Application-Specific Integrated Circuits and Machine Learning

Machine Learning (ML) algorithms build a mathematical model based upon representative sample data, known as 'training data', in order to make predictions or decisions without being explicitly programmed to perform the task. I limit my discussion here to *supervised learning* in the context of a potential application to seismic data image processing of a real marine seismic dataset, and then discuss how the computational scale of such exercises reinforce the need to develop computing technology that is customized for large ML problems. I then briefly describe the emergence and relevance of Application-Specific Integrated Circuit (ASIC) concepts to accelerate ML applications.

Machine Learning, Deep Learning, Supervised Learning and Convolutional Neural Networks

Supervised learning is the ML task of learning a function that maps an input to a desired output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples, each having one or more inputs and a desired output. Each training example is represented by an *array* or *vector*, sometimes called a feature vector, and the training data is represented by a *matrix*. Through iterative optimization of an objective function, the algorithms learn a function that can then be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a discrete set of values (e.g. photo recognition), and regression algorithms are used when the outputs may have a continuous range of values (e.g. the seismic processing example used below).

Deep learning (DL) is part of a broader family of ML methods based upon Artificial Neural Networks (ANNs), and uses multiple layers to progressively extract higher-level features from the raw input. A Deep Neural Network (DNN) finds the correct mathematical manipulation to turn the input into the desired output, whether it be via a linear or non-linear relationship. The network moves through the layers calculating the probability of each desired output. As discussed below, Graphics Processing Units (GPUs) are well-suited for the matrix/vector computations involved in ML, speeding up training algorithms by orders of magnitude in comparison to Computer Processing Units (CPUs), and have led the resurgence of interest in DL solutions over the past decade.

The DL example shown below used a Convolutional Neural Network (CNN). CNNs consist of an input and output layer, as well as several hidden layers; include regularized versions of fully connected neural networks in which each neuron in one layer is connected to all neurons in the next layer; and use convolution in place of general matrix multiplication in at least one of their layers.

Machine Learning Example of Towed Streamer Acquisition Footprint Attenuation

Figures 1 and 2 show a brief ML experiment to mitigate migration artifacts associated with acquisition footprint effects (courtesy of Elena Klochikhina, PGS). A DNN was trained to attenuate coherent noise on seismic images. The noise observed in the upper (cross-line) panel of Figure 2 is a result of uneven subsurface illumination due to limited data coverage and/or propagation through complex media. Most of the time it is easy to visually distinguish the artifacts in the images. Unfortunately, it is hard to describe them formally and therefore remove the noise without damaging the image resolution. Deep learning has proven useful in computer vision for similar applications. Instead of designing a filter formally, a CNN architecture was used to learn what filters need to be applied to remove the coherent noise. Synthetic data examples were created for the training (Figure 1), and the performance of the neural network was tested on a real field dataset (Figure 2). The application to the field dataset is encouraging, and it shows the potential of the DNN to successfully attenuate migration swings without compromising image resolution.

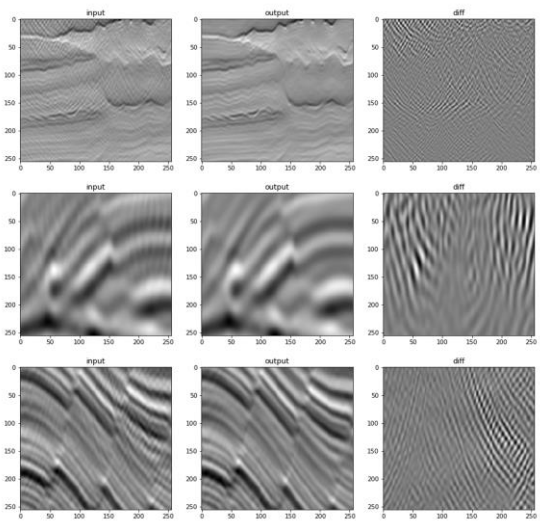

Figure 1. Example 256x256 pixel windows of synthetic seismic data used to train the CNN. The panels are arranged in sets of three with the left panel being the input to the DNN, the middle panel being the desired output from the DNN, and the right panel is the difference between the desired output and the input (noise).

This simple example, run on a GPU cluster, illustrates the significance of using large training datasets to improve ML accuracy. Furthermore, the accuracy achieved by ML solutions follows a curve of diminishing return vs. the number of computations, so the development of dedicated computer hardware is been promoted by some companies as necessary to accommodate increasingly vast training datasets whilst continuously seeking higher levels of accuracy.

Graphics Processing Units, Application Specific Integrated Circuits, and Tensor Processing Units

Graphics Processing Units (GPUs) traditionally use most of their transistors to perform calculations related to 3D computer graphics. Because most of these computations involve matrix and vector operations, research into their applications to ML has proliferated: GPUs can be 250 times faster than CPUs (Computer Processing Units) for training deep learning neural networks. The growth of deep learning applications has also encouraged development of Application-Specific Integrated Circuits (ASICs); for example, the Tensor Processing Unit (TPU) made by Google. TPUs are custom-made accelerators designed to accelerate ML workloads using TensorFlow, a symbolic math library used for machine learning applications such as neural networks (and one of several open-source libraries). Compared to a GPU, a TPU is designed for a high volume of low precision computation (as little as 8-bit) with more input/output operations per joule, and lacking hardware for rasterization and texture mapping that a GPU contains.

The 1st generation TPU (TPU v1) was announced in 2016 after more than a year of use, offered an 8-bit matrix multiplication engine limited to integer calculations, and was therefore limited only to inference (the way that NNs infer things about new data based upon its training). The 2nd generation TPU (TPU v2) was announced in 2017 with floating point capability, therefore applicable to both NN training and inference, and was introduced to the Google Cloud for use in TensorFlow applications. As discussed below, the 3rd generation TPU (TPU v3) was announced in 2018 with 8x the capacity of TPU v2. The ‘Edge TPU’ was also announced in 2018 for applications to edge computing, and is limited to 8-bit calculations.

Tensor Processing Unit Architecture

Each Cloud TPU includes a host with each accelerator connected via PCI (Peripheral Component Interconnect: a bus for attaching hardware devices). In the 2nd generation of TPUs, each Cloud TPU device has four chips below the heat sinks on the board. Cloud TPU v2 chips contain two cores that each contain both a vector unit and a scalar unit in the manner of a GPU, has 8 GB of high-bandwidth memory per core (16 GB in the 3rd generation, and four cores per board), and notably, includes a systolic matrix unit (MXU). The MXU comprises a 128x128 systolic array designed to rapidly perform large matrix multiplications (up to two 128x128 matrices in one pass) using a new, cost-effective floating-point format called ‘bfloat16’ (accumulations performed in 32 bit floating point).

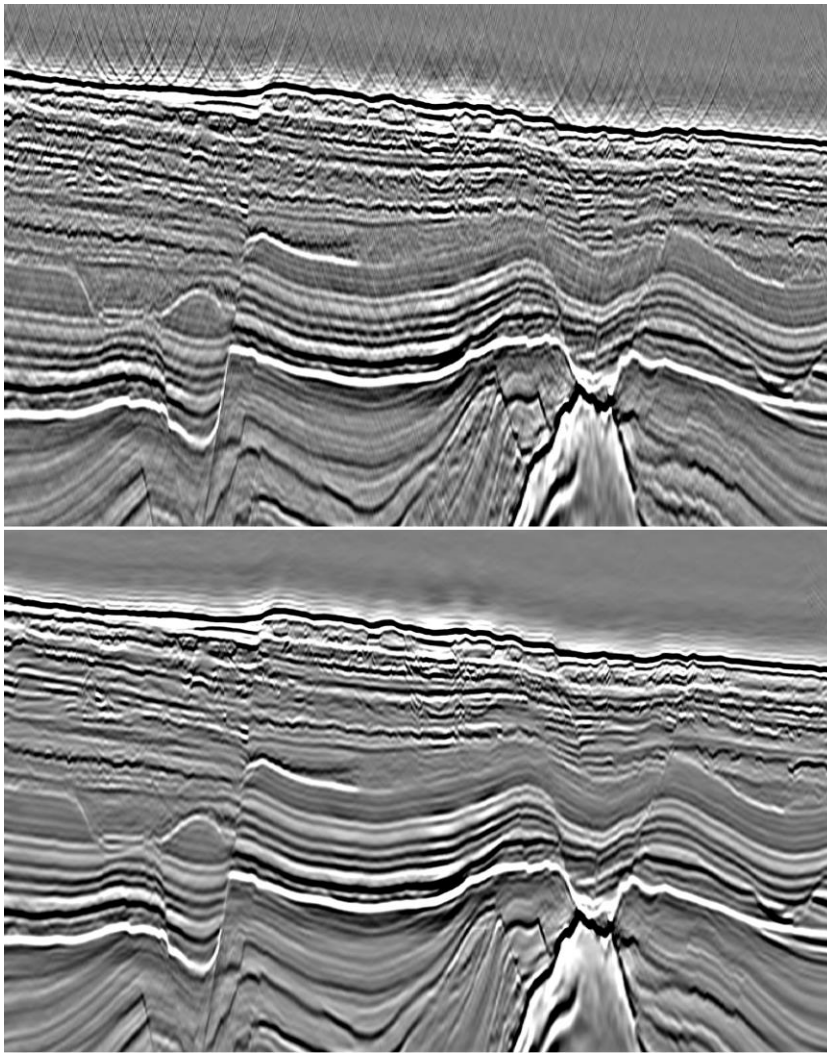

Figure 2. (upper) real migrated cross-line stack before footprint attenuation (no regularization applied); and (lower) result to demonstrate a CNN-based footprint attenuation. The coherent noise from shots with very large cross-line separation has been reduced after prediction without significant damage to the image resolution. Courtesy of Elena Klochikhina (PGS).

This new data format was motivated by the fact that the data range is more important than the data accuracy when training ML networks. NNs are quite resistant to the loss of precision, and the noise introduced by precision loss may sometimes act as a regularization operator and assist convergence.

Figure 3 schematically compares different floating-point formats. In decimal notation, very large numbers are shown with a mantissa and an exponent. e.g., $0.15*10^3$, where the 0.15 is the mantissa and the 10^3 is the exponent. The mantissa holds the main digits, and the exponent defines where the decimal point is placed. The same notation is used for binary numbers. In contrast to traditional half-precision (16 bit) IEEE floating point format, the 'brain' floating point format (bfloat16 in Figure 3) assigns extra bits for the exponent to preserve the same data range as single-precision IEEE floating point format (float32 in Figure 3); thereby being able to serve as a 'drop in' replacement for float32 but with lower computational cost. Each TPU core subsequently contains over 16 000 multiply-accumulators (128^2). The rule of thumb was that one TPU v2 with four chips was roughly as fast as five GPUs (also containing four dual-core chips), and almost three times cheaper.

The 2nd generation Cloud TPU 'Pod' setup was introduced in 2017 with up to 11.6 PFLOP computing capacity (512 TPU v2 cores), 4 TB of high-bandwidth memory, provided both training and inference, and was upgraded to the 3rd generation in 2018 to have more than 100 PFLOP capacity (2048 TPU v3 cores).

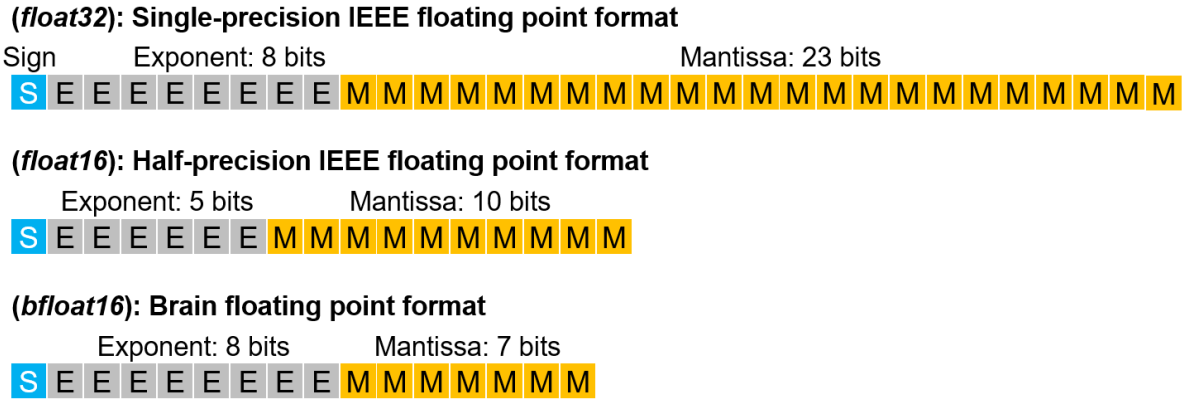

Figure 3. Schematic comparison of 32-bit IEEE floating point format, 16 bit IEEE floating point format, and the new 'brain' floating point format used for TPU chips.

Discussion and Summary

Modern 3D marine seismic volumes can represent arrays with several trillion values, so inversion-based solutions such as Full Waveform Inversion (FWI) was impossible on large scale datasets until computing capacity caught up in recent years. DNN pursuits with large 3D seismic datasets have not yet been attempted, but the computational overheads will at least be on a par in some scenarios with FWI. It is noted that published efforts to use DNNs to predict accurate velocity models from shot gathers are in its infancy (e.g., Araya-Polo et al., 2018), but it is clear that the training volumes and the associated computing resources will need to be significant to match FWI results. A quick look at the official 'Top 500' global supercomputers according to the High Performance Linpack (HPL) benchmark (https://top500.org/list/2019/06/) shows the largest computer has almost 2.5 million cores. The largest oil company representation (Total) has almost 300,000 cores. PGS is the only service company in the top 50 with almost 150,000 cores in its Cray XC30 installation. As the oil industry transitions to increased Cloud computing, ASIC computing resources on a vast scale may supplement the supercomputers used today.

The denoising example using a CNN in Figures 1 and 2 illustrated that Machine Learning has a future for seismic processing, but it also illustrated that huge training efforts are required for application to even modest input data sizes. As the applications of Machine Learning and Deep Neural Networks extend to ever-increasingly large and complex problems, the computational overheads will correspondingly become vast. Application Specific Integrated Circuits such as the Tensor Processing Unit developed for Google TensorFlow in the Cloud will expectably become more common, and will represent a new class of supercomputer with hundreds of PFLOPS of computing capacity available to solve each problem.

Additional Reading Material

- Araya-Polo, M., Jennings, J., Adler, A., and Dahlke, T., 2018, Deep-learning tomography. The Leading Edge, 37(1), 58-66. https://doi.org/10.1190/tle37010058.1
- Bekara, M., and Day, A., 2019, Automatic QC of denoise processing using a machine learning classification: First Break, 37(9), 51-58. https://www.pgs.com/globalassets/technical-library/tech-lib-pdfs/fb_bekara_et_al_sept2019_denoiseml.pdf
- Brandsberg-Dahl, S., 2017, High-performance computing for seismic imaging; from shoestrings to the cloud: 87th Technical program, SEG, Expanded Abstracts, 5273-5277. https://www.pgs.com/globalassets/technical-library/tech-lib-pdfs/seg2017_brandsberg-dahl_computing.pdf
- Convolutional Neural Network: Wikipedia. https://en.wikipedia.org/wiki/Convolutional_neural_network
- Long, A., 2019, Machine learning and seismic: Industry Insights. https://www.pgs.com/media-and-events/news/industry-insights---machine-learning-and-seismic/
- Long, A., and Martin, T., 2019, Automation of marine seismic data processing: Submitted to The Leading Edge, 38(12).