Towards Track Reconstruction with ML on FPGAs

Kurt Rinnert

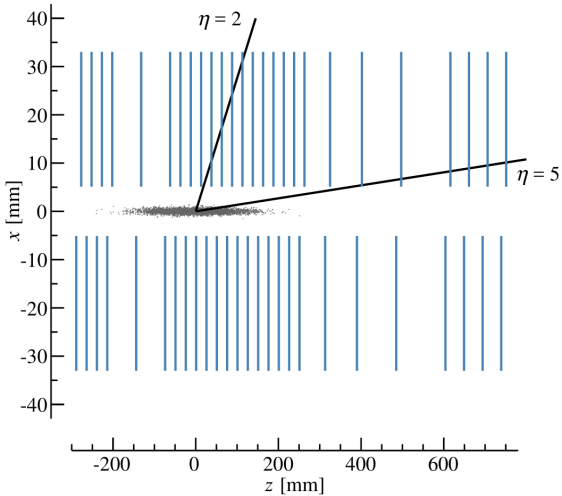
ECHEP
Edinburgh
18.02.2020

The Plan

- Use the upgraded LHCb VELO as a test-bed.
- Connect the dots – no fit at this stage.
- Replace the steps in conventional algorithms with learned functions.
- Port the resulting models to FPGAs.

Working with FBK an Microsoft, we have made some progress in all of these areas.

LHCb VELO Reminder



We are mostly interested in tracks with $2.0 < \eta < 5.0$.

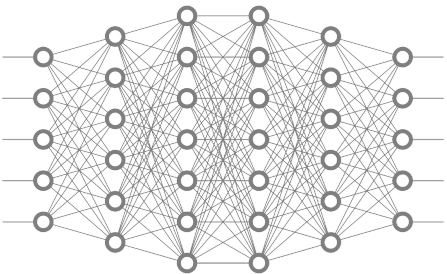
The Algorithm

1. Build ML model to classify space point pairs between detector planes.
2. Build ML model to classify triplets built from pairs.
3. Build tracks by sweeping up triplets downstream to upstream.
    - As of yet, this step is still old-fashioned.
    - Loopy code with hand-crafted clone removal and so on.
    - This will be amended.

Here, classification means calculating *probabilities*.

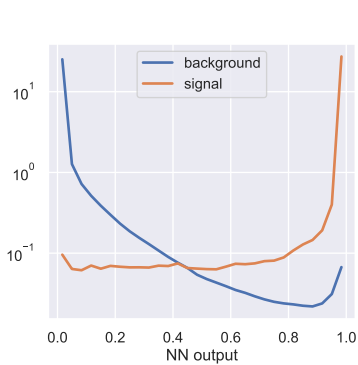
Pair Classification

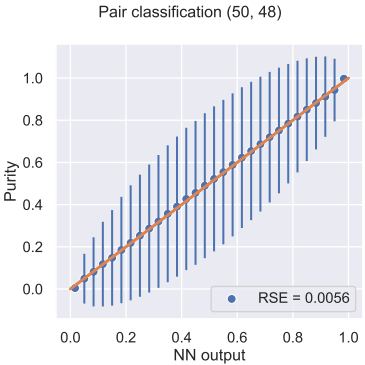
- Six input variables: (r, ϕ, size) of the two clusters.
- Network layout: $6 \times 16 \times 16 \times 1$
- This might seem small.
- But there are $\mathcal{O}(100)$ models with $\mathcal{O}(10000)$ feature vectors in a signal event.
- Train on balanced MC dataset.

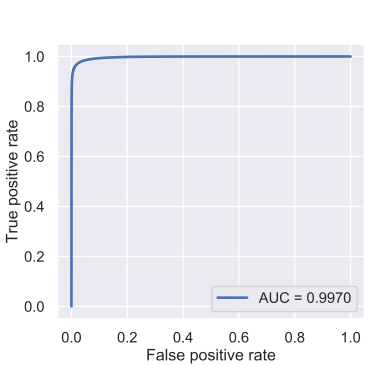


No precuts, no assumptions.

Pair Classification Performance

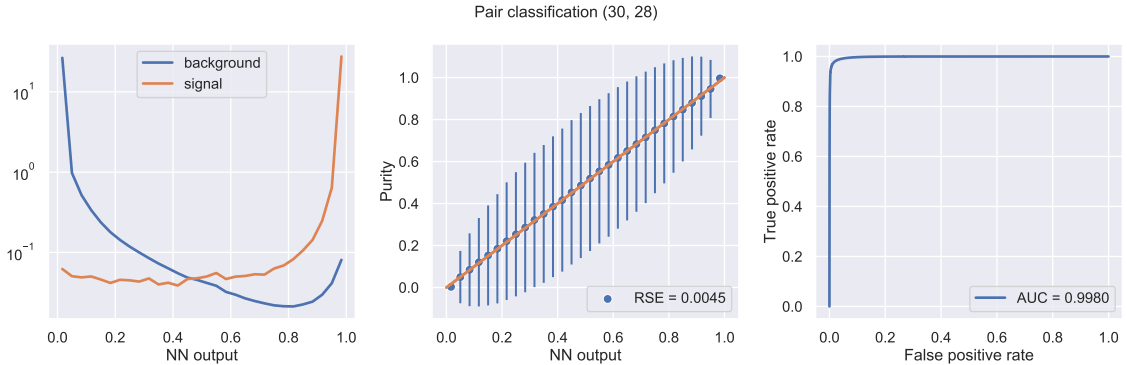






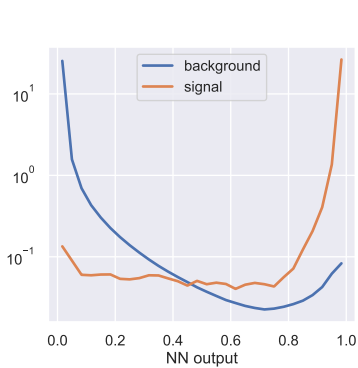
Most downstream pair, far from the interaction point.

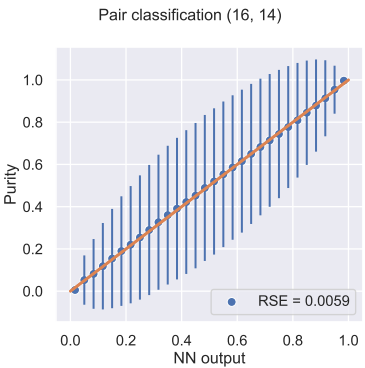
Pair Classification Performance

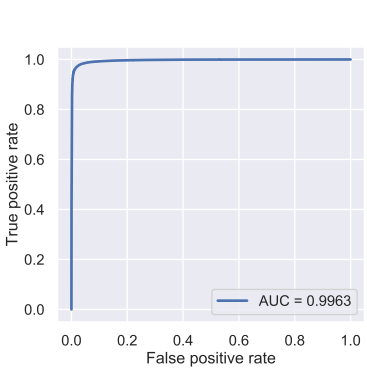


Midway to the interaction point.

Pair Classification Performance



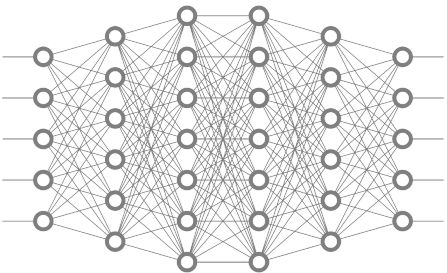




Close to the interaction point.

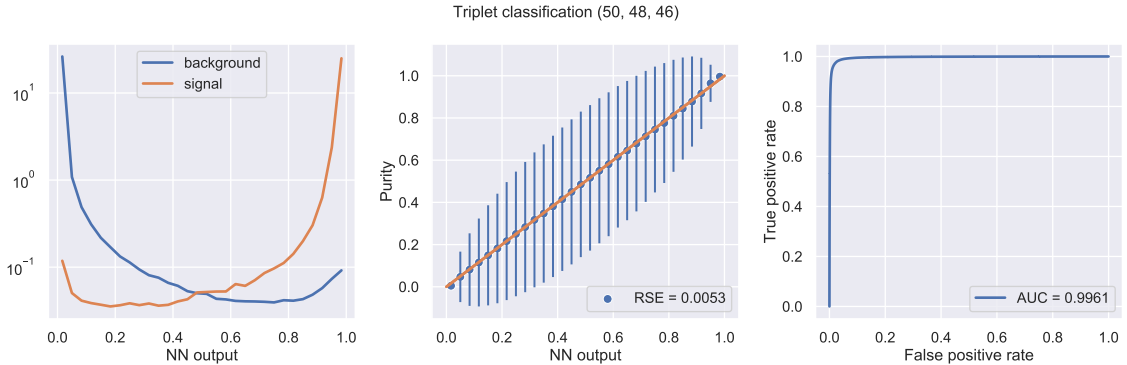
Triplet Classification

- Fifteen input variables.
  - (r, ϕ, size) of the three clusters.
  - Pair probabilities.
  - r and ϕ slopes
- Network layout: $15 \times 64 \times 64 \times 1$
- Needs more complexity than the pair network.
- Only "allowed" pairs are considered (middle clusters must match).
- Train on balanced MC dataset.



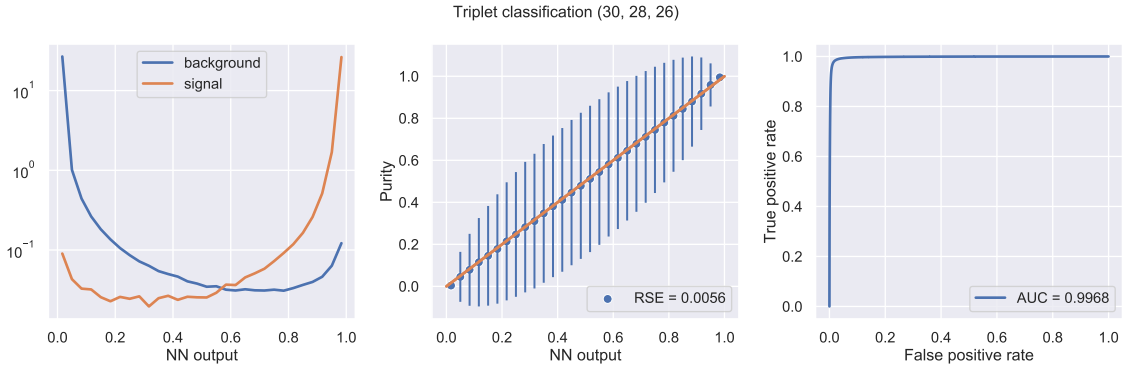
No further pre-cuts or assumptions.

Triplet Classification Performance



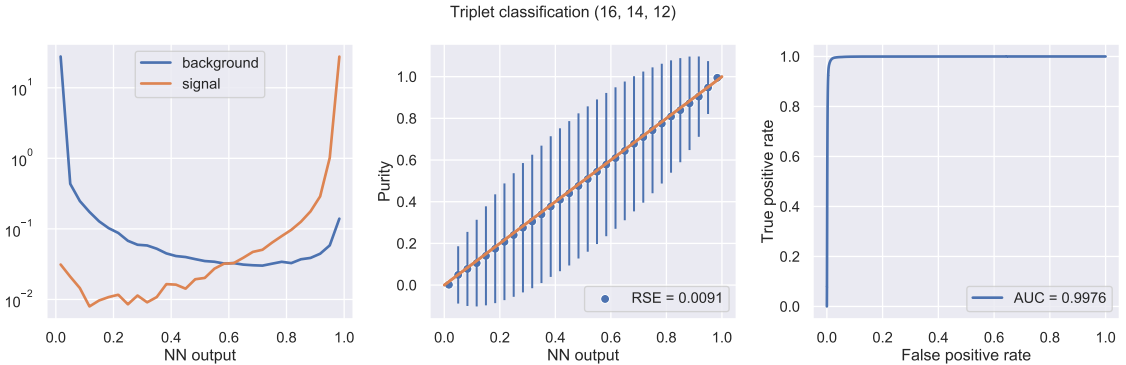
Most downstream triplet, far from the interaction point.

Triplet Classification Performance



Midway to the interaction point.

Triplet Classification Performance



Close to the interaction point.

Algorithm Performance: Efficiencies & Fake Rates

	Fakes	Velo	From B
Search by Pair “fast”	0.83	93.05	95.65
Search by Pair “best”	1.22	97.62	98.71
VELO Tracking SIMD	1.04	98.20	99.12
ML Hybrid	**0.04**	**98.89**	**99.73**

Baseline $B_S \rightarrow \phi\phi$, ML hybrid minimum bias.

Algorithm Performance: Clone Rates

	Velo clones	From B clones
Search by Pair “fast”	2.31	0.89
Search by Pair “best”	2.75	0.84
VELO Tracking SIMD	**1.35**	**0.68**
ML hybrid	1.66	0.80

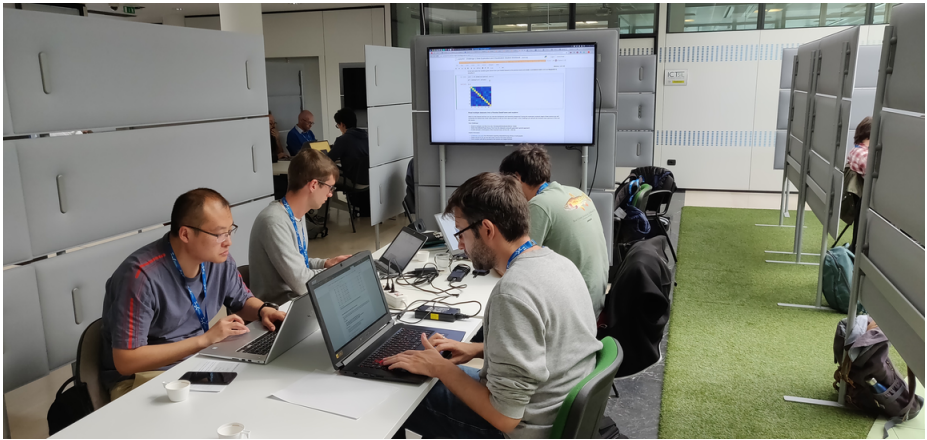
Baseline $B_s \rightarrow \phi\phi$, ML hybrid minimum bias.

Towards FPGA Deployment

- We have been working with Microsoft engineers.
- Exported models from PyTorch to ONNX.
- Import to via custom Brainwave interface.
- Succesful test comparing FPGA results to CPU/GPU inference.

A lot of work ahead, in particular data structures and batching.

Microsoft OpenHack at FBK



This was very well received, great commitment from FBK & MS.