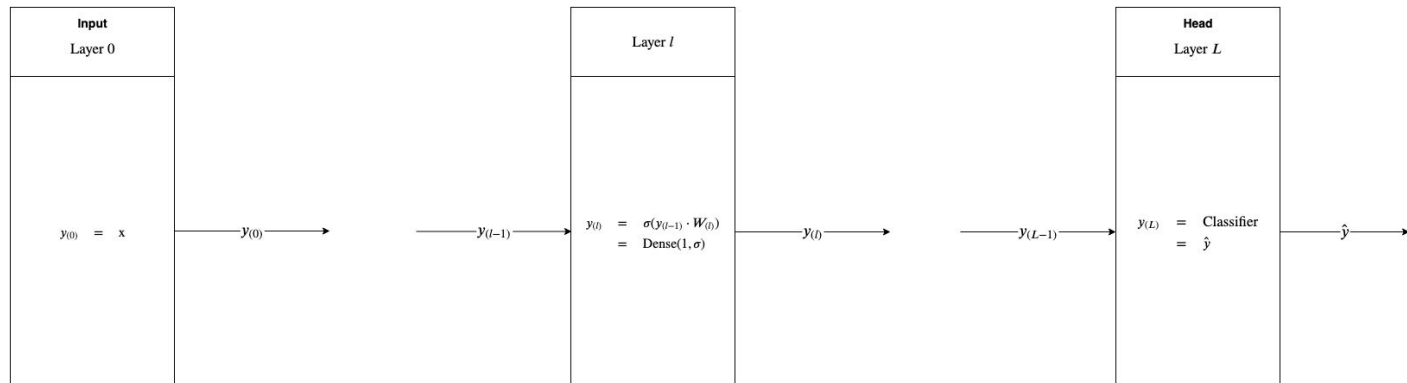


# Interpreting Representations: Preview

We have described an  $L$  layer (Sequential) Neural Network as

- a sequence of transformations of the input
  - each transformation a *layer*  $l$ , producing a new *representation*  $\mathbf{y}_{(l)}$
- that feed the final representation  $\mathbf{y}_{(L)}$  to a *head* (classifier, regressor)

# Layers



Is it possible to *interpret* each representation  $\mathbf{y}_{(l)}$  ?

- What do the new "synthetic features" mean ?
- Is there some structure among the new features ?
  - e.g., does each feature encode a "concept"

We will briefly introduce the topic of Interpretation.

A deeper dive will be the subject of a later lecture.

Our goal, for the moment, is to motivate Autoencoders.

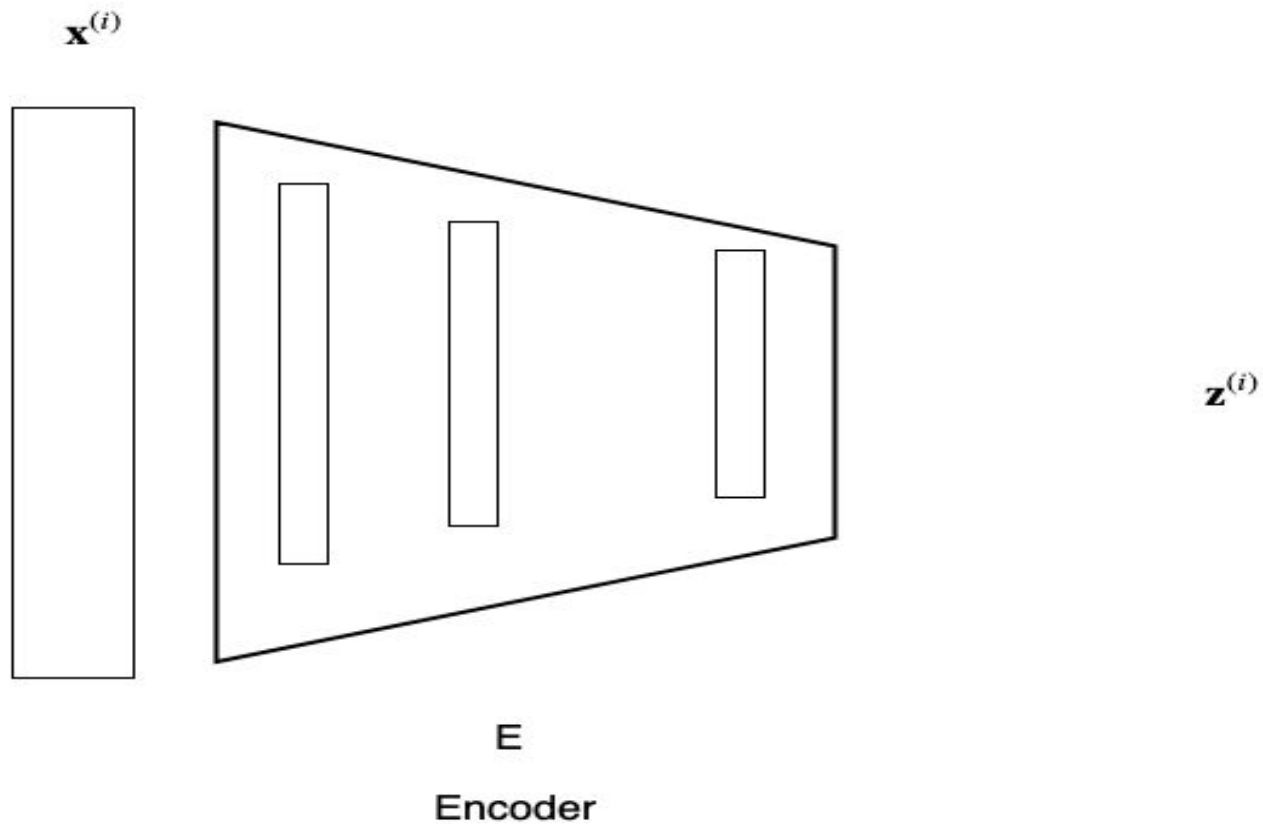
# Interpretation 1: Clustering of examples

One way to try to interpret  $\mathbf{y}_{(l)}$  is relative to a dataset  $\langle \mathbf{X}, \mathbf{y} \rangle = [ \langle \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \rangle \mid 1 \leq i \leq m ]$

- Compute  $\mathbf{y}_{(l)}^{(i)}$  by presenting  $\mathbf{x}^{(i)}$  to the NN
- Create a scatter plot (of dimension  $n_{(l)} = |\mathbf{y}_{(l)}|$ )
  - locate  $\mathbf{y}_{(l)}^{(i)}$  in the  $n_{(l)}$ -dimensional plot
  - label it with it's label  $\mathbf{y}^{(i)}$

## Mapping inputs to activations

---



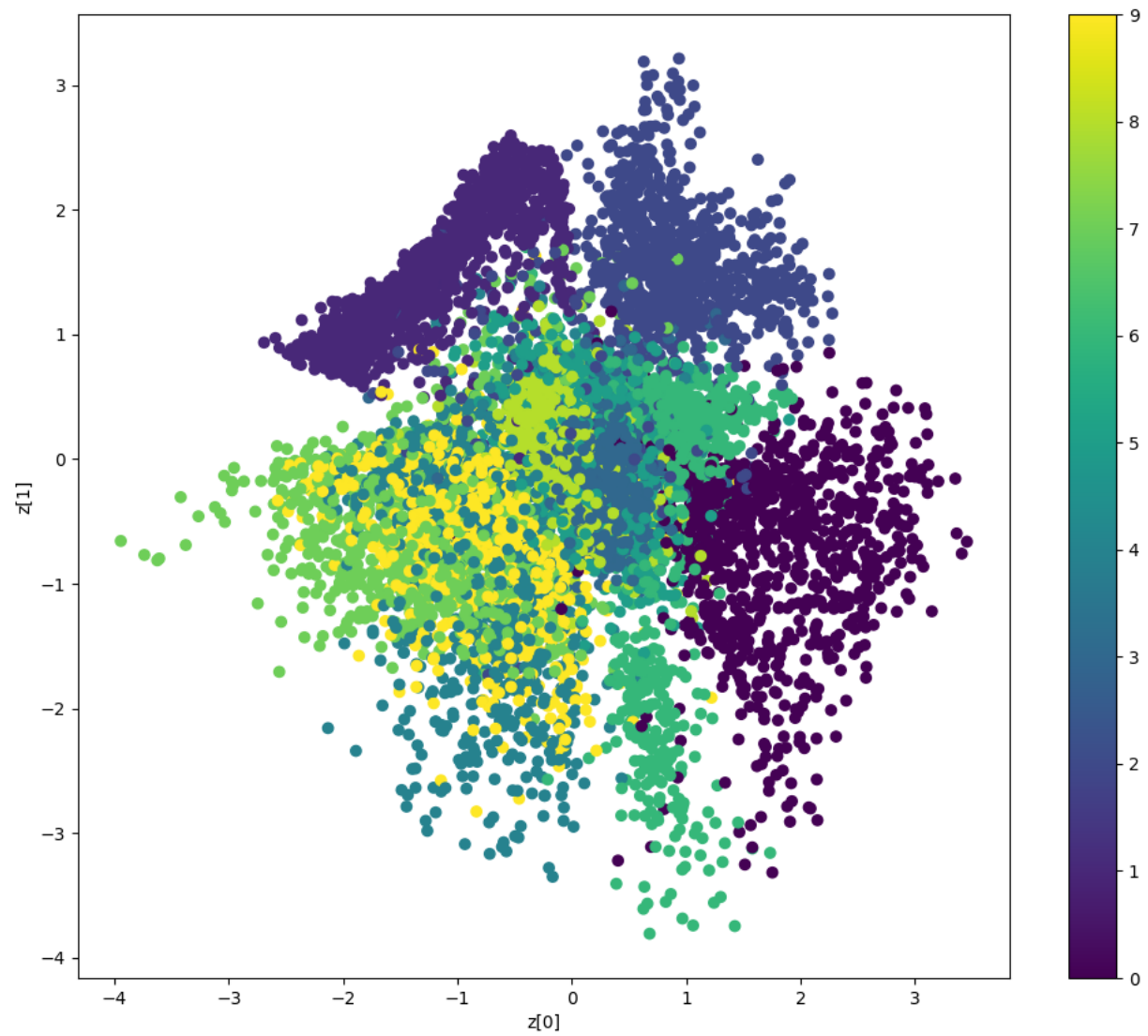
Do examples with identical labels form recognizable clusters ?

If so, perhaps we can interpret synthetic feature  $\mathbf{y}_{(l),j}$

- according to how variation in  $\mathbf{y}_{(l),j}$  affects the set of examples  $\mathbf{X}$

MNIST clustering produced by a VAE





- Each point is an example  $\mathbf{x}^{(i)}$
- The color corresponds to the label  $y^{(i)}$
- Axes are the first two synthetic features

You can see that some digits form tight clusters.

By understanding

- the clusters
- how the digit label's vary as a synthetic feature varies

we might be able to infer meaning to the synthetic features.

The first two synthetic features may correspond to properties of those digits

- digits with "tops"
- digits with "curves"

## **Note**

This is not too different from trying to interpret Principal Components:

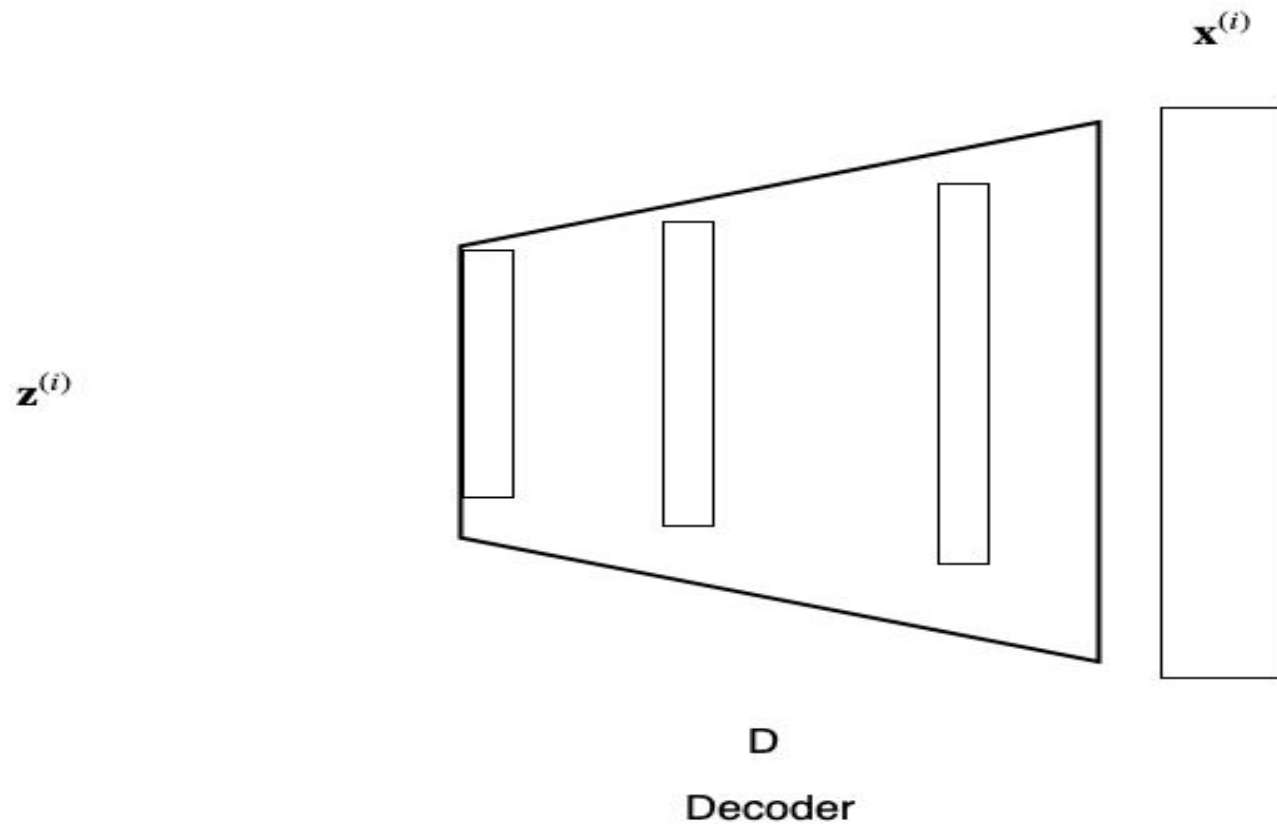
# Interpretation 2: Examining the latent space

Another method

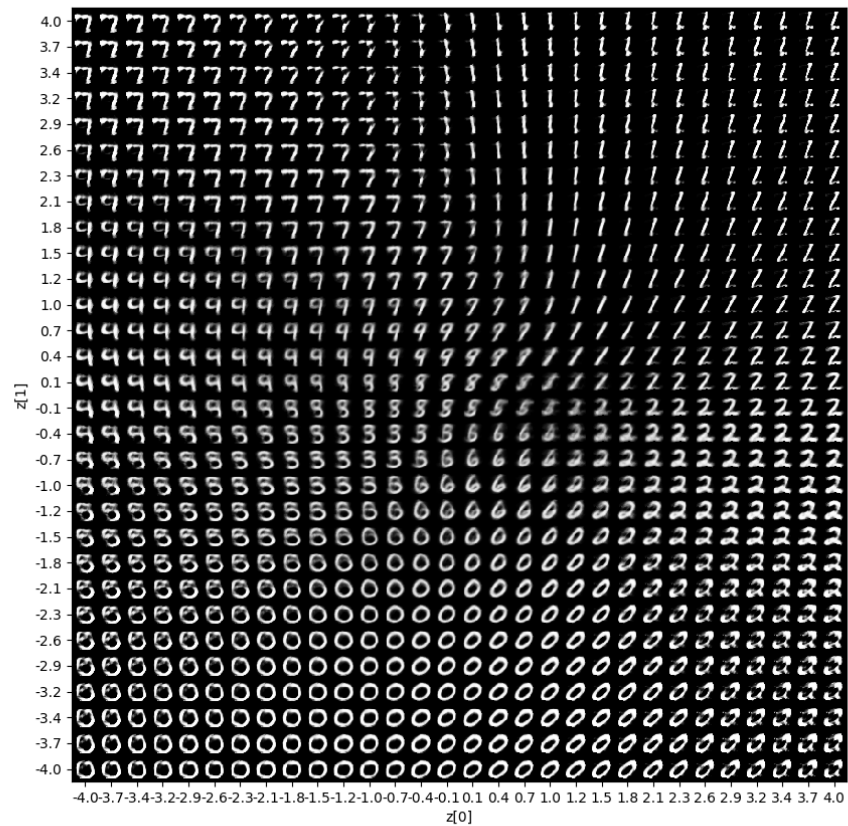
- Create an  $n_{(l)}$  dimensional grid of evenly spaced values of  $\mathbf{y}_{(l)}$
- Let  $\mathbf{y}_{(l)}^{(i')}$  be such a value
  - **Note** this is **not necessarily** an example produced from an  $\mathbf{x} \in \mathbf{X}$
- Map  $\mathbf{y}_{(l)}^{(i')}$  to some value in the input representation  $\mathbf{x}^{(i')}$ 
  - **Note** this is **not necessarily** an example from  $\mathbf{X}$
  - But presenting  $\mathbf{x}^{(i')}$  to the NN results in  $\mathbf{y}_{(l)} = \mathbf{y}_{(l)}^{(i')}$

Invert activations

---



# MNIST clustering produced by a VAE



- Axes are the first two synthetic features
- For  $\mathbf{y}_{(l)}$  at a given grid point:
  - find a value in the input representation that maps to this grid point



Note that there is *no reason* to expect that the inversion of an arbitrary representation *looks like* a digit

- it merely has the correct shape
- unless we impose some constraints

This is **not** just a different view of the first plot:

- we are able to infer a pseudo-input for a grid point  $\mathbf{y}_{(l)}^{(i')}$  that corresponds to **no actual input in  $\mathbf{X}$** 
  - For example, we infer a digit from an uninhabited region of the grid of the first plot

Some observations (with possible interpretation)

- Does the first synthetic feature control slant ?
  - Examine 0's along bottom row
- Does the second synthetic feature control "curviness" ?
  - Examine the 2's column at the edge, from bottom to top

In order for this method to work, we must be able to *invert*  $\mathbf{y}_{(l)}$ .

We will show how to do this in a later lecture.

## Deja vu: have we seen this before ?

These two methods of interpretation have been encountered in an earlier lecture

- mapping original features  $\mathbf{x}^{(i)}$  to synthetic features  $\tilde{\mathbf{x}}^{(i)}$
- inverting synthetic feature  $\tilde{\mathbf{x}}^{(i)}$  to obtain original feature  $\mathbf{x}^{(i)}$

Principal Component Analysis (PCA) !

PCA is an Unsupervised Learning task that can be used for

- dimensionality reduction
- clustering

The key to its interpretability was the simplicity of transforming and inverting

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad \text{SVD decomposition of } \mathbf{X}$$

$$\tilde{\mathbf{X}} = \mathbf{X}\mathbf{V} \quad \text{transformation to synthetic features}$$

$$\mathbf{X} = \tilde{\mathbf{X}}\mathbf{V}^T \quad \text{inverse transformation to original features}$$

The transformation  $V$  via matrix multiplication is *linear*.

We will explore *non-linear, invertible* transformations during our study of Autoencoders.



In [4]: `print("Done")`

Done