Neural networks and their application at CAST

Sebastian Schmidt

University of Bonn

26th May 2018

Physikalisches Institut

universitätbonn

CERN Axion Solar Telescope

CAST

the experiment

- search for solar axions, hypothetical pseudoscalar particle solving the strong $\mathcal{CP}$ problem
- potential dark matter candidate
- coupling to transverse B fields, production in the Sun!

CERN Axion Solar Telescope

CAST

to take away...

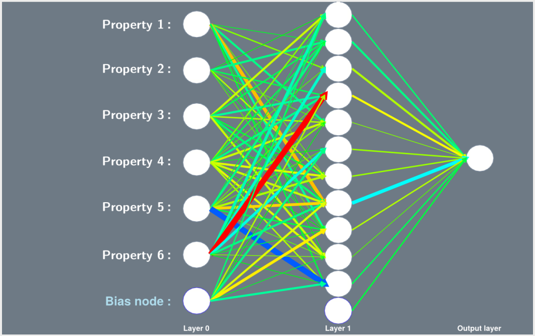
- exp. signal rates: $\leq 0.1\ \gamma\ \mathrm{h}^{-1}$
- background rate: $\sim 0.1\,\mathrm{s}^{-1}$
- need very good background suppression

Artificial Neural Networks (ANNs)

ANN primer

- type of multivariate analysis object providing highly non-linear, multidimensional representations of input data
- simplest type: feed-forward multilayer perceptron

MLP example



Artificial Neural Networks (ANNs)

Producing an output and training

Neuron output:

$$y_k = \varphi \sum_{j=0}^{m} w_{kj} x_j$$

φ: activation function, w_k weight vector
Training minimizes error function

$$E(\mathbf{x_1}, \ldots, \mathbf{x_N} | \mathbf{w}) = \sum_{a=1}^{N} \frac{1}{2} \left(y_{\mathsf{ANN},a} - \hat{y}_a\right)^2$$

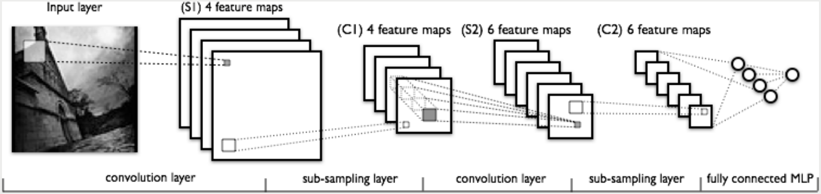
using gradient descent

$$\mathbf{w}^{n+1} = \mathbf{w}^n - \eta \nabla_w E$$

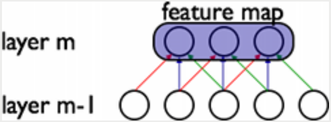
Convolutional Neural Networks

CNN schmatic

convolutional and pooling layers alternating:



where a convolutional layer is:



Convolution example in python

Python calc of 2D convolution (instead of a gif...)

```python
import numpy as np
from scipy.signal import convolve2d
A = np.identity(6)
B = np.array([[0,0,0],[0,5,0],[0,0,0]])
C = convolve2d(A, B, 'same')
print(C)
```

```
[[5. 0. 0. 0. 0. 0.]
 [0. 5. 0. 0. 0. 0.]
 [0. 0. 5. 0. 0. 0.]
 [0. 0. 0. 5. 0. 0.]
 [0. 0. 0. 0. 5. 0.]
 [0. 0. 0. 0. 0. 5.]]
```

Convolution example in python

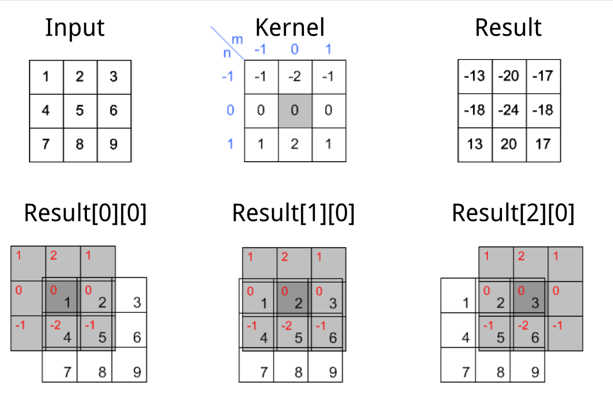
Python calc of 2D convolution (instead of a gif...)

```python
import numpy as np
from scipy.signal import convolve2d
A = np.identity(6)
B = np.array([[1,0,1],[0,1,0],[1,0,1]])
C = convolve2d(A, B, 'same')
print(C)
```

```
[[2. 0. 1. 0. 0. 0.]
 [0. 3. 0. 1. 0. 0.]
 [1. 0. 3. 0. 1. 0.]
 [0. 1. 0. 3. 0. 1.]
 [0. 0. 1. 0. 3. 0.]
 [0. 0. 0. 1. 0. 2.]]
```

Convolution example in pictures

A pictures is worth a thousand words?



source: `http://www.songho.ca/dsp/convolution/convolution2d_example.html`

Live demo of MLP training on MNIST

Simple demo of training simple ANN on MNIST

- MNIST: a dataset of 70 000 handwritten digits, size normalized to 28×28 pixels, centered
  - in the past used to benchmark image classification; nowadays fast to achieve good accuracies $\geq 90\,\%$
- network layout:
  - input neurons: 28×28 neurons (note: as 1D!)
  - 1 hidden layer: 1000 neurons
  - output layer: 10 neurons (1 for each digit)
  - activation function: rectified liner unit (`ReLU`):

$$f(x) = \max(0, x)$$

Live demo of MLP training on MNIST

What do I mean by live demo? 2 programs

- Program 1: trains multilayer perceptron (MLP)
  - written in Nim (`C` backend), using Arraymancer
    - linear algebra + neural network library
  - trains on 60 000 digits, performs validation on 10 000 digits
- after every 10 batches (1 batch: 64 digits) send to program 2:
  - random test digit
  - predicted output
  - current error
- Program 2 plots data live: written in Nim (`JS` backend), plots using `plotly.js`

Start training!

Back to CAST

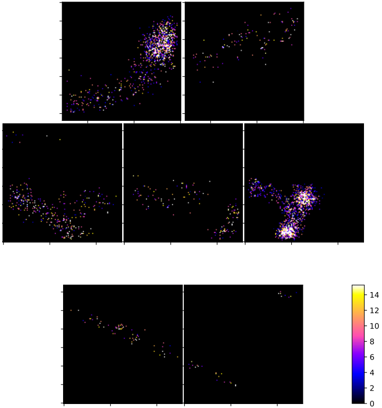
Requirements for detectors at CAST

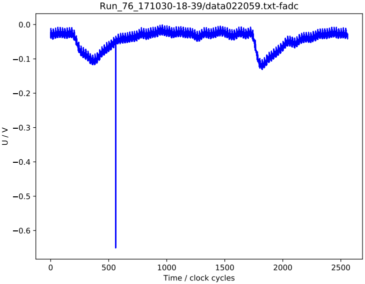
- CAST is a very low rate experiment!
- detectors should reach: $f_{\text{Background}} \leq 10^{-6}\,\text{keV}^{-1}\,\text{cm}^{-2}\,\text{s}^{-1}$
- signal / background ratio: $\frac{f_{\text{Background}}}{f_{\text{Signal}}} > 10^5$
  - need very good signal / background classification!

Background example

```
Hits
Chip #0 : 64
Chip #1 : 29
Chip #2 : 347
Chip #3 : 143
Chip #4 : 1365
Chip #5 : 123
Chip #6 : 1033
```

```
Run # :                76
Event # :              22059
Date :                 2017-10-31.09:01:17
Shutter time / mode :  32 / verylong
FADC triggered :       1
    at :               83508713 clock cycles
Veto scint clock :     4095
SiPM clock :           0
Filename :             data022059.txt
```

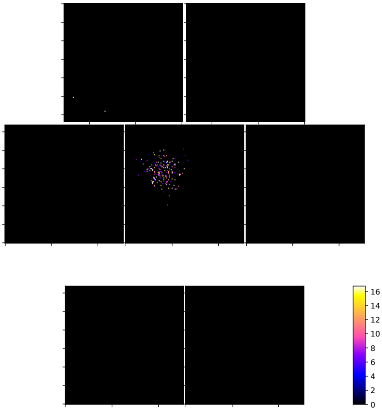


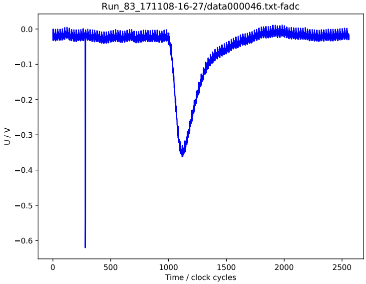


X-ray example

```
Hits
Chip #0 : 0
Chip #1 : 0
Chip #2 : 0
Chip #3 : 211
Chip #4 : 0
Chip #5 : 0
Chip #6 : 2
```

```
Run # :                  83
Event # :                46
Date :                   2017-11-08.16:28:08
Shutter time / mode :    32 / verylong
FADC triggered :         1
    at :                 316651 clock cycles
Veto scint clock :       4095
SiPM clock :             0
Filename :               data000046.txt
```





Back to CAST

Requirements for detectors at CAST

- CAST is a very low rate experiment!
- detectors should reach: $f_{\text{Background}} \leq 10^{-6}\,\text{keV}^{-1}\,\text{cm}^{-2}\,\text{s}^{-1}$
- signal / background ratio: $\frac{f_{\text{Background}}}{f_{\text{Signal}}} > 10^5$
  - need very good signal / background classification!
- events (as on previous slides) can be interpreted as images
- Convolutional Neural Networks extremely good at image classification

$\Rightarrow$ use Convolutional Neural Networks?

Old analysis - data and likelihood method

- visible from comparison of background to X-ray event that geometric shapes are very different
- utilize that to remove as much background as possible

Likelihood analysis

- energy range: 0 keV to 10 keV
- split into 8 unequal bins of distinct event properties

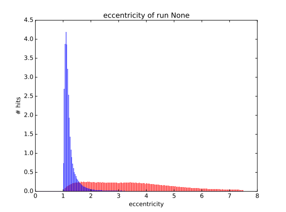
Baseline analysis

Analysis pipeline as follows

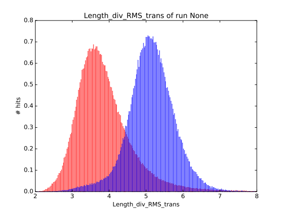
⇒ raw events
filter 'clusters'
calc (geometric) properties
calc likelihood distribution from:

- eccentricity
- length / transverse RMS
- fraction within transverse RMS

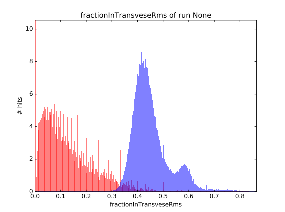
Eccentricity



Length / $\text{RMS}_{\text{trans}}$



pix in $\text{RMS}_{\text{trans}}$



Current analysis - data and likelihood method

Likelihood analysis & CNN analysis

- energy range: 0 keV to 10 keV
- split into 8 unequal bins of distinct event properties
- only based on properties of X-rays
- set cut on Likelihood distribution, s.t. 80 % of X-rays are recovered
- now: use artificial neural network to classify events as X-ray or background

ANNs applied to CAST

Two ANN approaches

1. calculate properties of event, use properties as input neurons
2. use whole events (256×256 pixels) as input layer
3. reg. 1:
   - small layout $\Rightarrow$ fast to train
   - potentially biased, not all information usable
4. reg. 2:
   - huge layout $\Rightarrow$ only trainable on GPU
   - all information available

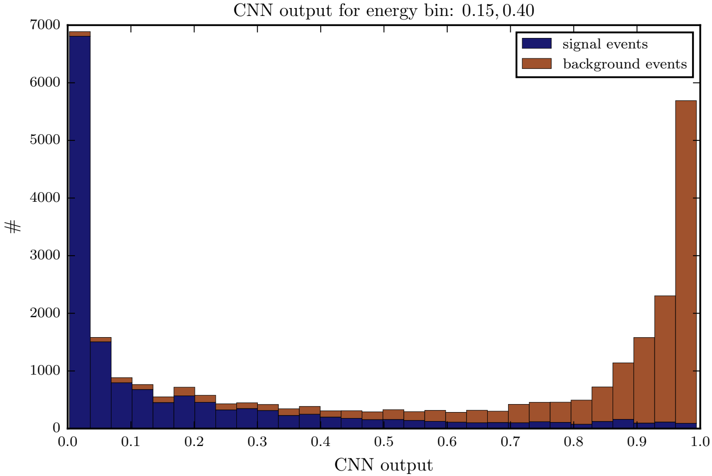
CNN implementation details

8 networks in total, one for each E bin

- input size: 256×256 neurons
- 3 convolutional and pooling layers alternating w/ 30, 70, 100 kernels using 15×15 filters
- pooling layers perform 2×2 max pooling
- $\tanh$ activation function
- 1 fully connected feed-forward layer: (1800, 30) neurons
- logistic regression layer: 2 output neurons
- training w/ 12 000 events per type on Nvidia GTX 1080
- training time: $\sim 1\,\text{h}$ to $10\,\text{h}$

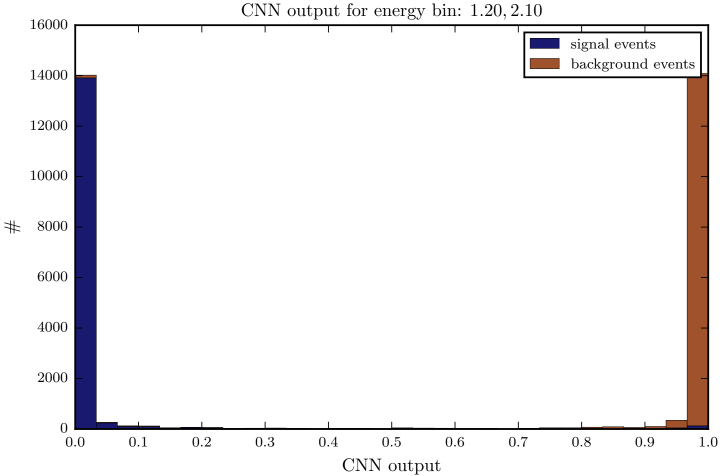
CNN example output distribution

CNN output distribution: bad



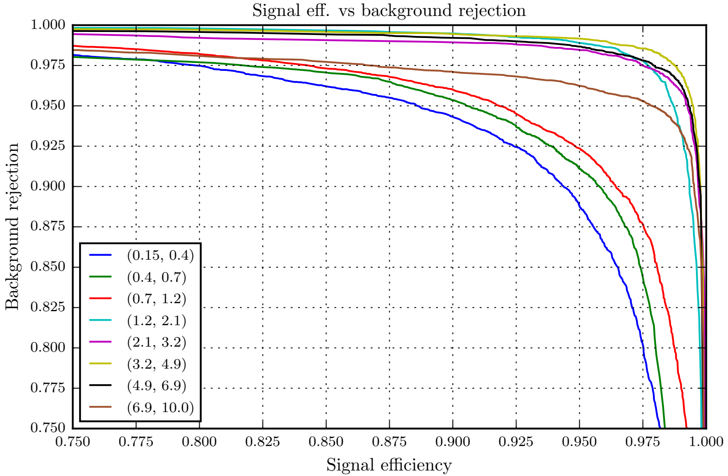
CNN example output distribution

CNN output distribution: good



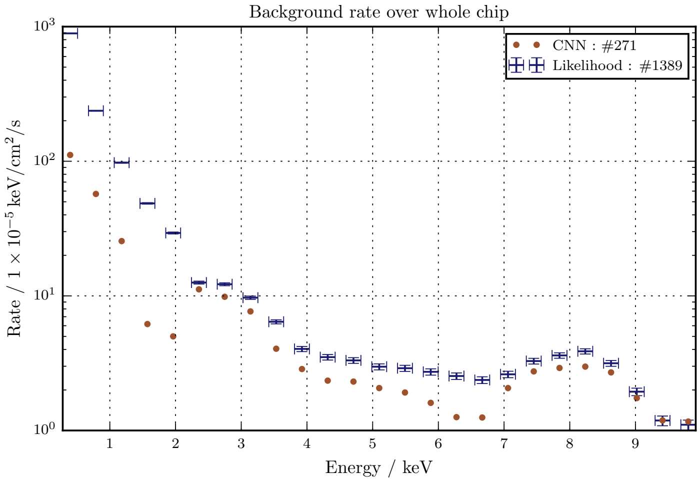
Potential improvements via CNNs

Signal eff. vs background rej.



Potential improvements via CNNs

baseline vs. CNNs: 5× background reduction (2014/15 data)



"Summary"

- I hope I could teach you something new / it was still interesting regardless :)
- if you're interested: this talk and the code for the live demo can be found on my GitHub: `https://github.com/vindaar/NeuralNetworkLiveDemo`