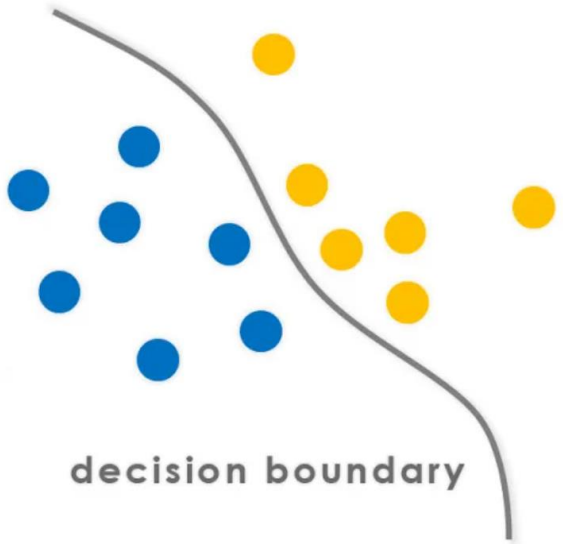
Generative Adversarial Networks (GAN)

Speaker: Guanyu Hu
School of Computer Science and Technology
Xi'an Jiaotong University

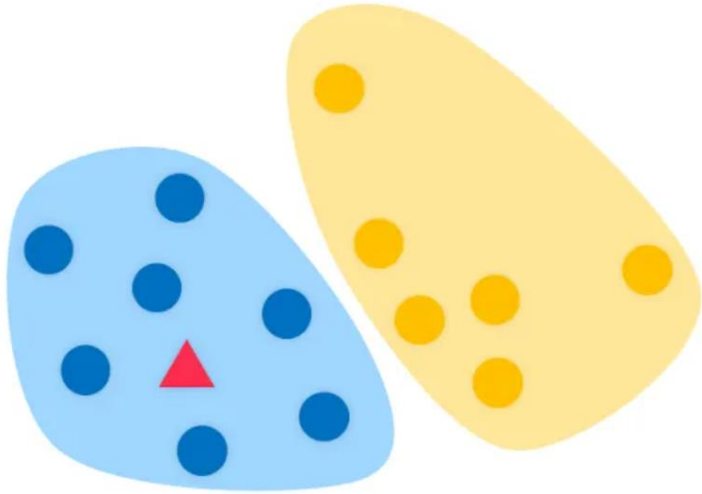
Outline

- **Generative Model**
  - What Are Generative Models
  - Types of Deep Generative Models
- **Generative Adversarial Networks (GAN)**
  - Discriminator
  - Generator
  - BCE Cost Function
  - Training GANs

What Are Generative Models



vs.



Discriminative Models

Generative Models

What Are Generative Models

Discriminative Models

How:

- 直接根据特征 X，来对 Y 建模，划定一个整体判别边界，每新来一个数据 X，就根据这个边界来判断它应该属于哪一类。
- 获得样本 X 属于类别 Y 的概率分布，是一个条件概率 $P(Y|X)$

Goal:

- 给定一组 feature X 来判定它是属于哪个分类 Y 的

Generative Models

How:

- 观察训练数据 X与 Y 的整体分布，求得联合概率分布 $P(X,Y)$，每新来一个数据 X，求出 X 与不同分类Y 之间的联合概率分布，将 X 分为联合概率大的那一类。

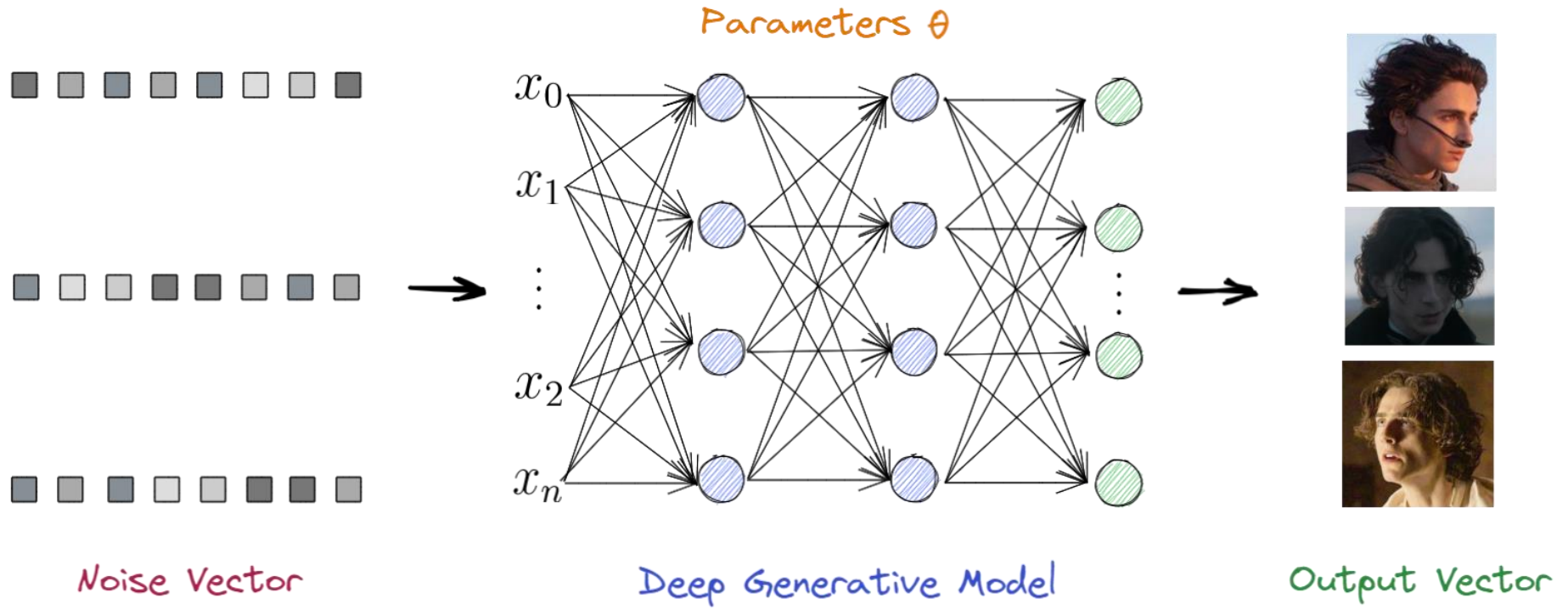
Goal:

- 给定一组 feature X 计算出与不同分类的联合概率 $P(X,Y)$

Methods:

- **朴素贝叶斯方法，隐马尔可夫模型**

What Are Generative Models



- Take some random input represented by the noise
- From these inputs, to generate a set of features X that look like a realistic representation of class Y

Outline

- **Generative Model**
  - **What Are Generative Models**
  - Types of Deep Generative Models
- **Generative Adversarial Networks (GAN)**
  - Discriminator
  - Generator
  - BCE Cost Function
  - Training GANs

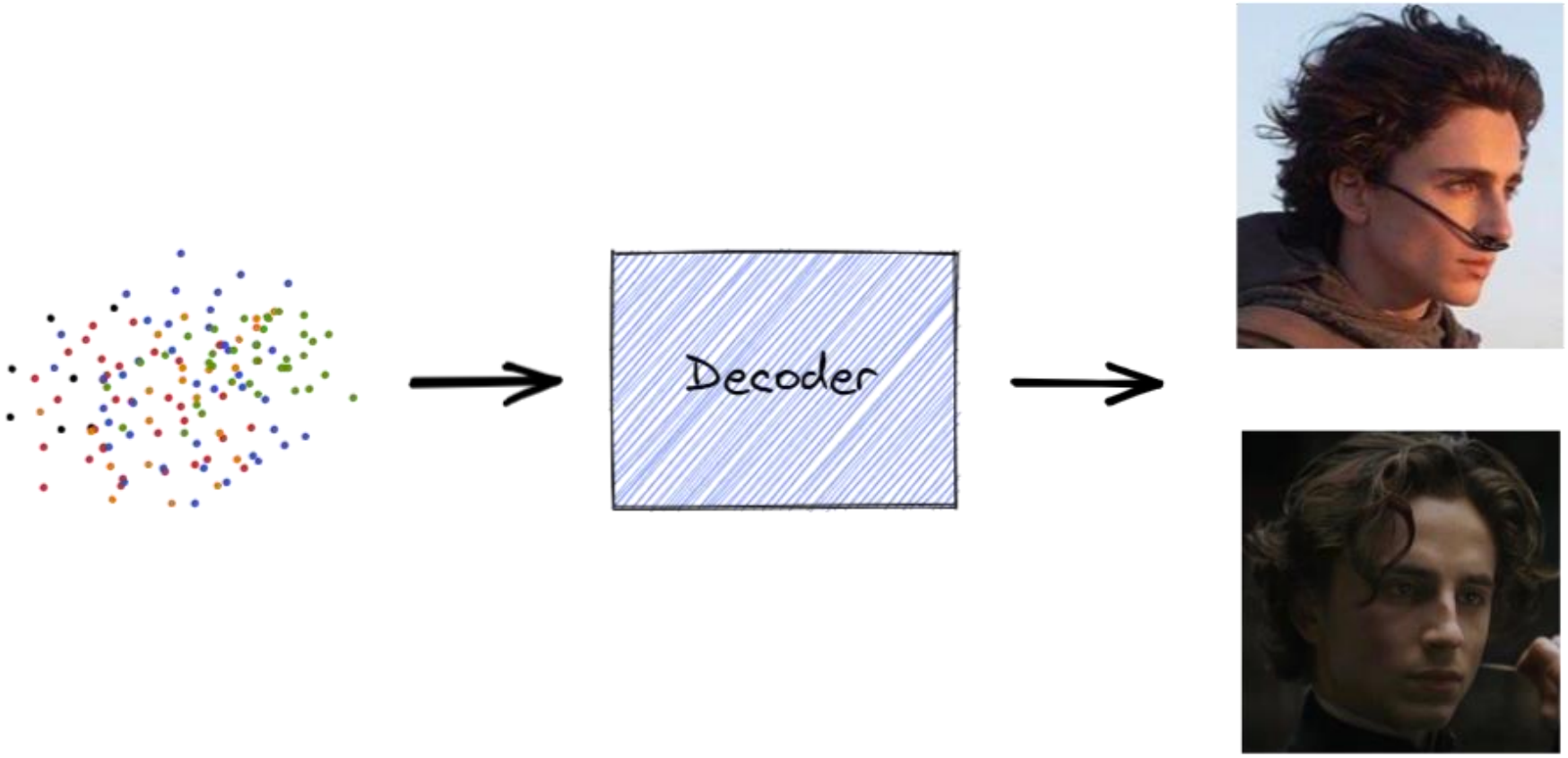
Outline

- **Generative Model**
  - What Are Generative Models
  - **Types of Deep Generative Models**
- **Generative Adversarial Networks (GAN)**
  - Discriminator
  - Generator
  - BCE Cost Function
  - Training GANs

Variational Autoencoders (VAE)

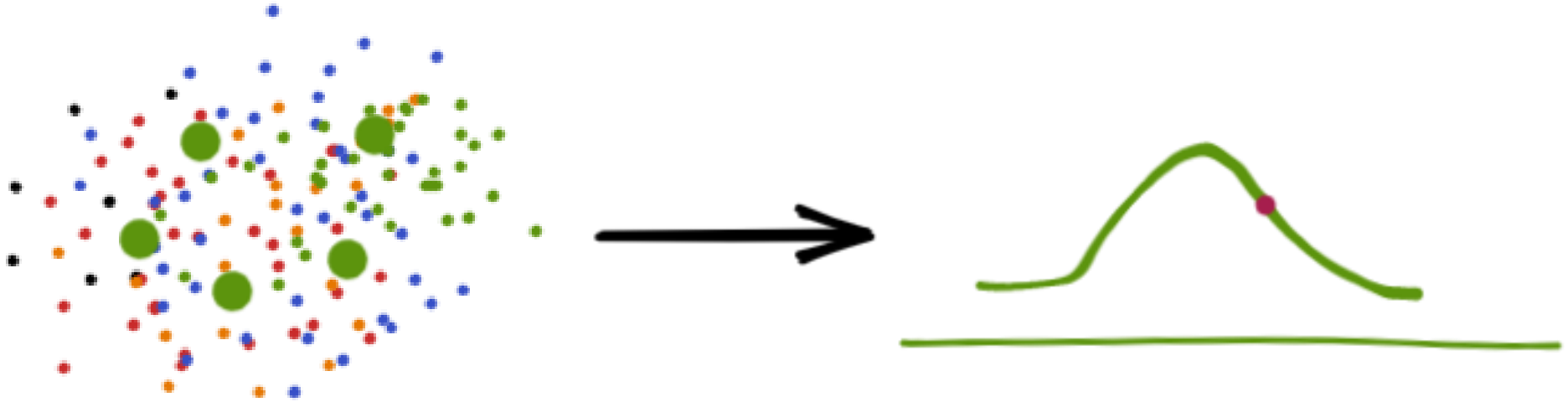
Types of Deep Generative Models

Variational Autoencoders (VAE)

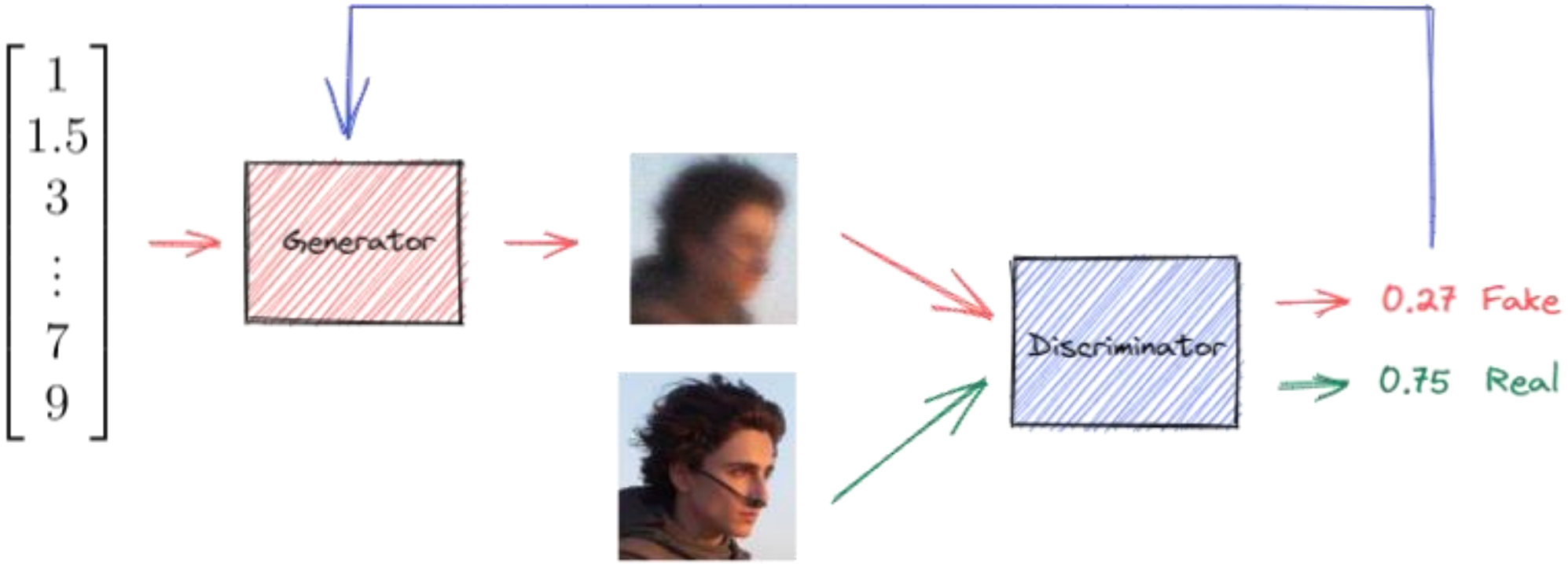


Types of Deep Generative Models

Variational



Generative Adversarial Networks (GAN)



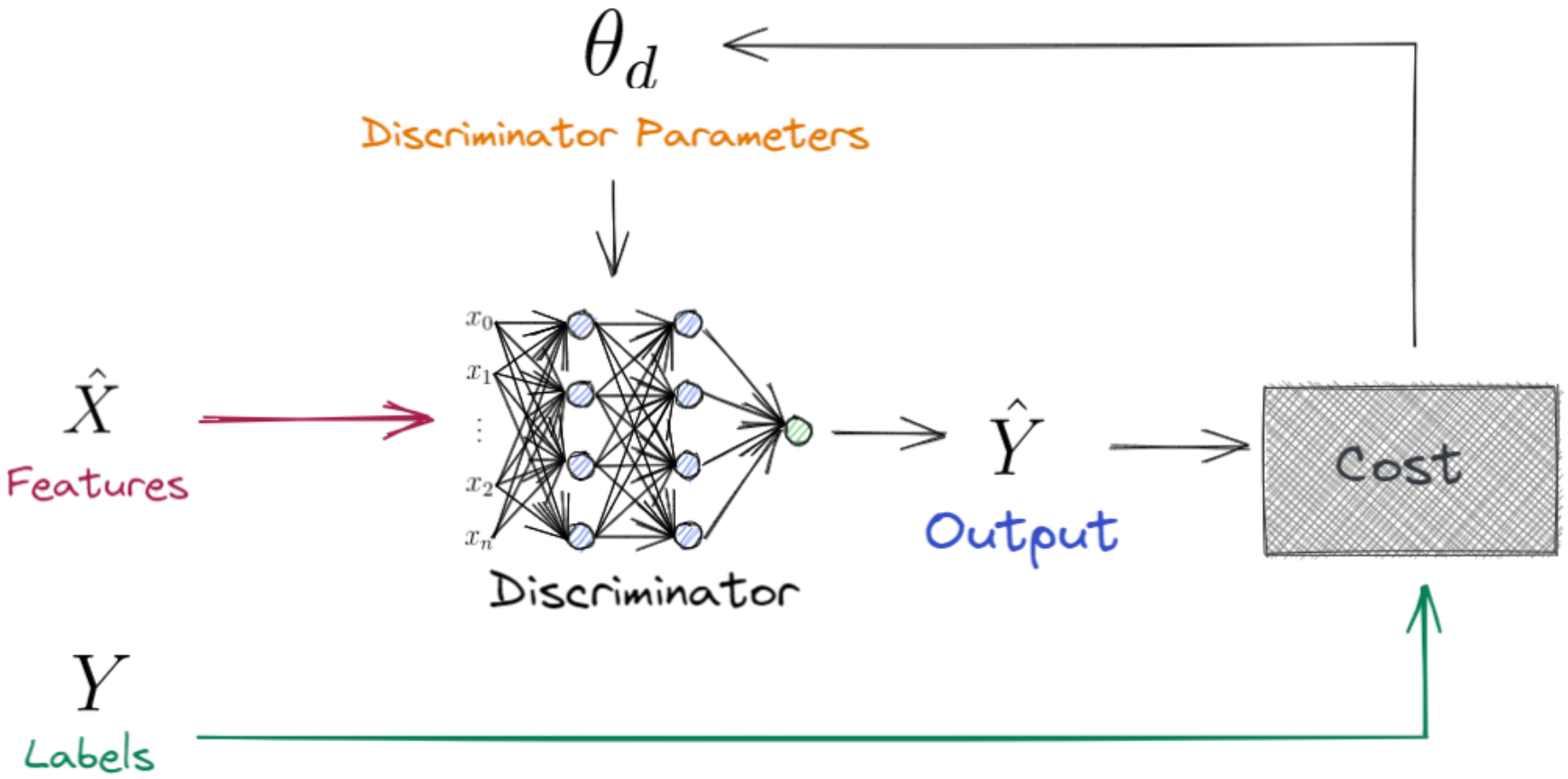
Outline

- **Generative Model**
  - What Are Generative Models
  - **Types of Deep Generative Models**
- **Generative Adversarial Networks (GAN)**
  - Discriminator
  - Generator
  - BCE Cost Function
  - Training GANs

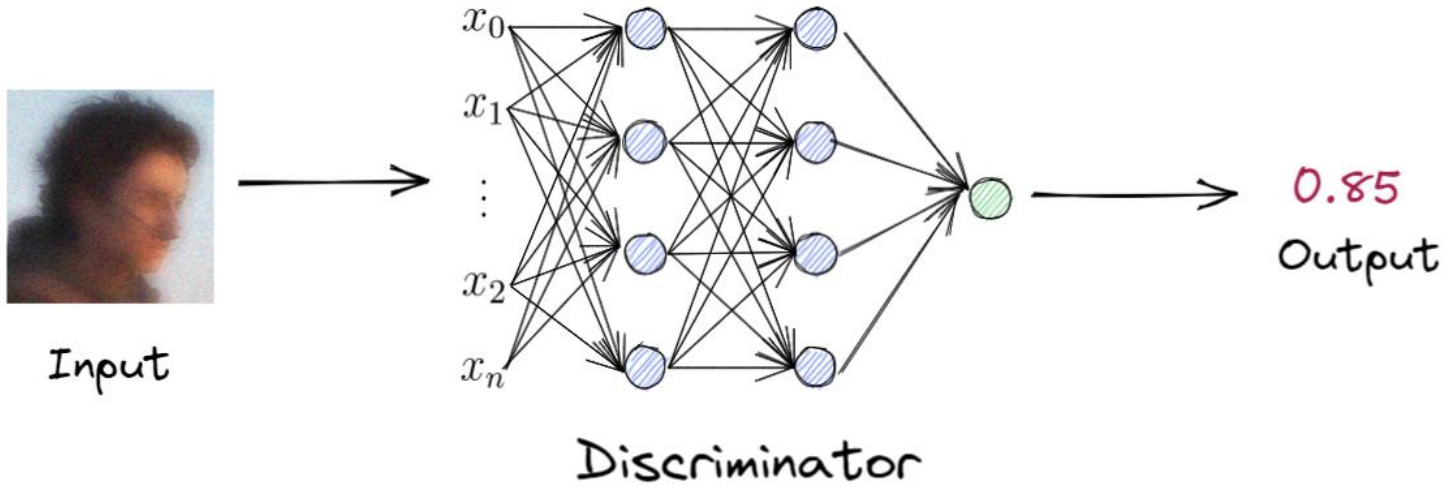
Outline

- Generative Model
  - What Are Generative Models
  - Types of Deep Generative Models
- **Generative Adversarial Networks (GAN)**
  - **Discriminator**
  - Generator
  - BCE Cost Function
  - Training GANs

Discriminator for Classification



Discriminator for Generation



$$P(\underbrace{Y}_{\text{Class}} | \underbrace{X}_{\text{Features}}) \longrightarrow P(\underbrace{\text{Fake}}_{\text{Class}} | \underbrace{\text{image}}_{\text{Features}}) = 0.85 \longrightarrow \boxed{\text{Fake}}$$

Outline

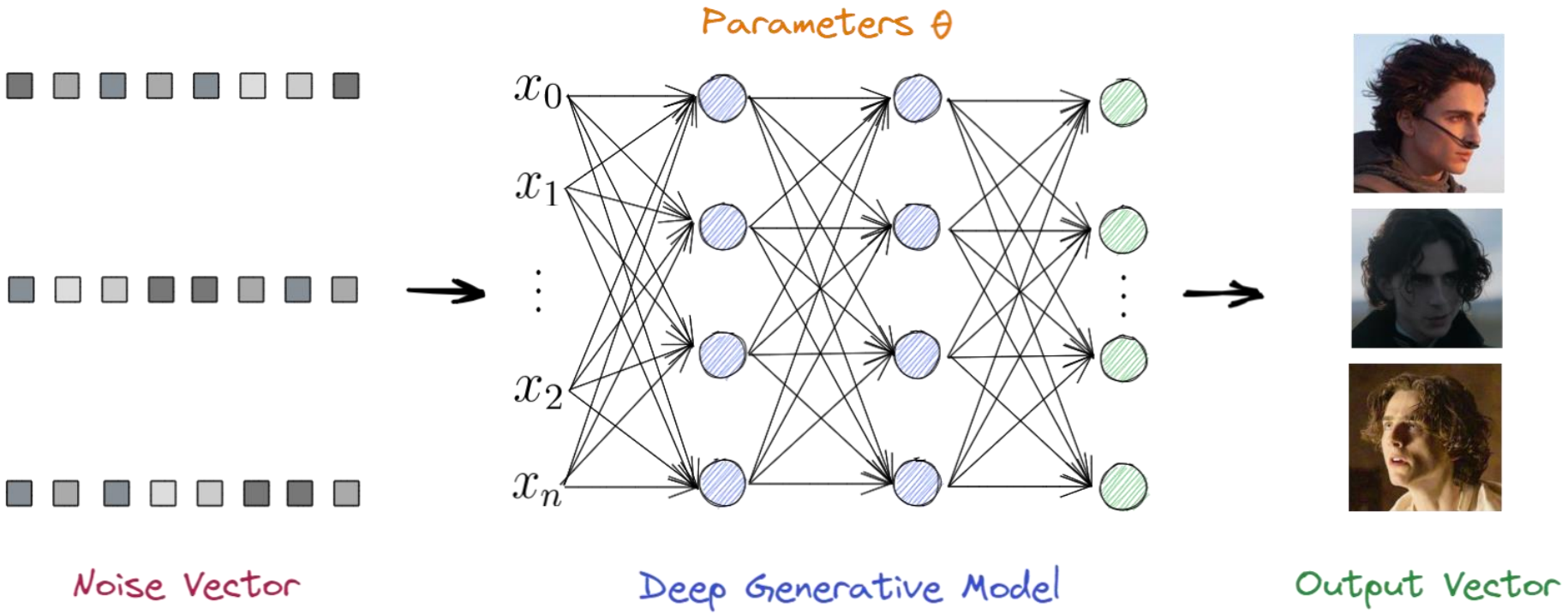
- Generative Model
  - What Are Generative Models
  - Types of Deep Generative Models
- **Generative Adversarial Networks (GAN)**
  - **Discriminator**
  - Generator
  - BCE Cost Function
  - Training GANs

Outline

- Generative Model
  - What Are Generative Models
  - Types of Deep Generative Models
- **Generative Adversarial Networks (GAN)**
  - Discriminator
  - **Generator**
  - BCE Cost Function
  - Training GANs

Generative Adversarial Networks (GAN)

Generator



Outline

- Generative Model
  - What Are Generative Models
  - Types of Deep Generative Models
- **Generative Adversarial Networks (GAN)**
  - Discriminator
  - **Generator**
  - BCE Cost Function
  - Training GANs

Outline

- Generative Model
  - What Are Generative Models
  - Types of Deep Generative Models
- **Generative Adversarial Networks (GAN)**
  - Discriminator
  - Generator
  - **BCE Cost Function**
  - Training GANs

Binary Cross Entropy function Cost Function

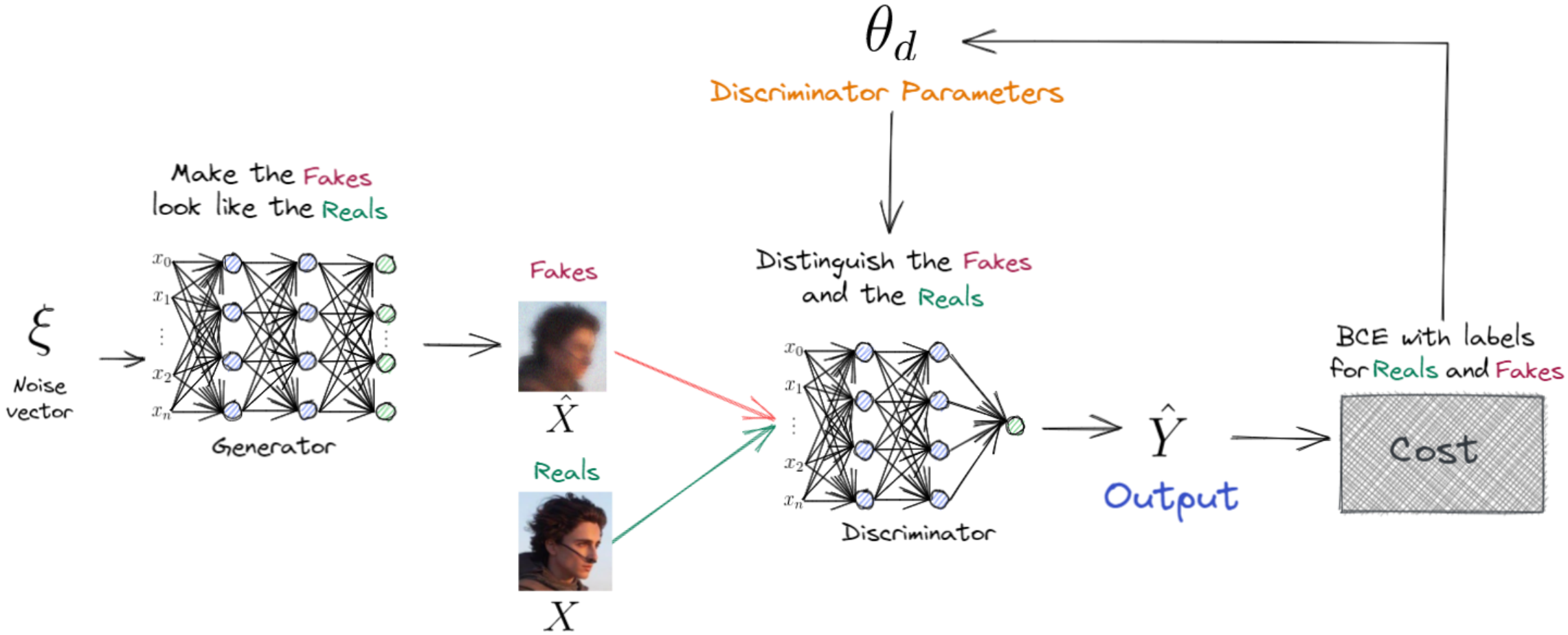
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h\left(x^{(i)},\theta\right) + \left(1-y^{(i)}\right)\log\left(1-h\left(x^{(i)},\theta\right)\right)\right]$$

- h: predictions made by the model
- y: is the labels for the different examples, true label of real fake
- x: feature passed in through the prediction, could be an image
- θ: parameters to model the classifier h

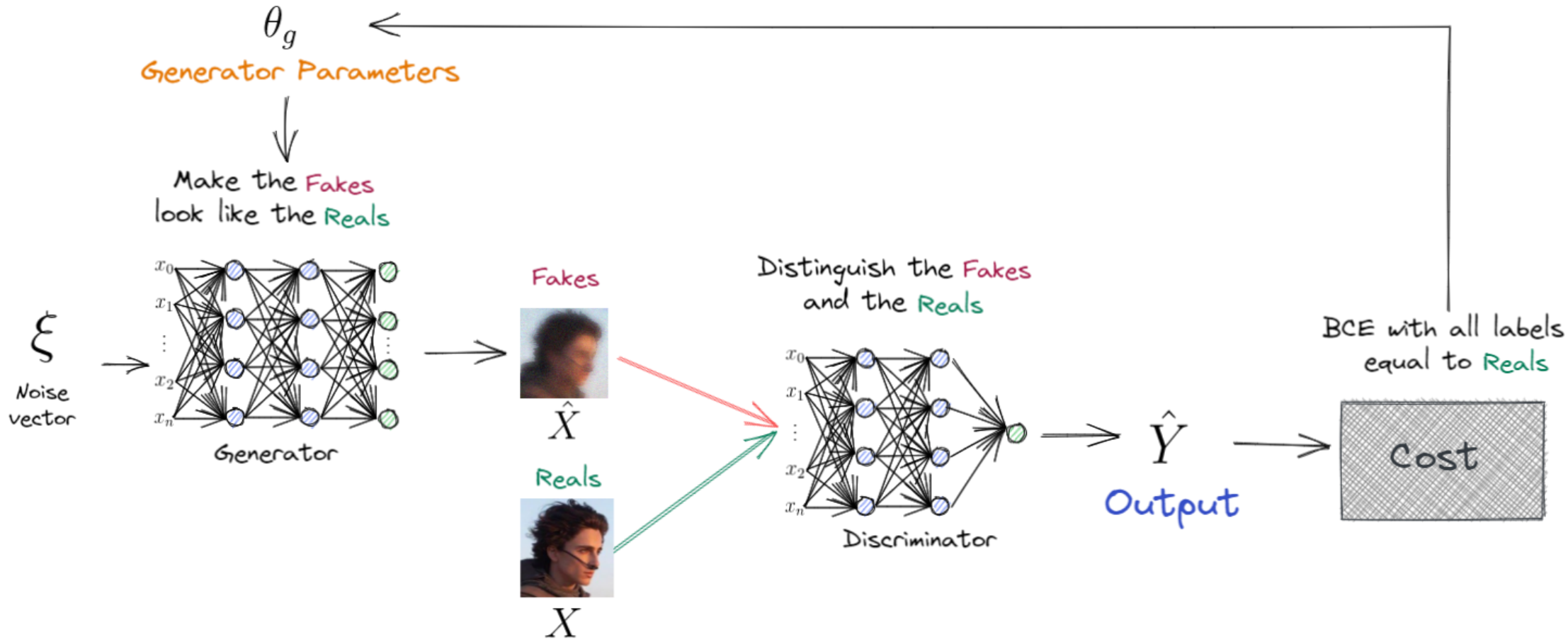
Outline

- Generative Model
  - What Are Generative Models
  - Types of Deep Generative Models
- **Generative Adversarial Networks (GAN)**
  - Discriminator
  - Generator
  - BCE Cost Function
  - **Training GANs**

Training Discriminator



Training Generator



Use Saved Model