1 Vocabulary Statistics

- latent variables (in statistics) are not observed, but rather inferred though other variables, that are observed
- observables (in statistics) are observed variables or factors of a dataset
- generative model in statistical classification including machine learning is not consistently defined, but wikipedia says "a generative model is a model of the conditional probability of the observable X given a target Y" $P(X|Y = y)$[1]
- discriminative models are (following wikipedia) "are models of the conditional probability of the target Y, given observation x $P(Y|X = x)$"
- unit Gaussian is the simplest gaussian distribution with unit variance and the integral equals to one
- diagonal Gaussian ... from stackexchange.. special case where the only entries are on the diagonal.. an ellisoid in 3D...most of the mass concentrated near the center
- mean - and log-variance
- ELBO (evicence lower bound) ...only for VAE's??? dunno pls check

2 Vocabulary Python

- super- inherit smth from one class to another
- keras.layers.Flatten -
- reduce_sum
- gradienttape

3 Convolution Neural Network (tensorflow.org/tutorials/generative/cvae)

3.1 Main Network Ideas

The generative and interference network are wired up with the Keras Model Sequential. **tf.keras.Sequential** is used for small networks. In this example 2 Convnets are bulid for the generative and interference network. The generative model takes a latent encoding as input, and outputs the parameters for a conditional distribution of the observation (x) $\rightarrow p(x|z)$. Additionally, a unit Gaussian prior $p(z)$ is used for the latent variable (z). The inference network defines an approximate posterior distribution $q(z|x)$. It takes as input an observation and outputs a set of parameters for the conditional distribution of

[1] `https://en.wikipedia.org/wiki/Generative_model`

the latent representation. q is defined as a diagonal Gaussian. The output is then the mean and log-variance of a factorized Gaussian. A reparameterization trick is used during optimization: durig optimization samples are taken from $q(z|x)$ by first sampling from a unit Gaussian, then multiplying by the standard deviation and adding the mean. "This ensures the gradients can pass through the sample to the inference network parameters." The inference network comprises two convolutional layers followed by a fully connected layer. The generative network mirrors this architecture by using a fully connected layer followed by three convolution transpose layers. **WHY?** it is common practice to avoid using batch normalization when training VAE's, since the additional stochasticity due to using mini-batches may aggravate instability on top of the stochastisity from sampling.

3.2 Loss function and optimizer

Maximizing the evidence lower bound (ELBO) on the marginal log-likelihood is typical to VAE:

$$logp(x) \geq ELBO = E_{q(z|x)}\left[log\frac{p(x,z)}{q(z|x)}\right]$$

In practice, the single sample Monte Carlo estimate of this expectation is optimized: $logp(x|z) + logp(z) - log(z|x)$, where z is samples from $q(z|x)$

3.3 Stopping here

The proposed CVAE did not work on my system eccept for the running in the juypiter notebook. As I am not using a variational autoencoder, this ection stops here. Working with tensorflow was first experienced here, so this experience is taken further.

4 Convolutional Autoencoder Sandia 2019

4.1 Initial condition satisfaction

Remark 3.1 (Initial-condition satisfaction). Satisfaction of the initial conditions requires the initial general- ized coordinates ˆx(0;) = ˆx 0 () to satisfy g(ˆx 0 ()) = x 0 (). This can be achieved for any choice of ˆx 0 () provided the reference state is set to x ref () = x 0 () g(ˆx 0 ()). (3.4) However, doing so must ensure that the decoder can accurately represent deviations from this reference state (3.4). In the context of the proposed autoencoder-based trial manifold, Section 5.3 describes a strategy for computing the initial generalized coordinates ˆx 0 () and dening training data for manifold construction such that the decoder accomplishes this.