Neural Network compression

NN compression methods

- Network compression is a widespread technique to reduce the size, energy consumption, and overtraining of deep neural networks

- Several approaches have been studied:

  - **parameter pruning:** selective removal of weights based on a particular ranking [arxiv.1510.00149, arxiv.1712.01312]

  - **low-rank factorization:** using matrix/tensor decomposition to estimate informative parameters [arxiv.1405.3866]

  - **transferred/compact convolutional filters:** special structural convolutional filters to save parameters [arxiv.1602.07576]

  - **knowledge distillation:** training a compact network with distilled knowledge of a large network [doi:10.1145/1150402.1150464]

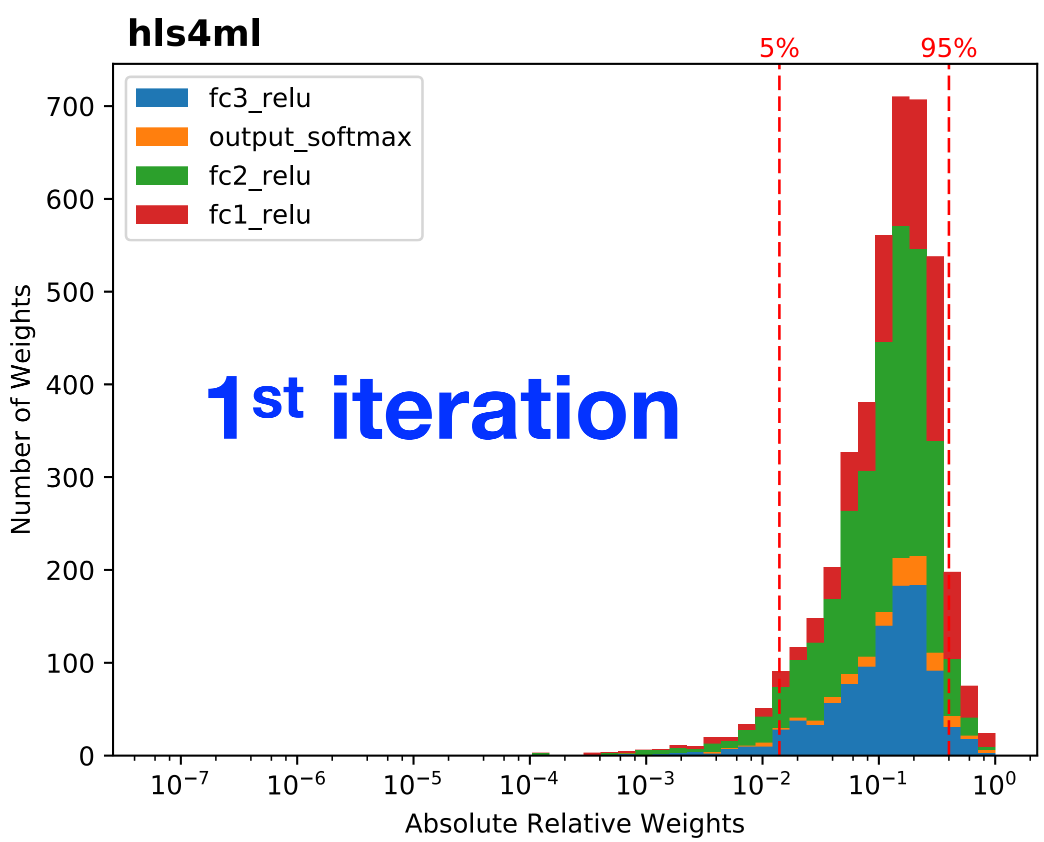
- Today you'll learn about the first method: **parameter pruning**

Compression with parameter pruning

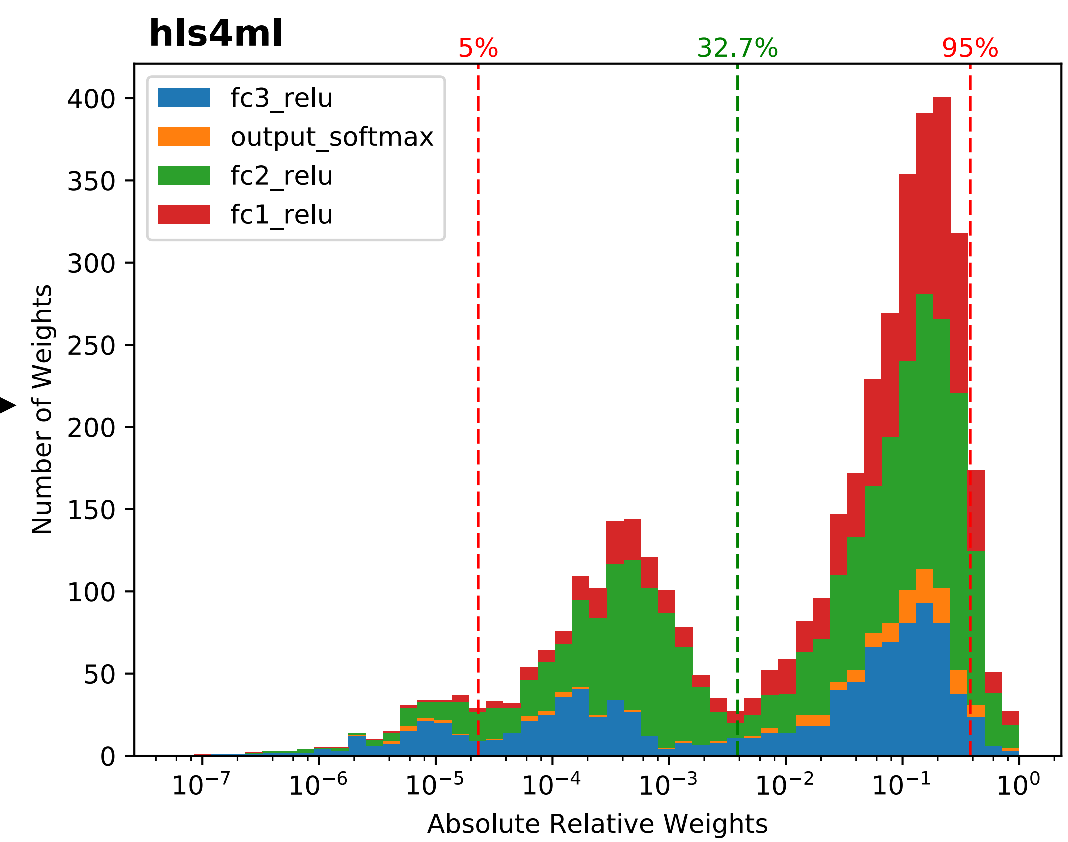
- Iterative approach:
  - train with **L1 regularization** (loss function augmented with penalty term):

$$L_\lambda(\vec{w}) = L(\vec{w}) + \lambda||\vec{w}_1||$$

  - sort the weights based on the value relative to the max value of the weights in that layer



Train with L_1 →



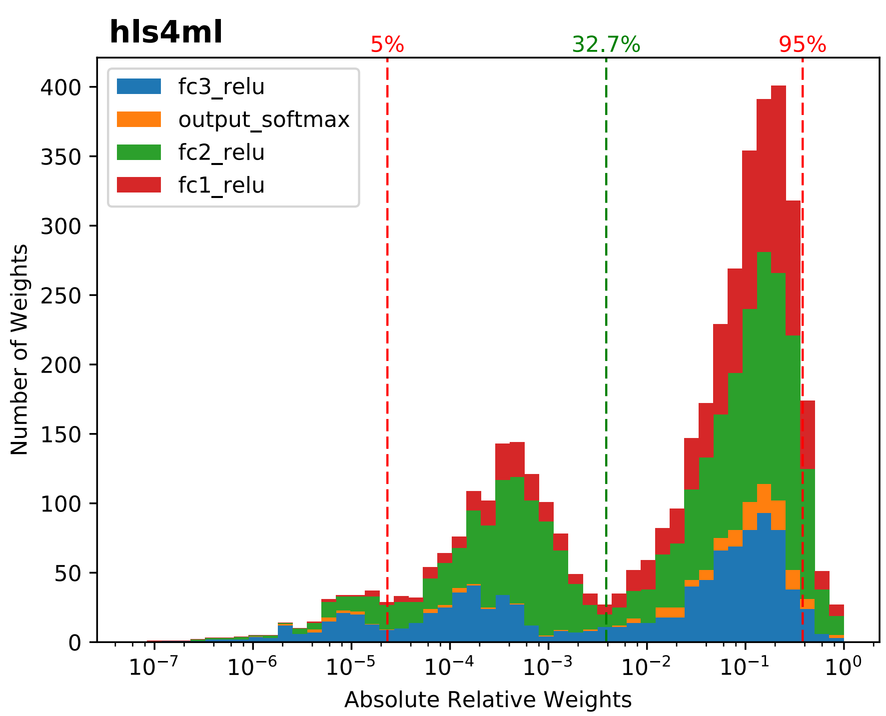
Compression with parameter pruning

- Iterative approach:
  - train with **L1 regularization** (loss function augmented with penalty term):

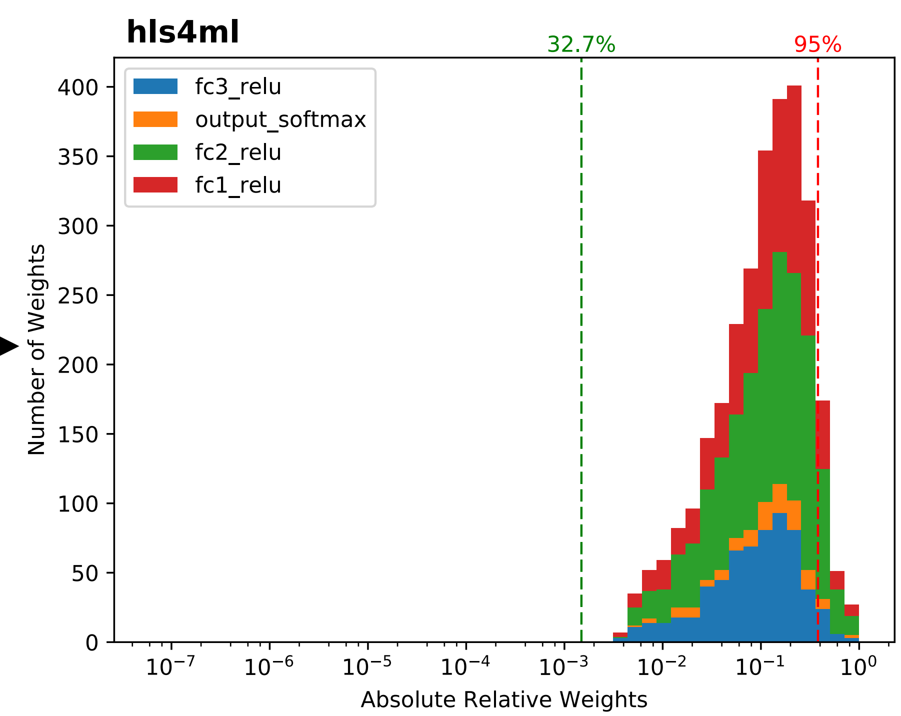
$$L_\lambda(\vec{w}) = L(\vec{w}) + \lambda||\vec{w}_1||$$

  - sort the weights based on the value relative to the max value of the weights in that layer
  - prune weights falling below a certain percentile and retrain

Train with L_1

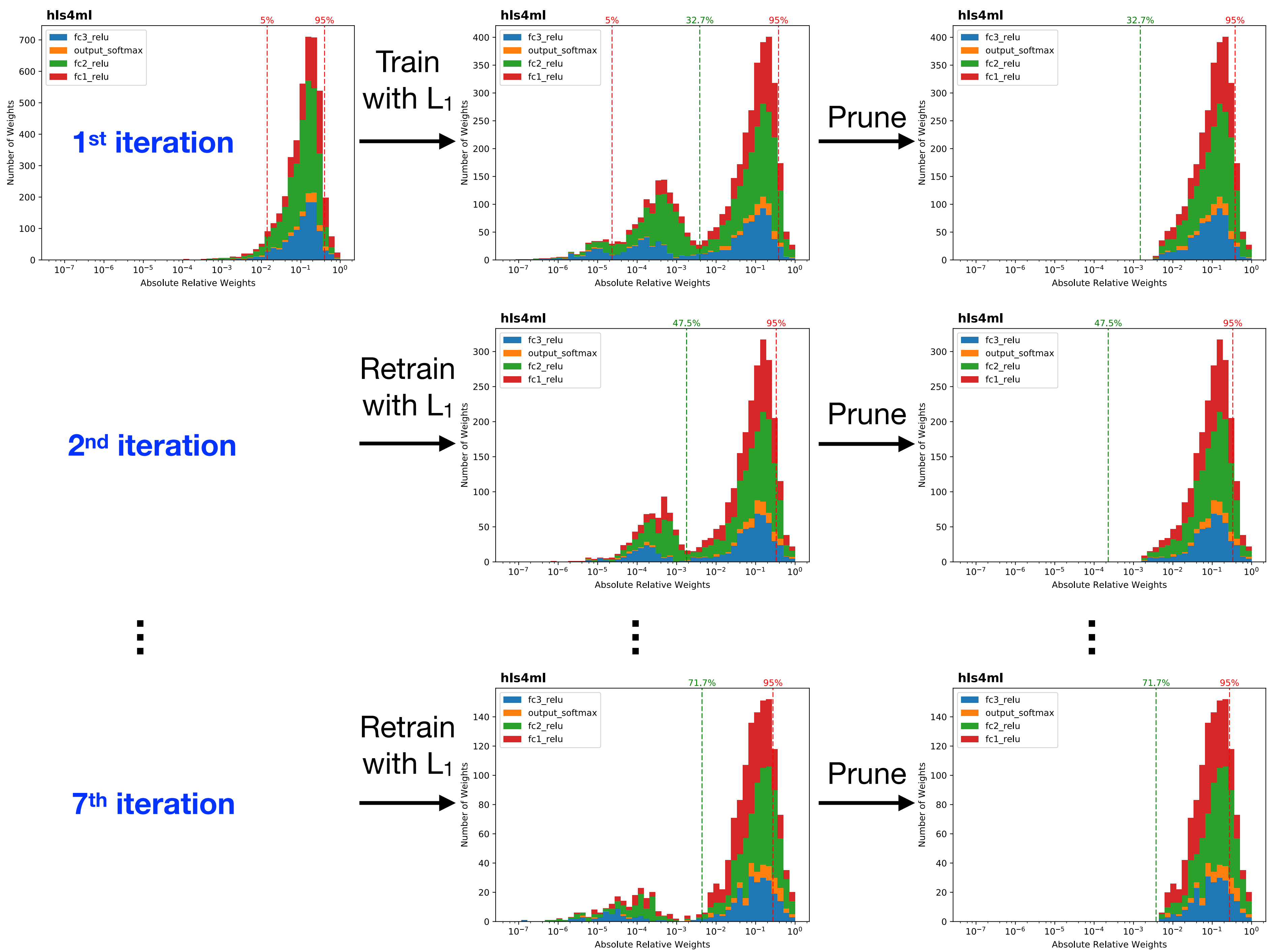


Prune



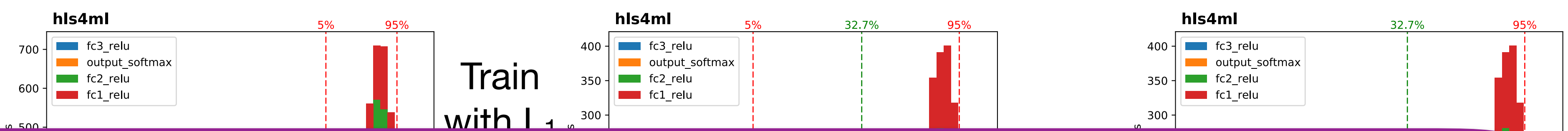
Compression with parameter pruning

Prune and repeat the train for 7 iterations

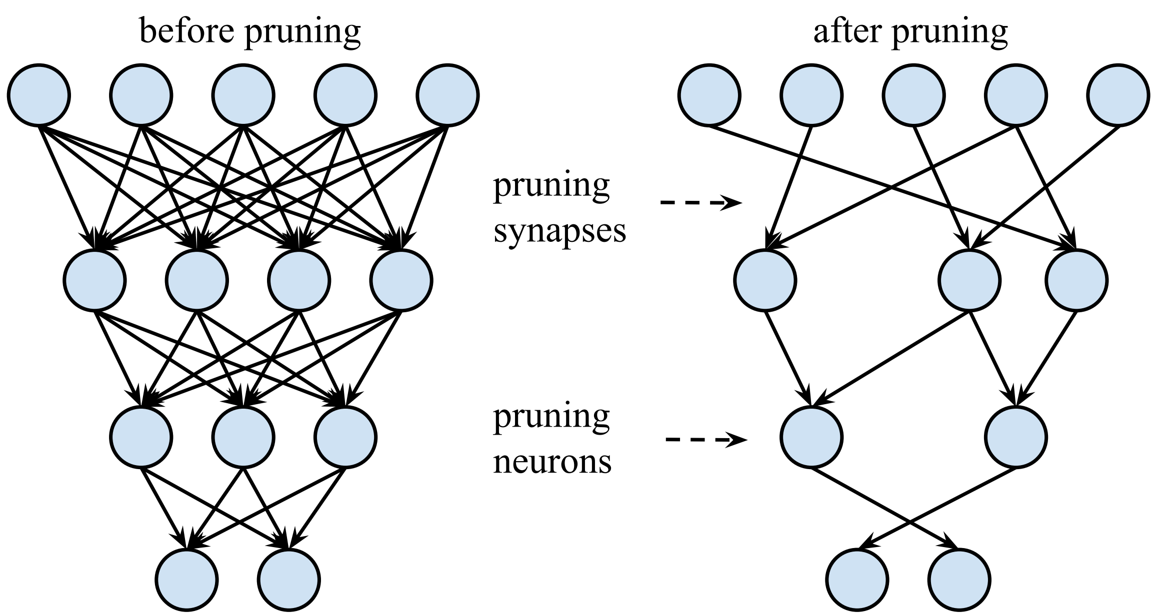


Compression with parameter pruning

Prune and repeat the train for 7 iterations



Train with L_1

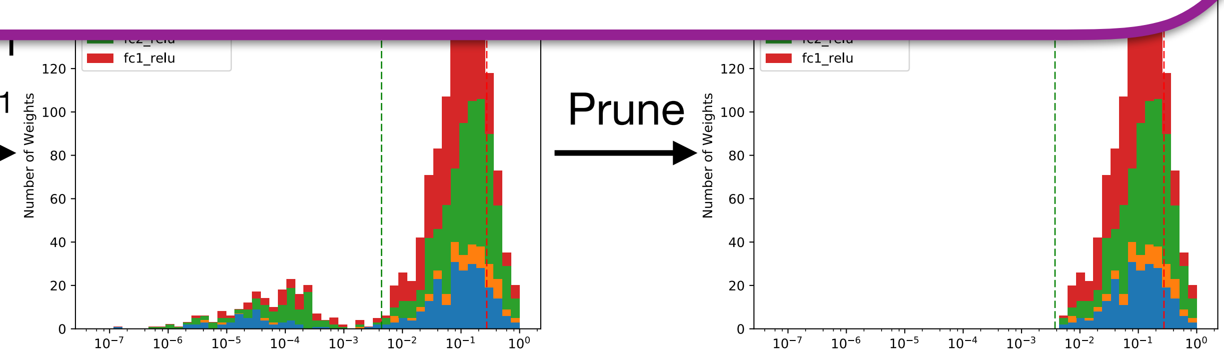


→ 70% reduction of weights and multiplications w/o performance loss

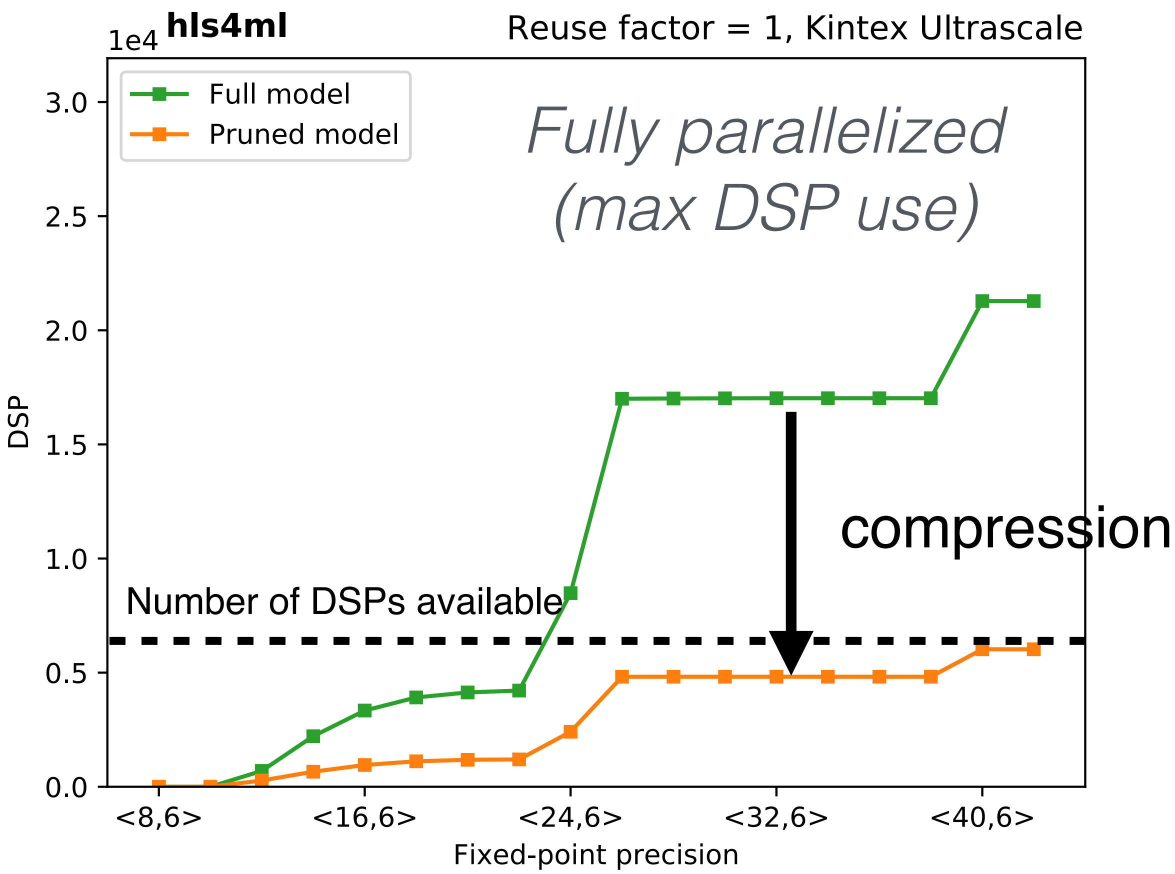
Retrain with L_1

7^{th} iteration

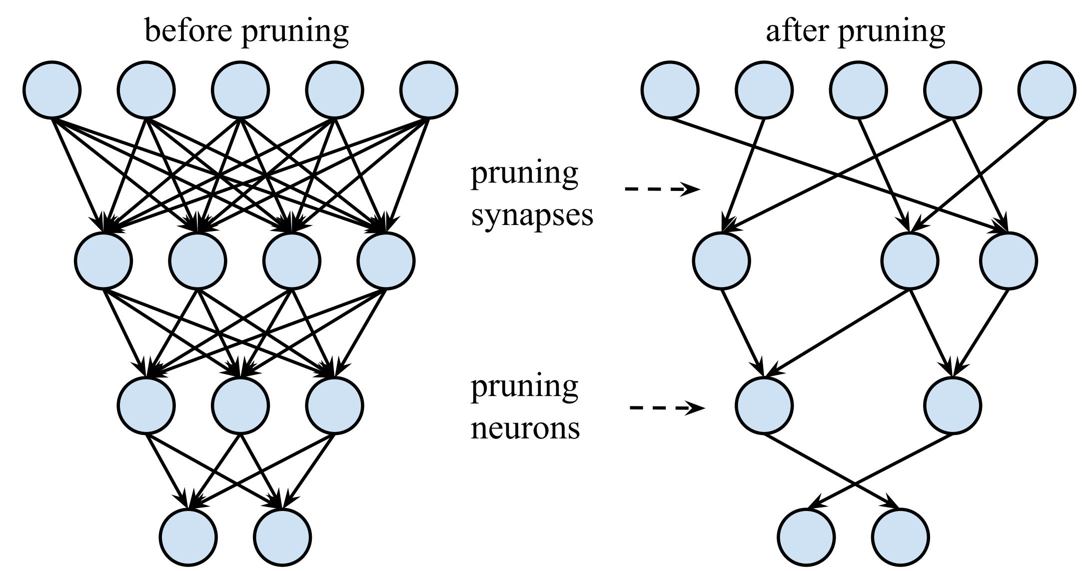
Prune



Efficient NN design: compression



70% compression ~ 70% fewer DSPs



- DSPs (used for multiplication) are often limiting resource
  - maximum use when fully parallelized
  - DSPs have a max size for input (e.g. 27x18 bits), so number of DSPs per multiplication changes with precision