(NIPS 2015)

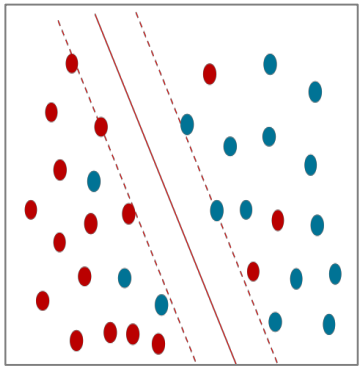
Learning Structured Output Representation Using Deep Conditional Generative Models

Sohn Kihyuk, Honglak Lee, Xinchen Yan

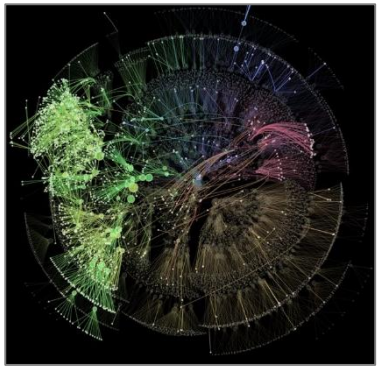
박정수

Data Mining & Information Systems Lab.
Department of Computer Science and Engineering,
College of Informatics, Korea University

1 Motivation



Output Space : 1D

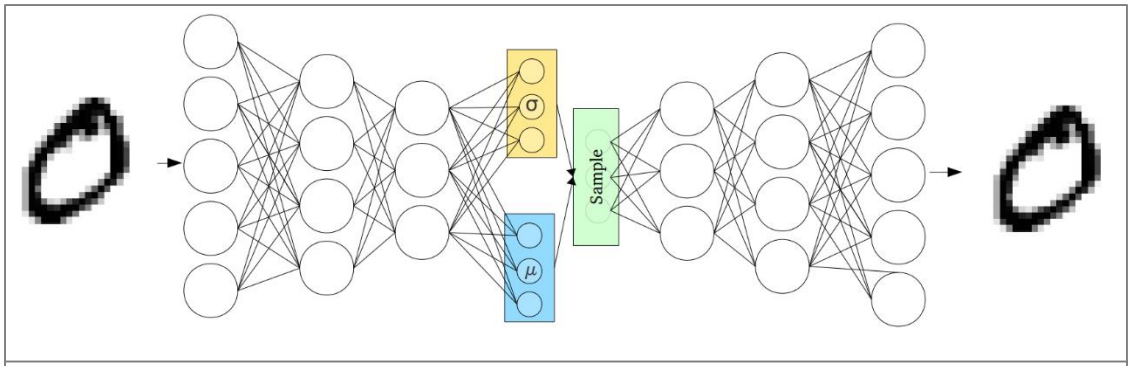


Output Space : High Dimension

- Output representation learning and structure(manifold) prediction required
- Deterministic NN algorithms(i.e. CNN) not suitable for modeling a distribution with multiple modes

2 Recap

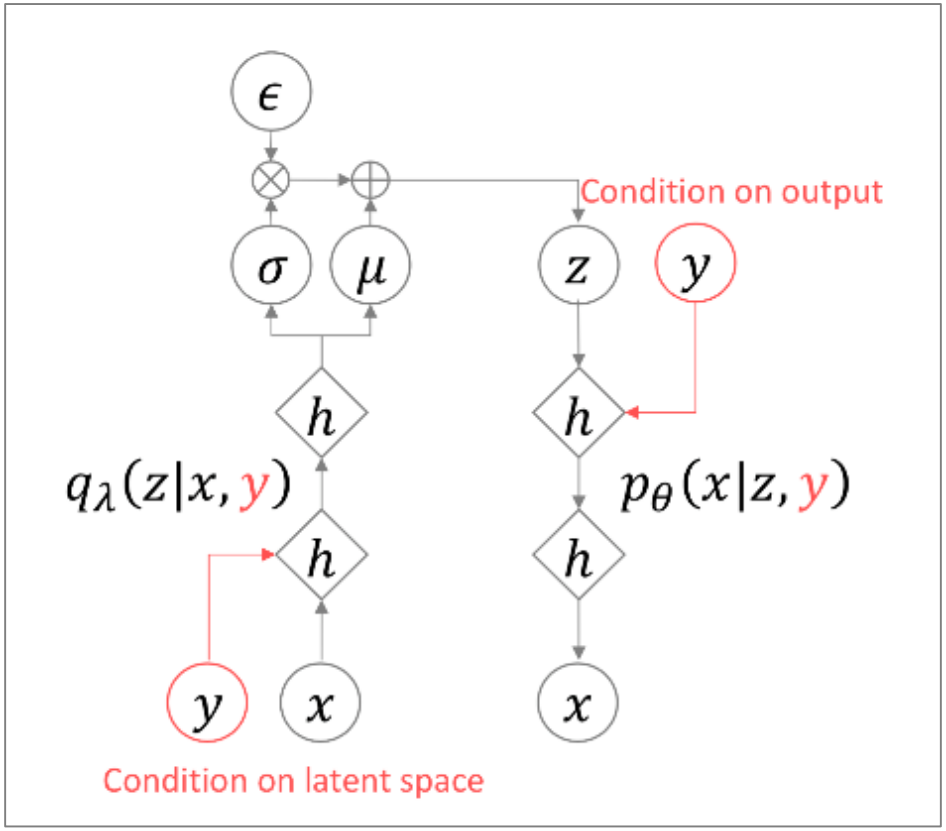
Variational Autoencoder



$$\log p_\theta(\mathbf{x}) = KL\left(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[-\log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_\theta(\mathbf{x},\mathbf{z})\right]$$
$$\geq -KL\left(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] \quad \textbf{ELBO}$$

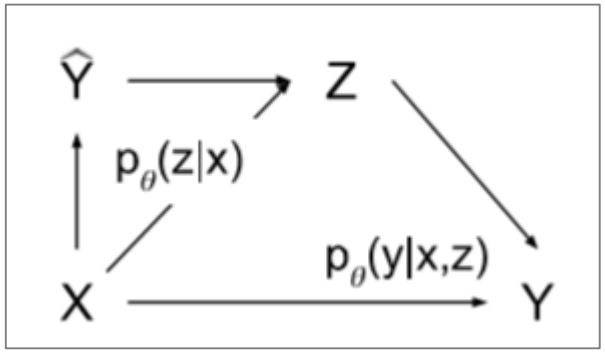
- Using the hypothesis that given data X can be generated from the normal distribution(Z), VAE uses variational approach for latent representation learning in the form of AE
- Another Loss Term(KL Divergence) was introduced as a result of variational approach

3 CVAE

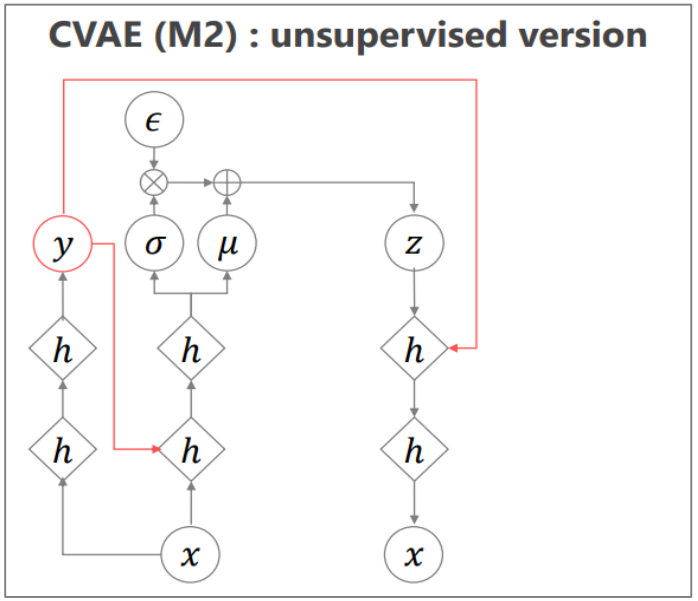


For modeling the high dimensional output space conditioned on the input observation, CVAE was proposed

4 Method



In the case of semi or un-supervised learning, we use the predicted y value as a condition label information



4 Method

$$\log p_\theta(\mathbf{y}|\mathbf{x}) \geq -KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})\|p_\theta(\mathbf{z}|\mathbf{x})\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})\right]$$

$$\widetilde{\mathcal{L}}_{\text{CVAE}}(\mathbf{x},\mathbf{y};\theta,\phi) = -KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})\|p_\theta(\mathbf{z}|\mathbf{x})\right) + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z}^{(l)}),$$

In the phase of training, we used the recognition model, but as for the testing phase, we only use prior distribution.

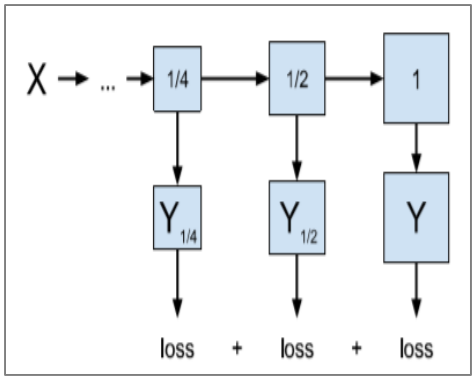
Therefore, we set the prior same as recognition model and derive the loss function as below.

$$\widetilde{\mathcal{L}}_{\text{GSNN}}(\mathbf{x},\mathbf{y};\theta,\phi) = \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z}^{(l)})$$

$$\widetilde{\mathcal{L}}_{\text{hybrid}} = \alpha\widetilde{\mathcal{L}}_{\text{CVAE}} + (1-\alpha)\widetilde{\mathcal{L}}_{\text{GSNN}},$$

4 Method

Multi scale prediction



- Predict output at different scales.
- Being able to achieve global to local prediction of pixel level segmentation

Input Omission noise

- Random block omission noise
- Providing more challenging output prediction task during training thus model becomes more robust

4 Method

As for the evaluation of trained model, we can make a prediction accuracy by performing a deterministic inference

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}^*), \mathbf{z}^* = \mathbb{E}\big[\mathbf{z}|\mathbf{x}\big]$$

Alternatively, we can compare the conditional likelihoods

$$p_\theta(\mathbf{y}|\mathbf{x}) \approx \frac{1}{S}\sum_{s=1}^{S} p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}^{(s)}), \quad \mathbf{z}^{(s)} \sim p_\theta(\mathbf{z}|\mathbf{x})$$

Since MC algorithm requires a large number of samples, we can use importance sampling

$$p_\theta(\mathbf{y}|\mathbf{x}) \approx \frac{1}{S}\sum_{s=1}^{S} \frac{p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}^{(s)})p_\theta(\mathbf{z}^{(s)}|\mathbf{x})}{q_\phi(\mathbf{z}^{(s)}|\mathbf{x}, \mathbf{y})}, \quad \mathbf{z}^{(s)} \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$$

5 Experiment

input

CVAE, epoch 1

CVAE, epoch 20

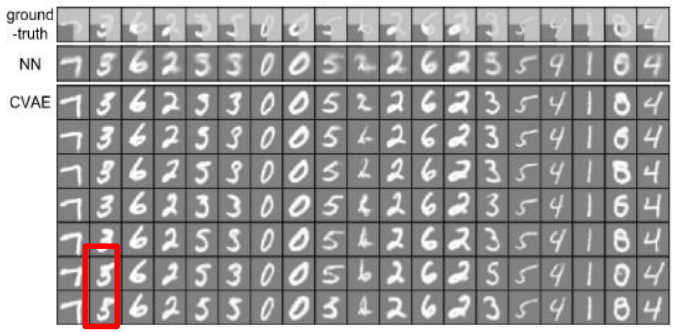
VAE, epoch 1

VAE, epoch 20

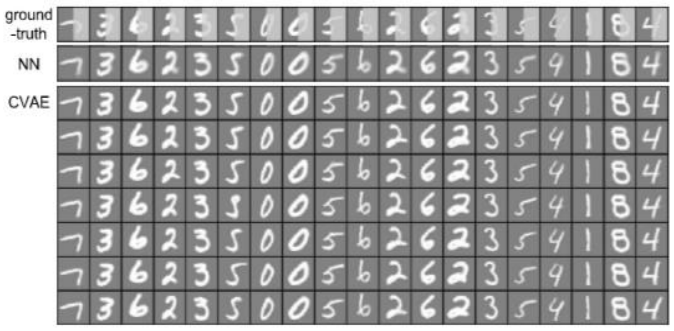
Converges much faster than VAE for the same epoch

5 Experiment

Imputation of MNIST images



1 quadrant

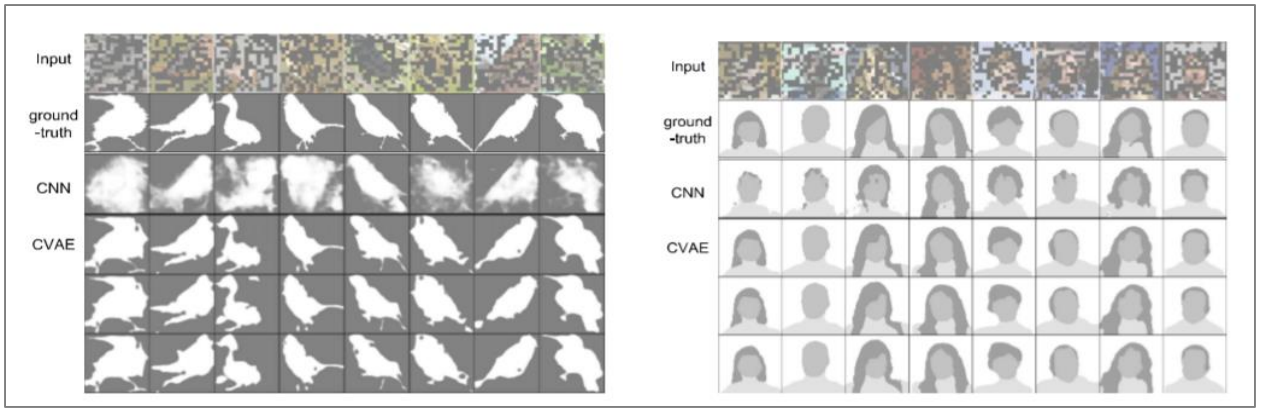


2 quadrant

negative CLL	1 quadrant		2 quadrants		3 quadrants	
	validation	test	validation	test	validation	test
NN (baseline)	100.03	99.75	62.14	62.18	26.01	25.99
GSNN (Monte Carlo)	100.03	99.82	62.48	62.41	26.20	26.29
CVAE (Monte Carlo)	68.62	68.39	45.57	45.34	20.97	20.96
CVAE (Importance Sampling)	**64.05**	**63.91**	**44.96**	**44.73**	**20.97**	**20.95**
Performance gap	35.98	35.91	17.51	17.68	5.23	5.33
- per pixel	0.061	0.061	0.045	0.045	0.027	0.027

5 Experiment

Object segmentation with Partial Observations



Dataset		CUB (IoU)		LFW (pixel)	
noise level	block size	GDNN	CVAE	GDNN	CVAE
25%	1	89.37	**98.52**	96.93	**99.22**
	4	88.74	**98.07**	96.55	**99.09**
	8	90.72	**96.78**	97.14	**98.73**
50%	1	74.95	**95.95**	91.84	**97.29**
	4	70.48	**94.25**	90.87	**97.08**
	8	76.07	**89.10**	92.68	**96.15**
70%	1	62.11	**89.44**	85.27	**89.71**
	4	57.68	**84.36**	85.70	**93.16**
	8	63.59	**76.87**	87.83	**92.06**

Appendix

Proof of ELBO in CVAE

$$
\begin{aligned}
\log p_\theta(\mathbf{y}|\mathbf{x}) &= KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})\|p_\theta(\mathbf{z}|\mathbf{x},\mathbf{y})\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[-\log q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) + \log p_\theta(\mathbf{y},\mathbf{z}|\mathbf{x})\right] && \text{(S1)}\\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[-\log q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) + \log p_\theta(\mathbf{y},\mathbf{z}|\mathbf{x})\right] && \text{(S2)}\\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[-\log q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) + \log p_\theta(\mathbf{z}|\mathbf{x})\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})\right] && \text{(S3)}\\
&= -KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})\|p_\theta(\mathbf{z}|\mathbf{x})\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})\right] && \text{(S4)}
\end{aligned}
$$