Deep learning or in general machine learning algorithms require immense amount of data to achieve high performance and to generalize well (Norvig, 2011; C. Sun et al., 2017). Data acquisition and annotation can be highly time consuming and prone to errors, especially in domain specific applications (Montani and Honnibal, 2017; Y. Sun et al., 2017).

Possible solution:
- Use generative modelling to produce artificial data samples that mimics properties of real data samples
- Artificial data samples might be used for pretraining deep learning models, while the valuable real data is saved for finetuning the model
- Generated data samples might provide different plausible combinations of modalities, than are present in the original real data
  - E.g. color, texture, shape

Challenges:
- Generate realistic looking data
- Model high intra- and inter-class variance in the artificial data
- And possible control of these variations through conditioning

METHODOLOGY

Generative adversarial networks (Goodfellow et al., 2014)
- Minmax game between two players:
  - A discriminator, distinguishing real and fake data samples
  - A generator, producing artificial samples in an effort to cheat the discriminator.

\[
\min_{\theta} \max_{\theta'} V(D, G) = \mathbb{E}_{x \sim P_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim P_{\text{noise}}} [\log (1 - D(G(z)))]
\]

InfoGAN (Chen et al., 2016)
- Unsupervised condition for structuring/controlling the appearance of the generated samples.
- To try to maximize the mutual information between the additional latent variable generator input and the generated samples
- Formally implemented as an additional regularization term in the objective:

\[
\min_{\theta} \max_{\theta'} V(D, G) = \mathbb{E}_{x \sim P_{\text{data}}} [\log D(x)] - \mathbb{E}_{z \sim P_{\text{noise}}} [\log (1 - D(G(z))] \\
\min \mathbb{E}_{x \sim P_{\text{data}}} [\lambda \mathbb{I}(G(z)) - \mathbb{E}_{z \sim P_{\text{noise}}} [\log G(z)]]
\]

EXPERIMENTS & RESULTS

Setup for creating artificial MNIST colour images:
- Increasing the challenge by randomly colouring of MNIST data
  - Changing the generator to output a RGB image, instead of grayscale
  - Latent variable design
    - 3 continuous variables (random uniform in the range [-1;1])
    - 1 categorical variable (one-hot encoded, K = 10, p = 0.1)
  - Able to capture the underlying categorical structure
  - Able to capture dominating continuous variations in the data (color, shape, style, etc.)

Improvements to be made:
- Higher resolution is needed for more details in the generated images (Karras et al., 2017; Zhang et al., 2017)
- Better objective formulation to increase training stability (Gulrajani et al., 2017; Salimans et al., 2018)

REFERENCES


Figure 1: continuous variables. Each figure show different combinations of manipulating using two of the continuous latent variables (one along the x-axis and one along the y-axis), with the last one fixed at 0.

Figure 2: Comparative variants. Each figure show the representation learned for each compound latent variable. Each is generated by interpolating along the first and second continuous variable (x-axis and y-axis respectively).