CPSC 440/540: Advanced Machine Learning

Image Generative Models + Course Wrap-Up

Danica Sutherland

University of British Columbia

Winter 2023

Generative Image Models

- Given $x^1, \ldots, x^n \overset{\text{iid}}{\sim} p_{\text{target}}$,
  we'd like to fit a model $p_\theta \approx p_{\text{target}}$, to:
  - Discover underlying structure in the data
  - Find representative data points / modes
  - Detect outliers, anomalies
  - Impute missing values (in-painting)
  - Produce "more samples"
  - Use as a prior for semi-supervised learning, guided sampling, ...
  - ...

https://www.reddit.com/r/midjourney/comments/120vhdc/the_pope_drip/

Last Time: Variational Auto-encoders

- Deep latent variable model: $p_\theta(x) = \int p_\theta(x \mid z)\, p_\theta(z)\, \mathrm{d}z$
  - Prior distribution over latent codes $p_\theta(z)$; usually $\mathcal{N}(0, I)$, $\dim(z) \approx 200$
  - Decoder network $p_\theta(x \mid z)$: usually $\mathcal{N}(f_\theta(z), \sigma^2 I)$ for deterministic net f_θ
- Hard to do the 200-dimensional integral to compute likelihoods (e.g. for MLE)
  - Encoder network $q_\phi(z \mid x)$ "amortizes inference"
    - Usually $\mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$, with $\Sigma_\phi(x)$ typically diagonal
- For approximate MLE, maximize the average ELBO:

$$\mathrm{ELBO}_{\theta,\phi}(x) = \underbrace{\mathop{\mathbb{E}}_{z\sim q_\phi(z|x)}[\log p_\theta(x \mid z)]}_{\text{Monte Carlo est. with reparameterization trick}} - \underbrace{\mathrm{KL}(q_\phi(z \mid x) \,\|\, p_\theta(z))}_{\text{usually closed-form for given } x,\ \phi} \le \log p_\theta(x)$$

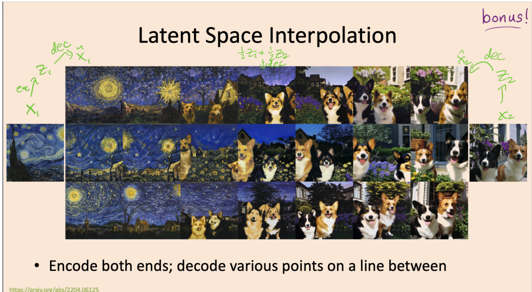
Last Time: Variational Auto-Encoders

- e.g. VQ-VAE-2: discrete hierarchical latents, learned autoregressive prior

Figure 1: Class-conditional 256x256 image samples from a two-level model trained on ImageNet.

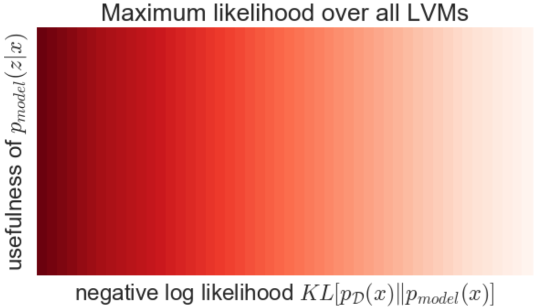
`https://arxiv.org/pdf/1906.00446.pdf`

- Latents sometimes "meaningful" (especially "disentangled": β-VAE/TC-VAE/...)



Representation Learning with Latent Variable Models

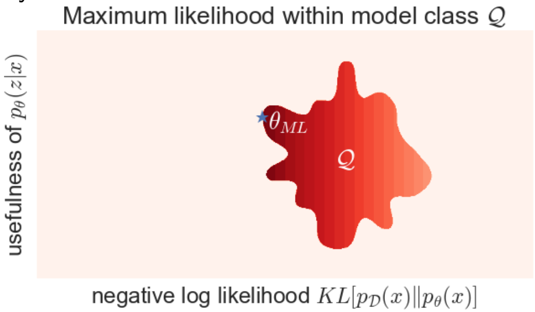
- We'd often like a "useful" $p_\theta(z \mid x)$
- Maximum likelihood minimizes KL between target and $p_\theta(x) = \int p_\theta(x, z)\mathrm{d}z$
- Objective wants a good fit for $p_\theta(x)$; doesn't care about usefulness at all
  - True for *any* objective that only cares about $p_\theta(x)$, not just MLE



https://www.inference.vc/maximum-likelihood-for-representation-learning-2/

Representation Learning with Latent Variable Models

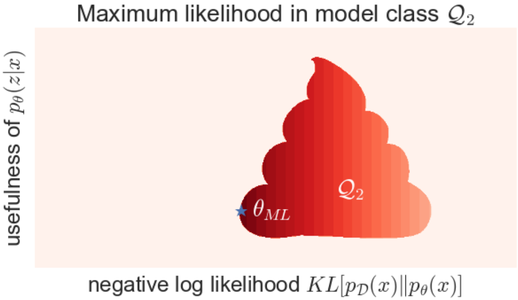
- We'd often like a "useful" $p_\theta(z \mid x)$
- Maximum likelihood minimizes KL between target and $p_\theta(x) = \int p_\theta(x, z)\mathrm{d}z$
- Objective wants a good fit for $p_\theta(x)$; doesn't care about usefulness at all
    - True for *any* objective that only cares about $p_\theta(x)$, not just MLE
- But we don't actually maximize over all latent variable models



https://www.inference.vc/maximum-likelihood-for-representation-learning-2/

Representation Learning with Latent Variable Models

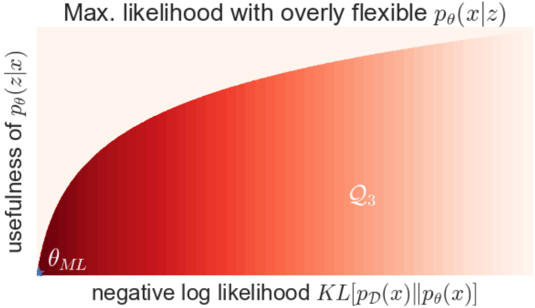
- We'd often like a "useful" $p_\theta(z \mid x)$
- Maximum likelihood minimizes KL between target and $p_\theta(x) = \int p_\theta(x, z) \mathrm{d}z$
- Objective wants a good fit for $p_\theta(x)$; doesn't care about usefulness at all
  - True for *any* objective that only cares about $p_\theta(x)$, not just MLE
- But we don't actually maximize over all latent variable models
- This relies on our model class (or really, learning process...) aligning well



https://www.inference.vc/maximum-likelihood-for-representation-learning-2/

Representation Learning with Latent Variable Models

- We'd often like a "useful" $p_\theta(z \mid x)$
- Maximum likelihood minimizes KL between target and $p_\theta(x) = \int p_\theta(x, z) \mathrm{d}z$
- Objective wants a good fit for $p_\theta(x)$; doesn't care about usefulness at all
  - True for *any* objective that only cares about $p_\theta(x)$, not just MLE
- But we don't actually maximize over all latent variable models
- This relies on our model class (or really, learning process. . .) aligning well
- Real(ish) case: if $p_\theta(x \mid z)$ is too powerful, can ignore z, i.e. useless representation



https://www.inference.vc/maximum-likelihood-for-representation-learning-2/

Representation Learning with VAEs

bonus!

- Maximizing the ELBO isn't *just* MLE...

$$\max_\phi \sum_i \mathrm{ELBO}_{\theta,\phi}(x^i) = \log p_\theta(\mathbf{X}) - \min_\phi \sum_i \mathrm{KL}(q_\phi(z^i \mid x^i) \parallel p_\theta(z^i \mid x^i))$$

  - If ϕ is perfect, it's just the MLE
  - Otherwise, we prefer the kinds of distributions that q_ϕ can successfully reconstruct
- And, to emphasize again, training a VAE isn't *just* minimizing the ELBO
  - Implicit bias of SGD training procedure likely plays a *very* important role
  - Likely even *more* true for complex models, e.g. transformer-based

Outline

1. A Quick Tour of (Image) Generative Models
2. Diffusion Models
3. Some Things We Didn't Cover

Normalizing Flows

bonus!

- Based on change-of-variables formula: if $x = f(z)$ for bijective, differentiable f,

$$p(x) = p(z)\ \left|\det(\nabla_z f^{-1}(z))\right|$$

- Limit layers to be invertible (and square) with easy det; get exact likelihoods
- Some variants: original, Real NVP, MAF, GLOW, FFJORD, Residual Flows

Real Data | Residual Flow

Figure 14: Random samples from 5bit CelebA-HQ 256×256. Most visually appealing batch out of five was chosen.

https://arxiv.org/abs/1906.02735

Autoregressive Models

bonus!

- Use $p(x) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_d \mid x_{1:d-1})$
  - Just a fully-connected DAG model
- Model each $p(x_j \mid x_{1:j-1})$ using some kind of neural net
- Some variants: RNADE, PixelRNN, PixelCNN, WaveNet, MADE
- First models with really good likelihoods and samples for complex datasets
- Slow: go through an image pixel-by-pixel

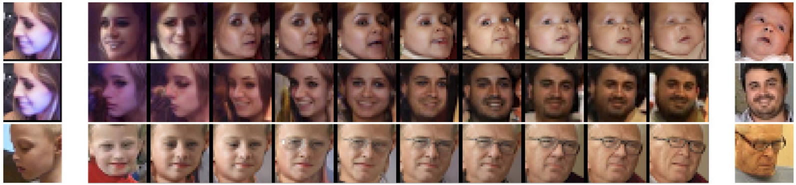


Figure 5: Linear interpolations in the embedding space decoded by the PixelCNN. Embeddings from leftmost and rightmost images are used for endpoints of the interpolation.

https://arxiv.org/abs/1606.05328

- Note: can have interesting behaviour with zero-probability prompts

Energy-Based Models

bonus!

- General term for models like $p_\theta(x) = \frac{1}{Z_\theta} \exp(-\mathcal{E}_\theta(x))$; $\mathcal{E}_\theta$ is "energy"
  - Important example: product of experts $p_1(x) p_2(x)$ has energy $\mathcal{E}_1(x) + \mathcal{E}_2(x)$
- Super-broad category (…essentially any distribution)
- Maximum likelihood: like exponential families, $\nabla_\theta \log \frac{1}{Z_\theta} = \mathbb{E}_{x \sim p_\theta} \nabla_\theta \mathcal{E}_\theta(x)$
  - Can estimate with MCMC sample, e.g. contrastive divergence / Younes algorithm
- Can also fit without estimating Z_θ using score matching, noise-contrastive estimation, Stein discrepancy, adversarial training, …

Score Matching

bonus!

- A way to fit unnormalized generative models
- Hyvärinen score is $s_\theta(x) = \nabla_x \log p_\theta(x) = \nabla_x \log \tilde{p}_\theta(x) - \underbrace{\nabla_x \log Z_\theta}_{0}$
  - Or we can just learn a function s_θ directly
- Score matching tries to match s_θ to target's Hyvärinen score:

$$\operatorname*{arg\,min}_\theta \mathop{\mathbb{E}}_{x \sim p_{\text{target}}} \|s_\theta(x) - \nabla_x \log p_{\text{target}}(x)\|^2$$

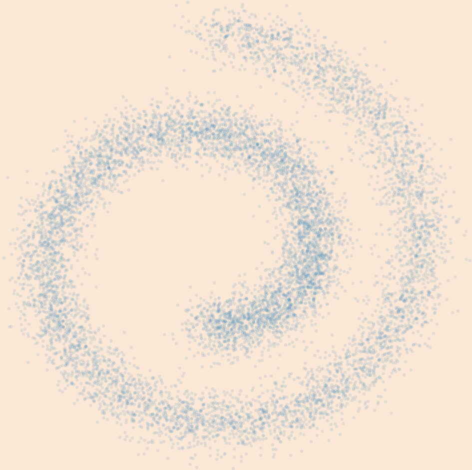
- Under some conditions (using integration by parts), this is equivalent to

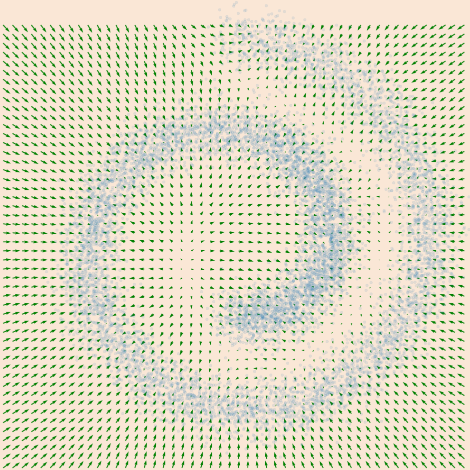
$$\operatorname*{arg\,min}_\theta \mathop{\mathbb{E}}_{x \sim p_{\text{target}}} \frac{1}{2}\|s_\theta(x)\|^2 + \operatorname{Tr}(\nabla_x s_\theta(x))$$

- Denoising score matching, sliced score matching to help with second derivative
- Close connection to contrastive divergence (see PML2 24.3.4)

Score matching a Swiss roll

bonus!





PML2's `score_matching_swiss_roll.ipynb`

Generative Adversarial Networks (GANs)

bonus!

- Generator network $G_\theta(z)$ produces samples based on $p_\theta(z)$
  - Train G_θ to trick a discriminator $D_\phi(x)$ that tries to classify real vs. fake
  - Adversarial game, $\min_\theta \max_\phi$; tricky to optimize
  - Sort of minimizes Jensen-Shannon, $\frac{1}{2}\operatorname{KL}(p_\theta \parallel \frac{p_\theta + p_{\text{target}}}{2}) + \frac{1}{2}\operatorname{KL}(p_{\text{target}} \parallel \frac{p_\theta + p_{\text{target}}}{2})$
    - Variants sort of minimize Wasserstein-1 or other distributional losses
- Not probabilistic – no attempt at computing $\int G_\theta(z) p_\theta(z) \mathrm{d}z$, only sampling

https://arxiv.org/abs/2202.00273

What's the best way to train?

bonus!

- It's not necessarily clear that MLE $= \arg\min_\theta \mathrm{KL}(p_{\text{target}} \,\|\, p_\theta)$ is best
  - MLE has some nice asymptotic properties, given some (strong!) assumptions
    - Classical results assume there is some θ^* where $p_{\text{target}} = p_{\theta^*}$

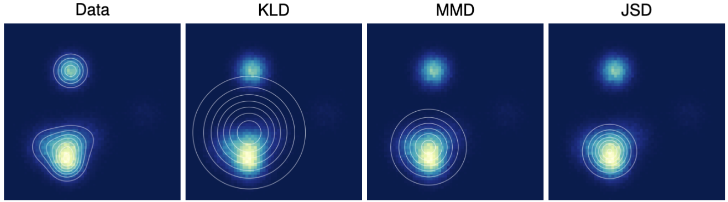


Figure 1: An isotropic Gaussian distribution was fit to data drawn from a mixture of Gaussians by either minimizing Kullback-Leibler divergence (KLD), maximum mean discrepancy (MMD), or Jensen-Shannon divergence (JSD). The different fits demonstrate different tradeoffs made by the three measures of distance between distributions.

https://arxiv.org/abs/1511.01844

- Which one you want depends a lot on what you're using it for

How do we tell if a generative model is any good anyway?

bonus!

- Held-out log-likelihood would be the usual thing to do for generative models
  - GANs can't do; VAEs under-estimate; energy-based models typically over-estimate
    - (Happens by Jensen's inequality; see this paper, section 3.2, to estimate by how much)
  - Images are usually in $\{0, 1, \dots, 255\}^d$: continuous models can get infinite likelihoods
    - Usually de-quantize by adding uniform noise from $[0, 1)^d$
    - Under-estimates log-likelihood of discrete model with $p_{\text{discrete}}(x) = \int_{[0,1)^d} p_\theta(x + u) \mathrm{d}u$ (Jensen's again; see this paper, section 3.1)
- Connection to sample quality is tenuous in high dimensions
  - Break samples, barely change log-likelihood: $p(x) = 0.001 p_\theta(x) + 0.999$ 💩 (x)
    - $\log p(x) \geq \log(0.001 p_\theta) > \underbrace{\log p_\theta(x)}_{\text{scales with } d} - \underbrace{7}_{\text{doesn't}}$
    - On 64×64 ImageNet, PixelCNN beats PixelRNN by 511 nats/img, Conv Draw by 4,514
  - Break log-likelihood, barely change samples: $p = \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}(\tilde{x}^i, \varepsilon^2 I)$ for $\tilde{x}^i \overset{\text{iid}}{\sim} p_\theta$
    - If N is big and ε tiny, unlikely to see duplicates, but it's a way-overfit KDE

How do we tell if a generative model is any good anyway?

bonus!

How do we tell if a generative model is any good anyway?

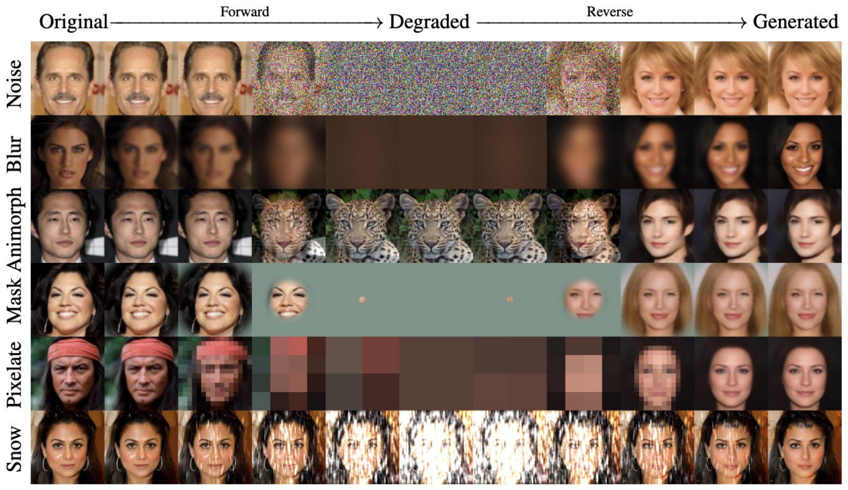
bonus!

- Most common sample evaluation method: Fréchet Inception Distance (FID)
  - Estimate mean, covariance of featurizer pretrained on ImageNet
  - Squared FID: $\|\hat{\mu}_{\text{model}} - \hat{\mu}_{\text{target}}\|^2 + \operatorname{Tr}(\hat{\Sigma}_{\text{model}}) + \operatorname{Tr}(\hat{\Sigma}_{\text{target}}) - 2\operatorname{Tr}\left((\hat{\Sigma}_{\text{model}}\hat{\Sigma}_{\text{target}})^{\frac{1}{2}}\right)$
  - Motivated as Wasserstein-2 (Fréchet) distance between Gaussians
  - Estimator has low variance but high bias (this paper, section 4 / appendix D)
- Precision/Recall, Density/Coverage metrics
  - Try to disambiguate "all samples look reasonable" versus "covering all the data"
- Classification Accuracy Score
  - Train a classifier on (class-conditional) model samples; see how it does on real data
- All of these have issues with "overfitting" by just reproducing training set

Outline

Diffusion Processes

bonus!



https://arxiv.org/abs/2208.09392

- Non-random ("cold diffusion") processes not well understood yet

Diffusion Models as Hierarchical VAEs

bonus!

- Start with data point x_0, add noise to get x_1, add noise to get x_2, ...
- Forward process is ($\approx$)fixed; should choose so $q(x_T \mid x_0) \approx p(x_T)$
- Reverse process $p_\theta(x_{t-1} \mid x_t)$ to remove the noise
- Normal ELBO would give us (see (34) to (45) in this note)

$$\log p_\theta(x_0) \geq \overbrace{\mathop{\mathbb{E}}_{q(x_1 \mid x_0)} \log p_\theta(x_0 \mid x_1)}^{\text{reconstruction}} - \overbrace{\mathop{\mathbb{E}}_{q(x_{T-1} \mid x_0)} \mathrm{KL}(q(x_T \mid x_{T-1}) \parallel p(x_T))}^{\text{prior matching; doesn't depend on } \theta}$$
$$- \sum_{t=1}^{T-1} \underbrace{\mathop{\mathbb{E}}_{q(x_{t-1}, x_{t+1} \mid x_0)} \mathrm{KL}(q(x_t \mid x_{t-1}) \parallel p_\theta(x_t \mid x_{t+1}))}_{\text{consistency}}$$

Diffusion Models as Hierarchical VAEs

bonus!

- Start with data point x_0, add noise to get x_1, add noise to get x_2, ...
- Forward process is $(\approx)$fixed; should choose so $q(x_T \mid x_0) \approx p(x_T)$
- Reverse process $p_\theta(x_{t-1} \mid x_t)$ to remove the noise
- Nicer ELBO (see (46) to (58) in this note) cancels tons of stuff:

$$\log p_\theta(x_0) \geq \overbrace{\mathop{\mathbb{E}}_{q(x_1|x_0)} \log p_\theta(x_0 \mid x_1)}^{\text{reconstruction}} - \overbrace{\mathrm{KL}(q(x_T \mid x_0) \parallel p(x_T))}^{\text{prior matching; no } \theta}$$
$$- \sum_{t=1}^{T-1} \underbrace{\mathop{\mathbb{E}}_{q(x_t|x_0)} \mathrm{KL}(q(x_{t-1} \mid x_t, x_0) \parallel p_\theta(x_{t-1} \mid x_t))}_{p_\theta \text{ should match true denoising process}}$$

- Recovers standard VAE ELBO if $T = 1$

Diffusion Models as Hierarchical VAEs

bonus!

$$\arg\max_\theta \mathop{\mathbb{E}}_{q(x_1|x_0)} \log p_\theta(x_0 \mid x_1) - \mathrm{KL}(q(x_T \mid x_0) \parallel p(x_T)) - \sum_{t=1}^{T-1} \mathop{\mathbb{E}}_{q(x_t|x_0)} \mathrm{KL}(q(x_{t-1} \mid x_t, x_0) \parallel p_\theta(x_{t-1} \mid x_t))$$

- Usual case is fixed normal noise: $q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t I)$
  - Implies $q(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\bar\alpha_t} x_0, (1-\bar\alpha_t) I\right)$ for $\bar\alpha_t = \prod_{\tau=1}^{t} (1-\beta_\tau)$
  - Choose T, β_t such that $\bar\alpha_T \approx 0$, so $q(x_T \mid x_0) \approx \mathcal{N}(0, I)$
  - Get that $q(x_{t-1} \mid x_t, x_0) = \mathcal{N}\left(x_{t-1}; \gamma_t x_t + \delta_t x_0, \sigma_t^2 I\right)$; $\gamma_t, \delta_t, \sigma_t$ depend only on β_ts
  - We can just choose $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \gamma_t x_t + \delta_t \hat{x}_\theta(x_t, t), \sigma_t^2 I)$!
  - KL, reconstruction terms simplify a lot: get

$$\arg\min_\theta \mathop{\mathbb{E}}_{\substack{x_0 \sim p_{\text{target}} \\ t \sim \mathrm{Unif}\{1,\dots,T\}}} \left[\mathop{\mathbb{E}}_{x_t \sim \mathcal{N}\left(\sqrt{\bar\alpha_t} x_0, (1-\bar\alpha_t) I\right)} \left[\frac{\delta_t^2}{2\sigma_t^2} \begin{cases} \|\hat{x}_\theta(x_1, 1) - x_0 - \gamma_1 x_1\|^2 & \text{if } t = 1 \\ \|\hat{x}_\theta(x_t, t) - x_0\|^2 & \text{otherwise} \end{cases} \right] \right]$$

  - Empirically can choose to ignore weighting δ_t^2 / σ_t^2 and the $t = 1$ special case:

$$\arg\min_\theta \mathop{\mathbb{E}}_{\substack{x_0 \sim p_{\text{target}} \\ t \sim \mathrm{Unif}\{1,\dots,T\}}} \left[\mathop{\mathbb{E}}_{x_t \sim \mathcal{N}\left(\sqrt{\bar\alpha_t} x_0, (1-\bar\alpha_t) I\right)} \left[\|\hat{x}_\theta(x_t, t) - x_0\|^2 \right] \right]$$

Other views of Diffusion Models

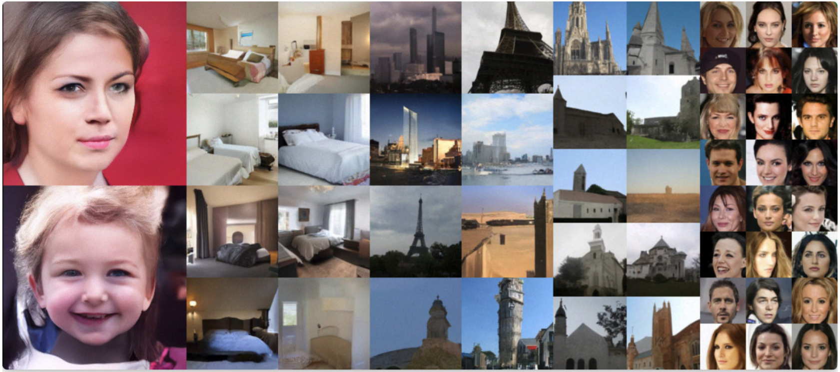
bonus!

- Can view essentially same objective as denoising score matching.
- Or as stacked denoising auto-encoders.

- Helpful descriptions by: Yang Song, Lilian Weng, Calvin Luo, and PML2 25

“Plain” Diffusion Samples

bonus!



Samples from the NCSNv2 [18] model. From left to right: FFHQ 256x256, LSUN bedroom 128x128, LSUN tower 128x128, LSUN church_outdoor 96x96, and CelebA 64x64.

https://yang-song.net/blog/2021/score/

Infinitely many noise levels

bonus!

- Can take the $T = \infty$ limit based on stochastic differential equations
  - See Yang Song's blog post
- Gives exact log-likelihoods and better ability to condition

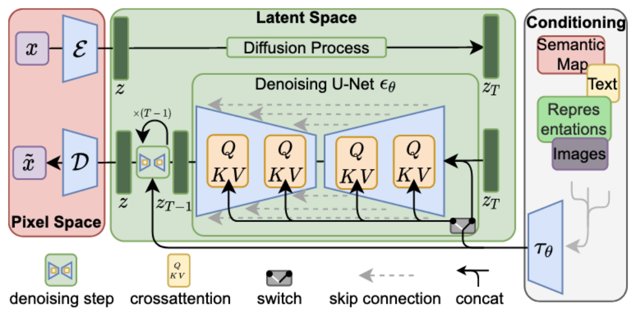
Image inpainting with a time-dependent score-based model trained on LSUN bedroom. The leftmost column is ground-truth. The second column shows masked images (y in our framework). The rest columns show different inpainted images, generated by solving the conditional reverse-time SDE.

https://yang-song.net/blog/2021/score/

Stable Diffusion

bonus!

- Train a fancy, high-quality auto-encoder
- Run diffusion model on the code distribution
- Condition the decoder on text embeddings



https://arxiv.org/abs/2112.10752

ControlNet

bonus!

- Allows "post-processing" to add new kinds of conditioning to pretrained model

https://www.reddit.com/r/StableDiffusion/comments/1281iva/new_controlnet_face_model/

bonus!

ARTIFICIAL INTELLIGENCE / TECH / LAW

Getty Images is suing the creators of AI art tool Stable Diffusion for scraping its content

An image created by Stable Diffusion showing a recreation of Getty Images' watermark. Image: The Verge / Stable Diffusion

/ Getty Images claims Stability AI 'unlawfully' scraped millions of images from its site. It's a significant escalation in the developing legal battles between generative AI firms and content creators.

By JAMES VINCENT

Jan 17, 2023, 2:30 AM PST | 18 Comments / 18 New

bonus!

Training Set | **Generated Image**

Caption: Living in the light with Ann Graham Lotz

Prompt: Ann Graham Lotz

Figure 1: Diffusion models memorize individual training examples and generate them at test time. **Left:** an image from Stable Diffusion's training set (licensed CC BY-SA 3.0, see [49]). **Right:** a Stable Diffusion generation when prompted with "Ann Graham Lotz". The reconstruction is nearly identical (ℓ_2 distance = 0.031).

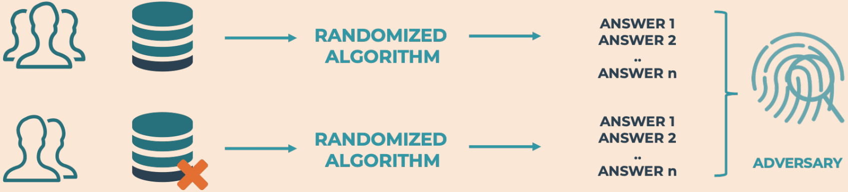
https://arxiv.org/abs/2301.13188

Outline

Privacy

bonus!

- How can we prevent models from memorizing individual data points?
- Leading framework is differential privacy



https://2021.ai/machine-learning-differential-privacy-overview/

- CPSC grad courses: 532P by Mijung Park, sometimes 538L by Mathias Lecuyer

Fairness, Accountability, Transparency

bonus!

- Tons of issues around ML models / applications
- Some have technical (partial) solutions
- Some can only be handled socially
- "Sociotechnical systems" (STS)

- FAccT and AIES conferences
- New undergrad course coming in DSCI, focusing mostly on fairness

Causality

bonus!

- 532Y: Causal ML by Mathias Lecuyer
- Math 605D by Elina Robeva (sometimes)

- Closely related to fairness
- More related things to be aware of:
  - Disentanglement
  - Independent components analysis
  - Out-of-distribution generalization, domain adaptation

More Deep Learning: NLP

bonus!

- Big, super-fast thing is large language models
  - GPT4 since we last talked about them. . .
  - We May be Surprised Again: Why I take LLMs seriously

- CPSC 436N: NLP (likely W1)
- CPSC 532V: Commonsense Reasoning in NLP by Vered Shwartz (planned W2)
- 532G (dialogue models) by Giuseppe Carenini
- courses by Muhammad Abdul-Mageed

- 532S: Multimodal Learning with Vision, Language and Sound

More Deep Learning: Vision/Graphics

bonus!

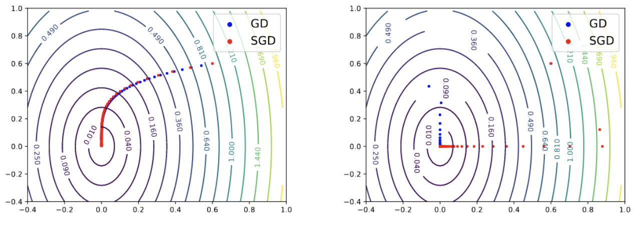
- Lots of vision to do beyond what was in this course!

- CPSC 425: Computer Vision
- 533Y: Visual Geometry with Deep Learning by Kwang Moo Yi (planned W1)
- 533R: Visual AI by Helge Rhodin (planned W2)
- 533V: Learning to Move by Michiel van de Panne (planned W2)

Theory

bonus!

- Why/when do ML models / optimizers work, mathematically?



(a) Small learning rate regime (b) Moderate learning rate regime

https://arxiv.org/abs/2011.02538

- 532D: Modern Statistical Learning Theory by me (planned W1)
- 406 and 536M by Michael Friedlander (planned W1)
- 5XX by Mark Schmidt (semi-ongoing plus maybe W2)
- EECE 571Z Convex Optimization by Christos Thrampoulidis
- Various stat courses

Probabilistic/Bayesian/. . . ML

bonus!

- Probabilistic programming: 532W by Frank Wood
- Stat 520A: Bayesian analysis by Alexandre Bouchard-Côté
- Stat 520B: Variational Bayes by Trevor Campbell
- Stat 547S: Topics on Symmetry by Benjamin Bloem-Reddy
- ECE 571F: Deep Learning with Structures by Renjie Liao
- Various more stat courses

- Some more things to be aware of:
  - Mutual information/dependence estimation
  - Graph neural networks, deep sets, other structured data
  - Particle filters
  - Bayesian neural networks

Reinforcement learning

bonus!

- 322, 422 – logic, more graphical models, search, planning, some RL
- 522 by David Poole (PGMs, some RL)

- 532J: Never Ending Reinforcement Learning by Jeff Clune
- 533V: Learning to Move by Michiel van de Panne (planned W2)

- Some more things to be aware of:
  - Meta-learning
  - Online learning
  - Active learning
  - Multi-armed bandits
  - Auto-ML

Other stuff

bonus!

- 532C: Human-Centred AI by Cristina Conati (planned W2)
- Somewhat relevant: 539L: Automated Testing by Caroline Lemieux
- 532L: Modes of Strategic Behaviour by Kevin Leyton-Brown
- Math 605D: Tensor decompositions by Elina Robeva (sometimes)
- Math 555: Compressed Sensing by Yaniv Plan

- Possible courses by
  - Shengjia Zhao (new in CS; information theory / econ / LLMs)
  - Geoff Pleiss (new in Stat; Gaussian processes)
  - Xiaxio Li (ECE; federated learning)
  - Lele Wang (ECE; coding theory)

- Reading groups: `https://ml.ubc.ca/reading-groups/`
- Talks: CAIDA (AI broadly), MILD (“mathematical” ML)

bonus!

Midjourney Bot ✓ BOT Today at 12:57 PM

high-res photo of a computer science professor thanking her students for the term, and wishing them luck on their upcoming finals and projects. the professor is an early-30s white trans woman, full-figured, with wavy brown hair, and wearing black, facing towards the students. the students are sitting in an auditorium-style lecture hall, filling about 1/4 of the seats. 4k HD photo, dramatic lighting, happy vibes, high-resolution, sharp details --v 5 - Image #1 @danica

3D CGI render of a young redheaded male professor thanking the class on the last day

Report issue

3D CGI render of a korean female professor thanking the class on the last day

Report issue