Autoencoder Shenanigans

Questions

- What loss to use for each VAE type?
- How to visualize latent decomposition?
- What can learn something from the latent space of AE/VAEs?
  - Is there a significant location in latent space that houses high n_e^{ped}?
  - How does the latent space change with CVAEs, i.e., by propogating certain parameters (e.g., β_N as conditional) change latent space?
- How does taking a smaller subset of the dataset effect the latent space?
  - AE/VAEs are sensitive to the data they are fitted with.

Observations

So to run an experiment, clone this repo and navigate to `/src/vae-shenanigans`.

There you have a file called `run.py` which if you check `python3 run.py --help` you can see a list of parameters you change within the VAE, for example `python3 run.py --epochs 200 --learning_rate 0.001 --latent_dim 20` will run a vanilla VAE for 200 epochs, with the learning rate 0.001,and a 20D latent dimensional space. After running, the directory `/vae_exps/{EXP_NAME}` will have the results of the metrics like loss, KL DIV loss, and reconstruction loss stored in `metrics.csv`. I am still working on getting the plotting component added. I am trying to write it as high level as possible for us after my vacation so that we can really do some fun experiments. So far just C-VAE, Beta-VAE, and Vanilla-VAE are implemented, but I am working on getting some more.

If you want to just to do simple experiments, and see the simplified version of the code, check `experiments.py`. It has same parser capabilities, but also plots the latent space at the end. Just make sure which autoencoder you are using, at the moment it uses the C-VAE, but you can swap on line 72 `cvae = ConditionalVAE(...)` with `cvae= VanillaVAE(...)` without problems.

If you want to use the basic Autoencoder and see the latent space, use the program `simple_AE.py`. All use the same argument parsing capabilities.

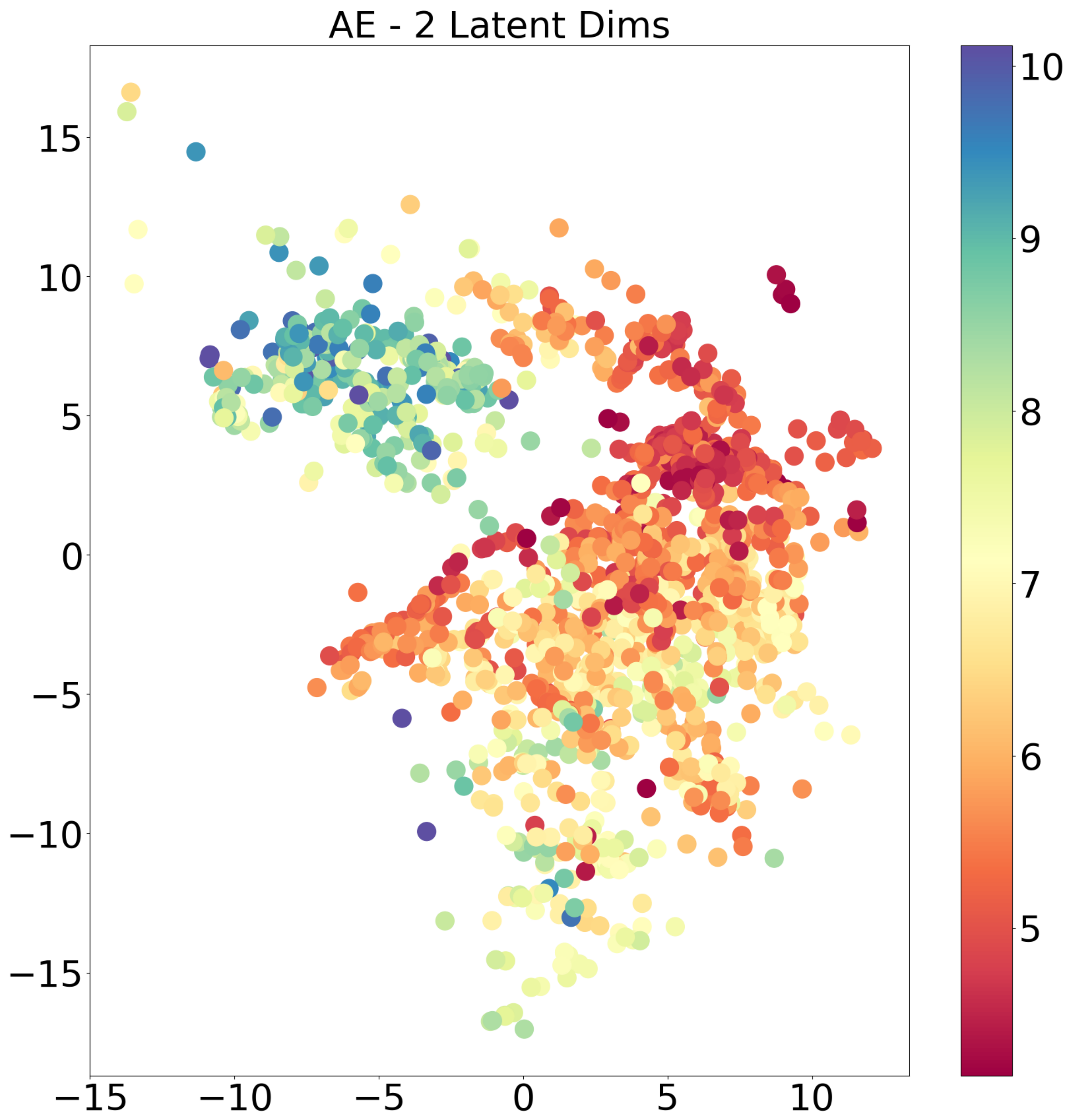
Requirements

You must install these libraries:

- pytorch
- pytorch_lightning
- test_tube
- numpy, pandas, sklearn

Autoencoder

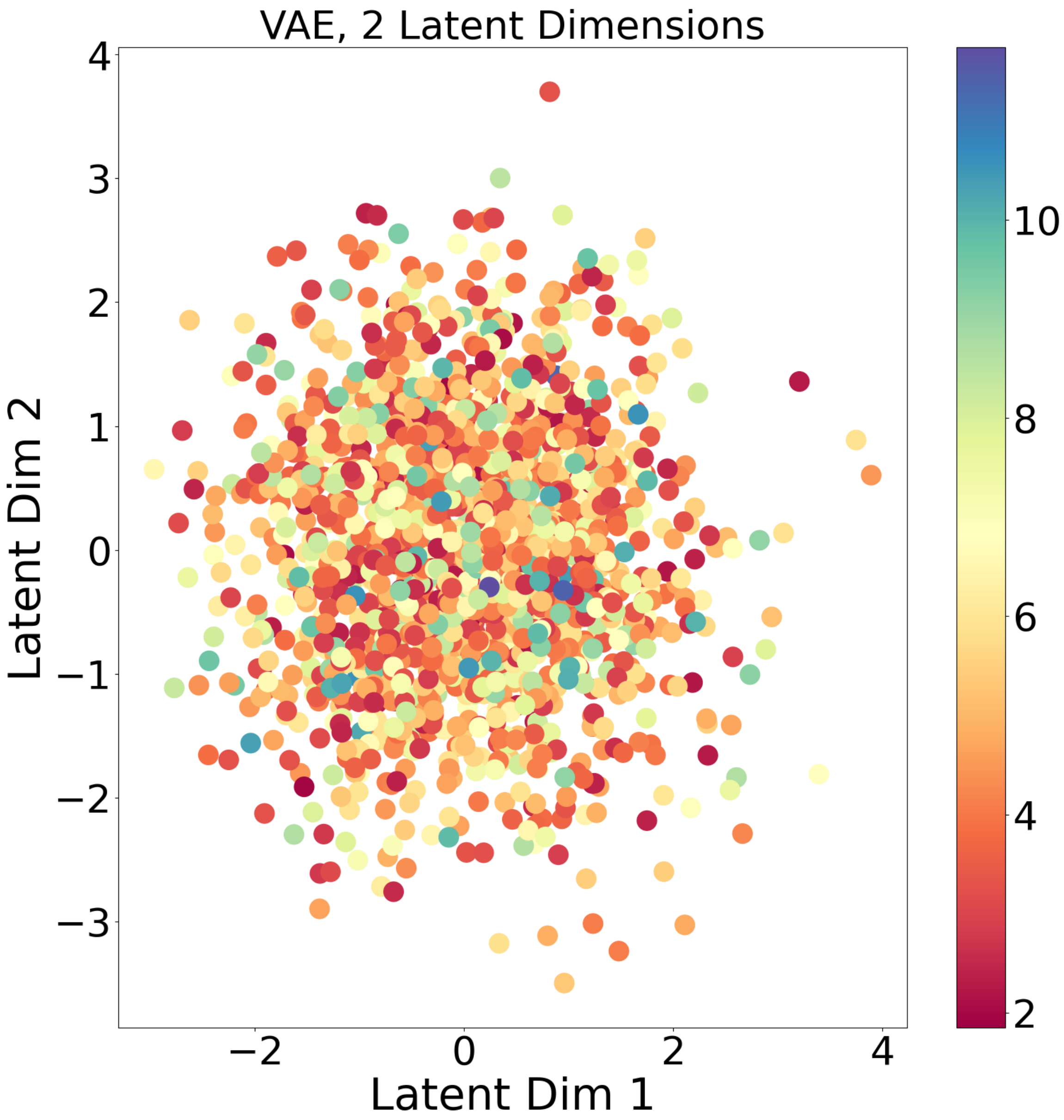
The loss for an AE is just the MSE of the reconstruction of input parameters against their actual values. By minimizing that, we get a NN that cna map the space.
The AE latent dimensions do show that high n_e^{ped} is located in a different space than the lower values (figure below).
However, we also see that the latent space is not continuous. This we would prefer, as we want to be able to interpolate, i.e., see the change from low neped to high neped (if it is possible). See [1] and this [medium article]



The thing is, at the moment, if we are really just concerned with solving the problem of high neped, I don't see this as an issue.
We do not really care about reconstructing, or generating new samples (at the moment), so in the context of modeling, this could be useful for handing to a guess to a new model.

Vanilla Variational Autoencoder

The model is able to overfit and find the mean of the data quite well, and thus generate some samples, but without much variation.
As seen in the figure below, the space is more or less continous, but lacks any form of generalization to the naked eye like there was for the AE above. I doubt it it is finding out about high neped, but it could be capturing other correlations of the data that I have not checked in on yet.
The problem is that the KL divergence loss factor limits the latent space from getting discontinous, therefore there is no hard separatation of points in latent space like there is for the AE.



C-VAE

The conditional VAE is achieves a better RMSE on reconstructing the input values, but has very similar results with the latent dimensional space (when propogating n_e^{ped}) as the VAE, meaning that there is some latent space, and it will take some time to figure out what it means XD.

β-VAE [2]

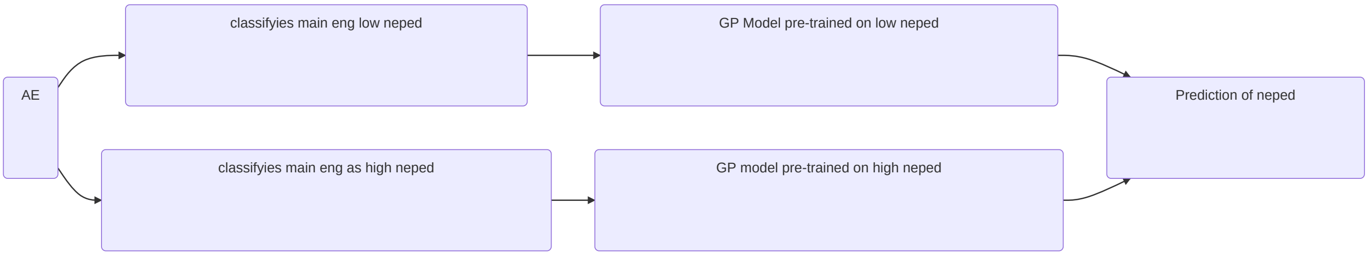
Implemented but not tested. Basically to balance the loss problem this is suggested but not sure.

Answers initial questions

- What loss to use?
    - Autoencoder
        - Reconstruction Loss (MSE)
    - Vanilla
        - Reconstruction Loss + KL Div of Reconstruction
    - Conditional
        1. Reconstruction loss + Conditional Loss+ KL div reconstruction + KLdiv
        2. Reconstruction loss + KL Div reconstructions
- How to visualize latent decomposition?
    - t-SNE
    - 2 latent dims side by side
- What can learn something from the latent space of AE/VAEs?
    - Is there a significant location in latent space that houses high n_e^{ped}?
        - Yes, for each latent dimension there seems to be some correlation in the variational autoencoder
            - **Potential for a classifyier for high neped that feeds to a dual model for predictions?**
        - Check joint distribution plots between global parameters and find out.
    - How does the latent space change with CVAEs?
        - Depends on which variable is added conditionally. Should try β
- How does taking a smaller subset of the dataset effect the latent space?

Future Work Questions

I think an idea worth looking into is to transform the Autoencoder into a classifying model, then use the following structure:



- Can we build a hetereoscedastic VAE for the JET-Pedestal database?

References

[1] Generating Sentences from a Continous Space

[2] β-VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK