1 General motivation

The inductive biases of deep learning optimziers are not well understood. SGD and its variants (e.g. Adam) are discrete update algorithms. At the beginning of training, the model parameters are initialized from some probability distribution (e.g. some gaussian distribution). Given some initial set of parameters, the SGD update has some stochasticity caused by sampling finite batches. These two sources of randomness means that SGD could end up in different minimizers from the same initial set of parameters. We want to characterize the distribution of the parameters during and at the end of training.

The motivation for safety is to be able to sample such distribution of parameters. If we had a good sampler of the distribution of parameters during training, we could systematically sample models that are compatible with minimizing the loss. We would have better guarantees that the sampling process is unbiased and how long it would take to sample the distribution of parameters that are accessible to the sampler. This would for example allows to determine the fraction of model minimizing the loss that are safe or unsafe and give us a lower bound on the probability of sampling a safe model in the set of degenerate solutions of the loss minimization and in particular and also help us understand the robustness of some safety property.

The simplest way to model SGD is to consider the continuous-time limit of SGD. The continuous-time limit of SGD is a stochastic differential equation (SDE) that describes the evolution of the parameters during training. Under Gaussian assumptions of the noise, this continuous-time limit leads to an SDE with a drift term and a Wiener process. Under further assuming that the noise covariance of the noise is constant, we can show that the stationary distribution of SGD obtained from Fokker-Planck is the same as the tempered Bayesian posterior distribution i.e. a Gibbs distribution with a temperature parameter.

This model of SGD has been heavily exploited in the literature, but the assumptions that are made by such models are not always valid. For example, the noise covariance is not constant (anisotropy) and SGD noise has heavy tails. Also, finite learning rate is important because of edge of stability phenomena where finite learning rate allows induced some sensitivity to the curvature in the dynamics inducing SGD to escape the sharpest minima. Subdiffusive phenomena have also been observed which can be modelled at the level of the Fokker-Planck equation by a fractional time derivative operator.

A more general model of SGD would take into account: a central flow term to model edge of stability phenomena from finite learning rate, a Levy process with alpha stable distribution to model heavy tails of the noise, a fractional time derivative operator to model subdiffusive phenomena and a non-constant noise covariance matrix to model anisotropy of the noise.

In this work we propose to bring together a more general SDE model of SGD taking into account these different effects. We study the empirical relevance of such SDE on toy models: deep linear networks, modular addition and toy transformer models. The main hypothesis that we test are wether the distribution of parameters obtained by such model better capture the training dynamics in toy models than the Langevin SDE model of SGD which leads to a tempered posterior distribution.

2 Introduction to Stochastic Gradient Descent

2.1 Problem Setup

Let $(\mathcal{X} \times \mathcal{Y}, P)$ be a probability space, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. We observe a dataset $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$ consisting of n i.i.d. samples from the distribution P.

Consider a parametric model $f : \Theta \times \mathcal{X} \to \mathcal{Y}$, where $\Theta \subseteq \mathbb{R}^d$ is the parameter space. For neural networks, $f(\theta; x)$ represents the network's output given parameters $\theta \in \Theta$ and input $x \in \mathcal{X}$.

2.2 Loss Functions

We define a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ that measures the discrepancy between predictions and targets. For regression tasks, we typically use the squared loss:

$$\ell(\hat{y}, y) = \frac{1}{2}\|\hat{y} - y\|^2$$

This induces the following hierarchy of loss functions:

1. **Individual sample loss**: For a single data point (x, y),
$$\ell(\theta; x, y) := \ell(f(\theta; x), y)$$
2. **Empirical risk** (finite-sample loss): Over the dataset $\mathcal{D}_n$,
$$L_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(\theta; X_i, Y_i)$$
3. **Population risk** (expected loss): Over the true distribution P,
$$L(\theta) := \mathbb{E}_{(X,Y)\sim P}[\ell(\theta; X, Y)]$$

In practice, we only have access to $L_n(\theta)$ and use it as a proxy for $L(\theta)$, which we truly wish to minimize.

2.3 Gradients

Define the gradients with respect to parameters θ:

1. **Individual gradient**: $g_i(\theta) := \nabla_\theta \ell(\theta; X_i, Y_i)$
2. **Empirical gradient**: $g_n(\theta) := \nabla L_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} g_i(\theta)$
3. **Stochastic gradient**: For a batch $\mathcal{B} \subset D_n$ of size B sampled uniformly at random,
$$g_{\mathcal{B}}(\theta) := \frac{1}{B}\sum_{i\in\mathcal{B}} g_i(\theta)$$
4. **Population gradient**: $g(\theta) := \nabla L(\theta) = \mathbb{E}_{(X,Y)\sim P}[\nabla_\theta \ell(\theta; X, Y)]$

Note that $\mathbb{E}_{\mathcal{B}}[g_{\mathcal{B}}(\theta)] = \nabla L_n(\theta)$, making the stochastic gradient an unbiased estimator of the empirical gradient.

2.4 The SGD Algorithm

Stochastic Gradient Descent (SGD) performs the following iterative update:
where $\eta_k > 0$ is the learning rate (or step size) at iteration k.

2.5 Mini-batch Sampling Strategies

There are two common sampling strategies:

1. **With replacement**: Each element of $\mathcal{B}_k$ is drawn independently and uniformly from $\{1, ..., n\}$
2. **Without replacement**: Sample B distinct indices uniformly from $\{1, ..., n\}$

The with-replacement case is often easier to analyze theoretically due to independence.

Algorithm 1 Stochastic Gradient Descent

```
Initialize $\theta_0 \in \Theta$
for $k = 0, 1, 2, ..., K - 1$ do
    Sample mini-batch $\mathcal{B}_k \subseteq D_n$ uniformly at random
    Compute stochastic gradient $g_{\mathcal{B}_k}(\theta_k)$
    Update: $\theta_{k+1} = \theta_k - \eta_k g_{\mathcal{B}_k}(\theta_k)$
end for
```

2.6 Decomposition into Drift and Noise

It is useful to decompose the SGD update as:

$$\theta_{k+1} = \theta_k - \eta_k g_n(\theta_k) + \eta_k \xi(\theta_k)$$

where the **gradient noise** is defined as:

$$\xi(\theta_k) := g_n(\theta_k) - g_{\mathcal{B}_k}(\theta_k)$$

This yields a decomposition into:

- **Drift**: $\eta_k g_n(\theta_k)$ (deterministic gradient)
- **Diffusion**: $\eta_k \xi(\theta_k)$ (stochastic perturbation)

2.7 Special Cases

1. **Full-batch gradient descent**: $\mathcal{B}_k = \{1, ..., n\}$, so $\xi_k = 0$ (deterministic)
2. **Single-sample SGD**: $|\mathcal{B}_k| = 1$, maximum noise
3. **Constant learning rate**: $\eta_k = \eta$ for all k
4. **Decaying learning rate**: $\eta_k \to 0$ as $k \to \infty$ (e.g., $\eta_k = \eta_0/\sqrt{k+1}$)

"'latex

2.8 Properties of the Gradient Noise

The stochasticity in SGD arises from two distinct sources of randomness, which we now analyze.

2.8.1 Two Sources of Gradient Noise

1. Sampling Noise (Empirical vs. Population)

The first source of randomness comes from using a finite dataset. Even if we compute the full empirical gradient $g_n(\theta)$, it differs from the true population gradient $g(\theta)$:

$$\nu_{\text{sampling}}(\theta) := g_n(\theta) - g(\theta) = \frac{1}{n}\sum_{i=1}^{n} g_i(\theta) - \mathbb{E}_{(X,Y)\sim P}[\nabla_\theta \ell(\theta; X, Y)]$$

Since the data points are i.i.d. samples from P, we have:

$$\mathbb{E}_{\mathcal{D}_n}[\nu_{\text{sampling}}(\theta)] = \mathbb{E}_{\mathcal{D}_n}\left[\frac{1}{n}\sum_{i=1}^{n} g_i(\theta)\right] - g(\theta) = 0 \tag{1}$$

The covariance is:

$$\operatorname{Cov}_{\mathcal{D}_n}[\nu_{\text{sampling}}(\theta)] = \operatorname{Cov}_{\mathcal{D}_n}\left[\frac{1}{n}\sum_{i=1}^{n} g_i(\theta)\right] \tag{2}$$

$$= \frac{1}{n^2}\sum_{i=1}^{n} \operatorname{Cov}_{(X_i,Y_i)}[g_i(\theta)] \tag{3}$$

$$= \frac{1}{n}\Sigma_{\text{pop}}(\theta) \tag{4}$$

where $\Sigma_{\text{pop}}(\theta) := \operatorname{Cov}_{(X,Y)\sim P}[\nabla_\theta \ell(\theta; X, Y)]$ is the population gradient covariance.
2. Mini-batch Noise (Empirical vs. Batch)
The second source of randomness comes from using mini-batches instead of the full dataset:

$$\nu_{\text{batch}}(\theta) := g_{\mathcal{B}}(\theta) - g_n(\theta) = \frac{1}{B}\sum_{i\in\mathcal{B}} g_i(\theta) - \frac{1}{n}\sum_{i=1}^{n} g_i(\theta)$$

For uniform sampling with replacement, conditioning on the dataset $\mathcal{D}_n$:

$$\mathbb{E}_{\mathcal{B}}[\nu_{\text{batch}}(\theta)|\mathcal{D}_n] = \mathbb{E}_{\mathcal{B}}\left[\frac{1}{B}\sum_{i\in\mathcal{B}} g_i(\theta)\right] - g_n(\theta) = 0 \tag{5}$$

The conditional covariance is:

$$\operatorname{Cov}_{\mathcal{B}}[\nu_{\text{batch}}(\theta)|\mathcal{D}_n] = \operatorname{Cov}_{\mathcal{B}}\left[\frac{1}{B}\sum_{i\in\mathcal{B}} g_i(\theta)\middle|\mathcal{D}_n\right] \tag{6}$$

$$= \frac{1}{B}\Sigma_{\text{emp}}(\theta, \mathcal{D}_n) \tag{7}$$

where $\Sigma_{\text{emp}}(\theta, \mathcal{D}_n) := \frac{1}{n}\sum_{i=1}^{n}[g_i(\theta) - g_n(\theta)][g_i(\theta) - g_n(\theta)]^T$ is the empirical gradient covariance.

2.8.2 Total Gradient Noise

The total noise in SGD is:

$$\xi_{\text{total}}(\theta) := g_{\mathcal{B}}(\theta) - g(\theta) = \underbrace{g_{\mathcal{B}}(\theta) - g_n(\theta)}_{\nu_{\text{batch}}(\theta)} + \underbrace{g_n(\theta) - g(\theta)}_{\nu_{\text{sampling}}(\theta)}$$

However, in the SGD algorithm as defined, we use:

$$\xi(\theta) := g_n(\theta) - g_{\mathcal{B}}(\theta) = -\nu_{\text{batch}}(\theta)$$

This is because we can only compute deviations from the empirical gradient, not the unknown population gradient.

2.8.3 Covariance Structure of SGD Noise

For the noise $\xi(\theta) = g_n(\theta) - g_{\mathcal{B}}(\theta)$ used in SGD:

Proposition 1. *Under uniform mini-batch sampling with replacement, the gradient noise has the following properties:*

1. **Unbiasedness**: $\mathbb{E}_{\mathcal{B}}[\xi(\theta)|\mathcal{D}_n, \theta] = 0$

2. **Covariance**:
$$\mathrm{Cov}_{\mathcal{B}}[\xi(\theta)|\mathcal{D}_n,\theta] = \frac{1}{B}\Sigma_{\mathrm{emp}}(\theta,\mathcal{D}_n)$$

3. **Expected Covariance**: Taking expectation over datasets,
$$\mathbb{E}_{\mathcal{D}_n}[\mathrm{Cov}_{\mathcal{B}}[\xi(\theta)|\mathcal{D}_n,\theta]] = \frac{1}{B}\cdot\frac{n-1}{n}\Sigma_{\mathrm{pop}}(\theta)$$

Proof.

1. Follows immediately from $\mathbb{E}_{\mathcal{B}}[g_{\mathcal{B}}(\theta)] = g_n(\theta)$.

2. For sampling with replacement:
$$\mathrm{Cov}_{\mathcal{B}}[\xi(\theta)|\mathcal{D}_n] = \mathrm{Cov}_{\mathcal{B}}[g_{\mathcal{B}}(\theta)|\mathcal{D}_n] \tag{8}$$
$$= \mathrm{Var}_{\mathcal{B}}\left[\frac{1}{B}\sum_{j=1}^{B} g_{I_j}(\theta)\middle|\mathcal{D}_n\right] \tag{9}$$
where I_j are i.i.d. uniform on $\{1,...,n\}$. Since the I_j are independent:
$$= \frac{1}{B^2}\sum_{j=1}^{B}\mathrm{Var}_{I_j}[g_{I_j}(\theta)|\mathcal{D}_n] \tag{10}$$
$$= \frac{1}{B}\cdot\frac{1}{n}\sum_{i=1}^{n}[g_i(\theta)-g_n(\theta)][g_i(\theta)-g_n(\theta)]^T \tag{11}$$

3. Using the identity $\mathbb{E}[\Sigma_{\mathrm{emp}}] = \frac{n-1}{n}\Sigma_{\mathrm{pop}}$ completes the proof.

2.8.4 Key Observations

1. **Batch size scaling**: The noise covariance scales inversely with batch size B. Larger batches reduce noise.

2. **Position dependence**: The covariance $\Sigma_{\mathrm{emp}}(\theta,\mathcal{D}_n)$ depends on the current parameters θ, making the noise *multiplicative* rather than additive.

3. **Finite sample correction**: The factor $\frac{n-1}{n}$ in the expected covariance reflects the finite dataset size. As $n \to \infty$, we recover $\mathbb{E}[\mathrm{Cov}[\xi]] = \frac{1}{B}\Sigma_{\mathrm{pop}}(\theta)$.

4. **Sampling without replacement**: If we sample without replacement, the covariance becomes:
$$\mathrm{Cov}_{\mathcal{B}}[\xi(\theta)|\mathcal{D}_n] = \frac{n-B}{n-1}\cdot\frac{1}{B}\Sigma_{\mathrm{emp}}(\theta,\mathcal{D}_n)$$
The factor $\frac{n-B}{n-1}$ represents the finite population correction.

2.8.5 Estimating the Noise Covariance

In practice, we can estimate $\Sigma_{\mathrm{emp}}(\theta,\mathcal{D}_n)$ by:

1. Computing multiple stochastic gradients $\{g_{\mathcal{B}_j}(\theta)\}_{j=1}^m$ at the same θ

2. Estimating: $\hat{\Sigma} = \frac{B}{m-1}\sum_{j=1}^m[g_{\mathcal{B}_j}(\theta)-\bar{g}][g_{\mathcal{B}_j}(\theta)-\bar{g}]^T$

where $\bar{g} = \frac{1}{m}\sum_{j=1}^m g_{\mathcal{B}_j}(\theta)$.

This characterization of gradient noise is fundamental for understanding the continuous-time limit of SGD, as the noise covariance $\Sigma(\theta)$ directly determines the diffusion coefficient in the limiting stochastic differential equation.

3 Continuous-time limit of SGD

3.1 Heuristic derivation of the continuous-time limit

Consider the SGG update:

$$\theta_{k+1} = \theta_k - \eta_k g_{\mathcal{B}_k}(\theta_k) \tag{12}$$

where $g_{\mathcal{B}_k}(\theta_k)$ is the gradient at the k-th iteration for batch $\mathcal{B}_k$. Define the gradient noise as:

$$\xi(\theta_k) := g_n(\theta_k) - g_{\mathcal{B}_k}(\theta_k) \tag{13}$$

where $g_n(\theta_k)$ is the empirical gradient. The new update can be written as:

$$\Delta\theta_{k+1} := \theta_{k+1} - \theta_k = -\eta_k g_{\mathcal{B}_k}(\theta_k) + \eta_k \xi(\theta_k) \tag{14}$$

The 1-sample gradient expectation is:

$$\mathbb{E}[g_i(\theta_k)] = \frac{1}{n}\sum_{i=1}^{n} g_i(\theta_k) = g_n(\theta_k) \tag{15}$$

Hence the expectation of the noise is zero. The 1-sample noise covariance is:

$$\Sigma(\theta_k) := \frac{1}{n}\sum_{i=1}^{n}[g_i(\theta_k) - g_n(\theta_k)][g_i(\theta_k) - g_n(\theta_k)]^T \tag{16}$$

where $g_i(\theta_k)$ is the gradient at the i-th data point. The batch noise covariance is:

$$\Sigma_{\text{batch}}(\theta_k) := \frac{1}{B^2}\sum_{i\in\mathcal{B}_k} Cov(\xi(\theta_k)) = \frac{1}{B}\Sigma(\theta_k) \tag{17}$$

Assume that the learning rate is independent of k. Assume that batch size is large enough to apply CLT and much smaller than the dataset size i.e. $B \ll n$ (**independent batches?**) and $\frac{B}{n} \ll 1$. Assume that number of samples is large enough to apply CLT. Assume that samples are i.i.d. and that the gradient update (and maybe gradient noise?) have finite variance. Under these assumptions, we can apply the CLT to the batch gradient and get:

$$\xi(\theta_k) \sim \mathcal{N}(0, \frac{1}{B}\Sigma(\theta_k)) \tag{18}$$

Proper scaling of the noise is crucial to apply CLT and get the continuous-time limit. Physical intuition: the average displacement of the noise is 0 but the the typical displacement of the noise is in $\sqrt{t}$ where $t = N\eta$ for N steps. More specifically

$$Var(\sum_{k=1}^{N}\eta\xi_k) = N\eta^2\Sigma(\theta_k) = t\eta\Sigma(\theta_k) \tag{19}$$

$$std(\sum_{k=1}^{N}\eta\xi_k) = \sqrt{t\eta}\sqrt{\Sigma(\theta_k)} \tag{20}$$

with $N = \frac{t}{\eta}$ for N steps. We see that as $\eta \to 0$, the typical displacement of the accumulated noise goes to 0. If we use the scaling $\sqrt(\eta)$ for the noise, we get:

$$Var(\sum_{k=1}^{N}\sqrt(\eta)\xi_k) = N\eta\Sigma(\theta_k) = t\Sigma(\theta_k) \tag{21}$$

$$std(\sum_{k=1}^{N}\sqrt(\eta)\xi_k) = \sqrt{t}\sqrt{\Sigma(\theta_k)} \tag{22}$$

With $N = t/\eta$ for N steps. We see that the noise as proper scaling as the typical displacement is in $\sqrt{t}$ and does not depends on η. The continuous-time limit of SGD is:

$$d\theta_t = -\nabla L_n(\theta_t)dt + \sqrt{\frac{\eta}{B}\Sigma(\theta_t)}dW_t \tag{23}$$

where W_t is a Wiener process. Note that a more rigorourous derivation would use the Lindeberg condition and the Donsker's theorem.

3.2 Why do we care: the tempered posterior distribution

Using the machinery of Fokker-Planck equation, we can derive the stationary distribution of the SGD process as the tempered posterior distribution. The Fokker-Planck equation is:

$$\frac{\partial p(\theta, t)}{\partial t} = \nabla \cdot \left(\nabla L_N(\theta) p(\theta, t) + \frac{1}{2}\frac{\eta}{B}\Sigma(\theta)\nabla p(\theta, t) \right) \tag{24}$$

Assume that the noise covariance matrix is positive definite. Let the probability current be:

$$j(\theta, t) = \nabla L_N(\theta) p(\theta, t) + \frac{1}{2}\frac{\eta}{B}\Sigma(\theta)\nabla p(\theta, t) \tag{25}$$

The stationarity condition implies that the probability current is divergence free:

$$\nabla \cdot j(\theta, t) = 0 \tag{26}$$

Assume a detailed balance condition i.e. $j(\theta, t) = 0$. This implies that the stationary distribution satisfies:

$$\nabla L_N(\theta) p(\theta) + \frac{1}{2}\frac{\eta}{B}\Sigma(\theta)\nabla p(\theta) = 0 \tag{27}$$

Assume that the noise covariance matrix is constant $\Sigma(\theta) = \Sigma$, positive and definite. Then solving the latter differential equation gives us the stationary distribution as the tempered posterior distribution:

$$p_\infty(\theta) = \frac{1}{Z}\exp\left(-\beta\Sigma^{-1}L_N(\theta)\right) \tag{28}$$

where $\beta = \frac{B}{2\eta}$ is the inverse temperature. In the case where the noise covariance matrix is not constant but still positive and definite, we get:

$$p_\infty(\theta) = \frac{1}{Z}|\Sigma(\theta)|^{-1/2}\exp\left(-\frac{2B}{\eta}L_N(\theta)\right) \tag{29}$$

where Z is the partition function. In that case, the stationary distribution is given by the tempered posterior i.e. up to the temperature parameter SGD is similar to Bayesian inference. This result is been investigated in more details by Mandt. The assumptions in this derivation are not realistic since the noise covariance is singular i.e. not definite positive. In fact, deep neural networks are highly degenerate and we also know that SGD favour flat minima.

3.3 List of assumptions

Assumptions for the continuous-time limit of SGD:

- GN satisfies the Lindeberg condition: necessary for CLT to apply. Unclear how much this matters.

- Gradient noise has finite variance: necessary for CLT to apply. An alternative would be to consider heavy tailed noise. It is unclear how much this matters. This is debated in the literature and it would be useful to understand better how much this matters by reviewing the literature.

- η_k is independent of k: Uncertain but technically should not be necessary as we could have $\eta_k \to dt$ after some updates and then take the continuous-time limit if we only care about the local behaviour of SGD. Unclear how much this matters, in practice it is fairly common to have time-varying learning rate.

- $\eta \to dt \ll 1$: makes the learning rate small which is central to take the continuous-time limit. This is an important difference with the practical usage of SGD. Although it might be possible to approximate the discrete time dynamics with a continuous time dynamics by using a central flows which essentially add the average jittered caused by the discrete learning rate to the continuous time dynamics. See for example this paper: Understanding Optimization in Deep Learning with Central Flows .

- $B \ll n$ and $\frac{B}{n} \ll 1$: Have enough batch and data samples to apply CLT

- No auto-correlation in the noise: makes the noise white which enables to take the continuous-time limit. Alternative is having coloured noise. This could happens for example if the batches are not independent, which could be the case, intuitively if the batches, are large. Another example is if the sampling of the batches is not without replacement. In this case it seems that the noise is anti-correlated as argued in this paper. Anti-correlation could be important to generalization as the latter paper argue that it biases SGD further towards flat minima. Momentum can also introduces auto-correlation in the noise. If the noise has some auto-correlation but decaying with time, the functional CLT still can be applied and we can get to the continuous-time with a Wiener process. If not we have to consider a fractional Brownian motion. **Idea**: Could we have a fractional Levy process with heavy tail noise and long-range auto-correlation? Is it the case that in practice long-range auto-correlation matter? **Note**: Fractional Brownnian or Levy process could be another connection with fractality in the training process. **Question**: if heavy tailed noise is considered, does it make sense to consider auto-correlation in the noise?

- Samples are i.i.d.: necessary for CLT to apply

- For now we are not considering momentum. Unsure how it changes the analysis. For example our analysis contrasts with Adam which has a momentum and a second order correction of past gradient.

Assumptions for the stationary distribution of SGD being the tempered posterior distribution:

- The noise covariance matrix is positive and definite: seems important to solve the Fokker-Planck equation as we invert the noise covariance matrix to get the stationary distribution. This assumption is violated in practice since deep neural networks are highly degenerate.

- The noise covariance matrix is constant: this is not necessary but makes the calculation easier.

- Detailed balance condition: helps with getting a first ODE for the stationary distribution which enables to get the tempered posterior distribution.

- According to the Helmoltz decomposition, the current can be decomposed into a curl-free part and a divergence-free part. If the noise is anistoropic then we have $\nabla \times j(\theta, t) \neq 0$ In other words, the current is not curl-free. The detailed balanced condition does not hold in that condition. When the diffusion matrix $\Sigma(\theta)$ depends on θ, the stationary dynamics of SGD is an *out-of-equilibrium* steady state with a non-zero *solenoidal probability current*. Probability circulates in parameter space rather than remaining static, so detailed balance—and the simple Boltzmann form $p \propto e^{-\beta L_N}$—no longer hold. **Note**: I am still a bit confused about the physical intuition about this. How should i think about this probability mass orbiting along some level set of the loss?

3.4 Ranking assumptions

After a deep research with chatGPT here and with elicit here this is my current ranking of the assumptions:

- Application of CLT from assuming gaussian gradient noise (finite variance in particular, likely that noise is not gaussian but Levy). Also an interesting point is that the heavy tailness could be anisotropic i.e. be sensitive to the degeneracies in the loss landscape.
- Detailed balance condition from constant noise covariance. In particular this means we must consider a curl component to the probability current.
- **Note**: I am a bit confused about the ranking of the next two items.
- Small learning rate: irrealistic but see central flows paper. Would be useful to look more into the edge of stability phenomena literature.
- No auto-correlation in the noise: this will be violated with momentum in practice (but maybe that's ok as long as it's not long-range auto-correlation?) **Note**: I am a bit confused about this one. I'd like to understand better violation of long-range auto-correlation. Modification might include: fractional Levy, generalized Langevin dynamics, etc. Need colored noise moodels with memory. Sampling without replacement is common introducing some auto-correlation.
- Batch size large enough but small compared with the dataset size seems realistic especially in large scale training.

4 SDE on DLNs

4.1 Gradients

For input–output pairs (x^μ, y^μ) and weight matrices $W^1 \in \mathbb{R}^{d_1 \times d_0}$, $W^2 \in \mathbb{R}^{d_2 \times d_1}$, let the empirical (squared–error) loss be

$$L_N(W^1, W^2) \;=\; \frac{1}{N}\sum_{\mu=1}^{N} \big\| \, y^\mu - W^2 W^1 x^\mu \big\|_2^2.$$

Expanded form. With Einstein summation over repeated indices (i, j, k, l, p, r),

$$\boxed{L_N = \frac{1}{N}\sum_{\mu=1}^{N}\Big(y_i^\mu y_i^\mu - 2\, y_i^\mu W_{ik}^2 W_{kj}^1 x_j^\mu + x_l^\mu W_{kl}^1 W_{ki}^2 W_{ip}^2 W_{pr}^1 x_r^\mu\Big)}$$

Gradient w.r.t. the first layer W^1.

$$\boxed{g^1_{mn} = \frac{\partial L_N}{\partial W^1_{mn}} = \frac{2}{N}\sum_{\mu=1}^{N} x^\mu_n \, W^2_{im}\big(W^2_{ik}W^1_{kj}x^\mu_j - y^\mu_i\big)}$$

Matrix form: $Y = [y^1 \dots y^N]$):

$$\boxed{g^1 = \nabla_{W^1} L_N = \frac{2}{N}\, W^{2\top} \sum_{\mu=1}^{N}\big(W^2 W^1 x^\mu x^{\mu T} - y^\mu x^{\mu T}\big)}$$

Gradient w.r.t. the second layer W^2.

$$\boxed{g^2_{mn} = \frac{\partial L_N}{\partial W^2_{mn}} = \frac{2}{N}\sum_{\mu=1}^{N}\big(W^2_{mk}W^1_{kj}x^\mu_j - y^\mu_m\big)\big(W^1_{nj}x^\mu_j\big)}$$

Matrix form:

$$\boxed{g^2 = \nabla_{W^2} L_N = \frac{2}{N}\sum_{\mu=1}^{N}\big(W^2 W^1 x^\mu x^{\mu T} - y^\mu x^{\mu T}\big) W^{1\top}}$$

Define the data and input-output covariance matrices

$$\Sigma^\mu_x = x^\mu x^{\mu T} \tag{30}$$

$$\Sigma^\mu_{xy} = y^\mu x^{\mu T} \tag{31}$$

$$\tag{32}$$

Then the gradients write

$$g^1 = \frac{2}{N} W^{2T} \sum_{\mu=1}^{N}\big(W^2 W^1 \Sigma^\mu_x - \Sigma^\mu_{xy}\big) \tag{33}$$

$$g^2 = \frac{2}{N}\sum_{\mu=1}^{N}\big(W^2 W^1 \Sigma^\mu_x - \Sigma^\mu_{xy}\big) W^{1T} \tag{34}$$

$$\tag{35}$$

The zero condition of the gradients give $d_0 d_1 + d_1 d_2$ quadratic equations with $d_0 d_1 + d_1 d_2 + d_0 d_2$ unknowns. Since there is a common factor among the equations we should expect the set of critical points to be infinite.

4.2 Noise covariance

We consider the 1-sample gradient noise covariance matrix:

$$\Sigma = \frac{1}{N}\sum_{\mu=1}^{N}(g_N - g_\mu)(g_N - g_\mu)^T \tag{36}$$

i.e.

$$\Sigma^{ij} = \frac{1}{N}\sum_{\mu=1}^{N}(g_N - g^i_\mu)(g_N - g^j_\mu)^T \tag{37}$$

$$\tag{38}$$

Abbreviations.

$$r^\mu := W^2 W^1 x^\mu - y^\mu, \qquad h^\mu := W^1 x^\mu, \qquad Z^\mu := r^\mu x^{\mu\top} \in \mathbb{R}^{d_2 \times d_0}, \qquad \bar{Z} := \frac{1}{N}\sum_{\nu=1}^{N} Z^\nu.$$

Per–sample and mean gradients.

$$\begin{aligned} g^1_\mu &= 2\, W^{2\top} Z^\mu, & g^1_N &= \frac{2}{N} W^{2\top} \sum_\nu Z^\nu = 2\, W^{2\top} \bar{Z}, \\ g^2_\mu &= 2\, r^\mu h^{\mu\top} = 2\, Z^\mu W^{1\top}, & g^2_N &= \frac{2}{N} \sum_\nu Z^\nu W^{1\top} = 2\, \bar{Z} W^{1\top}. \end{aligned}$$

Centred gradients.

$$\Delta^1_\mu := g^1_\mu - g^1_N = 2\, W^{2\top}(Z^\mu - \bar{Z}), \qquad \Delta^2_\mu := g^2_\mu - g^2_N = 2\,(Z^\mu - \bar{Z}) W^{1\top}.$$

Vectorisation trick. Let vec stack matrix columns and recall $\operatorname{vec}(A\,B) = (I \otimes A)\operatorname{vec} B$ and $\operatorname{vec}(B\,A) = (A^\top \otimes I)\operatorname{vec} B$. Define

$$\tilde{z}^\mu := \operatorname{vec}\big(Z^\mu - \bar{Z}\big), \qquad C := \frac{1}{N}\sum_{\mu=1}^{N} \tilde{z}^\mu \tilde{z}^{\mu\top} \in \mathbb{R}^{(d_2 d_0)\times(d_2 d_0)}.$$

Then

$$\operatorname{vec}(\Delta^1_\mu) = 2\,(I_{d_0} \otimes W^{2\top})\tilde{z}^\mu, \qquad \operatorname{vec}(\Delta^2_\mu) = 2\,(W^1 \otimes I_{d_2})\tilde{z}^\mu.$$

Gradient–noise covariance. Stack $\operatorname{vec}(\Delta^1_\mu)$ over $\operatorname{vec}(\Delta^2_\mu)$ and average:

$$\Sigma = \frac{1}{N}\sum_{\mu=1}^{N} \begin{bmatrix} \operatorname{vec}(\Delta^1_\mu) \\ \operatorname{vec}(\Delta^2_\mu) \end{bmatrix} \begin{bmatrix} \operatorname{vec}(\Delta^1_\mu) \\ \operatorname{vec}(\Delta^2_\mu) \end{bmatrix}^\top.$$

Insert the vectorised forms:

$$\boxed{\Sigma = 4 \begin{bmatrix} I_{d_0} \otimes W^{2\top} \\ W^1 \otimes I_{d_2} \end{bmatrix} C \begin{bmatrix} I_{d_0} \otimes W^{2\top} \\ W^1 \otimes I_{d_2} \end{bmatrix}^\top}$$

Hence the four blocks are

$$\begin{aligned} \Sigma^{11} &= 4\,(I_{d_0} \otimes W^{2\top})\, C\, (I_{d_0} \otimes W^{2\top})^\top, \\ \Sigma^{22} &= 4\,(W^1 \otimes I_{d_2})\, C\, (W^1 \otimes I_{d_2})^\top, \\ \Sigma^{12} &= 4\,(I_{d_0} \otimes W^{2\top})\, C\, (W^1 \otimes I_{d_2})^\top, \qquad \Sigma^{21} = \Sigma^{12\top}. \end{aligned}$$

Interpretation. *The entire gradient-noise covariance is obtained from the* outer-product covariance C of the centred matrices $Z^\mu = r^\mu x^{\mu\top}$ *and two simple linear maps* $I_{d_0} \otimes W^{2\top}$ *and* $W^1 \otimes I_{d_2}$.

4.3 Hessian

4.4 SDE

4.5 Plan

Motivation is to look into different assumptions about SDEs on DLNs. We can try to derive analytic results on DLNs. For this we need to compute the following:

- Find critical points of the loss functions
- Compute the Covariance of the Gradient Noise
- Compute the Hessian of the loss function
- Compute the stationary distribution of the Langevin SDE of the DLNs
- Compute the noise covariance matrix of SGD
- Compute escape time around saddle points
- Compute mixing time around saddle points

5 Empirical results on DLNs

Qualitative results that we want to get on DLNs: Some stuff around the gradient noise and finite learning rate:

- Gradient noise is anisotropic.
- Gradient noise is heavy tailed.
- Finite learning rate capture some different dynamics than the continuous limit (edge of stability) e.g. by changing escape times and mixing times.
- Gradient noise is correlated with the Hessian of the loss function and the loss function itself. When does this correlation break down?

Around non-strict saddle points:

- Check Escape time: look into extention of Kramers rate theory. Rates might follow a power law for ex.
- Check mixing time along degenerate directions: can we find the stationary distribution of the SDE and compare it with the distribution of SGD?
- Look at transitions between different saddle points: estimate probability of escaping from one saddle point to another in a given time, get some transition Kernel.
- Higher order statistics of SGD trajectories.
- Can look at SDEs at different scales i.e. different distances from the saddles points. Different SDEs might be more or less valid at different scales.

More specifically, we want to test which SDE better captures the dynamics of SGD around non-strict saddle points. So we can compare empirically the behaviour of the SDE arou nd non-strict saddle points with the behaviour of SGD around non-strict saddle points. Escape times and mixing times might be good indicators to track if we can compute them on DLNs and test with empirics.

For example, one conjecture would be that heavy-tailness around non-strict saddle better capture the escape dynamics of SGD. Perhaps we can compute the probability of escaping from one critical point to another in a given time.

6 Effect of noise anisotropy

What we need to know:

- Finding non-strict saddle points in DLNs.
- Dependency of the noise on the curvature (e.g. via Hessian)
- Get some conjecture on escape and mixing time around non-strict saddle

Def (non-strict saddle point): A non-strict saddle point is a critical point such that the Hessian has a zero eigenvalue and at least one negative eigenvalue (i.e. strict saddle point with a flat direction).
Def (escape time): We want to find the escape time around a non-strict saddle point. Mathematically, let θ^* be a non-strict saddle point. We are interested in the escape time i.e. the time it takes for the trajectory to escape from the saddle point toward lower losses regions.

$$\tau_r = \inf\{t > 0 : L(\theta^*) - L(\theta(t)) > \epsilon\} \tag{39}$$

In the following paper on escape time of SGD, the authors derive an expression for the escape time around a non-strict saddle point under a few assumptions: params. obey the quasi-stationary distribution around critical points, escape from critical point is dominated by most probable escape path (whose direction is given by some eigen value of the Hessian), SGD dynamics is restricted to the non zero eigen value of the Hessian (which forms a lower subspace of the loss). In such case the escape rate obeys a power law given by:

$$\tau_r \sim \frac{\sqrt{h_e^* \, |h_e^s|}}{2\pi} \left[\frac{L(\theta^s)}{L(\theta^*)}\right]^{\left(\frac{B}{\eta \, h_e^*}+1-\frac{n}{2}\right)}$$

For the Langevin SDE, the escape time is given by the Kramers law where:

$$< \tau >\sim \exp[-\beta \frac{\Delta L}{\sigma^2}] \tag{40}$$

(Note check if temperature is there) Empirically, we could check the escape time of SGD, Langevin and power law by estimating the time it takes for the trajectory to go from one saddle point to another. In DLNs, Find the first time at which the loss starts to plateau and the time it takes to escape the plateau. Take the same seed and average over many runs. We can plot the log of escape time with respect to $\log \Delta L$ and with respect to $\Delta \log L$. If anisotropy is present we should observe a power law as opposed to a linear relationship.

We could also look at the mixing time around a non-strict saddle point. Mathematically, let θ^* be a non-strict saddle point. We are interested in the mixing time i.e. the time it takes for the trajectory to mix around the saddle point.

6.1 Escape time for Langevin SDE

At a local minimum, we define the escape time of SGD as the exit time to escape the local minimum.

$$\tau_r = \inf\{t > 0 : |L_N(\theta(t)) - L_N(\theta^*)| > \epsilon\} \tag{41}$$

We model SGD as a Langevin SDE.