

Lab CudaVision
Learning Vision Systems on Graphics Cards (MA-INF 4308)

Autoencoders

11.12.2024

PROF. SVEN BEHNKE, ANGEL VILLAR-CORRALES

Contact: villar@ais.uni-bonn.de

Learning without Supervision

Motivation

- Labeled data is scarce
- Labeling is time consuming and expensive
- Category labelling in COCO:
 - 330k images
 - 91 classes
- Instance segmentation in COCO:
 - 2.5 million instances
- Semantic segmentation in Cityscapes:
 - > 5000 fully annotated frames



20 s/img



85 s/instance



1.5 hours/img

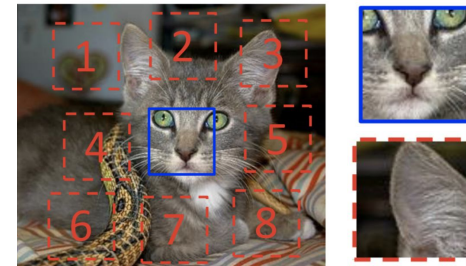
amazon
mechanical turk



Solution

- Weak supervision or no supervision
- **Weakly supervised learning:**
 - Using labels from a related task
- **Semi-supervised learning:**
 - Using large datasets with only few labeled data
- **Unsupervised learning:**
 - Using no labeled data
- **Self-supervised learning:**
 - Learning representations on pretext tasks

Brushing teeth



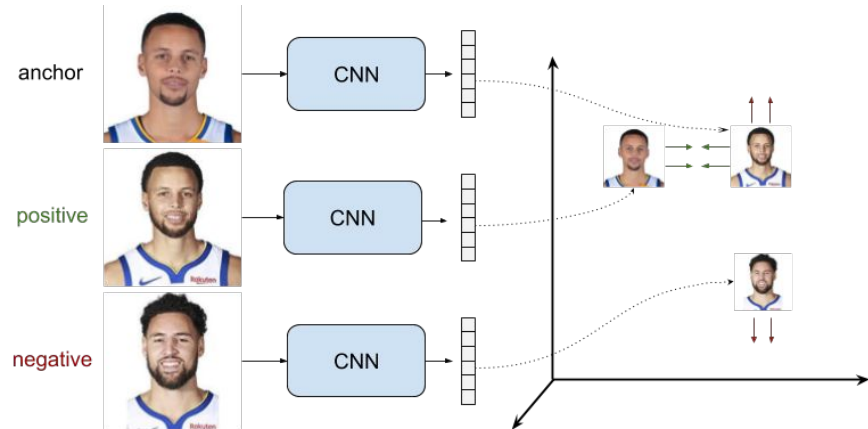
Applications

- Clustering and Similarity learning

Image Clustering

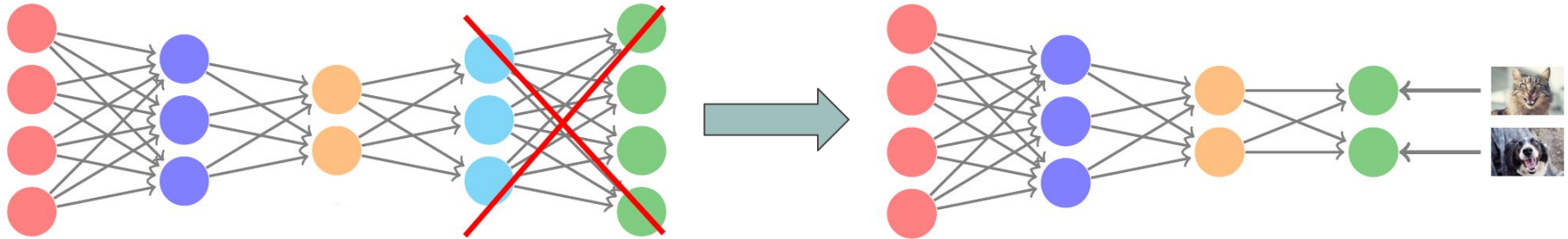


Deep Metric Learning



Applications

- Network initialization:
 - Model pretraining
 - Transfer learning

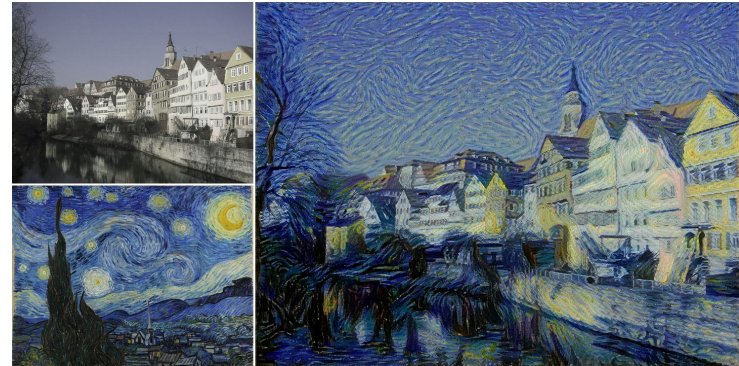


Applications

- Generative modelling
 - Generating new images
 - Image to image translation
 - Impainting and missing data

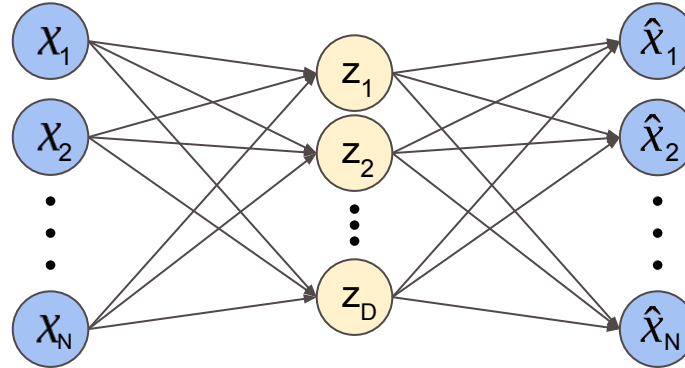


<https://thispersondoesnotexist.com/>



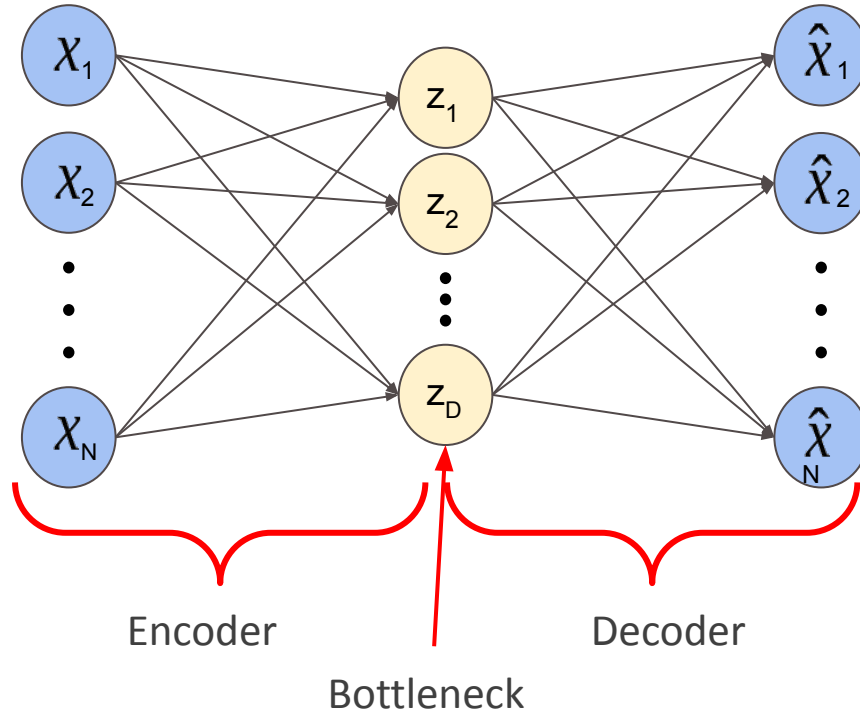
Autoencoders

Autoencoder (AE)



- Models that are trained to predict their input
 - Dimensionality reduction
 - Representation learning
- Autoencoders learn an approximation of the identity

Autoencoder (AE)

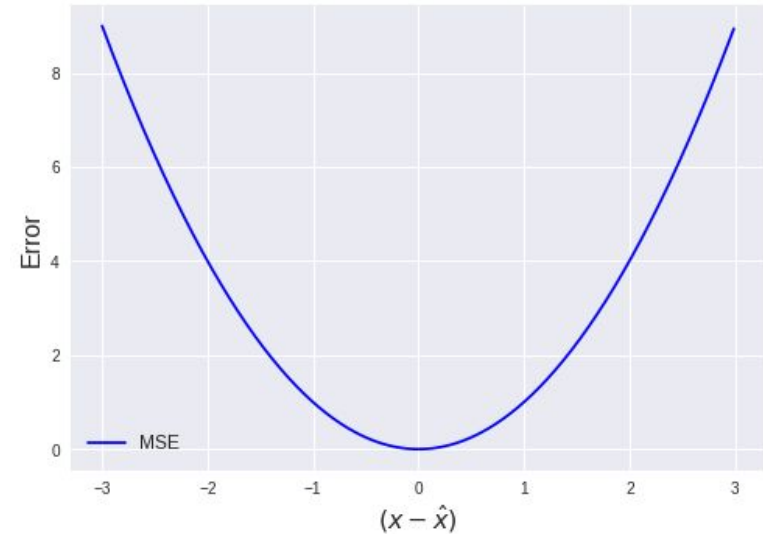


- $\mathbf{z} = E(\mathbf{X})$
- $\hat{\mathbf{X}} = D(\mathbf{z})$
- $\hat{\mathbf{X}} = D(E(\mathbf{X}))$

Training Autoencoders

- AEs are often trained with regression losses

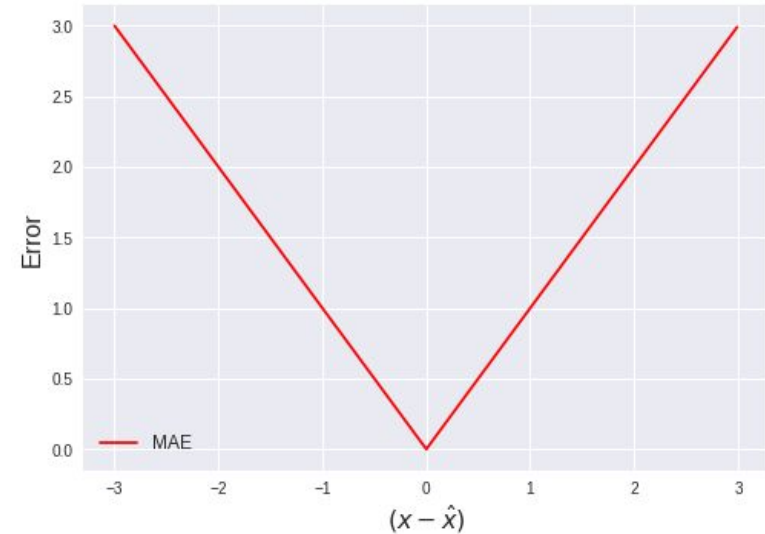
$$MSE = \frac{1}{N} \sum_i^N (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2$$



Training Autoencoders

- AEs are often trained with regression losses

$$MAE = \frac{1}{N} \sum_i^N |\mathbf{X}_i - \hat{\mathbf{X}}_i|$$



Training Autoencoders

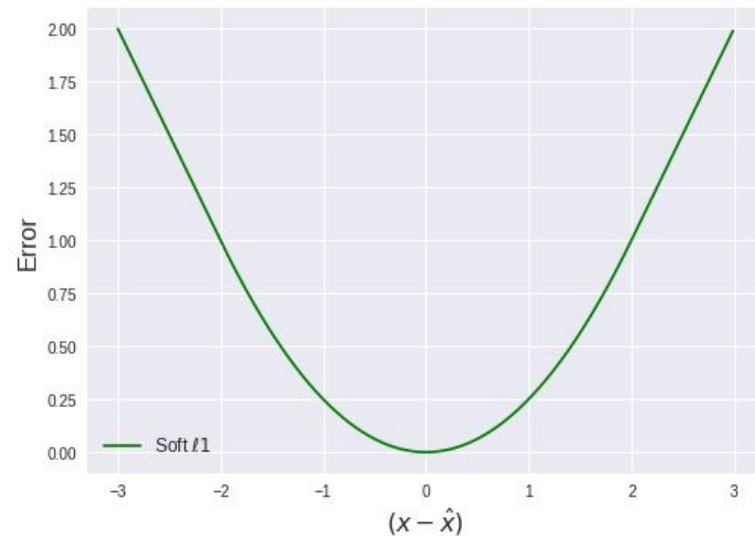
- AEs are often trained with regression losses

$$\text{Smooth } \ell 1 = \frac{1}{N} \sum_i l_i$$

$$l_i = \begin{cases} \frac{1}{2 \cdot \beta} (\mathbf{X}_i - \hat{\mathbf{X}}_i)^2 & |\mathbf{X}_i - \hat{\mathbf{X}}_i| \leq \beta \\ |\mathbf{X}_i - \hat{\mathbf{X}}_i| - 0.5 \cdot \beta & \text{otherwise} \end{cases}$$

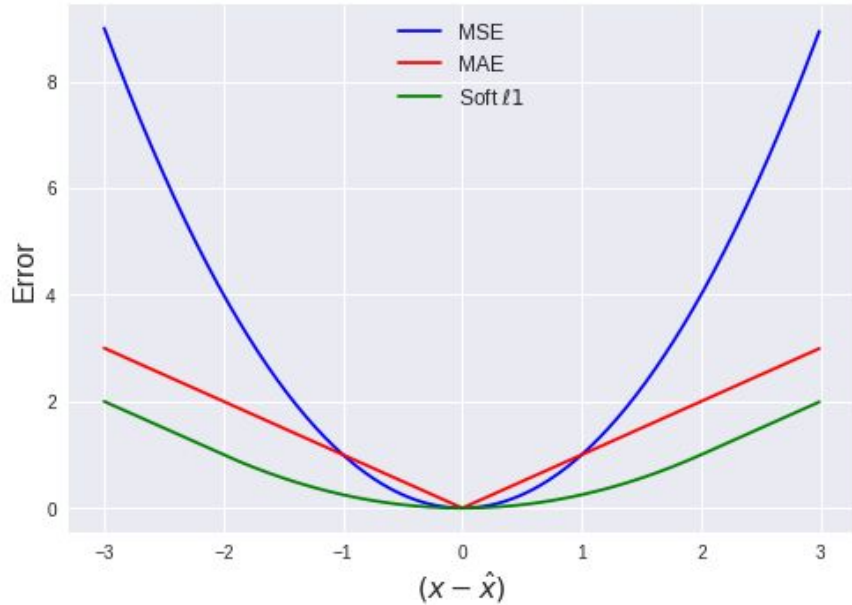
MSE for small errors

MAE for larger errors



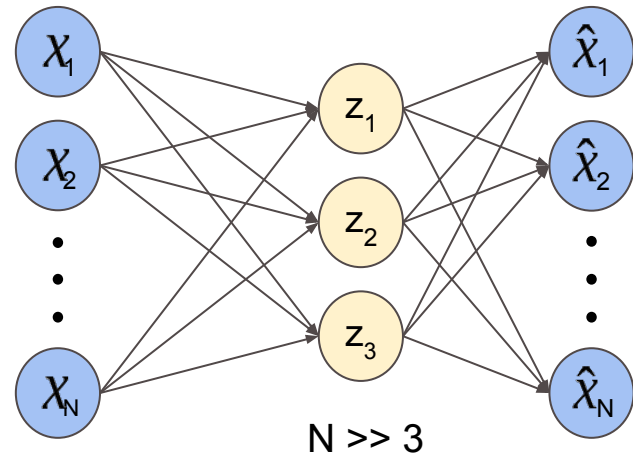
Training Autoencoders

- $MSE = \frac{1}{N} \sum_i (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2$
- $MAE = \frac{1}{N} \sum_i |\mathbf{x}_i - \hat{\mathbf{x}}_i|$
- Smooth $\ell_1 = \frac{1}{N} \sum_i l_i$
- Sigmoid + Cross Entropy



Regularizing AEs

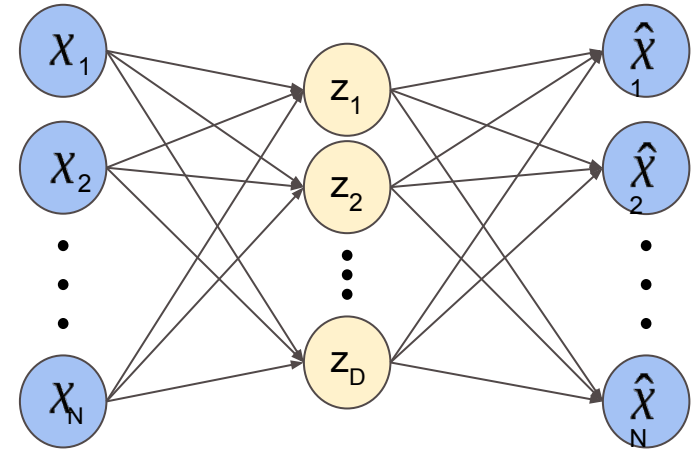
- Without regularization, AEs learn an identity map
 - Enforce constraints on the architecture or loss
- Undercomplete AE**
 - Low dimensional bottleneck
 - Prevents learning the identity
 - Enforces compression
- AE with one linear layer learns PCA
 - Encoder equivalent to projection matrix



Regularizing AEs

- Without regularization, AEs learn an identity map
 - Enforce constraints on the architecture or loss
- Sparse AE**
 - Enforce sparsity in bottleneck

$$\mathcal{L}_{\text{SAE}}(\mathbf{X}, \hat{\mathbf{X}}) = \mathcal{L}(\mathbf{X}, \hat{\mathbf{X}}) + \underbrace{\frac{1}{D} \sum_{i=1}^D |z_i|}_{\text{L1 reg.}}$$



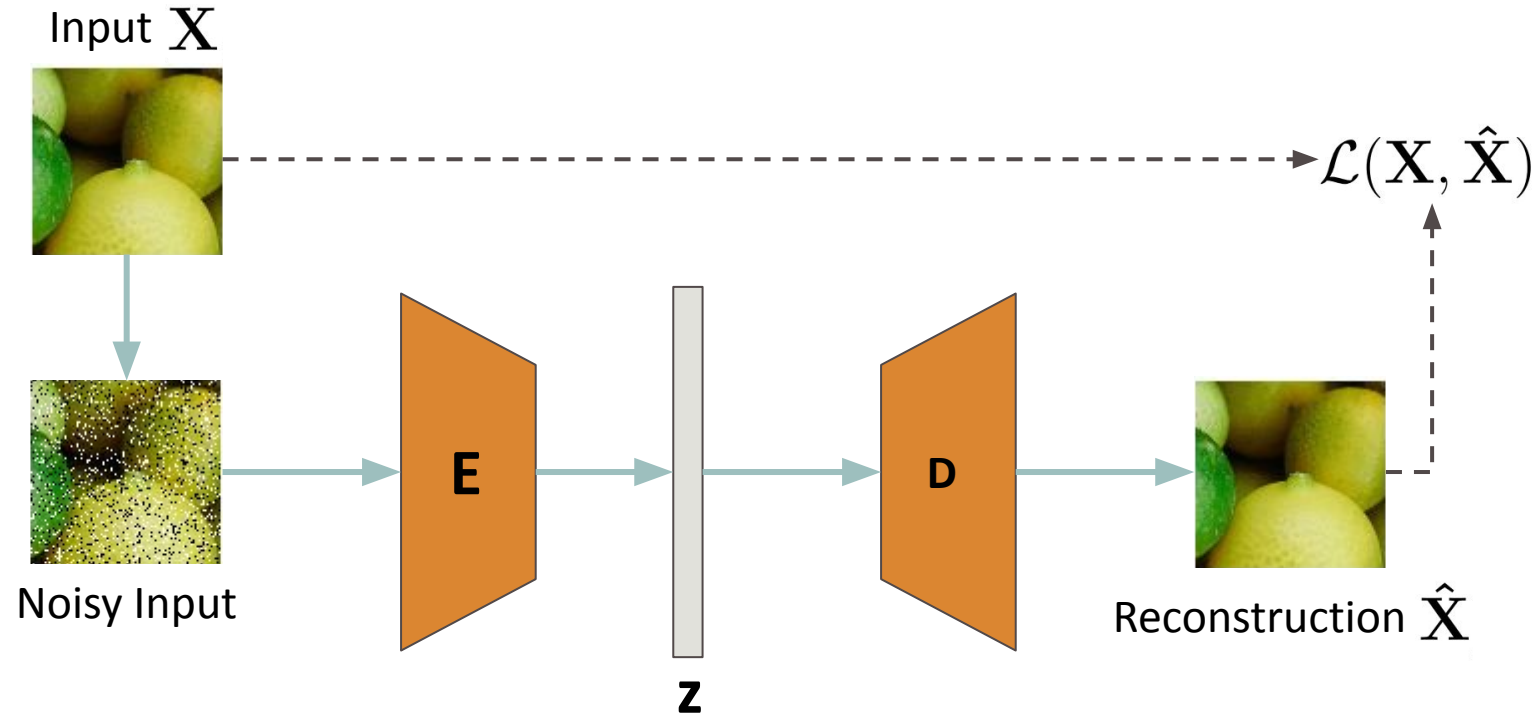
Denoising Autoencoders

Denoising Autoencoder (DAE)

- **Denoising:** *removing noise from a signal, keeping as much information and features as possible*
- AEs excel at performing denoising in images
- Information bottleneck
 - Required features to reconstruct input
 - Noise does not contain information
- Pretext task for learning robust representations



DAE Pipeline

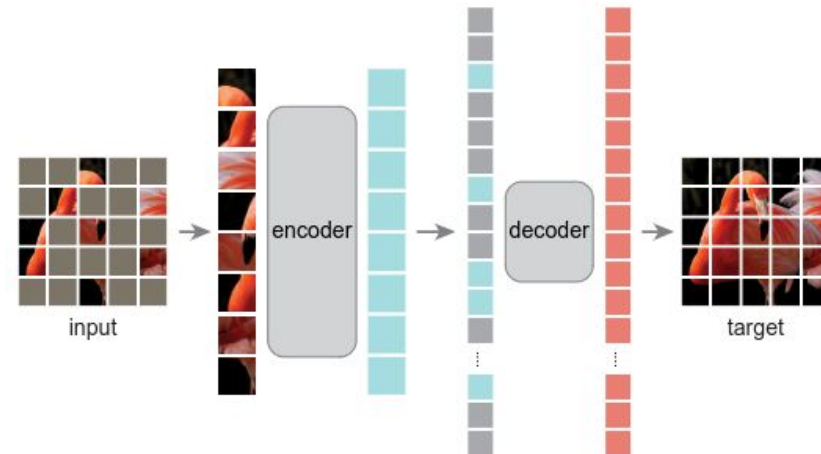


Masked Autoencoders (MAE)

- State-of-the-art self-supervised pretraining algorithm
- Pretrain large transformers using images only

method	pre-train data	ViT-B	ViT-L	ViT-H
scratch, our impl.	-	82.3	82.6	83.1
DINO [5]	IN1K	82.8	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-
BEiT [2]	IN1K+DALI	83.2	85.2	-
MAE	IN1K	<u>83.6</u>	<u>85.9</u>	<u>86.9</u>

1. Split image into patches
2. Throw away many patches (e.g. 80%)
3. Encode only the unmasked patches with deep transformer encoder (very efficient)
4. Decode embeddings of masked & unmasked patches with shallow transformer decoder
5. Compute recons. loss on masked patches



Let's try it!

Variational Autoencoders

Variational Autoencoder (VAE)

- AEs compute a **deterministic** latent vector for the input
 - Unstructured latent space
 - Deterministic mapping
 - Cannot be used to generate new data!
- Variational autoencoders (VAEs):
 - Describe latent space in a **probabilistic** manner
- VAEs map inputs into a probability distribution
 - Model uncertainty in the input data
 - Enforces smooth latent space

Auto-Encoding Variational Bayes

Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

Abstract

How can we perform efficient inference and learning in directed probabilistic models, in the presence of continuous latent variables with intractable posterior distributions, and large datasets? We introduce a stochastic variational inference and learning algorithm that scales to large datasets and, under some mild differentiability conditions, even works in the intractable case. Our contributions is two-fold. First, we show that a reparameterization of the variational lower bound with an independent noise variable yields a lower bound estimator that can be jointly optimized w.r.t. variational and generative parameters using standard gradient-based stochastic optimization methods. Second, we show that posterior inference can be made especially efficient by optimizing a probabilistic encoder (also called a recognition model) to approximate the intractable posterior, using the proposed estimator. Theoretical advantages are reflected in experimental results.

1 Introduction

How can we efficiently learn the parameters of directed probabilistic models whose continuous latent variables have intractable posterior distributions? The variational approach to Bayesian inference involves the introduction of an approximation to the intractable posterior, used to maximize the variational lower bound on the marginal likelihood. Unfortunately, the common mean-field approach requires analytical solutions of expectations w.r.t. the approximate posterior, which are also intractable in the general case. We show how a reparameterization of the variational lower bound yields a practical differentiable estimator of the lower bound. This SGVB (Stochastic Gradient Variational Bayes) estimator can be straightforwardly used as a stochastic objective function, and that can be jointly optimized w.r.t. both the variational and generative parameters, using standard stochastic gradient ascent techniques.

VAE: Statistical Motivation

- **Assumption:** Sample \mathbf{X} is generated by decoding latent variable \mathbf{z}
- Training VAE corresponds to determining $p(\mathbf{z}|\mathbf{X})$
 - Usually undefined and intractable
- Approximate $p(\mathbf{z}|\mathbf{X})$ by a tractable distribution $q(\mathbf{z}|\mathbf{X})$

$$\min \text{KL}(p(\mathbf{z}|\mathbf{X})||q(\mathbf{z}|\mathbf{X}))$$

which is equivalent to

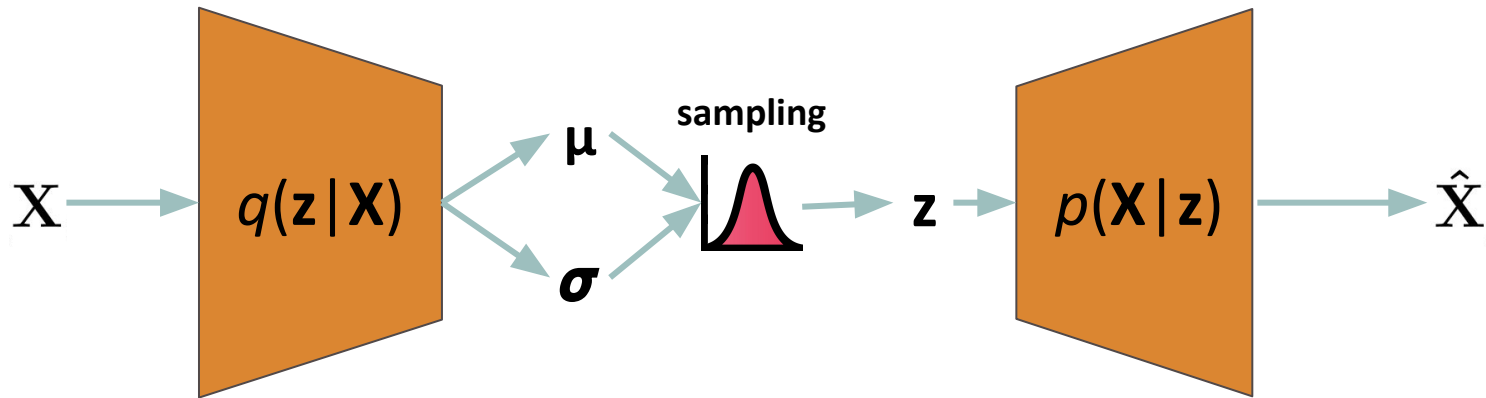
ELBO

$$\max \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{X})} \log p(\mathbf{X}|\mathbf{z})}_{\text{Reconstruction likelihood}} - \underbrace{\text{KL}(q(\mathbf{z}|\mathbf{X})||p(\mathbf{z}))}_{\text{Difference between } q(\mathbf{z}|\mathbf{X}) \text{ and true prior}}$$

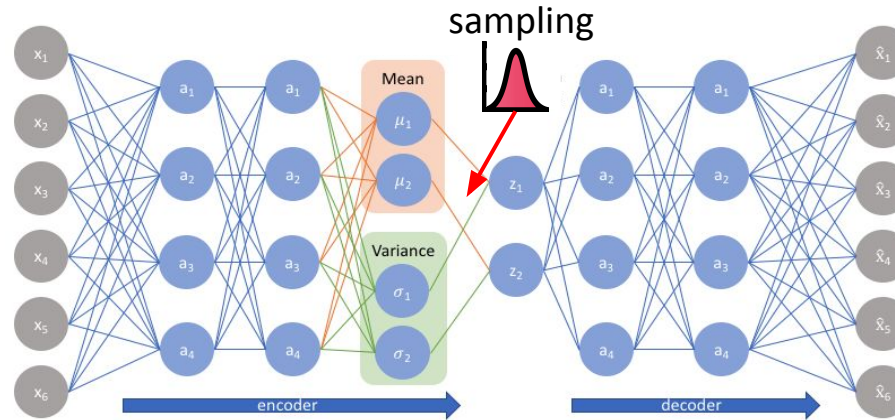
Actual distribution of the latent space

VAE: Statistical Motivation

- $p(\mathbf{z})$ is often assumed to be an Isotropic Gaussian distribution
 - For determining $q(\mathbf{z}|\mathbf{X})$ we just need $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$
 - We use neural networks to estimate $q(\mathbf{z}|\mathbf{X})$ and $p(\mathbf{X}|\mathbf{z})$



VAE Training



Loss requires sampling
How do we solve this?

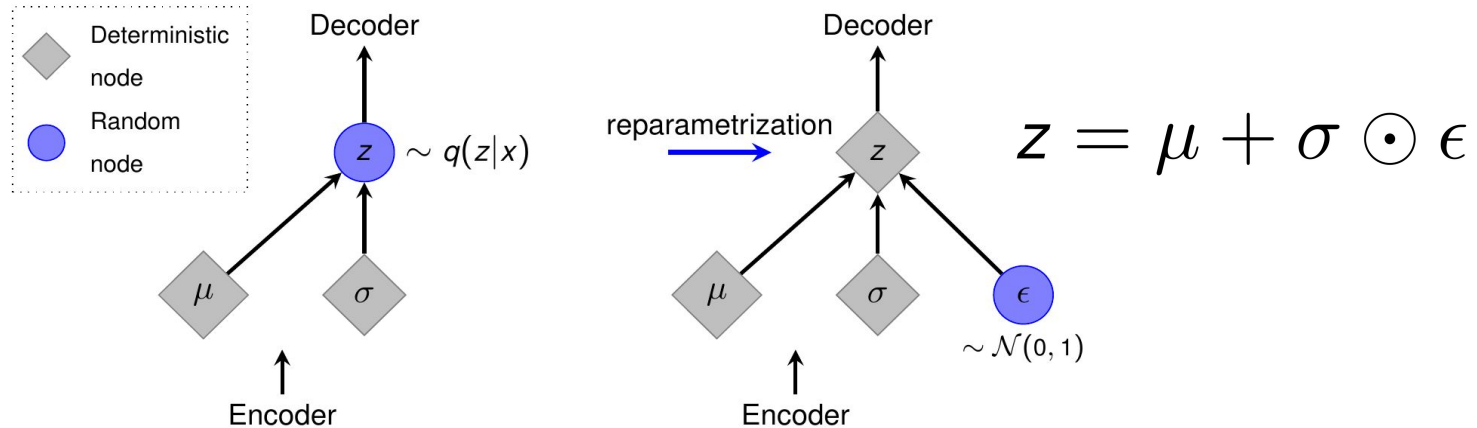


$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{Recons}} - \mathcal{L}_{\text{KL}}$$

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q(\mathbf{z}|\mathbf{X})} \log p(\mathbf{X}|\mathbf{z}) - \text{KL}(q(\mathbf{z}|\mathbf{X}) || p(\mathbf{z}))$$

Reparametrization Trick

- We cannot propagate through random sampling
- Move random sampling out of path by reparametrization
 - Backpropagation is deterministic



VAE as Generative Models

- New data can be generated by sampling from latent space distribution
 - Use learned mean and covariance
 - Sample from distribution
 - Reconstruct using decoder
- Diagonal covariance enforces independent latent variables
- Smooth latent space can be transversed



What can we do with modern VAEs?

Image Generation



NVAE: A Deep Hierarchical Variational Autoencoder

Arash Vahdat, Jan Kautz
NVIDIA
{avahdat, jkautz}@nvidia.com

Abstract

Normalizing flows, autoregressive models, variational autoencoders (VAEs), and deep energy-based models are among competing likelihood-based frameworks for deep generative learning. Among them, VAEs have the advantage of fast and tractable sampling and easy-to-access encoding networks. However, they are currently outperformed by other models such as normalizing flows and autoregressive models. While the majority of the research in VAEs is focused on the statistical challenges, we explore the orthogonal direction of carefully designing neural architectures for hierarchical VAEs. We propose Nouveau VAE (NVAE), a deep hierarchical VAE built for image generation using depth-wise separable convolutions and batch normalization. NVAE is equipped with a residual parameterization of Normal distributions and its training is stabilized by spectral regularization. We show that NVAE achieves state-of-the-art results among non-autoregressive likelihood-based models on the MNIST, CIFAR-10, CelebA 64, and CelebA HQ datasets and it provides a strong baseline on FFHQ. For example, on CIFAR-10, NVAE pushes the state-of-the-art from 2.98 to 2.91 bits per dimension, and it produces high-quality images on CelebA HQ as shown in Fig. 1. To the best of our knowledge, NVAE is the first successful VAE applied to natural images as large as 256 × 256 pixels. The source code is available at <https://github.com/AV1998/NVAE>.

1 Introduction

The majority of the research efforts on improving VAEs [1, 2] is dedicated to the statistical challenges, such as reducing the gap between approximate and true posterior distributions [3, 4, 5, 6, 7, 8, 9, 10], formulating tighter bounds [11, 12, 13, 14], reducing the gradient noise [15, 16], extending VAEs to discrete variables [17, 18, 19, 20, 21, 22, 23], or tackling posterior collapse [24, 25, 26, 27]. The role of neural network architectures for VAEs is somewhat overlooked, as most previous work borrows the architectures from classification tasks.



Figure 1: 256 × 256-pixel samples generated by NVAE, trained on CelebA HQ [28].

However, VAEs can benefit from designing special network architectures as they have fundamentally different requirements. First, VAEs maximize the mutual information between the input and latent variables [29, 30], requiring the networks to retain the information content of the input data as much

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

arXiv:2007.03898v3 [stat.ML] 8 Jan 2021

What can we do with modern VAEs?

Video Prediction



arXiv:1911.01655v1 [cs.CV] 5 Nov 2019

High Fidelity Video Prediction with Large Stochastic Recurrent Neural Networks

Ruben Villegas^{1,4} Arkanath Pathak¹ Harini Kannan²
Dumitru Erhan² Quoc V. Le³ Honglak Lee²

¹ University of Michigan

² Google Research

³ Google

⁴ Adobe Research

Abstract

Predicting future video frames is extremely challenging, as there are many factors of variation that make up the dynamics of how frames change through time. Previously proposed solutions require complex inductive biases inside network architectures with highly specialized computation, including segmentation masks, optical flow, and foreground and background separation. In this work, we question if such handcrafted architectures are necessary and instead propose a different approach: finding minimal inductive bias for video prediction while maximizing network capacity. We investigate this question by performing the first large-scale empirical study and demonstrate state-of-the-art performance by learning large models on three different datasets: one for modeling object interactions, one for modeling human motion, and one for modeling car driving¹.

1 Introduction

From throwing a ball to driving a car, humans are very good at being able to interact with objects in the world and anticipate the results of their actions. Being able to teach agents to do the same has enormous possibilities for training intelligent agents capable of generalizing to many tasks. Model-based reinforcement learning is one such technique that seeks to do this – by first learning a model of the world, and then by planning with the learned model. There has been some recent success with training agents in this manner by first using video prediction to model the world. Particularly, video prediction models combined with simple planning algorithms [Lillicrap et al., 2012] or policy-based learning [Kaiser et al., 2019] for model-based reinforcement learning have been shown to perform equally or better than model-free methods with far less interactions with the environment. Additionally, Ebert et al. [2018] showed that video prediction methods are also useful for robotic control, especially with regards to specifying unstructured goal positions.

However, training an agent to accurately predict what will happen next is still an open problem. Video prediction, the task of generating future frames given context frames, is notoriously hard. There are many spatio-temporal factors of variation present in videos that make this problem very difficult for neural networks to model. Many methods have been proposed to tackle this problem [Oh et al., 2015, Finn et al., 2016, Vondrick et al., 2016, Villegas et al., 2017a, Lotter et al., 2017, Tulyakov et al., 2018, Liang et al., 2017, Denton and Birodkar, 2017, Wichers et al., 2018, Babaeizadeh et al., 2018, Denton and Fergus, 2018, Lee et al., 2018, Byeon et al., 2018, Yan et al., 2018, Kumar et al., 2018]. Most of these works propose some type of separation of information streams (e.g., motion/pose and content streams), specialized computations (e.g., warping, optical flow, foreground/background masks, predictive coding, etc), additional high-level information (e.g., landmarks, semantic segmentation masks, etc) or are simply shown to work in relatively simpler environments (e.g., Atari, synthetic shapes, centered human faces and bodies, etc).

¹This work was done while the first author was an intern at Google

What can we do with modern VAEs?

Audio/music Generation

Vision Tasks

Planning

and much more!

A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music

Adam Roberts¹ Jesse Engel¹ Colin Raffel¹ Curtis Hawthorne² Douglas Eck¹

Abstract

The Variational Autoencoder (VAE) has proven to be an effective model for producing semantically meaningful latent representations for natural data. However, it has thus far seen limited application to sequential data, and, as we demonstrate, existing recurrent VAE models have difficulty modeling sequences with long-term structure. To address this issue, we propose the use of a *hierarchical* decoder, which first outputs embeddings for sub-sequences of the input and then uses these embeddings to generate each subsequent independently. We apply this architecture to modeling sequences of musical notes and find that it exhibits dramatically better performance, interpolation, and reconstruction performance than a “flat” baseline model. An implementation of our “MusicVAE” is available online.²

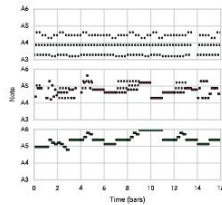


Figure 1. Demonstration of latent-space averaging using MusicVAE. The latent codes for the top and bottom sequences are averaged and decoded by our model to produce the middle sequence. The latent-space mean involves a similar repeating pattern to the top sequence, but in a higher register and with intermittent pauses like the bottom sequence. Audio for this example is available in the online supplement.³ See Figs. 12 and 13 in Appendix E for a baseline comparison.

1. Introduction

Generative modeling describes the framework of estimating the underlying probability distribution $p(z)$ to generate data z . This can facilitate a wide range of applications, from sampling novel datapoints to unsupervised representation learning to estimating the probability of an existing datapoint under the learned distribution. Much recent progress in generative modeling has been expedited by the use of deep neural networks, producing “deep generative models,” which leverage the expressive power of deep networks to model complex and high-dimensional distributions. Practical achievements include generating realistic images with

millions of pixels (Karras et al., 2017), generating synthetic audio with hundreds of thousands of timesteps (Jain et al., 2016a), and achieving state-of-the-art performance on semi-supervised learning tasks (Baldi et al., 2018). A wide variety of methods have been used in deep generative modeling, including implicit models such as Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and explicit deep autoregressive models such as PixelCNN (van den Oord et al., 2016a) and WaveNet (van den Oord et al., 2016a). In this work, we focus on deep latent variable models such as the Variational Autoencoder (VAE) (Kingma & Welling, 2014; Rezakhanlou et al., 2014). The advantage of these models is that they explicitly model both $p(z)$ and $p(x|z)$, where z is a latent vector that can either be inferred from existing data or sampled from a distribution over the

Generative Semantic Segmentation

Jiaqi Chen¹ Jiachen Lu¹ Xiattian Zhu² Li Zhang^{1*}
¹Fudan University ²University of Surrey

<https://github.com/fudan-zvg/GSS>

Abstract

We present *Generative Semantic Segmentation (GSS)*, a generative learning approach for semantic segmentation. Uniquely, we cast semantic segmentation as an *image-conditioned mask generation problem*. This is achieved by replacing the conventional per-pixel discriminative learning with a latent prior learning process. Specifically, we model the variational posterior distribution of latent variables given the segmentation mask. To that end, the segmentation mask is expressed with a special type of image (dubbed as *mask*). This posterior distribution allows to generate segmentation masks unconditionally. To achieve semantic segmentation on a given image, we further introduce a conditioning network. It is optimized by minimizing the divergence between the posterior distribution of *mask* (i.e. segmentation masks) and the latent prior distribution of input training images. Extensive experiments on standard benchmarks show that our GSS can perform competitively to prior art alternatives in the standard semantic segmentation setting, whilst achieving a new state-of-the-art in the more challenging cross-domain setting.

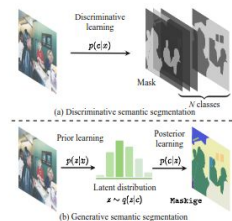


Figure 1. Schematic comparison between (a) conventional discriminative learning and (b) our generative learning based model for semantic segmentation. Our GSS introduces a latent variable z and, given the segmentation mask c , it learns the posterior distribution of z subject to the reconstruction constraint. Then, we train a conditioning network to model the prior of z by aligning with the corresponding posterior distribution. This formulation can thus generate the segmentation mask for an input image.

1. Introduction

The objective of semantic segmentation is to predict a label for every single pixel of an input image [24]. Conditioning on each pixel’s observation, existing segmentation methods [1, 2, 3, 22] naturally adopt the *discriminative learning* paradigm, along with dedicated efforts on integrating task prior knowledge (e.g., spatial correlation [19, 25, 26, 23]). For example, existing methods [1, 9, 30, 32] typically use a linear projection to optimize the log-likelihood classification for each pixel. Despite the claim of subverting per-pixel classification, the bipartite matching-based semantic segmentation [21] still cannot avoid the per-pixel mask log-likelihood.

In this paper, we introduce a new approach, *Generative Semantic Segmentation (GSS)*, that formulates semantic segmentation as an *image-conditioned mask generation problem*. This conceptually differs from the conventional formulation of discriminative per-pixel classification learning, based on the log-likelihood of a conditional probability (i.e. the classification probability of image pixels). Taking the manner of image generation instead [14, 11], we generate the whole segmentation masks with an *auxiliary latent variable distribution* introduced. This formulation is not only simple and more task-agnostic, but also facilitates the exploitation of off-the-shelf big generative models (e.g. DALL-E [10] trained by 3 billion iterations on a 300 million open-image dataset, far beyond both the data scale and training cost of semantic segmentation).

*Li Zhang (zhangli@fudan.edu.cn) is the corresponding author with School of Data Science, Fudan University.

Vector Quantized Models for Planning

Sherjil Ozair^{1,2} Yazhe Li¹ Ali Razavi¹ Ioannis Antonoglou¹ Airon van den Oord¹ Oriol Vinyals¹

Abstract

Recent developments in the field of model-based RL have proven successful in a range of environments, especially ones where planning is essential. However, such successes have been limited to deterministic fully-observed environments. We present a new approach that handles stochastic and partially-observable environments. Our key insight is to use discrete autoencoders to capture the multiple possible effects of an action in a stochastic environment. We use a stochastic variant of Monte Carlo tree search to plan over both the agent’s actions and the discrete latent variables representing the environment’s response. Our approach significantly outperforms an offline version of MuZero on a stochastic interpretation of chess where the opponent is considered part of the environment. We also show that our approach scales to *DeepMind Lab*, a first-person 3D environment with large visual observations and partial observability.

1. Introduction

Making predictions about the world may be a necessary ingredient towards building intelligent agents, as humans use these predictions to devise and enact plans to reach complex goals (Lake et al., 2017). However, in the field of reinforcement learning (RL), a tension still exists between model-based and model-free RL. Model-based RL and planning have big ingredients in many successes such as games like (Schumacher, 1990; Silver et al., 2012a; Go Silver et al., 2016a, 2017b), and Poker (Moravcsik et al., 2017; Brown et al.). However, their applicability to richer environments with larger action and state spaces remains limited due to some of the key assumptions made in such approaches. Other notable results have not used any form of model or planning, such as playing complex video games

Dota 2 (OpenAI et al., 2016) and StarCraft II (Vinyals et al., 2019), or robotics (OpenAI et al., 2018).

In this work we are motivated by widening the applicability of model-based planning by devising a solution which removes some of the key assumptions made by the MuZero algorithm (Schrittwieser et al., 2019). Table 1 and Figure 1 summarize the key features of model-based planning algorithms discussed in this paper. MuZero lifts the crucial requirement of having access to a perfect simulator of the environment dynamics found in previous model-based planning approaches (Gilmer et al., 2017a; Anthony et al., 2017). In many cases such a simulator is not available (e.g., weather forecasting), is expensive (e.g., scientific modeling), or is cumbersome to run (e.g. for complex games such as Dota 2 or StarCraft II).

However, MuZero still makes a few limiting assumptions. It assumes the environment to be deterministic, limiting which environments can be used. It assumes full access to the state, also limiting which environments can be used. The search and planning is over future agent’s actions, which could be millions in environments with complex action spaces. The search occurs at every agent-environment interaction step, which may be too fine-grained and wasteful.

Largely inspired by both MuZero and the recent successes of VQVAE (van den Oord et al., 2017; Razavi et al., 2019) and large language models (Radford et al., 2019), we devise VQVAE for planning, which in principle can remove most of these assumptions.

Our approach uses a state VQVAE and a transition model. The state VQVAE encodes future observations into discrete latent variables. This allows the use of Monte Carlo tree search (MCTS) (Coulton, 2009) for planning not only over

Table 1. Key features of different planning algorithms.

Method	Latent Model	Agent	Stochastic	Stochastic	Temporal Abstraction
AlphaZero	✓	✓	✓	✓	✓
DeepMind MuZero	✓	✓	✓	✓	✓
Impe-rl MuZero	✓	✓	✓	✓	✓
VQVAE	✓	✓	✓	✓	✓
Other	✓	✓	✓	✓	✓

*Procedings of the 35th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

¹Google Brain, Mountain View, CA, USA. Correspondence to: Adam Roberts <adamr@google.com>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

²<https://goo.gl/maps/ta/musica-vae-code>
³<https://goo.gl/maps/ta/musica-vae-examples>

References

1. <https://towardsdatascience.com/all-you-want-to-know-about-deep-learning-8d68dcffc258>
2. Goodfellow, Ian, et al. Deep learning. Vol. 1. No. 2. Cambridge: MIT press, 2016.
3. Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." Proceedings of the 25th international conference on Machine learning. 2008.
4. Alain, Guillaume, and Yoshua Bengio. "What regularized auto-encoders learn from the data-generating distribution." The Journal of Machine Learning Research 15.1 (2014): 3563-3593.
5. Doersch, Carl. "Tutorial on variational autoencoders." arXiv preprint arXiv:1606.05908 (2016).
6. Zhou, Bolei, et al. "Learning deep features for discriminative localization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
7. <https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html#vae-variational-autoencoder>
8. <https://www.jeremyjordan.me/variational-autoencoders/>