JST CREST Project on Scalable Deep Learning

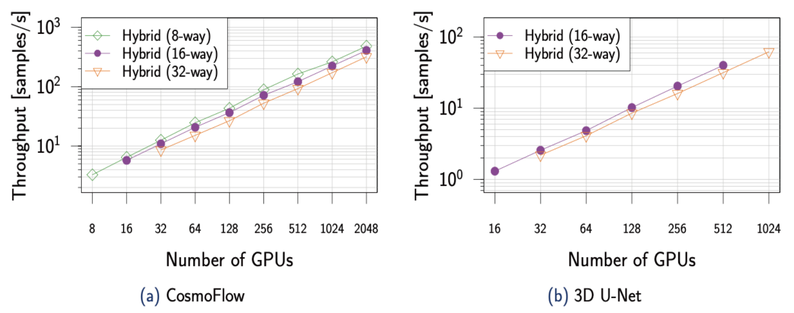
Scalable Deep Learning for Large Image and Video Data

About the project

The overreaching aim of the project is to establish a high-performance real-time deep learning algorithm basis for detecting objects and anomalies from a large amount of high definition videos recorded by drive recorders, surveillance cameras, and the like. Computer science researchers specializing in four different levels from architectures to applications including GPU fast computation, parallel computation, machine learning, and data mining, collaborate to realize 1000 times faster processing with 0.1% the amount of memory compared to conventional platforms.

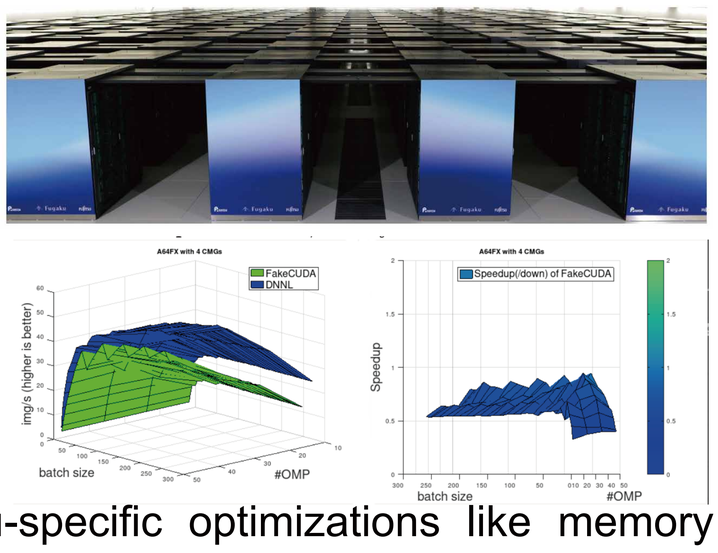
The Case for Strong Scaling in Deep Learning: Training Large 3D CNNs with Hybrid Parallelism

We present a new capability for spatially partitioning the training of 3D convolutional neural networks. Our approach enables training on very large data cubes that would otherwise be infeasible due to memory limitations. Due to the huge size of the data, we also develop a new data ingestion pipeline that leverages parallel I/O as well as an in-memory distributed data store in order to reduce I/O overheads. We study the impact of data size and present comprehensive performance and scaling results of two different huge 3D CNNs, the CosmoFlow network and 3D U-Net, including training a single model with up to 2048 V100 GPUs. Furthermore, we demonstrate an order-of-magnitude improvement in the prediction quality of the CosmoFlow network.



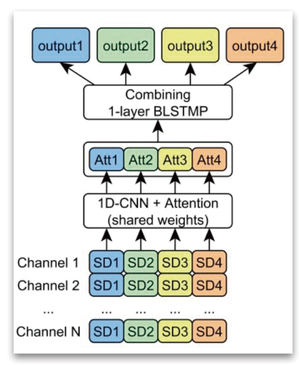
Creating Deep Learning Software Ecosystem for Fugaku

Fugaku has 6 TFLOPS FP32 performance and 1TB/s memory bandwidth per chip. Developing a deep learning software stack on this architecture requires porting DNNL to support SVE/A64FX optimization with some Fugaku-specific optimizations like memory management subsystems.

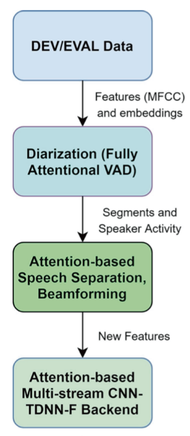


Fully-Attentional System For Noisy Speech

For the front-end (before speech recognition), use a fully attentional network for the improvement of speech diarization and speech separation. This in turn can lead to better speech enhancement from using speaker IDs. We represent our task of diarization in a way which utilizes axial- attention. We can also try various numbers of concurrent windows of MFCC spectrograms (previous, current, and lookahead) on which the attention model will focus. Additionally, since speech networks often use 1D CNNs, we can combine those vectors into a larger 2D vector.



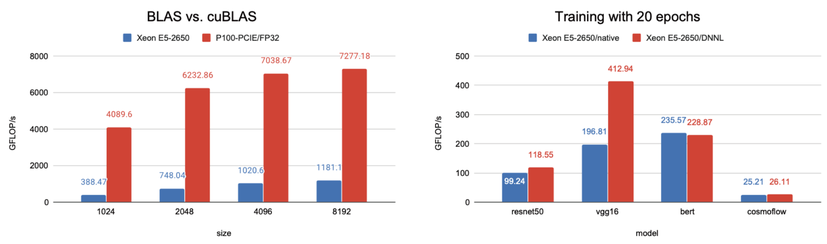
Multi-channel TS-VAD



Our Proposed System Pipeline

Understanding Deep Learning Performance

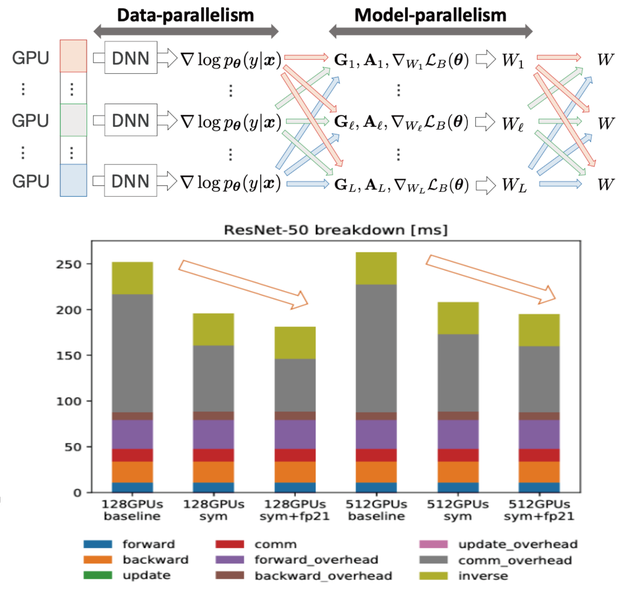
Benchmarking machine learning algorithms and the hardware it runs on is a non-trivial task.



Different models favor different hardware (e.g. transformer models are highly GPU organized, while smaller kernels tend to underutilize GPU parallelism). For this reason, detailed bench- marking is needed. We measured different models, examining the per- formance of their parts (transformers, convolutional layers, dense layers etc.) and compared them across different hardware (TPU and A64FX measurements are in progress).

A Systematic Performance Analysis of Second-order Optimization using K-FAC

We perform second order optimization on deep neural networks using Kronecker factored approximate curvature (K-FAC). Various performance optimizations were performedd including the use of fp16 gradients and fp21 Fisher matrices, hierarchical ring AllReduce communication, and the use of stale Fisher matrices. We achieved 2 minutes training of ResNet-50 / ImageNet-1K with 2048 GPUs by Distributed K-FAC.



Fast Approximations of Betweenness Centrality with Graph Neural Networks

Major key points of our feature aggregation schemes are: 1) Feature aggregation is done separately for incoming and outgoing paths. 2) Node' s own features are not aggregated. Hence at each layer node gets unique feature information corresponding to neighborhood structure. 3)Nodes with no shortest paths are identified and corresponding rows in A and AT are set to zero. This helps features flow along possible shortest paths.