CS294-158 Deep Unsupervised Learning

Lecture 8 Round-up of Strengths and Weaknesses

Pieter Abbeel, Xi (Peter) Chen, Jonathan Ho, Aravind Srinivas, Alex Li, Wilson Yan

UC Berkeley

“The brain has about 10^14 synapses and we only live for about 10^9 seconds. So we have a lot more parameters than data. This motivates the idea that we must do a lot of unsupervised learning since the perceptual input (including proprioception) is the only place we can get 10^5 dimensions of constraint per second.”

- Geoffrey Hinton

(in his 2014 AMA on Reddit)

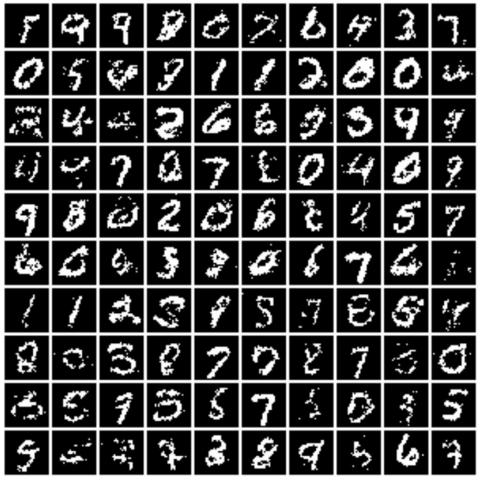
Summary of Course So Far

- Autoregressive models
  - PixelRNN, PixelCNN, PixelCNN++, PixelSNAIL
- Flow models
  - NICE, RealNVP, Autoregressive Flows, Inverse Autoregressive Flows, Glow, Flow++
- Latent Variable models
  - Approximate likelihood with Variational Lower Bound
  - Wake Sleep
  - Variational Auto-Encoder, IWAE, IAF-VAE, PixelVAE (VLAE), VQ-VAE
- Implicit models
  - Generative Adversarial Networks (GAN)
  - Other principles like moment matching, energy based models, etc
- Self-supervised Learning / Representation Learning
  - Learn meaningful features of raw sensory observations useful for downstream tasks
  - Data + Compute + Core Cognitive Principles (common sense tasks, invariance priors)

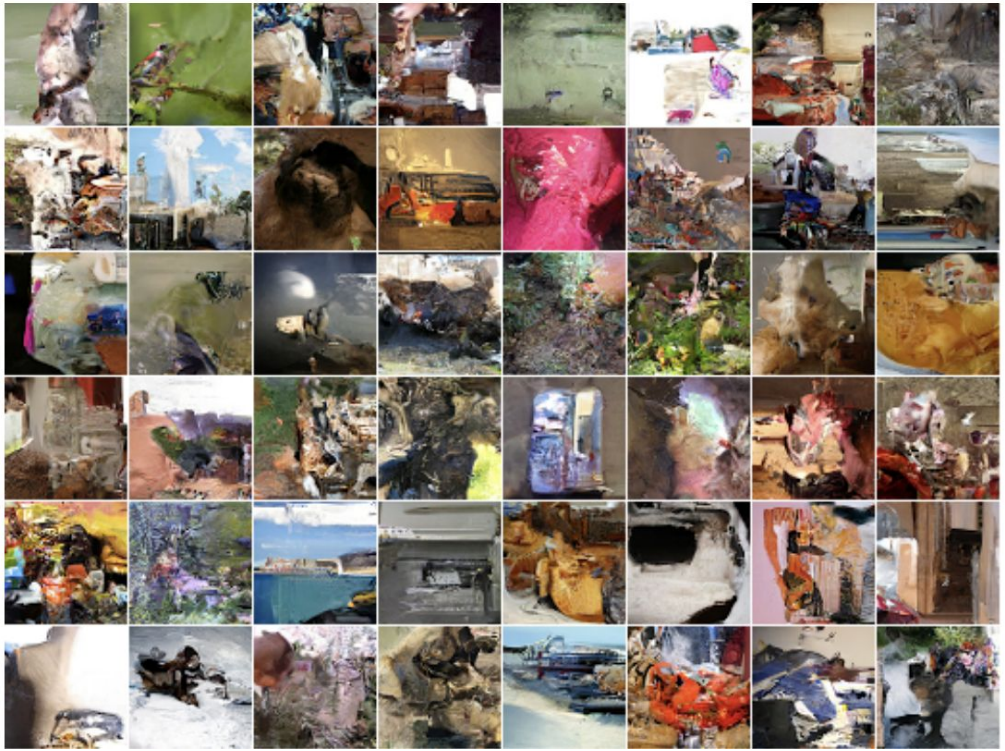
Summary of Course So Far

- Autoregressive models
  - PixelRNN, PixelCNN, PixelCNN++, PixelSNAIL
- Flow models
  - NICE, RealNVP, Autoregressive Flows, Inverse Autoregressive Flows, Glow, Flow++
- Latent Variable models
  - Approximate likelihood with Variational Lower Bound
  - Wake Sleep
  - Variational Auto-Encoder, IWAE, IAF-VAE, PixelVAE (VLAE)
- Implicit models
  - Generative Adversarial Networks
  - Other principles like moment matching, mapping noise to data, etc
- Self-supervised Learning / Representation Learning
  - Learn meaningful features of raw sensory observations useful for downstream tasks
  - Data + Compute + Core Cognitive Principles (common sense tasks)

Autoregressive Models

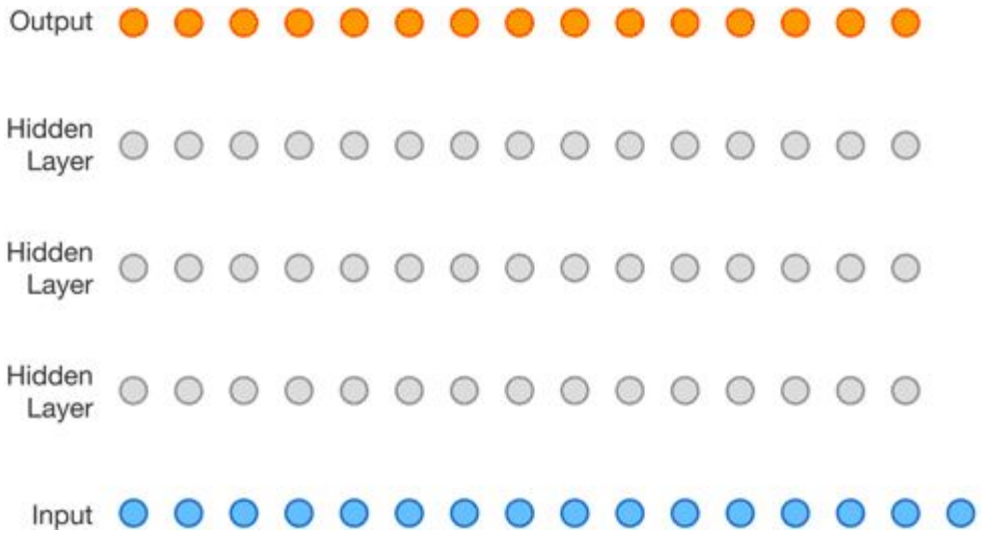


MADE - 2015



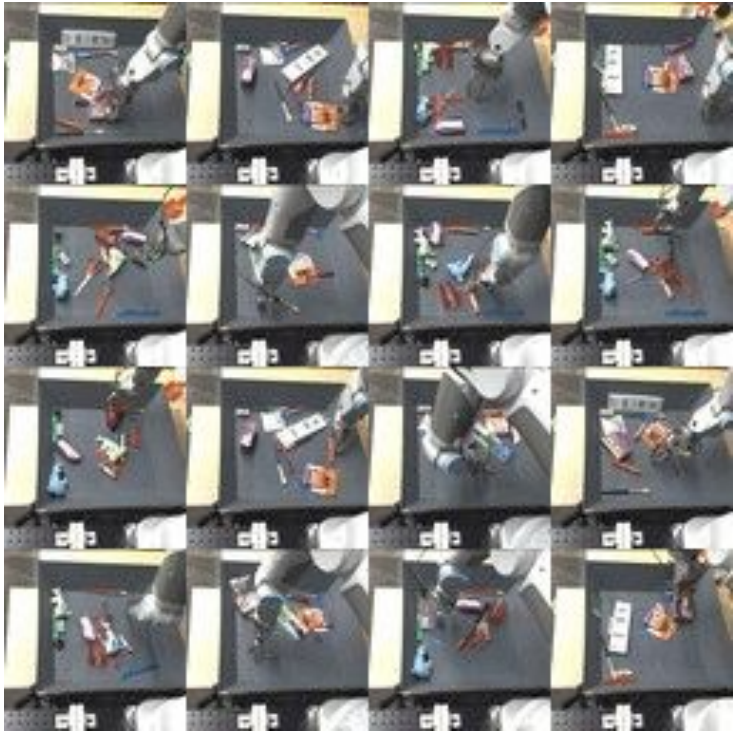
PixelRNN/CNN 2016

Autoregressive Models



WaveNet

Autoregressive Models



Video Pixel Networks

Autoregressive Models

Subscale Pixel Networks

Hierarchical Autoregressive Image Models with Auxiliary Decoders

Autoregressive Models - OpenAI GPT-2

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

Autoregressive Models - History of language models

SLP book, 2000 (Shannon, 1951), 3-gram

They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Sutskever et al, 2011, RNNs

The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger

Jozefowicz et al, 2016, BIG LSTMs

With even more new technologies coming onto the market quickly during the past three years , an increasing number of companies now must tackle the ever-changing and ever-changing environmental challenges online .

Liu et al, 2018, Transformer

==wings over kansas

==wings over kansas is a 2010 dhamma feature film written and directed by brian ig ariyoshi . it premiered on march 17, 2010 the film tells the story of three americans who bravely achieved a victory without expected daknfi .

==Wings Over Kansas Plot

the story begins with the faltering success of egypt 's hungry dakfunctionality when he loses his lives around the time when the embarked [...]

Radford et al, 2019, BIG Transformer

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Perez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Perez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Perez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Perez.

Perez and his friends were astonished to see the unicorn herd. [...]

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Compute power
- Several days / weeks of training
- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models - Future

- Still only scratching the surface of what's possible.
  - Advances with model-parallelism to come
  - Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
  - Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
  - Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
  - Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
  - New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models - Future

- Still only scratching the surface of what's possible.
  - Advances with model-parallelism to come
  - Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
  - Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
  - Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
  - Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
  - New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models - Future

- Still only scratching the surface of what's possible.
  - Advances with model-parallelism to come
  - Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
  - Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
  - Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
  - Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
  - New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models - Future

- Still only scratching the surface of what's possible.
  - Advances with model-parallelism to come
  - Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
  - Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
  - Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
  - Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
  - New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models - Future

- Still only scratching the surface of what's possible.
  - Advances with model-parallelism to come
  - Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
  - Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
  - Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
  - Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
  - New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models - Future

- Still only scratching the surface of what's possible.
  - Advances with model-parallelism to come
  - Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
  - Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
  - Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
  - Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
  - New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models - Future

- Still only scratching the surface of what's possible.
  - Advances with model-parallelism to come
  - Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
  - Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
  - Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
  - Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
  - New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models - Future

- Still only scratching the surface of what's possible.
  - Advances with model-parallelism to come
  - Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
  - Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
  - Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
  - Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
  - New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models - Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models - Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models - Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models - Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models - Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models - Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models - Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

Autoregressive Models - Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

Autoregressive Models - Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

Autoregressive Models - Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

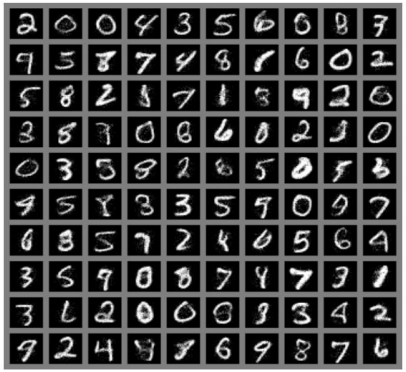
Autoregressive Models - Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

Summary of Course So Far

- Autoregressive models
  - PixelRNN, PixelCNN, PixelCNN++, PixelSNAIL
- Flow models
  - NICE, RealNVP, Autoregressive Flows, Inverse Autoregressive Flows, Glow, Flow++
- Latent Variable models
  - Approximate likelihood with Variational Lower Bound
  - Wake Sleep
  - Variational Auto-Encoder, IWAE, IAF-VAE, PixelVAE (VLAE)
- Implicit models
  - Generative Adversarial Networks
  - Other principles like moment matching, mapping noise to data, etc
- Self-supervised Learning / Representation Learning
  - Learn meaningful features of raw sensory observations useful for downstream tasks
  - Data + Compute + Core Cognitive Principles (common sense tasks)

Flow Models



(a) Model trained on MNIST

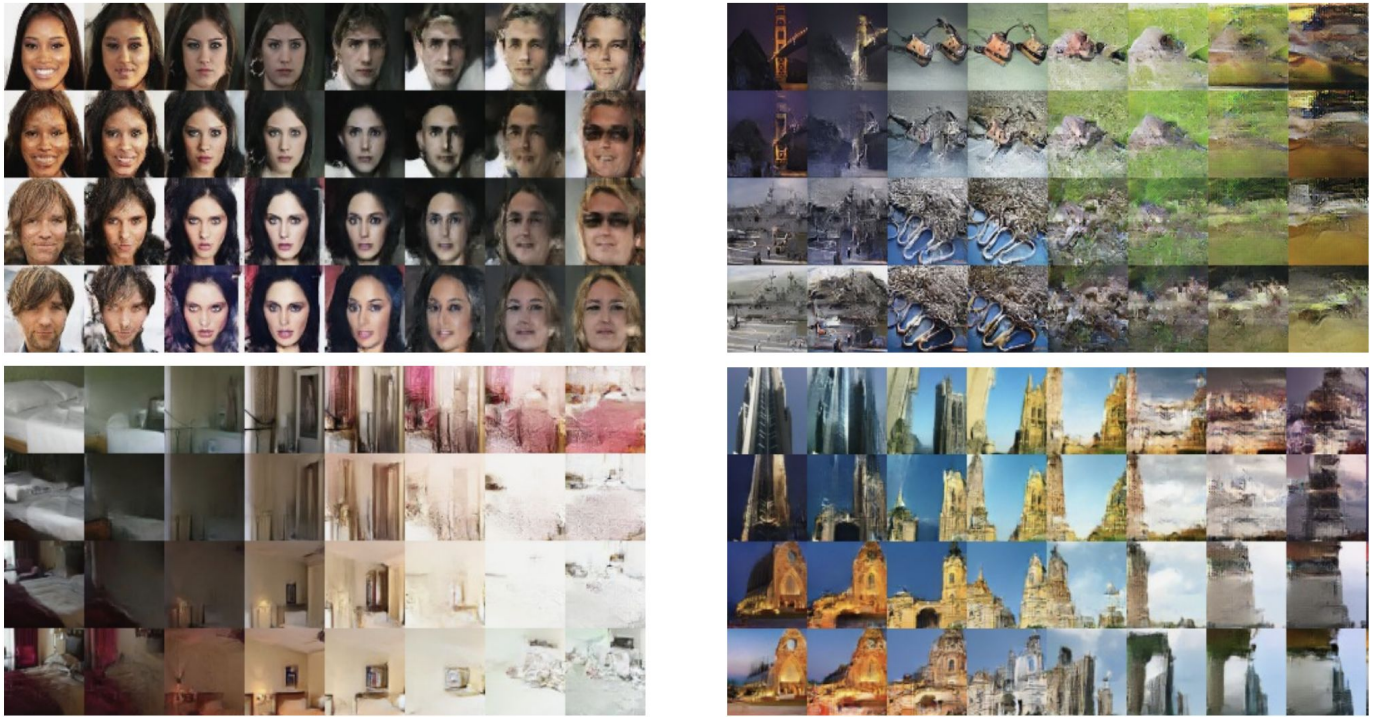
(b) Model trained on TFD

(c) Model trained on SVHN

(d) Model trained on CIFAR-10

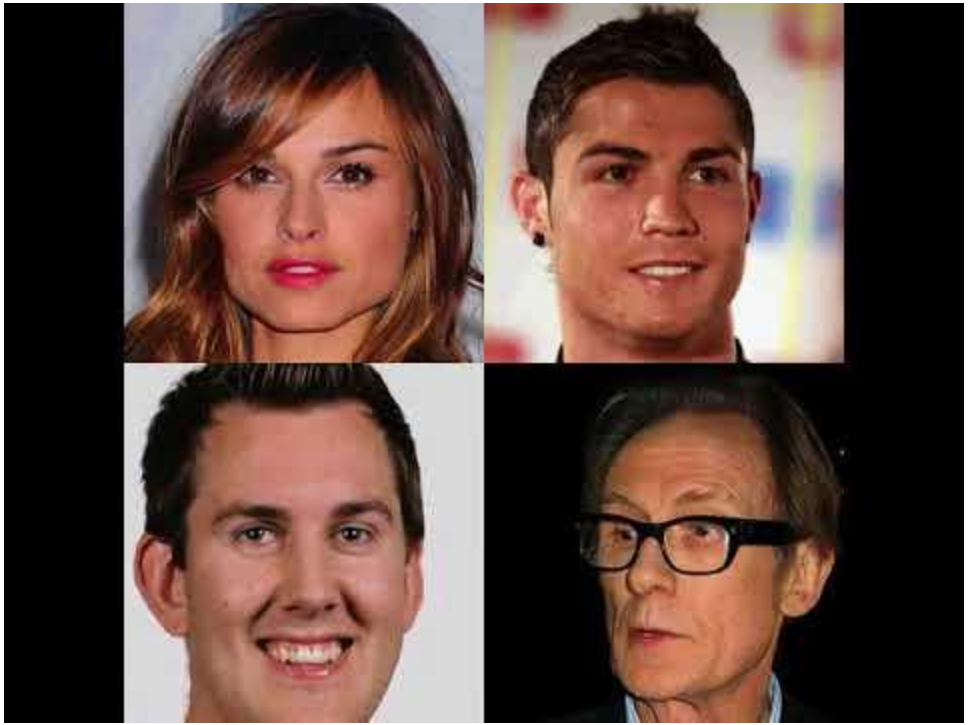
NICE (Dinh et al 2014)

Flow Models



RealNVP (Dinh et al 2016)

Glow - Big progress on sample quality



OpenAI Glow

Flow++- Progress on bits/dim on high entropy datasets

Model family	Model	CIFAR10 bits/dim	ImageNet 32x32 bits/dim	ImageNet 64x64 bits/dim
Non-autoregressive	RealNVP (Dinh et al., 2016)	3.49	4.28	–
	Glow (Kingma & Dhariwal, 2018)	3.35	4.09	3.81
	IAF-VAE (Kingma et al., 2016)	3.11	–	–
	Flow++ (ours)	**3.09**	**3.86**	**3.69**
Autoregressive	Multiscale PixelCNN (Reed et al., 2017)	–	3.95	3.70
	PixelCNN (van den Oord et al., 2016b)	3.14	–	–
	PixelRNN (van den Oord et al., 2016b)	3.00	3.86	3.63
	Gated PixelCNN (van den Oord et al., 2016c)	3.03	3.83	3.57
	PixelCNN++ (Salimans et al., 2017)	2.92	–	–
	Image Transformer (Parmar et al., 2018)	2.90	3.77	–
	PixelSNAIL (Chen et al., 2017)	2.85	3.80	3.52

Flow++

Flow Models - Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models - Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models - Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models - Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models - Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models - Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models - Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models - Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 megapixel (1024 x 1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models - Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 megapixel (1024 x 1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models - Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 megapixel (1024 x 1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models - Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 megapixel (1024 x 1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models - Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 megapixel (1024 x 1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models - Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 megapixel (1024 x 1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

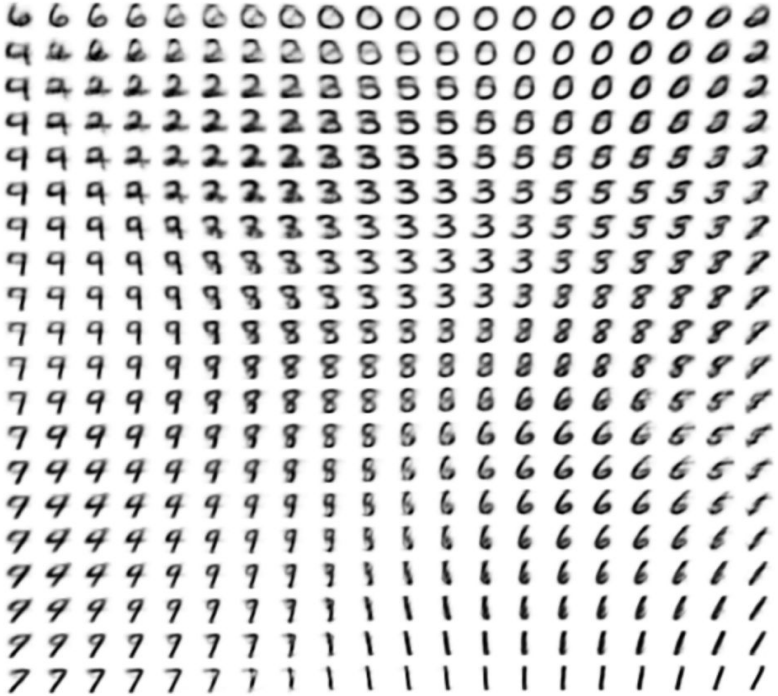
Flow Models - Negatives

- z is as big as x. Models end up becoming *big*.
- As of now, no notion of *lower* dimensional embedding.
- Careful initialization (not really a negative)

Summary of Course So Far

- Autoregressive models
  - PixelRNN, PixelCNN, PixelCNN++, PixelSNAIL
- Flow models
  - NICE, RealNVP, Autoregressive Flows, Inverse Autoregressive Flows, Glow, Flow++
- Latent Variable models
  - Approximate likelihood with Variational Lower Bound
  - Wake Sleep
  - Variational Auto-Encoder, IWAE, IAF-VAE, PixelVAE (VLAE)
- Implicit models
  - Generative Adversarial Networks
  - Other principles like moment matching, mapping noise to data, etc
- Self-supervised Learning / Representation Learning
  - Learn meaningful features of raw sensory observations useful for downstream tasks
  - Data + Compute + Core Cognitive Principles (common sense tasks)

Latent Variable Models

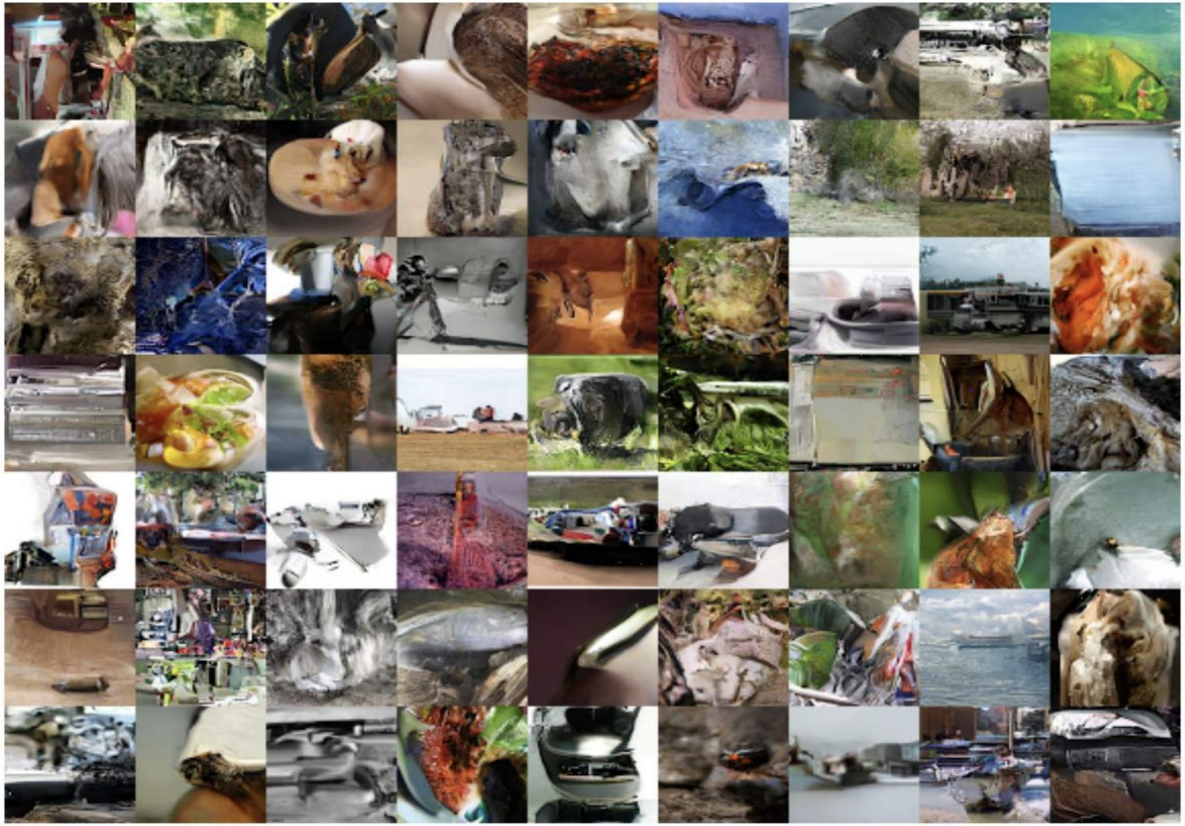


Auto-Encoding Variational Bayes (Kingma 2013)

Latent Variable Models - PixelVAE (Gulrajani)

PixelVAE
Gularajani et al 2016

Latent Variable Models - PixelVAE (Gulrajani)



PixelVAE
Gularajani et al 2016

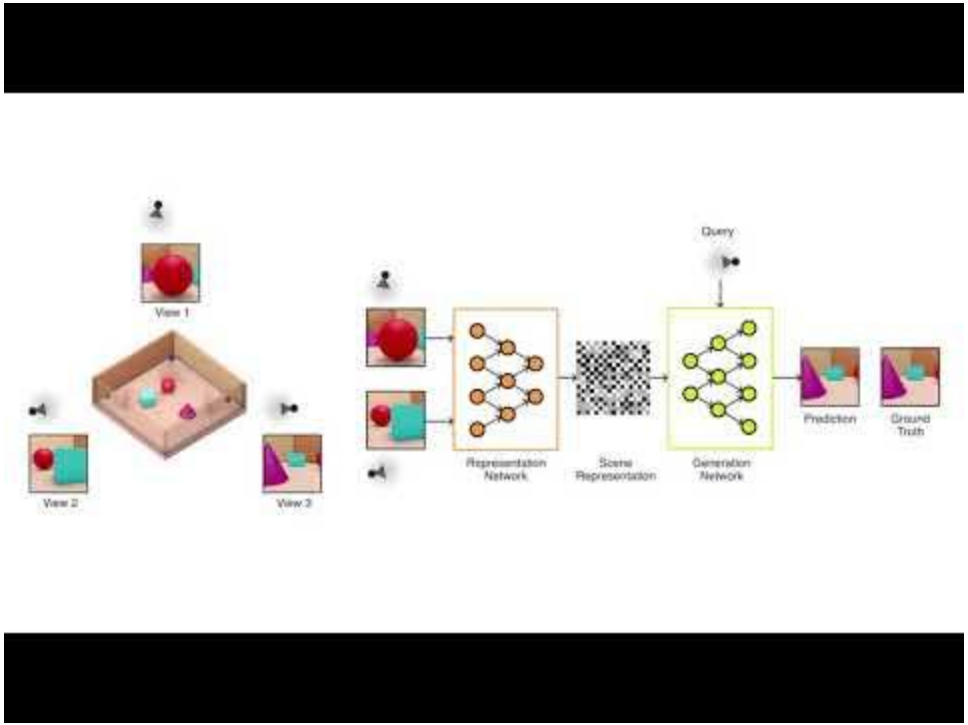
Latent Variable Models - BIVA (Maaloe et al)

BIVA (Maaloe et al 2019)

Well known VAE Applications

- Sketch-RNN
- World Models
- Visual concepts for RL (beta-VAE)

Well known VAE Applications - Generative Query Networks



Generative Query Networks

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

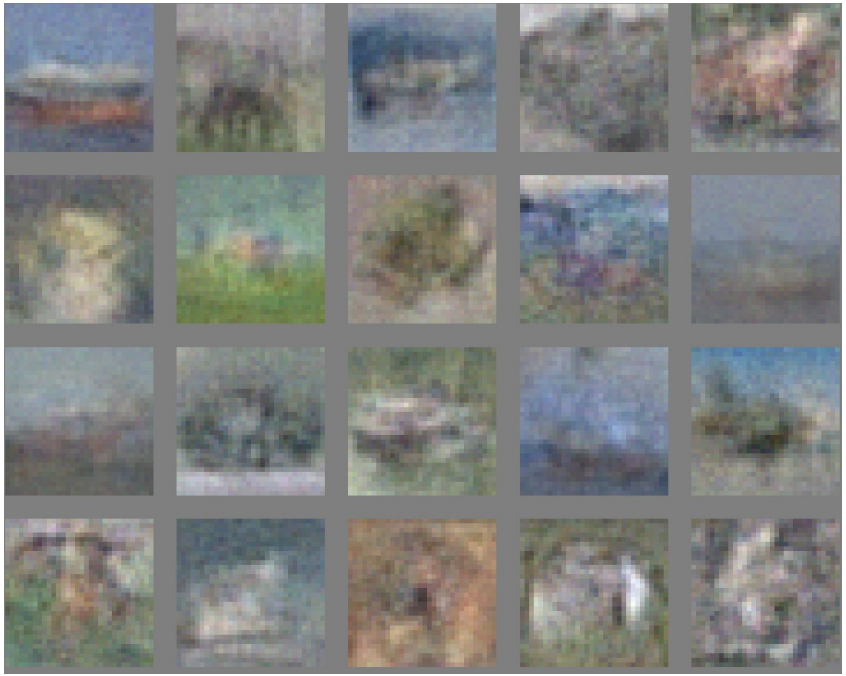
VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

Summary of Course So Far

- Autoregressive models
  - PixelRNN, PixelCNN, PixelCNN++, PixelSNAIL
- Flow models
  - NICE, RealNVP, Autoregressive Flows, Inverse Autoregressive Flows, Glow, Flow++
- Latent Variable models
  - Approximate likelihood with Variational Lower Bound
  - Wake Sleep
  - Variational Auto-Encoder, IWAE, IAF-VAE, PixelVAE (VLAE)
- Implicit models
  - Generative Adversarial Networks
  - Other principles like moment matching, mapping noise to data, etc
- Self-supervised Learning / Representation Learning
  - Learn meaningful features of raw sensory observations useful for downstream tasks
  - Data + Compute + Core Cognitive Principles (common sense tasks)

Generative Adversarial Networks

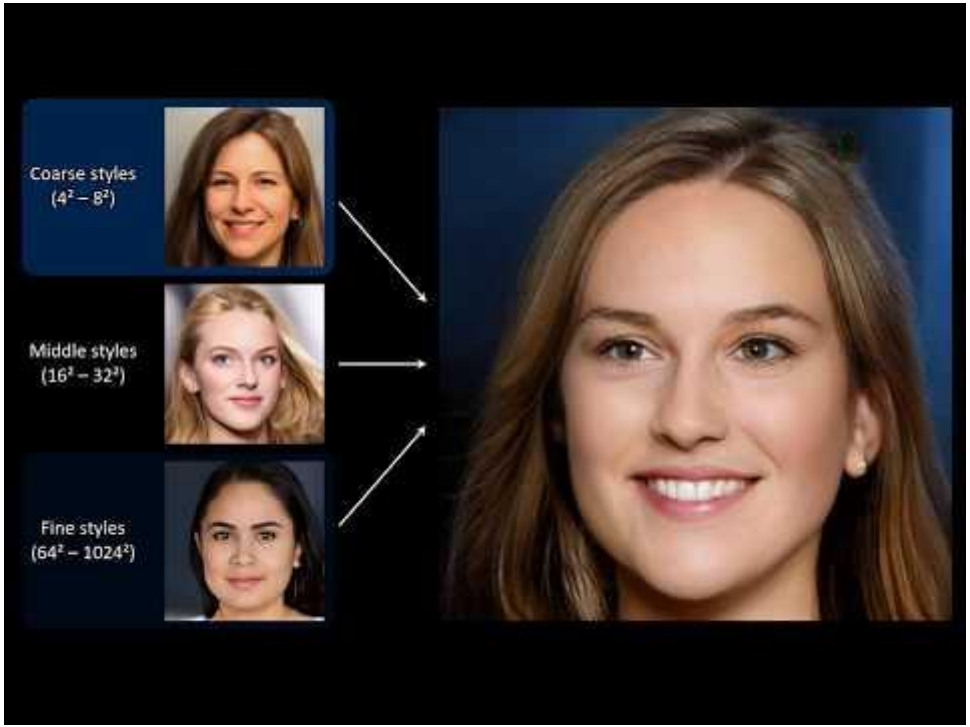


Original GAN (2014) - Goodfellow et al

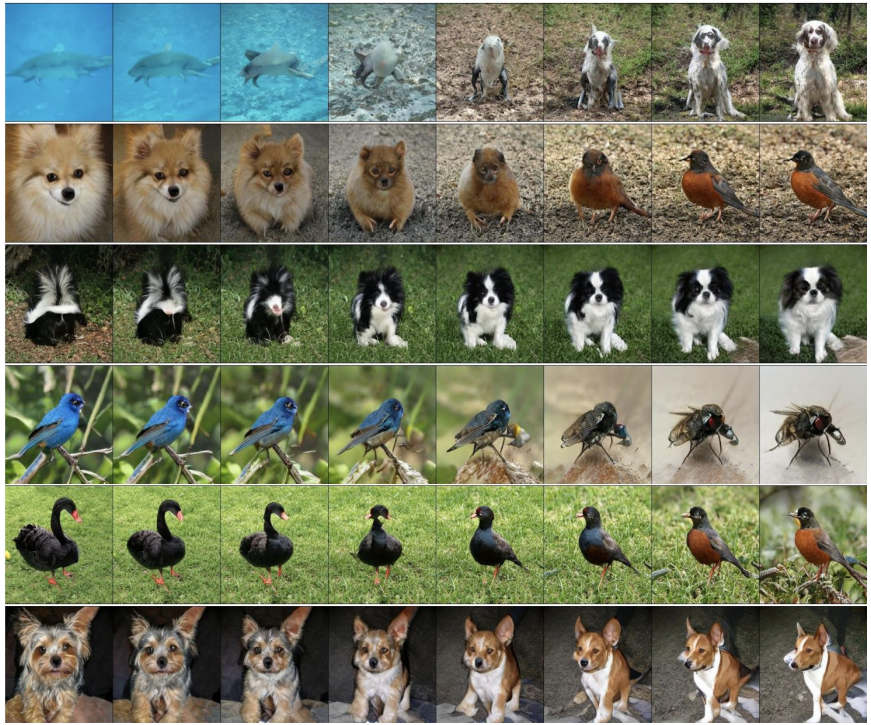
Generative Adversarial Networks

DCGAN - Radford, Metz, Chintala 2015

Generative Adversarial Networks



StyleGAN



BigGAN

Generative Adversarial Networks - Future

- Hard to predict *against* them given an array of the most powerful generation results for images.
- Progress in unconditional GANs.
- Handling more fine-grained details
- More complex scenes (multiple people with objects)
- Video generation

Generative Adversarial Networks - Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks - Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks - Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks - Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks - Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks - Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks - Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks - Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks - Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks - Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks - Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks - Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks - Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks - Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn’t a negative specific to GANs

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (Ex: FiLM conditioning, gating, LayerNorm, ActNorm, ...).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: *Good* 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (Ex: FiLM conditioning, gating, LayerNorm, ActNorm, ...).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: *Good* 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (Ex: FiLM conditioning, gating, LayerNorm, ActNorm, ...).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: *Good* 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (Ex: FiLM conditioning, gating, LayerNorm, ActNorm, ...).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: *Good* 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (Ex: FiLM conditioning, gating, LayerNorm, ActNorm, ...).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: *Good* 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (Ex: FiLM conditioning, gating, LayerNorm, ActNorm, ...).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: *Good* 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

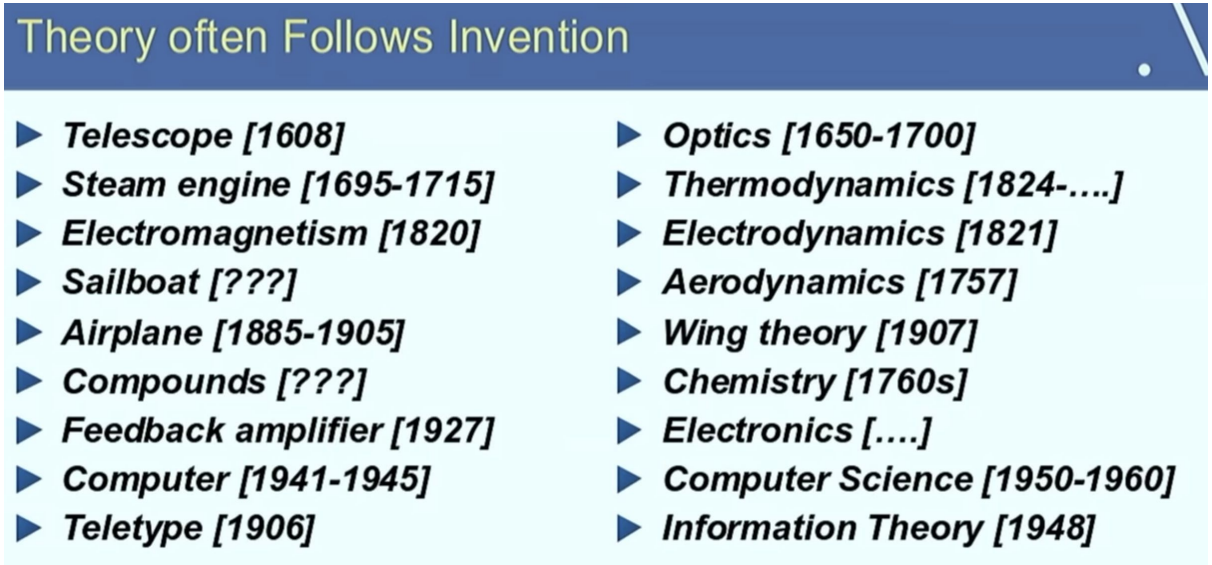
- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (Ex: FiLM conditioning, gating, LayerNorm, ActNorm, ...).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: *Good* 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

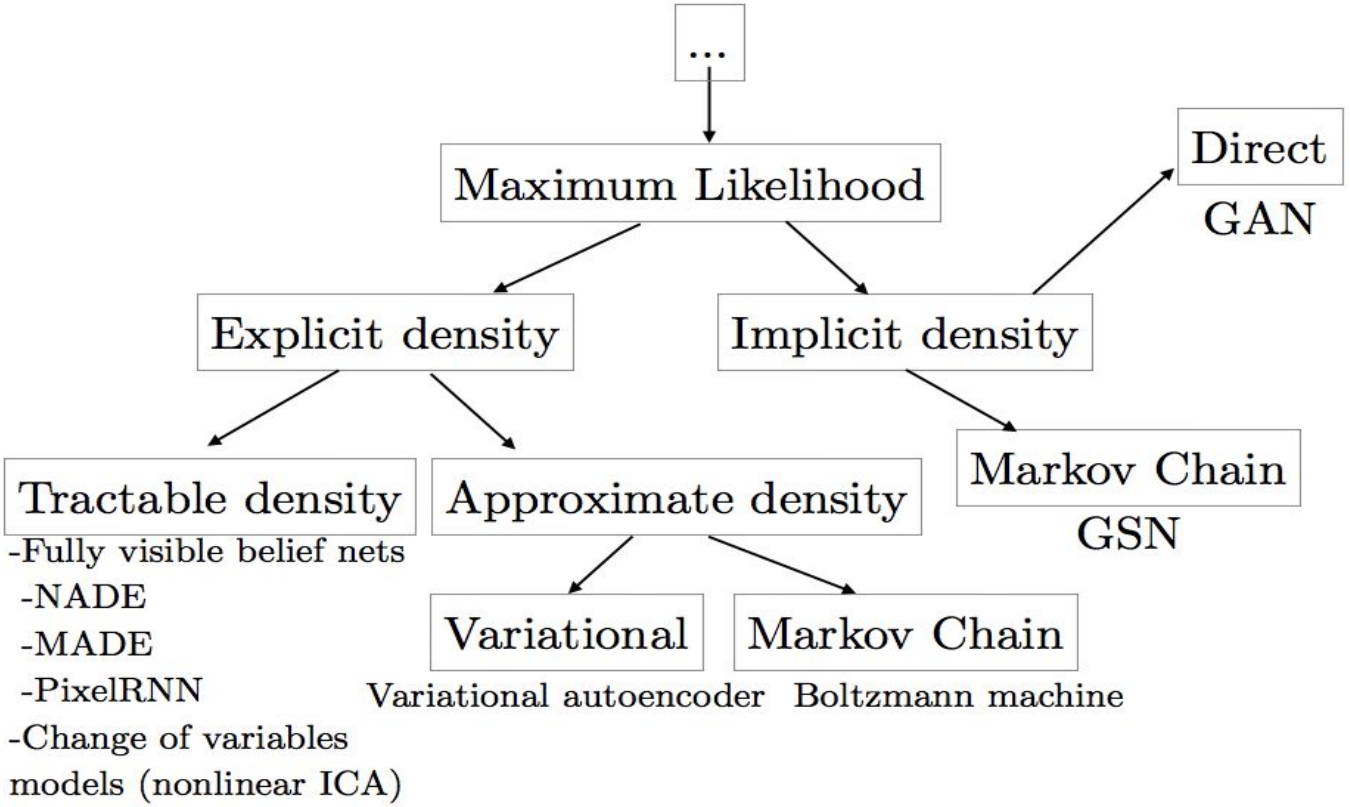
- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (Ex: FiLM conditioning, gating, LayerNorm, ActNorm, ...).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: *Good* 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Bright side : Upto to aesthetics / taste in terms of betting on one (density models vs GAN)
- Many technological advances in the past have been possible without rigorous science built for them (science followed later) [Le Cun: Epistemology of DL]



Taxonomy


...
Maximum Likelihood
Direct
GAN
Explicit density
Implicit density
Tractable density
Approximate density
Markov Chain
GSN
-Fully visible belief nets
-NADE
-MADE
-PixelRNN
-Change of variables
models (nonlinear ICA)
Variational
Markov Chain
Variational autoencoder
Boltzmann machine

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image -> Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don’t have lot of compute, GAN is the best choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image -> Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don’t have lot of compute, GAN is the best choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image -> Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don’t have lot of compute, GAN is the best choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image -> Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don't have lot of compute, GAN is the best choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image -> Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don't have lot of compute, GAN is the best choice.

When GANs?

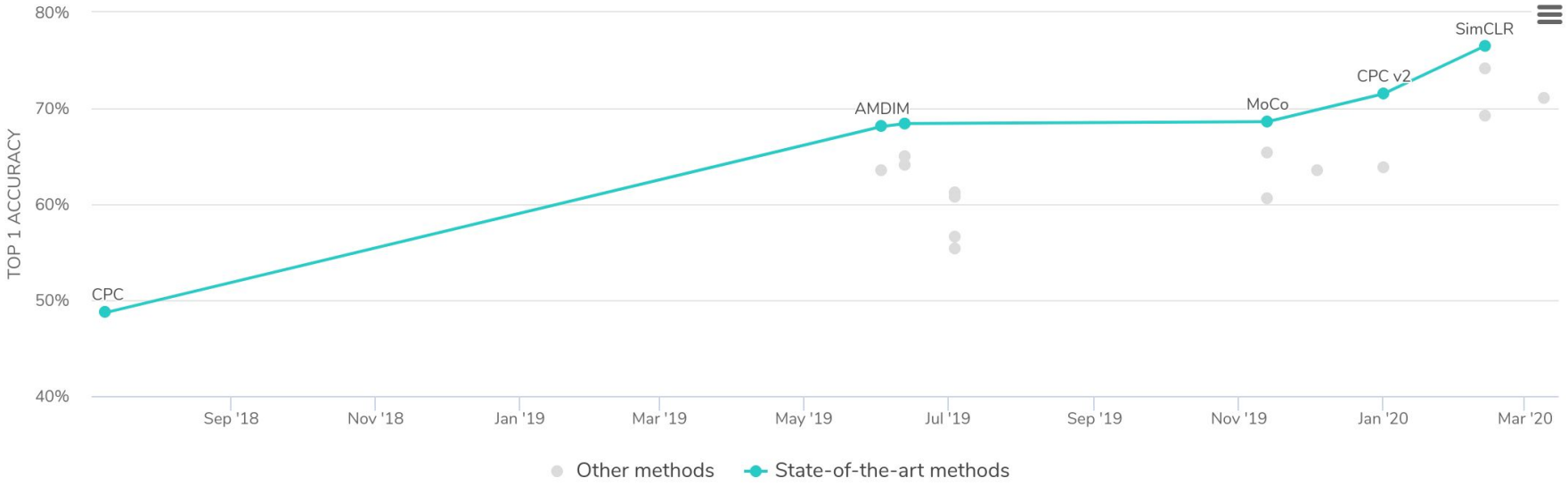
- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image -> Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don’t have lot of compute, GAN is the best choice.

Summary of Course So Far

- Autoregressive models
  - PixelRNN, PixelCNN, PixelCNN++, PixelSNAIL
- Flow models
  - NICE, RealNVP, Autoregressive Flows, Inverse Autoregressive Flows, Glow, Flow++
- Latent Variable models
  - Approximate likelihood with Variational Lower Bound
  - Wake Sleep
  - Variational Auto-Encoder, IWAE, IAF-VAE, PixelVAE (VLAE)
- Implicit models
  - Generative Adversarial Networks
  - Other principles like moment matching, mapping noise to data, etc
- Self-supervised Learning / Representation Learning
  - Learn meaningful features of raw sensory observations useful for downstream tasks
  - Data + Compute + Core Cognitive Principles (common sense tasks)

Self-Supervision on Images: Progress

Self-Supervised Image Classification on ImageNet



Summary of contrastive learning

- Contrastive Learning: Dictionary look up task.
- Predictive Coding (or) Instance Discrimination
- Mechanism for dictionary lookup: (1) end-to-end, (2) momentum
- Predictive Coding Success: CPCv2
- Instance Discrimination Success:
  - MoCo - Momentum Contrast
  - SimCLR - End-to-End Instance Contrast

Summary of contrastive learning

- Contrastive Learning: Dictionary look up task.
- Predictive Coding (or) Instance Discrimination
- Mechanism for dictionary lookup: (1) end-to-end, (2) momentum
- Predictive Coding Success: CPCv2
- Instance Discrimination Success:
  - MoCo - Momentum Contrast
  - SimCLR - End-to-End Instance Contrast

Summary of contrastive learning

- Contrastive Learning: Dictionary look up task.
- Predictive Coding (or) Instance Discrimination
- Mechanism for dictionary lookup: (1) end-to-end, (2) momentum
- Predictive Coding Success: CPCv2
- Instance Discrimination Success:
  - MoCo - Momentum Contrast
  - SimCLR - End-to-End Instance Contrast

Summary of contrastive learning

- Contrastive Learning: Dictionary look up task.
- Predictive Coding (or) Instance Discrimination
- Mechanism for dictionary lookup: (1) end-to-end, (2) momentum
- Predictive Coding Success: CPCv2
- Instance Discrimination Success:
  - MoCo - Momentum Contrast
  - SimCLR - End-to-End Instance Contrast

Summary of contrastive learning

- Contrastive Learning: Dictionary look up task.
- Predictive Coding (or) Instance Discrimination
- Mechanism for dictionary lookup: (1) end-to-end, (2) momentum
- Predictive Coding Success: CPCv2
- Instance Discrimination Success:
  - MoCo - Momentum Contrast
  - SimCLR - End-to-End Instance Contrast

Summary of contrastive learning

- Contrastive Learning: Dictionary look up task.
- Predictive Coding (or) Instance Discrimination
- Mechanism for dictionary lookup: (1) end-to-end, (2) momentum
- Predictive Coding Success: CPCv2
- Instance Discrimination Success:
  - MoCo - Momentum Contrast
  - SimCLR - End-to-End Instance Contrast

Critical view of CPCv2

- Advantages:
  - Spatial Prediction and Contrast is generic and can apply to any modality or domain where you do not already know the underlying invariances.
  - In principle, can be used to train a latent space generative model.
  - Easier to adopt for video, audio and text and perform multimodal training.
- Disadvantages:
  - Splitting raw input to patches, or frames and patches, or audio chunks involves design choices for patch size, stride, etc.
  - Multiple forward passes of smaller versions of inputs. Pre-training on smaller images may not be optimal for downstream task.
  - Applying BatchNorm is hard.
  - Split and process mechanism is slow on GEMM specialized hardware like TPU

Critical view of CPCv2

- Advantages:
  - Spatial Prediction and Contrast is generic and can apply to any modality or domain where you do not already know the underlying invariances.
  - In principle, can be used to train a latent space generative model.
  - Easier to adopt for video, audio and text and perform multimodal training.
- Disadvantages:
  - Splitting raw input to patches, or frames and patches, or audio chunks involves design choices for patch size, stride, etc.
  - Multiple forward passes of smaller versions of inputs. Pre-training on smaller images may not be optimal for downstream task.
  - Applying BatchNorm is hard.
  - Split and process mechanism is slow on GEMM specialized hardware like TPU

Critical view of CPCv2

- Advantages:
  - Spatial Prediction and Contrast is generic and can apply to any modality or domain where you do not already know the underlying invariances.
  - In principle, can be used to train a latent space generative model.
  - Easier to adopt for video, audio and text and perform multimodal training.
- Disadvantages:
  - Splitting raw input to patches, or frames and patches, or audio chunks involves design choices for patch size, stride, etc.
  - Multiple forward passes of smaller versions of inputs. Pre-training on smaller images may not be optimal for downstream task.
  - Applying BatchNorm is hard.
  - Split and process mechanism is slow on GEMM specialized hardware like TPU

Critical view of CPCv2

- Advantages:
  - Spatial Prediction and Contrast is generic and can apply to any modality or domain where you do not already know the underlying invariances.
  - In principle, can be used to train a latent space generative model.
  - Easier to adopt for video, audio and text and perform multimodal training.
- Disadvantages:
  - Splitting raw input to patches, or frames and patches, or audio chunks involves design choices for patch size, stride, etc.
  - Multiple forward passes of smaller versions of inputs. Pre-training on smaller images may not be optimal for downstream task.
  - Applying BatchNorm is hard.
  - Split and process mechanism is slow on GEMM specialized hardware like TPU

Critical view of CPCv2

- Advantages:
  - Spatial Prediction and Contrast is generic and can apply to any modality or domain where you do not already know the underlying invariances.
  - In principle, can be used to train a latent space generative model.
  - Easier to adopt for video, audio and text and perform multimodal training.
- Disadvantages:
  - Splitting raw input to patches, or frames and patches, or audio chunks involves design choices for patch size, stride, etc.
  - Multiple forward passes of smaller versions of inputs. Pre-training on smaller images may not be optimal for downstream task.
  - Applying BatchNorm is hard.
  - Split and process mechanism is slow on GEMM specialized hardware like TPU

Critical view of CPCv2

- Advantages:
  - Spatial Prediction and Contrast is generic and can apply to any modality or domain where you do not already know the underlying invariances.
  - In principle, can be used to train a latent space generative model.
  - Easier to adopt for video, audio and text and perform multimodal training.
- Disadvantages:
  - Splitting raw input to patches, or frames and patches, or audio chunks involves design choices for patch size, stride, etc.
  - Multiple forward passes of smaller versions of inputs. Pre-training on smaller images may not be optimal for downstream task.
  - Applying BatchNorm is hard.
  - Split and process mechanism is slow on GEMM specialized hardware like TPU

Critical view of CPCv2

- Advantages:
  - Spatial Prediction and Contrast is generic and can apply to any modality or domain where you do not already know the underlying invariances.
  - In principle, can be used to train a latent space generative model.
  - Easier to adopt for video, audio and text and perform multimodal training.
- Disadvantages:
  - Splitting raw input to patches, or frames and patches, or audio chunks involves design choices for patch size, stride, etc.
  - Multiple forward passes of smaller versions of inputs. Pre-training on smaller images may not be optimal for downstream task.
  - Applying BatchNorm is hard.
  - Split and process mechanism is slow on GEMM specialized hardware like TPU

Critical view of CPCv2

- Advantages:
  - Spatial Prediction and Contrast is generic and can apply to any modality or domain where you do not already know the underlying invariances.
  - In principle, can be used to train a latent space generative model.
  - Easier to adopt for video, audio and text and perform multimodal training.
- Disadvantages:
  - Splitting raw input to patches, or frames and patches, or audio chunks involves design choices for patch size, stride, etc.
  - Multiple forward passes of smaller versions of inputs. Pre-training on smaller images may not be optimal for downstream task.
  - Applying BatchNorm is hard.
  - Split and process mechanism is slow on GEMM specialized hardware like TPU

Critical view of MoCo

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Momentum encoder adds stability and works with SGD+momentum.
  - Memory bank decouples batch size from number of negatives.
- Disadvantages:
  - Momentum update adds hyperparameter for decay rate.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of MoCo

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Momentum encoder adds stability and works with SGD+momentum.
  - Memory bank decouples batch size from number of negatives.
- Disadvantages:
  - Momentum update adds hyperparameter for decay rate.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of MoCo

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Momentum encoder adds stability and works with SGD+momentum.
  - Memory bank decouples batch size from number of negatives.
- Disadvantages:
  - Momentum update adds hyperparameter for decay rate.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of MoCo

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Momentum encoder adds stability and works with SGD+momentum.
  - Memory bank decouples batch size from number of negatives.
- Disadvantages:
  - Momentum update adds hyperparameter for decay rate.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of MoCo

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Momentum encoder adds stability and works with SGD+momentum.
  - Memory bank decouples batch size from number of negatives.
- Disadvantages:
  - Momentum update adds hyperparameter for decay rate.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of MoCo

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Momentum encoder adds stability and works with SGD+momentum.
  - Memory bank decouples batch size from number of negatives.
- Disadvantages:
  - Momentum update adds hyperparameter for decay rate.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of MoCo

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Momentum encoder adds stability and works with SGD+momentum.
  - Memory bank decouples batch size from number of negatives.
- Disadvantages:
  - Momentum update adds hyperparameter for decay rate.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of SimCLR

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Avoids using memory bank and therefore literally looks like supervised learning.
- Disadvantages:
  - Needs large batch sizes for sufficient negatives. This in turn needs lot of compute (TPUs) and shared negatives across workers.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of SimCLR

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Avoids using memory bank and therefore literally looks like supervised learning.
- Disadvantages:
  - Needs large batch sizes for sufficient negatives. This in turn needs lot of compute (TPUs) and shared negatives across workers.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of SimCLR

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Avoids using memory bank and therefore literally looks like supervised learning.
- Disadvantages:
  - Needs large batch sizes for sufficient negatives. This in turn needs lot of compute (TPUs) and shared negatives across workers.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of SimCLR

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Avoids using memory bank and therefore literally looks like supervised learning.
- Disadvantages:
  - Needs large batch sizes for sufficient negatives. This in turn needs lot of compute (TPUs) and shared negatives across workers.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Critical view of SimCLR

- Advantages:
  - Minimalistic design. Very simple to use and replicate.
  - No architecture change. Can be easily used for downstream tasks.
  - Invariance distillation for images is ideal and makes pre-training look like supervised learning.
  - Avoids using memory bank and therefore literally looks like supervised learning.
- Disadvantages:
  - Needs large batch sizes for sufficient negatives. This in turn needs lot of compute (TPUs) and shared negatives across workers.
  - Invariance to data augmentation may not be readily applicable to other modalities like text, audio, video.

Future of Self-Supervision

- Completely close the gap between self-supervised and supervised for the same amount of compute, data augmentation and training time.
- Fine-tuning to downstream tasks: Only few tasks benefit a lot from self-supervised backbones relative to supervised. Objectives more correlated with downstream tasks other than classification may be necessary.
- Still requires well curated dataset like ImageNet even if labels aren't used. Ideal dream of self-supervised learning is to learn from raw data from the real world or internet without much filtering.

Future of Self-Supervision

- Completely close the gap between self-supervised and supervised for the same amount of compute, data augmentation and training time.

- Fine-tuning to downstream tasks: Only few tasks benefit a lot from self-supervised backbones relative to supervised. Objectives more correlated with downstream tasks other than classification may be necessary.

- Still requires well curated dataset like ImageNet even if labels aren't used. Ideal dream of self-supervised learning is to learn from raw data from the real world or internet without much filtering.

Future of Self-Supervision

- Completely close the gap between self-supervised and supervised for the same amount of compute, data augmentation and training time.
- Fine-tuning to downstream tasks: Only few tasks benefit a lot from self-supervised backbones relative to supervised. Objectives more correlated with downstream tasks other than classification may be necessary.
- Still requires well curated dataset like ImageNet even if labels aren't used. Ideal dream of self-supervised learning is to learn from raw data from the real world or internet without much filtering.

Future of Self-Supervision

- Completely close the gap between self-supervised and supervised for the same amount of compute, data augmentation and training time.

- Fine-tuning to downstream tasks: Only few tasks benefit a lot from self-supervised backbones relative to supervised. Objectives more correlated with downstream tasks other than classification may be necessary.

- Still requires well curated dataset like ImageNet even if labels aren't used. Ideal dream of self-supervised learning is to learn from raw data from the real world or internet without much filtering.

Generation or not?

Modeling future in latent spaces

- Modeling in pixel space is really hard especially for high dimensional videos
- Intelligent agents can reason over long horizons very quickly (faster than real time)
- Build internal world models that run on an abstract space
- Abstractions must capture useful aspects of sensory stream and ignore noise

Modeling future in latent spaces

- Modeling in pixel space is really hard especially for high dimensional videos
- Intelligent agents can reason over long horizons very quickly (faster than real time)
- Build internal world models that run on an abstract space
- Abstractions must capture useful aspects of sensory stream and ignore noise

Modeling future in latent spaces

- Modeling in pixel space is really hard especially for high dimensional videos
- Intelligent agents can reason over long horizons very quickly (faster than real time)
- Build internal world models that run on an abstract space
- Abstractions must capture useful aspects of sensory stream and ignore noise

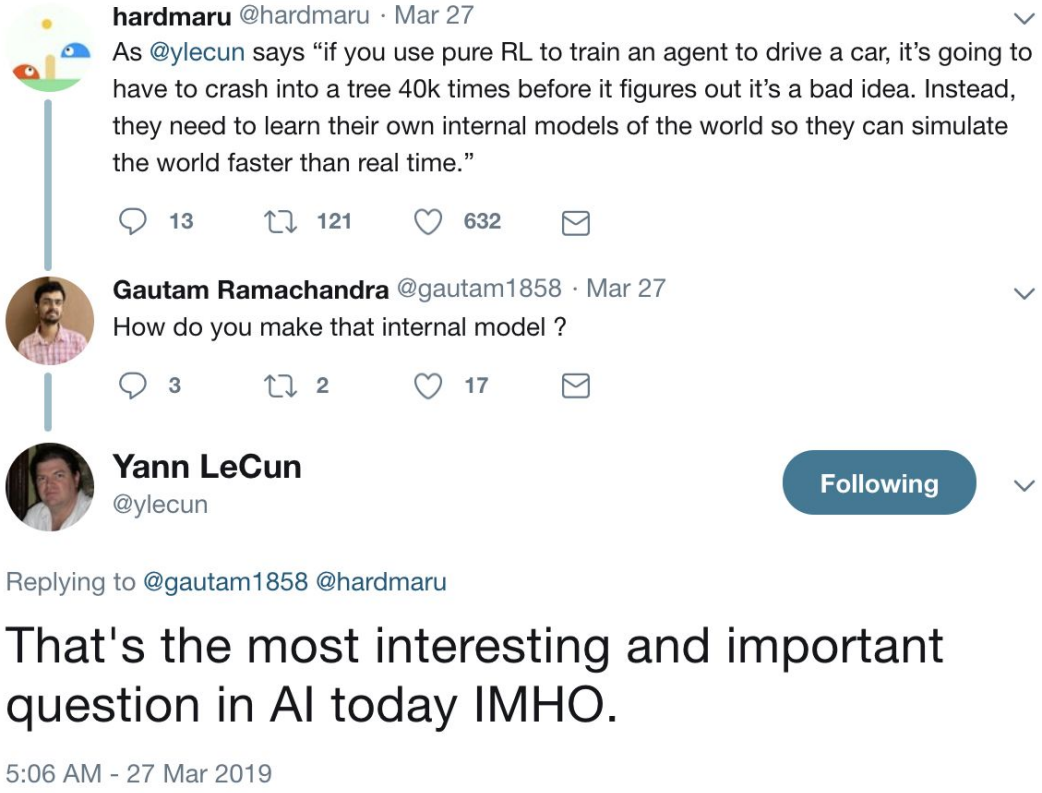
Modeling future in latent spaces

- Modeling in pixel space is really hard especially for high dimensional videos
- Intelligent agents can reason over long horizons very quickly (faster than real time)
- Build internal world models that run on an abstract space
- Abstractions must capture useful aspects of sensory stream and ignore noise

Modeling future in latent spaces

- Modeling in pixel space is really hard especially for high dimensional videos
- Intelligent agents can reason over long horizons very quickly (faster than real time)
- Build internal world models that run on an abstract space
- Abstractions must capture useful aspects of sensory stream and ignore noise

Modeling future in latent spaces



hardmaru @hardmaru · Mar 27

As @ylecun says “if you use pure RL to train an agent to drive a car, it’s going to have to crash into a tree 40k times before it figures out it’s a bad idea. Instead, they need to learn their own internal models of the world so they can simulate the world faster than real time.”

13 121 632

Gautam Ramachandra @gautam1858 · Mar 27

How do you make that internal model ?

3 2 17

Yann LeCun
@ylecun

Following

Replying to @gautam1858 @hardmaru

That's the most interesting and important question in AI today IMHO.

5:06 AM - 27 Mar 2019

Current state of self-supervision

- Promising progress in closing the gap with respect to pure supervised learning on Imagenet, Librispeech (CPCv2, MoCo, SimCLR, MoCo v2).
- Transfer Learning results in Language (BERT, word2vec, fastText, ...)
- Transfer Learning in Vision (CPCv2, MoCo, MoCo v2)
- Close to nothing in Reinforcement Learning
- Promise / main hope as far as building general intelligence is concerned - transfer learning, learning reusable concepts / abstractions, building plans and imaginations, reasoning using constructed plans, etc.

Current state of self-supervision

- Promising progress in closing the gap with respect to pure supervised learning on Imagenet, Librispeech (CPCv2, MoCo, SimCLR, MoCo v2).
- Transfer Learning results in Language (BERT, word2vec, fastText, ...)
- Transfer Learning in Vision (CPCv2, MoCo, MoCo v2)
- Close to nothing in Reinforcement Learning
- Promise / main hope as far as building general intelligence is concerned - transfer learning, learning reusable concepts / abstractions, building plans and imaginations, reasoning using constructed plans, etc.

Current state of self-supervision

- Promising progress in closing the gap with respect to pure supervised learning on Imagenet, Librispeech (CPCv2, MoCo, SimCLR, MoCo v2).
- Transfer Learning results in Language (BERT, word2vec, fastText, ...)
- Transfer Learning in Vision (CPCv2, MoCo, MoCo v2)
- Close to nothing in Reinforcement Learning
- Promise / main hope as far as building general intelligence is concerned - transfer learning, learning reusable concepts / abstractions, building plans and imaginations, reasoning using constructed plans, etc.

Current state of self-supervision

- Promising progress in closing the gap with respect to pure supervised learning on Imagenet, Librispeech (CPCv2, MoCo, SimCLR, MoCo v2).
- Transfer Learning results in Language (BERT, word2vec, fastText, ...)
- Transfer Learning in Vision (CPCv2, MoCo, MoCo v2)
- Close to nothing in Reinforcement Learning
- Promise / main hope as far as building general intelligence is concerned - transfer learning, learning reusable concepts / abstractions, building plans and imaginations, reasoning using constructed plans, etc.

Current state of self-supervision

- Promising progress in closing the gap with respect to pure supervised learning on Imagenet, Librispeech (CPCv2, MoCo, SimCLR, MoCo v2).
- Transfer Learning results in Language (BERT, word2vec, fastText, ...)
- Transfer Learning in Vision (CPCv2, MoCo, MoCo v2)
- Close to nothing in Reinforcement Learning
- Promise / main hope as far as building general intelligence is concerned - transfer learning, learning reusable concepts / abstractions, building plans and imaginations, reasoning using constructed plans, etc.

Current state of self-supervision

- Promising progress in closing the gap with respect to pure supervised learning on Imagenet, Librispeech (CPCv2, MoCo, SimCLR, MoCo v2).
- Transfer Learning results in Language (BERT, word2vec, fastText, ...)
- Transfer Learning in Vision (CPCv2, MoCo, MoCo v2)
- Close to nothing in Reinforcement Learning
- Promise / main hope as far as building general intelligence is concerned - transfer learning, learning reusable concepts / abstractions, building plans and imaginations, reasoning using constructed plans, etc.

Let's end it with the cake

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Yann LeCun's cake