Compression strategies and space-conscious representations for deep neural networks

G. Marinò, G. Ghidoli, M. Frasca and D. Malchiodi

UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI INFORMATICA

Problem description

Deploying large convolutional neural networks (CNNs) on limited-resource devices is still an open challenge in the big data era. The proposed implementation offers different compression schemes and two compact representations: the Huffman Address Map compression (*HAM*) and its sparse version *sHAM*.

Compression

Pruning (Pr)

Method: "cutting" all connections whose weights has small absolute value, so that the global network output does not sensibly change.
Parameters: threshold p used in order to deem a connection as negligible, defined in function of connection weight empirical quantiles.
Post-processing: retraining of the network, ignoring the erased connections.

Weight sharing (WS)

Method: clustering all learnt weights using the k-means algorithm, obtaining representative centroids used to replace weight values.
Parameters: number k of centroids.
Post-processing: retraining of the network, updating centroids through cumulative gradient.

New technique

Probabilistic quantization (PQ)

Method: uniformly partitioning weights, randomly "collapsing" each weight within to one of the extremes of the interval it belongs to:

$$\mathrm{P}(W = \underline{w}) = \frac{\overline{w} - w}{\overline{w} - \underline{w}}, \mathrm{P}(W = \overline{w}) = \frac{w - \underline{w}}{\overline{w} - \underline{w}}$$

Parameters: number k of extremes.
Post-processing: same as in WS.

HAM and *sHAM*

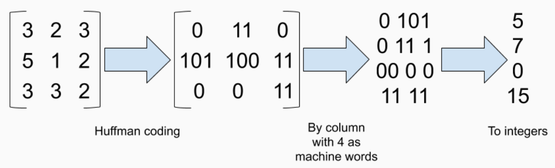
Novel formats

Weight values are represented through Huffman coding, subsequently concatenating the corresponding codewords by column order of connection matrix, obtaining a unique binary string.

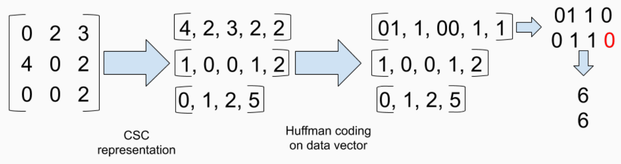
HAM encodes all weights, and the lower the number of distinct weights in the matrix, the lower the average codeword length.

To benefit also from the matrix sparsity, *sHAM* applies Huffman coding only to non zero elements, stored through Compressed Sparse Column (CSC) format.

In both *HAM* and *sHAM* cases, the generated bit sequence is organized as a succession of machine words, interpreted as an array of integers.



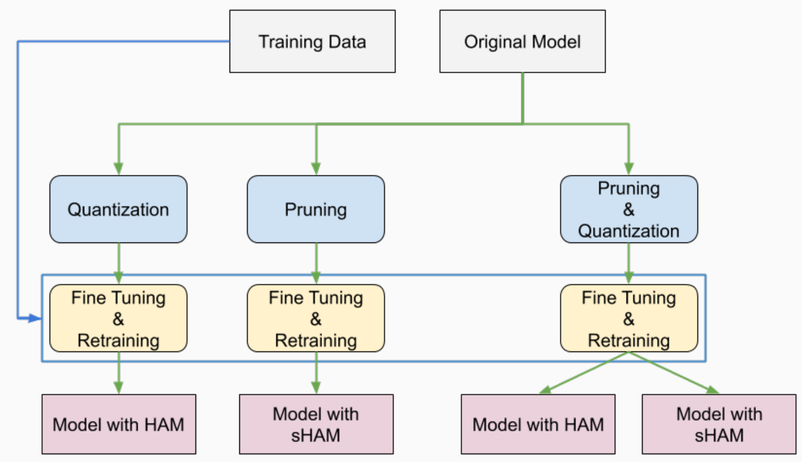
HAM



sHAM

Architecture & processing pipeline

1. Retrieval of pre-trained CNN + training dataset.
2. Network pruning and/or quantization.
3. Model retraining.
4. Model transformation to *HAM* or *sHAM* formats.
5. Assessment of the compressed model performance.



Networks

VGG19 Sixteen convolutional layers and one fully-connected block, trained on CIFAR-10 and MNIST.

DeepDTA Two separate convolutional blocks (for proteins and ligands) joined through three fully-connected hidden layers, trained on DAVIS and KIBA.

Data

Image recognition (classification)

	Size	Resolution
MNIST	70k	28x28 grayscale
CIFAR-10	60k	32x32 color

Drug targeting (regression)

	# Proteins	# Ligands	# Interactions
DAVIS	422	68	30056
KIBA	229	2111	118254

Evaluation

- metrics: accuracy (classification) and MSE (regression);
- performance difference $\triangle_{\text{perf}}$ (compressed w.r.t. uncompressed);
- testing time ratio (compressed w.r.t. uncompressed);
- space occupancy ratio ψ (compressed w.r.t. uncompressed).

Time and space performance account only for the actually compressed weights, that is those in fully-connected layers.

Results

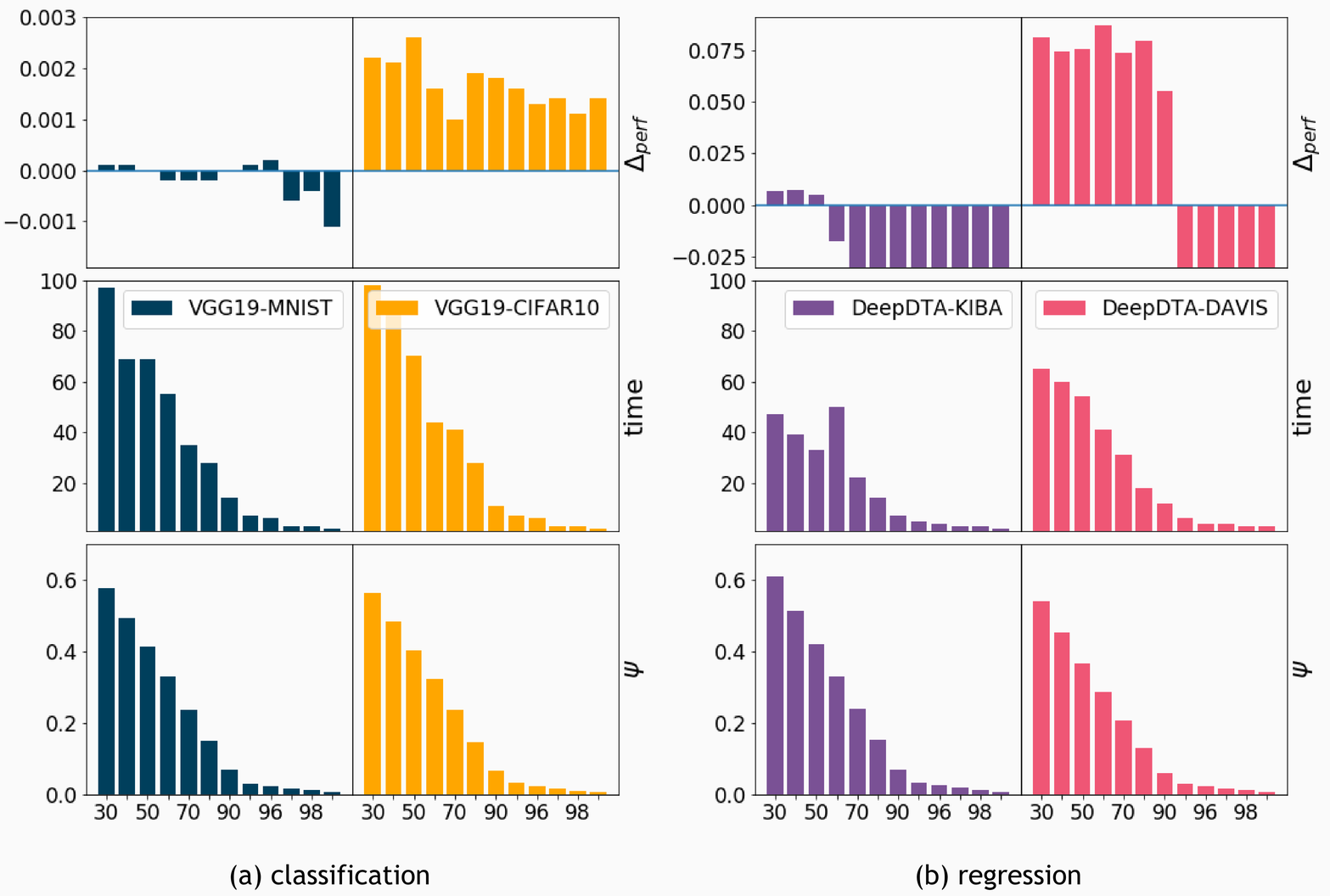
Best occupancy ratio ensuring no decay in performance w.r.t. uncompressed model. *Type* is the compression technique, *Perf* contains Accuracy for VGG19 and MSE for DeepDTA, ψ is the occupancy ratio, whereas $*$ denotes sHAM representation as the lowest occupancy on that setting (w.r.t. HAM).

Net-Dataset	Type	Configuration	Perf	ψ
VGG19-MNIST	Pr-PQ	97/32-32-2	0.9955	0.018*
VGG19-CIFAR10	Pr-WS	99/32-32-2	0.9358	0.006*
DeepDTA-KIBA	Pr-WS	60/32-2-32-2	0.1739	0.127
DeepDTA-DAVIS	Pr-PQ	90/128-32-32-32	0.2671	0.060*

Top testing performance achieved by compression techniques. Same notations as in the table above.

Net-Dataset	Type	Configuration	Perf	ψ
VGG19-MNIST	Pr-PQ	50/32-32-2	0.9958	0.187
VGG19-CIFAR10	WS	32-32-2	0.9371	0.306
DeepDTA-KIBA	Pr	60	0.1599	0.8
DeepDTA-DAVIS	Pr	80	0.2242	0.4

On the right: joint application of (a) pruning and weight sharing on classification datasets; (b) pruning and probabilistic quantization on regression datasets. We show the opposite values of $\triangle_{\text{perf}}$ for visualization purposes.



(a) classification

(b) regression