Session 12

Autoencoders & GANs

Outline

1. **Recap**
2. Unsupervised Learning
3. Autoencoders
4. Variational Autoencoders
5. GANs

Sequence Data

"Hello, Hello, Hello, is anybody in there?"

"Hello, Hello, Hello, is anybody in there?"

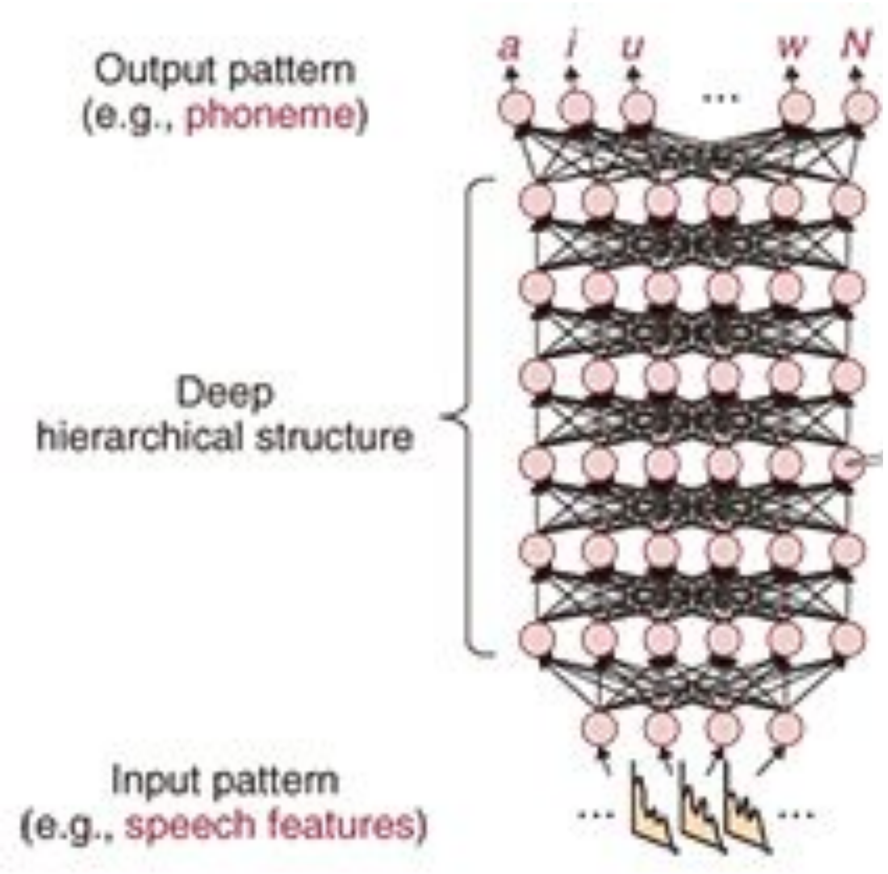
"Hola, Hola, Hola, ¿hay alguien ahí?"

INPUT IS A SEQUENCE OF N

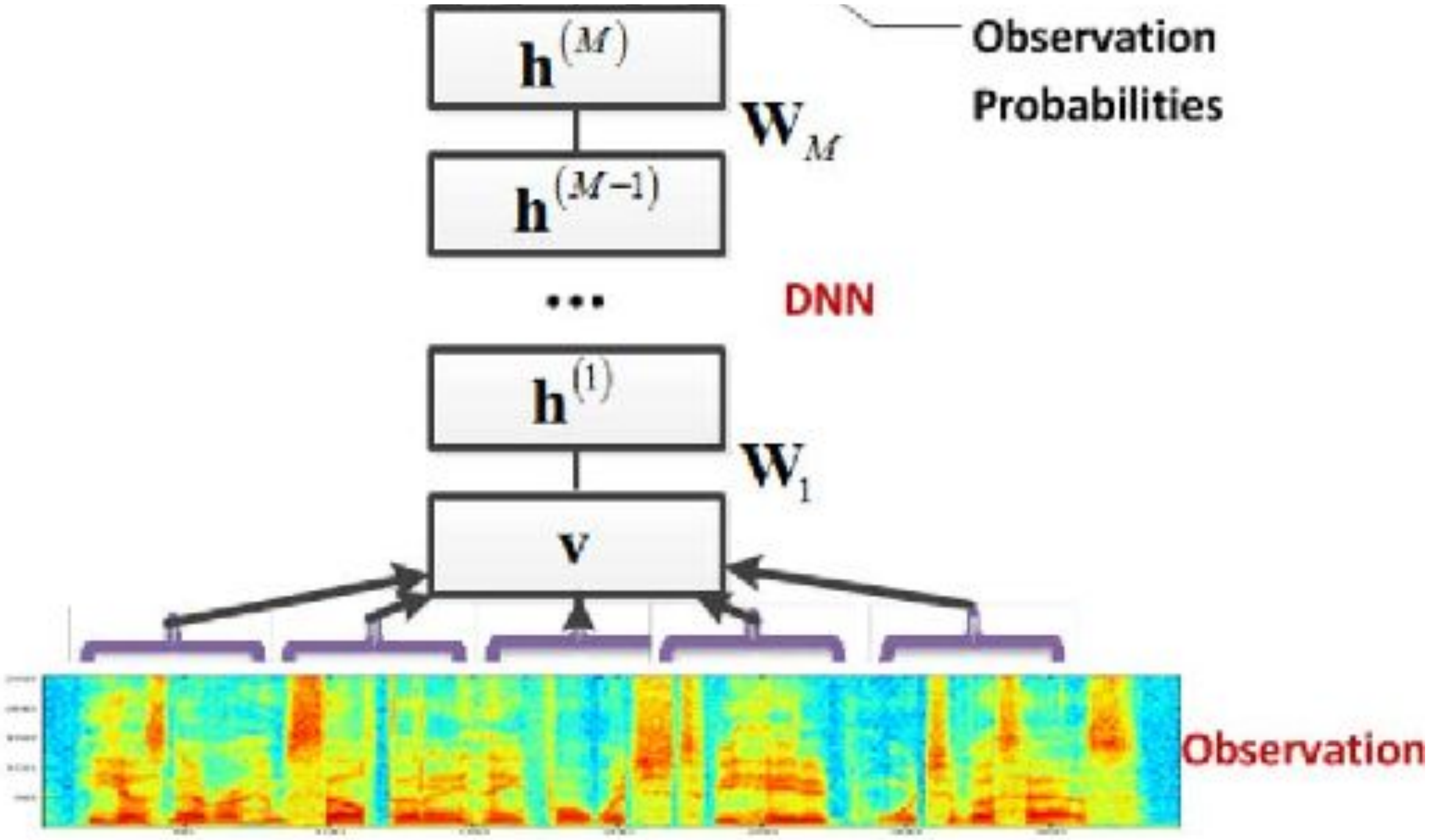
OUTPUT IS A SEQUENCE OF M

TIME

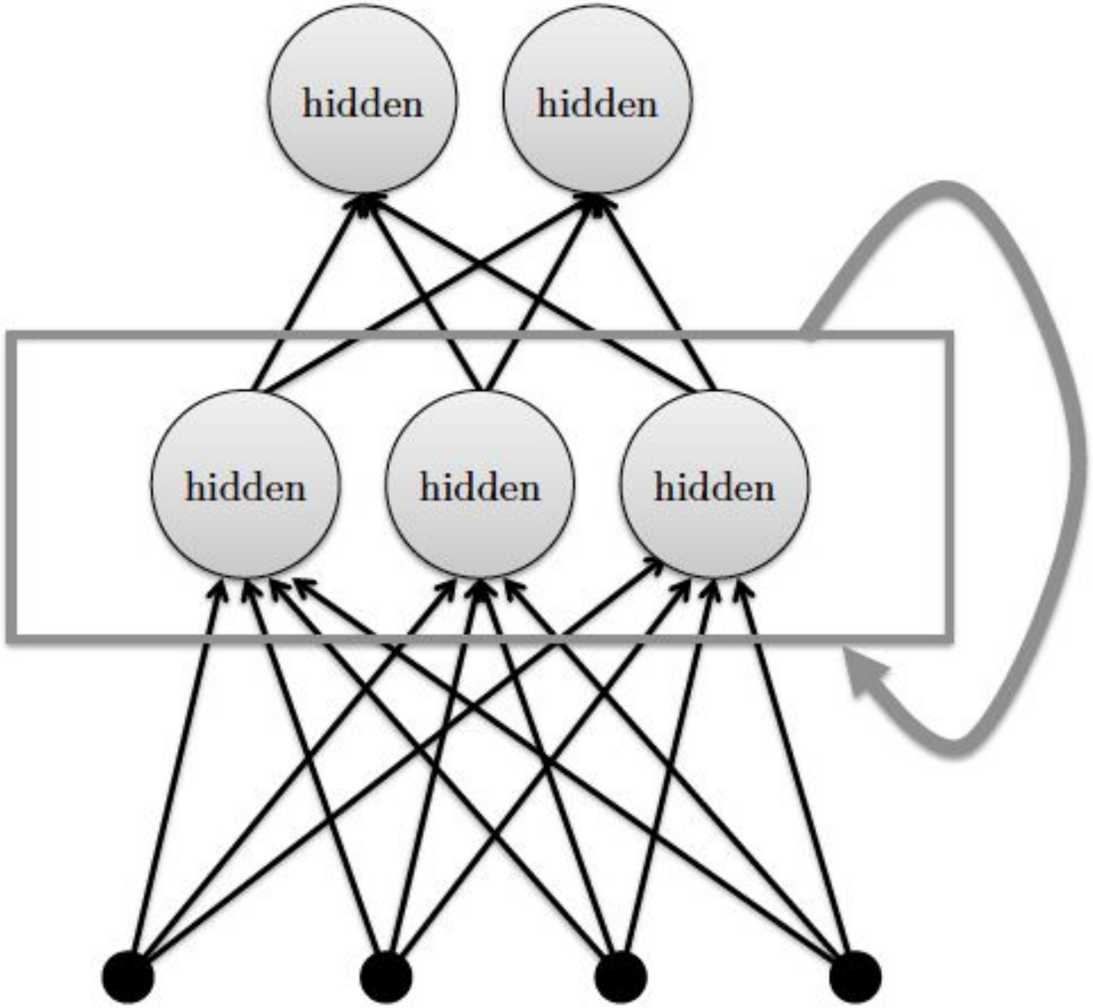
What if we use a multilayer perceptron?



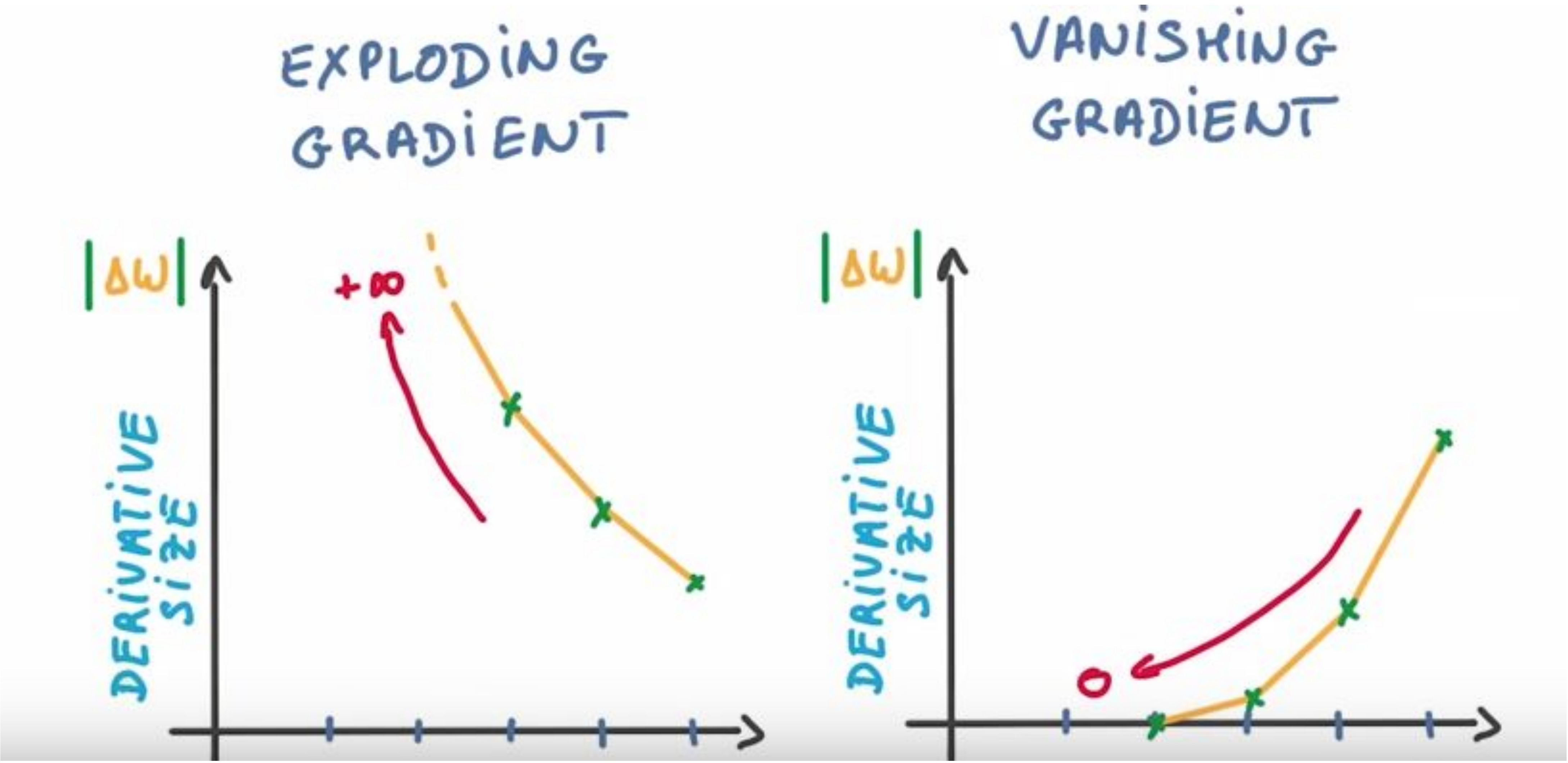
What if we use a multilayer perceptron?



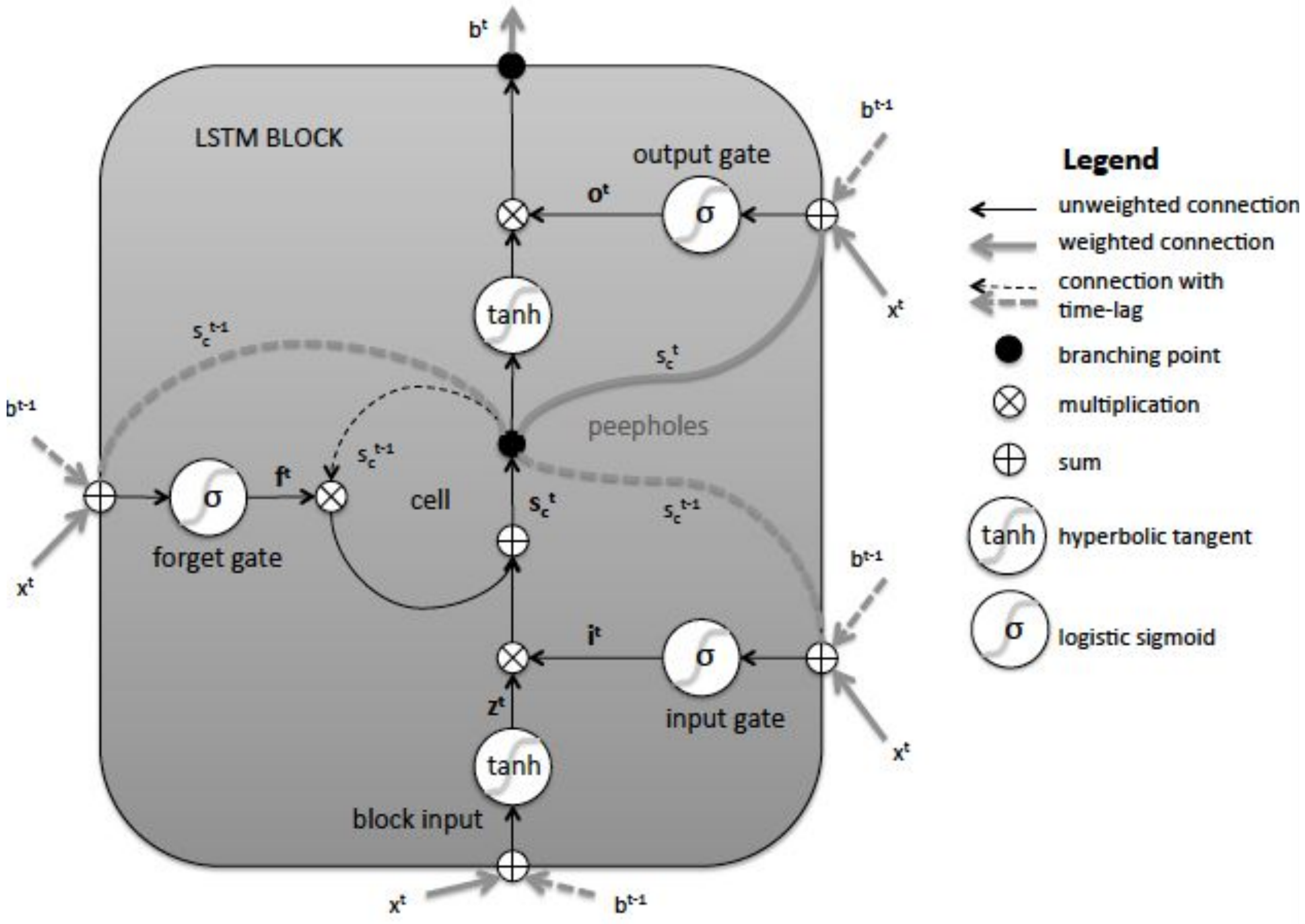
Recurrent Neural Network



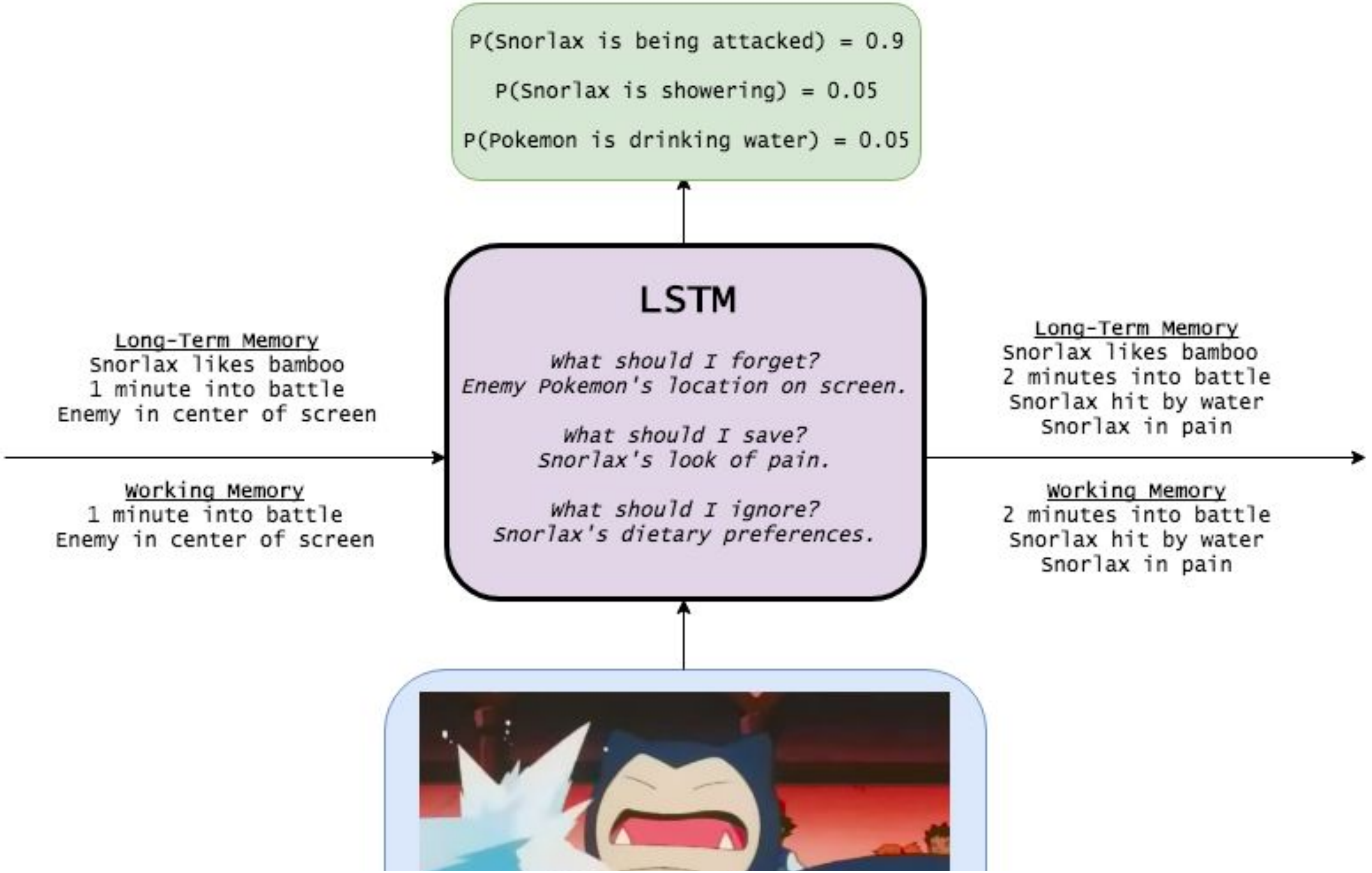
The problem: Vanishing gradients



LSTM's: The unit



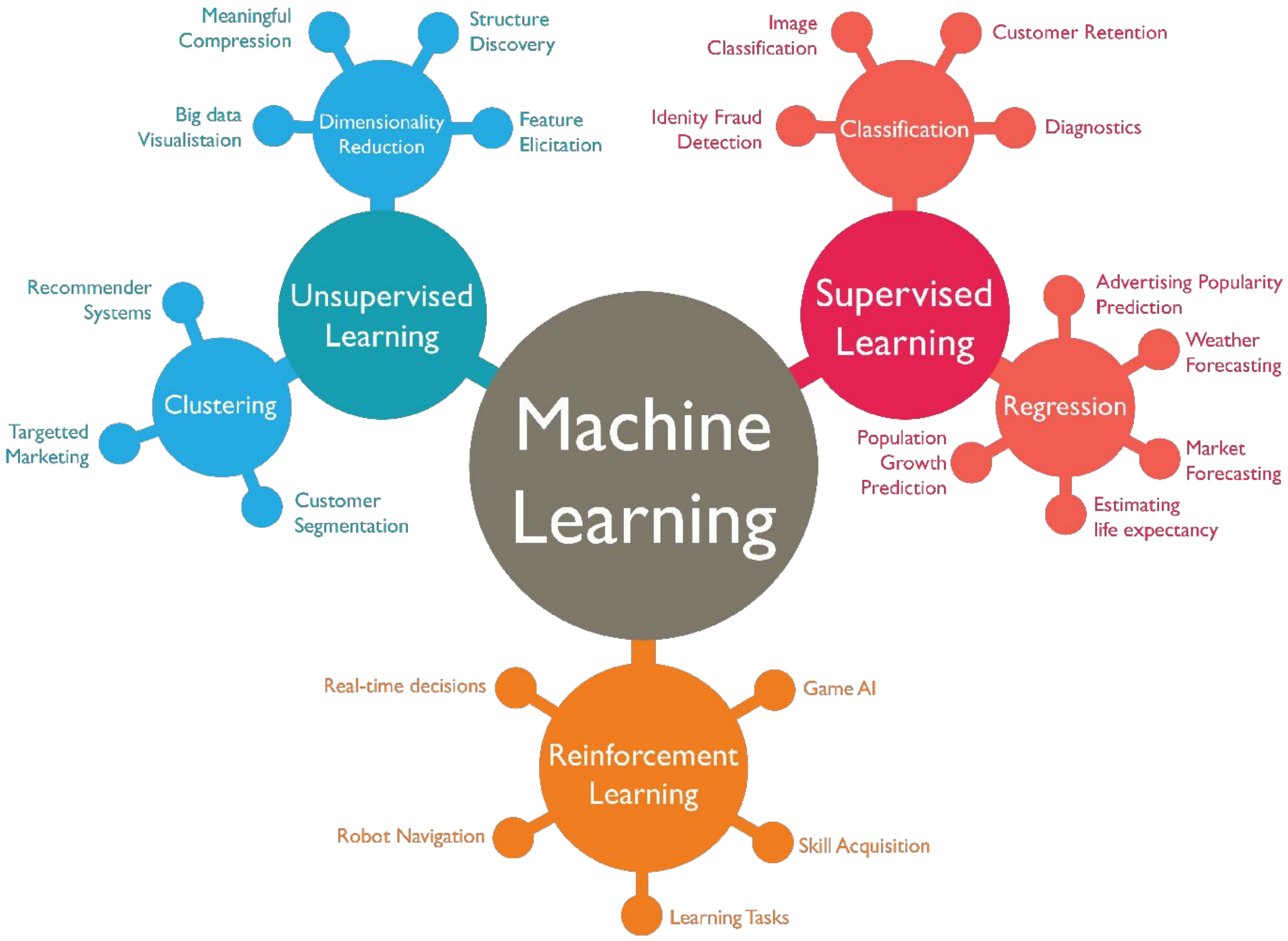
LSTM’s: An intuitive vision



Outline

1. Recap
2. **Unsupervised Learning**
3. Autoencoders
4. Variational Autoencoders
5. GANs

Unsupervised Learning



The future of AI is Unsupervised Learning

The-next-ai-revolution-will-come-from-machine-learnings-most-underrated-form

The rise of Artificial Intelligence trough Deep Learning

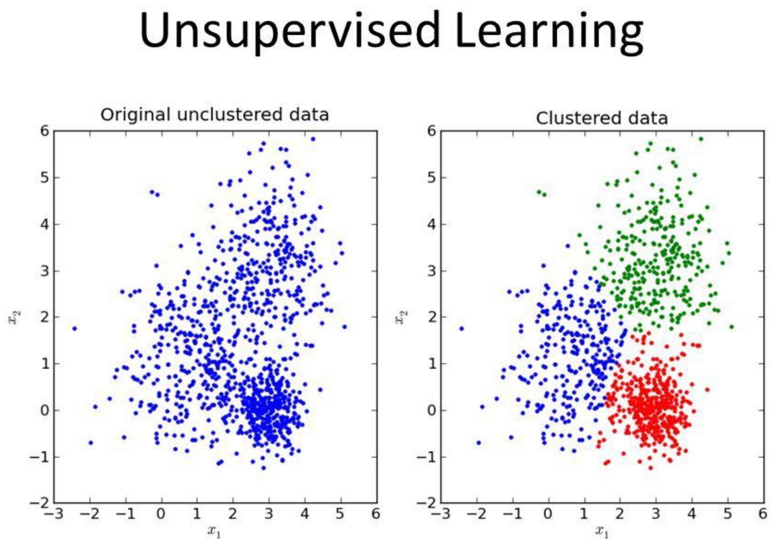
Geoffrey Hinton & Google Brain Unsupervised Learning Algorithm Improves SOTA Accuracy on ImageNet by 7%

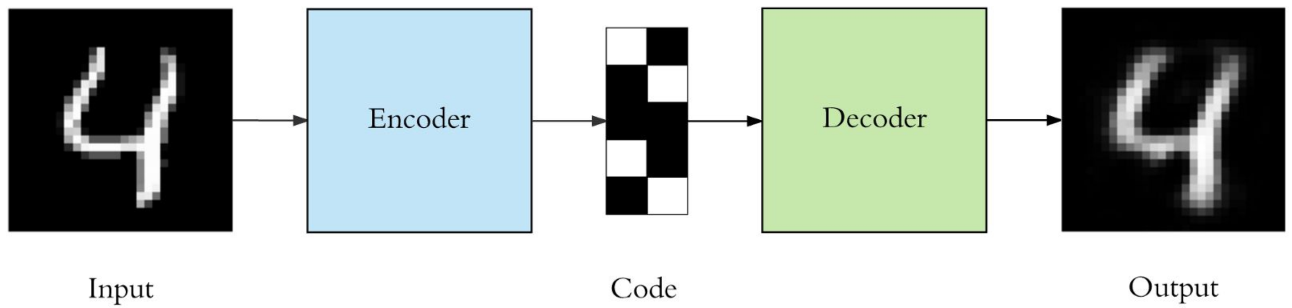
Unsupervised Learning

There is no labels. We just can understand the structure of the data.

Applications. Clustering, Dimensional Reduction, Denoising…

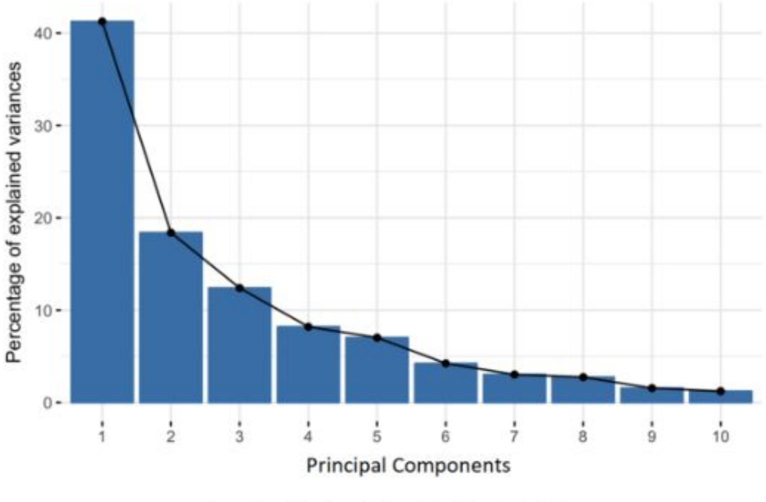
Different methods: Principal Component Analysis, K-Means, Autoencoders…



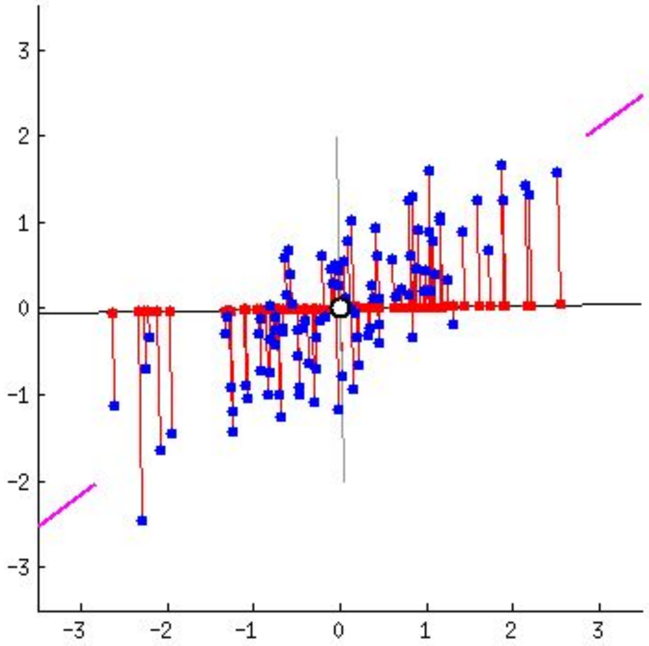


Unsupervised Learning:PCA

PCA finds new components (linear combinations of original variables) which accumulate the largest variances.
Common use: dimensionality reduction.



Percentage of variance (information) for by each PC

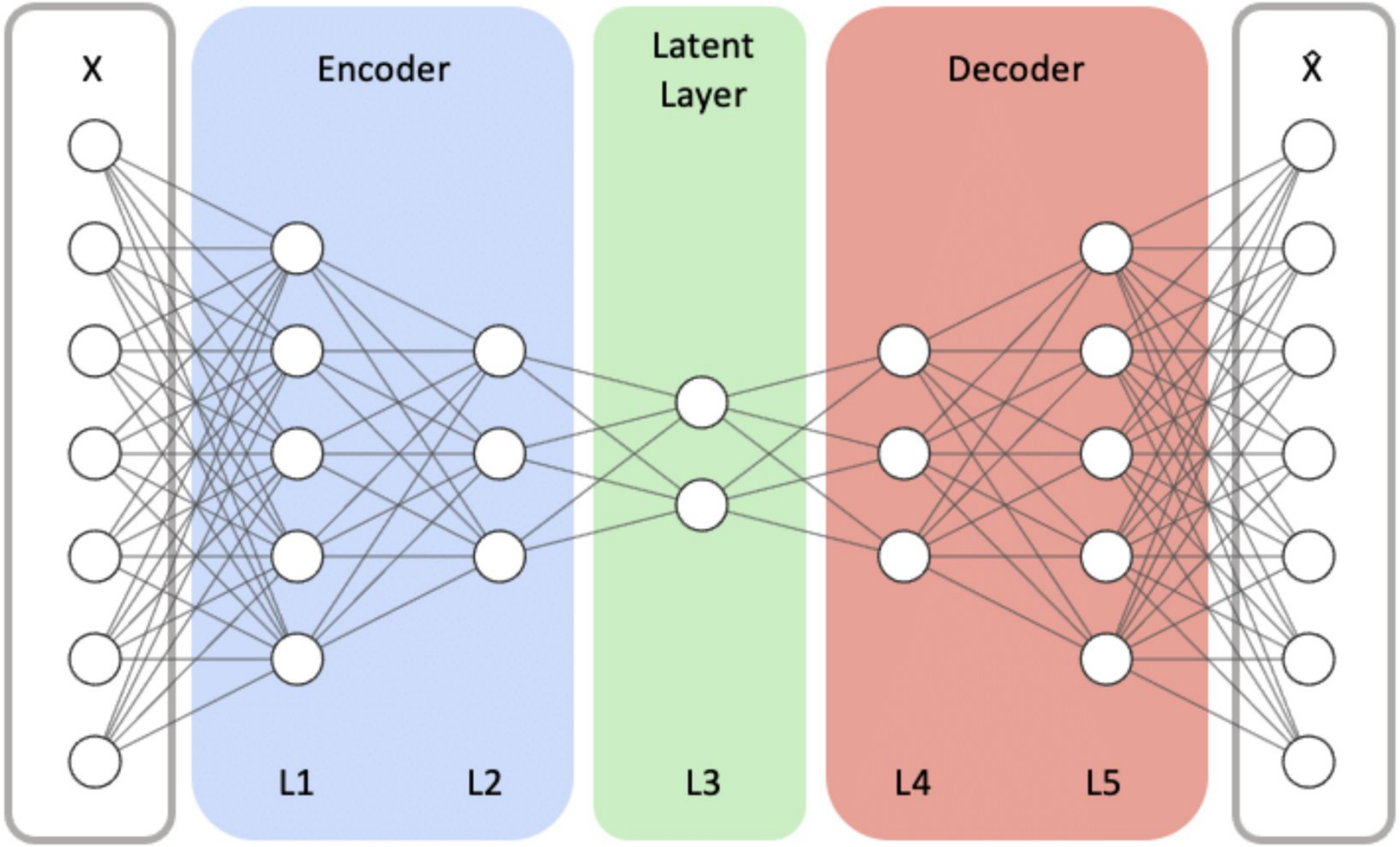


 @RZC. DL

Outline

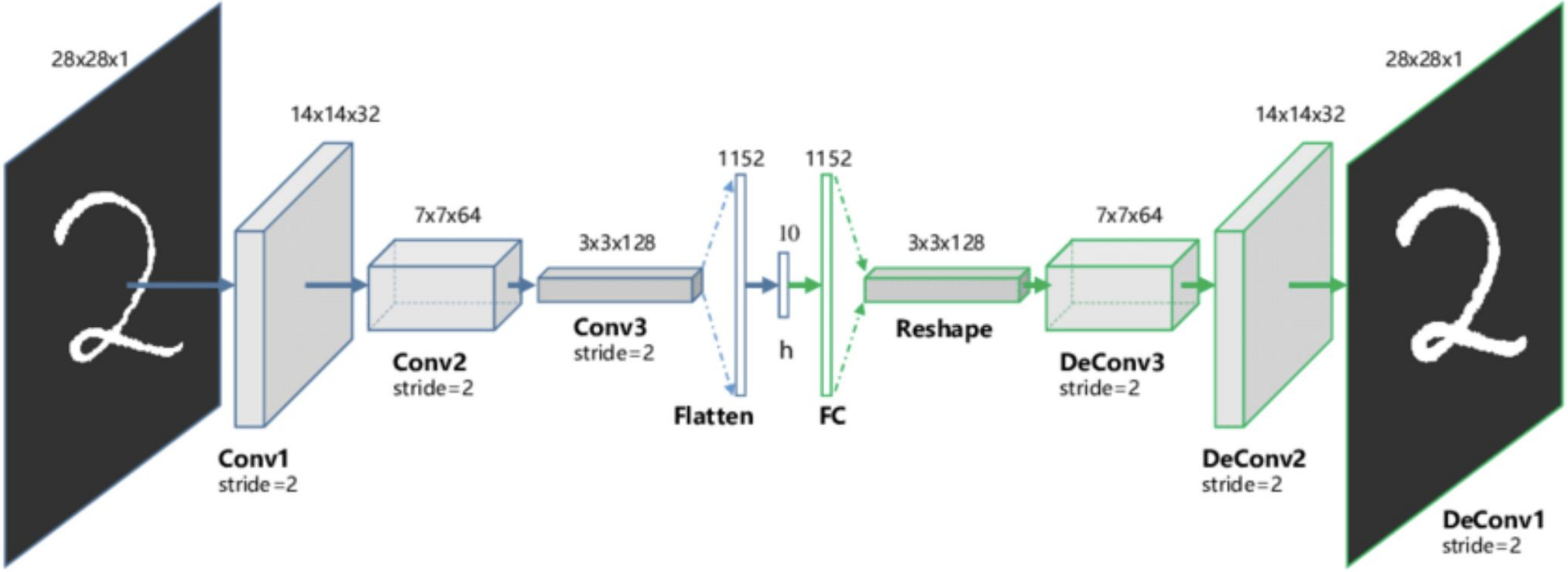
1. Recap
2. Unsupervised Learning
3. **Autoencoders**
4. Variational Autoencoders
5. GANs

Unsupervised Learning: Autoencoder



The autoencoder tries to learn a function f(x)≈x. In other words, it is trying to learn an approximation to the identity function, so as to output $\hat{x}$ that is similar to x.

Unsupervised Learning: Autoencoder

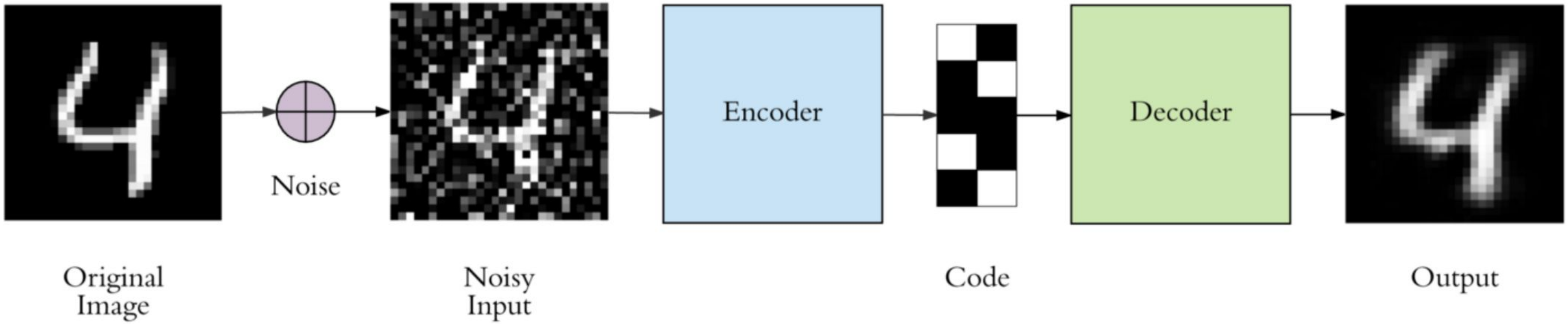


Autoencoders: Applications

1. Data Compression
1. Data Denoising
1. Dimensionality Reduction
1. Data Generation
1. ….

https://medium.com/synaptech/autoencoders-what-are-they-good-for-48bd21a49dc7

Denoising Autoencoders



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

Denoising Autoencoders

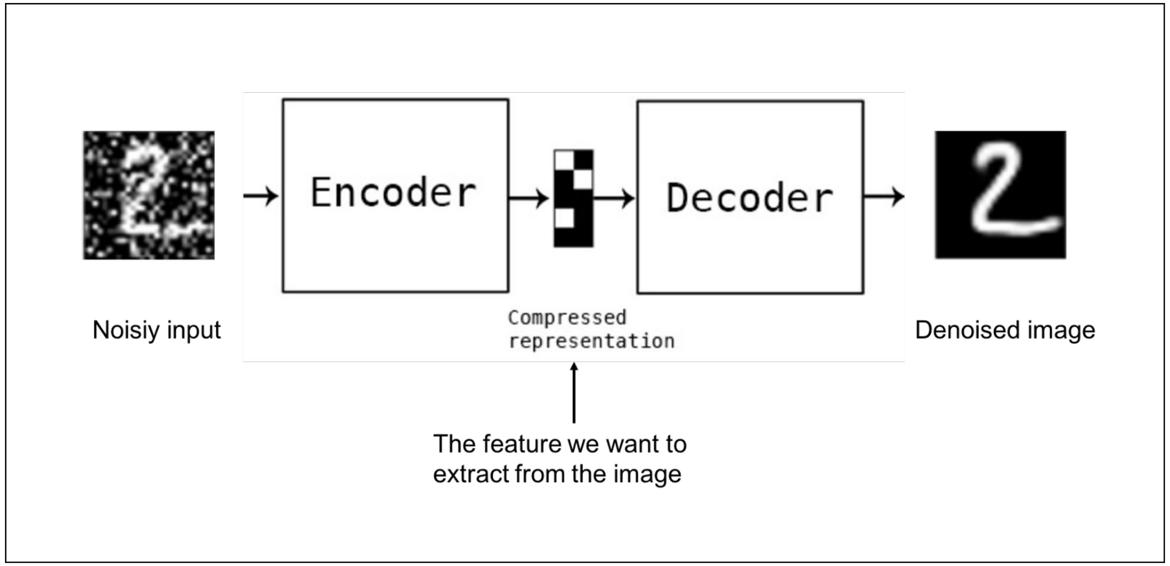
Let's see the following notebook --> ImageDenoising_Autoencoder_tf2_template.ipynb

Denoising Autoencoders: How it works?

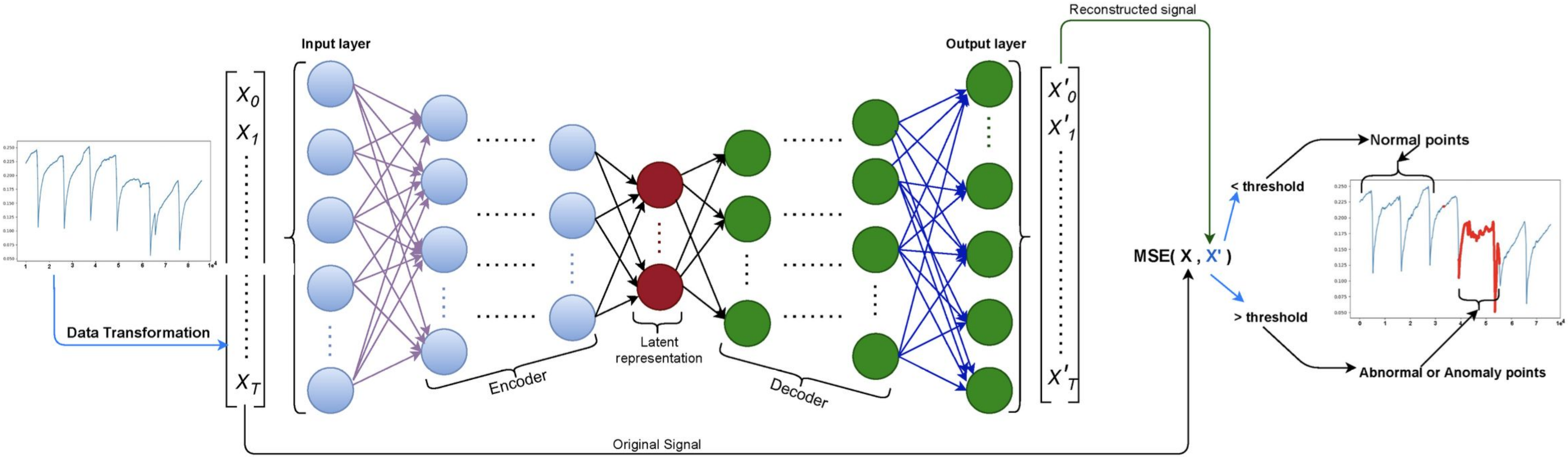
1. We intentionally add random noise to some inputs

2. We pass that data through our autoencoder: encoder and decoder

3. The key: when computing the loss error $|x – g(f(x))|$ we use the original input rather than the noisy one. In that way we force the autoencoder to capture the relevant info about the input discarding the noise



Anomaly Detection with Autoencoders

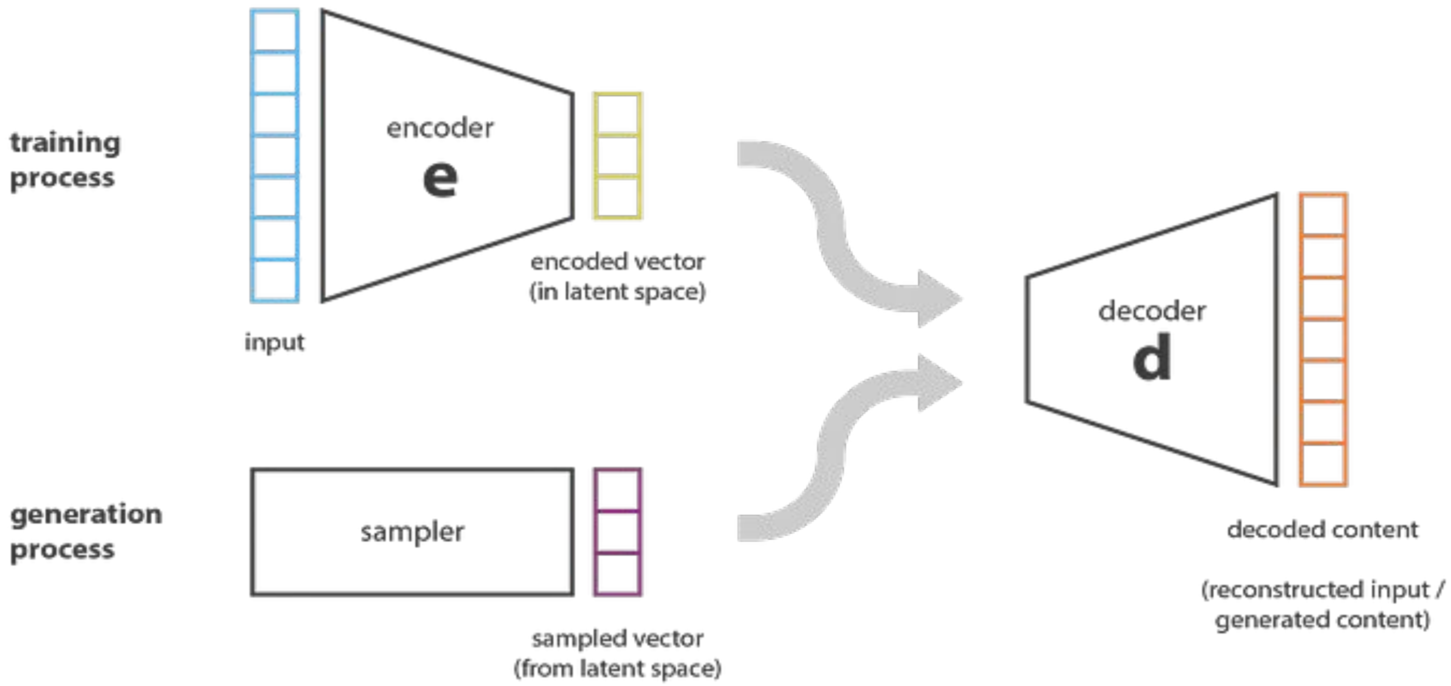


https://medium.com/analytics-vidhya/anomaly-detection-in-cardio-dataset-using-deep-learning-technique-auto encoder- fd24ca9e5c69

Outline

1. Recap
2. Unsupervised Learning
3. Autoencoders
4. **Variational Autoencoders**
5. GANs

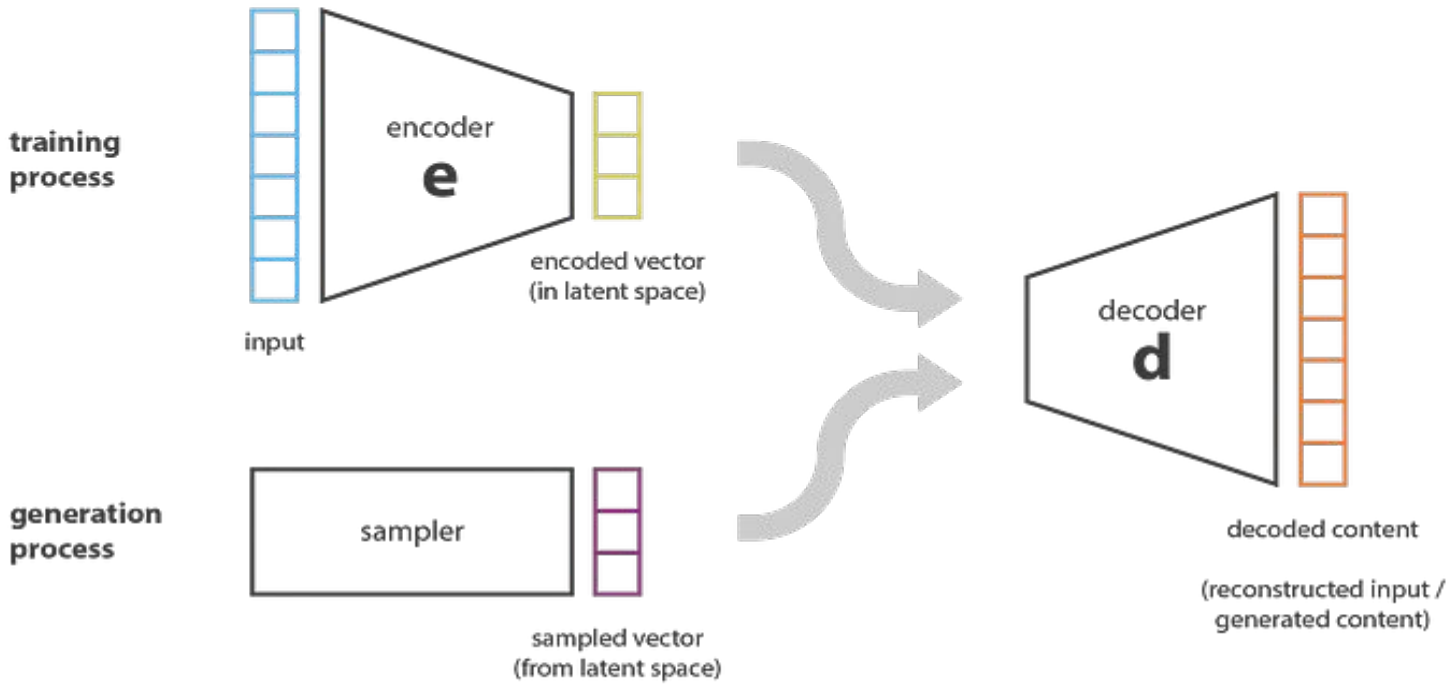
Variational Autoencoders: Motivation



Can we use an Autoencoder to generate new content?

How would you do it?

Variational Autoencoders: Motivation

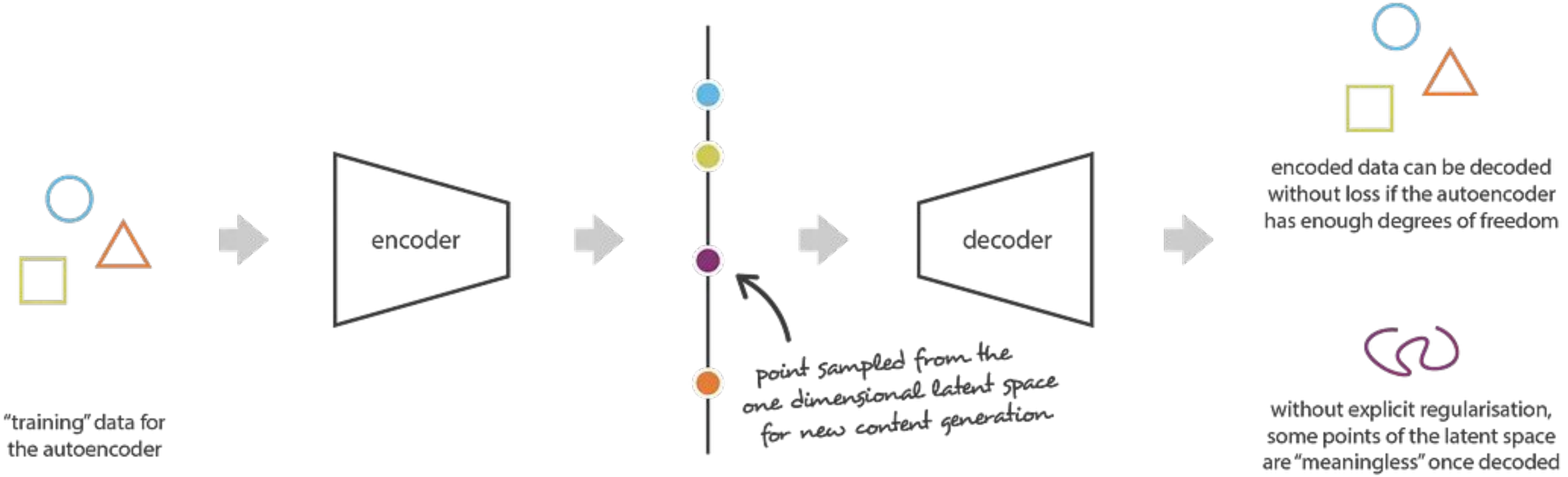


Can we use an Autoencoder to generate new content?

- Select a point in the latent space (¿Which onel?)
- Feed the decoder with that point, obtain a new sample

Variational Autoencoders: Motivation

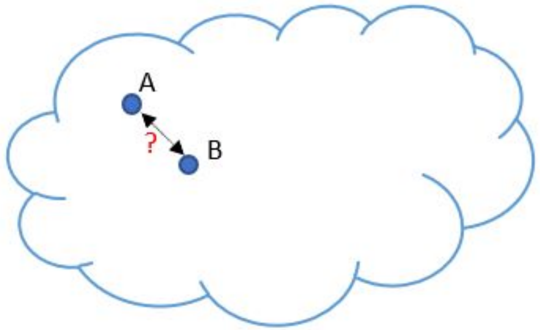
- Autoencoders built with neural networks are very powerful. They can achieve great results but they are not optimized for content generation:
  - Cost Function: Optimize data reconstruction with lowest possible error, no matter how the latent space is organized

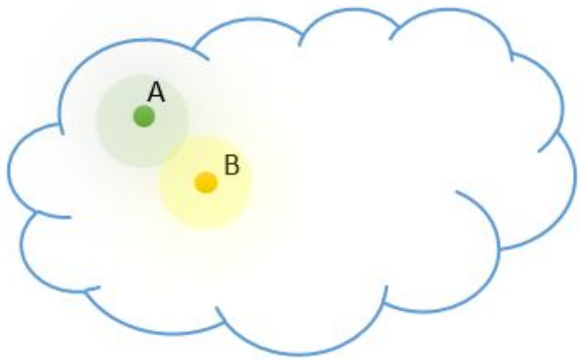


Variational Autoencoders: Motivation

Let us assume we train an autoencoder with manuscript letters

- In the latent space we codify the information about origin data so that it is enough for reconstruction.
- Each letter should correspond to a point in the latent space, which we can see as a "cloud" of possible latent variables.

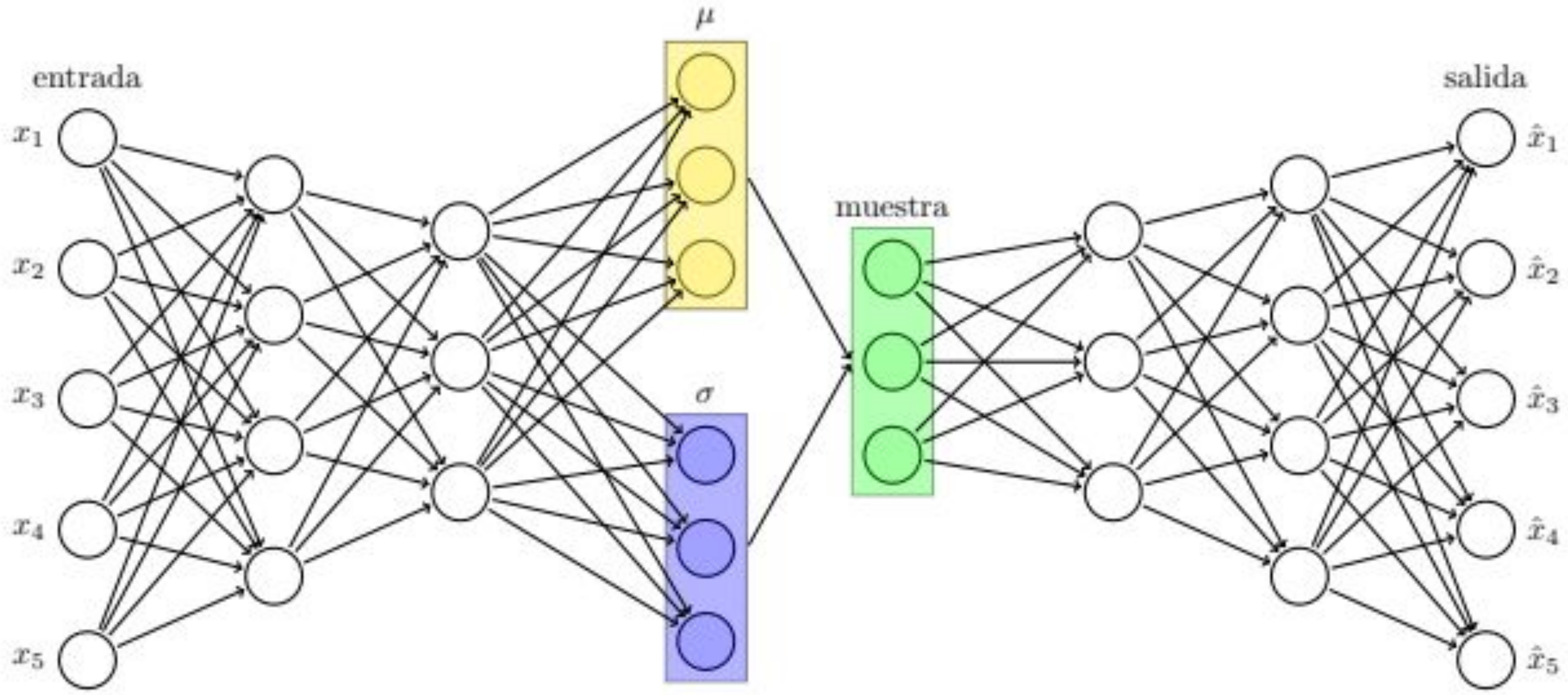




If we use the autoencoder to generate new images we encounter a problem:

- Decoder can reconstruct well the points used in training but it behaves very poorly when we make small variations
  - Latent space is not continuous!

Variational Autoencoders



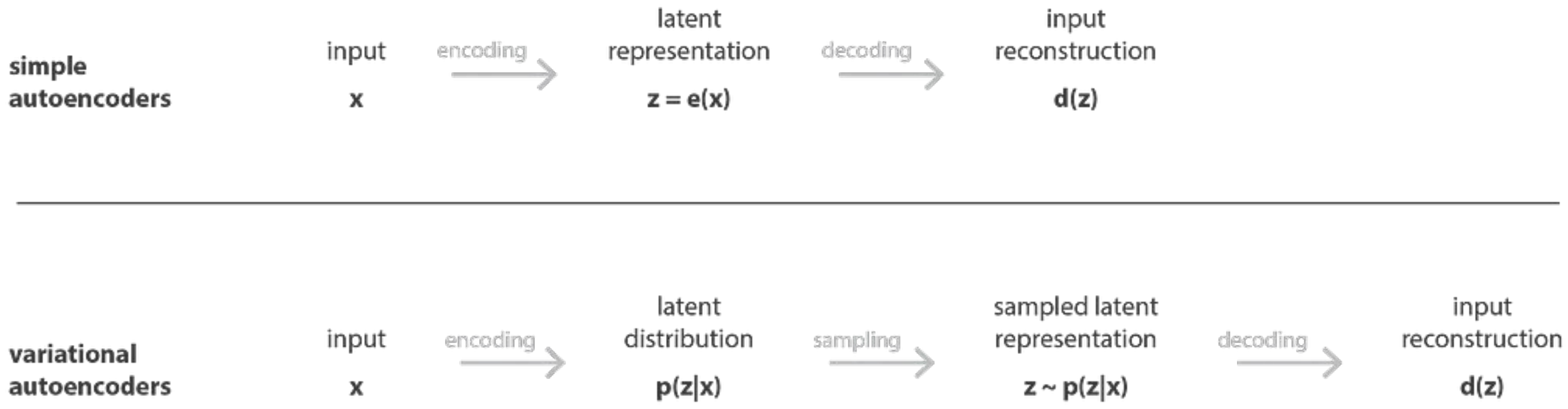
- A VAE (Variational Autoencoder) is an autoencoder modified so that we ensure that the latent space has "good properties" for generating new samples

- We match each input sample to a normal probability distribution (represented with mean and std) instead of matching it with a point.
- Each probability distribution (representing an input) can be sampled to generate infinite points within the latent space that can be decoded
  - Each input is assigned to an area instead of being assigned to a point within the latent space

Variational Autoencoders

A VAE, as an autoencoder, has an encoder and a decoder and it is trained to minimize reconstruction error.

However, instead of encoding every input as a point it is encoded as an area, a probability distribution within the latent space



- Each sample is encoded as a probability distribution within the latent space
- Then, we sample that distribution to obtain some points
- Next, every sampled point is decoded and reconstruction error is computed.
- Finally, we use that error for back-propagation (training of the networks)

Variational Autoencoders

Latent space intuition, what are good properties?

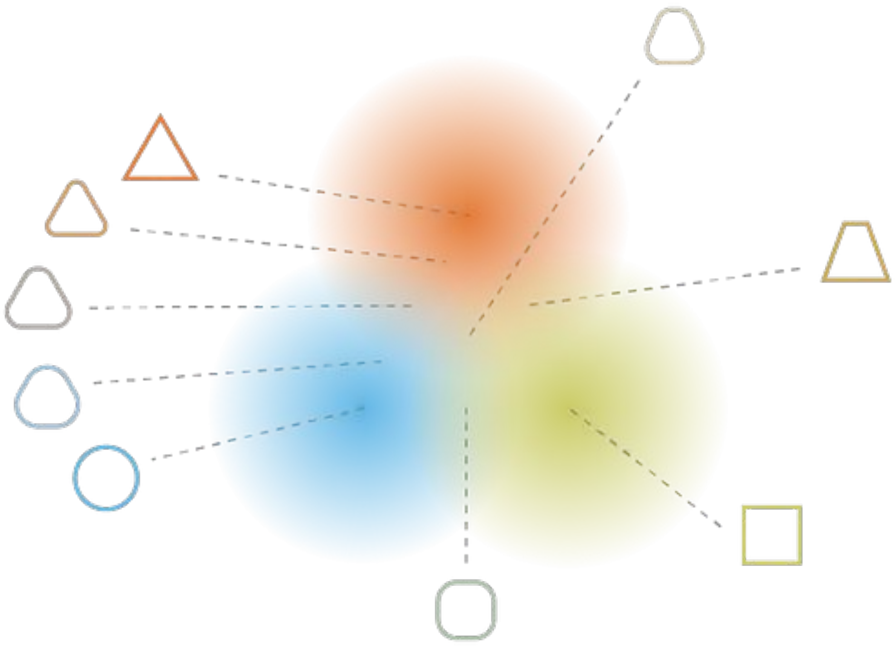
- Continuity: Two points that are close in the latent space should be decoded into similar content.
- Complete: For any point within the latent space we should obtain reasonable content when decoded

Variational Autoencoders

Latent space intuition, what are good properties?

- Probability distribution regularization: In order to obtain a latent space with these properties we need to regularized the probability distributions being used in order to normalized them.

- Lots of math behind these ideas.

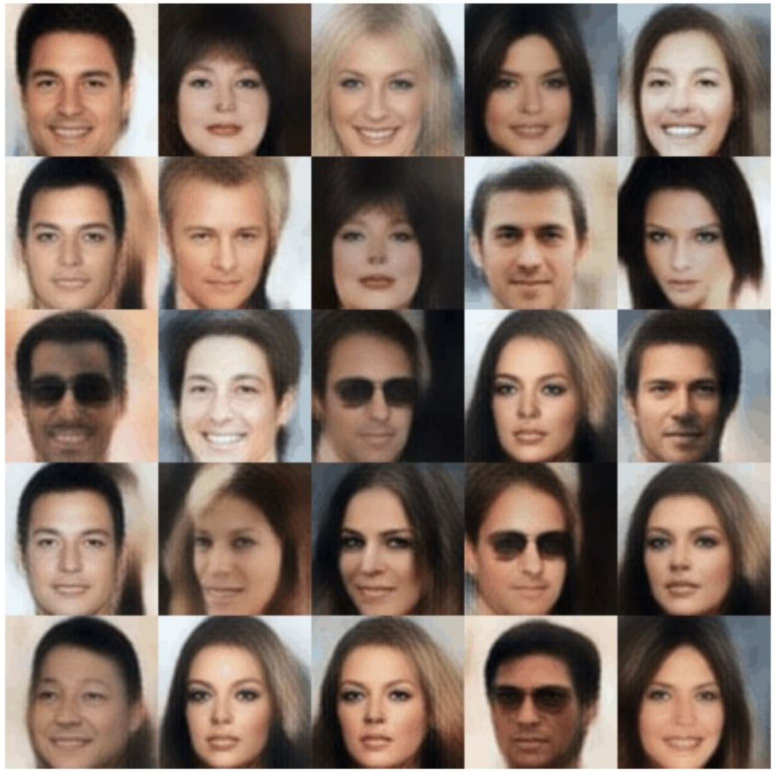


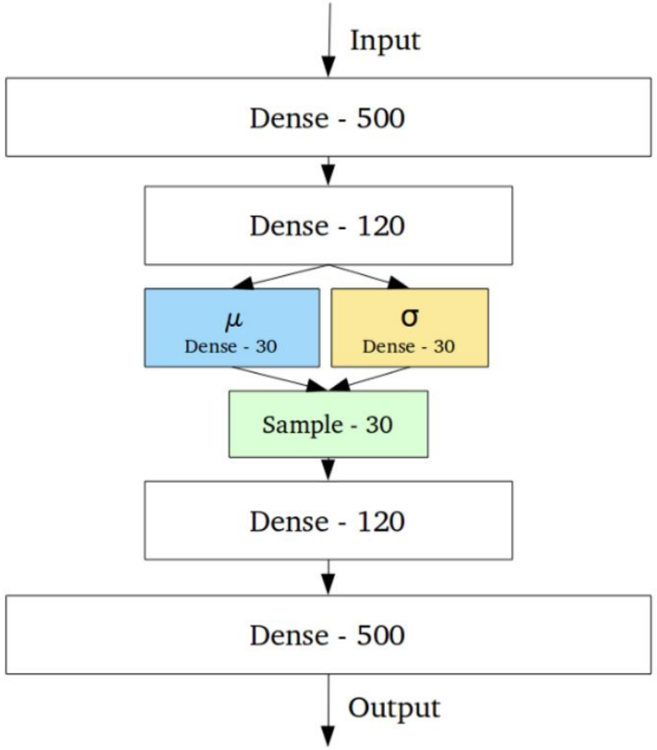
-

Variational Autoencoders

A variation of standard autoencoders with the aim of generating new data.

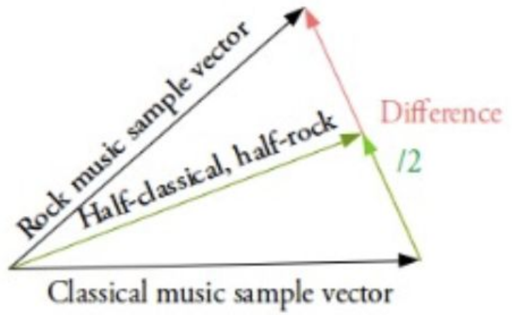
We force the 'latent space' to be continuous in order to generate new data



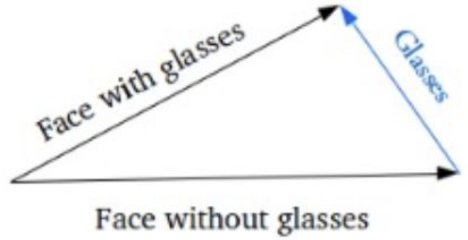


https://medium.com/vitrox-publication/generative-modeling-with-variational-auto-encoder-vae-fc449be9890e

Variational Autoencoders



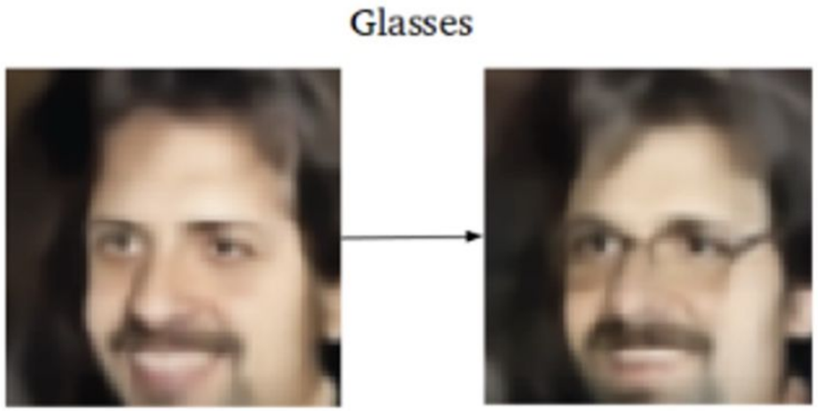
Interpolating between samples



Adding new features to samples

For example, if you wish to generate a new sample halfway between two samples, just find the difference between their mean (μ) vectors, and add half the difference to the original, and then simply decode it.

What about generating *specific features*, such as generating glasses on a face? Find two samples, one with glasses, one without, obtain their encoded vectors from the encoder, and save the difference. Add this new "glasses" vector to any other face image, and decode it.



Variational Autoencoders

VAEs in real life:

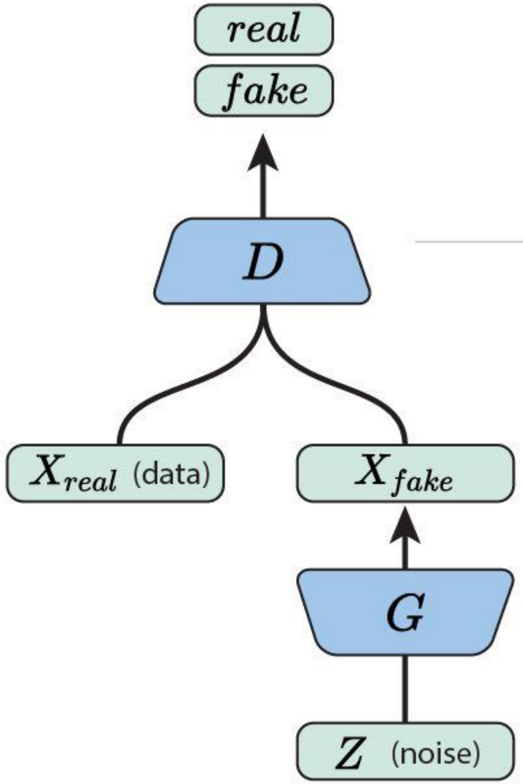
https://www.tensorflow.org/tutorials/generative/cvae

Outline

1. Recap
2. Unsupervised Learning
3. Autoencoders
4. Variational Autoencoders
5. **GANs**

Generative Adversarial Networks

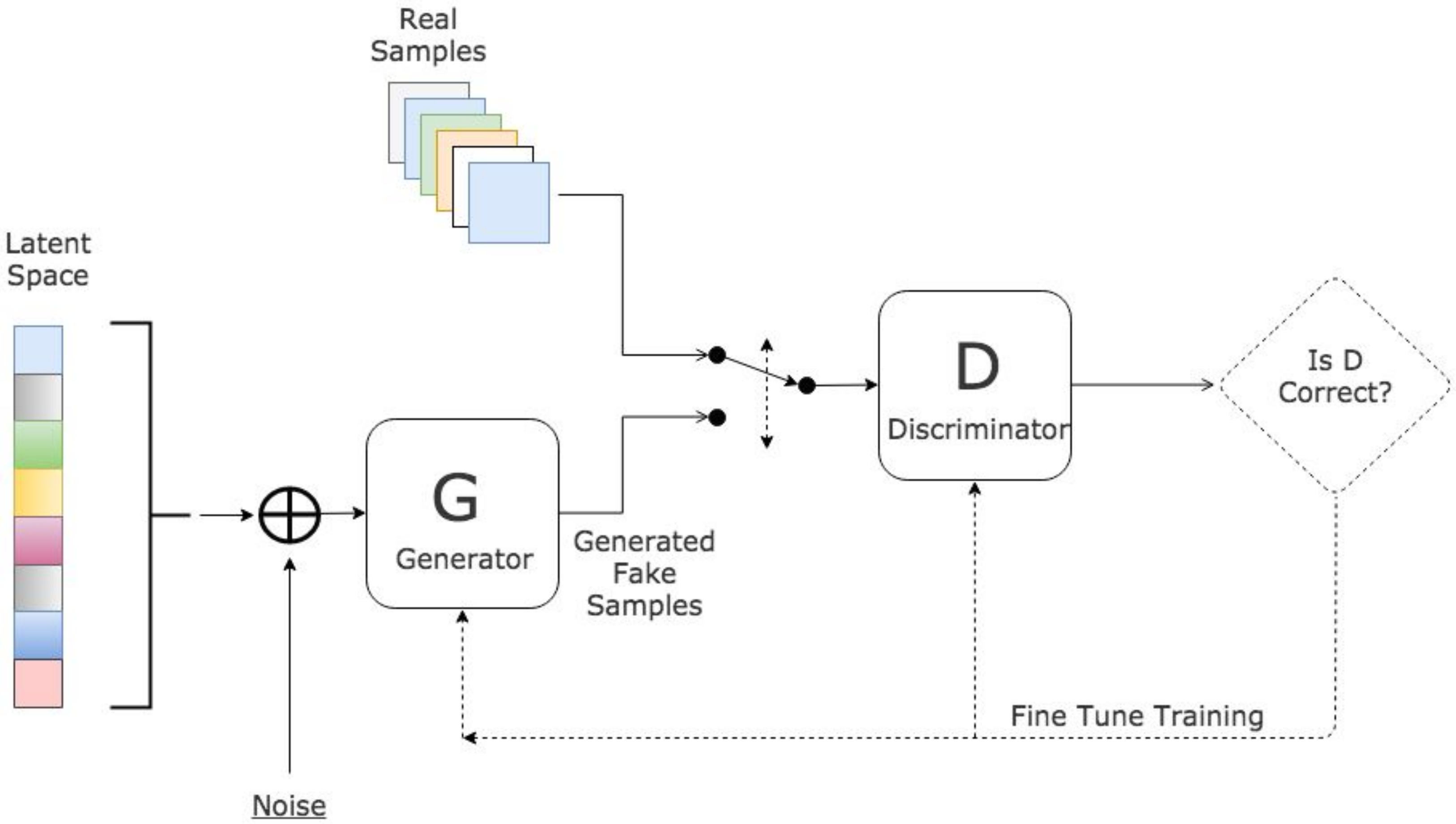
Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.



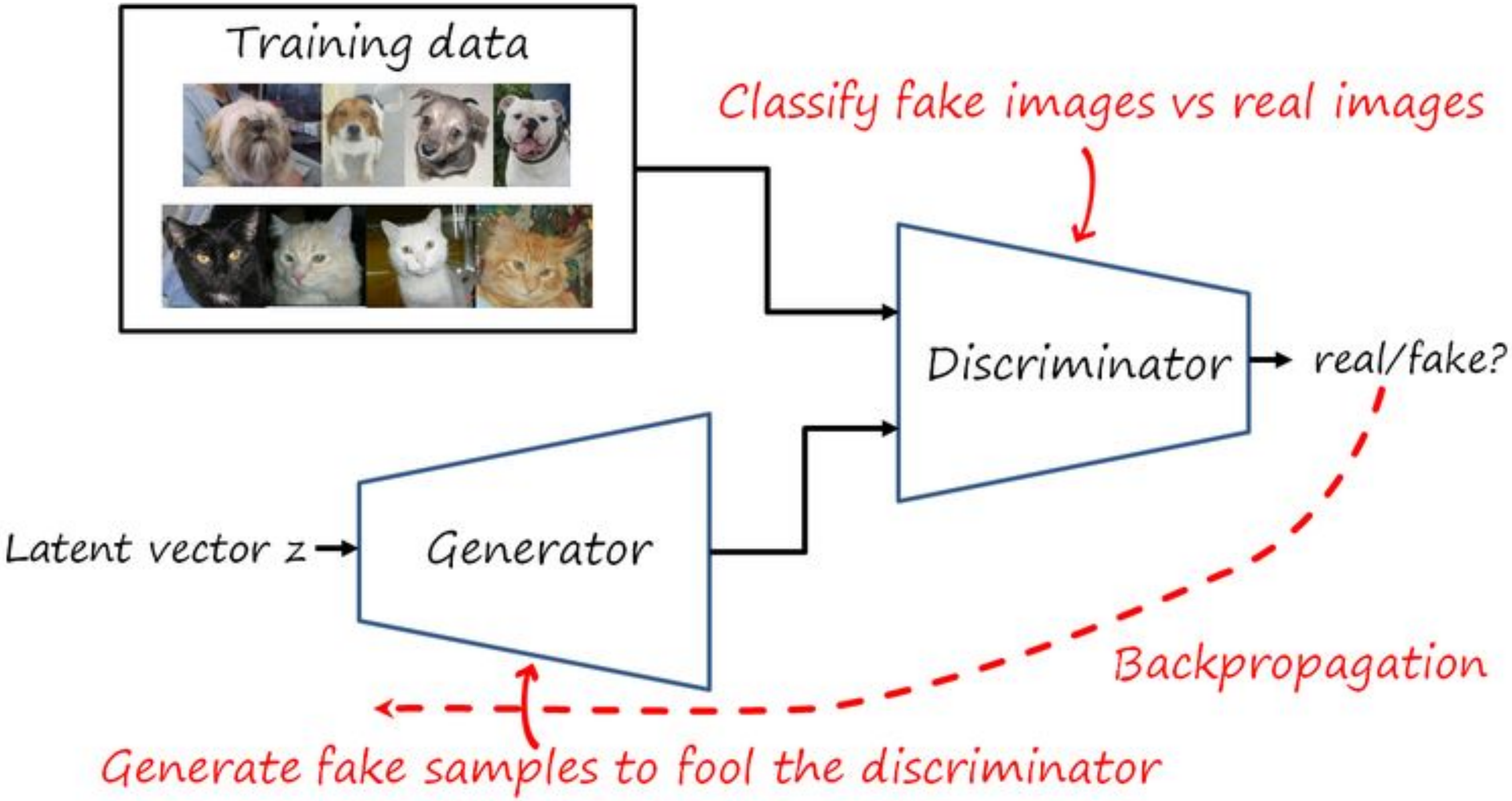
The **discriminator** tries to distinguish genuine data from forgeries created by the generator.

The **generator** turns random noise into immitations of the data, in an attempt to fool the discriminator.

Generative Adversarial Networks



Generative Adversarial Networks



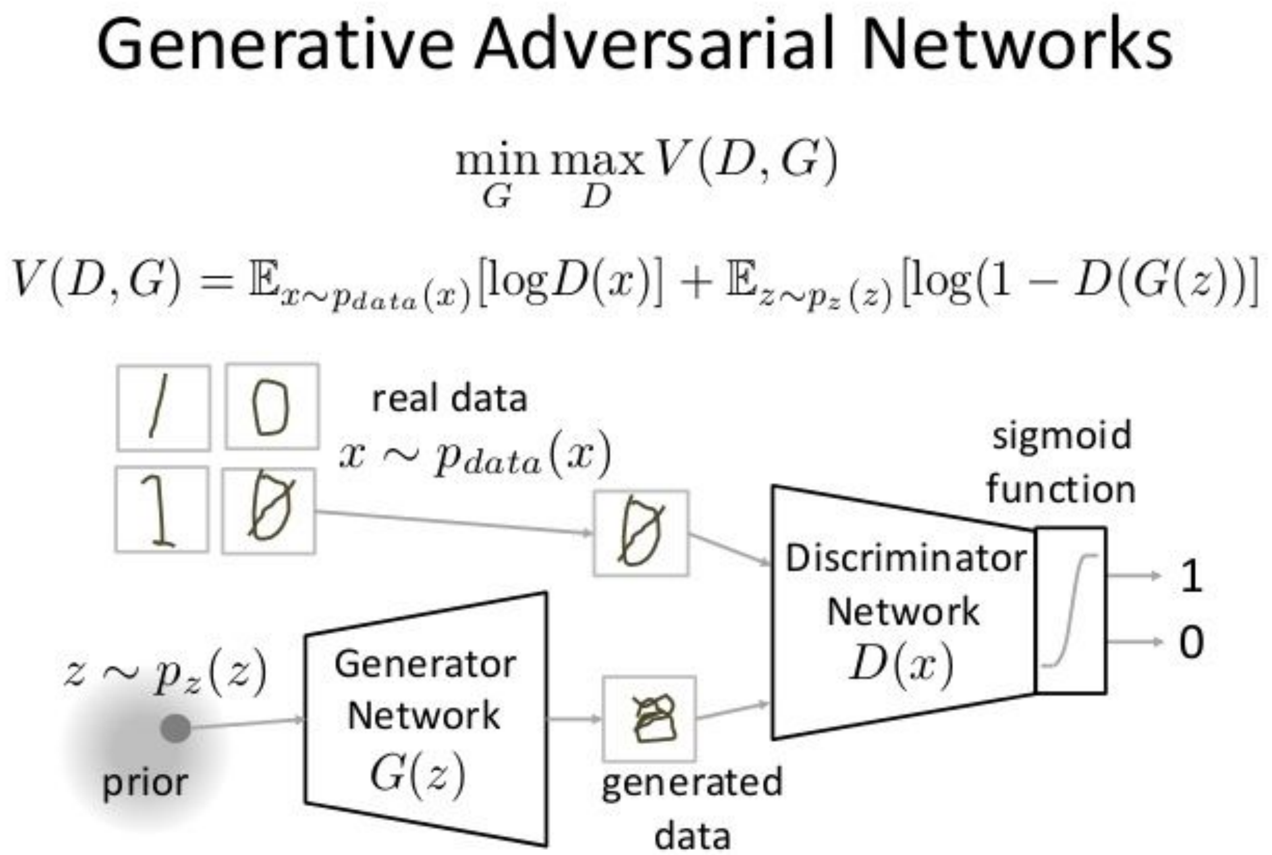
Generative Adversarial Networks: Training

La función de coste es un juego min-max donde:

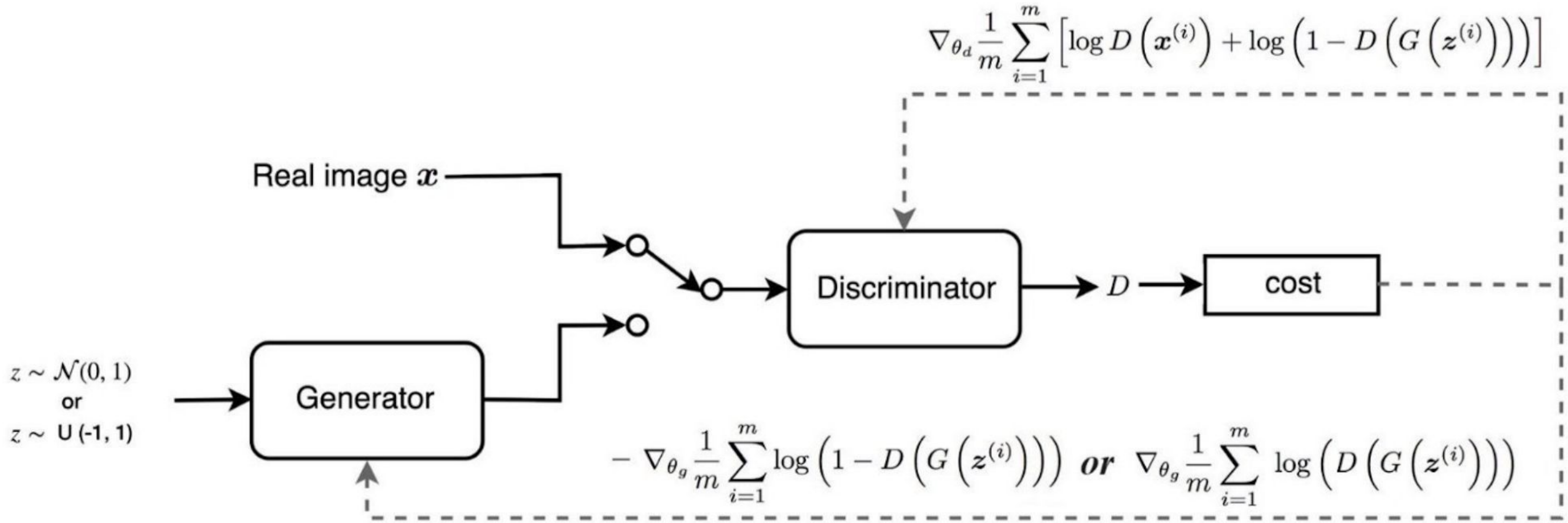
- El Discriminador intenta discernir muestras sintéticas (0) vs reales (1)

- El generador intenta engañar al discriminador, es decir, quiere que el discriminador se confunda.

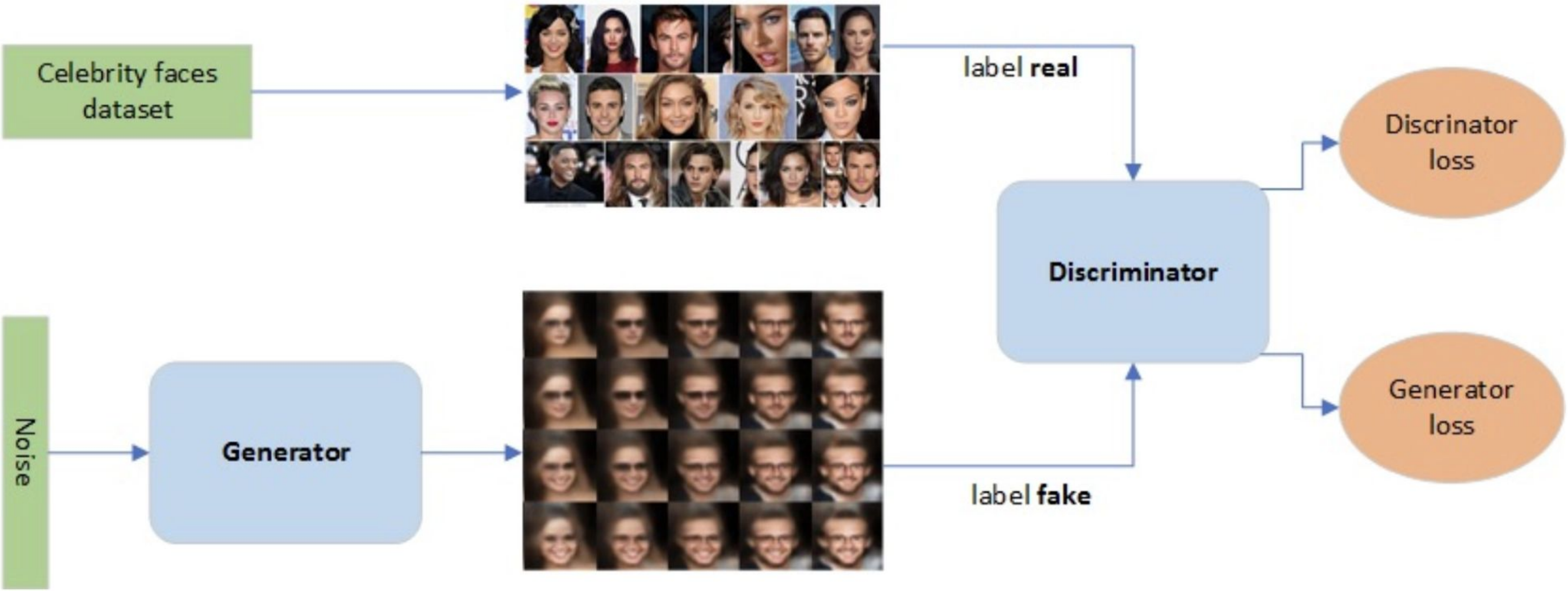
Tras entrenar la arquitectura, si converge, el discriminador terminará sacando 0.5 para cualquier imagen, real o fake, ya que no podrá discernir entre unas y otras.



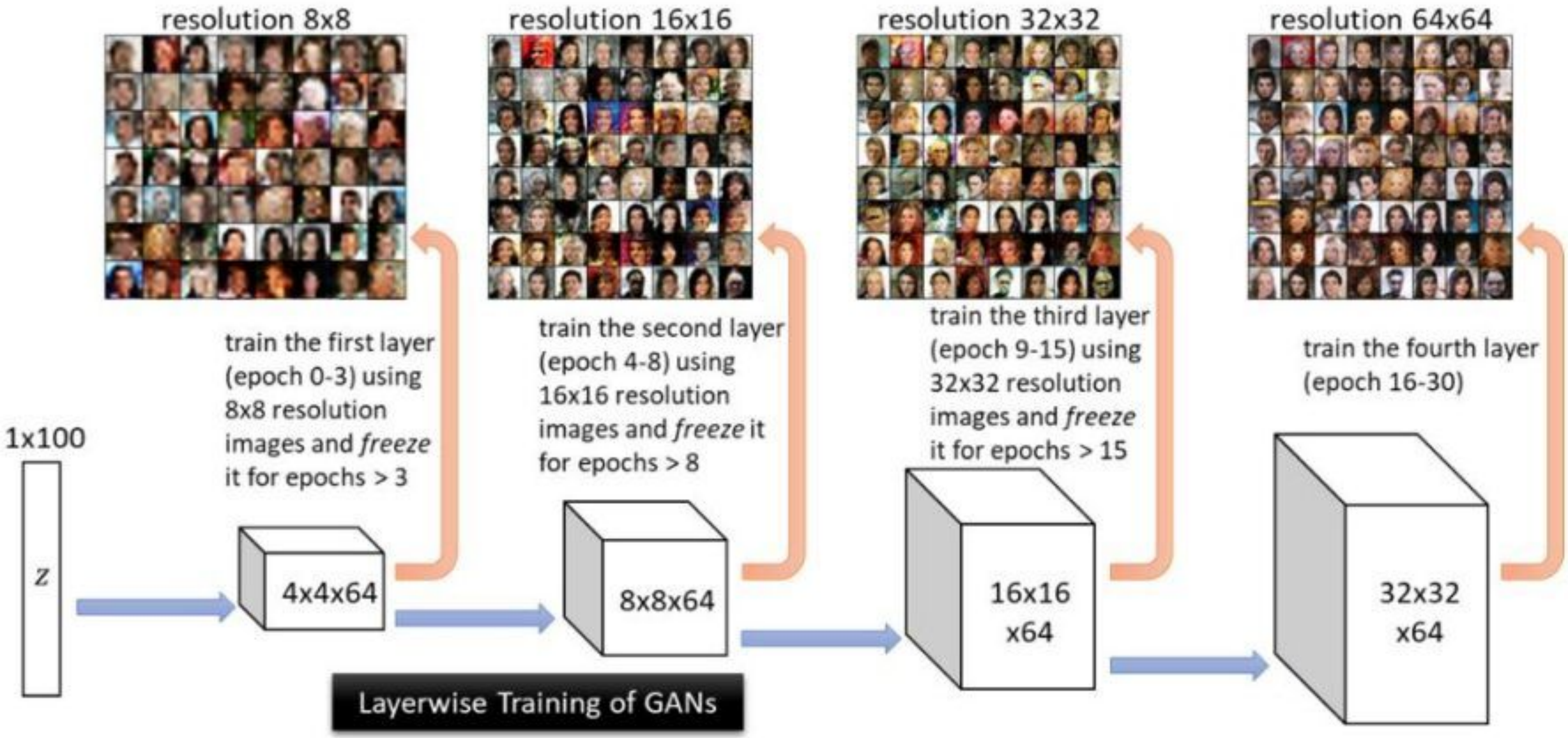
Generative Adversarial Networks: Training



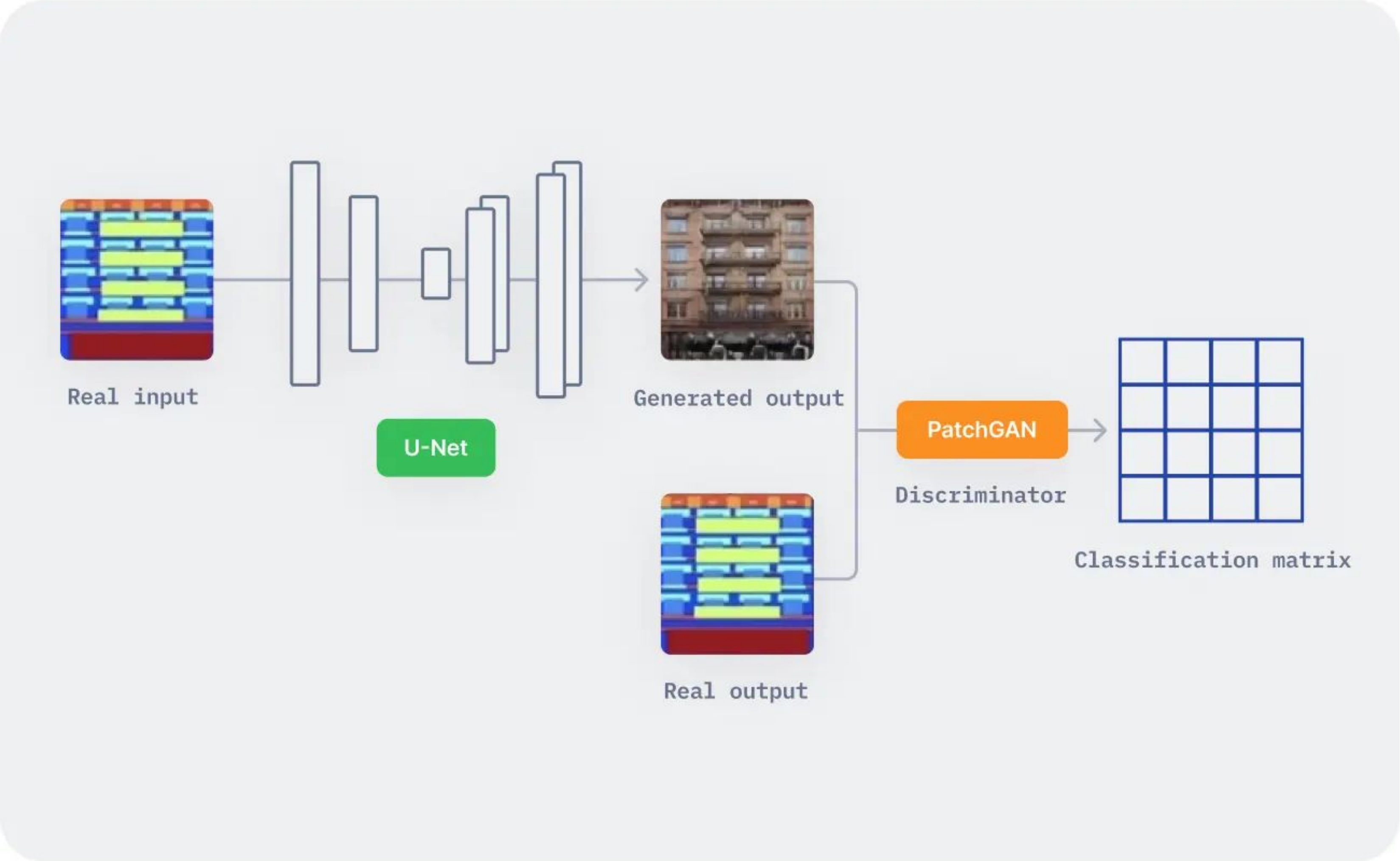
Generative Adversarial Networks: Example



Generative Adversarial Networks: Advanced



Generative Adversarial Networks: Advanced



V7

Outline

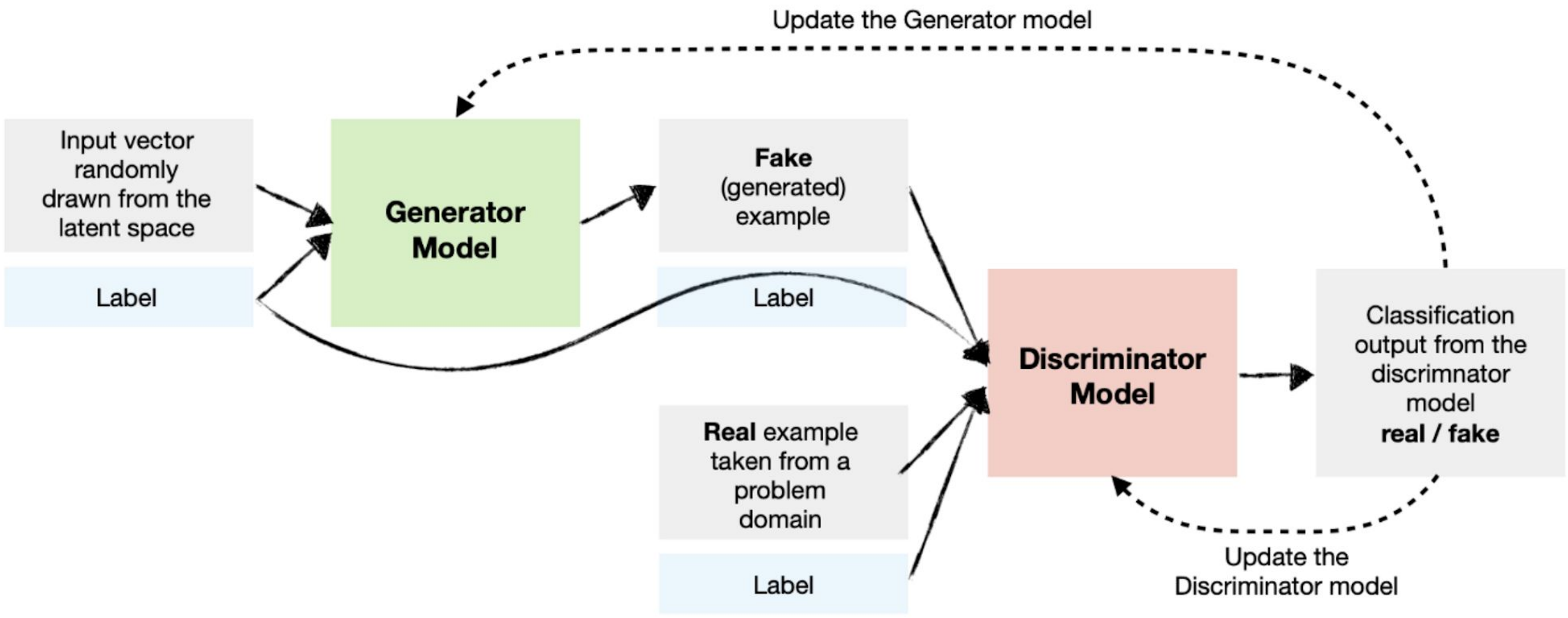
1. Recap
2. Unsupervised Learning
3. Autoencoders
4. Variational Autoencoders
5. **GANs**
   a. **Conditional GANs**

Conditional GANs: Motivation

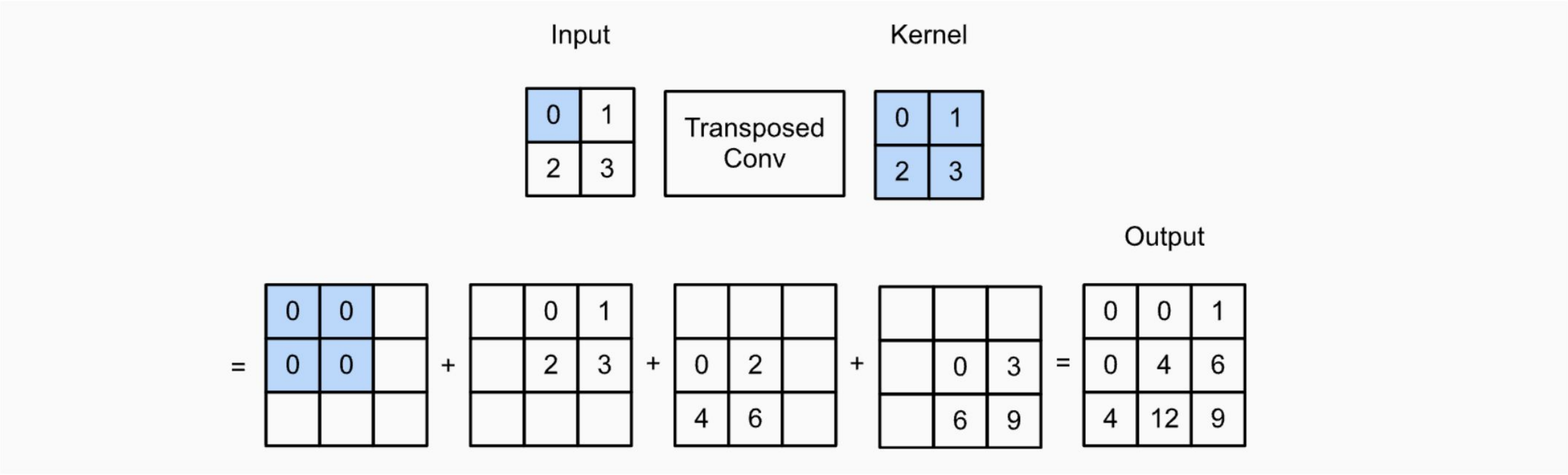
Can we build a fake identity?

thispersondoesnotexist

Conditional GANs



Generative Adversarial Nets



Transposed Convolution (Stride 2)

Generative Adversarial Nets

Conv2DTranspose layer

Conv2DTranspose class [source]

```
tf.keras.layers.Conv2DTranspose(
    filters,
    kernel_size,
    strides=(1, 1),
    padding="valid",
    output_padding=None,
    data_format=None,
    dilation_rate=(1, 1),
    activation=None,
    use_bias=True,
    kernel_initializer="glorot_uniform",
    bias_initializer="zeros",
    kernel_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    bias_constraint=None,
    **kwargs
)
```