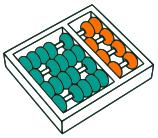


Strenghts and weaknesses of unsupervised learning methods

Lesson No. 10

Rosa Yuliana Gabriela Paccotacya Yanque - RA:263068

1 Introduction

In 2014, Geoffrey Hinton on an AMA(Ask me Anything) session on Reddit stated:

" 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."*

Since then, he motivated unsupervised learning in order to be able to learn from as much data as possible to the kind of capacity that neural networks allow to do. Currently, the only limitation to train larger models is the kind of computing, even though, ImageNet seems big and now can be trained to 75% accuracy in around 70 seconds, in the future we will be using larger models training in larger data sets. Therefore, here we will summarize the strenghts and weaknesses of unsupervised learning methods presented in [1].

2 Autoregressive Models

Autoregressive Models are based on Bayes Nets(Directed Acyclic Graphs that defines a dependency/conditions between random variables) and allows to work in a feed-forward way to predict its future from past values. Some models examples are MADE[2], Pixel RNN/CNN[3], Subscale Pixel Network[4], Hierarchical Autoregressive Model[5] that generate new images with an impressive level of details. These models were also extended to work in text to speech problems resulting in WaveNet[6] that is actually now deployed on Google Assistant. It was also applied in videos to synthesize new frames[7] and in language, Open AI GPT-2[8] predicts the next word, given all of the previous words within some text in this way complete and create a coherent new story given a beginning.

2.1 Causes for the big progress till the moment

- Traininig of the model with larger batch sizes
- Architecture with more hidden units and layers and clever ways to condition on auxiliary variables
- Preprocessing, e.g. GPT-2 uses byte pair encoding.
- Use of compute power (TPU) and several days / weeks of training to see good results
- Fewer assumptions e.g. in PixelRNN-CNN modeled pixels as independent categories(256 softmax)
- Architectural advances : Masked / Causal Convolutions(parallel architectural convolutions where future input depends only in the past), Dilated Convolutions (that gives a larger receptive field than regular convolutions), Transformers(allows to have a global context all over the model instead of relying on memory).
- Loss functions, researchers have used cross-entropy loss in order to make the whole architecture model look like supervised learning.

2.2 Possible future works

- Advances with model-parallelism, Mesh-TensorFlow is a new feature that allows to share the model parameters across TPU cores. This will enable to train around trillion parameter language models trained on all the Internet's text.

Figure 1. Random Glow model samples Source: Glow[11]

- Same model for both text and pixels (image / video). Share self-attention blocks and compress both types of data with separate blocks of encoders and decoders.
- To reduce time in training and sampling, find new sparse kernels that allows fast sampling with better low-level core engineering
- Hybrid models with weaker autoregressive structure but trained on a larger scale
- New creative architecture design choices as it was self-attention that reduced the needed computation per parameter introduced.
- Use larger models and datasets

2.3 Drawbacks

- There is no single layer of learned representation to use in another downstream tasks(because since the beginning it is conditioned on the task), it is better to fine-tune the entire model.
- Currently, sampling time is slow for practical deployment.
- No interpolations across samples. Flow Models and GANs do allow to interpolate.

3 Flow Models

Flow Models are likelihood models, able to learn the probability density function of real data by a chain of invertible transformations(inference), in this way they are able to compress information without loss. Some models examples are Nice[9], RealNVP[10], Glow[11], Flow++[12].

3.1 Possible future works

- Learning the mask for coupling and close the gap with autoregressive models (hybrid flows)
- Experiments to evaluate wheter it is better to have Fewer expressive(lot of layer per flow) flows or Several shallow(few layers) flows.
- Measure the impact of Usage of Multiscale Loss (bits/dim vs sample quality tradeoffs)
- Check if variables obatined in flows can be used in other tasks (Representation Learning) e.g. classification
- Different types of initializating the flows
- Try to get Glow-level samples(Figure 1) with fewer parameters, they use 615 million parameters.
- Dimension reduction, new design of architecture(Conditional flow model) always verifying to have a stable training.

3.2 Drawbacks

- It is a early field, so there is a long way to get GAN level samples or autoregressive model-level likelihood scores and samples.
- Models are really big

4 Latent Variable Models

Started with Auto-Encoding Variational Bayes [13], then continued with Variational Autoencoders such as Pixel-VAE [14] and BIVA [15]. One of the most-know application is Generative Query Networks [16] that given different cameras pauses decodes for different pauses and ends up with a 3D-mapping.

4.1 Advantages

- It works like a “compressed” representation learning (Lower dimensions) and gets the approximate log-likelihood
- Interpolations between latent variables to what the model learns
- If the coefficient of the KL term is increased, Disentangled representations can be obtained
- It behaves as a Generative Model ,Approximate Density Model, Latent Variables, Dimensionality Reduction

4.2 Disadvantages

- Most of the times gives blurry samples, therefore large scale VAEs is still on-going work
- Disentanglement with the KL term just worked on toy domains
- If just one task (learn better representations or to get better samples or get better density estimates) is needed, it is better to use another method.

4.3 Possible future works

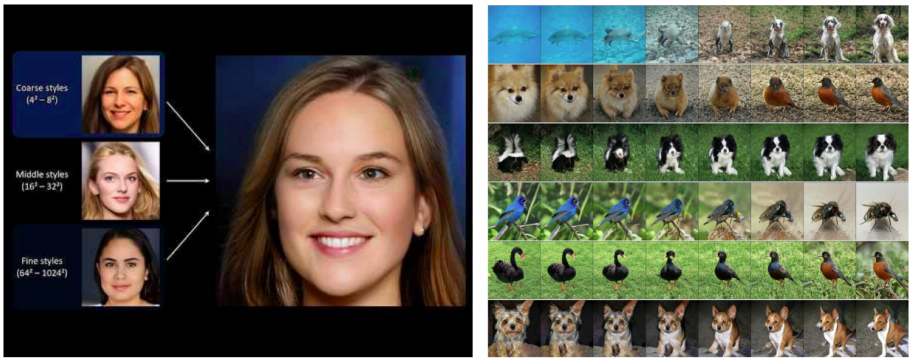
- Try to use more powerful posteriors, hierarchical latent variable models in order to learn coarse-fine features and interpolations
- Experiment with discrete latent variable models to prevent posterior collapse
- Scale at the level of Flow Models training, make decoders less-autoregressive

5 Implicit Models: Generative Adversarial Networks GAN

GAN’s consist of a discriminator that distinguish between real or fake and a generator that create samples from the distribution of data and use the discriminator’s feedback to "cheat" it better in each training step. Some recent GAN’s(Figure 2) are StyleGAN [17] that mixtures styles to create realistic people and BigGAN [18] that interpolates classes.

5.1 Possible future works

- Handle more fine grained details like more complex scenes in videos.
- Experiment with Lipschitzness constraints, Feeding in noise at different levels, batch / instance normalization
- Make changes in the architecture design, use objective functions (Hinge Loss)
- Look for stability and scalability that is Deeper(more layers) models with fewer parameters and larger batch sizes



StyleGAN BigGAN

Figure 2. Example of different GANs

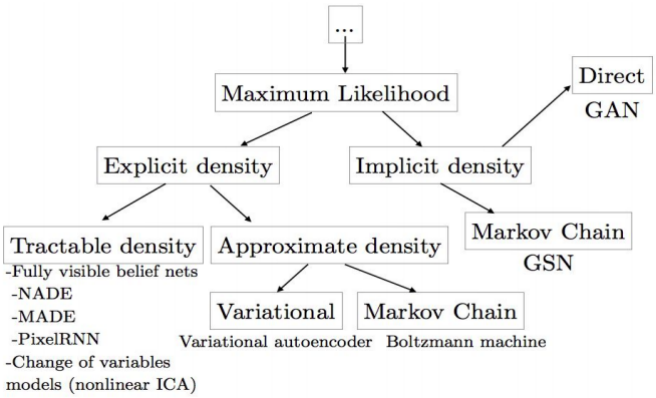


Figure 3. taxonomy of Generative Models Source: Glow[1]

5.2 Drawbacks

- Enginnering in the code is really complex and test changes in large scale datasets is time-consuming
- Unconditional GANs is still a new topic, so sample diversity is still missing
- Evaluation metrics to account for generalization are changing

6 GANs or Density Models?

- It is not true that engineering in the code is less in Density Models than GAN's, it is not true that there's no theory for GANs either
- Blurry samples versus having Mode Collapse? The direction of KL between the real data distribution and the generated distribution decides to take into account. If data distribution is in the beginning, compression is obtained. But if generated distribution is in the beginning, the sample quality is better,
- GANs are more popular not only because of sample quality and interpolations but because they are not as big as Density models so its training can be done in a small GPU instead of needing distributed computation.

Figure 3 shows a taxonomy for Generative Models, Autoregressive and Flow Models fit in tractable density category.

7 Current State of Self-Supervised Learning

It has shown promising progress in closing the gap with respect to pure supervised learning on Imagenet, Librispeech (CPCv2, MoCo, SimCLR, MoCo v2). It has allowed Transfer Learning in Language and Vision. It is hope for building general intelligence because of its utility in tasks such as transfer learning, learning reusable concepts / abstractions, building plans and imaginations, reasoning using constructed plans, etc.

References

[1] *Week 8 (parts a, b) cs294-158 deep unsupervised learning (4/3/19) - youtube*, https://www.youtube.com/watch?v=7o9dT6puHHg&feature=youtu.be, (Accessed on 07/08/2020).

[2] M. Germain, K. Gregor, I. Murray, and H. Larochelle, *Made: Masked autoencoder for distribution estimation*, 2015. arXiv: 1502.03509 [cs.LG].

[3] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, *Pixel recurrent neural networks*, 2016. arXiv: 1601.06759 [cs.CV].

[4] J. Menick and N. Kalchbrenner, *Generating high fidelity images with subscale pixel networks and multidimensional upscaling*, 2018. arXiv: 1812.01608 [cs.CV].

[5] J. D. Fauw, S. Dieleman, and K. Simonyan, *Hierarchical autoregressive image models with auxiliary decoders*, 2019. arXiv: 1903.04933 [cs.CV].

[6] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, *Wavenet: A generative model for raw audio*, 2016. arXiv: 1609.03499 [cs.SD].

[7] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, *Video pixel networks*, 2016. arXiv: 1610.00527 [cs.CV].

[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[9] L. Dinh, D. Krueger, and Y. Bengio, *Nice: Non-linear independent components estimation*, 2014. arXiv: 1410.8516 [cs.LG].

[10] L. Dinh, J. Sohl-Dickstein, and S. Bengio, *Density estimation using real nvp*, 2016. arXiv: 1605.08803 [cs.LG].

[11] D. P. Kingma and P. Dhariwal, *Glow: Generative flow with invertible 1x1 convolutions*, 2018. arXiv: 1807.03039 [stat.ML].

[12] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, *Flow++: Improving flow-based generative models with variational dequantization and architecture design*, 2019. arXiv: 1902.00275 [cs.LG].

[13] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 2013. arXiv: 1312.6114 [stat.ML].

[14] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville, *Pixelvae: A latent variable model for natural images*, 2016. arXiv: 1611.05013 [cs.LG].

[15] L. Maaløe, M. Fraccaro, V. Liévin, and O. Winther, *Biva: A very deep hierarchy of latent variables for generative modeling*, 2019. arXiv: 1902.02102 [stat.ML].

[16] A. Kumar, S. M. A. Eslami, D. J. Rezende, M. Garnelo, F. Viola, E. Lockhart, and M. Shanahan, *Consistent generative query networks*, 2018. arXiv: 1807.02033 [cs.CV].

[17] T. Karras, S. Laine, and T. Aila, *A style-based generator architecture for generative adversarial networks*, 2018. arXiv: 1812.04948 [cs.NE].

[18] A. Brock, J. Donahue, and K. Simonyan, *Large scale gan training for high fidelity natural image synthesis*, 2018. arXiv: 1809.11096 [cs.LG].