Generative Adversarial Stacked Autoencoders

Ariel Ruiz-Garcia*, Ibrahim Almakky•, Vasile Palade•, Luke Hicks•

*LatinX in AI and SeeChange.ai

•Data Science Centre, Coventry University, UK

Motivation

- Generative Adversarial Networks (GANs) are predominant in image generation tasks.
- Difficult to train:
  - Prone to vanishing gradients.
  - Prone to non-convergence.
  - Sensitive to hyperparameter and parameter initialization.
- The Generator model is prone to mode collapse:
  - Often results in generated images with little variation.
- Hardly ever used for pre-training tasks.

Generative Adversarial Autoencoders

Vanilla Autoencoders:

$$y = g(f(x))$$

GAN Autoencoders:

Given a data distribution $p_d(x_\phi)$ and target data distribution $p_d(x_\mu)$:

- G maps x_ϕ to latent space z and back to reconstruction y that resembles x_μ and lies in $q(y)$.
- Impose $p_d(x_\mu)$ on $q(y)$:

$$q(y) = \mathbb{R}_{x_\mu} q(x|\mu) p_d(x_\phi) dx_\mu$$

- G tries to fool the discriminator that y came from $p_d(x_\mu)$ and not from $q(y)$:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D\left(x_\mu^{(i)}\right) + \log(1 - D\left(G\left(y^{(i)}\right)\right))]$$

- D rates samples from $p_d(x_\mu)$ with a higher probability and samples from $q(y)$ with a low probability:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D\left(x_\mu^{(i)}\right) + \log(1 - D\left(G\left(y^{(i)}\right)\right))]$$

Generative Adversarial Stacked Autoencoders

- For a given GAN Autoencoder:
  - Decompose the Generator model into smaller shallow autoencoders.
  - Decompose the discriminator into one-layer classifiers.
- Train one shallow adversarial autoencoder at a time.
  - Place in corresponding stack.
  - Fine-tune entire stack.
  - Repeat until done.

Formally defined as:

$[G^1, D^1] \leftarrow L(G^1, D^1, X, X^{\sim})$
$D^{(k)} \circ D$
$D^{(k)} \circ D$
$\boldsymbol{for}\ k = 2, \dots, m\ \boldsymbol{do}$
  $[\xi, \delta] \leftarrow D$
  $[X_g, X_{\tilde{g}}] \leftarrow \xi(X, X^{\sim})$
  $[G^k, D^k] \leftarrow L\left(G^k, D^k, X_d, X_{\tilde{d}}\right)$
  $G \leftarrow G^{(k)} \circ G$
  $D \leftarrow D^{(k)} \circ D$
  $G \leftarrow T(G, X, X^{\sim})$
  $X_\varphi \leftarrow G\left(x_\varphi\right)$
  $D \leftarrow T\left(D, \{X_\varphi, x_\mu \subset X\}\right)$
$\boldsymbol{end\ for}$
$\boldsymbol{return}\ G, D$

Main advantages over vanilla autoencoders:

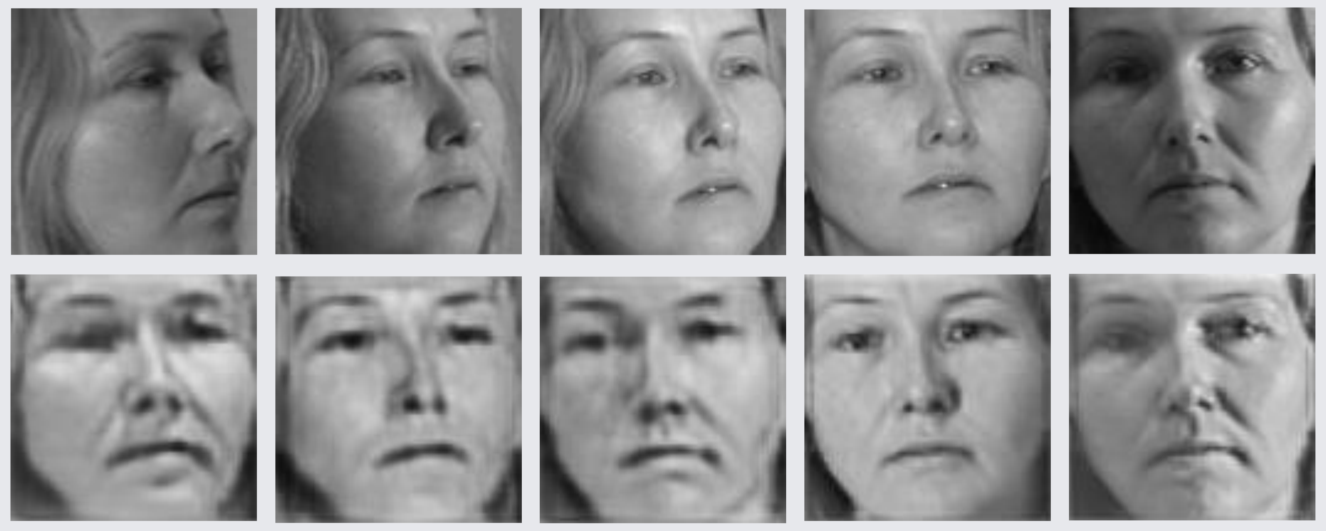
- Better reconstructions

Main advantages over GAN Autoencoders:

- Reduced error accumulation.
- Better reconstructions.
- Significantly faster to train ~1 tenth of GPU time.
- Less sensitive to parameter initialization.
- Can be used as a pre-training method.
- Allows for reconstruction targets to be different than inputs.

Reconstruction Results

Training to reconstruct profile images as frontal images (CK+ dataset):



Training to improve illumination (Yale Faces Database):

Pre-training Classifiers

Pre-training a classifier as a GAN Stacked Autoencoder (where the classifier is used as the Generator (tested on KDEF dataset):

	ResNet34	CNN (pose)	CNN (pose and illumination)
Angry	84.127	94.444	**96.825**
Disgust	85.600	**97.600**	**97.600**
Fear	73.810	89.683	**93.651**
Happy	98.413	**100.00**	**100.00**
Neutral	90.400	**100.00**	**100.00**
Sad	84.921	**98.413**	**98.413**
Surprise	95.161	97.571	**98.070**