Title: Can we trust phase diagrams produced by artificial neural networks?

Speakers: Sebastian Wetzel

Series: Machine Learning Initiative

Date: April 11, 2019 - 11:00 AM

URL: http://pirsa.org/19040090

Abstract: So far artificial neural networks have been applied to discover phase diagrams in many different physical models. However, none of these studies have revealed any fundamentally new physics. A major problem is that these neural networks are mainly considered as black box algorithms. On the journey to detect new physics it is important to interpret what artificial neural networks learn. On the one hand this allows us to judge whether to trust the results, and on the other hand this can give us insight to possible new physics. In this talk I will
discuss applications to different models where we successfully interpreted what was learned by the neural networks.

Can We Trust Phase Diagrams Produced by Artificial Neural Networks?

Sebastian J. Wetzel

Institute for Theoretical Physics, University of Heidelberg

STRUCTURES
CLUSTER OF
EXCELLENCE

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Invitation: Phase transitions from microscopic physics

Hamiltonian

Physical Deduction

$$H(S) = -J \sum_{<ij>_{NN}} s_i s_j$$

(Ising Model)

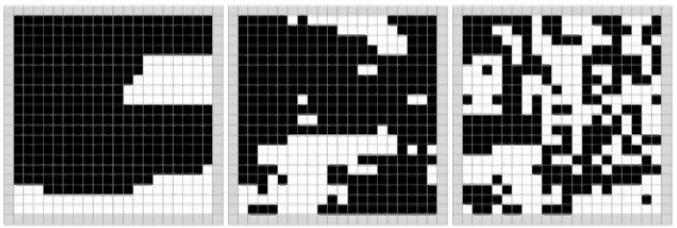
Computer

Order Parameter

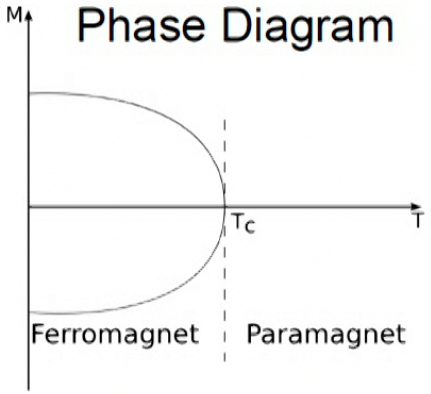
$$M(S) = \frac{1}{N} \sum_i s_i$$

$$\bar{M}(T) = \frac{1}{Z} \sum_{S \in \Lambda} |M(S)| \exp(-H(S)/T)$$

Monte-Carlo-Simulation



Phase Diagram



Invitation: Phase transitions from microscopic physics

Hamiltonian

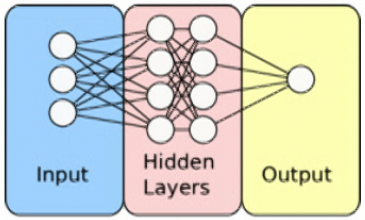
$$H(S) = -J \sum_{<ij>_{NN}} s_i s_j$$

(Ising Model)

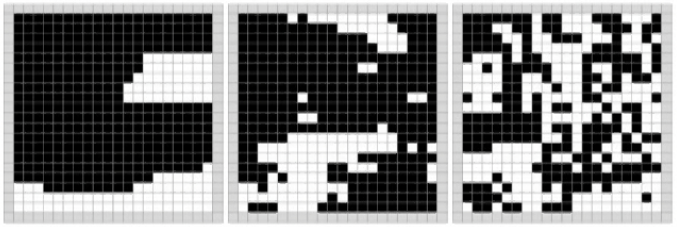
Computer

Computer

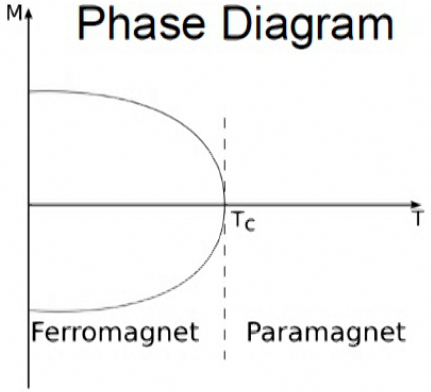
Neural Network ???



Monte-Carlo-Simulation



Phase Diagram

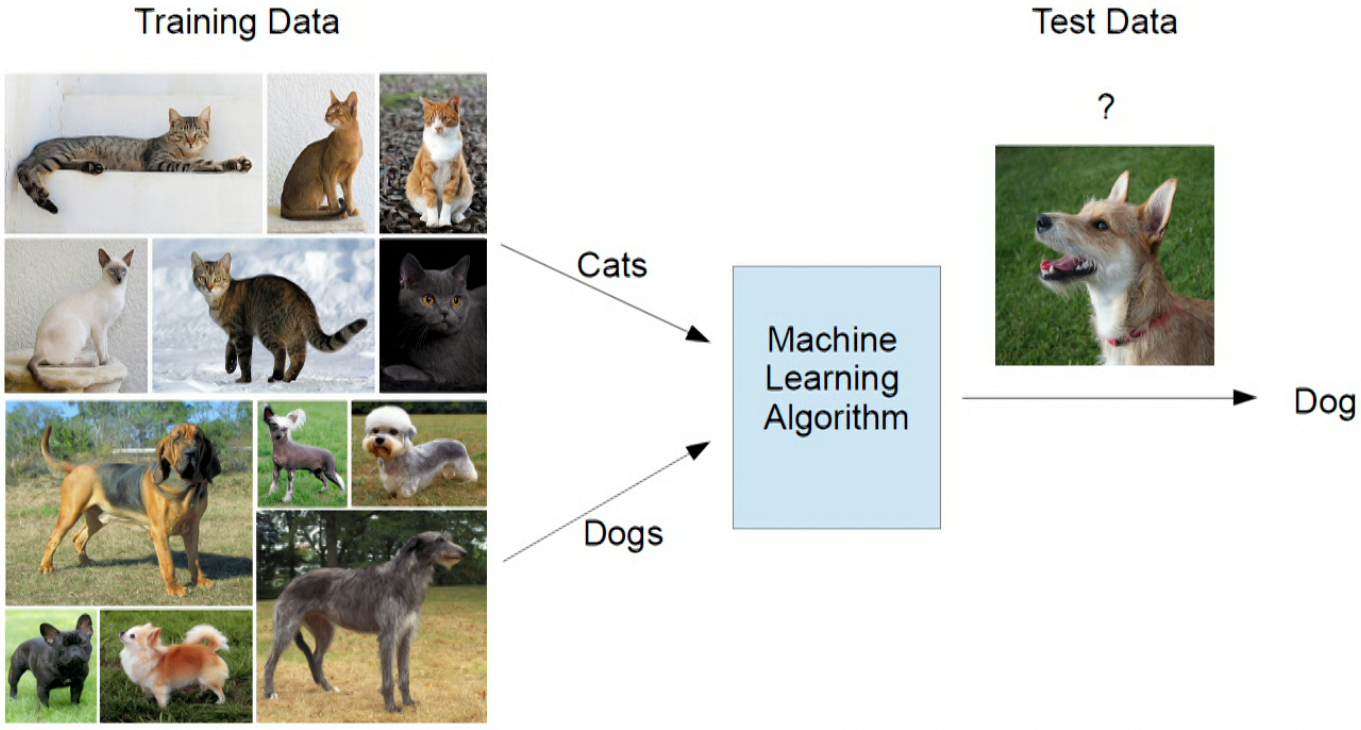


Outline

- Introduction to Machine Learning
- Interpreting Neural Networks
- Examples:
  - Ising Model in 2d
  - SU(2) Lattice Gauge Theory
  - Hubbard Model on the Hexagonal Lattice (preliminary)

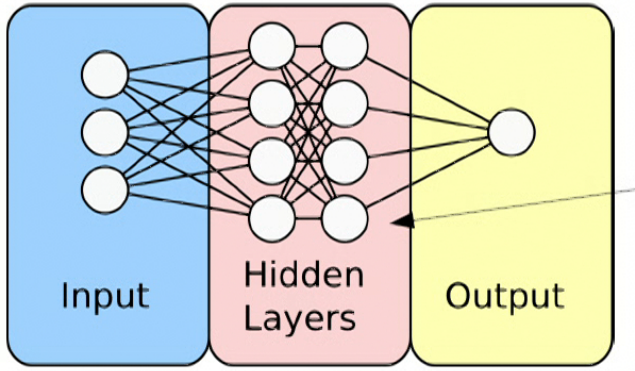
(Supervised) Machine Learning

„Machine learning is the subfield of computer science that gives computers the ability to learn without being explicitly programmed.“ - Wikipedia

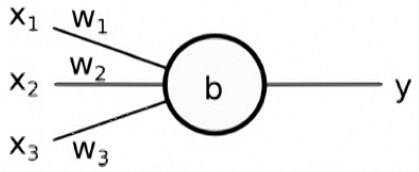


Artificial Neural Networks

Feed forward neural network



Perceptron



$$y = f(\vec{x} \cdot \vec{w} + b)$$

Input: Data $X = (\vec{x}_1, ..., \vec{x}_n)$, Label $Y = (y_1, ..., y_n)$

Output: $Y_{pred} = F(X, w_{ij}^L, b_i^L)$

Goal: choose w_{ij}^L and b_i^L such that $Y_{pred} \approx Y$

Training

Objective functions (loss functions)

- Eg mean squared error (average over all samples)

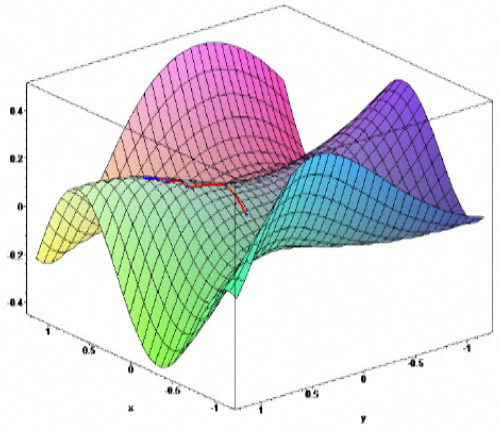
$$MSE = \frac{1}{N} \sum_k \left(\vec{y}_k - F(\vec{x}_k, w_{ij}^L, b_i^L) \right)^2$$

Training

- Determination of w_{ij}^L and b_i^L
- Gradient descent

$$\frac{\partial MSE}{\partial w_{ij}^L} \text{ and } \frac{\partial MSE}{\partial b_i^L}$$

- Backpropagation algorithm



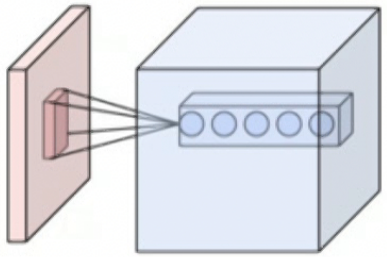
Convolutional Neural Networks

Best performers at image recognition competitions among all machine learning algorithms

- AlexNet (2012), VggNet (2014),
  GoogLeNet(2014), ResNet (2015)

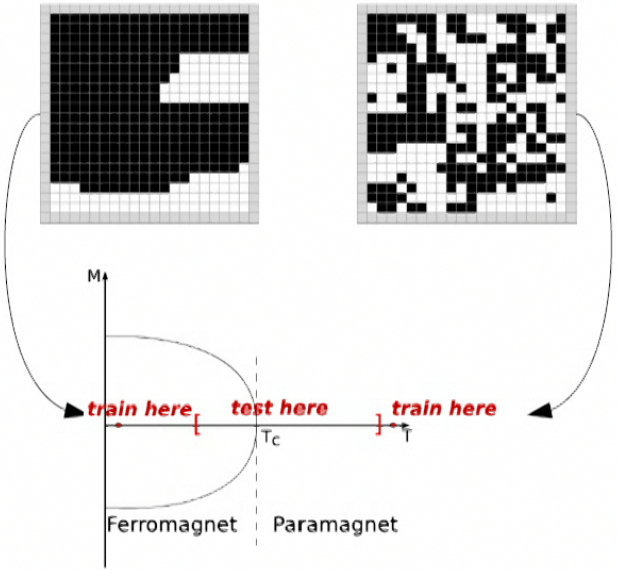
Each neuron of the next layer only sees a small part of the previous layer (Recepive Field)

- Translational Symmetry
- Shared Weights (less parameters)
- Preserves spatial structure

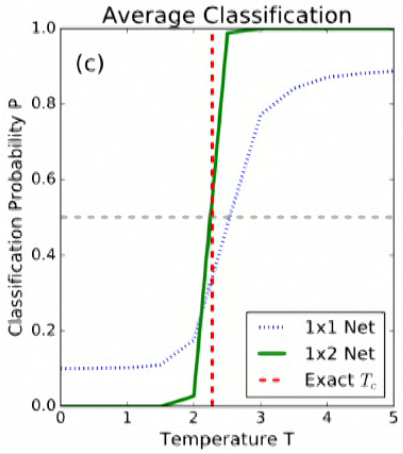


Supervised Learning
2d Ising Model

- Data: Monte Carlo samples
- Training at well known points in phase diagram
- Labels: Phase



- Testing in interval containing phase transition
- Estimate within 1% of exact value $T_c = \frac{2}{\ln(1+\sqrt{2})}$



Carrasquila, Melko, Nature 2017

Machine Learning of Phase Diagrams Overview

		Pro +		Con -	
Supervised	Feed Forward Neural Network	Most powerful	Conv Layer Spatial Structure	Least Interpretable	*Carrasquila, Melko, Nature 2017*
Supervised	Support Vector Machine	Interpretability		Not suitable for large datasets	*Ponte, Melko, Phys Rev B 2017*
Supervised	Recurrent Neural Network	Dynamical Systems			*Nieuwenburg, Bairey, Refael, Phys Rev B 2018*
Unsupervised	Principal Component Analysis	Interpretability	Most easy to use		*Wang, Phys Rev B 2016*
Unsupervised	Autoencoder (Neural Network)		Conv Layer Spatial Structure		*Wetzel, Phys Rev E 2017*
Hybrid	Learning by Confusion				*Nieuwenburg, Liu, Huber, Nature 2017*

Progress Towards New Physics

So far machine learning of phase diagrams did not find any fundamentally new physics.

- My bet is on feed forward neural networks

However, there are serious problems for making progress

Cristoforetti, Jurman, Nardelli, Furlanello, arxiv 2017

Example XY-Model in 2d:

- Unbinding phase transition of topological vortices

Beach, Gloubuleva, Melko, Phys Rev B 2018

- Direct application of Neural Networks yields wrong phase boundary
- Correct results require significant feature engineering

If the system is unknown there is a high chance to make a wrong prediction!

Machine Learning of Phase Diagrams Overview

		Pro +	Con -	Problem	
Supervised	Feed Forward Neural Network	Most powerful	Conv Layer Spatial Structure	Least Interpretable	*Carrasquila, Melko, Nature 2017*
Supervised	Support Vector Machine	Interpretability		Not suitable for large datasets	*Ponte, Melko, Phys Rev B 2017*
Supervised	Recurrent Neural Network	Dynamical Systems			*Nieuwenburg, Bairey, Refael, Phys Rev B 2018*
Unsupervised	Principal Component Analysis	Interpretability	Most easy to use		*Wang, Phys Rev B 2016*
Unsupervised	Autoencoder (Neural Network)		Conv Layer Spatial Structure		*Wetzel, Phys Rev E 2017*
Hybrid	Learning by Confusion				*Nieuwenburg, Liu, Huber, Nature 2017*

Notion of Interpretability

If the neural network bases its decision on <u>one</u> single quantity/obervable $Q(S)$

- The larger the observable, the higher the classification probability.
- If two inputs have the same value of the observable, they have the same classification probability.

The Neural network can be mapped via a bijective function to the observable

$$F(S) = f(Q(S))$$

Notion of Interpretability

If the neural network bases its decision on <u>one</u> single quantity/obervable $Q(S)$

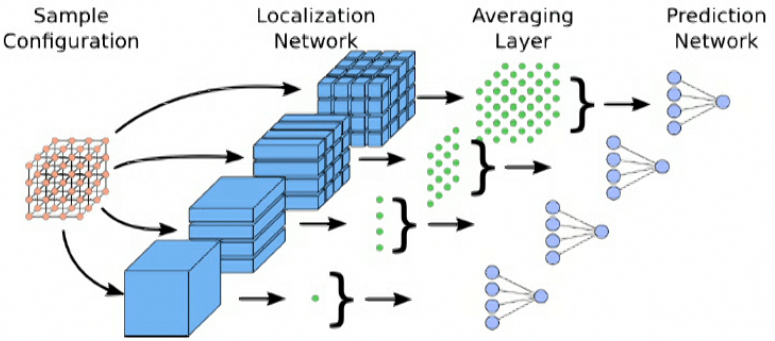
- The larger the observable, the higher the classification probability.
- If two inputs have the same value of the observable, they have the same classification probability.

The Neural network can be mapped via a bijective function to the observable

$$F(S) = f(Q(S))$$

Interpretation of Neural Network

Interpretation Net:



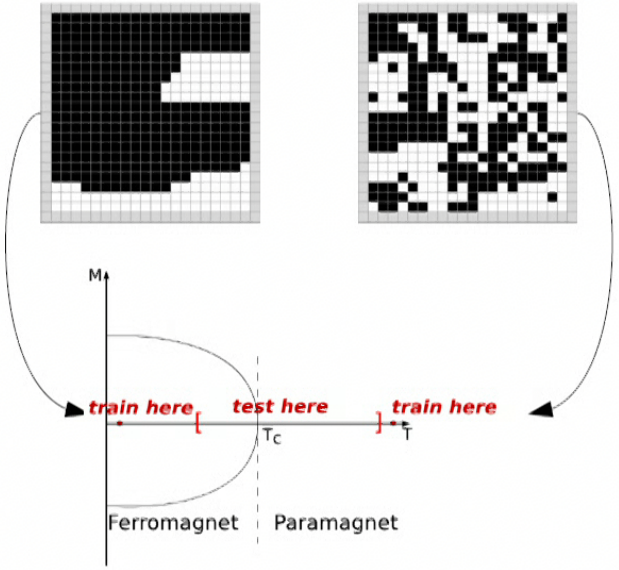
Wetzel, Scherzer, PRB 2017

- Interpretation Net interpolates between a general NN and a minimal optimal NN which has the same performance
- Interpretation by reducing the NN capacity in an ordered manner until one observes a performance drop
- Inspired by extensive physical quantities (averaging layer provides extensiveness)

Interpretation of Neural Network
2d Ising Model

Starting Neural Network:

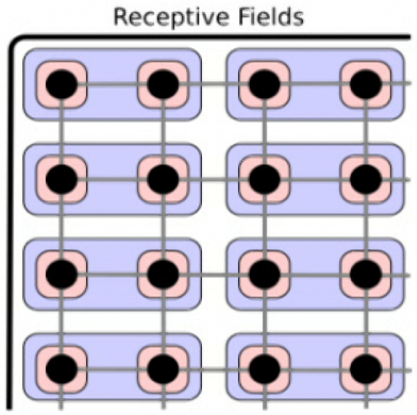
- Conv Net with full receptive field
- Training until converged
- Remember Loss value as measure of performance



Interpretation of Neural Network
2d Ising Model

Reinitialize the neural network with reduce receptive field sizes

- Train again until converged and compare the loss to the previous network
- Observe drop in performance from 1x2 to 1x1 and from 1x1 to baseline
- Dominant contributions must contain functions of spins and neighboring spins



Receptive Field Size	Train Loss	Validation Loss
28×28	$6.1588e-04$	0.0232
$\mathbf{1 \times 2}$	**1.2559e-04**	**1.2105e-07**
$\mathbf{1 \times 1}$	**0.2015**	**0.1886**
baseline	0.6931	0.6931

Interpretation of Neural Network
2d Ising Model

2nd Network: 1x2 receptive field

- Express the full neural network in 1x2 form

$$F(S) = F\left(\frac{1}{N}\sum_{<i,j>_T} f(s_i, s_j)\right)$$

- Taylor expansion contains only one addition to 1x1 case

$$f(s_i, s_j) = f_{0,0} + f_{1,0}\, s_i + f_{0,1}\, s_j + f_{2,0}\, s_i^2 + f_{1,1}\, s_i\, s_j + f_{0,2}\, s_j^2 + \ldots$$

- Regression yields explicit form

$$D(S) \approx \text{sigmoid}\left(w\left(\frac{1}{N}\sum_{<i,j>_T} s_i s_j\right) + b\right)$$

Energy / 2

- Only half the energy since we dont sum over all neighbors

Interpretation of Neural Network
2d Ising Model

Decision functions

$$F(S) = \text{sigmoid}(w\,Q(S) + b)$$

- $Q(S) = |1/N \sum_i s_i|$:
- $Q(S) = \frac{1}{N} \sum_{<i,j>nn} s_i s_j$:

Receptive Field Size	Train Loss	Validation Loss
28×28	$6.1588e-04$	0.0232
1×2	**1.2559e-04**	**1.2105e-07**
1×1	**0.2015**	**0.1886**
baseline	0.6931	0.6931

Magnetization

Kashiwa, Kikuchi, Tomiya, arxiv 2019

Kim, Kim, Phys Rev E 2018

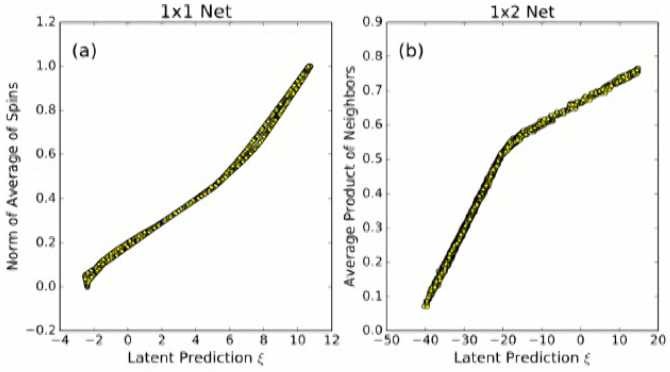
Expected Energy per site

Deduction easily confirmed:

- Perfect correlation

Note:

1x2 Network also has the Magnetization minimum which is easier to find!



SU(2) Lattice Gauge Theory

$$S_{\mathrm{Wilson}}[U] = \beta_{\mathrm{latt}} \sum_x \sum_{\mu<\nu} \mathrm{Re\ tr}\left(1 - U^x_{\mu\nu}\right)$$

Describes smallest loop on the lattice

$$U^x_{\mu\nu} = U^x_\mu U^{x+\hat{\mu}}_\nu U^{x+\hat{\mu}+\hat{\nu}}_{-\mu} U^{x+\hat{\nu}}_{-\nu}$$

$$U^x_\mu \in SU(2)$$

Each Matrix connects two lattice sites

$$U^x_\mu = a^x_\mu 1 + i\left(b^x_\mu \sigma_1 + c^x_\mu \sigma_2 + d^x_\mu \sigma_3\right)$$

- Each Matrix is parametrized by 4 real numbers.

  We performed a MC simulation on a lattice of size 8x8x8x2 as input for the Neural Network

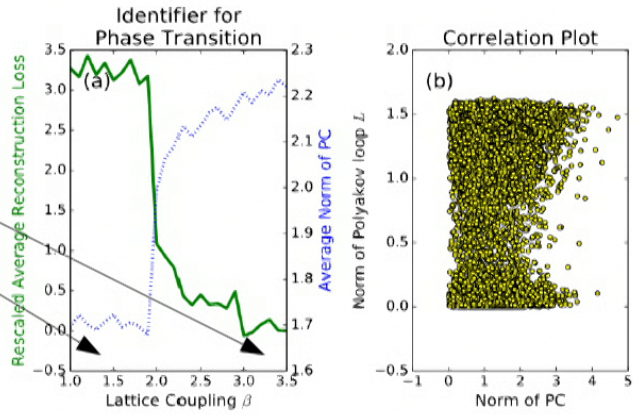
- Toy model for confinement in QCD.
- Polyakov Loop is Order Parameter for in the limit of infinitely heavy quarks.
- Perfect Testing Ground: Polyakov Loop Order Parameter is non-linear and non-local.

Unsupervised Learning (PCA)
SU(2) Lattice Gauge Theory

$$S_{\text{Wilson}}[U] = \beta_{\text{latt}} \sum_{x} \sum_{\mu<\nu} \text{Re tr} \left(1 - U^{x}_{\mu\nu}\right)$$

- Latent parameter does not correspond to order parameter
- PCA + Reconstruction loss can be used to infer different phases

Training at phase indications from unsupervised learning (wait for next slide)



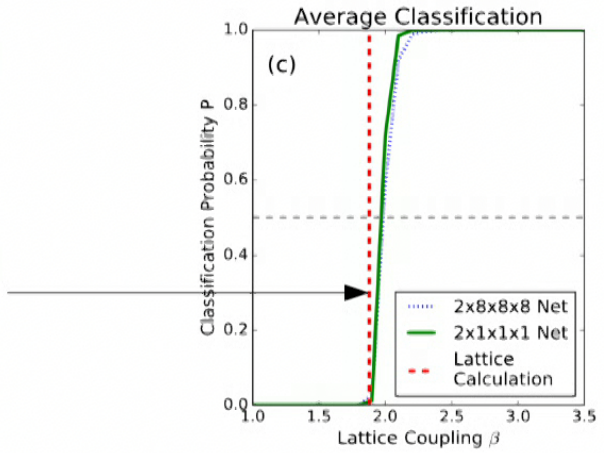
Supervised Learning
SU(2) Lattice Gauge Theory

$$S_{\text{Wilson}}[U] = \beta_{\text{latt}} \sum_{x} \sum_{\mu<\nu} \text{Re tr} \left(1 - U^{x}_{\mu\nu}\right)$$

- Find phase transition close to lattice calculation
- Prediction is inaccurate: Monte Caro Simulations not thermalized

Training at phase indications from unsupervised learning

Testing in interval containing phase transition



Interpretation of Neural Network
SU(2) Gauge Theory (2x8x8x8 Lattice)

General decision function:

$$F(S) = \text{sigmoid}(w\,Q(S) + b)$$

2x1x1x1 Decision function:

Receptive Field Size	Train Loss	Validation Loss
$2 \times 8 \times 8 \times 8$	$1.0004e-04$	$2.6266e-04$
$\mathbf{2 \times 1 \times 1 \times 1}$	**8.8104e-08**	**6.8276e-08**
$\mathbf{2 \times 1 \times 1 \times 1^*}$	**2.2292e-07**	**4.2958e-07**
$1 \times 1 \times 1 \times 1$	0.6620	0.9482
baseline	0.6931	0.6931

$$F(S) \approx \text{sigmoid}\left(w\left(\frac{2}{N}\sum_{\vec{x}} f(\{U_\mu^{x_0,\vec{x}}\})\right) + b\right)$$

Regression yields 561 terms:

$$\begin{aligned} f(\{U_\mu^{x_0}\}) \approx &+ 7.3816\; a_\tau^0 a_\tau^1 + 0.2529\; a_\tau^1 b_\tau^1 + \ \ldots \\ &- 0.2869\; d_\tau^0 c_\tau^1 - 7.2279\; b_\tau^0 b_\tau^1 \\ &- 7.3005\; c_\tau^0 c_\tau^1 - 7.4642\; d_\tau^0 d_\tau^1 \ . \end{aligned}$$

$$f(\{U_\mu^{x_0}\}) = a_\tau^0 a_\tau^1 - b_\tau^0 b_\tau^1 - c_\tau^0 c_\tau^1 - d_\tau^0 d_\tau^1 = \text{tr}\left(U_\tau^0 U_\tau^1\right)$$

- Polyakov Loop

Interpretation of Neural Network
SU(2) Gauge Theory (2x8x8x8 Lattice)

General decision function:

$$F(S) = \text{sigmoid}(w\,Q(S) + b)$$

2x1x1x1 Decision function:

Receptive Field Size	Train Loss	Validation Loss
$2 \times 8 \times 8 \times 8$	$1.0004e-04$	$2.6266e-04$
$\mathbf{2 \times 1 \times 1 \times 1}$	**8.8104e-08**	**6.8276e-08**
$\mathbf{2 \times 1 \times 1 \times 1^*}$	**2.2292e-07**	**4.2958e-07**
$1 \times 1 \times 1 \times 1$	0.6620	0.9482
baseline	0.6931	0.6931

$$F(S) \approx \text{sigmoid}\left(w\left(\frac{2}{N}\sum_{\vec{x}} f(\{U_\mu^{x_0,\vec{x}}\})\right) + b\right)$$

Regression yields 561 terms:

$$\begin{aligned} f(\{U_\mu^{x_0}\}) \approx &+ 7.3816\; a_\tau^0 a_\tau^1 + 0.2529\; a_\tau^1 b_\tau^1 + \ \ldots \\ &- 0.2869\; d_\tau^0 c_\tau^1 - 7.2279\; b_\tau^0 b_\tau^1 \\ &- 7.3005\; c_\tau^0 c_\tau^1 - 7.4642\; d_\tau^0 d_\tau^1 \ . \end{aligned}$$

$$f(\{U_\mu^{x_0}\}) = a_\tau^0 a_\tau^1 - b_\tau^0 b_\tau^1 - c_\tau^0 c_\tau^1 - d_\tau^0 d_\tau^1 = \text{tr}\left(U_\tau^0 U_\tau^1\right)$$

- Polyakov Loop

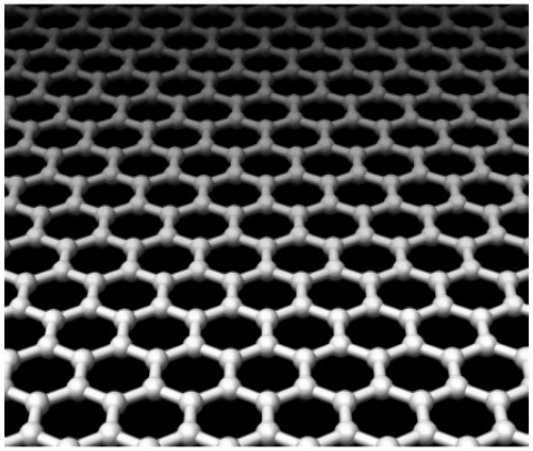
Note: We have constructed the PL without prior knowledge!

Hubbard Model (Hexagonal Lattice)

Model for Graphene

$$H = -t \sum_{<i,j>_{nn},\sigma} (c^{\dagger}_{i,\sigma} c_{j,\sigma} + h.c.) + U \sum_{i} n_{i,\uparrow} n_{i,\downarrow}$$

$$n_{i,\sigma} = c^{\dagger}_{i,\sigma} c_{i,\sigma}$$



- Introduce complex Hubbard-Stratonovich fields

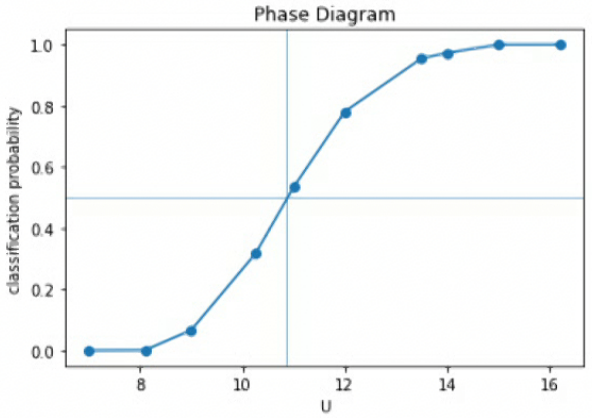
  $$\phi_1, \phi_1^*, \phi_2, \phi_2^*$$

- Produce bosonic field configurations on 12x12x256 lattice
- Examine the transition between Semimetal and Spin Density Wave with NN

Supervised Learning
Hubbard Model (Hexagonal Lattice)

(Very preliminary!)

Employ Convolutional Neural Network on raw bosonic configurations.



Observations:

- Classification curve is relatively flat and thus gives an inaccurate phase boundary.
- A different weight initialization can lead to a shifted phase boundary.

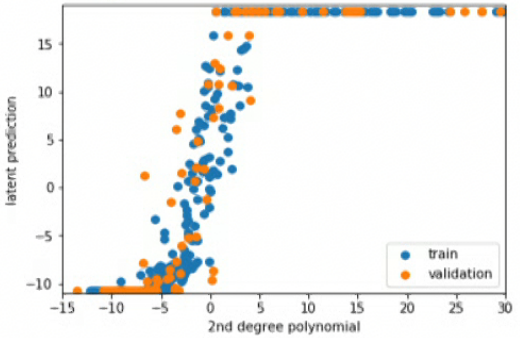
Interpretation of Neural Network
Hubbard Model (Hexagonal Lattice)

(Very preliminary!)

Interpret NN like before.

$$F(S) = \text{sigmoid}(w\,Q(S) + b)$$

Correlation of neural network with regression



- The NN employs a completely local quantity to distinguish between phases.

$$Q(S) \approx 1/N \sum_{\vec{x}} (\phi^2_{1,\vec{x}} + 1.5\phi^{*2}_{1,\vec{x}} + \phi^2_{2,\vec{x}} - 0.5\phi^{*2}_{2,\vec{x}})$$

- All known order parameters are nonlocal in terms of bosonic fields!

Conclusion

Neural Networks are capable of producing phase diagrams for many physical systems.

- In order to trust the results of NNs we need to understand what they learn.
- NNs are no longer a black box algorithm in the context of order parameter based phase transitions.
- Neural Networks learn the same physical quantities that we humans use (Landau/Ehrenfest)
- In some cases we can determine the nature of phases by constructively interpreting what neural networks learn.