GPU Day 2020

The Future of Computing, Graphics and Data Analysis

20-21 10 2020

Comparison of Very Deep Learning performance on GPU and CPU

Olena Linnyk[1,2]

J. Pawlowski[1], Manjunath O.K. [1], J. Steinheimer[1], K. Zhou[1], H. Stöcker[1], K. Schmidt[3], T. L. Weber[2], I. Teetz[2]

[1]Frankfurt Institute for Advanced Studies (FIAS), Frankfurt am Main, Germany

[2]„milch & zucker" Talent Acquisition & Talent Management Company, Giessen, Germany

[3]Institute of Physics, University of Silesia, Poland

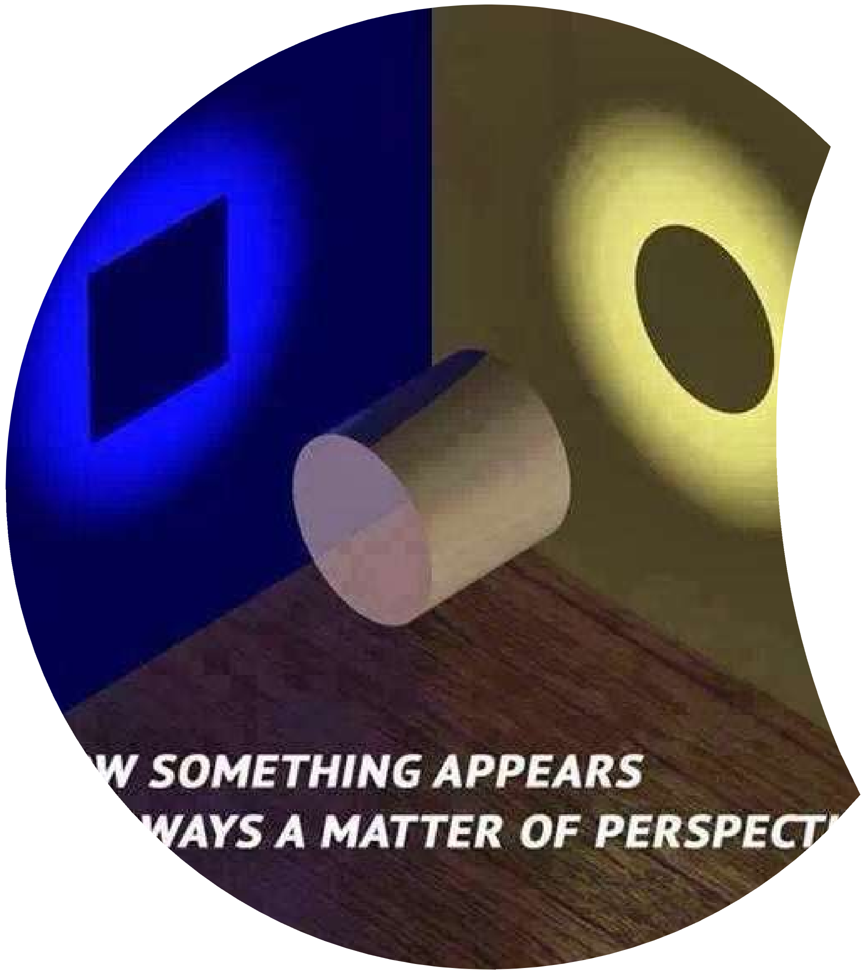
seriously creative

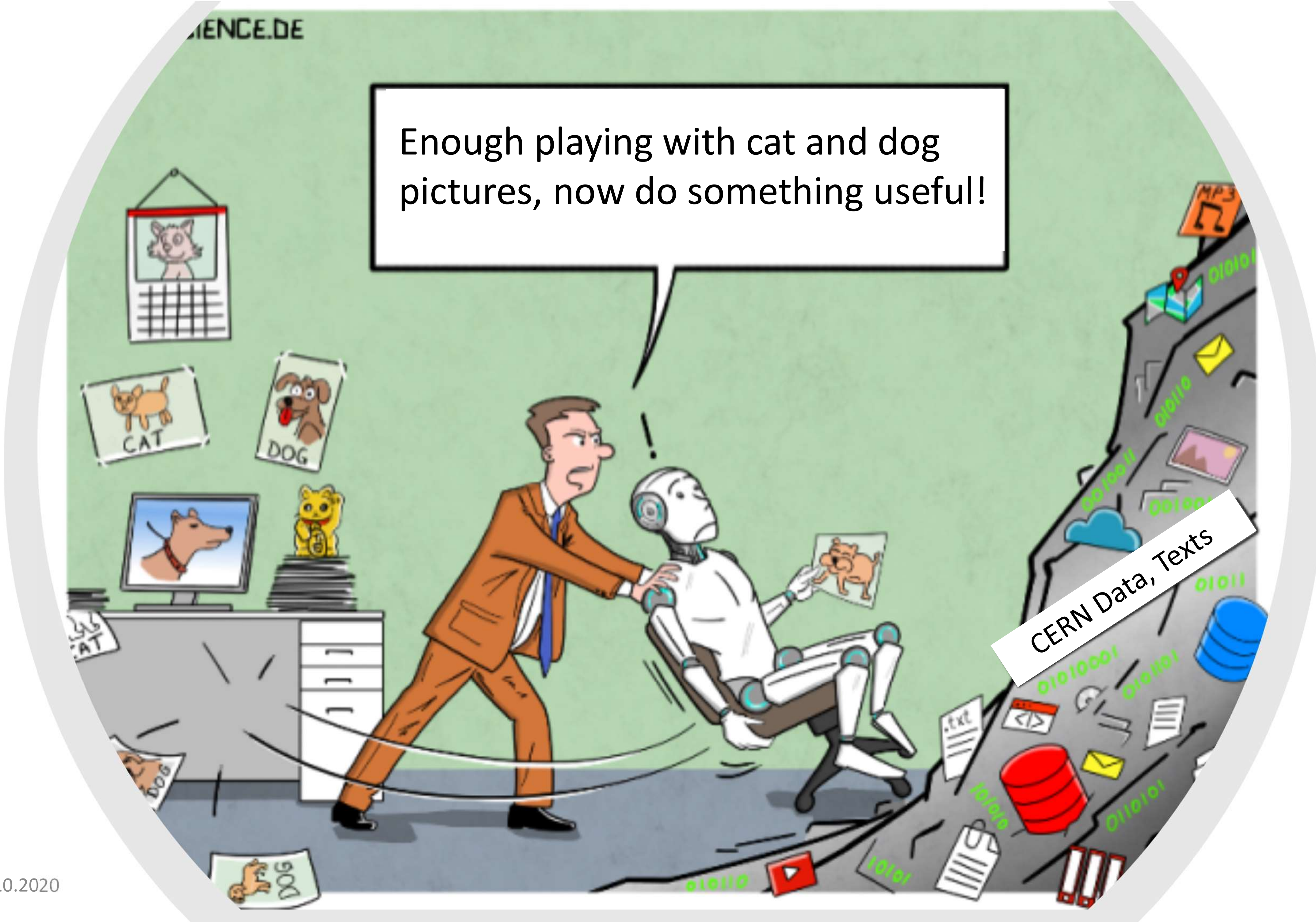
FIAS Frankfurt Institute for Advanced Studies

21.10.2020

creatively serious

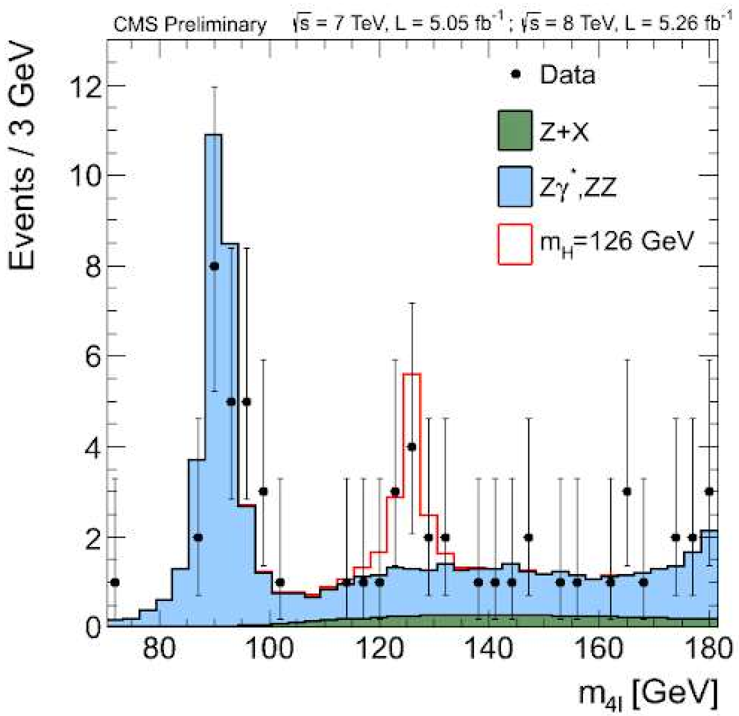
Classification = understanding

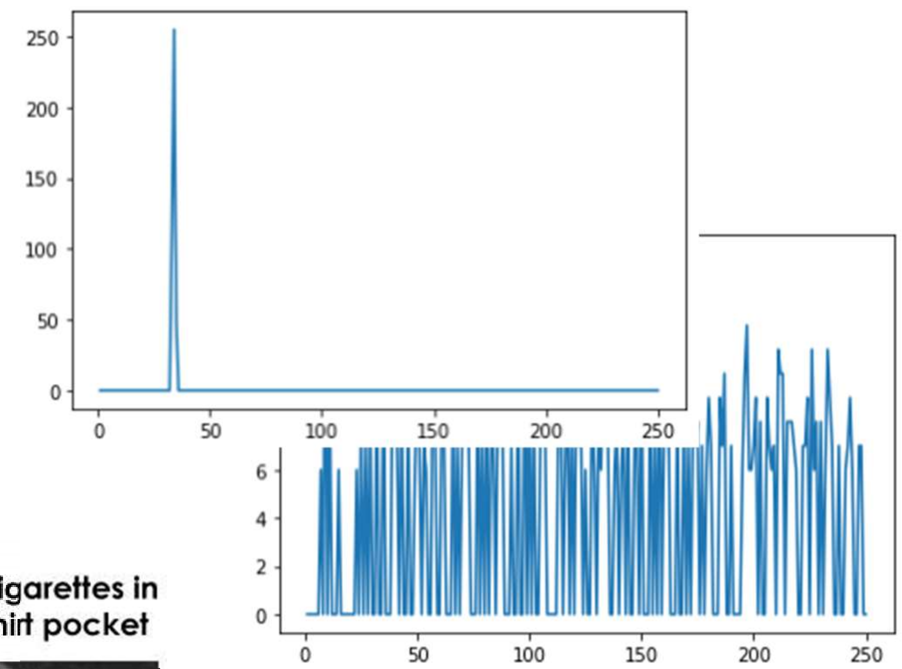




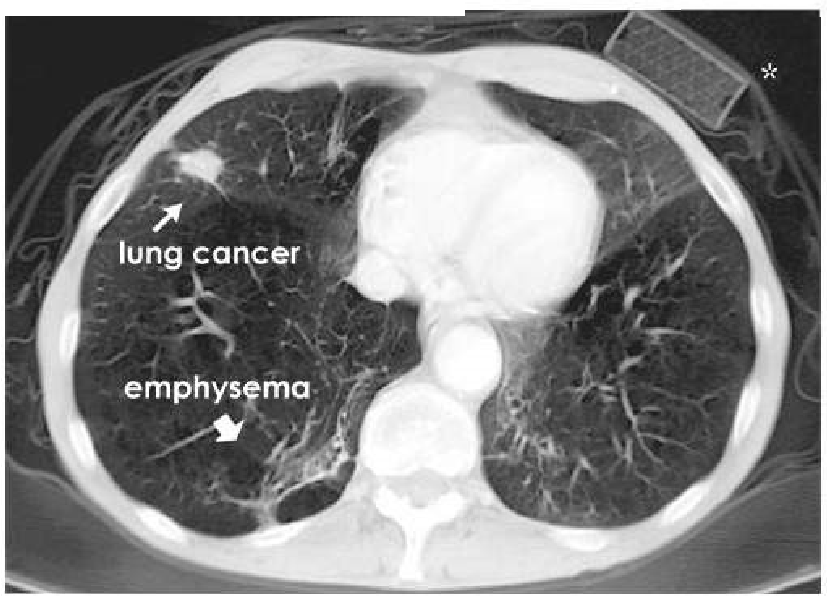
IENCE.DE
Enough playing with cat and dog pictures, now do something useful!
CAT
DOG
CERN Data, Texts

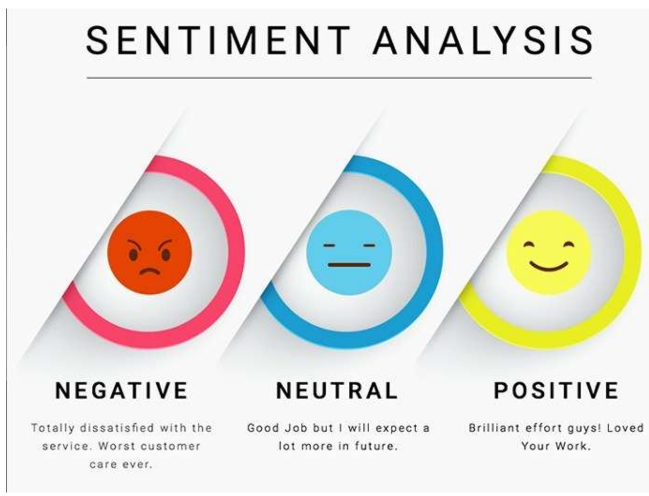
Some useful tasks

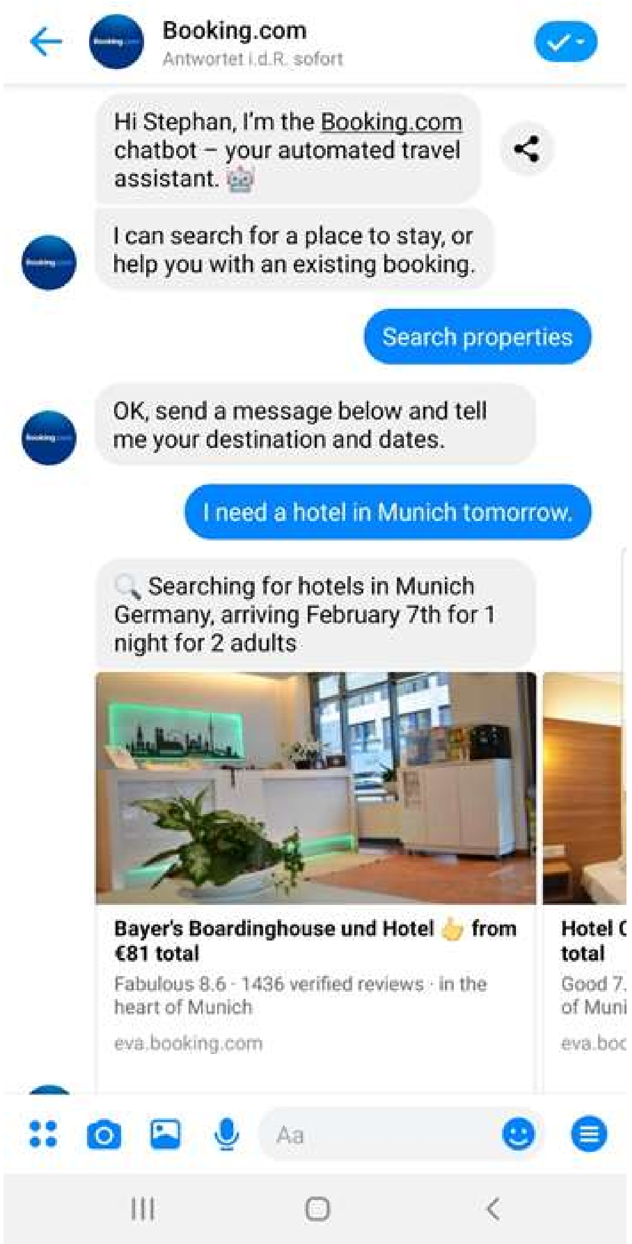




cigarettes in shirt pocket







We tested machine learning approaches ranging from desicion trees to deep neural networks

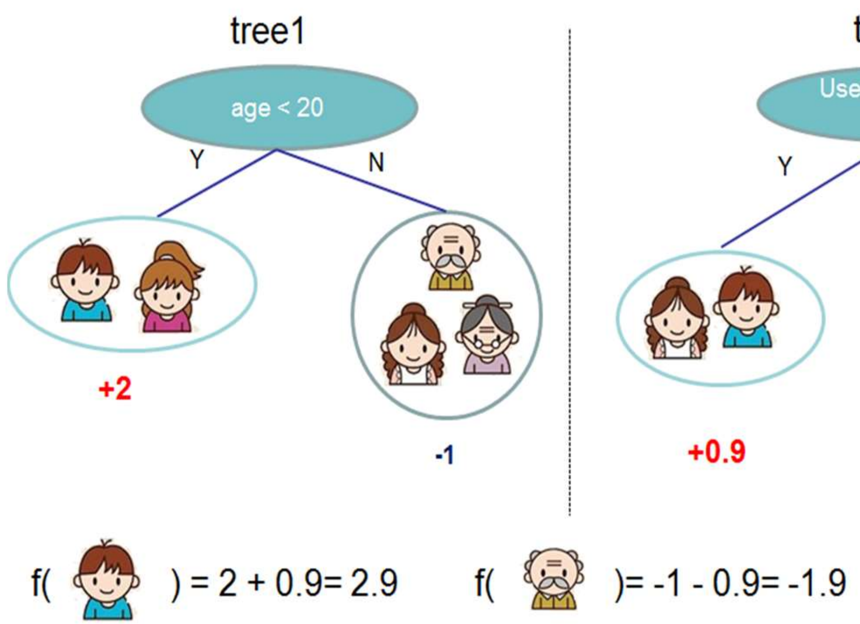
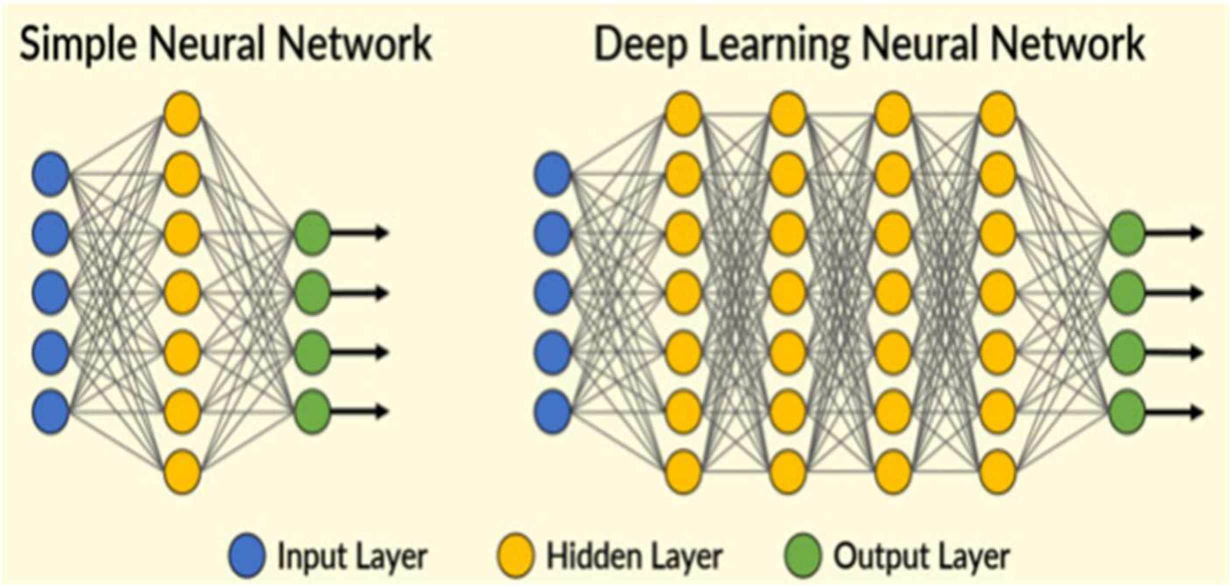
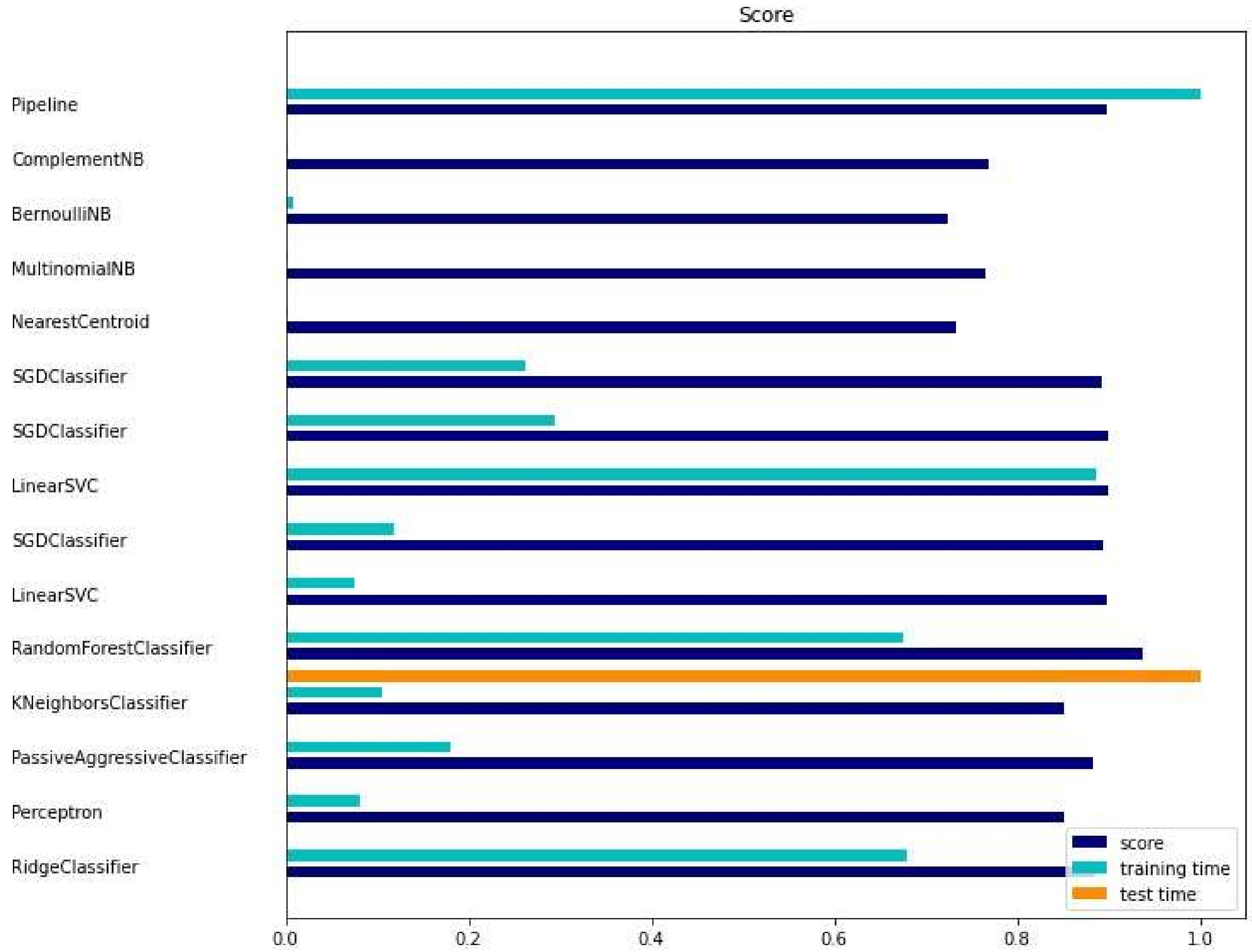


Image: https://xgboost.readthedocs.io



Best accuracy is achieved by the deep convolutional neural networks



	Model	Training Set Accuracy	Test Set Accuracy
4	Random Forest	0.998843	0.941393
0	Gradient Boosting	0.998843	0.937705
6	Multilayer Perceptron	0.977937	0.930738
2	Logistic Regression	0.912471	0.899180
5	SVM	0.890408	0.883197
1	KNN	0.998843	0.864344
3	Multinomial Naïve Bayes	0.757595	0.763934

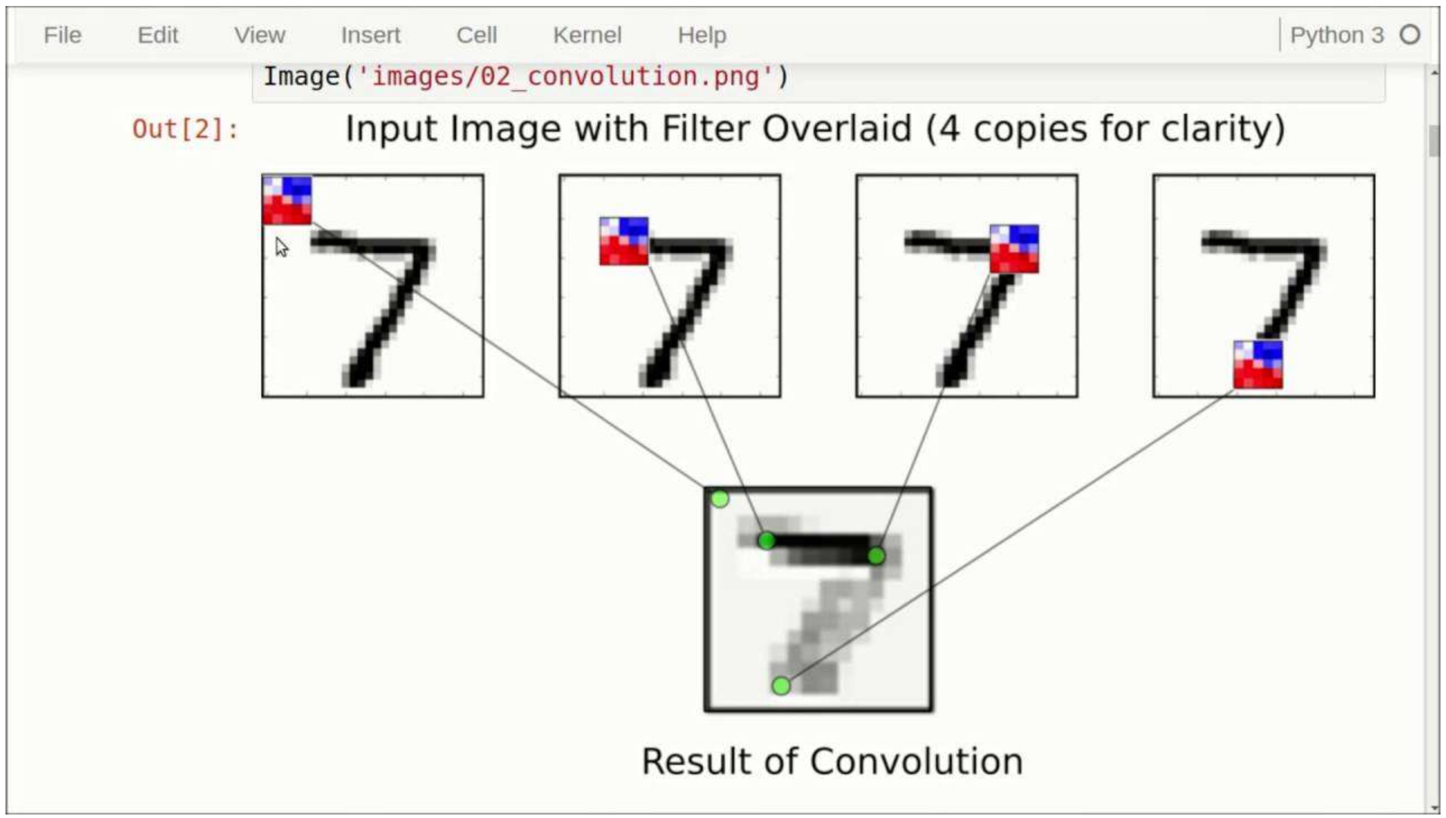
… … …

Convolutional Neural Net 97.8% on Test

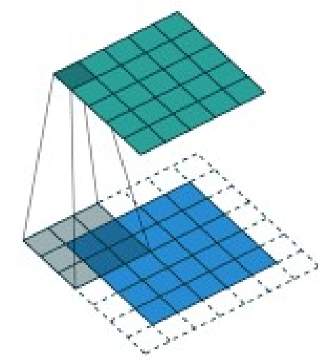
Example: Text classification on real life data from the web portal jobstairs.de © milch&zucker

Convolutional neural networks

Krizhevsky won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 competition with the brilliant deep convolutional neural networks. This was the first time this architecture was more successful that traditional, hand-crafted feature learning.



$$\underbrace{s}_{\text{Merkmalskatalog}}[t]=(\underbrace{x}_{\text{Input}}\star\underbrace{w}_{\text{Kernel}})[t]=\sum_{a} x[a]w[a+t]$$



Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. 2012.

Very deep convolutional networks suggested

“Previous very deep convolutional neural networks were trained on the giant ImageNet datasets. Small datasets like CIFAR-10 has rarely taken advantage of the power of depth since deep models are easy to overfit. By adding stronger regularizer and using Batch Normalization, very deep CNN can be used to fit small datasets with simple and proper modifications and don't need to re-design specific small networks.

Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014.

Better understanding -> More layers, more dimentions, more filters?

Vanishing gradients preventing the benefit of the depth

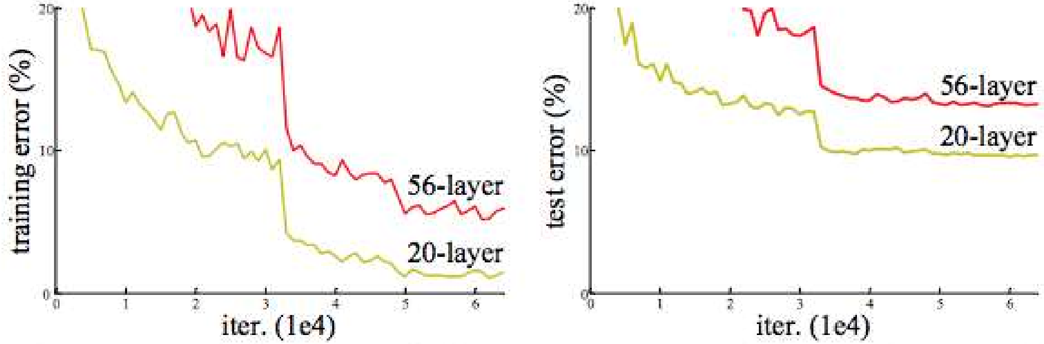
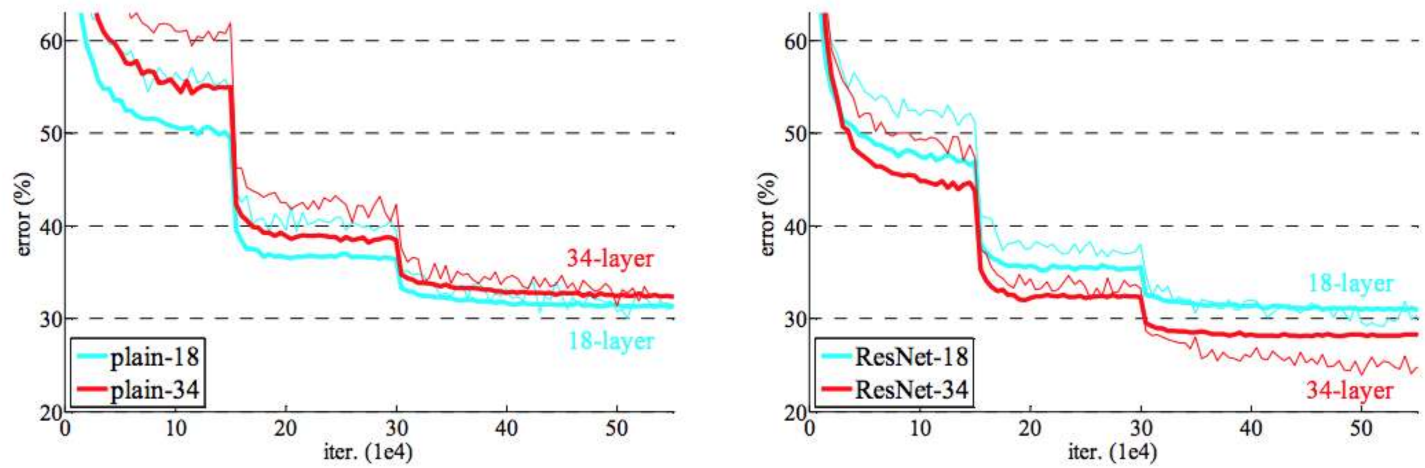


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

This Deeper networks do not lead to better accuracy on the test data sent, because when the network is too deep, the gradients from where the loss function is calculated easily shrink to zero after several applications of the chain rule. This result on the weights never updating its values and therefore, no learning is being performed.

Solution: ResNet

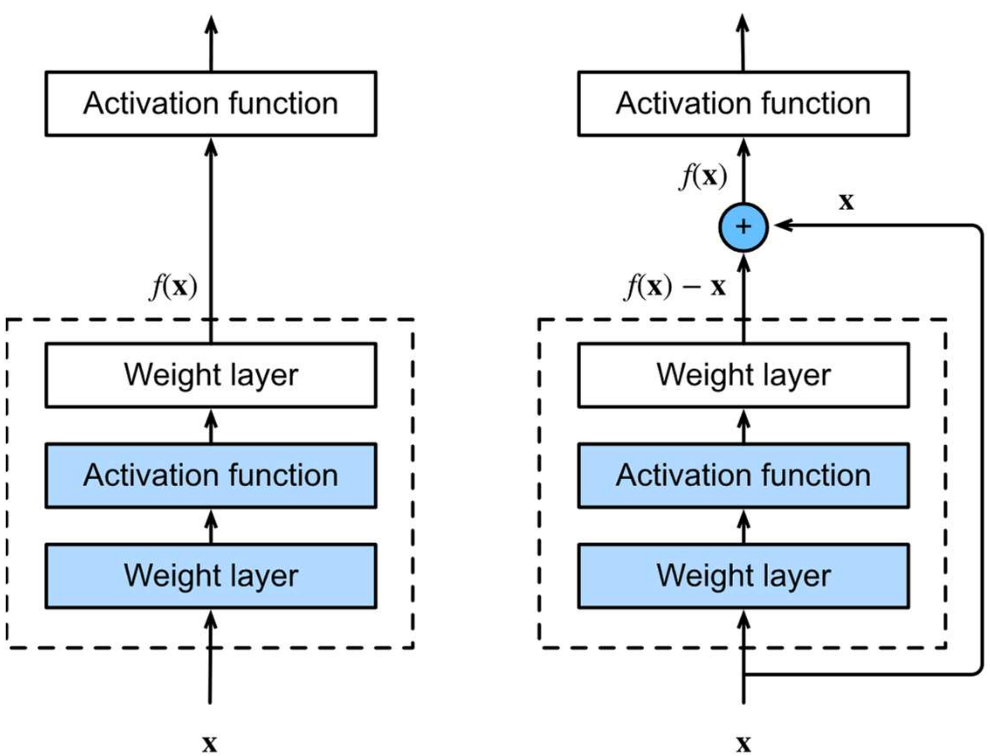
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. 2015.



ResNet

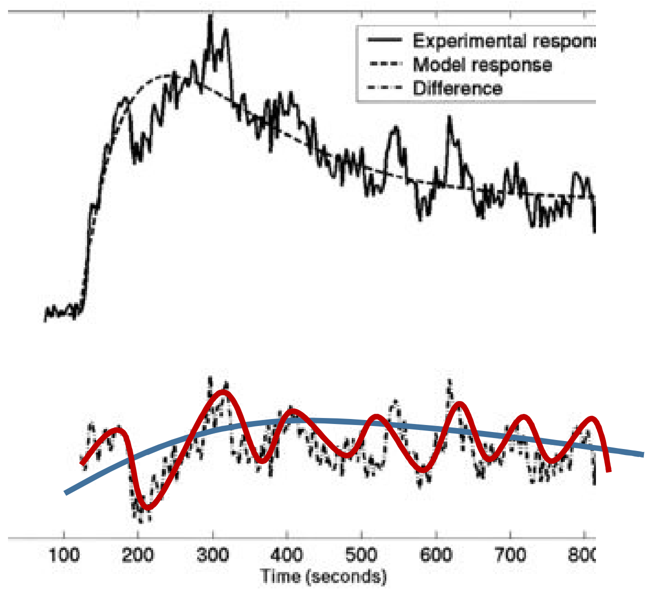
Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com



Effectively it means fitting f(x)-x in stead of f(x).

By adding several blocks, we fit first the main feature, then more detailes by fitting the residue of the function and the approximation in the second block etc.



The iterative approach prevents „jumping over“ the global optimum.

ResNet also applicable to the understanding of texts

Very Deep Convolutional Networks for Text Classification

Alexis Conneau
Facebook AI Research
aconneau@fb.com

Holger Schwenk
Facebook AI Research
schwenk@fb.com

Yann Le Cun
Facebook AI Research
yann@fb.com

Loïc Barrault
LIUM, University of Le Mans, France
loic.barrault@univ-lemans.fr

depth	without shortcut	with shortcut
9	37.63	40.27
17	36.10	39.18
29	35.28	36.01
49	37.41	36.15

Table 6: Test error on the Yelp Full data set for all depths, with or without residual connections.

Corpus:	AG	Sogou	DBP.	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
Method	n-TFIDF	n-TFIDF	n-TFIDF	ngrams	Conv	Conv+RNN	Conv	Conv
Author	[Zhang]	[Zhang]	[Zhang]	[Zhang]	[Zhang]	[Xiao]	[Zhang]	[Zhang]
Error	7.64	2.81	1.31	4.36	37.95*	28.26	40.43*	4.93*
[Yang]	-	-	-	-	-	24.2	36.4	-

Table 4: Best published results from previous work. Zhang et al. (2015) best results use a Thesaurus data augmentation technique (marked with an *). Yang et al. (2016)'s hierarchical methods is particularly adapted to datasets whose samples contain multiple sentences.

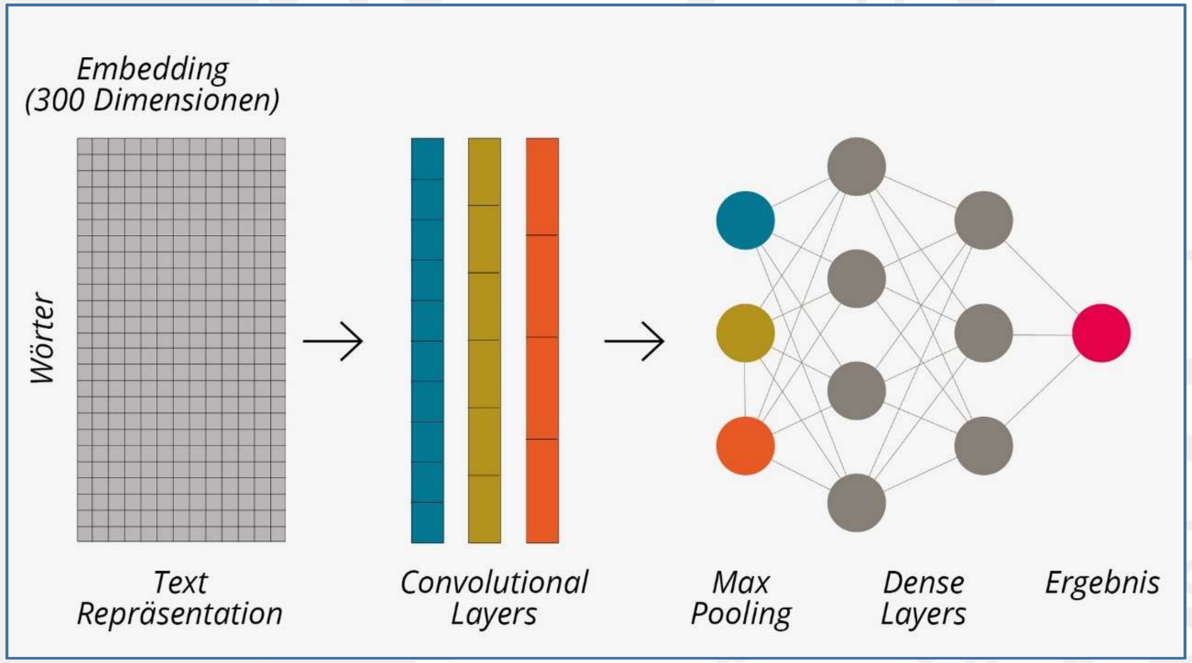
Depth	Pooling	AG	Sogou	DBP.	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
9	Convolution	10.17	4.22	1.64	5.01	37.63	28.10	38.52	4.94
9	KMaxPooling	9.83	3.58	1.56	5.27	38.04	28.24	39.19	5.69
9	MaxPooling	9.17	3.70	1.35	4.88	36.73	27.60	37.95	4.70
17	Convolution	9.29	3.94	1.42	4.96	36.10	27.35	37.50	4.53
17	KMaxPooling	9.39	3.51	1.61	5.05	37.41	28.25	38.81	5.43
17	MaxPooling	8.88	3.54	1.40	4.50	36.07	27.51	37.39	4.41
29	Convolution	9.36	3.61	1.36	4.35	**35.28**	27.17	37.58	**4.28**
29	KMaxPooling	**8.67**	**3.18**	1.41	4.63	37.00	27.16	38.39	4.94
29	MaxPooling	8.73	3.36	**1.29**	**4.28**	35.74	**26.57**	**37.00**	4.31

Table 5: Testing error of our models on the 8 data sets. No data preprocessing or augmentation is used.

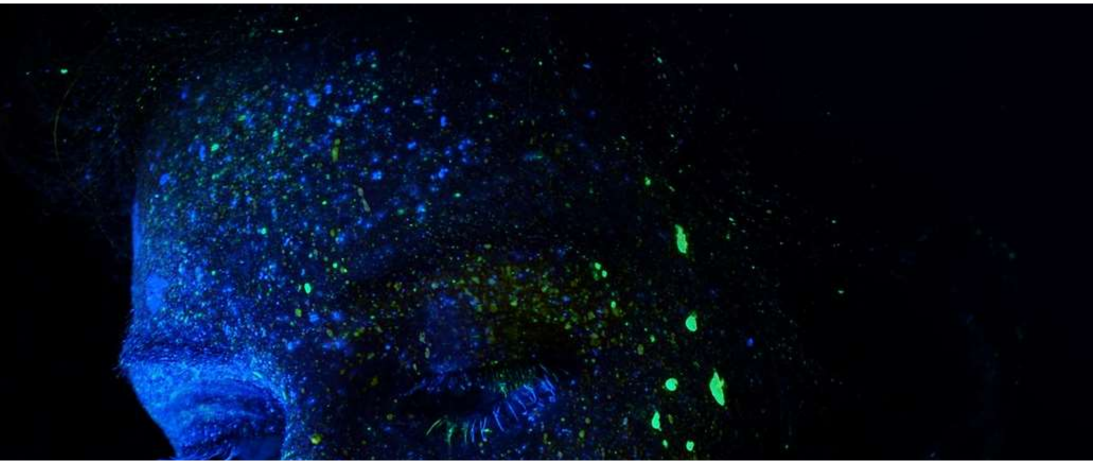
EXAMPLE:

PREDICTION OF CLICK RATES

Stellentitel	Score
Web-Marketing Controller (m/w/d)	0.217
Web-Marketing ControllerIn (m/w/d)	0.616
Online-Marketing ControllerIn (m/w/d)	0.680
ControllerIn Online-Marketing (m/w/d)	0.680
Marketing-ControllerIn (m/w/d) Online	0.777
ControllerIn (m/w/d) Online-Marketing	0.872
ControllerIn (m/w/x) Online-Marketing	0.960

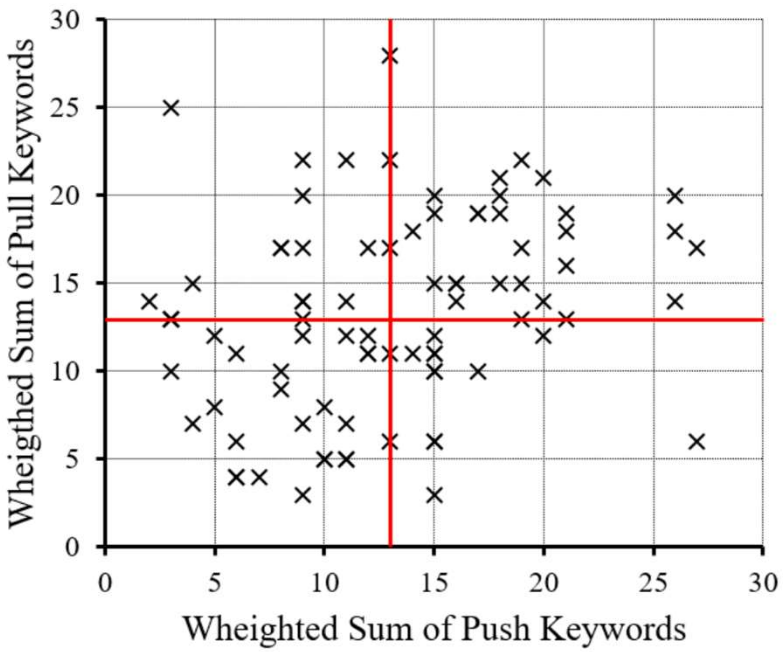


Web sites indeed perform better after the optimization



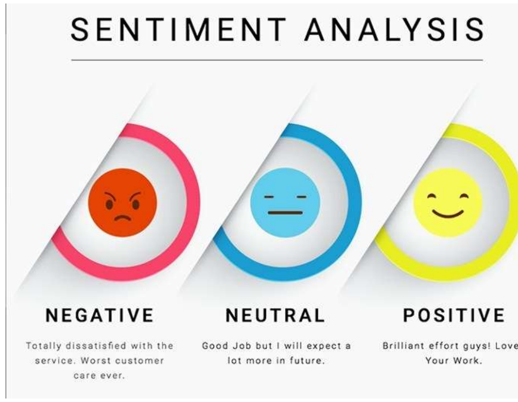
EXAMPLE 3: GENDER „SENTIMENT“

Hidden Bias



Stephan Böhm, Olena Linnyk, Jens Kohl, Tim Weber, Ingolf Teetz, Katarzyna Bandurka, and Martin Kersting. 2020. Analysing Gender Bias in IT Job Postings: A Pre-Study Based on Samples from the German Job Market. In *Proceedings of the 2020 Computers and People Research Conference (SIGMIS-CPR '20), June 19–21, 2020, Nuremberg, Germany*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3378539.3393862

Sertain key words were defined in job ads to influence the text in the direction of the gender typical description, which decreases the chance of especially female job seakers to apply for the job.



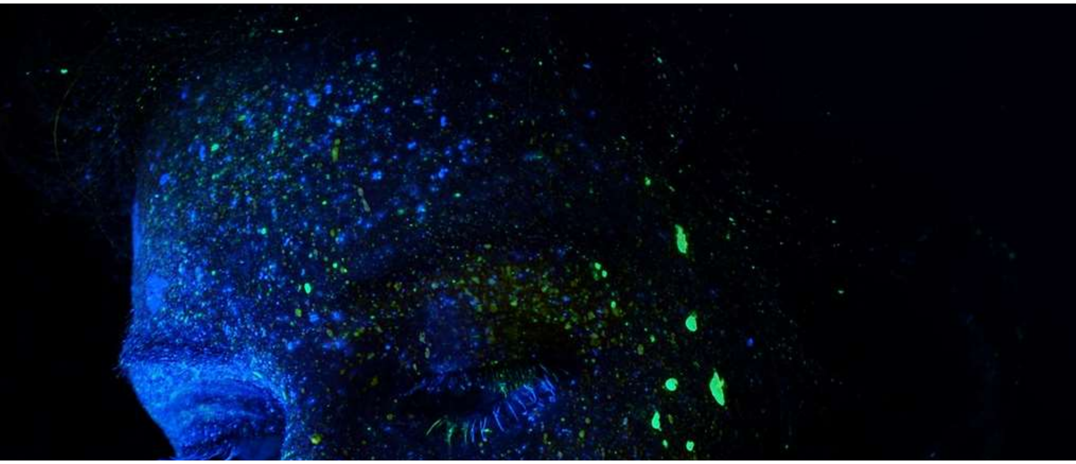
Analysing Gender Bias in IT Job Postings: A Pre-Study Based on Samples from the German Job Market

Stephan Böhm
RheinMain University
of Applied Sciences
Wiesbaden, Germany
stephan.boehm@hs-rm.de

Olena Linnyk*
Jens Kohl
Tim Weber
Ingolf Teetz
milch & zucker AG
Gießen, Germany
olena.linnyk@milchundzucker.de

Katarzyna Bandurka
Martin Kersting
Justus Liebig University of Gießen
Gießen, Germany
martin.kersting@psychol.uni-giessen.de

MILCH & ZUCKER, 2019
KI IN HR



EXPERIEMENT:

Randomized Single-Blind A/B-Test

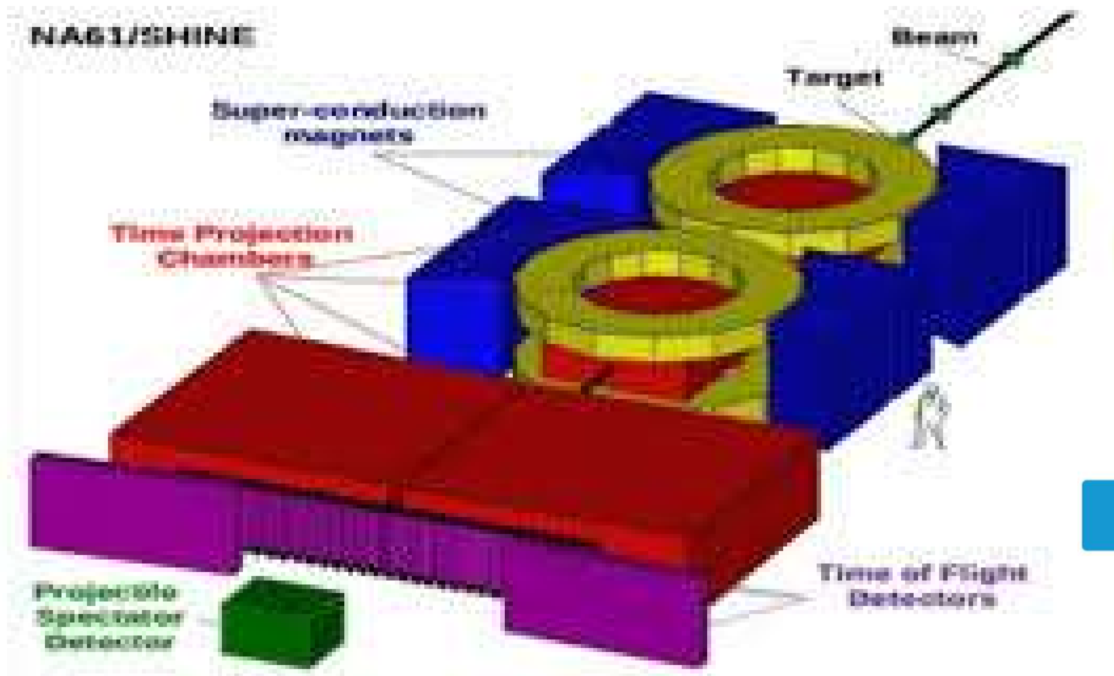
- Originaltext vs. Better-Ad Text
- Plattform: JobStairs.de
- More than 50 Job ads from various entry levels and industries

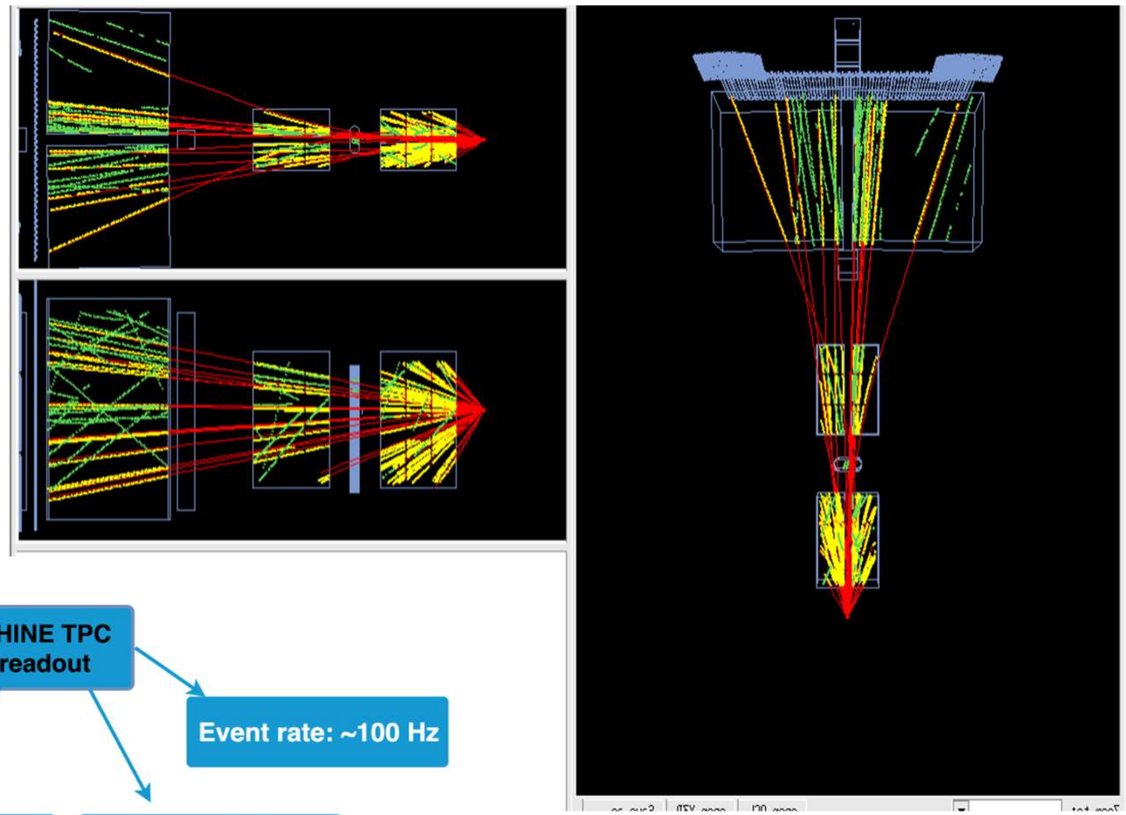
  ca. 1000 visitors

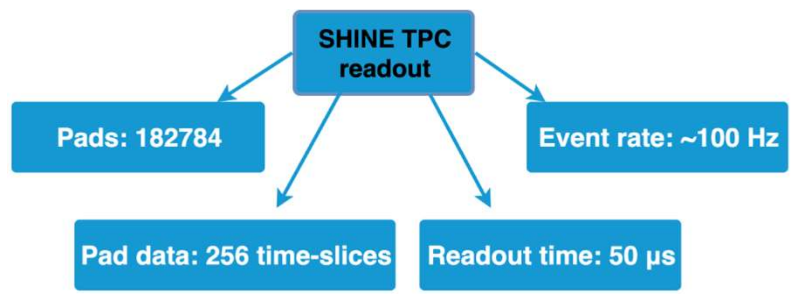
  Analyse: Web Tracking

Performance critical application: Experiment NA61/SHINE at CERN

- Noisy clusters: Up to 70%
  decision has to be done in real time (online): faster than 1 millisecond/event
  maximum 3 sec / node / event
  CPU?
  GPU?







Work in progress: O. Linnyk, W. Bryliński, M. Gaździcki, N. Davis, A. Rybicki

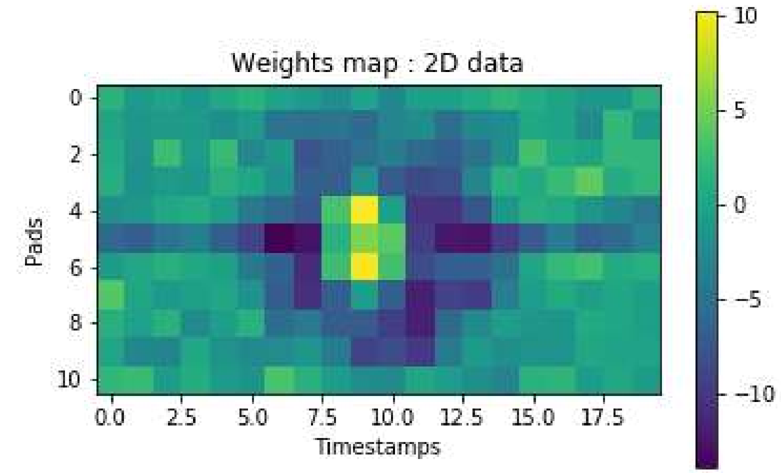
Machine learning for cluster classification

1. 10613825 samples
2. 74% noise, 26% signal (inbalanced data)
3. 28763.75 average number of clusters in the event (important to calculate the performance time)
4. 80% | 20% train - test split

We care about the performance time, therefore we present the hardware on which the data was processed:

GPU: GeForce RTX 2080 VENTUS 8G OC
CPU1: Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz

CPU2: Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz



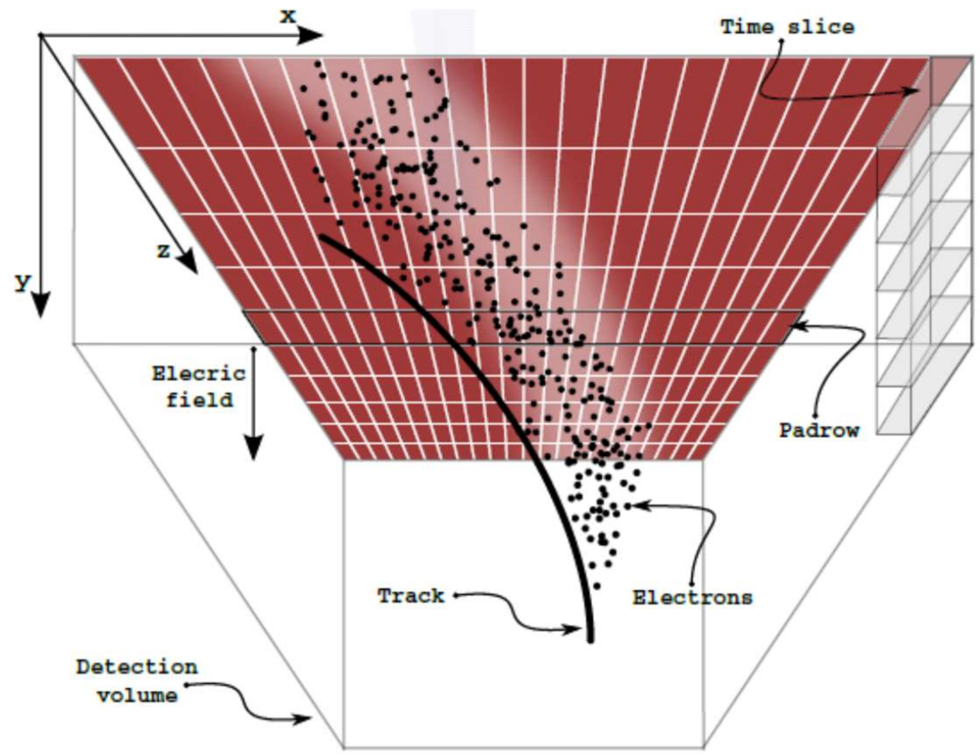
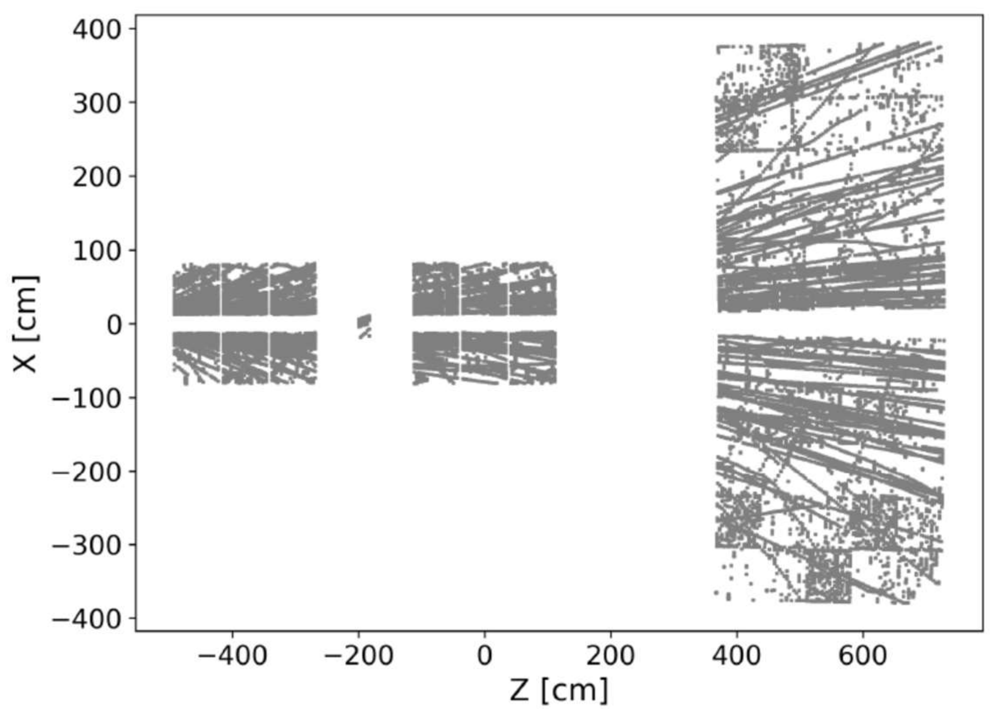
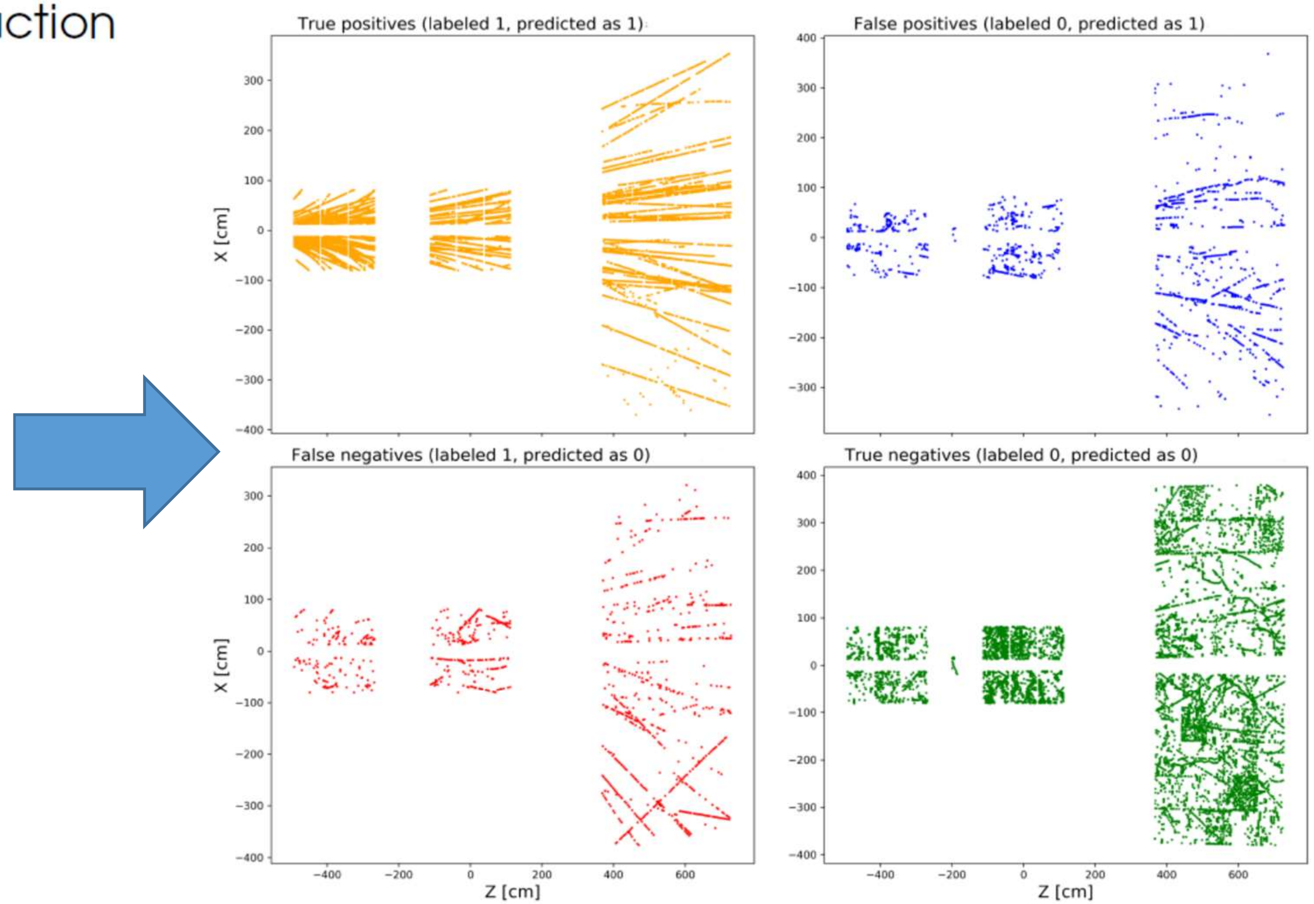


Figure 2. Simplified illustration of TPC working principle.

Tracking algorithm (offline) to label. Convusion matrix as a result

Our goal is to separate noise clusters from the clusters which form the track (signal) before the reconstruction using the Machine Learning techniques.





Which machine learning method to chose?

Data for all groups come from the same chunk: Ar + Sc at 30GeV (run ....)

Test dataset: 16000 samples
train dataset: 4000 samples

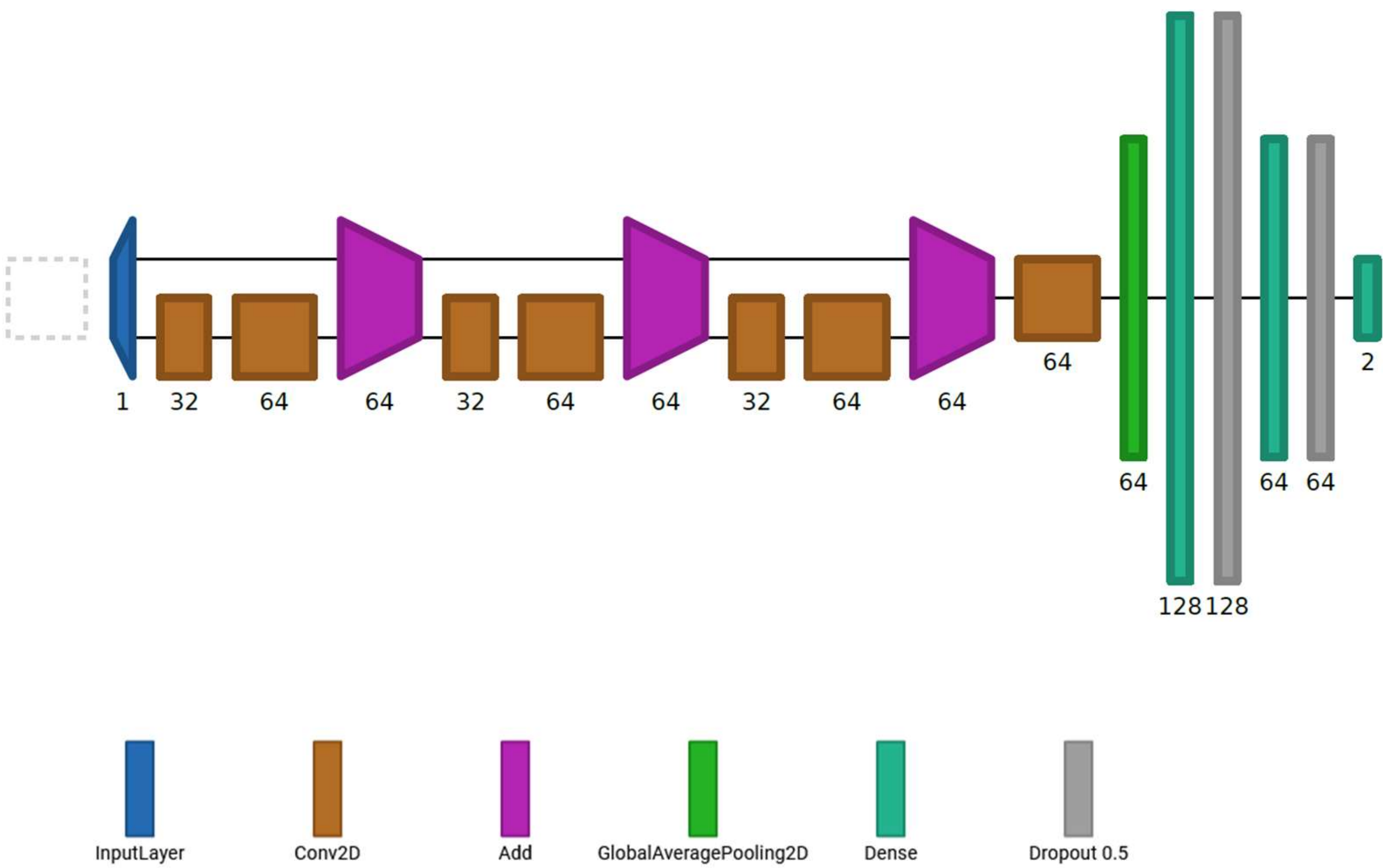
74% noise, 26% signal (inbalanced data)

28763.75 average number of clusters in the event (important to calculate the performance time)

80% | 20% train - test split

CPU: Intel Xeon X5550, 2.67 GHz	GPU: GeForce RTX2080 Ti	Noise reduction	Params	FalseNegative
?? Seconds / Event	**?? Seconds / Event**	**XX%?**	**NN**	**YY%?**

ResNet



*Visualization with Net2Vis

- Input fed into 3 blocks of convolutional layers with shortcuts between blocks
- Feature maps are then averaged to a single value (Global Average Pooling)
- Values fed into Dense network
- 2 Outputs
  (corresponding to „good“ and „bad“ classes)

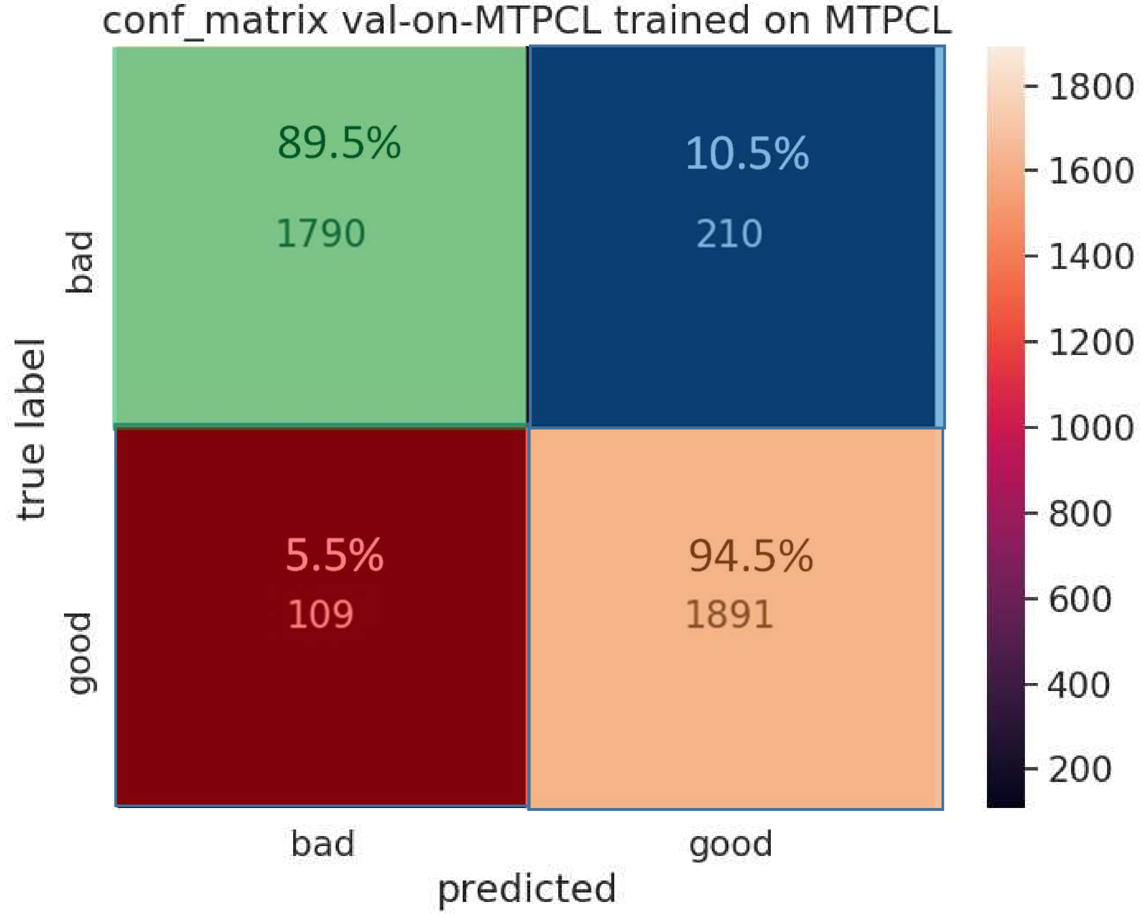
Trainable Parameters: 95.170

Validation Accuracy:
(trained and validated on the same dataset)

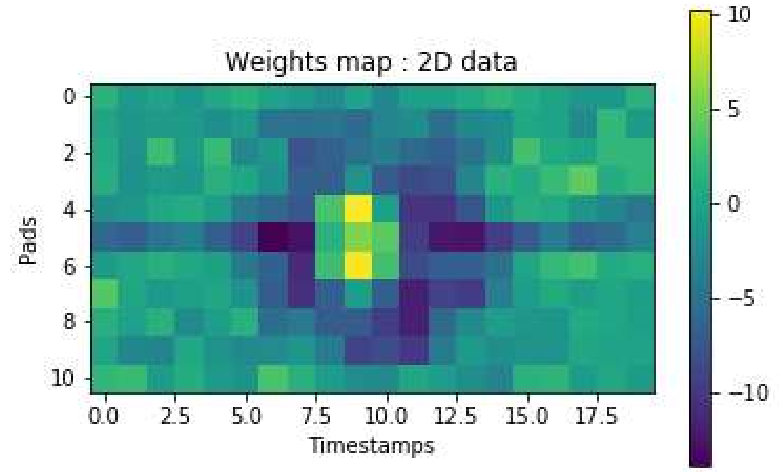
- 92,0 % on MTPCL
- 92,6 % on VTPC2

3.2 ResNet

Trained on MTPCL 2d dataset



Input data



- 92,0% overall accuracy
- **90 % of noise is removed**
- 5,5% of „good“ clusters are wrongly predicted as „bad

Improvement of 2D over the 1D input.
Questions:
Speed?
Generalisation?

3.2 ResNet

Generalisation from one TPC to the other:

90,4 % accuracy trained on VTPC2 validated on MTPCL

→ - 2,2 % to validation on VTPC2

91,0 % accuracy trained on MTPCL validated on VTPC2

→ - 1,0 % to validation on MTPCL

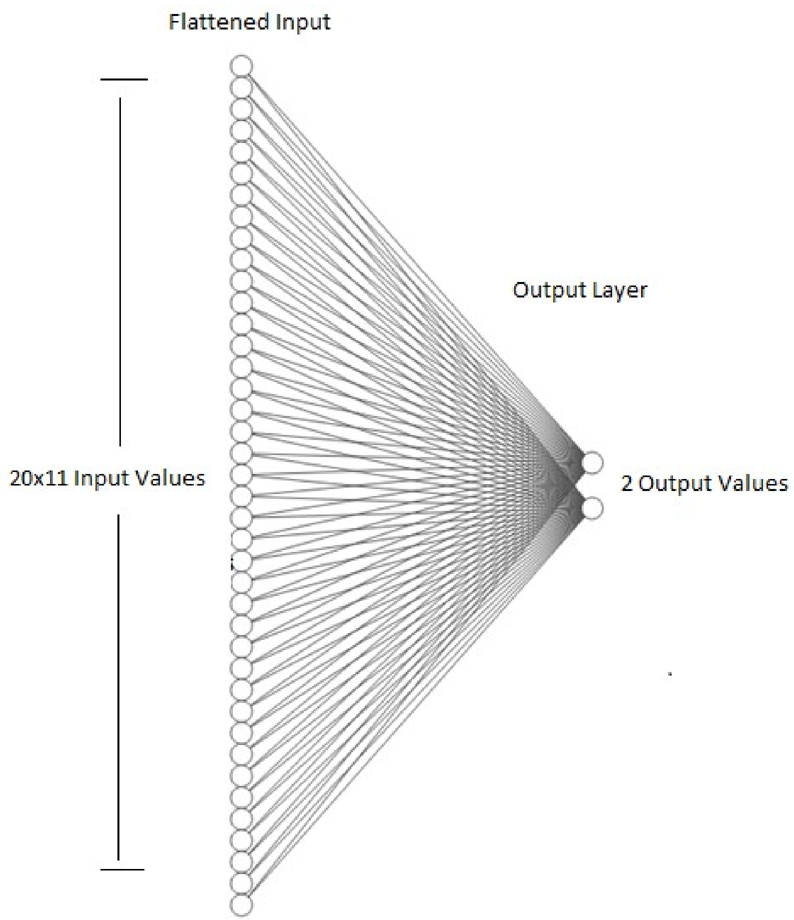
Where does this difference to learning on MTPCL data come from?

→ **Overfitted on VTPC2 dataset!**

Understand therefore trust

Can we understand the reasons behind the decision of the network?

Two-Neuron perceptron



Strategy

- Input flattened and fed directly into 2 output neurons (Perceptron)
- Softmax Activation Function Output example: (0.2|0.8)

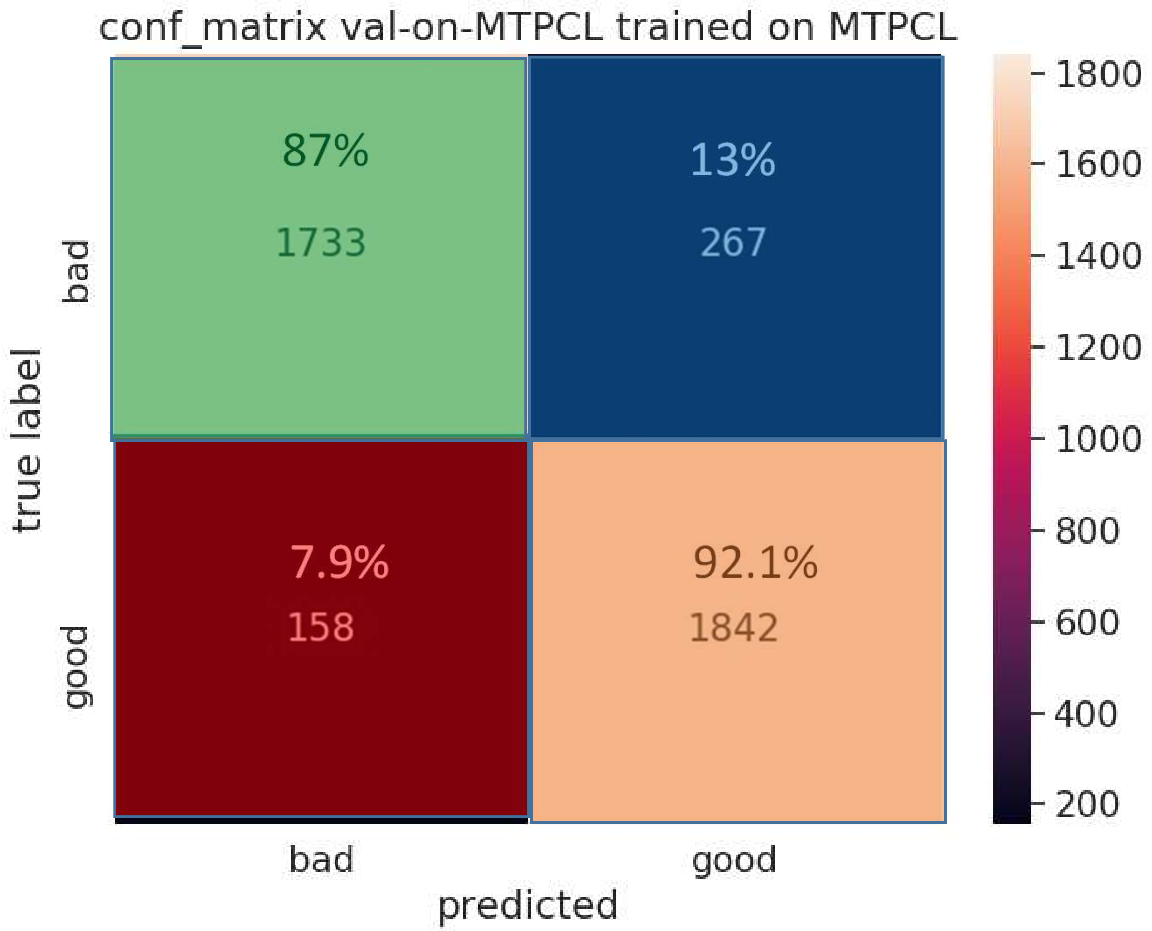
Both outputs combined always add up to 1
→ can be interpreted as propability for the corresponding class label

Trainable Parameters: 442

Validation Accuracy:
(trained and validated on same dataset)
- 89,3 % on MTPCL
- 89,4 % on VTPC2

Cross Validation:
- 89,1 % trained on MTPCL validated on VTPC2
- 89,0 % trained on VTPC2 validated on MTPCL

Confusion matrix



- 87% of noise removed
- 8% of „good“ clusters are wrongly predicted as „bad“

Question:
Why is the number of False Negatives btw. False Positives not symmetrical?
(Tracking algorithm not perfect ?)

Understanding the desicion of the network

What do the Weights look like?

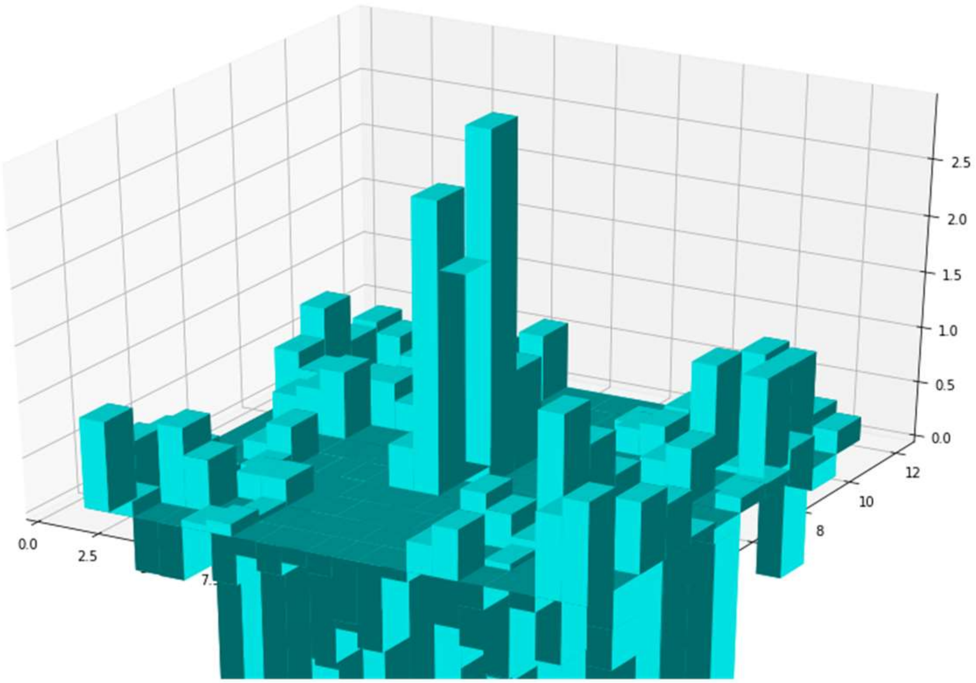


Abbildung 1: 3D Plot of the network weights for the second neuron (if output > 0.5 → „good“ cluster)

**Trained and validated on MTPCL 2D data*

- „inner“ pixel values are weighted heavily (up to x2.5)
- Outer pixel values are mostly low or negatively weighted

„Output is simplified the outer values substracted by the values in the center.“

→ one „peak“ in the center predicted as „good“

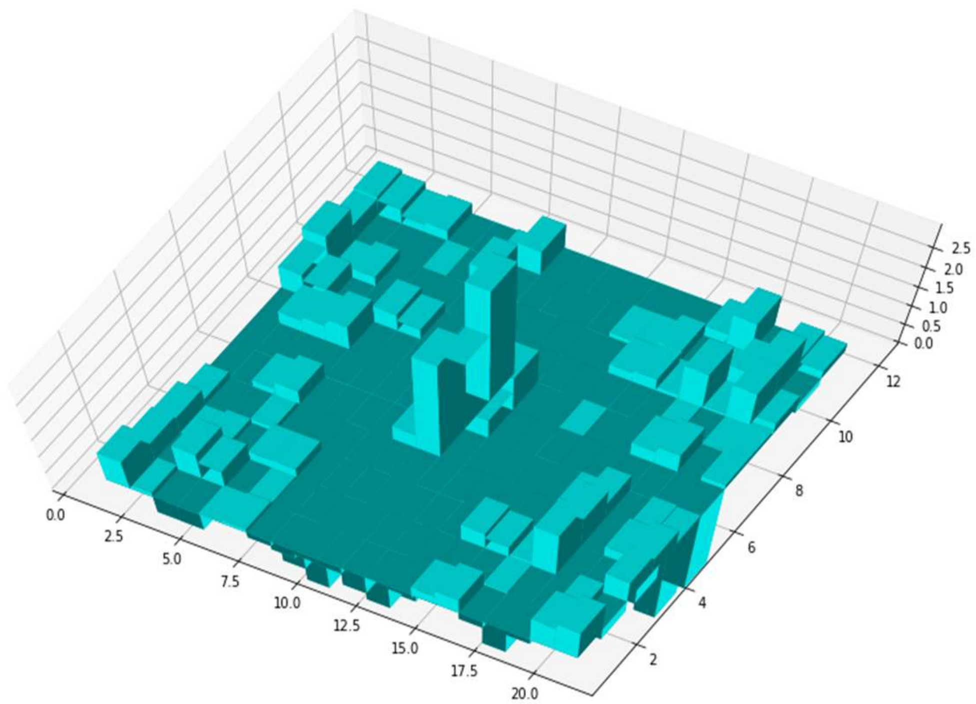
→ The weights „on the cross“ are negative

Question: Why? → Causality!

-> let‘s use this!!

Understanding the desicion of the network

What do the Weights look like?



**Trained and validated on MTPCL 2D data*

- „inner“ pixel values are weighted heavily (up to x2.5)
- Outer pixel values are mostly low or negatively weighted

„Output is simplified the outer values substracted by the values in the center.“

→ one „peak“ in the center predicted as „good“

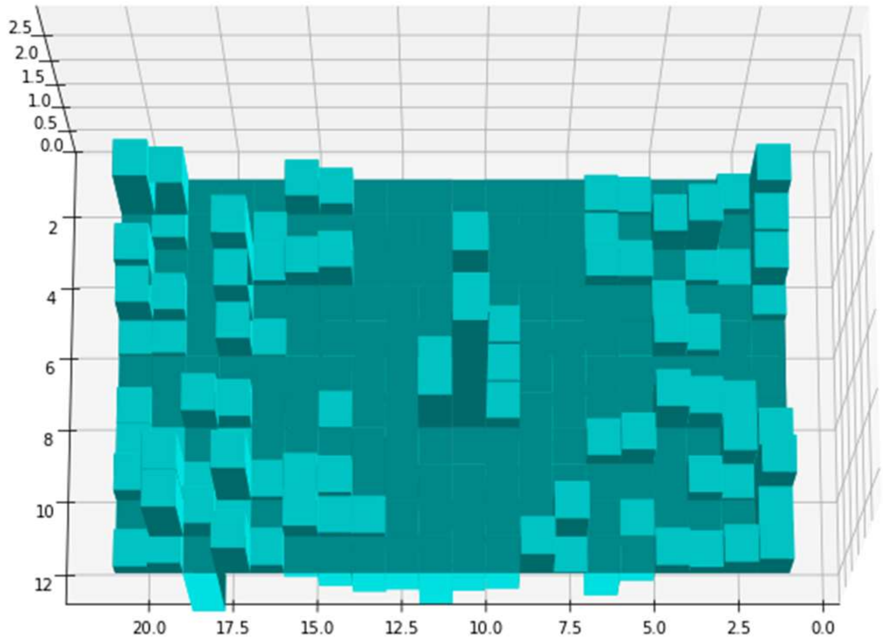
→ The weights „on the cross“ are negative

Question: Why? → Causality!

-> let‘s use this!!

Understanding the desicion of the network

What do the Weights look like?



**Trained and validated on MTPCL 2D data*

- „inner“ pixel values are weighted heavily (up to x2.5)
- Outer pixel values are mostly low or negatively weighted

„Output is simplified the outer values substracted by the values in the center.“

→ one „peak“ in the center predicted as „good“

→ The weights „on the cross“ are negative

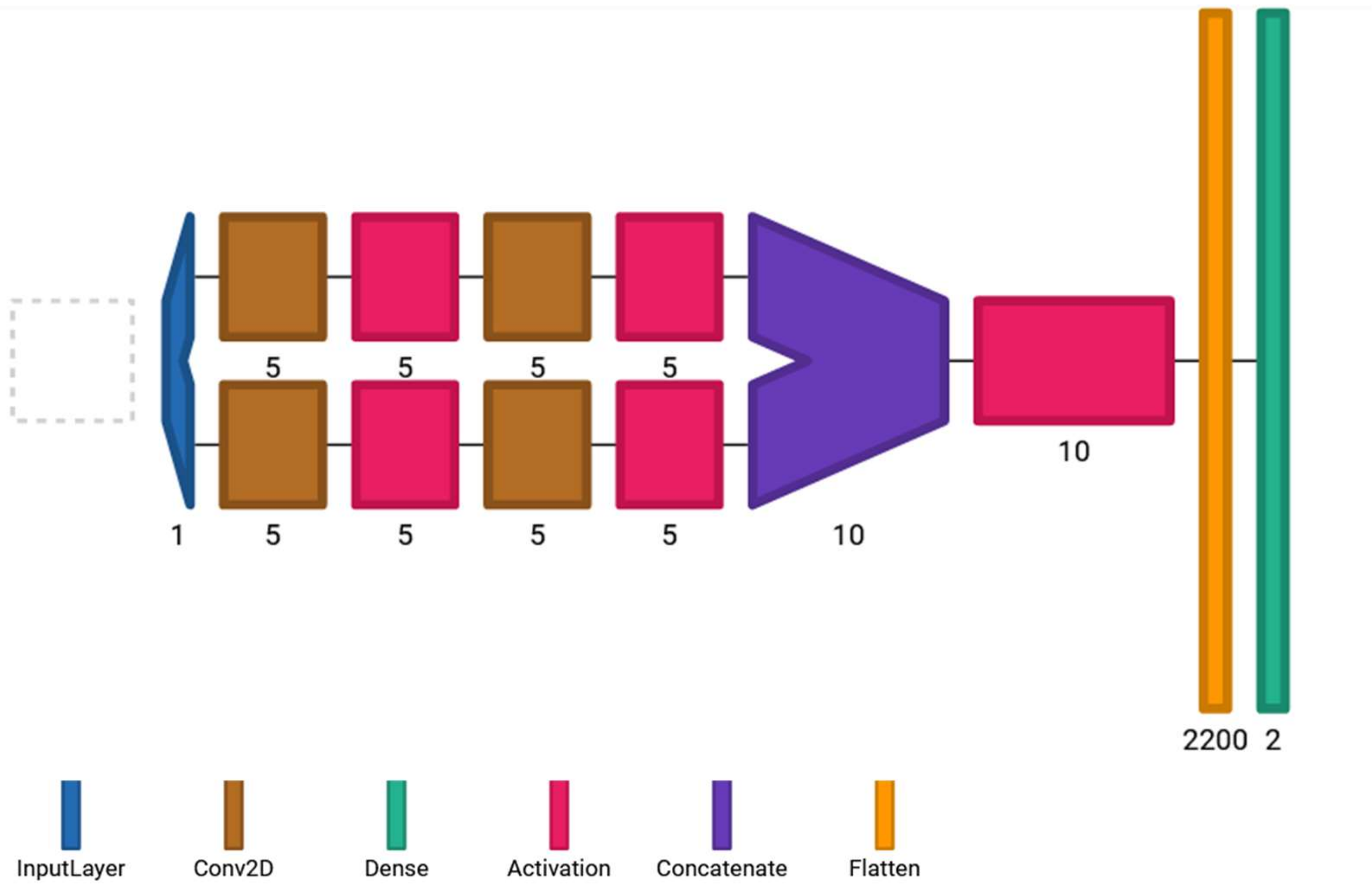
Question: Why? → Causality!

-> let‘s use this!!

Splitted convolution

Network architecture based on physics

Splitted Convolution



*Visualization with Net2Vis

Strategy:

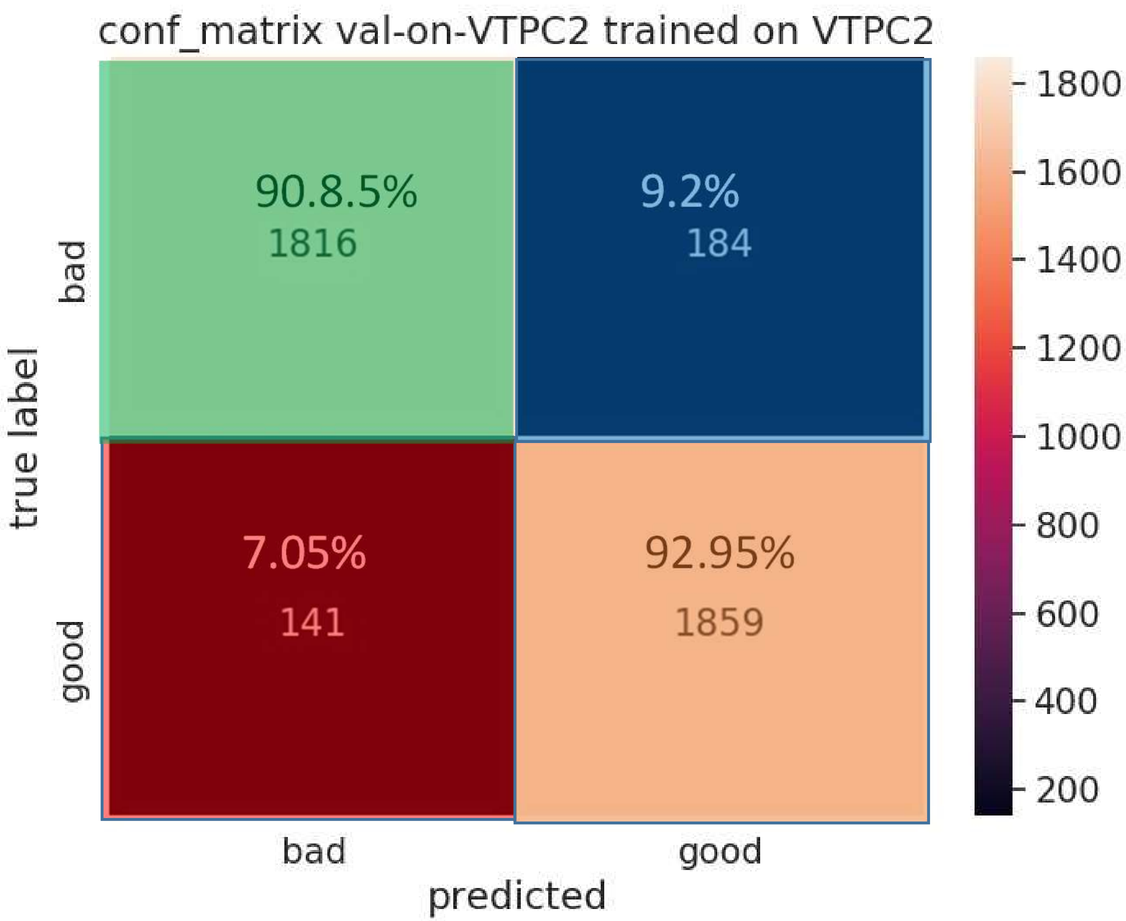
- Input fed into 2 seperate blocks of convolutional layers
- Feature maps are then concatenated and flattened
- Values fed into 2 output nodes (corresponding to „good“ and „bad“ classes)

Trainable Parameters: 4.612 (≈ 1/20 of ResNet)

Accuracy:

- 91,1 % on MTPCL
- 91,9 % on VTPC2

Splitted Convolution



- 91 % of noise is removed
- 7 % of „good“ clusters are wrongly predicted as „bad

Overall Accuracy: 91,9 %

Cross-Validation on a different TPC: 89,4 %

Computational Time

Data for all groups come from the same chunk: Ar + Sc at 30GeV (run ....)

74% noise, 26% signal (inbalanced data)

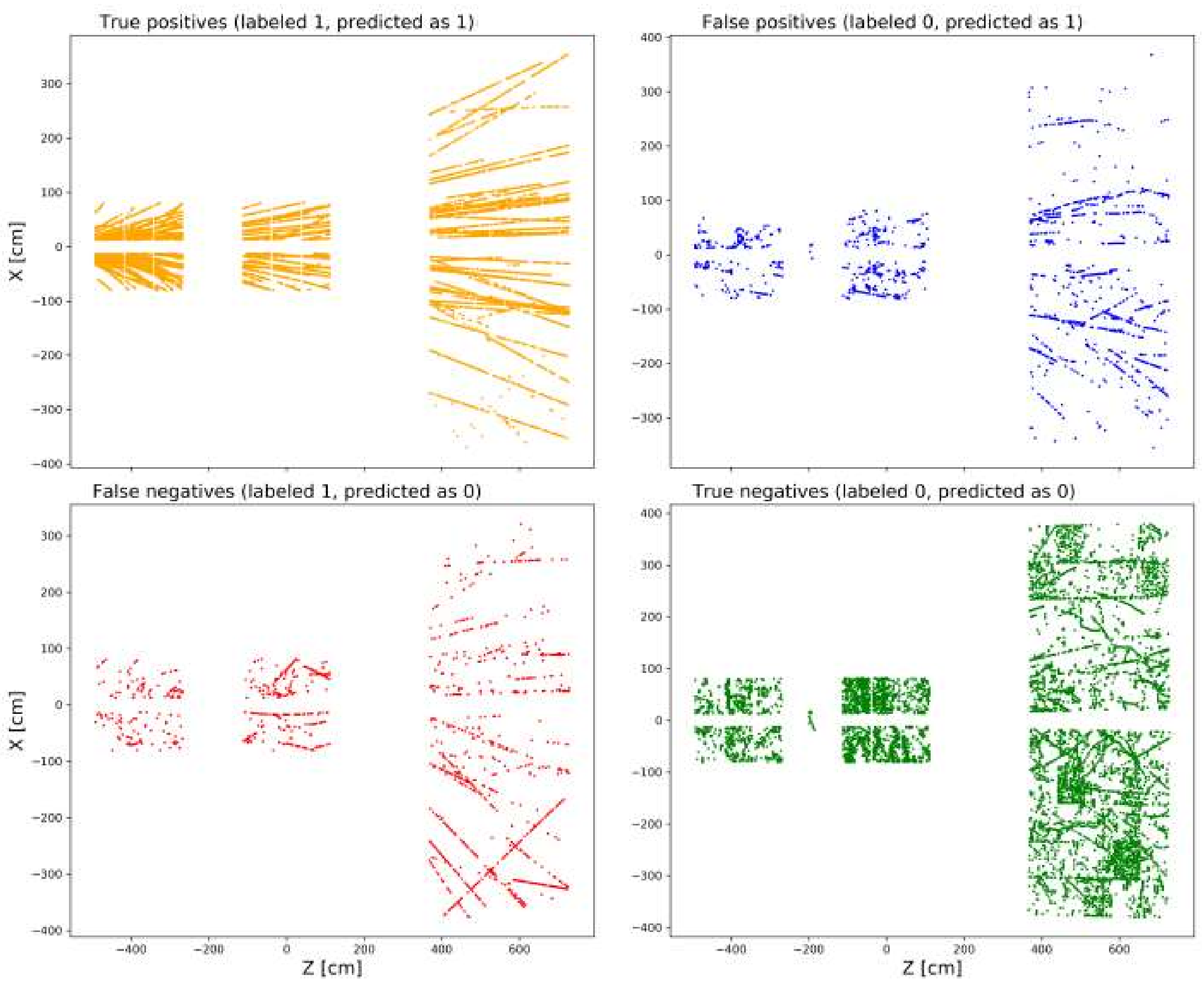
28763.75 average number of clusters in the event (important to calculate the performance time)

80% | 20% train - test split

Test dataset: 16000 samples
train dataset: 4000 samples

CPU: Intel Xeon X5550, 2.67 GHz	GPU: GeForce RTX2080 Ti	Noise reduction	Params	FalseNegatives
network: split_input7 **2.6 s/Event**	network: split_input7 **0.33 s/Event**	**91%**	**4 000**	**6-8%**
network: res2_final **20.4 s/Event**	network: res2_final **0.78 s/Event**	**90%**	**100 000**	**4-5%**
network: simple_final **0.1 s/Event**	network: simple_final **0.01 s/Event**	**87%**	**400**	**8-10%**

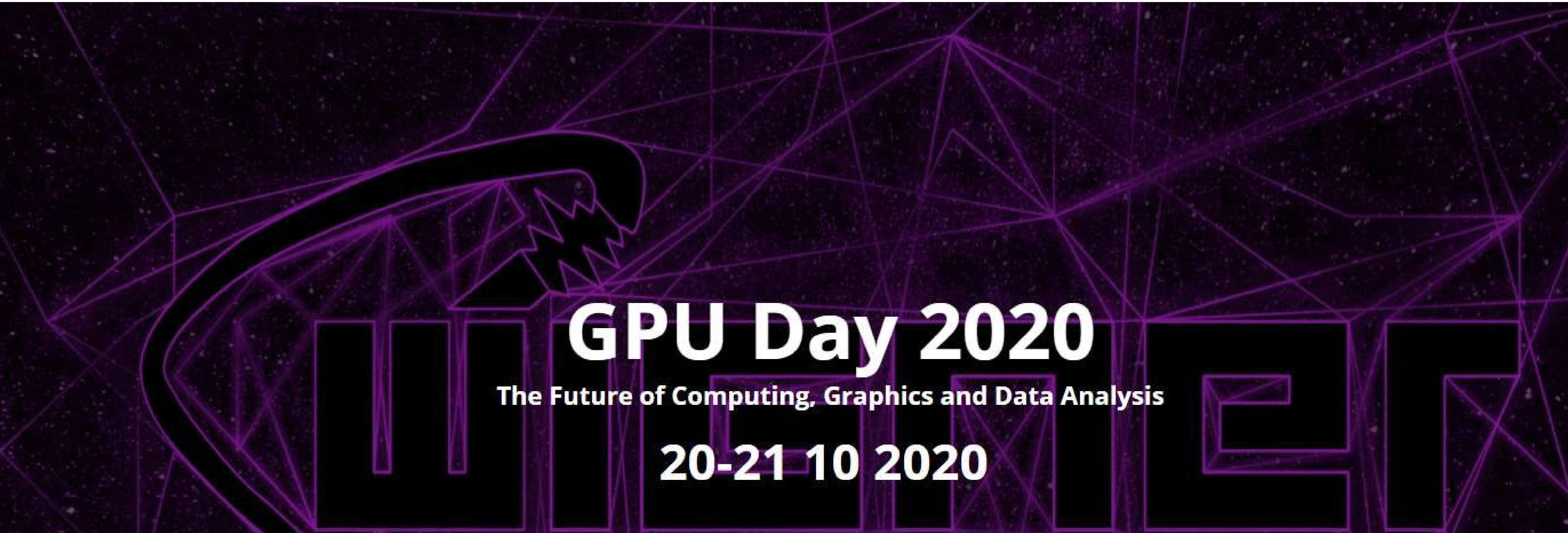
Conclusions and Outlook



Check-list

1) Improvements:
   - ✓ over 90% noise reduction
   - ✓ ~67% of hits can be removed from the tracking
2) Speed:
   - ✓ <1 s/Event/Node possible on one GPU
3) Generalisation:
   - ✓ TPCV to TPCL works,
   - 30 AGeV to … 158 AGeV – to be checked
   - … collision systems – to be cheked
4) Efficiency:
   - ✓ ~4% of signal is lost,
   - … Lost signal vs p_T, mass, PID – to be checked

GPUs allow the benefit of the depth in the cases where performance is key



THANK YOU FOR YOUR ATTENTION