Hcore-Init: Neural Network Initialization based on Graph Degeneracy
Stratis Limnios1,2, George Dasoulas1,3, Dimitrios M. Thilikos4, Michalis Vazirgiannis1
École Polytechnique France1, Alan Turing Institute London2, Noah’s Ark Lab Huawei Technologies3, LIRMM Univ
Montpellier/CNRS4

Introduction

Goal: Extraction of meaningful information from a Neural Network (NN) architecture:
- Construction of a Degeneracy-based Decomposition of a Neural Network architecture.
- Capitalization on the graph structure of a Neural Network for performance improvement.

Contributions:
- A unified method of constructing the graph representation of a neural network as a block composition of the given architecture.
- A new degeneracy framework, namely the k-hypercore, extending the concept of k-core to bipartite graphs.
- A novel weight initialization scheme, Hcore-init by using the information provided by the weighted version of the k-hypercore of a NN extracted graph, to re-initialize the weights of the given NN.

Preliminary Concepts and Definitions

INITIALIZATIONS:
- Glorot Initialization [Glorot, X. & Bengio, Y. in AISTATS (2010)]
  - The weights W are drawn from a normal distribution.
  - We ensure E[|W|] = 0 Var(|W|) = fanin/4, where fanin is the number of incoming neurons.
- Kaiming He Initialization [He, K., Zhang, X., Ren, S., & Sun, J. In ICCV (2015)]
  - Unlike the Glorot initialization, this method takes into account the activation function used.
  - The weights W are drawn from a normal distribution.
  - In the case of a ReLU: E[|W|] = 0 and Var(|W|) = fanin/4, where l is the index of the l-th layer and n the number of neurons in the given layer.

Note that the condition E[|W|] = 0 is essential for the Variance to be optimal.

HYPERGRAPH:
A hypergraph is a generalization of a graph in which an edge can join any number of vertices. It can be represented as H = (V, E_H) where V is the set of nodes, and E_H is the set of hyperedges, i.e. a set of subsets of V. Therefore E_H is a subset of 2^V.

Hence, we can transform any given MLP or Convolutional NN into a series of bipartite graphs.

Hcore (Hcore) Decomposition

HCORE DEFINITION:
Given a hypergraph H = (V, E_H) we define the (k, l)-hypercore as a maximal connected subgraph of H in which all vertices have hyperdegree at least k and all hyperedges have at least l incident nodes.

Hcore-Init: Weight Initialization

METHOD: The graph-based initialization method consists of:
- Pretraining of NN for x epochs.
- Construction of weighted graph structure of NN architecture.
- Hypercore decomposition of the constructed graph.
- Weight initialization of the NN based on the output hypercore values.

Weights on MLP: Re-initialization with weights drawn from a normal distribution with expectancy:
- for all i if w_ij ≥ 0, M = \frac{s_i}{\sum_{j=1}^{n} c_{ij}}.
- else M = \frac{s_i}{\sum_{j=1}^{n} c_{ij}}.

Hence w_ij follow a N(M, \frac{1}{2}) which variance is from the He initialization method.

CNN: For a given filter W ∈ ℝ^Hk×Hn it’s values are re-initialized with the following method:
- We define m for a given filter W as m(W') = \frac{1}{\sqrt{2}} \sum_j c_{ij} and m(W'') = \frac{1}{\sqrt{2}} \sum_j c_{ij}, if m(W') - m(W'') > 0 then M = m(W')
- else M = -m(W'')

Hence the general formula for m is given by:
M = sign(\arg\max(m(W'), m(W''))) max(m(W'), m(W''))
with sign(m(W')) = 1 and sign(W'') = -1.

PROPOSITION:
Let X_1 and X_2 two centered i.i.d. random variables with symmetric distribution.
We define X' = \max(X_0, X), X'' = \max(X_0, -X), and a real valued measurable function f : ℝ → ℝ such that ||f(X')|| < ∞ and ||f(-X'')|| < ∞.

Then:
- X', X'' are positive i.i.d. random variables.
- The random variable:
M = sign(\arg\max(f(X'), f(X''))) max(f(X'), f(-X'))
with sign(f(X')) = ±1, is centered, i.e. E[M] = 0.

Experimental Evaluation

Test accuracy (left) and train-loss (right) on CIFAR-10 on a fully connected convolutional neural network. The x in the label Hcore-Init-x stands for the number of pretraining epochs before applying hcore-init.

<table>
<thead>
<tr>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kaiming He</td>
<td>64.92</td>
<td>32.56</td>
</tr>
<tr>
<td>Hcore-Init</td>
<td>65.22</td>
<td>33.48</td>
</tr>
<tr>
<td>Hcore-Init-1</td>
<td>64.91</td>
<td>32.87</td>
</tr>
<tr>
<td>Hcore-Init-5</td>
<td>64.41</td>
<td>32.96</td>
</tr>
<tr>
<td>Hcore-Init-10</td>
<td>65.22</td>
<td>33.41</td>
</tr>
<tr>
<td>Hcore-Init-15</td>
<td>64.94</td>
<td>33.45</td>
</tr>
<tr>
<td>Hcore-Init-20</td>
<td>65.05</td>
<td>33.39</td>
</tr>
<tr>
<td>Hcore-Init-25</td>
<td>64.72</td>
<td>33.48</td>
</tr>
</tbody>
</table>

Table: Top Accuracy results over initializing the full model, only the CNN and only the FCNN for CIFAR-10, CIFAR-100, and MNIST. Hcore-Init represent the top performance over all the pretraining epochs configurations up to 25.