AET vs. AED: Unsupervised Representation Learning by Auto-Encoding Transformations rather than Data

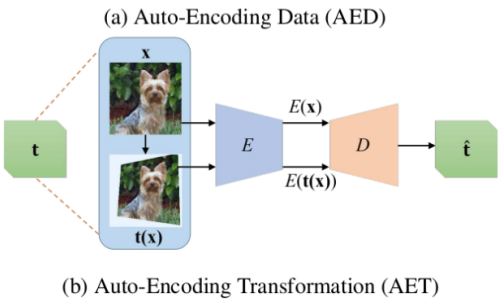
(2019)

Liheng Zhang, Guo-Jun Qi, Liqiang Wang, Jiebo Luo

Notes

Contributions

The authors propose a new self-supervised algorithm, that consists in auto-encoding transformations (AET) instead of data (AED). First some operation to transform the images are samples, these operator are then applied to training images, both the original and transformed images are fed into an encoder, the resulting features are forwarded into a decoder and the objective is to decode the transformation from the intermediate features, and as long as the trained features are sufficiently informative, these transformations can be decoded from features that well encode visual structures of images. And since the prediction on the input transformation is made through the encoded features rather than the original and transformed images, it forces the model to extract expressive features as a proxy to represent images.



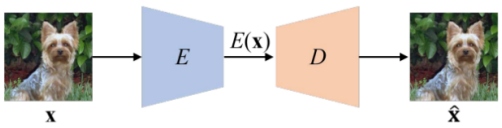


Figure 1: An illustrative comparison between AED and AET, where AET attempts to estimate the input transformation rather than the data at the output end. This forces the encoder network E to extract the features that contain the sufficient information about visual structures to decode the input transformation.

Method

Training We first sample a transformation t from a distribution $\mathcal{T}$, and an example x from the data distribution $\mathcal{X}$, we forward both the image x and its transformed version $t(x)$ into the encoder E, and obtain two encoded representations $E(x)$ and $E(t(x))$. The decoder then uses these representations to predict the transformations applied to the input $\hat{\mathbf{t}}$. Both the encoder and the decoder are trained jointly:

$$\min_{E,D} \mathbf{t}\sim\mathcal{T},\mathbf{x}\sim\mathcal{X}\, \ell(\mathbf{t},\hat{\mathbf{t}}), \text{ and } \hat{\mathbf{t}} = D[E(\mathbf{x}), E(\mathbf{t}(\mathbf{x}))]$$

Family of transformations

- Parameterized Transformations: The family of transformations that can be specified by their parameters θ, such as affine and projective transformations that can be parameterized by a matrix $M(\boldsymbol{\theta}) \in \mathbb{R}^{3\times3}$ relating the input and transformed image, the loss is this case is a simple

MSE $\ell\left(\mathbf{t}_{\boldsymbol{\theta}}, \mathbf{t}_{\hat{\boldsymbol{\theta}}}\right) = \frac{1}{2}\|M(\boldsymbol{\theta}) - M(\hat{\boldsymbol{\theta}})\|_2^2$ to model the difference between the target and estimated transformation.

- GAN-Induced Transformations: We can also use a local generator G, that takes a randomly sampled noise z and an input image and produces a transformed version $\mathbf{t}_{\mathbf{z}}(\mathbf{x}) = G(\mathbf{x}, \mathbf{z})$, the decoder in this case will predict the randomly sampled noise z, and the loss is: $\ell\left(\mathbf{t}_{\mathbf{z}}, \mathbf{t}_{\hat{\mathbf{z}}}\right) = \frac{1}{2}\|\mathbf{z} - \hat{\mathbf{z}}\|_2^2$
- Non-Parametric Transformations: Even if a transformation $\mathbf{t} \in \mathcal{T}$ is hard to parameterize, we can still define the loss $\ell(\mathbf{t}, \hat{\mathbf{t}})$ by measuring the average difference between the transformations of randomly sampled images: $\ell(\mathbf{t}, \hat{\mathbf{t}}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{X}} \text{dist}(\mathbf{t}(\mathbf{x}), \hat{\mathbf{t}}(\mathbf{x}))$, in this case $\hat{\mathbf{t}}$ is approximated by a parameterized transformation $\mathbf{t}_{\hat{\boldsymbol{\theta}}}$ that the decoder outputs.

Results

AET network has two NIN branches, each taking the original and the transformed images as its input. The output features of the forth block of two branches are concatenated and average-pooled to form a 384-d feature vector. Then an output layer follows to predict the parameters of input transformation. The AET networks are trained by SGD with a batch size of 512 original images and their transformed counterparts. Momentum and weight decay are set to 0.9 and 5×10^{-4}. The learning rate is initialized to 0.1 and scheduled to drop by a factor of 5 after 240, 480, 640, 800 and 1, 000 epochs.

Table 1: Comparison between unsupervised feature learning methods on CIFAR-10. The fully supervised NIN and the random Init. + conv have the same three-block NIN architecture, but the first is fully supervised while the second is trained on top of the first two blocks that are randomly initialized and stay frozen during training.

Method	Error rate
Supervised NIN (Lower Bound)	7.20
Random Init. + conv (Upper Bound)	27.50
Roto-Scat + SVM [21]	17.7
ExamplarCNN [7]	15.7
DCGAN [25]	17.2
Scattering [20]	15.3
RotNet + FC [10]	10.94
RotNet + conv [10]	8.84
(Ours) AET-affine + FC	9.77
(Ours) AET-affine + conv	8.05
(Ours) AET-project + FC	**9.41**
(Ours) AET-project + conv	**7.82**

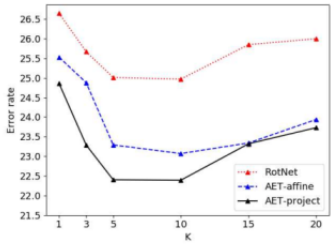


Figure 3: The comparison of the KNN error rates by different models with varying numbers K of nearest neighbors on CIFAR-10.

Table 2: Comparison of RotNet vs. AETs on CIFAR-10 with different classifiers on top of learned representations for evaluation. The RotNet is chosen as the baseline since it has the exactly same architecture for the unsupervised training. Here n-FC denotes a n-layer fully connected (FC) classifier, and the KNN is obtained with $K = 10$ nearest neighbors. The numbers in parentheses are the *relative* reduction in error rates w.r.t. the RotNet baseline.

	KNN	1-FC	2-FC	3-FC	conv
RotNet baseline [10]	24.97	18.21	11.34	10.94	8.84
AET-affine	23.07 (↓7.6%)	17.16 (↓ 5.8%)	9.77 (↓ 13.8%)	10.16 (↓ 7.1%)	8.05(↓ 8.9%)
AET-project	**22.39** (↓ 10.3%)	**16.65** (↓ 8.6%)	**9.41** (↓ 17.0%)	**9.92** (↓ 9.3%)	**7.82**(↓ 11.5%)

Method	Conv4	Conv5
ImageNet Labels [3](Upper Bound)	59.7	59.7
Random [19] (Lower Bound)	27.1	12.0
Tracking [28]	38.8	29.8
Context [5]	45.6	30.4
Colorization [30]	40.7	35.2
Jigsaw Puzzles [18]	45.3	34.6
BiGAN [6]	41.9	32.2
NAT [3]	-	36.0
DeepCluster [4]	-	44.0
RotNet [10]	50.0	43.8
(Ours) AET-project	**53.2**	**47.0**

Table 3: Top-1 accuracy with non-linear layers on ImageNet. AlexNet is used as backbone to train the unsupervised models. After unsupervised features are learned, nonlinear classifiers are trained on top of Conv4 and Conv5 layers with labeled examples to compare their performances. We also compare with the fully supervised models and random models that give upper and lower bounded performances. For a fair comparison, only a single crop is applied in AET and no dropout or local response normalization is applied during the testing.

Table 4: Top-1 accuracy with linear layers on ImageNet. AlexNet is used as backbone to train the unsupervised models under comparison. A $1,000$-way linear classifier is trained upon various convolutional layers of feature maps that are spatially resized to have about $9,000$ elements. Fully supervised and random models are also reported to show the upper and the lower bounds of unsupervised model performances. Only a single crop is used and no dropout or local response normalization is used during testing for the AET, except the models denoted with * where ten crops are applied to compare results.

Method	Conv1	Conv2	Conv3	Conv4	Conv5
ImageNet Labels (Upper Bound) [10]	19.3	36.3	44.2	48.3	50.5
Random (Lower Bound)[10]	11.6	17.1	16.9	16.3	14.1
Random rescaled [16](Lower Bound)	17.5	23.0	24.5	23.2	20.6
Context [5]	16.2	23.3	30.2	31.7	29.6
Context Encoders [22]	14.1	20.7	21.0	19.8	15.5
Colorization[30]	12.5	24.5	30.4	31.5	30.3
Jigsaw Puzzles [18]	18.2	28.8	34.0	33.9	27.1
BiGAN [6]	17.7	24.5	31.0	29.9	28.0
Split-Brain [29]	17.7	29.3	35.4	35.2	32.8
Counting [19]	18.0	30.6	34.3	32.5	25.7
RotNet [10]	18.8	31.7	38.7	38.2	36.5
(Ours) AET-project	**19.2**	**32.8**	**40.6**	**39.7**	**37.7**
DeepCluster* [4]	13.4	32.3	41.0	39.6	38.2
(Ours) AET-project*	**19.3**	**35.4**	**44.0**	**43.6**	**42.4**