AAE722 Machine Learning in Applied Economic Analysis

DEEP LEARNING

Agricultural & Applied Economics @ UW-Madison
Yuji Saikai

Many Pathways to Deep Learning

- Keras
  Quickest way to implement deep learning

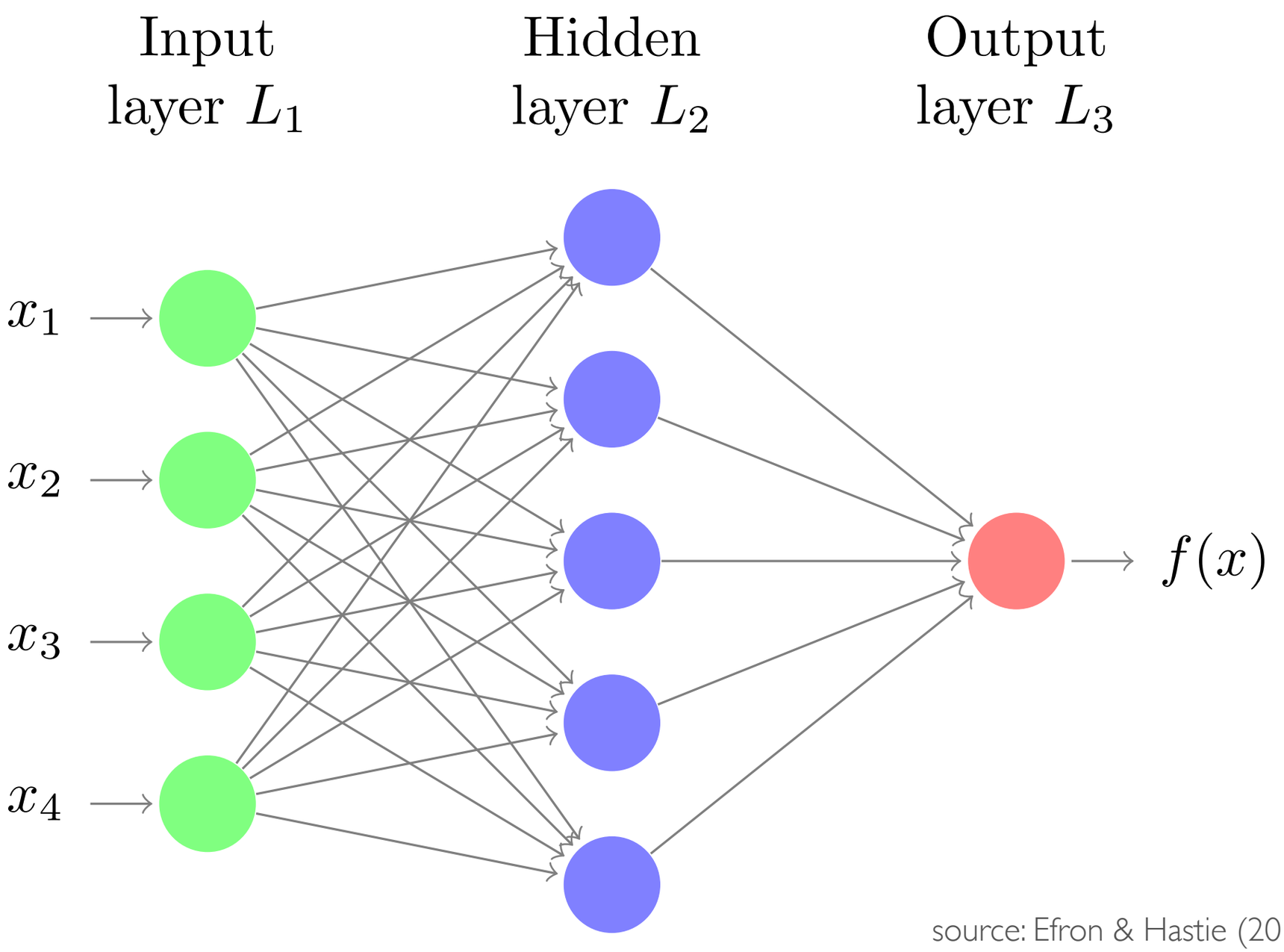
- Andrew Ng (Coursera)
  Foundations for practitioners

- Sebastian Raschka (STAT479, UW-Madison)
  More in-depth training for both research & practice

SUMMARY

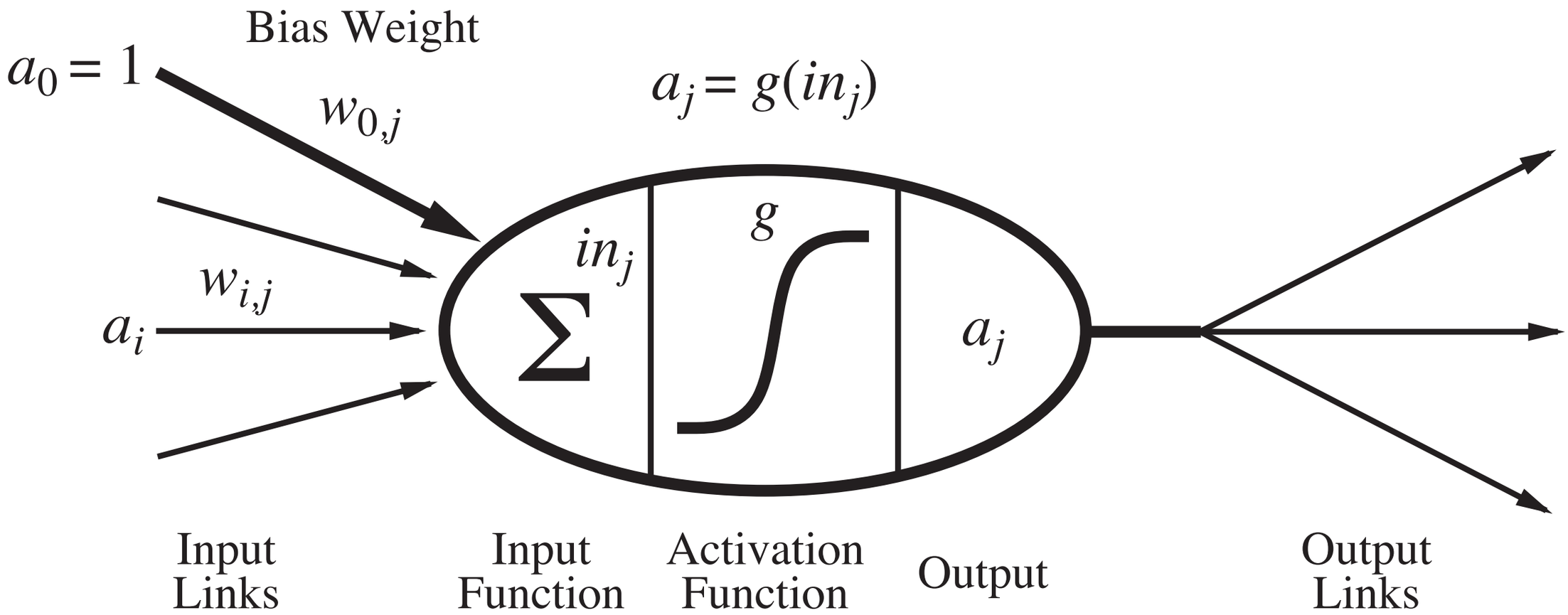
- Review of neural network
- Representation learning
- Architectural nature
- Applications
- Demos

Neural Network



source: Efron & Hastie (2016)

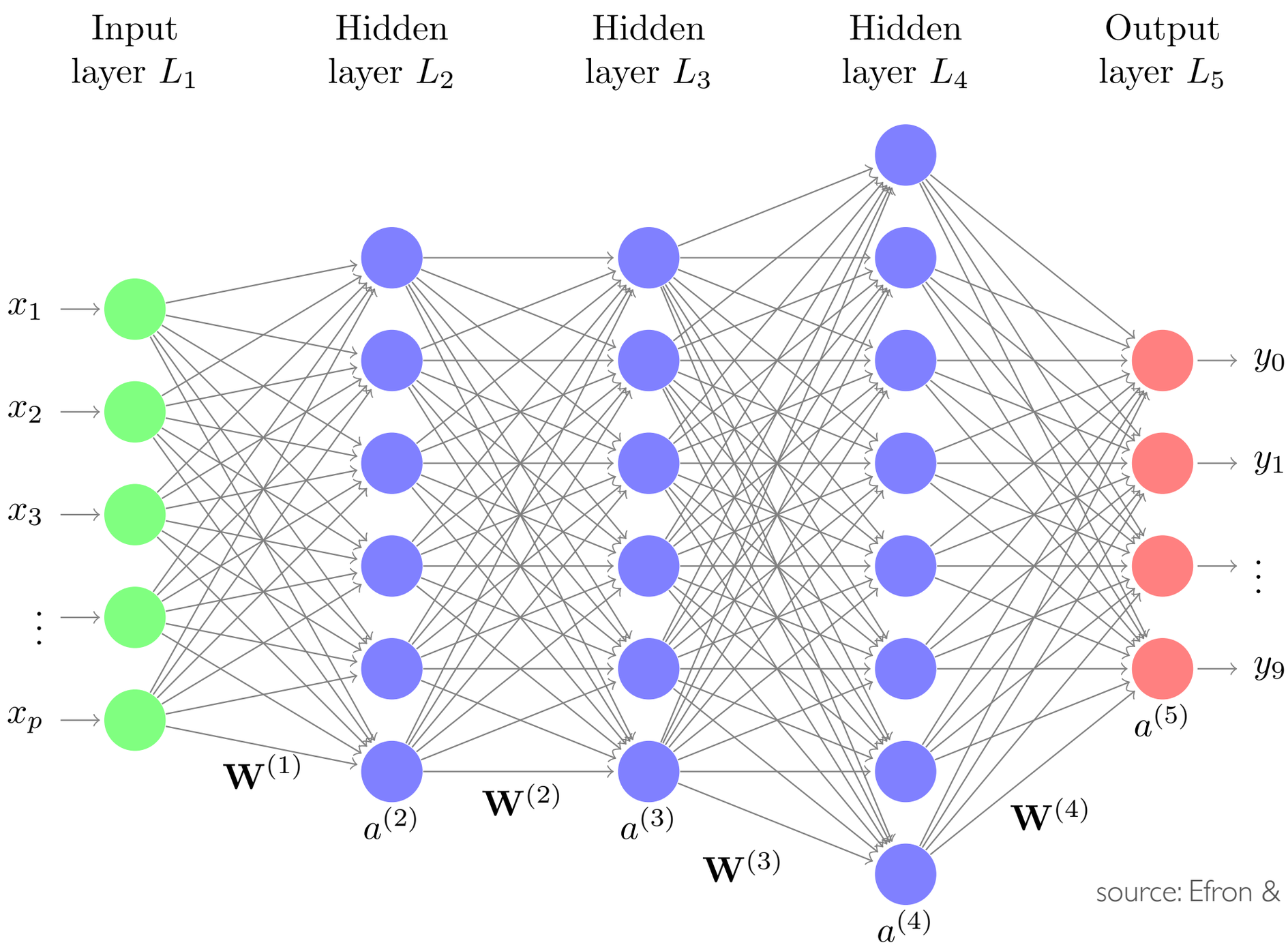
Computational Unit



source: Russell & Norvig (2010)

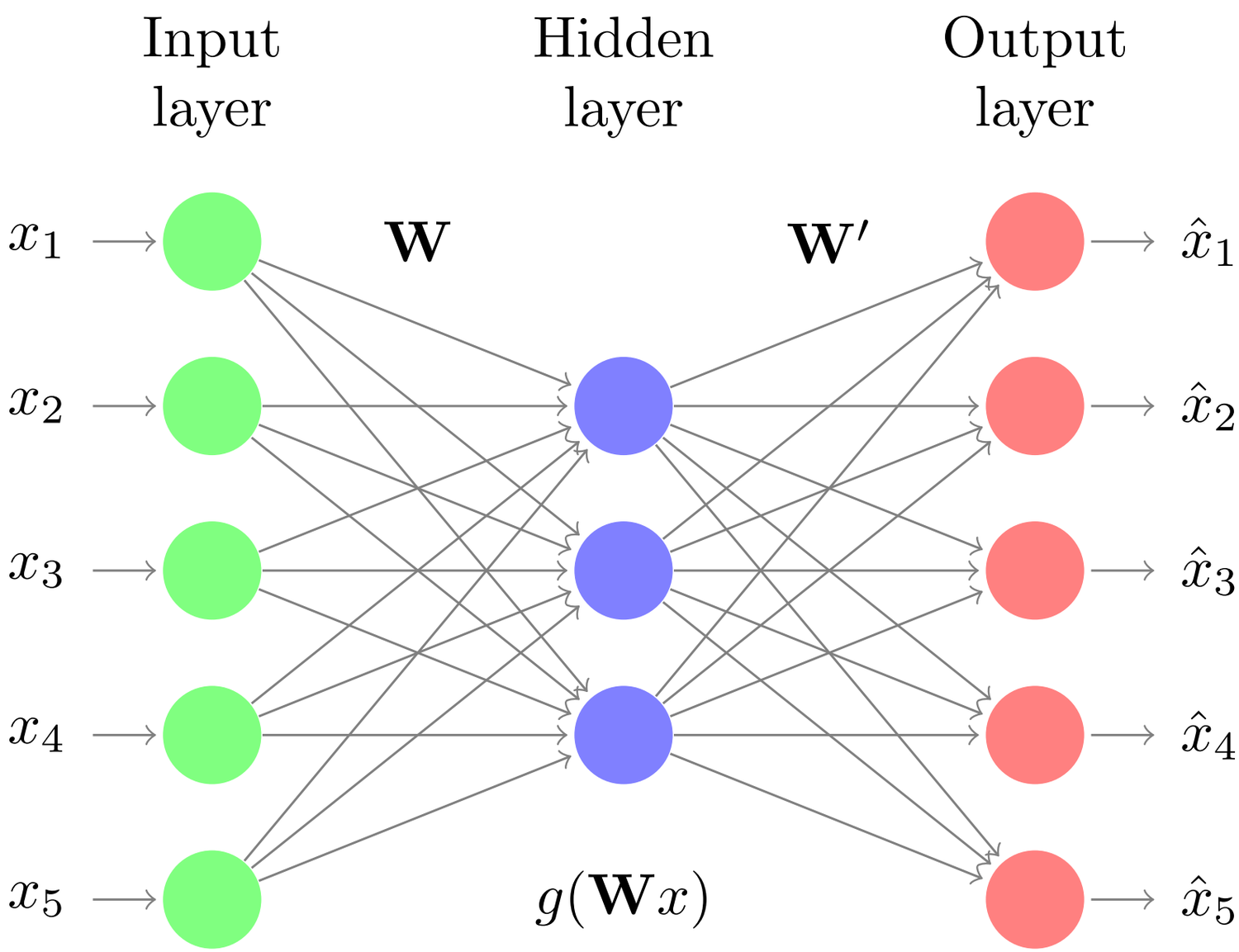
The most common $g(\ )$ is ReLU — $g(in_j) = max(0, in_j)$

Deeper Network & Multiple Outputs



source: Efron & Hastie (2016)

Autoencoder



source: Efron & Hastie (2016)

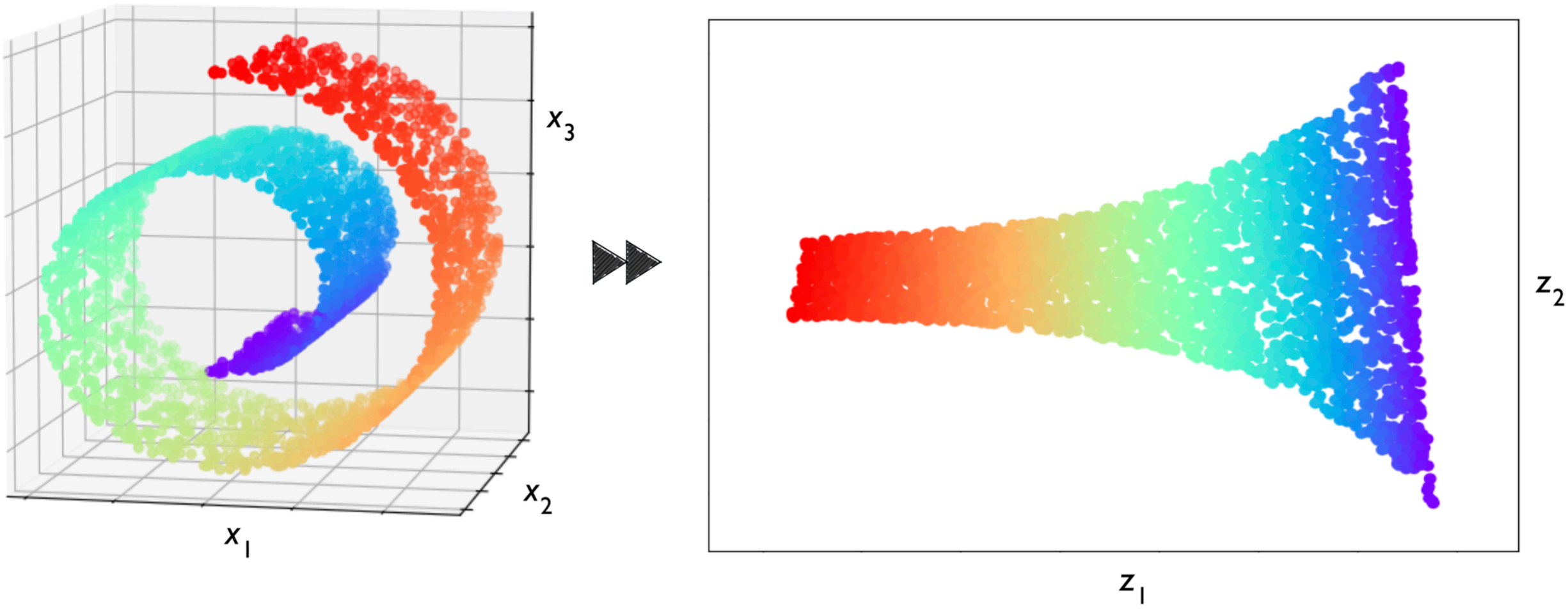
Representation Learning

- A key perspective to understand DL

- Some **representations** of data are more useful than others

- **Learning** such representations also from data, rather than handcrafting

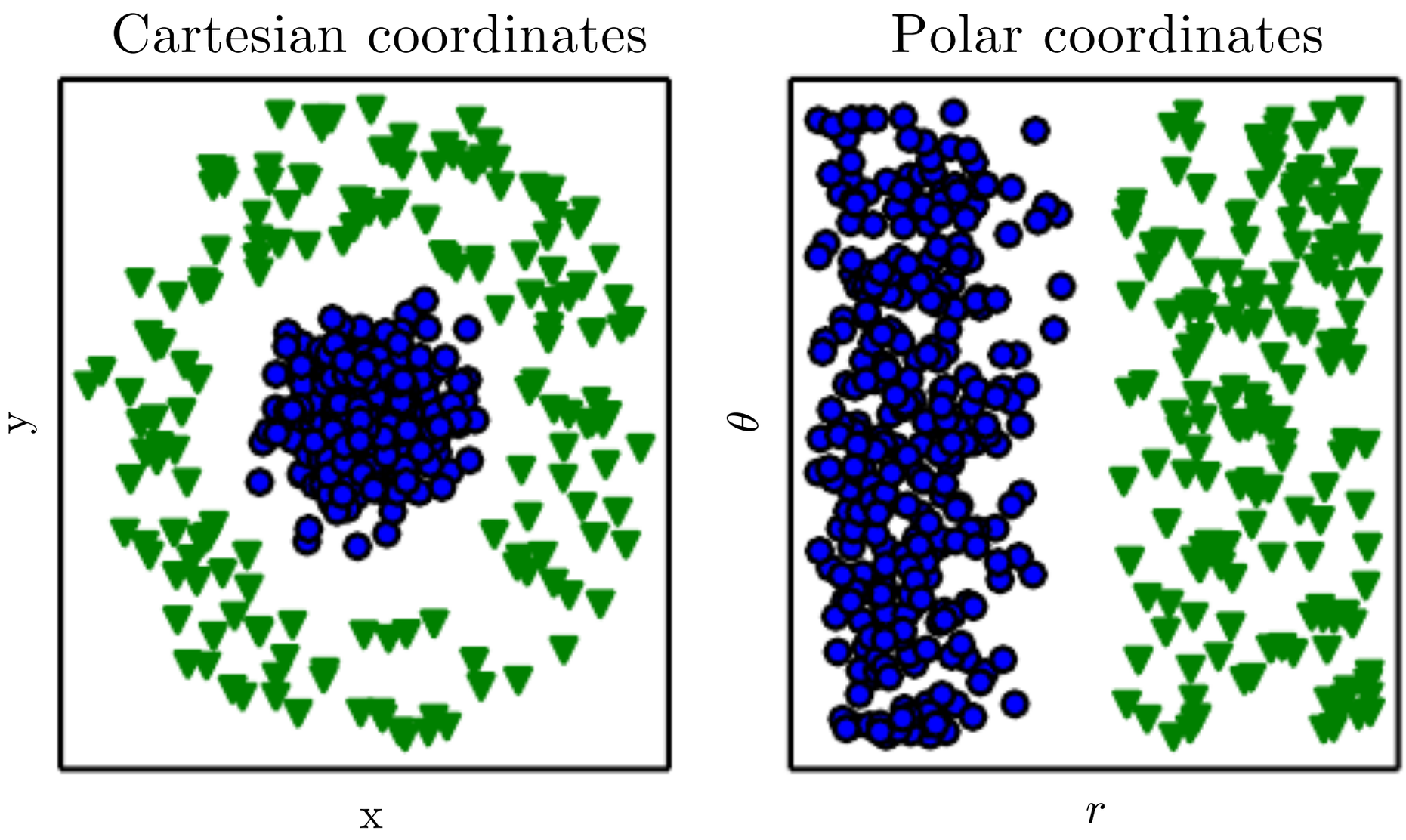
Representation Matters



Manifold learning using kernel PCA

source: Sebastian Raschka

Key to Understand Causality



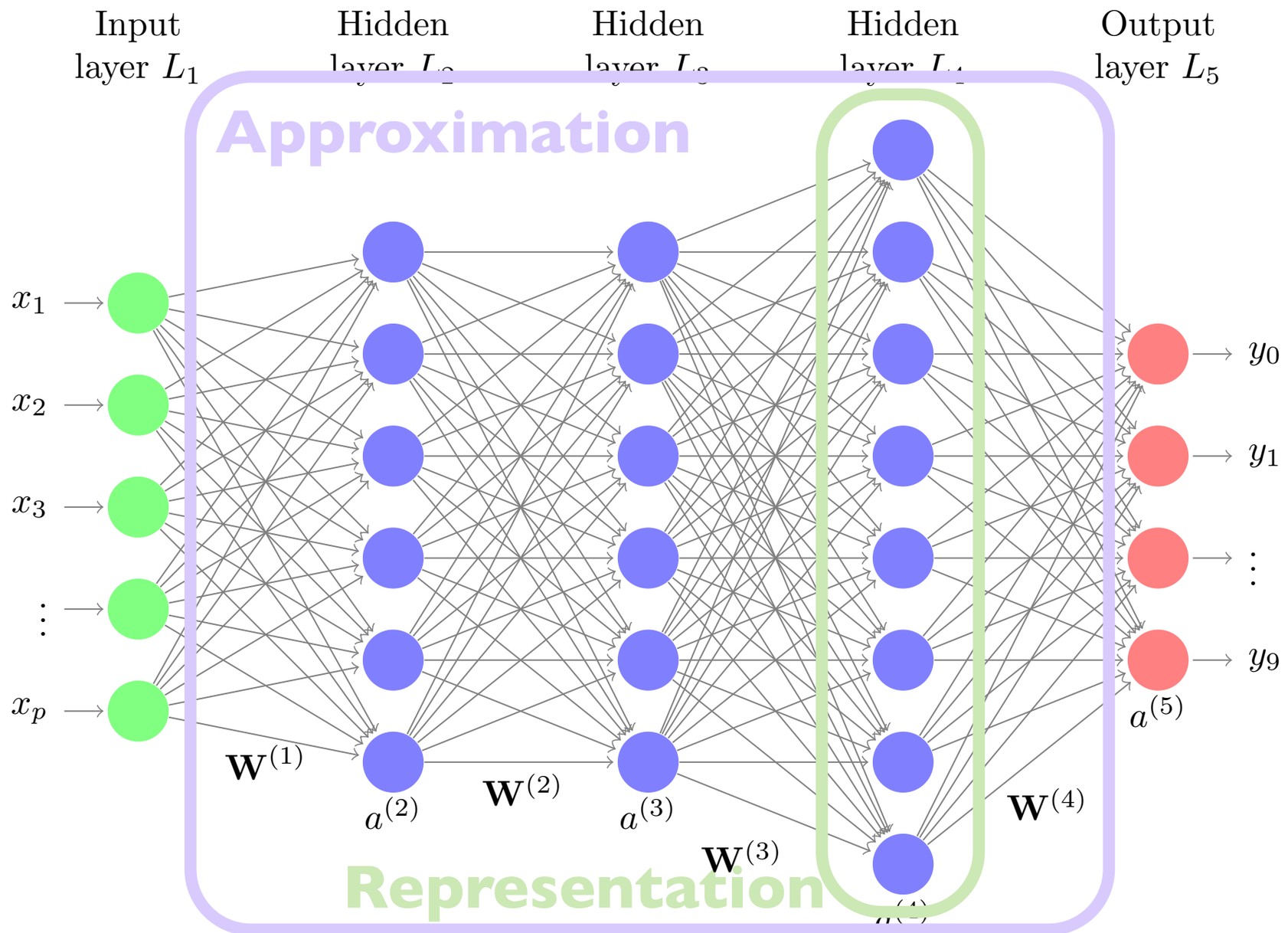
source: Goodfellow, Bengio, & Courville (2016)

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"?

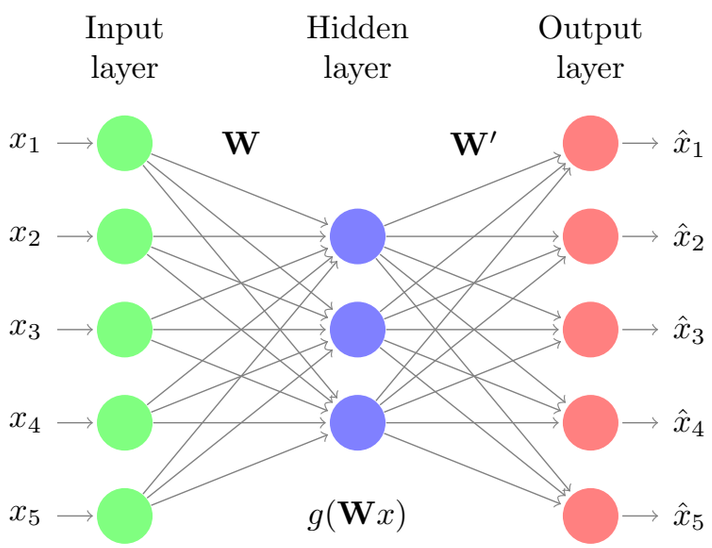
- "*many natural signals are compositional hierarchies, in which higher-level features are obtained by composing lower-level ones*" —LuCun, Bengio, & Hinton (2015)

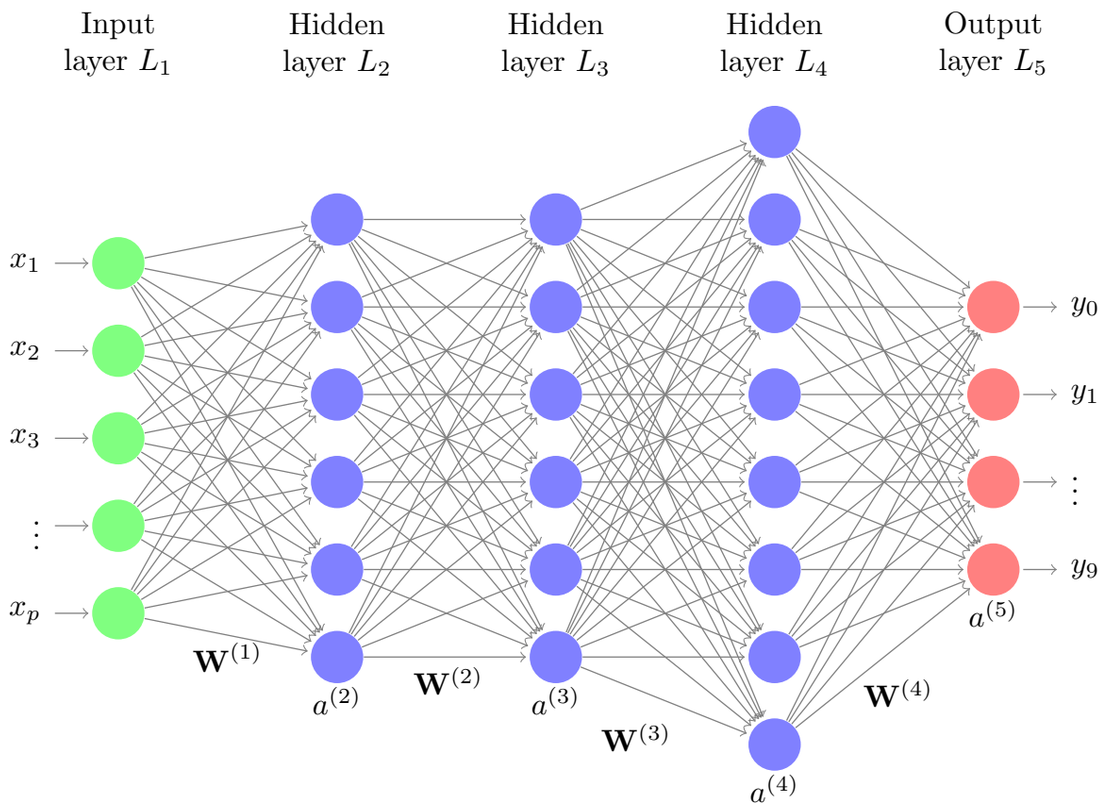
- shallow networks can "*require many more computational elements, potentially exponentially more*" —Bengio (2009)

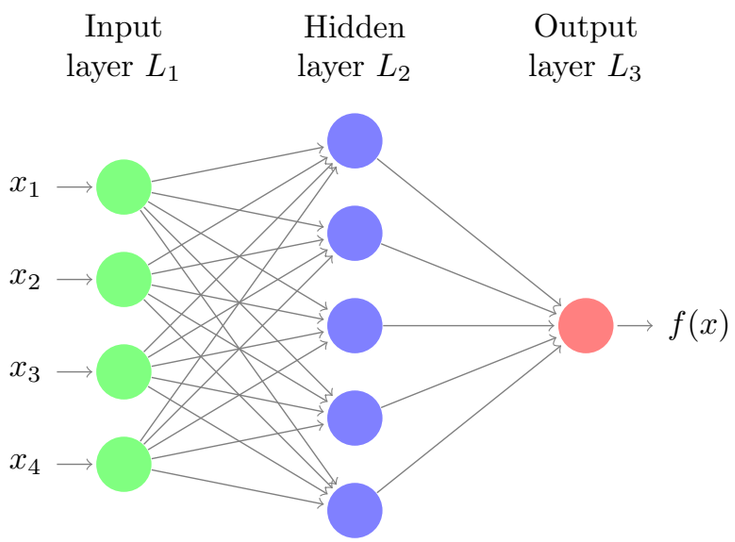
- "*the phrase is just a great brand, it's just so deep*" —Andrew Ng

More Art than Science

so many "architectures"

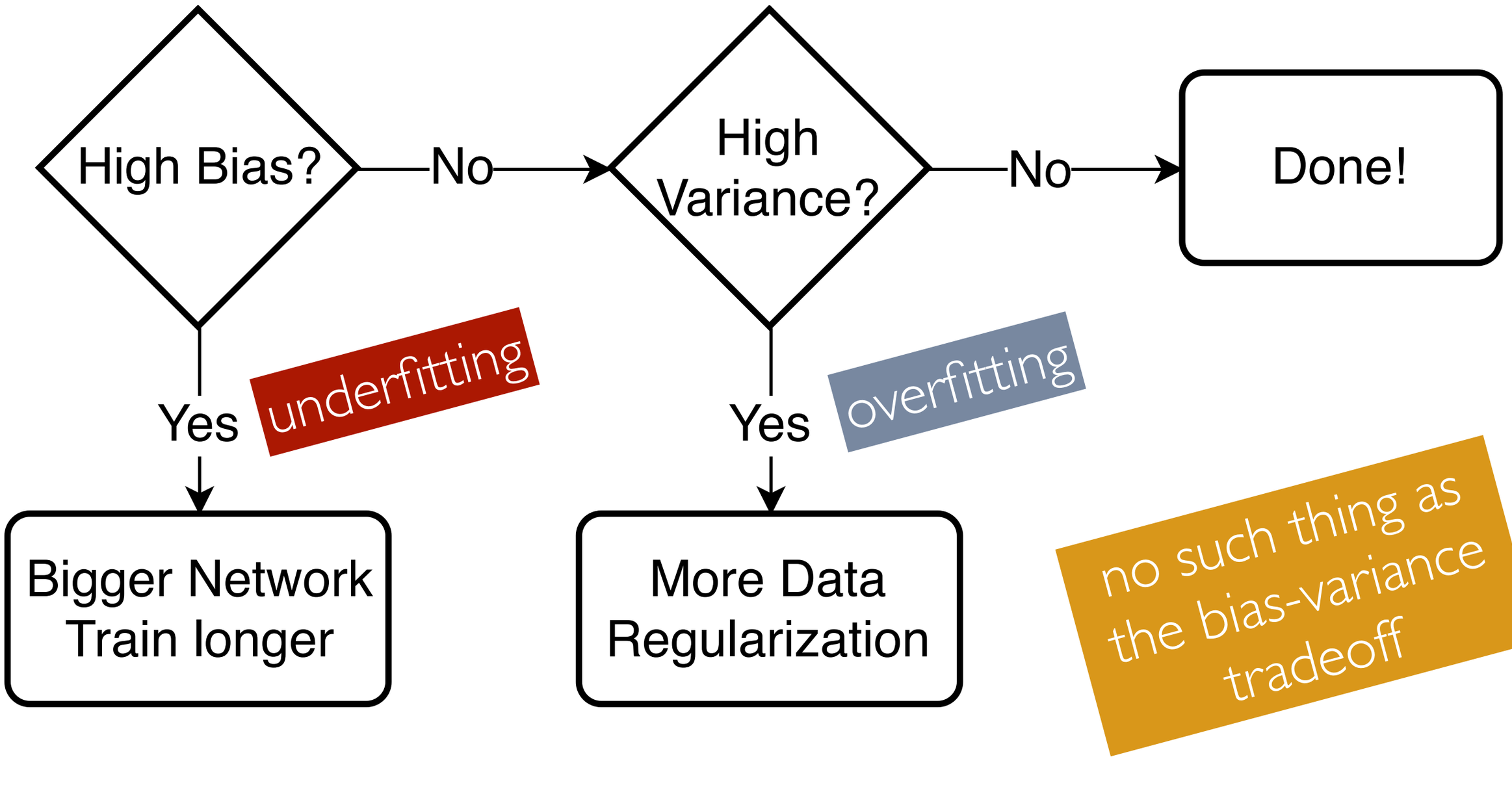






There is a basic recipe but no hard-and-fast rule!

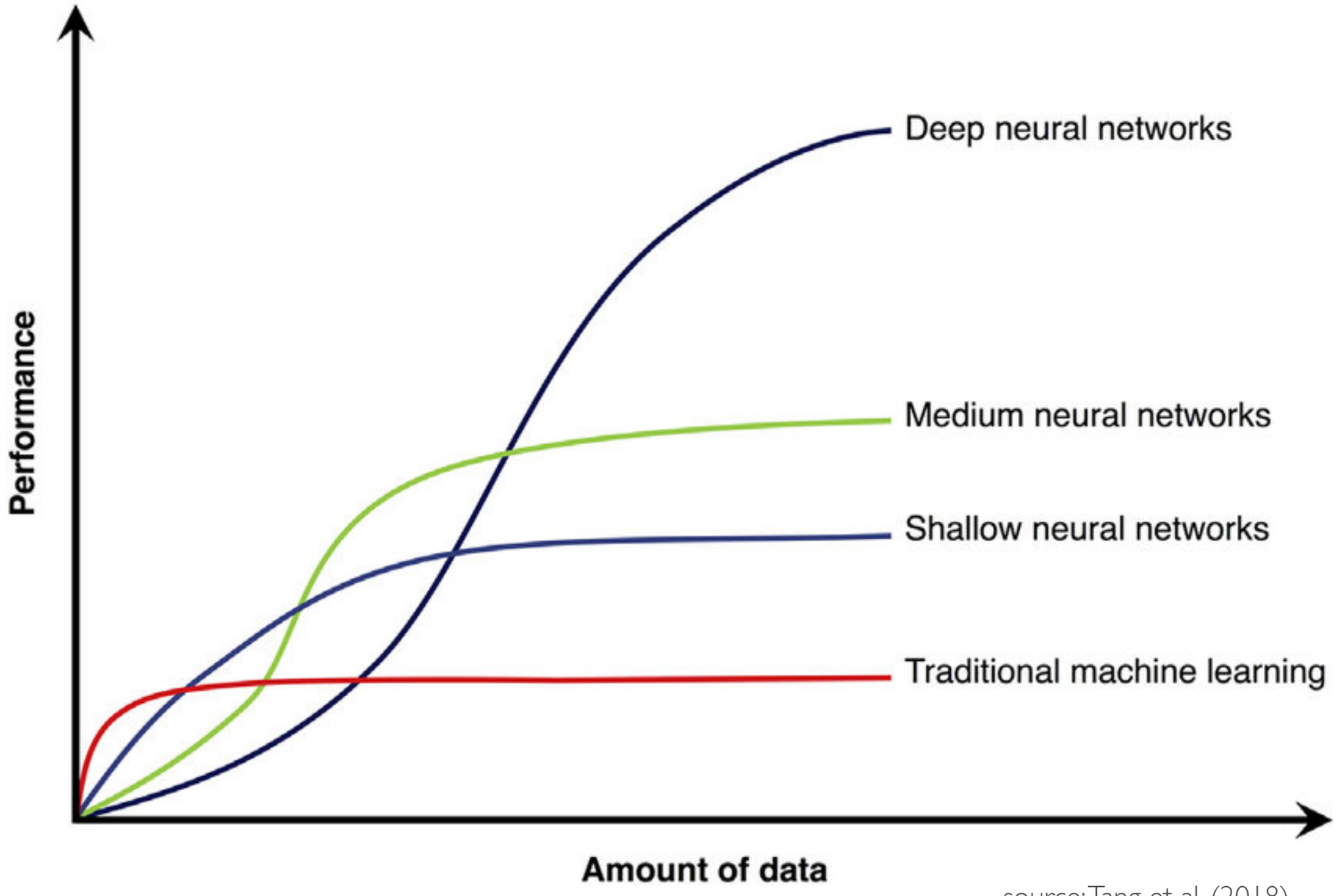
Basic Recipe for Deep Learning Projects



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?

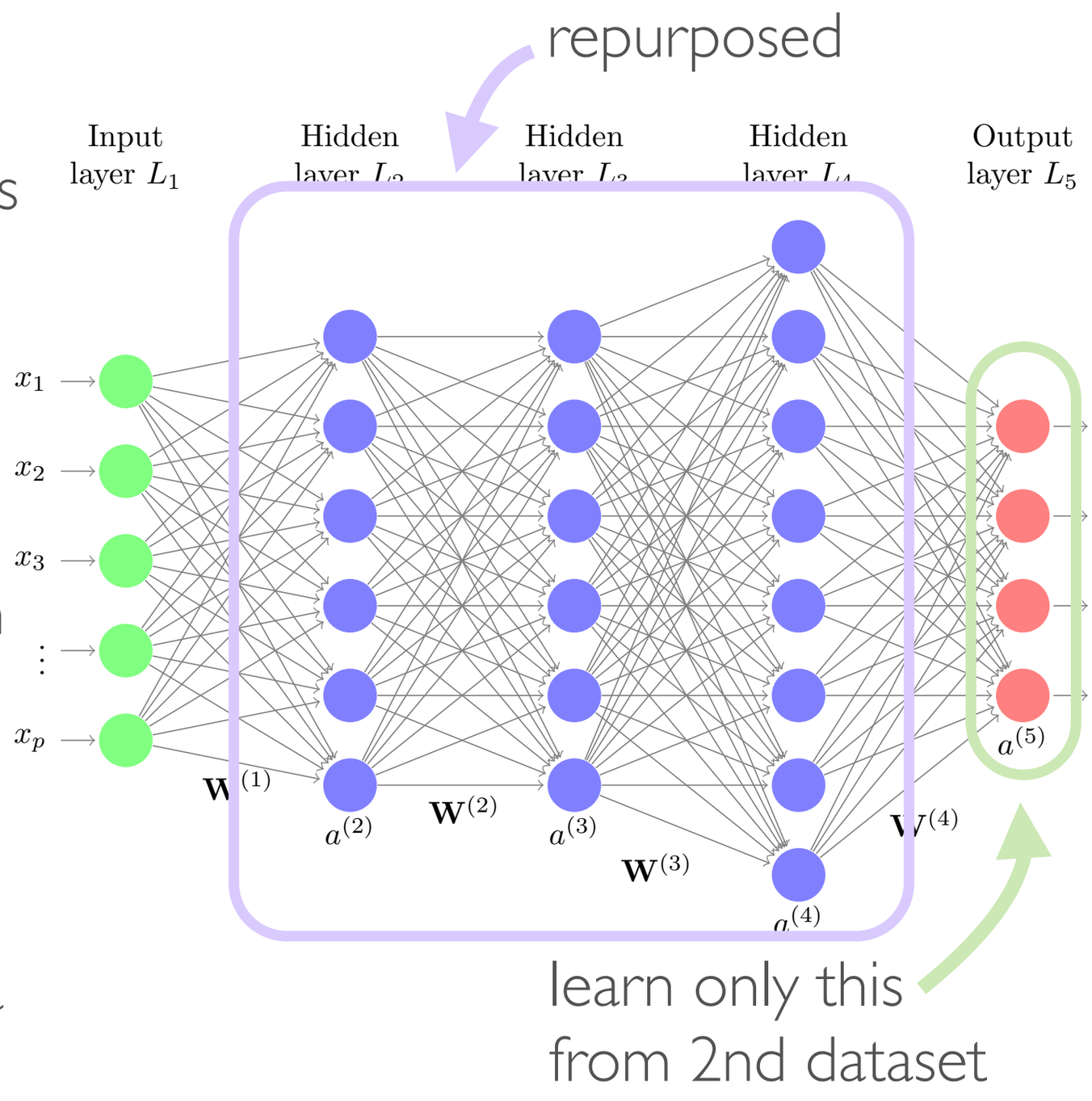
SCALE DRIVES DEEP LEARNING



source: Tang et al. (2018)

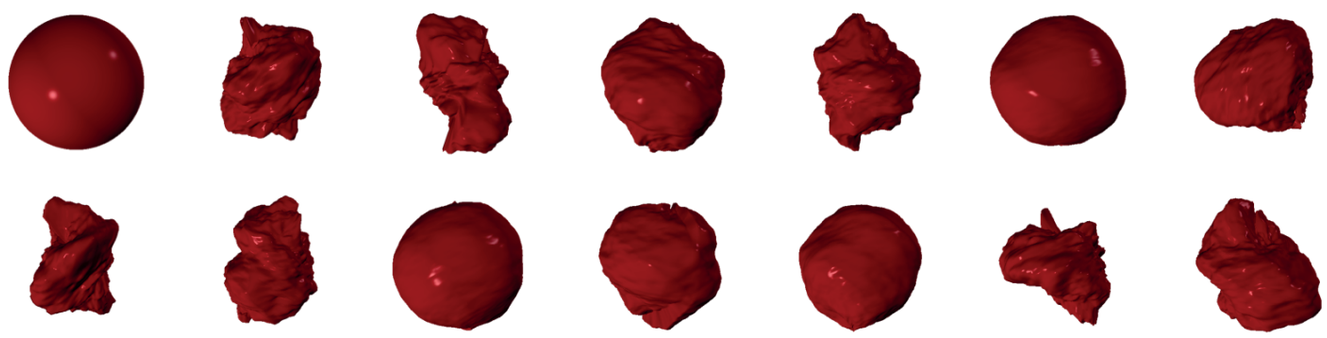
Transfer Learning

- 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

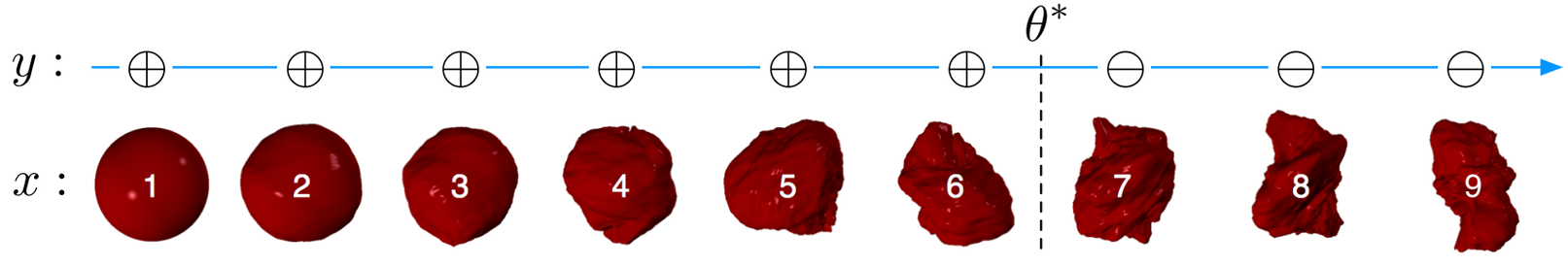


Why Small Data Is OK

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?

source: Settles (2012)

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

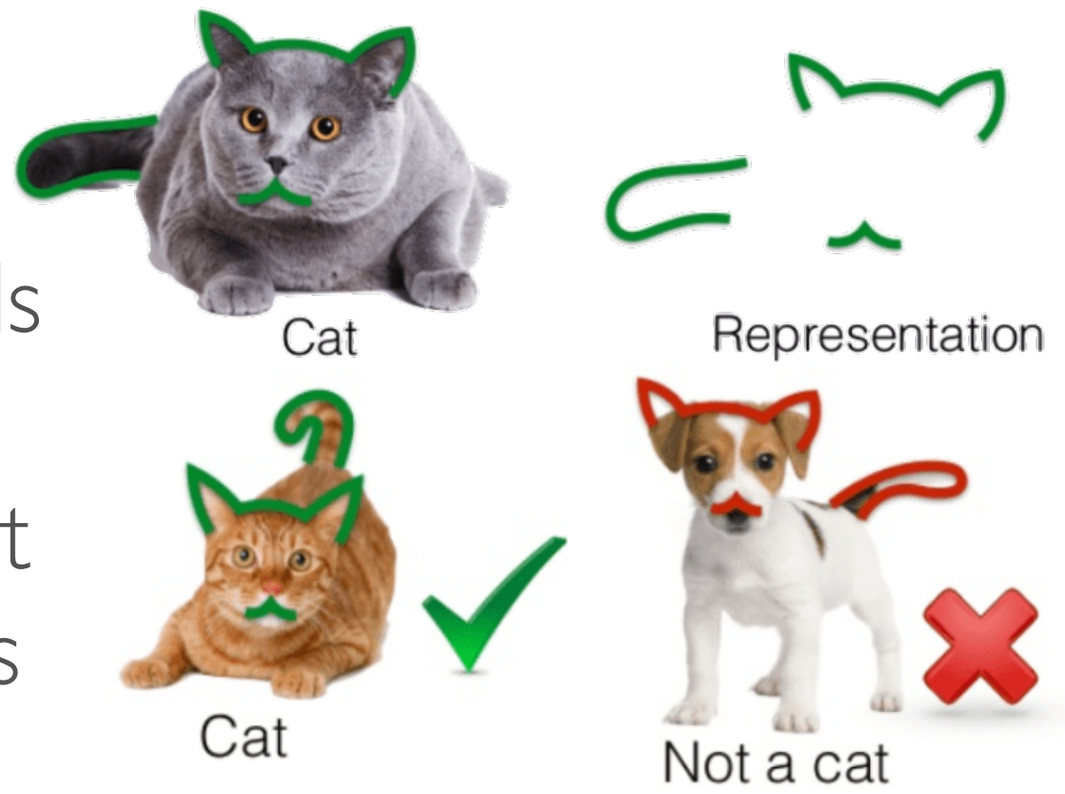
End-To-End or Not?

- 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).



source: Sefik Ilkin Serengil

Deep Learning Libraries

- Programing 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/
  - Open exercise 1, 2, or 3 links
  - press Open in playground at the top-left corner
  - (You may need Google account or Wisc account)