Single Electron NN

Nicholas Todoroff

July 19, 2018

Model

Event $\longrightarrow$ Cell Prediction $\longrightarrow$ Point Prediction

- Line read from the input file
- Beginning 256 elements arranged into a 16×16 array E assuming column-major ordering
- Normalized by dividing by maximum energy

Model

Event $\longrightarrow$ Cell Prediction $\longrightarrow$ Point Prediction

- Single 3×3 convolution layer
- Produces cell-wise probability distribution X of electron starting position
- Essentially trains to select the highest energy cell
- Learned parameters:

$$\begin{array}{cc} \text{Convolution} & \text{Bias} \\ \begin{pmatrix} 0.9939 & 3.9430 & 1.0285 \\ 3.9849 & 4.7335 & 3.9563 \\ 0.9052 & 3.9405 & 1.0783 \end{pmatrix} & 3.1200 \cdot 10^{-9} \end{array}$$

Model

Event $\longrightarrow$ Cell Prediction $\longrightarrow$ Point Prediction

- Selects the most probable cell p from the previous layer
- Transforms $X \mapsto X'$ such that X'_p is the central element of X'
- Applies activation function σ (ReLU)
- Applies a dense NN with one hidden layer of 512 neurons
- Normalizes output by the factor $48/16$ (detector size / width in cells)
- Outputs starting position P of the electron relative to the center p' of p

Example

$$X = \frac{1}{19}\begin{pmatrix} 1 & 3 & 4 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}, \quad p = (1,3), \quad X' = \frac{1}{19}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 4 & 0 & 0 \\ 1 & 2 & 3 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Model

Mathematical Description

Definitions

- $E \in \mathbf{R}^{16\times 16},\ X \in \mathbf{R}^{16\times 16},\ X' \in \mathbf{R}^{31\times 31}$
- $C \in \mathbf{R}^{3\times 3},\ W_1, W_2 \in \mathbf{R}^{512\times 961}, b \in \mathbf{R},\ b_1 \in \mathbf{R}^{512},\ b_2 \in \mathbf{R}^2$
- $p = (p_1, p_2),\ p' = (p'_1, p'_2)^\mathsf{T},\ P = (P_1, P_2)^\mathsf{T}$
- $\left[\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}\right] = (1, 3, 2, 4)^\mathsf{T}$

Model

$$P = b_2 + W_2\sigma(b_1 + W_1\sigma([X']))$$

$$X'_{ij} = \begin{cases} X_{i+p_1-16,\, j+p_2-16} & \text{if } 1 \le i + p_1 - 16 \le 16 \text{ and} \\ & \quad 1 \le j + p_2 - 16 \le 16, \\ 0 & \text{otherwise.} \end{cases}$$

$$X_{ij} = (C * E)_{ij} + b, \quad p = \max_q X_q, \quad p'_1 = \frac{48}{16}\left(p_1 - \frac{1}{2}\right) - \frac{48}{2}, \quad p'_2 = \text{similar}$$

Model

Optimization

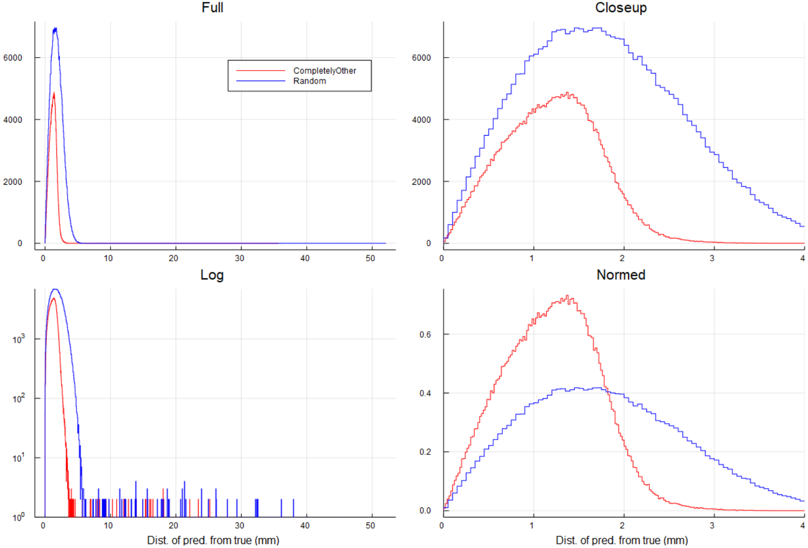
Loss Function:

$$F(X,P,Q) = -\ln(\epsilon + \max(0, X_P)) - \sum_{r \neq P} \ln(1 + \epsilon - \min(1, X_r)) + \lambda ||P - Q||^2$$

- For the tested model, $\epsilon = 0.01$, $\lambda = 1$
- Used the ADAM solver with a learning rate $\eta = 0.01$
  - Not as harsh as gradient descent
- Negligible difference in performance after training for second epoch

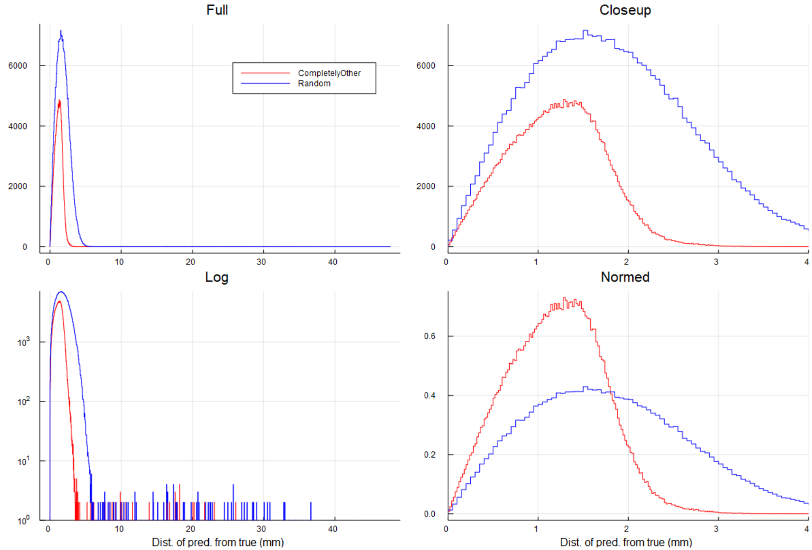
Results

Error Histogram: Training Set



Results

Error Histogram: Validation Set



Results

Statistics

- "Random" selects highest energy cell, then a uniformly random point within that cell

Model Prediction Statistics

Model	Set	Mean	Mode	Stdev	$P(< 3$ mm$)$	90^{th}-	95^{th}-	99^{th}-%tile
Random	Train	1.823	1.712	1.115	89.40%	3.036	3.426	4.176
	Valid	1.819	1.543	1.107	89.60%	3.024	3.411	4.150
Other	Train	1.216	1.041	0.700	99.78%	1.860	2.045	2.472
	Valid	1.215	1.945	0.700	99.80%	1.859	2.043	2.472

All numbers are in units of mm (sans percentages)