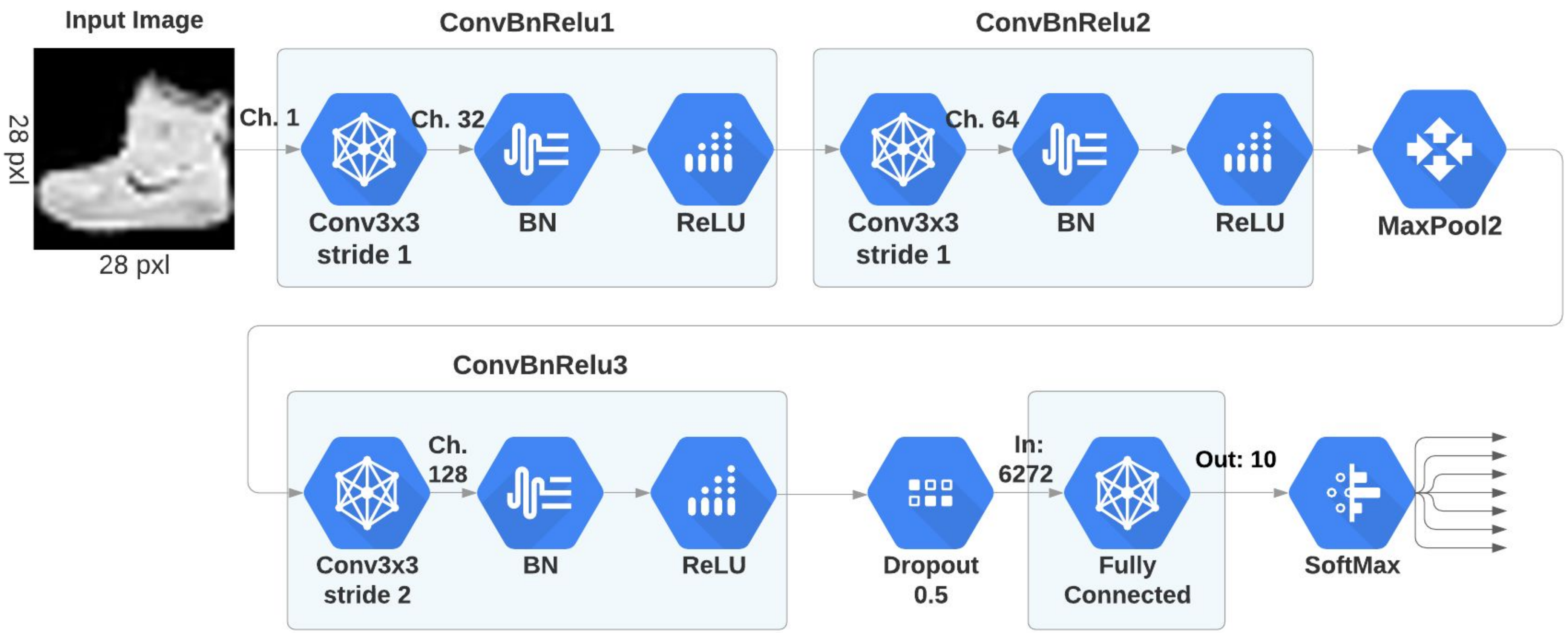
- Activate/deactivate GPU: Runtime -> Change runtime type
- **Note:** If you use for too much time the GPU, your account will be limited to CPU for 24h.



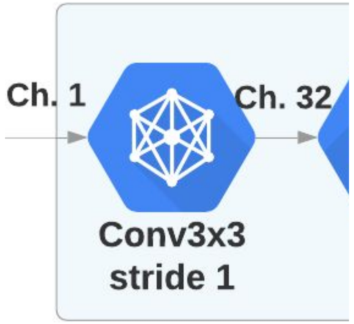
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

A BIT OF THEORY

LAB1 - CNN


Input Image
28 pxl
28 pxl
Ch. 1
ConvBnRelu1
Conv3x3
stride 1
Ch. 32
BN
ReLU
ConvBnRelu2
Conv3x3
stride 1
Ch. 64
BN
ReLU
MaxPool2
ConvBnRelu3
Conv3x3
stride 2
Ch.
128
BN
ReLU
Dropout
0.5
In:
6272
Fully
Connected
Out: 10
SoftMax

LAB1 - CNN



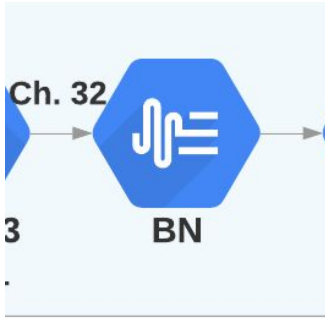
2D Convolution (Visualizer: https://ezyang.github.io/convolution-visualizer/)
Kernel: size of the convolution kernel (height, width)
Channels (input / output): channels of the input/output feature map
Stride: how many "pixels" of the input image you skip when the kernel slides
Padding: how many zeroes to add on the border of the input feature map

Formula for the size (height/width) of the output feature map (dilation here is 1):

- Input: $(N, C_{in}, H_{in}, W_{in})$ or (C_{in}, H_{in}, W_{in})
- Output: $(N, C_{out}, H_{out}, W_{out})$ or $(C_{out}, H_{out}, W_{out})$, where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times \text{padding}[0] - \text{dilation}[0] \times (\text{kernel\_size}[0] - 1) - 1}{\text{stride}[0]} + 1 \right\rfloor$$

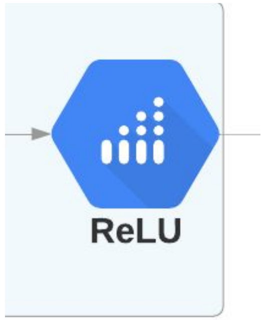
$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times \text{padding}[1] - \text{dilation}[1] \times (\text{kernel\_size}[1] - 1) - 1}{\text{stride}[1]} + 1 \right\rfloor$$



Batch Normalization
Layer that adjusts the statistics of the input feature map at training time. Input and output sizes are the same (does not change the NCHW shape)

num_features: channels of the input feature map.

LAB1 - CNN



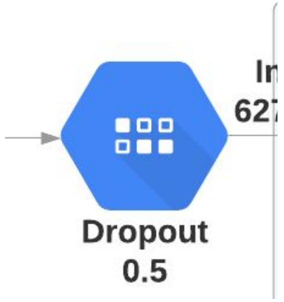
Rectified Linear Unit Activation
Activation function that passes all values above 0, zeroes values below zero.

Max Pooling
Reduces the size of the input feature map according to stride and kernel size.

- Input: (N, C, H_{in}, W_{in}) or (C, H_{in}, W_{in})
- Output: (N, C, H_{out}, W_{out}) or (C, H_{out}, W_{out}), where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 * \text{padding}[0] - \text{dilation}[0] \times (\text{kernel\_size}[0] - 1) - 1}{\text{stride}[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 * \text{padding}[1] - \text{dilation}[1] \times (\text{kernel\_size}[1] - 1) - 1}{\text{stride}[1]} + 1 \right\rfloor$$



Dropout
Layer used to prevent overfitting. In each training iteration, zeroes the channels of the input feature map with probability p.

Softmax Activation
Transforms the input feature array into a probability array.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The LAB starts now !