A Johnson–Lindenstrauss Framework for Randomly Initialized CNNs

Ido Nachum[1] Jan Hązła[1] Michael Gastpar[1] Anatoly Khina[2]

[1]École Polytechnique Fédérale de Lausanne [2]Tel Aviv University

Abstract

How does the geometric representation of a dataset change after the application of each randomly initialized layer of a neural network? The celebrated Johnson–Lindenstrauss lemma answers this question for linear fully-connected neural networks (FNNs), stating that the geometry is essentially preserved. For FNNs with the ReLU activation, the angle between two inputs contracts according to a known mapping. The question for non-linear convolutional neural networks (CNNs) becomes much more intricate. To answer this question, we introduce a geometric framework. For linear CNNs, we show that the Johnson–Lindenstrauss lemma continues to hold, namely, that the angle between two inputs is preserved. For CNNs with ReLU activation, on the other hand, the behavior is richer: The angle between the outputs contracts, where the level of contraction depends on the nature of the inputs. In particular, after one layer, the geometry of natural images is essentially preserved, whereas for Gaussian correlated inputs, CNNs exhibit the same contracting behavior as FNNs with ReLU activation.

Setting

How does the geometric representation of a dataset change after the application of each randomly initialized layer of a neural network?

We study two quantities:

$\langle x, y\rangle$ The scalar *inner product* between vectors x and y.

$\frac{\langle x,y\rangle}{\|x\|\|y\|}$ The *cosine similarity* (or simply *similarity*) between vectors x and y. This is a proxy for the angle between x and y.

A linear fully-connected network does not change the geometry of the dataset:

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.

Motivating Question

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

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

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

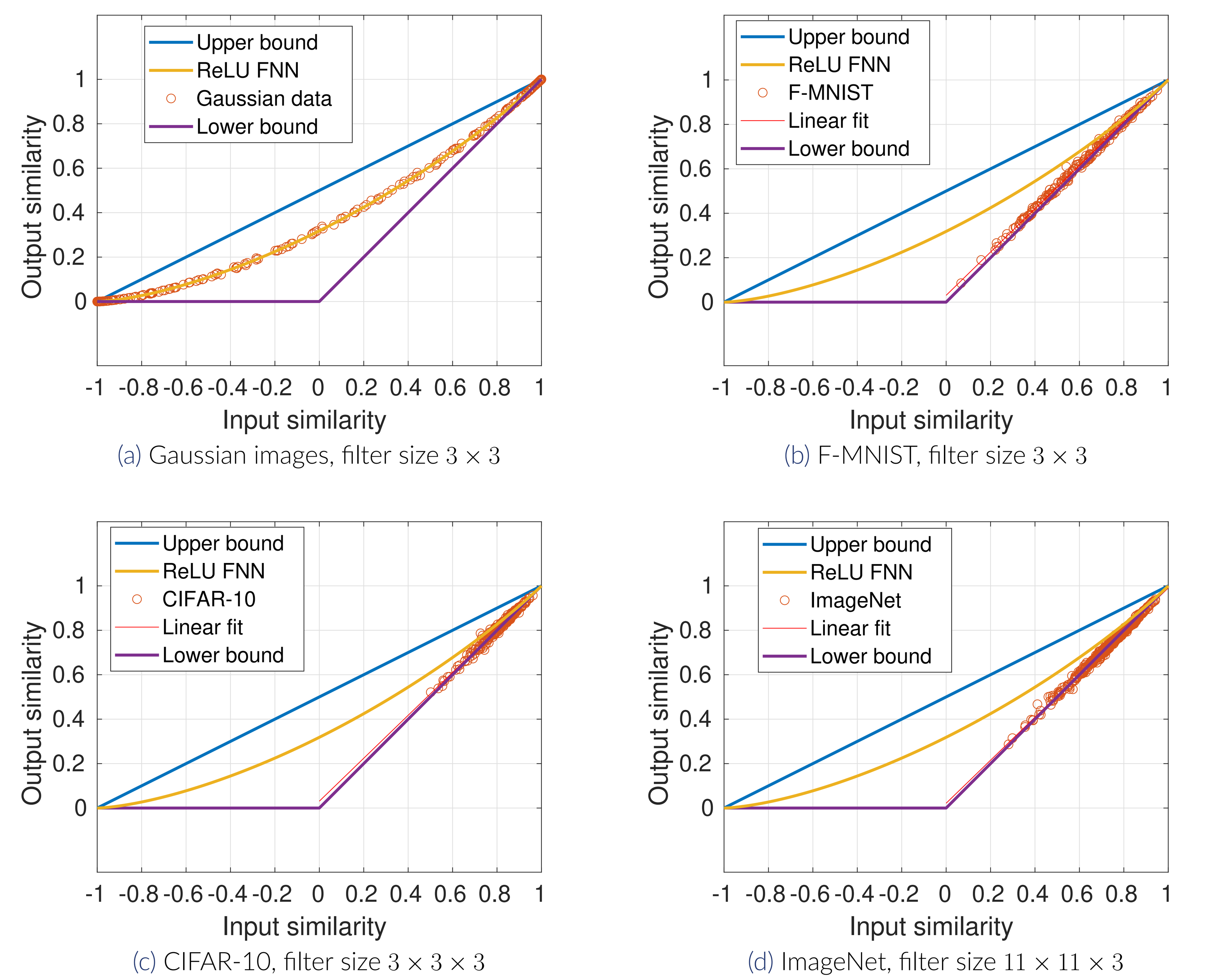
Note: the output similarity is only a function of the input similarity.

What about a randomly initialized convolutional neural network?

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

The Answer

The relation between input and output similarity for a randomly initialized ReLU convolutional neural network is more intricate:



(a) Gaussian images, filter size 3×3

(b) F-MNIST, filter size 3×3

(c) CIFAR-10, filter size $3 \times 3 \times 3$

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

Figure 1. Input and output cosine similarities of a single randomly initialized convolutional layer with 100 filters. Each red circle in the figures represents a random pair of images chosen from the corresponding dataset. 200 pairs were sampled in each figure.

Note: unlike a ReLU FNN, for a ReLU CNN, the output similarity is not only a function of the input inner product.
For natural images, a randomly initialized ReLU CNN behaves as a linear FNN.
For Gaussian data, a randomly initialized ReLU CNN behaves as a ReLU FNN.

Main Results

Lemma 2

With high probability, for linear CNNs we have a typical Johnson–Lindenstrauss type result

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

Theorem 3

With high probability, for ReLU CNNs we have the following tight bounds

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

Theorem 4

With high probability, for ReLU CNNs with Gaussian inputs

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

Theorem 5

With high probability, for ReLU CNNs with filters of dimensions $(2r+1)\times(2r+1)$ and binary images x and y

$$|\rho_{\text{out}} - \rho_{\text{in}}| \le \frac{\partial_r(x,y)}{\|x\|\,\|y\|}$$

Shared r-Boundary $\partial_r(x, y)$

Let $x, y \subset \mathbb{Z}_n \times \mathbb{Z}_n$ and let $r \in [n]$. Then, the r-boundary of the intersection between x and y, denoted by $\partial_r(x, y)$, is defined as the set of all pixels $(i, j) \in \mathbb{Z}_n \times \mathbb{Z}_n$ such that

$$\exists a_1, b_1, c_1, d_1 \in \{-r, \ldots, r\} : (i + a_1, j + b_1) \in x \wedge (i + c_1, j + d_1) \in y \quad (1a)$$
$$\exists a_2, b_2, c_2, d_2 \in \{-r, \ldots, r\} : (i + a_2, j + b_2) \notin x \vee (i + c_2, j + d_2) \notin y \quad (1b)$$

where the addition in equation 1a and equation 1b is over $\mathbb{Z}_n$.

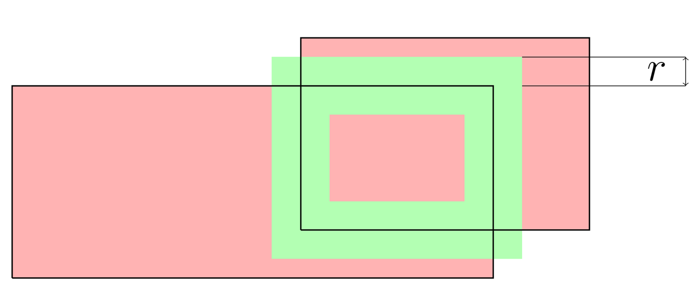


Figure 2. Illustration of the shared r-boundary for two axis-aligned rectangles. The boundary is marked in green.