university of groningen

faculty of science and engineering

Statistical Mechanics of a Nonlinear Learning System

Oscar Tanis

university of groningen / faculty of science and engineering

University of Groningen

Statistical Mechanics of a Nonlinear Learning System

Master's Thesis

To fulfill the requirements for the degree of
Master of Science in Physics
at University of Groningen under the supervision of
Supervisor: Prof. dr. Thomas la Cour Jansen (Zernike Institute for Advanced Materials, University of Groningen),
Second examiner: Prof. dr. Tamalika Banerjee (Zernike Institute for Advanced Materials, University of Groningen)
and
Daily supervisor: dr. Celestine Lawrence (Groningen Cognitive Systems and Materials Center, University of Groningen)

Oscar Tanis (s3124134)

March 30, 2022

Contents

Acknowledgments

I thank my supervisors dr. Celestine Lawrence and prof. dr. Thomas la Cour Jansen for the many hours of discussion which have helped me gain insight and focus pivotal to the success of this project. Further thanks go out to prof. dr. Michael Biehl for his expert opinions and helpful guidance and navigation in the field of statistical mechanics of learning. Finally, I thank prof. dr. Herbert Jaeger for bringing me in touch with the right people to get this project started.

Abstract

Statistical mechanics can give valuable fundamental insight into the qualities of Artificial Intelligence (AI) architectures. Modern developments in AI architectures call for further insight into the qualities of fundamentally nonlinear classifiers. The off-line generalizing behavior of a multiplicative nonlinear extension of the perceptron was explored using mathematical methods derived from statistical mechanics. While a general treatment is hindered by challenges related to nonlinearity, exact results were obtained for a simplified low-dimensional binary version of the system in the limits of high and low temperature. At high temperature, the generalization error was found to conform to a quotient of exponential functions of the number of training examples presented, while in the low temperature limit the generalization error conforms to a sum of exponential decaying functions. These results may facilitate fundamental comparison with neural networks' performance on identical tasks, which may motivate whether nonlinear architectures can be competitive in learning nonlinear behavior.

1 Introduction

1.1 Background

For decades, mathematical tools from Statistical Physics have enabled analytic research into the fundamental learning behavior of a variety of Artificial Intelligence (AI) systems [1–5]. In particular, interesting qualitative results have been obtained for Artificial Neural Networks (ANN), demonstrating the origin of phenomena such as phase transitions and unit specialization [6–11], as well as the effect thereof on the system's performance. Such results aid intelligent decision-making in the design and application of these AI concepts [12–16]. This analysis often treats simple and representative models for the qualitative investigation of AI architectures. Notably, much early research has concerned the simplest possible (single-layer) neural network: the perceptron (e.g. Ref. 17). Later, much work has also been done concerning multi-layered ANN with a small number of layers. See Ref. 8 for a historical overview.

Though such ANN can realize any regression task given that it is large enough [18], alterations from conventional neural networks are proposed: so-called 'dendritic neural networks' [19]. Here, a neuron's activation is explicitly a function of products of its inputs. Meanwhile, recent developments in unconventional computing draw attention to an alternative, more direct approach to realizing (in particular) nonlinear behavior, namely where intelligent behavior is realized physically, rather than virtually. Such so-called neuromorphic computing offers a broad and fascinating perspective regarding both energy- and computational efficiency in computing. A distinction can be made between classes of neuromorphic systems: On the one hand much interest goes out to systems with spiking neuron dynamics as discussed in e.g. Refs. 20–22. On the other hand there exists a class of more unstructured systems, such as in-materio systems [23–26]. Developments in the field of in-materio computing present us with the ability to harness the output of some unknown nonlinear multivariate function to realize desired behavior. This is done by leveraging inherent deterministic nonlinearities in the internal dynamics of electric charge jumping between impurities in matter. As an example, a logical XOR gate was realized by invoking an evolution algorithm in the configuration of voltages applied to a designless, unstructured cluster of nanoparticles [23]. This is just one example of how disorganized systems and resulting nonlinearity can be configured to suit AI purposes [27].

Yet, there is little theoretical understanding of the implications of such nonlinearity for the performance on a regression task; results are often empirical/simulational in nature and therefore offer little fundamental understanding. Abstracting away the precise implementation, these types of systems can be categorized as intrinsically nonlinear. In the present work, these are modeled as a straightforward nonlinear extension of the perceptron in order to qualitatively analyze the capacity of such nonlinear perceptrons to generalize from training data. Specifically, I attempt to apply the familiar tools from (equilibrium) Statistical Physics in order to obtain an analytic expression for the generalization error as a function of the number of examples presented during a training process. When systems with arbitrarily high-dimensional continuous inputs are considered, this computation presents fundamental challenges presented in this work. For the limiting cases of high and low formal

training temperature, and under simplifying assumptions of low-dimensional discrete (binary) inputs, exact results are obtained for the generalization capacity of such a nonlinear system when learning nonlinear behavior. These results - the learning curves - show no surprising features: a gradual, uniform and asymptotic convergence to perfect learning is observed, which is qualitatively very similar to the performance of a perceptron learning linear behavior. However, in the high temperature limit the generalization decays linearly for small training set sizes, revealing qualitatively different behavior than that seen in a linear system.

1.2 Research Questions

To summarize, this thesis focuses on the following problems:

Q1. How can Statistical Mechanics be applied to analyze the learning behavior of nonlinear perceptrons?

Q2. What can be concluded qualitatively about the fundamental learning behavior and generalizing ability of nonlinear perceptrons?

1.3 Thesis Outline

I begin by introducing the reader to the relevant mathematical concepts and relations from the field of the Statistical Mechanics of Learning. I then proceed by introducing the model for the nonlinear perceptron as well as the assumptions made throughout the analysis. I then present in detail what results were obtained in answering the research questions, as well as how these are arrived at. Following this, I provide an extensive discussion on the challenges faced in further answering these questions, as well as possible directions for future research. Finally, I summarize my main conclusions.

2 Theory

The present work applies a theoretical framework commonly known as the statistical mechanics of off-line learning dynamics. This revolves around the application of mathematical concepts from statistical mechanics in equilibrium, a branch of physics, to systems from the field of AI. Specifically, this is applied to systems which 'learn' a mapping between inputs and outputs, and which therefore form the basis for an immense number of AI applications. The statistical mechanics of these systems serves to provide fundamental understanding of the capacities of different architectures of learning systems, and can often be solved exactly for simplified cases [2].
'Off-line learning' in this context refers to the premise that whatever training process is applied has already been concluded for a particular dataset. It therefore focuses on the outcome of a training process, rather than the process of training. The latter is referred to as on-line learning; examples of this are treated in Refs. 28–31. In such analysis, non-equilibrium statistical mechanics may also be applied.
Here, I will explain the core concepts and methods required for understanding the present work, both for readers with a physics background as well as readers with an AI background. For a more complete review of this theory, see Refs. 32–34.

2.1 Phase Space and the Gibbs-Boltzmann distribution

In physics, phase space refers to the space of all possible configurations which define a given (physical) system. In a scenario known as the canonical ensemble each such configuration, or microstate, of the system is then associated with a certain energy, which together with the temperature of the system determines the probability that the system occupies this specific microstate after reaching an equilibrium situation. In general, microstates with lower energy are favored by chance, as governed by the Gibbs-Boltzmann distribution [35] for the given temperature. Here, the system may be found in a particular configuration (or microstate, here labelled by i) with a probability proportional to the Boltzmann factor:

$$p_i \propto \exp(-\beta E_i) \tag{1}$$

Where E_i is the energy associated with this configuration. I use the conventional shorthand $\beta = 1/T$, where T is the temperature (and units were chosen such that Boltzmann's constant equals 1). This distribution typically applies to equilibrium situations wherever the lowest energy state is bounded from below. Note that in some situations different statistics apply, though the reasoning remains the same: configurations are associated with a certain probability determined by some distribution. Analysis here restricts to the Gibbs-Boltzmann distribution.
As there is often a far greater number of possible microstates associated with higher energy, in aggregate one is virtually guaranteed to find that the system tends to occupy a state of an energy greater than the lowest possible energy. Intuitively, this is why everything in this world is not frozen. Only in the extreme case where the temperature tends to zero does chance (according to the Gibbs-Boltzmann distribution) overwhelmingly favor the lowest possible energy, and only then will we be guaranteed to find the system in a state with the lowest possible energy (the

ground state). For convenience, this energy maybe (re)defined such that the ground state has zero energy ($E = 0$).
A completely analogous phenomenon is seen in AI, when a system has been 'trained' on a given set of training data, consisting of training examples (inputs) and training labels (outputs). That is to say, the system has been presented with a set of input data points in some input space, say $\mathbb{R}^N$, and some training algorithm has sought to configure the parameters of the system such that the system's output is equal to the training example output. In other words, the system has been trained to learn a mapping between inputs and outputs. We restrict ourselves to mappings to a scalar $\mathbb{R}^N \to \mathbb{R}$. If we suppose that the system is capable of learning the mapping (also known as a realizable rule) then it is easy to imagine that after training (at zero temperature) only a proper subset of all possible configurations (microstates) is eliminated, while any configuration which gives the correct output for all training examples may still be arrived at with equal probability. We now observe that if the degree of error is viewed as an analog of physical energy, this situation is exactly analogous to the physics picture described above in the zero-temperature limit: only the optimal states remain. For simplicity we assume that the training algorithm has no bias towards any particular microstate, and the system is equally likely to occupy a microstate with equal performance. This is a specific case of what is known as the ergodic hypothesis, which facilitates a lot of statistical analysis. Here lay the motivation for referring to the set of all possible configurations of the AI system as 'phase space'. Following Ref. 32, we may refer to the subspace of phase space which satisfies a given training set $\{\boldsymbol{x}^\gamma\}$ of P examples ($\gamma = 1, \ldots, P$) as the 'version space', whose volume is denoted by $\Omega(\{\boldsymbol{x}^\gamma\})$.
Illustratively, we may consider a simple scenario with a learning system consisting of a vector $\boldsymbol{W} \in \mathbf{R}^N$ which produces from input $\boldsymbol{x} \in \mathbf{R}^N$ the output

$$\sigma(\boldsymbol{x}) = g(y) \equiv g(\boldsymbol{W} \cdot \boldsymbol{x} + b) \tag{2}$$

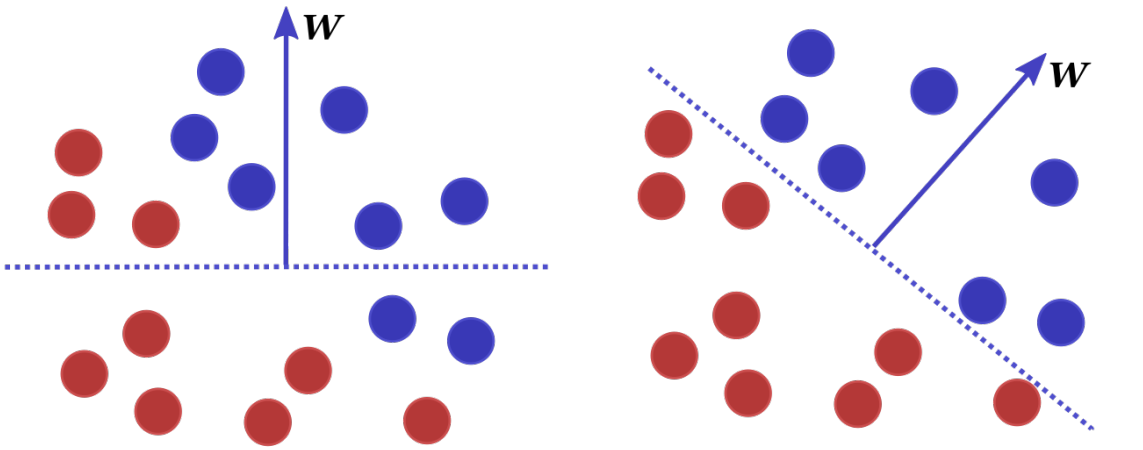
where b is called the bias, g is known as the activation function (typically a sigmoid function or similar) and y is called the activation. This system is known as the perceptron, and it can be seen as the simplest possible neural network. More complex neural networks are often no more than interconnected perceptrons. Therefore perceptrons are the primary focus of a lot of fundamental analysis of neural networks. For binary classification it is common to instead define the output

$$\sigma(\boldsymbol{x}) = \operatorname{sgn}(g(y)) \equiv \operatorname{sgn}(g(\boldsymbol{W} \cdot \boldsymbol{x} + b)) \tag{3}$$

where sgn is the sign function, defined by:

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \tag{4}$$

This system is then known as the binary perceptron.
Central to this analysis is how well a system performs for a given dataset. This is quantified by the training error, which is the sum of the error function $\epsilon(\boldsymbol{W}, \boldsymbol{x})$ as evaluated for each of the examples:



(a) Some training examples are not yet classified correctly.

(b) The training algorithm explores other configurations and favors one with more correct classifications.

Fig. 1: Illustration of how a training algorithm may proceed for a binary perceptron: Each dot is an example. Its features are represented by its position. Incorrectly classified dots contribute to the training error (5).

$$E(\boldsymbol{W}) = \sum_{\gamma=1}^{P} \epsilon(\boldsymbol{W}, \boldsymbol{x}^\gamma) \equiv \sum_{\gamma=1}^{P} \frac{1}{2}(\sigma_t(\boldsymbol{x}^\gamma) - \sigma(\boldsymbol{x}^\gamma))^2 \tag{5}$$

Where σ_t gives the target output, as defined by the training data. Here the popular quadratic error function is chosen. Figures 1 illustrate how a binary perceptron might learn from a dataset: training algorithms tend towards configurations which classify more examples in accordance with the target labels, that is 'correctly' (illustrated in blue).
Now if we assign to each possible perceptron weight vector $\boldsymbol{W}$ (that is, each point in phase space) the training error which would result from it, we obtain a scalar field in phase space known as the error landscape. For the simple case of the perceptron, it is easy to see that a realizable rule must result in a convex error landscape.
It is then easy to imagine how a training algorithm may use the gradient of this error landscape to arrive at the optimal configuration for the system: a method known as gradient descent. A phenomenon analogous to gradient descent is found naturally in physics, where forces (also proportional to the gradient of an energy landscape) operate to reconfigure a system. When the temperature is non-zero, however thermal noise is what allows the physical system to explore the states with greater-than-minimal energy. It may be desirable to have a training algorithm which mimics this behavior. This may, for instance, be realized by an algorithm which updates the system step-by-step according to:

$$\frac{\partial \boldsymbol{W}}{\partial t} = -\nabla E(\boldsymbol{W}) + \boldsymbol{\eta}_T(t) \tag{6}$$

Where t is time, $E_{\boldsymbol{W}} =$ and $\boldsymbol{\eta}_T(t)$ is some uncorrelated zero-mean noise term at a

certain formal "temperature" T defined by

$$\langle \eta_{Ti}(t)\eta_{Tj}(t')\rangle = 2T\delta_{ij}\delta(t-t') \tag{7}$$

This type of dynamics is identical in form to overdamped Langevin dynamics (or Brownian dynamics), which in equilibrium is known to result in a Gibbs-Boltzmann distribution. The analysis which follows relates to any training algorithm which results in such a distribution; not exclusively to Langevin dynamics.
Many training algorithms applied in conventional learning systems (most notably neural networks), are not designed to eliminate all of phase space besides version space. Considerations such as overfitting, nonconvex error landscapes and noisy data have motivated learning algorithms which tend to favor configurations (microstates) which correctly label training examples, without totally excluding configurations which do not. Situations where this is desirable then motivate the analysis of the behavior of a learning system 'at some given temperature'.

2.2 Partition function

Normalizing the probabilities given by Eq. (1), we obtain a quantity also used in the canonical ensemble in physics, called the partition function:

$$Z = \int \exp(-\beta E(\boldsymbol{W}))d\mu(\boldsymbol{W}) \tag{8}$$

where again $\beta = 1/T$, integration runs over all of phase space, and $d\mu(W)$ encodes the capacity of the system to realize microstate $\boldsymbol{W}$ (often called the *a priori* measure of $\boldsymbol{W}$). In words, the partition function is obtained by integrating the so-called Boltzmann factor over all possible microstates. The partition function can be used to obtain useful information about the expectation value of 'macroscopic' variables, such as the energy. We may use it to take 'thermal averages' of quantities, denoted by $\langle \dots \rangle_T$:

$$\langle X\rangle_T = \frac{1}{Z}\int X \exp\left(-\beta E(\boldsymbol{W})\right) d\mu(\boldsymbol{W}) \tag{9}$$

2.3 Student Teacher Scenario

The training data target outputs $\sigma_t(\boldsymbol{x})$ in Eq. (5) are known as the teacher signal. The system which is trained to learn from this signal is often called the student. The behavior of the teacher signal defines the type of 'problem' for which the student's performance is analyzed. This can in theory be used to compare the capacity of one system architecture to imitate the behavior of another architecture.
Analysis is most straightforward and clarifying, however, when there exists a configuration which has zero error ($E = 0$). This is clearly the case in scenarios where the student and teacher have the same architecture, and only the configuration (microscate) of the system parameters may vary. This is the scenario we consider in the present work.
An intuitive picture of how this scenario functions is illustrated in Fig. 2. Conceptually, the student teacher scenario for a binary perceptron can be compared to a physical

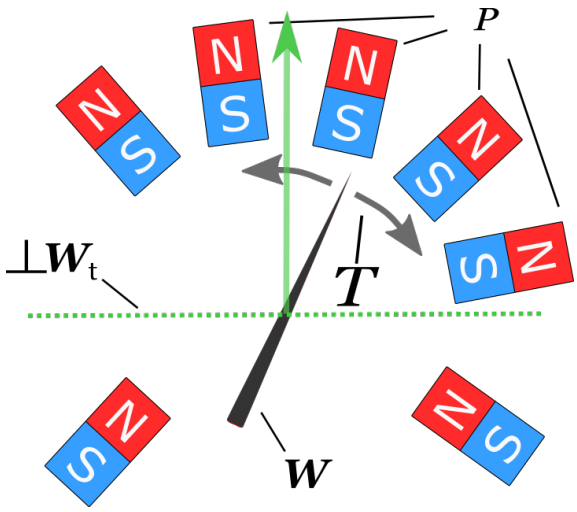


Fig. 2: An intuitive picture of the student teacher scenario for the binary perceptron, and an analogy to physics illustrating how the number of examples P, random fluctuations with temperature T and the teacher perceptron $\boldsymbol{W}_{\mathrm{t}}$ all affect the configuration of student perceptron $\boldsymbol{W}$. For simplicity the bias, see (2), is chosen to equal 0.

scenario where P magnets (analogous to the P examples) with random positions (analogous to random features) direct a compass needle (analogous to the student weight vector). Meanwhile, random fluctuations shake the compass needle with an intensity that is proportional to temperature T. All magnets are equally compelling to the needle, and some unknown polarization direction (analogous to the teacher vector) determines whether magnets face inward with their north pole or south pole (analogous to a positive or negative target label respectively).

2.4 Quenched Disorder and Generalization Error

In the AI context, the partition function may be used to obtain an estimate of the performance of the system on the given training set. Note that the partition function is specific to the particular training data provided. Consider Fig. 2: clearly, the behavior of the compass needle is affected by the particular randomly sampled positions of the magnets. For fundamental research, one should average any quantity of interest over all possible training datasets. This is known as averaging over the quenched disorder, and denoted by $\langle \dots \rangle_{\mathbb{D}}$:

$$\langle X \rangle_{\mathbb{D}} \equiv \int X \prod_{j}^{N} d\mu(x^j) \tag{10}$$

Where the $d\mu(x_j)$ term specifies the a priori measure of the example input's jth component, and reflects the nature of the dataset. For general analysis, each component is often assumed to be Gaussian distributed, but we will also consider a binary distribution of the input features.
An interesting quantity which can then be obtained is the training error per example, given by:

$$\epsilon_t(T,P) = P^{-1}\left\langle\langle E(\boldsymbol{W})\rangle_T\right\rangle_{\mathbb{D}} \tag{11}$$

An even more interesting quantity is the generalization error: this quantifies the expected error of the system for a randomly sampled example. This sets it apart from the training error, which averages only over those examples which happen to be in the training set. Precisely, it is given by:

$$\epsilon_g(T,P) = \left\langle\langle\epsilon(\boldsymbol{W})\rangle_T\right\rangle_{\mathbb{D}} \tag{12}$$

where

$$\epsilon(\boldsymbol{W}) = \int \epsilon(\boldsymbol{W},\boldsymbol{x})d\mu(\boldsymbol{x}) \tag{13}$$

is known as the generalization function, and this quantifies the performance of a *specific* system configuration on randomly sampled input.
In other words, the generalization error is the expected error of the system when presented with an arbitrary input, after being trained on a particular training dataset of size P at temperature T, averaged over all possible such training datasets. The generalization error is of great interest, as it quantifies the fundamental capacity of a system to generalize a given rule from examples, in other words, to *learn* a rule.
In general, one may obtain the expected training error through the so-called free energy F:

$$F(T,P) = -T\langle\ln Z\rangle_{\mathbb{D}} \tag{14}$$

using that: (See e.g. Ref. 34)

$$\epsilon_t = \frac{1}{P}\frac{\partial(\beta F)}{\partial\beta} \tag{15}$$

2.5 Interesting Limits

Clearly, thermal averaging is crucial in obtaining useful results, in particular the quantity $\langle\ln Z\rangle_T$ is crucial. In the general case, computation of the thermal average may be carried out using the so-called replica trick, which may be rather involved (See e.g. Ref. 2), and is in some cases non-solvable. Two simplifying limits are of interest:
As mentioned, when the training temperature is $T \to 0$, training ensures that only those configurations where the error is lowest are allowed to occur. This scenario gives qualitative insight into the capacity of the system to simply store information provided. Analysis in this limit may use the quenched average of version space:

$$\Omega_P(\boldsymbol{W}_{\mathrm{t}}) = \langle\Omega(\boldsymbol{W}_{\mathrm{t}},\{\boldsymbol{x}^\gamma\})\rangle_{\mathbb{D}} \tag{16}$$

where $\boldsymbol{W}_{\mathrm{t}}$ denotes the configuration of the teacher system.
Alternatively, one may consider the limit $T \to \infty$, $\beta \to 0$, which has proven useful in the qualitative investigation of various learning scenarios [2, 34]. A useful approximation exists:

$$\langle\ln Z\rangle \approx \ln\langle Z\rangle \tag{17}$$

This is known as the Annealed Approximation [36], and it allows us to focus on evaluating the average partition function rather than evaluating the average log partition function, which is generally far more difficult to compute. In the limit $T \to \infty$, this approximation becomes exact [6]. In analysis of this scenario, an infinite amount of noise must be compensated with an infinite amount of training data. We therefore define a quantity:

$$\alpha = \beta P / N \tag{18}$$

which can be seen as a properly scaled training set size [2]. As the training set size then tends towards infinity, the generalization error and training error become equivalent.
These limits both offer qualitative insight into the fundamental behavior of a system in the case of either noiseless or very noisy training dynamics.

2.6 Activation Space

In the annealed approximation, then, the crucial quantities to compute are of a form similar to:

$$\langle Z \rangle_{\mathbb{D}} = \int \prod_{\gamma=1}^{P} d\mu(\boldsymbol{x}^\gamma) \int d\mu(\boldsymbol{W}) \exp(-\beta \epsilon(\boldsymbol{W}, \boldsymbol{x}^\gamma))$$

In Refs. 32 and 34 (see Appendix C) a conventional approach to evaluating this quantity for the case of a perceptron is illustrated. First, it is assumed that all examples are independently sampled and identically distributed (IID), which gives:

$$\langle Z \rangle_{\mathbb{D}} = \int d\mu(\boldsymbol{x}) \int d\mu(\boldsymbol{W}) \exp(-P\beta \epsilon(\boldsymbol{W}, \boldsymbol{x})) \tag{19}$$

In a student teacher Scenario, we can write

$$\epsilon(\boldsymbol{W}, \boldsymbol{x}) = \frac{1}{2}(g(\boldsymbol{W} \cdot \boldsymbol{x}) - g(\tilde{\boldsymbol{W}} \cdot \boldsymbol{x}))^2$$

Where $\tilde{\boldsymbol{W}}$ is the configuration of the teacher perceptron. A very popular approach is then to transform the N-dimensional integral over $\boldsymbol{x}$ into a two-dimensional integral over newly introduced auxiliary variables, say $u = \boldsymbol{W} \cdot \boldsymbol{x}$ and $v = \tilde{\boldsymbol{W}} \cdot \boldsymbol{x}$, which equate to respectively the student and teacher activations (2). This is realized by simply introducing dirac delta functions:

$$\begin{aligned} \langle Z \rangle_{\mathbb{D}} = \int d\mu(\boldsymbol{W}) \int \prod_{\gamma=1}^{P} du_\gamma dv_\gamma \exp\left(-\beta \frac{1}{2}(g(u_\gamma) - g(v_\gamma))^2\right) \\ \int d\mu(\boldsymbol{x}^\gamma) \delta^N(u_\gamma - \boldsymbol{W} \cdot \boldsymbol{x}^\gamma) \delta^N(v_\gamma - \tilde{\boldsymbol{W}} \cdot \boldsymbol{x}^\gamma) \end{aligned} \tag{20}$$

The delta functions are written in integral form using

$$\delta^N(\boldsymbol{z}) = \int \frac{d\boldsymbol{z} d\hat{\boldsymbol{z}}}{2\pi} \exp(i\hat{\boldsymbol{z}} \cdot \boldsymbol{z}) \tag{21}$$

Since we assume that examples $\boldsymbol{x}$ are IID and normally distributed, the integration over input space $\int d\mu(\boldsymbol{x})$ can readily be carried out resulting in a Gaussian integral in activation space.
As a corollary of Eq. (20), we may write:

$$\Omega_P(\tilde{\boldsymbol{W}}) = \int d\mu(\boldsymbol{W}) \int \prod_{\gamma=1}^{P} du_\gamma dv_\gamma \theta(u_\gamma v_\gamma) \int d\mu(\boldsymbol{x}^\gamma) \delta^N(u_\gamma - \boldsymbol{W}\cdot\boldsymbol{x}^\gamma) \delta^N(v_\gamma - \tilde{\boldsymbol{W}}\cdot\boldsymbol{x}^\gamma) \tag{22}$$

where the Heaviside function θ is known as the indicator function, indicating which microstates are included in version space.

3 Model

The present work analyses in general a class of systems which may be viewed as a nonlinear extension of a perceptron. The perceptron is linear in the sense that its activation is a linear combination of its inputs. This linear behavior forms the basis of the functioning of virtually all Artificial Neural Networks (ANN). Recent advances mentioned in the introduction pave the way for implementation of learning systems which include multiplicative (nonlinear) interactions between the input features. Most generally, the unit activation (2) could be a function of all arbitrarily high-order interactions (products) of the features. However, for first results on the behavior of such nonlinear systems, it should suffice to model them as a simple quadratic extension of a perceptron, now with activation also determined by first-order multiplicative interactions between the features, and output label given simply as:

$$\sigma(\boldsymbol{x}) = g(y) \equiv g(\boldsymbol{W}\cdot\boldsymbol{x} + \sum_{j,k=1}^{N} \omega_{jk} x_j x_k) \tag{23}$$

where the interaction matrix ω is an $N \times N$ upper triangular matrix with zeros on the diagonal, such that it models all multiplicative interactions between the inputs. We shall refer to this system as a Multiplicative Perceptron (MP). For simplicity, as we are only interested in the influence of the nonlinearity, we will restrict ourselves to the simplified case of $g(y) = y$ and do not consider a bias here. Similar to the case of the perceptron, for binary classification we restrict to;

$$\sigma(\boldsymbol{x}) = \mathrm{sgn}(g(y)) = \mathrm{sgn}(\boldsymbol{W}\cdot\boldsymbol{x} + \sum_{j,k=1}^{N} \omega_{jk} x_j x_k) \tag{24}$$

This shall be referred to as a binary MP or an MP classifier. The introduction of multiplicative feature interactions fundamentally changes the intuition behind classification: see Fig. 3 for an example of a classification boundary for $N = 2$.
It should be noted that this MP can be seen as a specific class of Support Vector Machines (SVM), for which statistical mechanical analysis has been performed [37]. However, while SVM are often trained using a maximal stability algorithm, we consider here a different class of training algorithms resulting in different dynamics analysis.
For the binary MP, as well as for the familiar binary perceptron, only the ratios between the configuration parameters are relevant to the system's output. That is, only the direction of a vector in phase space is relevant. In the case of the perceptron, this is nicely illustrated in e.g. Ref. 32. Here, this motivates the convention restricting analysis to spherical configurations, normalized to a magnitude of $\sqrt{N}$. That is, analysis commonly restricts perceptrons to have weight vectors of magnitude $|\boldsymbol{W}| = \sqrt{N}$, so as to avoid degeneracy and still obtain qualitatively insightful results. We will here extend this convention by imposing the restriction:

$$|\boldsymbol{W}|^2 + \sum_{j,k=1}^{N} \omega_{jk}^2 = N \tag{25}$$

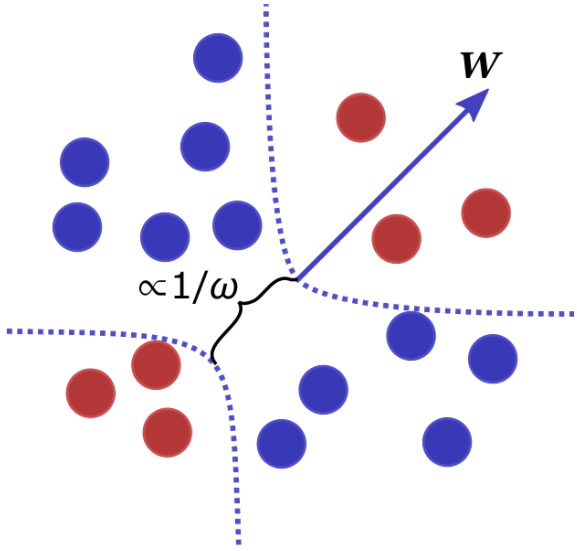


Fig. 3: A general picture of what the Binary Multiplicative Perceptron's classification boundary may look like for the case of $N = 2$. The hyperbolic shape results in fundamentally different behavior than that of the Perceptron.

Examples with continuous features are conventionally modeled most generally as vectors of IID components, sampled from a zero-mean, unit variance distribution. That is:

$$\langle x_j^\mu \rangle = 0, \langle x_j^\mu x_k^\nu \rangle = \delta_{j,k}\delta_{\mu,\nu}$$

We will assume these to be normally distributed, meaning:

$$d\mu(\boldsymbol{x}) = \prod_{j=1}^{N} \frac{dx_j}{\sqrt{2\pi}} e^{-x_j^2/2} \tag{26}$$

Another scenario to consider is that of strictly binary features. For ease of analysis, we encode binary features as:

$$\boldsymbol{x} \in \{-1, 1\}^N \tag{27}$$

Clearly, in this case there is only a finite number of possible binary inputs, and we naturally suppose that the features follow a Bernoulli distribution [38] with $p = 1/2$. We then see that integration over all of feature space becomes a summation:

$$\int \dots d\mu(\boldsymbol{x}) = \frac{1}{2^N} \sum_{\boldsymbol{x}} \dots \tag{28}$$

where summation runs over all possible binary inputs. Illustratively, one may then encode the behavior of a logical XOR gate as ($N = 2$):

$$\sigma_t(\boldsymbol{x}) = -x_1 x_2 \tag{29}$$

Which will be used as the teacher system in later analysis, as it is a benchmark example of nonlinear behavior which cannot be realized with a perceptron [25, 39]. Note that while the student teacher scenario for the MP, due to its inherent nonlinearity, does not lend itself well to visualization or analogy with physics as in Fig. 2, the

student teacher scenario remains an interesting case study, and functions identically for the MP as it does for the perceptron.

4 Results

Results were obtained for the binary MP in a particular student teacher scenario (see section 2.3). For the case where the teacher is a logical XOR gate and features are binary encoded as $\{-1, 1\}$, exact solutions for the generalization error exist for both the high-temperature and the zero-temperature limit. In the case of $N = 2$, the computation is tractable and illustrative.

4.1 High Temperature Limit

Firstly, we consider the $T \to \infty$ limit, and apply the annealed approximation (17). We therefore begin with writing the quenched average of the partition function as a summation (28):

$$\langle Z \rangle_{\mathbb{D}} = \frac{1}{4} \sum_{\boldsymbol{x}^1} \dots \frac{1}{4} \sum_{\boldsymbol{x}^P} d\mu(\boldsymbol{W}, \omega) \exp\left(-\beta \sum_{\gamma=1}^{P} \frac{1}{2} (\sigma_s(\boldsymbol{x}^\gamma) - \sigma_s(\boldsymbol{x}^\gamma))^2 \right) \tag{30}$$

We use that for binary outputs the error can be written as:

$$\frac{1}{2}(\sigma_s(\boldsymbol{x}) - \sigma_t(\boldsymbol{x}))^2 = 2\theta(-\sigma_s(\boldsymbol{x})\sigma_t(\boldsymbol{x})) \tag{31}$$

where θ is the Heaviside step function. Furthermore, we use the fact that in the high temperature limit, we also have $P \to \infty$, and therefore we can, using Eq. (18), replace the sum in the exponential (30) rewriting to:

$$\langle Z \rangle_{\mathbb{D}} = \left(\frac{1}{4}\right)^P 4^P d\mu(\boldsymbol{W}, \omega) \exp\left(-\alpha \sum_{\boldsymbol{x}} \theta(-\sigma_s(\boldsymbol{x})\sigma_t(\boldsymbol{x})) \right) \tag{32}$$

In evaluating this integral, we use the description for the XOR gate teacher as in Eq. (29). We use the fact that for the symmetric binary encoding (27) $x_1^2 = x_2^2 = 1$, and expand the summation as:

$$\begin{aligned} \exp\left(-\alpha \sum_{\boldsymbol{x}} \theta(-\sigma_s(\boldsymbol{x})\sigma_t(\boldsymbol{x})) \right) = \exp\left(-\alpha \sum_{\boldsymbol{x}} \theta(W_1 x_2 + W_2 x_1 + \omega) \right) \\ = \exp\big(-\alpha \big[\theta(W_1 + W_2 + \omega) \\ + \theta(-W_1 + W_2 + \omega) \\ + \theta(W_1 - W_2 + \omega) \\ + \theta(-W_1 - W_2 + \omega) \big] \big) \\ \equiv \exp\left(-\alpha(A + B + C + D)\right) \end{aligned} \tag{33}$$

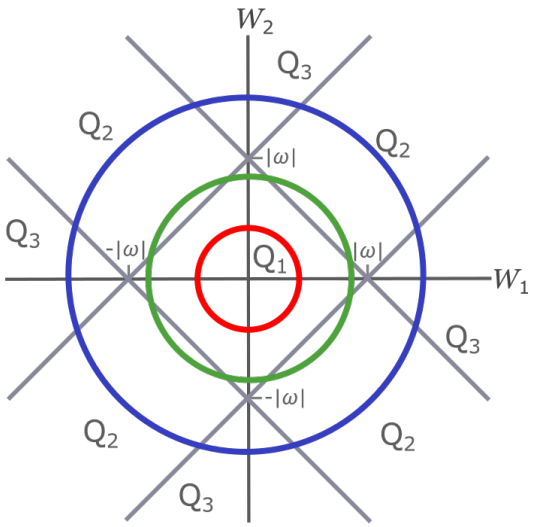
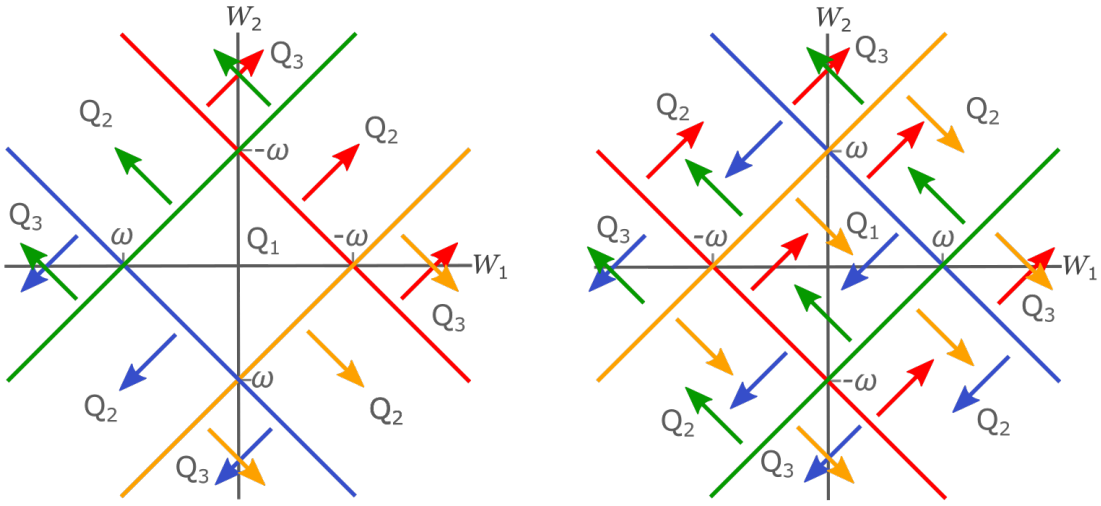


Fig. 5: The partition function is obtained by integration of the probability-per-microstate (Boltzmann factor) along circular slices of a sphere. Three trigonometrically distinct cases exist for these circles: $r < \frac{|\omega|}{\sqrt{2}}$ (red), $\frac{|\omega|}{\sqrt{2}} < r < |\omega|$ (green), $r > |\omega|$ (blue). See Eqs. (34), (35).



(a) $\omega < 0$ (b) $\omega > 0$

Fig. 4: Areas of phase space with constant Boltzmann factors, divided by lines depending on the value of ω. Colored arrows mark regions where the respective Heaviside function (31) evaluates to 1, indicating that the respective example was classified incorrectly. Hence, the number of arrows in a region indicate the number of examples classified incorrectly by MP's with configurations belonging to that region.

Each of the step functions now labeled A, B, C and D then indicate the error corresponding to a specific microstate when presented with a specific example. For a given value of ω, then, we find that we can view the value of each of the step functions as separated by a line in (W_1, W_2)-coordinates. These lines effectively

segment phase space into regions within which identical classifications are realized. This allows for the evaluation of the integral using trigonometric arguments, as we can use these linear boundaries to split up the integral over all spherical (25) MP configurations into integration over separate regions of constant value. Figures 4a and 4b illustrate schematically how for each point in phase space some number of the step functions A, B, C and D equal 1, depending on separating lines, and how the situation is geometrically different for positive ω and negative ω. Due to the spherical configuration, each value of ω corresponds to a circle in (W_1, W_2)-coordinates, along which integration can be carried out separately. Three qualitatively different cases for the radii r of these circles can be distinguished, namely $r < \frac{|\omega|}{\sqrt{2}}$, $\frac{\omega|}{\sqrt{2}} < r < |\omega|$, and $r > |\omega|$. This is illustrated in Fig. 5. A geometric argument, combined with the fact that $r = \sqrt{2-\omega^2}$ (27) reveals we can write

$$\langle Z \rangle = \int_{-\sqrt{2}}^{0} g_-(\omega) d\omega + \int_{0}^{\sqrt{2}} g_+(\omega) d\omega \tag{34}$$

where $g_\pm$ are the contribution to the partition function from microstates with positive and negative values of ω respectively. For each value of ω, this is obtained by integration along a circle (see Fig. 5) and is given as:

$$g_\pm(\omega) = \begin{cases} 2\sqrt{2}\pi Q_{1\pm}, & \text{for } |\omega| > \sqrt{\frac{4}{3}} \\ \sqrt{2}\arccos\left(\frac{|\omega|}{\sqrt{4-2\omega^2}}\right)(8Q_{2\pm} - 8Q_{1\pm}) + 2\sqrt{2}\pi Q_{1\pm}, & \text{for } \sqrt{\frac{4}{3}} > |\omega| > 1 \\ \sqrt{2}\arccos\left(\frac{|\omega|}{\sqrt{4-2\omega^2}}\right)(8Q_{3\pm} - 8Q_{2\pm}) + \sqrt{2}\pi\left(4Q_{2\pm} - 2Q_{3\pm}\right), & \text{for } 1 > |\omega| \end{cases} \tag{35}$$

Where we refer to the constants $Q_{1\pm}, Q_{2\pm}, Q_{3\pm}$ as 'center integrand', 'edge integrand' and 'corner integrand' respectively; these give the value of the integrand in the regions indicated in Figs. 4. These values are determined by the number of training examples which are classified incorrectly by microstates in the respective region, for positive and negative values of ω respectively, see Fig. 5:

$$\begin{aligned} Q_{1-} &= \exp\left(-\alpha \times 0\right) = 1 \\ Q_{2-} &= \exp\left(-\alpha \times 1\right) \\ Q_{3-} &= \exp\left(-\alpha \times 2\right) \\ Q_{1+} &= \exp\left(-\alpha \times 4\right) \\ Q_{2+} &= \exp\left(-\alpha \times 3\right) \\ Q_{3+} &= \exp\left(-\alpha \times 2\right) \end{aligned} \tag{36}$$

Integration is then performed in Wolfram Mathematica [40]. Using Eqs. (14) and (15), and $\frac{1}{P}\frac{\partial}{\partial\beta} = \frac{1}{2}\frac{\partial}{\partial\alpha}$, as well as the fact that in this limit $\epsilon_g \to \epsilon_t$, we obtain a plot of the generalization error, see Fig. 6. The plot characterizes the learning behavior of a MP classifier when learning a nonlinear rule from binary examples in a very noisy training process: a uniform and asymptotic decrease of discrepancy between the behavior of the XOR gate and that of the student MP classifier is observed. Specifically, the curve is characterized as the quotient of two sums of exponential functions, given exactly in appendix B.

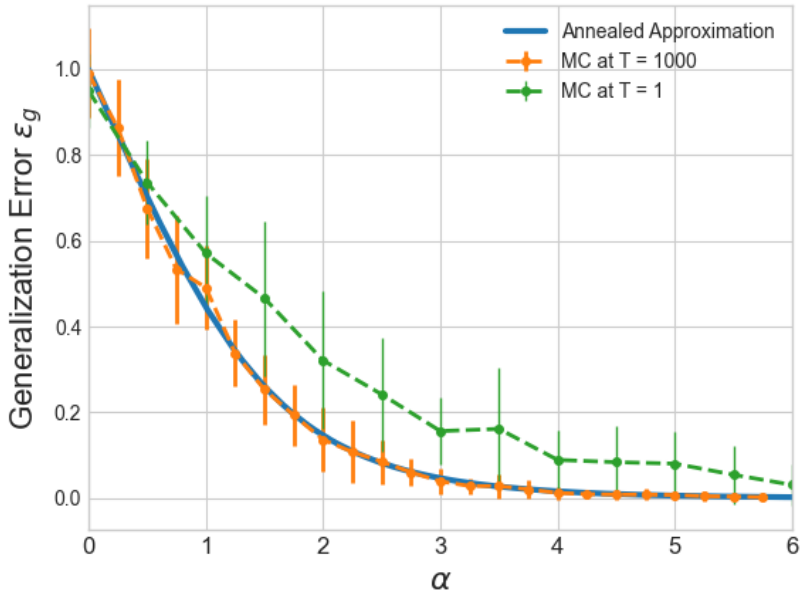


Fig. 6: Generalization error ϵ_g, which in this limiting case with $T \to \infty$ equals ϵ_t, see Eqs. (12), (15), of the $N = 2$ binary examples MP classifier versus α (18), according to both the analytic result obtained through the partition function (Fig. 5) and the Monte Carlo Simulation (Error bars show standard deviations).

4.2 Zero Temperature Limit

For binary examples with $T = 0$ or $\beta \to \infty$, the exact generalization error may be arrived at through a different avenue than in the high T limit: For binary examples we may write Eq. (12) as:

$$\begin{aligned} \epsilon_g &= \langle \frac{1}{4} \sum_{\boldsymbol{x}} \langle \epsilon(\boldsymbol{W}, \omega, \boldsymbol{x}) \rangle_T \rangle_{\mathbb{D}} \\ &= \frac{1}{2} \langle \langle \sum_{\boldsymbol{x}} \theta(W_1 x_2 + W_2 x_1 + \omega) \rangle_T \rangle_{\mathbb{D}} \end{aligned} \tag{37}$$

where we used Eqs. (31) and (29). This thermal average is obtained by finding the probability of the system occupying microstates in each of the individual regions indicated in Figs. 4a, 4b, which depends on the Boltzmann factor of the microstate. Let, for example, Q_{2+} denote the Boltzmann factor corresponding to any microstate belonging to the region Q_2 in the figure, for a positive value of ω (See Fig. 5). We then multiply the probability of finding the system in any such microstate by the respective generalization function (13), which is known for each of the regions.
Analysis is more involved than in the $T \to \infty$ case because due to the small training-set cardinality, we must consider the different possible scenarios for the resulting Boltzmann factors, weighted by their respective probability. This proceeds as follows: For an arbitrary example set $\{\boldsymbol{x}^\gamma\}$, computing the thermal average amounts to integrating over version space as obtained for said example set. We therefore define and compute a 'version-space-specific' generalization error:

$$\begin{aligned}\xi(\{\boldsymbol{x}^\gamma\}) &\equiv 2\langle\sum_{\boldsymbol{x}}\theta(W_1x_2+W_2x_1+\omega)\rangle_T \\ &= \frac{2}{\Omega(\{\boldsymbol{x}^\gamma\})}\left(\int_{-\sqrt{2}}^{0}h_-(\omega)d\omega+\int_0^{\sqrt{2}}h_+(\omega)d\omega\right)\end{aligned} \tag{38}$$

Here $h_\pm(\omega)$ equals the generalization function integrated over all microstates in version space with a specific value of ω. The observation that

$$\epsilon(\boldsymbol{W},\omega)=\begin{cases}2\times 0 & \text{if } (\boldsymbol{W},\omega)\in Q_{1-}\\ 2\times 1/4 & \text{if } (\boldsymbol{W},\omega)\in Q_{2-}\\ 2\times 1/2 & \text{if } (\boldsymbol{W},\omega)\in Q_{3-}\\ 2\times 1 & \text{if } (\boldsymbol{W},\omega)\in Q_{1+}\\ 2\times 3/4 & \text{if } (\boldsymbol{W},\omega)\in Q_{2+}\\ 2\times 1/2 & \text{if } (\boldsymbol{W},\omega)\in Q_{3+}\end{cases} \tag{39}$$

combined with the same geometric reasoning as in the $T\to\infty$ case implies that:

$$h_-(\omega)=\begin{cases}0, & \text{for } |\omega|>\sqrt{\frac{4}{3}}\\ \sqrt{2}\arccos\left(\frac{|\omega|}{\sqrt{4-2\omega^2}}\right)(2Q_{2-}), & \text{for } \sqrt{\frac{4}{3}}>|\omega|>1\\ \sqrt{2}\arccos\left(\frac{|\omega|}{\sqrt{4-2\omega^2}}\right)(4Q_{3-}-2Q_{2-})+\sqrt{2}\pi\,(Q_{2-}-Q_{3-}), & \text{for } 1>|\omega|\end{cases} \tag{40}$$

and

$$h_+(\omega)=\begin{cases}2\sqrt{2}\pi Q_+, & \text{for } |\omega|>\sqrt{\frac{4}{3}}\\ \sqrt{2}\arccos\left(\frac{|\omega|}{\sqrt{4-2\omega^2}}\right)(6Q_{2+}-8Q_{1+})+2\sqrt{2}\pi Q_{1+}, & \text{for } \sqrt{\frac{4}{3}}>|\omega|>1\\ \sqrt{2}\arccos\left(\frac{|\omega|}{\sqrt{4-2\omega^2}}\right)(4Q_{3+}-6Q_{2+})+\sqrt{2}\pi\,(3Q_{2+}-Q_{3+}), & \text{for } 1>|\omega|\end{cases} \tag{41}$$

In order to find $\Omega(\{\boldsymbol{x}^\gamma\})\equiv Z$, then, we perform the integral in Eq. (34), where now the values (Boltzmann factors) of $Q_{1\pm}$, $Q_{2\pm}$ and $Q_{3\pm}$ take values from 0 to 1, indicating how much the of respective region is included in the version space.
Finally, in obtaining the quenched average of this thermal average we first observe that for each point in version space we have:

$$\prod_{\gamma=1}^{P}\theta\left(-W_1x_2^\gamma-W_2x_1^\gamma-\omega\right)=1 \tag{42}$$

which reflects the fact that any unique example may serve to eliminate a region of phase space from version space. Now let the set of all possible example sets of cardinality P be denoted by $\mathcal{X}\equiv\{\{\boldsymbol{x}^\gamma\}_{\gamma=1}^P\}$. The quenched average can then be written as:

$$\epsilon_g = \left(\frac{1}{4}\right)^P \sum_{\{\boldsymbol{x}^\gamma\}\in\mathcal{X}} \left(\prod_{\gamma=1}^{P} \left[\theta(-W_1 x_2^\gamma - W_2 x_1^\gamma - \omega)\right] \xi(\{\boldsymbol{x}^\gamma\}) \right) \tag{43}$$

We now note that

$$\sum_{\{\boldsymbol{x}^\gamma\}\in\mathcal{X}} \left(\prod_{\gamma=1}^{P} \left[\theta(-W_1 x_2^\gamma - W_2 x_1^\gamma - \omega)\right] \right) = \left[\sum_{\boldsymbol{x}} \theta(-W_1 x_2^\gamma - W_2 x_1^\gamma - \omega) \right]^P \tag{44}$$

and we use that whenever $A^2 = A$, $B^2 = B$, etc. (As is the case when A and B are Heaviside step functions):

$$\begin{aligned} [A+B+C+D]^P = &(A+B+C+D)\times(I) \\ &+ (AB+BD+BC+AC+AD+CD)\times(II) \\ &+ (ABC+BCD+CDA+DAB)\times(III) \\ &+ (ABCD)\times(IV) \end{aligned} \tag{45}$$

where

$$\begin{aligned} (I) \equiv& 1 \\ (II) \equiv& \sum_{j=1}^{P-1} \binom{P}{j} \\ (III) \equiv& \sum_{j=1}^{P-1} \binom{P}{j} \sum_{k=1}^{j-1} \binom{j}{k} \\ (IV) \equiv& \sum_{j=1}^{P-1} \binom{P}{j} \sum_{k=1}^{j-1} \binom{j}{k} \sum_{l=1}^{P-j-1} \binom{P-j}{l} \end{aligned} \tag{46}$$

(See appendix A). If we then identify A, B, C and D with the step functions (42) corresponding to examples $\boldsymbol{x} = \binom{1}{1}, \binom{-1}{1}, \binom{1}{-1}, \binom{-1}{-1}$ respectively, we can then use the coefficients (46) to quantify the number of possible example sets resulting in each possible version space. This leads to a natural and exhaustive classification of all example sets into subsets, and therefore to the probability of each of the 5 scenarios for the Boltzmann factors. Specifically:

- Q_{1-} is included in version space for all example sets, regardless of the example set. Furthermore:

1. For all example sets where A, B, C or D equals 1 (that is, in (I) number of example sets), $^3/_4$ Q_{2-} regions, $^2/_4$ Q_{3-} regions, $^2/_4$ Q_{3+} regions and $^1/_4$ Q_{2+} region are included in version space.

   $\rightarrow Q_{1-} = 1$, $Q_{2-} = {}^3/_4$, $Q_{3-} = Q_{3+} = {}^1/_2$, $Q_{2+} = {}^1/_4$, $Q_{1+} = 0$;

2. For all example sets where AB, AC, CD or BD equals 1 (that is, in (II) number of example sets), $^2/_4$ Q_{2-} regions, $^1/_4$ Q_{3-} region and $^1/_4$ Q_{3+} region are included.

   $\rightarrow Q_{1-} = 1$, $Q_{2-} = {}^2/_4$, $Q_{3-} = Q_{3+} = {}^1/_4$, $Q_{2+} = Q_{1+} = 0$;

3. For all example sets where AD or BC equals 1 (that is, in (II) number of example sets), $^2/_4$ Q_{2-} regions are included

   $\rightarrow Q_{1-} = 1$, $Q_{2-} = {}^2/_4$, $Q_{3-} = Q_{3+} = Q_{2+} = Q_{1+} = 0$;

4. For all example sets where ABC, BCD, CDA or DAB equals 1 (that is, in (III) number of example sets), $^1/_4$ Q_{2-} region is included.

   $\rightarrow Q_{1-} = 1$, $Q_{2-} = {}^1/_4$, $Q_{3-} = Q_{3+} = Q_{2+} = Q_{1+} = 0$;

5. For all example sets where $ABCD$ equals 1 (that is, in (IV) number of example sets), no further regions are included. In these cases, version space is restricted to Q_{1-}.

   $\rightarrow Q_{1-} = 1$, $Q_{2-} = Q_{3-} = Q_{3+} = Q_{2+} = Q_{1+} = 0$;

The expansion in Eq. (45) assumes $P > 0, P \in \mathbb{Z}$. Only when $P = 0$ may Q_{1+} be included in version space.
From this reasoning, we may now deduce the exact generalization error as:

$$\epsilon_g = \left(\frac{1}{4}\right)^P (4(\mathrm{I})\xi_1 + 4(\mathrm{II})\xi_2 + 2(\mathrm{II})\xi_3 + 4(\mathrm{III})\xi_4 + (\mathrm{IV})\xi_5) \tag{47}$$

where each ξ is defined as in Eq. (38) and to be computed with the above indicated values of $Q_{1\pm}, Q_{2\pm}, Q_{3\pm}$ in Eqs. (40) and (41). The integration is then performed in Wolfram Mathematica, resulting in the exact expression given in appendix C. The result is evaluated for varying P and plotted in Fig. 7.

4.3 Monte Carlo Simulations

Demonstrating the validity of the analytical results obtained above, Monte Carlo simulations of the training process were performed using a standard Metropolis-like scheme [6, 41]. Simulations were performed for the low temperature extreme $T = 0$, the high temperature of $T = 1000$ and for a relatively low formal temperature of $T = 1$. Results are compared in Figs. 6 and 7.
Both for $T \to \infty$ and for $T = 0$, the simulations proceeded as follows: The student was initialized to a random spherical configuration and a set of randomly sampled binary $N = 2$ examples of desired cardinality was generated. Training error is computed, and then in each Monte Carlo Step (MCS), Gaussian noise was added to the 3-vector consisting of $\boldsymbol{W}$ and ω, before normalizing the configuration to a spherical one again. Hereafter, the change in training energy ΔE resulting from this modification in configuration is computed, and the modification is accepted with a probability of $\min\{1, \exp(-\beta\Delta E)\}$. The variance of the Gaussian noise is constant throughout each simulation, and is chosen such that approximately half of all modifications are accepted (except in the limit $T = 0$, in which no modification from the optimal configuration is allowed).

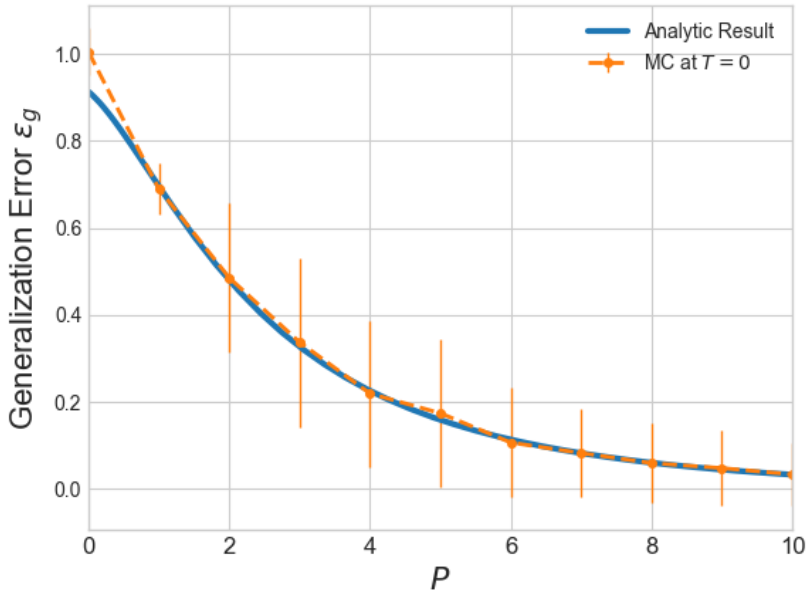


Fig. 7: Generalization error (12) of the $N = 2$ binary features MP classifier against P at $T = 0$. Both analytical results and Monte Carlo simulation results shown (Standard deviations also shown). Note the discrepancy for $P = 0$ results from the assumption $P > 0$ in Eq. (45).

For each configuration, generalization error is readily computed by simply averaging the error generated for each of the 4 possible binary examples for $N = 2$. As a consequence, there are only 4 available levels which the generalization error may occupy for any given configuration, which is reflected in a square-step-like graph of ϵ_g versus MCS (See for instance Fig. 10). However, we average over all MCS, revealing the typical behavior and the interesting trends.
Due to the small size of the system, only 500 MCS were performed, and the first 100 MCS were discarded for equilibration before obtaining the average generalization errors. Results were averaged over 30 simulations for $T = 1000$ and $T = 1$, and 200 simulations for $T = 0$. Excellent agreement with analytic results is observed in the high and low temperature extremes.
The Monte Carlo simulations were also used to obtain the evolution of student ω versus α in the high-temperature limit (Fig. 11, appendix D). Results are as expected and show ω converging asymptotically.

5 Discussion/Outlook

5.1 Challenges of Nonlinearity

In analyzing the statistical mechanics of the MP, it has become clear that the nonlinearities pose a great obstacle to obtaining exact, scalable results for the partition function, and by extent the generalization error. The results provided here have been obtained by considering only the heavily simplified scenario of low N and binary features. Several attempts were made to evaluate the version space in the $T = 0$ limit following a procedure of integration over activation space. This procedure is the field standard for analysis of conventional ANN (See e.g. Refs. 32, 34), where transformation into activation space readily reveals how the performance of the system is governed by a small number of order parameters. In the case of ANN, these order parameters typically quantify the alignment of the configurations of the student and teacher, and it is then not difficult to imagine how they determine the generalization error through geometric relations.
However, for the multiplicative perceptron we have failed to obtain an exact expression in this way due to the nonlinearity of the transformation to activation space. To illustrate the obstacles, the calculation is outlined here:
We adapt Eq. (20) to the case of the MP, and begin with:

$$\begin{aligned}\Omega_p = \langle \Omega(\{\boldsymbol{x}^\gamma\}, \tilde{\boldsymbol{W}}, \tilde{\omega})\rangle_{\mathbb{D}} = \int d\mu(\boldsymbol{W}, \omega) \int \prod^{p} du dv \theta(uv) \\ \times \langle \delta(u - \boldsymbol{W}\cdot\boldsymbol{x} - \sum_{j,k=1}^{N} \omega_{jk} x_j x_k) \\ \times \delta(v - \tilde{\boldsymbol{W}}\cdot\boldsymbol{x} - \sum_{j,k=1}^{N} \tilde{\omega}_{jk} x_j x_k)\rangle_{\mathbb{D}}\end{aligned} \tag{48}$$

Where $\tilde{\boldsymbol{W}}, \tilde{\omega}$ are the configuration of the teacher, which means that u and v are the student and teacher activation respectively. We then rewrite using Eq. (21) to:

$$\begin{aligned}\Omega_p = \int d\mu(\boldsymbol{W}, \omega) \int \prod^{p} \frac{du d\hat{u}}{2\pi} \frac{dv d\hat{v}}{2\pi} \theta(uv) e^{i\hat{u}u + i\hat{v}v} \\ \times \int d\mu(\boldsymbol{x}) \exp\left(-i(\boldsymbol{W}\hat{u} + \tilde{\boldsymbol{W}}\hat{v})\cdot\boldsymbol{x} - i\sum_{j,k=1}^{N} (\omega_{jk}\hat{u} + \tilde{\omega}_{jk}\hat{v}) x_j x_k \right)\end{aligned} \tag{49}$$

Following the convenient and conventional scenario we now assume Eq. (26) applied, and obtain the following N-dimensional Gaussian integral:

$$\begin{aligned}\Omega_p = \int d\mu(\boldsymbol{W}, \omega) \int \prod^{p} \frac{du d\hat{u}}{2\pi} \frac{dv d\hat{v}}{2\pi} \theta(uv) e^{i\hat{u}u + i\hat{v}v} \\ \times \int d\mu(\boldsymbol{x}) \exp\left(-i(\boldsymbol{W}\hat{u} + \tilde{\boldsymbol{W}}\hat{v})\cdot\boldsymbol{x} - i\sum_{j,k=1}^{N} (\omega_{jk}\hat{u} + \tilde{\omega}_{jk}\hat{v}) x_j x_k \right)\end{aligned} \tag{50}$$

Evaluating this integral requires finding the inverse of

$$A_{ij} = \begin{cases} 1, & \text{for } i = j \\ \frac{2i}{\sqrt{N}}(\omega_{ij}\hat{u} + \tilde{\omega}_{ij}\hat{v}), & \text{for } i \neq j \end{cases} \tag{51}$$

which can in principle be done (if we redefine ω and $\tilde{\omega}$ to be symmetric rather than upper triangular), but the result could not be written in a usable form and the procedure would not be scalable to higher N. Moreover, the matrix is not always invertible, and the singular behavior would further complicate computation.
In search for a more feasible approach, another mathematical trick was attempted: The choice evaluate version space by integrating over activation space can be thought of as a method of factoring out the complexity of the transformation which the model imposes on the input features, thus yielding in activation space a very straightforward integral in u_γ, v_γ (22). One may consider repeating a similar transformation prior to evaluating the Gaussian integral over the feature space, and writing:

$$\Omega_p(\tilde{\boldsymbol{W}}, \tilde{\omega}) = \int d\mu(\boldsymbol{W}, \omega) \Bigg[\int \prod_{j=1}^{N} \left(\frac{e^{x_j^2/2} dx_j}{\sqrt{2\pi}} \right) \int du_1 dv_1 d\boldsymbol{u}_2 d\boldsymbol{v}_2 \theta\left(u_1 v_1\right)$$
$$\delta\left(u_1 - (\boldsymbol{W} + \boldsymbol{u}_2) \cdot \boldsymbol{x}\right) \delta\left(v_1 - (\tilde{\boldsymbol{W}} + \boldsymbol{v}_2) \cdot \boldsymbol{x}\right) \delta^N\left(\boldsymbol{u}_2 - \sum_{j=1}^{N} x_j \boldsymbol{\omega}_j\right) \delta^N\left(\boldsymbol{v}_2 - \sum_{j=1}^{N} x_j \tilde{\boldsymbol{\omega}}_j\right) \Bigg]^p \tag{52}$$

where we have now introduced auxiliary variables $u_1, v_1, \boldsymbol{u}_2, \boldsymbol{v}_2$ (where the latter 2 are N-vectors) instead of u, v.
$\boldsymbol{\omega}_j \equiv (\omega_{j1}, \ldots, \omega_{jN})^T$
and we assumed for demonstrative purposes that ω is symmetric and thus

$$\sum_{j=1}^{N} x_j \boldsymbol{\omega}_j \equiv \omega^T \boldsymbol{x} = \omega \boldsymbol{x}$$

Now one might attempt following the procedure of performing the Gaussian integral in feature space, and then evaluating the remaining Gaussian integrals in all the auxiliary variables. Problematically, however, after integrating out one of the auxiliary variables the remaining integral is no longer of Gaussian form. In fact, no method known to us exists for evaluating the integral which results. The sought after expression for the version space of the MP for general N and continuous features remains elusive. More generally, the same applies to the quenched average partition function and any function thereof. Without such expressions, the application of the (offline) statistical mechanical framework is hindered altogether.
Statistical mechanical analysis of nonlinear classifiers is not a wholly novel idea: If a bias is included in the activation (23) the MP is equivalent to a Support Vector Machine (SVM) with a quadratic kernel. As mentioned, an analysis of the statistical mechanics of an SVM with a generic kernel was presented in [37], however the premise here is that an optimal stability training algorithm is used, which is fundamentally different than the situation treated in the present work. Hence, the answers and methods we seek in this work are not found there.

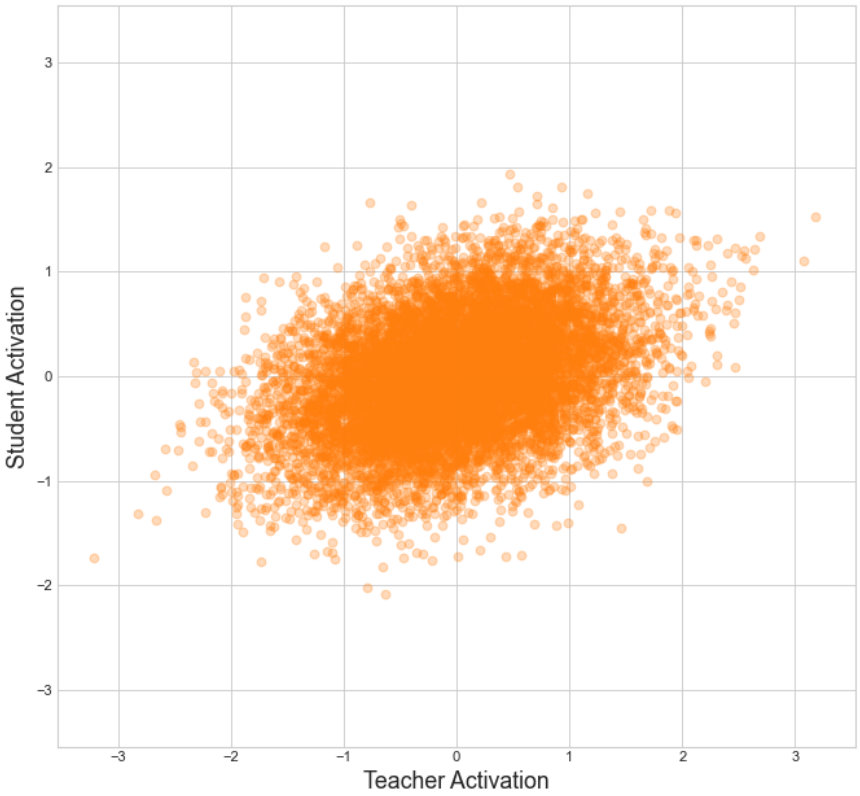


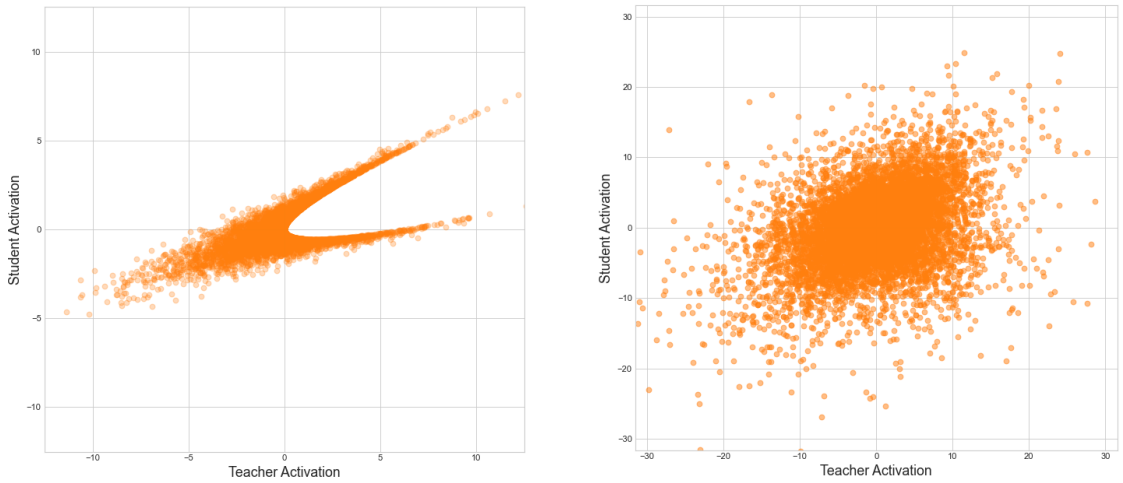
Fig. 8: A plot of Gaussian examples transformed into activation space (20) for a student and teacher perceptron, illustrating the 2-dimensional Normal distributed nature. In the case of perfect alignment between student and teacher, these points lay along the rising diagonal. Illustratively, for $T = 0$, all points in the upper-right and lower-left quarter contribute towards the likelihood that this configuration is included in version space.

5.1.1 The distribution of the activation

In order to further understand the complexity of obtaining the statistical mechanical behavior of this nonlinear system, let us look into the nature of the integral which would result in version space and contrast the situations in the case of a perceptron and an MP.

In the case of the perceptron, the fact that activation is a linear combination of the features allows for a convenient application of the central limit theorem (CLT) to conclude that regardless of the exact nature of the feature's distributions, the activation will follow a distribution which approaches a Normal distribution for large N. Moreover this holds both for the student and the teacher perceptron, which means that distribution in activation space (20) is a two-dimensional Normal distribution, thus determined by a small number of order parameters (specifically the dot product between student and teacher's respective weight vectors) resulting in a situation as illustrated in Fig. 8. This is widely utilized in the treatment of the perceptron, see e.g. Refs. 2, 32, 34.

Upon failing to evaluate the integral for the MP directly, the nature of this distribution was investigated for the case of the MP. Point clouds were plotted as a first inquiry



(a) Some arbitrary $N = 2$ student with XOR-like teacher (29).

(b) Some arbitrary MP student and teacher with $N = 8$.

Fig. 9: Gaussian examples transformed to activation space (20) for illustrative MP scenarios. Note the asymmetry and non-Gaussianity of the distributions.

into the nature of the distribution in activation space, and in the hope of identifying order parameters to describe this distribution. For small N, the non-gaussianity of the distribution in activation space is immediately evident, even in cases where the features were normally distributed (see Fig. 9a). For larger N, this non-gaussianity persists (Fig. 9b), continuing to complicate analysis further.
One might argue that for large N the activation must converge towards a normal distribution, as it is a summation of random variables. However we will see that it does not, as the random variables which are summed to arrive at the activation are in fact not independent, meaning that the CLT does not apply. More generally, the mutual dependence of the summand random variables means that we cannot construct the distribution of the activation even by construction through a cumulant expansion or an Edgeworth series [42]. Perhaps a more detailed treatment of this probability-theoretic approach accounting for the mutual dependence between the summand random variables may give clues as to the large-N behavior of the partition function, however it seems unlikely that exact results for the distribution can be obtained at all, as summations of χ^2 random variables follow no known distribution [43]. Further analysis on the nature of this distribution could be very insightful, but this is beyond the scope of the present work.

5.2 Discussion of results

The analytic results obtained show a convincing correspondence with simulation data, affirming the validity of our work. Generalization error is observed to tend asymptotically to zero with increasing α, qualitatively very similar to the performance of a perceptron in a linear student teacher scenario. However, the curve of the generalization error is characterized as a quotient of exponential functions, resulting in qualitatively different behavior than simple exponential decay as the curve approximates a straight line for low α. The spread of the generalization error across different Monte

Carlo simulations is substantial, revealing that while the expected behavior of the MP is captured well by the exact results, an arbitrary MP's performance may not be assumed to conform to this behavior. This is also in part a consequence of the low N, preventing self-averaging behavior which may be seen in higher-dimensional systems. Fig. 6 also shows how the generalization error behaves differently for lower temperatures ($T = 1$), illustrating that only qualitative observations results are of interest.
While similar behavior is observed in the limit of $T = 0$, the generalization error in this limit is instead characterized by a sum of simple exponential functions in P. Though the analytic expression is invalid for $P = 0$, it is interesting to see how (as also suggested by the MC simulations) the line-approximating behavior seen in the $T \to \infty$ limit is absent here, suggesting fundamentally different generalizing behavior in both limits. It is further worth noting that the spread of the generalization error across the MC simulations are substantially greater in this case than in the $T \to \infty$ limit, suggesting a higher level of dependence on the stochasticity of the examples, especially around the region $2 \leq P \leq 6$. One explanation could be that for a small number of binary features it is largely a matter of chance whether the same example is presented more than once, which would leave version space unaffected. This is in line with the low standard deviation at $P = 1$. This is then an artifact of the simplifications imposed, and does not pertain to the nonlinearity of the system.
The heavy degree of simplification which was necessary in order to obtain exact results limits the applicability of these results. The results were obtained only for binary features. This bypassed the analytical challenges introduced by the nonlinearity, as it replaced nonlinear integrals with summations which could be performed step-by-step. However, it is not proven that the qualitative generalizing behavior revealed in our results are also valid for the case of continuous features, nor in the case of large N; the fundamental scalability of the MP is therefore still unconfirmed. Additionally, it should be noted that while existing literature on off-line training dynamics typically also averages the partition function over all possible teacher configurations we have only considered the illustrative XOR-gate teacher in our results, further limiting the applicability; the initial linear decay in generalization error in the high temperature limit (Fig. 6) may hint at a fundamentally different type of generalizing behavior to be expected when learning with a binary MP, though it has only been confirmed for this nonlinear low-dimensional teacher with binary examples. It may therefore also be an artifact of the low-dimensionality combined with the binary features, or be specific to the XOR gate teacher.
Regardless, no further surprising features were found in the learning behavior of the MP classifier under these simplifications. This suggests that it may be sensible to consider MP-like systems such as dendritic neural networks or in-materio systems in practical applications involving nonlinear behavior, although further research is certainly needed.

5.3 Future research

There are many possible future steps in the investigation of the performance of nonlinear extensions of perceptrons. First, it is possible to perform a similar computation as presented here for binary features for other values of N as well as a larger discrete

set of allowed feature values (rather than just binary features) and other types of teacher configurations. However, the complexity quickly becomes prohibitively impractical. Ideally, perhaps different mathematical avenues may shed light onto the computation of the partition function for continuous (Gaussian) features and/or arbitrary N. For instance, perhaps light can be shed on this by considering the MP as a regular perceptron where the dataset is considered to be structured in some way that is determined by the teacher's configuration, as described in Ref. 44. In any case, if such a method is discovered, analysis may move beyond the low and high temperature limits, and treat the case of arbitrary T, perhaps through application of the replica trick (see e.g. Ref. 34). Further, analysis may evolve towards higher-order nonlinear interactions between features, rather than just the quadratic interactions considered here.

Besides this, the prior distribution of the configurations has been assumed spherical throughout the present work. This has served to explore fundamentally the influence of nonlinearity on the process of statistical mechanical analysis. Yet this, too, may be later extended to a more specific model of the system at hand: the dendritic neural networks would accurately be modeled very differently than in-materio systems, and this would be reflected in this prior distribution.

Another possibility is to consider the on-line training of the MP instead of the off-line training. Throughout the present work we have assumed that some suitable training algorithm exists resulting in a Gibbs-Boltzmann (or some other) distribution, yet much remains unclear about how one may go about training the systems we here model as an MP. For instance, Ref. 19 points out that the conventional backpropagation algorithm is not suited for dendritic neural networks in many cases and applies different training algorithms instead. In the case of in-materio systems, an evolutionary algorithm has been applied [25] and the possibility of applying gradient descent has been proposed [24]. The choice of training method will strongly affect the performance of the model, yet little fundamental understanding of this exists. Statistical mechanics of on-line training may help inform this choice.

A more realistic next step may be to compare the results obtained here with analogs for other types of systems capable of performing the same task. For instance, a simple 2-layer neural network with biases may already realize XOR-gate behavior; generalization error performance may be compared between the MP and such ANN to further explore the advantages and disadvantages of either. Alternatively the influence of including a bias in the MP model may reveal interesting behavior.

6 Conclusion

This work has explored the possibilities for applying the statistical mechanics of learning to gain insight into the behavior of nonlinear extensions of the perceptron. We have sought an exact expression for the generalizing behavior of a multiplicative perceptron in the student teacher scenario. Attempts were made to adapt to the system of interest the mathematical approach of obtaining the partition function by considering integration over respectively the student- and teacher activation. However, the nonlinearity resulted in an integral which could not be evaluated. Neither did the analysis yield a natural set of order parameters describing the system's behavior. Further attempts to obtain insight into the nature of the partition function and any order parameters governing it were made by visualizing the two-dimensional distribution of the student- and teacher activations. This illustrated the difficulty of the analysis, however no conclusions could be drawn from this either.
Finally, a simplified case where only binary two-dimensional examples following logical XOR gate behavior were considered did yield an exact expression for the partition function and generalization error, both in the limit of high and low (formal) temperature. The resulting learning curve in the high temperature limit is characterized as the quotient of exponential functions, while the curve for the low temperature limit is characterized as a sum of exponential functions. The results highlight the complexity of analysis on multiplicative perceptrons subject to mean-squared error, and illustrate the need for different mathematical techniques in order to obtain more general results. Yet, a first result was obtained as to the generalizing behavior of such nonlinear systems, and this offers a means of fundamentally and qualitatively comparing the performance of systems which can be described using the multiplicative perceptron (such as dendritic neural networks and in-materio systems) and other architectures such as multilayer neural networks in certain scenarios. Further, the initial linear decay in the generalization error of the system when learning this nonlinear rule is qualitatively different than that observed for a linear system learning a linear rule, hinting at fundamental difficulties in learning nonlinear behavior based on mean-squared error. This may help motivate further research into the systems which may be modeled as a multiplicative perceptron.

Bibliography

[1] J. M. Richardson, "Pattern recognition and statistical mechanics," *Journal of Statistical Physics*, vol. 1, no. 1, pp. 71–88, 1969.

[2] M. Biehl, N. Caticha, M. Opper, and T. Villmann, "Statistical physics of learning and inference.," in *ESANN*, 2019.

[3] N. Ampazis, S. J. Perantonis, and J. G. Taylor, "Dynamics of multilayer networks in the vicinity of temporary minima," *Neural Networks*, vol. 12, no. 1, pp. 43–58, 1999.

[4] L. Ein-Dor and I. Kanter, "Time series generation by multilayer networks," *Physical Review E*, vol. 57, no. 6, p. 6564, 1998.

[5] M. Gabrié, A. Manoel, C. Luneau, J. Barbier, N. Macris, F. Krzakala, and L. Zdeborová, "Entropy and mutual information in models of deep neural networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2019, no. 12, p. 124014, 2019.

[6] E. Oostwal, M. Straat, and M. Biehl, "Hidden unit specialization in layered neural networks: Relu vs. sigmoidal activation," *Physica A: Statistical Mechanics and its Applications*, vol. 564, p. 125517, 2021.

[7] N. Barkai, H. S. Seung, and H. Sompolinsky, "Scaling laws in learning of classification tasks," *Phys. Rev. Lett.*, vol. 70, pp. 3167–3170, May 1993.

[8] W. Kinzel, "Phase transitions of neural networks," *Philosophical Magazine B*, vol. 77, no. 5, pp. 1455–1477, 1998.

[9] M. Ahr, M. Biehl, and E. Schlösser, "Weight-decay induced phase transitions in multilayer neural networks," *Journal of Physics A: Mathematical and General*, vol. 32, no. 27, p. 5003, 1999.

[10] M. Ahr, M. Biehl, and R. Urbanczik, "Statistical physics and practical training of soft-committee machines," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 10, no. 3, pp. 583–588, 1999.

[11] T. Uezu, "Learning from stochastic rules under finite temperature-optimal temperature and asymptotic learning curve," *Journal of Physics A: Mathematical and General*, vol. 30, no. 22, p. L777, 1997.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] L. Zdeborová and F. Krzakala, "Statistical physics of inference: Thresholds and algorithms," *Advances in Physics*, vol. 65, no. 5, pp. 453–552, 2016.

[14] L. Zdeborová, "Understanding deep learning is also a job for physicists," *Nature Physics*, vol. 16, no. 6, pp. 602–604, 2020.

[15] M. Opper and W. Kinzel, "Statistical mechanics of generalization," in *Models of neural networks III*, pp. 151–209, Springer, 1996.

[16] Q. Wei, R. G. Melko, and J. Z. Chen, "Identifying polymer states by machine learning," *Physical Review E*, vol. 95, no. 3, p. 032504, 2017.

[17] E. Gardner and B. Derrida, "Three unfinished works on the optimal storage capacity of networks," *Journal of Physics A: Mathematical and General*, vol. 22, no. 12, p. 1983, 1989.

[18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[19] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 2, pp. 601–614, 2018.

[20] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1367–1388, 2014.

[21] C. Bartolozzi and G. Indiveri, "Synaptic dynamics in analog vlsi," *Neural computation*, vol. 19, no. 10, pp. 2581–2603, 2007.

[22] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-based neuromorphic systems*. John Wiley & Sons, 2014.

[23] S. Bose, C. P. Lawrence, Z. Liu, K. Makarenko, R. M. van Damme, H. J. Broersma, and W. G. van der Wiel, "Evolution of a designless nanoparticle network into reconfigurable boolean logic," *Nature nanotechnology*, vol. 10, no. 12, pp. 1048–1052, 2015.

[24] M. N. Boon, H.-C. R. Euler, T. Chen, B. van de Ven, U. A. Ibarra, P. A. Bobbert, and W. G. van der Wiel, "Gradient descent in materio," *arXiv preprint arXiv:2105.11233*, 2021.

[25] T. Chen, J. van Gelder, B. van de Ven, S. V. Amitonov, B. de Wilde, H.-C. R. Euler, H. Broersma, P. A. Bobbert, F. A. Zwanenburg, and W. G. van der Wiel, "Classification with a disordered dopant-atom network in silicon," *Nature*, vol. 577, no. 7790, pp. 341–345, 2020.

[26] C. Lawrence, *Evolving Networks To Have Intelligence Realized At Nanoscale*. PhD thesis, UT, Netherlands, May 2018. IDS PhD thesis series no. 18-464 We acknowledge financial support from the MESA+ Institute for Nanotechnology and the CTIT Institute for ICT research, as well as the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement No. 317662.

[27] L. G. Valiant, "Evolvability," *Journal of the ACM (JACM)*, vol. 56, no. 1, pp. 1–21, 2009.

[28] M. Straat and M. Biehl, "On-line learning dynamics of relu neural networks using statistical physics techniques," *arXiv preprint arXiv:1903.07378*, 2019.

[29] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli, "Statistical mechanics of deep learning," *Annual Review of Condensed Matter Physics*, 2020.

[30] D. Saad and S. A. Solla, "On-line learning in soft committee machines," *Physical Review E*, vol. 52, no. 4, p. 4225, 1995.

[31] Y. Yoshida, R. Karakida, M. Okada, and S.-i. Amari, "Statistical mechanical analysis of online learning with weight normalization in single layer perceptron," *Journal of the Physical Society of Japan*, vol. 86, no. 4, p. 044002, 2017.

[32] A. Engel and C. Van den Broeck, *Statistical Mechanics of Learning*. Cambridge University Press, 2001.

[33] T. L. H. Watkin, A. Rau, and M. Biehl, "The statistical mechanics of learning a rule," *Rev. Mod. Phys.*, vol. 65, pp. 499–556, Apr 1993.

[34] H. S. Seung, H. Sompolinsky, and N. Tishby, "Statistical mechanics of learning from examples," *Physical review A*, vol. 45, no. 8, p. 6056, 1992.

[35] S. J. Blundell and K. M. Blundell, *Concepts in thermal physics*. Oxford University Press on Demand, 2010.

[36] S. A. Solla and E. Levin, "Learning in linear neural networks: The validity of the annealed approximation," *Physical Review A*, vol. 46, no. 4, p. 2124, 1992.

[37] R. Dietrich, M. Opper, and H. Sompolinsky, "Statistical mechanics of support vector networks," *Physical review letters*, vol. 82, no. 14, p. 2975, 1999.

[38] M. Evans, N. Hastings, and B. Peacock, "Statistical distributions," 2001.

[39] M. Minsky and S. Papert, "An introduction to computational geometry," *Cambridge tiass., HIT*, vol. 479, p. 480, 1969.

[40] W. R. Inc., "Mathematica, Version 13.0.0." Champaign, IL, 2021.

[41] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[42] J. E. Kolassa, *Series approximation methods in statistics*, vol. 88. Springer Science & Business Media, 2006.

[43] J. Bausch, "On the efficient calculation of a linear combination of chi-square random variables with an application in counting string vacua," *Journal of Physics A: Mathematical and Theoretical*, vol. 46, no. 50, p. 505202, 2013.

[44] S. Goldt, M. Mézard, F. Krzakala, and L. Zdeborová, "Modeling the influence of data structure on learning in neural networks: The hidden manifold model," *Physical Review X*, vol. 10, no. 4, p. 041044, 2020.

Appendices

A Derivation of coefficients for different scenarios for version space

The coefficients presented in Eqs. 45 and 46 were derived as follows:
We begin with the binomial theorem, which can be stated as:

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k} \tag{53}$$

It then follows that whenever $P \geq 1$:

$$[A+B+C+D]^P = \sum_{j=0}^{P} \binom{P}{j} \left(\sum_{k=0}^{j} \binom{j}{k} A^k B^{j-k} \right) \left(\sum_{l=0}^{P-j} \binom{P-j}{l} C^l D^{P-j-l} \right) \tag{54}$$

We now use the fact that $A^{n\geq 1} = A$, $A^0 = 1$ (and the same holds for B, C and D) to rewrite Eq. 54 into qualitatively different terms as:

$$\begin{aligned} &= \sum_{l=0}^{P} \binom{P}{l} C^l D^{P-l} \\ &+ \sum_{l=0}^{P} \binom{P}{l} A^l B^{P-l} \\ &+ \sum_{j=1}^{P-1} \binom{P}{j} \left(\sum_{k=0}^{j} \binom{j}{k} A^k B^{j-k} \right) \left(\sum_{l=0}^{P-j} \binom{P-j}{l} C^l D^{P-j-l} \right) \end{aligned} \tag{55}$$

This step of isolating and simplifying summand terms is repeated until the coefficients in Eq. 46 are obtained.

B Exact expression for ϵ_g for $N = 2$ binary features in the $T \to \infty$ limit

The full expression for the generalization error plotted in Fig. 6 is included here for completeness. Note the decimal numbers are rounded values of exact expressions, meant to make character of the expression more readable:

$$\epsilon_g(\alpha) = \frac{5.44e^{-4\alpha} + 6.62e^{-3\alpha} + 10.87e^{-2\alpha} + 2.21e^{-\alpha}}{2.71 + 2.71e^{-4\alpha} + 4.41e^{-3\alpha} + 10.87e^{-2\alpha} + 4.41e^{-\alpha}} \tag{56}$$

For the very interested reader, the exact expression is given as:

```
Eg := -0.5 D[Log[Zann], α];
Eg
```

Out[109]=

$$-\left(\left(0.5\times\left(-8\sqrt{2}\left(\sqrt{2}-\frac{2}{\sqrt{3}}\right)e^{-4\alpha}\pi+16\,e^{-2\alpha}\,\text{ArcCot}\left[2\sqrt{2}\right]+2\,e^{-2\alpha}\left(\sqrt{2}\,e^{\alpha}\pi-8\,e^{\alpha}\,\text{ArcCot}\left[2\sqrt{2}\right]\right)-\right.\right.\right.$$
$$4\,e^{-2\alpha}\left(\sqrt{2}\,e^{\alpha}\pi+8\,\text{ArcCot}\left[2\sqrt{2}\right]-8\,e^{\alpha}\,\text{ArcCot}\left[2\sqrt{2}\right]\right)-$$
$$6\,e^{-3\alpha}\left(\sqrt{2}\,\pi-8\,\text{ArcCot}\left[2\sqrt{2}\right]+8\,e^{\alpha}\,\text{ArcCot}\left[2\sqrt{2}\right]\right)-$$
$$\frac{2}{3}e^{-4\alpha}\left(3\sqrt{2}\,e^{\alpha}\pi-24\,e^{\alpha}\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]\right)+\frac{2}{3}e^{-\alpha}\left(2\sqrt{6}\,e^{\alpha}\pi-24\,e^{\alpha}\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]\right)+$$
$$\frac{8}{3}e^{-4\alpha}\left(-2\sqrt{6}\,\pi+3\sqrt{2}\,e^{\alpha}\pi+24\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]-24\,e^{\alpha}\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]\right)-$$
$$\left.\left.\left.\frac{2}{3}e^{-\alpha}\left(-3\sqrt{2}\,\pi+2\sqrt{6}\,e^{\alpha}\pi+24\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]-24\,e^{\alpha}\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]\right)\right)\right)\Big/$$
$$\left(2\sqrt{2}\left(\sqrt{2}-\frac{2}{\sqrt{3}}\right)\pi+2\sqrt{2}\left(\sqrt{2}-\frac{2}{\sqrt{3}}\right)e^{-4\alpha}\pi+\right.$$
$$2\,e^{-2\alpha}\left(\sqrt{2}\,e^{\alpha}\pi+8\,\text{ArcCot}\left[2\sqrt{2}\right]-8\,e^{\alpha}\,\text{ArcCot}\left[2\sqrt{2}\right]\right)+$$
$$2\,e^{-3\alpha}\left(\sqrt{2}\,\pi-8\,\text{ArcCot}\left[2\sqrt{2}\right]+8\,e^{\alpha}\,\text{ArcCot}\left[2\sqrt{2}\right]\right)-$$
$$\frac{2}{3}e^{-4\alpha}\left(-2\sqrt{6}\,\pi+3\sqrt{2}\,e^{\alpha}\pi+24\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]-24\,e^{\alpha}\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]\right)+$$
$$\left.\left.\frac{2}{3}e^{-\alpha}\left(-3\sqrt{2}\,\pi+2\sqrt{6}\,e^{\alpha}\pi+24\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]-24\,e^{\alpha}\,\text{ArcTan}\left[\frac{1}{\sqrt{2}}\right]\right)\right)\right)$$

Where 'Zann' is the (annealed) partition function as computed following Eq. 34.

C Exact expression for ϵ_g for $N = 2$ binary features in the $T = 0$ limit

The full expression for the generalization error ϵ_g plotted in Fig. 7 is a sum of exponential functions in P, and it is given exactly as:

$$
\begin{aligned}
&\frac{2^{2-2P}\,\mathrm{ArcCot}\left[2\sqrt{2}\right]}{\pi}
- \frac{3\times 2^{3-2P}\,\mathrm{ArcCot}\left[2\sqrt{2}\right]}{4\pi - 4\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 12\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]} \\
&+ \frac{3\times 2^{3-P}\,\mathrm{ArcCot}\left[2\sqrt{2}\right]}{4\pi - 4\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 12\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}
- \frac{2^{3-2P}\times 3^{P}\,\mathrm{ArcCot}\left[2\sqrt{2}\right]}{4\pi - 4\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 12\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]} \\
&- \frac{2^{5-2P}\,\mathrm{ArcCot}\left[2\sqrt{2}\right]}{4\pi - 8\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]} + \frac{2^{4-P}\,\mathrm{ArcCot}\left[2\sqrt{2}\right]}{4\pi - 8\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}
+ \frac{2^{4-2P}\,\mathrm{ArcCot}\left[2\sqrt{2}\right]}{4\pi - 8\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 8\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]} \\
&- \frac{2^{3-P}\,\mathrm{ArcCot}\left[2\sqrt{2}\right]}{4\pi - 8\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 8\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}
+ \frac{3\times 2^{2-2P}\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}{\pi} \\
&+ \frac{3\times 2^{3-2P}\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}{4\pi - 4\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 12\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}
- \frac{3\times 2^{3-P}\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}{4\pi - 4\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 12\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]} \\
&+ \frac{2^{3-2P}\times 3^{P}\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}{4\pi - 4\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 12\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}
- \frac{2^{5-2P}\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}{4\pi - 8\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]} + \frac{2^{4-P}\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}{4\pi - 8\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]} \\
&- \frac{2^{4-2P}\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}{4\pi - 8\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 8\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}
+ \frac{2^{3-P}\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}{4\pi - 8\,\mathrm{ArcCot}\left[2\sqrt{2}\right] - 8\,\mathrm{ArcTan}\left[\frac{1}{\sqrt{2}}\right]}
\end{aligned}
$$

D Monte Carlo simulations: Evolution of ω versus α

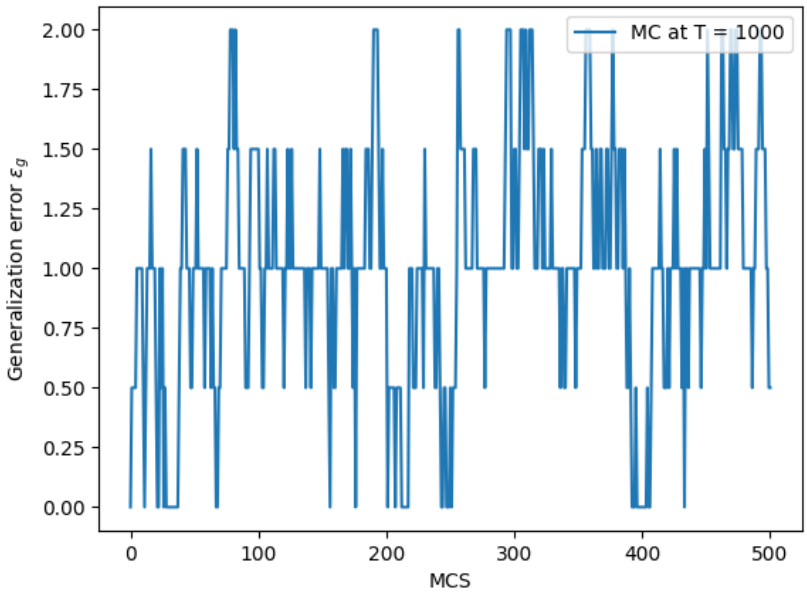


Fig. 10: Generalization error ϵ_g (approximately equal to the training error per example ϵ_t) throughout the steps of a single Monte Carlo simulation at high temperature.

In all MC simulations performed, only 4 levels of energy (training error) were available, due to the low-dimensional and discrete nature of the examples. For a single MC simulation, this resulted in step-like transitions in the energy, as illustrated in Fig. 10.

Further MC simulations were performed for the $N = 2$ binary features MP classifier and the evolution of ω was recorded (Fig. 11). This shows how the student gradually learns to become more nonlinear. Note how ω stabilizes before reaching the teacher's value of -1, as perfect classifying behavior may already be reached sooner due to the discrete set of possible examples. The apparent peak around $\alpha = 1.5$ seems incidental and falls well within the standard deviation. The plot shows no further surprising features, though it nicely illustrates the gradual way in which nonlinear rules may be learned.

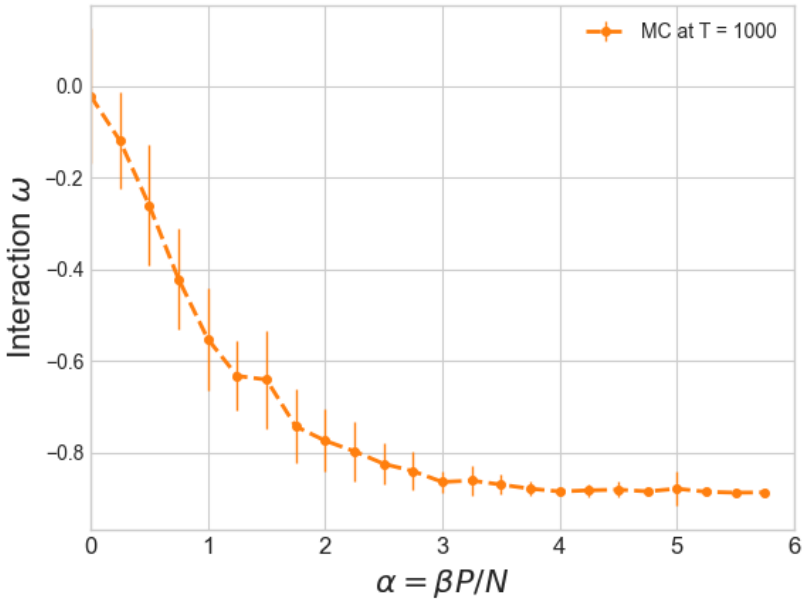


Fig. 11: Nonlinear interaction parameter ω of the student $N = 2$ binary features MP classifier versus α (Eq. 18), according to Monte Carlo Simulation (Standard deviations also shown).

The python and Mathematica source code generating all simulations, plots and analytic results can be found at https://github.com/Ushwald/MasterThesisCode.