INTRODUCTION TO MACHINE LEARNING

Jane Kim

Ohio University

INPP Summer Tutorial Series

13 June 2024

OHIO UNIVERSITY

QUESTIONS

- What is artificial intelligence?
- How is it different from machine learning?
- What are the types of machine learning algorithms?
- How can machine learning be used for physics problems?
- How can you get started?

FOREVER OHIO

OHIO UNIVERSITY

ARTIFICIAL INTELLIGENCE

- A broad field that aims to give computers human-like capabilities, e.g.
  - Interpret large amounts of complicated data
  - Solve problems
  - Adapt to new situations
  - Plan and make decisions autonomously
  - Interact via text, speech, images, etc.
  - Perceive their environment

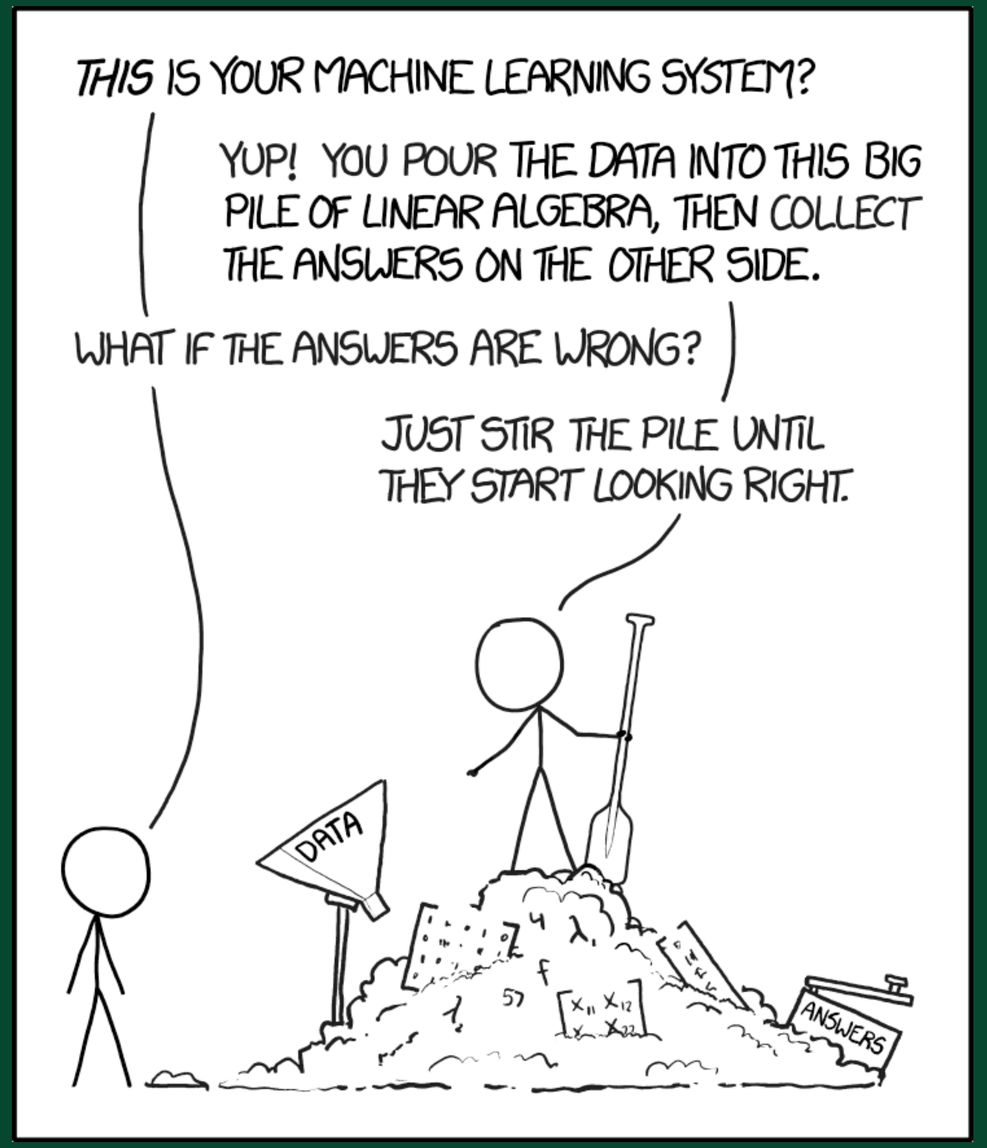
- Goals can be somewhat vague and complex $\Longrightarrow$ Need to break down into smaller pieces!

FOREVER OHIO

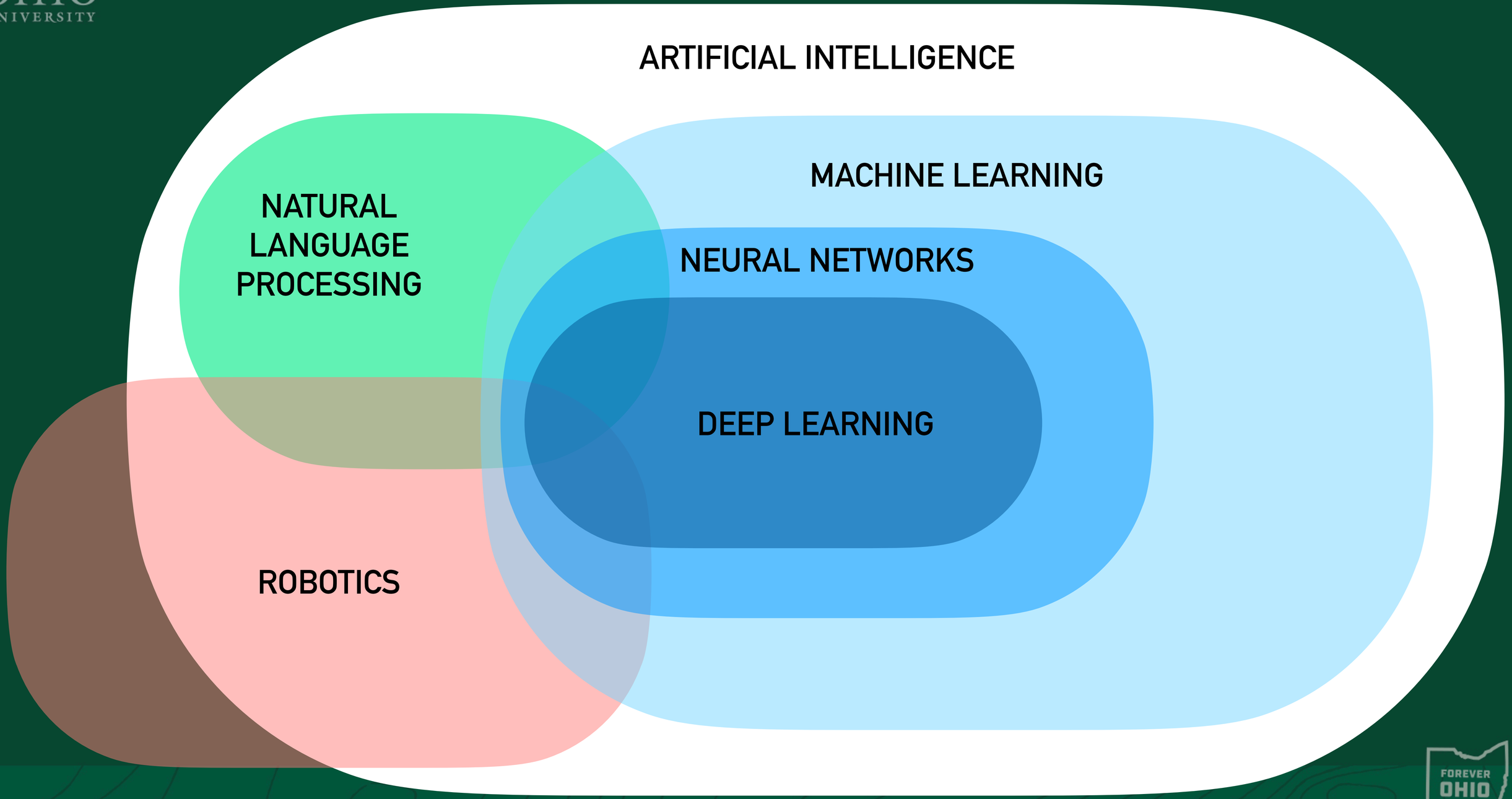
OHIO UNIVERSITY

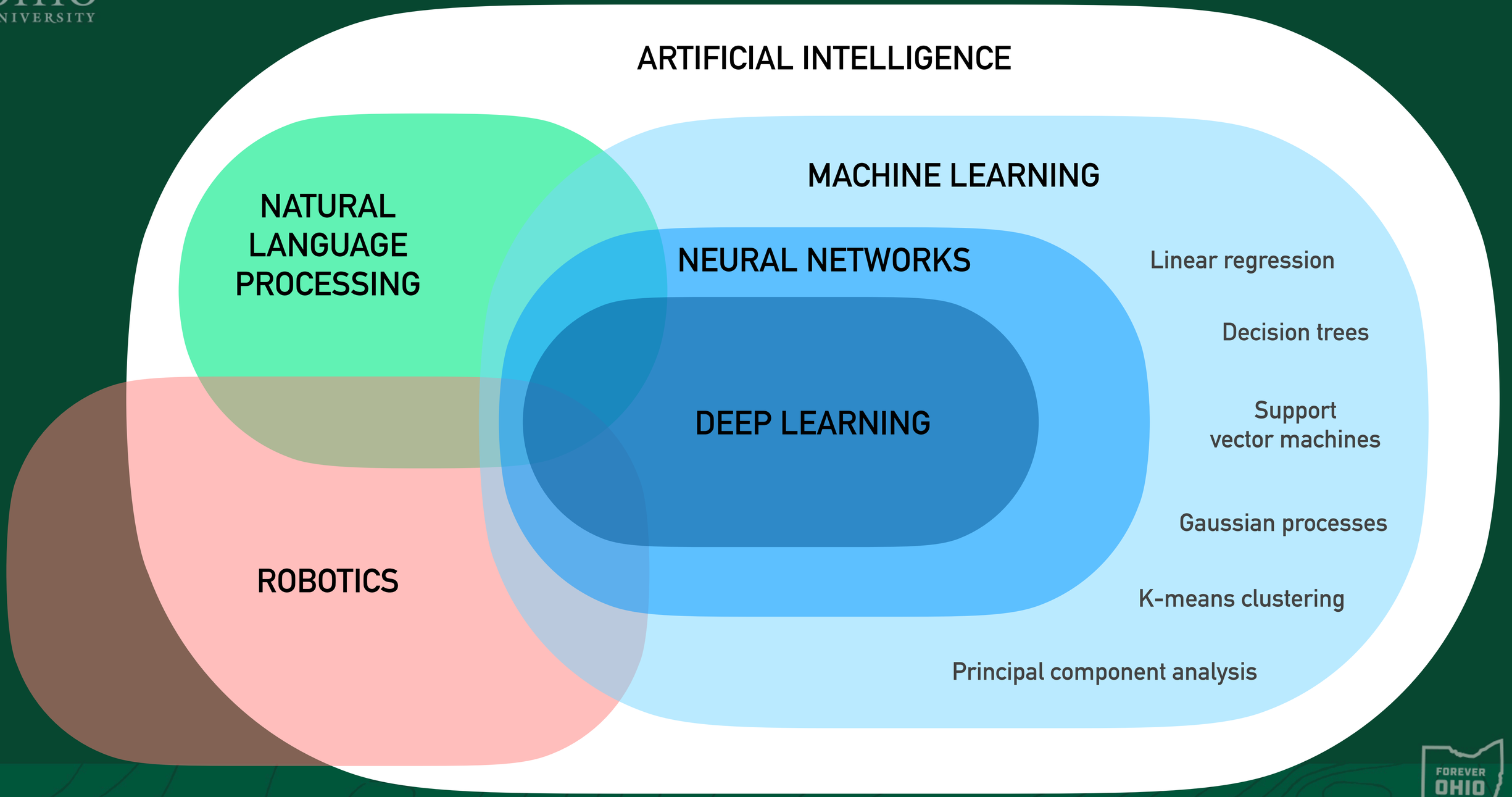
MACHINE LEARNING

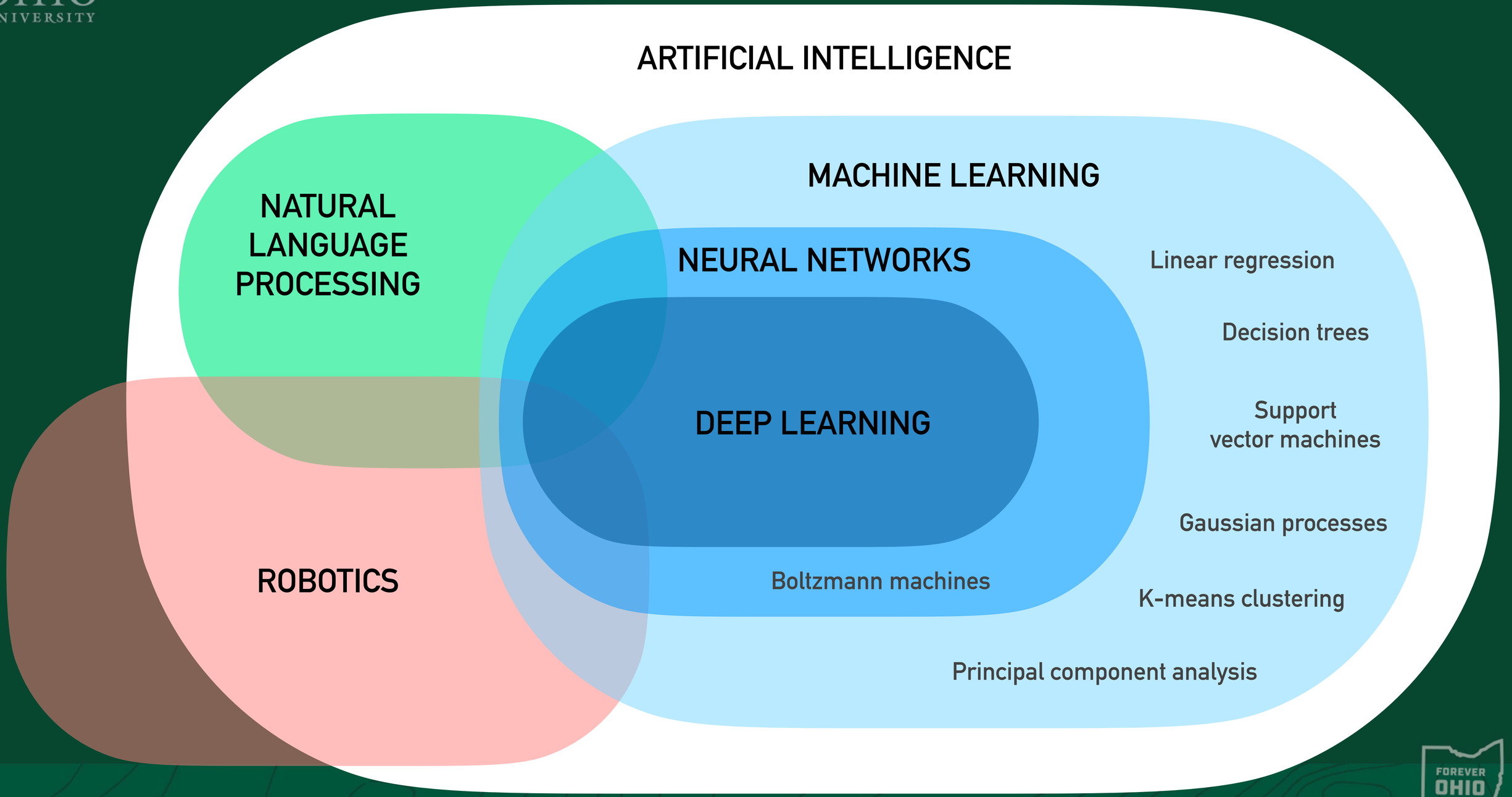
- Computational models and algorithms designed to learn from data rather than rely on explicit programming
- Goals are narrow in scope, e.g.
    - Fit a curve to data
    - Classify data into discrete categories
    - Reduce the dimension of the data
    - Generate fake data that looks like real data
    - So much more…

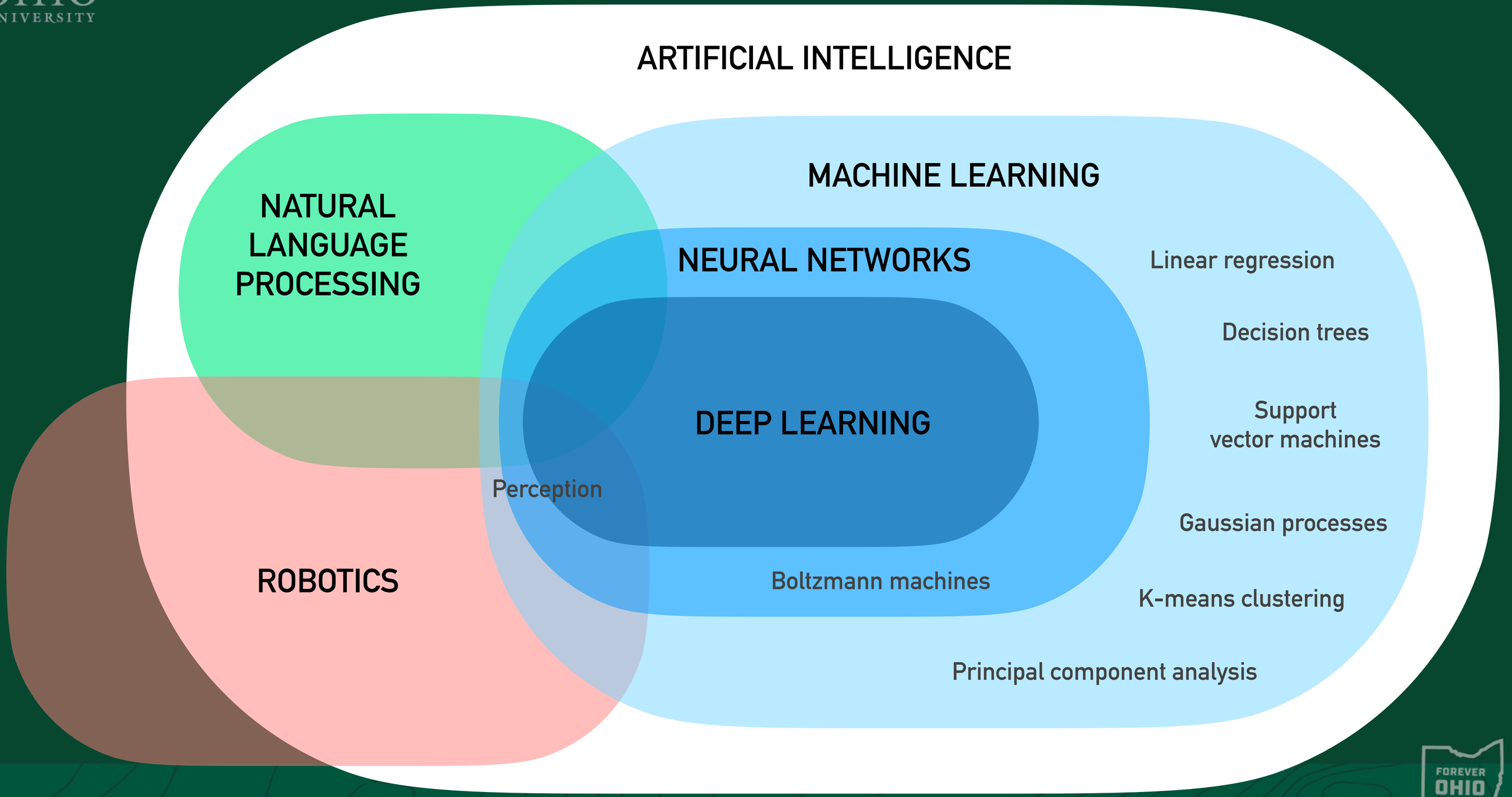


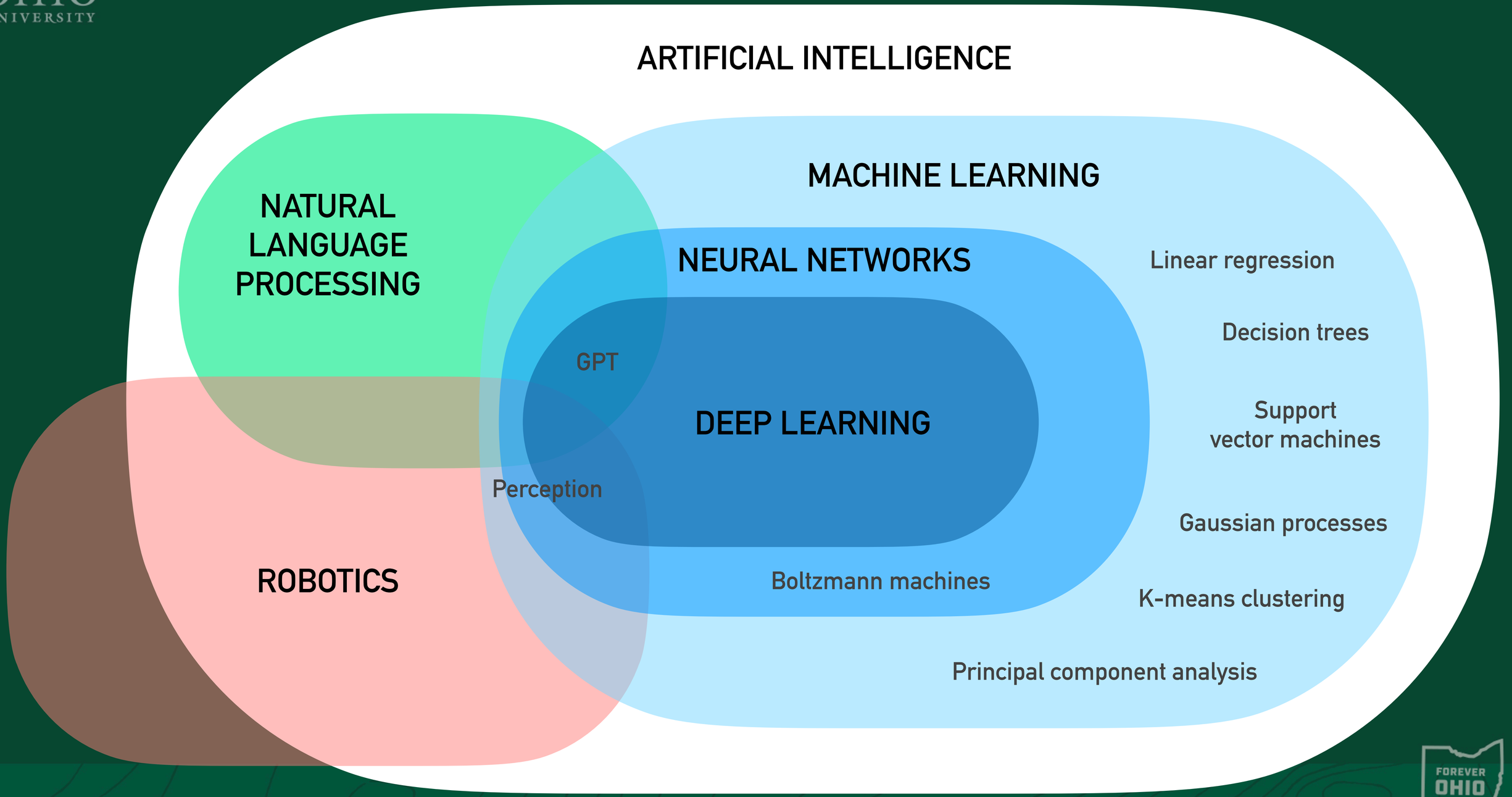
FOREVER OHIO


ARTIFICIAL INTELLIGENCE
MACHINE LEARNING
NATURAL LANGUAGE PROCESSING
NEURAL NETWORKS
DEEP LEARNING
ROBOTICS


ARTIFICIAL INTELLIGENCE
NATURAL LANGUAGE PROCESSING
MACHINE LEARNING
NEURAL NETWORKS
DEEP LEARNING
ROBOTICS
Linear regression
Decision trees
Support vector machines
Gaussian processes
K-means clustering
Principal component analysis


ARTIFICIAL INTELLIGENCE
NATURAL LANGUAGE PROCESSING
MACHINE LEARNING
NEURAL NETWORKS
DEEP LEARNING
ROBOTICS
Linear regression
Decision trees
Support vector machines
Gaussian processes
Boltzmann machines
K-means clustering
Principal component analysis


ARTIFICIAL INTELLIGENCE
NATURAL LANGUAGE PROCESSING
MACHINE LEARNING
NEURAL NETWORKS
DEEP LEARNING
Perception
ROBOTICS
Boltzmann machines
Linear regression
Decision trees
Support vector machines
Gaussian processes
K-means clustering
Principal component analysis


ARTIFICIAL INTELLIGENCE
NATURAL LANGUAGE PROCESSING
MACHINE LEARNING
NEURAL NETWORKS
GPT
DEEP LEARNING
Perception
ROBOTICS
Boltzmann machines
Linear regression
Decision trees
Support vector machines
Gaussian processes
K-means clustering
Principal component analysis

OHIO UNIVERSITY

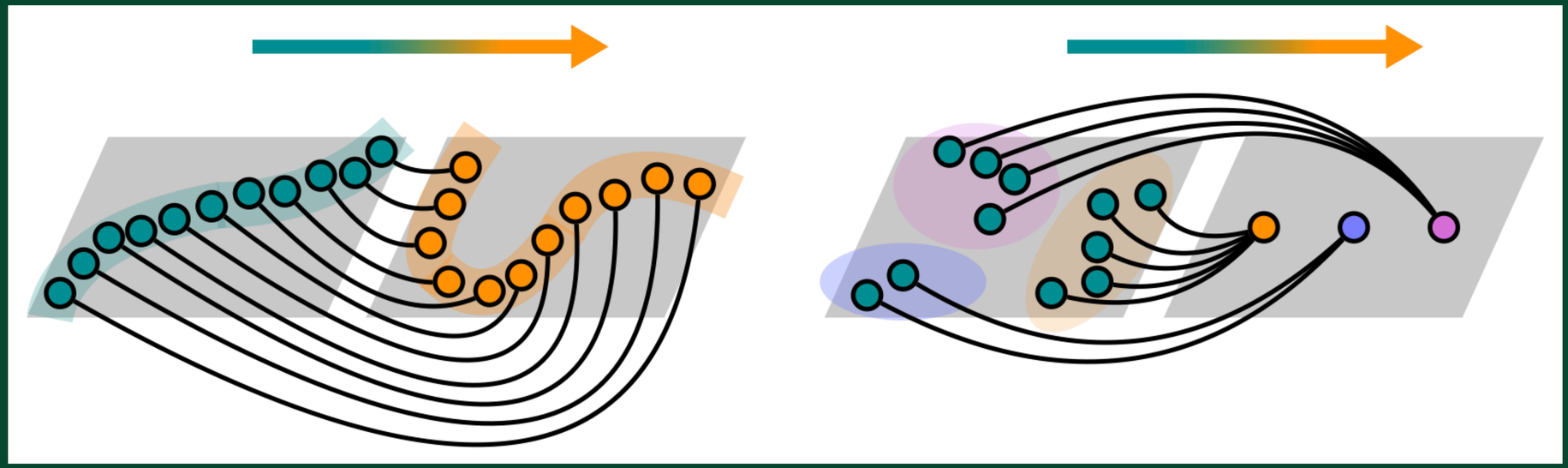
BRANCHES OF MACHINE LEARNING

- Supervised learning:
    - Map input data X to output data Y
- Unsupervised learning
    - Understand the structure of input data X
- Reinforcement learning
    - Perform tasks while interacting with a dynamic environment
    - Input data X_t depends on prior experiences $X_{t-1}, X_{t-2}, \ldots, X_0$
- These branches are not necessarily distinct! All three can be used together to complete a task.

FOREVER OHIO

SUPERVISED LEARNING

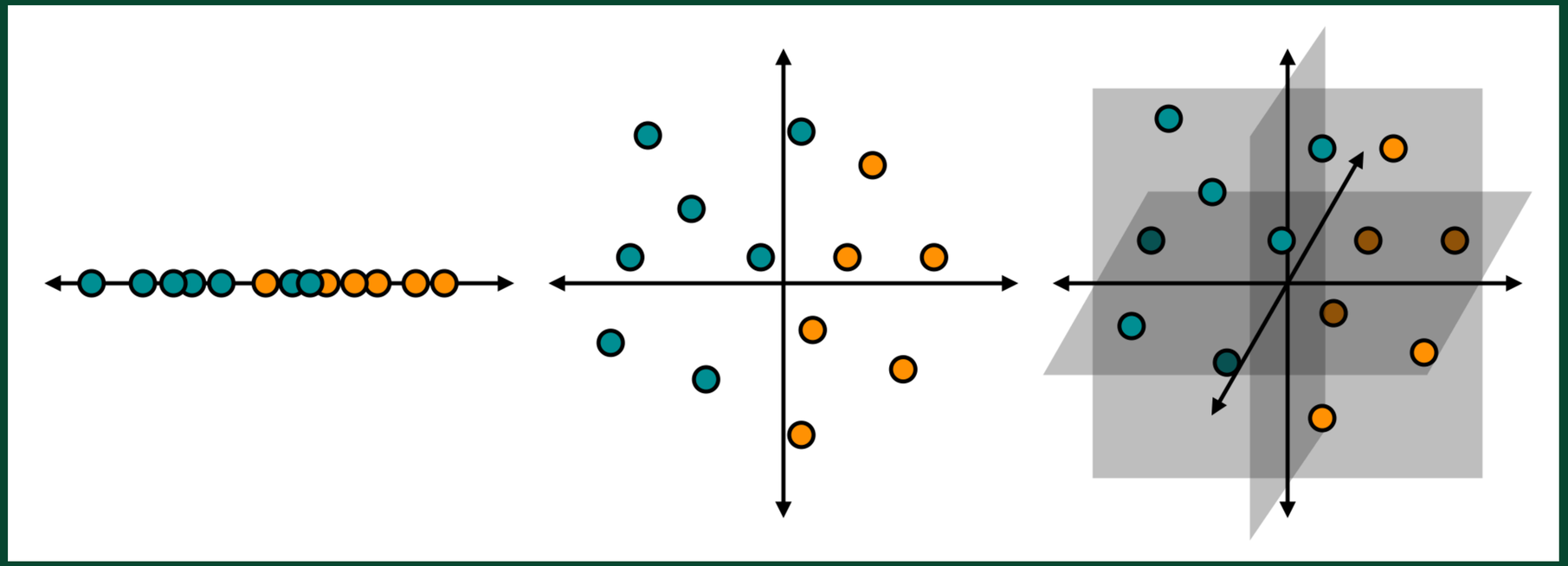
- Map input data X to output data Y
- Regression (continuous Y) and classification (discrete Y)
- Common algorithms: linear regression, decision trees, support vector machines, k-nearest neighbors, Gaussian processes, neural networks, etc.



OHIO UNIVERSITY

THE CURSE OF DIMENSIONALITY

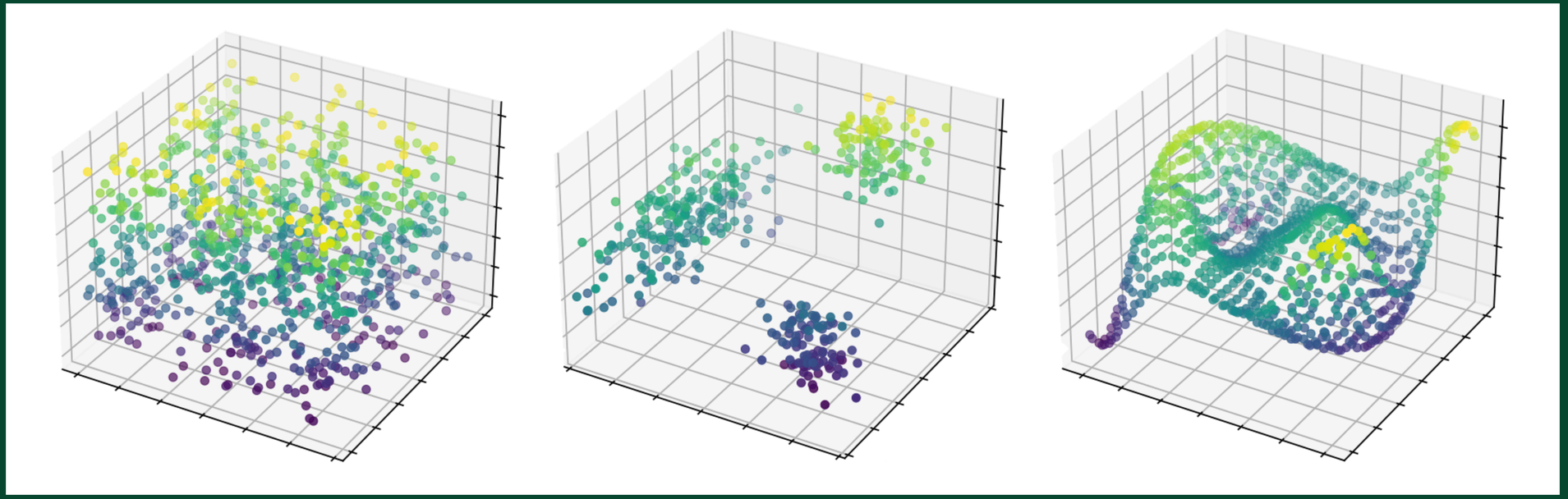
- As the dimension of the data increases, data sets become exponentially more sparse



OHIO
UNIVERSITY

THE SILVER LINING

- Real-life data sets have “structure”
- Structure = information

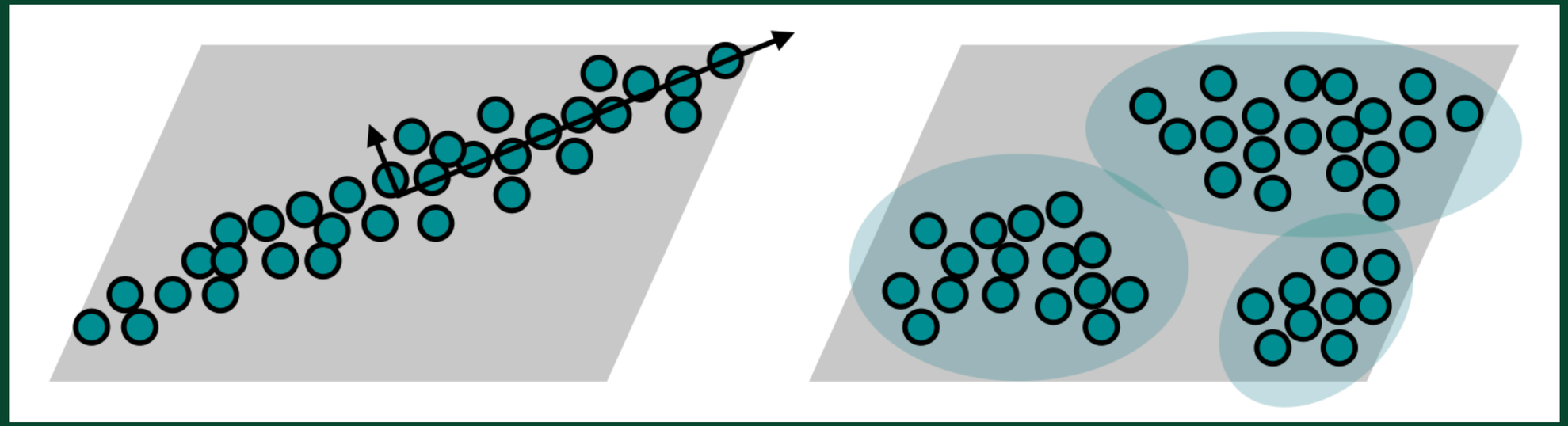


FOREVER OHIO

OHIO UNIVERSITY

UNSUPERVISED LEARNING

- Understand the structure of input data X
- Clustering, dimensionality reduction, anomaly detection, density estimation
- Common algorithms: principal component analysis, k-means clustering, autoencoders, Boltzmann machines, etc.



OHIO UNIVERSITY

REINFORCEMENT LEARNING

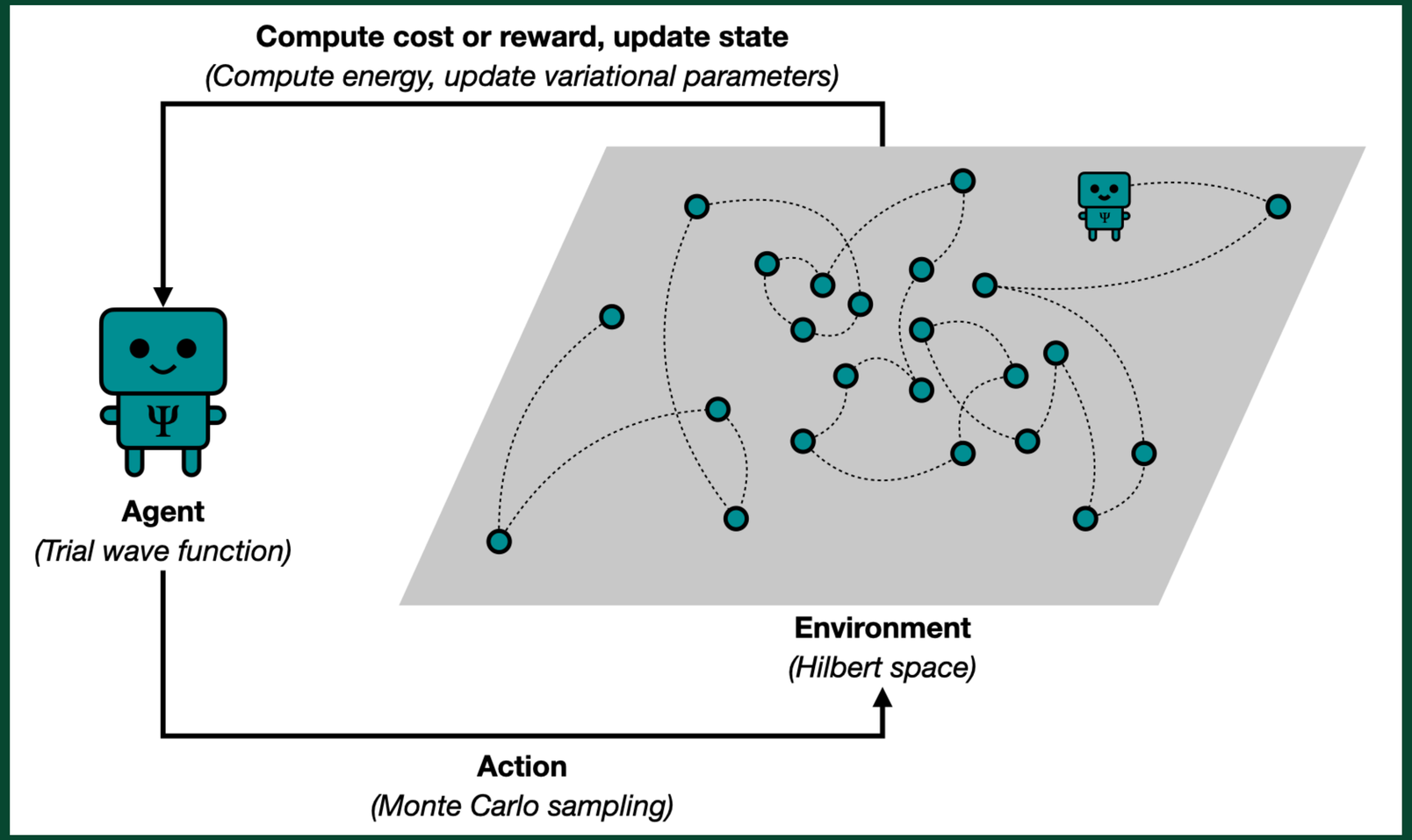
- Perform tasks while interacting with a dynamic environment
- Input data X_t depends on prior experiences X_{t-1}, X_{t-2}, $\ldots$, X_0
- Collecting new data is a part of the learning process
- Useful when it's inefficient or impossible to collect or store enough traditional training data

FOREVER OHIO

OHIO UNIVERSITY

REINFORCEMENT LEARNING

- E.g. training wave functions by minimizing the energy
- In practice, we may also use supervised and unsupervised learning to prepare the initial wave function or learn a better representation of the data



FOREVER OHIO

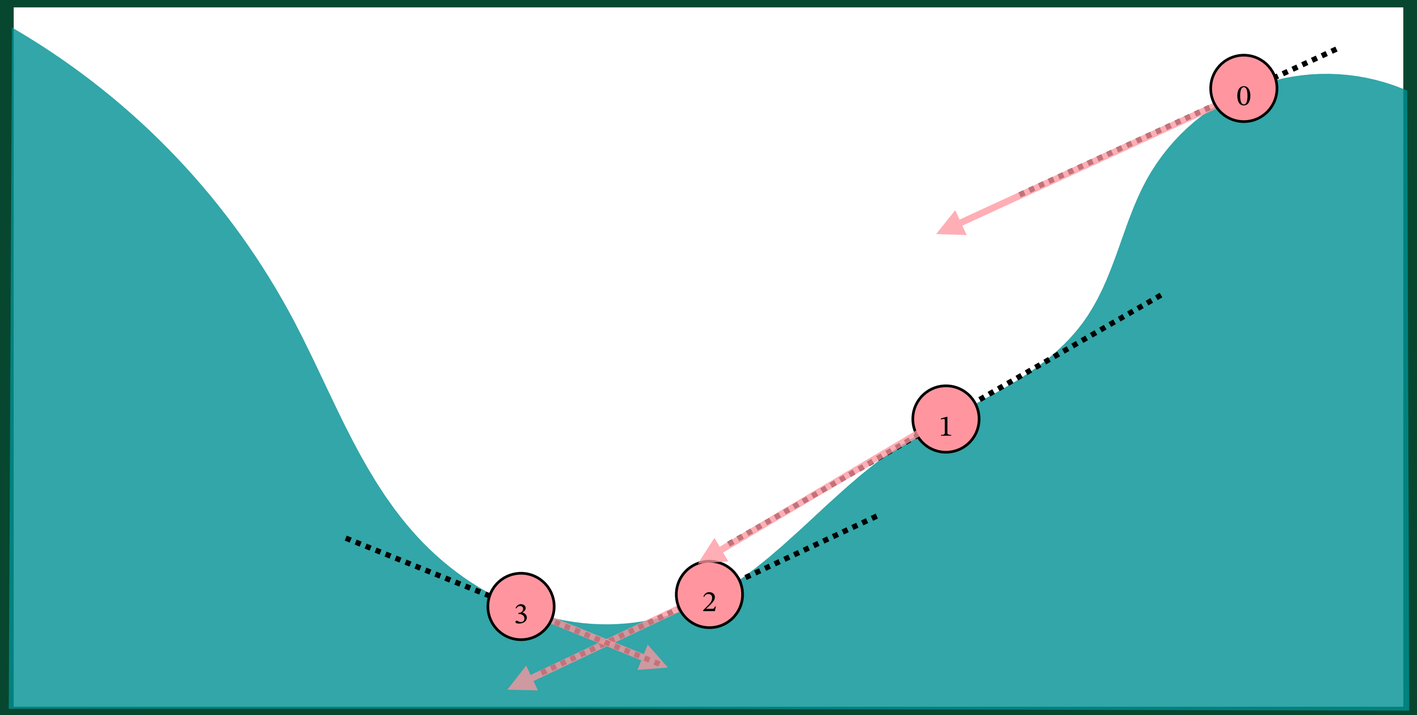
OPTIMIZATION

- Sometimes can be done analytically, e.g. closed form solutions for linear regression and Gaussian processes
- Sometimes done iteratively with special algorithms, e.g. k-means clustering
- Often times we opt for iterative gradient descent methods
    - Need an explicit function to minimize/maximize
    - Need to compute or approximate the gradient of this function

OHIO UNIVERSITY

EXAMPLE: GRADIENT DESCENT

- Go to: https://github.com/kim-jane/IntroMachineLearning/
  For an introduction on some common gradient descent methods

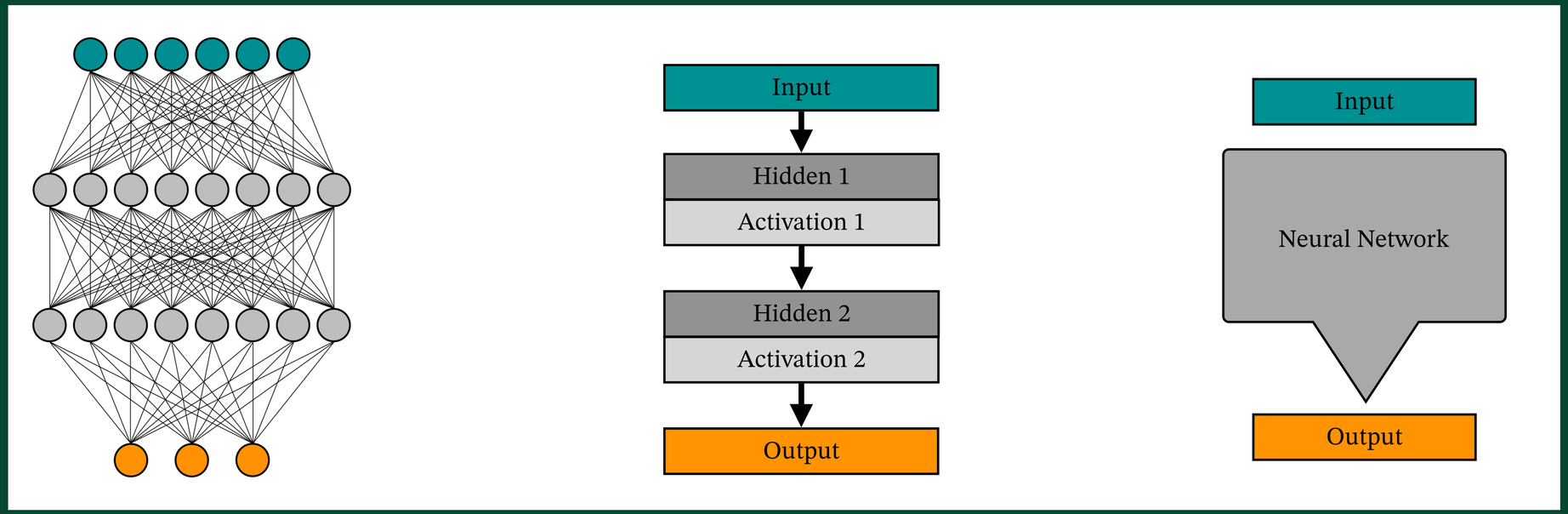


FOREVER OHIO

OHIO UNIVERSITY

NEURAL NETWORKS

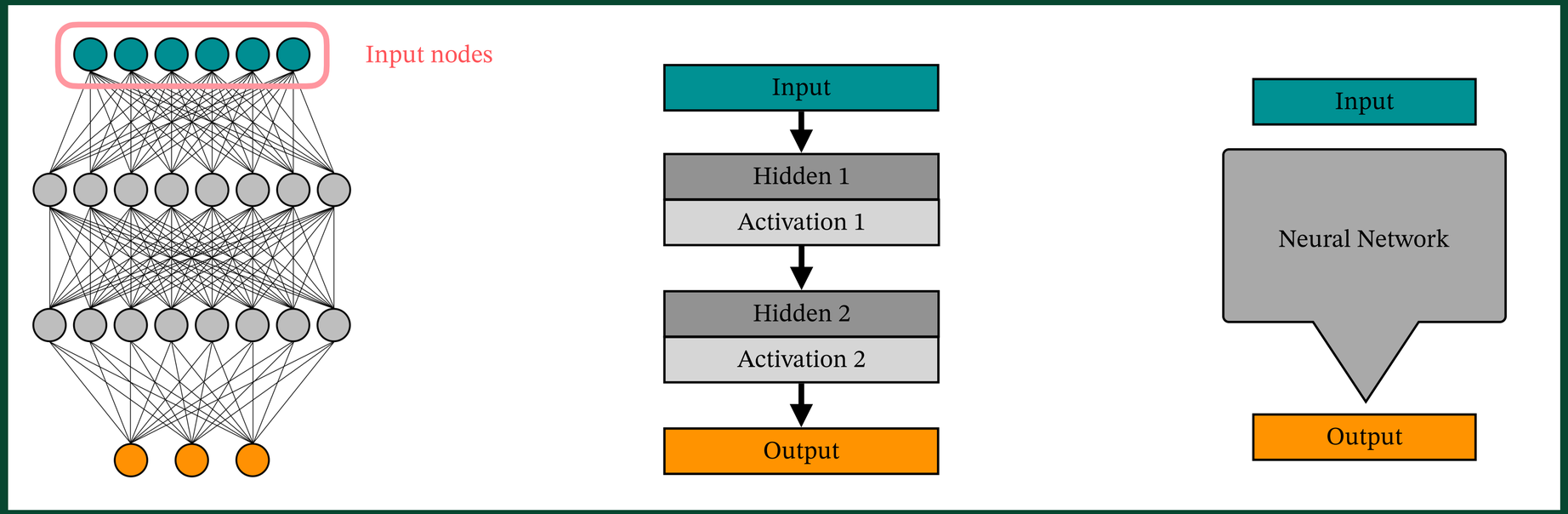
- Very flexible models inspired by the structure of biological brains
- Feedforward neural networks are “universal” function approximators (conditions apply)
- WARNING: Do not use these if you’re data set is small or if you’re task can be handled by a more interpretable model.



OHIO UNIVERSITY

NEURAL NETWORKS

- Very flexible models inspired by the structure of biological brains
- Feedforward neural networks are “universal” function approximators (conditions apply)
- WARNING: Do not use these if you’re data set is small or if you’re task can be handled by a more interpretable model.

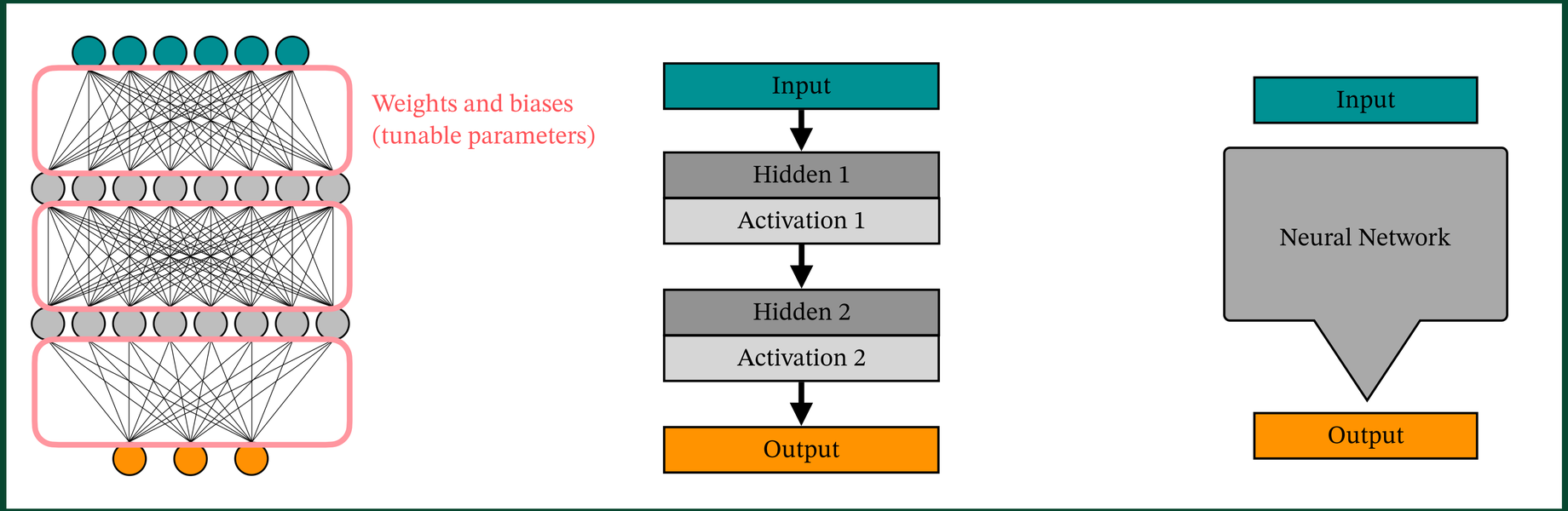


FOREVER OHIO

OHIO UNIVERSITY

NEURAL NETWORKS

- Very flexible models inspired by the structure of biological brains
- Feedforward neural networks are “universal” function approximators (conditions apply)
- WARNING: Do not use these if you’re data set is small or if you’re task can be handled by a more interpretable model.

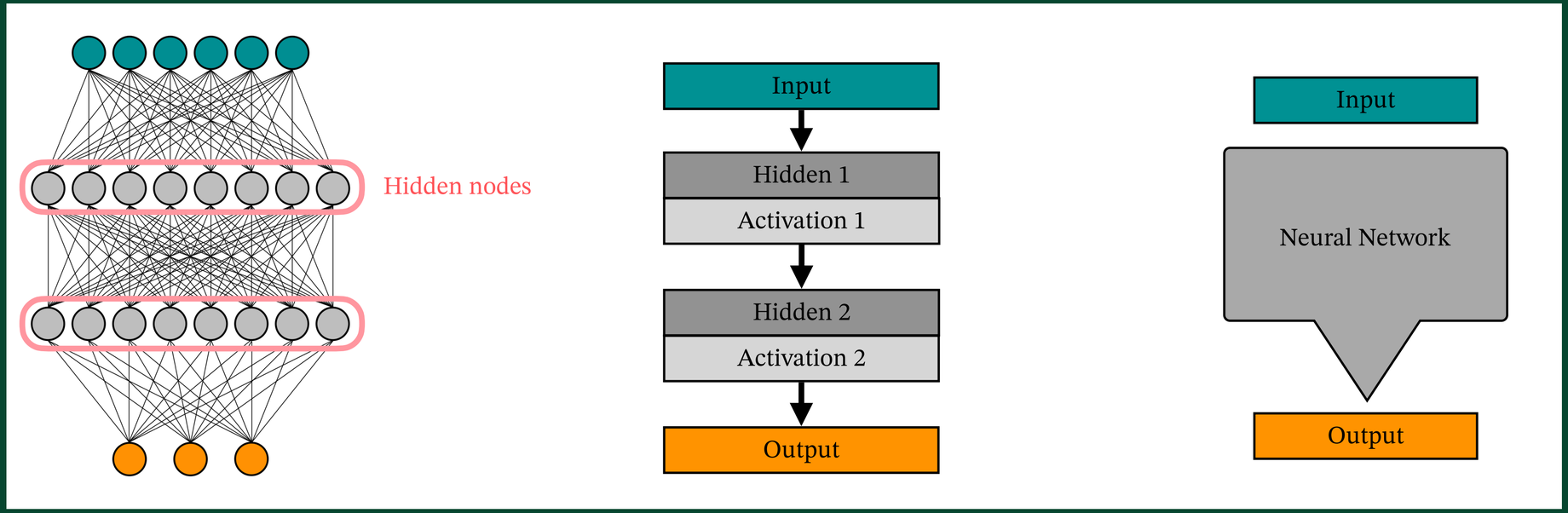


FOREVER OHIO

OHIO UNIVERSITY

NEURAL NETWORKS

- Very flexible models inspired by the structure of biological brains
- Feedforward neural networks are “universal” function approximators (conditions apply)
- WARNING: Do not use these if you’re data set is small or if you’re task can be handled by a more interpretable model.

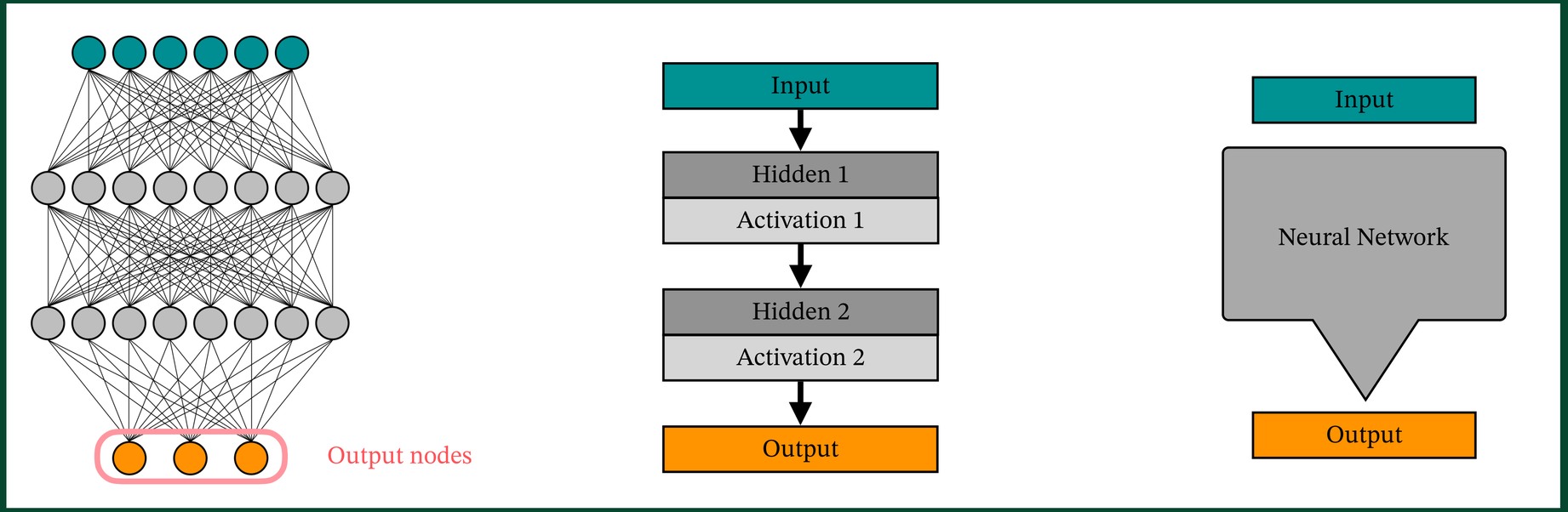


FOREVER OHIO

OHIO UNIVERSITY

NEURAL NETWORKS

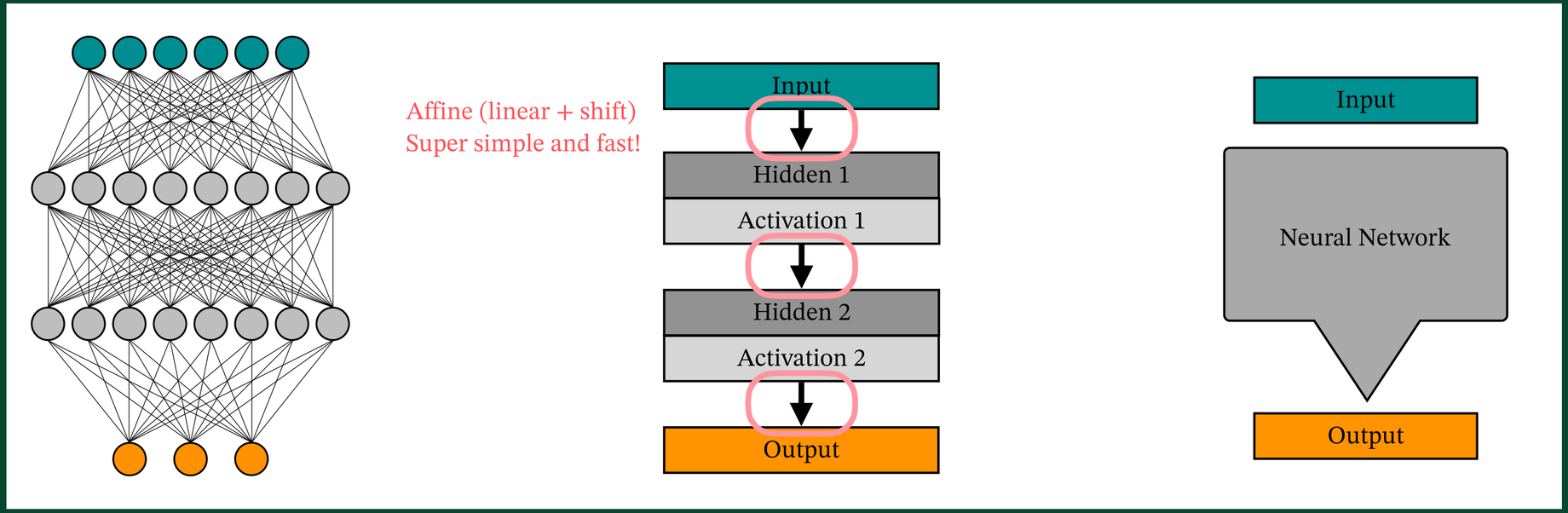
- Very flexible models inspired by the structure of biological brains
- Feedforward neural networks are “universal” function approximators (conditions apply)
- WARNING: Do not use these if you’re data set is small or if you’re task can be handled by a more interpretable model.



OHIO UNIVERSITY

NEURAL NETWORKS

- Very flexible models inspired by the structure of biological brains
- Feedforward neural networks are “universal” function approximators (conditions apply)
- WARNING: Do not use these if you’re data set is small or if you’re task can be handled by a more interpretable model.

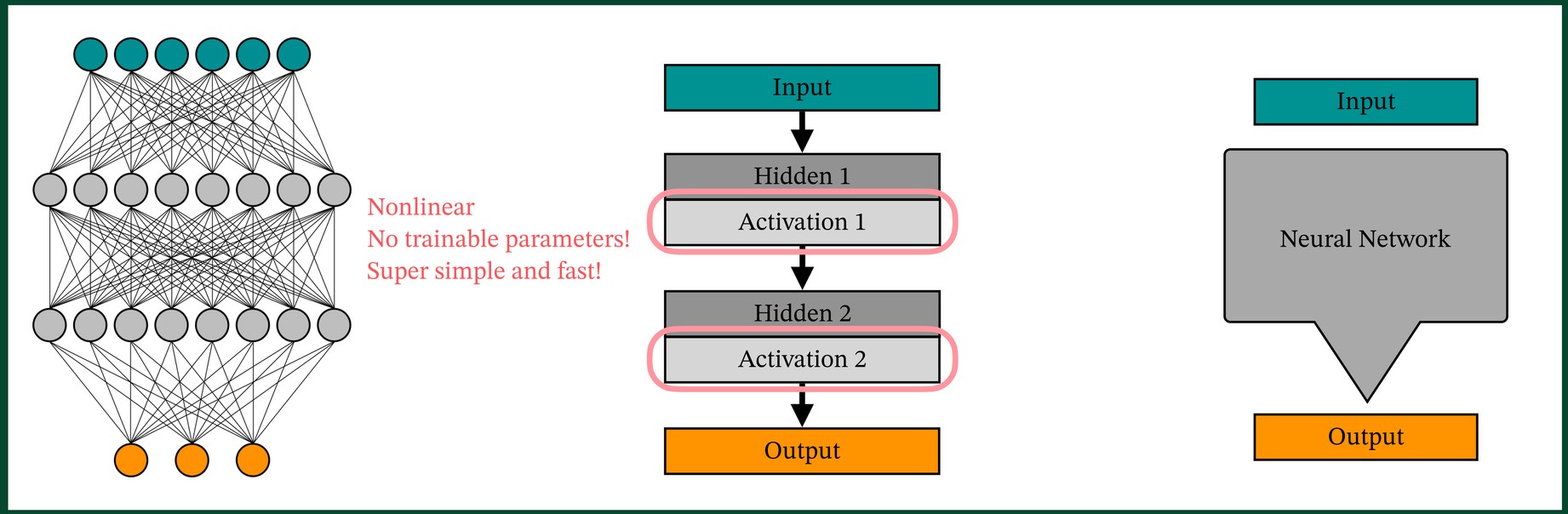


FOREVER OHIO

OHIO UNIVERSITY

NEURAL NETWORKS

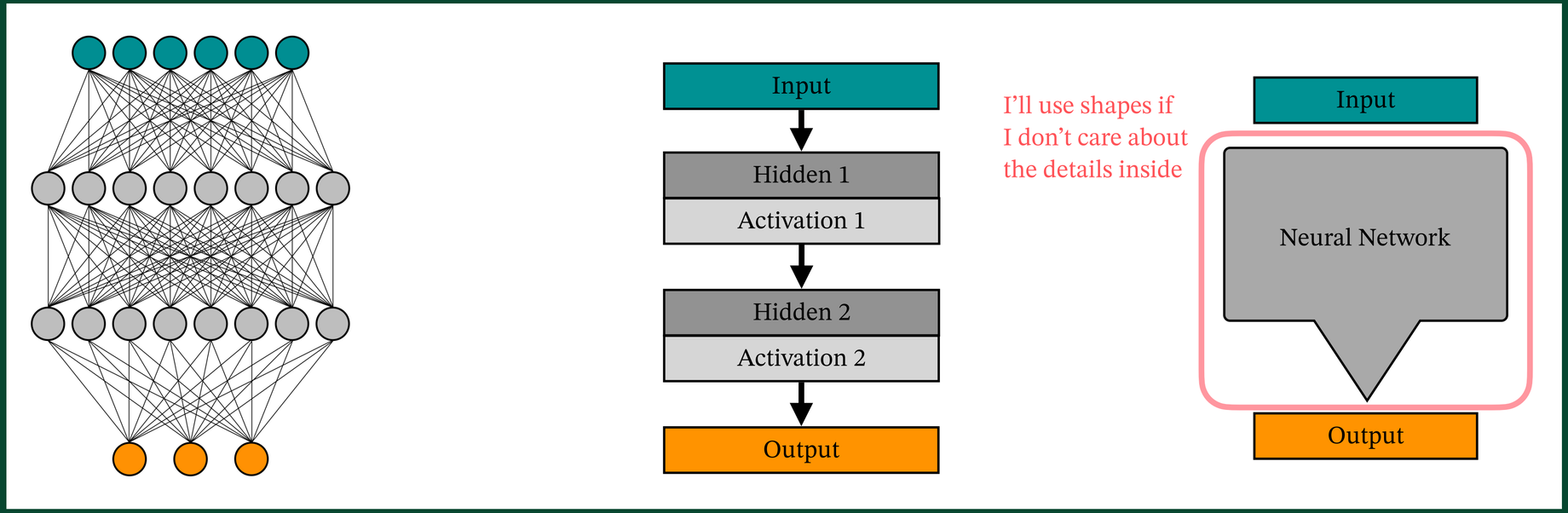
- Very flexible models inspired by the structure of biological brains
- Feedforward neural networks are “universal” function approximators (conditions apply)
- WARNING: Do not use these if you’re data set is small or if you’re task can be handled by a more interpretable model.



OHIO UNIVERSITY

NEURAL NETWORKS

- Very flexible models inspired by the structure of biological brains
- Feedforward neural networks are “universal” function approximators (conditions apply)
- WARNING: Do not use these if you’re data set is small or if you’re task can be handled by a more interpretable model.

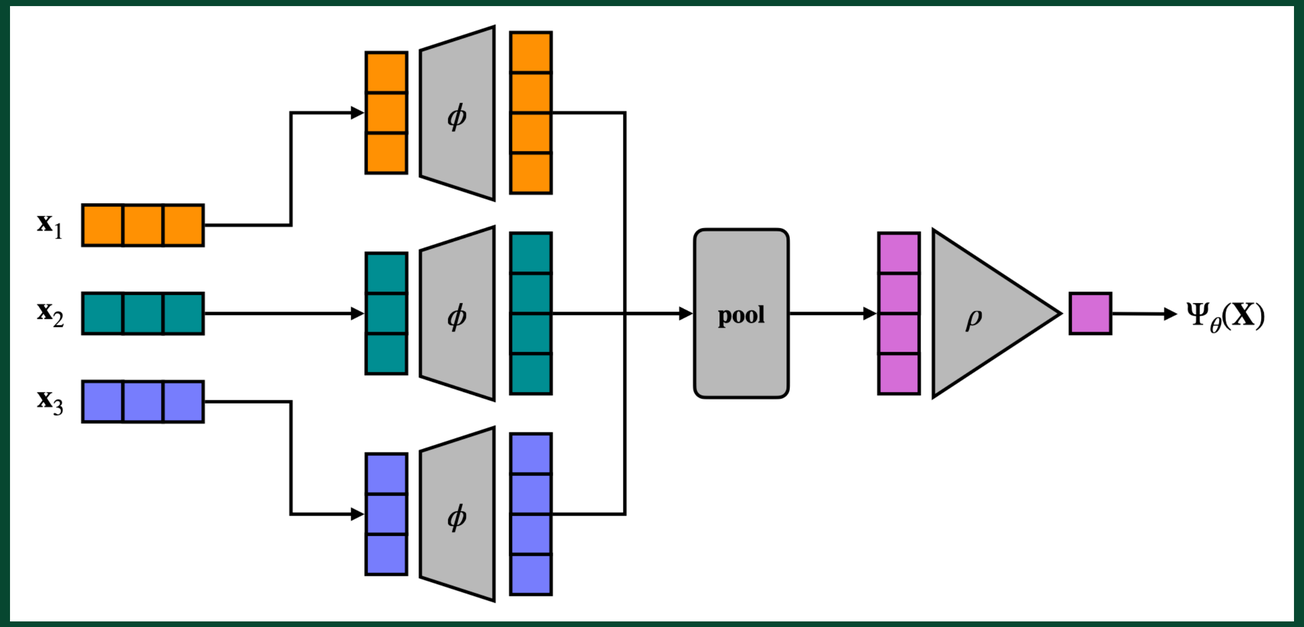


FOREVER OHIO

OHIO UNIVERSITY

SYMMETRIES

- In physics, we often need to constrain symmetries or boundary conditions
- In machine learning, algorithms that take advantage of the symmetries perform better
- E.g. convolutional neural networks assume the data has grid-like structure (great for images)
- E.g. two neural networks can be used to construct a permutation-invariant function (deep set)

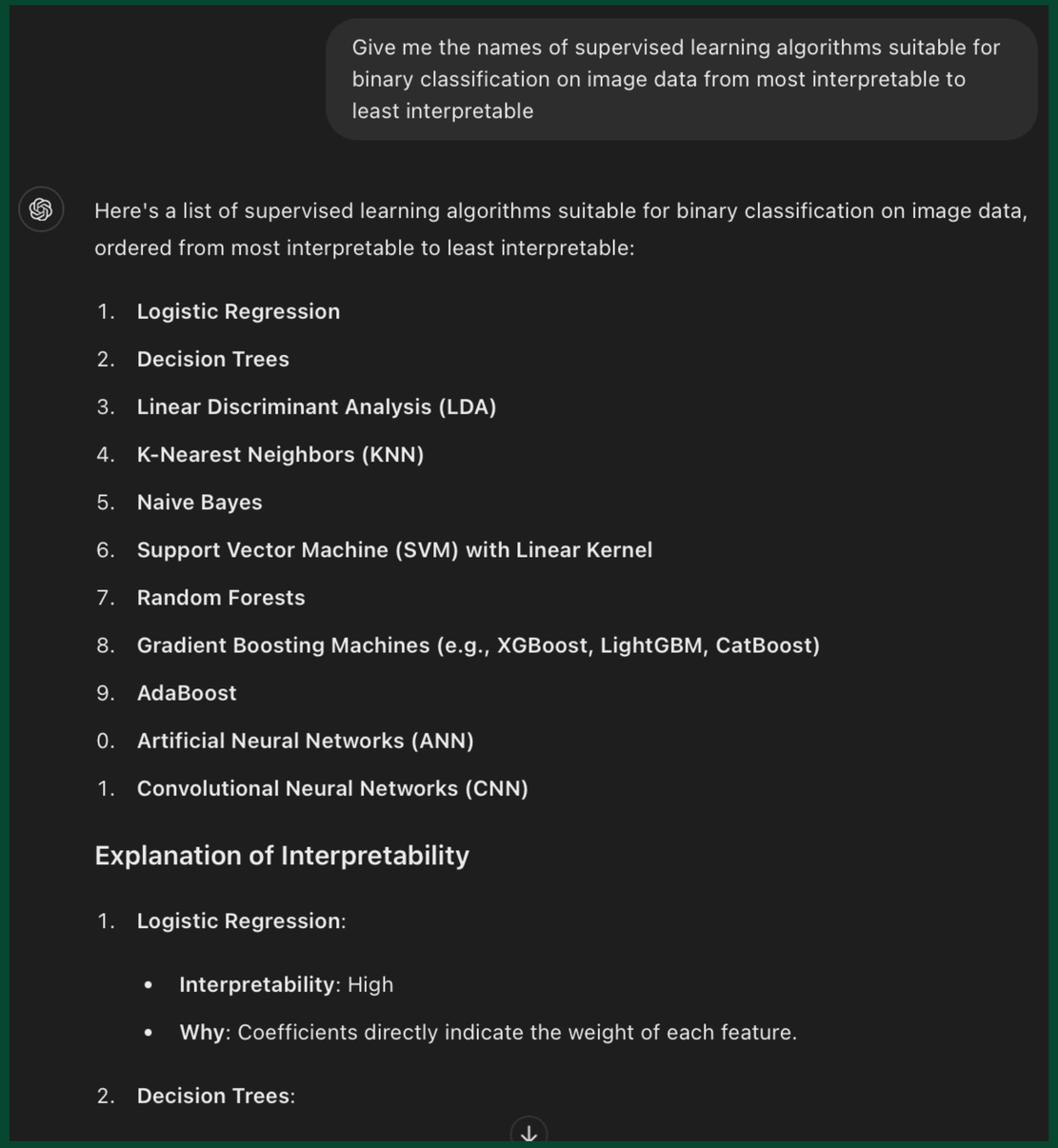


FOREVER OHIO

OHIO UNIVERSITY

APPROACHING A TASK

- Identify the task
- What kind of data do you have access to?
- Does your data have labels?
- Is it high-dimensional?
- Figure out which branch your problem naturally falls under
- Find the names of methods that are suitable for your specific problem
- For most applications, you can use the algorithms “out-of-the-box”, but understanding the math can help you get better performance



FOREVER OHIO

EXAMPLE: SELF-DRIVING CAR

- Goal: Take a passenger from point A to point B upon request, safely and efficiently.
- What are the individual problems we need to solve?

EXAMPLE: SELF-DRIVING CAR

- Goal: Take a passenger from point A to point B upon request, safely and efficiently.
- What are the individual problems we need to solve?
    - Taking requests:
        - Interpret human speech or text
        - Determine point A and point B

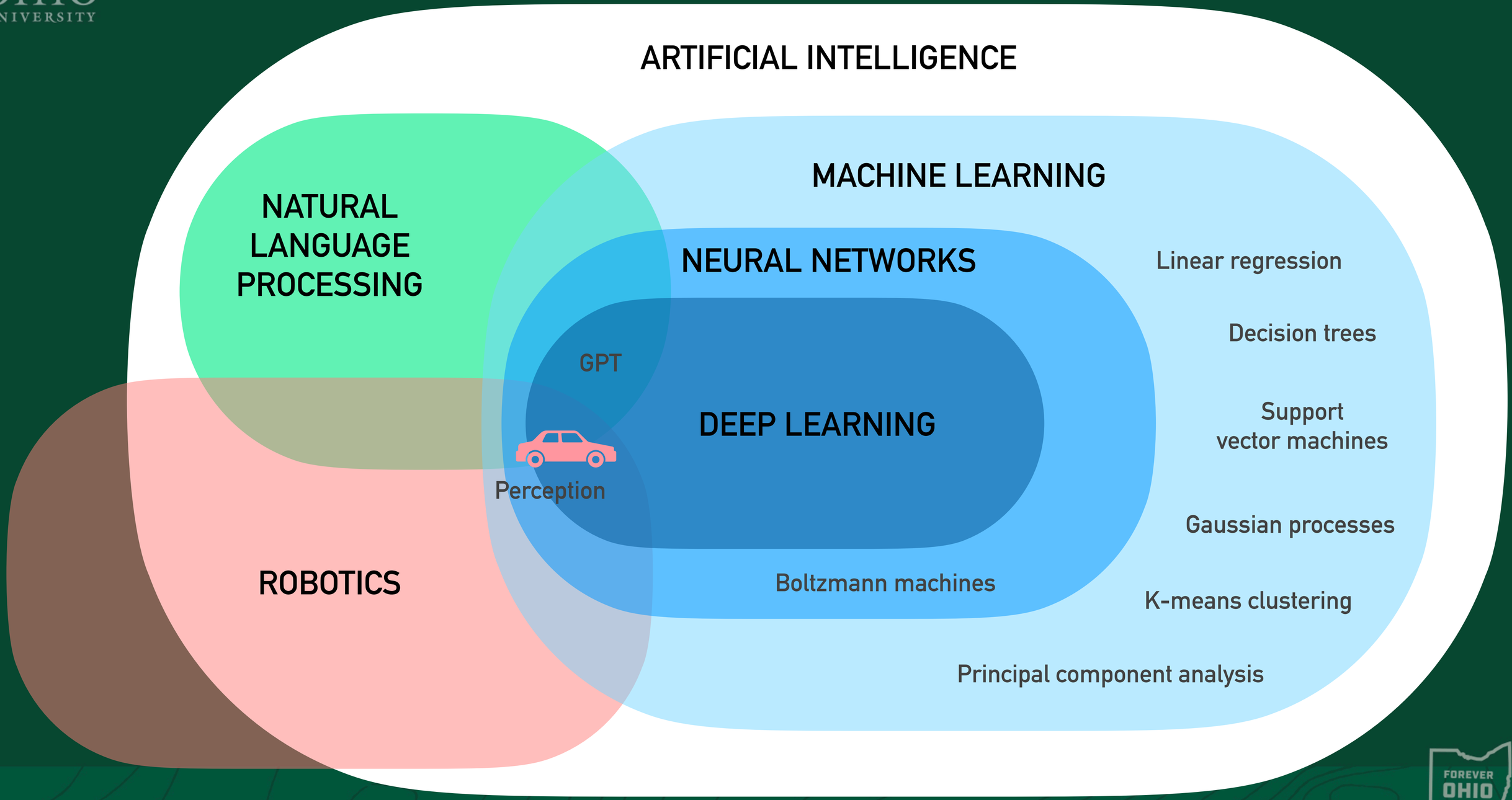
EXAMPLE: SELF-DRIVING CAR

- Goal: Take a passenger from point A to point B upon request, safely and efficiently.
- What are the individual problems we need to solve?
    - Taking requests:
        - Interpret human speech or text
        - Determine point A and point B
    - Mapping route:
        - What is the most efficient way to get from A to B?

EXAMPLE: SELF-DRIVING CAR

- Goal: Take a passenger from point A to point B upon request, safely and efficiently.
- What are the individual problems we need to solve?
    - Taking requests:
        - Interpret human speech or text
        - Determine point A and point B
    - Mapping route:
        - What is the most efficient way to get from A to B?
    - Safety / comfort:
        - Obeying the rules of the road
        - What's the best braking rate, turning radius, etc?
        - How to identify and react to surroundings
        - How to make decisions in case of emergency

OHIO UNIVERSITY



FOREVER OHIO

EXAMPLE: SELF-DRIVING CAR

- Goal: Take a passenger from point A to point B upon request, safely and efficiently.
- What are the individual problems we need to solve?
    - Taking requests:
        - Interpret human speech or text
        - Determine point A and point B
    - Mapping route:
        - What is the most efficient way to get from A to B?
    - Safety / comfort:
        - Obeying the rules of the road
        - What's the best braking rate, turning radius, etc?
        - How to identify and react to surroundings
        - How to make decisions in case of emergency

EXAMPLE: SELF-DRIVING CAR

- Goal: Take a passenger from point A to point B upon request, safely and efficiently.
- What are the individual problems we need to solve?
    - Taking requests:
        - Interpret human speech or text
        - Determine point A and point B
    - Mapping route:
        - What is the most efficient way to get from A to B?
    - Safety / comfort:
        - Obeying the rules of the road
        - What's the best braking rate, turning radius, etc?
        - How to identify and react to surroundings
        - How to make decisions in case of emergency

EXAMPLE: SELF-DRIVING CAR

- How to identify surroundings with supervised learning
  - The car is equipped with many cameras, each producing many high-resolution images every second
  - Goal: detect whether certain objects exist within the input image

EXAMPLE: SELF-DRIVING CAR

- How to identify surroundings with supervised learning
  - The car is equipped with many cameras, each producing many high-resolution images every second
  - Goal: detect whether certain objects exist within the input image
  - Static objects: lane markings, road signs, traffic lights, etc.
  - Dynamic objects: other vehicles, pedestrians, cyclists, animals, etc.

EXAMPLE: SELF-DRIVING CAR

- How to identify surroundings with supervised learning
  - The car is equipped with many cameras, each producing many high-resolution images every second
  - Goal: detect whether certain objects exist within the input image
  - Static objects: lane markings, road signs, traffic lights, etc.
  - Dynamic objects: other vehicles, pedestrians, cyclists, animals, etc.
  - Assign labels to each of the categories of objects
  - Let your model guess a label
  - Quantitatively measure the similarity between your model's guess and the true label
  - Slightly adjust your model to get a better similarity next time

EXAMPLE: SELF-DRIVING CAR

- Just consider these objects for now:
  - Lane markings
  - Stop signs
  - Other vehicles
- This is a multi-label classification problem
- For supervised learning, we need a human to label the images
- Suppose the human labels this image $y_i = (1, 0, 1)$

EXAMPLE: SELF-DRIVING CAR

- Just consider these objects for now:
  - Lane markings
  - Stop signs
  - Other vehicles
- This is a multi-label classification problem
- For supervised learning, we need a human to label the images
- Suppose the human labels this image $y_i = (1, 0, 1)$

Yes, there are lane markings

EXAMPLE: SELF-DRIVING CAR

- Just consider these objects for now:
  - Lane markings
  - Stop signs
  - Other vehicles
- This is a multi-label classification problem
- For supervised learning, we need a human to label the images
- Suppose the human labels this image $y_i = (1, 0, 1)$

Yes, there are lane markings

No, there are no stop signs

EXAMPLE: SELF-DRIVING CAR

- Just consider these objects for now:
  - Lane markings
  - Stop signs
  - Other vehicles
- This is a multi-label classification problem
- For supervised learning, we need a human to label the images
- Suppose the human labels this image $y_i = (1, 0, 1)$

Yes, there are lane markings

No, there are no stop signs

Yes, there are other vehicles

EXAMPLE: SELF-DRIVING CAR

- Just consider these objects for now:
  - Lane markings
  - Stop signs
  - Other vehicles
- This is a multi-label classification problem
- For supervised learning, we need a human to label the images
- Suppose the human labels this image $y_i = (1, 0, 1)$
- Suppose the output of my model is $\hat{y}_i = (0.9, 0.1, 0.5)$

EXAMPLE: SELF-DRIVING CAR

- Just consider these objects for now:
  - Lane markings
  - Stop signs
  - Other vehicles
- This is a multi-label classification problem
- For supervised learning, we need a human to label the images
- Suppose the human labels this image $y_i = (1, 0, 1)$
- Suppose the output of my model is $\hat{y}_i = (0.9, 0.1, 0.5)$

Pretty sure there are lane markings

EXAMPLE: SELF-DRIVING CAR

- Just consider these objects for now:
  - Lane markings
  - Stop signs
  - Other vehicles
- This is a multi-label classification problem
- For supervised learning, we need a human to label the images
- Suppose the human labels this image $y_i = (1, 0, 1)$
- Suppose the output of my model is $\hat{y}_i = (0.9, 0.1, 0.5)$

Pretty sure there are lane markings

Probably no stop signs

EXAMPLE: SELF-DRIVING CAR

- Just consider these objects for now:
  - Lane markings
  - Stop signs
  - Other vehicles
- This is a multi-label classification problem
- For supervised learning, we need a human to label the images
- Suppose the human labels this image $y_i = (1, 0, 1)$
- Suppose the output of my model is $\hat{y}_i = (0.9, 0.1, 0.5)$

Pretty sure there are lane markings

Probably no stop signs

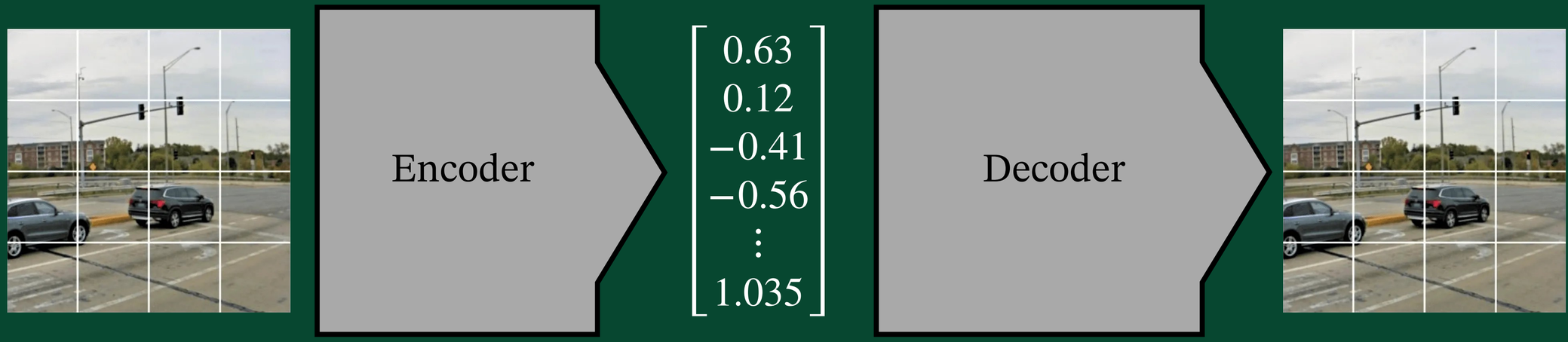
Not sure if there are other vehicles here

EXAMPLE: SELF-DRIVING CAR

- How to identify surroundings with supervised learning
  - Measure the similarity between the true label $y_i = (1,\ 0,\ 1)$ and the model's label $\hat{y}_i = (0.9,\ 0.1,\ 0.5)$
  - Binary cross-entropy is a good choice of loss function for multi-label classification
  - If $\hat{y}_i$ is close to y_i , the cross-entropy is small and positive
  - Else, the cross-entropy is large and positive
  - Want to minimize the cross-entropy

EXAMPLE: SELF-DRIVING CAR

- How to react to compress images with unsupervised learning
  - There are too many images to handle, most of the info is junk anyways
  - How can we compress the image into a vector that retains all or most of the information?
  - Try autoencoders: map your image to a smaller space, then try to reproduce the original image.
  - After training, the output of your encoder can be used as input for your supervised learning model.



EXAMPLE: SELF-DRIVING CAR

- How to react to surroundings with reinforcement learning
  - Suppose our model has been pretrained to identify important objects
  - Now we want to extend our model to make decisions based on the surroundings
  - Construct a cost function that rewards desired behavior and penalizes unwanted behavior
  - Supervised pretraining helps prepare the model for harder tasks like this

Encoder

$$\begin{bmatrix} 0.63 \\ 0.12 \\ -0.41 \\ -0.56 \\ \vdots \\ 1.035 \end{bmatrix}$$

Object Recognizer

$$\begin{bmatrix} 0.99 \\ 0.01 \\ 0.98 \end{bmatrix}$$

Decision Maker

$$\begin{bmatrix} a_L \\ a_T \end{bmatrix}$$

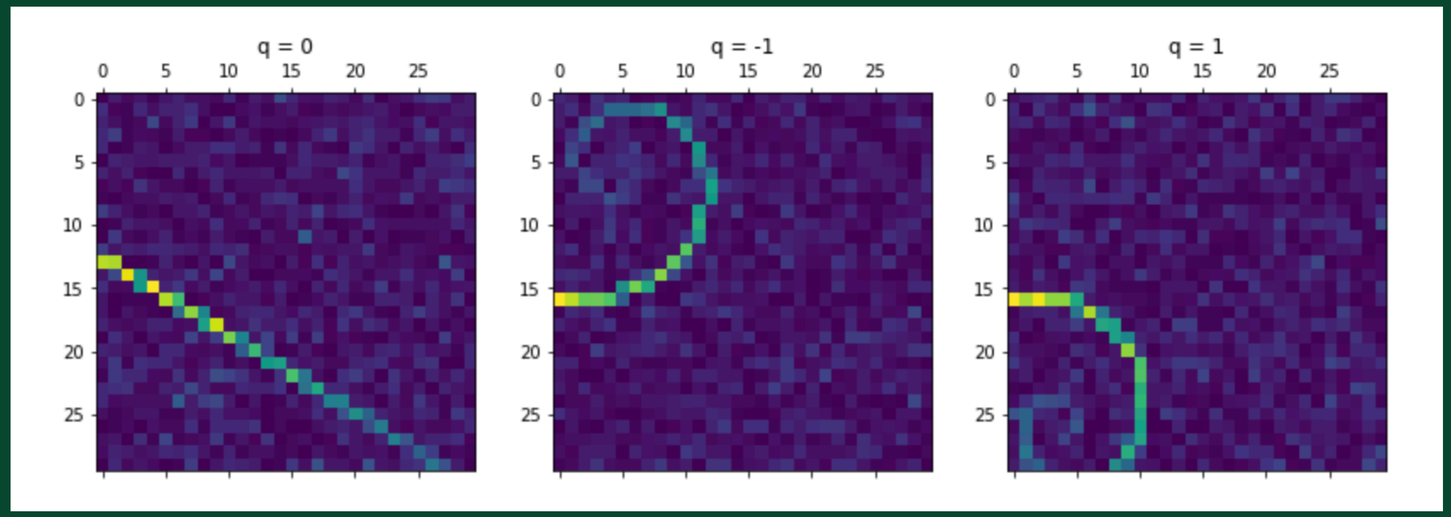
Longitudinal acceleration (forward/backward)

Transverse acceleration (left/right)

OHIO UNIVERSITY

EXAMPLE: CLASSIFYING CHARGED PARTICLES

- Go to: https://github.com/kim-jane/IntroMachineLearning/
- 30 pixel × 30 pixel images of particle tracks
- Pixel values are proportional to energy deposited
- Lots of background noise!
- Goal: classify images of tracks to the charges of particles



FOREVER OHIO

OHIO
UNIVERSITY

THANK YOU!

FOREVER
OHIO