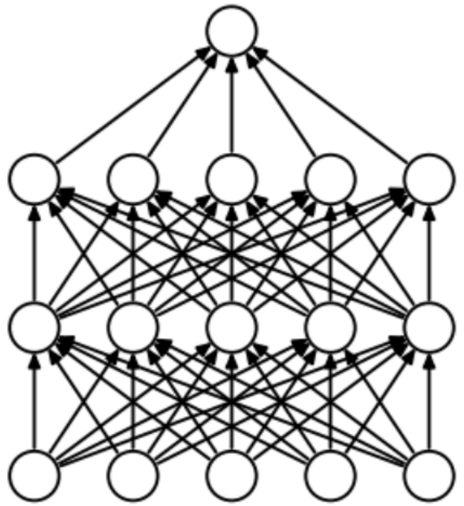
Deep Learning
Part I: Introduction

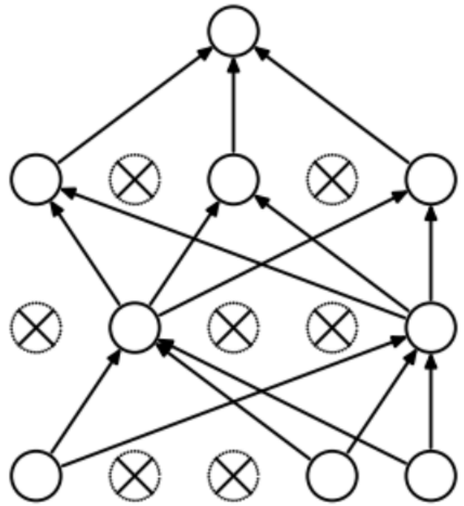
Maurizio Pierini

Dropout Layer

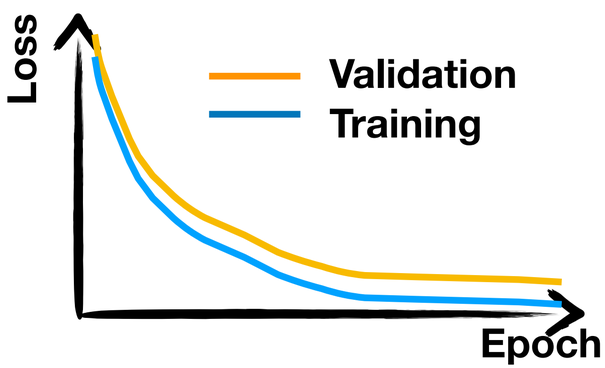
- *A special kind of layer, introduced for regularisation purpose*
- *Randomly drop links between neurons, with probability p*
- *The connections are re-established during the validation and inference steps*
- *Typical sign of it: invert hierarchy between training and validation loss*

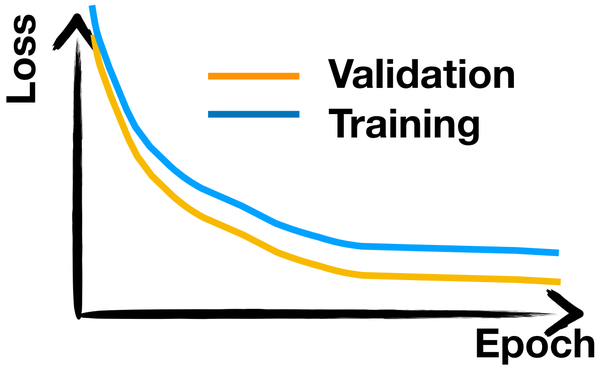


(a) Standard Neural Net



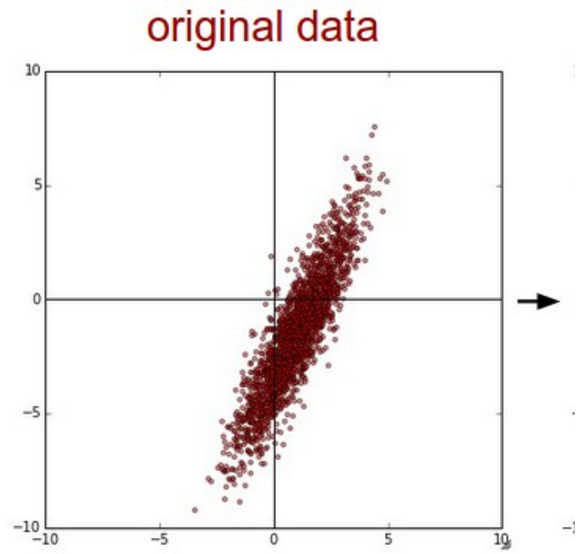
(b) After applying dropout.

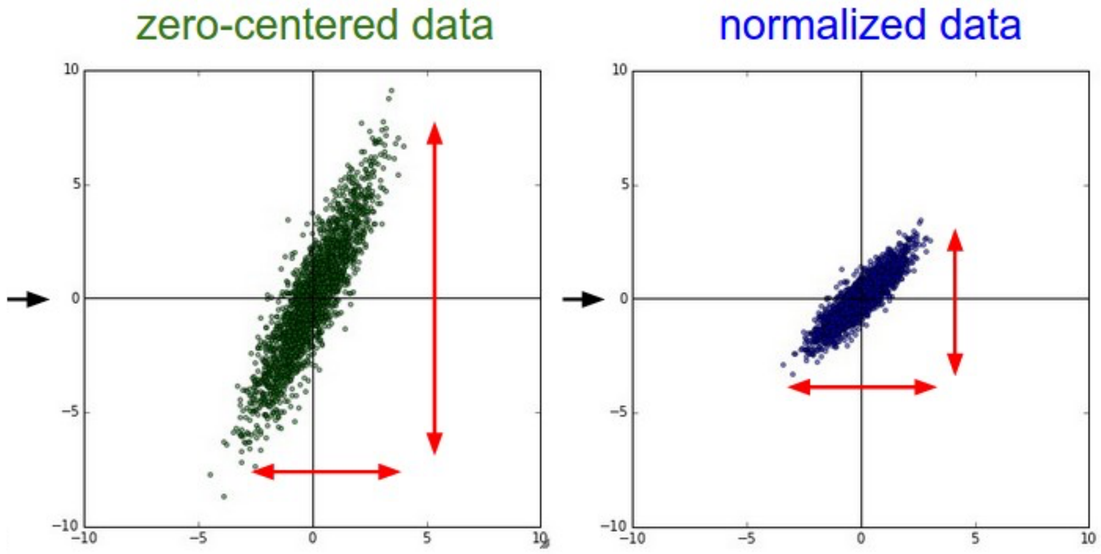




BatchNormalization Layer

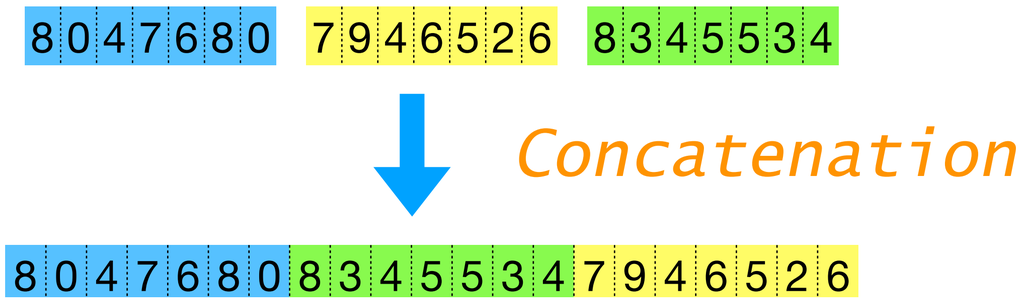
- *It is good practice to give normalized inputs to a layer*
  - *With all inputs having the same order of magnitude, all weights are equal important in the gradient*
  - *Prevents explosion of the loss function*
- *This can be done automatically with BatchNormalization*
  - *non-learnable shift and scale parameters, adjusted batch by batch*

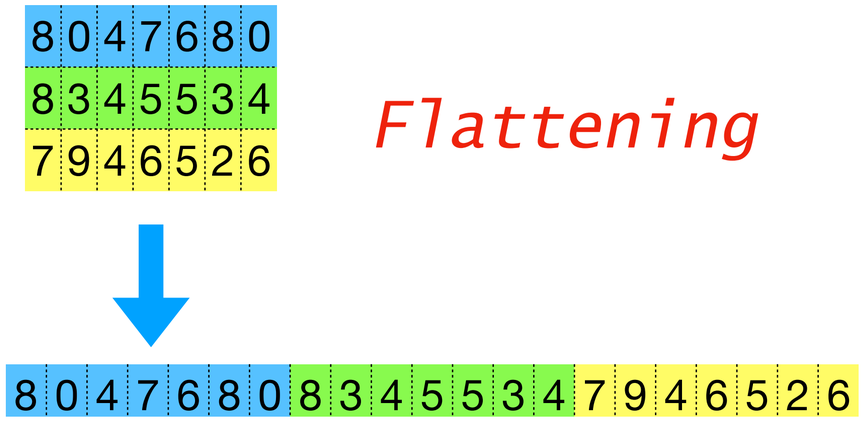




More complex structures

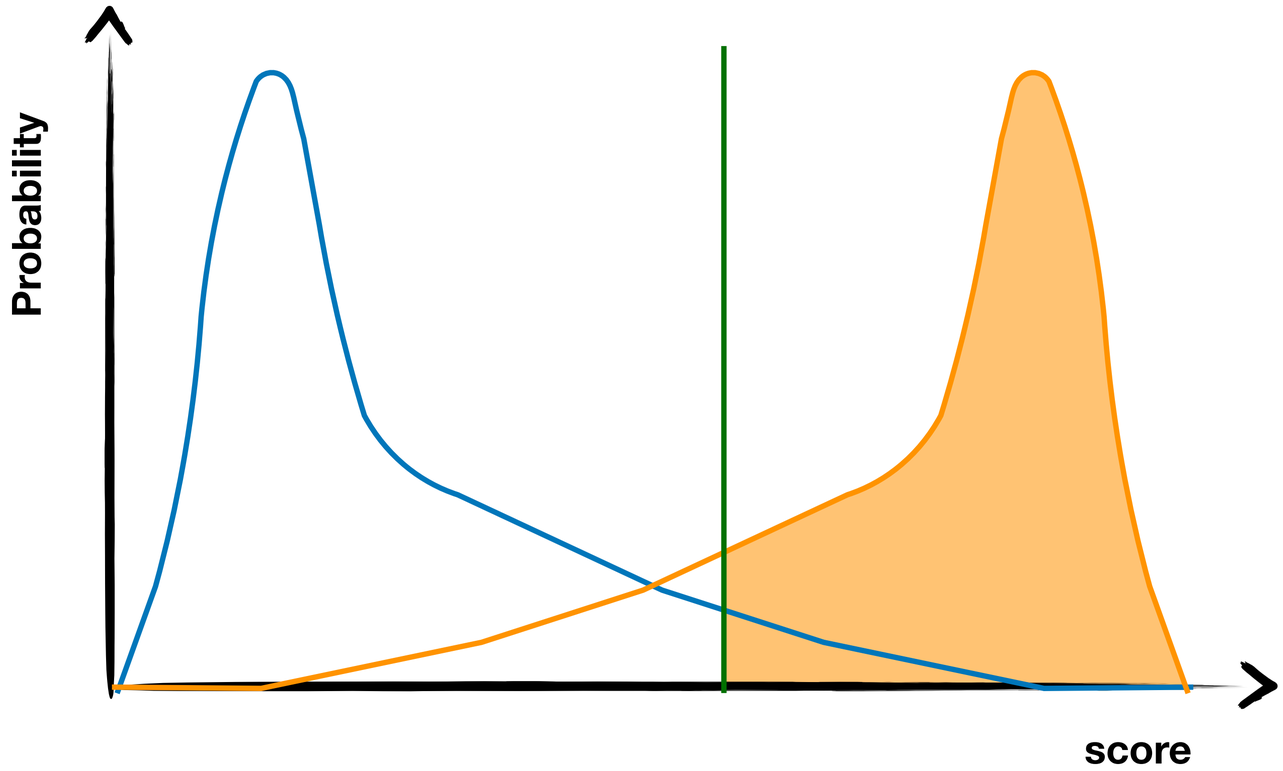
- *Dense NN architectures can be made more complex*
  - *Multiple inputs*
  - *Multiple outputs*
  - *Different networks branches*
- *This is possible thanks to layer-manipulation layers*
  - *Add, Subtract, etc.*
  - *Concatenation*
  - *Flattening*
- *All these operations are usually provided with NN training libraries*





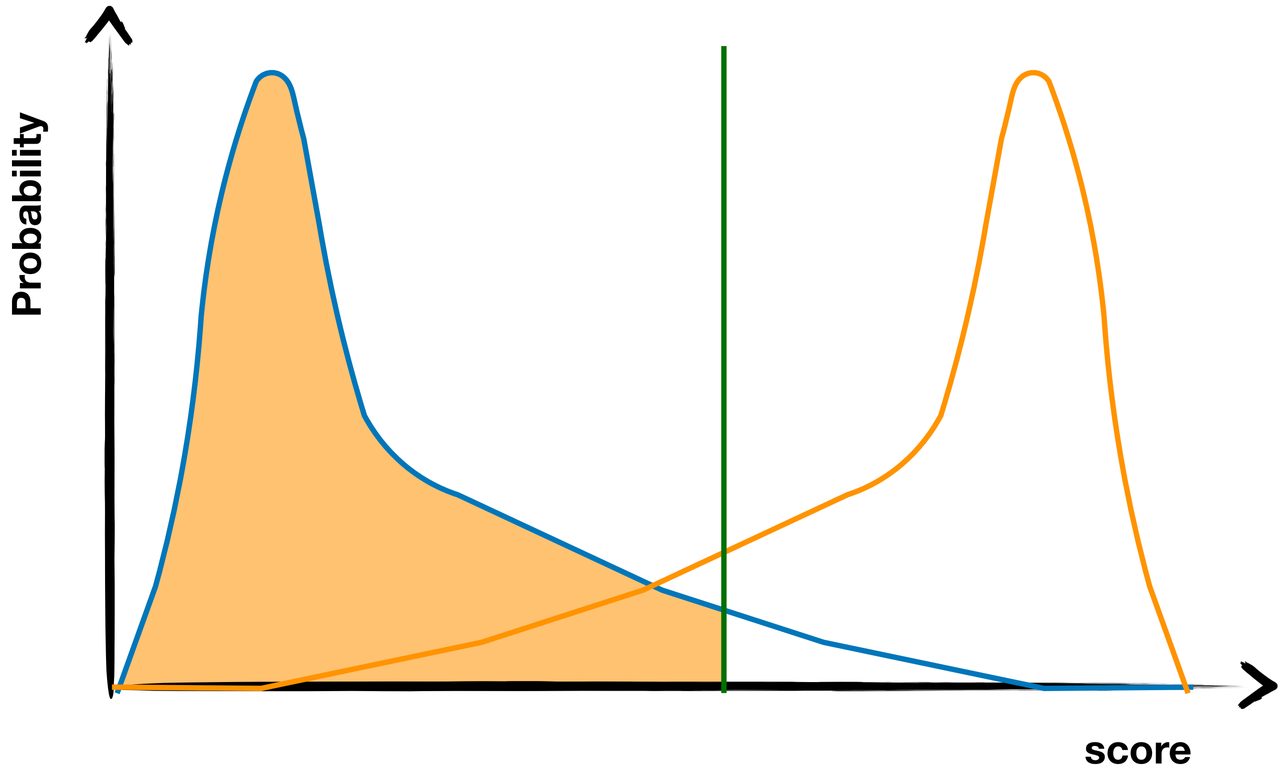
Classifier metrics

- A given threefold defines the following qualities
  - True-positives: Class-1 events above the threshold
  - True-negatives: Class-0 events below the threshold
  - False-positives: Class-0 events above the threshold
  - False-negatives: Class-1 events below the threshold



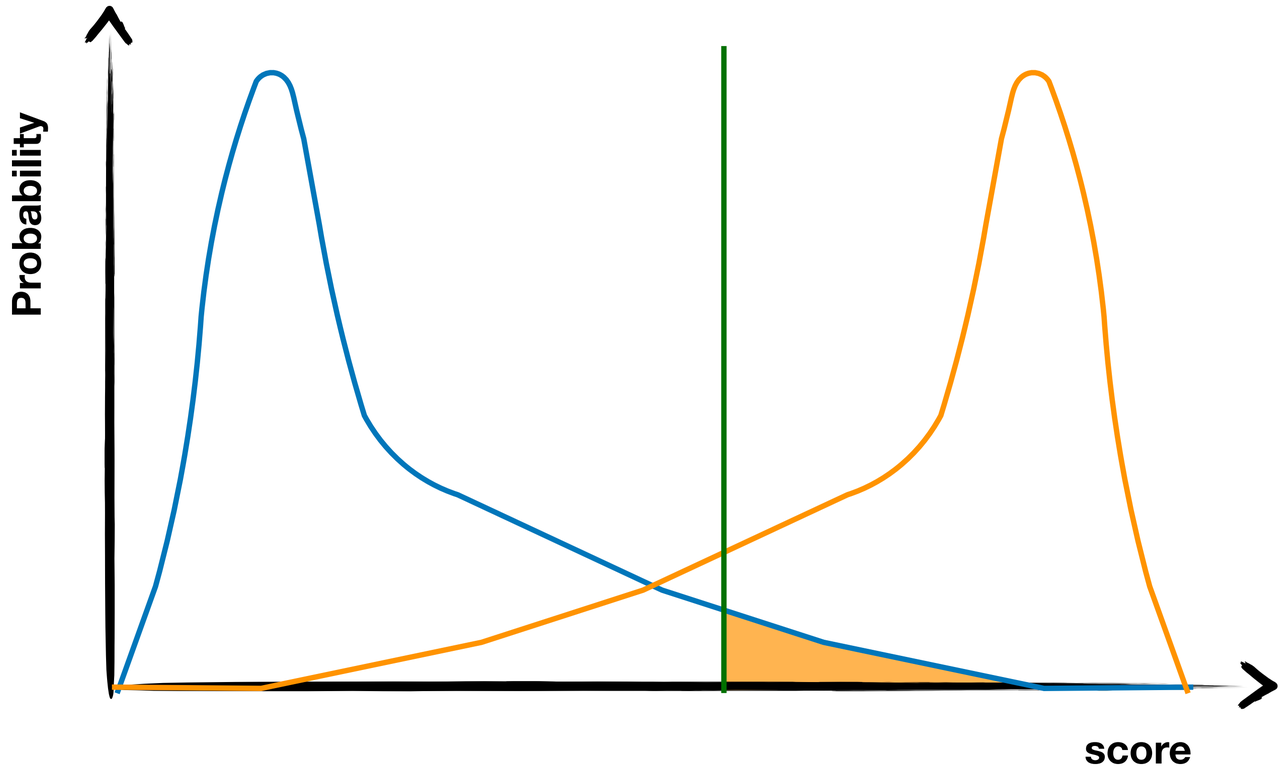
Classifier metrics

- *A given threefold defines the following qualities*
  - *True-positives: Class-1 events above the threshold*
  - *True-negatives: Class-0 events below the threshold*
  - *False-positives: Class-0 events above the threshold*
  - *False-negatives: Class-1 events below the threshold*



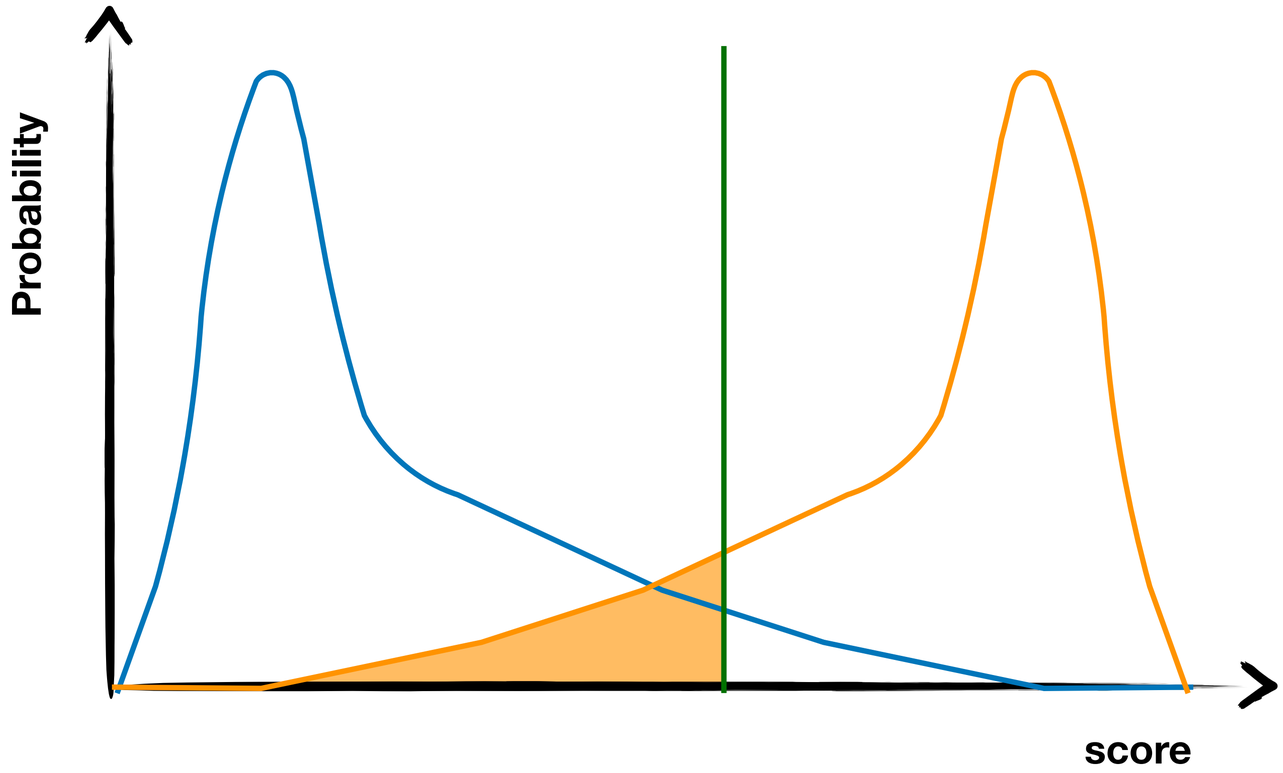
Classifier metrics

- *A given threefold defines the following qualities*
  - *True-positives: Class-1 events above the threshold*
  - *True-negatives: Class-0 events below the threshold*
  - *False-positives: Class-0 events above the threshold*
  - *False-negatives: Class-1 events below the threshold*



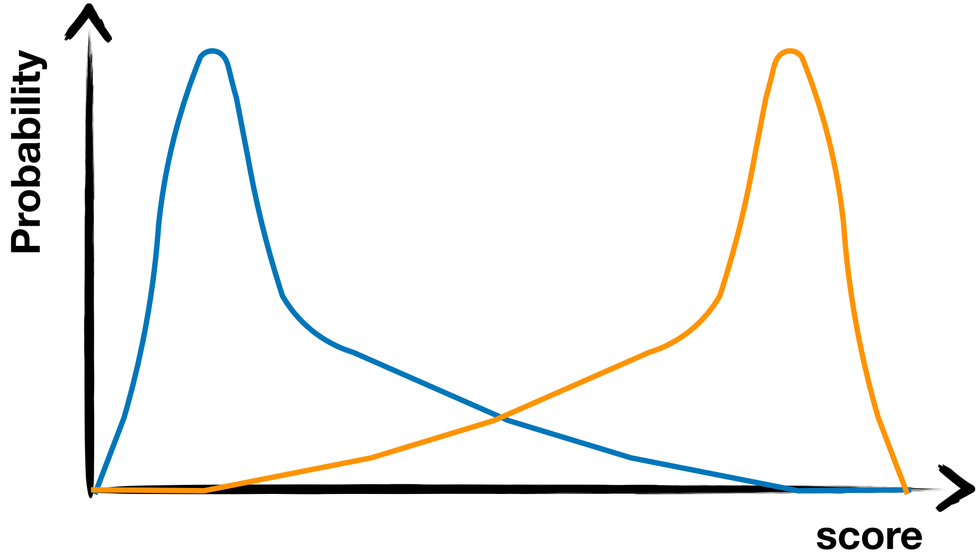
Classifier metrics

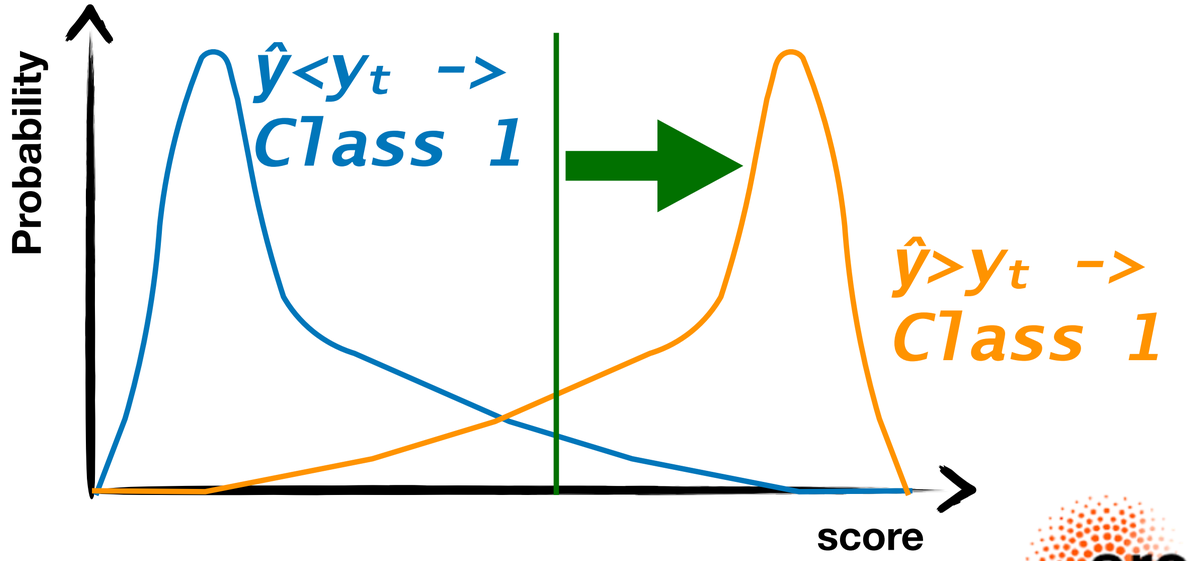
- *A given threefold defines the following qualities*
  - *True-positives: Class-1 events above the threshold*
  - *True-negatives: Class-0 events below the threshold*
  - *False-positives: Class-0 events above the threshold*
  - *False-negatives: Class-1 events below the threshold*



Classifier metrics

- Consider a binary classifier
- Its output $\hat{y}$ is a number in [0,1]
- If well trained, value should be close to 0 (1) for class-0 (class-1) examples
- One usually defines a threshold y_t such that:
  - $\hat{y} > y_t$ -> Class 1
  - $\hat{y} < y_t$ -> Class 0



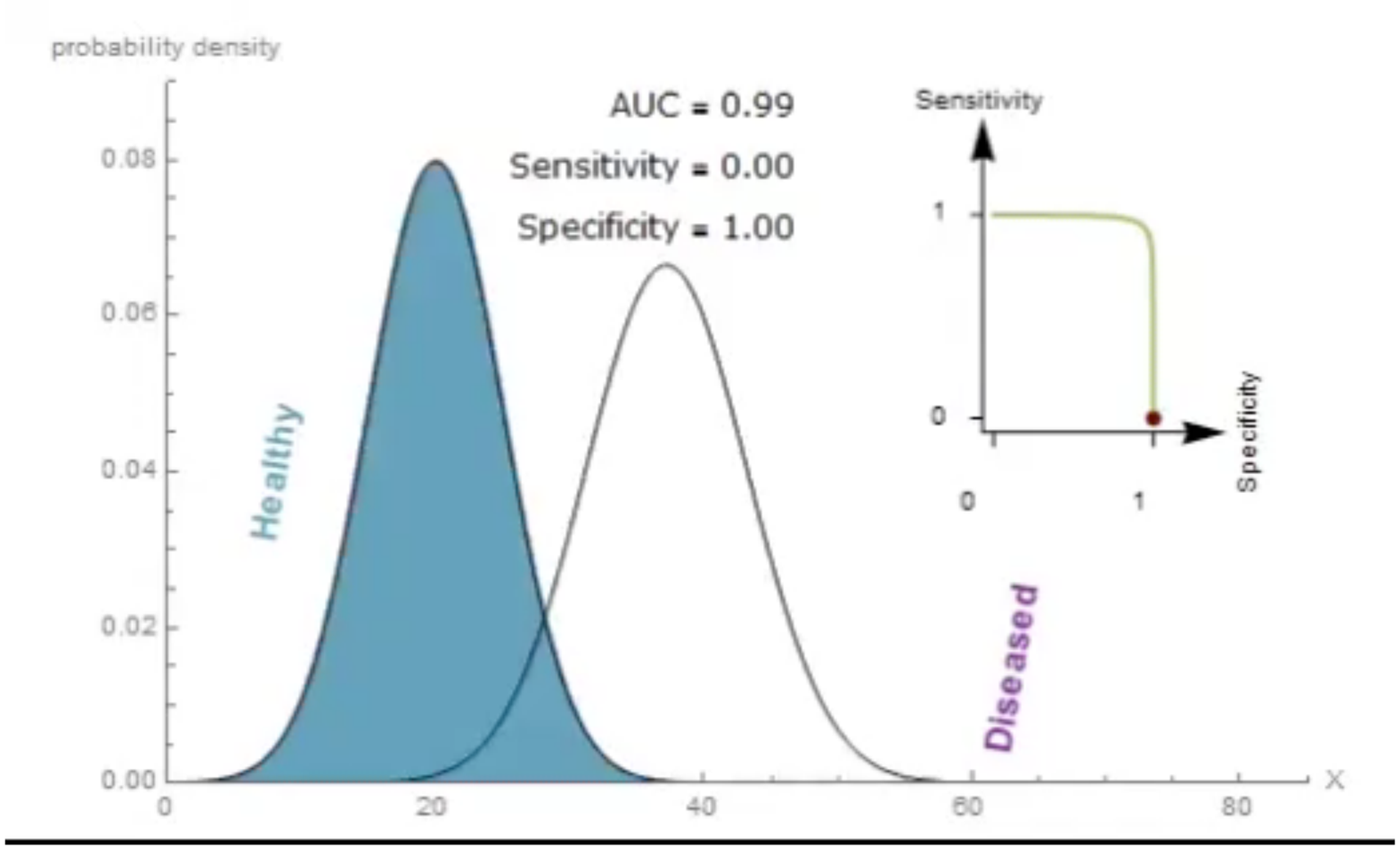


Classifier metrics

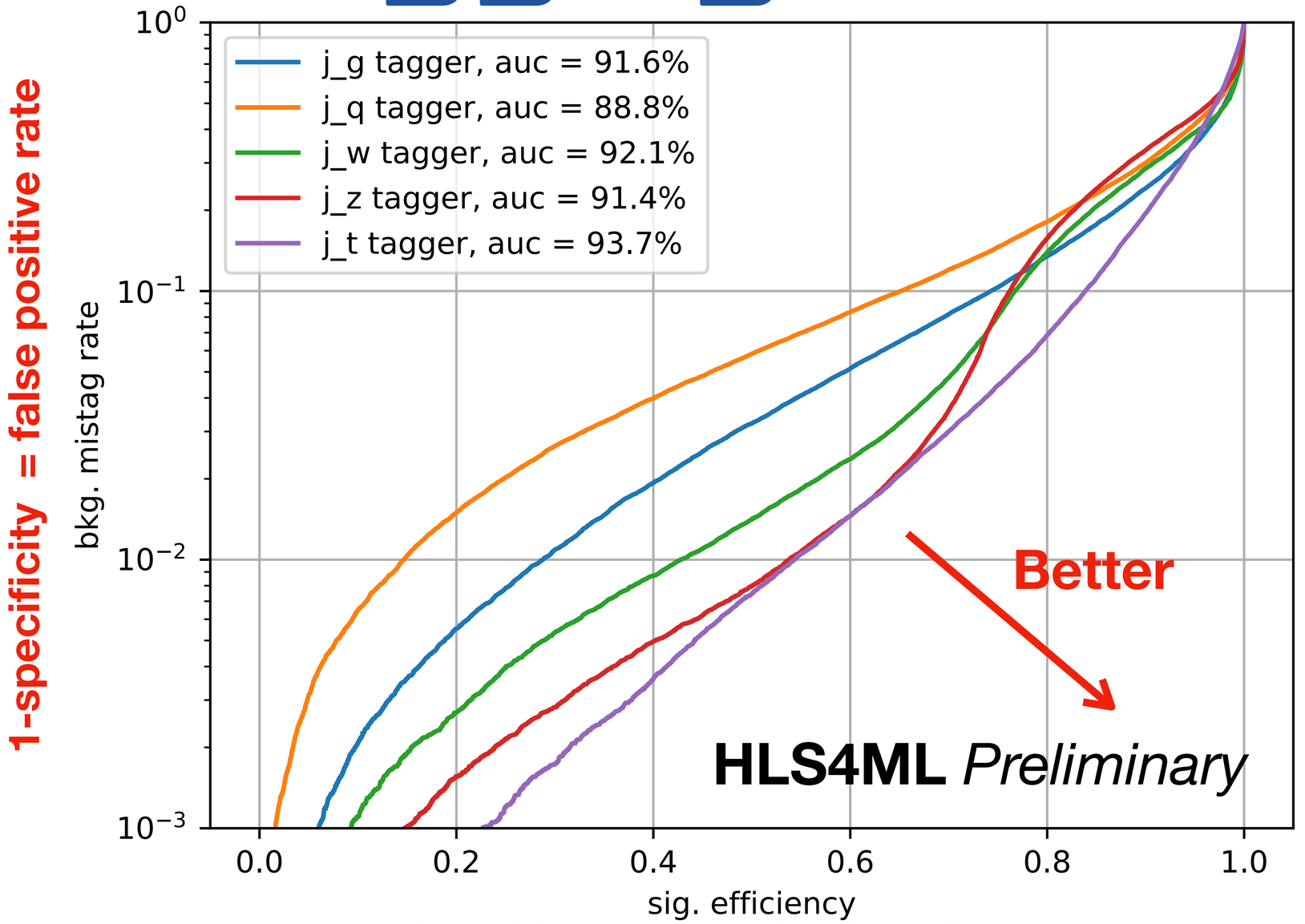
- *Starting ingredients are true positive (TP) and true negative (TN) rates*
- *Accuracy: (TP+TN)/Total*
  - *The fraction of events correctly classified*
- *Sensitivity: TP/(Total positive)*
  - *AKA signal efficiency in HEP*
- *Specificity: TN/(Total negative)*
  - *AKA mistag rate in HEP*

European Research Council

Receiver operating characteristic



Jet tagging ROC curve

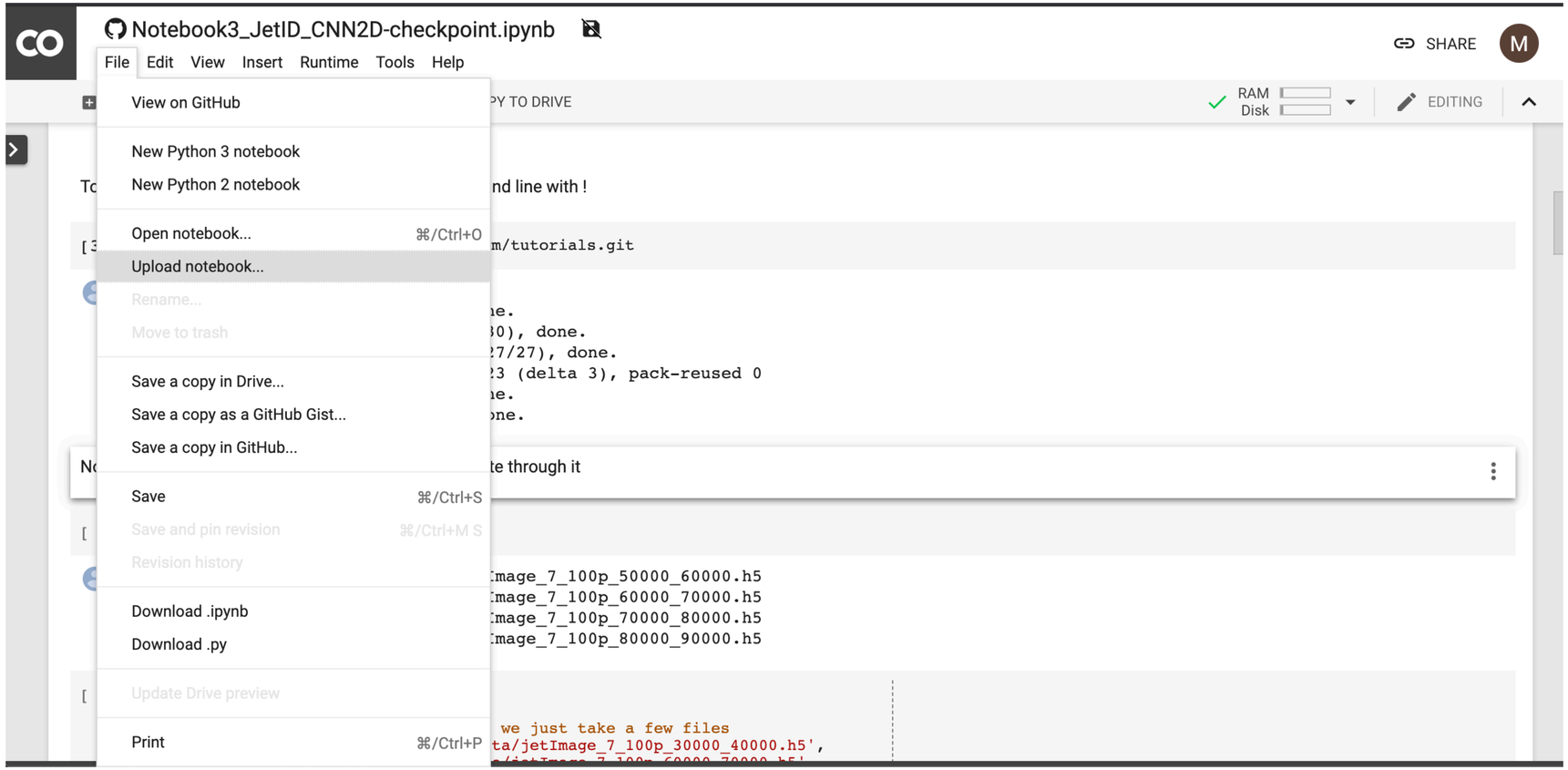


1-specificity = false positive rate

Sensitivity = True Positive Rate

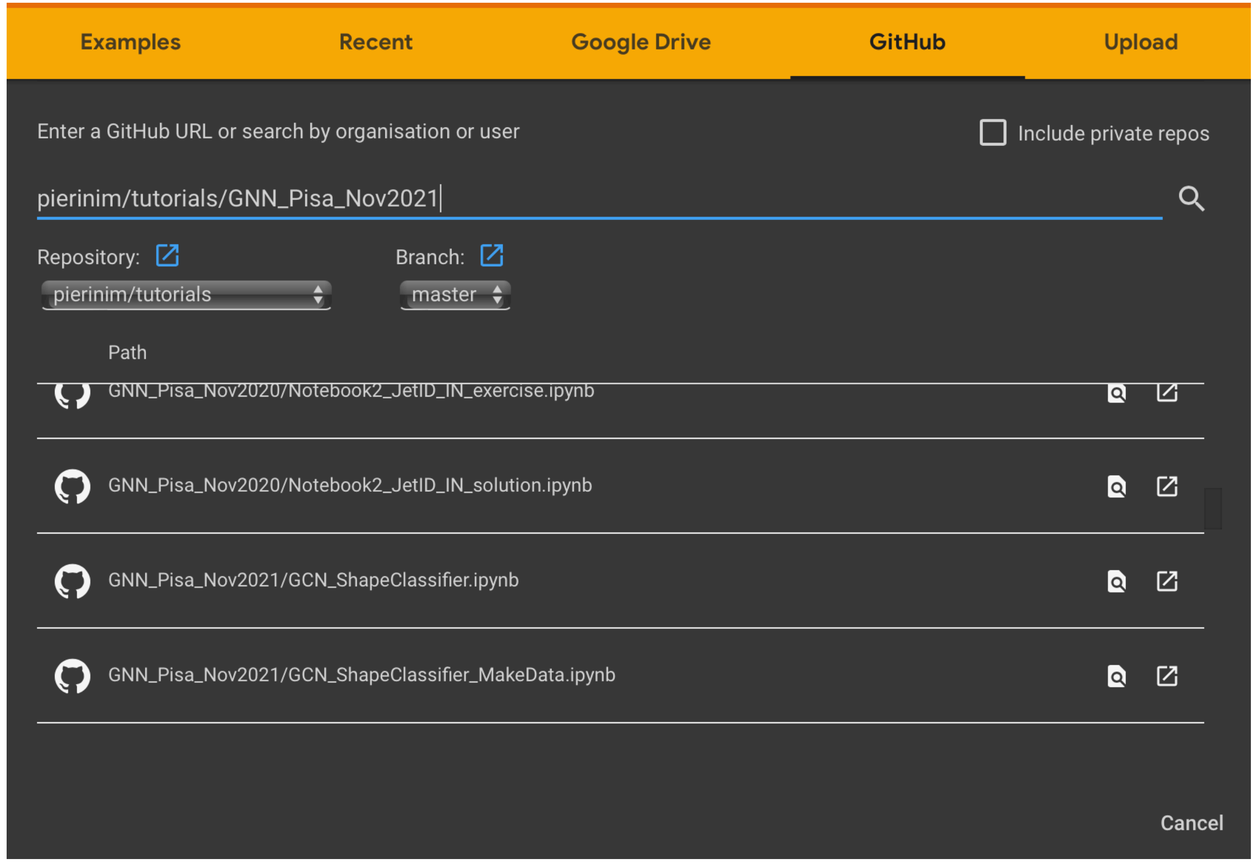
Step1: Open Notebook on Colab

- Go to https://colab.research.google.com

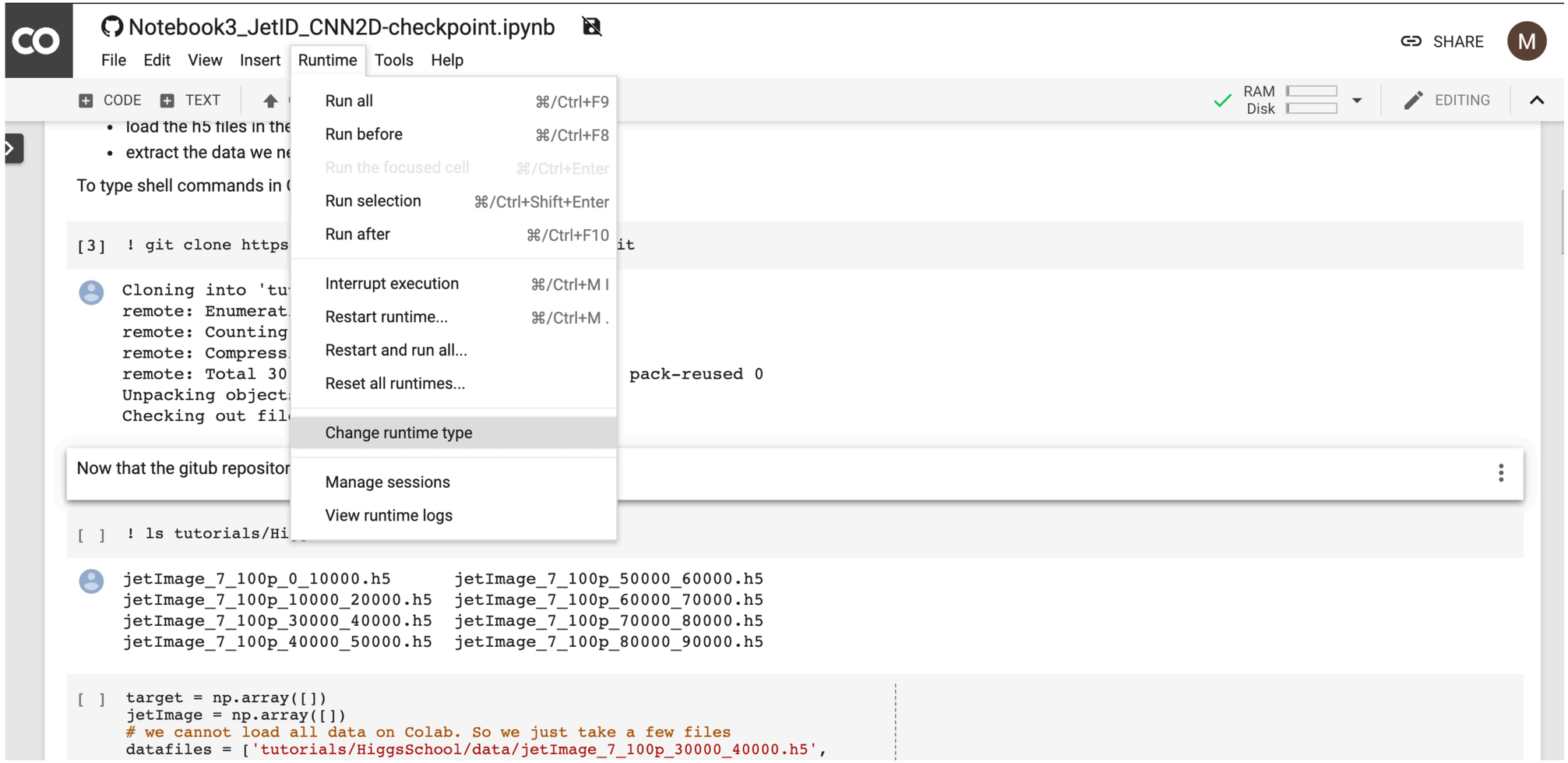


Step2: import the Tutorial from gitlab

- *Click on the GITHUB tab*
- *Specify the repository pierinim/tutorials*
- *Click on the notebook*



Set your resources to use GPUs



Set your resources to use GPUs