Open questions in deep learning techniques for the radio detection

Dmitriy Kostunin

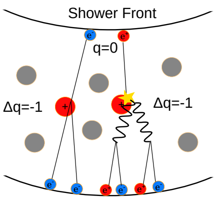
DESY
JetBrains Research

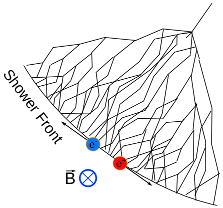
February 3, 2022

Introduction

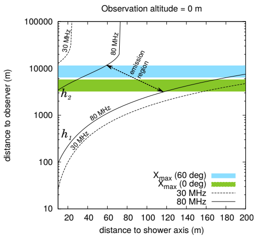
This presentation is given on the Workshop on Machine Learning for Cosmic-Ray Air Showers. I have tried to summarize the progress in radio data analysis and include comments appeared during discussion of the talk.

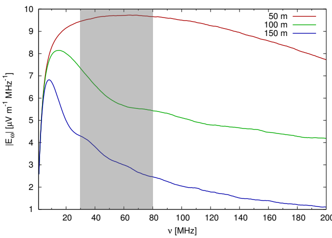
Quick introduction in radio generation

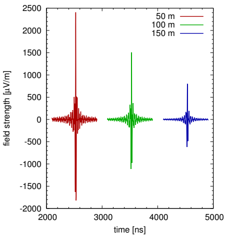


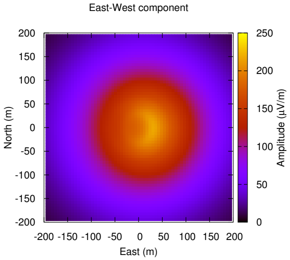


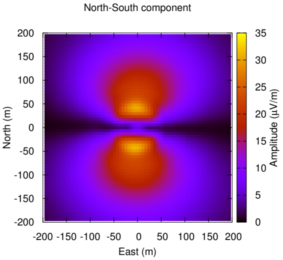
- Askaryan effect
- Geomagnetic effect

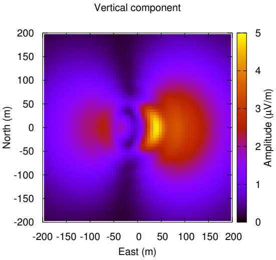




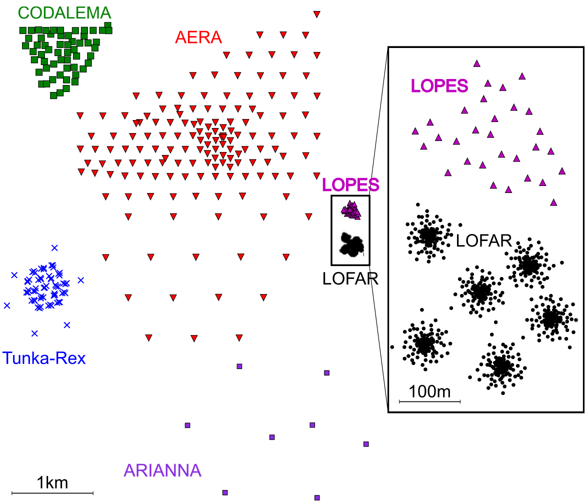


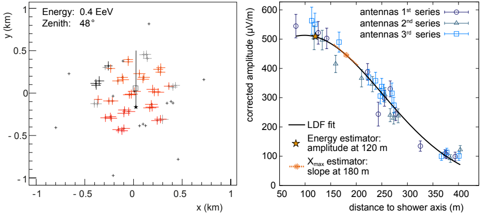


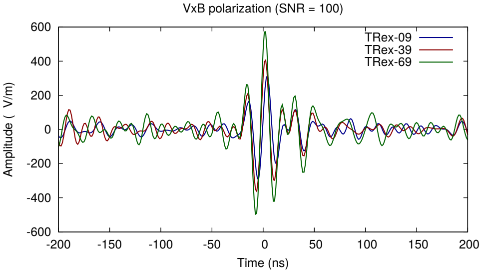




Quick introduction in radio detection

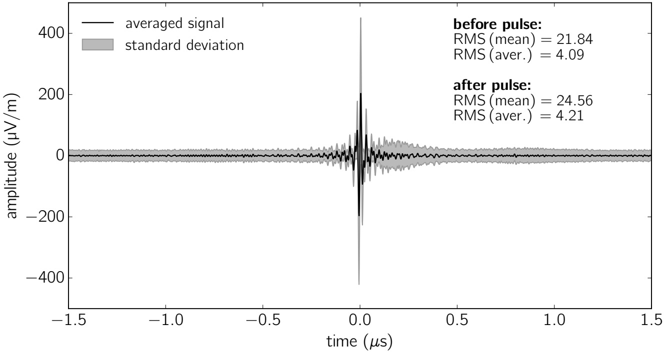


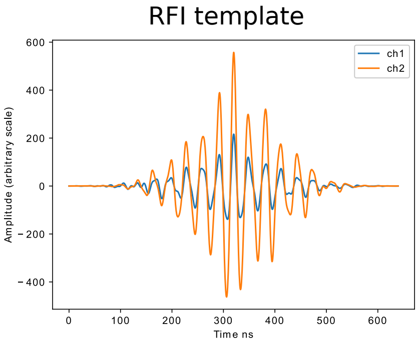




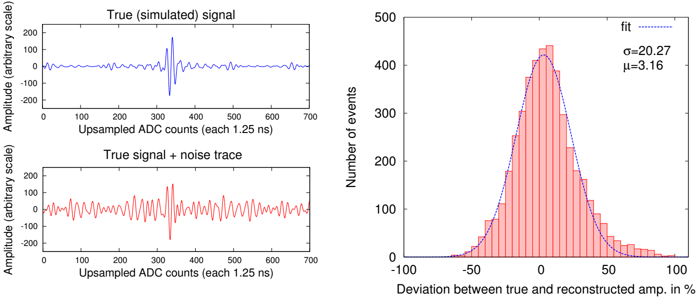
Why do we need machine learning for radio?

- Most of the detectors operate in the low signal-to-noise (SNR) environments
- Background is not completely white $\Rightarrow$ limitation to classical methods
- Radio-frequency interference (RFI) are diverse $\Rightarrow$ hard to account for all





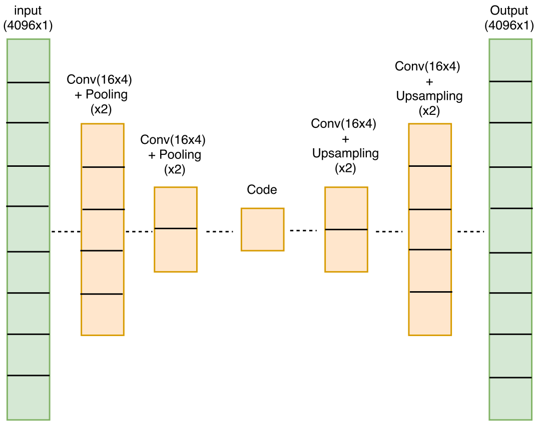
First attempts (back in 2015)



arxiv:1701.05158 (Bezyazeekov+ at ECRS2016)

- Simple perceptron (PyBrain): 200 input neurons, 3 layers $\times$ 500 each, 1 output
- Amplitudes are not normalized, no augmentation $\overset{?}{\rightarrow}$ non-invariant in (t, A)
- Was not tested on real data

Autoencoder approach



Motivation:

- Signal of interest is compact and short comparing to input waveform
- Convolutional filters effectively store signal and background features
- Can be used as denoiser

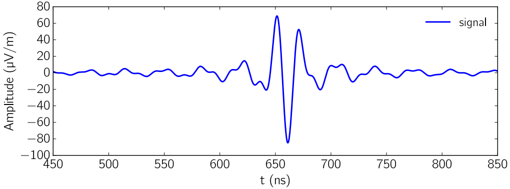
Implementation decisions:

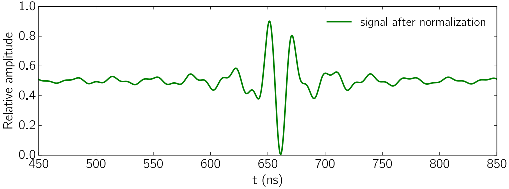
- Size of input waveforms
- Length of convolutional filters
- "Stack more layers" rule should work with large training set

Autoencoder implementation: Preprocessing of the input waveforms

(based on Shipilov+, but others are very similar)

- Nyquist upsampling
- Normalization of amplitudes: $A(t) \to \bar{A} \in [0,1]$
  $\Rightarrow$ amplitude-invariant
- Cropping input to $\mathcal{O}(1024)$ + augmentation: signal is randomly shifted
  $\Rightarrow$ translation-invariant



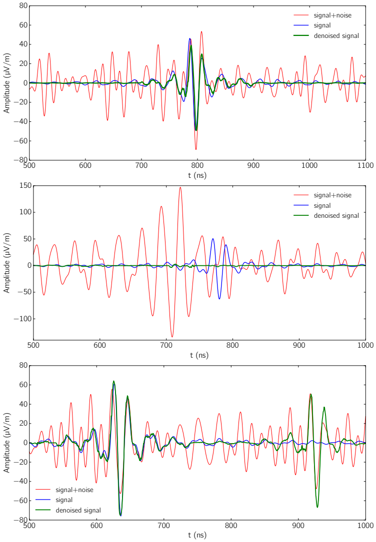


Autoencoder implementation: design and training

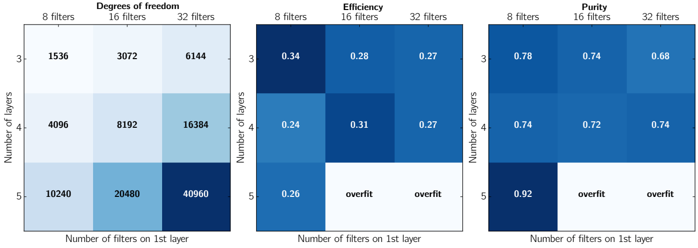
Your input is welcome!

- Filter design
  - Use physical input: air-shower pulse and RFI duration!
  - Upper and lower boundaries for kernel size?
  - Constant or variable size?
  - Upsampling might help autoencoder: optimal sampling factor
- Input waveforms
  - Median filter for preprocessing?
  - Sensitivity to narrow-band background
  - Optimal size in samples and nanoseconds
- Training strategy
  - Optimal size of dataset
  - Binning in amplitude/SNR?
  - Including pure noise in training?
  - How to properly include samples with strong pulse-like RFI

Response and metrics



- Threshold amplitude $\Leftrightarrow$ 5% tolerance to false positives
- Efficiency: $N_{\text{rec.}}/N_{\text{tot.}}$, fraction of events passed the threshold
- Purity: $N_{\text{hit}}/N_{\text{rec.}}$, fraction of events with reconstructed position of the peak: $|t_{\text{rec.}} - t_{\text{true}}| < 5$ ns



Current state of autoencoder implementations

	Shipilov+ 1812.03347	**Schlüter+** 1901.04079	**Rehman+** PoS ICRC2021 (2021) 417
Band (MHz)	30–80	30–80	50–350
Background	Real	Simulated	Simulated*
Normalization	$[0;1]$	$[0;1]$	$[-1;1]$
Input	4096	1000	1024
Bin size (ns)	0.3125	5.6	0.25
Kernel size	$32 \to 16 \to \ldots$	fixed to 5	fixed to 256
Kernel size at 1st layer (ns)	10	28	64
Filters/layer	$16 \to 32 \to \ldots$	$16 \to 32 \to \ldots$	fixed to 8
Layers	3	5	2
Training set	$\approx$15k	$\approx$70k	$\approx$103k+50k
Tested on real data	Yes	No	No

*improved by Kullgren+ (doi:10.5281/zenodo.6011170)

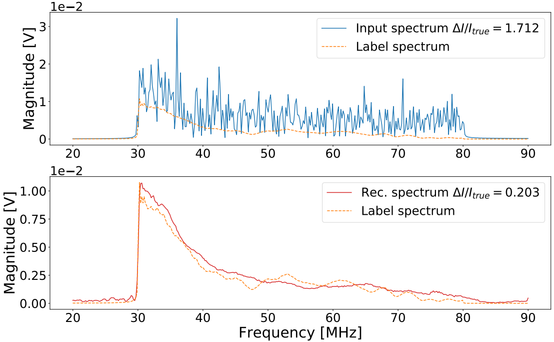
Hereafter we consider only denoisers

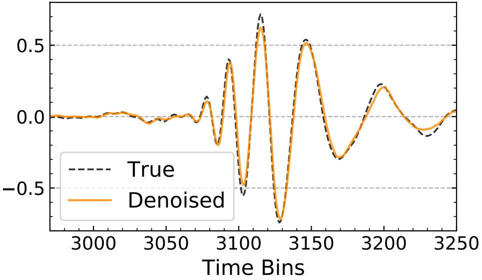
Artificial spectral noise

Artifact when high-rate upsampling is applied (Shipilov+):

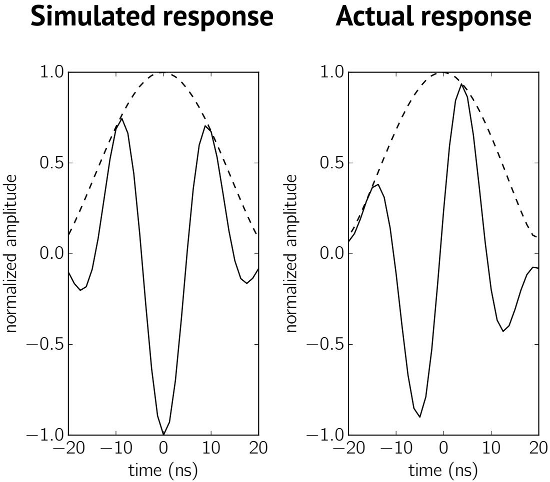
- Nyquist upsampling adds zero-amplitude higher frequencies
- Reconstruction contains noise in-between initial bins
  $\Rightarrow$ artificial noise at higher frequencies

Less impact for non-upsampled (Schlüter+) or low-rate upsampled (Rehman+) signals:





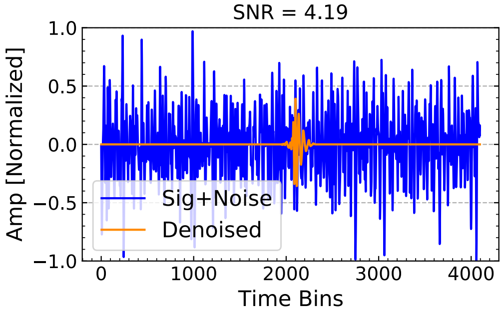
Antenna and phase response invariance

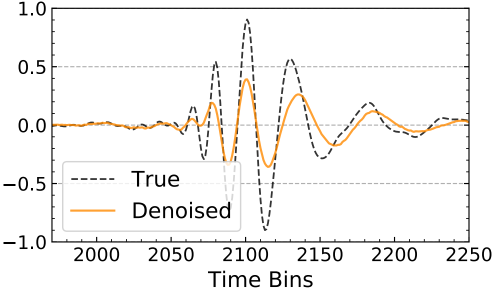


- Calibration does not describe hardware response precisely
- Autoencoder trained by Shipilov+ works surprisingly well both with and without antenna response
- Indication of phase shift invariance
- Implication of this feature?

Amplitude reconstruction

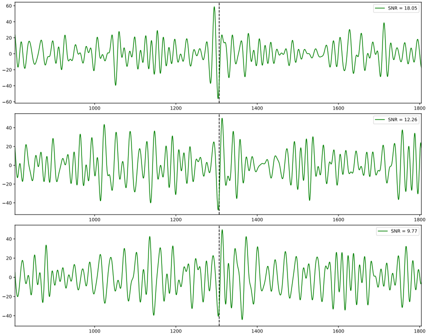
- At low SNR normalization is done w.r.t. noise peaks
- Due to normalization information about absolute amplitude is erased
- Is it fundamental problem for only this architecture or general one?



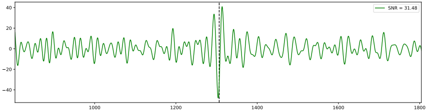


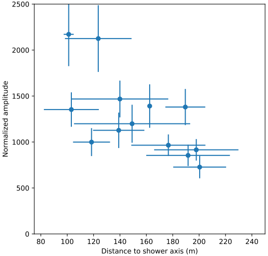
Hybrid approach: autoencoder + interferometry

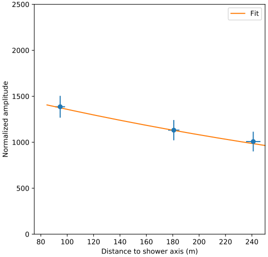
Synthesis of signals using reconstructed arrival times

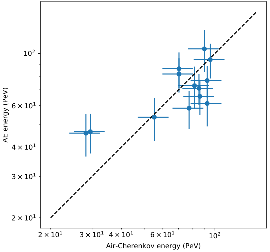


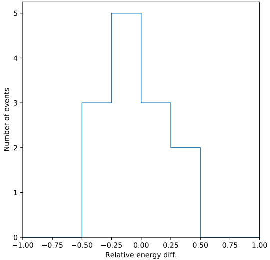
Synthesized signal:











Next steps

Improvements of the current networks

- Alternative architectures, e.g. U-Net, RNN, transformers, LIGO experience
- Adding more channels, wavelets, spectral information
- Integration of modern tools for interpretation, e.g. SHAP
- Spatial information $\Rightarrow$ combination with GNN
- Timing information $\Rightarrow$ combination with CNN encoding sky noise

Compressed neural networks for radio trigger on FPGA

- `hls4ml`: firmware implementations of machine learning algorithms using high level synthesis language (HLS)
- Intensively used in collider physics
- Successfully tested on old ARIANNA hardware (arXiv:2112.01031)
- The only solution for ultra-large scale sparse arrays featuring single antennas

Path to universal network

Search for Holy Grail (and reduce carbon footprint of CoREAS)

Start from GAN?

- High-dimension space: $A(\vec{r},t)$
- Likely more accurate than analytical models, but still not good enough
- Ideal for design studies and template production
- Can be used for RFI and air-shower pulses generation

Ultimate denoiser

- Universality in background? RFI library?
- Train in frequency domain
  - Do they adjustable to every frequency band?
  - CNN will likely not work. Architecture?
  - $F_{\mathrm{true}} = F_{\mathrm{meas}} * F_{\mathrm{IRF}}^{-1}$. How to propagate F_{IRF} to NN?

Conclusion

- Radio community has successfully learned how to use neural networks
- Several architectures are implemented, but only one tested in production
- The recipes for optimal design are not well defined
- Few unique features of radio autoencoders have already been discovered
- The killer feature of technology is FPGA trigger for stand-alone antennas

---

- Training in the frequency domain is almost unexplored:
  - Representation of signals: complex numbers? phase-amplitudes?
  - Treatment of noise? CNN for dynamic spectra? LIGO approach?
  - Efficient phase unwrapping?