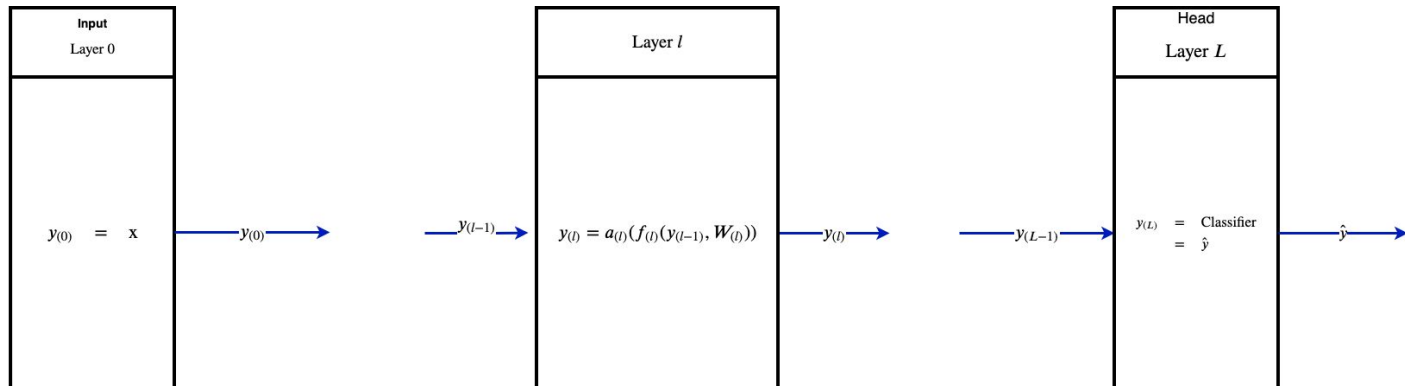


# Interpreting Representations: Preview

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

- a sequence of transformations of the input
  - each transformation a *layer*  $1 \leq l \leq (L - 1)$ , producing a new *representation*  $\mathbf{y}_{(l)}$
- that feed the final representation  $\mathbf{y}_{(L-1)}$  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

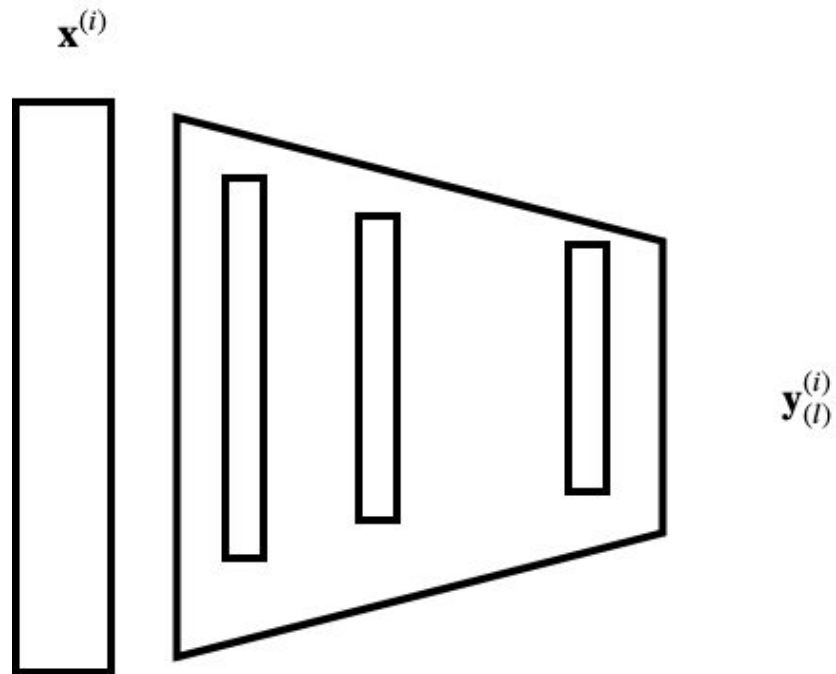
$$\langle \mathbf{X}, \mathbf{y} \rangle = \{\mathbf{x}^{(i)}, y^{(i)} \mid 1 \leq i \leq m\}$$

By passing each example  $\mathbf{x}^{(i)}$  through the layers to obtain  $\mathbf{y}_{(l)}^{(i)}$  we create a mapping from examples to layer  $l$  representations

$$\langle \mathbf{X}, \mathbf{y}_{(l)} \rangle = \{\mathbf{x}^{(i)}, \mathbf{y}_{(l)}^{(i)} \mid 1 \leq i \leq m\}$$

# Mapping inputs to layer l representations

---



Let's create a scatter plot of each example's representation  $\mathbf{y}_{(l)}^{(i)}$

- In  $n_{(l)}$ -dimensional space
- Labelling each point
- With the target  $\mathbf{y}_{(l)}$
- Or with a set of input attributes, e.g.,  $(\mathbf{x}_j^{(i)}, \mathbf{x}_{j'}^{(i)})$

Perhaps clusters of examples will appear.

If all points in the cluster have the same label

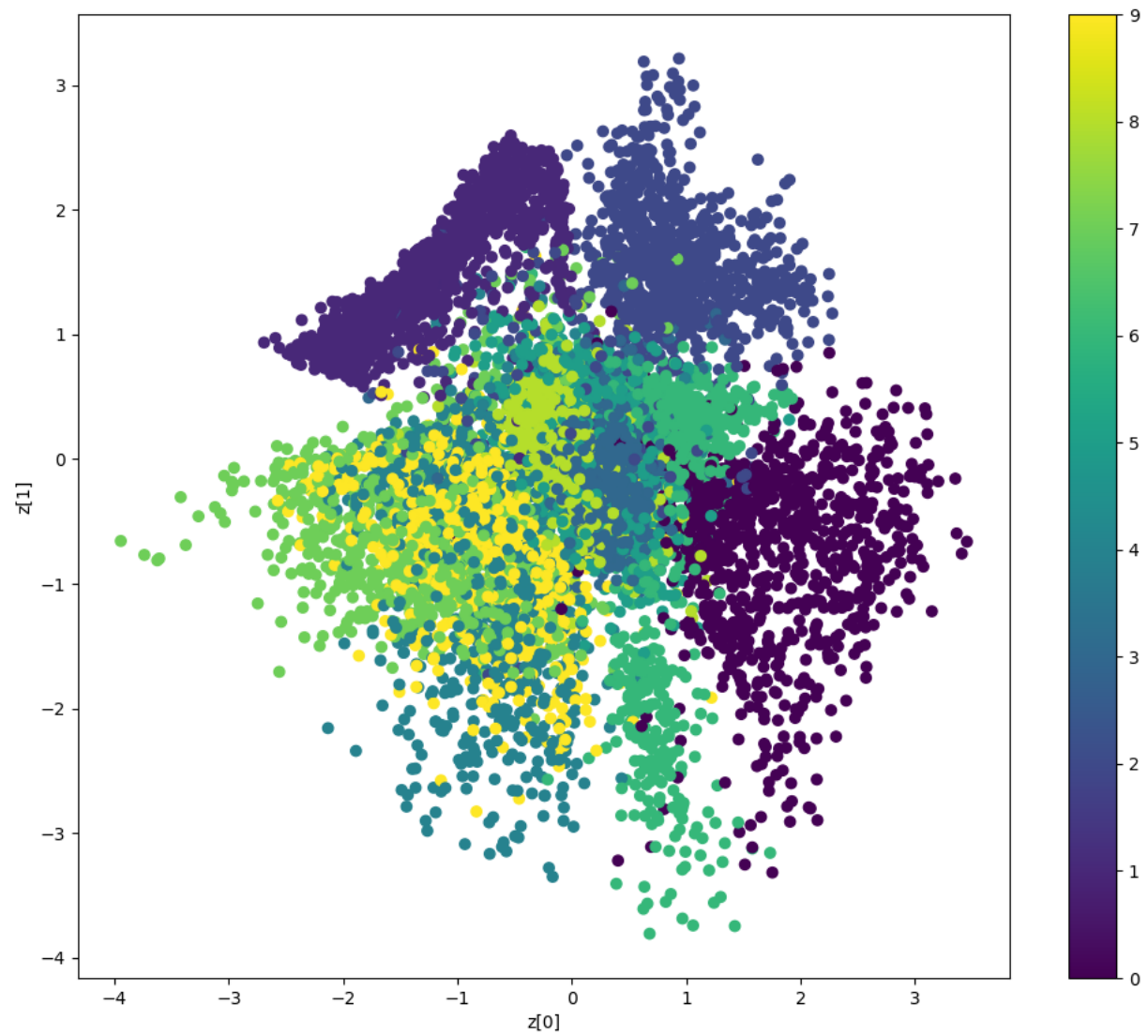
- We might be able to identify the representation with a target or set of input features



Here is an example of the representation of the MNIST digits in an intermediate layer of a particular network

- The output of the Encoder half of an Autoencoder
- Which we will study in a subsequent lecture

MNIST clustering produced by a VAE



- Each point is an example  $\mathbf{x}^{(i)}$
- With coordinates chosen from two of the synthetic features in  $\mathbf{y}_{(l)}$
- The color corresponds to the label  $\mathbf{y}^{(i)}$  (i.e., the digit that is represented by the image)

You can see that some digits form tight clusters.

By understanding

- The commonality of examples within a cluster
- How the digit label's vary as a synthetic feature varies

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

The first two synthetic features in  $\mathbf{y}_{(l)}$  of MNIST 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

Suppose we could *invert* the representation  $\mathbf{y}_{(l)}$  to obtain a value  $\mathbf{x}$  that lies in the input domain.

Then

- By perturbing individual synthetic features  $\mathbf{y}_{(l),j}$  in a given representation  $\mathbf{y}_{(l)}$  to obtain  $\mathbf{y}'_{(l)}$
- And examining the effect on the inverted value  $\mathbf{x}'$
- We might be able to assign meaning to the layer  $l$  feature  $\mathbf{y}_{(l),j}$



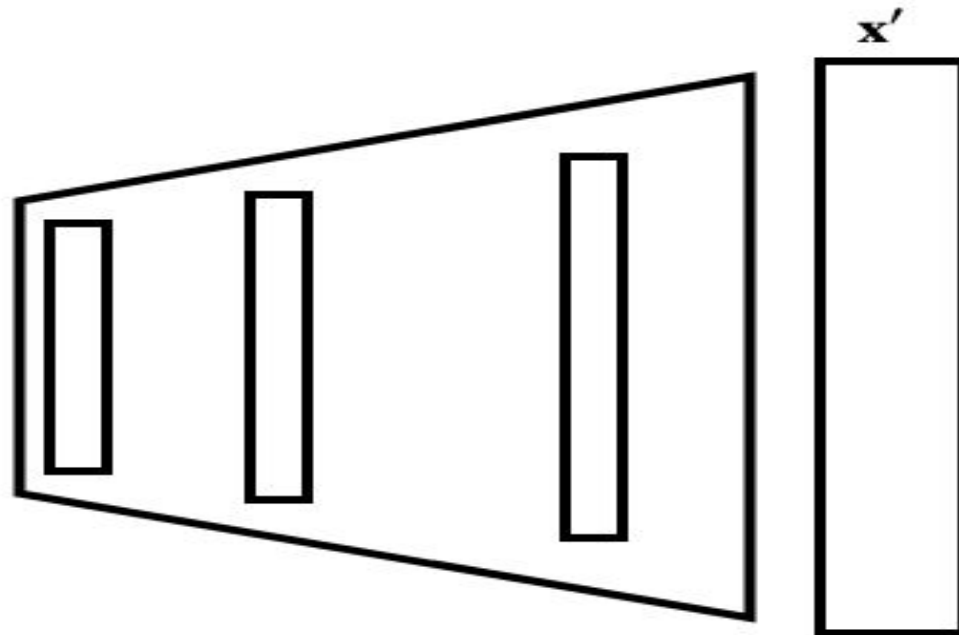
Note that the inversed value  $\mathbf{x}'$  **is not necessarily** (and probably not) a value in training set  $\mathbf{X}$  !

- It is merely a value obtained by the mathematical inversion of a function
- Especially since the perturbed  $\mathbf{y}'$  may not be the mapping of any example  $\mathbf{x}^{(i)} \in \mathbf{X}$

Invert layer l representation

---

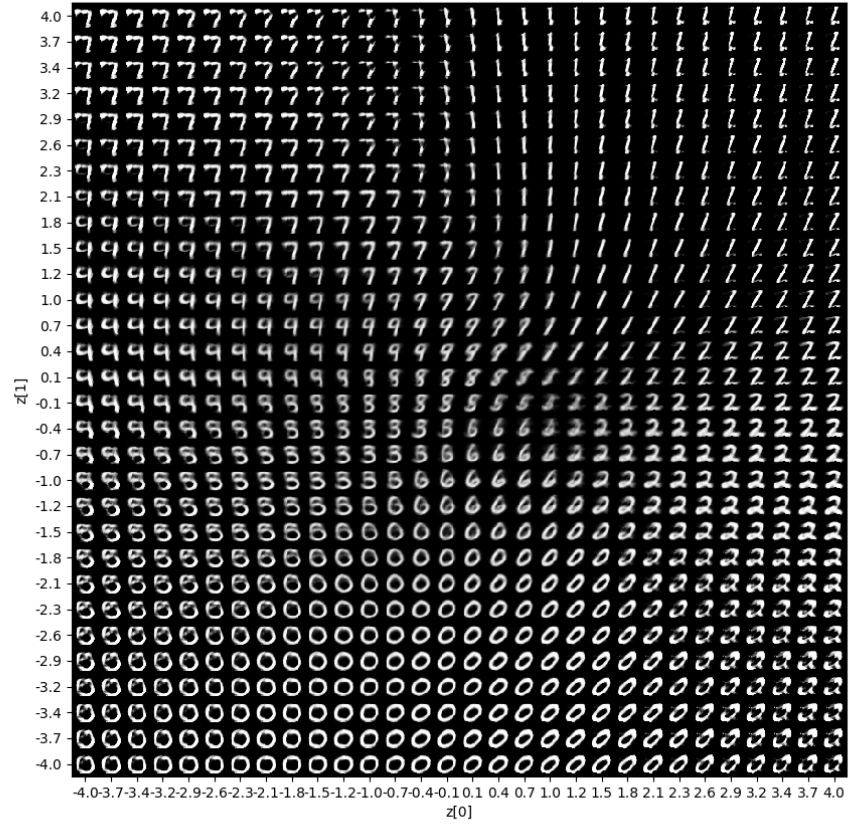
$\mathbf{y}'_{(l)}$



Here are the inverted images obtained by perturbing two synthetic features in  $\mathbf{y}_{(l)}$

- Horizontal axis perturbs one feature
- Vertical axis perturbs a second feature

## MNIST clustering produced by a VAE



Some observations (with possible interpretation)

- Does the synthetic feature on the horizontal axis control slant ?
  - Examine 0's along bottom row
- Does the synthetic feature on the vertical axis control "curviness" ?
  - Examine the 2's column at the right edge, from bottom to top

There is *no reason to expect* that the inversion of an arbitrary representation *looks like* a digit but it does !

Perhaps

- The mapping from inputs to representations is such that similar inputs have very similar representations
- Or we impose some constraints on the inversion to force the inverted value to look like a digit

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

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

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

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

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

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

# Conclusion

Neural Networks have the reputation of being magical but opaque.

We hope this brief introduction to interpretation provides some hope that we can understand their inner workings.

A separate lecture will explore this topic in greater depth.

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

Done