CASE with variational autoencoders

CMG Anomaly Meeting – Sept 16, 2021

Anthony, *Benedikt*, *Javier*, *Jennifer*, *Kinga*, *Maurizio*, *Nadya*, *Thea*

Introduction

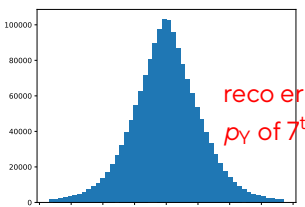
- We now have a framework (Kinga's software packages) that is able to read the "official" CASE datasets, train an algorithm, and look at ROCs … up to QR, etc
- Required some plumbing and getting familiar with the framework
- GitHub packages and PRs:
  - https://github.com/kingusiu/vande/pull/1
  - https://github.com/kingusiu/sample-read-write-transform/pull/1
  - https://github.com/dr-stringfellow/preprocessCASE
  - https://github.com/kingusiu/pofah/pull/1
- Using files at /eos/cms/store/group/phys_b2g/CASE/h5_files/full_run2/
  - This is UltraLegacy for QCD
  - For the signals, we use _very small_ legacy datasets from Feb 2021

Starting slow

... with a simple, fully connected network to encode and decode QCD based on the first 100 constituents in a jet.

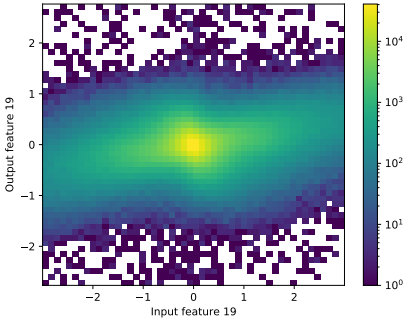
- Implemented in torch (i.e. outside of Kinga's framework)
- Features are normalized to have mean 0, stddev 1
- Training based minimizing reconstruction loss (no KL)
- Train on 2/3 of events in sideband ($\Delta\eta(jj) > 1.4$), validate on 1/3

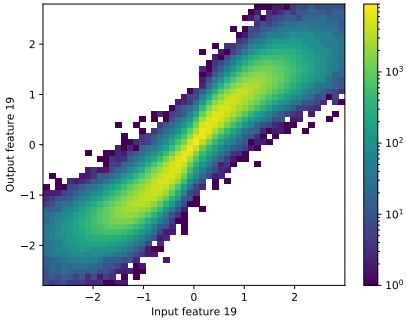
n_epochs: 10
n_features: 300
n_latent: 12
n_hidden: [64, 32, 16]
batch_size: 512
lr: 0.01
lr_decay: 0.03
kl: False
variational: True



reco error for feature 19
p_Y of 7th jet constituent

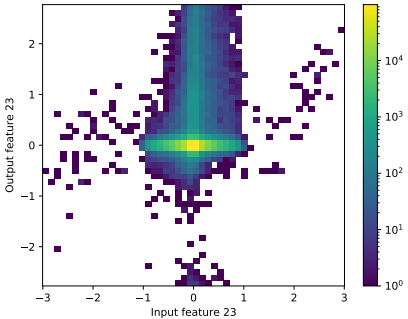
Reconstruction performance

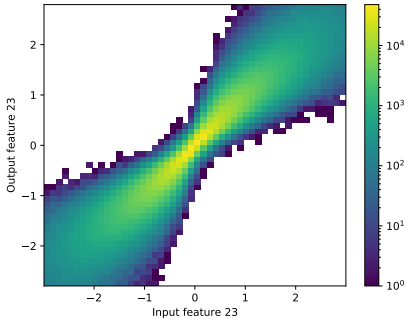




- For some reason, network cannot reconstruct normalized features (p_T^C/p_T^J, $\eta^C - \eta^J$, $\phi^C - \phi^J$), but it can reconstruct bare features p_x^C, p_y^C, p_z^C
- Looks more like a bug than anything to me
- Currently also implementing these plots in Kinga's framework
- Files are also inferred, but have not had time yet to plot a ROC curve

Reconstruction performance, cont.





VAE 2DConv (Kinga's model)

```
params = Parameters(run_n=113,
                    input_shape=(100,3),
                    kernel_sz=(1,3),
                    kernel_ini_n=12,
                    beta=0.01,
                    epochs=60,
                    train_total_n=int(2e6),
                    valid_total_n=int(5e5),
                    gen_part_n=int(5e5),
                    batch_n=512,
                    z_sz=12,
                    activation='elu',
                    initializer='he_uniform',
                    learning_rate=0.001,
                    max_lr_decay=8,
                    lambda_reg=0.0) # 'L1L2'
```

- Took the parameter config that is in GitHub (I assume latest and greatest)
- Model uses a 2D convolutional layer and a series on 1D convolutions in the encoder (accordingly, ConvTranspose in the decoder)
- Trainable params: 417,701

VAE 2DConv (Kinga's model): ROC curves

Instead of showing a few plots here, let's just look at all of them on the www:

http://t3serv001.mit.edu/ bmaier/view.php?dir=figs/case/kvae/v0

- This _should_ be directly comparable to Nadya's ROC curves.
- To-Do's: make sure we have a set of results (QR, fit, limit)
- In parallel investigate and understand architectures