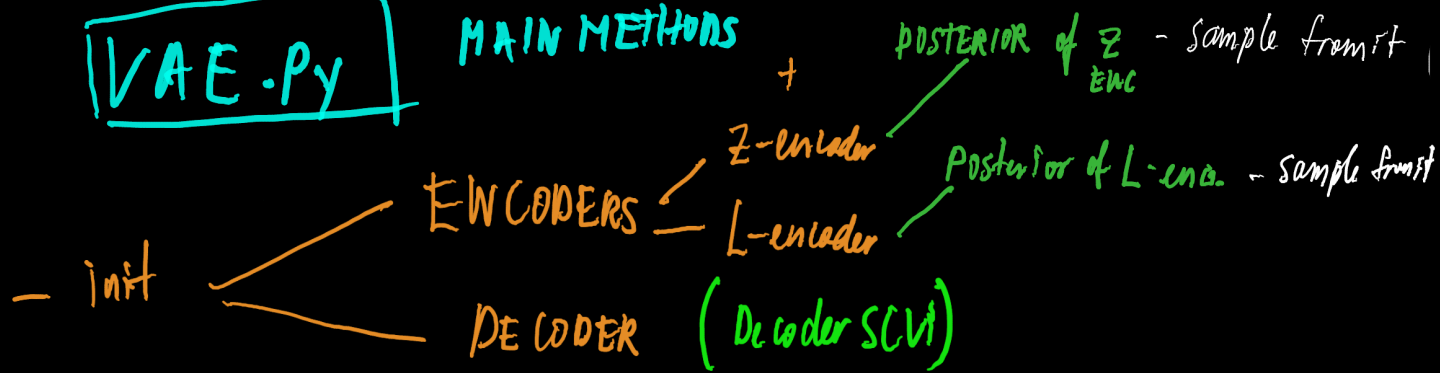


VAE.py

MAIN METHODS



FIND out: what is p_X ?

p_X is a type of a decoder
↳ FC layers

INFERENCE

Encoders

↳ Sample from Encoders

↳ Decoder: Use these samples, building

3 decoders
based on latent space

- Sequential Linear Softmax RN
- Linear for R
- Linear for Dropout

+ dependent whether you have

- gene w/ label (gene-label)
- gene w/ batch
- gene

↳ takes gene r or linear comb. from $p_{z,r}$ from the val-object

↳ Inference is called in FORWARD PASS.

Forward pass finds the KL divergence between the $\left\{ \begin{array}{l} \text{pre-defined param} \\ \text{variational } W\text{-dist} \end{array} \right\}$ and the $W\text{-dist}$ found by Inference, both GLOBALLY (z) and locally (l)

In addition, forward pass finds the REconstruction loss using input x (data) and parameters obtained from inference.

Forward pass returns: (Reconst. loss + Local-KL, Global-KL)

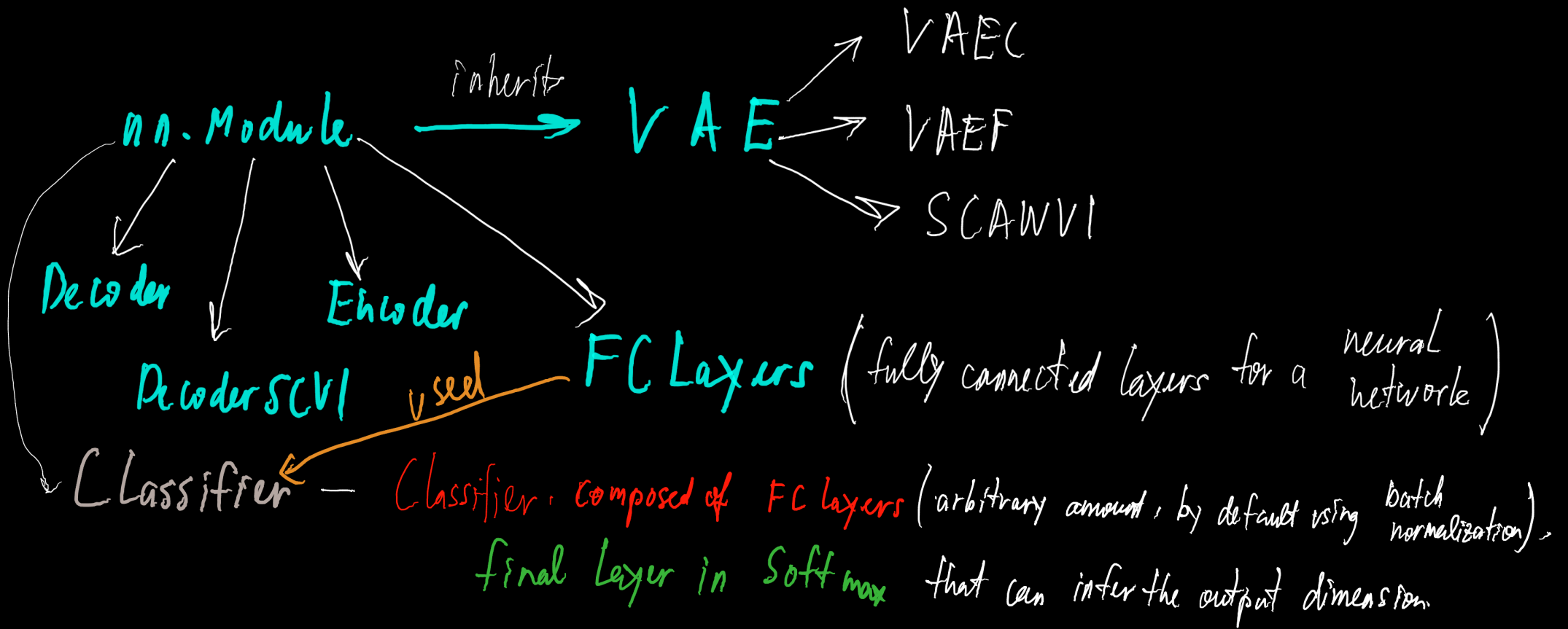
~~A abstract~~
High-level Class Trainer that Unsupervised Trainer inherits from:

↳ Mostly has training and perf related methods e.g. to compute metrics on the posterior distribution.

+ Methods to both CORRUPT and UNCORRUPT Posteriors - adding noise to the posteriors

Differentiate between inference method inside VAE and INFERENCE module that has a file called Inference.py.

MODELS - MODULES - Children of nn.Module



MERYA: Need Corruption, don't need differential expression

TOTAL LOSS = binary cross entropy + KL-divergence

CHRIS reconstruction Loss

VAE

using encoder

encode: return reparam μ, σ

reparam: returns samples from posterior $q(z)$

→ Z-encoder, L-encoder

→ get-latents → given (x, y) , sample from $q(z)$

forward { encode $\xrightarrow{\mu, \sigma}$ reparam \xrightarrow{z} return $\left(\begin{matrix} \text{decode}(z), \\ \mu, \\ \text{Log}(\sigma) \end{matrix} \right)$ }
+ Loss
↓
returns reconst + variational dist params

FORWARD PASS

forward calls inference

→ from our VAE to PBMC

$\text{Log var} = \text{Log}(\sigma^2)$ - Log var used because of numerical stability?

$$\text{Log var} \cdot \exp() = \sigma^2$$

$$\text{Log}(\sigma^2) = \frac{1}{2} \text{Log var}$$

Variational Lower bound:

$$\mathcal{L} = E_q \left[\text{Log} \frac{P(X, Z)}{q(Z)} \right] = E_q \left[\text{Log} P(Z, X) - \text{Log} q(Z) \right]$$

$$D_{KL}(q(Z) || P(Z|X)) = -\mathcal{L} + \text{Log} P(X) = -E_q \left[\text{Log} P(Z, X) - \text{Log} q(Z) \right] + \text{Log} P(X)$$

In our case:

$$-E_q \left[\text{Log} P(\tilde{x}|z) P(z) - \text{Log} q(z|x) \right] + \text{Log} P(X) = \text{DROP}$$
$$E_q \left[\text{Log} [\mathcal{N}(f(z)|0, f(z)|1), \mathcal{N}(0, I_2)] \right] + E_q \left[\text{Log} [\mathcal{N}(g(x)|0, g(x)|1)] \right]$$

Loss: \rightarrow Bin-centr (data, reconstruction) \div bce

$$\rightarrow \text{KLD} = \left[-\frac{1}{2} \cdot \sum \left(1 + \text{Log var} - \mu^2 - \sigma^2 \right) \right] / \text{norm} \leftarrow \text{If data is binary, KLD has closed form}$$

Return: bce + KLD

THEORY:

$$D_{KL}(\mathcal{N}_0 || \mathcal{N}_1) = \frac{1}{2} \left[\text{tr} \left(\Sigma_1^{-1} \Sigma_0 \right) + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_0 - \mu_1) - k + \ln \left(\frac{\det \Sigma_1}{\det \Sigma_0} \right) \right]$$

In variational inference:

$$D_{KL} \left(\mathcal{N}(\mu_1, \dots, \mu_k)^T, \text{diag}(\sigma_1^2, \dots, \sigma_k^2) || \mathcal{N}(0, I) \right) = \frac{1}{2} \sum_{i=1}^k (\sigma_i^2 + \mu_i^2 - \ln(\sigma_i^2) - 1)$$
$$= \frac{1}{2} \sum (\sigma_i^2 + \mu_i^2 - \log \sigma_i^2 - 1)$$

= D_{KL}

6 VIDEOS:

0: Introduction to
SCVI as Dimensionality
Reduction Problem
all about
latents

ENCODE R

DECODER

w/ new
computer

1 Problem Setup and Mathematical background

2. PyTorch & nn. Linear

3. DEMO: Running the network

4. Math Details:
→ REPARAM TRICK
→ Loss fun.
→ ELBO

5. Comparison with SCVI ones

MAIN MODULE: INFERENCE

So
POSTERIOR
Inference

POSTERIOR

functional
DATASET

= MODEL + DATASET +
+ DATA LOADER +
+ SAMPLER

DATA

POSTERIOR Instance is used as an Attribute, it's like a data set, e.g.
TRAIN-SET OR TEST-SET

MAIN METHODS:

- LL and marginal LL to compute likelihoods // Find Entropy of Batch Mixing
- getting the latent space
- Finding Differential-Expression Statistics and Score, also for cluster
- Batch Mixing Entropy

Additional. Different scores, TSNE, Mixing Entropy, finding Entropy from indices