Neural Networks as Stat Mech Systems

Based on arXiv:1710.06570 [stat.ML], “A Correspondence Between Random Neural Networks and Statistical Field Theory”

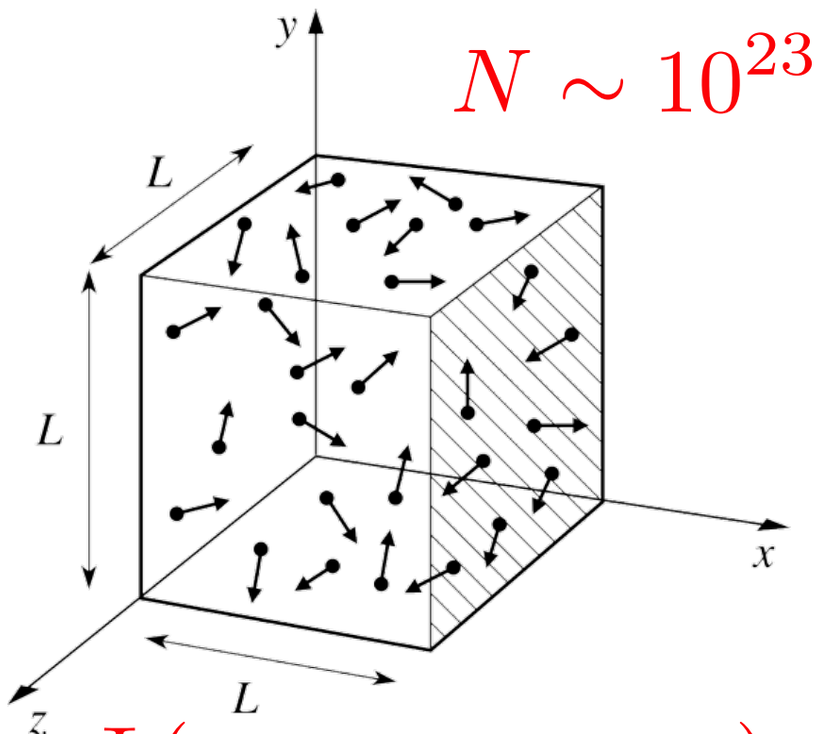
Yoni Kahn, KICP/UIUC

hep-ai 1/8/19

Motivation

Neural networks have extremely large numbers of parameters. On the surface, that makes them hard to understand analytically.

But physics provides examples where systems simplify in the limit of large numbers of constituents: thermodynamics/stat mech.



$N \sim 10^{23}$

$\Psi(x_1, x_2, \ldots, x_N)$

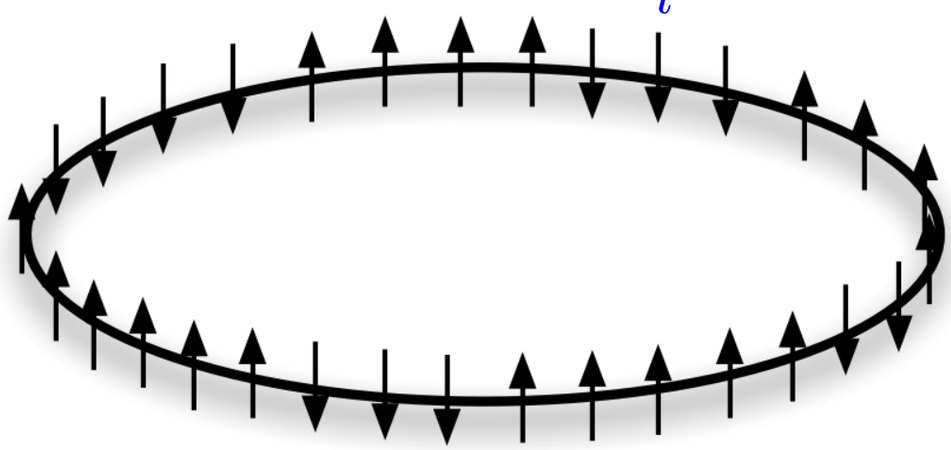
$Z(N, V, T)$

What can we do with a stat mech analogy?

Computing averages/expectation values is “easy.” A stat mech theory gives us fluctuations, mean-field EFT, higher-point correlators, 1/N expansion, etc.

Ising model:

$$H(\{z^l\}) = -\frac{J}{2}\sum_l z^l z^{l+1}$$



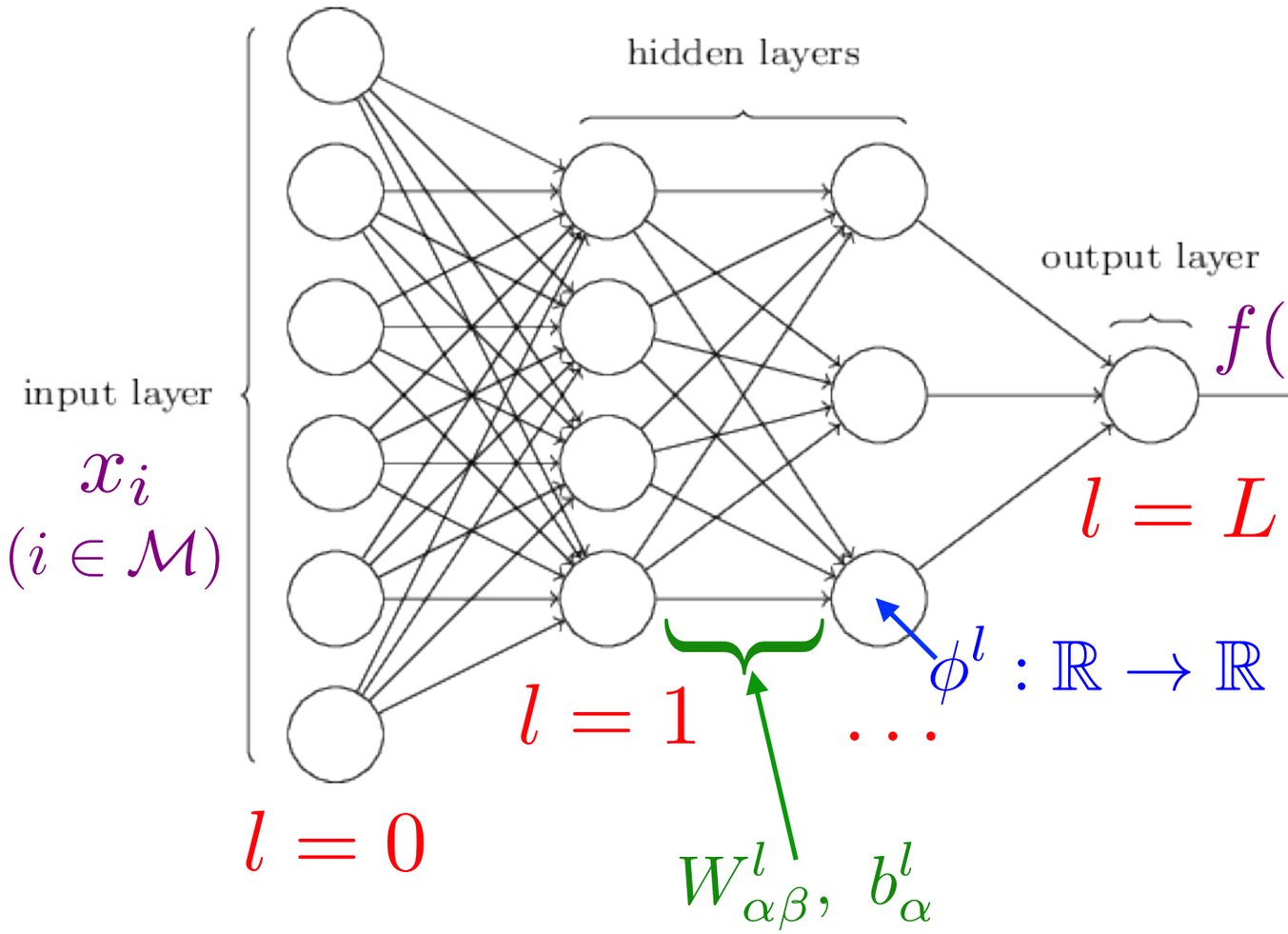
$$z^l = \pm 1$$

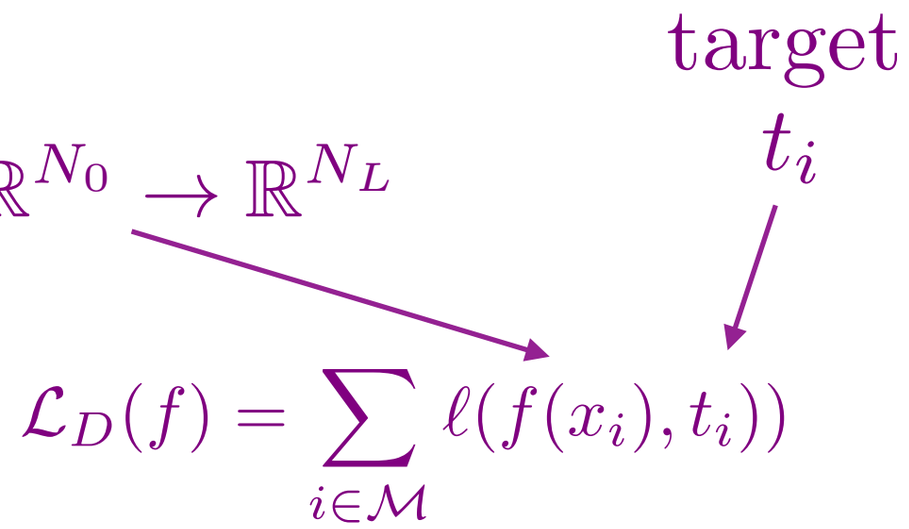
average and take continuum limit

$$\beta H = \frac{1}{2}\int dx\,[m\,u(x)^2 + v\,u(x)^4 + K(\nabla u(x))^2 + \cdots]$$

Feed-forward network: setting notation

$$f(x) = \phi^{L+1}(W^L \phi^L(\cdots \phi^0(W^0 x + b^0)\cdots) + b^L)$$





$$\mathcal{L}(f) = \frac{J_D}{2}\mathcal{L}_D(f) + \sum_{l=0}^{L}\left(\frac{N_l}{2\sigma_w^2}W^l_{\alpha\beta}W^l_{\alpha\beta} + \frac{1}{2\sigma_b^2}b^l_\alpha b^l_\alpha\right)$$

data **not data**

What do we (usually) want from a neural network?

- Network architecture can accommodate “infinite” data without the network growing in size (we’re drawing from a distribution)
- L^2 weight/bias regularization is great, but what you really care about is the loss function on the data
- You want to train your network, not just use random weights and biases
- Loss depends on input data: you actually want the network to learn something

Point of this paper: if you give up on all these, you can have your stat mech model. But what have we learned?

Choosing a random network

Goal of the paper is to study the statistics of an ensemble of random networks. What measure should we choose?

"As the objective we hope to minimize is the total loss, with reference to Jaynes ["Information theory and statistical mechanics", Phys. Rev. 1957] we select the maximum entropy distribution over *f* subject to a measurement of the expected loss function."

What's the maximum entropy distribution holding the average loss (energy) fixed? Why, lo and behold, it's the Boltzmann distribution:

$$P(f) = e^{-\mathcal{L}(f)/Q} \qquad Q = \int [dW]\,[db]\, e^{-\mathcal{L}}$$

(Does anybody know the Jaynes reference? This seems like words draped around the conclusion they wanted from the beginning: if you start with the canonical ensemble, you probably get a stat mech system)

Changing variables

The nontrivial result of the paper is that pre-activations are a nicer choice of variables than weights and biases.

$$(\boldsymbol{z}_\alpha^l)^T = (z_{\alpha;1}^l, z_{\alpha;2}^l, \cdots z_{\alpha;|\mathcal{M}|}^l) \qquad \boldsymbol{\phi}_\alpha^l = \phi^l(\boldsymbol{z}_\alpha^{l-1}) \qquad \boldsymbol{\Sigma}^l = \sigma_w^2 N_l^{-1} \boldsymbol{\phi}_\alpha^l (\boldsymbol{\phi}_\alpha^l)^T + \mathbf{1}\sigma_b^2$$

$$Q = \int [dz][d\lambda] \exp\left[-\frac{J_D}{2} \sum_{i \in \mathcal{M}} \ell(\phi^{L+1}(z_i^L), t_i) - \sum_{l=0}^{L} \left\{ \frac{1}{2} (\boldsymbol{\lambda}_\alpha^l)^T \boldsymbol{\Sigma}^l \boldsymbol{\lambda}_\alpha^l - i(\boldsymbol{\lambda}_\alpha^l)^T \boldsymbol{z}_\alpha^l \right\} \right]. \tag{37}$$

Data loss only depends on the pre-activation of the final layer…

…at the price of coupling all the pre-activations together.

Things which are not addressed in this paper but may be worth investigating:

- What is the temperature? What else can we compute from Q?
- Can we do perturbation theory in J_D?
- Can we treat the input data as an external source field?

Quick overview of the proof

Remember Faddeev-Popov? Introduce pre-activations
as delta-function constraints, exponentiate them to put them in the action,
complete the square to integrate out weights and biases

$$Q = \int [dW][db] \left(\prod_{i \in \mathcal{M}} \exp \left[-\frac{J_D}{2} \sum_i \ell(\phi^L(\cdots \phi^1(W^0 x_i + b^0) \cdots), t_i) \right] \right)$$
$$\times \exp \left[-\sum_{l=0}^{L} \left(\frac{N_l}{2\sigma_w^2} W^l_{\alpha\beta} W^l_{\alpha\beta} + \frac{1}{2\sigma_b^2} b^l_\alpha b^l_\alpha \right) \right] \tag{27}$$

$$= \int [dW][db] \left(\prod_{i \in \mathcal{M}} \int [dz_i^0] \exp \left[-\frac{J_D}{2} \sum_i \ell(\phi^L(\cdots \phi^2(W^1 \phi^1(z_i^0) + b^1) \cdots), t_i) \right] \right.$$
$$\left. \times \delta(z_i^0 - W^0 x_i - b^0) \right) \exp \left[-\sum_{l=0}^{L} \left(\frac{N_l}{2\sigma_w^2} W^l_{\alpha\beta} W^l_{\alpha\beta} + \frac{1}{2\sigma_b^2} b^l_\alpha b^l_\alpha \right) \right] . \tag{28}$$

Iterate to expand the activation function composition in the data loss:

$$Q = \int [dW][db] \left(\prod_{i \in \mathcal{M}} \prod_{l=0}^{L} \int [dz_i^l] \exp \left[-\frac{J_D}{2} \sum_i \ell(\phi^{L+1}(z_i^L), t_i) \right] \right.$$
$$\left. \times \prod_{l=0}^{L} \delta(z_i^l - W^l \phi^l(z_i^{l-1}) + b^l) \right) \exp \left[-\sum_{l=0}^{L} \left(\frac{N_l}{2\sigma_w^2} W^l_{\alpha\beta} W^l_{\alpha\beta} + \frac{1}{2\sigma_b^2} b^l_\alpha b^l_\alpha \right) \right] \tag{29}$$

only final layer pre-activation

all other activations hiding here

Quick overview of the proof

Remember Faddeev-Popov? Introduce pre-activations
as delta-function constraints, exponentiate them to put them in the action,
complete the square to integrate out weights and biases

$$\delta(x) = \int d\lambda \, e^{-ix\lambda}$$

$$\implies \quad Q = \int [d\Omega] \exp\Bigg[- \sum_{i \in \mathcal{M}} \Bigg\{ \frac{J_D}{2} \ell(\phi^{L+1}(z_i^L), t_i) + i\lambda^0_{\alpha;i}(z^0_{\alpha;i} - W^0_{\alpha\beta} x_{\beta;i} - b^0_{\alpha;i}) + \sum_{l=1}^{L} i\lambda^l_{\alpha;i}(z^l_{\alpha;i} - W^l_{\alpha\beta}\phi^l(z^{l-1}_{\beta;i}) - b^l_\alpha) \Bigg\} - \sum_{l=0}^{L} \left(\frac{N_l}{2\sigma_w^2} W^l_{\alpha\beta} W^l_{\alpha\beta} + \frac{1}{2\sigma_b^2} b^l_\alpha b^l_\alpha \right) \Bigg] \tag{25}$$

where we have let $[d\Omega] = [dW][db][dz][d\lambda]$ *for notational convenience.*

Quick overview of the proof

Remember Faddeev-Popov? Introduce pre-activations
as delta-function constraints, exponentiate them to put them in the action,
complete the square to integrate out weights and biases

$$\delta(x) = \int d\lambda \, e^{-ix\lambda}$$

$$\Longrightarrow \quad Q = \int [d\Omega] \exp\left[- \sum_{i\in\mathcal{M}} \left\{ \frac{J_D}{2} \ell(\phi^{L+1}(z_i^L), t_i) + i\lambda^0_{\alpha;i}(z^0_{\alpha;i} - W^0_{\alpha\beta}x_{\beta;i} - b^0_{\alpha;i}) \right.\right.$$
$$\left.\left. + \sum_{l=1}^{L} i\lambda^l_{\alpha;i}(z^l_{\alpha;i} - W^l_{\alpha\beta}\phi^l(z^{l-1}_{\beta;i}) - b^l_\alpha) \right\} - \sum_{l=0}^{L} \left(\frac{N_l}{2\sigma_w^2} W^l_{\alpha\beta}W^l_{\alpha\beta} + \frac{1}{2\sigma_b^2} b^l_\alpha b^l_\alpha \right) \right] \tag{25}$$

where we have let $[d\Omega] = [dW][db][dz][d\lambda]$ *for notational convenience.*

At this stage, all weights and biases appear at most quadratically,
at the expense of adding pre-activations *z* and auxiliary fields λ to the action.
Complete the square, integrate out W and b to get

$$Q = \int [dz][d\lambda] \exp\left[-\frac{J_D}{2} \sum_{i\in\mathcal{M}} \ell(\phi^{L+1}(z_i^L), t_i) - \sum_{l=0}^{L} \left\{ \frac{1}{2} (\boldsymbol{\lambda}^l_\alpha)^T \boldsymbol{\Sigma}^l \boldsymbol{\lambda}^l_\alpha - i(\boldsymbol{\lambda}^l_\alpha)^T \boldsymbol{z}^l_\alpha \right\} \right]. \tag{37}$$

The rest of the paper

- ~~Network architecture can accommodate "infinite" data without the network growing in size (we're drawing from a distribution)~~
  require $N_l \gg |\mathcal{M}|$
- ~~L^2 weight/bias regularization is great, but what you really care about is the loss function on the data~~
  $J_D \ll 1$
- ~~You want to train your network, not just use random weights and biases~~
  no training, only initialization according to Q
- ~~Loss depends on input data: you actually want the network to learn something~~
  put the network on a circle: $J_D = 0,\ |\mathcal{M}| = 0$

Super-deep networks

In order to integrate out the auxiliary fields, the matrix Σ has to be full rank.
But it's a sum of N_l+1 outer products, so its rank is at most N_l+1.
The size of the matrix is the size of the dataset,
so to get rid of auxiliaries, we need to assume

$$N_l + 1 \gg |\mathcal{M}|$$

(this is deeply unrealistic, and not very robust. Is this assumption even necessary? What's wrong with auxiliary fields?)

Under this assumption, we have the main result of the paper,

$$Q = \int [dz] \exp\left[-\frac{J_D}{2} \sum_{i \in \mathcal{M}} \ell(\phi^{L+1}(z_i^L), t_i) - \frac{1}{2} \sum_{l=0}^{L} \left((\boldsymbol{z}_\alpha^l)^T (\boldsymbol{\Sigma}^l)^{-1} \boldsymbol{z}_\alpha^l + \ln |\boldsymbol{\Sigma}^l| \right) \right]. \tag{9}$$

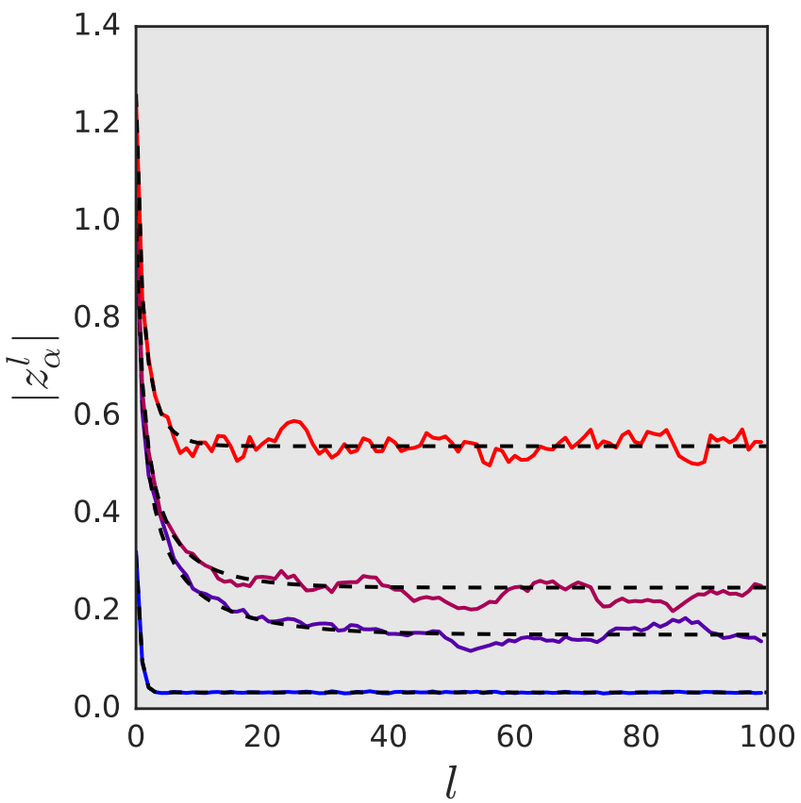
Now what?

$$Q = \int [dz] \exp\left[-\frac{J_D}{2} \sum_{i \in \mathcal{M}} \ell(\phi^{L+1}(z_i^L), t_i) - \frac{1}{2} \sum_{l=0}^{L} \left((\boldsymbol{z}_\alpha^l)^T (\boldsymbol{\Sigma}^l)^{-1} \boldsymbol{z}_\alpha^l + \ln |\boldsymbol{\Sigma}^l| \right) \right]. \tag{9}$$

Poole, Lahiri, Raghu, Sohl-Dickstein, and Ganguli [2016] have a mean-field description of this ensemble from replacing

$$\phi^l(z_\alpha^l)\phi^l(z_\alpha^l) \to \langle \phi^l(z_\alpha^l)\phi^l(z_\alpha^l) \rangle$$

It works, but doesn't get the fluctuations (as expected):



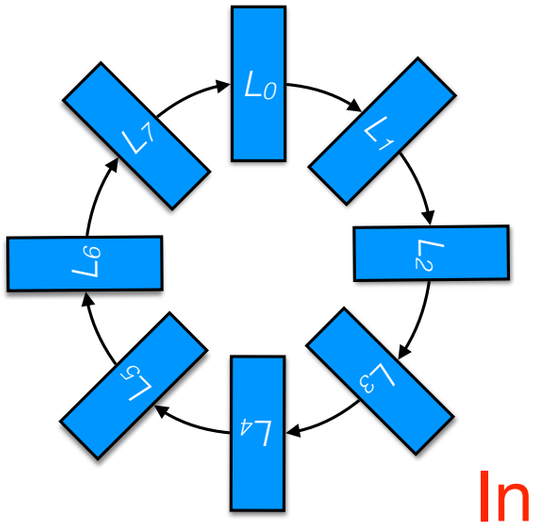
"To overcome these issues, we pursue a more principled solution to Eq. (9) by considering a controlled expansion for large N_l."

From 1 datum to 0

Except that's not what they do. First they simplify to the case of a single input:

$$Q = \int [dz] \exp\left[-\frac{1}{2}\left\{ J_D \ell(\phi^{L+1}(z^L), t) + \frac{z^0_\alpha z^0_\alpha}{\sigma_w^2 N_0^{-1} x_\beta x_\beta + \sigma_b^2} + \sum_{l=1}^{L} \frac{z^l_\alpha z^l_\alpha}{\sigma_w^2 N_l^{-1} \phi^l(z^{l-1}_\beta)\phi^l(z^{l-1}_\beta) + \sigma_b^2} + \sum_{l=1}^{L} N_l \log(\sigma_w^2 N_l^{-1} \phi^l(z^l_\alpha)\phi^l(z^l_\alpha) + \sigma_b^2) \right\}\right]. \tag{10}$$

This is still too hard, so they get rid of the input and put the network on a circle, which necessarily requires $J_D = 0$.



$$\mathcal{L}(\{z^l\}) = \frac{1}{2}\sum_{l=0}^{L}\left\{ \frac{z^l_\alpha z^l_\alpha}{\sigma_w^2 N_l^{-1}\phi(z^{l-1}_\beta)\phi(z^{l-1}_\beta) + \sigma_b^2} + N \log(\sigma_w^2 N^{-1}\phi(z^l_\beta)\phi(z^l_\beta) + \sigma_b^2) \right\} \tag{11}$$

In my opinion, this is a cute interacting field theory model, but by this point has basically nothing to do with neural networks.

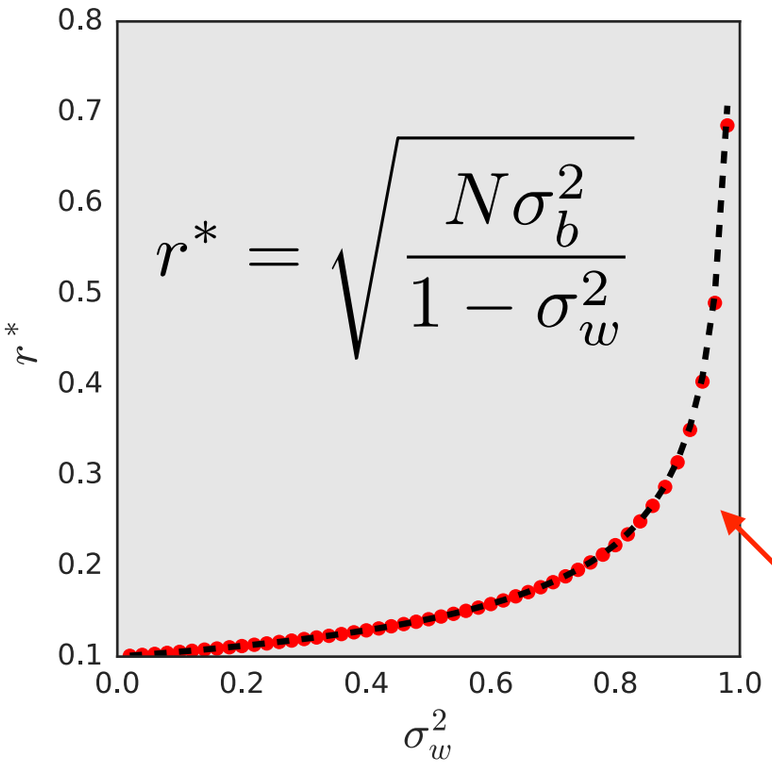
What's left: sanity checks

Try a linear activation function: action is now isotropic in z, so switch to spherical coords,

$$\mathcal{L}(\{r^l\}) = \frac{1}{2}\sum_{l=0}^{L}\left\{\frac{(r^l)^2}{\sigma_w^2 N^{-1}(r^{l-1})^2+\sigma_b^2} - N\log\left(\frac{(r^l)^2}{\sigma_w^2 N^{-1}(r^l)^2+\sigma_b^2}\right)\right\}. \quad (12)$$

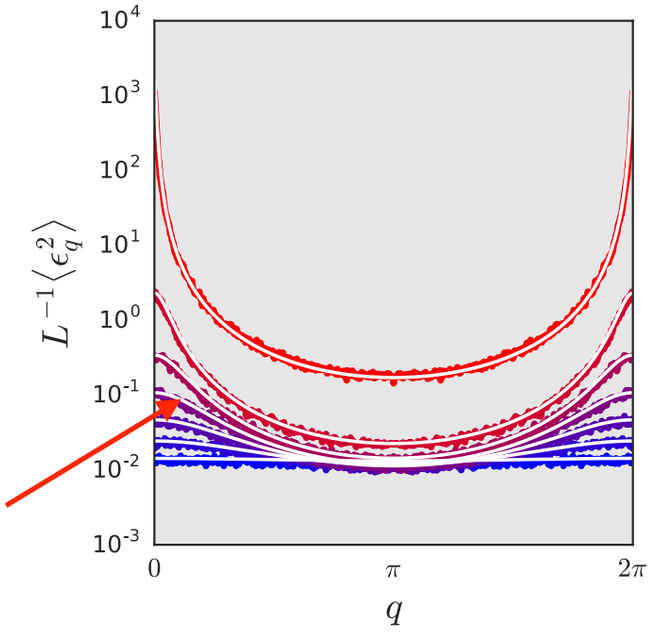
$$(N \equiv N_l,\ (r^l)^2 = z_\alpha^l z_\alpha^l)$$

Find minimum with saddle-point approximation:



Expand around minimum, Fourier transform, find EFT of fluctuations to quadratic order:

$$U(\{\epsilon_q\}) = \frac{L\Delta_w}{\sigma_b^2}\sum_q \left\{(1+\sigma_w^4) - 2\sigma_w^2\cos q\right\}|\epsilon_q|^2$$



Stat mech works!

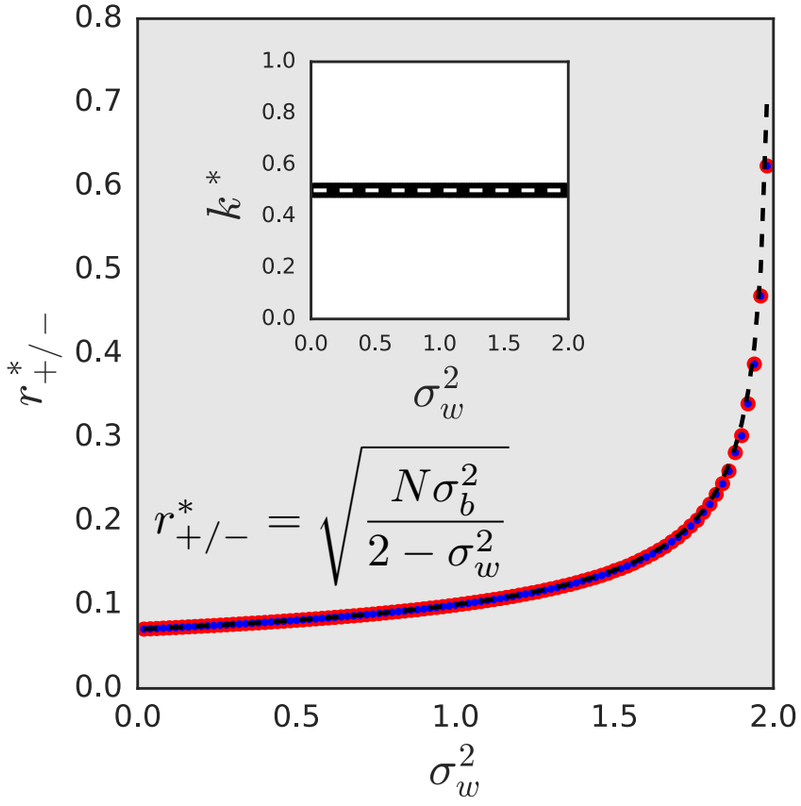
What's left: sanity checks

Try a ReLU activation function: break pre-activations into + and - components

$$\mathcal{L}(\{z^l\}) = \frac{1}{2}\sum_{l=0}^{L}\left\{\frac{(z_+^l)^2+(z_-^l)^2}{\sigma_w^2 N^{-1}(z_+^l)^2+\sigma_b^2} + N\log(\sigma_w^2 N^{-1}(z_+^l)^2+\sigma_b^2)\right\}. \qquad (18)$$

Can switch to spherical coords. with introduction of new field k which counts positive/negative orthants.

Find minimum:
half of pre-activations are zero,
as may be expected



Expand around minimum, find EFT of fluctuations to quadratic order:

$$U = \frac{1}{2}\int dx\Big[(1-\sigma_w^2+\sigma_w^4/2)(\epsilon_+(x))^2+(\epsilon_-(x))^2+3(\epsilon_k(x))^2+\sigma_w^2\epsilon_+(x)\epsilon_-(x)$$
$$+\epsilon_k(x)(\epsilon_+(x)-\epsilon_-(x))+\sigma_w^2\left(\frac{\partial\epsilon_+(x)}{\partial x}\right)^2+\sigma_w^2\frac{\partial\epsilon_+(x)}{\partial x}\epsilon_-(x)\Big]. \qquad (23)$$

This breaks symmetry $\epsilon \longrightarrow -\epsilon$,
as expected because ReLU
distinguishes between forward/backward

What have we learned?

(my heavily subjective conclusions, not the authors')

- If you decide from the outset you want a stat mech model for a neural network, you can get one, as long as you're veeeeery flexible with your definition of "neural network."

- That said, change of variables to pre-activations seems like a useful trick. Do we need to be afraid of auxiliary fields? Probably not (c.f. Faddeev-Popov ghosts, very useful for computing tree-level Feynman diagrams in Yang-Mills).

- Lots of obvious follow-ups: higher-point correlators, perturbation theory in J_D, data as external sources in EFT, 1/N corrections, etc.

- More generally, is there any high-level connection between ensembles of random networks and the behavior of a single network during training? Can we think of training as a flow in the space of random networks? If so, this might be a useful formalism to understand generalization.