Introduction to Linear Algebra

Mark Goldman
Emily Mackevicius

Outline

1. Matrix arithmetic
2. Matrix properties
3. Eigenvectors & eigenvalues

-BREAK-

4. Examples (on blackboard)
5. Recap, additional matrix properties, SVD

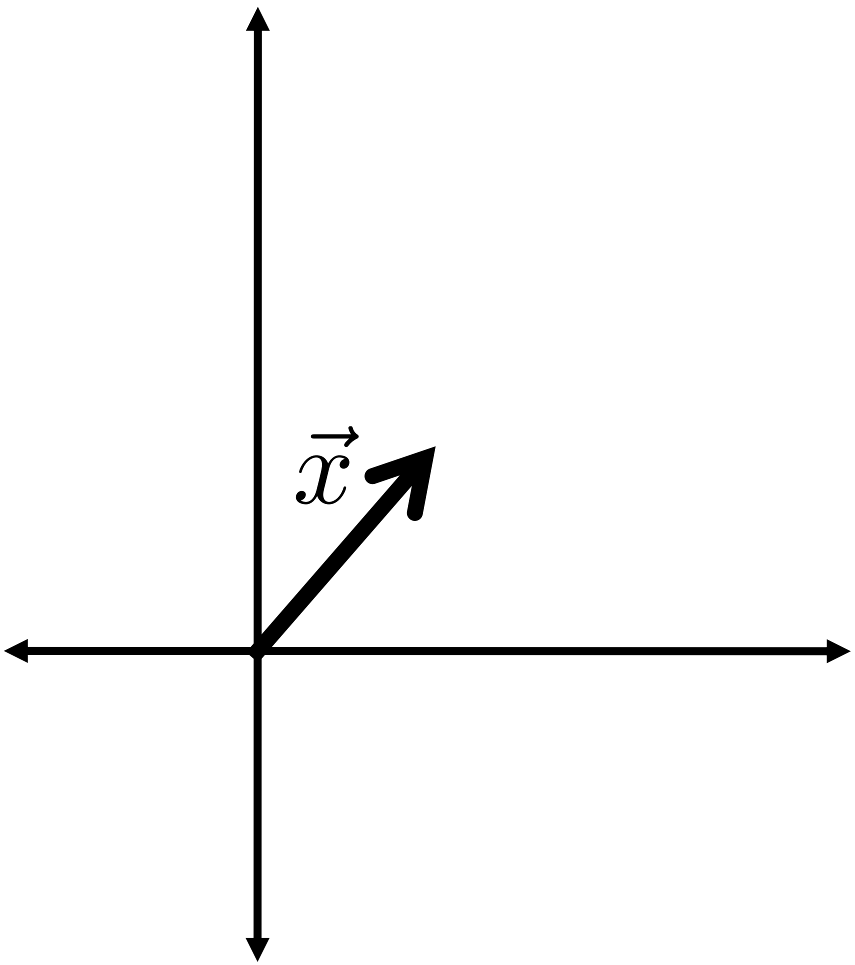
Part 1: Matrix Arithmetic

(w/applications to neural networks)

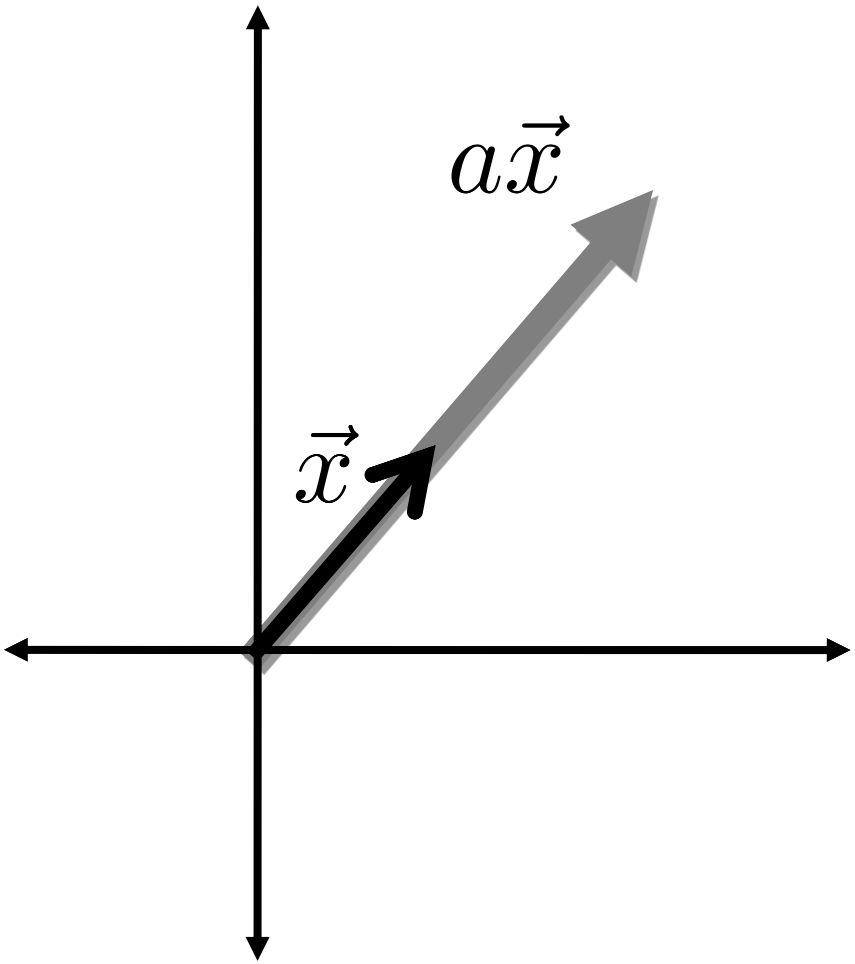
Matrix addition

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} + \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 6 & 8 \\ 10 & 12 \end{pmatrix}$$

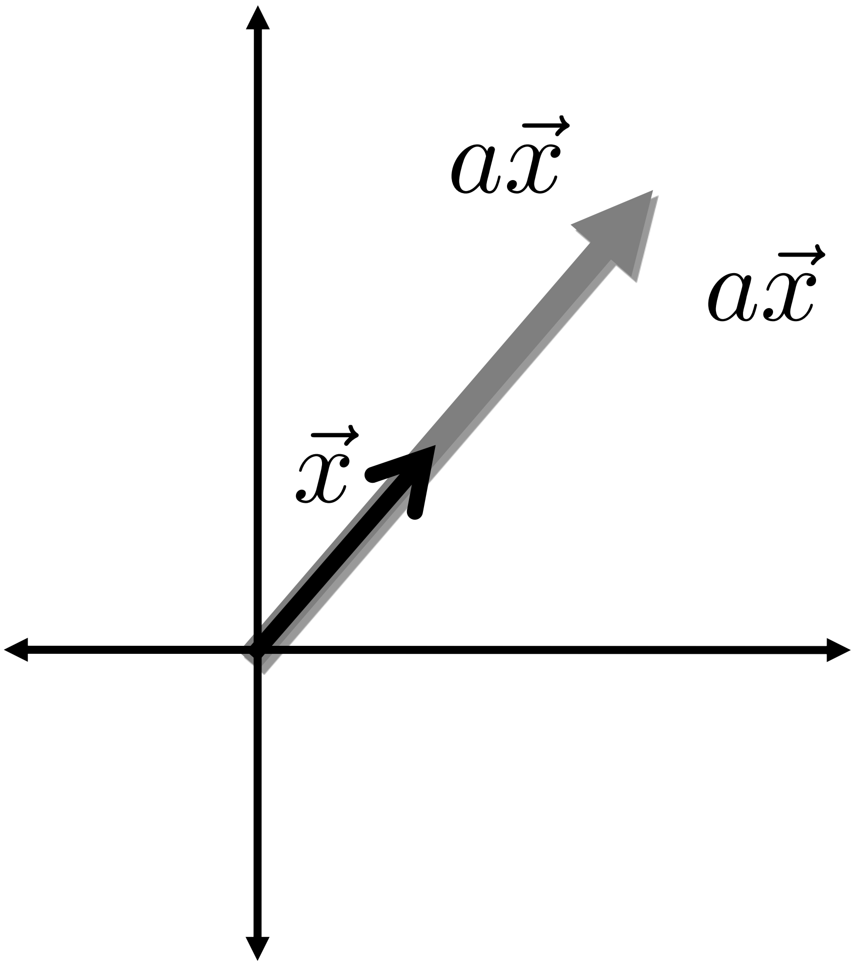
Scalar times vector



Scalar times vector



Scalar times vector



$$a\vec{x} = a \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} ax_1 \\ ax_2 \\ \vdots \\ ax_N \end{pmatrix}$$

Product of 2 Vectors

Three ways to multiply

- Element-by-element
- Inner product
- Outer product

Element-by-element product (Hadamard product)

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} .* \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_2 b_2 \end{pmatrix}$$

- Element-wise multiplication (.* in MATLAB)

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

Multiplication: **Dot product (inner product)**

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

$$= \sum_{i=1}^{N} x_i y_i$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

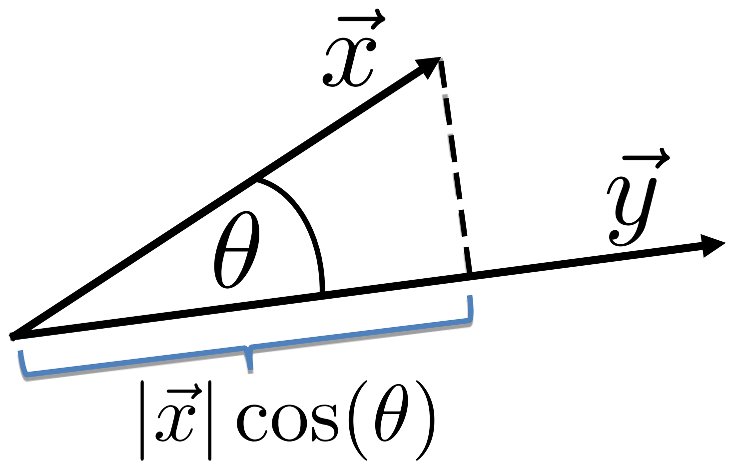
$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

1 X N N X 1 1 X 1

- MATLAB: ‘inner matrix dimensions must agree’

Outer dimensions give size of resulting matrix

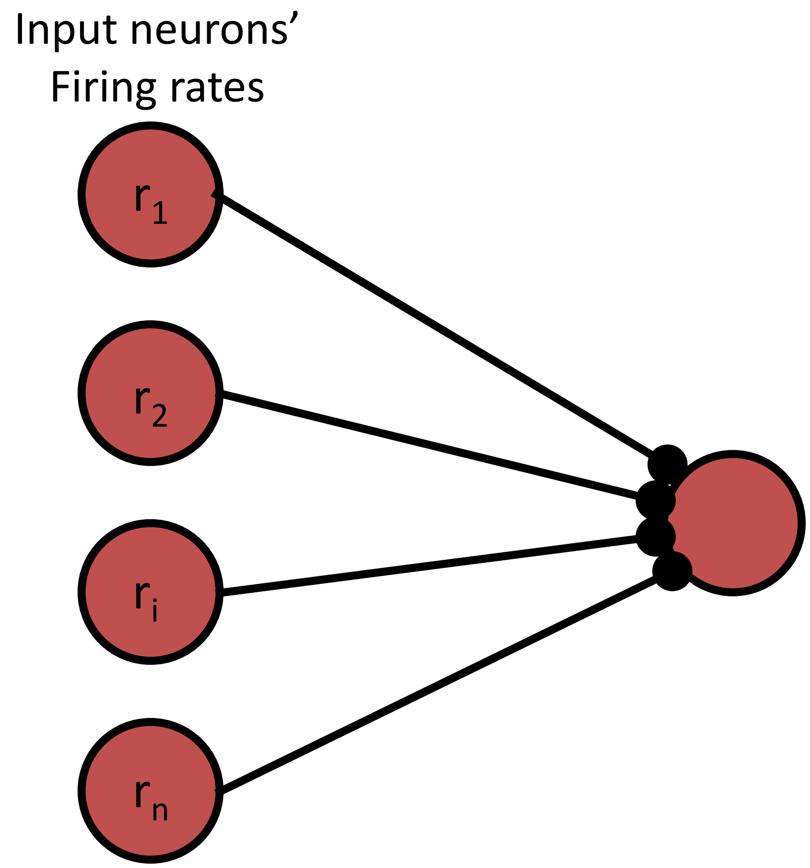
Dot product geometric intuition: "Overlap" of 2 vectors



$$\vec{x} \cdot \vec{y} = |\vec{x}||\vec{y}|\cos(\theta)$$

Example: linear feed-forward network

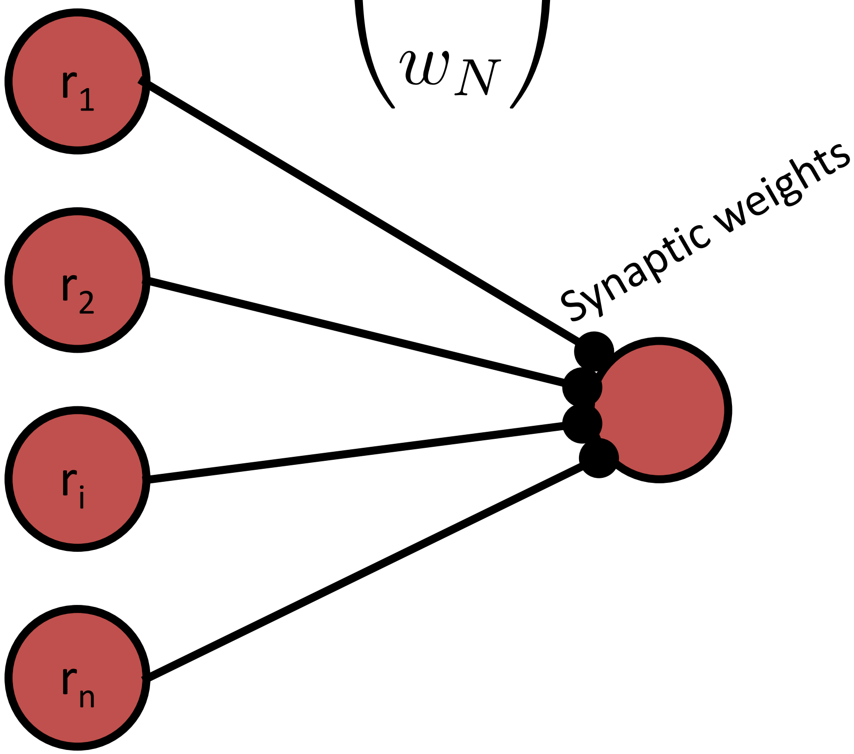
$$\begin{pmatrix} r_1 & r_2 & \cdots & r_N \end{pmatrix}$$



Example: linear feed-forward network

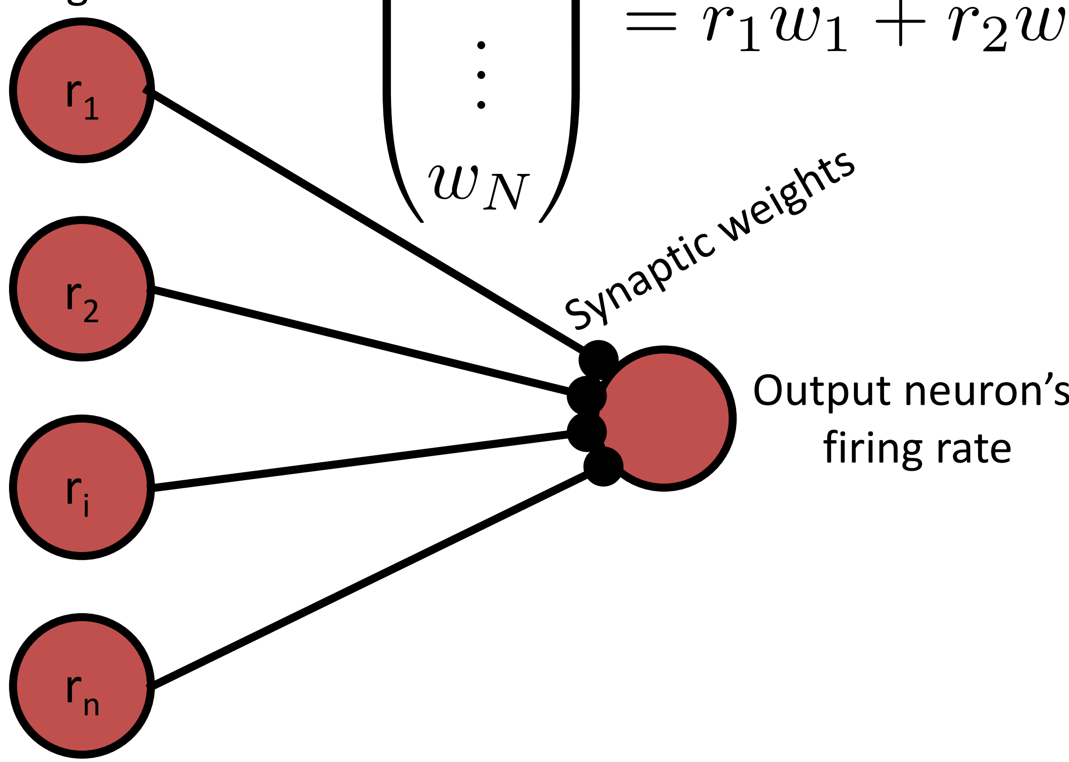
$$\begin{pmatrix} r_1 & r_2 & \cdots & r_N \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix}$$

Input neurons'
Firing rates

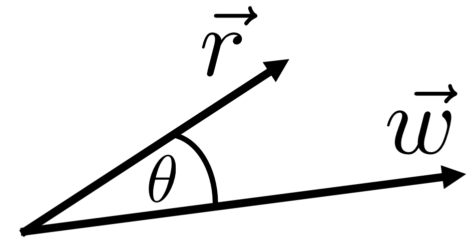


Example: linear feed-forward network

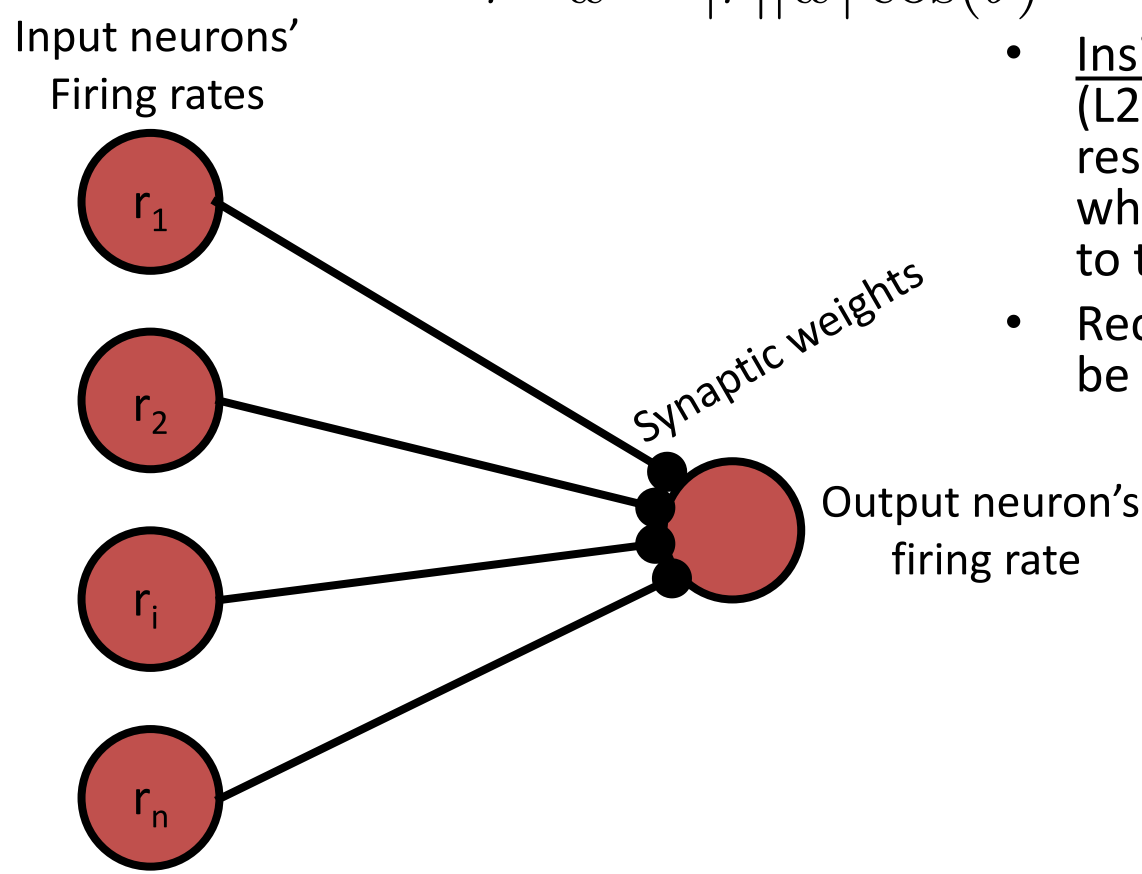
$$\begin{pmatrix} r_1 & r_2 & \cdots & r_N \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix} = r_1 w_1 + r_2 w_2 + \cdots + r_N w_N$$



Example: linear feed-forward network



$$\vec{r} \cdot \vec{w} = |\vec{r}||\vec{w}| \cos(\theta)$$



- <u>Insight</u>: for a given input (L2) magnitude, the response is maximized when the input is parallel to the weight vector
- Receptive fields also can be thought of this way

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \ldots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

N X 1   1 X M   N X M

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \ldots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \ldots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \ldots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \cdots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \ldots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \vdots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \vdots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

- Note: each column or each row is a multiple of the others

Matrix times a vector

$$\overrightarrow{y} = \overleftrightarrow{W}\,\overrightarrow{x}$$

Matrix times a vector

$$\overrightarrow{y} = \overleftrightarrow{W}\,\overrightarrow{x}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

Matrix times a vector

$$\overrightarrow{y} = \overleftrightarrow{W}\,\overrightarrow{x}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

M X 1 M X N N X 1

Matrix times a vector: **inner product interpretation**

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- Rule: the i^{th} element of **y** is the dot product of the i^{th} row of W with **x**

Matrix times a vector: **inner product interpretation**

$$
\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}
$$

- Rule: the ith element of **y** is the dot product of the ith row of W with **x**

Matrix times a vector: **inner product interpretation**

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- Rule: the ith element of **y** is the dot product of the ith row of W with **x**

Matrix times a vector: **inner product interpretation**

$$
\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}
$$

- Rule: the i^{th} element of **y** is the dot product of the i^{th} row of W with **x**

Matrix times a vector: **inner product interpretation**

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- Rule: the ith element of **y** is the dot product of the ith row of W with **x**

Matrix times a vector: **outer product interpretation**

$$
\begin{array}{c}
\begin{array}{cccc} \vec{W}^{(1)} & \phantom{W_{M2}} & & \phantom{W_{MN}} \\ \downarrow & & & \end{array} \\
\begin{pmatrix}
W_{11} & W_{12} & \cdots & W_{1N} \\
W_{21} & W_{22} & \cdots & W_{2N} \\
\vdots & \vdots & & \vdots \\
W_{M1} & W_{M2} & \cdots & W_{MN}
\end{pmatrix}
\end{array}
\begin{pmatrix}
x_1 \\ x_2 \\ \vdots \\ x_N
\end{pmatrix} =
$$

- The product is a weighted sum of the columns of W, weighted by the entries of x

Matrix times a vector:
outer product interpretation

$$\begin{array}{l} \begin{array}{cccc} \overrightarrow{W}^{(1)} & \phantom{W_{12}} & & \phantom{W_{1N}} \end{array} \\ \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \end{array} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = x_1 \overrightarrow{W}^{(1)} +$$

- The product is a weighted sum of the columns of W, weighted by the entries of x

Matrix times a vector: **outer product interpretation**

$$\begin{array}{c} \overrightarrow{W}^{(1)} \\ \downarrow \end{array}$$

$$\begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = x_1 \overrightarrow{W}^{(1)} + x_2 \overrightarrow{W}^{(2)}$$

- The product is a weighted sum of the columns of W, weighted by the entries of x

Matrix times a vector: **outer product interpretation**

$$
\begin{array}{c}
\overrightarrow{W}^{(1)} \\
\downarrow
\end{array}
$$

$$
\begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = x_1 \overrightarrow{W}^{(1)} + x_2 \overrightarrow{W}^{(2)} + \cdots + x_N \overrightarrow{W}^{(N)}
$$

- The product is a weighted sum of the columns of W, weighted by the entries of x

Example of the outer product method

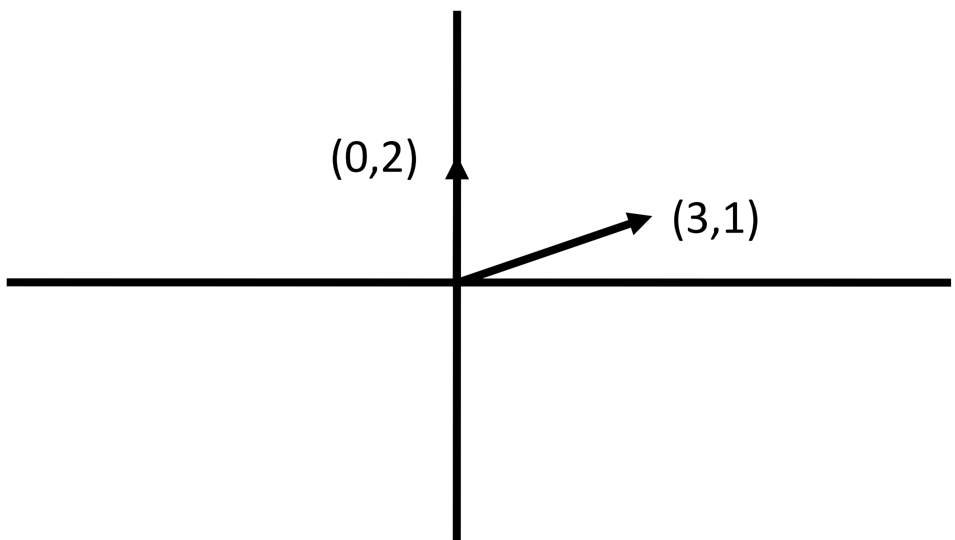
$\overleftrightarrow{M}$

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

Example of the outer product method

$\overleftrightarrow{M}$

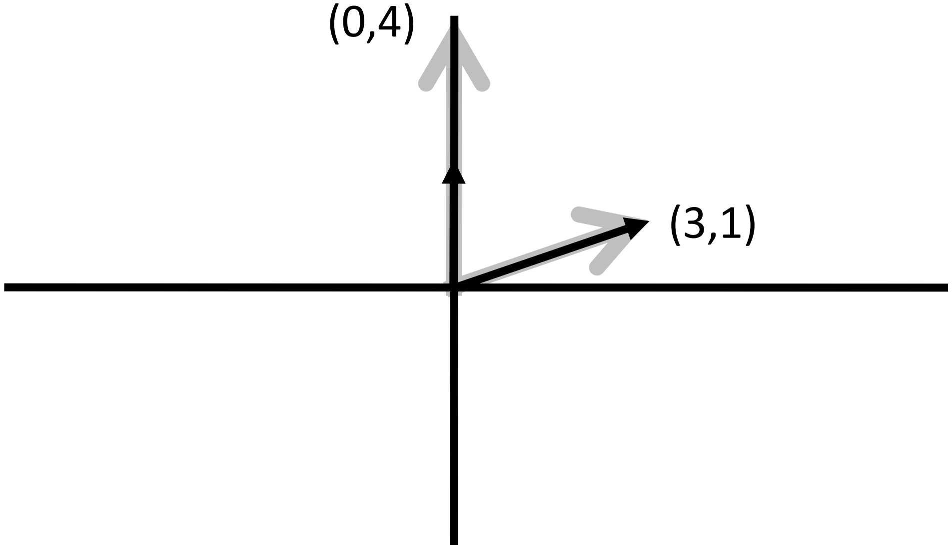
$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 0 \\ 2 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$



Example of the outer product method

$$\overleftrightarrow{M}$$

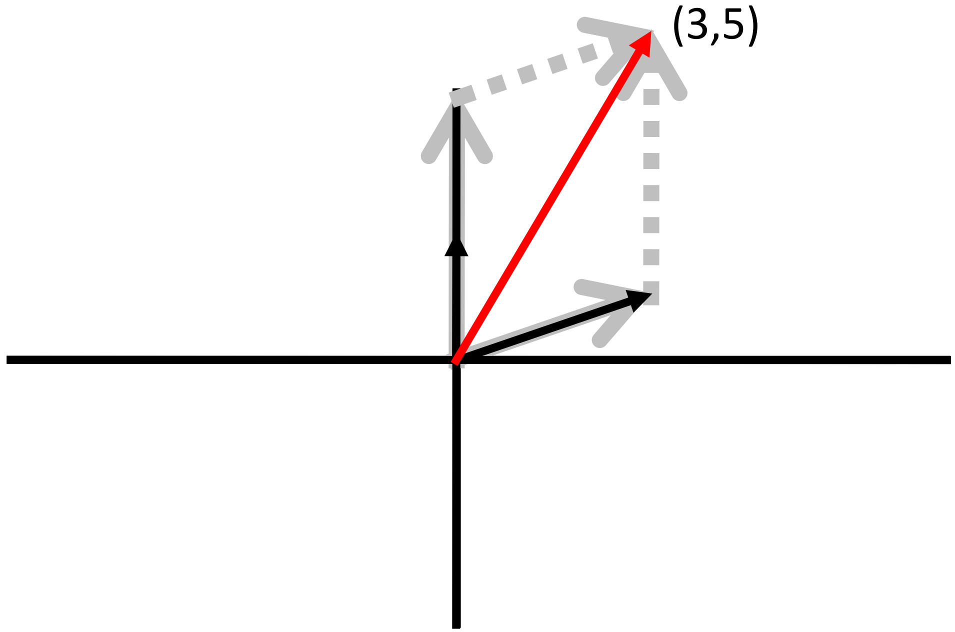
$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 0 \\ 2 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$



Example of the outer product method

$$\overleftrightarrow{M}$$

$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 0 \\ 2 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$



- Note: different combinations of the columns of **M** can give you any vector in the plane

(we say the columns of **M** "span" the plane)

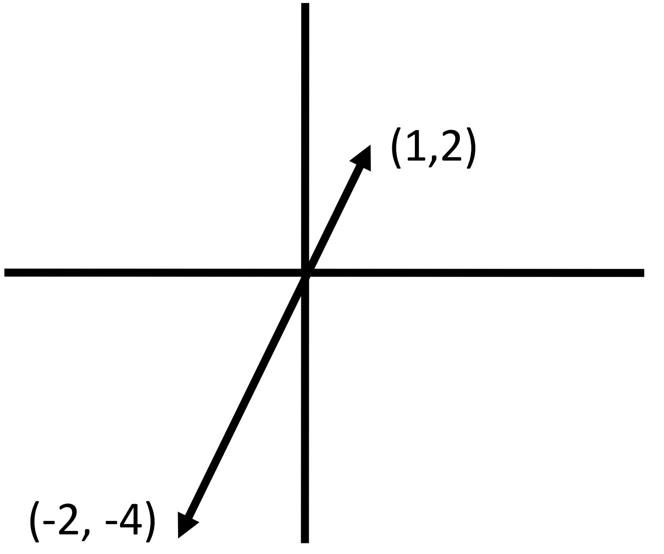
Rank of a Matrix

- Are there special matrices whose columns don't span the full plane?

Rank of a Matrix

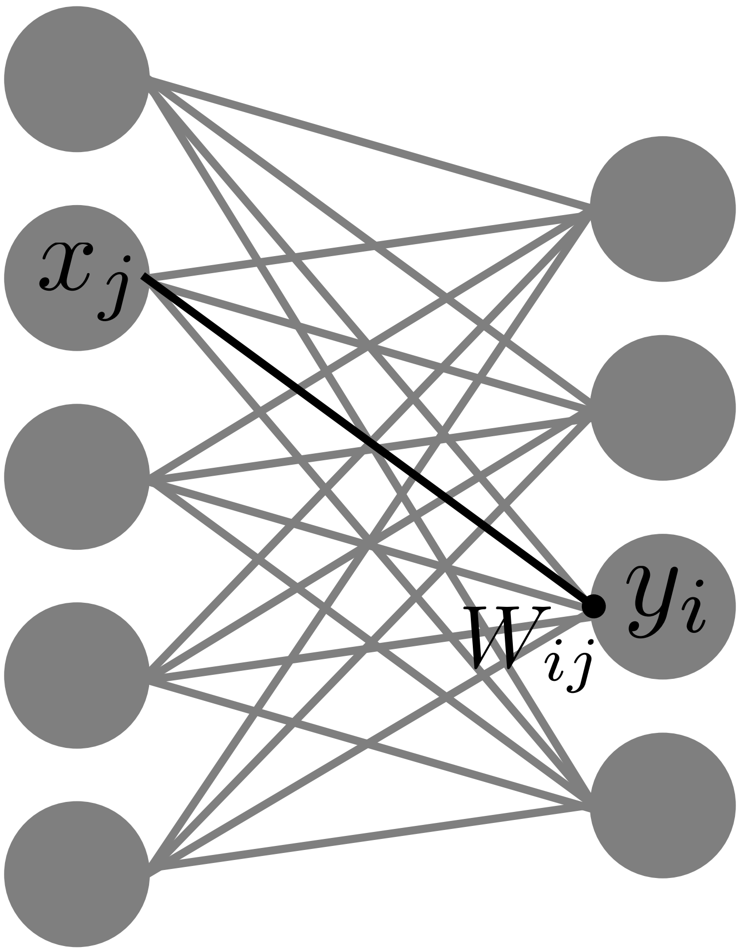
- Are there special matrices whose columns don't span the full plane?

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



- You can only get vectors along the (1,2) direction (i.e. outputs live in 1 dimension, so we call the matrix *rank 1*)

Example: 2-layer linear network

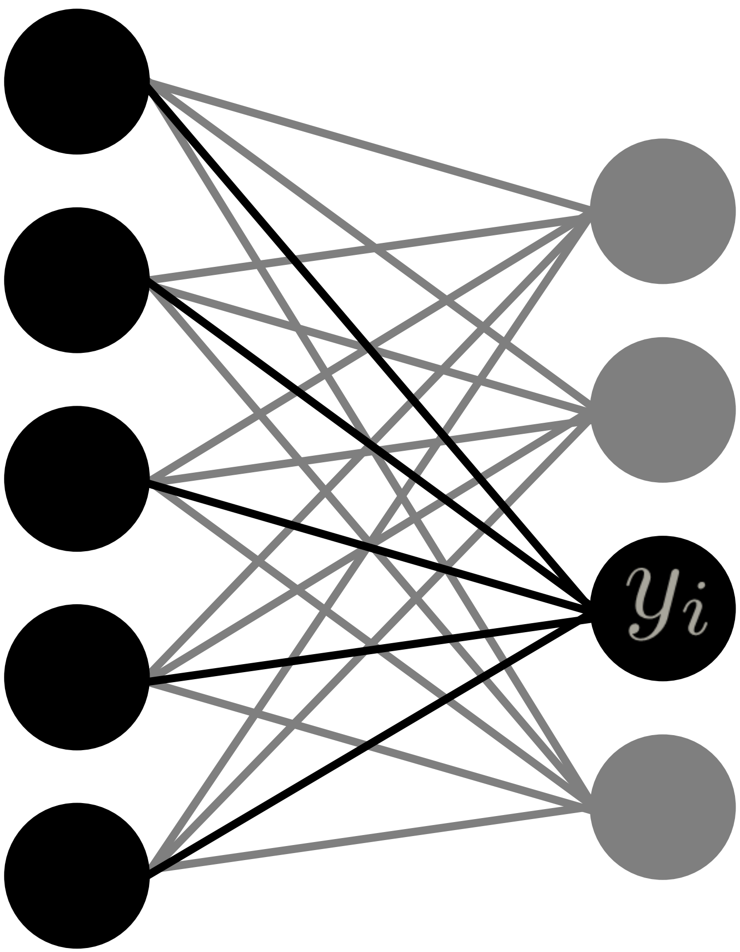


$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- W_{ij} is the connection strength (weight) onto neuron y_i from neuron x_j.

Example: 2-layer linear network: **inner product point of view**

- *What is the response of cell* y_i *of the second layer?*

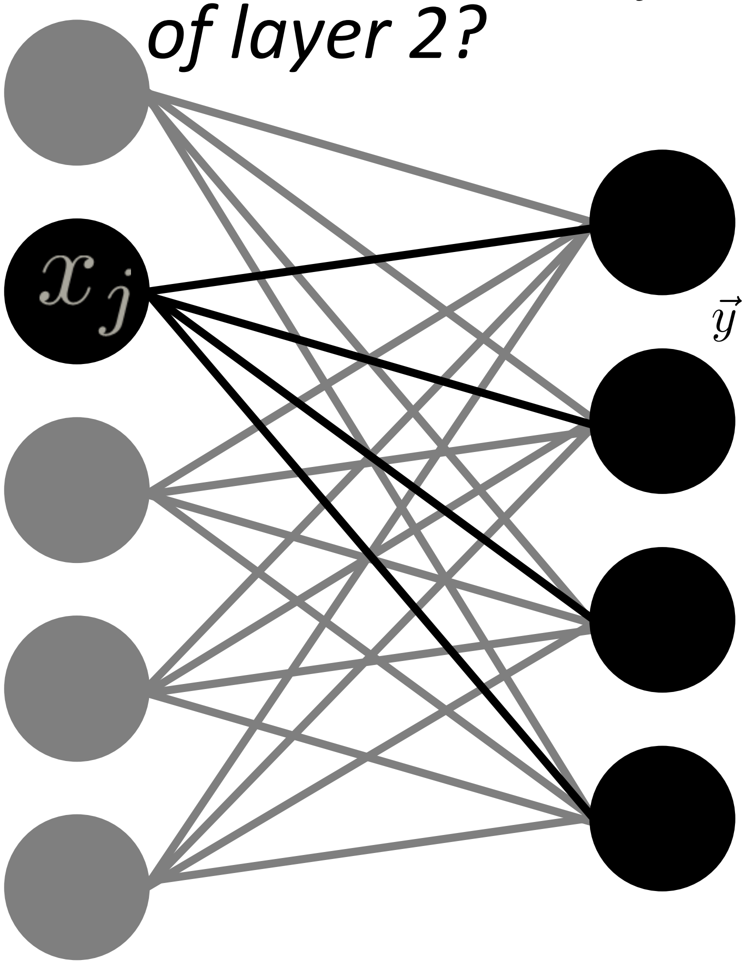


$$y_i = \sum_{j=1}^{N} W_{ij} x_j$$

- *The response is the dot product of the* i^{th} *row of W with the vector x*

Example: 2-layer linear network: **outer product point of view**

- *How does cell x_j contribute to the pattern of firing of layer 2?*



$$\vec{y} = x_1 \overbrace{\overrightarrow{W}^{(1)}}^{\text{1st column of W}} + x_2 \overrightarrow{W}^{(2)} + \cdots + \underbrace{x_j \overrightarrow{W}^{(j)}}_{\text{Contribution of } x_j \textit{ to network output}} + \cdots + x_N \overrightarrow{W}^{(N)}$$

Product of 2 Matrices

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

N X P   P X M   N X M

- MATLAB: 'inner matrix dimensions must agree'
- **Note:** Matrix multiplication doesn't (generally) commute, **AB** ≠ **BA**

Matrix times Matrix: **by inner products**

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{iP} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1j} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2j} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{Pj} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & C_{ij} & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

- C_{ij} is the inner product of the i^{th} row with the j^{th} column

Matrix times Matrix:
by inner products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{iP} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1j} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2j} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{Pj} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & C_{ij} & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

- C_{ij} is the inner product of the i^{th} row with the j^{th} column

Matrix times Matrix:
by inner products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{iP} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1j} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2j} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{Pj} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & C_{ij} & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

- C_{ij} is the inner product of the i^{th} row with the j^{th} column

Matrix times Matrix: **by inner products**

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{iP} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1j} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2j} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{Pj} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & C_{ij} & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$C_{ij} = \sum_{k=1}^{P} A_{ik} B_{kj}$$

- C_{ij} is the inner product of the i^{th} row of **A** with the j^{th} column of **B**

Matrix times Matrix:
by outer products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$\overleftrightarrow{C} =$$

Matrix times Matrix:
by outer products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$\overleftrightarrow{C} = \begin{pmatrix} \\ A^{\mathrm{c}1} \\ \\ \end{pmatrix} \begin{pmatrix} \quad B^{\mathrm{r}1} \quad \end{pmatrix} +$$

Matrix times Matrix: **by outer products**

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$\overleftrightarrow{C} = \begin{pmatrix} \\ A^{\mathrm{c}1} \\ \\ \end{pmatrix} \begin{pmatrix} \quad B^{\mathrm{r}1} \quad \end{pmatrix} + \begin{pmatrix} \\ A^{\mathrm{c}2} \\ \\ \end{pmatrix} \begin{pmatrix} \quad B^{\mathrm{r}2} \quad \end{pmatrix} +$$

Matrix times Matrix:
by outer products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$\overleftrightarrow{C} = \begin{pmatrix} \\ A^{\mathrm{c}1} \\ \\ \end{pmatrix} \begin{pmatrix} & B^{\mathrm{r}1} & \end{pmatrix} + \begin{pmatrix} \\ A^{\mathrm{c}2} \\ \\ \end{pmatrix} \begin{pmatrix} & B^{\mathrm{r}2} & \end{pmatrix} + \cdots + \begin{pmatrix} \\ A^{\mathrm{cP}} \\ \\ \end{pmatrix} \begin{pmatrix} & B^{\mathrm{rP}} & \end{pmatrix}$$

- **C** is a sum of outer products of the columns of **A** with the rows of **B**

Part 2: Matrix Properties

- (A few) special matrices
- Matrix transformations & the determinant
- Matrices & systems of algebraic equations

Special matrices: **diagonal matrix**

$$\overleftrightarrow{D} = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{pmatrix}$$

$$\overleftrightarrow{D}\,\overrightarrow{x} = \begin{pmatrix} d_1 x_1 \\ d_2 x_2 \\ \vdots \\ d_n x_n \end{pmatrix}$$

- This acts like scalar multiplication

Special matrices: **identity matrix**

$$\overleftrightarrow{1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

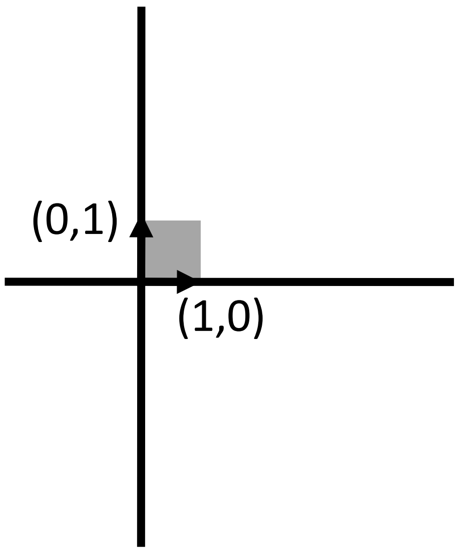
for all $\overleftrightarrow{A}$, $\overleftrightarrow{1}\overleftrightarrow{A} = \overleftrightarrow{A}\overleftrightarrow{1} = \overleftrightarrow{A}$

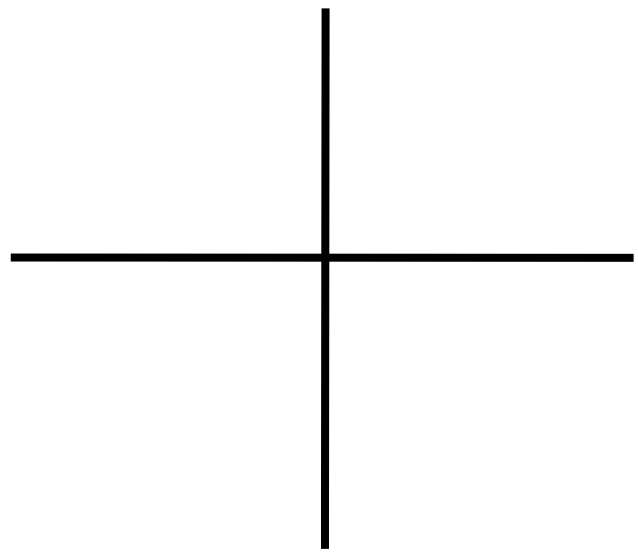
Special matrices: **inverse matrix**

$$\overleftrightarrow{A}\,\overleftrightarrow{A}^{-1} = \overleftrightarrow{A}^{-1}\overleftrightarrow{A} = \overleftrightarrow{1}$$

- Does the inverse always exist?

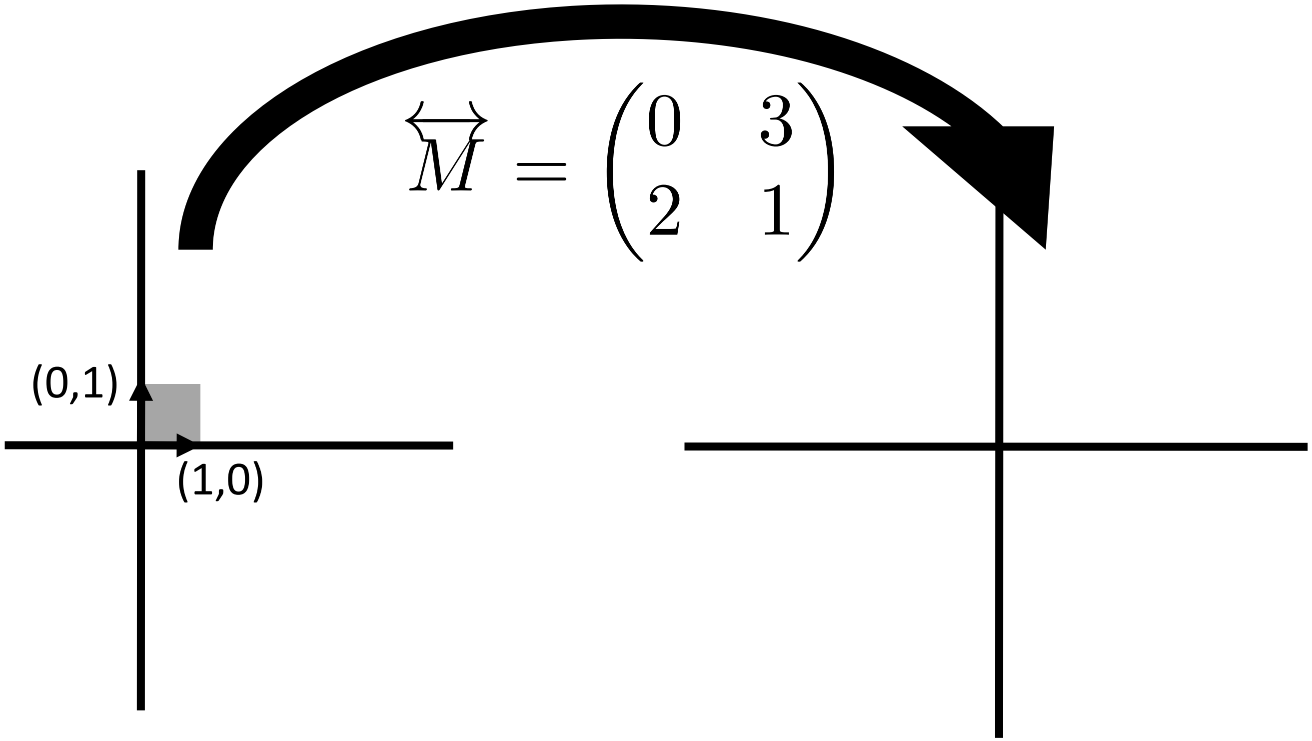
How does a matrix transform a square?





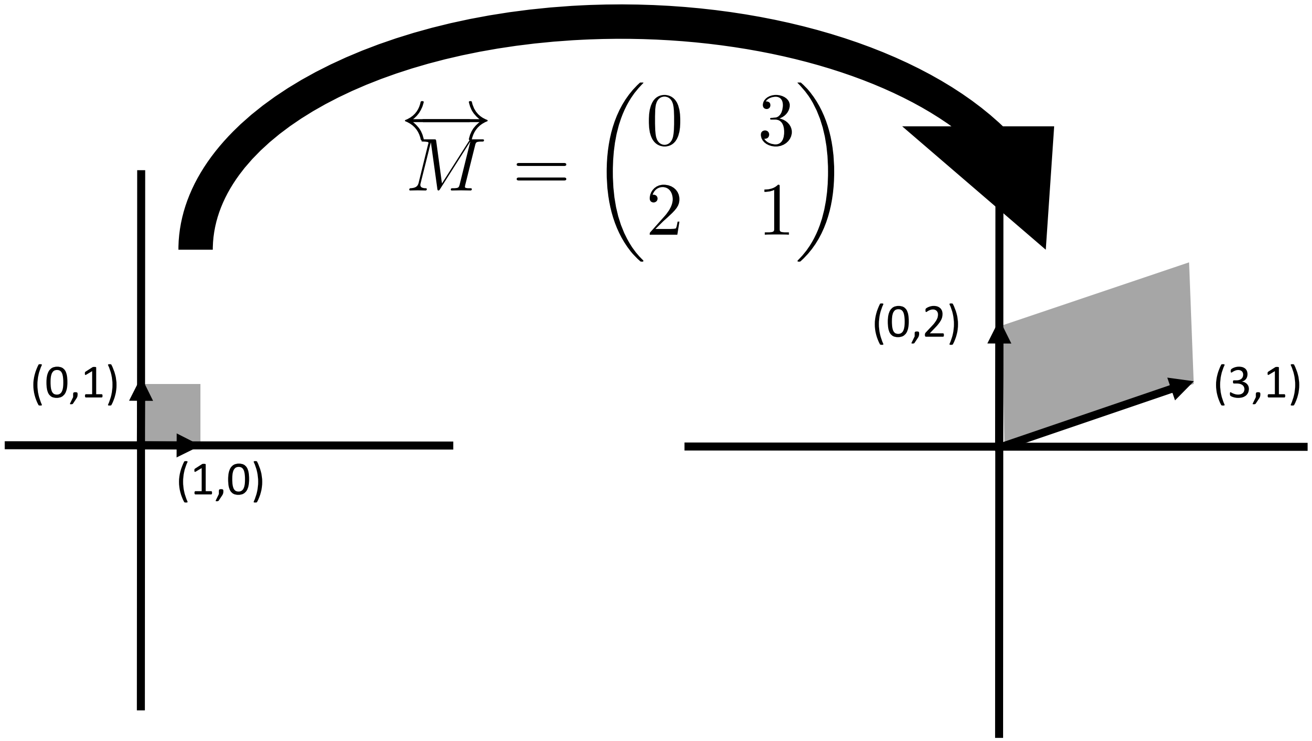
How does a matrix transform a square?

$$\overleftrightarrow{M} = \begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}$$

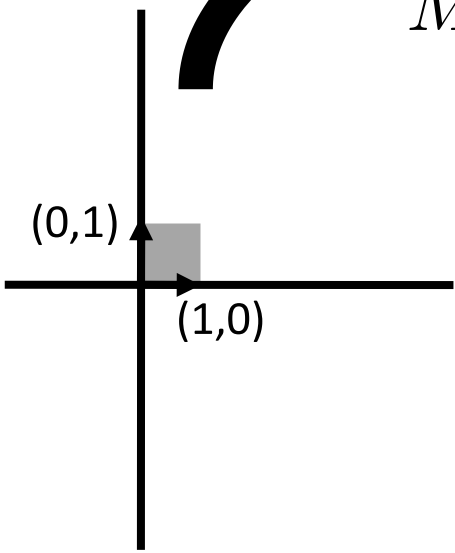


How does a matrix transform a square?

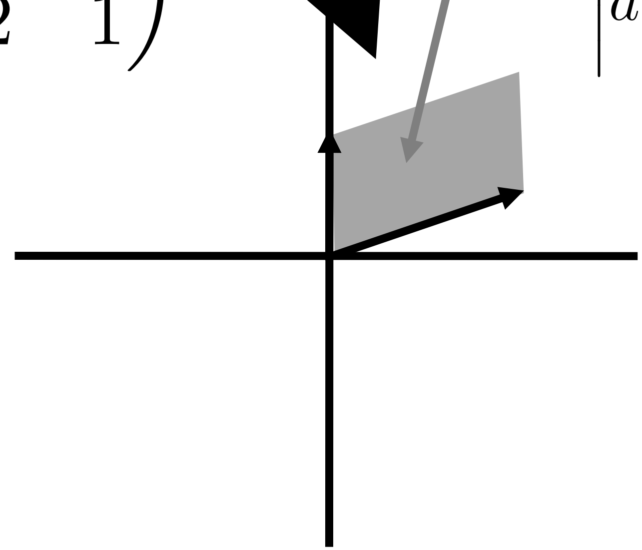
$$\overleftrightarrow{M} = \begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}$$



Geometric definition of the determinant: How does a matrix transform a square?



$$\overleftrightarrow{M} = \begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}$$



$$Area = \left| det(\overleftrightarrow{M}) \right|$$

$$\left| det \begin{pmatrix} a & b \\ c & d \end{pmatrix} \right| = |ad - bc|$$

Example: solve the algebraic equation

$$\overleftrightarrow{A}\,\vec{x} = \lambda \vec{x}$$

Example: solve the algebraic equation

$$\overleftrightarrow{A}\,\vec{x} = \lambda \vec{x}$$

$$\left(\overleftrightarrow{A} - \lambda \overleftrightarrow{1} \right) \vec{x} = 0$$

Example: solve the algebraic equation

$$\overleftrightarrow{A}\vec{x} = \lambda\vec{x}$$

$$\left(\overleftrightarrow{A} - \lambda\overleftrightarrow{1}\right)\vec{x} = 0$$

$$\Rightarrow\ det\left(\overleftrightarrow{A} - \lambda\overleftrightarrow{1}\right) = 0 \text{ or } \vec{x} = 0$$

Example of an underdetermined system

$$\begin{pmatrix} 1 & -1 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Example of an underdetermined system

$$\begin{pmatrix} 1 & -1 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$x_1 - x_2 = 0, \; 2x_1 - 2x_2 = 0$$

Example of an underdetermined system

$$\begin{pmatrix} 1 & -1 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$x_1 - x_2 = 0,\ 2x_1 - 2x_2 = 0$$

$$\Rightarrow x_1 = x_2$$

Example of an underdetermined system

$$\begin{pmatrix} 1 & -1 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$x_1 - x_2 = 0, \;\; 2x_1 - 2x_2 = 0$$

$$\Rightarrow x_1 = x_2$$

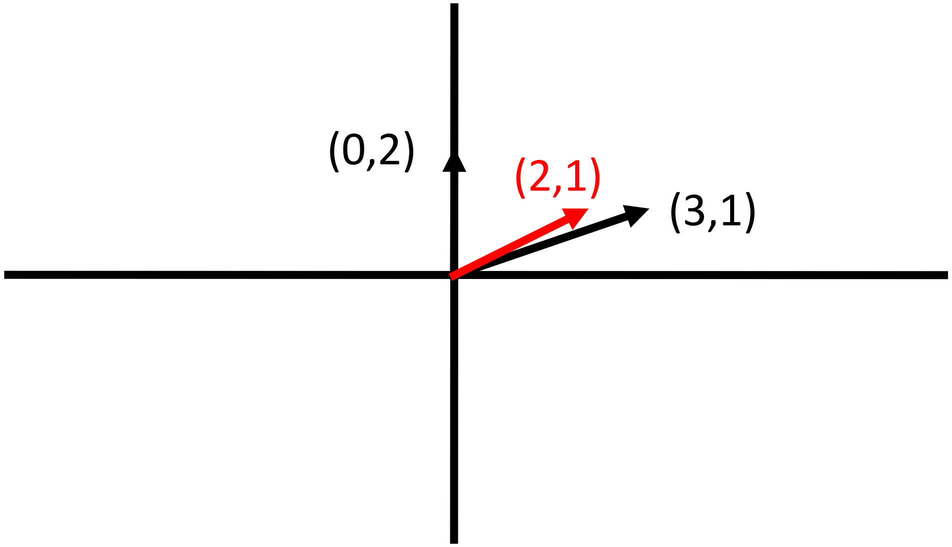
- Some non-zero **x** are sent to 0 (the set of all x with Mx=0 are called the “nullspace” of M)
- This is because det(**M**)=0 so **M** is not invertible. (If det(**M**) isn’t 0, the only solution is **x** = 0)

Part 3: Eigenvectors & eigenvalues

What do matrices do to vectors?

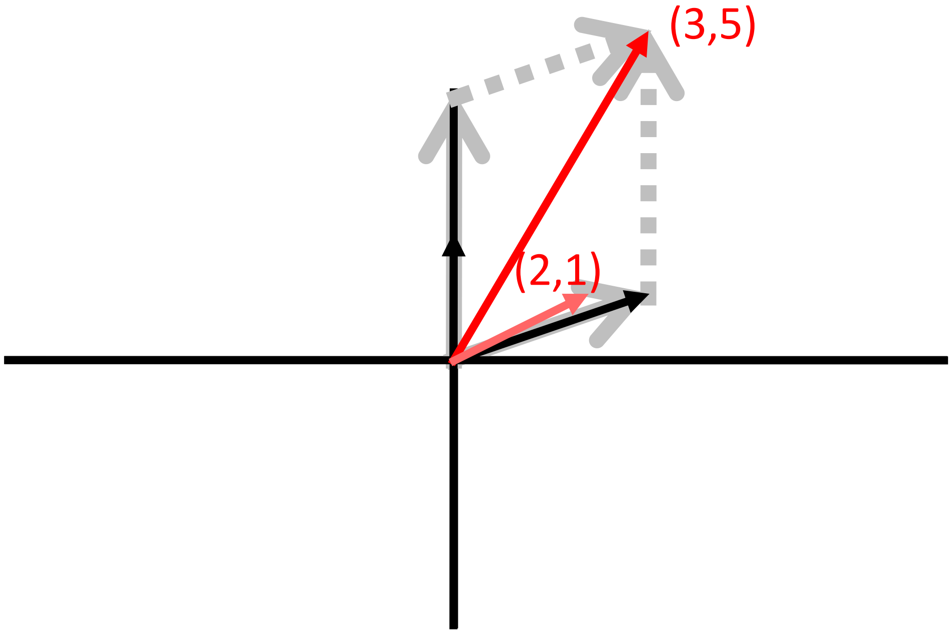
$$\overleftrightarrow{M}$$

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



Recall

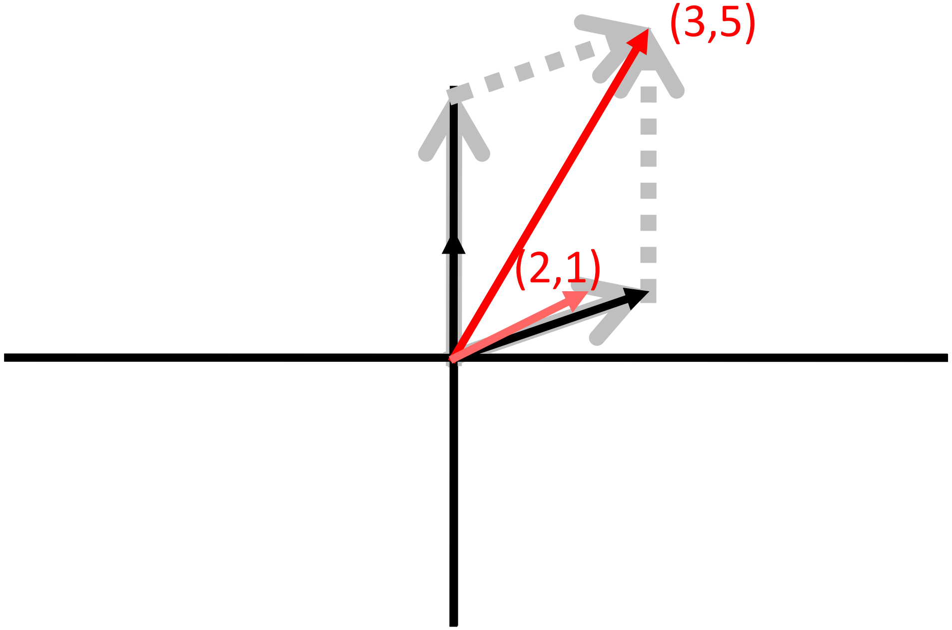
$$\overbrace{\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}}^{M} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 0 \\ 2 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$



What do matrices do to vectors?

$$\overleftrightarrow{M}$$

$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 0 \\ 2 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$



- The new vector is:
  1) **rotated**
  2) **scaled**

Are there any special vectors that **only get scaled**?

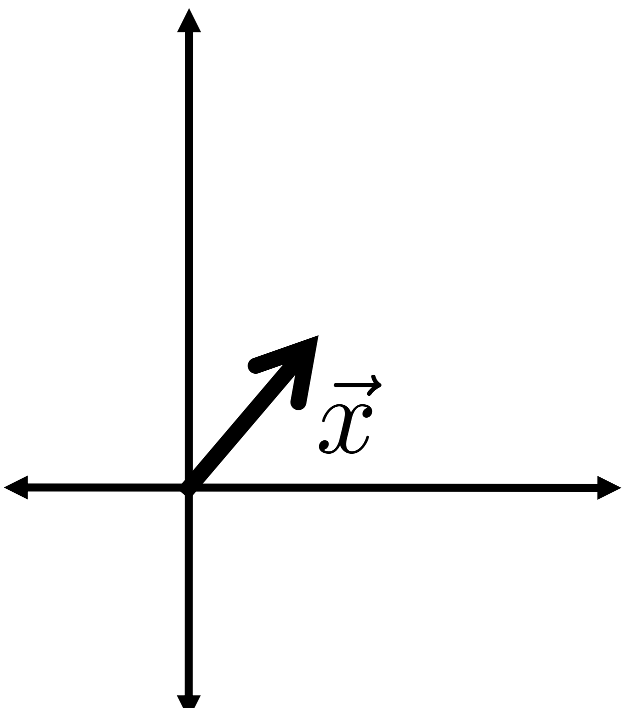
$$\overleftrightarrow{M}$$

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

Are there any special vectors that **only get scaled**?

$$\overleftrightarrow{M}$$

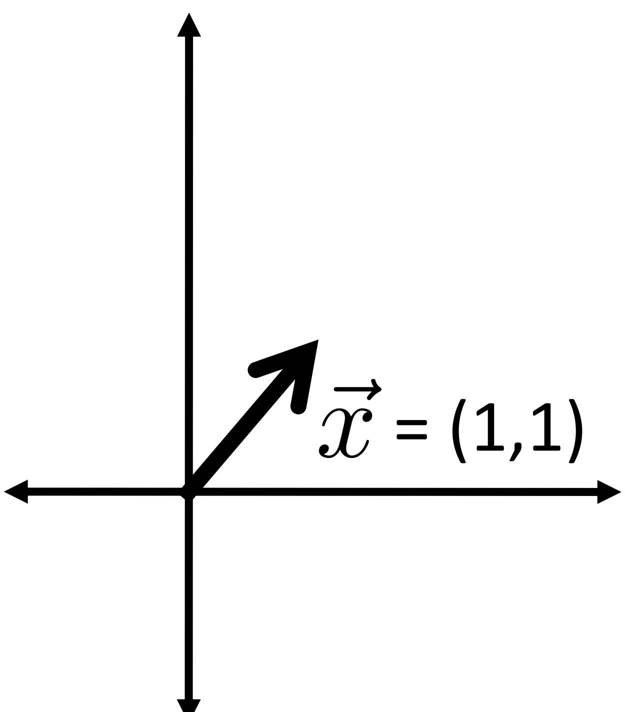
$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



Try (1,1)

Are there any special vectors that **only get scaled**?

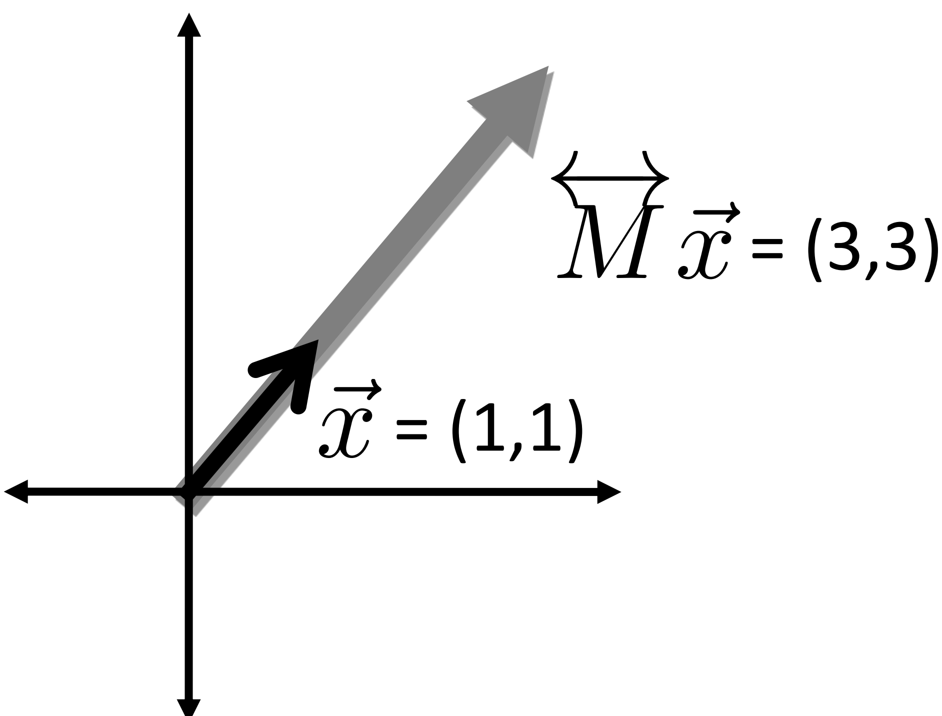
$$\overbrace{\begin{pmatrix}0 & 3\\ 2 & 1\end{pmatrix}}^{M}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}0\\2\end{pmatrix} + \begin{pmatrix}3\\1\end{pmatrix} = \begin{pmatrix}3\\3\end{pmatrix}$$



Are there any special vectors that **only get scaled**?

$\overleftrightarrow{M}$

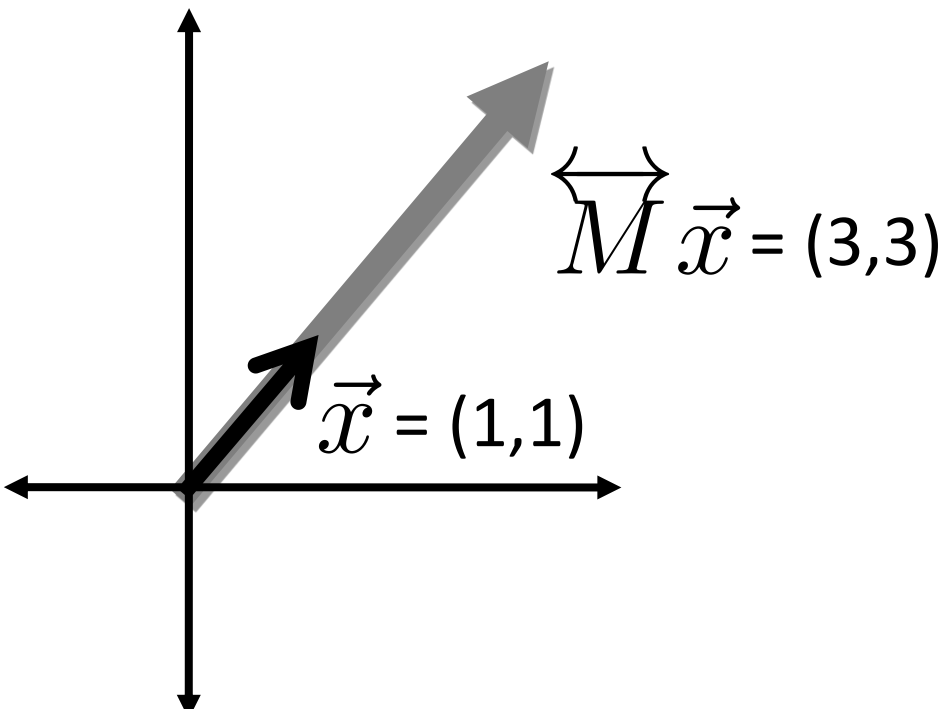
$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



Are there any special vectors that **only get scaled**?

$\overleftrightarrow{M}$

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



- For this special vector, multiplying by M is like multiplying by a scalar.
- (1,1) is called an **eigenvector** of M
- 3 (the scaling factor) is called the **eigenvalue** associated with this eigenvector

Are there any other eigenvectors?

- Yes! The easiest way to find is with MATLAB's eig command.

$$\vec{e}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; \vec{e}^{(2)} = \begin{pmatrix} -1.5 \\ 1 \end{pmatrix}$$

- *Exercise*: verify that (-1.5, 1) is also an eigenvector of M.
- Note: eigenvectors are only defined up to a scale factor.
  – Conventions are either to make **e**'s unit vectors, or make one of the elements 1

Step back:
Eigenvectors obey this equation

$$\overleftrightarrow{M}\vec{e} = \lambda\vec{e}$$

Step back:
Eigenvectors obey this equation

$$\overleftrightarrow{M}\vec{e} = \lambda\vec{e}$$

$$\text{Solve } \left(\overleftrightarrow{M} - \lambda\overleftrightarrow{1}\right)\vec{e} = 0 \text{ for } \vec{e} \neq 0$$

Step back:
Eigenvectors obey this equation

$$\overleftrightarrow{M}\vec{e} = \lambda\vec{e}$$

$$\text{Solve } \left(\overleftrightarrow{M} - \lambda\overleftrightarrow{1}\right)\vec{e} = 0 \text{ for } \vec{e} \neq 0$$

$$\text{So set } det\left(\overleftrightarrow{M} - \lambda\overleftrightarrow{1}\right) = 0$$

Step back: Eigenvectors obey this equation

$$\overleftrightarrow{M}\vec{e} = \lambda\vec{e}$$

$$\text{Solve } \left(\overleftrightarrow{M} - \lambda\overleftrightarrow{1}\right)\vec{e} = 0 \text{ for } \vec{e} \neq 0$$

$$\text{So set } det\left(\overleftrightarrow{M} - \lambda\overleftrightarrow{1}\right) = 0$$

- This is called the characteristic equation for λ
- In general, for an N x N matrix, there are N eigenvectors

BREAK

Part 4: Examples (on blackboard)

- Principal Components Analysis (PCA)
- Single, linear differential equation
- Coupled differential equations

Part 5: Recap & Additional useful stuff

- Matrix diagonalization recap:
transforming between original & eigenvector coordinates
- More special matrices & matrix properties
- Singular Value Decomposition (SVD)

Coupled differential equations

$$\frac{d\vec{x}}{dt} = \overrightarrow{M}\vec{x} + \overrightarrow{I}$$

- Calculate the eigenvectors and eigenvalues.
  - Eigenvalues have typical form:

$$\lambda = \lambda_R + \lambda_I i, \text{ where } i = \sqrt{-1}$$

- The corresponding eigenvector component has dynamics: $e^{\lambda t} = \underbrace{e^{\lambda_R t}}\underbrace{e^{i\lambda_I t}}$

$\lambda_I \neq 0$: oscillations of ang. freq. λ_I

$(e^{i\lambda_I t} = cos(\lambda_I t) + isin(\lambda_I t))$

$\lambda_R < 0$: stable

$\lambda_R > 0$: unstable

Practical program for approaching equations coupled through a term **Mx**

- **Step 1**: Find the eigenvalues and eigenvectors of **M**.
- **Step 2**: Decompose **x** into its eigenvector components
- **Step 3**: Stretch/scale each eigenvalue component
- **Step 4:** (solve for c and) transform back to original coordinates.

eig(M) in MATLAB

Practical program for approaching equations coupled through a term **Mx**

- **Step 1**: Find the eigenvalues and eigenvectors of **M**.
- **Step 2**: Decompose **x** into its eigenvector components
- **Step 3**: Stretch/scale each eigenvalue component
- **Step 4:** (solve for c and) transform back to original coordinates.

$$\vec{x} = c_1\vec{e}^{(1)} + c_2\vec{e}^{(2)} + \cdots + c_n\vec{e}^{(n)}$$

Practical program for approaching equations coupled through a term **Mx**

- **Step 1**: Find the eigenvalues and eigenvectors of **M**.
- **Step 2**: Decompose **x** into its eigenvector components
- **Step 3**: Stretch/scale each eigenvalue component
- **Step 4:** (solve for c and) transform back to original coordinates.

$$\vec{x} = c_1\vec{e}^{(1)} + c_2\vec{e}^{(2)} + \cdots + c_n\vec{e}^{(n)}$$

$$= \begin{pmatrix} | & | & & | \\ \vec{e}^{(1)} & \vec{e}^{(2)} & \cdots & \vec{e}^{(n)} \\ | & | & & | \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

Practical program for approaching equations coupled through a term **Mx**

- **Step 1**: Find the eigenvalues and eigenvectors of **M**.
- **Step 2**: Decompose **x** into its eigenvector components
- **Step 3**: Stretch/scale each eigenvalue component
- **Step 4:** (solve for c and) transform back to original coordinates.

$$\vec{x} = c_1 \vec{e}^{(1)} + c_2 \vec{e}^{(2)} + \cdots + c_n \vec{e}^{(n)}$$

$$= \begin{pmatrix} | & | & & | \\ \vec{e}^{(1)} & \vec{e}^{(2)} & \cdots & \vec{e}^{(n)} \\ | & | & & | \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

$$= \overleftrightarrow{E} \vec{c}$$

$$\text{or, } \vec{c} = \overleftrightarrow{E}^{-1} \vec{x}$$

Practical program for approaching equations coupled through a term **Mx**

- **Step 1**: Find the eigenvalues and eigenvectors of **M**.
- **Step 2**: Decompose **x** into its eigenvector components
- **Step 3**: Stretch/scale each eigenvalue component
- **Step 4:** (solve for c and) transform back to original coordinates.

$$
\begin{aligned}
\overleftrightarrow{M}\vec{x} &= \overleftrightarrow{M}\left(c_1\vec{e}^{(1)} + c_2\vec{e}^{(2)} + \cdots + c_n\vec{e}^{(n)}\right) \\
&= \overleftrightarrow{M}\left(c_1\vec{e}^{(1)}\right) + \overleftrightarrow{M}\left(c_2\vec{e}^{(2)}\right) + \cdots + \overleftrightarrow{M}\left(c_n\vec{e}^{(n)}\right) \\
&= \lambda_1 c_1\vec{e}^{(1)} + \lambda_2 c_2\vec{e}^{(2)} + \cdots + \lambda_n c_n\vec{e}^{(n)}
\end{aligned}
$$

Practical program for approaching equations coupled through a term **Mx**

- **Step 1**: Find the eigenvalues and eigenvectors of **M**.
- **Step 2**: Decompose **x** into its eigenvector components
- **Step 3**: Stretch/scale each eigenvalue component
- **Step 4:** (solve for c and) transform back to original coordinates.

$$\vec{x} = c_1\vec{e}^{(1)} + c_2\vec{e}^{(2)} + \cdots + c_n\vec{e}^{(n)}$$

$$= \overleftrightarrow{E}\,\vec{c}$$

Putting it all together…

$$\overleftrightarrow{M} = \overleftrightarrow{E}\,\overleftrightarrow{\Lambda}\,\overleftrightarrow{E}^{-1}$$

Where (step 1):

$$\overleftrightarrow{E} = \begin{pmatrix} | & | & & | \\ \vec{e}^{(1)} & \vec{e}^{(2)} & \cdots & \vec{e}^{(n)} \\ | & | & & | \end{pmatrix} \qquad \overleftrightarrow{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

MATLAB: $(\overleftrightarrow{\Lambda}, \overleftrightarrow{E}) = eig(\overleftrightarrow{M})$

Putting it all together…

$$\overleftrightarrow{M}\vec{x} = \overleftrightarrow{E}\,\overleftrightarrow{\Lambda}\,\overleftrightarrow{E}^{-1}\vec{x}$$

Step 2: Transform into eigencoordinates

Step 3: Scale by λ_i along the i^{th} eigencoordinate

Step 4: Transform back to original coordinate system

Left eigenvectors

$$E^{-1} = \begin{pmatrix} - & \vec{e}^{(1)}_{left} & - \\ - & \vec{e}^{(2)}_{left} & - \\ & \vdots & \\ - & \vec{e}^{(N)}_{left} & - \end{pmatrix}$$

-The rows of E inverse are called the left eigenvectors because they satisfy $E^{-1} M = \Lambda E^{-1}$.

-Together with the eigenvalues, they determine how x is decomposed into each of its eigenvector components.

Putting it all together…

$$\underbrace{\overleftrightarrow{M}}_{\textit{Original Matrix}} = \overleftrightarrow{E}\,\underbrace{\overleftrightarrow{\Lambda}}_{\textit{Matrix in eigencoordinate system}}\,\overleftrightarrow{E}^{-1}$$

Where:

$$\overleftrightarrow{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

Trace and Determinant

$$\underbrace{\overleftrightarrow{M}}_{\textit{Original Matrix}} = \overleftrightarrow{E}\,\underbrace{\overleftrightarrow{\Lambda}}_{\textit{Matrix in eigencoordinate system}}\,\overleftrightarrow{E}^{-1}$$

- Note: **M** and **Lambda** look very different.

  Q: Are there any properties that are preserved between them?

  A: Yes, 2 very important ones:

$$1.\ \text{sum of diagonal entries } = Tr(\overleftrightarrow{M}) = Tr(\overleftrightarrow{\Lambda})$$

$$= \lambda_1 + \lambda_2 + \cdots + \lambda_N = \sum_{i=1}^{N} \lambda_i$$

$$2.\ det(\overleftrightarrow{M}) = det(\overleftrightarrow{\Lambda}) = \lambda_1 \lambda_2 \cdots \lambda_N = \prod_{i=1}^{N} \lambda_i$$

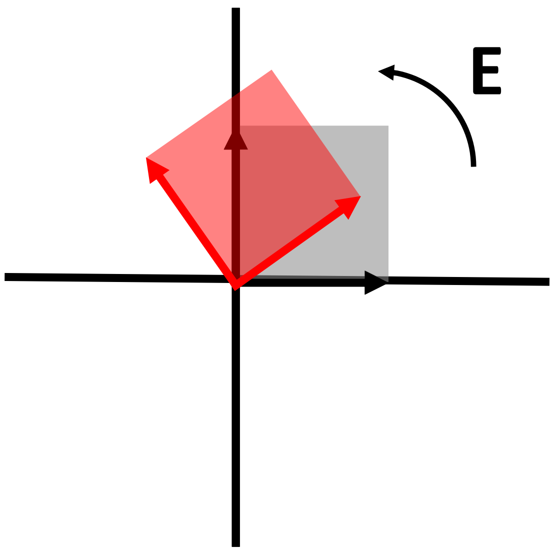
Special Matrices: Normal matrix

- **Normal matrix:** all eigenvectors are orthogonal

  → Can transform to eigencoordinates ("change basis") with a simple rotation* of the coordinate axes

  → A normal matrix's eigenvector matrix **E** is a *generalized rotation (unitary or orthonormal) matrix, defined by:

Picture:



$$\overleftrightarrow{E}^{-1} = \overleftrightarrow{E}^{T}$$

(*note: generalized means one can also do reflections of the eigenvectors through a line/plane")

Special Matrices: Normal matrix

- **Normal matrix:** all eigenvectors are orthogonal

  → Can transform to eigencoordinates ("change basis") with a simple rotation of the coordinate axes

  → **E** is a rotation (unitary or orthogonal) matrix, defined by:

$$\overleftrightarrow{E}^{-1} = \overleftrightarrow{E}^{T}$$

where if:

$$\overleftrightarrow{E} = \begin{pmatrix} E_{11} & E_{12} & \cdots & E_{1P} \\ E_{21} & E_{22} & \cdots & E_{2P} \\ \vdots & \vdots & & \vdots \\ E_{N1} & E_{N2} & \cdots & E_{NP} \end{pmatrix}$$

then:

$$\overleftrightarrow{E}^{T} = \begin{pmatrix} E_{11} & E_{21} & \cdots & E_{N1} \\ E_{12} & E_{22} & \cdots & E_{N2} \\ \vdots & \vdots & & \vdots \\ E_{1P} & E_{2P} & \cdots & E_{NP} \end{pmatrix}$$

Special Matrices: Normal matrix

- Eigenvector decomposition in this case:

$$\overleftrightarrow{M} = \overleftrightarrow{E}\,\overleftrightarrow{\Lambda}\,\overleftrightarrow{E}^{-1}$$

$$= \overleftrightarrow{E}\,\overleftrightarrow{\Lambda}\,\overleftrightarrow{E}^{T}$$

$$= \begin{pmatrix} | & | & & | \\ \vec{e}^{(1)} & \vec{e}^{(2)} & \cdots & \vec{e}^{(N)} \\ | & | & & | \end{pmatrix} \overleftrightarrow{\Lambda} \begin{pmatrix} - & \vec{e}^{(1)} & - \\ - & \vec{e}^{(2)} & - \\ & \vdots & \\ - & \vec{e}^{(N)} & - \end{pmatrix}$$

- Left and right eigenvectors are identical!

Special Matrices

- **Symmetric Matrix:** $\overleftrightarrow{S}^T = \overleftrightarrow{S}$
  - e.g. Covariance matrices, Hopfield network
  - Properties:
    - Eigenvalues are real
    - Eigenvectors are orthogonal (i.e. it's a normal matrix)

SVD: Decomposes matrix into outer products (e.g. of a neural/spatial mode and a temporal mode)

$$\underbrace{\vec{M}}_{\mathrm{N\ x\ T}} = \begin{array}{c} \begin{array}{cccc} t = 1 & t = 2 & & t = T \end{array} & \\ \begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1T} \\ M_{21} & M_{22} & \cdots & M_{2T} \\ \vdots & \vdots & & \vdots \\ M_{N1} & M_{N2} & \cdots & M_{NT} \end{pmatrix} & \begin{array}{l} n = 1 \\ n = 2 \\ \\ n = N \end{array} \end{array},$$

SVD: Decomposes matrix into outer products (e.g. of a neural/spatial mode and a temporal mode)

$$\underbrace{\overrightarrow{M}}_{\mathrm{N\ x\ T}} = \begin{array}{cc} \begin{array}{cccc} t = 1 & t = 2 & & t = T \end{array} & \\ \begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1T} \\ M_{21} & M_{22} & \cdots & M_{2T} \\ \vdots & \vdots & & \vdots \\ M_{N1} & M_{N2} & \cdots & M_{NT} \end{pmatrix} & \begin{array}{l} n = 1 \\ n = 2 \\ \\ n = N \end{array} \end{array},$$

SVD: Decomposes matrix into outer products (e.g. of a neural/spatial mode and a temporal mode)

$$\underbrace{\overrightarrow{M}}_{\mathrm{N\,x\,T}} = \underbrace{\overrightarrow{U}}_{\mathrm{N\,x\,N}}\,\underbrace{\overrightarrow{S}}_{\mathrm{N\,x\,T}}\,\underbrace{\overrightarrow{V}}_{\mathrm{T\,x\,T}}{}^{T}$$

Columns of U are eigenvectors of MM^T

Rows of V^T are eigenvectors of M^TM

$$= \begin{pmatrix} | & | & & | \\ \vec{u}^{(1)} & \vec{u}^{(2)} & \cdots & \vec{u}^{(N)} \\ | & | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \lambda_N & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} - & \vec{v}^{(1)} & - \\ - & \vec{v}^{(2)} & - \\ & \vdots & \\ - & \vec{v}^{(N)} & - \\ & other & \end{pmatrix}$$

- Note: the eigenvalues are the same for M^TM and MM^T

SVD: Decomposes matrix into outer products (e.g. of a neural/spatial mode and a temporal mode)

Columns of U are eigenvectors of MM^T

$$\overrightarrow{M} = \overrightarrow{U}\,\overrightarrow{S}\,\overrightarrow{V}^T$$

Rows of V^T are eigenvectors of M^TM

$$= \begin{pmatrix} | & | & & | \\ \vec{u}^{(1)} & \vec{u}^{(2)} & \cdots & \vec{u}^{(N)} \\ | & | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \lambda_N & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} - & \vec{v}^{(1)} & - \\ - & \vec{v}^{(2)} & - \\ & \vdots & \\ - & \vec{v}^{(N)} & - \\ & other & \end{pmatrix}$$

$$= \begin{pmatrix} | & | & & | \\ \vec{u}^{(1)} & \vec{u}^{(2)} & \cdots & \vec{u}^{(N)} \\ | & | & & | \end{pmatrix} \begin{pmatrix} - & \lambda_1\vec{v}^{(1)} & - \\ - & \lambda_2\vec{v}^{(2)} & - \\ & \vdots & \\ - & \lambda_N\vec{v}^{(N)} & - \end{pmatrix}$$

$$= \lambda_1 \begin{pmatrix} | \\ \vec{u}^{(1)} \\ | \end{pmatrix} \begin{pmatrix} - & \vec{v}^{(1)} & - \end{pmatrix} + \lambda_2 \begin{pmatrix} | \\ \vec{u}^{(2)} \\ | \end{pmatrix} \begin{pmatrix} - & \vec{v}^{(2)} & - \end{pmatrix} + \cdots + \lambda_N \begin{pmatrix} | \\ \vec{u}^{(N)} \\ | \end{pmatrix} \begin{pmatrix} - & \vec{v}^{(N)} & - \end{pmatrix}$$

- Thus, SVD pairs “spatial” patterns with associated “temporal” profiles through the outer product

The End