TMVA SOFIE

Enhancing the Machine Learning Inference Engine

Sanjiban Sengupta
EP-SFT

Supervisor: Lorenzo Moneta

/sanjibansg /sanjiban-sengupta

ROOT TMVA

- Toolkit for Multivariate Analysis
- Provides a Machine Learning environment for training, testing and evaluation of multivariate methods.

TMVA SOFIE

SOFIE

SOFIE

SOFIE

System for

SOFIE

System for Optimized

SOFIE

System for Optimized Fast

SOFIE

System for Optimized Fast Inference

SOFIE

System for Optimized Fast Inference code Emit

SOFIE

System for Optimized Fast Inference code Emit

inference code, fast to operate, with least dependencies

Motivation

Motivation

- ML ecosystem mostly focuses on model training.

Motivation

- ML ecosystem mostly focuses on model training.
- Inference in Tensorflow & PyTorch
  - supports only their own model
  - usage of C++ environment is difficult
  - heavy dependency

Motivation

- ML ecosystem mostly focuses on model training.
- Inference in Tensorflow & PyTorch
  - supports only their own model
  - usage of C++ environment is difficult
  - heavy dependency
- Inference in ONNX (Open Neural Network Exchange)
  - can use ONNXRuntime by Microsoft
  - large dependency
  - difficult to integrate in HEP applications
    - control of libraries, threads
    - not optimized for single event evaluation

TMVA SOFIE

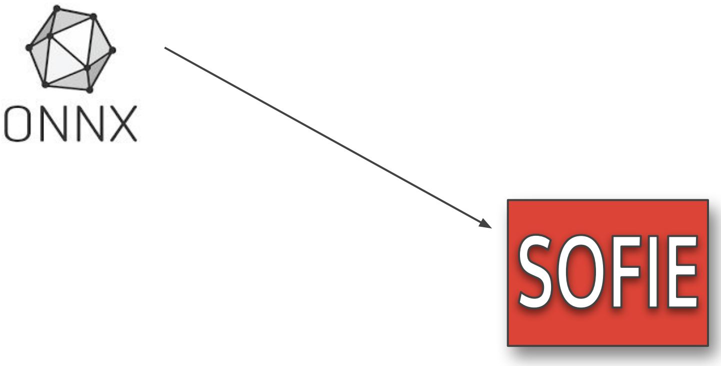
- Intermediate representation following ONNX standards.

TMVA SOFIE

- Intermediate representation following ONNX standards.
- Inference code generation with least latency and minimal dependency

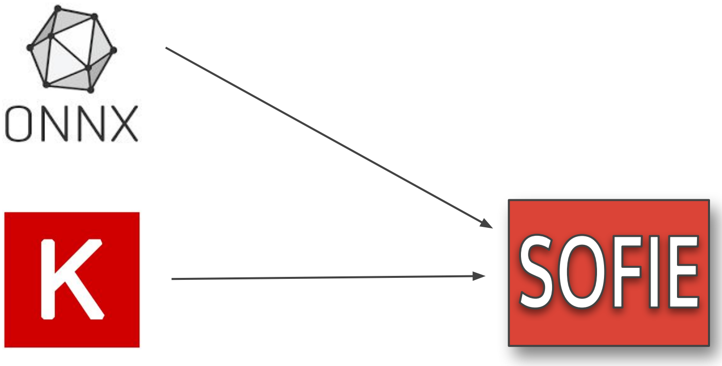
TMVA SOFIE

- Intermediate representation following ONNX standards.
- Inference code generation with least latency and minimal dependency



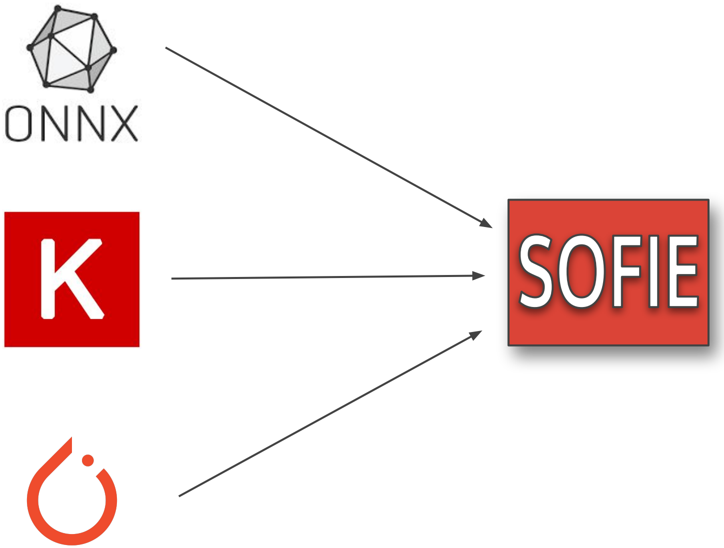
TMVA SOFIE

- Intermediate representation following ONNX standards.
- Inference code generation with least latency and minimal dependency



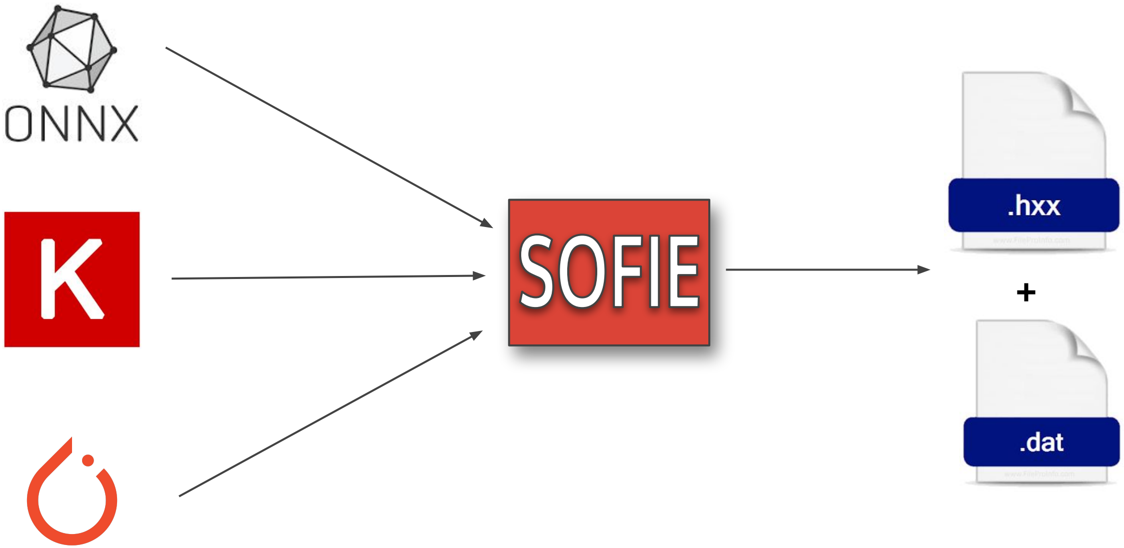
TMVA SOFIE

- Intermediate representation following ONNX standards.
- Inference code generation with least latency and minimal dependency

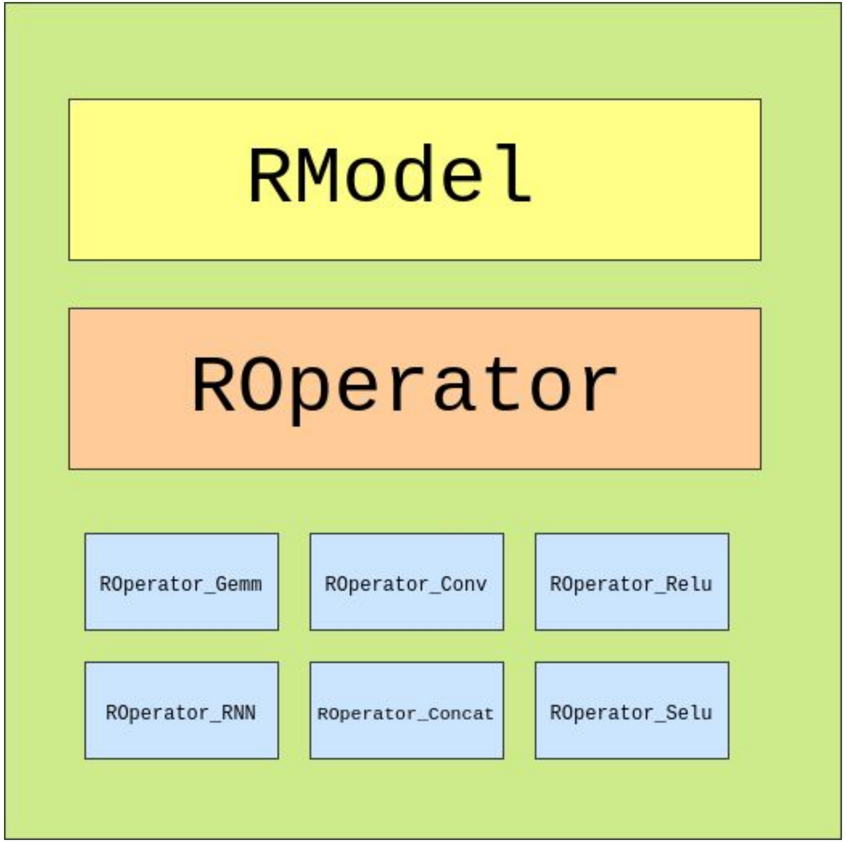


TMVA SOFIE

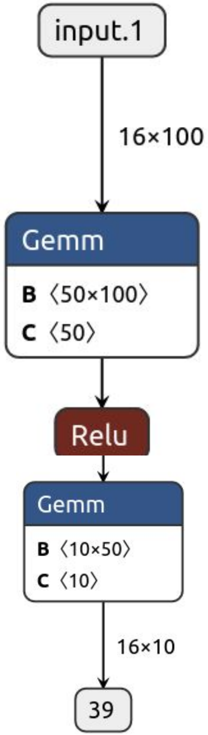
- Intermediate representation following ONNX standards.
- Inference code generation with least latency and minimal dependency



TMVA SOFIE


RModel
ROperator
ROperator_Gemm
ROperator_Conv
ROperator_Relu
ROperator_RNN
ROperator_Concat
ROperator_Selu

TMVA SOFIE



TMVA SOFIE

Parser for translating an ONNX model to SOFIE's IR

```
using namespace TMVA::Experimental::SOFIE;
RModelParser_ONNX parser;
RModel model = parser.Parse("model.onnx");
```

TMVA SOFIE

Parser for translating an ONNX model to SOFIE's IR

```
using namespace TMVA::Experimental::SOFIE;
RModelParser_ONNX parser;
RModel model = parser.Parse("model.onnx");
```

Parser for translating Keras (`.h5`) models

```
SOFIE::RModel model = SOFIE::PyKeras::Parse("KerasModel.h5");
```

TMVA SOFIE

Parser for translating an ONNX model to SOFIE's IR

```
using namespace TMVA::Experimental::SOFIE;
RModelParser_ONNX parser;
RModel model = parser.Parse("model.onnx");
```

Parser for translating Keras (`.h5`) models

```
SOFIE::RModel model = SOFIE::PyKeras::Parse("KerasModel.h5");
```

Inference code generation

```
// generate text code internally (with some options)
model.Generate();
// write output header file and data weight file
model.OutputGenerated();
```

SOFIE's Generated code

```
// Code auto generated by TMVA SOFIE

namespace TMVA_SOFIE_Linear_event{

struct Session {

Session(std::string filename ="") {
 if (filename.empty()) filename = "Linear_event.dat";
 std::ifstream f;
 f.open(filename);
 // read weight data file
 ………………….. . .
}
std::vector<float> infer(float* tensor_input1){
 ………………….. . .
```

Project Objectives

Project Objectives

- Extending support of SOFIE Keras parser

Project Objectives

- Extending support of SOFIE Keras parser
- Implement SOFIE Custom operator support

Project Objectives

- Extending support of SOFIE Keras parser
- Implement SOFIE Custom operator support
- Implement support for parsing Graph Neural Networks in SOFIE

SOFIE Keras Parser

SOFIE Keras Parser

- No native support for ONNX translation

SOFIE Keras Parser

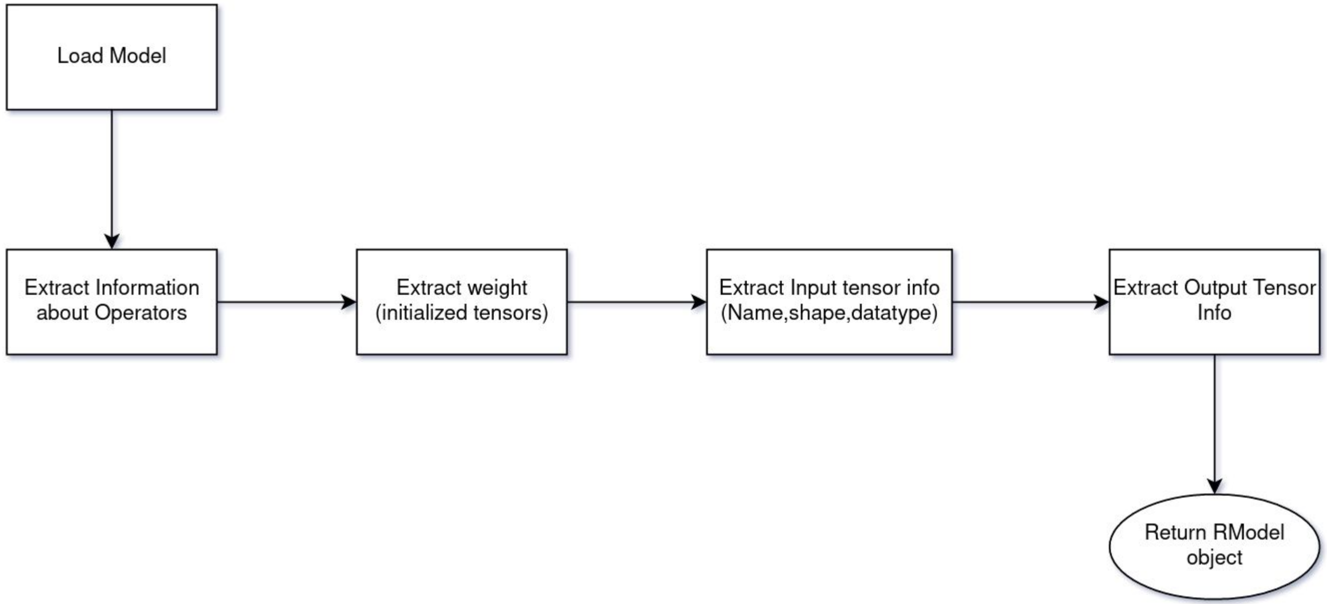
- No native support for ONNX translation
- `TF2ONNX` may convert a Keras .h5 model to ONNX

SOFIE Keras Parser

- No native support for ONNX translation
- `TF2ONNX` may convert a Keras .h5 model to ONNX
- SOFIE Keras Parser!
    - simpler to use
    - no need for input spec
    - built on latest opset

SOFIE Keras Parser

Algorithm for Parser



SOFIE Keras Parser

Current Support

Keras Layer	Status
Dense	Implemented & Integrated
Permute	Implemented & Integrated
ReLU, Selu, Sigmoid	Implemented & Integrated
Batch Normalization	PR Merged
Convolution (2D)	PR Merged
Reshape	PR Merged
Basic Binary Operators: Add, Subtract, Multiply	PR Under Review
Activations: Softmax, LeakyRelu, Tanh	PR Drafted
Concat	PR Drafted

SOFIE Custom Operator

SOFIE Custom Operator

- ONNX standards specifies 183 operators currently.

SOFIE Custom Operator

- ONNX standards specifies 183 operators currently.
- Need a custom user operator specification

SOFIE Custom Operator

- ONNX standards specifies 183 operators currently.
- Need a custom user operator specification
    - simple to define

SOFIE Custom Operator

- ONNX standards specifies 183 operators currently.
- Need a custom user operator specification
  - simple to define
  - easy to test, debug, and evaluate

SOFIE Custom Operator

- ONNX standards specifies 183 operators currently.
- Need a custom user operator specification
  - simple to define
  - easy to test, debug, and evaluate
  - few overheads and dependencies

SOFIE Custom Operator

Definition

SOFIE Custom Operator

Definition

- Define custom operator with the required attributes
    - Operator name
    - Input tensor names
    - Output tensor names
    - Output tensor shapes
    - Header file name

SOFIE Custom Operator

Definition

- Define custom operator with the required attributes
  - Operator name
  - Input tensor names
  - Output tensor names
  - Output tensor shapes
  - Header file name
- Define Compute function in Header file

SOFIE Custom Operator

Definition

- Define custom operator with the required attributes
  - Operator name
  - Input tensor names
  - Output tensor names
  - Output tensor shapes
  - Header file name
- Define Compute function in Header file

SOFIE GNN Support

SOFIE GNN Support

- High demands of Graph Neural Networks in High energy physics research

SOFIE GNN Support

- High demands of Graph Neural Networks in High energy physics research
- CMS

SOFIE GNN Support

- High demands of Graph Neural Networks in High energy physics research
- CMS
    - uses Particlenet; graph neural network supporting graph convolution, i.e. edge convolution and dynamic graph CNN methods

SOFIE GNN Support

- High demands of Graph Neural Networks in High energy physics research
- CMS
  - uses Particlenet; graph neural network supporting graph convolution, i.e. edge convolution and dynamic graph CNN methods
- LHCb

SOFIE GNN Support

- High demands of Graph Neural Networks in High energy physics research
- CMS
  - uses Particlenet; graph neural network supporting graph convolution, i.e. edge convolution and dynamic graph CNN methods
- LHCb
  - plans to use DeepMind’s Graph Nets library; builds GNN on top of Tensorflow & Sonnet

SOFIE GNN Support

- High demands of Graph Neural Networks in High energy physics research
- CMS
    - uses Particlenet; graph neural network supporting graph convolution, i.e. edge convolution and dynamic graph CNN methods
- LHCb
    - plans to use DeepMind's Graph Nets library; builds GNN on top of Tensorflow & Sonnet

SOFIE GNN Support

Current Plans & Implementation

Current Plans & Implementation

- Following the DeepMind's Graph Nets architecture

SOFIE GNN Support

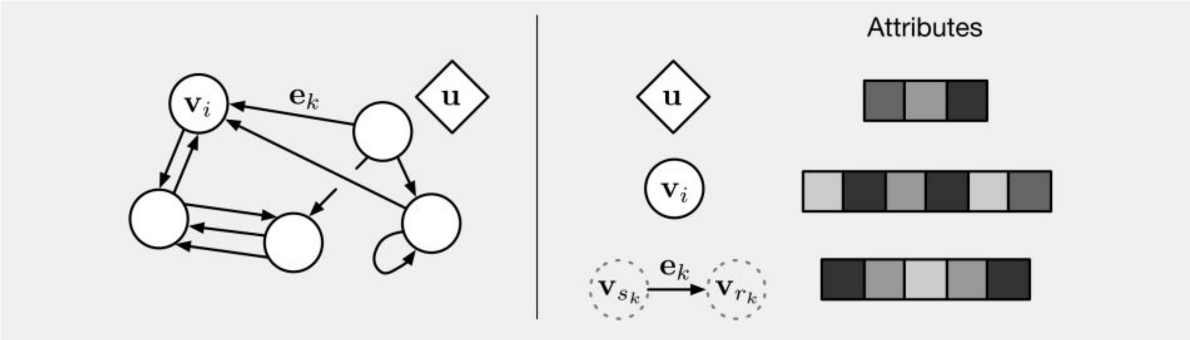
Current Plans & Implementation

- Following the DeepMind's Graph Nets architecture
- Intermediate Representation
  - Nodes list
  - Edges list
  - Sender's list
  - Receiver's list
  - Global values

SOFIE GNN Support

Current Plans & Implementation

- Following the DeepMind's Graph Nets architecture
- Intermediate Representation
  - Nodes list
  - Edges list
  - Sender's list
  - Receiver's list
  - Global values



SOFIE GNN Support

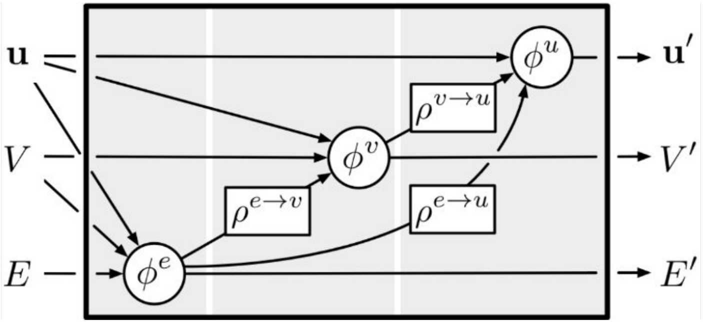
Current Plans & Implementation

- Following the DeepMind's Graph Nets architecture
- Intermediate Representation
  - Nodes list
  - Edges list
  - Sender's list
  - Receiver's list
  - Global values

SOFIE GNN Support

Current plans for implementation

- Following the DeepMind's Graph Nets architecture
- Intermediate Representation
  - Nodes list
  - Edges list
  - Sender's list
  - Receiver's list
  - Global values



- Operating functions
  - Updation functions
  - Aggregation functions

Future Plan

Future Plan

- Finishing implementation & integration of Graph Neural Network support in SOFIE

Future Plan

- Finishing implementation & integration of Graph Neural Network support in SOFIE
- Implementing support for new operators in TMVA SOFIE

Conclusion

- Link to Forked Repository
  github.com/sanjibansg/root
- Link to SOFIE in current ROOT master
  github.com/root-project/root/tree/master/tmva/sofie
- Link to TMVA/SOFIE tutorials
  root.cern.ch/doc/master/group__tutorial__tmva.html
- Link to SOFIE notebooks
  github.com/lmoneta/tmva-tutorial/tree/master/sofie

Using SOFIE’s Generated code

```
// Code generated automatically by TMVA for Inference of Model file [model.h5] at [Wed Aug  3 20:32:37
2022]

#include “Model.hxx”
// create session class
TMVA_SOFIE_Model::Session s();
//—- event loop
…….
{
// evaluate model: input is an array of type float *
std::vector<float> result = s.infer(input);
}
```

Using SOFIE's Generated code

```
import ROOT
# compile generate SOFIE code using ROOT interpreter
ROOT.gInterpreter.Declare('#include "Model.hxx"')
# create session class
s = ROOT.TMVA_SOFIE_Model.Session()
#-- event loop
…….
 # evaluate the model , input can be a numpy array of type float32
 result = s.infer(input)
```

SOFIE Custom Operator

Definition

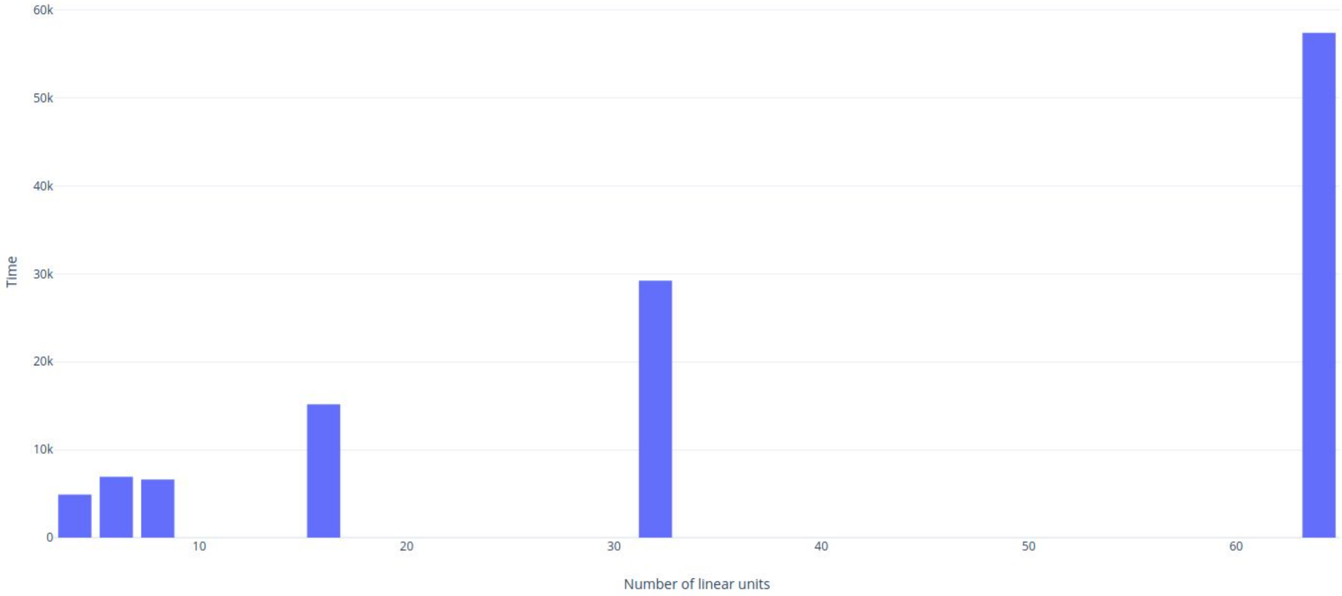
- Define custom operator with the required attributes
  - Operator name
  - Input tensor names
  - Output tensor names
  - Output tensor shapes
  - Header file name
- Define Compute function in Header file

Interface

```
std::unique_ptr<SOFIE::ROperator> op;
op.reset(new SOFIE::ROperator_Custom<float>("Exp", {"denseBiasAdd0"}, {"exp_out"}, {{1,4}},
"exp_compute.hxx"));
```

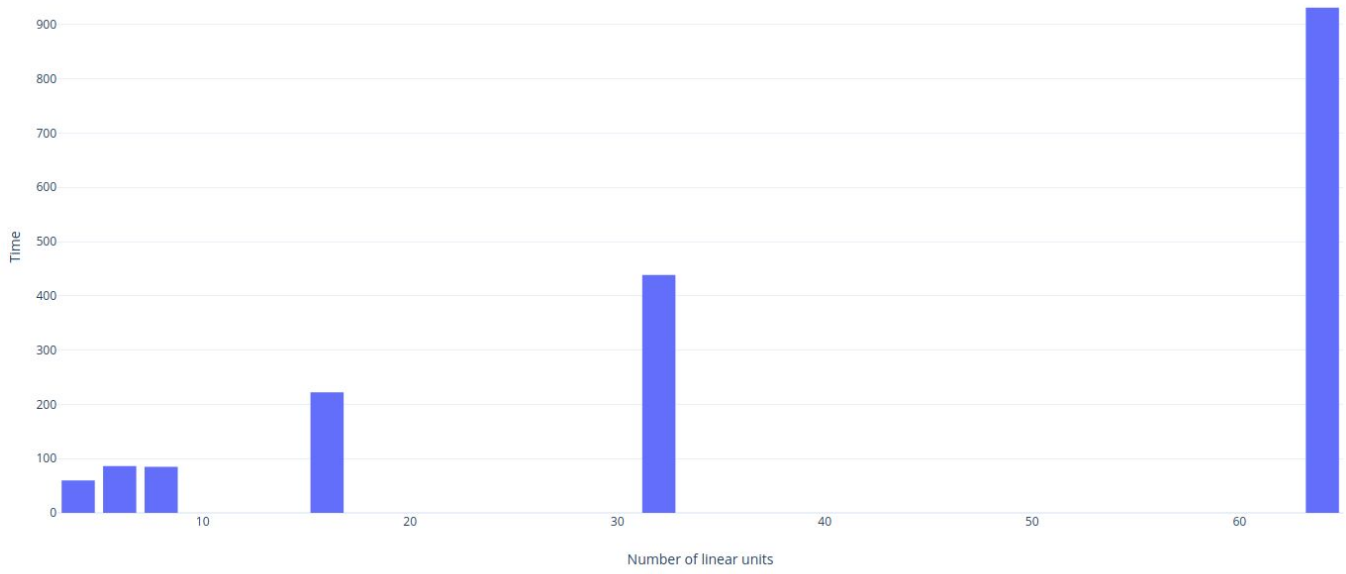
SOFIE Keras Parser

Benchmarking for Keras

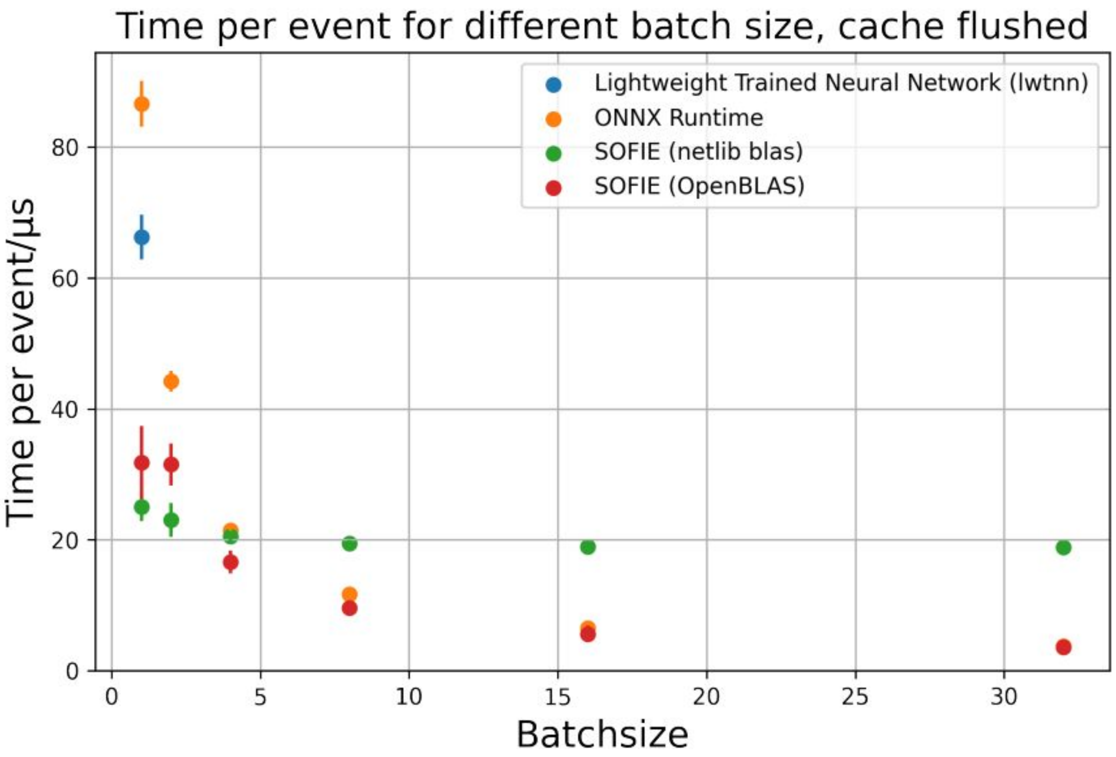


SOFIE Keras Parser

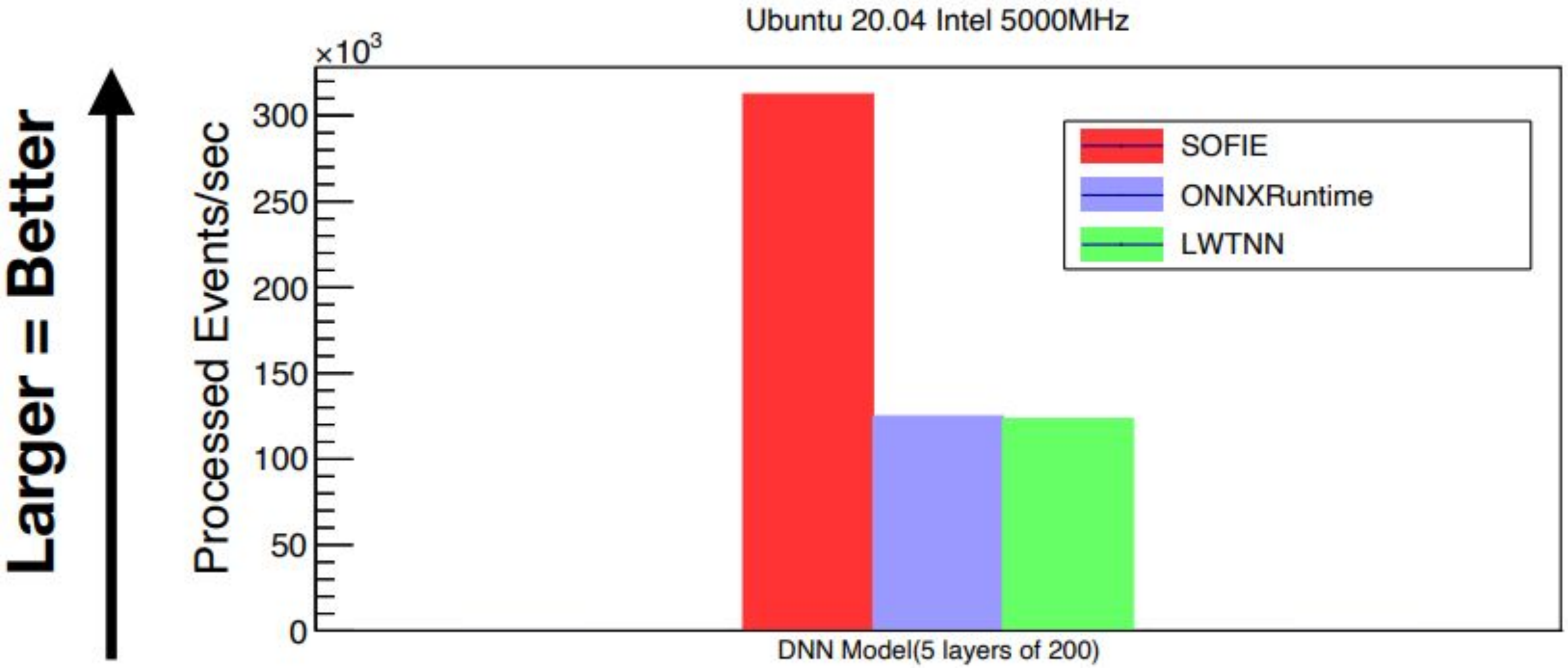
Benchmarking for SOFIE



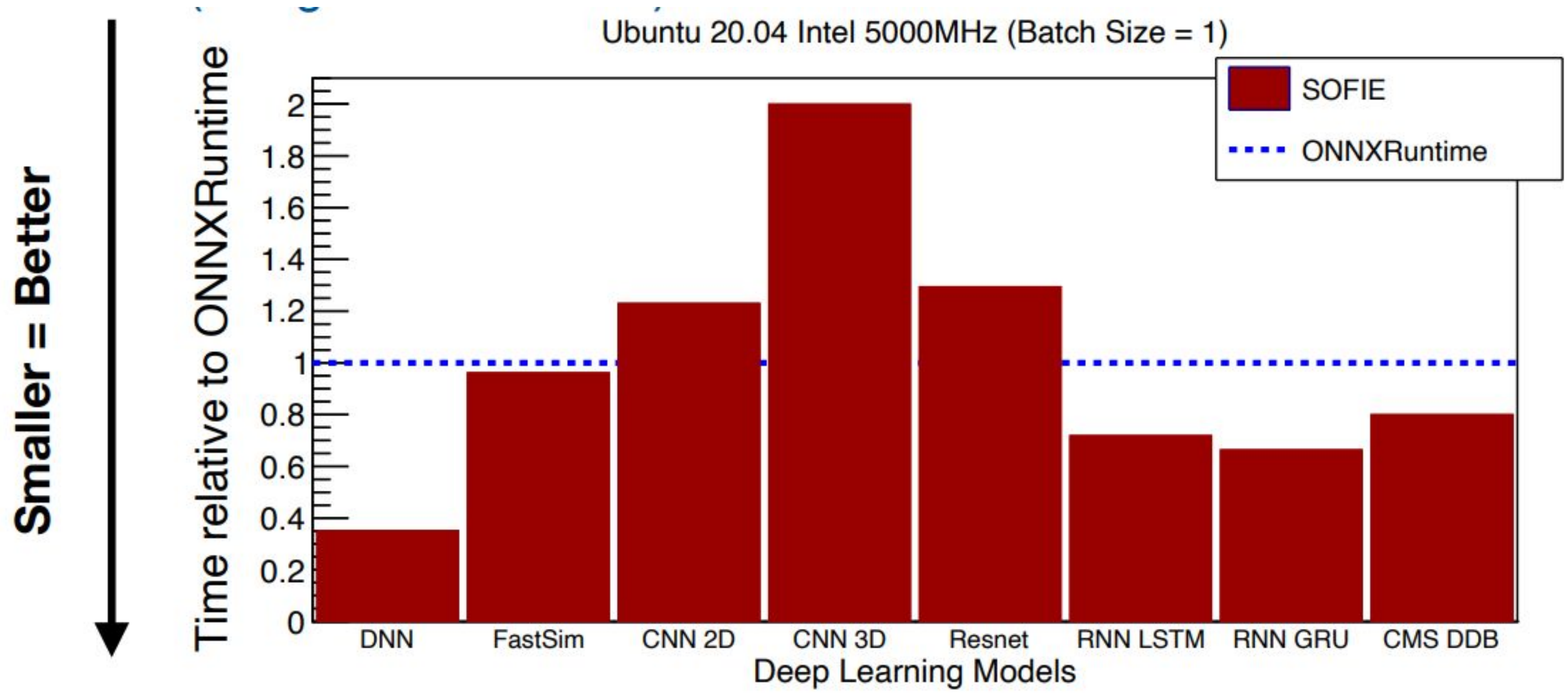
TMVA SOFIE


Time per event for different batch size, cache flushed
Lightweight Trained Neural Network (lwtnn)
ONNX Runtime
SOFIE (netlib blas)
SOFIE (OpenBLAS)
Time per event/μs
0
20
40
60
80
0
5
10
15
20
25
30
Batchsize

TMVA SOFIE


Larger = Better
Ubuntu 20.04 Intel 5000MHz
$\times 10^3$
Processed Events/sec
300
250
200
150
100
50
0
SOFIE
ONNXRuntime
LWTNN
DNN Model(5 layers of 200)

TMVA SOFIE



Smaller = Better
Ubuntu 20.04 Intel 5000MHz (Batch Size = 1)
Time relative to ONNXRuntime
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
SOFIE
ONNXRuntime
DNN
FastSim
CNN 2D
CNN 3D
Resnet
RNN LSTM
RNN GRU
CMS DDB
Deep Learning Models