Reconstruction of Micropattern Detector Signals using Convolutional Neural Networks

L. Flekova (TU Darmstadt), M. Schott (Uni Mainz)

Outline
• Micromegas Detectors
• Convolutional Networks
• Performance Results
• FPGA’s and Reduced CNN
• Conclusion
Micromegas Detectors

- Incident particles ionize gas-atoms (Ar/CO$_2$)
  - Resulting electrons drift towards amplification mesh

- Electric field in amplification is high enough to cause avalanches

- Amplification of $10^3$-$10^4$ of primary electrons, which induce charge at readout-strips

- Sparking can be prevented by resistive strips on top of readout layers
Possible Applications

- Micromegas detectors are low-cost alternatives for tube-chambers, with
  - high background rate stability
  - good spatial (40μm)
  - and timing resolution (5ns)

- Applications
  - Small areas in CAST and COMPASS
  - First large scale application: ATLAS Upgrade of the Muon Systems
  - Future: Any high-rate, large-area environment
Most current Micromegas detectors in HEP are readout through APV25 hybrid cards

- 128 channels readout in 25 time-steps stored at 40 Mhz rate
- Analog CR-RC shaped signals gives information on measured charge on channel

Goal: Efficiently identify particles and determine spatial-position and signal-time

- Use Convolutional Neural Network (on FPGA’s)
Conv. Neural Networks

- Fully connected networks do not scale well (one node for each pixel) and do not take into account spatial information (pixel far apart treated the same as neighboring ones)
- In CNN filters learn specific features at spatial positions of input
- Local connectivity
- Shift-invariant
Network Architecture

- Raw data
- Conv. Filter Layer
- Pooling Layer
- 1st Hidden Layer
- 2nd Hidden Layer
- Merge Layer
- Output Layer
Filter Layer

- Several filter types extract relevant shape characteristics.
- Information reduced per subset from 80 input variables to 20.
- Maximum filter via weighted mean used for position extraction.
Merge Layer

- Information of all networks gets combined and clusters restructured
- 360x25 input reduced to 90x4
- Output: Common regions of identified clusters
Dataset and Training

- Use MC simulated events of event signatures, in order to quantify resolution performance
  - Include typical signal characteristics
  - Background noise
  - Dead channels
  - Overlapping signals

- Results are cross-checked with real detector data

- First filter layer optimized to be parallelized as much as possible
- Advanced backpropagation algorithm for training
Discretization for FPGA

- Training is performed using floating point precision
- Input to FPGA has to be reduced to allow for the limited bandwidth (+operations on FPGA)
- Input-patterns reduced to 4-bit
- Weights reduced to 6-bit
- Sigmoidal reduced to 4-bit
First Results

- Total training set of 20k events with different background and detector features
  - Each event has between 0-5 individual particle signatures
- Uncorrelated test statistics of 1k events + real detector data
A particle is identified correctly if the reconstructed position is within 1mm of the expected position. Otherwise, fake reconstruction.

Reconstruction efficiency >90% for full network information.

>85% for reduced values.

No dependency on number of particles per event.

Fake rate <5% and is an artifact of spatial resolution.
Spatial resolution $\approx 250\mu m$ (=width of readout strip) for 4 Byte and 6 Bit precision similar
- Full reconstruction allows for 50$\mu m$
- Similar timing resolution of 36ns for both approaches
Further Input Reduction

- Input of 360x25 values too large for typical FPGA's (600-1000 bits per cycle)
  - Corresponds to a detector of 25-50cm in length
- Reduce input information to 90x4 values (5 cycles to read-in) before filtering
Performance Estimate

- Reduced setup very similar performance to full CNN
- No dependence on number of signal clusters
- No increase of fake-reconstructions
Reduced CNN has been tested with real data taken at:
- Testbeam at MAMI electron accelerator at JGU Mainz
- GIF++ facility at CERN (high background rates)
- See in both cases reconstruction efficiencies >90%

Reduced CNN is currently implemented on standard FPGA (Xilinx Virtex 6) is expected to generate results in 30-50 cycles (approx 100 ns)
Summary & Next Steps

- Convolutional Neural Network developed for fast Micromegas Detector signal reconstruction
- CNN optimized for the implementation on FPGAs
- Identification efficiency >90%
- Prototype for several other (more complicated) applications
- Implementation on Xilinx Virtex 6 FPGA ongoing – expect to have results soon