Neural Networks for Universal Analytic Continuation of Response Functions

Simon Verret
Département de physique
Institut Quantique
Université de Sherbrooke
Mila, Université de Montréal

Reza Nourafkan
Département de physique
Institut Quantique
Université de Sherbrooke

Quinton Weyrich
Département de physique
Institut Quantique
Université de Sherbrooke

Samuel Desrosiers
Département de physique
Institut Quantique
Université de Sherbrooke

A.-M. S. Tremblay
Département de physique
Institut Quantique
Université de Sherbrooke

Abstract

Recent work demonstrated that Neural Network can outperform Maximum Entropy methods for the analytical continuation of noisy Matsubara Green's function in many-body physics. Here we generalize this approach to the conductivity response functions and we introduce a rescaling procedure that help trained networks to generalize to all temperatures.

1 Introduction

In many-body quantum physics, many material properties such as the electrical conductivity, the thermoelectric power, etc. can be expressed as linear response functions. In the most difficult settings, sophisticated quantum Monte Carlo (QMC) methods are required to compute these response functions. Unfortunately, these methods are often formulated on the imaginary time axis, and recovering the real-time response functions from numerical data in imaginary time is a highly ill-conditioned inverse problem known as "analytic continuation" [1]. Recent work shows that supervised learning methods [2], and more particularly deep neural networks [3, 4, 5] can outperform standard approaches to analytic continuation such as Maximum Entropy [1, 6]. Indeed, thanks to the stability of the forward problem, it is easy to generate as many training examples as desired to get very accurate models. So far, the literature focuses on the analytic continuation of one-particle Green's functions, which are different from the response functions for conductivity. More importantly, analytic continuation has been performed for single values of temperature and very specific synthetic training sets. Neural networks trained this way do not generalize to other temperatures and more realistic data. In this work, we train neural networks for the analytic continuation of the conductivity, and we introduce a rescaling procedure that improves generalization to temperatures out of the training set.

The task we consider is to inverse the following integral equation, which is encountered in linear response theory for the conductivity [7],

$$\Pi(i\omega_n) = \int_{-\infty}^{\infty} \frac{d\omega}{\pi} \frac{\omega^2}{\omega^2 + \omega_n^2} \mathrm{Re}\{\sigma(\omega)\}. \tag{1}$$

We seek to recover the real-part of the conductivity function $\mathrm{Re}\{\sigma(\omega)\}$, which depends on real frequency ω, from the response function $\Pi(i\omega_n)$, calculated on discrete Matsubara frequencies $i\omega_n = i2\pi n/\beta$ with inverse temperature $\beta = 1/T$ and $n \in \{0, 1, 2, \cdots\}$. Note that we work in units where Boltzmann's constant k_B and Planck's constant $\hbar$ are equal to unity.

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

In practice, we give to the neural network an input vector $\mathbf{\Pi} \in \mathbb{R}^N$ with components $\Pi_n = \Pi(i\omega_n)$ for the first $N = 128$ Matsubara frequencies, and it outputs a vector $\hat{\boldsymbol{\sigma}} \in \mathbb{R}^M$ targeting components $\sigma_m = \text{Re}\{\sigma(\omega_m)\}$ sampled on a grid of $M = 512$ real frequencies (illustrated in Fig. 1). The Matsubara frequencies are $\omega_n = 2\pi n/\beta$ with $\{n = 0, 1, ..., N\}$ and the inverse temperature β to be specified; and the real frequencies are $\omega_m = \omega_{\text{max}} m/M$ with $m \in \{0, 1, ..., M\}$ and ω_{max} to be specified. The neural network has no access to frequencies, only the sampled input $\mathbf{\Pi}$ and target $\boldsymbol{\sigma}$.

Our main goal is to make the training procedure independent of temperature and energy scale. In other words, we want neural networks trained on data with certain ω_{max} and β to perform well on data with different ω_{max} and β. Since QMC methods can only provide noisy estimates of $\Pi(i\omega_n)$, we verified that our conclusions are robust to input noise with variance 10^{-5}, 10^{-3}, and 10^{-2}, as in references [4, 5].

2 Datasets

We consider three datasets (F, G and B), of different complexity, each composed of 100 000 random synthetic conductivity functions $\sigma(\omega)$ (see below) for training plus 10 000 for validation. For each $\sigma(\omega)$, five input vectors $\mathbf{\Pi}$ are computed by numerical integration of equation (1), one for each inverse temperature $\beta \in \{10, 15, 20, 25, 30\}$. Then, two target vectors $\boldsymbol{\sigma}$ are sampled from $\sigma(\omega)$, one using the default ω_{max} of the dataset (see below) and the other using the rescaled ω_{max} described in section 4, specific to each function.

Gaussian peaks A first way to generate random $\sigma(\omega)$ is a weighted sum of Gaussian distributions

$$\tilde{\sigma}(\omega) = \frac{1}{R}\sum_{r=1}^{R} \frac{\pi A_r}{\sqrt{2\pi}\sigma_r} \exp\Big(-\frac{1}{2\sigma_r^2}(\omega - \omega_r)^2\Big), \tag{2}$$

parametrized by random weights A_r, centers ω_r, and widths σ_r. The conductivity has to be even, which is ensured through $\text{Re}\{\sigma(\omega)\} = \frac{1}{2}(\tilde{\sigma}(\omega) + \tilde{\sigma}(-\omega))$. Note that the factor π cancels the one in (1) so that the contribution of each peak to the normalization $\int \frac{d\omega}{\pi}\sigma(\omega)$ is precisely A_r. All previous works on analytic continuation with machine learning have used such sets of Gaussian peaks [2, 3, 4, 5].

We denote our first dataset $\sigma(\omega) \sim F$. It reproduces the one of Fournier et al. [4, 5], with default $\omega_{\text{max}} = 15$ and uniformly distributed number of peaks $R \in [1, 21]$, centers $\omega_r \sim [0, 6]$ except for $\omega_1 \sim [0, 0.5]$, and widths $\sigma_r \in [0.1, 1]$. In the original definition, the prefactor is removed ($A_r \propto \sigma_r$) which favors rounded structures. Resulting conductivities are normalized to $\int \frac{d\omega}{\pi}\sigma(\omega) = 1$.

We denote our second dataset $\sigma(\omega) \sim G$. It is intended to better represent physical conductivities, which are usually found with a sharp Drude peak at $\omega = 0$ (conductors) or without such a peak (insulators), with or without other structures at higher ω. This is realized for $\omega_{\text{max}} = 20$, by uniformly sampling one to four Drude peaks with $\omega_r = 0$, $\sigma_r \in [0.2, 4]$, and $A_r \in [0, 1]$ with an additional set of one to six finite frequency peaks with $\omega_r \sim [4, 16]$, $\sigma_r \in [0.4, 4]$, and $A_r \in [0, 1]$. Again, the conductivities are normalized so that $\int \frac{d\omega}{\pi}\sigma(\omega) = \sum_r A_r = 1$, but this time with truly random A_r.

Beta peaks A second way to generate random $\sigma(\omega)$ is a weighted sum of Beta distributions

$$\tilde{\sigma}(\omega) = \frac{1}{R}\sum_{j=1}^{R} \frac{\pi A_r}{\tilde{\sigma}_r} \text{Beta}\Big(\frac{\omega - \tilde{\omega}_r}{\tilde{\sigma}_r}; a_r, b_r\Big), \tag{3}$$

parametrized by random amplitudes A_r, centers ω_r, widths σ_r and random parameters a_r and b_r for the Beta distribution. Since Beta $(x; a_r, b_r)$ is non-zero only for $0 \leq x \leq 1$, the centers ω_r and the widths σ_r of the peaks must be adjusted to the distribution's natural mean and standard deviation as

$$\tilde{\omega}_r = \omega_r + \sigma_r \frac{a_r}{a_r + b_r}, \qquad \tilde{\sigma}_r = \sigma_r \sqrt{\frac{(a_r + b_r + 1)(a_r + b_r)^2}{a_r b_r}}. \tag{4}$$

We thus generate a third dataset $\sigma(\omega) \sim B$ with the same weights, centers, and widths, as for the $\sigma(\omega) \sim G$ dataset, but using Beta distributions with uniformly sampled $a_r, b_r \in [0.5, 10]$. Note that

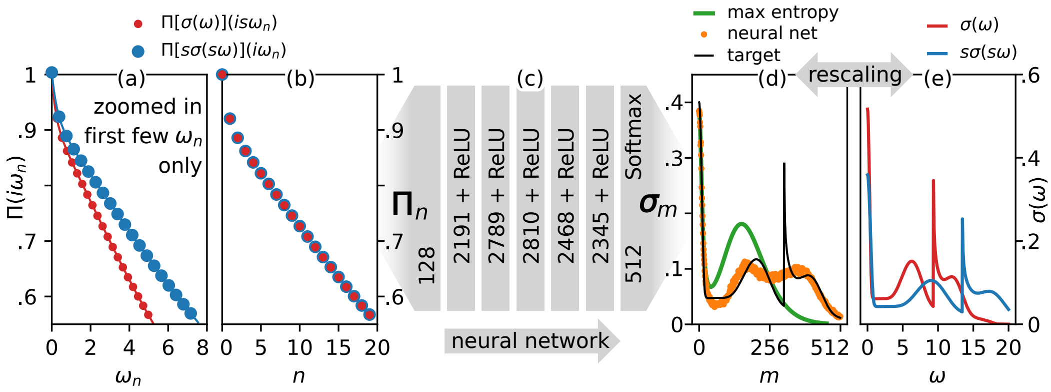

Figure 1: (a) Response functions for two conductivities ($\sigma(\omega)$ and $s\sigma(s\omega)$) on two Matsubara frequency grids (two temperatures β and $s\beta$). (b) As a function of n, they both lead to the same vector $\mathbf{\Pi}$. (c) Illustration of one of the fully connected neural networks used to perform analytic continuation. (d) Comparison of the neural network prediction $\hat{\boldsymbol{\sigma}}$ and maximum entropy prediction [6] with the target $\boldsymbol{\sigma}$, taken from the $\sigma(\omega) \sim B$ dataset, and prepared with the rescaling procedure of section 4. (e) $\sigma(\omega)$ and $s\sigma(s\omega)$, which are contradictory targets.

for large $a_r = b_r$, these adjusted Beta distributions become very similar to Gaussian distributions (except for the tails, which are always bounded within $0 \leq \frac{\omega - \tilde{\omega}_r}{\tilde{\sigma}_r} \leq 1$), and we observed that neural networks trained on Beta peaks generalize without problem to Gaussian peaks.

Beta peaks can be very sharp, enabling more diversity in the dataset. In many cases, sharp structures are impossible to recover with analytic continuation, because too little information is contained in $\mathbf{\Pi}$, especially at high temperature (low β). Such sharp structure can give trouble to maximum entropy methods [6], and neural networks might help in those cases, by learning to ignore them (see Fig. 1(d)).

3 Invariances

The integral (1) is invariant in two important ways that play to our advantage. We can make them explicit by writing the $\Pi(i\omega_n)$ integral as a functional $\Pi[\sigma(\omega)](i\omega_n)$. The first invariance is that Π and σ are proportional (one can take a constant α in and out of the integral),

$$\Pi[\alpha\sigma(\omega)](i\omega_n) = \alpha\Pi[\sigma(\omega)](i\omega_n), \tag{5}$$

so a model that works for input $\mathbf{\Pi}^{\text{ref}}$ can be used for any proportional input $\mathbf{\Pi} = \alpha\mathbf{\Pi}^{\text{ref}}$, given that the latter is properly renormalized. $\Pi_0 = \Pi[\sigma(\omega)](i\omega_n = 0)$ provides a natural reference amplitude to do this, because it corresponds to the normalization of $\text{Re}\{\sigma(\omega)\}$,

$$\Pi_0 = \int_{-\infty}^{\infty} \frac{d\omega}{\pi} \text{Re}\{\sigma(\omega)\}. \tag{6}$$

We thus train the model exclusively on data for which $\Pi_0 = 1$ without loss of generality; unnormalized inputs can be normalized as $\mathbf{\Pi}/\Pi_0$ and their outputs can be recovered as $\Pi_0\boldsymbol{\sigma}$.

The second invariance is that $\mathbf{\Pi}$ is unchanged for combined rescaling of the conductivity $\sigma(\omega) \to s\sigma(s\omega)$ and Matsubara frequencies $\omega_n \to \omega_n/s$ for any constant s. This is seen with a simple change of variable $\omega \to s\omega$ in (1),

$$\Pi[\sigma(\omega)](is\omega_n) = \Pi[s\sigma(s\omega)](i\omega_n) = s\Pi[\sigma(s\omega)](i\omega_n). \tag{7}$$

Thus, with the ω_n grid unspecified, a single input vector $\mathbf{\Pi}$ can correspond to many functions $s\sigma(s\omega)$, as illustrated in Fig. 1(a), 1(b) and 1(e). Since the ω_n grid spacing is proportional to temperature, working with many values of β at training leads to contradictory examples in the training set since the neural network has no access to frequency or temperature.

This can be used as an advantage. Indeed, equation (7) shows that an algorithm trained at fixed β^{ref} (fixed ω_n grid) can be used for other $\beta = s\beta^{\text{ref}}$ as long as the output is reinterpreted as

$[\boldsymbol{\sigma}]_m = (\beta/\beta^{\text{ref}})\sigma((\beta/\beta^{\text{ref}})\omega)$. The performance at this new β will depend on the diversity of spectra in the training set for that new scale $s = (\beta/\beta^{\text{ref}})$. As one can see in Fig. 1(a), s corresponds to the width (scale) of $\sigma(\omega)$ in frequency, so generating the random $\sigma(\omega)$ of section 2 inherently covers multiple values of s. Neural networks trained on these datasets should therefore be usable at other temperatures with variable performance. But is it ideal?

4 Rescaling procedure

Ideally, the ω_n grid, the structure of $\sigma(\omega)$, and the real-frequency width (or scale s) should all be controlled separately in the training set. Even though random spectra will come with effectively random real-frequency widths, we can control the latter a posteriori by choosing a cutoff such that the real-frequency grid $[\boldsymbol{\sigma}]_m = \text{Re}\{\sigma(\frac{m}{M}\omega_{\max})\}$ is consistent with the second moment of $\sigma(\omega)$ as

$$\omega_{\max} = c\left(\int_{-\infty}^{\infty} \frac{d\omega}{\pi}\omega^2 \text{Re}\{\sigma(\omega)\}\right)^{\frac{1}{3}}. \tag{8}$$

Here, the constant c simply ensures that we capture the whole spectrum (we used $c = 4$). The second moment is related the tail of $\Pi(i\omega_n)$ by

$$\lim_{i\omega_n \to \infty} \omega_n^2 \Pi(i\omega_n) = \int_{-\infty}^{\infty} \frac{d\omega}{\pi}\omega^2 \text{Re}\{\sigma(\omega)\}. \tag{9}$$

Equations (6) and (9) provide the relations between $\omega_{\max}$, β, $\boldsymbol{\sigma}$, and $\boldsymbol{\Pi}$. Supposing the vectors provide faithful discretization to these integrals, they can be written as

$$\Pi_0 = \frac{\omega_{\max}}{M\pi} 2\sum_m \sigma_m, \tag{10}$$

and

$$\left(\frac{2\pi N}{\beta}\right)^2 \Pi_N = \frac{\omega_{\max}^3}{M^3\pi} 2\sum_m m^2\sigma_m. \tag{11}$$

These should clarify equation (8) and how a user with fixed β and $\boldsymbol{\Pi}$ can reinterpret $\boldsymbol{\sigma}$ and deduce $\omega_{\max}$ from the output of the neural network. Finally, the rescaling (8) also has the advantage of making every component of $\boldsymbol{\sigma}$ useful, because the sampling of narrow functions ends exactly where necessary rather than sampling long tails with vanishingly small values.

5 Experiment & Conclusions

To verify that the above rescaling procedure helps neural networks to generalize to out-of-training temperatures, we trained over a thousand fully connected neural networks. Half of them were trained with rescaled targets $\boldsymbol{\sigma}$ using the $\omega_{\max}$ of equation (8) and the other half with the default $\omega_{\max}$. Each neural network was trained on one of the three datasets of section 4, and one of five inverse temperature settings $\beta = 10, 20, 30$, or β randomly selected during training from $\beta \in \{15, 20, 25\}$ or $\beta \in \{10, 15, 20, 25, 30\}$ (respectively producing 300 000 possible $\boldsymbol{\Pi}$ and 500 000 $\boldsymbol{\Pi}$ for the same 100 000 $\boldsymbol{\sigma}$). We evaluated the performance of all these neural networks on all combinations of validation sets and β settings. The best validation loss obtained for given validation β setting as a function of training β setting are reported in Fig. 2.

The best models were all found for 2 to 5 layers deep, combining wide layers of 2000 to 3000 units with narrower layers of 500 to 1500 units in arbitrary order. One of these neural networks is illustrated in Fig. 1(c). Reducing the learning rate when reaching plateaus of the training loss, using a learning rate warmup, and using Batchnorm were all found to improve performance significantly, whereas Dropout and weight decay were found detrimental. Using Softmax or linear output units, as well as any batch size between 100 and 1000 all produced similar results. We used the Adam optimizer for mean square error (MSE) and mean absolute error (L1) loss functions.

Fig. 2 shows that using the rescaling procedure improves the performance on all validation sets, demonstrating better generalization to out-of-training temperatures. Note that poor performance on lower values of β is expected because it corresponds to larger spacing of ω_n on the imaginary axis

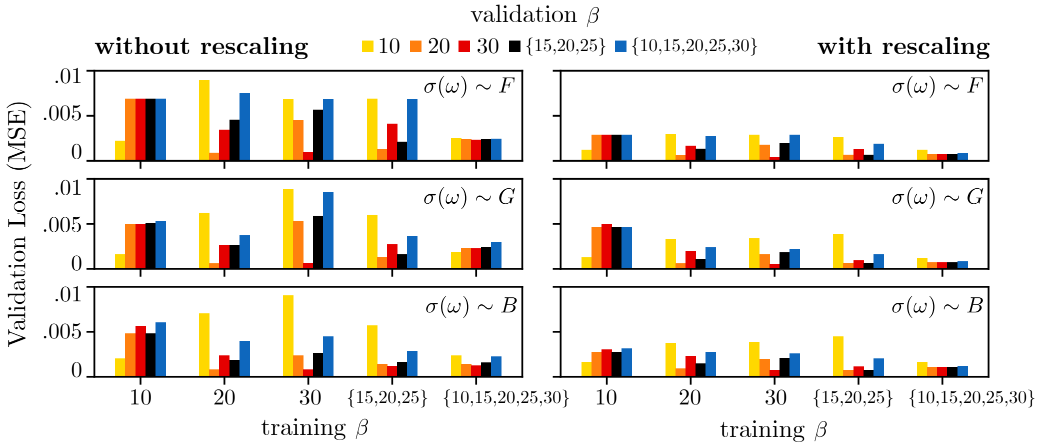

Figure 2: Best validation loss (mean square error, MSE) for five inverse temperatures setting ($\beta = 10$ in yellow, 20 in orange, 30 in red, randomly selected from $\{15, 20, 25\}$ in black or randomly selected from $\{10, 15, 20, 25, 30\}$ in blue) as a function of the training temperature setting (in abscissa). The lower and flatter groups of values correspond to better generalizations to out-of-training temperatures, which shows that generalization is better with the rescaling procedure of section 4 (right) than without it (left). This holds for all three datasets F, G and B of section 2 (respectively top, middle, bottom). These results are for noise with variance 10^{-3}, similar trends were obtained with 10^{-2} and 10^{-5}.

input, thus making it harder to extrapolate to the real axis. Figs. 1(d) and (e) illustrate the rescaling procedure and an example of better performance compared with Maximum entropy [1, 6].

In conclusion, this work shows that it is possible to train neural networks for analytic continuation of response functions for multiple temperatures. We identified the relevant invariances and a rescaling procedure to prepare training data and demonstrated the improved capacity for generalization that this provides. It will be interesting to see if these invariances can be used with more sophisticated neural network architectures.

Broader Impact

Imaginary time Quantum Monte Carlo methods are at the forefront of progress in our understanding of strongly correlated materials such as Mott insulators, magnetocaloric materials, spin glasses, and high-temperature superconductors. These methods are crucial for the development of materials in the field of Energy, Electronics and Quantum computing. But since the prediction of physical properties using Quantum Monte Carlo methods must proceed through analytic continuation, the software infrastructure for computational predictions of materials must include robust methods for analytic continuation. By taking advantage of the power of neural networks for analytic continuation, the present work provides an important step forward in the development of this software infrastructure.

Acknowledgments and Disclosure of Funding

We acknowledge useful discussions with Y. Bengio and M. Charlebois. This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund, the Natural Sciences and Engineering Research Council (Canada) under Grant Nos. RGPIN-2014-04584, and RGPIN-2019-05312, and the Fonds Nature et Technologie (Québec). Computing resources were provided by Compute Canada and Calcul Québec.

References

[1] Mark Jarrell and J. E. Gubernatis. Bayesian inference and the analytic continuation of imaginary-time quantum Monte Carlo data. *Phys. Rep.*, 269(3):133–195, 1996.

[2] Louis François Arsenault, Richard Neuberg, Lauren A. Hannah, and Andrew J. Millis. Projected regression method for solving Fredholm integral equations arising in the analytic continuation problem of quantum physics. *Inverse Probl.*, 33(11):1–9, 2017.

[3] Hongkee Yoon, Jae Hoon Sim, and Myung Joon Han. Analytic continuation via domain knowledge free machine learning. *Phys. Rev. B*, 98(24):1–7, 2018.

[4] Xuping Xie, Feng Bao, Thomas Maier, and Clayton Webster. Analytic Continuation of Noisy Data Using Adams Bashforth ResNet. (1):1–8, 2019.

[5] Romain Fournier, Lei Wang, Oleg V. Yazyev, and Quan Sheng Wu. Artificial Neural Network Approach to the Analytic Continuation Problem. *Phys. Rev. Lett.*, 124(5):1–5, 2020.

[6] Dominic Bergeron and A.-M. S. Tremblay. Algorithms for optimized maximum entropy and diagnostic tools for analytic continuation. *Physical Review E*, 94(2):023303, August 2016.

[7] Gerald D. Mahan. *Many-Particle Physics*. Plenum Press, New York, 1981.