Using Principal Components to understand a Neural Network

Principal Components Analysis (PCA) is a *dimension reduction* technique

- Transforms an example with many, correlated features
- Into an example with fewer, independent features

A feature map of a Convolutional Neural Networks (CNN) is big

- Single feature in a map
- But at *many* spatial locations
- Which may be highly correlated

The advantage of PCA is

- Its ability to be able to express the data in smaller dimension
- Ordering of the synthetic features it creates (the components)

It is common to apply PCA to layer 0: the input $\mathbf{x}$.

This can be used to find clusters of examples that have common input properties.

But one can apply PCA to *any* layer, where the synthetic properties may be more complex.

PCA will be used to find *clusters* of examples

- That produce a similar feature map k at layer l

We will reduce the large spatial dimension

- To a smaller dimension
- Retaining only the "most important" locations

If we can identify a property that is common to all examples in the cluster

- We can interpret feature map k of layer l as implementing the feature

"Is the property present in the input ?"

PCA of Feature Maps

It is hard to find clusters when objects are of high dimension

- With so many dimensions
- Any distance measurement tends to be large even for similar objects
- Because the number of *irrelevant* elements
- May be larger than the number of relevant elements

Consider a feature map $\mathbf{y}_{(l),\ldots,k}$ with spatial dimension (1000×1000)

- A typical image size
- Two examples have a dog in the center
- Surrounded by much different backgrounds

If the number of spatial locations in the background is much larger than the region containing the dog

- Then these two similar examples
- Have large distance
- Due to the different, but irrelevant, backgrounds

We can use *dimensionality* reduction techniques of Classical Machine Learning.

One such technique is Principal Components Analysis

- Find a small number of synthetic features
- That express commonalities of many examples
- Represent an example in a synthetic feature space
- Of reduced dimensions

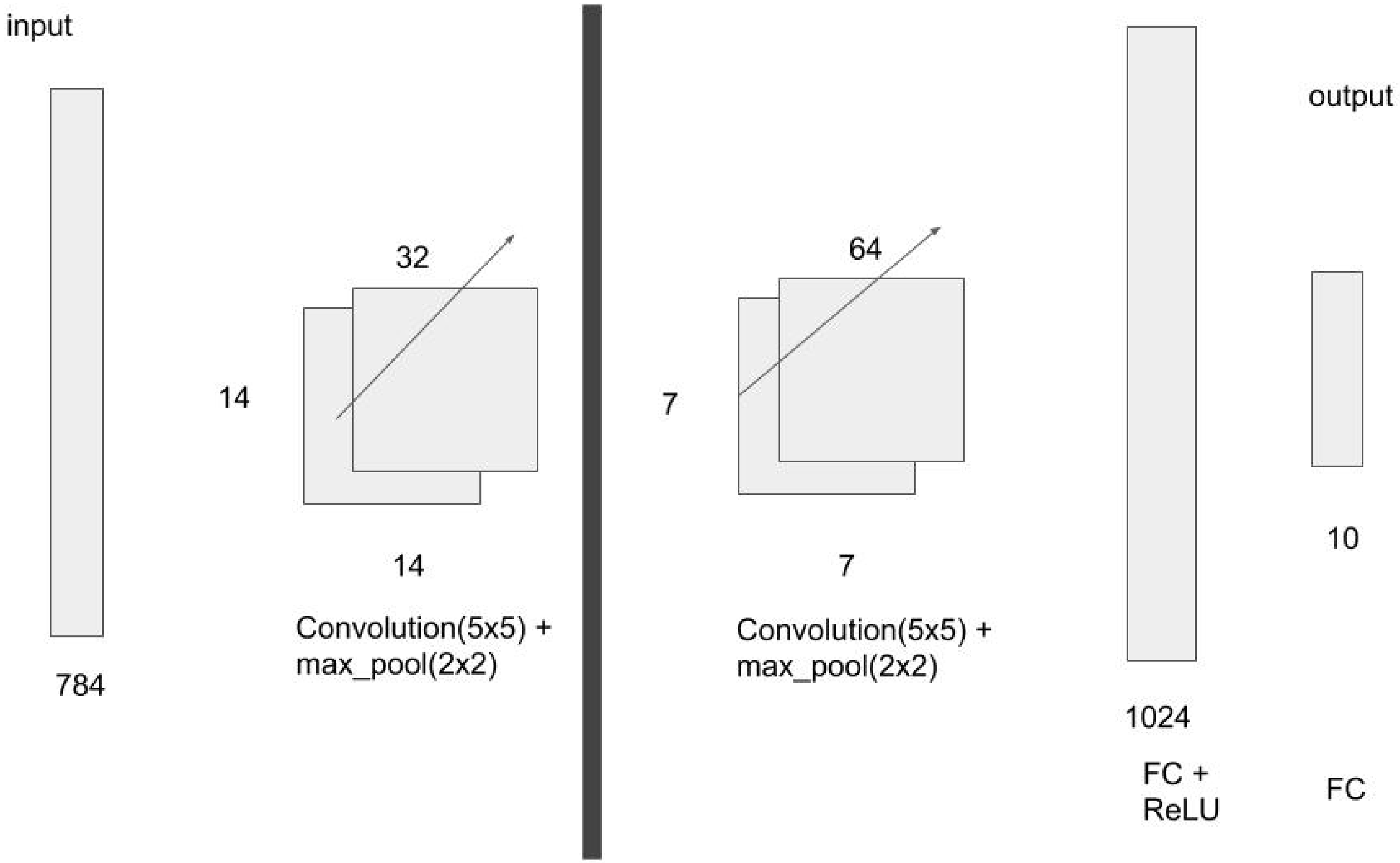
In this case: we are reducing the number of spatial locations

Here is a two layer Neural Network that we built to classify digits in the MNIST dataset

MNIST CNN

PCA

- Interpret the intermediate representation (what is the transformed feature ?)

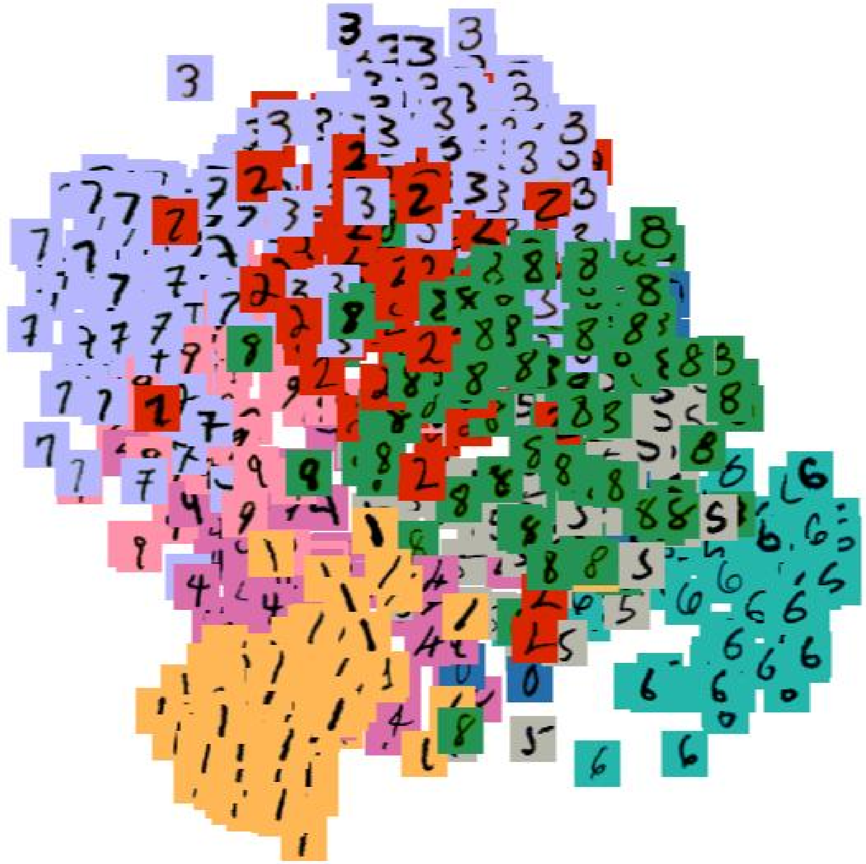


We perform PCA on the representations produced by the first Convolutional Layer (dark vertical line)

- Plotting each example
- Using the two most important synthetic features (components) as coordinates in the plot

MNIST CNN Conv1 PCA

PCA: MNIST Deep Classifier (post conv1)



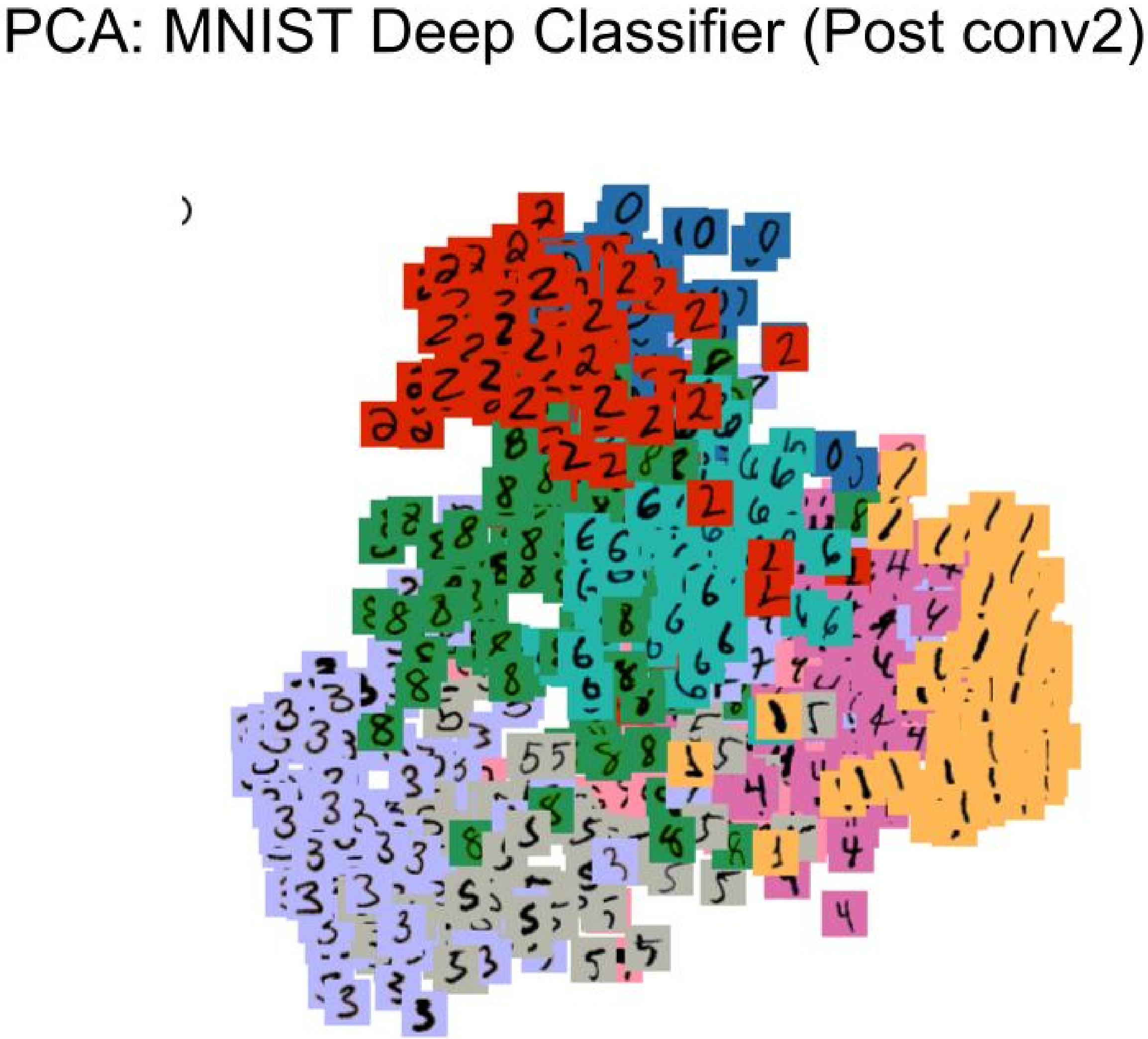
Clusters are starting to appear.

Do these clusters give us a clue as to the property that the layer is representing ?

- Left to right: strong vertical ("1", "7") to less vertical ?
- Bottom to top: digits *without* "curved tops" to those with tops ?

Let's perform the same analysis on the representations of the second Convolutional layer

MNIST CNN Conv1 PCA



The clusters become "more pure".

So the deeper representation

- May be finding *combinations* of input features
- That cluster similar digits

So we might be able to interpret what the first two Convolutional Layers are representing

- Without necessarily understanding what the second layer is doing in isolation

In [4]:

```
print("Done")
```

```
Done
```