Chapter 1: Linear Algebra

Navid Seidi

Programmers House

August 2025

Why is Linear Algebra the Language of ML?

- **Data is a Matrix:** A dataset is a table of numbers. An image is a grid of pixel values. We call these grids **matrices**.
- **Features are Vectors:** A single data point (like one house) is a list of numbers. We call these lists **vectors**.
- **Models are Functions:** A neural network layer is a function that takes an input vector and creates an output vector. This is a **linear transformation**.
- **Training is Optimization:** To find the best model, we use calculus with matrices and vectors.

Topic 1: Scalars

Simple Definition: A single number.

Formal Definition: A scalar is an element of a field used to define a vector space. For us, this is usually a real number.

Intuitive Example: Think of a recipe. If the recipe calls for 2 cups of flour, the number '2' is a scalar. It scales the amount of the ingredient (flour).

Scalars in Python

```python
# Scalars are just regular variables in Python.
learning_rate = 0.01
temperature = 25.5

# We use scalars to scale vectors or matrices.
vector = [10, 20, 30]
scaled_vector = [learning_rate * x for x in vector]

print("Original vector:", vector)
print("Scaled vector:", scaled_vector)
```

Quiz: Scalars

Which of the following is a scalar?

1. A list of student ages: $[18, 19, 18]$
2. The price of a single movie ticket: \$12.50
3. A 2D image from a camera.
4. A student's scores in [Math, Science].

Quiz: Scalars

Which of the following is a scalar?

1. A list of student ages: $[18, 19, 18]$
2. The price of a single movie ticket: \$12.50
3. A 2D image from a camera.
4. A student's scores in [Math, Science].

Answer: (2). It is a single number.

Topic 2: Vectors

Simple Definition: A list of numbers in a specific order.

Formal Definition: A vector v in $\mathbb{R}^n$ is an ordered set of n real numbers, called components.

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Intuitive Example: A vector can represent a location on a map. To get from the start to point X, you go "3 units East" and "4 units North". The vector would be $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$.

Vectors in Python (NumPy)

```python
import numpy as np

# A vector representing a student's scores
scores = np.array([95, 88, 72])

# A vector representing house features [size_sqft,
    num_bedrooms]
house_features = np.array([1500, 3])

print("Student scores vector:", scores)
print("House features vector:", house_features)
```

Quiz: Vectors

You want to represent a single user's ratings for 3 movies: 'Movie A' (5 stars), 'Movie B' (3 stars), 'Movie C' (4 stars). How would you represent this as a vector?

1. 12
2. $\begin{bmatrix} 5 \\ 3 \\ 4 \end{bmatrix}$
3. $\begin{bmatrix} 5 & 3 & 4 \\ 5 & 3 & 4 \end{bmatrix}$
4. $5 + 3 + 4$

Quiz: Vectors

You want to represent a single user's ratings for 3 movies: 'Movie A' (5 stars), 'Movie B' (3 stars), 'Movie C' (4 stars). How would you represent this as a vector?

1. 12
2. $\begin{bmatrix} 5 \\ 3 \\ 4 \end{bmatrix}$
3. $\begin{bmatrix} 5 & 3 & 4 \\ 5 & 3 & 4 \end{bmatrix}$
4. $5 + 3 + 4$

Answer: (2). It's an ordered list of the ratings.

Topic 3: Dot Product

Simple Definition: A way to multiply two vectors to get a single number (a scalar). It measures how much they point in the same direction.

Formal Definition: For vectors $u, v \in \mathbb{R}^n$, the dot product is:

$$u \cdot v = \sum_{i=1}^{n} u_i v_i = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n$$

Intuitive Example: Imagine you are selling fruits. Your prices are in a vector $p = [\$2, \$1, \$3]$. A customer buys quantities in a vector $q = [5, 10, 2]$. The total cost is the dot product:
$p \cdot q = (2 \times 5) + (1 \times 10) + (3 \times 2) = \$26$.

Dot Product in Python

```python
import numpy as np

prices = np.array([2, 1, 3])       # Price per fruit
quantities = np.array([5, 10, 2]) # Quantity of each
    fruit bought

# Calculate the total cost using the dot product
total_cost = np.dot(prices, quantities)
# Or using the @ operator for dot product
total_cost_alt = prices @ quantities

print("Total cost:", total_cost)
print("Total cost (alt):", total_cost_alt)
```

Quiz: Dot Product

Given $u = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ and $v = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$, what is their dot product $u \cdot v$?

1. $\begin{bmatrix} 4 \\ 10 \\ 18 \end{bmatrix}$
2. 32
3. 21
4. It's not possible to compute.

Quiz: Dot Product

Given $u = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ and $v = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$, what is their dot product $u \cdot v$?

1. $\begin{bmatrix} 4 \\ 10 \\ 18 \end{bmatrix}$
2. 32
3. 21
4. It's not possible to compute.

Answer: (2). $(1 \times 4) + (2 \times 5) + (3 \times 6) = 4 + 10 + 18 = 32$.

Topic 4: Vector Norms (Length)

Simple Definition: A norm measures the length or size of a vector.

Formal Definition (L^2 Norm): The Euclidean norm, or L^2 norm, of a vector v is:

$$\|v\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2} = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

Intuitive Example: If your position vector is $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ (3 blocks East, 4 blocks North), the norm is the direct "as the crow flies" distance from the start. $\|v\|_2 = \sqrt{3^2 + 4^2} = 5$ blocks.

Vector Norms in Python

```
import numpy as np

position = np.array([3, 4])

# Calculate the L2 norm (Euclidean distance)
distance = np.linalg.norm(position)

print("Position vector:", position)
print("Direct distance (L2 Norm):", distance)

# L1 Norm (Manhattan distance)
manhattan_distance = np.linalg.norm(position, ord=1)
print("Manhattan distance (L1 Norm):",
    manhattan_distance)
```

Quiz: Vector Norms

What is the L^2 norm (length) of the vector $v = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$?

1. 0
2. 25
3. 5
4. $\sqrt{5}$

Quiz: Vector Norms

What is the L^2 norm (length) of the vector $v = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$?

1. 0
2. 25
3. 5
4. $\sqrt{5}$

Answer: (3). $\sqrt{0^2 + 5^2} = \sqrt{25} = 5$.

Simple Definition: A rectangular grid (or table) of numbers, arranged in rows and columns.

Formal Definition: A matrix $A \in \mathbb{R}^{m \times n}$ has m rows and n columns.

$$A = \begin{bmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \dots & A_{mn} \end{bmatrix}$$

Intuitive Example: A grayscale image is a matrix where each number is the brightness of a pixel. A 10×10 pixel image is a 10×10 matrix.

Matrices in Python

```python
import numpy as np

# A matrix representing a small grayscale image (3x3
    pixels)
# 0 = black, 255 = white
image_matrix = np.array([
    [0, 128, 255],
    [50, 100, 150],
    [200, 225, 0]
])

print("Image Matrix:\n", image_matrix)
print("Shape of matrix:", image_matrix.shape)
```

Quiz: Matrices

A dataset has 100 students (rows) and 5 features (columns) for each student (age, grade, height, etc.). What is the shape of the matrix representing this dataset?

1. 5×5
2. 100×100
3. 5×100
4. 100×5

Quiz: Matrices

A dataset has 100 students (rows) and 5 features (columns) for each student (age, grade, height, etc.). What is the shape of the matrix representing this dataset?

1. 5×5
2. 100×100
3. 5×100
4. 100×5

Answer: (4). The convention is (rows x columns), so (students x features).

Topic 6: Matrix-Vector Product

Simple Definition: Multiplying a matrix by a vector to create a new, transformed vector.

Formal Definition: If $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$, the product $y = Ax$ is a vector in $\mathbb{R}^m$ where each element y_i is the dot product of the i-th row of A with x.

Intuitive Example: A matrix can be a "transformation machine". If you have a point (vector) $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, a rotation matrix can turn it into a new point (vector) $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. This is how computer graphics work.

Matrix-Vector Product in Python

```python
import numpy as np

# A transformation matrix that rotates by 90 degrees
rotation_matrix = np.array([
    [0, -1],
    [1,  0]
])

# A vector representing a point
point = np.array([3, 1])

# Apply the transformation
transformed_point = rotation_matrix @ point

print("Original point:", point)
print("Transformed point:", transformed_point)
```

Quiz: Matrix-Vector Product

If a matrix A has shape 3×2 and a vector v has shape 2×1, what is the shape of the resulting vector Av?

1. 2×1
2. 3×2
3. 3×1
4. It will cause an error.

Quiz: Matrix-Vector Product

If a matrix A has shape 3×2 and a vector v has shape 2×1, what is the shape of the resulting vector Av?

1. 2×1
2. 3×2
3. 3×1
4. It will cause an error.

Answer: (3). The inner dimensions (2) match, and the result has the outer dimensions (3×1).

Simple Definition: An **eigenvector** is a special vector that does not change its direction when transformed by a matrix. It only gets scaled. The **eigenvalue** is the number it gets scaled by.

Formal Definition: For a square matrix A, a non-zero vector v is an eigenvector if $Av = \lambda v$, where λ is the scalar eigenvalue.

Intuitive Example: Imagine a spinning globe. The axis of rotation is an eigenvector. Points on the axis only get scaled (stay in place, so $\lambda = 1$), they don't change their direction relative to the globe.

Eigen-stuff in Python

```
import numpy as np

# A matrix that stretches space horizontally
A = np.array([
    [2, 0],
    [0, 1]
])

eigenvalues, eigenvectors = np.linalg.eig(A)

print("Eigenvalues:", eigenvalues)
print("Eigenvectors (columns):\n", eigenvectors)

# The eigenvector [1, 0] is scaled by its eigenvalue
    2.
# The eigenvector [0, 1] is scaled by its eigenvalue
    1.
```

Quiz: Eigenvectors

A matrix A transforms a vector $v = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ into the vector $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$. What can we say about v?

1. It is not an eigenvector of A.
2. It is an eigenvector of A with eigenvalue 3.
3. It is an eigenvector of A with eigenvalue 1.
4. It is an eigenvector of A with eigenvalue $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$.

Quiz: Eigenvectors

A matrix A transforms a vector $v = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ into the vector $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$. What can we say about v?

1. It is not an eigenvector of A.
2. It is an eigenvector of A with eigenvalue 3.
3. It is an eigenvector of A with eigenvalue 1.
4. It is an eigenvector of A with eigenvalue $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$.

Answer: (2). The direction is unchanged, and the vector is scaled by 3.

What You Now Know

- **Scalars:** Single numbers.
- **Vectors:** Lists of numbers for features/points.
- **Dot Product:** Combines vectors into a scalar.
- **Norms:** Measure the length of a vector.
- **Matrices:** Grids of numbers for data/transformations.
- **Matrix-Vector Product:** Transforms a vector.
- **Eigenvectors:** Special vectors whose direction is unchanged by a transformation.

This is the foundation for almost everything in ML!