technische universität
dortmund

lehrstuhl
physik e5

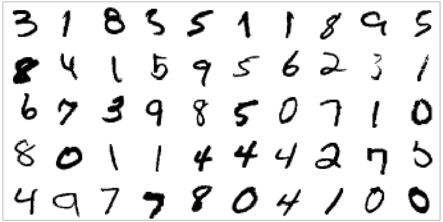
Deep Learning in Astroparticle Physics exemplified by the Reconstruction of Muon-Neutrino Events in IceCube

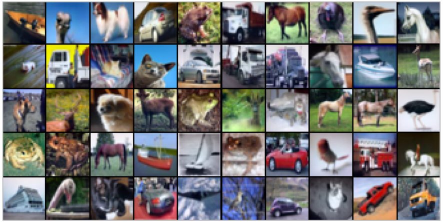
Mirco Hünnefeld

21.02.2017

Why use Deep Learning in Astroparticle-Physics?

- **MNIST**
  - 10 classes
  - 70000 images
- **CIFAR10** / **CIFAR100**
  - 32x32 images
  - 10/100 classes
  - 60000 images
- **ImageNet 2012**
  - 1000 classes
  - 1.3 million images

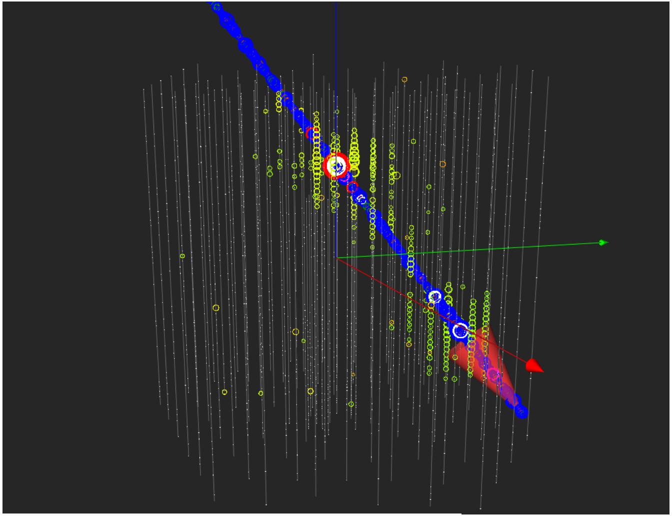




Source: *Daniele Ciriello [1]*

Why use Deep Learning in Astroparticle-Physics?

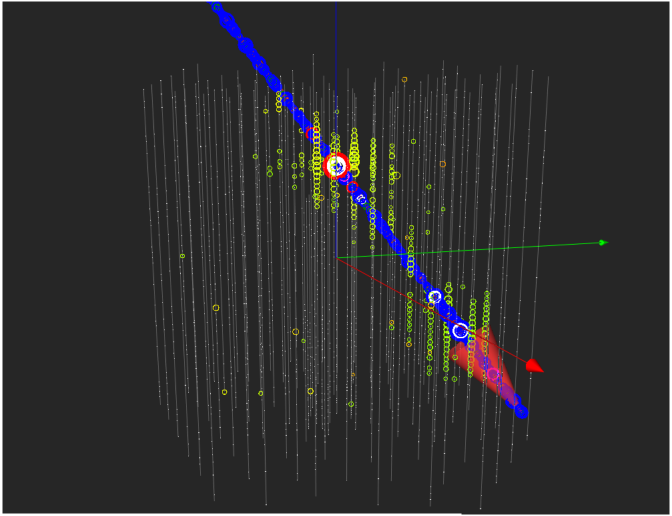
- **MNIST**
  - 10 classes
  - 70000 images
- **CIFAR10 / CIFAR100**
  - 32x32 images
  - 10/100 classes
  - 60000 images
- **ImageNet 2012**
  - 1000 classes
  - 1.3 million images
- **Astroparticle-Physics**
  - Only a handful classes
  - Millions of MC events



Why use Deep Learning in Astroparticle-Physics?

- **MNIST**
  - 10 classes
  - 70000 images
- **CIFAR10 / CIFAR100**
  - 32x32 images
  - 10/100 classes
  - 60000 images
- **ImageNet 2012**
  - 1000 classes
  - 1.3 million images
- **Astroparticle-Physics**
  - Only a handful classes
  - Millions of MC events

→ **Datasets are a lot bigger**



Why use Deep Learning in Astroparticle-Physics?

Dataset Size

- Labeled MC-data is comparatively inexpensive to obtain
- Datasets are bigger
- Reduces danger of overfitting

Generalization

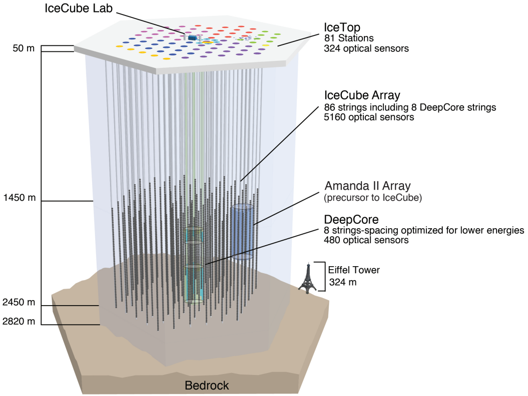
- Big Problem in traditional image recognition
- Monte-Carlo-Simulation:
  - Goal: complete description of detector
  - Generalization is not really a problem

→ **These bottlenecks do not exist in Astroparticle-Physics**

How can we use Deep Learning in Astroparticle-Physics?

(Example: Reconstruction of Muon-Neutrino Events in IceCube)

IceCube Neutrino Observatory



Source: *IceCube Collaboration [3]*

- Particle detector at the South Pole
- Detects Cherenkov radiation of secondary particles
- 86 vertical Strings
- Per String: 60 Digital Optical Modules (DOMs)
- Per DOM: Pulse series
- Amplitude and time for each pulse

→ 5160 Pulse series with variable length

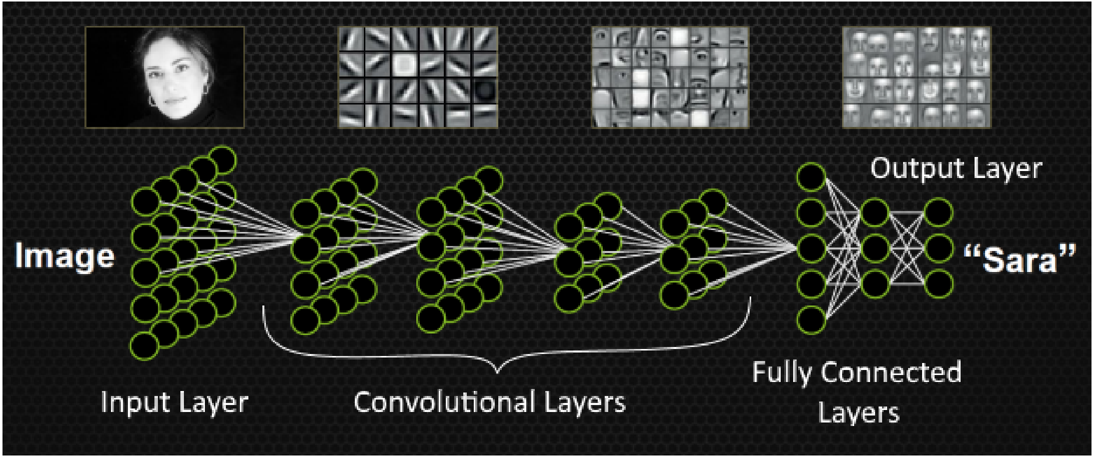
Why use Deep Learning?

- Online Processing of selected events
- Limited Resources available at the South Pole
- Restricted to basic and fast algorithms
- Raw/ low-level Data: hard to handle

Module	Mean Time	σ	99% percentile
SPE 2-it. Fit	0.085 s	0.129 s	0.670 s
MPE Fit	0.054 s	0.102 s	0.485 s
Cramer Rao Fits	0.049 ms	0.133 ms	0.560 ms
Bayesian Fit	1.022 ms	5.825 ms	0.024 s
Split Fits	0.066 s	0.236 s	1.291 s
MuEx	6.576 ms	0.023 s	0.091 s
Truncated Energy	2.612 ms	6.204 ms	0.022 s
Paraboloid	0.014 s	0.107 s	0.311 s
SplineMPE	0.036 s	0.152 s	0.793 s
All modules (status quo)	0.237 s	0.473 s	2.523 s
Status quo + SplineMPE	0.273 s	0.605 s	3.297 s

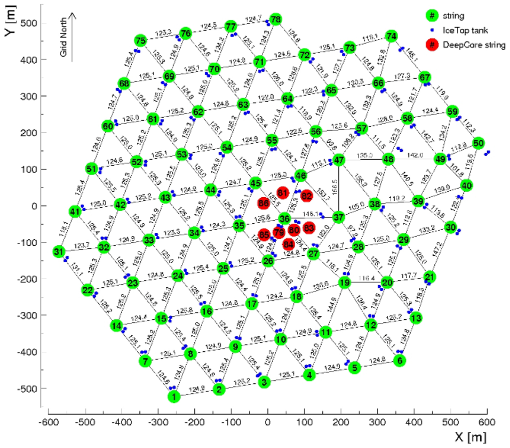
Source: *2014 TFT Proposal [5]*

Image Recognition with Convolutional Networks

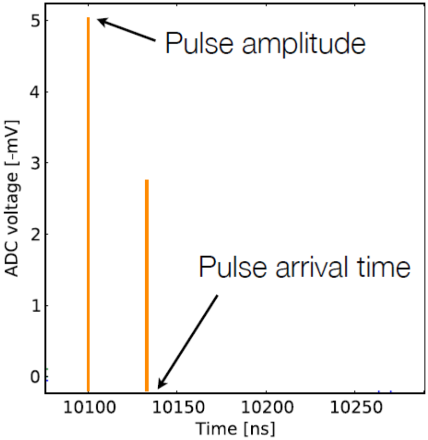


Source: *Honglak Lee u. a./ cuDNN [4]*

Deep Learning in IceCube



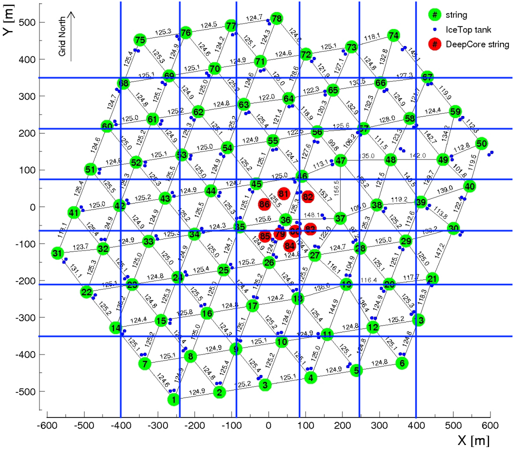
Source: *IceCube/Kurt Woschnagg [6]*



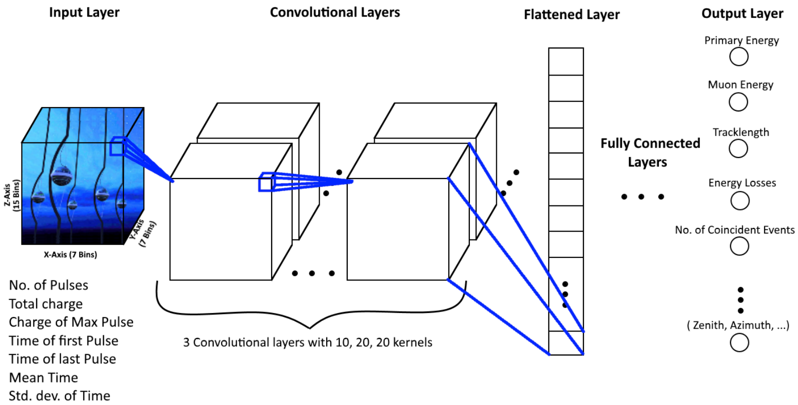
Source: *J. van Santen*

Basic Model

- Proof of concept
- Divide IceCube into coarse bins
- Summary information per bin:
    - Number of pulses
    - Total charge
    - Charge of highest pulse
    - Time of first Pulse
    - Time of last Pulse
    - Average time of Pulses
    - Standard deviation of pulse times

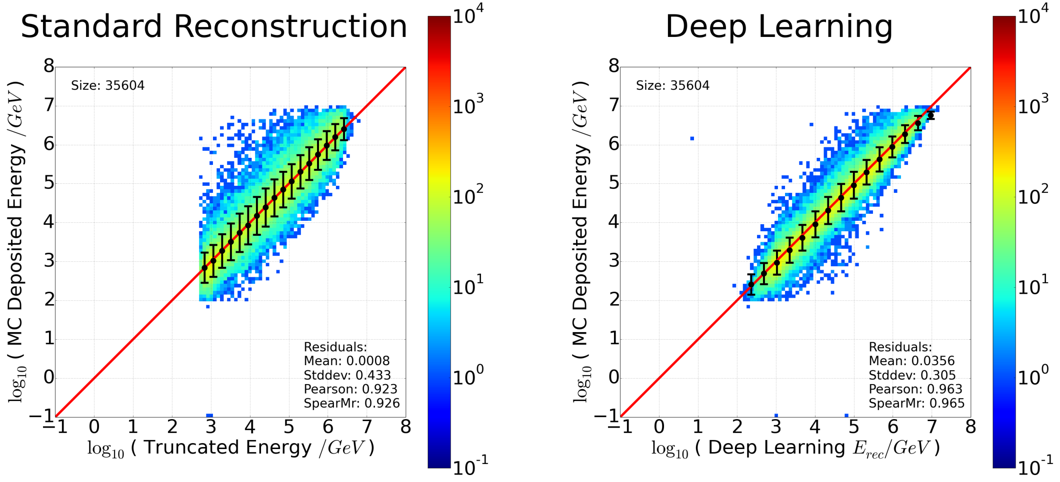


Basic Model

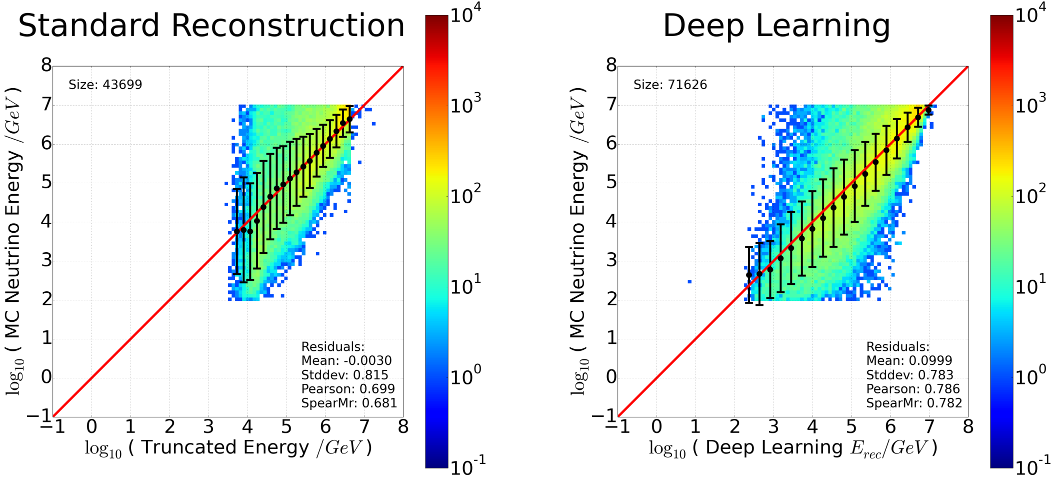


Time needed for prediction: 0.4 ms/Event

Basic Model – Deposited Energy



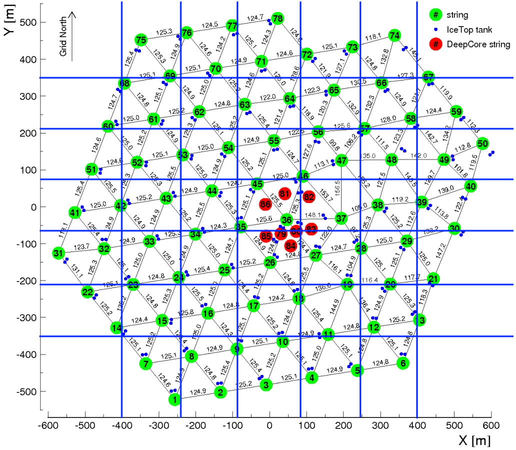
Basic Model – Primary Energy



More Complex Model

Basic model

- Proof of concept: Deep Learning seems applicable and promising
  → Ready for more appropriate Model



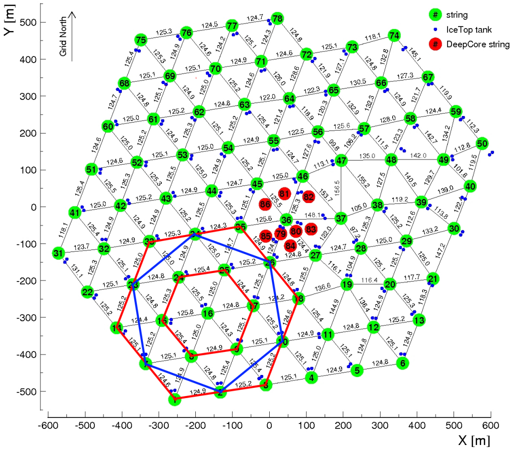
More Complex Model

Basic model

- Proof of concept: Deep Learning seems applicable and promising
  → Ready for more appropriate Model

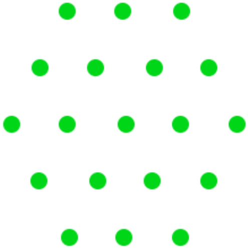
More complex model

- Every DOM as a bin/pixel
- Hexagonally shaped Input
- Hexagonally shaped convolution kernels



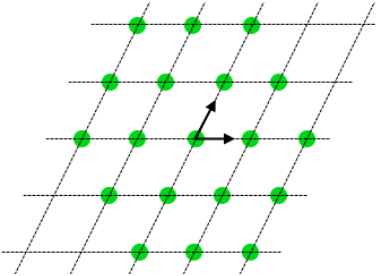
→ **How to deal with hexagonal shape?**

How to deal with hexagonal shape?



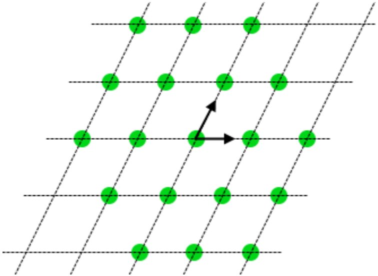
How to deal with hexagonal shape?

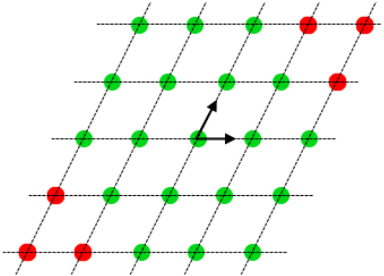
- Add coordinate system



How to deal with hexagonal shape?

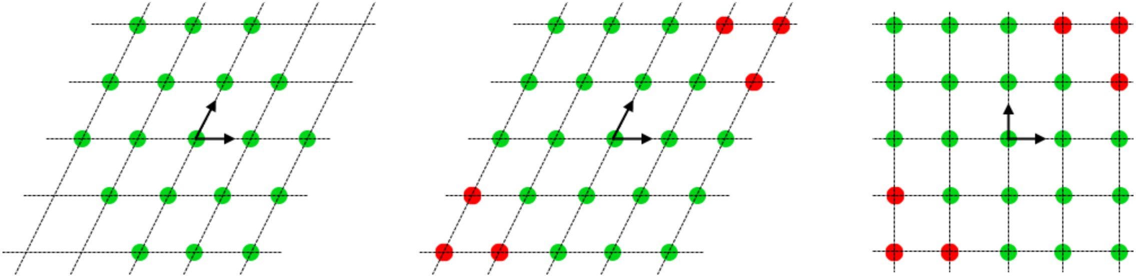
- Add coordinate system
- Pad with zeros





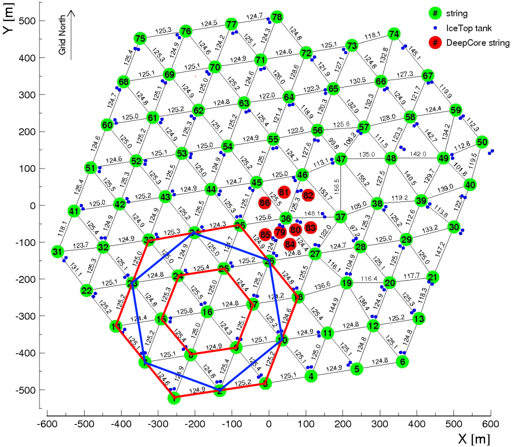
How to deal with hexagonal shape?

- Add coordinate system
- Pad with zeros
- Align rows

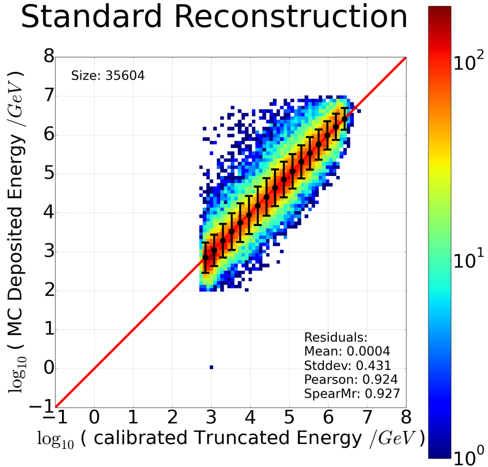


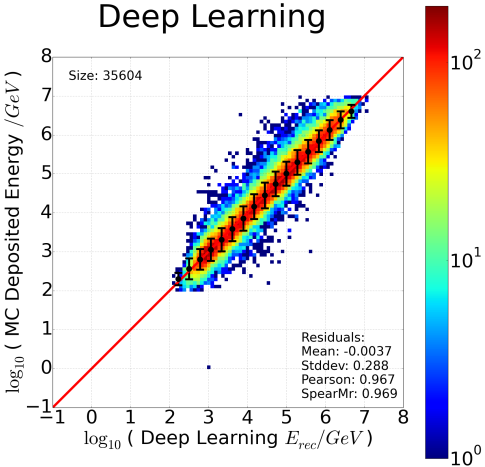
More Complex Model

- Hexagonally shaped Input
- Hexagonally shaped convolution kernels
- Every DOM is now a bin/pixel
- 17 values per DOM:
  number of pulses, total charge, mean time, ...
- Architecture:
  - 5 convolutional layers over IC79 strings (green dots)
  - 4 convolutional layers over Deepcore strings (red dots)
  - Flattened layer combining convolutions over IC79 and Deepcore strings
  - 6 Fully connected layers
- Time needed for prediction:
  20 ms/Event

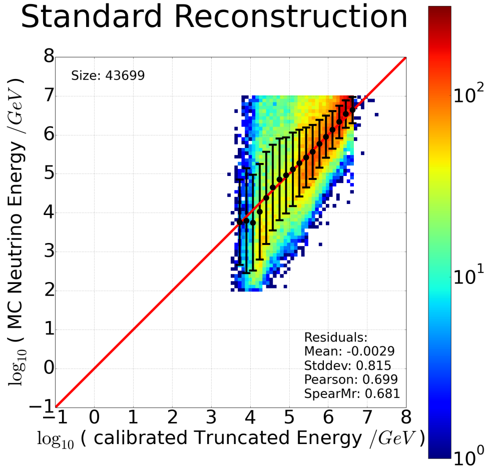


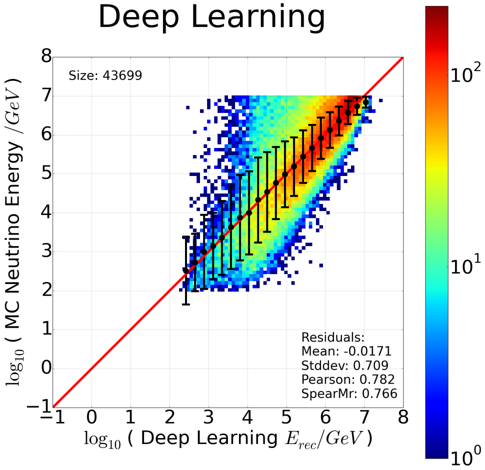
More Complex Model – Deposited Energy





More Complex Model – Primary Energy





Deep Learning – Runtime

- Current online filter:
  mean time: 273 ms/Event
  std. deviation: 605 ms/Event
- Reconstruction is fast
  basic model: 0.4 ms/Event
  complex model: 20 ms/Event
- Reconstruction is a set number of mathematical operations: runtime is the same for all events

Module	Mean Time	σ	99% percentile
SPE 2-it. Fit	0.085 s	0.129 s	0.670 s
MPE Fit	0.054 s	0.102 s	0.485 s
Cramer Rao Fits	0.049 ms	0.133 ms	0.560 ms
Bayesian Fit	1.022 ms	5.825 ms	0.024 s
Split Fits	0.066 s	0.236 s	1.291 s
MuEx	6.576 ms	0.023 s	0.091 s
Truncated Energy	2.612 ms	6.204 ms	0.022 s
Paraboloid	0.014 s	0.107 s	0.311 s
SplineMPE	0.036 s	0.152 s	0.793 s
All modules (status quo)	0.237 s	0.473 s	2.523 s
Status quo + SplineMPE	0.273 s	0.605 s	3.297 s

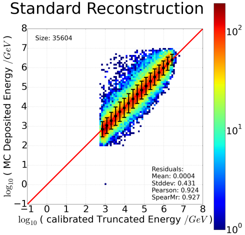
Source: *2014 TFT Proposal [5]*

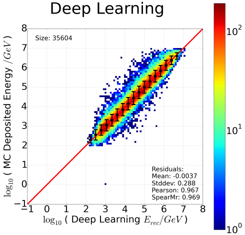
Summary

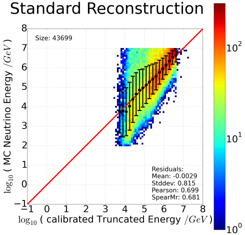
- Typical bottlenecks in image classifcation do not apply to physics
  → This includes the lack of labeled data and the difficulty of generalization
- Deep Learning can handle low-level data
- Runtime: fast and very stable
  → Well suited for applications with limited resources
- Reconstruction of Muon-Neutrino Events in IceCube:
  Improvements in energy reconstruction compared to current
  online reconstructions while reducing runtime

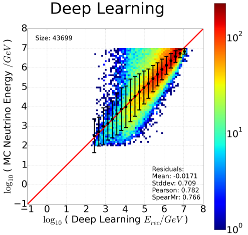
Outlook

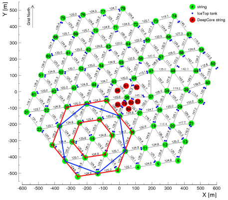
- Fully include time dimension and perform 4D convolution
- Focus on track reconstruction: azimuth and zenith
- Optimize hyperparameters

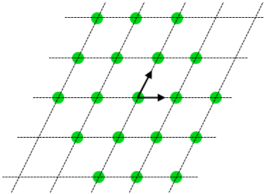

Standard Reconstruction
Size: 35604
Residuals:
Mean: 0.0004
Stddev: 0.431
Pearson: 0.924
SpearMr: 0.927
$\log_{10}$ (MC Deposited Energy $/GeV$)
$\log_{10}$ (calibrated Truncated Energy $/GeV$)

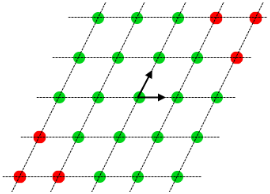

Deep Learning
Size: 35604
Residuals:
Mean: -0.0037
Stddev: 0.288
Pearson: 0.967
SpearMr: 0.969
$\log_{10}$ (MC Deposited Energy $/GeV$)
$\log_{10}$ (Deep Learning E_{rec}/GeV)

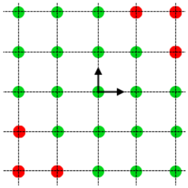

Standard Reconstruction
Size: 43699
Residuals:
Mean: -0.0029
Stddev: 0.815
Pearson: 0.699
SpearMr: 0.681
$\log_{10}$ (MC Neutrino Energy $/GeV$)
$\log_{10}$ (calibrated Truncated Energy $/GeV$)


Deep Learning
Size: 43699
Residuals:
Mean: -0.0171
Stddev: 0.709
Pearson: 0.782
SpearMr: 0.766
$\log_{10}$ (MC Neutrino Energy $/GeV$)
$\log_{10}$ (Deep Learning E_{rec}/GeV)


string
IceTop tank
DeepCore string
Y [m]
X [m]
Grid North







Questions?

Bibliography I

[1] Daniele Ciriello. *CIFAR10*. Feb. 2017. URL: `https://dnlcrl.github.io/projects/2016/06/22/Deep-Residual-Networks-for-Image-Classification-with-Python+NumPy.html`.

[2] developer.apple.com. *Kernel Convolution*. Sep. 2016. URL: `https://developer.apple.com/library/content/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html`.

[3] IceCube/NSF. *The IceCube Neutrino Observatory*. Juni 2014. URL: `http://gallery.icecube.wisc.edu/web/var/albums/Press-Root-Album/Detector/ArrayWSeasonsLabelsAmanda.jpg?m=1371067618`.

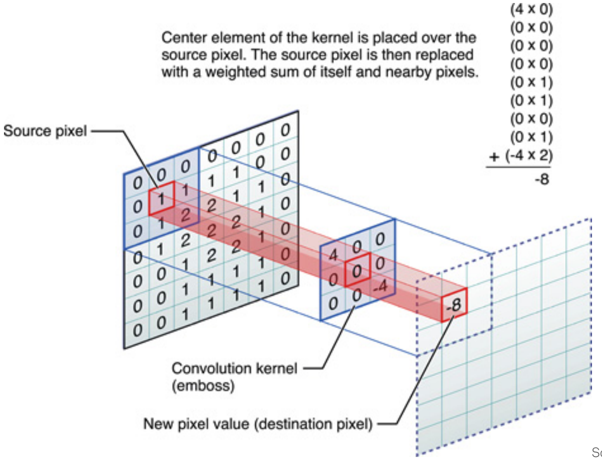
[4] Honglak Lee u. a. “Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML ’09. Montreal, Quebec, Canada: ACM, 2009, S. 609–616. ISBN: 978-1-60558-516-1. DOI: `10.1145/1553374.1553453`. URL: `http://doi.acm.org/10.1145/1553374.1553453`.

Bibliography II

[5] Markus Voge. *2014 TFT Proposal Request for a Level2 Online Muon Filter*. Nov. 2013. URL: https://docushare.icecube.wisc.edu/dsweb/Get/Document-67821/onlinel2proposal2014.pdf.

[6] IceCube/Kurt Woschnagg. *IceCube Surface Map*. Sep. 2016. URL: https://wiki.icecube.wisc.edu/index.php/File:Distances.i86.jpg.

Convolutional Filter



Source: *developer.apple.com [2]*

Truncated Energy without Calibration