Deep Neural Networks' Compression Techniques Study

Appling Compressing Methods for Investigating their Effects on Siren based Deep Neural Networks

Chiarlo Francesco Maria

University Polytechnic of Turin

February 16, 2021

POLITECNICO
DI TORINO

Siren Deep Neural Network Architectures

Main characteristics of Siren based Deep Models employed for as major purpose of implicitly representing input processed image:

- It belongs to **MLP family** of Deep Learning Networks, this means that It resembles a Fully Connected Architecture;
- It employes as non-linear activation function a **trigonometric sine function**.
- It is feeded by means of a set of **input coordinates** throught which **output pixel value**, in other words pixel magnitude, in a given scale and so range of values is predicted;
- It does not employes a well-known and widespread weigths initialization technique as **Xavier Initialization** but instead decided to adopt a **Custom Uniform Initialization**, as described within the same Siren paper.

Siren Based Network Topolgy

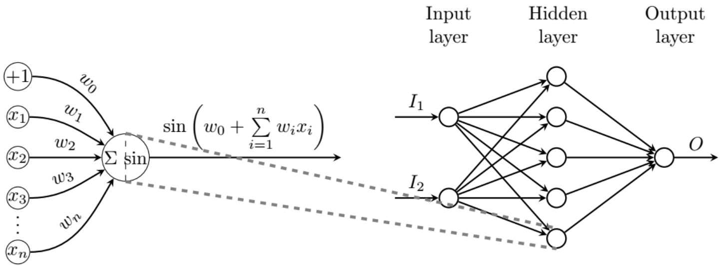


Figure: Siren Architecture Example

Example of tiny Siren like architecture, in order to show the minimal configuration that can be employed to describe network's major behavior.

Compression Techniques

Deep Neural Network Models Compression Techniques can be devided into a given number of different classes or categories depending on the approaches they follow. In particular we can devide such methods, without the claim of being for sure exhaustive but at least quite accurate, as follows:

- Pushing state-of-the-art models on salient tasks within domanins as **Computer Vision, NLP** corresponds to models becoming larger, increasing **Memory and Storage Requirements** at both computation and storing time.
- Bigger models lead to **Larger Carbon Footprint**, where this issue follows also from observation just done above.
- Other issues araise when we are attempting to deploy state-of-the-art models within those contexts where we find ourselves constrained by computing limited resources, as well as, reduced amount of battery capability as for **Mobile Applications and IoT Devices**.

Compression Techniques (2)

Deep Neural Network Models Compression Techniques have been studied and are now a day in ever high demand due to several reasons which can be summerized in the following:

Weight Sharing

- Cluster-based Weight Sharing
- Learning Weight Sharing
- Weight Sharing in Large Archs
- ...

Network Pruning

- Pruning via Weights Regularization
- Pruning via Loss Sensitivity
- Structured Pruning
- ...

Quantization

- Adaptive Range and Clipping
- Linear Range Quant
- DoReFa Net Quant
- WRPN Net Quant
- ...

Knowledge Distillation

- Recurrent (Autoregressive NNs
- Transformer-based (Non-Autoregressive) NNs
- Data Free KD
- ...

Selected Compression Techniques

The Compression Techniques we decided to adopt and investigate among the wide varieaty of possible choices are the two following:

- **Automated Gradual Pruning** by *Michael Zhu et al., 2017* - an instance of a possible pruning like compressing method:
  - Model pruning is the art of discarding the weights that **do not improve a model's performance**, in other words **non-significant or non-salient parameters**;
  - Careful pruning enables us to compress and deploy our **state-of-the-art neural networks** onto **mobile phones and other resource-constrained devices**.
- **Linear Range Quantization** by *Benoit et al., 2018* - an instance of a possible quantizing aware training method;

Automated Gradual Pruner

Automated Gradual Pruner (AGP), discussed in **To prune, or not to prune: exploring the efficacy of pruning for model compression**, the authors Michael Zhu and Suyog Gupta propose an intuitive, fresh new pruning approach briefly described as:

- **new automated gradual pruning algorithm** in which the sparsity is increased from an initial sparsity value s_i (usually 0) to a final sparsity value s_f over a span of n pruning steps;
- The intuition behind this sparsity function in equation below is to **prune the network rapidly in the initial phase** when:
  - the redundant connections are abundant, and;
  - gradually reduce the number of weights being pruned each time as there are fewer and fewer weights remaining in the network.

Automated Gradual Pruner (2)

Other interesting properties related to **Automated Gradual Pruner (AGP)**, and discussedwithin Michael Zhu and Suyog Gupta's paper, for better appreciating and understanding the usefulness of such a technique are:

- It Requires a lower number of trials, and attempts for identifying meaningful set of hyper-params for leading the pruning approach, compared to other techniques such Magnitude Level Pruning and other similars;
- It does not made particular assumptions on weight values density distribution;
- It is agnostic with respect to the particular Deep Neural Network Architecture chosen;

Target Image for Training Siren Models

Differently from Siren paper's Camera Image, we decide to resize it, by cropping the full image down to 256x256 image about its center, leading to the following update image:

Figure: Camera 256x256 target image

Image Feature	Value
name	Camera
shape	(256, 256)
size_byte	65536
image_band	(L,)

Table: Cropped Camera Image Characteristics

Data Distribution for Target Image for Training Siren Models

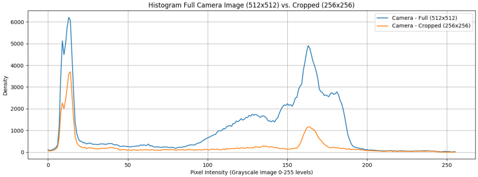
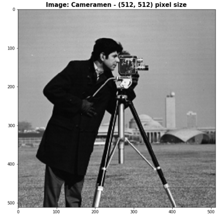
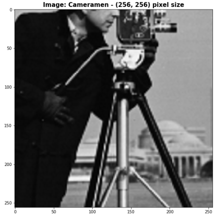
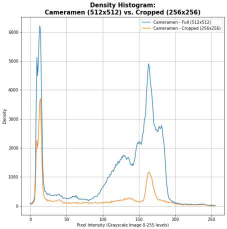


Figure: Camera target image, full and cropped data distribution

General Summarizing Image Overview







Summary Table Image Cameramen: full size and cropped versions

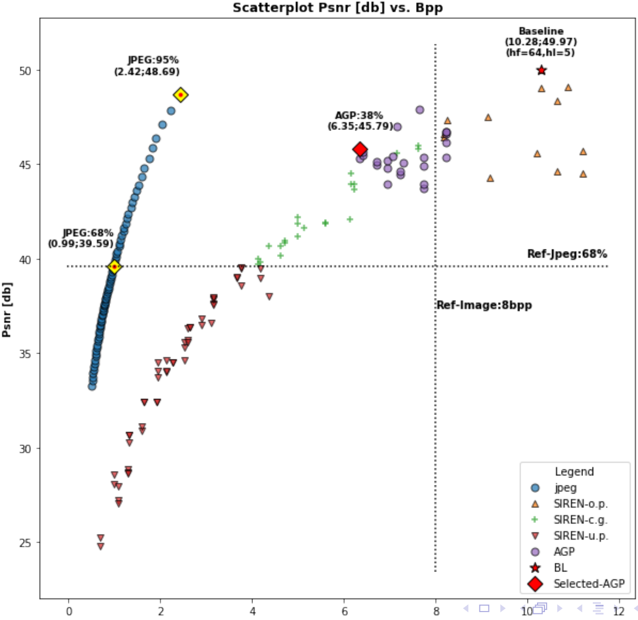
	name	shape	size_byte	image_band	entropy
full	Cameramen	(512, 512)	262144	(L,)	7.047955
cropped	Cameramen	(256, 256)	65536	(L,)	6.780318

Pruning Technique Workflow

For pruning Siren based Deep Models via AGP pruning method, and after having performed some random trials, via **Random Search Approach** to indetify most suitable hyper-params, we follow the subsequent training strategy:

- We determine the degree of pruning for each layer depending on the **Sensitivity Level Analysis** done ahead of pruning time.
- We let the **Frequency** Hyper-parameter to be picked up from: $\{50, 100, 200\}$ possible choices, essential to let net model to mitigate or recovery from **brain damage** induced while removing not salient weigths;
- We train each model for a **Number of Epochs** equals to 150000 before turning off pruning and let the model to finish remaining epochs;
- We both train **from scratch** by means of AGP algorithm or prune an **already trained, for reaching overfitting state, models**;
- We follow a generically **Suggested Heuristic in literature** for not pruning first and last layers in order to let performance to not degradate too much;

AGP Pruning Technique: obtained results



Linear Range Quant

Automated Gradual Pruner (AGP), discussed in **To prune, or not to prune: exploring the efficacy of pruning for model compression**, the authors Michael Zhu and Suyog Gupta propose an intuitive, fresh new pruning approach briefly described as:

- **new automated gradual pruning algorithm** in which the sparsity is increased from an initial sparsity value s_i (usually 0) to a final sparsity value s_f over a span of n pruning steps;
- The intuition behind this sparsity function in equation below is to **prune the network rapidly in the initial phase** when:
  - the redundant connections are abundant, and;
  - gradually reduce the number of weights being pruned each time as there are fewer and fewer weights remaining in the network.

Linear Range Quant (2)

Other interesting properties related to **Automated Gradual Pruner (AGP)**, and discussedwithin Michael Zhu and Suyog Gupta's paper, for better appreciating and understanding the usefulness of such a technique are:

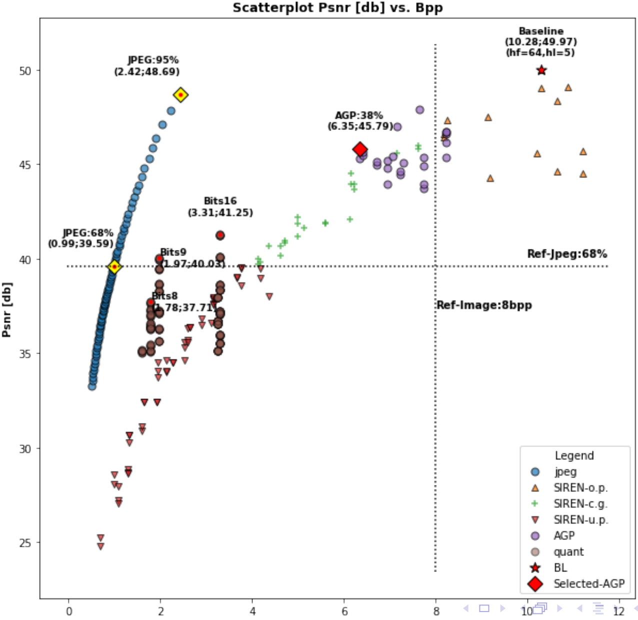
- It Requires a lower number of trials, and attempts for identifying meaningful set of hyper-params for leading the pruning approach, compared to other techniques such Magnitude Level Pruning and other similars;
- It does not made particular assumptions on weight values density distribution;
- It is agnostic with respect to the particular Deep Neural Network Architecture chosen;

Linear Range Quant Workflow

For pruning Siren based Deep Models via AGP pruning method, and after having performed some random trials, via **Random Search Approach** to indetify most suitable hyper-params, we follow the subsequent training strategy:

- We determine the degree of pruning for each layer depending on the **Sensitivity Level Analysis** done ahead of pruning time.
- We let the **Frequency** Hyper-parameter to be picked up from: $\{50, 100, 200\}$ possible choices, essential to let net model to mitigate or recovery from **brain damage** induced while removing not salient weigths;
- We train each model for a **Number of Epochs** equals to 150000 before turning off pruning and let the model to finish remaining epochs;
- We both train **from scratch** by means of AGP algorithm or prune an **already trained, for reaching overfitting state, models**;
- We follow a generically **Suggested Heuristic in literature** for not pruning first and last layers in order to let performance to not degradate too much;

Linear Range Quantization: obtained results



Conclusions and Remarks