RAMAIAH
Institute of Technology

Samsung Prism

Mentors

- Pradeep N S
- Dr. Mayukh Das
- Dr. Annapurna P. Patil
- Dr. S. Rajarajeswari

Team Members

- Akshat Jaitly
- Himangshu Shekhar Jha
- Aditya S M
- Sahil Bhave

Worklet Topic

Design and development of hardware aware DNN for Faster Inference by learning replacement models and approximation using PAUs

Work-let Area – AI, ML | Design and develop the hardware aware DNN for faster inference

SAMSUNG

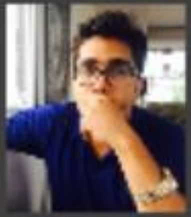
Problem Statement

- Design and develop the hardware aware DNN for faster inference by learning replacement model of operations and approximation using PAUs (https://arxiv.org/abs/1907.06732)

Additional Documentation:

- https://lilianweng.github.io/lil-log/2020/08/06/neural-architecture-search.html

Pradeep N S, Senior Chief Engnr II
[illegible]
+91-99 0008 3177



Dr. Mayukh Das, Senior Chief Engr
[illegible]
+91-797 5640 625

Work-let expected duration – 3 months

4 Members

Expectations

- Understanding in detail how different operations behave in DNNs
- This includes 2 major steps :
  a. Ad-hoc replacement of operations inside any given DNNs
  b. Integrate approximators with the adhoc replacement map
  c. Integrated into the automation pipeline
- Achieve pre-defined KPI for usecases that require adhoc customization of DNNs

Training/ Pre-requisites

- Courseera trainings available for ML / DNN basics
- Python Programming
- Pytorch / Tensorflow fundamentals
  https://arxiv.org/abs/1907.06732,
  https://lilianweng.github.io/lil-log/2020/08/06/neural-architecture-search.html

Kick Off < 2nd Week >

- Understanding Deep Learning concepts.
- Getting proficient with Deep learning development frameworks Such as Tensorflow/Pytorch
- Behavior of different DNN ops + understanding of PAUs

Milestone 1 < 1st Month >

- Implement learning pipeline for automated DNN customization
- Build a MAP for what can be changed while preserving performance

Milestone 2 < 2nd Month >

- Implement adhoc customization with approximation
- Develop DNN retraining post customization – all in a single pipeline

Closure < 3rd Month >

- Extensive experimentation to evaluate effectiveness of the approach

Kick Off < 2nd Week >

- Understanding Deep Learning concepts.
- Getting proficient with Deep learning development frameworks Such as Tensorflow/Pytorch
- Behavior of different DNN ops + understanding of PAUs

Milestone 1 < 1st Month >

- Implement learning pipeline for automated DNN customization
- Build a MAP for what can be changed while preserving performance

Milestone 2 < 2nd Month >

- Implement adhoc customization with approximation
- Develop DNN retraining post customization – all in a single pipeline

Closure < 3rd Month >

- Extensive experimentation to evaluate effectiveness of the approach

Implentation

Implemented a part of learning pipeline for automated DNN customization on activation functions :

- Given a model, our function extracts the layers and converts the activation function within those layers.

Model Convertors

Convert different Model to Pytorch Model

PyTorch convertor

Convert to PyTorch model.

nerox8664/gluon2pytorch

Convert mxnet / gluon graph to PyTorch source + weights.

ruotianluo/pytorch-resnet

Convert resnet trained in caffe to pytorch model.

clcarwin/convert_torch_to_pytorch

Convert torch t7 model to pytorch model and source.

vzhong/chainer2pytorch

`chainer2pytorch` implements conversions from Chainer modules to PyTorch modules, setting parameters of each modules such that one can port over models on a module basis.

pytorch-caffe

Load caffe prototxt and weights directly in pytorch without explicitly converting model from caffe to pytorch.

nn-transfer

Convert between Keras and PyTorch models.

Doubt

Is there a better way to convert model from one platform to pytorch?

Literature Survey

1) The Correct Way to Measure Inference Time of Deep Neural Networks https://towardsdatascience.com/the-correct-way-to-measure-inference-time-of-deep-neural-networks-304a54e5187f :-

- Asynchronous execution offers huge advantages for deep learning, such as the ability to decrease run-time by a large factor. For example, at inference of multiple batches, the second batch can be preprocessed on the CPU while the first batch is fed forward through the network on the GPU.
- When the GPU is not being used for any purpose it shuts down . So, if we measure time for a network that takes 10 milliseconds for one example, running over 1000 examples may result in most of our running time being wasted on initializing the GPU. This is called the GPU Warmup Phase

Literature Survey

Throughput Measurement

- The throughput of a neural network is defined as the maximal number of input instances the network can process in a unit time (e.g., a second).
- To achieve maximal throughput we would like to process in parallel as many instances as possible. This parallelism is data, device and model dependent.
- Steps involved in measuring the throughput :-

1)estimate the optimal batch size that allows for maximum parallelism

2) given this optimal batch size measure the number of instances the network can process in one second

Literature Survey

Throughput Measurement Elaborated

- To find the maximal batch size :-

We must reach the memory limit of our GPU for the given data type. This size depends on the hardware type and the size of the network.

Then , the number of examples our network can process in one second would be: (number of batches X batch size)/(total time in seconds).

Literature Survey

2) Latency and Throughput Characterization of Convolutional Neural Networks for Mobile Computer Vision
https://arxiv.org/pdf/1803.09492.pdf

- With both the Android and Jetson the latency-throughput increases are not linearly dependent on batch size. Instead, sometimes an increase of one image in the batch size leads to a better throughput with a minimal increase in latency, and on other other occasions the increase results in both worse throughput and increased latency. Depending on the inference model and the computation platform certain batch sizes are thus more optimal than others. The optimal batch size is difficult to estimate without actual measurements.

Literature Survey Cont.

"The good news is that the results confirm that you can significantly speed up the model inference without retraining or modifying it. The challenge is to determine which configuration should be used on which device."

What we propose

- **Brute Force**

  Replace and check with all activation function until we find better results.

- **Learn**

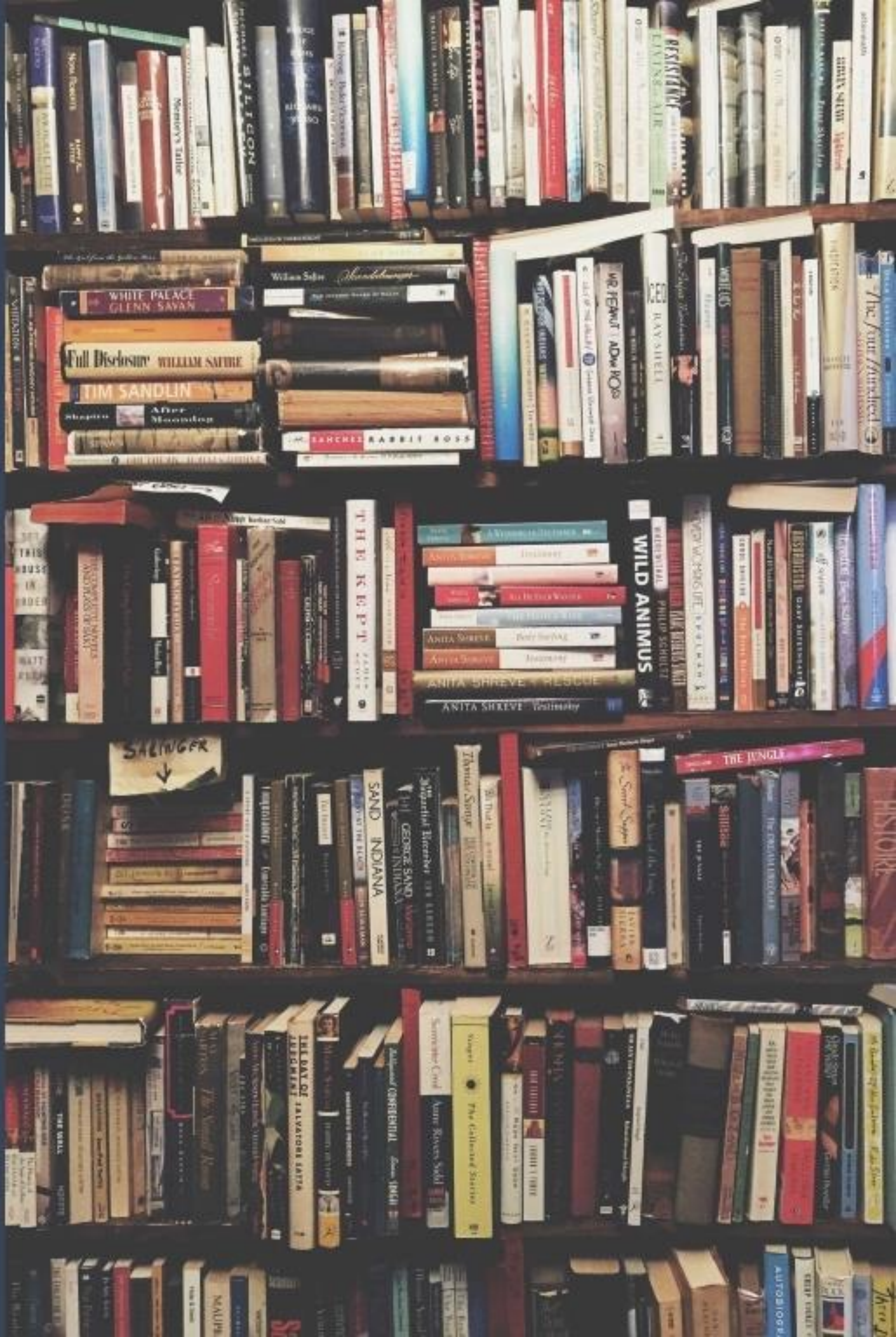
  Learn from the previous step and approximate the behavior of each change performed in previous step.

- **Narrow Down the search space**

  For a new hardware with similar specs, Predict the behavior before hand and prune the brute force search space.

Example

Suppose we have specs X and a subset of activations that have high accuracy and low latency (say {RELU,RELU6}) so, if we get some specs Y that is nearer to X , rather than searching for multiple activation functions, we can narrow down our search to {RELU, RELU6}.



Doubts

1. Is it possible to reduce the search space based on the activation functions that probably works best on the particular device specs.
2. Can you guide us on how to run and check the model in our mobile.
3. What are some hardware specifications or points on which the latency and throughput could depend.

References

- Latency and Throughput Characterization of Convolutional Neural Networks for Mobile Computer Vision https://arxiv.org/pdf/1803.09492.pdf
- The Correct Way to Measure Inference Time of Deep Neural Networks : https://towardsdatascience.com/the-correct-way-to-measure- inference-time-of-deep-neural-networks-304a54e5187f :-
- https://github.com/ysh329/deep-learning-model-convertor

Thank You
for listening!