Machine Learning Tutorials for HEP

Jie Ren

Wuhan, Oct 12, 2021

- Tutorial 1 – Higgs boson discovery (BDT)
- Tutorial 2 – Higgs boson discovery (FCNN)
- Tutorial 3 – Jet tagging (CNN)

DATA

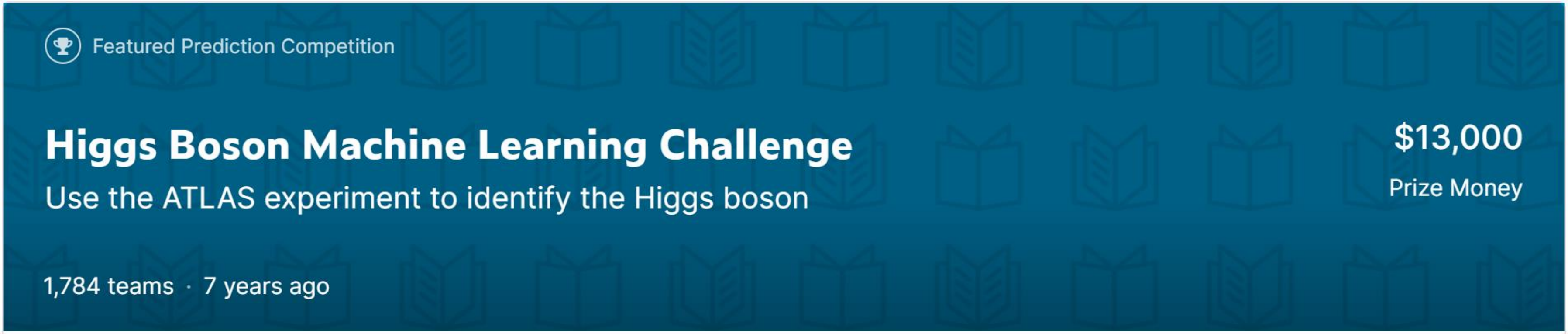
https://gitee.com/jieren21/hepml-tutorials

TUTORIAL 1

- Introduction
- Dataset
- Toolkits
- Steps
- Performance

Higgs boson discovery (BDT)

Tutorial 1 Introduction



Featured Prediction Competition

Higgs Boson Machine Learning Challenge

Use the ATLAS experiment to identify the Higgs boson

$13,000

Prize Money

1,784 teams · 7 years ago

Paris-Saclay Center for Data Science

CERN

ATLAS EXPERIMENT

LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

Inria INVENTORS FOR THE DIGITAL WORLD

Google

https://www.kaggle.com/c/higgs-boson

2014

- ✓ $H \to VV(\gamma\gamma, WW, ZZ)$
- ➢ $H \to \tau(\to e/\mu + 2\nu)\ \tau(\text{hadrons})$

Tutorial 2 Higgs boson discovery (BDT)

training.csv

250000 events

an ID column

30 feature columns

a weight column

a label column

Primitives

PRI_tau_pt
PRI_tau_eta
PRI_tau_phi
PRI_lep_pt
PRI_lep_eta
PRI_lep_phi
PRI_met
PRI_met_phi
PRI_met_sumet
PRI_jet_num
PRI_jet_leading_pt
PRI_jet_leading_eta
PRI_jet_leading_phi
PRI_jet_subleading_pt
PRI_jet_subleading_eta
PRI_jet_subleading_phi
PRI_jet_all_pt

Derived

DER_mass_MMC
DER_mass_transverse_met_lep
DER_mass_vis
DER_pt_h
DER_deltaeta_jet_jet
DER_mass_jet_jet
DER_prodeta_jet_jet
DER_deltar_tau_lep
DER_pt_tot
DER_sum_pt
DER_pt_ratio_lep_tau
DER_met_phi_centrality
DER_lep_eta_centrality

Tutorial 1 Toolkits

Python 3

- Python is a programming language that lets you work more quickly and integrate your systems more effectively.

Scikit-learn

- Machine Learning in Python
- Simple and efficient tools for predictive data analysis

Matplotlib

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Tutorial 1 Steps

1. **Preprocess data**

   Format
   Plot
   Split into training set and validation set

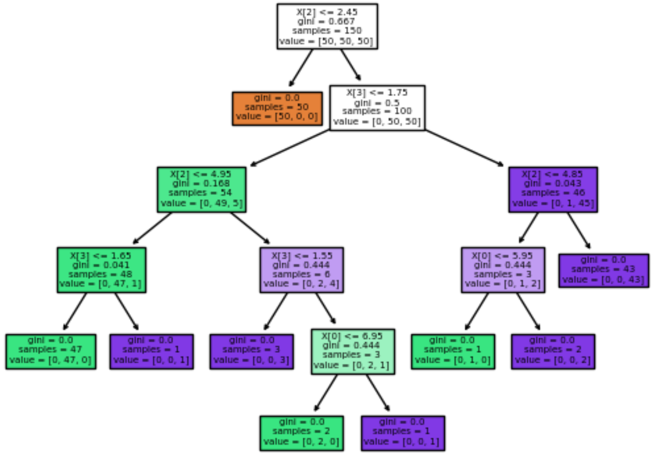
2. **Build model**

   Decision tree classifier (Gini impurity)
   AdaBoost

3. **Train model**

4. **Make predictions**

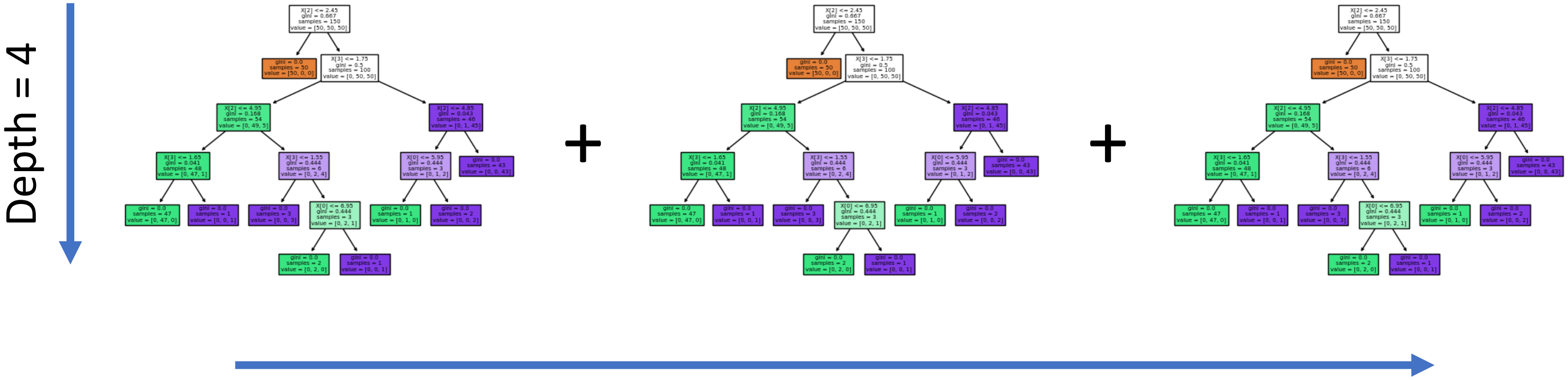
5. **Evaluate model**


X[2] <= 2.45 gini = 0.667 samples = 150 value = [50, 50, 50]
gini = 0.0 samples = 50 value = [50, 0, 0]
X[3] <= 1.75 gini = 0.5 samples = 100 value = [0, 50, 50]
X[2] <= 4.95 gini = 0.168 samples = 54 value = [0, 49, 5]
X[2] <= 4.85 gini = 0.043 samples = 46 value = [0, 1, 45]
X[3] <= 1.65 gini = 0.041 samples = 48 value = [0, 47, 1]
X[3] <= 1.55 gini = 0.444 samples = 6 value = [0, 2, 4]
X[0] <= 5.95 gini = 0.444 samples = 3 value = [0, 1, 2]
gini = 0.0 samples = 43 value = [0, 0, 43]
gini = 0.0 samples = 47 value = [0, 47, 0]
gini = 0.0 samples = 1 value = [0, 0, 1]
gini = 0.0 samples = 3 value = [0, 0, 3]
X[0] <= 6.95 gini = 0.444 samples = 3 value = [0, 2, 1]
gini = 0.0 samples = 1 value = [0, 1, 0]
gini = 0.0 samples = 2 value = [0, 0, 2]
gini = 0.0 samples = 2 value = [0, 2, 0]
gini = 0.0 samples = 1 value = [0, 0, 1]

https://scikit-learn.org/stable/modules/tree.html#classification

https://scikit-learn.org/stable/modules/ensemble.html#adaboost

Tutorial 1 Steps


X[2] <= 2.45 gini = 0.667 samples = 150 value = [50, 50, 50]
gini = 0.0 samples = 50 value = [50, 0, 0]
X[3] <= 1.75 gini = 0.5 samples = 100 value = [0, 50, 50]
X[2] <= 4.95 gini = 0.168 samples = 54 value = [0, 49, 5]
X[2] <= 4.85 gini = 0.043 samples = 46 value = [0, 1, 45]
X[3] <= 1.65 gini = 0.041 samples = 48 value = [0, 47, 1]
X[3] <= 1.55 gini = 0.444 samples = 6 value = [0, 2, 4]
X[0] <= 5.95 gini = 0.444 samples = 3 value = [0, 1, 2]
gini = 0.0 samples = 43 value = [0, 0, 43]
gini = 0.0 samples = 47 value = [0, 47, 0]
gini = 0.0 samples = 1 value = [0, 0, 1]
gini = 0.0 samples = 3 value = [0, 0, 3]
X[0] <= 6.95 gini = 0.444 samples = 3 value = [0, 2, 1]
gini = 0.0 samples = 1 value = [0, 1, 0]
gini = 0.0 samples = 2 value = [0, 0, 2]
gini = 0.0 samples = 2 value = [0, 2, 0]
gini = 0.0 samples = 1 value = [0, 0, 1]
+
+

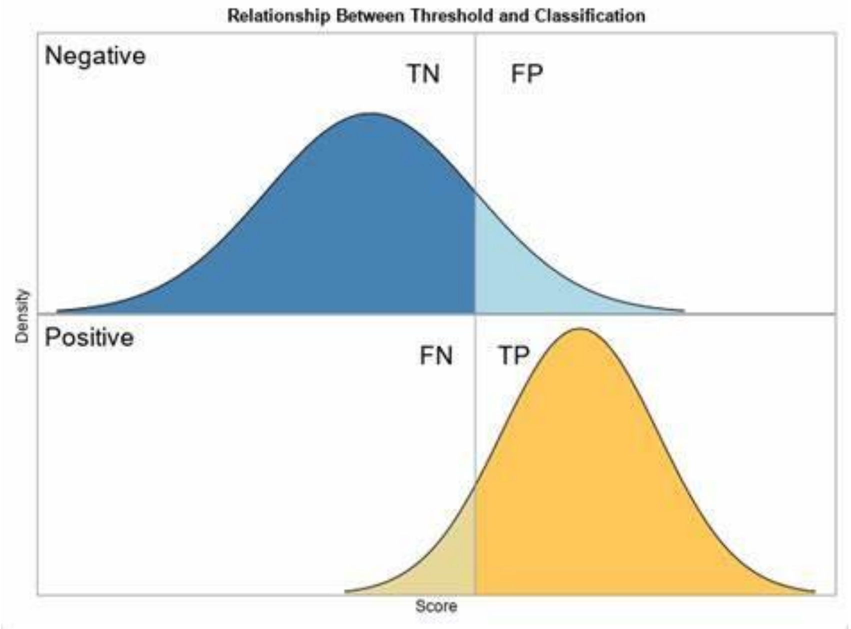
Depth = 4

Number of estimators = 5

$$
\begin{aligned}
y &= \alpha_1 f_1(x) + \epsilon_1 \\
&= \alpha_1 f_1(x) + \alpha_2 f_2(x) + \epsilon_2 \\
&= \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \epsilon_3 \\
&= \cdots
\end{aligned}
$$

https://scikit-learn.org/stable/modules/tree.html#classification

https://scikit-learn.org/stable/modules/ensemble.html#adaboost



- **Score threshold**
- **Selection efficiencies**
    - TPR for signal
    - FPR for background
- **Detection significance**
    - AMS: approximate median significance

$$\mathrm{AMS} = \sqrt{2\left(\left(s + b + b_{\mathrm{reg}}\right)\ln\left(1 + \frac{s}{b + b_{\mathrm{reg}}}\right) - s\right)}$$

TUTORIAL 2

- Toolkits
- Steps
- Best Model Selection

Higgs boson discovery (FCNN)

Higgs challenge

Tutorial 2 Toolkits

Python 3

- Python is a programming language that lets you work more quickly and integrate your systems more effectively.

PyTorch

- An open source machine learning framework that accelerates the path from research prototyping to production deployment.

Matplotlib

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Tutorial 2 Steps

1 **Preprocess data**

Format

Plot

Normalization

Split into training set and validation set

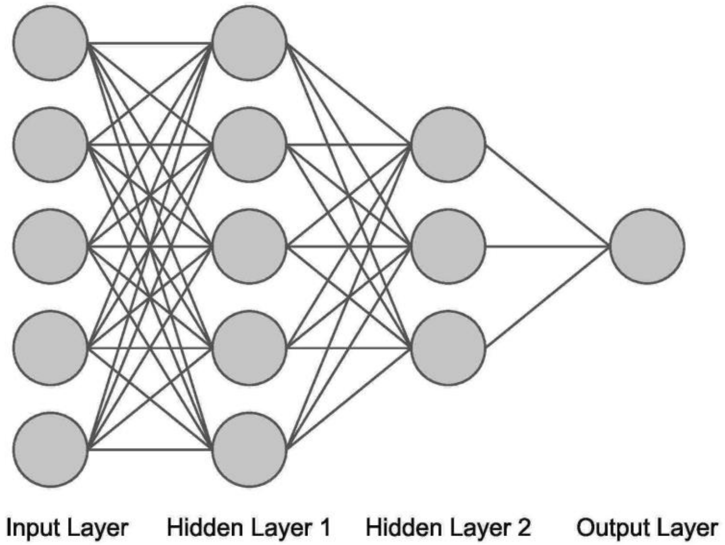
2 **Build model**

FCNN

3 **Train and validate iteratively**

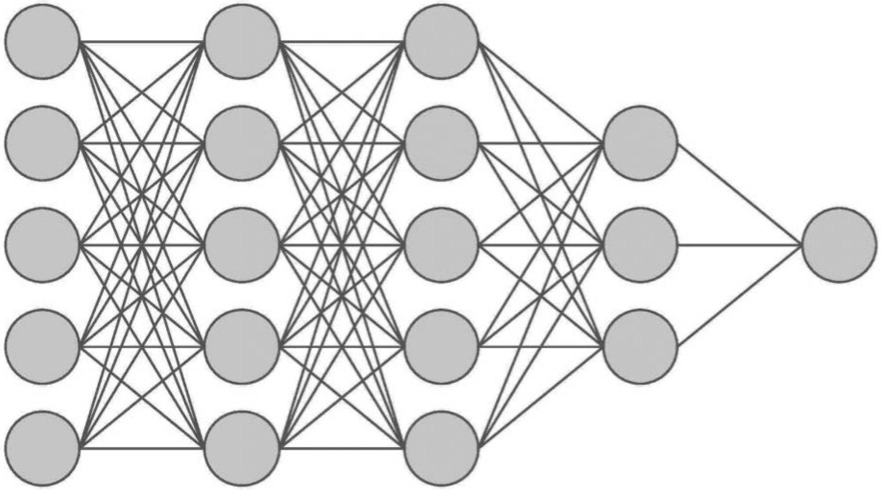
4 **Make predictions**

5 **Evaluate model**



Tutorial 2 Steps

Model



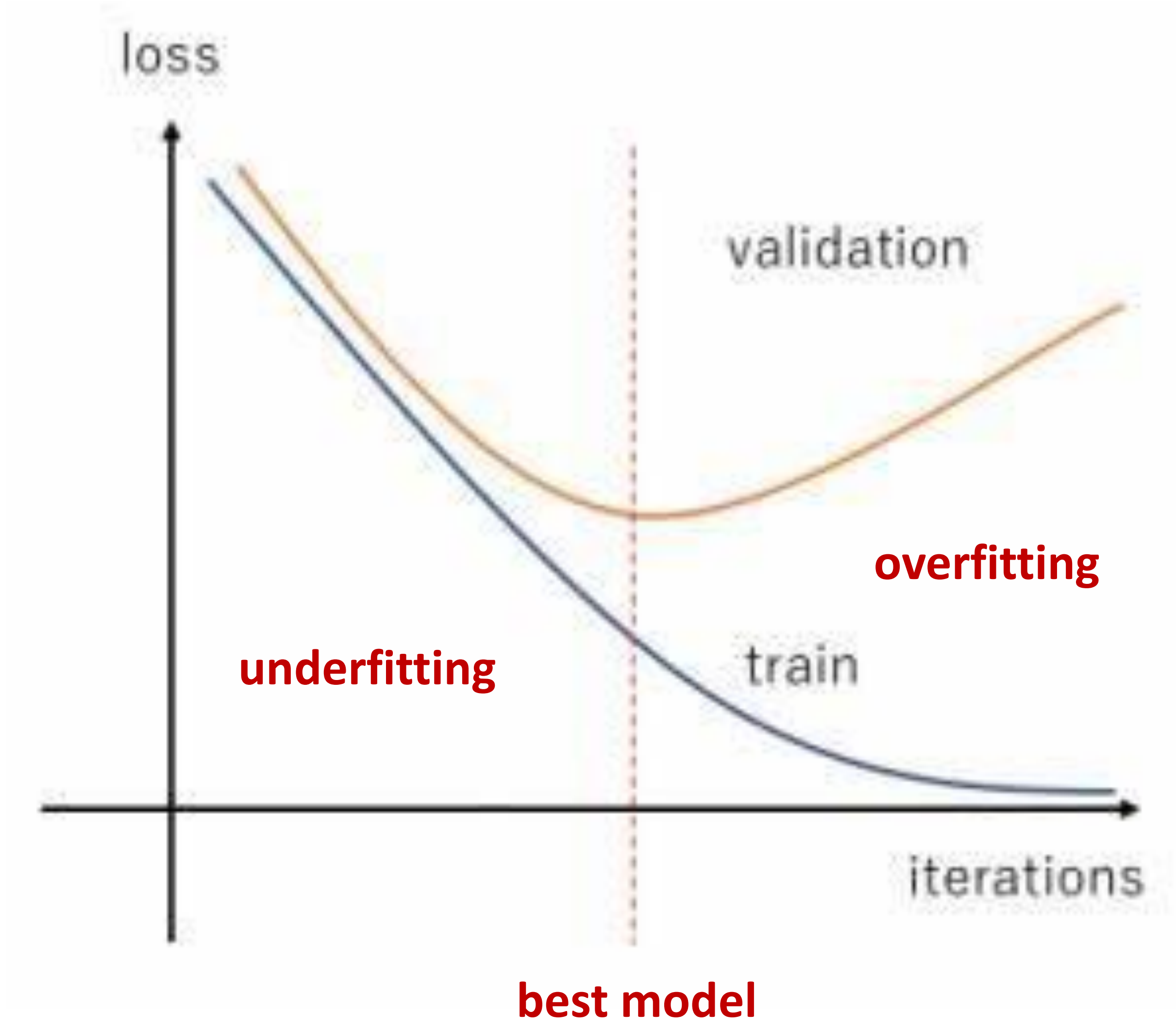
30	64	128	64	1
	ReLU	ReLU	ReLU	Sigmoid

Loss

$$L = \frac{1}{N}\sum_i I(t_i = 1)\log y(\boldsymbol{x}_i) + I(t_i = 0)\log(1 - y(\boldsymbol{x}_i))$$

Binary cross entropy

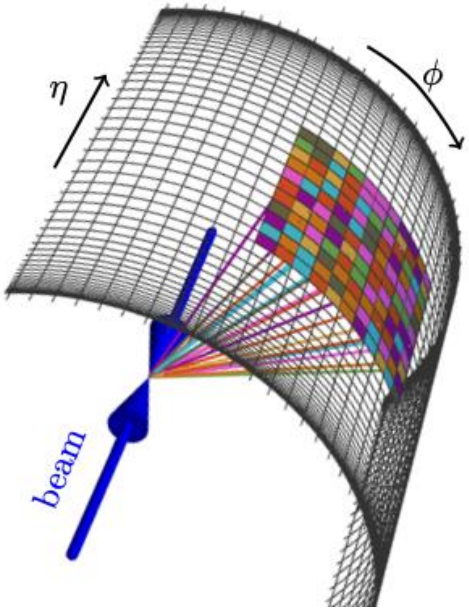
Tutorial 2 Best Model Selection



TUTORIAL 3

- Introduction
- Dataset
- Toolkit
- Steps
- Performance

Jet Tagging (CNN)



Jet Image

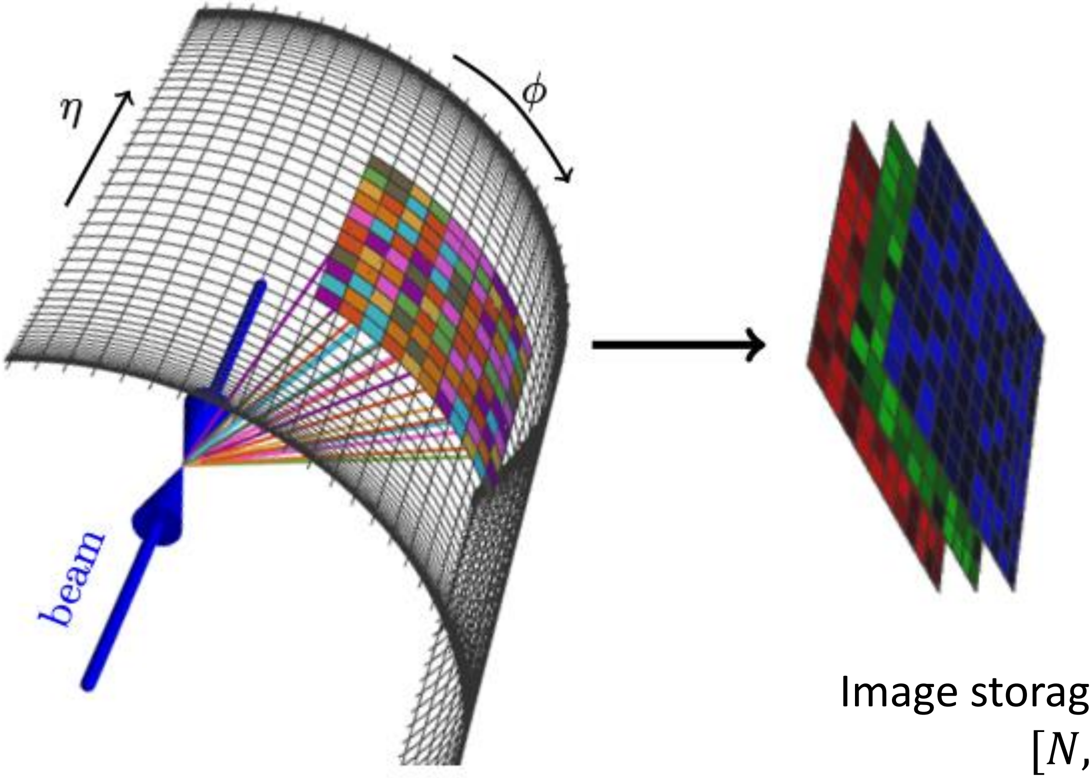
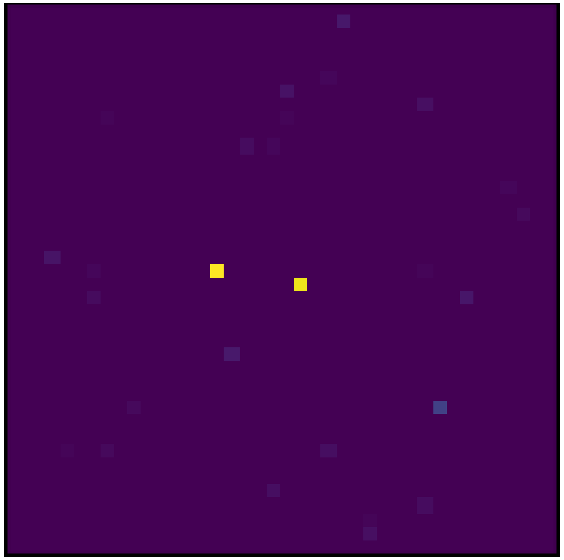


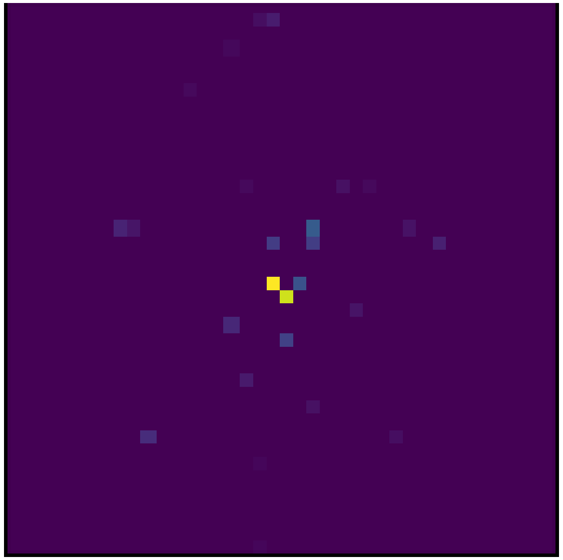
Image storage in NVIDIA format
$[N, C, H, W]$

arXiv:2106.07018

Photon-jet



QCD-jet



40×40 pixels

$\Delta\eta \times \Delta\phi = 0.02 \times 0.02$

$R_j = 0.4$

Tutorial 3 Introduction

train_data.npz

10000 jets

Images [10000, 5, 40, 40]

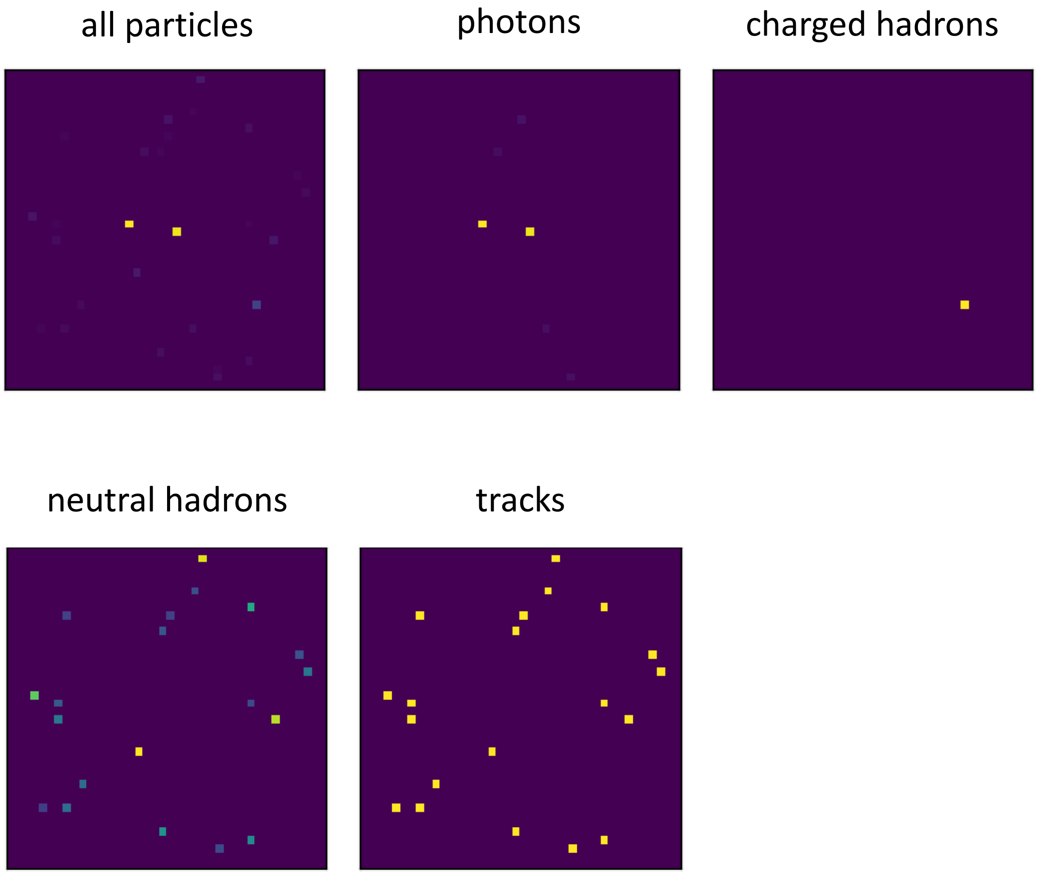
labels [10000]

valid_data.npz

10000 jets

Images [10000, 5, 40, 40]

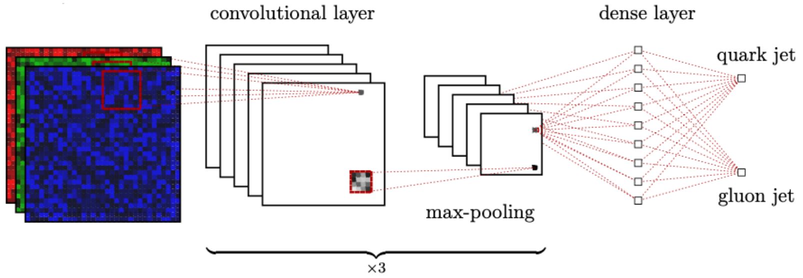
labels [10000]



Tutorial 3 Steps

1 **Preprocess data**

Format
Plot
Normalization
Split into training set and validation set



2 **Build model**

CNN

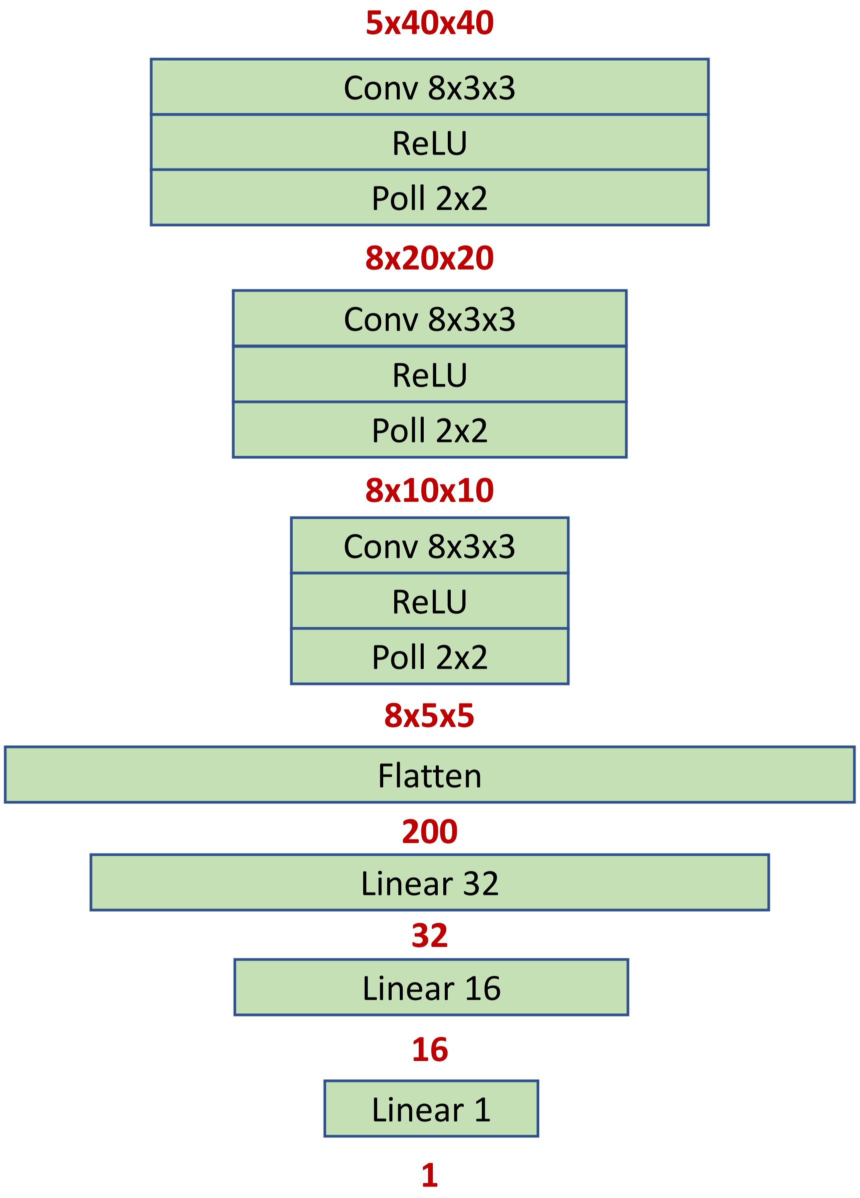
3 **Train and validate iteratively**

4 **Make predictions**

5 **Evaluate model**

Tutorial 2 Steps

Model


5x40x40
Conv 8x3x3
ReLU
Poll 2x2
8x20x20
Conv 8x3x3
ReLU
Poll 2x2
8x10x10
Conv 8x3x3
ReLU
Poll 2x2
8x5x5
Flatten
200
Linear 32
32
Linear 16
16
Linear 1
1