Deep Learning

Episode 8

Generative & Unsupervised models

Yandex Data Factory

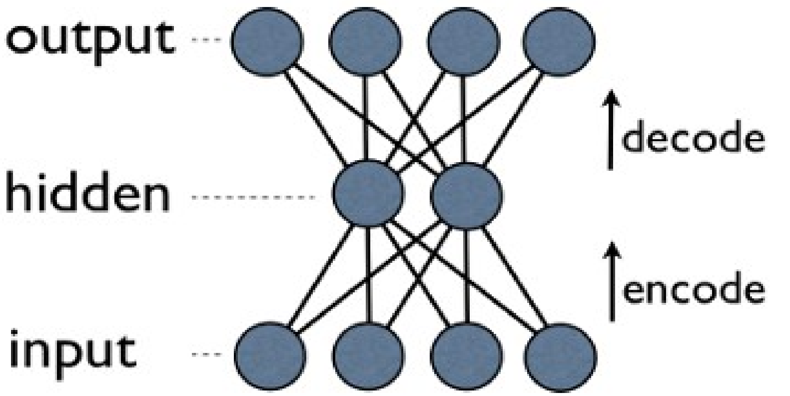
LAMBDA

British Hedgehog Preservation Society

Autoencoders 101

Main idea:

- Take data in some original (high-dimensional) space;
- Project data into a new space **from which it can then be accurately restored;**
- Encoder = data to hidden
- Decoder = hidden to data
- Decoder(Encoder(x)) ~ x



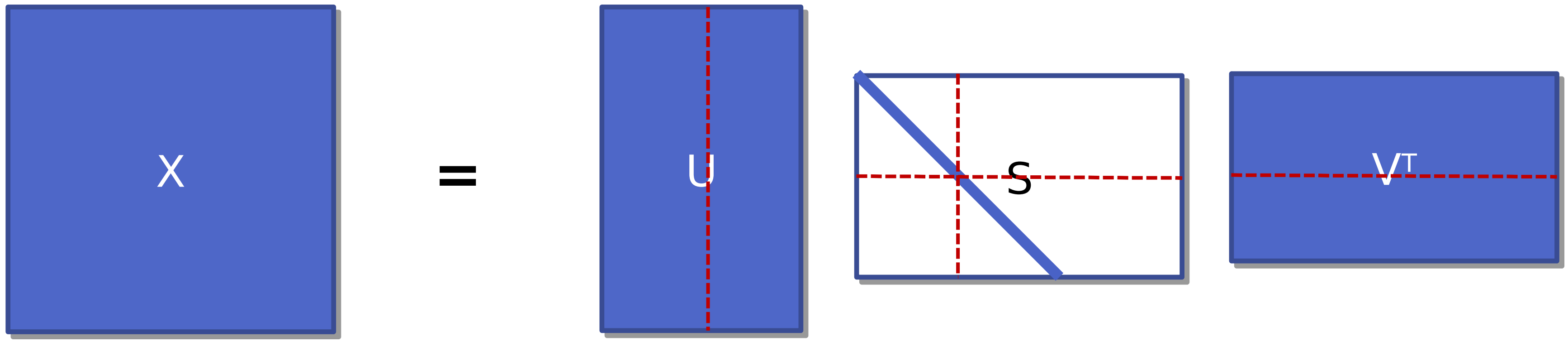
Why do we ever need that?

- Compress data
  - |code| << |data|
- Dimensionality reduction
  - Before feeding data to your XGBoost

<to be continued>

Matrix decompositions

- Example: SVD/PCA

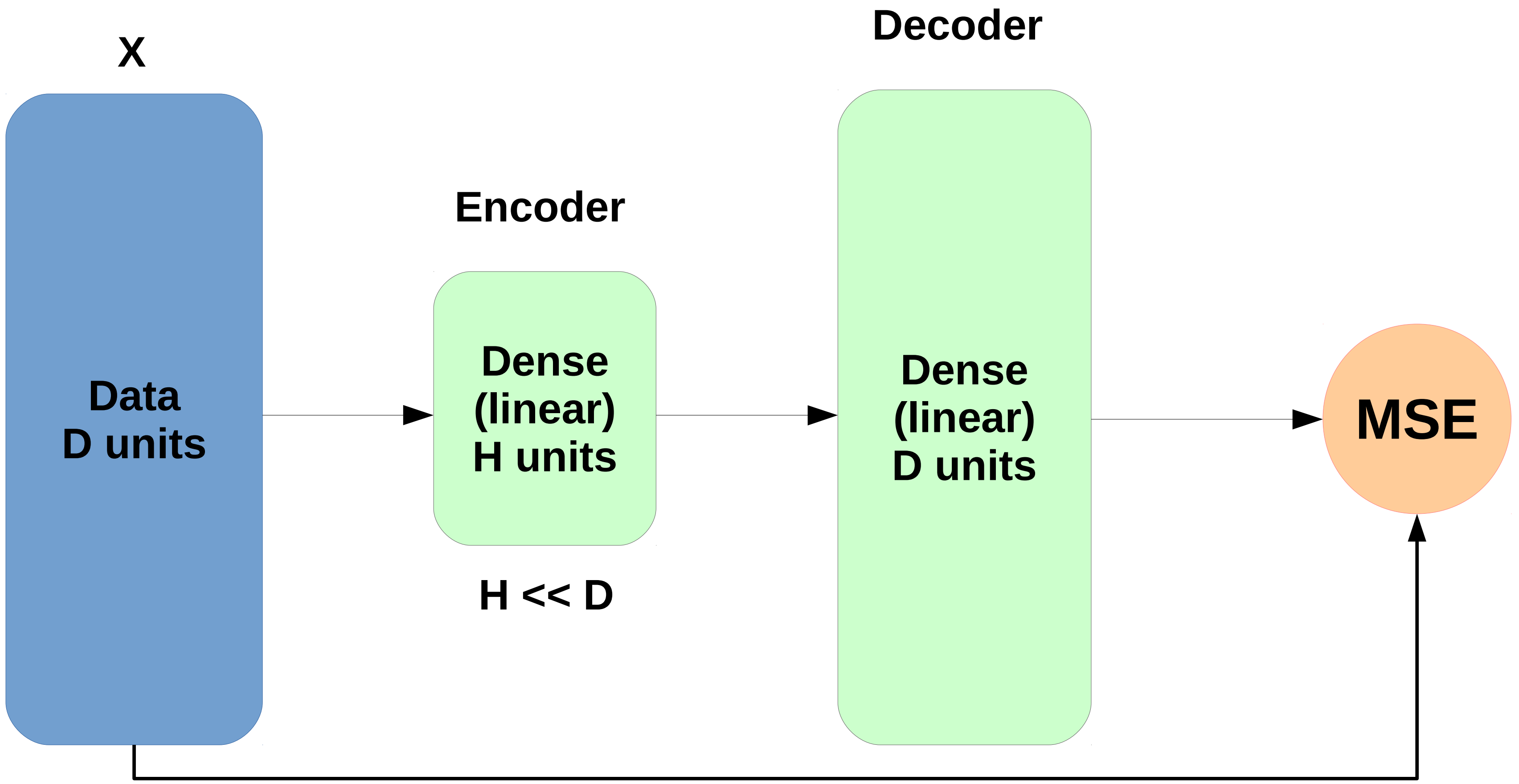


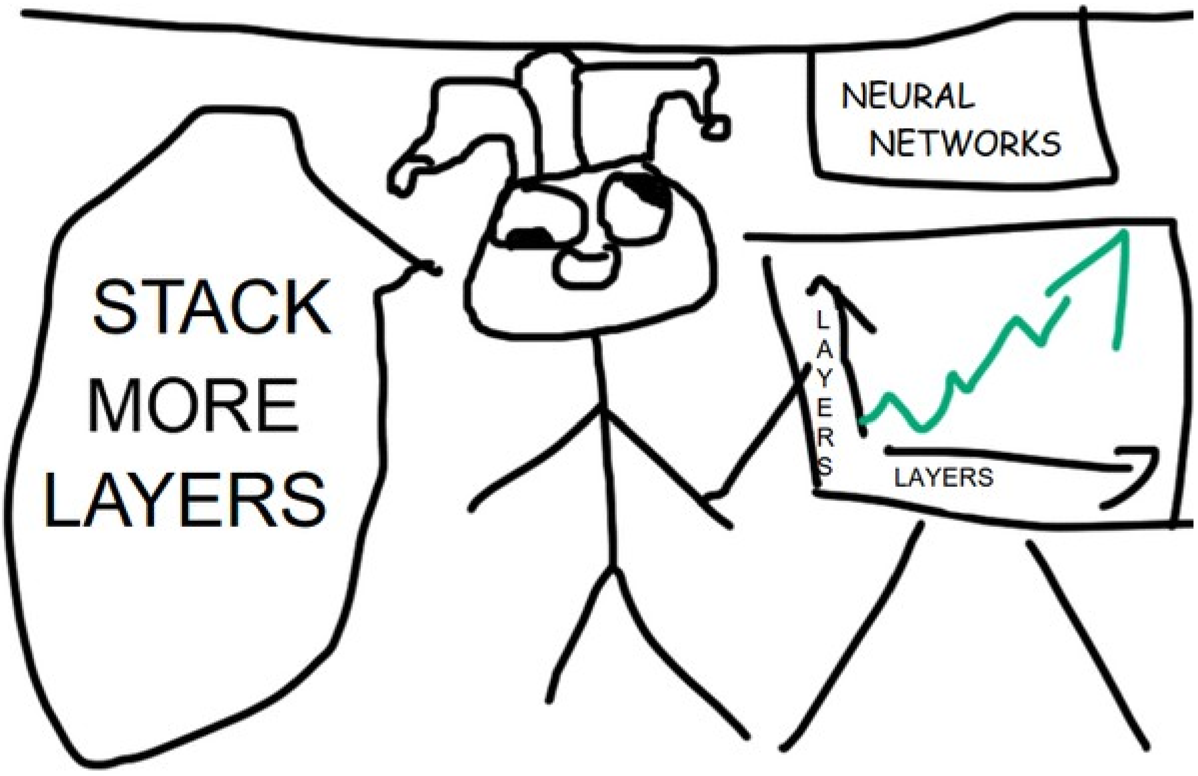
- Minimizing reconstruction error

$$L = \| X - U \cdot S \cdot V^T \|$$

Matrix decomposition

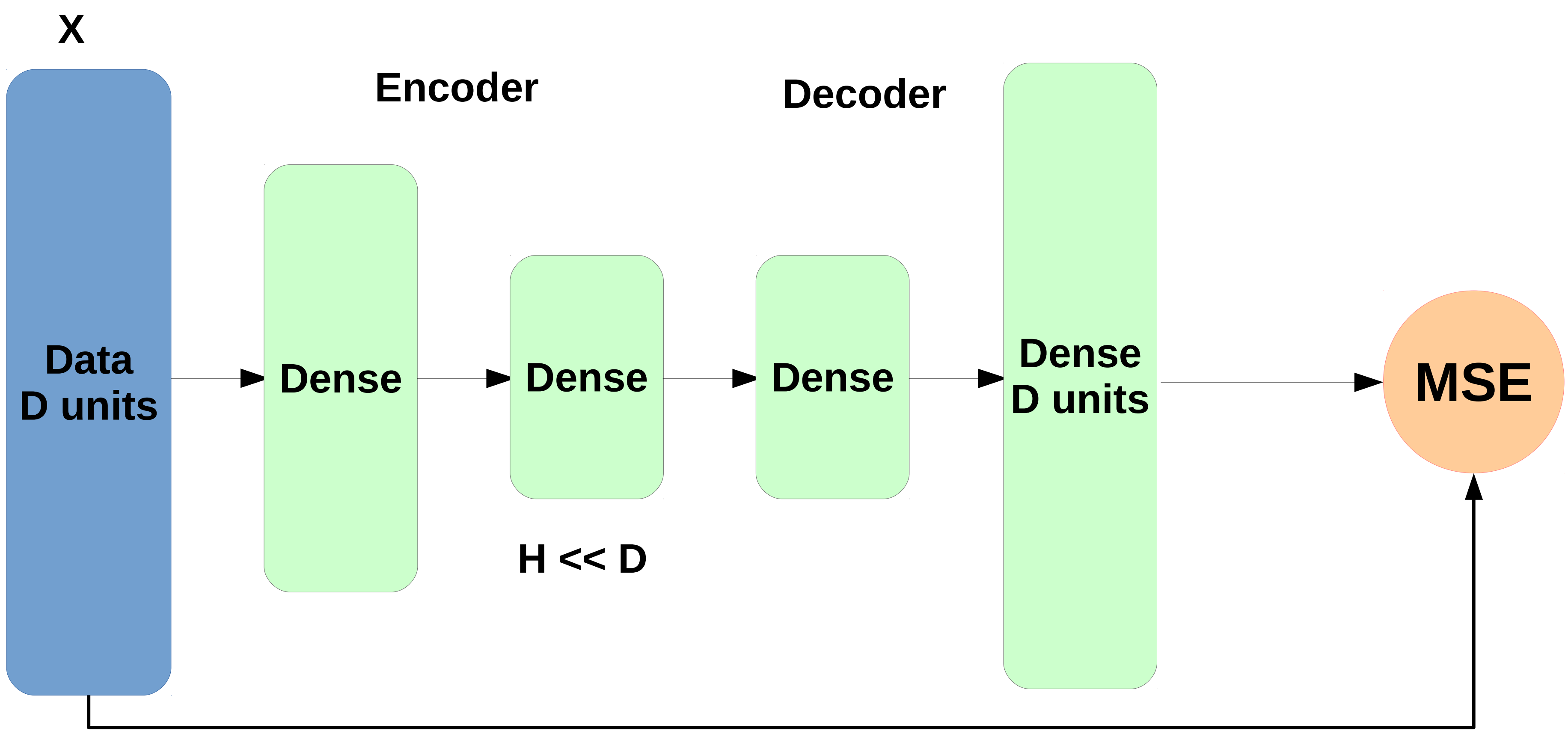
- A different perspective




STACK MORE LAYERS
NEURAL NETWORKS
LAYERS
LAYERS

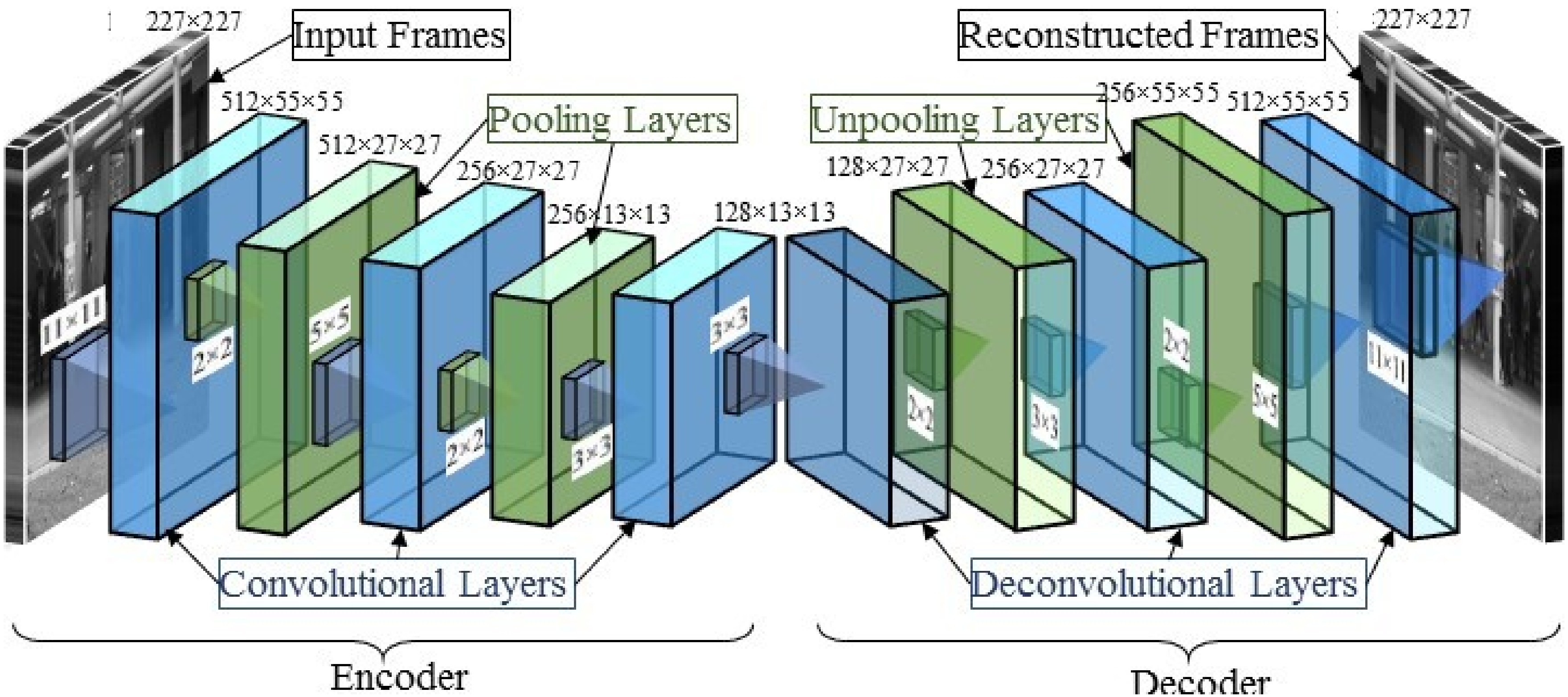
(kinda) Deep autoencoder

- Stack more layers!



Quiz: What if data is an image?

Image2image: fully-convolutional



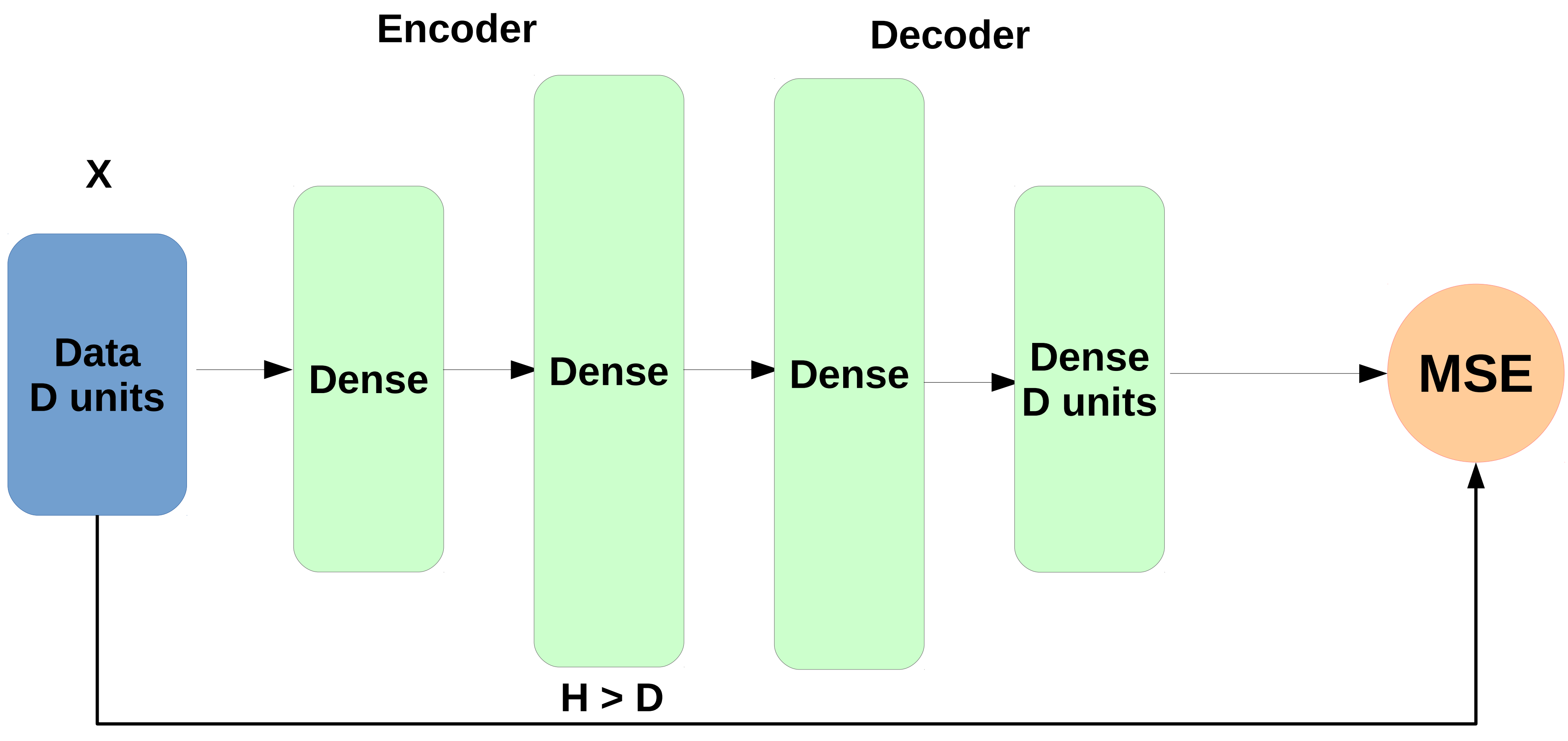
Quiz: what is the compression rate here?

Why do we ever need that?

- Compress data
  - |code| << |data|
- Dimensionality reduction
  - Before feeding data to your XGBoost
- **Learn some great features!**
  - Before feeding data to your XGBoost

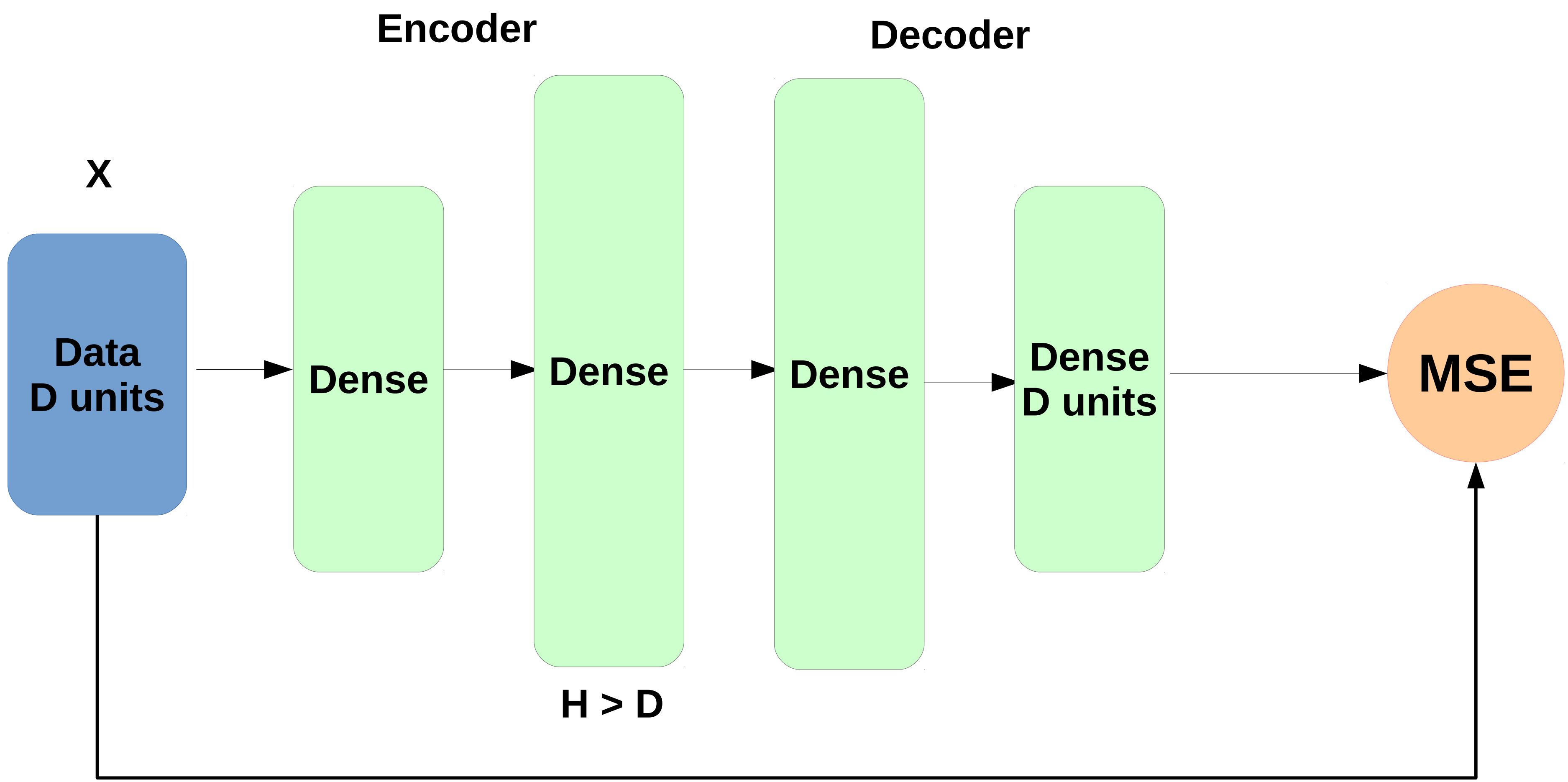
Expanding autoencoder

- Bigger/richer representation



Expanding autoencoder

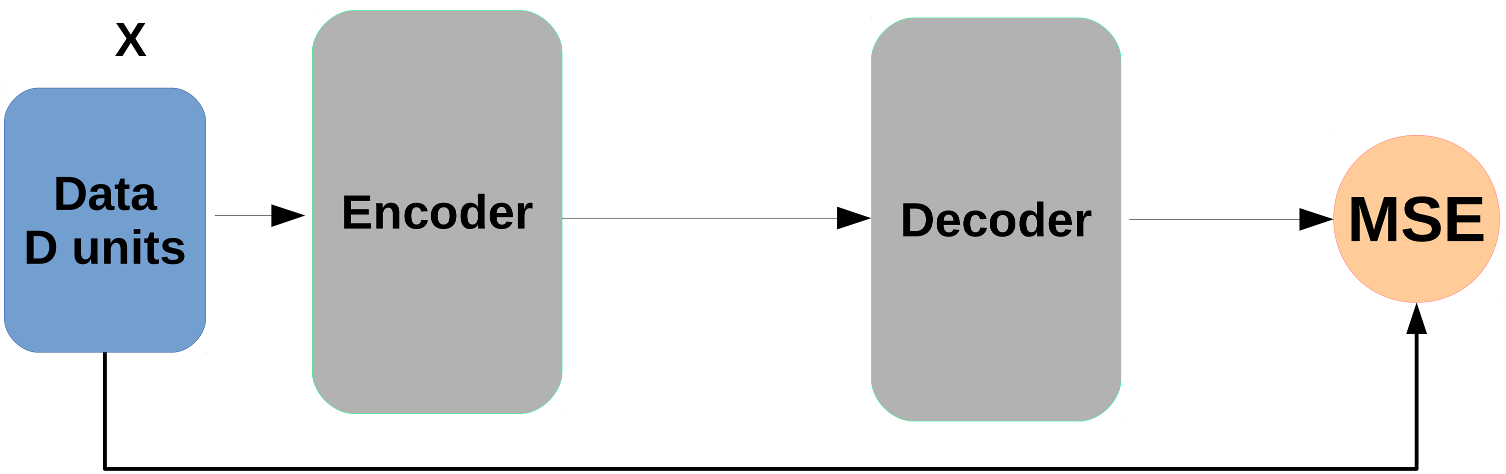
- Bigger/richer representation



Something's wrong with this guy. **Ideas?**

Expanding autoencoder

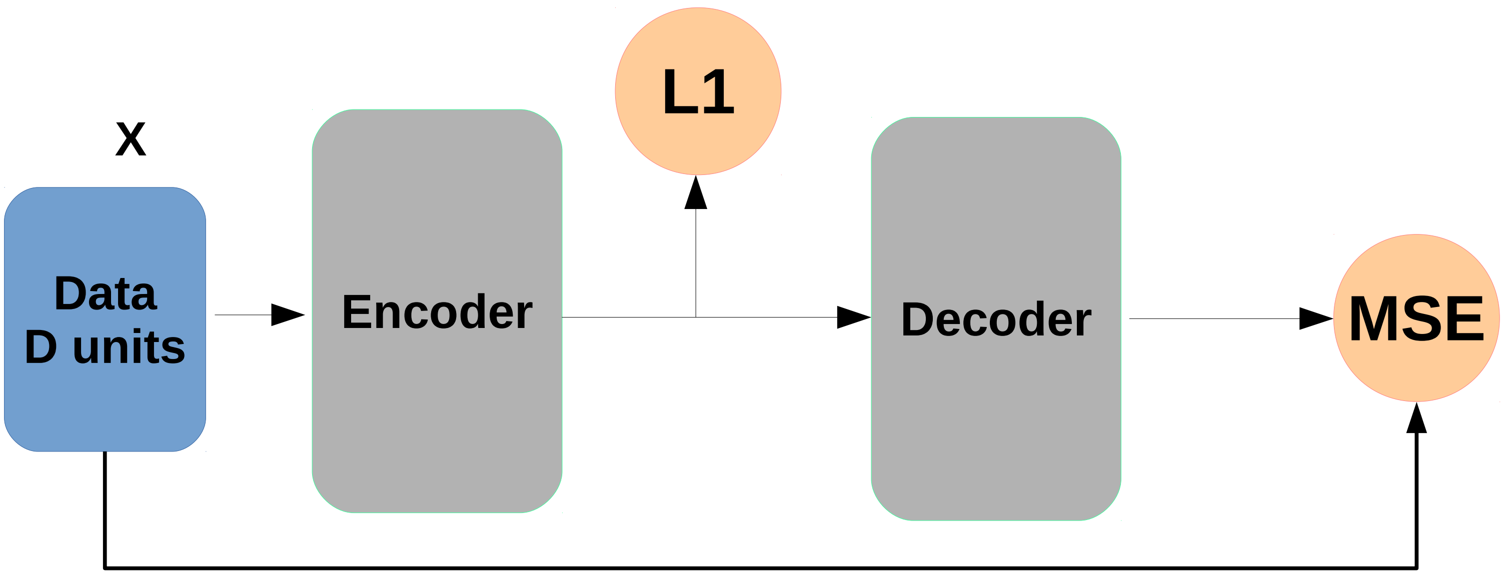
- Naive approach will learn identity function!
- Gotta regularize!



$$L = \| X - Dec(Enc(X)) \|$$

Sparse autoencoder

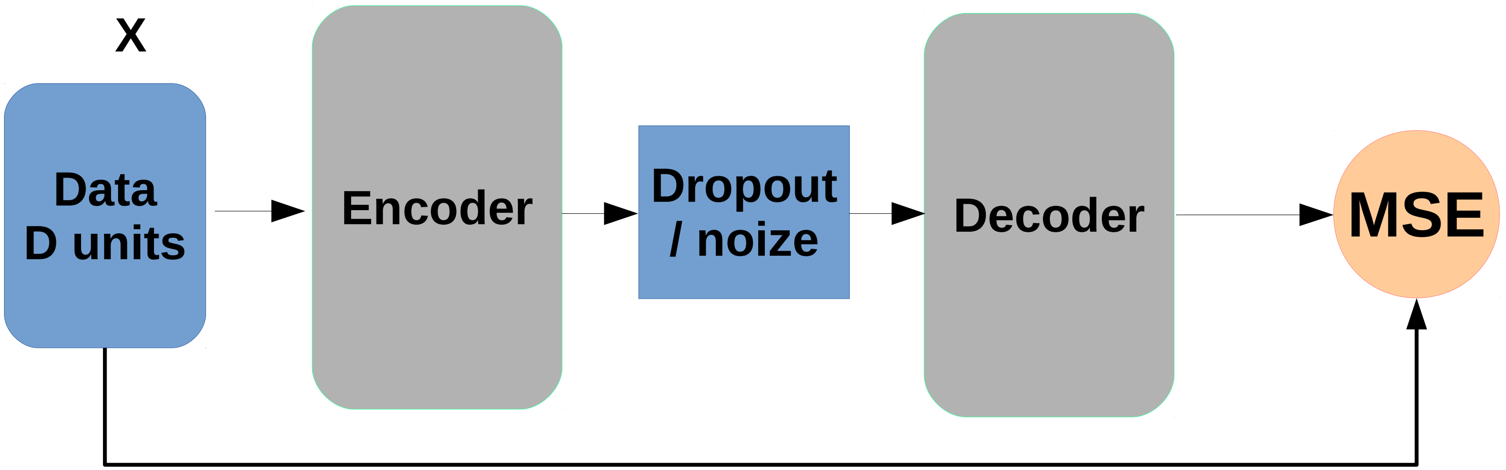
- Naive approach will learn identity function!
- Idea 1: L1 on **activations**, sparse code



$$L = \|X - Dec(Enc(X))\| + \sum_i |Enc_i(X)|$$

Redundant autoencoder

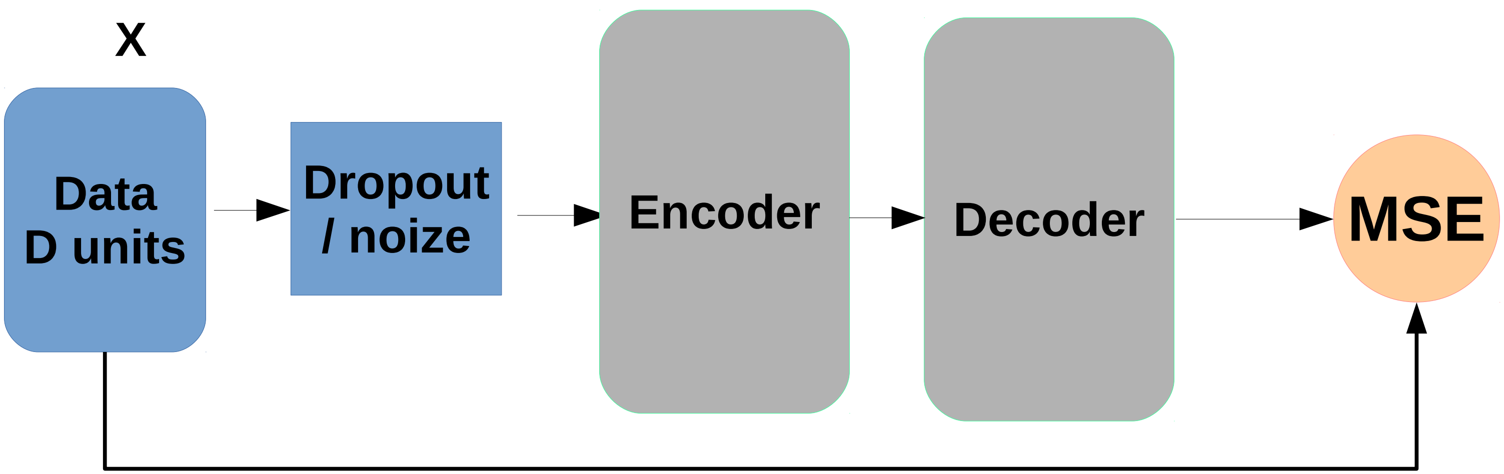
- Naive approach will learn identity function!
- Idea 2: noize/dropout, redundant code



$$L = \| X - Enc(Noize(Dec(X))) \|$$

Denoizing autoencoder

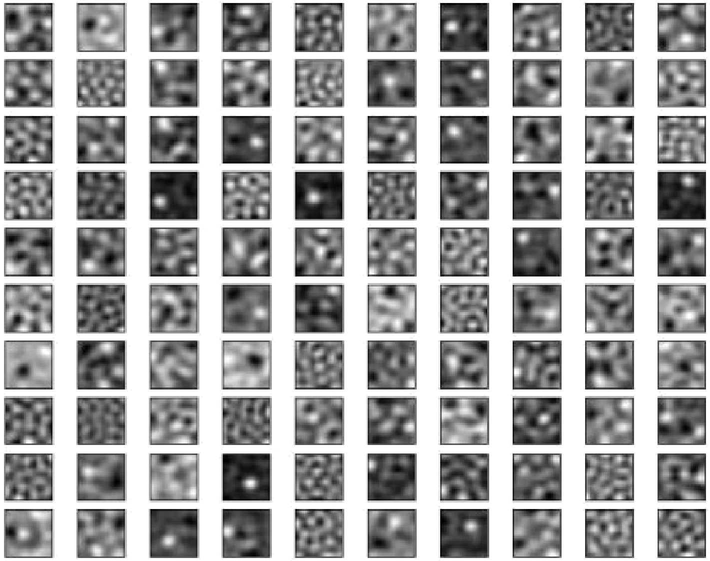
- Naive approach will learn identity function!
- Idea 3: distort input, learn to undo distorsion



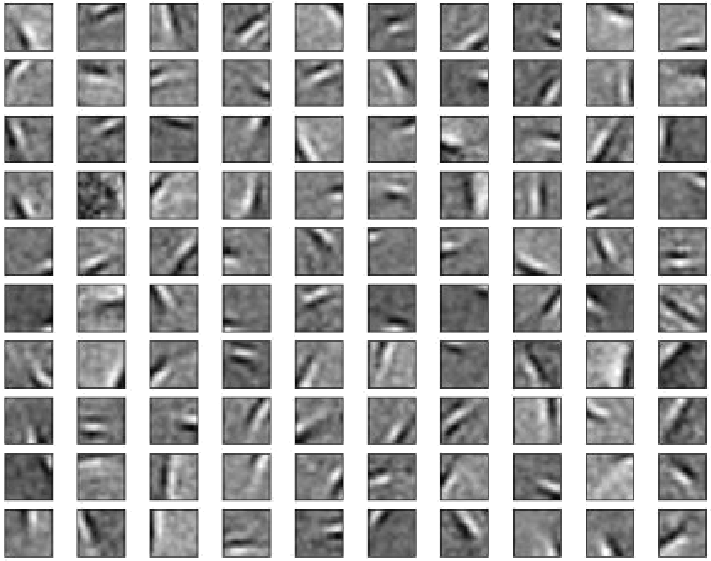
$$L = \| X - Enc(Dec(Noize(X))) \|$$

Sparse Vs Denoizing

- Filter weights, 12x12 patches



Sparse AE



Denoizing AE

Actually meaningless :)

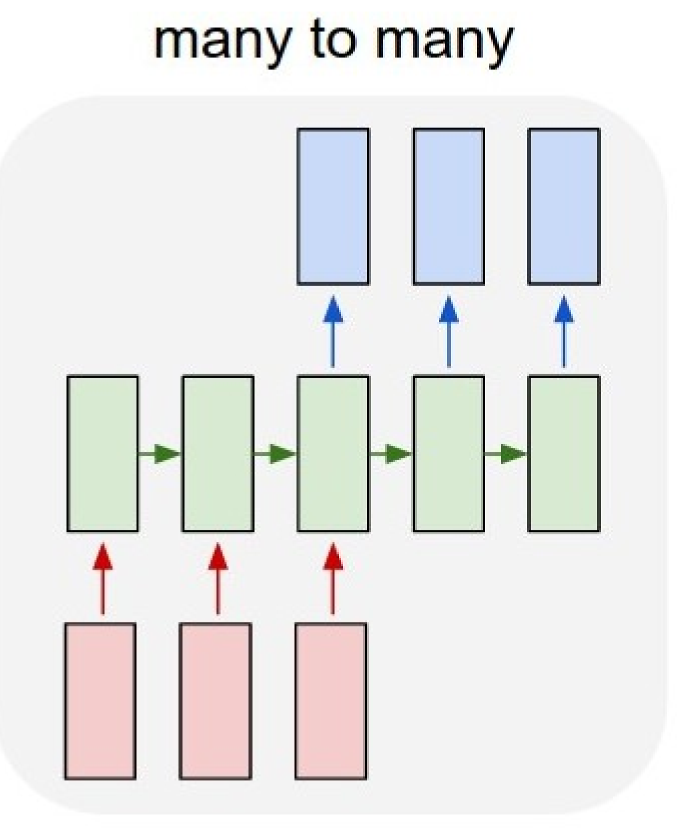
[Vincent et al. 2010]

Why do we ever need that?

- Compress data
  - |code| << |data|
- Dimensionality reduction
  - Before feeding data to your XGBoost
- Learn some great features!
- **Unsupervised pretraining**
  - Large amounts of data
  - Features may be irrelevant

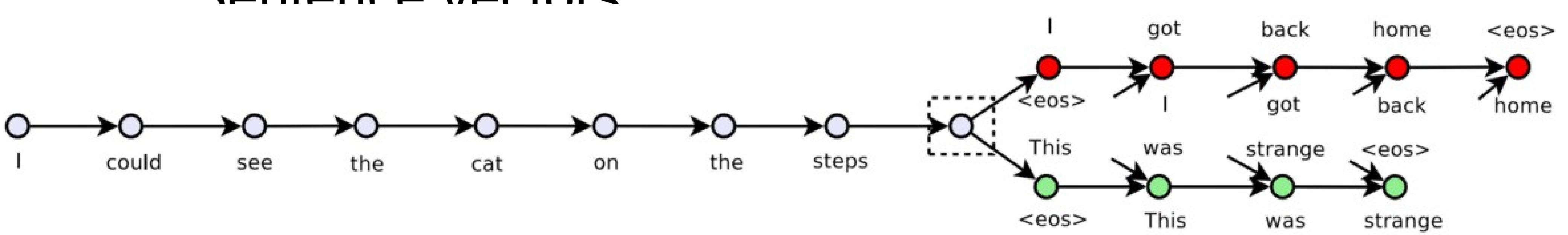
Recurrent autoencoders

- Regular encoder-decoder
- Where is the bottleneck?
- How do we train it?



Skip-thought

- Word2vec **skip-gram**:
  - Word → neighboring words
  - Embedding + Dense
  - Word vectors
- Phrase2vec **skip-thought**:
  - Sentence → prev/next sentence
  - Encoder-decoder
  - Sentence vectors



Why do we ever need that?

- Compress data
  - |code| << |data|
- Dimensionality reduction
  - Before feeding data to your XGBoost
- Learn some great features!
- Unsupervised pretraining
  - Large amounts of data
  - Features may be irrelevant
- **Generating new images!**

Image morphing with AE

Idea:

- If Enc(image1) = c1

  Enc(image2) = c2

- Than maybe (c1+c2)/2 is a semantic average of the two images

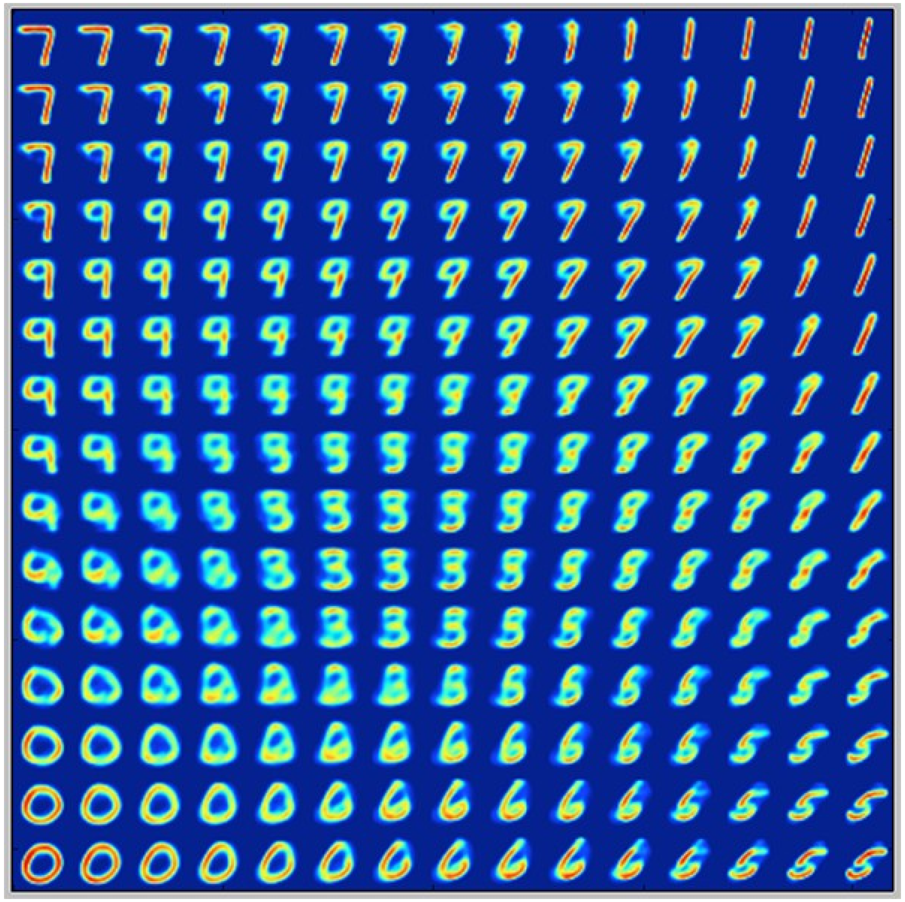


Image morphing with AE

Idea:

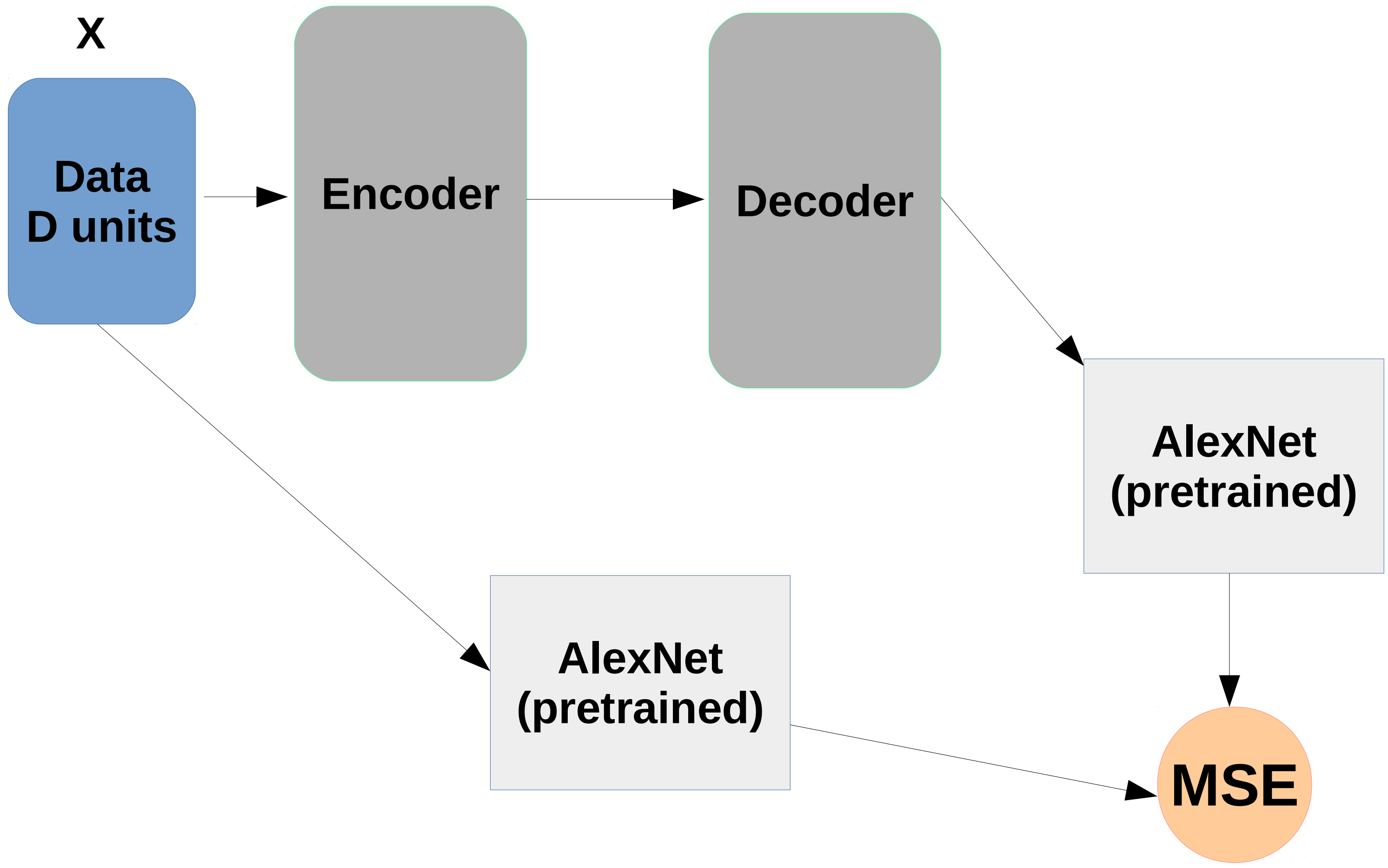
- Look for a common direction vector for "add mustache" or "add age" changes.
- Apply to new images

Mean Squared Error

Pixelwise MSE:

- A **“cat on the left”** is closer to **“dog on the left”** than to **“cat on the right”**
- We may want to avoid that effect
- Can we obtain image representation that is less sensitive to small shifts?

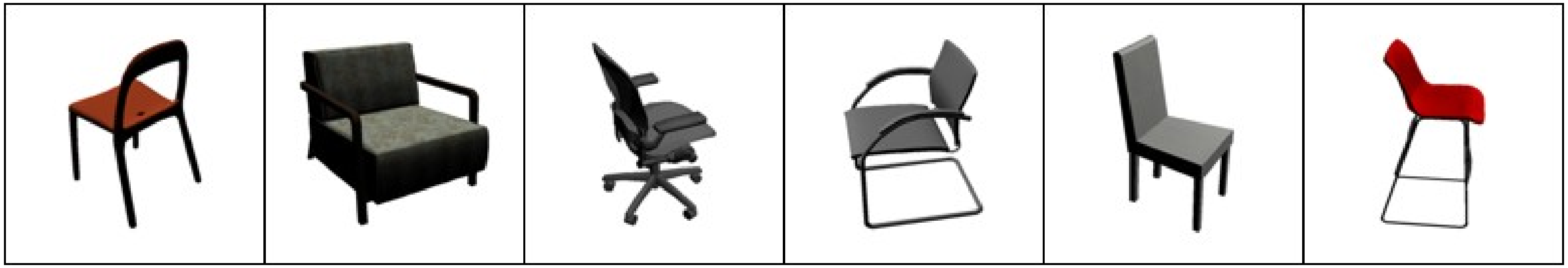
Sketch: using pre-trained nets



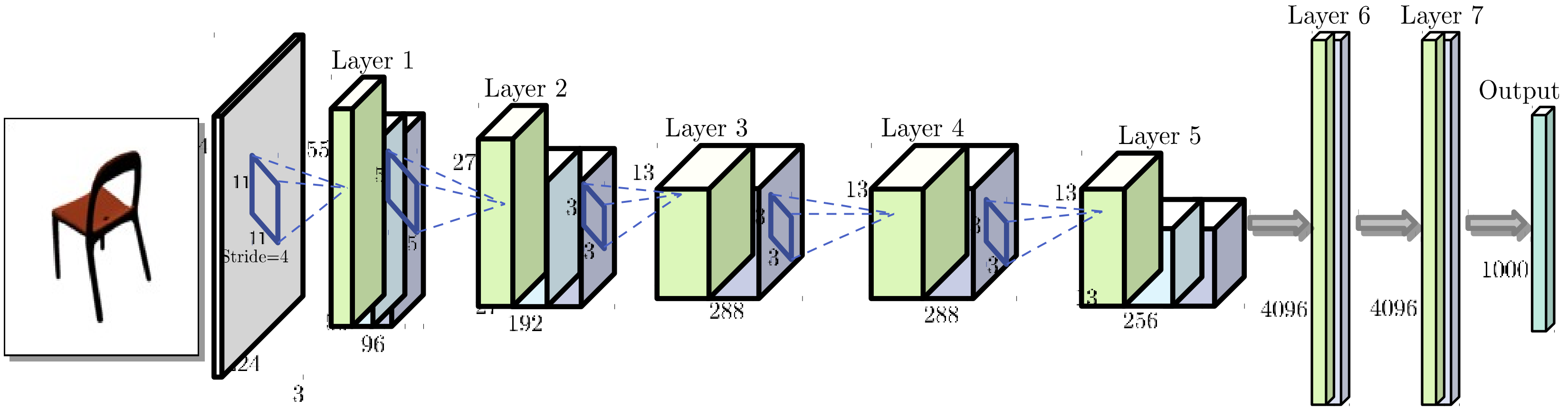
$$L = \| f(X) - f(Dec(Enc(X))) \|$$

Image generation

- Chairs (type, view, orientation)



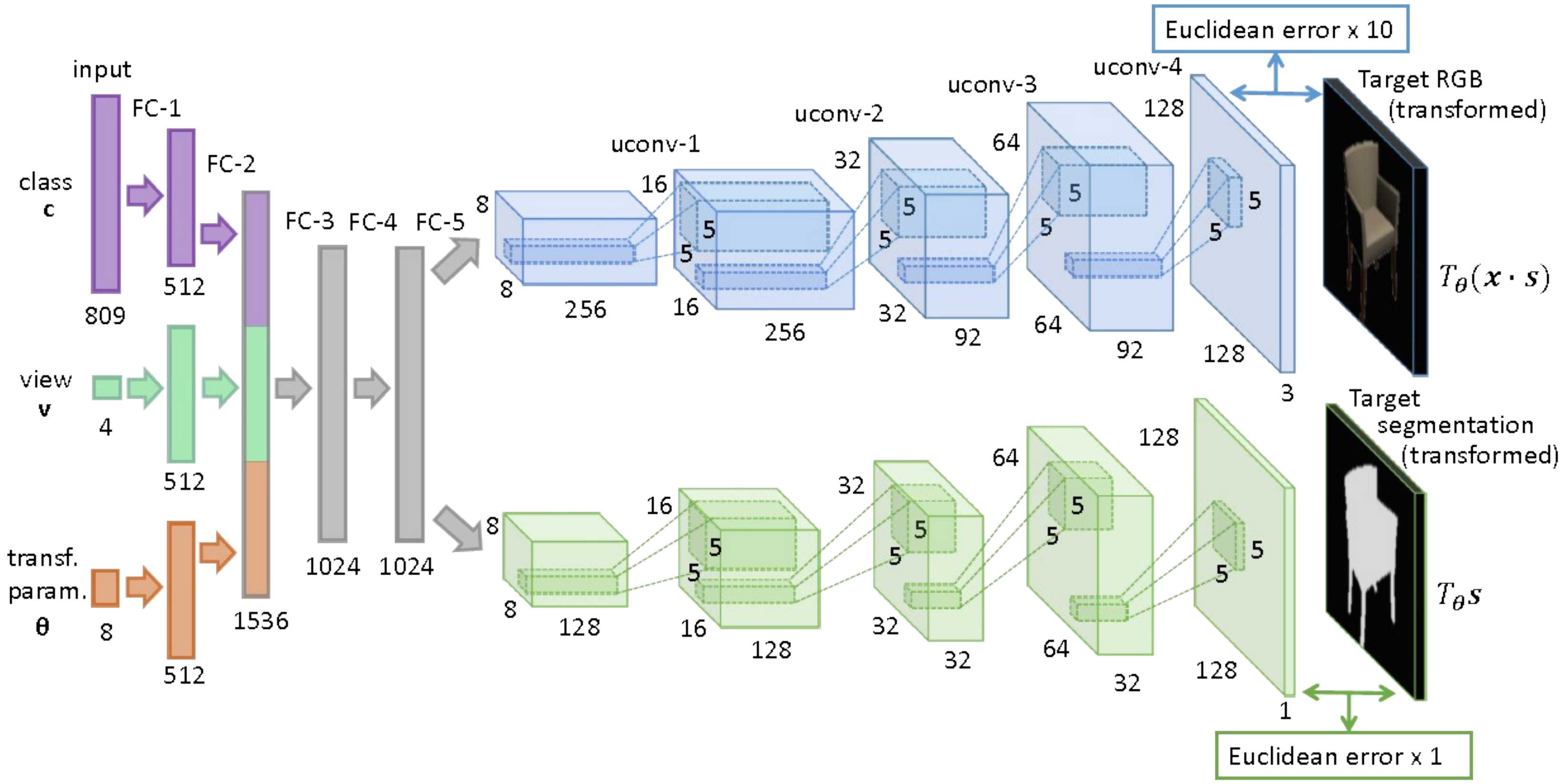
- Classifier



P(y|x)

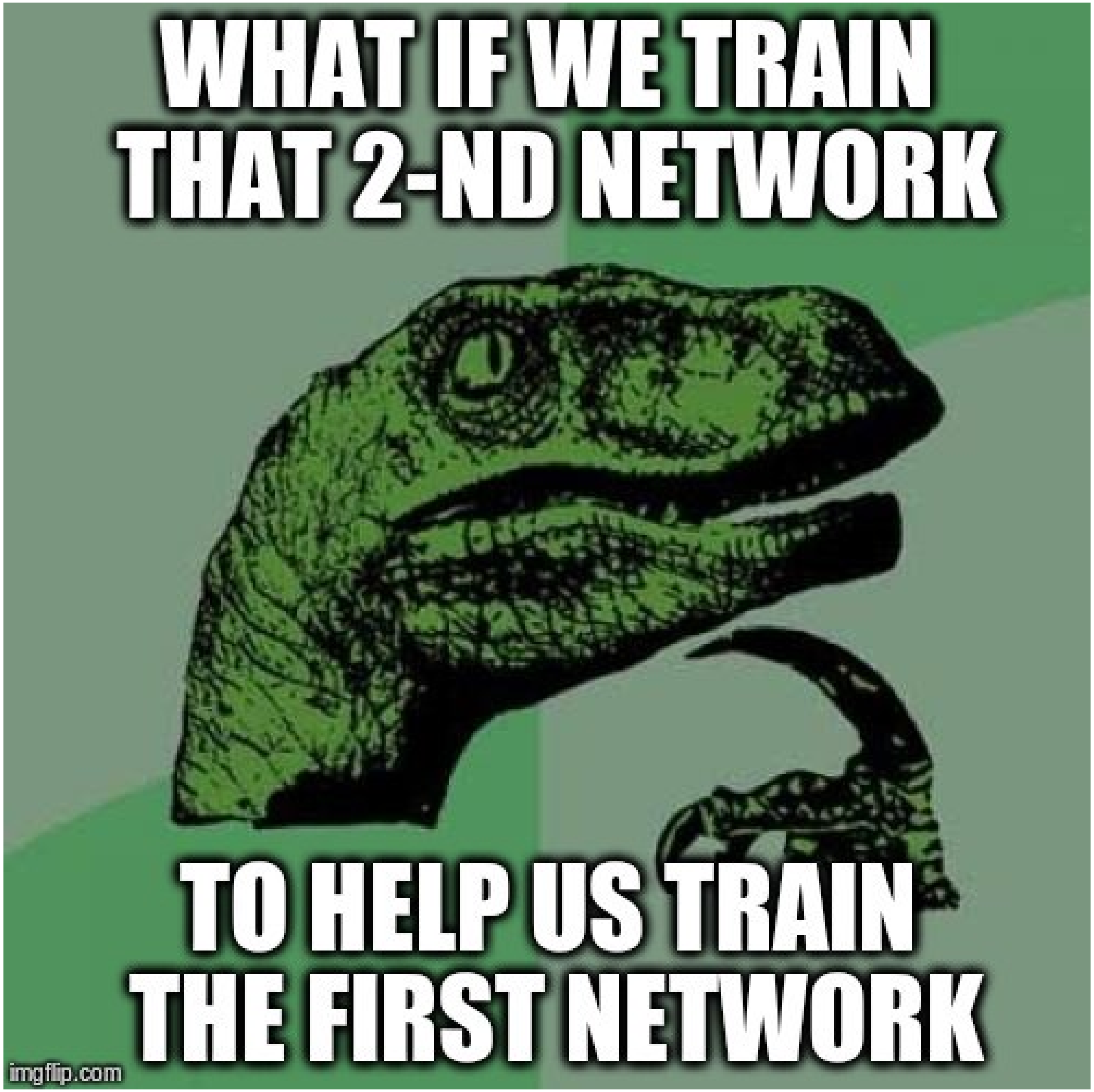
Image generation

- Generator



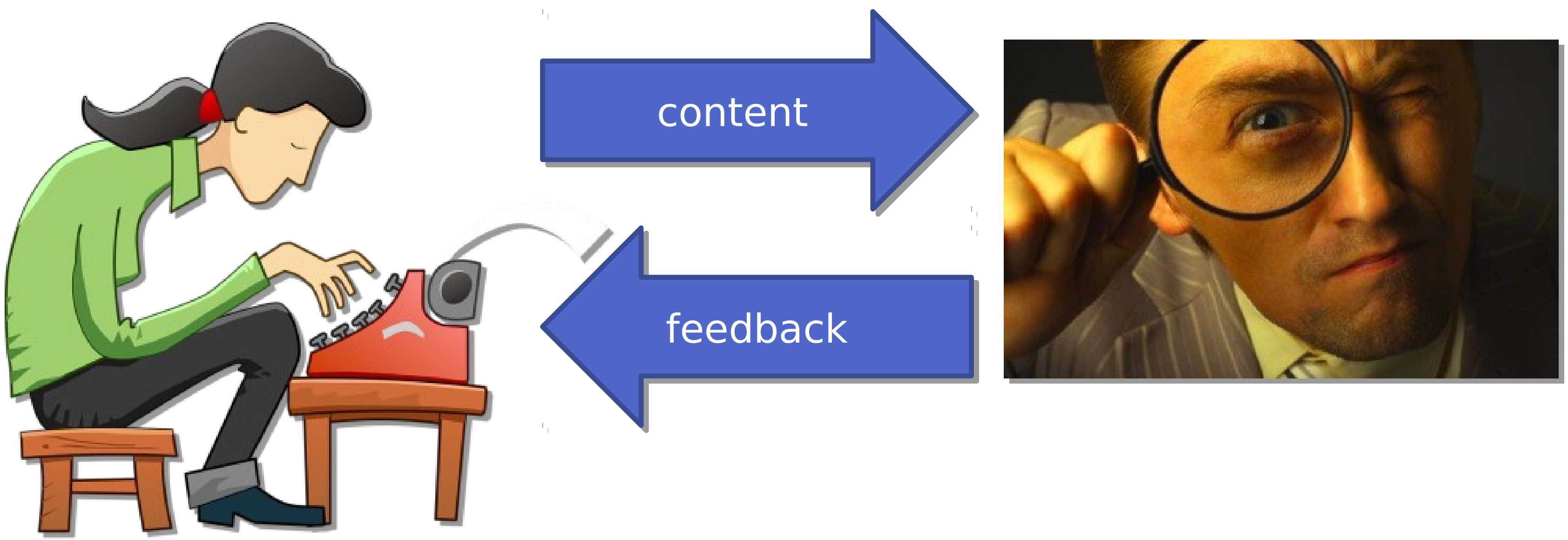
Problem: MSE sucks at this task.

Ideas?


WHAT IF WE TRAIN
THAT 2-ND NETWORK
TO HELP US TRAIN
THE FIRST NETWORK
imgflip.com

Generative Adversatial Networks

Generator | **Discriminator**

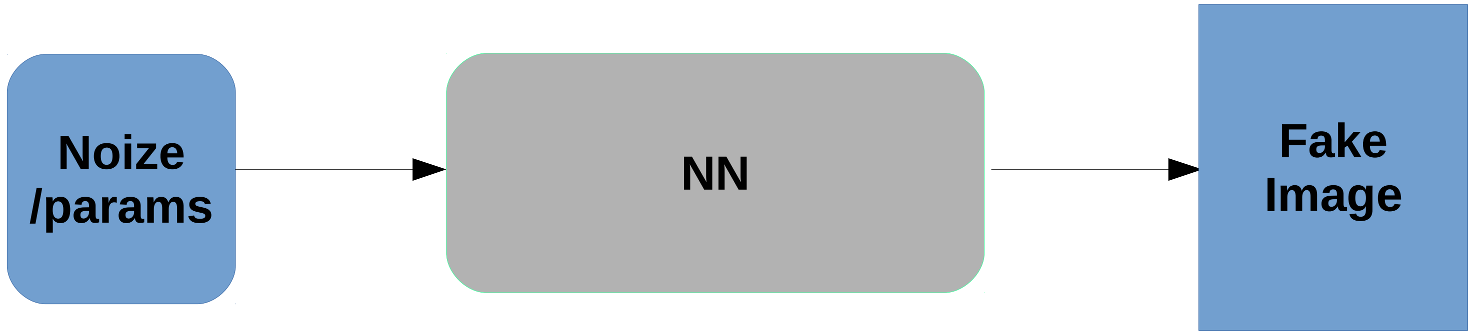


Generate image
(should be plausible)

Tell if image is plausible
(image) → P(fake)

Generative Adversarial Networks

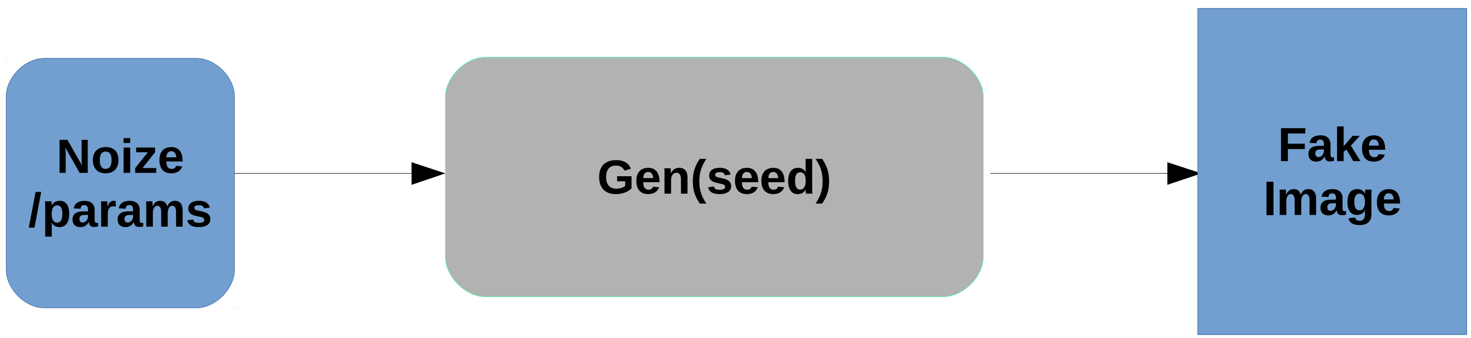
- Generator



- Discriminator

Generative Adversarial Networks

- Generator

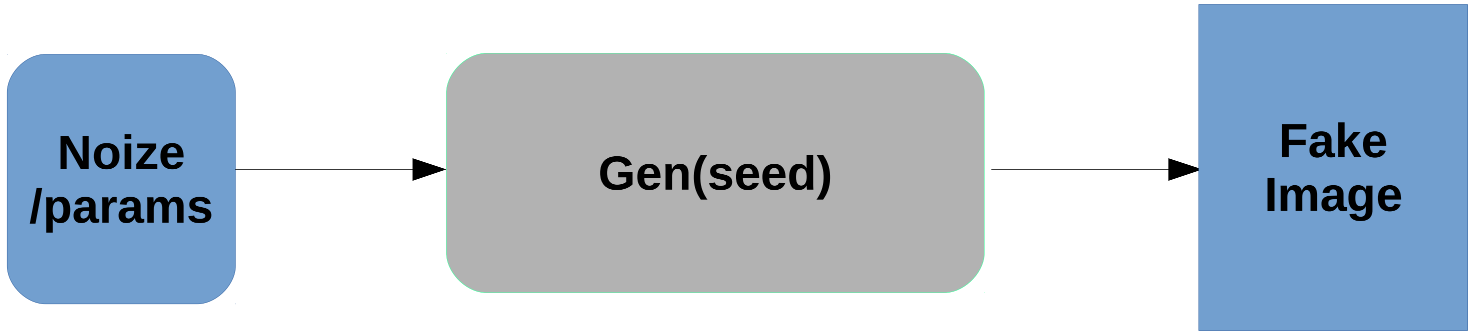


- Discriminator

$$L_D = -\log\left[1 - Disc\left(real\ data\right)\right] - \log Disc\left(Gen\left(seed\right)\right)$$

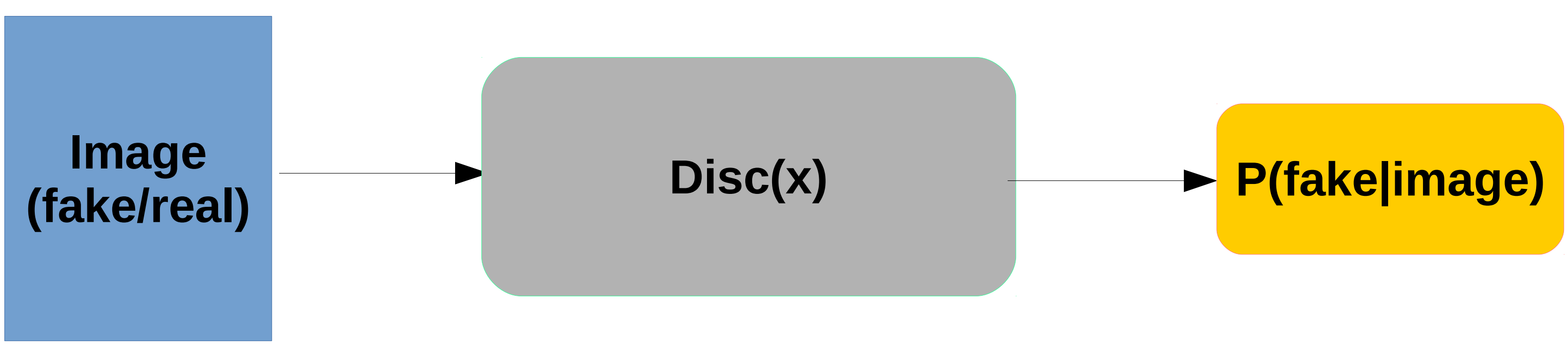
Generative Adversarial Networks

- Generator $L_G = -\log\left[1 - Disc\left(Gen\left(seed\right)\right)\right]$



- Discriminator

$$L_D = -\log\left[1 - Disc\left(real\ data\right)\right] - \log Disc\left(Gen\left(seed\right)\right)$$



Generative Adversarial Networks

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

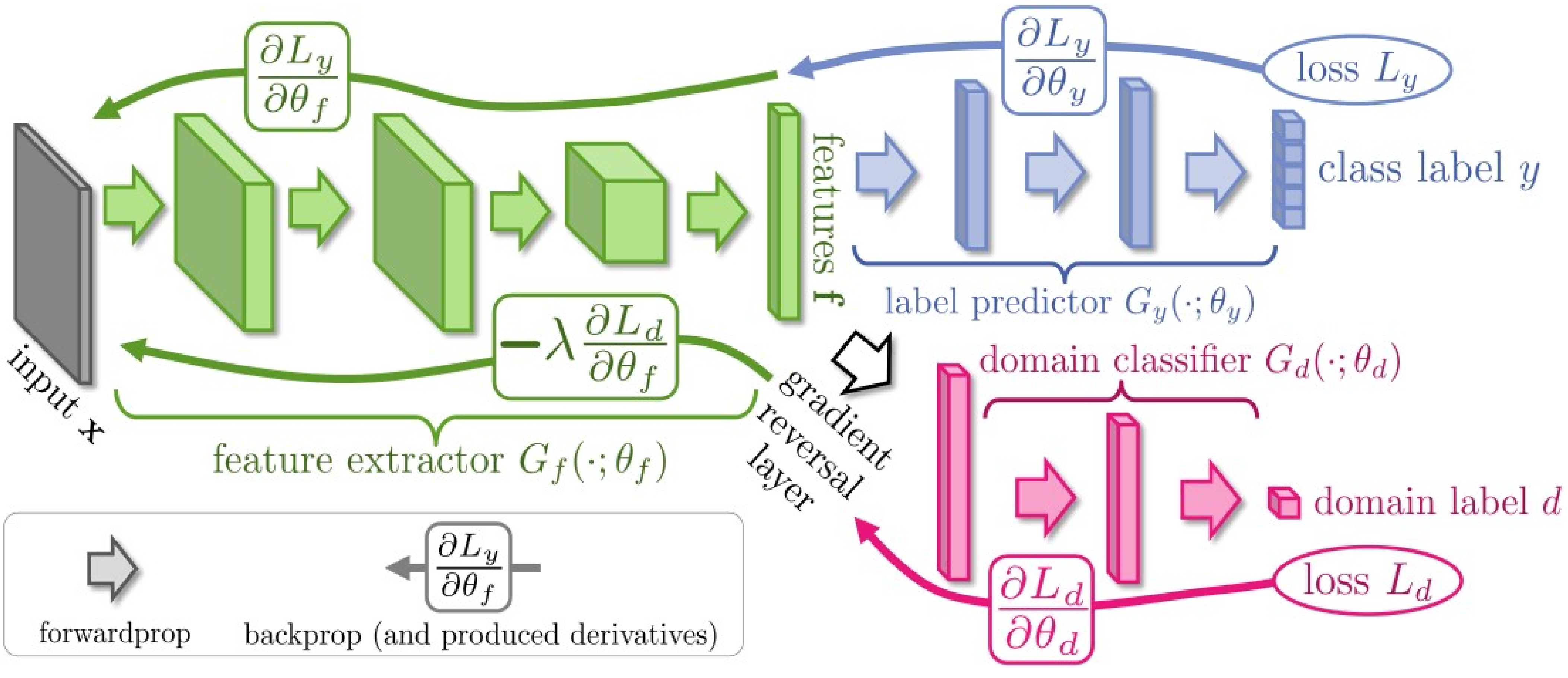
end for

Adversarial domain adaptation

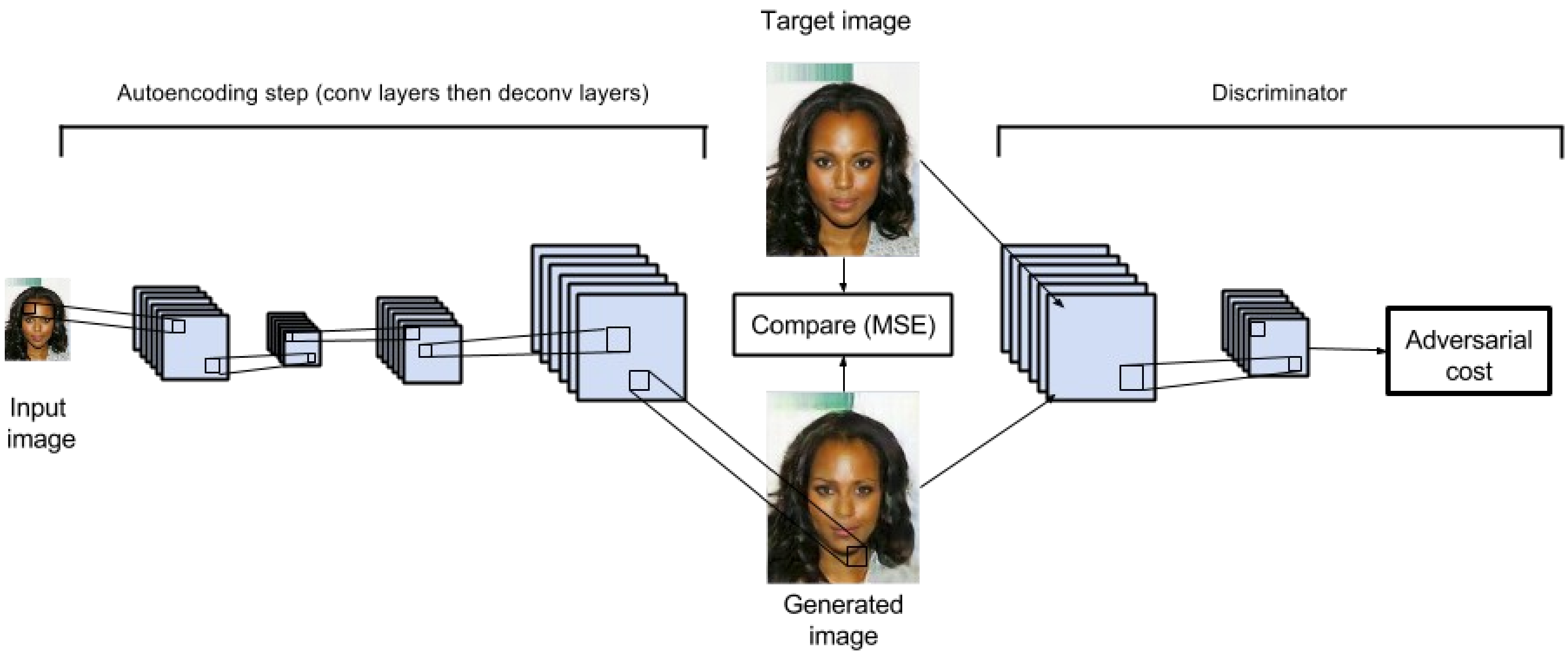
- Two domains
  - e.g. mnist digits Vs actual digits on photos
- First domain is labeled, second is not
- Wanna learn for the second domain

Domain adaptation

- Idea: discriminator should not be able to distinguish features on two domains



Adversarial autoencoders



Brace yourselves

Art style transfer

- Ideas?