# Emergence of Structure in Ensembles of Random Neural Networks

**Luca Muscarnera**[1]
**Luigi Loreti**[2] **Giovanni Todeschini**[3]
**Alessio Fumagalli**[1] **Francesco Regazzoni**[1]

[1]MOX Laboratory, Department of Mathematics, Politecnico di Milano (Italy)
{luca.muscarnera,alessio.fumagalli,francesco.regazzoni}@polimi.it

[2] Optopc SA, Chiasso (Switzerland)
luigi.loreti@optopc.com

[3] Rebel Dynamics Srl, Cesana Brianza (Italy)
giovanni.todeschini@rebeldynamics.it

## Abstract

Randomness is ubiquitous in many applications across data science and machine learning. Remarkably, systems composed of random components often display emergent global behaviors that appear deterministic, manifesting a transition from microscopic disorder to macroscopic organization. In this work, we introduce a theoretical model for studying the emergence of collective behaviors in ensembles of random classifiers. We argue that, if the ensemble is weighted through the Gibbs measure defined by adopting the classification loss as an energy, then there exists a finite temperature parameter for the distribution such that the classification is optimal, with respect to the loss (or the energy). Interestingly, for the case in which samples are generated by a Gaussian distribution and labels are constructed by employing a teacher perceptron, we analytically prove and numerically confirm that such optimal temperature does not depend neither on the teacher classifier (which is, by construction of the learning problem, unknown), nor on the number of random classifiers, highlighting the universal nature of the observed behavior. Experiments on the MNIST dataset underline the relevance of this phenomenon in high-quality, noiseless, datasets. Finally, a physical analogy allows us to shed light on the self-organizing nature of the studied phenomenon.

## 1 Introduction

The study of Random Neural Networks (also known as Untrained Neural Networks or, in a slightly different context, as Neural Networks with Random Weights [1]) is inherently entangled to the foundations of the Learning problem [1, 2]. A neural network can be, in fact, studied as the evolution of a disordered system — since at initialization the realization of the parameters is a form of "frozen" (or more technically, quenched) disorder, making the network, in fact, random [3] — towards a state in the space of parameters — reached through any possible training algorithm [4] — where the randomness is diluted in favor of the emergence of some form of structure in the biases and in the weight matrices, which are learned through the observation of data [5]. In fact, when a Neural Network is initialized, weight matrices are obtained by sampling from random matrices whose entries — in the classical initialization schemes [6, 7] — lack of a covariance structure which therefore leads to random (agnostic) classification; on the other hand, the high accuracies obtained after training

Preprint. Under review.

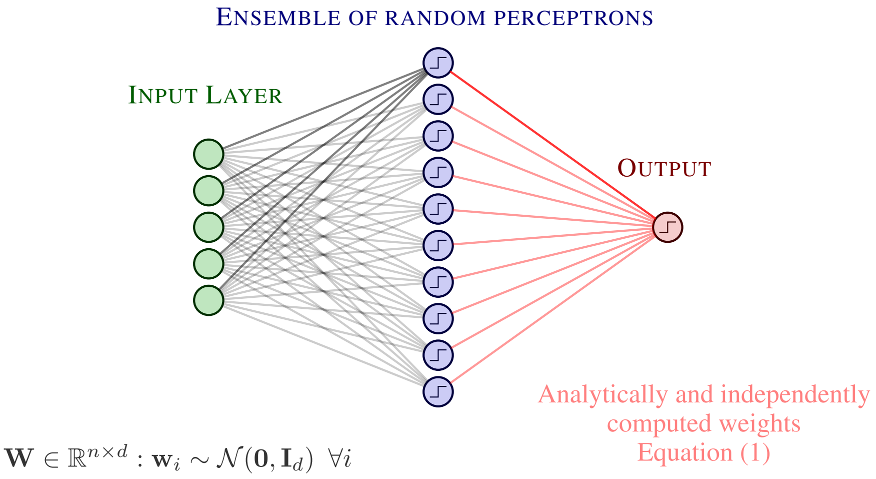

ENSEMBLE OF RANDOM PERCEPTRONS
INPUT LAYER
OUTPUT
Analytically and independently
computed weights
Equation (1)
$\mathbf{W} \in \mathbb{R}^{n\times d} : \mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d) \;\; \forall i$

Figure 1: Model of the proposed architecture, with null biases. The gray weights are sampled from a normal distribution, while red weights are computed analytically by the rule described in Equation 1. The highlighted subnetwork represents the flow of information from the input, to one randomly constructed classifier and to the output. The hidden layer can be considered as an ensemble of random elementary neural networks (perceptrons) whose output is then combined through the analytically computed weights, which do not require a training phase to be estimated, with each weight in the last layer being independent from the others. Both hidden and output neurons adopt the $\mathrm{sign}$ activation function.

suggest the existence of some form of highly structured information encoded into the weights [8]. The emergence of such information in the parameters of the network allows the performing of many different tasks, often with characteristics that are superior in quality with respect to their human based counterpart [9, 10]. However, it is still not clear what characterizes the geometry and the distribution of weights in a well-trained neural network [11, 12]. What property is responsible for making a certain realization of the parameters more efficient than the others? Is it possible to impose the same form of structure *before* the training, or even as *alternative* training strategy? These questions, inevitably linked with the complex nature of neural networks [13, 14], drove us towards the construction of a minimal model of a Random Neural Network where randomness in the system is preserved, and the classification outcome is constructed through a weighted majority voting performed by the randomly initialized neurons of the network. The underlying motif of our work can be summarized by the following question: can a system of many [15] *randomly* and independently initialized neurons perform non trivial tasks if the magnitude of their influence on the final classification is informed by the loss function? More formally, we propose a framework where the output of a certain (large) number $n$ of randomly initialized perceptrons [16] is combined by means of a weighted sum wherein the weight $\alpha_{\mathbf{w}}$ associated to each perceptron with parameter $\mathbf{w} \in \mathbb{R}^d$ satisfies the following proportionality relationship

$$\alpha_{\mathbf{w}}(\beta) \propto \exp(-\beta\mathcal{L}(\phi(\mathbf{w}))) \tag{1}$$

with $\mathcal{L}$ being the loss function associated to each perceptron (mapping each classifier into an empirical risk measure computed on observed data), $\beta$ being a parameter that allows to modulate the influence of low loss configurations, denoted from now on as the inverse temperature of the model (being, thus, a positive value) and $\phi$ the *realization* map that associates a weight vector $\mathbf{w}$ to the perceptron $\mathbf{x} \mapsto \mathrm{sign}(\mathbf{w}^T\mathbf{x})$. This choice of the weight is consistent with the idea of defining a Gibbs measure [17] over the possible classifiers, with $\mathcal{L}$ playing the role of an energy. The ensemble is taking the form

$$\hat{y}_{ens}(\mathbf{W}, \mathbf{x}) := \mathrm{sign}\left(\sum_i^n \alpha_{\mathbf{w}_i}(\beta)\,\mathrm{sign}(\mathbf{w}_i^T\mathbf{x})\right) = \mathrm{sign}\left(\boldsymbol{\alpha}_{\mathbf{W}}^T(\beta)\,\mathrm{sign}\left(\mathbf{W}\mathbf{x}\right)\right),$$

where $\mathbf{W} \in \mathbb{R}^{n\times d}$ is the realization of a suitably defined random matrix, with statistically independent entries and with the $i-$th row denoted as $\mathbf{w}_i$. From a probabilistic perspective, we can therefore think that the vector of the parameters of the network, defined concatenating $\mathrm{flat}(\mathbf{W})$ and $\boldsymbol{\alpha}_{\mathbf{W}}(\beta)$, may be imagined as a non isotropic random vector since the vector of weights on the last layer is statistically non independent (even in a deterministic relationship, if one assumes that the data that induces the loss is fixed) from the entries of the random matrix $\mathbf{W}$ that models the action of the first layer. This article aims to prove, empirically and analytically when possible, that this specific structure in the statistics of weights allows the definition of a training free learning paradigm, which may be — after further development — computationally efficient especially in the context of dedicated hardware able to process massively parallel workflows, due to the avoidance of the seriality bottleneck which is intrinsic in more classical training methods .

Our paper is structured in the following way: in Section 2 we briefly discuss the literature on Random Neural Networks, with a sharp focus on the methods that involve stochasticity as a tool to perform inference, in Section 3 we discuss the main analytical result based on the hypothesis of Gaussian uncorrelated data and the corresponding experimental validation setup, proving in both cases that the $\beta$ parameter is independent of the specific properties of the task that we aim to learn and in Section 4 we describe an experiment on organic data — the `MNIST` [18] dataset — underlining a non obvious compatibility with the proposed analytical model. Finally, in Section 5 a physical analogy sheds a light on the underlying dynamics of the classification and the implications of our results are critically discussed.

## 2 Related work

To the best of the authors' knowledge, the approach of imposing a covariance structure to the weights of a Neural Network has not been extensively tackled in literature. The survey [7] presents some examples of data-driven initializations, where some information from data is distilled into the weights in order to achieve better convergence properties or higher accuracy. Nevertheless, literature appears to be lacking of examples where the absence of independence among the neural networks' weights is actively exploited to perform inference, without a further training phase. Some similarities could be found in the conceptual ideas behind Reservoir Computing [19], where properties of random weight matrices are exploited to perform useful tasks, or Extreme Learning Machines [20], where random weights allow to perform a sort of uninformed feature selection. In an extremely broad sense, architectural bias constitutes a form of statistical dependence among weights; for instance, a Convolutional Neural Network may be interpreted as a Multi Layer Perceptron where, in each layer, some weights are constrained to be equal [21] forcing therefore a non-isotropic correlation structure among weights (which are of course not part of the actual parameters of the network, since they simply would be redundant). Nonetheless, our research in literature [22, 23, 24, 25] suggests that this may be the first work to actively discuss the injection of statistical dependence between random weights of the untrained neural network to perform inference.

## 3 Training-free Gaussian data classification

Consider the case of Gaussian (isotropic) data where we imagine that labels are generated by means of an unknown teacher [26] vector $\mathbf{w}_* \in \mathbb{R}^d$ (a fixed $d-$dimensional vector) which induces the definition of two half-spaces, one where data is classified as $+1$ and one where data is classified as $-1$ according to the sign of the dot product of a query vector and $\mathbf{w}_*$. Let $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ be an isotropic multivariate normal random variable. Suppose to observe a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times \{-1, 1\}$ where each sample $\mathbf{x}_i$ is a realization of $\mathbf{x}$ and $y_i = y_*(\mathbf{x}_i)\ \forall i$ where $y_*(\mathbf{x}) = \mathrm{sign}(\mathbf{w}_*^T \mathbf{x})$. Our objective, in this context, is to study wether for a given family of random classifiers of the form

$$\hat{y}_i(\mathbf{x}) = \mathrm{sign}(\mathbf{w}_i^T \mathbf{x}) \qquad i = 1, 2, ..., n$$

there exists a (untrained) weighted majority voting scheme to combine their outputs such that the classification performed by the ensemble significantly outperforms the best (according to a certain loss function $\mathcal{L}$) classifier in the family. More formally, we study the case where $\mathbf{w}_1, \ldots, \mathbf{w}_n$ are the rows of a random matrix of gaussian entries $\mathbf{W}$ (where $W_{ij} \sim \mathcal{N}(0, 1)\ \forall i \in [n], j \in [d]$) and thus, due to the radial symmetry of the isotropic multivariate normal distribution, each random classifier defines a random separating hyperplane passing through the origin by sampling uniformly over the possible normal vectors. As we mentioned, the weights of the majority voting are defined according to a Gibbs measure over the different quenched random functionals, leading to the ensemble classification

$$\hat{y} = \mathrm{sign}\left(\frac{1}{Z_n(\beta)}\sum_i^n \exp(-\beta\mathcal{L}(\hat{y}_i))\hat{y}_i\right),$$

where the loss function $\mathcal{L}$ plays the role of the energy in the Gibbs measure, the $\mathrm{sign}$ function is intended — with a slight abuse of notation — as the operator that maps a function $f : \mathbb{R}^d \to \mathbb{R}$ into its projection on the codomain $\{-1, 1\}$ with $(\mathrm{sign}(f))(\mathbf{x}) = \mathrm{sign}(f(\mathbf{x}))\ \forall \mathbf{x} \in \mathbb{R}^d$ and the Partition Function $Z_n(\beta) = \sum_i^n \exp(-\beta\mathcal{L}(\hat{y}_i))$ acts as a normalization constant which, being composed with a $\mathrm{sign}$, serves the only purpose of making the limit $\beta \to \infty$ well defined. It is important to underline that the possibility of omitting $Z_n(\beta)$ for finite $\beta$ makes the weights of the last layer independent on each other. Being $\beta$ the only degree of freedom of the Gibbs measure, the problem reduces to the study of the existence of an optimal parameter $\beta^*$ (and thus, an optimal inverse temperature of the system) for which, we conjecture, the ensemble manifests an optimal prediction quality (with respect to the loss function $\mathcal{L}$), possibly unpredictable from the observation of single individuals from the population of classifiers, exhibiting therefore an emerging behavior. To motivate (and formalize) this idea, we propose — firstly — a conceptual experiment; consider, in fact, the case where the number of random classifiers is infinite, and thus we obtain

$$\lim_{n\to+\infty}\mathrm{sign}\left(\frac{1}{Z_n(\beta)}\sum_i^n \exp(-\beta\mathcal{L}(\hat{y}_i))\hat{y}_i\right) = \mathrm{sign}\left(\frac{1}{Z_\infty(\beta)}\int_{\mathbb{R}^d}\mu(d\mathbf{w})\exp(-\beta\mathcal{L}(\phi(\mathbf{w}_i)))\phi(\mathbf{w}_i)\right)$$

with $\mu$ being the probability measure over the weights $\mathbf{w}$, which we defined as normally distributed and $Z_\infty(\beta) = \int_{\mathbb{R}^n}\mu(d\mathbf{w})\exp(-\beta\mathcal{L}(\phi(\mathbf{w}_i)))$. Hence, being $\mathrm{supp}(\mu) = \mathbb{R}^d$ we obtain the following equality based on the limit of the softmin [27] function

$$\lim_{\beta\to+\infty}\lim_{n\to+\infty}\mathrm{sign}\left(\frac{1}{Z_n(\beta)}\sum_i^n \exp(-\beta\mathcal{L}(\hat{y}_i))\hat{y}_i\right) = \phi\left(\operatorname*{argmin}_{\mathbf{w}\in\mathrm{supp}(\mu)}\mathcal{L}(\phi(\mathbf{w}))\right) = \phi(\mathbf{w}_*) = y_*$$

meaning that, in the infinite classifiers scenario $\beta^* = +\infty$, since $\mathbf{w}_* \in \mathrm{supp}(\mu)$ and the Gibbs measure converges to a Dirac Delta centered in the global minimizer. Nonetheless, in a context when $n$ is instead finite and thus we observe only a sample from the population of classifiers the probability measure $\mu$ is approximated by $\hat{\mu}$ given by

$$\hat{\mu}(\mathbf{w}) = n^{-1}\sum_i^n \delta(\mathbf{w}-\mathbf{w}_i)$$

which, unlike $\mu$, has the zero measure support $\mathrm{supp}(\hat{\mu}) = \bigcup_{i=1}^n\{\mathbf{w}_i\}$ with therefore $0$ probability of containing $\mathbf{w}_*$ and with

$$\lim_{\beta\to+\infty}\mathrm{sign}\left(\frac{1}{Z_n(\beta)}\sum_i^n \exp(-\beta\mathcal{L}(\hat{y}_i))\hat{y}_i\right) = \phi\left(\operatorname*{argmin}_{\mathbf{w}\in\{\mathbf{w}_i\}_{i=1}^n}\mathcal{L}(\phi(\mathbf{w}_i))\right).$$

The implication is direct; if in the infinite classifiers scenario the $\beta \to \infty$ limit retrieves the global minimizer, in the finite case the minimizer of the sample is obtained. On the other hand, if we consider the curve

$$\gamma(\beta) = \mathrm{sign}\left(\frac{1}{Z_n(\beta)}\sum_i^n \exp(-\beta\mathcal{L}(\hat{y}_i))\hat{y}_i\right) \qquad \text{with } \beta \in [0, \infty)$$

we note that it is defined as a $1-$dimensional curve in the functional space $\{f : \mathbb{R}^d \to \{-1, 1\}\}$, and that its trajectory lies in the image of the convex hull of the set of the sampled classifiers through the $\mathrm{sign}$ operator, and it intersects the set of sampled classifiers for $\beta \to \infty$ in its minimizer as we explained before. Due to this intersection, we obtain the following upper bound

$$\min_\beta \mathcal{L}(\gamma(\beta)) \leq \min_i \mathcal{L}(\phi(\mathbf{w}_i)) = \lim_{\beta\to\infty}\mathcal{L}(\gamma(\beta)),$$

suggesting that, unless $\mathcal{L}(\gamma(\beta))$ is monotonically decreasing with respect to $\beta$, it should exists a minimizer $\beta^*$ for which the ensemble classification outperforms the best classifier in the ensemble. The intuition behind our analysis relies in the idea that if $\beta$ acts modulating the amount of information

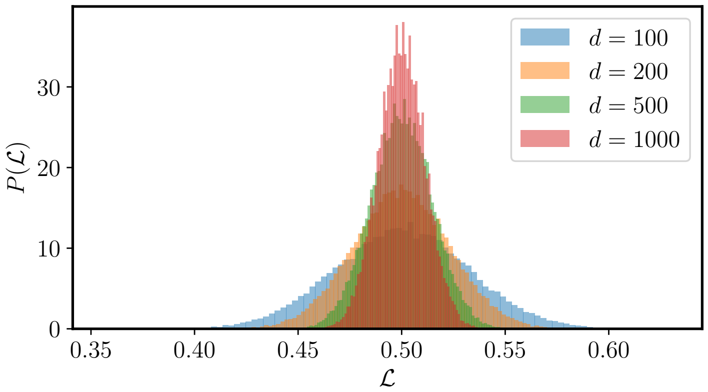


Figure 2: Distribution of the composition of the loss function and the random variable $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, as the dimensions increase it becomes exponentially harder to sample classifier that significantly outperform the average case

in each classifier that enters the ensemble classification (with the already discussed trivial behavior for $\beta \to \infty$ and the agnostic average for $\beta \approx 0$), the ensemble could still benefit — in terms of accuracy — from employing information contained in every classifier, rather than neglecting suboptimal ones (where optimality is defined adopting the sample as domain).

Following a semi-empirical approach, a first numerical experiment is conducted to evaluate if the model exhibits a non-trivial minimum in its loss for a finite $\beta$. The experiment is prepared by considering a random dataset of $N$ samples defined as

$$D = \{(\mathbf{x}_i, y_i)_i\}_{i=1}^N \qquad \text{with } \mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d) \text{ and } y_i = \text{sign}(\mathbf{w}_*^T \mathbf{x}),$$

with $\mathbf{w}_*$ a randomly selected teacher vector, and then constructing the ensemble of random perceptrons by sampling the Gaussian random matrix $\mathbf{W} \in \mathbb{R}^{n \times d}$. To understand the challenging nature of the problem of performing correct classification through randomization, we perform, prior to the experiment, a preliminary study of the distribution of the loss function among the different classifiers.

The result of the simulation can be observed in Figure 2, where the distribution of the images of random parameters of the perceptron through the loss function is studied. The observed concentration phenomenon (with the empirical estimate of the variance vanishing for large $d$) suggests that in a infinite-dimensional scenario every random classifier is agnostic with respect to the loss function, concentrating steeply around the average case. Connecting to our discussion on the behavior of the ensemble for $\beta \to +\infty$, the implication of this property is that in a high dimensional but finite classifiers case we expect that the ensemble behaves equally for $\beta = 0$ and $\beta = +\infty$, with both cases performing a classification equivalent — in quality — to a random guess. To investigate the behavior of the system for finite values of $\beta$, we analyzed the ensemble loss for different values of $\beta$, in order to study the image (hereafter referred as *loss profile*) of the curve $\gamma$ through the loss function.

The result of the experiment can be observed in Figure 3. We note some interesting facts on the behavior of the ensemble; first, one may notice that the loss profile appears to be "smooth", with a clearly continuous structure, with vanishing fluctuations, a clear minimum at $\beta \approx 100$ and thus with a trend that is perfectly consistent with our expectations. As a second, remarkable, point the ensemble shows generalization abilities; the network appears to be able to generalize also on unseen data, despite of the high dimensional geometry of Isotropic Gaussian data (which, due to the full-rankness of its covariance, has its support on the whole $\mathbb{R}^d$), with the position of the minimum coinciding for both test and training data (note that in this context the term "training set" is an abuse of notation, since data are not actively used in a real training phase but only employed in the direct construction of the last layer). A third, interesting aspect of our simulation, is underlined by the emergence of quality in the classification despite the high error of the best classifier in the ensemble ($\mathcal{L}(\mathbf{w}_{best}) \approx 0.45$). This last property, is in our opinion, at the core of our work; how could we justify this emerging characteristics in an ensemble of *completely random classifiers*? Under *which conditions* such emerging behavior manifests? To uncover the nature of the phenomenon a further analysis was conducted on $\beta^*$ under the assumption that a characterization of the optimal inverse

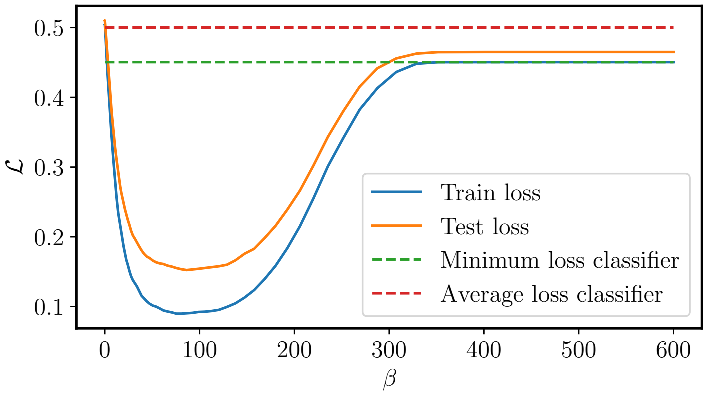
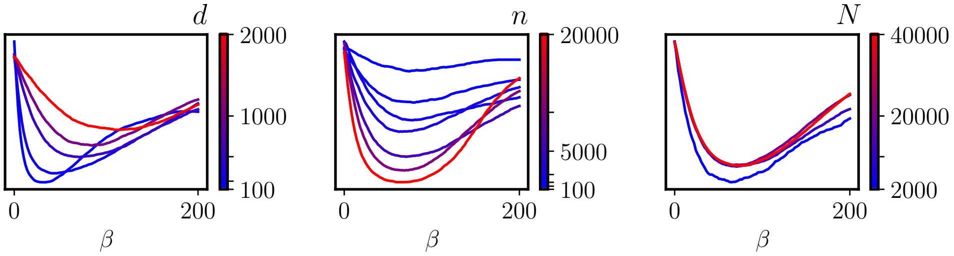


Figure 3: Loss of the ensemble for different values of $\beta$. The profile of the function shows a clear structure with a distinct minimmum, which significantly overcomes the concentration of the loss in high dimensional data.



(a) Loss profile for different dimensionalities

(b) Loss profile for different number of classifiers

(c) Loss Profile for different sizes of the "Training" set

Figure 4: Structure of the loss profiles in function of perturbations of $d$, $n$ and $N$. The color scale indicates an increase of the fixed value from blue to red. The position of the optimal inverse temperature appears to be influenced solely by $d$

temperature could reveal the intrinsic mechanics of the model. Thus, a second experiment was conducted by numerically estimating $\beta^*$ for different parameters of the model (varying, therefore, the number of classifiers $n$, the dimensionality of data $d$ and the number $N$ of sample that are observed).

Figure 4 shows the result of the experiment. Each trial is performed by varying a specific quantity (such as $d$) while keeping the other two fixed (such as $n$, $N$). The empirical evidence suggests that the position of the minimum is not influenced neither by the number of classifiers $n$ nor by the number of datapoints $N$, depending solely on the number of dimensions of the problem $d$. On the other hand, the number of classifiers shows a clear impact on the quality of the classification, highlighting that what we denoted as "emergence of structure" is naturally more relevant for large ensembles of classifiers. In light of these considerations, a final numerical experiment was constructed to study wether the specific choice of the teacher vector $\mathbf{w}_*$ could have some effect on the position and the quality of the minimum. We generate a single dataset (with a given seed) of datapoints $\mathbf{X}$ and a quenched set of classifier described by the random matrix $\mathbf{W}$ and then generate different teacher vectors, one for each different seed, following a multivariate normal distribution (which we remember, lead to teachers that are uniform in direction). The resulting loss profile is obtained, for each tuple $(\mathbf{X}, \mathbf{W}, \mathbf{w}_*^{(i)})$.

In Figure 5 we see the result of the simulation for 10 different teacher vectors. What we can conclude, from our experiments, is that while different factors may impact the shape of the loss profile, the position of the minimum (which is the inverse temperature $\beta$ at which optimal "learning" happens) appears to be depending only on the number of dimensions $d$. Moreover, in Figure 5 we observe that the quality of the solution obtained by reaching the minimum appears to be invariant with respect to

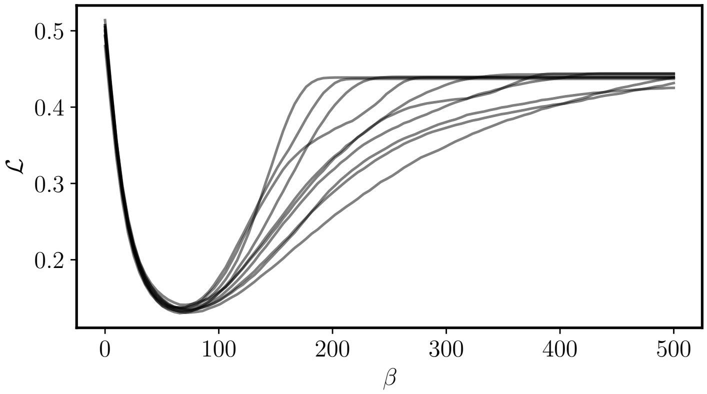


Figure 5: Profile of the loss for different teachers classifiers. Despite of the fluctuations, the position of the minimum is virtually constant among the different simulations

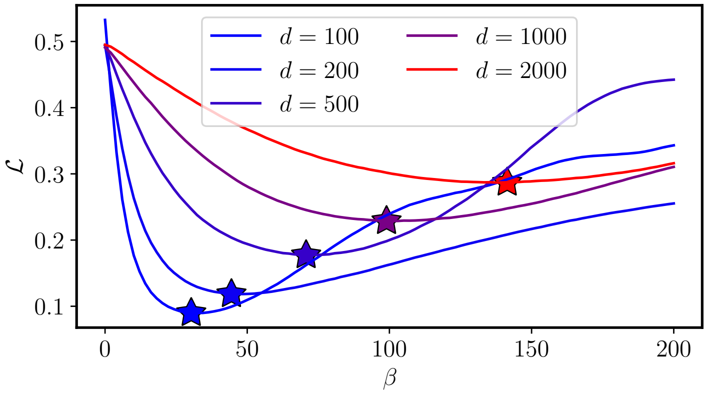


Figure 6: Predictions ($\star$) of $\beta^*$ using the numerical maximization of the function presented in 2, for different values of $d$. The model appears to be able to predict the position of the minimum on the loss curve

the structure of the teacher classifier, suggesting that the phenomenon is disentangled from the nature of the specific task that has been learned by the model.

The experimental result can be further validated by the construction of an asymptotic model of the loss profile, that may be obtained by means of the Central Limit Theorem, which explains the loss of the model for a given value of $\beta$.

A demonstration (see Appendix A for further details on the construction of this result) shows in fact that the position of the optimal $\beta$ for classification is approximated by the maximizer of the real function

$$\xi(\beta) = \frac{\exp\left\{\frac{1}{2}\left(\frac{\beta}{\pi\sqrt{d-2}}\right)^2\right\}\beta\left(\frac{1}{\pi\sqrt{d-2}}\right)^2}{\sqrt{\exp\left(\frac{2\beta^2}{\pi^2(d-2)}\right) - 4\exp\left\{\left(\frac{\beta}{\pi\sqrt{d-2}}\right)^2\right\}\beta^2\left(\frac{1}{\pi\sqrt{d-2}}\right)^4}}, \tag{2}$$

which depends solely on $d$, confirming the empirical evidence of independence (up to fluctuations) of the optimal inverse temperature for learning from the structure of the learning task.

A further simulation was performed to confirm our analysis, as shown in Figure 6. For each considered dimensionality, the maximizer of $\xi$ (computed through a grid search algorithm) and the empirical loss profile are compared. Despite the approximate nature of the proposed model, the reconstruction appears to be consistent with the empirical observation. Interestingly, a detailed computation (described briefly in Appendix B) shows that the maximization of $\xi$ admits the closed form

$$\beta^* = \pi\sqrt{d-2},$$

which provides a solution for the optimal learning inverse temperature in the Gaussian case.

## 4 Numerical experiments on the `MNIST` dataset

Naturally, the behavior of our model on Gaussian data represents a very specific scenario in terms of data geometry. The adoption of an isotropic model of data forces, in fact, symmetries that may be non optimal or even unnatural in describing the structure of real world data. In light of this consideration, it is interesting to investigate wether the proposed model is able or not to provide a realistic description of the behavior of the Random Neural Network with organic data, which is often, by contrast, highly structured [28, 29].

Hence, a further numerical investigation was performed on the `MNIST` dataset. Such dataset offers an interesting playground for testing our hypothesis; the data is, in fact, highly structured, providing well defined clusters [30] and a low intrinsic dimension [31]. In our experiment we constructed $N = 100$ random classifiers, using as a target vector the parity of the digit ($y_i = +1$ if $\mathbf{x}_i$ represents an even digit, otherwise $y = -1$ if the represented digit is odd). We employed as loss function the 0-1 Loss, as we did in the Gaussian model, by estimating the sample average of misclassified samples over the whole training set, composed of 60000 digits.

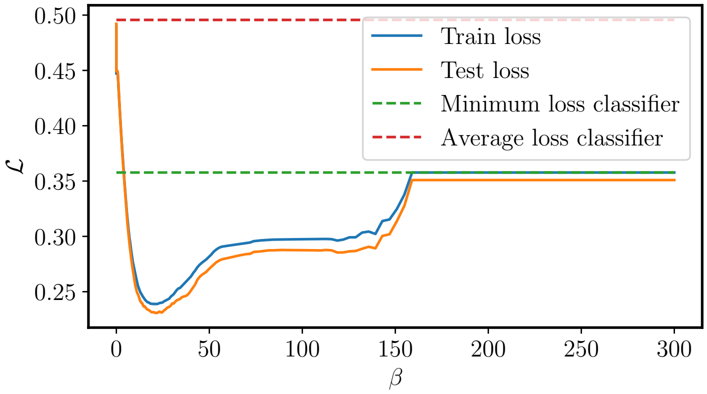


Figure 7: Different values of $\beta$ are considered in this experiment with the `MNIST` dataset. The profile of the loss has a clear minimum in $\beta \approx 20$, which is able to outperform the best classifier

The result of the simulation can be observed in Figure 7. Interestingly, we note a similar structure of the curve with respect to the Gaussian model, with a clear minimum for $\beta \approx 20$.

Furthermore, we note that the ensemble generalizes, as may be noted from the loss profile for training and test loss. We interpret this result as a consequence of the low dimensional structure of the `MNIST` dataset, which also explains the relatively low best classifier loss despite of the high dimensionality of the dataset ($d = 784$).

The experiment suggests that the property that we observed for Gaussian data may be in some sense universal, showing similar features in distributions that differ from the simpler isotropic scenario. However, we still have to verify that the value of the optimal inverse temperature $\beta^*$ does not, approximately, depend on the nature of the training task. To validate this idea we constructed three different problems, with

$$\begin{aligned} y_1(\mathbf{x}) &= +1 \text{ if the digit is even, } -1 \text{ otherwise} \\ y_2(\mathbf{x}) &= +1 \text{ if the digit is less or equal than } 5, -1 \text{ otherwise} \\ y_3(\mathbf{x}) &= \text{sign}(\mathbf{w}_*^T\mathbf{x}) \qquad \mathbf{w}_* \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \end{aligned}$$

The loss profiles are then plotted for the three problems, as shown in Figure 8. Despite of the different nature of the tasks the optimal inverse temperature $\beta^*$ coincide. This represents an important feature

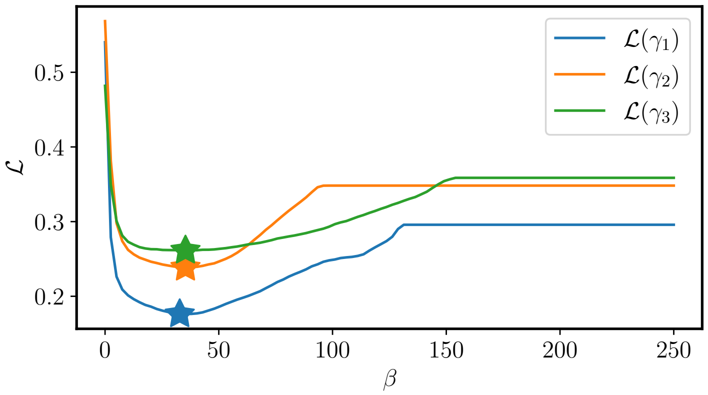


Figure 8: Three different loss profiles based on the MNIST dataset. Stars ($\star$) represent the numerically computed optimal inverse temperature $\beta^*$

of our model, highlighting the connection of the Isotropic case with more general classes of data distributions.

## 5 Physical interpretation, future work and conclusion

It is worth noting that the proposed ensemble admits a meaningful physical interpretation. Suppose in fact that an infinitely large ensemble of non-interacting perceptrons $\{P_i\}_{i=1}^{\infty}$ is employed in the classification. The state $\mathcal{S}(P)$ of each perceptron is represented by the realization of its weights, that can be one of the rows of the quenched random matrix $\mathbf{W}$. If the system is "heated" at temperature $1/\beta$ we could imagine that the number of classifiers whose state is $\mathbf{w}$ is given by the following proportionality relationship

$$\#\{P : \mathcal{S}(P) = \mathbf{w}\} \propto \exp(-\beta\mathcal{L}(\mathbf{w}))$$

with the result becoming more precise as the size of the ensemble of classifiers grows. If in our model one interprets neurons not as individual perceptrons, but rather as representatives of perceptrons in the same state (given by the random realization of the weights) belonging to the larger ensemble then the analytically computed weights represent — in the infinite size limit — the cardinality of the equivalence class associated to each perceptron. This viewpoint on the model highlight how the observed phenomenon could be theoretically treated as an example of self-organization, with the further constraint of non-interaction among agents, a part from the initial consensus on the choice of the random matrix $\mathbf{W}$.

As a final consideration, the theoretical possibility of accelerating the inference has been investigated, showing that under mild assumptions it is possible to merge the predictions by averaging classifiers in the parameter space rather than in the functional space, enabling the construction of a single perceptron classifier that approximates the family of random perceptrons (See Appendix D). A further analysis of this idea could be fruitful in the context of constructing algorithms based on our work.

The natural evolution of this work is to analyze the case in which the perceptron is replaced by more sophisticated examples of random neural networks, such as random multilayer perceptrons or even random convolutional layers, to understand if it is possible to observe this emerging mechanism also in richer models. Moreover, we shall analyze also if it could be possible to observe similar results with noisy observation of the loss function (for instance, when the sample mean is computed on batches of the dataset) to move our theoretical analysis into a more practical algorithm. We also argue that this example could be significative in stimulating novel techniques of pretraining for larger models based on the injection of a covariance structure into the random parameters of the network. Finally, we want to highlight that the computational cost associated to the analytical construction of weights may be very high for traditional parallel computing systems.

We believe that the methods, which was now proposed under a theoretical lens, could become an effective training method in the context of a dedicated hardware architecture able to exploit the embarrassingly parallel nature of our proposal.

## References

[1] W. Cao, X. Wang, Z. Ming, and J. Gao, "A review on neural networks with random weights," *Neurocomputing*, vol. 275, pp. 278–287, 2018.

[2] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[3] R. Couillet and Z. Liao, *Random matrix methods for machine learning*. Cambridge University Press, 2022.

[4] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[5] A. Engel, *Statistical mechanics of learning*. Cambridge University Press, 2001.

[6] S. K. Kumar, "On weight initialization in deep neural networks," *arXiv preprint arXiv:1704.08863*, 2017.

[7] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone, "A review on weight initialization strategies for neural networks," *Artificial intelligence review*, vol. 55, no. 1, pp. 291–322, 2022.

[8] Y. Le Cun, I. Kanter, and S. A. Solla, "Eigenvalues of covariance matrices: Application to neural-network learning," *Physical review letters*, vol. 66, no. 18, p. 2396, 1991.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.*, "Highly accurate protein structure prediction with alphafold," *nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[11] G. Jin, X. Yi, L. Zhang, L. Zhang, S. Schewe, and X. Huang, "How does weight correlation affect generalisation ability of deep neural networks?," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21346–21356, 2020.

[12] C. H. Martin and M. W. Mahoney, "Heavy-tailed universality predicts trends in test accuracies for very large pre-trained deep neural networks," in *Proceedings of the 2020 SIAM International Conference on Data Mining*, pp. 505–513, SIAM, 2020.

[13] J. L. Balcázar, R. Gavalda, and H. T. Siegelmann, "Computational power of neural networks: A characterization in terms of kolmogorov complexity," *IEEE Transactions on Information Theory*, vol. 43, no. 4, pp. 1175–1183, 2002.

[14] M. Mézard, "Spin glass theory and its new challenge: structured disorder," *Indian Journal of Physics*, vol. 98, no. 11, pp. 3757–3768, 2024.

[15] P. W. Anderson, "More is different: Broken symmetry and the nature of the hierarchical structure of science.," *Science*, vol. 177, no. 4047, pp. 393–396, 1972.

[16] H.-D. Block, "The perceptron: A model for brain functioning. i," *Reviews of Modern Physics*, vol. 34, no. 1, p. 123, 1962.

[17] K. Huang, *Statistical mechanics*. John Wiley & Sons, 2008.

[18] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[19] M. Yan, C. Huang, P. Bienstman, P. Tino, W. Lin, and J. Sun, "Emerging opportunities and challenges for the future of reservoir computing," *Nature Communications*, vol. 15, no. 1, p. 2056, 2024.

[20] J. Wang, S. Lu, S.-H. Wang, and Y.-D. Zhang, "A review on extreme learning machine," *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 41611–41660, 2022.

[21] K. Fukushima, "Visual feature extraction by a multilayered network of analog threshold elements," *IEEE Transactions on Systems Science and Cybernetics*, vol. 5, no. 4, pp. 322–333, 1969.

[22] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, "On random weights and unsupervised feature learning.," in *Icml*, vol. 2, p. 6, 2011.

[23] D. Cox and N. Pinto, "Beyond simple features: A large-scale feature search approach to unconstrained face recognition," in *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pp. 8 – 15, 04 2011.

[24] R. Giryes, G. Sapiro, and A. M. Bronstein, "Deep neural networks with random gaussian weights: A universal classification strategy?," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3444–3457, 2016.

[25] C. Gallicchio and S. Scardapane, "Deep randomized neural networks," in *Recent Trends in Learning From Data: Tutorials from the INNS Big Data and Deep Learning Conference (INNSBDDL2019)*, pp. 43–68, Springer, 2020.

[26] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[27] B. Gao and L. Pavel, "On the properties of the softmax function with application in game theory and reinforcement learning," *arXiv preprint arXiv:1704.00805*, 2017.

[28] A. Sclocchi, A. Favero, and M. Wyart, "A phase transition in diffusion models reveals the hierarchical nature of data," *Proceedings of the National Academy of Sciences*, vol. 122, no. 1, p. e2408799121, 2025.

[29] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review," *International Journal of Automation and Computing*, vol. 14, no. 5, pp. 503–519, 2017.

[30] M. Hein and J.-Y. Audibert, "Intrinsic dimensionality estimation of submanifolds in rd," in *Proceedings of the 22nd international conference on Machine learning*, pp. 289–296, 2005.

[31] S. Spigler, M. Geiger, and M. Wyart, "Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2020, no. 12, p. 124001, 2020.

[32] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*, vol. 47. Cambridge university press, 2018.

## Appendix A Analytical model for the optimal $\beta$

Consider the infinite data (that means that we neglect the fluctuations of the loss over variations of the training set), high (but finite) $d, n$ scenario, with labels generated according to the sign of the dot product against a teacher vector $\mathbf{w}_* \in \mathbb{R}^d$ with $||\mathbf{w}_*|| = 1$. Let $\mathbf{W}$ be a Gaussian random matrix and $\mathbf{x}$ a Gaussian random vector that models the data population. The expectation (that we denote with $\bar{L}$), with respect to realizations of the random matrix $\mathbf{W}$, of the 0-1 loss of the ensemble (that we denote with $L$) can be written as (with $\mathbb{1}$ intended as a function that maps a logical statement into its truth value, encoded with 1 for true and 0 for false)

$$\bar{L} := \mathbb{E}_{\mathbf{x},\mathbf{W}} \mathbb{1}\left[\operatorname{sign}\left(\sum_i^n \exp\left(-\beta \mathbb{E}_{\mathbf{x}'}\mathbb{1}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}') \neq \operatorname{sign}(\mathbf{w}_i^T\mathbf{x}')\right]\right)\operatorname{sign}(\mathbf{x}^T\mathbf{w}_i)\right) \neq \operatorname{sign}(\mathbf{x}^T\mathbf{w}_*)\right]$$

with the adoption of the expectation with respect to $\mathbf{x}$ and $\mathbf{x}'$ (which are identically distributed, as isotropic multivariate normal random vectors) in place of the sample mean with respect the observed data points thanks to the infinite $N$ assumption. Note that also individual perceptrons $\mathbf{x} \mapsto \mathbf{w}_i^T\mathbf{x}$ employ the 0-1 loss to quantify their risk measure. We aim, in this context, to build a mathematical model to describe the optimal value of $\beta$, such that the classification of the ensemble is optimal respect to the 0-1 loss, with the further challenge that such loss is not differentiable by definition. We start our argument by observing that

$$\mathbb{1}[x \neq y] = -\frac{1}{2}xy + \frac{1}{2} \qquad \forall x, y \in \{-1, 1\}$$

and we can thus rewrite

$$\bar{L} = -\frac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{W}}\left[\operatorname{sign}\left\{\sum_i^n \exp\left(-\beta\mathbb{E}_{\mathbf{x}'}\left[-\frac{1}{2}\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}')\operatorname{sign}(\mathbf{w}_i^T\mathbf{x}') + \frac{1}{2}\right]\right) \times \right.\right.$$
$$\left.\left. \times \operatorname{sign}(\mathbf{x}^T\mathbf{w}_i)\right\}\operatorname{sign}(\mathbf{x}^T\mathbf{w}_*)\right] + \frac{1}{2}.$$

By a substitution ($\tilde{\beta} = \beta/2$) and an affine transformation $x \mapsto 2(-x + 1/2)$ we can rewrite, casting the problem from a minimization to a maximization (note that affine transformations cannot alter the number or the position of critical points), as

$$-\bar{L} \propto \mathbb{E}_{\mathbf{x},\mathbf{W}}\left[\operatorname{sign}\left(\sum_i^n \exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w}_i)\right]\right)\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}\mathbf{x}^T\mathbf{w}_i)\right)\right],$$

which can be rewritten, employing the identity $\operatorname{sign}(x) = 2 \cdot \mathbb{1}_{\geq}(x) - 1$ that holds for every $x \neq 0$, as

$$-\bar{L} \propto \mathbb{E}_{\mathbf{x},\mathbf{W}}\mathbb{1}\left[\frac{1}{n}\sum_i^n \exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w}_i)\right]\right)\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}\mathbf{x}^T\mathbf{w}_i) \geq 0\right]$$

and employing the equality $\mathbb{E}[\mathbb{1}_\Omega(x)] = \mathbb{P}[x \in \Omega]$ we can rewrite

$$-\bar{L} \propto \mathbb{P}_{\mathbf{x},\mathbf{W}}\left[\frac{1}{n}\sum_i^n \exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w}_i)\right]\right)\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}\mathbf{x}^T\mathbf{w}_i) \geq 0\right].$$

Let us, therefore, define the family of random variables $\{Y_i\}_{i=1}^N$ as

$$Y_i = \exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w}_i)\right]\right)\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}\mathbf{x}^T\mathbf{w}_i) \;\; \forall i \in [N]$$

and, since $\mathbf{w}_i \overset{iid}{\sim} \mathbf{w}_j \; \forall i, j$ (by construction of the random matrix $\mathbf{W}$), the family $\{Y_i\}_{i=1}^N$ can be considered as a set of identically distributed random variables, copies of $Y := Y_1$.

We can therefore study the sum as a normally distributed random variable, thanks to the Central Limit Theorem that states the convergence of distribution of the sum

$$\lim_{n\to\infty}\frac{1}{n}\sum_i^n Y_i \sim \mathcal{N}\left(\mathbb{E}_{\mathbf{x},\mathbf{w}}Y, \frac{1}{n}\operatorname{Var}_{\mathbf{x},\mathbf{w}} Y\right),$$

which validates the normal approximation of $n^{-1}\sum_i^n Y_i$ for large numbers of classifiers. To describe the distribution of such sum, we need to describe its two parameters. The mean, in particular, is given by

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{x},\mathbf{w}}[Y] = \\
&= \mathbb{E}_{\mathbf{x},\mathbf{w}} \exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w})\right]\right)\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}\mathbf{x}^T\mathbf{w}) \\
&= \mathbb{E}_{\mathbf{w}}\Bigg\{ \exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w})\right]\right) \times \\
&\qquad\qquad \times \mathbb{E}_{\mathbf{x}}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}\mathbf{x}^T\mathbf{w})\right]\Bigg\} \\
&= \mathbb{E}_{\mathbf{w}}\left\{\partial_{\tilde{\beta}}\exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w})\right]\right)\right\} && \partial_\alpha\exp(\alpha x) = \exp(\alpha x)x \\
&= \partial_{\tilde{\beta}}\mathbb{E}_{\mathbf{w}}\left\{\exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w})\right]\right)\right\} && \text{Dominated Convergence Theorem.}
\end{aligned}
$$

We can study the argument of the exponential in an analytic way employing the Grothendieck's Identity, which explains the expectation of the left and right product against the rank 1 matrix given by the outer product of a Gaussian vector against itself by means of the cosine similarity between the left and the right multiplying vector (remember that, by Hypothesis, $||\mathbf{w}_*|| = 1$)

$$
= \partial_{\tilde{\beta}}\mathbb{E}_{\mathbf{w}}\left\{\exp\left(\tilde{\beta}\frac{2}{\pi}\arcsin\left(\frac{1}{||\mathbf{w}||}\mathbf{w}_*^T\mathbf{w}\right)\right)\right\} \qquad \text{Grothendieck's Identity [32].}
$$

We then linearize the $\arcsin$ function by noting that the variance of its argument vanishes in higher dimensions, allowing us to approximate the trigonometric function with its (linear) behavior around the origin. We then employ a concentration result relative to the inverse norm of a Gaussian vector to obtain a normally distributed argument of the exponential ($Z$ in this context denotes $Z \sim \mathcal{N}(0,1)$ and should be not confused with the definition of the partition function $Z_n$ that we employed in the introduction of our model). We remand to Appendix C for further details on the logic behind our approximation.

$$
\begin{aligned}
&\approx \partial_{\tilde{\beta}}\mathbb{E}_{\mathbf{w}}\left\{\exp\left(\tilde{\beta}\frac{2}{\pi}\frac{1}{||\mathbf{w}||}\mathbf{w}_*^T\mathbf{w}\right)\right\} \\
&\approx \partial_{\tilde{\beta}}\mathbb{E}_{\mathbf{w}}\left\{\exp\left(\tilde{\beta}\frac{2}{\pi}\mathbb{E}\left[\frac{1}{||\mathbf{w}||}\right]\mathbf{w}_*^T\mathbf{w}\right)\right\} \\
&\approx \partial_{\tilde{\beta}}\mathbb{E}_{\mathbf{w}}\left\{\exp\left(\tilde{\beta}\frac{2}{\pi\sqrt{d-2}}\mathbf{w}_*^T\mathbf{w}\right)\right\} && \text{See Appendix C} \\
&= \partial_{\tilde{\beta}}\mathbb{E}_{Z}\left\{\exp\left(\tilde{\beta}\frac{2}{\pi\sqrt{d-2}}Z\right)\right\} && \mathbf{w}^T\mathbf{w}_* \sim \mathcal{N}(0, ||\mathbf{w}_*||^2),\ ||\mathbf{w}_*|| = 1.
\end{aligned}
$$

Finally, we can employ the well known Moment Generating Function of the Normal distribution to construct an analytic expression of the mean of the distribution

$$
\begin{aligned}
&= \partial_{\tilde{\beta}}\exp\left\{\frac{1}{2}\left(\tilde{\beta}\frac{2}{\pi\sqrt{d-2}}\right)^2\right\} && \text{M.G.F of the Normal distribution} \\
&= \exp\left\{\frac{1}{2}\left(\tilde{\beta}\frac{2}{\pi\sqrt{d-2}}\right)^2\right\}\tilde{\beta}\left(\frac{2}{\pi\sqrt{d-2}}\right)^2 \\
&=: \mu.
\end{aligned}
$$

The variance, instead, is obtained — with the same theoretical tools — as

$$
\begin{aligned}
\mathrm{Var}_{\mathbf{x},\mathbf{w}}[Y] &= \mathbb{E}_{\mathbf{x},\mathbf{w}}[Y^2] - \mu^2 \\
&= \mathbb{E}_{\mathbf{x},\mathbf{w}} \left\{ \exp\left( 2\tilde{\beta}\mathbb{E}_{\mathbf{x}'} \left[ \mathrm{sign}(\mathbf{w}_*^T \mathbf{x}' \mathbf{x}'^T \mathbf{w}) \right] \right) \left( \mathrm{sign}(\mathbf{w}_*^T \mathbf{x}\mathbf{x}^T \mathbf{w}) \right)^2 \right\} - \mu^2 \\
&= \mathbb{E}_{\mathbf{w}} \left\{ \exp\left( 2\tilde{\beta}\mathbb{E}_{\mathbf{x}'} \left[ \mathrm{sign}(\mathbf{w}_*^T \mathbf{x}' \mathbf{x}'^T \mathbf{w}) \right] \right) \right\} - \mu^2 \\
&\approx \mathbb{E}_{\mathbf{w}} \left\{ \exp\left( 2\tilde{\beta} \frac{2}{\pi\sqrt{d-2}} \mathbf{w}_*^T \mathbf{w} \right) \right\} - \mu^2 && \text{See Appendix C} \\
&= \exp\left( \frac{1}{2} \left( 2\tilde{\beta} \frac{2}{\pi\sqrt{d-2}} \right)^2 \right) - \mu^2 \\
&= \exp\left( \frac{1}{2} \frac{16\tilde{\beta}^2}{\pi^2(d-2)} \right) - \mu^2 \\
&= \exp\left( \frac{8\tilde{\beta}^2}{\pi^2(d-2)} \right) - \mu^2 \\
&=: \sigma^2.
\end{aligned}
$$

Thus, we can study the probability of positive value of the average among $Y_1, ..., Y_n$ as

$$
\begin{aligned}
\mathbb{P}_{\mathbf{Y}} \left[ \frac{1}{n} \sum_i Y_i \geq 0 \right] &\approx \mathbb{P}_Z \left[ \mu + \frac{1}{\sqrt{n}} \sigma Z \geq 0 \right] \\
&= \mathbb{P}_Z \left[ Z \geq -\frac{\mu}{\sigma} \sqrt{n} \right] \\
&= \mathbb{P}_Z \left[ -Z \leq \frac{\mu\sqrt{n}}{\sigma} \right] \\
&= \Phi\left( \frac{\mu\sqrt{n}}{\sigma} \right),
\end{aligned}
$$

where $\Phi$ is the cumulative density function of the normal distribution. Thus we get

$$
-\bar{L} \propto \Phi\left( \sqrt{n} \frac{\exp\left\{ \frac{1}{2} \left( \tilde{\beta} \frac{2}{\pi\sqrt{d-2}} \right)^2 \right\} \tilde{\beta} \left( \frac{2}{\pi\sqrt{d-2}} \right)^2}{\sqrt{\exp\left( \frac{8\tilde{\beta}^2}{\pi^2(d-2)} \right) - \left( \exp\left\{ \frac{1}{2} \left( \tilde{\beta} \frac{2}{\pi\sqrt{d-2}} \right)^2 \right\} \tilde{\beta} \left( \frac{2}{\pi\sqrt{d-2}} \right)^2 \right)^2}} \right)
$$

and after applying conversely $\beta = 2\tilde{\beta}$

$$
-\bar{L} \propto \Phi\left( \sqrt{n} \frac{\exp\left\{ \frac{1}{2} \left( \frac{\beta}{2} \frac{2}{\pi\sqrt{d-2}} \right)^2 \right\} \frac{\beta}{2} \left( \frac{2}{\pi\sqrt{d-2}} \right)^2}{\sqrt{\exp\left( \frac{8\left(\frac{\beta}{2}\right)^2}{\pi^2(d-2)} \right) - \left( \exp\left\{ \frac{1}{2} \left( \frac{\beta}{2} \frac{2}{\pi\sqrt{d-2}} \right)^2 \right\} \frac{\beta}{2} \left( \frac{2}{\pi\sqrt{d-2}} \right)^2 \right)^2}} \right)
$$

and since the function $\Phi$ is monotonically increasing we obtain the final result

$$
\beta^* = \operatorname*{argmax}_{\beta} \xi(\beta) = \operatorname*{argmax}_{\beta} \frac{\exp\left\{ \frac{1}{2} \left( \frac{\beta}{\pi\sqrt{d-2}} \right)^2 \right\} \beta \left( \frac{1}{\pi\sqrt{d-2}} \right)^2}{\sqrt{\exp\left( \frac{2\beta^2}{\pi^2(d-2)} \right) - 4\exp\left\{ \left( \frac{\beta}{\pi\sqrt{d-2}} \right)^2 \right\} \beta^2 \left( \frac{1}{\pi\sqrt{d-2}} \right)^4}} \tag{3}
$$

that leads to the minimization of the expected loss $\bar{L}$.

## Appendix B Analytical construction of the maximizer of $\xi$

The derivative of $\xi(\beta)$ with respect to $\beta$ is given by the expression

$$\partial_\beta \xi(\beta) = \frac{\left(2\pi^2 d - 2\beta^2 - 4\pi^2\right)}{\left((\pi^4 d^2 - 4\pi^4 d + 4\pi^4)\exp\left(\frac{\beta^2}{\pi^2 d - 2\pi^2}\right) - 4\beta^2\right)} \times$$
$$\times \frac{\sqrt{\pi^4 d^2 - 4\pi^4 d + 4\pi^4}\exp\left(\frac{\beta^2}{2\pi^2 d - 4\pi^2} + \frac{\beta^2}{\pi^2 d - 2\pi^2}\right)}{\sqrt{(\pi^4 d^2 - 4\pi^4 d + 4\pi^4)\exp\left(\frac{2\beta^2}{\pi^2 d - 2\pi^2}\right) - 4\beta^2 \exp\left(\frac{\beta^2}{\pi^2 d - 2\pi^2}\right)}},$$

which, due to the factor $2(\pi^2 d - 2\pi^2 - \beta^2)$, is 0 for

$$\beta = \pm\sqrt{\pi^2(d-2)} = \pm\pi\sqrt{d-2}.$$

Since we are constrained to the case $\beta > 0$, the solution becomes

$$\beta^* = \pi\sqrt{d-2},$$

which is experimentally compatible with minima in the loss profiles for Gaussian data.

## Appendix C Linearization of the $\arcsin$ and approximation of the uniform distribution over the sphere

In Appendix A we employed the approximation

$$\mathbb{E}_{\mathbf{w}}\left\{\exp\left(\tilde{\beta}\frac{2}{\pi}\arcsin\left(\frac{1}{||\mathbf{w}||}\mathbf{w}_*^T\mathbf{w}\right)\right)\right\} \approx \mathbb{E}_{\mathbf{w}}\left\{\exp\left(\tilde{\beta}\frac{2}{\pi\sqrt{d-2}}\mathbf{w}_*^T\mathbf{w}\right)\right\}$$

where the random vector $\mathbf{w}$ is assumed to be normally distributed with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. The first step to understand the approximation is the study of the random variable

$$A := \frac{1}{||\mathbf{w}||}\mathbf{w}_*^T\mathbf{w}.$$

Since $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ we obtain that

$$\mathbf{w}_*^T\mathbf{w} \sim \mathcal{N}\left(\mathbf{0}, ||\mathbf{w}_*||^2\right),$$

but being the teacher vector of unitary norm with $||\mathbf{w}_*|| = 1$ the scalar product behaves as a normal random variable. To study the distribution of $A$ it is still necessary, however, to understand the impact of the term $\frac{1}{||\mathbf{w}||}$ to the statistics. If $\mathbf{w}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ then, defining the fluctuation as

$$F(\mathbf{w}_d) = \left|\frac{1}{||\mathbf{w}_d||} - \mathbb{E}\left[\frac{1}{||\mathbf{w}_d||}\right]\right|, \tag{4}$$

we claim that

$$\lim_{d\to\infty}\mathbb{E}\left[\frac{F(\mathbf{w}_d)}{\mathbb{E}\left[\frac{1}{||\mathbf{w}_d||}\right]}\right] = 0,$$

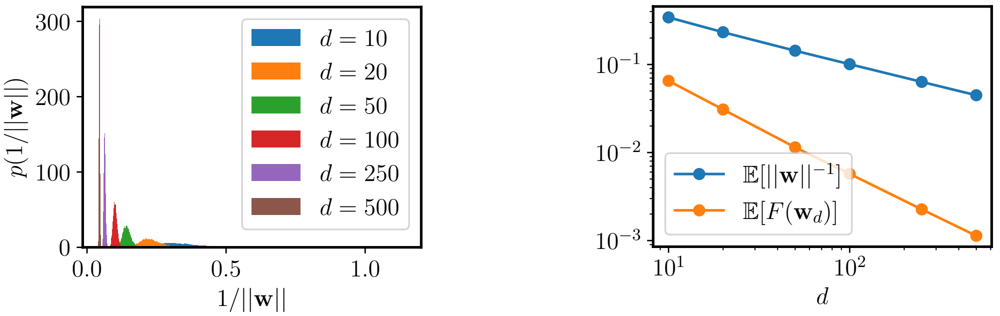
meaning that the expected fluctuation around the mean of the random variable $\frac{1}{||\mathbf{w}_i||}$ vanishes faster than its mean (as shown in Figure 9 and Figure 10), allowing us to employ, for large $d$, the approximation

$$\frac{1}{||\mathbf{w}_d||} \approx \mathbb{E}\left[\frac{1}{||\mathbf{w}_d||}\right]$$

treating the inverse norm as a constant, or to be more precise as a concentrating quantity.

We can employ the approximation to compute the expectation, by considering

$$\mathbb{E}\left[\frac{1}{||\mathbf{w}_i||}\right] = \mathbb{E}\left[\sqrt{\frac{1}{||\mathbf{w}_i||^2}}\right]$$



(a) Concentration of the distribution of the image of a normal random vector through the inverse norm function

(b) Comparison of the speed of decay of the expected inverse norm $\mathbb{E}||\mathbf{w}_d||^{-1}$ and the speed of decay of the expected fluctuation $\mathbb{E}\left|||\mathbf{w}_d||^{-1} - \mathbb{E}\left[||\mathbf{w}_d||^{-1}\right]\right|$

Figure 9: Numerical study of the concentration properties of the inverse norm of a normal random vector

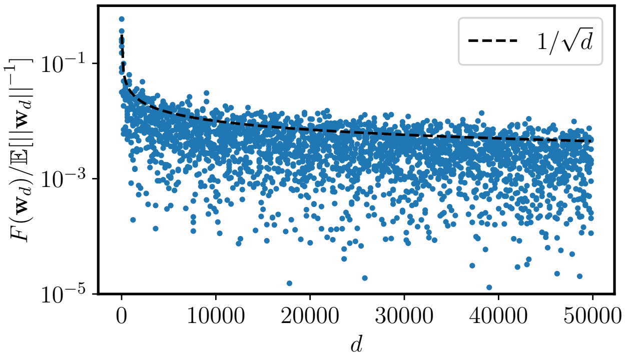


Figure 10: Observed magnitude of the normalized fluctuations of the inverse norm.

and, under the hypothesis of concentration of the probability distribution, since $f\left(\int_{\mathbb{R}^d} d\mathbf{x}\; \delta(\mathbf{x} - \bar{\mathbf{x}})\mathbf{x}\right) = \left(\int_{\mathbb{R}^d} d\mathbf{x}\; \delta(\mathbf{x} - \bar{\mathbf{x}}) f(\mathbf{x})\right)$, then

$$\mathbb{E}\left[\frac{1}{||\mathbf{w}_i||}\right] \approx \sqrt{\mathbb{E}\left[\frac{1}{||\mathbf{w}_i||^2}\right]}$$

and noting that $\frac{1}{||\mathbf{w}_i||^2} \sim \chi^{-2}(d)$ — where $\chi^{-2}(d)$ denotes the inverse chi square distribution with $d$ degrees of freedom — we obtain

$$\mathbb{E}\left[\frac{1}{||\mathbf{w}_i||}\right] \approx \sqrt{\frac{1}{d-2}}.$$

This implies that

$$A \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{d-2}\right)$$

meaning that, for large $d$, the fluctuations around the mean are of order $\mathcal{O}(\sqrt{d^{-1}})$ and allowing us to approximate the arcsin function with its Taylor expansion around the origin

$$\arcsin(A) \approx A$$

with the approximation improving as $d$ grows. This result enforces the idea that

$$\frac{\mathbf{w}_d}{||\mathbf{w}_d||} \sim \mathcal{U}(\mathbb{S}_{d-1}) \approx \mathcal{N}\left(\mathbf{0}, \sqrt{\frac{1}{d-2}}\mathbf{I}_d\right)$$

that is to say, the (scaled) high dimensional normal distribution becomes — when $d$ grows — a valid approximation of the uniform distribution over the unit sphere (which is defined by means of a normalized Gaussian vector). This is a well known fact in the literature of Concentration of Measure [32] and High Dimensional Geometry and was for us a valid tool for constructing our solution of the optimal $\beta$ problem.

## Appendix D Perceptron merging

A similar idea to the one proposed in Appendix C can be employed for averaging the contribution of the perceptrons directly in the parameters space rather than in the functional space where the classifiers exist, imposing that

$$\operatorname{sign}\left(\sum_i^N \alpha_i \phi\left(\mathbf{w}_i\right)\right) \approx \phi\left(\sum_i^N \alpha_i \mathbf{w}_i\right)$$

with the appealing property that

$$\sum_i^N \alpha_i \mathbf{w}_i =: \mathbf{w}_{ens} \in \mathbb{R}^d$$

reducing the parallel workload during the inference phase to the computation of the dot product $\mathbf{x}^T\mathbf{w}_{ens}$. This approximation can be easily obtained by noting that the dot product can be decomposed into the product of two random variables

$$\mathbf{w}^T\mathbf{x} = \operatorname{sign}\left(\mathbf{w}^T\mathbf{x}\right)\left|\mathbf{w}^T\mathbf{x}\right|$$

where we denote

$$\begin{cases} A(\mathbf{x}) \sim \left|\mathbf{w}^T\mathbf{x}\right| \\ B(\mathbf{x}) \sim \operatorname{sign}\left(\mathbf{w}^T\mathbf{x}\right) \end{cases}$$

and thus, we can imagine $A$ and $B$ as random functions from $\mathbb{R}^d \to \mathbb{R}$. Conceptually, the idea is to treat the transpose vector $\mathbf{w}^T$ as a random linear and continuous functional from the dual space of $\mathbb{R}^d$ to $\mathbb{R}$ and to decompose it into the product of two random non linear functionals ($A$ and $B$). Let us assume, for the sake of the approximation, that the number of classifiers is very high, and we can approximate the sample average of classifiers with their expected value. Note that, in this case, $\beta$ is fixed and thus we can — *a posteriori* — neglect the fluctuations (which are, in our theory, responsible of the determination of $\beta$). Hence we can rewrite

$$\begin{aligned} y_{ens}^{lin}(\mathbf{x}) &:= \operatorname{sign}\left(\frac{1}{n}\sum_i^n \left[\exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w}_i)\right]\right)\mathbf{x}^T\mathbf{w}_i\right]\right) \\ &\approx \operatorname{sign}\left(\mathbb{E}_{\mathbf{w}}\left[\exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w})\right]\right)\mathbf{x}^T\mathbf{w}\right]\right) \\ &= \operatorname{sign}\left(\mathbb{E}_{A,B}\left[\exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'B(\mathbf{x}'))\right]\right)B(\mathbf{x})A(\mathbf{x})\right]\right). \end{aligned}$$

Note that the term $A(\mathbf{x}')$ in the exponential has been neglected since it is composed with the $\operatorname{sign}$ function. Under the assumption of statistical independence between $A$ and $B$ we can rewrite

$$y_{ens}^{lin} \approx \operatorname{sign}\left(\mathbb{E}_B\left[\exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'B(\mathbf{x}'))\right]\right)B(\mathbf{x})\right]\mathbb{E}_A\left[A(\mathbf{x})\right]\right)$$

and since $\mathbb{E}_A[A(\mathbf{x})] = \mathbb{E}_{\mathbf{w}}|\mathbf{w}^T\mathbf{x}|$ is a positive constant we can rewrite

$$\begin{aligned} y_{ens}^{lin} &\approx \operatorname{sign}\left(\mathbb{E}_B\left[\exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'B(\mathbf{x}'))\right]\right)B(\mathbf{x})\right]\right) \\ &\approx \operatorname{sign}\left(\frac{1}{n}\sum_i^n \left[\exp\left(\tilde{\beta}\mathbb{E}_{\mathbf{x}'}\left[\operatorname{sign}(\mathbf{w}_*^T\mathbf{x}'\mathbf{x}'^T\mathbf{w}_i)\right]\right)\operatorname{sign}\left(\mathbf{x}^T\mathbf{w}_i\right)\right]\right) \end{aligned}$$

for any choice of $\mathbf{x}$ we conclude that

$$y_{ens}(\mathbf{x}) \approx y_{ens}^{lin}(\mathbf{x}),$$

with the approximation improving when $n$ is large (i.e. when the fluctuations of the sample average of classifiers become negligible). The possibility of averaging the contribution of classifiers directly in the parameter space appears as a counterintuitive result due to the strong non linearity of the realization map $\phi$, since

$$\operatorname{sign}\left\{(\alpha \mathbf{w}_1^T + \beta \mathbf{w}_2^T)\mathbf{x}\right\} \neq \alpha \operatorname{sign}\left(\mathbf{w}_1^T \mathbf{x}\right) + \beta \operatorname{sign}\left(\mathbf{w}_2^T \mathbf{x}\right), \tag{5}$$

with the latter term possibly being even outside of the codomain of the sign function. This property offers a significant computational advantage (since we can approximately merge the ensemble of perceptrons into a single perceptron, consuming fewer resources) and is, supposedly, a remarkable consequence of the symmetries in the geometry of the Isotropic Gaussian distribution. A minimal working example in Python is proposed in Appendix E.

## Appendix E Minimal Working Example

We provide a minimal Python code that may be used as a model for the reproduction of the experiments listed across our work. The code is shown in Code Snippet 1.

```python
import jax
import numpy as np
from jax.numpy import sign, exp, ones
from jax.numpy.linalg import norm
from jax.random import PRNGKey, normal

# Hyperparameters
N           = 10000           # number of samples
d           = 500             # dimensions
n           = 20000           # number of classifiers
train_seed  = 42              # simulation seed
test_seed   = 777
w_star      = ones(d)         # replace with any teacher vector
w_star     /= norm(w_star)    # the teacher has unitary norm

# Generation of data (Train)
X = normal(PRNGKey(train_seed),shape = (N,d))
Y = sign(X @ w_star)

# Generation of data (Test)
X_test = normal(PRNGKey(test_seed),shape = (N,d))
Y_test = sign(X_test @ w_star)

# Construction of random classifiers
perceptron_seed = 2025
W = normal(PRNGKey(perceptron_seed),shape = (n,d))

# compute the 0-1 loss for each classifier
loss_01  = jax.vmap(lambda w: (sign(X@w) != Y).mean() )(W)

# compute the score
beta     = np.pi * np.sqrt(d - 2)
score    = exp( - beta * loss_01 )
alfa     = score / score.sum()

# predict train
Y_hat_train = sign(sign(X@W.T) @ alfa)
Y_hat_test  = sign(sign(X_test@W.T) @ alfa)

# output
train_acc   = (Y_hat_train == Y).mean()
test_acc    = (Y_hat_test  == Y_test).mean()
print("Train accuracy : %.3f" % (train_acc) )
print("Test  accuracy : %.3f" % (test_acc) )
```

Code Snippet 1: Minimal Working Example

The construction of $\widehat{\mathbf{y}}$ (denoted in Code Snippet 1 as `Y_hat_train` for train data and `Y_hat_test` for test data) is based on

$$\begin{aligned}
\widehat{\mathbf{y}}(\mathbf{X}, \mathbf{W}) &= \sum_i \mathbf{e}_i \operatorname{sign}\left(\sum_j \operatorname{sign}\left(\mathbf{w}_j^T \mathbf{x}_i\right) \alpha_j\right) \\
&= \sum_i \mathbf{e}_i \operatorname{sign}\left(\sum_j \operatorname{sign}\left(\mathbf{e}_i^T \mathbf{X} \mathbf{w}_j\right) \alpha_j\right) \\
&= \sum_i \mathbf{e}_i \operatorname{sign}\left(\mathbf{e}_i^T \sum_j \operatorname{sign}\left(\mathbf{X} \mathbf{w}_j\right) \alpha_j\right) \\
&= \sum_i \mathbf{e}_i \mathbf{e}_i^T \operatorname{sign}\left(\operatorname{sign}\left(\mathbf{X} \mathbf{W}^T\right) \boldsymbol{\alpha}\right) \\
&= \operatorname{sign}\left(\operatorname{sign}\left(\mathbf{X} \mathbf{W}^T\right) \boldsymbol{\alpha}\right)
\end{aligned}$$

where the sign function is intended in a generalized sense as a map from tensors (and thus also scalars) to tensors of the same dimensions whose entries are the sign of the corresponding entries in the input tensor. The resulting output is shown in Terminal 2.

```
Train accuracy : 0.901
Test  accuracy : 0.862
```

Terminal 2: Output of the Code Snippet 1

It is also possible to test the proposal of Appendix D with few lines of code, as shown in Code Snippet 3, obtaining a result (shown in Terminal 4) which is compatible with the metrics obtained for the unmerged ensemble (shown in Terminal 2)

```python
w_ens = score @ W
Y_hat_train_merge = sign(X @ w_ens)
Y_hat_test_merge  = sign(X_test @ w_ens)

train_acc_merge   = (Y_hat_train_merge == Y).mean()
test_acc_merge    = (Y_hat_test_merge  == Y_test).mean()
print("Train accuracy merged perceptrons : %.3f" % (train_acc_merge) )
print("Test  accuracy merged perceptrons : %.3f" % (test_acc_merge) )
```

Code Snippet 3: Output of the snippet

```
Train accuracy merged perceptrons : 0.900
Test  accuracy merged perceptrons : 0.874
```

Terminal 4: Output of the snippet 3