ENERGY REGRESSION WITH DEEP LEARNING IN PARTICLE COLLIDER PHYSICS

Simon Schnake

17. Juni 2019

Universität Hamburg

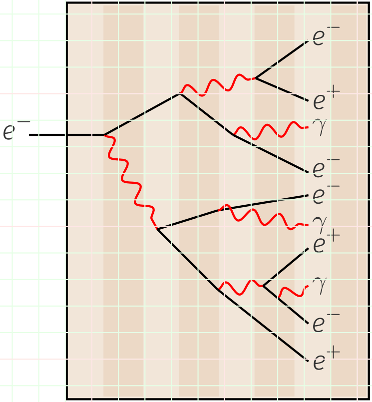
Flow of Content

TODO: Ablaufplan

Toy Calorimeter Analysis

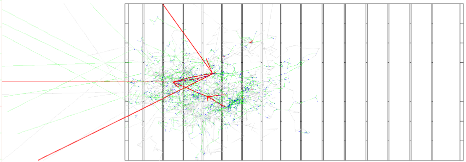
Calorimetry

- leading processes in shower development
  - bremsstrahlung
  - pair production
- measure the energy of the particle shower (destructive)
- resolution of a calorimeter $\frac{\sigma}{E} = \frac{a}{\sqrt{E}} \otimes b \otimes \frac{c}{E}$



Simulation Setup

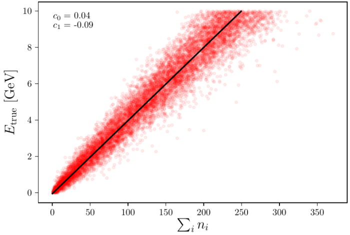
- Simulation with Geant4
- e^{-} from 0. to 10 GeV
- 300,000 events
- 8x8x17 (1088) scint cells
- measure charged tracks in each scintillator cell



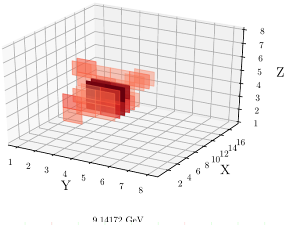
layer	scint	absorber
layer 0	9mm	40 mm steel
layer 1 - 8	3.7mm	50.5 mm brass
layer 9 - 14	3.7mm	56.5 mm brass
layer 15	3.7 mm	75 mm steel
layer 16	9mm	

Resulting Data

Calibration by linear fitting:

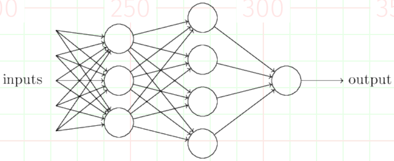


Visualization of the data:



⇒ data structure suggest 3D Conv. net

Deep Learning



- ss input X to the network
- Calculate output $\hat{Y}$ and difference to true value Y (loss function)
- inimize loss function:
  - Calculate gradient of the loss
  - Use gradient to update weights W, b

$\Big\}$ *Backpropagation*

$a^{[0]} := X$

$a^{[l]} := \sigma^{[l]}(W^{[l]}a^{[l-1]} + b^{[l]})$

$\hat{Y}(X) := a^{[L]}$

Example Loss:

$\mathcal{L}(\hat{Y}, Y) = (\hat{Y} - Y)^2$

Deep Learning Model

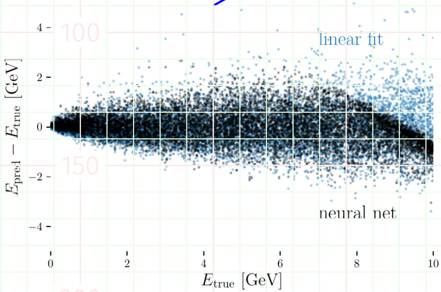
Layer	Type	Activation	Output Shape	# Parameter
0	Input		(8, 8, [illegible], 1)	
1	Conv3D(32, (3,3,3))	ReLU	(8, 8, 17, 32)	896
2	Conv3D(10, (3,3,3))	ReLU	(6, 6, 15, 10)	8650
3	Conv3D(5, (5, 5, 5))	ReLU	(2, 2, 11, 5)	6255
4	Flatten()		220	
5	Dense(128)	ReLU	128	28288
6-7	Dense(128)	ReLU	128	16512
8	Dense(10)	ReLU	10	1290
10	Dense(1)	Linear	1	11

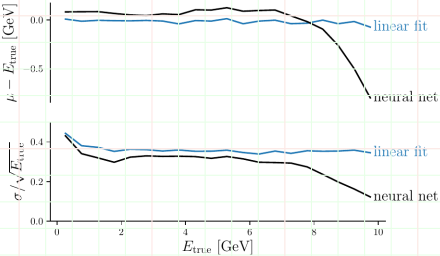
- Loss = mean squared error
- Optimizer = RMSprop
- Data Augmentation with symmetry transformation (flipping, rotating and shifting)

Data Augmentation

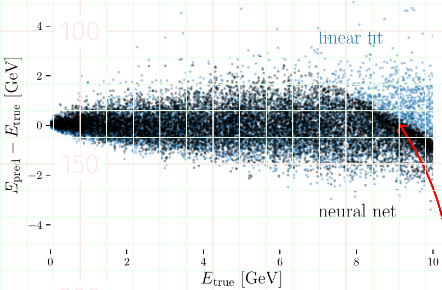
First Results

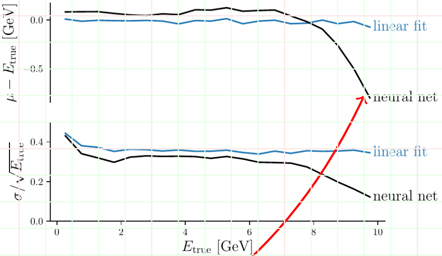
binned statistics





First Results

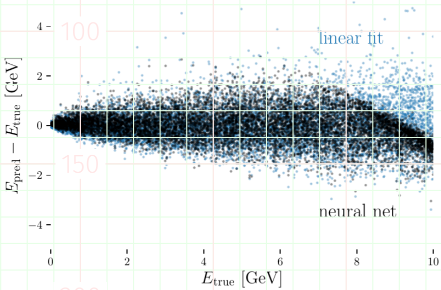


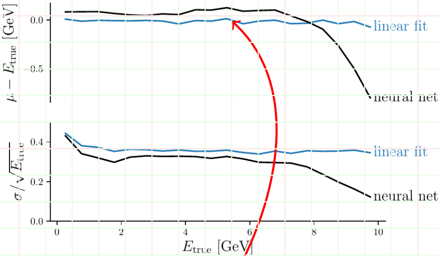


Problems:

- kink in distribution
- shift to higher values

First Results



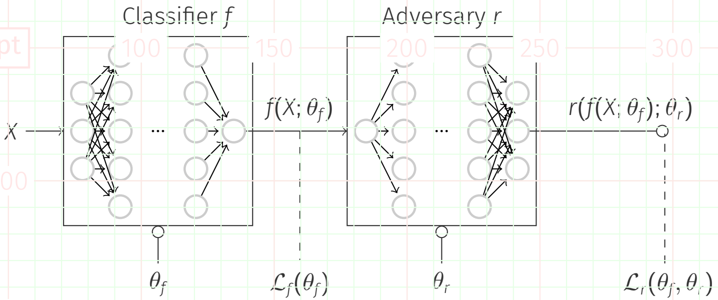


Problems:

- kink in distribution
- shift to higher values

Conv Setup

Adversarial Training



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

Procedure:

- Pretrained Classifier
- in Adversarial with $\mathcal{L}_r(\theta_f, \theta_r)$
- Train Classifier with $E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda\mathcal{L}_r(\theta_f, \theta_r)$
- repeat last two steps until convergence

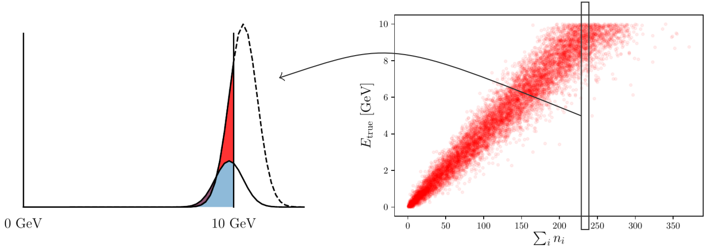
$\Rightarrow$ no stable results

Maximum Likelihood Loss

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

By using the mean squared error as a loss function we imply that our values are gaussian distributed with constand σ

Maximum Likelihood Loss

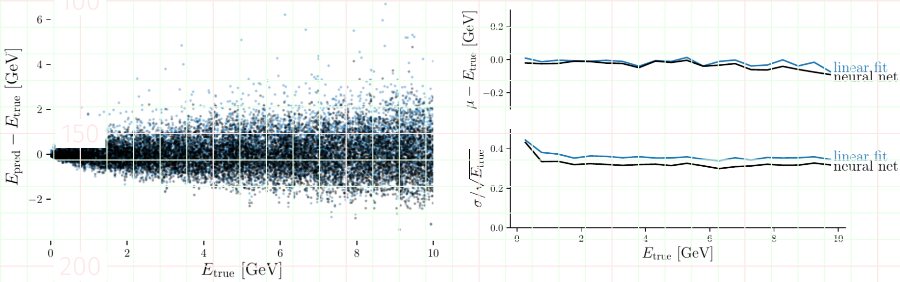


$$\text{max log likelihood} = -\min \sum \ln\left(\frac{\mathrm{Norm}(y_{\text{true}}|y_{\text{pred}},\sigma)}{\int_a^b \mathrm{Norm}(y_{\text{true}}|y_{\text{pred}},\sigma)}\right)$$

$$= \min \sum \underbrace{\frac{(y_{\text{true}} - y_{\text{pred}})^2}{\sigma^2}}_{\text{mean squared weighted error}} + \underbrace{\ln\left(\frac{\pi\sigma^2}{2}\left(\mathrm{erf}\left(\frac{y_{\text{pred}} - a}{\sqrt{2}\sigma}\right) - \mathrm{erf}\left(\frac{y_{\text{pred}} - b}{\sqrt{2}\sigma}\right)\right)^2\right)}_{\text{boundary correction term}}$$

Maximum Likelihood Loss

- Start with pretrained model
- Train network with loss from last slide and $\sigma = 0.31\sqrt{y_{\text{true}}}$



- Compensation for shift and boundary kink
- Network performs $\approx 10\%$ better than the linear fit

jets

Jet Physics

TODO: what is a jet

QCD Jet CMS Simulation

TODO: Simulation vorstellen TODO: Dataset vorstellen

Particle Flow Network

TODO: particle flow einführen

Daten vorstellen

pt 100 150 200 250 300 350

100

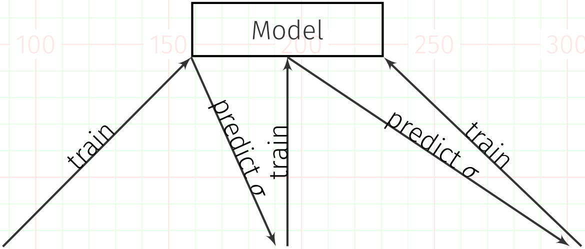
150

200

250

Binned Loss

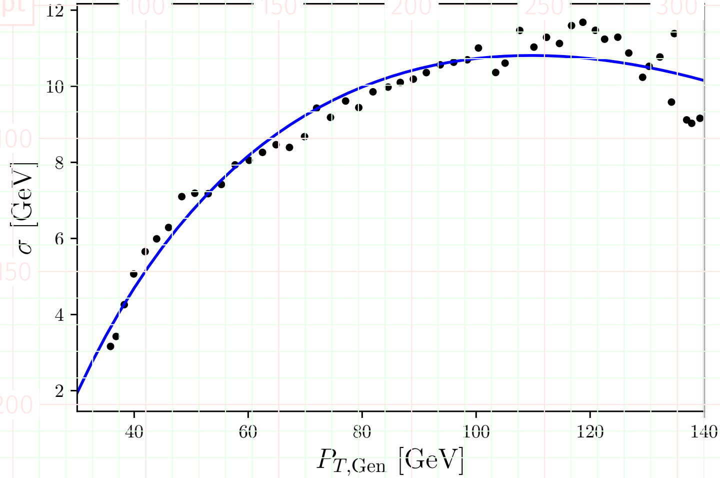
Training Setup



Loss:	$(\hat{Y} - Y)^2$	$\mathcal{L}_J(\hat{Y}, Y, \sigma)$	$\mathcal{L}_J(\hat{Y}, Y, \sigma)$
	MSE	1st	2nd

$$\mathcal{L}_J = \text{ binned } \underbrace{\frac{(y_{\text{true}} - y_{\text{pred}})^2}{\sigma^2}}_{\color{blue}{\text{mean squared weighted error}}} \underbrace{+ \ln\left(\frac{\pi\sigma^2}{2}\left(\operatorname{erf}\left(\frac{y_{\text{pred}} - a}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{y_{\text{pred}} - b}{\sqrt{2}\sigma}\right)\right)^2\right)}_{\color{red}{\text{boundary correction term}}}$$

Predict σ



Summary and Outlook

Summary:

- Simulation of a calorimeter
- Presentation of two problems
- Adversarial Training
- Develop a customized loss function

Outlook:

- Simulation of pions and other particles
- Compare different architectures and hyperparameter tuning
- Energy calibration for jets

Summary and Outlook

Thank you for your attention

Summary:

- Simulation of a calorimeter
- Presentation of two problems
- Adversarial Training
- Develop a customized loss function

Outlook:

- Simulation of pions and other particles
- Compare different architectures and hyperparameter tuning
- Energy calibration for jets