Welcome to Math for Machine Learning!
Linear Algebra, Sessions 1–4, 8

Disclaimer: Work in progress. Portions of these written materials are incomplete.

Copyright 2020 by Google LLC. This work is licensed under the Creative Commons CC BY 4.0 License.

Tensors, Gradients and Operators

A Course in Linear Algebra and other Mathematical Tools with applications in ML and Quantum Computing

Topics Covered - part I

- Vectors and Vector Operations
- Tensors, Matrices, Vectors, Scalars
- Vector-Valued functions
- Norms
- Dot Product
- The Projection Operator
- Linear Transformations & Operators
- Basic Operations with Matrices
- How to Build a Vector Space
  - Sets, Groups, Rings, Fields, Vector Spaces, Algebras
- Linear Independence and Basis Vectors
- Subspaces
- Span and Dim
- Gram-Schmidt Process
- The Identity Matrix
- Determinants
- Matrix Inversion
- Tensor Product
- Solving Systems of Linear Equations
- Linear Transformations as Matrices
- Eigenvectors
- Singular Value Decomposition
- Rank, Kernel, Rank-Nullity Theorem

Topics Covered -- part II

- Image, Injection, Surjection, Bijection
- Triangle Inequality
- Cross Product
- Hermitian Operators
- Jacobian
- Hessian
- SparseML
- Linear Algebra and Deep Learning
- Inner Products, Hilbert Space, Dirac
- Linear Algebra and Quantum Mechanics

What Is Linear Algebra?

Linear algebra is critical to machine learning, quantum mechanics and many other fields. You probably have come across some elements of linear algebra, such as vectors and matrices, in other work. In this course, we will treat these objects more rigorously and figure out how to apply them.

A formal way of describing linear algebra is the branch of mathematics concerning vector spaces and linear mappings between such spaces. It includes the study of lines, planes, and subspaces, but is also concerned with properties common to all vector spaces.

Course Goals

- Provide a more rigorous foundation for linear algebra in particular and higher mathematics more generally
- Provide a deeper *intuition* for the concepts of linear algebra
  - Example: why is the determinant computed the way that it is - and what is the geometric implication of the det?
- Provide hands-on experience in how linear algebra is used in machine learning and other applications
- *Bonus material:* Linear algebra for quantum mechanics and computing - Dirac notation, unitary operators and more

Which mathematical object is this?

$1. \forall a, b \in F, a + b \in F$

$2. \forall a, b, c \in F, (a + b) + c = a + (b + c)$

$3. \exists 0 \in F \forall 0 + a = a + 0 = a$

$4. \forall a \in F, \exists (-a) \in F a + -a = -a + a = 0$

$5. \forall a, b \in F, a + b = b + a$

$6. \forall a, b \in F, a \times b \in F$

$7. \forall a, b, c \in F, (a \times b) \times c = a \times (b \times c)$

$8. \exists 1 \in F, 1 \times a = a \times 1 = a$

$9. \forall a, b, c, \in F, a \times (b + c) = a \times b + a \times c \& \forall a, b, c \in F, (a + b) \times c = a \times c + b \times c$

$10. \forall a, b \in F, a \times b = b \times a$

$11. \forall a \neq 0 \exists a^{-1} \in F \mathrm{a}^{-1} \times a = a \times a^{-1} = 1$

Is this linear?

$$Y(x) = mx + b$$

where b $\neq$ 0

Is this the Pythagorean Theorem?

$$||v-u||^2 = ||v||^2 - 2||v||||u||\cos\,(\theta) + ||u||^2$$

Does this Product involve Multiplication?

$$\mathbb{R}^2 := \mathbb{R} \times \mathbb{R} = \{(a, b) : a \in \mathbb{R}, b \in \mathbb{R}\}$$

Is this the only kind of Binary?

Some Sets of Interest in Linear Algebra

$\mathbb{Z}$ denotes the set of integers, all whole numbers, like ..., -2,-1,0,1,2, etc.

$\mathbb{Q}$ denotes the set of *ratio*nal numbers, i.e. all *ratios* of integers, e.g. ½, ¾, -⅞, etc.

$\mathbb{R}$ denotes the set of all real numbers, i.e. π , e , $\sqrt{2}$, $6.4\bar{4}$, 3.0, etc.

$\mathbb{C}$ denotes the set of complex numbers, i.e. i, 1+i, 3-7i, etc.

Vectors and Vector Operations

What is a Vector?

To start our discussion, we can think of a vector in the following ways:

- An ordered collection of numbers (a 1d array)
- A mathematical object having a magnitude (length) and direction
- Can be represented geometrically as a ray, usually starts at the origin (though the start position can be changed); *standard position* is to start at the origin
- A vector is denoted as a lowercase **bold** letter, occasionally with an arrow

$$\vec{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \qquad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ v_n \end{pmatrix}$$

Polar Coordinates

R^2: Instead of using cartesian coordinates to specify the terminal point of a vector, we can also use polar coordinates. We would specify a radius r and and an angle θ to determine the vector.

The radius r is the distance from the central point and θ is the angle from the x-axis.

Cylindrical Coordinates

To specify cylindrical coordinates, specify

the angle theta “on the floor of the cylinder”,

then the radius r (distance from the origin),

and finally, the height z “off the floor”.

Spherical Coordinates

To specify spherical coordinates,

specify an angle theta “on the floor”, then

an angle phi “off of the z-axis”,

and finally, a distance rho from the origin.

Vectors with Real Components: Examples

$$\begin{bmatrix} 1 \\ 2 \\ \vdots \\ 10 \end{bmatrix} \quad \begin{bmatrix} \sqrt{2} \\ e \\ \pi \end{bmatrix}$$

$$\mathbb{R}$$

Vectors with Complex Components: Examples

$$\begin{bmatrix} \log(-1) \\ 1+i \\ e^{i\frac{\pi}{2}} \end{bmatrix} \qquad \begin{bmatrix} 2+3i \\ 1-6i \\ 5+7i \end{bmatrix}$$

$$\mathbb{C}$$

The Gradient as a Vector

The gradient of a vector-valued function $f:R^n$ to R is the vector of the partial derivatives of f, i.e.

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

The gradient of a function f at a point encodes the direction of steepest ascent of the values of f from that point. The opposite direction is the direction of steepest descent. Understanding the gradient is crucial to gradient descent techniques in machine learning.

Deep Learning: Gradient Descent

An example of the Gradient of a Function

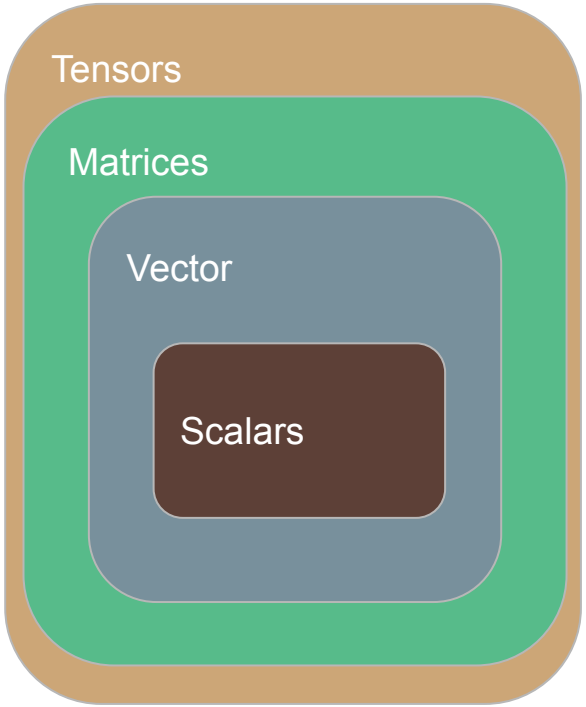
Let f be the function from R^2 to R given by $f(x,y)=x^2-y^2$. Then, the gradient of f is the following vector:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right) = (2x, -2y)$$

So, at the point (1,2), f will increase most rapidly in the direction of the gradient vector evaluated at (1,2): (2*1, -2*2) = (2,-4). So for gradient descent we would flip this to the point (-2,4)

What is a Tensor?

- Tensors are multi-indexed arrays of numbers, or functions, that transform according to certain rules under a change of coordinates.
- Tensor orders:
  - 0-tensor: scalar
  - 1-tensor: vector
  - 2-tensor: matrix
  - 3-tensor: a 3d cube of numbers
  - *n*-tensor....



5

Scalar

1x1 matrix

Vector Operations

- Vector Addition
- Multiplication by a Scalar
- Vector Subtraction

Addition

- Addition: add the components of each vector. Note that the vectors must have an equal number of components.

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} + \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} x_0 + y_0 \\ x_1 + y_1 \\ \vdots \\ x_{n-1} + y_{n-1} \end{bmatrix}$$

- Note that the vectors must be of equal dimension to perform addition with them
- Geometrically, you place vectors “head to tail”, then draw the **resultant** vector from the tail of the first to the head of the second vector.

Vector Addition: Example

- Vector addition is both associative and commutative

$$\begin{pmatrix} -1 \\ 2 \end{pmatrix} + \left(\begin{pmatrix} -3 \\ -2 \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} \right) = \begin{pmatrix} -3 \\ 2 \end{pmatrix}$$

$$\left(\begin{pmatrix} -1 \\ 2 \end{pmatrix} + \begin{pmatrix} -3 \\ -2 \end{pmatrix} \right) + \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} -3 \\ 2 \end{pmatrix}$$

Vector Subtraction

$$u - v$$

- As with vector addition, we subtract vectors by subtracting the components of the two vectors from each other, yielding a new vector of equal dimension

$$\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{bmatrix} - \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} u_0 - v_0 \\ u_1 - v_1 \\ \vdots \\ u_{n-1} - v_{n-1} \end{bmatrix}$$

- Subtracting v from u can also be defined as:

$$u - v = u + (-v)$$

- So in order to subtract, negate v, and then add to u.

Vector Subtraction: Geometric View

a-b denotes the vector starting at the head of b and ending at the head of a.

b-a would denote the same length vector but starting at the head of a and ending at the head of b

Vector Subtraction: Example

- Subtraction is **not** commutative.

$$\mathbf{u} = (1, -3)$$
$$\mathbf{w} = (-6, 6)$$
$$\mathbf{u} - \mathbf{w} = (7, -9)$$
$$\mathbf{w} - \mathbf{u} = (-7, 9)$$

Multiplication of a Vector by a Scalar: Algebraic View

- When multiplying a vector by a number, multiply all the components of the vector by that number. These factors **scale** the vector

$$a \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} au_0 \\ au_1 \\ \vdots \\ au_{n-1} \end{bmatrix}$$

Geometric Interpretation of Multiplication by Scalar

- The new vector will have the same direction if the scalar is positive, but the vector will have a different magnitude.
- When multiplying by a negative number the direction of the arrow will reverse, and scale in the opposite direction.

Multiplication by a Scalar

- Multiplication by a scalar is commutative.

$$4\begin{bmatrix}21\\8\end{bmatrix} = \begin{bmatrix}4\times 21\\4\times 8\end{bmatrix} = \begin{bmatrix}84\\32\end{bmatrix}$$

$$7\begin{bmatrix}3\\8\end{bmatrix} = \begin{bmatrix}7\times 3\\7\times 8\end{bmatrix} = \begin{bmatrix}21\\56\end{bmatrix}$$

Which vector is equal to:

- 2**a** = d
- (½)**a** = b
- **a**(½) = b
- -(½)**a** = c

Linear Combination

- A linear combination of u and v is a vector a**u**+b**v** where a and b are scalars.

- A unit basis vector is a vector with 0 in all entries, except for a 1 in one entry.
- All vectors can be represented as a linear combination of the unit basis vectors.

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\alpha \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \gamma \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

Linear Combination - Example

$$\alpha \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \gamma \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 5 \\ 7 \end{bmatrix}$$

$$\alpha =$$

$$\beta =$$

$$\gamma =$$

Linear Combination - Example

$$\alpha \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \gamma \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 5 \\ 7 \end{bmatrix}$$

$$\alpha = 8$$

$$\beta = 5$$

$$\gamma = 7$$

Alpha, beta, and gamma are formally called weights, as they “weigh” the value of each unit vector.

Superposition can be represented by a **linear combination** of states

check-in

Verify that the algebraic and geometric definitions of scalar multiplication and addition of vectors agree.

$$3\begin{pmatrix}1\\2\end{pmatrix}+\begin{pmatrix}4\\5\end{pmatrix}$$

So, first compute algebraically, then sketch the vectors and apply the geometric rules and check that in each case you get the same vector!

Vector-valued Functions

- What is a vector-valued function?
- The Dot Product

What is a vector-valued function?

- Definition: A vector-valued function is a function of one or more numbers or vectors to an output vector.
- We may think of a scalar as a one-dimensional vector, so the function can take in a scalar as well.

Example:

$$f(\alpha, \beta) = \begin{pmatrix} \alpha + \beta \\ \alpha - \beta \end{pmatrix} \quad \text{so that} \quad f(-2, 1) = \begin{pmatrix} -2 + 1 \\ -2 - 1 \end{pmatrix} = \begin{pmatrix} -1 \\ -3 \end{pmatrix}$$

Examples of Vector-Valued Functions

- Please evaluate the following:

$$f\left(\alpha, \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} x_0 + \alpha \\ x_1 + \alpha \\ x_2 + \alpha \end{pmatrix}$$

$$f\left(\beta, \alpha \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} \alpha x_0 + \beta \\ \alpha x_1 + \beta \\ \alpha x_2 + \beta \end{pmatrix}$$

$$f\left(-2, \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}\right) = \begin{pmatrix} 1 + (-2) \\ 2 + (-2) \\ 3 + (-2) \end{pmatrix}$$

$$= \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

Dot or Scalar Product

- The dot product is a vector-valued function that takes in two vectors and yields a scalar.
- For the dot product in real vector spaces, multiply the corresponding entries in each vector, and add the products. This yields a scalar.

- If the dot product of 2 non-trivial (non-zero) vectors is 0, then they are orthogonal (meet at a right angle).

$$x^T y = \text{dot}\ (x, y) = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}^T \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 & \dots & x_{n-1} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = x_0 y_0 + x_1 y_1 + \dots + x_{n-1} y_{n-1}$$

Dot Product - Algebraic View

To compute the dot product we follow the following protocol:

1. First transpose the first vector
2. Then multiply each component of the first vector with its respective component in the second vector
3. Then sum up all these component products yielding a **scalar** answer

$$v \cdot w$$

Two column vectors:

$$v = \begin{pmatrix} 3 \\ -2 \\ -6 \end{pmatrix}, w = \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix}$$

Dot Product Example I

$$\begin{pmatrix} 3 \\ -2 \\ -6 \end{pmatrix}^T \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix} = \begin{pmatrix} 3 & -2 & -6 \end{pmatrix} \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix}$$

$$\mapsto 3 \cdot 2 + (-2) \cdot (-3) + (-6) \cdot 5 = -18$$

Dot Product - Example II

$$\begin{pmatrix} 2 \\ 5 \\ -6 \\ 1 \end{pmatrix}^T \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \text{Undefined}$$

- The **dimensions** of the vectors need to be **identical** in order to perform the operation.
- Dot Products (and several other functions) return a scalar, not a vector. We consider the result, a scalar, as a 1x1 vector.

Dot Product - Example III

$$\begin{pmatrix} 2 \\ 5 \\ -6 \\ 1 \end{pmatrix}^T \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 2 \qquad \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}^T \begin{pmatrix} 2 \\ 5 \\ -6 \\ 1 \end{pmatrix} = 2$$

The dot product of vectors with real components is commutative because multiplication in R is commutative.

Inner Product of Complex Vectors

We can generalize the dot product to the **inner product** to now also encompass vectors with complex-valued components. When computing the inner product of vectors with complex components, **don't forget to conjugate!**

So therefore the inner product of two vectors where at least one of the components of either vector is not real, **is not commutative.**

A vector with a dagger in the superscript indicates we should take the conjugate transpose of that vector, i.e.,

$$\begin{pmatrix} 3+i \\ -2-i \\ -6i \end{pmatrix}^{\dagger} \begin{pmatrix} 2+5i \\ -3i \\ 5 \end{pmatrix} = \begin{pmatrix} 3-i & -2+i & +6i \end{pmatrix} \begin{pmatrix} 2+5i \\ -3i \\ 5 \end{pmatrix} = (11+13i) + (3+6i) + 30i = 14+49i$$

Slicing and Dicing - Dot Product

- When **partitioning** 2 vectors of equal sizes into congruent parts,their dot products can be added to obtain the original

$$\begin{pmatrix} 6 \\ 5 \\ 3 \end{pmatrix} \xrightarrow{T} \begin{pmatrix} 4 \\ 5 \\ 5 \end{pmatrix} \quad \begin{matrix} 24 + 25 = 49 \\ \\ = 15 \end{matrix}$$

$$15 + 49 = 64$$

Numpy for Dot Product

```
y = np.array([1,2,3])
x = np.array([2,3,4])
np.dot(y,x) = 20
```

Vector Norms

- Norms are used to determine the magnitude or length of the vector
- The most common norm is the L^2 norm, which is the Cartesian distance of the vector.

- L^2 (Also called the Euclidean Norm or length)
- L^1 (the Manhattan distance)

Vector Norms

The norm is a vector-valued function that gives us some sense of the magnitude of a vector. It allows us to compare two or more vectors.

Here is the generalized form of the norm. In practice, we focus on norms of p=1 and p=2

$$||x||_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$$

The L^2/ Euclidean Norm

Example 1:

- The L^2 Norm is the square root of the sum of the squares of the elements in a vector.

$$||x||_2 = \sqrt{x_0^2 + x_1^2 + \ldots + x_n^2} = \sqrt{\sum_{i=0}^{n} x_i^2}$$

The L^2 Norm of a vector is its length, a direct consequence of the Pythagorean theorem.

Observe that the square of the L^2 norm of a vector is the dot product with itself!

p=2

$$||x||_2 := \left(\sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}} = \sqrt{\sum_{i=1}^{n} x_i^2}$$

The L^2/ Euclidean Norm Examples

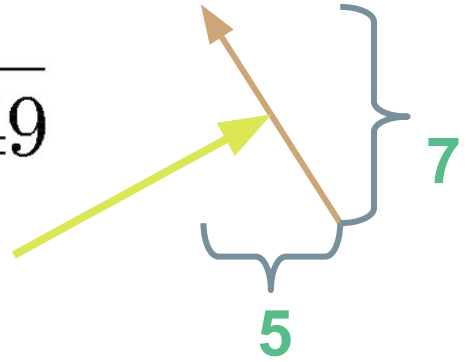
Example 2:

$$\begin{pmatrix}6\\7\\3\end{pmatrix} \rightarrow \sqrt{6^2+7^2+3^2} \rightarrow \sqrt{36+49+9} \rightarrow \sqrt{94} \rightarrow 9.695$$

Example 3:

$$\sqrt{5^2+7^2} = \sqrt{25+49}$$

$$= \sqrt{74} \approx 8.60$$



L^1 Norm

The L^1 norm (also known as the “Manhattan Distance” or “TaxiCab Norm” is calculated by adding together the absolute value of every component in the vector.

$$\left| \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \right|_1 = |u_1| + |u_2|$$

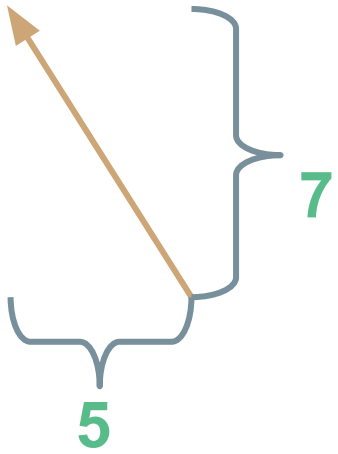
p=1

$$||x||_1 := \left(\sum_{i=1}^{n} |x_i|^1 \right)^{\frac{1}{1}} = \sum_{i=1}^{n} |x_i|$$

L^1 Norm - Example

Find the L^1 Norm of the following:

$$\begin{pmatrix} 6 \\ 7 \\ 3 \end{pmatrix} \rightarrow |6| + |7| + |3| = 16$$



$|5| + |7| = 12$

True or False? The L1 Norm will always be larger than or equal to the L2 Norm.

True: Because the L1 Norm is the equivalent to adding 2 sides of a triangle, and the L2 Norm is just the Hypotenuse (only 1 side).

In a triangle, any 2 sides must be $\geq$ than 1.

L^1 vs L^2 Norm - Graphically

Vectors where the L2 Norm is equal to 5

Vectors where the L1 Norm is equal to 5

L1 and L2 Regularization in Machine Learning

We can use the L1 and L2 norms to improve results in machine learning.

What is Regularization? It is a procedure that adds a penalty to the cost function of a model <u>to prevent overfitting</u>. For example, in linear regression, we may want to find a "best-fit" line (with slope β) that matches the data but isn't too "complex".

$$\min_{\beta} \|X\beta - Y\|_2^2 + \lambda\|\beta\|_p$$

best-fit objective

how much regularization

L^p regularization

when p=2 (the L^2 norm), this is called Ridge Regression. When p=1 (the L^1 norm), this is called Lasso Regression.

When to use L^1 and L^2 regularization

Adding L^1 and L^2 norms regularization encourages “simple” solutions.

- **L^2 norm (ridge regression):** penalizes β that are far from the origin in Euclidean space, leading to many small values. No single feature affects the output very much.
- **L^1 norm (lasso regression):** prefers β to be either large or exactly 0, leading to sparse solutions (many zeros). Only a few features have any impact.

Try it Yourself: Regularization & L2 Norm

- L^1 and L^2 regularization are both widely used in practice.
- When is it better to use one or the other in practice?

Try it Yourself: Regularization & L2 Norm

- L^1 and L^2 regularization are both widely used in practice.
- Compare two models A and B, with coefficients *a* and *b*. Both have the same performance.
  - Using **L^1** regularization, which model would be favored?
  - Using **L^2** regularization, which model would be favored?

$$a = \begin{pmatrix} 3 & 2 & 2 & 1 \end{pmatrix}$$

$$||a||_1 = |3| + |2| + |2| + |1| = 8$$

$$||a||_2 = \sqrt{3^2 + 2^2 + 2^2 + 1^2} = \sqrt{18}$$

$$b = \begin{pmatrix} 5 & 1 & 0 & 0 \end{pmatrix}$$

$$||b||_1 = |5| + |1| + |0| + |0| = 6$$

$$||b||_2 = \sqrt{5^2 + 1^2 + 0^2 + 0^2} = \sqrt{26}$$

Try it Yourself: Regularization & L2 Norm

- L^1 and L^2 regularization are both widely used in practice.
- Compare two models A and B, with coefficients *a* and *b*. Both have the same performance.
  - Using **L^1** regularization, which model would be favored? **a**
    - **L^1** favors sparser models (more 0 coefficients)
  - Using **L^2** regularization, which model would be favored? **b**
    - **L^2** allows for more smaller nonzero coefficients

$$a = \begin{pmatrix} 3 & 2 & 2 & 1 \end{pmatrix}$$

$$||a||_1 = |3| + |2| + |2| + |1| = 8$$

$$||a||_2 = \sqrt{3^2 + 2^2 + 2^2 + 1^2} = \sqrt{18}$$

$$b = \begin{pmatrix} 5 & 1 & 0 & 0 \end{pmatrix}$$

$$||b||_1 = |5| + |1| + |0| + |0| = 6$$

$$||b||_2 = \sqrt{5^2 + 1^2 + 0^2 + 0^2} = \sqrt{26}$$

L1 and L2 Regularization in Machine Learning

- We can use the L1 and L2 norms to improve our results in machine learning. For example, we can use the L2 norm in a procedure called Ridge Regularization.
- What is Regularization? It is a procedure that adds a penalty to the cost function of a model. The cost function is a function used to compute the error of the model and improve the model in an iterative manner. By adding an additional term to the cost function such as the square of the L2 norm, for example, we are penalizing the weights that are large. This can drive our model to better convergence and results.
- Similarly, we can add the square of the L1 norm and we call this Lasso Regularization.

Dot or Scalar Product

- For the dot product in real vector spaces, multiply the corresponding entries in each vector, and add the products. This yields a scalar.

The dot product is a **binary** operation. You can't dot three vectors - try it!

The dot product is an example of a vector function: it accepts as inputs two vectors and outputs a scalar (which may be thought of as a vector with one component).

$$x^T y = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}^T \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

$$= \begin{pmatrix} x_0 & x_1 & \dots & x_{n-1} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

$$= x_0 y_0 + x_1 y_1 + \dots + x_{n-1} y_{n-1}$$

Dot Product - Algebraic View

To compute the dot product we follow the following protocol:

1. First transpose the first vector
2. Then multiply each component of the first vector with its respective component in the second vector
3. Then sum up all these component products yielding a scalar answer

$$v \cdot w$$

Two column vectors:

$$v = \begin{pmatrix} 3 \\ -2 \\ -6 \end{pmatrix}, w = \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix}$$

Algebraic Aspects of the Dot Product

- The dot product of vectors with **real components** satisfies the algebraic properties at right, which we'll want to know to investigate the geometric interpretation of the dot product.

$$||u||^2 = u \cdot u$$

$$(au) \cdot v = a(u \cdot v) = u \cdot (av)$$

$$u \cdot v = v \cdot u$$

$$(u + v) \cdot w = u \cdot w + v \cdot w$$

Distributing the Dot Product over Addition

- We can distribute the dot product over addition.
- For example:

$$(u + v) \cdot w = u \cdot w + v \cdot w$$

$$u \cdot (v + w) = u \cdot v + u \cdot w$$

Scalar and Dot Product

If we wish to multiply scalar by the result of a dot product we can do so these ways:

$$a(u \cdot v) = (au) \cdot v = u \cdot (av)$$

$$a^2(u \cdot v) = (a^2 u) \cdot v = (au) \cdot (av) = u \cdot (a^2 v)$$

Dot Product Example I

$$\begin{pmatrix} 3 \\ -2 \\ -6 \end{pmatrix}^T \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix} = \begin{pmatrix} 3 & -2 & -6 \end{pmatrix} \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix}$$

$$\mapsto 3 \cdot 2 + (-2) \cdot (-3) + (-6) \cdot 5 = -18$$

The Dot Product of a Vector with Itself

To see why the dot product of a vector with itself is its L^2 norm squared, let

$$\mathbf{v} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix}$$

be any vector. Let's compute the dot product of **v** with itself...

$$\mathbf{v} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix}^T \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix} = \begin{pmatrix} v_0 & v_1 & \ldots & v_{n-1} \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix} = v_0^2 + v_1^2 + \ldots + v_{n-1}^2 = \left(\sqrt{v_0^2 + v_1^2 + \ldots + v_{n-1}^2}\right)^2$$

A Useful Identity

In particular, we'll want to know the identity below:

By definition, the square of the norm of the difference of v and u is:

$$||v - u||^2 = (v - u) \cdot (v - u)$$

And since the dot product distributes over sums (so, we think of v-u as v+(-u)):

$$= (v - u) \cdot v + (v - u) \cdot (-u)$$

The computation continues...

Applying the distributive property again, to each summand:

$$(v-u)\cdot v+(v-u)\cdot(-u)$$
$$=v\cdot v+(-u)\cdot v+v\cdot(-u)+(-u)\cdot(-u)$$

Remembering that the dot product of a vector with itself is its norm squared:

$$=||v||^2-u\cdot v-v\cdot u-u\cdot(-u)$$

... the finale!

An application of the commutativity of the dot product and the scalar multiplication property yields:

$$||v||^2 - u \cdot v - v \cdot u - u \cdot (-u)$$
$$= ||v||^2 - v \cdot u - v \cdot u - (u \cdot (-u))$$

Cleaning things up, we have a useful identity for what's to come:

$$||v - u||^2 = ||v||^2 - 2v \cdot u + ||u||^2$$

Arithmetic with Dot Products

All of those properties are overwhelming - let's compute something!

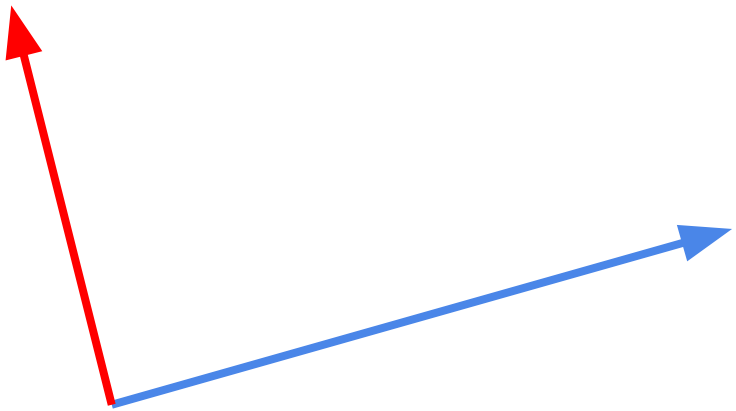
Let **u**=(4,6) and **v**=(1,2). First, let's compute $||\mathbf{u}-\mathbf{v}||^2$ with the observation that **u**-**v**=(3,4). We've done this! It's 5^2 = 25. Now, let's work at this to verify the previous identity with the properties we've just learned...

$$
\begin{aligned}
&||u-v||^2 = ((4,6)-(1,2))\cdot((4,6)-(1,2)) \\
&= ((4,6)-(1,2))\cdot(4,6) + ((4,6)-(1,2))\cdot \\
&(-(1,2)) \\
&= (4,6)\cdot(4,6) + (-(1,2))\cdot(4,6)) + (4,6)\cdot \\
&(-(1,2)) + (-(1,2))\cdot(-(1,2)) \\
&= ||(4,6)||^2 - (1,2)\cdot(4,6) - (4,6)\cdot(1,2) + \\
&||(1,2)||^2 \\
&= ||(4,6)||^2 - 2(4,6)\cdot(1,2) + ||(1,2)||^2 \\
&= ||v||^2 - 2v\cdot u + ||u||^2
\end{aligned}
$$

The Dot Product, Geometric View

Now that we're familiar with norms, we can speak of the geometric interpretation of the dot product.

- If the dot product of 2 nonzero vectors is 0, then they are perpendicular (form a right angle) - we'll see why soon!
- The dot product of a vector with itself is the square of its L^2 norm (the square of its length). Try this with (3,4)!



The Law of Cosines

The Law of Cosines will elucidate the geometric interpretation of the dot product.

The Law of Cosines is a generalization of The Pythagorean Theorem (PT), so we may think of the Pythagorean Theorem as a special case of the Law of Cosines where the angle is a right angle.

Analogously, we may think of what is being subtracted on the right as an error term.

$$||v-u||^2 = ||v||^2 - 2||v||||u||\cos\,(\theta) + ||u||^2$$

The Law of Cosines Continued...

The Law of Cosines tell us

$$||v-u||^2 = ||v||^2 - 2||v||||u||\cos(\theta) + ||u||^2$$

And we've derived the identity

$$||v-u||^2 = ||v||^2 - 2v \cdot u + ||u||^2$$

So, we may conclude

$$\boxed{v \cdot u = ||v||||u||\cos(\theta)}$$

Finding the Angle Between Two Vectors = Ex 2

Equipped with this machinery, let's compute the angle between two vectors!

$$v_1 = (1, 0, 0), v_2 = (0, 1, 1)$$

$$\cos\ (\theta) = \frac{(1,0,0)\cdot(0,1,1)}{||(1,0,0)||||(0,1,1)||} = 0$$

… and cosine is zero when theta is pi/2, i.e. 90 degrees. Was it obvious to you that these two vectors are orthogonal? It wasn't to me. Try to visualize them and verify for yourself that they are orthogonal geometrically.

The Dot Product and the Projection Operator

The dot product equals the product of the L2 norm of each vector and the cos of the angle between the two vectors. Observe that our definition of the dot product as

$$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}||||\mathbf{v}||\cos\ (\theta)$$

allows us to define the projection **proj$_\mathbf{v}$u** of a vector **u** onto a vector **v** via

$$\mathbf{proj}_{\mathbf{v}}(\mathbf{u}) = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{v}||||\mathbf{v}||}\mathbf{v}$$

$$\mathbf{proj}_{\mathbf{v}}(\mathbf{u}) = \frac{||\mathbf{u}||||\mathbf{v}||\cos\ (\theta)}{||\mathbf{v}||||\mathbf{v}||}\mathbf{v}$$

$$= \frac{||\mathbf{u}||\cos\ (\theta)}{||\mathbf{v}||}\mathbf{v}$$

$$= ||\mathbf{u}||\cos\ (\theta)\frac{\mathbf{v}}{||\mathbf{v}||}$$

So, the projection operator is defined correctly!

The Projection Operator

The dot product equals the product of the L2 norm of each vector and the cos of the angle between the two vectors. Observe that our definition of the dot product as

$$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}|| ||\mathbf{v}|| \cos(\theta)$$

allows us to define the projection **proj$_{\mathbf{v}}$u** of a vector **u** onto a vector **v** via

$$\mathbf{proj}_{\mathbf{v}}(\mathbf{u}) = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{v}||\mathbf{v}|}\mathbf{v}$$

Unraveling this a bit, $\frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{v}||}$ is a number, specifically the magnitude of the projection of the vector **u** onto the vector **v**.

$\frac{\mathbf{v}}{||\mathbf{v}||} = \frac{1}{||\mathbf{v}||}\mathbf{v}$ is the vector **v** after being scaled by its length, so it's exactly the unit vector in the direction of **v**.

So, realizing the expression at left as

$$\frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{v}||||\mathbf{v}||}\mathbf{v} = \left(\frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{v}||}\right)\left(\frac{1}{||\mathbf{v}||}\mathbf{v}\right)$$

... we see the projection is defined correctly!

Finding the Angle Between 2 Vectors

Use this formula to determine the cosine of the angle between two vectors which is the dot product of the two vectors divided by the product of the two L2 norms (length)

$$\cos\theta = \frac{F_1 \bullet F_2}{|F_1||F_2|}$$

$$\therefore \theta = \operatorname{acos}\left(\frac{(2)(5)+(4)(3)}{\sqrt{2^2+4^2}\sqrt{5^2+3^2}}\right)$$

$$\therefore \theta = \operatorname{acos}\left(\frac{22}{(4.47)(6)}\right) = 32.5^\circ$$

$$F_1 = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

$$F_2 = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

Projecting to Return Vector Components

The dot product of a vector with any of the standard basis vectors e_1=(1,0,...,0), e_2=(0,1,0,...,0), etc. returns the ith component of the vector. In other words, such a dot product is the same as taking the projection of the first vector onto that basis vector.

This is referred to as an orthogonal projection (we'll learn what orthogonal means soon).

For example, let's project the vector (1,2,3,4) onto the 3rd basis vector e_3=(0,0,1,0)...

$$\begin{pmatrix}1\\2\\3\\4\end{pmatrix}^T \begin{pmatrix}0\\0\\1\\0\end{pmatrix} = \begin{pmatrix}1 & 2 & 3 & 4\end{pmatrix} \cdot \begin{pmatrix}0\\0\\1\\0\end{pmatrix} = 1\cdot 0 + 2\cdot 0 + 3\cdot 1 + 4\cdot 0 = 3$$

Projecting using the all-1's vector

For vectors with nonnegative entries, taking the dot product with the all-one vector is literally computing the Manhattan distance.

Why do the entries have to be nonnegative for this to be true?

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}^T \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$= 1 \cdot 1 + 2 \cdot 1 + 3 \cdot 1 + 4 \cdot 1 = 1 + 2 + 3 + 4$$

Summary of Vector Operations

$$\begin{bmatrix} u_0 \\ u_1 \end{bmatrix} + \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = \begin{bmatrix} u_0 + v_0 \\ u_1 + v_1 \end{bmatrix}$$

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} - \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} x_0 - y_0 \\ x_1 - y_1 \\ \vdots \\ x_{n-1} - y_{n-1} \end{bmatrix}$$

$$\left|\left|\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}\right|\right|_1 = |u_1| + |u_2|$$

$$||x||_2 = \sqrt{x_0^2 + x_1^2 + \ldots + x_n^2} = \sqrt{\sum_{i=0}^{n} x_i^2}$$

$$a \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} au_0 \\ au_1 \\ \vdots \\ au_{n-1} \end{bmatrix}$$

$$u \cdot v = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}^T \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} = \sum_{i=1}^{n} u_i v_i = u_1 v_1 + \cdots + u_n v_n$$

Try it Yourself: Compare Word Similarities

- In the same way words that compose a document can be one-hot encoded into vectors, letters that compose words can one-hot encoded as well.

- This allows us to graph the vectors created in this manner and find the distance between them. The closer they are, the more similar they are!

- Try it yourself

Check-in

Verify algebraically that the only scenario where the L^1 and L^2 norms of a vector agree is when at most one of its components is non-zero.

Interpret this geometrically! Why should this be true?

Verify that the vectors (1,0) and (0,1) are perpendicular using the definition of cosine given in the previous slides.

Compute the projection of the vector (3,4) onto the vector (7,8) using the previously provided machinery.

The Matrix

What is a Matrix?

In mathematics, a matrix is a rectangular array of numbers, symbols, or expressions, arranged in rows and columns.

- We refer to the size of the matrix by the number of rows followed by the number of columns.
  We refer to a matrix with 4 rows and 5 columns as a 4 by 5 matrix.
- We refer to elements in the matrix by their coordinates i (for rows) and j (for columns) such that in the matrix **A**, element $A_{3,2}$ is in the 3rd row, and second column.
- A matrix is often denoted as an uppercase bold letter, such as **A**.
- We specify an entry in **A** with $\mathbf{A}_{m,n}$ and a row of **A** with $\mathbf{A}_{m,:}$ and a column of **A** with $\mathbf{A}_{:,n}$

Numpy for Matrices

```
In [46]: Y = np.arange(12).reshape(3,4)

In [47]: Y

Out[47]: array([[ 0, 1, 2, 3], [ 4, 5, 6, 7], [ 8, 9, 10, 11]])

In [48]: Y.shape

Out[48]: (3, 4)

In [49]: Y.shape[0]

Out[49]: 3
```

Basic Operations with Matrices

- Matrix Addition
- Matrix Subtraction
- Direct Sum of Matrices
- Transposing a Matrix
- Multiplying a Matrix by a Matrix

Matrix Addition

- When adding a matrix to another matrix, both must be the same size.
- To add the matrix, we add the entries of the matrix.
- Matrix addition is commutative:

Matrix Addition Examples

- Calculate the following

$$\begin{bmatrix} 1 & 3 \\ 1 & 0 \\ 1 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 7 & 5 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 \\ 1+7 & 0+5 \\ 1+2 & 2+1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 8 & 5 \\ 3 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 6 & -7 \\ 4 & 7 & -5 \end{bmatrix} + \begin{bmatrix} 3 & 6 \\ -7 & 4 \\ 9 & -5 \end{bmatrix} = \text{Undefined}$$

Scalar Multiplication of Matrices

We may also multiply matrices by scalars, as demonstrated here:

$$3\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 3\cdot 1 & 3\cdot 2 \\ 3\cdot 3 & 3\cdot 4 \end{pmatrix} = \begin{pmatrix} 3 & 6 \\ 9 & 12 \end{pmatrix}$$

Matrix Subtraction

- When subtracting a matrix from another matrix, both must be the same size.
- To subtract the matrix, we subtract the corresponding elements of the matrices:

$$\begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} - \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} = \begin{bmatrix} u_{11} - v_{11} & u_{12} - v_{12} \\ u_{21} - v_{21} & u_{22} - v_{22} \end{bmatrix}$$

Matrix Subtraction Examples

Please compute the following:

$$\begin{bmatrix} 5 & 1 \\ 3 & -3 \end{bmatrix} - \begin{bmatrix} 3 & 4 \\ 5 & 5 \end{bmatrix} = \begin{bmatrix} 5-3 & 1-4 \\ 3-5 & -3-5 \end{bmatrix} = \begin{bmatrix} 2 & -3 \\ -2 & -8 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 1 \\ 3 & -3 \\ 5 & 5 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 5-0 & 1-0 \\ 3-0 & -3-0 \\ 5=0 & 5-0 \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 3 & -3 \\ 5 & 5 \end{bmatrix}$$

This matrix is called the zero matrix as it contains only zeros.

Direct Sum of a Matrix

- The direct sum of matrices is a combination of 2 matrices (of any size) into one larger combined matrix.
- A direct sum is denoted as $\oplus$
- If A and B are operators on vector spaces V and W, respectively, then $A \oplus B$ is an operator on the direct sum of the vector spaces $V \oplus W$.

Direct Sum of Matrices - Example

Not so bad, right?

$$\begin{bmatrix} 1 & 3 & 2 \\ 2 & 3 & 1 \end{bmatrix} \oplus \begin{bmatrix} 1 & 6 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 2 & 0 & 0 \\ 2 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Transposing a Matrix

- To transpose a matrix, exchange the rows and columns, i.e.

  for all entries of A

  $a^T_{ij} := a_{ji}$

Numpy for Transpose

```
a = np.array([

  [1, 2],

  [3, 4]])
a.T
[[1, 3],
[2, 4]]
```

$$\mathrm{M} = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}$$

$$\mathrm{T} = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}$$

Symmetric Matrices

- A symmetric matrix is a square matrix that is equal to its transpose. Formally, matrix A is symmetric if $A = A^T$
- Only square matrices can be symmetric.
- The entries of a symmetric matrix are symmetric with respect to the main diagonal. So if the entries are written as **A** = (a_{ij}), then $a_{ij} = a_{ji}$, for all indices i and j.
- The following 3 × 3 matrix is symmetric:

$$\begin{pmatrix} 2 & 1 & 9 \\ 1 & -7 & 11 \\ 9 & 11 & -5 \end{pmatrix}$$

Matrix Multiplication

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 \cdot 5 + 2 \cdot 7 & 1 \cdot 6 + 2 \cdot 8 \\ 3 \cdot 5 + 4 \cdot 7 & 3 \cdot 6 + 4 \cdot 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

- Matrix multiplication is in essence just several dot products.
- The solution matrix has the number of rows in the first matrix and number of columns from the second matrix, that is m of the first matrix and n of the second matrix.
- Calculate the dot product of the top row (of the first matrix) and the left column (of the second matrix), and that goes in the top left, etc...

Recall….Dot Product Example I

$$\begin{pmatrix} 3 \\ -2 \\ -6 \end{pmatrix}^T \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix} = \begin{pmatrix} 3 & -2 & -6 \end{pmatrix} \begin{pmatrix} 2 \\ -3 \\ 5 \end{pmatrix}$$

$$\mapsto 3 \cdot 2 + (-2) \cdot (-3) + (-6) \cdot 5 = -18$$

Matrix Multiplication- Procedure

- Here is another view of matrix multiplication to make it clear that dot product each row of the first matrix with each column of the second matrix
- **This yields a third matrix with the same number of rows as the first matrix and the same number of columns as the second matrix.**

Matrix Multiplication as Composition of Functions

The definition of matrix multiplication might seem a bit weird, but it turns out to be exactly the definition so that we have the following result:

Proposition. For any matrices **A** and **B**, the matrix of the composition of the linear transformations, **A**○**B**, is **AB**.

So, if you compose the two functions A and B and determine the matrix of the resulting function, then you get exactly the matrix from the definition of matrix multiplication!

Matrix Multiplication as Composition of Functions

Consider the two matrices:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Compute the product:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Matrix Multiplication as Composition of Functions

You should get the product

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Now, apply that product to the vector (0,1). What do you get?

Matrix Multiplication as Composition of Functions

See if you get:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

Matrix Multiplication as Composition of Functions

Separately, apply the two matrices to the vector (0,1) one by one. First, matrix multiplication is associative, so

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \left(\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right)$$

This yields

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \left(\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

Matrix Multiplication as Composition of Functions

So, iteratively applying the matrices, i.e. composing the functions they represent yields the same result as multiplying the two matrices first and then applying the resulting matrix!

In other words, matrix multiplication is literally composition of functions!

Matrix as a Transformation

We can think of a matrix as a transformation of space. The matrix tells us how to stretch, squeeze or rotate a vector space. To see how the matrix transforms a space, we can apply it to the vector space's basis vectors.

Note that matrices in the same dimensionality of the vector space CANNOT translate a vector. This is a non-linear operation.

One handy rule: any matrix maps the zero vector back to the zero vector. This is not the case in translation.

Test for Comprehension

Let $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Compute the transpose of A.

Then, compute each of $A^T A$ and AA^T. (Isn't that nice?)

Let **v**=(1,0). Compute A**v**, yielding a vector. Then, apply A^T to the resulting vector. So, you've computed $(A^T(Av))$. Now, using your previous computation of $A^T A$, compute $(A^T A)v$.

Observe that they are the same! This verifies that the multiplication of matrices correspond to composition of functions.

Multilayer Perceptron Calculations

Linear Transformation

- What is the definition of linear?
- What is a linear transformation?
- Why do we care about linear transformations?

What is the Definition of Linear?

Let **u**, **v** be vector in R^n
Let α, β be scalars in R

$$f(\mathbf{u}+\mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$$
$$f(\alpha\mathbf{u}) = \alpha f(\mathbf{u})$$
$$\Rightarrow f(\alpha\mathbf{u}+\beta\mathbf{v}) = \alpha f(\mathbf{u}) + \beta f(\mathbf{v})$$

What is the Definition of a Linear Transformation?

A vector function $L : R^n \rightarrow R^m$ is said to be a linear transformation:
if for all $\mathbf{u}, \mathbf{v} \in R^n$ and $\alpha \in R$:

- Transforming the sum of two vectors is the same as summing the two transformed vectors:

$$L(\mathbf{u}+\mathbf{v}) = L(\mathbf{u}) + L(\mathbf{v})$$

- Transforming a scaled vector is the same as scaling the transformed vector:

$$L(\alpha\mathbf{u}) = \alpha L(\mathbf{u})$$

Linear Transformation - Examples

Is $f\left(\begin{pmatrix} u_0 \\ u_1 \end{pmatrix}\right) = \begin{pmatrix} u_0^2 \\ u_1 \end{pmatrix}$ a linear transformation?

No, because in this case, applying the transformation and then scaling is NOT equal to scaling and then applying the transformation. An explicit example is given at right. It fails the test of a linear transformation.

$$f(1,0) + f(-1,0) = (1^2, 0) + ((-1)^2, 0)$$
$$= (1,0) + (1,0)$$
$$= (2,0) \neq (0,0) = f((0,0))$$
$$= f((1,0) + (-1,0)).$$

Is this transformation linear?

$$Y(x) = mx + b$$

where b $\neq$ 0

Is this transformation linear?

$$Y(x) = mx + b$$

Answer: No, if b is not 0

Why Not?

$$Y(x_1 + x_2) = m(x_1 + x_2) + b$$
$$= mx_1 + mx_2 + b \neq mx_1 + b + mx_2 + b$$

Linear Transformations and Operators

A **linear transformation** maps objects from one vector space to another vector space.

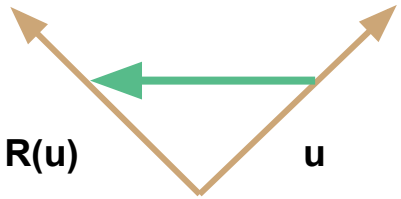
$$L : V \to W$$

A **linear operator** is a linear transformation that maps from a vector space to the same vector space.

$$L : V \to V$$

Linear Transformations

- A linear transformation is a vector function that only changes the direction/magnitude of a vector - no translations!
- Example: R(**u**) rotates **u** by 90°



Linear Independence
Span

Linearly Independent Vectors

Def 1: Two vectors are linearly independent iff they are not collinear.

Def 2: Two or more vectors are linearly independent iff you cannot express any of the vectors as a linear combination of the other vectors.

Def 3: two or more vectors are linearly independent iff none of the vectors are in the span of a subset of the vectors.

Def 4: $v_1 \ldots\ldots v_n$ are linearly independent iff the only way to express $a_1v_1+\ldots+a_nv_n = 0$ is with $a_1 \ldots\ldots a_n = 0$ (i.e. the only way to make a linear combination of the vectors that results in zero is to set all the scalar coefficients equal to zero.)

Linearly Dependent Vectors

We say that two vectors are linearly dependent if either is a scalar multiple of the other, i.e. u and v are linearly dependent iff au=v for some a in the field.

Geometrically, this is to say that two vectors are linearly dependent iff they are collinear.

We extend the definition of dependence to a set of vectors $v_1,\ldots,v_n$ by declaring them to linearly dependent iff any of them can be expressed as a linear combination of the others.

It turns out that the above definition is equivalent to the definition that $v_1,\ldots,v_n$ are linearly dependent iff there exist scalars $a_1,\ldots,a_n$ (not all zero) such that $a_1v_1+\ldots+a_nv_n=0$.

Span

The span of a single vector is the set of vectors that are collinear with it, i.e. the span of a single vector is the set of all scalar multiples of that vector.

The span of a vector is a subspace.

The span of a set of vectors is the sum of their spans and so is a subspace. So, if we have two vectors **u** and **v** of vector space V, the span of **u** and **v** is the set of linear combinations of **u** and **v**.

Span - example

What is the span of the vector (1,0) in R^2?

By definition, the span of (1,0) is the subspace of all multiples of (1,0) - i.e. the x axis!

What is the span of (0,1)?

What is the span of their sum?

Is it a direct sum?

Basis Vectors

For any finite dimensional vector space, V, there is a set of basis vectors:

a set of linearly independent vectors spanning V.

In fact, in most cases, there are infinite number of bases for any finite dim V.

Remarkably, any basis for a finite_dimensional vector space has the same size.

The adventurous audience member is welcome to try and construct a basis for R^2 of more or less than 2 vectors.

Basis Vectors

$$\alpha \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \gamma \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 5 \\ 7 \end{bmatrix}$$

$$\alpha = 8$$

$$\beta = 5$$

$$\gamma = 7$$

Alpha, beta, and gamma are formally called weights, as they “weigh” the value of each unit vector.

Lattices for cryptography

2D lattice

A lattice is a repeating pattern of points in R^n

A lattice is described by a set of basis vectors (known together as a basis), i.e. every vector in the lattice can be written as a linear combination of the basis vectors

$\mathbf{V} = x_1\, \mathbf{b_1} + x_2\, \mathbf{b_2} + \ldots + x_n\, \mathbf{b_n}$

(x_i are *integers*, $\mathbf{b_i}$ are basis vectors)

The same lattice has *infinitely many* different bases

Lattices for cryptography

Some bases contain vectors that are short and orthogonal (green arrows)

Others contain vectors that are long and non-orthogonal (red arrows)

Finding short vectors given a basis of long vectors is a difficult one-way problem

Lattices for cryptography

Public key - bad (long) basis. *ANYONE* can encrypt using bad basis

Private key - good (short) basis. *Only the person who knows this* can decrypt

If you can find short vectors given a bad basis, you can read private communications and **break the cryptosystem!**

Good basis to bad basis - EASY

Bad basis to good basis - HARD (even for quantum)

Orthogonal and orthonormal

A set of vectors is pairwise **orthogonal** if they are perpendicular, which implies their dot product is zero, i.e. $\mathbf{v}^T\mathbf{w}=0$ for all $\mathbf{v}$ and $\mathbf{w}$ in the set.

A set of vectors is **orthonormal** iff they are pairwise orthogonal and each have norm 1 (or unit norm).

e.g. the standard basis vectors in R^3: (1,0,0), (0,1,0), (0,0,1).

Orthogonal Matrix

An orthogonal matrix is a matrix where the column vectors of the matrix are pairwise orthogonal.

If we specify further that the columns of an m x n matrix A are orthonormal,

$$A^T A = I_n \qquad AA^T = I_m$$

Notice the resulting identity matrices are of different dimensions!

Dimension (often abbreviated “dim”)

The dimension is the number of basis vectors for a vector space. Remarkably, this definition of dimension is well-defined, since any basis for a vector space has the same number of vectors.

For example, R^2 has dimension two, since it has two basis vectors.

Non-square matrices and Vector Space Dimensions

A nonsquare matrix describes a linear transformation between vector spaces of different dimensions. In particular, if T is an m by n matrix, T maps from an n-dimensional space to an m-dimensional space.

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

The transformation above maps R^3 to R^2. Can you describe the map geometrically?

Check-in

Verify that the vectors (1,0) and (1,1) span R^2.

Equipped with this knowledge, express the vector (1,2) as a linear combination of (1,0) and (1,1).

Verify that the vectors (1,0) and (1,1) are not orthonormal. In particular, which of them are of unit length? Are they orthogonal?

Operations with Vectors and Matrices

- The Identity Matrix
- The Determinant

Diagonal Matrix

- A Diagonal Matrix is a special case of a Symmetric Matrix in which everything but the diagonal is zero.

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & -7 & 0 \\ 0 & 0 & -5 \end{pmatrix}$$

The Identity Matrix

- The identity matrix is a square matrix that contain 1's in the diagonal and zeros elsewhere.
- The identity matrix acts as the number 1 for matrices.
- See examples below:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Determinants

- **The determinant is a computable scalar quantity associated with every square matrix.**
- Determinants can only be computed on a square matrix.
- The determinant is a vector function which encodes algebraic and geometric information.

The Determinant as an Algebraic Phenomenon

The below are equivalent for any matrix A:

- det(A)=0
- the columns of A are linearly dependent
- the rows of A are linearly dependent
- A is not invertible
- The rank of A is not full
- The null space of A is nontrivial

The determinant is an **alternating** function in the sense that exchanging any two row or column vectors negates it.

The Determinant as a Geometric Phenomenon

Geometrically, we can think of the det of a linear transformation (equivalently, a matrix constructed via the standard basis vectors) as being the directed volume of the unit parallelepiped after the linear transformation.

Hence, for example, the linear transformation $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$ scales each of the standard basis vectors in the unit parallelepiped (unit cube in this case) by 2, so ultimately scales the entire volume by 2^3 (not just 2!), which is its determinant.

The sign of the determinant is positive iff the linear transformation preserves orientation (so, negative iff orientation-reversing).

The Parallelepiped

The parallelepiped is a higher-dimensional generalization of the two-dimensional parallelogram we know from Euclidean geometry.

Computing the Determinant for a 2x2 Matrix

- Compute the determinant of a 2x2 matrix given this formula

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\det(A) = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

$$\det(A) = ad - bc$$

- Example:

$$\mathbf{A} = \begin{bmatrix} 4 & 3 \\ 5 & 2 \end{bmatrix}$$

$$\det(A) = (2 \times 4) - (5 \times 3)$$

Computing the Determinant for a 3x3 Matrix

- $|A| = a(ei - fh) - b(di - fg) + c(dh - eg)$
- To work out the determinant of a 3×3 matrix:
  - Multiply a by the determinant of the 2×2 matrix that is not in a's row or column (these are called the cofactors)
  - Likewise for b, and for *c*
  - Sum them up, but remember the minus in front of the b
- Overall:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$|A| = a \cdot \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \cdot \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \cdot \begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

Example: Computing the Determinant for a 3x3 Matrix

- Assume we have the following matrix **C**:

$$\begin{pmatrix} 6 & 1 & 1 \\ 4 & -2 & 5 \\ 2 & 8 & 7 \end{pmatrix}$$

$$|C| = 6 \cdot (-2 \cdot 7 - 5 \cdot 8) - 1 \cdot (4 \cdot 7 - 5 \cdot 2) + 1 \cdot (4 \cdot 8 - (-2 \cdot 2)) = 6 \cdot (-54) - 1 \cdot (18) + 1 \cdot (36) = -306$$

Rank of a Matrix

The rank of a matrix is the maximum number of linearly independent column vectors in that matrix.

i.e. the rank of a matrix A where $A = (v_1 | \ldots | v_n)$ where v_i are column vectors, is the maximum number of linearly independent column vectors.

$$\begin{bmatrix} 1 & 3 & 2 & 0 & 0 \\ 2 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Example 1: the rank of this matrix is 4 since column vector 2 can be expressed as a linear combination of columns 1 and 3. The remaining column vectors are linearly independent

Calculating the Det for higher Dim Matrices

There's a handy technique called the cofactor expansion (or Laplace expansion) for calculating the determinant of higher dimensional matrices that ultimately boils down to computing determinants of dimension 2. Let's do this one:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Laplace Expansion

Choose any row or column along which you'd like to work. I'll choose the first row.

Then, use the components of that row or column as coefficients for the cofactor matrices (of one less dimension), each of which is multiplied by -1 to the power given by the sum of the indices of that component. Keep doing this until you've exhausted the row...

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} = (-1)^{1+1} \cdot 1 \cdot \begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix} + (-1)^{1+2} \cdot 2 \cdot \begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix} + (-1)^{1+3} \cdot 3 \cdot \begin{vmatrix} 4 & 5 \\ 7 & 8 \end{vmatrix}$$

Laplace Expansion continued...

... and we know how to compute this - finish it!

$$= \begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix} - 2 \cdot \begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix} + 3 \cdot \begin{vmatrix} 4 & 5 \\ 7 & 8 \end{vmatrix}$$

How to Build a Vector Space

Cartesian Product of Sets

The Cartesian product of sets is the set of ordered pairs of elements of S and T.

$$S \times T = \{(s, t) : s \in S \text{ and } t \in T\}$$

Cartesian Product of Vector Spaces

The Cartesian product of vector spaces and the direct sum of vector spaces are the same for a finite number of vector spaces.

$$S \times T = \{(s, t) : s \in S \text{ and } t \in T\}$$

We equip the Cartesian product of vector spaces S and T with the usual scalar multiplication and addition structure, i.e.

$$a \cdot (s, t) = (a \cdot s, a \cdot t) \; (s_1, t_1) + (s_2, t_2) = (s_1 + s_2, t_1 + t_2)$$

Cartesian Product of Vector Spaces - Examples

$$\mathbb{R}^2 := \mathbb{R} \times \mathbb{R} = \{(a, b) : a \in \mathbb{R}, b \in \mathbb{R}\}$$

How to Build a Vector Space

Vector Space

Algebra

Field

Ring

Module

Abelian Group

Non-Abelian Group

Set

Group Theory and Symmetry

How to Build a Vector Space

Vector Space: an abelian group equipped with a field action

Field: a commutative ring with a multiplicative inverse for each non-zero element of the ring

An Algebra: a ring, A, equipped w/ a commutative unital ring R and a ring homomorphism that maps from R to Z(A) (the center of A) and maps 1_R to 1_A.

Ring: an abelian group with another operation (typically multiplication) that is assoc, has identity and is distributive

Commutative Ring: an abelian group with another operation (typically multiplication) that is assoc, has identity and is distributive and commutes

Abelian Group: a set that meets all conditions of a group + commutativity of the operation

Module: an abelian group equipped with a ring action

Group: a set with an operation that meets 4 conditions (closure, assoc., identity, inverse) - the operation is typically addition

Set - collection of unique objects (order does not matter)

Groups

A **group** is a set G with a binary operation

(meaning that it is a function that operates on two elements of G at a time) that is

- closure $G \times G \rightarrow G$
- Associative a*(b*c) = (a*b)*c
- An identity element e in G such that a*e=e*a=a for all a in G
- Each element of G has an inverse

An **abelian group** is a group whose operation is commutative. (i.e. a+b = b+a)

Typically, the operation of an abelian group is denoted by +.

Niels Henrik Abel

Some examples of groups

The quintessential abelian group is the set of integers $\mathbb{Z}$ equipped with addition. With the same logic, we see that Q, R, and C are abelian groups (with addition).

- Addition of two integers yields an integer (closure),
- addition is associative,
- 0 serves as the identity under this operation,
- and each element of $\mathbb{Z}$ has its negative as an inverse under the operation.

Why is $\mathbb{Z}$ not a group when equipped instead with multiplication? Which element is the problem? Why is the set of natural numbers (including zero or not) not a group under addition?

Rings

A **ring** is an abelian group with +, and an additional binary operation * that is:

- associative,
- has an identity element
- is distributive (i.e. a*(b+c) = a*b + b*c) and (b+c)*a = b*a+ c*a)

A ring is called commutative if * is commutative too.

The quintessential commutative ring is the integers $\mathbb{Z}$, equipped with usual addition and multiplication. Not all rings are commutative, as will be demonstrated in the following slide!

Fields

We use F to denote a field. What is a Field?

A **field** is a commutative ring where each non-zero element has a multiplicative inverse.

So for each non-zero element of a field, you can find another element of the field such that their product is equal to 1.

Both the set C and the set R are fields. Each fulfill this condition of having a multiplicative inverse for each non-zero element.

For example in C, 3+2i has a multiplicative inverse of (3-2i)/13. In R, for example, the m. inverse of 5 is ⅕).

However, Z, the set of all integers, is **not** a field. For example, 2 does not have a multiplicative inverse in Z, since there does not exist an integer whose product with 2 is the multiplicative identity, 1. Naturally, ½ is the multiplicative inverse of 2, but ½ is not an integer, it is a rational number.

Field: A Set F, with + and x defined such that:

$1. \forall a, b \in F, a + b \in F$

$2. \forall a, b, c \in F, (a + b) + c = a + (b + c)$

$3. \exists 0 \in F \forall 0 + a = a + 0 = a$

$4. \forall a \in F, \exists (-a) \in F a + -a = -a + a = 0$

$5. \forall a, b \in F, a + b = b + a$

$6. \forall a, b \in F, a \times b \in F$

$7. \forall a, b, c \in F, (a \times b) \times c = a \times (b \times c)$

$8. \exists 1 \in F, 1 \times a = a \times 1 = a$

$9. \forall a, b, c, \in F, a \times (b + c) = a \times b + a \times c \& \forall a, b, c \in F, (a + b) \times c = a \times c + b \times c$

$10. \forall a, b \in F, a \times b = b \times a$

$11. \forall a \neq 0 \exists a^{-1} \in F \mathrm{a}^{-1} \times a = a \times a^{-1} = 1$

Which of these are Fields?

$\mathbb{Z}$ denotes the set of integers, all whole numbers, like ..., -2,-1,0,1,2, etc.

$\mathbb{Q}$ denotes the set of *ratio*nal numbers, i.e. all *ratios* of integers, e.g. ½, ¾, -⅞, etc.

$\mathbb{R}$ denotes the set of all real numbers, i.e. π , e , $\sqrt{2}$,$6.4\bar{4}$, 3.0, etc.

$\mathbb{C}$ denotes the set of complex numbers, i.e. i, 1+i, 3-7i, etc.

The fields R and C

R Real number field

C Complex field

F can refer to either R or C fields

A vector composed of real elements of R^2 space looks the same as a vector in C^1 space described by a complex number with the same coefficients. In fact, more generally C^N can be thought of as R^{2N}, topologically.

Galois Field (Finite Fields)

It might seem that all fields have to be infinite given the previous examples, but this is far from the case - in fact, there is a field with only 2 elements!

The Galois field GF(2) with two elements, typically denoted 0, and 1 has addition and multiplication defined via the tables

The addition table reflects the truth table of the logical operator XOR, and the multiplication table reflects AND.

We can make finite fields with number of elements equal to any power of a prime number.

So, in particular, there is a finite field with $2^2=4$ elements, but not a finite field with 2*3=6 elements.

To construct a finite field with a prime number p of elements, simply take the elements to be 0,1,2,...,p and add and multiply them mod p.

Constructing finite fields with a specified number of elements is a bit outside the scope of this course, but we know they exist!

Vector Spaces

A vector space is defined in terms of the field it is over. We say, for example, let **V** be a vector space over **R**. This tells us that the scalars used to scale vectors in this space are elements of R

- Example 1: $V=R^3$ over F=R.
- Example 2: $V= R^2$ over F=GF(2) (the Galois Field consists of {0,1} with binary (modulo) addition and multiplication.

- Example 3: $V=GF(3)^3$ over GF(3)

More formally, a vector space is an abelian group equipped with a field action. For example, we may take the abelian group to be C with usual vector addition and the field action to be usual scalar multiplication by R.

A Vector Space

A vector space is a nonempty set **V** of objects, called vectors, on which are defined two operations — addition, and multiplication by scalars from the field — and is subject to the ten axioms below. The axioms must hold for all u, v and w in V and for all scalars c and d.

1. $\mathbf{u} + \mathbf{v}$ is in V.
2. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$.
3. $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$
4. There is a vector (called the zero vector) 0 in V such that $\mathbf{u} + 0 = \mathbf{u}$.
5. For each **u** in V, there is vector $-\mathbf{u}$ in V satisfying $\mathbf{u} + -\mathbf{u} = 0$.
6. $c\mathbf{u}$ is in V.
7. $c(\mathbf{u} + \mathbf{v}) = c\mathbf{u} + c\mathbf{v}$.
8. $(c + d)\mathbf{u} = c\mathbf{u} + d\mathbf{u}$.
9. $(cd)\mathbf{u} = c(d\mathbf{u})$
10. $1\mathbf{u} = \mathbf{u}$

Subspaces

A subspace of a vector space **V** is a subset H of V that has three properties:

a. The zero vector of **V** is in H.

b. For each **u** and **v** in H, **u** + **v** is in H. (In this case we say H is closed under vector addition.)

c. For each **u** in H and each scalar c, c**u** is in H. (In this case we say H is closed under scalar multiplication.)

Subspaces are vector spaces.

Example of a Vector Space and a Subspace

- In the vector space $\mathbf{R^2}$ (the x-y plane)
  - {(a,0)} where a is any scalar in R is a subspace of R^2
  - So, some elements of the proposed subspace include (2,0), (-7,0), etc.
  - Why?
    - The subspace contains (0,0).
    - Every (v,0)+(u,0)=(v+u,0) is in (a,0).
    - For every scalar c in R,
      - c(a,0)=(ca,0) is in (a,0).
  - Also, {(0,a)} is a vector space in R^2 for the reasons above.

The Intersection of Subspaces

- Is the intersection of two subspaces a vector space?
  - Yes iff the intersection is non-empty. Why?
  - Since each subspace contains zero, their intersection does too.
  - Additive and scalar multiplicative closure applies to both subspaces, thus to their intersection.

The union of subspaces is not usually a subspace. Consider the subspaces defined by the axes in R^2. Their union is the cross, which surely contains zero, but does not have additive closure. Namely, the unit vectors (1,0) and (0,1) sum to (1,1), which *does not* belong to either axis.

Adding Subspaces

- We can add subspaces!
- Let U and V be subspaces of some vector space V.
- Define the sum of U and V to be the set {u+v : u is in U and v is in V}.
- That is, the sum of the subspaces is all of the sums of vectors in U and vectors in V.
- For example, the sum of the axes in the plane is all of the plane. This is precisely why their union is not a subspace! The sum includes many vectors not already in the summands. In fact, this is a very special sum...

Direct Sum of Subspaces

$$\begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix} \oplus \begin{bmatrix} a \\ b \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ c \\ d \end{bmatrix} = \begin{bmatrix} a \\ b+c \\ d \end{bmatrix}$$

- A direct sum is the sum of subspaces which only have zero vector in common (in other words they have trivial overlap, i.e. only intersect at the origin).
- A direct sum of subspaces does not necessarily give the whole vector space.
- Example 2 is **not** a direct sum despite yielding R^3: the subspaces overlap nontrivially.

Direct Sum of Subspaces

$$\overset{U}{\begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}} \oplus \overset{V}{\begin{bmatrix} a \\ b \\ 0 \end{bmatrix}} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$\overset{X}{\begin{bmatrix} a \\ b \\ 0 \end{bmatrix}} + \overset{Y}{\begin{bmatrix} 0 \\ c \\ d \end{bmatrix}} = \begin{bmatrix} a \\ b + c \\ d \end{bmatrix}$$

- Any element of R^3 can be written as either an element of U + V or X + Y.
- However, any element can be uniquely represented in U + V.
- In X + Y, there are multiple ways to express elements in R^3

Direct Sum of Subspaces - Example

For example, we can build R^3 inductively as follows:

First, take the direct sum of the subspaces spanned by the x and y axes, respectively, i.e.

$$\{(x,0,0) : x \in \mathbb{R}\} \oplus \{(0,y,0) : y \in \mathbb{R}\} = \{(x,y,0) : x,y \in \mathbb{R}\}$$

Notice that this is xy plane, and that each element of this resulting space has a unique representation. Then, add the subspace spanned by the z axis to the xy plane to make all of R^3:

$$\{(x,0,0) : x \in \mathbb{R}\} \oplus \{(0,y,0) : y \in \mathbb{R}\} \oplus \{(0,0,z) : z \in \mathbb{R}\} = \{(x,y,z) : x,y,z \in \mathbb{R}\} = \mathbb{R}^3$$

How to Build a Vector Space

Vector Space: an abelian group equipped with a field action

Field: a commutative ring with a multiplicative inverse for each non-zero element of the ring

An Algebra: a ring, A, equipped w/ a commutative unital ring R and a ring homomorphism that maps from R to Z(A) (the center of A) and maps 1_R to 1_A.

Ring: an abelian group with another operation (typically multiplication) that is assoc, has identity and is distributive

Commutative Ring: an abelian group with another operation (typically multiplication) that is assoc, has identity and is distributive and commutes

Abelian Group: a set that meets all conditions of a group + commutativity of the operation

Module: an abelian group equipped with a ring action

Group: a set with an operation that meets 4 conditions (closure, assoc., identity, inverse) - the operation is typically addition

Set - collection of objects (order does not matter)

Nullity or the Kernel

The null space of a linear map (i.e. a matrix) L is the subspace (it is in fact a subspace of the domain of L) of vectors that map to zero under L.

$$\mathrm{null}(L) = \{v \in V : Lv = 0\}$$

Another term for the null space is the kernel.

The nullity of L is the rank of its null space.

Example of Nullity

What is the nullity of the projection operator given by this matrix? Take the determinant of this matrix. What do you get? Why do you now know the nullity is not zero?

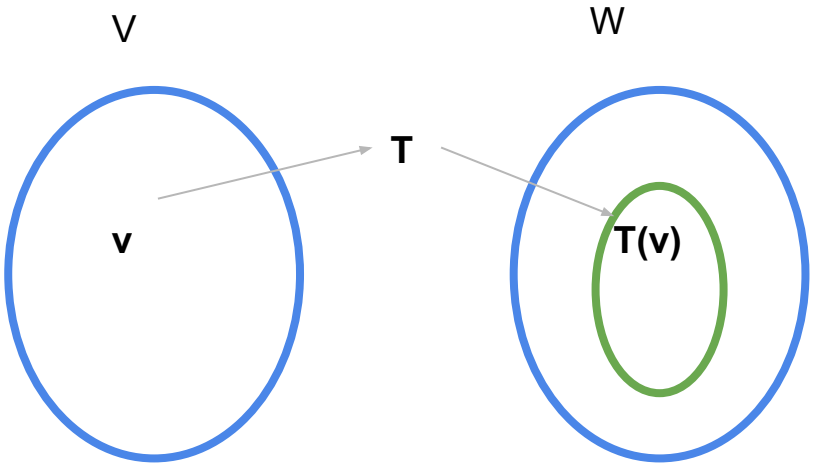
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Think about what this linear transform does to R^3? Which vectors get mapped to 0?

Image

The image of a map T is the set of vectors in the codomain of T which are the mappings of T applied to all vectors in the domain V.

If Im(T) = W then T is surjective (T maps onto W)
If all vectors in V map in a one-to-one manner T(v)s in W then T is injective (T maps into W)
If T is both surjective and injective it is bijective



Rank Nullity Theorem

For any linear map T from V to W,

$$\dim(V) = \operatorname{rank}(T) + \operatorname{nullity}(T)$$

where

$$\operatorname{rank}(T) = \dim(\operatorname{Im}(T)) \text{ and } \operatorname{nullity}(T) = \dim(\operatorname{null}(T))$$

Example of Rank-Nullity

Verify for yourself that two of the column vectors in this matrix are dependent:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Then, the rank cannot be 3, and so must be 2 or less. Then, after removing the vector you found that depends on the other two, verify that the remaining two are linearly independent.

So, this matrix has rank 2. By the Rank-Nullity Theorem, it must also have nullity 1, i.e. a 1-dimensional subspace of R^3 is mapped to 0 by this transformation. Which subspace is that? Hint: Who depends on who?

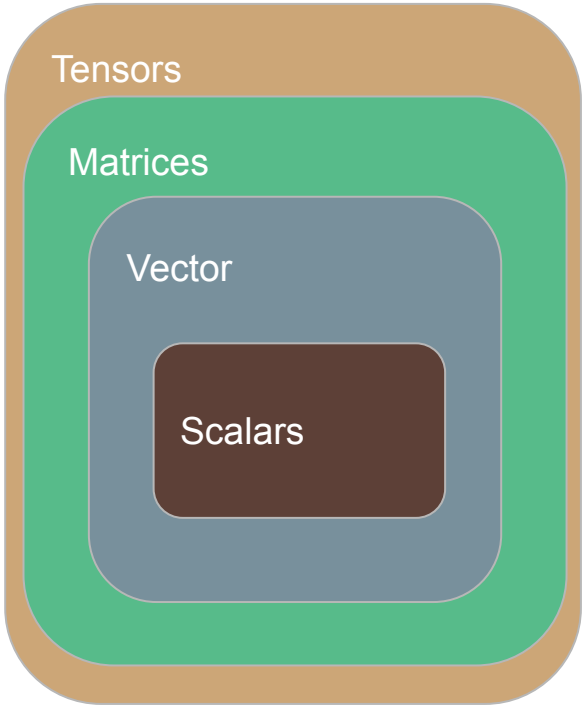
A More Interesting Example

What is the rank of this matrix? So, what is its nullity? How do you know?

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

What is a Tensor?

- Tensors are simply arrays of numbers, or functions, that transform according to certain rules under a change of coordinates.
- Tensor orders:
  - 0-tensor: scalar
  - 1-tensor: vector
  - 2-tensor: matrix
  - 3-tensor: a 3d box of numbers
  - *n*-tensor....



Tensor Product

- When computing the tensor product of two vectors from the same vector space of dim n_1 and n_2, the resulting vector has a dim of n_1*n_2
- Formula:

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 \times b_1 \\ a_1 \times b_2 \\ a_2 \times b_1 \\ a_2 \times b_2 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \cdot 1 \\ 1 \cdot 2 \\ 2 \cdot 1 \\ 2 \cdot 2 \\ 3 \cdot 1 \\ 3 \cdot 2 \end{pmatrix}$$

Tensor Product Example

$$\begin{pmatrix} 3 \\ 4 \end{pmatrix} \otimes \begin{pmatrix} 8 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \times 8 \\ 3 \times 3 \\ 4 \times 8 \\ 4 \times 3 \end{pmatrix} = \begin{pmatrix} 24 \\ 9 \\ 32 \\ 14 \end{pmatrix}$$

Tensor Product Example 2

$$\begin{pmatrix} -1 \\ 2 \\ 4 \end{pmatrix} \otimes \begin{pmatrix} 4 \\ 6 \end{pmatrix} = \begin{pmatrix} -4 \\ -6 \\ 8 \\ 12 \\ 16 \\ 24 \end{pmatrix}$$

Test for Comprehension

Compute the determinant of the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

is zero by brute force. What does this mean geometrically?

Afterwards, more elegantly discover the linear dependence between the column vectors of A and apply the result that the determinant of a matrix is zero iff its column vectors are linearly dependent.

Matrix Inversions and Solving Systems of Linear Equations

What is an Invertible Matrix

- A matrix **A** is invertible if another matrix **B** exists such that:

  $$\mathbf{A}\,\mathbf{B} = \mathbf{I} = \mathbf{B}\,\mathbf{A} \quad \textbf{(matrix multiplication)}$$

  Note: **I** always refers to the Identity Matrix and the inverse has to "work" on the right and left. This implies a matrix is invertible (with this definition) only if A is square!

- Matrices that are not invertible are called "singular".
- Remark: If **B** exists, it is unique (a matrix only has one inverse)
- Notation: The inverse of a matrix is denoted as $\mathbf{A}^{-1}$ (and called "**A** inverse")

Example of an Invertible Matrix

- Here is an example of Matrix **A** and its inverse

$$A = \begin{bmatrix} 2 & 1 \\ 4 & 5 \end{bmatrix} \; then \;\; A^{-1} = \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} \\ -\frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

- In order to prove that $\mathbf{A^{-1}}$ is truly the inverse of **A** all that is needed is to multiply it by **A**, and it should return **I**

$$\begin{bmatrix} 2 & 1 \\ 4 & 5 \end{bmatrix} \times \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} \\ -\frac{2}{3} & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Formula for the Inverse of a 2x2 Matrix

- For 2x2 matrices there is a shortcut for finding the inverse of a the matrix using the det

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Example of a Non-Invertible Matrix

- $A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$ Is not invertible. It has a det of 0.

- How can we prove that no such matrix exists?
  Let's assume that it does exists, so the following equation would have a solution:

$$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Continued…

Example of a Non-Invertible Matrix

- Continued… $$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- This should yield a system of equation (from Matrix Multiplication):

$$\begin{aligned} 1 \cdot a + 1 \cdot c &= 1 \\ 1 \cdot b + 1 \cdot d &= 0 \\ 2 \cdot a + 2 \cdot c &= 0 \\ 2 \cdot b + 2 \cdot d &= 1 \end{aligned}$$

- This system of equations is inconsistent, as equation 1 and 3 contradict each other (divide eq 3 by 2 to see this)
- Note: This is a method that is used to determine if a matrix is invertible

Singular Matrices

We say that a matrix is singular if it has no inverse.

Not invertible iff det=0

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

Example: Finding the Inverse of a Matrix

- Find the Inverse of Matrix A = $\begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$
- Let's start by creating the equation: $\begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix} \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- And the system of equations that results
  - 3a + 4c = 1
  - 3b + 4d = 0
  - 6a + 8c = 0
  - 6b + 8d = 1
- There is no inverse for this matrix because:
  - -2(3a + 4c = 1) = -6a + -8c = -2
  - 6a + 8c = 0
  - 0 ≠ -2

Example: Finding the Inverse of a Matrix

- Find the Inverse of Matrix A = $\begin{bmatrix} -1 & 1 \\ -2 & 0 \end{bmatrix}$
- Let's start by creating the equation: $\begin{bmatrix} -1 & 1 \\ -2 & 0 \end{bmatrix} \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- And the system of equations that results
  - -1a + 1c = 1
  - -1b + 1d = 0
  - -2a + 0c = 0
  - -2b + 0d = 1
- Do substitutions and…
  - a = 0
  - b = ½
  - c = 1
  - d = -½

Involution

A matrix is involutary iff it is its own inverse e.g. $A^2 = AA = I$

Or we can say that $A = A^{-1}$ if A is involutary.

Rotation by π is an involution The matrix for this involution is represented by

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

Euler's Formula

$$e^{i\frac{\pi}{2}} = \cos\left(\frac{\pi}{2}\right) + i \cdot \sin\left(\frac{\pi}{2}\right) = 0 + i \cdot 1 = i$$

Using Euler's Formula to Express Rotations

We can express a rotation by pi radians via Euler's formula within a matrix.

$$\begin{pmatrix} e^{i\pi} & 0 \\ 0 & e^{i\pi} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

Solving Systems of Linear Equations with Matrices

- How does this systems of equations translate into a matrix equation?

$$3x - y = 7$$
$$2x + 3y = 1$$

- Easy!

$$\begin{bmatrix} 3 & -1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 1 \end{bmatrix}$$

- This form is called the **Ax** = **b** form. All the coefficients go to the **A** matrix and the variables in the **x** vector. While the solutions go in **b** vector.

Solving Systems of Linear Equations

- Now let's solve

$$\begin{bmatrix} 3 & -1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 1 \end{bmatrix}$$

- First find the inverse of **A**:

$$\begin{bmatrix} \frac{3}{11} & \frac{1}{11} \\ \frac{-2}{11} & \frac{3}{11} \end{bmatrix}$$

- Then multiply by $\mathbf{A}^{-1}$ on both sides...

Solving Systems of Linear Equations

- Then multiply by **A**$^{-1}$ on both side and evaluate

$$\begin{bmatrix} \frac{3}{11} & \frac{1}{11} \\ \frac{-2}{11} & \frac{3}{11} \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{3}{11} & \frac{1}{11} \\ \frac{-2}{11} & \frac{3}{11} \end{bmatrix} \begin{bmatrix} 7 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

Row Equivalence and Similarity

We say that two matrices A and B are row-equivalent iff A can be transformed into B by a sequence of elementary row operations.

The elementary row operations include exchanging two rows, multiplying a row by a scalar, and adding two rows.

Two matrices A and B are **similar** iff there exists an invertible matrix X such that

$$B = X^{-1}AX$$

Conjugacy - meaning 1

Two matrices are **conjugate** if they are similar - we use “similar” in the context of matrices - conjugacy is a more a general term that can also apply to groups, etc.

$$B = X^{-1}AX$$

Conjugate (or similar) matrices share similar properties, such as the trace, determinant, invertibility, etc, so knowing two matrices are similar is quite powerful!

Conjugate - meaning 2

When we take the conjugate of a matrix we take the conjugate of each entry of a matrix and produce a new matrix with these complex numbers.

For example:

$$\begin{pmatrix} 1+i & 2-3i \\ 5+7i & 11-13i \end{pmatrix} \mapsto \begin{pmatrix} 1-i & 2+3i \\ 5-7i & 11+13i \end{pmatrix}$$

Hermitian Matrices

A Hermitian matrix is a matrix that is equal to its complex conjugate transpose.

Conjugate Transpose

- The conjugate transpose may be referred to as the “adjoint matrix” or “Hermitian conjugate”.
- An adjoint matrix is the conjugate transpose

  $A^* = A^H$

  $a_{ij} = \bar{a}_{ji}$
- First, take the transpose of matrix A

  $a_{ij} = a_{ji}$
- Then, take the conjugate of the imaginary number.

  $a + bi \rightarrow a - bi$
- This process is also known as:
  - Conjugate Transpose
  - Hermitian Conjugate

Technical Definition of Adjoint Operator

We say that a linear operator T^*: V to V is adjoint to a linear operator T: V to V iff for all v,w in V

$$\langle Tv, w \rangle = \langle v, T^* w \rangle$$

T is self-adjoint if T^* is equal to T, in which case T preserves inner products, and consequently, norms.

Example of a conjugate transpose

$$\begin{pmatrix} 1+i & 2-3i \\ 5+7i & 11-13i \end{pmatrix}$$

$$\mapsto \begin{pmatrix} 1-i & 2+3i \\ 5-7i & 11+13i \end{pmatrix}$$

$$\mapsto \begin{pmatrix} 1-i & 5-7i \\ 2+3i & 11+13i \end{pmatrix}$$

Test for Comprehension

Why is a real symmetric matrix equal to its conjugate transpose? So, why are these matrices Hermitian?

What are some examples of real symmetric matrices you might encounter in computer science?

Verify that the matrix $A = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 3 & 0 & 0 \end{pmatrix}$ is similar to the identity matrix via

a sequence of elementary row operations. Knowing this, we know this matrix is invertible, since invertibility is a conjugacy invariant!

Linear Transformations as Matrices

- How do we represent linear transformations as Matrices?
- How do we use this representation?
- The inverse of transformation
- Inverse of a Matrix as the Inverse Transformation
- Kernel
- Image
- Change in Kernel and Image for inverse.

Linear Maps and Matrices

Theorem: Every linear map (function) can be represented as a matrix and every matrix is a linear map.

Example: How we represent Linear Transformations as Matrices?

Linear Transformations can be represented as matrices by passing the unit basis vectors through the function and then putting the results in a matrix.

$$f\left(\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}\right) = \begin{pmatrix} x_0 + x_1 \\ x_1 \end{pmatrix}$$

$$f\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} 1 + 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$f\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} 0 + 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

Transforming through the Matrix

- We can now multiply any vector by this Matrix, and the function will be applied.

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

$$(1 \cdot 4) + (1 \cdot 2) = 6$$
$$(0 \cdot 4) + (1 \cdot 2) = 2$$

$$\begin{pmatrix} 6 \\ 2 \end{pmatrix}$$

Implications

A linear map (that maps from a domain to a codomain of equal dimension) that is injective must be surjective and vice -versa - this is NOT true of all functions.

Can you think of an injective map that is not surjective? The theory we've developed says we shouldn't be looking at linear maps!

How does the Inverse relate to the Kernel & Image

If $T: V \longrightarrow W$ where V and W have the same dimension

Then the following are equivalent:

ker(T) = 0 (T is injective) = null(T)=0

Im(T) = W (T is surjective)

Rank of T is full (T is surjective, the set {T(v): v in V} spans W)

det(T) ≠ 0

T is invertible

Test for Comprehension

What is the matrix associated to the linear transformation that rotates a vector in R^2 counterclockwise by 90 degrees, then scales it by 2?

Hint: Determine how this transformation acts on the the basis vectors (1,0) and (0,1), then construct the matrix with this knowledge.

Eigenvectors

- Eigenvector
- Eigenvalues
- Eigendecomposition

Eigenvalues and Eigenvectors

- Eigenvalues and eigenvectors of a matrix **A** are based on the equation:

$$\mathbf{Av} = \lambda\mathbf{v}$$

- **A** is the matrix, **v** is an eigenvector, and **λ** is the eigenvalue
- In others words, for a vector **v** to be an **eigenvector** of matrix **A**, **when A is applied to v it only scales v, it does not change the direction of v. It scales it by the factor λ.**
- In general, we focus on the right eigenvector; in other words, we apply **A** to a vector to the right of **A**. We can also determine the left eigenvector, but in practice, this is seldom done: $v^T A = \lambda v^T$

How Do We Find Eigenvectors?

Let A be any matrix and v be a nonzero vector such that

$$Av = \lambda v$$

for some λ in the real numbers, i.e. a nonzero vector that is scaled by the matrix A. Then, some arithmetic yields

$$\lambda v - Av = 0 \rightarrow (\lambda I - A)v = 0$$

In other words, v is a nonzero vector that λI - A maps to 0.

Remember that if a matrix maps a nonzero vector to 0, then it is not invertible, and so its determinant must be zero!

Characteristic Polynomial

The polynomial

$$\det(\lambda I - A)$$

is called the characteristic polynomial because it is “characteristic” of the matrix A. The equation where it’s set to zero is the characteristic equation.

Solving this equation for lambda yields the eigenvalues of A.

Computing the Eigenvalues of a Matrix

Let's compute the eigenvalues of this matrix M:

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

To do this, we solve the equation

$$\det\left(\lambda I - M\right) = 0$$

Eigendecomposition

- Just like integers can be broken up into prime factors, we can **decompose** matrices.

  $12 = 2 \cdot 2 \cdot 3$

- The eigendecomposition of a matrix, A, is a way of decomposing a matrix into a product of its eigenvectors and eigenvalues. $A=Q\Lambda Q^{-1}$ where A is a matrix, Q is the matrix whose column vectors are the eigenvectors of A and Λ is the diagonal matrix whose elements are the eigenvalues of A.
- Eigendecomposition can be thought of as an advantageous change of basis.

Towards the eigenvalues of M

$$\det(\lambda I - M) = \left| \begin{pmatrix} \lambda - 1 & 2 \\ 2 & \lambda - 1 \end{pmatrix} \right| = (\lambda - 1)^2 - 2^2 = 0.$$

So, the eigenvalues are -1 and 3.

To find the eigenvectors, solve the system

$$Mv = \lambda v$$

for some nonzero vector v and eigenvalue lambda.

Example

Let’s verify that the vector (1,1) is actually an eigenvector of the matrix

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} = 3 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Eigen

The word eigen means “own” or “self” in German. Hilbert was the first to use this terminology.

Eigenvectors and Eigenvalues

$Ax = \boldsymbol{\lambda} x$

- a matrix is non-invertible if it has an eigenvalue of 0
- the eigenvectors of any real symmetric matrix A are orthogonal
- any real symmetric matrix A is diagonalizable

See https://youtu.be/PFDu9oVAE-g

Computing the Eigenvalues of a Matrix

To solve for the eigenvalue λ and the corresponding eigenvector, x_i of an nxn matrix do the following:

1. Multiply the identity matrix by the scalar λ
2. Subtract the identity matrix from the matrix A
3. Find the determinant of the resulting matrix
4. Solve for the values of λ where $\det(A-\lambda I) = 0$
5. Solve for the corresponding vector to each λ

Example: Finding the Eigenvalues

$$A = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix}$$

1. Multiply the identity matrix by the scalar **λ**
2. Subtract the identity matrix from the matrix A
3. Find the determinant of the resulting matrix
4. Solve for the values of **λ** where det(A-**λ**I) = 0
5. Solve for the corresponding vector to each **λ**

$$1. \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \lambda = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$2. \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix}$$

$$3. det \begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix}$$
$$= (7-\lambda)(-1-\lambda) - (3)(3)$$
$$= \lambda^2 - 6\lambda - 16$$

$$4. = \lambda^2 - 6\lambda - 16 = 0\lambda = 8\lambda = -2$$

It is equivalent to do this for **λ**I-A, which might be easier.

Eigenvectors and Eigenvalues: Examples

So, any vector in R^2 is an eigenvector of the identity matrix, since any vector will be mapped to itself, any vector is an eigenvector of the identity matrix with eigenvalue 1.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 1 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

So, for example, (1,1) is an eigenvector of the identity matrix with eigenvalue 1.

Check that any other vector will also be an eigenvector of the identity matrix with eigenvalue 1.

Positive-Definite and Positive-Semidefinite

A matrix whose eigenvalues are all positive is positive-definite, while a matrix whose eigenvalues are all nonnegative (so, some may be zero) is positive-semidefinite.

The eigenvalues of the Hessian matrix (of second partial derivatives of a vector-valued function f) encode the convexity or concavity of a hypersurface (think of a surface, but in higher dimensions possibly).

If the Hessian matrix is positive-definite at some point, i.e. all of its eigenvalues are positive, the hypersurface has a local minimum at that point. Mutatis mutandis, if the Hessian is negative-definite (so, all eigenvalues are negative), the function has a local maximum there.

The Diagonalization of the Matrix M

Eigenspaces

The eigenspace associated to an eigenvalue is the subspace of the vector space on which the matrix acts spanned by the eigenvectors whose eigenvalues are that eigenvalue.

More precisely, the eigenspace of an eigenvalue lambda of a matrix A is

$$E_\lambda = \langle \{ v \in V : (\lambda I - A)v = 0 \} \rangle$$

where the angle brackets indicate the generation of the subspace by the contained set. Notice this is the kernel of lambda I -A!

When is a matrix diagonalizable?

Not every matrix is diagonalizable!

It turns out that a matrix is diagonalizable iff the sum of the dimensions of its eigenspaces is equal to the dimension of the space iff the eigenvectors of the matrix form a basis for the space.

A matrix may have repeated eigenvalues and still be diagonalizable! Also, if the eigenvalues of a matrix are distinct, the matrix is diagonalizable.

The Trace Operator

The Trace operator gives you the sum of all the diagonal entries of a matrix. Equivalently, the trace of a matrix is the sum of its eigenvalues.

So, $\mathrm{Tr}(A) = \sum A_{i,i}$ for all i from 1 to n where n is the dimension of A (where A is a square matrix).

Example 1:

$$\mathrm{Tr}(A) = -2$$

$$A = \begin{pmatrix} 2 & 1 & 1 \\ -5 & -3 & 0 \\ 1 & 1 & -1 \end{pmatrix}$$

Frobenius Norm (for matrices) and Distance

To compute the Frobenius norm of a matrix we first treat the matrix as if it were a 1-vector and then take the euclidean norm of that vector. We can think of this norm as measuring how far a matrix is from the zero matrix.

$$||A||_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2\right)^{\frac{1}{2}}$$

$$= \sqrt{\mathrm{Tr}\left(AA^\dagger\right)}$$

Singular Values

How do you decompose a matrix which is not square? A non-square matrix cannot be decomposed using eigenvectors and eigenvalues.

Instead, we perform a set of operations that are analogous to eigendecomposition.

The singular values of a matrix A are the eigenvalues of the matrix resulting from taking square root of the product of the conjugate transpose of A with A, i.e. the matrix $(AA^*)^{.5}$

Singular Value Decomposition

The SVD of the matrix A is a product $A=UDV^T$ where U is a matrix whose columns are the left singular vectors of A and V is the matrix whose columns are the right singular vectors of A and D (which is not necessarily a diagonal matrix) is the matrix whose diagonal entries are the singular values of the matrix A.

Note that AA^T and A^TA are always square matrices.

The left singular vectors of A are the eigenvectors of AA^T.

The right singular vectors of A are the eigenvectors of A^TA.

Geometric Interpretation of the SVD

The singular value decomposition of a linear transformation describes it in three steps: a rotation V^*, a dilation in the principal axes Σ, and a final rotation U.

Singular Value Decomposition Example

Let's compute the singular value decomposition of the transformation A from R^2 to R^4 below:

$$\begin{pmatrix} 2 & 4 \\ 1 & 3 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Computation of the SVD

Recall that the SVD of M is defined as $U\Sigma V^*$, where U is the matrix whose column vectors are the eigenvectors corresponding to the nonzero eigenvalues of MM^*, V is the matrix whose column vectors are the eigenvectors corresponding to the nonzero eigenvectors of M^*M, and Σ is the rectangular matrix whose diagonal entries (as many as possible) are the square roots of the nonzero eigenvalues of MM^*.

So, first we'll need to compute the matrices MM^* and M^*M.

Computation of the SVD continued

You should find each of MM^* and M^*M to be

$$\begin{pmatrix} 20 & 14 & 0 & 0 \\ 14 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} 8 & 20 \\ 5 & 13 \end{pmatrix}$$

We'll compute the eigenvalues of M^*M (since it's of lower dimension and the eigenvalues of M^*M are the same).

Let's find the eigenpairs...

You should find the eigenvalues of MM^* to be

$$\frac{1}{2}(21 \pm 5\sqrt{17})$$

To find the eigenvectors of MM^*, we'll need to solve the matrix equation

$$M^*Mv = \lambda v \iff (\lambda I - M^*M)v = 0$$

for some nonzero vector v. So, we compute

$$\lambda I - M^*M = \tfrac{1}{2}(21 \pm 5\sqrt{17})I - M^*M = \tfrac{1}{2}(21 \pm 5\sqrt{17})I - \begin{pmatrix} 8 & 20 \\ 5 & 13 \end{pmatrix}$$

Towards the eigenvectors...

You should find that

$$\tfrac{1}{2}(21 \pm 5\sqrt{17})I - \begin{pmatrix} 8 & 20 \\ 5 & 13 \end{pmatrix} = \begin{pmatrix} \tfrac{1}{2}(21 \pm 5\sqrt{17}) - 8 & \tfrac{1}{2}(21 \pm 5\sqrt{17}) - 20 \\ \tfrac{1}{2}(21 \pm 5\sqrt{17}) - 5 & \tfrac{1}{2}(21 \pm 5\sqrt{17}) - 13 \end{pmatrix}$$

Towards the eigenvectors

The eigenpairs of this matrix are

$$\lambda_1 = \frac{1}{2}(21 + 5\sqrt{17}), v_1 = \begin{pmatrix} \frac{1}{2}(-1 - \sqrt{17}) \\ 1 \end{pmatrix}$$

$$\lambda_2 = \frac{1}{2}(21 - 5\sqrt{17}), v_2 = \begin{pmatrix} \frac{1}{2}(-1 + \sqrt{17}) \\ 1 \end{pmatrix}$$

The eigenpairs of MM^*

$$\Lambda_1 = 15 + \sqrt{221}, w_1 = \begin{pmatrix} \frac{1}{14}(5+\sqrt{221}) \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad \Lambda_2 = 15 - \sqrt{221}, w_2 = \begin{pmatrix} \frac{1}{14}(5-\sqrt{221}) \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\Lambda_3 = 0, w_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad \Lambda_4 = 0, w_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

The Matrix U

So, the matrix U (whose columns are the eigenvectors of MM^*) in the decomposition is

$$\begin{pmatrix} \frac{1}{14}(5+\sqrt{221}) & \frac{1}{14}(5-\sqrt{221} & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The Matrix V

... and the matrix V (whose columns are the eigenvectors of M^*M) is

$$\begin{pmatrix} \frac{1}{2}(-1-\sqrt{17}) & \frac{1}{2}(-1+\sqrt{17}) \\ 1 & 1 \end{pmatrix}$$

The Matrix Σ

... and since the eigenvalues of MM^* (equivalently, M^*M) have been found to be

$$\frac{1}{2}(21 \pm 5\sqrt{17})$$

the matrix Σ of singular values is the matrix whose diagonal entries (as many as possible) are the square roots of the eigenvalues, i.e.

$$\begin{pmatrix} \sqrt{\frac{1}{2}(21+5\sqrt{17})} & 0 \\ 0 & \sqrt{\frac{1}{2}(21-5\sqrt{17})} \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Putting it all together

So, the SVD of M is $U\Sigma V^*$, which, after normalization of the column vectors of U and V^* (so, N is the normalization factor which scales the column vectors of each to unit length vectors), is

$$U\Sigma V^* = \frac{1}{N}\begin{pmatrix} \frac{1}{14}(5+\sqrt{221}) & \frac{1}{14}(5-\sqrt{221} & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{\frac{1}{2}(21+5\sqrt{17})} & 0 \\ 0 & \sqrt{\frac{1}{2}(21-5\sqrt{17})} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{2}(-1-\sqrt{17}) & 1 \\ \frac{1}{2}(-1+\sqrt{17}) & 1 \end{pmatrix}$$

Singular Value Decomposition Example

Let's compute the singular value decomposition of the shear transformation M below. As a linear transformation of R^2, the shear map looks like:

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

Pseudoinverse

The Moore-Penrose Pseudoinverse of a matrix A is

$A^{+} = VD^{+}U^{T}$

- UDV is the SVD of A, i.e. A = UDV,
- D+ is obtained by producing the transpose of the matrix whose diagonal elements are reciprocals of the diagonal elements of D.

Cauchy-Schwarz Inequality

The Cauchy-Schwarz Inequality states that for any vectors **u** and **v** in an inner product space (so that we may take norms)

$$\left|\mathbf{u} \cdot \mathbf{v}\right| \leq \left|\left|\mathbf{u}\right|\right| \cdot \left|\left|\mathbf{v}\right|\right|$$

This makes perfect sense when compared with an identity derived earlier regarding cosine, since the absolute value of cosine is less than or equal to 1.

$$\frac{\mathbf{u}\cdot\mathbf{v}}{\left|\left|\mathbf{u}\right|\right|\cdot\left|\left|\mathbf{v}\right|\right|} = \cos(\theta) \Rightarrow \left|\frac{\mathbf{u}\cdot\mathbf{v}}{\left|\left|\mathbf{u}\right|\right|\cdot\left|\left|\mathbf{v}\right|\right|}\right| = \left|\cos(\theta)\right| \leq 1$$

$$\Rightarrow \left|\mathbf{u} \cdot \mathbf{v}\right| \leq \left|\left|\mathbf{u}\right|\right| \cdot \left|\left|\mathbf{v}\right|\right|$$

The Triangle Inequality

The Cauchy-Schwarz Inequality gives rise to one of the most famous (and intuitive) inequalities in mathematics, the triangle inequality!

It states that for any vectors **u** and **v**,

$$||\mathbf{u} + \mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||$$

We can prove this with the usual trick of considering the squared norms...

A Proof of the Triangle Inequality via Cauchy-Schwarz

To demonstrate that $||\mathbf{u}+\mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||$, it suffices to show that the inequality is true for the squared norms, i.e. $||\mathbf{u}+\mathbf{v}||^2 \leq (||\mathbf{u}|| + ||\mathbf{v}||)^2$, but this is just a computation and application of Cauchy-Schwarz...

$$
\begin{aligned}
||\mathbf{u}+\mathbf{v}||^2 &= (\mathbf{u}+\mathbf{v}) \cdot (\mathbf{u}+\mathbf{v}) \\
&= ||\mathbf{u}||^2 + 2\mathbf{u} \cdot \mathbf{v} + ||\mathbf{v}||^2 \\
&\leq ||\mathbf{u}||^2 + 2||\mathbf{u}||||\mathbf{v}|| + ||\mathbf{v}||^2 \\
&= \big(||\mathbf{u}|| + ||\mathbf{v}||\big)^2
\end{aligned}
$$

Intuition Behind the Triangle Inequality

We said it was intuitive!

PCA

Principal **C**omponents **A**nalysis is a method to find patterns in a dataset. Specifically, we can use PCA to reduce a data set substantially without losing significant information.

We do this by finding the main drivers of subsets of the dataset and eliminating portions of the data that are not part of these main driving portions.

Dim Reduction with PCA

We can use PCA to reduce the dimensionality of the data. In this example, we find the plane defined by the two largest components and discard all other data, thereby reducing the dataset from 3D to 2D

Low vs. High Redundancy

Performing PCA

To perform PCA on a data set, we compute the eigendecomposition of the covariance matrix of the data set.

Often, we even compute the SVD.

The eigenpairs represent the directions of the principal components of the n-dimensional ellipsoid fitting the data set.

Try it Yourself: Principal Component Analysis

- First, calculate and graph principal components for the iris dataset “manually” for understanding.

- Second, check out the efficient sklearn version.

- Link: Colab

Bilinearity and Multilinearity

- A bilinear function is a binary function in that it takes two inputs and maps them to a co-domain.
- In the case of a bilinear functional, it takes two vectors and maps them both to the ground field - in other words the result is a scalar and element of the ground field.
- A multilinear functional does this for any number of input vectors, n.

$$af((u,v)) = f((au,v)) = f((u,av))$$

$$f((u+v,w)) = f((u,w)) + f((v,w))$$

$$f((u,v+w)) = f((u,v)) + f((u,w))$$

$$f((u+v,w+x)) = f((u+v,w)) + f((u+v,x)) = [f((u,w)) + f((v,w))] + [f((u,x)) + f((v,x))]$$

Antilinearity aka Conjugate Linearity

A function is said to be antilinear if the function applied to a sum of inputs is equal to the sum of the conjugate of the coefficents applied to each input and then input into the function.

$$f(ax + by) = \bar{a}f(x) + \bar{b}f(y)$$

Tensors as Functionals

We can think of a tensor product of two functionals T* and S* as a bilinear functional

$$T^* \otimes S^*$$

which accepts two dual vectors u and v and outputs the product of the result of applying S* to u and T* to v:

$$(T^* \otimes S^*)(u, v) = (S^* u)(T^* v)$$

Hadamard product of matrices

In contrast with usual matrix multiplication (which is essentially a series of dot products of each row of the first matrix dot product with the respective column of the second matrix), the Hadamard product is the elementwise multiplication of each element of the first matrix with the respective element of the second matrix.

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \odot \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \times 3 \\ 2 \times 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$$

Numpy for Hadamard

```
y = np.array([1,2,3])
x = np.array([2,3,4])
y * x = [2, 6, 12]
```

Linear Algebra in the Real World

Linear Algebra Applications: Deep Learning

Deep Learning: Weights and Biases

- In deep learning, we use weights and biases to determine the output of a neuron.
- Linear algebra speeds the computation of the inputs, weights and biases
- Weights are represented in a weight matrix
- Biases are represented in a vector.
- We perform a matrix multiplication between a matrix containing a batch of flattened input data and the weight matrix
- After which, we add (broadcasting addition) the bias vector to the matrix.

An explanation of Weights and Biases

An image can be thought of as a matrix whose entries are pixel intensities. We can consider the column vector whose entries are all entries of that matrix. Thus, two images can be thought of as two column vectors and can be compared via the dot product. Images (vectors) whose dot product is close to 1 are considered similar, and images whose dot product is close to 0 are considered dissimilar.

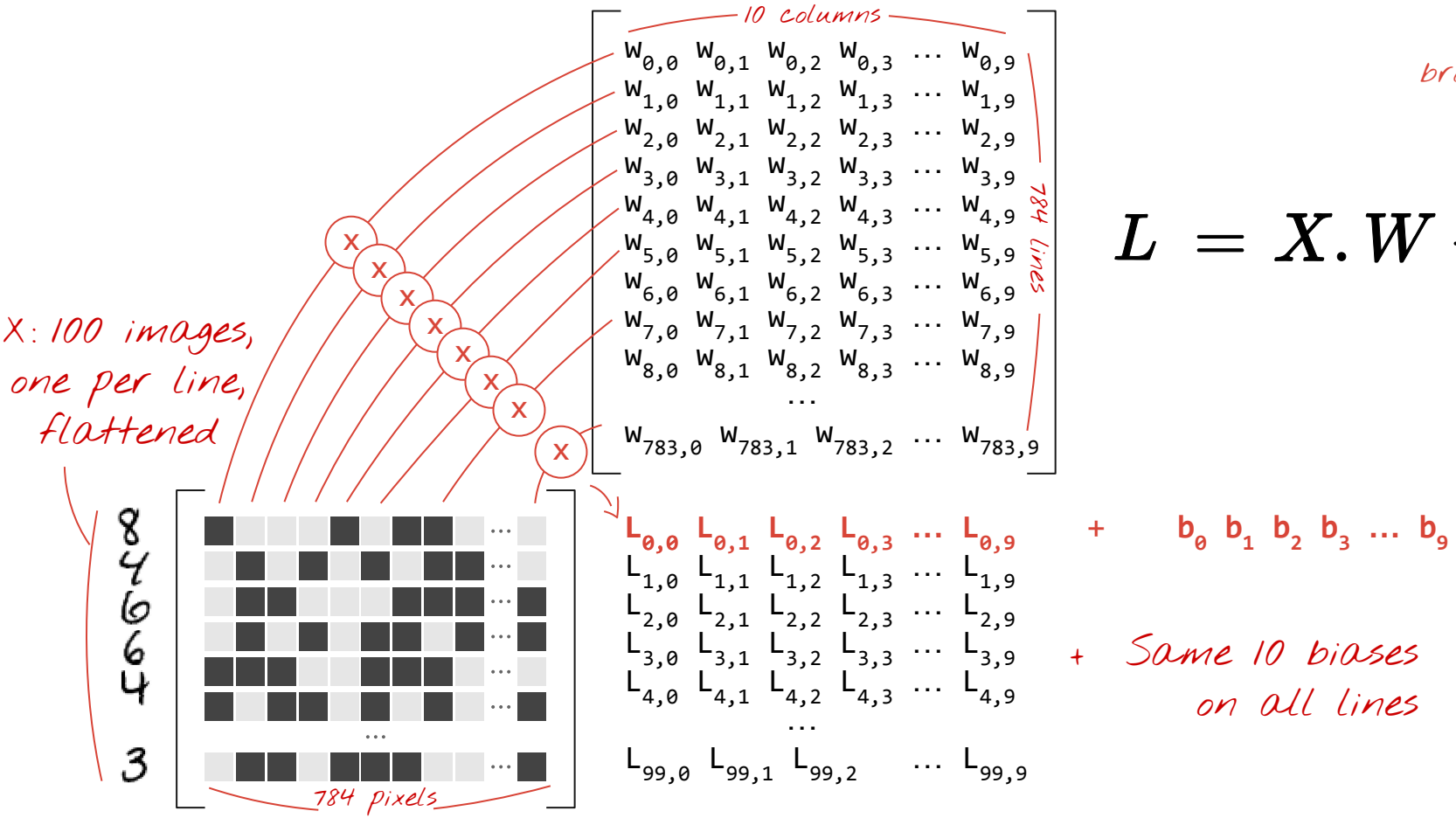
Turning an image into a vector

- We can take this 28 X 28 pixel and convert it into a 784 dimensional vector.

- Then we can take two such image vectors and compare them using the dot product

$$(0, 0, 0, \ldots, 123, 145, 1679, 120, 12, \ldots, 0, 0, 0)$$

Deep Learning: Weights and Biases



$$L = X.W + b$$

Deep Learning: Weights and Biases

Predictions

Y[100, 10]

Images

X[100, 784]

Weights

W[784,10]

Biases

b[10]

$$Y = softmax(X.W + b)$$

applied line by line

matrix multiply

broadcast on all lines

tensor shapes in [ ]

Deep Learning: Activation Function

- After we sum the dot products of the inputs and weights, we apply an activation function.
- Activation functions normalize the output of the node
- This is done in order to insure that only the relative outputs are considered.
- There are several choices of activation functions
- **Often ReLU is used as that yields good results**

Deep Learning: Gradient Descent

Try it Yourself: Label Handwritten Digits

- The task of autonomously understanding handwritten digits is ubiquitous in our lives.

- From the mailroom to law enforcement this technology makes our society more efficient.

- Try it yourself

See https://www.youtube.com/watch?v=Ilg3gGewQ5U

Adjacency Matrices

Any graph has an associated adjacency matrix A, where A_{ij} is the weight of the edge from vertex j to vertex i.

If there is no edge from vertex j to vertex i, A_ij=0.

If there are no weights, we assume the weights are 1.

More precisely,

$$A_{ij} = \begin{cases} 1 & \text{if } (v_j, v_i) \in \text{edges} \\ 0 & \text{if } (v_j, v_i) \notin \text{edges} \end{cases}$$

So, what's the adjacency matrix for that graph?

Check that it's

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Summary

- We have discussed the key objects and tools of linear algebra in this course.
- Linear Algebra offers a very powerful set of tools for many fields including: **computer science, machine learning** and **quantum physics**
- In **deep learning**, in particular, we use linear algebra to speed up calculations on large sets of data (input and parameter weights of neural networks) for learning and inference.
- We encourage the reader to continue delving into linear algebra via the resources on the next slide.

Links to Further Linear Algebra Material

- MIT Courseware: https://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/index.htm Gilbert Strand, Intro to linear algebra, http://math.mit.edu/~gs/linearalgebra/
- Technion M1 Algebra
- Deep Learning by Ian Goodfellow (Chapter 2 Section 1) http://www.deeplearningbook.org/contents/linear_algebra.html
- 3Brown1Blue: Linear Algebra
- Matrix Cookbook
- Linear Algebra Done Right

Advanced Topics

Characteristic Polynomial

The characteristic polynomial of the matrix

$$A = \begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix}$$

is the determinant of

$$\lambda I - A = \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} \lambda - 2 & -1 \\ 1 & \lambda \end{pmatrix}$$

i.e.

$$|\lambda I - A| = \left| \begin{pmatrix} \lambda - 2 & -1 \\ 1 & \lambda \end{pmatrix} \right| = (\lambda - 2)\lambda - 1(-1) = (\lambda - 1)^2$$

Jordan Normal Form

The Jordan normal (or canonical) form of a matrix a block diagonal matrix - so, a direct sum of matrices, each of which is in what is known as a “Jordan block”.. Each “Jordan block” is nearly a diagonal matrix: the eigenvalues associated to the eigenspace corresponding to that Jordan block form the diagonal, and ones populate the entry above each diagonal entry.

Observe that in the special case where a matrix has distinct eigenvalues, the eigenspaces are one-dimensional, and the Jordan blocks are simply the eigenvalues themselves.

A bit more about the Jordan Form

So, we realize the Jordan form is a generalization of the eigendecomposition of a matrix. If the matrix is diagonalizable, its Jordan form will be the lambda matrix in the eigendecomposition.

If a matrix is not diagonalizable (this can happen if the eigenvectors of the matrix do not span the space), then the Jordan form is the next best thing we have.

Homomorphism

$\mathrm{Hom}_F(V,F)$

In the context of linear algebra, a homomorphism is a linear map.

More generally, a homomorphism is a function between algebraic structures that preserves their structure.

Then, what's a *homeo*morphism?

Homeomorphism

A *homeo*morphism is a morphism of topological spaces.

More specifically, a homeomorphism f between two topological spaces X and Y is a continuous function with a continuous inverse function.

Intuitively, a homeomorphism is a bijection between X and Y that may bend, stretch, and twist things, but does not break, tear, or rip them.

Homeomorphism should not be confused with homotopy, a weaker notion of topological equivalence.

Homeomorphism

For example, a topologist might not be able to tell a coffee mug and a donut (more precisely, solid torus) apart, since they are homotopic, and even homeomorphic.

Jacobian

For a vector-valued smooth map $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$, determined by m smooth coordinate functions f^1, f^2, ..., f^m, the Jacobian (or derivative) of f is the matrix of partial derivatives

$$\begin{pmatrix} \frac{\partial f^1}{\partial x^1} & \frac{\partial f^1}{\partial x^2} & \cdots & \frac{\partial f^1}{\partial x^n} \\ \frac{\partial f^2}{\partial x^1} & \frac{\partial f^2}{\partial x^2} & \cdots & \frac{\partial f^2}{\partial x^n} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial f^m}{\partial x^1} & \frac{\partial f^m}{\partial x^2} & \cdots & \frac{\partial f^m}{\partial x^n} \end{pmatrix}$$

The determinant of the Jacobian matrix tells us information about the function f...

The Geometry of the Jacobian Matrix

The sign of the determinant of the Jacobian matrix encodes whether f is orientation-preserving or orientation-reversing.

The matrix (linear map) itself tells us how f is distorting space locally (in a small neighborhood about each point in the domain).

The Jacobian also gives rise to another matrix of exceptional importance in machine learning - in particular gradient descent methods!

Hessian Matrix

- Suppose f : ℝn → ℝ is a function taking as input a vector x ∈ ℝn and outputting a scalar f(x) ∈ ℝ; if all second partial derivatives of f exist and are continuous over the domain of the function, then the Hessian matrix H of f (at a point x in Rn) is a square, real, symmetric n×n matrix, usually defined and arranged as follows:
- The determinant of the above matrix is also sometimes referred to as the Hessian.
- The Hessian matrix is related to the Jacobian matrix by H(f(x)) = J(∇f(x))^T.

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

Wedge Product

The wedge product is a way of "multiplying" for example two 1-dimensional vectors to make a "bivector". A bivector is a 2-dimensional area. (But the area is oriented ... a weird concept, but check the diagrams)

Then if you wedged a bivector to a regular vector, you would get a trivector (an object with 3-dimensional oriented volume). And so on up.

Cross Product

- The cross product of **a** and **b**
  - Returns a vector (not a scalar like the dot product)
  - It is perpendicular to both **a** and **b**
  - Magnitude of the vector is the area of the parallelogram formed from **a** and **b**
  - The direction of the vector is defined by the right hand rule
  - The cross product is anticommutative: **a** x **b** = **- b** x **a**

Computing the Cross Product

- The cross product is defined as: $$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

- Example: $$\begin{bmatrix} 4 \\ -4 \\ 2 \end{bmatrix} \times \begin{bmatrix} 2 \\ 11 \\ 3 \end{bmatrix} = \begin{bmatrix} -4 \times 3 - 2 \times 11 \\ 2 \times 2 - 4 \times 3 \\ 4 \times 11 - -4 \times 2 \end{bmatrix} =$$

$$\begin{bmatrix} -12 - 22 \\ 4 - 12 \\ 44 - -8 \end{bmatrix} = \begin{bmatrix} -34 \\ -8 \\ 36 \end{bmatrix}$$

I J K Method for Computing Cross-Product

$$\overrightarrow{PQ} \times \overrightarrow{PR} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & 1 & 1 \\ 1 & -1 & 3 \end{vmatrix} = 4\vec{i} + 1\vec{j} - 1\vec{k}$$

Proving a Transformation is Linear

The transformation $f(\begin{pmatrix} \chi_0 \\ \chi_1 \end{pmatrix}) = \begin{pmatrix} \chi_0 + \chi_1 \\ \chi_0 \end{pmatrix}$ *is* a linear transformation.

The way we prove this is to pick arbitrary

$$\alpha \in \mathbb{R}, \quad x = \begin{pmatrix} \chi_0 \\ \chi_1 \end{pmatrix}, \quad \text{and} \quad y = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix}.$$

with which we then show that

- $f(\alpha x) = \alpha f(x)$; and
- $f(x + y) = f(x) + f(y)$.

Proof (Cont)

Show $f(\alpha x) = \alpha f(x)$:

$$\begin{aligned} f(\alpha x) &= f(\alpha \begin{pmatrix} \chi_0 \\ \chi_1 \end{pmatrix}) = f(\begin{pmatrix} \alpha\chi_0 \\ \alpha\chi_1 \end{pmatrix}) \\ &= \begin{pmatrix} \alpha\chi_0 + \alpha\chi_1 \\ \alpha\chi_0 \end{pmatrix} = \begin{pmatrix} \alpha(\chi_0 + \chi_1) \\ \alpha\chi_0 \end{pmatrix} \end{aligned}$$

and

$$\begin{aligned} \alpha f(x) &= \alpha f(\begin{pmatrix} \chi_0 \\ \chi_1 \end{pmatrix}) = \alpha \begin{pmatrix} \chi_0 + \chi_1 \\ \chi_0 \end{pmatrix} \\ &= \begin{pmatrix} \alpha(\chi_0 + \chi_1) \\ \alpha\chi_0 \end{pmatrix} \end{aligned}$$

so that both $f(\alpha x)$ and $\alpha f(x)$ evaluate to the same expression.

Show $f(x + y) = f(x) + f(y)$:

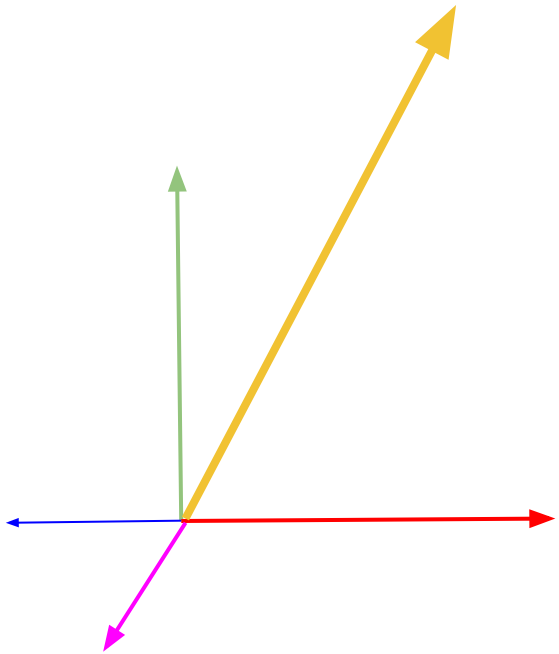
$$\begin{aligned} f(x+y) &= f(\begin{pmatrix} \chi_0 \\ \chi_1 \end{pmatrix} + \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix}) = f(\begin{pmatrix} \chi_0 + \psi_0 \\ \chi_1 + \psi_1 \end{pmatrix}) \\ &= \begin{pmatrix} (\chi_0 + \psi_0) + (\chi_1 + \psi_1) \\ \chi_0 + \psi_0 \end{pmatrix} \end{aligned}$$

and

$$\begin{aligned} f(x) + f(y). &= f(\begin{pmatrix} \chi_0 \\ \chi_1 \end{pmatrix}) + f(\begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix}) \\ &= \begin{pmatrix} \chi_0 + \chi_1 \\ \chi_0 \end{pmatrix} + \begin{pmatrix} \psi_0 + \psi_1 \\ \psi_0 \end{pmatrix} \\ &= \begin{pmatrix} (\chi_0 + \chi_1) + (\psi_0 + \psi_1) \\ \chi_0 + \psi_0 \end{pmatrix}. \end{aligned}$$

SparseML with Linear Algebra

Sparse Algebra and L^1 Norm



Yellow = Red + Blue + Green - Pink
(1,1,1,-1) = Non sparse

Yellow = 0*Red + 0*Blue + 0* Green - 3*Pink
(0,0,0,-3) = Sparse

L1 Vs L2 Norm Tangents - Principle of Sparsity

Tangents touch in **2D (non-sparse)** with high probability (all but 4 directions: 90*n)

Tangents touch in **1D (sparse)** with high probability (all but 4 directions: 45*n)

Images are Patches - Vectors in High Dimensions

Images are seen as a collection of patches. JPEG uses 8x8 non-overlapping ones (blocks), but we consider all possible (8x8 for example) patches, so there is overlap. You process (denoise) patches and then put them back together in the image.

Sparse ML

Example:
n=64 (8x8 patch)
m=256 (“overcomplete dictionary”)
3 dimensional subspace in 64 dimensions

How many options for each patch?
3 choose 64! -
The 64 dimensional space covered by a lot of 3 dimensional ones…

Use sparsity for selecting:

Quantum Mechanics and Computation

Functionals

A functional or linear form is a function which maps an element of a vector space to the ground field of that vector space. These are also called co-vectors.

For example, for the vector **a** = $\begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$ in vector space V over field R, there is a linear map, or functional, (1 0 0) which, using the dot product, maps **a** to the number 3 which is in field R.

You can construct a linear map for any vector in the vector space. These maps can be surjective (such as the linear map (1 0 0)) but they cannot be injective (since we are mapping from a higher dimension to a lower dimension).

Dual space

The dual space of a vector space, V over F (where F is R or C), is the vector space of all linear maps or transformations from V to F, i.e., all the functionals of V. The functionals are represented as 1xd row vectors in the dual space.

The basis for the dual space is made up of the co-vectors of the basis vectors of V. If V has a basis $\{e_1.....e_n\}$, then the dual basis is $\{e_1^*.....e_n^*\}$, where each e_i^* is defined on the basis for V via $e_i^*(e_j)=\delta_{i,j}$ =1 if i=j, 0 else.

Using Linear Algebra: Quantum Physics

See https://www.youtube.com/watch?v=MzRCDLre1b4&t=9s

Kronecker delta

The Kronecker delta function simply asks whether two indices are the same. If they are, its output is 1, if not, 0.

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Linear Algebra Applications: Quantum Physics

- Paul Dirac introduced a notation system to represent states in quantum mechanics (Dirac 1930)
- In dirac notation, a state of a system is represented as a ket in bra-ket notation) with the bra representing the dual of the state.
  - If we enclose a label in <label| that is a bra and |label> is a ket
  - |0> is the ket with the label 0 which refers to the state 0 of, say, a quantum bit (qubit).
  - Similarly, the following dirac notation $1/\sqrt{2}|0> +1/\sqrt{2}|1>$ represents the superposition of states 0 and 1 for a certain qubit.
- A Ket with a label such as |0> represents a vector in a vector space called Hilbert Space (defined in previous slides).

Ket examples

If we have a particle with spin up or spin down, we can represent these two states as either $|\uparrow>$ or $|\downarrow>$. If the state is definite then we use just one of these kets. If the particle is in a superposition of these two states we can represent this as $\frac{1}{\sqrt{2}}(|\uparrow> + |\downarrow>)$

In this formulation, the coefficient is known as the amplitude and the amplitude ($\frac{1}{\sqrt{2}}$) squared is equal to the probability of the measurement resulting in that state. This is the Born rule.

A measurement must result in one state or the other so the probabilities must add up to 1. In this case , the probability of each is 50% (which is the square of the amplitudes) and 2 x .5 = 1.

Tensor products and QM

To represent a pair of entangled particles we use the tensor product.

So $\frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B)$ is a two particle entangled system where each particle is in a superposition of 0 and 1. Measurement of one particle (A) will force the state to definite of B.

The Hilbert space for this entangled pair is 4 dimensional. Since the initial Hilbert space of each particle is 2 dimensional, the Hilbert space of the tensor product of the two vectors representing the two particles is 4 dimensional.

Inner products and Quantum States

We can perform an inner product on two quantum states. So <0|1> is the inner product of state 0 and state 1. This would equal 0 since the two states are orthogonal to each other. Thus the angle between them is 90 degrees and since cos(90) degrees is 0 the dot product is zero.

If we took the inner product <1|1> then this would equal 1 since the two states are collinear.

The Brief History of Quantum Computing

- 1979 - Paul Benioff details the principles of quantum computing
- 1980 - Manin publishes Computable and Non-Computable, a book which describes quantum computers
- 1981 - Feynman proposed the idea of creating machines based on the laws of quantum mechanics instead of the laws of classical physics.
- 1985 - David Deutsch developed the quantum Turing machine, showing that quantum circuits are universal.
- 1994 - Peter Shor develops a quantum algorithm to factor very large numbers in polynomial time.
- 1997 - Lov Grover develops a quantum search algorithm with O(N^.5) complexity.
- 1999 - Yasunobu Nakamura realizes the first superconducting qubit.
- 2000 - David DiVincenzo publishes the basic requirements of a full-fledged quantum computer.
- 2001 - Isaac Chuang et. al. demonstrate the first realization of Shor's algorithm on a nuclear magnetic resonance platform.

Linear Algebra Applications: Quantum Physics

- Paul Dirac introduced a notation system to represent states in quantum mechanics (Dirac 1930)
- In dirac notation, a state of a system is represented as a ket in bra-ket notation) with the bra representing the dual of the state.
  - If we enclose a label in <label| that is a bra and |label> is a ket
  - |0> is the ket with the label 0 which refers to the state 0 of, say, a quantum bit (qubit).
  - Similarly, the following dirac notation $1/\sqrt{2}|0> +1/\sqrt{2}|1>$ represents the superposition of states 0 and 1 for a certain qubit.
- A Ket with a label such as |0> represents a vector in a vector space called Hilbert Space (defined in previous slides).

Different architectures of QC

There are several different approaches to implementing QC including:

1. Superconducting QC
2. Trapped ion
3. Cold Atom Lattice
4. Topological
5. Spin qubits

Applying an operator to a state

If we represent the state |0> as a column vector (1 0) and then wish to apply an operator to it, we can represent the operator as a matrix and simply do matrix-vector multiplication. For example, to apply NOT to the state |0>:

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

.

Identity Operator

The Identity operator simply returns the same state

NOT operator

Rotation Operator - Pauli Y

Z Operator

The Z operator is defined as:

Z anti-commutes with the X operator so that

$ZX = -XZ$

and

Pauli Gates

- Pauli-X
- Pauli-Y
- Pauli-Z

Hadamard operator

The Hadamard gate operates on a single qubit. The state of a single qubit can be described as $\alpha|0\rangle+\beta|1\rangle$, where $|\alpha|^2+|\beta|^2=1$. If you measure the qubit, the output is 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.

We use the H operator to put a qubit into a superposition of two states.

So the {0> state is a state of definite spin in the Z direction. By applying H we move it to a superposition of |0>+|1> (which is also the definite spin state in the X direction

Square root of NOT operator (NOT)

We can construct operators which when applied twice in succession are equivalent to our standard operators.

Controlled NOT (CNOT)

The CNOT gate is a 2-qubit operator which takes in a control bit and target bit. If the control bit is 0, the target bit is left unchanged. If the control bit is 1 then the target bit is flipped. We use the CNOT gate to entangle two qubits.

Toffoli operator

This is a ternary operator. It uses the boolean AND on the two control qubits. We apply X in one of the eight cases

$|000\rangle \rightarrow |000\rangle$

$|001\rangle \rightarrow |001\rangle$

$|010\rangle \rightarrow |010\rangle$

$|011\rangle \rightarrow |011\rangle$

$|100\rangle \rightarrow |100\rangle$

$|101\rangle \rightarrow |101\rangle$

$|110\rangle \rightarrow |110\rangle$

$|111\rangle \rightarrow |110\rangle$

The Inner Product

A number of courses confuse dot product with inner product. To be precise we should take care here to define a dot product as a type of inner product (in fact it is the standard or usual inner product), but the concept of inner product is a more general one. We can have inner products that are NOT dot products.

We denote an inner product like this: $\langle .,. \rangle$

An inner product is a binary operation (in the sense that it takes in two inputs). An inner product produces a scalar (real or complex). $\langle .,. \rangle : V \times V \longrightarrow F$

An inner product is a map from the vector space X the vector space to the underlying field.

An inner product is defined to satisfy the following axioms:

Conjugate symmetry: $\langle x,y \rangle = \langle y,x \rangle^*$

Linearity in the first argument: $\langle u+v,w \rangle = \langle u,w \rangle + \langle v,w \rangle$.

Positive-definiteness: $\langle v,v \rangle$ is nonnegative and $\langle v,v \rangle = 0$ implies $v=0$.

Inner Product Space

A vector space with an inner product is an inner product space.

Examples:

- The vector space R^n over R is an inner product space with the usual dot product.
- C^n over C is an inner product space equipped with the inner product satisfying the previous axioms. Observe that real vector spaces also satisfy those axioms, but conjugates are unnecessary because they're all real numbers.

Hilbert Space

A real or complex inner product space *that is complete* is a Hilbert Space. A Hilbert space can be finite or infinite. In quantum computation, we usually refer to finite Hilbert spaces. In quantum mechanics we often refer to *countably infinite* Hilbert spaces.

Completeness:

there is a distance function (or metric) that is induced by the inner product. We can use the inner product to take two vectors and produce a scalar. We can also use the inner product of a vector with itself which is the norm of that vector squared.

A space is **complete** if it is closed with respect to this metric iff the following two conditions are met:

1. Every sum of the norms of any sequence of vectors in the vector space converges to a number in the field. (note: there are vector spaces which meet this condition, but which do not also meet the next condition)
2. Every sum of any sequence of vectors in the vector space converges to a vector that is within the vector space

Note: converge = has a well-defined limit

Fredkin operator (CSWAP)

Controlled swap

Superdense Coding

[q] + [qq] >= [cc]

1 qubit + 1 ebit ≥ 2 cbits

Quantum Teleportation

[qq] + [cc] >= [q]

One ebit + two classical bits enable us to send the state of one qubit via QT.

Measurement at A informs Error Correction at B to recreate the state.

Controlled Z Gate

$|00\rangle \rightarrow |00\rangle$

$|01\rangle \rightarrow |01\rangle$

$|10\rangle \rightarrow |10\rangle$

$|11\rangle \rightarrow -|11\rangle$

Ising operator (C-Phase Shift)

If the control qubit is 0 do nothing

If the control qubit is 1, apply the phase shift operator to the target qubit.

Number Operator

n is the number operator. It operates on a state and gives the projection of that state along the (0 1)^T unit vector.

n|x> = x|x>

So that |x> is an eigenvector of n with x as the eigenvalue

~n is the complement of the n operator and is given by

~n = 1- n

Quantum Circuit Example: Quantum Teleportation

Square root of Swap operator

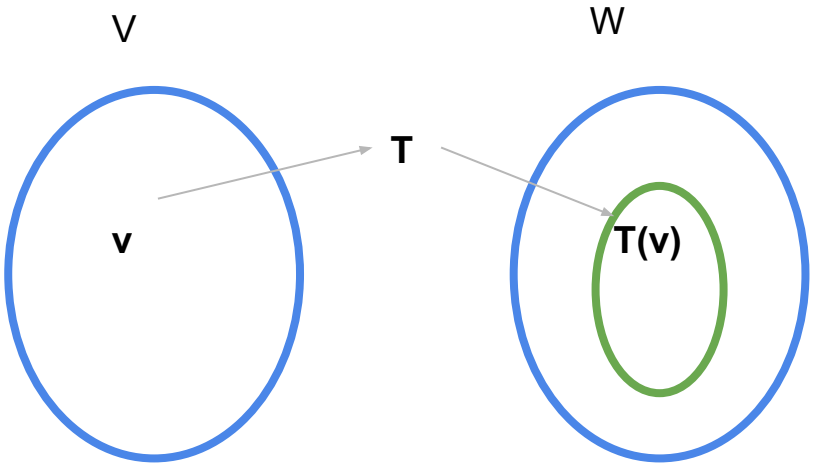
The sqrt(swap) gate performs half-way of a two-qubit swap. It is universal such that any quantum many qubit gate can be constructed from only sqrt(swap) and single qubit gates.

Additional Topics

Domain, Codomain, Image

The image of a map T is the set of vectors in the codomain of T which are the mappings of T applied to all vectors in the domain V.

- If Im(T) = W then T is surjective (T maps onto W)
- If all vectors in V map in a one-to-one manner T(v)s in W then T is injective (T maps into W)
- If T is both surjective and injective it is bijective



Injection

We say that a function $f : X \to Y$ is an injection iff $x \neq y \Rightarrow f(x) \neq f(y)$

Another formulation of injectivity,

exploiting the contrapositive, $f(x) = f(y) \Rightarrow x = y$

Informally, a function is injective if distinct elements of the domain do not map to the same element of the codomain.

An example of an injection is the function

$$\begin{aligned} f &: \mathbb{R} \to \mathbb{R} \\ x &\mapsto 2x \end{aligned}$$

A nonexample is

$$\begin{aligned} f &: \mathbb{R} \to \mathbb{R} \\ x &\mapsto x^2 \end{aligned}$$

Surjection

A function $f : X \to Y$ is **surjective** iff for each element of the codomain Y, there exists an element x in the domain X such that f(x)=y.

Informally, f is surjective iff each element of the codomain has some element of the domain that maps to it (nobody is left out). One of these is surjective and the other is not...

$$f : \mathbb{R} \to \mathbb{R}$$
$$x \mapsto x^2$$

$$f : \mathbb{R} \to \mathbb{R}$$
$$x \mapsto x^3$$

Bijection

A function that is an injection and also a surjection is a **bijection.**

That a bijection is injective and surjective ensures its invertibility, i.e. bijections have inverse functions. This will be important in the example that follows.

Think about all of the bijections of set with three elements to itself...

A non-abelian group

The symmetric group S_Ω on a set Ω is the set of all bijections from that set to itself equipped with the operation of function composition.

Another way to think of the symmetric group on a set is as the set of all permutations of that set. Observe that a permutation is a bijection, so these ways of thinking are the same.

Let's consider the symmetric group S_3 on a three-element set. For ease of notation, assume the three-element set is {1,2,3}. An example of an element of S_3 is the function specified by

$$f : \{1, 2, 3\} \to \{1, 2, 3\}$$
$$1 \mapsto 2$$
$$2 \mapsto 3$$
$$3 \mapsto 1$$

We can abbreviate f with the cycle notation (1 2 3). We read the parenthesized numbers from left to right to determine how the numbers are assigned. For example, 1 leads to 2, 2 leads to 3, and 3 "cycles" back around to 1, determining the function f. If a number is omitted (e.g. (1 3)). We can compose cycles as we compose functions. Check that (1 2 3)(1 2) = (1 3), i.e. transposing 1 and 2 and then cycling 1, 2, and 3.

But why is this a group?

This is a group because:

- composition of bijections yields a bijection (closure),
- composition of functions is associative,
- the identity function serves as the identity with the operation of composition,
- and each bijection (by definition of a bijection) has an inverse function under composition.

It is hopelessly non abelian because composition of functions is (very) non commutative.

Groups acting on Sets

A **group** G acts upon a set S if there is a function f:

$$G \times S \to S$$

such that

- $e \cdot s = s$ for all s in S, where e is the identity element of G
- $g \cdot (h \cdot s) = (g *_G h) \cdot s$for all g,h in G and s in S

e denotes the identity element of the group G

Division Rings and Fields

Notice that we never said that each nonzero element in a ring has a multiplicative inverse. Rings with this extra property are called **division rings**. The quaternions, denoted $\mathbb{H}$, is an example of a noncommutative division ring.

A commutative division ring is a **field**. Actually, we can make the integers into a field, by adjoining multiplicative inverses for nonzero elements - we know this field affectionately as $\mathbb{Q}$, the rational numbers ("Q" for the quotient field of $\mathbb{Z}$). The real numbers $\mathbb{R}$ and the complex numbers $\mathbb{C}$ are fields we know and love.

Norms with Matrices

- 1 Norm
- 2 Norm
- p Norm
- ∞ Norm
- The Euclidean norm

Test for Comprehension

Is the union of the axes in R^2 a subspace of R^2? Why or why not?

If you think not, provide an example of how the definition of subspace is violated. For instance, you might provide two vectors whose sum is no longer an element of the union of the axes, or a vector whose scalar multiple is no longer an element of the union of the axes.

Challenge: Verify that the set of all polynomials of maximum degree 3 is a subspace of the vector space of all polynomials.

Additional Slides

Alt. method for Computing the Determinant for a 3x3 Matrix

- Copy the matrix
- Draw diagonal lines starting from the top left down for every element in the original matrix
- Multiply the elements on every line together, then add the product of each line
- Draw diagonal lines starting from the top right down for every element in the original matrix
- Multiply the elements on every line together, then add the product of each line
- Subtract the second sum from the first sum

- For Matrix $\mathbf{A} = \begin{bmatrix} 4 & 3 & 3 \\ 5 & 2 & 7 \\ 3 & 6 & 9 \end{bmatrix}$

$$\begin{matrix} 4 & 3 & 3 & 4 & 3 & 3 \\ 5 & 2 & 7 & 5 & 2 & 7 \\ 3 & 6 & 9 & 3 & 6 & 9 \end{matrix}$$

(4*2*9) + (3*7*3) + (3*5*3) = 180

$$\begin{matrix} 4 & 3 & 3 & 4 & 3 & 3 \\ 5 & 2 & 7 & 5 & 2 & 7 \\ 3 & 6 & 9 & 3 & 6 & 9 \end{matrix}$$

(3*2*3) + (3*5*9) + (4*7*6) = 321

180-321=
-141

Integer, Rational, Real, and Complex Numbers

To investigate linear algebra properly, we'll need to brush up on some sets of numbers that may or may not be familiar.

Most everything begins with the set of integers, or whole numbers. We denote this set by $\mathbb{Z}$. The "Z" is derived from the German "zahlen", meaning "number". Examples include numbers like ...,-2,-1,0,1,2,...

We can get along pretty well with the integers, at least until we want to divide. This calls for the rational numbers (the fraction field of the integers),

denoted $\mathbb{Q}$, for "quotient". It consists of all ratios of integers (hence, **ratio**nal). Examples include ½, ¾, -⅚, etc.

More Numbers

But is this really enough? Wouldn't it be nice to have numbers like $\sqrt{2}$? Then, we'll have to expand our collection of numbers to all of the real numbers, denoted $\mathbb{R}$. With this, we pick up numbers like π, e, $\sqrt{\sqrt{2}+2}$, etc.

... and we're still not finished! Even with all of the real numbers at our disposable, we can't solve this silly little equation $x^2 + 1 = 0$, so we toss in the solution to that equation, call it "i", and now have all of the complex numbers $\mathbb{C}$. Examples include i, of course, 1+i, π*i, $\sqrt{2}$+ $\sqrt{2}$i.

What is AXPY?

- AXPY = Scalar **a**lpha times **x** **p**lus **y**
- A function that is used many times throughout linear algebra to save on memory operations, by adding the scalar to the registry

This is an affine transformation

Kernel is the Null space

The kernel is the subspace of all vectors in the domain of a linear map that are mapped to a zero vector in the co-domain (the co-domain can be the same subspace as the domain or a different one).

Expectation Value

To compute an observable, we apply an operator to a state. For example,

The Stress Tensor

ML Example: Converting Words to Vectors

- The sentence in question:
  - “The cat sat on the mat”

- To the right, we have converted the six word sentence into a 6x5 matrix - each vector corresponds to a unique word. This method of encoding is called “one-hot” encoding.

- As we can see: For a corpus of 100,000 words with 15,000 unique words it would require a matrix of dimensions 100,000 x 15,000.

the = [1 0 0 0 0]
cat = [0 1 0 0 0]
sat = [0 0 1 0 0]
on = [0 0 0 1 0]
the = [1 0 0 0 0]
mat = [0 0 0 0 1]

What do we do with that Matrix?

- There are variety of techniques such as one-hot encoding, word2vec, GloVe, and various neural network techniques that create vectors from a corpus.

- Using the one-hot encoding is a simple to check the similarity between two corpora.

- Check out this example of a method for character to vector similarity checking.

Try it Yourself: Compare Word Similarities

- In the same way words that compose a document can be one-hot encoded into vectors, letters that compose words can one-hot encoded as well.

- This allows us to graph the vectors created in this manner and find the distance between them. The closer they are, the more similar they are!

- Try it yourself

Word2vec: Optimizations

- One-hot encoding alone is inefficient. For a corpus of 100,000 words with 15,000 unique words it would require a matrix of dimensions 100,000 x 15,000.

- This method also strips away the context of words (whether or not they are often placed near other particular words).

- There are many solutions, a popular one is word2vec which uses either the Skip-gram approach **OR** the Continuous Bag of Words approach.