KeNet (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:
  $$D = \{ \text{argmax} \ p(x|y, z) \ | \ y \sim \hat{p}_y(y) \text{ and } z \sim p_z(z) \}$$
- The argmax function is approximated as follows:
  $$\text{argmax} \ p(x|y, z) \approx \text{argmax} (\log p(y|x) + \log p(z|x))$$

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
- **Classifer loss** makes $M(G(y))$ similar to $y$
- 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 $\hat{z}$

**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+KeNet**: Estimate $p_x$ by the generator network $G$