AUTOENCODERS

Lecture 4

MAL2, Spring 2025

AUTOENCODERS

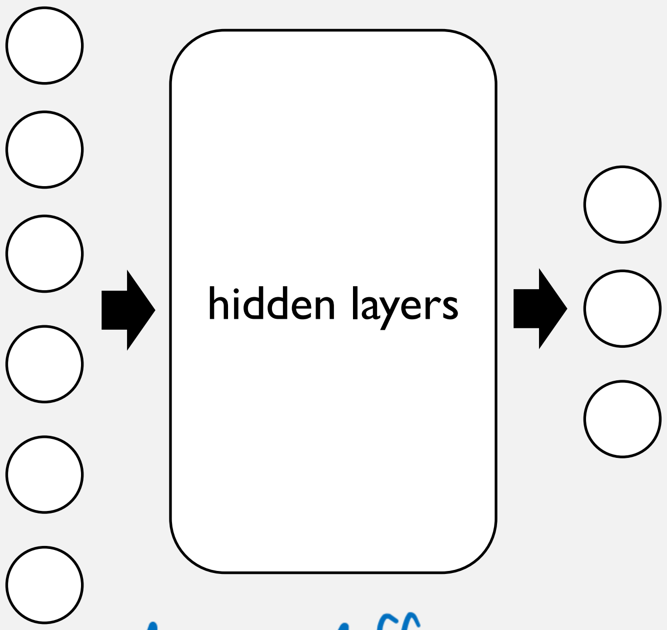
- What if Y was X?
- Dimensionality reduction
- Unsupervised pretraining
- Denoising images
- Colorization

WHAT IF Y WAS X?

Every neural network we have seen so far has looked like this:

input layer

#neurons= #features



output layer

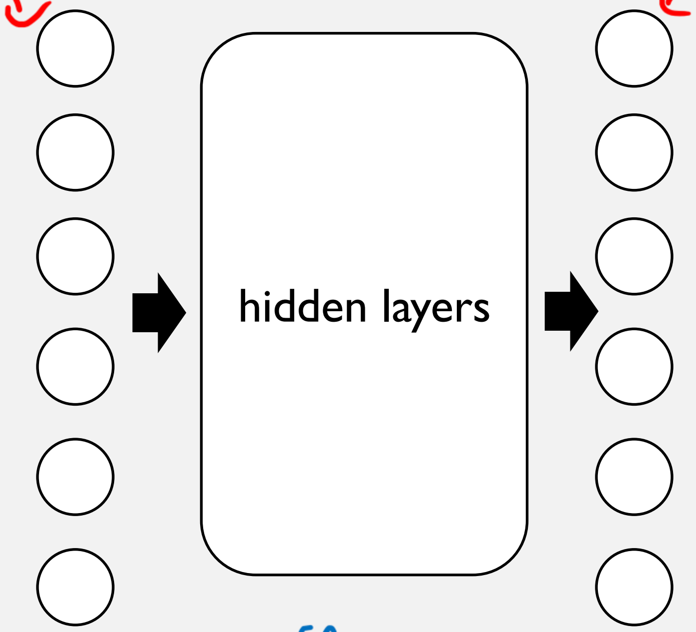
#neurons= #classes

loss = difference between X & y

What if we were crazy enough to make it look like this?

reconstruction

X

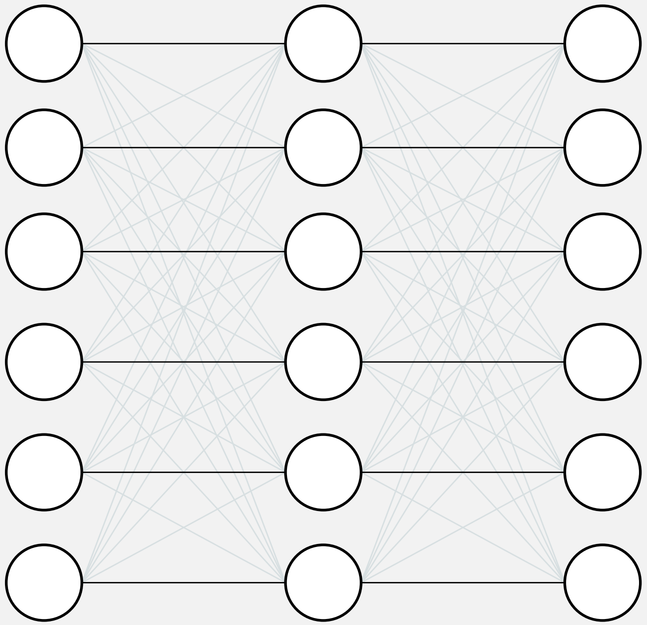


X again

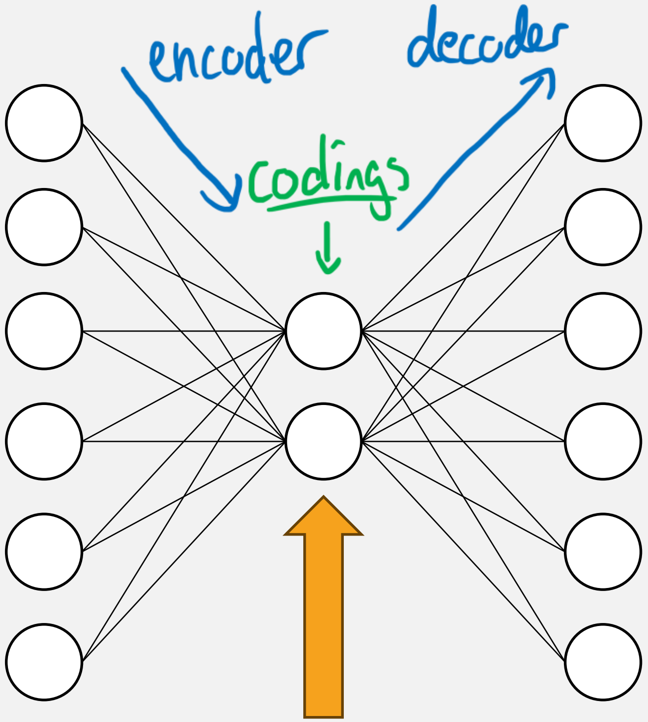
"reconstruction" loss=difference between X & X

BUT ISN'T THAT EASY?

Yes, but not if we constrain it!



AN UNDERCOMPLETE AUTOENCODER

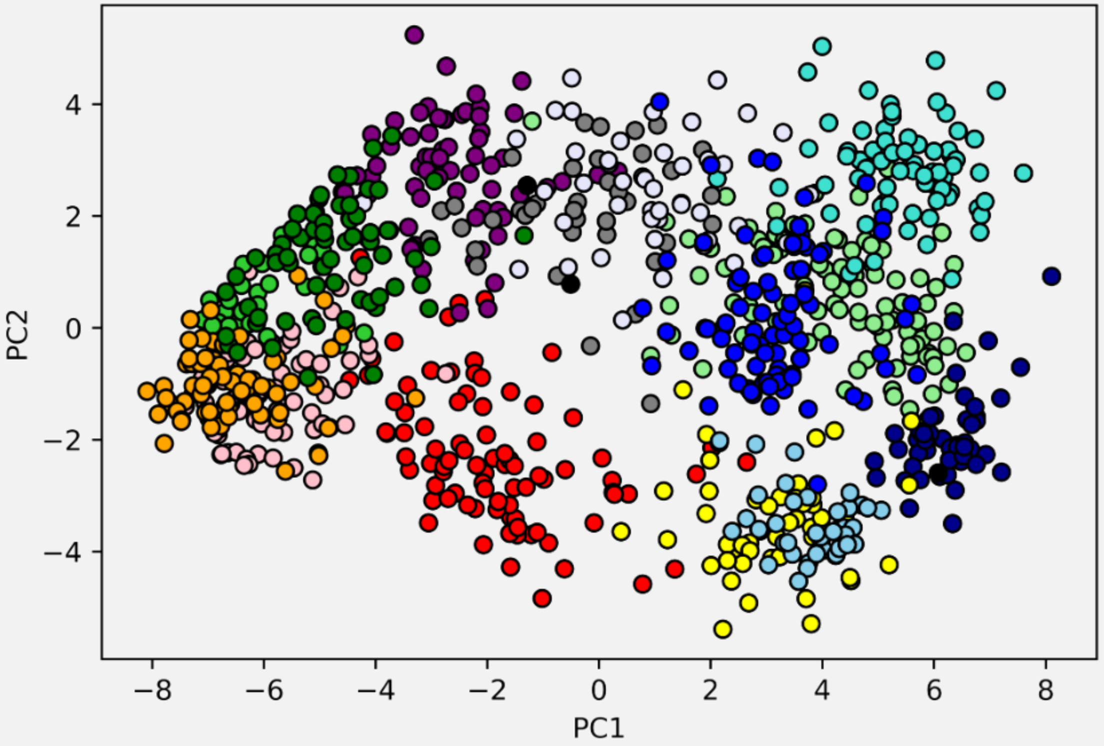


To reconstruct X, we need to express as much information as possible in these two neurons

AUTOENCODERS

- What if Y was X?
- Dimensionality reduction
- Unsupervised pretraining
- Denoising images
- Colorization

REMEMBER THIS?



PERFORMING PCA WITH AN AUTOENCODER

like PCA

linear

```
encoder = Sequential([Dense(2)])
```

encoder w/ 2 neurons, no activation

```
decoder = Sequential([Dense(49)])
```

#questions

```
autoencoder = Sequential([encoder, decoder])
```

construct full NN

```
optimizer = SGD(learning_rate=0.5)
```

it will be fast, so GD ok

```
autoencoder.compile(loss="mse", optimizer=optimizer)
```

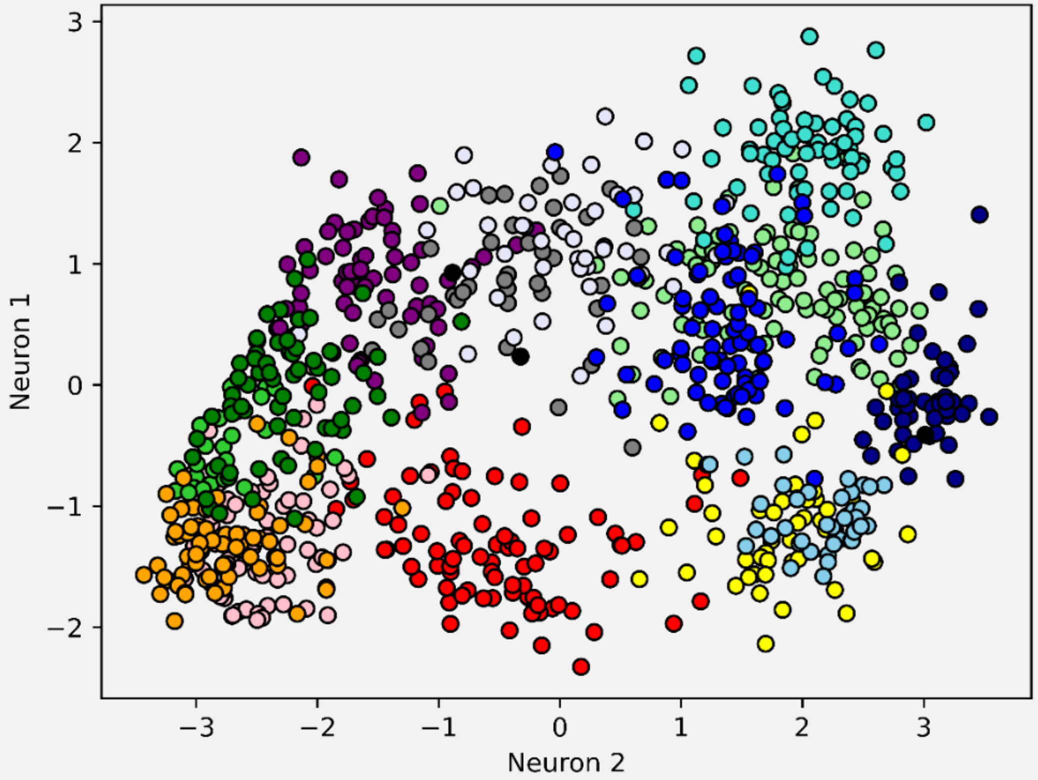
reconstruction loss $(X-\hat{X})^2$

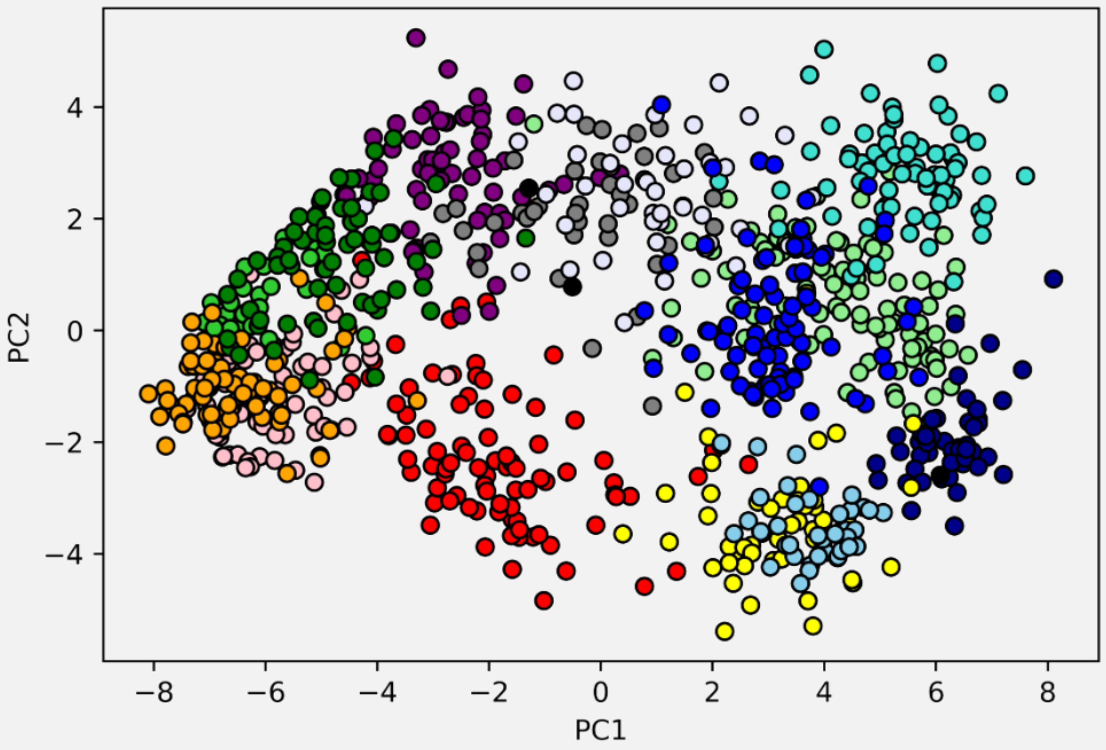
long time

```
history = autoencoder.fit(X, X, epochs=500, verbose = False)
```

```
codings = encoder.predict(X)
```

AUTOENCODERS VS PCA

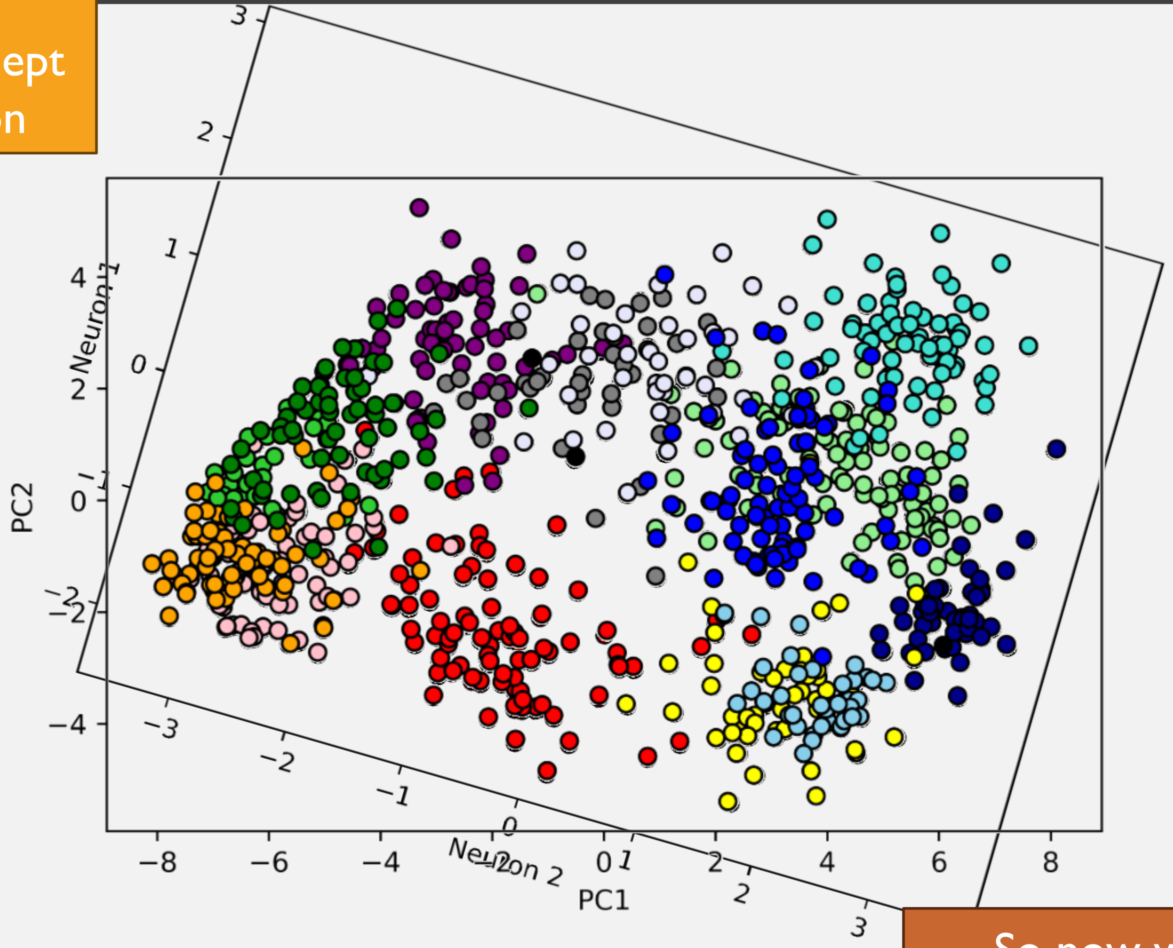




AUTOENCODERS VS PCA

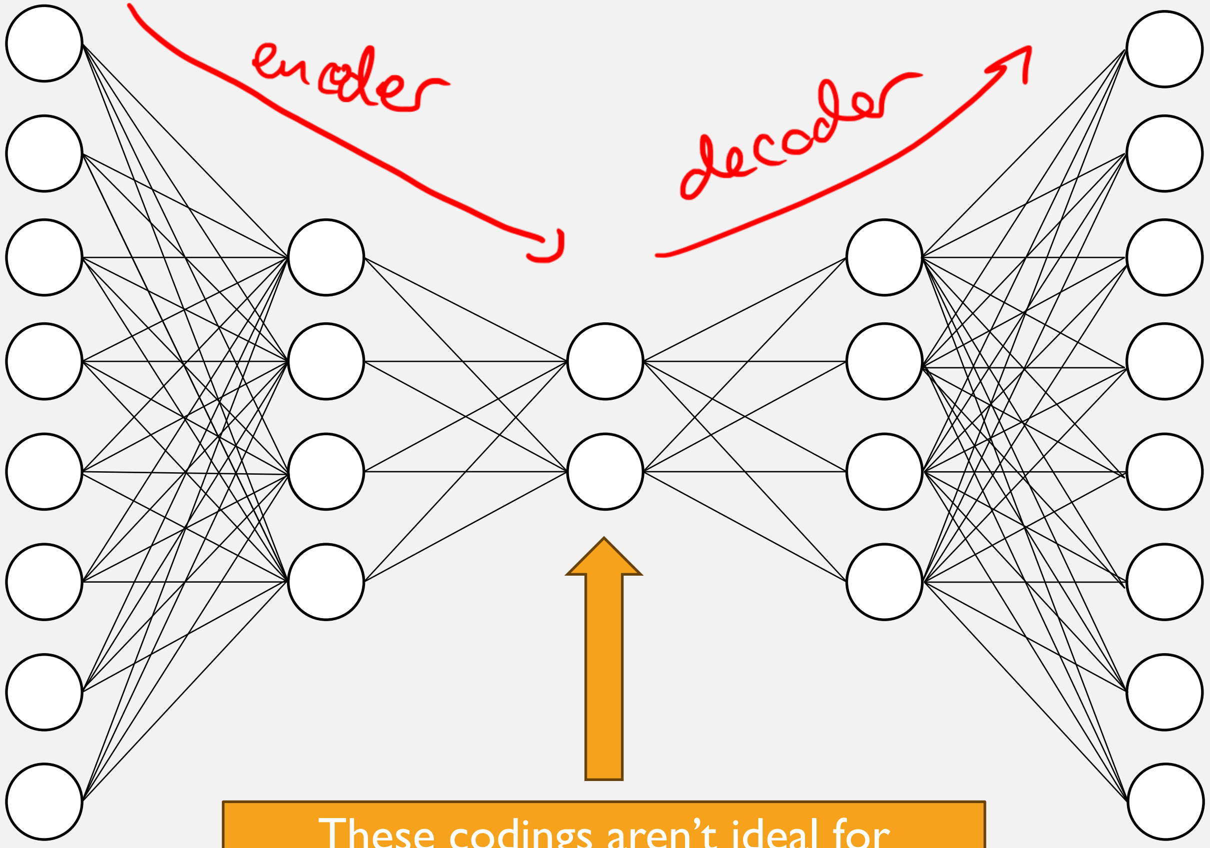
An autoencoder with linear activation is essentially PCA except for a scale factor and a rotation

linear algebra:
they span
the same
subspace



So now we can do PCA in an overcomplicated way … why care?

DEEP AUTOENCODERS



These codings aren't ideal for visualization as the transformation to the reconstructions is complex

symmetrical
non-linear
NN's capable
of learning
complex
codings

AUTOENCODERS

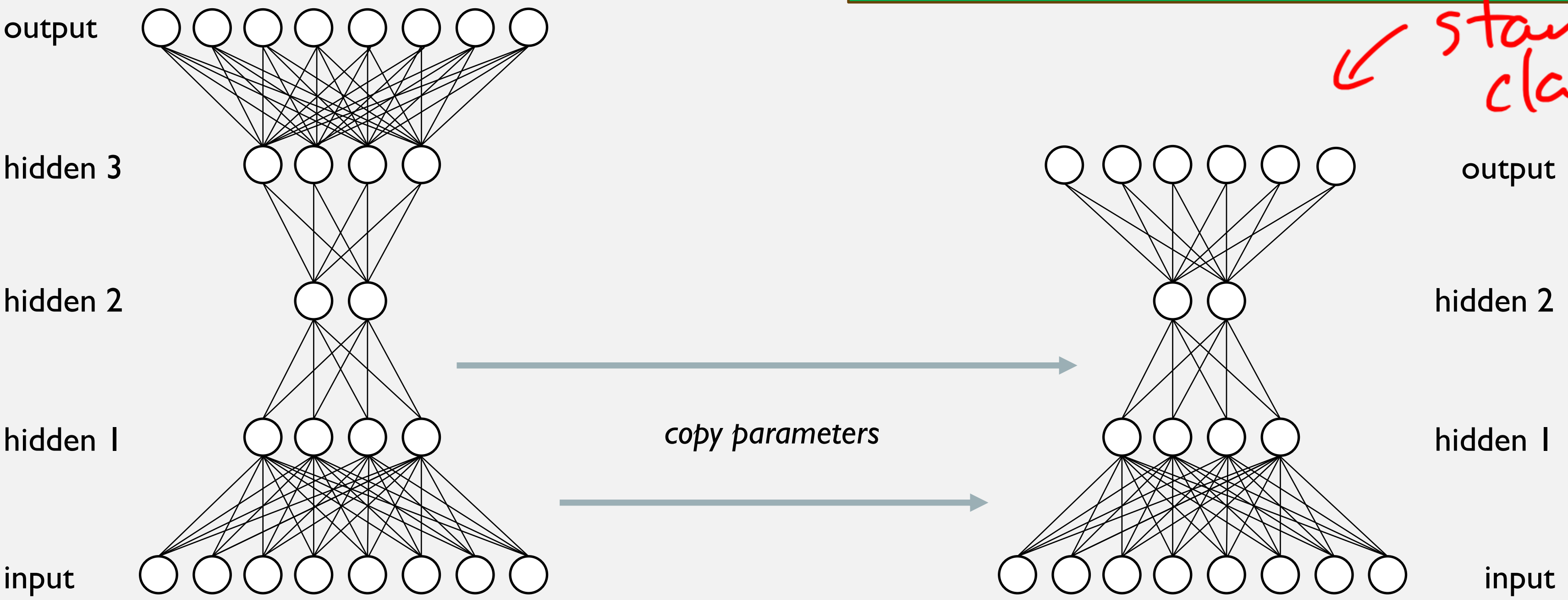
- What if Y was X?
- Dimensionality reduction
- Unsupervised pretraining
- Denoising images
- Colorization

UNSUPERVISED PRETRAINING

Sometimes, you have a lot of data, but only a fraction of it is labeled

It works because the classifier won't have to learn all the low-level features

standard classifier



① Train AE on all the data

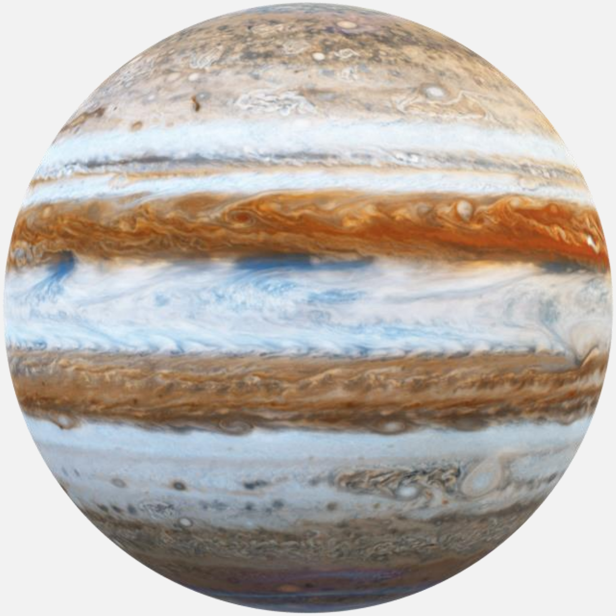
② Train classifier on the labeled data

AUTOENCODERS

- What if Y was X?
- Dimensionality reduction
- Unsupervised pretraining
- Denoising images
- Colorization

DENOISING

Add noise to the input and try to reconstruct the noise-free image



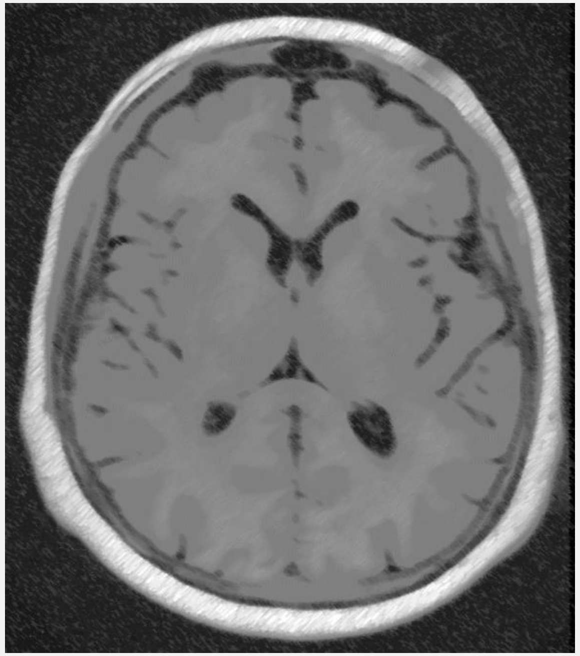
DENOISING

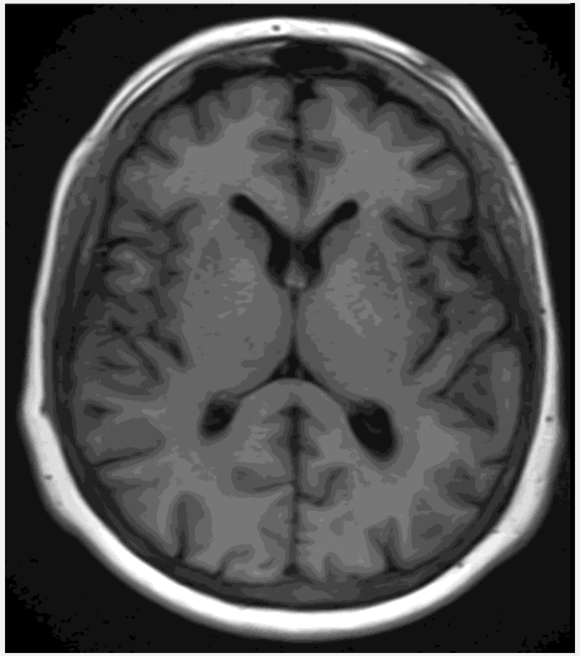
Take the autoencoder we just made

and experiment with the constraint

You have 15 minutes

AN INTERESTING APPLICATION





train AE to turn low-res into high-res
⇒ high-res w/ low rad. dose

low-radiation dose image of brain

high-radiation image

AUTOENCODERS

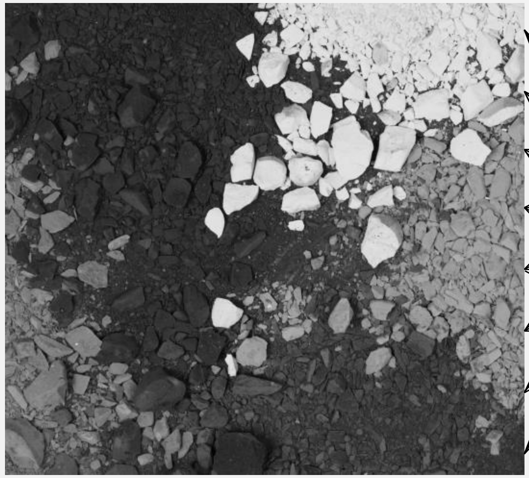
- What if Y was X?
- Dimensionality reduction
- Unsupervised pretraining
- Denoising images
- Colorization

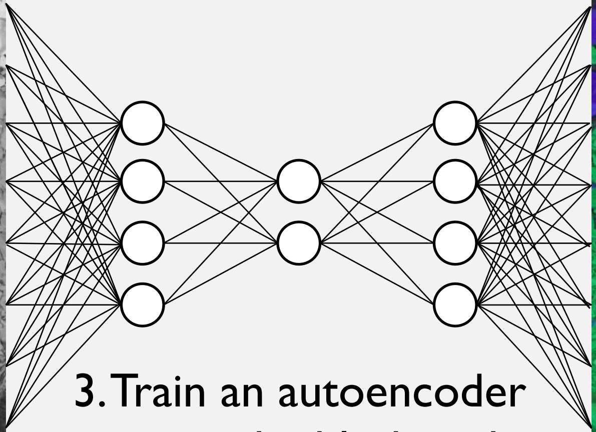
Great idea for a final project!

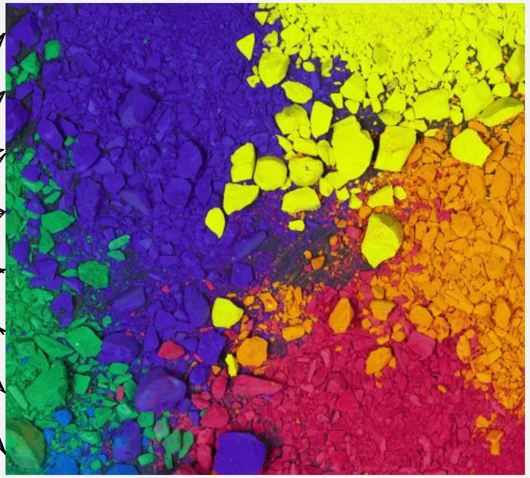
COLORIZATION

2. Make a copy of the dataset and turn each picture black-and-white.

1. Get a dataset of color pictures.







3. Train an autoencoder to turn the black-and-white pictures into the colored ones.

```
conv_encoder = Sequential([
    Conv2D(…),
    MaxPool2D(…),
    …
])
```

```
conv_decoder = Sequential([
    Conv2D(…),
    UpSamling2D(…),
    …
])
```

Conv2DTranspose

YOUR TICKET OUT THE DOOR

Scan this QR code

and tell me about something
you are still unsure about