



# DEEP LEARNING

AAE722 Machine Learning in Applied Economic Analysis

Agricultural & Applied Economics @ UW-Madison

Yuji Saikai

# MANY PATHWAYS TO DEEP LEARNING

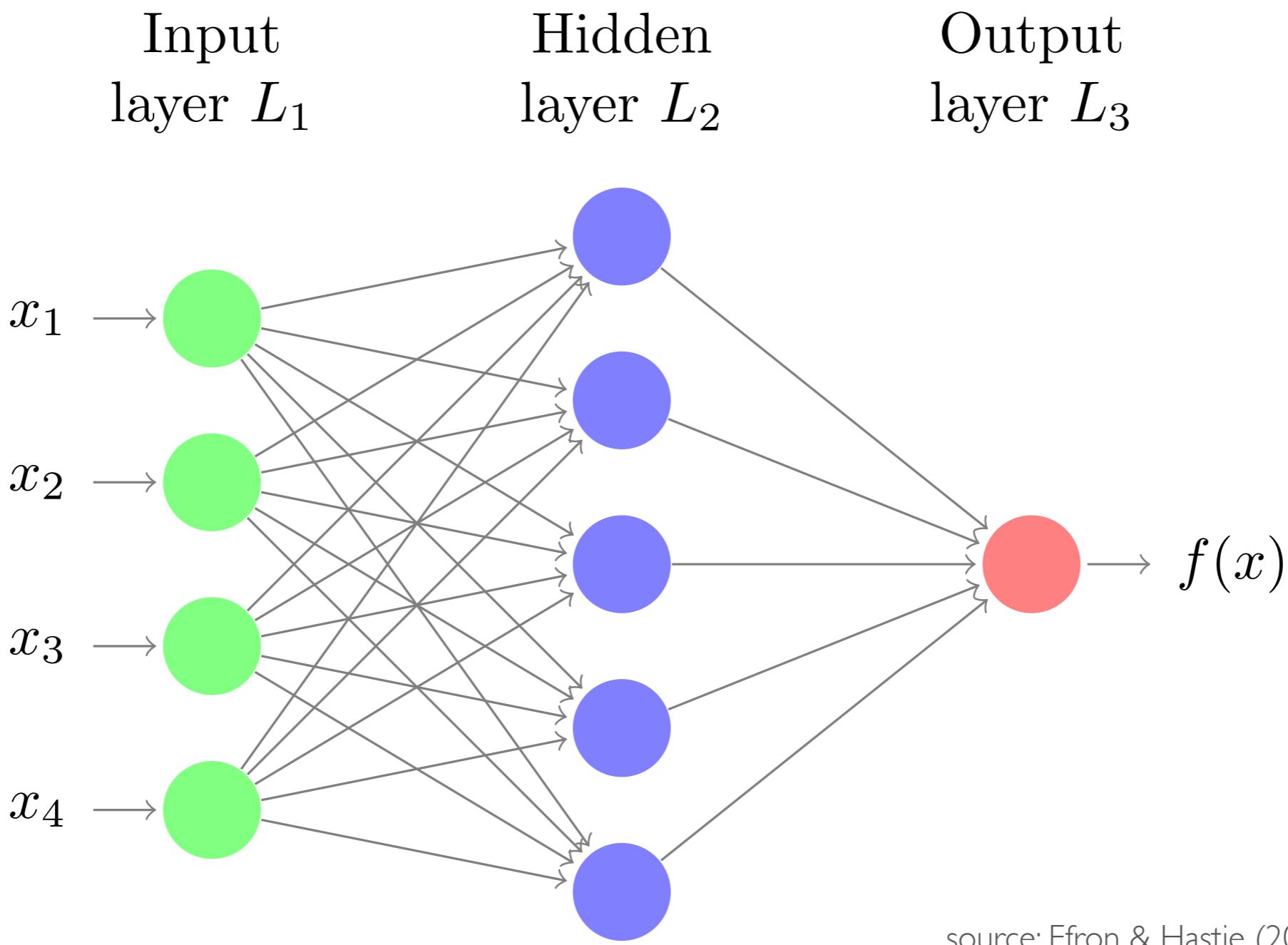
- Keras  
Quickest way to implement deep learning
- Andrew Ng (Coursera)  
Foundations for practitioners
- Sebastian Raschka (STAT479, UW-Madison)  
More in-depth training for both research & practice

# SUMMARY

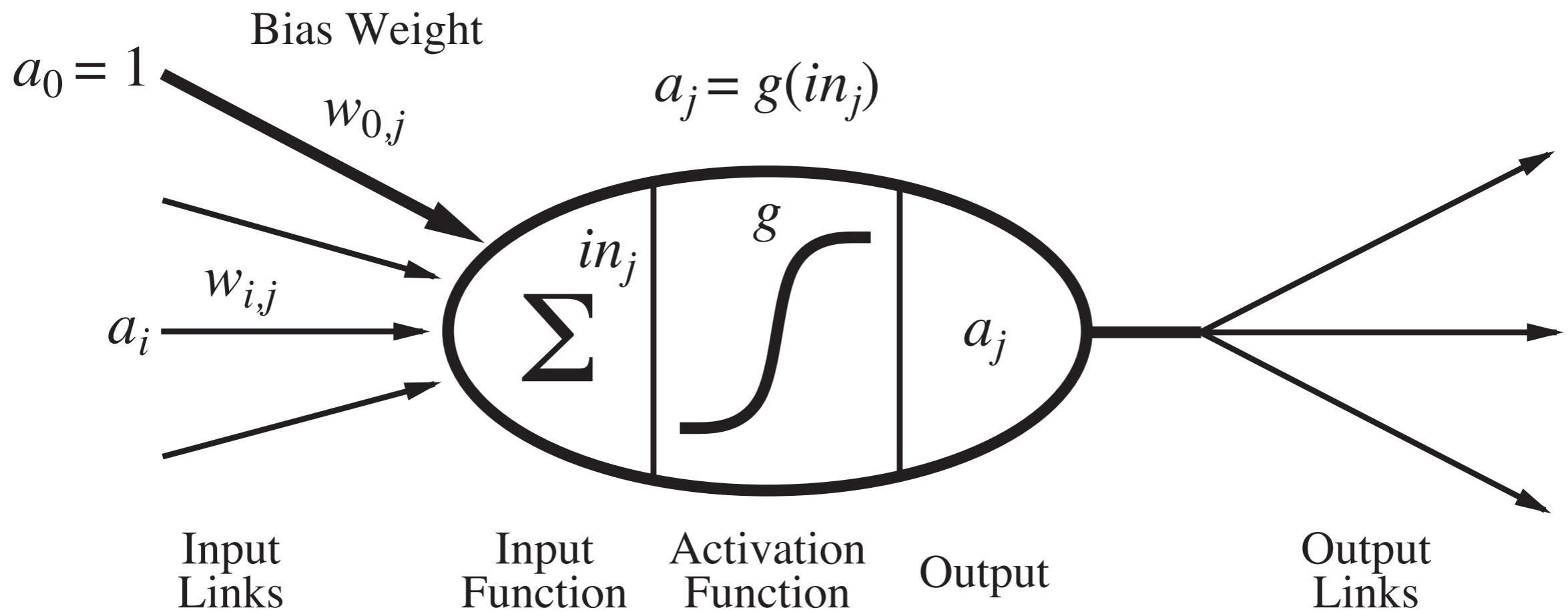
- Review of neural network
- Representation learning
- Architectural nature
- Applications
- Demos



# NEURAL NETWORK



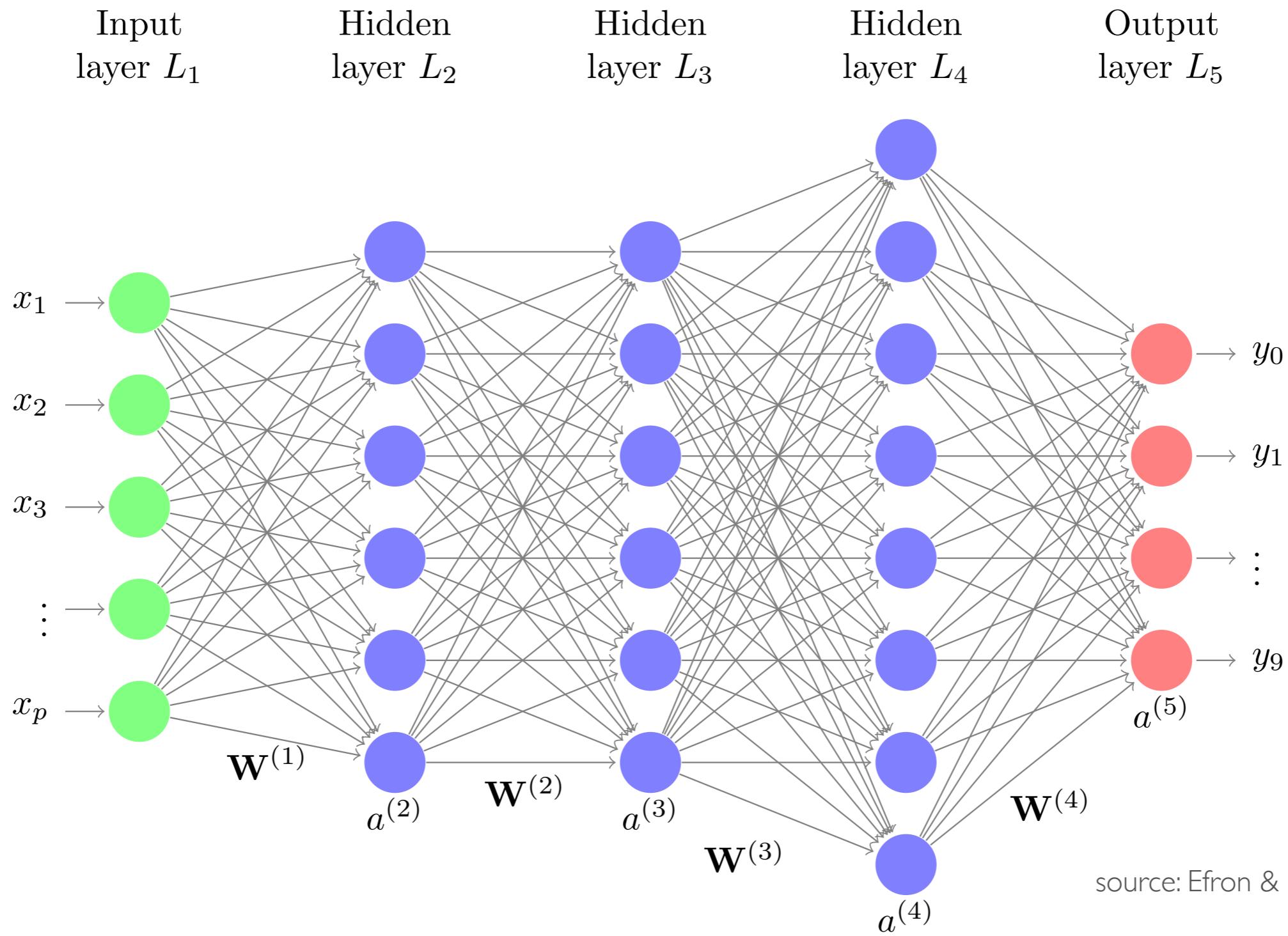
# COMPUTATIONAL UNIT



source: Russell & Norvig (2010)

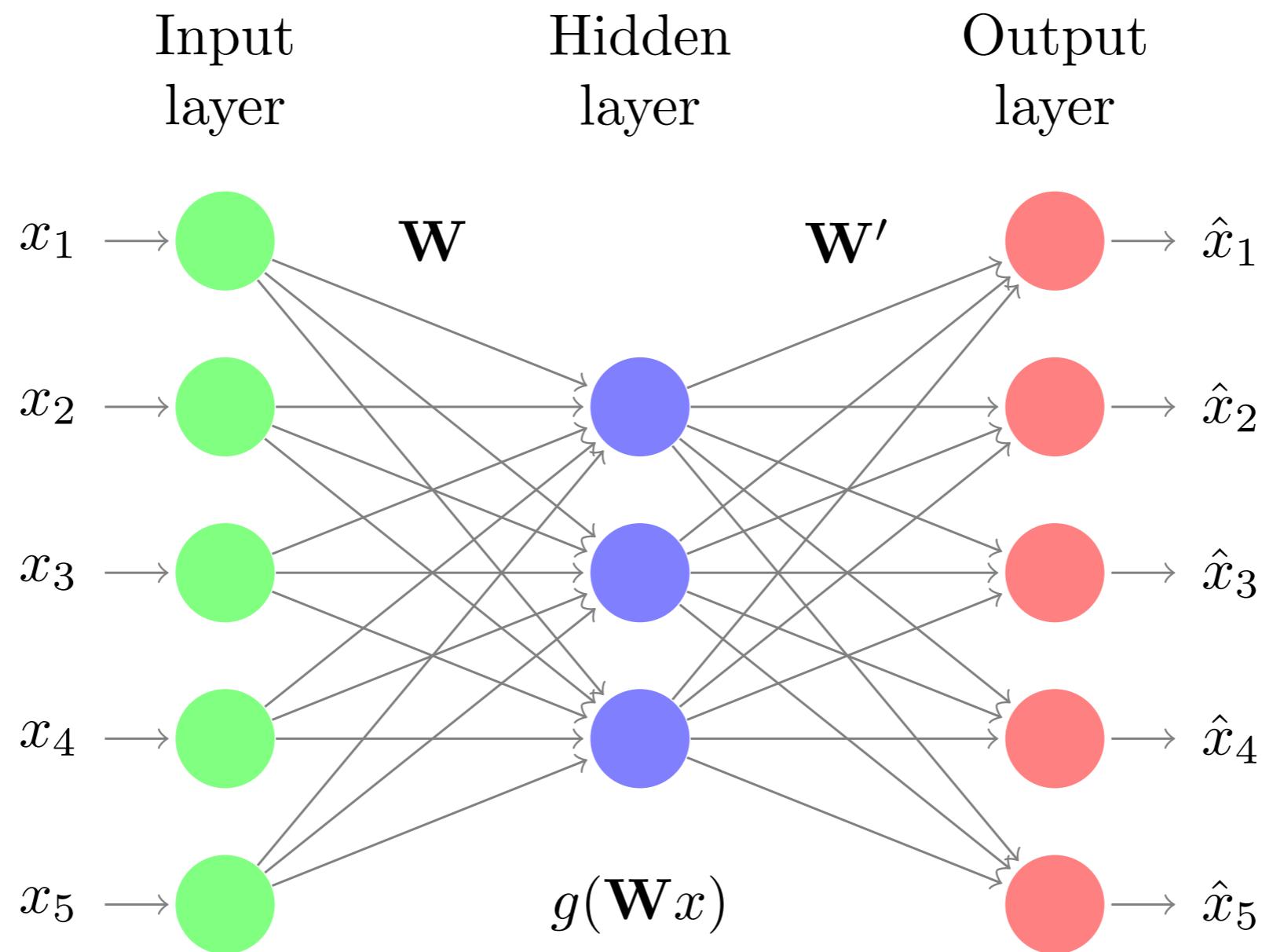
The most common  $g( )$  is ReLU —  $g(in_j) = \max(0, in_j)$

# DEEPER NETWORK & MULTIPLE OUTPUTS



source: Efron & Hastie (2016)

# AUTOENCODER

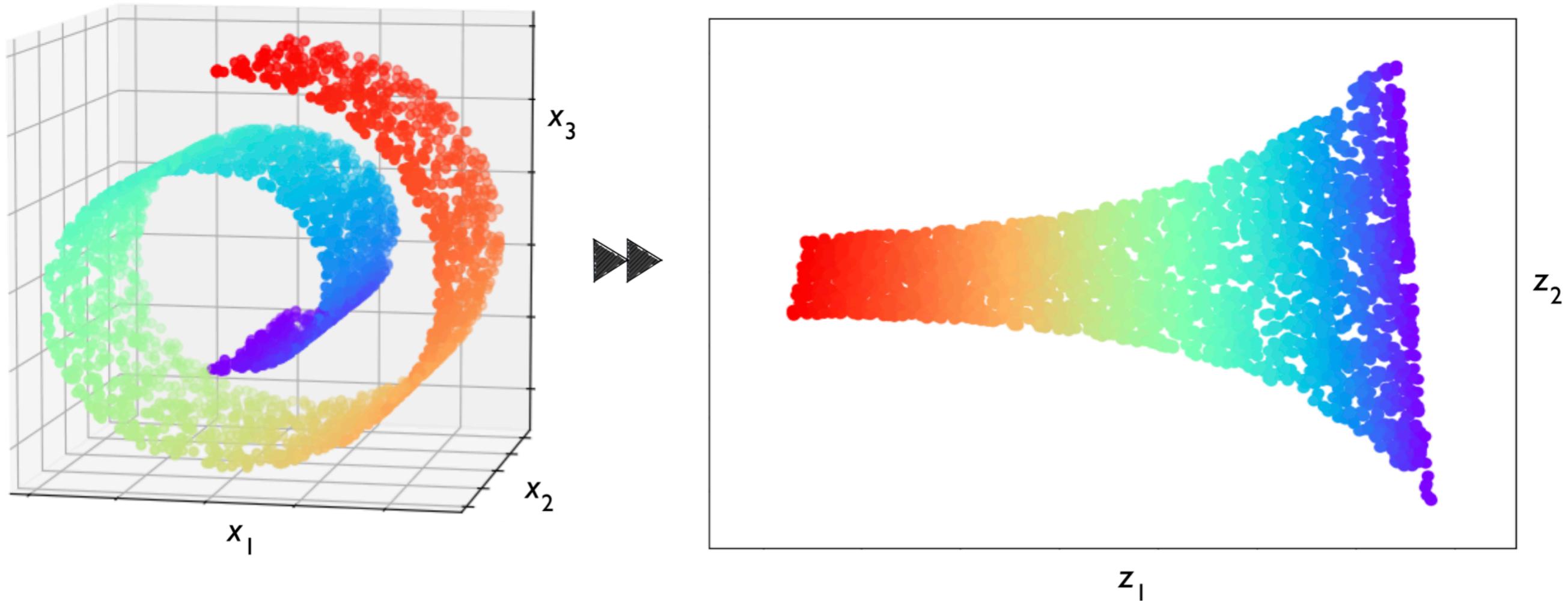


source: Efron & Hastie (2016)

# REPRESENTATION LEARNING

- A key perspective to understand DL
- Some **representations** of data are more useful than others
- **Learning** such representations also from data, rather than handcrafting

# REPRESENTATION MATTERS

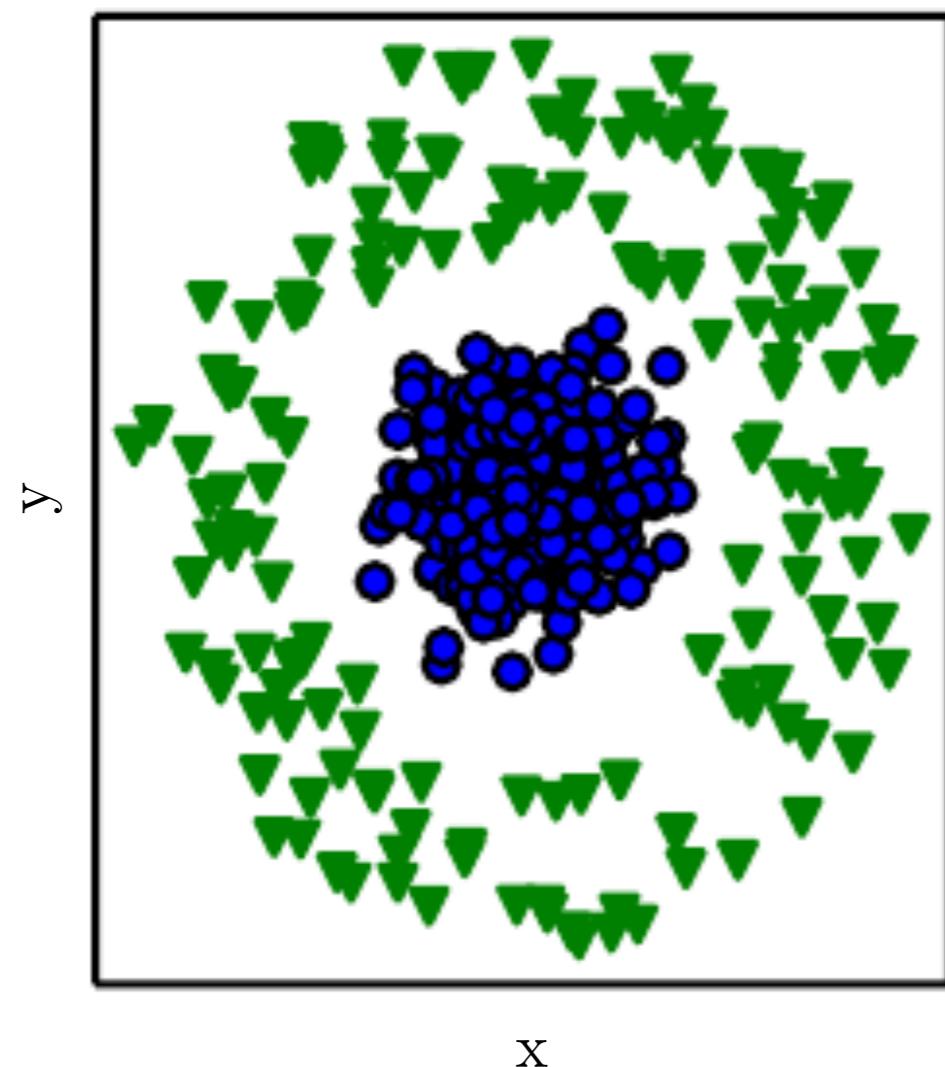


Manifold learning using kernel PCA

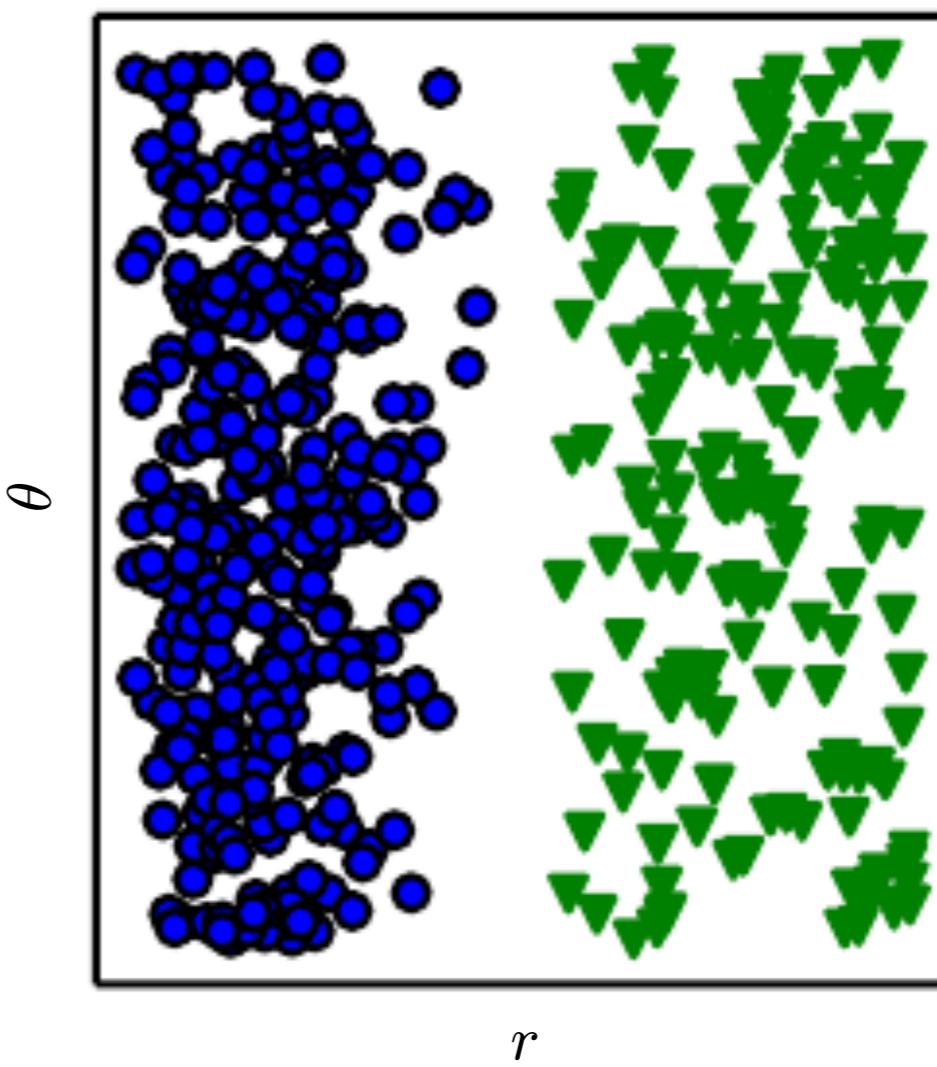
source: Sebastian Raschka

# KEY TO UNDERSTAND CAUSALITY

Cartesian coordinates



Polar coordinates



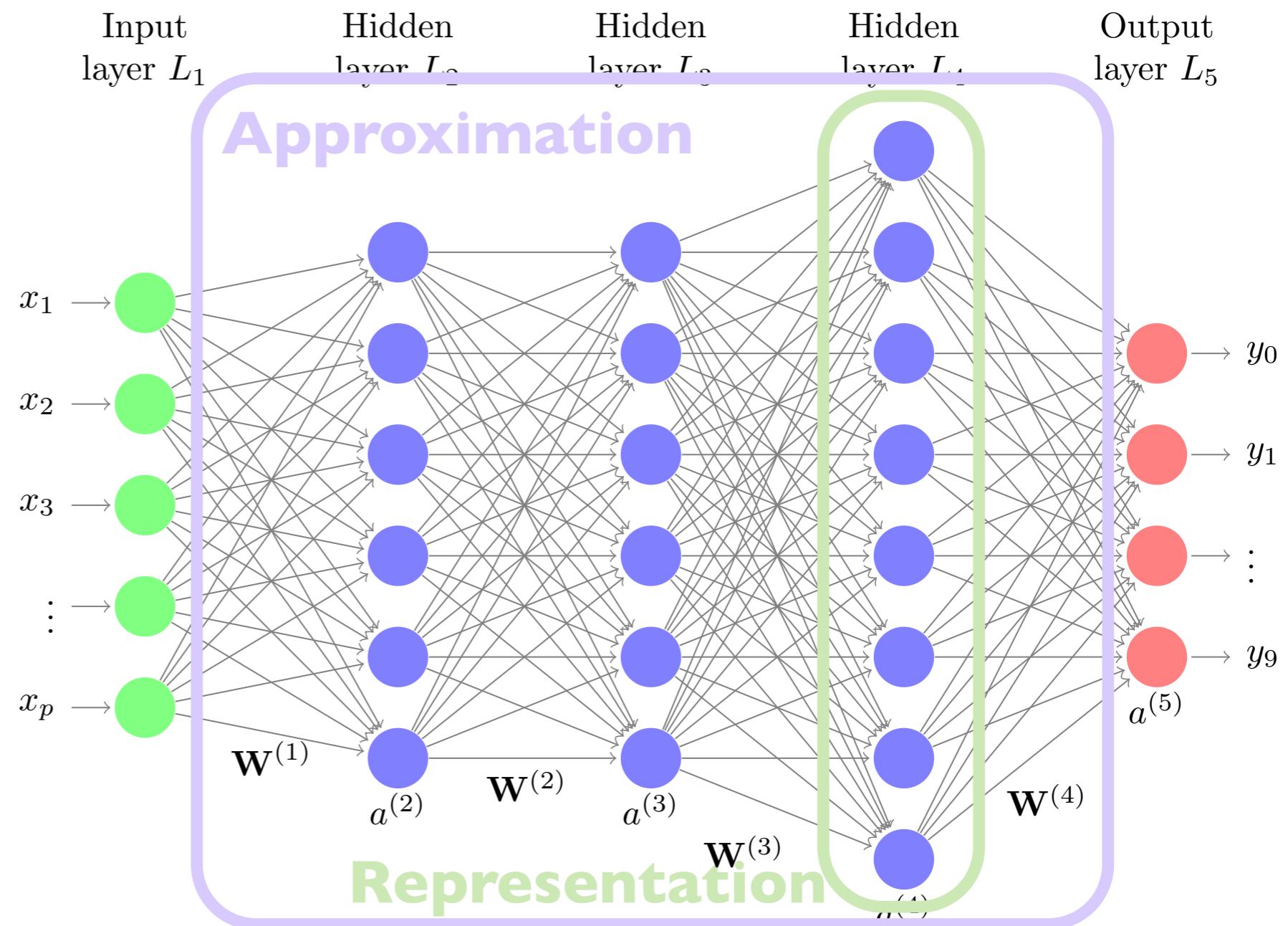
source: Goodfellow, Bengio, & Courville (2016)

# HANDCRAFT Is HARD

- Polynomial regression, but in what degree? with which interaction terms?
- Support vector machine, but what kernel to use?
- For machines to recognize faces, which parts are important? Eyes and mouth? Why not nose? How should we express them in numbers?

# APPROXIMATION OR REPRESENTATION?

- All we want is to predict  $y$  from  $x$
- “Representation” to emphasize the key roles of transformed  $x$

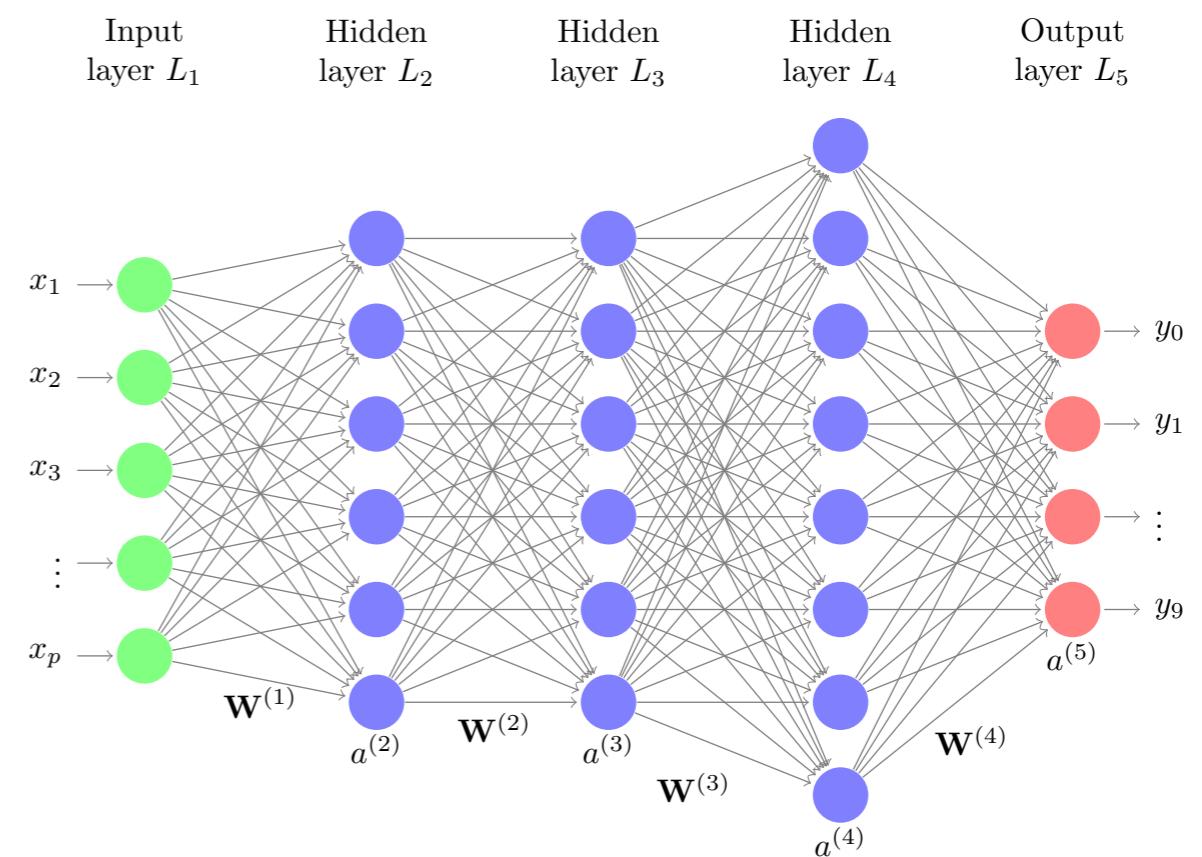
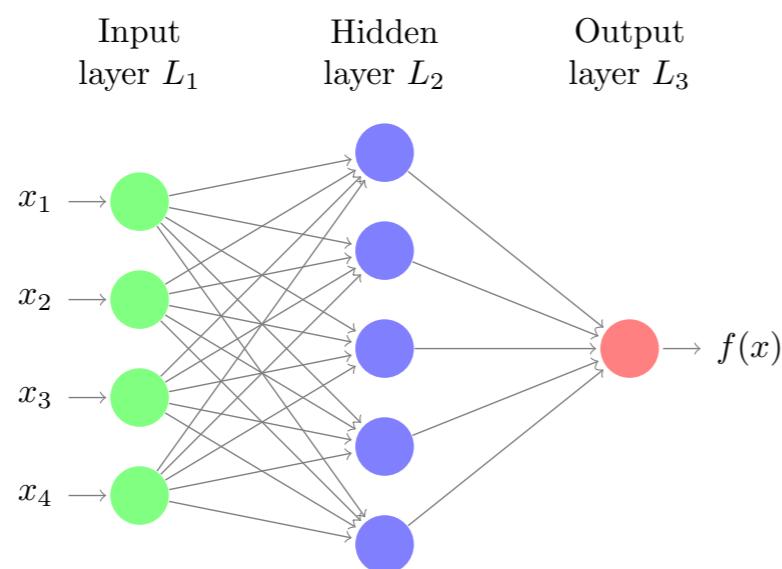
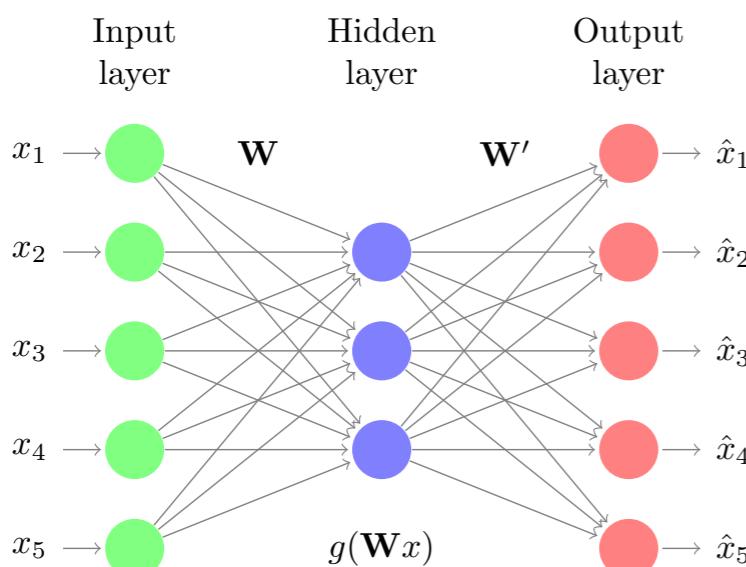


# WHY “DEEP”?

- “many natural signals are compositional hierarchies, in which higher-level features are obtained by composing lower-level ones”—LuCun, Bengio, & Hinton (2015)
- shallow networks can “require many more computational elements, potentially exponentially more”—Bengio (2009)
- “the phrase is just a great brand, it’s just so deep”—Andrew Ng

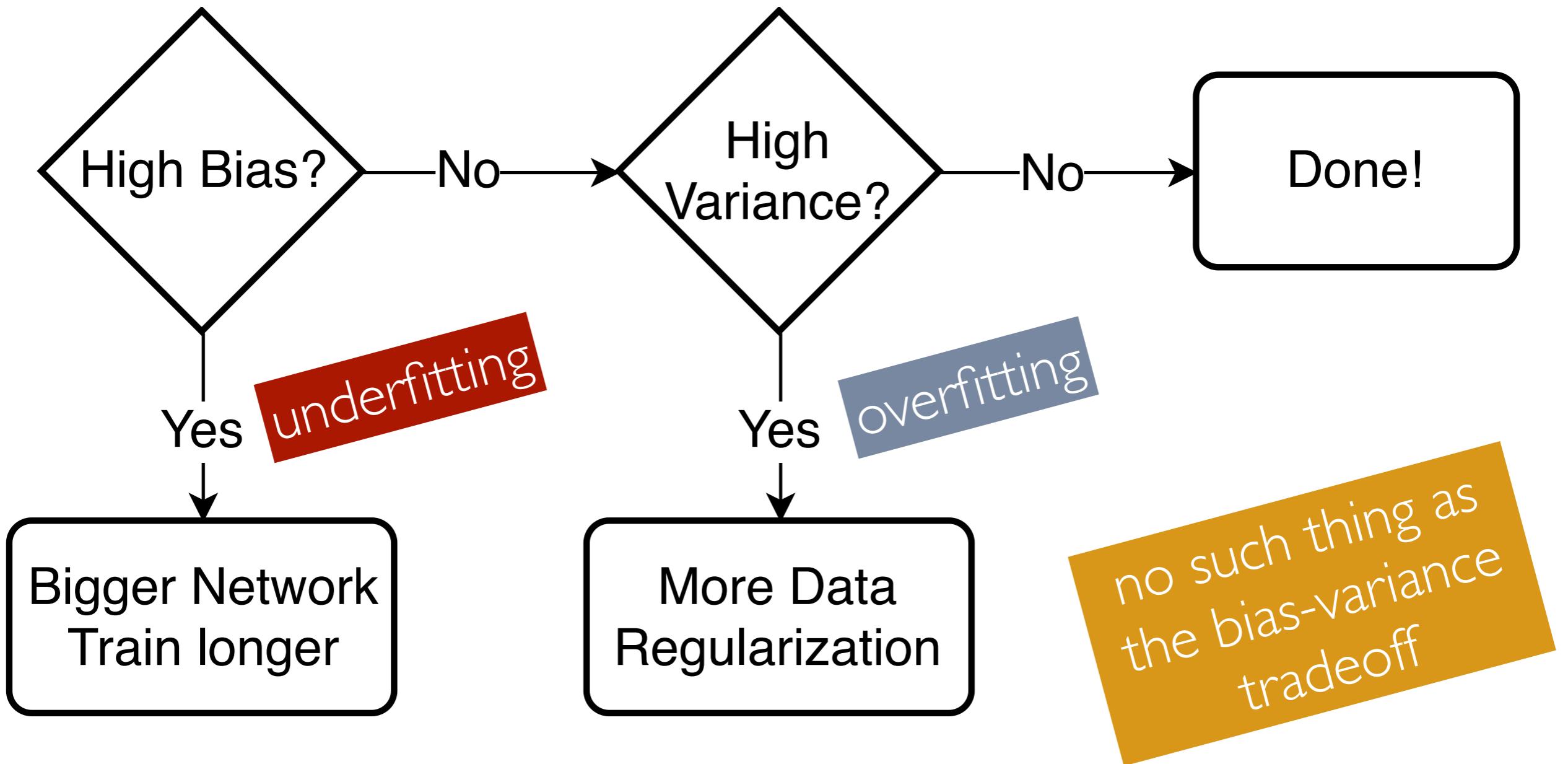
# MORE ART THAN SCIENCE

so many “architectures”



There is a basic recipe but  
no hard-and-fast rule!

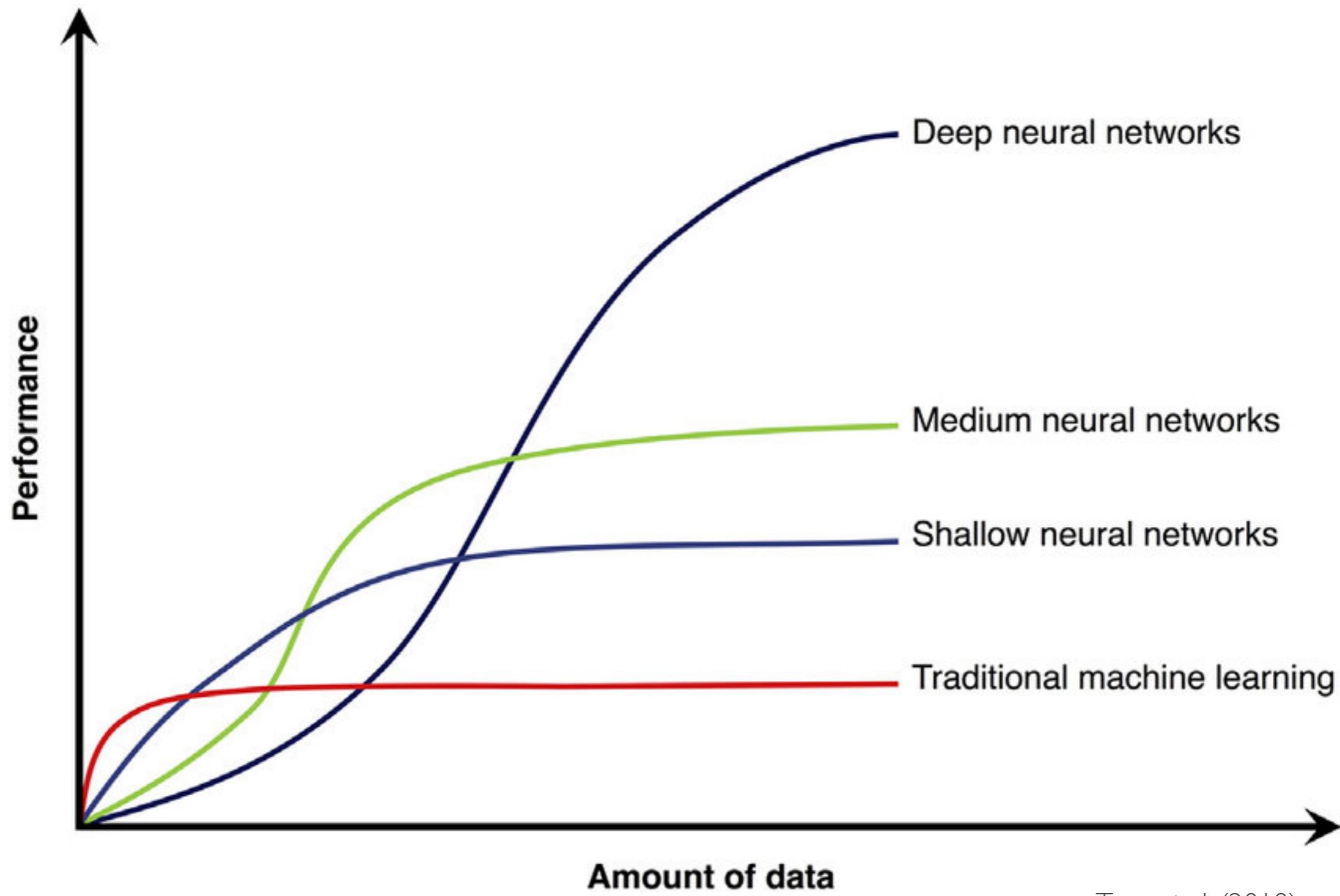
# BASIC RECIPE FOR DEEP LEARNING PROJECTS



# TRIAL AND ERROR

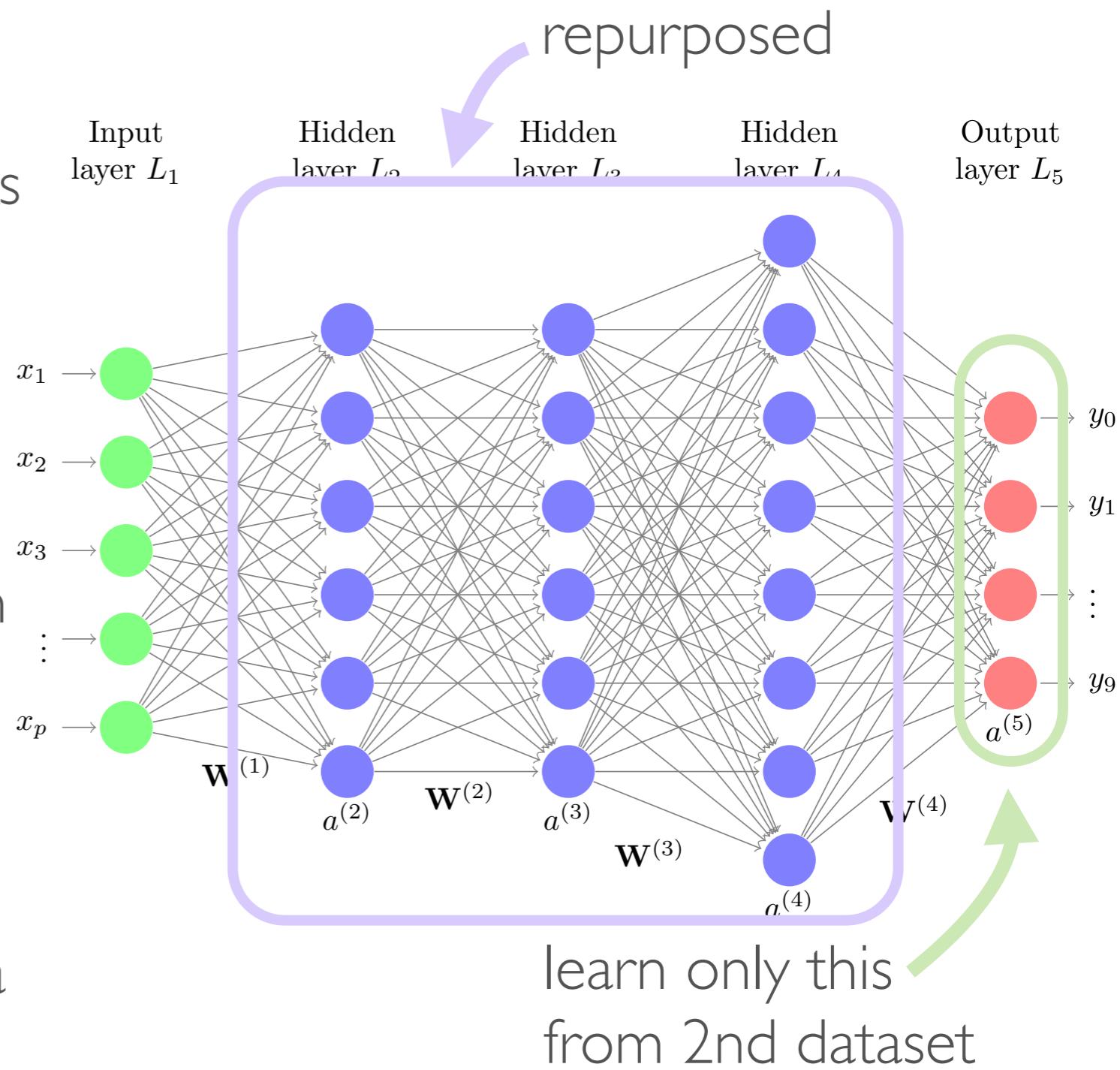
- Which regularization techniques to use?
- How many hidden layers and hidden units in each layer?
- What are good learning rates in gradient descent?
- What are good mini-batch sizes for SGD?
- Does transfer learning work?
- Either end-to-end DL or a series of smaller models?

# SCALE DRIVES DEEP LEARNING



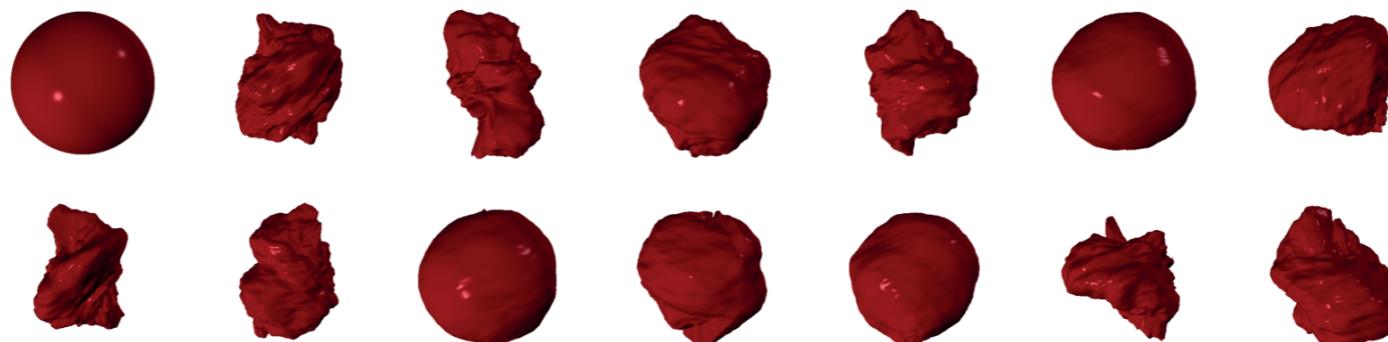
# TRANSFER LEARNING

- Some intermediate representations are useful for several *related but different* tasks
  - e.g. economic data shared for predicting apple price and orange price
- A representation learned from one dataset could be used with another dataset
  - e.g. learn from abundant data in one country and transfer it to another country with much less data

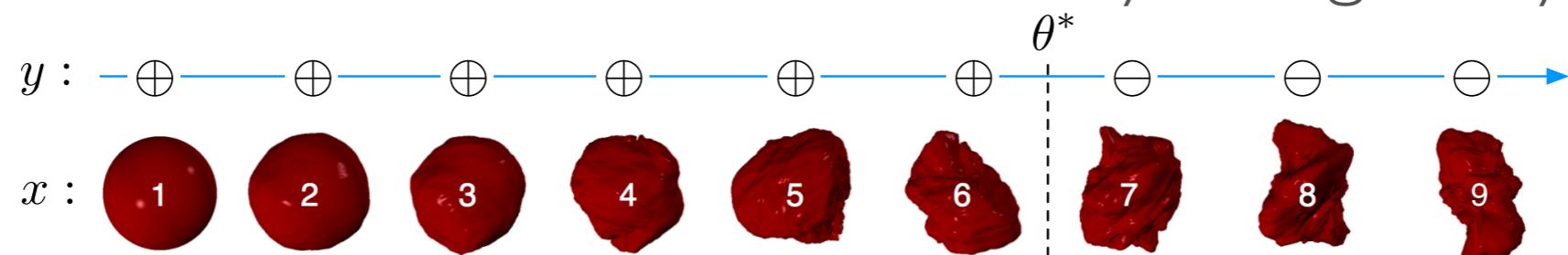


# WHY SMALL DATA IS OK

Imagine a situation where you want to classify a totally exotic fruit into edible or noxious. It comes in a smooth, round, bumpy or just so irregular shape of different sizes with various colors.



But, from the past experience on different fruits, you know the right representation — what matters is only “irregularity”.



To learn a good classifier, how many samples at best you need?

# END-TO-END DEEP LEARNING

- Direct mapping of  $x$  to  $y$  with a single neural network
- Pros
  - Simplifying the model
  - Letting the data speak without subjective judgments that limit the model's performance
- Cons
  - Large data requirement
  - Ignoring potentially useful domain knowledge
- DL is so powerful that it is tempting to do direct mapping

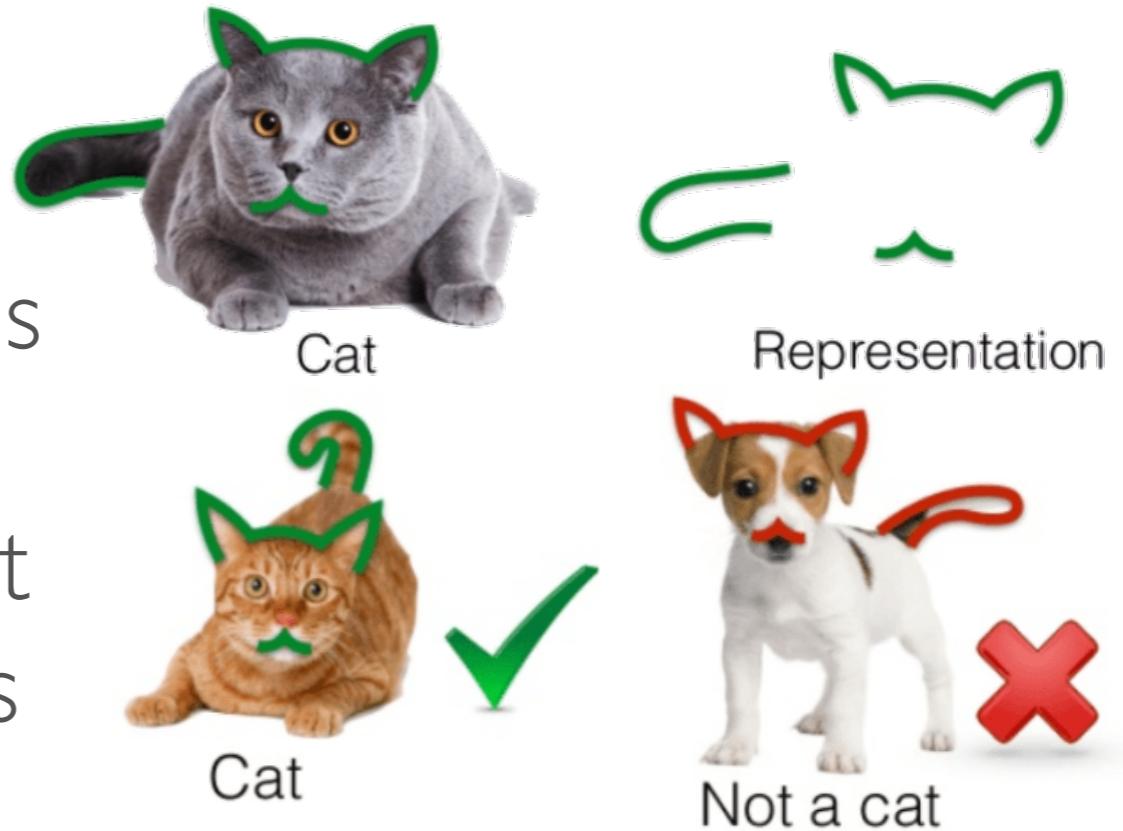
# END-TO-END OR NOT?

- Some systems are too complex even for DL or we do not have enough data:
  - e.g. self-driving car mapping remote-sensing data ( $x$ ) to {accelerate, brake, steer} decisions ( $y$ )
  - e.g. predicting poverty ( $y$ ) from satellite image ( $x$ )
- Depends on whether we have:
  - Sufficient data to learn such complex systems
  - Quality of domain knowledge



# CONVOLUTIONAL NEURAL NETWORK (CNN)

- One of the most successful DL models
  - especially for image recognition
- Highlighting representation learning and the architectural nature of DL
- Successively learns representations from raw pixels ( $x$ ) to simple features to more abstract ones. Based on the last representation, then, it classifies the image ( $y$ ).



# DEEP LEARNING LIBRARIES

- Programming platforms to implement DL
- Nobody writes codes from scratch for applying DL these days, just like we don't use Assembly to implement linear regression.
- Remember the trial-and-error model building. Fast implementation is very important.
- TensorFlow, PyTorch, Keras, etc. Ask Siraj Raval and find your best friends.

# DEMO / PRACTICE

- We use Google Colab + PyTorch
- Go to
  - <https://yujisaikai.com/teaching/>
  - Find exercise 1, 2, and 3 links
- You may need Google account or Wisc account