Del Algoritmo al Hardware:
Aprendizaje Automático en Sistemas Embebidos

From Algorithm to Hardware: Machine Learning in Embedded Systems

1 al 11 de Abril, 2025. Universidad Nacional de Mar del Plata - Mar del Plata - Argentina.

Model Compression For Machine Learning-based Models: Pruning, Quantization, and Knowledge Distillation

Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Mar del Plata, Argentina - 2025 -

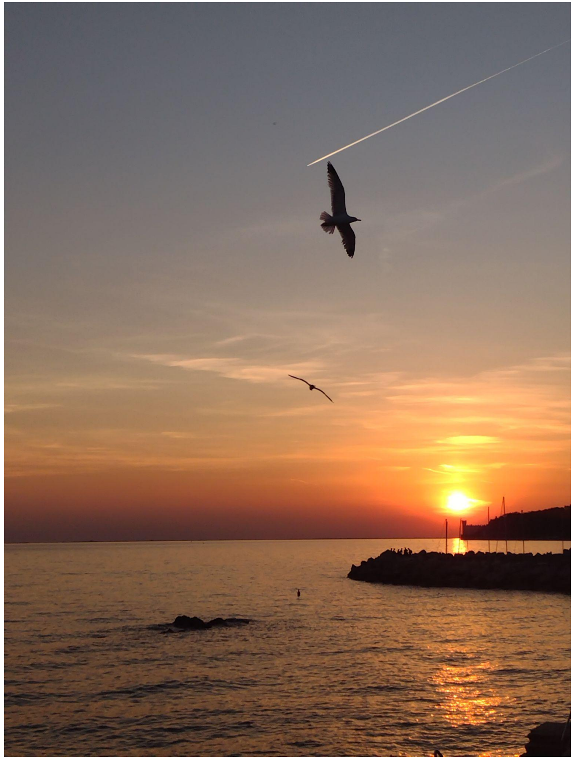
Universidad Nacional de Mar del Plata

Fundación Williams

ICTP The Abdus Salam International Centre for Theoretical Physics

Outline

- ML and model compression techniques.
- Pruning.
- Quantization.
- Knowledge distillation.
- How do we combine compression techniques?.
- Hyperparameters tuning and compression.
- Demo: MNIST-based binary classification - QAP -



Machine Learning
and model compression techniques

ML and model compression techniques

- *Question: What is compression?*

ML and model compression techniques

ML and model compression techniques

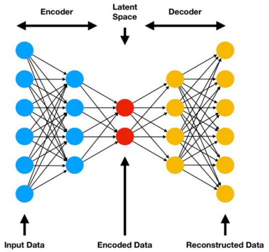
- **Compression** is the process of reducing the size of data, often with the aim of preserving important information or, in some cases, enabling perfect reconstruction. The goal is to make data storage or transmission more efficient by using fewer resources (e.g., memory, storage, bandwidth).

ML and model compression techniques

- *Question: Why is important in Machine Learning?*

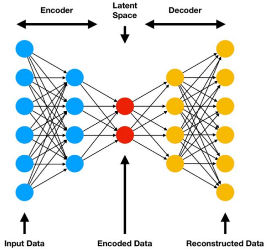
ML and model compression techniques

- **Compression,** in the context of Machine Learning, involves techniques aiming to reduce the size of models or datasets while preserving performance.



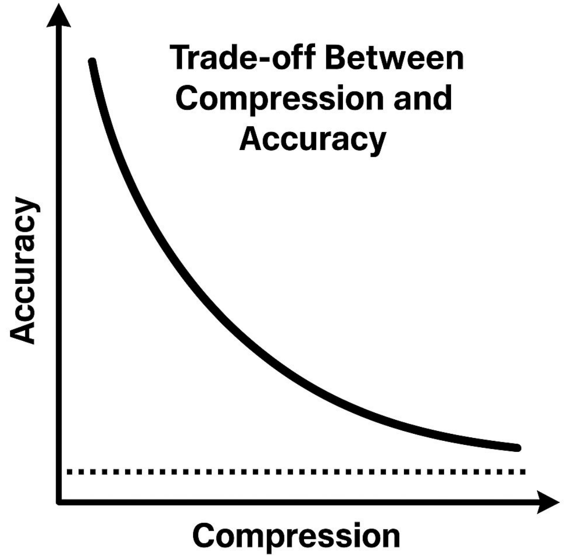
ML and model compression techniques

- **Compression,** in the context of Machine Learning, involves techniques aiming to reduce the size of models or datasets while preserving performance.
    - **Model Compression:** Reducing the size of machine learning models (such as neural networks) without significantly affecting their accuracy.
    - **Data compression:** Reducing the size of the data used for training, validation, and testing. Example: Autoencoders.



ML and model compression techniques

- **Trade-off Between Compression and Accuracy**
  - Balance between reducing the size of the model or data and maintaining its performance.



ML and model compression techniques

- **Applicability**
  - Not all compression techniques are suitable for every type of model or dataset.

ML and model compression techniques

- **Applicability**
    - Not all compression techniques are suitable for every type of model or dataset.
    - The decision of which compression strategy to apply depends on factors such as the desired trade-off between model size and performance, the computational resources available, and the nature of the data being processed.

ML and model compression techniques

- **Problem:**
  - **Very large models → high memory consumption and inference time.**

The most accurate models (such as deep neural networks) are large and expensive in terms of memory and processing time.

ML and model compression techniques

The most accurate models (such as deep neural networks) are large and expensive in terms of memory and processing time.

Model	Parameters (millions)	Disk size (MB)
ResNet-50	~25.6M	~98 MB
BERT-base	~110M	~440 MB
BERT-large	~340M	~1.3 GB
VGG-16	~138M	~528 MB
VGG-19	~144M	~548 MB
YOLOv3	~62M	~236 MB
SpineNet-49S (small)	~11M	~45 MB
MobileNetV3-Large	~5.4M	~20 MB

ML and model compression techniques

VGG-16

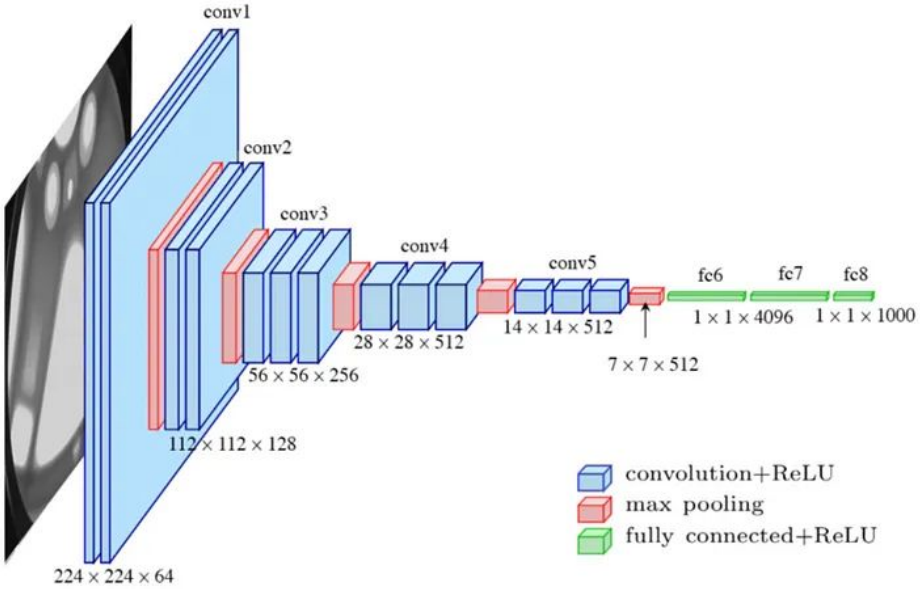


Image from https://lekhuyen.medium.com/an-overview-of-vgg16-and-nin-models-96e4bf398484

ML and model compression techniques

ResNet-50

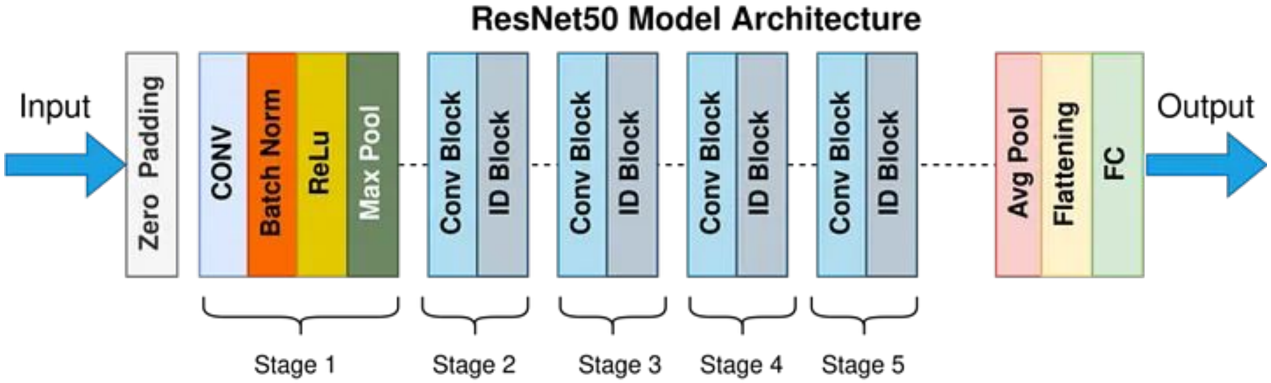


Image from https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f

ML and model compression techniques

SpineNet-49

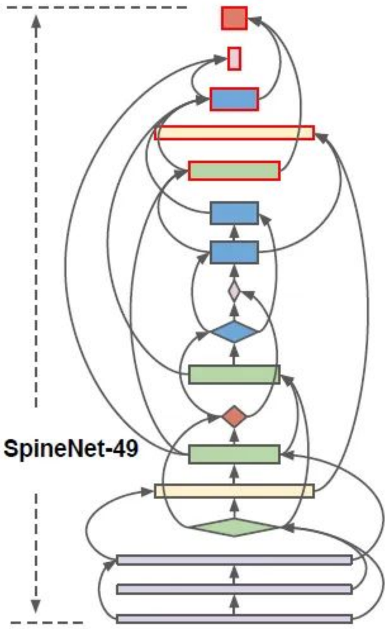
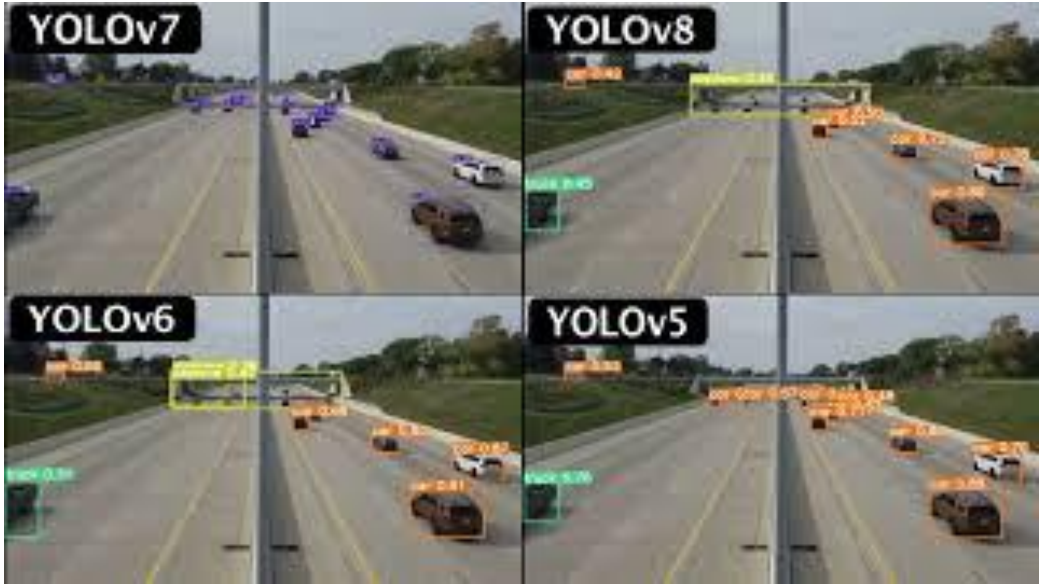


Image from Du, X., Lin, T. Y., Jin, P., Ghiasi, G., Tan, M., Cui, Y., ... & Song, X. (2020). Spinenet: Learning scale-permuted backbone for recognition and localization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11592-11601).

ML and model compression techniques



ML and model compression techniques

- Problem:
  - Very large models → high memory consumption and inference time.
- Solution:
  - Compression

ML and model compression techniques

Distilled versions

Model	Parameters (millions)	Disk size (MB)	Accuracy
DistilBERT	66M (↓40%)	~250 MB	97% of BERT
SqueezeNet	1.2M (↓99%)	~4.8 MB (↓99%)	Similar to VGG-16
YOLOv8-Nano	3.2M (↓92%)	~8 MB (↓90%)	Good trade-off with YOLOv8-L
YOLO-Fastest	0.2M (↓99.5%)	~1 MB (↓99%)	Lower accuracy

Pruning, Quantization, and Knowledge Distillation

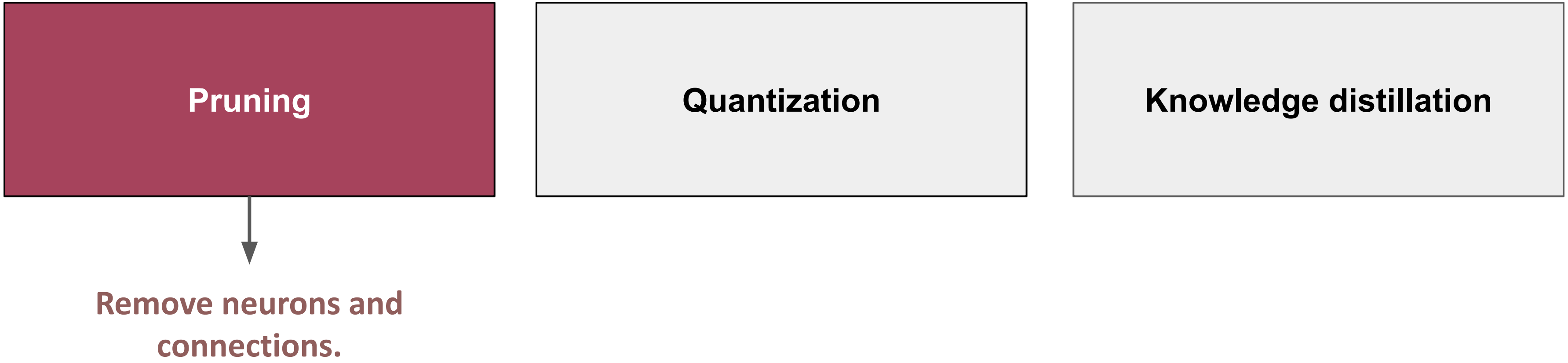
ML and model compression techniques

Pruning

Quantization

Knowledge distillation

ML and model compression techniques



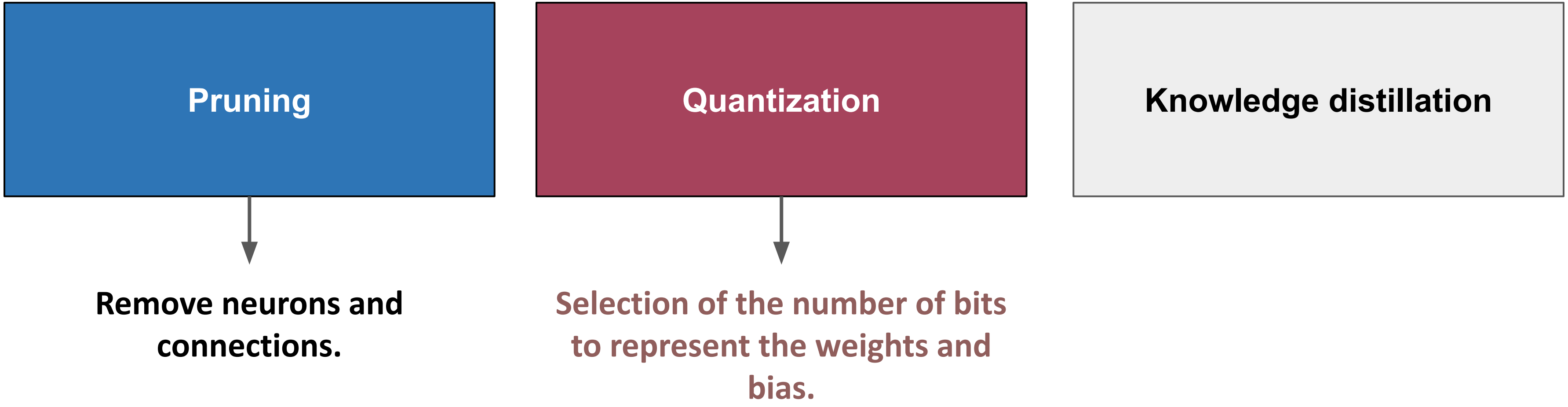
Pruning

Quantization

Knowledge distillation

Remove neurons and connections.

ML and model compression techniques



Pruning

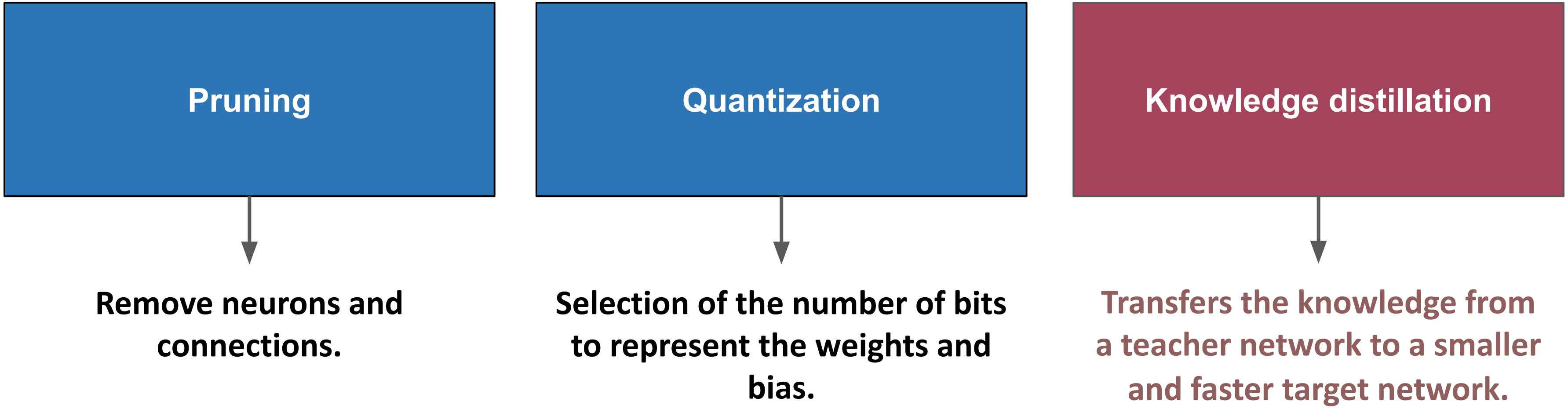
Quantization

Knowledge distillation

Remove neurons and connections.

Selection of the number of bits to represent the weights and bias.

ML and model compression techniques



Pruning

Remove neurons and connections.

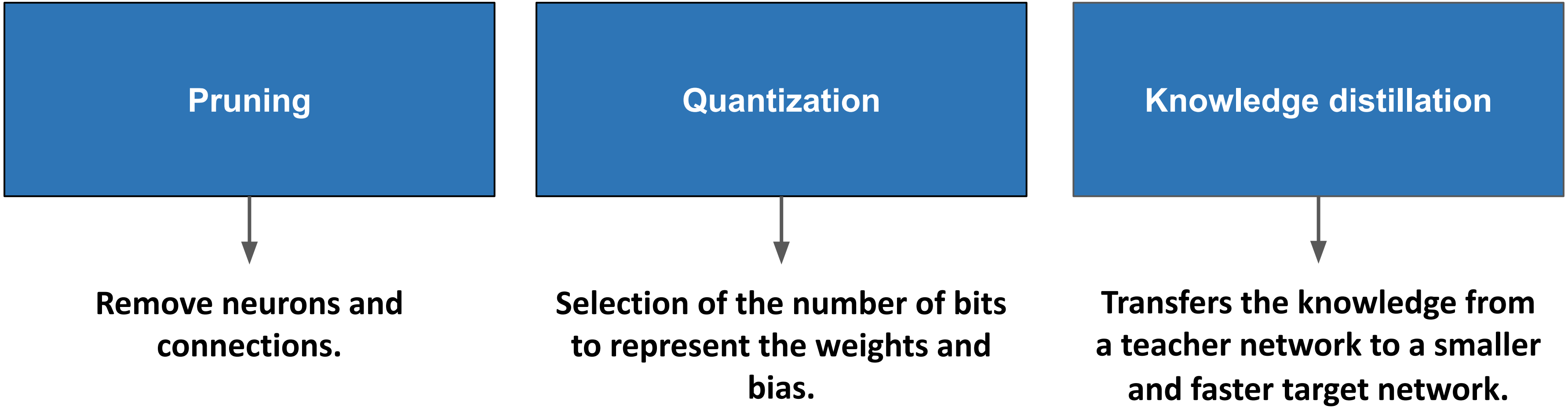
Quantization

Selection of the number of bits to represent the weights and bias.

Knowledge distillation

Transfers the knowledge from a teacher network to a smaller and faster target network.

ML and model compression techniques



Pruning

Remove neurons and connections.

Quantization

Selection of the number of bits to represent the weights and bias.

Knowledge distillation

Transfers the knowledge from a teacher network to a smaller and faster target network.

Fully on-chip deployment

Pruning

ML and model compression techniques

Pruning

- This technique is used to **reduce the size and complexity** of a deep learning model by eliminating unnecessary weights or neurons.

ML and model compression techniques

Pruning

- This technique is used to **reduce the size and complexity** of a deep learning model by eliminating unnecessary weights or neurons.
- The primary objective is to **enhance the model's efficiency by decreasing memory usage and speeding up inference times**, all while maintaining its performance.

ML and model compression techniques

Pruning

- This technique is used to **reduce the size and complexity** of a deep learning model by eliminating unnecessary weights or neurons.
- The primary objective is to **enhance the model's efficiency by decreasing memory usage and speeding up inference times**, all while maintaining its performance.
- **Prune weights:** setting the weights of chosen individual parameters in to zero.

ML and model compression techniques

Pruning

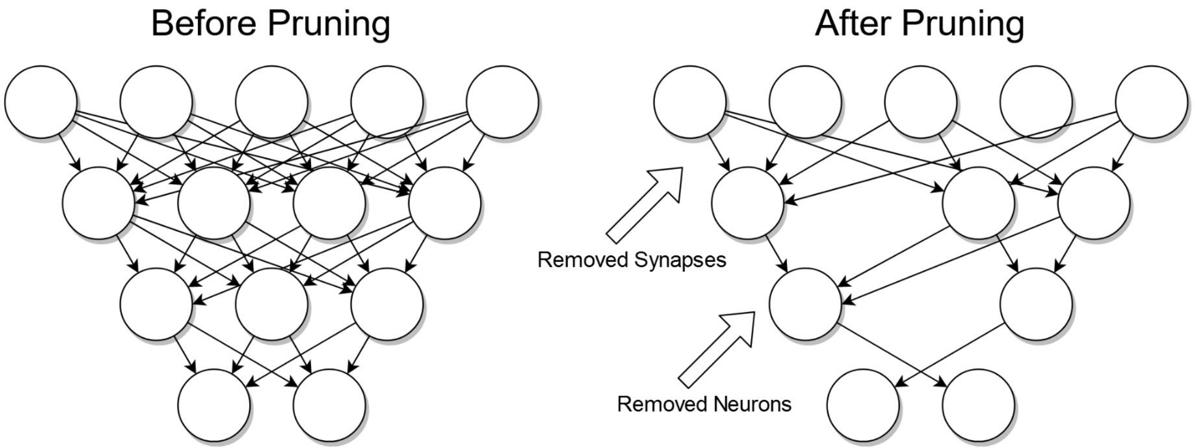


Image from https://en.wikipedia.org/wiki/Decision_tree_pruning#:~:text=Pruning%20is%20a%20data%20compression,and%20redundant%20to%20classify%20instances.

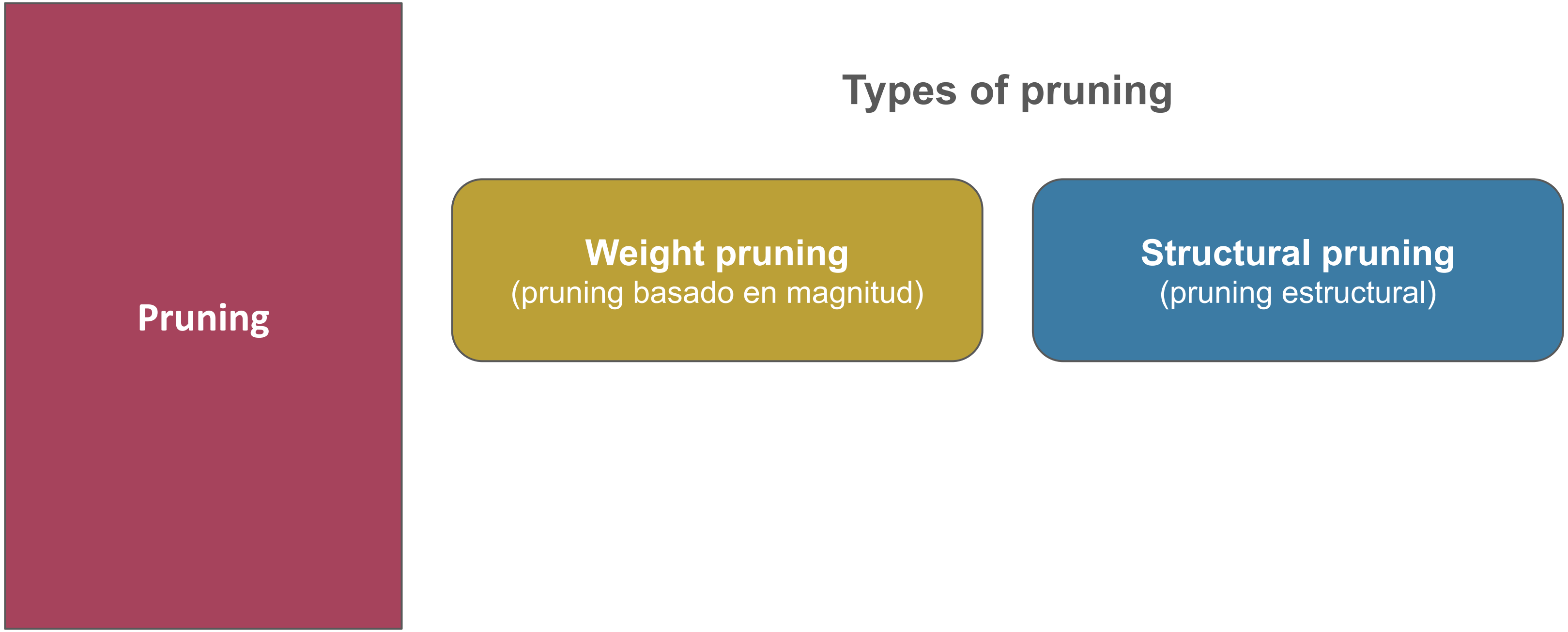
ML and model compression techniques

Pruning

Types of pruning

Weight pruning
(pruning basado en magnitud)

ML and model compression techniques



Pruning

Types of pruning

Weight pruning
(pruning basado en magnitud)

Structural pruning
(pruning estructural)

ML and model compression techniques

Pruning

Types of pruning

Weight pruning
(pruning basado en magnitud)

Structural pruning
(pruning estructural)

Activation-based pruning
(pruning basado en activación)

ML and model compression techniques

Pruning

Types of pruning

Weight pruning
(pruning basado en magnitud)

Structural pruning
(pruning estructural)

Activation-based pruning
(pruning basado en activación)

Iterative pruning vs one-step pruning
(Pruning iterativo vs. pruning único)

ML and model compression techniques

Pruning

Weight pruning

- Removes individual weights within a layer.
- Results in a sparse weight matrix with many zero values.
- How to take advantage of sparsity?

1	0	0	3	0	4
0	3	0	0	0	0
0	0	5	0	7	0
0	0	6	0	0	0
0	0	6	0	0	8

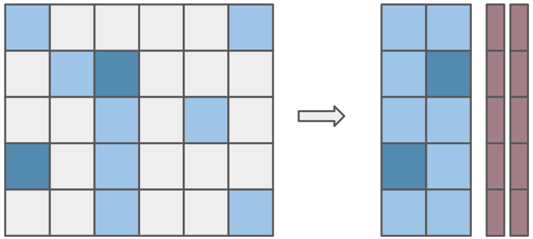
Sparse matrix

ML and model compression techniques

Pruning

Structural pruning

- Removes entire neurons, channels, or filters instead of individual weights.
- Produces a smaller, dense model, improving efficiency.
- Reduces molde size.



Sparse Matrix

Dense Matrix + Index matrix

ML and model compression techniques

for reconfigurable hardware accelerators

Pruning

Advantages

- Model size reduction.
- Acceleration of the inference stage.
- Lower energy consumption.

Drawbacks

- Loss of precision.
- Requires fine-tuning (to recover lost accuracy).
- Hardware compatibility.

Demo:
Pruning for MLP and Fashion MNIST

Quantization

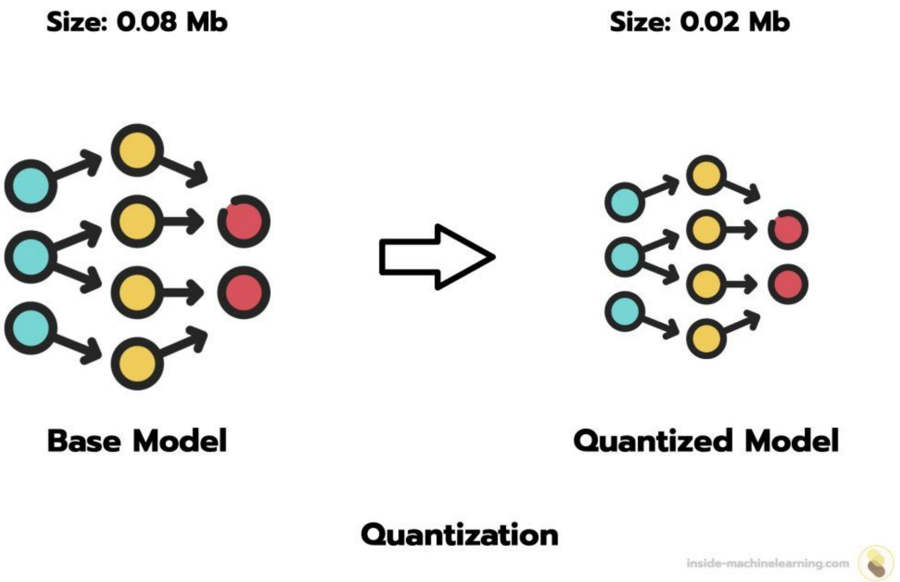
ML and model compression techniques

Quantization

- **Quantization** is a technique that reduces the numerical precision of a neural network's parameters by transforming floating-point values (e.g., 32-bit) into **lower-precision representations**, such as 16-bit or even 8-bit.
- The main goal is to **reduce the model size and speed up inferences**, especially on resource-constrained devices.

ML and model compression techniques

Quantization



ML and model compression techniques

Quantization

- PTQ -

- **Post-Training Quantization (PTQ)** is a quantization technique applied to the trained model.
- It converts the weights and activations from floating-point precision to a lower precision format.
- It is used when the goal is to reduce the model size **without requiring retraining**.

ML and model compression techniques

Quantization

- QAT -

- **Quantization-aware training (QAT)** is a training technique in which the model learns to adapt to quantization before being deployed on hardware. Instead of training the model in full precision (32-bit floating point) and then quantizing it, quantization is introduced during training.

- It provides better results than PTQ but requires more computational effort.

ML and model compression techniques

Quantization
- DQ -

- **Dynamic Quantization (DQ)** is a quantization method that focuses on quantizing only the model's weights, leaving the activations in floating-point format.
- This approach is particularly beneficial for models where **fast inference takes priority over storage efficiency**.

ML and model compression techniques

Quantization

- FIQ -

- **Full Integer Quantization (FIQ)** is a quantization technique where both the weights and activations are fully converted to integer format (INT8).

- This method is commonly used in specialized hardware, such as TPUs, NPUs, or microcontrollers, to maximize efficiency and performance.

ML and model compression techniques

Quantization-aware pruning
- QAP -

- **Quantization-aware pruning (QAP)** combines pruning with quantization-aware training. The goal is to reduce the model size after quantization, resulting in a more efficient network without sacrificing accuracy.

ML and model compression techniques

Quantization

Advantages

- Model size reduction.
- Acceleration of the inference stage.
- Lower energy consumption.

Drawbacks

- Loss of precision.
- Requires fine-tuning (to recover lost accuracy).
- Hardware compatibility.

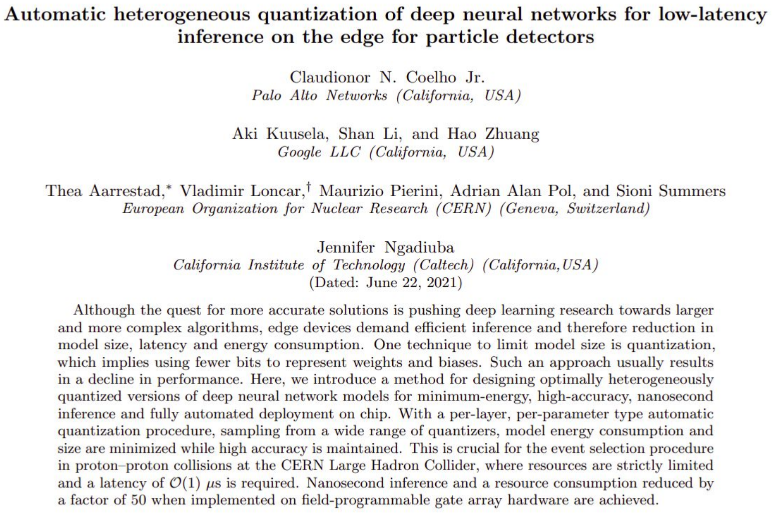
ML and model compression techniques

Quantization

- **QKeras**
  - Extension of Keras designed to quantize neural network models
  - Useful when training models with lower precision.
  - Auto QKeras.
  - https://github.com/google/qkeras

ML and model compression techniques

Quantization



Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors

Claudionor N. Coelho Jr.
Palo Alto Networks (California, USA)

Aki Kuusela, Shan Li, and Hao Zhuang
Google LLC (California, USA)

Thea Aarrestad,[*] Vladimir Loncar,[†] Maurizio Pierini, Adrian Alan Pol, and Sioni Summers
European Organization for Nuclear Research (CERN) (Geneva, Switzerland)

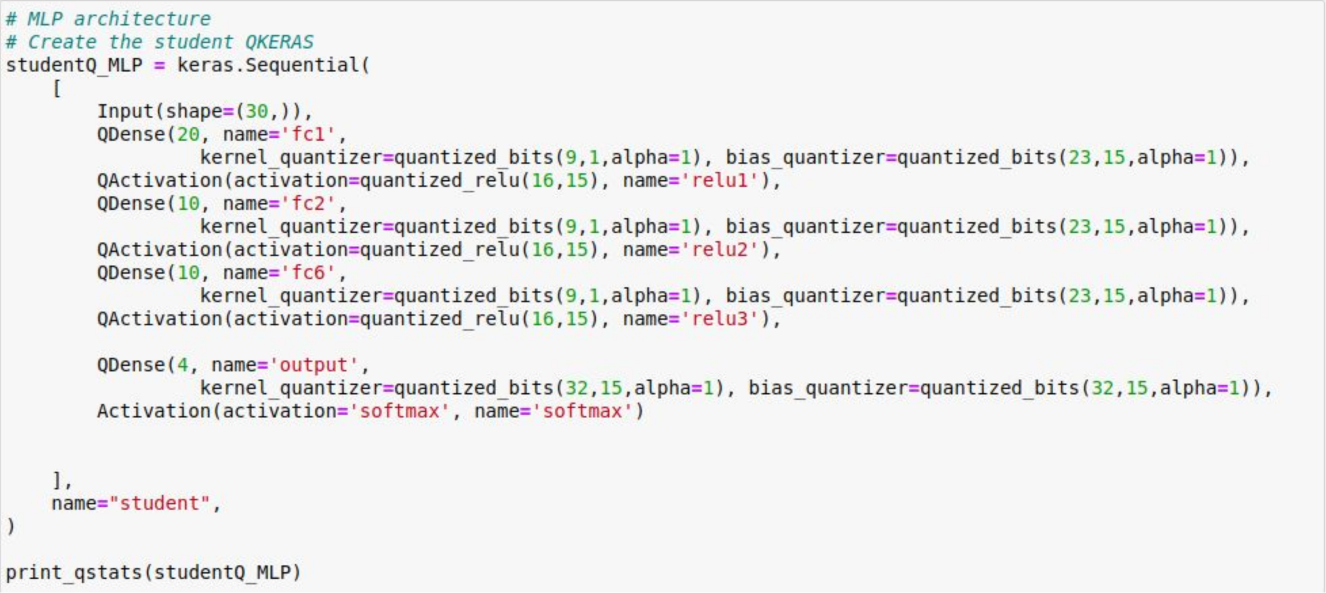
Jennifer Ngadiuba
California Institute of Technology (Caltech) (California,USA)

(Dated: June 22, 2021)

Although the quest for more accurate solutions is pushing deep learning research towards larger and more complex algorithms, edge devices demand efficient inference and therefore reduction in model size, latency and energy consumption. One technique to limit model size is quantization, which implies using fewer bits to represent weights and biases. Such an approach usually results in a decline in performance. Here, we introduce a method for designing optimally heterogeneously quantized versions of deep neural network models for minimum-energy, high-accuracy, nanosecond inference and fully automated deployment on chip. With a per-layer, per-parameter type automatic quantization procedure, sampling from a wide range of quantizers, model energy consumption and size are minimized while high accuracy is maintained. This is crucial for the event selection procedure in proton–proton collisions at the CERN Large Hadron Collider, where resources are strictly limited and a latency of $\mathcal{O}(1)$ μs is required. Nanosecond inference and a resource consumption reduced by a factor of 50 when implemented on field-programmable gate array hardware are achieved.

ML and model compression techniques

Quantization

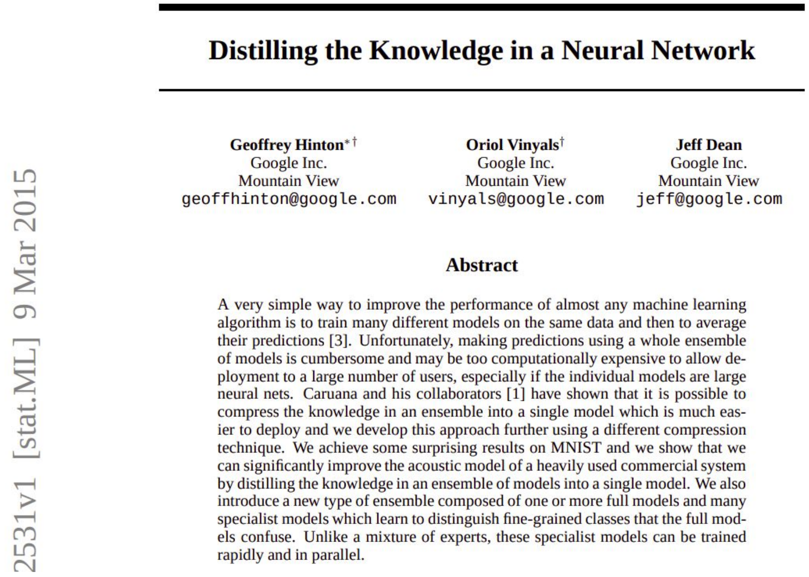


```
# MLP architecture
# Create the student QKERAS
studentQ_MLP = keras.Sequential(
    [
        Input(shape=(30,)),
        QDense(20, name='fc1',
                 kernel_quantizer=quantized_bits(9,1,alpha=1), bias_quantizer=quantized_bits(23,15,alpha=1)),
        QActivation(activation=quantized_relu(16,15), name='relu1'),
        QDense(10, name='fc2',
                 kernel_quantizer=quantized_bits(9,1,alpha=1), bias_quantizer=quantized_bits(23,15,alpha=1)),
        QActivation(activation=quantized_relu(16,15), name='relu2'),
        QDense(10, name='fc6',
                 kernel_quantizer=quantized_bits(9,1,alpha=1), bias_quantizer=quantized_bits(23,15,alpha=1)),
        QActivation(activation=quantized_relu(16,15), name='relu3'),

        QDense(4, name='output',
                 kernel_quantizer=quantized_bits(32,15,alpha=1), bias_quantizer=quantized_bits(32,15,alpha=1)),
        Activation(activation='softmax', name='softmax')


    ],
    name="student",
)

print_qstats(studentQ_MLP)
```

Demo:
Quantization for MLP and Fashion MNIST

ML and model compression techniques

Knowledge distillation



Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†]
Google Inc.
Mountain View
geoffhinton@google.com

Oriol Vinyals[†]
Google Inc.
Mountain View
vinyals@google.com

Jeff Dean
Google Inc.
Mountain View
jeff@google.com

Abstract

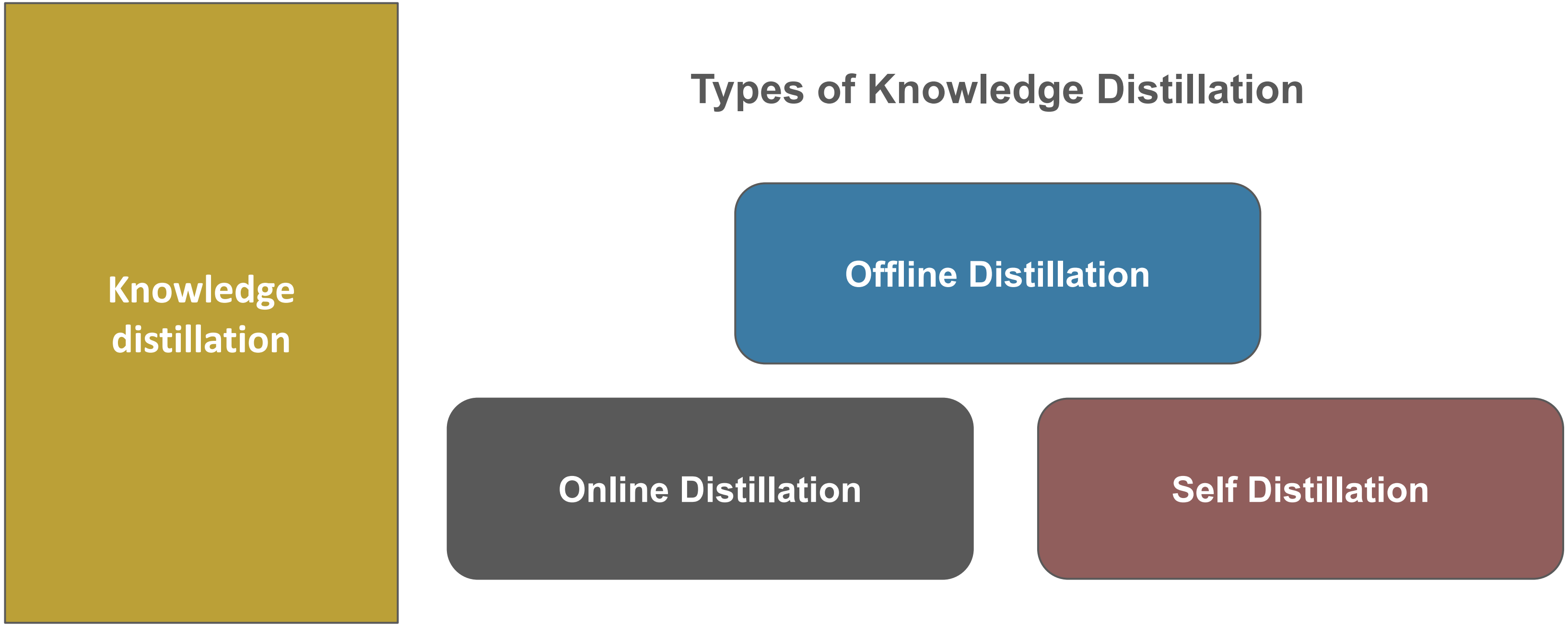
A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions [3]. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators [1] have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

ML and model compression techniques

Knowledge distillation

- **Knowledge Distillation (KD)** is devoted to transferring knowledge from a teacher network to a smaller and faster target network (named a distilled or student network) that can reproduce the teacher's behavior while being computationally less expensive.

ML and model compression techniques



Knowledge distillation

Types of Knowledge Distillation

Offline Distillation

Online Distillation

Self Distillation

ML and model compression techniques

Knowledge distillation

Offline Distillation

- Use of a pre-trained teacher model to guide the student model
- Knowledge
  - The output probabilities not only indicate the correct class but also contain the relative probabilities of incorrect classes.

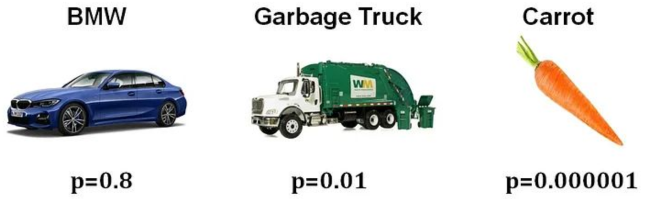


Image from https://medium.com/@aryamaanthakur/knowledge-distillation-make-your-neural-networks-smaller-398485f811c6

ML and model compression techniques

Knowledge distillation

General Steps

- Train the teacher model
  - soft labels (probability distributions) for each sample.

ML and model compression techniques

Knowledge distillation

General Steps

- Train the teacher model
  - soft labels (probability distributions) for each sample.
- Generate Soft Labels (Logits) from the Teacher.
  - softmax with temperature T (a higher T makes the probabilities smoother)

ML and model compression techniques

Knowledge distillation

General Steps

- Train the teacher model
  - soft labels (probability distributions) for each sample.
- Generate Soft Labels (Logits) from the Teacher.
  - softmax with temperature T (a higher T makes the probabilities smoother)

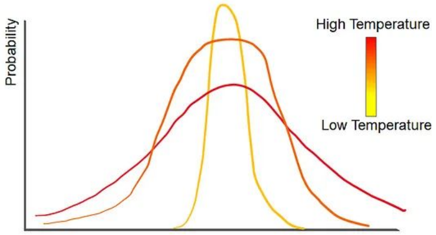


Image from https://medium.com/@aryamaanthakur/knowledge-distillation-make-your-neural-networks-smaller-398485f811c6

ML and model compression techniques

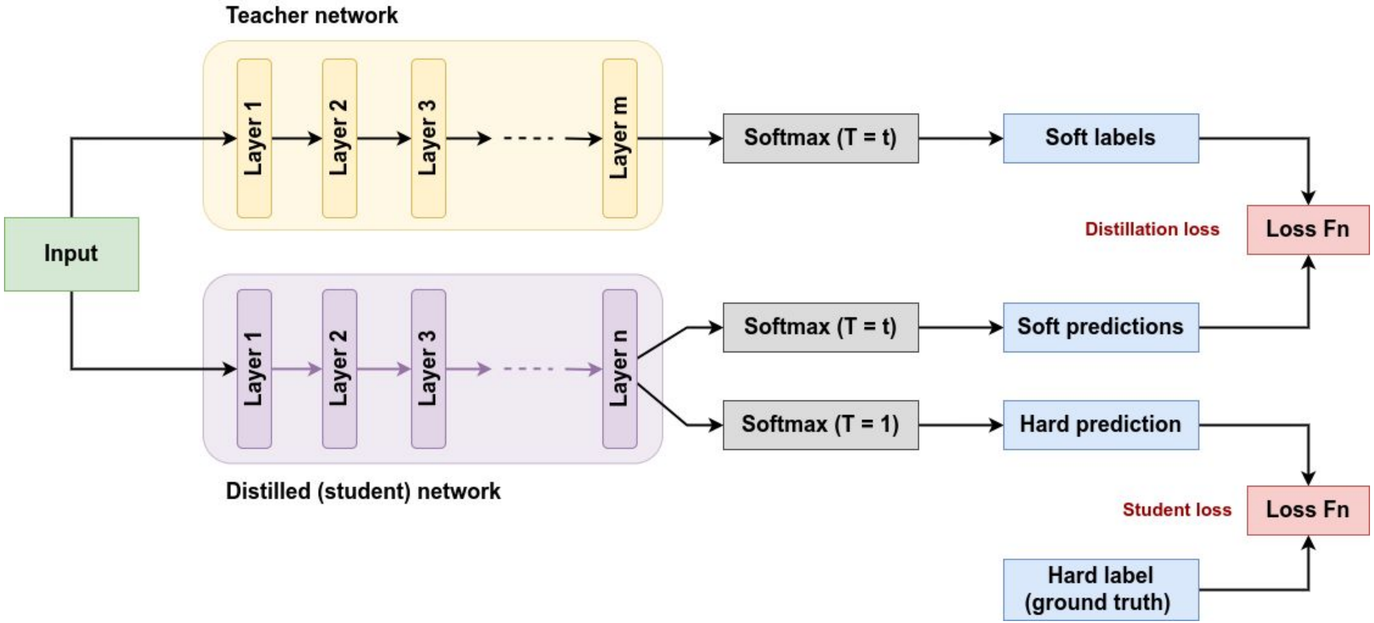
Knowledge distillation

General Steps

- Train the teacher model
  - soft labels (probability distributions) for each sample.
- Generate Soft Labels (Logits) from the Teacher.
  - softmax with temperature T (a higher T makes the probabilities smoother)
- Train the student with two losses:
  - Distillation loss: Matches Student's **soft labels** to the Teacher’s **soft labels**.
  - Classification loss: Matches Student’s **predictions** to ground **truth labels**.

ML and model compression techniques

Knowledge distillation



ML and model compression techniques

Knowledge distillation

Online Distillation

- The teacher and student models are updated simultaneously in a single training process.
- When the teacher is not available.

ML and model compression techniques

Knowledge distillation

Online Distillation

- The teacher and student models are updated simultaneously in a single training process.
- When the teacher is not available.

Self Distillation

- Similar to Online Distillation.
- The technique uses the same model as the teacher as well as the student.

ML and model compression techniques

Knowledge distillation

Advantages

- Reduction of model size.
- Acceleration of the inference stage.
- Improves computational efficiency.
- Leverages the knowledge of the master model.

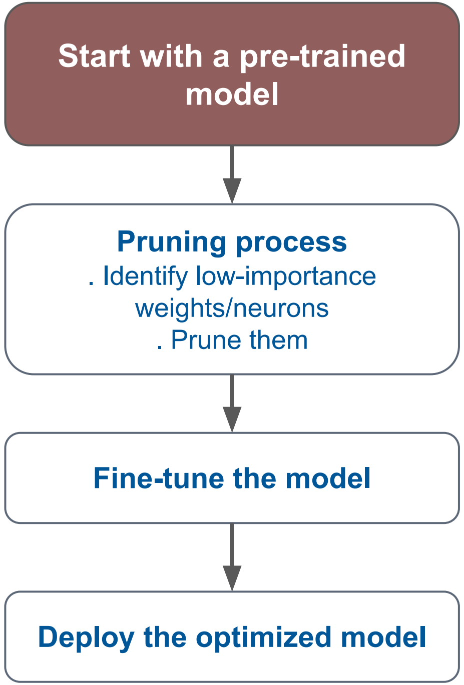
Drawbacks

- Complexity of proper implementation.
- Not always effective; it depends on the generalization ability of the master model.
- Expensive training.

How do we combine compression techniques?

How do we combine compression techniques?

Pruning



Start with a pre-trained model

Pruning process

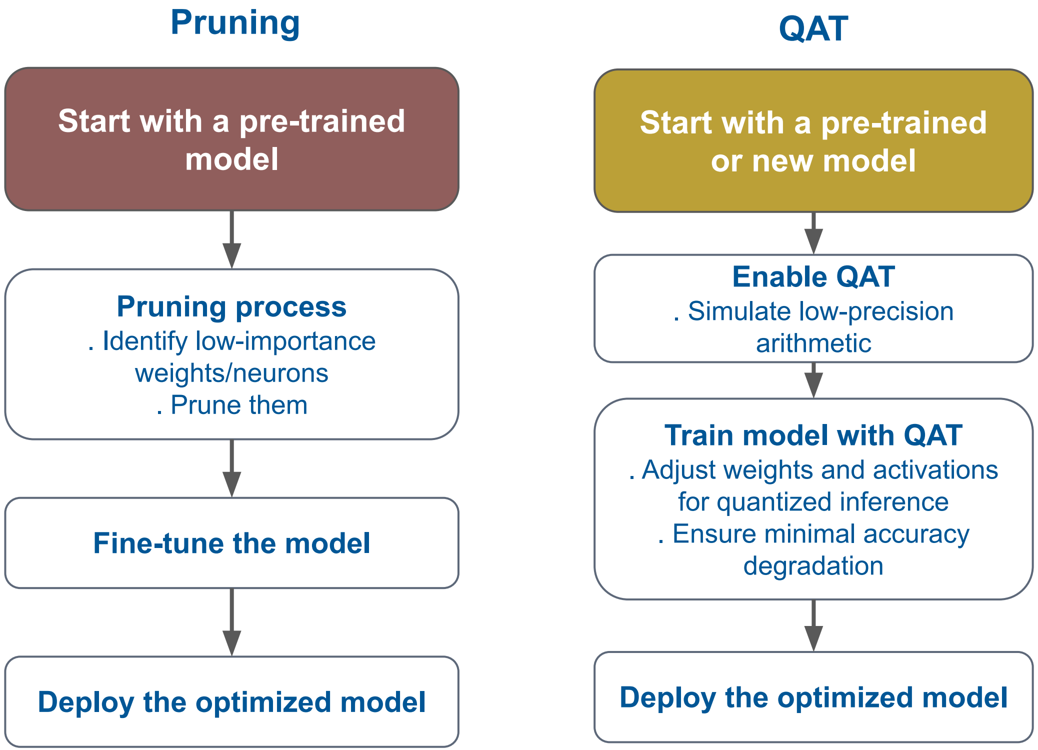
. Identify low-importance weights/neurons

. Prune them

Fine-tune the model

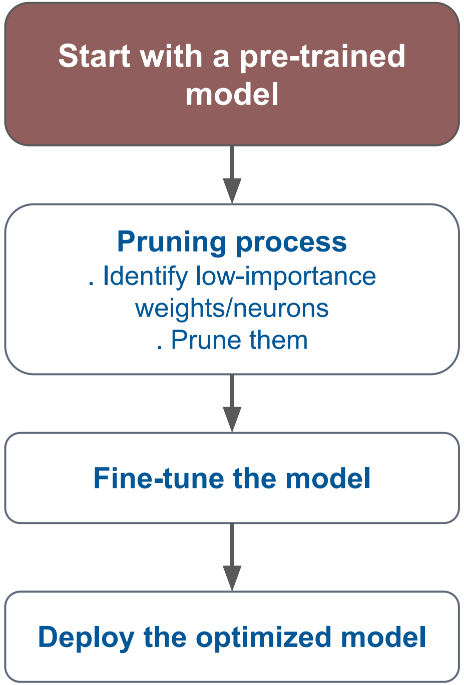
Deploy the optimized model

How do we combine compression techniques?


Pruning
Start with a pre-trained model
Pruning process
. Identify low-importance weights/neurons
. Prune them
Fine-tune the model
Deploy the optimized model
QAT
Start with a pre-trained or new model
Enable QAT
. Simulate low-precision arithmetic
Train model with QAT
. Adjust weights and activations for quantized inference
. Ensure minimal accuracy degradation
Deploy the optimized model

How do we combine compression techniques?

Pruning



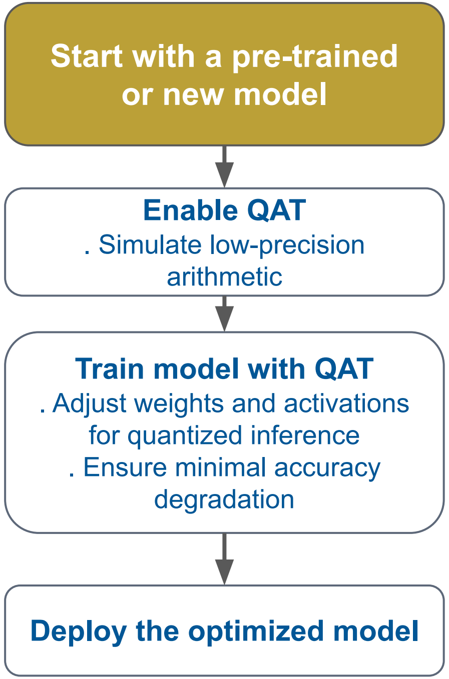
Start with a pre-trained model

Pruning process
. Identify low-importance weights/neurons
. Prune them

Fine-tune the model

Deploy the optimized model

QAT



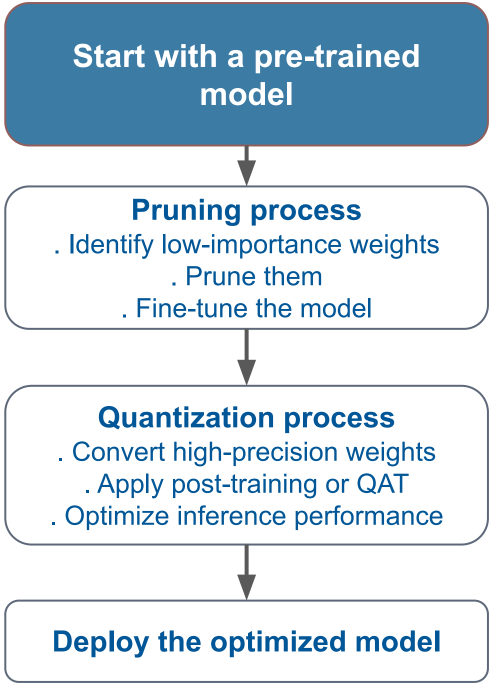
Start with a pre-trained or new model

Enable QAT
. Simulate low-precision arithmetic

Train model with QAT
. Adjust weights and activations for quantized inference
. Ensure minimal accuracy degradation

Deploy the optimized model

P + Q



Start with a pre-trained model

Pruning process
. Identify low-importance weights
. Prune them
. Fine-tune the model

Quantization process
. Convert high-precision weights
. Apply post-training or QAT
. Optimize inference performance

Deploy the optimized model

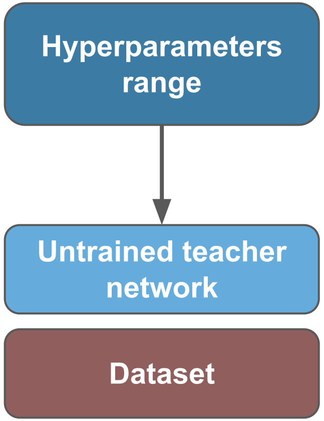
ML and model compression techniques

- *When to choose pruning, quantization, or KD?*

Hyperparameters tuning and compression

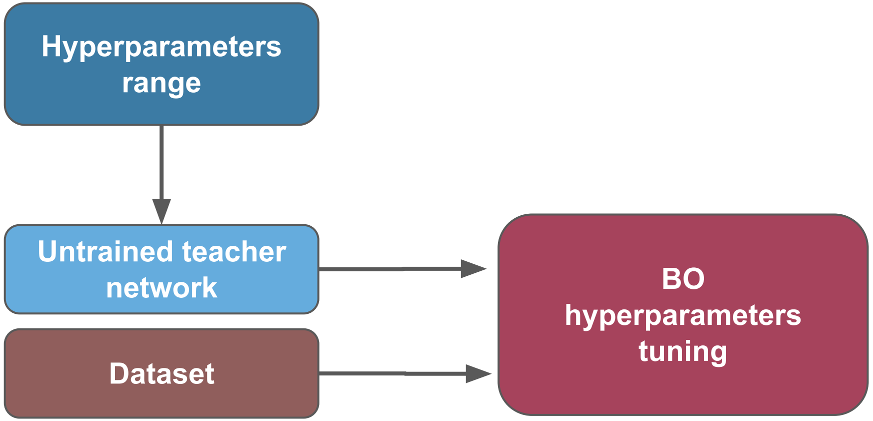
DNN training and compression

Stage 1 - Teacher training



DNN training and compression

Stage 1 - Teacher training



Hyperparameters range

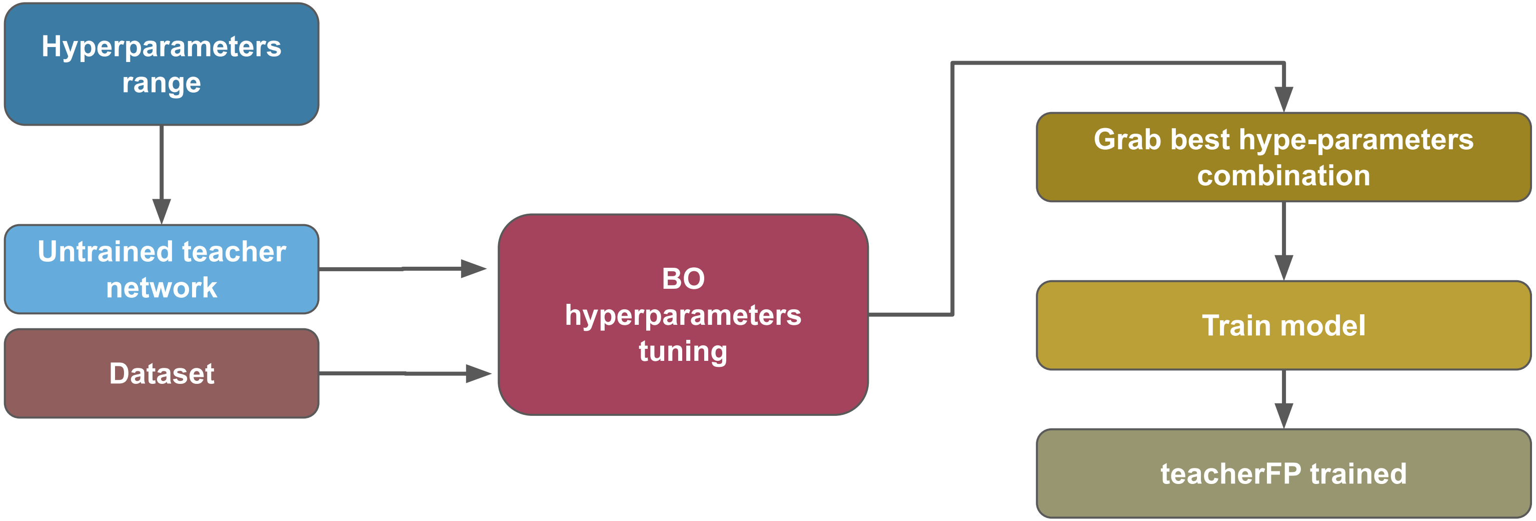
Untrained teacher network

Dataset

BO hyperparameters tuning

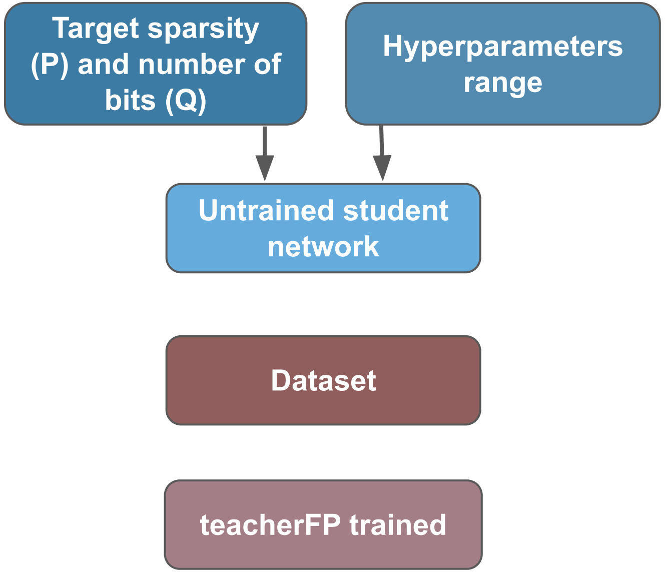
DNN training and compression

Stage 1 - Teacher training


Hyperparameters range
Untrained teacher network
Dataset
BO hyperparameters tuning
Grab best hype-parameters combination
Train model
teacherFP trained

DNN training and compression

Stage 2 - Student training



Target sparsity (P) and number of bits (Q)

Hyperparameters range

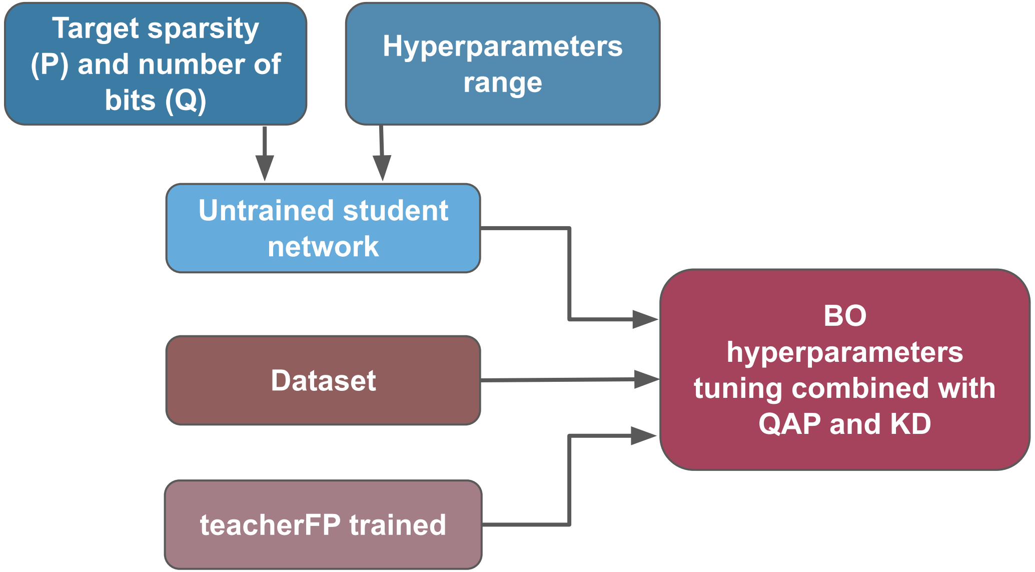
Untrained student network

Dataset

teacherFP trained

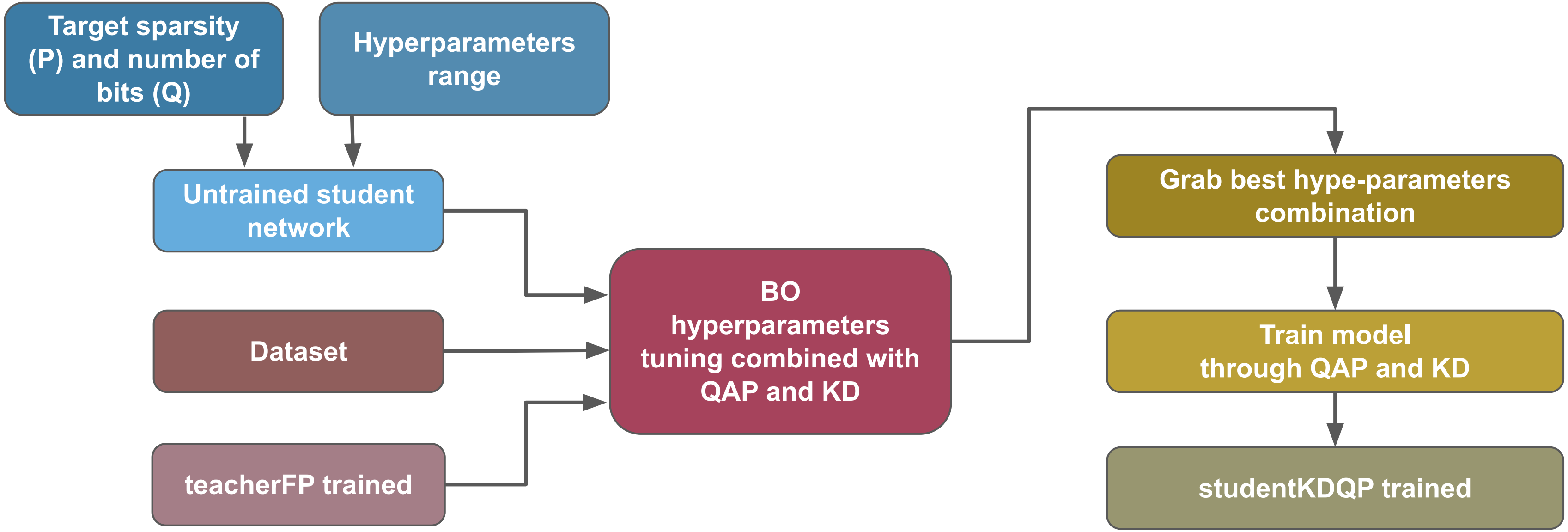
DNN training and compression

Stage 2 - Student training


Target sparsity (P) and number of bits (Q)
Hyperparameters range
Untrained student network
Dataset
teacherFP trained
BO hyperparameters tuning combined with QAP and KD

DNN training and compression

Stage 2 - Student training



Target sparsity (P) and number of bits (Q)

Hyperparameters range

Untrained student network

Dataset

teacherFP trained

BO hyperparameters tuning combined with QAP and KD

Grab best hype-parameters combination

Train model through QAP and KD

studentKDQP trained

Demo:
MNIST-based binary classification
- QAP -

Thank you!

Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Mar del Plata, Argentina - 2025 -

Universidad Nacional de Mar del Plata

Fundación Williams

ICTP The Abdus Salam International Centre for Theoretical Physics