Introduction

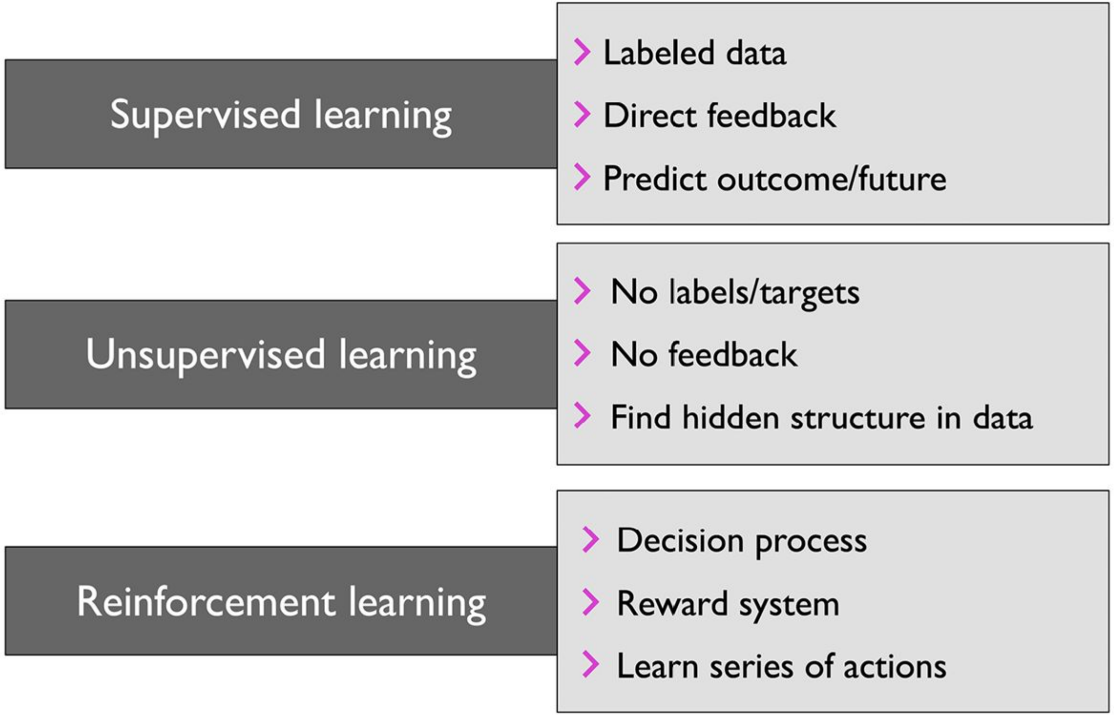
Vijay Raghavan

Agenda

1. AI/ML
2. Generative AI
3. Generative Modeling
4. Representation Learning
5. Probability Theory
6. Generative AI Family

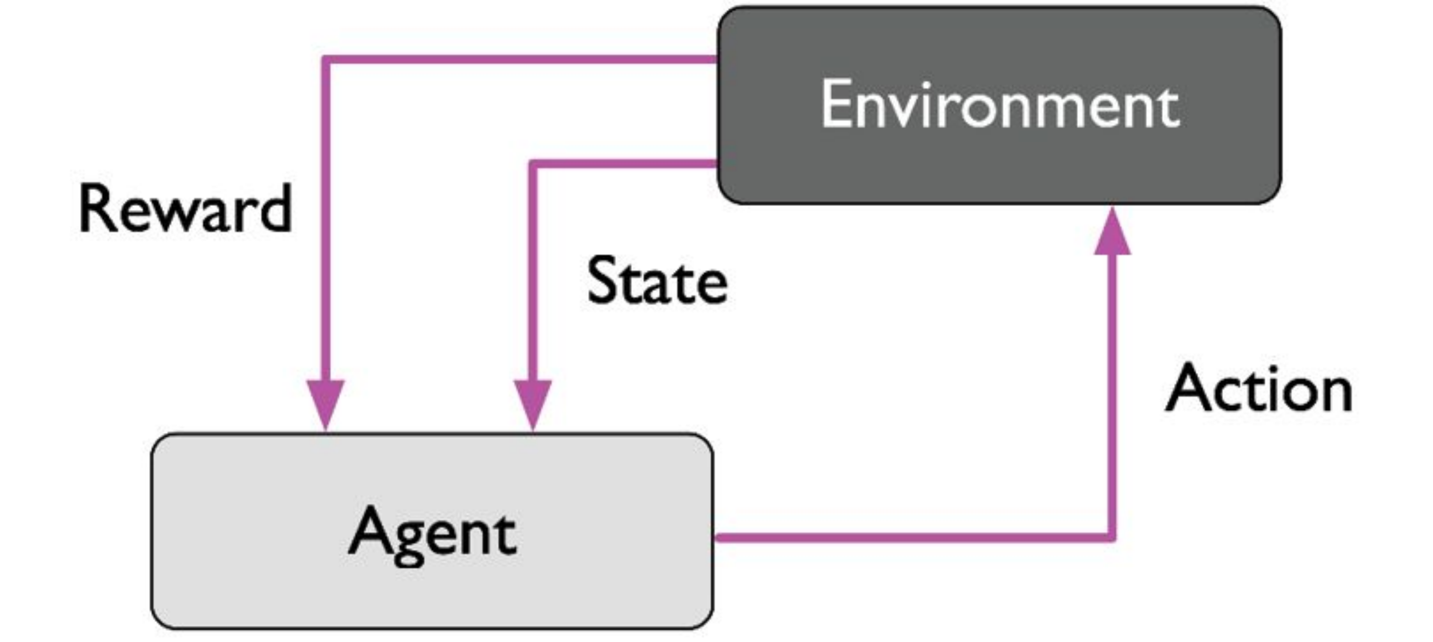
AI/ML

Types of AI/ML



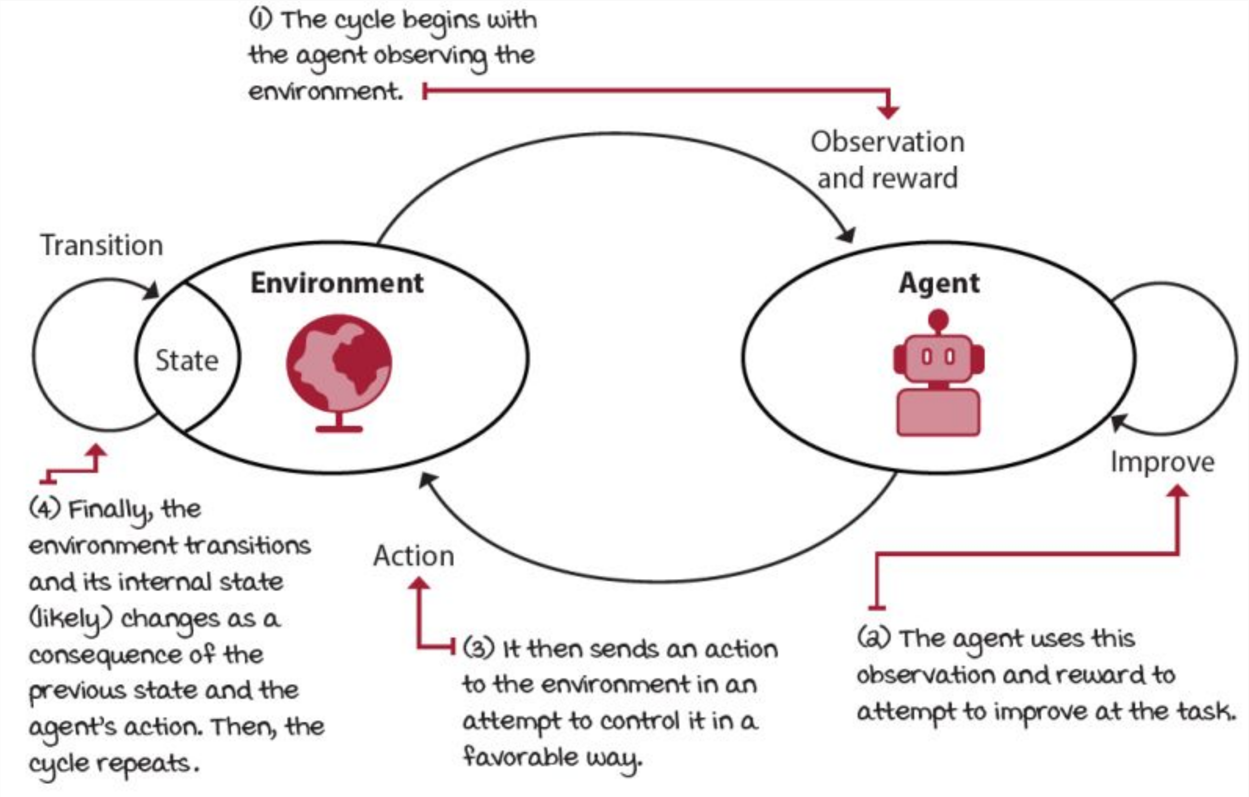
Raschka, S., Liu, Y. H., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing Ltd.

Reinforcement Learning



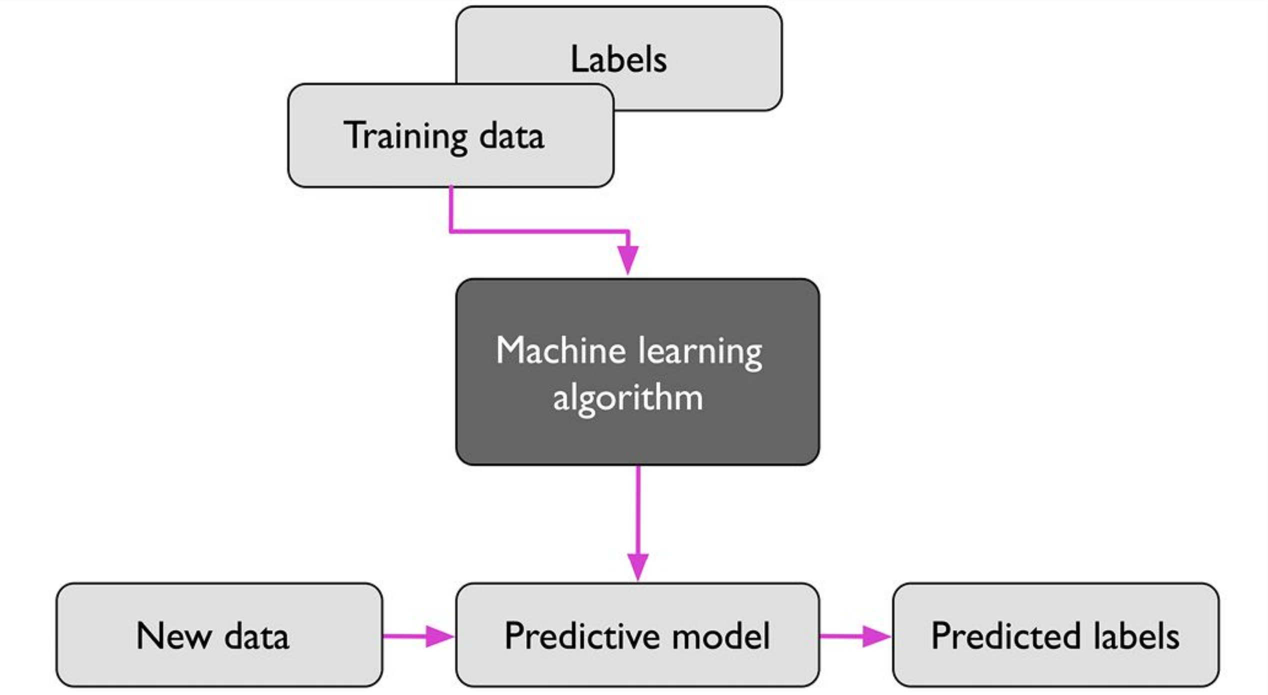
Raschka, S., Liu, Y. H., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing Ltd.

RL - Workflow



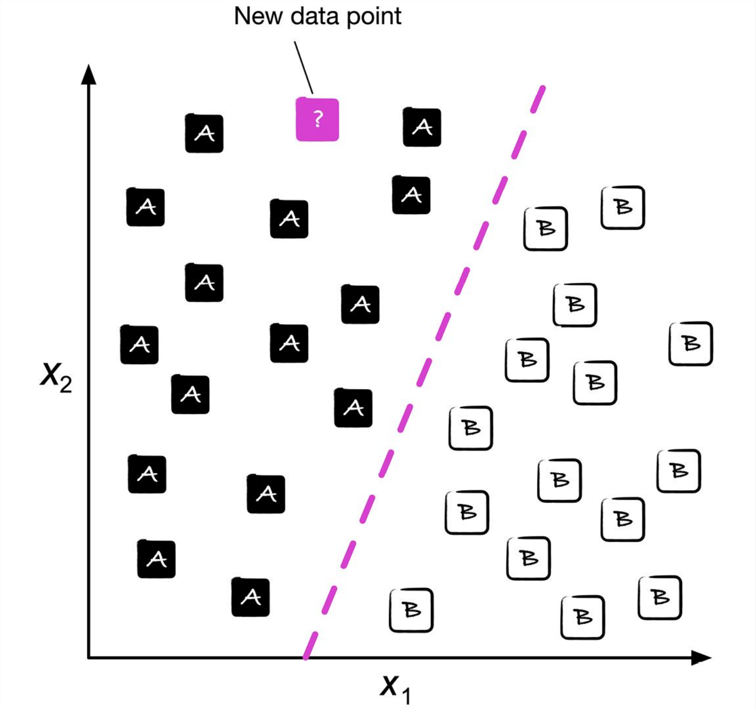
Morales, M. (2020). *Grokking deep reinforcement learning*. Manning Publications.

Supervised Learning



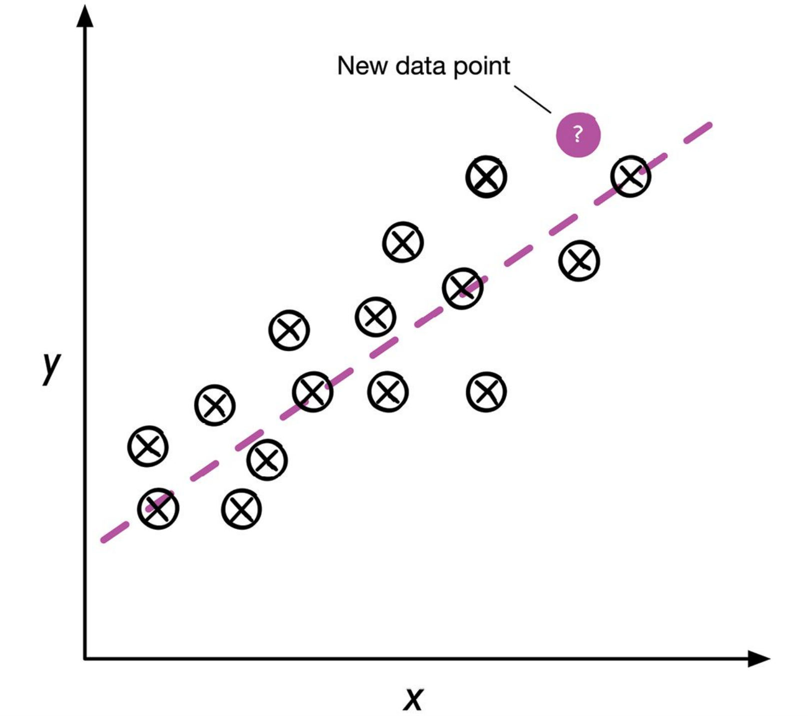
Raschka, S., Liu, Y. H., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing Ltd.

Supervised Learning - Classification for predicting class labels



Raschka, S., Liu, Y. H., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing Ltd.

Supervised Learning - Regression for predicting continuous outcomes



Raschka, S., Liu, Y. H., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing Ltd.

Discriminative Modelling

- “Discriminative models, also referred to as conditional models, are a class of logistical models used for classification or regression.”
- “They distinguish decision boundaries through observed data, such as pass/fail, win/lose, alive/dead or healthy/sick.”
- “Typical discriminative models include logistic regression (LR), conditional random fields (CRFs), decision trees, and many others.”

https://en.wikipedia.org/wiki/Discriminative_model

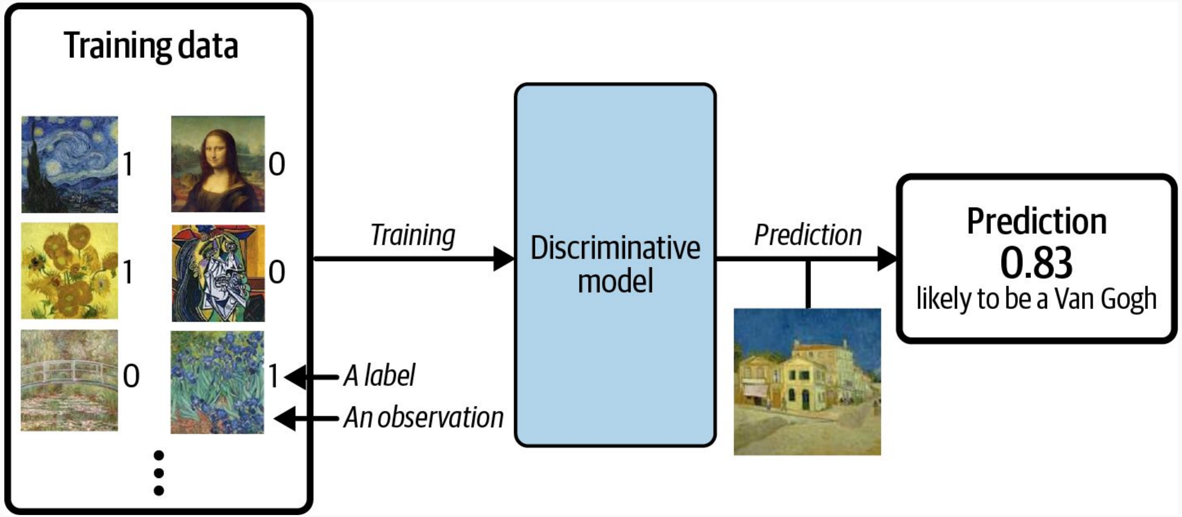
Discriminative model estimates

- Discriminative modeling estimates $p(y|\mathbf{x})$
- Discriminative modeling aims to model the probability of a label $\mathbf{y}$ given some observation $\mathbf{x}$

Discriminative Modelling - Example

- A discriminative model is used to predict labels or classes rather than generate new data points.
- As an example, we could train a discriminative model on paintings to predict whether a given painting was created by Van Gogh.
- The model would analyze features like colors, shapes, and textures that are indicative of Van Gogh's style.
- When shown a new painting with similar features, the model would predict a higher probability that it is a Van Gogh.

Discriminative Modelling - Illustration



- The training data consists of images with associated labels.
- The model learns the visual patterns associated with each label.

Foster, D. (2022). *Generative deep learning*. " O'Reilly Media, Inc.".

Word of Caution

Even if we were able to build a perfect discriminative model to identify Van Gogh paintings:

- It would still have no idea how to create a painting that looks like a Van Gogh.
- It can only output probabilities against existing images, as this is what it has been trained to do.

Unsupervised Learning

Algorithm

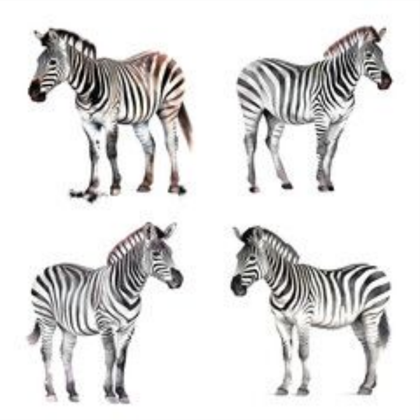
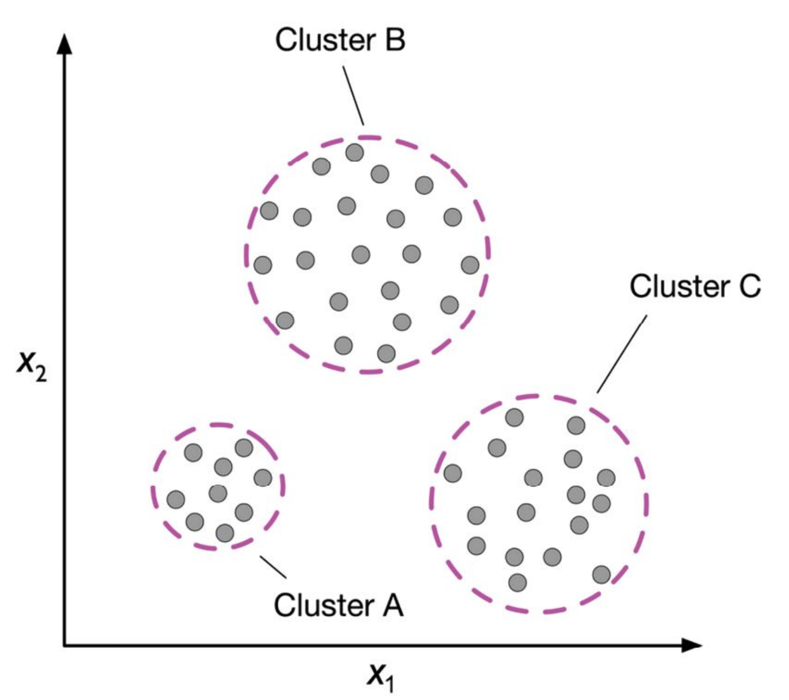


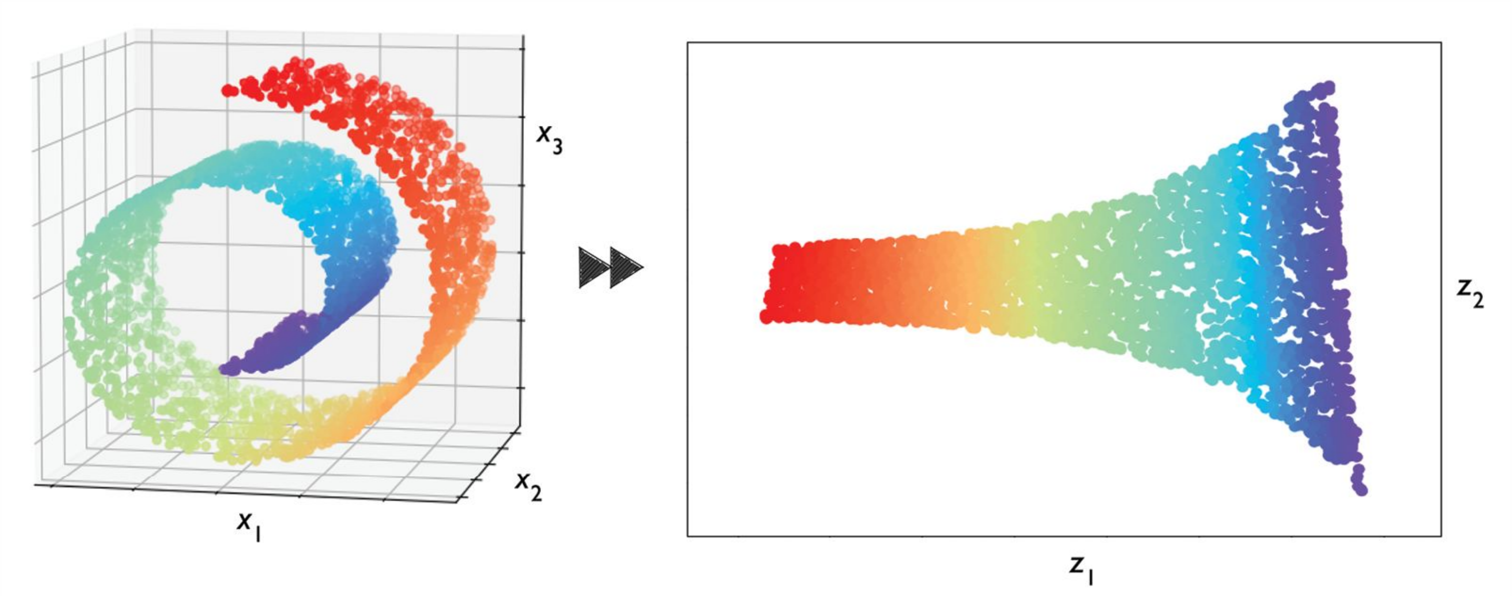
Image courtesy - Adobe

Unsupervised Learning - Clustering



Raschka, S., Liu, Y. H., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing Ltd.

Unsupervised Learning - Dimensionality Reduction



Raschka, S., Liu, Y. H., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*. Packt Publishing Ltd.

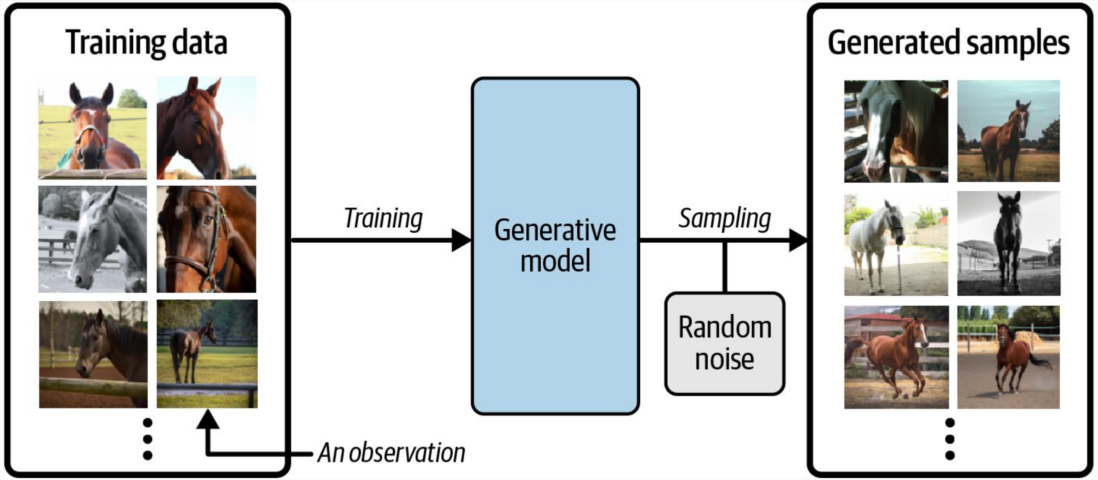
Generative AI

What Is Generative Modeling?

“Generative modeling is a branch of machine learning that involves training a model to produce new data that is similar to a given dataset.”

Foster, D. (2022). *Generative deep learning*. " O'Reilly Media, Inc.".

Generative Modeling

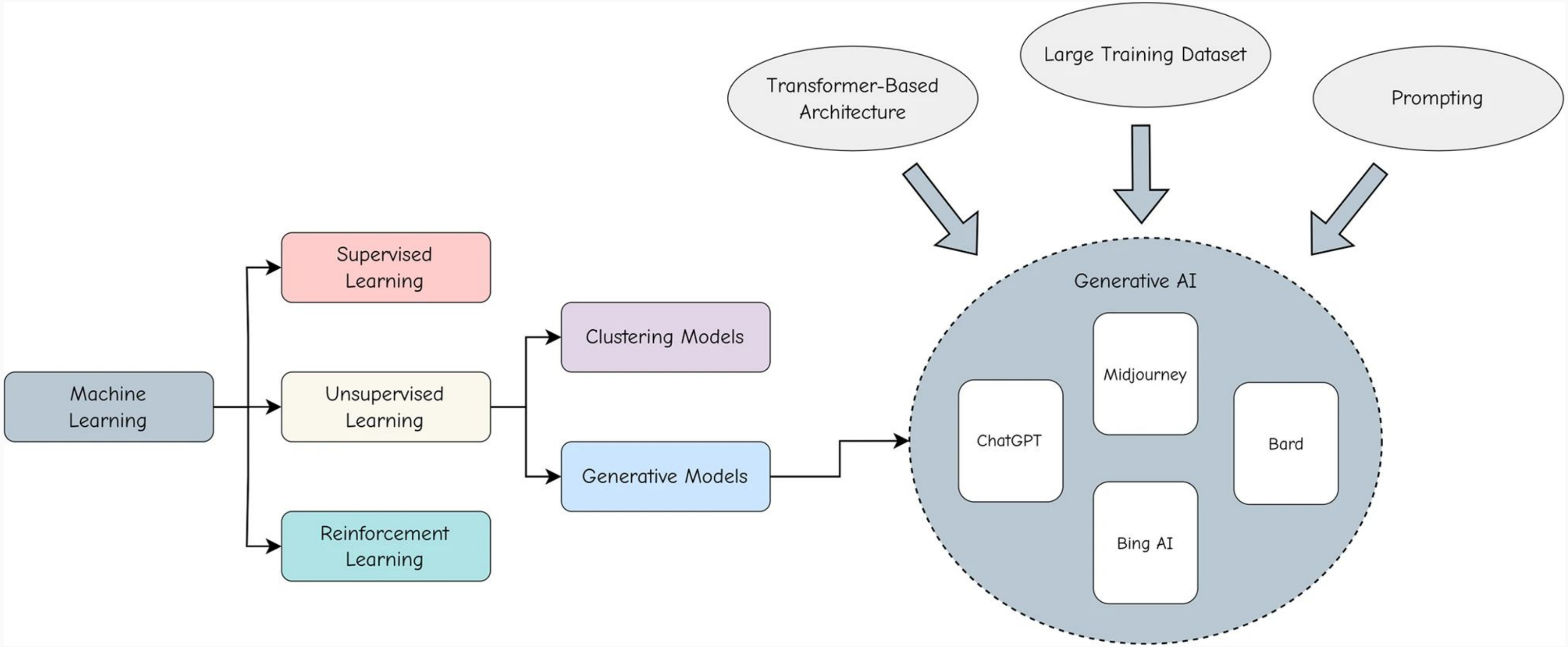


Foster, D. (2022). *Generative deep learning*. " O'Reilly Media, Inc.".

Probabilistic Vs. Deterministic

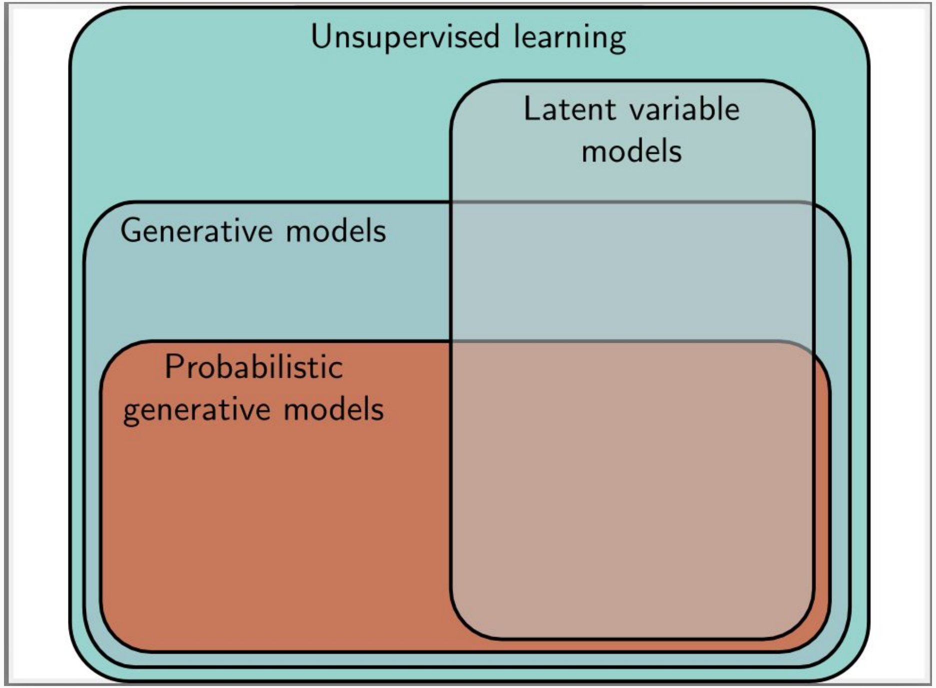
- A generative model must also be *probabilistic* rather than *deterministic*, because we want to be able to sample many different variations of the output, rather than get the same output every time.
- If our model is merely a fixed calculation, such as taking the average value of each pixel in the training dataset, it is not generative.
- A generative model must include a random component that influences the individual samples generated by the model.

The relationship between general ML and modern generative AI.



Oniani, D., Hilsman, J., Peng, Y. *et al.* Adopting and expanding ethical principles for generative artificial intelligence from military to healthcare. *npj Digit. Med.* 6, 225 (2023). https://doi.org/10.1038/s41746-023-00965-x

Taxonomy of unsupervised learning models



- Generative models can generate new examples with similar statistics to the training data.
- A subset of these are probabilistic and define a distribution over the data. We draw samples from this distribution to generate new examples.
- Latent variable models define a mapping between an underlying explanatory (latent) variable and the data.

Prince, S. J. (2023). *Understanding Deep Learning*. MIT press.

Training Data for Generative Modelling

- Training data consists of many observations of the target entity (e.g. images of horses)
- An observation represents one data point (e.g. a single horse photo)
- Observations are characterized by features (e.g pixels for images, words for text)
- Features capture different attributes of each observation
- Goal is to learn intricate rules governing relationships between features that define the entity

Goals of Generative Modelling

- Generate completely novel observations
- New feature combinations should mimic training data patterns
- Enormously challenging due to exponential combination possibilities
- Vast majority of arrangements don't resemble plausible observations
- Model must have randomized component to produce variation
- Cannot be fixed calculation like averaging feature values

True Generating Distribution

- True but unknown complex distribution generates real observations
- Some observations are very likely under this distribution
- Others are improbable or impossible
- Model distribution attempts to mimic true distribution
- Then we can sample from model distribution
- Create new observations that capture nuances of true distribution

Why is this a Hard Problem?

- True distribution exists in exponentially high dimensional space
- Observe only a sparse subset of possibilities during training
- Must creatively generalize to produce innovative but realistic new data points
- Balance novelty and plausibility
- Intractable to explicitly model full dimensionality

Generative Modeling estimates

- Generative modeling estimates $p(\mathbf{x})$
- Generative modeling aims to model the probability of observing an observation $\mathbf{x}$
- Sampling from this distribution allows us to generate new observations.

Conditional Generative Models

- We can also build a generative model to model the conditional probability $p(\mathbf{x}|y)$
- The probability of seeing an observation $\mathbf{x}$ with a specific label $\mathbf{y}$
- For example, if our dataset contains different types of fruit, we could tell our generative model to specifically generate an image of an apple.

The Rise of Generative Modeling

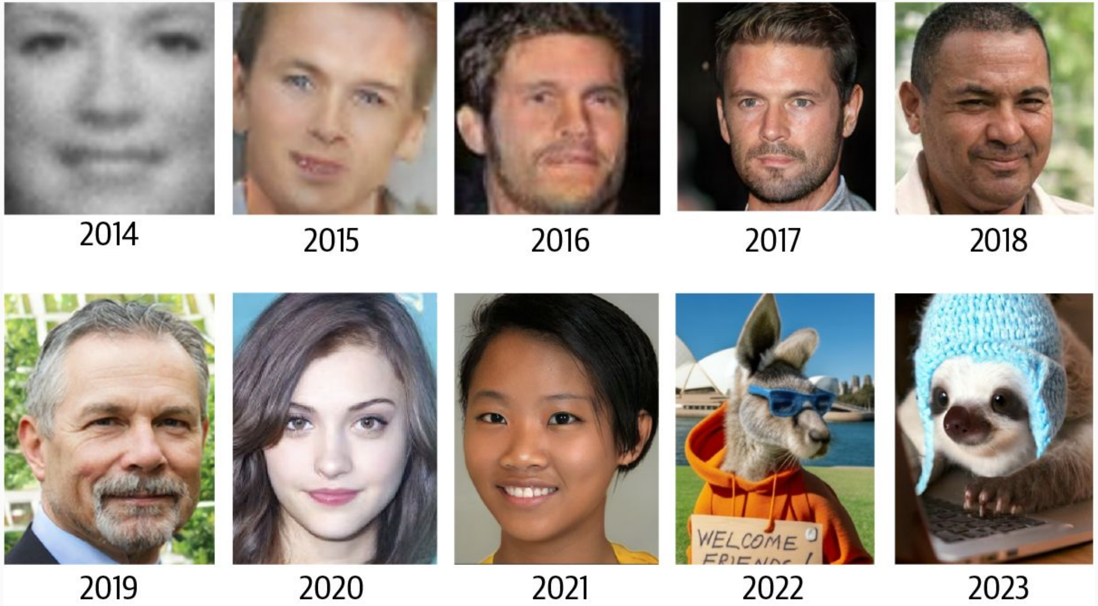
Historically…

- Discriminative models have driven majority of machine learning progress
  - Easier to implement and apply to real-world problems
- Categorize data based on labels rather than create from scratch
- Examples:
  - Classify painting as Van Gogh or not Van Gogh
  - Identify if text was written by Charles Dickens
- Corresponding generative tasks seen as extraordinarily challenging

Recent Improvements in Generative Modeling

- In the last decade, generative modeling has seen major advances
- Powered by progress in deep learning techniques
- Enabled modeling of increasingly complex distributions
- Surpassing prior expectations on feasibility

Face generation using generative modeling has improved significantly over the last decade



Brundage et al., 2018

Industry - Applied discriminative models

- Historically, industry focused applied discriminative models more
- Doctors benefited from diagnostic predictions, not eye image creation
- Shift as companies provide generative modeling as a service
- API-based solutions for goals like:
    - Automated content generation
    - Visual product configs and rendering
    - Brand-tailored advertising and social posts

Relevance for AI

- Human creativity and generative capacity considered exceptional
- Seen as insurmountable challenge for artificial intelligence
- Rapid improvements imply this assumption needs reevaluating
- Sophisticated generative modeling essential for future AI progress
- Better understanding of data distributions and achieving human parity

Generative Modeling Framework

Framework

- We have a dataset of observations $\mathbf{X}$
- We assume that the observations have been generated according to some unknown distribution $p_{data}(x)$
- We want to build a generative model $p_{model}(x)$ that mimics $p_{data}(x)$
- If we achieve this goal, we can sample from $p_{model}(x)$
- To generate observations that appear to have been drawn from $p_{data}(x)$

Credit: Shon Mohsin

Desirable properties of the model

Accuracy

If $p_{model}(x)$ is high for a generated observation, it should look like it has been drawn from $p_{data}(x)$

If $p_{model}(x)$ is low for a generated observation, it should *NOT* look like it has been drawn from $p_{data}(x)$

Generation

It should be possible to easily sample a new observation from $p_{model}(x)$

Representation

It should be possible to understand how different high-level features in the data are represented by $p_{model}(x)$

.

Representation Learning

Simplifying High dimensional data

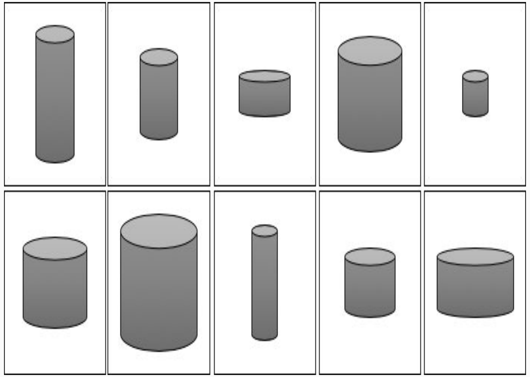
- To describe your appearance to someone who doesn't know what you look like, you wouldn't list every pixel color in your image.
- Instead, you would assume they have a general idea of a human's appearance, and describe differences using high-level features like hair color and whether you wear glasses.
- With around 10 such descriptive features, they could generate a rough image of you in their mind and pick you out of a crowd, even having never seen you.
- The image wouldn't be perfect, but good enough to identify you using the key features rather than all pixel values.

Core Idea

- Rather than directly modeling extremely high-dimensional data like images, representation learning involves encoding the data in a lower-dimensional latent space.
- Each point in this latent space represents a particular observation from the training data.
- A mapping function is learned which maps points from this compact latent space to points in the original, complex domain.
- So in a sense, each latent point acts as an efficient "representation" which summarizes the most important aspects of a higher-dimensional observation.

Biscuit tins

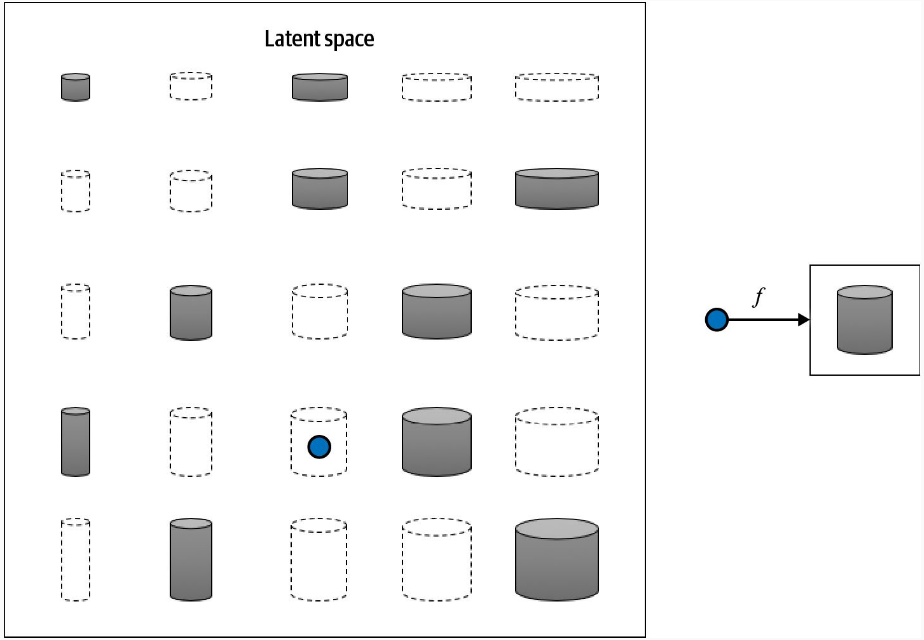
- For a dataset of biscuit tin images, the height and width are two features that can uniquely represent each image.
- So rather than modeling the images in high-dimensional pixel space, we can convert them to points in a compact 2D latent space of height and width.
- We can then sample new points from this space and apply a learned mapping function to generate new biscuit tin images, including ones not in the original training set.
- It is nontrivial for a machine to identify these latent features and mapping on its own, without guidance.
- Representation Learning gives models the ability to learn these relationships between domains automatically from data.



Foster, D. (2022). *Generative deep learning*. " O'Reilly Media, Inc.".

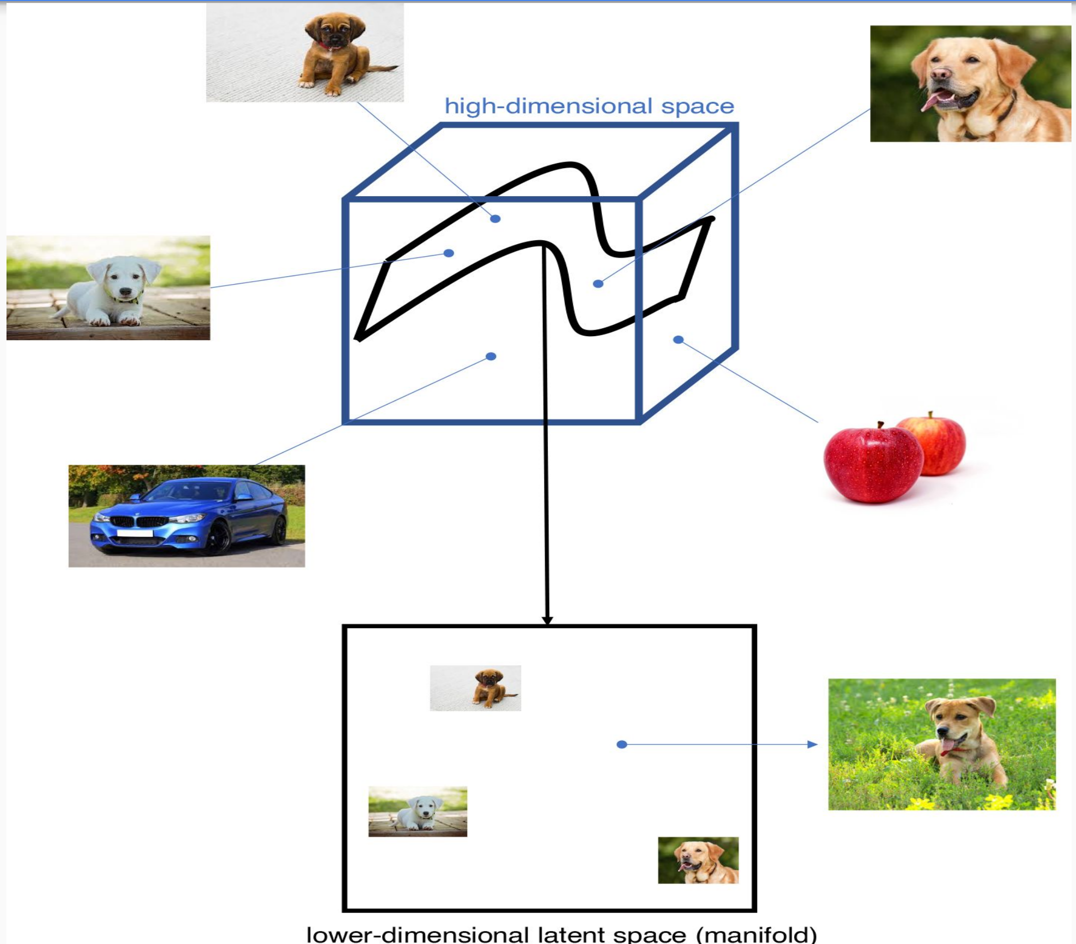
Latent Space

The 2D latent space of biscuit tins and the function
 that maps a point in the latent space back to the original image domain



Foster, D. (2022). *Generative deep learning*. " O'Reilly Media, Inc.".

High to Lower Dimensional space



The cube represents the extremely high-dimensional space of all images; representation learning tries to find the lower-dimensional latent subspace or *manifold* on which particular kinds of image lie (for example, the *dog* manifold)

Foster, D. (2020). *Generative deep learning*. " O'Reilly Media, Inc.".

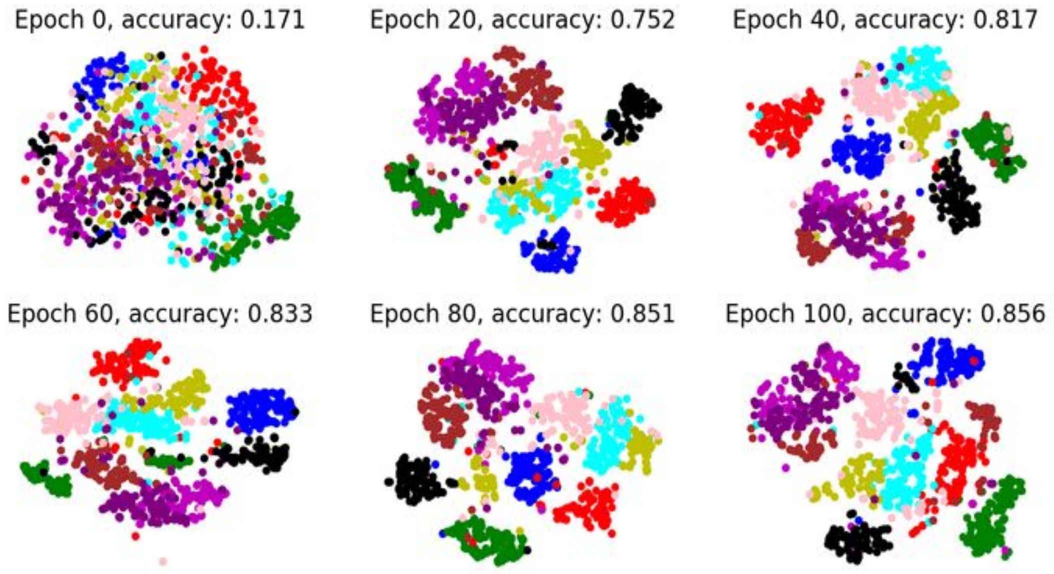
Latent Space Captures Essence

- Latent space abstracts core aspects of data
- Uses much fewer dimensions than original data
- Points represent values of key features for each data sample
- Example: Height and width attributes describe a biscuit tin image well
- Map these compact representations back to full images

Simplifies Operations

- Editing the latent point changes the generated image appropriately
- Much more straightforward than altering every pixel
- e.g. in the Biscuit tin case just tweak height value to stretch the image vertically

Visualizing the Latent Space



Latent vectors from the MNIST

projected to a 2-D space

Zhang, D., Sun, Y., Eriksson, B., & Balzano, L. (2017). Deep unsupervised clustering using mixture of autoencoders. 12 2017. *URL http://arxiv. org/abs/1712, 7788*, 2-3.

Probability Theory

Sample Space

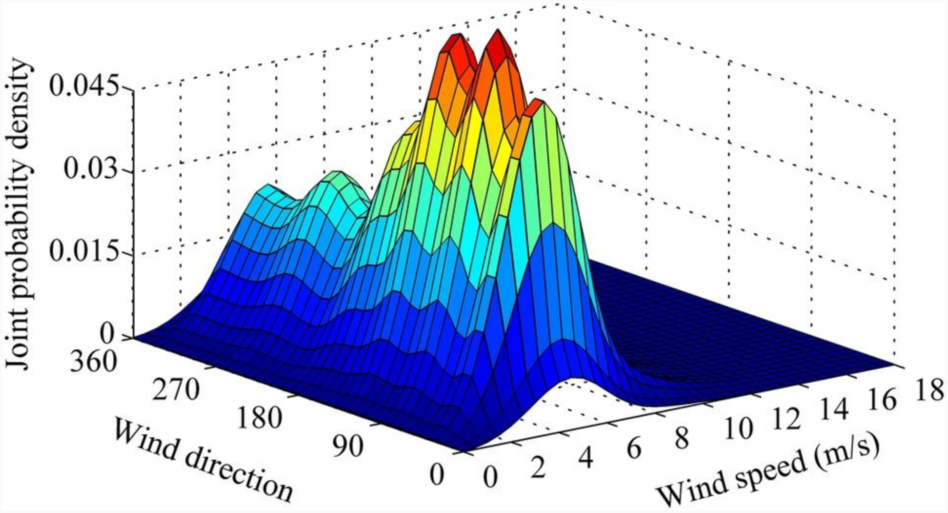
- Sample space - is the complete set of all values an observation $\mathbf{x}$ can take
- If the sample space consists of all points of latitude and longitude $\mathbf{x} = (x_1, x_2)$ on the world map.
- For example,
  - $\mathbf{x} = (40.7306, -73.9352)$ is a point in the sample space (New York City) that belongs to the true data-generating distribution.
  - $\mathbf{x} = (11.3493, 142.1996)$ is a point in the sample space that does not belong to the true data-generating distribution (it's in the sea).

Foster, D. (2022). *Generative deep learning*. " O'Reilly Media, Inc.".

Probability Density Function (PDF)

- A *probability density function* is a function $p(\mathbf{x})$ that describes the relative likelihood for a continuous random variable to take on a given value.
- The integral of the density function over all points in the sample space must equal 1, so that it is a well-defined probability distribution.

Joint probability density of wind speed and direction.



Wang, Z., & Liu, W. (2021). Wind energy potential assessment based on wind speed, its direction and power data. *Scientific reports*, *11*(1), 16879.

True Density Function

- There is only one true density function $p_{data}(x)$ that is assumed to have generated the observable dataset
- However, there are infinitely many density functions $p_{model}(x)$ that we can use to estimate $p_{data}(x)$

Parametric modeling

- Parametric modeling is a technique that we can use to structure our approach to finding a suitable $p_{model}(x)$
- A parametric model is a family of density functions $p_\theta(x)$ that can be described using a finite number of parameters θ

Likelihood

The *likelihood* $\mathcal{L}(\theta|x)$ of a parameter set θ is a function that measures the plausibility of θ given some observed point $\mathbf{x}$ is $\mathcal{L}(\theta|x) = p_\theta(x)$

i.e. the likelihood of θ given some observed point $\mathbf{x}$ is defined to be the value of the density function parameterized by θ

Independent Observations

If we have a whole dataset $\mathbf{X}$ of independent observations, then we can write:

$$\mathcal{L}(\theta|x) = \prod_{x \in X} p_\theta(x)$$

Log Likelihood

The product of a large number of terms between 0 and 1 can be quite computationally difficult to work with, we often use the log-likelihood ℓ instead:

$$l\,(\theta|\mathbf{X}) = \sum_{x \in X} log\ p_\theta(\mathbf{x})$$

The likelihood of a set of parameters θ is defined to be the probability of seeing the data if the true data-generating distribution was model parameterized by θ

Maximum likelihood estimation

Maximum likelihood estimation is the technique that allows us to estimate $\hat{\theta}$ —the set of parameters θ of a density function $p_\theta(x)$ that is most likely to explain some observed data $\mathbf{X}$

$$\hat{\theta} = \arg\max_{x} \; l(\theta|\mathbf{X})$$

Generative modeling and MLE

- Generative modeling can be thought of as a form of maximum likelihood estimation.
- Where the parameters θ are the weights of the neural networks contained in the model.
- We are trying to find the values of these parameters that maximize the likelihood of observing the given data (or equivalently, minimize the negative log-likelihood).

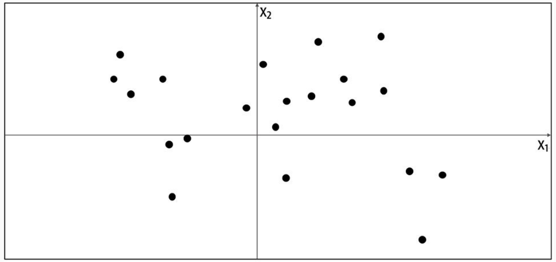
Issues with high dimensions

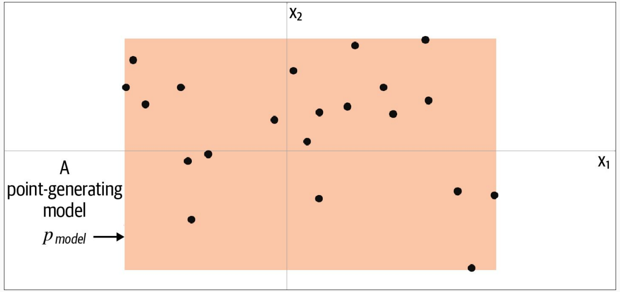
- For high-dimensional problems, it is generally not possible to directly calculate $p_\theta(x)$ —it is intractable.
- We will different families of generative models take different approaches to tackling this problem.

First Generative Model

Hello World!

- We have chosen a rule $p_{data}(x)$ that has been used to generate the set of points $\mathbf{X}$
- The challenge is to choose a different point $\boldsymbol{x} = (x_1, x_2)$ in the space that looks like it has been generated by the same rule.

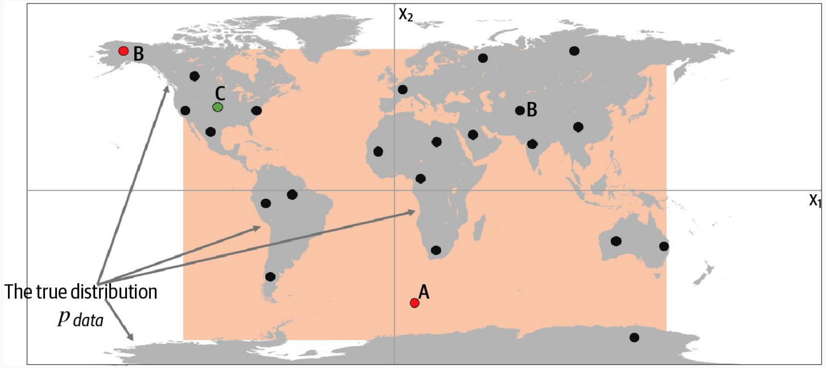




The orange box $p_{model}(x)$ is an estimate of the true data-generating distribution $p_{data}(x)$

Generative Modeling Framework

- We have a dataset of observations $\mathbf{X}$
- We assume that the observations have been generated according to some unknown distribution $p_{data}(x)$
- We want to build a generative model $p_{model}(x)$ that mimics $p_{data}(x)$
- If we achieve this goal, we can sample from $p_{model}(x)$ to generate observations that appear to have been drawn from $p_{data}(x)$



The orange box, $p_{model}(x)$ is an estimate of the true data-generating distribution, $p_{data}(x)$ (the gray area)

Sample Space

The sample space consists of all points of latitude and longitude $\boldsymbol{x} = (x_1, x_2)$ on the map.

For example,

- $\mathbf{x}$ = (40.7306, –73.9352) is a point in the sample space (New York City) that belongs to the true data-generating distribution.
- $\mathbf{x}$ = (11.3493, 142.1996) is a point in the sample space that does not belong to the true data-generating distribution (it’s in the sea).

Probability density function

In the world map example, the density function of our generative model is;

- 0 outside of the orange box
- Constant inside of the box
- The integral of the density function over the entire sample space equals 1.

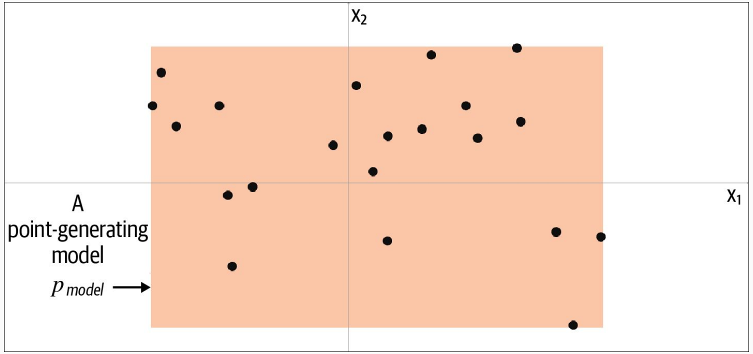
Parametric modeling

- If we assume a uniform distribution as our model family, then the set all possible boxes we could draw is an example of a parametric model.
- In this case, there are four parameters: the coordinates of the bottom-left (θ_1, θ_2) and top-right (θ_3, θ_4) corners of the box.
- Thus, each density function $p_\theta(x)$ in this parametric model (i.e., each box) can be uniquely represented by four numbers $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$

.

Likelihood

- In the world map example, an orange box that only covered the left half of the map would have a likelihood of 0
- It couldn't possibly have generated the dataset, as we have observed points in the right half of the map.
- The orange box has a positive likelihood, as the density function is positive for all data points under this model.



Generative AI Family

Approaches

1. Explicitly model the density function, but constrain the model in some way, so that the density function is tractable (i.e., it can be calculated).
2. Explicitly model a tractable approximation of the density function.
3. Implicitly model the density function, through a stochastic process that directly generates data.

Implicit density models

- Implicit density models do not aim to estimate the probability density at all, but instead focus solely on producing a stochastic process that directly generates data.
- The best-known example of an implicit generative model is a generative adversarial network.
- We can further split explicit density models into those that directly optimize the density function (tractable models) and those that only optimize an approximation of it.

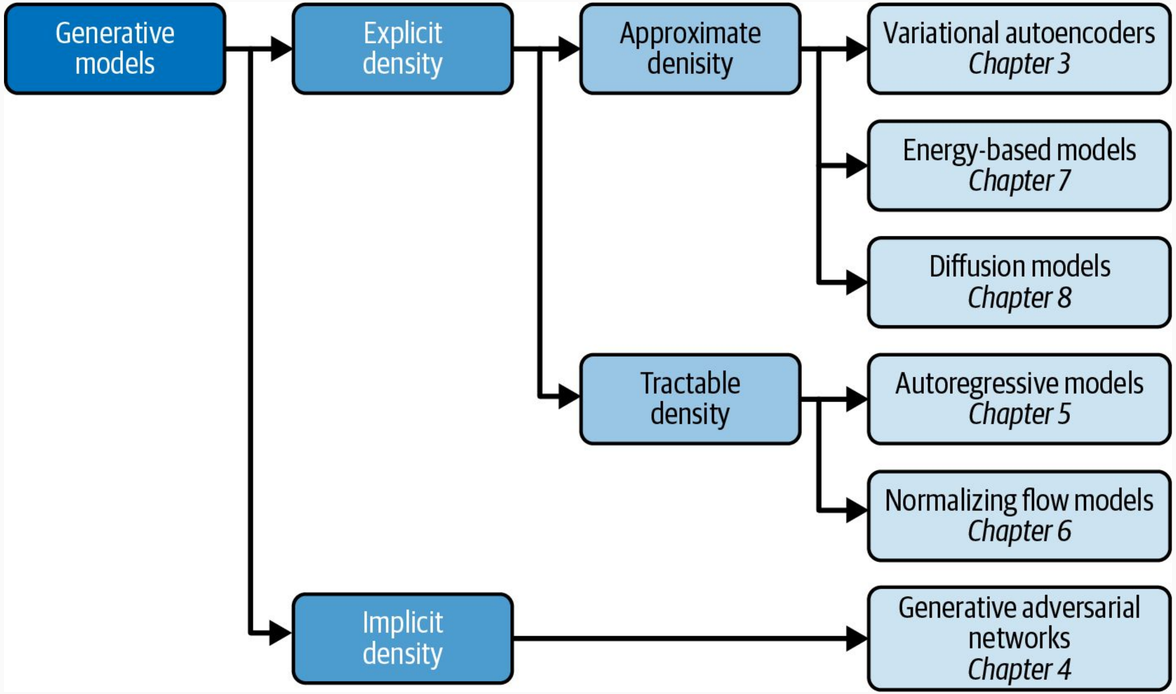
Tractable models

- Tractable models place constraints on the model architecture, so that the density function has a form that makes it easy to calculate.
- For example, autoregressive models impose an ordering on the input features, so that the output can be generated sequentially—e.g., word by word, or pixel by pixel.
- Normalizing flow models apply a series of tractable, invertible functions to a simple distribution, in order to generate more complex distributions.

Approximate density models

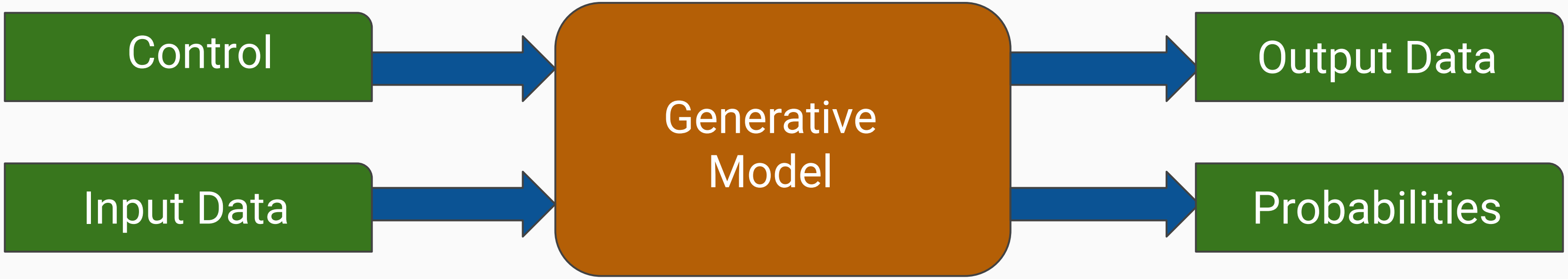
- Approximate density models include variational autoencoders, which introduce a latent variable and optimize an approximation of the joint density function.
- Energy-based models also utilize approximate methods, but do so via Markov chain sampling, rather than variational methods.
- Diffusion models approximate the density function by training a model to gradually denoise a given image that has been previously corrupted

A taxonomy of generative modeling approaches

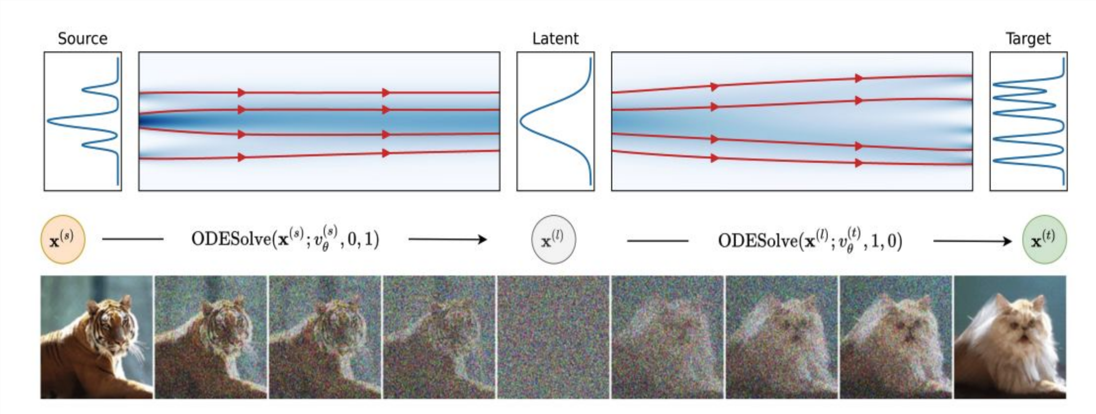


Gen AI Ecosystem

Process


Control
Input Data
Generative Model
Output Data
Probabilities

Dual Diffusion Implicit Bridges

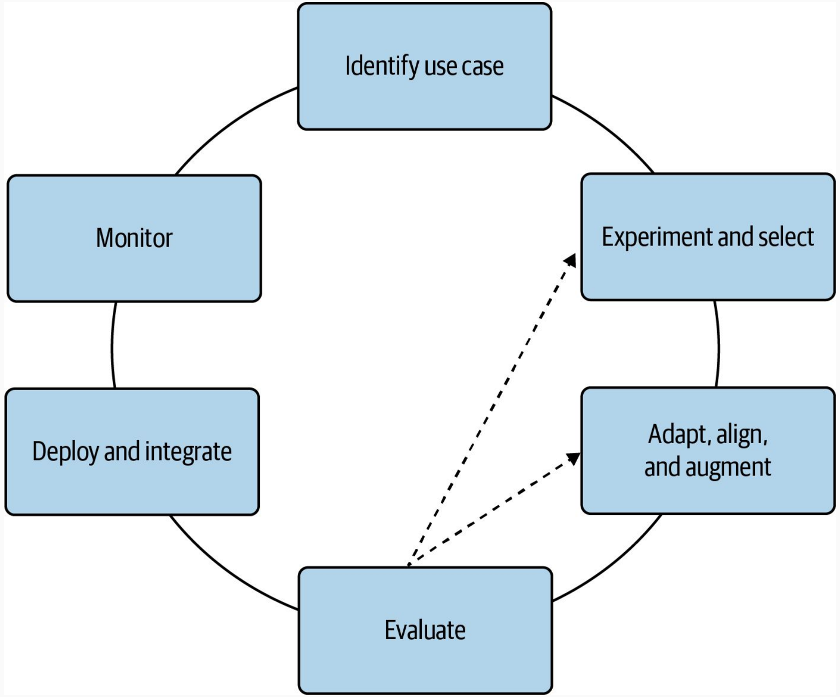


- Given a source image x (s) , the source ODE runs in the forward direction to convert it to the latent x (l) , while the target, reverse ODE then constructs the target image x (t) .
- (Top) Illustration of the DDIBs idea between two one-dimensional distributions.
- (Bottom) DDIBs from a tiger to a cat using a pre-trained conditional diffusion model.

Su, X., Song, J., Meng, C., & Ermon, S. (2022). Dual diffusion implicit bridges for image-to-image translation. *arXiv preprint arXiv:2203.08382*.

Generative AI Project Life Cycle & Ecosystem

Generative AI Project Life Cycle



Identify and Select

Identify Use Case

- Define scope and specific generative task
- Start with single, well-defined use case
- Evaluate different models for fit

Experiment and Select Model

- Try out existing foundation models
- Start small (7B parameters) to iterate quickly
- Compare on playground like SageMaker JumpStart

Adapt and Evaluate

Adapt, Align and Augment

- Fine-tune model on custom data
- Align to human values (helpful, honest, harmless)
- Augment with external data sources

Evaluate

- Establish metrics to measure effectiveness
- Evaluate alignment to business goals

Deploy and Monitor

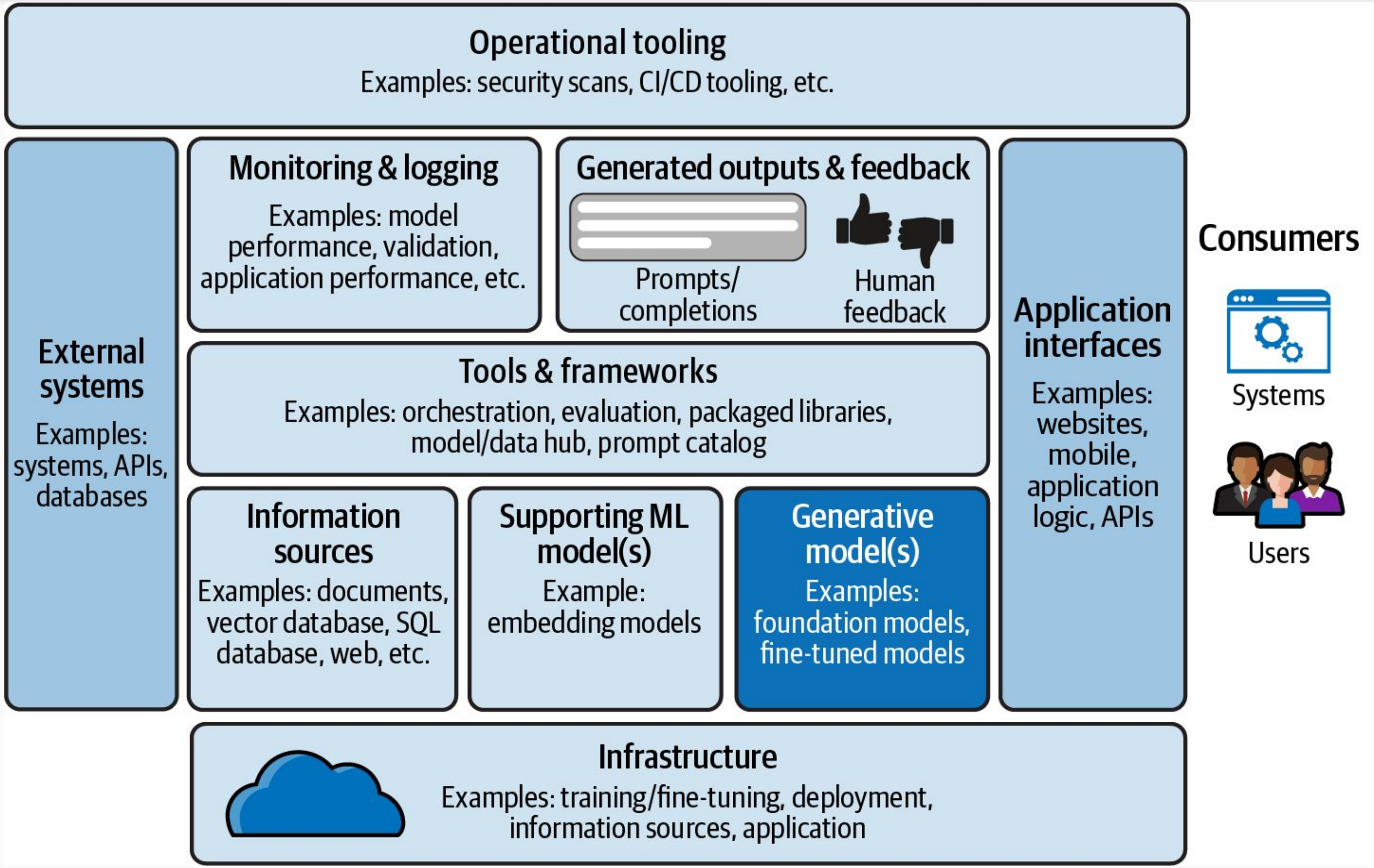
Deploy and Integrate

- Optimize for inference before deployment
- Use SageMaker endpoints for scalable serving
- Integrate into applications

Monitor

- Collect metrics with CloudWatch
- Monitor systems with CloudTrail
- Works well with Bedrock for generative AI

Gen AI Ecosystem



Generative AI on AWS_By Chris Fregly, Antje Barth, Shelbee Eigenbrode, O'Reilly Media

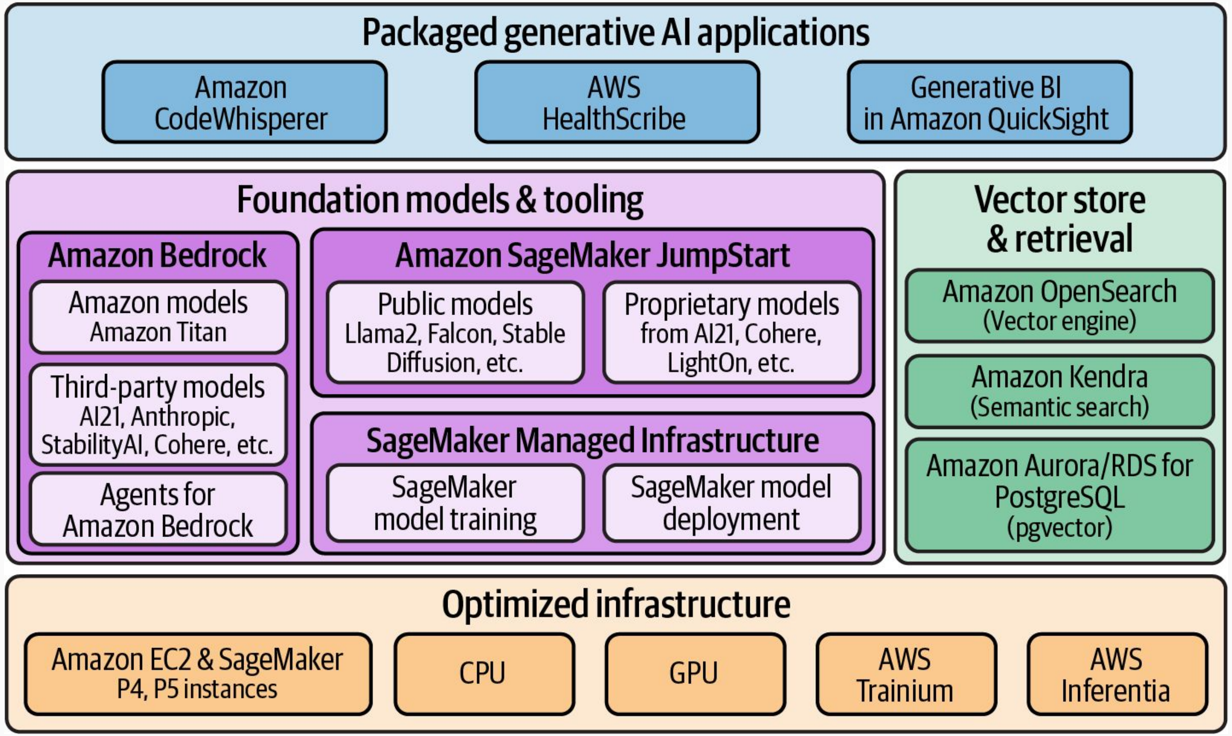
Components

- A generative AI application requires multiple components beyond just the generative model itself to build a reliable, scalable, and secure system.
- When using a fully-managed service like Amazon Code Whisperer, these components are all handled for you.

Gen AI Models need....

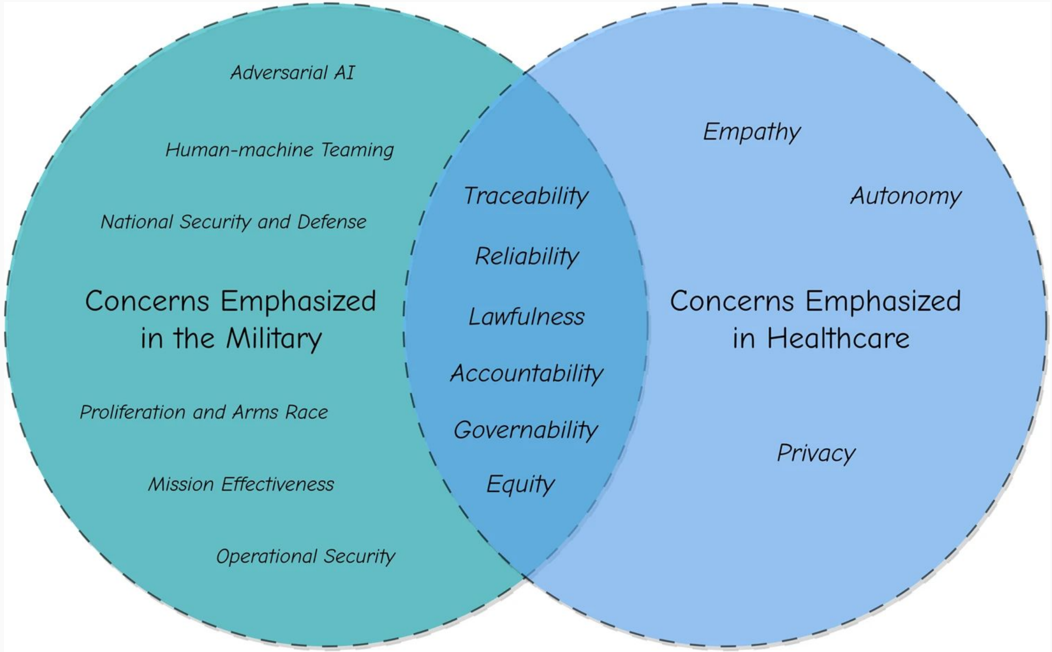
1. Compute - For model training/inference
2. Storage - For datasets and model artifacts
3. Orchestration - To coordinate workflow
4. Monitoring - To track system health
5. Security - For access control, encryption
6. Others - Integration tools, pipelines, etc.

Gen AI Ecosystem in AWS



Generative AI on AWS_By Chris Fregly, Antje Barth, Shelbee Eigenbrode, O'Reilly Media

Ethical issues



Oniani, D., Hilsman, J., Peng, Y. *et al.* Adopting and expanding ethical principles for generative artificial intelligence from military to healthcare. *npj Digit. Med.* 6, 225 (2023). https://doi.org/10.1038/s41746-023-00965-x

This Week

- Pick a Generative AI use case for your company or your organization and explain the pros and cons in less than 200 words.
- Run Class-1-LatentSpace-PyTorch.ipynb on colab. Either at the end of the notebook or at the beginning add a text cell and in a paragraph or two explain your understand of the code and what you see in the results.

Github Repo for the class: https://github.com/vijaygwu/SEAS8525/