A Johnson–Lindenstrauss Framework for Randomly Initialized CNNs

Ido Nachum Jan Hązła Michael Gastpar Anatoly Khina

ICLR 2022

Setting

$\langle x, y \rangle$	The scalar *inner product* between vectors x and y.[1]
$\frac{\langle x,y \rangle}{\|x\|\|y\|}$	The *cosine similarity* (or simply *similarity)* between vectors x and y.

The Johnson–Lindenstrauss Lemma
A high-dimensional random projection W satisfies

$$\langle x, y \rangle \approx \langle W \cdot x, W \cdot y \rangle$$

with high probability.

[1] We mean here a vector in a wider sense: x and y may be matrices and tensors (of the same dimensions). In this situation, the standard inner product is equal to the vectorization thereof: $\langle x, y \rangle = \langle \mathrm{vec}(x), \mathrm{vec}(y) \rangle$.

Motivating Question

How does the geometry ($\rho := \langle x, y \rangle$) change after a non-linear FNN layer?

Cho–Saul (2009), Giryes–Sapiro–Bronstein (2016), Daniely–Frostig–Singer (2016):

$$\langle \mathrm{ReLU}(W \cdot x), \mathrm{ReLU}(W \cdot y) \rangle \approx \frac{\sqrt{1-\rho^2} + \left(\pi - \cos^{-1}(\rho)\right)\rho}{\pi}$$

Motivating Question

How does the geometry ($\rho := \langle x, y \rangle$) change after a non-linear FNN layer?

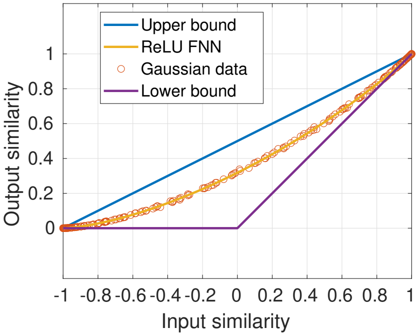
Cho–Saul (2009), Giryes–Sapiro–Bronstein (2016), Daniely–Frostig–Singer (2016):

$$\langle \mathrm{ReLU}(W \cdot x), \mathrm{ReLU}(W \cdot y) \rangle \approx \frac{\sqrt{1-\rho^2} + \left(\pi - \cos^{-1}(\rho)\right)\rho}{\pi}$$

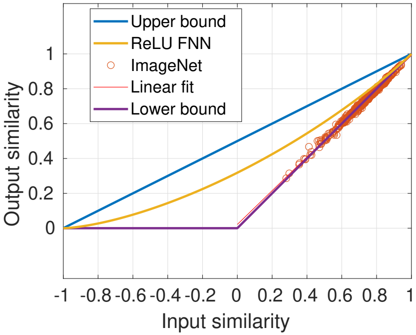
What about a randomly initialized convolutional neural network?

$$\langle \mathrm{ReLU}(W * x), \mathrm{ReLU}(W * y) \rangle = \ ???$$

Main Results

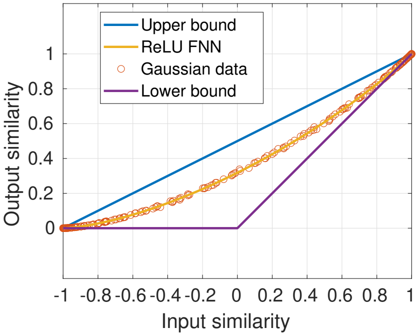


(a) Gaussian images, filter size 11×11

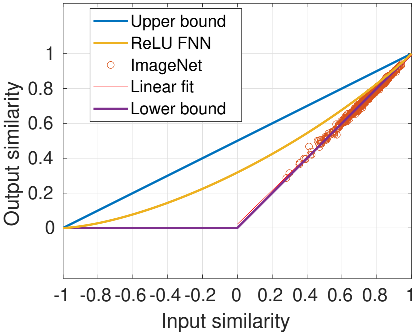


(b) ImageNet, filter size $11 \times 11 \times 3$

Main Results



(a) Gaussian images, filter size 11×11

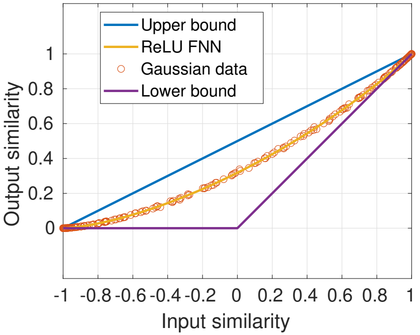


(b) ImageNet, filter size $11 \times 11 \times 3$

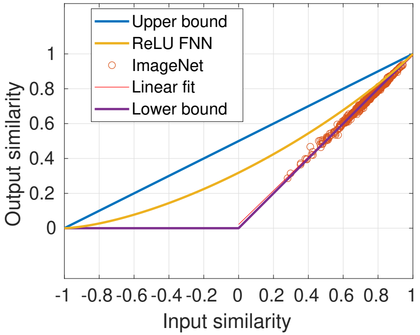
Lemma 2: w.h.p., for linear CNNs we have a typical Johnson–Lindenstrauss type result

$$\rho_{\text{out}} \approx \rho_{\text{in}}$$

Main Results



(a) Gaussian images, filter size 11×11

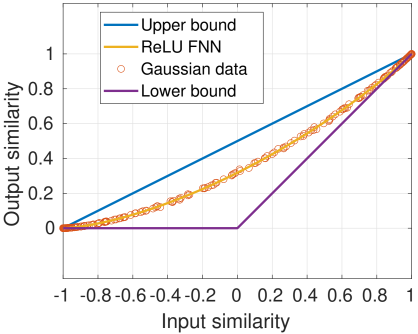


(b) ImageNet, filter size $11 \times 11 \times 3$

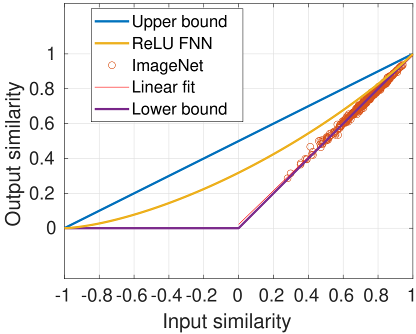
Theorem 3: w.h.p., for ReLU CNNs we have the following tight bounds

$$\max\{\rho_{\text{in}}, 0\} \lesssim \rho_{\text{out}} \lesssim \frac{1 + \rho_{\text{in}}}{2}$$

Main Results



(a) Gaussian images, filter size 11×11

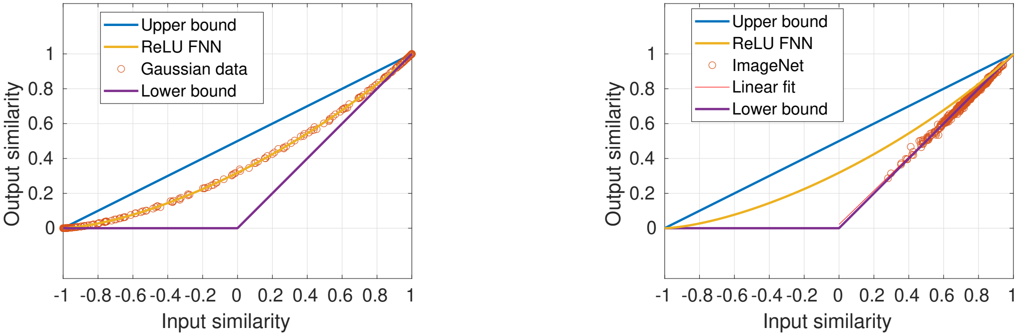


(b) ImageNet, filter size $11 \times 11 \times 3$

Theorem 4: w.h.p., for ReLU CNNs with Gaussian inputs

$$\rho_{\text{out}} \approx \frac{\sqrt{1-\rho_{\text{in}}^2} + \left(\pi - \cos^{-1}(\rho_{\text{in}})\right)\rho_{\text{in}}}{\pi}$$

Main Results



(a) Gaussian images, filter size 11×11

(b) ImageNet, filter size $11 \times 11 \times 3$

Theorem 5: w.h.p., for ReLU CNNs and a model for natural images

$$\rho_{\text{out}} \approx \rho_{\text{in}}$$