Supervised Learning & Bayesian inference

Andrea Perin, 1206078, Carlo Polato, 1205688, and Vincenzo Maria Schimmenti, 1204565

(Dated: 14 June 2019)

In this exercise, we performed some classification tasks on 2-D Ising configurations, with the aim of distinguishing between ordered and disordered configurations which were sampled from the subcritical, critical and supercritical regimes. Classical deep learning techniques and a Bayesian approach were explored and compared.

I. PRELIMINARIES

The dataset was collected from Pankaj Mehta's review. It consists of a set of $N = 160000$ grids of 40×40 spins, representative of a finite 2-D Ising configurations in a temperature range of $[0.25, 4]$.

II. TASKS 1-2: SUPERVISED LEARNING BY WAY OF NEURAL NETWORKS

The first task was to classify the states using a neural network (NN). This approach relies on supervised learning, meaning that we provide the network with a set of pairs $\{\mathbf{X}; y\}$ where $\mathbf{X}$ is the configuration (a vector of 40×40 spins, meaning that $\mathbf{X} \in \{-1, 1\}^{40\times 40}$) and y is a label, corresponding to either being ordered or disordered.

In this framework, the neural network will learn the mapping between a given configuration and the corresponding label.

Different architectures are explored, both in terms of number of neurons and hidden layers, however the best results are achieved with just one hidden layer; this can be explained by the fact that we are dealing with a rather simple task (with respect to the neural network power). An important distinction, however, lies in the choice of the data that is fed to the networks: at first, the networks are trained on configurations sampled exclusively from either sub or supercritical regimes. By doing so, the networks have no experience of the data that is sampled from critical configurations resulting in an expected worse performance. Then, the networks are trained on the whole dataset, including also the configurations sampled from the critical regime.

For the restricted dataset, the performances of the various architectures are collected in fig 1.

For the best performing architecture we choose the one with $N = 50$ hidden neurons and one hidden layer and we plot the relative ROC curve in figure 2. Our results are convincingly close to an ideal classifier, having an AUC of 0.99 for train and test and 0.96 for critical.

1

For the whole dataset the results are given in figure 3. We chose the same best architecture as before obtaining an AUC score of 0.99 for both train and test (we stress again that here the critical regime configurations are included both in train and test sets). The ROC is shown in 4.

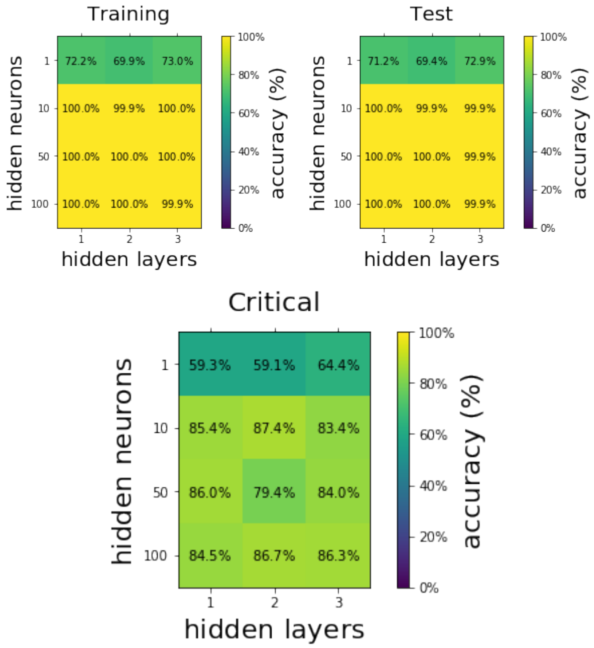

FIG. 1. Accuracy tables for restricted dataset

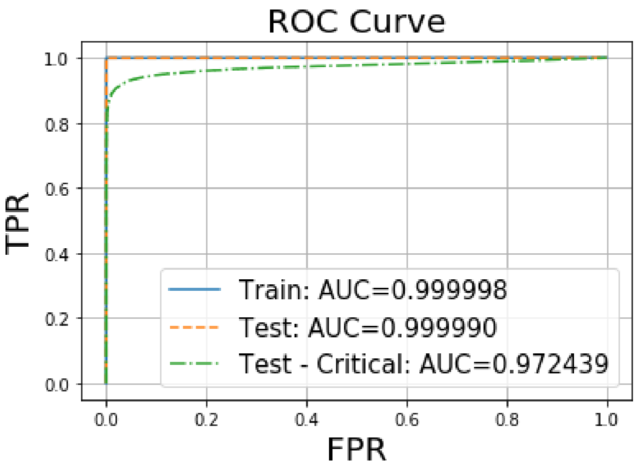

FIG. 2. ROC Curve for restricted dataset. Notice how the train and test overlap and are almost perfect while the critical one is performing worse.

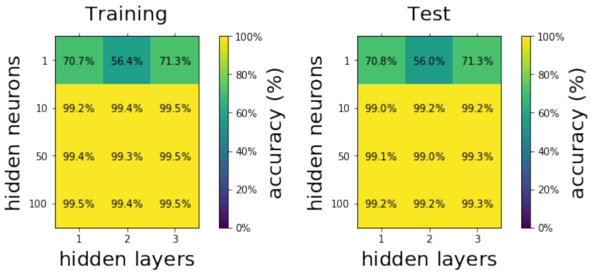

FIG. 3. Accuracy tables for full dataset

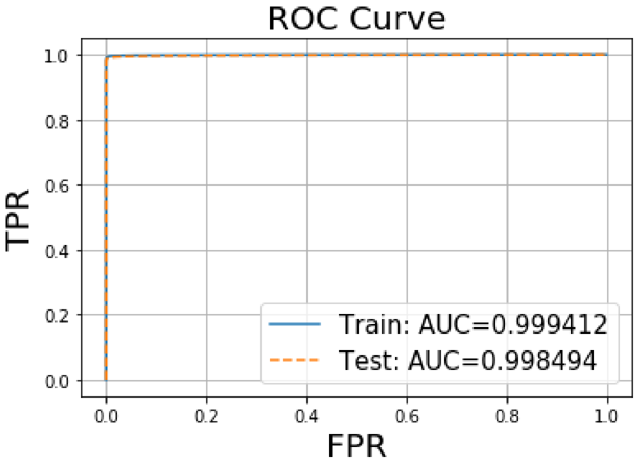

FIG. 4. ROC Curve for full dataset. Notice how the train and test overlap and are almost perfect, showing no overfitting.

III. TASK 3: BAYESIAN APPROACH

In the previous section, in order to classify the phase of an Ising configuration, we used directly the configurations (i.e., the spin values); the employed model is way too complicated for the task at hand (we know how easy is to classify an Ising Model if one knows the temperature), so a good idea would be to use physical arguments to reduce the dimensionality and classify the configurations using this restricted number of variables. The first idea, in this setting, would be to use the probability distribution for the Ising state and learn the ferromagnetic coupling constant J using a maximization procedure, even incorporating a prior distribution on it:

$$P(J|\sigma) = \frac{P(\sigma|J)P(J)}{P(\sigma)} = \frac{1}{P(\sigma)}\frac{e^{J\sum_{\langle i,j\rangle}\sigma_i\sigma_j}}{Z(J)}P(J)$$
$$J^\star = \arg\max_J \{\log P(J|\sigma)\} \implies$$
$$\frac{\partial \log P(J)}{\partial J} + \left\langle \sum_{\langle i,j\rangle}\sigma_i\sigma_j \right\rangle_{data} - \left\langle \sum_{\langle i,j\rangle}\sigma_i\sigma_j \right\rangle_{model} = 0$$

The resulting estimate, $J^\star$, is then the best approximation attainable by employing this Bayesian framework.

The problem here does not lie on the choice of the prior distribution, but instead on the estimation of the model expected value $\langle\sum_{\langle i,j\rangle}\sigma_i\sigma_j\rangle_{model}$: this must be either extracted from the analytical expression (which is not available to us) or from Monte Carlo simulations (which is too computationally requiring for the task we are trying to solve).

To circumvent the problem, one should completely change strategy. The idea we are proposing here is to extract two physically meaningful quantities from the system: the two point correlation function (not the connected one) and the absolute value for the magnetization:

$$m \equiv \frac{1}{L^d z}\left|\sum_i \sigma_i\right|$$

$$g \equiv \frac{1}{L^d z}\sum_{\langle i,j\rangle}\sigma_i\sigma_j$$

These two quantities are invariant under $\mathbb{Z}^2$ symmetry, so they are suited to separate the two phases: the quasi-uniform configurations, composed of either almost all $+1$ or -1 states, are mapped to the same variables, and we expect that for values of g greater than some critical g_c, the model is ordered, otherwise disordered. The way we choose to classify the phases is by means of Logistic Regression, where we impose some prior on the model parameters (an approach that is effectively analogous to regularization in machine learning). We denote the two variables (m, g) collectively as $Z = (m, g)$, and the labels for the phase by $y = \{-1, 1\}$ (1 is ordered, -1 disordered). Again, we assume a logistic model:

$$P(y = 1|Z, w, w_0) = \frac{1}{1 + e^{-y(\langle w,Z\rangle + w_0)}}$$

For each of the parameters w_0 and $w = (w_1, w_2)$ we assume a gaussian prior with precision $\lambda > 0$. The optimal prior parameter is chosen using a grid search procedure (i.e. maximization of the posterior and picking the best λ). Results for the incomplete dataset are collected in figure 5, while the ones for the complete dataset can be found in figure 6.

By following this approach, we are effectively introducing an inductive bias in our problem. The concept of criticality is deeply linked to that of correlation: our physical intuitions are appropriate enough to allow a simpler algorithm, such as a logistic regression, to outperform a comparatively much more complicated model, as a neural network.

IV. BONUS TASK: AN UNSUPERVISED APPROACH

Staying in the spirit of neural networks, we try to take an unsupervised approach to the task of classifying the phase of a configuration. We use for this task an Autoencoder with two hidden neurons and we try to see if,

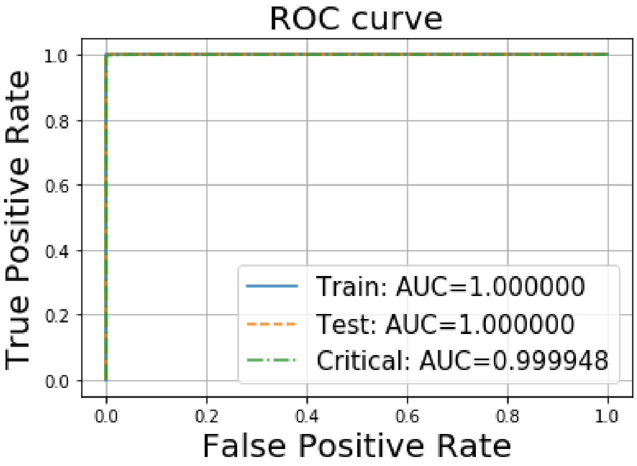

FIG. 5. ROC Curve for the incomplete dataset, using the correlation-based approach. The curves overlap almost perfectly. The AUC score certifies that the predictor has a perfect (up to floating point precision) behavior over training and test datasets. The score over the critical part of the dataset, while not quite reaching the same level, is still remarkably high.

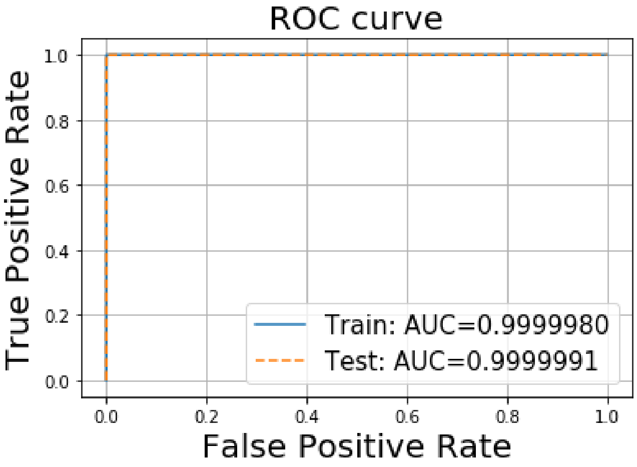

FIG. 6. ROC Curve for full dataset, using the correlation-based approach. Once again, the two curves show an almost complete overlap, and the AUC score differs from 1 only at the 6^{th} decimal. It is worth noting that the performance is actually better on the test set than on the training set, even if for a very small quantity.

by looking at the internal representation, one can distinguish the phases; we repeat the same separation as before: first we use the restricted dataset, then the full one. We plot in the following figures the internal representation for the data and we color them by their true labels: we observe how the internal representation performs a clustering on the configurations, making the two phases visible. Again, using the restricted dataset, we have problems in the critical regime, as can be seen in figure 7.

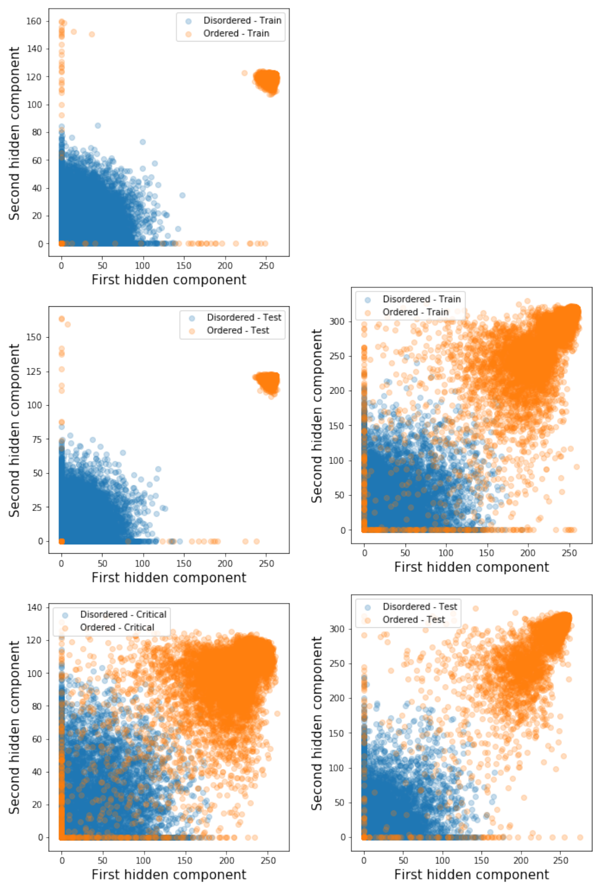

FIG. 7. Clustering and classification using autoencoders: restricted dataset on the left and full on the right. Notice how the autoencoder that was trained on the whole dataset is able to cluster in a more distinct fashion.

[1] About ROCs: an ideal classifier (i.e., one that can ‘see through’ the noise) has an ROC curve that looks like a step function, meaning that the true positive rate is 1, independent of the false positive rate. The quality of an ROC curve can also be evaluated by looking at its underlying area, named AUC (area under the curve), which should be 0.5 for a random classifier and 1 for a perfect one.