Yet More Linear Algebra

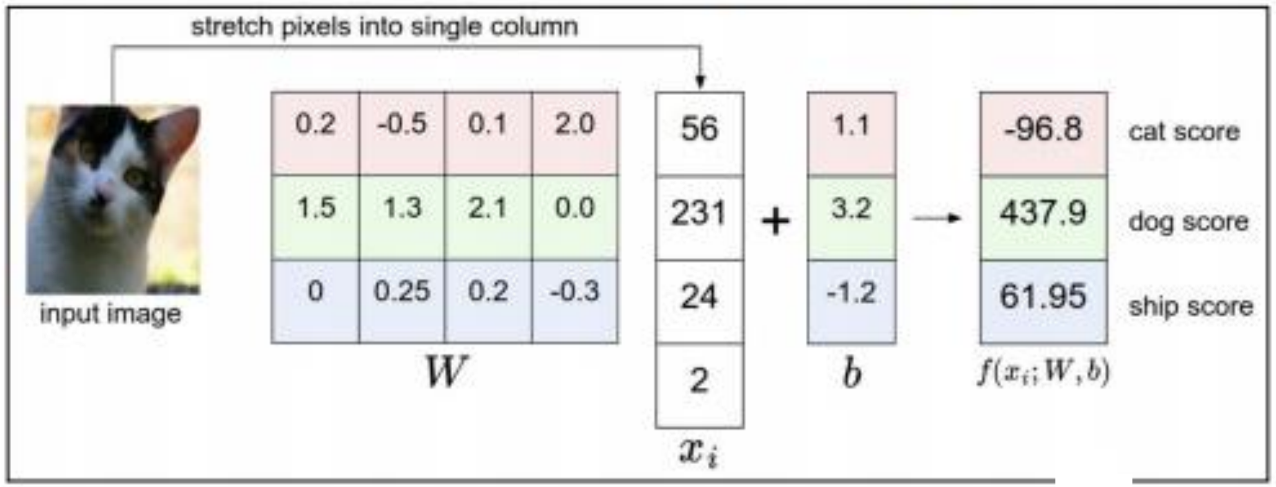
Alexander G. Ororbia II

COGS-621: Foundations of Scientific Computing

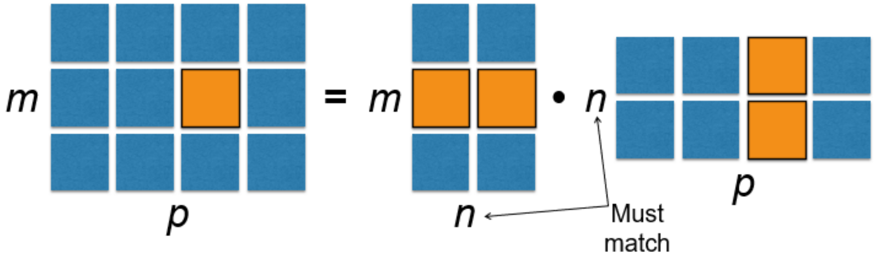
9/30/2025

Tensors in statistical learning

A linear classifier $y = Wx^{T} + b$



Vector x is converted into vector y by multiplying x by a matrix W



Identity matrices

Creating identity matrix in NumPy:

```
In [62]: np.identity(4)
Out[62]: array([[ 1.,  0.,  0.,  0.],
                [ 0.,  1.,  0.,  0.],
                [ 0.,  0.,  1.,  0.],
                [ 0.,  0.,  0.,  1.]])
```

```
In [63]: np.eye(3, k=1)
Out[63]: array([[ 0.,  1.,  0.],
                [ 0.,  0.,  1.],
                [ 0.,  0.,  0.]])
In [64]: np.eye(3, k=-1)
Out[64]: array([[ 0.,  0.,  0.],
                [ 1.,  0.,  0.],
                [ 0.,  1.,  0.]])
```

- Matrix inversion is a powerful tool to analytically solve $Ax=b$
- Needs concept of Identity matrix
- Identity matrix does not change value of vector when we multiply the vector by identity matrix
  - Denote identity matrix that preserves n-dimensional vectors as I_n
  - Formally $I_n \in \mathbb{R}^{n \times n}$ and $\forall \mathbf{x} \in \mathbb{R}^n, I_n \mathbf{x} = \mathbf{x}$
  - Example of I_3 $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

18

Norms

- Used for measuring the size of a vector
- Norms map vectors to non-negative values
- Norm of vector $\boldsymbol{x} = [x_1,..,x_n]^{\mathrm{T}}$ is distance from origin to $\boldsymbol{x}$
  - It is any function f that satisfies:

$$f(\boldsymbol{x}) = 0 \Rightarrow \boldsymbol{x} = 0$$

$$f(\boldsymbol{x}+\boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}) \quad \text{Triangle Inequality}$$

$$\forall \alpha \in R \quad f(\alpha \boldsymbol{x}) = |\alpha| f(\boldsymbol{x})$$

NumPy Equivalent:

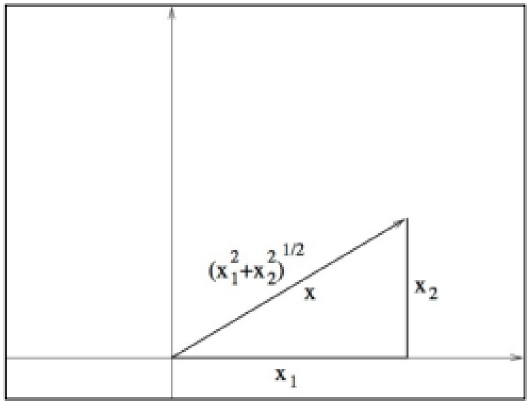
```
>>> x = np.array([[6., 9.]])
>>> y = np.array([[14., 18.]])
>>> np.linalg.norm(x + y)
33.60059523282288
>>> np.linalg.norm(x) + np.linalg.norm(y)
33.620162328374725
>>>
```

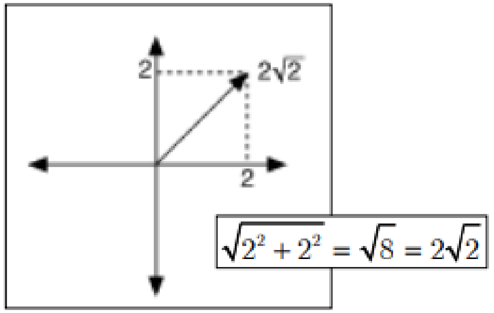
L^p Norm

```
linalg.norm(x, ord=None, axis=None, keepdims=False)
```

- Definition:

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





  - L^2 Norm
    - Called Euclidean norm
      - Simply the Euclidean distance between the origin and the point x
      - written simply as $\|\boldsymbol{x}\|$
      - Squared Euclidean norm is same as $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$
  - L^1 Norm
    - Useful when 0 and non-zero have to be distinguished
      - Note that L^2 increases slowly near origin, e.g., $0.1^2=0.01$)
  - L^∞ Norm

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

    - Called max norm

The Frobenius norm

Similar to L^2 norm

$$\|A\|_F = \left(\sum_{i,j} A_{i,j}^2\right)^{\frac{1}{2}}$$

$$A = \begin{bmatrix} 2 & -1 & 5 \\ 0 & 2 & 1 \\ 3 & 1 & 1 \end{bmatrix} \quad \|A\| = \sqrt{4 + 1 + 25 + .. + 1} = \sqrt{46}$$

Norms Can Serve as the Building Blocks for Distance Measurements!

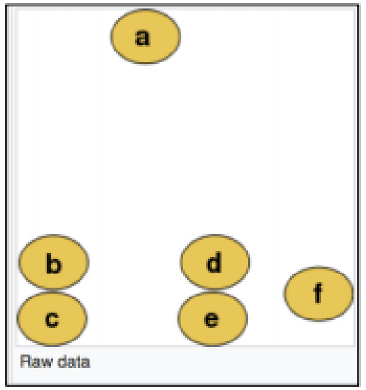
- Distance between two vectors (v, w)
  - $\text{dist}(v,w) = ||v - w||$

$$= \sqrt{(v_1 - w_1)^2 + .. + (v_n - w_n)^2}$$

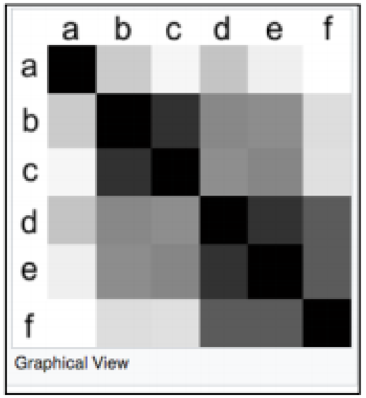
Euclidean distance

The symmetric matrix

- A symmetric matrix equals its transpose: $A = A^T$
  - E.g., a distance matrix is symmetric with $A_{ij} = A_{ji}$



	a	b	c	d	e	f
a	0	184	222	177	216	231
b	184	0	45	123	128	200
c	222	45	0	129	121	203
d	177	123	129	0	46	83
e	216	128	121	46	0	83
f	231	200	203	83	83	0



  - E.g., covariance matrices are symmetric

$$\Sigma = \begin{pmatrix} 1 & .5 & .15 & .15 & 0 & 0 \\ .5 & 1 & .15 & .15 & 0 & 0 \\ .15 & .15 & 1 & .25 & 0 & 0 \\ .15 & .15 & .25 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & .10 \\ 0 & 0 & 0 & 0 & .10 & 1 \end{pmatrix}$$

Inversion: use numpy.linalg

A

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

Determinant, i.e., det(A) = |A|

$$A^{-1}A = I_n$$

$$Ax = b$$

$$A^{-1}Ax = A^{-1}b$$

$$I_n x = A^{-1}b$$

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

Standard inversion operator:

```
>>> np.linalg.inv(X)
array([[-2. ,  1. ],
       [ 1.5, -0.5]])
>>>
```

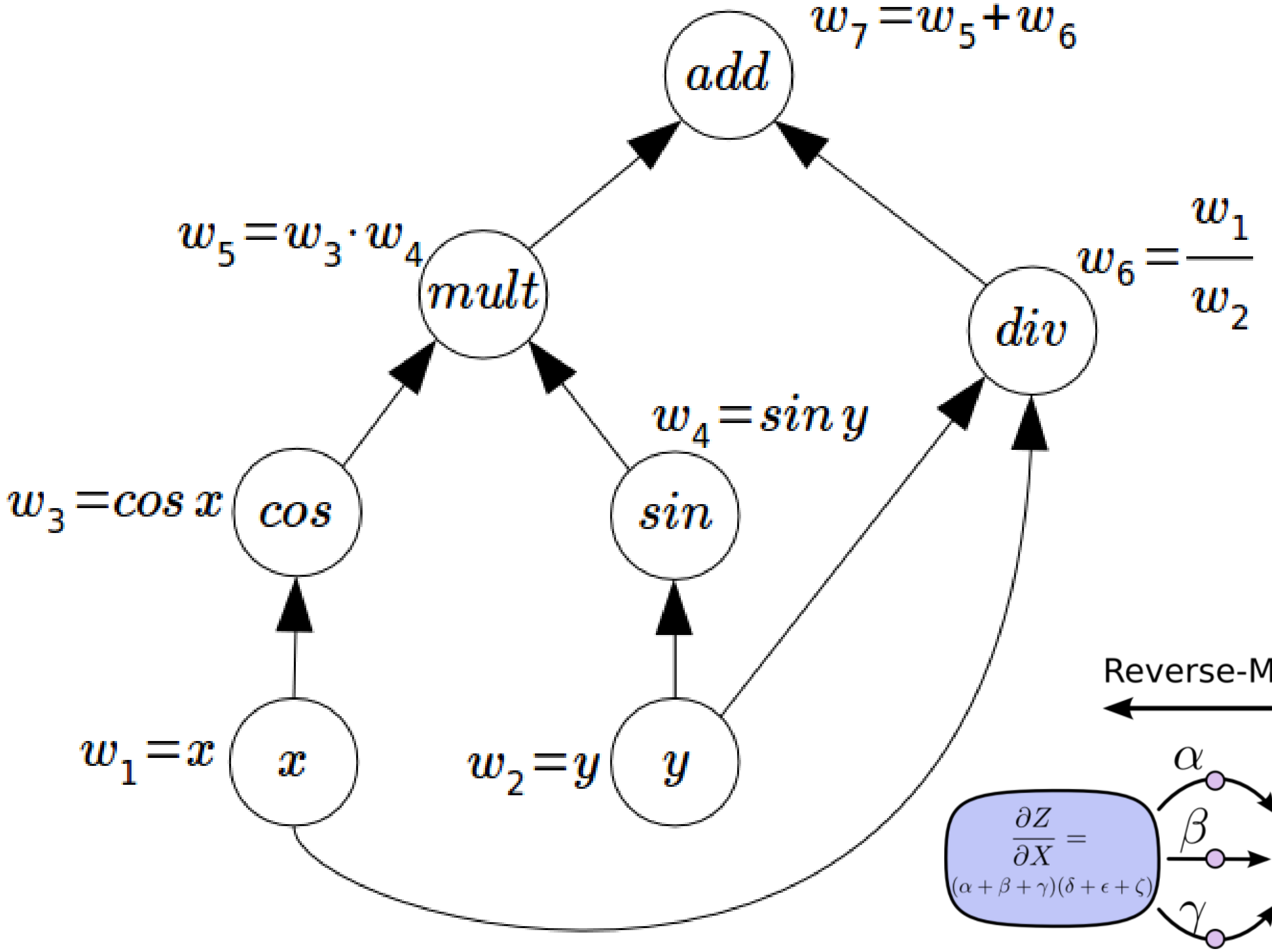
(Moore-Penrose) pseudo-inverse operator:

```
>>> np.linalg.pinv(X)
array([[-2. ,  1. ],
       [ 1.5, -0.5]])
```

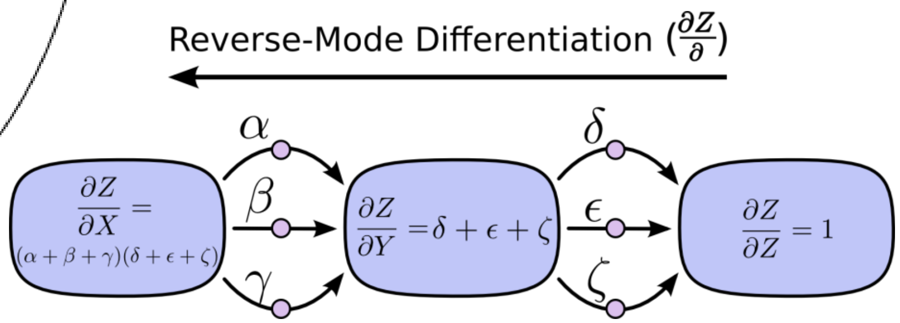
Sometimes, there is no matrix inverse!

Why do we care about ops?

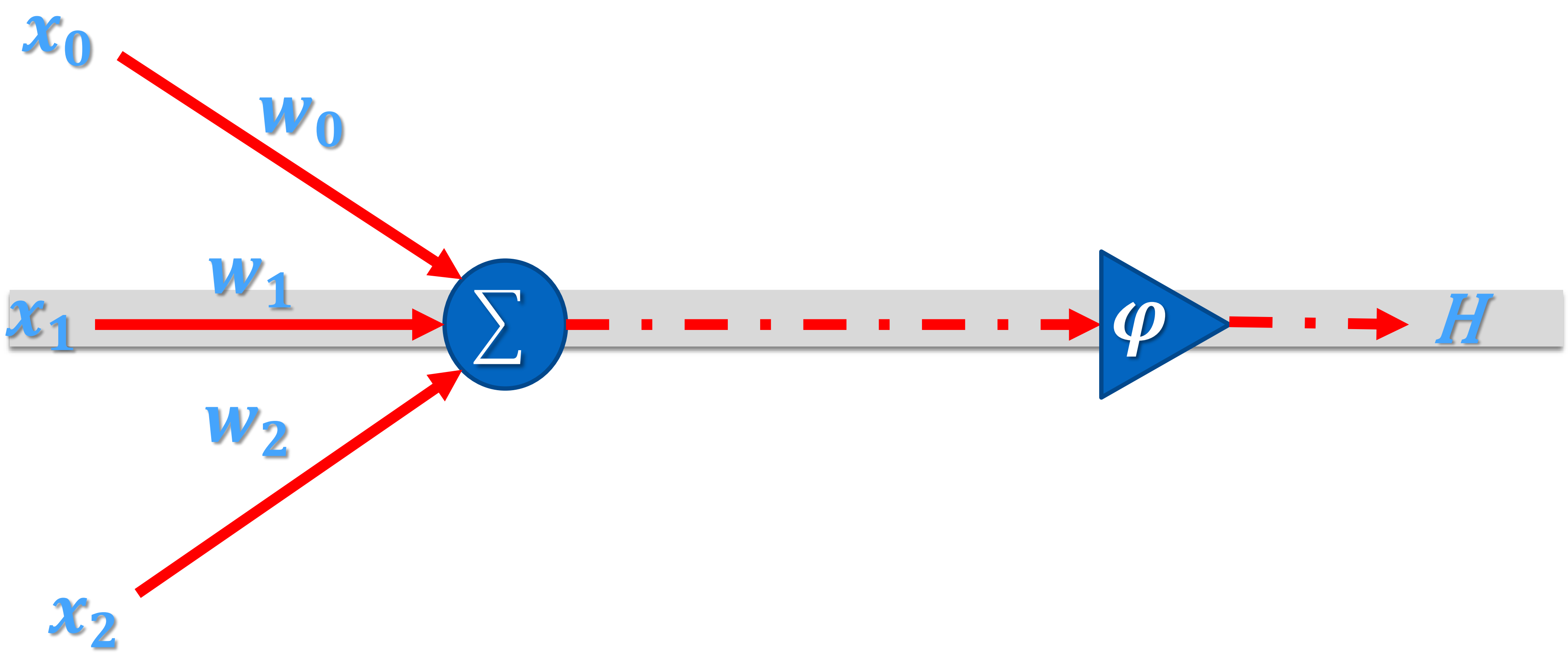
Computation graphs



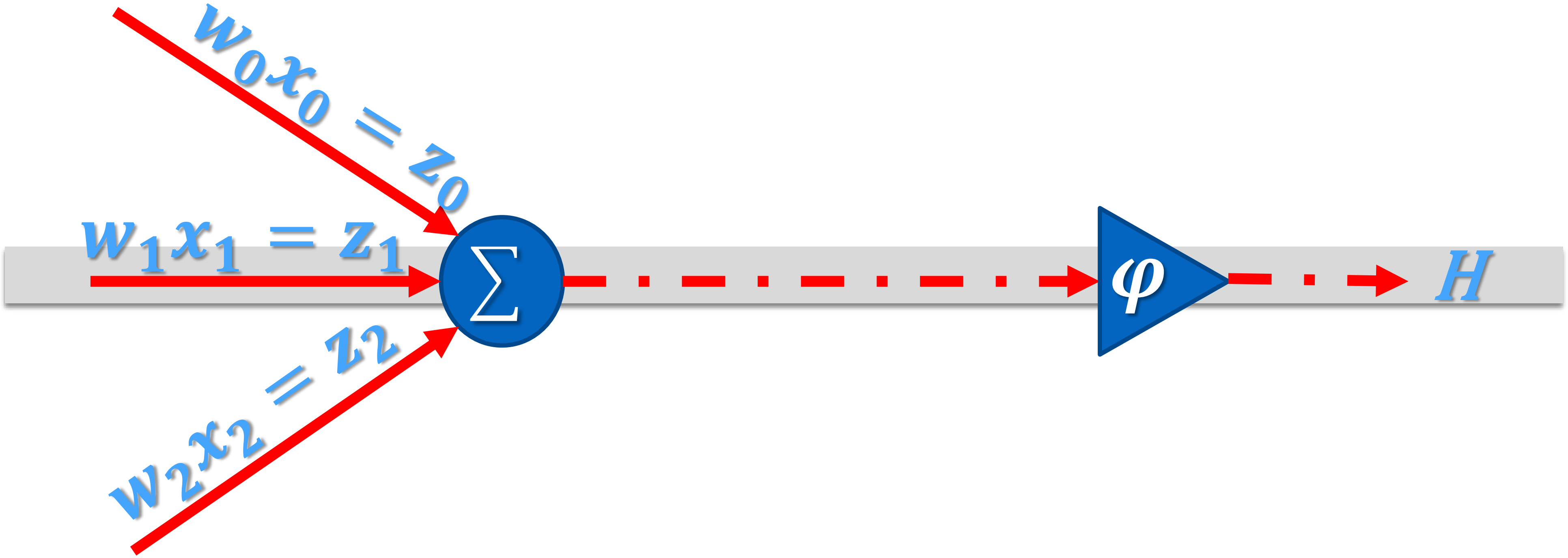
Linear algebra operators can be arranged in a directed graph!



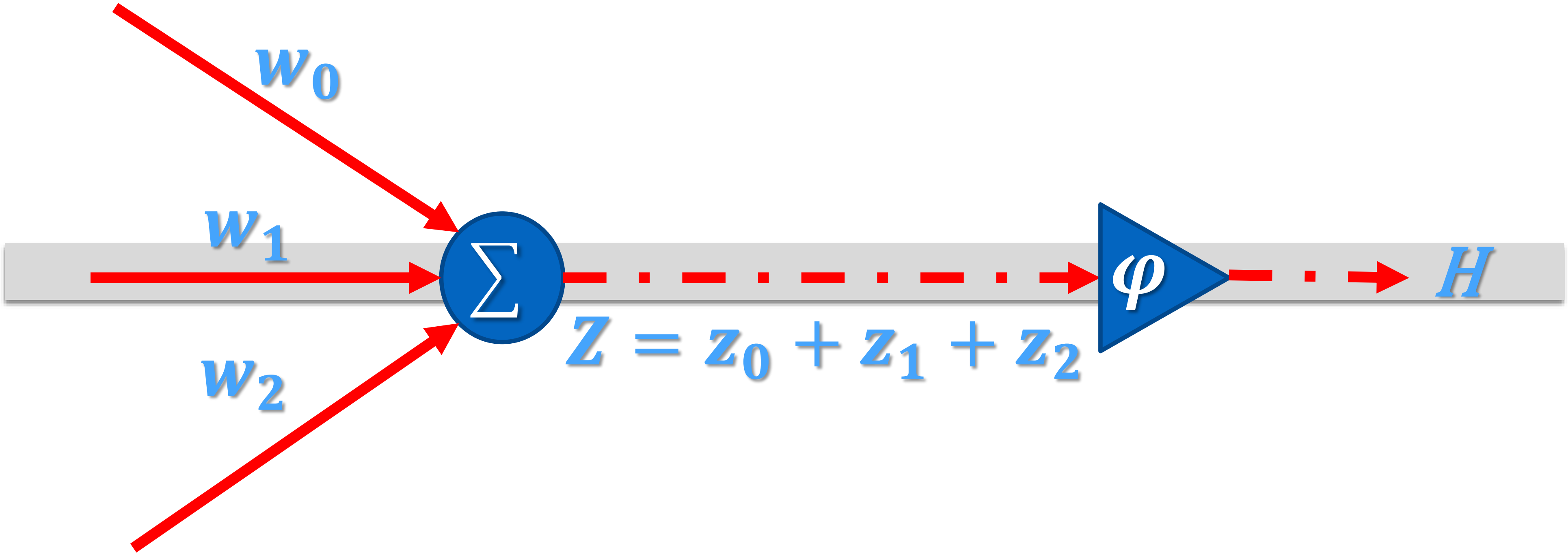
A Simple Linear Predictor



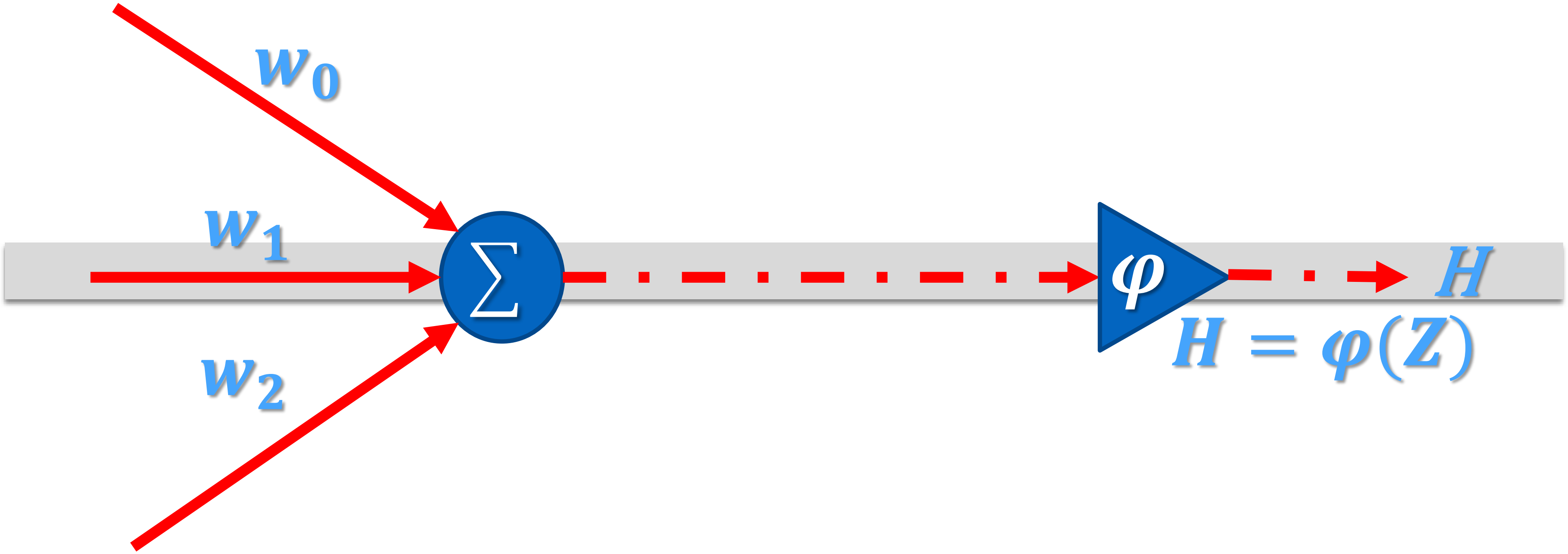
A Simple Linear Predictor



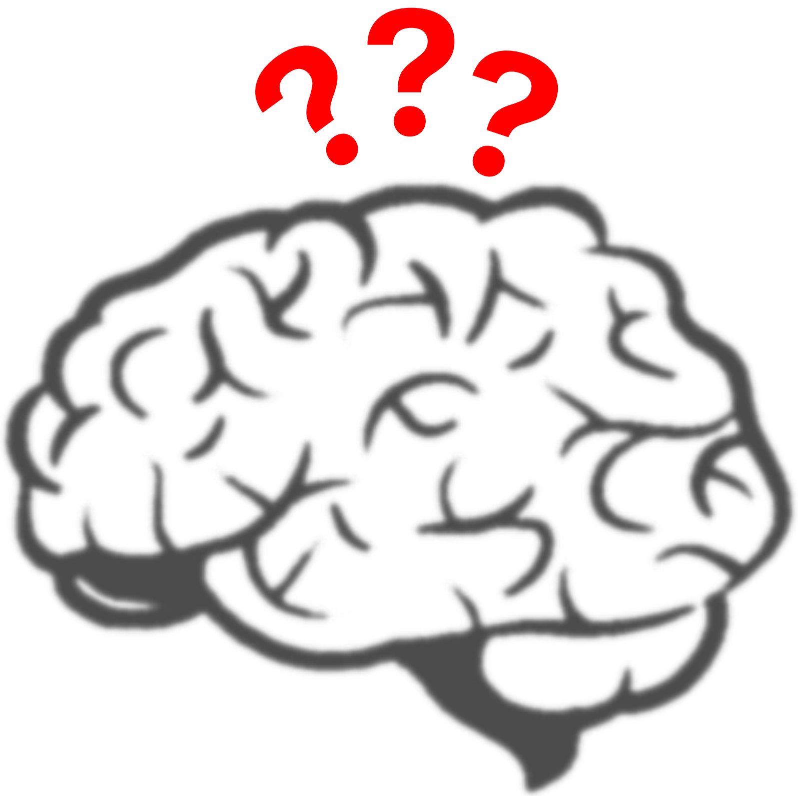
A Simple Linear Predictor



A Simple Linear Predictor



Questions?



Numerical Optimization

Alexander G. Ororbia II

COGS-621: Foundations of Scientific Computing

9/30/2025

Categories of decision making problems

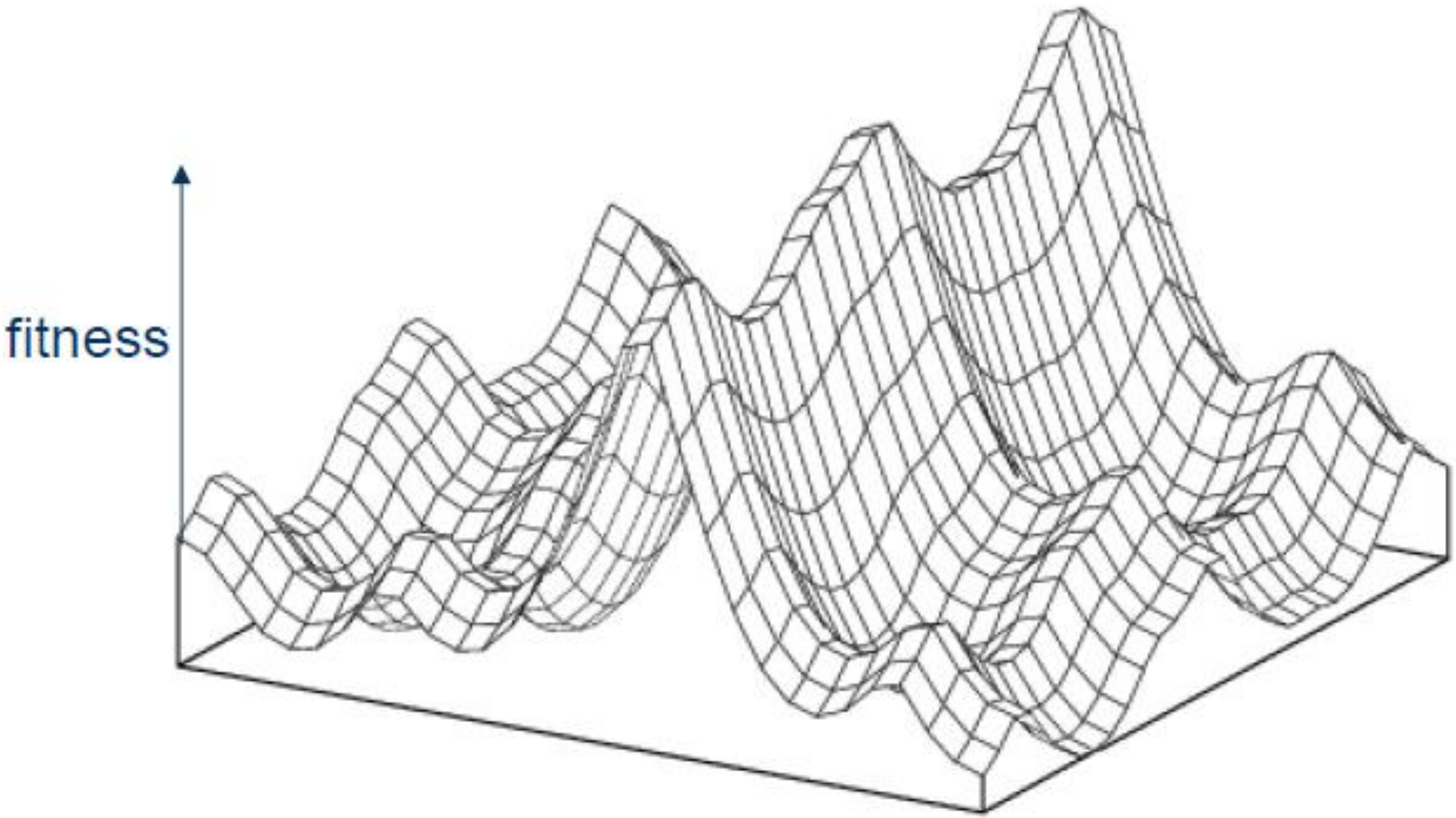
Category 1:

- The set of possible alternatives for the decision is a finite discrete set typically consisting of a small number of elements.
  - Example: “A teenage girl knows four boys all of whom she likes, and has to decide who among them to go steady with.”
- Solution: scoring methods

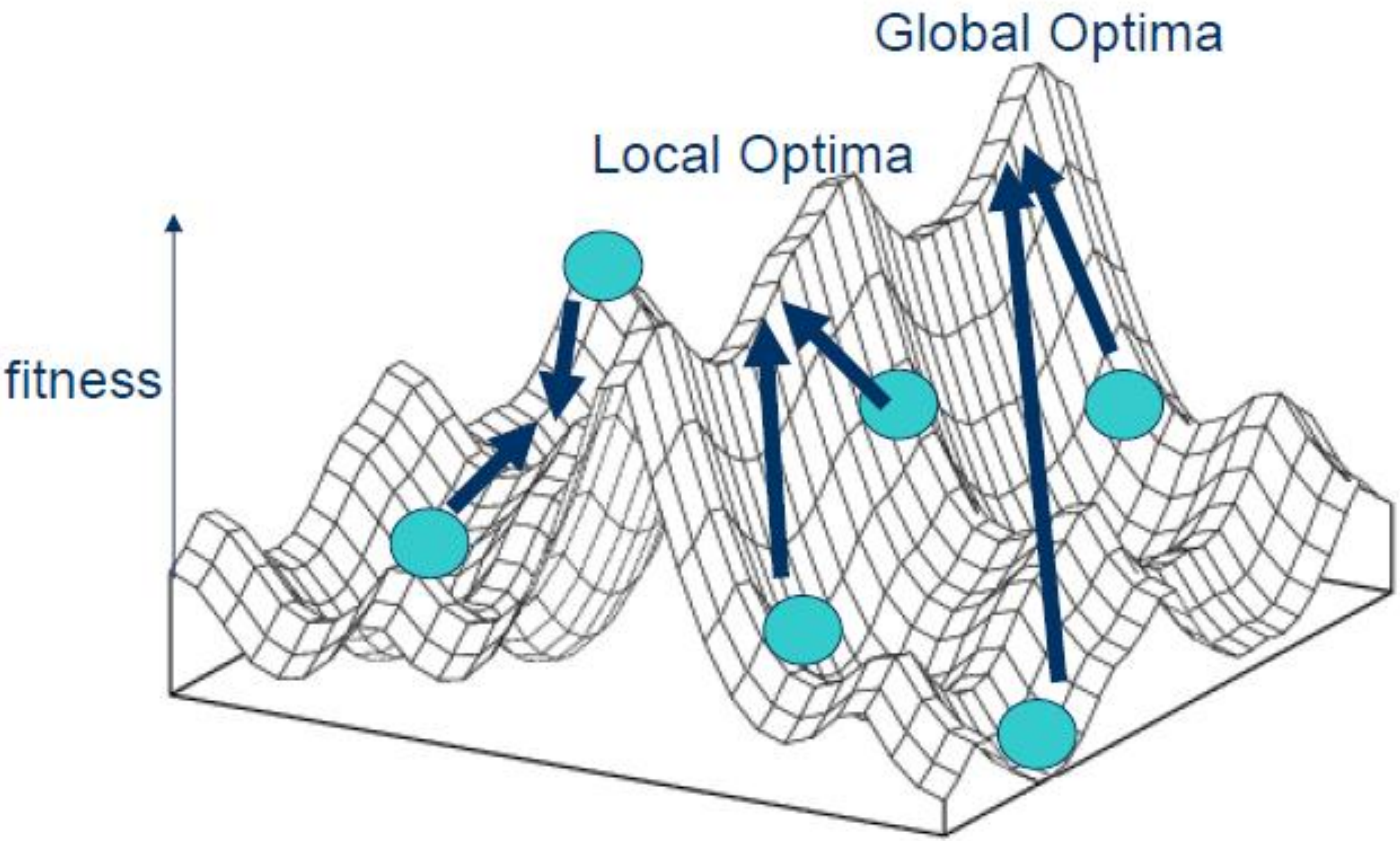
Category 2:

- The number of possible alternatives is either infinite, or finite but very large, and the decision may be required to satisfy some restrictions and constraints
- Solution: unconstrained and constrained optimization methods

Combinatorial Problems: Fitness Landscape



Combinatorial Problems: Fitness Landscape



So… what is mathematical optimization, anyway?

"Optimization" comes from the same root as "optimal", which means *best*. When you optimize something, you are "making it best".

But "best" can vary. If you're a football player, you might want to maximize your running yards, and also minimize your fumbles. Both maximizing and minimizing are types of optimization problems.

In the modern world, pennies matter, microseconds matter, microns matter.

Category 2 Decision Problems & Solution Flow

1. Get a precise definition of the problem, all relevant data and information on it.
    - Uncontrollable factors (random variables)
    - Controllable inputs (decision variables)

2. Construct a mathematical (optimization) model of the problem.
    - Build objective functions and constraints

3. Solve the model
    - Apply the most appropriate algorithms for the given problem

4. Implement the solution

Mathematical Optimization in the "Real World"

Mathematical Optimization is a branch of applied mathematics which is useful in many different fields. Here are a few examples:

- Manufacturing
- Production
- Inventory control
- Transportation
- Scheduling
- Networks
- Finance
- Engineering
- Mechanics
- Economics
- Control engineering
- Marketing
- Policy Modeling

Optimization Vocabulary

Your basic optimization problem consists of…

- The objective function, $f(x)$, which is the output you're trying to maximize or minimize.

Optimization Vocabulary

Your basic optimization problem consists of…

- The objective function, $f(x)$, which is the output you're trying to maximize or minimize.
- Variables, x_1 x_2 x_3 and so on, which are the inputs – things you can control. They are abbreviated x_n to refer to individuals or x to refer to them as a group.

Optimization Vocabulary

Your basic optimization problem consists of…

- The objective function, $f(x)$, which is the output you're trying to maximize or minimize.
- Variables, x_1 x_2 x_3 and so on, which are the inputs – things you can control. They are abbreviated x_n to refer to individuals or x to refer to them as a group.
- Constraints, which are equations that place limits on how big or small some variables can get. Equality constraints are usually noted $h_n(x)$ and inequality constraints are noted $g_n(x)$.

Optimization Vocabulary

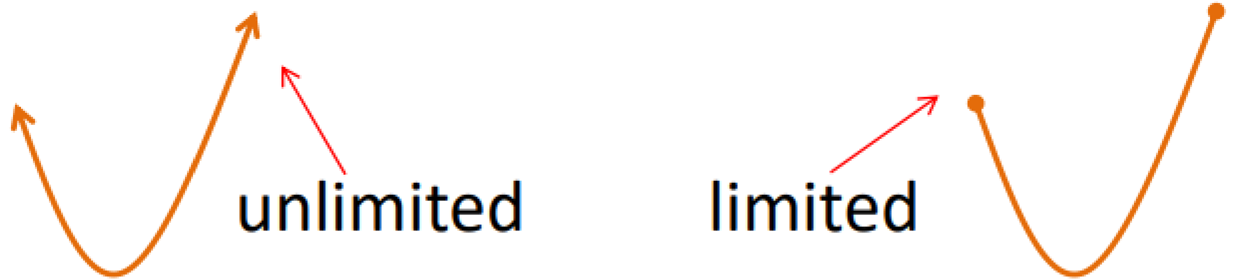
A football coach is planning practices for his running backs.

- His main goal is to maximize running yards – this will become his objective function.
- He can make his athletes spend practice time in the weight room; running sprints; or practicing ball protection. The amount of time spent on each is a variable.
- However, there are limits to the total amount of time he has. Also, if he completely sacrifices ball protection he may see running yards go up, but also fumbles, so he may place an upper limit on the amount of fumbles he considers acceptable. These are constraints.

Note that the variables influence the objective function and the constraints place limits on the domain of the variables.

Types of Optimization Problems

- Some problems have constraints and some do not.



Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.

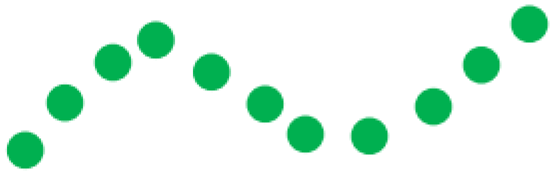
x_1 x_3 x_4 x_2

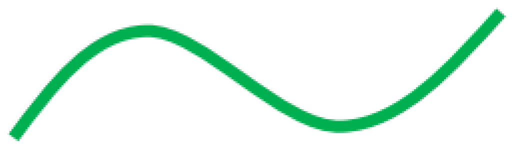
x_6 x_5 x_7

x_8

Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.





Types of Optimization Problems

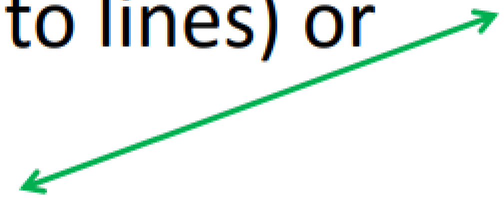
- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.
- Some problems are static (do not change over time) while some are dynamic (continual adjustments must be made as changes occur).

Types of Optimization Problems

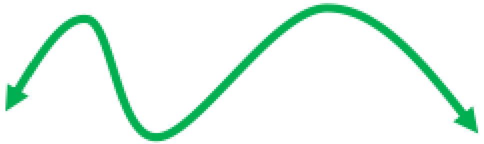
- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.
- Some problems are static (do not change over time) while some are dynamic (continual adjustments must be made as changes occur).
- Systems can be deterministic (specific causes produce specific effects) or stochastic (involve randomness/ probability).

Types of Optimization Problems

- Some problems have constraints and some do not.
- There can be one variable or many.
- Variables can be discrete (for example, only have integer values) or continuous.
- Some problems are static (do not change over time) while some are dynamic (continual adjustments must be made as changes occur).
- Systems can be deterministic (specific causes produce specific effects) or stochastic (involve randomness/ probability).
- Equations can be linear (graph to lines) or nonlinear (graph to curves)



Convex vs. non-convex optimization problems!



Why Mathematical Optimization is worth learning

Q: Which of these things is not like the others?

a) A degree in engineering

b) A degree in chemistry

c) A degree in pure mathematics

d) A large pepperoni pizza

Why Mathematical Optimization is worth learning

Q: Which of these things is not like the others?

a) A degree in engineering

b) A degree in chemistry

c) A degree in pure mathematics

d) A large pepperoni pizza

(With the others, you can feed a family of four)

Questions?