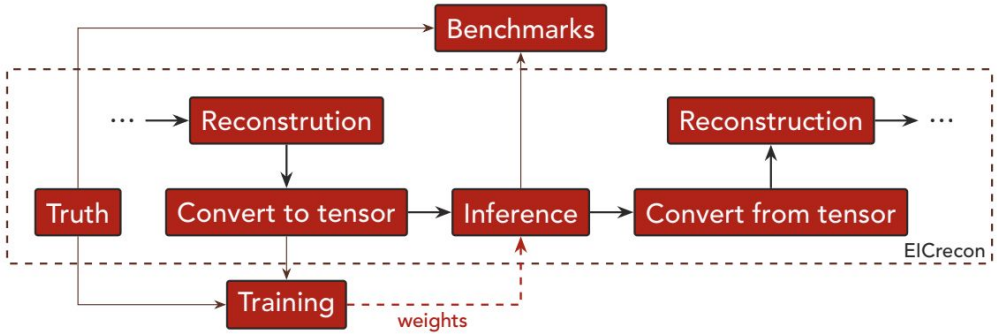
ML infrastructure in EICrecon

October 28, 2024

Dmitry Kalinkin

Some context

- We don’t want to lock our algorithms into a specific inference engine at this time. (Options are: ONNXruntime, TMVA, TMVA-SOFIE, TorchScript, …)
- PODIO records don’t map onto numeric tensors directly, some *input*/*output* transformations will be needed
- Transformed *input* tensors need to be save-able to enable training, it makes sense to also generate *target* values in EICrecon
- This motivates some level of factorization (no pun intended)



Proposal

- Our IO format of choice is PODIO, our approach to composition is to provide factories (e.g. because different factories may be CPU vs GPU bound)
- This warrants addition of a new type for data exchange

```
edm4eic::Tensor:
  Description: "Tensor type for use in training in inference of ML models"
  Author: "D. Kalinkin"
  Members:
    - int32_t           elementType     // Data type in the same encoding as "ONNXTensorElementDataType", 1 - float, 7 - int64
  VectorMembers:
    - int64_t           shape           // Vector of tensor lengths along its axes
    - float             floatData       // Iff elementType==1, values are stored here
    - int64_t           int64Data       // Iff elementType==7, values are stored here
```

- This is largely based off what ONNX defines (can be extended with other data types as will be necessary)
- Unfortunately, it is not clear how to make this zero-copy

Prototype/Roadmap

- EDM4eic PR #96 - provide edm4eic::Tensor extension
- EICrecon PR #1618 - provides generic ONNXruntime interface and EEEMCal PID use case
- detector_benchmarks PR #91 - provides reproducible training
- epic-data PR #22 - provides trained weights in .onnx format for EICrecon inference