Deep Learning with R

Generative Adversarial Networks, Autoencoders

Mikhail Dozmorov

Virginia Commonwealth University

2020-06-12

Generative adversarial networks (GANs)

> The most important [recent development], in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks). This is an idea that was originally proposed by Ian Goodfellow when he was a student with Yoshua Bengio at the University of Montreal (he since moved to Google Brain and recently to OpenAI).
>
> This, and the variations that are now being proposed, is the most interesting idea in the last 10 years in ML, in my opinion.
>
> Yann LeCun

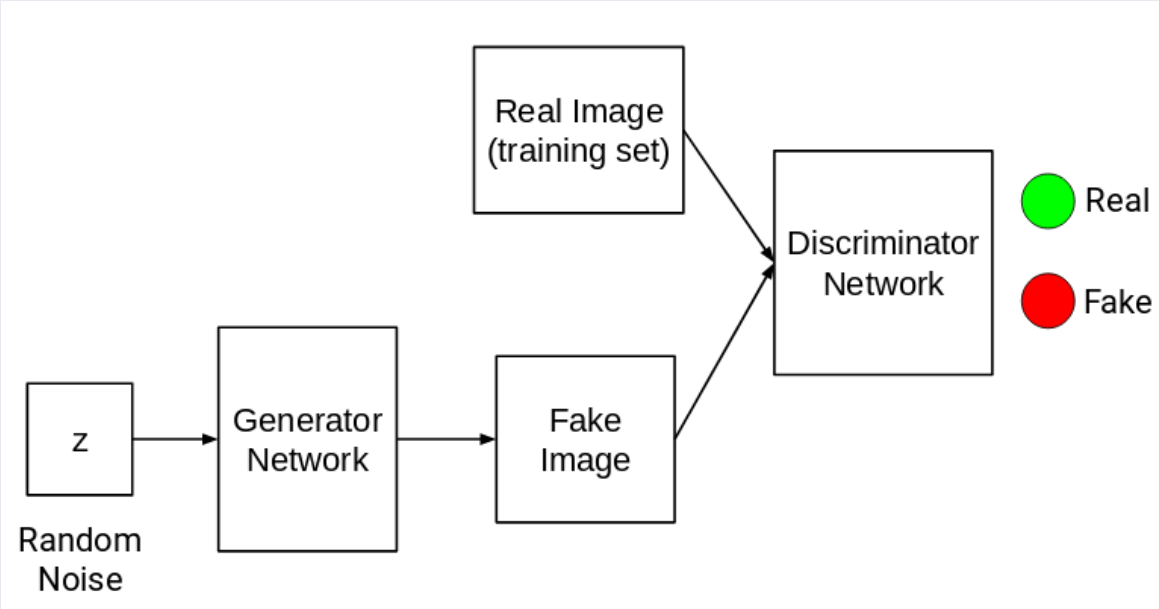
https://danieltakeshi.github.io/2017/03/05/understanding-generative-adversarial-networks/

Generative adversarial networks (GANs)

- Unsupervised learning models that aim to generate data points that are indistinguishable from the observed ones
- Aim to learn the data-generating process
- GANs were proposed as a radically different approach to generative modeling that involves two neural networks, a discriminator and a generator network
- They are trained jointly, whereby the generator aims to generate realistic data points, and the discriminator classifies whether a given sample is real or generated by the generator

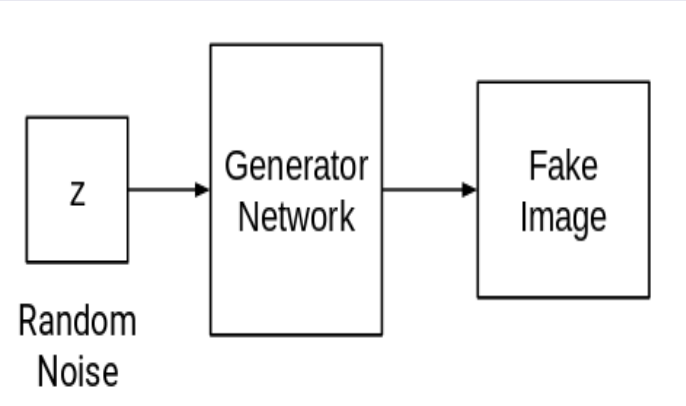
Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Networks" ArXiv, 2014

Generative adversarial networks (GANs)

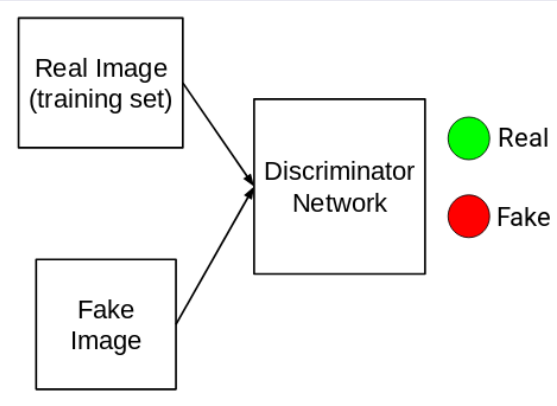


https://www.analyticsvidhya.com/blog/2020/01/generative-models-gans-computer-vision/

Generative adversarial networks (GANs)



We



train the model, calculate the loss function at the end of the discriminator network and backpropagate the loss into both discriminator and generator models

https://www.analyticsvidhya.com/blog/2020/01/generative-models-gans-computer-vision/

Applications of GANs

- GANs for Image Editing
- Using GANs for Security
- Generating Data using GANs (music, text, speech, etc.)
- GANs for Attention Prediction
- GANs for 3D Object Generation

https://www.analyticsvidhya.com/blog/2019/04/top-5-interesting-applications-gans-deep-learning/

Style transfer

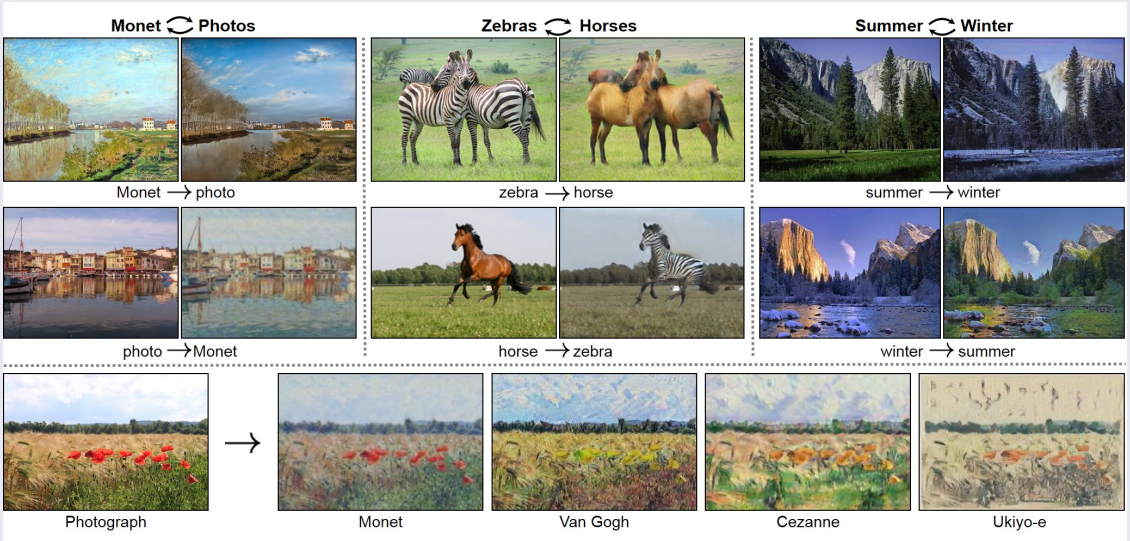
- Style transfer consists of creating a new image that preserves the contents of a target image while also capturing the style of a reference image
- Content can be captured by the high-level activations of a convnet
- Style can be captured by the internal correlations of the activations of different layers of a convnet

Chapter 8.3

http://www.byungsoo.me/project/Inst/index.html

CycleGAN: domain transformation

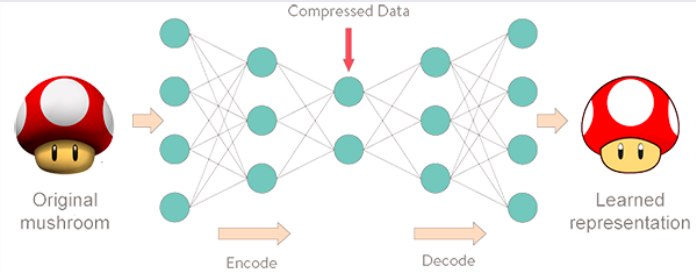
CycleGAN learns transformation across domains with unpaired data



https://junyanz.github.io/CycleGAN/

Autoencoders

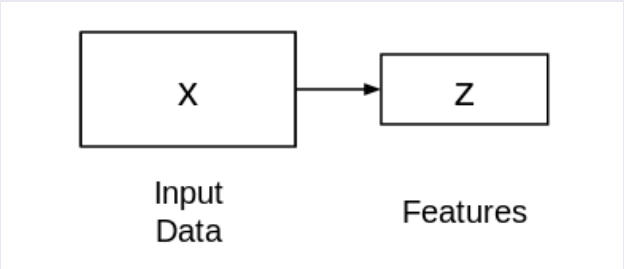
- Autoencoder is an unsupervised neural network trained to reconstruct the input. **Auto**matically **encoding** data
- One or more bottleneck layers have lower dimensionality than the input, which leads to compression of data and forces the autoencoder to extract useful features and omit unimportant features in the reconstruction



https://www.pyimagesearch.com/2020/02/17/autoencoders-with-keras-tensorflow-and-deep-learning/

Autoencoders

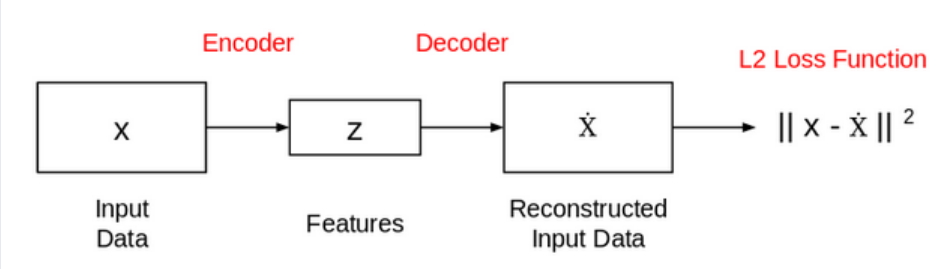
- Autoencoders learn a **compressed representation** of the input data by reconstructing it on the output of the network
- Goal: capture the structure of the data x (i.e., intrinsic relationships between the data variables) in a low-dimensional latent space z, and allows for more accurate downstream analyses



Autoencoder

- Generally, an autoencoder consists of two networks, an encoder and a decoder, which broadly perform the following tasks:
  - **Encoder**: Maps the high dimensional input data into a latent variable embedding which has lower dimensions than the input.
  - **Decoder**: Attempts to reconstruct the input data from the embedding.
- Areas of application:
  - Dimensionality reduction
  - Data denoising
  - Compression and data generation

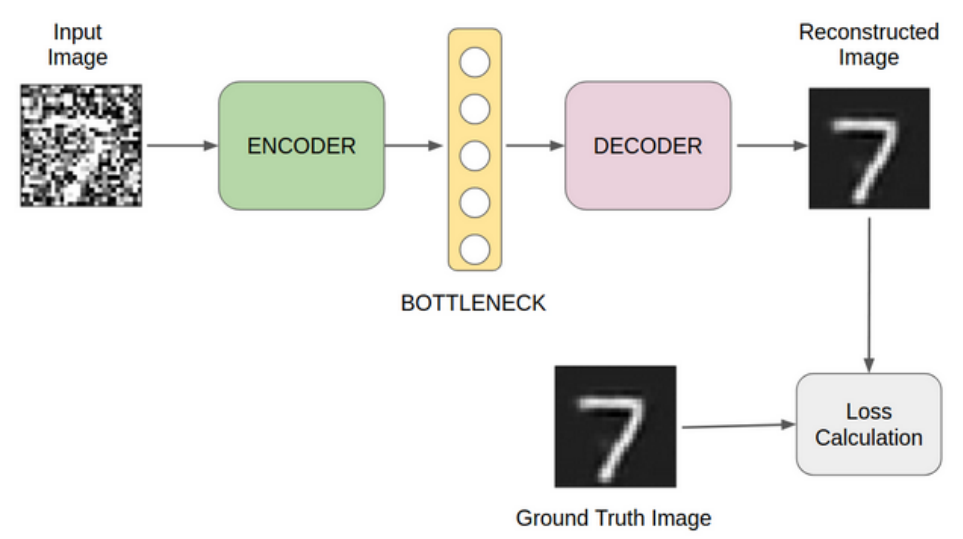
Basic autoencoder network



This network is trained in such a way that the features (z) can be used to reconstruct the original input data (x). If the output ($\hat{X}$) is different from the input (x), the loss penalizes it and helps to reconstruct the input data

How autoencoder learns

- Image denoising problem - removing noise from images



https://www.analyticsvidhya.com/blog/2020/02/what-is-autoencoder-enhance-image-resolution/

Autoencoder calculations

- The model contains an encoder function $f(.)$ parameterised by θ and a decoder function $g(.)$ parameterised by ϕ. The lower dimensional embedding learned for an input x in the bottleneck layer is $h = f_\theta(x)$ and the reconstructed input is $x' = g_\phi(f_\theta(x))$.
- The parameters θ, ϕ are learned together to output a reconstructed data sample that is ideally the same as the original input $x' \approx g_\phi(f_\theta(x))$
- There are various metrics used to quantify the error between the input and output such as cross-entropy (CE) or simpler metrics such as mean squared error: $L_{AE}(\theta, \phi) = \frac{1}{n}\sum_{i=0}^{n}(x_i - g_\phi(f_\theta(x_i))^2$

Autoencoder variants

The main challenge when designing an autoencoder is its sensitivity to the input data. While an autoencoder should learn a representation that embeds the key data traits as accurately as possible, it should also be able to encode traits which generalize beyond the original training set and capture similar characteristics in other data sets

Thus, several variants have been proposed since autoencoders were first introduced. These variants mainly aim to address shortcomings such as improved generalization, disentanglement and modification to sequence input models. Some significant examples include the **Denoising Autoencoder** (DAE), **Sparse Autoencoder** (SAE), and more recently the **Variational Autoencoder** (VAE)

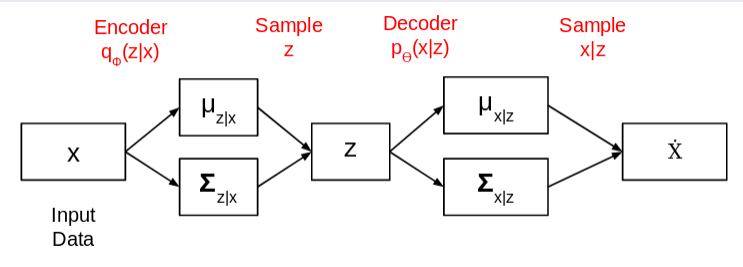
Vincent et al., 2008, Extracting and Composing Robust Features with Denoising Autoencoders

Makhzani and Frey, 2014, k-Sparse Autoencoders

Kingma and Welling, 2014, Auto-Encoding Variational Bayes

Variational Autoencoder

- VAEs are autoencoders with additional distribution assumptions that enable them to generate new random samples
- A VAE, instead of compressing its input image into a fixed code in the latent space, turns the image into the parameters of a statistical distribution: a mean and a variance
- The assumption is that the input image has been generated by a statistical process, and that the randomness of this process should be taken into account during encoding and decoding



Variational Autoencoder

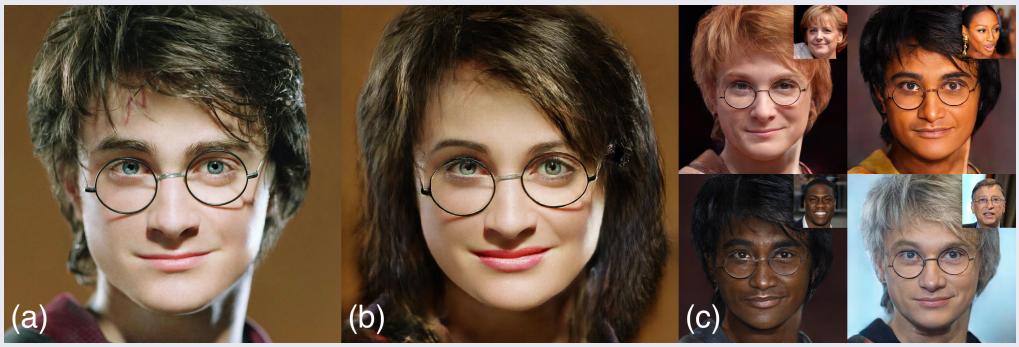
- The VAE then uses the mean and variance parameters to randomly sample one element of the distribution and decodes that element back to the original input
- The parameters of a VAE are trained via two loss functions: a *reconstruction loss* that forces the decoded samples to match the initial inputs, and a *regularization loss* that helps learn well-formed latent spaces and reduce overfitting to the training data

Image generation

- The key idea of image generation is to learn latent spaces that capture statistical information about a dataset of images
- The module capable of realizing this mapping, taking as input a latent point and outputting an image (a grid of pixels), is called a *generator* (in the case of GANs) or a *decoder* (in the case of VAEs)
- Once such a latent space has been developed, you can sample points from it, either deliberately or at random, and, by mapping them to image space, generate images that have never been seen before

GAN applications

StyleGAN2 is a state-of-the-art network in generating realistic images. Besides, it was explicitly trained to have disentangled directions in latent space, which allows efficient image manipulation by varying latent factors



Viazovetskyi Y. et al., 2020, "StyleGAN2 Distillation for Feed-forward Image Manipulation", arXiv:2003.035816
https://github.com/EvgenyKashin/stylegan2-distillation

Fake celebrity faces, https://medium.com/datadriveninvestor/artificial-intelligence-gans-can-create-fake-celebrity-faces-44fe80d419f7

LSTMs as generative networks

- LSTMs trained on collections of text can be run to generate text - predict the next token(s) given previous tokens
- Language model, can be the word- or character-based
- Can be done for handwriting generation, music, speech generation

ConvNets as generative networks

- ConvNets trained on collections of images can be run in reverse to generate images based on the representation learned by the network
- Visual representation model, DeepDream
- Can be done for speech, music, and more

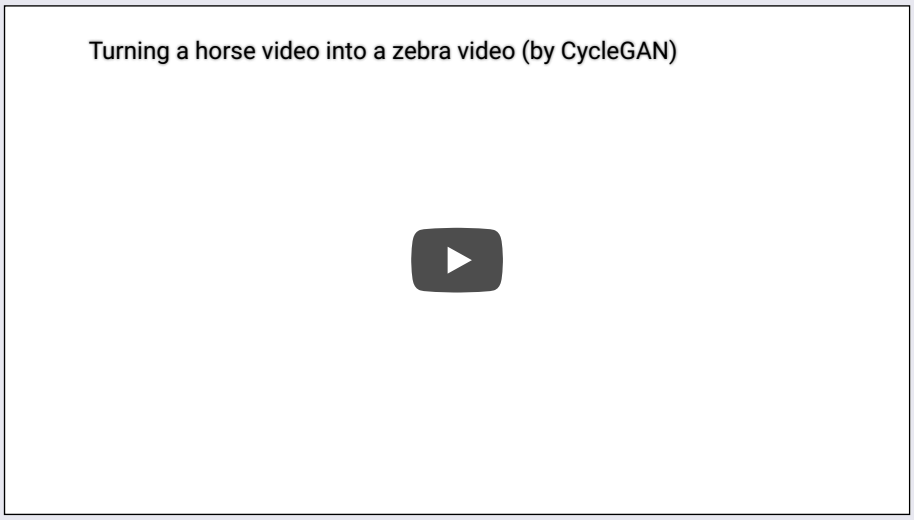
https://deepdreamgenerator.com/

https://www.tensorflow.org/tutorials/generative/deepdream

Deep belief networks

http://www.scholarpedia.org/article/Deep_belief_networks

CycleGAN: domain transformation



https://junyanz.github.io/CycleGAN/

https://interestingengineering.com/elon-musks-deepfake-video-of-singing-soviet-space-song-breaks-the-internet