# Implicit Regularization of SGD via Thermophoresis

**Mingwei Wei**
Department of Physics and Astronomy
Northwestern University
Evanston, IL 60208
m.wei@u.northwestern.edu

**David J Schwab**
The Graduate Center
The City University of New York
New York, NY 10016
dschwab@gc.cuny.edu

## Abstract

A central ingredient in the impressive predictive performance of deep neural networks is optimization via stochastic gradient descent (SGD). While some theoretical progress has been made, the effect of SGD in neural networks is still unclear, especially during the early phase of training. Here we generalize the theory of *thermophoresis* from statistical mechanics and show that there exists an effective entropic force from SGD that pushes to reduce the gradient variance. We study this effect in detail in a simple two-layer model, where the thermophoretic force functions to decreases the weight norm and activation rate of the units. The strength of this effect is proportional to squared learning rate and inverse batch size, and is more effective during the early phase of training when the model's predictions are poor. Lastly we test our quantitative predictions with experiments on various models and datasets.

## 1 Introduction

Deep neural networks have achieved remarkable success in the past decade on tasks that were out of reach prior to the era of deep learning. Yet fundamental questions remain regarding the strong performance of over-parameterized models and optimization schemes that typically involve only first-order information, such as stochastic gradient descent (SGD) and its variants.

In particular, optimization via SGD is known in many cases to result in models that generalize better than those trained with full-batch optimization. To explain this, much work has focused on how SGD navigates towards so-called flat minima, which tend to generalize better than sharp minima [14, 17]. This has been argued by nonvacuous PAC-Bayes bounds [5] and Bayesian evidence [26]. More recently, [30] discuss how optimization via SGD pushes models to flatter regions within a minimal valley by decreasing the trace of the Hessian.

However, these perspectives apply to models towards the end of training, whereas it is known that proper treatment of hyperparameters during the early phase is vital. In particular, when training a deep network one typically starts with a large learning rate and small batch size if possible. After training has progressed, the learning rate is annealed and decreased so that the model can be further trained to better fit the training set [18, 25, 13, 12, 33, 29]. Crucially, using a small learning rate during the first phase of training usually leads to poor generalization and also results in large gradient variance in practice [16, 7].

However, limited theoretical work has been done to understand the effect of SGD on the early phase of training. [16] argues for the existence of a "break-even" point on an SGD trajectory. However their analysis focuses only on the leading eigenvalue of the Hessian spectrum and requires the strong assumption that the loss function in the leading eigen-subspace is quadratic. Meanwhile [22] studied the simple setting of two-layer neural networks. This work relies heavily on the existence of these two

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

distinct types of features in the data and the specific network architecture. Moreover, their analysis focuses mainly on learning rate instead of the effect of SGD.

In this paper, we study the dynamics of model parameter motion during SGD training by borrowing and generalizing the theory of thermophoresis from physics. With this framework, we show that during SGD optimization, especially during the early phase of training, the activation rate of hidden nodes is reduced as is the growth of parameter weight norm. This effect is proportional to squared learning rate and inverse batch size. Thus, thermophoresis in deep learning acts as an implicit regularization that may improve the model's ability to generalize.

## 2 Thermophoresis in General

In this section, we study thermophoresis in a generalized inhomogeneous and anisotropic random walk. A brief discussion of thermophoresis theory in physics can be found in Appendix A.3. We first define a kind of generalized random walk that has evolution equations for a particle state with coordinate $\mathbf{q} = \{q_i\}_{i=1,\dots,n}$ as

$$\mathbf{q}_{t+1} = \mathbf{q}_t - \eta\gamma\mathbf{f}(\mathbf{q}_t, \xi) \ , \tag{1}$$

where $\mathbf{f}$ is a vector function, $\gamma$ and $\xi$ are random variables, and $\eta$ is a small number controlling the step size. Notice that this is a generalized inhomogeneous random walk for the particle. Before further analysis, it is noted that the evolution equations 1 is similar to SGD updates in machine learning and we will show this in the next section.

To isolate the effect of thermophoresis, we assume the random walk is unbiased, in which case

$$P(\gamma\mathbf{f}(\mathbf{q}, \xi) = \mathbf{a}) = P(\gamma\mathbf{f}(\mathbf{q}, \xi) = -\mathbf{a}), \tag{2}$$

for an arbitrary vector $\mathbf{a}$. Thus there is no explicit force exerted on the particle. We also denote the particle mass density as $\rho(\mathbf{q})$ and

$$g_i(\mathbf{q}) := \sqrt{\int \gamma^2 f_i^2(\mathbf{q}, \xi) d\mu(\gamma, \xi)}, \tag{3}$$

so that $\eta g_i(\mathbf{q})$ is the standard deviation of the random walk in the $i$th direction.

From a position $\mathbf{q}$, we consider a subset of coordinate indices, $U \subseteq \{1, \dots, n\}$, wherein

$$(f_i(\mathbf{q}, x)) = (f_j(\mathbf{q}, x)) \ \text{ and } \ \partial_i g_j(\mathbf{q}) \geq 0 \tag{4}$$

for all $i, j \in U$.

In order to study the dynamics of the particle and its density function, we focus on the mass flow induced by the inhomogeneous random walk. We will show that there is always a flow from regions with larger $g_i(\mathbf{q})$ to those with smaller $g_i(\mathbf{q})$ for $i \in U$, which is a generalization of thermophoresis in physics.

Since $\eta \ll 1$, the movement of the particle will have a mean free path of $g_i(\mathbf{q})$ in $i$th direction. Therefore the random walk equation 1 becomes

$$q_i = q_i - \eta g_i(\mathbf{q})\zeta_i, \tag{5}$$

where $i = 1, \dots, n$ and $\zeta_i$ is a binary random variable with $P(\zeta_i = -1) = P(\zeta_i = 1) = 0.5$. Moreover, from Eq. 4, we also have that $\zeta_i = \zeta_j$ for all $i$ and $j \in U$.

It can be shown that there exists a mass flow[1] is

$$J = -\eta^2 \sqrt{\sum_{i \in U} g_i^2(\mathbf{q})} \sum_{i \in U} g_i(\mathbf{q}) \partial_i \rho(\mathbf{q}) - \eta^2 \frac{\sum_{i,j \in U} g_i(\mathbf{q}) g_j(\mathbf{q}) \partial_j g_i(\mathbf{q})}{\sqrt{\sum_{i \in U} g_i^2(\mathbf{q})}} \rho(\mathbf{q}) + O(\eta^3), \tag{6}$$

where the derivation can be found in Appendix A.4. This can be understood as describe in Diagram 2.

Notice that the mass flow consists of two main terms. The first one represents the diffusion and the second term corresponds to our goal in this section, which results in thermophoresis. By definition of

[1] More specifically, flow density.

function $g$ and Eq. 4, we know that the coefficient of thermophoresis (Soret coefficient), which is defined as

$$\begin{aligned} & c - \eta^2 \frac{\sum_{i,j \in U} g_i(\mathbf{q}) g_j(\mathbf{q}) \partial_j g_i(\mathbf{q})}{2\sqrt{\sum_{i \in U} g_i^2(\mathbf{q})}} && (7) \\ & \geq 0, && (8) \end{aligned}$$

is negative. This means that there is an effective force exerted on particle at $\mathbf{q}$ towards the smaller variance regime (by analogy, the colder area). The coefficient is proportional to $\eta^2$.

## 3 Thermophoresis in Deep Learning

To study the physics behind SGD optimization in detail, we consider the simple setting of one-hidden layer neural networks. The network is a function $f : \mathbb{R}^M \to \mathbb{R}$ parameterized as follows:

$$f(\mathbf{x}; \mathbf{V}, \mathbf{W}, \mathbf{b}) = \mathbf{V}\sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \ .$$

We also write $f(x)$ for simplicity. The network has a scalar output, which is widely used in regression and binary classification. $x$ is the network input with dimension $M$, $\mathbf{W}$ and $\mathbf{b}$ are the weights and biases in the first layer with dimension $N \times M$ and $N$ respectively, where $N$ is the number of hidden nodes in the hidden layer, and $\sigma$ is the ReLU activation function defined as $\sigma(a) = \max(0, \mathrm{a})$.

We consider use binary cross entropy as the loss function, so the mini-batch gradient becomes

$$\nabla \mathcal{L}_B(\mathbf{V}, \mathbf{W}, \mathbf{b}) = \frac{1}{|B|} \sum_{i=1}^{|B|} (p_i - y_i) \nabla f(\mathbf{x}_i). \tag{9}$$

The detailed analysis of training dynamics can be found in Appendix A.6. There we show that the parameters in this model and their dynamics approximately satisfy the criteria of the previous section, and that the biases are pushed negative and $V^2$ is suppressed during training, the effects of both of which are proportional to squared learning rate $\eta^2$ and inverse batch size $1/|B|$.

It is shown in Appendix A.7 that there exists an effective force that pushes to decrease the model's activation rate, defined in equation 25, and reduces the weight norm of the second layer. The strength of this force scales as

$$F \propto \frac{\eta^2}{|B|} \ . \tag{10}$$

In [21], Theorem 4.1 presents a linear relation between learning rate and training iterations for a target training error $\epsilon$ and small learning rate. This implies that if one uses a learning rate $k$ times larger, the model will require $k$ times fewer optimization steps for the same training performance. Together with our results, this implies the following: for the same model and initialization, comparing two optimization schemes with $\eta_1 \leq \eta_2$ each achieving a given training error, the activation rate for scheme 1 will be at least as large as that for scheme 2, i.e. $\sigma_1 \geq \sigma_2$. Similarly, denoting the weight norm for scheme 1(2) by $v_1(v_2)$, we have that $v_1 \geq v_2$.

Model sparsity can mean two different things: sparsity of the weights, and frequency with which units are activated, called the activation rate. Intuitively, a sparser model has a smaller capacity [2, 19, 1, 23]. One advantage of sparsity is for model pruning, where model parameters or units can be removed systematically in order to obtain an effective model with smaller size [11, 10, 32]. Furthermore, model pruning has been shown to improve generalization [8, 9]. Therefore it may be expected that a small activation rate correlates with generalization. Moreover, in Appendix A.8, we construct a upper bound of Hessian norm which depends monotonically on activation rate and weight norm. This also sheds light on the connection between sparsity, weight norm, and generalization.

Our theory can also be generalized beyond two-layer models. We have shown that there exists an effective force in deep neural networks from SGD that reduces the gradient variance and have quantitatively characterized it.

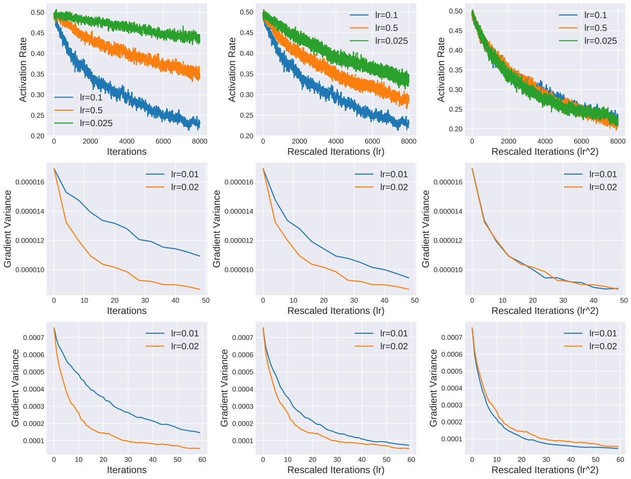

Figure 1: All rows include rescaled x-axes as described in the main text. Top Row: Plots of activation rate as a function of (rescaled) training iterations with different learning rates. The model is a two-layer fully-connected network with 100 hidden units. Training data is drawn from a normal distribution. Middle Row: Plots of average gradient variance as a function of (rescaled) training iterations with different learning rates in 6-layer fully-connected neural networks. Training data is drawn from normal distribution. Bottom Row: Same as middle row but for 6-layer convolutional neural networks trained on Fashion-MNIST.

## 4 Experiments and Conclusion

The essential result from the previous section is that there exists an effective force from SGD, analogous to thermophoresis, that pushes to decrease the gradient variance, and in one-hidden-layer neural networks decreases the model's activation rate and reduces the weight norm of the second layer. The strength of the force is proportional to squared learning rate and inverse batch size. In this section, we present experiments to test these results.

First we consider a one hidden layer model with input dimension $100$ and $100$ hidden units. The input data, $\mathbf{x}$, is distributed as $\mathcal{N}(0, \mathrm{I})$ where I is the identity matrix, and the label is randomly chosen from $\{0, 1\}$. Batch size is set to $1$ and the learning rate is varied from $0.025$ to $0.1$. We calculate the activation rate and L2 norm of the vector $\mathbf{V}$ after each training iteration. The result for activation rate is shown in the first row of Fig. 1. The leftmost plot shows activation rate as a function of true iteration on the x-axis, and we see that activation rate decreases during training, and the decreasing is more rapid with larger learning rate. In the middle plot we rescale the x-axis by a factor proportional to learning rate $\eta^2$. This rescaling factor is to offset the movement difference due to learning rate difference. It is clear that even after this rescaling, we still observe that larger learning rates decrease the activation rate faster. Finally, on the rightmost plot we rescale the x-axis with a factor proportional to squared learning rate $\eta^2$. We see that all trajectories now overlap, which matches our prediction in the previous section that decreasing rate is proportional to $\eta^2$.

We next test our results for deep neural networks beyond the two-layer model. Instead of activation rate and weight norm, we plot the gradient variance as predicted by our theory. Network architectures are 6-layer fully-connected with hidden layer sizes of 100 and 6-layer convolutional with 10 channels with kernel size of 5*5 and stride 1 except the last fully-connected layer output. The results are shown in the second row of Fig. 1 and the third row of Fig. 1, respectively.

To summarize, in this paper we generalized the theory of *thermophoresis*, showing that there exists an effective thermophoretic force from SGD $\propto \eta^2/|B|$ that pushes to reduce the gradient variance, and is more effective during the early phase of training when the model's predictions are poor.

[2] For example, if raw iteration number for $\eta = 0.05$ is $1000$ and rescaled iteration number is also $1000$, the rescaled iteration number for $\eta = 0.1$ is $1000$ then its true iteration number is $500$.

# 5 Acknowledgements

DJS was partially supported by the Simons Foundation as an Investigator in the MMLS, the NSF through the Center for the Physics of Biological Function (PHY-1734030), and the NIH (R01EB026943-01).

## References

[1] A. Aghasi, A. Abdi, N. Nguyen, and J. Romberg. Net-trim: Convex pruning of deep neural networks with performance guarantee. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[2] P. Bizopoulos and D. Koutsouris. Sparsely activated networks. *arXiv preprint arXiv:1907.06592*, 2020.

[3] J. Chipman. The soret effect. *Journal of the American Chemical Society*, 48(10):2577–2589, 1926.

[4] S. de Groot and P. Mazur. *Non-equlibrium Thermodynamics*. North Holland Publishing, 1962.

[5] G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *UAI*, 2017.

[6] E. D. Eastman. Thermodynamics of non-isothermal systems. *Journal of the American Chemical Society*, 48(6):1482–1493, 1926.

[7] F. Faghri, D. Duvenaud, D. J. Fleet, and J. Ba. A study of gradient variance in deep learning. *arXiv preprint arXiv:2007.04532*, 2020.

[8] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[9] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin. The lottery ticket hypothesis at scale. *arXiv preprint arXiv:1903.01611*, 2020.

[10] T. Gale, E. Elsen, and S. Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

[11] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[15] J. Janek, C. Korte, and A. B. Lidiard. *Thermodiffusion in Ionic Solids —Model Experiments and Theory*, pages 146–183. Springer Berlin Heidelberg, 2002.

[16] S. Jastrzebski, M. Szymczak, S. Fort, D. Arpit, J. Tabor, K. Cho, and K. Geras. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[17] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[19] M. Kurtz, J. Kopinsky, R. Gelashvili, A. Matveev, J. Carr, M. Goin, W. Leiserson, S. Moore, N. Shavit, and D. Alistarh. Inducing and exploiting activation sparsity for fast neural network inference. In *International Conference on Machine Learning (ICML)*, 2020.

[20] W. Köhler and K. I. Morozov. The soret effect in liquid mixtures – a review. *Journal of Non-Equilibrium Thermodynamics*, 41, 2016.

[21] Y. Li and Y. Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[22] Y. Li, C. Wei, and T. Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *arXiv preprint arXiv:1907.04595*, 2020.

[23] J. Lin, Y. Rao, J. Lu, and J. Zhou. Runtime neural pruning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[24] C. Ludwig. *Sitz. Ber. Akad. Wiss. Wien*, 1859.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[26] S. L. Smith and Q. V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *ICLR*, 2018.

[27] C. Soret. *Arch Sci Phys Nat*, 1897.

[28] H. Tyrell and R. Colledge. Thermal diffusion potentials and the soret effect. *Nature*, 173: 264–265, 1954.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[30] M. Wei and D. J. Schwab. How noise affects the hessian spectrum in overparameterized neural networks. *arXiv preprint arXiv:1910.00195*, 2019.

[31] A. Wurger. Is soret equilibrium a non-equilibrium effect? *arXiv preprint arXiv:1401.7546*, 2014.

[32] J. L. Yoojin Choi, Mostafa El-Khamy. Jointly sparse convolutional neural networks in dual spatial-winograd domains. *arXiv preprint arXiv:1902.08192*, 2019.

[33] Y. You, I. Gitman, and B. Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888,*, 2017.

# A Appendix

## A.1 Diagram

Diagram of thermophoresis flow calculation.

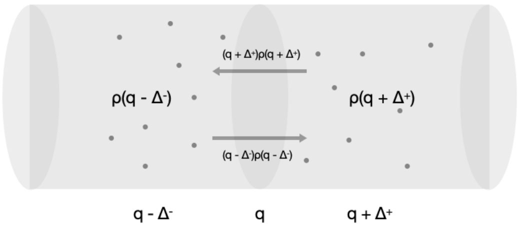

Figure 2: Diagram of mass flow in a generalized inhomogeneous random walk used in the derivation of the Soret coefficient.

## A.2 Proof of Property A.1

*Proof.* By definition, we have

$$\sigma(\sum_{j=1}^{M} W_j x_j + b_1) \leq \sigma(\sum_{j=1}^{M} W_j x_j + b_2) \ , \tag{11}$$

$$\sigma'(\sum_{j=1}^{M} W_j x_j + b_1) \leq \sigma'(\sum_{j=1}^{M} W_j x_j + b_2) \ . \tag{12}$$

For $h_v$,

$$\begin{aligned} h_v(V_1, \mathbf{W}, b_1) &= \mathbb{E}_x g_v^2(\mathbf{x}, V_1, \mathbf{W}, b_1) \ , \\ &= \mathbb{E}_x \sigma^2(\sum_{j=1}^{M} W_j x_j + b_1) \ , \\ &\leq \mathbb{E}_x \sigma^2(\sum_{j=1}^{M} W_j x_j + b_2) \ , \\ &= h_v(V_2, \mathbf{W}, b_2) \ . \end{aligned}$$

Similarly, we have

$$\begin{aligned} h_{w_i}(V_1, \mathbf{W}, b_1) &= \mathbb{E}_x V_1^2 x_i^2 \sigma'(\sum_{k=1}^{M} W_k x_k + b_1) \ , \\ &\leq \mathbb{E}_x V_2^2 x_i^2 \sigma'(\sum_{k=1}^{M} W_k x_k + b_1) \ , \\ &\leq \mathbb{E}_x V_2^2 x_i^2 \sigma'(\sum_{k=1}^{M} W_k x_k + b_2) \ , \\ &= h_{w_i}(V_2, \mathbf{W}, b_2) \ . \end{aligned}$$

Clearly the inequality also holds for $h_b$. □

### A.3 Thermophoresis in Physics

Thermophoresis, also known as the Soret effect, describes particle mass flow in response to both diffusion and temperature gradient. The effect was first discovered in electrolyte solutions [24, 27, 3]. However it was discovered in other systems such as gases, colloids, and biological fluids and solid [15, 20].

Thermophoresis typically refers to particle diffusion in a continuum with a temperature gradient. The non-uniform steady-state density $\rho$ is given by the "Soret Equilibrium" [6, 28, 31],

$$\nabla\rho + \rho S_T \nabla T = 0 \ , \tag{13}$$

where $T$ is temperature and $S_T$ is called the Soret coefficient.

In [4], mass flow was calculated by non-equilibrium theory. They considered two types of processes for entropy balance equation. The reversible process stands for the entropy transfer and the irreversible process corresponds to the entropy production, or dissipation. The resulting mass flow induced by diffusion and temperature gradient is found to be

$$J = -D\nabla\rho - \rho D_T \nabla T \ , \tag{14}$$

where $D$ is the Einstein diffusion coefficient and $D_T$ is defined as thermal diffusion coefficient. Comparing the steady state in 13 and setting the flow to be zero, the Soret coefficient is simply

$$S_T = \frac{D_T}{D} \ . \tag{15}$$

The Soret coefficient can be calculated from molecular interaction potentials based on specific molecular models [31].

### A.4 Derivation of Eq. 6

We will show that the flow projecting on the subspace $U$ is always toward negative $g_i(\mathbf{q})$. Notice that although $U$ can be multi-dimensional, the degree of freedom of the particle dynamics is 1 within $U$ due to the sharing of the $\zeta$s, and therefore the mass flow projecting on it is also 1-dimensional. For each $i \in U$, we define the average flow in this dimension to be the mass that enters $q_i$ from $q_i^-$ minus the mass from the opposite direction $q_i^+$. From Eq. 5 and the assumption that $\eta \ll 0$, only mass close to $q_i$ will move across $q_i$ at each step. We let the farthest mass that is likely to flow across $q_i$ to be $q_i + \Delta_i^+$ and $q_i - \Delta_i^-$, where $\Delta_i^+$ and $\Delta_i^+$ are positive. By definition, we have $\Delta_i^+$ and $\Delta_i^-$ satisfy $\Delta_i^+ = \eta g_i(\mathbf{q} + \mathbf{\Delta}^+)$ and $\Delta_i^- = \eta g_i(\mathbf{q} - \mathbf{\Delta}^-)$, respectively. Notice that if the random walk were homogeneous, we would have $\Delta_i^+ = \Delta_i^-$. In our inhomogeneous case, we have $\Delta_i^+ \sim \Delta_i^- \sim \eta g_i(\mathbf{q})$ up to the first leading order of $\eta$, and the next to leading order will be calculated in order to compute the difference between $\Delta_i^+$and $\Delta_i^-$.

Now we are ready to calculate the mass flow through $\mathbf{q}$. The mass flow projecting onto the subspace $U$ is calculated by the mass through $\mathbf{q}$ from $\mathbf{q} + \mathbf{\Delta}^+$ minus the mass from $\mathbf{q} - \mathbf{\Delta}^-$ where $\Delta_i^+$ and $\Delta_i^-$ are as above if $i \in U$ and $\Delta_i^+ = \Delta_i^- = 0$ otherwise. By definition of $\Delta_i^+$ and $\Delta_i^-$ we also have

$$\begin{aligned}
\Delta_i^+ - \Delta_i^- &= \eta g_i(\mathbf{q} + \mathbf{\Delta}^+) - \eta g_i(\mathbf{q} - \mathbf{\Delta}^-) \ , && (16) \\
&= \eta \sum_{j \in U}^{n} (\Delta_j^+ + \Delta_j^-)\partial_j g_i(\mathbf{q}) + O(\eta\Delta^2) \ , && (17) \\
&= 2\eta^2 \sum_{j \in U} g_j(\mathbf{q})\partial_j g_i(\mathbf{q}) + O(\eta^3) \ . && (18)
\end{aligned}$$

Therefore the flow[3] is

$$
\begin{aligned}
J =& -\frac{1}{2}|\mathbf{\Delta}^+|\rho(\mathbf{q}+\mathbf{\Delta}^+) + \frac{1}{2}|\mathbf{\Delta}^-|\rho(\mathbf{q}-\mathbf{\Delta}^-) \ , \\
=& \frac{1}{2}|\mathbf{\Delta}^+|[\rho(\mathbf{q}-\mathbf{\Delta}^-) - \rho(\mathbf{q}+\mathbf{\Delta}^+)] + \frac{1}{2}(|\mathbf{\Delta}^-| - |\mathbf{\Delta}^+|)\rho(\mathbf{q}-\mathbf{\Delta}^-) \ , \\
=& -\frac{1}{2}|\mathbf{\Delta}^+|(|\mathbf{\Delta}^+ + \mathbf{\Delta}^-|)\frac{\rho(\mathbf{q}+\mathbf{\Delta}^+) - \rho(\mathbf{q}-\mathbf{\Delta}^-)}{|\mathbf{\Delta}^+ + \mathbf{\Delta}^-|} \\
& - \frac{1}{2}\frac{(|\mathbf{\Delta}^+|^2 - |\mathbf{\Delta}^-|^2)}{|\mathbf{\Delta}^+| + |\mathbf{\Delta}^-|}\rho(\mathbf{q}-\mathbf{\Delta}^-) \ , \\
\approx& -\frac{1}{2}|\mathbf{\Delta}^+|(\mathbf{\Delta}^+ + \mathbf{\Delta}^-)\nabla\rho(\mathbf{q}) - \frac{1}{2}\frac{\sum_{i\in U}(\Delta_i^+ + \Delta_i^-)(\Delta_i^+ - \Delta_i^-)}{|\mathbf{\Delta}^+| + |\mathbf{\Delta}^-|}\rho(\mathbf{q}-\mathbf{\Delta}^-) \ , \\
=& -\eta^2\sqrt{\sum_{i\in U} g_i^2(\mathbf{q})}\sum_{i\in U} g_i(\mathbf{q})\partial_i\rho(\mathbf{q}) - \eta^2\frac{\sum_{i,j\in U} g_i(\mathbf{q})g_j(\mathbf{q})\partial_j g_i(\mathbf{q})}{\sqrt{\sum_{i\in U} g_i^2(\mathbf{q})}}\rho(\mathbf{q}) + O(\eta^3) \ .
\end{aligned}
$$

where the derivation can be found in Appendix A.4. This can be understood as describe in Diagram 2.

Notice that the mass flow consists of two main terms. The first one represents the diffusion and the second term corresponds to our goal in this section, which results in thermophoresis. By definition of function $g$ and Eq. 4, we know that the coefficient of thermophoresis (Soret coefficient), which is defined as

$$
\begin{aligned}
& c - \eta^2\frac{\sum_{i,j\in U} g_i(\mathbf{q})g_j(\mathbf{q})\partial_j g_i(\mathbf{q})}{2\sqrt{\sum_{i\in U} g_i^2(\mathbf{q})}} && (19) \\
& \geq 0, && (20)
\end{aligned}
$$

is negative. This means that there is an effective force exerted on particle at $\mathbf{q}$ towards the smaller variance regime (by analogy, the colder area). The coefficient is proportional to $\eta^2$.

### A.5 Sanity Check of Generalized Theory

If $|U| = 1$ and $g_i(\mathbf{q}) = g_i(q_i)$, the model will reduce to aforementioned physics model and the Soret coefficient reduces to

$$
\begin{aligned}
c =& \frac{\eta^2}{2}g(q)g'(q) \ , && (21) \\
=& [(\frac{\eta g(q)}{2})^2]' \ , && (22) \\
\approx& \nabla T \ , && (23)
\end{aligned}
$$

where $T$ is the effective temperature in the model. This result is consistent with thermophoresis model in physics.

### A.6 Model and its Training

The dataset is drawn i.i.d. from the data distribution, $\{(\mathbf{x}, y)|(\mathbf{x}, y) \sim \mathcal{D}(\mathbf{x}, y)\}$. In this paper we consider two cases, where either $x_i \geq 0$[4] or $x_i \sim \mathcal{N}(0, 1)$[5]. Here $y \in \mathbb{Y}$ and we denote the marginal distribution of $y$ as $\mathcal{D}_Y$. Finally, we have the loss function $L : \mathbb{R} \times \mathbb{Y} \to \mathbb{R}^+$.

We consider optimization via SGD, where the gradient of the loss on a batch of size $|B|$ is given by

$$\nabla\mathcal{L}_B(\mathbf{V}, \mathbf{W}, \mathbf{b}) = \frac{1}{|B|}\sum_{i=1}^{|B|}\nabla_f L(f(\mathbf{x}_i), y_i)\nabla f(\mathbf{x}_i). \tag{24}$$

[3] More specifically, flow density.

[4] Usually in convolutional neural networks or intermediate layers.

[5] Often found when the data are normalized.

In our two-layer model, we have

$$\nabla_{V_i} f(\mathbf{x}) = \sigma(\sum_{j=1}^{M} W_{ij} x_j + b_i),$$

$$\nabla_{W_{ij}} f(\mathbf{x}) = V_i x_j \sigma'(\sum_{k=1}^{M} W_{ik} x_k + b_i),$$

$$\nabla_{b_i} f(\mathbf{x}) = V_i \sigma'(\sum_{k=1}^{M} W_{ik} x_k + b_i).$$

For an input vector $\mathbf{x}$, we call the hidden node $i$ *activated* when $\sigma'(\sum_{k=1}^{M} W_{ik} x_k + b_i) = 1$, or equivalently $W_{ik} x_k + b_i > 0$. We thus define the *activation rate* of the network to be

$$\overline{\sigma'} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_x \sigma'(\sum_{k=1}^{M} W_{ik} x_k + b_i) \ . \tag{25}$$

This is an important concept to which we will return.

Henceforth, we drop the index $i$, since the dynamical equations are invariant with respect to node index, and write $V V_i$, $W_j W_{ij}$ and $b b_i$ by abuse of notation. We also denote

$$h_v(V, \mathbf{W}, b) \mathbb{E}_x[\nabla_V f(\mathbf{x})]^2,$$
$$h_{\mathbf{w}}(V, \mathbf{W}, b) \mathbb{E}_x[\nabla_{\mathbf{W}} f(\mathbf{x})]^2,$$
$$h_b(V, \mathbf{W}, b) \mathbb{E}_x[\nabla_b f(\mathbf{x})]^2,$$

where $\mathbb{E}_x$ denotes average over input $\mathbf{x}$. We have the following property for the functions $h$:

**Property A.1.** *Given* $\mathbf{W}$*, if* $V_1^2 \leq V_2^2$ *and* $b_1 \leq b_2$*, we have*

$$h_v(V_1, \mathbf{W}, b_1) \leq h_v(V_2, \mathbf{W}, b_2),$$
$$h_{\mathbf{w}}(V_1, \mathbf{W}, b_1) \leq h_{\mathbf{w}}(V_2, \mathbf{W}, b_2),$$
$$h_b(V_1, \mathbf{W}, b_1) \leq h_b(V_2, \mathbf{W}, b_2),$$
$$\overline{\sigma'}(V_1, \mathbf{W}, b_1) \leq \overline{\sigma'}(V_2, \mathbf{W}, b_2).$$

*Here we define* $\mathbf{a} \leq \mathbf{b}$ *as* $\min(\mathbf{b} - \mathbf{a}) \geq 0$*.*

It is straightforward to see the following:

**Property A.2.** *When the case of* $x_i \geq 0$ *is considered, if* $V_1^2 \leq V_2^2$*,* $\mathbf{W}_1 \leq \mathbf{W}_2$ *and* $b_1 \leq b_2$*, we have*

$$h_v(V_1, \mathbf{W}_1, b_1) \leq h_v(V_2, \mathbf{W}_2, b_2),$$
$$h_{\mathbf{w}}(V_1, \mathbf{W}_2, b_1) \leq h_{\mathbf{w}}(V_2, \mathbf{W}, b_2).$$
$$h_b(V_1, \mathbf{W}_2, b_1) \leq h_b(V_2, \mathbf{W}, b_2),$$
$$\overline{\sigma'}(V_1, \mathbf{W}, b_1) \leq \overline{\sigma'}(V_2, \mathbf{W}, b_2).$$

In our analysis, we focus for simplicity on binary classification tasks, where the loss is typically binary cross-entropy: $L(f, y) = y \ln p(f) + (1 - y) \ln(1 - p(f))$ and $p(f) = 1/(1 + \exp(f))$. We thus have

$$\nabla_f L(f, y) = p(f) - y. \tag{26}$$

Substituting into Eq. 24, the mini-batch gradient becomes

$$\nabla \mathcal{L}_B(\mathbf{V}, \mathbf{W}, \mathbf{b}) = \frac{1}{|B|} \sum_{i=1}^{|B|} (p_i - y_i) \nabla f(\mathbf{x}_i). \tag{27}$$

Our results, however, can be generalized to arbitrary loss.

### A.7 Derivation of Thermophoresis Flow in Deep Learning

The gradient that dominates model training is defined in 27. Because training samples are i.i.d., the variance of the gradient is

$$\mathrm{var}\big[\nabla\mathcal{L}_B(\mathbf{V},\mathbf{W},\mathbf{b})\big] = \mathrm{var}\left[\frac{1}{|B|}\sum_{i=1}^{|B|}(p_i - y_i)\nabla f(\mathbf{x}_i)\right] \ , \tag{28}$$

$$= \frac{1}{|B|}\mathrm{var}\big[(p-y)\nabla f(\mathbf{x})\big] \tag{29}$$

The gradient has two components: $p - y$ corresponding to $\gamma$ in equation 1 and $\nabla f(\mathbf{x})$ corresponding to $\mathbf{f}(\mathbf{q}, \xi)$. We assume that the dataset is unbiased, in which case $P(y = 0) = P(y = 1) = 0.5$ and $P(p - y = a) = P(p - y = -a)$, and that $p - y$ and $\nabla f(\mathbf{x})$ are independent in the first period of training given that the dataset is complex and can't be learned by linear model. It is straightforward to see that it satisfies Eq. 2.

Next we will show that $V$ and $b$ are always in the set of $U$ defined in the previous section. First, if $V_i \geq 0$, we have

$$\nabla_{V_i} f(\mathbf{x}) = \sigma(\sum_{j=1}^{M} W_{ij}x_j + b_i), \tag{30}$$

$$\geq 0. \tag{31}$$

and

$$\nabla_{b_i} f(\mathbf{x}) = V_i\sigma'(\sum_{k=1}^{M} W_{ik}x_k + b_i), \tag{32}$$

$$\geq 0. \tag{33}$$

Since we also have Property A.1, the conditions in Eq. 4 are satisfied. If $V_i < 0$, we consider a coordinate transform that maps $V_i$ to $\bar{V}_i = -V_i$. It is easy to show that Eq. 4 is again satisfied after this transform.

Next we consider $\mathbf{W}$. The gradient of $f$ with respect to $W_{ij}$ is the product of $\nabla_{b_i} f$ and $x_i$. If $x_i$ for $i = 1, \ldots, M$ are always $\geq 0$, which is usually the case in convolutional neural networks, it is easy to show that $W_{ij}$ is also in set $U$ and smaller $W_{ij}$ corresponds to smaller variance according to Property A.2. If $x_i \sim \mathcal{N}(0, 1)$, on the other hand, $\mathbf{W}$ is excluded from $U$.

For the following, we consider the case where $x_i \sim \mathcal{N}(0, 1)$, and

$$g_V(V_i, \mathbf{W}_i, b_i) = \frac{1}{\sqrt{|B|}}\sqrt{\int \big[(p-y)\sigma(\sum_{j=1}^{M} W_{ij}x_j + b_i)\big]^2 d\mu(x,y)} \ , \tag{34}$$

$$= \frac{1}{\sqrt{|B|}}\phi_1(\mathbf{W}_i, b_i) \ , \tag{35}$$

$$g_b(V_i, \mathbf{W}_i, b_i) = \frac{1}{\sqrt{|B|}}\sqrt{\int \big[(p-y)V_i\sigma'(\sum_{j=1}^{M} W_{ij}x_j + b_i)\big]^2 d\mu(x,y)} \ , \tag{36}$$

$$= \frac{V_i}{\sqrt{|B|}}\phi_2(\mathbf{W}_i, b_i) \ , \tag{37}$$

where $g$ is defined as in 3. Inserting these into Eq. 6, we find the thermophoresis flow density to be

$$J_t = \frac{\eta^2}{|B|}\psi \ , \tag{38}$$

where $\psi = \frac{V_i\phi_1\phi_2^2 + V_i\phi_1\phi_2\partial_b\phi_1 + V^3\phi_2^2\partial_b\phi_2}{2\sqrt{\phi_1^2 + V_i^2\phi_2^2}}\rho$. This flow biases the model toward smaller $b_i$ and smaller $V_i$[6] with the strength proportional to squared learning rate $\eta^2$ and inverse batch size. It is also noted

[6] larger $V_i$ if $V_i < 0$.

that the function $\psi$ can be bounded by a function multiplying with a scalar $\int (p-y)^2 \mu(x,y)$. It is clear that this scalar measures the L-2 distance between model predictions and sample labels and decreases on average during training as prediction getting better. Thus thermophoresis is more effective during the early phase of training.

### A.8 Sparsity, Weight Norm and Their Relation to Generalization

In this section, we demonstrate how sparsity is related to the Hessian norm. We first denote the model's probabilistic prediction on a $C$-class classification as

$$p_k^\mu = \frac{\exp z_k^\mu}{\sum_{l=1}^{C} \exp z_l^\mu} \ , \tag{39}$$

where $k$ is the probability for label $k$, $\mu$ is the data index, $z$ is model output and $C$ is the total number of categories. We consider cross entropy loss of the form

$$L(w) = -\frac{1}{B} \sum_{\mu=1}^{B} \sum_{k=1}^{C} y_k^\mu \log p_k^\mu \ , \tag{40}$$

where $y$ is sample labels and $p$ stands for model probability prediction, similar to the previous definition. We denote the loss for individual sample to be

$$L^\mu = -\sum_{k=1}^{C} y_k^\mu \log p_k^\mu \ . \tag{41}$$

The gradient with respect to the model output is

$$(\nabla_z L^\mu)_k = -y_k^\mu + p_k^\mu \ . \tag{42}$$

And it is easy to show that the Hessian with respect to output is

$$(\nabla_z^2 L^\mu)_{kl} = \delta_{kl} p_k^\mu - p_k^\mu p_l^\mu \ . \tag{43}$$

Therefore the Hessian with respect to model parameters is

$$\begin{aligned} H^\mu &= \nabla_w^2 L(z(w)) \ , && (44)\\ &= \nabla_w (\nabla_z L * \nabla_w z) \ , && (45)\\ &= (\nabla_w z)(\nabla_z^2 L)(\nabla_w z) + \nabla_z L \nabla_w^2 z \ , && (46)\\ &\approx (\nabla_w z^\mu)_{ij} (\nabla_z^2 L^\mu)_{jk} (\nabla_w z^\mu)_{kl} \ . && (47) \end{aligned}$$

To study the spectrum of the Hessian, we calculate the trace and have

$$\begin{aligned} Tr(H^\mu) &\approx Tr((\nabla_w z^\mu)(\nabla_z^2 L^\mu)(\nabla_w z^\mu)^T) \ , && (48)\\ &= Tr((\nabla_z^2 L^\mu)(\nabla_w z^\mu)^T (\nabla_w z^\mu)) \ , && (49)\\ &= Tr(P * K) \ , && (50)\\ &\leq Tr(P) * Tr(K) \ , && (51) \end{aligned}$$

where

$$\begin{aligned} P &= \nabla_z^2 L^\mu \ , && (52)\\ K_{\mu\nu} &= \sum_l \sum_{ij} (\frac{\partial z_\mu}{\partial w_{ij}^l})(\frac{\partial z_\nu}{\partial w_{ij}^l}) \ . && (53) \end{aligned}$$

The trace of K therefore can be calculated by chain rule,

$$\begin{aligned} Tr(K) &= \sum_\mu \sum_l \sum_{ij} (\frac{\partial z_\mu}{\partial w_{ij}^l})^2 \ , && (54)\\ &= \sum_l (\sum_{iu} (\delta_i^l[\mu])^2 \sum_j (h_j^{l-1})^2) \ , && (55) \end{aligned}$$

where $\delta$ and $h$ carry backward and forward information respectively. They are defined as

$$\delta_i^l[\mu] = \delta_{\mu n_L} W^L_{n_L n_{L-1}} \sigma' ... W^{l+1}_{n_{l+2} i} \sigma' \ , \tag{56}$$

$$h_j^{l-1} = \sigma W^{l-1}_{j n_l} \sigma ... W_{n_1 n_0} x_{n_0} \ . \tag{57}$$

It can further be shown that

$$\sum_{iu} (\delta_i^l[\mu])^2 = \sum_{iu} \delta_{\mu n_L} W^L_{n_L n_{L-1}} \sigma' \dots W^{l+1}_{n_{l+2} i} \sigma' * \sigma' \overline{W^{l+1}}_{i n_{l+2}} \dots \sigma' \overline{W^L}_{n_{L-1} n_L} \delta_{n_L \mu} \ , \tag{58}$$

$$= Tr(W^L \sigma' \dots W^{l+1} \sigma' \sigma' \overline{W^{l+1}} \dots \sigma' \bar{W}^L) \ , \tag{59}$$

$$\leq Tr(\sigma' \overline{W^L} W^L \sigma') Tr(\sigma' \overline{W^{L-1}} W^{L-1} \sigma') \dots Tr(\sigma' \overline{W^{l+1}} W^{l+1} \sigma') \ , \tag{60}$$

as well as

$$\sum_j (h_j^{l-1})^2 \leq \|X\|_2 Tr(\overline{W^1} \sigma \sigma W^1) Tr(\overline{W^2} \sigma \sigma W^2) ... Tr(\overline{W^{l-1}} \sigma \sigma W^{l-1}) \ . \tag{61}$$

Together with the previous calculations and the definition of $K$, we have

$$Tr(K) \leq \|X\|_2 \sum_l \Pi_{n=1}^{l-1} Tr(\overline{W^n} \sigma \sigma W^n) \Pi_{n=l+1}^{L} Tr(\sigma' \overline{W^n} W^n \sigma') \ , \tag{62}$$

$$= \|X\|_2 \sum_l \Pi_{n=1}^{l-1} \|\sigma W^n\|_F^2 \Pi_{n=l+1}^{L} \|W^n \sigma'\|_F^2 \ . \tag{63}$$

Finally, we derive an upper bound for the trace of the Hessian,

$$Tr(H^\mu) \leq Tr(P) \|X\|_2 \sum_l \Pi_{n=1}^{l-1} \|\sigma W^n\|_F^2 \Pi_{n=l+1}^{L} \|W^n \sigma'\|_F^2 \ . \tag{64}$$

Notice that activation rate and weight norm control the magnitude of $\|\sigma W^n\|_F^2$ and $\|W^n \sigma'\|_F^2$. Therefore smaller activation rate and weight norm lead to tiger upper bound of the Hessian trace and thus indicate smaller matrix norm. This analysis connects sparsity with Hessian norm, Hessian trace specifically.