Tensors

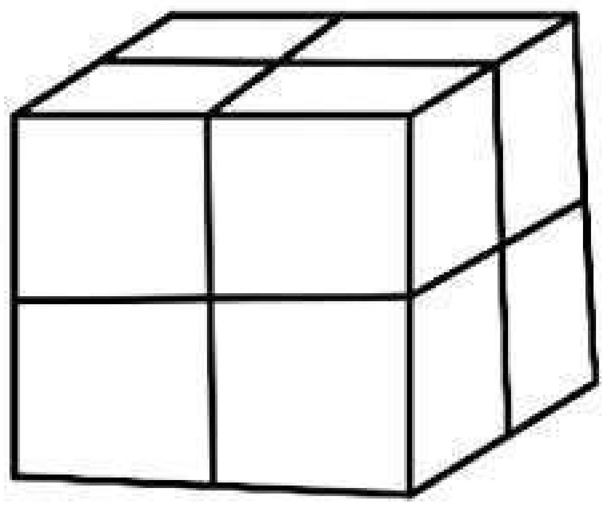
"ML generalization of vectors and matrices to any number of dimensions"

scalar x

vector $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$

matrix $\begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{bmatrix}$

3-tensor



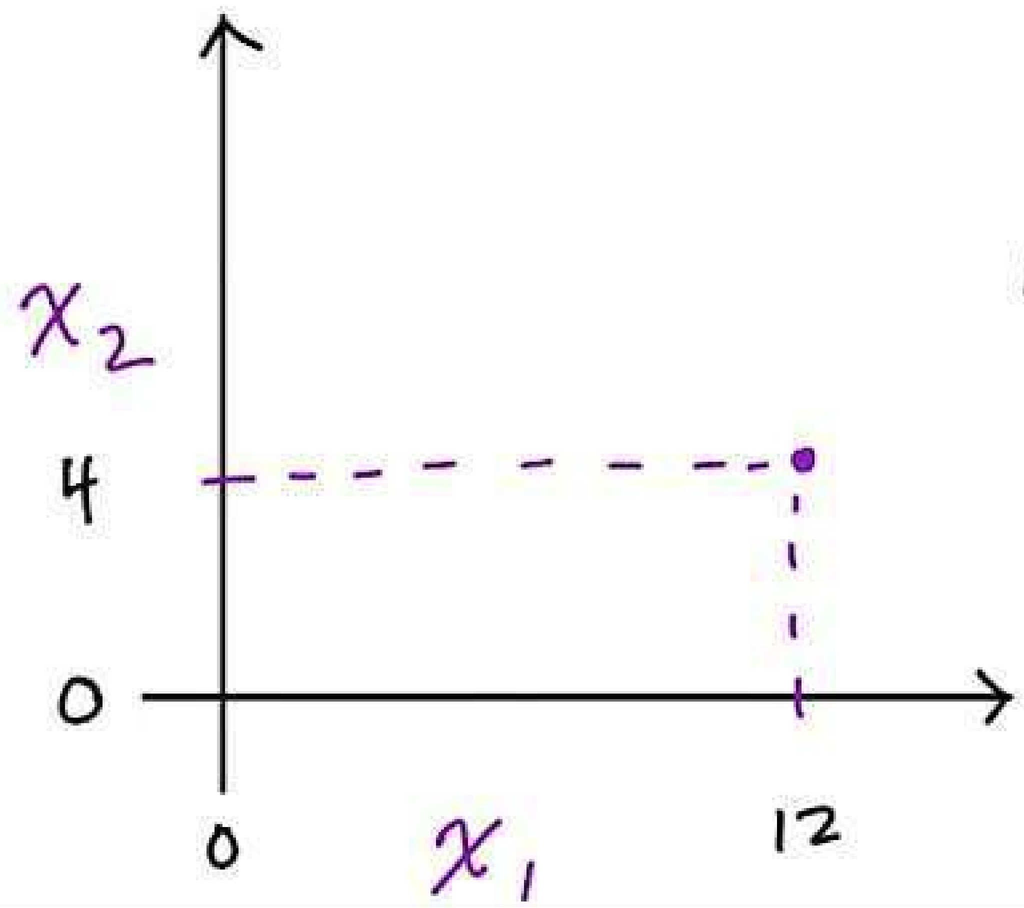
Dimensions	Mathematical Name	Description
0	scalar	magnitude only
1	vector	array
2	matrix	flat table, e.g., square
3	3-tensor	3D table, e.g., cube
n	*n*-tensor	higher dimensional

Scalars

- No dimensions
- Single number
- Denoted in lowercase, italics, e.g.: x
- Should be *typed*, like all other tensors: e.g., int, float32

Vectors

- One-dimensional array of numbers
- Denoted in lowercase, italics, bold, e.g.: $\boldsymbol{x}$
- Arranged in an order, so element can be accessed by its index
  - Elements are scalars so *not* bold, e.g., second element of $\boldsymbol{x}$ is x_2
- Representing a point in space:
  - Vector of length two represents location in 2D matrix (shown)
  - Length of three represents location in 3D cube
  - Length of *n* represents location in *n*-dimensional tensor



$[x_1 \;\; x_2] = [12 \;\; 4]$

Vector Transposition

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

row vector

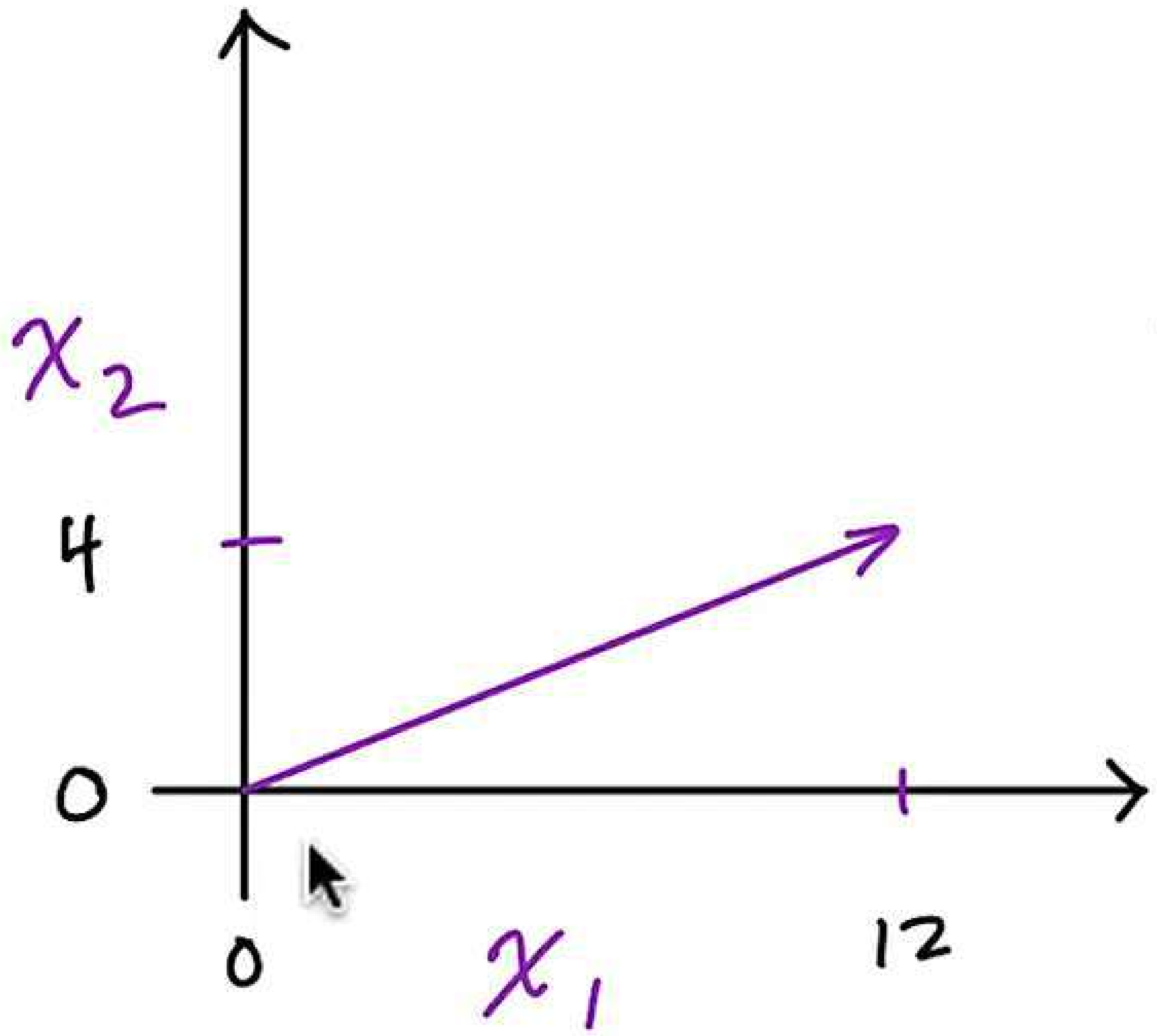
column vector

shape is (1, 3)

(3, 1)

Norms

Vectors represent a magnitude and direction from origin:



$[x_1 \quad x_2] = [12 \quad 4]$

Norms are functions that quantify vector magnitude.

L^2 Norm

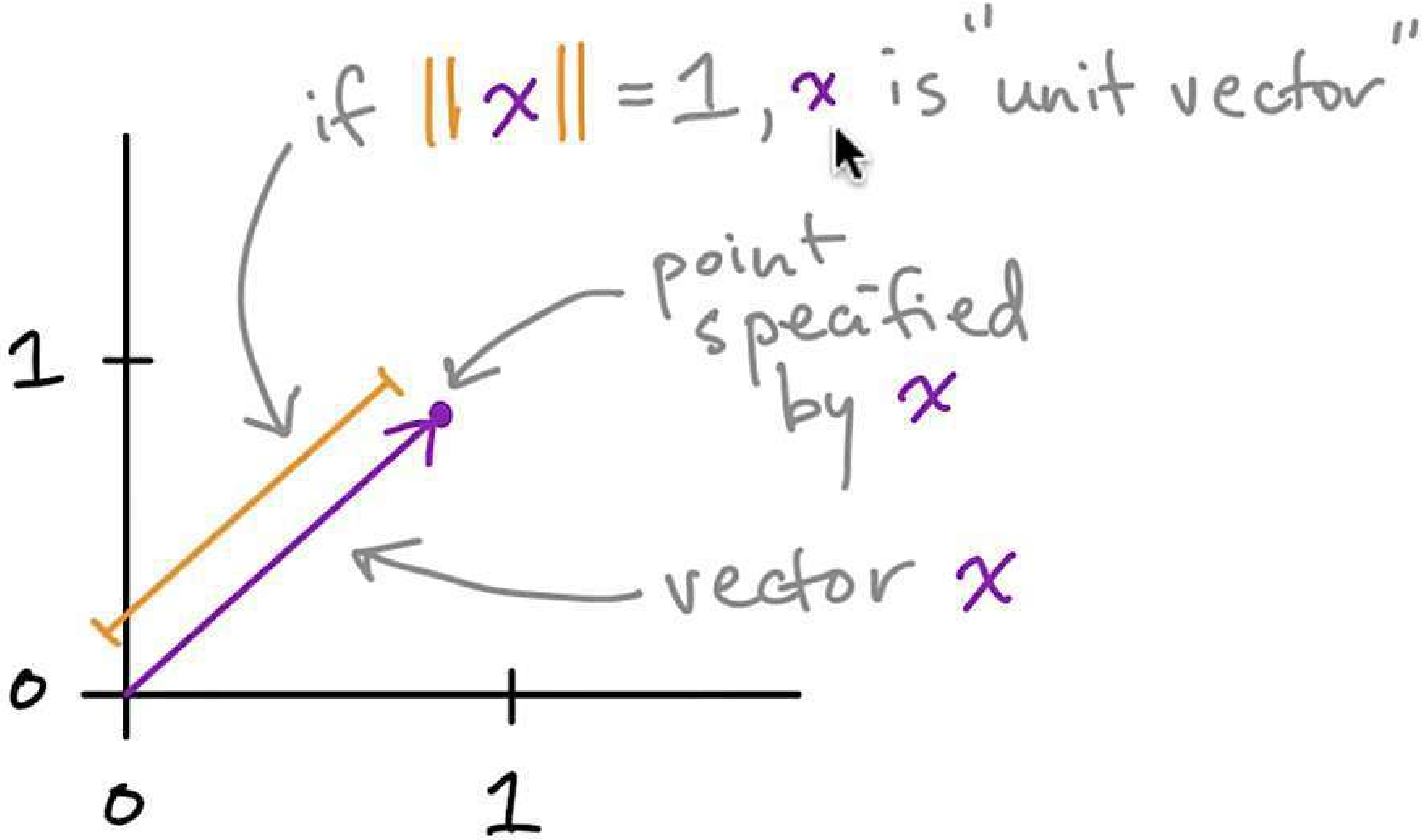
- Described by:

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

- Measures simple (Euclidean) distance from origin
- Most common norm in machine learning
  - Instead of $||\boldsymbol{x}||_2$, it can be denoted as $||\boldsymbol{x}||$

Unit Vectors

- Special case of vector where its length is equal to one
- Technically, $\boldsymbol{x}$ is a unit vector with "unit norm", i.e.: $||\boldsymbol{x}|| = 1$



L^1 Norm

- Described by:

$$\|x\|_1 = \sum_i |x_i|$$

- Another common norm in ML
- Varies linearly at all locations whether near or far from origin
- Used whenever difference between zero and non-zero is key

Squared L^2 Norm

- Described by: $$\|x\|_2^2 = \sum_i x_i^2$$

- Computationally cheaper to use than L^2 norm because:
  - Squared L^2 norm equals simply $\boldsymbol{x}^\mathrm{T}\boldsymbol{x}$
  - Derivative (used to train many ML algorithms) of element x requires that element alone, whereas L^2 norm requires $\boldsymbol{x}$ vector
- Downside is it grows slowly near origin so can't be used if distinguishing between zero and near-zero is important

Max Norm (or L^∞ Norm)

- Described by:

$$\|x\|_\infty = \max_i |x_i|$$

- Final norm we'll discuss; also occurs frequently in ML
- Returns the absolute value of the largest-magnitude element

Generalized L^p Norm

- Described by:

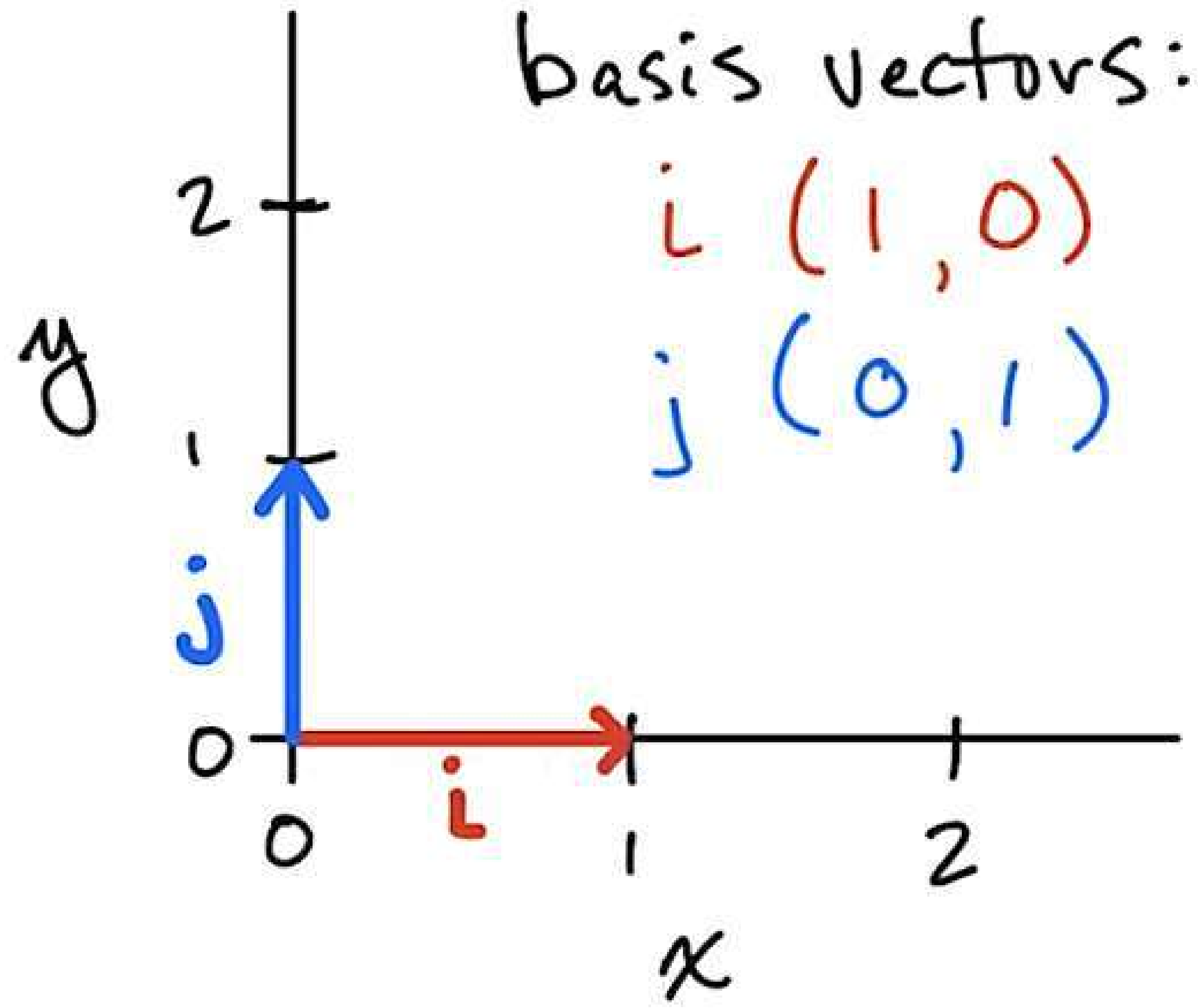
$$\|\boldsymbol{x}\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$$

- p must be:
  - A real number
  - Greater than or equal to one
- Can derive L^1, L^2, and L^∞ norm formulae by substituting for p
- Norms, particularly L^1 and L^2, used to regularize objective functions

Orthogonal Vectors

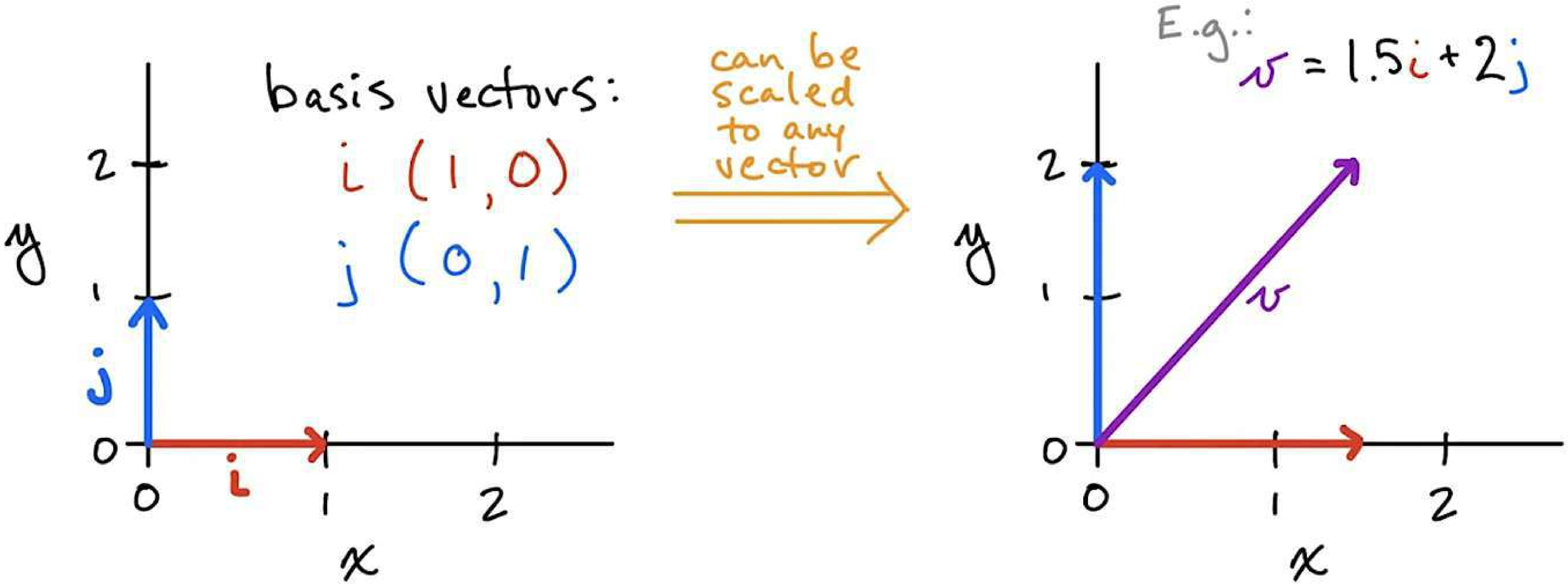
- $\boldsymbol{x}$ and $\boldsymbol{y}$ are orthogonal vectors if $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = 0$
- Are at 90° angle to each other (assuming non-zero norms)
- n-dimensional space has max n mutually orthogonal vectors (again, assuming non-zero norms)
- **Orthonormal** vectors are orthogonal *and* all have unit norm
    - Basis vectors are an example

Hands-on code demo



Basis Vectors

- Can be scaled to represent *any* vector in a given vector space
- Typically use unit vectors along axes of vector space (shown)



Exercises

1. What is the transpose of this vector?

$$\begin{bmatrix} 25 \\ 2 \\ -3 \\ -23 \end{bmatrix}$$

2. Using algebraic notation, what are the dimensions of this matrix ***Y***?

$$y = \begin{bmatrix} 42 & 4 & 7 & 99 \\ -99 & -3 & 17 & 22 \end{bmatrix}$$

3. Using algebraic notation, what is the position of the element in this matrix ***Y*** with the value of 17?

Generic Tensor Notation

- Upper-case, bold, italics, sans serif, e.g., $\boldsymbol{\mathsf{X}}$
- In a 4-tensor $\boldsymbol{\mathsf{X}}$, element at position (i, j, k, l) denoted as $\boldsymbol{\mathsf{X}}_{(i, j, k, l)}$

Segment 1: Data Structures for Algebra

- What Linear Algebra Is
- A Brief History of Algebra
- Tensors
- Scalars
- Vectors and Vector Transposition
- Norms and Unit Vectors
- Basis, Orthogonal, and Orthonormal Vectors
- Arrays in NumPy
- Matrices
- Tensors in TensorFlow and PyTorch

ML Foundations Series

Intro to Linear Algebra is **foundational for**:

1. **Intro to Linear Algebra**
2. **Linear Algebra II: Matrix Operations**
3. Calculus I: Limits & Derivatives
4. Calculus II: Partial Derivatives & Integrals
5. Probability & Information Theory
6. Intro to Statistics
7. Algorithms & Data Structures
8. **Optimization**

Intro to Linear Algebra

1. Data Structures for Algebra
2. **Common Tensor Operations**
3. Matrix Properties

Segment 2: Tensor Operations

- Tensor Transposition
- Basic Tensor Arithmetic
- Reduction
- The Dot Product
- Solving Linear Systems

Tensor Transposition

- Transpose of scalar is itself, e.g.: $x^{\mathrm{T}} = x$
- Transpose of vector, seen earlier, converts column to row (and vice versa)
- Scalar and vector transposition are special cases of **matrix transposition**:
  - Flip of axes over **main diagonal** such that:

$$(\boldsymbol{X}^{\mathrm{T}})_{i,j} = \boldsymbol{X}_{j,i}$$

$$\begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} x_{1,1} & x_{2,1} & x_{3,1} \\ x_{1,2} & x_{2,2} & x_{3,2} \end{bmatrix}$$

Exercises

1. What is $\boldsymbol{Y}^{\mathrm{T}}$?

$$Y = \begin{bmatrix} 42 & 4 & 7 & 99 \\ -99 & -3 & 17 & 22 \end{bmatrix}$$

2. What is the Hadamard product of these matrices?

$$\begin{bmatrix} 25 & 10 \\ -2 & 1 \end{bmatrix} \odot \begin{bmatrix} -1 & 7 \\ 10 & 8 \end{bmatrix}$$

3. What is the dot product of the tensors $\boldsymbol{w}$ and $\boldsymbol{x}$?

$$w = \begin{bmatrix} -1 & 2 & -2 \end{bmatrix}$$

$$x = \begin{bmatrix} 5 & 10 & 0 \end{bmatrix}$$