INTRODUCTION TO MACHINE LEARNING COMPSCI 4ML3

Lecture 28

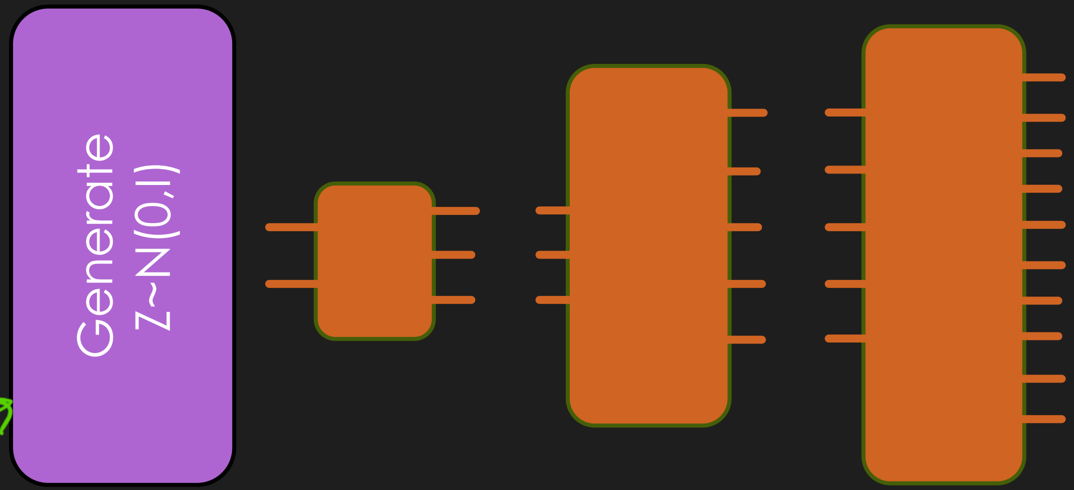
Hassan Ashtiani

MAXIMIZE LIKELIHOOD?

$\Pr[\hat{X} =$ ☹ $]$

- $\underset{\theta}{\text{MAX}} \sum_x \widehat{p(x)}$?

log(   )



$\hat{X}$ image

→ $P(X|Z)$: easy ✓

$P(X)$ ✗ hard

$\int_Z P(X|Z)P(Z) = P(X)$

integration is slow...

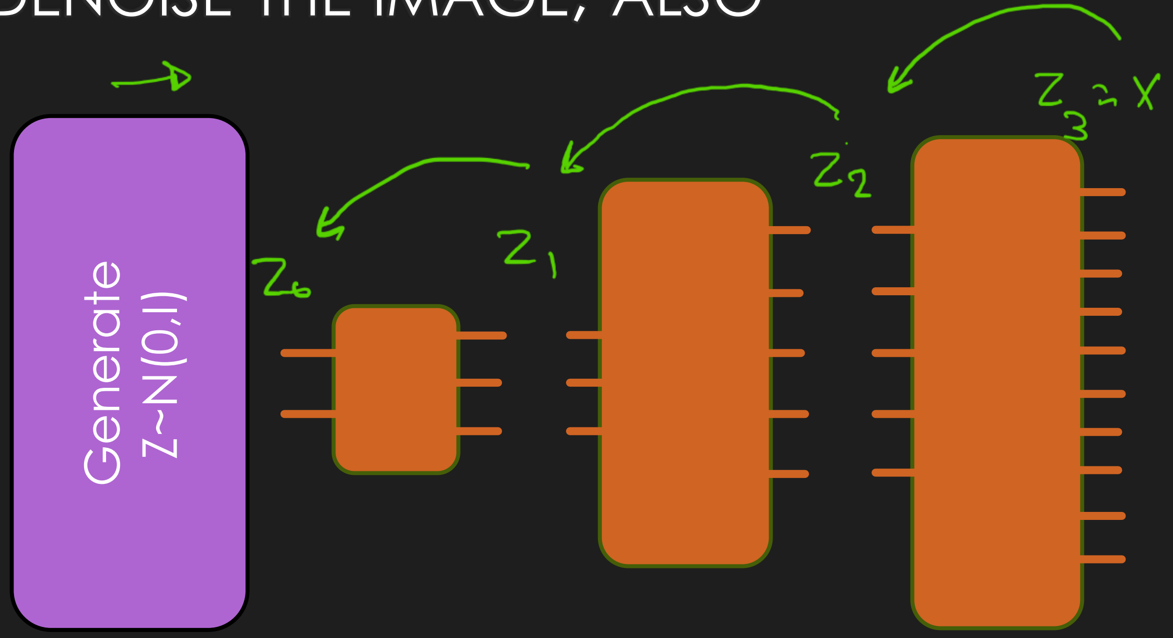
TEXT TO IMAGE MODELS

- SUCCESS DUE TO: LLMs + DIFFUSION MODELS

- DALL-E 2 IMAGEN

DIFFUSION MODELS

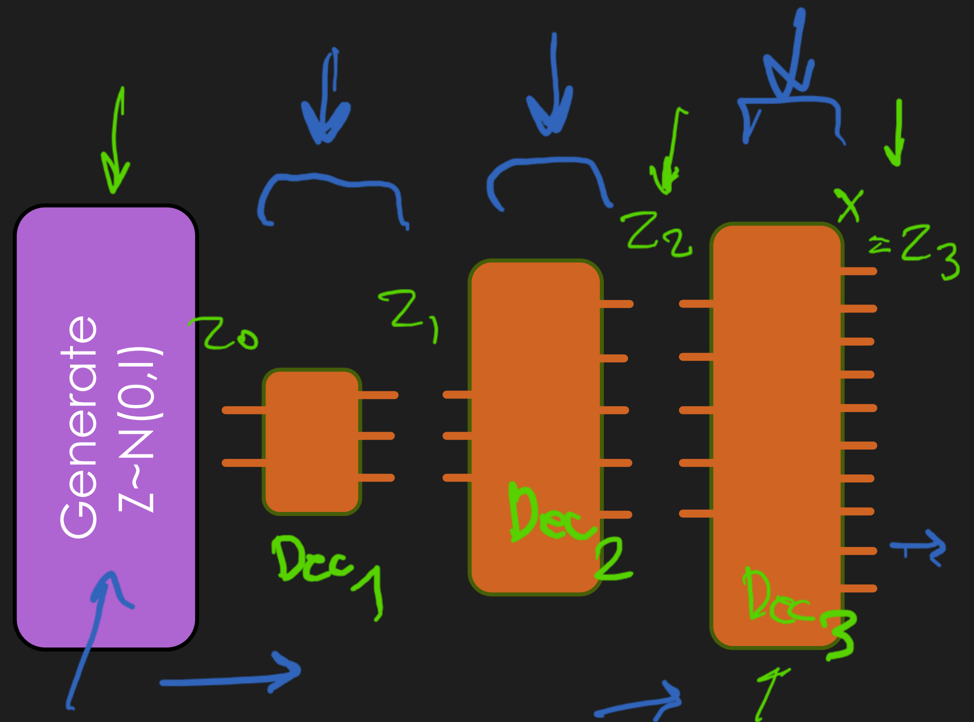
- VIEW THE NETWORK IN TWO DIRECTIONS:
  - RIGHT TO LEFT: PROGRESSIVELY ADD MORE NOISE, ALSO MAKE THE IMAGE SMALLER IN RESOLUTION
  - LEFT TO RIGHT: PROGRESSIVELY DENOISE THE IMAGE, ALSO MAKE IT HIGH RESOLUTION



TRAINING DIFFUSION MODELS

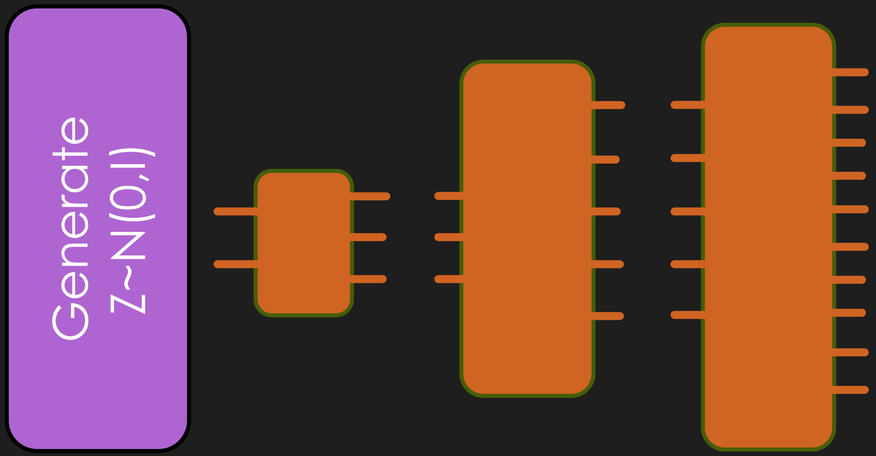
- $t \in [T]$ REPRESENTS A LAYER OF THE NETWORK
- PROGRESSIVELY ADDING NOISE TO AN IMAGE:
  - $z_T = x$
  - $z_{t-1} = Noisy(z_t)$ FOR ALL t
  - $z_0 = Noisy(z_1)$
- PROGRESSIVELY DENOISING AN IMAGE:
  - $\hat{z}_t = Dec_t(z_{t-1})$
- GENERATE AN IMAGE FROM $z_0 \sim N(0, I)$
  - $\hat{x} = Dec(z_0) = Dec_T(\ldots Dec_2(Dec_1(z_0))\ldots)$
- TRAINING DENOISERS
  - $\sum_t w_t \|Dec_t(z_{t-1}) - z_t\|_2^2$

loss for one data point



CONDITIONING IMAGE GENERATION

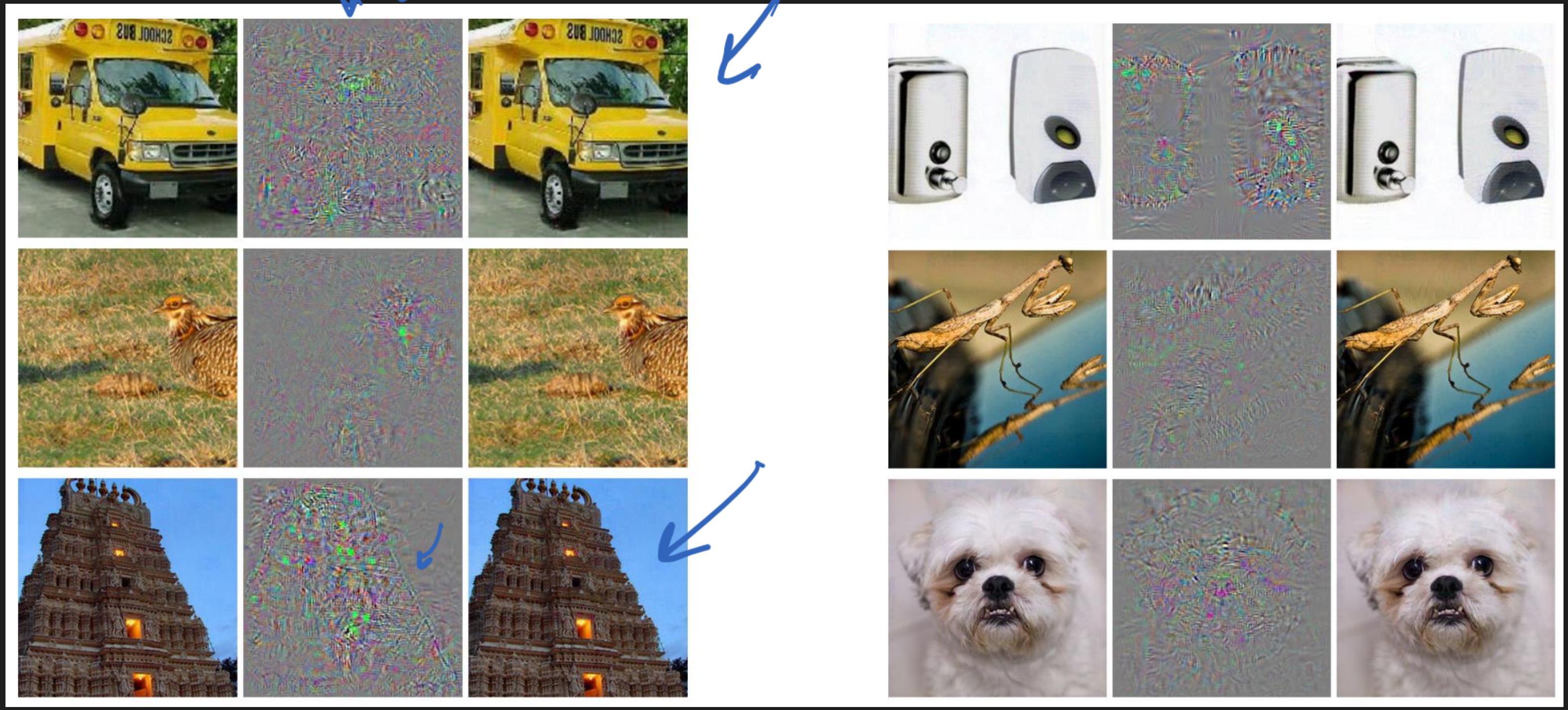
- How to control what the model generates?
  - E.g., force to generated image from class c?
  - Assume we have labeled data (x, c)
- Denoiser receives label as well
  - $\widehat{z_t} = Dec_t(z_{t-1}, c)$
- Training denoisers
  - $\sum_{(x,c)} \sum_t w_t \|Dec_t(z_{t-1}, c) - z_t\|_2^2$
- IMAGEN: randomly drop c 10% of the times during training
- For text to image models, c can be the latent representation of text rather than just the label.



ADVERSARIAL PERTURBATIONS

TYPICAL CLASSIFIERS ARE SENSITIVE TO (IMPERCEPTIBLE) "ADVERSARIAL" PERTURBATIONS

SZEGEDY ET AL.'14



RISKS OF ADVERSARIAL PERTURBATIONS

- Vulnerable to malicious attacks
- Ignore the "invariance"/domain-knowledge

- Why are these called adversarial?
  - The noise is not random
  - Carefully selected to fool the model

Eykholt et al. '18

DECISION BOUNDARY VISUALIZATION

- SMOOTHING THE DECISION BOUNDARY HELPS

FINDING ADVERSARIAL PERTURBATIONS

- USUAL GRADIENT DESCENT TO FIND PARAMS θ
  - $\nabla_\theta\left(\mathrm{L}(\theta)\right) = \frac{1}{m}\sum_i \nabla_\theta\left(l\left(f_\theta(x^i), y^i\right)\right)$
  - $\theta = \theta - \alpha\nabla_\theta\left(\mathrm{L}(\theta)\right)$
- GRADIENT ASCENT FOR FINDING PERTURBATIONS
  - $\nabla_x\left(l(x,y)\right) = \nabla_x\left(l\left(f_\theta(x^i), y^i\right)\right)$
  - $x = x + \alpha\nabla_x\left(l(x,y)\right)$