Intel® Edge AI for IoT Developers Nanodegree Program

Glossary – Course 3 – Optimization Techniques and Tools

Lesson 1: Introduction to Software Optimization

Edge system

Your complete edge AI project, including the hardware (like the neural compute stick), the software, and the power supply.

Floating Point Operation (FLOP)

Any operation on a float value. Multiple floating point operations can be abbreviated as **FLOPs** (with a lowercase "s"). Note that this refers to a *quantity* (contrast with FLOPS, below, which is a *rate*).

Floating Point Operations per Second (FLOPS)

Hardware devices are generally rated for the number of floating point operations they can perform in a second. Note that **FLOPS** (with a capital "S") is a *rate*.

Hardware optimization

Can be as simple as changing to a new hardware platform or as complicated as building a custom hardware specifically designed to increase the performance of a particular application.

Latency

The time taken from when an image is available for inference until the result for that image is produced.

Metric

A quantity or attribute of a system that can be measured. Should help us infer useful information about a system.

Multiply-Accumulate Operation (MAC)

Computations in neural networks typically involve a *multiplication* and then an *addition* and are thus referred to as Multiply-Accumulate Operations (MACs).

Performance

A measure of how efficiently a system is performing its task. Typically measured in terms of the accuracy, inference speed, or energy efficiency of the system.

Reducing model size

This will reduce the time it takes to load the model and perform inference by removing unimportant or redundant parameters from the network.

Reducing number of operations

This will reduce the time taken to perform inference by reducing the number of operations or calculations needed to execute the network. This can be done by using more efficient layers and by removing connections in between neurons in our model.

Software optimization

Involves making changes to your code or model to improve your application's performance. As applied to Edge computing, this will involve techniques and algorithms that reduce the computational complexity of the models.

Throughput

The amount of data that is processed in a given interval of time.

Lesson 2: Reducing Model Operations

Pooling layers

Subsampling layers that reduce the amount of data or parameters being passed from one layer to another

Pruning

A model compression technique where redundant network parameters are removed while trying to preserve the original accuracy (or other metric) of the network

Separable convolution

A separable convolutional layer divides the standard convolutional layer into two layers: a *depthwise convolution* and a *pointwise convolution*. By doing so, the number of FLOPs or operations required to run the model is reduced.

Lesson 3: Reducing Model Size

Quantization

The process of mapping values from a larger set to a smaller one. In quantization, we might start off with a continuous (and perhaps infinite) number of possible values, and map these to a smaller (finite) set of values. In other words, quantization is the process of reducing large numbers of continuous values down into a limited number of discrete quantities.

Knowledge distillation

A method where we try to transfer the knowledge learned by a large, accurate model (the teacher model) to a smaller and computationally less expensive model (the student model).

Model compression

Refers to a group of algorithms that help us reduce the amount of memory needed to store our model, and also make our models more compact (in terms of the number of parameters).

Weight quantization

Quantization in which we reduce the amount of space required for the weight values—such as by reducing our weight values from 32 bits to 8 bits.

Weight sharing

In weight sharing, the goal is for multiple weights to share the same value. This reduces the number of unique weights that need to be stored, thus saving memory and reducing model size.

Lesson 4: Other Optimization Tools and Techniques

Deployment manager

The deployment manager is a tool that can be used to package your model files, application code, and OpenVINO installation so that it can be easily deployed to other devices.

Hotspots

These are areas of your code that are inefficient or require a lot of time to execute.

VTune Amplifier

An advanced profiler that can be used to find hotspots in our application code (unlike the DL Workbench which is used to find the performance of your model).