Imperial College Summer Research

Under the supervision of Professor Greg Pavliotis

Pantelis Tassopoulos

Summer 2023

Contents

1 Overview

1.1 Objectives

- To examine the existing literature on recent developments in the context of theoretical machine learning that integrate tools from statistical physics and probability theory, i.e., the theory of interacting particle systems.
- To analyse the approximation quality and trainability of neural networks using algorithms, such as Stochastic Gradient Descent (SGD), informed by such ideas on toy models and examples with real life examples such as the MNIST digit classification dataset.
- To perform numerical experiments by training neural networks under various circumstances, thereby graining practical insights.
- To try and extend results from the literature by attempting to provide theoretical guarantees for accuracy and robustness of machine learning algorithms other than SGD or new insights from numerical simulations.

1.2 Outcomes

This Summer Project (UROP) gave me a better insight into cutting-edge research in theoretical machine learning and mathematical optimisation.

I reviewed the requisite background material in mathematics from reference material, including textbooks and relevant papers. For instance, I read up on topics in probability, namely, martingale inequalities (Doob's and Hoeffman's inequalities in the book of Bremaud entitled 'Probability Theory and Stochastic Processes' [**?**]) that Mei et al. in their 2018 paper entitled 'A mean-field view of the landscape of two-layer neural networks' used in proofs of convergence of the SGD dynamics to the evolution of a Partial Differential Equation (PDE) as the hidden layer had an ever-increasing number of nodes, which enabled to perform novel theoretical analyses and provide theoretical guarantees of convergence.

I did some additional reading to supplement my understanding of the 2019 papers by Spiliopoulos and Sirgiano entitled 'Mean Field Analysis of Neural Networks: A Law of Large Numbers' and its companion paper [**?**], [**?**] . I read part of the book entitled 'Markov Processes: Characterisation and Convergence' by Stewart N. Ethier Thomas G. Kurtz [**?**], specifically the chapter on weak convergence of probability measures with values on the Skorokhod space $\mathcal{D}_{E[0,\infty)}$, which was necessary for understating the author's arguments on propagation of chaos and analogous convergence arguments.

Another crucial component of the project was the emphasis on numerical experiments. They allowed me to demonstrate the validity of theoretical findings and strengthen the case for the arguments presented. Numerical simulations involved training neural networks using existing algorithms from the literature and using insights gained to develop new algorithms.

For instance, regarding the above papers by Spiliopoulos, to supplement my understanding and empirically demonstrate claims made in the above paper, I performed numerical simulations by training a family of single-layer neural networks that achieved single-digit classification on the MNIST data set (used for digit classification and is a well-known benchmark for testing models). Upon expanding their hidden layer and training them, I plotted histograms of the distribution parameters which, for sufficiently many hidden nodes, the distribution of node values seemed to stabilise around a fixed bimodal distribution, which is also what the authors reported (while they did not specify the exact nature of the neural network they trained).

The project's theme shifted from analysing plain SGD towards understanding the wildly non-convex landscape of the underlying objective/loss function one typically encounters in machine learning applications.

In this direction, I demonstrated, among other observations, which can be found on my GitHub page [**?**] using numerical simulations that Nesterov accelerated gradient descent escaped a 'bad minimum', where SGD got stuck in a loss function that was constructed in [**?**].

This motivated me to generalise further insights gained by examining the dynamics of SGD to momentum-based algorithms, including Nesterov's accelerated Gradient Descent.

At that time, I read the 2017 paper by Chaudhari et al. entitled 'Deep Relaxation: partial differential equations for optimising deep neural networks' [**?**]. They introduced various approaches centred around 'regularising' the loss function. I incorporated both momentum-based methods (including 'restarting' the momentum if the gradient in the change in position was in the direction of the gradient-maximal increase, as was introduced in the 2012 Candes et al. paper entitled 'Adaptive Restart for Accelerated Gradient Schemes' [**?**]) and regularising the potential (by leveraging the analytical properties of solutions to the Hamilton-Jacobi-Bellman equation) as suggested above to create an algorithm that attempted to escape bad minima.

I also revisited the 2023 paper by Andrew Stuart et al. entitled 'Gradient Flows for Sampling: Mean-Field Models, Gaussian, Approximations and Affine Invariance'[**?**] initially suggested by my supervisor to produce another algorithm based on theoretical insights gained from the paper.

Furthermore, I read the paper coauthored by my supervisor entitled 'The sharp, the flat and the shallow: Can weakly interacting agents learn to escape bad minima?' [**?**] I was also led to study the analysis of multiscale algorithms in the literature, e.g. in the Weinan et al. (2005) paper [**?**] on the analysis of multiscale methods for SDEs. The authors devised an algorithm that escaped bad minima in a toy example they introduced in the paper.

As suggested by my supervisor, I implemented the above algorithms by performing descent on a loss that was a Muller-Brown potential (the canonical example of a potential surface in theoretical chemistry). My instance had a narrow global minimum; SGD would perform poorly and tend to converge to two local minima, of which there were two in a relatively confined domain. For each algorithm mentioned above, I performed random initialisations, ran the algorithms for a fixed number of steps and recorded the final 'losses'. One notable observation is that the implementation of the algorithm in my supervisor's paper performed noticeably better than the rest, including plain SGD.

This research experience presented an excellent opportunity for me to go beyond the scope of material covered in class and explore developments in the literature in a structured and rigorous manner.

Please note all the code referenced herein can be found on my personal GitHub page [**?**] .

2 Mean Field Limits of Neural Networks

2.1 Background theory

The process of a neural network 'learning' from data requires solving a complex optimisation problem with millions of variables. This is done by stochastic gradient descent (SGD) algorithms. One can study the case of two-layer networks and derive a compact description of the SGD dynamics in terms of a limiting partial differential equation. This a major insight in [**?**], where they authors also suggest with their findings that SGD dynamics do not become more complex when the network size increases.

Now, more formally, one typically encounters, in the context of supervised learning the following:

- Observed data points $(x_i, y_i)_{i\in\mathbb{N}} \subseteq \mathbb{R}^d \times \mathbb{R}$, where they are modelled as being independent and indentically distributed (iid).
- The $x \in \mathbb{R}^d$ are called feature vectors and the $y \in \mathbb{R}$ the labels.
- The neural network essentially is a function that depends on some hidden parameters and the feature vector. In the case of a two-layer neural network, the dependence is modelled by:

$$\begin{aligned} \hat{y} : \mathbb{R}^d \times \mathbb{R}^{ND} &\to \mathbb{R} \\ (x;\theta) &\mapsto \tfrac{1}{N}\sum_{i=1}^{N} \sigma_*(x;\theta_i) \end{aligned} \tag{1}$$

  where N is the number of hidden units (neurons), $\sigma_* : \mathbb{R}^d \times \mathbb{R}^D \to \mathbb{R}$ an activation function and $\theta = (\theta)_{i\leqslant N}$, $\theta_i \in \mathbb{R}^D$ are parameters, often $\theta_i = (a_i, b_i, w_i)$ for real a_i, b_i, w_i and $\sigma_*(x;\theta_i) = a_i\sigma(\langle x, w_i\rangle + b_i)$ for some function $\sigma : \mathbb{R} \to \mathbb{R}$ (see figure 1).

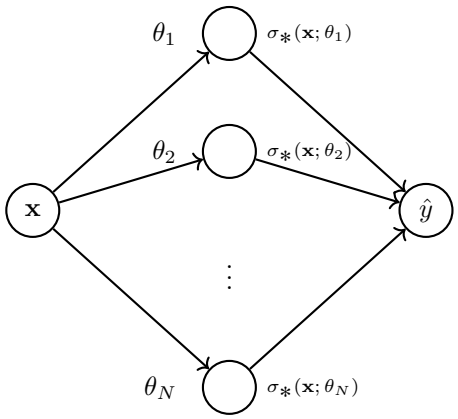


Figure 1: Illustration of a two layer neural network.

Naturally, one wants to chose parameters θ so as to minimise the risk function

$$R_N(x;\theta) = \mathbb{E}[\ell(y, \hat{y}(x;\theta))] \tag{2}$$

for a loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, typically and in our case the square loss $\ell(y - \hat{y}) = (y - \hat{y})^2$. This is achieved in practice by stochastic gradient descent summarised below:

Stochastic Gradient Descent (SGD)

Initialise the parameters $(\theta_i)_{i\leqslant\mathbb{N}} \sim \rho_0$, that is according to some initial distribution ρ_0.
while loss is greater than tolerance **do**
  Generate iid sample $(x, y) \sim \mathbb{P}$
  for $1 \leqslant i \leqslant N$ **do**
    $\theta_i \leftarrow \theta_i + 2s \cdot (y - \hat{y}(x;\theta)) \cdot \nabla_{\theta_i}\sigma_*(x;\theta_i)$ ▷ square loss is used
    Update learning rate s
  end for
end while

Observe that we have the alternative characterisation of the loss

$$R_N(\theta) = R_\# + \frac{2}{N}\sum_{i=1}^{N} V(\theta_i) + \frac{1}{N^2}\sum_{i,j=1}^{N} U(\theta_i, \theta_j). \tag{3}$$

where $V(x;\theta) = -\mathbb{E}[y \cdot \sigma_(x : \theta)]$, $U(\theta_1, \theta_2) = \mathbb{E}[\sigma_(x;\theta_1) \cdot \sigma_*(x;\theta_2)]$ and $R_\# = \mathbb{E}[y^2]$ is the risk of the trivial predictor $\hat{y} = 0$.

Notice that the collection of weights $\theta \in \mathbb{R}^{ND}$ induces a probability measure on $\mathbb{R}^{ND}$, namely its *empirical measure*:

$$\hat{\rho}^{(N)} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i} \tag{4}$$

Consider the function on the space of probability measures on $\mathbb{R}^D$, $\mathcal{P}(\mathbb{R}^D)$:

$$R : \mathcal{P}(\mathbb{R}^D) \to \mathbb{R}$$
$$\rho \mapsto R_\# + 2\int V(\theta)\rho(\mathrm{d}\theta) + \int\int U(\theta_1, \theta_2)\rho(\mathrm{d}\theta_1)\rho(\mathrm{d}\theta_2)$$

Observe we can thus express $R_N(\theta) = R(\hat{\rho}^{(N)})$. Now, performing the SGD algorithm 2.1 for k steps say (with step size $s_k = \epsilon \cdot \xi(k\epsilon)$ for some $\epsilon > 0$ and $\xi : \mathbb{R}_{\geqq 0} \to \mathbb{R}_{\geqq 0}$ sufficiently regular-see 2.1), we obtain the parameters $(\theta^k_{i\leqslant N})$ and their respective empirical measures $\hat{\rho}^{(N)}_k$. In [**?**], Theorem 2.1 here, it is shown that for all $t \geqq 0$, as $N \to \infty$ and $\epsilon \to 0$ in an appropriate way, the empirical measures $\hat{\rho}^{(N)}_{t/\epsilon}$ converge in the weak sense to some probability measure ρ_t whose dynamics are governed by the following PDE, which is referred to as *distributional dynamics* (DD) in [**?**]

$$\begin{aligned} \partial_t\rho_t &= 2\xi(t)\,\nabla_\theta \cdot \Big(\rho_t \nabla_\theta \Psi(\theta;\rho_t)\Big), \\ \Psi(\theta;\rho) &\equiv V(\theta) + \textstyle\int U(\theta,\theta')\,\rho(\mathrm{d}\theta'). \end{aligned} \tag{5}$$

(Note that $\nabla_\theta \cdot \mathbf{v}(\theta)$ denotes the divergence of the vector field $\mathbf{v}(\theta)$). This should be interpreted as an evolution equation in $\mathcal{P}(\mathbb{R}^D)$.

There is rich mathematical literature on the PDE 5 which was motivated by the study of interacting particle systems in mathematical physics (see the references in [**?**]). The authors in [**?**] use this to observe that 5 can be viewed as a gradient flow for the cost function $R(\rho)$ in the space $(\mathcal{P}(\mathbb{R}^D), W_2)$, of probability measures on $\mathbb{R}^D$ endowed with the Wasserstein metric.

Aside:
Regarding Wasserstein flows, I looked through the paper by Y. Chen , et al. [**?**] on Gradient Flows for Sampling and noted down some key insights from their paperIn brief, they study the problem of sampling a probability distribution with an unknown normalization constant, which is a fundamental problem in computational science and engineering. They recast it as an optimisation problem on the space of probability measures, using gradient flows.

(a) Given a gradient flow that one has constructed wrt a posterior distribution that one wants to sample from without having an explicit normalization, one can formulate a gradient flow and a system of particles with SDE of the MCKean Vlasov type with FK equation

the gradient flow (i.e. the evolution equation of the density).

(b) By making the gradient flow 'invariant' under affine reparameterizations (through preconditioning or by suitable choice of metric or energy functional on $\mathcal{P}(\mathbb{R}^d)$), one hope to improve performance of algorithms in the case of highly anisotripic posteriors, if there is an affine transformation that reduces the anisotropic nature of said posterior.

Recall that Wasserstein distance is defined as

$$W_2(\rho_1,\rho_2)=\Big(\inf_{\gamma\in\mathcal{C}(\rho_1,\rho_2)}\int\|\theta_1-\theta_2\|_2^2\gamma(\mathrm{d}\theta_1,\mathrm{d}\theta_2)\Big)^{1/2}. \tag{6}$$

In order to establish that these PDEs indeed describe the limit of the SGD dynamics, we make the following assumptions.

A1. $t\mapsto\xi(t)$ is bounded Lipschitz: $\|\xi\|_\infty,\|\xi\|_{\mathrm{Lip}}\leqslant K_1$, with $\int_0^\infty\xi(t)\,\mathrm{d}t=\infty$.

A2. The activation function $(\mathbf{x},\theta)\mapsto\sigma_*(\mathbf{x};\theta)$ is bounded, with sub-Gaussian gradient: $\|\sigma_*\|_\infty\leqslant K_2$, $\|\nabla_\theta\sigma_*(\mathbf{X};\theta)\|_{\psi_2}\leqslant K_2$. Labels are bounded $|y_k|\leqslant K_2$.

A3. The gradients $\theta\mapsto\nabla V(\theta)$, $(\theta_1,\theta_2)\mapsto\nabla_{\theta_1}U(\theta_1,\theta_2)$ are bounded, Lipschitz continuous (namely $\|\nabla_\theta V(\theta)\|_2,\|\nabla_{\theta_1}U(\theta_1,\theta_2)\|_2\leqslant K_3$, $\|\nabla_\theta V(\theta)-\nabla_\theta V(\theta')\|_2\leqslant K_3\|\theta-\theta'\|_2$, $\|\nabla_{\theta_1}U(\theta_1,\theta_2)-\nabla_{\theta_1}U(\theta_1',\theta_2')\|_2\leqslant K_3\|(\theta_1,\theta_2)-(\theta_1',\theta_2')\|_2$).

Theorem 2.1 (PM. Nguyen et al. (2018)). Assume that conditions A1, A2, A3 hold. For $\rho_0\in\mathcal{P}(\mathbb{R}^D)$, consider SGD with initialization $(\mathbf{theta}_i^0)_{i\leqslant N}\sim_{iid}\rho_0$ and step size $s_k=\epsilon\xi(k\epsilon)$. For $t\geqslant 0$, let ρ_t be the solution of PDE 5. Then, for any fixed $t\geqslant 0$, $\hat{\rho}^{(N)}_{\lfloor t/\epsilon\rfloor}\Rightarrow\rho_t$ almost surely along any sequence $(N,\epsilon=\epsilon_N)$ such that $N\to\infty$, $\epsilon_N\to 0$, $N/\log(N/\epsilon_N)\to\infty$ and $\epsilon_N\log(N/\epsilon_N)\to 0$. Further, there exists a constant C (depending uniquely on the parameters K_i of conditions A1-A3) such that, for any $f:\mathbb{R}^D\times\mathbb{R}\to\mathbb{R}$, with $\|f\|_\infty,\|f\|_{\mathrm{Lip}}\leqslant 1$, $\epsilon\leqslant 1$,

$$\begin{aligned}\sup_{k\in[0,T/\epsilon]\cap\mathbb{N}}\Big|\frac{1}{N}\sum_{i=1}^N f(\theta_i^k)-\int f(\theta)\rho_{k\epsilon}(\mathrm{d}\theta)\Big|&\leqslant Ce^{CT}\,\mathrm{Err}_{N,D}(z)\,,\\ \sup_{k\in[0,T/\epsilon]\cap\mathbb{N}}\big|R_N(\theta^k)-R(\rho_{k\epsilon})\big|&\leqslant Ce^{CT}\,\mathrm{Err}_{N,D}(z)\,,\end{aligned} \tag{7}$$

with probability $1-e^{-z^2}$ where $\mathrm{Err}_{N,D}(z)$ is given by

$$\sqrt{1/N\vee\epsilon}\cdot\Big[\sqrt{D+\log N/\epsilon}+z\Big] \tag{8}$$

Theorem 2.2 (Doob's martingale inequality). Let $(\mathcal{F}_t)_{t\geqslant 0}$ be a filtration on the probability space $(\Omega,\mathcal{F},\mathbb{P})$ and $(M_t)_{t\geqslant 0}$ be a continuous martinagle adapted to the filtration $(\mathcal{F}_t)_{t\geqslant 0}$. Let $p\geqslant 1$ and $T>0$. If $\mathbb{E}[|M_T|^p]<\infty$ and $\lambda>0$, then

$$\mathbb{P}\left(\sup_{t\in[0,T]}|M_t|\geqslant\lambda\right)\leqslant\frac{\mathbb{E}[|M_T|^p]}{\lambda^p} \tag{9}$$

Lemma 2.1 (Hoeffding's Lemma). Let $(M_n)_{n\in\mathbb{N}}$ be a martingale adapted to the filtration $(\mathcal{F}_n)_{n\in\mathbb{N}}$ such that for some sequence $c_1,c_2,\cdots$ of real numbers

$$\mathbb{P}(|M_n-M_{n-1}|\leqslant c_n)=1\quad\text{for all } n\in N. \tag{10}$$

Then for all $x \geqslant 0$ and all $n \geqslant 1$,

$$\mathbb{P}(|M_n - M_{n-1}| \geqslant x) \leqslant 2\exp\left(-\frac{1}{2}x^2 \Big/ \sum_{i=1}^{n} c_i^2\right) \tag{11}$$

Proof. (Rough Sketch) The conditions A1 and A3 guarantee the existence and uniqueness of solutions to the PDE 5, interpereted in the weak sense. The discrete SGD dynamics $(\theta_i^k)_{i\leqslant N}$ appxorimate the continuous time dynamics. Then the proof becomes technical and the aim is to control error terms incurred when comparing the deviation of the discrete and continuous dynamics in probability.

Notice we can also re-express 7 in terms of the empirical measure to deduce for all function f with $\|f\|_{\text{Lip}} \leqslant 1$, $\pi \in \mathcal{C}(\hat{\rho}_k^N, \rho_{k\epsilon})$ and $k \in [0, T/\epsilon]$

$$\begin{aligned} \Big|\int f(\theta)\hat{\rho}_k^N(\mathrm{d}\theta) - \int f(\theta)\rho_{k\epsilon}(\mathrm{d}\theta)\Big| &\leqslant \int |f(\theta) - f(\phi)|\pi(\mathrm{d}\theta, \mathrm{d}\phi) \\ &\leqslant \int \|\theta - \phi\|_2 \,\pi(\mathrm{d}\theta, \mathrm{d}\phi) \leqslant W_2(\hat{\rho}_k^N, \rho_{k\epsilon}) \end{aligned} \tag{12}$$

using Cauchy-Schwarz and taking the infimum over such couplings. Hence we obtain the bound

$$\sup_{k\in[0,T/\epsilon]\cap\mathbb{N}} \Big|\int f(\theta)\hat{\rho}_k^N(\mathrm{d}\theta) - \int f(\theta)\rho_{k\epsilon}(\mathrm{d}\theta)\Big| \leqslant \sup_{k\in[0,T/\epsilon]\cap\mathbb{N}} W_2(\hat{\rho}_k^N, \rho_{k\epsilon}) \tag{13}$$

This estimate helps one get a sense of the terms that need to be controlled in the proof of the theorem.

Moreover, the sub-gaussianity and Lipscitz continuity feature prominently and the tools used to achieve bounds on the the probabilities are mainly Doob's maximal inequality 2.2 and Hoefdding's lemma 2.1. □

The PDE formulation leads to several insights and simplifications. One can exploit symmetries in the data distribution $\mathbb{P}$ for instance. If $\mathbb{P}$ has rotational symmetry, then one can look for solutions to the PDE problem that share such rotational symmetry, thereby reducing the dimensionality of the problem which facilitates theoretical and numerical analysis. This is manifest in the case of two isotropic Gaussians considered later. Such symmetry cannot be achieved when considering the discrete dynamics since no set of poitns $\theta_1, \cdots, \theta_N \in \mathbb{R}^d$ is invariant under rotations (excluding trivial cases).

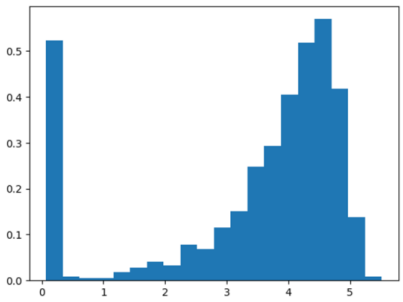
2.2 Centred Isotropic Gaussians

The authors in [**?**] were interested in numerically testing their PDE framework on the classification problem of Gaussians with the same mean. That is, assume the joint law $\mathbb{P}$ of $(\mathbf{x}, y)$ to be:

$$\begin{aligned} &\text{with probability} 1/2 : y = +1, \mathbf{x} \sim N(0, (1+\Delta)^2 \cdot \mathrm{Id}_d) \\ &\text{with probability} 1/2 : y = -1, \mathbf{x} \sim N(0, (1-\Delta)^2 \cdot \mathrm{Id}_d) \end{aligned} \tag{14}$$

For the activation function set $\sigma(\mathbf{x}; \theta) = \sigma(\langle w, \mathbf{x}\rangle)$ where σ is a simple piewewise linear activation function.

To try and reproduce the findings in [**?**],I implemented the SGD and the asymptotic PDE for the isotropic Gaussian case (SGD for isotropic gaussians with 10^7 iterations) with $(w_i^0)_{i\leqslant\mathbb{N}} \sim_{iid} \rho_0$, where ρ_0 is spherically symmetric. More specifically, I ran a monte carlo simulation of the discrete SGD dynamics and recorded the distance of the particles (hidden parameters θ) after a set amount of iterations and aggreagated them, producing figure 2a. It compares nicely with the last subplot in 2b.



(a) fig: SGD histogram for isotropic Gaussians with 10^7 iterations.

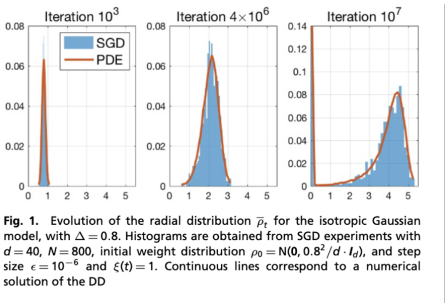


Fig. 1. Evolution of the radial distribution $\bar{\rho}_t$ for the isotropic Gaussian model, with $\Delta = 0.8$. Histograms are obtained from SGD experiments with $d = 40$, $N = 800$, initial weight distribution $\rho_0 = \mathrm{N}(0, 0.8^2/d \cdot I_d)$, and step size $\epsilon = 10^{-6}$ and $\xi(t) = 1$. Continuous lines correspond to a numerical solution of the DD

(b) Corresponding simulation in the paper by Nguyen et al.

Figure 2: Isotropic Gaussian SGD simulation.

Upon suggestion of my supervisor, I studied the possible non-uniqueness of stationary states for the mean field PDE that is derived in the paper by Mei et al.' and its connection to the fact that SGD does not always converge to a near global optimum. There they introduce a non-monotone activation function

$$\sigma_*(\mathbf{x};\theta) = \sigma(\langle w, x\rangle), \tag{15}$$

where $\sigma(t) = 2.5 for t \leqslant 0, (t) = 7.5$ for $t \geqslant 1.5$, and $\sigma(t)$ linearly interpolates from $(0, 2.5)$ to $(0.5, 4)$, and from $(0.5, 4)$ to $(1.5, 7.5)$, see figure 3.

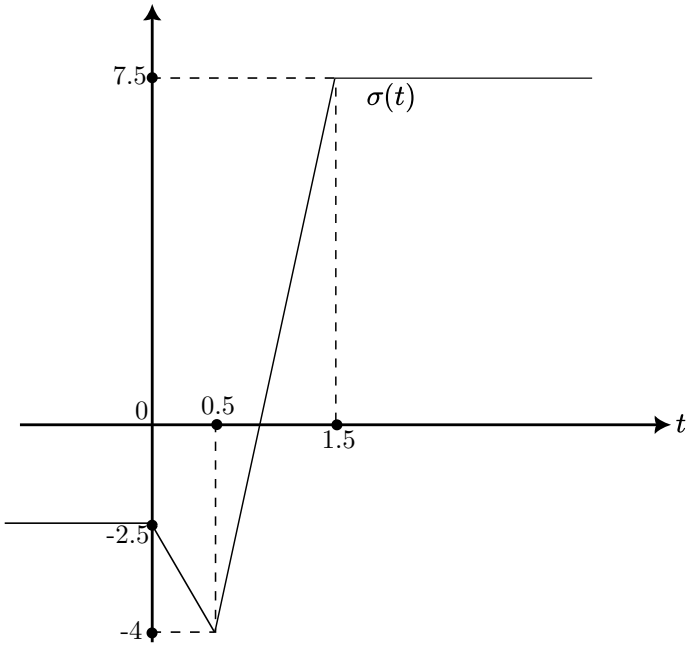
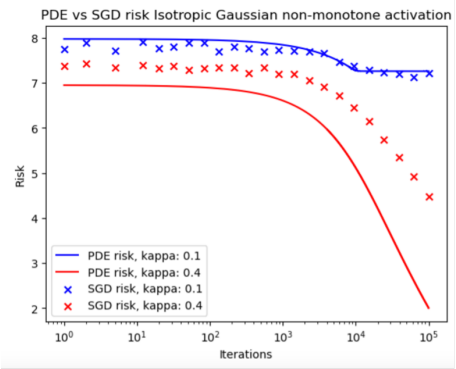


Figure 3: Non-monotone activation function σ.

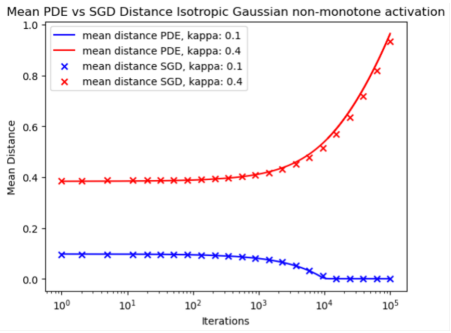
Depending on the initialization,

$$(w_i^0)_{i\leqslant N} \sim_{iid} N(0, \kappa^2/d\,\mathrm{Id}_d) \tag{16}$$

with $= 0.4, 0.1$, the SGD converges to two different limits, one with a small risk, and the second with high risk (respectively). I reproduced this phenomenon in with a close match to the data presented in the [**?**], see figure 4.



(a) Plot comparing PDE vs SGD risk for the non-monotone activation σ.



(b) Plot comparing the average distance of the weights $\|w\|_2$ in the PDE vs SGD simulations for the non-monotone activation.

Figure 4: Separating two isotropic Gaussians, with a non-monotone activation function σ. Here $N = 800, d = 320, \Delta = 0.5$. Continuous lines are prediction obtained with the Distributional Dynamics simplified to reflect the spherical symmetry of the problemS.

2.3 Anisotropic Gaussians

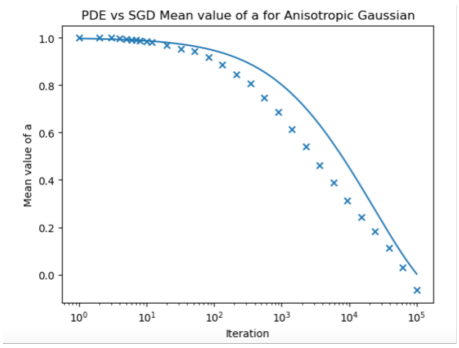
I also implemented code for SGD and PDE simulations of risk for the non-isotropic Gaussian case of failure of SGD given two initializations motivated by the theory developed in [**?**]. The PDE simulations reproduce the numerical findings in the paper, but SGD runs fail to match the PDE profiles exactly, though qualitatively, they are similar (figure 5. More precisely, the data is now

$$\begin{aligned} &\text{with probability} 1/2 : y = +1, \mathbf{x} \sim N(0, \Sigma_+) \\ &\text{with probability} 1/2 : y = -1, \mathbf{x} \sim N(0, \Sigma_-) \end{aligned} \tag{17}$$

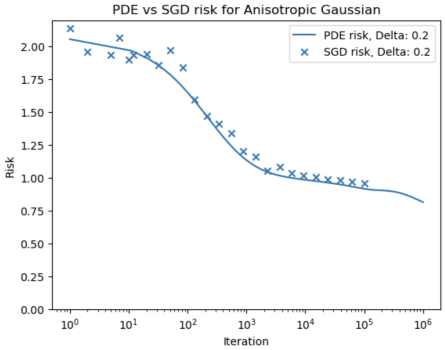
where

$$\begin{aligned} \Sigma_+ &= \text{Diag}(\underbrace{(1+\Delta)^2, \cdots, (1+\Delta)^2}_{s_0}, \underbrace{1, \cdots, 1}_{d-s_0}) \\ \Sigma_- &= \text{Diag}(\underbrace{(1-\Delta)^2, \cdots, (1-\Delta)^2}_{s_0}, \underbrace{1, \cdots, 1}_{d-s_0}) \end{aligned} \tag{18}$$

and as in the previous case, we choose $\sigma_*(\mathbf{x}; \theta_i) = a_i \sigma_{\text{ReLU}}(\langle \mathbf{x}, w_i \rangle + b_i)$ where $\sigma_{\text{ReLU}}(x) = \max x, 0$.



(a) Mean value of a for PDE and SGD in the anisotropic Gaussian case.



(b) PDE vs SGD risk for Anisotropic Gaussian

Figure 5: caption

2.4 MNIST data classification

I studied the proofs for the propagation of chaos and the mean-field limit of the distribution of neural network weights in the 2019 paper of Spiliopoulos and Sirgiano entitled 'Mean Field Analysis of Neural Networks: A Law of Large Numbers' and its companion paper [**?**], [**?**]. I did some background reading to supplement my understanding of the above papers of a book entitled 'Markov Processes: Characterization and Convergence' by Stewart N. Ethier, Thomas G. Kurtz

[?], specifically the chapter on weak convergence of probability measures with values in Skorokhod spaces defined below.

Definition 2.1 (Skorokhod space). Let $E = (\mathcal{M}, d)$ be a metric space and $T > 0$. Then, we deefine the Skorokhod space

$$\mathcal{D}([0,T];E) := \{f : [0,T] \to E : f \text{ is cadlag}\}. \tag{19}$$

This mean field convergence of the empirical measures induced by the weights of neural networks was also performed in papers [?] and [?]. In a similar setup to [?], the SGD algorithm produces obtains empirical measures expended in a piewewise constant manner to $\mu_t^N = \hat{\rho}^N_{\lfloor Nt \rfloor}$ for $t \geqslant 0$, see figure 6.

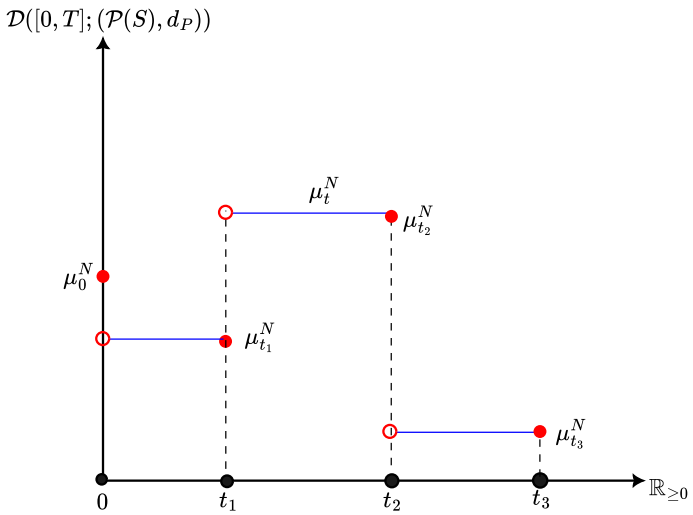


Figure 6: Piecewise constant extension of scaled empirical measures μ_t^N.

Now, by construction, we have that the empirical measure process $(\mu_t^N)_{t\in[0,T]}$ is an element of the space of locally finite Borel measures on $\mathbb{R}^d$, $\mathcal{M}(\mathbb{R})^d$. One can define the notion of *vague convergence* of a family $(\nu_n)_{n\in\mathbb{N}} \xrightarrow{v} \nu \in \mathcal{M}(\mathbb{R}^d)$ by

$$\int_{\mathbb{R}^d} f \,\mathrm{d}\nu_n \overset{n\to\infty}{\to} \int_{\mathbb{R}^d} f \,\mathrm{d}\nu_n \quad f \in \hat{\mathcal{C}}(\mathbb{R}^d). \tag{20}$$

where $\hat{\mathcal{C}}(\mathbb{R}^d)$ denotes the space of all bounded continuous non-negative functions with bounded support. Note that the family of maps $\{\pi_f : f \in \hat{\mathcal{C}}_{\mathbb{R}^d}\}$ induces the vague topology $\mathcal{T}$:

Lemma 2.2 (Vague topology on $\mathcal{M}_\mathcal{S}$). Let $\mathcal{S}$ be a complete separable metric space, then there exists a topology $\mathcal{T}$ on $\mathcal{M}_\mathcal{S}$ such that

1. $\mathcal{T}$ induces the convergence $\nu_n \xrightarrow{v} \nu$ in 20,
2. $\mathcal{M}_\mathcal{S}$ is Polish under $\mathcal{T}$,
3. $\mathcal{T}$ generates the Borel sigma algebra $\sigma(\{\pi_f : f \in \hat{\mathcal{C}}_{\mathbb{R}^d}\})$.

Hence, we have that the scaled empirical measure $(\mu_t^N)_{t\in[0,T]})$ is a random element of $\mathcal{D}_{\mathcal{M}(\mathbb{R})^d} := \mathcal{D}([0,T];(\mathcal{M}(\mathbb{R}^d), d_\mathcal{T}))$, where $d_\mathcal{T}$ is the induced metric from 2.2. Note that $\mathcal{D}_{\mathcal{M}(\mathbb{R})^d}$ space is a Polish space in its own space endowed with the *Skorokhod topology* with well-understood criteria for compactness that feature prominently in the proof of the main theorem in [?], and [?].

The main result of the paper [?] concerns the convergence in distribution of μ_t^N in the aforementioned Skorokhod space under certain 'reasonable' structural assumptions.

Theorem 2.3 (Spiliopoulos LLN). For all $T > 0$, the scaled empirical measure μ_t^N on $[0, T]$ converges in distribution to a limit measure $\bar{\mu}_t$ with values in $\mathcal{D}_{\mathcal{M}_{\mathbb{R}^d}}$ as $N \to \infty$.

Remark. μ_t has a characterisation as the unique deterministic weak solution to a PDE, interpeted in the weak sense. Also, since the limiting measure μ_t is deterministic for all $t \geqslant 0$, we have the stronger convergence in Probability, that is for all $\delta > 0$

$$\lim_{N\to\infty} \mathbb{P}(d_{\mathcal{D}_{\mathcal{M}_{\mathbb{R}^d}}}(\mu^N, \bar{\mu}) \geqslant \delta) = 0$$

Moreover, I read the companion paper of Spiliopoulos (2019) [**?**] where the authors proved a CLT for a one-layer neural network. To this end, the authors in [**?**] the fluctuation process

$$\eta_t^N = \sqrt{N}(\mu_t^N - \bar{\mu}_t) \tag{21}$$

The main result in [**?**] is that asymptotically, as $N \to \infty$, the fluctuations converge in distribution, in a way made precise below, to some measure-valued process $\bar{\eta}$, where satisfies a stochastic partial differential equation. This result achieves to give a characterisation of the fluctuations of the finite empirical measure μ^N around its mean-field limit $\bar{\mu}$ for large N . It is noted that the $\bar{\eta}$ has a Gaussian distribution.

Theorem 2.4 (Spiliopoulos CLT). Under the 'reasonable' assumptions outlined in [**?**], $J \geqslant 3\left\lceil \frac{d}{2} \right\rceil + 7$ and any $0 < T < \infty$. The sequence

$$((\eta_t^N)_{t\in[0,T]})_{N\in\mathbb{N}} \xrightarrow{d} ((\bar{\eta}_t)_{t\in[0,T]})_{N\in\mathbb{N}} \tag{22}$$

in $\mathcal{D}([0, T]; W^{-J,2})$, as $\mathbb{N} \to \infty$ where $W^{-J,2}$ is the space of all continuous linear functionals on the *Sobolev space* $W_0^{J,2}(\Theta)$, where $\Theta \subseteq \mathbb{R}^d$ is a bounded domain independent of N.

Remark. For a brief introduction into the Sobolev spaces mentioned above, refer to section 2 to in [**?**].

Furthermore, regarding the paper of Spiliopoulos (2019) on the LLN for the one-layer neural net [**?**], I implemented a single-digit classifier with the architecture satisfying the assumptions made in the paper and was able to reproduce the distribution of parameters in the paper.

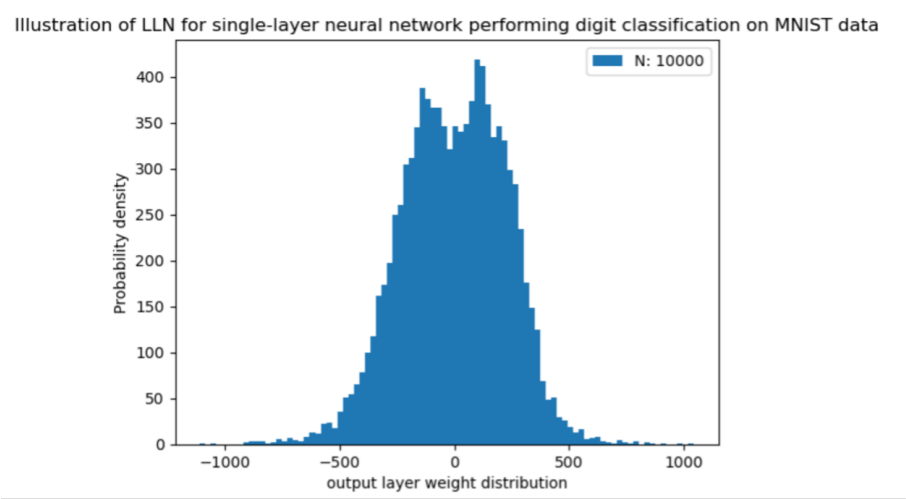


Figure 7: weights of MNIST single digit classification single layer neural net

3 Non-convex landscape

3.1 Approaches

At this point in the project, the focus started to shift from the theoretical mean-field analysis of neural network algorithms towards studying possible approaches to alleviate the failure of SGD

to reach a global minimum by potentially getting stuck in very sharp, yet non-global minima when the potential is wildly non-convex.

In this direction, I read my supervisor's paper on shallow minima: 'The sharp, the flat and the shallow: Can weakly interacting agents learn to escape bad minima?', [**?**]. In thie paper, the authors review several variants of SGD and illustrate that a system of interacting, rather than i.i.d., agents (essentially an interacting particle system) performing gradient descent can help to smooth out sharp minima and thus implicitly convexify the loss function.

The setting is a modification of the Stochastic Gradient Langevin Dynamics (SGLD) framework:

$$\mathrm{d}X_t = -\nabla\Phi(X_t)\,\mathrm{d}t + \sqrt{2}_t, \quad X_0 \sim \eta_0 \tag{23}$$

where Φ the loss function B_t is a standard Brownian motion and η_0 is the initial distribution.

Three approaches are discussed before a synthesis of the last two yields their proposed algorithm, see 3.1.4.

3.1.1 Regularise potential directly by convolution

To eliminate sharp local minima one could replace the gradient term in the basic gradient descent algorithm with a smoother version. In order to eliminate these local minima one could simulate the gradient descent dynamics of a "smoothed" version of the cost function instead

$$dX_t = -\nabla\Phi^h(X_t)dt \tag{24}$$

where we denote

$$\Phi^h(y) = (G_h \star \Phi)(y) = \int G_h(y-x)\Phi(x)dx, \tag{25}$$

i.e. $\star$ denotes the convolution. A typical choice for the smoothing kernel G_h is the Gaussian kernel with variance h

$$G_h(z) = \frac{1}{(2\pi h)^{d/2}} \exp\left(-\frac{\|z\|^2}{2h}\right).$$

For technical conditions for the above modification of the gradient, see the references in [**?**]. Regardless of the choice of the smoothing kernel, Φ^h can be interpreted as an expectation

$$\Phi^h(x) = \int \Phi(x+y)\mu(dy),$$

for a suitably chosen probability measure μ. Furthermore, (under appropriate conditions)

$$\nabla\Phi^h(x) = \int \nabla\Phi(x+y)\mu(dy). \tag{26}$$

An additional point here is that when $y \sim \mu$, $\nabla\Phi(x+y)$ is an unbiased estimate of the "biased" gradient $\nabla\Phi^h$. This implies, in particular, that there is an additional option for randomization of the smoothed gradient.

Loosely speaking the effect of μ here is to smooth Φ. It is natural to ask how one designs μ (or G_h) to get the desired effect of smoothing of Φ. There a multiple complications with such an approach, a pressing one being that computing the integral in 25 for the type of loss functions that appear in machine learning applications is intractable. To help mitigate these issues, the authros look at approaches where a smoothing measure μ does not act directly on Φ and is constructed from the stochastic process itself.

3.1.2 Regularise the potential implicitly weakly interacting agents

Next, I studied variants of 23 so that the process X_t will eventually end up closer to a minimizer of Φ with the rate of convergence taken into account. It is known that the law of the diffusion process X_t has a smooth density with respect to the Lebesgue measure that satisfies the Fokker-Planck (forward Kolmogorov) equation:

$$\partial_t\rho = \nabla\cdot\left(\rho\nabla\left(\beta\log\rho + \Phi\right)\right), \quad \rho(0,\cdot) = \eta_0(\cdot).$$

Under appropriate relatively mild assumptions on the growth of the loss function Φ at infinity, the density ρ converges exponentially fast in relative entropy (Kullback-Leibler divergence) to the unique stationary state

$$\rho_\infty = \frac{1}{Z}\exp(-\beta\Phi(x)),$$

(see for instance [**?**, Ch. 4]) where Z denotes the normalization constant. Finding the mode or maximiser of ρ_∞ is equivalent finding x^* (the minimizer of Φ). Here it is noted that higher values of β result to ρ_∞ putting more of its mass around lower valued local minima of Φ. For high β, the SGLD **??** reduces to a deterministic gradient descent, so it could be harder to escape from local minima. In practice, the trade-off between convergence speed and optimality is difficult to balance.

An alternative approach is to use interacting SGLD, as opposed to i.i.d. copies of the Langevin dynamics **??**. In [**?**], a system of interacting SGLD of the form

$$dX_t^i = -\nabla\Phi(X_t^i)dt - \left(\nabla D \star \eta_t^N\right)(X_t^i)dt + \sqrt{2\beta^{-1}}dB_t^i, \tag{27}$$

where $i = 1, \dots, N$, $\eta_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{X_t^i}$, $X_0^i \sim \eta_0(\cdot)$. Compared to the i.i.d. SGLD **??**, the dynamics 27 uses $D(x, y)$ as an interaction potential, which, in this paper, we will take to be convex. In particular, we will consider the so-called Curie-Weiss interaction

$$D(x, y) = \frac{\lambda}{2}\|x - y\|^2 \tag{28}$$

so that each particle experiences a linear attractive (mean reverting) force to the empirical mean of all particles

$$\nabla D \star \eta_t^N(X_t^i) = \lambda\left(X_t^i - \frac{1}{N}\sum_{j=1}^N X_t^j\right).$$

The framework developed here can be conceived of as an abstraction of popular machine learning algorithms, with ample references made in the paper [**?**, p. 4].

Under appropriate assumptions on the loss function, and on the initial conditions the position of each agent converges, in the limit $N \to \infty$ to the solution of the McKean SDE

$$\begin{aligned} d\bar{X}_t &= -\nabla\Phi(\bar{X}_t)dt - \nabla D \star \eta_t(\bar{X}_t)dt + \sqrt{2\beta^{-1}}dB_t, \\ \eta_t &= \mathcal{L}aw\left(\bar{X}_t\right). \end{aligned}$$

The density of the law of the process $\bar{X}_t$ is given by the McKean-Vlasov equation:

$$\partial_t\eta = \nabla\cdot\left(\eta\nabla\left(\beta\log\eta + \Phi + D\star\eta\right)\right), \quad \eta(0,\cdot) = \eta_0(\cdot). \tag{29}$$

This approach uses $\Phi + D \star \tilde{\eta}$ instead of Φ and acts to regularise or smooth out the cost function. From an optimization point of view, substituting $-\nabla\Phi(x) - \nabla D \star \eta_t^N(x)$ and using a linear interaction for ∇D is equivalent to using an ℓ_2-penalty in the objective function for the constraint: $X_t^i = \frac{1}{N}\sum_{j=1}^N X_t^j$, for each agent i. Therefore, for an appropriate choice of the interaction strength λ, the objective function is approximately convex.

3.1.3 Homogenisation

In the previous section 3.1.2 the empirical measure η_t^N was used to smooth the potential based on empirical properties of interacting agents. In this section, the approach that was developed in [**?**] to convert 23 into the following gradient descent algorithm:

$$d\tilde{X}_t = -\nabla\Phi^{\beta,\gamma}(\tilde{X}_t)dt, \quad \Phi^{\beta,\gamma}(x) = \int\Phi(x-y)\rho_\infty^x(dy), \tag{30}$$

is briefly discussed, where ρ_∞^x is the invariant measure of Y_t that appears in the limit when $\epsilon \to 0$ for the following fast/slow SDE system

$$dX_t = -\nabla\Phi(X_t - Y_t)dt \tag{31a}$$

$$dY_t = -\frac{1}{\epsilon}\left(\frac{1}{\gamma}Y_t - \nabla\Phi(X_t - Y_t)\right)dt + \sqrt{\frac{2\beta^{-1}}{\epsilon}}dB_t \tag{31b}$$

The parameter ϵ measures scale separation. The limit $\epsilon \to 0$ can be justified using multiscale analysis. Note that this is a gradient scheme for the modified loss function $\Phi(x - \frac{y}{\epsilon}) + \frac{1}{2\gamma}\left\|\frac{y}{\epsilon}\right\|^2$. It is noted in [**?**] that γ acts as a regularization parameter,precisely like the inverse of the interaction strength λ in the previous section. We emphasize the similarities between 25 and 30.

It is important to note that the smoothed loss function in 30 can also be calculated via convolution with a Gaussian kernel:

$$\Phi^{\beta,\gamma}(x) = \frac{1}{\beta}\log\Big(G_{\beta^{-1}\gamma} \star \exp(-\beta\Phi)\Big). \tag{32}$$

This is the Cole-Hopf formula for the solution of the viscous Hamilton-Jacobi equation with the loss function Φ as the initial condition, posed on the time interval $[0, \gamma]$. The larger γ is, the more regularized the effective potential (or relative entropy) $\Phi^{\beta,\gamma}(x)$ is.

Importantly for the authors in [**?**], in [**?**] an equivalent formulation to 31:

$$dX_t = -\frac{1}{\gamma}(X_t - Y_t)dt \tag{33a}$$

$$dY_t = -\frac{1}{\epsilon}\left(\nabla\Phi(Y_t) - \frac{1}{\gamma}(X_t - Y_t)\right)dt + \sqrt{\frac{2\beta^{-1}}{\epsilon}}dB_t. \tag{33b}$$

Here the regularized cost appears as $\Phi(\frac{y}{\epsilon}) + \frac{1}{2\gamma}\left\|x - \frac{y}{\epsilon}\right\|^2$. This form is more convenient for the numerical implementation and is the one that will be used in Algorithm 3.1.4.

3.1.4 Synthesis: combine both multi-scale analysis and weakly interacting gents for MF Hom SGLD

In brief this algorithm, 3.1.4 corresponds to a discretization of the dynamics of gradient descent against a potential with an ℓ_2 penalty and a regularized version of the original potential Φ, using the method introduced by Chaudhari et al. (2018).

More precisely, combining 30 with 27 we obtain:

$$dX_t^i = -\frac{1}{\gamma}(X_t^i - Y_t^i)dt - \lambda\left(X_t^i - \frac{1}{N}\sum_{j=1}^{N} X_t^j\right)dt \tag{34}$$

$$dY_t^i = -\frac{1}{\epsilon}\left(\nabla\Phi(Y_t^i) - \frac{1}{\gamma}(X_t^i - Y_t^i)\right)dt + \sqrt{\frac{2\beta^{-1}}{\epsilon}}dW_t^i \tag{35}$$

This scheme was tested numerically in the context of learning for the single layer neural network (see Section 2.1) with a sufficiently small value of ϵ, to approximate better the homogenized limit, as per [**?**]. The theoretical justification of this algorithm requires the study of the joint limits $\epsilon \to 0$ and $N \to +\infty$ (see [**?**, p. 6] for details and references).

To discretize 34-35 effectively for small ϵ I followed [**?**] and used the heterogeneous multiscale method [**?**] in Algorithm 3.1.4:

MF Hom SGLD

Require: $\mathsf{X}_0^i \sim \eta_0, \lambda \sim 1\Delta > 0$ ▷ Δ is a step size

for $n \geqslant 1, i = 1, \dots, N$ **do**

Set $\mathsf{Y}_{n,0}^i = \mathsf{Y}_{n-1,m'+M-1}^i$;

for $m = 1, \dots, M$ **do**

$$\begin{aligned}\mathsf{Y}_{n,m}^i &= \mathsf{Y}_{n,m-1}^i - \tfrac{\delta}{\epsilon}\left(\nabla\Phi(\mathsf{Y}_{n,m-1}^i) - \tfrac{1}{\gamma}(\mathsf{X}_{n-1}^i - \mathsf{Y}_{n,m-1}^i)\right)\\ &\quad + \sqrt{\tfrac{2\beta^{-1}\delta}{\epsilon}}Z_{n,m}^i;\ Z_{n,m}^i \sim N(0, I).\end{aligned}$$

end for

Compute average $\mathcal{Y}_n^i = \frac{1}{(m'+M-1)}\sum_{m=m'}^{m'+M-1}\mathsf{Y}_{n,m-1}^i$

Update

$$\mathsf{X}_n^i = \mathsf{X}_{n-1}^i - \frac{1}{\gamma}(\mathsf{X}_{n-1}^i - \mathcal{Y}_n^i)\Delta - \lambda\left(\mathsf{X}_{n-1}^i - \frac{1}{N}\sum_{j=1}^{N}\mathsf{X}_{n-1}^i\right)\Delta$$

end for

3.1.5 Nesterov SGD

Nesterov SGD
make connection to regularisation

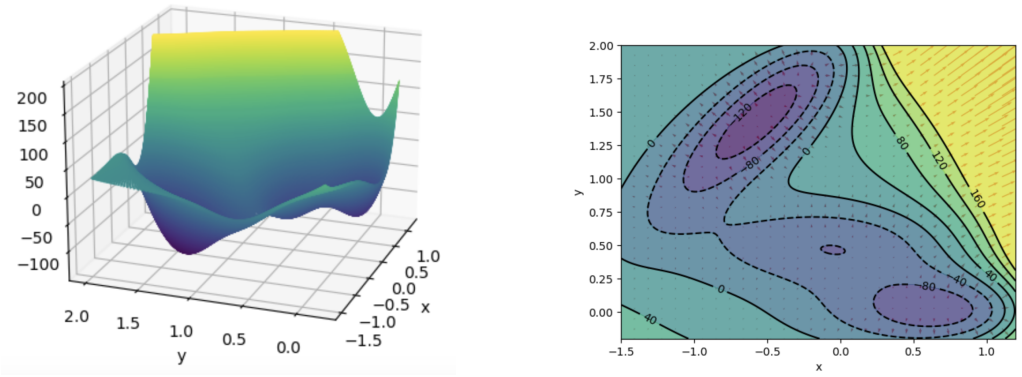
3.2 Applications

3.2.1 Muller Brown Potential

Having developed some theory regarding modifications to the vanilla SGD for non-convex landscapes, we applied it to the canonical example, at least from chemistry [**?**] of the Muller-Brown potential

$$V(x,y) = \sum_{i=i}^{4} A_i \cdot \exp[a_i \cdot (x-x_0)^2 + b_i \cdot (x-x_0) + c_i \cdot (y-y_0) + d_i \cdot (y-y_0)^2] \tag{36}$$

where $A_i, a_i, b_i, c_i, d_i, i \leqslant 4$ are as in the paper [**?**]. This potential has multiple local minima in close proximity, making it difficult for SGD to converge to the global minimum, see 8a.

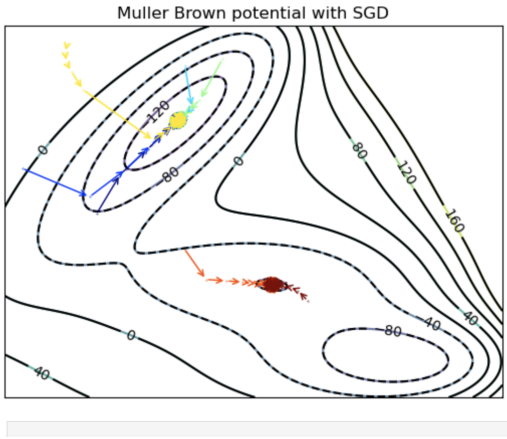


(a) 3d plot of the Muller-Brown potential

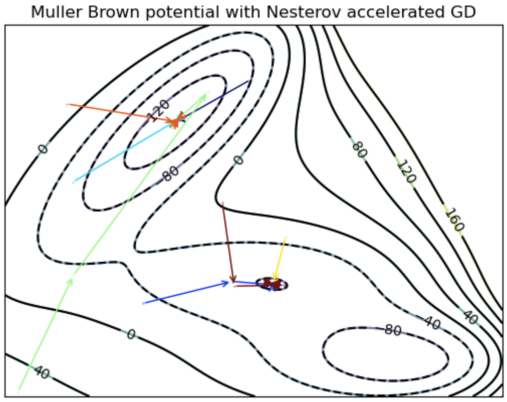
(b) caption

Figure 8: Contour plot of the Muller-Brown potential

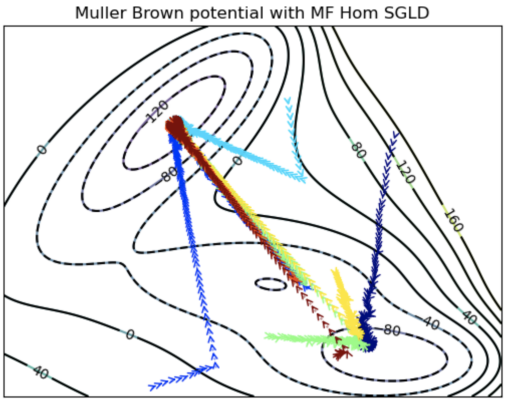
• Muller-Brown potential analysis: o - Most algorithms on the MB potential get stuck equally on two local minima, i.e. the global one (which is narrow) and the one with the next smallest local minimum o - Convolving with a solution to the heat equation does not improve performance as the narrow steep global minimum (as seen from the plot) is smeared out first thus giving no hope of real convergence, unless the algorithms is lucky with the initialisation o - The Hom-MF-SGLD works surprisingly well against all others since it performs a gradient flow of a regularised potential, where regularisation is done at the level of the gibbs measure o - Idea, sample points (to initialise GD) more judiciously, i.e. with gibbs measure (inspired by Andrew Stuart's paper) by performing a gradient flow and use that 'educated guess as the initialisation of a gd algorithm. (e.g. Wasserstein gradient flow, i.e. sgd on log of Gibbs measure of potential, or affine invariant Wasserstein) o - This idea seems to perform better that all algorithms except the MF-Hom-SGLD algorithm o - Regularising using the HJB equation (essentially performed in the MF-Hom SGLD algorithm) is better suited to minima that are narrower compared to regularisation with the heat equation that destroys such peaks first (due to large curvature) o - Tried regularising wrt soln of HJB equation directly and apply nesterov's accelerated gd algorithm with gradient restarting



(a) legend



(b) legend



(c) legend

Figure 9: my fig

3.2.2 3−d spin model analysis

"Neural networks, a central tool in machine learning, have demonstrated remarkable, high fidelity performance on image recognition and classification tasks. These successes evince an ability to accurately represent high dimensional functions, but rigorous results about the ap- proximation error of neural networks after training are few. Here we establish conditions for global convergence of the standard optimization algorithm used in machine learning applications, sto- chastic gradient descent (SGD), and quantify the scaling of its error with the size of the network. This is done by reinterpreting SGD as the evolution of a particle system with interactions governed by a potential related to the objective or "loss" function used to train the network. We show that, when the number n of units is large, the empirical distribution of the particles descends on a con- vex landscape towards the global minimum at a rate independent of n, with a resulting approximation error that universally scales as O(n1). These properties are established in the form of a Law of Large Numbers and a Central Limit Theorem for the empirical distribution. Our analysis also quantifies the scale and nature of the noise introduced by SGD and provides guidelines for the step size and batch size to use when training a neural network. We illustrate our findings on examples in which we train neural networks to learn the energy function of the continuous 3-spin model on the sphere. The approximation error scales as our analysis predicts in as high a dimension as d = 25.

Read the van Eijnden paper and studied the proofs of asymptotic convergence to a gradient flow in the mean field limit and how this is preferable due to the convexification of the loss (as a functional of measures), similar to that observed in [**?**].

To test our results, we use a function known for its complex features in high-dimensions: the spherical 3-spin model, $f : S^{d-1}(\sqrt{d}) \to \mathbb{R}$, given by

$$f(\mathbf{x}) = \frac{1}{d} \sum_{p,q,r=1}^{d} a_{p,q,r} x_p x_q x_r, \qquad \mathbf{x} \in S^{d-1}(\sqrt{d}) \subset \mathbb{R}^d \tag{37}$$

where the coefficients $\{a_{p,q,r}\}_{p,q,r=1}^{d}$ are independent Gaussian random variables with mean zero and variance one. The function (37) is known to have a number of critical points that grows exponentially with the dimensionality d [**?**, **?**, **?**]. We note that previous works have sought to draw a parallel between the glassy 3-spin function and generic loss functions [**?**], but we are not exploring such an analogy here. Rather, we simply use the function (37) as a difficult target for approximation by neural networks. That is, throughout this section, we train networks to learn f with a particular realization of $a_{p,q,r}$ and study the accuracy of that representation as a function of the number of particles n.

We first consider the case when $D = S^{d-1}(\sqrt{d})$ and we use

$$\varphi(\mathbf{x}, \mathbf{z}) = e^{-\frac{1}{2}\alpha|\mathbf{x}-\mathbf{z}|^2} \tag{38}$$

for some fixed $\alpha > 0$. In this case, the parameters are elements of the domain of the function (here the d-dimensional sphere). Note that, since $|\mathbf{x}| = |\mathbf{z}| = \sqrt{d}$, up to an irrelevant constant that can be absorbed in the weights c, we can also write (38) as

$$\varphi(\mathbf{x}, \mathbf{z}) = e^{-\alpha \mathbf{x}\cdot\mathbf{z}} \tag{39}$$

This setting allow us to simplify the problem. Using

$$f^{(n)}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} c_i \varphi(\mathbf{x}, \mathbf{z}_i) = \frac{1}{n} \sum_{i=1}^{n} c_i e^{-\alpha \mathbf{x}\cdot\mathbf{z}_i}, \tag{40}$$

we can use as alternative loss

$$\mathcal{L}[f^{(n)}] = -\frac{1}{n} \sum_{i=1}^{n} c_i f(\mathbf{z}_i) + \frac{1}{2n^2} \sum_{i,j=1}^{n} c_i c_j \varphi(\mathbf{z}_i, \mathbf{z}_j) \tag{41}$$

i.e. eliminate the need for data beside the set $\{\mathbf{z}_i\}_{i=1}^{n}$. In terms of the empirical distribution, the loss can be represented as

$$\mathcal{L}[f^{(n)}] = -\int_{\hat{D}} f(\mathbf{z}) \gamma^{(n)}(d\mathbf{z}) + \frac{1}{2} \int_{\hat{D}\times\hat{D}} \varphi(\mathbf{z}, \mathbf{z}') \gamma^{(n)}(d\mathbf{z}) \gamma^{(n)}(d\mathbf{z}') \tag{42}$$

where $\gamma^{(n)} = \int_{\mathbb{R}} c\mu^{(n)}(dc, \cdot)$. Viewed as an integral kernel, φ is positive definite, as a result the loss is a convex functional of $\gamma^{(n)}$ (or $\mu^{(n)}$). Hence, the results established above apply to this special case, as well. The GD flow on the loss (41) can now be written explicitly as

$$\begin{cases} \dot{z}_i = c_i \nabla f(\mathbf{z}_i) + \dfrac{\alpha}{n} \displaystyle\sum_{j=1}^{n} c_i c_j \mathbf{z}_j e^{-\alpha \mathbf{z}_i \cdot \mathbf{z}_j} - \lambda_i \mathbf{z}_i \\ \dot{c}_i = f(\mathbf{z}_i) - \dfrac{1}{n} \displaystyle\sum_{j=1}^{n} c_j e^{-\alpha \mathbf{z}_i \cdot \mathbf{z}_j} \end{cases} \tag{43}$$

where $-\lambda_i \mathbf{z}_i$ is a Lagrange multiplier term added to enforce $|\mathbf{z}_i| = \sqrt{d}$ for all $i = 1, \ldots, n$, $f(\mathbf{x})$ is given by (37), and $\nabla f(\mathbf{z})$ is given componentwise by

$$\frac{\partial f}{\partial z_p} = \frac{1}{d} \sum_{q,r=1}^{d} \left(a_{p,q,r} + a_{r,p,q} + a_{q,r,p} \right) z_q z_r, \tag{44}$$

As is apparent from (43) the advantage of using radial basis function networks (or, in fact, any unit $\hat{\phi}$ which is (i) such that $\hat{D} = \Omega$ and (ii) positive definite) is that we can use $f(\mathbf{x})$ and the unit $\varphi(\mathbf{x}, \mathbf{z})$ directly, and do not need to evaluate $\hat{F}(\mathbf{z})$ and $\hat{K}(\mathbf{z}, \mathbf{z}')$ (that is, we need no batch). In other words, the cost of running (43) scales like $(dn)^2$, instead of $P(Nn)^2$ in the case of a general network optimized by SGD with a batch of size P and $\mathbf{z} \in \hat{D} \subset \mathbb{R}^N$. If we make P scale with n, like $P = Cn^{2\alpha}$ for some $C > 0$, as we need to do to obtain the scalings discussed in Sec. **??**, the cost of SGD becomes $N^2 n^{2+2\alpha}$, which is quickly becomes much worse than $(dn)^2$ as n grows.

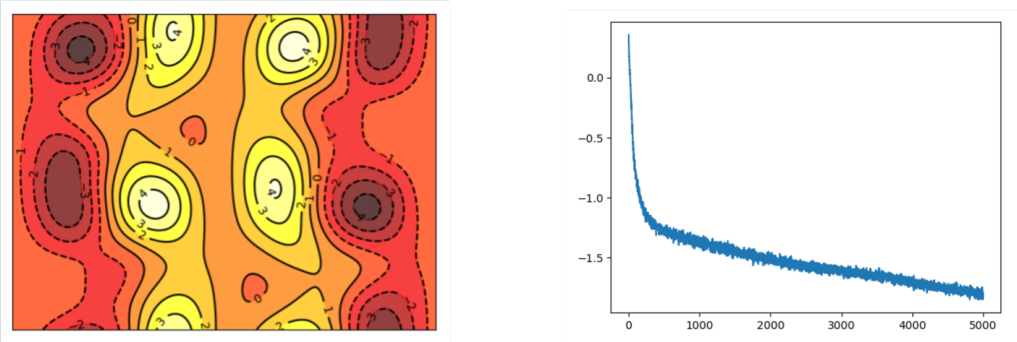
We tested the representation (40) in $d = 5$ using $n = 16$, 32, 64, 128, and 256 and setting $\alpha = 5/d = 1$. The training was done by running a time-discretized version of (43) with time step $\Delta t = 10^3$ for 2×10^5 steps: during the first 10^5 we added thermal noise to (43), which we then remove during the second half of the run. The representation (40) proves to be accurate even at rather low value of n: for example, the right panel of Fig. **??** shows a contour plot of the original function f and its representation $f^{(n)}$ with $n = 128$ through a slice of the sphere defined as

$$\mathbf{x}(\theta) = \sqrt{d} \left(\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta), 0, 0 \right), \tag{45}$$

with $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$. The level sets of both functions are in good agreement. Also shown on this figure is the projection on the slice of the position of the 64 particles on the sphere. In this result, the parameters c_i take values that are initially uniformly distributed by about $-40d^2 = -10^3$ and $40d^2 = 10^3$. To test the accuracy of the representation, we used the following Monte Carlo estimate of the loss function

$$\mathcal{L}_P[f_t^{(n)}] = \frac{1}{2P} \sum_{p=1}^{P} \left| f(\mathbf{x}_p) - f_t^{(n)}(\mathbf{x}_p) \right|^2. \tag{46}$$

which is in close analogy to the risk 2 This empirical loss function was computed with a batch of 10^6 points $\mathbf{x}_p$ uniformly distributed on the sphere. The value (46) calculated at the end of the calculation is shown as a function of n in the right panel of Fig. **??**: the empty circles show (46) for 4 individual realizations of the coefficient $a_{p,q,r}$ in (37), the full circle shows the average of (46) over these 4 realizations. The blue line scale as n^{-1}, the red one as n^{-2}: as can be seen, the empirical loss decays with n faster than n^{-1}, which is as expected.



(a) caption (b) caption

Figure 10: Left panel: Comparison between the level sets of the original function f in (37) (black dotted curves) and its approximation by the neural network in (40) with $n = 128$ and $d = 5$ in the slice defined by (45). Also shown are the projection in the slice of the particle position. Right panel: empirical loss in (46) vs n at the end of the calculation. The stars show the empirical loss for 10 independent realizations of the coefficients $a_{p,q,r}$ in (37).

3.3 Learning with Gaussian kernels

• Upon completing a preliminary reading of the [**?**] paper, my supervisor suggested that I look at the Vanden Eijnden et al paper [**?**], in particular the example with radially symmetric functions?

• Implemented gaussian kernel-approximation using SGD to 3-spin model, vanilla version and implemented an algorithm where no new sampling was necessary due to the network parameters and inputs having the same constraint. •

3.4 Week 8-10

I have implemented second-order methods, namely, Nesterov accelerated gradient descent and 'Discretized mean field SGLD with homogenization' as conceived in your paper entitled 'The sharp, the flat and the shallow: Can weakly interacting agents learn to escape bad minima?'. I also came across a paper entitled 'ACCELERATING SGD WITH MOMENTUM FOR OVER-PARAMETERIZED LEARNING' by Liu and Belkin, where the authors claim that Nesterov SGD with any parameter selection does not in general provide acceleration over ordinary SGD'. There the authors come up with a modified algorithm which they call 'Momentum-added stochastic solver (MaSS)'. I have made the following observations:

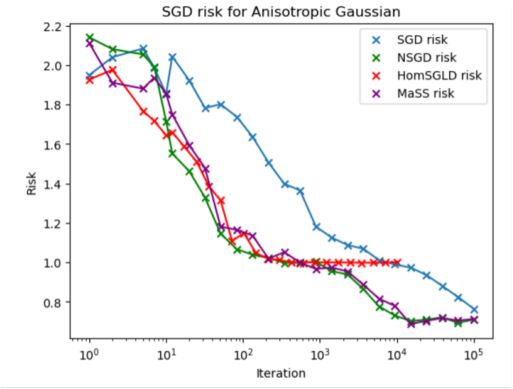
• For the Isotropic Gaussian learning problem in the 2018 paper by Mei et al. o Nesterov-accelerated GD beats plain SGD, as expected

• Anisotropic Gaussian learning problem in the 2018 paper by Mei et al. o Here Nesterov accelerated SGD performs the best, outperforming plain SGD, and while in the beginning, the MF-HomSGLD matches the performance of plain SGD, it seems to get stuck for larger iterations.

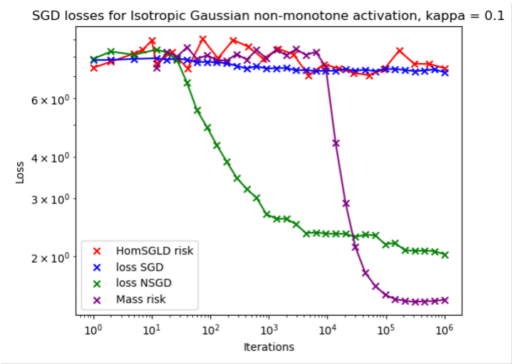
• Isotropic Gaussian with non-monotone activation learning problem in the 2018 paper by Mei et al. o Here Nesterov accelerated and plain SGD were implemented o The non-monotone activation function in the neural network introduced some non-global minima where SGD seemed to get stuck, whereas Nesterov SGD seemed to avoid such 'bad minima' and attain monotonically decreasing losses characteristic of a global minimum. MF-HomSGLD seems to take longer to converge, maybe the hyper-parameters of the algorithm are not optimally tuned. o Here The MaSS algorithm seems to outperform the Nesterov accelerated sgd only in later iterations

• 3d-spin model considered in Vanden-Eijnden's 2018 paper o In the deterministic gradient flow (with random initialization), the Nesterov accelerated loss (green) decreases towards a minimum much faster than regular gradient descent (blue), as expected

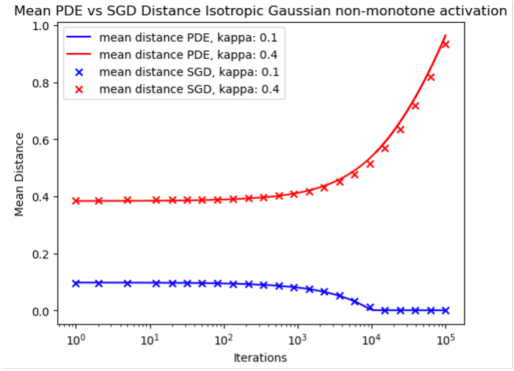
o In the stochastic gradient flow (with random initialization), the Nesterov accelerated loss (green) decreases towards a minimum faster than regular gradient descent (blue), though the losses stay closer together



(a) legend



(b) legend



(c) legend

Figure 11: my fig

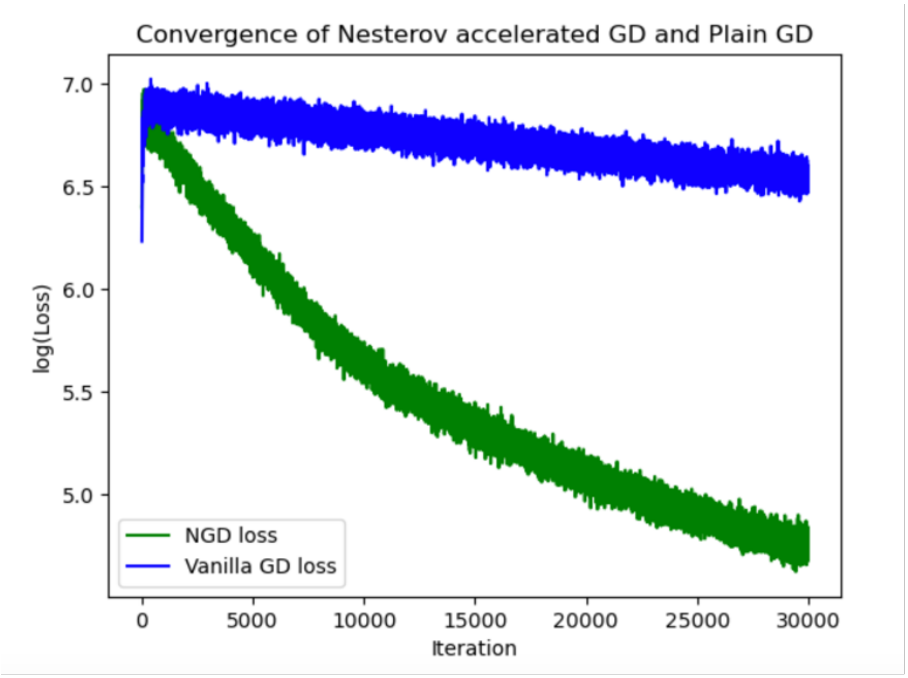
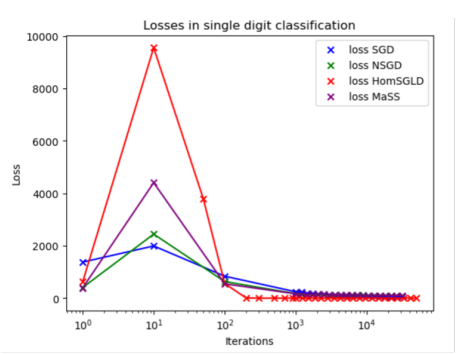
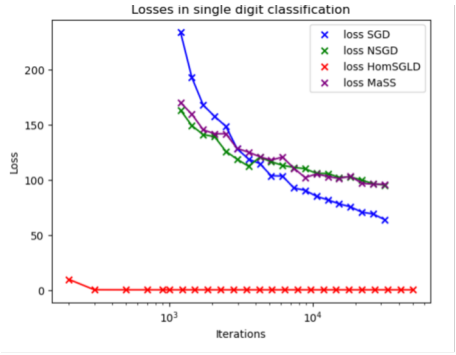


Figure 12: caption

• Single-digit classification algorithm on the MNIST dataset o Nesterov acceleration beat plain SGD, but was beaten by MF-HomSGLD, which plateaued early but achieved a substantially smaller loss in the same amount of time



(a) caption



(b) caption

Figure 13: caption

Some general observations: • For Nesterov accelerated descent: o In the deterministic case, as showcased by the 3d-spin model, Nesterov clearly beats plain GD; this is to be expected. o This acceleration is harder to see in the stochastic setting. Nesterov-accelerated SGD seems on all occasions to perform better than plain SGD, but in some cases marginally so. o The more noteworthy observation is that it gets 'unstuck' at the 'bad' minimum in the non-monotone activation case. • The MF-HomSGLD algorithm: o I chose to implement this algorithm because it corresponds to a discretisation of a gradient flow with respect to a regularized potential. o MF-HomSGLD matches the performance of plain SGD, but it seems to get stuck for larger iterations and plateaus. However in the MNIST one-digit classification, the algorithm attains a substantially smaller loss, about two orders of magnitude less than the other algorithms, but it suffers from plateauing early again. • The MaSS algorithm: o In most cases, despite claims in the paper by Liu and Belkin of exponential convergence (assuming certain regularity restraints on the loss which may be optimistic in my case), the performance at least matches Nesterov SGD, since it is a perturbation thereof. However, in the non-monotone activation function case, it beats all algorithms in later iterations and descends the fastest in loss.