N8104: Artificial Neural Networks

Linear Algebra

Sachchida Nand Chaurasia

Assistant Professor

Department of Computer Science
Banaras Hindu University
Varanasi
Email id: snchaurasia@bhu.ac.in, sachchidanand.mca07@gmail.com

Outline

Basic Notations

Linear Algebra in the context of ANN

Scalars, Vectors, Matrices and Tensors

- Scalars
- Vectors
- Matrix
- Tensors

Eigenvalues & Eigenvectors

Singular Value Decomposition

Principal Component Analysis

Outline

Basic Notations

Linear Algebra in the context of ANN

Scalars, Vectors, Matrices and Tensors

- Scalars
- Vectors
- Matrix
- Tensors

Eigenvalues & Eigenvectors

Singular Value Decomposition

Principal Component Analysis

Basic Notations I

- a : A scalar(integer or real)
- $\mathbf{a}$: A vector
- $\mathbf{A}$: A matrix
- A: A tensor
- I_n: Identity matrix with n row and n columns
- I: Identity matrix with dimensionality implied by context
- $e^{(i)}$: standard basis vector $[0, \ldots, 0, 1, 0, \ldots, 0]$ with a 1 at position i
- $\mathrm{diag}(\mathbf{a})$: A square, diagonal matrix with diagonal entities given by $\mathbf{a}$
- a : A scalar random variable
- $\mathbf{a}$: A vector-valued random variable

Basic Notations II

- $\mathbf{A}$: A matrix-valued random variable
- $\mathbb{A}$: A set
- $\mathbb{R}$: The set of real numbers
- $\mathbb{A} \setminus \mathbb{B}$: Set subtraction
- $\mathcal{G}$: A graph
- a_i: Element i of vector $\mathbf{a}$, with indexing starting at 1
- a_{-i}: All elements of vector $\mathbf{a}$ except for element i
- $A_{i,j}$: Element i, j of matrix $\mathbf{A}$
- $A_{i,:}$: Row i of matrix $\mathbf{A}$
- $A_{:,i}$: Column i of matrix $\mathbf{A}$

Basic Notations III

- Ai, j, k: Element i, j, k of a 3-D tensor A
- A :, :, i: 2-D slice of a 3-D tensor
- a_i: Element i of the random vector i $\mathbf{a}$
- $\mathbf{A}^{\mathrm{T}}$: Transpose of matrix $\mathbf{A}$
- $A \bigodot B$: Element-wise (Hadamard) product of $\mathbf{A}$ and $\mathbf{B}$
- det($\mathbf{A}$): Determinant of A
- $\frac{dy}{dx}$: Derivative of y with respect to x
- $\frac{\partial y}{\partial x}$: Partial derivative of y with respect to x
- $\nabla_{\mathbf{x}} y$: Gradient derivative with respect to x
- $\nabla_{\mathbf{X}} y$: Matrix derivatives of y with respect to $\boldsymbol{X}$

Basic Notations IV

- $\nabla_{\mathsf{X}} y$: Tensor containing derivatives of y with respect X
- $\frac{\partial f}{\partial x}$: Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$
- $\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x})$ or $\boldsymbol{H}(f)(\boldsymbol{x})$: The Hessian matrix of f at input point $\boldsymbol{x}$
- $\int f(\boldsymbol{x}) d\boldsymbol{x}$:Definite integral over the entire domain of $\boldsymbol{x}$
- $\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$:Definite integral with respect to $\boldsymbol{x}$ over the set $\mathbb{S}$
- $\mathrm{a} \perp \mathrm{b}$: The random variables a and b are independent
- $\mathrm{a} \perp \mathrm{b} \mid \mathrm{c}$: They are conditionally independent given c
- $P(\mathrm{a})$: A probability distribution over a discrete variable
- $p(\mathrm{a})$: A probability distribution over a continuous variable, or over a variable whose type has not been specified

Basic Notations V

- a$\sim P$: Random variable a has distribution P
- $\mathbb{E}_{x\sim P}[f(x)]$ or $\mathbb{E}f(x)$: Expectation of $f(x)$ with respect to P(x)
- Var($f(x)$): Variance of $f(x)$ under P(x)
- Cov($f(x)$,$g(x)$): Covariance of $f(x)$ and $g(x)$ under P(x)
- $\mathcal{N}(\boldsymbol{x}; \mu, \sum)$: Gaussian distribution over $\boldsymbol{x}$ with mean μ and covariance $\sum$
- $f : \mathbb{A} \rightarrow \mathbb{B}$: The function f with domain $\mathbb{A}$ and range $\mathbb{B}$
- $f \circ g$: Composition of the function f and g
- $f(\boldsymbol{x};\theta)$: A function of $\boldsymbol{x}$ parameterized by θ
- $\sigma(x)$: Logistic sigmoid, $\frac{1}{1+\exp(-x)}$
- $||\mathbf{x}||_{\mathbf{p}}$: L^p norm of $\mathbf{x}$

Basic Notations VI

- $\|\mathbf{x}\|$: L^2 norm of $\mathbf{x}$

Datasets and Distributions

- p_data: The data generating distribution
- $\mathbb{X}$; A set of training examples
- $\mathbf{x}^{(\mathbf{i})}$: The i-th example (input) from dataset
- $\boldsymbol{y}^{(i)}$ or $\mathbf{y}^{(\mathbf{i})}$: The target associated with $\mathbf{x}^{(\mathbf{i})}$ for supervised learning
- $\mathbf{X}$: The $m \times n$ matrix with example $\mathbf{x}^{(\mathbf{i})}$ in row $\mathbf{X}_{\mathbf{i},:}$

Outline

Basic Notations

Linear Algebra in the context of ANN

Scalars, Vectors, Matrices and Tensors

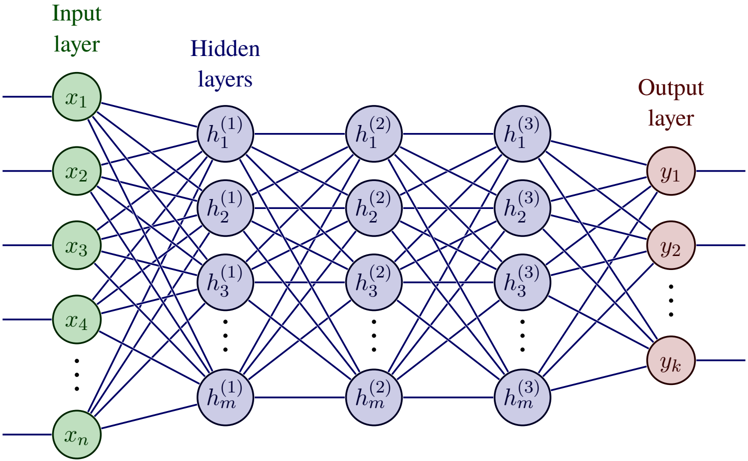
- Scalars
- Vectors
- Matrix
- Tensors

Eigenvalues & Eigenvectors

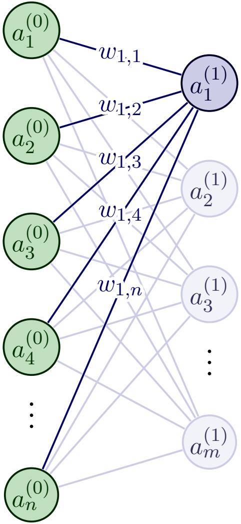
Singular Value Decomposition

Principal Component Analysis

Linear Algebra in the context of ANN I



Linear Algebra in the context of ANN II



$$= \sigma\left(w_{1,0}a_0^{(0)} + w_{1,1}a_1^{(0)} + \ldots + w_{1,n}a_n^{(0)} + b_1^{(0)}\right)$$

$$= \sigma\left(\sum_{i=1}^{n} w_{1,i}a_i^{(0)} + b_1^{(0)}\right)$$

$$\begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_m^{(1)} \end{pmatrix} = \sigma\left[\begin{pmatrix} w_{1,0} & w_{1,1} & \ldots & w_{1,n} \\ w_{2,0} & w_{2,1} & \ldots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,0} & w_{m,1} & \ldots & w_{m,n} \end{pmatrix}\begin{pmatrix} a_1^{(0)} \\ a_2^{(0)} \\ \vdots \\ a_n^{(0)} \end{pmatrix} + \begin{pmatrix} b_1^{(0)} \\ b_2^{(0)} \\ \vdots \\ b_m^{(0)} \end{pmatrix}\right]$$

$$a^{(1)} = \sigma\left(\mathbf{W}^{(0)}a^{(0)} + \mathbf{b}^{(0)}\right)$$

Linear Algebra in the context of ANN III

✓ In a computer, the layers of the neural network are represented as **vectors**.

✓ Consider the input layer as X and the hidden layer as H.

✓ The above image shows the operations needed to compute the output for the first and the only hidden layer of the above neural network.

✓ Every single column of the network are vectors.Vectors are dynamic arrays that are a collection of data(or features).

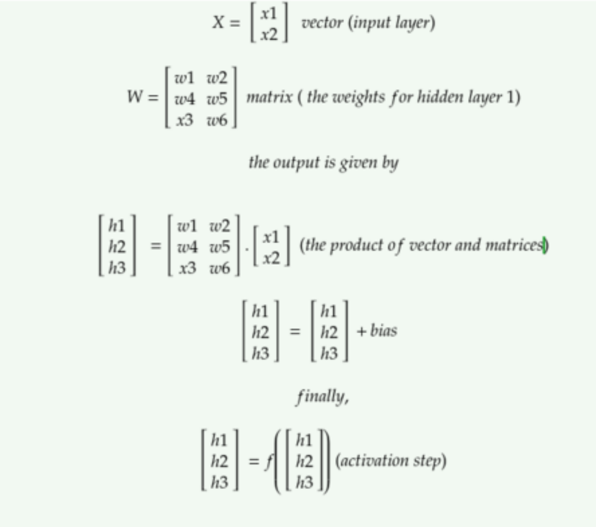
✓ In the current neural network, the vector **x** holds the input.

✓ The hidden layer H's output is calculated by performing $\mathbf{H} = f(\mathbf{Wx} + \mathbf{b})$.

✓ Here **W** is called as the Weight matrix, **b** is called bias and f is the activation function.

✓ The first component is **W.x**; this is a **matrix-vector product**, because **W** is a matrix and **x** is a vector.

Linear Algebra in the context of ANN IV



$$X = \begin{bmatrix} x1 \\ x2 \end{bmatrix} \quad \textit{vector (input layer)}$$

$$W = \begin{bmatrix} w1 & w2 \\ w4 & w5 \\ x3 & w6 \end{bmatrix} \quad \textit{matrix (the weights for hidden layer 1)}$$

the output is given by

$$\begin{bmatrix} h1 \\ h2 \\ h3 \end{bmatrix} = \begin{bmatrix} w1 & w2 \\ w4 & w5 \\ x3 & w6 \end{bmatrix} \cdot \begin{bmatrix} x1 \\ x2 \end{bmatrix} \quad \textit{(the product of vector and matrices)}$$

$$\begin{bmatrix} h1 \\ h2 \\ h3 \end{bmatrix} = \begin{bmatrix} h1 \\ h2 \\ h3 \end{bmatrix} + bias$$

finally,

$$\begin{bmatrix} h1 \\ h2 \\ h3 \end{bmatrix} = f\left(\begin{bmatrix} h1 \\ h2 \\ h3 \end{bmatrix}\right) \textit{(activation step)}$$

Outline

Basic Notations

Linear Algebra in the context of ANN

Scalars, Vectors, Matrices and Tensors

- Scalars
- Vectors
- Matrix
- Tensors

Eigenvalues & Eigenvectors

Singular Value Decomposition

Principal Component Analysis

Scalars I

Scalars: A scalar is just a single number, in contrast to most of the other objects studied in linear algebra, which are usually arrays of multiple numbers.

✓ Usually gives scalars lowercase italic variable names.

✓ When we introduce them, we specify what kind of number they are. For example, we might say “Let $s \in \mathbb{R}$ be the slope of the line,” while defining a real-valued scalar, or “Let $n \in \mathbb{N}$ be the number of units,” while defining a natural number scalar.

Vectors I

Vectors : A vector is an array of numbers. The numbers are arranged in order. We can identify each individual number by its index in that ordering.

✓ Typically we give vectors lowercase names in bold typeface, such as $\boldsymbol{x}$.

✓ The elements of the vector are identified by writing its name in italic typeface, with a subscript.

✓ The first element of $\boldsymbol{x}$ is x_1 , the second element is x_2 , and so on. We also need to say what kind of numbers are stored in the vector.

✓ If each element is in $\mathbb{R}$, and the vector has n elements, then the vector lies in the set formed by taking the Cartesian product of $\mathbb{R}_n$ times, denoted as $\mathbb{R}^n$.

✓ When we need to explicitly identify the elements of a vector, we write them as a column enclosed in square brackets:

Vectors II

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

✓ We can think of vectors as identifying points in space, with each element giving the coordinate along a different axis.

✓ Sometimes we need to index a set of elements of a vector. In this case, we define a set containing the indices and write the set as a subscript.

✓ For example, to access x_1 , x_3 and x_6 , we define the set $\mathbb{S} = \{1,\ 3,\ 6\}$ and write $x_{\mathbb{S}}$.

✓ We use the - sign to index the complement of a set.

✓ For example x_{-1} is the vector containing all elements of x except for x_1 , and $x_{-\mathbb{S}}$ is the vector containing all elements of x except for x_1 , x_3 and x_6.

Vectors III

A Basis for a Vector Space:

✓ Let V be a subspace of $\mathbb{R}^n$ for some n. A collection $B = \{v_1, v_2, \ldots, v_r\}$ of vectors from V is said to be a basis for V if B is linearly independent and spans V.

✓ If either one of these criterial is not satisfied, then the collection is not a basis for V. If a collection of vectors spans V, then it contains enough vectors so that every vector in V can be written as a linear combination of those in the collection.

✓ If the collection is linearly independent, then it doesn't contain so many vectors that some become dependent on the others.

Example 1: The collection $\{i, j\}$ is a basis for $\mathbb{R}^2$, since it spans $\mathbb{R}^2$ and the vectors i and j are linearly independent (because neither is a multiple of the other).

Vectors IV

✓ This is called the standard basis for $\mathbb{R}^2$. Similarly, the set $\{i, j, k\}$ is called the standard basis for $\mathbb{R}^3$, and, in general,

$$e_1 = (1, 0, 0, \ldots, 0) \tag{1}$$

$$e_2 = (0, 2, 0, \ldots, 0) \tag{2}$$

$$\ldots e_n = (0, 0, 0, \ldots, 1) \tag{3}$$

is the standard basis for $\mathbb{R}^2$.

Example 2: The collection $\{i, i + j, 2j\}$ is not a basis for $\mathbb{R}^2$. Although it spans $\mathbb{R}^2$, it is not linearly independent. No collection of 3 or more vectors from $\mathbb{R}^2$ can be independent.

Vectors V

Example 3: The collection $\{i + j, j + k\}$ is not a basis for $\mathbb{R}^3$. Although it is linearly independent, it does not span all of $\mathbb{R}^3$. For example, there exists no linear combination of $i + j$ and $j + k$ that equals $i + j + k$.

Matrix I

Matrices: A matrix is a 2-D array of numbers, so each element is identified by two indices instead of just one.

✓ We usually give matrices uppercase variable names with bold typeface, such as **A** . If a real-valued matrix A has a height of m and a width of n, then we say that $A \in \mathbb{R}^{m \times n}$.

✓ We usually identify the elements of a matrix using its name in italic but not bold font, and the indices are listed with separating commas.

✓ For example, $A_{1,1}$ is the upper left entry of **A** and $A_{m,n}$ is the bottom right entry.

✓ We can identify all the numbers with vertical coordinate i by writing a " : " for the horizontal coordinate. For example, $A_{i,:}$ denotes the horizontal cross section of **A** with vertical coordinate i. This is known as the $i-th$ row of **A**. Likewise, $\mathbf{A}$:, i is the $i-th$ **column** of **A**.

Matrix II

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}$$

✓ For example, $f(A)_{i,j}$ gives element (i,j) of the matrix computed by applying the function f to $\mathbf{A}$.

Transpose: One important operation on matrices is the transpose.

✓ The transpose of a matrix is the mirror image of the matrix across a diagonal line, called the main diagonal, running down and to the right, starting from its upper left corner.

$$(\mathbf{A}^{\mathbf{T}})_{\mathbf{i,j}} = \mathbf{A}_{\mathbf{j,i}} \tag{4}$$

✓ The transpose of a matrix product has a simple form:

$$(AB)^T = B^T A^T \tag{5}$$

Matrix III

The Rank of a Matrix:

✓ The maximum number of linearly independent rows in a matrix A is called the row rank of A, and the maximum number of linearly independent columns in A is called the column rank of A.

✓ If A is an m by n matrix, that is, if A has m rows and n columns, then :

$$row\ rank\ of\ A \leq m \tag{6}$$

$$column\ rank\ of\ A \leq n \tag{7}$$

$$rank(A_{m \times n}) \leq min(m, n) \tag{8}$$

✓ A vector r is said to be linearly independent of vectors r_1 and r_2 if it cannot be expressed as a linear combination of r_1 and r_2.

Matrix IV

$$r \neq ar_1 + br_2$$

$$\text{A=}\begin{bmatrix}1 & 3 & 7\\ 2 & 6 & 14\end{bmatrix}\ \text{B=}\begin{bmatrix}1 & 3 & 7\\ 2 & 6 & 14\end{bmatrix}\ \text{C=A=}\begin{bmatrix}1 & 3 & 7\\ 2 & 6 & 14\end{bmatrix}$$

- In matrix A, row r2 is a multiple of r1, r2 = 2 r1, so it has only one independent row. Rank(A) = 1
- In matrix B, row r3 is a sum of r1 and r2, r3 = r1 + r2, but r1 and r2 are independent. Rank(B) = 2
- In matrix C, all 3 rows are independent of each other. Rank(C) = 3

The rank of a matrix can be thought of as a representative of the amount of unique information represented by the matrix. Higher the rank, higher the information.

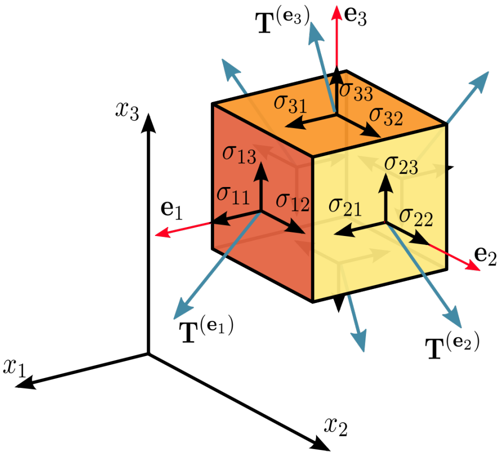
Tensors I

Tensors : In some cases we will need an array with more than two axes.

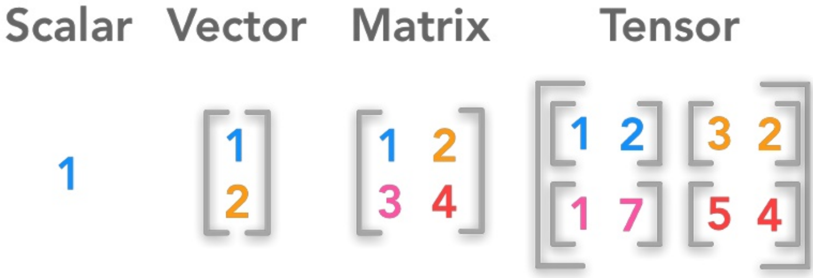
✓ In the general case, an array of numbers arranged on a regular grid with a variable number of axes is known as a tensor.

✓ We denote a tensor named "**A**" with this typeface: bfA. We identify the element of **A** at coordinates (i, j, k) by writing $\mathbf{A_{i,j,k}}$.

Tensors II


$\mathbf{T}^{(\mathbf{e}_3)}$
$\mathbf{e}_3$
σ_{33}
σ_{31}
σ_{32}
x_3
σ_{13}
σ_{23}
$\mathbf{e}_1$
σ_{11}
σ_{12}
σ_{21}
σ_{22}
$\mathbf{e}_2$
$\mathbf{T}^{(\mathbf{e}_1)}$
$\mathbf{T}^{(\mathbf{e}_2)}$
x_1
x_2

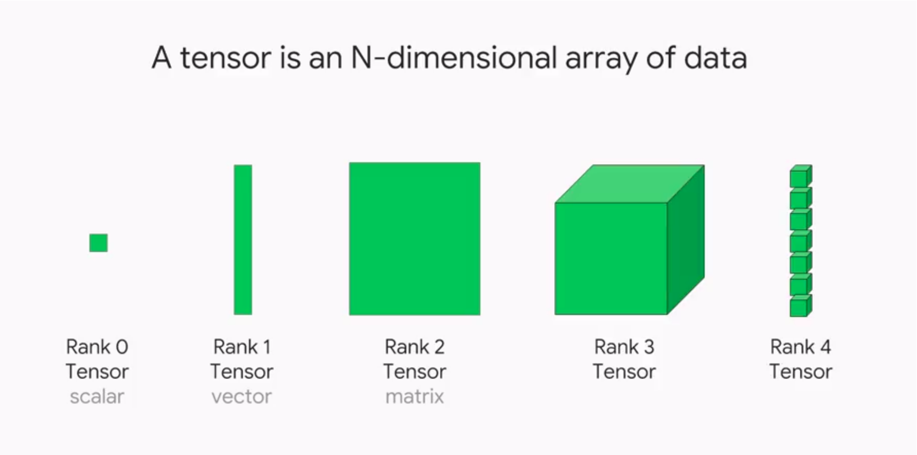
Tensors III



Scalar Vector Matrix Tensor

$$1 \qquad \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad \begin{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 4 \end{bmatrix} \end{bmatrix}$$

Tensors IV



A tensor is an N-dimensional array of data

Rank 0 Tensor
scalar

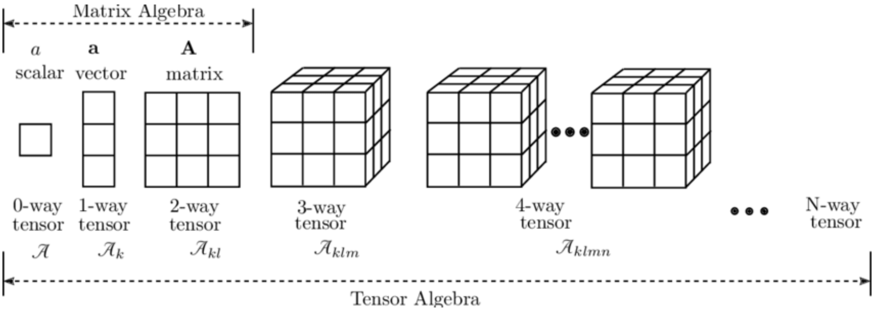
Rank 1 Tensor
vector

Rank 2 Tensor
matrix

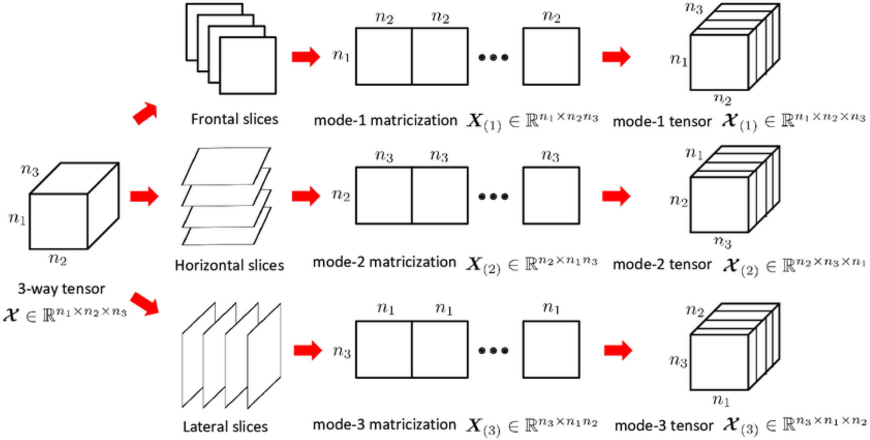
Rank 3 Tensor

Rank 4 Tensor

Tensors V


Matrix Algebra
a
scalar
$\mathbf{a}$
vector
$\mathbf{A}$
matrix
0-way tensor
$\mathcal{A}$
1-way tensor
$\mathcal{A}_k$
2-way tensor
$\mathcal{A}_{kl}$
3-way tensor
$\mathcal{A}_{klm}$
4-way tensor
$\mathcal{A}_{klmn}$
N-way tensor
Tensor Algebra

Tensors VI



3-way tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

Frontal slices

mode-1 matricization $\boldsymbol{X}_{(1)} \in \mathbb{R}^{n_1 \times n_2 n_3}$

mode-1 tensor $\mathcal{X}_{(1)} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

Horizontal slices

mode-2 matricization $\boldsymbol{X}_{(2)} \in \mathbb{R}^{n_2 \times n_1 n_3}$

mode-2 tensor $\mathcal{X}_{(2)} \in \mathbb{R}^{n_2 \times n_3 \times n_1}$

Lateral slices

mode-3 matricization $\boldsymbol{X}_{(3)} \in \mathbb{R}^{n_3 \times n_1 n_2}$

mode-3 tensor $\mathcal{X}_{(3)} \in \mathbb{R}^{n_3 \times n_1 \times n_2}$

Multiplying Matrices and Vectors I

$A_{m\times n}$ and $B_{n\times p}$, then C is shape of $C_{m\times p}$

$$C = ABC_{i,j} = \sum_{b} A_{i,j}B_{k,j} \tag{9}$$

$$C_{i,j} = \sum_{b} A_{i,j}B_{k,j} \tag{10}$$

✓ The standard product of two matrices called the element-wise product or Hadamard product, and denoted as $A \bigodot B$.

✓ The dot product between two vectors ***x*** and ***y*** of the same dimensionality is the matrix product $x^T y$.

$$\mathbf{x}.\mathbf{y} = x^T y \tag{11}$$

Multiplying Matrices and Vectors II

✓ Matrix product operations have many useful properties that make mathematical analysis of matrices more convenient. For example, matrix multiplication is distributive:

$$\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB}+\mathbf{AC} \quad (\mathbf{Dsitributive}) \tag{12}$$

$$\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C} \quad (\mathbf{Associative}) \tag{13}$$

✓ Matrix multiplication is not commutative (the condition AB = BA does not always hold), unlike scalar multiplication.

✓ However, the dot product between two vectors is commutative:

$$\boldsymbol{x}^T\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{x}. \tag{14}$$

System of linear equations I

$$A\boldsymbol{x} = \boldsymbol{b} \tag{15}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a known matrix,

$\boldsymbol{b} \in \mathbb{R}^{m}$ is a known vector, and

$\boldsymbol{x} \in \mathbb{R}^{n}$ is a vector of unknown variables that need to be solved for.

Each element x_i of $\boldsymbol{x}$ is one of these unknown variables.

Each row of $\mathbf{A}$ and each each element of $\boldsymbol{b}$ provide another constraint.

System of linear equations II

Equation (15) can be a form of as:

$$A1, : x = b_1 \tag{16}$$

$$A2, : x = b_2 \tag{17}$$

$$A3, : x = b_3 \tag{18}$$

$$\dots \tag{19}$$

$$A_m, : x = b_m \tag{20}$$

System of linear equations III

✓ More explicitly as:

$$A_{1,1}x_1 + A_{1,2}x_2 + \ldots A_{1,n}x_n = b_1 \tag{21}$$

$$A_{12,1}x_1 + A_{2,2}x_2 + \ldots A_{2,n}x_n = b_2 \tag{22}$$

$$\ldots \tag{23}$$

$$A_{m,1}x_1 + A_{m,2}x_2 + \ldots A_{m,n}x_n = b_m \tag{24}$$

✓ Matrix-vector product notation provides a more compact representation for equations of this form.

Identity and Inverse Matrices I

✓ Linear algebra offers a powerful tool called matrix inversion that enables us to analytically solve $Ax = b$(Equation (15)) for many values of $\mathbf{A}$.

✓ An identity matrix is a matrix that does not change any vector when we multiply that vector by that matrix.

✓ We denote the identity matrix that preserves n-dimensional vectors as I_n . Formally, $I_n \in \mathbb{R}^{n \times n}$, and

$$\forall x \in \mathbb{R}^n, I_n x = x \tag{25}$$

$$A^{-1}A = I_n \tag{26}$$

Identity and Inverse Matrices II

$$Ax = b \tag{27}$$

$$A^{-1}Ax = A^{-1}b \tag{28}$$

$$I_n x = A^{-1}b \tag{29}$$

$$x = A^{-1}b \tag{30}$$

✓ When A^{-1} exists, several different algorithms can find it in closed form.

✓ In theory, the same inverse matrix can then be used to solve the equation many times for different values of b .

✓ A^{-1} is primarily useful as a theoretical tool, however, and should not actually be used in practice for most software applications.

Identity and Inverse Matrices III

✓ Because A^{-1} can be represented with only limited precision on a digital computer, algorithms that make use of the value of b can usually obtain more accurate estimates of x.

Linear Combinations

Let $v_1, v_2, \ldots, v_r$ be vectors in $\mathbb{R}^n$. A linear combination of these vectors is any expression of the form:

$$k_1 v_1 + k_2 v_2, \ldots, k_r v_r \tag{31}$$

where the coefficients $k_1, k_2, \ldots, k_r$ are scalars.

Example : The vector $v = (-7, -6)$ is a linear combination of the vectors $v_1 = (-2, 3)$ and $v_2 = (1, 4)$, since $v = 2v_1 - 3v_2$.

Liner Combination, Dependence and Span II

The zero vector is also a linear combination of v_1 and v_2, since $0 = 0v_1 + 0v_2$. In fact, it is easy to see that the zero vector in $\mathbb{R}^n$ is always a linear combination of any collection of vectors $v_1, v_2, \dots, v_r$ from $\mathbb{R}^n$.

Span

✓ The set of all linear combinations of a collection of vectors $v_1, v_2, \dots, v_r$ from $\mathbb{R}^n$ is called the span of $\{v_1, v_2, \dots, v_r\}$.

✓ This set, denoted span $\{v_1, v_2, \dots, v_r\}$, is always a subspace of $\mathbb{R}^n$, since it is clearly closed under addition and scalar multiplication (because it contains all linear combinations of $v_1, v_2, \dots, v_r$). If $V = span\{v_1, v_2, \dots, v_r\}$, then V is said to be spanned by $v_1, v_2, \dots, v_r$.

✓ **span** of a set of vector is the set of all points obtained by linear combination of the original vectors.

Liner Combination, Dependence and Span III

Problem: Let v_1 = (1, 2, 0) , v_2 = (3, 1, 1), and W=(4, -7, 3). Determine whether W belongs to Span(v_1, v_2).

Answer: W= -5v_1 + 3v_2 $\in$ Span(v_1, v_2)

Outline

Basic Notations

Linear Algebra in the context of ANN

Scalars, Vectors, Matrices and Tensors

- Scalars
- Vectors
- Matrix
- Tensors

Eigenvalues & Eigenvectors

Singular Value Decomposition

Principal Component Analysis

Eigenvalues & Eigenvectors I

✓ Eigenvalues and eigenvectors feature prominently in the analysis of linear transformations.

✓ In linear algebra, an eigenvector characteristic vector of a linear transformation is a nonzero vector that changes at most by a scalar factor when that linear transformation is applied to it.

✓ The corresponding eigenvalue, often denoted by (λ), is the factor by which the eigenvector is scaled.

✓ Geometrically, an eigenvector, corresponding to a real nonzero eigenvalue, points in a direction in which it is stretched by the transformation and the eigenvalue is the factor by which it is stretched. If the eigenvalue is negative, the direction is reversed. Loosely speaking, in a multidimensional vector space, the eigenvector is not rotated.

✓ In other words, When a matrix is multiplied by one of its eigenvectors the output is the same eigenvector multiplied by a constant.

Eigenvalues & Eigenvectors II

✓ For example, λ may be negative, in which case the eigenvector reverses direction as part of the scaling, or it may be zero or **complex**.

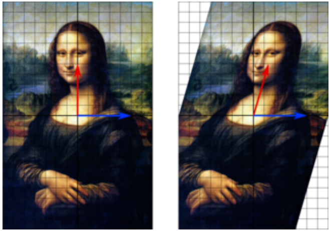


Figure 1: Matrix A acts by stretching the vector $\boldsymbol{x}$, not changing its direction, so $\boldsymbol{x}$ is an eigenvector of A.

Eigenvalues & Eigenvectors III

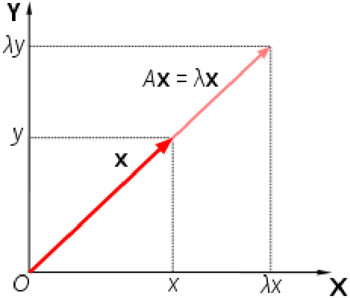


Figure 2: In this shear mapping the red arrow changes direction, but the blue arrow does not. The blue arrow is an eigenvector of this shear mapping because it does not change direction, and since its length is unchanged, its eigenvalue is 1.

Eigenvalues & Eigenvectors IV

The basic equation is

$$A\vec{\mathbf{v}} = \lambda\vec{\mathbf{v}} \tag{32}$$

✓ Left side is Matrix-vector multiplication, whereas right side is Scale multiplication.

The number λ is an eigenvalue of A.

$$A\vec{\mathbf{v}} = (\lambda I)\vec{\mathbf{v}} \tag{33}$$

✓ After multiplying by unit matrix, both the sides have matrix-vector multiplication.

$$A\vec{\mathbf{v}} - (\lambda I)\vec{\mathbf{v}} = 0 \tag{34}$$

$$(A - \lambda I)\vec{\mathbf{v}} = 0 \tag{35}$$

Eigenvalues & Eigenvectors V

Example : Determine the eigenvectors of the matrix

$A = \begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix}$

The eigenvalues of this matrix are λ = -1, -2. Therefore, there are nonzero vectors **v** such that A**v** =λ**v**

For λ=-1, eigenvector: A**v** =-1**v**= $\begin{bmatrix} 1-1 & -2 \\ 3 & -4-1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

For λ=-2, eigenvector: A**v** =-2**v**= $\begin{bmatrix} 1-2 & -2 \\ 3 & -4-2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Eigenvalues & Eigenvectors VI

- ✓ The trace of A, defined as the sum of its diagonal elements, is also the sum of all eigenvalues:

$$tr(A) = \sum_{i=1}^{n} a_{ij} = \sum_{i=1}^{n} \lambda_i = \lambda_1 + \lambda_2 + \cdots + \lambda_n. \tag{36}$$

- ✓ The determinant of A is the product of all its eigenvalues

$$det(A) = \prod_{i=1}^{n} \lambda_i = \lambda_1 \lambda_2 \ldots \lambda_n. \tag{37}$$

Outline

Basic Notations

Linear Algebra in the context of ANN

Scalars, Vectors, Matrices and Tensors

- Scalars
- Vectors
- Matrix
- Tensors

Eigenvalues & Eigenvectors

Singular Value Decomposition

Principal Component Analysis

Singular Value Decomposition(SVD) I

Singular Values

Let $A \in \mathcal{R}^{m \times n}$. Consider the matrix $A^T A$. It is a symmetric $n \times n$ matrx which is positive semi definite. The eigenvalues of $A^T A$ are :

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$$

Let,

$$\sigma_i = \sqrt{\lambda_i}$$

$$\rightarrow \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$$

The numbers $\sigma_1, \sigma_2, \ dots, \sigma_n$ are called singular values of A.

Singular Value Decomposition(SVD) II

A singular value decomposition of $m \times n$ matric A is a factorization as

$$A = U \sum V^T$$

Where,

- U is a $m \times m$ orthoganal matrix $(U^T = U^{-1})$.
- V is a $n \times n$ orthoganal matrix$(V^T = V^{-1})$.
- $\sum$ is a $m \times n$ matrix where i^{th} diagonal entry equalts to i^{th} singular value σ_i for $i = 1, 2, \ldots, r$, where

$$r = rank(A) \leq min(m, n)$$

- All other entries of $\sum$ are zero.

Singular Value Decomposition(SVD) III

Singular Vector:

✓ For any real or complex $m \times n$ matrix M the **left-singular vectors** of M are the eigenvectors of MM^t. They are equal to the columns of the matrix **u** in the singular value decomposition {**u,w,v**} of M.

✓ The right-singular vectors of M are the eigenvectors of the matrix **v** in the singular value decomposition of M.

✓ Singular value decomposition (SVD) is a matrix factorization method that generalizes the eigen decomposition of a square matrix $(n \times n)$ to any matrix $(n \times m)$.

✓ SVD is similar to Principal Component Analysis (PCA), but more general. PCA assumes that input square matrix, SVD doesn't have this assumption. General formula of SVD is:

$$M = U \sum V^t, \tag{38}$$

Singular Value Decomposition(SVD) IV

Or

$$M_{[m\times n]} = U_{[m\times r]} \sum_{[r\times r]} V^t_{[n\times r]}, \tag{39}$$

where:

- **M**- is original (input) matrix we want to decompose.
  Example: $m \times n$ matrix(e.g. m documents, n terms(words))
- **U**- is left singular matrix(columns are left singular vectors). **U** columns contain eigenvectors of matrix $\mathbf{MM}^t$.
  Example: $m \times r$ matrix (m documents, r concepts)

Singular Value Decomposition(SVD) V

- $\sum$ - is a diagonal matrix containing singular (eigen) values

  Example: $r \times r$ diagonal matrix (strength of each 'concepts')

  (r: rank of the matrix M)

- **V**- is right singular matrix (columns are right singular vectors). **V** columns contain eigenvector of matrix $\mathbf{MM}^t$.

  Example: $n \times r$ matrix(n terms, r concepts)

Singular Value Decomposition(SVD) VI

Example: Left-singular vectors of a 2×3 matrix

M= $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ u,w,v= SingularValueDecomposition[A];

MatrixForm[N[Transpose[u]]] = $\begin{bmatrix} 0.386318 & 0.922366 \\ -0.922366 & 0.386318 \end{bmatrix}$

MatrixForm [MM^t = M.Transpose[M]] = $\begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}$

MatrixForm[Eigenvectors[N[MM^t]]] = $\begin{bmatrix} 0.386318 & 0.922366 \\ -0.922366 & 0.386318 \end{bmatrix}$

✓ Right-singular vectors of a 2×3 matrix MatrixForm[M] = $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

{u,w,v} = SingularValueDecomposition[MM];

Singular Value Decomposition(SVD) VII

$$\text{MatrixForm[N[Transpose[v]]]} = \begin{bmatrix} 0.428667 & 0.566307 & 0.703947 \\ 0.805964 & 0.112382 & -0.581199 \\ 0.408248 & -0.816497 & 0.408248 \end{bmatrix}$$

$$\text{MatrixForm}[M^tM = \text{Transpose [M].M]} = \begin{bmatrix} 17 & 22 & 37 \\ 22 & 29 & 36 \\ 27 & 36 & 45 \end{bmatrix}$$

$$\text{MatrixForm [Eigenvectors[N}[M^tM]]] = \begin{bmatrix} 0.428667 & 0.566307 & 0.703947 \\ 0.805964 & 0.112382 & -0.581199 \\ 0.408248 & -0.816497 & 0.408248 \end{bmatrix}$$

Singular Value Decomposition(SVD) VIII

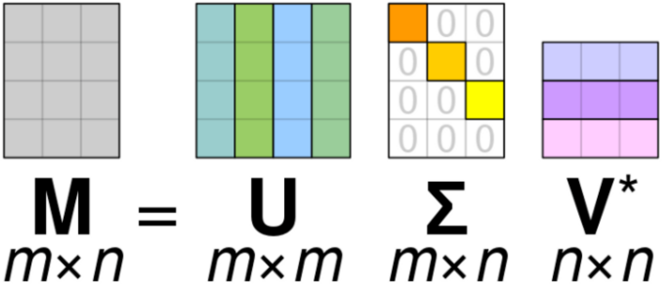
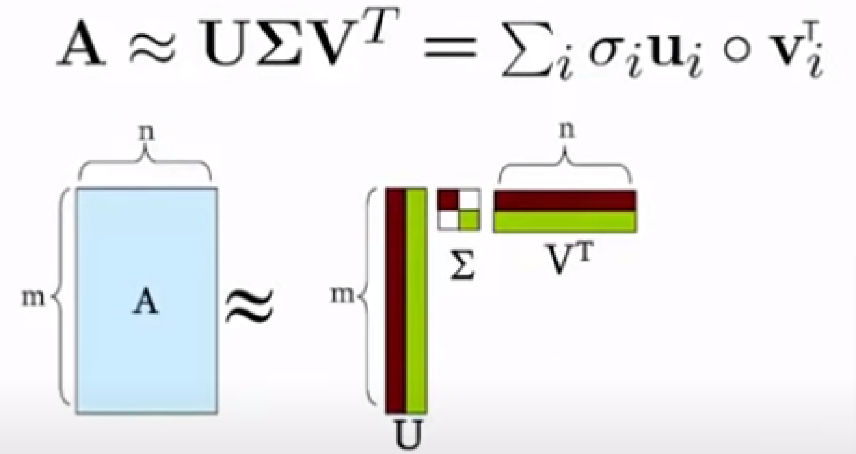


Figure 3: SVD matrices

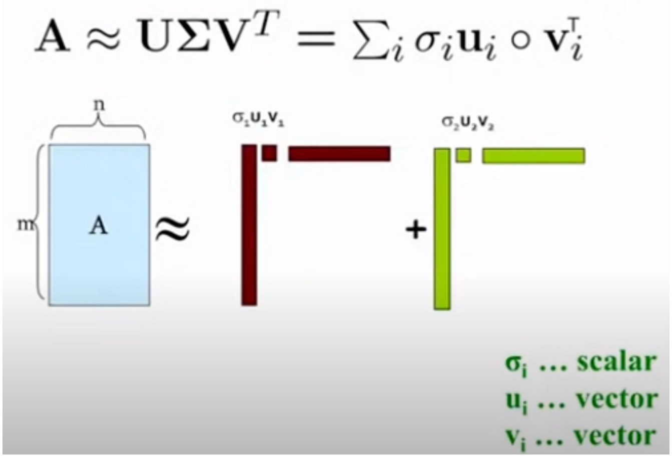
Singular Value Decomposition(SVD) IX



$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$

Figure 4: SVD matrices

Singular Value Decomposition(SVD) X

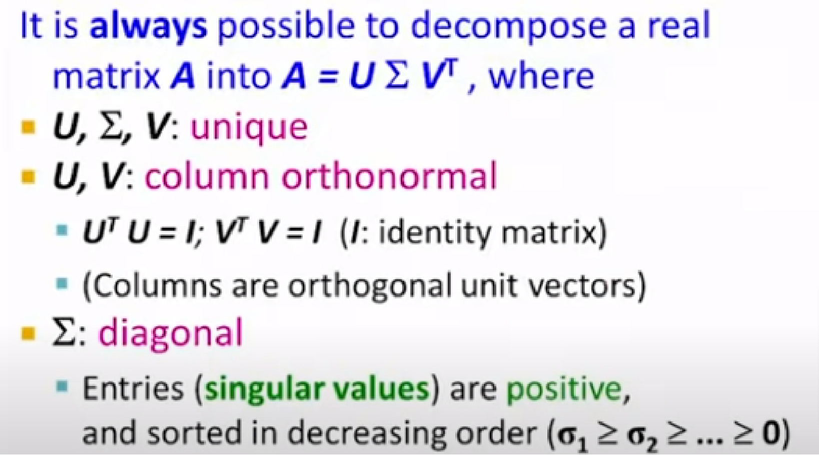


$$\mathbf{A} \approx \mathbf{U}\Sigma\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$

σ_i … scalar
u_i … vector
v_i … vector

Figure 5: SVD matrices

Singular Value Decomposition(SVD) XI

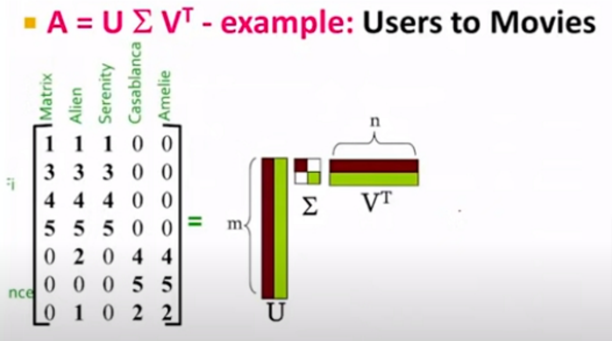


It is **always** possible to decompose a real matrix A into $A = U \Sigma V^T$, where

- U, Σ, V: unique
- U, V: column orthonormal
  - $U^T U = I; V^T V = I$ (I: identity matrix)
  - (Columns are orthogonal unit vectors)
- Σ: diagonal
  - Entries (**singular values**) are positive, and sorted in decreasing order ($\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$)

Figure 6: SVD properties

Singular Value Decomposition(SVD) XII



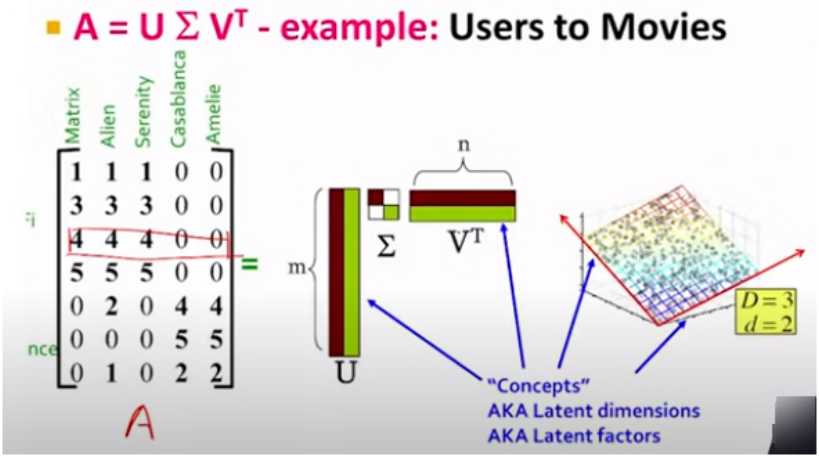
- $A = U\,\Sigma\,V^T$ - example: Users to Movies

$$\left[\begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\ 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{array}\right] = \underbrace{U}_{m}\;\Sigma\;\overbrace{V^T}^{n}$$

Figure 7: SVD-Example: User-to-Movies

✓ In the figure, values in the matrix A is likeness of users, i.e, lower value means less vote, whereas higher value means more votes, and zero means no votes.

Singular Value Decomposition(SVD) XIII



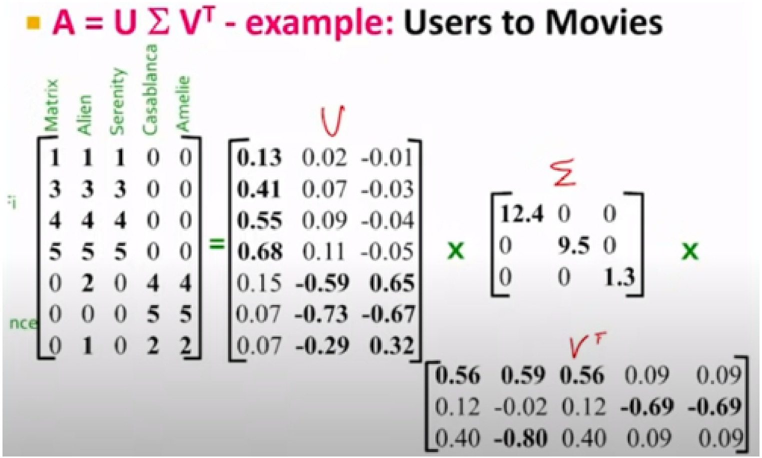
A = U Σ V^T - example: Users to Movies

$$\begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\ \hline 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{array} =$$

"Concepts"
AKA Latent dimensions
AKA Latent factors

Figure 8: SVD-Example: User-to-Movies

Singular Value Decomposition(SVD) XIV



- $A = U\, \Sigma\, V^T$ - example: Users to Movies

$$\begin{array}{c}\begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{array}\\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0\\ 3 & 3 & 3 & 0 & 0\\ 4 & 4 & 4 & 0 & 0\\ 5 & 5 & 5 & 0 & 0\\ 0 & 2 & 0 & 4 & 4\\ 0 & 0 & 0 & 5 & 5\\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}\end{array} = \overset{U}{\begin{bmatrix} 0.13 & 0.02 & -0.01\\ 0.41 & 0.07 & -0.03\\ 0.55 & 0.09 & -0.04\\ 0.68 & 0.11 & -0.05\\ 0.15 & -0.59 & 0.65\\ 0.07 & -0.73 & -0.67\\ 0.07 & -0.29 & 0.32 \end{bmatrix}} \times \overset{\Sigma}{\begin{bmatrix} 12.4 & 0 & 0\\ 0 & 9.5 & 0\\ 0 & 0 & 1.3 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09\\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69\\ 0.40 & -0.80 & 0.40 & 0.09 & 0.09 \end{bmatrix}}$$

Figure 9: SVD-Example: User-to-Movies

Singular Value Decomposition(SVD) XV

✓ Values in matrices U and V are strength of concepts.

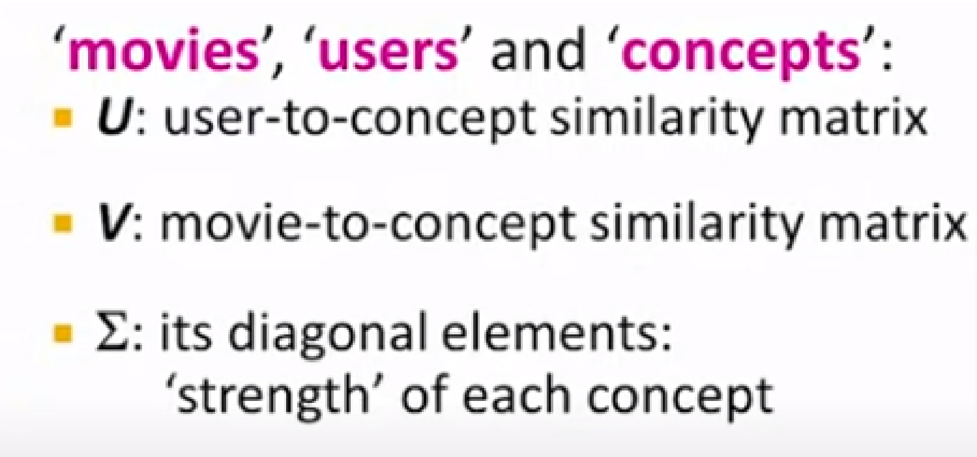
✓ U is "user-to-concept" similarity matrix,

✓ V is "movie-to-concept" similarity matrix

✓ In matrix U, columns first and second have two different concepts; SciFi(science fiction)-concept and Romance-concept, respectively.

✓ Similarly, in matrix V, rows first and second have two different concepts; SciFi(science fiction)-concept and Romance-concept, respectively.

✓ In matrices U and V, third column and third row, respectively, have very low strength.



'**movies**', '**users**' and '**concepts**':

- U: user-to-concept similarity matrix
- V: movie-to-concept similarity matrix
- Σ: its diagonal elements: 'strength' of each concept

Figure 10: SVD-Interpretation

Outline

Basic Notations

Linear Algebra in the context of ANN

Scalars, Vectors, Matrices and Tensors

- Scalars
- Vectors
- Matrix
- Tensors

Eigenvalues & Eigenvectors

Singular Value Decomposition

Principal Component Analysis

Principal Component Analysis(PCA) I

✓ It is about finding useful information in a data matrix.

✓ Principal component analysis or PCA in short is famously known as a dimensionality reduction technique.

✓ It has been around since 1901 and still used as a predominant dimensionality reduction method in machine learning and statistics. PCA is an unsupervised statistical method.

✓ Start by measuring m properties (m features) of n samples.

Example: A matrix could be grades in m courses for n students. A row for each course, a column for each student.

✓ from each row, subtract its average so the sample means are zero.

We look for a combination of courses and/or combination of students for which the data provides the most information.

Principal Component Analysis(PCA) II

✓ The information is "distance from randomness" and it is measured by **variance**. A large variance in course grade means greater information than a small variance.

Principal Component Analysis(PCA) III

✓ Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

✓ Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

The idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible.

Principal Component Analysis(PCA) IV

Why use PCA in the first place?

Suppose we have a dataset having two variables and 10 number of data points.

If we were asked to visualize the data points, we can do it very easily.

The result is very interpretable as well.

X1	2	8	1	4	22	15	25	29	4	2
X2	3	6	2	6	18	16	20	23	6	4

Principal Component Analysis(PCA) V

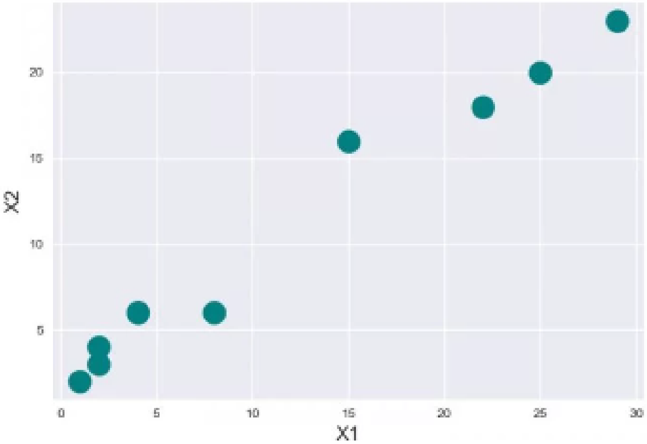


Figure 11: Plotting Data On Two Dimensions

Principal Component Analysis(PCA) VI

✓ Now if we try to increase the number of variables it gets almost impossible for us to imagine a dimension higher than three-dimensions.

✓ This problem we face when analyzing higher-dimensional datasets.

Principal Component analysis reduces high dimensional data to lower dimensions while capturing maximum variability of the dataset. Data visualization is the most common application of PCA. PCA is also used to make the training of an algorithm faster by reducing the number of dimensions of the data.

✓ We can think of PCA to be like fitting an *n-dimensional* ellipsoid to the data so that each axis of the ellipsoid represents a principal component. The larger the principal component axis the larger the variability in data it represents.

Principal Component Analysis(PCA) VII

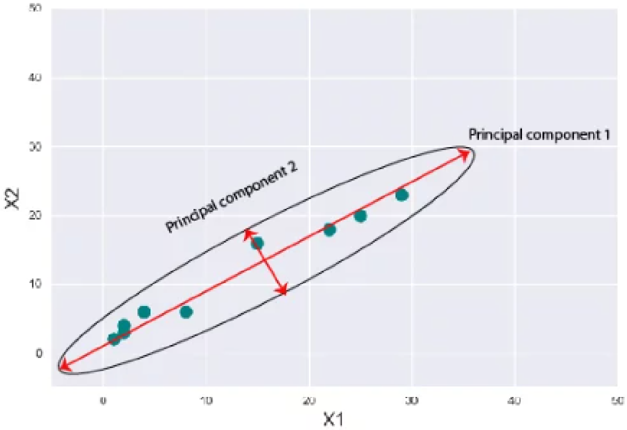


Figure 12: Fitting An Ellipse To Data

Principal Component Analysis(PCA) VIII

HOW DO WE DO A PCA?

1. Standardize the range of continuous initial variables
2. Compute the covariance matrix to identify correlations
3. Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components
4. Create a feature vector to decide which principal components to keep
5. Recast the data along the principal components axes

Principal Component Analysis(PCA) IX

1. **STANDARDIZATION**: The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.

   Example: If there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem.

   ✓ Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standard\ deviation}$$

Principal Component Analysis(PCA) X

2. COVARIANCE MATRIX COMPUTATION: The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other or to see if there is any relationship between them.

   ✓ Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

   ✓ The covariance matrix is a $p \times p$ symmetric matrix (where p is the number of dimensions) that has as entries the covariances associated with all possible pairs of the initial variables. For example, for a 3-dimensional data set with 3 variables x, y, and z, the covariance matrix is a 3×3 matrix of this from:

$$cov_{x,y} = \frac{\sum (x_i - \hat{x})(y_i - \hat{y})}{N - 1} \tag{40}$$

Principal Component Analysis(PCA) XI

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

✓ Since the covariance of a variable with itself is its variance ($Cov(a,a) = Var(a)$), in the main diagonal (Top left to bottom right) we actually have the variances of each initial variable. And since the covariance is commutative ($Cov(a,b) = Cov(b,a)$), the entries of the covariance matrix are symmetric with respect to the main diagonal, which means that the upper and the lower triangular portions are equal.

Principal Component Analysis(PCA) XII

What do the covariances that we have as entries of the matrix tell us about the correlations between the variables?

✓ It's actually the sign of the covariance that matters :

1. if positive then : the two variables increase or decrease together (correlated).
2. if negative then : One increases when the other decreases (Inversely correlated).

✓ The covariance matrix is not more than a table that summaries the correlations between all the possible pairs of variables.

Principal Component Analysis(PCA) XIII

3. COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS:

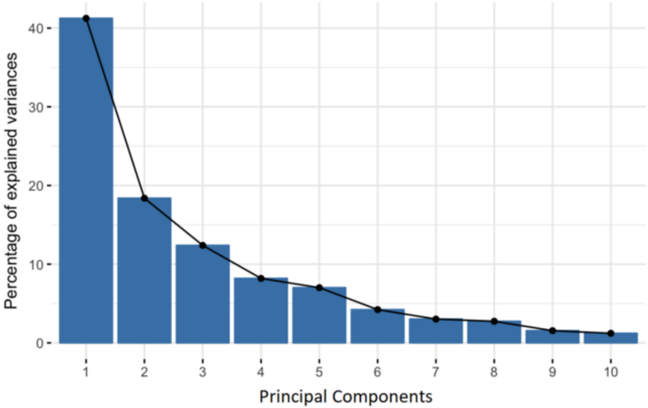
✓ Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data.

✓ Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables.

✓ These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components.

✓ So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on.

Principal Component Analysis(PCA) XIV



Principal Component Analysis(PCA) XV

✓ **Organizing information in principal components this way, will allow you to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables**.

✓ An important thing to realize here is that, the principal components are less interpretable and don't have any real meaning since they are constructed as linear combinations of the initial variables.

✓ Geometrically speaking, principal components represent the directions of the data that explain a **maximal amount of variance**, that is to say, the lines that capture most information of the data.

✓ The relationship between variance and information here, is that, the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more the information it has.

Principal Component Analysis(PCA) XVI

✓ **To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.**

How PCA Constructs the Principal Components

✓ As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set.

Principal Component Analysis(PCA) XVIII

✓ For example, let's assume that the scatter plot of our data set is as shown in Figure 13.

Can we guess the first principal component ?

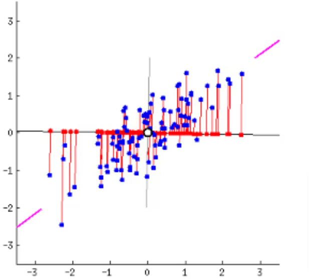


Figure 13: PCA Constructs the Principal Components

Principal Component Analysis(PCA) XIX

Answer: Yes, it’s approximately the line that matches the purple marks because it goes through the origin and it’s the line in which the projection of the points (red dots) is the most spread out. Or mathematically speaking, it’s the line that maximizes the variance (the average of the squared distances from the projected points (red dots) to the origin).

✓ The second principal component is calculated in the same way, with the condition that it is **uncorrelated with (i.e., perpendicular to) the first principal component** and that it accounts for the next highest variance. This continues until a total of p principal components have been calculated, equal to the original number of variables.

✓ Eigenvector and eigenvalue always come in pairs, so that every eigenvector has an eigenvalue. And their number is equal to the number of dimensions of the data.

✓ **For example**, for a 3-dimensional data set, there are 3 variables, therefore there are 3 eigenvectors with 3 corresponding eigenvalues.

Principal Component Analysis(PCA) XX

✓ *The eigenvectors of the Covariance matrix are actually the directions of the axes where there is the most variance(most information) and that we call Principal Components. And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component.*

By ranking the eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.

Principal Component Analysis(PCA) XXI

Example:

Let's suppose that our data set is 2-dimensional with 2 variables x,y and that the eigenvectors and eigenvalues of the covariance matrix are as follows:

$$v_1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \qquad \lambda_1 = 1.284028$$

$$v_1 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \qquad \lambda_2 = 0.04908323$$

✓ If we rank the eigenvalues in descending order, we get $\lambda_1 > \lambda_2$, which means that the eigenvector that corresponds to the first principal component (PC1) is v_1 and the one that corresponds to the second component (PC2) is v_2.

✓ After having the principal components, to compute the percentage of variance (information) accounted for by each component, we divide the eigenvalue of each component

Principal Component Analysis(PCA) XXII

by the sum of eigenvalues. If we apply this on the example above, we find that PC1 and PC2 carry respectively 96% and 4% of the variance of the data.

4. FEATURE VECTOR:

✓ As we saw in the previous step, computing the eigenvectors and ordering them by their eigenvalues in descending order, allow us to find the principal components in order of significance.

✓ In this step, to choose whether to keep **all these components or discard those of lesser significance** (of low eigenvalues), and form with the remaining ones a matrix of vectors is called **Feature vector**.

✓ The **Feature vector** is simply a matrix that has as columns the eigenvectors of the components that is decided to keep.

Principal Component Analysis(PCA) XXIII

✓ This makes it the first step towards **dimensionality reduction**, because if we choose to keep only p eigenvectors (components) out of n, the final data set will have only p dimensions.

Example: Continuing with the example from the previous step, we can either form a feature vector with both of the eigenvectors v_1 and v_2:

$$\begin{bmatrix} 0.6778736 & -0.7351785 \\ 0.7351785 & 0.6778736 \end{bmatrix}$$

Or discard the eigenvector v_2, which is the one of lesser significance, and form a feature vector with v_1 only:

$$\begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}$$

Principal Component Analysis(PCA) XXIV

✓ Discarding the eigenvector v_2 will reduce dimensionality by 1, and will consequently cause a loss of information in the final data set. But given that v_2 was carrying only 4% of the information, the loss will be therefore not important and we will still have 96% of the information that is carried by v_1.

It's up to you to choose whether to keep all the components or discard the ones of lesser significance, depending on what you are looking for.

Principal Component Analysis(PCA) XXV

5. RECAST THE DATA ALONG THE PRINCIPAL COMPONENTS AXES:

✓ In the previous steps, apart from standardization, we do not make any changes on the data, we just select the principal components and form the feature vector, but the input data set remains always in terms of the original axes (i.e, in terms of the initial variables).

✓ The aim of RECAST the data is to use the feature vector formed using the eigenvectors of the covariance matrix, to **reorient** the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis).

✓ This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

$$FinalDataSet =$$

$$FeatureVactor^T \times StandardizedOriginalDataSet^T$$

Principal Component Analysis(PCA) XXVI

Application of PCA

✓ **Gene expression data:** Determining the function of genes, and combinations of genes, is a central problem of genetics. Which genes combine to give which properties? Which genes malfunction to give which diseases?

✓ The understanding of genetic data (bioinformatics) has become a tremendous application of linear algebra.

Application of linear algebra I

1. **Dataset and Data Files:** In machine learning, you fit a model on a dataset.

   ✓ This is the table-like set of numbers where each row represents an observation and each column represents a feature of the observation.

   ✓ *Example:*

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
```

Application of linear algebra II

✓ This data is in fact a matrix: a key data structure in linear algebra.

✓ Further, when you split the data into inputs and outputs to fit a supervised machine learning model, such as the measurements and the flower species, you have a matrix (X) and a vector (y).

✓ The vector is another key data structure in linear algebra.

2. **Images and Photographs:**
3. **One-Hot Encoding:**
4. **Linear Regression:**
5. **Regularization:**
6. **Principal Component Analysis:**

Application of linear algebra III

7. **Singular-Value Decomposition:**

8. **Latent Semantic Analysis:**

9. **Recommender Systems:**

10. **Deep Learning:**