Calorimetry and Deep Learning

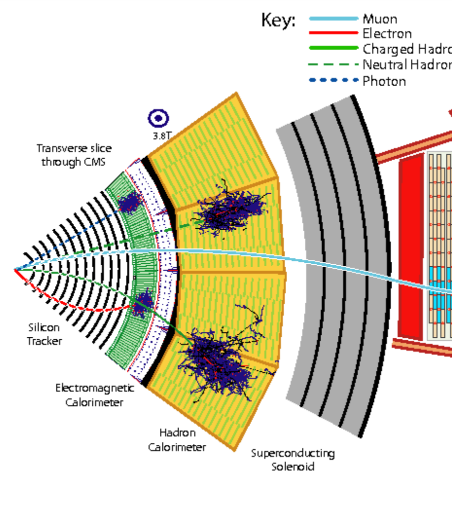
Simon Schnake

30. Oktober 2018

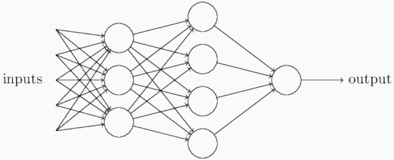
Universität Hamburg

Calorimetry in Particle Physics

- apparatus to measure the energy of a particle
- destructive process
- particle shower
- sampling or homogenous
- electromagnetic calorimeter
  $\rightarrow$ via EM interaction
- hadronic calorimeter
  $\rightarrow$ via strong nuclear force
- $\frac{\sigma_E}{E} \propto \frac{1}{\sqrt{E}}$



Deep Learning



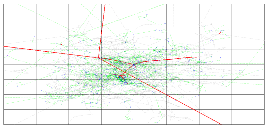
- Neuron: $\sigma_i(W_i X_{i-1} + b_i)$
- Goal: Fitting NeuralNet(X_0) to y_{true}
- Procedure:
    - Define $Loss(y_{\text{true}}, y_{\text{pred}})$ to minimize
    - Use Backpropagation to update W_i, b_i via an optimizer

Simulation Setup

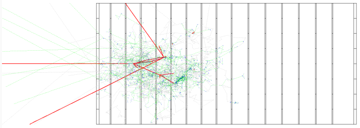
- Simulation with Geant4
- e^- from 0. to 10 GeV
- 300,000 events

layer	scint	absorber
layer 0	9mm	40 mm SS
layer 1 - 8	3.7mm	50.5 mm brass
layer 9 - 14	3.7mm	56.5 mm brass
layer 15	3.7 mm	75 mm SS
layer 16	9mm	

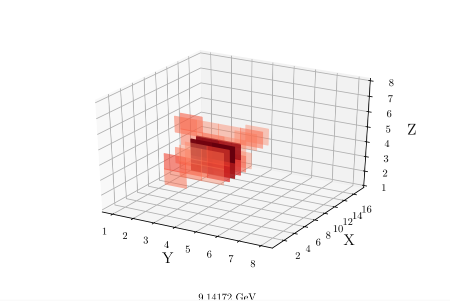
Front view

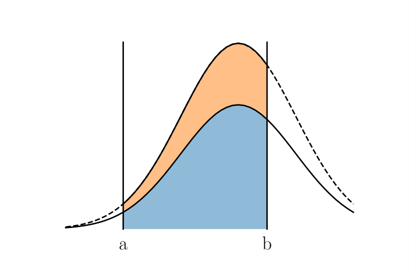


Side view



Resulting Data

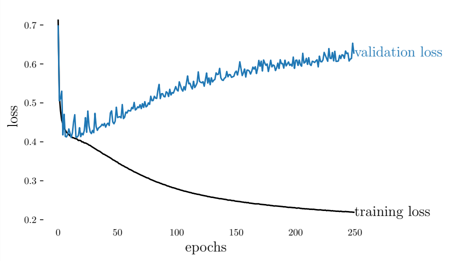


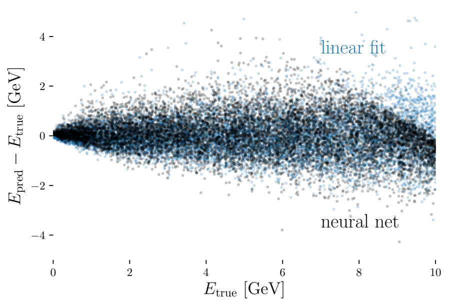


Neuralnet

Layer	Type	Activation
1	Conv2D(32, (2,2), strides = (1, 1))	ReLu
2	Flatten()	
3-5	Dense(128)	ReLu
6	Dense(10)	ReLu
7	Dense(1)	Linear
- Loss = Mean Squared Error
- Optimizer = RMSprop

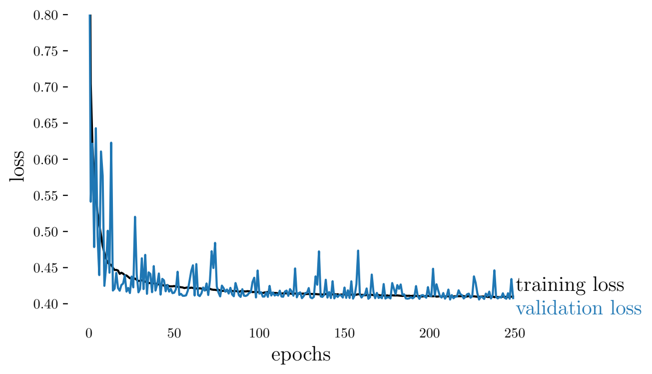
First Results of the Neuralnet



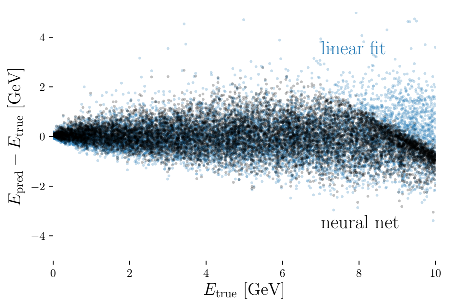


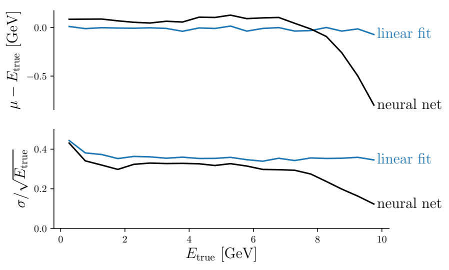
Data Augmentation

- flip the data arrays in the y axes
- rotate around the x axes
- shift in y and z



Data Augmentation



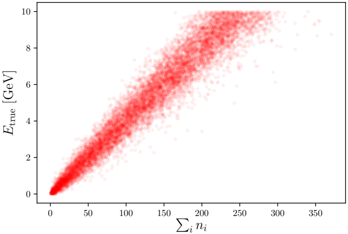


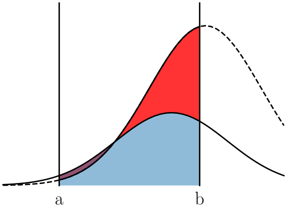
Maximum Likelihood Loss

Assumption: true values are gaussian distributed with const std σ

$$\begin{aligned}
\text{max likelihood} &= \max \prod \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_{\text{true}} - y_{\text{pred}})^2}{2\sigma^2}} \\
\Rightarrow \text{max log likelihood} &= \max \sum -\frac{(y_{\text{true}} - y_{\text{pred}})^2}{2\sigma^2} - \ln\left(\sqrt{2\pi\sigma^2}\right) \\
&= -\max \sum \frac{(y_{\text{true}} - y_{\text{pred}})^2}{\cancel{2\sigma^2}} \cancel{- \ln\left(\sqrt{2\pi\sigma^2}\right)} \\
&= \min \sum (y_{\text{true}} - y_{\text{pred}})^2
\end{aligned}$$

Maximum Likelihood Loss



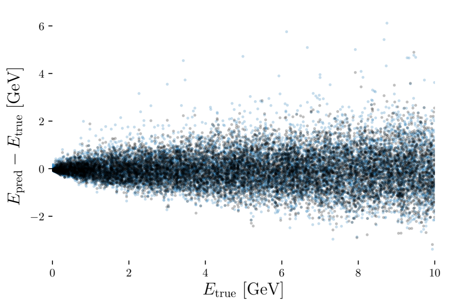


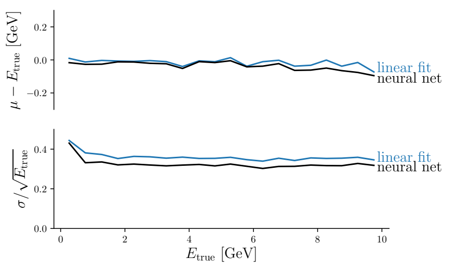
$$\Rightarrow \text{max log likelihood} = \min \sum \ln\left(\frac{\mathrm{Norm}(y_{\mathrm{true}}|y_{\mathrm{pred}},\sigma)}{\int_a^b \mathrm{Norm}(y_{\mathrm{true}}|y_{\mathrm{pred}},\sigma)}\right)$$

$$= \min \sum \frac{(y_{\mathrm{true}} - y_{\mathrm{pred}})^2}{2\sigma^2} + \ln\left(\frac{1}{2}\left(\mathrm{erf}(\frac{y_{\mathrm{pred}} - a}{\sqrt{2}\sigma}) - \mathrm{erf}(\frac{y_{\mathrm{pred}} - b}{\sqrt{2}\sigma})\right)\right)$$

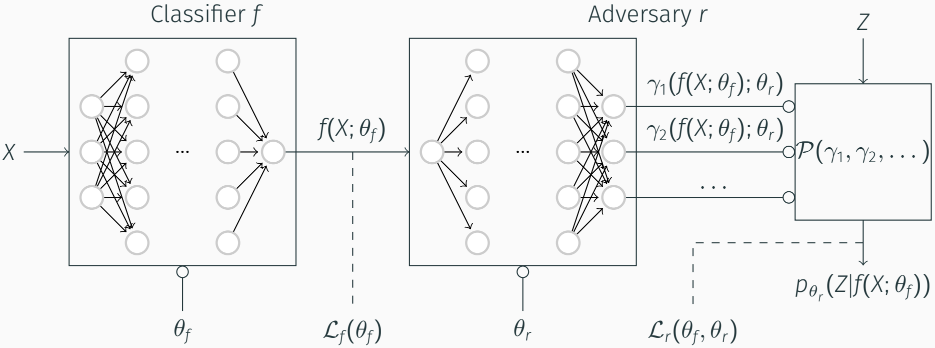
Maximum Likelihood Loss

$$\sigma = 0.31\sqrt{y_{\text{true}}}$$





Adversarial Training

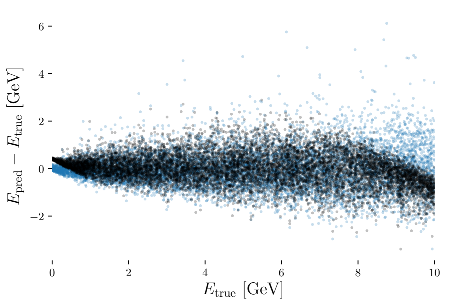


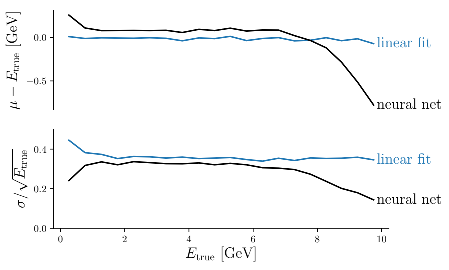
$$E_\lambda(\theta_f,\theta_r) = \mathcal{L}_f(\theta_f) - \lambda\mathcal{L}_r(\theta_f,\theta_r)$$

G. Louppe, ”Learning to Pivot with Adversarial Networks”, 1611.01046

Adversarial Training

- R estimates y_{true} given $\frac{y_{\text{true}} - D(X)}{\sqrt{y_{\text{true}}}}$





Summary

- The usage of neural nets at analyzing calorimeter data
- Data augmentation is good tool for preventing overfitting in calorimeter data
- Two different ways of targeting systematic errors in neural nets
- The presented neuralnet performs $\approx 10\%$ better than a linear fit
- In this setup the usage of a maximum likelihood loss performs better than the adversarial approach