AI applications in QFT
Part II

Gert Aarts

Swansea University
Prifysgol Abertawe

MITP, Mainz, July 2025

Stochastic gradient descent, random matrix theory and phase diagrams

Gert Aarts

Swansea University
Prifysgol Abertawe

MITP, Mainz, July 2025

Background and references

this lecture is mostly based on what we understood working with PhD student **Chanju Park** and Biagio Lucini

- GA, B Lucini, **Chanju Park**, PRD 109 (2024) 3, 034521 [2309.15002 [hep-lat]]
- GA, B Lucini, **Chanju Park**, PRE 111 (2025) 1, 015303 [2407.16427 [cond-mat.dis-nn]]
- GA, O Hajizadeh, B Lucini, **Chanju Park**, contribution to *NeurIPS* 2024 workshop *ML and the Physical Sciences*, 2411.13512 [cond-mat.dis-nn]

I also enjoyed

- D Roberts, S Yaida, B Hanin, *The Principles of Deep Learning Theory*, Cambridge University Press [2106.10165 [cs.LG]].

Broader relation between ML and QFT/LFT

- what can theoretical physics do for ML? intriguing connections, exchange of methodology

why explore this?

- theoretical physicists are/should not satisfied with a ‘black-box’ algorithm
- in the future: apply these methods to large scale numerical simulations
- should understand and trust them

Science4AI

- QFT/LFT: extensive experience in analytical and computational studies of systems with many fluctuating degrees of freedom → fairly unique perspective

ML for a theoretical physicist

- neural network = system with many fluctuating degrees of freedom → use statistical mechanics
- learning = optimisation: weight matrices are updated using stochastic gradient descent (SGD)
- SGD = stochastic matrix dynamics → random matrix theory
- Coulomb gas description with effective temperature given by learning rate/batch size
- phase diagram of neural networks resembles those of disordered systems

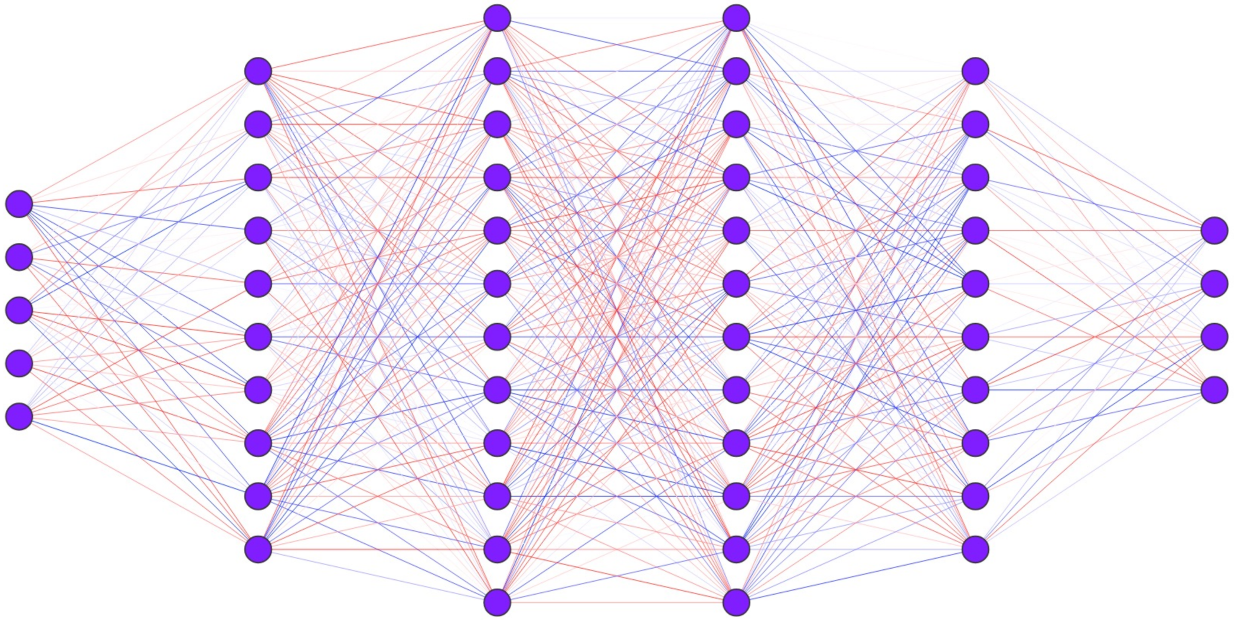
Outline

- basics of feed-forward neural networks (NNs) and stochastic gradient descent (SGD)
- stochastic gradient descent, Dyson Brownian motion and random matrix theory
- examples: restricted Boltzmann machines, transformers
- neural network phase diagram
- outlook

Feed-forward neural network

supervised learning: data set $\mathcal{D} = \{\boldsymbol{x}, \boldsymbol{y}\}$, input $\boldsymbol{x}$ and associated output $\boldsymbol{y}$

input data $\boldsymbol{x}$



output $\hat{\boldsymbol{y}} = f(\boldsymbol{x})$

NN is a “universal approximator”: $\hat{\boldsymbol{y}} \sim \boldsymbol{y}$, should be able to generalise, predict $\boldsymbol{y}$ for unseen data $\boldsymbol{x}$

combination of linear transformations (matrices) and nonlinear ‘activations’ on the nodes

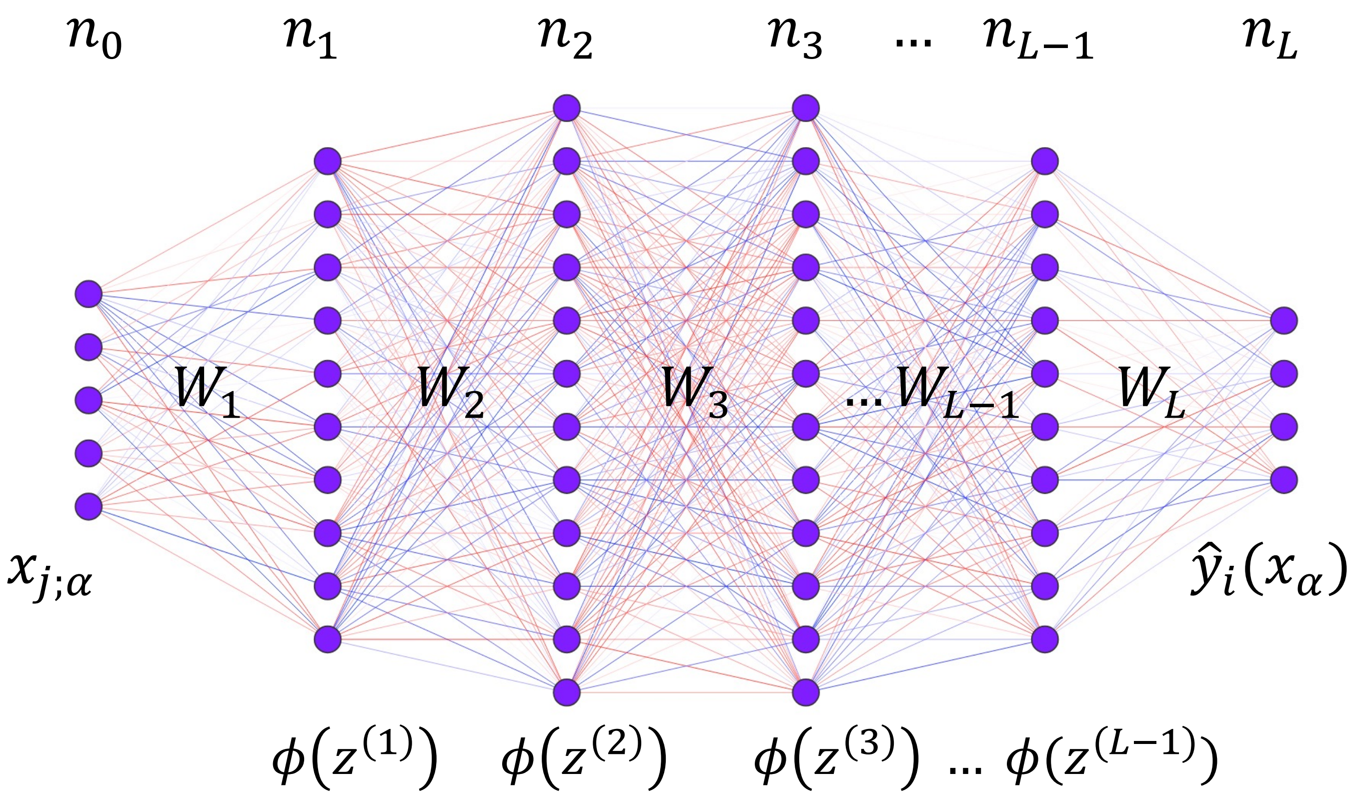
Feed-forward neural network: explicit function

NN function: $$\hat{y}_i(x_\alpha;\theta) \equiv z_i^{(L)}(x_\alpha) = \sum_{j=1}^{n_{L-1}} W_{ij}^{(L)} \phi\left(z_j^{(L-1)}(x_\alpha)\right)$$ $$\theta = \{W^{(1)}, \ldots, W^{(L)}\}$$

pre-activations:

$$z_i^{(l+1)}(x_\alpha) = \sum_{j=1}^{n_l} W_{ij}^{(l+1)} \phi\left(z_j^{(l)}(x_\alpha)\right)$$

$$z_i^{(1)}(x_\alpha) = \sum_{j=1}^{n_0} W_{ij}^{(1)} x_{j;\alpha}$$



Loss function example: mean squared error

$$\mathcal{L}(\theta) \equiv \frac{1}{|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \ell\left(y(x_\alpha), \hat{y}(x_\alpha; \theta)\right)$$

$$\ell(y, \hat{y}) \equiv \frac{1}{2} \sum_{i=1}^{n_L} (y_i - \hat{y}_i)^2$$

- gradient descent $$W'_{ij} = W_{ij} - \alpha \frac{\partial \mathcal{L}(\theta)}{\partial W_{ij}}$$ $$\theta = \{W^{(1)}, \ldots, W^{(L)}\}$$
- learning rate or step size α (usually highly optimised) *
- updates computed over batches of data: batch size $|\mathcal{B}|$

✓ hyperparameters $\alpha, |\mathcal{B}|$

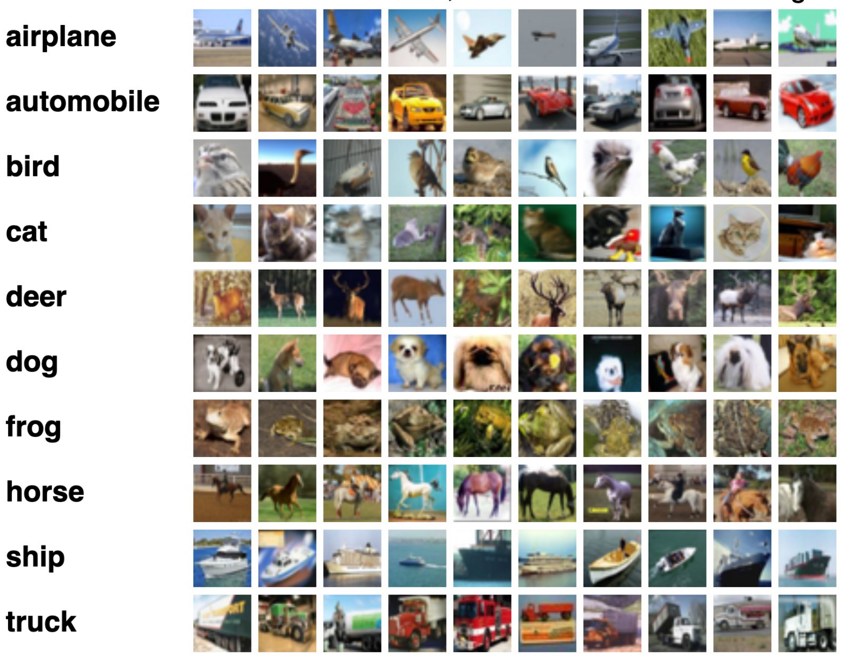
** Adam: A method for stochastic optimization*
DP Kingma, J Ba [1412.6980 [cs.LG]]
> 220k cites (Google Scholar)

Weight matrix initialisation

- weight matrices need to be initialised
- usual choice $W_{ij} \sim \mathcal{N}(0, \sigma_W^2/n_{l-1})$
- introduces another hyperparameter σ_W
- can compute moments of weight matrices at initialisation *
- simplifies in infinite-width limit ("large N limit")

* D Roberts, S Yaida, B Hanin, *The Principles of Deep Learning Theory*, Cambridge University Press [2106.10165 [cs.LG]]

Data sets



what type of data sets are used?

- ‘standard’ data sets: MNIST, CIFAR, Imagenet
- popular in stat. mech. community: teacher-student models
  - random input $x_i \sim \mathcal{N}(0,1)$ teacher NN: random weight matrices $W_{ij}^{(l)} \sim \mathcal{N}(0,1)$
  - produces a random output, $\boldsymbol{y}_\alpha = f(\boldsymbol{x}_\alpha;\theta)$
  - student has to ‘learn’ which weight matrices the teacher used
  - useful for analytical studies and numerical experiments

Outline

- basics of feed-forward neural networks (NNs) and stochastic gradient descent (SGD)
- **stochastic gradient descent, Dyson Brownian motion and random matrix theory**
- examples: restricted Boltzmann machines, transformers
- neural network phase diagram
- outlook

Stochastic weight matrix dynamics

- consider some $M \times N$ weight matrix W
- update using stochastic gradient descent: $W \to W' = W + \delta W$ with $\delta W = -\alpha \frac{\delta \mathcal{L}}{\delta W}$
- obtained from loss function $\mathcal{L}[W]$, learning rate (or step size) α
- δW is estimated using a batch $\mathcal{B}$ with batch size $|\mathcal{B}|$: $\delta W_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \delta W_b$
- fluctuations controlled by finite batch size (CLT): $\frac{1}{\sqrt{|\mathcal{B}|}} \sqrt{\mathbb{V}[\delta W)]}$

Stochastic weight matrix dynamics

- using CLT stochastic update $W \to W' = W + \delta W$ becomes

$$\delta W = \mathbb{E}\left[\delta W\right] + \frac{1}{\sqrt{|\mathcal{B}|}}\sqrt{\mathbb{V}[\delta W)]}\,\eta \qquad \eta_{ij} \sim \mathcal{N}(0,1)$$

- or in terms of the gradient of the loss function:

$$W' = W - \alpha\mathbb{E}\left[\frac{\delta\mathcal{L}}{\delta W}\right] + \frac{\alpha}{\sqrt{|\mathcal{B}|}}\sqrt{\mathbb{V}\left[\frac{\delta\mathcal{L}}{\delta W}\right]}\,\eta$$

From rectangular to symmetric matrices

- W is $M \times N$ matrix: singular value decomposition: $W = U \Xi V^T$ $\quad UU^T = \mathbb{1} \quad VV^T = \mathbb{1}$
- singular values: $\xi_i \quad (i = 1 \dots N)$ [take $N \leq M$ without loss of generality]
- introduce symmetric semi-positive combination: $X = W^T W = V D V^T$
- and focus on the singular/eigenvalues (invariant under left/right rotations on W):

$$D = \Xi^T \Xi = \mathrm{diag}\left(\xi_1^2, \dots, \xi_N^2\right) = \mathrm{diag}\left(x_1, \dots, x_N\right)$$

- stochastic dynamics: $$X \to X' = X + \mathbb{E}\left[\delta X\right] + \frac{1}{\sqrt{|\mathcal{B}|}} \sqrt{\mathbb{V}\left[\delta X\right]}\, \eta$$

Initialisation: Marchenko-Pastur distribution

- if initial weight matrix $W_{ij} \sim \mathcal{N}(0, \sigma^2)$ then X follows Marchenko-Pastur distribution

$$P_{\mathrm{MP}}(x) = \frac{1}{2\pi\sigma^2 M r x}\sqrt{(x_+ - x)(x - x_-)} \qquad x_- < x < x_+ \qquad r = N/M \leq 1 \qquad x_\pm = M\sigma^2\left(1 \pm \sqrt{r}\right)^2$$

- ✓ how to choose σ^2 : distribution should depend on r only, safe to take large N, M limit

$$N \leq M$$

- ✓ spectrum is bounded for all r : $\sigma^2 = 1/M$

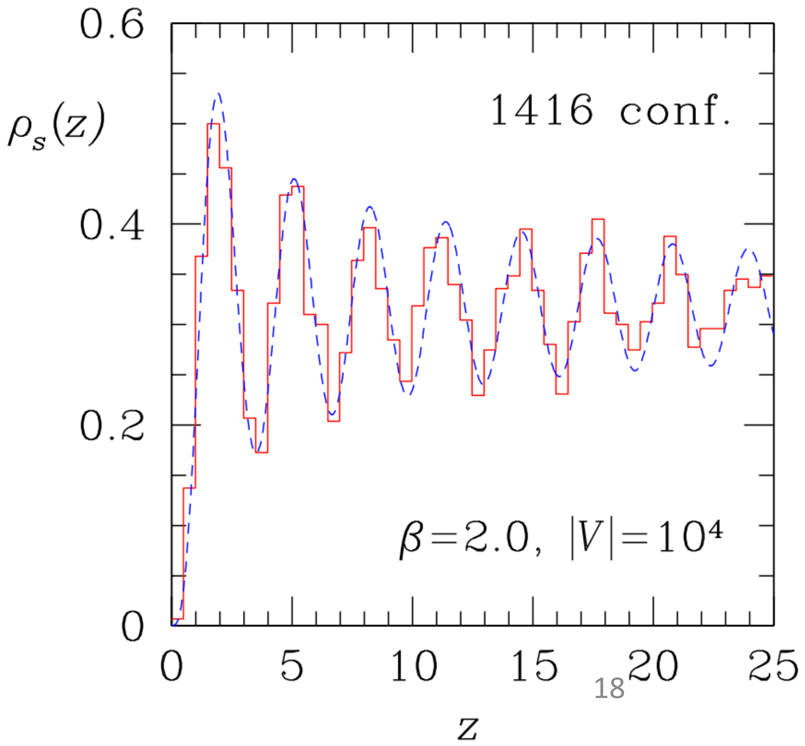
$$P_{\mathrm{MP}}(x) = \frac{1}{2\pi r x}\sqrt{(x_+ - x)(x - x_-)} \qquad 0 \leq x_- \leq x \leq x_+ \leq 4 \qquad x_\pm = \left(1 \pm \sqrt{r}\right)^2$$

Stochastic matrix dynamics

- what is the framework to consider stochastic matrix dynamics?
- goes back to **Wigner** (1955) and **Dyson** (1962): random matrix theory (RMT)
- stochastic matrix dynamics: Dyson Brownian motion (**Dyson**, 1962)
- first applied to nuclear spectra (1950/60s)
- applied in (lattice) QCD to spectrum of Dirac operator

Random Matrix Theory (RMT)

- describes universal features of matrices in symmetry classes (symmetric, hermitian, quaternionic)
- level spacing, Coulomb repulsion, Wigner surmise, fluctuations
- non-universal behaviour: spectral density
- successfully applied in QCD to describe Dirac operator



JJM Verbaarschot and T Wettig, *Random matrix theory and chiral symmetry in QCD*
Ann. Rev. Nucl. Part. Sci. 50 (2000) 343 [hep-ph/0003017 [hep-ph]]

Stochastic matrix dynamics: Dyson Brownian motion and the Coulomb gas

- framework to consider stochastic matrix dynamics for symmetric matrix X

- Dyson Brownian motion (in continuous time for now, see below):

$$\frac{dX_{ij}}{dt} = K_{ij}(X) + \sqrt{A_{ij}}\eta_{ij}$$

- eigenvalues then evolve according to

$$\frac{dx_i}{dt} = K_i(x_i) + \sum_{j \neq i} \frac{g_i^2}{x_i - x_j} + \sqrt{2}g_i\eta_i$$

$$\equiv K_i^{(\mathrm{eff})}(x_i) + \sqrt{2}g_i\eta_i$$

where $\sqrt{A_{ii}} = \sqrt{2}g_i$

Dyson Brownian motion and Coulomb gas

- eigenvalues dynamics: $$\frac{dx_i}{dt} = K_i(x_i) + \sum_{j \neq i} \frac{g_i^2}{x_i - x_j} + \sqrt{2} g_i \eta_i$$
- can be derived using 2nd order perturbation theory (some conditions on noise matrix A_{ij})
- Coulomb term: eigenvalue repulsion
- Fokker-Planck equation (FPE) for distribution of eigenvalues:

$$\partial_t P(\{x_i\}, t) = \sum_{i=1}^{N} \partial_{x_i} \left[\left(g_i^2 \partial_{x_i} - K_i^{(\mathrm{eff})}(x_i) \right) \right] P(\{x_i\}, t)$$

Dyson Brownian motion and Coulomb gas

- FPE: $$\partial_t P(\{x_i\},t) = \sum_{i=1}^{N} \partial_{x_i} \left[\left(g_i^2 \partial_{x_i} - K_i^{(\mathrm{eff})}(x_i) \right) \right] P(\{x_i\},t)$$
- stationary distribution: $$P_s(\{x_i\}) = \frac{1}{Z} \prod_{i<j} |x_i - x_j| \, e^{-\sum_i V_i(x_i)/g_i^2}$$
- with partition function: $$Z = \int dx_1 \dots dx_N \, P_s(\{x_i\})$$
- and provided drift can be derived from a potential $$K_i(x_i) = -\frac{dV_i(x_i)}{dx_i}$$
- known as Coulomb gas, describes universal features of random matrices

Back to weight matrix dynamics

- stochastic dynamics $$X \to X' = X + \mathbb{E}[\delta X] + \frac{1}{\sqrt{|\mathcal{B}|}}\sqrt{\mathbb{V}[\delta X)]}\,\eta$$
- what can be carried over from Dyson's matrix dynamics? implications? universality?
- eigenvalue equation: $$x_i \to x_i' = x_i + \delta x_i + \sum_{j \neq i} \frac{g_i^2}{x_i - x_j} + \sqrt{2} g_i \eta_i$$
- make explicit learning rate and batch size dependence

$$\delta x_i = \alpha K_i \qquad g_i = \frac{\alpha}{\sqrt{|\mathcal{B}|}}\tilde{g}_i \qquad \tilde{g}_i \sim \mathbb{V}[\delta \mathcal{L}/\delta W]\,\big|_{ii}$$

Back to weight matrix dynamics

- eigenvalue dynamics: $$x_i \to x_i' = x_i + \delta x_i + \sum_{j \neq i} \frac{g_i^2}{x_i - x_j} + \sqrt{2} g_i \eta_i$$
- insert learning rate and batch size dependence:

$$x_i \to x_i' = x_i + \alpha K_i + \frac{\alpha^2}{|\mathcal{B}|} \sum_{j \neq i} \frac{\tilde{g}_i^2}{x_i - x_j} + \frac{\alpha}{\sqrt{|\mathcal{B}|}} \sqrt{2} \tilde{g}_i \eta_i$$

- no usual scaling of drift and noise with learning rate (Ito calculus: ϵ , $\sqrt{\epsilon}$)
- only continuous time limit (SDE) in some weak sense

Q Li, C Tai and W E [1511.06251]
S Yaida [1810.00004]

$$x_i \to x_i' = x_i + \alpha K_i + \frac{\alpha^2}{|\mathcal{B}|} \sum_{j \neq i} \frac{\tilde{g}_i^2}{x_i - x_j} + \frac{\alpha}{\sqrt{|\mathcal{B}|}} \sqrt{2} \tilde{g}_i \eta_i$$

Stationary distribution

- distribution for fixed $\alpha, |\mathcal{B}|$: $$P_s(\{x_i\}) = \frac{1}{Z} \prod_{i<j} |x_i - x_j| \, e^{-\sum_i V_i(x_i)/g_i^2}$$
- make explicit dependence on learning rate and batch size

$$g_i = \frac{\alpha}{\sqrt{|\mathcal{B}|}} \tilde{g}_i \qquad V_i(x_i) = \alpha \tilde{V}_i(x_i) \qquad \frac{V_i(x_i)}{g_i^2} = \frac{1}{\alpha/|\mathcal{B}|} \frac{\tilde{V}_i(x_i)}{\tilde{g}_i^2}$$

- if drift vanishes at $x_i = x_i^s$ expand potential $\tilde{V}_i(x_i) = \tilde{V}_i(x_i^s) + \frac{1}{2} \Omega_i \, (x_i - x_i^s)^2 + \ldots$
- exponential is Gaussian with variance $\sigma_i^2 = (\alpha/|\mathcal{B}|) \; (\tilde{g}_i^2/\Omega_i)$

universal scaling with learning rate and batch size — model-dependent factor

Linear scaling relation

- dependence on $\alpha/|\mathcal{B}|$ in training has been observed before, empirically

✓ P. Goyal, P. Dollár, R.B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola et al.,
Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour [1706.02677]

✓ S.L. Smith and Q.V. Le,
A Bayesian Perspective on Generalization and Stochastic Gradient Descent [1710.06451]

✓ S.L. Smith, P. Kindermans and Q.V. Le,
Don't Decay the Learning Rate, Increase the Batch Size [1711.00489]

- finds a natural place in the framework of Dyson Brownian motion and Coulomb gas

GA, B Lucini, **Chanju Park**, PRE 111 (2025) 1, 015303 [2407.16427 [cond-mat.dis-nn]]

Outline

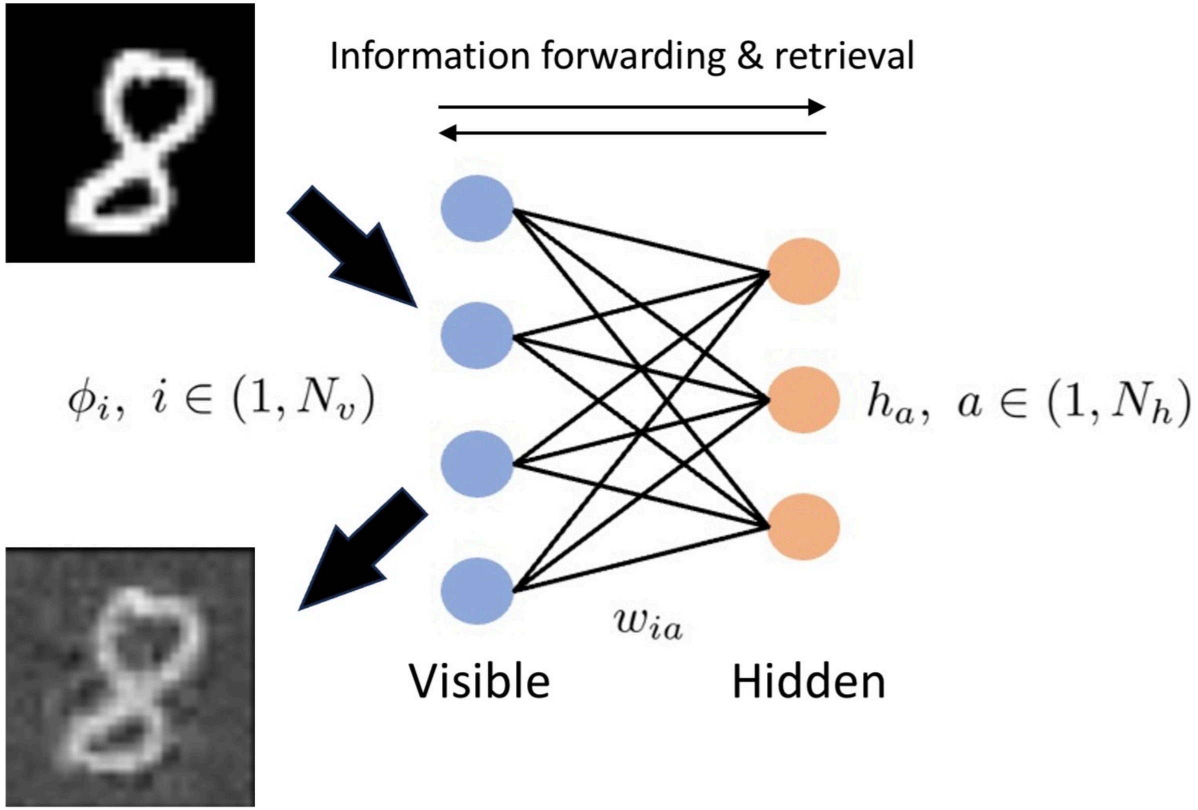
- basics of feed-forward neural networks (NNs) and stochastic gradient descent (SGD)
- stochastic gradient descent, Dyson Brownian motion and random matrix theory
- **examples: restricted Boltzmann machines, transformers**
- neural network phase diagram
- outlook

Manifestations of RMT in weight matrices

RMT predicts universal behaviour:

- universal distribution of level spacing $S_i = x_{i+1} - x_i$: Wigner surmise $P(S)$
- Coulomb repulsion of eigenvalues
- spectral density is problem-specific $\rho(x) = \left\langle \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i) \right\rangle$
- universal behaviour has indeed been observed for variety of ML algorithms and data sets
- some examples: restricted Boltzmann machine and transformer

Restricted Boltzmann Machine: generative network



- energy-based method
- probability distribution
- binary or continuous d.o.f.

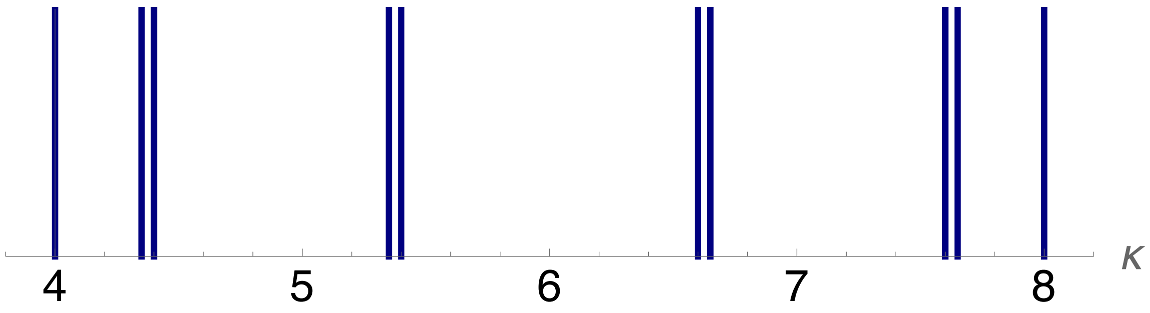
$$p(\phi, h) = \frac{1}{Z} e^{-S(\phi,h)}$$

$$Z = \int D\phi Dh\, e^{-S(\phi,h)}$$

one weight matrix: bilinear coupling: $\phi^T W h = \sum_{ij} \phi_i W_{ia} h_a$

Learning task

- target spectrum should be reflected in weight matrix, i.e. in $X = W^T W$
- evolution from initial Marchenko-Pastur distribution to target distribution
- updates using persistent contrastive divergence with mini-batches
- vary learning rate and batch size
- use very simple target spectrum:
  LFT dispersion relation in 1d
  doubly degenerate modes



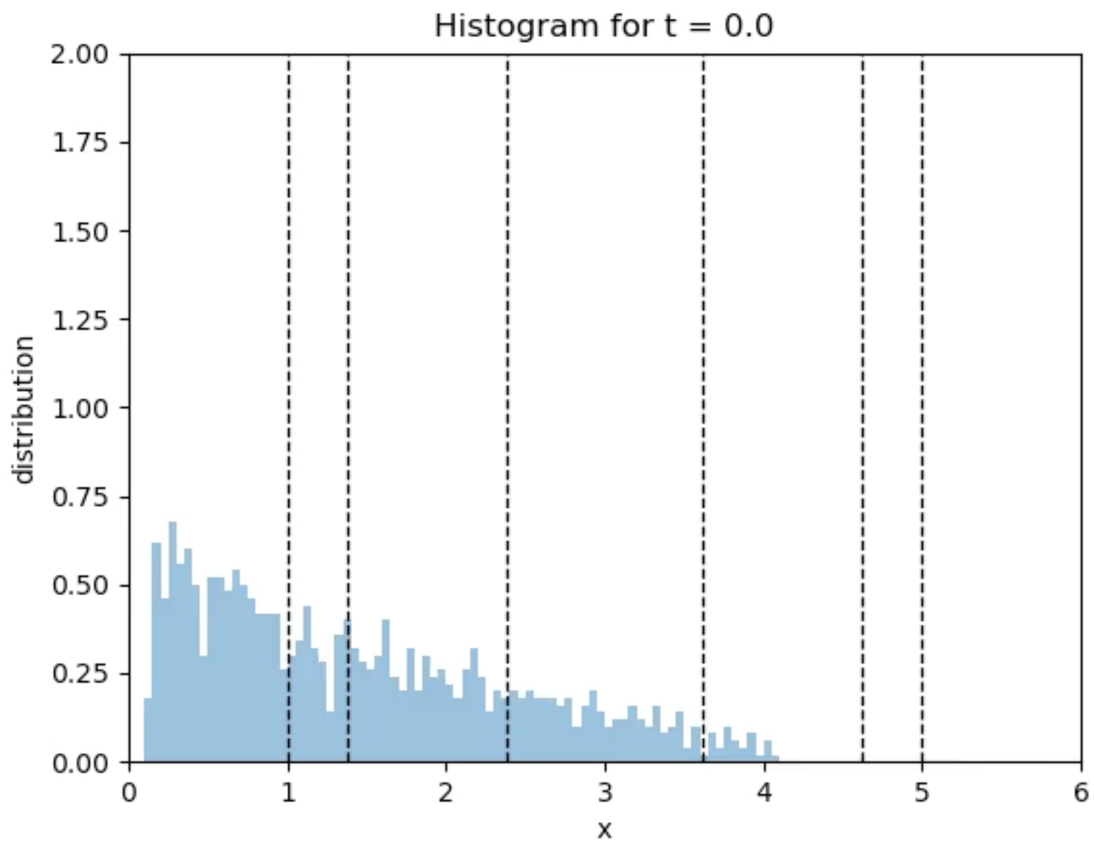
Dynamics of learning

- from Marchenko-Pastur distribution
  to stochastic target distribution

- 10 modes, 4 doubly degenerate ones

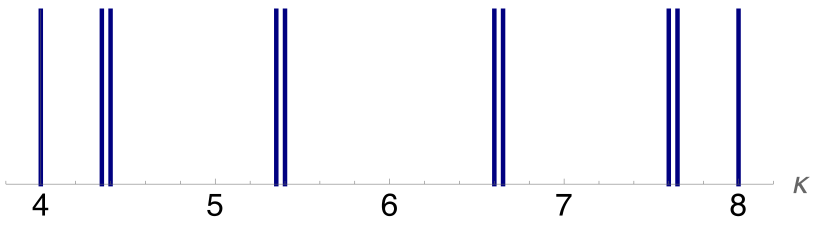
- follow evolution of eigenvalues

- test predictions from RMT:

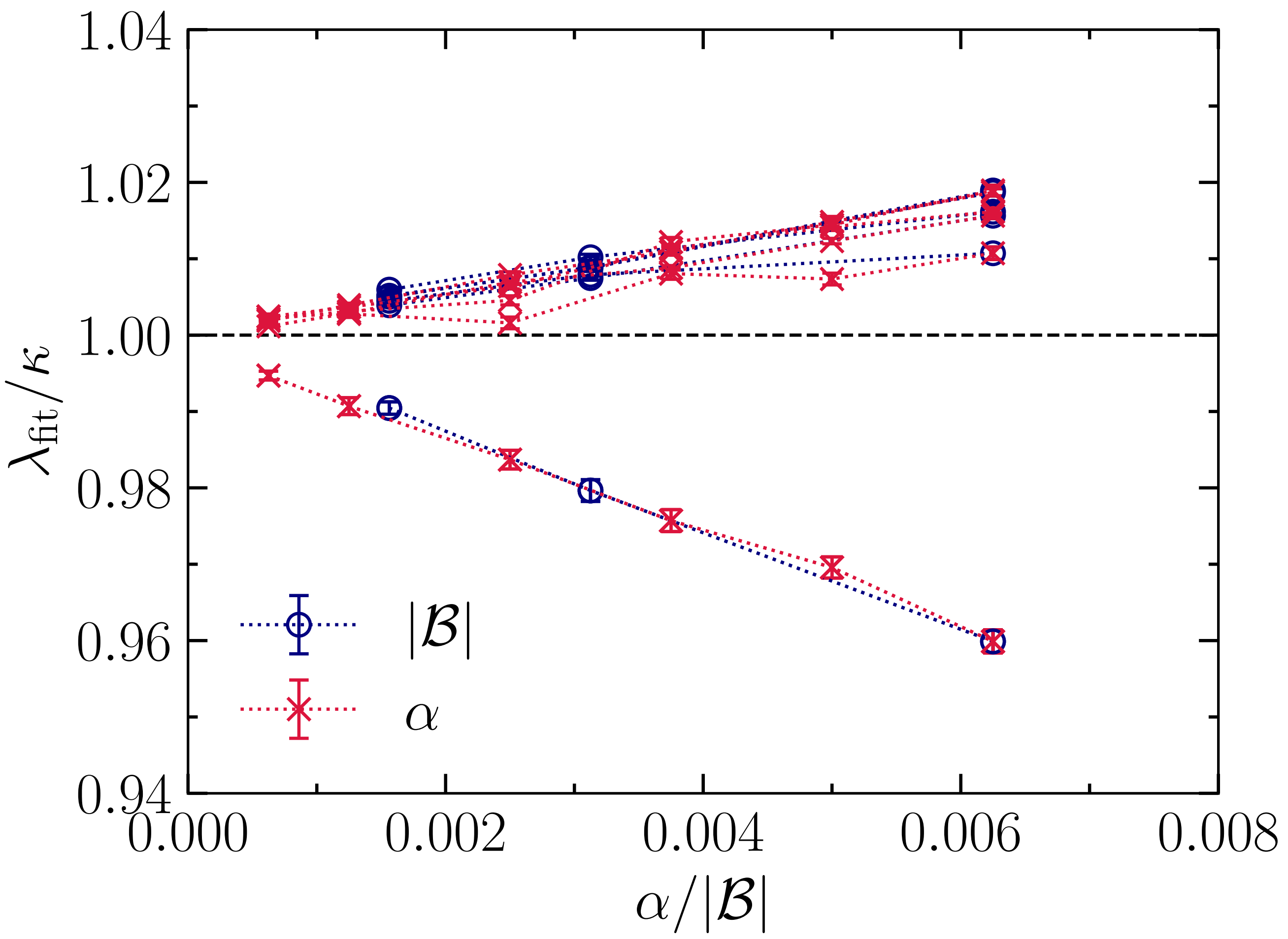


  - induced Coulomb term, eigenvalue repulsion, Wigner surmise
  - dependence on learning rate/batch size

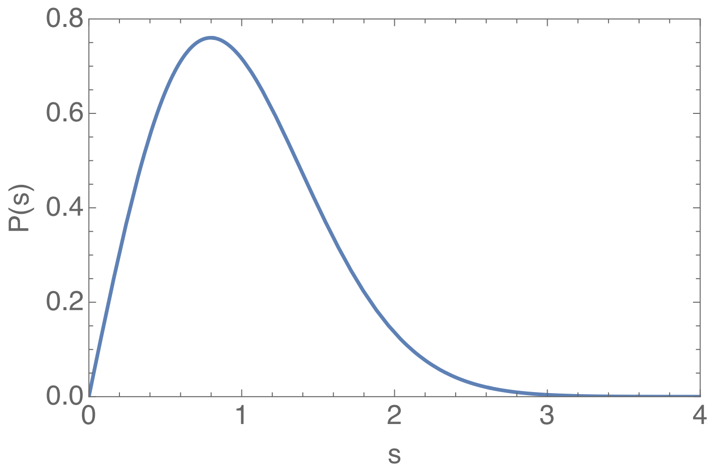
Eigenvalue repulsion



- Coulomb interaction between all eigenvalues
- learned eigenvalue/target
- repulsion for nonzero learning rate/batch size
- no "perfect learning" unless stochasticity vanishes
- overfitting, generalisation, …



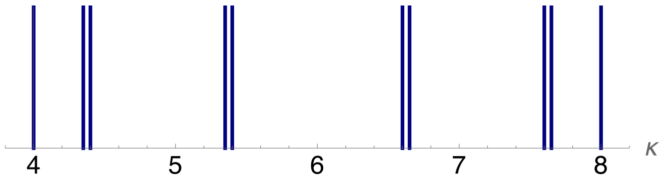
Wigner surmise*



- distribution $P(S) = \frac{S}{2\sigma^2} e^{-S^2/(4\sigma^2)}$, level spacing $S = x_1 - x_2$
- mean level spacing $\langle S \rangle = \int_0^\infty dS \, S P(S) = \sqrt{\pi}\sigma$.
- Wigner surmise for $s = S/\langle S \rangle$: $P(s) = \frac{\pi}{2} s e^{-\pi s^2/4}$ universal curve

- many RBM training runs, stochasticity due to mini-batches, collect histograms of x_i
- vary learning rate and batch size

* expression is derived in the exercises

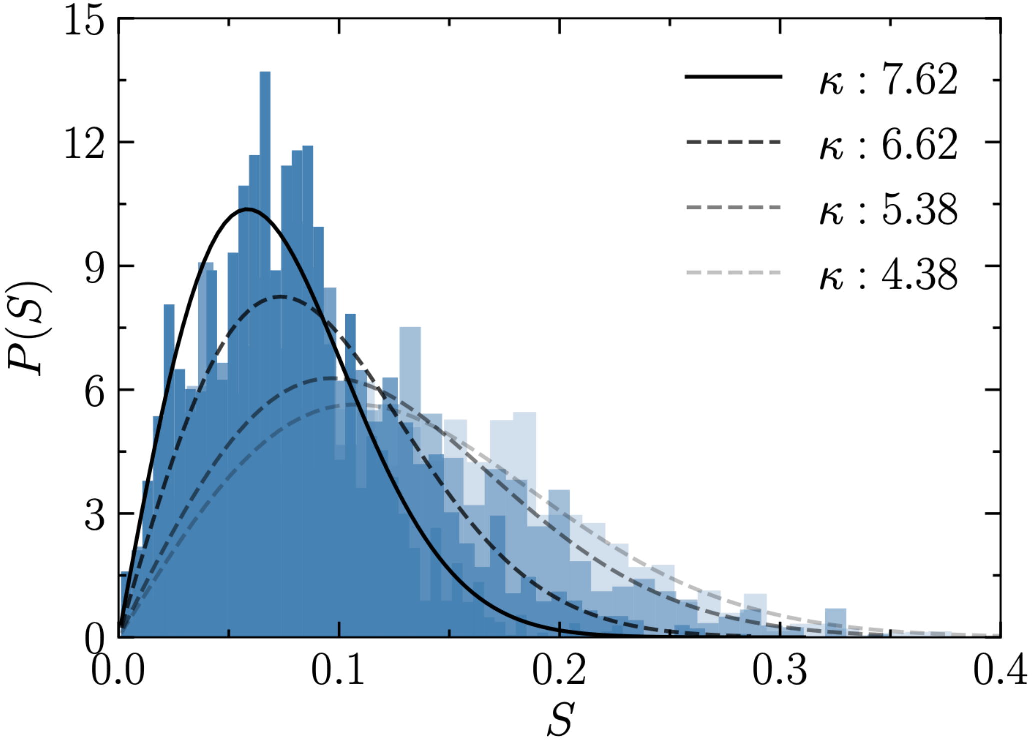


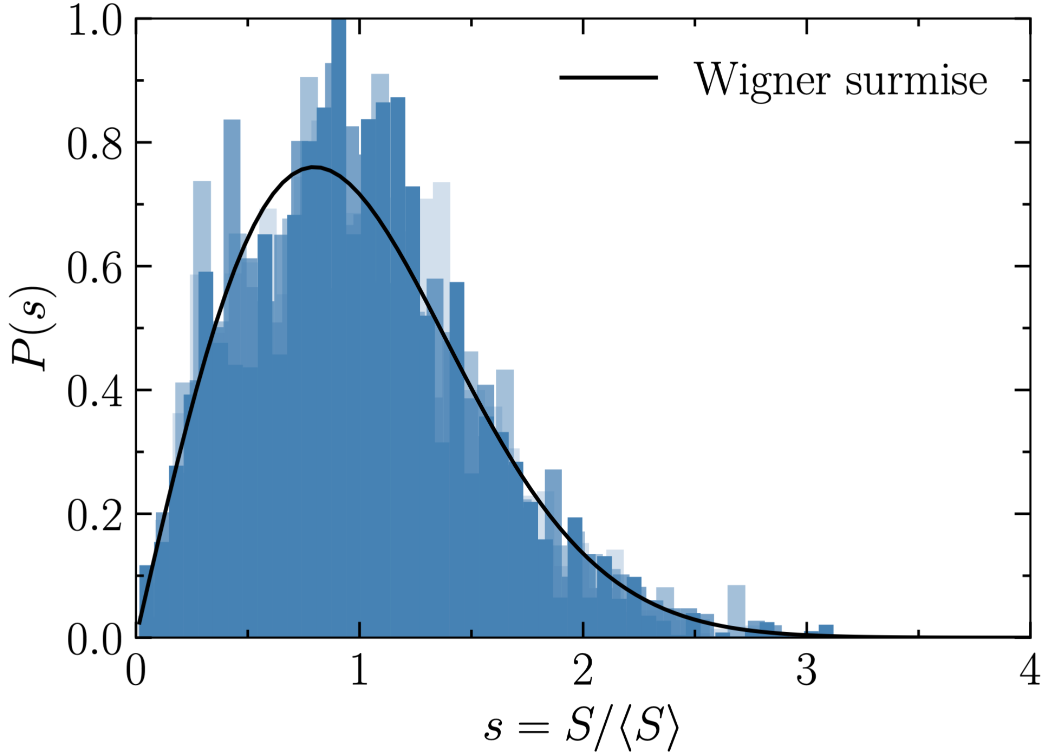
Wigner surmise: 4 degenerate pairs

$$P(S) = \frac{S}{2\sigma^2} e^{-S^2/(4\sigma^2)}$$

$$\langle S \rangle = \sqrt{\pi}\sigma$$

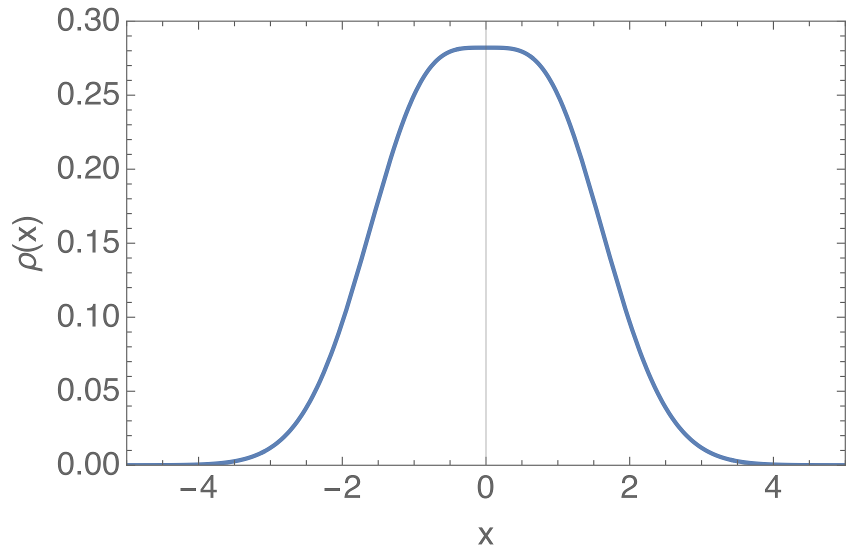
$$P(s) = \frac{\pi}{2} s e^{-\pi s^2/4}$$





data collapse

universality



Wigner semi-circle

- spectral density: $$\rho(x) = \left\langle \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i) \right\rangle$$
- for two modes:

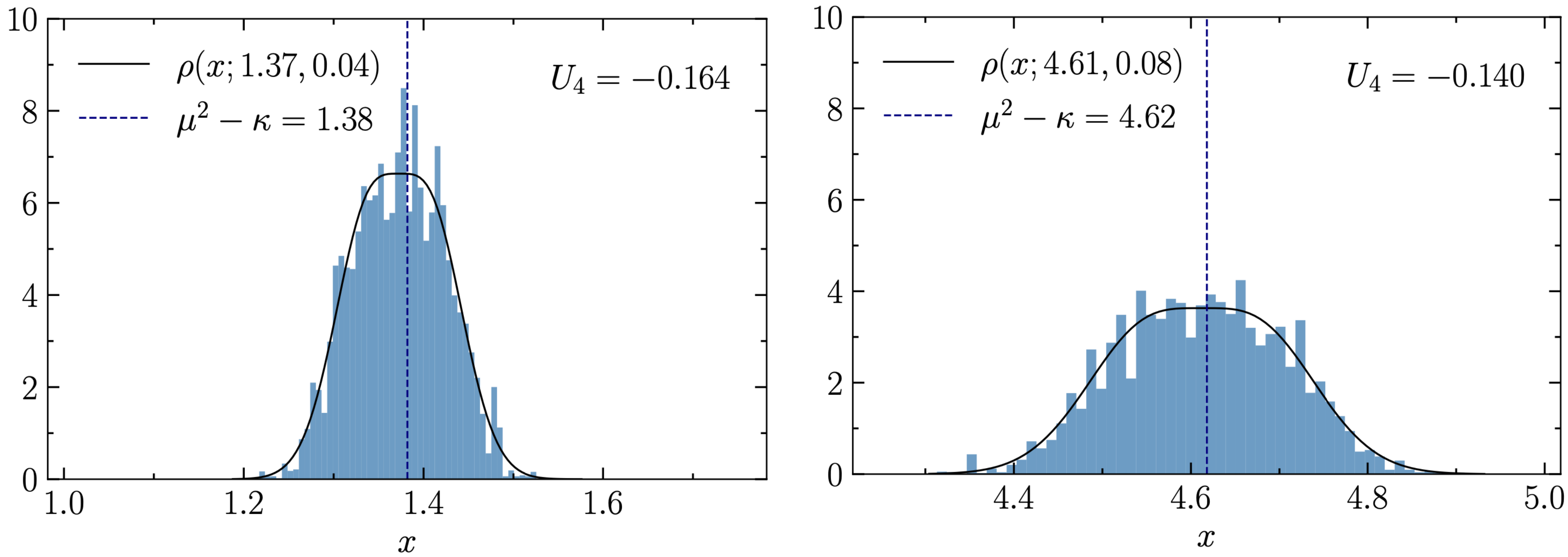
$$\rho(x) = \frac{e^{-x^2/(2\sigma^2)}}{4\sqrt{\pi}\sigma} \left[2e^{-x^2/(2\sigma^2)} + \sqrt{2\pi}\frac{x}{\sigma}\mathrm{Erf}\left(\frac{x}{\sqrt{2}\sigma}\right) \right]$$

- broadened and flattened Gaussian
- fit σ parameter and position for each doubly degenerate mode

$$\rho(x) = \frac{e^{-x^2/(2\sigma^2)}}{4\sqrt{\pi}\sigma}\left[2e^{-x^2/(2\sigma^2)} + \sqrt{2\pi}\frac{x}{\sigma}\mathrm{Erf}\left(\frac{x}{\sqrt{2}\sigma}\right)\right]$$

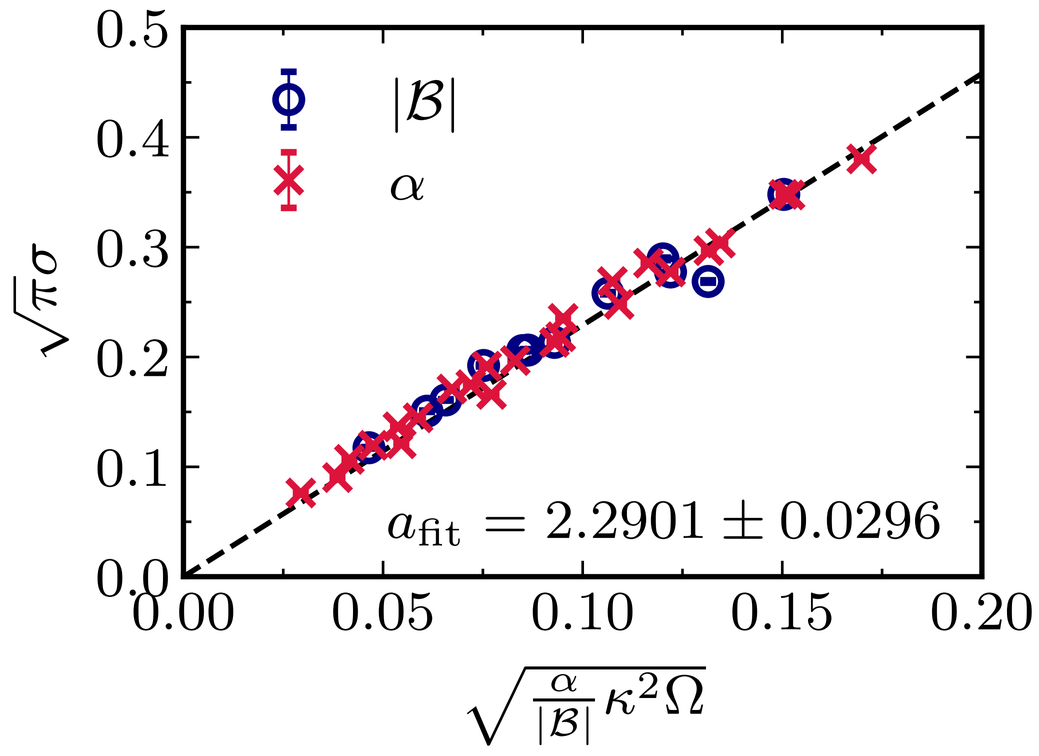
Wigner semi-circle

- fit to semi-circle for two different κ_i values with fixed learning rate and batch size
- Binder cumulant $U_4 = -4/27 \approx -0.148$ for semi-circle (vanishes for Gaussian)

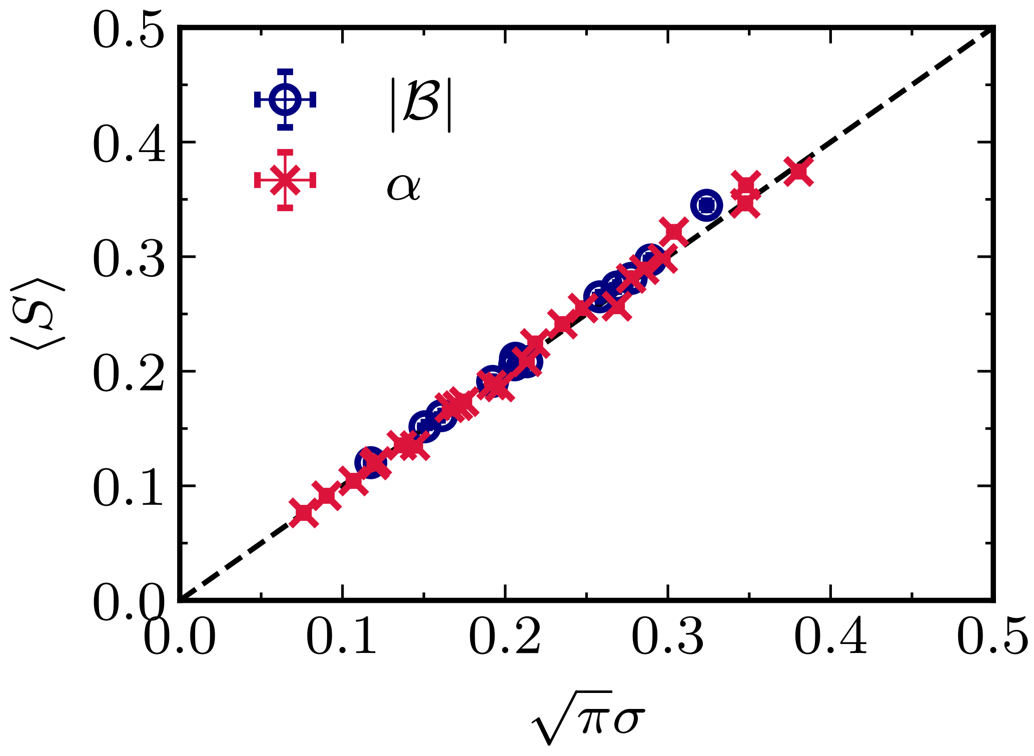


Wigner semi-circle and surmise

semi-circle
dependence on learning rate/batch size



consistency between surmise
and semi-circle fits



RBM: Wigner surmise and semi-circle

- ✓ parameter σ scales as: $\sigma_i^2 = (\alpha/|\mathcal{B}|)\;(\tilde{g}_i^2/\Omega_i)$

  universal scaling   model-dependent

- ✓ stochasticity leads to universal features in trained models

- ✓ derived and demonstrated that learning rate and finite batch size appear as ratio

  (linear scaling rule)

Second application: Transformers

- Gaussian RBM has one weight matrix, target spectrum is known, essentially solvable

in more advanced architectures:

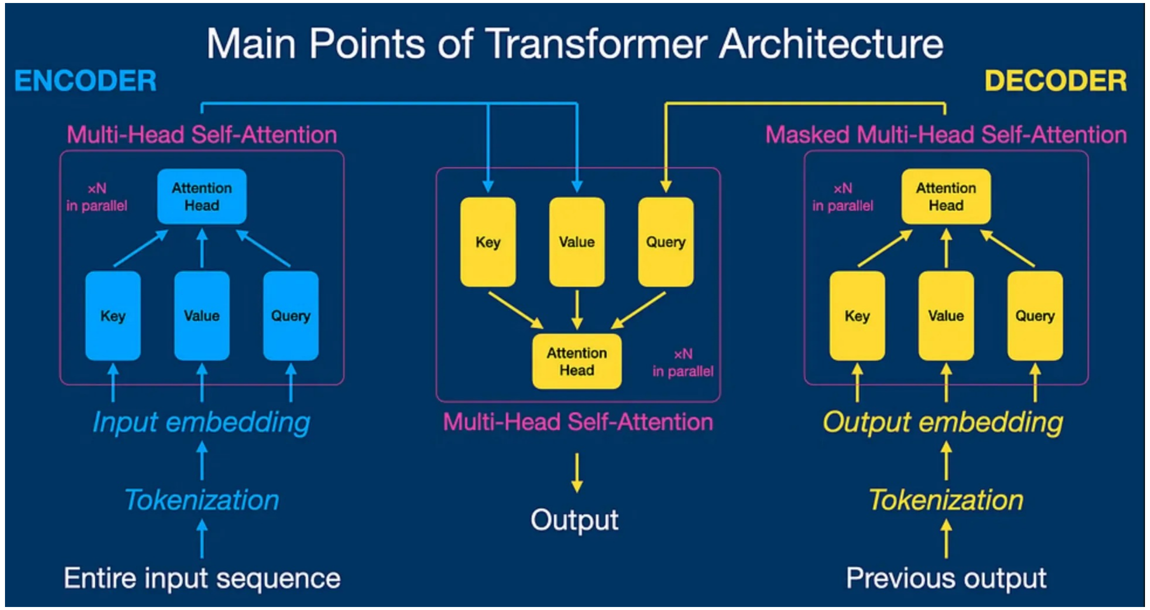
- many weight matrices, target spectra not known, do the spectra even exist?
- what is the loss function landscape? localised minima, flat directions, … ?
- empirical study following dynamics of eigenvalues of $X = W^T W$

GA, O Hajizadeh, B Lucini, C Park, NeurIPS 2024 workshop *ML and the Physical Sciences*, 2411.13512 [cond-mat.dis-nn]

Transformer: nano-GPT

- four attention blocks with each four attention heads: many more matrices
- each attention head:
  - one key (K) matrix
  - one query (Q) matrix
  - one value (V) matrix
- matrix sizes: $M \times N = 64 \times 16$
- about 2.1×10^5 parameters
- use AdamW optimiser

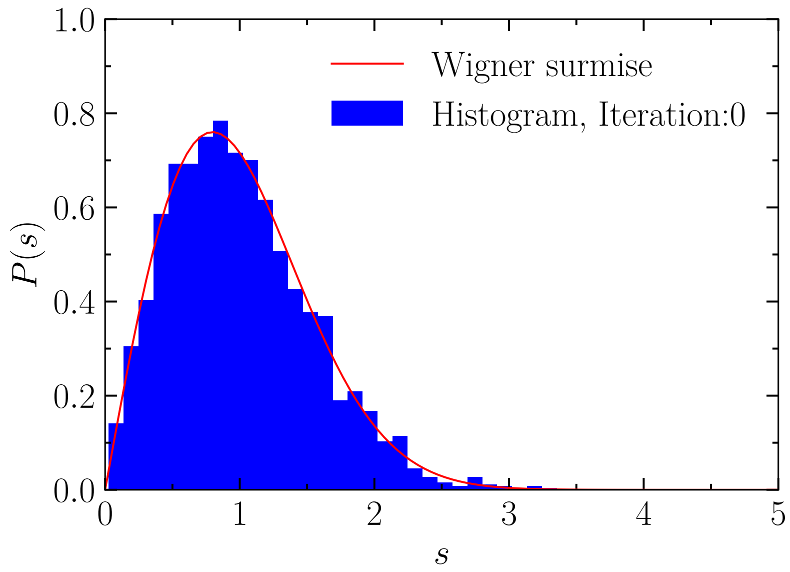
  (highly adaptive stepsize during training)
- trained on opus of Shakespeare



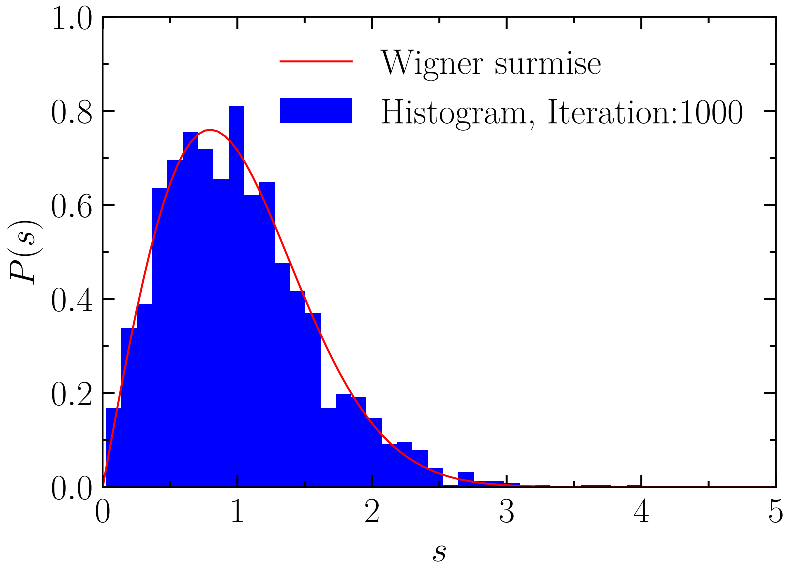
https://botpenguin.com/glossary/transformer-architecture

Transformer: Wigner surmise

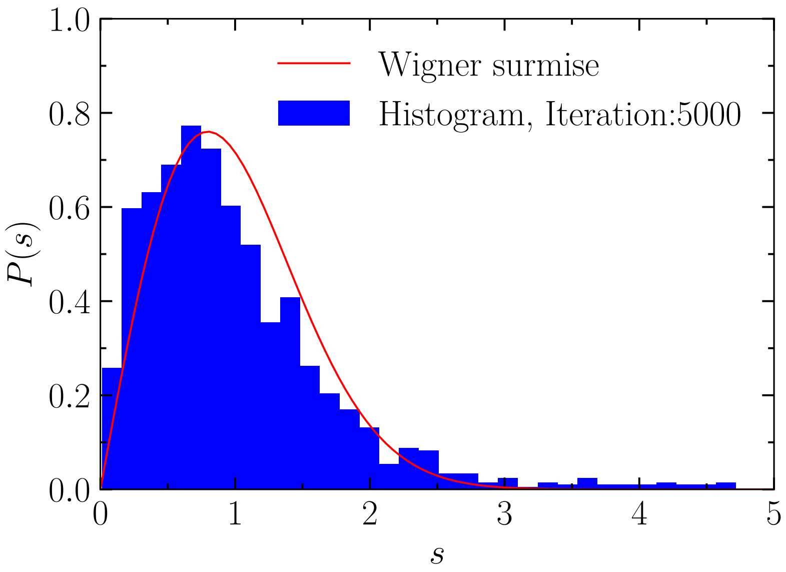
- short-distance fluctuations: level spacing described by Wigner surmise
- remains approximately described by RMT prediction (shown K matrix of layer 1)



iteration 0



iteration 1000

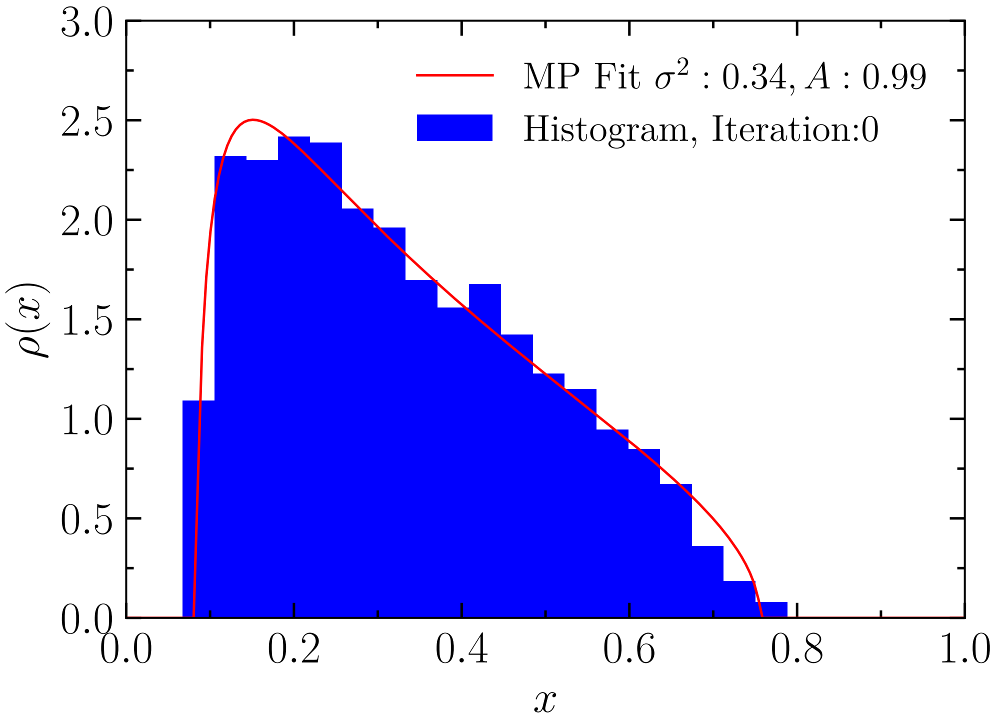


iteration 5000

Transformer: spectral density

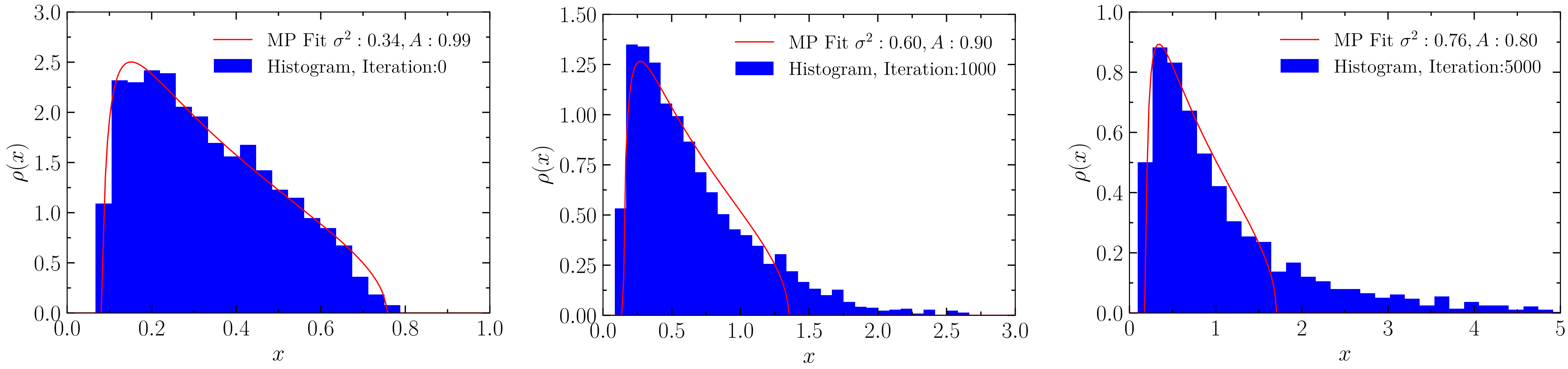
- initialisation: eigenvalues of $X = W^T W$ follow Marchenko-Pastur distribution

$$P_{\mathrm{MP}}(x; \sigma^2, A) = \frac{A}{2\pi\sigma^2 r x}\sqrt{(x_+ - x)(x - x_-)}\,\theta(x_+ - x)\theta(x - x_-)$$



Transformer: spectral density

- evolves from initial Marchenko-Pastur distribution to distribution with power decay
- shown K matrix of layer 1



iteration 0 | iteration 1000 | iteration 5000

Transformer: empirical analysis

requires further understanding:

- what is the "final" target spectrum? does it even exist?
- tail drops as a power, what does this imply? can the power be understood?

what if significant part of the spectrum remains MP: random matrix elements

- how relevant is this part of the spectrum? remove? sparse weight matrices?

open research questions!

see e.g. also CH Martin, MW Mahoney, *Traditional and Heavy-Tailed Self Regularization in Neural Network Models*, 1901.08276

Outline

- basics of feed-forward neural networks (NNs) and stochastic gradient descent (SGD)
- stochastic gradient descent, Dyson Brownian motion and random matrix theory
- examples: restricted Boltzmann machines, transformers
- **neural network phase diagram**
- outlook

Phase diagram of neural networks

- stochastic gradient descent introduces stochasticity: strength set by $\alpha/|\mathcal{B}|$
- interpret as effective temperature $T = \alpha/|\mathcal{B}|$
- dependence of learning on hyperparameters
- identify different phases?

C Park, GA, B Lucini,
in preparation

→ distinguish quality and efficiency of learning

- draw analogy to disordered systems and spin glasses

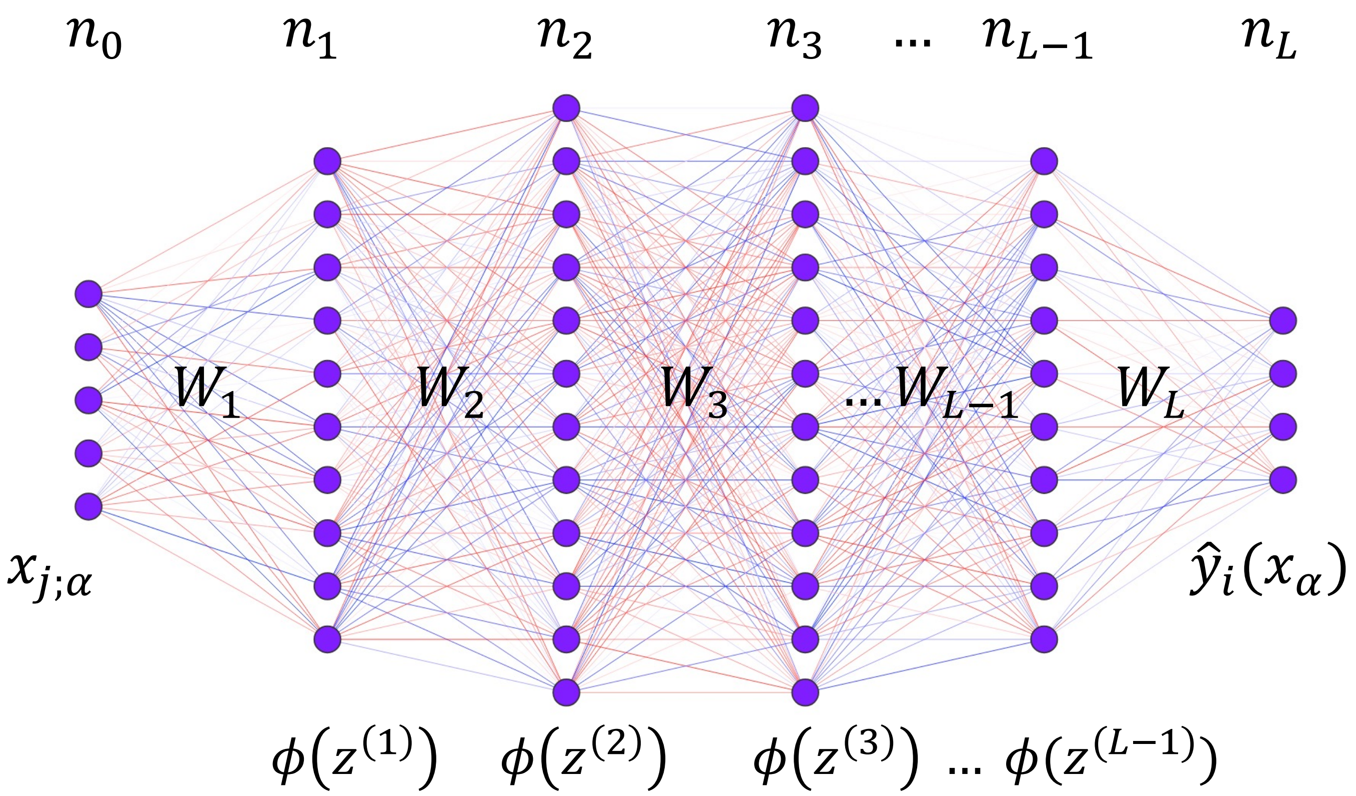
Return to feed-forward neural network

NN function: $$\hat{y}_i(x_\alpha;\theta) \equiv z_i^{(L)}(x_\alpha) = \sum_{j=1}^{n_{L-1}} W_{ij}^{(L)} \phi\left(z_j^{(L-1)}(x_\alpha)\right)$$

pre-activations:

$$z_i^{(l+1)}(x_\alpha) = \sum_{j=1}^{n_l} W_{ij}^{(l+1)} \phi\left(z_j^{(l)}(x_\alpha)\right)$$

$$z_i^{(1)}(x_\alpha) = \sum_{j=1}^{n_0} W_{ij}^{(1)} x_{j;\alpha}$$



Weight matrix initialisation

- weight matrices need to be initialised
- usual choice $W_{ij} \sim \mathcal{N}(0, \sigma_W^2/n_{l-1})$
- introduces another hyperparameter σ_W
- can compute moments of weight matrices at initialisation *

* D Roberts, S Yaida, B Hanin, *The Principles of Deep Learning Theory*, Cambridge University Press [2106.10165 [cs.LG]]

Mean squared error loss function

$$\mathcal{L}(\theta) \equiv \frac{1}{|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \ell\left(y(x_\alpha), \hat{y}(x_\alpha;\theta)\right) \qquad \ell(y, \hat{y}) \equiv \frac{1}{2} \sum_{i=1}^{n_L} (y_i - \hat{y}_i)^2$$

activations on final hidden layer: features $\phi_{j\alpha} \equiv \phi\left(z_j^{(L-1)}(x_\alpha)\right)$

network prediction is linear combination of features

$$\mathcal{L}(\theta) = \frac{1}{2|\mathcal{D}|} \sum_{\alpha=1}^{\mathcal{D}} \sum_{i=1}^{n_L} \left(y_{i\alpha} - \sum_{j=1}^{n_{L-1}} W_{ij}^{(L)} \phi_{j\alpha}\right)^2$$

$$= \frac{1}{2|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \sum_{i=1}^{n_L} \left(\sum_{j,k=1}^{n_{L-1}} W_{ij}^{(L)} W_{ik}^{(L)} \phi_{j\alpha}\phi_{k\alpha} - 2\sum_{j=1}^{n_{L-1}} y_{i\alpha} W_{ij}^{(L)} \phi_{j\alpha} + y_{i\alpha} y_{i\alpha}\right)$$

express loss function as function of features

$$= \frac{1}{2|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \sum_{i,j=1}^{n_{L-1}} J_{ij}\phi_{i\alpha}\phi_{j\alpha} - \frac{1}{|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \sum_{j=1}^{n_{L-1}} h_{j\alpha}\phi_{j\alpha} + C$$

Neural network as a disordered system

loss function as a function of features:

$$\mathcal{L}(\theta) = \frac{1}{2|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \sum_{i,j=1}^{n_{L-1}} J_{ij} \phi_{i\alpha} \phi_{j\alpha} - \frac{1}{|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \sum_{j=1}^{n_{L-1}} h_{j\alpha} \phi_{j\alpha}$$

with couplings:

$$J_{ij} \equiv \sum_{k=1}^{n_L} W_{ki}^{(L)} W_{kj}^{(L)}, \qquad h_{j\alpha} \equiv \sum_{i=1}^{n_L} y_{i\alpha} W_{ij}^{(L)}$$

resembles disordered “spin” system:

$$\mathcal{H} = -\frac{1}{2} \sum_{ij} J_{ij} s_i s_j + \sum_j h_j s_j$$

Solvable Model of a Spin-Glass

David Sherrington* and Scott Kirkpatrick
IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598
(Received 16 October 1975)

We consider an Ising model in which the spins are coupled by infinite-ranged random interactions independently distributed with a Gaussian probability density. Both "spin-glass" and ferromagnetic phases occur. The competition between the phases and the type of order present in each are studied.

$$\mathcal{H} = -\frac{1}{2}\sum_{ij} J_{ij} s_i s_j + \sum_j h_j s_j$$

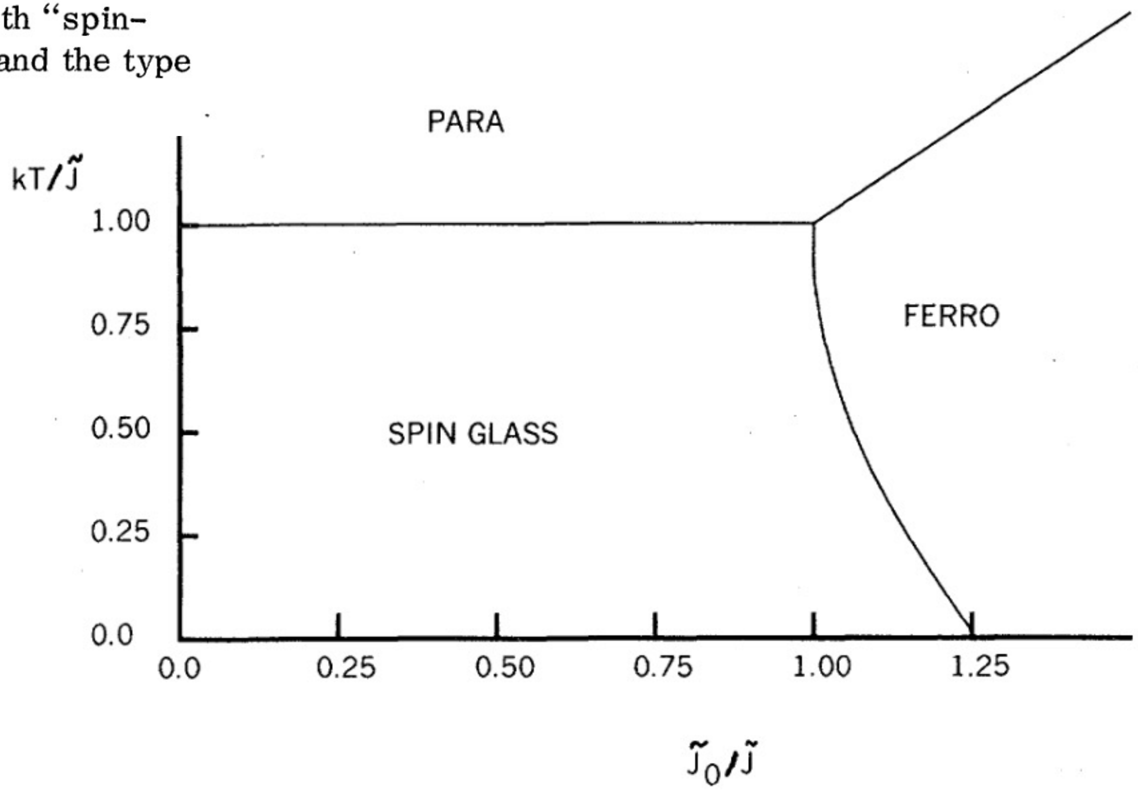


FIG. 1. Phase diagram of spin-glass ferromagnet.

Neural network phase diagram

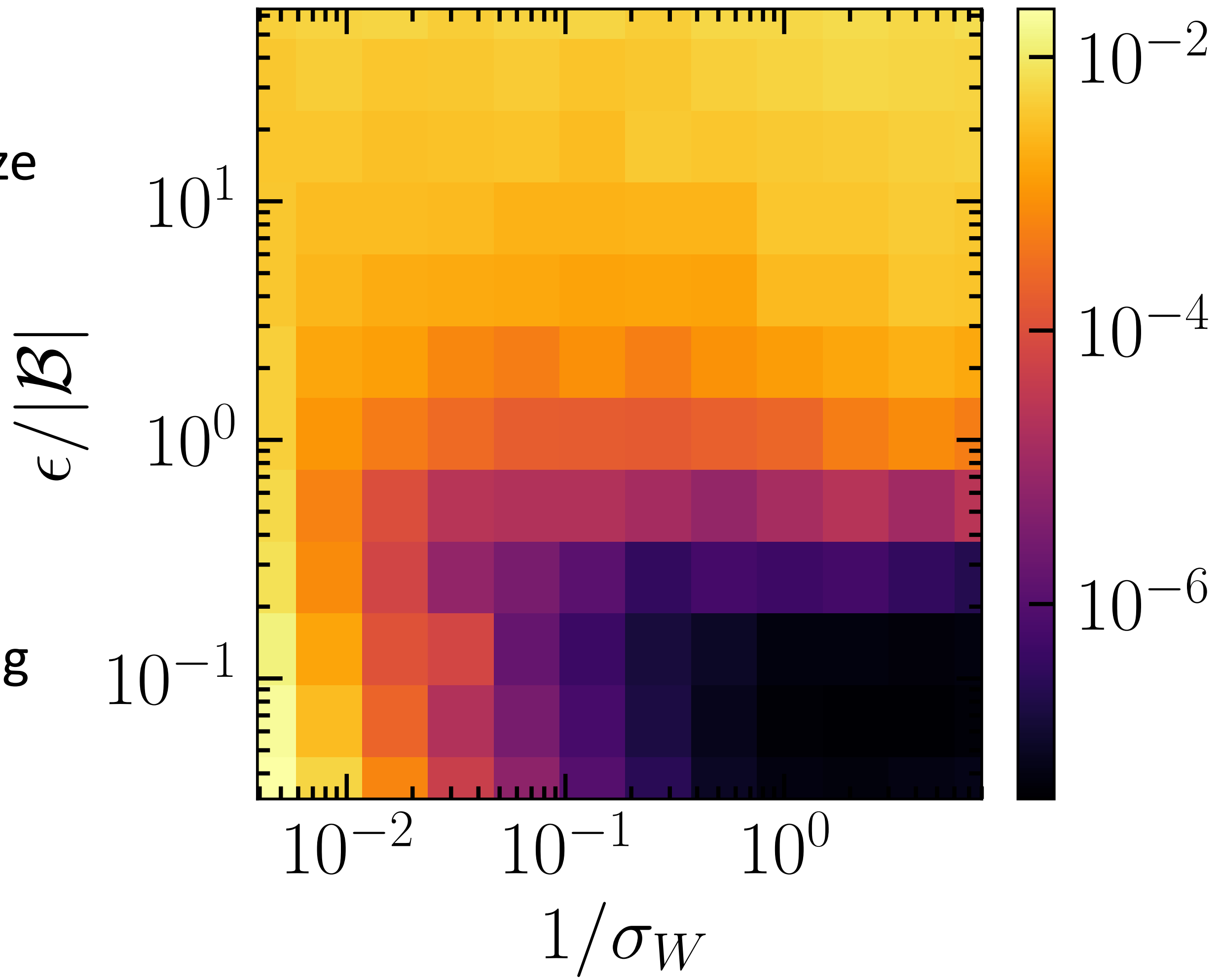
- trained a NN in the teacher-student setup
- teacher network has fixed weight matrices, student has to learn those, or equivalent ones
- two hidden layers [3,32,16,1]
- vary learning rate/batch size $T = \epsilon / |\mathcal{B}|$
- vary initial weight matrix variance $W_{ij}^{(l)} \sim \mathcal{N}\left(0, \sigma_W^2 / n_{l-1}\right)$
- 100 runs for each choice of parameter combination
- monitor number of ''observables'', loss, grad loss, feature alignment, …

NN phase diagram: loss at end of training

effective temperature
~ learning rate/batch size

$$T = \epsilon/|\mathcal{B}|$$

no learning, jamming
~ spin glass phase



no convergence
(loss is large)
~ paramagnetic phase

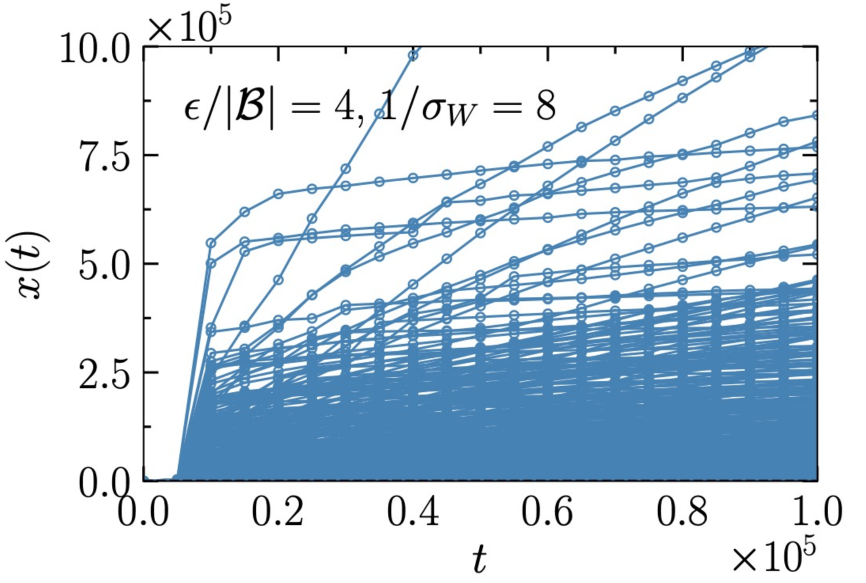
excellent learning
(loss is small)
~ ferromagnetic phase

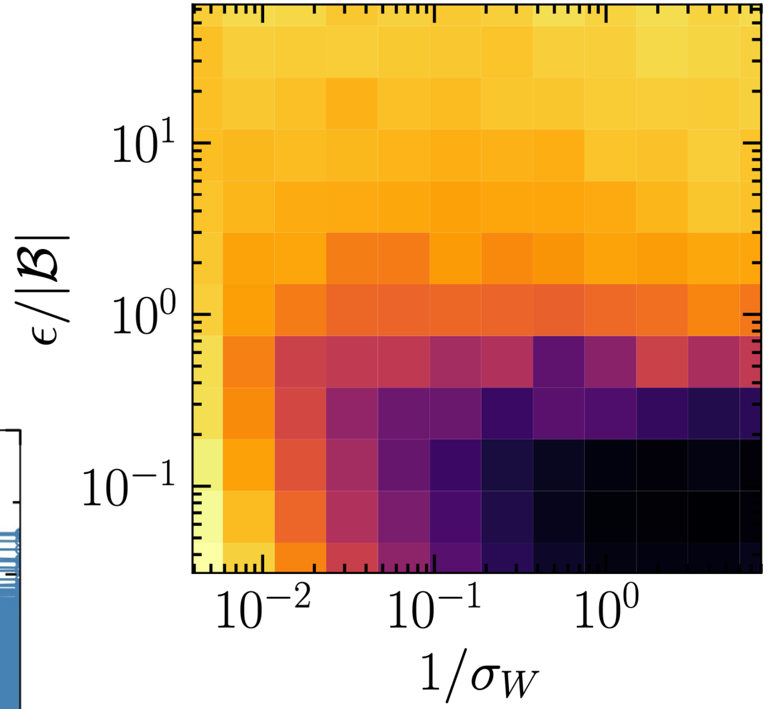
disorder ~ variance of weight matrices upon initialisation

Three distinct phases

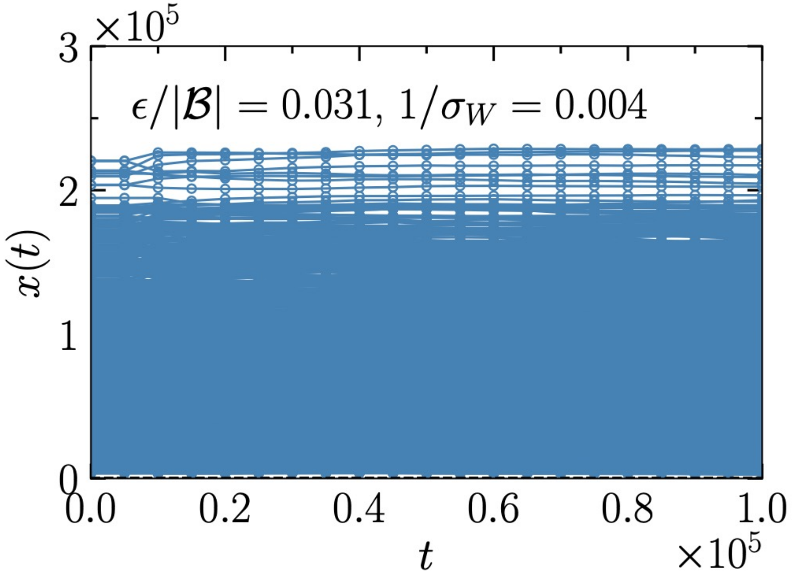
evolution of singular values of weight matrices

divergence

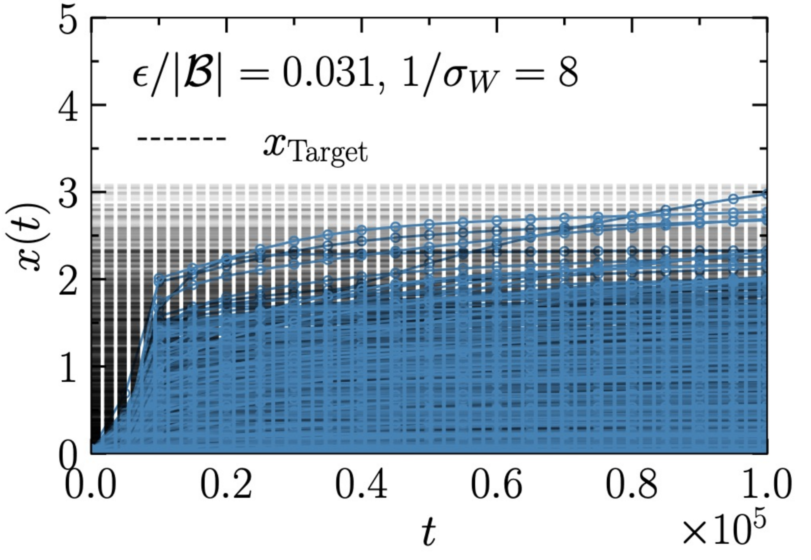




jamming

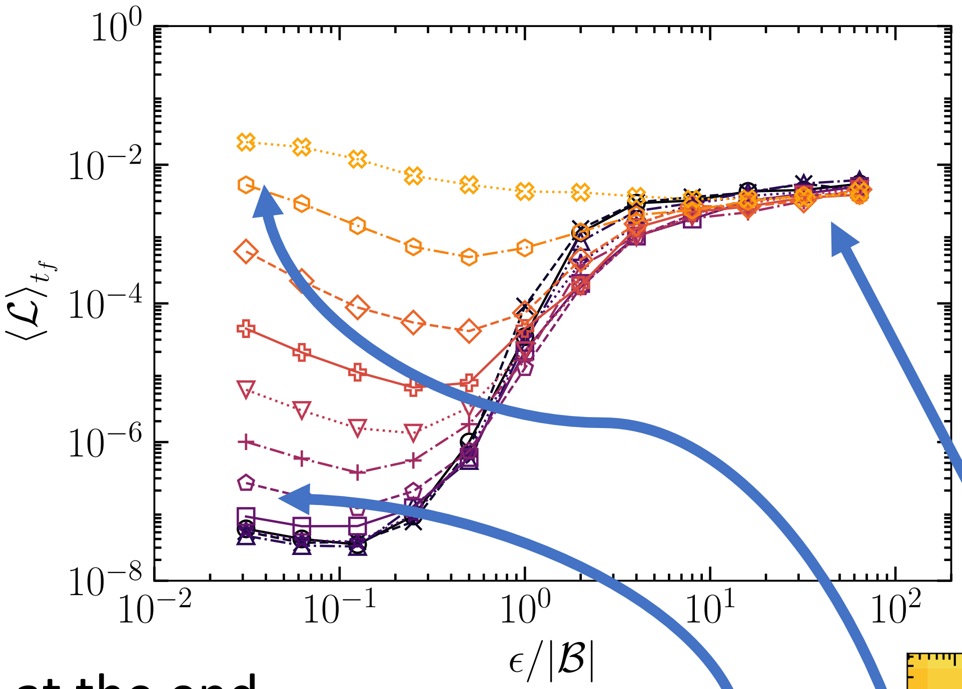


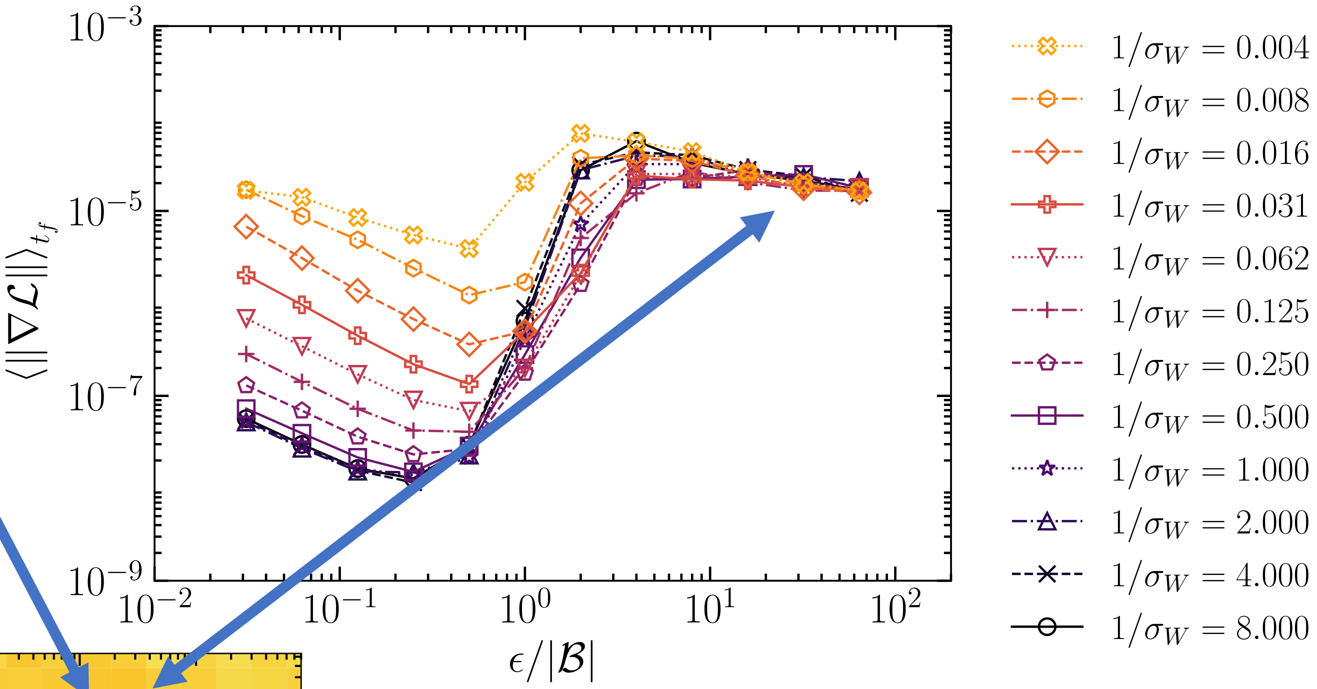
convergence



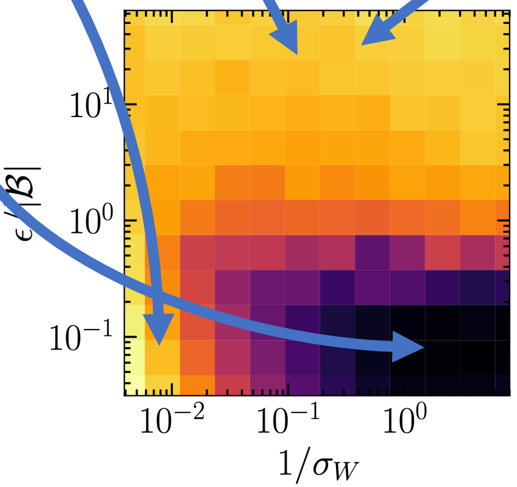
Three distinct phases

gradient of loss at the end





loss at the end



large loss, nonvanishing gradient ✗

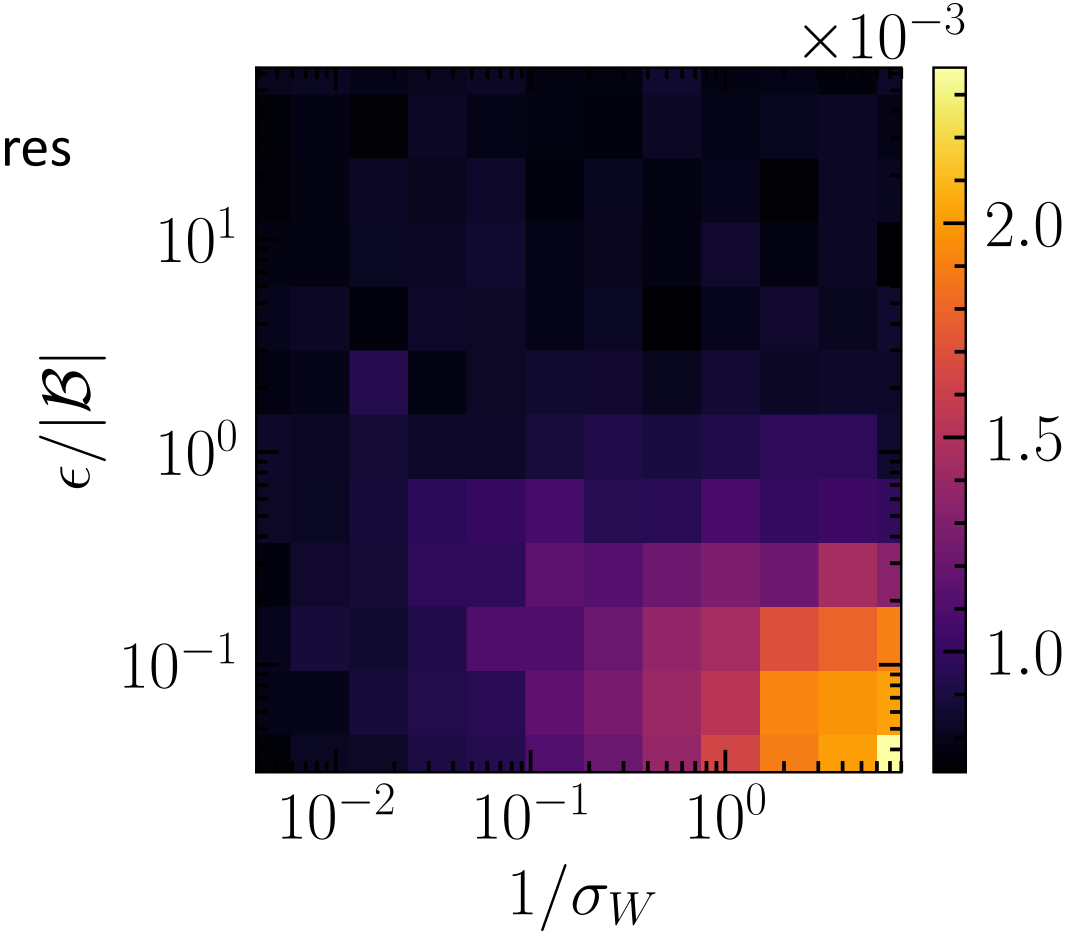
✗ large loss
large gradient

small loss, small gradient ✓

NN phase diagram: external field alignment

$$\mathcal{L}(\theta) = \frac{1}{2|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \sum_{i,j=1}^{n_{L-1}} J_{ij}\phi_{i\alpha}\phi_{j\alpha} - \frac{1}{|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \sum_{j=1}^{n_{L-1}} h_{j\alpha}\phi_{j\alpha}$$

alignment between features and external field



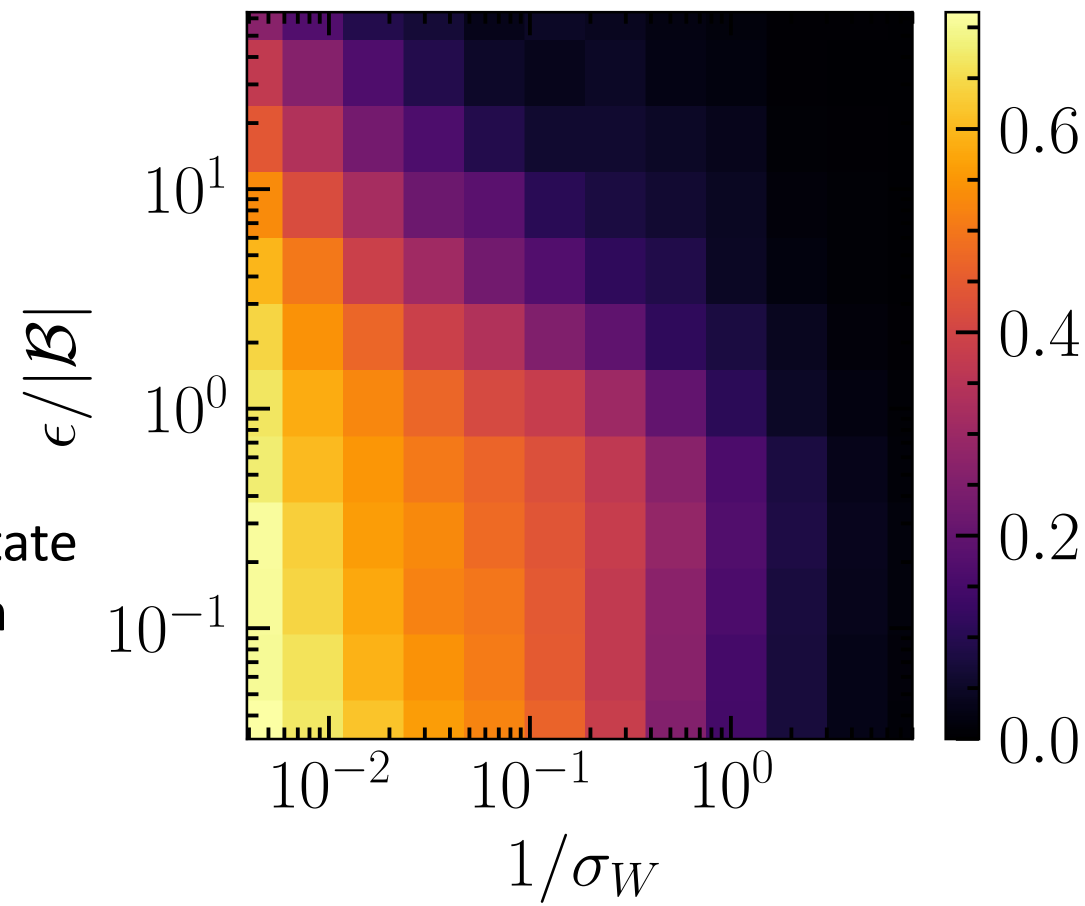
$$h \parallel \phi$$

most aligned in ferromagnetic phase

NN phase diagram: correlations in time

correlation in time between features

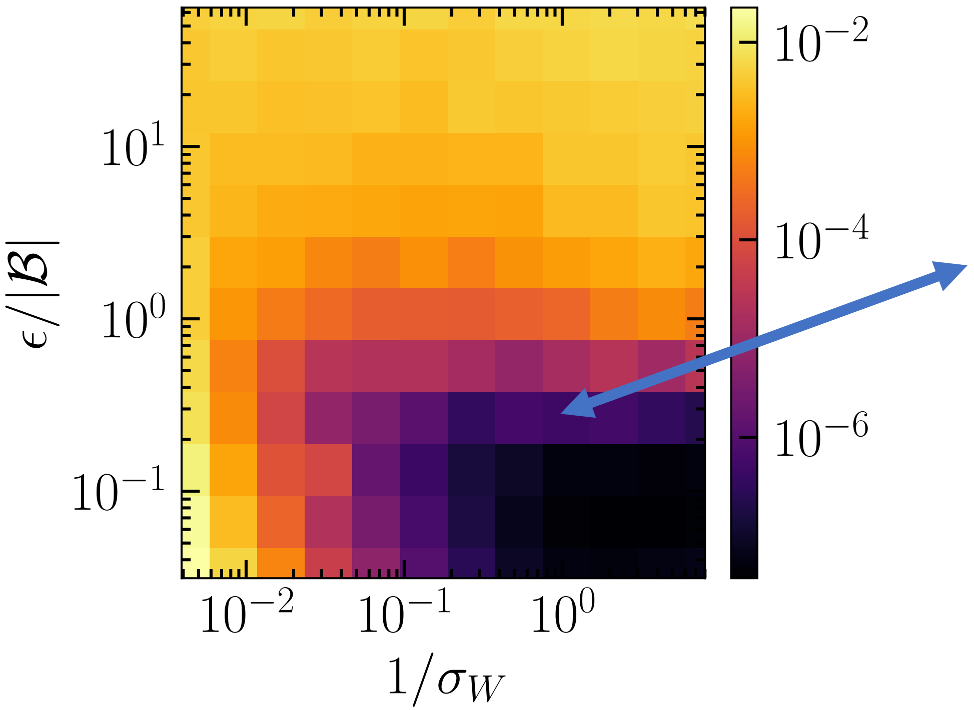
$$G(t,t') \equiv \mathbb{E}_{\mathcal{S}}[\phi(t)\phi(t')] = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{1}{|\mathcal{D}|}\sum_{\alpha\in\mathcal{D}}\frac{1}{n_{L-1}}\sum_{j=1}^{n_{L-1}}\phi^s_{j\alpha}(t)\phi^s_{j\alpha}(t')$$



memory of initial state mostly preserved in glassy phase

NN phase diagram and hyperparameters

- phase diagram in plane spanned by hyperparameters
- identification of ferro, paramagnetic and jammed or spin glass phases
- helps in understanding which choice of hyperparameters is preferred



choose as large $T = \epsilon/|\mathcal{B}|$ as possible with initial variance of weight matrices around 1

Theoretical physics analysis of neural networks

- treat NNs as a system with many fluctuating degrees of freedom
- hyperparameters are external parameters, like temperature and spin couplings
- quality and efficiency of learning can be understood by mapping phase diagram

why explore this?

- practical implications for ML practitioners: support in hyperparameter tuning
- theoretical physicists are/should not satisfied with a ‘black-box’ algorithm
- we can understand these algorithms by providing *physics input*

Summary lecture II: SGD, RMT, phase diagrams

- ML algorithms/neural networks are amenable to theoretical physics methodology
- stochastic weight matrix dynamics → universal features described by RMT
- eigenvalue repulsion, quantified by Wigner surmise and semi-circle observed in actual ML algorithms
- choice of hyperparameters can be guided by neural network phase diagrams

Open questions

- Dyson Brownian motion is present at "microscopic" level in weight matrix dynamics
- how does it manifest itself for more advanced architectures?
- is there universality beyond level repulsion (power law tails)?
- what are the practical implications? description of learning, algorithmic advances?