Generative models in creative practice

Sandor Madacsi

04th February 2022

1 Abstract

Generative Deep learning experiment focusing on Neural Network Architectures considering all the ethical, philosophical and resource related challenges these techniques pose. The aim is to explore the potential of Variational Auto Encoders and Generative Adversarial Networks and their use in conventional creative art practice.

Keywords: Deep learning, Generative modelling, Creative Practice, Experimentation

Repository: https://gitlab.doc.gold.ac.uk/smada001/finalproject

2 Introduction

People at both end of the spectrum have preconceptions of what machine learning can accomplish or how it accomplishes it. Whether an experienced scientist or every day's layman who only occasionally reads about its advances it can still surprise people. Fortunately, the current interest in Machine Learning specifically Deep Learning haven't hit a plateau yet, the community is growing rapidly.

As an Artist I find it important to be open minded and not only stick with only conventional techniques to produce art. Inspiration or the thought of a blank canvas is a real issue. It might be that our perception of art or our World for that matter is limiting us. Maybe machines can help with that.

There are fundamental differences between our human perception and the machine's. Machines perform astonishingly well at many fields that are unimaginably complex to a human such as vision or speech. They aren't close to human level of intelligence yet but there is a potential to do more with them by utilizing commonly available technologies more efficiently.

The project will focus on understanding the fundamentals of two generative modelling techniques, namely Variational Auto Encoders and Generative Adversarial Networks. Each model architecture has its own benefits and limitations. Through experimentation this project will explore various model architectures on gradually more complex datasets. As the computational resources are quite finite, careful considerations are placed to optimize the learning of models.

The other aim is to explore the latent space in a manner that can spark creativity for the artist and combine ideas or create new ones from scratch.

3 Background

Generative modelling is something that could very well be utilized by traditional artists but in order to unlock the full potential of these techniques it is necessary to dive deeper. For the purpose of learning the fundamentals it was crucial to use practical resources to learn from.

Deep Learning with python by Francois Chollet served as a foundation for both deep learning knowledge and the use of Tensorflow. Chollet used python scripts in order to make the theory more intuitive and easier to grasp. The book emphasizes on the benefits of using Keras on top of Tensorflow as it effortless to customize any model to fit a particular problem. [1]

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow started from the more traditional machine learning techniques before deep learning.[15]

A huge variety of problems included in these books with proper explanation provided clear understanding of key concept that contribute to developing generative models e.g.:

- pre-processing data
- building different model architectures
- how does the model learn trough back propagation
- reduce overfitting
- defining problems as linear regression, classification

Inspiration

There are many pioneers in the field of AI art who already have substantial body of work in AI. A not extensive list of a few inspirations:

Mario Klingemann's [7] installation called *"Memories of Passersby I"* [8] is a deep neural network that constantly generating portraits resembling never seen male and female features in real time.

Robbie Barrat's [9] early landscape and nude portraits [10] articulate very well the aspect of GANs that is so satisfying.

Scott Eaton's [11] AI generated bronze sculptures called *"Human Allocation of Space"* [12] translate AI in the physical domain.

Sofia Crespo's [13] project "Artificial Remnants 2.1" [14] is an exciting form of combination between nature with AI.

Core Ideas

In the field of creative AI, it is successful and popular to sample from latent space for image generation.[1] Namely two techniques stand out:

Variational Auto Encoders

Essentially the technique consists of an encoder model which have the original dataset as input and a decoder which makes the output. There is an element in the middle called latent space that connects these models. It is a low dimensional vector that maps every pixel from the input into a value in latent space. The data is well structured as it is compressed in a low dimensional vector. The decoder samples from it and recreates an image of the original dimensions best resembling the input. In this architecture there is no use of labels as the task is not to classify or predict a certain output more like replicate the input in another interpretation. With this technique we can utilize concept vectors meaning different points in latent space contain interesting variations that could be combined.[1] [2]

Generative Adversarial Networks

A generator and a discriminator model go head-to-head in order to best each other and improve. The generator samples from a normal distribution of noise in latent space. The discriminator must evaluate on real and false data (made by the generator) and define a loss. Once the discriminator made a prediction the generator tries to improve based on the feedback from the discriminator. This loops until both players are good enough that they can't improve further It's called a Nash equilibrium or a zero-sum game. This is hardly achieved even with the state-of-the-art models that we have today. This also suggest another problem that there isn't any reliable metric that we can use as feedback on performance. We can only rely on our perceptions of the output images if they are convincing enough to a human eye. [1] [2] [5]

Issues

GANs are notoriously hard to train [1] but with a few hacks we could improve the performance of our models. Although these aren't backed by science, they proved to be useful.[3] In later experiments on more complex data, these hacks could be handy such as introducing random noise to the labels to stabilise the discriminator or using convolutional layers. Based on our experiments convolutional layers are more effective than dense layers.

There are a few other aspects of this field that is debatable. It's up to interpretation whether the human does the work or the instrument (computer). *"If you hear somebody playing a piano would you ever say 'is the piano the artist?' - no "*As Mario Klingemann [7] points it out the computer serves as an instrument to the artist and it is a constant feedback loop.

Another important issue is data but most importantly the source. Data is one of the most crucial aspects in machine learning in forms of samples scraped from the internet or algorithms and models developed by others and distributed as open source. In the community it is hard to determine what's regarded as original. In the case of “Portrait of Edmond de Belamy” art project it wasn’t too obvious. The model of the GAN art piece made by three French students turned out to be mostly developed by some else namely Robbie Barrat [9] who is known as a GAN artist with many projects under his belt published on GitHub. Without openly crediting Barrat’s work and influence on the project from the beginning sprang huge debate. In terms of the project, the least I can do is properly credit those who worked hard on a problem that can later be incorporated in the development, to fit the purpose or to help.[4]

The use of resources also needs to be considered. As more complicated the model gets, it takes up more resources. It is beneficial to make a ballpark estimation on how much computational power would a particular model require as it can speed up the development process. The model architecture contains useful information on how much data passes through the model in bytes. Based on the summary of the models we can make an estimation on how much GPU memory is needed. Given our GPU and its memory capacity we can tweak the hyper parameters such as the batch size or image resolution.[6]

In all it is an interesting new technology that poses many philosophical and ethical questions, and the answers aren’t straight forward. The artist’s role is to find the right balance and try to stay true to themselves and to others.

4 Methods

The project is developed through various experiments in **Jupyter Notebooks** and concluded in images that are the results of different models. The use of notebooks makes the development smooth and manageable in long term. Code blocks can easily be transferred between different experiments. Modularity is key as models that work well on a particular dataset could be copied into another notebook and modified to be utilized on a different dataset. It is also possible to document during development with the addition of markdown blocks that are a more readable way of annotating the code than comments.

The main framework is **Tensorflow** with python. The python programming language is very intuitive with easy enough syntax to not steer away from the focus which is generative modelling.

It also has many libraries that are useful for the experiments such as:

Numpy for manipulating vectors.

Pandas for organizing hyper parameters in a formatted way.

Matplotlib for visualizing the results of both the learning progress as graphs and the outputs as images.

This list is not extensive as there are many other useful ones.

The **CUDA** developer toolkit utilizes NVIDIA GPUs that are highly effective for developing Deep learning models. The project will be run on a GTX 1660 Super.

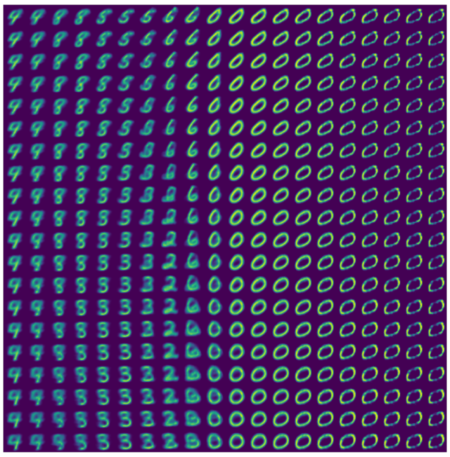
Experiments are run separately on each individual dataset. More models are tried on a single dataset to find out which performs best. In some cases, it is beneficial to plot results such as in VAE development where different clusters of data points can be visualized for better understanding. Additional features such as sampling from latent space will be added later and implemented on the easy dataset (MNIST) for testing. If the new features work as indented, they will be included in the other notebooks. For this reason, the first couple of notebooks will have more explanation on different functions as it wouldn't make sense to repeat it for all implementation throughout the project.

5 Results to date

VAE experiments

Dense model (left)

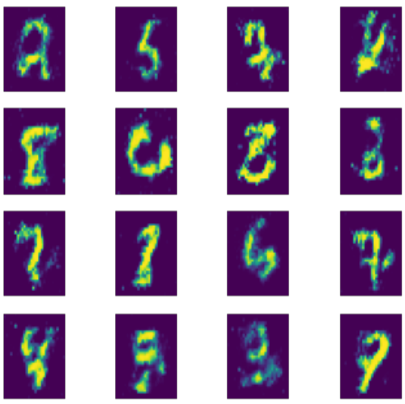
Convolutional model (right)



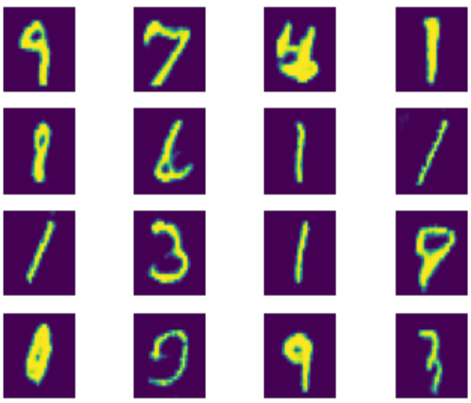
Gan experiments

MNIST digits

Dense minimal

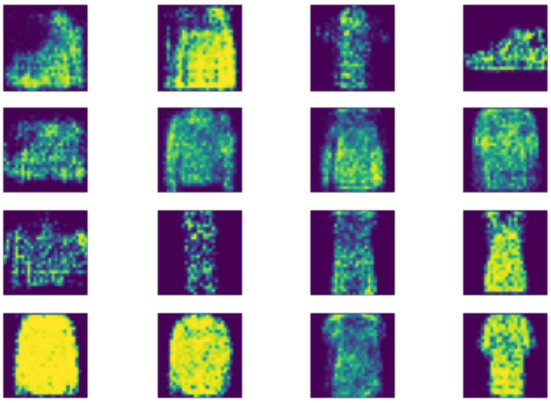


Convolutional minimal

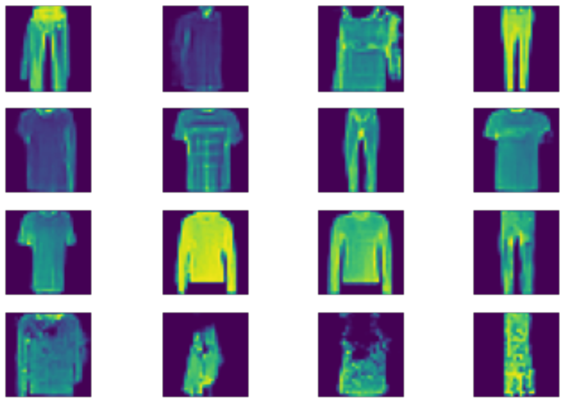


MNIST fashion

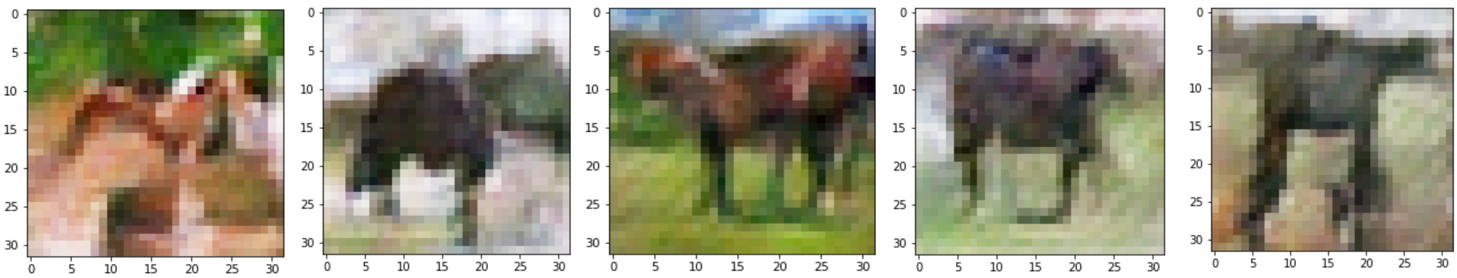
Dense model minimal



Convolutional minimal



CIFAR 10 horses



6 Discussion

MNIST dataset of handwritten digits

In the early stages of development after the concept was solidified the progress was mainly focused on trying out minimal neural networks to see what's the trade back for simplicity in terms of computer resources and the output. As the findings demonstrate that using only densely connected layers produce fairly noisy images but they are still clearly resembling the inputs in an abstract way.

In the case of variational auto encoders, the dense model blew up a few times during training (meaning it had unusable loss values from the beginning that produced nothing but noise) for some reason and couldn't produce anything valid but most of the time it made vague attempts to produce digits with not so much variance in between two points in latent space.

Performance

Dense: average training time was 7 sec per epoch on the whole training / test set concatenated of 60000 images.

Convolutional: also 7 sec per epoch but split up in 256 batches which is still fast considering the results and the complexity of the model compared to using only dense layers.

MNIST fashion dataset of clothing items

Performance

This dataset took much longer to run per epoch with the convolutional setting. On average 22 seconds per epoch compared to 6 for the dense model.

Cifar10 dataset of RGB images (on the horse class)

Using grayscale images had its advantages and limits. It allowed the comparison of densely connected and convolutional layers for minimal computation expense but haven't produced any interesting outputs. In case of colour images, it is safe to say that using only dense layers would be wasting both resources and time as all the nodes are connected therefore it can only resemble features in global space. Convolutional layers recognise local patterns which is highly beneficial for image generation.

The cifar10 dataset has many classes that are not as simple as digits. It would be interesting to combine cars and frogs but for simplicity one class is more than enough. The experiment was focused on 5000 samples of horse images. A batch size of 256 over 10000 epochs produced the images above. The model was run on an *Nvidia GTX 1660 super* for several hours. Despite the variety of horses, it made good progress in relatively short time.

These experiments showed the potential of generative modelling in terms of variations and complexity. It was necessary to train on low resolution data at the beginning to make a base model that works and has reasonable outputs. As the project has a foundation to build upon, other areas would be explored such as intentionally manipulating the latent space not just randomly sample from it.

7 Conclusion

Trough experimentation with different model architectures it became clear that convolutional neural networks are very powerful even in generative modelling. With a minimal architecture it is possible to train models that make good progress generating images in a short amount of time. It is also dangerous as these models are highly scalable and can exhaust resources very quickly even on low resolution data.

As this field is relatively new there isn't a science-based method of developing models using these generative techniques therefore we can only rely on tricks and intuition. It is important to give credit to those who provide hacks that are highly effective even for people who are new to the field. It is equally important to reference the source of data that is the backbone of machine learning and the inspiration for art.

With the basics and in a well-paced development process it is possible to get good intuition of generative modelling and it grants freedom of expression in the following phase to use this knowledge on traditional art that is unique to the artist. The output then can be used as a foundation for a piece or motivation for something totally different.

8 Commentary

Early challenges

The development process of this project wasn't straight forward as the technology stack wasn't solidified at the beginning. Most of the development was focused on the combination of the openFrameworks c++ library with Tensorflow. It took a bit of time and effort to realize that this idea is way too ambitious for the given time period. The main problem was that the initial concept restricted the creative process as the focus was on setting up the technology stack to work with in the first place. Without it the project wasn't built on solid grounds, therefore another approach needed to be in place.

Machine learning as the core with C++ on top was too vague. A different artistic output had to be considered. As I mentioned some artists provided inspiration for generative modelling and it was soon realized that the development of such technology for my personal use could be interesting and fun even without the use of creative tools like openFrameworks.

This initial struggle had a toll on the project in terms of progress as in the early stages time was spent on setting up technologies that didn't make it into the project. The development after the setback was relatively smooth as a clear and a solid idea was in place.

In terms of milestones, I tried to make up the shortfalls by developing simpler models and progressing in small steps on not to computationally demanding datasets. This way the training process is not time consuming, and I am able to test different model architectures that could work also on higher resolution images.

Future plans

In terms of GANs after developing solid models, the next step is to sample generated output and try out different transitions between two points in latent space. Transformation of two vectors in latent space could lead to interesting variations. During the development process most of the outputs will be saved and later curated by me to showcase the volatile nature of the models. Not always the best models create the most interesting outputs.

For VAE the focus is utilizing the variation aspect to create interesting images.

It is also important to develop custom functions and classes that perform well and less redundant. Generative modelling is quiet challenging therefore intuition is the key aspect to keep in mind.

My intention for this project is also to try it on my own data such as traditional sketches and artworks that I made with my own style. I will spend a good week away from computers during reading week therefore I could focus on creating many quick sketches on iPad that could be interesting to run models on and see how the model twists them.

Potential risks and solutions

Model complexity

GANS are hard to train therefore it takes time and finetuning to make a model work. By gradually increasing the difficulty of the datasets such as the resolution or variety of samples I can alter already working code that I tested on smaller models. This way I won't be wasting time on models that won't work.

Data quality / quantity

With the increase of resolution comes increase of computing power consumption. If the resources are exhausted, then reducing the batch sizes negatively effects the learning process and would take more time to learn. It can be mitigated by restricting the resolution to a certain size (no more than 128 * 128). Other techniques such as cropping images would certainly reduce the resolution further. The amount of data available also affects the process. An optimal sample pool needs to be determined for each dataset.

Custom art

Creating art by hand is time consuming and it takes effort. A small set of sketches (grayscale at first) would be made by hand over the next few weeks. By resizing the images and then train the model might create interesting variations. It is one of the most challenging issues regarding this project that I will be focusing on. Simple sketches would be easier to create and train.

Crediting code / data

Make it obvious that some of the code or concepts I use is borrowed and credit those who made it. This can be a model architecture a data scraping script and many more.

Customization

It is crucial to show my understanding of the topics by constructing custom functions with explanation. It also helps the development process.

Documenting performance

As explained earlier it is not possible to evaluate model performance on accuracy. The best I can do is monitor the outputs and choose the best ones that I find the most interesting. Hyper parameters such as learning rates, activation functions and other model specific information would be documented upon which the project could be evaluated on.

Additional ideas

- More custom functions for saving, plotting outputs, documenting hyperparameters
- Creating commands to train and run the models from the terminal
- Showcasing a curated portfolio of the best outputs in openFrameworks

Plan outline

2 weeks - Focus on modularity of models and various datasets / custom functions

1week – creating my own dataset of art sketches / presentation work / training

2weeks – Polishing model architectures / training

1 week – presentation fix up / work on cleaning up code base

2-weeks - more experiments on architectures with own dataset

4-weeks - final report work / any additional feature implementation if time permits

Bibliography

[1] F. Chollet, *Deep Learning with Python*. Shelter Island: Manning Publications Co., 2017.

[2] J. Langr and V. Bok, *GANs in Action*. Manning Publications Co., 2019.

[3] David Lopez-Paz, *NIPS 2016 Workshop on Adversarial Training - Soumith Chintala - How to train a GAN*, (Feb. 17, 2017). Accessed: Jan. 15, 2022. [Online]. Available: https://www.youtube.com/watch?v=X1mUN6dD8uE

[4] J. Vincent, 'How three French students used borrowed code to put the first AI portrait in Christie's', *The Verge*, Oct. 23, 2018. https://www.theverge.com/2018/10/23/18013190/ai-art-portrait-auction-christies-belamy-obvious-robbie-barrat-gans (accessed Oct.10, 2021).

[5] Steven Van Vaerenbergh, *Ian Goodfellow: Adversarial Machine Learning (ICLR 2019 invited talk)*, (May 12, 2019). Accessed: Jan. 5, 2022. [Online]. Available: https://www.youtube.com/watch?v=sucqskXRkss

[6] DigitalSreeni, *Tips Tricks 16 - How much memory to train a DL model on large images*, (Aug. 16, 2021). Accessed: Jan. 31, 2022. [Online]. Available: https://www.youtube.com/watch?v=wPuSAPOMVj4

[7] 'Quasimondo | Mario Klingemann, Artist'. http://quasimondo.com/ Accessed: Jan. 31, 2022.

[8] 'Memories of Passersby I | Quasimondo'. https://underdestruction.com/2018/12/29/memories-of-passersby-i/ Accessed: Jan. 31, 2022.

[9] 'Robbie Barrat'. https://robbiebarrat.github.io/ Accessed: Jan. 31, 2022.

[10] 'Robbie Barrat'. https://robbiebarrat.github.io/oldwork.html/ Accessed: Jan. 31, 2022.

[11] 'Scott Eaton'. https://www.scott-eaton.com/ Accessed: Jan. 31, 2022.

[12] '"Human Allocation of Space" » Scott Eaton'. https://www.scott-eaton.com/2020/human-allocation-of-space-2 Accessed: Jan. 31, 2022.

[13] 'Sofia Crespo'. https://sofiacrespo.com/ Accessed: Jan. 31, 2022.

[14] 'Artificial Remnants 2.1'. http://artificialremnants.com/ Accessed: Jan. 31, 2022.

[15] 'Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition [Book]'. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/ (accessed Feb. 01, 2022).