Knowledge Extraction with No Observable Data

Jaemin Yoo
Seoul National Univ.
jaeminyoo@snu.ac.kr

Minyong Cho
Seoul National Univ.
chominyong@gmail.com

Taebum Kim
Seoul National Univ.
k.taebum@snu.ac.kr

U Kang
Seoul National Univ.
ukang@snu.ac.kr

Data Mining Laboratory
Seoul National University

Summary

KegNet (Knowledge Extraction with Generative Networks)

- **Input:** a trained neural network M without data
- **Output:** a generator G that estimates unknown p_x
- **Main idea:** G is trained as a function $(y, z) \rightarrow x$
- **GitHub:** https://github.com/snudatalab/KegNet

Knowledge Extraction

Extracting the knowledge of a neural network

- It is intractable to estimate directly $p_x(x)$
  - The size is exponential to $|x|$
  - No prior knowledge is given
- Estimate $p(x|y,z)$ given random variables y and z
  - y is a probability vector representing a label
  - z is a low-dimensional embedding vector of data

Objective functions

- Generate artificial data examples:

Sampling distributions

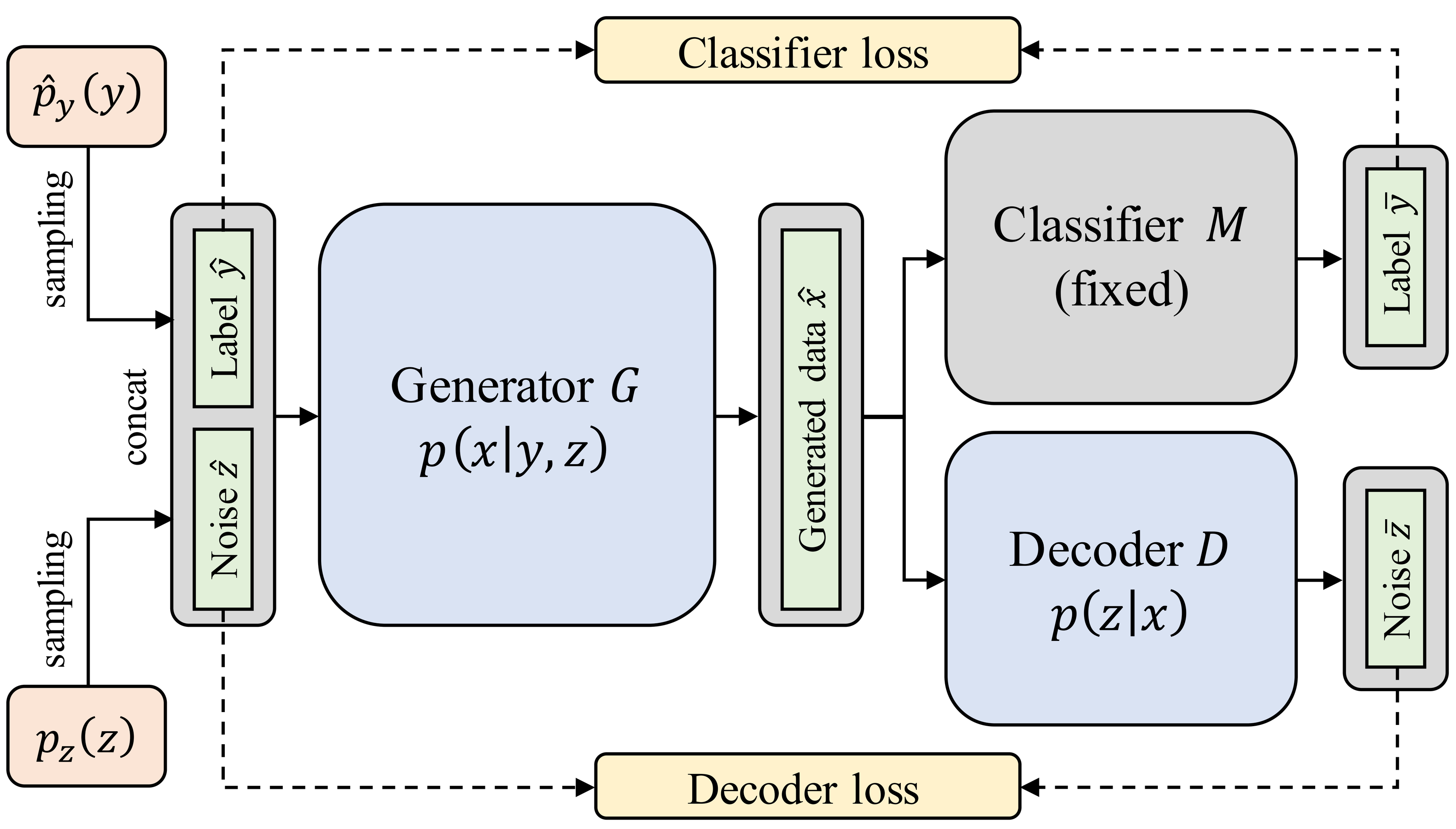
$$\mathcal{D} = \left\{ \underset{\hat{x}}{\operatorname{argmax}}\, p(\hat{x}|\hat{y},\hat{z}) \;\middle|\; \hat{y} \sim \underline{\hat{p}_y(y)} \text{ and } \hat{z} \sim \underline{p_z(z)} \right\}$$

- The argmax function is approximated as follows:

$$\underset{\hat{x}}{\operatorname{argmax}}\, p(\hat{x}|\hat{y},\hat{z}) \approx \underset{\hat{x}}{\operatorname{argmax}} (\log \underline{p(\hat{y}|\hat{x})} + \log p(\hat{z}|\hat{x}))$$

Given network

Proposed Architecture



Classifier M

- Given as an input and fixed
- Our only evidence to estimate the data distribution
- *LeNet4* or *ResNet14* in our experiments

Generator G

- Estimate $p(x|y,z)$ by a generator network
- Its structure is based on *ACGAN* in our experiments
- **Classifier loss** makes $M(G(\hat{y}))$ similar to $\hat{y}$
- Applying G alone, however, generates similar data

Decoder D

- Estimate $p(z|x)$ to find the meaning of $\hat{x}$
- Increase the variance of $\hat{x}$ given the same $\hat{y}$
- **Decoder loss** makes $D(G(\hat{y}))$ similar to $\hat{z}$

Experiments

Data-free model compression

- *To compress a deep neural network without data*
- **Models:** *LeNet5* and *ResNet14* for image classification
- **Datasets:** *MNIST*, *SVHN*, and *Fashion MNIST*
- **Baseline:** *Tucker decomposition* for model compression

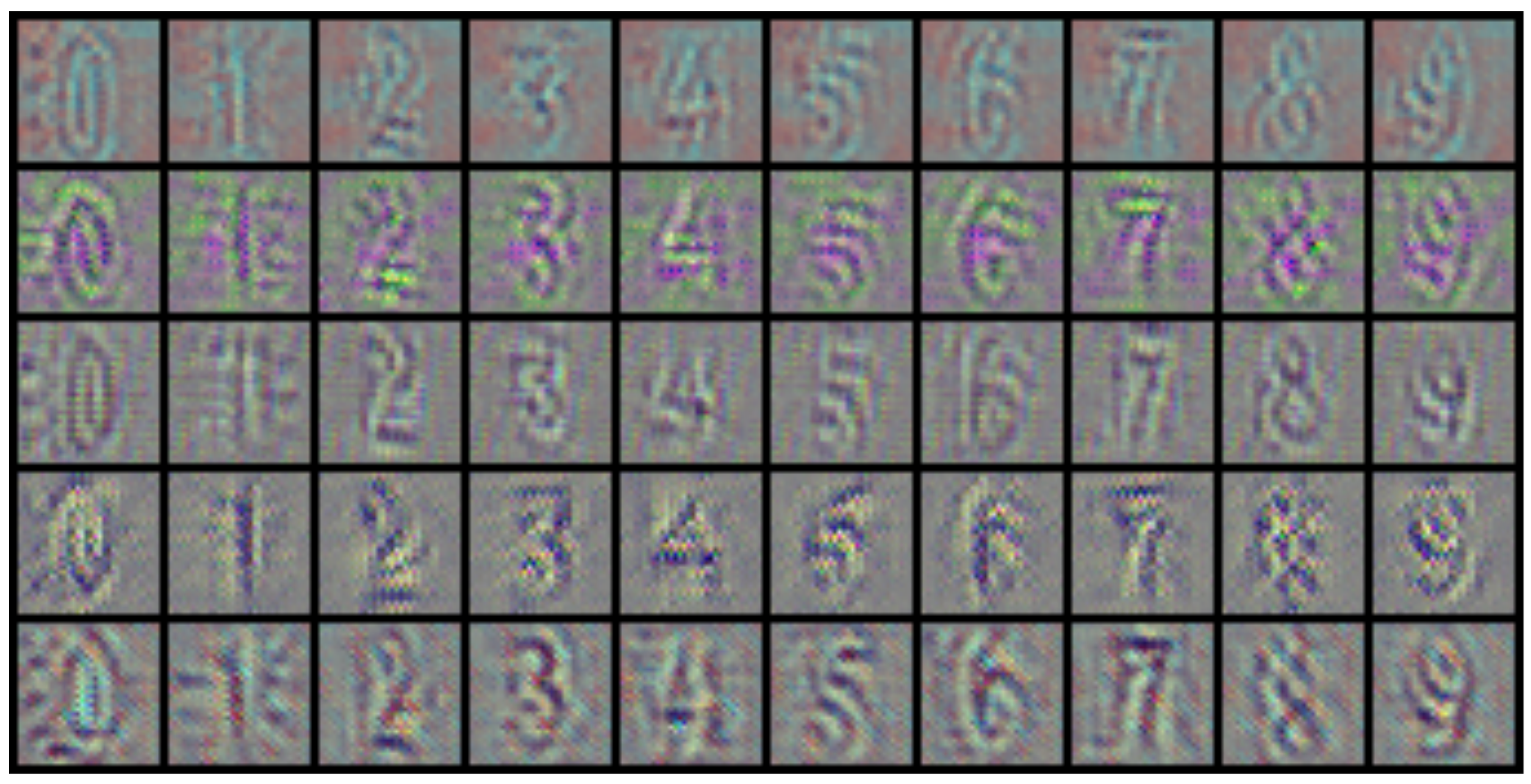
Competitors

- **Original:** the original network M
- **Tucker (T):** Tucker decomposition without fine-tuning
- **T+Uniform:** Estimate p_x as the uniform dist. $\mathcal{U}(-1, 1)$
- **T+Gaussian:** Estimate p_x as the normal dist. $\mathcal{N}(0, 1)$
- **T+KegNet:** Estimate p_x by the generator network G

Classification accuracy & compression ratios

Dataset	Model	Approach	Student 1	Student 2
MNIST	LeNet5	Original	98.90%	98.90%
MNIST	LeNet5	Tucker (T)	85.18% (3.62×)	67.35% (4.10×)
MNIST	LeNet5	T+Uniform	95.48 ± 0.11%	88.27 ± 0.07%
MNIST	LeNet5	T+Gaussian	95.45 ± 0.15%	87.70 ± 0.12%
MNIST	LeNet5	T+KegNet	**96.32 ± 0.05%**	**90.89 ± 0.11%**
SVHN	ResNet14	Original	93.23%	93.23%
SVHN	ResNet14	Tucker (T)	19.31% (1.44×)	11.02% (1.65×)
SVHN	ResNet14	T+Uniform	33.08 ± 1.47%	63.08 ± 1.77%
SVHN	ResNet14	T+Gaussian	26.58 ± 1.61%	60.22 ± 4.17%
SVHN	ResNet14	T+KegNet	**69.89 ± 1.24%**	**87.26 ± 0.46%**
Fashion	ResNet14	Original	92.50%	92.50%
Fashion	ResNet14	Tucker (T)	65.09% (1.40×)	75.80% (1.58×)
Fashion	ResNet14	T+Uniform	< 65.09%	< 75.80%
Fashion	ResNet14	T+Gaussian	< 65.09%	< 75.80%
Fashion	ResNet14	T+KegNet	**85.23 ± 1.36%**	**87.80 ± 0.31%**

Generated images for SVHN from 5 generators



Latent space walking from 0 to 5 in SVHN