Proof of Concept

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

Essential features:

- Variational Auto encoder : an example of generative modelling using a VAE neural network to experiment with it's model architecture and evaluate on the results in latent space. As the closest precursor to GAN's are variational Auto encoders I found it important to dig deeper into their functionality especially how they utilize the so called latent space. They differ in architecture therefore it offers another perspective creatively.
- Generative Adversarial Network : a basic implementation of a GAN as a sequential model architecture. Training GANs is "notoriously difficult to train" as *François Chollet statets it in his book for Deep learning in Python[3*]. I want to dig deeper as what's possible as an artist working with relatively basic hardware to get out a model. I am also interested in the different possibilities of different architectures on different data and the aim is to have a well rounded experiment In which I find answers to how useful generative modelling could be integrated into my creative practice.
- Separation of concerns: primary focus is on making the program as modular as possible in order to allow for easier experimentation and shorter codebase.
- Virtual environment to use all the necessary packages during development in isolation. ('requirements.txt' is provided with all the used packages and their dependencies for the project)

Essential Technology:

- Python programming language used with Jupyter notebooks makes it easy and modular to document and to code during development. It is also extremely easy to separate code blocks and move them around or in and out from a notebook.
- Tensorflow machine learning framework with Keras library. In my Artificial Intelligance module I learnt a great deal using Tensorflow intuitively with deep neaural networks. Keras makes it even easier to focus on the model architectures and training experiments rather than defining complicated loss functions and all other crucial elements that already comes with Keras out of the box.
- Pycharm code editor: It makes it easier to edit and run notebooks straight from the editor without the use of a terminal. Project management is also more convenient using Pycharm to work on multiple experiments in different notebooks at once.

Additional technologies:

- OpenFrameworks for displaying the curation of best images. By experimenting with the ofxTensorflow2 addon I came across some difficulties related to displaying images straight from a model. As Gans produce images while training I could only display a single image and modulate it in latent space. I came to the conclusion that combining two technologies

that both differ in syntax and functionality might be a too ambitious move at least for a minimal viable product.

- Tensorboard: In order to have more insight into training a model Tensorboard is a very helpful tool to utilize during training to save time when a model performs suboptimal.

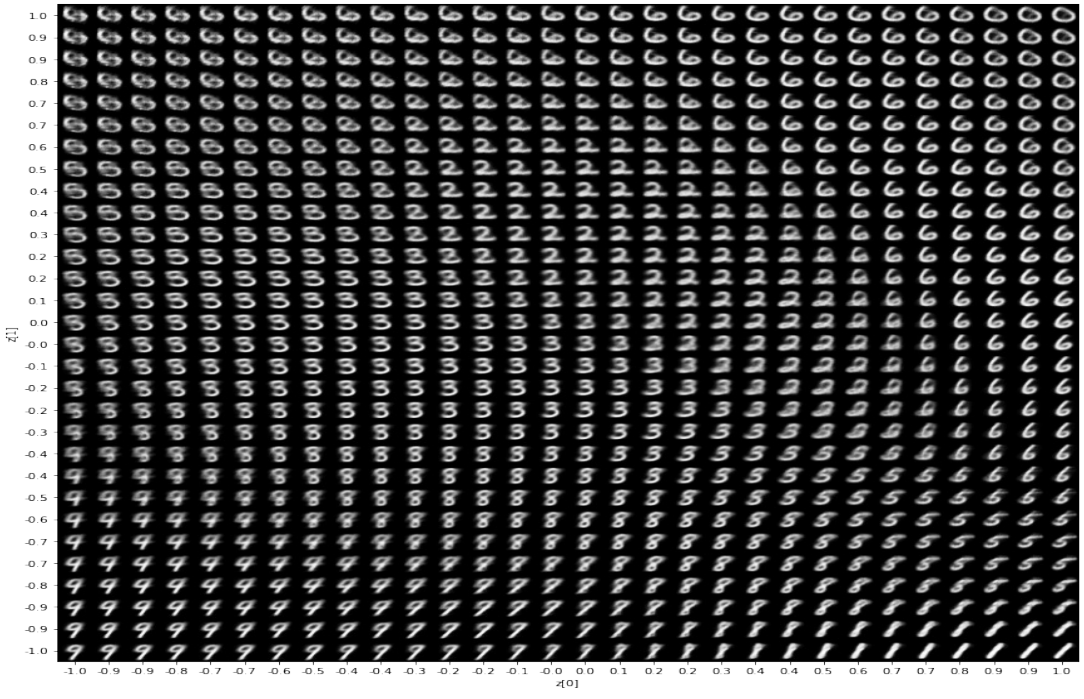
Experiments

As I am new to field, a fair amount of research took place in the previous months regarding all aspects of deep learning. From Dense layer classification and regression models to convolutional neural networks and sequence models. Generative models require a bit more resources to understand the concepts as they are advanced topics and the architectures are quiet unique. For this reason I was relying on source materials from Tensorflow [1] and Keras [2]. The code was run for the experiments and commented by me in markdown blocks to demonstrate my understanding of the concepts.

The fundation for my project is to get a good enough intuition of these different deep learning architectures to later build upon and customize the code based on my preference.

The MNIST dataset provided the models with a lot of variation from the 60000 training samples.

VAE model result in latent space (after 30 epochs):



GAN model end result with 16 seed image (after 50 epochs):

These two experiments gave me a fair amount of understanding of the topics and confidence to modify the code in order to optimize the architecture and train on more interesting data.

- The aims for the future development of this project:
- Customized modular code with custom functions and classes
- Experimentation with model architectures both sequential and the functional API
- Interesting datasets of various images
- Test GPU at hand on the highest resolution possible to get reasonable results
- Training model on handmade data of sketches and drawings

Bibliography

[1] "Deep Convolutional Generative Adversarial Network | TensorFlow Core", *TensorFlow*. [Online]. Available: https://www.tensorflow.org/tutorials/generative/dcgan.

[2] F. Chollet, "Keras documentation: Variational AutoEncoder", *Keras.io*, 2020. [Online]. Available: https://keras.io/examples/generative/vae/.

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

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