Hcore-Init: Neural Network Initialization based on Graph Degeneracy

Stratis Limnios[1,2], George Dasoulas[1,3], Dimitrios M. Thilikos[4], Michalis Vazirgiannis[1]

École Polytechnique France[1], Alan Turing Institute London[2], Noah's Ark Lab Huawei Technologies[3], LIRMM Univ Montpellier/CNRS[4]

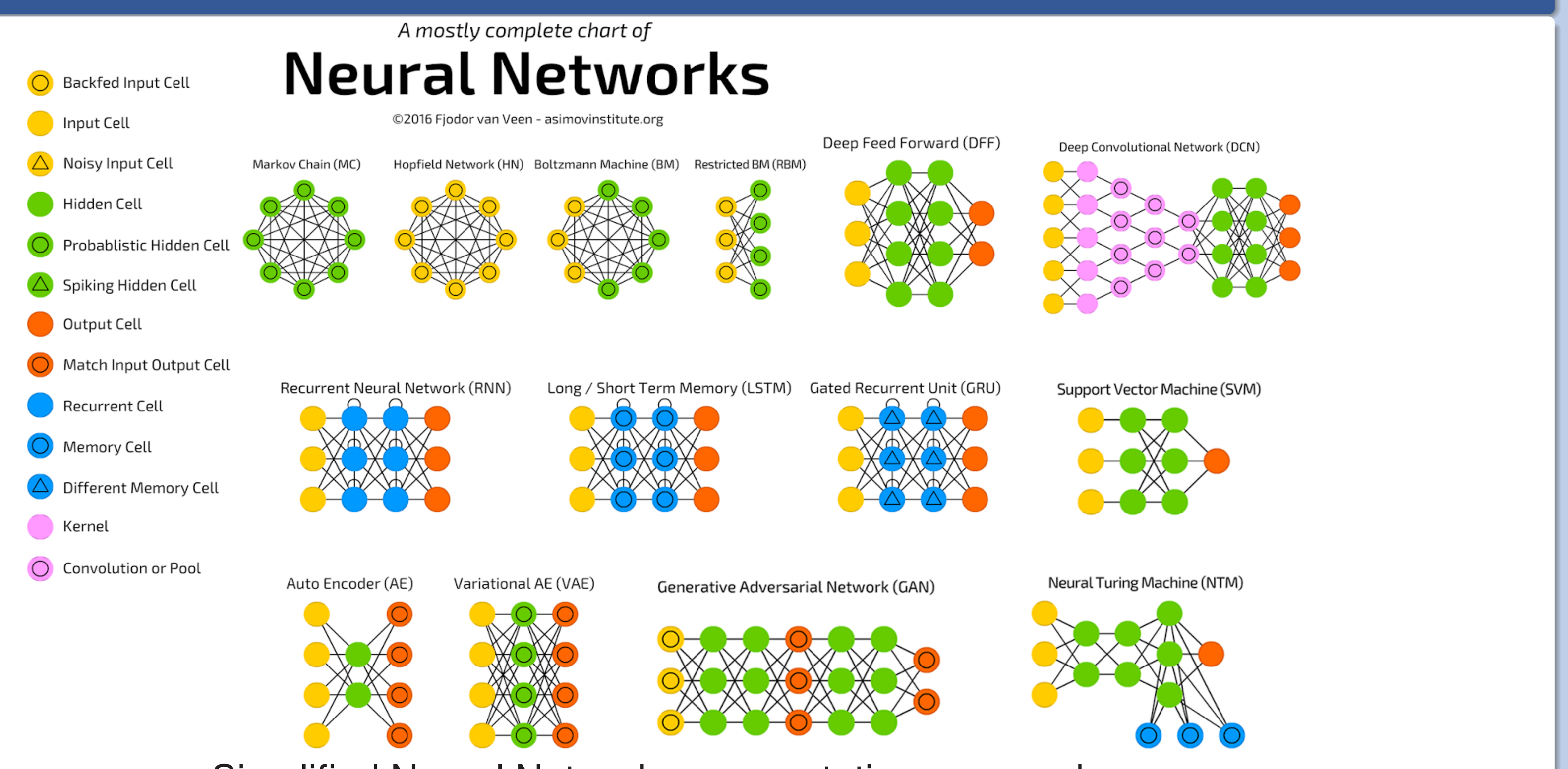
Introduction

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.



Simplified Neural Network representations as graphs

(https://www.asimovinstitute.org/author/fjodorvanveen/)

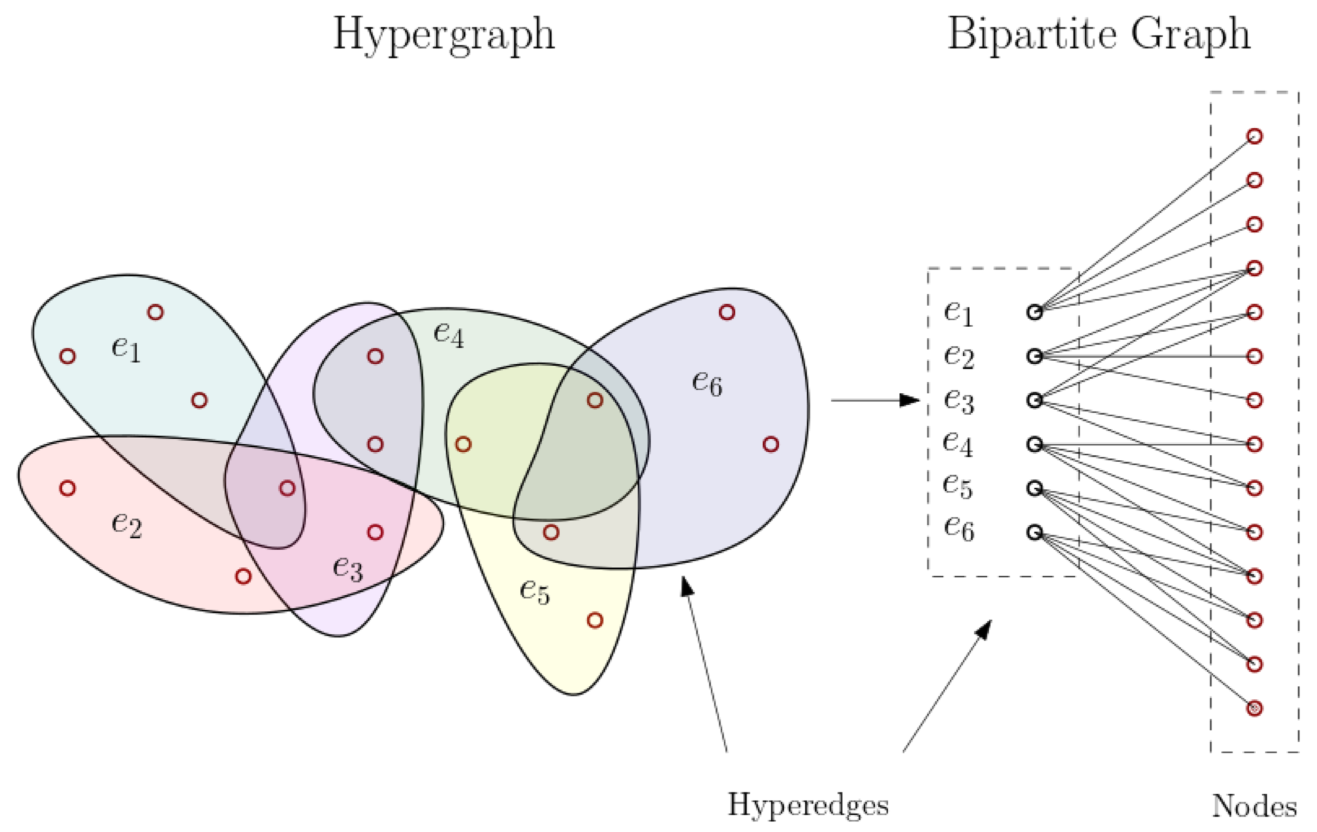
Preliminary Concepts and Definitions

INITIALIZATIONS METHODS:

1. **Glorot Initialization** [Glorot, X. & Bengio, Y. in AISTATS (2010)]
   - The weights W are drawn from a **normal** distribution.
   - We ensure $\mathbf{E}[W] = 0$ $\mathrm{Var}(w_i) = \frac{1}{\text{fanin}}$, where fanin is the number of incoming neurons.
   - Using both outgoing and ingoing neurons: $\mathrm{Var}(w_i) = \frac{1}{\text{fanin}+\text{fanout}}$.
2. **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*: $\mathbf{E}[W] = 0$ and $\mathrm{Var}[W] = \frac{2}{n^l}$, where l is the index of the l-th layer and n the number of neurons in the given layer.

Note that the condition $\mathbf{E}[W] = 0$ 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 $\mathcal{H} = (V, E_{\mathcal{H}})$ where V is the set of nodes, and $E_{\mathcal{H}}$ is the set of hyperedges, i.e. a set of subsets of V. Therefore $E_{\mathcal{H}}$ is a subset of $\mathcal{P}(V)$.

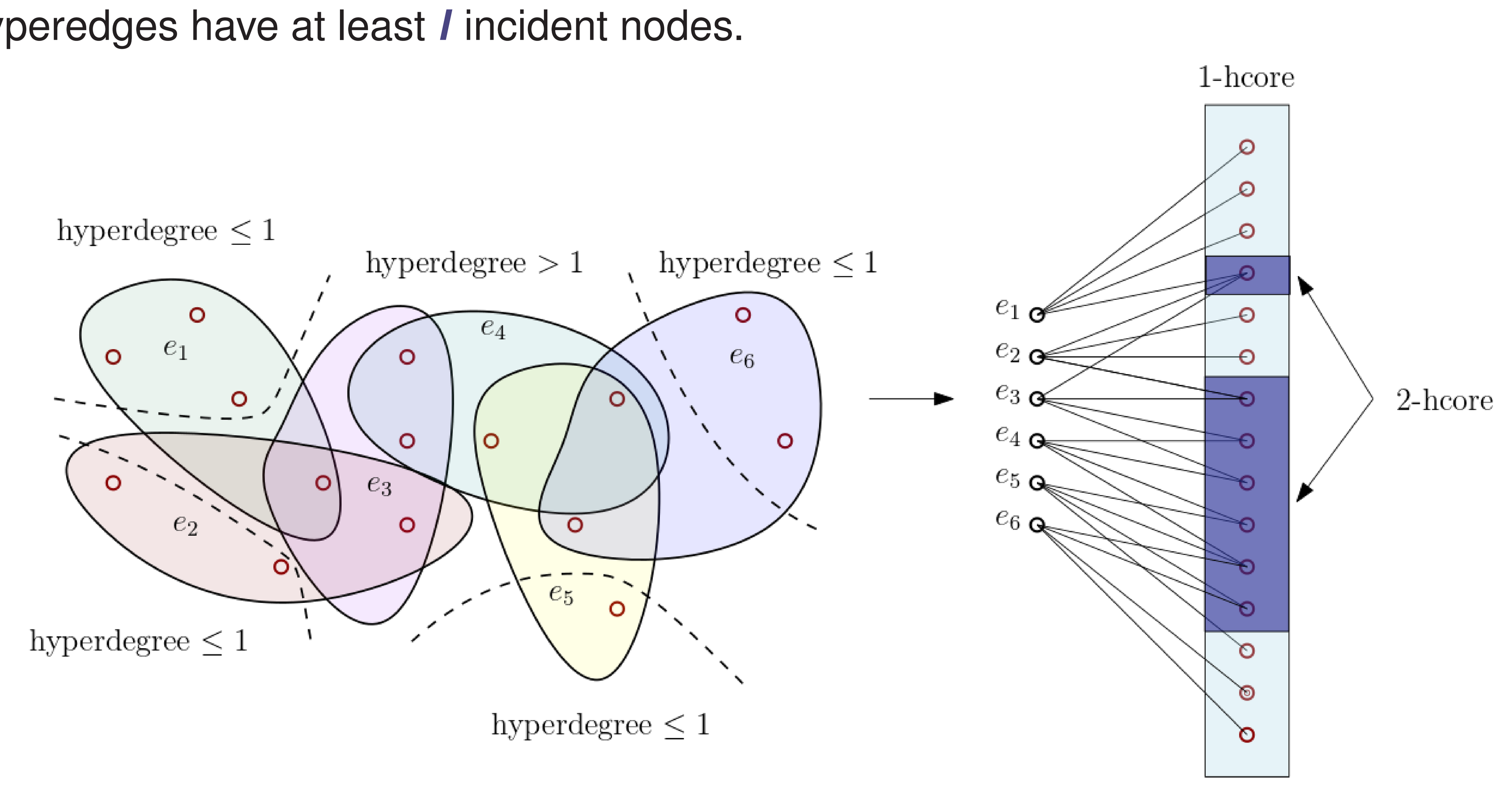


Bipartite graph as the incidence graph of a hypergraph

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

Hypercore (Hcore) Decomposition

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



Hcore number of the hypergraph's nodes

Hcore-init: Weight Initialization

METHOD: The graph-based initialization method consists of:

1. Pretraining of NN for x epochs.
2. Construction of weighted graph structure of NN architecture.
3. Hypercore decomposition of the contstructed graph.
4. 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 $w_{i,j} \geq 0$, $M = \frac{c_j^+}{\sum_{1 \leq k \leq n_2} c_k^+}$,
- else $M = \frac{c_j^-}{\sum_{1 \leq k \leq n_2} c_k^-}$.

Hence $w_{i,j}$ follow a $\mathcal{N}(M, \frac{2}{n_2^l})$ which variance is from the He initilization method.

CNN: For a given filter $W \in \mathbb{R}^{H \times H}$ its values are re-initialized with the following method:

- we define m for a given filter W as $m(W^+) = \frac{1}{H^2}\sum_j c_j^+$ and $m(W^-) = \frac{1}{H^2}\sum_j c_j^-$, if $m(W^+) - m(W^-) > 0$ then $M = m(W^+)$
- else $M = -m(W^-)$.

Hence the general formula for m is given by:

$$M = \mathrm{sign}(\mathrm{argmax}(m(W^+), m(W^-)))\max(m(W^+), m(W^-))$$

where $\mathrm{sign}(W^+) = 1$ and $\mathrm{sign}(W^-) = -1$.

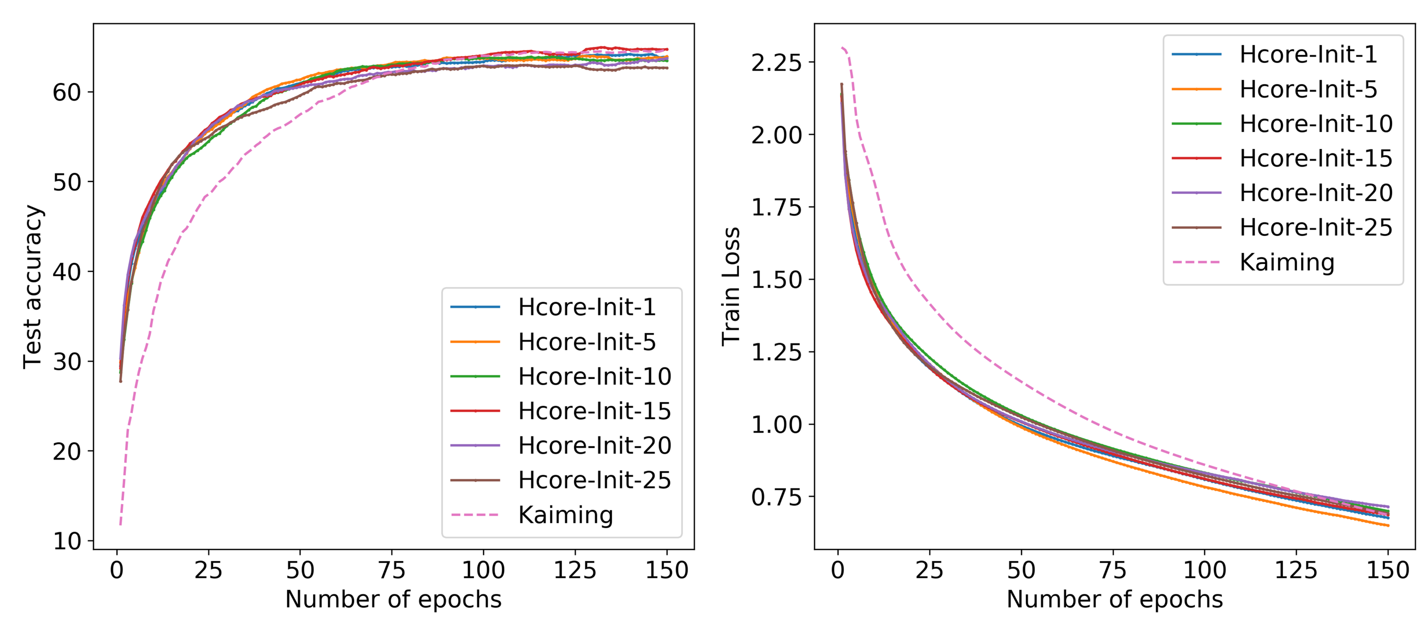
PROPOSITION:
Let X_1 and X_2 two centered i.i.d. random variables with symmetric distribution. We define $X^+ = \max\{X_1, 0\}$, $X^- = \max\{X_2, 0\}$, and a real valued measurable function $f : \mathbb{R}_+ \to \mathbb{R}$ such that $\mathbb{E}[|f(X^+)|] < \infty$ and $\mathbb{E}[|f(X^-)|] < \infty$.
Then:

- X^+, X^- are positive i.i.d. random variables.
- The random variable:

$$M = \mathrm{sign}(\mathrm{argmax}(f(X^+), f(X^-)))\max(f(X^+), f(X^-))$$

with $\mathrm{sign}(f(X^{\pm})) = \pm 1$, is centered, i.e. $\mathbb{E}[M] = 0$.

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.

	CIFAR-10	CIFAR-100	MNIST
Kaiming He	64.62	32.56	98, 71
Hcore-Init*	65.22	33.48	98.91
Hcore-Init-1	64.91	32.87	98.59
Hcore-Init-5	64.41	32.96	98.70
Hcore-Init-10	65.22	33.41	98.81
Hcore-Init-15	64.94	33.45	98.64
Hcore-Init-20	65.05	33.39	98.87
Hcore-Init-25	64.72	33.48	98.91

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

This work was supported by ANR project DEMOGRAPH (ANR-16-CE40-0028) and ESIGMA (ANR-17-CE23-0010)

stratis.limnios@polytechnique.edu