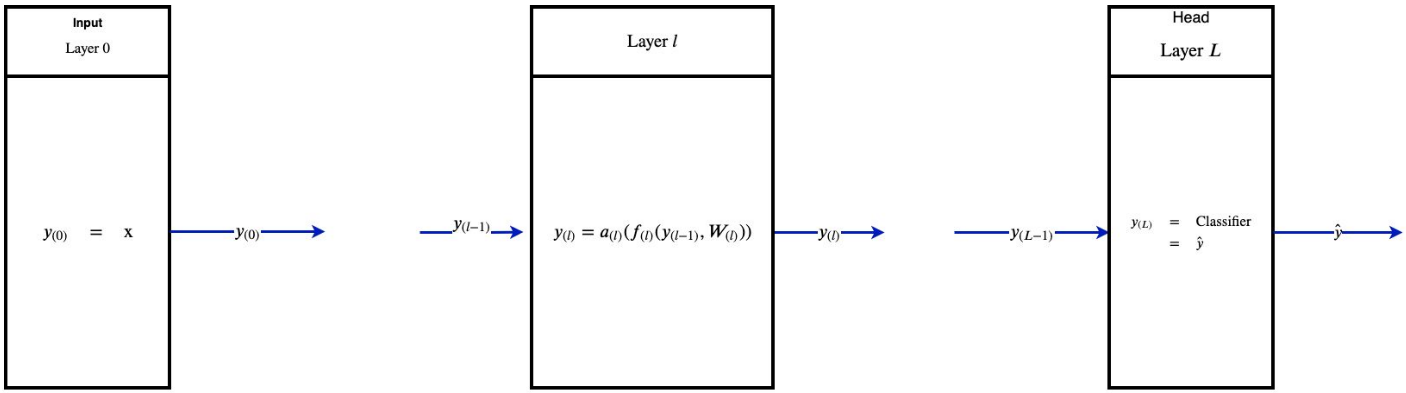
Interpretting Representations: Preview

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

- a sequence of tranformations of the input
    - each transformation a *layer* $1 \le l \le (L-1)$, producing a new *representation* $\mathbf{y}_{(l)}$
- that feed the final representation $\mathbf{y}_{(L-1)}$ to a *head* (classifier, regressor)

Layers


Input
Layer 0
$y_{(0)} = \mathrm{x}$
$y_{(0)}$
$y_{(l-1)}$
Layer l
$y_{(l)} = a_{(l)}(f_{(l)}(y_{(l-1)}, W_{(l)}))$
$y_{(l)}$
$y_{(L-1)}$
Head
Layer L
$y_{(L)}$ = Classifier
= $\hat{y}$
$\hat{y}$

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.

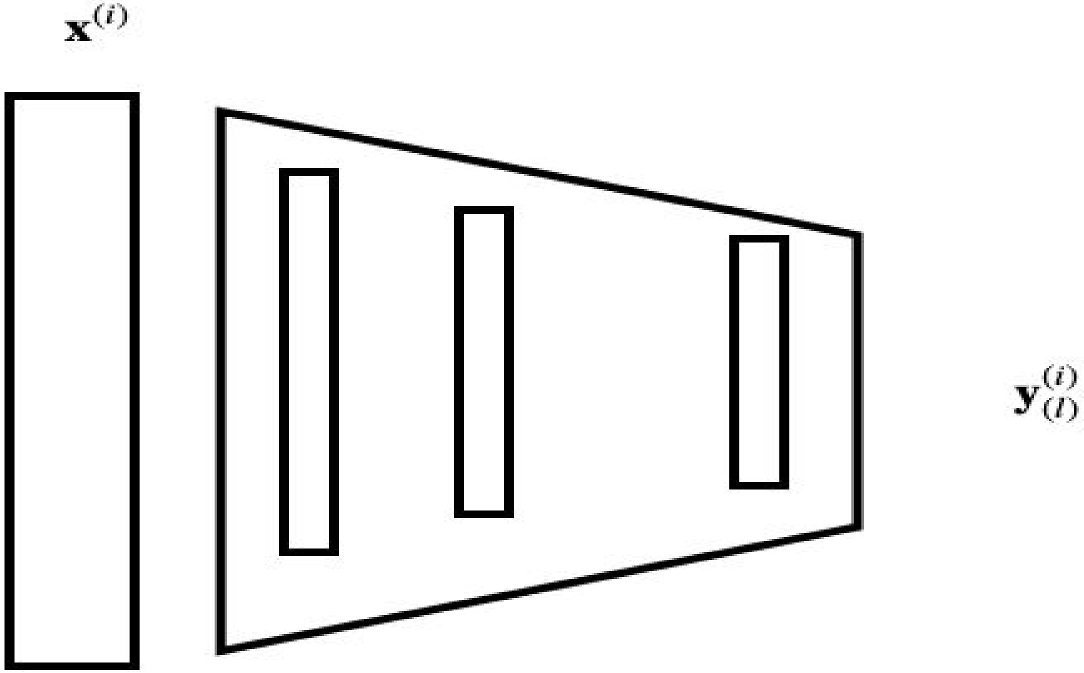
Interpretation 1: Clustering of examples

One way to try to interpet $\mathbf{y}_{(l)}$ is relative to a dataset
$\langle \mathbf{X}, \mathbf{y} \rangle = \{\mathbf{x}^{(\mathbf{i})}, y^{(\mathbf{i})} | 1 \le i \le m\}$

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

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

Mapping inputs to layer l representations



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

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

Perhaps clusters of examples will appear.

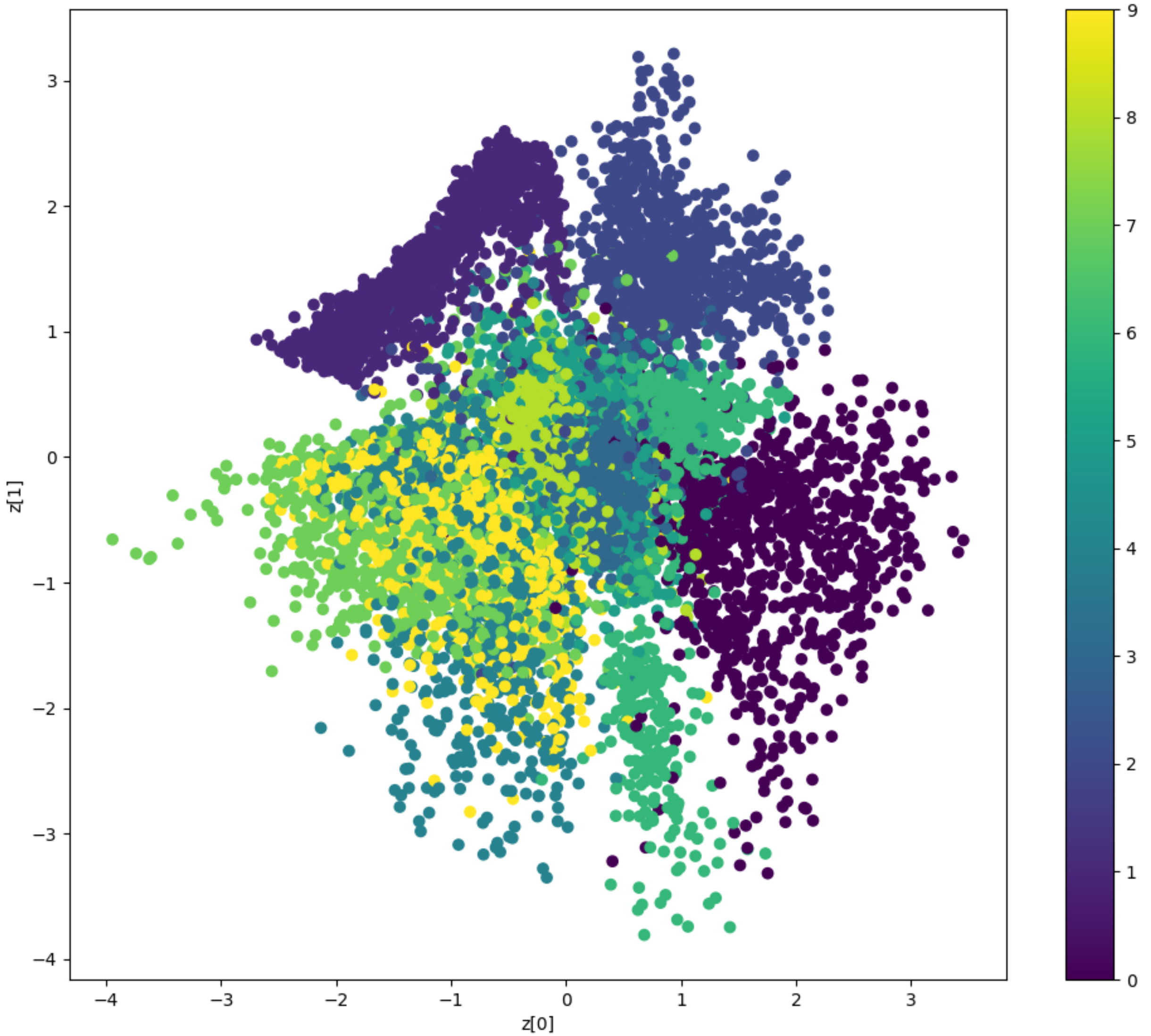
If all points in the cluster have the same label

- 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



3
2
1
0
−1
−2
−3
−4
z[1]
−4
−3
−2
−1
0
1
2
3
z[0]
9
8
7
6
5
4
3
2
1
0

- 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.

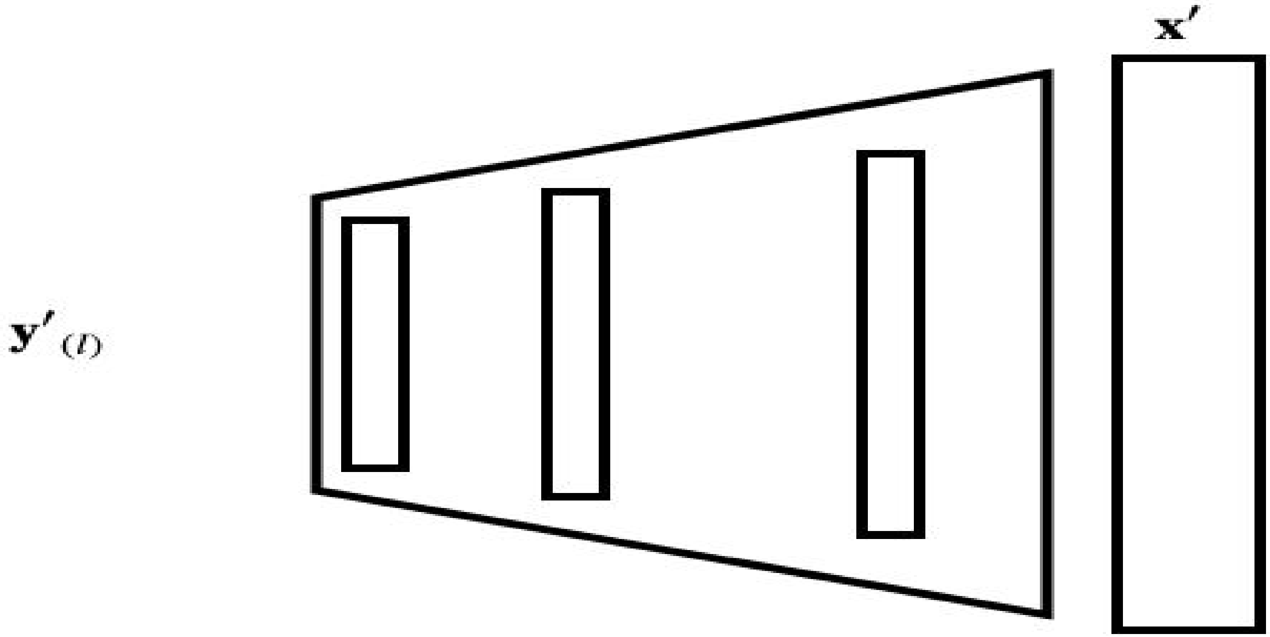
Then

- 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 $y_{(l),j}$

Note that the invered 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 $\mathbf{x^{(i)}} \in \mathbf{X}$

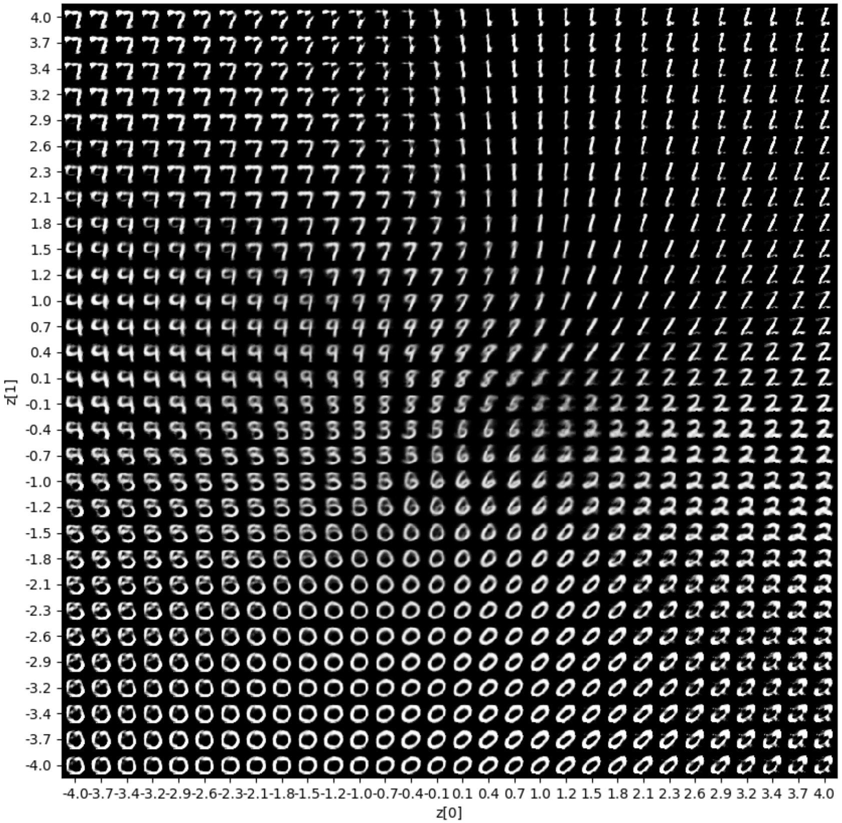
Invert layer l representation



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 intepretation)

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

Principal Component Analysis (PCA) !

PCA is an Unsupervised Learning task that can be used for

- dimensionality reduction
- clustering

The key to it's intepretability was the simplicity of transforming and inverting

$$\begin{aligned}
\mathbf{X} &= U\Sigma V^T && \text{SVD decomposition of } \mathbf{X} \\
\tilde{\mathbf{X}} &= \mathbf{X}V && \text{tranformation to synthetic features} \\
\mathbf{X} &= \tilde{\mathbf{X}}V^T && \text{inverse tranformation to original features}
\end{aligned}$$

The transformation V via matrix multiplication is *linear*.

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

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

```
Done
```