Principal Component Analysis

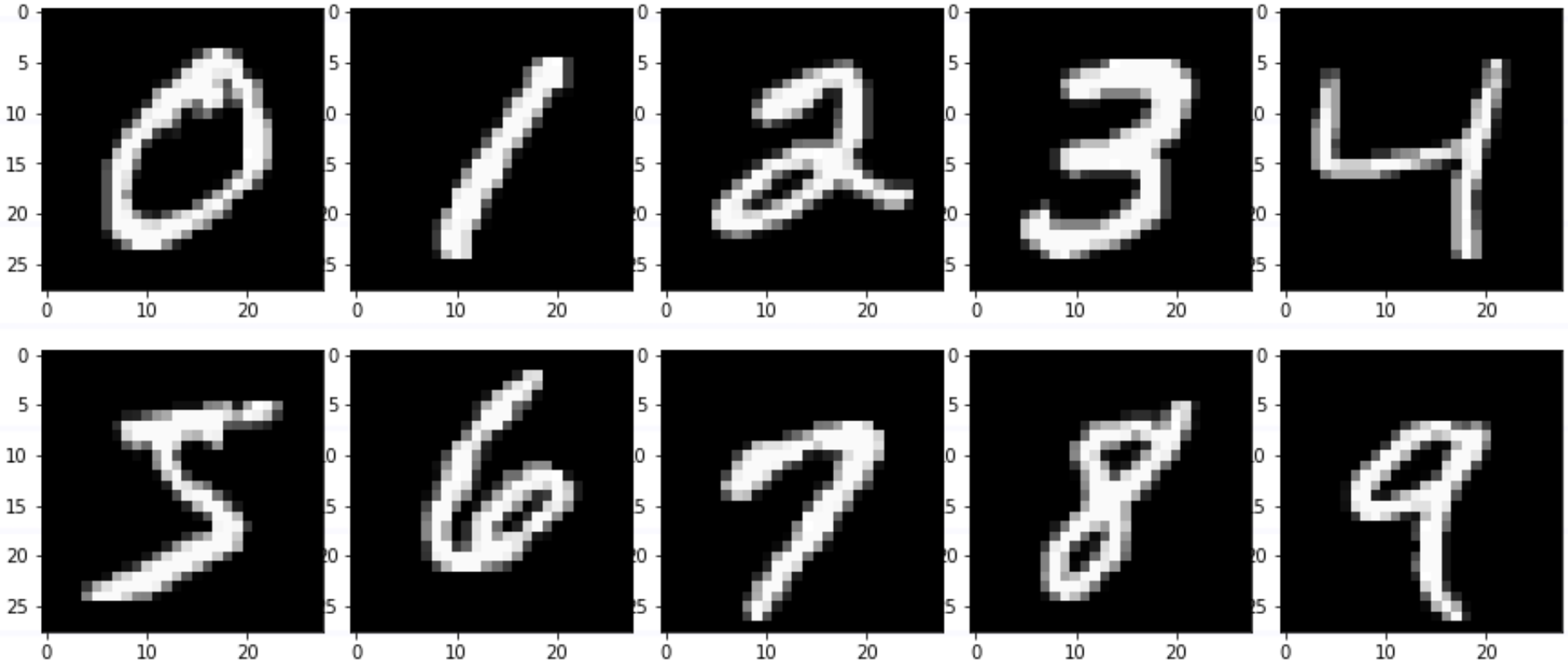
EE698V - Machine Learning for Signal Processing

Vipul Arora

References

- PRML Section 12.1 **(highly recommended)**

Image Classification



Variations

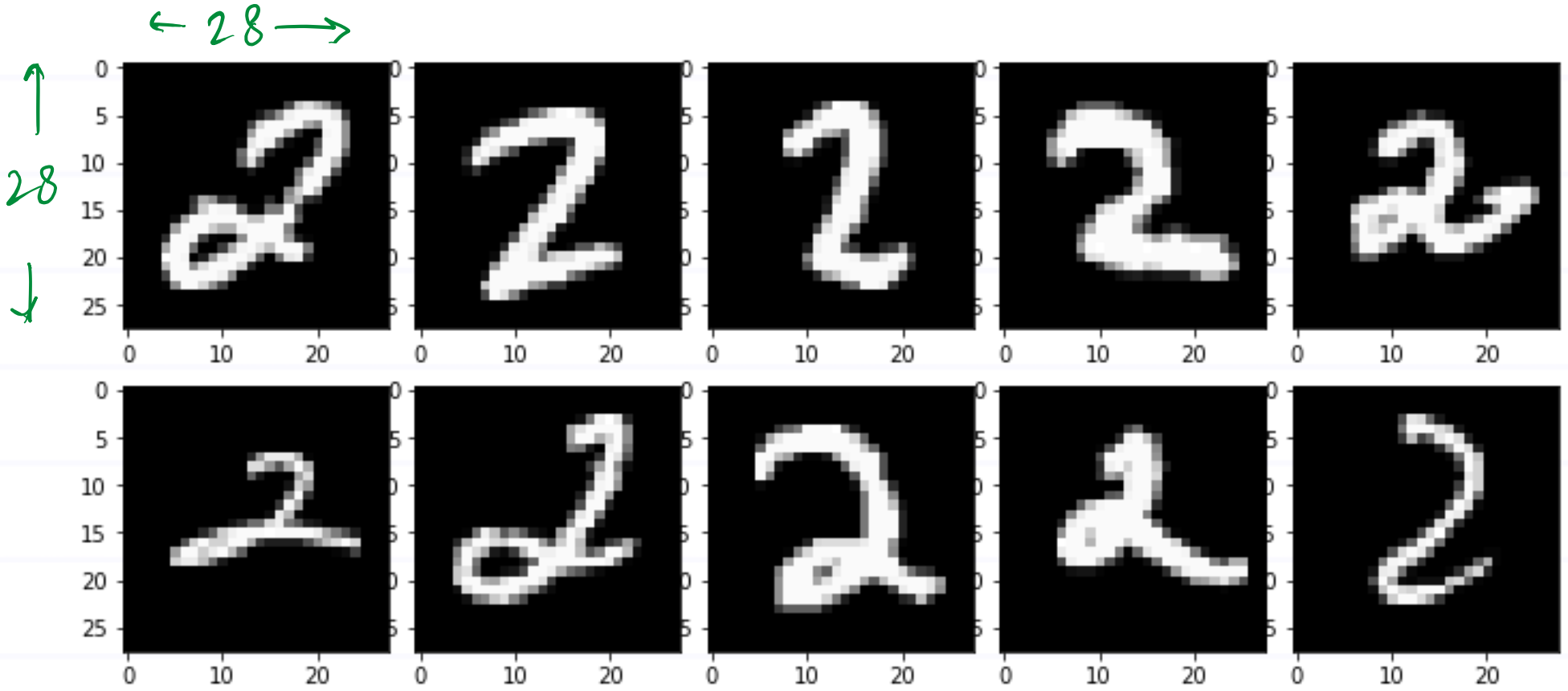
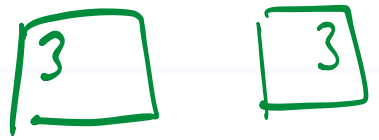


Image Classification

- What is one sample here?



X : 28 x 28 matrix

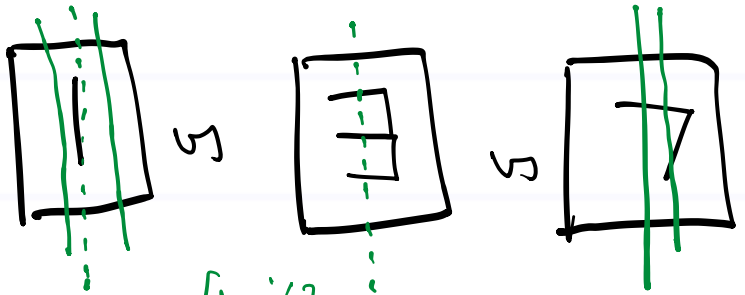
$X[i,j]$

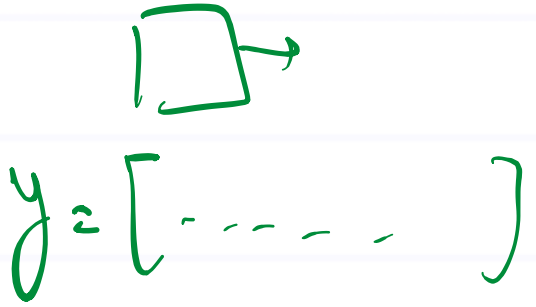
$X[i,:]$

$X[:,i]$

$X[:,:]$

dass $y \in \{0, \dots, 9\}$





$x[:, j]$

28 cols.

$y_j = 10 \times 1$

$Y = [10 \times 28]$

$y_j = \max(Y, \text{axis} = 1)$ sum avg

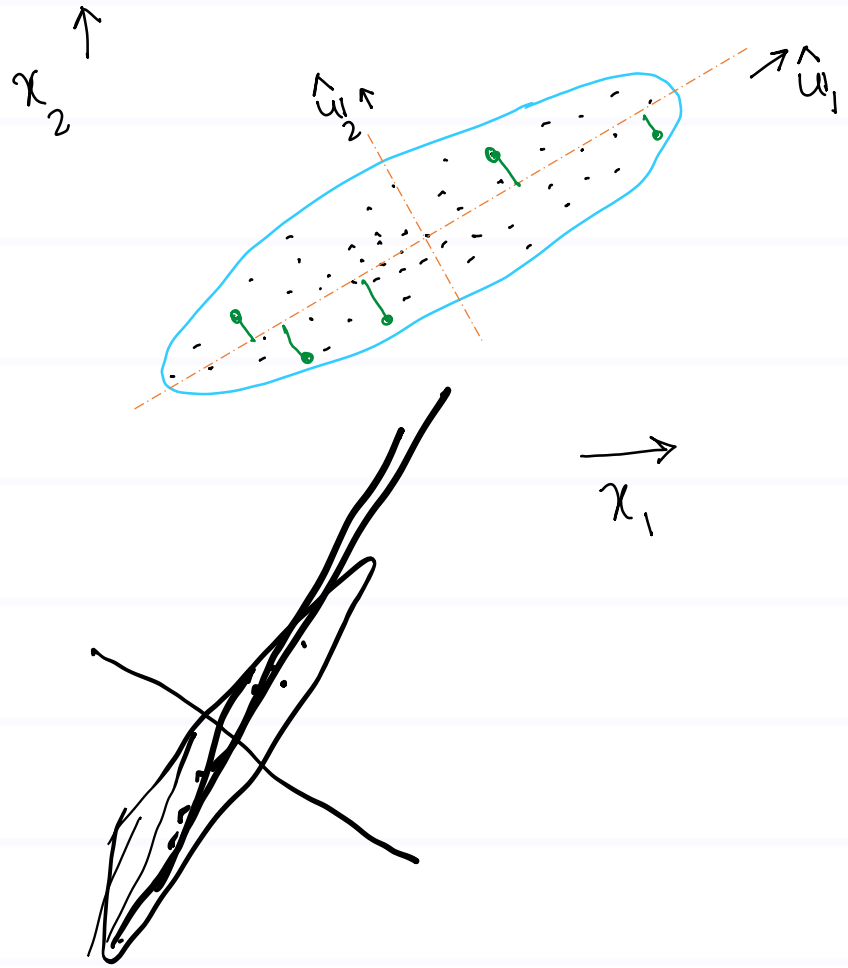
$y_j = 10 \times 1 \rightarrow$ to decide class

$c = \underset{c}{\text{argmax}}\ y[c]$

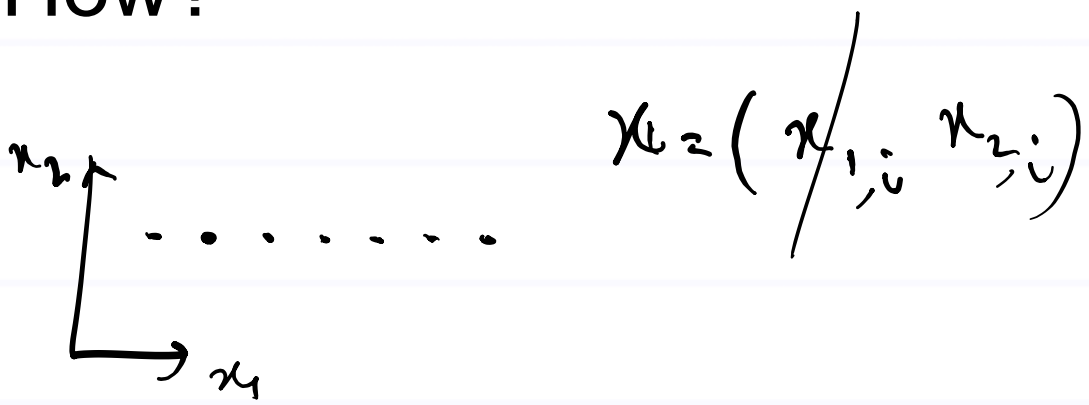
Classification

- Despite all variations in $\boldsymbol{x}$, the information lies in a space of much lower dimensionality
- Instead of processing large sized samples, can we transform them to a lower dimensional space?
- E.g., we extracted height and width of flowers!

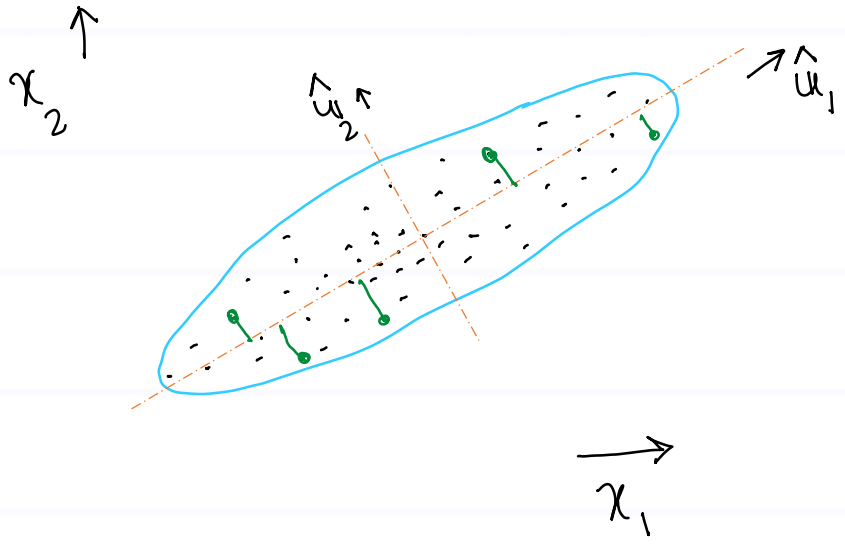
Projection to subspace



- The projection on u_1 axis contains more information than that on u_2 axis
- How?



Projection to subspace



- What is the projection on $\boldsymbol{u}_1$ axis?

$$x^T u_1$$

- What is the projection on $\boldsymbol{u}_2$ axis?

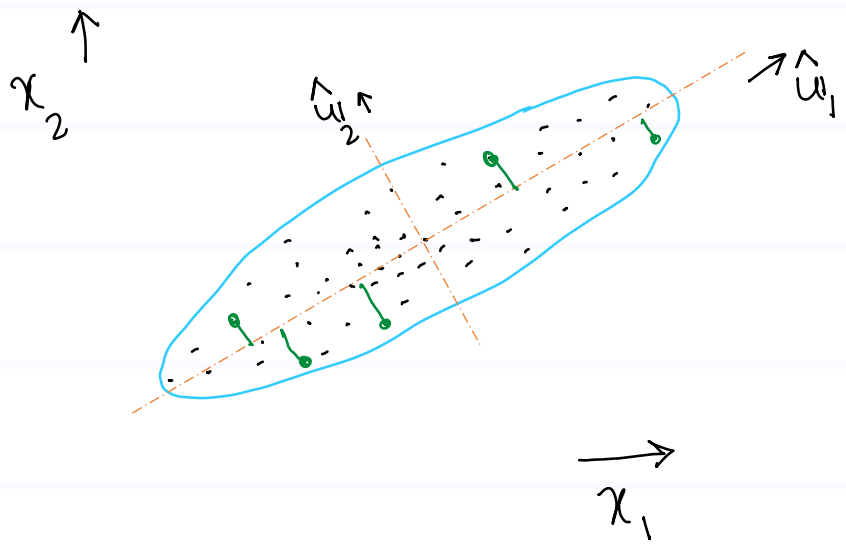
$$x^T u_2$$

- So, we can write $\boldsymbol{x}$ as:

$$\boldsymbol{x} = (x^T u_1)\, u_1 + (x^T u_2)\, u_2$$

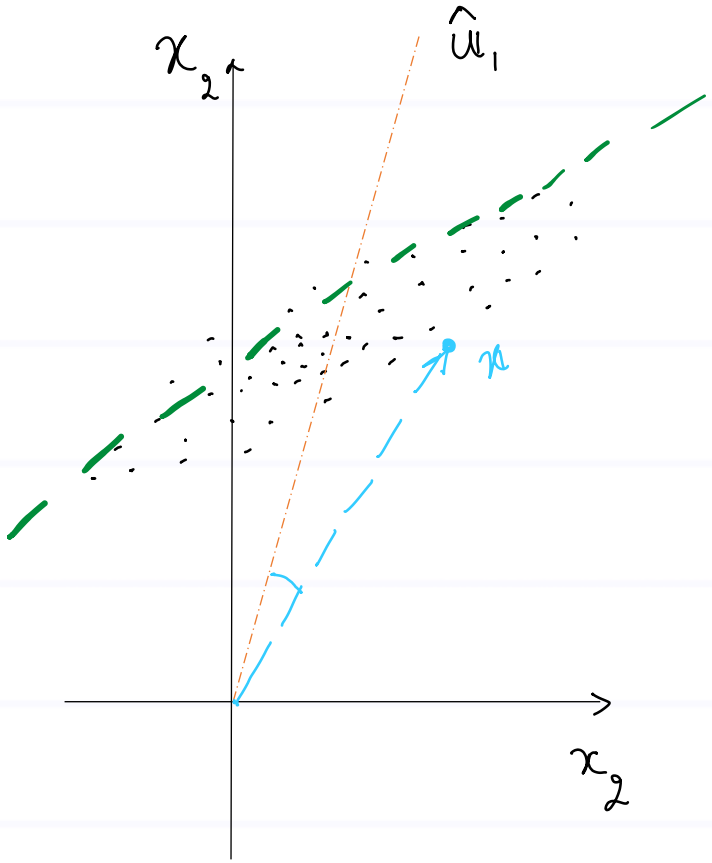
more info less info

Finding $\boldsymbol{u}_1$



- $\boldsymbol{u}_1$ is the axis along which there is maximum variation
- So, we can discard $\boldsymbol{u}_2$

Finding u_1



- But hold on ...
- What is the direction of maximum variation here?

$(x^T u_1) u_1 + (x^T u_2) u_2$

max min

- What is the error?

origin should be mean

Finding $\boldsymbol{u}_1$

- Mean shifting: make the data zero-mean first

How to find optimal $\boldsymbol{u}$ axes?

- Minimize the information lost

$$\mathbf{x} = \sum_{i=1}^{D} \left(\mathbf{x}^T \mathbf{u}_i\right) \mathbf{u}_i \quad ; \quad \mathbf{x} \in \mathbb{R}^D$$

$$\tilde{\mathbf{x}} = \sum_{i=1}^{M} \left(\mathbf{x}^T \mathbf{u}_i\right) \mathbf{u}_i \quad ; \quad M < D$$

How to find optimal u axes?

- Mean squared error ... again!

$$x - \tilde{x} = \sum_{i=M+1}^{D} (x^T u_i) u_i$$

$$E = \frac{1}{N} \sum_{n=1}^{N} (x_n - \tilde{x}_n)^2 = \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{i=M+1}^{D} (x_n^T u_i) u_i \right)^2$$

$$= \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} (x_n^T u_i)^2 \underbrace{u_i^T u_i}_{=1 \text{ if } u\text{'s orthonormal}} + \sum_{i=M+1}^{D} \sum_{\substack{j=M+1 \\ j \neq i}}^{D} (x_n^T u_i)(x_n^T u_j) \underbrace{u_i^T u_j}_{=0 \text{ if } u\text{'s orthonormal}}$$

$$= \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} (x_n^T u_i)^2$$

How to find optimal $\boldsymbol{u}$ axes?

$$E = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} \underbrace{\left(x_n^T u_i\right)^T \left(x_n^T u_i\right)}_{u_i^T x_n x_n^T u_i}$$

$$= \left(u_i^T x_n\right)^T \left(u_i^T x_n\right)$$

$$= x_n^T u_i u_i^T x_n$$

$$E = \sum_{i=M+1}^{D} u_i^T \underbrace{\left(\frac{1}{N} \sum_{n=1}^{N} x_n x_n^T \right)}_{S = \text{variance}} u_i$$

[remember x_n is zero mean]

Also, $u_i^T u_i = 1$, so add a Lagrange multiplier term

How to find optimal u axes?

$$E = \sum_{i=M+1}^{D} \left[u_i^T S u_i + \lambda_i (1 - u_i^T u_i) \right]$$

Now minimize E w.r.t. $u_{i'}$:

$$\frac{\partial E}{\partial u_{i'}} = 2 S u_{i'} - \lambda_{i'} 2 u_{i'} = 0$$

$$\Rightarrow \underset{D\times D}{S} \underset{D\times 1}{u_i} = \lambda_i \underset{D\times 1}{u_i}$$

($\Rightarrow \lambda_i$ is eigen value of S)

left multiply with u_i^T

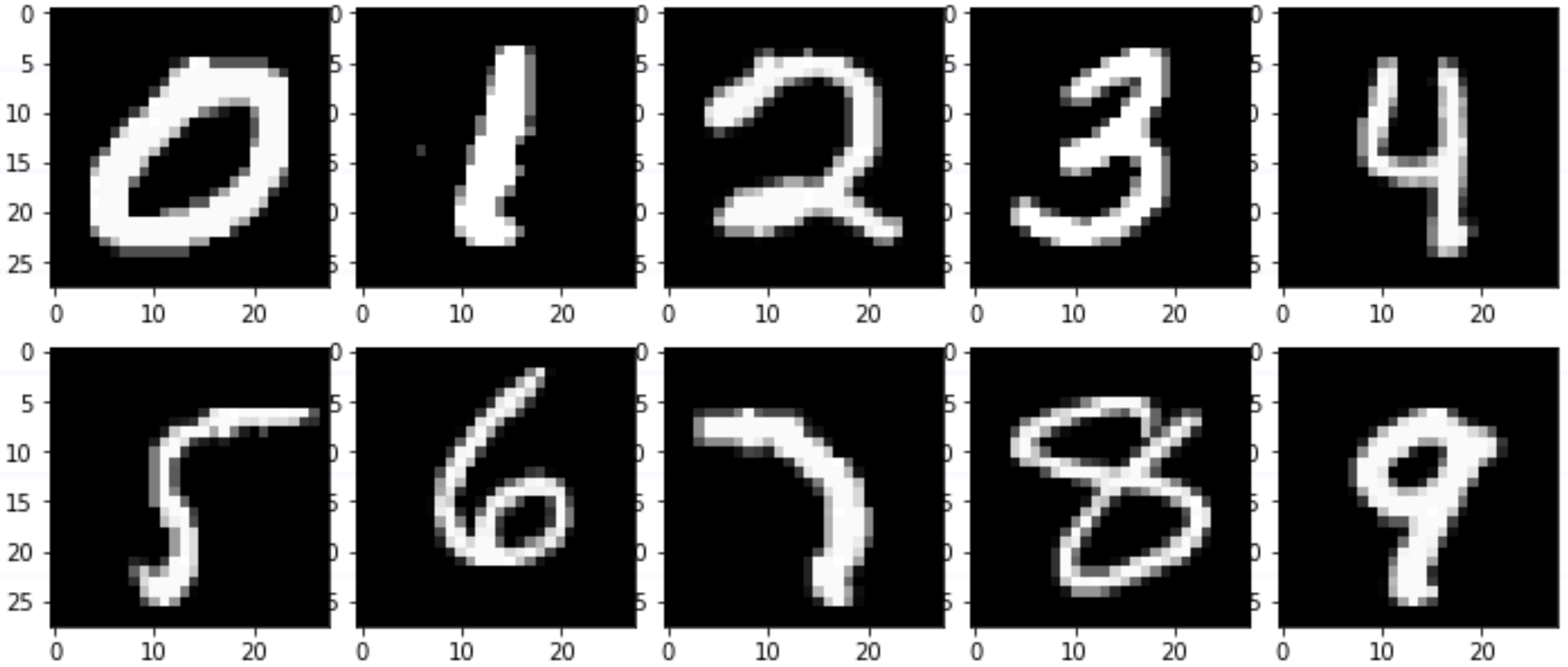
$$u_i^T S u_i = \lambda_i$$

$$\Rightarrow E = \sum_{i=M+1}^{D} \lambda_i$$

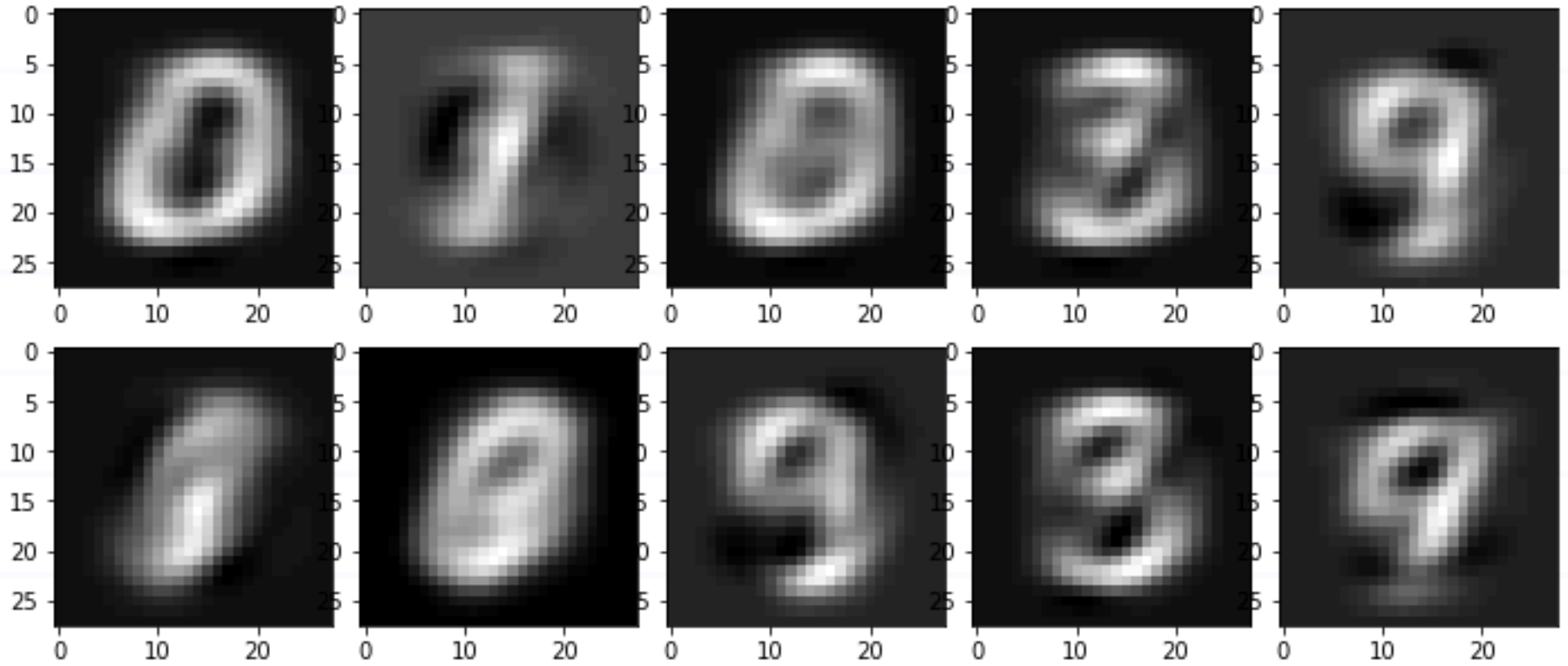
Principal Component Analysis

- Given data samples $\boldsymbol{s}_n \in \mathbb{R}^D$
- Normalize: $\boldsymbol{x}_n = \boldsymbol{s}_n - \mathbb{E}[\boldsymbol{s}]; \quad \mathbb{E}[\boldsymbol{s}] = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{s}_n$ $\quad x_n \in \mathbb{R}^D$
- Obtain variance matrix $S = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^T$
- Eigen value decomposition of S to get $\lambda_i, \boldsymbol{u}_i; i = 1, \ldots, D$ with λ_i in decreasing order $\quad u_i \in \mathbb{R}^D$
- Choose first M eigen vectors as the principal axes
- $\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{M} (\boldsymbol{x}_n^T \boldsymbol{u}_i)\boldsymbol{u}_i$ $\quad \tilde{x}_n \in \mathbb{R}^D$

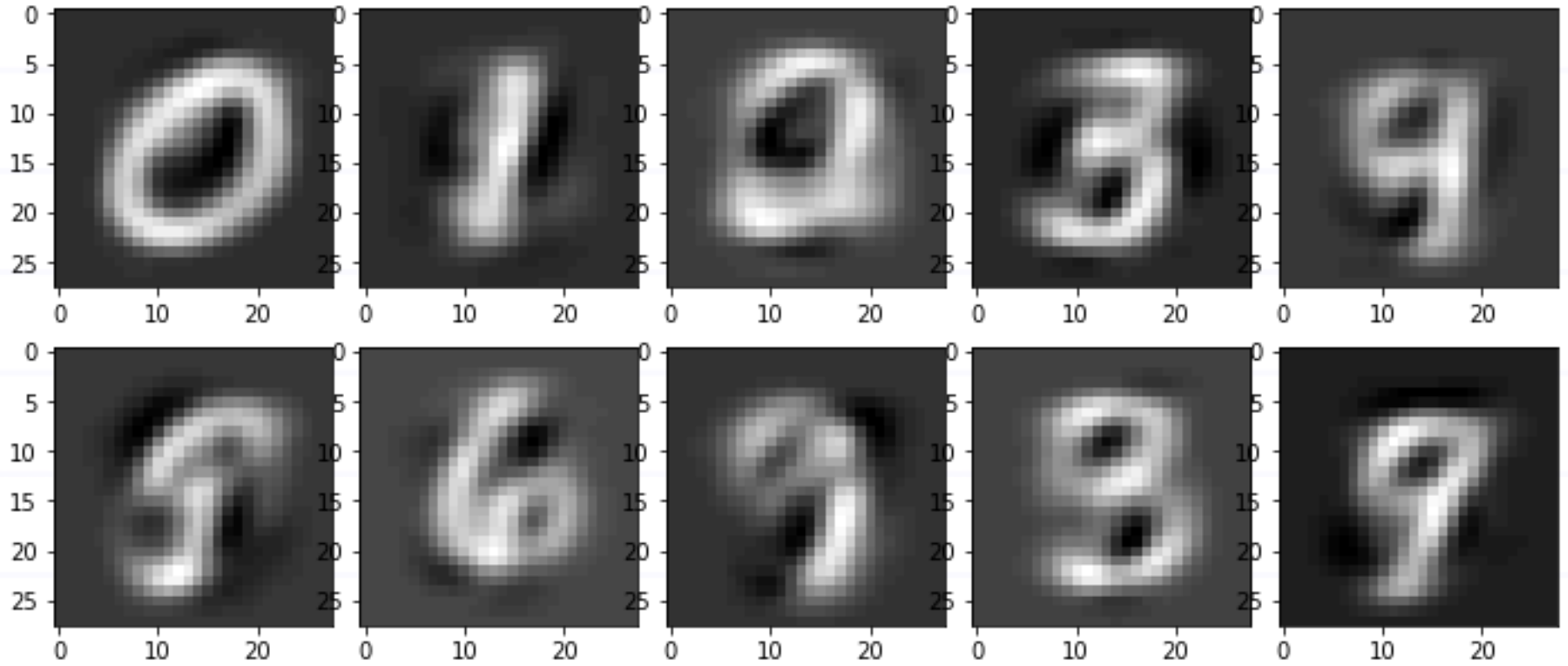
Original



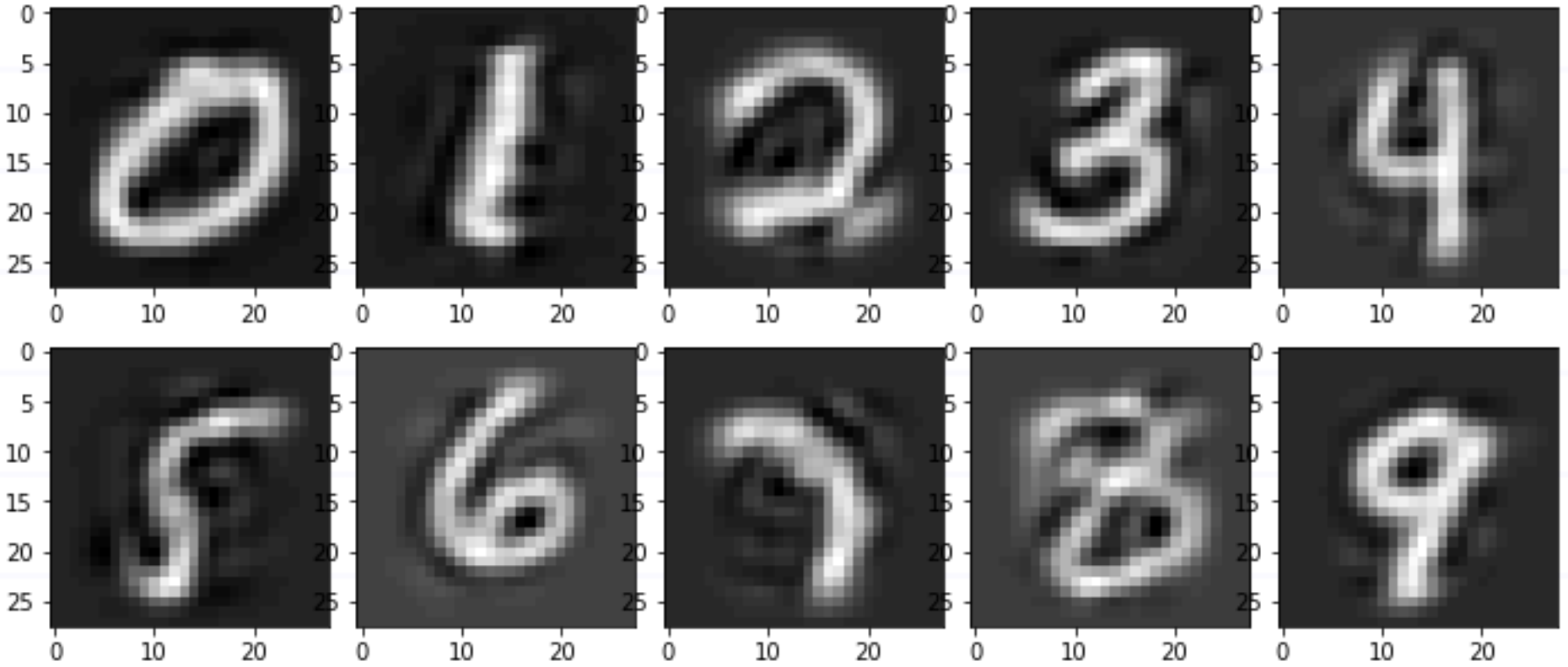
PCA with M=3



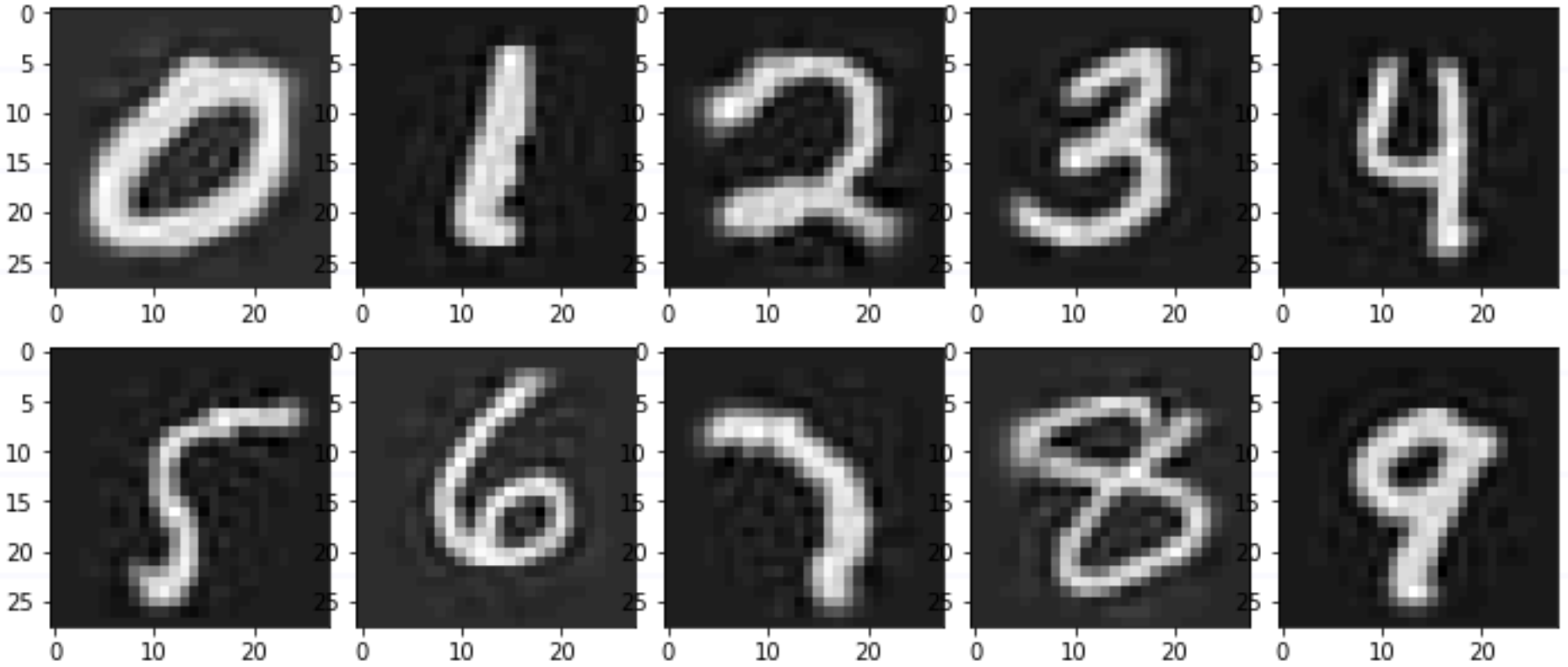
PCA with M=10



PCA with M=50



PCA with M=150



PCA as transformation

$$\tilde{x}_n = \sum_{i=1}^{M} \left(u_i^T x_n \right) u_i$$

$$y_n = \begin{bmatrix} u_1^T x_n \\ u_2^T x_n \\ \vdots \\ u_M^T x_n \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \dots & u_M \end{bmatrix}^T x_n = U^T x_n$$

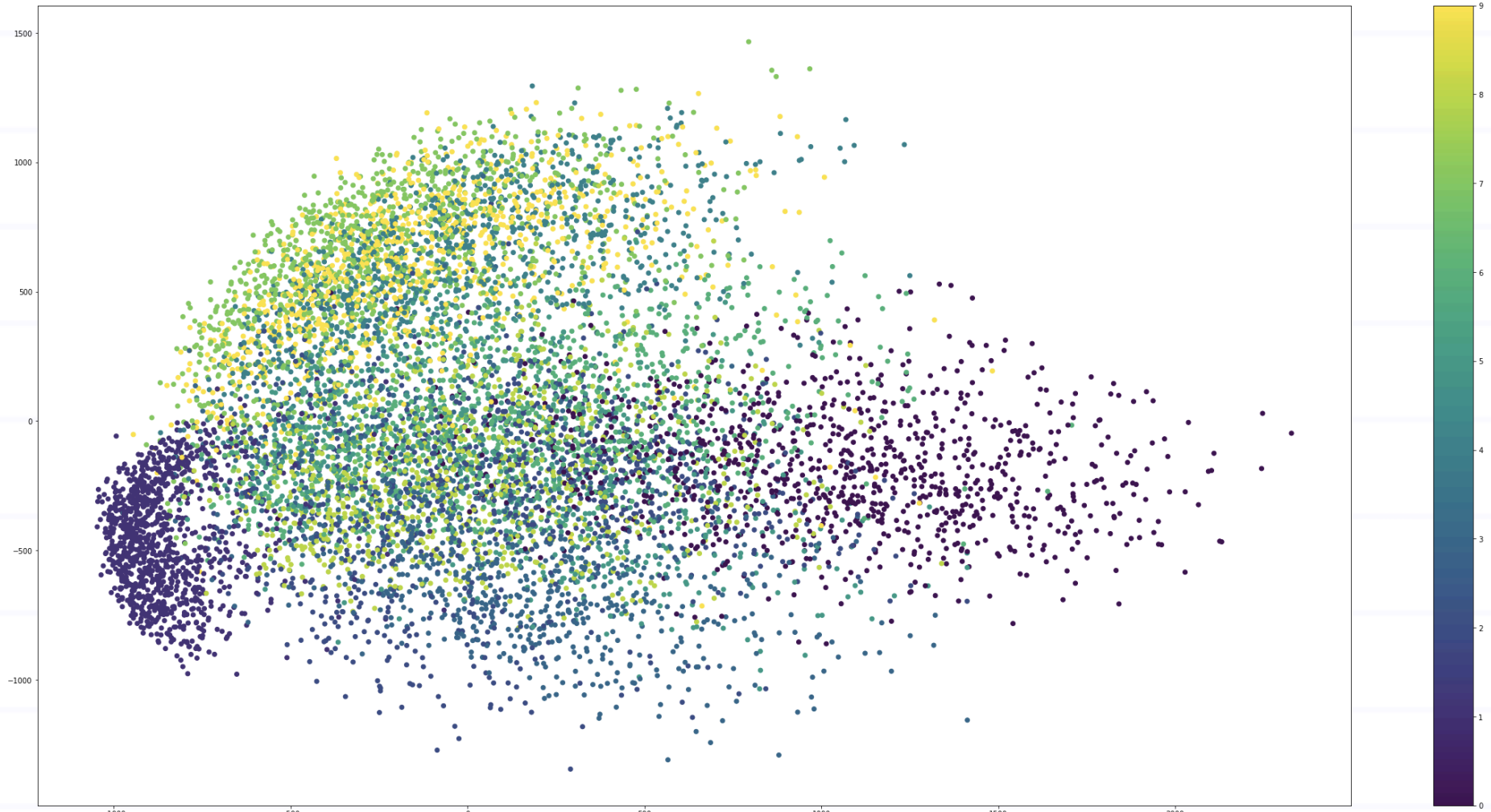
$x_n \in \mathbb{R}^D$

$y_n \in \mathbb{R}^M$

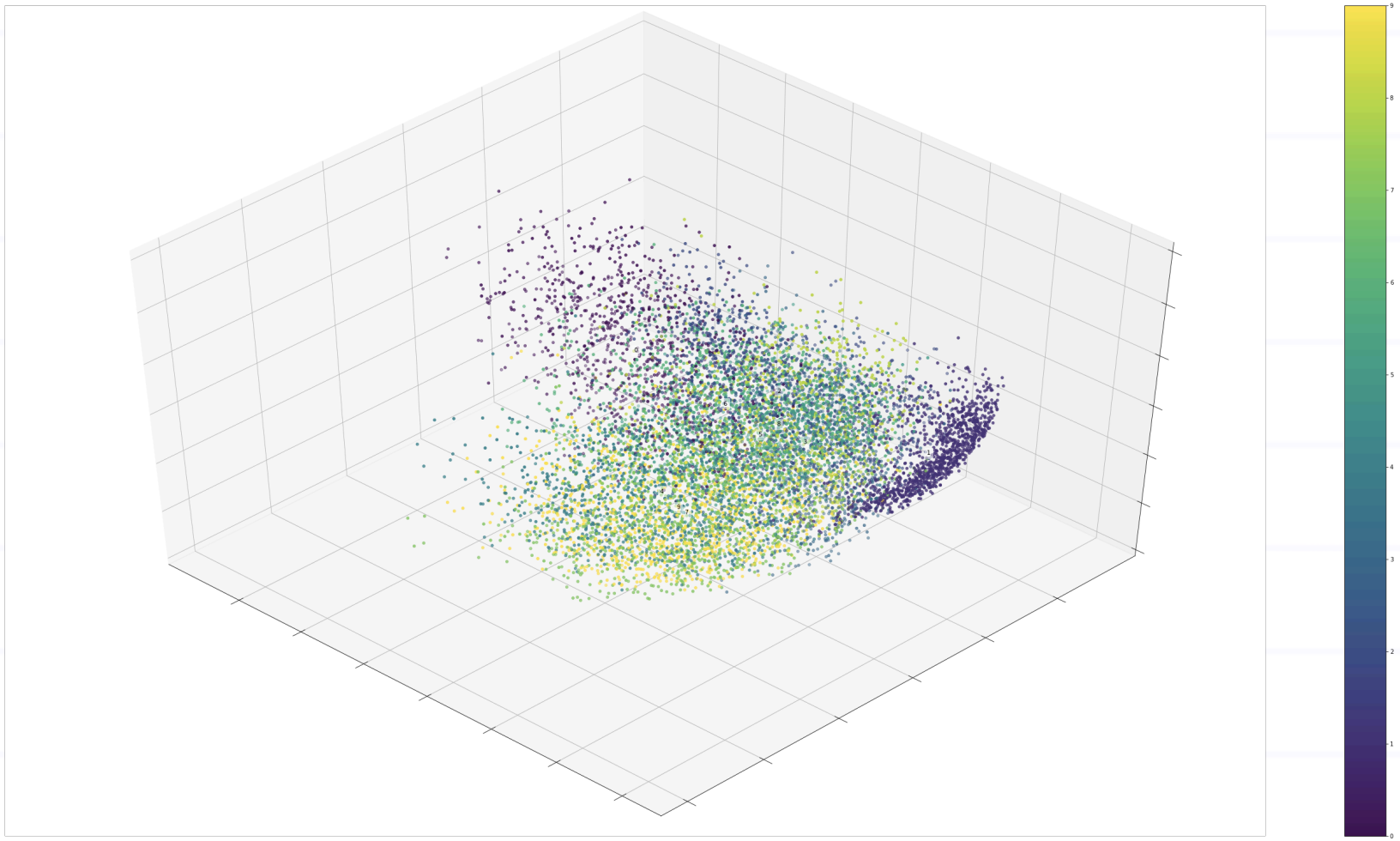
U is $D \times M$

- Prove that y_n is zero-mean
- What is the variance of y_n ?

PCA on MNIST images



PCA on MNIST images



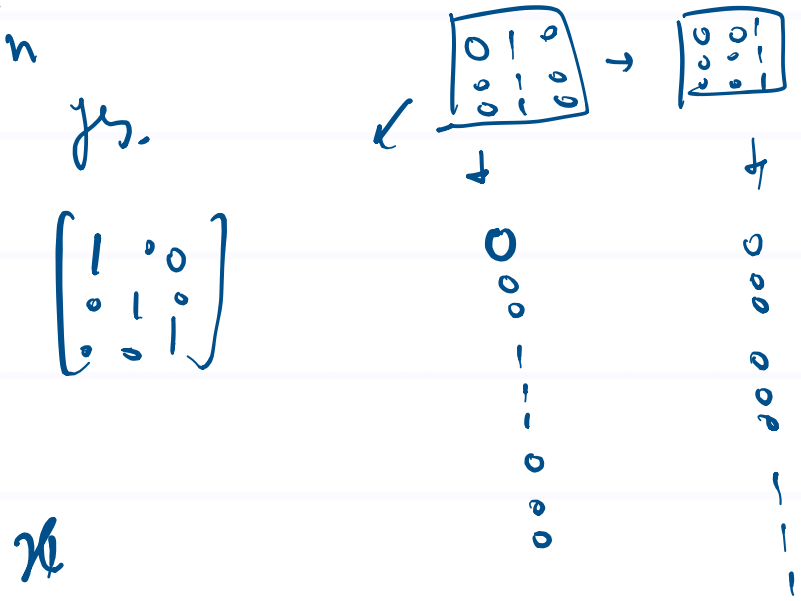
PCA as transformation

- Is PCA invariant to
  - Transversal shift
    - in images — no
    - in vectors — yes
  - Rotation
    - in images — no
    - in vectors — yes

$x'_n = x_n + a \qquad \forall n \in \{1, \dots, N\}$

$y'_n \stackrel{?}{=} y_n$

yes.



$x'_n = A x_n$

$y'_n \stackrel{!}{=} y_n$ yes

PCA as normalization

$$\frac{1}{N}\sum_n y_n y_n^T = \frac{1}{N}\sum_n U^T x_n x_n^T U = U^T S U$$

$$UL = L$$

Recall $SU = UL$, u_i were eigen vectors of S, and $L = \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ & & \ddots \\ 0 & & \lambda_M \end{bmatrix}$

$D \times D \quad D \times M \quad D \times M \quad M \times M$

and $U^T U = I$ $\because$ orthonormal

$$\therefore \frac{1}{N}\sum_n y_n y_n^T = L^{1/2} L^{1/2}$$

Can we make it unit variance? $y_n = L^{-1/2} U x_n$

PCA as Matrix Factorization

$$y_n = U^T x_n \quad \forall\, n \qquad\qquad U = [u_1\ u_2 \cdots\ u_M]$$

In matrix form: $[y_1\ y_2 \cdots\ y_N] = U^T [x_1\ x_2 \cdots\ x_N]$

$$Y = U^T X$$

Y: $M \times N$; U^T: $M \times 784$; X: $784 \times N$

- This is a way of matrix factorization
- Here, Y, U, X can have any real values
- But for images, negative values do not make sense.
- Can we constrain U to have only non-negative values? Then Y will also be non-negative

Discrete vs Continuous Latent Variables

- In discrete
  - We map each $\boldsymbol{x}$ to a discrete cluster
- In continuous
  - We map each $\boldsymbol{x}$ to a weighted sum of some vectors (i.e., $\boldsymbol{u}_i$)