ML @ ICL
Episode -1

Generative & Unsupervised deep learning

Yandex Data Factory

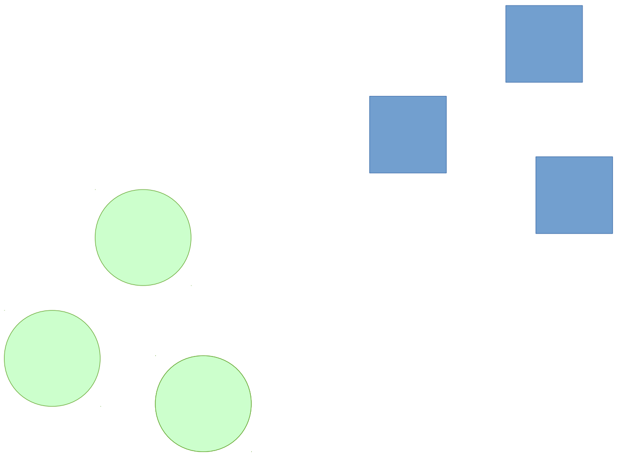
LAMBDA

British Hedgehog Preservation Society

Supervised vs Unsupervised

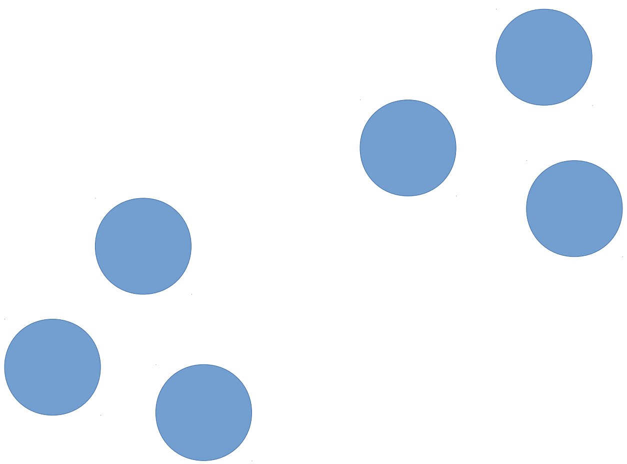
Supervised learning

- Take (x,y) pairs



Unsupervised learning

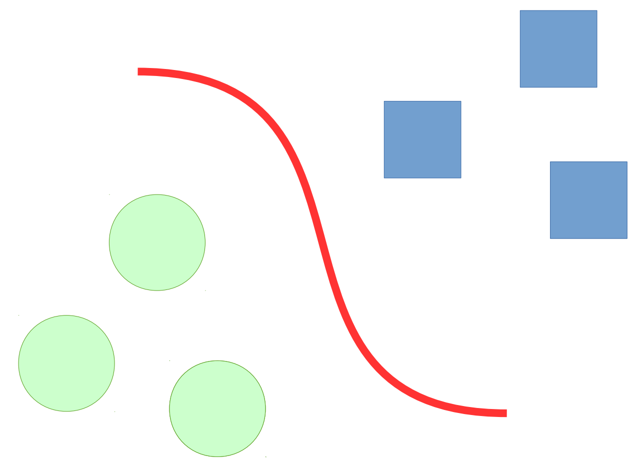
- Take x alone



Supervised vs Unsupervised

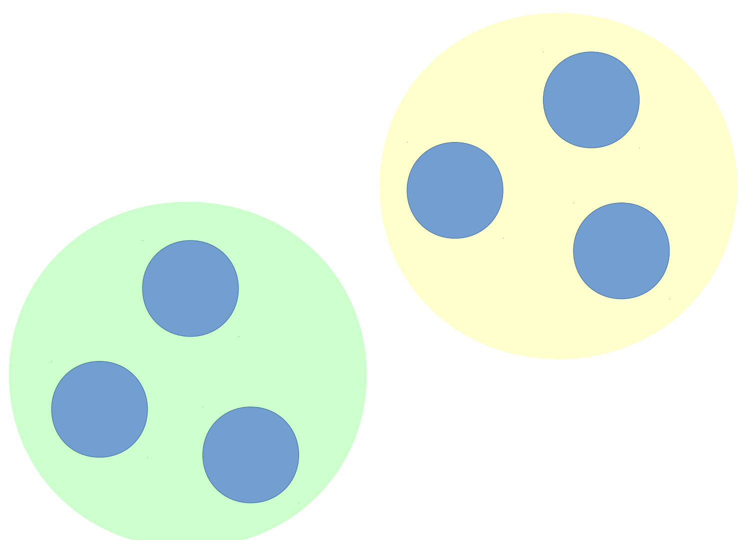
Supervised learning

- Take (x,y) pairs
- Learn mapping x → y



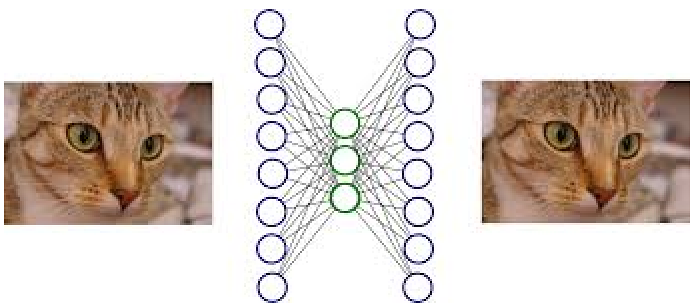
Unsupervised learning

- Take unlabeled x
- Learn hidden structure
  behind the data



Why bother?

- Find most relevant features



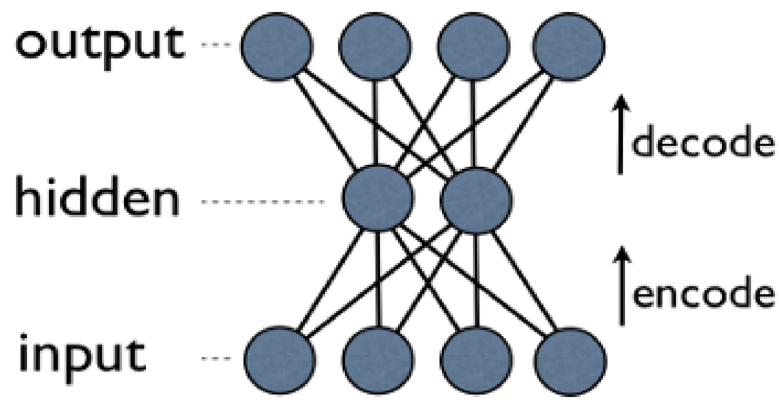
- Compress information
- Visualize high-dimensional data

- Retrieve similar objects
- Generate new data samples

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



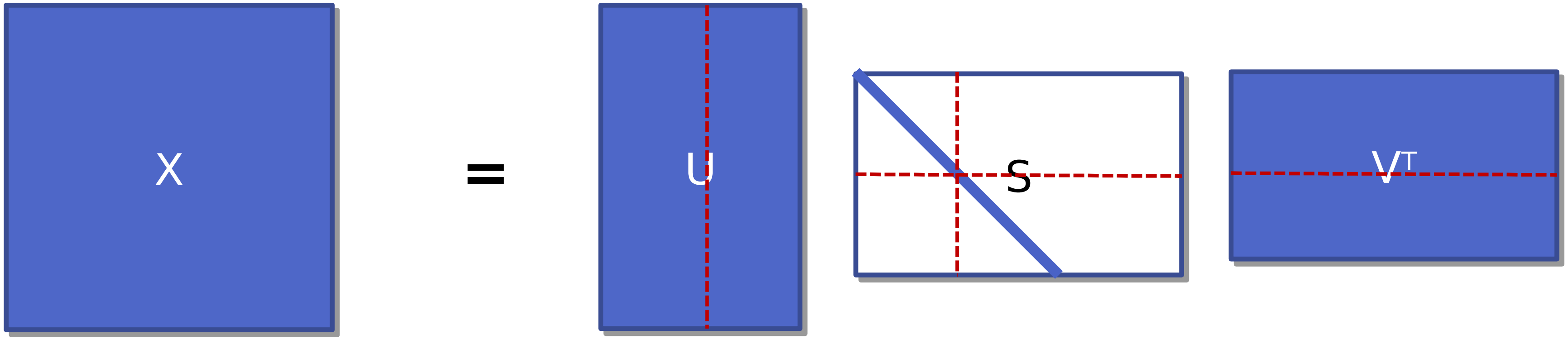
Why do we ever need that?

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

<to be continued>

Matrix decompositions

- Example: SVD/PCA

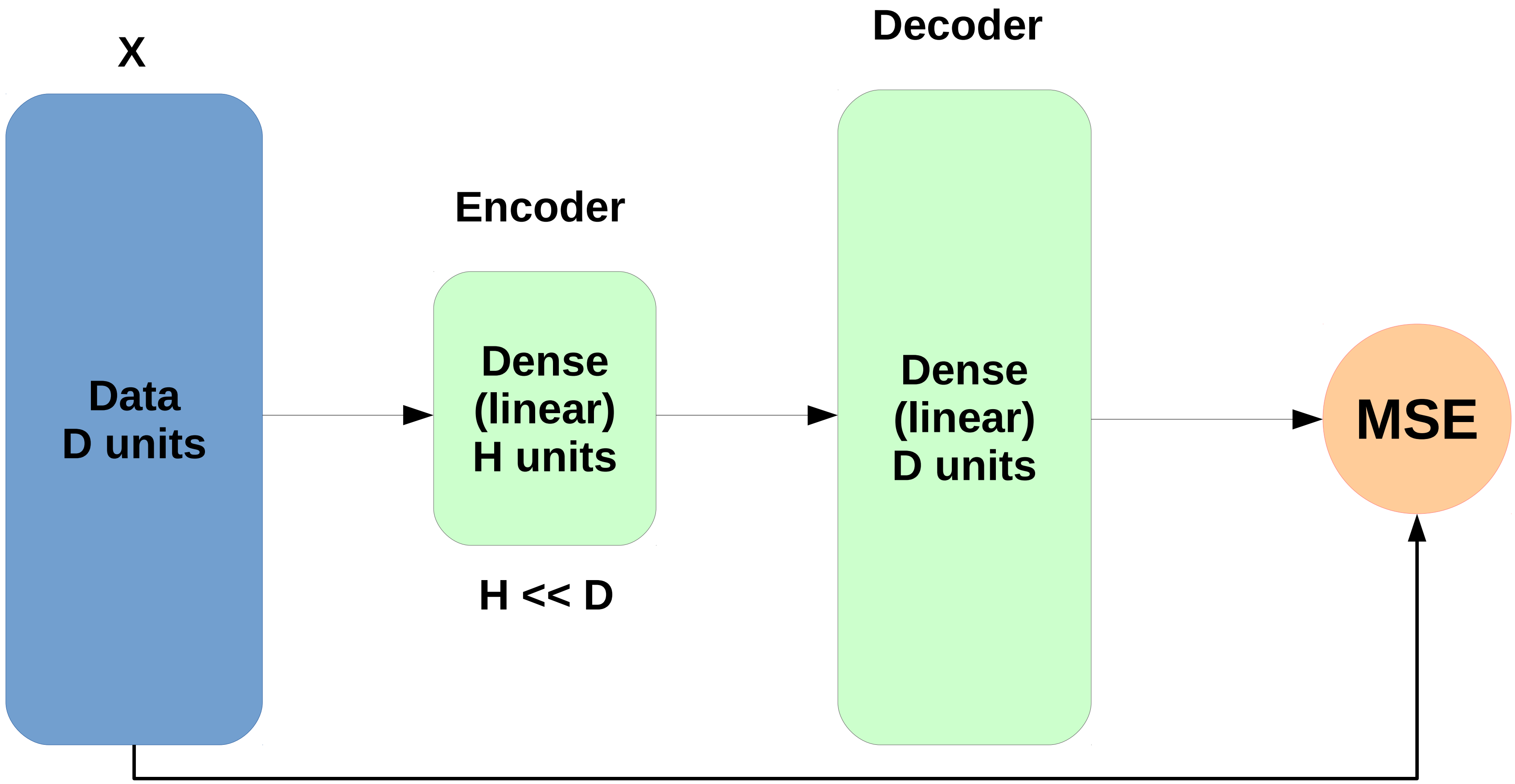


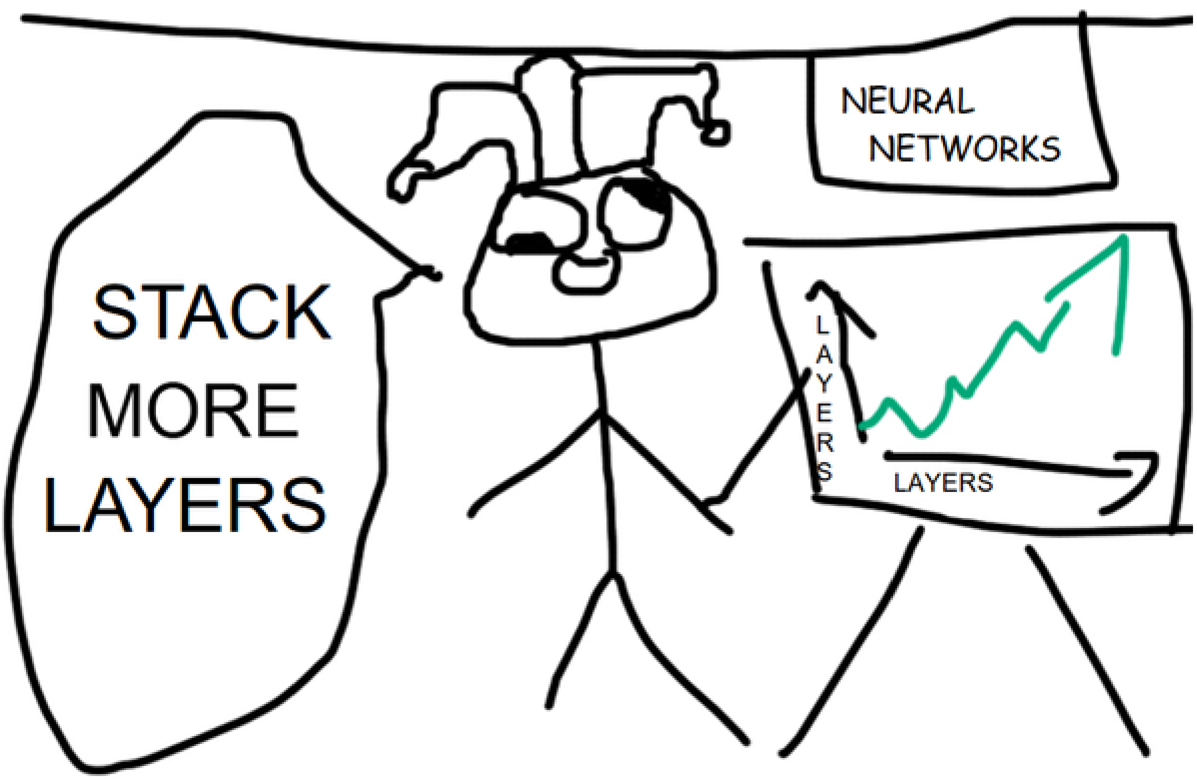
- Minimizing reconstruction error

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

Matrix decomposition

- A different perspective

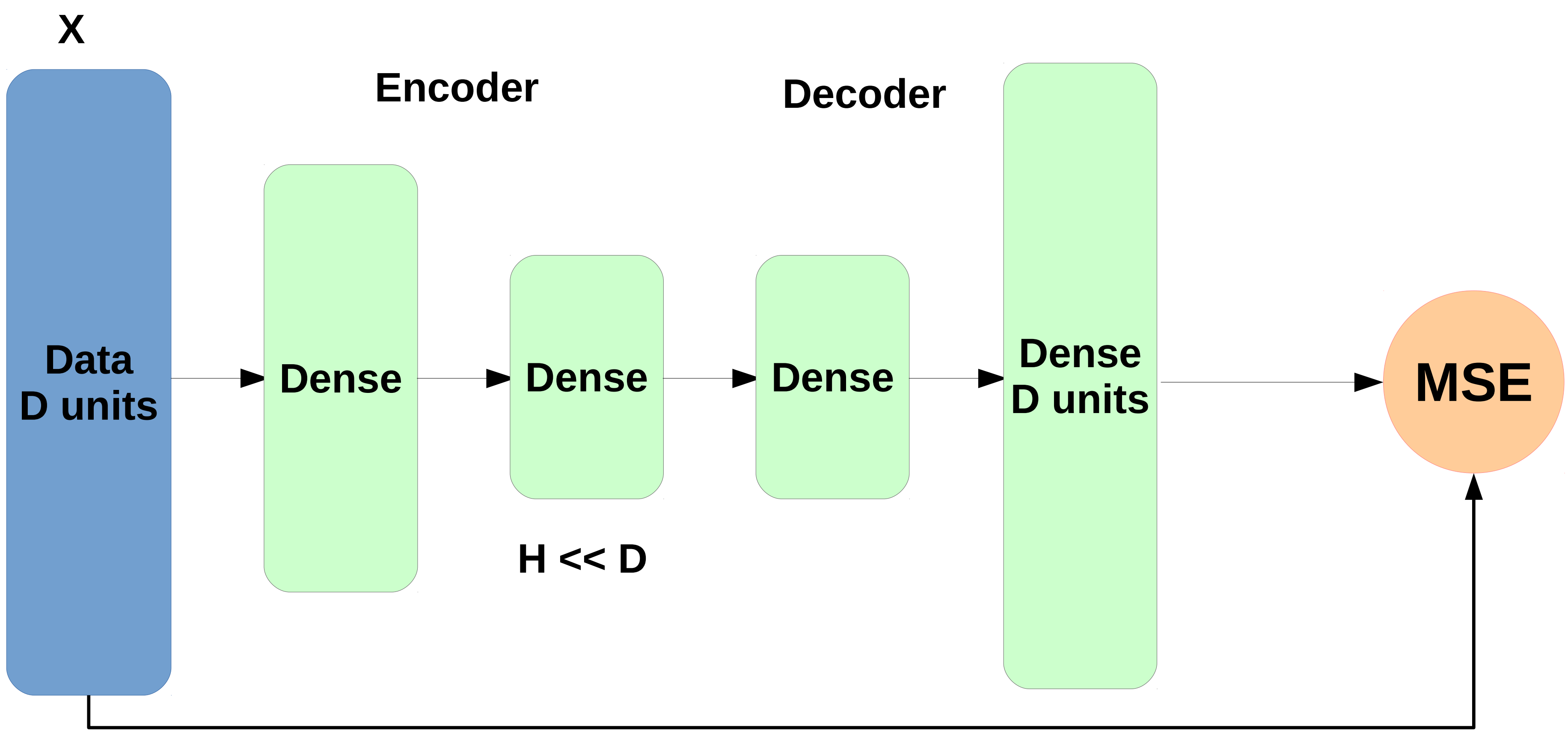




NEURAL NETWORKS
STACK MORE LAYERS
LAYERS
LAYERS

Deep autoencoder

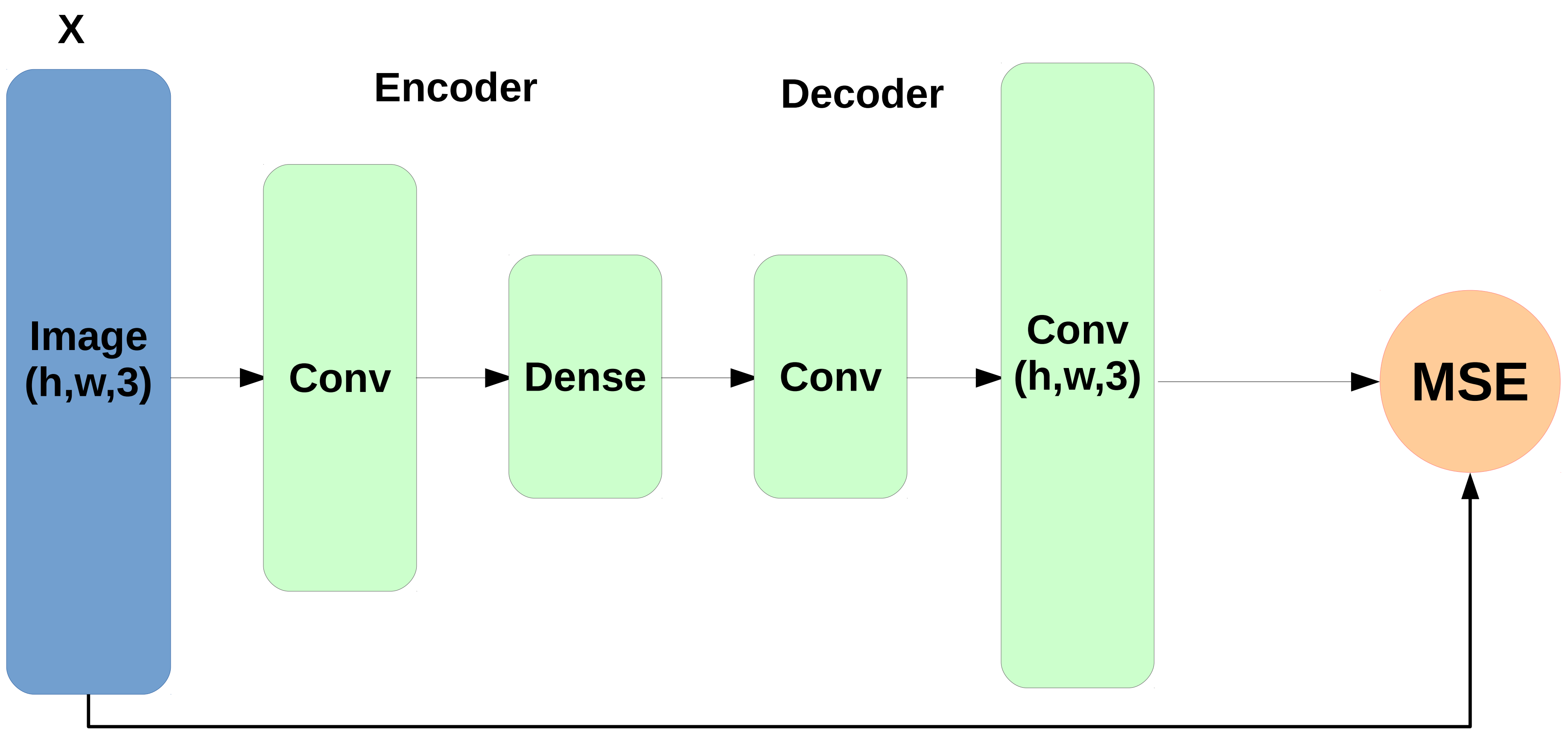
- Stack more layers!



Quiz: What if data is an image?

Deep autoencoder

- Gettin' convolutional

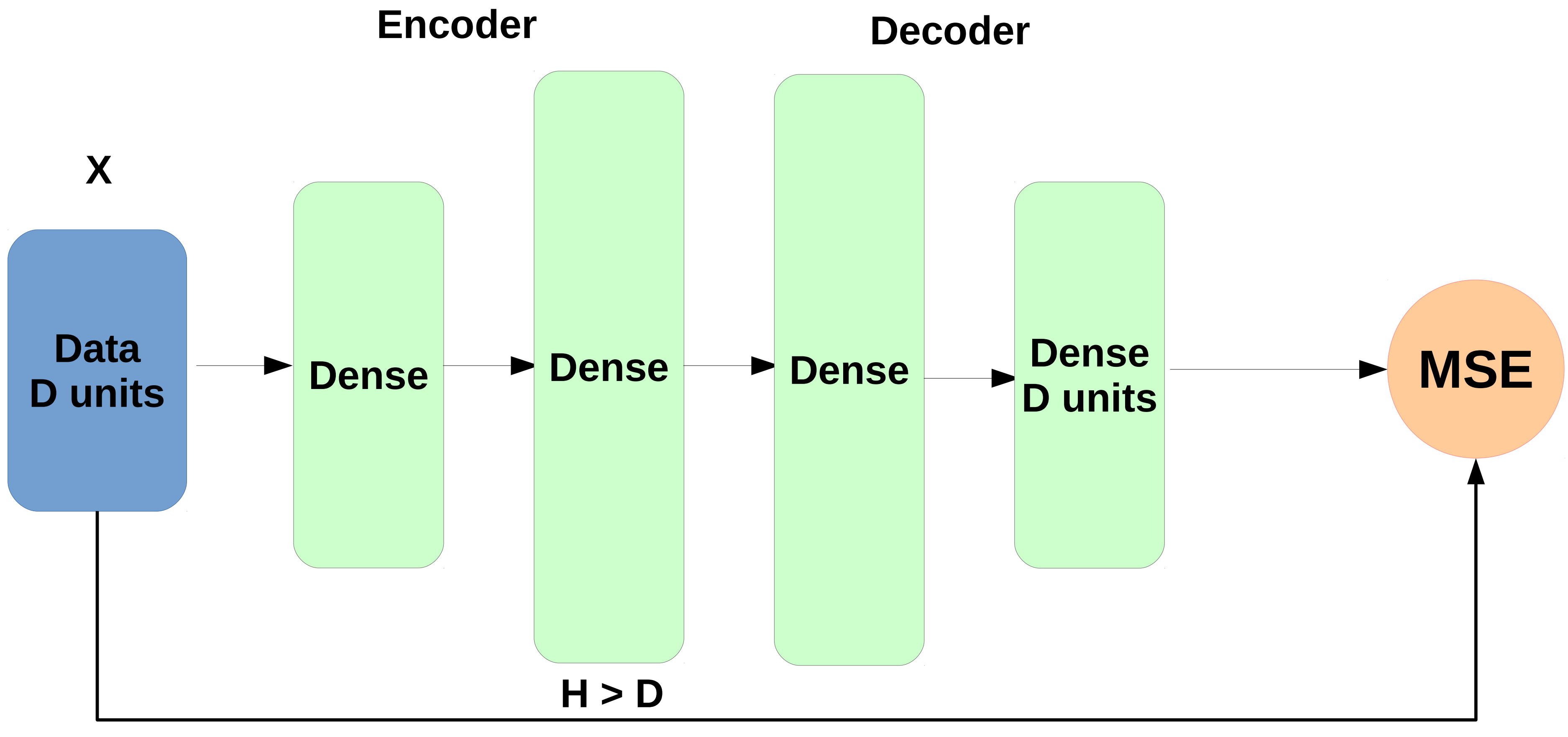


Why do we ever need that?

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

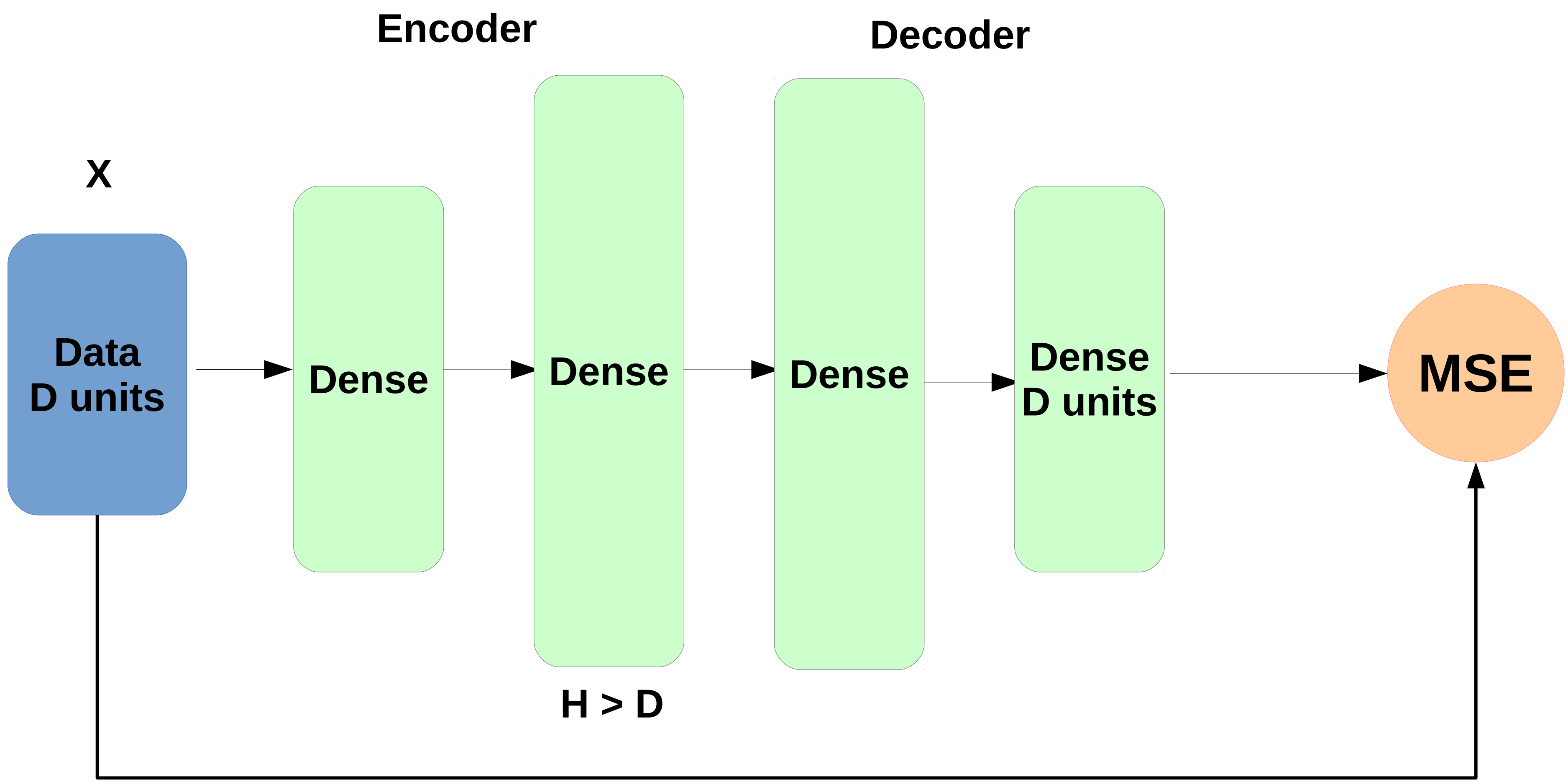
Expanding autoencoder

- Bigger/richer representation



Expanding autoencoder

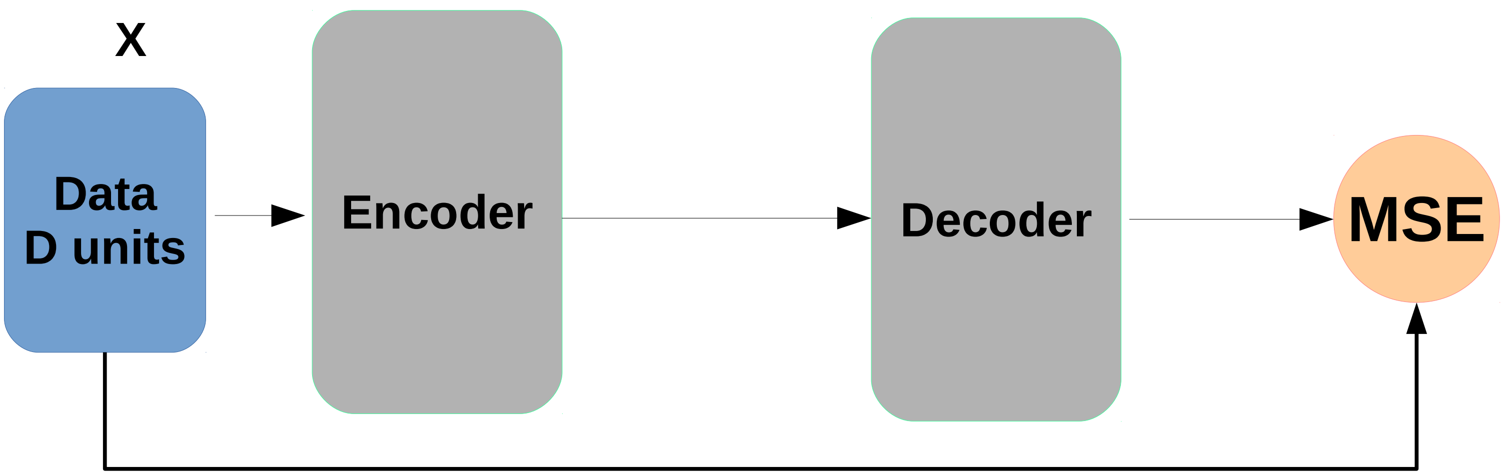
- Bigger/richer representation



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

Expanding autoencoder

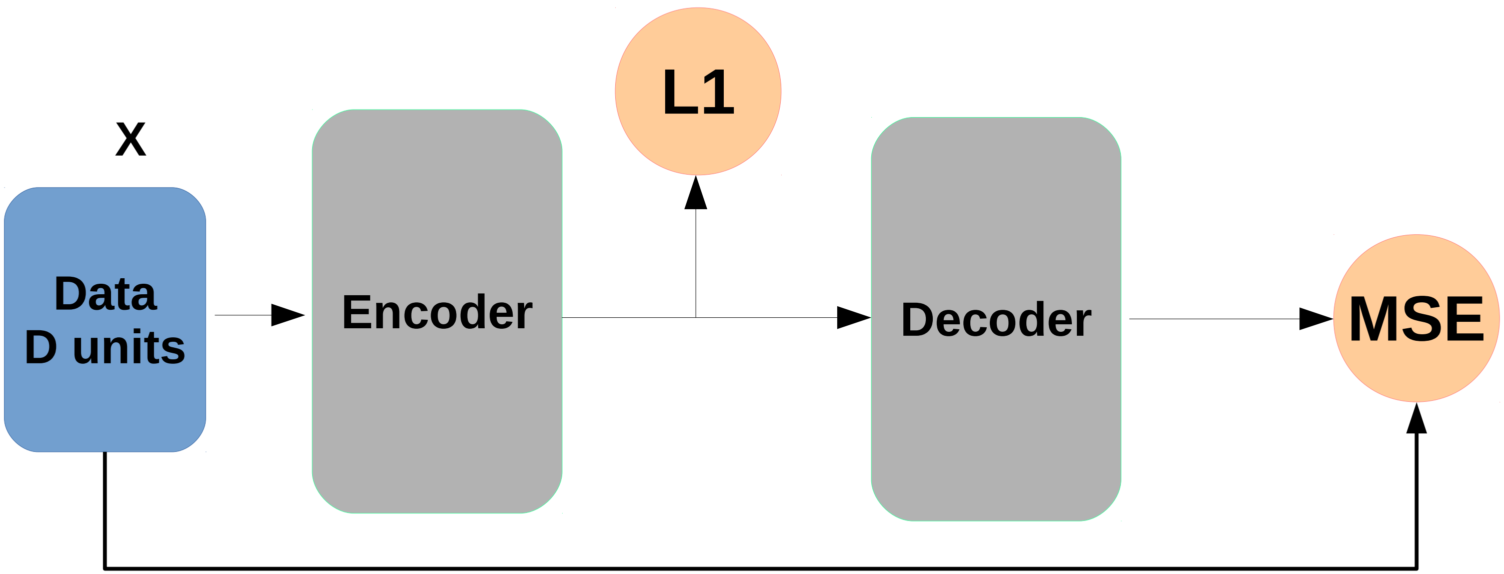
- Naive approach will learn identity function!
- Gotta regularize!



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

Sparse autoencoder

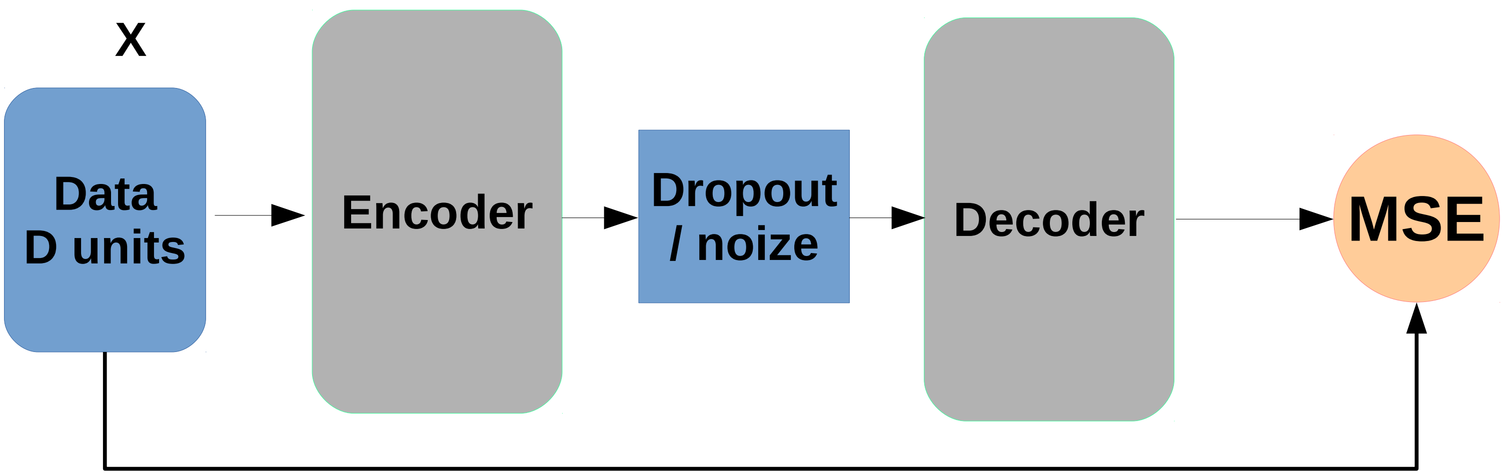
- Naive approach will learn identity function!
- Idea 1: L1 on **activations**, sparse code



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

Redundant autoencoder

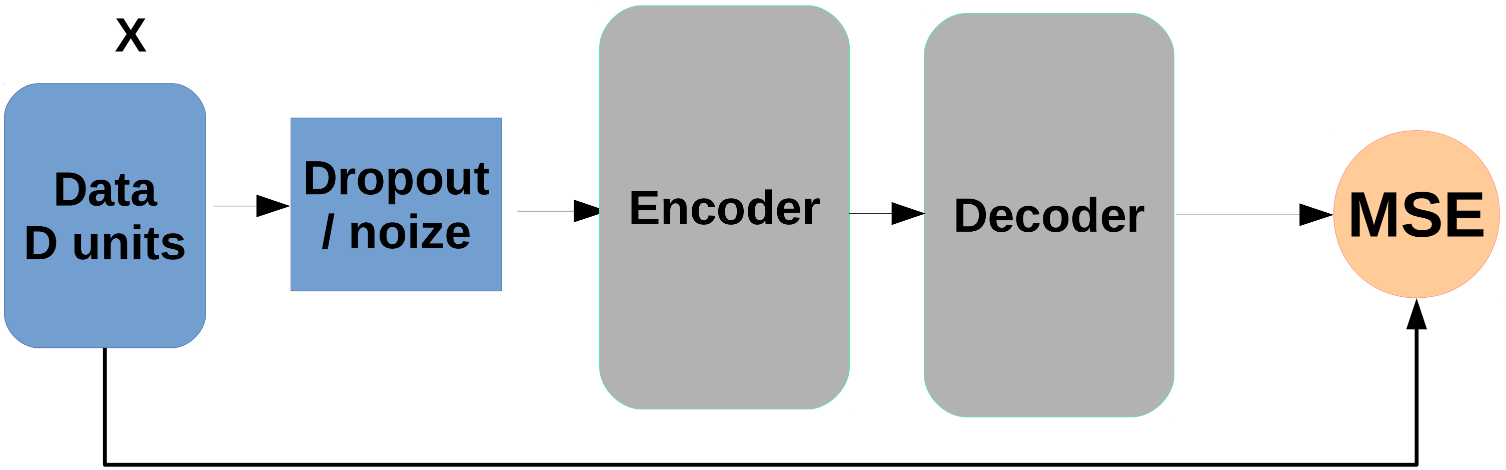
- Naive approach will learn identity function!
- Idea 2: noize/dropout, redundant code



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

Denoizing autoencoder

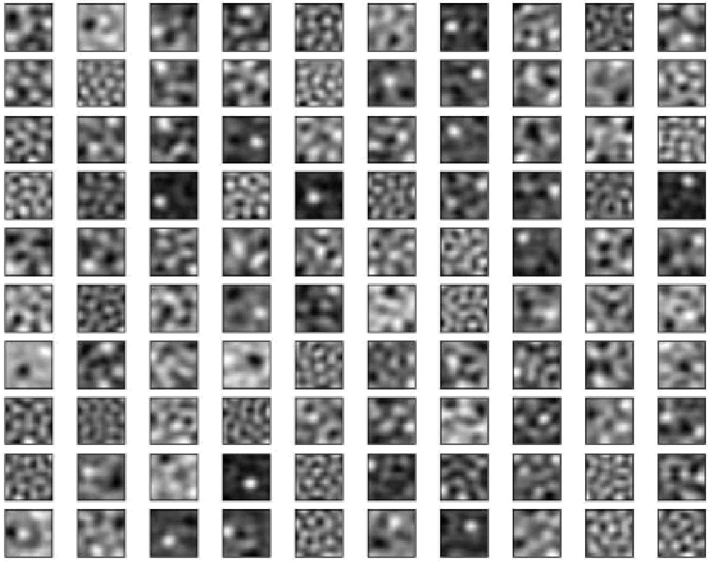
- Naive approach will learn identity function!
- Idea 3: distort input, learn to undo distorsion



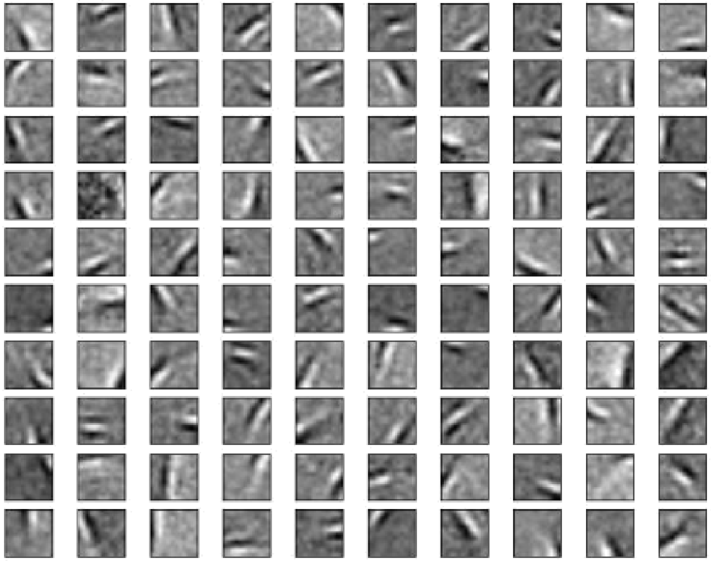
$$L = \| X - Enc(Dec(Noize(X))) \|$$

Sparse Vs Denoizing

- Filter weights, 12x12 patches



Sparse AE



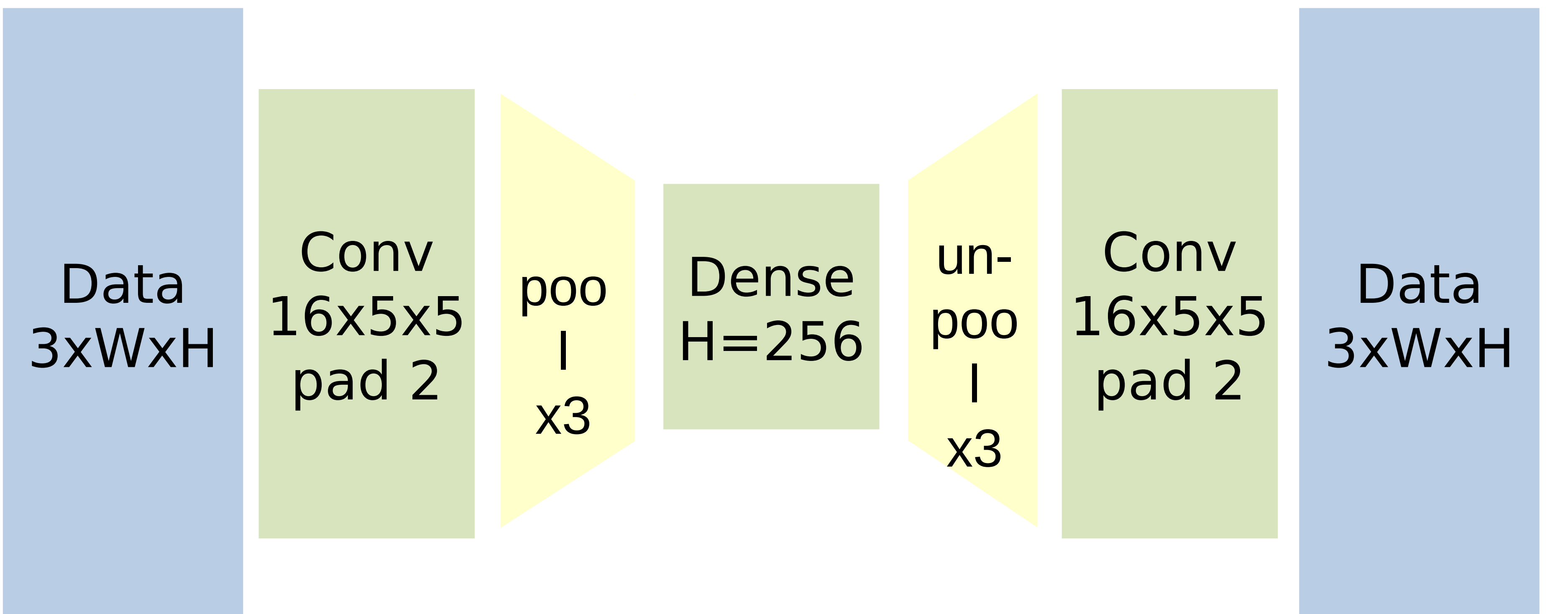
Denoizing AE

These images are actually clueless :)

[Vincent et al. 2010]

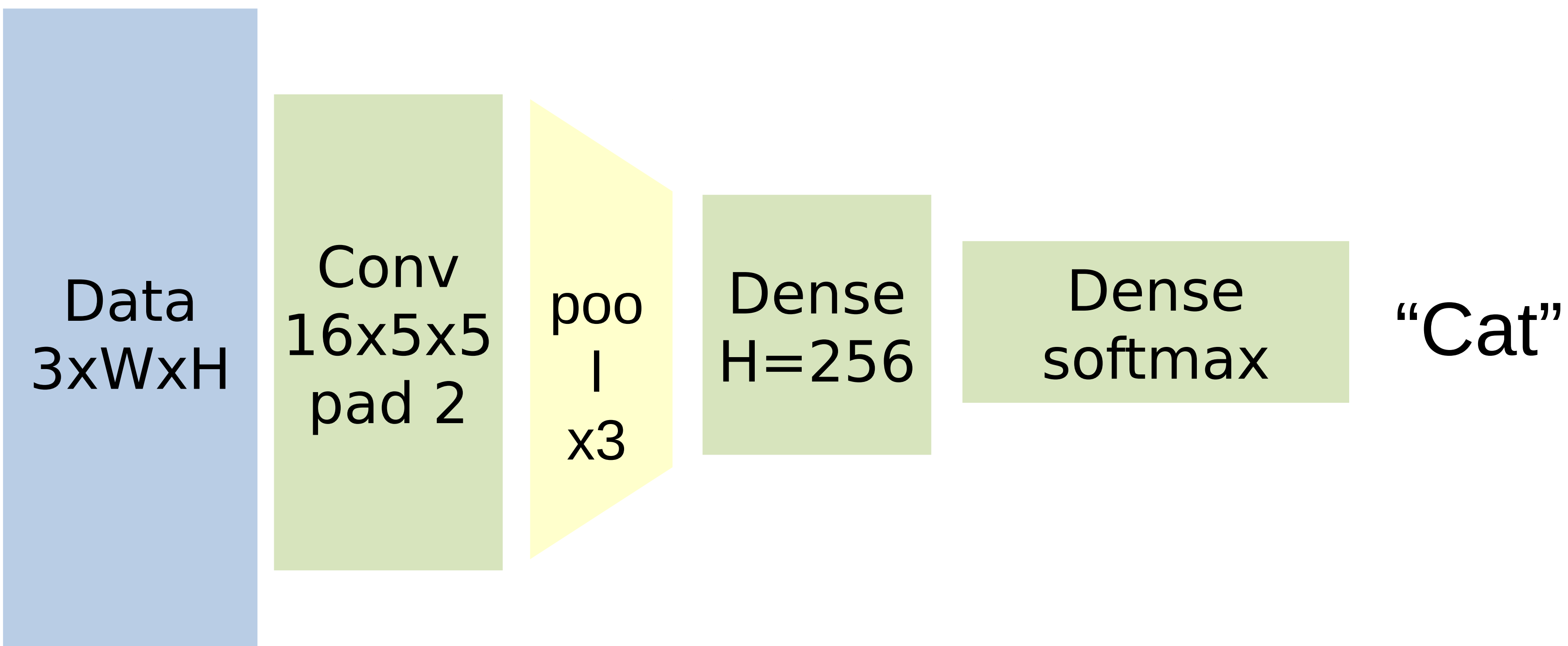
Autoencoders for pretraining

Use autoencoder as initialization



Autoencoders for pretraining

Use autoencoder as initialization



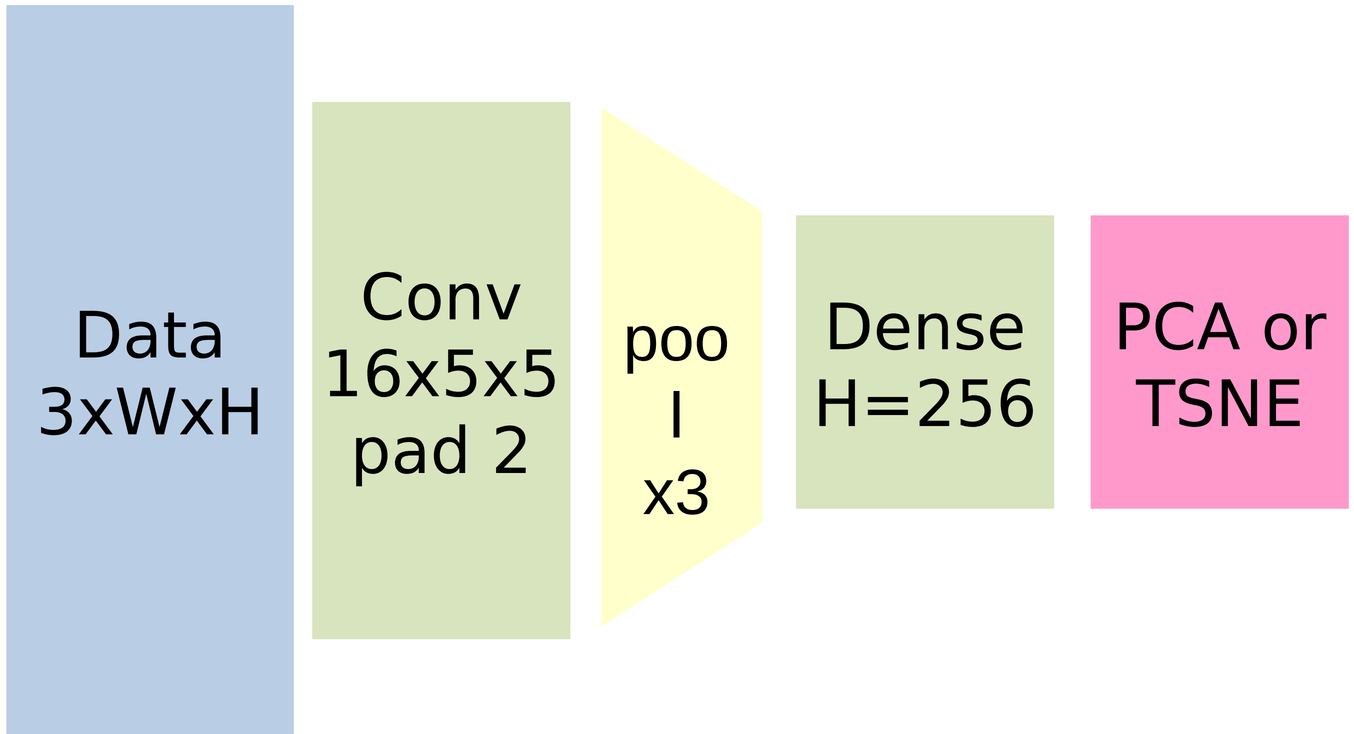
Pretraining

- Supervised pre-training (on similar task)
  - Needs labels for similar problem
  - Luckily, we have Imagenet and Model Zoo
    - Alas, it's only good for popular problems

- Unsupervised pretraining (autoencoder)
  - Needs no labels at all!
  - May learn features irrelevant to your problem
  - e.g. background sky color for object classification

Autoencoders for data visualization

Visualize data in hidden space



Autoencoders for data visualization

Visualize data in hidden space

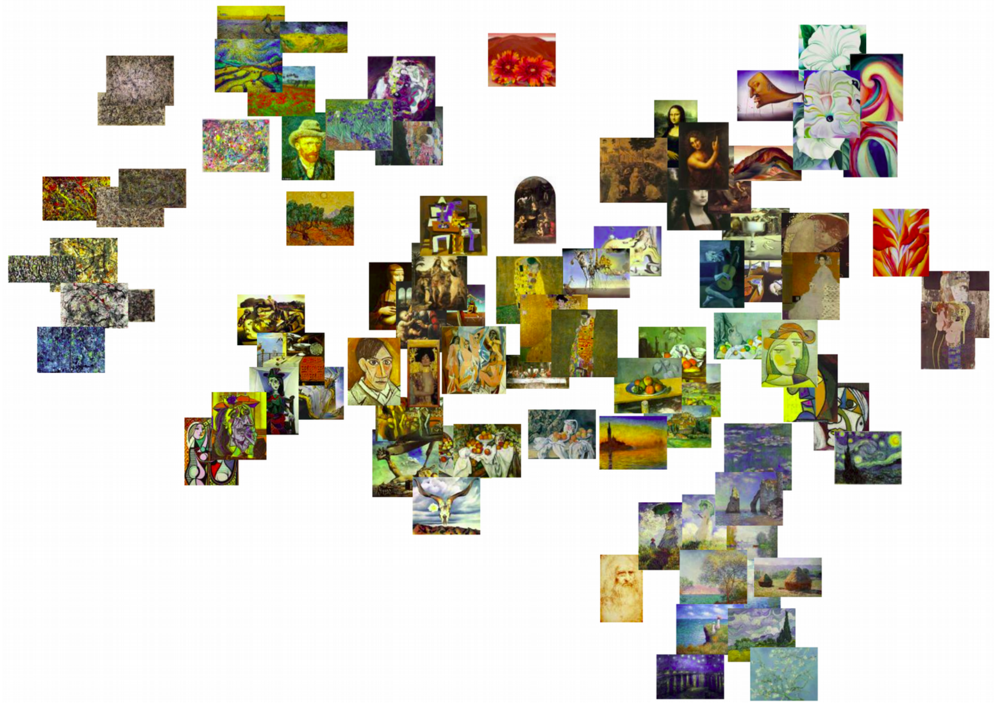


Image: https://razi.xyz/vgg2vec/picasso

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

Dirrections in hidden space:

- dec(0.5enc(img1)+0.5*enc(img2))

  is a semantic average of the two images

- dec(enc(woman)

  + enc(man_with_glasses)

  – enc(man_wo_glasses)

) = woman_with_glasses

Image morphing with AE

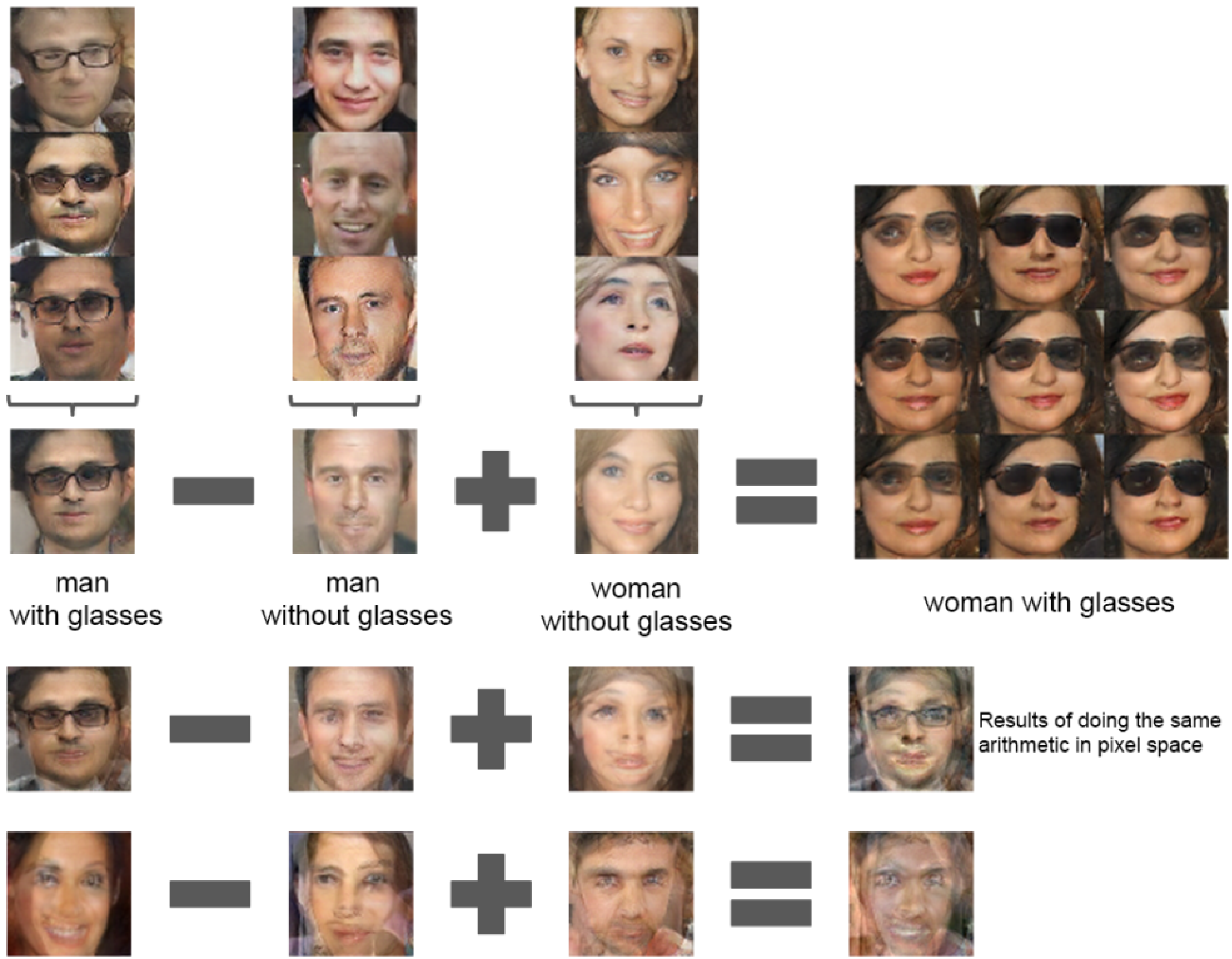
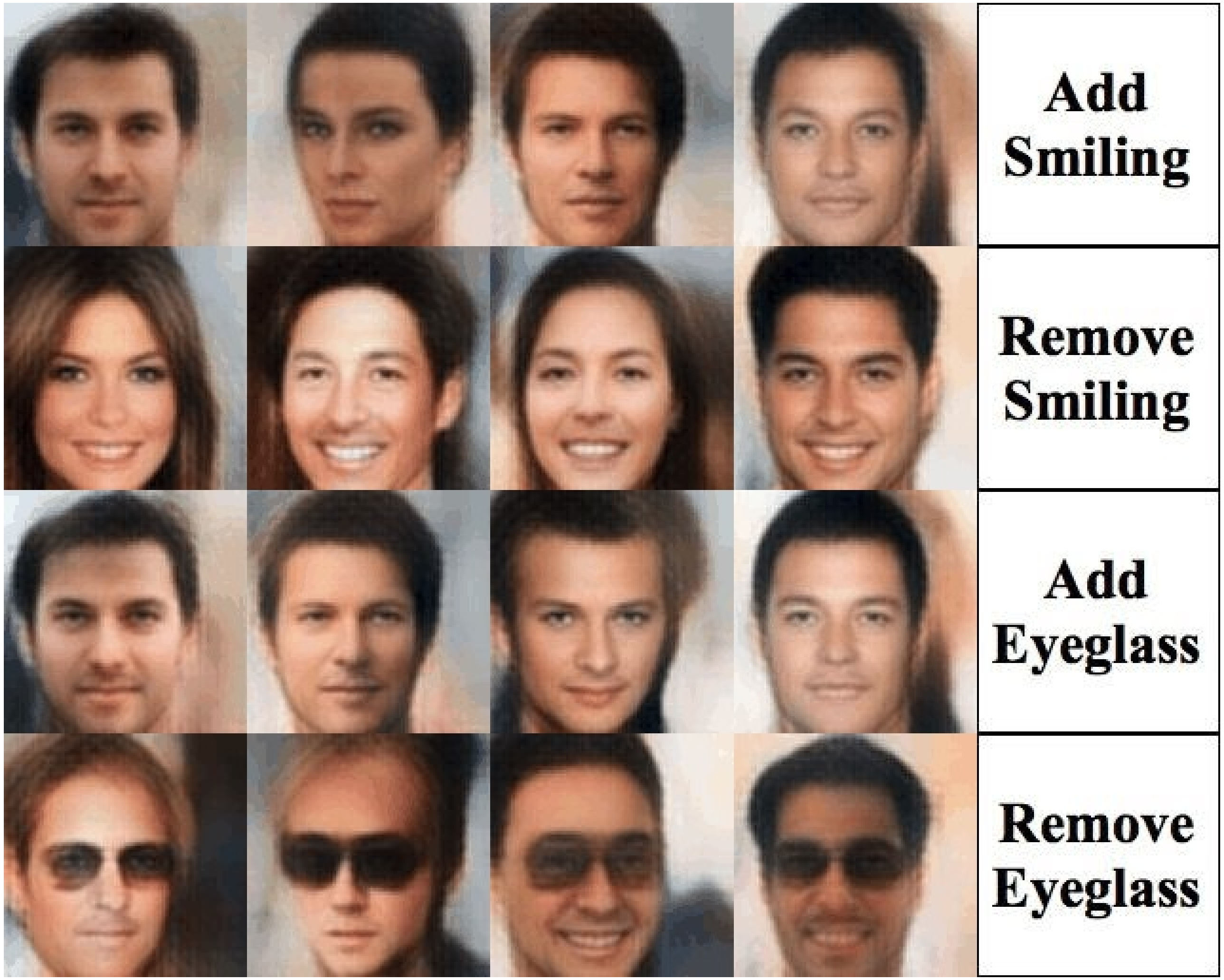


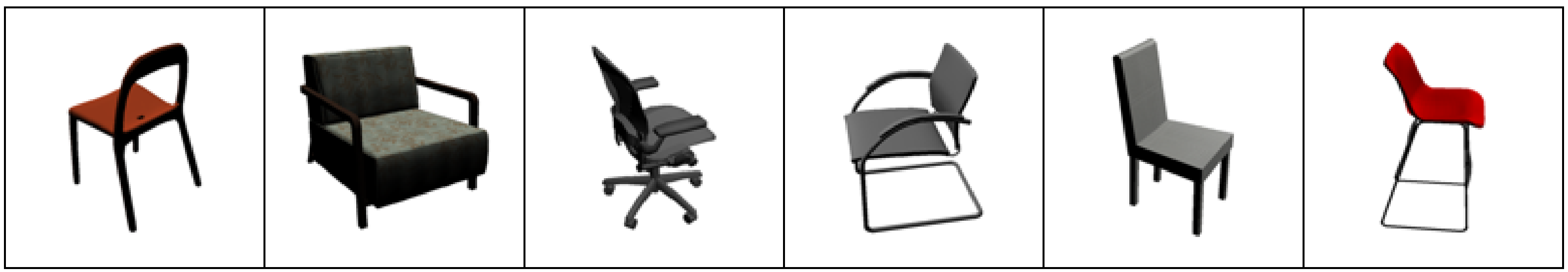
Image morphing with AE



Chapter II:
generative adversarial networks

Image generation

- Chairs (type, view, orientation)



- Classifier

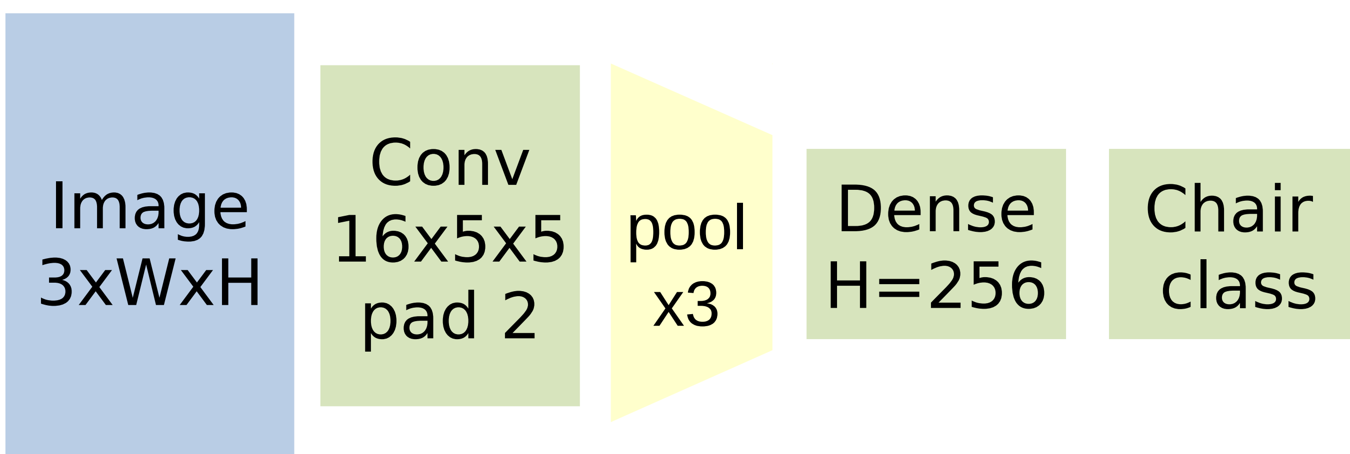
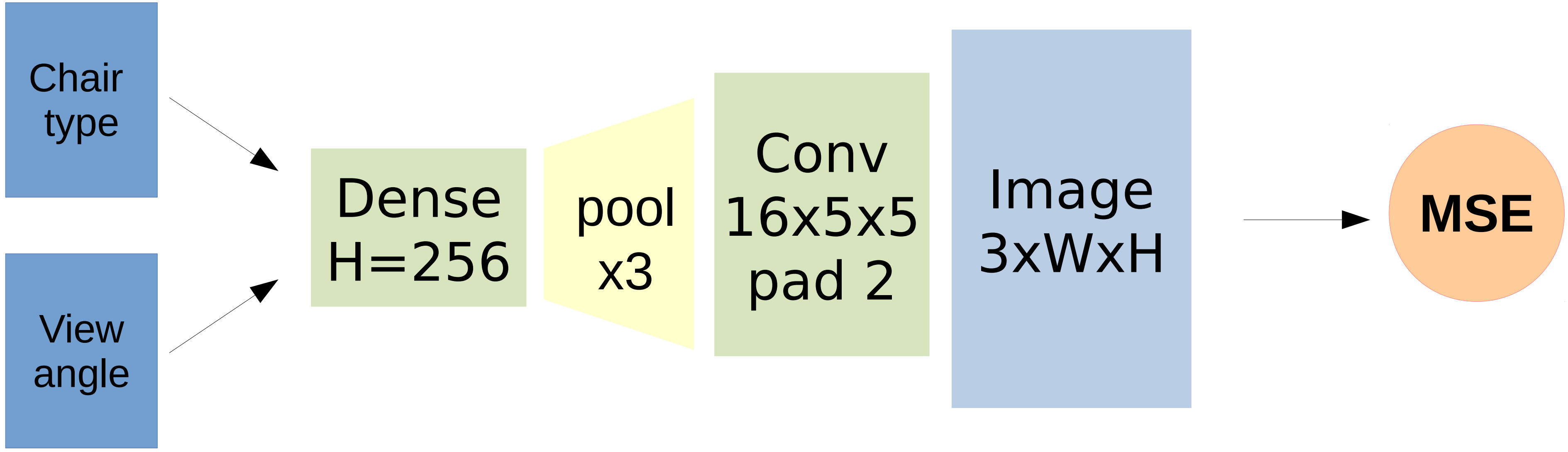


Image generation

- Generator



Mean Squared Error

Pixelwise MSE:

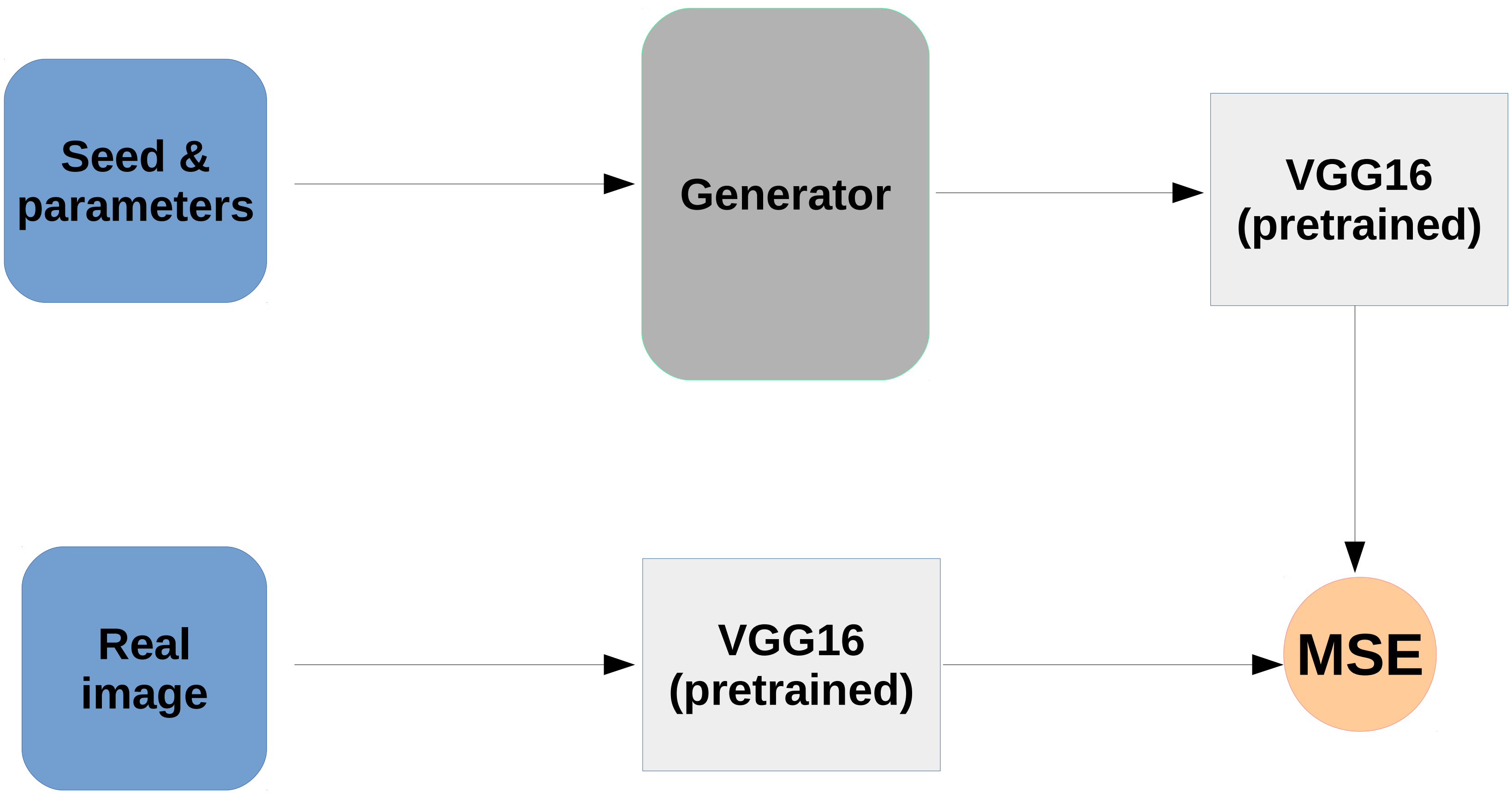
- A **“cat on the left”** is closer to **“dog on the left”** than to **“cat on the right”**
- We may want to avoid that effect
- Can we obtain image representation that is less sensitive to small shifts?

Problem: MSE sucks at this task.

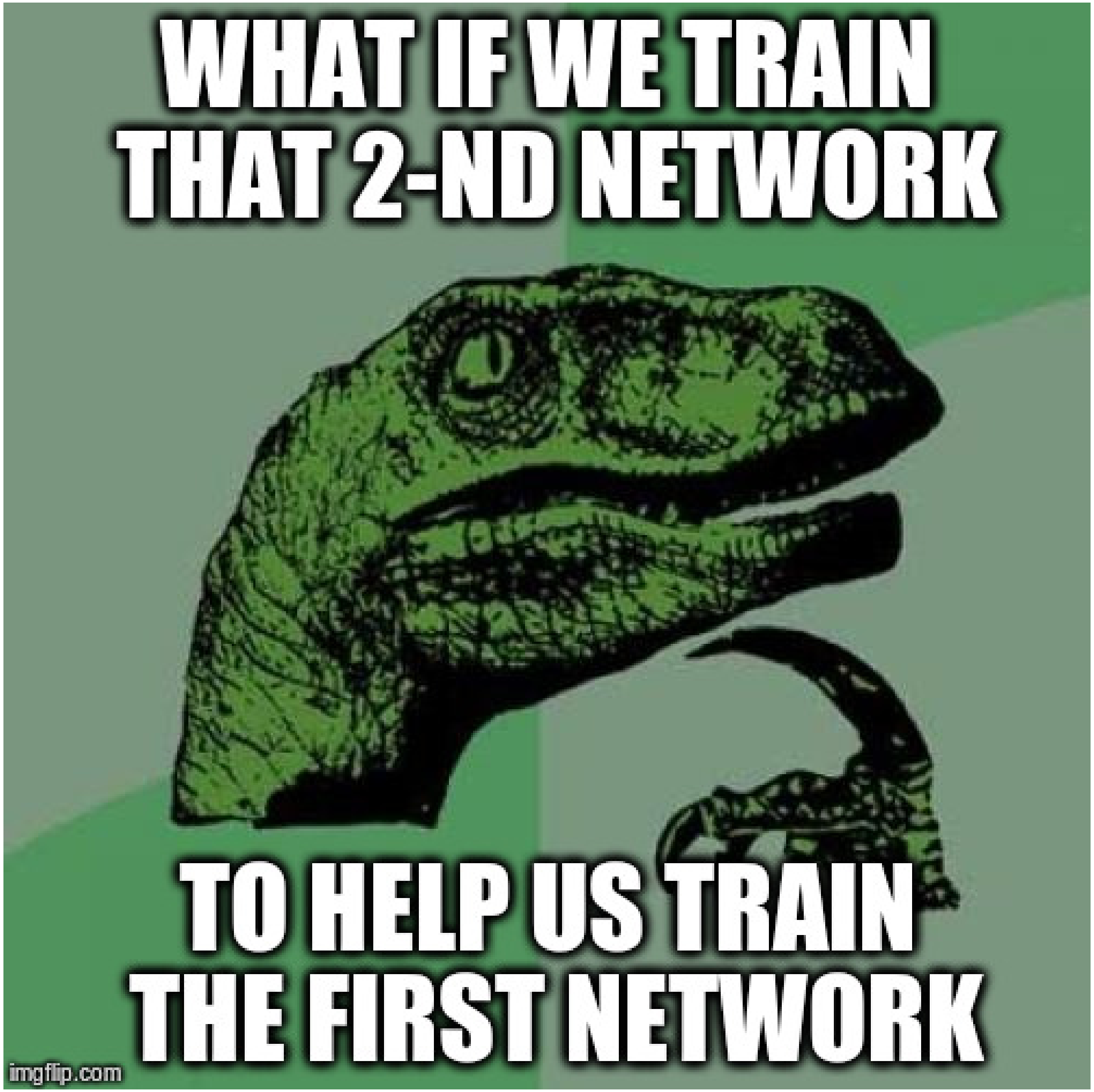
Ideas?

Do we have a representation that focuses on content while ignoring small positional shifts in data?

Sketch: using pre-trained nets

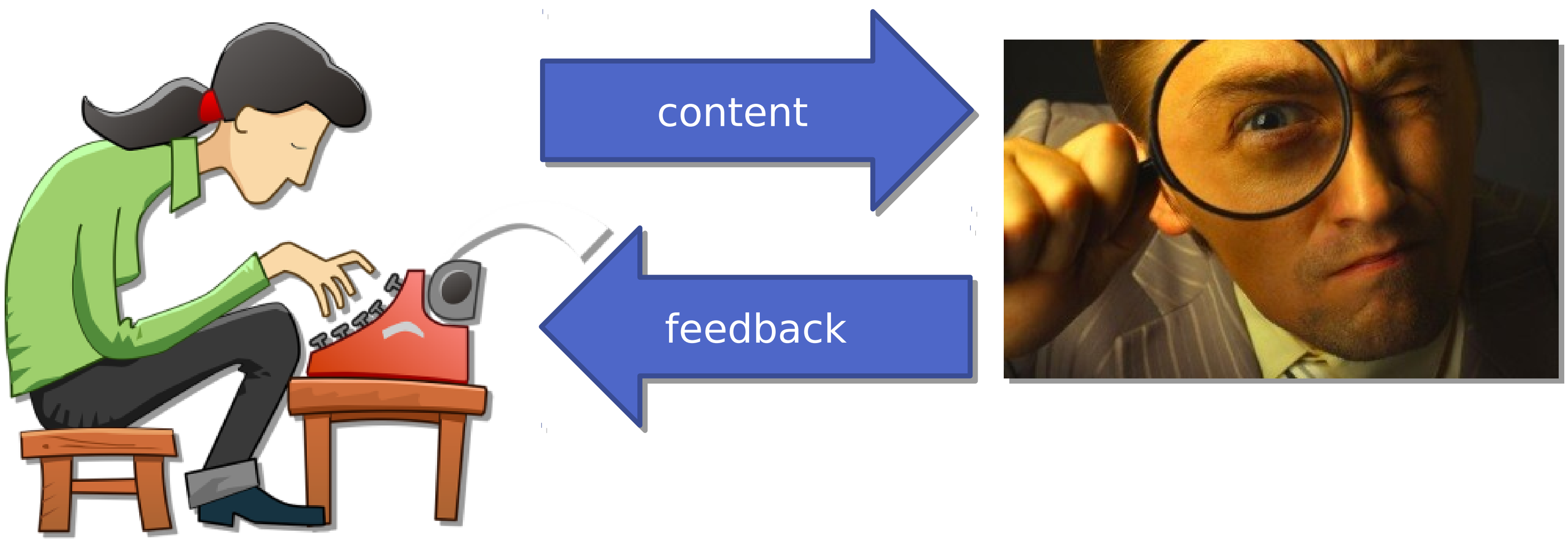


$$L = \|f(img) - f(Gen(seed))\|$$


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

Generative Adversatial Networks

Generator **Discriminator**

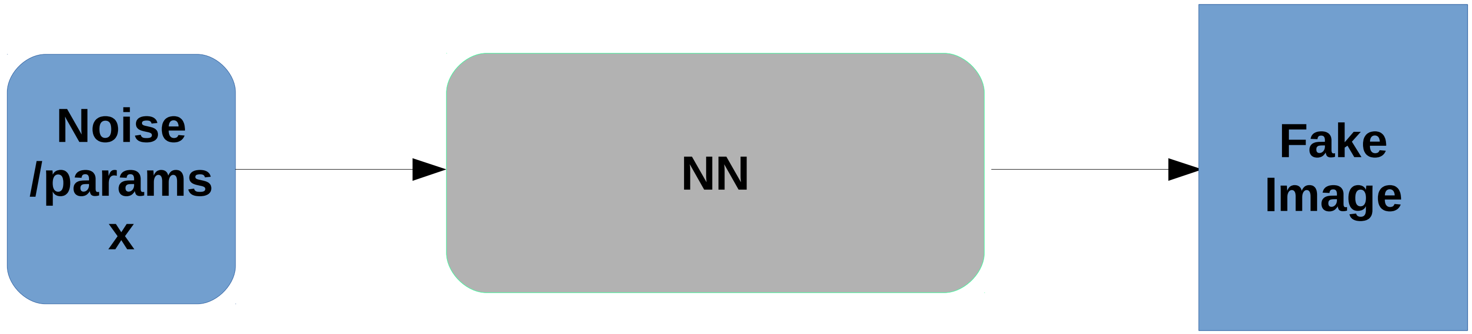


Generate image
(should be plausible)

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

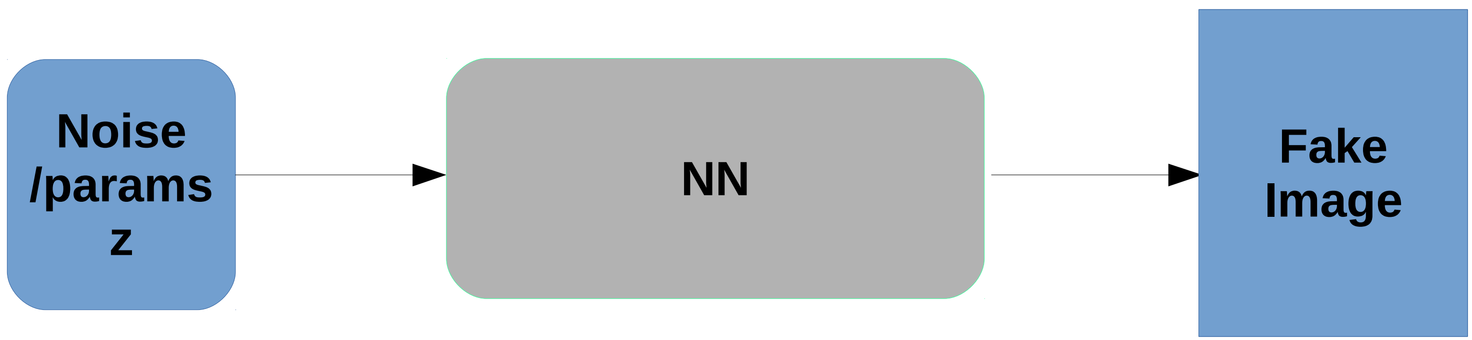
Generative Adversarial Networks

- Generator



Generative Adversarial Networks

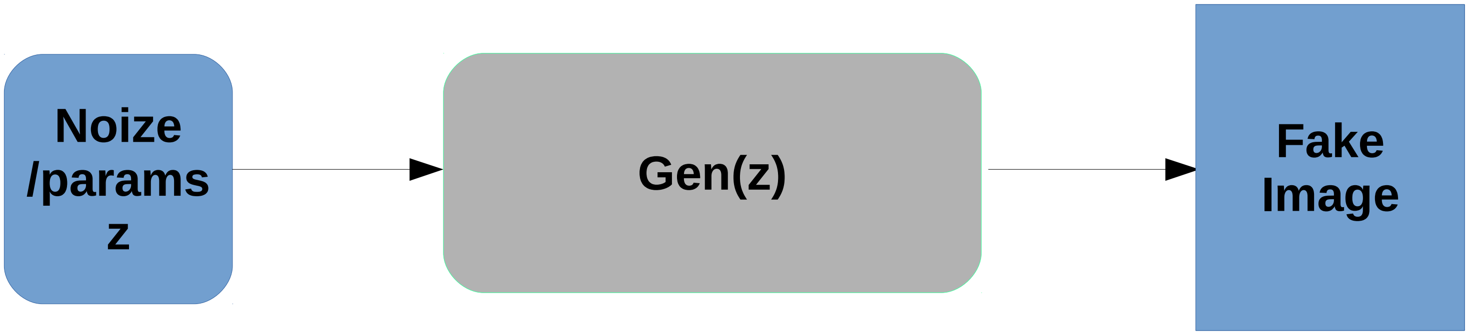
- Generator



- Discriminator

Generative Adversarial Networks

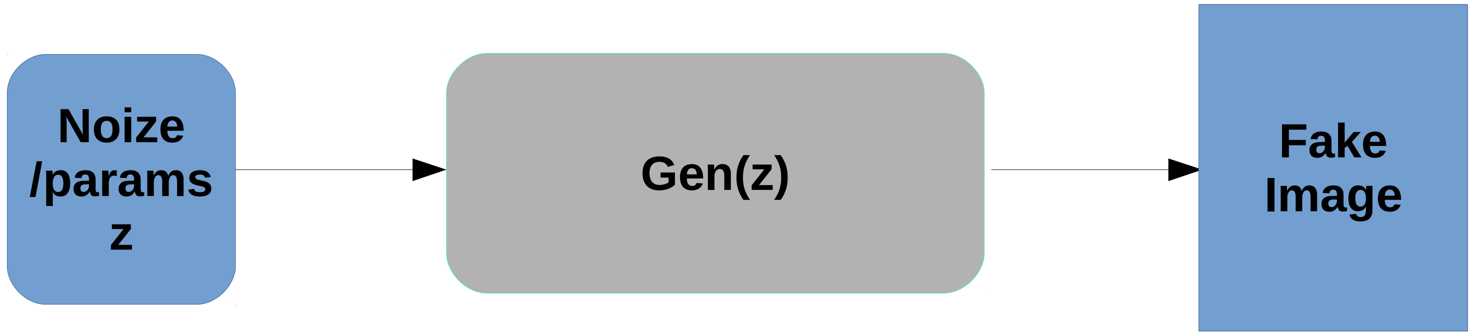
- Generator



- Discriminator

Generative Adversarial Networks

- Generator



- Discriminator

$$L_D = -\log\left[1 - Disc\left(Gen\left(z\right)\right)\right] - \log Disc\left(x\right)$$

Generative Adversarial Networks

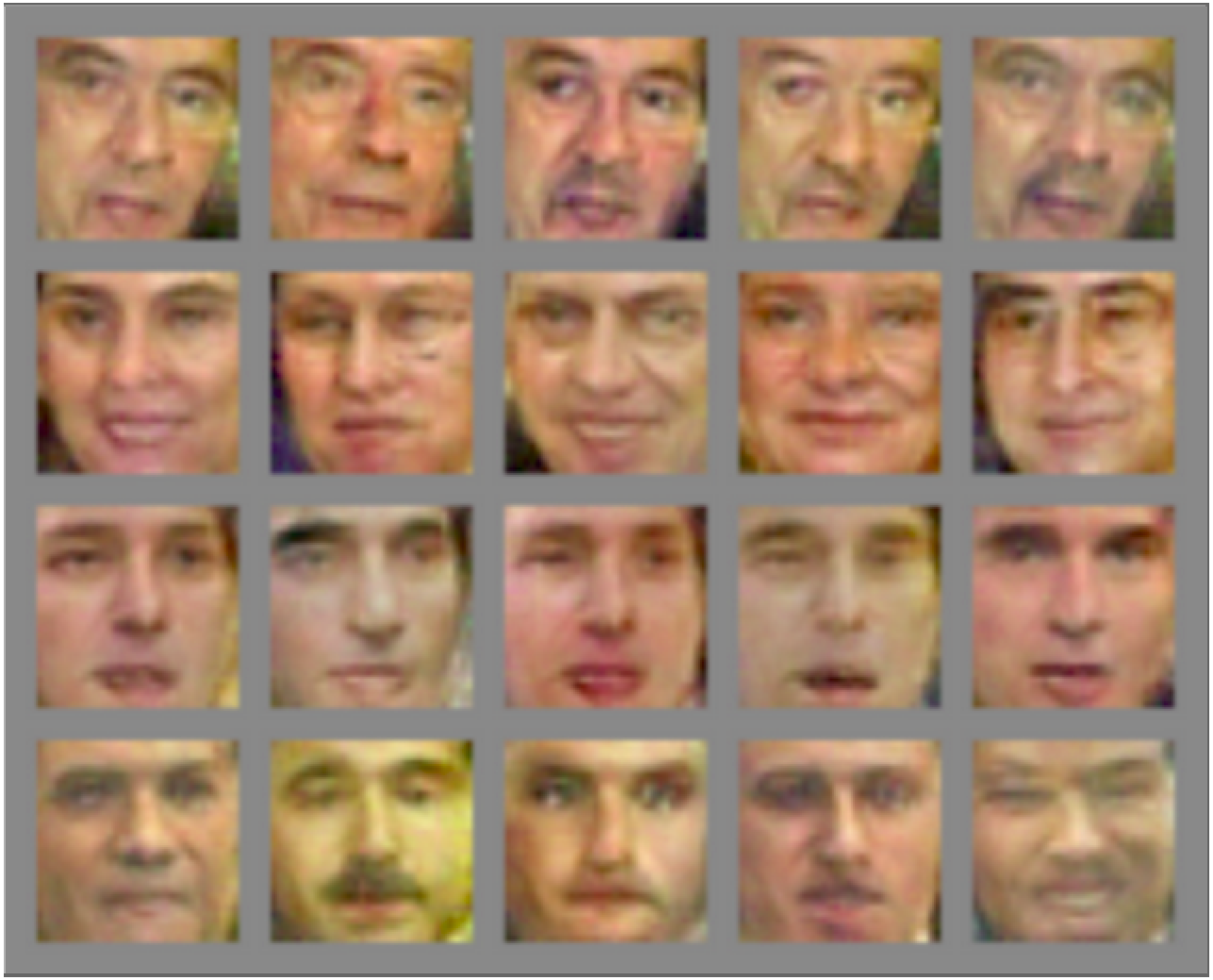
Algorithm

- sample noise **z** and images **x**
- for k in 1...K

  Train discriminator(**x**), discriminator(generator(**z**))
- For m in 1...M

  Train generator(z)

Generative Adversarial Networks



Adversarial applications

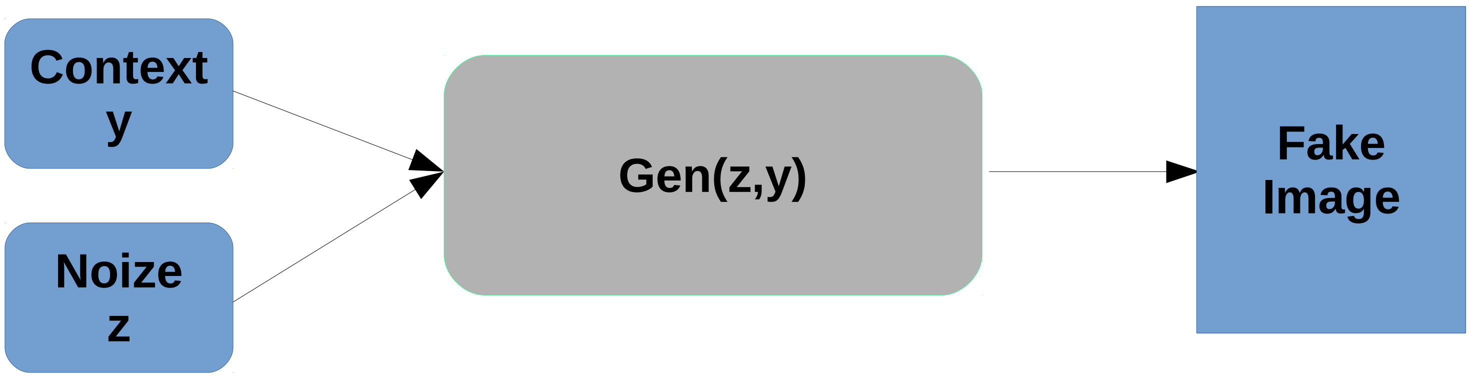
- Funsies with image generation
- Image colorization/upscaling/adaptation/refining
- Seq2seq: machine translation, conversarions, image2caption, text2speech, image2latex
- Domain adaptation
- Adversarial examples

Adversarial applications

- Funsies with image generation
- **Image colorization/upscaling/... .../adaptation/refining**
- Seq2seq: machine translation, conversarions, image2caption, text2speech, image2latex
- Domain adaptation
- Adversarial examples

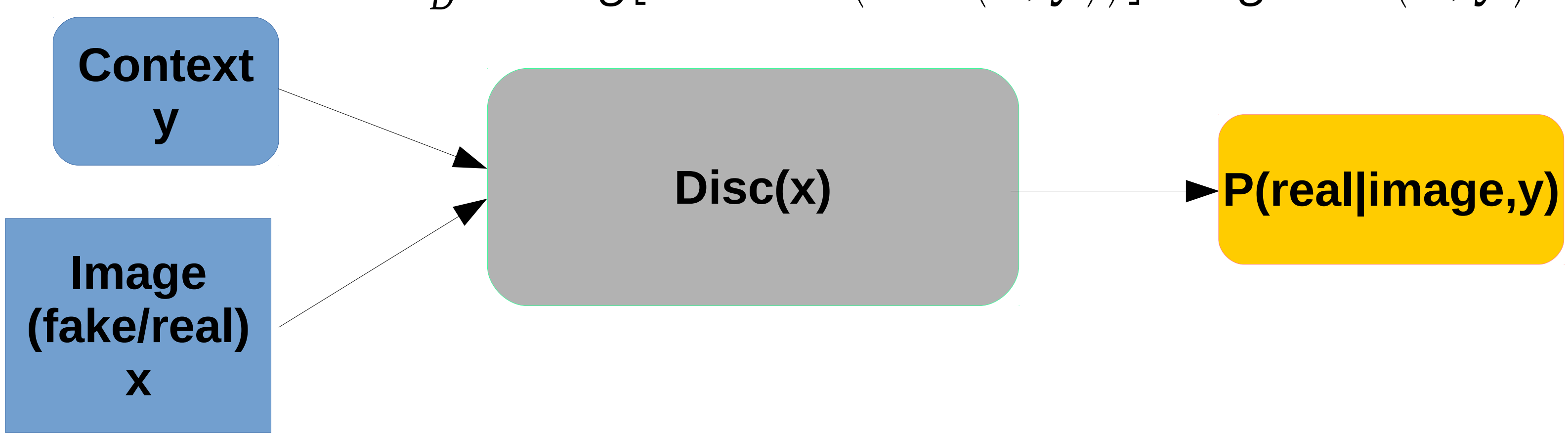
Conditional Adversarial Networks

- Generator $$L_G = -\log Disc\left(Gen\left(z, y\right)\right)$$



- Discriminator

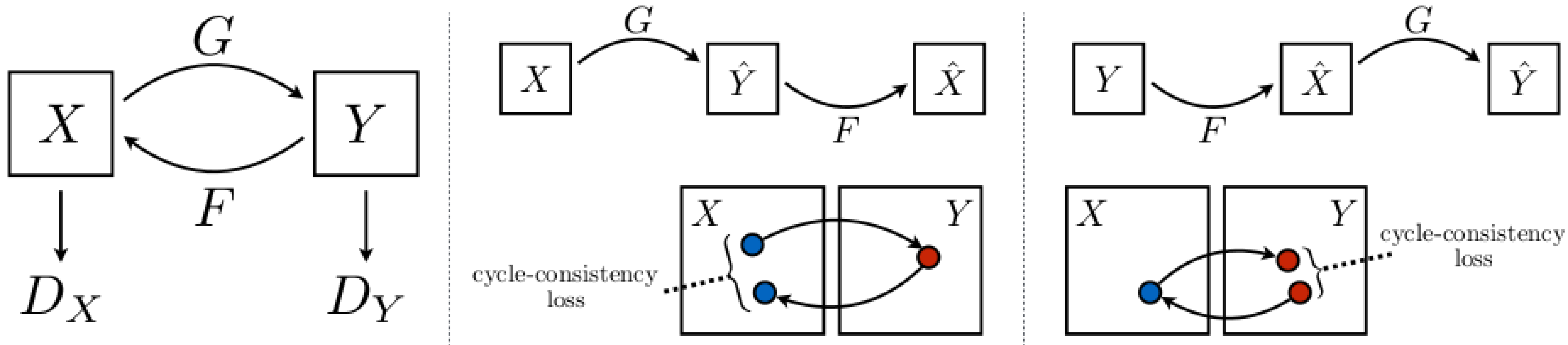
$$L_D = -\log\left[1 - Disc\left(Gen\left(z, y\right)\right)\right] - \log Disc\left(x, y\right)$$



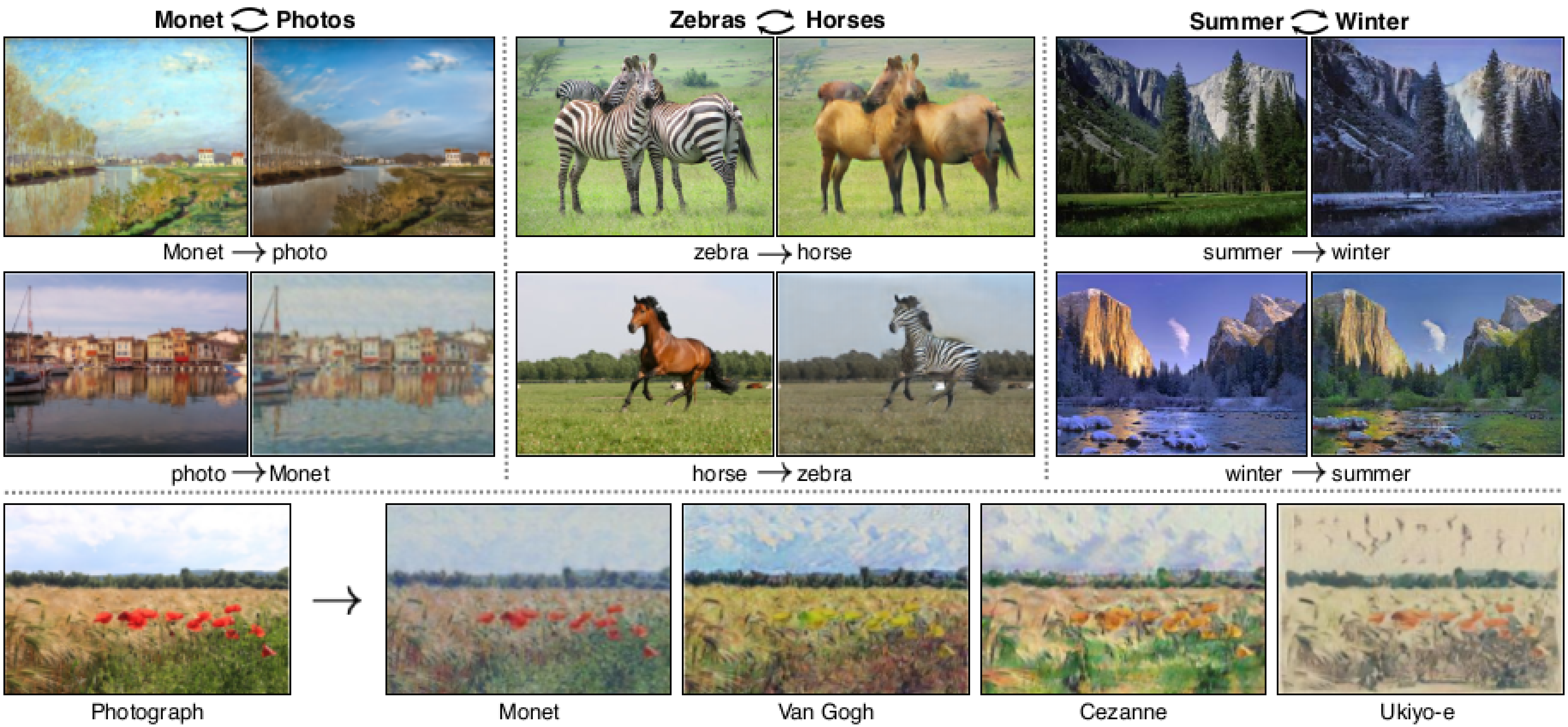
Cycle GAN for unpaired images

Idea: if we don't have image pairs,

- train two conditional generators
  G(z,y) → x, F(g,x) → y
- use non-conditional D(x), D(y)
- make sure |F(G(x)) – x| → min



Cycle GAN for unpaired images



Adversarial applications

- Funsies with image generation
- Image colorization/upscaling/adaptation/refining
- **Seq2seq: machine translation, conversarions, image2caption, text2speech, image2latex**
- Domain adaptation
- Adversarial examples

more on these in reinforcement learning course

Adversarial applications

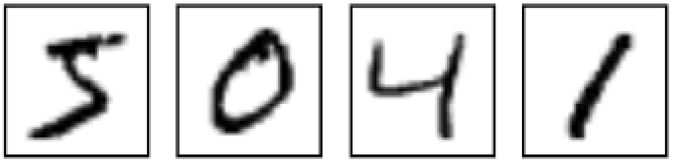
- Funsies with image generation
- Image colorization/upscaling/adaptation/refining
- Seq2seq: machine translation, conversarions, image2caption, text2speech, image2latex
- **Domain adaptation**
- Adversarial examples

Adversarial domain adaptation

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

Adversarial domain adaptation

- Two domains, for example:
  - handwritten digits
  - house number digits
- First domain is labeled, second isn't.
- You want your model to work on the second domain

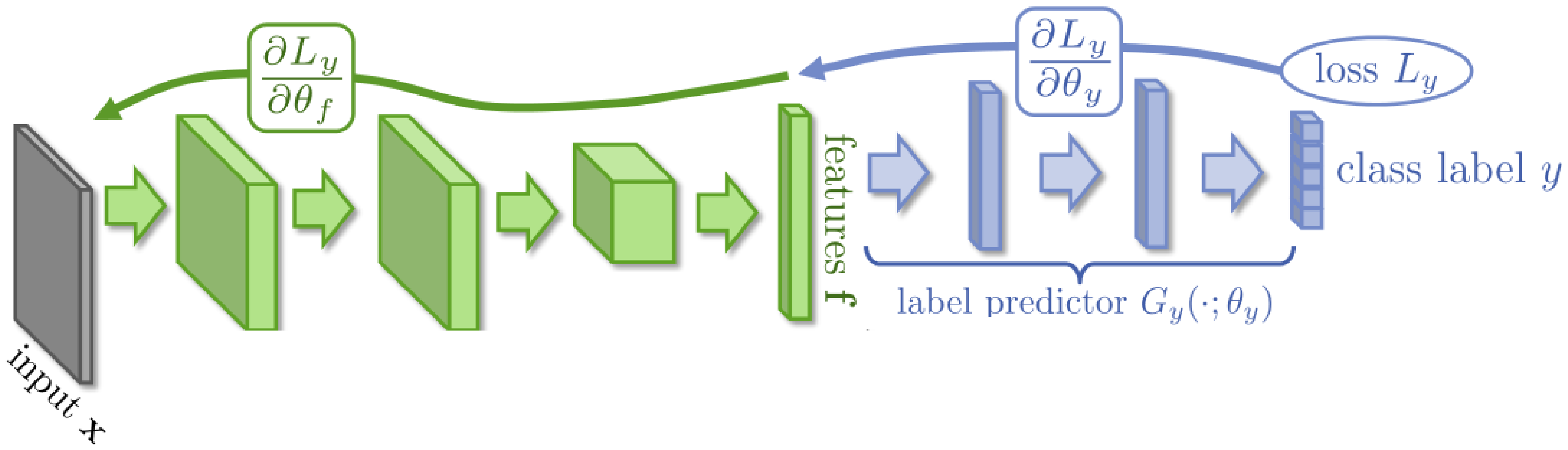


Labeled data

Unlabeled data

Domain adaptation

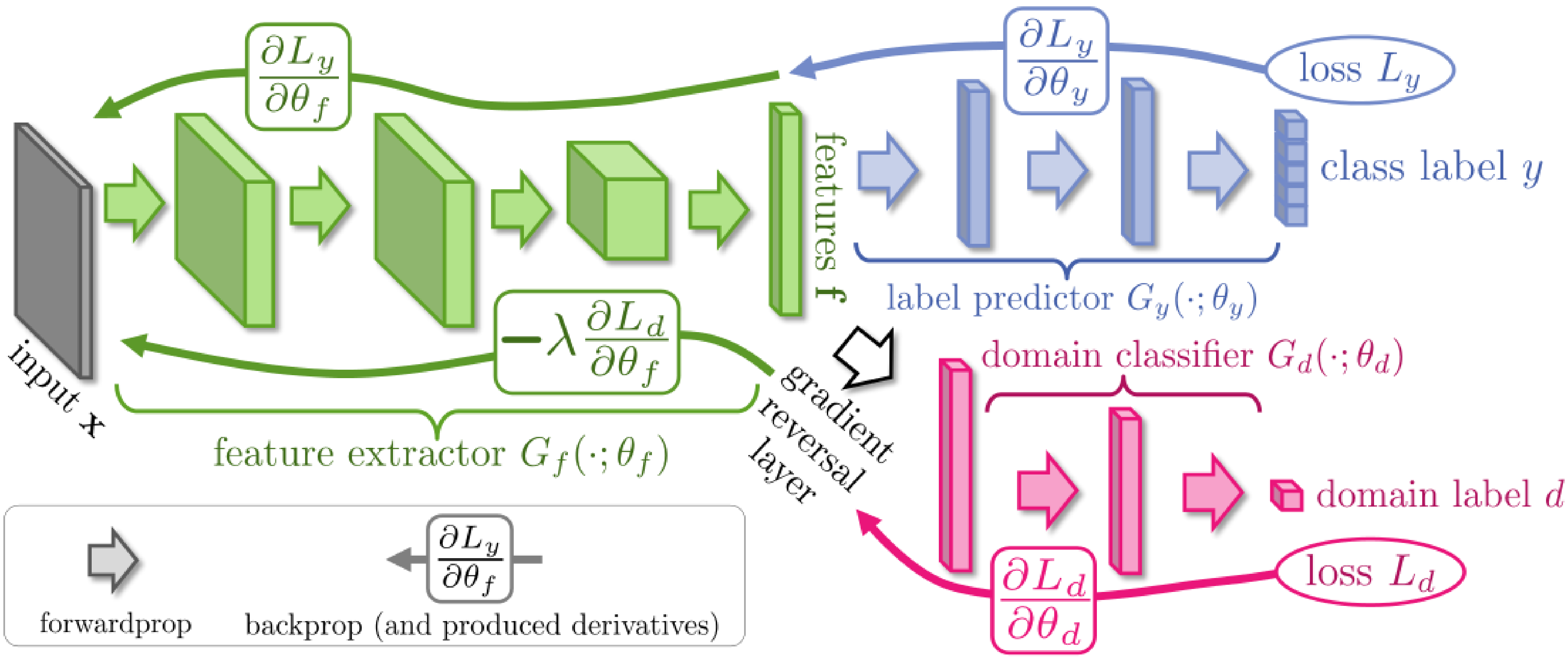
- Idea: discriminator should not be able to distinguish features on two domains



Source: https://arxiv.org/abs/1409.7495

Domain adaptation

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



Domain adaptation

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

$$-\log P\left(real \mid h\left(x_{real}\right)\right) - \log\left[1 - P\left(real \mid h\left(x_{mc}\right)\right)\right] \rightarrow \min_{discriminator}$$

$$L_{classifier}\left(y_{mc}, y\left(h\left(x_{mc}\right)\right)\right) - \log P\left(real \mid h\left(x_{mc}\right)\right) \rightarrow \min_{classifier}$$

What we learned?

Something? Anything?!

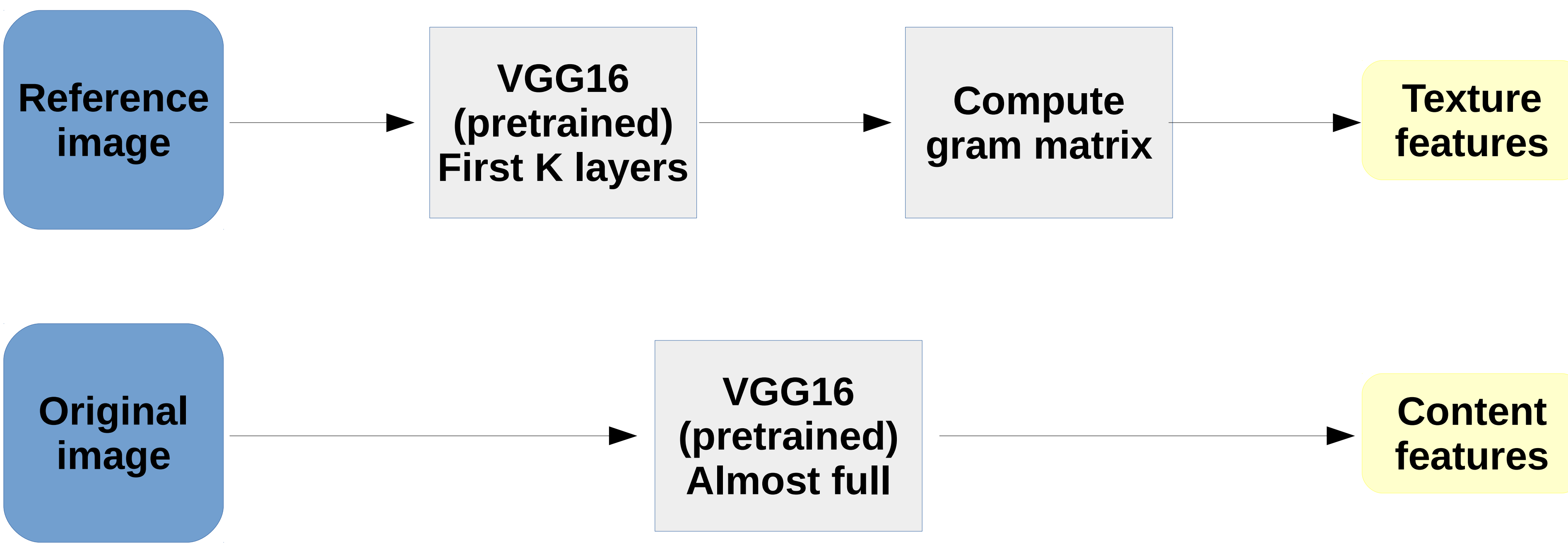
Art style transfer

- Ideas?

Art style transfer

- Formulate and optimize texture loss

$$L = \| Texture(x_{ref}) - Texture(x_{cand}) \| + \| Content(x_{orig}) - Content(x_{cand}) \|$$



Art style transfer

+

=

Art style transfer

+

=