# 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' [1]) 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 [9], [10] . I read part of the book entitled 'Markov Processes: Characterisation and Convergence' by Stewart N. Ethier Thomas G. Kurtz [5], 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 [11] using numerical simulations that Nesterov accelerated gradient descent escaped a 'bad minimum', where SGD got stuck in a loss function that was constructed in [7].

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' [2]. 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' [8]) 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'[3] 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?' [6] I was also led to study the analysis of multiscale algorithms in the literature, e.g. in the Weinan et al. (2005) paper [4] 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 [11] .

# 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 [7], 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 $\sigma_*(x;\theta_i) = a_i = a_i\sigma(\langle x, w_i\rangle + b_i)$ for some function $\sigma : \mathbb{R} \to \mathbb{R}$ (see figure 1).

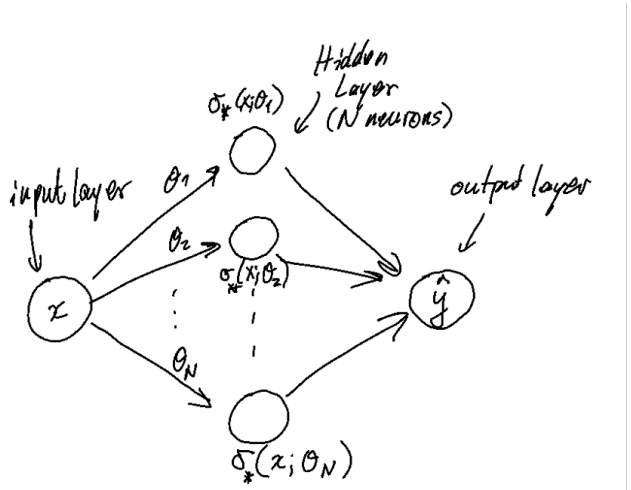



Figure 1: Illustration of a two layer neural network.

Naturally, one wants to chose parameters $\theta$ 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 $\rho_0$.
**while** loss is greater than tolerance **do**

Generate iid sample $(x, y) \sim \mathbb{P}$
**for** $1 \leqslant i \leqslant N$ **do**
$\theta_i\theta_i + 2s \cdot (y - \hat{y}(x;\theta)) \cdot \nabla_{\theta_i}\sigma_*(x;\theta_i)$ ▷ 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*:

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

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 $s_k = \cdot\xi(k)$ for some $> 0$ 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 [7], Theorem 2.1 here, it is shown that for all $t \geqq 0$, as $N \to \infty$ and $\to 0$ in an appropriate way, the empirical measures $\hat{\rho}^{(N)}_{t/}$ converge in the weak sense to some probability measure $\rho_t$ whose dynamics are governed by the following PDE, which is referred to as *distributional dynamics* (DD) in [7]

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

(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 [7]). The authors in [7] 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. [3] on Gradient Flows for Sampling and noted down some key insights from their paper(INCLUDE PAPER DETAILS - abstract)

(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 pre-conditioning 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 $\rho_t$ 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. 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).

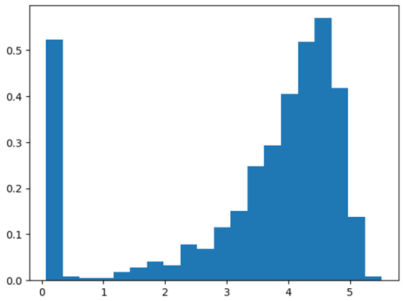
## 2.2 Centred Isotropic Gaussians

The authors in [7] 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:

$$\begin{array}{c} \text{with probability } 1/2\text{: y} = +1,\ \text{x}\sim N(0, (1+\Delta)^2 \cdot \mathrm{Id}_d) \\ \text{with probability } 1/2\text{: y} = \text{-1},\ \text{x}\sim N(0, (1-\Delta)^2 \cdot \mathrm{Id}_d) \end{array} \tag{12}$$

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

To try and reproduce the findings in [7],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 $\rho_0$ 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 $\theta$) 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.

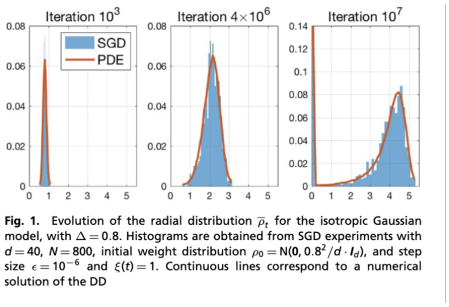



**Fig. 1.** Evolution of the radial distribution $\bar{\rho}_t$ for the isotropic Gaussian model, with $\Delta = 0.8$. Histograms are obtained from SGD experiments with $d = 40$, $N = 800$, initial weight distribution $\rho_0 = \mathsf{N}(\mathbf{0}, 0.8^2/d \cdot I_d)$, and step size $\epsilon = 10^{-6}$ 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.

• 'I think it would be useful to focus on the numerical experiments, particularly the study of the mean field limit. Mei, Montanari and Nguyen (PNAS, 115(33) 2018)' • 'Can you please also have a look at the Vanden Eijnden et al paper, in particular the example with radially symmetric functions? What I would like us to study is the possible non-uniqueness of stationary states for the mean field PDE that is derived in the paper by Mei et al.' • Managed to implement the PDE in python after optimizing the code (vectorizing, parallelizing) and obtained the following for the distributional dynamics (corresponding to $10^7$ SGD iterations): • (DD plot isotropic gaussians)

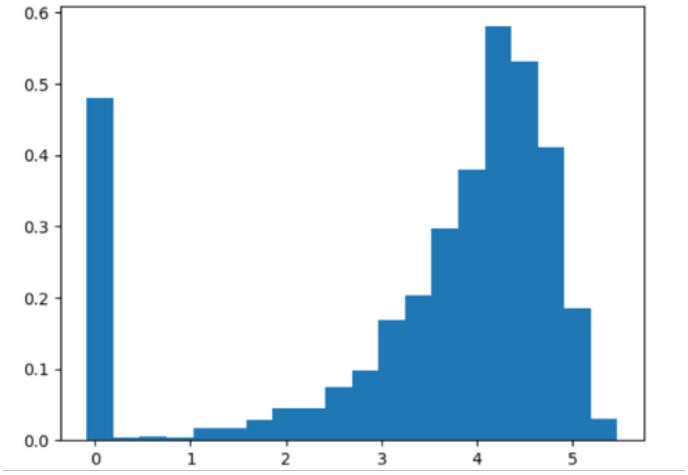
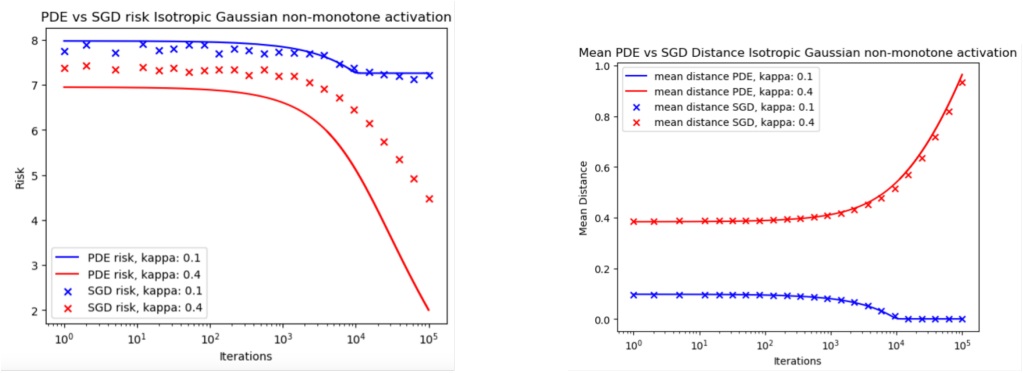



Figure 3: caption

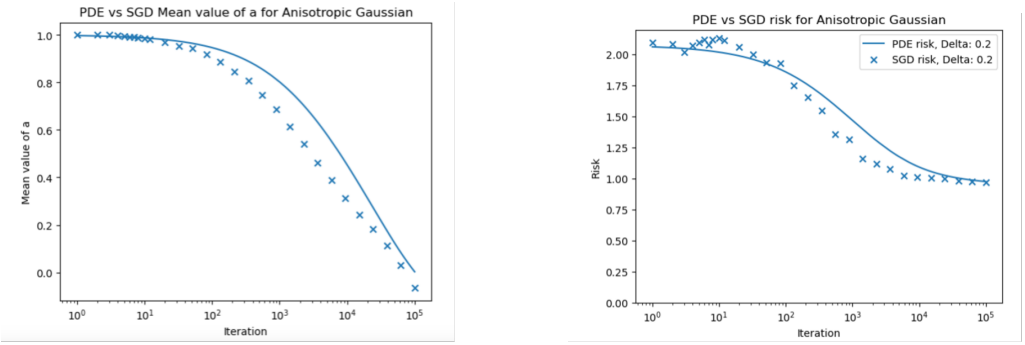
I examined numerically the case of failure of sgd given two initializations motivated by the theory developed in the 2018 paper of Mei et al. o Observed slight difference in simulated risk for initialization with kappa = 0.4




(a) caption (b) caption

Figure 4: caption

• 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 2018 paper of Mei et al. o PDE simulations reproduce the numerical findings in the paper, but sgd runs fail to match the pde profiles exactly:




(a) caption (b) caption

Figure 5: caption

• 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 [10], [9]. • Also 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 on the Skorokhod space $D_E[0, infinity)$.

I read the companion paper of Spiliopoulos (2019) where the authors proved a CLT for a one-layer neural net. My question is whether the convergence of the empirical measure, up to scaling, is proved in the dual of some Sobolev space. I imagine this is a rather weak form of convergence. • 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.

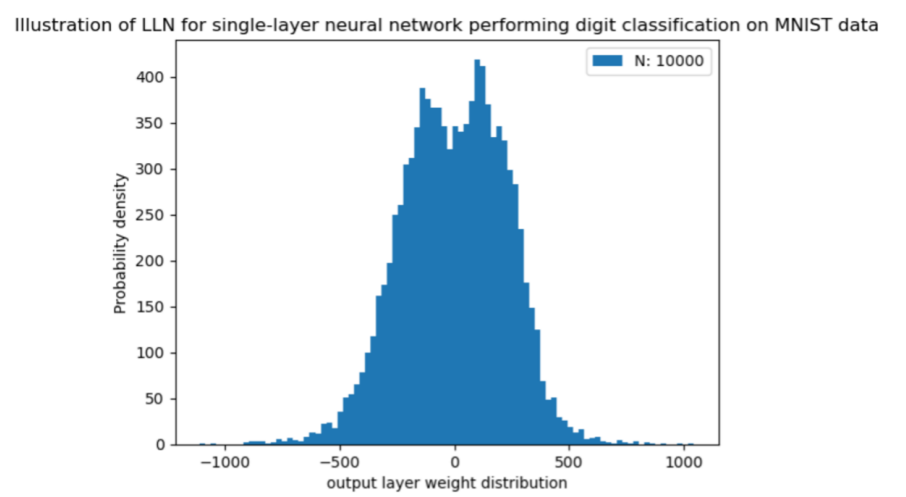



Figure 6: caption

• Does the empirical measure of the particles converging to a deterministic measure automatically yield the propagation of chaos? It seems these terms are used interchangeably, though the latter is not really elaborated upon.

# 3 Non-convex landscape

## 3.1 Approaches

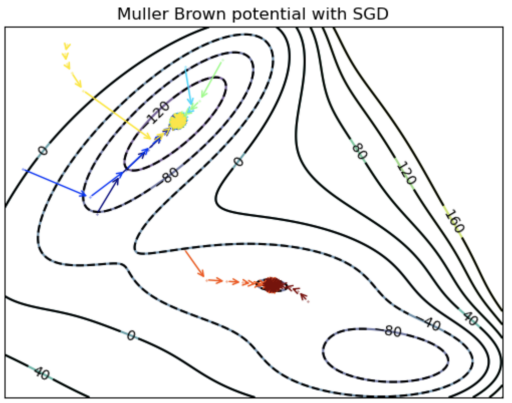
### 3.1.1 Potential regularisation

I have read your paper on shallow minima: 'The sharp, the flat and the shallow: Can weakly interacting agents learn to escape bad minima?' o I also looked at the algorithms and their justifications in the Weinan et al. (2005) paper on the analysis of multi-scale methods for SDEs o It seems to me the algorithm introduced in the paper 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 $\Phi$, using the method introduced by Chaudhari et al. (2018).

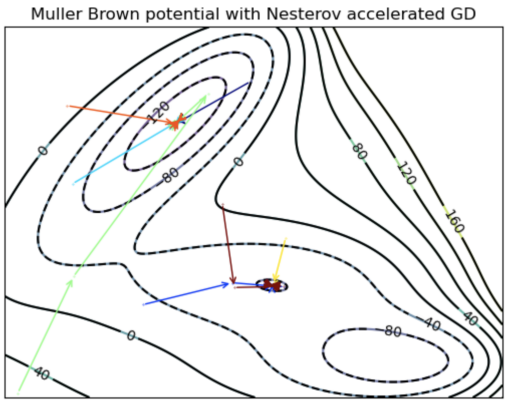
## 3.2 Applications

### 3.2.1 Muller Brown Potential

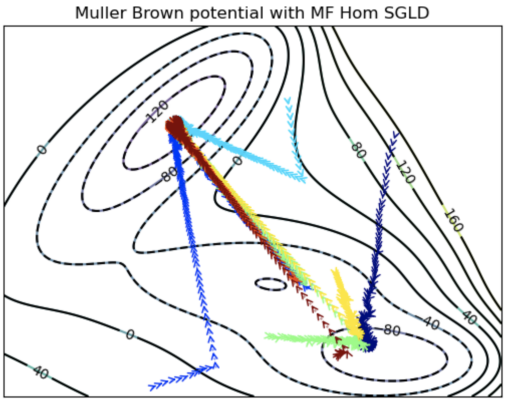
• Muller-Brown potential analysis: o - Most algorithms on the MB potential get stuck equally on two local minima, i.e. the global one (which is narrow) and the one with the next smallest local minimum o - Convolving with a solution to the heat equation does not improve performance as the narrow steep 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




(a) legend



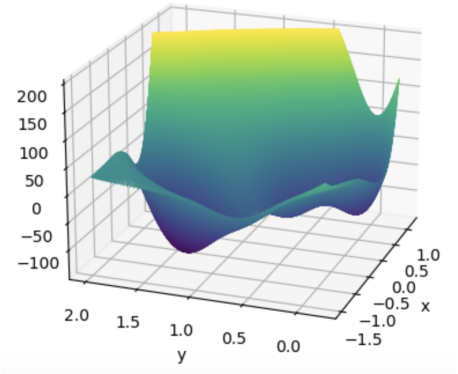

(b) legend



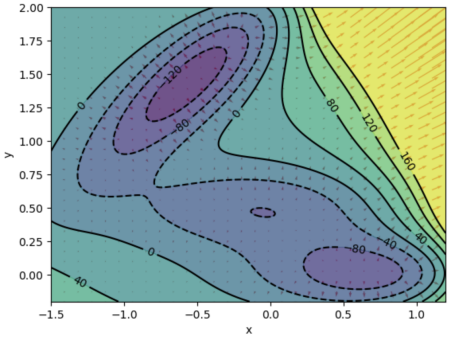

(c) legend

Figure 7: my fig

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) caption



(b) caption

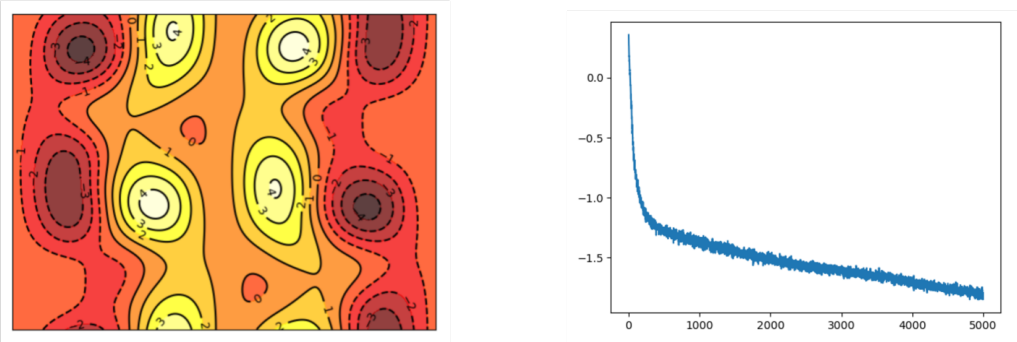
Figure 8: caption

### Nesterov SGD on Muller Brown Potential

```
import numpy as np
#Nesterov Potential
n = 256
x = np.linspace(-1.5, 1.2, n)
y = np.linspace(-0.2, 2, n)
X, Y = np.meshgrid(x, y)

intervals = np.arange(1, 1e5, 20)

ntraj = 1000
# Initialize holder for trajectories
colors = plt.cm.jet(np.linspace(0,1,np.minimum(ntraj, 7)))
minima_nesterov = []
for j in tqdm(range(ntraj)):
    points_x, points_y = train_nesterov(intervals,
    learning_rate = 1e-4, a = 1, tolerance = 1e-5)
    minima_nesterov.append(MB_potential(points_x[-1],points_y[-1]))
    if j <=6:
        plt.scatter(points_x, points_y, color = colors[j], s = 0.1)
        for i in range(len(points_x)-1):
            plt.annotate('', xy =[points_x[i+1],
            points_y[i+1]], xytext= [points_x[i], points_y[i]],
                         arrowprops={'arrowstyle': '->', 'color': colors[j],
'lw': 1},
                         va='center', ha='center')

plt.contour(X, Y, vMB_potential(X, Y).clip(max=200), 8,
alpha=.75, cmap='viridis')
C = plt.contour(X, Y, vMB_potential(X,Y).clip(max=200), 8)
plt.title('Muller-Brown potential with Nesterov accelerated GD')
plt.clabel(C, inline=1, fontsize=10)
plt.xticks([])
plt.yticks([])
#plt.legend()
```

#### 3.2.2 3−d spin model analysis

• 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. • 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). • Was not able to implement the gradient flow numerically suggested in equation (140) of the van Eijnden 2018 paper.




(a) caption (b) caption

Figure 9: caption

### 3.3 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:

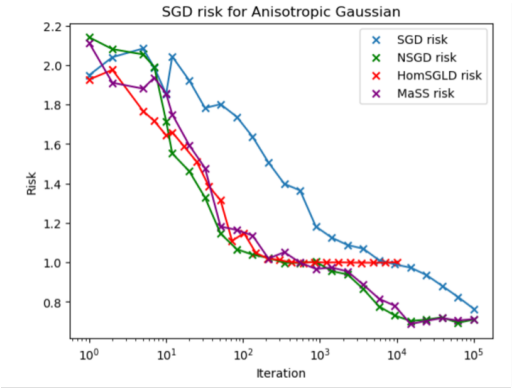
• For the Isotropic Gaussian learning problem in the 2018 paper by Mei et al. o Nesterov-accelerated GD beats plain SGD, as expected

• 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.

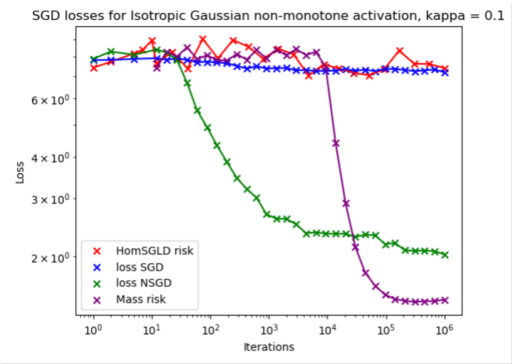
• Isotropic Gaussian with non-monotone activation learning problem in the 2018 paper by Mei et al. o Here Nesterov accelerated and plain SGD were implemented o The non-monotone activation function in the neural network introduced some non-global minima where SGD seemed to get stuck, whereas Nesterov 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

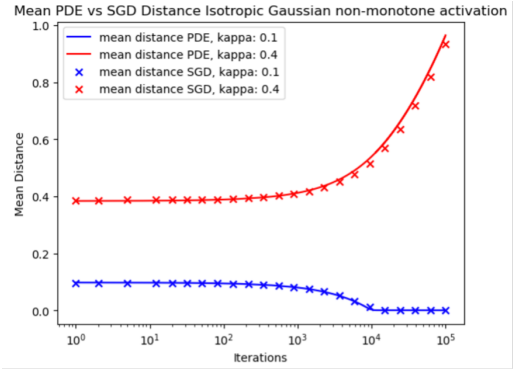
o In the stochastic gradient flow (with random initialization), the Nesterov accelerated loss (green) decreases towards a minimum faster than regular gradient descent (blue), though the losses stay closer together




(a) legend




(b) legend




(c) legend

Figure 10: my fig

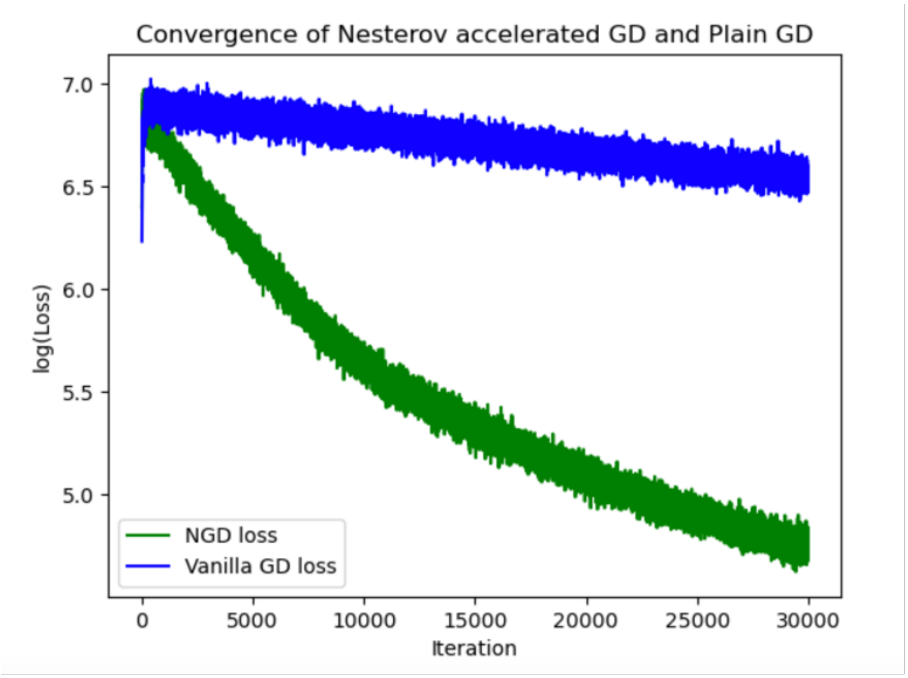
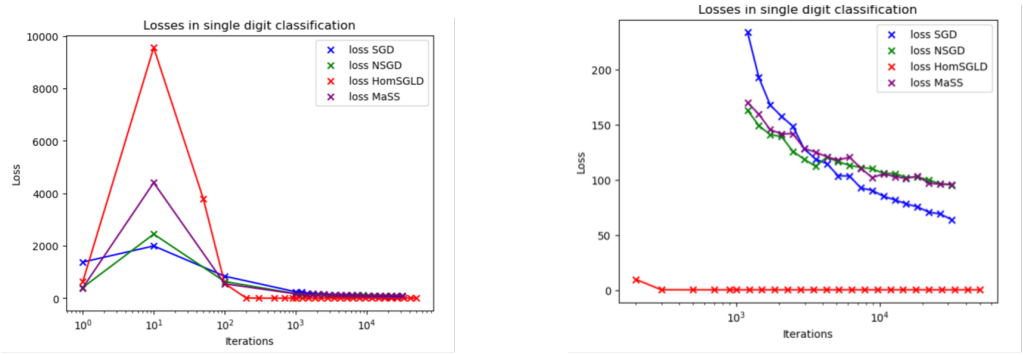



Figure 11: caption

• 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 12: 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.

# References

[1] P. Brémaud. *Probability Theory and Stochastic Processes*. Universitext. Springer International Publishing, 2020.

[2] Pratik Chaudhari, Adam M. Oberman, S. Osher, Stefano Soatto, and Guillaume Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5, 2017.

[3] Yifan Chen, Daniel Zhengyu Huang, Jiaoyang Huang, Sebastian Reich, and Andrew M. Stuart. Gradient flows for sampling: Mean-field models, gaussian approximations and affine invariance, 2023.

[4] Weinan E, Di Liu, and Eric Vanden-Eijnden. Analysis of multiscale methods for stochastic differential equations. *Communications on Pure and Applied Mathematics*, 58(11):1544–1585, November 2005.

[5] S.N. Ethier and T.G. Kurtz. *Markov Processes: Characterization and Convergence*. Wiley Series in Probability and Statistics. Wiley, 2009.

[6] Nikolas Kantas, Panos Parpas, and Grigorios A. Pavliotis. The sharp, the flat and the shallow: Can weakly interacting agents learn to escape bad minima?, 2019.

[7] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33), July 2018.

[8] Brendan O'Donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes, 2012.

[9] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A central limit theorem, 2019.

[10] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers, 2019.

[11] Pantelis Tassopoulos. Imperial Summer Research 2023 Code Repository, August.