University of Ljubljana
Faculty of Mathematics and Physics
Department of Physics

Seminar

End-to-End Classification

for Discovery of New Processes in High-Energy Physics

Author: Elijan Mastnak
Adviser: prof. dr. Borut Paul Kerševan

What is Particle Classification?

Classification

Two high-energy particles collide. Which particles were produced as a result of the collision?

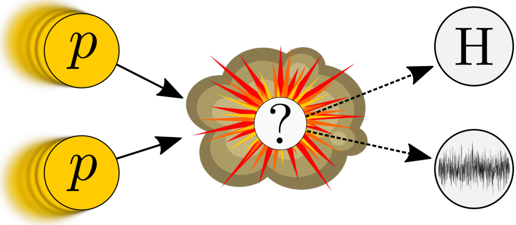
What is Particle Classification?

Classification

Two high-energy particles collide. Which particles were produced as a result of the collision?

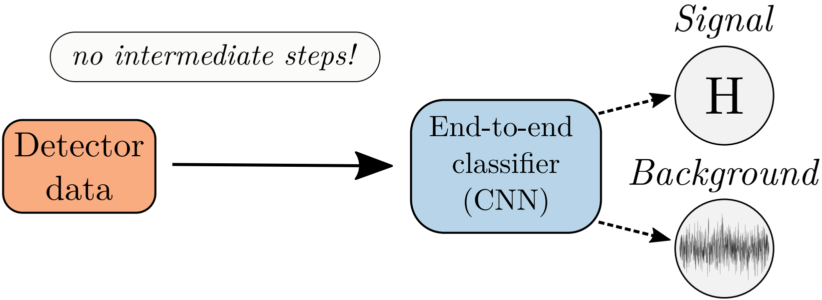
We will consider *binary* Higgs boson classification:

- Higgs boson (*signal*)
- anything else (*background*)



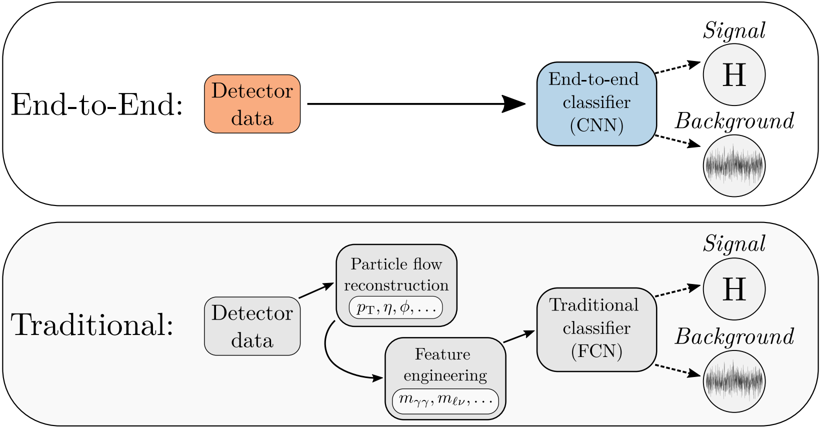
End-to-End Classification

- Directly uses raw detector data



End-to-End Classification

- Directly uses raw detector data
- Eliminates complicated intermediate steps



What is the Detector Data?

The set of measured physical quantities describing the products of a particle collision

- Produced by: Large Hadron Collider (LHC)
- Measured by: Compact Muon Solenoid (CMS)

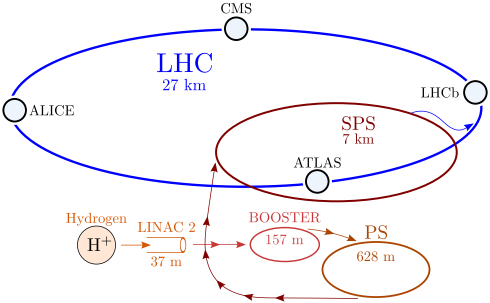
We will explain:

1. proton acceleration and collision at the LHC
2. physical principles of CMS subdetectors
3. how to interpret CMS detector data

Proton Acceleration at the LHC

Sequence:

1. hydrogen ions
2. boosting stages
3. LHC
4. nominal collisions points
5. detectors

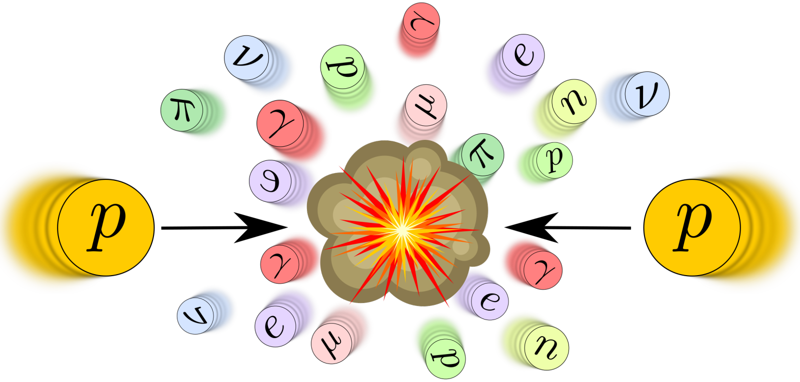


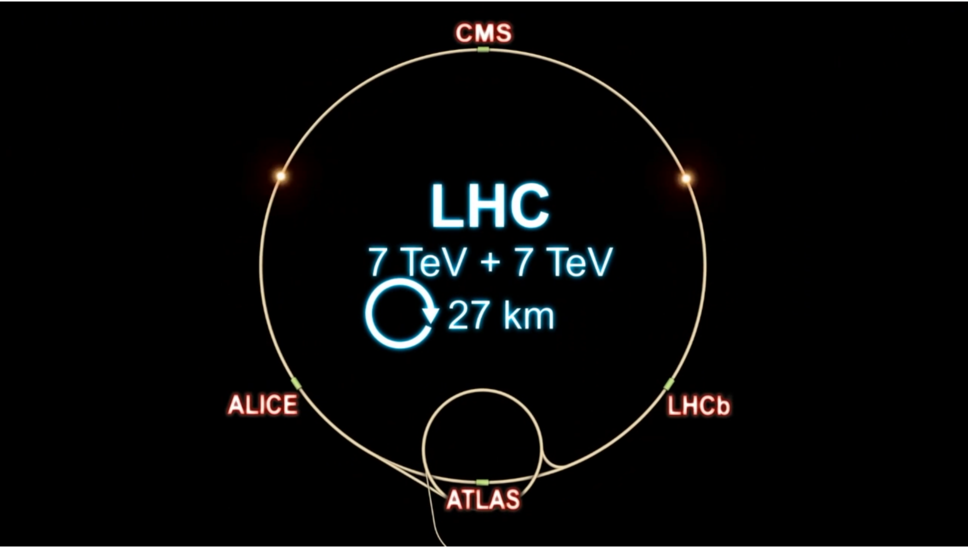
Adapted from [4]

$\sim 7\,\text{TeV}$ proton energy
$\sim 10^{11}$ particles per bunch
$\sim 25\,\text{ns}$ between collisions

Collision

- Two protons (rarely!) collide head-on
- Chain of secondary interactions
- Resulting particles are the *decay signature*

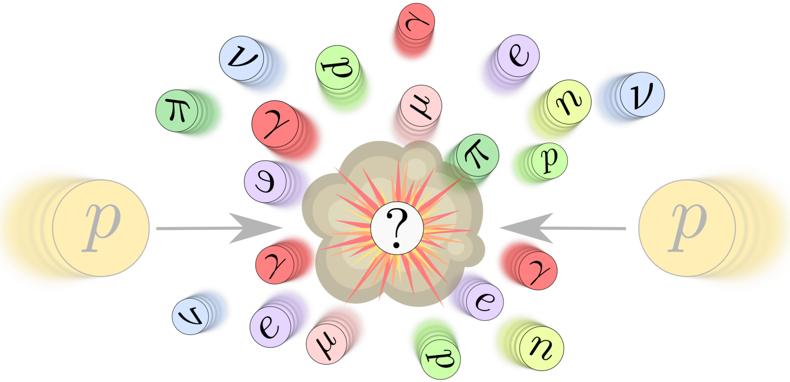




Video: proton-proton collision at the ATLAS detector [3]

An Important Limitation

- Interesting particles decay *rapidly* ($\tau_H \sim 10^{-22}\,$s)
- *We cannot detect a Higgs directly*
- *All we see is the decay signature*



Quantifying a Decay Signature

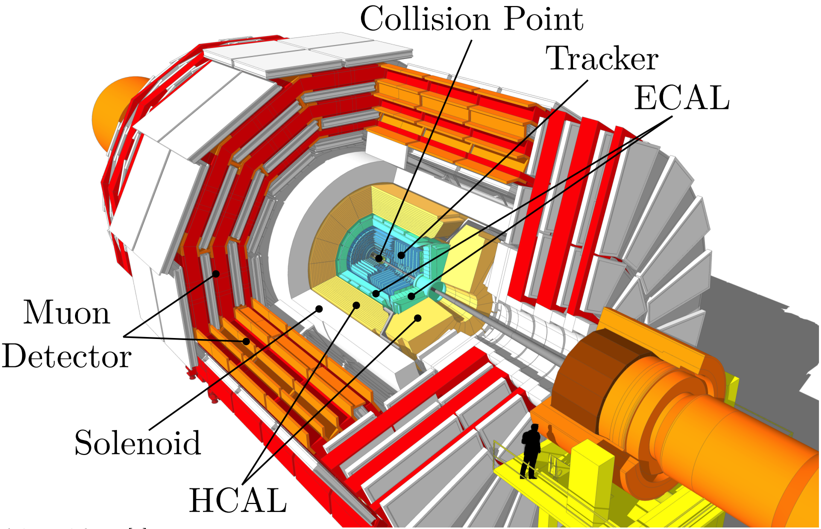
A particle detector measures:

- particle trajectory (using *trackers*)
- particle energy (using *calorimeters*)

With detector data, we can reconstruct:

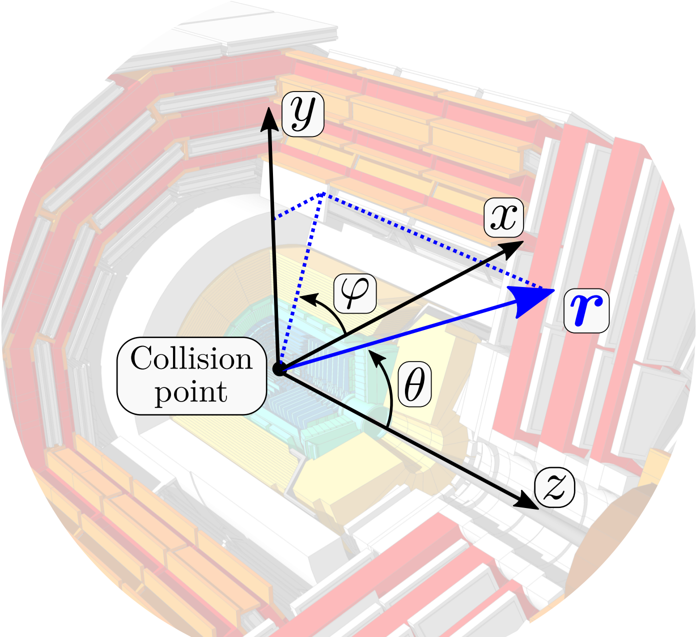
- particle identity
- particle momentum
- production and decay vertices...

The Compact Muon Solenoid



Adapted from [5]

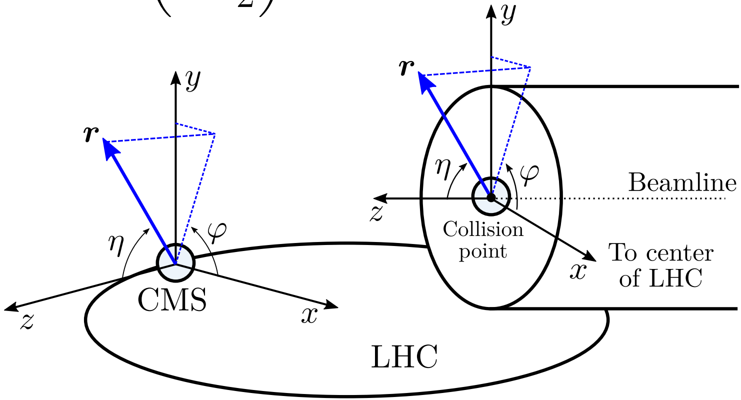
The CMS Coordinate System I



The CMS Coordinate System II

Pseudorapidity η preferred over θ

$$\eta \equiv -\ln\left(\tan\frac{\theta}{2}\right) \implies \theta = 2\arctan e^{-\eta}$$



Tracker: Measuring Trajectory

Working Principle

- Reverse-biased semiconductor
- Charged particle frees electron-hole pair
- Electron-hole pair registered as charge pulse

Tracker: Measuring Trajectory

Working Principle

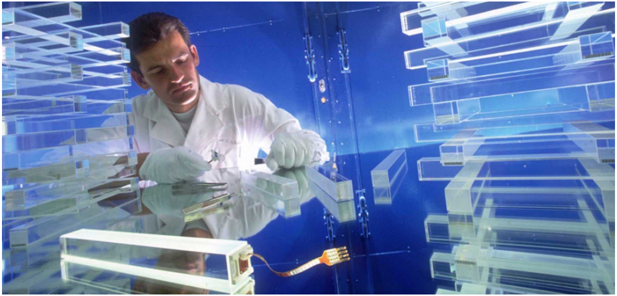
- Reverse-biased semiconductor
- Charged particle frees electron-hole pair
- Electron-hole pair registered as charge pulse

For orientation...

- 13 concentric layers of silicon pixels and strips
- Dimensions $\sim 10\,\mu\text{m}$ to $100\,\mu\text{m}$
- ~ 75 million read-out channels

Electromagnetic Calorimeter (ECAL)

- Measures energy of electromagnetically interacting particles
- Lead tungstate ($PbWO_4$) scintillator crystals
- Dimensions $\sim 2\,\mathrm{cm} \times 2\,\mathrm{cm} \times 20\,\mathrm{cm}$
- $\sim 75\,000$ total scintillator crystals



Source: [2]

ECAL Working Principle

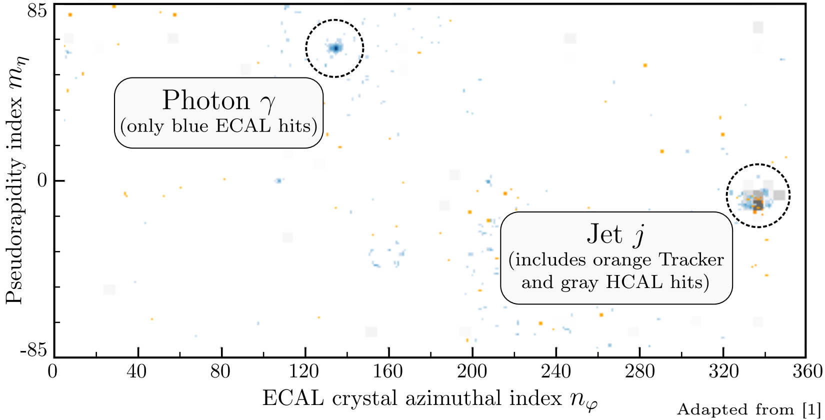
- Incident particle produces *electromagnetic shower*
- Electromagnetic shower excites ($PbWO_4$) scintillator
- Scintillator emits *scintillation photons*
- Photons free *photoelectrons* in reverse-biased semiconducting photodetector
- Photodetector registers photoelectrons as electric signal

$$U_0 \propto N_{e^-} \propto N_\gamma \propto E_{\mathrm{dep}}$$

Hadronic Calorimeter (HCAL)

- Measures energy of hadronic particles
- Brass absorbers and plastic scintillators
- Working principles similar to ECAL

Detector-Data I



Adapted from [1]

- image-like pixel grid
- 2 spatial dimensions (ϕ, η)
- 3 detector channels (Tracker, ECAL, HCAL)

Detector-Data II

There is a direct physical correspondence between pixel values and particle position and energy

$$\text{pixel intensity} \iff \begin{cases} \text{charge in Tracker} \\ \text{energy in ECAL/HCAL} \end{cases}$$

$$\text{pixel position} \iff \text{position of...} \begin{cases} \text{silicon pixel} \\ \text{ECAL crystal} \\ \text{HCAL tile} \end{cases}$$

Classification Options

(a) End-to-end classification: directly use image-based detector data

(b) Kinematic-based classification: first reconstruct kinematic features

(c) High-level classification: first reconstruct kinematic features, then hand-engineer custom *high-level* features

We will call (b) and (c) *traditional classification*.

Traditional Classification

(a) Training

- *Simulate* collision data ($\sim 10^6$ collisions)
- Train neural network with simulated data

Traditional Classification

(a) Training

- *Simulate* collision data ($\sim 10^6$ collisions)
- Train neural network with simulated data

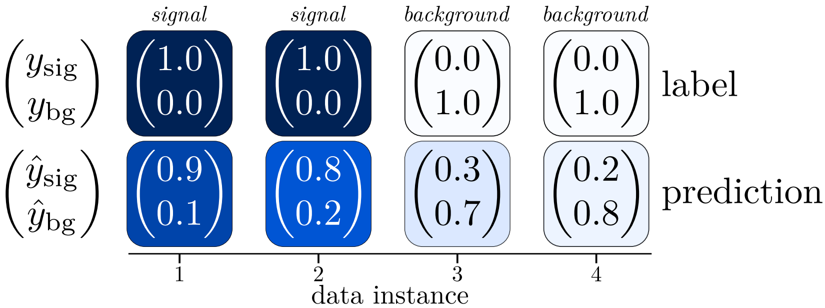
$$\boldsymbol{x}_{\text{feature}} = \begin{pmatrix} p_{\mathrm{T}} \\ \varphi \\ \eta \\ \vdots \end{pmatrix} \longrightarrow \boxed{\begin{array}{c}\text{Fully-Connected} \\ \text{Neural Network}\end{array}} \longrightarrow \hat{\boldsymbol{y}}_{\text{pred}} = \begin{pmatrix} \hat{y}_{\text{sig}} \\ \hat{y}_{\text{bg}} \end{pmatrix}$$

(b) Application

- Reconstruct kinematic quantities describing each LHC collision ($\boldsymbol{x}_{\text{feature}}$)
- Pass quantities into fully-connected network
- Output classification result

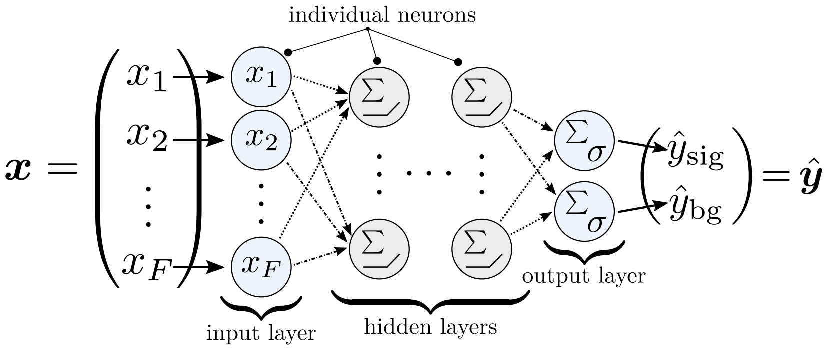
Understanding a Classifier's Output

- True result: $\boldsymbol{y} = \begin{pmatrix} y_{\text{sig}} \\ y_{\text{bg}} \end{pmatrix}$ (known from simulation)
- Prediction: $\hat{\boldsymbol{y}} = \begin{pmatrix} \hat{y}_{\text{sig}} \\ \hat{y}_{\text{bg}} \end{pmatrix}$
- Classes represented by binary $1/0$ values



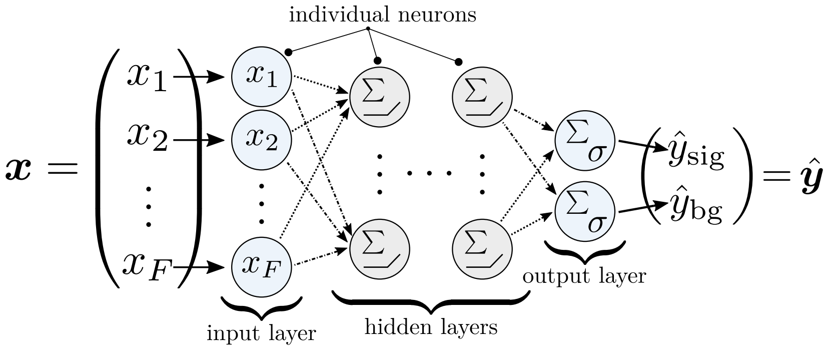
Anatomy of a Fully-Connected Network

- Hierarchy: (i) neuron (ii) layer (iii) network



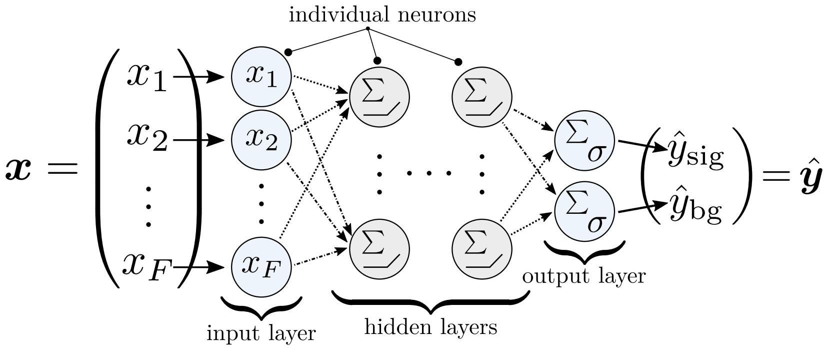
Anatomy of a Fully-Connected Network

- Hierarchy: (i) neuron (ii) layer (iii) network
- Input layer: collision information enters



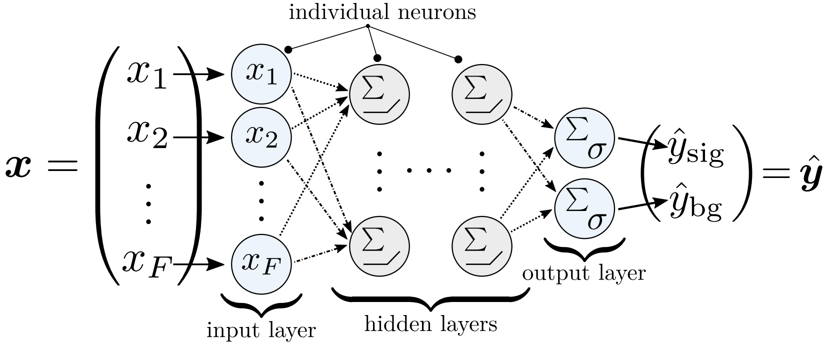
Anatomy of a Fully-Connected Network

- Hierarchy: (i) neuron (ii) layer (iii) network
- Input layer: collision information enters
- Hidden layers: calculations



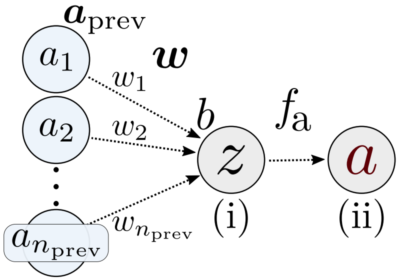
Anatomy of a Fully-Connected Network

- Hierarchy: (i) neuron (ii) layer (iii) network
- Input layer: collision information enters
- Hidden layers: calculations
- Output layer: classification scores



A Single Neuron

- Essentially a multi-variable scalar function
- Input: output of all neurons in previous layer
- Output: a scalar *activation value* $a \in \mathbb{R}$



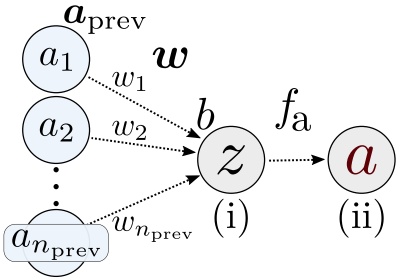
A Single Neuron

- Essentially a multi-variable scalar function
- Input: output of all neurons in previous layer
- Output: a scalar *activation value* $a \in \mathbb{R}$

Two steps:

(i) *linear* weighted sum $z = \boldsymbol{w} \cdot \boldsymbol{a}_{\text{prev}} + b$

(ii) *non-linear* activation $a = f_{\text{a}}(z)$



(i) $z = \boldsymbol{w} \cdot \boldsymbol{a}_{\text{prev}} + b \in \mathbb{R}$

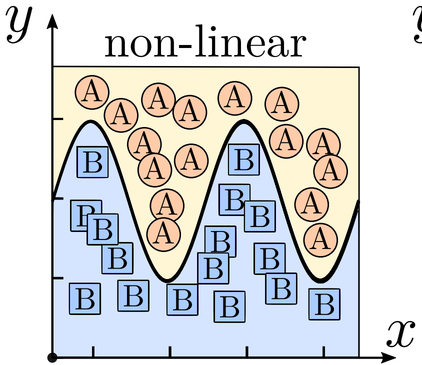
(ii) $a = f_{\text{a}}(z) \in \mathbb{R}$

Activation Function

Non-linear function of pre-activation value

$$a = f_{\mathrm{a}}(z) = f_{\mathrm{a}}\big(\boldsymbol{w}\cdot\boldsymbol{a}_{\mathrm{prev}} + b\big) \in \mathbb{R}$$

Non-linear activation functions allow non-linear decision boundaries!

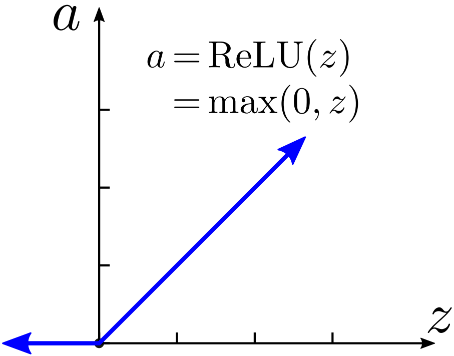


Activation Function

Non-linear function of pre-activation value

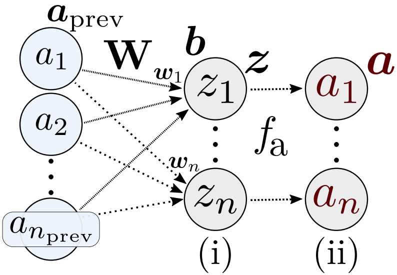
$$a = f_{\mathrm{a}}(z) = f_{\mathrm{a}}\big(\boldsymbol{w}\cdot\boldsymbol{a}_{\mathrm{prev}} + b\big) \in \mathbb{R}$$

- Common functions: ReLU and variants, sigmoid, tanh, etc...
- (Generally) continuously differentiable
- ReLU common in CNNs



A Network Layer

- Weight vectors $\boldsymbol{w} \longrightarrow$ weight matrix $\mathbf{W}$
- Biases $b \longrightarrow$ bias vector $\boldsymbol{b}$
- Activation $a \longrightarrow$ activation vector $\boldsymbol{a}$



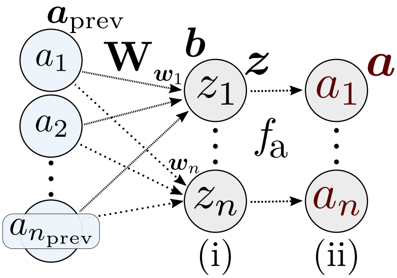
A Network Layer

- Weight vectors $\boldsymbol{w} \longrightarrow$ weight matrix $\mathbf{W}$
- Biases $b \longrightarrow$ bias vector $\boldsymbol{b}$
- Activation $a \longrightarrow$ activation vector $\boldsymbol{a}$

Two steps:

(i) *linear* affine transformation $\boldsymbol{z} = \mathbf{W}^\top \cdot \boldsymbol{a}_{\text{prev}} + \boldsymbol{b}$

(ii) *non-linear* activation $\boldsymbol{a} = f_{\text{a}}(\boldsymbol{z})$



(i) $\boldsymbol{z} = \mathbf{W}^\top \cdot \boldsymbol{a}_{\text{prev}} + \boldsymbol{b} \in \mathbb{R}^n$

(ii) $\boldsymbol{a} = f_{\text{a}}(\boldsymbol{z}) \in \mathbb{R}^n$

Interpreting a FCN

- F features (input) and C classes (output)
- Input: features $\boldsymbol{x} \in \mathbb{R}^F$ and labels $\boldsymbol{y} \in \mathbb{R}^C$
- Output: classification scores $\hat{\boldsymbol{y}} \in \mathbb{R}^C$

A FCN is a vector function $\boldsymbol{h} : \mathbb{R}^F \to \mathbb{R}^C$
parameterized by weights $\mathbf{W}^{(l)}$ *and biases* $\boldsymbol{b}^{(l)}$

Interpreting a FCN

- F features (input) and C classes (output)
- Input: features $\boldsymbol{x} \in \mathbb{R}^F$ and labels $\boldsymbol{y} \in \mathbb{R}^C$
- Output: classification scores $\hat{\boldsymbol{y}} \in \mathbb{R}^C$

A FCN is a vector function $\boldsymbol{h} : \mathbb{R}^F \to \mathbb{R}^C$ parameterized by weights $\mathbf{W}^{(l)}$ and biases $\boldsymbol{b}^{(l)}$

Training Goal

Find optimal values $\mathbf{W}^{(l)}_{\text{opt}}$ and $\boldsymbol{b}^{(l)}_{\text{opt}}$ such that prediction $\hat{\boldsymbol{y}} = \boldsymbol{h}(\boldsymbol{x})$ matches label $\boldsymbol{y}$

Optimization

- *Loss* $L : \mathbb{R}^C \to \mathbb{R}$ quantifies difference between prediction $\hat{\boldsymbol{y}}$ and true result $\boldsymbol{y}$
- Input predictions $\hat{\boldsymbol{y}} \in \mathbb{R}^C$, output loss $L \in \mathbb{R}$
- Example: *categorical cross entropy*

$$L(\hat{\boldsymbol{y}}; \boldsymbol{y}) = -\sum_{c=1}^{C} y_c \ln \hat{y}_c$$

We optimize weights and biases by minimizing loss!

Optimization

- *Loss* $L : \mathbb{R}^C \to \mathbb{R}$ quantifies difference between prediction $\hat{\boldsymbol{y}}$ and true result $\boldsymbol{y}$
- Input predictions $\hat{\boldsymbol{y}} \in \mathbb{R}^C$, output loss $L \in \mathbb{R}$
- Example: *categorical cross entropy*

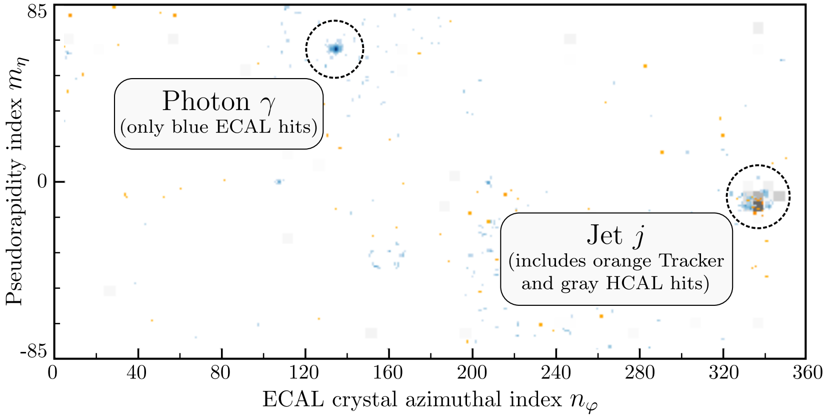
$$L(\hat{\boldsymbol{y}}; \boldsymbol{y}) = -\sum_{c=1}^{C} y_c \ln \hat{y}_c$$

We optimize weights and biases by minimizing loss!

...using numerical methods for multi-dimensional minimization problems adapted to very large parameter spaces and huge datasets.

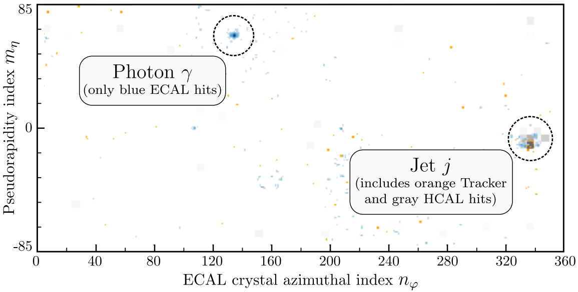
End-to-End Classification

Recall our image-based detector data...

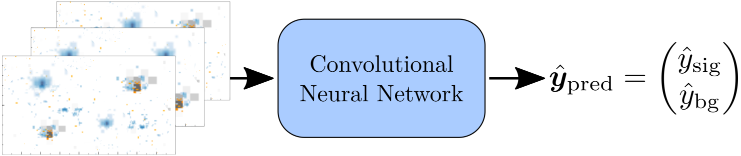


End-to-End Classification

Recall our image-based detector data...



End-to-end classification looks like this:

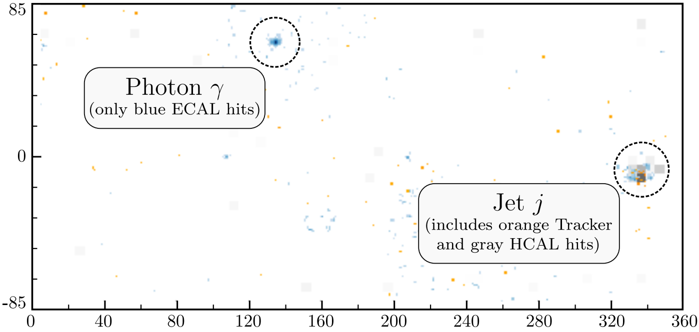


Motivation for Convolutional Networks

Let's examine the input data...

- stored as multi-dimensional arrays
- one *channel axis* for different subdetectors
- two *spatial axes* for coordinates φ and η

Spatial structure stores physical information!



Motivation for Convolutional Networks II

The Goal of Convolutional Networks

Preserve and leverage the information encoded in an input image's *spatial structure*

...in a way that FCNs, limited to flattened, *one-dimensional* vector inputs, cannot.

Motivation for Convolutional Networks II

The Goal of Convolutional Networks

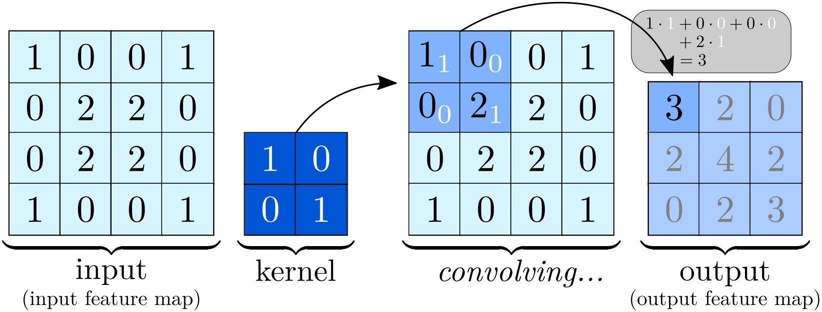
Preserve and leverage the information encoded in an input image's *spatial structure*

...in a way that FCNs, limited to flattened, *one-dimensional* vector inputs, cannot.

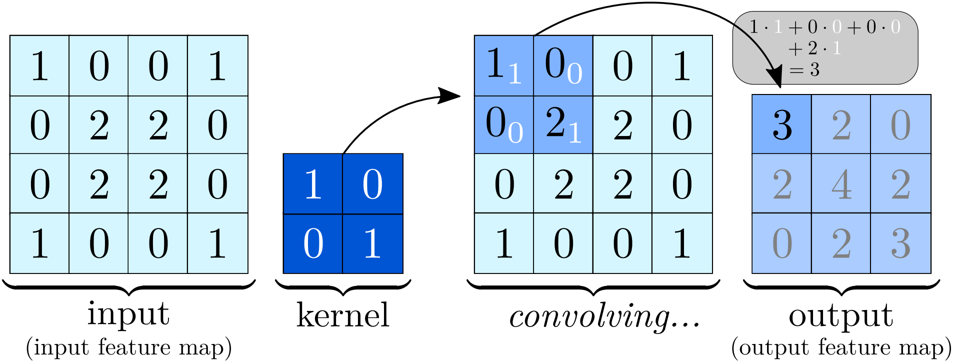
We need a *space-preserving* way for CNNs to interact with input images!

Discrete Convolution

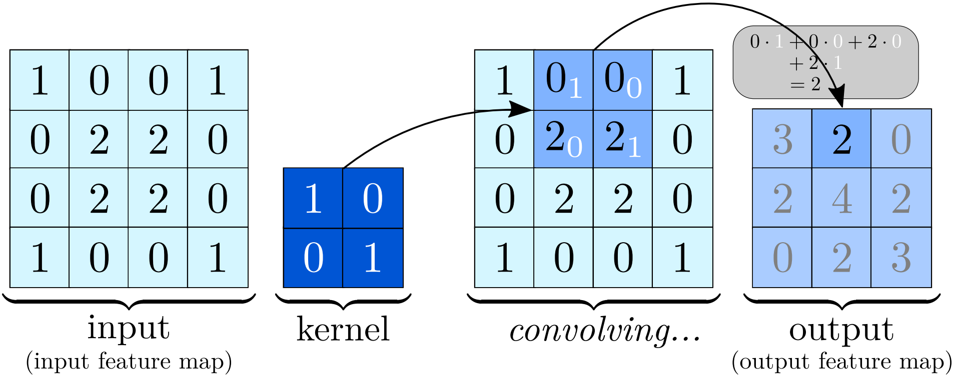
- Intuitively: "scan" 2D image with 2D "filter"
- Mathematically: *convolve* image with convolutional kernel
- Kernel has weights and bias (like FCN neuron)
- Parameters detect distinguishing features



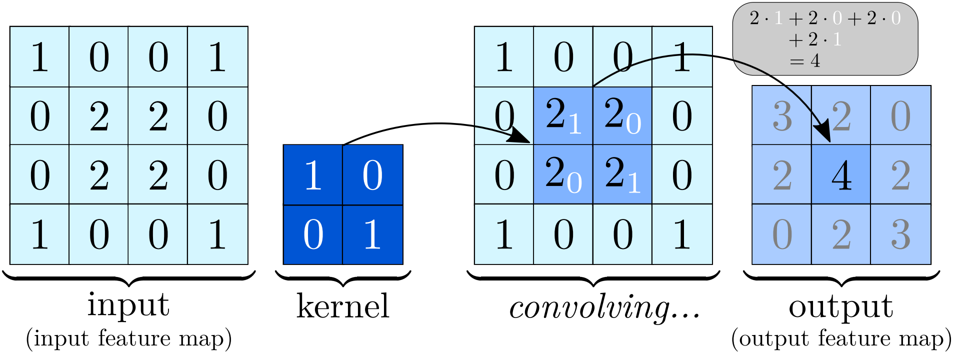
Discrete Convolution Examples



Discrete Convolution Examples



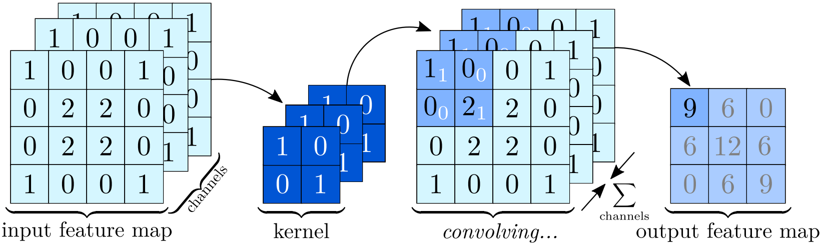
Discrete Convolution Examples



Generalizations...

Multi-channel images

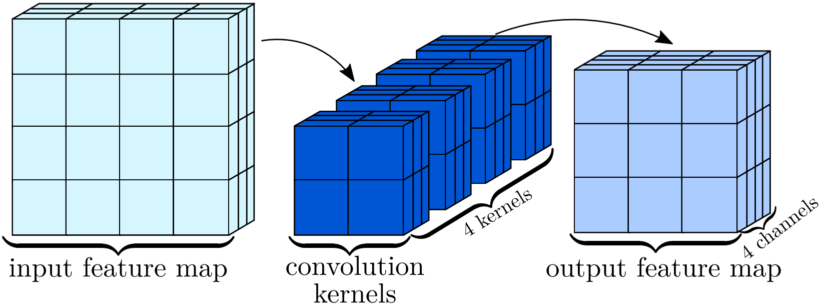
- Input images (3D) have multiple channels...
- So use a multi-channel (3D) kernel!
- Sum across channel axis to get 2D output



Generalizations...

Multiple kernels

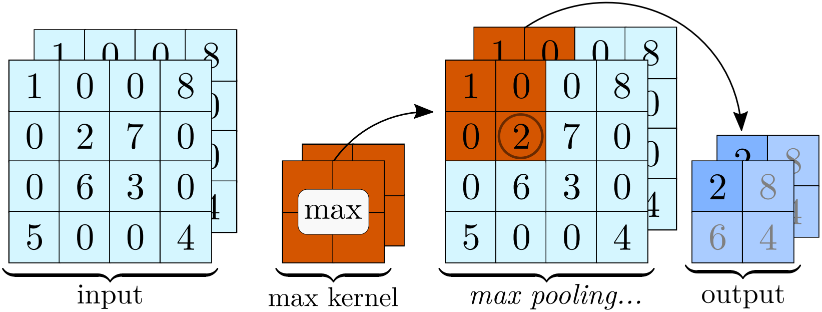
- Like multiple neurons in an FCN
- Each kernel captures a specific feature
  (edges, curves contrasting colors, shapes...)
- Output feature map is then 3D



(Max) Pooling

Goals:

- spatially downsample input
- introduce invariance to local translations
- preserve channel dimension

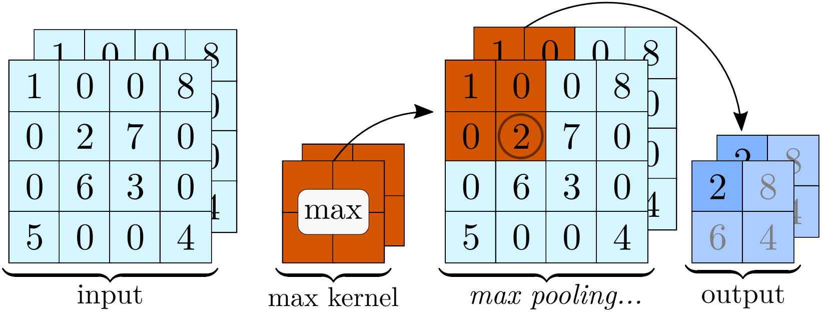


(Max) Pooling

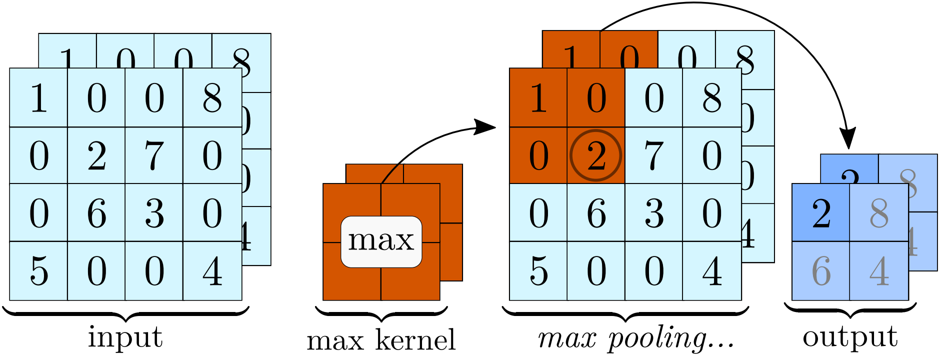
Goals:

- spatially downsample input
- introduce invariance to local translations
- preserve channel dimension

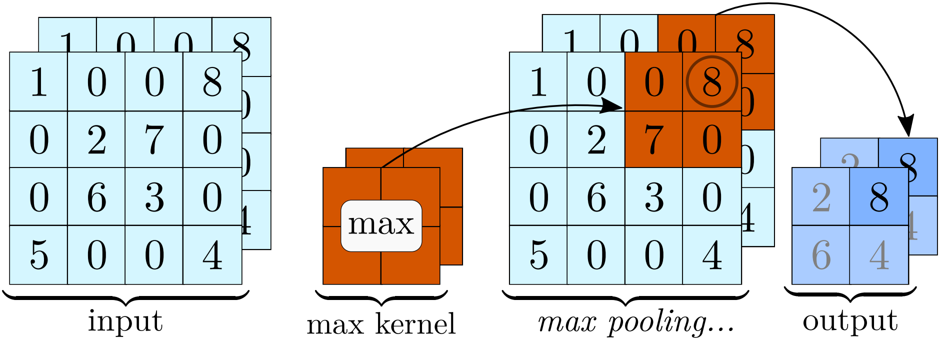
Operation: A *pooling kernel* outputs maximum pixel value at each kernel position in input



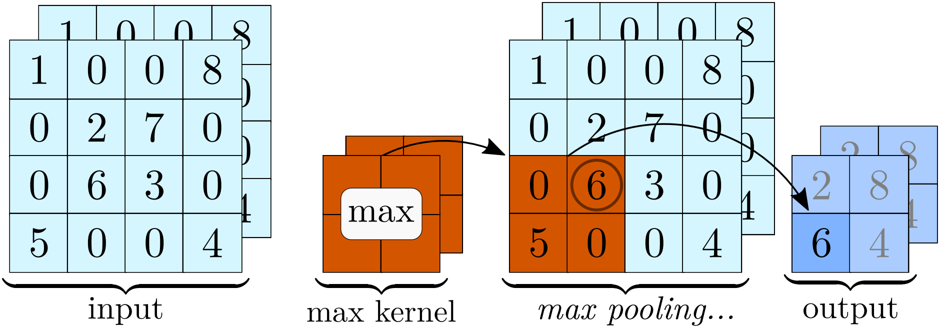
Max Pooling Examples



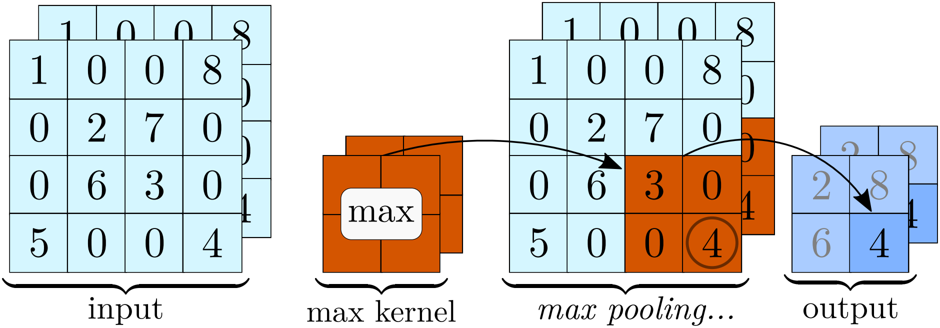
Max Pooling Examples



Max Pooling Examples



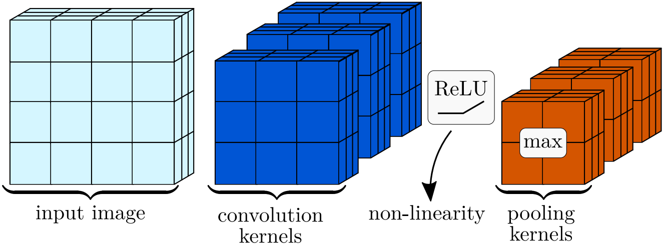
Max Pooling Examples



CNN Architecture

Typical convolutional layer sequence:

(a) convolution
(b) non-linearity (e.g. ReLU)
(c) pooling



CNN Architecture

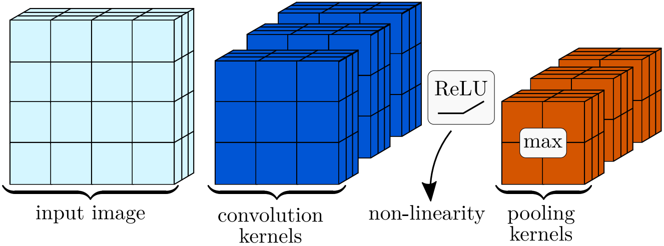
Typical convolutional layer sequence:

(a) convolution

(b) non-linearity (e.g. ReLU)

(c) pooling

Repeat... (not shown)



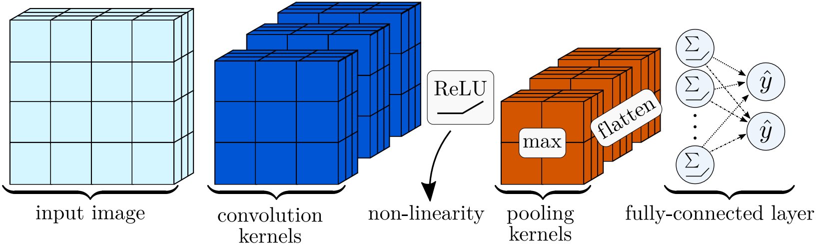
CNN Architecture

Typical convolutional layer sequence:

(a) convolution
(b) non-linearity (e.g. ReLU)
(c) pooling

Repeat... (not shown)

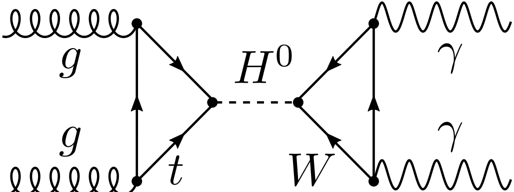
Flatten; use fully-connected layer for output



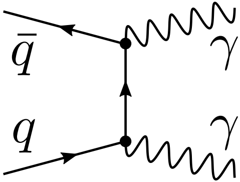
A Concrete Case Study

Andrews et al. *End-to-End Physics Event Classification with CMS Open Data*. 2020. [1]

- Higgs boson classification with CMS data
- Signal: $gg \to H^0 \to \gamma\gamma$
- Background 1: $q\bar{q} \to \gamma\gamma$
- Background 2: $q\bar{q} \to \gamma j$



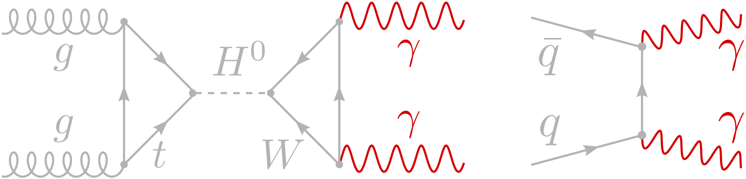
(signal)



(bg. 1)

Why the Processes Are Interesting

(a) irreducible backgrounds



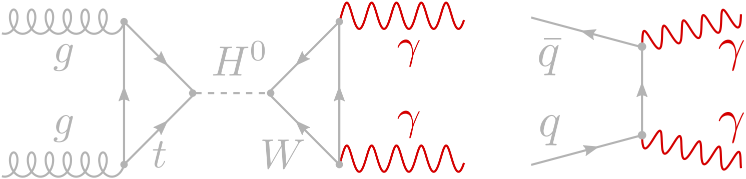
---

for reference...

sig: $gg \to H^0 \to \gamma\gamma$ bg 1: $q\bar{q} \to \gamma\gamma$ bg 2: $q\bar{q} \to \gamma j$

Why the Processes Are Interesting

(a) irreducible backgrounds



(b) unresolved decay products

$$\gamma j \approx \gamma\gamma \implies \text{bg 1} \approx \text{bg 2} \approx \text{sig}$$

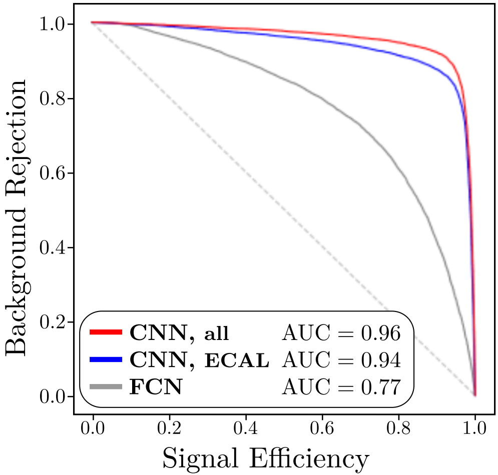
for reference...

sig: $gg \to H^0 \to \gamma\gamma$ bg 1: $q\bar{q} \to \gamma\gamma$ bg 2: $q\bar{q} \to \gamma j$

Example: Photon-Jet Classification

Task: classify $gg \to H \to \gamma\gamma$ and $q\bar{q} \to \gamma j$
Challenge: unresolved decay products

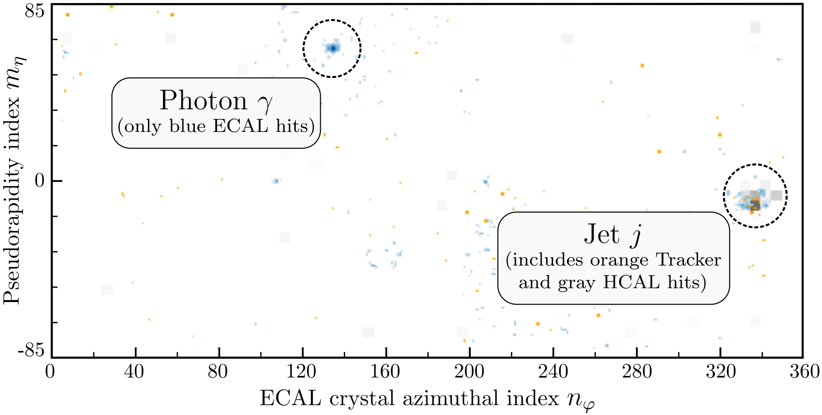
- Comparison: CNN vs. FCN
- CNN performs much better!



ROC curve adapted from [1]

Interpretation

Recall the input image...



Interpretation

What a CNN sees

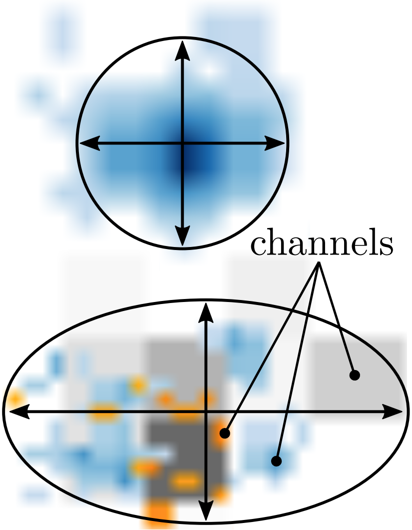
What a FCN sees

$$p_{\mathrm{T}} \approx 55\,\mathrm{GeV}$$
$$\varphi \approx 136^\circ$$
$$\eta \approx 1.10 \quad (\theta \approx 37^\circ)$$

$$p_{\mathrm{T}} \approx 65\,\mathrm{GeV}$$
$$\varphi \approx 335^\circ$$
$$\eta \approx -0.14 \quad (\theta \approx 98^\circ)$$

Interpretation

What a CNN sees



What a FCN sees

$p_\mathrm{T} \approx 55\,\mathrm{GeV}$

$\varphi \approx 136^\circ$

$\eta \approx 1.10 \quad (\theta \approx 37^\circ)$

$p_\mathrm{T} \approx 65\,\mathrm{GeV}$

$\varphi \approx 335^\circ$

$\eta \approx -0.14 \quad (\theta \approx 98^\circ)$

Takeaways and Conclusion

CNNs can distinguish shower distribution patterns even when kinematic quantities are identical.

Promising aspects of end-to-end classification

- Preserve maximum available information
- Learn from spatial distribution
- Flexible and general classification framework

Takeaways and Conclusion

CNNs can distinguish shower distribution patterns even when kinematic quantities are identical.

Promising aspects of end-to-end classification

- Preserve maximum available information
- Learn from spatial distribution
- Flexible and general classification framework

Thank you!

References

[1] M. Andrews, M. Paulini, S. Gleyzer, and B. Poczos, *End-to-End Physics Event Classification with CMS Open Data: Applying Image-Based Deep Learning to Detector Data for the Direct Classification of Collision Events at the LHC*, Computing and Software for Big Science **4** (2020), ISSN: 2510-2044, URL: http://dx.doi.org/10.1007/s41781-020-00038-8.

[2] CMS Collaboration, *The Crystal Tower*, 2014, URL: https://cds.cern.ch/record/1998528.

[3] ATLAS Experiment, *Proton Collision Event with Boosters and LHC*, 2011, URL: https://www.youtube.com/watch?v=NhXMXiXOWAA.

[4] Julie Haffner, *The CERN accelerator complex*, General Photo, 2013, URL: https://cds.cern.ch/record/1621894.

[5] Tai Sakuma, *Cutaway diagrams of CMS detector*, 2019, URL: https://cds.cern.ch/record/2665537.