Optimal Quantization

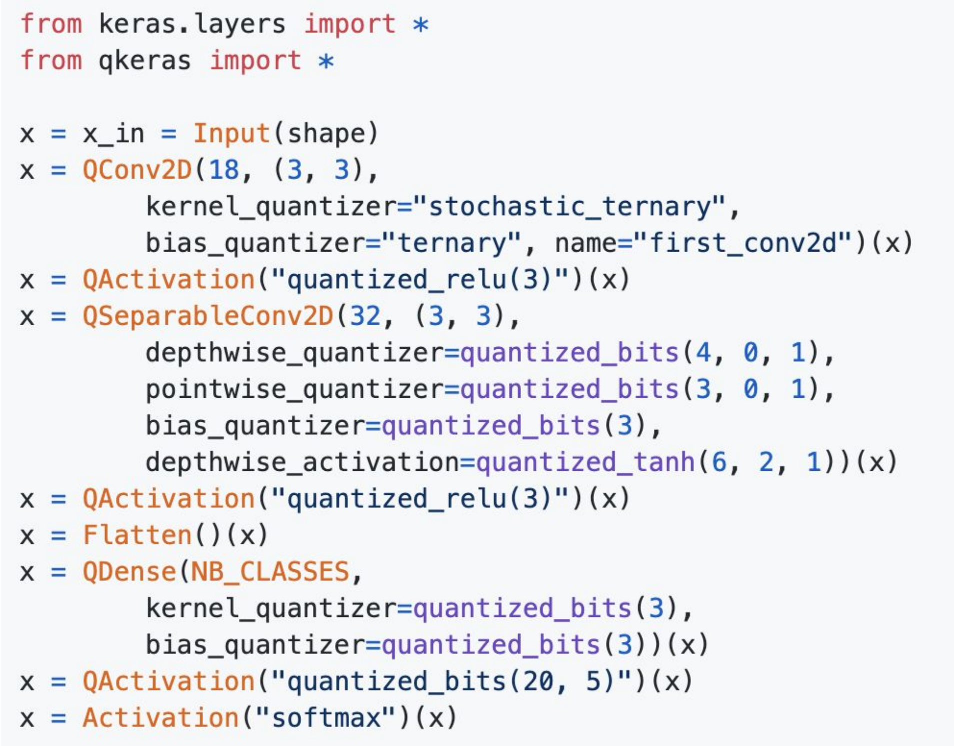
Quantization

- Reduce bit widths of weights, activations and biases
- Use integer values rather than floating point
- Lower precision than floating point
- Lower energy consumption, memory footprint, computation latency
- Implemented in Tensorflow via the QKeras library

The Problem: How many bits?

- Anomaly detection at L1 trigger in CMS:
  - High accuracy
  - Low resource consumption
  - Fast inference

- Hyperparameter optimization for
  - Model size
  - Energy utilization
  - Custom metric (sensitivity at specificity)



```
from keras.layers import *
from qkeras import *

x = x_in = Input(shape)
x = QConv2D(18, (3, 3),
        kernel_quantizer="stochastic_ternary",
        bias_quantizer="ternary", name="first_conv2d")(x)
x = QActivation("quantized_relu(3)")(x)
x = QSeparableConv2D(32, (3, 3),
        depthwise_quantizer=quantized_bits(4, 0, 1),
        pointwise_quantizer=quantized_bits(3, 0, 1),
        bias_quantizer=quantized_bits(3),
        depthwise_activation=quantized_tanh(6, 2, 1))(x)
x = QActivation("quantized_relu(3)")(x)
x = Flatten()(x)
x = QDense(NB_CLASSES,
        kernel_quantizer=quantized_bits(3),
        bias_quantizer=quantized_bits(3))(x)
x = QActivation("quantized_bits(20, 5)")(x)
x = Activation("softmax")(x)
```

The Tools

AutoQKeras:

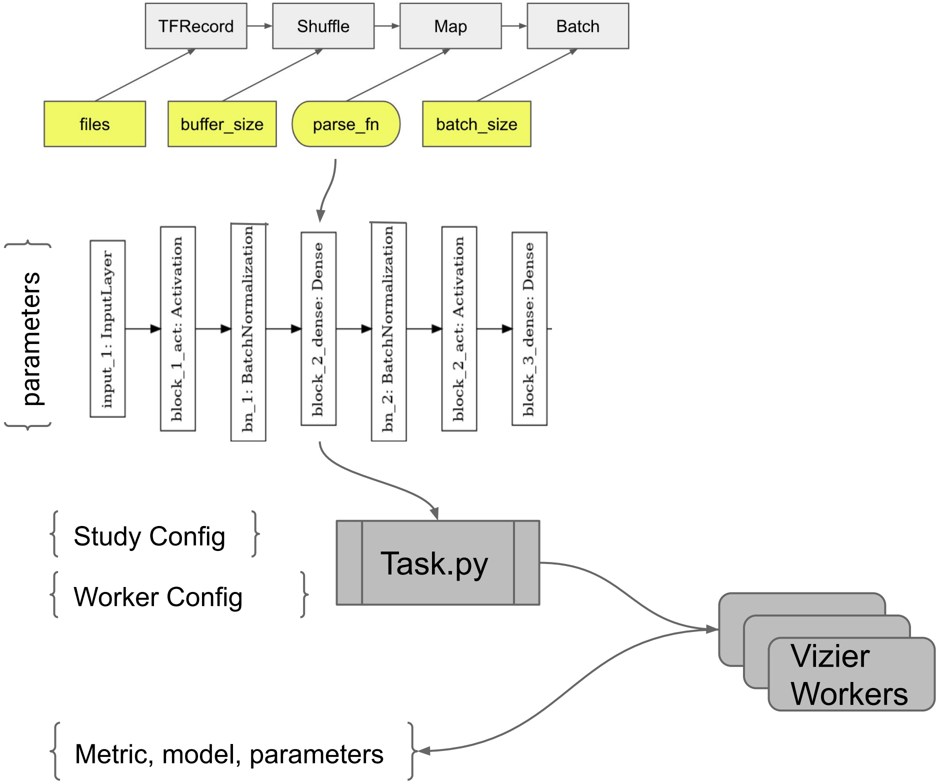
- In QKeras package
- QTools to estimate energy consumption
- Bayesian, Hyperband, GridSearch
- Can use multiple accelerators in 1 machine (GPUs, TPUs)
- CANNOT run through several scenarios in parallel, limited to 1 node.

Vizier:

- Google cloud service
- “Black box” optimization
- Single and multiple objectives (beta)
- Train on multiple accelerators per node
- Evaluate multiple scenarios in parallel on several nodes
- NOT specific to QKeras or even hyperparameter tuning

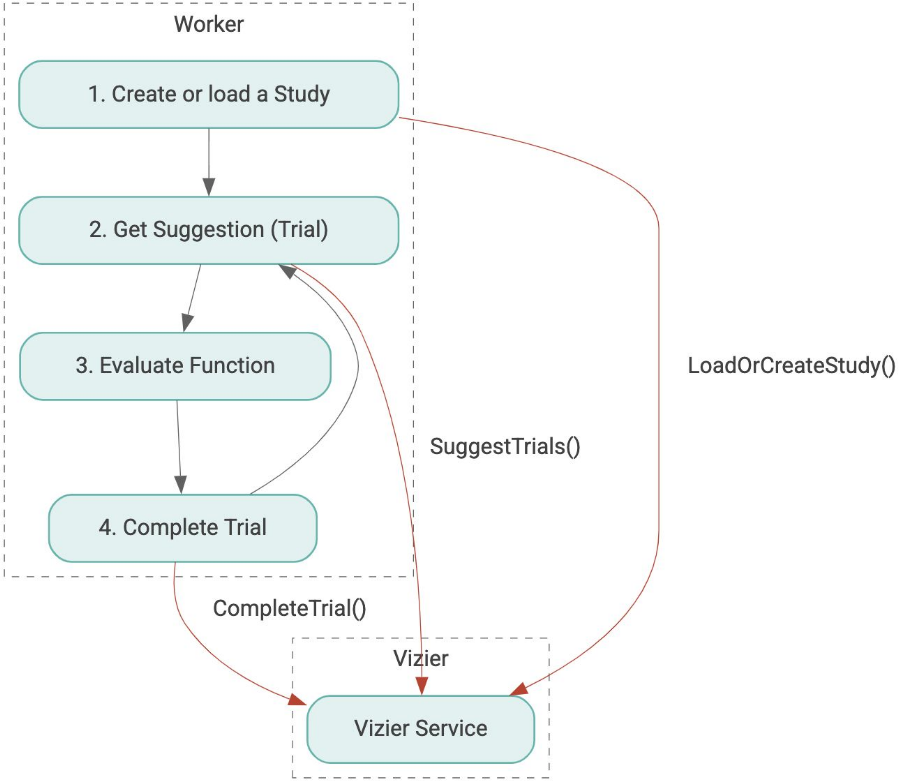
The Process

1. Build the pipe
2. Build the graph
3. Run the job
4. Get the result



Google

Vizier



Why Vizier?

Several algorithms available:

- Batched Gaussian process bandits
- Grid search
- Random search

Automated early stopping decisions

Transfer learning from prior studies

Pareto-optimal solutions (beta)

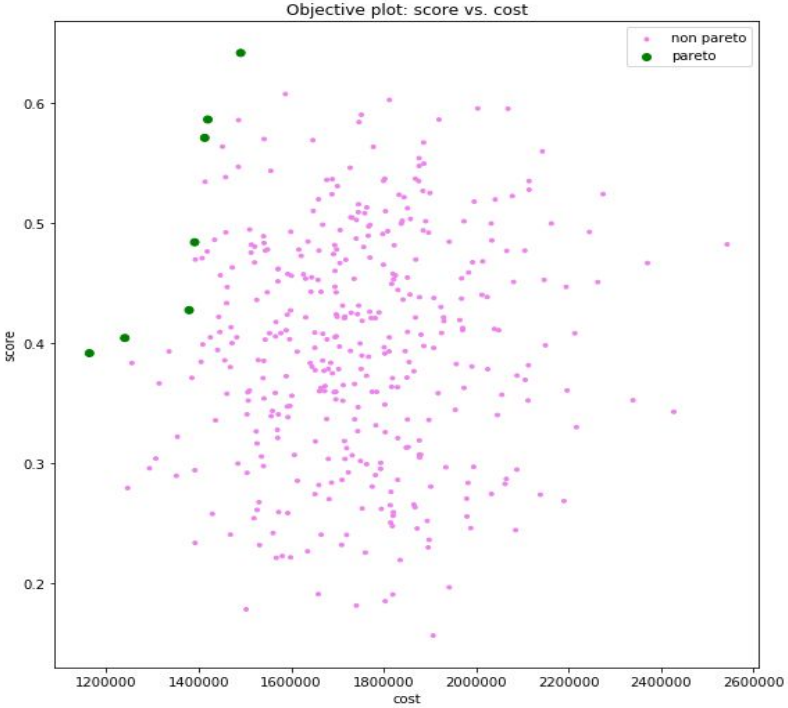
Demo

Result Example

- 1 Objective, 12GB data

AutoQKeras: 1 8-core node, 1 V100, 10 trials	Vizier: 4 x 8-core nodes, 1 V100 each, 10 trials
4 days	47 minutes

- 2 Objectives, Pareto frontier



Q&A

References

C. N. Coelho Jr., Aki Kuusela, Hao Zhuang, Thea Aarrestad, Vladimir Loncar, Jennifer Ngadiuba, Maurizio Pierini, Sioni Summers, "Ultra Low-latency, Low-area Inference Accelerators using Heterogeneous Deep Quantization with QKeras and hls4ml", http://arxiv.org/abs/2006.10159v1

Erwei Wang, James J. Davis, Daniele Moro, Piotr Zielinski, Claudionor Coelho, Satrajit Chatterjee, Peter Y. K. Cheung, George A. Constantinides, "Enabling Binary Neural Network Training on the Edge", https://arxiv.org/abs/2102.04270

Golovin, Daniel, et al. "Google vizier: A service for black-box optimization." *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017. https://dl.acm.org/doi/pdf/10.1145/3097983.3098043

QKeras Documentation: https://github.com/google/qkeras

Vizier Documentation: https://cloud.google.com/vertex-ai/docs/vizier/overview?hl=en

Next Steps

1. QKeras: a quantization deep learning library for Tensorflow Keras
2. Training Keras models with TensorFlow Cloud
3. HP Tuning on Google Cloud with CloudTuner
4. Vertex AI: Hyperparameter Tuning
5. Vertex AI documentation