Unfolding the W Momentum: Deep Neural Networks

Ainsleigh Hill
Outline

1. W boson production and detection overview
2. Unfolding:
   a. What is unfolding?
   b. Why do we need to unfold the W momentum?
   c. What are the current unfolding methods?
3. Neural Networks:
   a. How do they work?
   b. How can we use them to unfold the W momentum?
4. Bayes Unfolding:
   a. Priors and D’Agostini Method
   b. How can we apply this to a neural network?
5. Results
W Boson Mass

\[ W^- \mu^- \bar{e}^- \quad W^+ \mu^+ \bar{e}^+ \]

\[ p^+ \bar{p}^+ \]
Unfolding - What is it?

- Similar to reconstructing a blurred image
- Given a probability distribution, we want to determine the “true” distribution

Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]
Unfolding - Why?

- Momentum measurements collected from the CMS detector have a certain amount of “noise”
- We observe the picture on the left; we want to know the picture on the right
Unfolding - Current Methods

- Bin-by-bin matrix unfolding
- Downsides: does not scale well to multidimensional data

\[
\begin{bmatrix}
9 & 13 & 5 & 2 \\
1 & 11 & 7 & 6 \\
3 & 7 & 4 & 1 \\
6 & 0 & 7 & 10
\end{bmatrix}
\]
Deep Neural Network - What is it?

\[
\begin{align*}
\alpha_1^{(0)} & \quad \theta_1^{(0)} \\
\text{Node} & \quad \text{Weight} \\
\text{Layer 0, Node 1} & \quad \text{Layer 1, Node 1}
\end{align*}
\]

\[
\alpha_1^{(1)} = \frac{1}{1 + e^{-\left(\theta_0^{(0)} + \theta_1^{(0)} \alpha_1^{(0)} + \theta_2^{(0)} \alpha_2^{(0)} + \ldots\right)}}
\]
Deep Neural Network - Unfolding

Input Layer: Smeared (folded) data collected from detector

Hidden Layers: 2 hidden layers, 30 nodes each

Output Layer: Binned (unfolded) data, # nodes = # bins

\[
\begin{align*}
\mathbf{x}^{(i)} & \quad \text{Training data, input, Monte Carlo simulated (folded)} \\
\mathbf{z}^{(i)} & \quad \text{Training data, output, Monte Carlo simulated (unfolded, true)} \\
\mathbf{y}^{(i)} & \quad \text{Real data, input, collected from detector (folded)}
\end{align*}
\]
Bayesian Unfolding - Bayes’ Rule

\[ P(z \mid x) = \frac{P(x \mid z)P(z)}{P(x)} \]

- Probability of smeared distribution given true distribution
- Prior true distribution
- Probability of true distribution given smeared distribution
Bayesian Unfolding \[ P(z \mid x) = \frac{P(x \mid z)P(z)}{P(x)} \]

Even under the assumption that the Monte Carlo simulation is perfect, i.e.
\[ P(x^{data} \mid z^{data}) = P(x^{MC} \mid z^{MC}) \]
the prediction depends on prior true distribution \( P(z) \)

Since we are trying to measure the W boson momentum, we do not know the exact distribution to initialize the Monte Carlo simulation, so
\[ P(z^{data}) \neq P(z^{MC}) \]

Therefore:
\[ P(z^{data} \mid x^{data}) \neq P(z^{MC} \mid x^{MC}) \]
Bayesian Unfolding - D’Agostini

Problem: We are training our neural network on Monte Carlo data, which may result in a prediction biased towards the Monte Carlo prior distribution.

Idea to Build on: Giulio D’Agostini developed an iterative algorithm where the prior distribution, \( P(z) \), is updated using Bayes’ Theorem:

\[
P(z \mid x) = \frac{P(x \mid z)P(z)}{P(x)}
\]
Bayesian Unfolding - Reweighting

Our Idea: Apply D’Agostini’s unfolding to our neural network by iteratively reweighting the prior distribution in the loss function

\[ J = \sum_i \sum_k z_k^{(i)} \log h(x)_k^{(i)} + (1 - z_k^{(i)}) \log (1 - h(x)_k^{(i)}) \]

Prior distribution  Prediction

Use the prediction from the neural network to update the prior distribution:

\[ P(\tilde{z}) = P(\tilde{z})^{prev} \]
Putting it all Together

Train neural network using Monte Carlo simulated data

Get a prediction from the real data using the neural network

Update the distribution in the loss function using the prediction

Repeat until convergence
Neural Net Results: Training

Training data after 1000 epochs

Training data after 15000 epochs
Neural Net Results: Testing

Testing data after 1000 epochs

Testing data after 15000 epochs
Bayes Reweighting: No Error Term

No Reweighting

4 Iterations

1 Iteration

8 Iterations
Bayes Reweighting: Error Term

No Reweighting

1 Iteration

4 Iterations

8 Iterations