Capstone Bi-Monthly Update

Date: 09/27/2023

Team: Aditya, Elisa, Jenny, Zhanyi

Duke

Agenda

1) Updates
2) Parameter-Efficient Fine-Tuning
    a) LoRA
    b) Q-LoRA
3) Compression Techniques
    a) Quantization
    b) Distillation
    c) OpenVINO
    d) ONNX

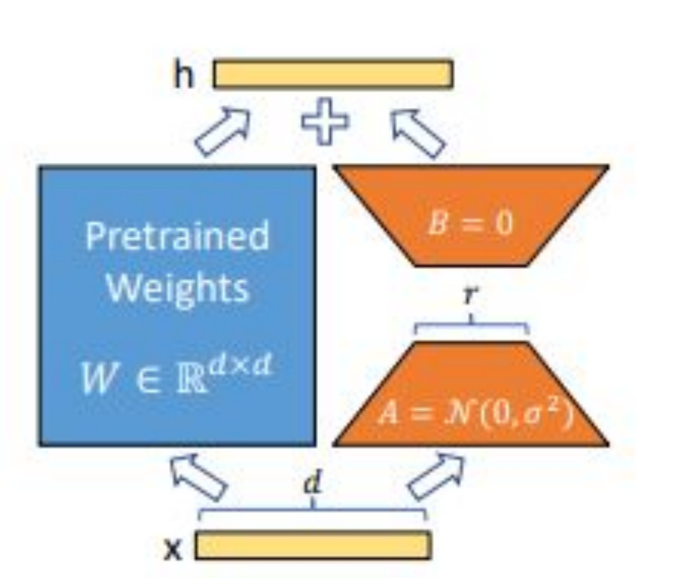
Duke

Updates

- Created a GitHub Repository for our work
- Spun up a small BERT model on in our workspace
- Future bi-monthly updates will be uploaded in our GitHub

LoRA - A compressed adapter based fine-tuning method

- Overview:
  - Key Idea: Common pre-trained models have a very low intrinsic dimension.
  - Freeze original model but update A and B.
  - Performance boosts are because r << d
- Orthogonal to other PEFT techniques
- Requires storing only r * d values during back-propagation.
- Available on hugging face



$$h = W_0 x + \Delta W x = W_0 x + BAx$$

Duke

Q-LoRA - LoRA + Quantization

- 3 Key optimizations that QLoRA brings on top of LoRA:
  - 4-bit NormalFloat (NF4)
  - Double Quantization
  - Paged Optimisation

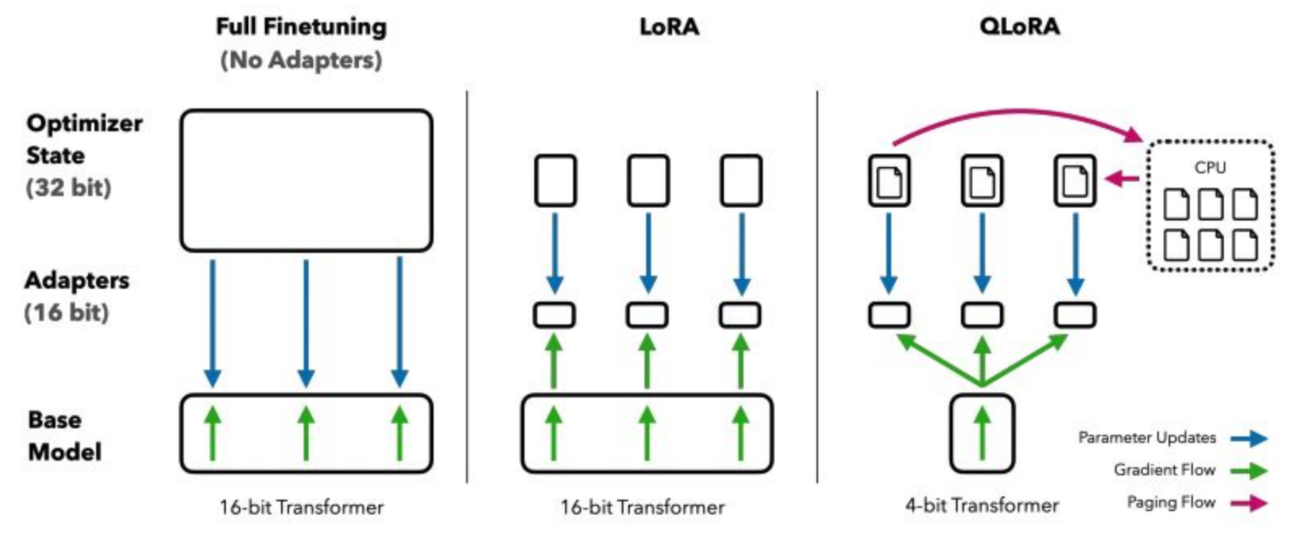


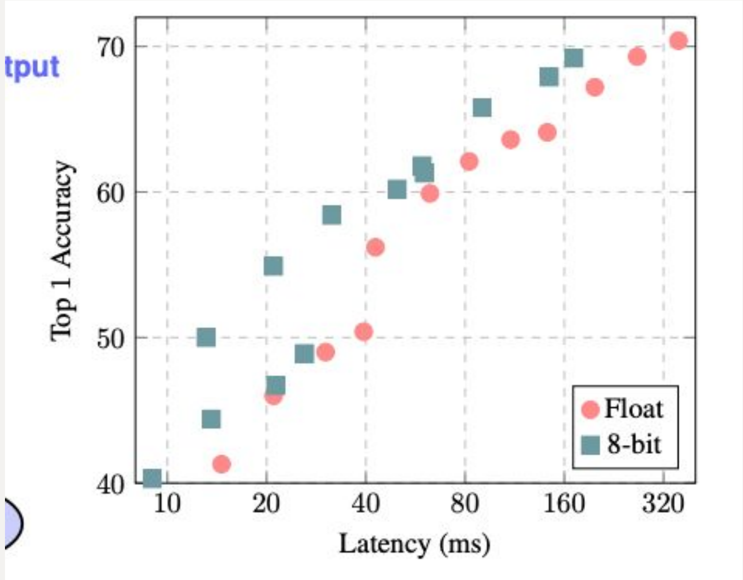
Figure 1: Different finetuning methods and their memory requirements. QLoRA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizers to handle memory spikes.

Duke

Sources: Tim et at. in 2023
GitHub Page

Quantization

- Technique used to reduce the computational and memory costs of running inference by representing the weights and activations of a neural net with low-precision data types
  - 32-bit (float32) → float16 / int8 (most common conversions)
- Difficult if weights must be represented with really big or small values



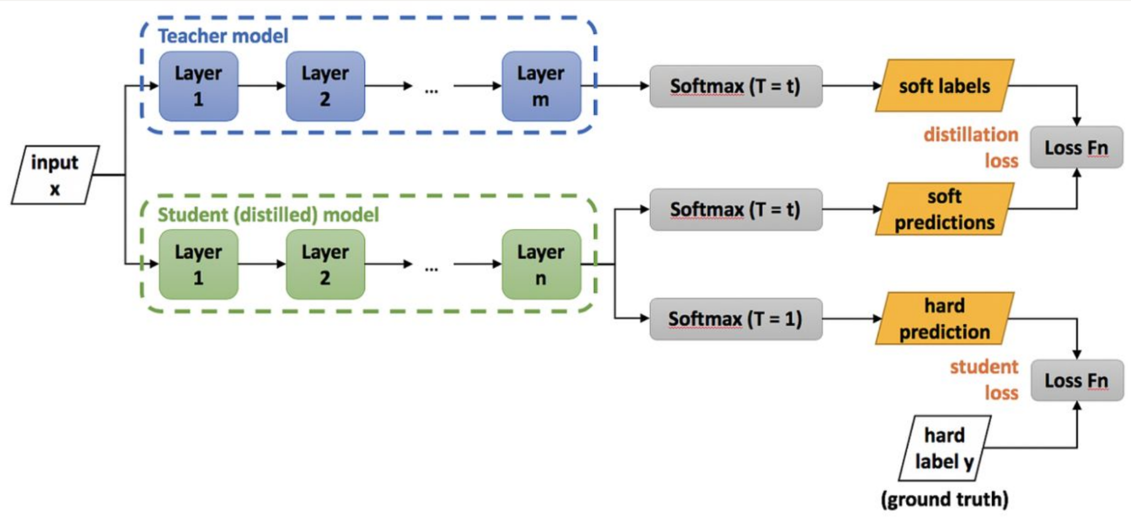
Duke

Distillation

- Originates from the masterpiece by [Hinton et al. in 2015]
- Process:
  1. Training the Teacher Model
  2. Distilling Knowledge
     - Temperature for softmax

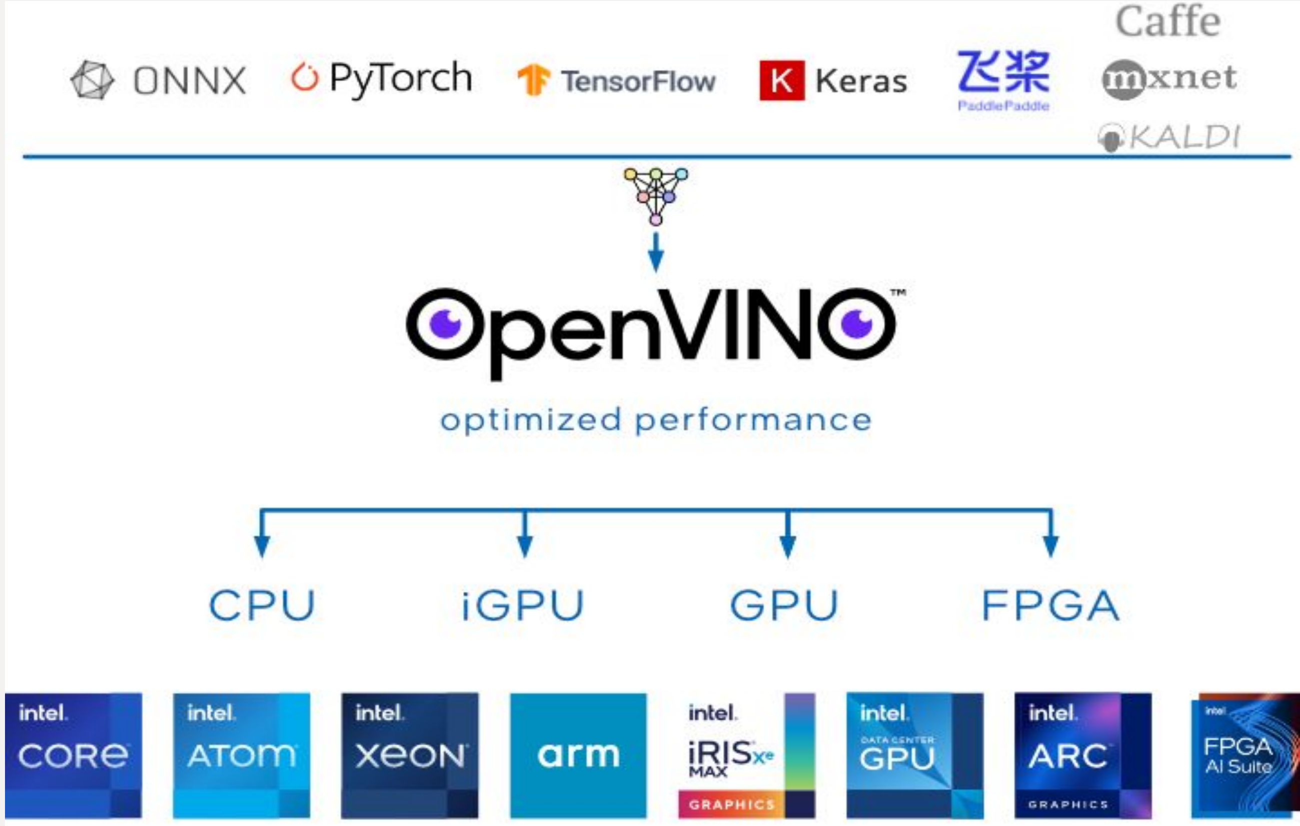
$$y_i' = \frac{exp(y_i)}{\sum_j exp(y_j)} \xrightarrow[T\,=\,100]{} y_i' = \frac{exp(y_i/T)}{\sum_j exp(y_j/T)}$$

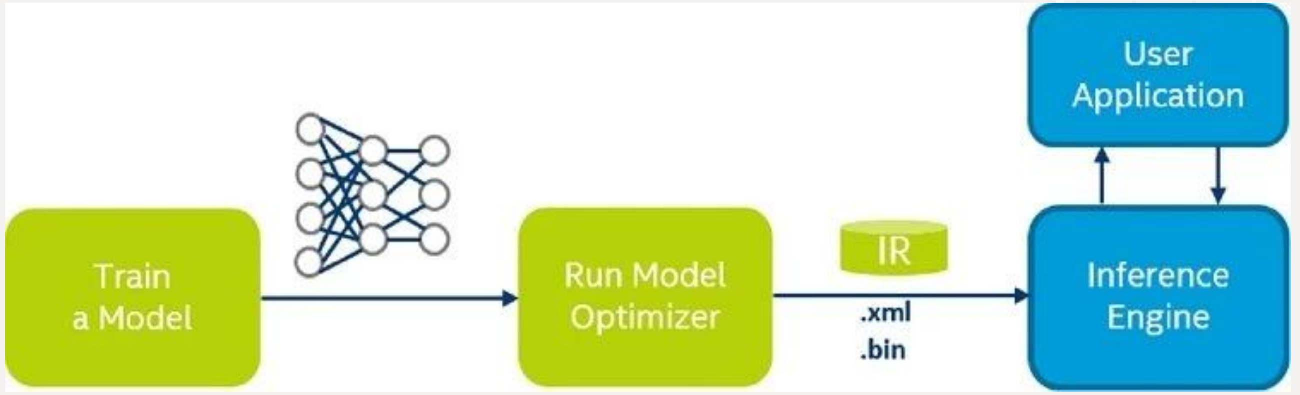
  3. Training the Student Model
     - the student network is trained to minimize the difference between its own predictions and the soft labels generated by the teacher network.



OpenVINO

- OpenVINO (Open Visual Inference and Neural Network Optimization) is a cross-platform deep learning toolkit developed by Intel.
- OpenVINO optimizes and accelerates neural network inference for Intel hardware by converting models to an intermediate representation and deploying through a high-performance inference engine.
- Supports models trained in popular frameworks like TensorFlow and PyTorch.

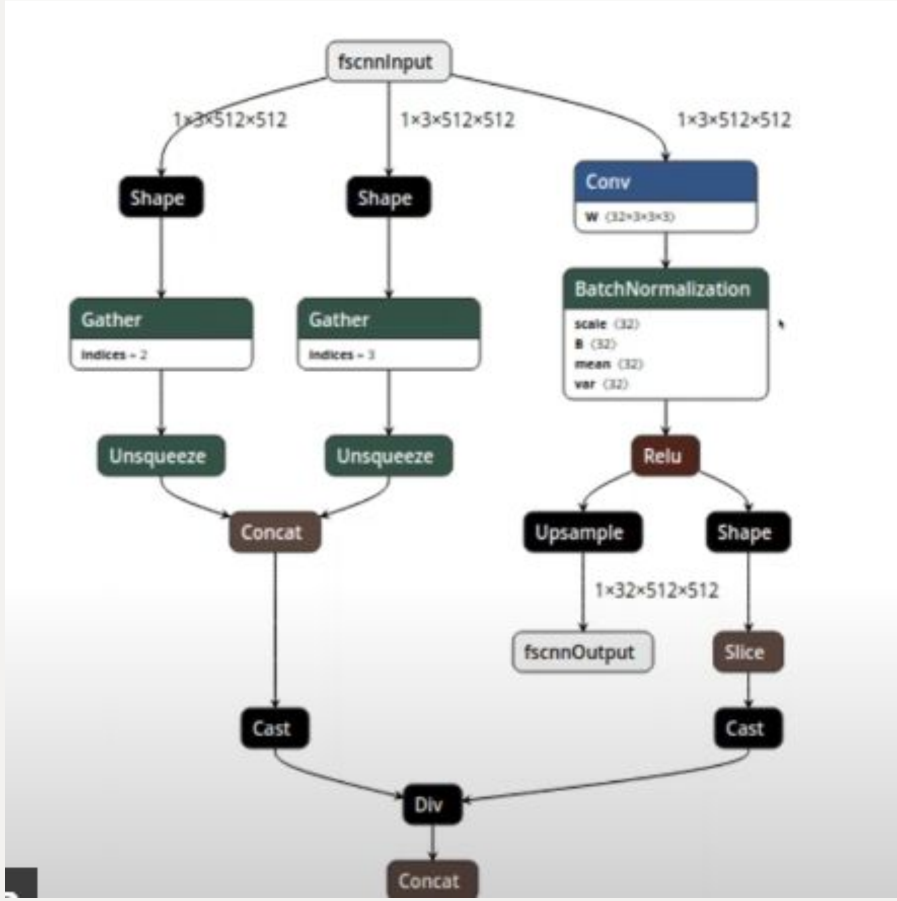




Duke

ONNX

- Intermediary to convert between different machine learning frameworks and hardware acceleration
- Models represented as DAGs
- Most optimizations performed under the hood
- Available on hugging face (Optimum library)



Duke

Potential Next Steps

- Begin testing different different PEFT methods via hugging face (deprioritize for next semester)
- Begin testing various neural network acceleration techniques
  - start off with BERT and another small LLM (baseline)
    - Distillation, quantization,
    - performance (latency, and some other data science performance metric of a dataset of our choice for baseline, **1) distilled version, 2) distilled + quantized**, 3) distilled + quantized + network optimization)
    - CPU and GPU comparison (OpenVINO only CPU) - final product should be compatible with both CPU and GPU (with CPU prioritization)
- Conduct additional compression literature review
- Begin trying to apply toolkits we have explored
- Continue looking for an appropriate dataset

Duke

Appendix: Common Quantization Schemes

- Affine Quantization Scheme
  - Project a range [a, b] of float32 values to the int8 space.
  - `x_q = clip(round(x/S + Z), round(a/S + Z), round(b/S + Z))`
- Scale Quantization
  - Like Affine quantization, but where zero point (Z) is set to 0 and does not play a role in the equations.
  - Symmetric quantization scheme: Project a range [-a, a] of float values into an integer space
- Calibration is the step during quantization where the float32 ranges [a,b] or [-a, a] are calculated.

Duke