Abstract:

Variational auto-encoders (VAEs) are a well-established class of models that specialize in the unsupervised learning of complex distributions. While standard autoencoders are discriminative and learn a compressed representation of the input, VAEs are generative and learn a latent representation of the input. This latent representation is parameterized by Gaussian means and standard deviations, which correspond to a set of Gaussian distributions that can be sampled from for generating output. We, at the Interactive Machines Group, would like to explore the effectiveness of VAEs in robot perception. If a robot can understand and synthesize sequential context clues from its surroundings, then it can build a more accurate model of the world. This would allow the robot to comprehend the emotions of nearby agents and make more informed decisions in complex social situations. In order to better understand the models we train, I developed an interactive web application to visualize VAE models. This app is built using Nuxt.js, an open-source framework based on Vue.js for building modern web applications, and Tensorflow.js, a JavaScript library for loading and deploying machine learning models. Through the web application, users can upload Tensorflow-trained models, visualize them online, and modify inputs and latent space parameters to determine their influence on outputs. The current version only works with models trained on the MNIST dataset, but it can easily be extended to work with models trained on any dataset. This web application is a powerful tool for visualizing how different parameters affect model performance and will facilitate our understanding of the VAEs we train for future projects.
Introduction:

Variational auto-encoders (VAEs) are a class of generative models that learn low-dimensional latent representations of inputs. As the name suggests, VAEs have the same structure as autoencoders, but the main difference is that the latent space representation is parameterized by Gaussian means and standard deviations, which allows for generative sampling of outputs. Because of their architecture, VAEs are effective models for encoding sequential data.\(^2\) In particular, a study by Yingzhen Li and Stephan Mandt showed that VAEs are capable of disentangling dynamic features (i.e. ones that change over time) from static features in video recordings.\(^2\)

We would like to investigate the effectiveness of VAEs in disentangling complex social contexts. If a robot can understand the thoughts and emotions of other agents in the world, then it can make more informed decisions. To better comprehend a specific VAE model, we have to closely examine its latent space and analyze how the Gaussian means and standard deviations affect reconstructions. VAEViz is an application that gives users a platform to play around with their model’s latent parameters and visualize how they affect the reconstructed output. By making the model more interpretable, the application exposes the structure of the latent space and allows the user to determine whether the VAE is adequately disentangling the input.

Implementation:

The application is built using Nuxt.js, which incorporates HTML, Vue.js, and CSS, and Tensorflow.js. To start using the interface, the user must first select a model to load. Additional models can be uploaded by first converting the h5 file into the appropriate Tensorflow.js format and then adding a folder to the static/models directory. Note that for a VAE, the encoder and decoder need to be saved separately.
After loading the model, the app takes the user to a new page to visualize output. Initially, the output is generated from random samples of the latent space, but the user can toggle between random samples and encoded input. The latter brings a slider that allows the user to select a specific input image. In the ‘Use Encoder’ mode, the input image is first fed to the encoder to compute a latent representation, and then, the latent representation is fed through the decoder to generate 100 output images. The means and standard deviations can be adjusted through the sliders on the right to examine how their values affect the output. The sliders are accompanied by text boxes that display their values.

The VAE model is loaded into the web browser itself, allowing for real-time interaction with the latent space parameters. This application is a powerful tool for visualizing how the Gaussian means and standard deviations affect reconstructions and will facilitate our understanding of the VAE models we train for future projects.

**Future Work:**

A few possible improvements to the current version of VAEViz include adding the option to visualize multiple models side-by-side, making input selection more intuitive, and training additional VAE models that force greater disentanglement of the latent space parameters.
However, now that we have a functioning visualization tool, we can easily extend VAEViz to work with models trained on other datasets, which will prove useful for future research with VAEs.

Acknowledgements:
Special thanks to Nathan Tsoi for guiding me through the development process!

References: