Autoencoders

Topics Covered

Autoencoders For Noise Reduction from Images

Denoising Autoencoder

The Denoising Autoencoder (DAE) approach is based on the addition of noise to the input image to corrupt the data and to mask some of the values, which is followed by image reconstruction.
During the image reconstruction, the DAE learns the input features resulting in overall improved extraction of latent representations. It should be noted that Denoising Autoencoder has a lower risk of learning identity function compared to the autoencoder due to the idea of the corruption of input before its consideration for analysis that will be discussed in detail in the following sections.

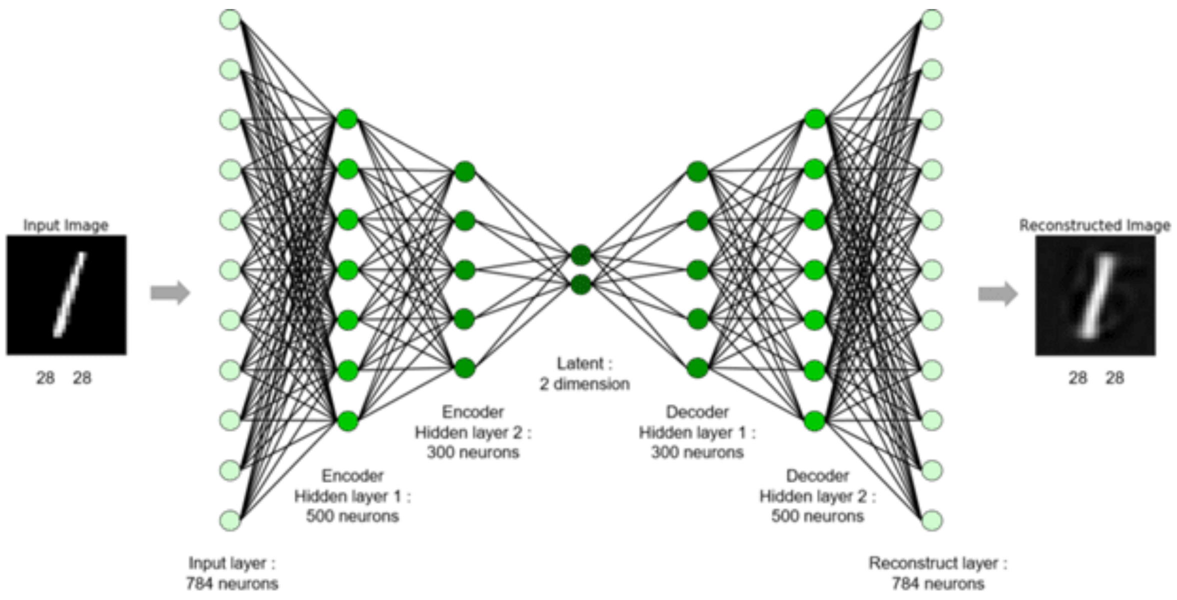
A few specifics about Denoising AutoEncoders (DAEs)

Denoising is recommended for training the model and DAEs provide the model with two important aspects; first DAEs preserve the input information (input encode), second DAEs attempt to remove (undo) the noise added to the auto-encoder input.

It should be noted that Denoising Autoencoders have been shown to be edge and larger stroke detectors from natural image patches and digit images, respectively. Finally, DAEs perform better compared to traditional filters for denoising since DAEs can be modified based on the input, unlike traditional filters which are not data specific.

In denoising, data is corrupted in some manner through the addition of random noise, and the model is trained to predict the original uncorrupted data. Another variation of this is about omitting parts of the input in contrast to adding noise to input so that model can learn to predict the original image. In this case, the idea is storing the output generated by the encoder as a feature vector, which can be used in a supervised model train-prediction approach.

Denoising autoencoders application is very versatile and can be focused on cleaning old stained scanned images or contribute to feature selection efforts in cancer biology. Regarding, old images encoder compression contributes to an output, which helps the model reconstructing the actual image using robust latent representations by the decoder. Regarding cancer biology, the extracted encoder features contribute to the efforts toward the improvement of a cancer diagnosis.


Input Image
28 28
Input layer : 784 neurons
Encoder Hidden layer 1 : 500 neurons
Encoder Hidden layer 2 : 300 neurons
Latent : 2 dimension
Decoder Hidden layer 1 : 300 neurons
Decoder Hidden layer 2 : 500 neurons
Reconstruct layer : 784 neurons
Reconstructed Image
28 28

Technical specifics of Denoising autoencoder

The idea of denoising is based on the intentional addition of noise to the input data before the presentation of data. The major technical specifics for this approach include several aspects as follows.

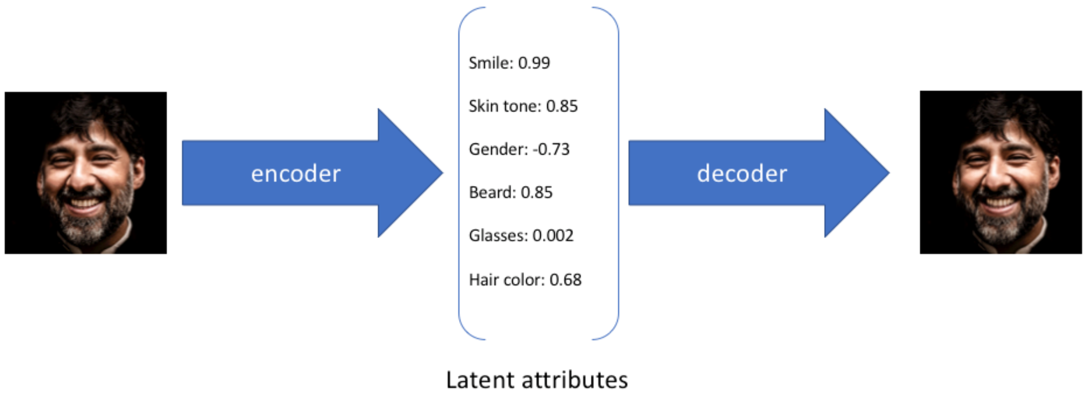
- The denoising autoencoders build corrupted copies of the input images by adding random noise.
- Next, denoising autoencoders attempt to remove the noise from the noisy input and reconstruct the output that is like the original input. A comparison is made between the original image, and the model prediction using a loss function and the goal is to minimize that loss.
- The loss function in denoising autoencoder is
- Denoising helps the autoencoder learn the latent representation in data and makes a robust representation of useful data possible hence supporting the recovery of the clean original input.
- A final note is about the random corruption/noise addition process in denoising autoencoders considering denoising as a stochastic autoencoder in this case.

Variational Autoencoders for Computer Vision

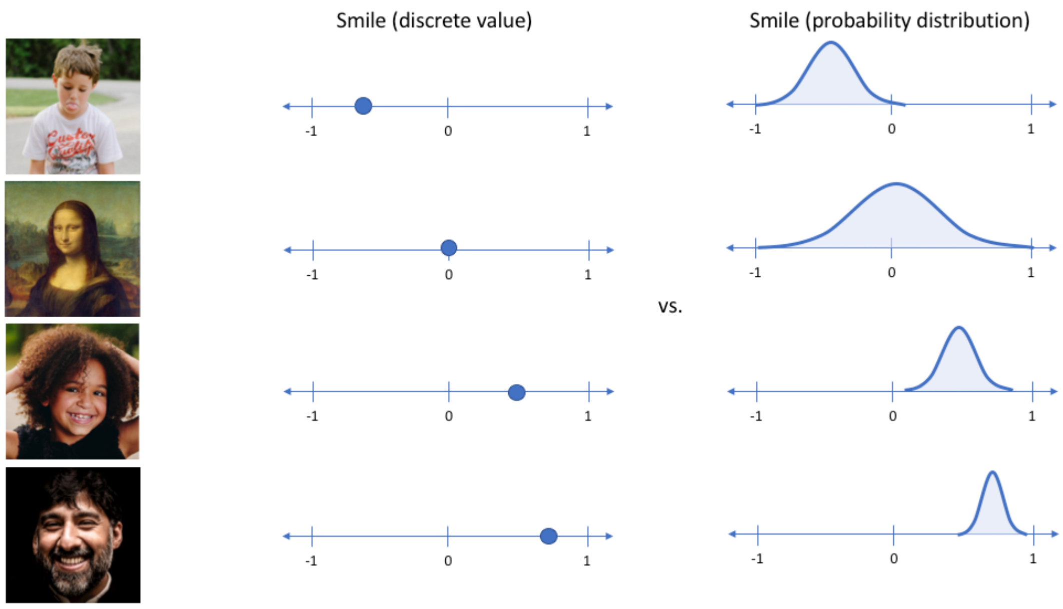
A variational autoencoder (VAE) provides a probabilistic manner for describing an observation in latent space. Thus, rather than building an encoder which outputs a single value to describe each latent state attribute, we'll formulate our encoder to describe a probability distribution for each latent attribute.

Intuition

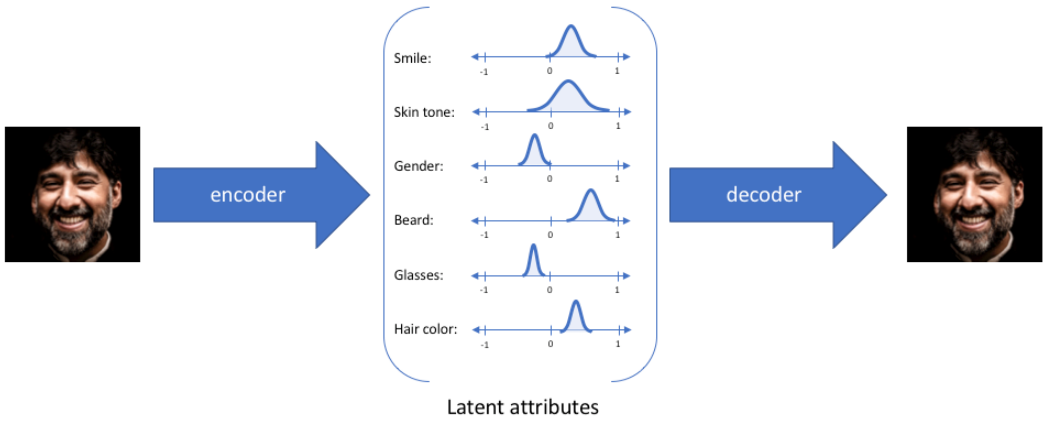
To provide an example, let's suppose we've trained an autoencoder model on a large dataset of faces with a encoding dimension of 6. An ideal autoencoder will learn descriptive attributes of faces such as skin color, whether or not the person is wearing glasses, etc. in an attempt to describe an observation in some compressed representation.



In the example above, we've described the input image in terms of its latent attributes using a single value to describe each attribute. However, we may prefer to represent each latent attribute as a range of possible values. For instance, what single value would you assign for the smile attribute if you feed in a photo of the Mona Lisa? Using a variational autoencoder, we can describe latent attributes in probabilistic terms.

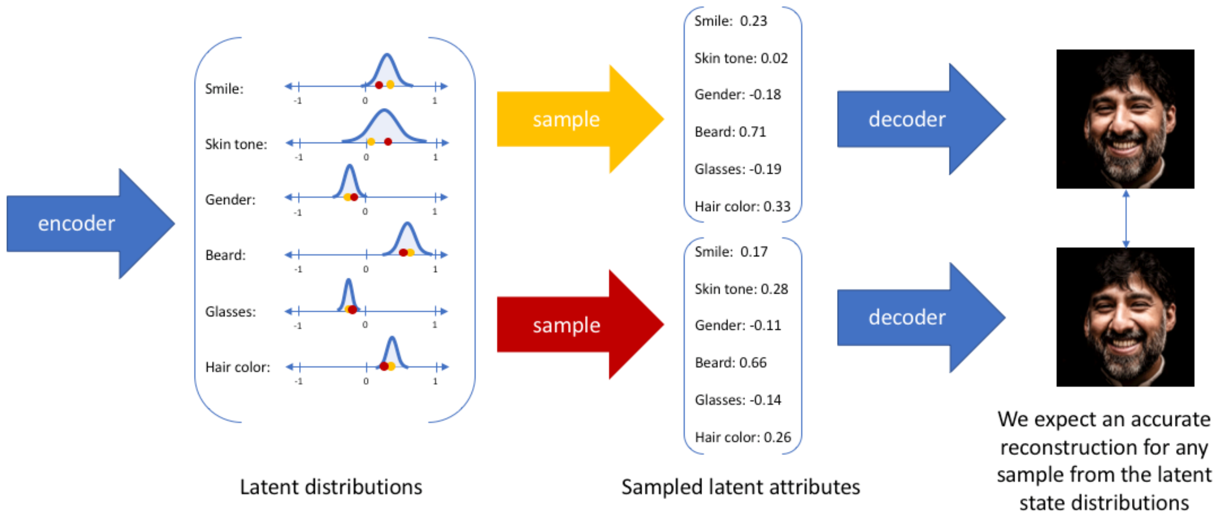


With this approach, we'll now represent each latent attribute for a given input as a probability distribution. When decoding from the latent state, we'll randomly sample from each latent state distribution to generate a vector as input for our decoder model.



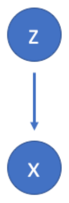
Note: For variational autoencoders, the encoder model is sometimes referred to as the recognition model whereas the decoder model is sometimes referred to as the generative model.

By constructing our encoder model to output a range of possible values (a statistical distribution) from which we'll randomly sample to feed into our decoder model, we're essentially enforcing a continuous, smooth latent space representation. For any sampling of the latent distributions, we're expecting our decoder model to be able to accurately reconstruct the input. Thus, values which are nearby to one another in latent space should correspond with very similar reconstructions.



Statistical behind VAEs

Suppose that there exists some hidden variable z which generates an observation x.



We can only see x, but we would like to infer the characteristics of z. In other words, we'd like to compute

p(z|x)

$$p\left(z|x\right) = \frac{p\left(x|z\right)p\left(z\right)}{p\left(x\right)}$$

Unfortunately, computing p(x) is quite difficult.

$$p\left(x\right) = \int p\left(x|z\right)p\left(z\right)dz$$

This usually turns out to be an intractable distribution. However, we can apply variational inference to estimate this value.

Let's approximate **p(z|x)** by another distribution **q(z|x)**) which we'll define such that it has a tractable distribution. If we can define the parameters of **q(z|x)** such that it is very similar to **p(z|x)**, we can use it to perform approximate inference of the intractable distribution.

Recall that the KL divergence is a measure of difference between two probability distributions. Thus, if we wanted to ensure that **q(z|x)** was similar to **p(z|x),** we could minimize the KL divergence between the two distributions.

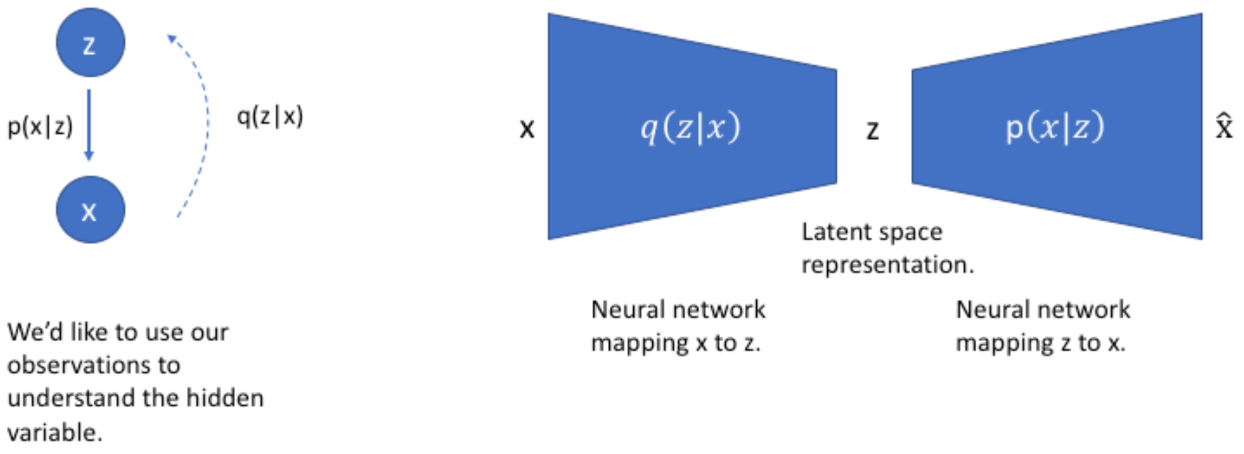
$$\min KL\left(q\left(z|x\right)||p\left(z|x\right)\right)$$

the result gives us that we can minimize the above expression by maximizing the following:

$$E_{q(z|x)} \log p\left(x|z\right) - KL\left(q\left(z|x\right)||p\left(z\right)\right)$$

The first term represents the reconstruction likelihood and the second term ensures that our learned distribution q is similar to the true prior distribution p.

To revisit our graphical model, we can use q to infer the possible hidden variables (ie. latent state) which was used to generate an observation. We can further construct this model into a neural network architecture where the encoder model learns a mapping from xx to z and the decoder model learns a mapping from z back to x.


z
p(x|z)
q(z|x)
x
We'd like to use our observations to understand the hidden variable.
x
q(z|x)
z
p(x|z)
x̂
Latent space representation.
Neural network mapping x to z.
Neural network mapping z to x.

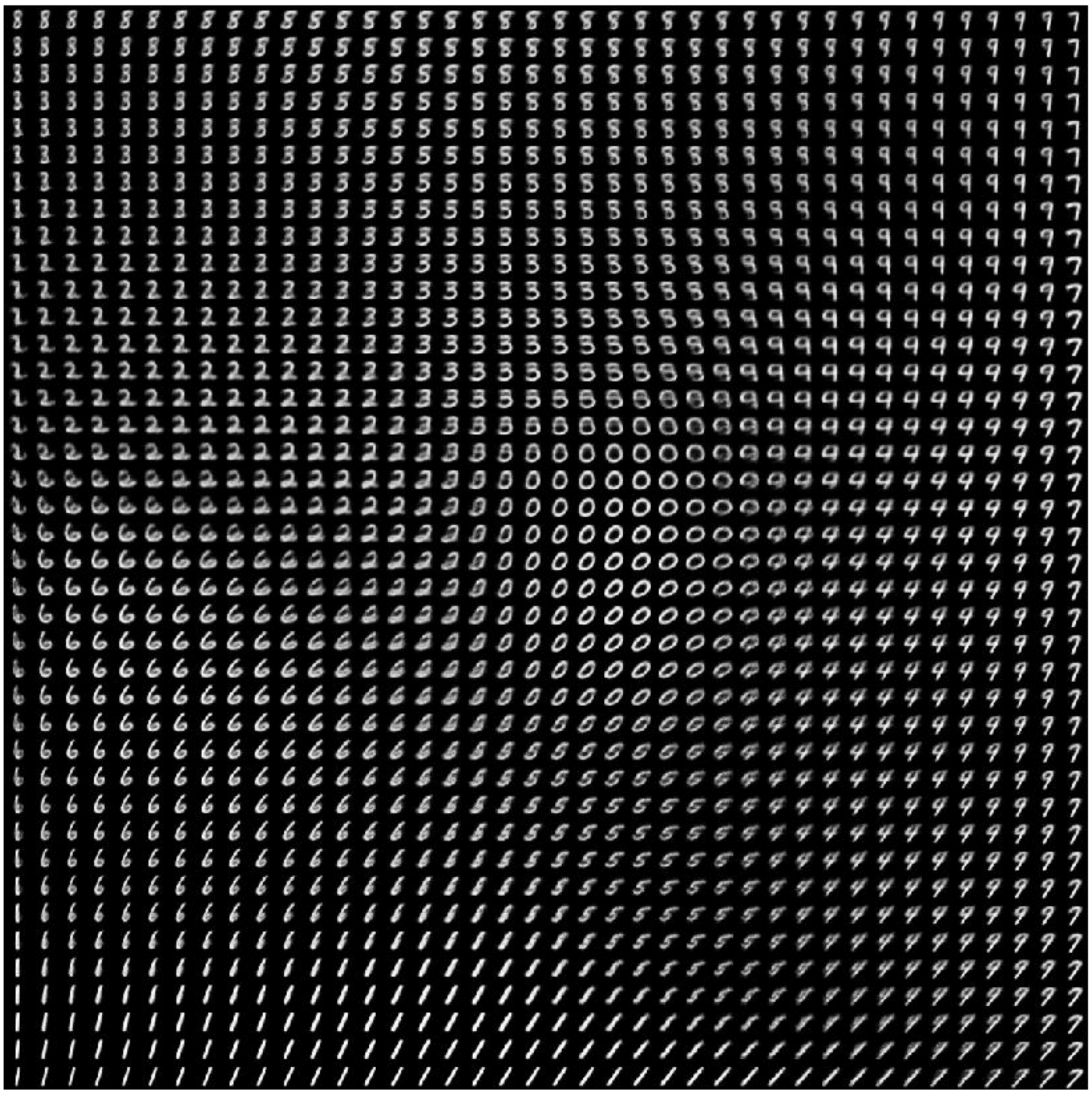
Our loss function for this network will consist of two terms, one which penalizes reconstruction error (which can be thought of maximizing the reconstruction likelihood as discussed earlier) and a second term which encourages our learned distribution **q(z|x)** to be similar to the true prior distribution **p(z),** which we'll assume follows a unit Gaussian distribution, for each dimension j of the latent space.

$$\mathcal{L}\left(x, \hat{x}\right) + \sum_{j} KL\left(q_j\left(z|x\right) || p\left(z\right)\right)$$

Variational autoencoders as a generative model

By sampling from the latent space, we can use the decoder network to form a generative model capable of creating new data similar to what was observed during training. Specifically, we'll sample from the prior distribution p(z)p(z) which we assumed follows a unit Gaussian distribution.

The figure below visualizes the data generated by the decoder network of a variational autoencoder trained on the MNIST handwritten digits dataset. Here, we've sampled a grid of values from a two-dimensional Gaussian and displayed the output of our decoder network.



As you can see, the distinct digits each exist in different regions of the latent space and smoothly transform from one digit to another. This smooth transformation can be quite useful when you'd like to interpolate between two observations,

Beta VAE

- Beta-VAE is a type of variational autoencoder that seeks to discovered disentangled latent factors. It modifies VAEs with an adjustable hyperparameter that balances latent channel capacity and independence constraints with reconstruction accuracy.

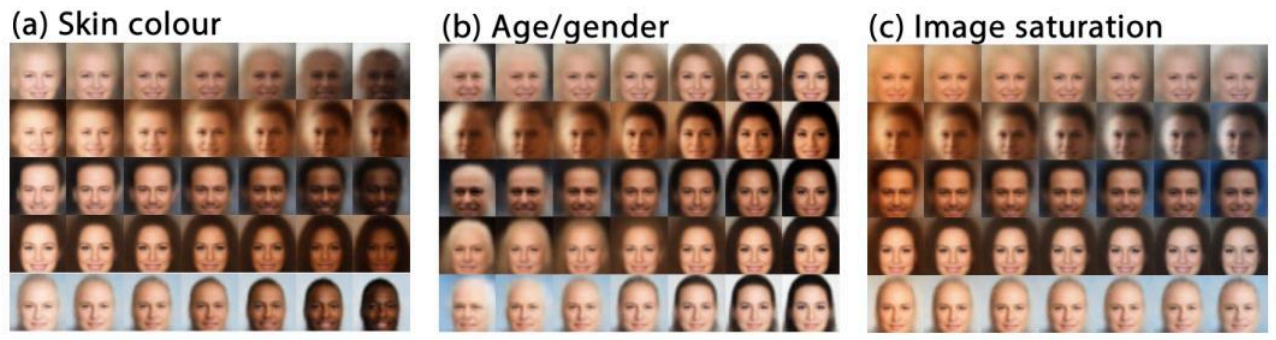


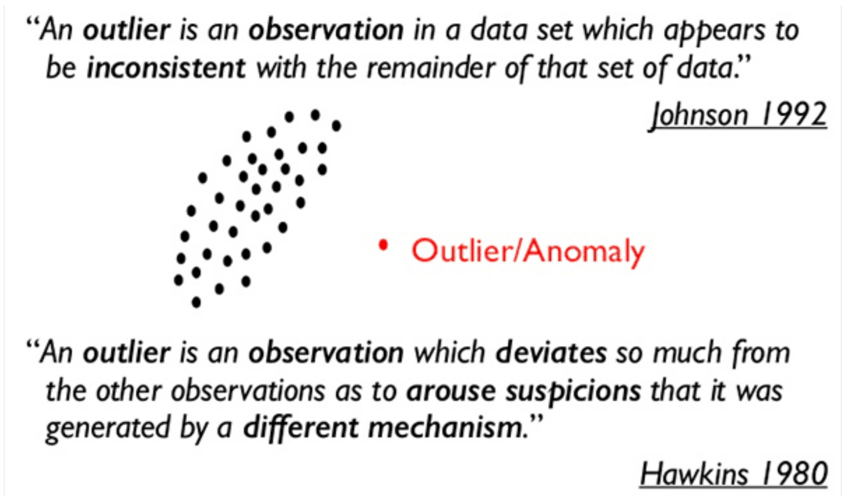
Figure 4: **Latent factors learnt by β-VAE on celebA:** traversal of individual latents demonstrates that β-VAE discovered in an unsupervised manner factors that encode skin colour, transition from an elderly male to younger female, and image saturation.

- In β-VAE, a factor β is introduced.
- A higher value of β puts more emphasis on the statistical independence than on the reconstruction.
- As it has been observed in experiments, this stronger focus on statistical independence helps a lot in making the latent code more interpretable.
- However, as focusing on statistical independence automatically means to care less about good reconstructions, we have to pay for the increased interpretability with reduced reconstruction quality.

Autoencoders For Anomaly Detection

What is anomaly detection?

Anomalies are defined as events that deviate from the standard, happen rarely, and don't follow the rest of the "pattern."



"An ***outlier*** is an ***observation*** in a data set which appears to be ***inconsistent*** with the remainder of that set of data."

Johnson 1992

"An ***outlier*** is an ***observation*** which ***deviates*** so much from the other observations as to ***arouse suspicions*** that it was generated by a ***different mechanism***."

Hawkins 1980

Examples of anomalies include:

- Large dips and spikes in the stock market due to world events
- Defective items in a factory/on a conveyor belt
- Contaminated samples in a lab

Depending on your exact use case and application, anomalies only typically occur 0.001-1% of the time — **that's an incredibly small fraction of the time.**

The problem is only compounded by the fact that there is a massive imbalance in our class labels. By definition, anomalies will *rarely occur,* so the majority of our data points will be of valid events.

To detect anomalies, machine learning researchers have created algorithms such as Isolation Forests, One-class SVMs, Elliptic Envelopes, and Local Outlier Factor to help detect such events; however, all of these methods are rooted in traditional machine learning.

How can autoencoders be used for anomaly detection?

Autoencoders are a type of unsupervised neural network that can:

1. Accept an input set of data
2. Internally compress the data into a latent-space representation
3. Reconstruct the input data from the latent representation

To accomplish this task, an autoencoder uses two components: an **encoder** and a **decoder.**
The **encoder** accepts the input data and compresses it into the latent-space representation.
The **decoder** then attempts to reconstruct the input data from the latent space.
When trained in an end-to-end fashion, the hidden layers of the network learn filters that are robust and even **capable of denoising the input data**.

However, what makes autoencoders so special from an anomaly detection perspective is the reconstruction loss. When we train an autoencoder, we typically measure the **mean-squared-error** (MSE) between:

1. The input image

2. The reconstructed image from the autoencoder

The lower the loss, the better a job the autoencoder is doing at reconstructing the image.

Let's now suppose that we trained an autoencoder on the entirety of the MNIST dataset:

Figure: Samples from the MNIST handwritten digit benchmarking dataset. We will use MNIST to develop an unsupervised autoencoder with Keras, TensorFlow, and deep learning.

We then present the autoencoder with a digit and tell it to reconstruct it:

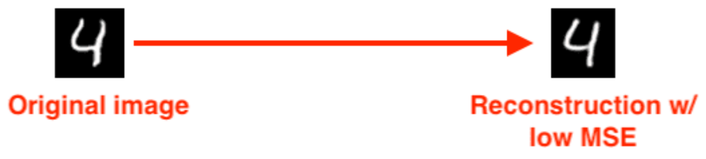


Figure: Reconstructing a digit from MNIST with autoencoders, Keras, TensorFlow, and deep learning.

We would expect the autoencoder to do a really good job at reconstructing the digit, as that is *exactly* what the autoencoder was trained to do — and if we were to look at the MSE between the input image and the reconstructed image, we would find that it's quite low.

Let's now suppose we presented our autoencoder with a photo of an elephant and asked it to reconstruct it:

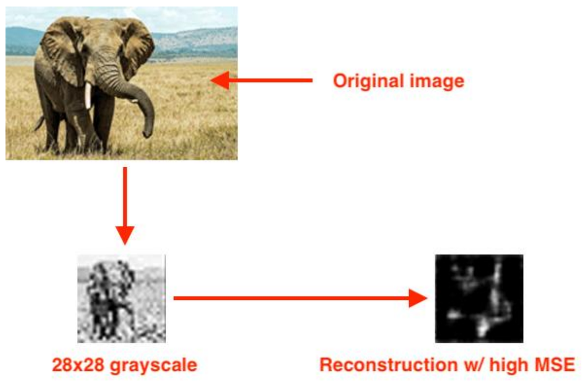


Figure: When we attempt to reconstruct an image with an autoencoder, but the result has a high MSE, we have an outlier. In this tutorial, we will detect anomalies with autoencoders, Keras, and deep learning.

Since the autoencoder has *never seen an elephant before*, and more to the point, *was never trained to reconstruct an elephant,* **our MSE will be *very high.***

If the MSE of the reconstruction is high, then we likely have an outlier.

Summary

- The Denoising Autoencoder (DAE) approach is based on the addition of noise to the input image to corrupt the data and to mask some of the values, which is followed by image reconstruction.
- In denoising, data is corrupted in some manner through the addition of random noise, and the model is trained to predict the original uncorrupted data.
- A variational autoencoder (VAE) provides a probabilistic manner for describing an observation in latent space.
- By sampling from the latent space, we can use the decoder network to form a generative model capable of creating new data similar to what was observed during training.
- Beta-VAE is a type of variational autoencoder that seeks to discovered disentangled latent factors. It modifies VAEs with an adjustable hyperparameter that balances latent channel capacity and independence constraints with reconstruction accuracy.
- Anomalies are defined as events that deviate from the standard, happen rarely, and don’t follow the rest of the pattern.