Perspectives on stochastic gradient descent

Jonas Latz

Department of Mathematics, University of Manchester

SIAG/UQ Early Career Prize, SIAM Conf. Uncertainty Quantification, Trieste, Italy, March 1st 2024.
Perspectives on stochastic gradient descent

*Stochastic gradient descent (SGD)* is a randomised algorithm for the optimisation of large sums of strongly convex functions.

**Perspective 1:** In modern machine learning, stochastic gradient descent is often used with a so-called constant learning rate, then:

- the algorithm *doesn’t converge to a minimiser*, but
- acts as an *implicit regulariser*

*Study the regularisation properties of stochastic gradient descent.*

**Perspective 2:** Stochastic gradient descent is an iterative algorithm of the form

\[ \theta_k \leftarrow F(\theta_{k-1}) \quad (k \in \mathbb{N}), \]

i.e., the algorithm generates a discrete-time dynamical system \((\theta_k)_{k=0}^{\infty}\).

*Propose a continuous-time variant of stochastic gradient descent \((\theta(t))_{t \geq 0}\) to analyse the constant learning rate setting.*
A continuous-time variant of stochastic gradient descent

Initial development:

Significant developments since then:

L. 2022: Subsampling error in Stochastic Gradient Langevin Diffusions, AISTATS.

Funding: Engineering and Physical Sciences Research Council, Swindon, UK

Credit for photo: Nick Saffell (Carola-Bibiane Schönlieb)
Outline

Motivation and background
  Optimisation in data science
  Challenges in supervised learning and stochastic gradient descent

Stochastic gradient descent in continuous time
  Algorithms in continuous time
  Stochastic gradient processes

Longtime behaviour of stochastic gradient processes and implicit regularisation

Conclusions
Outline

Motivation and background
  Optimisation in data science
    Challenges in supervised learning and stochastic gradient descent

Stochastic gradient descent in continuous time
  Algorithms in continuous time
  Stochastic gradient processes

Longtime behaviour of stochastic gradient processes and implicit regularisation

Conclusions
Problem setting

Consider a minimisation problem of the form

$$\min_{\theta \in X} \bar{\Phi}(\theta) := \frac{1}{N} \sum_{i=1}^{N} \Phi_i(\theta),$$

where $X := \mathbb{R}^n$ and the $\Phi_i$ are sufficiently smooth ($i = 1, \ldots, N$).

Optimisation in data science

- $\bar{\Phi}$ is some kind of potential ($\in \{\text{negative log-likelihood, loss function, misfit}\}$) given with respect to a large data set
- the $\Phi_i$ then represent the potential of a (small) data subsample
Optimisation in data science, e.g.,

- Image reconstruction
- Supervised learning
Optimisation in data science, e.g.,

Image reconstruction

Figure: Image of Galaxy M100 from Hubble before (left) and after fixing its mirror (right). [NASA, Hubble’s Mirror Flaw]

Use a variational approach to deblur an image by solving

\[
\min_{\theta \in X} \frac{1}{N} \sum_{i=1}^{N} (C_i \theta - y_i)^2 + \text{Reg}(\theta),
\]

where \((C_i)_{i=1}^{N}\) are the rows of a kernel matrix, \(\theta\) is the reconstructed image, \(y\) is the blurry image, and \(\text{Reg} : X \rightarrow \mathbb{R}\) is an appropriate regulariser.
Optimisation in data science, e.g.,

**Supervised learning**

Given a pair of random variables \((x, y) \sim \pi_{x,y}\).

- Learn how to predict \(y\) given \(x\), i.e. find \(f\):

\[
f(x) \approx y
\]

**Supervised learning:** approximate \(f\) using a parametric model \(f(\cdot; \theta)\) and sampled data \((x_1, y_1), \ldots, (x_n, y_n) \sim \pi_{x,y}\) by minimising

\[
\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \| \hat{f}(x_i; \theta) - y_i \|_2^2.
\]

**Image classification.** [Krizhevsky 2009]

Training data, e.g., from the CIFAR10 dataset:

- \((x_1, y_1) = (\text{cat}), \text{‘cat’})\),
- \((x_2, y_2) = (\text{frog}), \text{‘frog’})\),
- \((x_3, y_3) = (\text{airplane}), \text{‘airplane’})\),...
Optimisation in data science, e.g.,

**Supervised learning**

Given a pair of random variables \((x, y) \sim \pi_{x,y}\).

- Learn how to predict \(y\) given \(x\), i.e. find \(f\):

\[
    f(x) \approx y
\]

**Supervised learning:** approximate \(f\) using a parametric model \(\hat{f}(\cdot; \theta)\) and sampled data \((x_1, y_1), \ldots, (x_n, y_n) \sim \pi_{x,y}\) by minimising

\[
    \min_{\theta \in \mathcal{X}} \frac{1}{N} \sum_{i=1}^{N} \|\hat{f}(x_i; \theta) - y_i\|^2_{\Phi_i(\theta)}.
\]

**Image classification.**

[Dechter; 1986]

Deep neural networks have been particularly successful at imaging tasks. Here,

\[
    \hat{f}(x; \theta) = f^{(K)}
\]

\[
    f^{(k)} = \sigma(W^{(k)} f^{(k-1)} + b^{(k)}) \quad (k = 1, \ldots, K)
\]

\[
    f^{(0)} = x
\]

with \(\theta = (W^{(1)}, b^{(1)}, \ldots, W^{(K)}, b^{(K)})\) and \(\sigma\) being an activation function.
Gradient descent and stochastic gradient descent

How do we solve the following optimisation problem?

\[ \min_{\theta \in \mathcal{X}} \Phi(\theta) := \frac{1}{N} \sum_{i=1}^{N} \Phi_i(\theta) \]

Gradient descent (GD) for \( k = 1, 2, \ldots \):

\[ \theta_k \leftarrow \theta_{k-1} - \eta_k \nabla \Phi(\theta_{k-1}), \quad \nabla \Phi(\theta_{k-1}) := \frac{1}{N} \sum_{i=1}^{N} \nabla \Phi_i(\theta_{k-1}) \]

(N gradient evaluations)

- converges if \( \Phi \) has a minimiser and is convex and if the “step size” \( \eta_k \) is sufficiently small
Gradient descent and stochastic gradient descent

How do we solve the following optimisation problem?

$$\min_{\theta \in \mathcal{X}} \bar{\Phi}(\theta) := \frac{1}{N} \sum_{i=1}^{N} \Phi_i(\theta)$$

Stochastic gradient descent (SGD)

for $k = 1, 2, \ldots$:

$$\theta_k \leftarrow \theta_{k-1} - \eta_k \nabla \Phi_{i_k}(\theta_{k-1}), \quad i_k \sim \text{Unif}(I).$$

(= “subsampling”)

- converges if $\Phi_1, \ldots, \Phi_N$ are strongly convex and “learning rate” $\eta_k \downarrow 0 \ (k \to \infty)$ slowly

[Robbins + Monro; 1951]
Outline

Motivation and background
  Optimisation in data science
  Challenges in supervised learning and stochastic gradient descent

Stochastic gradient descent in continuous time
  Algorithms in continuous time
  Stochastic gradient processes

Longtime behaviour of stochastic gradient processes and implicit regularisation

Conclusions
Stochastic gradient descent in machine learning

Consider again the supervised learning problem

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \| \hat{f}(x_i; \theta) - y_i \|^2.$$

Problems in supervised learning.

- very large data sets ($N \gg 1$)
  → SGD can help with that.
- depending on the choice of $\hat{f}$, the target function may be non-convex
  - target functions in deep learning are usually non-convex  → SGD might struggle. [Du+al; 2017]
- solving this problem may overfit the data
  - machine learning models tend to be highly flexible and overparameterised
  - models may fit the noisy training data and generalise badly to unseen data  → Let’s see!
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with \(y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}, \ldots \sim \mathcal{N}(0, 1)\) iid.
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(R \times R\) with
\[y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}, \varepsilon^{(1)}_{\text{noise}}, \ldots \sim N(0, 1) \text{ iid}.

Construct model \(\hat{f}(x; \theta) := \sum_{i=0}^6 \theta_i H_i(x)\) on a Hermite basis.
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with \(y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}, \varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathcal{N}(0, 1)\) iid.

Construct model \(\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)\) on a Hermite basis.

![Graph showing data, truth, and estimate](image-url)
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with \(y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}, \varepsilon^{(1)}_{\text{noise}}, \ldots \sim N(0, 1)\) iid.

Construct model \(\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)\) on a Hermite basis.
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with 
\[ y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}^{(1)} \ldots \sim N(0, 1) \text{ iid}. \]

Construct model 
\[ \hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x) \] on a Hermite basis.

![Graph showing data, truth, and estimate](image)
Overfitting

- machine learning models tend to be highly flexible and overparameterised
- models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with
\[y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}, \varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathcal{N}(0, 1)\] iid.

Construct model \(\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)\) on a Hermite basis.
Overfitting

- machine learning models tend to be highly **flexible and overparameterised**
- models may fit the noisy training data and **generalise badly to unseen data**

Example. (Polynomial regression)

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with \(y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}, \ldots \sim \mathcal{N}(0, 1)\) iid.

Construct model \(\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)\) on a **Hermite basis**.
Overfitting

machine learning models tend to be highly flexible and overparameterised
models may fit the noisy training data and generalise badly to unseen data

Example. (Polynomial regression)

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with
\[ y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}, \ldots \sim \mathcal{N}(0, 1) \text{ iid.} \]

Construct model \(\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i \text{H}_i(x)\) on a Hermite basis.
Overfitting

- machine learning models tend to be highly **flexible and overparameterised**
- models may fit the noisy training data and **generalise badly to unseen data**

**Example. (Polynomial regression)**

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with \(y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}, \varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathcal{N}(0, 1) \) iid.

Construct model \(\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)\) on a Hermite basis.

Truth is badly estimated + estimated model is unstable with respect to resampling the noise. → overfitting
Regularisation

- The problem of overfitting in learning is related to that of ill-posedness in inverse problems
  - ill-posedness describes the non-existence, non-uniqueness or instability of estimates with respect to changes in the observational data
  - Instability:
    - Inversion: instability usually measured with \{Lipschitz, Hölder, -\} continuity with respect to data
    - Learning: study bias-variance tradeoff: instability $\Rightarrow$ large variance in training with respect to different data sets

- Ill-posedness in inverse problems can often be cured with regularisation
  - Variational regularisation: enforce additional information by modifying the target function
    $$\min_\theta \Phi(\theta) + \text{Reg}(\theta)$$
  - Bayesian approach: being aware of uncertainties usually leads to stability/well-posedness

- Overfitting can sometimes be addressed by regularisation

[Mohri+al.; 2018]
The Bayesian approach

Optimisation may not actually be the best way to learn a data set
- Fitting the model to the noisy data overfits the model
- Uncertainties remain in the model and are not quantified

**Bayesian approach**
- Consider the parameter $\theta$ to be uncertain and model knowledge/assumptions/... re $\theta$
  - with a so-called prior $\pi_{\text{prior}} = \mathbb{P}(\theta \in \cdot)$
- Use model and data to learn about $\theta$ by conditioning – obtain the posterior
  $$\pi_{\text{post}} = \mathbb{P}(\theta \in \cdot | y_i = \hat{f}(x_i; \theta) + \varepsilon^{(i)}_{\text{noise}}, i = 1, \ldots, m)$$
- Predictions of the trained model will be random/uncertain $\pi_{\text{post}}(\hat{f}(x; \theta) \in \cdot)$
- The posterior is usually stable with respect to perturbations in the data $\rightarrow$ well-posed

[Dashti+Stuart 2017] [Hosseini; 2017] [L.; 2020, 2023] [Sprungk; 2020] [Stuart; 2010] [Sullivan; 2017], ...
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\text{post}}(f(x; \theta) \in \cdot)$

**Bayesian polynomial regression**

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}^{(1)}, \ldots \sim \mathcal{N}(0, 1)$ iid.

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a **Hermite basis** and additionally enforce sparsity with **prior** $\pi_{\text{prior}} = \mathcal{N}(0, \text{diag}(2^{-1}, \ldots, 2^{-7}))$. 

![Graph showing data and truth](image)

Jonas Latz (Manchester)
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\text{post}}(f(x; \theta) \in \cdot)$

Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}^{(1)}, \ldots \sim \mathcal{N}(0, 1)$ iid.

Construct model $\widehat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\text{prior}} = \mathcal{N}(0, \text{diag}(2^{-1}, \ldots, 2^{-7}))$. 

![Graph showing data, truth, estimate, and +/- SD](image-url)
The Bayesian approach

- Predictions of the trained model will be random/uncertain \( \pi_{post}(f(x; \theta) \in \cdot) \)

Bayesian polynomial regression

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \( \mathbb{R} \times \mathbb{R} \) with \( y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}} \), \( \varepsilon^{(i)}_{\text{noise}} \), \( \ldots \sim N(0, 1) \) iid.

Construct model \( \hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x) \) on a Hermite basis and additionally enforce sparsity with prior \( \pi_{prior} = N(0, \text{diag}(2^{-1}, \ldots, 2^{-7})) \).
The Bayesian approach

- Predictions of the trained model will be random/uncertain \( \pi_{\text{post}}(f(x; \theta) \in \cdot) \)

### Bayesian polynomial regression

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with \(y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(1)}, \varepsilon_{\text{noise}}^{(1)}, \ldots \sim N(0, 1)\) iid.

Construct model \(\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)\) on a Hermite basis and additionally enforce sparsity with prior \(\pi_{\text{prior}} = N(0, \text{diag}(2^{-1}, \ldots, 2^{-7}))\).
The Bayesian approach

- Predictions of the trained model will be random/uncertain \( \pi_{\text{post}}(f(x; \theta) \in \cdot) \)

**Bayesian polynomial regression**

Let \((x_1, y_1), \ldots, (x_{10}, y_{10})\) be data pairs in \(\mathbb{R} \times \mathbb{R}\) with \(y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)} + \varepsilon_{\text{noise}}^{(1)}\) \(\ldots \sim N(0, 1)\) iid.

Construct model \(\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)\) on a Hermite basis and additionally enforce sparsity with prior \(\pi_{\text{prior}} = N(0, \text{diag}(2^{-1}, \ldots, 2^{-7}))\).
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\text{post}}(f(x; \theta) \in \cdot)$

### Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}^{(1)}, \ldots \sim N(0, 1)$ iid.

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\text{prior}} = N(0, \text{diag}(2^{-1}, \ldots, 2^{-7})).$
The Bayesian approach

- Predictions of the trained model will be random/uncertain $\pi_{\text{post}}(f(x; \theta) \in \cdot)$

Bayesian polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}^{(1)}, \ldots \sim \mathcal{N}(0, 1) \text{ iid.}$

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a Hermite basis and additionally enforce sparsity with prior $\pi_{\text{prior}} = \mathcal{N}(0, \text{diag}(2^{-1}, \ldots, 2^{-7})$.

Stable solution! → no overfitting
Stochastic gradient descent and implicit regularisation

- Regularisation of supervised learning problems is **difficult**
  - Correct choice of **regularisers or priors** is unclear in general (polynomial regression is easy)
  - **Bayesian learning** is computationally expensive

- **Idea:** Use stochastic gradient descent with a constant learning rate.
  - Markov chain Monte Carlo sampling from the posterior; often through a ‘noisy gradient descent’

\[
\theta_k \leftarrow \theta_{k-1} - \eta_k \nabla \tilde{\Phi}(\theta_{k-1}) + \eta_k \nabla \log \pi_{\text{prior}}(\theta_{k-1}) + \sqrt{\eta_k} \xi_k, \quad \xi_1, \xi_2, \ldots \sim \mathcal{N}(0, \text{Id}) \text{ iid.} \quad (\text{ULA})
\]

SGD is also a ‘noisy gradient descent’, maybe it can act as an approximate MCMC sampler?
- high variability in the model with respect to training data often suggests overfitting; **SGD leads to robustness with respect to small data sets at a time.**
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say $\eta_k$ constant)?

**SGD-regularised Polynomial regression**

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}^{(1)}, \ldots \sim N(0, 1)$ iid.

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a **Hermite basis** and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$. 

Not perfect, but not terrible and easy to obtain!
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say $\eta_k$ constant)?

**SGD-regularised Polynomial regression**

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}^{(1)}, \ldots \sim \mathcal{N}(0, 1)$ iid.

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a **Hermite basis** and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$. 

![Graph showing data, truth, estimate, and +/- SD](image_url)
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say $\eta_k$ constant)?

**SGD-regularised Polynomial regression**

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}, \varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathcal{N}(0, 1)$ iid.

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a **Hermite basis** and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.

![Graph](image.png)
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say $\eta_k$ constant)?

**SGD-regularised Polynomial regression**

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \epsilon_{\text{noise}}^{(i)}, \epsilon_{\text{noise}}^{(1)}, \ldots \sim N(0, 1)$ iid.

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$. 

Not perfect, but not terrible and easy to obtain!
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say $\eta_k$ constant)?

**SGD-regularised Polynomial regression**

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}, \varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathcal{N}(0, 1)$ iid.

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a **Hermite basis** and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say $\eta_k$ constant)?

**SGD-regularised Polynomial regression**

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon^{(i)}_{\text{noise}}, \varepsilon^{(1)}_{\text{noise}}, \ldots \sim \mathcal{N}(0, 1)$ iid.

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.
Can we just apply SGD in a non-convergent regime (say $\eta_k$ constant)?

### SGD-regularised Polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}^{(1)}, \ldots \sim \mathcal{N}(0, 1)$ iid.

Construct model $\hat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.
Stochastic gradient descent and implicit regularisation

- Can we just apply SGD in a non-convergent regime (say $\eta_k$ constant)?

## SGD-regularised Polynomial regression

Let $(x_1, y_1), \ldots, (x_{10}, y_{10})$ be data pairs in $\mathbb{R} \times \mathbb{R}$ with $y_i = 1 - x_i + \varepsilon_{\text{noise}}^{(i)}, \varepsilon_{\text{noise}}^{(1)}, \ldots \sim N(0, 1)$ iid.

Construct model $\widehat{f}(x; \theta) := \sum_{i=0}^{6} \theta_i H_i(x)$ on a Hermite basis and optimise the non-regularised loss function with SGD with learning rate $\eta_k = 8 \cdot 10^{-5}$.

Not perfect, but not terrible and easy to obtain!
Stochastic gradient descent and implicit regularisation

- Regularisation of supervised learning problems is difficult
  - Correct choice of regularisers or priors is unclear in general (polynomial regression is easy)
  - Bayesian learning is computationally expensive

- **Idea:** Use stochastic gradient descent with a constant learning rate.
  - Markov chain Monte Carlo sampling from the posterior; often through a ‘noisy gradient descent’
    \[ \theta_k \leftarrow \theta_{k-1} - \eta_k \nabla \Phi(\theta_{k-1}) + \eta_k \nabla \log \pi_{\text{prior}}(\theta_{k-1}) + \sqrt{\eta_k} \xi_k, \quad \xi_1, \xi_2, \ldots \sim \mathcal{N}(0, \text{Id}) \text{ iid.} \quad \text{ULA} \]

  SGD is also a ‘noisy gradient descent’, maybe it can act as an approximate MCMC sampler?
  - high variability in the model with respect to training data often suggests overfitting; SGD leads to robustness with respect to small data sets at a time.

- Indeed, this is a rather popular way of regularisation in machine learning; part of implicit regularisation.

- **Questions**
  - What actually happens when we apply SGD with constant learning rate? [Dieuleveut+al.; 2020]
Outline

Motivation and background
  Optimisation in data science
  Challenges in supervised learning and stochastic gradient descent

Stochastic gradient descent in continuous time
  Algorithms in continuous time
  Stochastic gradient processes

Longtime behaviour of stochastic gradient processes and implicit regularisation

Conclusions
Algorithms in continuous time

- **Iterative algorithms** can be understood as discrete-in-time dynamical systems
  \[ \xi_k \leftarrow F(\xi_{k-1}) \quad (k \in \mathbb{N}), \quad \xi_0 \in X \]

- We can sometimes find continuous-in-time dynamical systems, e.g.,
  \[ \dot{\xi}(t) = G(\xi(t)) \quad (t \geq 0), \quad \xi(0) = \xi_0 \]
  that behave analogous to the iterative algorithms

**Example. (Gradient descent and gradient flows)**

Gradient descent

\[ \zeta_k \leftarrow \zeta_{k-1} - \eta_k \nabla \Phi(\zeta_{k-1}) \quad (k \in \mathbb{N}) \]

is a **forward Euler discretisation** of the ordinary differential equation

\[ \dot{\zeta}(t) = -\nabla \Phi(\zeta(t)) \quad (t \geq 0) \]

which is a so-called **gradient flow**.
Algorithms in continuous time

It is sometimes easier or more appropriate to analyse algorithms in continuous time

- certain numerical artefacts that appear in the discrete setting are not particularly interesting: stiffness, step size restrictions,...
- certain effects can are more or only visible in a continuous setting: ill-posedness of deconvolution,...
- continuous time adds additional regularity

Continuous time allows us to compare algorithms to physical/biological processes

- Gradient flows appear everywhere, e.g., the heat equation $\dot{u} = \triangle u$
- Certain classification methods behave like partial differential equations that describe phase separation

Diffusion limit of SGD

Predominant model for SGD in continuous time: Diffusion process

- Idea: $\eta_k = \eta \approx 0 \Rightarrow$ gradient error is approximately Gaussian (CLT)
- Hence, $(\theta_k)_{k=1}^{\infty}$ can be represented by a diffusion process

$$\dot{\theta}_{\text{sde}}(t) = -\nabla \Phi(\theta_{\text{sde}}(t)) + \sqrt{\eta \Sigma(\theta_{\text{sde}}(t))^{1/2}} \dot{W}_t \quad (t \geq 0), \quad \theta(0) = \theta_0.$$  

[Hu+al.; 2019] [Li+al.; 2016, 2017, 2019] [Mandt+al.; 2015, 2016, 2017] [Wojtowytsch; 2024] ...

- for large $\eta_k$, the paths of $(\theta_k)_{k=1}^{\infty}$ are very different from a diffusion
  - preasymptotic phase and constant $\eta_k$ not explained
- diffusion does not actually explain subsampling in a continuous-time model
  - does not represent the discrete nature of the potential selection
  - needs access to $\Phi$
Outline

Motivation and background
  Optimisation in data science
  Challenges in supervised learning and stochastic gradient descent

Stochastic gradient descent in continuous time
  Algorithms in continuous time
  Stochastic gradient processes

Longtime behaviour of stochastic gradient processes and implicit regularisation

Conclusions
Observations and fundamental idea

- the update

\[ \theta_k \leftarrow \theta_{k-1} - \eta \nabla \Phi_i(\theta_{k-1}) \]  
  (discrete)

is a forward Euler discretisation of the gradient flow

\[ \dot{\theta}(t) = -\nabla \Phi_i(\theta(t)) \]  
  (continuous)

- learning rate \( \eta \) has two different meanings
  (i) \( \eta \) is the step size of the gradient flow discretisation
  (ii) \( \eta \) determines the length of the time interval with which we switch the \( \Phi_i \)

Idea.

Obtain a continuous time model for SGD, by

(i) let the step size go to 0, i.e. replace (discrete) by (continuous).
(ii) switch the potentials in the gradient flow at a rate of \( 1/\eta \)
Switching of the potentials

control the switching of the potentials by a continuous-time Markov process (CTMP) \((i(t))_{t \geq 0}\) on 
\(I := \{1, \ldots, N\}\) ("index process")

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure}
\caption{Cartoon of a CTMP}
\end{figure}

CTMPs 101

\begin{itemize}
\item \((i(t))_{t \geq 0}\) is piecewise constant
\item randomly jumps from one state to another after a random waiting time \(\Delta \sim \pi_{wt}(\cdot | t_0)\)
\end{itemize}
Stochastic gradient process

CTMP \((i(t))_{t \geq 0}\) representing a constant learning rate \(\eta_0 \equiv \eta > 0\)

- constant learning rates are popular in practice
- \(\pi_{wt}(\cdot | t_0)\) is constant in time (indeed this will be an exponential distribution)

\((i(t))_{t \geq 0}\) has constant transition rate matrix \(A \in \mathbb{R}^{N \times N}: A_{i,j} := \begin{cases} \frac{1}{(N-1)\eta}, & \text{if } i \neq j, \\ -\frac{1}{\eta}, & \text{if } i = j. \end{cases}\)

Definition. \[\text{[L.; 2021]}\]

We define the stochastic gradient process with constant learning rate (SGPC) by \((\theta(t))_{t \geq 0}\), which satisfies

\[
\dot{\theta}(t) = -\nabla \Phi_{i(t)}(\theta(t)) \quad (t \geq 0), \quad \theta(0) = \theta_0.
\]

\((\theta(t))_{t \geq 0}\) and \((\xi(t))_{t \geq 0}\) are almost surely well-defined, if

Assumption [Lipschitz]. For \(i \in I: \Phi_i \in C^1(X, \mathbb{R})\) and \(\nabla \Phi_i\) is Lipschitz continuous.
Stochastic gradient process

Figure: Cartoon of SGPC
Properties

- \((i(t), \theta(t))_{t \geq 0}\) is a **piecewise-deterministic Markov process**: essentially an ODE with a right-hand side that changes at random points in time

- the choice of the transition rate matrix of \((i(t), \theta(t))_{t \geq 0}\) leads to subsampling at rate \(1/\eta\)
  - mean waiting time \(\mathbb{E}[T_i - T_{i-1}] = \eta\)

- \((\theta(t))_{t \geq 0}\) approximates the gradient flow \((\zeta(t))_{t \geq 0}\) for small \(\eta\)

\[L.; 2021\]
Short learning rate ($\eta \downarrow 0$)

**Example.** Let $\Phi_1(\theta) := (\theta - 1)^2/2$ and $\Phi_2(\theta) := (\theta + 1)^2/2$. $\Rightarrow \bar{\Phi}(\theta) = (\theta^2 + 1)/2$.

![Exemplary realisations of SGPC and plot of precise gradient flow. Discretisation with ode45.](image)

**Figure:** Exemplary realisations of SGPC and plot of precise gradient flow. Discretisation with `ode45`.

Convergence proof in [L.; 2021] using techniques from [Kushner; 1984].
Properties

- $(i(t), \theta(t))_{t \geq 0}$ is a piecewise-deterministic Markov process: essentially an ODE with a right-hand side that changes at random points in time.
- The choice of the transition rate matrix of $(i(t), \theta(t))_{t \geq 0}$ leads to subsampling at rate $1/\eta$.
  - Mean waiting time $\mathbb{E}[T_i - T_{i-1}] = \eta$.
- $(\theta(t))_{t \geq 0}$ approximates the gradient flow $(\zeta(t))_{t \geq 0}$ for small $\eta$.
- Stochastic gradient flow has a biological interpretation.
  - Clonal populations that live in randomly changing environments use diversified bet-hedging strategies that follow similar dynamics.

[Jonas Latz (Manchester)]
Outline

Motivation and background
  Optimisation in data science
  Challenges in supervised learning and stochastic gradient descent

Stochastic gradient descent in continuous time
  Algorithms in continuous time
  Stochastic gradient processes

Longtime behaviour of stochastic gradient processes and implicit regularisation

Conclusions
Longtime behaviour \((t \to \infty)\)

What happens with \(\mathbb{P}(\theta(t) \in \cdot)\) as \(t \to \infty\)?

- Stability? Stationary measures?
- Speed of convergence?
- Characterisation of stationary measures?
- Implicit regularisation?
- Posteriors?
Preliminaries

Wasserstein distance

Let $q \in (0, 1]$. Consider Wasserstein distance between $\pi, \pi' \in \text{Prob}(X)$:

$$W(\pi, \pi') := \inf_{H \in \text{Coup}(\pi, \pi')} \int_{\times X} \min\{1, \|\theta - \theta'\|_2^q\} H(d\theta, d\theta'),$$

$$\text{Coup}(\pi, \pi') := \{G \in \text{Prob}(X^2) : G(\cdot \times X) = \pi, G(X \times \cdot) = \pi'\}.$$

Assumption [Smooth]. For any $i \in I$, let $\Phi_i \in C^2(X; \mathbb{R})$ and let $\nabla \Phi_i, H\Phi_i$ be continuous and bounded on bounded subsets of $X$.

Assumption [Convex]. There are some $\kappa_i \in \mathbb{R}$, with

$$\langle \theta_0 - \theta'_0, \nabla \Phi_i(\theta_0) - \nabla \Phi_i(\theta'_0) \rangle \geq \kappa_i \|\theta_0 - \theta'_0\|^2$$

($\theta_0, \theta'_0 \in X, i \in I$),

with $\kappa_1 + \cdots + \kappa_N > 0$. 
Longtime behaviour \((t \to \infty)\)

**Theorem.**

Let Assumptions [Smooth] and [Convex] hold. Then, \((\theta(t), i(t))_{t>0}\) has a unique stationary measure \(\pi_C\) on \((X \times I, \mathcal{B}X \otimes 2^I)\). Moreover, there exist \(\kappa', c > 0\) and \(q \in (0, 1]\), with

\[
W(\pi_C(\cdot \times I), \mathbb{P}(\theta(t) \in \cdot | \theta_0, i_0)) \leq c \exp(-\kappa' t) \left( 1 + \sum_{i \in I} \int_X \|\theta_0 - \theta'\|^q \pi_C(d\theta' \times \{i\}) \right)
\]

\((i_0 \in I, \theta_0 \in X)\).
Theorem. [L.; 2021]

Let Assumptions [Smooth] and [Convex] hold. Then, \((\theta(t), i(t))_{t>0}\) has a unique stationary measure \(\pi_C\) on \((\mathcal{X} \times I, \mathcal{B}_{\mathcal{X}} \otimes 2^I)\). Moreover, there exist \(\kappa', c > 0\) and \(q \in (0, 1]\), with

\[
W(\pi_C(\cdot \times I), \mathbb{P}(\theta(t) \in \cdot | \theta_0, i_0)) \leq c \exp(-\kappa' t) \left( 1 + \sum_{i \in I} \int_{\mathcal{X}} \|\theta_0 - \theta'|^q \pi_C(d\theta' \times \{i\}) \right) \quad (i_0 \in I, \theta_0 \in \mathcal{X}).
\]

- convergence with exponential speed
- proof based on results by [Benaïm+al.; 2012] [Cloez+Hairer; 2015]
- convexity assumption can be weakened (needs Hörmander Bracket condition)
- SGD with constant stepsize is safe to use in ‘more-convex-than-not’ settings and converges very quickly to its stationary regime
Longtime behaviour \((t \rightarrow \infty)\)

Figure: Kernel density estimates of \(P(\theta(10) \in \cdot | \theta(0) = -1.5) \approx \pi_C\) (SGPC) and \(P(\theta_{10/\eta} \in \cdot | \theta_0 = -1.5)\) (SGD) based on \(\eta \in \{1, 0.1, 0.01, 0.001\}\) using 10,000 samples each. [Example. Let \(N := 3\), i.e. \(I := \{1, 2, 3\}\), and \(X := \mathbb{R}\). We define the potentials \(\Phi_1(\theta) := \frac{1}{2}(\theta + 2)^2\), \(\Phi_2(\theta) := \frac{1}{2}(\theta - 1.5)^2\), \(\Phi_3(\theta) := \frac{1}{2}(\theta - 2)^2\) \((\theta \in X)\). Here, argmin \(\Phi = \{0.5\}\).]
Stationary distributions and implicit regularisation

- $\pi_C$ might be a good representation for the implicit regularisation achieved by SGD
- It appears as if $\pi_C \to \delta(\cdot - \theta^*)$, as $\eta \downarrow 0$, where $\theta^* \in \text{argmin} \Phi$. Indeed, we can show:

**Corollary.**
Let Assumptions [Smooth] and [Convex2] hold. Then, $\lim_{\eta \downarrow 0} W(\pi_C(\cdot \times I), \delta(\cdot - \theta^*)) = 0$.

**Assumption** [Convex2]. There is a $\kappa > 0$, with

$$\langle \theta_0 - \theta_0', \nabla \Phi_i(\theta_0) - \nabla \Phi_i(\theta_0') \rangle \geq \kappa \|\theta_0 - \theta_0'\|^2 \quad (\theta_0, \theta_0' \in X, i \in I).$$

- the corollary above is a simple application of Proposition 4(ii) in [L.; 2021]
- the result shows that $\eta$ controls the strength of the regularisation
  - decreasing $\eta$ over time corresponds to the classical SGD setting
  - we can also do that in the stochastic gradient process
Implicit regularisation and posteriors

- $\pi_C$ behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$
\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \|\theta - 1\|^2 + \frac{1}{2} \|\theta + 1\|^2
$$

Employ SGPC with $\sigma = 10$.

$\pi_C$ is concentrated on a subspace and rather different from the measure we would associate with this optimisation problem.

[Benaim+al.; 2015]
Implicit regularisation and posteriors

- $\pi_C$ behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \left( \| \theta - 1 \|^2 + \frac{1}{2} \| \theta + 1 \|^2 \right)$$

Employ SGPC with $\eta = 10$. 

[Benaim+al.; 2015]
Implicit regularisation and posteriors

- $\pi_C$ behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \left( \|\theta - 1\|^2 + \frac{1}{2} \|\theta + 1\|^2 \right)$$

Employ SGPC with $\eta = 10$. 
Implicit regularisation and posteriors

- $\pi_C$ behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \|\theta - \mathbf{1}\|^2 + \frac{1}{2} \|\theta + \mathbf{1}\|^2$$

Employ SGPC with $\eta = 10$. 

[Benaim+al.; 2015]
Implicit regularisation and posteriors

- \( \pi_C \) behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

\[
\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \left( \| \theta - 1 \|^2 + \frac{1}{2} \| \theta + 1 \|^2 \right)
\]

Employ SGPC with \( \eta = 10 \).
Implicit regularisation and posteriors

- $\pi_C$ behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \|\theta - \mathbf{1}\|^2 + \frac{1}{2} \|\theta + \mathbf{1}\|^2$$

:= $\Phi_1$  := $\Phi_1$

Employ SGPC with $\eta = 10$. 

[Benaim+al.; 2015]
Implicit regularisation and posteriors

- $\pi_C$ behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$
\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \|\theta - 1\|^2 + \frac{1}{2} \|\theta + 1\|^2
$$

Employ SGPC with $\eta = 10$. 

[Benaim+al.; 2015]
Implicit regularisation and posteriors

- $\pi_C$ behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

SGPC in a Gaussian setting

Consider the quadratic minimisation problem

$$\min_{\theta \in \mathbb{R}^2} \frac{1}{2} \left\| \theta - 1 \right\|^2 + \frac{1}{2} \left\| \theta + 1 \right\|^2$$

Employ SGPC with $\eta = 10$.

$\pi_C$ is concentrated on a subspace and rather different from the measure we would associate with this optimisation problem.
Implicit regularisation and posteriors

- $\pi_C$ behaves quite differently from a posterior!
  - there is no natural underlying prior
  - usually concentrated on a compact set, sometimes on a subspace

- Can we turn this into a posterior?
  - In principle, yes. The Stochastic Gradient Langevin Dynamics combines SGD and ULA by adding white noise to SGD. It approximates $\pi \propto \exp(-\bar{\Phi})$.
  - In continuous time, we obtain the Stochastic Gradient Langevin Diffusion

$$d\theta(t) = -\nabla \Phi_i(t)(\theta(t))dt + \sqrt{2}dW_t,$$

[Benaim+al.; 2015]
[Welling+Teh; 2011]
[Jin, Liu, L.; 2024]
Outline

Motivation and background
  Optimisation in data science
  Challenges in supervised learning and stochastic gradient descent

Stochastic gradient descent in continuous time
  Algorithms in continuous time
  Stochastic gradient processes

Longtime behaviour of stochastic gradient processes and implicit regularisation

Conclusions
Conclusions

today:

- **Implicit regularisation** is a vital tool in machine learning; the stochastic gradient descent algorithm can be used as such an implicit regulariser.

- the stochastic gradient process is a **natural continuous-time variant** of stochastic gradient descent.

- in convex settings, stochastic gradient processes are **stable and converge quickly to their stationary regime**.

- the stationary regime may explain **implicit regularisation**; the strength of regularisation is controlled by the learning rate parameter.

related results:

- mildly non-convex/non-smooth optimisation [L.; 2022]
- subsampling in particle-based optimisation [Hanu+L.+Schillings; 2023]
- subsampling with continuous data and other sampling patterns [Jin+Liu+L.+Schönlieb; 2023]