SVD-based Principal Component Analysis and Image Decomposition

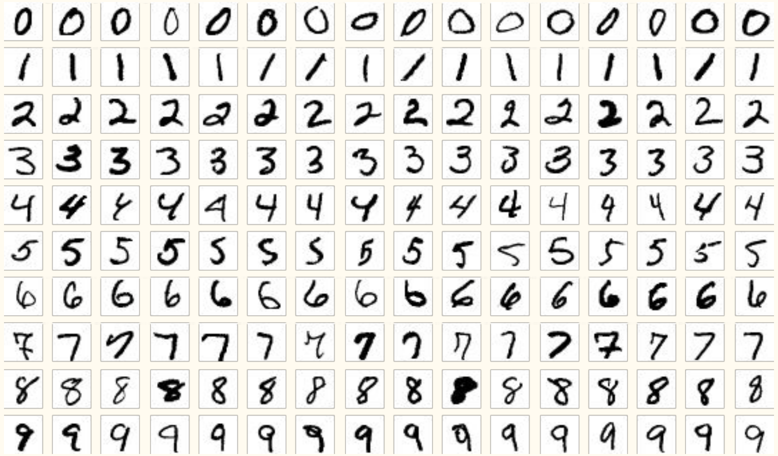
By Jonathan Ang, Marley Abowitz, Alexander Liu, Grayson Newell

Background

- Images used for machine learning (ML) modeling often use more data than necessary, causing ML applications to be more expensive.
- PCA identifies orthogonal principal components that maximize data variance. This allows us to extract important features, thus enabling effective compression.

MNIST Dataset

- The MNIST dataset is a set of images of digits 0-9, containing 70,000 images of handwritten digits.
- Each entry is a greyscale image (0-255) of size 28 x 28, containing 784 pixels. Each image is thus in a vector $x \in R^{784}$.
- Challenge posed by this dataset is it's high-dimensional qualities.

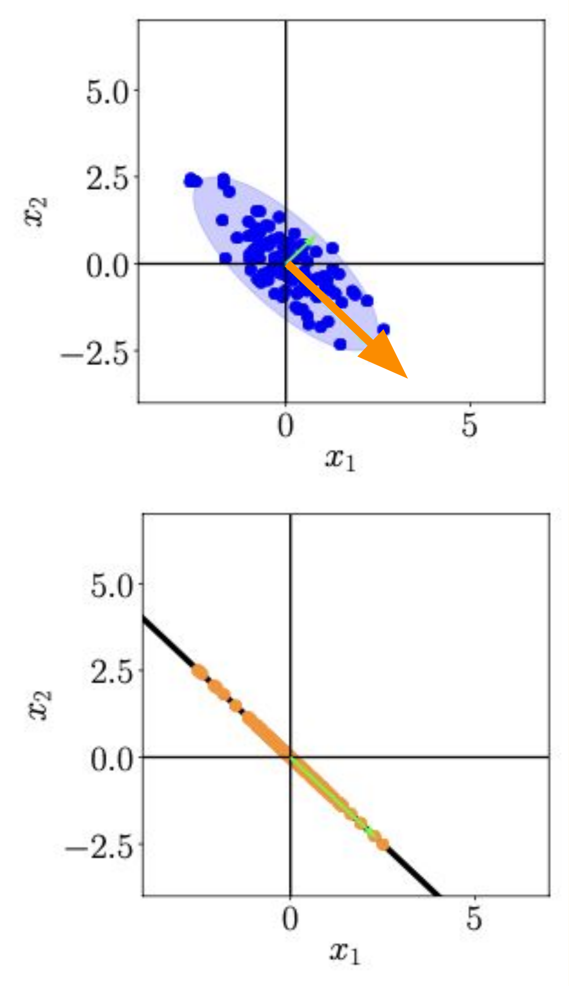


Methodology: Data Matrix

- The dataset matrix must be normalized so that each column has a mean of zero and standard deviation of one.
- For each column, the mean is subtracted from all of its entries, and the difference is then divided by the standard deviation.
- PCA can now be applied to this matrix by finding its Singular Value Decomposition, yielding a set of vector/value pairs.

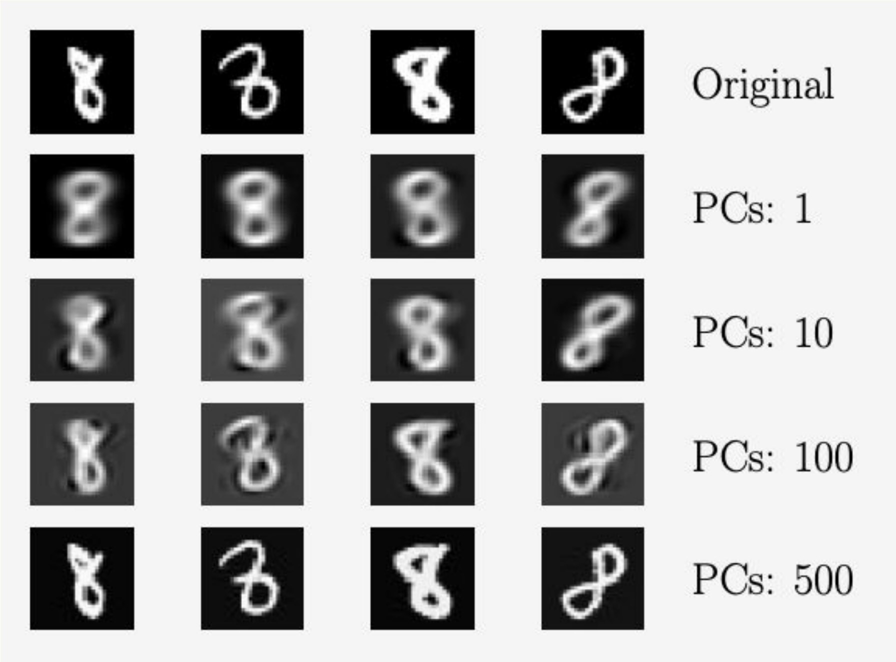
Methodology: Singular Values

- Each vector from SVD is interpreted as a principal component.
- The first principal component, with the highest singular value, will reflect the direction of most variance, as shown to the right.
- The set of the first m principal component vectors forms a PCA space.
- The data can then be projected onto this space, providing an approximation.



Methodology: PCA Spaces

- Rank-m approximations of the original data.
- Retains most important characteristics using less space.
- As m increases, PCA becomes more accurate but less concise.
- <u>What value of *m* maximizes the accuracy with which the PCA space represents the data while minimizing its dimensions?</u>



```python
from sklearn.decomposition import PCA
from sklearn.datasets import fetch_openml

mnist = fetch_openml('mnist_784', as_frame=False, parser="auto")
X_train, y_train = mnist.data[:60_000], mnist.target[:60_000]
X_test, y_test = mnist.data[60_000:], mnist.target[60_000:]

pca = PCA(n_components = 100)
X_reduced = pca.fit_transform(X_train)
X_test_reduced = pca.transform(X_test)
```

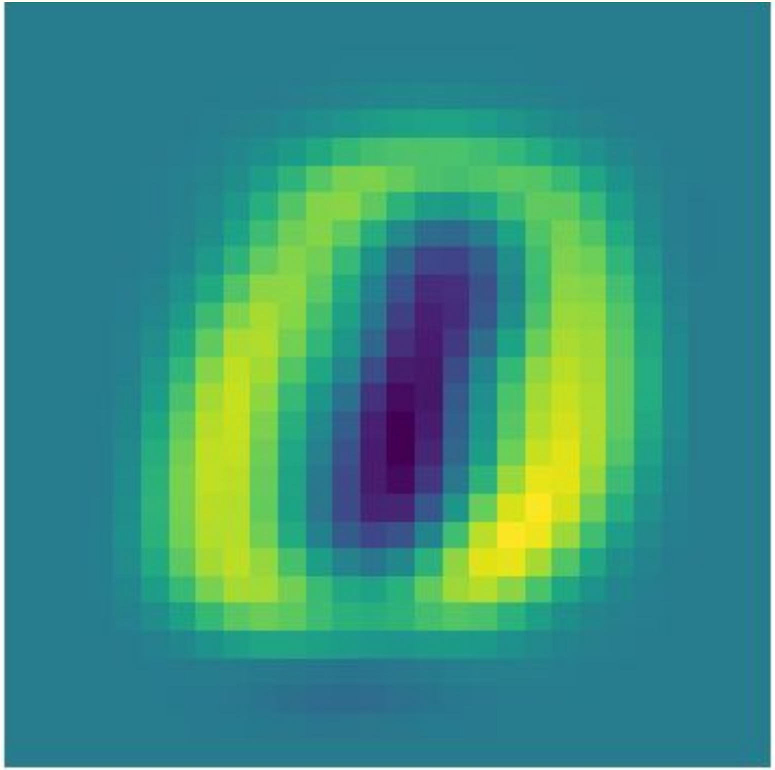
```
print(pca.components_)

[[-6.25278585e-18  8.05687112e-19 -5.24460416e-18 ... -0.00000000e+00
  -0.00000000e+00 -0.00000000e+00]
 [ 1.95930890e-17  1.93272025e-17  2.83953769e-17 ... -0.00000000e+00
  -0.00000000e+00 -0.00000000e+00]
 [ 1.51940654e-17  6.53495440e-17  3.21349485e-17 ... -0.00000000e+00
  -0.00000000e+00 -0.00000000e+00]
 ...
 [-2.77830478e-17 -2.21680715e-17  4.48948266e-17 ... -0.00000000e+00
  -0.00000000e+00 -0.00000000e+00]
 [-3.91815632e-17 -3.44865263e-17  3.98882184e-17 ...  0.00000000e+00
   0.00000000e+00  0.00000000e+00]
 [ 4.08302875e-17 -2.64599064e-17  1.46208157e-17 ... -0.00000000e+00
  -0.00000000e+00 -0.00000000e+00]]
```

```python
print(pca.components_[0])
```

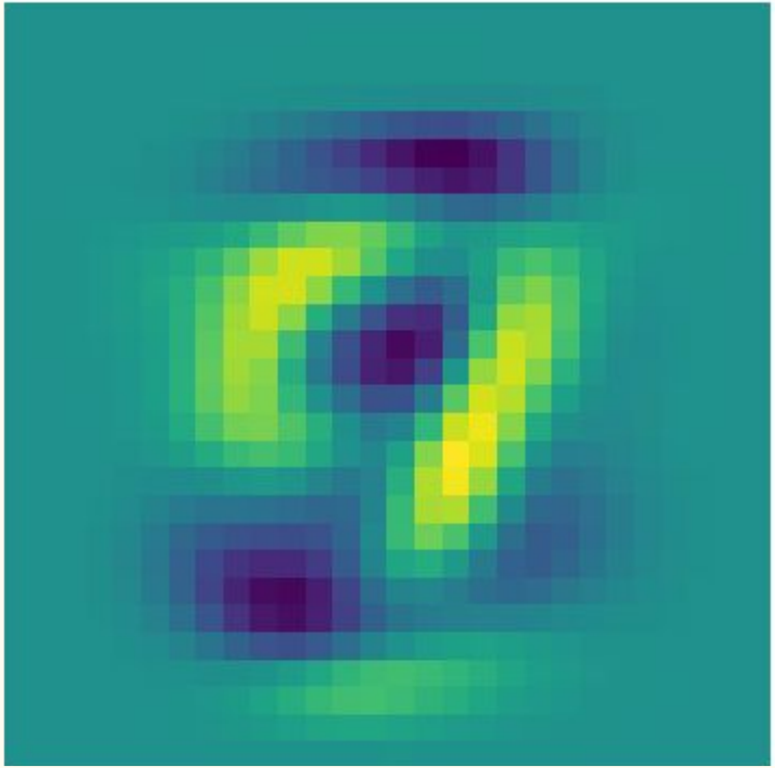
```
[-6.25278585e-18  8.05687112e-19 -5.24460416e-18  3.27144520e-19
  1.65516632e-18 -9.03756733e-20  1.39938810e-19  2.52303625e-20
 -2.26734354e-20 -1.26507311e-20 -2.40340720e-20 -9.53922034e-21
 -1.13022999e-06 -4.44987008e-06 -2.19785960e-06 -9.15774833e-08
 -2.94985563e-24  3.09330892e-22  3.72499363e-22  7.42305029e-22
 -6.17920651e-22 -2.07028897e-22  4.63872461e-22  4.91046421e-22
 -1.62403759e-22 -8.33652423e-23  1.73280445e-22  4.67986927e-23
 -1.71814134e-23  1.91702430e-23 -3.29633898e-23  3.65474621e-23
  2.46332117e-07  8.70394526e-07  8.07828577e-06  2.07899216e-05
  2.74484785e-05  4.43302473e-05  7.14293298e-05  9.03571728e-05
  8.92534083e-05  7.96187541e-05  8.37900230e-05  6.12003371e-05
  3.32476101e-05  3.18856915e-05  1.84435794e-05 -1.33003534e-05
 -1.08540941e-05  1.55973517e-06  1.06839782e-06  5.04492283e-07
  2.71245813e-25 -3.65834317e-25 -9.77936240e-25  1.70712010e-25
 -6.60389193e-26  1.64756268e-25  1.96615753e-06  1.09264414e-06
  3.78170707e-06  5.42556652e-06  4.54016550e-05  8.80063484e-05
  1.74906911e-04  3.49737899e-04  5.25037751e-04  8.19630806e-04
  1.33334263e-03  1.94378162e-03  2.35926280e-03  2.47464087e-03
  2.41523732e-03  1.99203160e-03  1.27312804e-03  7.61625621e-04
  4.15452682e-04  1.74690829e-04  6.92125344e-05  1.43318304e-05
  8.82496596e-06  3.65876221e-06  1.05053437e-26 -3.09705972e-27
  1.18757656e-26 -3.73895716e-27  2.56757368e-06  7.83497352e-06
  1.44677392e-05  7.88729492e-05  1.32228090e-04  2.45399260e-04
  5.15312055e-04  1.03806702e-03  1.77636679e-03  2.97179446e-03
  4.62742305e-03  6.57263692e-03  8.36456884e-03  9.33782758e-03
  8.86484917e-03  6.95610822e-03  5.00230419e-03  3.34948109e-03
  2.12952848e-03  1.05776479e-03  4.93275652e-04  1.43392142e-04
  1.35708910e-05  1.38381968e-06 -3.22083306e-06 -0.00000000e+00
 -0.00000000e+00  1.70816861e-07 -5.09471827e-06  1.23645490e-05
  4.66637490e-05  2.12590585e-04  5.41138070e-04  1.18119954e-03
  2.57056604e-03  4.95235519e-03  8.35734492e-03  1.29675166e-02
  1.82599204e-02  2.34413491e-02  2.70030959e-02  2.96040182e-02
  2.95255440e-02  2.56951517e-02  2.03682532e-02  1.42620877e-02
  8.75677477e-03  4.94330928e-03  2.41961391e-03  9.43476465e-04
  2.36047259e-04  1.49518827e-05 -7.26074638e-06 -3.34113987e-06
 -0.00000000e+00 -0.00000000e+00  3.82304506e-06  1.72698830e-05
  2.13448332e-04  7.36785192e-04  1.68791645e-03  3.68288404e-03
  7.56324728e-03  1.40248022e-02  2.30913180e-02  3.45098345e-02
```

First PC

Lighter = more positive
Darker = more negative

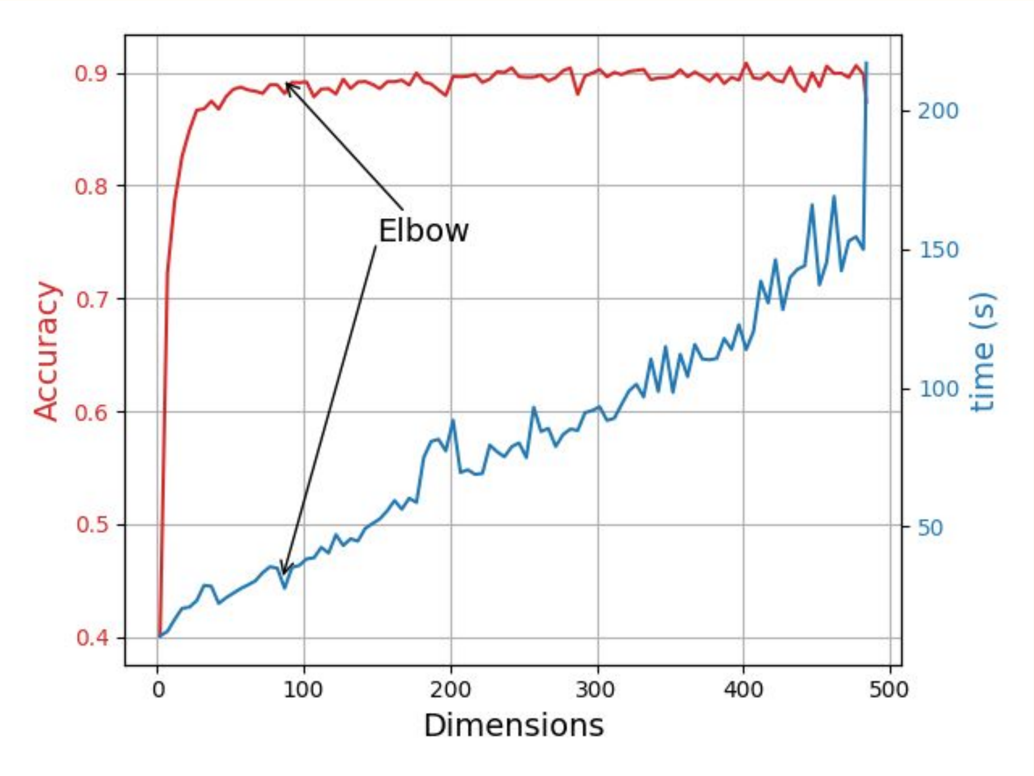


Second PC



Evaluation

1. 85 dimensions is the elbow curve for accuracy and reduction.
2. PCA enabled the model to achieve the highest accuracy of over 0.90 at around 400 dimensions.
3. When compared to the results at 484 dimensions, where PCA was not used at all, the model took a much longer time to train at over 3 minutes whilst also achieving a far lower accuracy at 0.874



Evaluation

- PCA is a useful tool for ML models.
- Two benefits of PCA:
  - It reduces the data size of data sets so that less computational power is require to train ML models on it.
  - It enables researchers to uncover patterns and relationships within the data that may not be apparent in the original dataset.

THANK YOU FOR WATCHING!