Just an Idea

January 28, 2023

Problem ?

» Particle level information (generated) get distorted in the detector level due to acceptance and in-efficiencies.

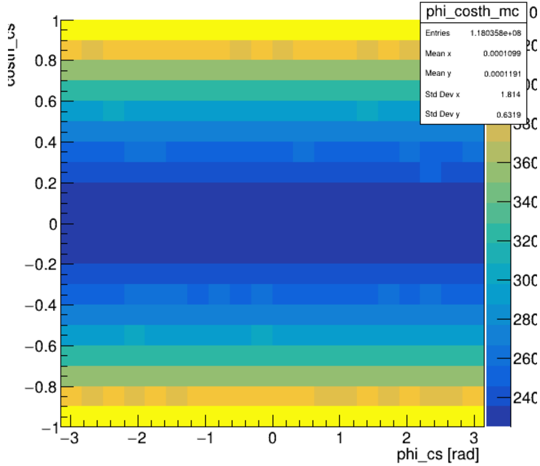


Figure 1: Generated ϕ vs. $\cos(\theta)$ distribution.

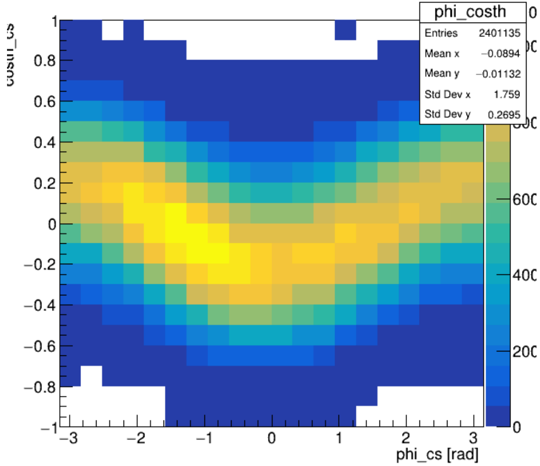


Figure 2: Reconstructed ϕ vs. $\cos(\theta)$ distribution.

» Need a method to extract particle level information using the detector level information (measured).

MNIST data and fully connected CNN's

- MNIST data set : Hand written numbers with 60k train images and 10k test images.
- Convolutional layers : Feature extraction.
- Fully connected layers : Classification.

How can we use this method to our problem ?

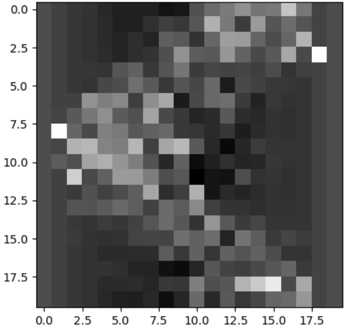


Figure 3: Reconstrued phi-costh distribution as a image. Note since we use event weight to fill the hitogram, we have scale the bin content using standard scaler in sklearn.

- We can assume bins in histogram is same as pixels in an image. We use reconstrued drell-yan events with FPGA1 trigger with $4.5 < mass < 8.0$.
- Input = phi-costh 2D histogram and target = [λ, μ, ν].
- We created 293 phi-costh histograms with λ, μ, ν = 1.0, 0.0, 0.0.
- Histograms were split to train: validation: test = 60: 20: 20.
- With batch size = 10, learning rate = 0.01, L2 penalty = 0.001 and epochs = 20.

Results

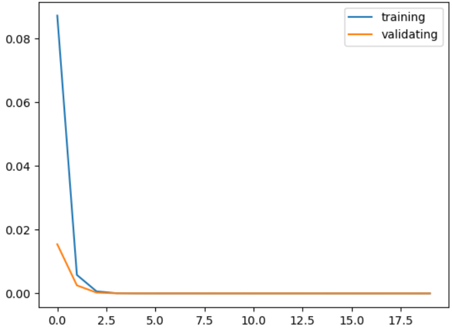


Figure 4: Loss curve

- Use fully connected CNN with regression (instead of classification as in MNIST data).
- We test the trained CNN with 10 images. Average values are;

  lambda = 1.0019 +/- 0.0037

  mu = -0.0006 +/- 0.0002

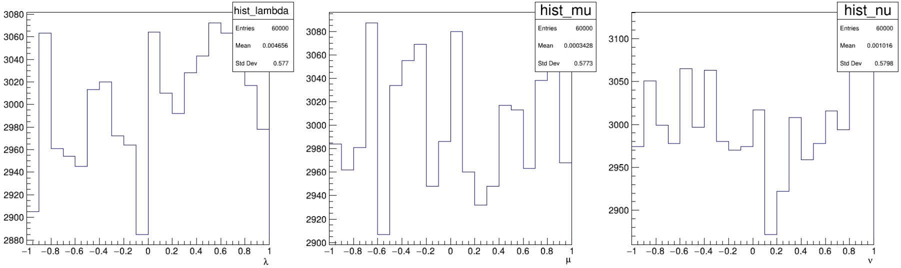
  nu = 0.0006 +/- 0.0005

- This results is biased (only one target).

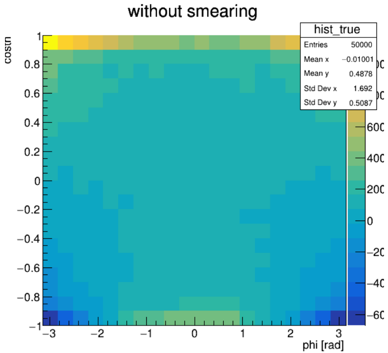
Pseudo data

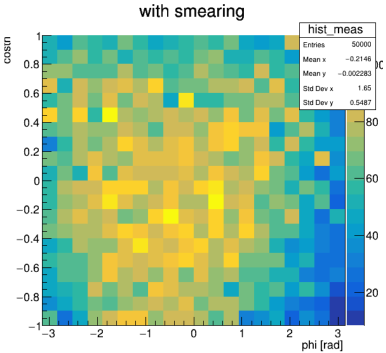
- We create $\phi = [-\pi, \pi]$ and $\theta = [0., \pi]$ randomly.
- Weights were created as $z = \lambda + \mu \cos(\phi) + \mu\phi^2 \cos(\theta)$ and $\lambda, \mu, \nu = [-1.0, 1.0]$ created randomly.
- Smearing were introduced for both θ and ϕ with;

```
double smear(double xt)
{
double xsmear = gRandom->Gaus(-0.5, 1.0);
return xt + xsmear;
}
```



- We create 60k histograms with 50k events per histogram. All the variables $[\phi, \theta, \lambda, \mu, \nu]$ are created randomly.
- Input = 2D histogram of ϕ vs. cos(θ) and target is λ, μ, ν. Our goal is to predict generated λ, μ, ν.





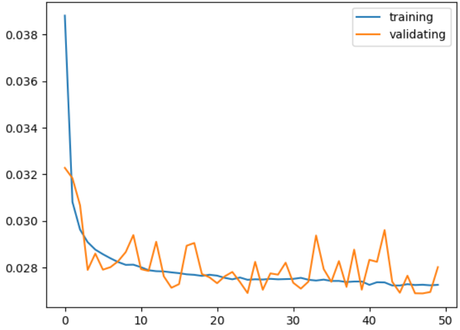


Figure 5: Loss curve for toy data.

- CNN is tested with 15 histograms with $\lambda, \mu, \nu = [0.7, 0.4, 0.3]$. The average values of the predictios are;

  lambda = 0.6492 +/- 0.0098

  mu = 0.4881 +/- 0.0620

  nu = 0.2280 +/- 0.0686

- Results are not that impressive. But can be improved.

- λ, μ, ν is introduced to the generated data by weights.
- If we can produce 2D histograms with different λ, μ, ν may be we can get better results.
- Git repo. https://github.com/dinupa1/unfoldML
- To do:
  - Plan to do a efficiency study after the survay is done.
  - Plan to do a false asymmetry study for J/ψ production.