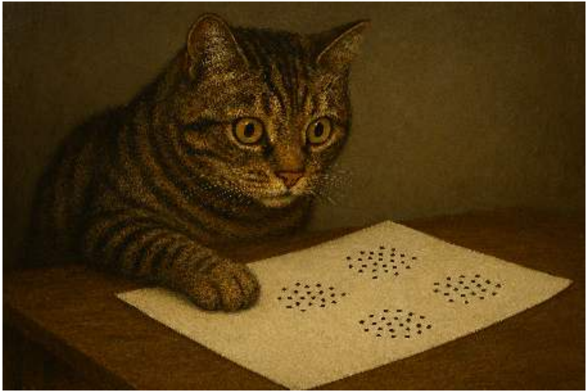
Advanced Topics in Machine Learning (deep generative modelling)

Lecture 1: Introduction and overview



Nikolay Malkin

13 January 2026

Outline of Lecture 1

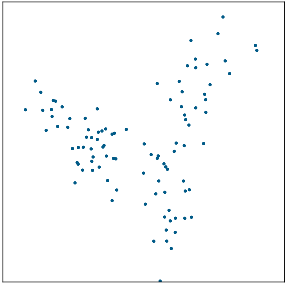
Not very technical today:

- What is (deep) generative modelling?
  - Very brief historical overview
- Why do generative modelling?
  - Why not do generative modelling?
- Overview of the course
  - What you should and should not know
  - Logistics

- What is (deep) generative modelling?
  - Very brief historical overview
- Why do generative modelling?
  - Why not do generative modelling?
- Overview of the course
  - What you should and should not know
  - Logistics

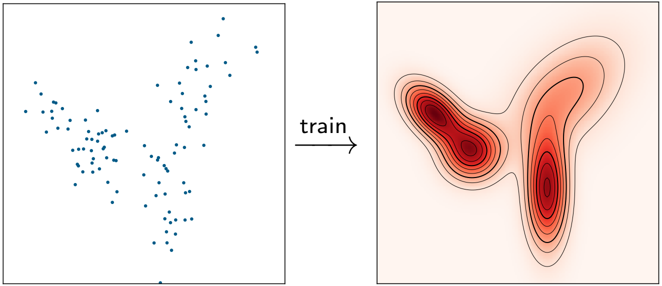
What is generative modelling?

Informally, generative modelling is the task of approximating the distribution that produced some observed data. (Next time, we make this formal.)



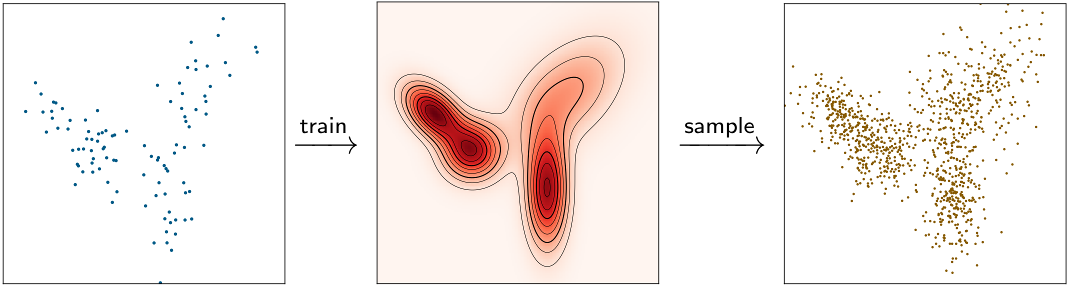
What is generative modelling?

Informally, generative modelling is the task of approximating the distribution that produced some observed data. (Next time, we make this formal.)



What is generative modelling?

Informally, generative modelling is the task of approximating the distribution that produced some observed data. (Next time, we make this formal.)



What is generative modelling?

Informally, generative modelling is the task of approximating the distribution that produced some observed data. (Next time, we make this formal.)

Terminology:

- A **generative model** is a probability distribution, or generative process, that is **derived from data** so as to approximate the distribution that produced the data.
- A **deep generative model** is one that uses **deep neural networks** to represent (components of) the generative process.
- A **deep generative modelling algorithm** consists of: a choice of generative process, a family of distributions parametrised by neural networks to represent that process, and a **learning algorithm** to fit those networks' parameters to data.

What is generative modelling?

Informally, generative modelling is the task of approximating the distribution that produced some observed data. (Next time, we make this formal.)

The questions:

- How to represent the approximating distribution (i.e., the choice of generative process and its parametrisation)
- How to fit it to data (the learning algorithm)

What is generative modelling?

Informally, generative modelling is the task of approximating the distribution that produced some observed data. (Next time, we make this formal.)
The questions:

- How to represent the approximating distribution (i.e., the choice of generative process and its parametrisation)
- How to fit it to data (the learning algorithm)

'Edinburgh from Calton Hill, pointillist style'

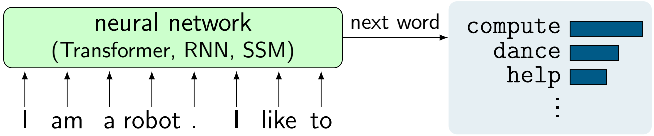
[Zhang and Gienger, χ:2409.01083]

What is a generative process?

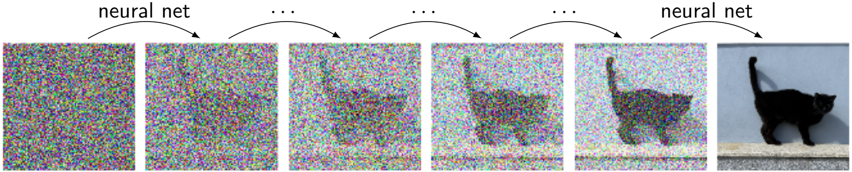
A procedure that generates samples from some probability distribution:

- A probabilistic program (program with random choices)
- May involve multiple steps and (learnt) internal parameters

Autoregressive language model



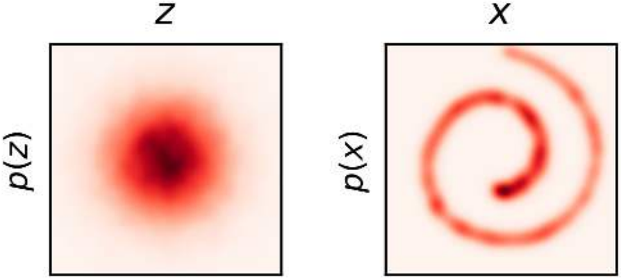
Diffusion image model



What is a generative process?

Examples:

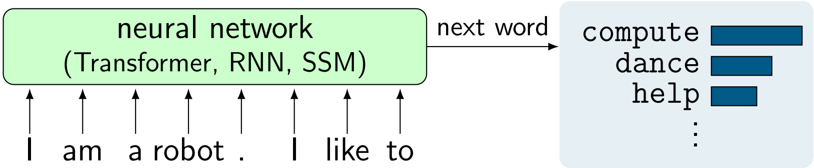
- Sample a vector $z \in \mathbb{R}^2$ from $\mathcal{N}(0, I)$, pass it through a neural network f_θ, output $x = f_\theta(z)$.



What is a generative process?

Examples:

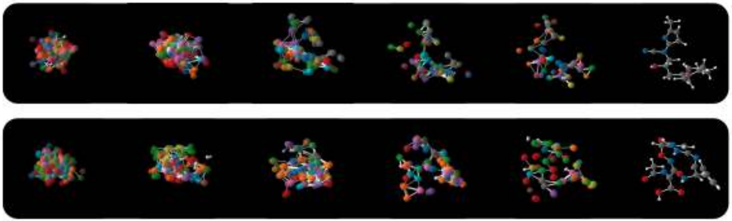
- Sample a vector $z \in \mathbb{R}^2$ from $\mathcal{N}(0, I)$, pass it through a neural network f_θ, output $x = f_\theta(z)$.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until <end> is produced; output the sequence.



What is a generative process?

Examples:

- Sample a vector $z \in \mathbb{R}^2$ from $\mathcal{N}(0, I)$, pass it through a neural network f_θ, output $x = f_\theta(z)$.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until `<end>` is produced; output the sequence.
- Sample a random point cloud and run a physics simulation (following a learnt force field) for a fixed time horizon. Output the resulting point cloud.



What is a generative process?

Examples:

- Sample a vector $z \in \mathbb{R}^2$ from $\mathcal{N}(0, I)$, pass it through a neural network f_θ, output $x = f_\theta(z)$.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until `<end>` is produced; output the sequence.
- Sample a random point cloud and run a physics simulation (following a learnt force field) for a fixed time horizon. Output the resulting point cloud.
- Sample ten random points from the training dataset. Supply them to a neural network that produces a new sample that is similar to them.

What kinds of data is each process good for?

What is a generative process?

Examples:

- Sample a vector $z \in \mathbb{R}^2$ from $\mathcal{N}(0, I)$, pass it through a neural network f_θ, output $x = f_\theta(z)$.
- Begin with an empty sequence. Pass the sequence through a neural network to get a distribution over the next symbol, sample from it, and append. Repeat until `<end>` is produced; output the sequence.
- Sample a random point cloud and run a physics simulation (following a learnt force field) for a fixed time horizon. Output the resulting point cloud.
- Sample ten random points from the training dataset. Supply them to a neural network that produces a new sample that is similar to them.

What kinds of data is each process good for?

(As we shall see, all of the above can be used for image generation.)

What is generative modelling not?

Generative modelling is different from:

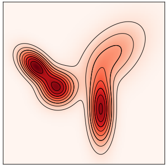
- **Discriminative modelling:** predicting one covariate from others (classification, regression, prediction)
  - Discriminative modelling is a kind of *conditional* generative modelling
- **Representation learning:** producing useful features of data without approximating the data distribution itself
- **Reinforcement learning:** learning from observations and rewards received during interaction with an environment

(However, generative modelling can be useful in / interacts with all of these domains.)

Historical sketch

- Classical (Bayesian) statistics and ML always studied generative models

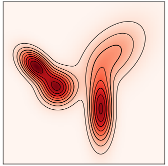
Gaussian mixture

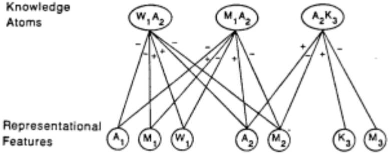


Historical sketch

- Classical (Bayesian) statistics and ML always studied generative models
- Many of the first neural nets (1980s-90s) are generative models
  - Mostly energy-based models (undirected graphical models); we do not focus on these in the course

Gaussian mixture



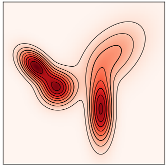


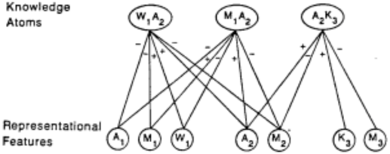
Restricted Boltzmann Machine (Harmonium, 1986)

Historical sketch

- Classical (Bayesian) statistics and ML always studied generative models
- Many of the first neural nets (1980s-90s) are generative models
- Deep learning revolution ($\sim$2010s): discriminative models (predicting low-dim covariates from high-dim data) dominate

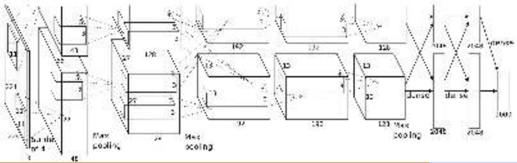
Gaussian mixture





Restricted Boltzmann Machine (Harmonium, 1986)

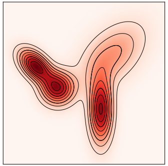
AlexNet (2012)

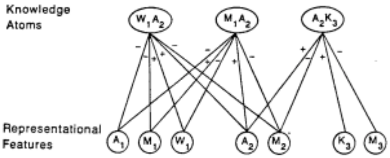


Historical sketch

- Classical (Bayesian) statistics and ML always studied generative models
- Many of the first neural nets (1980s-90s) are generative models
- Deep learning revolution ($\sim$2010s): discriminative models (predicting low-dim covariates from high-dim data) dominate
- Recently: new generative models for all kinds of data

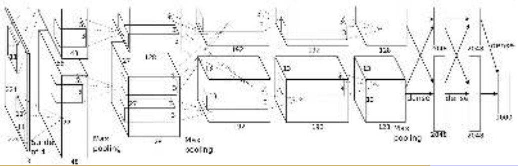
Gaussian mixture





Restricted Boltzmann Machine (Harmonium, 1986)

AlexNet (2012)

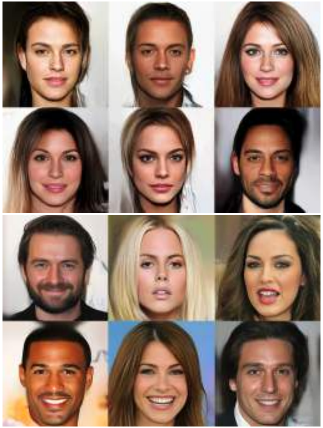


AlphaFold 3 (2024)

- What is (deep) generative modelling?
  - Very brief historical overview
- Why do generative modelling?
  - Why not do generative modelling?
- Overview of the course
  - What you should and should not know
  - Logistics

Desiderata for generative modelling

Generative models can produce new samples that look like they came from the same distribution as the training dataset. What do we want from such models?



Desiderata for generative modelling

Generative models can produce new samples that look like they came from the same distribution as the training dataset. What do we want from such models?

- Fidelity (samples should look like training data)
  - The model should not produce samples far from the training data with high probability

Desiderata for generative modelling

Generative models can produce new samples that look like they came from the same distribution as the training dataset. What do we want from such models?

- Fidelity (samples should look like training data)
- Diversity (samples should represent the variation in the training data)
  - The model should produce samples close to all parts of the training data with high probability

Desiderata for generative modelling

Generative models can produce new samples that look like they came from the same distribution as the training dataset. What do we want from such models?

- Fidelity (samples should look like training data)
- Diversity (samples should represent the variation in the training data)
- Novelty (samples should not be copies of training data)
  - The modelled distribution should be smooth to prevent memorisation (overfitting)

Desiderata for generative modelling

Generative models can produce new samples that look like they came from the same distribution as the training dataset. What do we want from such models?

- Fidelity (samples should look like training data)
- Diversity (samples should represent the variation in the training data)
- Novelty (samples should not be copies of training data)

Which goal is different from the others?

Desiderata for generative modelling

Generative models can produce new samples that look like they came from the same distribution as the training dataset. What do we want from such models?


- Fidelity (samples should look like training data)
- Diversity (samples should represent the variation in the training data)
- Novelty (samples should not be copies of training data)

Which goal is different from the others?

Next time, we will see ways to achieve and assess these goals. (But **making fake things that look like real ones** is not the only point of generative modelling.)

Other motivations for generative modelling

Why may we want to approximate data distributions, beyond just generating realistic samples?

Other motivations for generative modelling

Why may we want to approximate data distributions, beyond just generating realistic samples?

- Getting ‘reasonable’ approximations from a small number of samples (imitation learning for robotics, protein folding, ...)
- Anomaly/outlier detection
- Data compression and representation learning $\rightsquigarrow$ downstream tasks
- Smooth approximations help understand complex phenomena in nature
- Measuring and reasoning under uncertainty

(Examples of these to follow in the course.)

Why not do generative modelling?

Generative modelling (approximating probability distributions given samples)	$\neq$	"Generative AI" (branding for various deployments of generative models)

Why not do generative modelling?

Generative modelling (approximating probability distributions given samples)	$\neq$	"Generative AI" (branding for various deployments of generative models)

The focus of this track is how to approximate probability distributions

Why not do generative modelling?

Generative modelling (approximating probability distributions given samples)	$\neq$	"Generative AI" (branding for various deployments of generative models)

The focus of this track is how to approximate probability distributions

- A good approximation to a distribution is not the same as one that accurately represents the world (and can be used responsibly)!
- But understanding how models do and do not generalise can help understand the bounds of their applicability
- We will **not** cover ethics of AI (but come to the guest lecture)

However, some questions for thought...

Do we always want generalisation?

Inferring what is unstated is a key part of human communication, but do we want it from models?

Write a five-sentence story about Milo. They have a cat.

```
Milo is a lonely software engineer who lives in a small apartment in the city. He spends most of his time coding and playing video games. One day, he found a stray cat outside his apartment. He decided to take the cat in and named it Whiskers. Whiskers became Milo's best friend and kept him company while he worked. Milo was no longer lonely with Whiskers by his side.
```

What does this mean for bias and the ethics of deployment?

Do we always want generalisation?

Inferring what is unstated is a key part of human communication, but do we want it from models?

Write a five-sentence story about Milo (they/them).

```
Milo is a non-binary person who loves to write stories.  They have
a passion for creating characters and worlds that are diverse and
inclusive.  Milo's stories often feature LGBTQ+ characters and themes,
reflecting their own experiences and identity.  They hope that their
writing can inspire others to embrace their true selves and feel
represented in the media.  Milo dreams of becoming a published author
and sharing their stories with the world.
```

What does this mean for bias and the ethics of deployment?

Models reflect statistical patterns, not reality

They are not constrained by any notion of truth

How many frogs did Julius Caesar eat in his lifetime?

`This is a simple problem that can be solved by using the formula for the sum of the first` n `terms of an arithmetic progression [...] The number of frogs Julius Caesar ate in his lifetime can be represented by an arithmetic progression with the first term being 1 and the last term being 100. The sum of the first 100 terms of this arithmetic progression can be calculated using the formula:` $S_{100} = \frac{100}{2}(1 + 100) = 5050$`. Therefore, Julius Caesar ate 5050 frogs in his lifetime.`

What does this mean for reliability?

Models reflect statistical patterns, not reality

They are not constrained by any notion of truth

Edinburgh from Calton Hill, pointillist style

a goalkeeper hovering above a violet football

a mouse chasing a cat down a wynd

What does this mean for reliability?

The synthetic data feedback loop

Models generate synthetic data, other models fit to it. What could go wrong?

[Bertrand et al., χ:2310.00429]

What do we do when the training data is flooded with (even well-meaning) AI-generated content?

- What is (deep) generative modelling?
  - Very brief historical overview
- Why do generative modelling?
  - Why not do generative modelling?
- Overview of the course
  - What you should and should not know
  - Logistics

Plan for the track

- Week 2: distribution approximation and divergence measures
- Week 3: latent variable models (autoencoders)
- Week 4: normalising flows, autoregressive models, and density estimation
- Week 5: adversarial learning (GANs) and density-free evaluation
- Break
- Week 6: downstream uses: representation learning, conditioning, improvement with human feedback
- Week 7: guest lecture: "Evaluating generative AI"
- Weeks 8-10: dynamics-based generative models (diffusion models and friends)
- Week 11: revision and advanced topics (according to interest)

Introductions

Nikolay Malkin
they/she

Lecturer, Inf.
(instructor)

Kirill Tamogashev
he/him

PhD student, Inf.
(tutor)

Rajit Rajpal
he/him

PhD student, Maths
(TA)

Introductions

Nikolay Malkin
they/she

Lecturer, Inf.
(instructor)

Kirill Tamogashev
he/him

PhD student, Inf.
(tutor)

Rajit Rajpal
he/him

PhD student, Maths
(TA)

Introductions

Nikolay Malkin
they/she

Lecturer, Inf.
(instructor)

Kirill Tamogashev
he/him

PhD student, Inf.
(tutor)

Rajit Rajpal
he/him

PhD student, Maths
(TA)

Now, some questions for you. . .

What you should know

- Basics of neural nets and machine learning
  - You can implement a MLP (e.g., in Torch) for classification/regression, train it with a standard optimiser, and evaluate its performance
  - Not expected: domain-specific architectures, large-scale training, ML theory

What you should know

- Basics of neural nets and machine learning
  - You can implement a MLP (e.g., in Torch) for classification/regression, train it with a standard optimiser, and evaluate its performance
  - Not expected: domain-specific architectures, large-scale training, ML theory
- Probability theory and statistics
  - You can state and apply Bayes' theorem, write the density of a Gaussian distribution and of a Gaussian mixture, estimate an expectation using Monte Carlo, understand (conditional) independence
  - Not expected: measure theory, stochastic processes, advanced hypothesis testing

What you should know

- Basics of neural nets and machine learning
  - You can implement a MLP (e.g., in Torch) for classification/regression, train it with a standard optimiser, and evaluate its performance
  - Not expected: domain-specific architectures, large-scale training, ML theory
- Probability theory and statistics
  - You can state and apply Bayes' theorem, write the density of a Gaussian distribution and of a Gaussian mixture, estimate an expectation using Monte Carlo, understand (conditional) independence
  - Not expected: measure theory, stochastic processes, advanced hypothesis testing
- Linear algebra and calculus
  - You can compute gradients of common functions and simple multivariate integrals (incl. changes of variables), are comfortable with matrix/vector notation
  - Not expected: functional analysis, differential geometry, 'advanced' linear algebra (Jordan decomposition, tensor algebra, . . .)

What you should know

- Basics of neural nets and machine learning
  - You can implement a MLP (e.g., in Torch) for classification/regression, train it with a standard optimiser, and evaluate its performance
  - Not expected: domain-specific architectures, large-scale training, ML theory
- Probability theory and statistics
  - You can state and apply Bayes' theorem, write the density of a Gaussian distribution and of a Gaussian mixture, estimate an expectation using Monte Carlo, understand (conditional) independence
  - Not expected: measure theory, stochastic processes, advanced hypothesis testing
- Linear algebra and calculus
  - You can compute gradients of common functions and simple multivariate integrals (incl. changes of variables), are comfortable with matrix/vector notation
  - Not expected: functional analysis, differential geometry, 'advanced' linear algebra (Jordan decomposition, tensor algebra, ...)

We will review necessary concepts as we need them.

Of course, more knowledge is always better than less!

Logistics

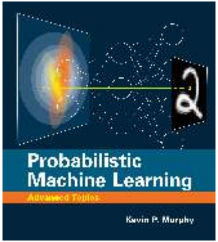
- Lectures: Tuesdays 17:10-18:00, ALT. Sorry for the late time.
- Tutorials: Mondays 13:10-14:00, 14:10-15:00, and Wednesdays 13:10-14:00, Appleton Tower M2, starting Week 3.
  - Schedule to be published online
  - Coding illustrations (Python + Torch) of lecture concepts and exercises
  - Piazza forum always open for questions/discussion

- Assessment based on final exam
  - Sample exams posted within the week

Logistics

- Lectures: Tuesdays 17:10-18:00, ALT. Sorry for the late time.
- Tutorials: Mondays 13:10-14:00, 14:10-15:00, and Wednesdays 13:10-14:00, Appleton Tower M2, starting Week 3.
  - Schedule to be published online
  - Coding illustrations (Python + Torch) of lecture concepts and exercises
  - Piazza forum always open for questions/discussion

- Assessment based on final exam
  - Sample exams posted within the week
- Recommended reference: Kevin Murphy, *Probabilistic Machine Learning: An Introduction* and ... *Advanced Topics* (esp. Chapter IV), free online



Why do the tutorials and code?

Yes, a LLM can code a diffusion model for you in a few seconds.

Why do the tutorials and code?

Yes, a LLM can code a diffusion model for you in a few seconds.But whether you want to use generative models or do research on/with them:

- Knowing how something works deeply helps you apply it effectively, be aware of its limitations, use it responsibly
- Generative modelling is an active research area; even if you are not pushing its frontiers, you may want to keep up
  - Lots of buggy code and incorrect intuitions are out there!

Why do the tutorials and code?

Yes, a LLM can code a diffusion model for you in a few seconds.But whether you want to use generative models or do research on/with them:

- Knowing how something works deeply helps you apply it effectively, be aware of its limitations, use it responsibly
- Generative modelling is an active research area; even if you are not pushing its frontiers, you may want to keep up
  - Lots of buggy code and incorrect intuitions are out there!
- **DIY everything!** (For autonomy, reliability, and the satisfaction of understanding.)

Tutorials and exercises are designed to help you understand and explore in a supportive environment.

Conclusion and looking ahead

- Generative modelling is about learning to represent, fit, and sample from approximations to data distributions
- It has many applications beyond replicating real data; understanding how models work informs their responsible use
- We will study the main methods in deep generative modelling and their uses

Conclusion and looking ahead

- Generative modelling is about learning to represent, fit, and sample from approximations to data distributions
- It has many applications beyond replicating real data; understanding how models work informs their responsible use
- We will study the main methods in deep generative modelling and their uses
- Next: a more formal introduction to distribution approximation, then on to latent variable models