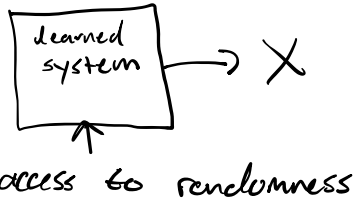
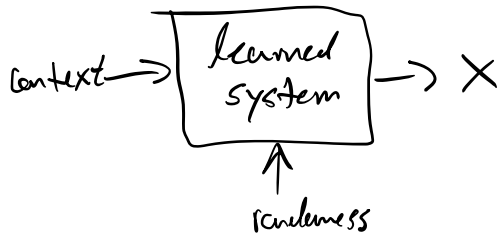


generative models

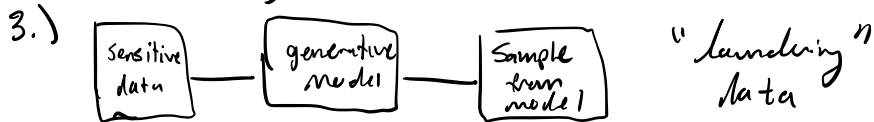
generation task: create a new example(s) of X



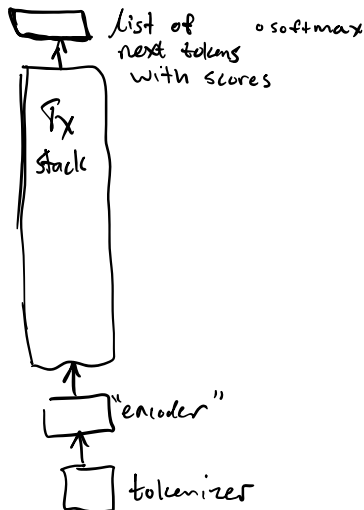
- Conditional generation:



- Uses:
- 1.) direct purpose of generation
 - 2.) create training data (like data augmentation)



GPT:



aside: how to sample randomly:

$$X = F_Y^{-1}(U) \quad Y \sim f_Y(\cdot)$$


Suppose $f_x(\vec{x}) = f_{x_1}(x[1]) \cdot f_{x_2|x_1}(x[2]|x[1]) \dots$

- 1.) start with $\vec{u} \sim \text{unif}(0,1)$ d times
- 2.) $u[1]$ to get $x[1]$ using $F_{x_1}(\cdot)$
- 3.) $u[2]$ to get $x[2]$ using $F_{x_2|x_1=x[1]}(\cdot)$
- \vdots Repeat d -times

→ "Auto-regressive Generation"

AR for image

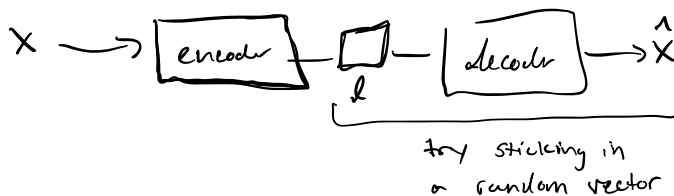
⇒ demand one pixel at a time
- need an ordering

Idea: 

↳ challenge: time complexity

Other generative models:

Naive → just use an autoencoder



other naive → use a classifier



start with random noise, gradient descent to maximize classifier

→ surprisingly, it looks like noise

→ on images, network outputs itself and fools classifier

GAN approach \rightarrow generator and classifier together during training

Classifier: Real vs. fake