How to write a good course project report

CSC413 Tutorial 5

Keiran Paster - February 11, 2021

Purpose

What will you learn from doing the final project?

- Gain experience working on some original research
  - Check your understanding of concepts from the course
  - Identify important issues with the field and think about how to fix them
- Gain experience writing results in a paper style format
  - Clearly express where the field was before your paper
  - Clearly express how your project has contributed to pushing the field forward

Purpose

What will you learn from doing the final project?

- Gain experience working on some original research
  - Check your understanding of concepts from the course
  - Identify important issues with the field and think about how to fix them
- **Gain experience writing results in a paper style format** ← **This tutorial**
  - Clearly express where the field was before your paper
  - Clearly express how your project has contributed to pushing the field forward

Resources

Much of this tutorial is inspired by the following resources:

- https://billf.mit.edu/sites/default/files/documents/cvprPapers.pdf
- http://www.ai.mit.edu/courses/6.899/papers/ted.htm
- http://approximatelycorrect.com/2018/01/29/heuristics-technical-scientific-writing-machine-learning-perspective/

Organization

Typical Sections

- Introduction
- Related Work
- Method
- Experiments
- Conclusion
- Future Work / Limitations

Organization

From Ted Adelson

- Start by stating which problem you are addressing, keeping the audience in mind. They must care about it, which means that sometimes you must tell them why they should care about the problem.
- Then state briefly what the other solutions are to the problem, and why they aren't satisfactory. If they were satisfactory, you wouldn’t need to do the work.
- Then explain your own solution, compare it with other solutions, and say why it’s better.
- At the end, talk about related work where similar techniques and experiments have been used, but applied to a different problem.

http://www.ai.mit.edu/courses/6.899/papers/ted.htm

Introduction

Example

Cycle-GAN

Objective:

Help the reader understand the problem and why it is important.

Figure gives visual examples of what “image translation” means

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu* Taesung Park* Phillip Isola Alexei A. Efros
Berkeley AI Research (BAIR) laboratory, UC Berkeley

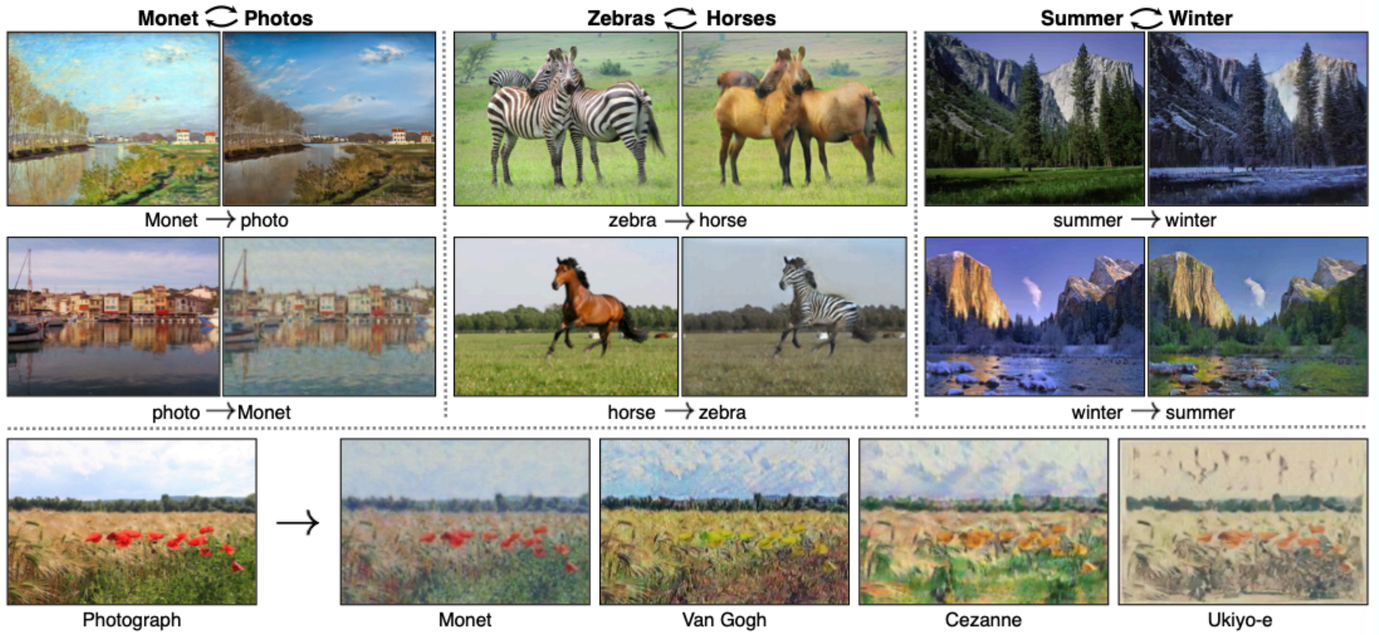

Figure 1: Given any two unordered image collections X and Y, our algorithm learns to automatically “translate” an image from one into the other and vice versa: *(left)* Monet paintings and landscape photos from Flickr; *(center)* zebras and horses from ImageNet; *(right)* summer and winter Yosemite photos from Flickr. Example application *(bottom)*: using a collection of paintings of famous artists, our method learns to render natural photographs into the respective styles.

Example

Cycle-GAN

Objective:

Help the reader understand the problem and why it is important.

Figure 1: Given any two unordered image collections X and Y, our algorithm learns to automatically "translate" an image from one into the other and vice versa: *(left)* Monet paintings and landscape photos from Flickr; *(center)* zebras and horses from ImageNet; *(right)* summer and winter Yosemite photos from Flickr. Example application *(bottom)*: using a collection of paintings of famous artists, our method learns to render natural photographs into the respective styles.

Give an example of "image translation", refer to the figure.

1. Introduction

What did Claude Monet see as he placed his easel by the bank of the Seine near Argenteuil on a lovely spring day in 1873 (Figure 1, top-left)? A color photograph, had it been invented, may have documented a crisp blue sky and a glassy river reflecting it. Monet conveyed his *impression* of this same scene through wispy brush strokes and a bright palette.

What if Monet had happened upon the little harbor in Cassis on a cool summer evening (Figure 1, bottom-left)? A brief stroll through a gallery of Monet paintings makes it possible to imagine how he would have rendered the scene: perhaps in pastel shades, with abrupt dabs of paint, and a somewhat flattened dynamic range.

We can imagine all this despite never having seen a side by side example of a Monet painting next to a photo of the scene he painted. Instead, we have knowledge of the set of Monet paintings and of the set of landscape photographs. We can reason about the stylistic differences between these two sets, and thereby imagine what a scene might look like if we were to "translate" it from one set into the other.

Example

Cycle-GAN

Objective:

Show that other solutions to the problem are unsatisfactory.

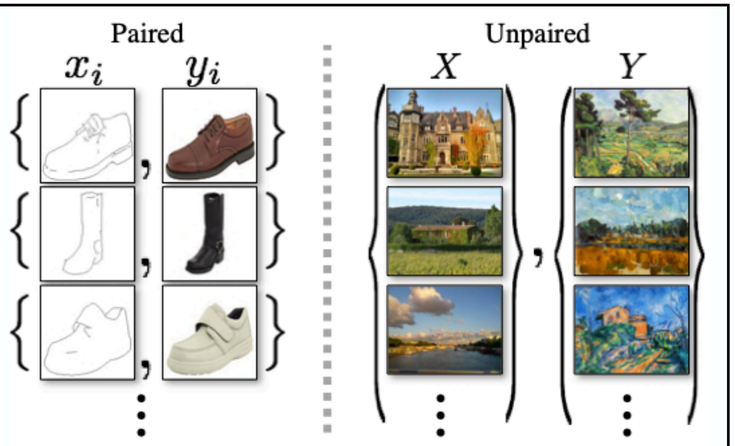

Figure 2: *Paired* training data (left) consists of training examples $\{x_i, y_i\}_{i=1}^N$, where the correspondence between x_i and y_i exists [22]. We instead consider *unpaired* training data (right), consisting of a source set $\{x_i\}_{i=1}^N$ $(x_i \in X)$ and a target set $\{y_j\}_{j=1}^M$ $(y_j \in Y)$, with no information provided as to which x_i matches which y_j.

two sets, and thereby imagine what a scene might look like if we were to "translate" it from one set into the other.

In this paper, we present a method that can learn to do the same: capturing special characteristics of one image collection and figuring out how these characteristics could be translated into the other image collection, all in the absence of any paired training examples.

This problem can be more broadly described as image-to-image translation [22], converting an image from one representation of a given scene, x, to another, y, e.g., grayscale to color, image to semantic labels, edge-map to photograph. Years of research in computer vision, image processing, computational photography, and graphics have produced powerful translation systems in the supervised

Define "image-to-image translation" explicitly

"Other solutions to the problem are unsatisfactory" (and why)

setting, where example image pairs $\{x_i, y_i\}_{i=1}^N$ are available (Figure 2, left), e.g., [11, 19, 22, 23, 28, 33, 45, 56, 58, 62]. However, obtaining paired training data can be difficult and expensive. For example, only a couple of datasets exist for tasks like semantic segmentation (e.g., [4]), and they are relatively small. Obtaining input-output pairs for graphics tasks like artistic stylization can be even more difficult since the desired output is highly complex, typically requiring artistic authoring. For many tasks, like object transfiguration (e.g., zebra↔horse, Figure 1 top-middle), the desired output is not even well-defined.

We therefore seek an algorithm that can learn to translate between domains without paired input-output examples (Figure 2, right). We assume there is some underlying relationship between the domains – for example, that they are two different renderings of the same underlying scene – and seek to learn that relationship. Although we lack supervision in the form of paired examples, we can exploit supervision at the level of sets: we are given one set of images in domain X and a different set in domain Y. We may train

Example

Cycle-GAN

Objective:

Show the new solution to the problem and say why its better

Another unsatisfactory solution

Their solution and why it's better

Experimental results

a mapping $G : X \rightarrow Y$ such that the output $\hat{y} = G(x)$, $x \in X$, is indistinguishable from images $y \in Y$ by an adversary trained to classify $\hat{y}$ apart from y. In theory, this objective can induce an output distribution over $\hat{y}$ that matches the empirical distribution $p_{data}(y)$ (in general, this requires G to be stochastic) [16]. The optimal G thereby translates the domain X to a domain $\hat{Y}$ distributed identically to Y. However, such a translation does not guarantee that an individual input x and output y are paired up in a meaningful way – there are infinitely many mappings G that will induce the same distribution over $\hat{y}$. Moreover, in practice, we have found it difficult to optimize the adversarial objective in isolation: standard procedures often lead to the well-known problem of mode collapse, where all input images map to the same output image and the optimization fails to make progress [15].

These issues call for adding more structure to our objective. Therefore, we exploit the property that translation should be "cycle consistent", in the sense that if we translate, e.g., a sentence from English to French, and then translate it back from French to English, we should arrive back at the original sentence [3]. Mathematically, if we have a translator $G : X \rightarrow Y$ and another translator $F : Y \rightarrow X$, then G and F should be inverses of each other, and both mappings should be bijections. We apply this structural assumption by training both the mapping G and F simultaneously, and adding a *cycle consistency loss* [64] that encourages $F(G(x)) \approx x$ and $G(F(y)) \approx y$. Combining this loss with adversarial losses on domains X and Y yields our full objective for unpaired image-to-image translation.

We apply our method to a wide range of applications, including collection style transfer, object transfiguration, season transfer and photo enhancement. We also compare against previous approaches that rely either on hand-defined factorizations of style and content, or on shared embedding functions, and show that our method outperforms these baselines. We provide both PyTorch and Torch implementations. Check out more results at our website.

Example

BigGAN

Figure gives visual examples of what “high fidelity natural image synthesis” means

Published as a conference paper at ICLR 2019

LARGE SCALE GAN TRAINING FOR HIGH FIDELITY NATURAL IMAGE SYNTHESIS

Andrew Brock[*†]
Heriot-Watt University
ajb5@hw.ac.uk

Jeff Donahue[†]
DeepMind
jeffdonahue@google.com

Karen Simonyan[†]
DeepMind
simonyan@google.com

ABSTRACT

Despite recent progress in generative image modeling, successfully generating high-resolution, diverse samples from complex datasets such as ImageNet remains an elusive goal. To this end, we train Generative Adversarial Networks at the largest scale yet attempted, and study the instabilities specific to such scale. We find that applying orthogonal regularization to the generator renders it amenable to a simple “truncation trick,” allowing fine control over the trade-off between sample fidelity and variety by reducing the variance of the Generator’s input. Our modifications lead to models which set the new state of the art in class-conditional image synthesis. When trained on ImageNet at 128×128 resolution, our models (BigGANs) achieve an Inception Score (IS) of 166.5 and Fréchet Inception Distance (FID) of 7.4, improving over the previous best IS of 52.52 and FID of 18.65.

1 INTRODUCTION

Figure 1: Class-conditional samples generated by our model.

Example

BigGAN

The problem is interesting because there is a large gap in the fidelity of synthetic and natural images.

Bullet points showing the main contributions of the paper

Experimental results

The state of generative image modeling has advanced dramatically in recent years, with Generative Adversarial Networks (GANs, Goodfellow et al. (2014)) at the forefront of efforts to generate high-fidelity, diverse images with models learned directly from data. GAN training is dynamic, and sensitive to nearly every aspect of its setup (from optimization parameters to model architecture), but a torrent of research has yielded empirical and theoretical insights enabling stable training in a variety of settings. Despite this progress, the current state of the art in conditional ImageNet modeling (Zhang et al., 2018) achieves an Inception Score (Salimans et al., 2016) of 52.5, compared to 233 for real data.

In this work, we set out to close the gap in fidelity and variety between images generated by GANs and real-world images from the ImageNet dataset. We make the following three contributions towards this goal:

- We demonstrate that GANs benefit dramatically from scaling, and train models with two to four times as many parameters and eight times the batch size compared to prior art. We introduce two simple, general architectural changes that improve scalability, and modify a regularization scheme to improve conditioning, demonstrably boosting performance.

[*] Work done at DeepMind
[†] Equal contribution

- As a side effect of our modifications, our models become amenable to the "truncation trick," a simple sampling technique that allows explicit, fine-grained control of the trade-off between sample variety and fidelity.
- We discover instabilities specific to large scale GANs, and characterize them empirically. Leveraging insights from this analysis, we demonstrate that a combination of novel and existing techniques can reduce these instabilities, but complete training stability can only be achieved at a dramatic cost to performance.

Our modifications substantially improve class-conditional GANs. When trained on ImageNet at 128×128 resolution, our models (BigGANs) improve the state-of-the-art Inception Score (IS) and Fréchet Inception Distance (FID) from 52.52 and 18.65 to 166.5 and 7.4 respectively. We also successfully train BigGANs on ImageNet at 256×256 and 512×512 resolution, and achieve IS and FID of 232.5 and 8.1 at 256×256 and IS and FID of 241.5 and 11.5 at 512×512. Finally, we train our models on an even larger dataset – JFT-300M – and demonstrate that our design choices transfer well from ImageNet. Code and weights for our pretrained generators are publicly available [1].

Introduction

Summary

- State the problem and context. Why is this problem important?
- Why are other solutions unsatisfactory? What is your solution? What are the implications of your solution?
- What is new in your paper?
  - Bullet points for the main contributions can be really helpful for the reader!

Related Work

Related Work

Motivation

- Does the reader have a background in the subject of the paper?
  - No? Then this is a good place to provide some references to important prior works, explain the history of solutions to the problem, and place your paper's contributions in context.
  - Yes? The reader will commonly wonder how the method differs from an already published prior method. This is a good place to show that you understand the subject and have read the relevant works.

https://beehive.cs.princeton.edu/wiki/writing/how-to-write-a-related-work-section/

Related Work

Tips

- There should be a common theme to each paragraph in this section.
- Be extremely generous in including related work. Be intellectually honest.

https://beehive.cs.princeton.edu/wiki/writing/how-to-write-a-related-work-section/

Example

Cycle-GAN

- Paragraphs are organized by high-level topic.
- The authors demonstrate that they are aware of a lot of related work.

images cannot be distinguished from images in the target domain.

Image-to-Image Translation The idea of image-to-image translation goes back at least to Hertzmann et al.'s Image Analogies [19], who employ a non-parametric texture model [10] on a single input-output training image pair. More recent approaches use a *dataset* of input-output examples to learn a parametric translation function using CNNs (e.g., [33]). Our approach builds on the "pix2pix" framework of Isola et al. [22], which uses a conditional generative adversarial network [16] to learn a mapping from input to output images. Similar ideas have been applied to various tasks such as generating photographs from sketches [44] or from attribute and semantic layouts [25]. However, unlike the above prior work, we learn the mapping without paired training examples.

Unpaired Image-to-Image Translation Several other methods also tackle the unpaired setting, where the goal is to relate two data domains: X and Y. Rosales et al. [42] propose a Bayesian framework that includes a prior based on a patch-based Markov random field computed from a source image and a likelihood term obtained from multiple style images. More recently, CoGAN [32] and cross-modal scene networks [1] use a weight-sharing strategy to learn a common representation across domains. Concurrent to our method, Liu et al. [31] extends the above framework with a combination of variational autoencoders [27] and generative adversarial networks [16]. Another line of concurrent work [46, 49, 2] encourages the input and output to share specific "content" features even though they may differ in "style". These methods also use adversarial networks, with additional terms to enforce the output to be close to the input in a predefined metric space, such as class label space [2], image pixel space [46], and image feature space [49].

Unlike the above approaches, our formulation does not rely on any task-specific, predefined similarity function between the input and output, nor do we assume that the input and output have to lie in the same low-dimensional embedding space. This makes our method a general-purpose solution for many vision and graphics tasks. We directly compare against several prior and contemporary approaches in Section 5.1.

Cycle Consistency The idea of using transitivity as a way to regularize structured data has a long history. In visual tracking, enforcing simple forward-backward consistency has been a standard trick for decades [24, 48]. In the language domain, verifying and improving translations via "back translation and reconciliation" is a technique used by human translators [3] (including, humorously, by Mark Twain [51]), as well as by machines [17]. More recently, higher-order cycle consistency has been used in structure from motion [61], 3D shape matching [21], cosegmentation [55], dense semantic alignment [65, 64], and depth estimation [14]. Of these, Zhou et al. [64] and Godard et al. [14] are most similar to our work, as they use a *cycle consistency loss* as a way of using transitivity to supervise CNN training. In this work, we are introducing a similar loss to push G and F to be consistent with each other. Concurrent with our work, in these same proceedings, Yi et al. [59] independently use a similar objective for unpaired image-to-image translation, inspired by dual learning in machine translation [17].

Neural Style Transfer [13, 23, 52, 12] is another way to perform image-to-image translation, which synthesizes a novel image by combining the content of one image with the style of another image (typically a painting) based on matching the Gram matrix statistics of pre-trained deep features. Our primary focus, on the other hand, is learning the mapping between two image collections, rather than between two specific images, by trying to capture correspondences between higher-level appearance structures. Therefore, our method can be applied to other tasks, such as

Method

Method

Motivation

- Present the main (non-experimental) results.
- Show in detail what the method is and why it is (theoretically) justified.

Method

Tips

- Include an algorithm box, equations describing your model, theorems or formally stated conjectures.
- This section is fairly easy to write in my experience.
    - Just describe what you did and why! Ideally you should already have a good idea of what to write here by the time it’s time to write the report.

Example

Cycle-GAN

- The authors describe their new loss in detail.

3. Formulation

Our goal is to learn mapping functions between two domains X and Y given training samples $\{x_i\}_{i=1}^N$ where $x_i \in X$ and $\{y_j\}_{j=1}^M$ where $y_j \in Y$[1]. We denote the data distribution as $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$. As illustrated in Figure 3 (a), our model includes two mappings $G : X \rightarrow Y$ and $F : Y \rightarrow X$. In addition, we introduce two adversarial discriminators D_X and D_Y, where D_X aims to distinguish between images $\{x\}$ and translated images $\{F(y)\}$; in the same way, D_Y aims to discriminate between $\{y\}$ and $\{G(x)\}$. Our objective contains two types of terms: *adversarial losses* [16] for matching the distribution of generated images to the data distribution in the target domain; and *cycle consistency losses* to prevent the learned mappings G and F from contradicting each other.

3.1. Adversarial Loss

We apply adversarial losses [16] to both mapping functions. For the mapping function $G : X \rightarrow Y$ and its discriminator D_Y, we express the objective as:

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))], \quad (1)$$

where G tries to generate images $G(x)$ that look similar to images from domain Y, while D_Y aims to distinguish between translated samples $G(x)$ and real samples y. G aims to minimize this objective against an adversary D that tries to maximize it, i.e., $\min_G \max_{D_Y} \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y)$. We introduce a similar adversarial loss for the mapping function $F : Y \rightarrow X$ and its discriminator D_X as well: i.e., $\min_F \max_{D_X} \mathcal{L}_{\text{GAN}}(F, D_X, Y, X)$.

3.2. Cycle Consistency Loss

Adversarial training can, in theory, learn mappings G and F that produce outputs identically distributed as target domains Y and X respectively (strictly speaking, this requires G and F to be stochastic functions) [15]. However, with large enough capacity, a network can map the same set of input images to any random permutation of images in the target domain, where any of the learned mappings can induce an output distribution that matches the target distribution. Thus, adversarial losses alone cannot guarantee that the learned function can map an individual input x_i to a desired output y_i. To further reduce the space of possible mapping functions, we argue that the learned mapping functions should be cycle-consistent: as shown in Figure 3 (b), for each image x from domain X, the image translation cycle should be able to bring x back to the original image, i.e., $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$. We call this *forward cycle consistency*. Similarly, as illustrated in Figure 3 (c), for each image y from domain Y, G and F should also satisfy *backward cycle consistency*: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$. We incentivize this behavior using a *cycle consistency loss*:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]. \quad (2)$$

In preliminary experiments, we also tried replacing the L1 norm in this loss with an adversarial loss between $F(G(x))$ and x, and between $G(F(y))$ and y, but did not observe improved performance.

The behavior induced by the cycle consistency loss can be observed in Figure 4: the reconstructed images $F(G(x))$ end up matching closely to the input images x.

Figure 4: The input images x, output images $G(x)$ and the reconstructed images $F(G(x))$ from various experiments. From top to bottom: photo↔Cezanne, horses↔zebras, winter→summer Yosemite, aerial photos↔Google maps.

3.3. Full Objective

Our full objective is:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F), \quad (3)$$

[1] We often omit the subscript i and j for simplicity.

Example

CPC

- Walk through the theory of the paper as clearly as possible.

2.2 Contrastive Predictive Coding

Figure 1 shows the architecture of Contrastive Predictive Coding models. First, a non-linear encoder g_{enc} maps the input sequence of observations x_t to a sequence of latent representations $z_t = g_{\text{enc}}(x_t)$, potentially with a lower temporal resolution. Next, an autoregressive model g_{ar} summarizes all $z_{\leq t}$ in the latent space and produces a context latent representation $c_t = g_{\text{ar}}(z_{\leq t})$.

As argued in the previous section we do not predict future observations x_{t+k} directly with a generative model $p_k(x_{t+k}|c_t)$. Instead we model a density ratio which preserves the mutual information between x_{t+k} and c_t (Equation 1) as follows (see next sub-section for further details):

$$f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})} \tag{2}$$

where $\propto$ stands for 'proportional to' (i.e. up to a multiplicative constant). Note that the density ratio f can be unnormalized (does not have to integrate to 1). Although any positive real score can be used here, we use a simple log-bilinear model:

$$f_k(x_{t+k}, c_t) = \exp\left(z_{t+k}^T W_k c_t\right), \tag{3}$$

In our experiments a linear transformation $W_k^T c_t$ is used for the prediction with a different W_k for every step k. Alternatively, non-linear networks or recurrent neural networks could be used.

By using a density ratio $f(x_{t+k}, c_t)$ and inferring z_{t+k} with an encoder, we relieve the model from modeling the high dimensional distribution x_{t_k}. Although we cannot evaluate $p(x)$ or $p(x|c)$ directly, we can use samples from these distributions, allowing us to use techniques such as Noise-Contrastive Estimation [12, 14, 15] and Importance Sampling [16] that are based on comparing the target value with randomly sampled negative values.

In the proposed model, either of z_t and c_t could be used as representation for downstream tasks. The autoregressive model output c_t can be used if extra context from the past is useful. One such example is speech recognition, where the receptive field of z_t might not contain enough information to capture phonetic content. In other cases, where no additional context is required, z_t might instead be better. If the downstream task requires one representation for the whole sequence, as in e.g. image classification, one can pool the representations from either z_t or c_t over all locations.

Finally, note that any type of encoder and autoregressive model can be used in the proposed framework. For simplicity we opted for standard architectures such as strided convolutional layers with resnet blocks for the encoder, and GRUs [17] for the autoregresssive model. More recent advancements in autoregressive modeling such as masked convolutional architectures [18, 19] or self-attention networks [20] could help improve results further.

2.3 InfoNCE Loss and Mutual Information Estimation

Both the encoder and autoregressive model are trained to jointly optimize a loss based on NCE, which we will call InfoNCE. Given a set $X = \{x_1, \dots x_N\}$ of N random samples containing one positive sample from $p(x_{t+k}|c_t)$ and $N-1$ negative samples from the 'proposal' distribution $p(x_{t+k})$, we optimize:

$$\mathcal{L}_{\text{N}} = -\underset{X}{\mathbb{E}}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right] \tag{4}$$

Experiments

Experiments

Tips

- Usually you will want to convince the reader that your approach is useful in the real world.
- State hypotheses and motivation for each experiment.
  - Think about whether you have clear evidence to verify or nullify the hypothesis.
- Don’t cherry pick results!
  - Including negative results is fine, especially for this project report.

Experiments

Tips

- You will be comparing the performance of your algorithm with a prior approach.
  - Describe the implementation of the prior approach. Did you use existing code or reimplement it yourself?
- Describe the metrics you are using to define success.

Experiments

Reproducibility

- Your report should provide enough information to reproduce results.
  - Architecture, optimizer details, hyperparameters (and how you found those hyperparameters)
  - What tricks were useful to make your approach work?
  - What were the most important parameters to tune?
  - How much compute was used?

How to Read a Paper

How to Read a Paper

- Most people will look at:
  - abstract
  - first paragraph of the intro to understand what problem the paper is trying to solve
  - last paragraph (bullet points) of the intro for contributions of the paper
  - interesting figures and their captions

Inspired by: https://twitter.com/hardmaru/status/1305758751798910976

Figures

- A reader should be able to understand your paper without looking at any figures
  - Key details in the paper should not be hidden within figures
- A reader should be able to understand your paper only looking at the figures
  - Figures stand out and readers should be able to get a general idea about the paper just from looking at them.
  - When writing captions, keep in mind that some people may not have read the text that refers to the figure. Include enough details to explain the main idea of the figure.

References

References

Tips

- Use dblp.org instead of arxiv to get bibtex entries.
- Cite generously
  - Make sure to cite sources wherever you use them (not just the first time; not just in the related works section)
- Use \citet{} for a textual citation, \citep{} for parenthetical.

While forward dynamics models map a state and action to the next state, an inverse dynamics model maps two subsequent states to an action. Inverse dynamics models have been used in various ways in sequential decision making. In exploration, inverse dynamics serves as a way to learn representations of the controllable aspects of the state (Pathak et al., 2017). In imitation learning, inverse dynamics models can be used to map a sequence of states to the actions needed to imitate the trajectory (Pavse et al., 2019). Christiano et al. (2016) use inverse dynamics models to translate actions taken in a simulated environment to the real world.

How to get a good score

Rubric

- Quality [35%] (do good technical work)
- Clarity [25%] (follow the writing guidelines in this tutorial)
- Originality [20%] (is what you are doing novel?)
- Significance [5%] (are the results important?)
- Participation [15%] (write reviews of other students' projects)

Reviewing

How to Write a Review

From the ICLR 2021 Reviewer Guide

- Read the paper carefully, considering the following:
  - What is the goal of the paper?
  - What did the authors do well?
  - What are the weaknesses of the paper?
  - Keep in mind that even if a paper isn’t interesting to you, it may be interesting to others.

How to Write a Review

From the ICLR 2021 Reviewer Guide

- Answer these key questions:
  - What is specifically being tackled by the paper?
  - Is the approach well motivated?
  - Does the paper support the claims it makes?

How to Write a Review

From the ICLR 2021 Reviewer Guide

- To write the review:
  - Summarize what the paper contributes.
  - List strong and weak points.
  - State the score you are giving the paper.
  - Provide support for your score.
  - Write what would be needed to improve the score.

Summary

Summary

How to write a good course project report

- Write a good intro, clearly introducing the problem, why it is important and unsolved, and the contributions of your paper.
- Cite generously.
- Clearly state hypotheses and whether the evidence verifies or nullifies them.

Good Luck!