Efficiency of Shallow Cascades for Improving Deep Learning AI Systems

1st Ivan Y. Tyukin[1,2,3]
[1] *Department of Mathematics*
University of Leicester
[2] *Lobachevsky University*
[3] *St-Petersburg State Electrotechnical University*
[1] Leicester, United Kingdom
I.Tyukin@le.ac.uk

2nd Alexander N Gorban
University of Leicester
and Lobachevsky University
Leicester, UK, and Nizhni Novgorod, Russia
a.n.gorban@le.ac.uk

3rd Danil V. Prokhorov
Toyota Research Institute
Toyota Motor Corporation
Ann-Arbor, United States

4th Stephen Green
Department of Mathematics
University of Leicester
Leicester, United Kingdom
slg46@le.ac.uk

***Abstract*—This paper presents a technology for simple and non-iterative improvements of Multilayer and Deep Learning neural networks and Artificial Intelligence (AI) systems. The improvements are, in essence, shallow networks constructed on top of the existing Deep Learning architecture. Theoretical foundation of the technology is based on Stochastic Separation Theorems and the ideas of measure concentration. We show that, subject to mild technical assumptions on statistical properties of internal signals in Deep Learning AI, with probability close to one the technology enables instantaneous "learning away" of spurious and systematic errors. The method is illustrated with numerical examples.**

***Index Terms*—Deep Learning, Stochastic Separation Theorems, Linear Separability, Perceptron, Shallow Networks**

I. Introduction

In recent years, Artificial Intelligence (AI) systems have risen dramatically from being the subject of mere academical and focused specialized practical interests to the level of commonly accepted and widely-spread technology. Industrial giants such as Google, Amazon, IBM, Microsoft offer a broad range of AI-based services, including intelligent image and sound processing and recognition.

As a rule of thumb, Deep Learning and related computational technologies [1], [2] are currently perceived as the state-of-the art systems capable of handling large volumes of data and delivering unprecedented accuracy [3] at a reasonable computational costs, albeit after some optimization [4]. Despite these advances, several fundamental challenges hider further progress of the technology.

All Artificial Intelligence systems make mistakes. Mistakes may arise due to uncertainty that is inherently present in empirical data, data misrepresentation, or imprecise or inaccurate training. Conventional approaches aimed at tackling inevitable errors include altering training data and improving design procedures [5], [6], [7], [8]. AI knowledge transfer, transfer learning [9], [10], [11], and privileged learning [12] constitute a viable way to reduce generalization errors and hence improve performance. These approaches, however, invoke extensive training procedures. The latter, whilst eradicating some errors, are inherently prone to new errors by the very virtue of steps involved (e.g. mini-batches, randomized training sets etc).

In this work, we propose an alternative. Instead of trying to solve the issue of inevitable spurious errors arising in iterative trial-end-error re-training of state-of-the art large AI systems with sophisticated Deep Learning architectures, we advocate the technology of non-iterative *shallow correctors*. As a concept, the technology has been presented in [13] (cf. [14]). Here we develop this concept further, extend it to Deep Learning AI systems, and demonstrate viability of the technology both theoretically and numerically. Main building blocks of the technology are simple linear perceptron-type [15] classifiers. In the theoretical basis of this work are the ideas of measure concentration [16], [17], [18], [19], [20], and Stochastic Separation Theorems [21].

We show that, subject to mild assumptions on statistical properties of "internal" signals in Deep Learning systems, shallow cascades of simple linear classifiers are an efficient tool for learning away spurious and systematic errors of Deep Learning systems. These cascades can be used for learning new skills too. Remarkably, construction of the cascades themselves can be achieved in a non-iterative one-shot manner, making the technology particularly efficient for systems that have already been deployed and are in operation.

The paper is organized as follows: Section II contains a formal statement of the problem, Section III presents the main results, in Section IV we relate functionality of the proposed linear classifiers to the quadratic ones, Section V provides

The work is supported by Innovate UK Technology Strategy Board (Knowledge Transfer Partnership grants KTP009890 and KTP010522) and by the Ministry of Education and Science of Russian Federation (Project No. 14.Y26.31.0022).

 978-1-5090-6014-6/18/$31.00

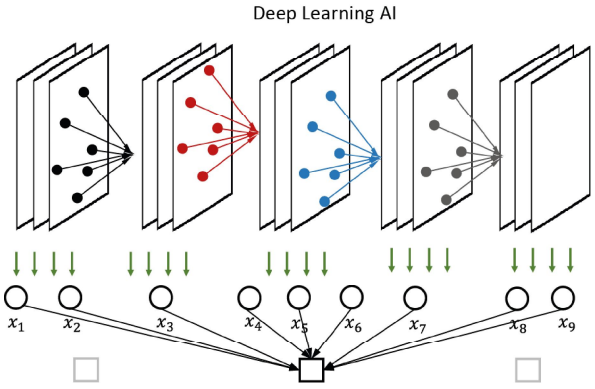

Fig. 1. Shallow cascades for Deep Learning AI.

numerical examples illustrating the concept, and Section VI concludes the paper.

NOTATION

The following notational agreements are used throughout the paper:

- $\mathbb{R}$ denotes the field of real numbers;
- $\mathbb{N}$ is the set of natural numbers;
- $\mathbb{R}^n$ stands for the n-dimensional real space; unless stated otherwise symbol n is reserved to denote dimension of the underlying linear space;
- let $\boldsymbol{x} \in \mathbb{R}^n$, then $\|\boldsymbol{x}\|$ is the Euclidean norm of $\boldsymbol{x}$: $\|\boldsymbol{x}\| = \sqrt{x_1^2 + \cdots + x_n^2}$;
- $B_n(R)$ denotes a n-ball of radius R centered at 0: $B_n(R) = \{\boldsymbol{x} \in \mathbb{R}^n | \ \|\boldsymbol{x}\| \leq R\}$;
- $\mathcal{V}(\Xi)$ is the Lebesgue volume of $\Xi \subset \mathbb{R}^n$;
- if X is a random variable then $E[X]$ is its expected value.

II. PROBLEM STATEMENT

Our main question is: If there exist shallow networks such that

1) they are able to "improve" performance of state-of-the art Deep Learning Networks with guaranteed probability;
2) training of these shallow networks can be accomplished in non-iterative and computationally efficient way?

In order to address this question formally, we shall suppose that an AI system is an operator mapping elements of its input set, $\mathcal{U}$, to the set of outputs, $\mathcal{Q}$. Examples of inputs $\boldsymbol{u} \in \mathcal{U}$ are images, temporal or spatiotemporal signals, and the outputs $\boldsymbol{q} \in \mathcal{Q}$ correspond to labels, classes, or some quantitative characteristics of the inputs. Inputs $\boldsymbol{u}$, outputs $\boldsymbol{q}$, and interval variables $\boldsymbol{z} \in \mathcal{Z}$ of the system represent the system's state. The state itself may not be available for observation but some of its variables or relations may be accessed. In other words, we assume that there is mapping or a process which assigns an element of $\boldsymbol{x} \in \mathbb{R}^n$ to the triple $(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{q})$. A diagram illustrating the setup for a Deep Learning AI system is shown in Fig. 1 Following standard assumptions (see e.g. [22], [23]), we suppose that all $\boldsymbol{x}$ are generated in accordance with some distribution, and the actual measurements $\boldsymbol{x}_i$ that are samples from this distribution. For simplicity, let all such samples be identically and independently distributed (i.i.d.). With regards to the elements $\boldsymbol{x}_i$, the following technical condition is assumed:

Assumption 1: Elements $\boldsymbol{x}_i$ are random i.i.d. vectors drawn from a product measure distribution:

A1) their x_{ij}-th components are independent and bounded random variables X_j: $-1 \leq X_j \leq 1$, $j = 1, \ldots, n$,

A2) $E[X_j] = 0$, and $E[X_j^2] = \sigma_j^2$.

Over a relevant period of time, the AI system generates a finite but large set of measurements $\boldsymbol{x}_i$. This set is assessed by an external supervisor and is partitioned into the union of the sets $\mathcal{M}$ and $\mathcal{Y}$

$$\mathcal{M} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\},$$
$$\mathcal{Y} = \{\boldsymbol{x}_{M+1}, \ldots, \boldsymbol{x}_{M+k}\}.$$

The set $\mathcal{M}$ may contain measurements corresponding to expected operation of the AI, whereas elements $\mathcal{Y}$ constitute singularities. The singularities may be both desired (related e.g. to "important" inputs $\boldsymbol{u}$) and undesired (related e.g. to errors of the AI).

The formal question therefore is: If there exists a shallow network capable of separating the set $\mathcal{Y}$ from $\mathcal{M}$? The answer is provided in the next section.

III. MAIN RESULTS

Let

$$R_0^2 = \sum_{i=1}^{n} \sigma_i^2.$$

Then the following result holds (cf. [21]).

Theorem 1: Let $\boldsymbol{x}_i$ be i.i.d. random points from the product distribution satisfying Assumption 1, $0 < \delta < 1$, $0 < \varepsilon < 1$ and $R_0 > 0$. Then

1) for any i,

$$P\left(1 - \varepsilon \leq \frac{\|\boldsymbol{x}_i\|^2}{R_0^2} \leq 1 + \varepsilon\right) \geq 1 - 2\exp\left(-\frac{2R_0^4\varepsilon^2}{n}\right); \quad (1)$$

2) for any i, j, $i \neq j$,

$$P\left(\left\langle \frac{\boldsymbol{x}_i}{R_0}, \frac{\boldsymbol{x}_j}{R_0} \right\rangle < \delta\right) \geq 1 - \exp\left(-\frac{R_0^4\delta^2}{n}\right); \quad (2)$$

3) for any given $\boldsymbol{y} \in [-1, 1]^n$ and any i

$$P\left(\left\langle \frac{\boldsymbol{x}_i}{R_0}, \frac{\boldsymbol{y}}{R_0} \right\rangle < \delta\right) \geq 1 - \exp\left(-\frac{R_0^4\delta^2}{n}\right). \quad (3)$$

Proof of Theorem 1. The proof follows immediately from Theorem 2 in [21] and Hoeffding inequality [24]. Indeed, if $t > 0$, X_i are independent bounded random variables, i.e. $a_i \leq X_i \leq b_i$, $i = 1, \ldots, n$, and $\overline{X} = 1/n \sum_{i=1}^n X_i$ then Hoeffding inequality implies that

$$P\left(\overline{X} - E[\overline{X}] \geq t\right) \leq \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$
$$P\left(|\overline{X} - E[\overline{X}]| \geq t\right) \leq 2\exp\left(-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Given that $x_{ij} = X_j$ where X_j are independent random variables with $-1 \le X_j \le 1$, $E[X_j^2] = \sigma_j^2$ (Assumption 1), we observe that $\|\boldsymbol{x}_i\|^2 = \sum_{j=1}^n X_j^2$ and

$$P\left(\left|\frac{\sum_{j=1}^n X_j^2}{n} - E\left[\frac{\sum_{j=1}^n X_j^2}{n}\right]\right| \ge t\right) =$$
$$P\left(\left|\frac{\sum_{j=1}^n X_j^2}{n} - \frac{R_0^2}{n}\right| \ge t\right) = P\left(\left|\frac{\|\boldsymbol{x}_i\|^2}{R_0^2} - 1\right| \ge \frac{tn}{R_0^2}\right)$$
$$\le 2\exp\left(-2nt^2\right).$$

Denoting $\varepsilon = tn/R_0^2$ and recalling that $0 \le X_j^2 \le 1$ we conclude that (1) holds true. Noticing that $E[x_{ik}x_{jk}] = E[x_{ik}]E[x_{jk}] = 0$, $E[y_k x_{ik}] = 0$, $-1 \le x_{ik} \le 1$, $-1 \le x_{jk} \le 1$, and $-1 \le y_k x_{ik} \le 1$ we observe that estimates (2) and (3) follow. □

Remark 1: Notice that if $\sigma_i^2 > 0$ then $R_0^2 > n\min_i\{\sigma_i^2\}$. Hence the r.h.s. of (1), (2) become exponentially close to 1 for n large enough.

The following Theorem is now immediate.

Theorem 2 (1-Element separation): Let elements of the set $\mathcal{M} \cup \mathcal{Y}$ be i.i.d. random points from the product distribution satisfying Assumption 1, $0 < \varepsilon < 1$, and $R_0 > 0$. Let the set $\mathcal{Y}$ comprises of a single element, i.e. $\mathcal{Y} = \{\boldsymbol{x}_{M+1}\}$, and consider

$$\ell_1(\boldsymbol{x}) = \left\langle \frac{\boldsymbol{x}}{R_0}, \frac{\boldsymbol{x}_{M+1}}{R_0} \right\rangle, \; h(\boldsymbol{x}) = \ell_1(\boldsymbol{x}) - 1 + \varepsilon. \tag{4}$$

Then

$$P\left(h(\boldsymbol{x}_{M+1}) \ge 0 \text{ and } h(\boldsymbol{x}_i) < 0 \text{ for all } \boldsymbol{x}_i \in \mathcal{M}\right)$$
$$\ge 1 - 2\exp\left(-\frac{2R_0^4\varepsilon^2}{n}\right) - M\exp\left(-\frac{R_0^4(1-\varepsilon)^2}{n}\right). \tag{5}$$

Proof of Theorem 2. Recall that for any events $A_1, \ldots, A_k$ the following estimate holds

$$P(A_1 \& A_2 \& \cdots \& A_k) \ge 1 - \sum_{i=1}^k (1 - P(A_i)). \tag{6}$$

The statement now follows from (1), (2) with $\delta = 1 - \varepsilon$, and (6). □

Remark 2: Theorem 2 not only establishes the fact that the set $\mathcal{M}$ can be separated away from $\mathcal{Y}$ by a linear functional with reasonably high probability. It also specifies the separating hyperplane, Eq. (4), and provides an estimate of the probability of such an event, Eq. (5). Note again that the probability, as a function of n, approaches 1 exponentially fast.

Let us now move to the case when the set $\mathcal{Y}$ contains more than one element. Theorem 3 below summarizes the result.

Theorem 3 (k-Element separation. Case 1): Let elements of the set $\mathcal{M} \cup \mathcal{Y}$ be i.i.d. random points from the product distribution satisfying Assumption 1, and $R_0 > 0$. Pick

$$0 < \delta < 1, \; 0 < \varepsilon < 1, \; 1 - \varepsilon - \delta(k-1) > 0,$$

and consider

$$\ell_k(\boldsymbol{x}) = \left\langle \frac{\boldsymbol{x}}{R_0}, \frac{\overline{\boldsymbol{x}}}{R_0} \right\rangle, \; \overline{\boldsymbol{x}} = \frac{1}{k}\sum_{i=1}^k \boldsymbol{x}_{M+i},$$
$$h_k(\boldsymbol{x}) = \ell_k(\boldsymbol{x}) - \frac{1 - \varepsilon - \delta(k-1)}{k}. \tag{7}$$

Then

$$P\left(h_k(\boldsymbol{x}_j) \ge 0 \;\&\; h_k(\boldsymbol{x}_i) < 0 \text{ for all } \boldsymbol{x}_i \in \mathcal{M}, \; \boldsymbol{x}_j \in \mathcal{Y}\right)$$
$$\ge 1 - 2k\exp\left(-\frac{2R_0^4\varepsilon^2}{n}\right) - k(k-1)\exp\left(-\frac{R_0^4\delta^2}{n}\right)$$
$$- kM\exp\left(-\frac{R_0^4(1 - \varepsilon - \delta(k-1))^2}{k^2 n}\right). \tag{8}$$

Proof of Theorem 3. Suppose that $\|\boldsymbol{x}_i\|^2/R_0^2 \ge 1 - \varepsilon$ for all $\boldsymbol{x}_i \in \mathcal{Y}$ (event A_1) and $\left\langle \frac{\boldsymbol{x}_i}{R_0}, \frac{\boldsymbol{x}_j}{R_0} \right\rangle \ge -\delta$ for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{Y}$, $i \ne j$ (event A_2). Then

$$\ell_k(\boldsymbol{x}_i) \ge \frac{1 - \varepsilon - \delta(k-1)}{k} \text{ for all } \boldsymbol{x}_i \in \mathcal{Y}.$$

Consider events A_{2+i}: $\left\langle \frac{\boldsymbol{x}}{R_0}, \frac{\boldsymbol{x}_{M+i}}{R_0} \right\rangle < \frac{1-\varepsilon-\delta(k-1)}{k}$ for all $\boldsymbol{x} \in \mathcal{M}$. According to Theorem 1 and Eq. (6)

$$P(A_1) \ge 1 - 2k\exp\left(-\frac{2R_0^4\varepsilon^2}{n}\right)$$
$$P(A_2) \ge 1 - k(k-1)\exp\left(-\frac{R_0^4\delta^2}{n}\right)$$
$$P(A_{2+i}) \ge 1 - M\exp\left(-\frac{R_0^4(1 - \varepsilon - \delta(k-1))^2}{k^2 n}\right).$$

Moreover, invoking (6) we obtain

$$P(A_1 \& \cdots \& A_{2+k}) \ge$$
$$1 - 2k\exp\left(-\frac{2R_0^4\varepsilon^2}{n}\right) - k(k-1)\exp\left(-\frac{R_0^4\delta^2}{n}\right)$$
$$- kM\exp\left(-\frac{R_0^4(1 - \varepsilon - \delta(k-1))^2}{k^2 n}\right).$$

Noticing that

$$\ell_k(\boldsymbol{x}) = \left\langle \frac{\boldsymbol{x}}{R_0}, \frac{\overline{\boldsymbol{x}}}{R_0} \right\rangle = \sum_{i=1}^k \frac{1}{k}\left\langle \frac{\boldsymbol{x}}{R_0}, \frac{\boldsymbol{x}_{M+i}}{R_0} \right\rangle$$

we conclude that $A_1 \& \cdots \& A_{2+k}$ imply that $\ell_k(\boldsymbol{x}) < \frac{1-\varepsilon-\delta(k-1)}{k}$ for all $\boldsymbol{x} \in \mathcal{M}$. Furthermore, $A_1 \& A_2$ imply that $\ell_k(\boldsymbol{x}) \ge \frac{1-\varepsilon-\delta(k-1)}{k}$ for all $\boldsymbol{x} \in \mathcal{Y}$. The result now follows □.

Estimate (8) in Theorem 3 does not account for any spurious correlations in the sets that are to be separated from $\mathcal{M}$. In practice, however, such correlations might occur. Theorem 4 presents an adapted statement enabling to deal with spurious or natural correlations.

Theorem 4 (k-Element separation. Case 2): Let elements of the set $\mathcal{M} \cup \mathcal{Y}$ be i.i.d. random points from the product

distribution satisfying Assumption 1, $0 < \varepsilon < 1, 0 < \mu < 1-\varepsilon$ and $R_0 > 0$. Pick $\boldsymbol{x}_j \in \mathcal{Y}$ and consider

$$\ell_k(\boldsymbol{x}) = \left\langle \frac{\boldsymbol{x}}{R_0}, \frac{\boldsymbol{x}_j}{R_0} \right\rangle, \; h(\boldsymbol{x}) = \ell_k(\boldsymbol{x}) - 1 + \varepsilon + \mu,$$
$$\Omega = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \left\langle \frac{\boldsymbol{x}_j}{R_0}, \frac{\boldsymbol{x}_j - \boldsymbol{x}}{R_0} \right\rangle \leq \mu \right\} \tag{9}$$

Then

$$P\left(h(\boldsymbol{x}) \geq 0 \text{ and } h(\boldsymbol{x}_i) < 0 \text{ for all } \boldsymbol{x}_i \in \mathcal{M}, \boldsymbol{x} \in \Omega\right)$$
$$\geq 1 - 2k \exp\left(-\frac{2R_0^4 \varepsilon^2}{n}\right) - M \exp\left(-\frac{R_0^4 (1-\varepsilon-\mu)^2}{n}\right). \tag{10}$$

Proof of Theorem 4. Let $\|\boldsymbol{x}_j\|^2 \geq 1 - \varepsilon$. Then $h(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \Omega$. Estimate (10) hence follows from Theorem 1 and Eq. (6). □.

IV. Discussion

The results presented in Theorems 1 – 4 state that simple and elementary shallow systems are capable of singling out random spurious errors of larger AI systems. The results can be generalized to a broad range of distributions, including to non i.i.d. settings (see [25] for details and bounds). Isolation of the errors can be implemented in a non-iterative and remarkably simple way. Moreover, such isolation can be achieved with linear functionals. In addition to simplicity and computational efficiency of linear functionals, they also offer good generalization capabilities. And in fact, as we show below, they may be exponentially better than e.g. n-balls or ellipsoids, depending on the radius.

To demonstrate this point, recall that $\boldsymbol{x}_i$ concentrate in a vicinity of an $n-1$-sphere centered at the origin. Linear functionals (4), (7), (9) "cut"-off their corresponding spherical caps from this sphere. Consider

$$C_n(\varepsilon) = B_n(1) \cap \left\{ \xi \in \mathbb{R}^n \;\middle|\; \left\langle \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}, \xi \right\rangle \geq 1 - \varepsilon \right\}. \tag{11}$$

Let

$$\rho(\varepsilon) = (1 - (1-\varepsilon)^2)^{\frac{1}{2}}.$$

Note that $\rho(\varepsilon)$ is the radius of the ball containing the spherical cup C_n. Lemma 1 estimates volumes of spherical caps $C_n(\varepsilon)$ relative to relevant n-balls of radius $\rho(\varepsilon)$.

Lemma 1: Let $C_n(\varepsilon)$ be a spherical cap defined as in (11), $\varepsilon \in (0,1)$. Then

$$\frac{\rho(\varepsilon)^{n+1}}{2} \left[\frac{1}{\pi^{\frac{1}{2}}} \frac{\Gamma\left(\frac{n}{2}+1\right)}{\Gamma\left(\frac{n}{2}+\frac{3}{2}\right)} \right] < \frac{\mathcal{V}(C_n(\varepsilon))}{\mathcal{V}(B_n(1))} \leq \frac{\rho(\varepsilon)^n}{2}.$$

Note that [26] $\mathcal{V}(B_n(r)) = r^n \mathcal{V}(B_n(1))$ for all $n \in \mathbb{N}$ $r > 0$. Hence the estimate of $\mathcal{V}(C_n(\varepsilon))$ from above is:

$$\mathcal{V}(C_n(\varepsilon)) \leq \frac{1}{2} \mathcal{V}(B_n(1)) \rho(\varepsilon)^n. \tag{12}$$

Let us calculate the estimate of $\mathcal{V}(C_n(\varepsilon))$ from below. It is clear that

$$\mathcal{V}(C_n(\varepsilon)) = V(B_{n-1}(1)) \int_{1-\varepsilon}^{1} (1-x^2)^{\frac{n-1}{2}} dx$$

The integral in the right-hand side of the above expression can be estimated from below as

$$\int_{1-\varepsilon}^{1} (1-x^2)^{\frac{n-1}{2}} dx > \int_{1-\varepsilon}^{1} (1-x^2)^{\frac{n-1}{2}} x dx$$
$$= \frac{1}{2} \cdot \frac{1}{\frac{n}{2}+\frac{1}{2}} \cdot (1 - (1-\varepsilon)^2)^{\frac{n+1}{2}} = \frac{1}{2} \cdot \frac{1}{\frac{n}{2}+\frac{1}{2}} \cdot \rho(\varepsilon)^{n+1}$$

Recall that $B_n(1) = \frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}+1\right)}$, $\Gamma(x+1) = x\Gamma(x)$. Hence

$$B_{n-1}(1) \cdot \frac{1}{\frac{n}{2}+\frac{1}{2}} = \frac{\pi^{\frac{n-1}{2}}}{\Gamma\left(\frac{n}{2}+\frac{1}{2}\right)} \frac{1}{\frac{n}{2}+\frac{1}{2}}$$
$$= \frac{\pi^{\frac{n-1}{2}}}{\Gamma\left(\frac{n+1}{2}+1\right)},$$

and

$$\int_{1-\varepsilon}^{1} (1-x^2)^{\frac{n-1}{2}} dx > \frac{1}{2} \mathcal{V}(B_n(1)) \rho(\varepsilon)^{n+1} \left[\frac{1}{\pi^{\frac{1}{2}}} \frac{\Gamma\left(\frac{n}{2}+1\right)}{\Gamma\left(\frac{n}{2}+\frac{3}{2}\right)} \right]$$

□

Corollary 1: Let $C_n(\varepsilon)$ be a spherical cap defined as in (11), $\varepsilon \in (0,1)$, and $B_n(k\rho(\varepsilon))$ be an n-ball with radius $k\rho(\varepsilon)$, $k \in \mathbb{R}_{>0}$. Then

$$\frac{\mathcal{V}(B_n(k\rho(\varepsilon)))}{\mathcal{V}(C_n(\varepsilon))} < k^n \frac{2\pi^{\frac{1}{2}}}{\rho(\varepsilon)} \left[\frac{\Gamma\left(\frac{n}{2}+1\right)}{\Gamma\left(\frac{n}{2}+\frac{3}{2}\right)} \right]^{-1}$$

Remark 3: Using Stirling's approximation we observe that

$$\frac{\Gamma\left(\frac{n}{2}+1\right)}{\Gamma\left(\frac{n}{2}+\frac{3}{2}\right)} = O\left(n^{-\frac{1}{2}}\right).$$

Thus $\frac{\mathcal{V}(B_n(\varepsilon))}{\mathcal{V}(C_n(\varepsilon))} < k^n H(n,\varepsilon)$ where $H(n,\varepsilon) = O\left(\frac{n^{1/2}}{\rho(\varepsilon)}\right)$.

According to Corollary 1 the volumes of $B_n(\varepsilon)$, $B_n(\kappa\rho(\varepsilon))$, $\kappa \in (0,1)$ decay exponentially with dimension n relative to that of $C_n(\varepsilon)$. This implies that distance-based detectors are extremely localized, and in comparison with simple perceptrons, the proportion of points to which they respond positively is negligibly small. On the other hand, filtering properties of simple perceptrons are extreme in high dimension (Theorems 2–4). This combination of properties makes perceptrons and their ensembles particularly attractive for fine-tuning of existing AI systems.

V. Examples

In this section we illustrate our theoretical results with two numerical examples: 1) a synthetic test in an n-hypercube illustrating how linear separability changes with dimensionality of n (Theorem 2), and 2) recognition of gestures from American Sign Language.

A. Example 1. A synthetic test in a hypercube

In this example, we generated sets of $M = 10^4$ random vectors in $[-1,1]^n$ for various values of n. We randomly picked a point in each sample and constructed separation hyperplanes h in accordance with (4). This was followed by the assessment of whether the sign of $h(\boldsymbol{x})$ for all remaining

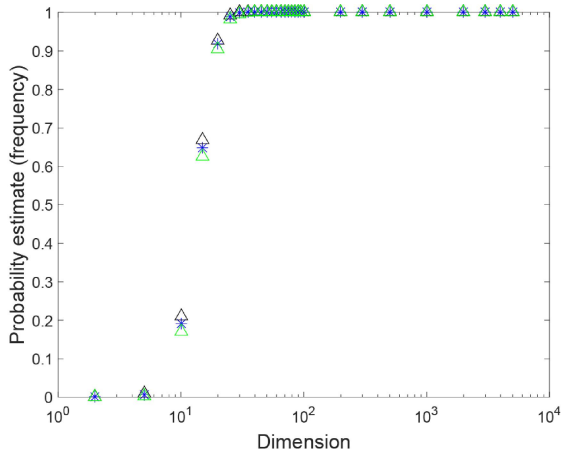

Fig. 2. The probability (frequency) that an element of the set $\mathcal{M}$ (formed by random i.i.d. vectors drawn from the n-cube $[-1,1]^n$) is linearly separable from the rest as a function of n. $M = |\mathcal{M}| = 10^4$. Triangles indicate minimum/maximum values in each trial, stars show sample means.

elements $\boldsymbol{x}$ in the sample is negative. The procedure was repeated 200 times for each n, and the frequency of successful separations recorded. The results are summarized in Figure 2. As one can see from this figure, the probability estimate (frequency) rapidly approaches 1 when $n \simeq 20$ and stays close to 1 for larger values of n, as expected.

B. Example 2. An Algorithm For Distinguishing The Ten Digits In American Sign Language

1) Shallow corrector algorithm: The algorithm is a six-step process where the inputs are the sets $\mathcal{S}$ and $\mathcal{Y}$ (cf. [27], [28]). The set $\mathcal{S}$ contains states $\boldsymbol{x}_i$ for all images that have been assessed. The states $\boldsymbol{x}_i$ are the vectors containing the values of pre-softmax layer bottlenecks of size n for however many neurons are in the penultimate layer. Elements of this set that gave incorrect readings are noted and copied into the set $\mathcal{Y}$.

1) *Centering*. First the current data available is centered. The centered sets are denoted as $\mathcal{S}_c$ and $\mathcal{Y}_c$ and are formed by subtracting the means $\overline{\boldsymbol{x}}(\mathcal{S})$, $\overline{\boldsymbol{x}}(\mathcal{Y})$ from the elements of $\mathcal{S}$ and $\mathcal{Y}$, respectively:

$$\mathcal{S}_c = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} = \xi - \overline{\boldsymbol{x}}(\mathcal{S}),\, \xi \in \mathcal{S}\}$$

$$\mathcal{Y}_c = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} = \xi - \overline{\boldsymbol{x}}(\mathcal{S}),\, \xi \in \mathcal{Y}\}.$$

2) *Regularization*. The covariance matrix of $\mathcal{S}$ is calculated along with the corresponding eigenvalues and eigenvectors. The mean eigenvalue is then calculated and new regularized sets $\mathcal{S}_r$, $\mathcal{Y}_r$ are produced as follows. All eigenvectors that correspond to the eigenvalues which are above the given mean are combined into a single matrix H. The transpose of this new matrix is multiplied by the original values of $\boldsymbol{x}_i \in \mathcal{S}_c$ creating regularised data of smaller dimension (Kaiser-Guttman test [29]):

$$\mathcal{S}_r = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} = H^T \xi,\, \xi \in \mathcal{S}_c\}$$

$$\mathcal{Y}_r = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} = H^T \xi,\, \xi \in \mathcal{Y}_c\}.$$

3) *Whitening*. The two sets then undergo a whitening coordinate transformation ensuring that the covariance matrix of the transformed data is the identity matrix:

$$\mathcal{S}_w = \{\boldsymbol{x} \in R^m | \boldsymbol{x} = Cov(\mathcal{S}_r)^{-\frac{1}{2}} \xi,\, \xi \in \mathcal{S}_r\}$$

$$\mathcal{Y}_w = \{\boldsymbol{x} \in R^m | \boldsymbol{x} = Cov(\mathcal{S}_r)^{-\frac{1}{2}} \xi,\, \xi \in \mathcal{Y}_r\}.$$

4) *Training: Clustering.* The set $\mathcal{Y}_w$ (the set of errors) is then partitioned into p clusters $\mathcal{Y}_{w,1}, \ldots, \mathcal{Y}_{w,p}$ that's elements are pairwise positively correlated.

5) *Training: Shallow aggregation.* For each $\mathcal{Y}_{w,i}$, $i = 1, \ldots, p$ and its complement $\mathcal{S}_w \setminus \mathcal{Y}_{w,\,i}$ we construct the following separating hyperplanes:

$$\begin{aligned} h_i(\boldsymbol{x}) =& \ell_i(\boldsymbol{x}) - c_i, \\ \ell_i(\boldsymbol{x}) =& \left\langle \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}, \boldsymbol{x} \right\rangle,\ c_i = \min_{\xi \in \mathcal{Y}_{w,\,i}} \left\langle \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}, \xi \right\rangle \\ \boldsymbol{w}_i =& (Cov(\mathcal{S}_w \setminus \mathcal{Y}_{w,i}) + Cov(\mathcal{Y}_{w,i}))^{-1} \times \\ & (\overline{\boldsymbol{x}}(\mathcal{Y}_{w,i}) - \overline{\boldsymbol{x}}(\mathcal{S}_w \setminus \mathcal{Y}_{w,i})). \end{aligned}$$

6) *Deployment stage*. At the deployment stage, any $\boldsymbol{x}$ that is generated by the original Deep Learning AI is put through the ensemble $h_i(\boldsymbol{x})$, and then if for some $\boldsymbol{x}$ any of the values of $h_i(\boldsymbol{x}) \geq 0$ then the corresponding $\boldsymbol{x}$ is reported accordingly and can be swapped, or reported as error or deleted from the set if necessary.

In addition to the sequence of steps outlined above we have also experimented with a slightly modified procedure in which an optional projection step, step 3^*), is introduced after the whitening transformation:

3^*) *Optional: projection onto the unit sphere*. Project the whitened data onto the unit sphere by replacing the elements of $\mathcal{S}_w$, $\mathcal{Y}_w$ with their corresponding normalized values: $\boldsymbol{x} \mapsto \boldsymbol{x}/\|\boldsymbol{x}\|$.

Note that individual components of the data vectors may no longer satisfy the independence assumption. Nevertheless, if the data is reasonably equidistributed, stochastic separation theorems [21], [25], [28], [27] may apply to this case too.

2) Setup and Datasets: In this example we run Inception algorithm[1] on ten sets of images that correspond to the American Sign Language pictures for 0-9 (see Fig. 3). The

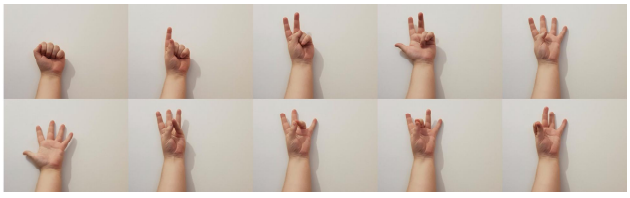

Fig. 3. Examples of the sort of images that appear in the current model's data set of the American Sign Language single hand positions for 0 (top left) to 9 (bottom right)

original set of flower pictures that Inception provided by

[1]https://www.tensorflow.org/tutorials/image_retraining

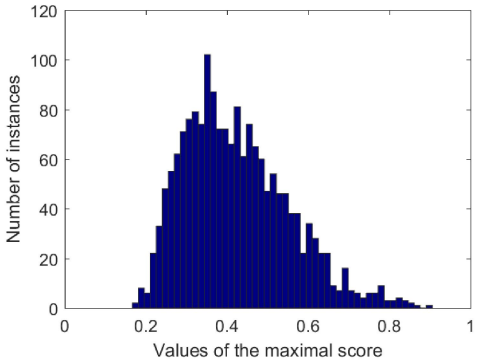

Fig. 4. Histogram of the maximal scores of the images corresponding to errors.

TABLE I
ERRORS PER GESTURE (MISCLASSIFICATIONS). TOP ROW CORRESPONDS TO THE GESTURE NUMBER, THE BOTTOM ROW INDICATES THE NUMBER OF ERRORS FOR EACH GESTURE.

0	1	2	3	4	5	6	7	8	9
10	52	235	62	410	80	269	327	207	108

default were switched with a series of 1000 images we took for each of the ten gestures. These sets contained profile shots of the person's hand, along with 3/4 profiles and looking from above and below.

3) Experiments and results: Once the network was trained, additional 10000 images of the same ratio were evaluated using the trained system. The result was an 82.4% success rate for the adapted algorithm. For these experiments the classification decision rule was to return a gesture number that corresponds to the network output with the highest score (winner-takes-all). Ties are broken arbitrarily at random. The observed performance was comparable/similar to that reported in e.g. [30] (see also references therein).

The system was forced to make a decision regardless of the value of the maximal score. In terms of conventional ROC curves, the setup corresponds to the rightmost top point on the curve (the value of the threshold is 0). The histogram of maximal scores corresponding to errors is shown in Fig. 4, and numbers of errors per each gesture in the trained system are shown in Table I. The variance of errors is mostly consistent among the ten classes with very few errors for the "0" gesture, likely due to its unique shape among the classes.

Once the errors were isolated, shallow single-element error correction perceptrons (as described in Section V-B1) have been created to improve the original AI. For simplicity, we focused on the task of building an AI corrector capable of separating the original AI's (i.e. trained Inception) correct responses from the ones that have been labelled as errors. In this task, the deployment step (step 6) in the shallow corrector algorithm) was as follows:

$$\text{If } \exists\ i \in \{1,\dots,p\}:\ h_i(\boldsymbol{x}) \geq 0\ \Rightarrow\ \text{report } \boldsymbol{x} \text{ as error,} \quad (13)$$

TABLE II
ERROR TYPES IN THE SYSTEM WITH CORRECTOR.

Inception's behaviour	Corrector's response	Error Type
Error	≥ 0	True Positive
	< 0	False Negative
Correct	≥ 0	False Positive
	< 0	True Negative

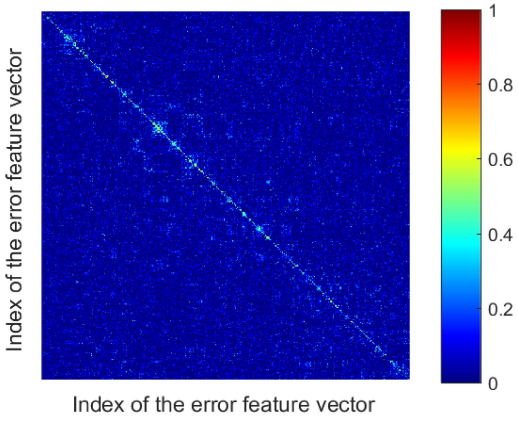

Fig. 5. Values of the normalized inner products $\langle \boldsymbol{x}_i/\|\boldsymbol{x}_i\|, \boldsymbol{x}_j/\|\boldsymbol{x}_j\|\rangle$ (color-coded) for the data points labelled as errors.

where p is the number of clusters used in the algorithm.

To train the shallow correctors, the testing data set of 10000 images that has been used to assess performance of Inception was split into two non-overlapping subsets. The first subset, comprised of 6592 records of data points corresponding to correct responses and 1408 records corresponding to errors, was used to train the correctors. This subset was the corrector's training set, and it accounted for 80% of the data. The second subset, the corrector's testing set, combining 1648 data points of correct responses and 352 elements labelled as errors was used to test the corrector.

To quantify and assess performance of the corrector in both training and testing phases we used the definitions of True Positives, True Negatives, False Positives, and False Negatives as specified in Table II.

The shallow corrector algorithm was run on the first subset, the corrector's training set. For this dataset the regularization step, step 2), returned 174 principal components reducing the original dimensionality more than 10 times. After the whitening transformation, step 3), we assessed the values of $\langle \boldsymbol{x}_i/\|\boldsymbol{x}_i\|, \boldsymbol{x}_j/\|\boldsymbol{x}_j\|\rangle$ (shown in Fig. 5). According to Fig 5, data points labelled as errors are largely orthogonal to each other apart from few modestly-sized groups.

The number of clusters, parameter p in step 4), was varied from 2 to 1408 in regular increments. As a clustering algorithm we used standard k-means routine (k-means ++) supplied with MATLAB 2016a. For each value of p, we run the k-means algorithm 10 times. For each clustering pass we constructed the corresponding separating hyperplanes h_i as prescribed

in step 5), and combined them into a single corrector in accordance with (13). Performance of the corrector is shown in Fig. 6. Note that as the number of clusters increases, the True Negative rate approaches 1 for both versions of the algorithm (with and without step 3*). This is consistent with theoretical predictions stemming from Theorem 2. We also observed that performance drops rapidly with the average number of elements assigned to a cluster. In view of our earlier observation that vectors labelled as "errors" appear to be nearly orthogonal to each other, this drop is consistent with the bound provided in Theorem 3.

Next we assessed performance of the corrector on the corrector's testing set. Results are shown in Fig. 7. As before, the behavior is consistent with theoretical bounds provided in Theorem 2–4. Notably, both versions of the algorithm removed larger relative percentages of errors than they introduced.

VI. Conclusion

In this work we presented a novel technology for computationally cheap and non-iterative improvements of sophisticated Multilayer and Deep Learning neural networks and Artificial Intelligence (AI) systems by shallow cascades. These improvements can be employed for both learning new skills as well as for "learning errors away" in the existing architectures. Theoretical results are not limited to the realm of Artificial Intelligence. Similar to [31], the results can be employed to explain extreme selectivity of neurons and reveal simple mechanisms of learning in stratified brain structures.

The proposed concept builds on our previous work [13] and explicitly extends to Deep Learning architectures and extends to product measure distributions. In contrast to [14], and when the clustering structure is fixed, the method is inherently one-shot and non-iterative. This makes the proposal particularly suitable for large-scale multi-agent and distributed systems. Developing the technology further to suit this specific class of applications appears to be a natural way forward.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[2] P. P. Brahma, D. Wu, and Y. She, "Why deep learning works: a manifold disentanglement perspective," *IEEE Transactions On Neural Networks And Learning Systems*, vol. 27, no. 10, pp. 1997–2008, 2016.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, pp. 1–42, 2014, dOI:10.1007/s11263-015-0816-y.

[4] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5mb model size," *arXiv preprint, arXiv:1602.07360*, 2016.

[5] A. Kuznetsova, S. Hwang, B. Rosenhahn, and L. Sigal, "Expanding object detectors horizon: Incremental learning framework for object detection in videos," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 28–36.

[6] I. Misra, A. Shrivastava, and M. Hebert, "Semi-supervised learning for object detectors from video," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3594–3602.

[7] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3282–3289.

[8] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, https://arxiv.org/abs/1604.04326.

[9] L. Pratt, "Discriminability-based transfer between neural networks," *Advances in Neural Information Processing*, no. 5, pp. 204–211, 1992.

[10] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[11] T. Chen, I. Goodfellow, and J. Shlens, "Net2net: Accelerating learning via knowledge transfer," *ICLR 2016*, 2015.

[12] V. Vapnik and R. Izmailov, "Knowledge transfer in svm and neural networks," *Annals of Mathematics and Artificial Intelligence*, pp. 1–17, 2017.

[13] A. Gorban, R. Burton, I. Romanenko, and T. I., "One-trial correction of legacy AI systems and stochastic separation theorems," 2016.

[14] T. J. Draelos, N. E. Miner, C. C. Lamb, C. M. Vineyard, K. D. Carlson, C. D. James, and J. B. Aimone, "Neurogenesis deep learning," *arXiv preprint arXiv:1612.03770*, 2016.

[15] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962.

[16] M. Gromov, *Metric Structures for Riemannian and non-Riemannian Spaces. With appendices by M. Katz, P. Pansu, S. Semmes. Translated from the French by Sean Muchael Bates*. Boston, MA: Birkhauser, 1999.

[17] ——, "Isoperimetry of waists and concentration of maps," *GAFA, Geomteric and Functional Analysis*, vol. 13, pp. 178–215, 2003.

[18] A. Gorban, "Order-disorder separation: Geometric revision," *Physica A*, vol. 374, pp. 85–102, 2007.

[19] J. Gibbs, *Elementary Principles in Statistical Mechanics, developed with especial reference to the rational foundation of thermodynamics*. New York: Dover Publications, 1960 [1902].

[20] P. Lévy, *Problèmes concrets d'analyse fonctionnelle*, 2nd ed. Paris: Gauthier-Villars, 1951.

[21] A. Gorban and I. Tyukin, "Stochastic separation theorems," *Neural Networks*, vol. 94, pp. 255–259, 2017.

[22] V. Vapnik and O. Chapelle, "Bounds on error expectation for support vector machines," *Neural Computation*, vol. 12, no. 9, pp. 2013–2036, 2000.

[23] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bulletin of the American mathematical society*, vol. 39, no. 1, pp. 1–49, 2002.

[24] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American statistical association*, vol. 58, no. 301, pp. 13–30, 1963.

[25] A. N. Gorban, B. Grechuk, and I. Y. Tyukin, "Augmented artificial intelligence," *arXiv preprint arXiv:1802.02172*, 2018.

[26] K. Ball, "An elementary introduction to modern convex geometry," *Flavors of geometry*, vol. 31, pp. 1–58, 1997.

[27] I. Y. Tyukin, A. N. Gorban, K. Sofeikov, and I. Romanenko, "Knowledge transfer between artificial intelligence systems," *arXiv preprint arXiv:1709.01547*, 2017.

[28] A. Gorban and I. Tyukin, "Blessing of dimensionality: mathematical foundations of the statistical physics of data," *Philosophical Transactions of the Royal Society A*, vol. 376, p. 20170237, 2018.

[29] D. Jackson, "Stopping rules in principal components analysis: A comparison of heuristical and statistical approaches," *Ecology*, vol. 74, no. 8, pp. 2204–2214, 1993.

[30] V. Bheda and D. Radpour, "Using deep convolutional networks for gesture recognition in american sign language," *arXiv preprint arXiv:1710.06836*, 2017.

[31] I. Y. Tyukin, A. N. Gorban, C. Calvo, J. Makarova, and V. A. Makarov, "High-dimensional brain. A tool for encoding and rapid learning of memories by single neurons," *Bulletin of Mathematical Biology*, pp. 1–33, 2018. [Online]. Available: https://doi.org/10.1007/s11538-018-0415-5