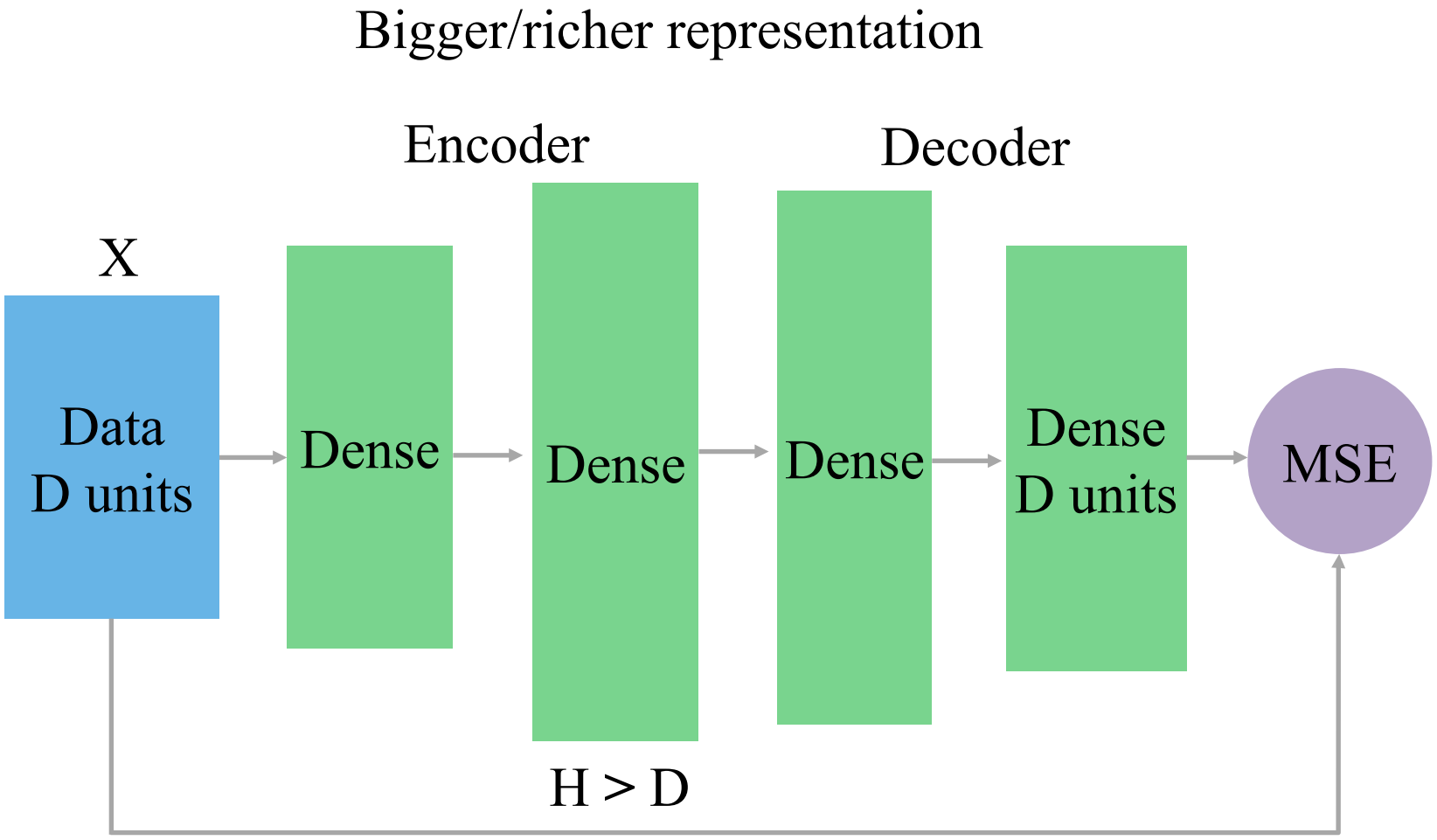
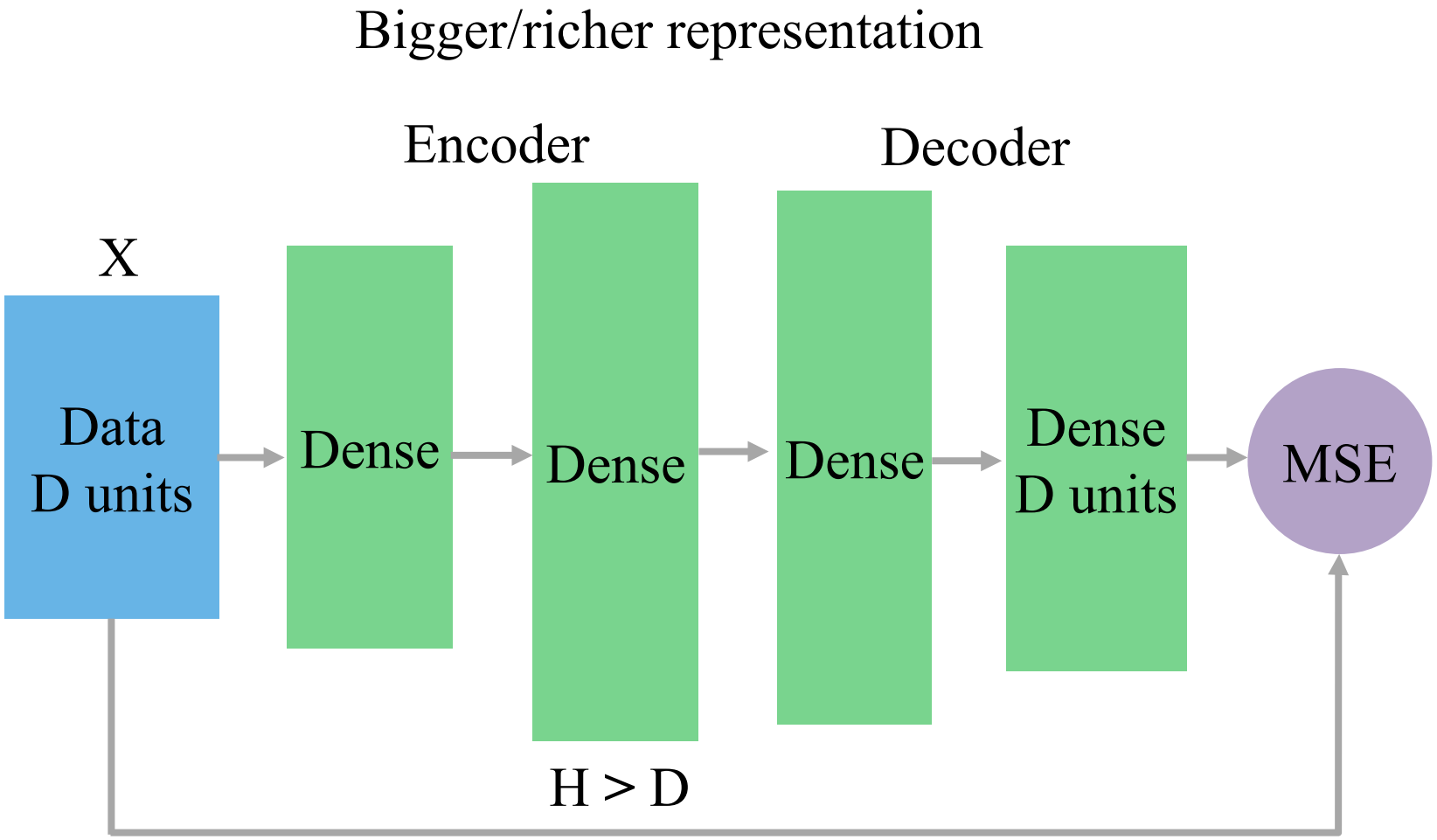
Why do we ever need that?

- 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



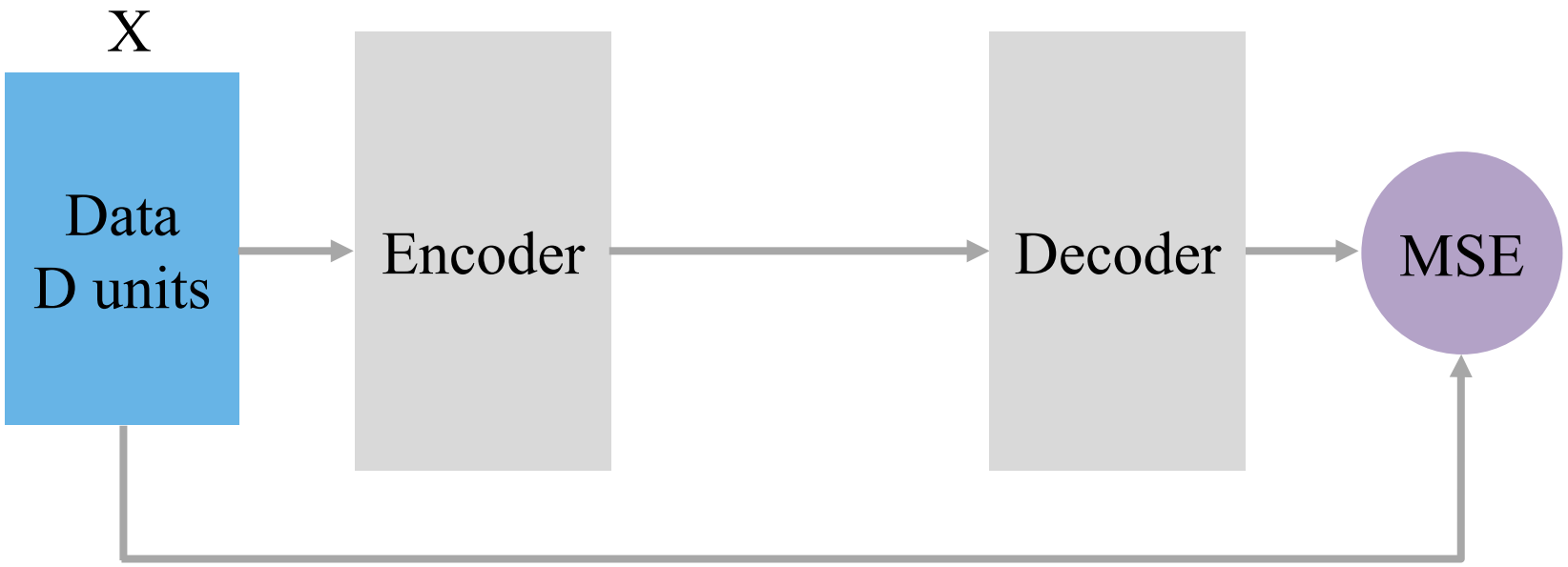
Expanding autoencoder



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

Expanding autoencoder

Naive approach will learn identity function!
Gotta regularize!

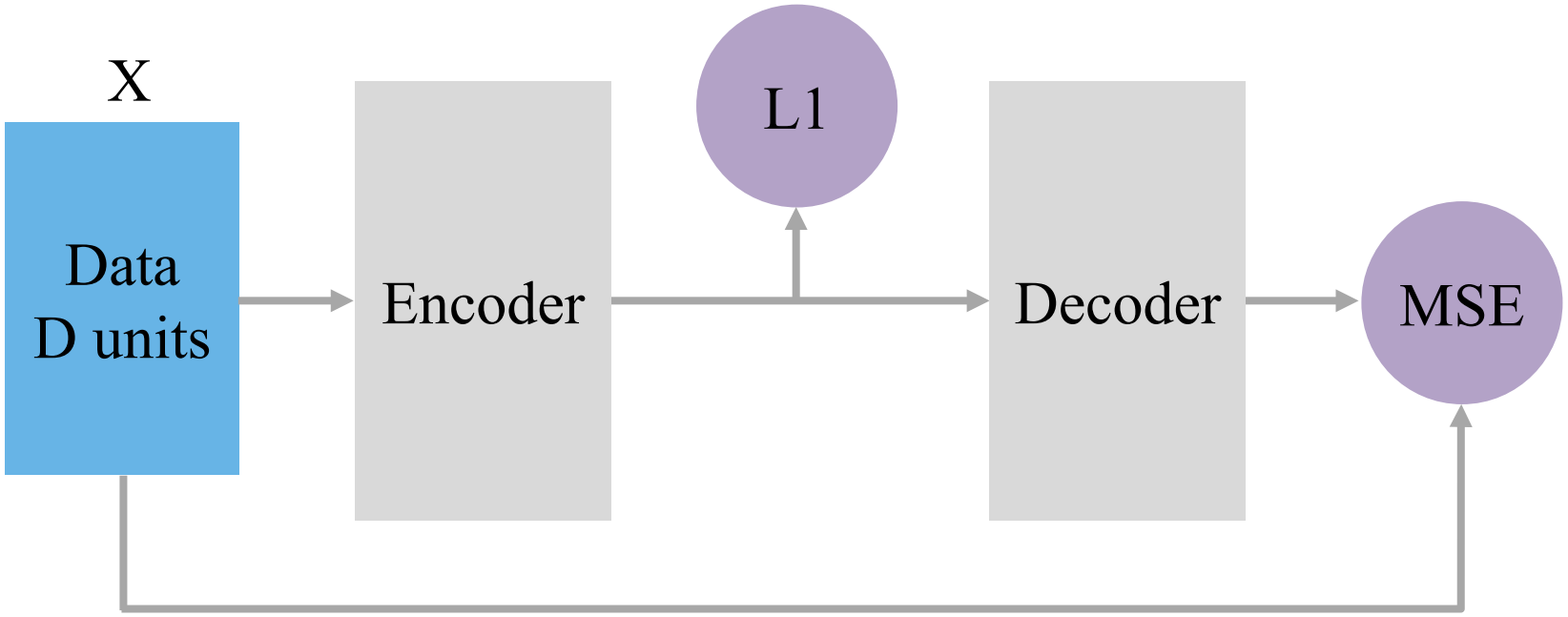


$$L = \left\| X - Dec\left(Enc(X)\right) \right\|$$

Sparse autoencoder

Naive approach will learn identity function!

Idea 1: L1 on **activations**, sparse code

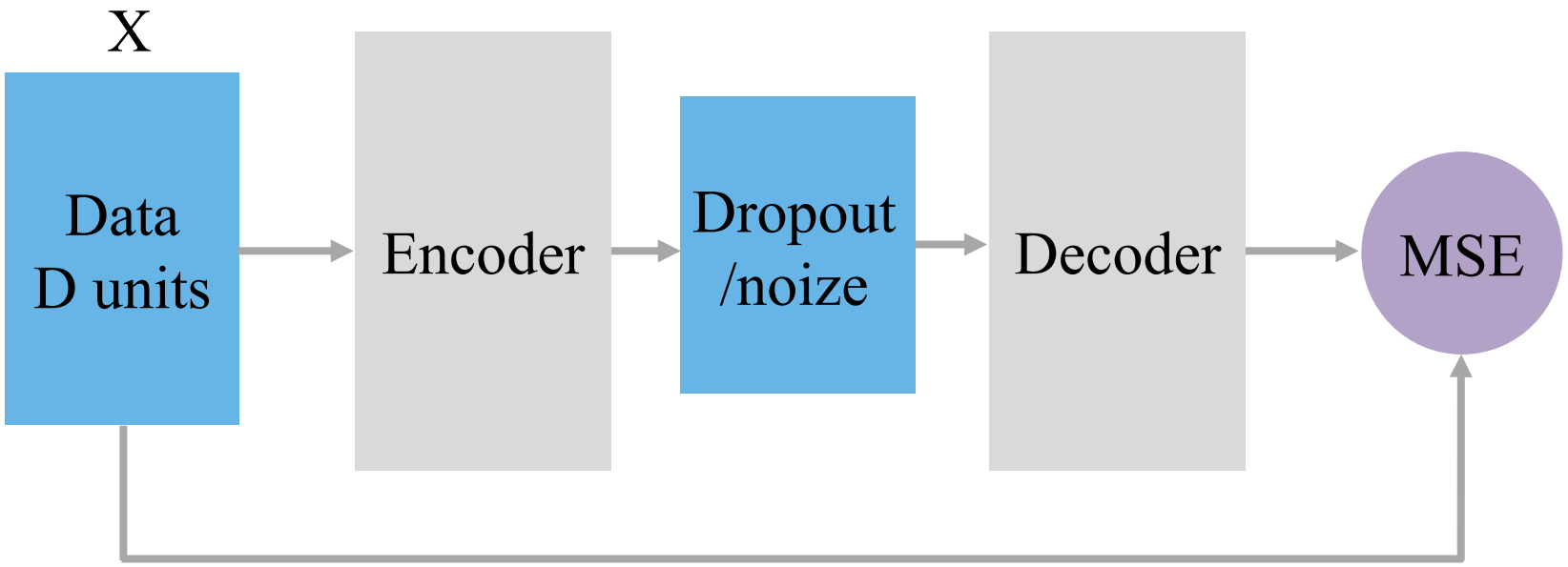


$$L = \left\| X - Dec\left(Enc(X)\right) \right\| + \sum_i |Enc_i(X)|$$

Redundant autoencoder

Naive approach will learn identity function!

Idea 2: noize/dropout, redundant code

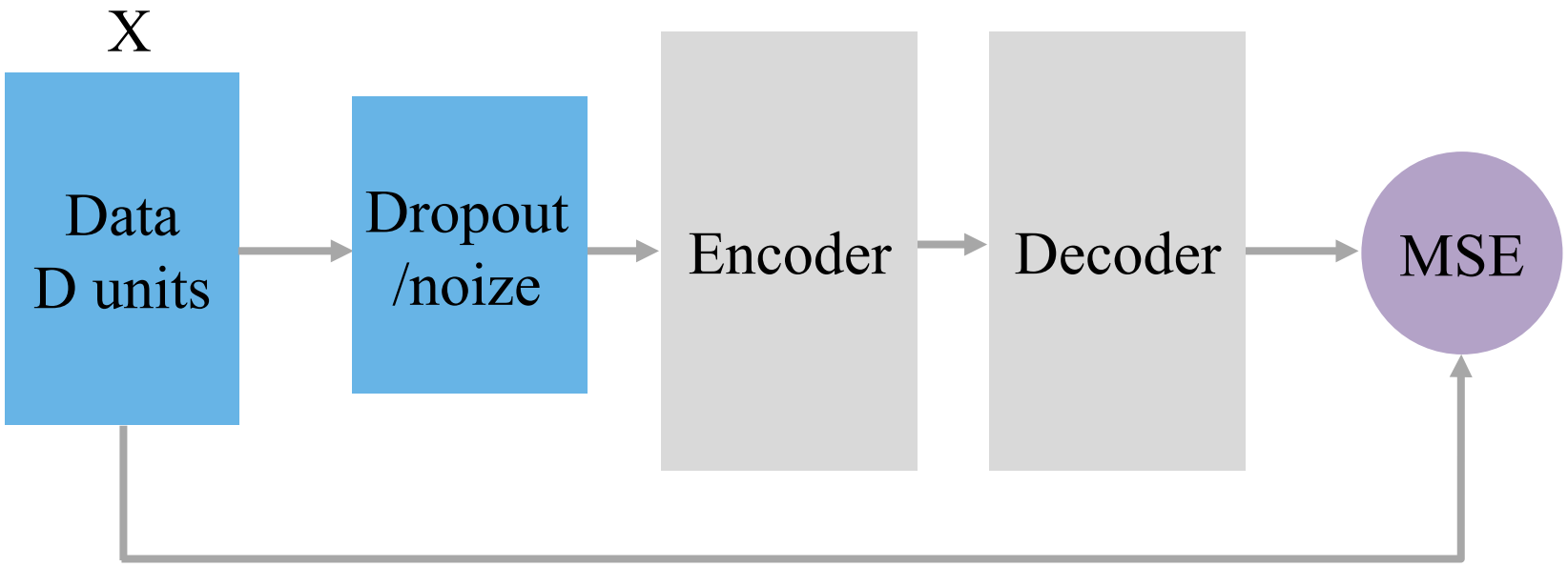


$$L = \left\| X - Enc\left(Dec\left(Noize(X)\right)\right) \right\|$$

Denoizing autoencoder

Naive approach will learn identity function!

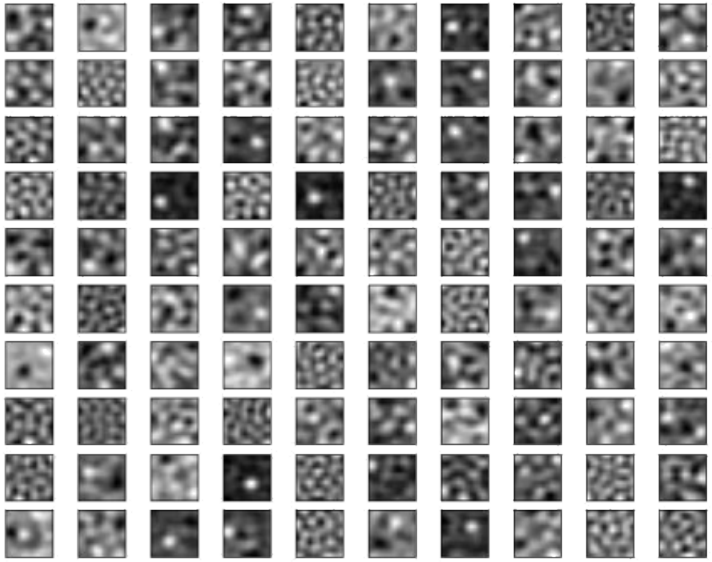
Idea 3: distort input, learn to fix distorsion



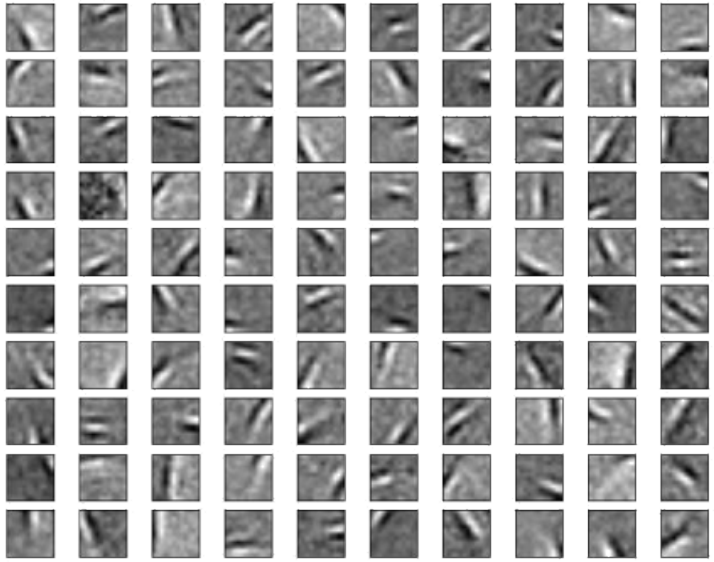
$$L = \left\| X - Enc\left(Dec\left(Noize(X)\right)\right) \right\|$$

Sparse Vs Denoizing

Filter weights, 12x12 patches



Sparse AE



Denoizing AE

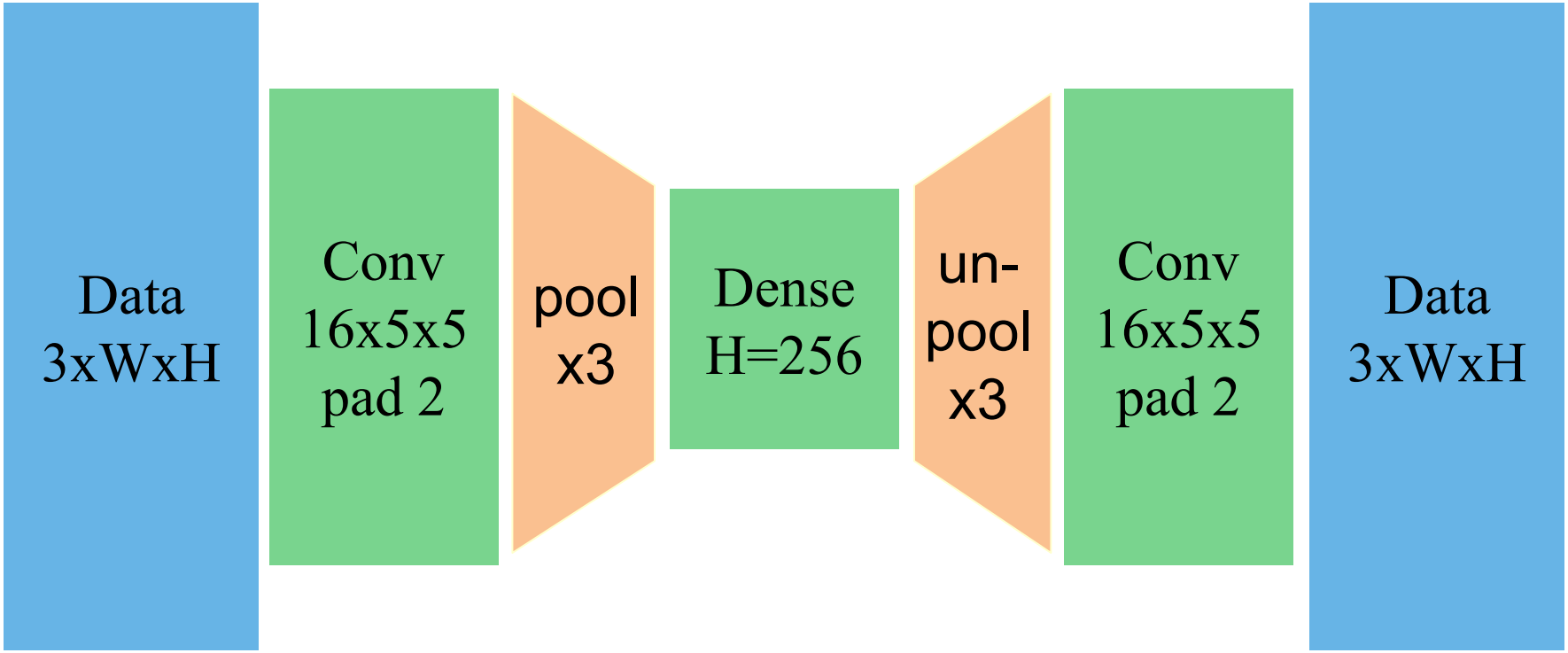
These images are actually clueless :)

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

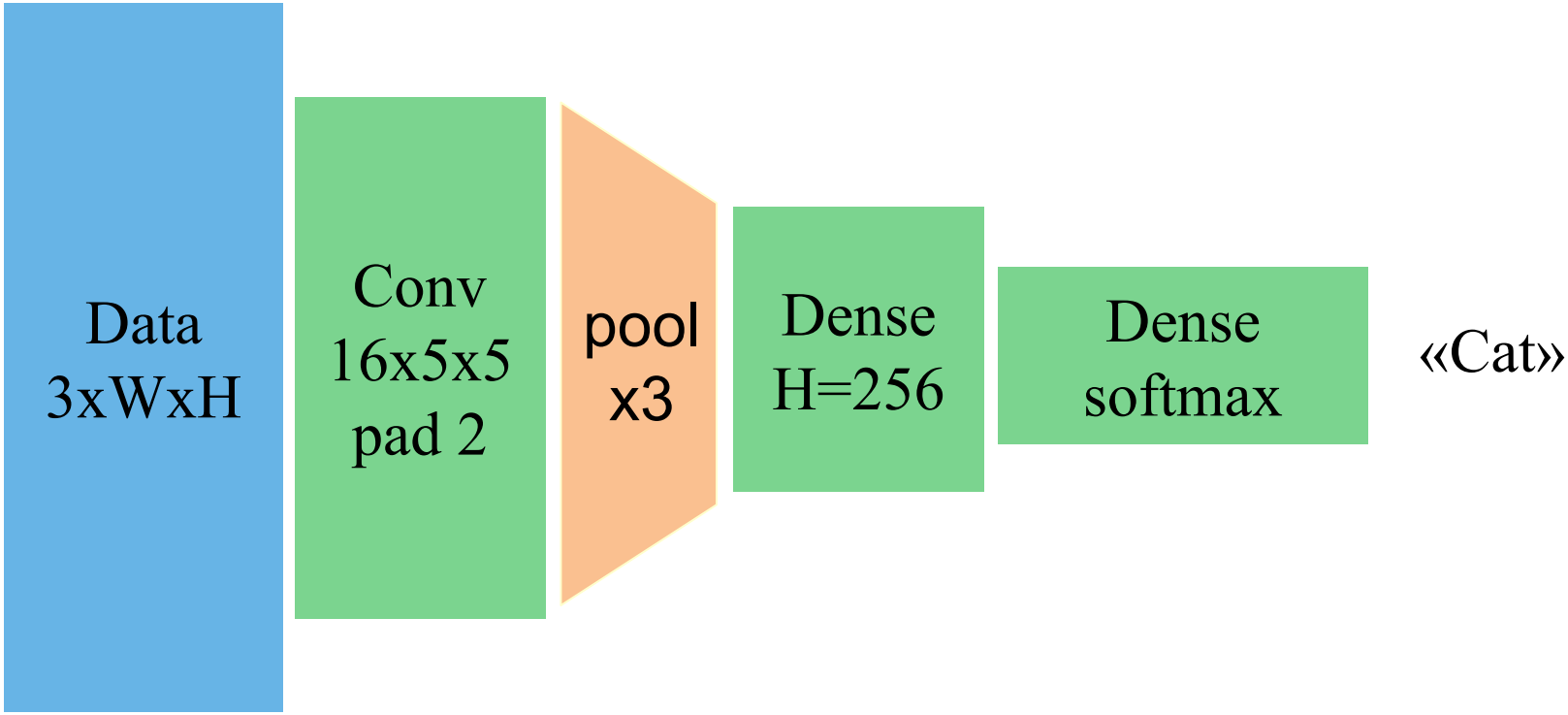
Pretraining

Use autoencoder as initialization



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