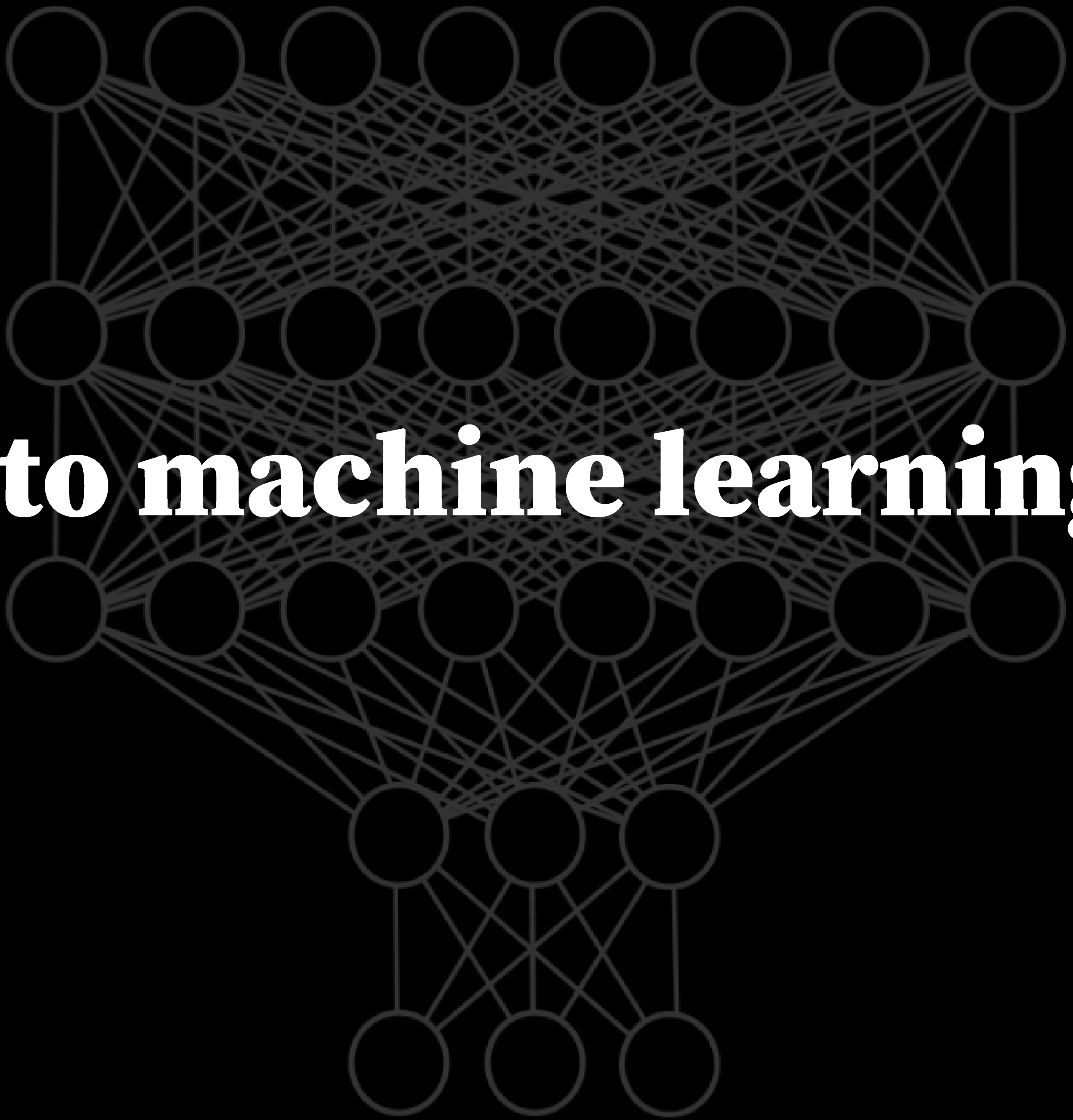
Introduction to machine learning



Peter Kvam

2026 SBI Summer School

The Ohio State University

July 27, 2026

Agenda

- Machine learning fundamentals
  - What are AI, ML, neural networks, and deep learning?
  - Why are they so successful at a variety of tasks?
- Applications and special additions
  - Images / videos, text, sequential data
  - Problems with no correct solution (unsupervised learning)
  - Generative adversarial networks
- Potential uses
  - Model fitting, comparison, discovery
  - Simulation studies of learning, evolution

Agenda

- Machine learning fundamentals
  - What are AI, ML, neural networks, and deep learning?
  - Why are they so successful at a variety of tasks?
- Applications and special additions
  - Images / videos, text, sequential data
  - Problems with no correct solution (unsupervised learning)
  - Generative adversarial networks
- Potential uses
  - Model fitting, comparison, discovery
  - Simulation studies of learning, evolution

Defining AI & ML

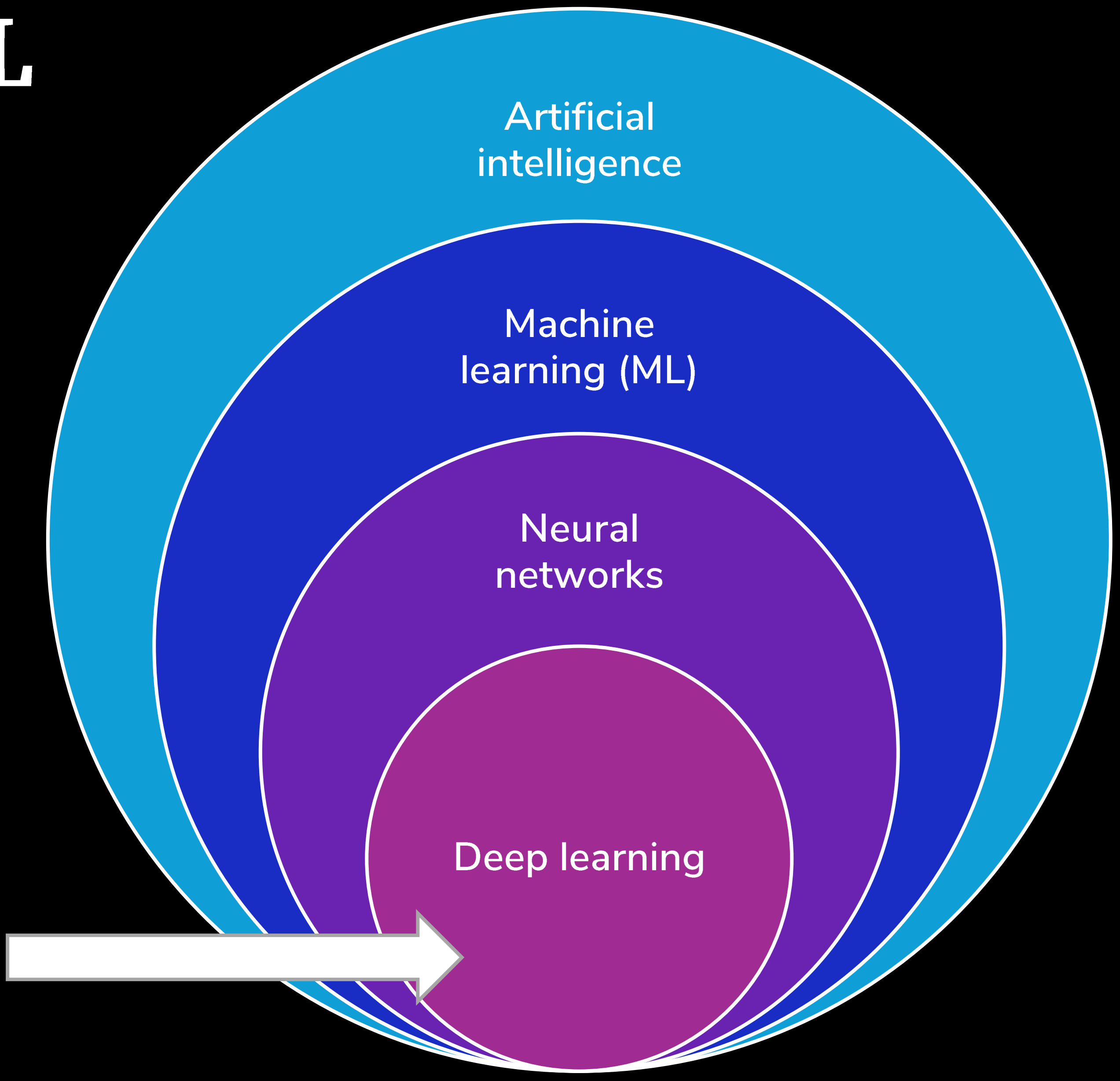
AI: machines that can mimic (some) human tasks or behaviors

ML: A special type of AI where a computer learns to do something outside its direct programming

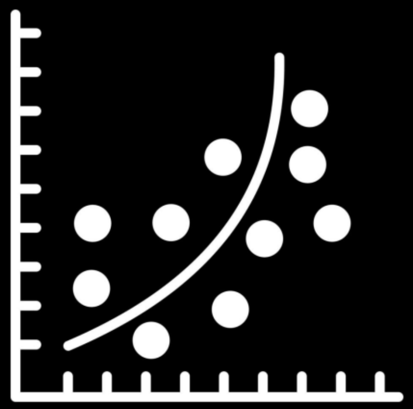
Neural networks: A special type of ML using brain-inspired structures of "neurons" / nodes and connections

Deep learning: Neural networks with multiple layers of nodes

Most applications we'll cover fall down here

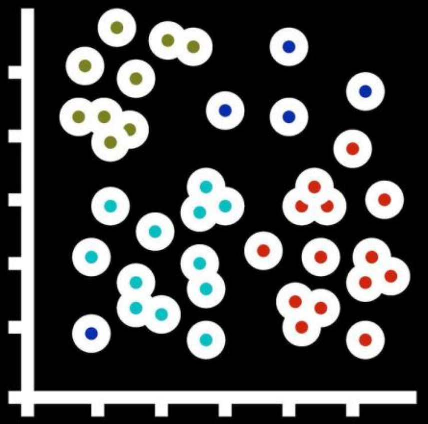


Flavors of machine learning



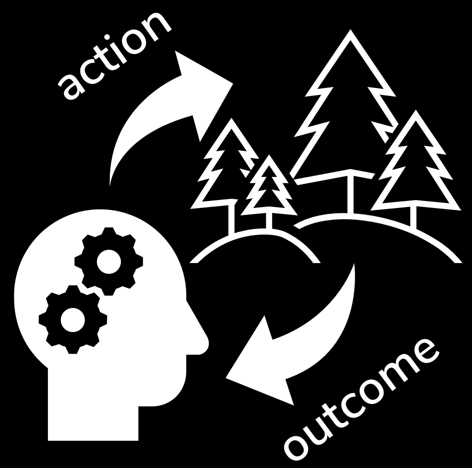
Prediction
Supervised learning

An algorithm approximates the relationship between variables based on known pairs of values



Discovery
Unsupervised learning

An algorithm creates a structure to organize data based on their quantitative attributes



Exploration
Reinforcement learning

An agent interacts with its environment to determine action-response mappings from experience

Regression

- Regression uses inputs (regressors / predictors, b) to predict some output (outcomes, **c**) using weights (a) and sums:

$$c = a_1 * b_1 + a_2 * b_2 + a_3 * b_3$$

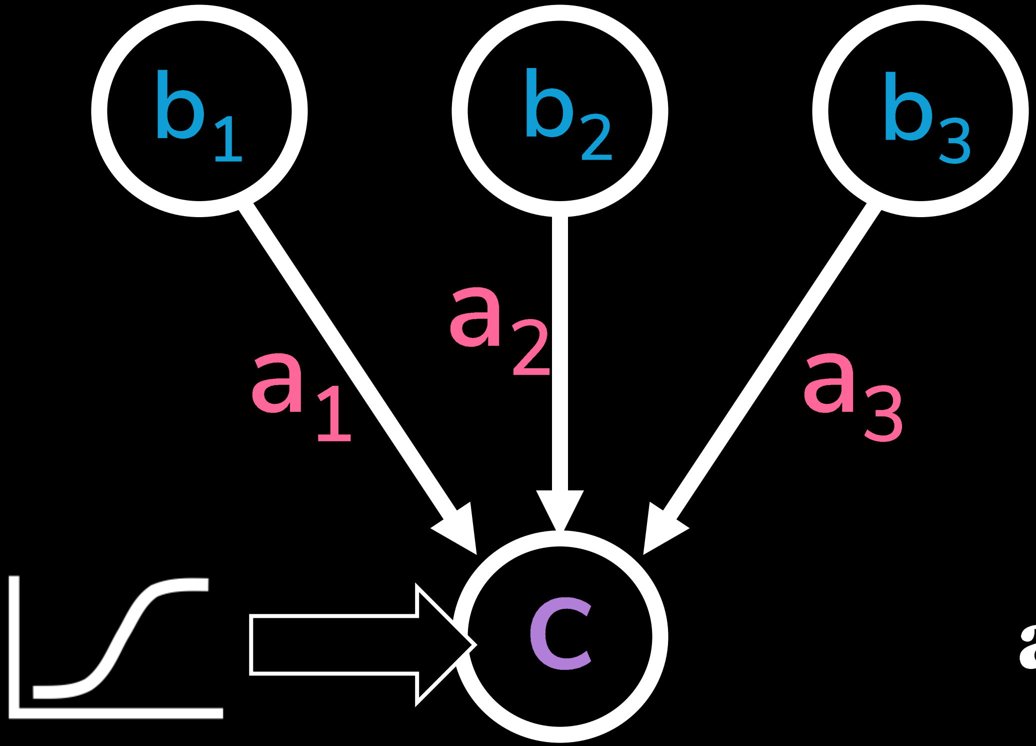
- In matrix form, it might look like this

$$C = A * B \quad A = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Regression

$$c = a_1 * b_1 + a_2 * b_2 + a_3 * b_3$$

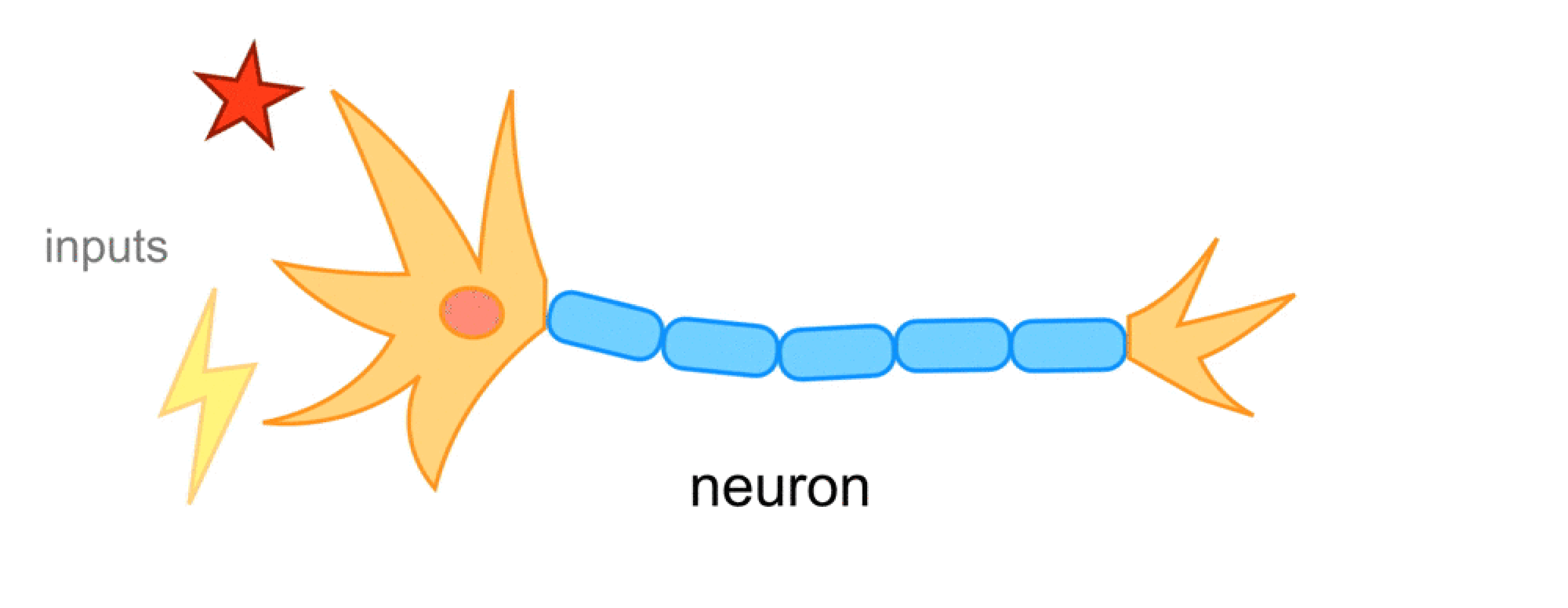
- Put visually, regression might look something like this:



In logistic regression, you might transform this outcome variable

This is now a (n artificial) neural network!

Inspired by neural systems



But: they use a very simplistic picture, and now ANNs are diverging quite sharply from real anatomy!

Scaling it up

- Start by adding another outcome – maybe we're interested in two things!
- We have our predictors **b**, weights **a**, and outcomes **c**
    - Now have two indices because we have two outcomes
    - Sorry for flipping a & b!
- Again can be represented by equation, matrix, or network

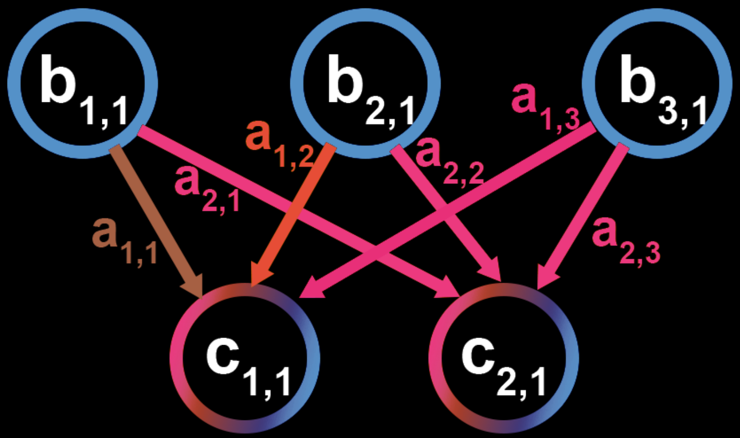
Equation form

$$c_{1,1} = a_{1,1} * b_{1,1} + a_{1,2} * b_{2,1} + a_{1,3} * b_{3,1}$$
$$c_{2,1} = a_{2,1} * b_{1,1} + a_{2,2} * b_{2,1} + a_{2,3} * b_{3,1}$$

Matrix form

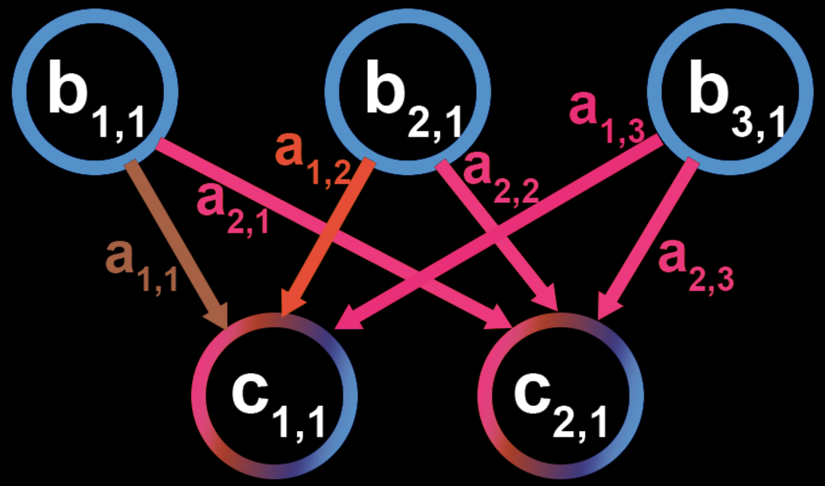
$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix} \begin{bmatrix} b_{1,1} \\ b_{2,1} \\ b_{3,1} \end{bmatrix} = \begin{bmatrix} c_{1,1} \\ c_{2,1} \end{bmatrix}$$

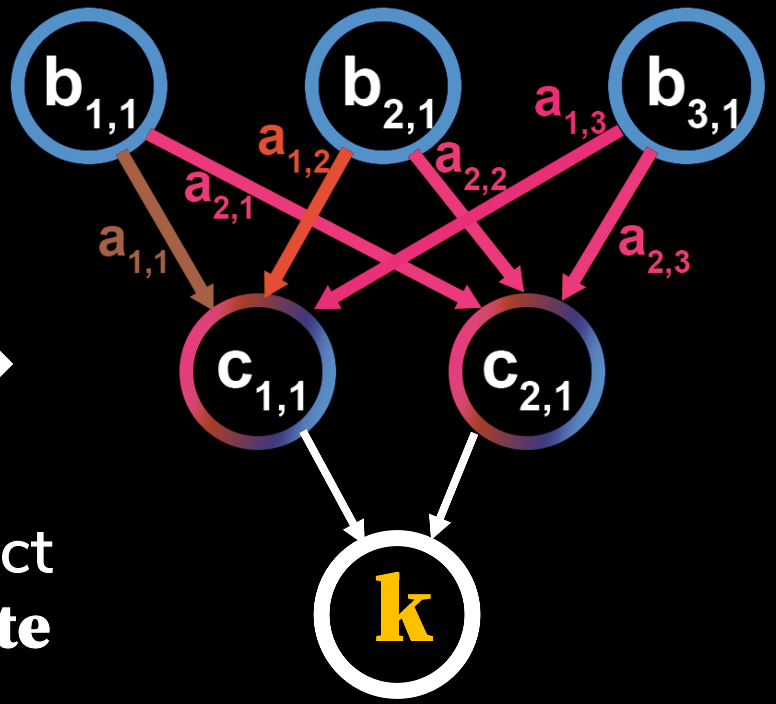
Network form



Information in the network

- The nodes c1 and c2 contain some (weighted, summed) **information** about the inputs
  - Can we use these to better predict another outcome **k**?





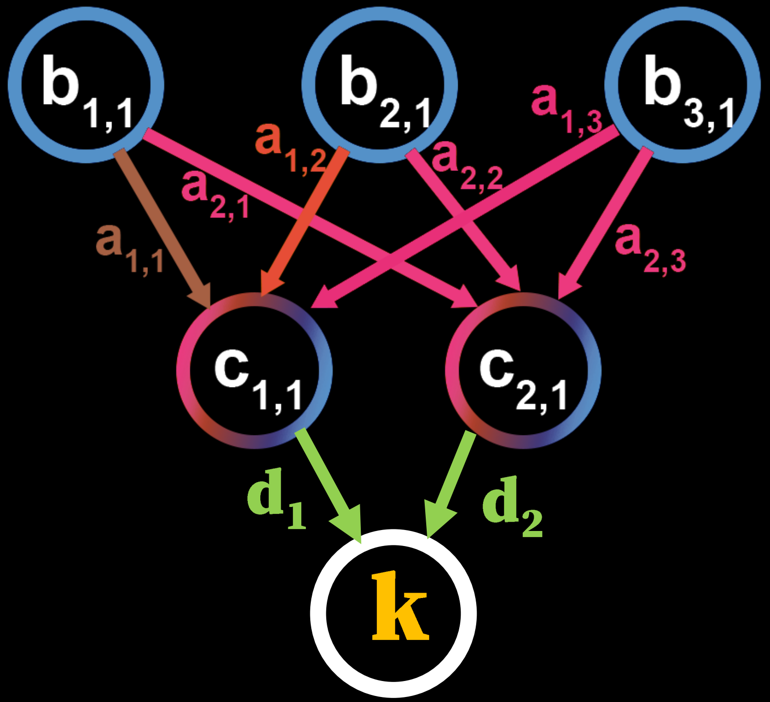
$c_{1,1}$ and $c_{2,1}$ act like **composite scores**

Problem!

- The outcome k can just as easily be predicted by the initial inputs/ predictors (**a**)
- In matrix form:

$$\left.\begin{array}{l} C = A*B \\ K = D*C \end{array}\right\} K = \underbrace{D*A}*B$$

$$K = \mathbf{W}*B$$

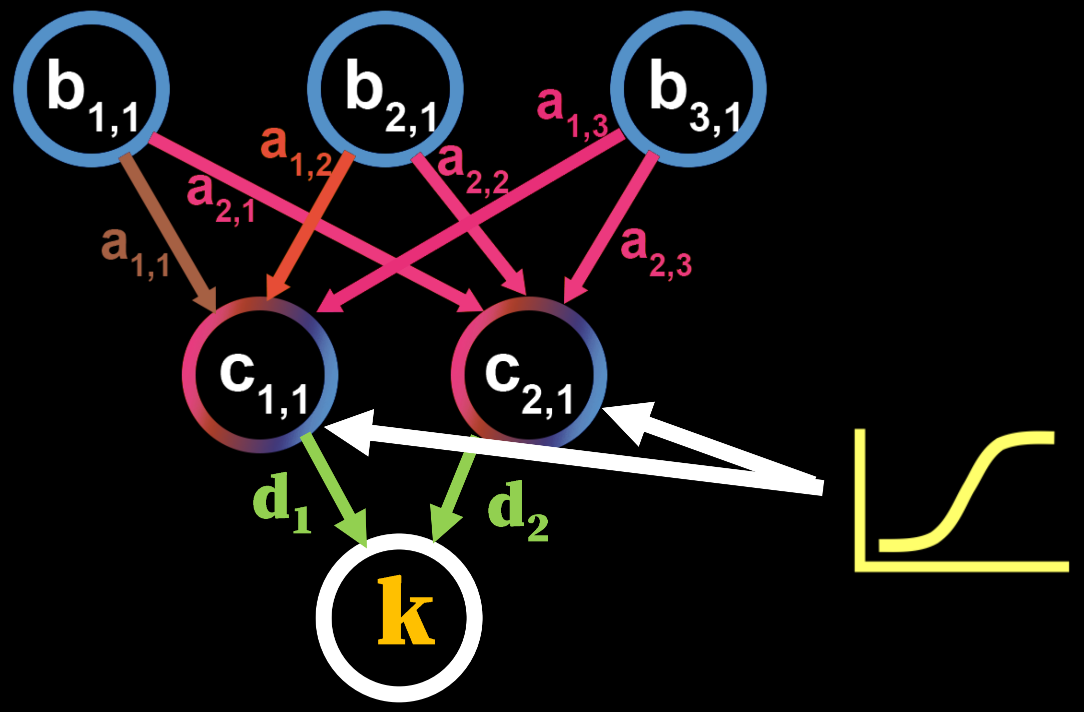


This is equivalent to a single-layer network – we don't need the composites C!

Activation functions

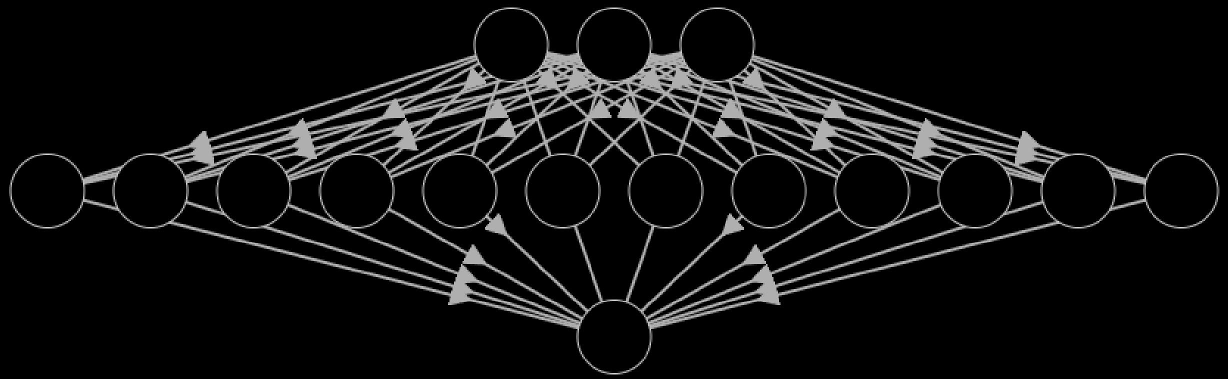
- This can be saved by carrying out some transformation of the composite variables c
  - Makes it so the weight matrices a and d can't just be combined

- This is called an **activation function** applied to a **hidden layer** (c)

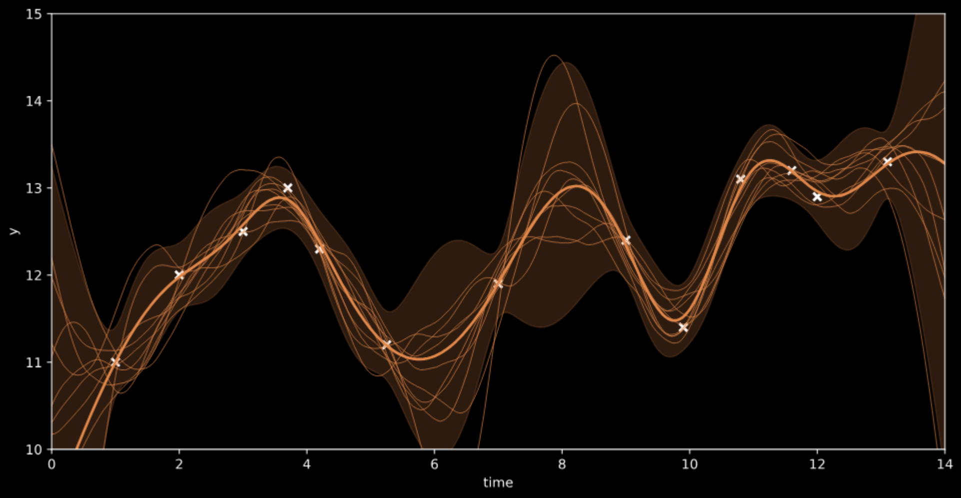


Universal approximation

- It is possible to prove that with enough nodes (c values) in the hidden layer, you can approximate any smooth function relating your predictors to your outcomes
  - Called the **universal approximation theorem** (Cybenko, 1989)



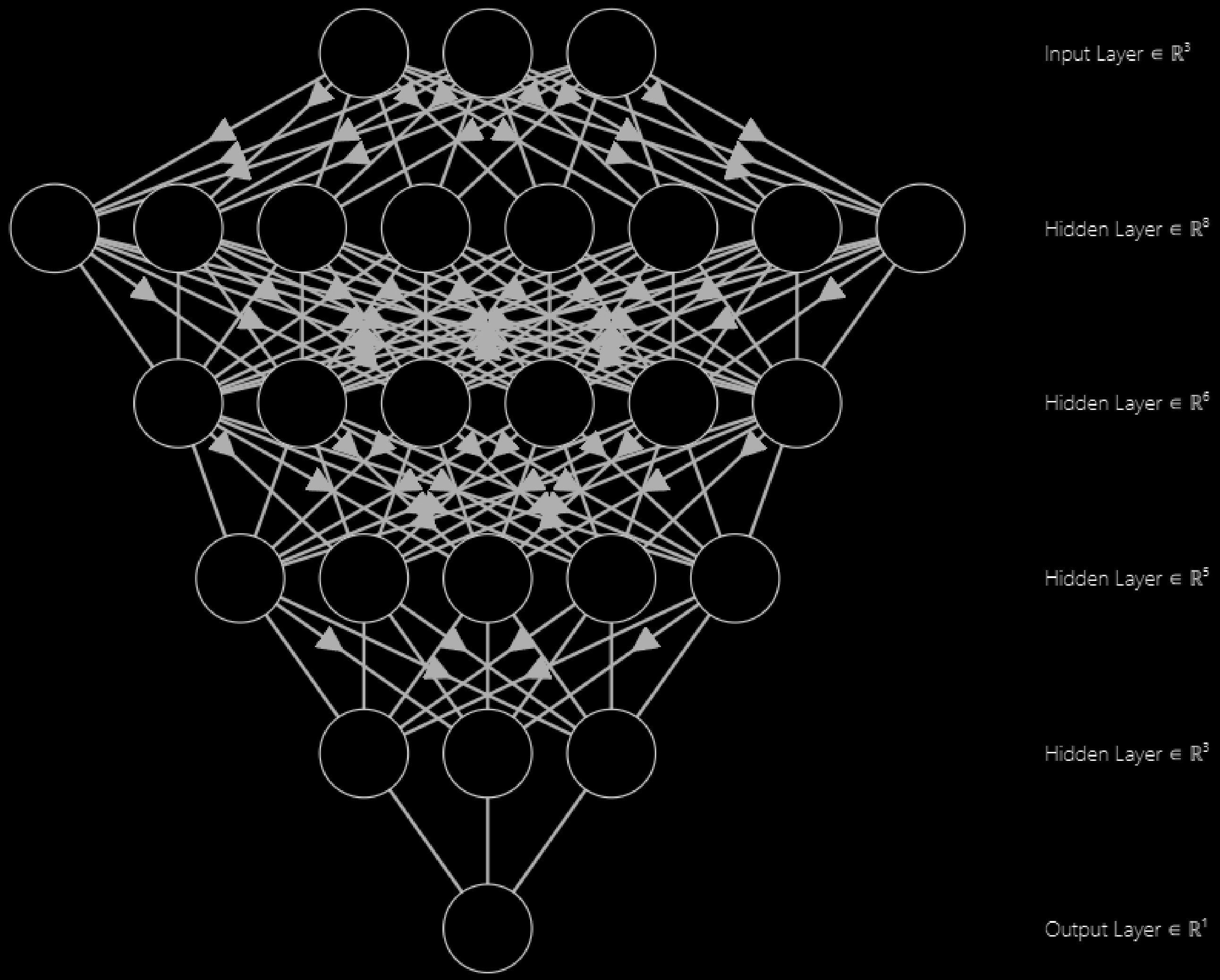
$\sim$



Deep neural networks

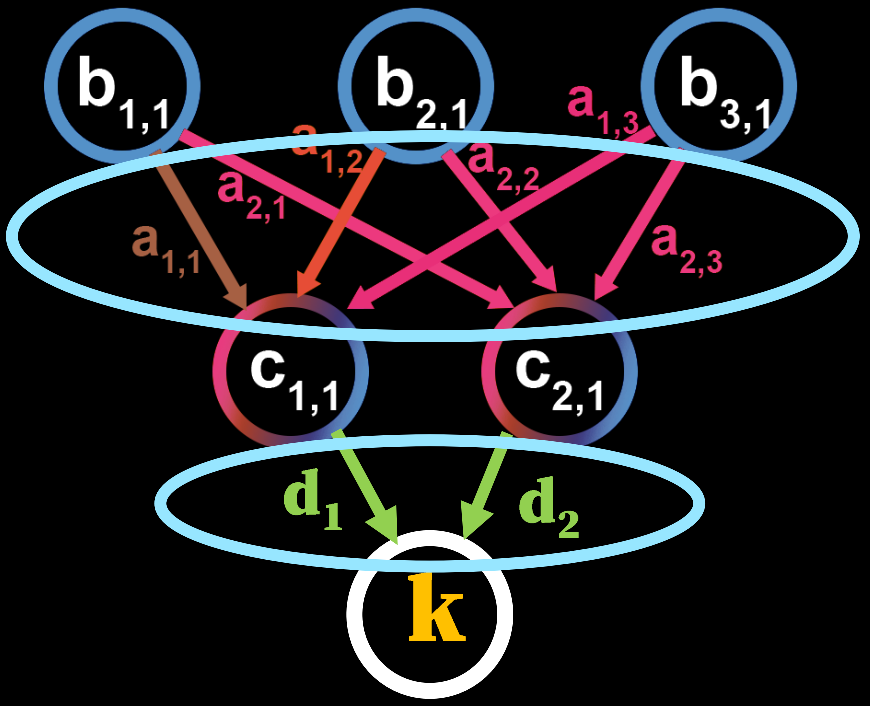
- Later, it was also proven that this could be done (often more efficiently) with **multiple hidden layers**
  - Gripenberg, 2003
  - Subsequent results suggested it could be done with relatively few layers and nodes
- Necessitated by XOR problem

- This property ultimately led to deep neural networks becoming *the* dominant machine learning approach



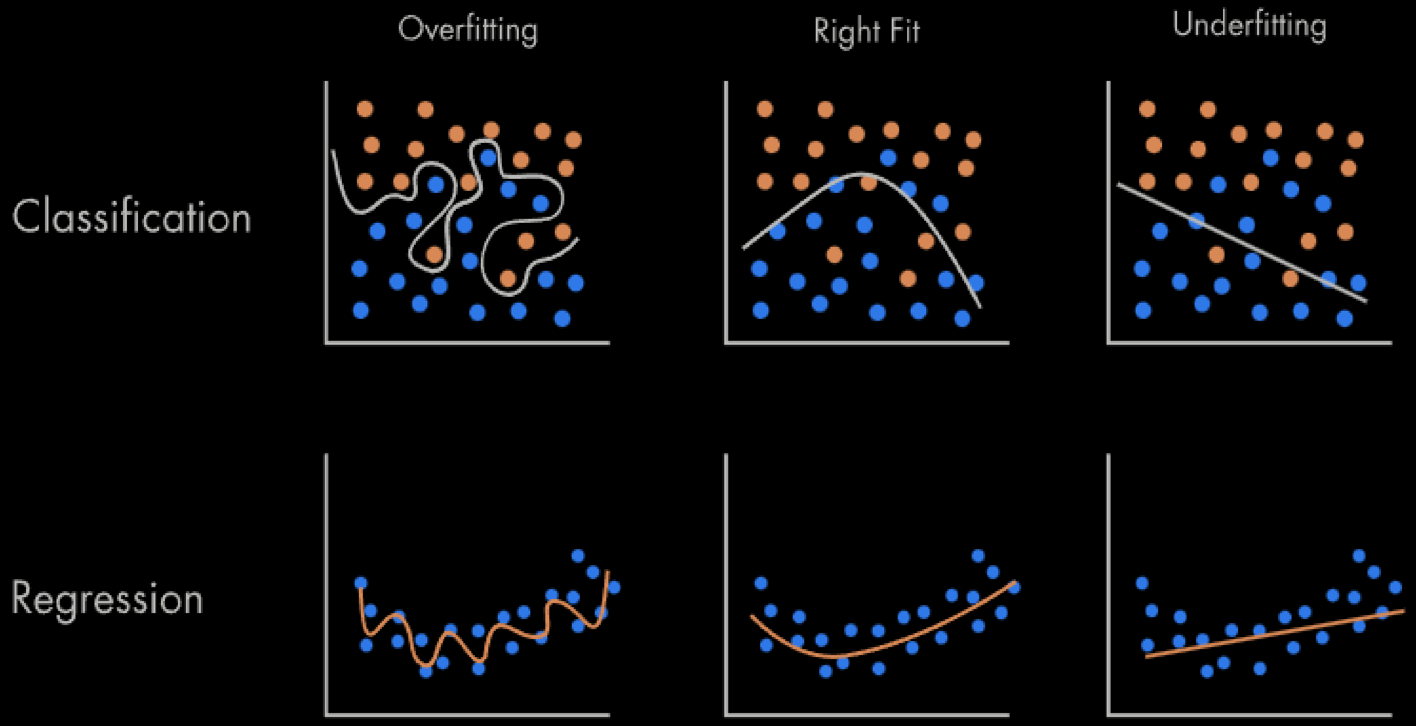
Training a neural network

- The next step is to teach your neural network what the relationships between predictors and outcomes are
- Estimate **weights** in the network
  - Values of all the lines connecting layers
- Try to minimize a **loss function**
  - How bad is your fit to the real data?
  - Often just mean squared error (L2)
  - Optimizers for a variety of losses
    - Cross-entropy (classification)
    - Mutual information
    - Absolute error (L1)



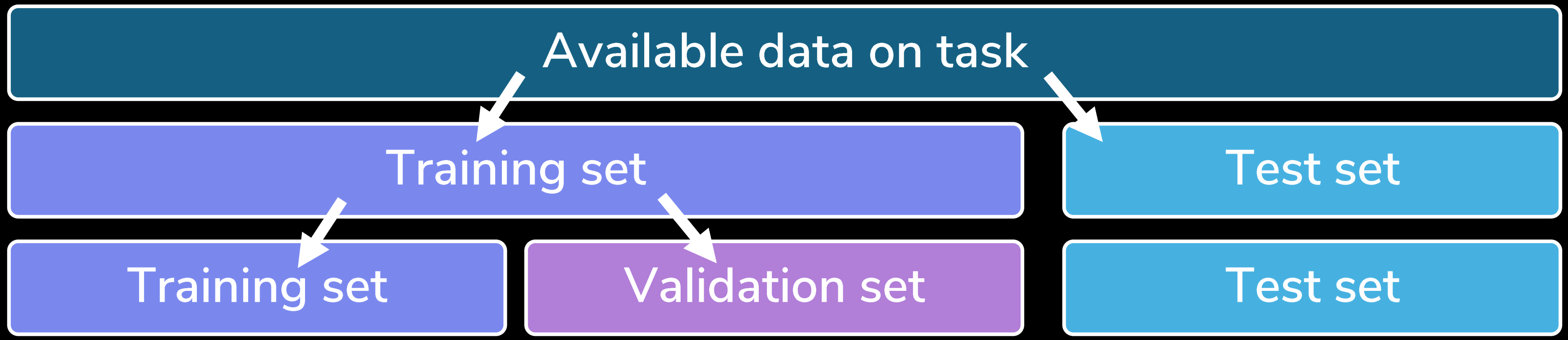
Testing a neural network

- Next, we see if it does what we want it to do
  - Typically scored via **cross-validation** (give some data that wasn't the training set, use tha to evaluate its performance)
- Check for things like **over-fitting**
  - Does it do well in training but not held-o data?



Training sets

- Any ML algorithm needs data from the world to adapt to its structure
  - In supervised learning, these are input-output mappings
  - In unsupervised learning, these may be quantifiable attributes / dimensions
  - Split into **training sets** and **test sets** (and sometimes validation)



- Training sets are absolutely critical to getting ML to do anything!
  - Examples of the task or structure you want it to learn

Agenda

- Machine learning fundamentals
  - What are AI, ML, neural networks, and deep learning?
  - Why are they so successful at a variety of tasks?

- Applications and special additions
  - Images / videos, text, sequential data
  - Problems with no correct solution (unsupervised learning)
  - ~~Generative adversarial networks~~

- Potential uses
  - Model fitting, comparison, discovery
  - Simulation studies of learning, evolution

Encoding & decoding

"What do encoding and decoding mean?"

Tokenizing

$$\begin{Bmatrix} \textit{what} \\ \textit{do} \\ \textit{en}- \\ \textit{coding} \\ \textit{and} \\ \textit{de}- \\ \textit{coding} \\ \textit{mean} \\ ? \end{Bmatrix}$$

Encoding

$$\begin{bmatrix} 1955 \\ 0009 \\ 8454 \\ 0004 \\ 8455 \\ 2675 \end{bmatrix}$$

Decoding

Encoding and decoding refer to the process of condensing and then re-creating information between particular (different) formats

Text input **Tokens** **Latent space** **Text output**

Sequential data – recurrent networks

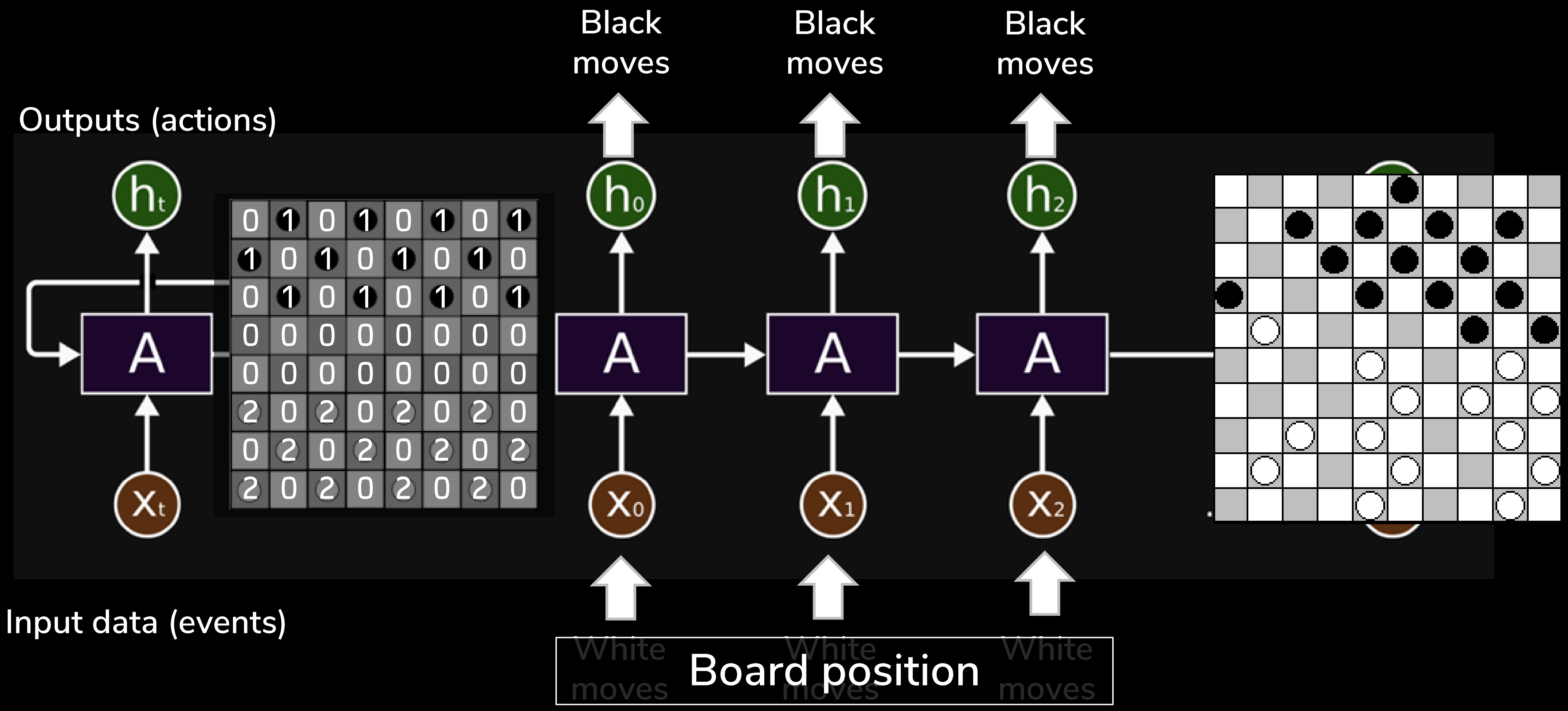
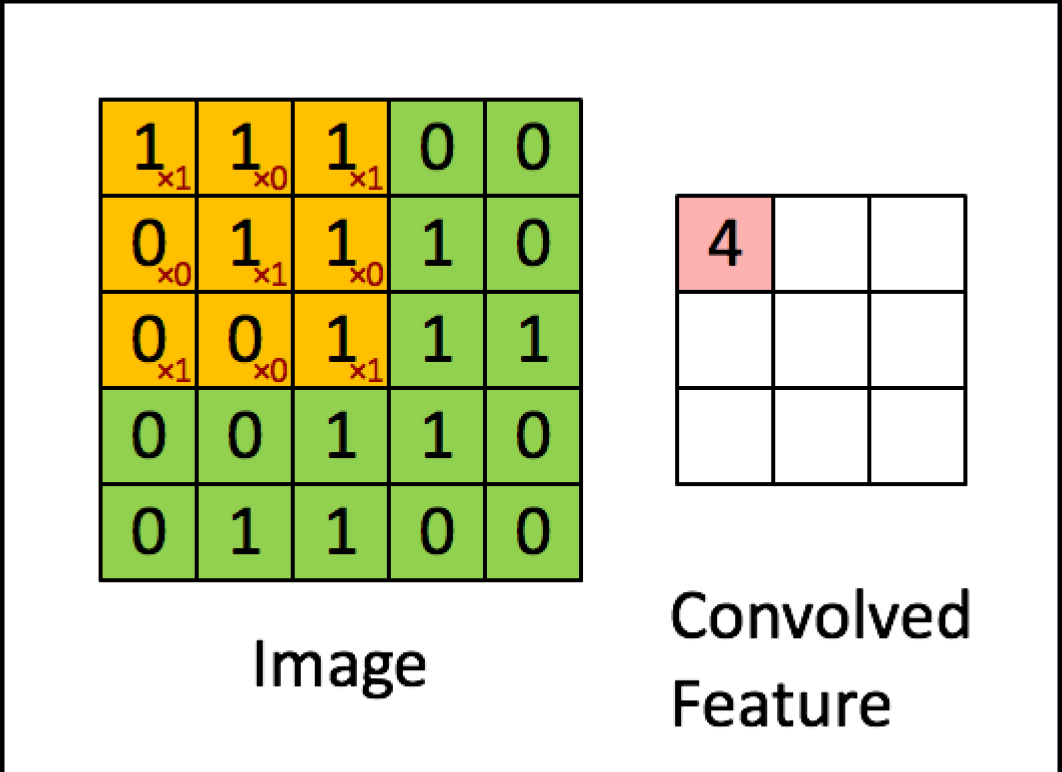


Image data - convolution



For example:

Input image

Convolution Kernel

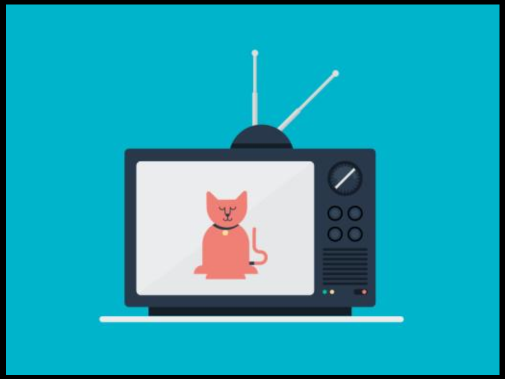
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map

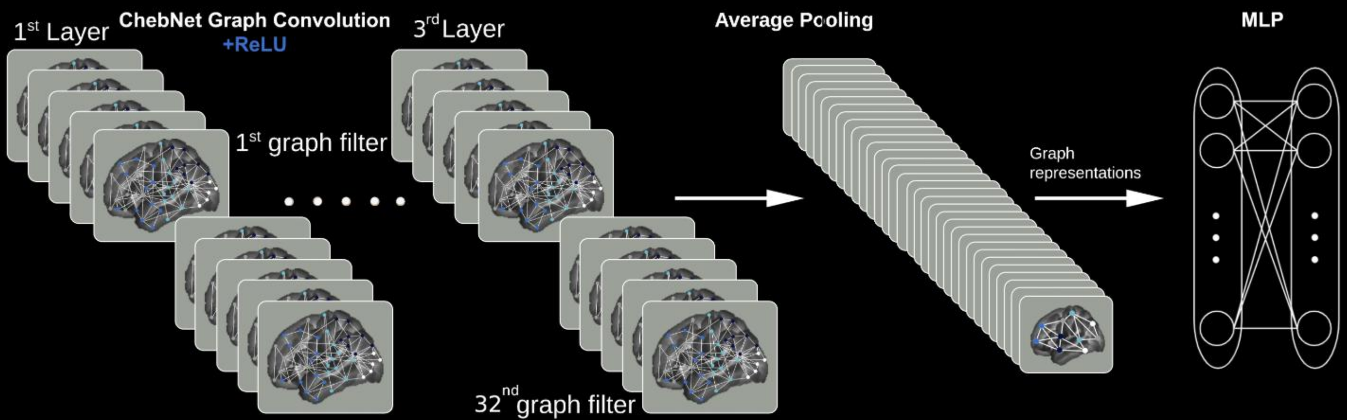
Process RGB pixels into (often smaller) data formats that can go into a neural network

Image data

- Convolutional, pooling, normalization layers allow the image to be represented efficiently as numbers
- Next, turn the encoded representation into some desired output:
  - Image label
  - Confidence (e.g., medical)
  - Another image (restoration, enhancement)

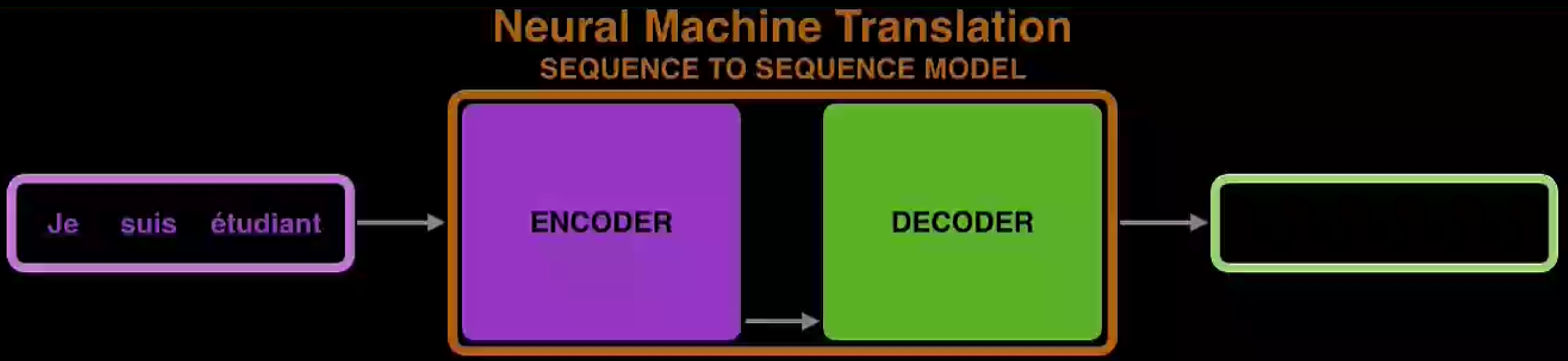


Color video: 4-D image data
Width
Height
Color channel (R, G, B)
Time



fMRI: 4+D image data

Text data – transformers / LLMs



Transformers involve "self attention" that modifies a word **encoding** based on its **context**, considering both input and position information in parallel.

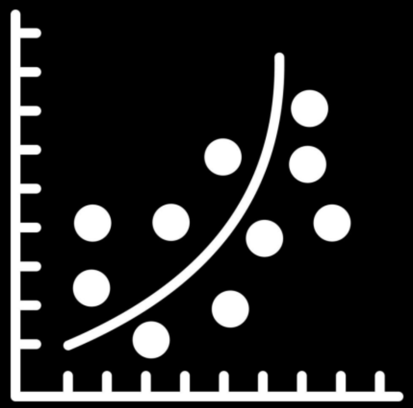
The **decoder** then uses self-attention and "encoder-decoder attention" to consider different properties (e.g., semantic, grammatical rules) when constructing an output.

Alammar, J (2018). The Illustrated Transformer https://jalammar.github.io/illustrated-transformer/

Agenda

- Machine learning fundamentals
  - What are AI, ML, neural networks, and deep learning?
  - Why are they so successful at a variety of tasks?
- Applications and special additions
  - Images / videos, text, sequential data
  - Problems with no correct solution (unsupervised learning)
  - Generative adversarial networks
- Potential uses
  - Model fitting, comparison, discovery
  - Simulation studies of learning, evolution

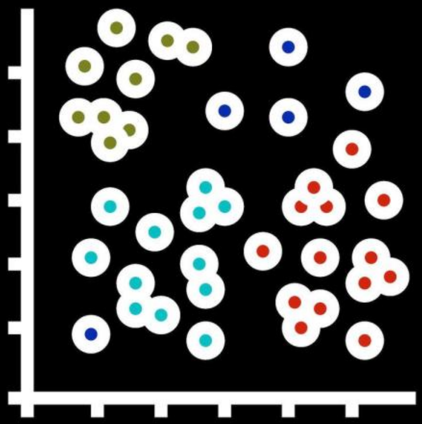
Flavors of machine learning



Prediction

Supervised learning

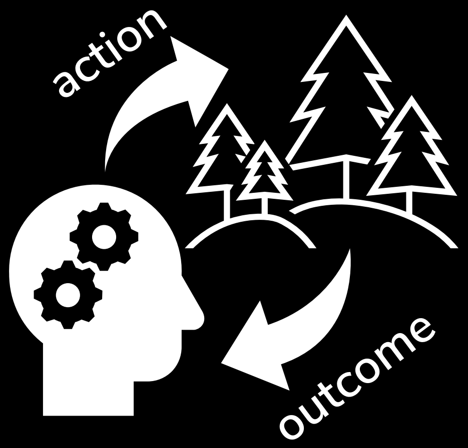
An algorithm approximates the relationship between variables



Organization

Unsupervised learning

An algorithm creates a structure to organize data based on their quantitative attributes



Exploration

Reinforcement learning

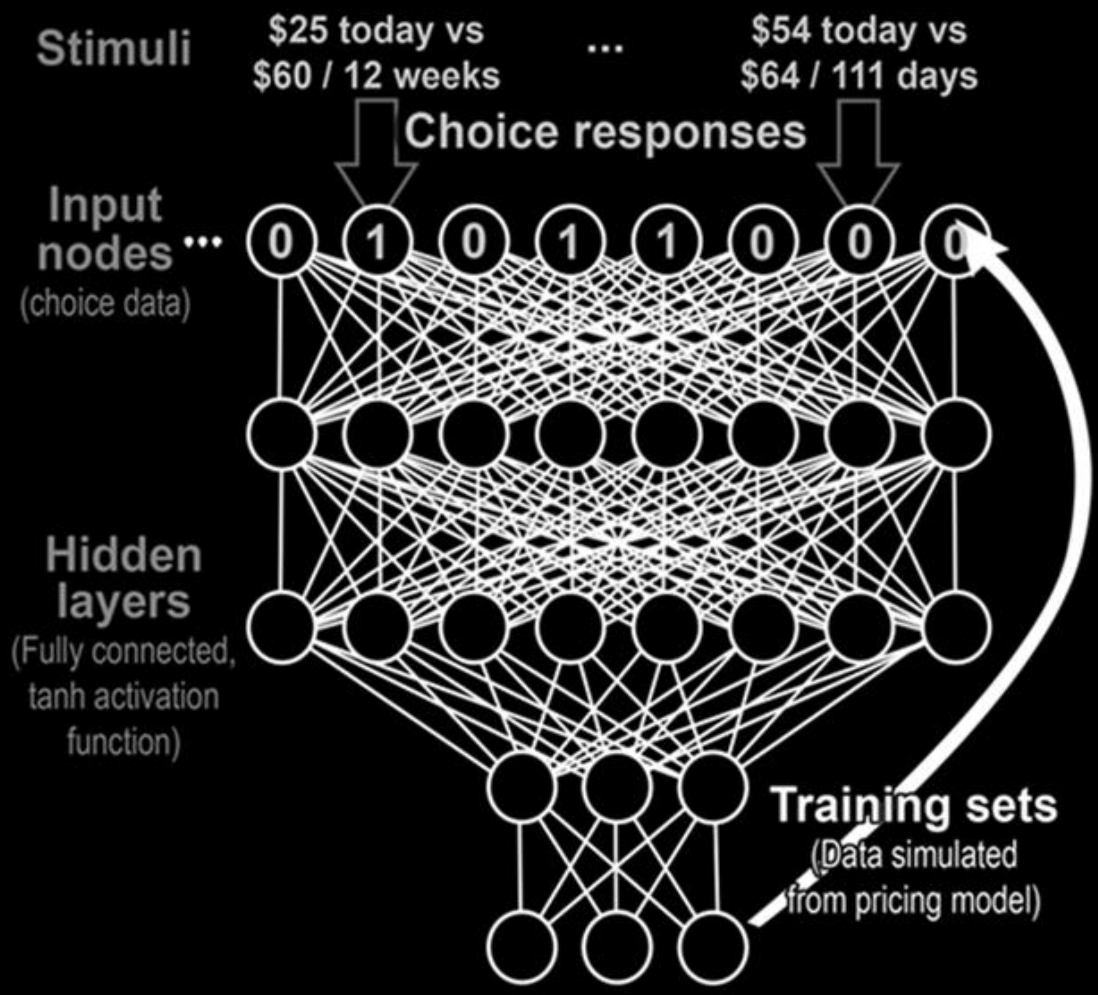
An agent interacts with its environment to learn action-response mappings

AI for model fitting

- Teach a machine to do the model fitting for us!
  - **Deep learning** for parameter estimation and model comparison

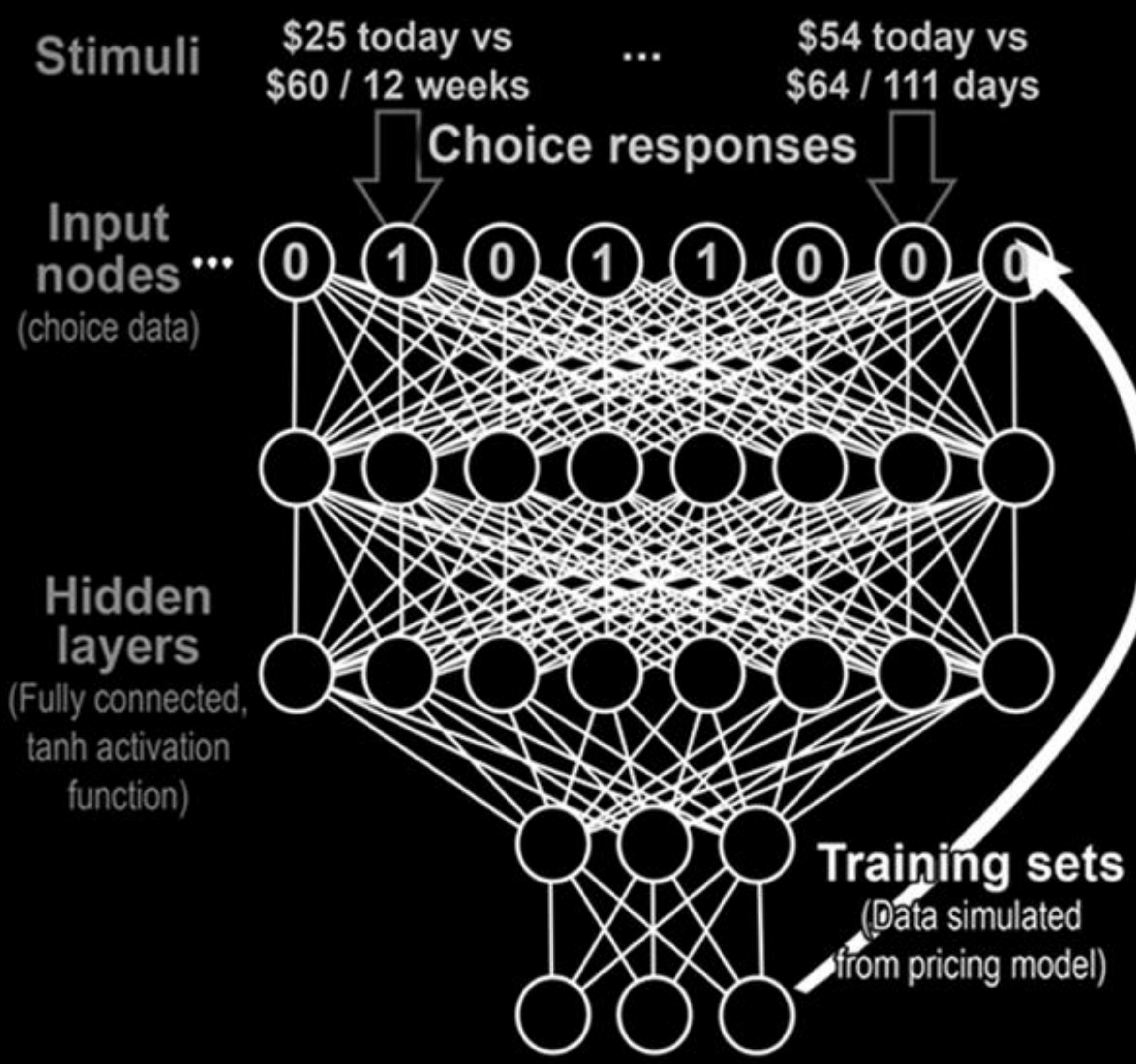
Simulation
Parameters → Data

Invert it!
Data → Parameters

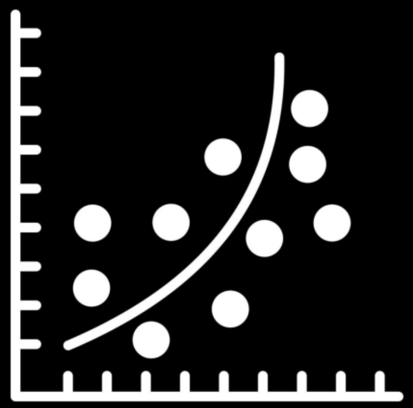


Model comparison

- Instead of parameters, the outputs are different models
    - Model comparison as a **classification** problem



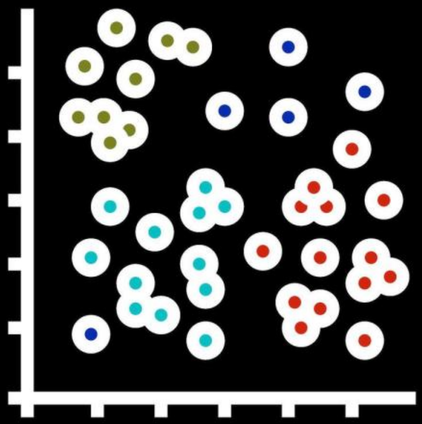
Flavors of machine learning



Prediction

Supervised learning

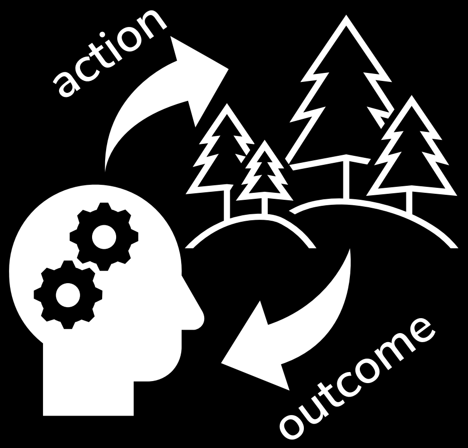
An algorithm approximates the relationship between variables



Organization

Unsupervised learning

An algorithm creates a structure to organize data based on their quantitative attributes



Exploration

Reinforcement learning

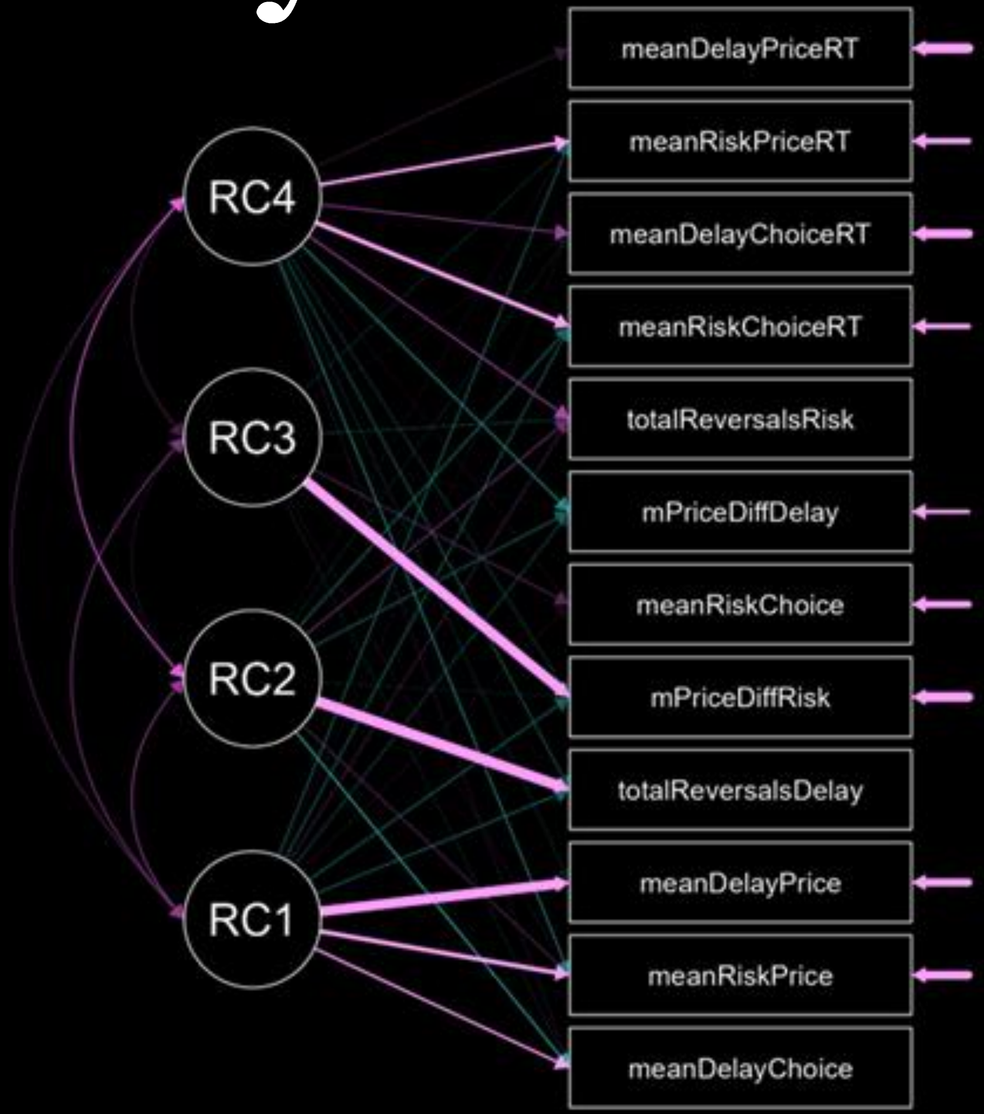
An agent interacts with its environment to learn action-response mappings

Exploratory data analysis

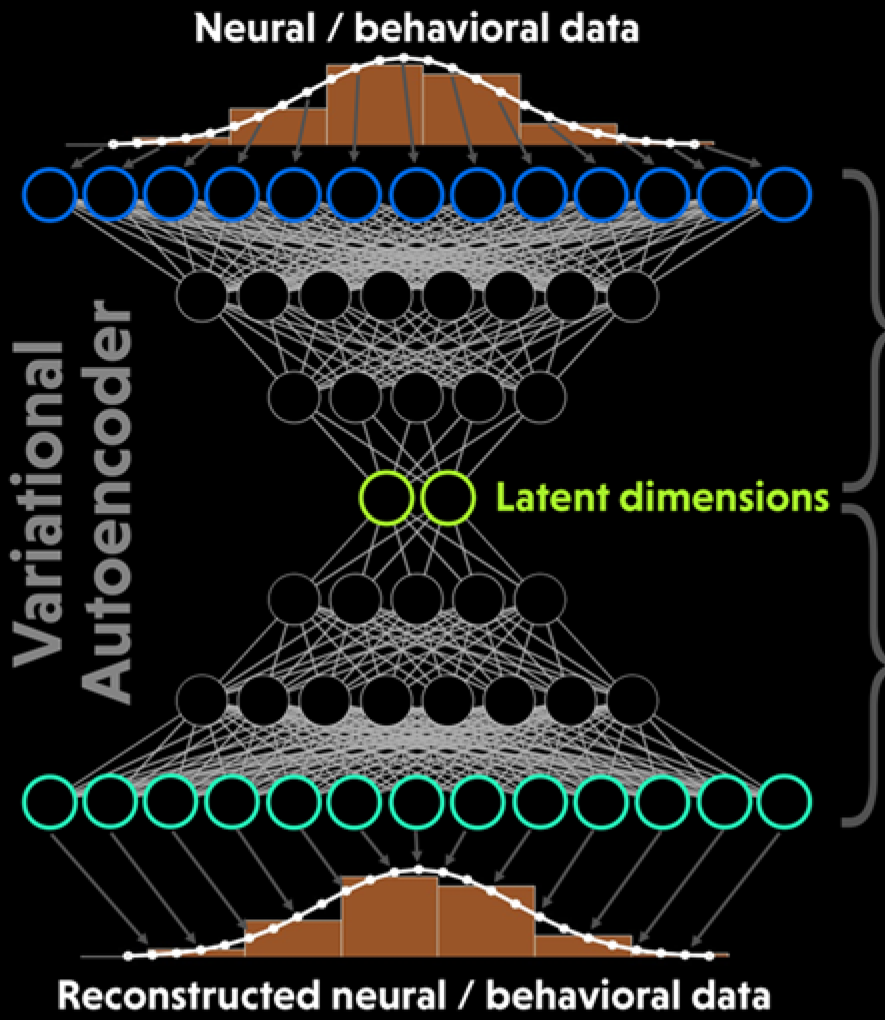
- Traditionally use approaches like **exploratory factor analysis** or **principal components analysis** to identify latent traits / processes

- However, these assume linear relationships between measures
  - Linear correlation / covariance matrix

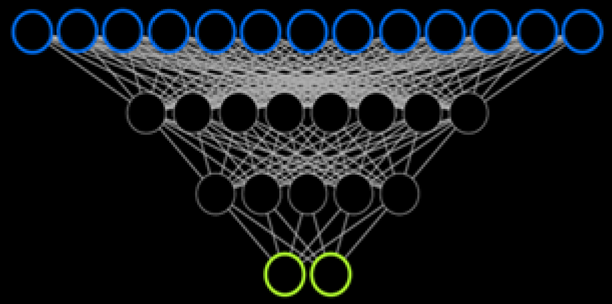
- They can also create “phantom oscillations” (Shinn, 2023)



Autoencoders

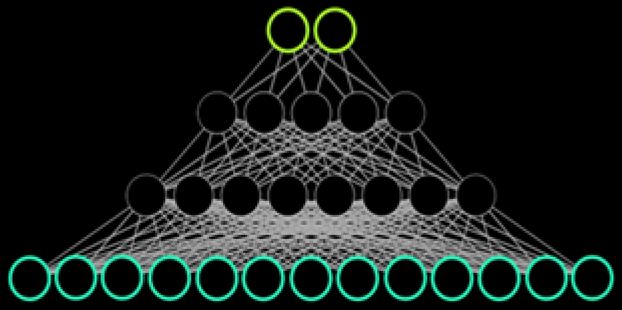


Encoder network



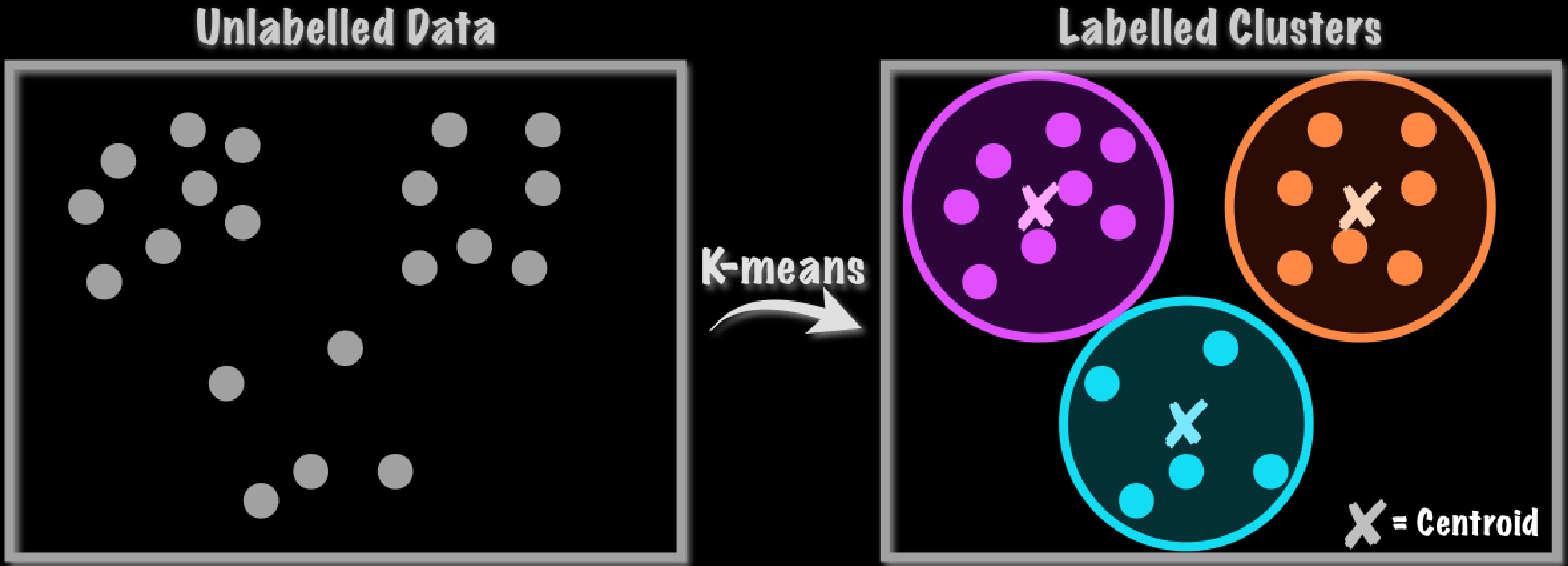
- Condense incoming data while preserving total information
- Identify latent constructs that predict / explain behavior
- Identify latent groups, new clusters (in combination with unsupervised learning)
- Exploratory data analysis

Decoder network



- Simulate expected data by varying latent dimensions
- Evaluate error in predictions
- Optimally impute / reconstruct missing data
- Explore nonlinear relationships between different outcome measures (via latent dimensions)

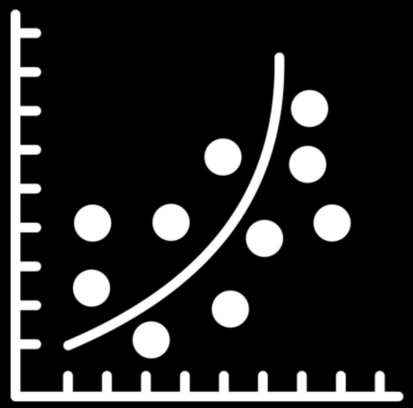
Clustering (unsupervised learning)



Lots of clustering methods: Connectivity (hierarchical) clustering, centroids (k-means), distribution, density, graph-based models

Lots of applications: Patient classification, genetic clustering, mental health typology, grouping decision strategies into types

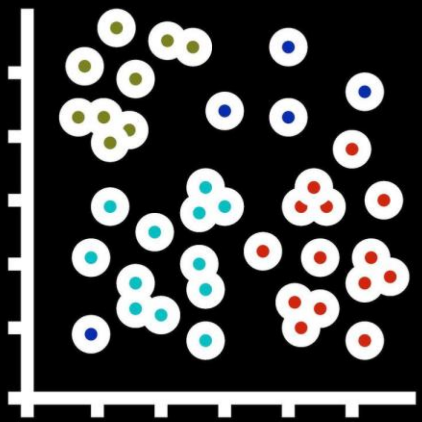
Flavors of machine learning



Prediction

Supervised learning

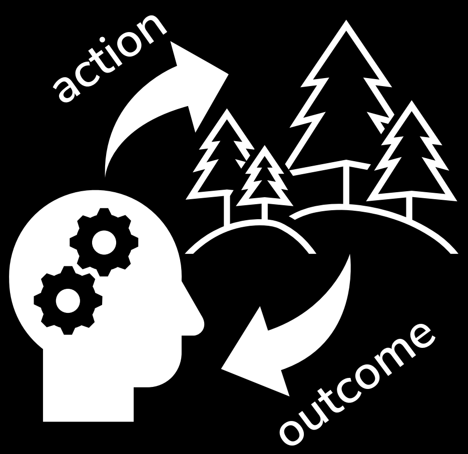
An algorithm approximates the relationship between variables



Organization

Unsupervised learning

An algorithm creates a structure to organize data based on their quantitative attributes

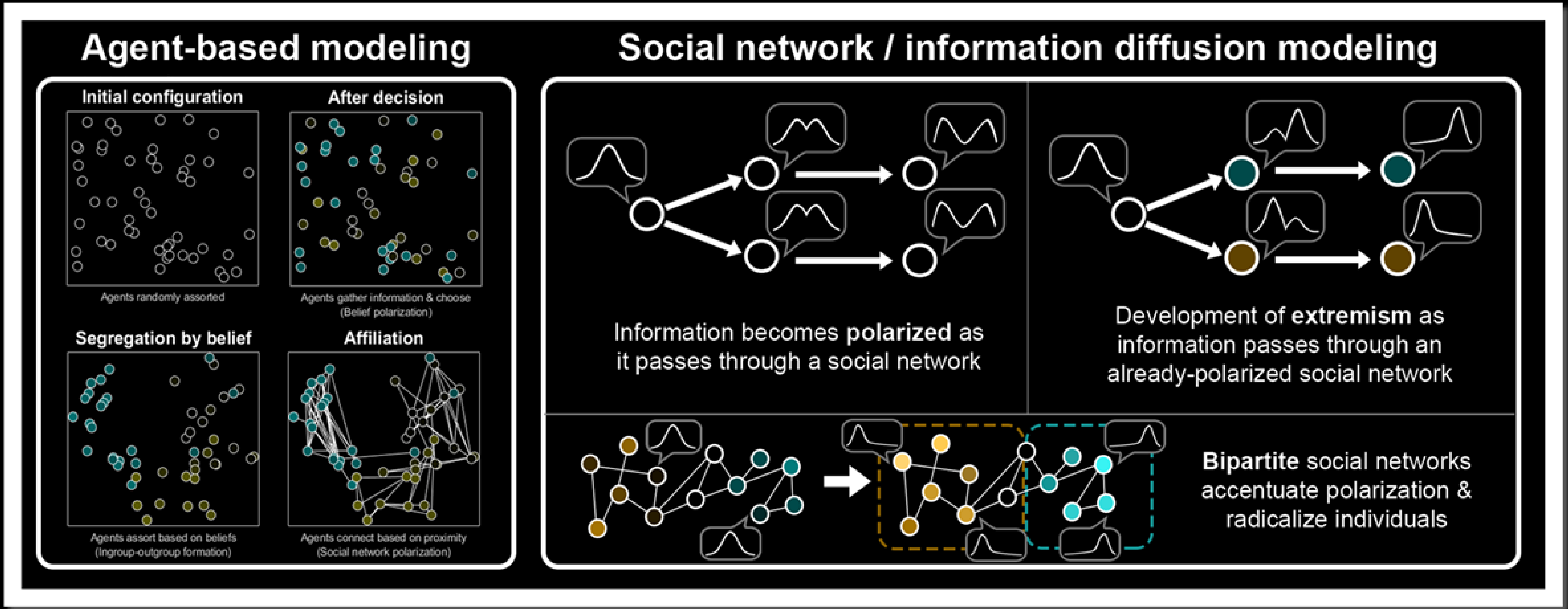


Exploration

Reinforcement learning

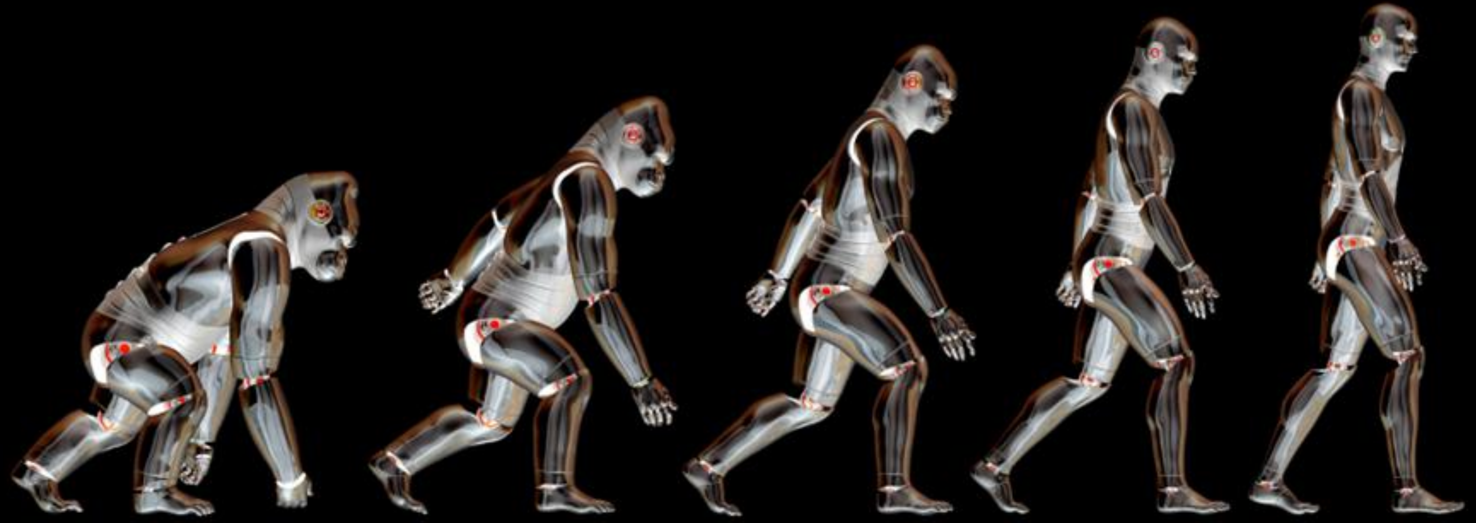
An agent interacts with its environment to learn action-response mappings

Agent-based modeling



Exploring hypotheses

- Test evolutionary preconditions by evolving artificial agents
  - Genetic programming, mutation, selection, reproduction, inheritance
  - **Computational evolution** for evolving intelligent systems



- Test theories of (reinforcement) learning by simulating them in artificial agents / brains
- Both are ways to examine emergent behavior from simple principles

Conclusions

- Modern machine learning relies on its ability to **approximate relationships between inputs & outputs**
    - Structure inspired by human neural networks, but diverge in important and meaningful ways – don't treat it like a human

- Tasks involving prediction, discovering structure, or simulation can make good use of existing tools
    - Sometimes they can be used **out of the box**: tokenizing / analyzing text, analyzing images / video, proofreading
    - Other times, we need to **train systems from scratch** ourselves: model fitting / comparison, patient classification, simulations
        - Different fields need different tools – what is particular to your field?

Breakout-relevant questions

- Does your field benefit from:
  - General-purpose tools (image, text, video analysis)?
  - Field-specific tools (e.g., protein folding, specific models)?

- What industry applications can we imagine?
  - Intelligent tutors working in real time
  - Defense systems that detect latent human goals (nefarious or not)
  - Assessment tools for neurological / mental health
  - Enhancing human-machine teams with mutual "understanding"