Event Classification with Deep Learning

Cristian A. Marocico

Center for Information Technology,
University of Groningen, The Netherlands

Friday, September 25th 2020

Contents

Contents

Contents

Contents

A Very Brief Introduction to Deep Learning

A Very Brief Introduction to Deep Learning

- A set of data inputs and labels

$$(x_i, y_i)$$

A Very Brief Introduction to Deep Learning

- A set of data inputs and labels

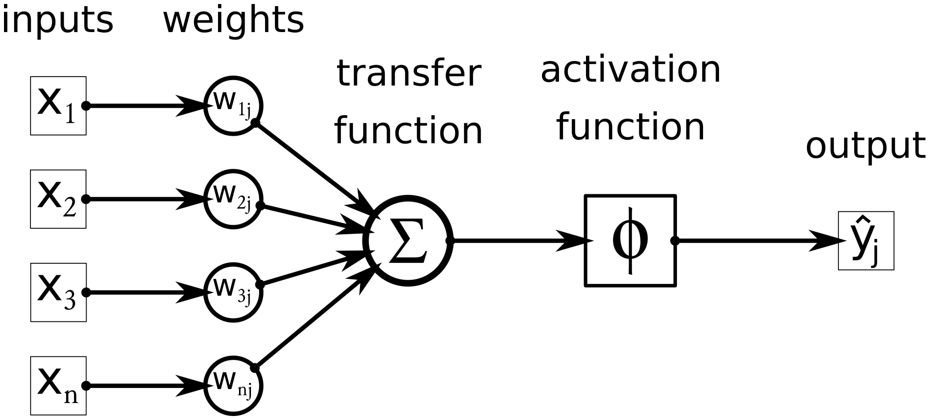
$$(x_i, y_i)$$

- A mapping

$$f(x_i) = y_i$$

A Very Brief Introduction to Deep Learning

A Very Brief Introduction to Deep Learning



A Very Brief Introduction to Deep Learning

A Very Brief Introduction to Deep Learning

- y_j – true labels

A Very Brief Introduction to Deep Learning

- y_j – true labels
- $\hat{y}_j$ – predicted labels

A Very Brief Introduction to Deep Learning

- y_j – true labels
- $\hat{y}_j$ – predicted labels
- $\hat{y}_j = \phi(w_{ij}x_i + b_j)$

A Very Brief Introduction to Deep Learning

- y_j – true labels
- $\hat{y}_j$ – predicted labels
- $\hat{y}_j = \phi(w_{ij}x_i + b_j)$
- Define a loss function (e.g. cross-entropy loss):

$$L(W) = -\sum_j \left[y_j \log(\hat{y}_j) - (1 - y_j)\log(1 - \hat{y}_j)\right]$$

A Very Brief Introduction to Deep Learning

- y_j – true labels
- $\hat{y}_j$ – predicted labels
- $\hat{y}_j = \phi(w_{ij}x_i + b_j)$
- Define a loss function (e.g. cross-entropy loss):

$$L(W) = -\sum_j \left[y_j \log(\hat{y}_j) - (1 - y_j)\log(1 - \hat{y}_j)\right]$$

- Minimize loss function with respect to the weights W :

$$\frac{\partial L}{\partial W} = 0$$

A Very Brief Introduction to Deep Learning

- y_j – true labels
- $\hat{y}_j$ – predicted labels
- $\hat{y}_j = \phi(w_{ij}x_i + b_j)$
- Define a loss function (e.g. cross-entropy loss):

$$L(W) = -\sum_j \left[y_j \log(\hat{y}_j) - (1 - y_j) \log(1 - \hat{y}_j)\right]$$

- Minimize loss function with respect to the weights W :

$$\frac{\partial L}{\partial W} = 0$$

- High-dimensional optimization problem, gradient descent, etc.

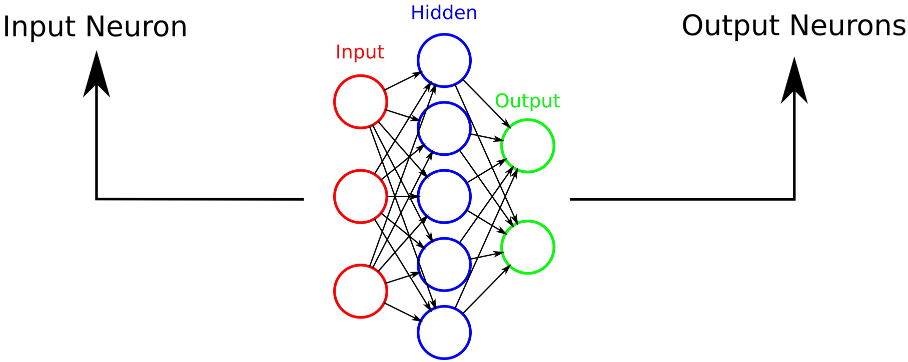
A Very Brief Introduction to Deep Learning

A Very Brief Introduction to Deep Learning

Artificial Neural Network

A Very Brief Introduction to Deep Learning

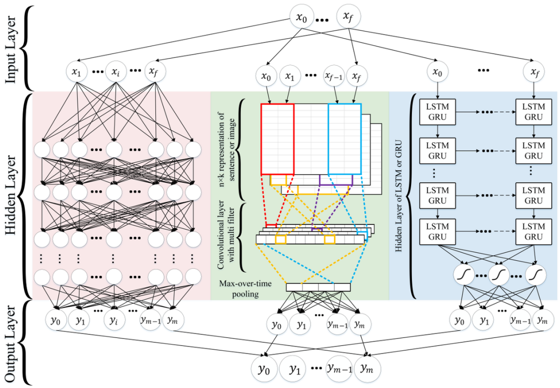
Artificial Neural Network



A Very Brief Introduction to Deep Learning

Deep Neural Networks

Deep Neural Networks



In Praise of Interdisciplinarity

In Praise of Interdisciplinarity

- I am here because of a collaboration between physicists from KVI@UG ...

In Praise of Interdisciplinarity

- I am here because of a collaboration between physicists from KVI@UG ...
- ...and data scientists from CIT@UG

In Praise of Interdisciplinarity

- I am here because of a collaboration between physicists from KVI@UG ...
- ...and data scientists from CIT@UG
- I am neither, and a bit of both

In Praise of Interdisciplinarity

- I am here because of a collaboration between physicists from KVI@UG ...
- ...and data scientists from CIT@UG
- I am neither, and a bit of both
- CIT organizes annual calls for project proposals from researchers across the University

In Praise of Interdisciplinarity

- I am here because of a collaboration between physicists from KVI@UG ...
- ...and data scientists from CIT@UG
- I am neither, and a bit of both
- CIT organizes annual calls for project proposals from researchers across the University
- Selected proposals receive up to 450 hours of Data Science support from the CIT

In Praise of Interdisciplinarity

- I am here because of a collaboration between physicists from KVI@UG ...
- ...and data scientists from CIT@UG
- I am neither, and a bit of both
- CIT organizes annual calls for project proposals from researchers across the University
- Selected proposals receive up to 450 hours of Data Science support from the CIT
- Winning projects encompass Philosophy of Science, Economics, Ancient History, Astronomy, Particle Physics and others

In Praise of Interdisciplinarity

- I am here because of a collaboration between physicists from KVI@UG ...
- ...and data scientists from CIT@UG
- I am neither, and a bit of both
- CIT organizes annual calls for project proposals from researchers across the University
- Selected proposals receive up to 450 hours of Data Science support from the CIT
- Winning projects encompass Philosophy of Science, Economics, Ancient History, Astronomy, Particle Physics and others
- Several of these projects have led to joint publications

In Praise of Interdisciplinarity

- I am here because of a collaboration between physicists from KVI@UG ...
- ...and data scientists from CIT@UG
- I am neither, and a bit of both
- CIT organizes annual calls for project proposals from researchers across the University
- Selected proposals receive up to 450 hours of Data Science support from the CIT
- Winning projects encompass Philosophy of Science, Economics, Ancient History, Astronomy, Particle Physics and others
- Several of these projects have led to joint publications
- It's a lot of fun!

In Praise of Interdisciplinarity

- I am here because of a collaboration between physicists from KVI@UG ...
- ...and data scientists from CIT@UG
- I am neither, and a bit of both
- CIT organizes annual calls for project proposals from researchers across the University
- Selected proposals receive up to 450 hours of Data Science support from the CIT
- Winning projects encompass Philosophy of Science, Economics, Ancient History, Astronomy, Particle Physics and others
- Several of these projects have led to joint publications
- It's a lot of fun!
- If your institution doesn't do this, I encourage you to encourage them to start

What This Is

What This Is

What you can expect from this presentation:

What This Is

What you can expect from this presentation:

- Not the latest and greatest research results

What This Is

What you can expect from this presentation:

- Not the latest and greatest research results
- An overview of a Deep Learning approach to Particle Physics, using as an example the joint CIT + KVI project

What This Is

What you can expect from this presentation:

- Not the latest and greatest research results
- An overview of a Deep Learning approach to Particle Physics, using as an example the joint CIT + KVI project
- Some insights, hopefully

Event Data

Event Data

The data that we have comes from a simulation of the following reaction:

$$e^+ + e^- \rightarrow \Psi\text{'}(3686) \rightarrow \pi^+ + \pi^- + J/\Psi(3097) \rightarrow \begin{cases} \pi^+ + \pi^- + e^+ + e^- \\ \pi^+ + \pi^- + \mu^+ + \mu^- \end{cases}$$

or the simpler one:

$$e^+ + e^- \rightarrow \pi^+ + \pi^- + \mu^+ + \mu^-$$

Event Data

Event Data

The output file produced by the simulations looks like this:

```python
data = open('data/pipijpsi_ee.dat')
for line in data.readlines()[:100]:
    print(line)
```

Event Data

```
Event number:    0
Number of digis: 277
T0:              656

MC truth information:
Pi+: px=0.242978   py=-0.029811  pz=-0.234886 xi=0.021781 yi=-0.154217 zi=0.625080
     xf=87.543005  yf=50.414698  zf=-104.047484
Pi-: px=-0.069618  py=-0.011896  pz=0.152243  xi=0.021781 yi=-0.154217 zi=0.625080
     xf=-24.346692 yf=41.429941  zf=135.987877
e+:  px=1.153744   py=0.669807   pz=-0.660627 xi=0.021781 yi=-0.154217 zi=0.625080
     xf=77.586081  yf=56.726887  zf=-48.333559
e-:  px=-1.286557  py=-0.628100  pz=0.741661  xi=0.021781 yi=-0.154217 zi=0.625080
     xf=-95.105332 yf=-35.066791 zf=53.326682

Detector information:
layer=0,  wire=21, stereo=1, XE=-76.671368,  YE=-18.407167, XW=-74.990806,  YW=24.365990,  RT=500.156250,  RC=2097, TRK=1
layer=2,  wire=24, stereo=1, XE=-102.679683, YE=-6.729982,  XW=-95.067204,  YW=39.378125,  RT=516.468750,  RC=534,  TRK=1
layer=3,  wire=29, stereo=1, XE=-114.525332, YE=-12.903906, XW=-110.745721, YW=31.905293,  RT=1139.062500, RC=506,  TRK=1
layer=3,  wire=28, stereo=1, XE=-115.250000, YE=0.000000,   XW=-106.477116, YW=44.104266,  RT=560.343750,  RC=639,  TRK=1
layer=4,  wire=30, stereo=1, XE=-125.823252, YE=18.664116,  XW=-121.722811, YW=-36.924211, RT=485.625000,  RC=835,  TRK=1
layer=5,  wire=35, stereo=1, XE=-138.222014, YE=12.092859,  XW=-132.328227, YW=-41.722930, RT=1276.593750, RC=671,  TRK=1
layer=5,  wire=34, stereo=1, XE=-136.642076, YE=24.093685,  XW=-135.461071, YW=-30.030996, RT=582.937500,  RC=830,  TRK=1
layer=6,  wire=38, stereo=1, XE=-149.059575, YE=17.642363,  XW=-145.952725, YW=-35.040149, RT=556.031250,  RC=752,  TRK=1
layer=7,  wire=38, stereo=1, XE=-159.610436, YE=25.279810,  XW=-158.494901, YW=-31.526596, RT=453.562500,  RC=551,  TRK=1
layer=8,  wire=35, stereo=0, XE=-193.598870, YE=36.448287,  XW=-193.598870, YW=36.448287,  RT=538.781250,  RC=915,  TRK=1
layer=9,  wire=35, stereo=0, XE=-207.515266, YE=48.025142,  XW=-207.515266, YW=48.025142,  RT=505.500000,  RC=634,  TRK=1
layer=10, wire=39, stereo=0, XE=-219.249655, YE=67.991112,  XW=-219.249655, YW=67.991112,  RT=593.718750,  RC=497,  TRK=1
layer=11, wire=40, stereo=0, XE=-236.978178, YE=64.312875,  XW=-236.978178, YW=64.312875,  RT=1335.656250, RC=561,  TRK=1
layer=11, wire=39, stereo=0, XE=-231.786356, YE=81.054842,  XW=-231.786356, YW=81.054842,  RT=614.343750,  RC=487,  TRK=1
```

Event Data

Event Data

The relevant input consists of:

Event Data

The relevant input consists of:

- `XE(XW)` $=$ the x-position of a hit

Event Data

The relevant input consists of:

- `XE(XW)` $=$ the x-position of a hit
- `YE(YW)` $=$ the y-position of a hit

Event Data

The relevant input consists of:

- `XE(XW)` = the *x*-position of a hit
- `YE(YW)` = the *y*-position of a hit
- `RC` = the energy of a hit

Event Data

Event Data

Possible “labels”:

Event Data

Possible "labels":

- `event type` = which type of event does the data describe

Event Data

Possible "labels":

- `event type` = which type of event does the data describe
- `TRK` = the "track" to which the hit belongs: $\pi^{\pm}, e^{\pm}(\mu^{\pm})$

Event Data

Possible "labels":

- `event type` = which type of event does the data describe
- `TRK` = the "track" to which the hit belongs: $\pi^{\pm}, e^{\pm}(\mu^{\pm})$
- `px, py, pz` = the momenta of each "track"

Event Data

Event Data

Possible problems to be solved:

Event Data

Possible problems to be solved:

- `event type` = Event classification

Event Data

Possible problems to be solved:

- `event type` = Event classification
- `TRK` = Track identification / classification

Event Data

Possible problems to be solved:

- `event type` = Event classification
- `TRK` = Track identification / classification
- `px, py, pz` = Momenta regression

A Bit of Context

A Bit of Context

- ATLAS and CMS collaborations rely more on Deep Learning (DL), especially for the HL-LHC

A Bit of Context

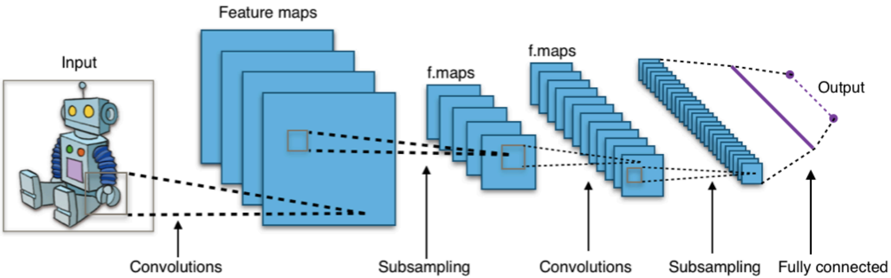
- ATLAS and CMS collaborations rely more on Deep Learning (DL), especially for the HL-LHC
- Convolutional Neural Networks have proven the best for event selection

A Bit of Context

- ATLAS and CMS collaborations rely more on Deep Learning (DL), especially for the HL-LHC
- Convolutional Neural Networks have proven the best for event selection
- Use of low-level data in DL pipelines is optimal

Back to DL: Convolutional Neural Networks (CNNs)

Back to DL: Convolutional Neural Networks (CNNs)



Input Preprocessing: Step 1

Input Preprocessing: Step 1

Step 1 Separate the data into files, one per event; a typical file looks like:

Input Preprocessing: Step 1

Step 1 Separate the data into files, one per event; a typical file looks like:

```
ID, Layer, Wire, Stereo,      X,       Y,      RT,     RC, TRK
 0,     0,   10,      1, 14.805,  74.431, 277.312,  705.0,   1
 1,     1,   12,      1,  9.394,  87.379, 482.156,  697.0,   1
 2,     2,   12,      1, 16.324,  98.873, 485.906,  480.0,   1
 3,     4,   12,      1, 15.192, 123.175, 444.937,  670.0,   1
 4,     5,   14,      1, 20.701, 134.500, 539.531,  418.0,   1
 5,     6,   16,      1, 14.483, 147.050, 555.0,    601.0,   1
 6,     7,   16,      1, 21.796, 157.583, 440.437,  724.0,   1
 7,     8,   16,      0, 36.448, 193.598, 513.375, 1052.0,   1
 8,     9,   17,      0, 30.724, 210.772, 660.375,  515.0,   1
 9,    10,   19,      0, 36.096, 226.694, 625.312,  609.0,   1
10,    11,   19,      0, 47.243, 240.962, 430.031,  435.0,   1
```

Input Preprocessing: Step 1

Step 1 Separate the data into files, one per event; a typical file looks like:

```
ID, Layer, Wire, Stereo,       X,       Y,      RT,     RC, TRK
 0,     0,   10,      1,  14.805,  74.431, 277.312,  705.0,   1
 1,     1,   12,      1,   9.394,  87.379, 482.156,  697.0,   1
 2,     2,   12,      1,  16.324,  98.873, 485.906,  480.0,   1
 3,     4,   12,      1,  15.192, 123.175, 444.937,  670.0,   1
 4,     5,   14,      1,  20.701, 134.500, 539.531,  418.0,   1
 5,     6,   16,      1,  14.483, 147.050, 555.0,    601.0,   1
 6,     7,   16,      1,  21.796, 157.583, 440.437,  724.0,   1
 7,     8,   16,      0,  36.448, 193.598, 513.375, 1052.0,   1
 8,     9,   17,      0,  30.724, 210.772, 660.375,  515.0,   1
 9,    10,   19,      0,  36.096, 226.694, 625.312,  609.0,   1
10,    11,   19,      0,  47.243, 240.962, 430.031,  435.0,   1
```

where

$$X = \frac{\mathrm{XE} + \mathrm{XW}}{2}, Y = \frac{\mathrm{YE} + \mathrm{YW}}{2}$$

Input Preprocessing: Step 2

Input Preprocessing: Step 2

Step 2 Generate images from each separate `csv` file

Input Preprocessing: Step 2

Step 2 Generate images from each separate `csv` file

We'll be looking at four types of images extracted from the data:

Input Preprocessing: Step 2

Step 2 Generate images from each separate `csv` file

We'll be looking at four types of images extracted from the data:

- `binary`

Input Preprocessing: Step 2

Step 2 Generate images from each separate `csv` file

We'll be looking at four types of images extracted from the data:

- `binary`
- `total`

Input Preprocessing: Step 2

Step 2 Generate images from each separate `csv` file

We'll be looking at four types of images extracted from the data:

- `binary`
- `total`
- `average`

Input Preprocessing: Step 2

Step 2 Generate images from each separate `csv` file

We'll be looking at four types of images extracted from the data:

- `binary`
- `total`
- `average`
- `per_hit`

Input Images: `binary`

Input Images: `binary`

- Build an image with only information from X and Y (no RC, or energy)

Input Images: `binary`

- Build an image with only information from X and Y (no RC, or energy)
- Effectively, this means we retain, at most, information about the momenta of each track

Input Images: `binary`

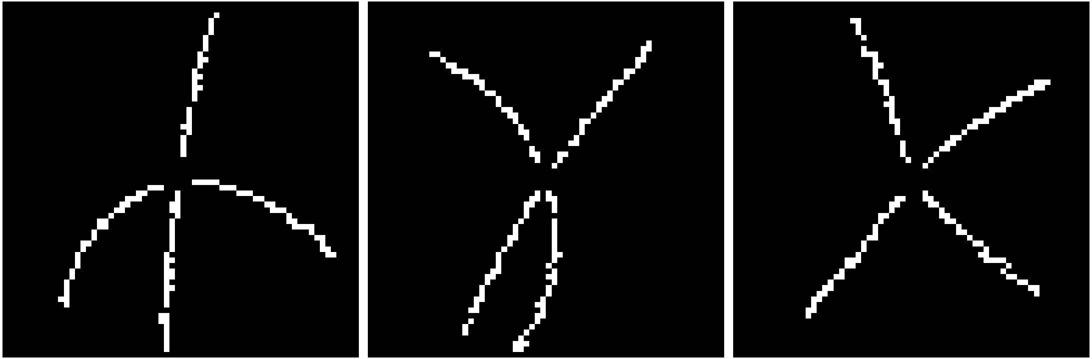
- Build an image with only information from X and Y (no RC, or energy)
- Effectively, this means we retain, at most, information about the momenta of each track
- The images will be binary, 0 for no hit, 1 for hit

Input Images: `binary`

- Build an image with only information from X and Y (no RC, or energy)
- Effectively, this means we retain, at most, information about the momenta of each track
- The images will be `binary`, 0 for no hit, 1 for hit

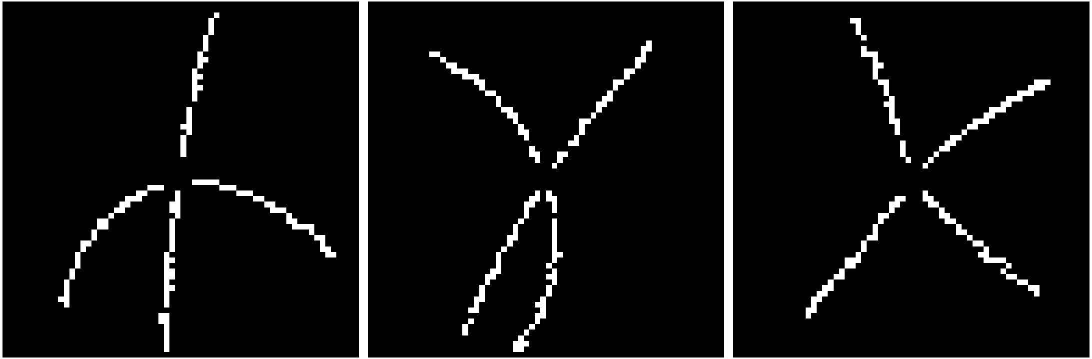
Input Images: `binary`

- Build an image with only information from X and Y (no RC, or energy)
- Effectively, this means we retain, at most, information about the momenta of each track
- The images will be `binary`, 0 for no hit, 1 for hit



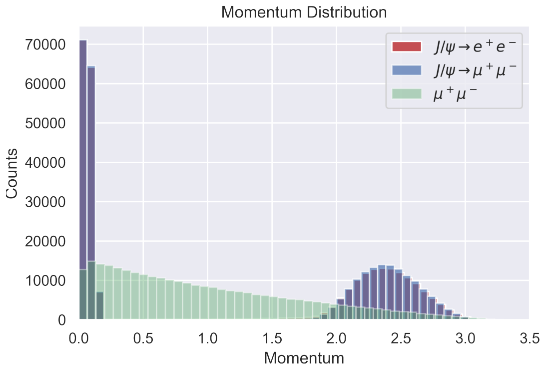
Input Images: `binary`

- Build an image with only information from X and Y (no RC, or energy)
- Effectively, this means we retain, at most, information about the momenta of each track
- The images will be `binary`, 0 for no hit, 1 for hit

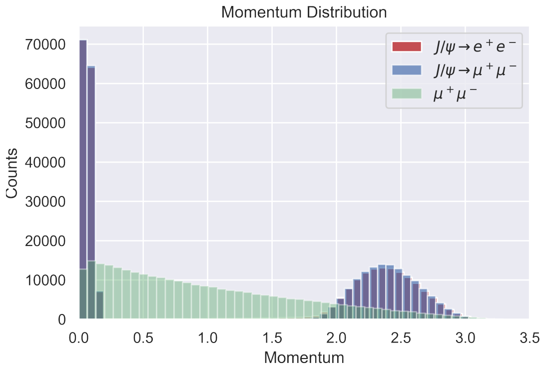


- Can we do something with this?

binary Images: Momentum Distribution

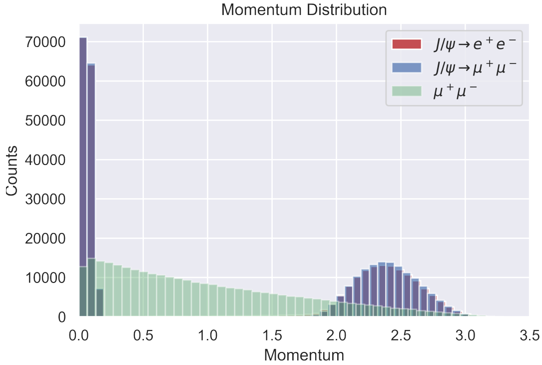


binary Images: Momentum Distribution



- For the $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$, the momenta distributions are virtually indistinguishable

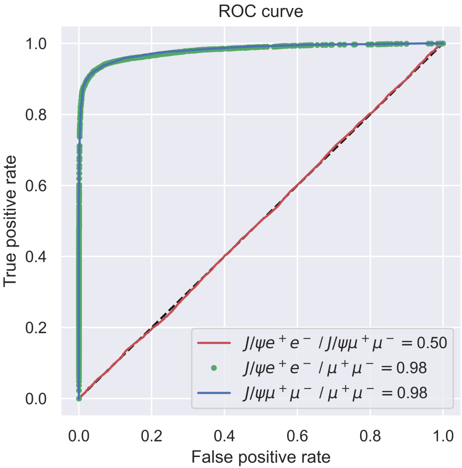
binary Images: Momentum Distribution



- For the $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$, the momenta distributions are virtually indistinguishable
- We predict that the classification between those events will be no better than random

`binary` Images: Results

binary Images: Results



binary Images: Results

binary Images: Results

With only information about the momenta:

binary Images: Results

With only information about the momenta:

- One cannot distinguish between $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$, the AUC being 0.5

binary Images: Results

With only information about the momenta:

- One cannot distinguish between $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$, the AUC being 0.5
- On the other hand, it is quite easy to distinguish between $J/\psi \to e^+e^-$ ($\mu^+\mu^-$) and $\mu^+\mu^-$, with an AUC of 0.98

Input Images: `total`

Input Images: `total`

We now incorporate some information about the particle energy:

Input Images: `total`

We now incorporate some information about the particle energy:

- For each event we only consider the `total` energy of that event

Input Images: `total`

We now incorporate some information about the particle energy:

- For each event we only consider the `total` energy of that event
- Each hit is assigned to have that value of the energy...

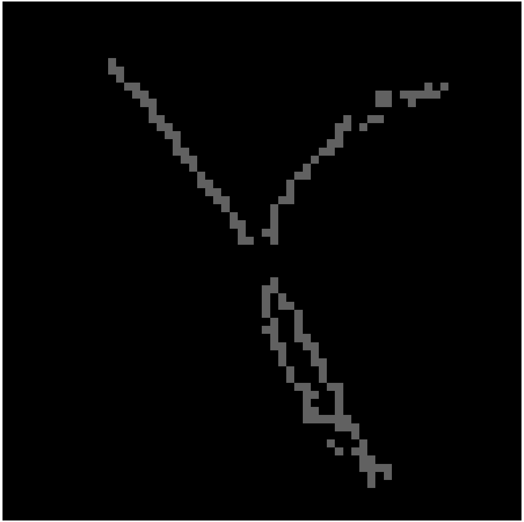
Input Images: `total`

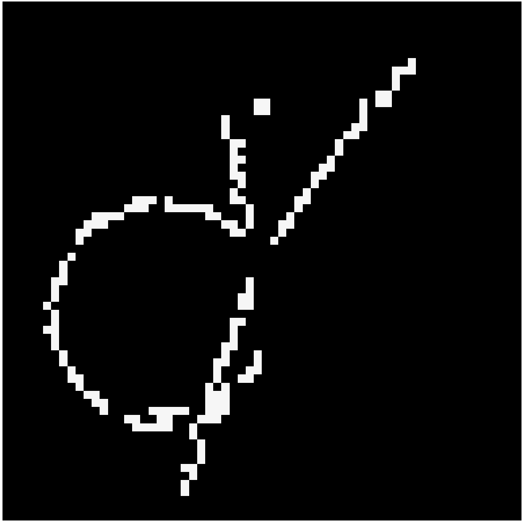
We now incorporate some information about the particle energy:

- For each event we only consider the `total` energy of that event
- Each hit is assigned to have that value of the energy...
- ...and then the entire data set is normalized

Input Images: `total`

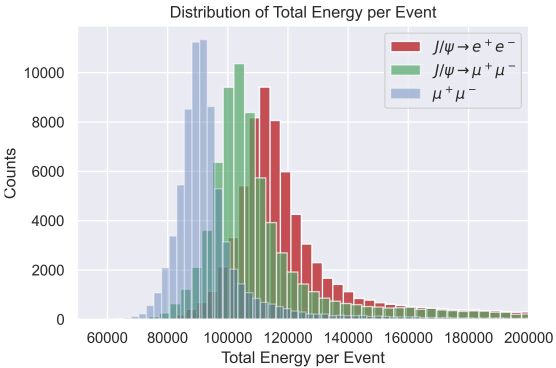
Input Images: `total`



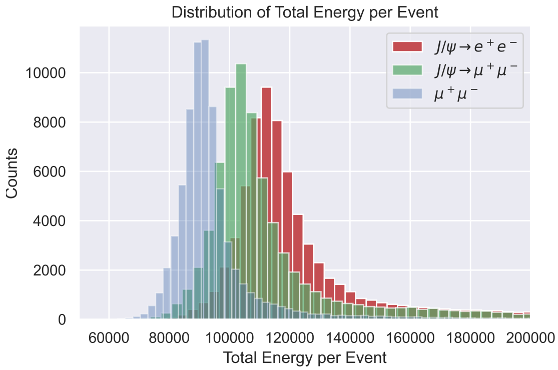


total Images: Energy Distribution

`total` Images: Energy Distribution

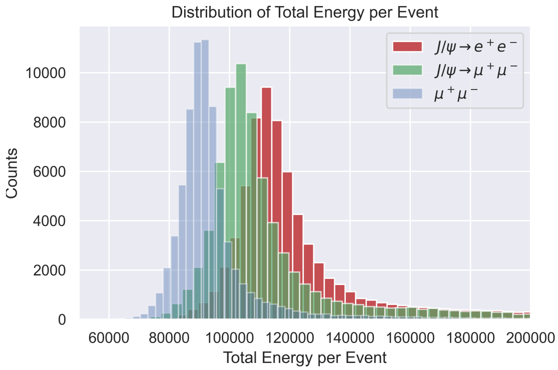


total Images: Energy Distribution



- The $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$ total energy distributions are more separated, though not completely

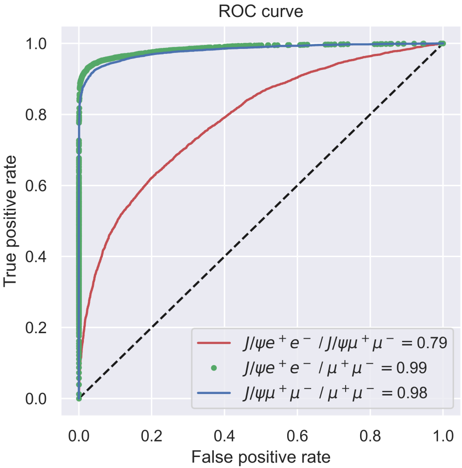
total Images: Energy Distribution



- The $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$ total energy distributions are more separated, though not completely
- We do expect better than random chance of distinguishing between these events

total Images: Results

total Images: Results



`total` Images: Results

`total` Images: Results

Including the information about energy, even at a gross level, allows one to:

`total` Images: Results

Including the information about energy, even at a gross level, allows one to:

- Distinguish now between $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$, with an AUC of ≈ 0.8

`total` Images: Results

Including the information about energy, even at a gross level, allows one to:

- Distinguish now between $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$, with an AUC of ≈ 0.8
- The AUC for the $J/\psi \to e^+e^-$ vs. $\mu^+\mu^-$ classification marginally increases from 0.98 to 0.99

`total` Images: Results

Including the information about energy, even at a gross level, allows one to:

- Distinguish now between $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$, with an AUC of ≈ 0.8
- The AUC for the $J/\psi \to e^+e^-$ vs. $\mu^+\mu^-$ classification marginally increases from 0.98 to 0.99
- The reason might be that the `total` energy distributions for these events has a smaller overlap

Input Images: `average`

Input Images: `average`

The `total` energy per event does not take into account the number of hits:

Input Images: `average`

The `total` energy per event does not take into account the number of hits:

- For each event we consider the energy `averaged` over the number of hits

Input Images: `average`

The `total` energy per event does not take into account the number of hits:

- For each event we consider the energy `averaged` over the number of hits
- Each hit is assigned to have that value of the energy...

Input Images: `average`

The `total` energy per event does not take into account the number of hits:

- For each event we consider the energy `averaged` over the number of hits
- Each hit is assigned to have that value of the energy...
- ...and then the entire data set is normalized

Input Images: `average`

The `total` energy per event does not take into account the number of hits:

- For each event we consider the energy `averaged` over the number of hits
- Each hit is assigned to have that value of the energy...
- ...and then the entire data set is normalized

Input Images: `average`

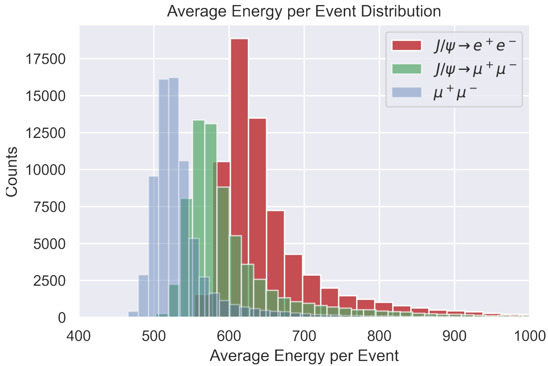
The `total` energy per event does not take into account the number of hits:

- For each event we consider the energy `averaged` over the number of hits
- Each hit is assigned to have that value of the energy...
- ...and then the entire data set is normalized

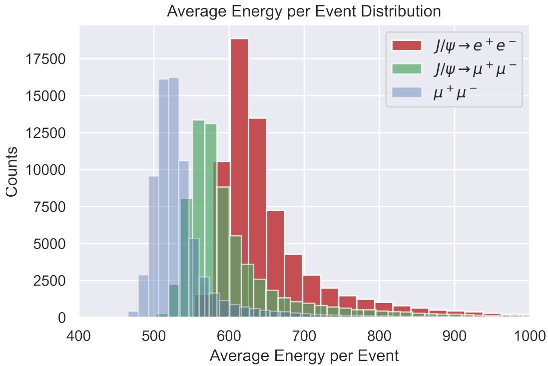
This should not be that different from the `total` energy case

average Images: Energy Distribution

average Images: Energy Distribution

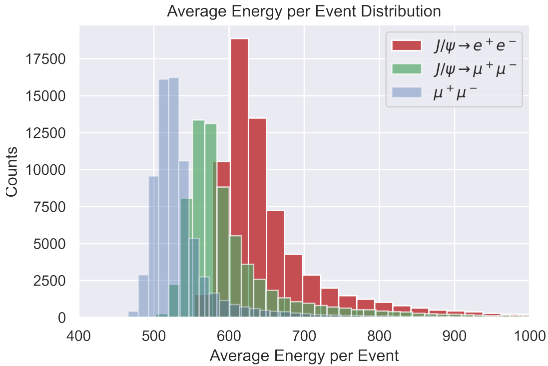


average Images: Energy Distribution



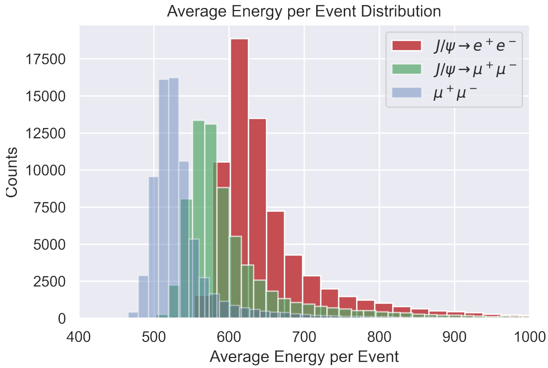
- This looks **very** similar to the energy distribution for `total` images…not surprisingly …

average Images: Energy Distribution



- This looks **very** similar to the energy distribution for `total` images...not surprisingly ...
- ...with maybe less overlap for the $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$ events

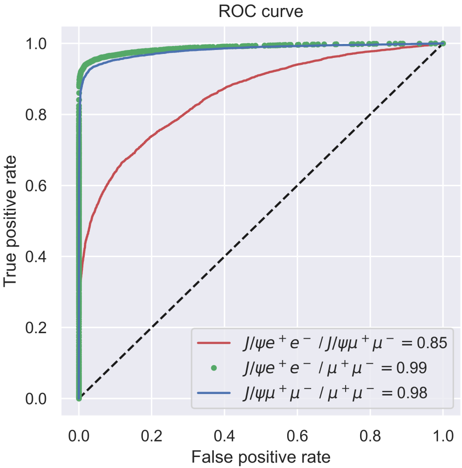
average Images: Energy Distribution



- This looks **very** similar to the energy distribution for `total` images...not surprisingly ...
- ...with maybe less overlap for the $J/\psi \rightarrow e^+e^-$ and $J/\psi \rightarrow \mu^+\mu^-$ events
- We'd expect similar classification performance, maybe slightly better for the two J/ψ events

average Images: Results

average Images: Results



`average` Images: Results

`average` Images: Results

Incorporating more energy information (`average`, rather than `total`), leads to:

`average` Images: Results

Incorporating more energy information (`average`, rather than `total`), leads to:

- No drastic changes in the classification performance ...

`average` Images: Results

Incorporating more energy information (`average`, rather than `total`), leads to:

- No drastic changes in the classification performance ...
- ...but a noticeable increase, from 0.79 to 0.85, for the J/ψ events

Input Images: per_hit

Input Images: `per_hit`

Finally, let's make use of even more finely grained information:

Input Images: per_hit

Finally, let's make use of even more finely grained information: :

- Each hit will be assigned it's own value of the energy...

Input Images: `per_hit`

Finally, let's make use of even more finely grained information: :

- Each hit will be assigned it's own value of the energy...
- ...which means that the images will no longer be have uniform intensity ...

Input Images: `per_hit`

Finally, let's make use of even more finely grained information: :

- Each hit will be assigned it's own value of the energy...
- ...which means that the images will no longer be have uniform intensity ...
- ...and then the entire data set is normalized

Input Images: `per_hit`

Finally, let's make use of even more finely grained information: :

- Each hit will be assigned it's own value of the energy...
- ...which means that the images will no longer be have uniform intensity ...
- ...and then the entire data set is normalized

Input Images: `per_hit`

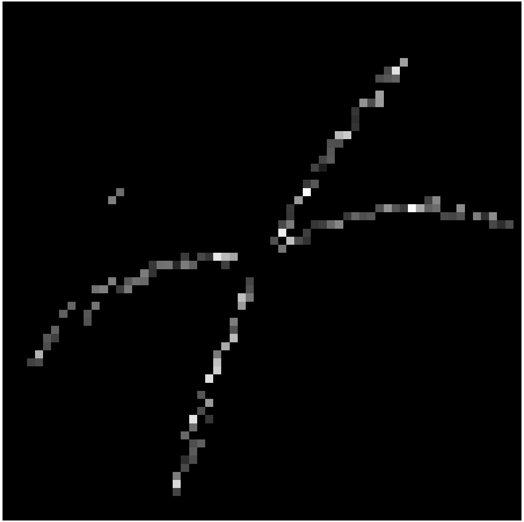
Finally, let's make use of even more finely grained information: :

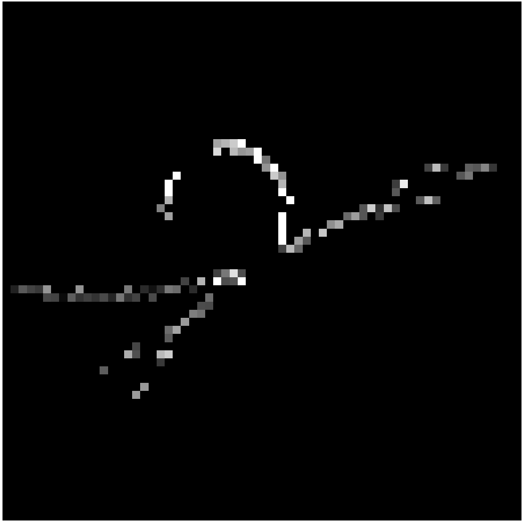
- Each hit will be assigned it's own value of the energy...
- ...which means that the images will no longer be have uniform intensity ...
- ...and then the entire data set is normalized

Can we do any better than before?

Input Images: `per_hit`

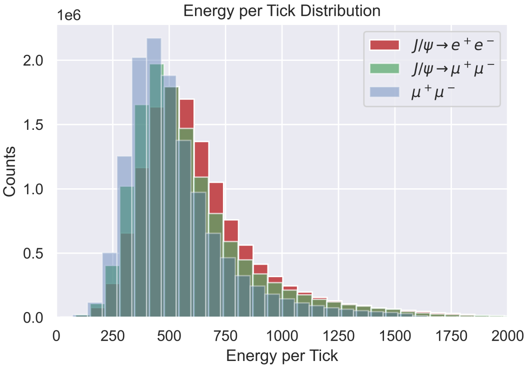
Input Images: per_hit



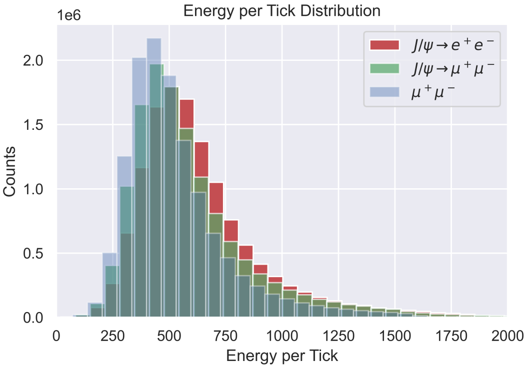


per_hit Images: Energy Distribution

per_hit Images: Energy Distribution

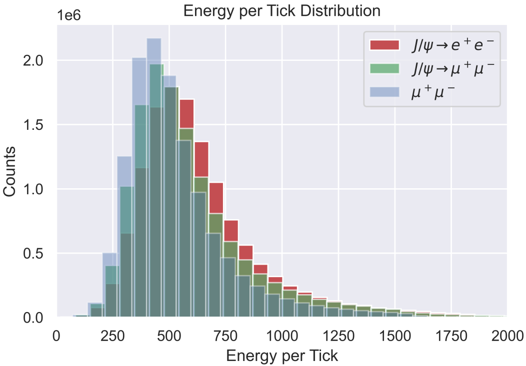


per_hit Images: Energy Distribution



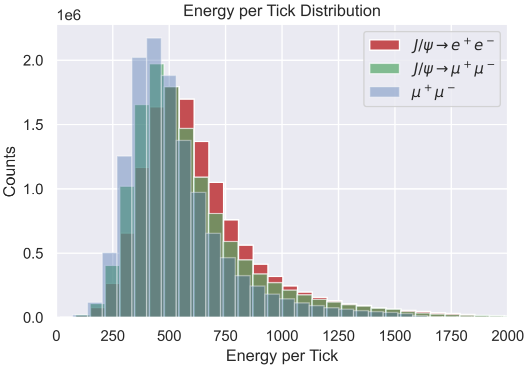
- This doesn't look promising **at all!**

per_hit Images: Energy Distribution



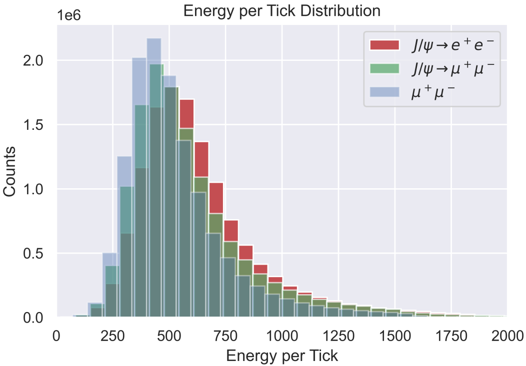
- This doesn't look promising **at all!**
- The `per_hit` energy distribution is essentially the same for all three types of events

`per_hit` Images: Energy Distribution



- This doesn't look promising **at all!**
- The `per_hit` energy distribution is essentially the same for all three types of events
- We'd expect similar performance as in the `binary` case...

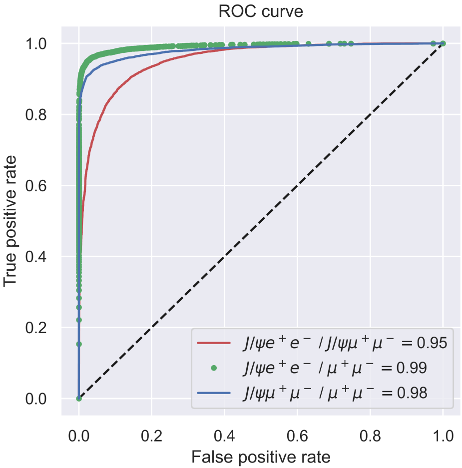
per_hit Images: Energy Distribution



- This doesn't look promising **at all!**
- The `per_hit` energy distribution is essentially the same for all three types of events
- We'd expect similar performance as in the `binary` case...
- ...wouldn't we?

per_hit Images: Results

per_hit Images: Results



per_hit Images: Results

per_hit Images: Results

Including all the energy information possible has some interesting effects:

`per_hit` Images: Results

Including all the energy information possible has some interesting effects:

- From the energy-per-hit distributions, one would expect doing no better than the `binary` case

per_hit Images: Results

Including all the energy information possible has some interesting effects:

- From the energy-per-hit distributions, one would expect doing no better than the binary case
- The performance, however, is the best we've gotten so far

per_hit Images: Results

Including all the energy information possible has some interesting effects:

- From the energy-per-hit distributions, one would expect doing no better than the binary case
- The performance, however, is the best we've gotten so far
- Why?

per_hit Images: Results

per_hit Images: Results

- The energy-per-hit distribution doesn't capture all the energy information from the images

per_hit Images: Results

- The energy-per-hit distribution doesn't capture all the energy information from the images
- In particular, it doesn't include any spatial relationships between hits

per_hit Images: Results

- The energy-per-hit distribution doesn't capture all the energy information from the images
- In particular, it doesn't include any spatial relationships between hits
- CNN's are **very** good at extracting spatial information from images

per_hit Images: Results

- The energy-per-hit distribution doesn't capture all the energy information from the images
- In particular, it doesn't include any spatial relationships between hits
- CNN's are **very** good at extracting spatial information from images
- We used a CNN for classification, so this might be the reason we've been successful...

per_hit Images: Results

- The energy-per-hit distribution doesn't capture all the energy information from the images
- In particular, it doesn't include any spatial relationships between hits
- CNN's are **very** good at extracting spatial information from images
- We used a CNN for classification, so this might be the reason we've been successful...
- ...but it's difficult to be sure

Multiclass classification

Multiclass classification

So far, we've consider **binary** classification, involving discriminating between two types of events

Multiclass classification

So far, we've consider **binary** classification, involving discriminating between two types of events

Realistically, we'd want a network that can take as input images from either of the three (or more) events, and classify them

Multiclass classification

So far, we've consider **binary** classification, involving discriminating between two types of events

Realistically, we'd want a network that can take as input images from either of the three (or more) events, and classify them

Enter multiclass classification

Multiclass classification

So far, we've consider **binary** classification, involving discriminating between two types of events

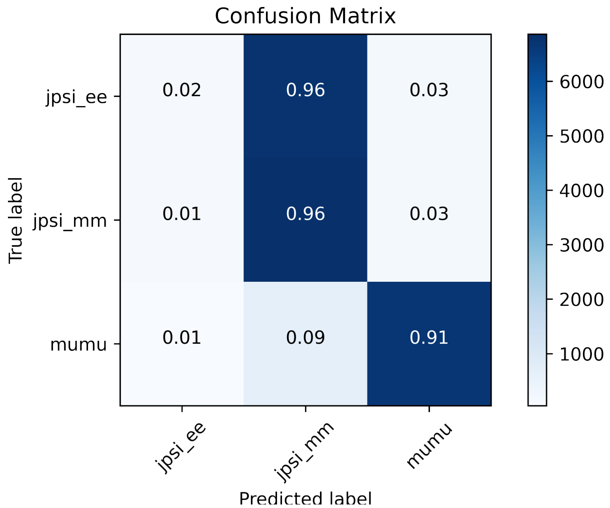
Realistically, we'd want a network that can take as input images from either of the three (or more) events, and classify them

Enter multiclass classification

We'll be using the same four encodings of the energy as before: `binary`, `total`, `average`, `per_hit`

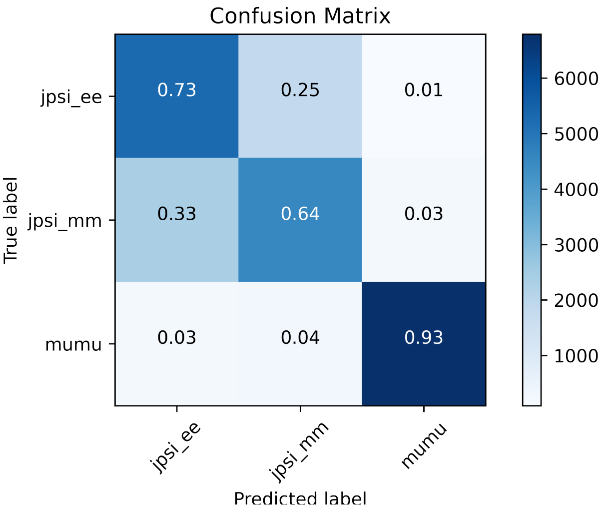
binary Images: Results

binary Images: Results



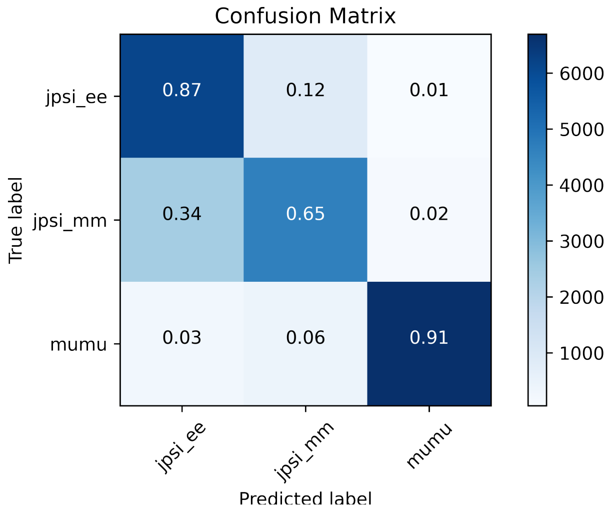
`total` Images: Results

total Images: Results



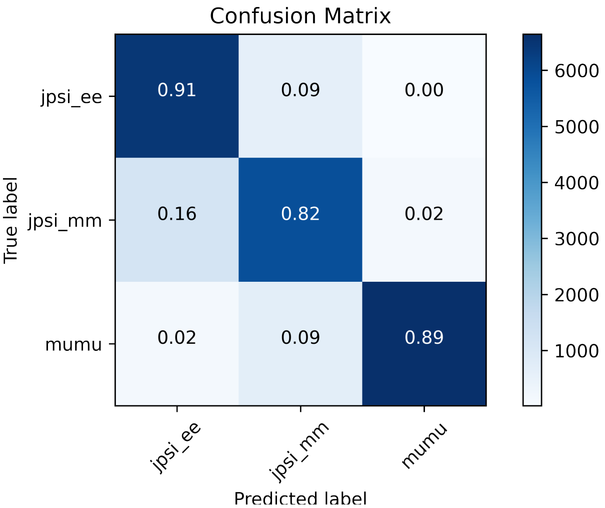
average Images: Results

average Images: Results



per_hit Image Results

per_hit Image Results



Multiclass classification: Summary

Multiclass classification: Summary

- Multiclass classification doesn't seem to work as well as binary classification

Multiclass classification: Summary

- Multiclass classification doesn't seem to work as well as binary classification
- However, in these plots, we've used the same "vanilla" CNN used for binary classification (with an appropriate output layer)

Multiclass classification: Summary

- Multiclass classification doesn't seem to work as well as binary classification
- However, in these plots, we've used the same "vanilla" CNN used for binary classification (with an appropriate output layer)
- We can do hyperparameter optimization to improve the classification performance

Concluding

Concluding

- Event classification with CNN is eminently doable

Concluding

- Event classification with CNN is eminently doable
- For this simple case, a very simple CNN can achieve high AUC in binary classification

Concluding

- Event classification with CNN is eminently doable
- For this simple case, a very simple CNN can achieve high AUC in binary classification
- And reasonable accuracy for multiclass classification

Concluding

- Event classification with CNN is eminently doable
- For this simple case, a very simple CNN can achieve high AUC in binary classification
- And reasonable accuracy for multiclass classification
- The way detector data is encoded into images makes a huge difference on the performance of the network (`binary`, `total`, `average`, `per_hit`)

Concluding

- Event classification with CNN is eminently doable
- For this simple case, a very simple CNN can achieve high AUC in binary classification
- And reasonable accuracy for multiclass classification
- The way detector data is encoded into images makes a huge difference on the performance of the network (`binary`, `total`, `average`, `per_hit`)
- This goes some way to make the model explainable

Where does this fit in a broader context?

Where does this fit in a broader context?

This is all very well, but not the whole story, obviously

Where does this fit in a broader context?

This is all very well, but not the whole story, obviously

- We cheated a bit, by ignoring the noise in the data 😋

Where does this fit in a broader context?

This is all very well, but not the whole story, obviously

- We cheated a bit, by ignoring the noise in the data 😋
- On the other hand, one can use semantic segmentation to train another network that will remove the noise, so it's all good!

Where does this fit in a broader context?

This is all very well, but not the whole story, obviously

- We cheated a bit, by ignoring the noise in the data 😋
- On the other hand, one can use semantic segmentation to train another network that will remove the noise, so it's all good!
- What about after classifying the events, what then?

Where does this fit in a broader context?

This is all very well, but not the whole story, obviously

- We cheated a bit, by ignoring the noise in the data 😋
- On the other hand, one can use semantic segmentation to train another network that will remove the noise, so it's all good!
- What about after classifying the events, what then?
- We might want to train another network, which will classify individual tracks

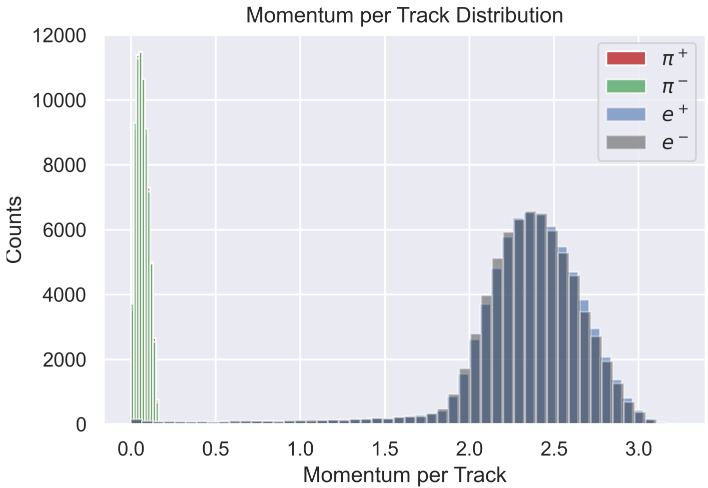
Where does this fit in a broader context?

This is all very well, but not the whole story, obviously

- We cheated a bit, by ignoring the noise in the data 😋
- On the other hand, one can use semantic segmentation to train another network that will remove the noise, so it's all good!
- What about after classifying the events, what then?
- We might want to train another network, which will classify individual tracks
- Before you ask, yes, you first have to identify the tracks, but we've seen this week that it can be done

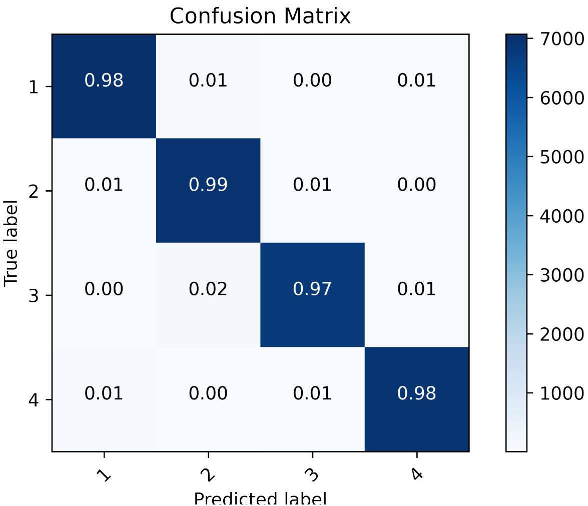
Track Classification: Some more Distributions

Track Classification: Some more Distributions



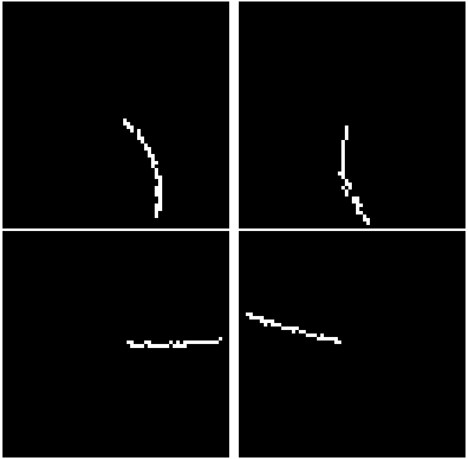
Track Classification: Some Results and Musings

Track Classification: Some Results and Musings



Track Classification: Some Results and Musings

Track Classification: Some Results and Musings



Track Classification: Some Results and Musings

Track Classification: Some Results and Musings

- No wonder the confusion matrix looks so pleasingly diagonal, the network can tell the difference between particles and antiparticles!

Track Classification: Some Results and Musings

- No wonder the confusion matrix looks so pleasingly diagonal, the network can tell the difference between particles and antiparticles!
- Momentum information seems to be enough for track classification (`binary`)

Further Steps

Further Steps

- It is also possible to use CNNs for regression:

Further Steps

- It is also possible to use CNNs for regression:
- That is to extract the momenta for each track

Further Steps

- It is also possible to use CNNs for regression:
- That is to extract the momenta for each track
- This is a bit more involved:

Further Steps

- It is also possible to use CNNs for regression:
- That is to extract the momenta for each track
- This is a bit more involved:
  - Mean-squared error loss

Further Steps

- It is also possible to use CNNs for regression:
- That is to extract the momenta for each track
- This is a bit more involved:
  - Mean-squared error loss
  - Custom loss encoding momentum conservation

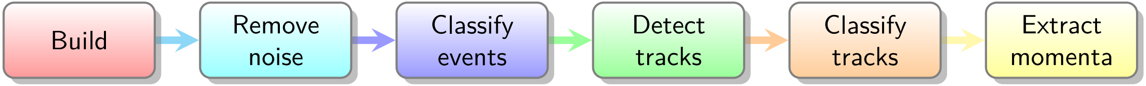
Putting It All Together

Putting It All Together

A pipeline which one could envision would look something like:

Putting It All Together

A pipeline which one could envision would look something like:



The Collaboration

The Collaboration

Team CIT

- Cristian Marocico[†]
- Leslie Zwerwer[‡]

Team KVI

- Johan Messchendorp
- Nasser Kalantar-Nayestanaki

The Collaboration

Team CIT

- Cristian Marocico[†]
- Leslie Zwerwer[‡]

Team KVI

- Johan Messchendorp
- Nasser Kalantar-Nayestanaki

[†] All responsibility for the style and any errors resides here
[‡] Did most of the work

Questions?

Thanks for your attention!