Research Summary

Yookoon Park

Jan 25th, 2019

Publications

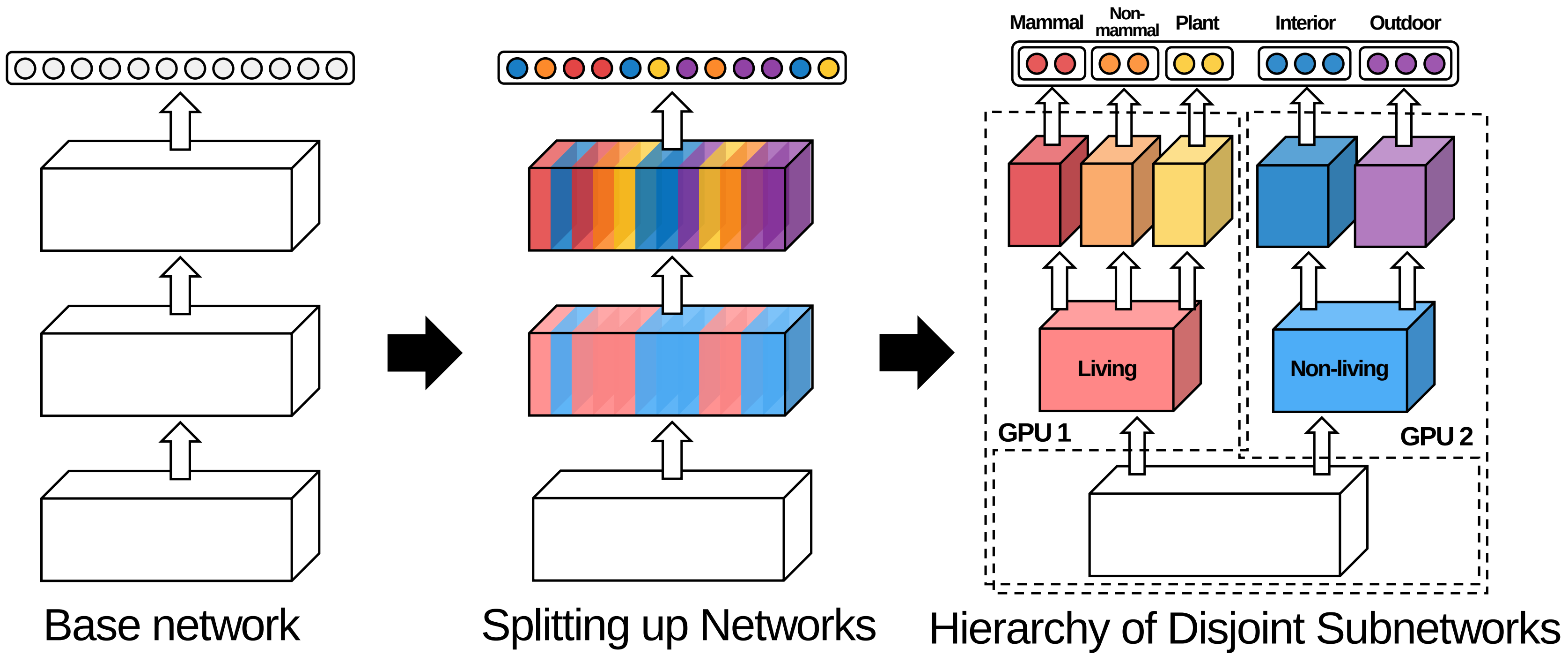
- Yookoon Park, Chris Donjoo Kim, Gunhee Kim. Variational Laplace Autoencoders. In *submission to ICML 2019*.

- Yookoon Park, Jaemin Cho, Gunhee Kim. A hierarchical latent structure for variational conversation modeling. In *NAACL, 2018 (Oral).*

- Yookoon Park*, Juyong Kim*, Gunhee Kim, Sung Ju Hwang. SplitNet: Learning to semantically split deep networks for parameter reduction and model parallelization. *In ICML, 2017*. (* equal contribution)

SplitNet

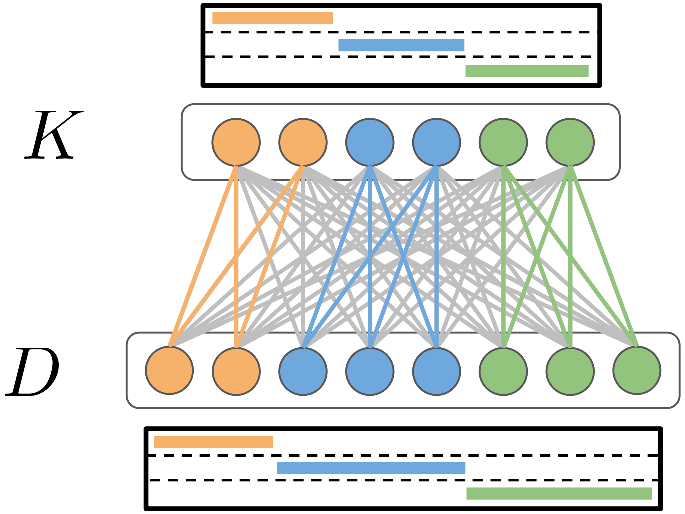
1. Start from a whole base network
2. Split layers into a tree structure through learning
3. Place branches on separate GPUs for efficient parallelization

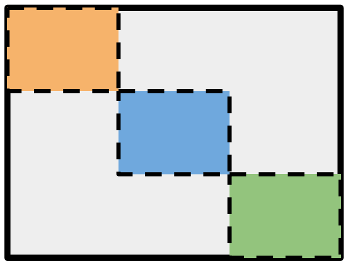


SplitNet

- The problem boils down to learning
  1) Group assignments $\mathbf{P}, \mathbf{Q}$
  2) Corresponding *block-diagonal* weight matrix $\mathbf{W}$

$\boldsymbol{Q} \in \mathbb{R}^{G \times K}$ (class-to-group assignments)





$\boldsymbol{W} \in \mathbb{R}^{D \times K}$

$\boldsymbol{P} \in \mathbb{R}^{G \times D}$ (feature-to-group assignments)

SplitNet

- Regularization objectives
  - Group-sparsity

$$\sum_g \sum_i \left\| \left((\boldsymbol{I} - \boldsymbol{P}_g) \boldsymbol{W} \boldsymbol{Q}_g \right)_{i*} \right\|_2 + \sum_g \sum_j \left\| \left(\boldsymbol{P}_g \boldsymbol{W} (\boldsymbol{I} - \boldsymbol{Q}_g) \right)_{*j} \right\|_2$$

  - Disjoint group assignments

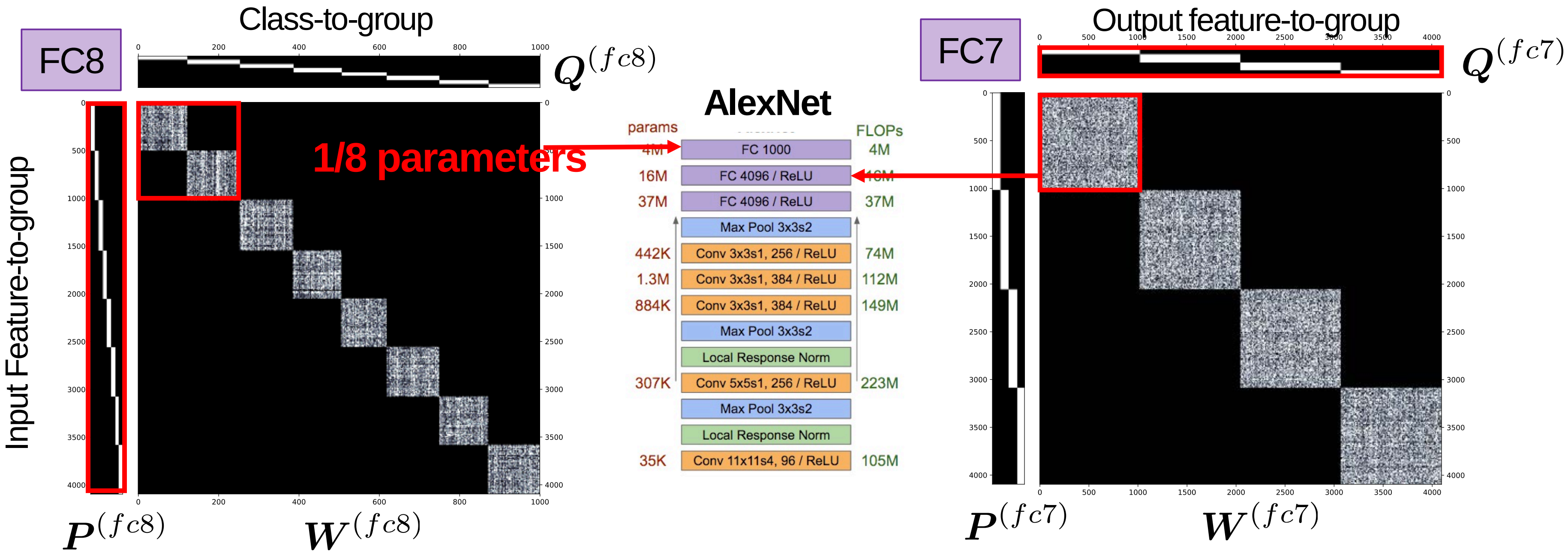
$$\sum_{i<j} \boldsymbol{p}_i \cdot \boldsymbol{p}_j + \sum_{i<j} \boldsymbol{q}_i \cdot \boldsymbol{q}_j$$

  - Balanced group assignments

$$\sum_g \left(\left(\sum_i \boldsymbol{p}_{gi} \right)^2 + \left(\sum_i \boldsymbol{q}_{gj} \right)^2 \right)$$

SplitNet

- Learned weight matrices
  - The weight blocks can be distributed onto separate GPUs



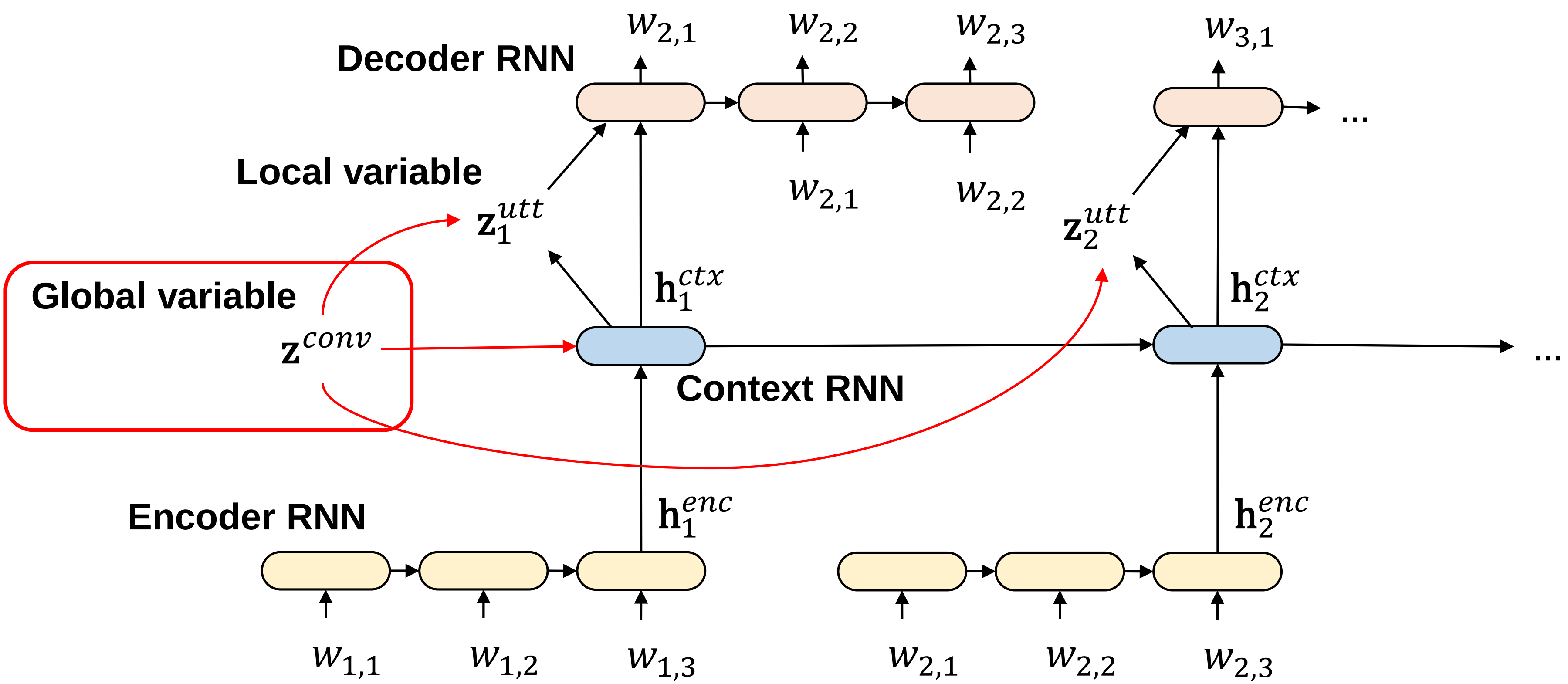
Conversation Modeling Using VAEs

- RNN+VAE suffers from *uninformative latent variables*
  - The model does not use $\boldsymbol{z}$ at all

- Two causes:
  1. Inefficient latent information coding due to variational gap
     - Diagonal Gaussian posterior assumption is too weak
  2. Expressiveness of autoregressive model is powerful
     - Prefer modeling data with autoregressive power

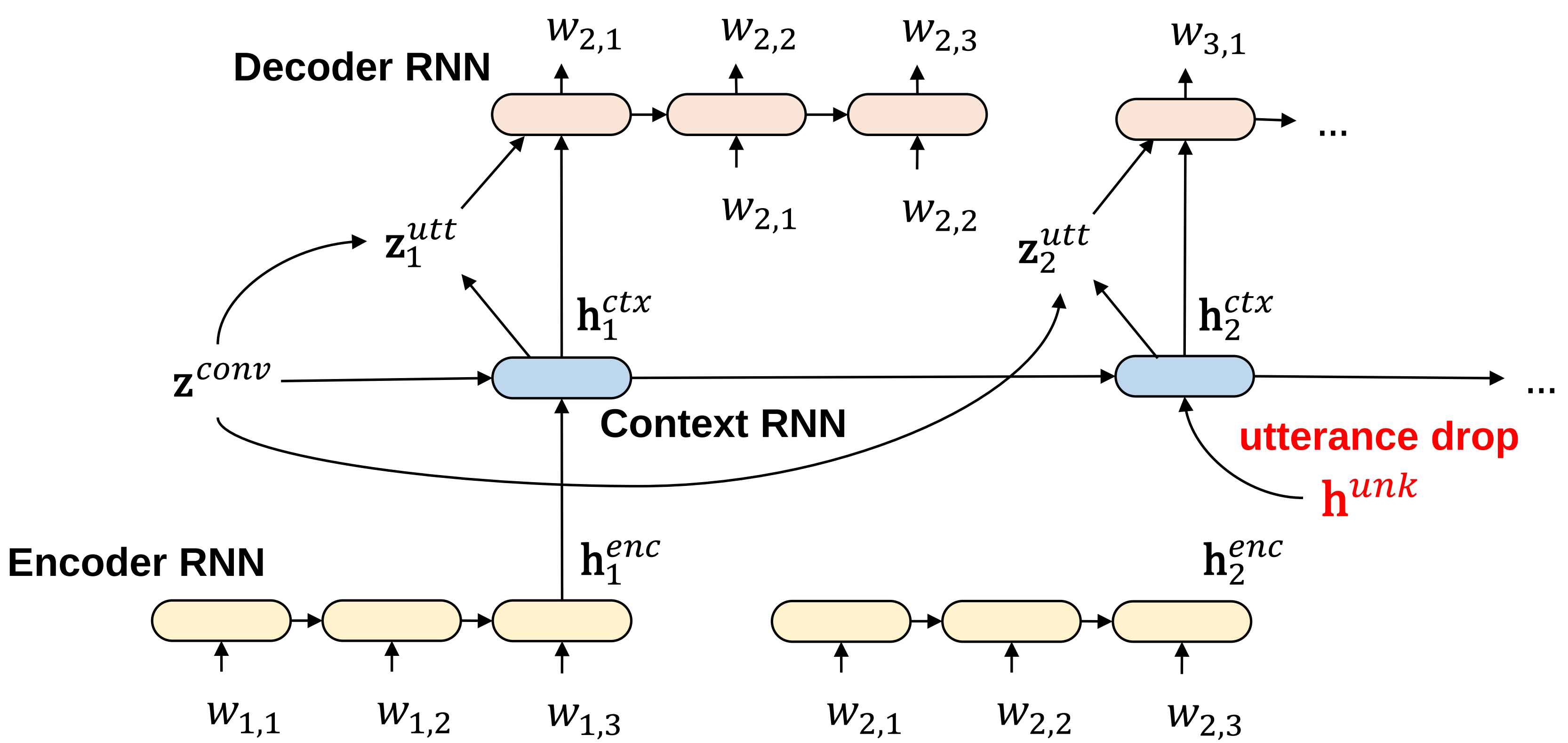
Conversation Modeling Using VAEs

1. Hierarchical latent variable model
    - Efficient coding by sharing global information
    - More flexible posterior approximation



Conversation Modeling Using VAEs

2. Regularize the autoregressive power of RNNs
   - Randomly drop hidden states of encoder RNN



Variational Laplace Autoencoders

- Two challenges for VAEs
    1. Reducing amortization error
    2. Using expressive posterior assumptions

- We tackle both challenges using:
    1. Iterative update for finding the mode of posterior
    2. Local full-covariance Gaussian assumption

- In the paper, we show that this is in principle equivalent to the *Laplace approximation*

Variational Laplace Autoencoders

- Probabilistic PCA
  - A linear Gaussian latent variable model:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}),$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \mathbf{b}, \sigma^2\mathbf{I}),$$

  - Posterior can be calculated in closed-form:

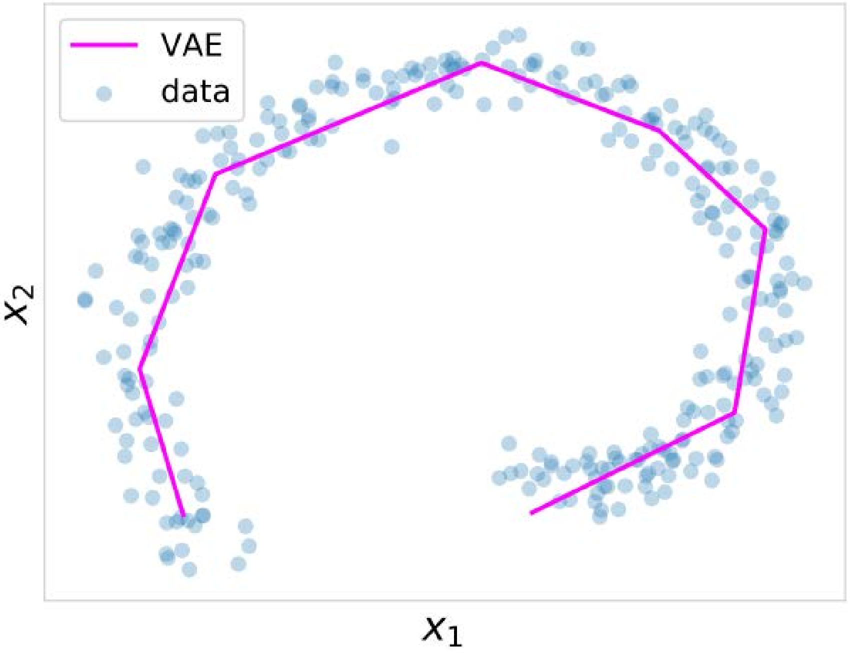
$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\frac{1}{\sigma^2}\boldsymbol{\Sigma}\mathbf{W}^T(\mathbf{x} - \mathbf{b}), \boldsymbol{\Sigma}),$$
$$\text{where } \boldsymbol{\Sigma} = (\frac{1}{\sigma^2}\mathbf{W}^T\mathbf{W} + \mathbf{I})^{-1}.$$

Variational Laplace Autoencoders

- Neural network model:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}),$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(g_{\boldsymbol{\theta}}(\mathbf{z}), \sigma^2 \mathbf{I}),$$

- ReLU networks are *piece-wise linear*
  - It *locally* perform probabilistic PCA
  - $g_{\boldsymbol{\theta}}(\boldsymbol{z}_t) \approx \boldsymbol{W}_t\, \boldsymbol{z}_t + \boldsymbol{b}_t$



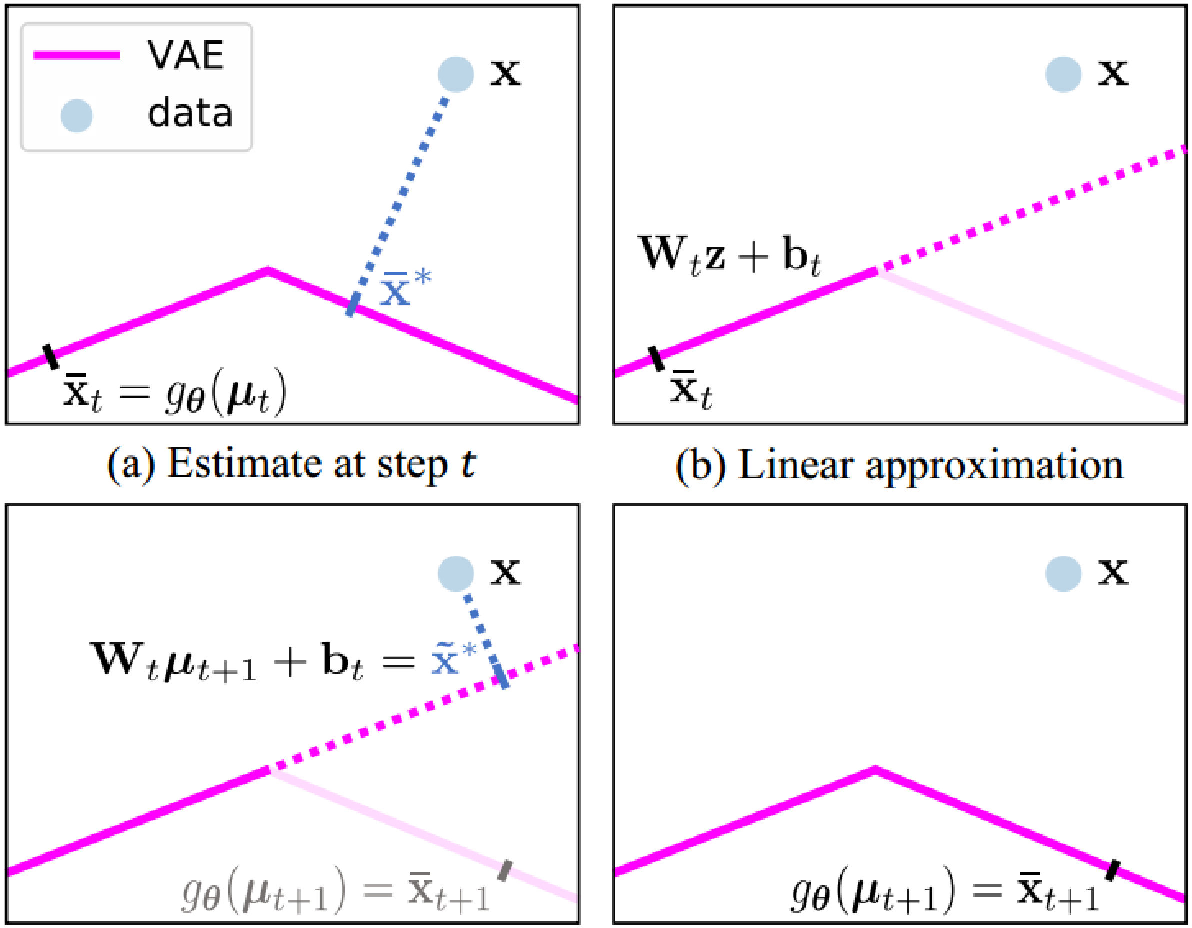
Variational Laplace Autoencoders

- Iteratively find the mode of posterior distribution
  - Using local linearity and the results of probabilistic PCA

Algorithm 1 Iterative Laplace Inference

Input: piece-wise linear generative network $g_{\boldsymbol{\theta}}$, inference model $e_{\boldsymbol{\phi}}$, update steps T, learning rate α_t
Output: approximate Gaussian posterior $q(\mathbf{z}|\mathbf{x})$
Sample $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$
$\boldsymbol{\mu}_0 = e_{\boldsymbol{\phi}}(\mathbf{x})$
for $t = 0$ **to** $T - 1$ **do**
  Compute local linear map $\mathbf{W}_t$ using Eq.(18), (19)
  $\boldsymbol{\Sigma}_t \leftarrow (\sigma^{-2}\mathbf{W}_t^T\mathbf{W}_t + \mathbf{I})^{-1}$
  $\boldsymbol{\mu}' \leftarrow \sigma^{-2}\boldsymbol{\Sigma}_t\mathbf{W}_t^T(\mathbf{x} - \mathbf{b}_t)$
  $\boldsymbol{\mu}_{t+1} \leftarrow (1 - \alpha_t)\boldsymbol{\mu}_t + \alpha_t\boldsymbol{\mu}'$
end for
Compute local linear map $\mathbf{W}_T$ using Eq.(18), (19)
$\boldsymbol{\Sigma}_T \leftarrow (\sigma^{-2}\mathbf{W}_T^T\mathbf{W}_T + \mathbf{I})^{-1}$
$q(\mathbf{z}|\mathbf{x}) \leftarrow \mathcal{N}(\boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T)$
Return $q(\mathbf{z}|\mathbf{x})$

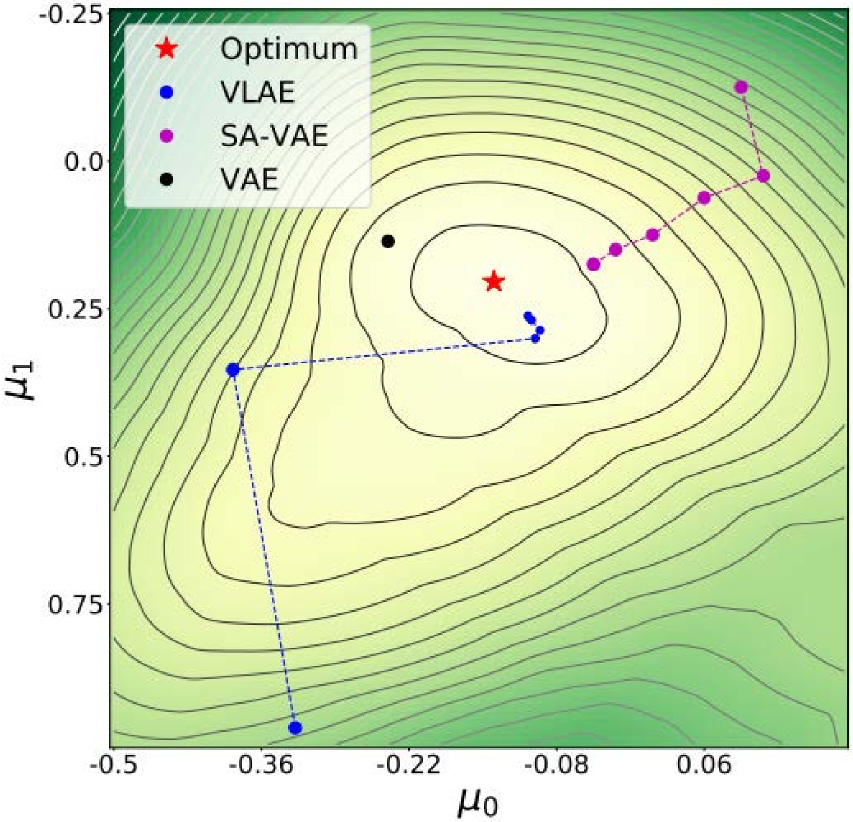
Variational Laplace Autoencoders



(a) Estimate at step t (b) Linear approximation (c) Solution under linearity (d) Updated estimate

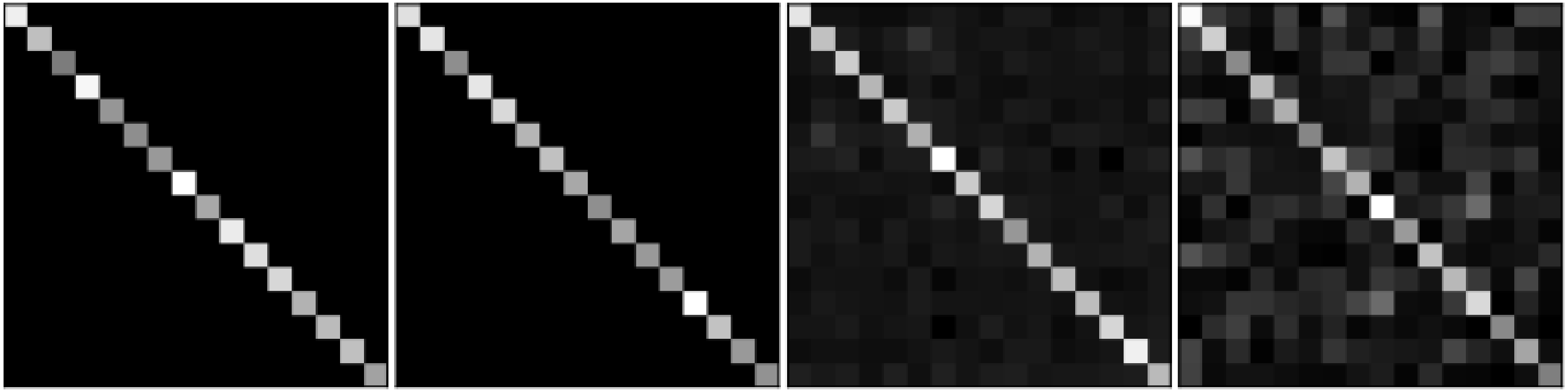
Variational Laplace Autoencoders

- Illustration of updates on ELBO landscape



Variational Laplace Autoencoders

- Expressive posterior with full-covariance Gaussian
  - Compare to fully-factorized assumption of VAE
  - No additional parameters are required



(a) VAE (b) SA-VAE (c) VAE + HF (d) VLAE

Variational Laplace Autoencoders

Table 1. Comparison of the marginal log-likelihood $\log p(\mathbf{x})$ for Gaussian and Bernoulli output models, estimated with 100 importance samples. T refers to the number of updates for VLAE and SA-VAE (Kim et al., 2018) or the number of flows for VAE+HF (Tomczak & Welling, 2016).

	Gaussian					*Bernoulli*
	MNIST	OMNIGLOT	FASHIONMNIST	SVHN	CIFAR10	MNIST
VAE	612.9	343.5	606.3	4555	2364	-96.73
SA-VAE (T=1)	614.1	341.4	606.7	4553	2366	-96.85
SA-VAE (T=2)	615.2	346.6	604.1	4551	2366	-96.73
SA-VAE (T=4)	612.8	348.6	606.6	4553	2366	-96.71
SA-VAE (T=8)	612.1	345.5	608.0	4559	2365	-96.89
VAE+HF (T=1)	610.5	341.5	604.3	4557	2366	-96.75
VAE+HF (T=2)	613.1	343.1	606.5	4569	2361	-96.52
VAE+HF (T=4)	612.9	333.8	604.9	4564	2362	-96.44
VAE+HF (T=8)	615.6	332.6	605.5	4536	2357	-96.14
VLAE (T=1)	638.6	362.0	614.9	4639	2374	-94.68
VLAE (T=2)	645.4	372.7	615.5	4681	2381	-94.46
VLAE (T=4)	649.9	372.3	615.6	4711	2387	**-94.41**
VLAE (T=8)	**650.3**	**380.7**	**618.8**	**4718**	**2392**	-94.57

Recent Interest

- Learning disentangled representations
  - In probabilistic PCA

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\frac{1}{\sigma^2}\boldsymbol{\Sigma}\mathbf{W}^T(\mathbf{x}-\mathbf{b}), \boldsymbol{\Sigma}),$$

$$\text{where } \boldsymbol{\Sigma} = (\frac{1}{\sigma^2}\mathbf{W}^T\mathbf{W} + \mathbf{I})^{-1}.$$

  - Using diagonal $q(\mathbf{z}|\mathbf{x})$ encourage diagonal $\boldsymbol{\Sigma}$
  - As a result, it drives columns of $\mathbf{W}$ to be orthogonal
  - This may be the source of disentanglement in VAEs