Accelerating dark matter search in the emulsion SHiP detector with deep learning

March 2019

Sergey Shirobokov$^{12}$, Andrey Ustyuzhanin$^{23}$, Andrey Golutvin$^1$

$^1$ Imperial College London, $^2$ Yandex School of Data Analysis, $^3$ National Research University Higher School of Economics
SHiP experiment

• Future experiment at CERN, dedicated to search of BSM physics.
• One way of looking for dark matter is to detect its scattering inside dense material by identifying electromagnetic (EM) showers.
• Main background comes from neutrino interactions.
iSHiP detector

- Detector consists from a "lasagna" structure of emulsion, lead and active tracking detectors.
- EM shower is a 3D cloud of points. Each point is described by 5 features.
- Points (tracks) are detected in emulsion films and active detectors.
Task

• Inputs:
  Shower energy $E \in \mathbb{R}^+$, coordinate $(X, Y, Z) \in \mathbb{R}^3$ of shower origin,
  Responses of target trackers – $N$ sets of points $\{(I_i, X_i, Y_i)\}, i = 1, \ldots, t$., where $X_i, Y_i$ - coordinate of point, $I_i$ - intensity and $N$ is the number of target trackers.

• Output: shower energy estimation $\hat{E}$,
  shower origin position estimation $(\hat{X}, \hat{Y}, \hat{Z})$.

• Metrics:
  \[
  \text{RMSE} = \sqrt{E((\theta - \hat{\theta})^2)}
  \]
  Parameter resolution $\sigma = \sigma\left(\frac{\theta - \hat{\theta}}{\theta}\right); \sigma(\theta - \hat{\theta})$
  where $\theta$ – estimated parameter.

• Summarizing – this is a classical regression task in machine learning:
  \[
  f: N \times \{(I_i, X_i, Y_i)\} \to (E, X, Y, Z)
  \]
Preprocessing: compression

- Target trackers has fine position resolution - 70 $\mu m$
- This gives high resolution "images" to work with, which leads to:
  - Increase in training/inference time
  - Overfitting to the data
- The resolution is artificially worsened to 700 $\mu m$, which was found to be good tradeoff between compression rate and speed.

Compression rate = \[ \frac{\# \text{ non zero pixels after compression}}{\# \text{ non zero pixels in initial image}} \]
Gaussian based approach

- 2D Gaussian is fitted to each of the target trackers.
- Gaussian variance, covariance and total intensity in each of the trackers are selected as features for regression model.
- Total: \(N(3 + 3)\) features, \(N\)- number of trackers.
- Regression model (XGBoost) is fitted twice:
  - Using initial features to predict energy.
  - Using initial features and predicted energy to predict position.
CNN based approach

- Point cloud is transformed to image.
- Images from different tracker plane are located as channels of the image.

Final image size: \( \text{Image}_{width} \times \text{Image}_{height} \times N_{TT} \).

- Different loss functions are tested: MSE, MAE, Huber loss, RMSPE

\[
L_{\text{Huber}} = \begin{cases} 
0.5(y_i - \hat{y}_i)^2, & \text{if } |y_i - \hat{y}_i| < 0.5 \\
|y_i - \hat{y}_i| - 0.5, & \text{otherwise} 
\end{cases} \\
L_{\text{RMSPE}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{y_i^2}}
\]
CNN based approach: architecture

- Classical CNN architectures with Dropout or/and BatchNorm layers result in biased predictions on train and test sets.
- Solution: Remove Dropout and add $L2$ regularization with learning rate and weight decay.

**With Dropout**

**Distance (Inference mode)**

- True distance to end of brick
- Reco distance to end of brick

**Distance (Train mode)**

- True distance to end of brick
- Reco distance to end of brick

**Without Dropout**

**Distance (Inference mode)**

- True distance to end of brick
- Reco distance to end of brick

**Distance (Train mode)**

- True distance to end of brick
- Reco distance to end of brick

Distance to the end of the brick, cm
CNN based approach: CoordConv

- CoordConv* layer – add two new maps to the initial images: coordinates of pixels in x and y planes, then performs convolution.
- Provides faster convergence.
- Result in same value of loss function.

Algorithms: results

- 2D Gaussian: $\text{RMSE}(E) = 7.23 \pm 0.03 \text{ GeV}, \text{RMSE}(d) = 0.92 \pm 0.01 \text{ cm}$.
- CNN: $\text{RMSE}(E) = 6.84 \pm 0.02 \text{ GeV}, \text{RMSE}(d) = 0.79 \pm 0.01 \text{ cm}$.
## Algorithms: comparison

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RMSE: (E) GeV; (d) cm</th>
<th>Resolution: (E), %; (d) cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian fit</td>
<td>~7.23; ~0.92</td>
<td>~26%; 0.92</td>
</tr>
<tr>
<td>CNN, MSE loss</td>
<td>~7; ~0.87</td>
<td>~25%; 0.87</td>
</tr>
<tr>
<td>CNN, Huber loss</td>
<td>~6.84; ~0.79</td>
<td>~24%; 0.79</td>
</tr>
<tr>
<td>CoordConvNN, Huber loss</td>
<td>~6.84; ~0.79</td>
<td>~24%; 0.79</td>
</tr>
</tbody>
</table>
X,Y coordinates prediction

• This is clusterisation task, since target tracker response already explicitly contains information about X and Y coordinates.

Algorithm 1: (X,Y) coordinates of vertex location

\[
\begin{align*}
TT_{\text{stations}} &= \text{sort}(TT_{\text{stations}}, \text{key=number of hits}); \\
\text{for } \text{station in } TT_{\text{stations}} \text{ do} \\
    &\quad n_{\text{clusters}} = \text{FindClusters(station);} \\
    &\quad \text{if } n_{\text{clusters}} > 0 \text{ then} \\
    &\quad \quad \text{for } \text{cluster in } n_{\text{clusters}} \text{ do} \\
    &\quad \quad \quad \text{cluster}._\text{centers}.\text{add}(\text{FindCenter(cluster)}); \\
    &\quad \quad \text{break;} \\
    &\quad \text{if } \text{len(cluster}._\text{centers}) > 0 \text{ then} \\
    &\quad \quad \text{SelectBestCluster(cluster}._\text{centers})_; \\
    &\quad \text{else} \\
    &\quad \quad \text{reject event;} \\
\end{align*}
\]

where

\[
\text{FindCenter(cluster)}_{x,y} = \frac{\sum_{i\in\text{cluster}} (x, y) \times I_i}{\sum_{i\in\text{cluster}} I_i},
\]
X,Y coordinates prediction

- Base algorithm: $\sigma_x \sim 0.3 \text{ cm}$, $\sigma_y \sim 0.3 \text{ cm}$
- Clusterisation: $\sigma_x \sim 0.1 \text{ cm}$, $\sigma_y \sim 0.15 \text{ cm}$
Conclusion

• New approaches in electromagnetic shower identification are devised. They do not rely on calorimeter information (emulsion) and use non-purpose detectors to achieve comparable to emulsion performance.

• Obtained position and energy resolution $\sigma_d \sim 0.79\ cm$, $\sigma_E \sim 24\ %$ improved base algorithm in 1.2 times.

• Obtained XY position resolution $\sigma_X \sim 0.1\ cm$, $\sigma_Y \sim 0.15\ cm$ improved base algorithm factor 2 — 3 times.

• Obtained results will help to optimize the cost of the SHiP experiment and improve physics performance.
Backup
Data

• It is easier to predict \((d, \hat{E})\) and \((\hat{Y}, \hat{X})\), where \(d = Z_{\text{vertex}} - Z_{\text{wall}}\).

Data:
• 500000 shower events.
• \(E \in [1, 100] \, GeV, \, (X, Y) \in [-1, 1]^2 \, cm, \, d \in [-7.5, 0] \, cm\).
• At least 5 points in tracker response.
Gaussian approach: features

TT: 0

TT: 1

TT: 2

Energy, GeV  Energy, GeV  Energy, GeV
RMSPE answer distribution
CNN algos: what was tested

- Binarise \((d, \hat{E})\). Make classification task first. Build regression inside each bin.
- Use full resolution images, without compression.
- Using of denoising AE to remove nose hits in the trackers – low energy electrons away from shower origin.
- Usage of locally connected layers in the CNN.
- Usage of CoordConv layers in the CNN.

https://arxiv.org/abs/1807.03247
Energy resolution for different length of brick
Position resolution for different length of brick
Dependence on X0

- To compare results, we have calculated mean resolution over bins in given energy range:

\[ \sigma = \frac{1}{N} \sum_{i=1}^{n} \sigma_i \]

n - number of bins in given energy range.