Deep Learning & Quantum Entanglement:
Fundamental Connections with Implications to Network Design

Based on ***arxiv: 1704:01552 (ICLR 2018)***
by Y. Levine, et al.

Yasaman Bahri
hep-ai
Aug 7 2018

Motivation

Inductive bias: assumptions made about the class of target functions

- We should incorporate our ***priors*** regarding desired task

- Full understanding of the inductive bias of existing networks (and the networks we want to design) is lacking.

For instance: CNN architecture has many design elements currently made ~ heuristically: number of layers, **distribution of channels (this paper)**, pooling pattern, convolution kernel size and stride, …

- Contrast with e.g. “*expressive efficiency.*”

- (This talk is only about representation: no optimization.)

Tensor Preliminaries

Terminology:

- Each index of tensor is a *mode,* order of a tensor = number of modes

  Tensor $\mathcal{A}$ with elements $\mathcal{A}_{d_1 d_2 \ldots d_N}$, $d_i \in [M_i] := \{1, ..., M_i\}$, has order N and $\in \mathbb{R}^{M_1 \times \ldots \times M_N}$.

- Matricization of a tensor with respect to a partition:

  Let $\mathcal{A}$ be a tensor of order N with dimensions M_i in each mode $i \in [N]$. The matricization of $\mathcal{A}$ *w.r.t. the partition* (I, J) is written $[[\mathcal{A}]]_{I,J}$ and has shape $(\prod_{t=1}^{|I|} M_{i_t}) \times (\prod_{t=1}^{|J|} M_{j_t})$.

- A rank-1 tensor is the tensor product of vectors:

$$\mathcal{A}^{\text{rank-1}} = \boldsymbol{v}^{(1)} \otimes \ldots \otimes \boldsymbol{v}^{(N)} \Rightarrow \mathcal{A}^{\text{rank-1}}_{d_1 \ldots d_N} = \prod_{j=1}^{N} v^{(j)}_{d_j}.$$

Model: Convolutional Arithmetic Circuits (ConvAC)

A number of works by (overlapping) authors on *convolutional arithmetic circuits.*

- View them as “representative of the class of convolutional NNs.”

- Usual CNNs have pointwise nonlinearities following convolution and max or average pooling.

- ConvACs have linear activations and product pooling (the nonlinear part).
    - Because amenable to theoretical analysis.
    - (Even if different …. get testable predictions? → verify empirically.)

ConvAC Computation

Input $X = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ with $\boldsymbol{x}_i \in \mathbb{R}^s$.

First layer: representation layer

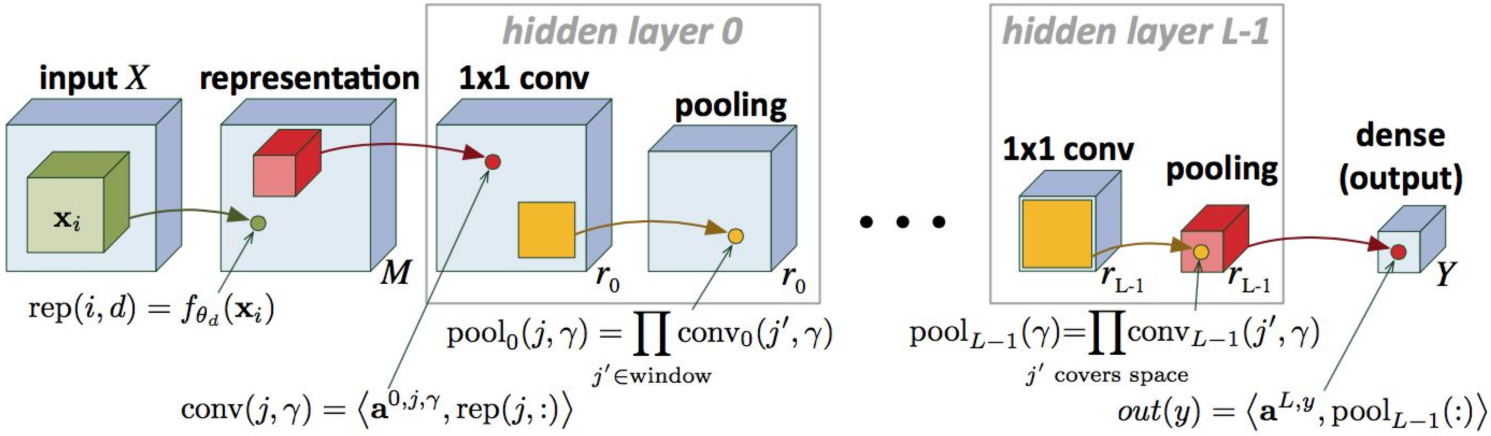
- M representation functions $f_{\theta_1}, ..., f_{\theta_M} : \mathbb{R}^s \to \mathbb{R}$ applied to each local patch $\boldsymbol{x}_i$.
- Example: $f_{\theta_d}(\boldsymbol{x}) = \sigma(\boldsymbol{w}_d^T \boldsymbol{x} + b_d)$ with $\theta_d = (\boldsymbol{w}_d, b_d)$

Following layers $\ell = 0, ..., L-1$:

- Each begins with a 1×1 conv, with $r_{\ell-1}$ input channels and r_ℓ output channels.
- Followed by spatial (same channel) pooling that takes products over non-overlapping windows. Final pooling will be global, over all remaining dimensions $\to r_{L-1}$ dim output vector.

Final dense linear layer: $r_{L-1} \to Y$ dimensional output for Y classes

Model: ConvAC



Can be written in the following form:

$$\mathbf{h}_y(\mathbf{x}_1,\ldots,\mathbf{x}_N) = \sum_{d_1,\ldots,d_N=1}^{M} \mathcal{A}^y_{d_1\ldots d_N} \prod_{j=1}^{N} f_{\theta_{d_j}}(\mathbf{x}_j)$$

$$\mathbf{h}_y(\mathbf{x}_1,\ldots,\mathbf{x}_N) = \sum_{d_1,\ldots,d_N=1}^{M} \mathcal{A}^y_{d_1\ldots d_N} \mathcal{A}^{(\text{rank } 1)}_{d_1\ldots d_N}(\mathbf{x}_1,\ldots,\mathbf{x}_N)$$

(Will return to the decomposition of coefficients tensor)

Interpret ~ Quantum Wavefunction

General quantum state:

$$|\psi\rangle = \sum_{d_1 \ldots d_N = 1}^{M} \mathcal{A}_{d_1 \ldots d_N} |\psi_{d_1}\rangle \otimes \cdots \otimes |\psi_{d_N}\rangle$$

Consider the following product state:

$$|\psi^{\text{ ps}}\rangle = |\phi_1\rangle \otimes \cdots \otimes |\phi_N\rangle \qquad |\phi_j\rangle = \sum_{d_j=1}^{M} v_{d_j}^{(j)} |\psi_{d_j}\rangle \qquad v_d^{(j)} = \langle \psi_d | \phi_j \rangle = f_{\theta_d}(\mathbf{x}_j)$$

Then: $$|\psi^{\text{ ps}}\rangle = \sum_{d_1 \ldots d_N = 1}^{M} \mathcal{A}^{\text{ps}}_{d_1 \ldots d_N} |\psi_{d_1}\rangle \otimes \cdots \otimes |\psi_{d_N}\rangle$$

(With rank-1 coefficient tensor)

$$\mathcal{A}^{\text{ps}}_{d_1 \ldots d_N} = \prod_{j=1}^{N} v_{d_j}^{(j)}$$

$$\langle \psi^{\text{ ps}} | \psi \rangle = \sum_{d_1 \ldots d_N = 1}^{M} \mathcal{A}_{d_1 \ldots d_N} \prod_{j=1}^{N} f_{\theta_{d_j}}(\mathbf{x}_j) = \sum_{d_1 \ldots d_N = 1}^{M} \mathcal{A}_{d_1 \ldots d_N} \mathcal{A}^{\text{ps}}_{d_1 \ldots d_N}(\mathbf{x}_1, \ldots, \mathbf{x}_N)$$

What we wanted: function computed by ConvAC

Why?

- "Entanglement measures as natural quantifiers of dependencies"

- In this domain, have a better understanding of how representation is tied to structure (of quantum states)

$$|\psi\rangle = \sum_{\alpha=1}^{\dim(\mathcal{H}^A)} \sum_{\beta=1}^{\dim(\mathcal{H}^B)} (\llbracket \mathcal{A} \rrbracket_{A,B})_{\alpha,\beta} \left|\psi_\alpha^A\right\rangle \otimes \left|\psi_\beta^B\right\rangle$$

e.g. things like Schmidt number (rank of matricization) or entanglement entropy...

Specifically, we will import some (recently proven) results from the quantum side to inform a particular design choice: distribution of channel sizes in the network.

Aside: separation rank

Measure distance from separability via notion of *separation rank*

For a function $h : (\mathbb{R}^s)^N \to \mathbb{R}$, the *separation rank w.r.t the partition (I,J)* is the minimum R such that

$$h(\boldsymbol{x}_1, ..., \boldsymbol{x}_N) = \sum_{\nu=1}^{R} g_\nu(\boldsymbol{x}_{i_1}, ..., \boldsymbol{x}_{i_{|I|}}) g'_\nu(\boldsymbol{x}_{j_1}, ..., \boldsymbol{x}_{j_{|J|}})$$

Claim (see [1]):

The separation rank $sep(h_y; I, J)$ of the function computed by the ConvAC is equal to the (matrix) rank of $[[\mathcal{A}^y]]_{I,J}$ (i.e. the Schmidt number).

Also ([1]):

Finally, $sep(h_y; I, J)$ can be related to the L^2 distance of h from the set of separable functions w.r.t (I, J). Let

$$D(h; I, J) := \frac{1}{||h||} \cdot \inf_{g,g' \in L^2} ||h(\boldsymbol{x}_1, ...\boldsymbol{x}_N) - g(\boldsymbol{x}_{i_1}, ..., \boldsymbol{x}_{i_{|I|}}) g'(\boldsymbol{x}_{j_1}, ..., \boldsymbol{x}_{j_{|J|}})||$$

Then $D(h; I, J) \leq \sqrt{1 - \frac{1}{sep(h;I,J)}}$.

[1]. Cohen & Shashua. arxiv 1605:06743, ICLR 2017.

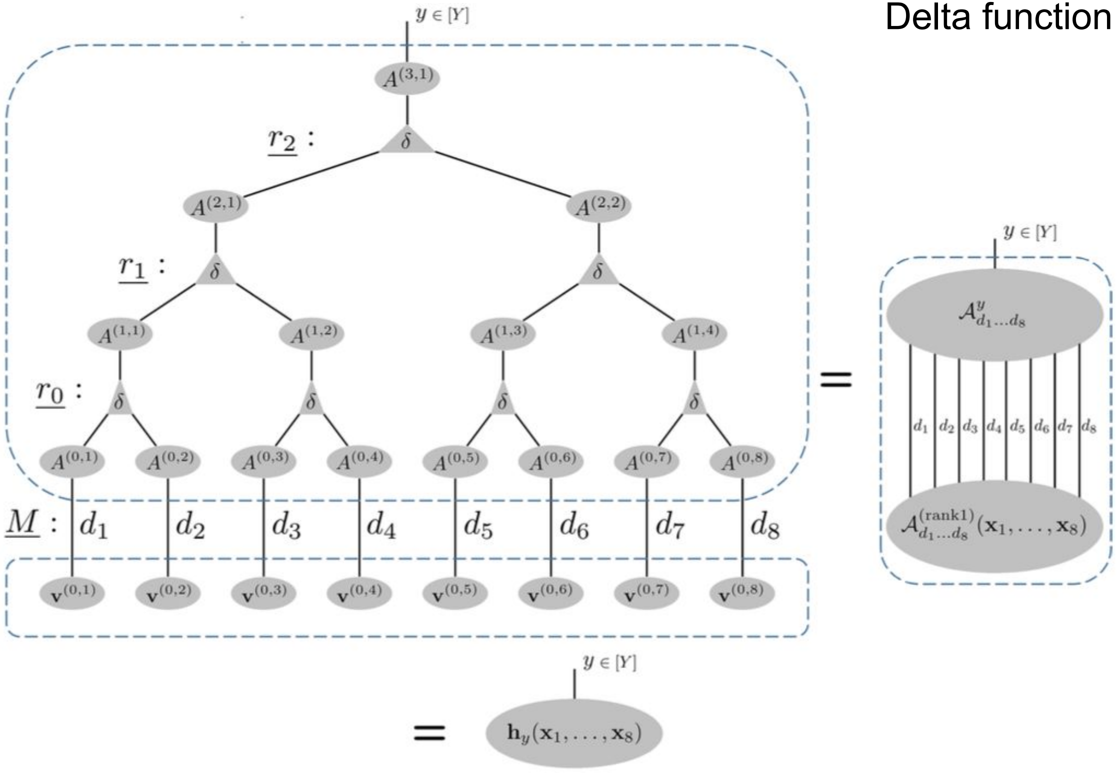
Recast ConvAC as a Tensor Network

(Mainly semantic differences)

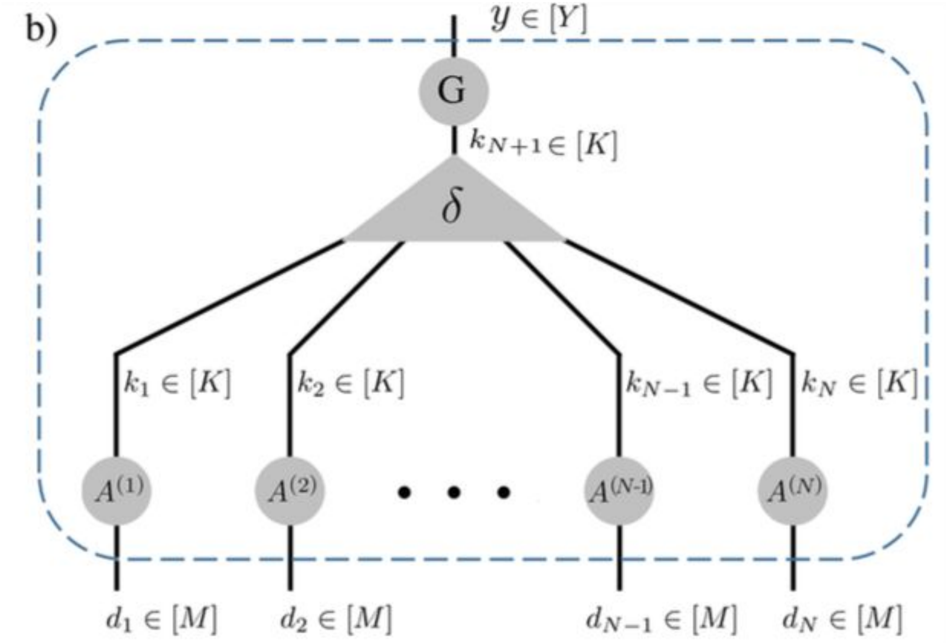
Delta function

Example for $N = 8$. Each matrix $A^{(\ell,j)} \in \mathbb{R}^{r_\ell \times r_{\ell-1}}$ (with $r_{-1} := M$) holds the conv weight vector $\boldsymbol{a}^{\ell,j,\gamma} \in \mathbb{R}^{r_{l-1}}$, $\gamma \in [r_\ell]$, in its γ-th row.

Same channel pooling because of δ, which is $\in \mathbb{R}^{r_{\ell-1} \times r_{\ell-1} \times r_{\ell-1}}$.



Example of a Shallow ConvAC → TN



(Also known as a CP decomposition)

$$\mathcal{A}^{y}_{d_1 \ldots d_N} = \sum_{k_1,\ldots,k_{N+1}=1}^{K} \delta_{k_1 \ldots k_{N+1}} A^{(1)}_{k_1 d_1} \ldots A^{(N)}_{k_N d_N} G_{y k_{N+1}}$$

ConvAC/TN as a Graph

Important elements now: connectivity among individual tensors and the *bond dimension* on each edge

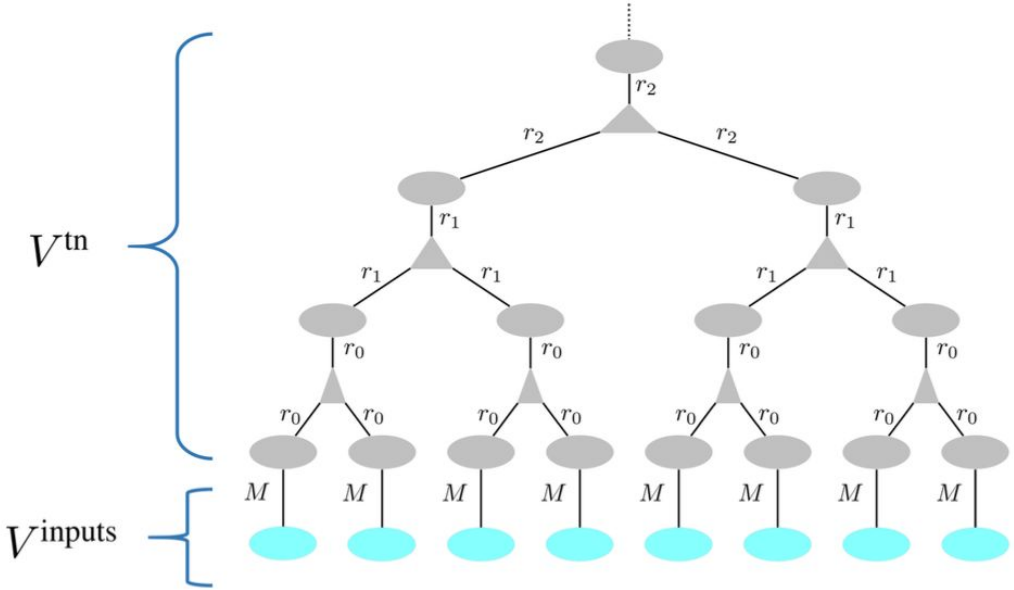
- Bond dimension = number of channels (feature maps)

$$G(V, E)$$

$$V = V^{\mathrm{tn}} \dot{\cup} V^{\mathrm{inputs}}$$

$$c : E \rightarrow \mathbb{N}$$

We will analyze the cuts between input nodes (specifically, a partition into sets A, B).



A Definition

An **edge-cut set** w.r.t the partition $V^A \dot\cup V^B = V^{inputs}$ is a set of edges C s.t. $\exists$ a partition $\tilde{V}^A \dot\cup \tilde{V}^B = V$ with $V^A \subset \tilde{V}^A, V^B \subset \tilde{V}^B$, and $C = \{(u,v) \in E : u \in \tilde{V}^A, v \in \tilde{V}^B\}$.

Let $C = \{e_1, ..., e_{|C|}\}$. Then the multiplicative cut weight is = product of all bond dimensions along the cut:

$$W_C = \prod_{i=1}^{|C|} c(e_i)$$

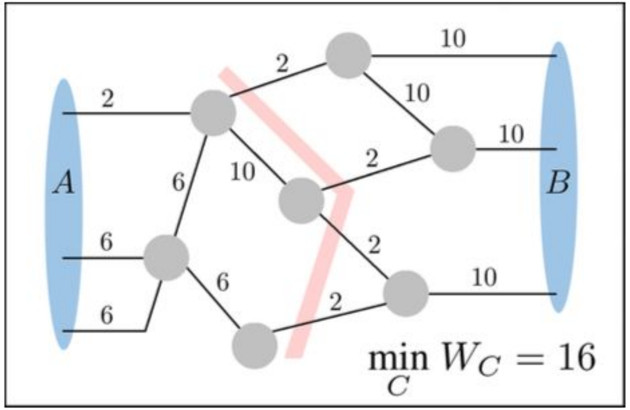
Bounds on Entanglement

Specifically, the Schmidt entanglement measure (importing a known bound).

Claim: Let (A, B) be a partition of $[N]$ and $[[\mathcal{A}^y]]_{A,B}$ be the matricization w.r.t (A, B) of the convolutional weights tensor $\mathcal{A}^y$ with pooling window of size 2. Then the rank of the matricization $[[\mathcal{A}^y]]_{A,B}$ obeys:

$$[[\mathcal{A}^y]]_{A,B} \leq \min{}_C W_C$$

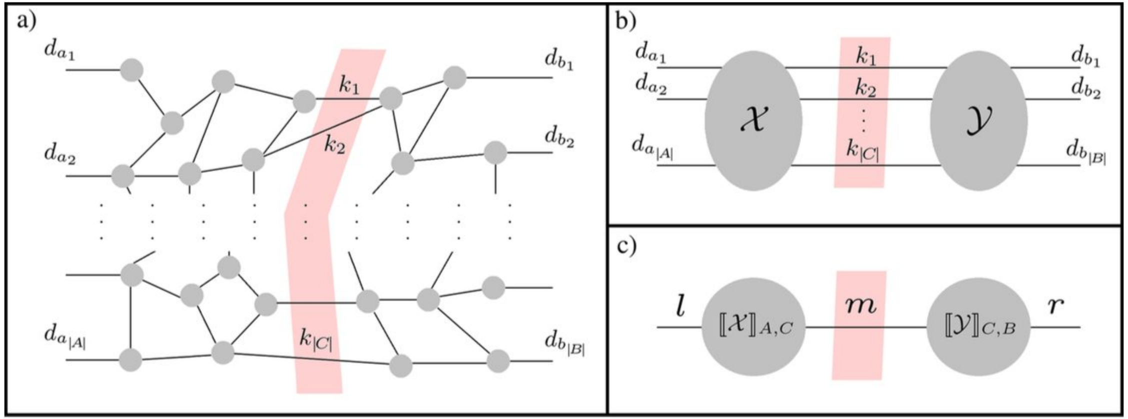
Compare to classical min-cut/max-flow in a graph.



See e.g. S. Cui, et al. arxiv 1508.04644.

Quantum Min-Cut/Max-Flow

Why? Consider the following bipartition and ways of contracting the network:



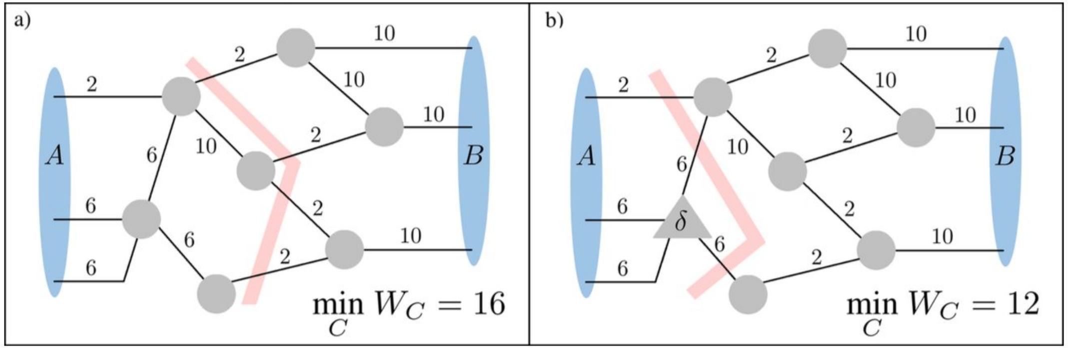
$$([\![\mathcal{A}]\!]_{A,B})_{lr} = \sum_{m=1}^{W_C} ([\![\mathcal{X}]\!]_{A,C})_{lm} ([\![\mathcal{Y}]\!]_{C,B})_{mr}$$

Leave the network with an inner (composite) index left uncontracted, so as to get a product of two matrices. Then because rank is limited by the inner dimension, we have an inequality.

Quantum Min-Cut/Max-Flow

General pooling case (window size > 2): need to adjust upper bound.

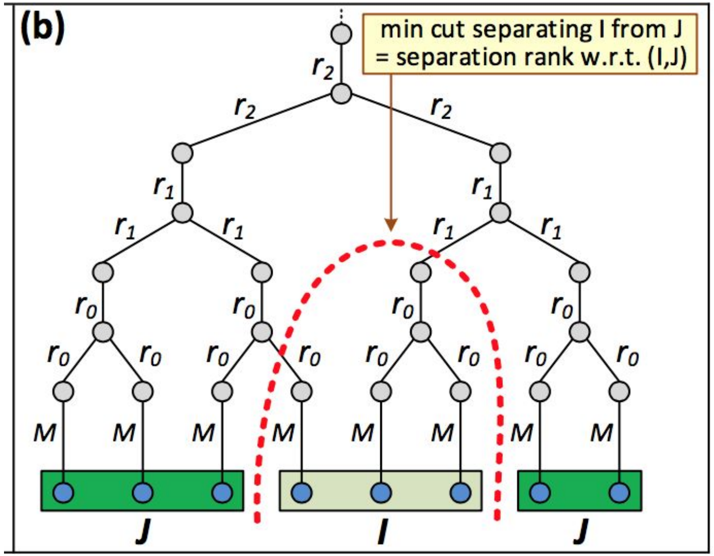
(In this case, delta tensor forces indices to be the same -- reduced dimensionality.)



Only count repeated edges from the Delta tensor once in the multiplicative cut weight.

(Note implication for a shallow network, which has global pooling.)

Bounds on Entanglement

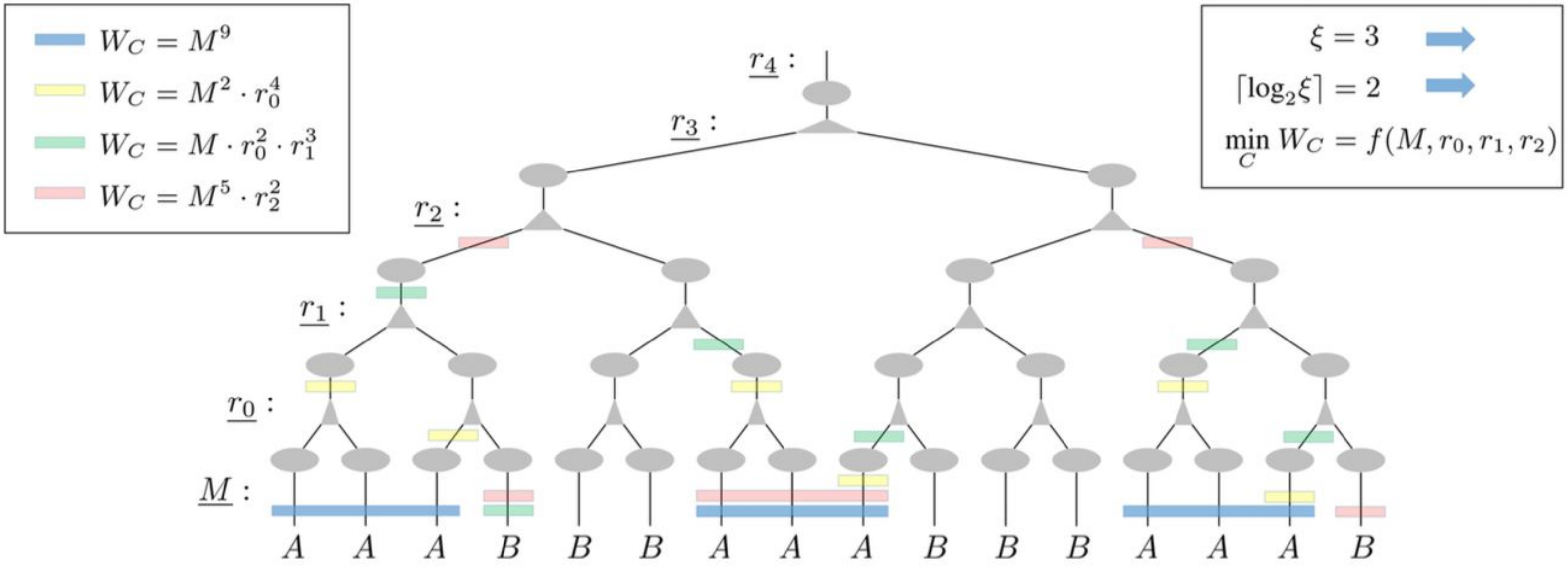


See paper for a lower bound on rank of matricization (Schmidt rank).

For bounds to be useful, would like them to be tight.

Their simulations (Gaussian weights and channel # drawn from some set) show negligible deviations from the upper bound (min cut value).

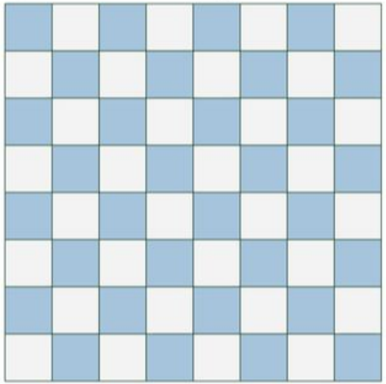
Example



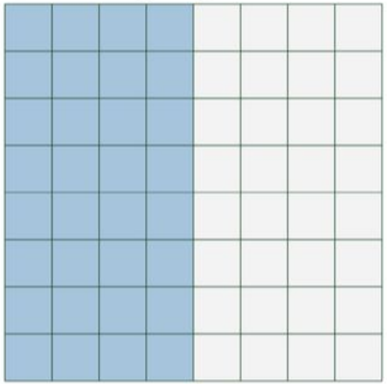
For shorter ranged correlations, channel distribution in lower layers matters.

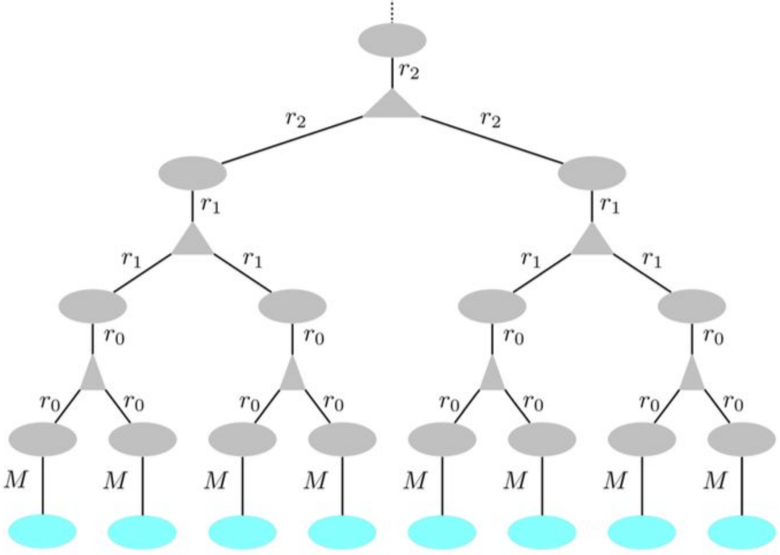
Example

a) Interleaved partition



b) Left-right partition

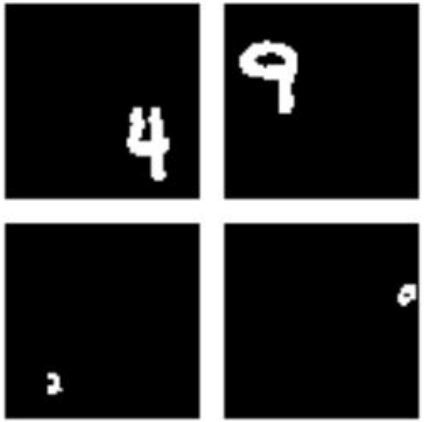




$$W_C^{\text{interleaved}} = \min(r_0^{N/4}, M^{N/2})$$

$$W_C^{\text{left-right}} = \min(r_{L-1}, r_{L-2}, \ldots, r_l^{2^{(L-2-l)}}, \ldots, r_0^{N/4}, M^{N/2}) \qquad \left(L = \log_2(N)\right)$$

An Experiment



Task:
64 x 64 binary MNIST in random positions.
“Local” task: digits resized to 8 x 8 (within 64 x 64).
“Global” task: resized to 32 x 32.

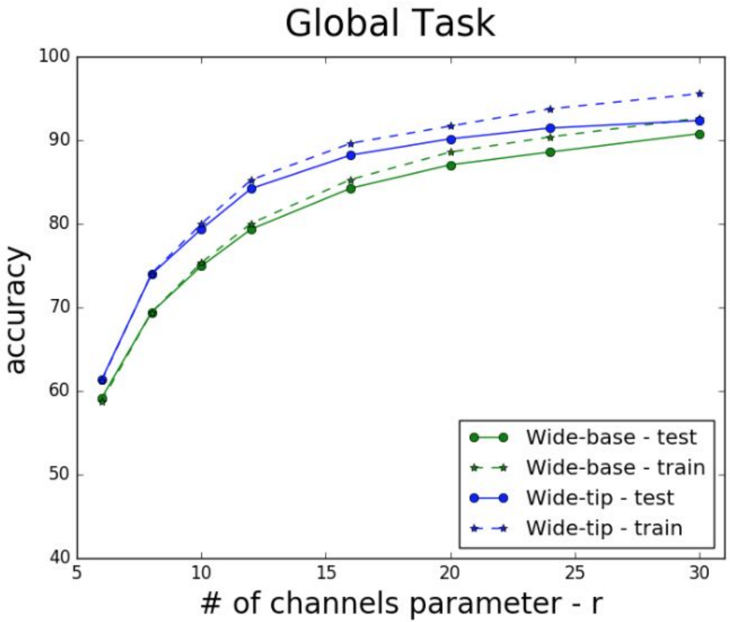
Architecture:
Two networks, only difference between them is channel ordering scheme.

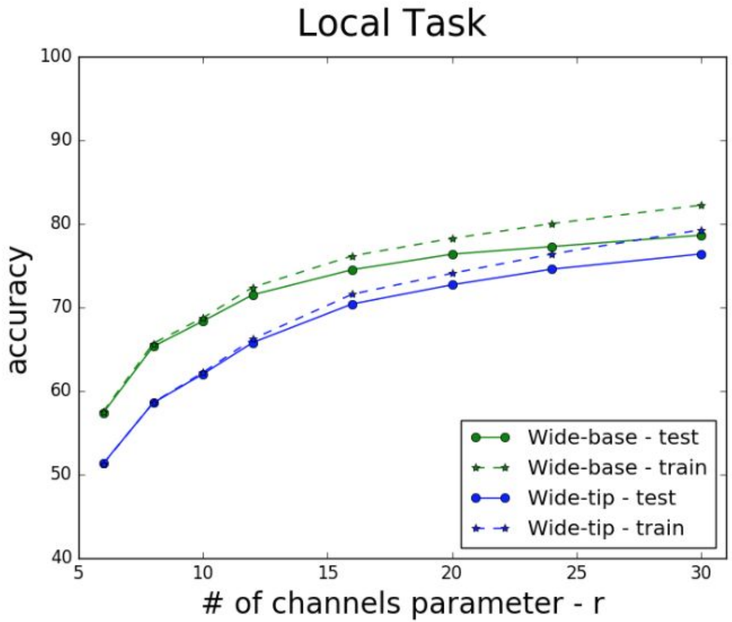
First (representation) layer: 3 x 3 shared conv (with stride 1)
Followed by 6 hidden layers, each with 1 x 1 shared conv → ReLU → 2 x 2 max pooling (with stride 2).
Final layer Y = 10 classes.

An Experiment

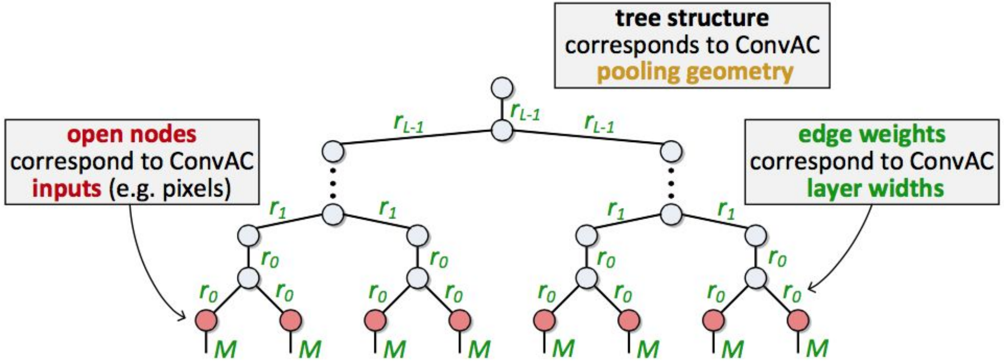
Ordering scheme (same total number of parameters $= 31r^2 + 50r$):

- Wide-base: $[10; 4r; 4r; 2r; 2r; r; r; 10]$
- Wide-tip: $[10; r; r; 2r; 2r; 4r; 4r; 10]$





Conclusion



- Graph theoretic analysis of architectures
- Theory vs. practice w.r.t. distribution of channel sizes in real CNNs
- Even if theoretical model not exact: predictive and accurate?

See also Y. Levine, et al. arxiv 1803.09780.

The End