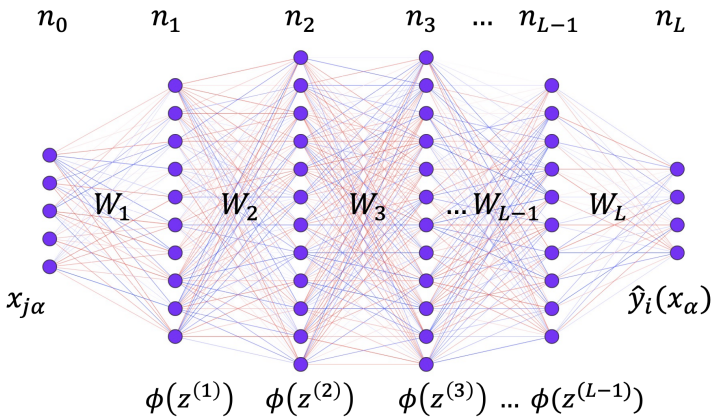
Exercises MITP2025 – ML and QFT – Gert Aarts

Weight matrices

In this exercise we will collect some observations about weight matrices. This exercise relies heavily on Refs. [1, 2] and I am indebted especially to Chanju Park for the joint work used in the preparation of this exercise.

1. Definition of a feed-forward neural network (NN).

   Given an input $\boldsymbol{x}_\alpha$ and associated output $\boldsymbol{y}_\alpha$, where α labels the data set $\mathcal{D}$, a neural network provides a function $\hat{\boldsymbol{y}}_\alpha = f(\boldsymbol{x}_\alpha)$. If the NN is well trained, $\hat{\boldsymbol{y}}_\alpha \simeq \boldsymbol{y}_\alpha$. The function is a combination of linear transformations, via weight matrices $W^{(l)}$, and nonlinear transformations, via activation functions $\phi(z)$, where z denotes the so-called pre-activation. The weight matrices are typically rectangular, of size $n_l \times n_{l-1}$.



   The feed-forward neural network in the figure is a graphical representation of the following expressions:

$$z_i^{(l+1)}(\boldsymbol{x}_\alpha) = \sum_{j=1}^{n_l} W_{ij}^{(l+1)} \phi\left(z_j^{(l)}(\boldsymbol{x}_\alpha)\right), \qquad z_i^{(1)}(\boldsymbol{x}_\alpha) = \sum_{j=1}^{n_0} W_{ij}^{(1)} x_{j\alpha}, \tag{28}$$

   where i, j are the indices of the vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$. The matrix multiplication is componentwise. The result on the final layer defines the neural network function

$$\hat{y}_i(\boldsymbol{x}_\alpha; \theta) \equiv z_i^{(L)}(\boldsymbol{x}_\alpha) = \sum_{j=1}^{n_{L-1}} W_{ij}^{(L)} \phi\left(z_j^{(L-1)}(\boldsymbol{x}_\alpha)\right). \tag{29}$$

   Here we follow closely the notation of Ref. [3]. We do not include a bias (but this can easily be done). Make sure that you understand the relation between these equations and the figure.

   Some jargon: the number of nodes in a layer is referred to as the width and the number of layers as the depth. Deep learning means many layers. There are N_θ learnable parameters in the network function, collectively denoted as

$$\theta = \{W^{(1)}, \ldots, W^{(L)}\}, \qquad N_\theta = \sum_{l=1}^{L} n_{l-1} n_l. \tag{30}$$

The combination of linear and non-linear transformations yields a "universal approximator". This is the content of the "universal approximation theorem".[2] Examples of activation functions are $\phi(z) = \tanh(z)$, the sigmoid $\sigma(z) = 1/(1+e^{-z})$,[3] and the rectified linear unit, $\mathrm{ReLU}(z) = \max(0, z)$.

To demonstrate that non-linear activation is essential, take ϕ as the identity, $\phi(z) = z$, and show that the NN then simply provides a linear function,

$$\hat{\boldsymbol{y}}_\alpha = \mathcal{W}\boldsymbol{x}_\alpha, \qquad \mathcal{W} = W^{(L)}W^{(L-1)}\dots W^{(1)}, \tag{31}$$

where $\mathcal{W}$ is the product of all weight matrices. Deep linear NNs are discussed in detail in Ref. [3].

2. Weight matrix initialisation.

   To initialise the NN, the weight matrices are given random elements, e.g., from a normal distribution, $W_{ij} \sim \mathcal{N}(0, \sigma_W^2/n_{l-1})$. One can then compute moments of the pre-activations recursively. Show that[4]

$$\mathbb{E}[z_i^{(l)}(\boldsymbol{x}_\alpha)] = 0, \qquad \mathbb{E}\left[z_i^{(l)}(\boldsymbol{x}_\alpha) z_j^{(l)}(\boldsymbol{x}_\beta)\right] = \delta_{ij}\frac{\sigma_W^2}{n_{l-1}}\sum_{k=1}^{n_{l-1}}\phi\left(z_k^{(l-1)}(\boldsymbol{x}_\alpha)\right)\phi\left(z_k^{(l-1)}(\boldsymbol{x}_\beta)\right). \tag{32}$$

   What is the expression for $l = 1$? Higher-order moments can also be considered; this is an exercise in Gaussian expectation values and Wick's theorem (and hence easy for field theorists). As in field theory, this simplifies in a large N expansion, where N is the width, see again Ref. [3] for details.

3. Singular value decomposition (SVD).

   Weight matrices are usually rectangular. One can do a SVD and write them as products of a left rotation, a rectangular diagonal matrix, and a right rotation. Consider an $M \times N$ matrix W, with for definiteness $N \leq M$. The SVD is

$$W = U\Xi V^T, \qquad UU^T = \mathbb{1}_{M\times M}, \qquad VV^T = \mathbb{1}_{N\times N}, \qquad \Xi = \mathrm{diag}_{M\times N}(\xi_1, \xi_2, \dots, \xi_N). \tag{33}$$

   The positive ξ_i's are the singular values. Express the eigenvalues of $X = W^TW$ and $\tilde{X} = WW^T$ in terms of the ξ_i's.

4. Marchenko-Pastur distribution at initialisation.

   It is useful to focus on the singular or eigenvalues, since the left/right rotations simply rotate the nodes. Since in physics we are used to symmetric real matrices, we consider from now on $X = W^TW$. We are interested in the distribution of eigenvalues, both at initialisation and after learning.

   Let's start with initialisation. If $W_{ij} \sim \mathcal{N}(0, \sigma^2/M)$, then the distribution of eigenvalues of X is given by the Marchenko-Pastur distribution,[5]

$$p_{\mathrm{MP}}(x) = \frac{1}{2\pi\sigma^2 r x}\sqrt{(x_+ - x)(x - x_-)}, \qquad x_- < x < x_+ \qquad x_\pm = \sigma^2\left(1 \pm \sqrt{r}\right)^2, \tag{34}$$

[2] https://en.wikipedia.org/wiki/Universal_approximation_theorem

[3] Yes, the sigmoid is indeed almost the Fermi-Dirac distribution.

[4] It is common to denote expectation values with $\mathbb{E}[.]$, rather than with brackets.

[5] https://en.wikipedia.org/wiki/Marchenko-Pastur_distribution

where $r = N/M \leq 1$. Plot this distribution for several values of r. What is the effect changing σ^2?

5. Distribution after training.

   Weight matrices are trained by minimising some loss function, e.g., the mean-squared error (MSE). Using the notation introduced above, this reads

$$\mathcal{L}(\theta) = \frac{1}{2|\mathcal{D}|} \sum_{\alpha=1}^{|\mathcal{D}|} \left(\boldsymbol{y}_\alpha - \hat{\boldsymbol{y}}_\alpha\right)^2, \tag{35}$$

   which is a function of all the weight matrices. It seems that for a well-trained NN the spectral densities contain a power decay, $p(x) \sim 1/x^\gamma$, for a wide range of learning tasks, see e.g. Refs. [4–6]. It is an open research question to derive the form of the eigenvalue distributions semi-analytically.

6. Universal description from random matrix theory (RMT): Coulomb gas

   As discussed in the lecture, when weight matrices are updated using stochastic gradient descent, they are subject to Dyson Brownian motion. Below we will derive the expressions for the Coulomb potential and for the Wigner surmise, which concerns the level spacing in spectra. We assume the standard assumptions in RMT hold [7], but this is not obvious.

   Consider N eigenvalues x_i $(i = 1, \ldots, N)$. The evolution equation for eigenvalue x_i is

$$\dot{x}_i = K_i + \sum_{j \neq i} \frac{g^2}{x_i - x_j} + \sqrt{2}\, g \eta_i, \qquad \eta_i \sim \mathcal{N}(0, 1), \tag{36}$$

   where it is assumed the drift can be derived from some potential $K_i = -\partial_{x_i} V(\{x_i\})$. The derivation of Eq. (36) essentially follows second-order perturbation theory in QM.

   Show that the corresponding FPE reads[6]

$$\partial_t p(\{x_i\}, t) = \sum_{i=1}^{N} \partial_{x_i} \left[\left(g^2 \partial_{x_i} - K_i^{(\mathrm{eff})} \right) \right] p(\{x_i\}, t). \tag{37}$$

   where the combined drift term is

$$K_i^{(\mathrm{eff})} = K_i + \sum_{j \neq i} \frac{g^2}{x_i - x_j}. \tag{38}$$

   Show that the stationary solution is given by the so-called Coulomb gas,

$$p_s(\{x_i\}) = \frac{1}{Z} \prod_{i<j} |x_i - x_j|\, e^{-V(\{x_i\})/g^2}, \qquad Z = \int dx_1 \ldots dx_N\, p_s(\{x_i\}). \tag{39}$$

   The term in the measure reflects the eigenvalue repulsion.

[6] We use the convention where time is rescaled with a factor of 2, see footnote 1.

7. Universal fluctuations: Wigner surmise

   To analyse the eigenvalue repulsion, one can study the behaviour of the spacing between adjacent eigenvalues, $S_i = x_{i+1} - x_i$. It turns out that its distribution $P(S)$ is universal; it is known as the Wigner surmise.

   To derive it we consider $N = 2$ eigenvalues and assume the potential has a simple quadratic form around degenerate minima,

$$V(x_1, x_2)/g^2 = \frac{1}{2\sigma^2}\left[(x_1 - \kappa)^2 + (x_2 - \kappa)^2\right]. \tag{40}$$

   Without repulsion, $\langle S\rangle = \langle x_2 - x_1\rangle \to 0$, due to the degeneracy.

   We assume the distribution is sufficiently peaked around κ, such that the integral boundaries can be taken as $\pm\infty$. The partition function is then

$$Z = \frac{1}{N_0}\int dx_1 dx_2\, |x_1 - x_2|\, e^{-\left(x_1^2+x_2^2\right)/(2\sigma^2)}, \qquad N_0 = 4\sqrt{\pi}\sigma^3. \tag{41}$$

   Change variables to $x_{1,2} = x \pm S/2$ to show that the Wigner surmise $P(S)$ reads

$$P(S) = \frac{S}{2\sigma^2}e^{-S^2/(4\sigma^2)}, \qquad Z = \int_0^\infty dS\, P(S). \tag{42}$$

   Show that the mean level spacing is

$$\langle S\rangle = \int_0^\infty dS\, SP(S) = \sqrt{\pi}\sigma. \tag{43}$$

   In terms of $s = S/\langle S\rangle$ the surmise is parameter-free,

$$P(s) = \frac{\pi}{2}se^{-\pi s^2/4}. \tag{44}$$

   This universal behaviour has indeed been observed in NNs, see e.g. Refs. [1, 5, 6].

References

[1] G. Aarts, B. Lucini and C. Park, "Stochastic weight matrix dynamics during learning and Dyson Brownian motion," Phys. Rev. E **111** (2025) no.1, 015303 [2407.16427 [cond-mat.dis-nn]].

[2] C. Park, B. Lucini and G. Aarts, "Phase diagram and eigenvalue dynamics of stochastic gradient descent in neural networks," in preparation.

[3] D. A. Roberts, S. Yaida and B. Hanin, "The Principles of Deep Learning Theory," Cambridge University Press, 2022 [2106.10165 [cs.LG]].

[4] C. H. Martin and M. W. Mahoney, "Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning," Journal of Machine Learning Research **22** (2021) no.165, 1–73 [1810.01075 [cs.LG]].

[5] N. P. Baskerville, D. Granziol and J P. Keating, "Appearance of Random Matrix Theory in Deep Learning," Physica A **590** (2022) 126742 [2102.06740 [cs.LG]].

[6] G. Aarts, O. Hajizadeh, B. Lucini and C. Park, "Dyson Brownian motion and random matrix dynamics of weight matrices during learning," in *NeurIPS* 2024 workshop *Machine Learning and the Physical Sciences* [2411.13512 [cond-mat.dis-nn]].

[7] M.L. Mehta, *Random Matrices*, Academic Press, New York, 3rd ed. (2004).