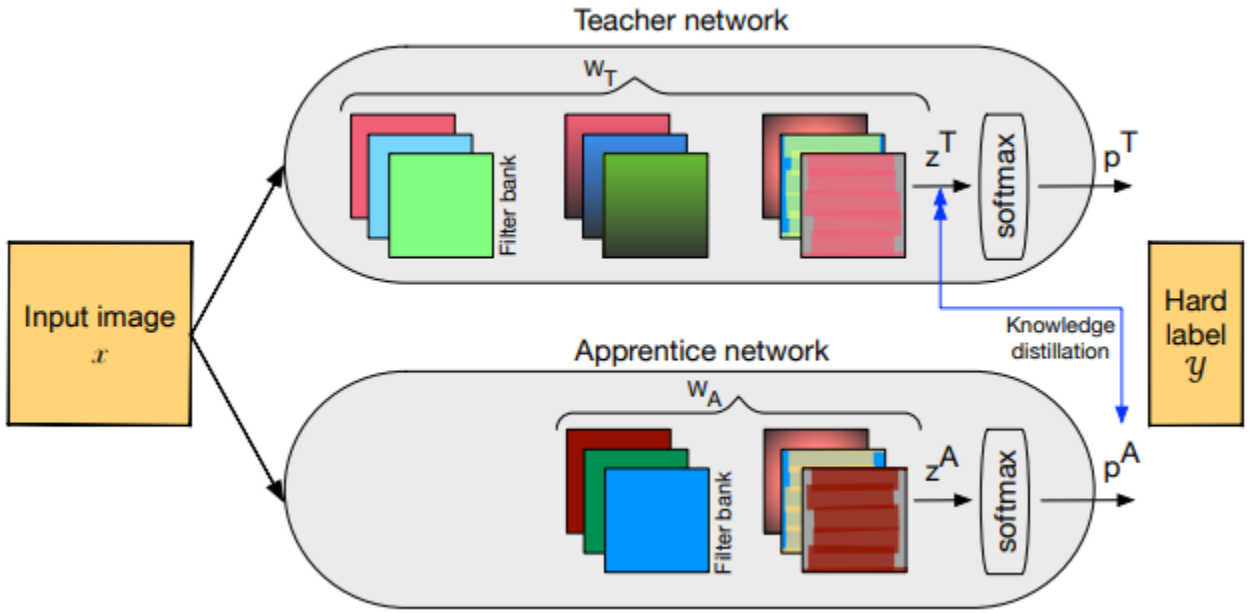
Data-Free Learning of Student Networks

- Most existing deep neural network compression and speed-up methods are very effective for training compact deep models, when we can directly access the training dataset



如果没有数据怎么办

既然学生网络是学习同一输入下教师网络的输出，那么我们没有origin data作为输入的时候，我们是否可以创造一批数据用于输入呢？
常规的想法是生成一批高斯分布的输入，在这上面学生网络去拟合教师网络。
但是实际上，我们的神经网络会拥有较高的建模能力，这导致了网络对高斯分布数据的拟合不会影响到正常数据上的泛化性能。

Table 3. Classification result on the CIFAR dataset.

Algorithm	Required data	FLOPS	#params	CIFAR-10	CIFAR-100
Teacher	Original data	~1.16G	~21M	95.58%	77.84%
Standard back-propagation	Original data	~557M	~11M	93.92%	76.53%
Knowledge Distillation [8]	Original data	~557M	~11M	94.34%	76.87%
Normal distribution	No data	~557M	~11M	14.89%	1.44%
Alternative data	Similar data	~557M	~11M	90.65%	69.88%
Data-Free Learning (DAFL)	No data	~557M	~11M	92.22%	74.47%

Related Works

- 《Data-free knowledge distillation for deep neural networks》
  利用meta data进行蒸馏
- 《Zero-Shot Knowledge Distillation in Deep Networks》
  和此文差不多的思路，效果会差一点

GAN for Generating Training Samples

- Gan optimizing problem： $G^* = \arg\min_G \mathbb{E}_{z \sim p_z(z)}[\log(1 - D^*(G(z)))],$
- 因为训练D需要origin data，然而我们拿不到，所以我们用 teacher model 代替 D，我们试图在T上找到几个合适的点来拟合 teacher的输入分布
- 1：one-hot loss $\mathcal{L}_{oh} = \frac{1}{n}\sum_i \mathcal{H}_{cross}(\mathbf{y}_T^i, \mathbf{t}^i),$
- 2：activation loss $\mathcal{L}_a = -\frac{1}{n}\sum_i \|f_T^i\|_1,$
- 3：information entropy loss： $\mathcal{L}_{ie} = -\mathcal{H}_{info}(\frac{1}{n}\sum_i \mathbf{y}_T^i).$

$$\mathcal{L}_{Total} = \mathcal{L}_{oh} + \alpha\mathcal{L}_a + \beta\mathcal{L}_{ie},$$

Algorithm

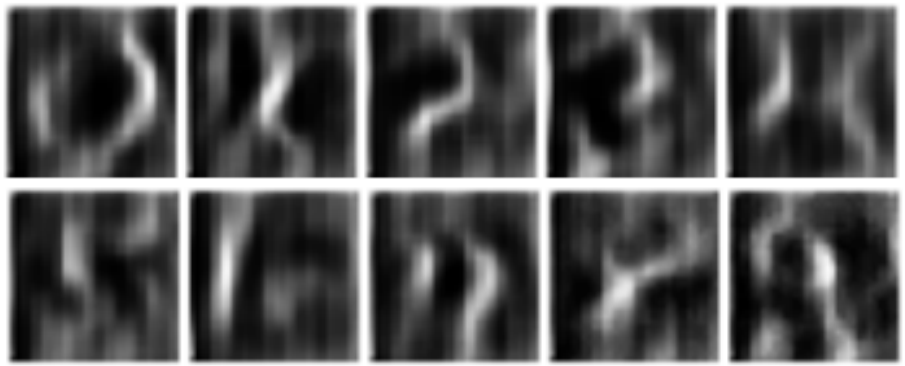
Algorithm 1 DAFL for learning portable student networks.

Input: A given teacher network $\mathcal{N}_T$, parameters of different objects: α and β.

Initialize the generator G, the student network $\mathcal{N}_S$ with fewer memory usage and computational complexity;
repeat
  Module 1: Training the Generator.
  Randomly generate a batch of vector: $\{\mathrm{z}^i\}_{i=1}^n$;
  Generate the training samples: $x \leftarrow G(\mathrm{z})$;
  Employ the teacher network on the mini-batch:
  $$[\mathrm{y}_T, \mathrm{t}, f_T] \leftarrow \mathcal{N}_T(\mathrm{x});$$
  Calculate the loss function $\mathcal{L}_{Total}$ (Fcn.7):
  Update weights in G using back-propagation;
  Module 2: Training the student network.
  Randomly generate a batch of vector $\{\mathrm{z}^i\}_{i=1}^n$;
  Utlize the generator on the mini-batch: $x \leftarrow G(\mathrm{z})$;
  Employ the teacher network and the student network on the mini-batch simultaneously:
  $$\mathrm{y}_S \leftarrow \mathcal{N}_S(\mathrm{x}), \mathrm{y}_T \leftarrow \mathcal{N}_T(\mathrm{x});$$
  Calculate the knowledge distillation loss:
  $$\mathcal{L}_{KD} \leftarrow \frac{1}{n}\sum_i \mathcal{H}(\mathrm{y}_S^i, \mathrm{y}_T^i);$$
  Update weights in $\mathcal{N}_S$ according to the gradient;
until convergence

Output: The student network $\mathcal{N}_S$.

(a) Averaged images on the MNIST dataset.



(b) Averaged images on the generated dataset.

可能存在的改进

$$\mathcal{L}_{ie} = -\mathcal{H}_{info}(\frac{1}{n}\sum_{i} \mathrm{y}_T^i).$$

这个需要比较大的batchsize才可行
实际上，我们可以参考acgan的做法，
指定gan生成的lable，然后one-hot loss 中对你指定的label算crossentryloss就可以了。