Machine learning with quantum field theories

Dimitrios Bachtis

Joint work with Profs. Gert Aarts and Biagio Lucini.

Can we view machine learning as part of quantum field theory?

And why?

Probability distribution

A probability distribution is a product of strictly positive and appropriately normalized factors (or potential functions) ψ:

$$p(\phi) = \frac{\prod_{c \in C} \psi_c(\phi)}{\int_{\boldsymbol{\phi}} \prod_{c \in C} \psi_c(\boldsymbol{\phi}) d\boldsymbol{\phi}},$$

Probability distribution

A probability distribution is a product of strictly positive and appropriately normalized factors (or potential functions) ψ:

$$p(\phi) = \frac{\prod_{c \in C} \psi_c(\phi)}{\int_{\boldsymbol{\phi}} \prod_{c \in C} \psi_c(\boldsymbol{\phi}) d\boldsymbol{\phi}},$$

1. Factors are the fundamental building blocks of probability distributions.

Probability distribution

A probability distribution is a product of strictly positive and appropriately normalized factors (or potential functions) ψ:

$$p(\phi) = \frac{\prod_{c \in C} \psi_c(\phi)}{\int_{\boldsymbol{\phi}} \prod_{c \in C} \psi_c(\boldsymbol{\phi}) d\boldsymbol{\phi}},$$

1. Factors are the fundamental building blocks of probability distributions.

2. By controlling the factors we are able to control the probability distribution.

Representation

We require some form of representation to construct the probability distribution. We are going to use a finite set Λ that we express as a graph $G(\Lambda,e)$ where e is the set of edges in G.

A clique c is a subset of Λ where the points are pairwise connected. A maximal clique is a clique where we cannot add another point that is pairwise connected with ***all*** the points in the subset.

Representation

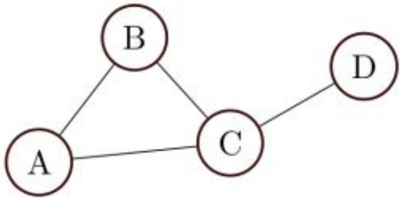
We require some form of representation to construct the probability distribution. We are going to use a finite set Λ that we express as a graph $G(\Lambda,e)$ where e is the set of edges in G.

A clique c is a subset of Λ where the points are pairwise connected. A maximal clique is a clique where we cannot add another point that is pairwise connected with ***all*** the points in the subset.

There are only two maximal cliques, the subsets {A,B,C} and {C,D}.

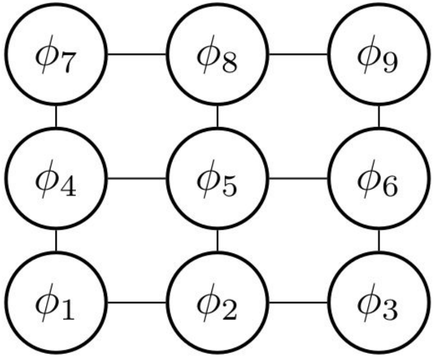
{A,B} is a clique but it is **not** maximal because another point (C) can be included that is pairwise connected with both {A,B}.

{A,B,C,D} is **not** a clique and **not** maximal because (D) is not pairwise connected with all other points (and vice versa).



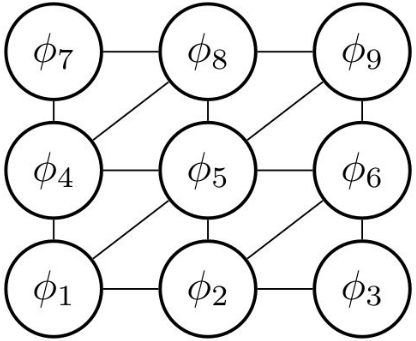
Representation

On the square lattice a maximal clique is an edge.



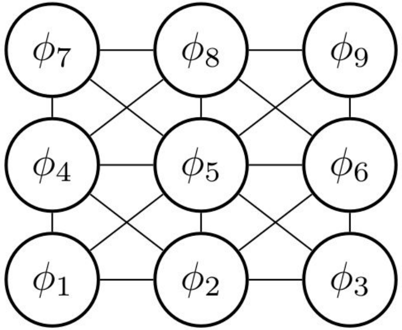
(a)

On a triangular lattice a maximal clique is a triangle.

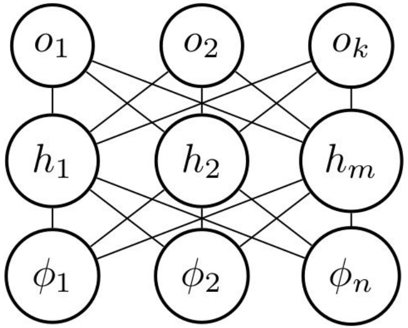


(b)

On the square lattice with both diagonals a maximal clique is a square.



(c)



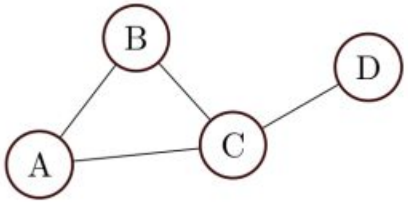
(d)

On the bipartite graph, which represents standard neural network architectures a maximal clique is an edge.

Representation

Given a graph $G(\Lambda,e)$, the random variables φ_i at each point i define a Markov random field if they fulfill the local Markov property with respect to G.

The local Markov property denotes that a random variable φ_i depends only on its neighbors and it is conditionally independent of all other random variables in the set:



$$p(\phi_i|(\phi_j)_{j\in\Lambda-\phi_i}) = p(\phi_i|(\phi_j)_{j\in n_i}).$$

Representation

Hammersley-Clifford theorem

A strictly positive distribution p satisfies the local Markov property of an undirected graph G, if and only if p can be represented as a product of strictly positive potential functions ψ_c over G, one per maximal clique c, i.e.

$$p(\phi) = \frac{1}{Z} \prod_{c \in C} \psi_c(\phi), \quad Z = \int_{\boldsymbol{\phi}} \prod_{c \in C} \psi_c(\boldsymbol{\phi}) d\boldsymbol{\phi}$$

where Z is the partition function and $\boldsymbol{\varphi}$ are all possible states of the system.

Representation

There are two different directions to pursue:

1. We can devise potential functions that satisfy the Hammersley-Clifford theorem to construct a Markov random field.

Representation

There are two different directions to pursue:

1. We can devise potential functions that satisfy the Hammersley-Clifford theorem to construct a Markov random field.
2. We can evaluate if known physical systems can be recast within this mathematical framework by verifying instead if they satisfy the theorem.

We will pursue the second direction.

Representation

2d φ^4 theory:

$$\mathcal{L}_E = \frac{\kappa}{2}(\nabla\phi)^2 + \frac{\mu_0^2}{2}\phi^2 + \frac{\lambda}{4}\phi^4,$$

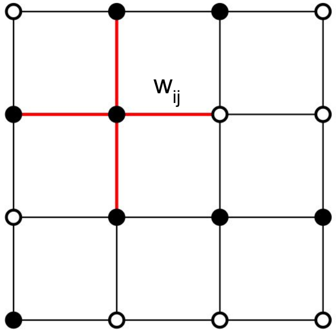
$$S_E = -\kappa_L \sum_{\langle ij \rangle} \phi_i \phi_j + \frac{(\mu_L^2 + 4\kappa_L)}{2} \sum_i \phi_i^2 + \frac{\lambda_L}{4} \sum_i \phi_i^4.$$

k_L, μ_L, λ_L dimensionless parameters

$$w = \kappa_L, \; a = (\mu_L^2 + 4\kappa_L)/2, \; b = \lambda_L/4$$

Inhomogeneous φ^4 theory:

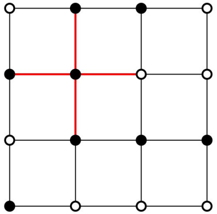
$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$



Representation

The φ^4 lattice field theory is, by definition, formulated on a square lattice which is equivalent to a graph $G(\Lambda, e)$. A non-unique choice of potential function per each maximal clique is:

$$\psi_c = \exp\left[-w_{ij}\phi_i\phi_j + \frac{1}{4}(a_i\phi_i^2 + a_j\phi_j^2 + b_i\phi_i^4 + b_j\phi_j^4)\right],$$



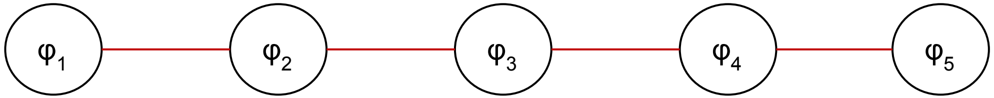
The probability distribution is expressed as a product of strictly positive potential functions ψ, over each maximal clique:

$$p(\phi;\theta) = \frac{\exp\left[\sum_{c\in C}\ln\psi_c(\phi)\right]}{\int_\phi \exp\left[\sum_{c\in C}\ln\psi_c(\phi)\right]d\phi} = \frac{1}{Z}\prod_{c\in C}\psi_c(\phi).$$

The φ^4 theory satisfies Markov properties and it is therefore a Markov random field.

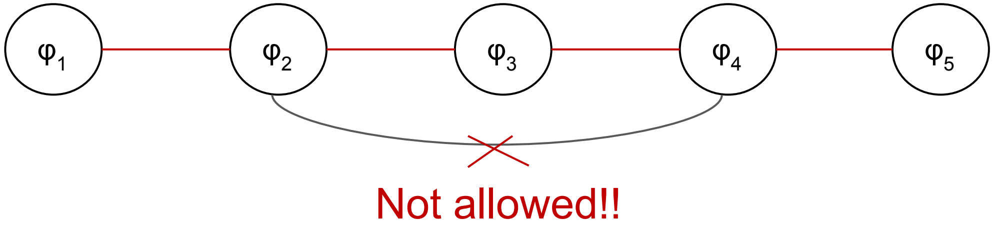
Representation

The Markov property in a Markov chain



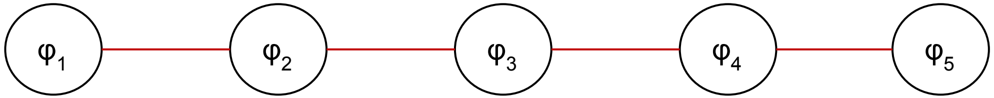
Representation

The Markov property in a Markov chain

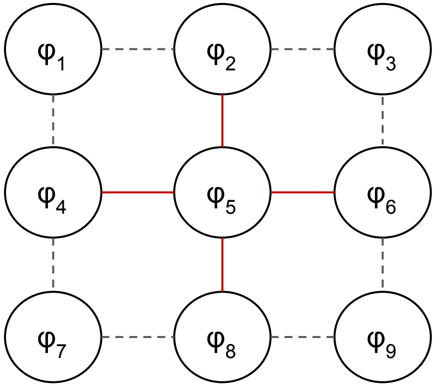


Representation

The Markov property in a Markov chain



A Markov random field satisfies the Markov property in high-dimensions



Learning

Having established that certain physical systems are Markov random fields, how do we use them for machine learning?

Learning

Having established that certain physical systems are Markov random fields, how do we use them for machine learning?

Exactly in the same way as any other machine learning algorithm...

Learning

The φ^4 theory has a probability distribution p(φ;θ) with action S(φ;θ):

$$p(\phi;\theta) = \frac{\exp\big[-S(\phi;\theta)\big]}{\int_{\boldsymbol{\phi}} \exp[-S(\boldsymbol{\phi},\theta)]d\boldsymbol{\phi}}.$$

We now consider a quantum field theory with action A and a target probability distribution q(φ):

$$q(\phi) = \exp[-\mathcal{A}]/Z_{\mathcal{A}}$$

Learning

We can then define an asymmetric distance between the probability distributions p(φ;θ) and q(φ), which is called the Kullback-Leibler divergence:

$$KL(p||q) = \int_{-\infty}^{\infty} p(\phi;\theta) \ln \frac{p(\phi;\theta)}{q(\phi)} d\phi \geq 0.$$

Learning

We can then define an asymmetric distance between the probability distributions p(φ;θ) and q(φ), which is called the Kullback-Leibler divergence:

$$KL(p||q) = \int_{-\infty}^{\infty} p(\phi;\theta) \ln \frac{p(\phi;\theta)}{q(\phi)} d\phi \geq 0.$$

We want to minimize the Kullback-Leibler divergence.

By minimizing it we would make the two probability distributions equal. We can then use the probability distribution p(φ;θ) of the φ^4 theory to draw samples from the target distribution q(φ) of action A.

Learning

We substitute the two probability distributions in the Kullback-Leibler divergence to obtain:

$$F_{\mathcal{A}} \leq \langle \mathcal{A} - S \rangle_{p(\phi;\theta)} + F \equiv \mathcal{F},$$

Bogoliubov Inequality

There are two important observations on the above equation:

1. It sets a rigorous upper bound to the calculation of the free energy of the system with action A.
2. The bound is dependent entirely on samples drawn from the distribution p(φ;θ) of the φ^4 theory.

Learning

To minimize the variational free energy we implement a gradient-based approach:

$$\frac{\partial \mathcal{F}}{\partial \theta_i} = \langle \mathcal{A} \rangle \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle - \left\langle \mathcal{A} \frac{\partial S}{\partial \theta_i} \right\rangle + \left\langle S \frac{\partial S}{\partial \theta_i} \right\rangle - \langle S \rangle \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle,$$

We then update the coupling constants θ at each step t until convergence.

$$\theta^{(t+1)} = \theta^{(t)} - \eta * \mathcal{L}, \qquad \mathcal{L} = \partial \mathcal{F} / \partial \theta^{(t)}$$

After training we expect that, practically:

$$\mathcal{F} \approx F_A \qquad p(\phi; \theta) \approx q(\phi).$$

Learning

A first proof-of-principle demonstration is to use the inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij\rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

to learn a homogeneous action A:

$$\mathcal{A}(\phi) = -\sum_{\langle ij\rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4$$

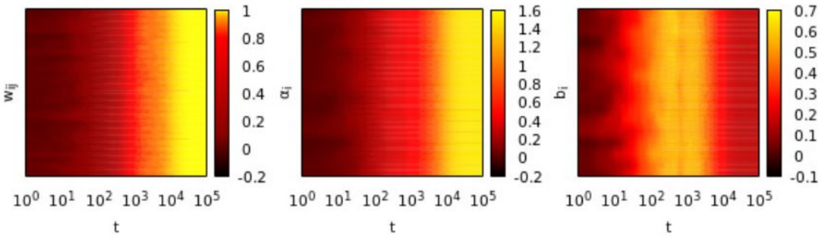


FIG. 2. Variational parameters $\theta = \{w_{ij}, a_i, b_i\}$ versus epochs t on logarithmic scale. The figures depict the evolution of the parameters θ towards the expected values of the coupling constants in the target homogeneous action.

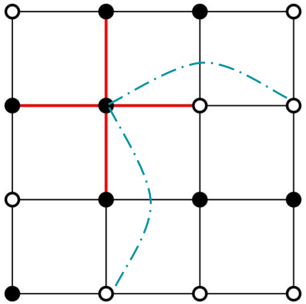
Learning

Another proof-of-principle demonstration is to use the inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

to learn an action that includes longer-range interactions:

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} \phi_i\phi_j$$



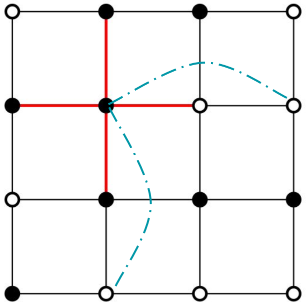
Learning

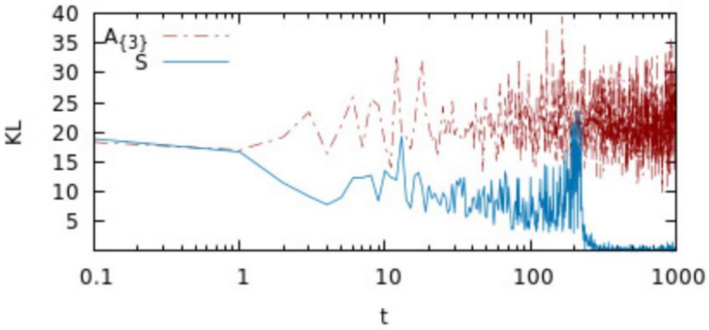
Another proof-of-principle demonstration is to use the inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

to learn an action that includes longer-range interactions:

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij \rangle}\phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} \phi_i\phi_j$$





Learning

Three reweighting (simultaneous) steps: Make the (already trained) inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij\rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Equal to the target action A (acts as a correction step):

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij\rangle}\phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij\rangle_{nnn}}\phi_i\phi_j$$

Learning

Three reweighting (simultaneous) steps: Make the (already trained) inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij\rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Equal to the target action A (acts as a correction step):

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij\rangle}\phi_i\phi_j + 1.52425\sum_i\phi_i^2 + 0.175\sum_i\phi_i^4 - \sum_{\langle ij\rangle_{nnn}}\phi_i\phi_j$$

Extrapolate in the parameter space along the trajectory of a coupling constant g' of A

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij\rangle}\phi_i\phi_j + 1.52425\sum_i\phi_i^2 + 0.175\sum_i\phi_i^4 - \sum_{\langle ij\rangle_{nnn}} g'\,\phi_i\phi_j$$

Learning

Three reweighting (simultaneous) steps: Make the (already trained) inhomogeneous action S:

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Equal to the target action A (acts as a correction step):

$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} \phi_i\phi_j$$

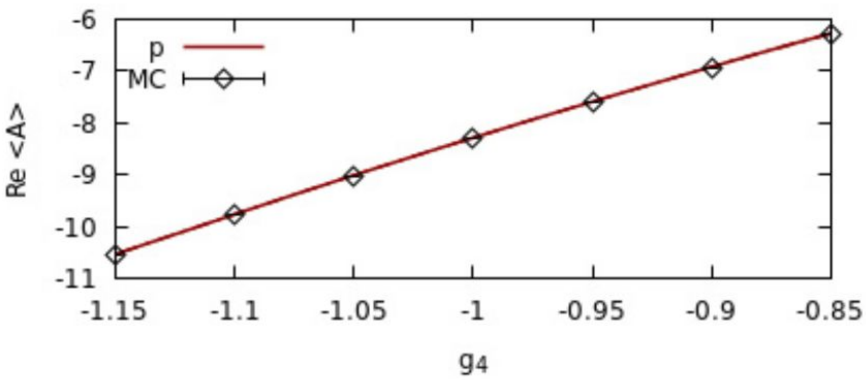
Extrapolate in the parameter space along the trajectory of a coupling constant g' of A

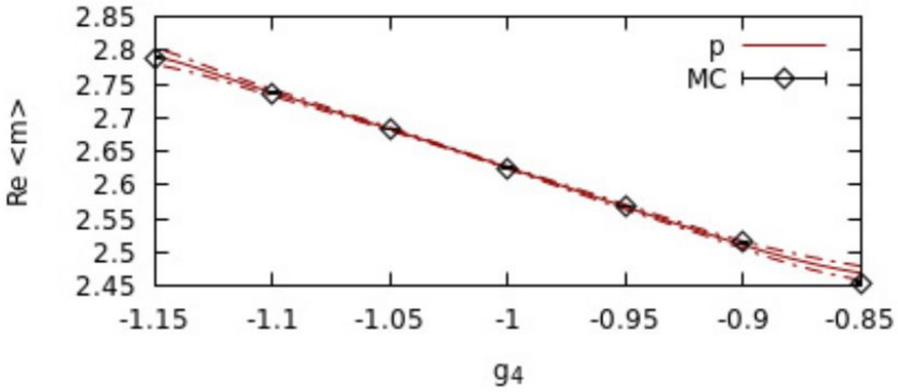
$$\mathcal{A}_{\{4\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} g'\phi_i\phi_j$$

Extrapolate to an imaginary term

$$\mathcal{A}_{\{5\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i\phi_j + 1.52425\sum_i \phi_i^2 + 0.175\sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} g'\phi_i\phi_j + i0.15\sum_i \phi_i^2$$

Learning





The results include reweighting to a complex-valued coupling constant on the mass term and extrapolations in parameter space along the trajectory of the coupling constant g_4 in the longer-range interaction.

$$\mathcal{A}_{\{5\}}(\phi) = -\sum_{\langle ij \rangle} \phi_i \phi_j + 1.52425 \sum_i \phi_i^2 + 0.175 \sum_i \phi_i^4 - \sum_{\langle ij \rangle_{nnn}} g' \phi_i \phi_j + i0.15 \sum_i \phi_i^2$$

Learning

The previous results considered cases where data were not available but the form of the probability distribution $q(\varphi)$, which was considered Boltzmann with action A, was known.

Learning

What if the target probability distribution $q(\varphi)$ is unknown?

Learning

Earlier we defined the Kullback-Leibler divergence as:

$$KL(p||q) = \int_{-\infty}^{\infty} p(\boldsymbol{\phi};\theta) \ln \frac{p(\boldsymbol{\phi};\theta)}{q(\boldsymbol{\phi})} d\boldsymbol{\phi} \geq 0.$$

We will now consider the opposite divergence:

$$KL(q||p) = \int_{-\infty}^{\infty} q(\boldsymbol{\phi}) \ln \frac{q(\boldsymbol{\phi})}{p(\boldsymbol{\phi};\theta)} d\boldsymbol{\phi}.$$

Learning

We can expand the Kullback-Leibler divergence and obtain:

$$KL(q||p) = \langle \ln q(\phi) \rangle_{q(\phi)} - \langle \ln p(\phi;\theta) \rangle_{q(\phi)}.$$

The first right-hand term is constant. Minimizing the Kullback Leibler divergence is equivalent to maximizing the second right-hand term.

We can do this by relying again on a gradient-based approach.

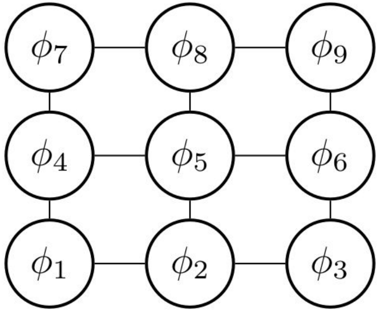
Learning

The derivative of the log-likelihood is:

$$\frac{\partial \ln p(\phi;\theta)}{\partial \theta} = \boxed{\frac{\partial}{\partial \theta}(-S(\phi;\theta))} - \boxed{\left\langle \frac{\partial}{\partial \theta}(-S(\phi;\theta)) \right\rangle_{p(\phi;\theta)}}$$

Calculated on the data

Calculated on samples from the equilibrium distribution



Learning

We are searching for the optimal values of the coupling constants in the φ^4 action that are able to reproduce the data as configurations in the equilibrium distribution.

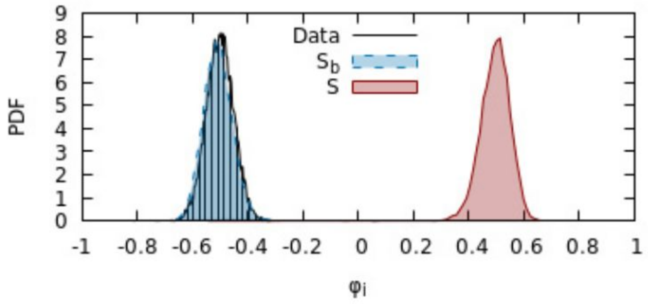
$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Learning

We are searching for the optimal values of the coupling constants in the φ^4 action that are able to reproduce the data as configurations in the equilibrium distribution.

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Case of a Gaussian distribution:

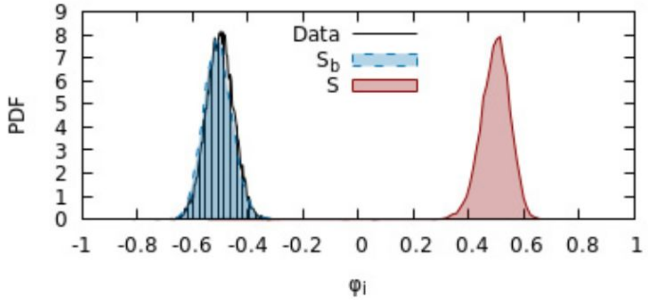


Learning

We are searching for the optimal values of the coupling constants in the φ^4 action that are able to reproduce the data as configurations in the equilibrium distribution.

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4 + \sum_i r_i\phi_i$$

Case of a Gaussian distribution:

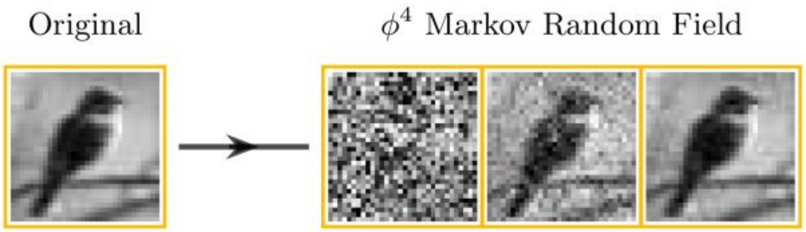


Learning

We are searching for the optimal values of the coupling constants in the φ^4 action that are able to reproduce the data as configurations in the equilibrium distribution.

$$S(\phi;\theta) = -\sum_{\langle ij \rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

Case of an image:



Neural Networks

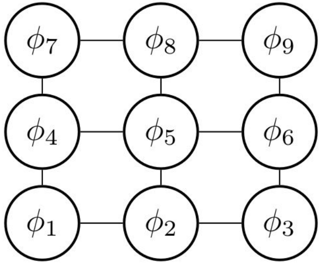
To increase the expressivity of the machine learning algorithm, we need to include a set of latent/hidden variables that extract dependencies on the training data used as input.

This can be achieved with neural networks.

We will derive neural networks from the φ^4 theory.

Neural Networks

φ4 Markov random field

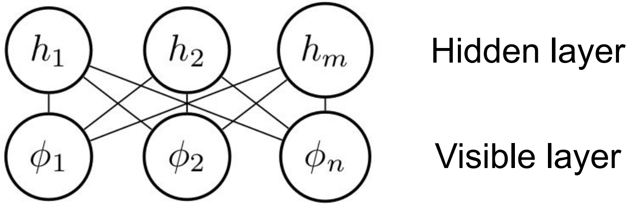


$$S(\phi;\theta) = -\sum_{\langle ij\rangle} w_{ij}\phi_i\phi_j + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4,$$

$$\theta = \{w_{ij}, a_i, b_i\}$$

$$p(\phi;\theta) = \frac{\exp\big[-S(\phi;\theta)\big]}{\int_{\boldsymbol{\phi}} \exp[-S(\boldsymbol{\phi},\theta)]d\boldsymbol{\phi}}.$$

φ4 neural network



$$S(\phi,h;\theta) = -\sum_{i,j} w_{ij}\phi_i h_j + \sum_i r_i\phi_i + \sum_i a_i\phi_i^2$$
$$+\sum_i b_i\phi_i^4 + \sum_j s_j h_j + \sum_j m_j h_j^2 + \sum_j n_j h_j^4,$$

$$\theta = \{w_{ij}, r_i, a_i, b_i, s_j, m_j, n_j\}$$

$$p(\phi,h;\theta) = \frac{\exp[-S(\phi,h;\theta)]}{\int_{\boldsymbol{\phi},\boldsymbol{h}} \exp[-S(\boldsymbol{\phi},\boldsymbol{h};\theta)]d\boldsymbol{\phi}d\boldsymbol{h}}.$$

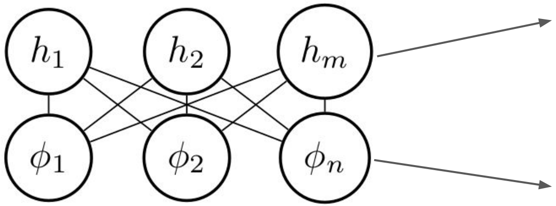
Neural Networks

From the joint probability distribution of the φ^4 neural network

$$p(\phi, h; \theta) = \frac{\exp[-S(\phi, h; \theta)]}{\int_{\boldsymbol{\phi},\boldsymbol{h}} \exp[-S(\boldsymbol{\phi}, \boldsymbol{h}; \theta)] d\boldsymbol{\phi} d\boldsymbol{h}}.$$

We are able to marginalize out variables and derive marginal probability distributions $p(\varphi;\theta)$ and $p(h;\theta)$:

Hidden layer



Visible layer

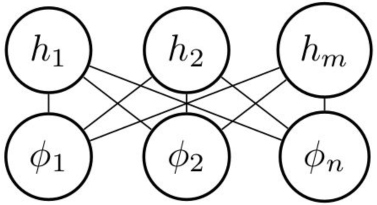
$$p(h; \theta) = \int_{\boldsymbol{\phi}} p(\boldsymbol{\phi}, h; \theta) d\boldsymbol{\phi} = \frac{\int_{\boldsymbol{\phi}} \exp[-S(\boldsymbol{\phi}, h; \theta)] d\boldsymbol{\phi}}{\int_{\boldsymbol{\phi},\boldsymbol{h}} \exp[-S(\boldsymbol{\phi}, \boldsymbol{h}; \theta)] d\boldsymbol{\phi} d\boldsymbol{h}},$$

$$p(\phi; \theta) = \int_{\boldsymbol{h}} p(\phi, \boldsymbol{h}; \theta) d\boldsymbol{h} = \frac{\int_{\boldsymbol{h}} \exp[-S(\phi, \boldsymbol{h}; \theta)] d\boldsymbol{h}}{\int_{\boldsymbol{\phi},\boldsymbol{h}} \exp[-S(\boldsymbol{\phi}, \boldsymbol{h}; \theta)] d\boldsymbol{\phi} d\boldsymbol{h}},$$

Neural Networks

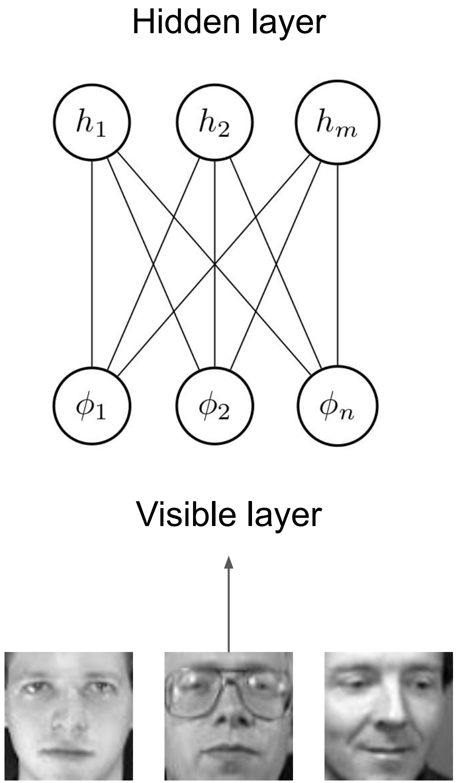
We now want to minimize the asymmetric distance between the empirical probability distribution q(φ) and the marginal probability distribution p(φ;θ):

$$KL(q||p) = \int_{-\infty}^{\infty} q(\phi) \ln \frac{q(\phi)}{p(\phi;\theta)} d\phi.$$



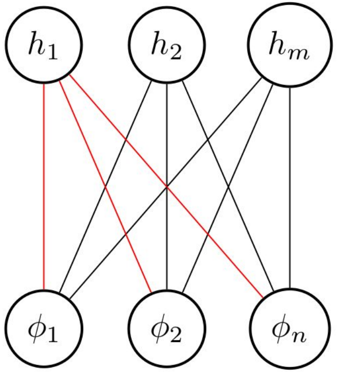
In other words, we want to reproduce the dataset in the visible layer. The hidden layer will then uncover dependencies on the data.

Neural Networks



Neural Networks

Hidden layer



Examples of the coupling constants w_{ij} with j fixed

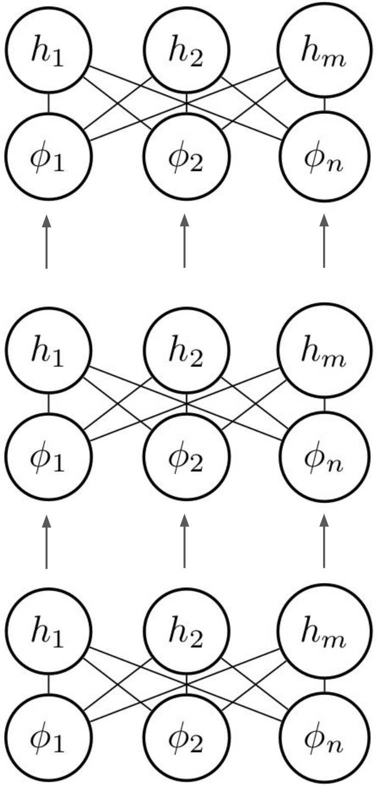
$$S(\phi, h; \theta) = -\sum_{i,j} w_{ij}\phi_i h_j + \sum_i r_i \phi_i + \sum_i a_i \phi_i^2 + \sum_i b_i \phi_i^4 + \sum_j s_j h_j + \sum_j m_j h_j^2 + \sum_j n_j h_j^4,$$

Visible layer

Neural Networks

The hidden layer can serve as input to a new stacked φ^4 neural network to progressively extract features of increased abstraction

Eventually we obtain an architecture that is a universal approximator of a probability distribution.



Neural Networks

The φ^4 neural network:

$$S(\phi, h; \theta) = -\sum_{i,j} w_{ij}\phi_i h_j + \sum_i r_i\phi_i + \sum_i a_i\phi_i^2 + \sum_i b_i\phi_i^4 + \sum_j s_j h_j + \sum_j m_j h_j^2 + \sum_j n_j h_j^4,$$

is a generalization of other neural network architectures:

Gaussian-Gaussian restricted Boltzmann machine:

$b_i = n_j = 0$

Gaussian-Bernoulli restricted Boltzmann machine:

$b_i = n_j = m_j = 0$
h_j binary

Bernoulli-Bernoulli restricted Boltzmann machine:

$b_i = n_j = m_j = a_i = 0$
φ_i, h_j binary

φ^4-Bernoulli restricted Boltzmann machine:

$m_j = n_j = 0$
h_j binary

φ^4 equivalence with the Ising model (under an appropriate limit)

Conclusions

Are there any unique applications?

Inverse Renormalization Group

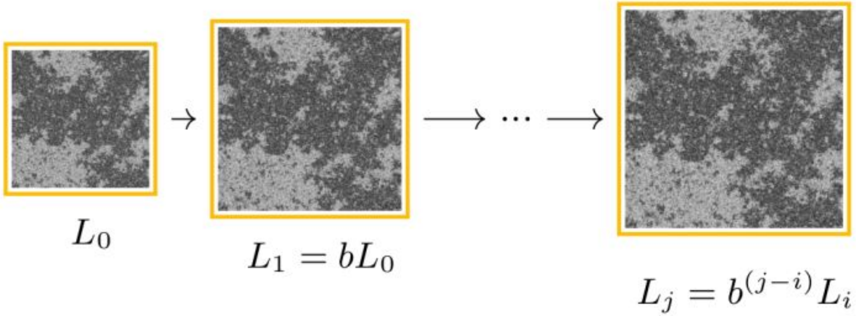


FIG. 1. Illustration of the inverse renormalization group. Inverse transformations are applied to iteratively increase the size L in each dimension by a factor of b, therefore evading the critical slowing down effect.

Inverse renormalization group in quantum field theory, D. Bachtis, G. Aarts, F. Di Renzo, and B. Lucini, (arXiv:2107.00466).

Conclusions

1. What one needs to do machine learning is simply a probability distribution. Lattice field theories therefore emerge as natural machine learning algorithms. We can investigate machine learning as a physical concept within quantum field theory: e.g. what are the phase transitions of quantum field-theoretic machine learning algorithms? How do they behave when they interact with external fields?

Conclusions

1. What one needs to do machine learning is simply a probability distribution. Lattice field theories therefore emerge as natural machine learning algorithms. We can investigate machine learning as a physical concept within quantum field theory: e.g. what are the phase transitions of quantum field-theoretic machine learning algorithms? How do they behave when they interact with external fields?
2. There is an overlap with work in mathematical foundations of quantum field theory. Lattice field theory is also a way to rigorously define quantum fields in a mathematical manner (and now their machine learning counterparts). (Construction of quantum fields from Markoff fields, E. Nelson, J. Funct. Anal. 12, 97 (1973))

Conclusions

1. What one needs to do machine learning is simply a probability distribution. Lattice field theories therefore emerge as natural machine learning algorithms. We can investigate machine learning as a physical concept within quantum field theory: e.g. what are the phase transitions of quantum field-theoretic machine learning algorithms? How do they behave when they interact with external fields?
2. There is an overlap with work in mathematical foundations of quantum field theory. Lattice field theory is also a way to rigorously define quantum fields in a mathematical manner (and now their machine learning counterparts). (Construction of quantum fields from Markoff fields, E. Nelson, J. Funct. Anal. 12, 97 (1973))
3. Lattice field theory is inherently a computational research field. Easy to implement quantum field-theoretic machine learning algorithms and to study their physics computationally. (Renormalization Group in Field Theories with Quantum Quenched Disorder, V. Narovlansky and O. Aharony, Phys. Rev. Lett. 121, 071601 (2018))

Conclusions

1. What one needs to do machine learning is simply a probability distribution. Lattice field theories therefore emerge as natural machine learning algorithms. We can investigate machine learning as a physical concept within quantum field theory: e.g. what are the phase transitions of quantum field-theoretic machine learning algorithms? How do they behave when they interact with external fields?
2. There is an overlap with work in mathematical foundations of quantum field theory. Lattice field theory is also a way to rigorously define quantum fields in a mathematical manner (and now their machine learning counterparts). (Construction of quantum fields from Markoff fields, E. Nelson, J. Funct. Anal. 12, 97 (1973))
3. Lattice field theory is inherently a computational research field. Easy to implement quantum field-theoretic machine learning algorithms and to study their physics computationally. (Renormalization Group in Field Theories with Quantum Quenched Disorder, V. Narovlansky and O. Aharony, Phys. Rev. Lett. 121, 071601 (2018))
4. Experimental implementations of machine learning based on quantum field theory? An interesting read: The Hintons in your Neural Network: a Quantum Field Theory View of Deep Learning, R. Bondesan, M. Welling, arXiv:2103.04913 (2021).

Thank you for your attention!

Quantum field-theoretic machine learning, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. D 103, 074510, (arXiv:2102.09449).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 813942

EuroPLEx
European network for Particle physics, Lattice field theory and Extreme computing

Conclusions

1. What one needs to do machine learning is simply a probability distribution. Lattice field theories therefore emerge as natural machine learning algorithms. We can investigate machine learning as a physical concept within quantum field theory: e.g. what are the phase transitions of quantum field-theoretic machine learning algorithms? How do they behave when they interact with external fields?
2. There is an overlap with work in mathematical foundations of quantum field theory. Lattice field theory is also a way to rigorously define quantum fields in a mathematical manner (and now their machine learning counterparts). (Construction of quantum fields from Markoff fields, E. Nelson, J. Funct. Anal. 12, 97 (1973))
3. Lattice field theory is inherently a computational research field. Easy to implement quantum field-theoretic machine learning algorithms and to study their physics computationally. (Renormalization Group in Field Theories with Quantum Quenched Disorder, V. Narovlansky and O. Aharony, Phys. Rev. Lett. 121, 071601 (2018))
4. Experimental implementations of machine learning based on quantum field theory? An interesting read: The Hintons in your Neural Network: a Quantum Field Theory View of Deep Learning, R. Bondesan, M. Welling, arXiv:2103.04913 (2021).

Thank you for your attention!

Quantum field-theoretic machine learning, D. Bachtis, G. Aarts and B. Lucini, Phys. Rev. D 103, 074510, (arXiv:2102.09449).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 813942

EuroPLEx
European network for Particle physics, Lattice field theory and Extreme computing