Machine Learning approaches to top-quark tagging

INFIERI Summer School, Wuhan
12-26 May 2019

Lisa Benato (1), Patrick Connor (2), Gregor Kasieczka (1), Dirk Krücker (2), Mareike Meyer(2)
(1) Universität Hamburg; (2) DESY

Introduction

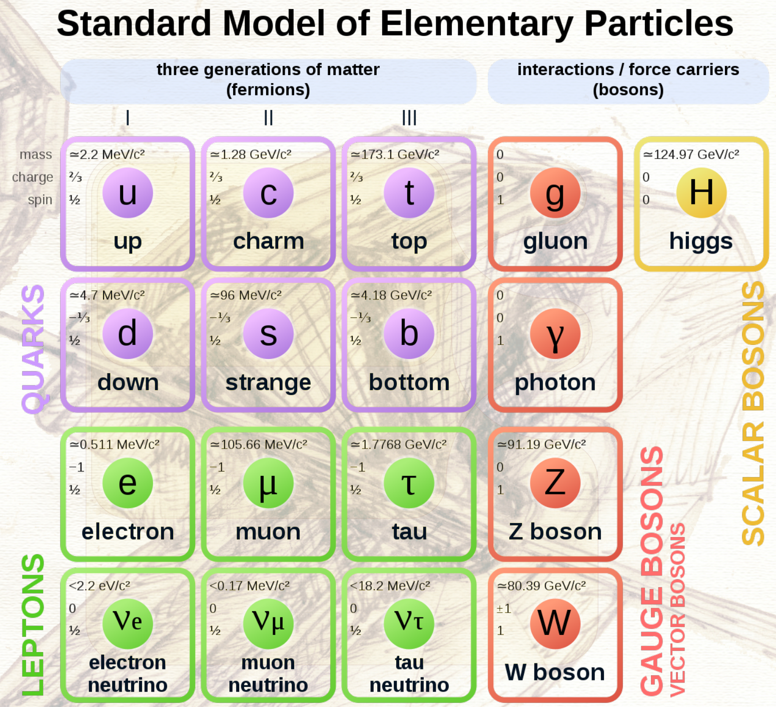
- If you are here, most likely you have attended the first part of this lab…
- ...or, you already have some experience with machine learning (ML)
- We assume you have a basic knowledge of python, and that you know the meaning of *training* and *testing* the performances of a neural network (if not, ask)

- In this lab, we will apply machine learning techniques to solve one high-energy physics problem
- This presentation is just a quick overview: you will find very detailed explanations in the exercise’s notebooks

- We have organized a ML challenge
- Everybody is welcome to participate: rules explained in the next slides
- The winners of the challenge will present their solution at the poster session!

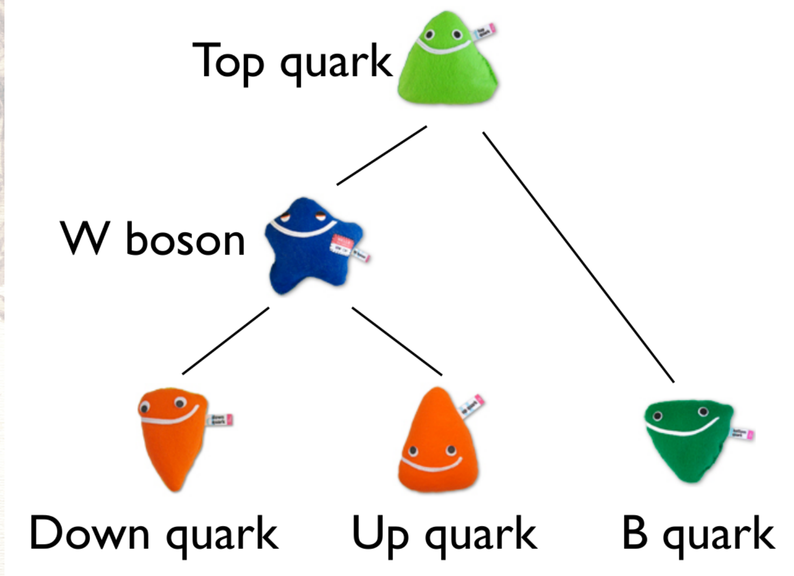
Particle physics in a nutshell

- The Standard Model of particles is our present knowledge of the microscopic world
- It describes the matter constituents (quarks and leptons) and their interactions (mediated by bosons)
- Most recent success: discovery of the Higgs boson in 2012 by ATLAS and CMS experiments at LHC (Geneva)
- But some questions are still open!
- We are trying to answer with precision measurements and searching for “new physics”...



Starting from the top

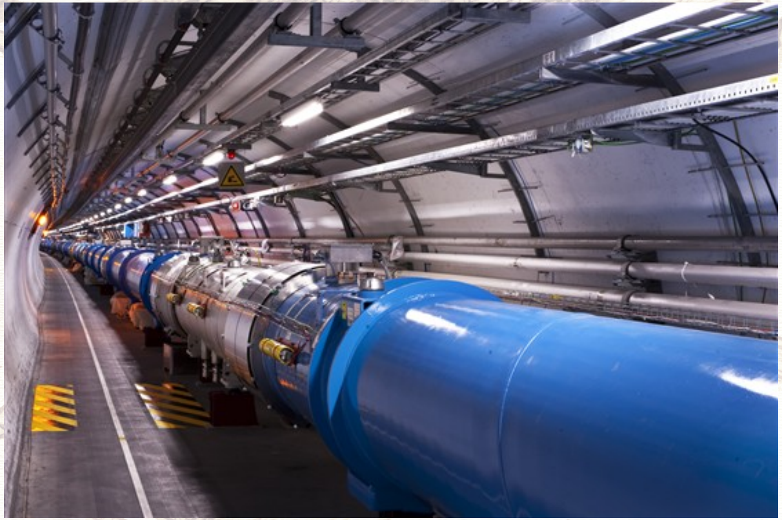
- Top quark is the heaviest known particle (mass of 172.5 GeV)
- Very short lifetime (10^{-25} seconds): we can only see its decay products
- Discovered in 1995 at D0 and CDF experiments at Fermilab (Chicago)
- Key particle to searches for new physics beyond the Standard Model and to precision measurements
- Most challenging (and interesting) top quark decay: "hadronic"
  $t \rightarrow W\,b \rightarrow q\,q'\,b$



In this example, q and q' are a down and an up quark

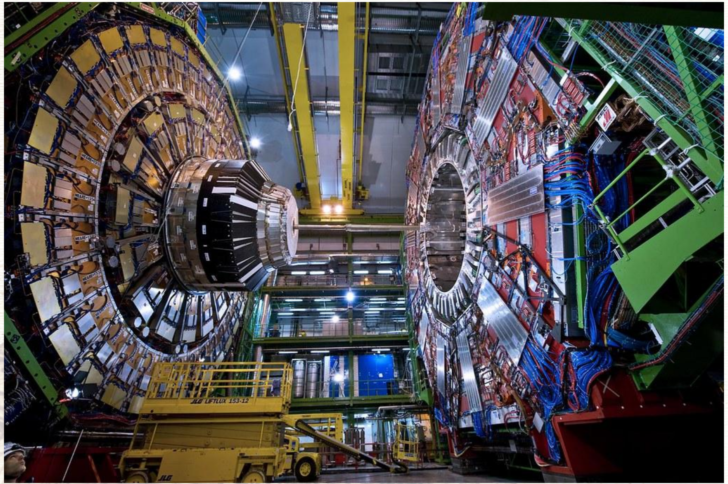
How to find a top quark (I)

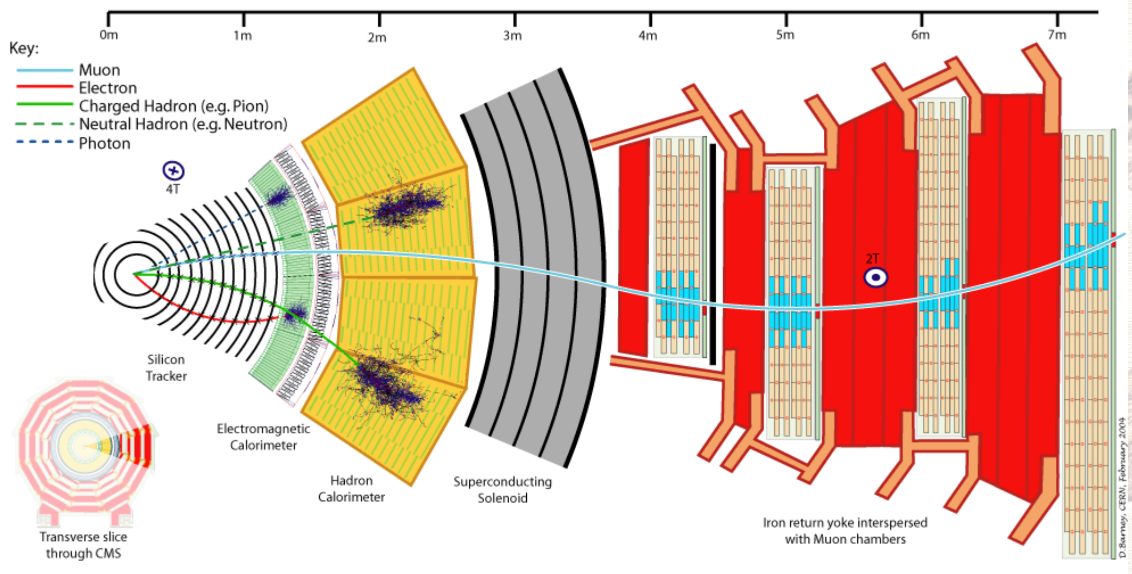
1. Produce it → take an hadron collider, LHC



How to find a top quark (II)

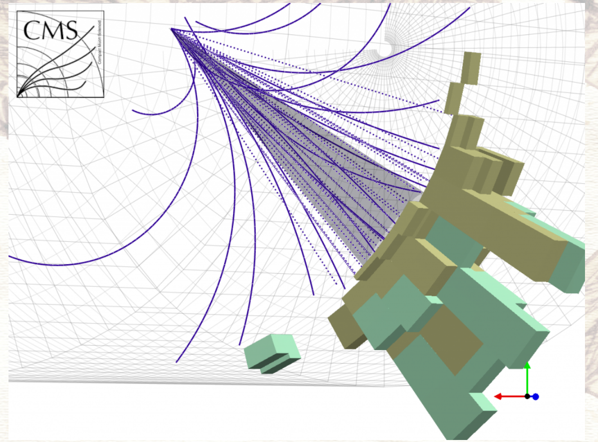
1. Produce it → take an hadron collider, LHC
2. Detect its decay products → take a detector, such as CMS, that reconstructs the energy and position of each particle

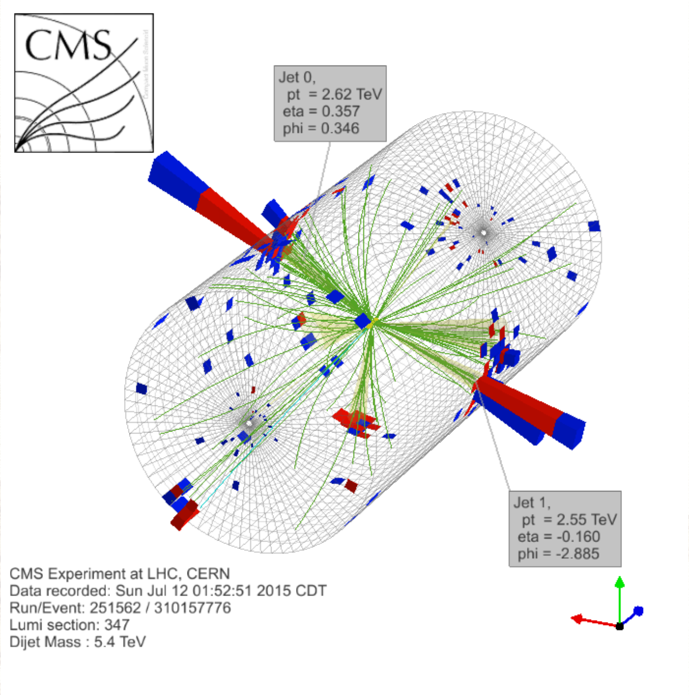




How to find a top quark (III)

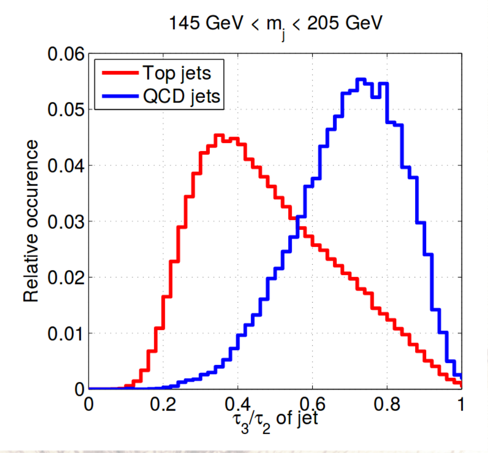
1. Produce it → take an hadron collider, LHC
2. Detect its decay products → take a detector, such as CMS, that reconstructs the energy and position of each particle
3. Combine the reconstructed particles in higher level objects → use dedicated "jet" algorithms



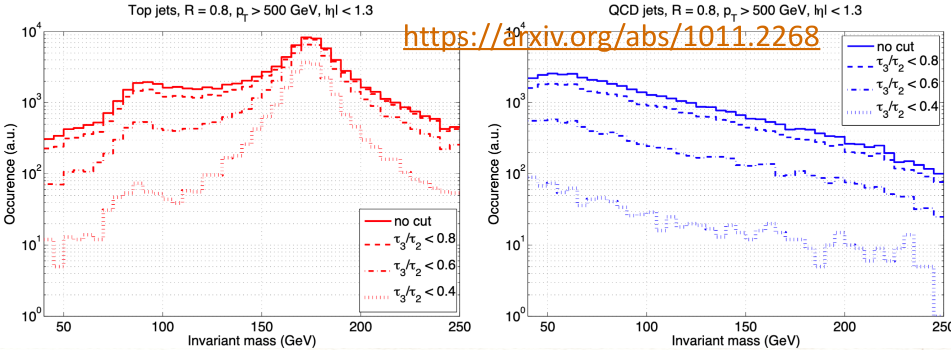


How to find a top quark (IV)

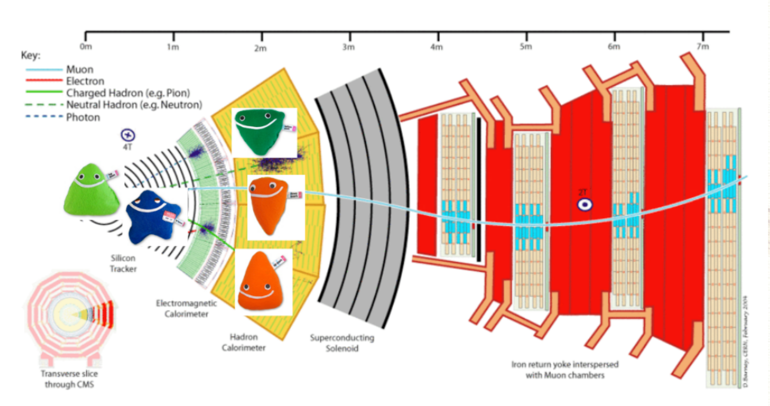
https://arxiv.org/abs/1011.2268



1. Produce it → take an hadron collider, LHC
2. Detect its decay products → take a detector, such as CMS, that reconstructs the energy and position of each particle
3. Combine the reconstructed particles in higher level objects → use dedicated “jet” algorithms
4. Distinguish top decay products from background events → use your physical knowledge to understand the differences

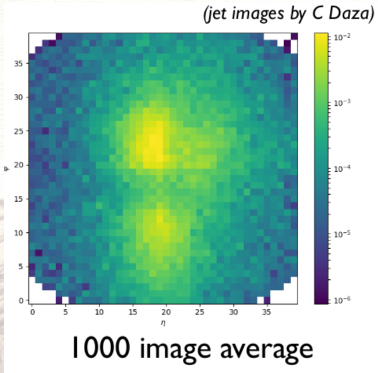


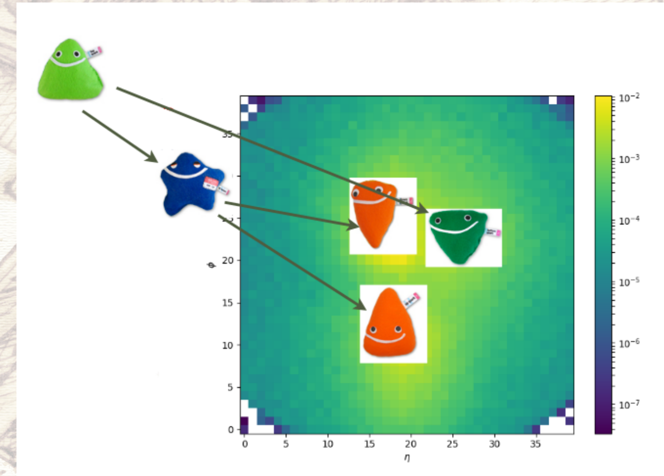
https://arxiv.org/abs/1011.2268



How to find a top quark (V)

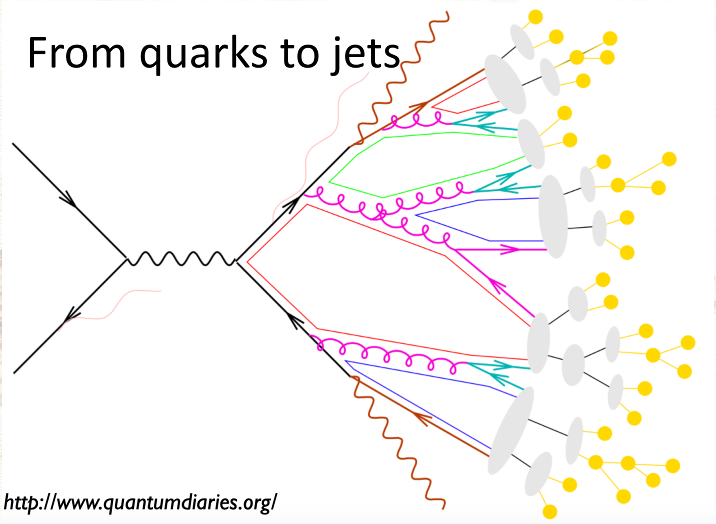
1. Produce it → take an hadron collider, LHC
2. Detect its decay products → take a detector, such as CMS, that reconstructs the energy and position of each particle
3. Combine the reconstructed particles in higher level objects → use dedicated "jet" algorithms
4. Distinguish top decay products from background events → use your physical knowledge to understand the differences
5. Improve results with machine learning taggers!





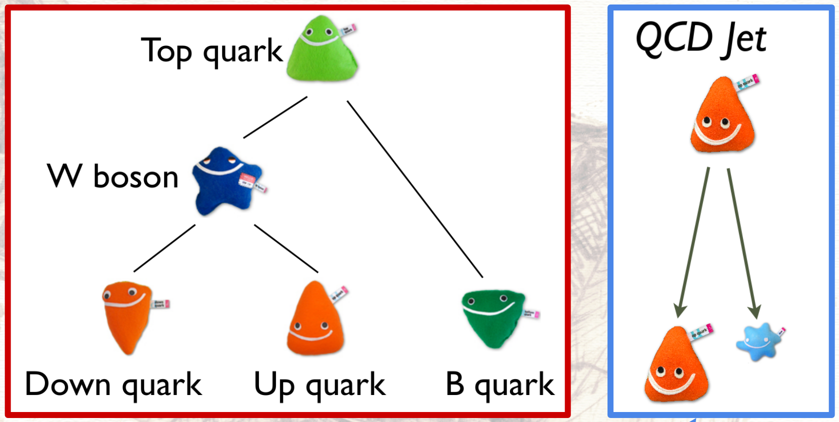
Why is top tagging complicated? (I)

- Due to the nature of strong interaction, quarks do not travel free
- They are forced to be "confined" into hadrons ("combination" of quarks that is neutral under the strong interaction)
- Quarks are not detected as single isolated particles, but as a **jet** of particles
- Jet algorithms are able to cluster together the particles coming from a quark
- Designed such in a way that the momentum of the clustered jet is proportional to the initial energy of the quark



Why is top tagging complicated? (II)

- Producing top quarks is “difficult”
- Top quark production is a relatively “rare” phenomenon (*top quark production has a small cross-section*)
- Other processes initiated by strong interaction (*QCD*) occur way more often
- They produce lighter quarks (up, down, strange, ...)
- They look similar to top quarks and they happen enormously more often
- Fighting against this background is a huge challenge!

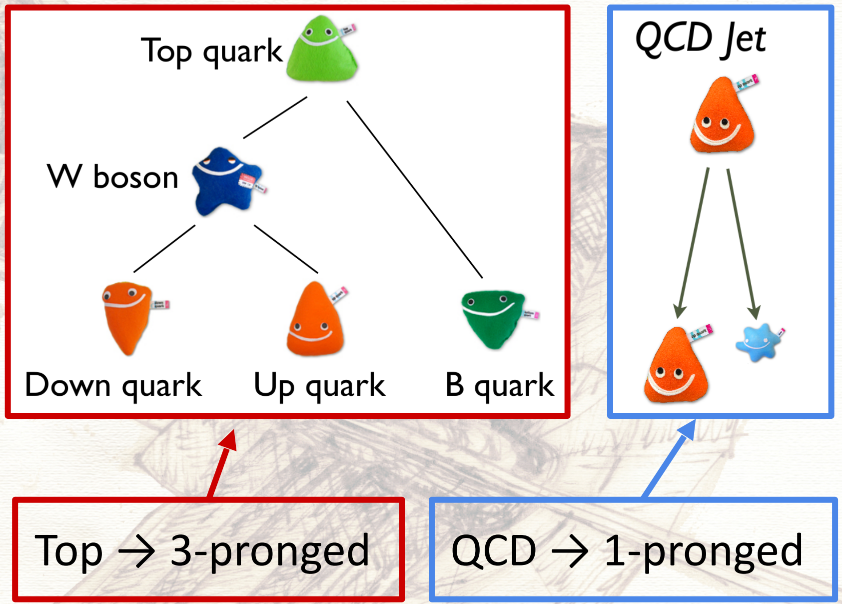


Interesting event
Signal
Rare event

Uninteresting
Background
Frequent event

Physically motivated approach: jet substructure

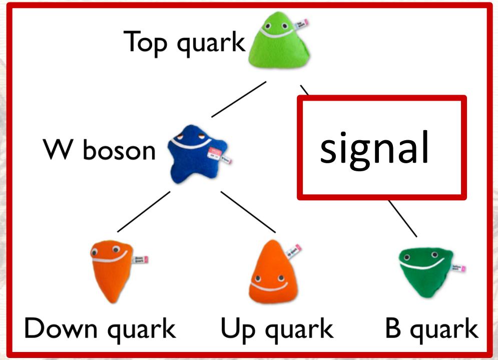
- Very intuitive idea:
  - top quark decays produce 3 quarks
  - strong interaction process involves (usually) 1 quark
- *n-subjettiness*: distinguishes how many "sub-jets" are included in a jet
  - Top → 3-pronged jet
  - QCD → 1-pronged jet
- Jet invariant mass is also a good discriminator
- These properties can be learned by ML approaches!

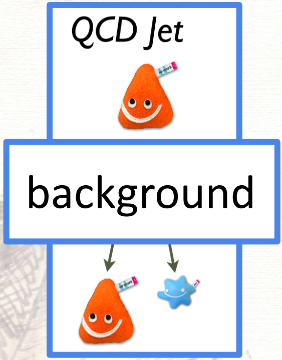


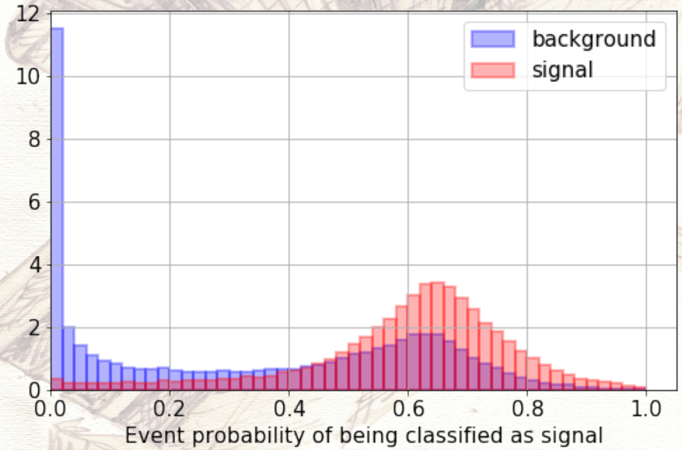
Machine learning formulation

- We must solve a binary classification problem
  - class 0: background (QCD)
  - class 1: signal (top)

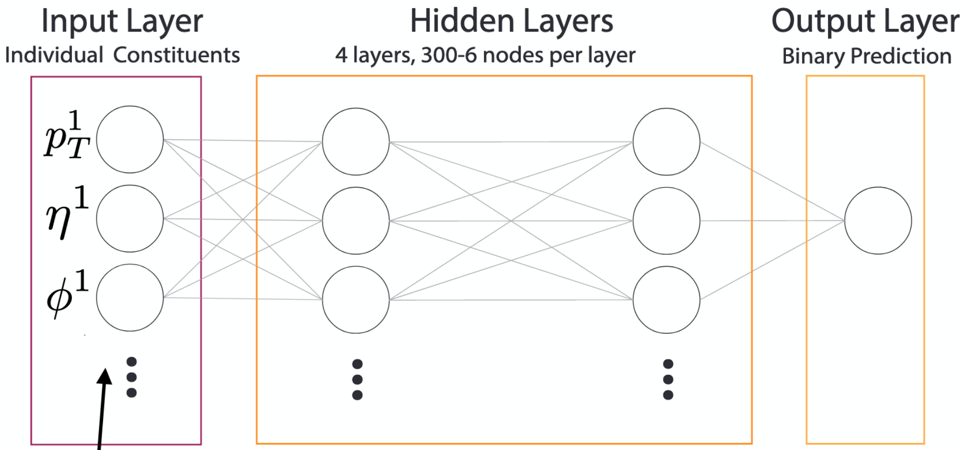
- We can use jet constituents as inputs
- We must build a good architecture:
  - capture the important details
  - not over complicated (*reasonable training times*)
  - able to generalize (*no overfitting*)
  - good performances (*ROC curve*)







Fully Connected Neural Networks



https://arxiv.org/pdf/1704.02124.pdf

Directly input jet constituents

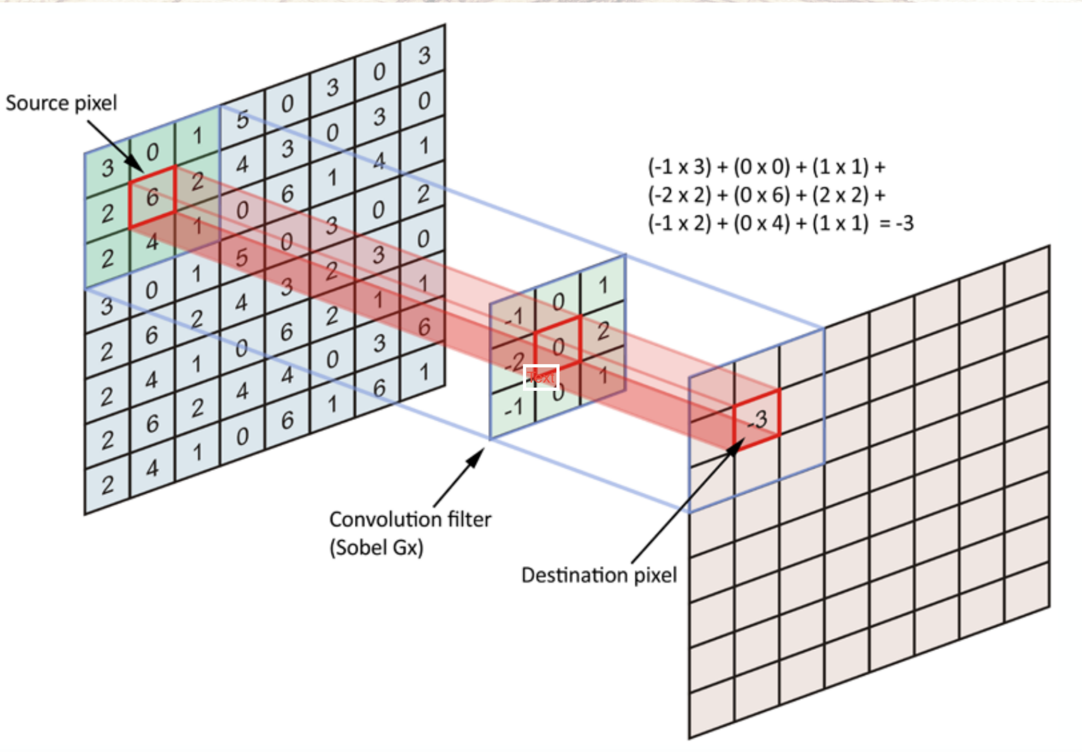
- Very generic structures, that can be applied in many different classification problems
- Excellent as a starting point
- Sometimes they provide (too) many weights
- They can be quite inefficient

TopTagging_1: jet constituents

- You will use the 4-momenta of the particles clustered into jets as input features of your network
- E, p_x, p_y, p_z of 200 jet constituents are stored in pandas DataFrames
- Constituents are sorted by their transverse momentum (the fist constituents is the most energetic)
- A flag (1 for top events, 0 for background) is kept for each jet. It is called "is_signal_new"
- The starting point is a fully connected architecture but you can try something else

- *You will be guided to understand the data content, to evaluate performances and to understand the meaning of a ROC curve*
- *You will find some hints to improve your results*

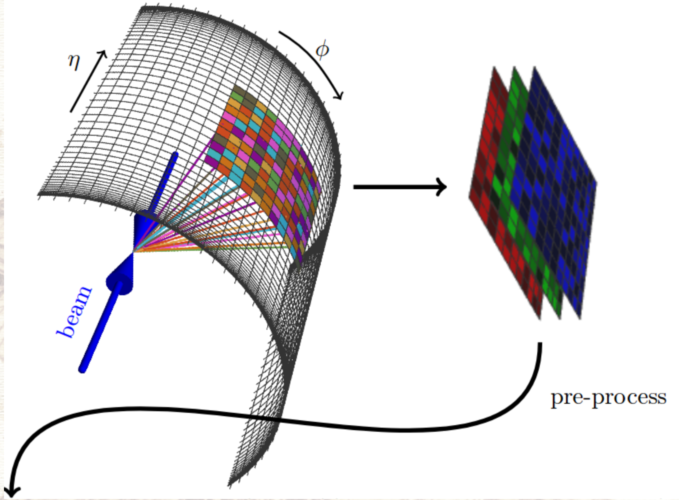
Convolutional Neural Networks



- Used in technology for image recognition
- Basic idea: filters reduce the size of the input image, “summarizing” the important features of a picture
- Network learns the elements of the filters
- Filters operate as matrices multiplications
- Designed to detect edges or particular patterns
- First we need to “transform” jets into images!

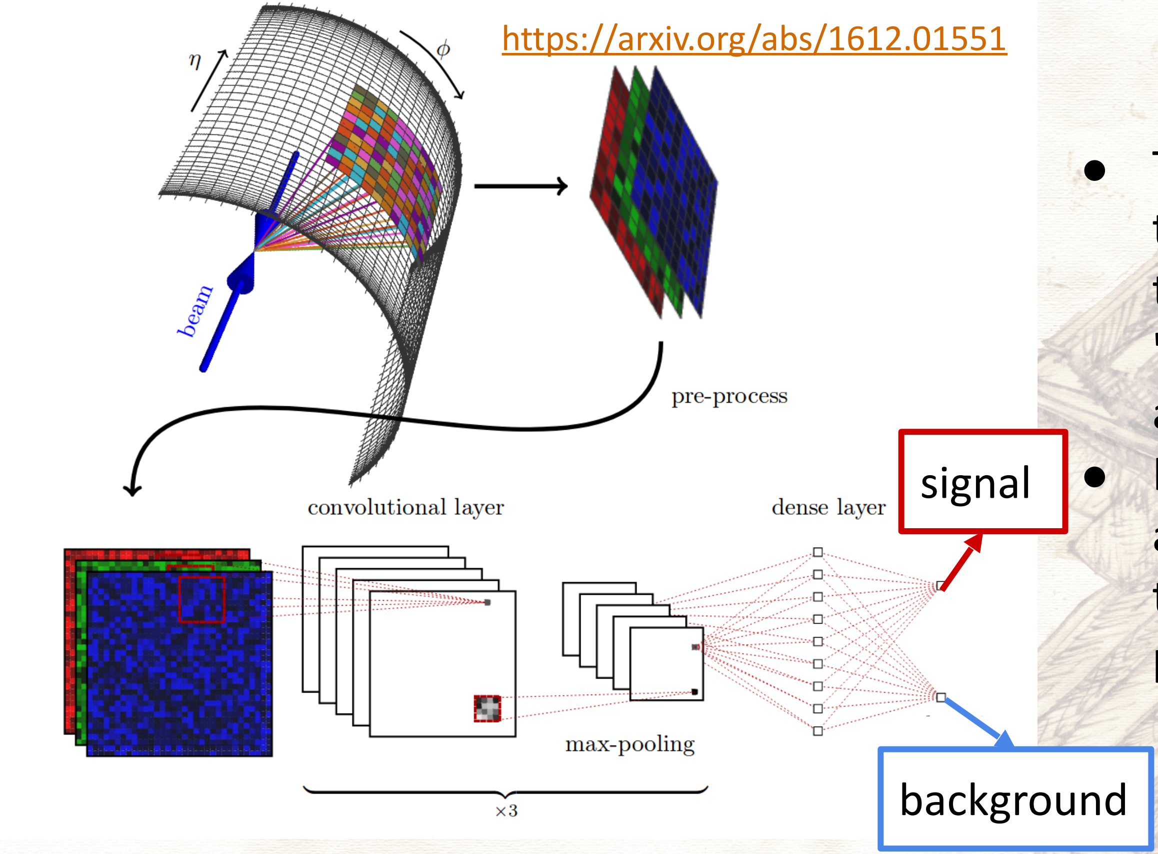
TopTagging_2: jet images (I)

- Shape of CMS detector → a cylinder
- The cylindrical surface can be unrolled along the radial and the longitudinal coordinates
- This surface, that will be a rectangle, can then be divided into "pixels".
- The particle energy deposits can be converted into "colour intensities" within each pixel
- The more dense and the more energetic the particles, the more color density in one particular pixel
- We will work in b&w



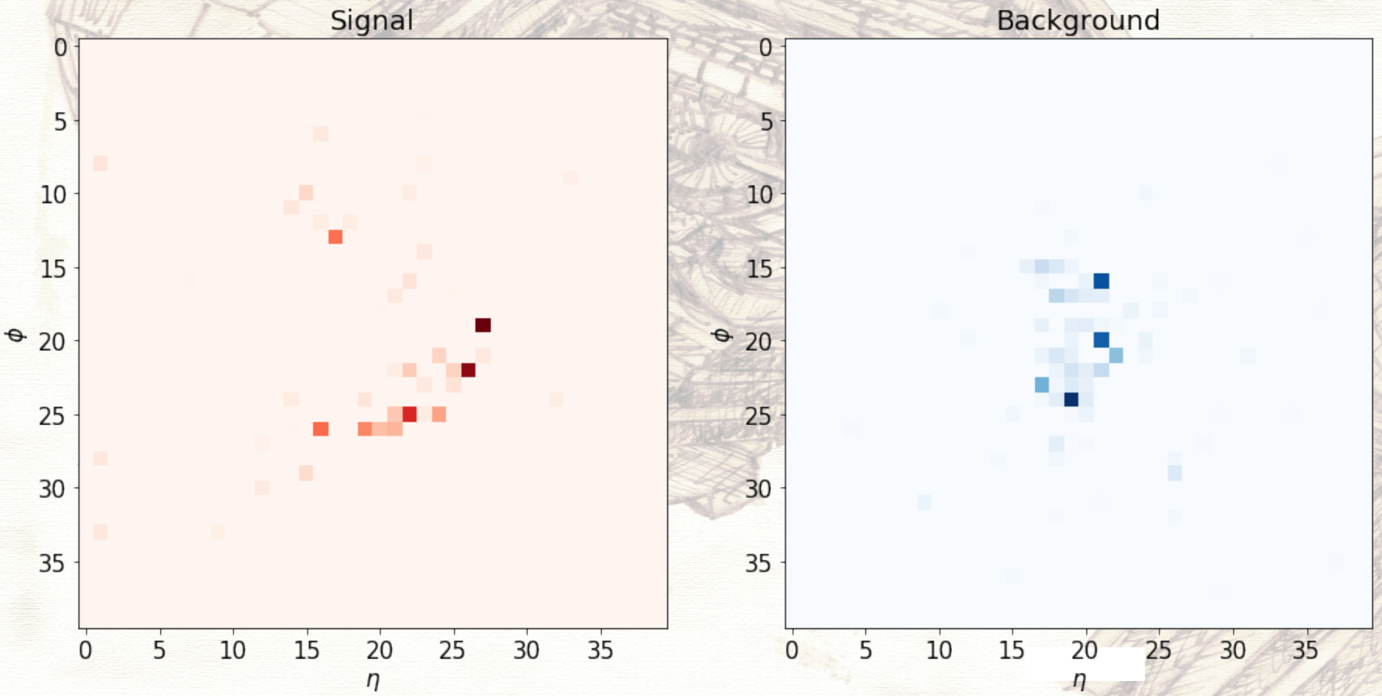
https://arxiv.org/abs/1612.01551

TopTagging_2: jet images (II)



- The energy deposits of the jets constituents are transformed into "intensities" of a 2D black and white image
- Image recognition algorithms can be applied to a high-energy physics problem!

TopTagging_2: jet images (III)



- How does one jet image look like?
- They are rather sparse
- Can you tell which one is signal and which one is background?
- …not easy!

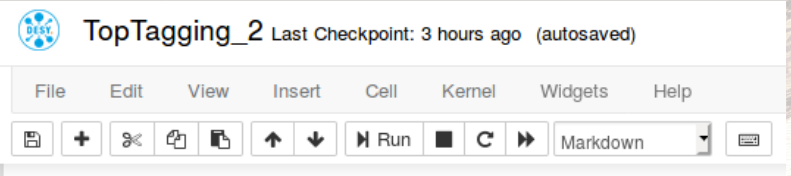
TopTagging_2: jet images (III)

- The 4-momenta of the particles clustered into jets are transformed into 40x40 pixelated images
- The content of these 1600 pixels are stored as columns in a pandas DataFrame
- A flag (1 for top events, 0 for background) is kept for each jet. It is called “is_signal_new”
- This time you will be using convolutional neural networks and more advanced concepts (such as pooling)

- *You will be guided to understand and visualize the jet images, to evaluate performances and to understand the meaning of a ROC curve*
- *You will find some hints to improve your results*

Instructions (I)

- Exercises are provided in jupyter notebooks
- The environment is set into Amazon Web Services (China version – expect differences in EU, USA, Japan, Korea, …)
- We provide a large data sample for training and testing your network
- We will use Keras and Tensorflow machine learning libraries




```
# Define the network
model = keras.models.Sequential()
model.add(keras.layers.Flatten(input_shape=(40,40,1)))
model.add(keras.layers.Dense(2, activation='softmax'))
print(model.summary())
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_1 (Flatten)          (None, 1600)              0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 3202
=================================================================
Total params: 3,202
Trainable params: 3,202
Non-trainable params: 0
_________________________________________________________________
None
```

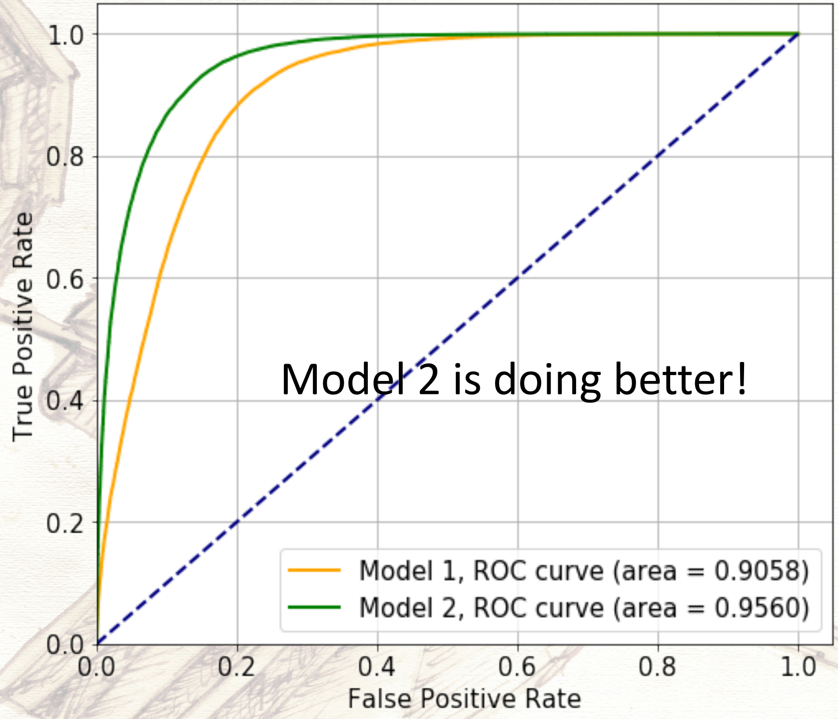
Keras

Instructions (II)

- Save the pem-key (hkwas.pem) you received via mail and take note of the machine name
- On your computer:
```
chmod 400 hkwas.pem
ssh -S tmp -i hkwas.pem ec2-user@AWS_MACHINE_NAME.amazonaws.com -L localhost:1087:localhost:8888
```
- On AWS (Amazon Web Service)
```
cd exercise
jupyter notebook
```
- You will get a link to copy and paste in your browser for accessing the notebook (you might need to modify the localhost number)
- AWS are temporary machines. Everything will disappear at the end of the exercise. scp everything you want to keep to a safe place!
- Windows user? See backup slides

Scoring performances

- Performance measurement for binary classification: receiver operating characteristic curve, or **ROC curve**
- It compares how often the network predicts a signal outcome, when the input is signal (*true positive rate*) vs how often the network predicts a signal outcome, when the input is background (*false positive rate*)
- The higher the area under roc curve (**AUC**), the better the performance of the classifier



Public dataset and top scores

- Data used in these exercises are public and available here: https://goo.gl/XGYju3
- They are currently used to compare different top taggers result → you are playing with a real ML problem!
- If you get an AUC larger than 0.98, please let us know! You deserve a publication!

	AUC	Acc	$1/\epsilon_B$ ($\epsilon_S = 0.3$) single	mean	median	#Param
CNN [16]	0.981	0.930	914±14	995±15	975±18	610k
ResNeXt [30]	0.984	0.936	1122±47	1270±28	1286±31	1.46M
TopoDNN [18]	0.972	0.916	295±5	382± 5	378 ± 8	59k
Multi-body N-subjettiness 6 [24]	0.979	0.922	792±18	798±12	808±13	57k
Multi-body N-subjettiness 8 [24]	0.981	0.929	867±15	918±20	926±18	58k
TreeNiN [43]	0.982	0.933	1025±11	1202±23	1188±24	34k
P-CNN	0.980	0.930	732±24	845±13	834±14	348k
ParticleNet [47]	0.985	0.938	1298±46	1412±45	1393±41	498k
LBN [19]	0.981	0.931	836±17	859±67	966±20	705k
LoLa [22]	0.980	0.929	722±17	768±11	765±11	127k
Energy Flow Polynomials [21]	0.980	0.932	384			1k
Energy Flow Network [23]	0.979	0.927	633±31	729±13	726±11	82k
Particle Flow Network [23]	0.982	0.932	891±18	1063±21	1052±29	82k
GoaT	0.985	0.939	1368±140		1549±208	35k

https://arxiv.org/pdf/1902.09914.pdf

Challenge rules

- You can participate as a single participant or as a team
- The winner is the one scoring the best AUC in the challenge test sample
- In the notebooks, you will find some lines of code for preparing an output zip file, containing your model and the weights you obtained out of your training
- Choose a meaningful name for your result zip file (i.e. your name, or your team name)
- Download the zip file and upload it here: https://desycloud.desy.de/index.php/s/n38qi4eGdgKWLTQ
- You can submit multiple results, paying attention to name them accordingly (add the version number, such as v1, v34, etc.)
- You can use both TopTagging_1 or TopTagging_2 as a starting point (train over constituents or over images)
- We will consider your best result for the final score
- The winner(s) will be asked to present his/her architecture

Deadline for submission: today at 17.00!

Challenge rules

- The most important rules:

Don’t be afraid to ask questions!
Learn as much as you can!
Have fun!

Backup slides

Unix settings

To connect to the machine you need the name of **your** machine and a pem-key `hkaws.pem`.

```
ssh -i hkaws.pem ec2-user@MACHINENAME.amazonaws.com  -L localhost:1087:localhost:8888
```

We will provide them to you *personally* by mail.

Some explaination:

- We connect by **`ssh`** with an identity file (certificate): `hkaws.pem`
- **The `-S tmp` is sometimes necessary on a Mac due to some longish filenames `ssh` creates**
- **`ec2-user`** is the standard user
- **`ec2-18-162-44-11.ap-east-1.compute.amazonaws.com`** is an example for a machine name
- **`-L localhost:1087:localhost:8888`** creates a tunnel that maps a web application from the remote machine to your laptop. The tunnel allows you to connect on your laptop with `http://localhost:1087` to a remote Jupyter notebook.

Once on the machine: `cd wuhan; jupyter notebook`

Windows settings

To connect to the machine you need the name of **your** machine and a pem-key `hkaws.pem`.
We will provide them to you *personally* by mail.

Step 0: install PuTTY
https://www.putty.org/

Step 1: generate PPK key
Change PEM key into PPK with `PuTTYgen`
- Load private key
- Save public key

Step 2: configure SSH
Open `PuTTY`
- *Session*: `ec2-user@MACHINENAME.amazonaws.com`
- *SSH > Auth*: load public PPK key
- *Tunnels*: add Dynamic port 8888
- Click *Open*

Step 3: start notebook
Still in `PuTTY`
- `cd wuhan`
- `jupyter notebook`

Take note of URL

Step 4: Configure your browser

Manual proxy configuration:
- SOCKS-host: `localhost`
- Port: `8888`