



# Deep Neural Networks

## Chapter 2: Linear Algebra for Deep Learning

## 2.1 Scalars, Vectors, Matrices, Tensors



- **Scalars:** a single number, e.g.  $x_1$
- **Vectors:**  $x$  is an array of numbers, written in *Euclid bold italics*.  
Special conventions:  
If  $S = \{1, 3, 6\}$  and  $n = 6$  then

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$$

$$\mathbf{x}_S = \begin{bmatrix} x_1 \\ x_3 \\ x_6 \end{bmatrix} \in \mathbb{R}^3 \quad \mathbf{x}_{-S} = \begin{bmatrix} x_2 \\ x_4 \\ x_5 \end{bmatrix} \in \mathbb{R}^3 \quad \mathbf{x}_{-1} = \begin{bmatrix} x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n-1}$$



- **Matrices**: a 2D array of numbers.

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \end{bmatrix} \in \mathbb{R}^{3 \times 2}$$

This matrix has 3 rows and 2 columns.

- $A_{i,:}$  denotes the row vector with row number  $i$ .
- $A_{:,j}$  denotes the column vector with column number  $j$ .
- $f(\mathbf{A})_{i,j}$  gives the element  $(i,j)$  of the matrix computed by applying the function  $f$  to  $A$ .
- **Tensors**: an  $n$ -dimensional array of numbers with  $n > 2$ .  
If  $\mathbf{A}$  is a 3D Tensor, then  $\mathbf{A}_{i,j,k}$  denotes an element at coordinates  $(i, j, k)$ .

- An important matrix operation is the transpose. It is the mirror image of the matrix on the diagonal line.

$$\text{If } \mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \end{bmatrix} \in \mathbb{R}^{3 \times 2}, \text{ then } \mathbf{A}^T = \begin{bmatrix} A_{1,1} & A_{2,1} & A_{3,1} \\ A_{1,2} & A_{2,2} & A_{3,2} \end{bmatrix} \in \mathbb{R}^{2 \times 3}$$

- In general it is defined  $(\mathbf{A}^T)_{i,j} = A_{j,i}$ .
- To turn a row vector  $x$  into a column vector  $y$  we write  $y = x^T$ .
- We can add two matrices, if they have the same shape, by adding their corresponding elements:

$$\mathbf{C} = \mathbf{A} + \mathbf{B}, \text{ where } C_{i,j} = A_{i,j} + B_{i,j}$$

## 2.2 Multiplying Matrices and Vectors



- The **matrix product** of  $A \in \mathbb{R}^{m \times n}$  and  $B \in \mathbb{R}^{n \times p}$  is a third matrix  $C \in \mathbb{R}^{m \times p}$ .  $C = A \cdot B = AB$

Note that the number of columns of  $A$  must match the number of rows of  $B$ .

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}$$

- The element-wise multiplication of two matrices  $A$  and  $B$  of identical dimensions is called the **Hadamard product**

$$D = A \odot B$$

- The **dot product**  $\langle x, y \rangle$  between two vectors  $x$  and  $y$  of the same dimension is the scalar  $x^T y$ .

- Matrix multiplication is distributive:

$$A(B + C) = AB + AC$$

- It is also associative:

$$A(BC) = (AB)C$$

- It is not commutative:

$$AB \neq BA \text{ in most cases}$$

- The dot product between two vectors is commutative:

$$x^T y = y^T x$$

- The transpose of a matrix has a simple form:

$$(AB)^T = B^T A^T$$

- The standard form of a linear equation system is

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is a known matrix,  $\mathbf{b} \in \mathbb{R}^m$  is a known vector, and  $\mathbf{x} \in \mathbb{R}^n$  is a vector of unknown variables for which we want to know the values.

- This is a short form of

$$A_{1,1}x_1 + \dots + A_{1,n}x_n = b_1$$

...

$$A_{m,1}x_1 + \dots + A_{m,n}x_n = b_m$$

or of

$$A_{1,:}x = b_1$$

...

$$A_{m,:}x = b_m$$

- There are several methods to solve these linear equation systems, like Gaussian elimination, LU decomposition, Cholesky decomposition, iterative methods.

## 2.3 Identity and Inverse Matrices

- Matrix inversion is a tool to analytically solve the matrix equation
- The identity matrix is a matrix that does not change any vector when we multiply it with the matrix

$$\forall \mathbf{x} \in \mathbb{R}^{n \times n}, \mathbf{I}_n \mathbf{x} = \mathbf{x}$$

- It consists of ones in the diagonal and zeroes everywhere else.
- The matrix inverse of  $\mathbf{A}$  is denoted as  $\mathbf{A}^{-1}$  and is defined as the matrix such that

$$\mathbf{I}_n = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$$

$$\mathbf{A}^{-1} \mathbf{A} = \mathbf{I}_n$$

- We can now solve the matrix equation 2.1 with the following steps:  $\mathbf{A} \mathbf{x} = \mathbf{b}$



$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

$$\mathbf{A}^{-1} \mathbf{A} \mathbf{x} = \mathbf{A}^{-1} \mathbf{b}$$

$$\mathbf{I}_n \mathbf{x} = \mathbf{A}^{-1} \mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{b}$$

- This derivation requires that  $\mathbf{A}^{-1}$  exists, which is not always the case.
- Usually a linear system of equations is solved more efficiently with other methods than computing  $\mathbf{A}^{-1}$ .

## 2.4 Linear Dependence and Span

- For  $A^{-1}$  to exist, the linear equation system  $A x = b$  must have exactly one solution for every value of  $b$ . The other alternatives are no solution or infinitely many solutions for a particular  $b$ .

- If both  $x$  and  $y$  are solutions of  $A x = b$ , then
$$z = \alpha x + (1 - \alpha)y$$

is also a solution for any real  $\alpha$ .

- The column vectors of  $A$  specify basis vectors pointing in different directions from the origin.

$$A x = \sum_i x_i A_{:,i}$$

- A linear combination of a set of vectors  $\{v^{(1)}, \dots, v^{(n)}\}$  is
$$\sum_i c_i v^{(i)}$$

- In this case the  $x_i$  correspond to the  $c_i$ , and the column vectors  $A_{:,i}$  correspond to the vectors  $v_i$ .
- The **span** (dt.: lineare Hülle) of a set of vectors is the set of all linear combinations of these vectors.
- Determining whether  $A x = b$  has a solution amounts to testing whether  $b$  is in the span of the columns of  $A$ . This span is also called the **column space**, or the **range** of  $A$ .
- In order for the system  $A x = b$  to have a solution for all values of  $b \in \mathbb{R}^m$ , the column space of  $A$  must be all  $\mathbb{R}^m$ . Therefore  $A$  must have at least  $m$  columns, i.e.  $n \geq m$ .



- A set of vectors is **linearly independent**, if no vector in the set is a linear combination of the other vectors.
- So for the column space of the matrix  $A$  to encompass all of  $\mathbb{R}^m$ , the matrix must contain at least  $m$  linearly independent columns.
- This ensures that  $A$  has a solution for all  $b \in \mathbb{R}^m$ .
- For the matrix  $A$  to have an inverse we also need to ensure that  $Ax = b$  has at most one solution for each  $b$ . To do so the matrix may have at most  $m$  columns.
- Together, this means that  $A$  must be square, i.e.  $m = n$ , and that all columns are independent.
- A square matrix with linearly dependent columns is called **singular**.

## 2.5 Norms

- If we need to measure the size of a vector we can use a **norm**.
- The  $L^p$  norm is defined as  $\|\mathbf{x}\|_p = \left( \sum_i |x_i|^p \right)^{\frac{1}{p}}$  for  $p \in \mathbb{R}, p \geq 1$ .
- Norms map vectors to non-negative values.
- A **norm** is a function satisfying the following 3 properties:
  - $f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = 0$
  - $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$  the triangle inequality
  - $\forall \alpha \in \mathbb{R}, f(\alpha \mathbf{x}) = |\alpha| f(\mathbf{x})$
- The most frequent norm is the  $L^2$  norm (**Euclidian norm**)

$$\|\mathbf{x}\|_2 = \|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$$

## 2.5 Norms

- The **squared  $L^2$  norm**  $\|\mathbf{x}\|^2 = \sum_i x_i^2$  is more convenient to deal with, especially with derivatives, as

$$\frac{\partial}{\partial x_j} \|\mathbf{x}\|^2 = \frac{\partial}{\partial x_j} \sum_i x_i^2 = 2x_j$$

- If the difference between small values and zero is important, we use the  **$L^1$  norm**:

$$\|\mathbf{x}\|_1 = \sum_i |x_i|$$

This norm adds up each small distance from 0.

- Another norm is the  **$L^\infty$  norm**, also called **max norm**.

$$\|\mathbf{x}\|_\infty = \max_i |x_i|$$

It computes the absolute value of the element of the largest magnitude in the vector:

- If we want to measure the size of a matrix, we can do this with the Frobenius norm:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$$

which is analogous to the  $L^2$  norm of a vector.

- The dot product can be written in terms of norms:

$$\mathbf{x}^T \mathbf{y} = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \cos \theta$$

However, the dot product itself is not a norm.

## 2.6 Special Kinds of Matrices and Vectors



- **Diagonal matrices** have nonzero entries only in their main diagonal.  $D$  is diagonal iff  $D_{i,j} = 0$  for all  $i \neq j$ .
- $\text{diag}(v)$  denotes a square diagonal matrix whose diagonal entries are given by the vector  $v$ .
- Multiplying a vector  $x$  with a diagonal matrix  $\text{diag}(v)$  is done by simply scaling each element  $x_i$  by  $v_i$ :
$$\text{diag}(v) \cdot x = v \odot x$$
- Inverting a square diagonal matrix is also efficient
$$\text{diag}(v)^{-1} = \text{diag}([1/v_1, \dots, 1/v_n]^T)$$
The inverse exists only if every  $v_i$  is nonzero.
- It is possible to construct non-square (rectangular) diagonal matrices, which have trailing columns or rows with 0s. The product  $D \cdot x$  is computed similarly.



- **Symmetric matrices** are any matrix that is equal to its transpose:

$$\mathbf{A} = \mathbf{A}^T$$

- For example, distance matrices are symmetric. If  $A_{ij}$  is the distance from point  $i$  to point  $j$ , then  $A_{i,j} = A_{j,i}$ , because the distance function is symmetric.
- A **unit vector** is a vector with unit norm:
$$\|\mathbf{x}\|_2 = 1$$
- Vectors  $\mathbf{x}$  and  $\mathbf{y}$  are **orthogonal** to each other, if  $\mathbf{x}^T \mathbf{y} = 0$ . If  $\mathbf{x}$  and  $\mathbf{y}$  are orthogonal and have nonzero norm, they have an angle of  $90^\circ$  to each other.

- An **orthogonal matrix** is a square matrix whose rows are mutually orthonormal (orthogonal and normal) and whose columns are mutually orthonormal:

$$\mathbf{A}^T \mathbf{A} = \mathbf{A} \mathbf{A}^T = \mathbf{I}$$

- This implies that

$$\mathbf{A}^{-1} = \mathbf{A}^T$$

- Orthogonal matrices are of interest because their inverse is very cheap to compute.
- Note that for a matrix to be orthogonal, the columns and rows need to be mutually orthonormal, not only orthogonal.

## 2.7 Eigendecomposition

- In **eigendecomposition** we decompose a matrix into a set of **eigenvectors** and **eigenvalues**.
- An **eigenvector** of a square matrix  $A$  is a nonzero vector  $v$  which has the property that multiplication by  $A$  only alters the scale of  $v$ .

$$Av = \lambda v$$

- The scalar  $\lambda$  is called the **eigenvalue** of  $v$ .
- If  $v$  is an eigenvector of  $A$ , then so is any rescaled vector  $s \cdot v$  for  $s \in \mathbb{R}, s \neq 0$ . Also,  $s \cdot v$  still has the same eigenvalue.
- Suppose that a matrix  $A$  has  $n$  linearly independent eigenvectors  $\{v^{(1)}, \dots, v^{(n)}\}$  with eigenvalues  $\{\lambda_1, \dots, \lambda_n\}$ .

## 2.7 Eigendecomposition

- We now concatenate all eigenvectors to form a matrix  $V$  with one eigenvector per column:  $V = [v^{(1)}, \dots, v^{(n)}]$ .  
We also concatenate all eigenvalues to form a vector  $\lambda$ :  
 $\lambda = [\lambda_1, \dots, \lambda_n]$  .
- Then the **eigendecomposition** of  $A$  is given by
$$A = V \cdot \text{diag}(\lambda) \cdot V^{-1}$$
- We often want to decompose matrices into their eigenvectors and eigenvalues.
- Not every matrix may be decomposed into eigenvectors and eigenvalues. Sometimes the decomposition involves complex numbers.

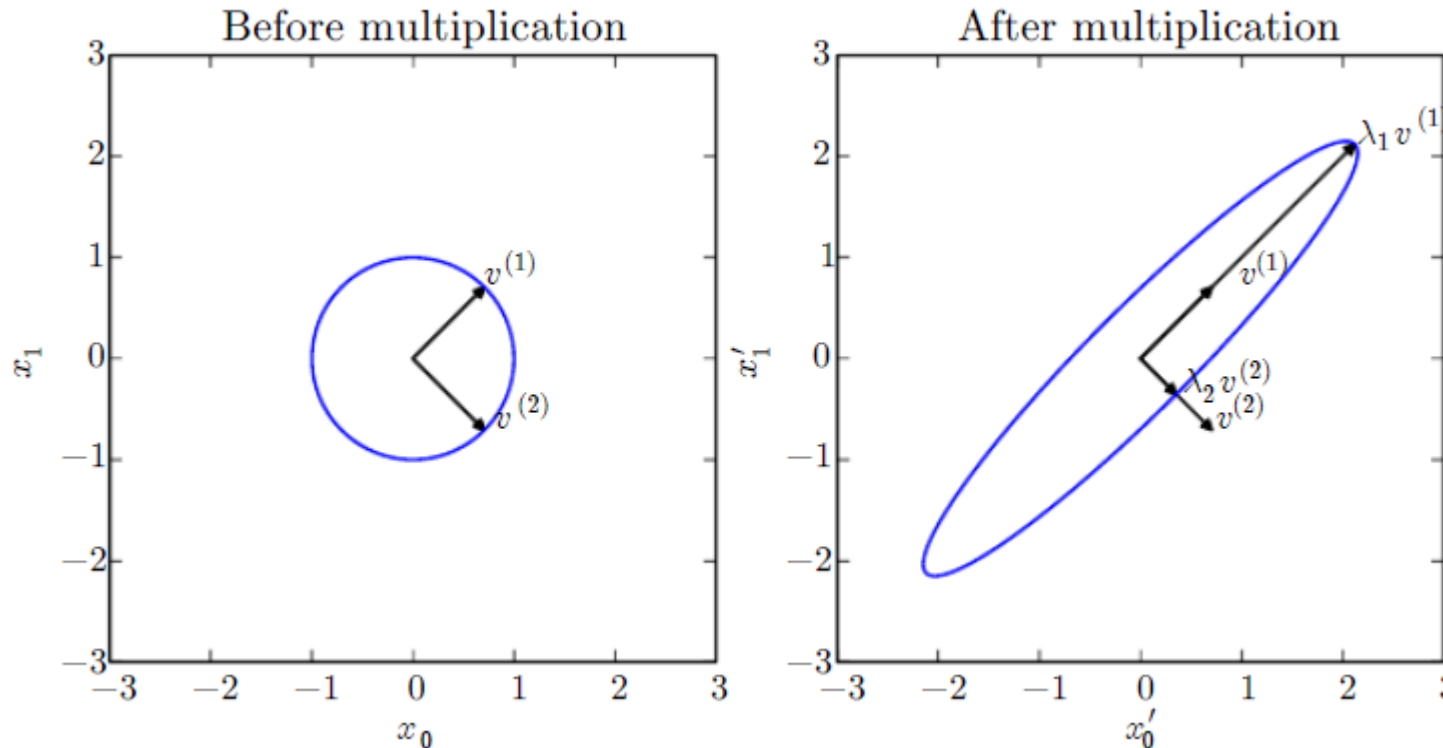
- Every real symmetric matrix  $A$  can be decomposed into matrices of real-valued eigenvectors and eigenvalues:

$$A = Q \cdot \Lambda \cdot Q^T$$

where  $Q$  is an orthogonal matrix composed of eigenvectors of  $A$ , and  $\Lambda$  is a diagonal matrix. The eigenvalue  $\Lambda_{i,i}$  is associated with column  $i$  of  $Q$ . Because  $Q$  is an orthogonal matrix, we can think of  $A$  as scaling the space by  $\lambda_i$  in the direction  $v^{(i)}$ .

- The eigendecomposition of a real symmetric matrix may not be unique. If any two or more eigenvectors share the same eigenvalue, then any set of orthogonal vectors lying in their span are also eigenvectors with that eigenvalue, and we could choose a  $Q$  with these eigenvectors instead.

# Effect of Eigenvectors & Eigenvalues



- Assume a matrix  $A$  with eigenvectors  $v^{(1)}$  and  $v^{(2)}$  with eigenvalues  $\lambda_1$  and  $\lambda_2$ . Left: the set of all unit vectors  $u \in \mathbb{R}^2$ . Right: the set of all points  $Au$ . We see that  $A$  scales the space in direction  $v^{(i)}$  by eigenvalue  $\lambda_i$ .

- A matrix is **singular** iff any of the eigenvalues are zero.
- A matrix is **positive definite** if all eigenvalues are positive.
- A matrix is **positive semidefinite** if all eigenvalues are positive or zero.
- A matrix is **negative definite** if all eigenvalues are negative.
- A matrix is **negative semidefinite** if all eigenvalues are negative or zero.
- Positive semidefinite matrices assure that  $\forall \mathbf{x}, \mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$
- Positive definite matrices additionally guarantee that
$$\mathbf{x}^T \mathbf{A} \mathbf{x} = 0 \Rightarrow \mathbf{x} = 0$$

## 2.8 Singular Value Decomposition

- The **singular value decomposition** is a way to factorize a matrix into singular vectors and singular values.
- Every real matrix has a singular value decomposition but not every real matrix has an eigenvalue decomposition.
- The eigendecomposition decomposes a matrix  $A$  into a Matrix  $V$  of eigenvectors and a vector  $\lambda$  of eigenvalues:

$$A = V \cdot \text{diag}(\lambda) \cdot V^{-1}$$

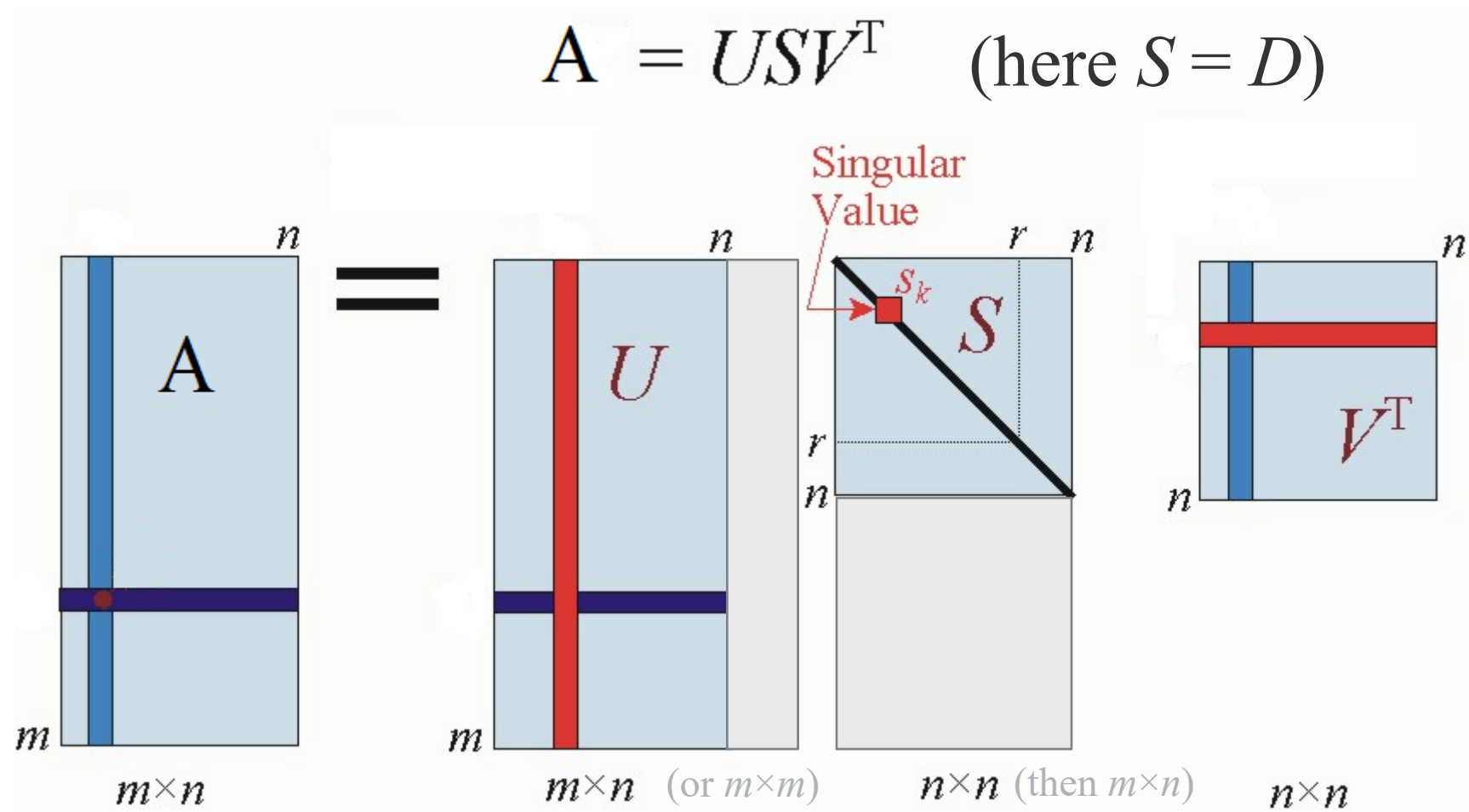
- The **singular value decomposition** decomposes  $A$  into three matrices:

$$A = U \cdot D \cdot V^T = U \cdot \Sigma \cdot V^T$$

- If  $A$  is an  $m \times n$  matrix then  $U$  is an  $m \times m$  matrix and  $V$  is an  $n \times n$  matrix.  $U$ ,  $D$  and  $V$  have a special structure.



# Singular Value Decomposition



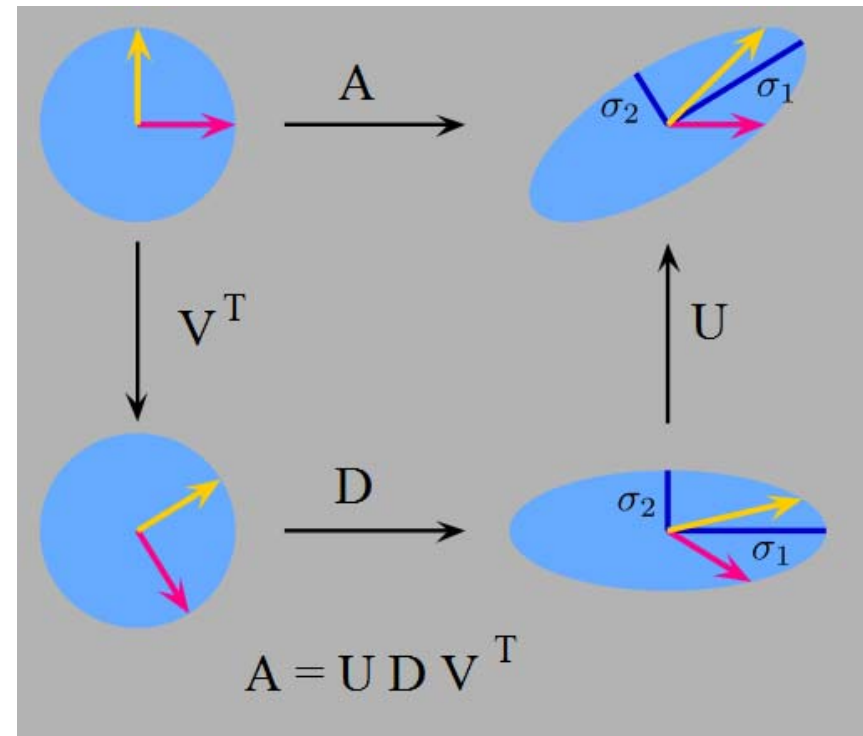


- $U$  and  $V$  are orthogonal matrices,  $D$  is a diagonal matrix, but is not necessarily square.
- The elements along the diagonal of  $D$  are known as the **singular values** of the matrix  $A$ .
- The columns of  $U$  are known as the **left-singular vectors**.
- The columns of  $V$  are known as **right-singular vectors**.
- The left-singular vectors of  $A$  are eigenvectors of  $AA^T$ .
- The right-sing. vectors of  $A$  are eigenvectors of  $A^T A$ .
- The non-zero singular values of  $A$  are the square roots of the eigenvalues of  $A^T A$ .
- We can use the singular value decomposition (SVD) to generalize matrix inversion to non-square matrices.

# Singular value decomposition



- First, we see the unit disc with the two unit vectors. The matrix  $A$  distorts the disk to an ellipse.
- The SVD decomposes  $A$  into 3 transformations: an initial rotation  $V^T$ , a scaling  $D$  along the coordinate axes, and a final rotation  $U$ .
- The lengths  $\sigma_1$  and  $\sigma_2$  of the semi-axes of the ellipse are the singular values of  $A$ , namely  $D_{1,1}$  and  $D_{2,2}$ .



<https://de.wikipedia.org/wiki/Datei:Singular-Value-Decomposition.svg>

## 2.9 The Moore-Penrose Pseudoinverse

- Matrix inversion is only defined for square matrices.
- Suppose we want to make a left-inverse  $B$  of a matrix  $A$ , so that we can solve a linear equation

$$A \cdot x = b$$

by left-multiplying each side to obtain

$$x = B \cdot y$$

- If  $A$  is taller than wide, there may be no solution.
- If  $A$  is wider than tall, there could be multiple solutions.
- The **Moore-Penrose pseudoinverse** is defined as

$$A^+ = \lim_{\alpha \rightarrow 0} (A^T A + \alpha I)^{-1} A^T$$

yielding in the limit

$$A^+ = (A^T A)^{-1} A^T$$



- Practical algorithms for the pseudoinverse are often based on the formula

$$\mathbf{A}^+ = \mathbf{V}\mathbf{D}^+\mathbf{U}^T$$

where  $\mathbf{U}$ ,  $\mathbf{D}$  and  $\mathbf{V}$  are the singular value decomposition of  $\mathbf{A}$ , and the pseudoinverse  $\mathbf{D}^+$  of a diagonal matrix  $\mathbf{D}$  is obtained by taking the reciprocal of its non-zero elements, then taking the transpose of this matrix.

- When  $\mathbf{A}$  has more columns than rows, the pseudoinverse provides the solution  $\mathbf{x} = \mathbf{A}^+\mathbf{y}$  with minimal Euclidean norm  $\|\mathbf{x}\|_2$  among all possible solutions.
- When  $\mathbf{A}$  has more rows than columns, the pseudoinverse gives us the  $\mathbf{x}$  for which  $\mathbf{Ax}$  is as close as possible to  $\mathbf{y}$  in terms of Euclidean norm  $\|\mathbf{Ax} - \mathbf{y}\|_2$ .

## 2.10 The Trace Operator

- The **trace operator** gives the sum of all of the diagonal entries of a matrix:

$$Tr(\mathbf{A}) = \sum_i A_{i,i}$$

- The trace operator provides an alternative way of writing the Frobenius norm of a matrix:

$$\|\mathbf{A}\|_F = \sqrt{Tr(\mathbf{A}\mathbf{A}^T)}$$

- The trace operator is invariant to the transpose operator

$$Tr(\mathbf{A}) = Tr(\mathbf{A}^T)$$

- The trace of a square matrix composed of many factors is also invariant to cyclic permutation, if the shapes of the corresponding matrices are suitable:

$$Tr(\mathbf{ABC}) = Tr(\mathbf{CAB}) = Tr(\mathbf{BCA})$$

- This invariance to cyclic permutation holds even if the resulting product has a different shape.

For example, for  $A \in \mathbb{R}^{m \times n}$  and  $B \in \mathbb{R}^{n \times m}$ , we have

$$\text{Tr}(AB) = \text{Tr}(BA)$$

even though  $AB \in \mathbb{R}^{m \times m}$  and  $BA \in \mathbb{R}^{n \times n}$ .

- A scalar is its own trace:  $\text{Tr}(a) = a$ .

## 2.11 The Determinant

- The **determinant** of a square matrix, denoted  $\det(A)$ , is equal to the product of all the eigenvalues of the matrix.
- The absolute value of the determinant can be thought of as a measure of how much the multiplication by the matrix expands or contracts space.
- If the determinant is 0, then space is contracted completely along at least one dimension, causing it to lose all of its volume.
- If the determinant is 1, then the transformation preserves volume.

$$\det\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

$$\det\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = 4 - 6 = -2$$

$$\det\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = 4 - 4 = 0$$