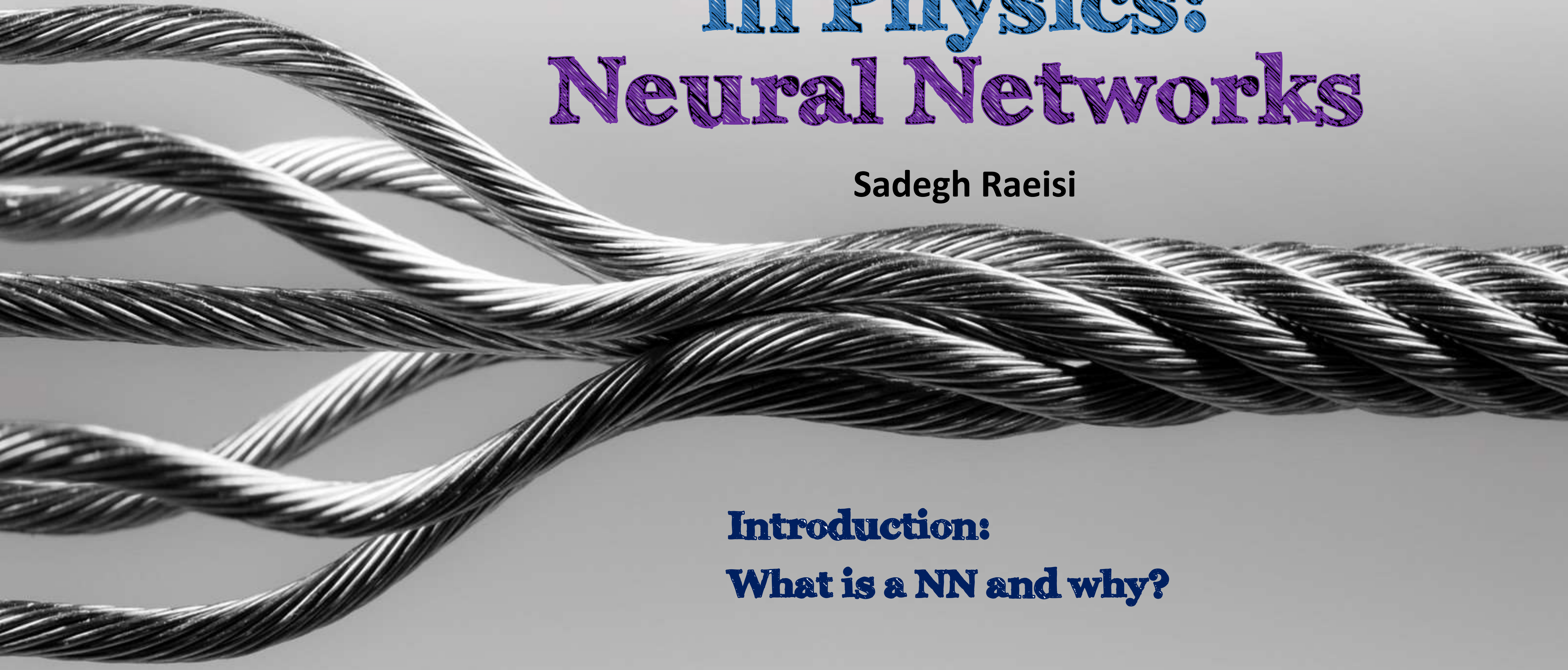
Machine Learning in Physics: Neural Networks

Sadegh Raeisi



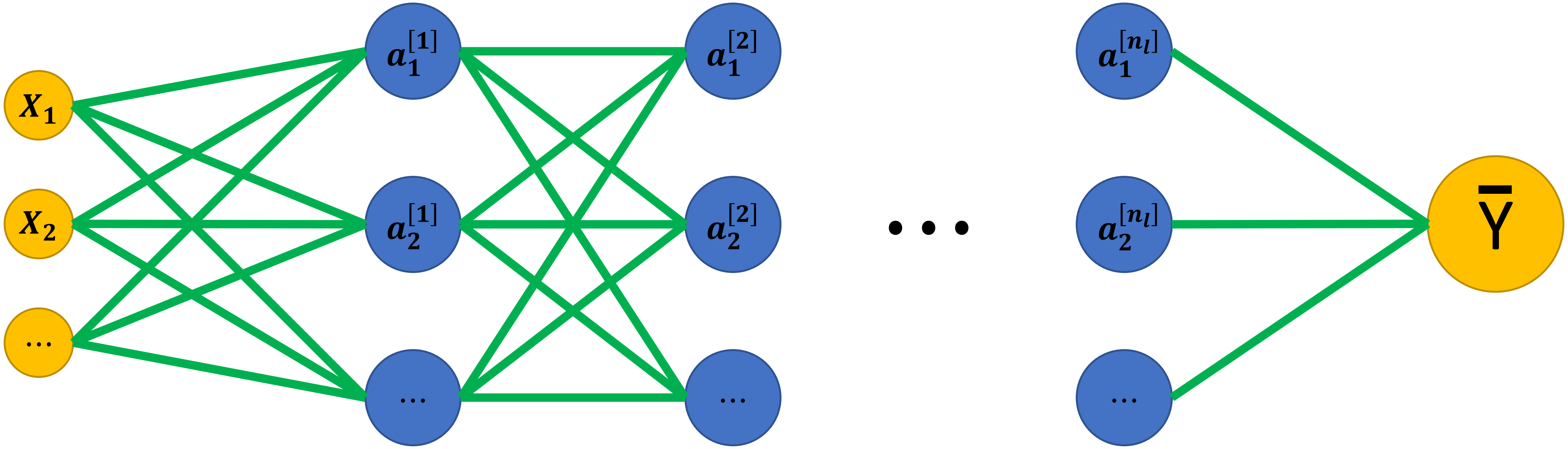
Introduction:
What is a NN and why?

Plan for the next part

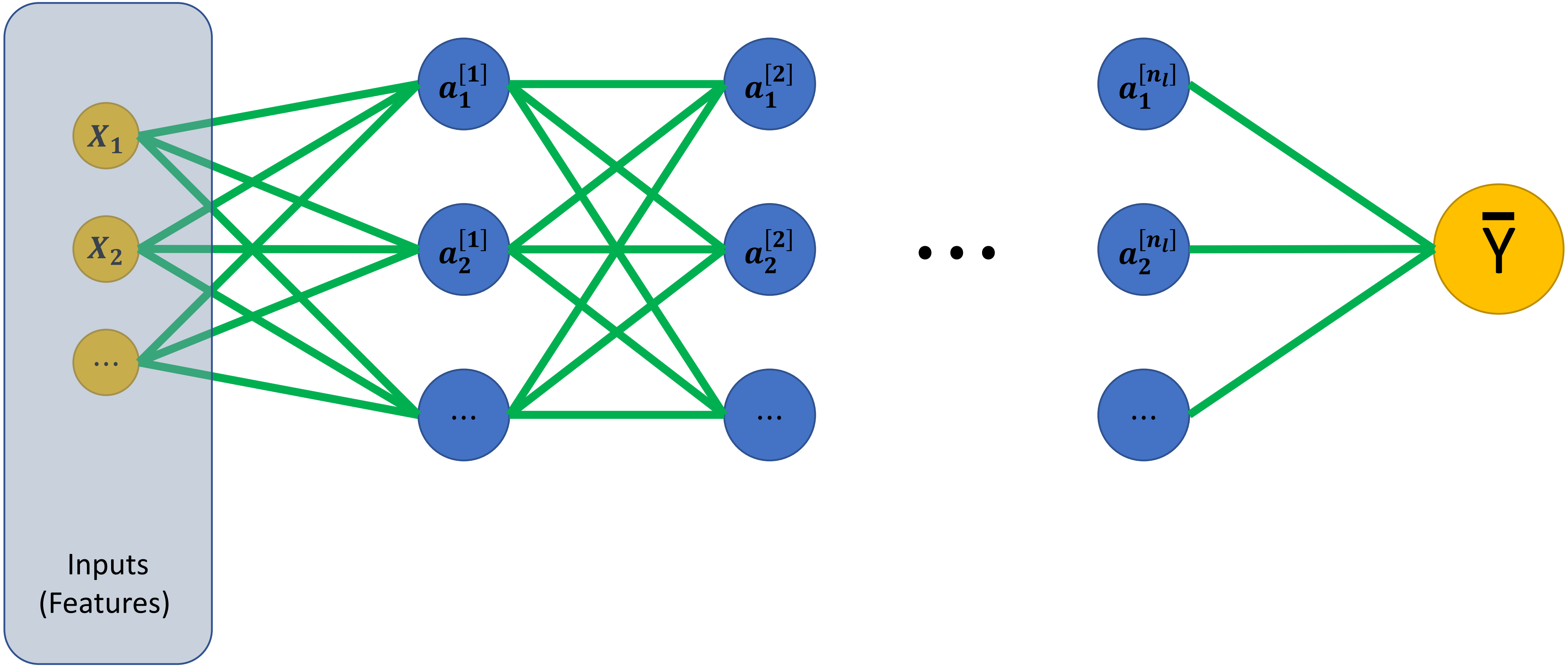
- Intro to NN
  - Structure
  - Intuition
  - Universality of NN
- Back propagation
- Hands-on example
- Convolutional NN
- Recurrent NN
- …

What is a NN?

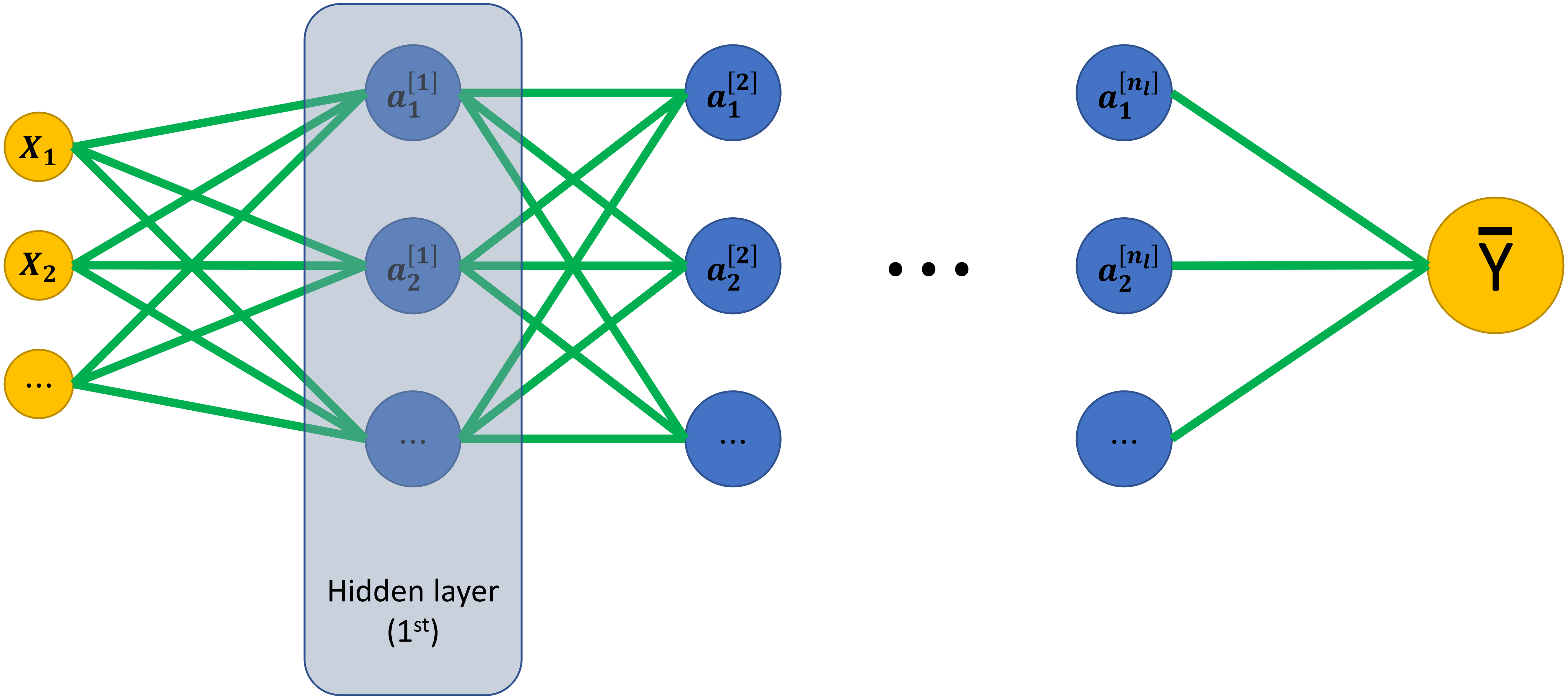
Structure and terminology



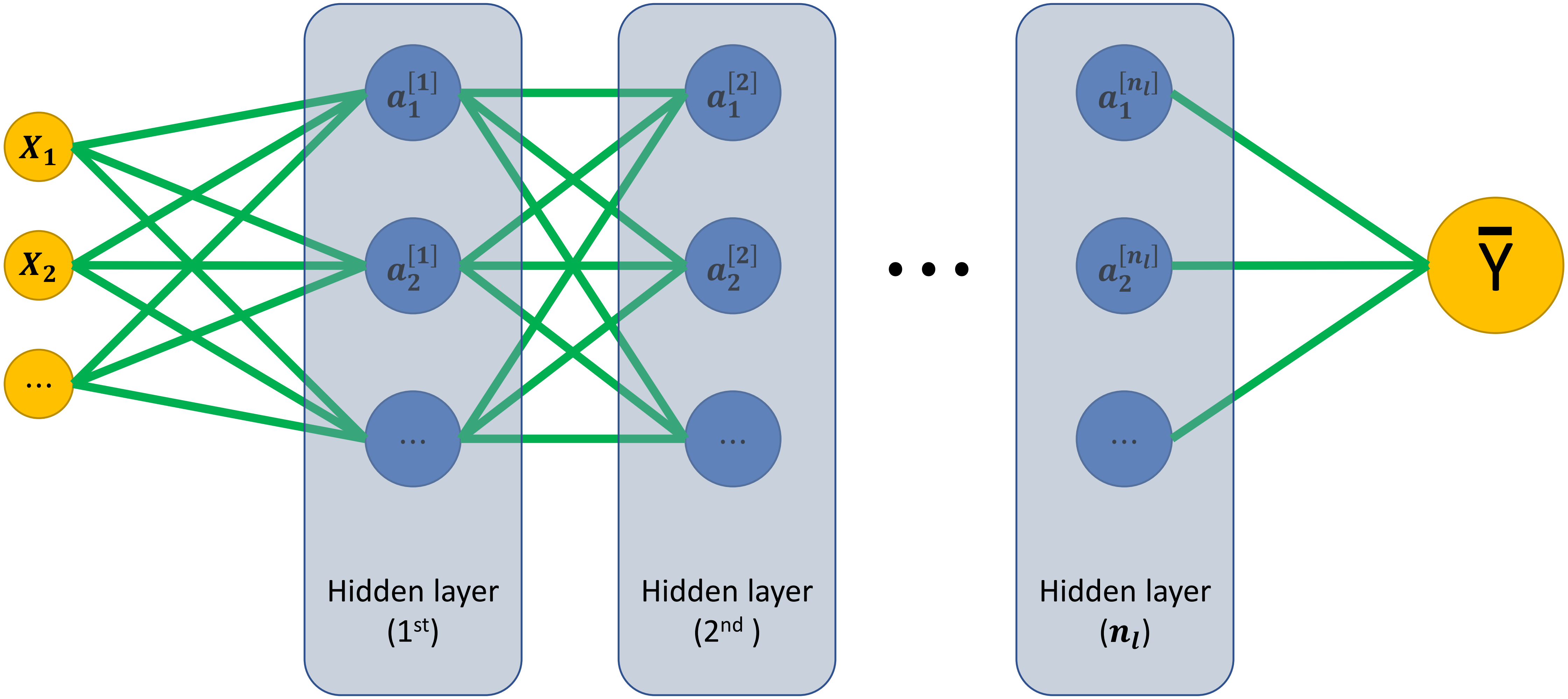
Structure and terminology



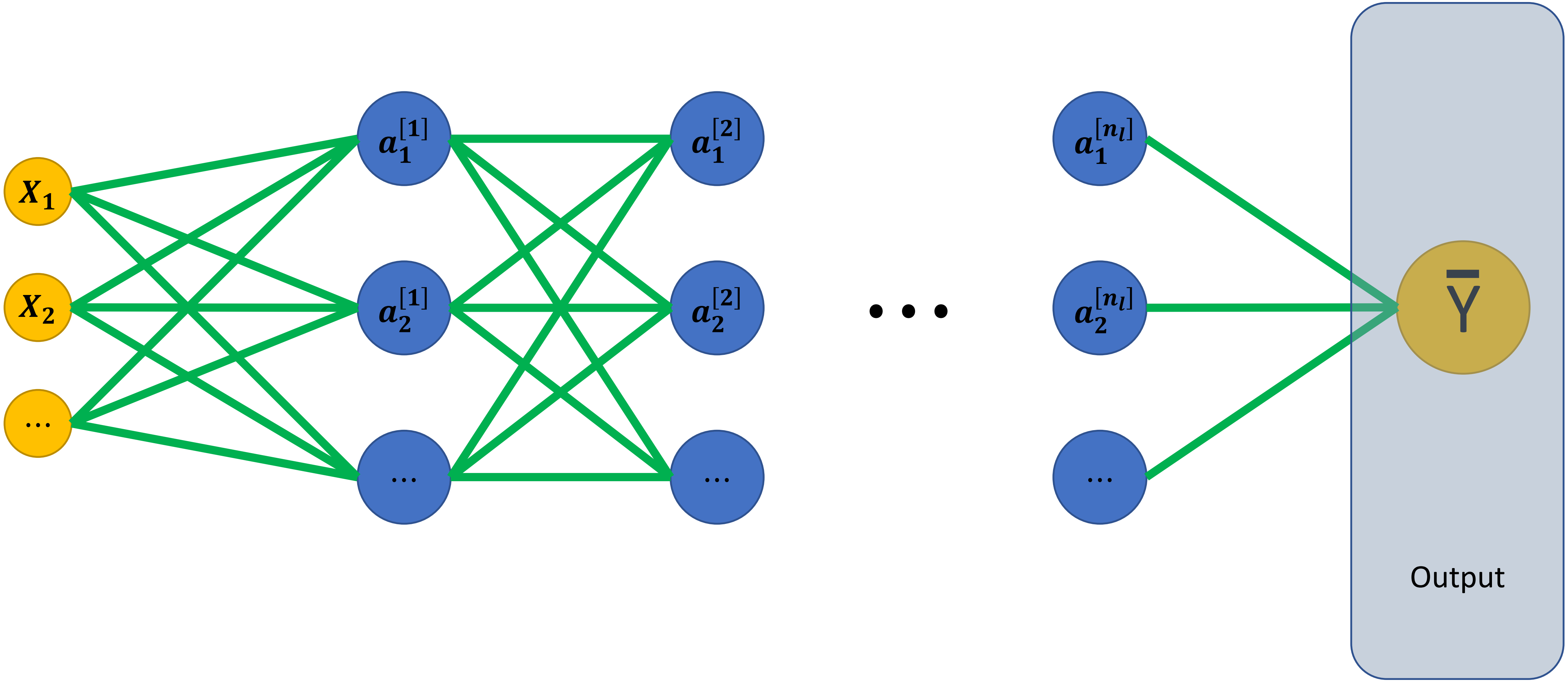
Structure and terminology



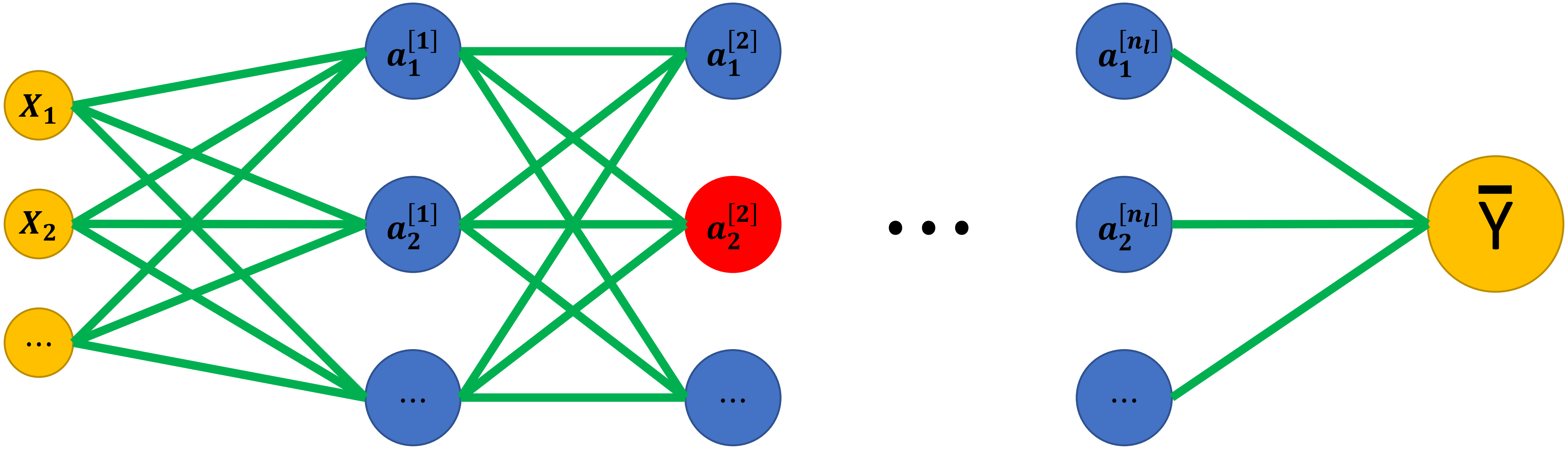
Structure and terminology



Structure and terminology

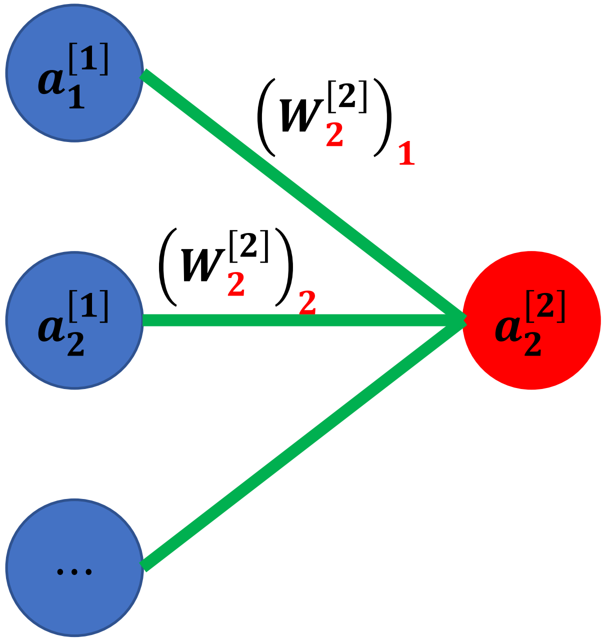


Structure and terminology



A node

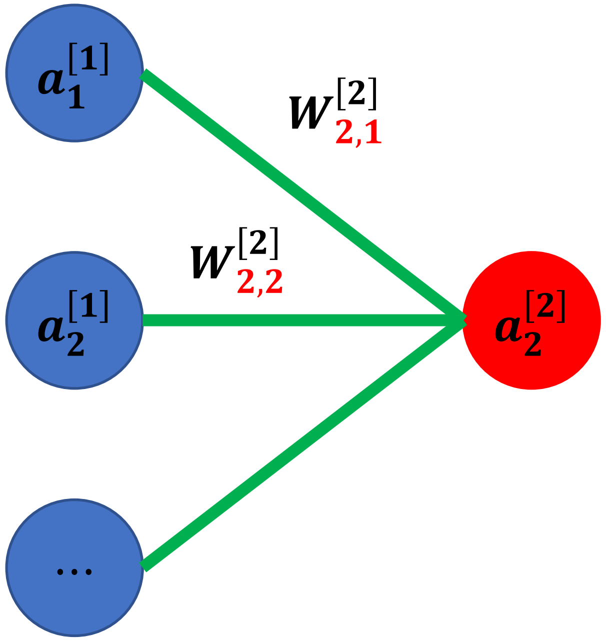
Structure and terminology



$$z_2^{[2]} = \left(W_2^{[2]}\right)_1 a_1^{[1]} + \left(W_2^{[2]}\right)_2 a_2^{[1]} + \cdots$$

$$a_2^{[2]} = f_{act}\left(z_2^{[2]}\right)$$

Structure and terminology



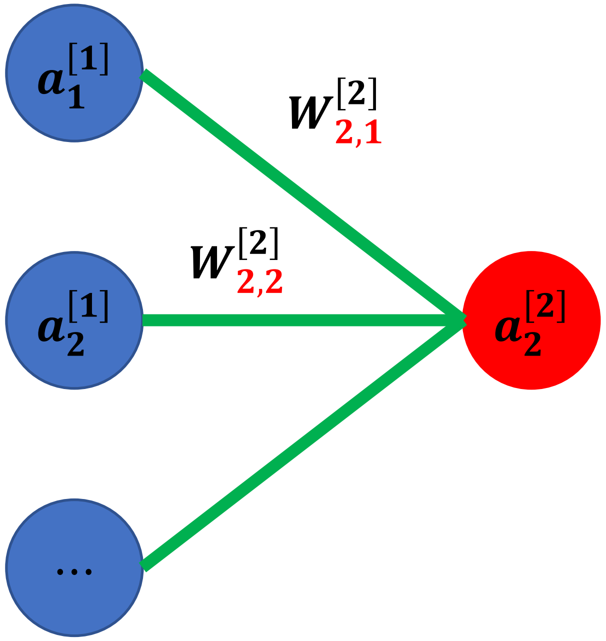
$$z_2^{[2]} = W_{2,1}^{[2]} \; a_1^{[1]}$$

$$+W_{2,2}^{[2]} \; a_2^{[1]}$$

$$+\cdots$$

$$a_2^{[2]} = f_{act}\left(z_2^{[2]}\right)$$

Structure and terminology



A node in layer l

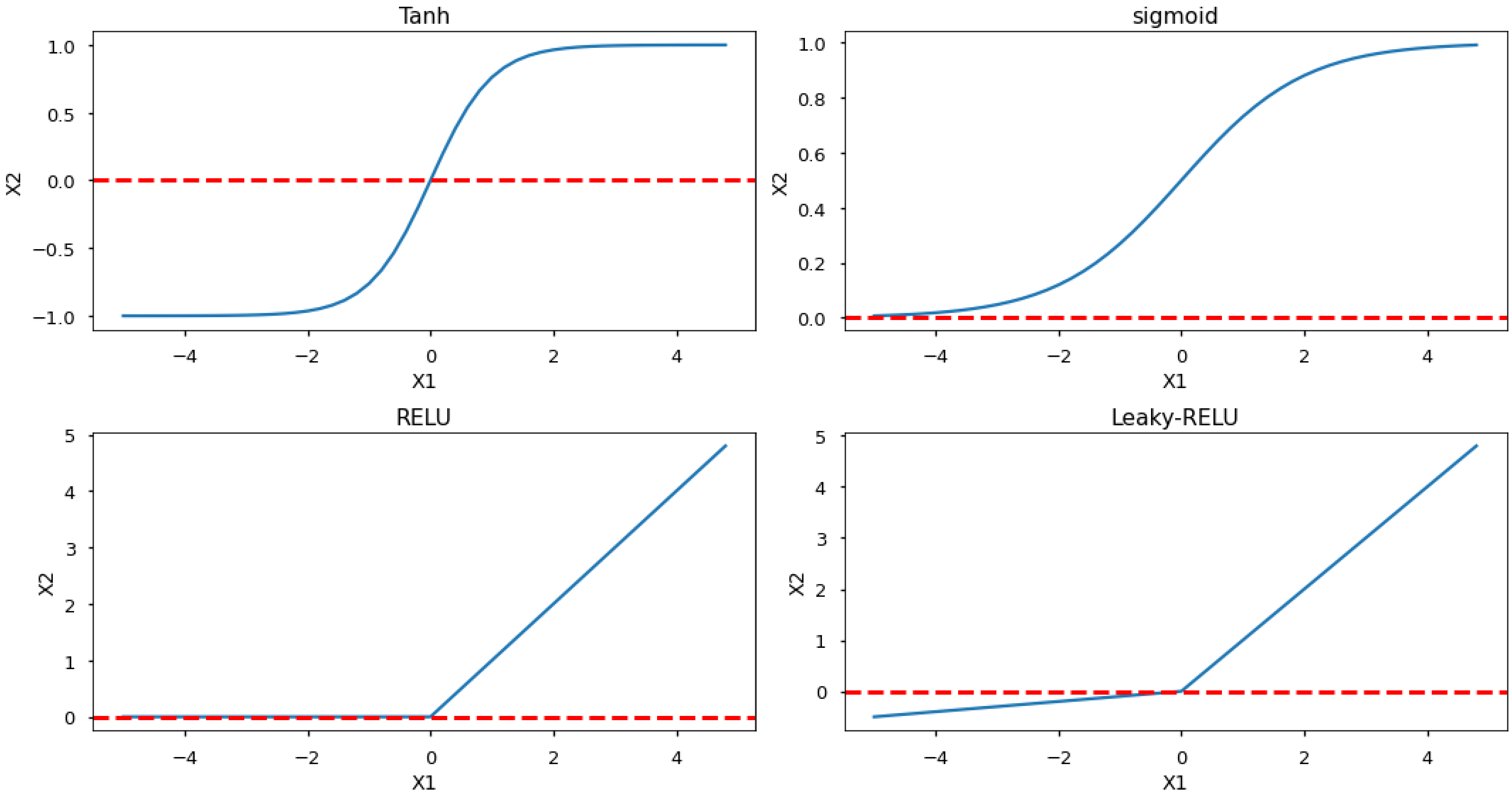
Linear combination of previous nodes

$$a_i^{[l]} = f_{act}^{[l]}\left(\sum_j W_{i,j}^{[l]} a_j^{[l-1]}\right)$$

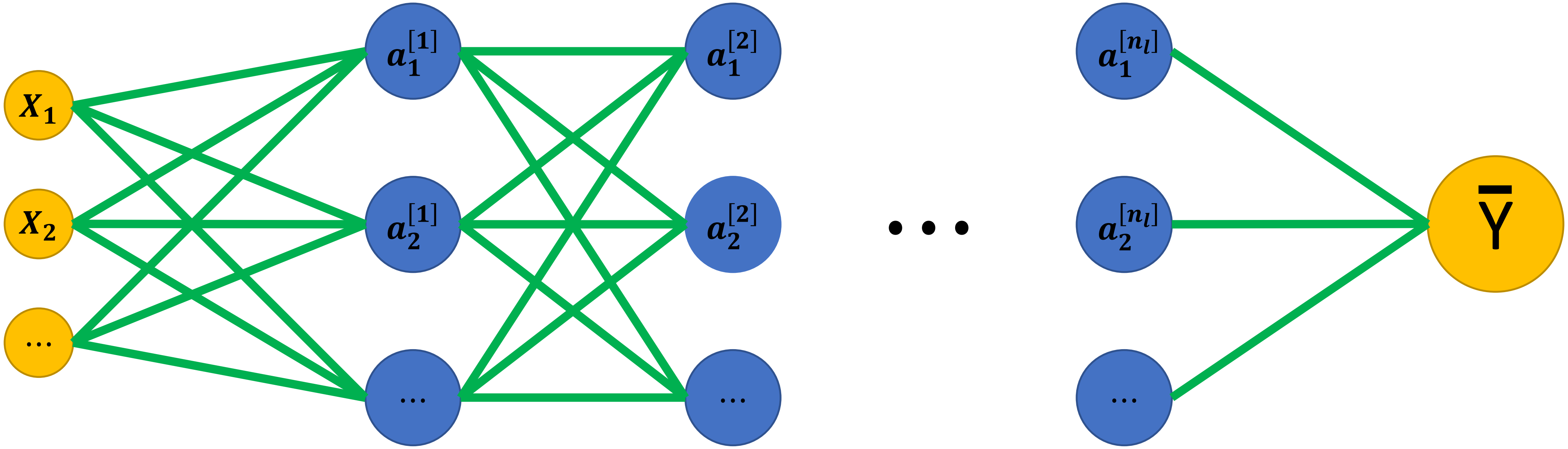
Activation function

Weights

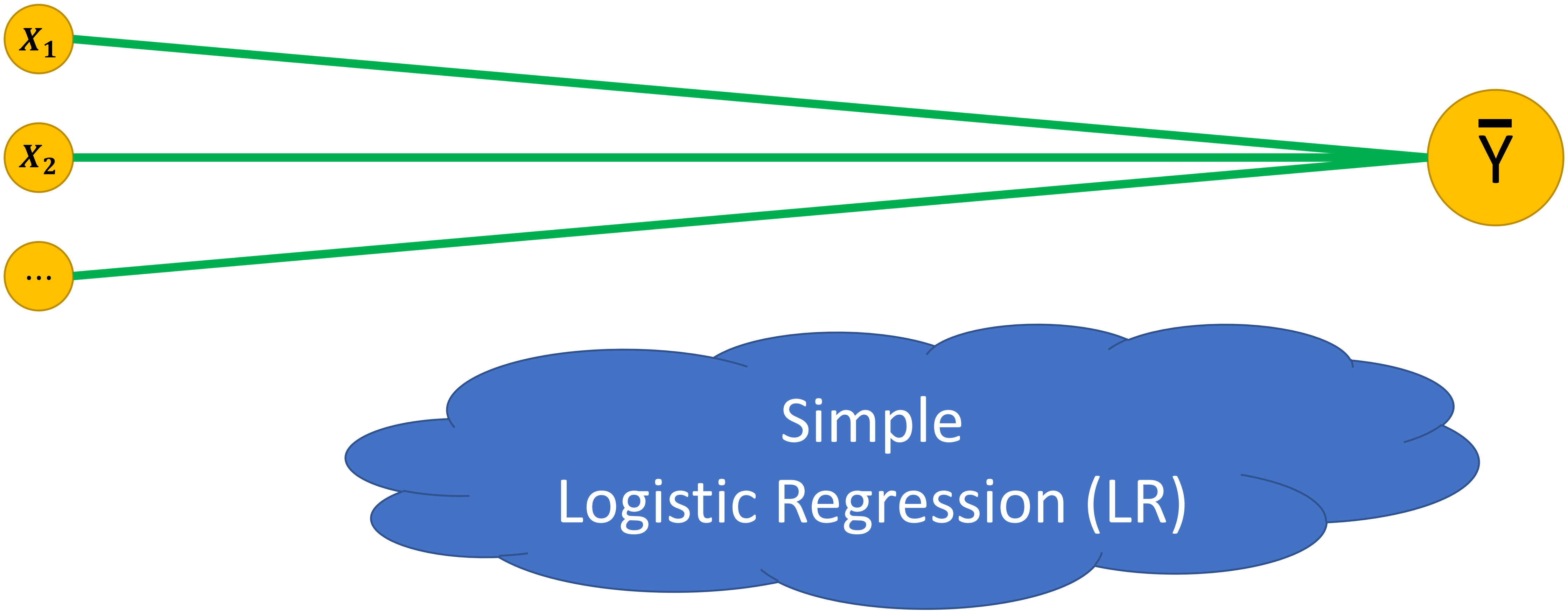
Activation Functions



What would happen without activation func?



What would happen without activation func?



What are the parameters?

What are the hyper-parameters?

Notation

Notation

$$z_i^{[l]} = \sum_j W_{i,j}^{[l]} a_j^{[l-1]} + B_i^{[l]}$$

$$a_i^{[l]} = f_{act}^{[l]}\left(z_i^{[l]}\right)$$

Notation

$\boldsymbol{W}^{[l]}$: Weights in layer l

$\boldsymbol{B}^{[l]}$: Bias in layer l

$\boldsymbol{Z}^{[l]}$: Linear outcome in layer l

$\boldsymbol{f}_{\boldsymbol{act}}^{[l]}$: Activation func. in layer l

$\boldsymbol{A}^{[l]}$: Full outcome in layer l

$$\boldsymbol{Z}^{[l]} = \boldsymbol{W}^{[l]}.\boldsymbol{A}^{[l-1]} + \boldsymbol{B}^{[l]}$$

$$\boldsymbol{A}^{[l]} = \boldsymbol{f}_{\boldsymbol{act}}^{[l]}(\boldsymbol{Z}^{[l]})$$

Notation

$$\boldsymbol{Z}^{[l]} = \boldsymbol{W}^{[l]}.\boldsymbol{A}^{[l-1]} + \boldsymbol{B}^{[l]}$$

$$\boldsymbol{A}^{[l]} = \boldsymbol{f}_{\boldsymbol{act}}^{[l]}(\boldsymbol{Z}^{[l]})$$

$\boldsymbol{n}^{[l]}$: #nodes in layer l

$\boldsymbol{W}^{[l]}$: Weights in layer l — $\boldsymbol{W}^{[l]}: (n^{[l]}, n^{[l-1]})$

$\boldsymbol{B}^{[l]}$: Bias in layer l — $\boldsymbol{B}^{[l]}: (n^{[l]}\ )$

$\boldsymbol{Z}^{[l]}$: Linear outcome in layer l — $\boldsymbol{Z}^{[l]}: (n^{[l]}, n_s)$

$\boldsymbol{f}_{\boldsymbol{act}}^{[l]}$: Activation func. in layer l

$\boldsymbol{A}^{[l]}$: Full outcome in layer l — $\boldsymbol{A}^{[l]}: (n^{[l]}, n_s)$

Notation

$$\boldsymbol{Z}^{[l]} = \boldsymbol{W}^{[l]} . \boldsymbol{A}^{[l-1]} + \boldsymbol{B}^{[l]}$$

$$\boldsymbol{A}^{[l]} = \boldsymbol{f}_{\boldsymbol{act}}^{[l]}(\boldsymbol{Z}^{[l]})$$

$$\boldsymbol{W}^{[l]} : \left(n^{[l]}, n^{[l-1]}\right)$$

$$\boldsymbol{B}^{[l]} : \left(n^{[l]}\right)$$

$$\boldsymbol{Z}^{[l]} : \left(n^{[l]}, n_s\right)$$

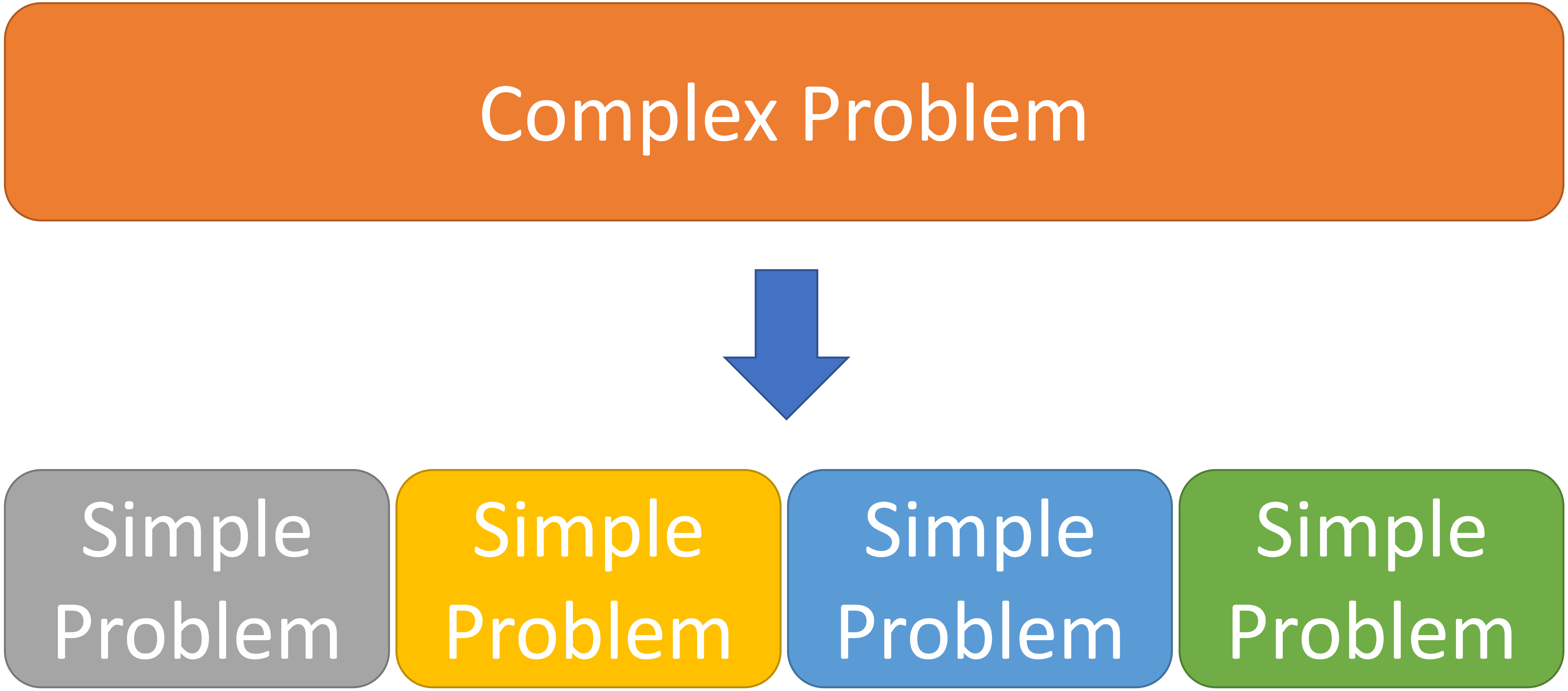
$$\boldsymbol{A}^{[l]} : \left(n^{[l]}, n_s\right)$$

Why?

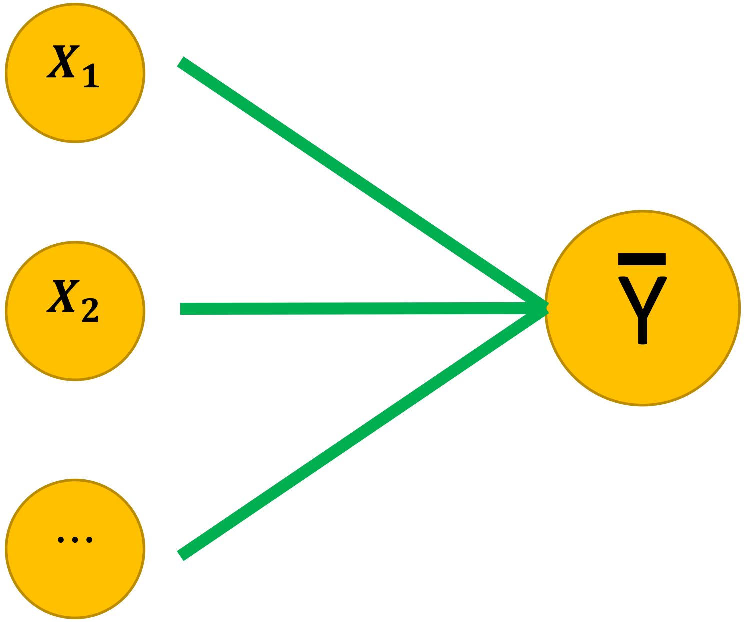
The Logic of using a layered model

Why do we need to do layers?

Reductionism: how we solve problems …



Let’s take a step back and take a look at Logistic Regression.

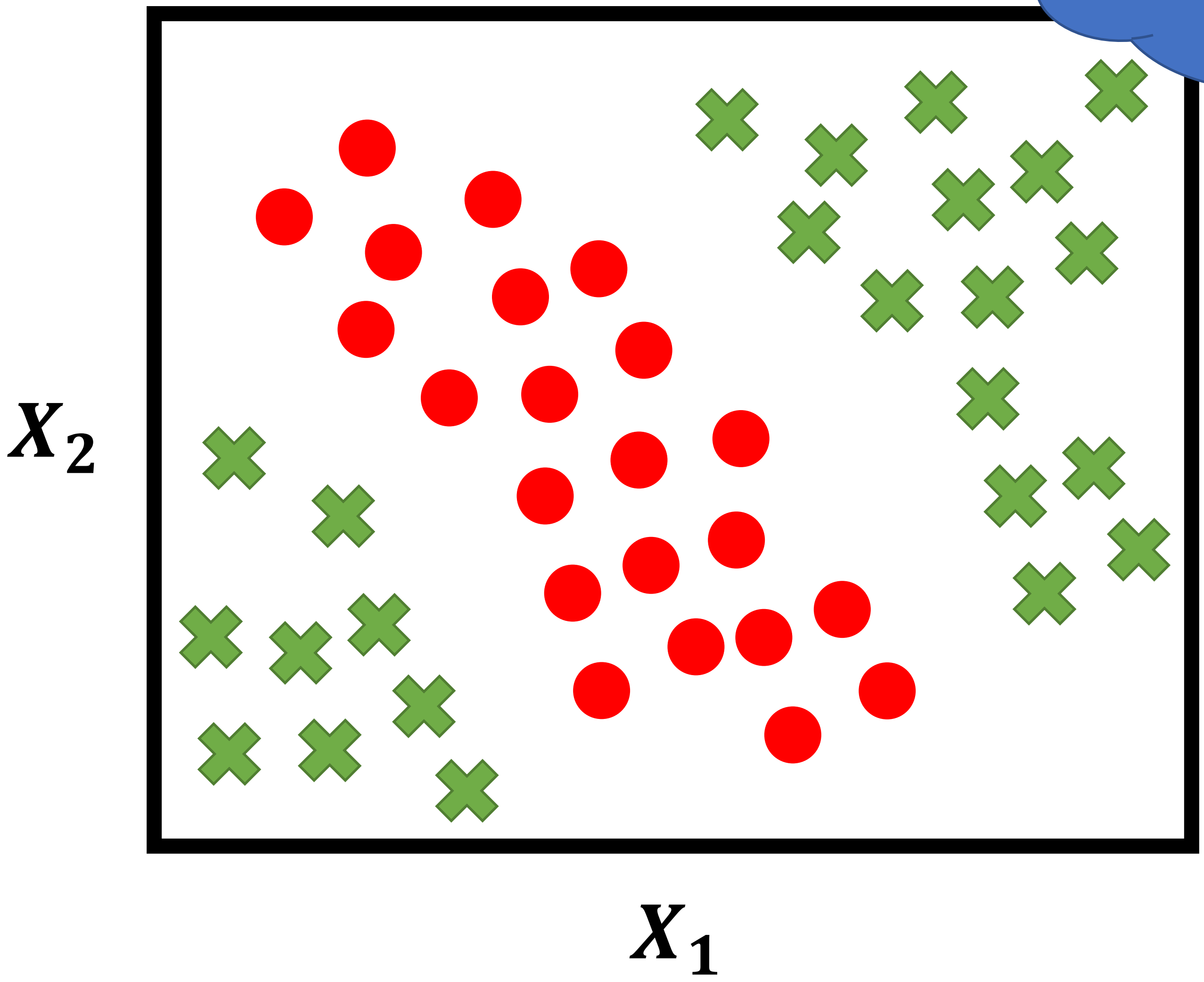


$$= \text{sigmoid}\left(\sum_{j} \boldsymbol{W}_{j}\ \boldsymbol{X}_{j}\right)$$

Linear

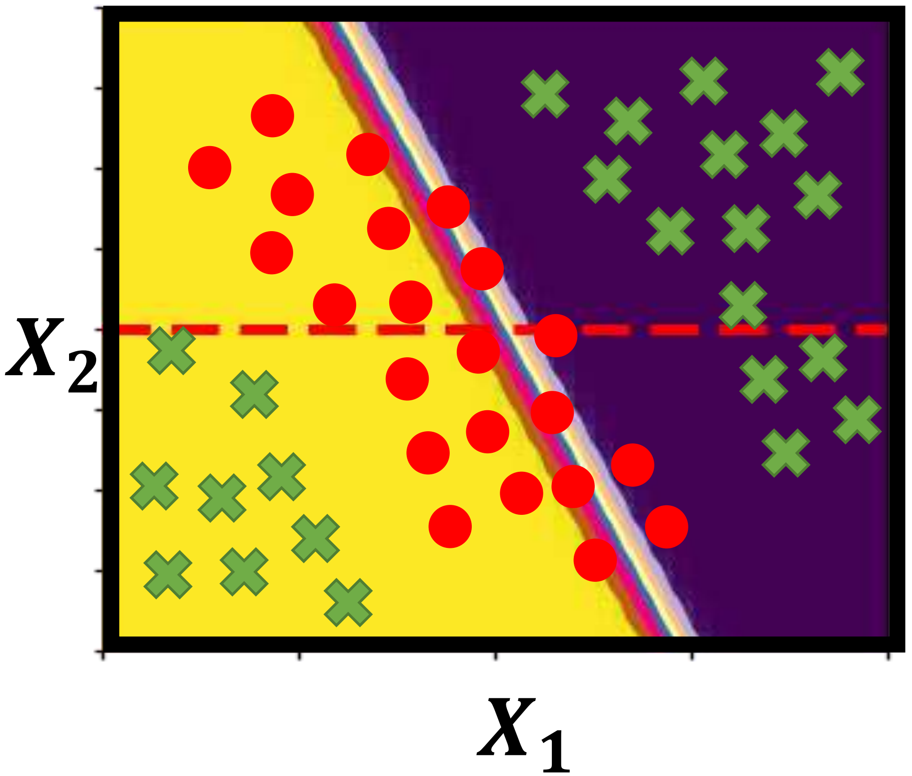
Consider the following classification

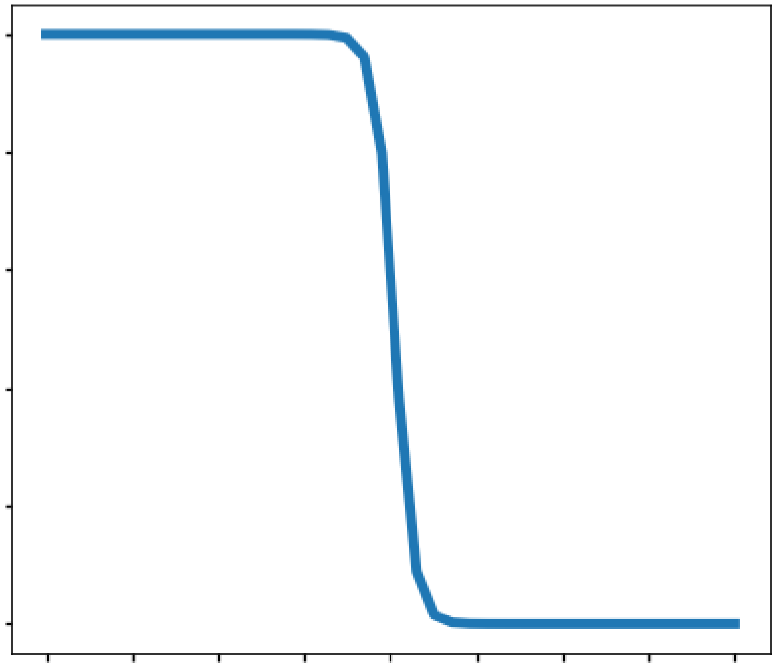
Example



How well would LR do on this?

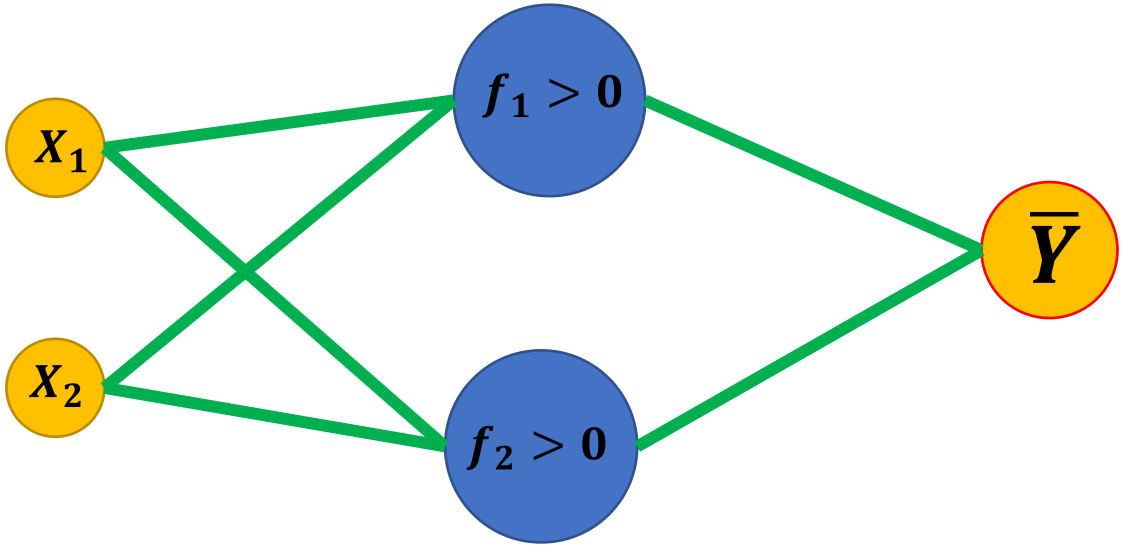
Example

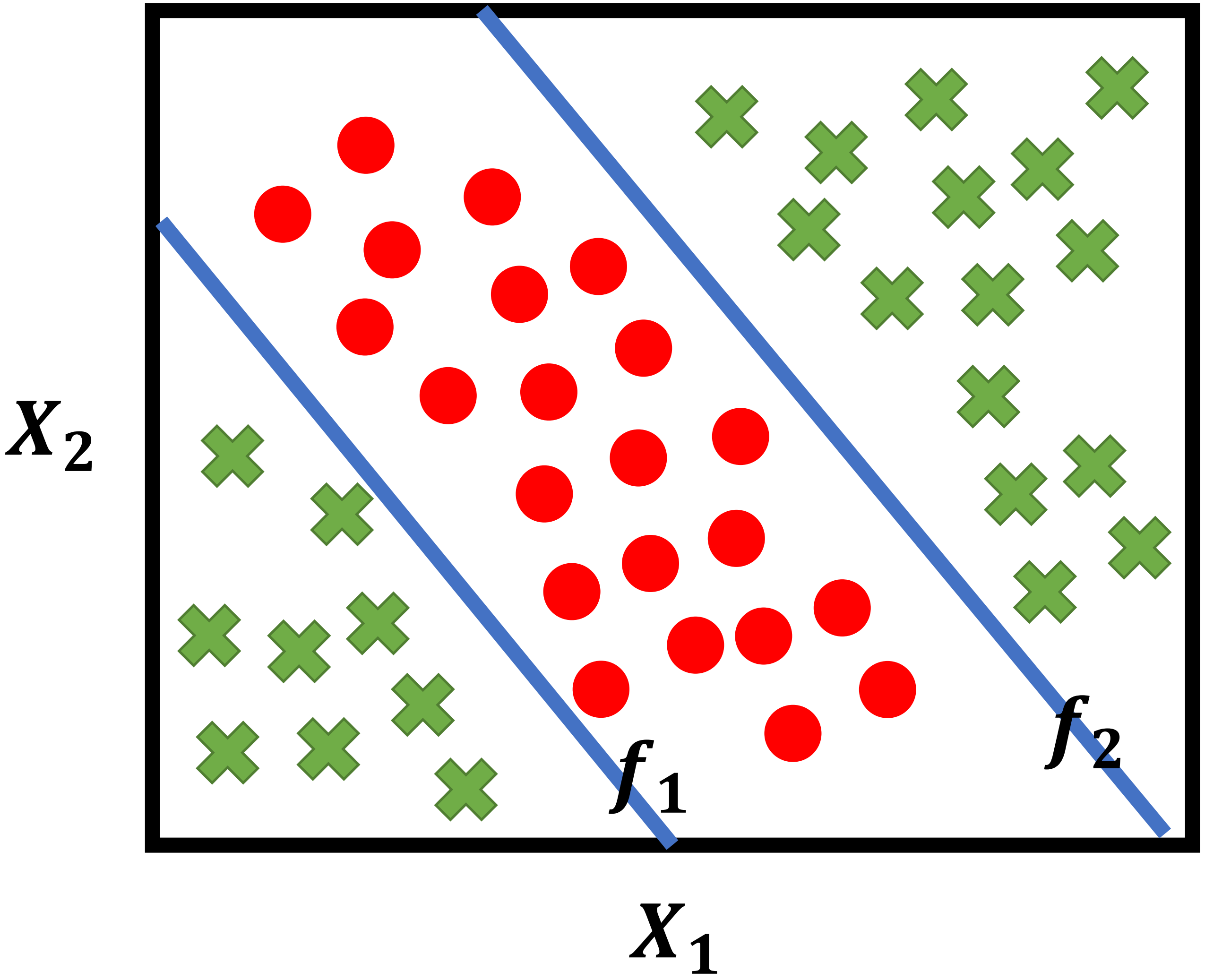




What if we were to combine two LR

Example: what is the solution





Example: NMR spectrum

Universality of NN:

Universal approximation theorem

- Hornik, Kurt; Tinchcombe, Maxwell; White, Halbert (1989). *Multilayer Feedforward Networks are Universal Approximators*. *Neural Networks*. **2**. Pergamon Press. pp. 359–366.

Universality of NN:

- Arbitrary width and Sigmoid, George Cybenko 89

Depth

- Enough depth, can reduce the width to n+c

Activation Funciton

- Any activation function, Kurt Hornik 91
- Non-polynomial activation func are universal, Moshe Leshno et al 93

[1905.08539] Universal Approximation with Deep Narrow Networks (arxiv.org)
Universal approximation theorem - Wikipedia