An overview of machine learning

Introductory talk

Friedrich Anders & Emily L. Hunt | MWGaia-DN introductory school | September 10th, 2024

Hi!

Emily

Research interests:

star clusters, machine learning, variable stars, Gaia, LSST

Career:

Postdoc in Heidelberg, 🇩🇪

Career path: 🇬🇧→🇩🇪

Friedrich

Research interests:

galactic archeology, data science, stellar populations

Career:

Postdoc at ICCUB, Barcelona 🇪🇸

Career path: 🇩🇪→🇪🇸

Topics we'll cover over the next two days

- **Supervised machine learning**

  classification; regression; neural networks; decision trees

- **Unsupervised machine learning**

  clustering algorithms, dimensionality reduction techniques

- **The future of machine learning**

  interpreting models; uncertainty; symbolic regression; the future

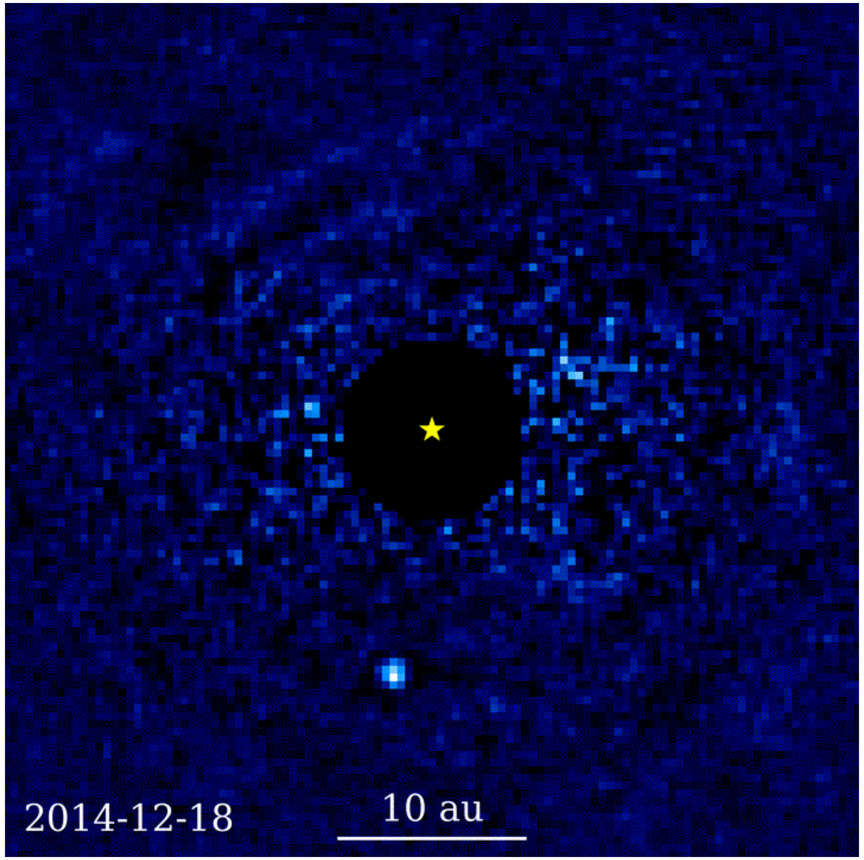
Question:

Who has **used machine learning** (ML) before?

Question:

Who feels **especially comfortable** using ML?

When ML is not the best answer



Please do not re-derive Kepler's laws with ML!

Machine learning is the **flashy** solution, but **often not the best one**

If you can **write down a model** for your problem, then solve it that way

'Traditional' statistics has stuck around for **centuries** for a reason

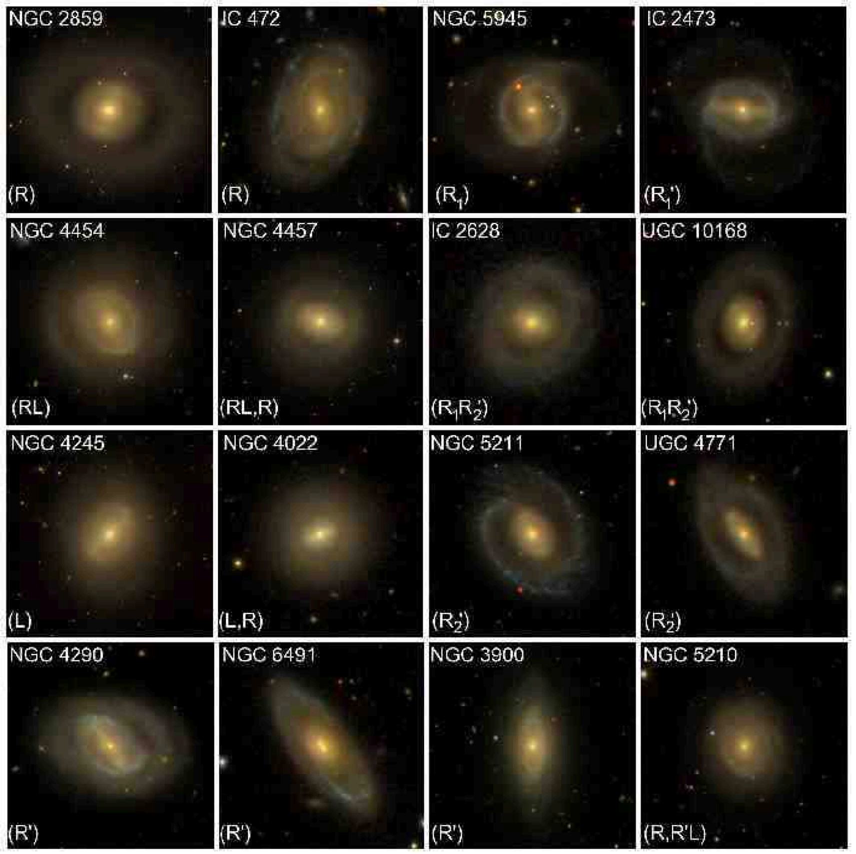
When a model is too hard to write down

But machine learning is **amazing** when you **cannot write down a model**

Image recognition is one such case

Imagine modelling a 128×128 pixel 3-color image with classical techniques!

(that's **~50 000** 'input parameters')



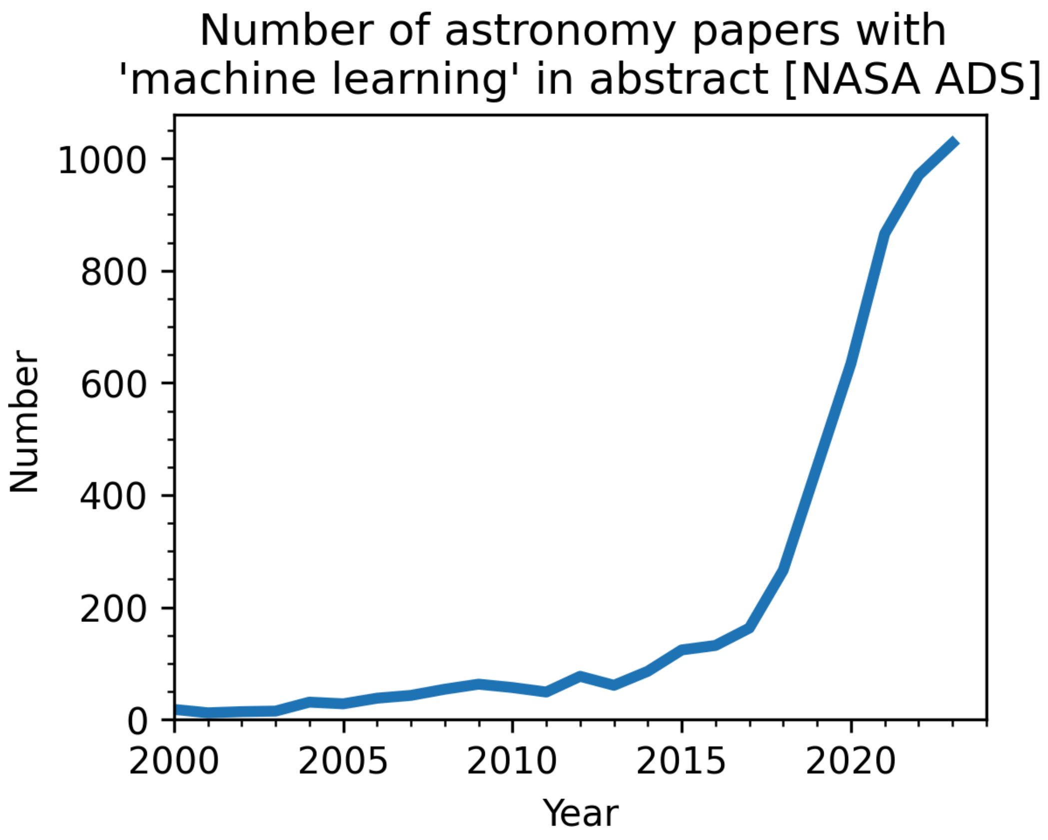
Credit: CTIO/NOIRLab/DOE/NSF/AURA/J. Moustakas

Many things are very hard to model!

ML is common in astronomy to...

- Classify galaxy images
- Extract stellar parameters from spectra
- Blindly search for clusters in data

+ MANY more things!



What is machine learning?

What is machine learning?

Machine learning is about using **flexible models** to make predictions from data.

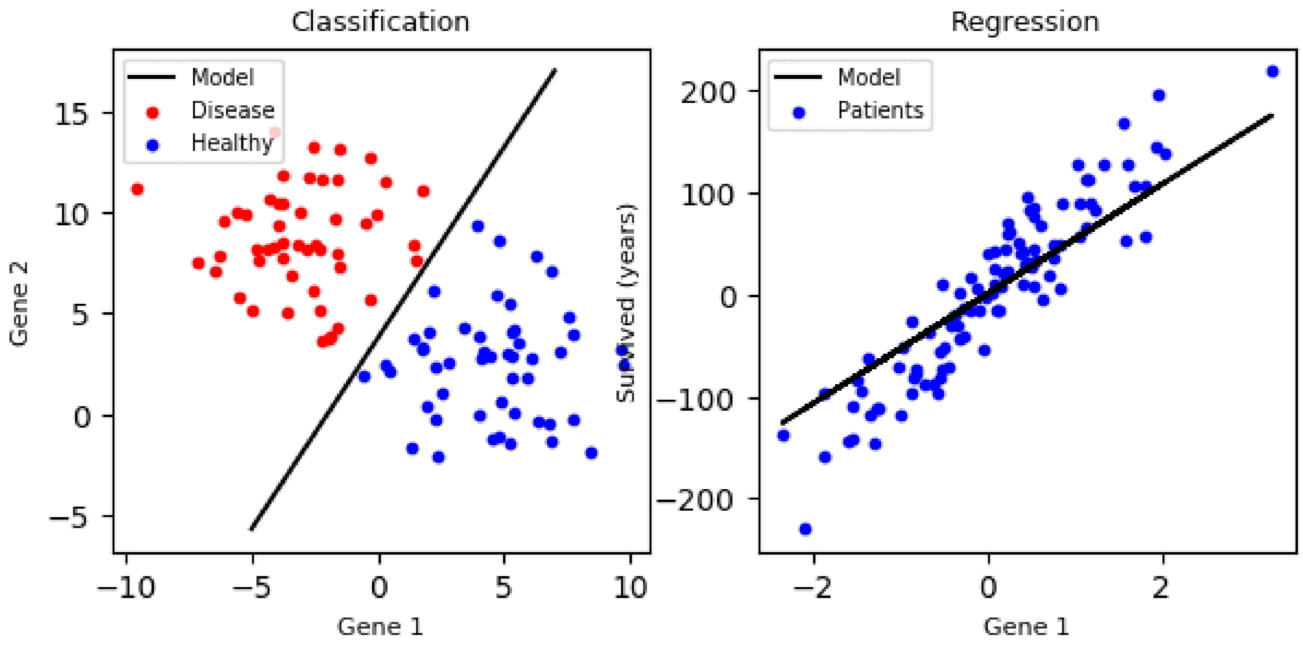
What is machine learning?

Machine learning is about using **flexible models** to make predictions from data.

A few **popular techniques** like neural networks or decision trees can solve many different problems.

Terminology 1: the type of problem

Machine learning can be divided into **classification** or **regression** problems.



Terminiology 2: the type of method

ML methods are usually either **supervised** or **unsupervised.**

Supervised ML:

fit your method on data with **known labels**, a.k.a. '**training data**'.
Then, use it to predict labels of **new data.**

Unsupervised ML:

fit your method on data with **unknown labels,** using a set of parameters to try and find patterns.

How flexible is machine learning?

Universal approximation theorem:

A **sufficiently large** neural network can approximate **any** function.

How flexible is machine learning?

Universal approximation theorem:

A **sufficiently large** neural network can approximate **any** function.

In plainer English:

if you make your neural network **big enough**, it can do **anything!**

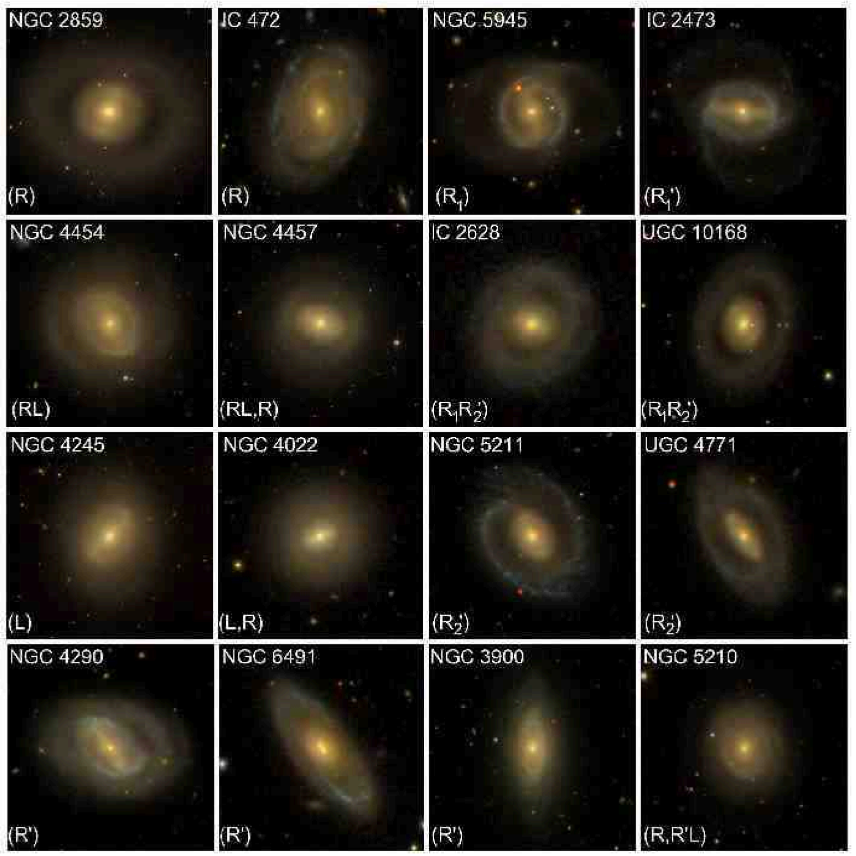
(This applies to many other architectures too, but neural networks are most proven)

The craziest example of this

Let's go back to thinking about galaxy classification again.

Isn't it pretty **crazy** that we have access to models **so flexible** that image classification is possible?

The **universal approximation theorem** seems impossible, right?



Credit: CTIO/NOIRLab/DOE/NSF/AURA/J. Moustakas

Question time!

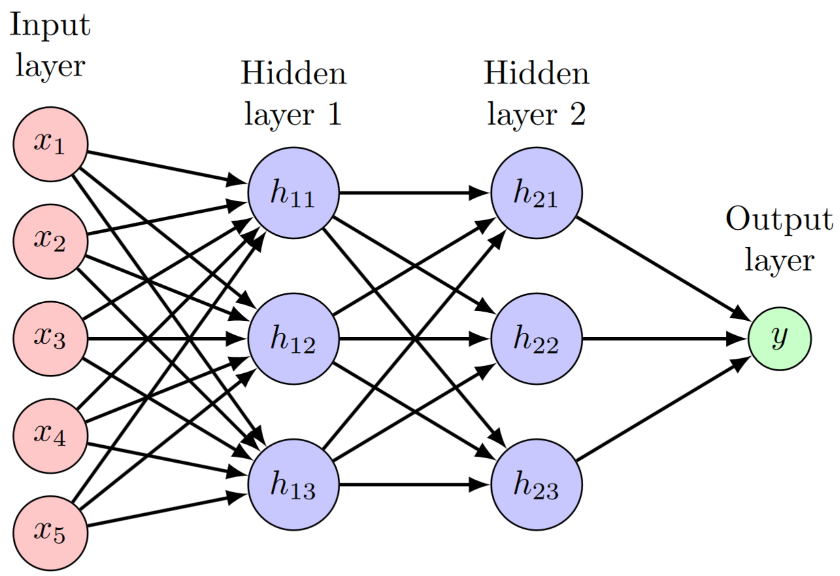
How the hell does the maths for something like galaxy classification even work?

Do you think it must be really complicated?

Neural networks

- They're modelled after "human brains"
- They're usually drawn with these obtuse diagrams like on the right

Must be hard, right?



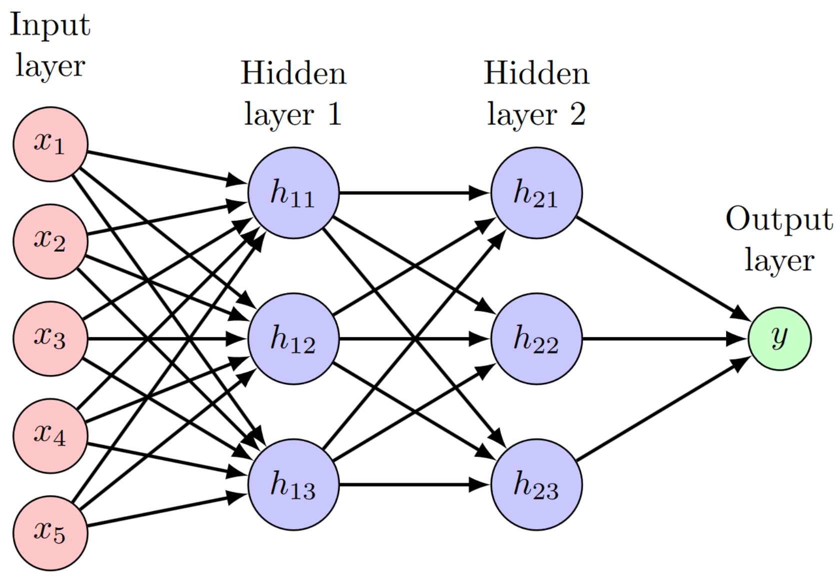
I'm allowed to insult this diagram because it's from my master's thesis!

Neural networks

- They're modelled after "human brains"
- They're usually drawn with these obtuse diagrams like on the right

Must be hard, right?

Let's dive into the maths of how they work as an exercise in why **you** can do machine learning.



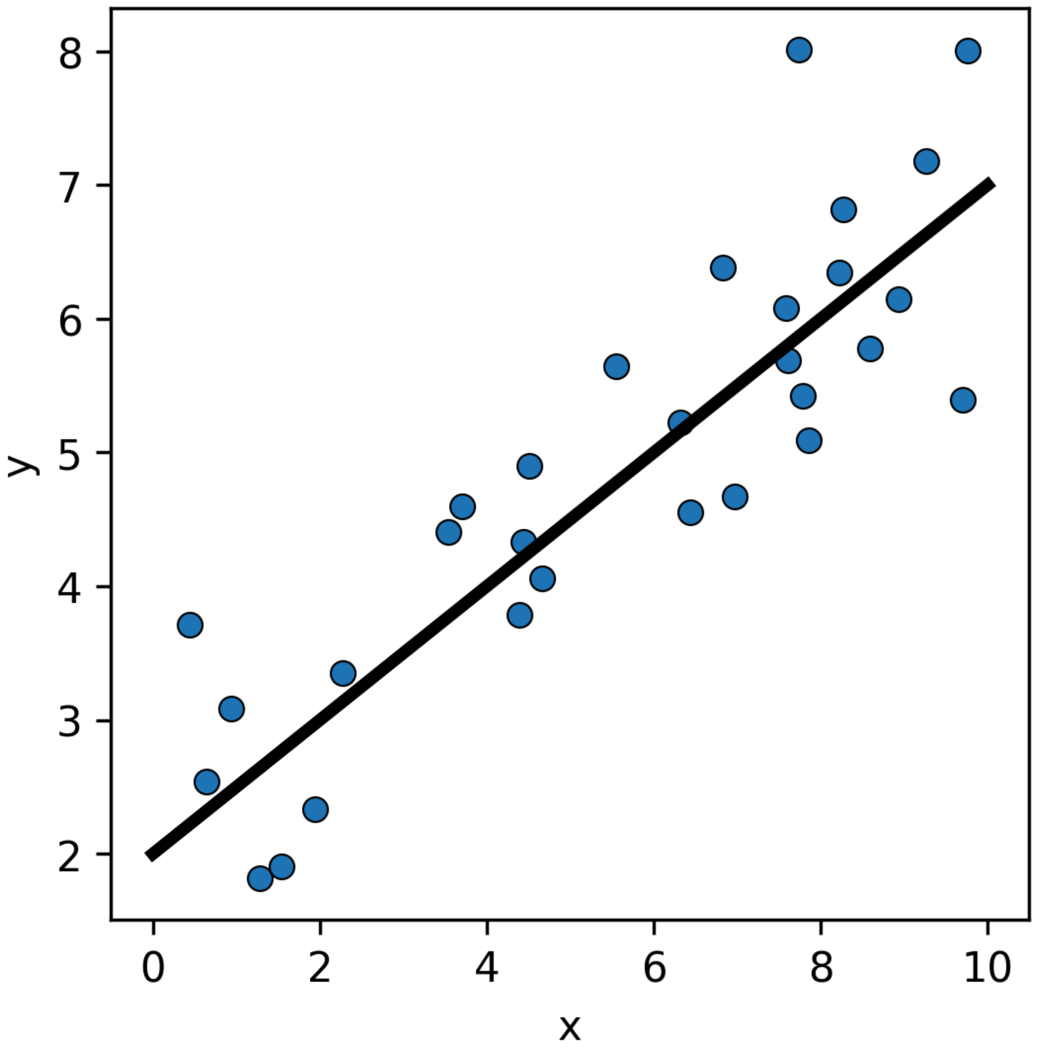
I'm allowed to insult this diagram because it's from my master's thesis!

The simplest model in science

Let's build them up from scratch!

You've all seen $y = mx + c$.

Linear regression is as simple as it comes.



The simplest model in science

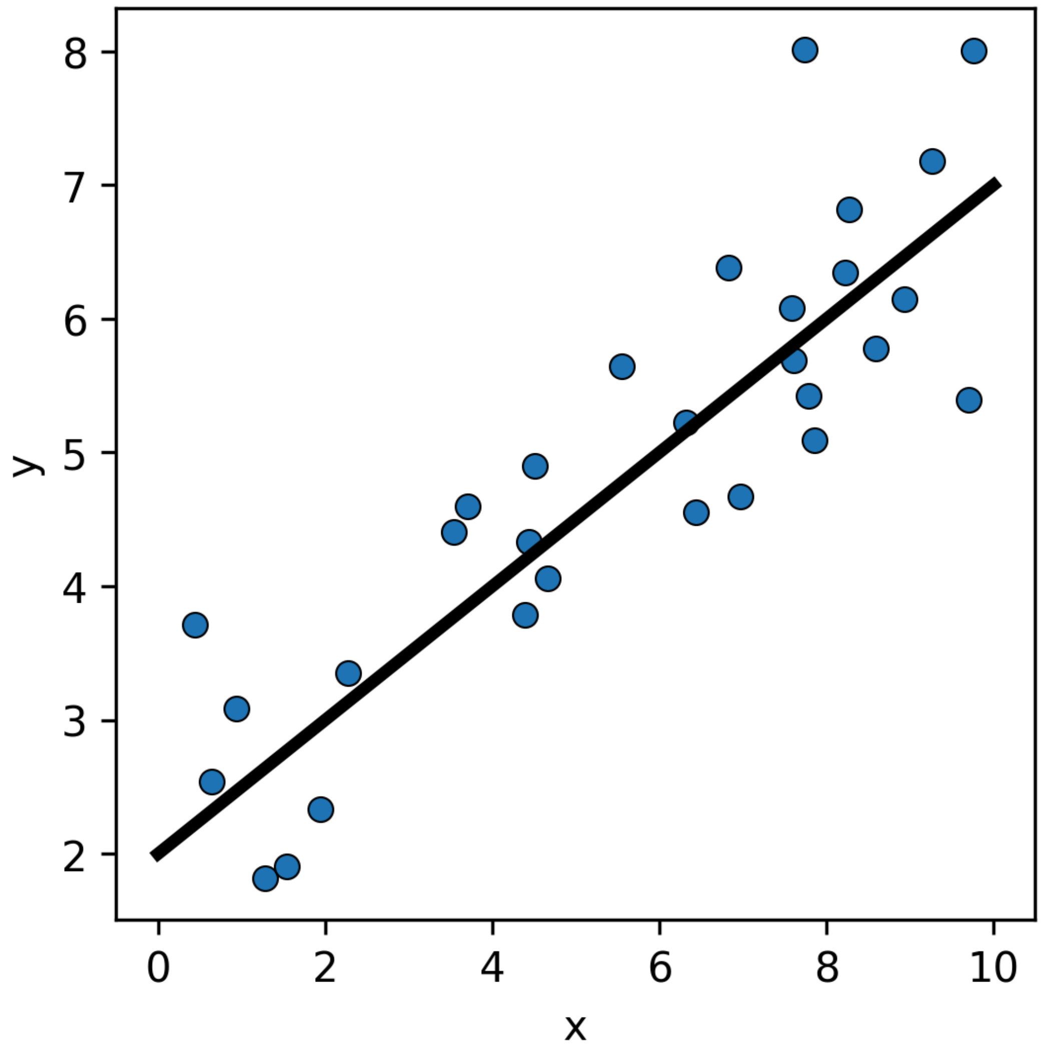
Let's build them up from scratch!

You've all seen $y = mx + c$.

Linear regression is as simple as it comes.

Let's use slightly different letters:

$$y = wx + b$$

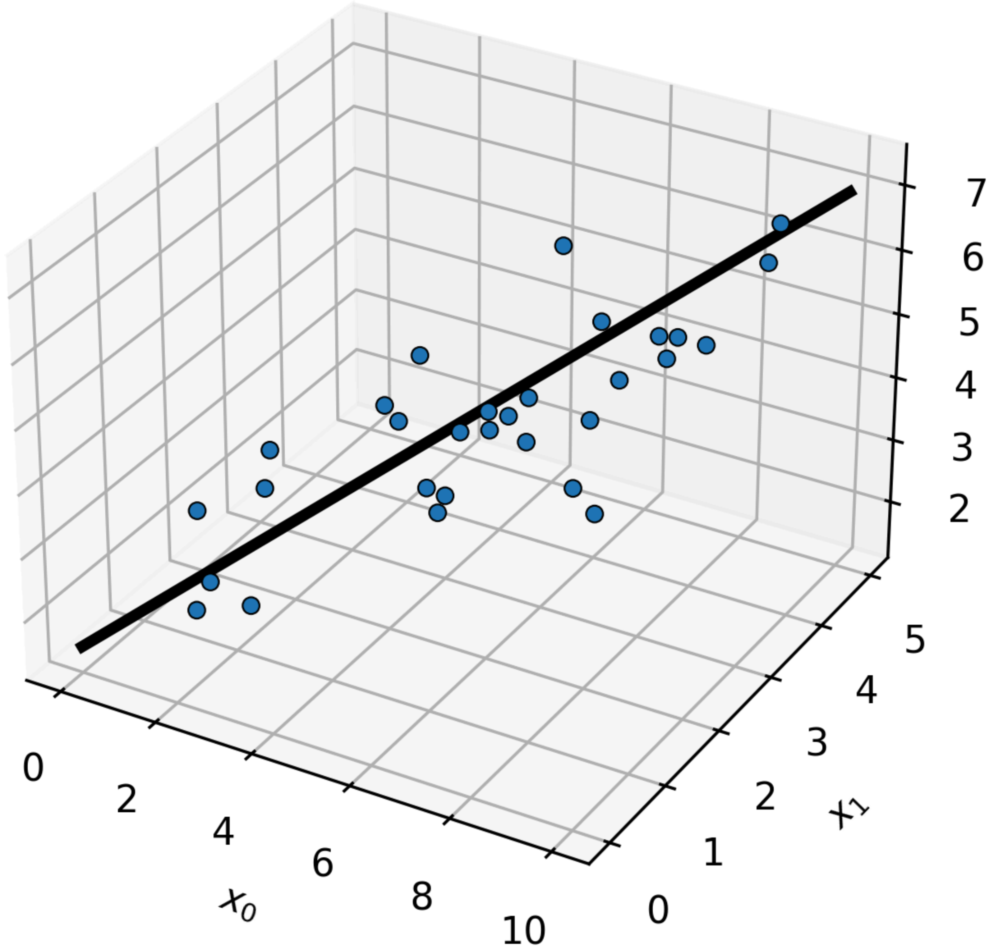


Making it multi-dimensional

n-dimensional linear regression

isn't much harder:

$$y = w_1 x_1 + \ldots + w_n x_n + b$$



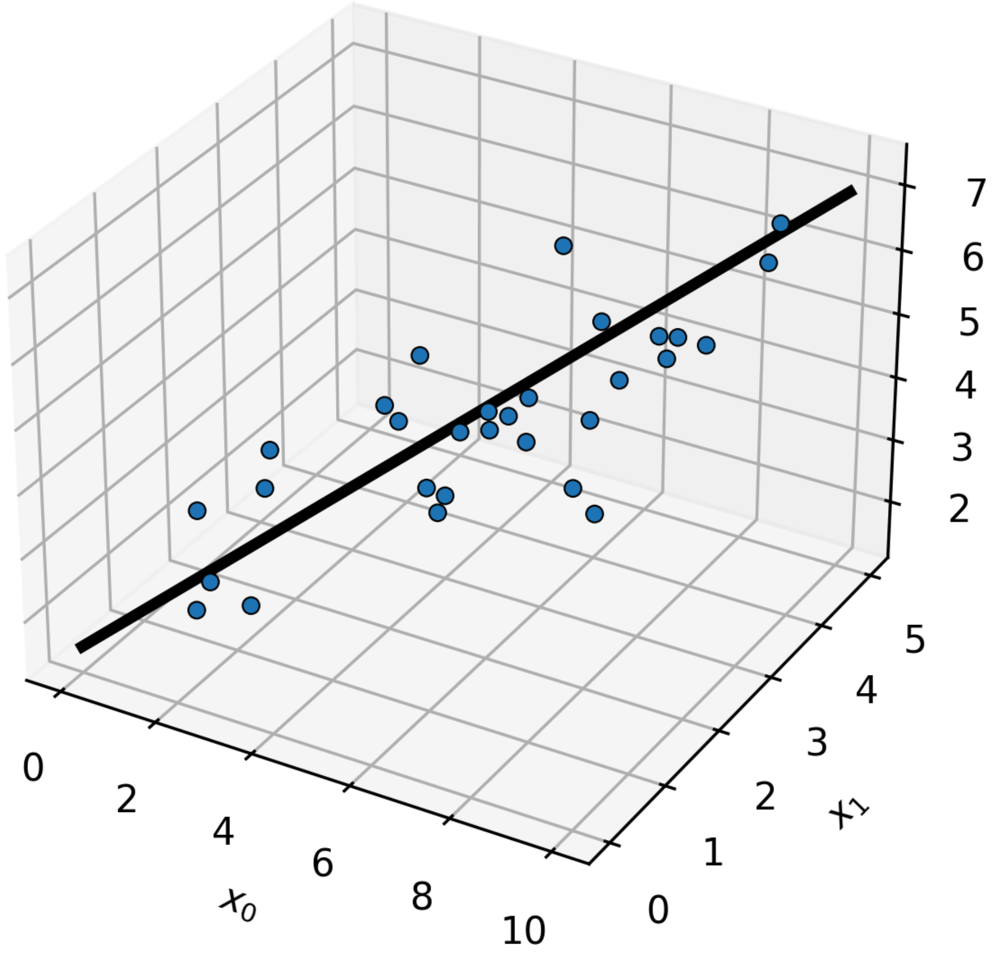
Making it multi-dimensional

n-dimensional linear regression isn't much harder:

$$y = w_1 x_1 + \dots + w_n x_n + b$$

Or, written as **vectors**:

$$y = \vec{w} \cdot \vec{x} + b$$

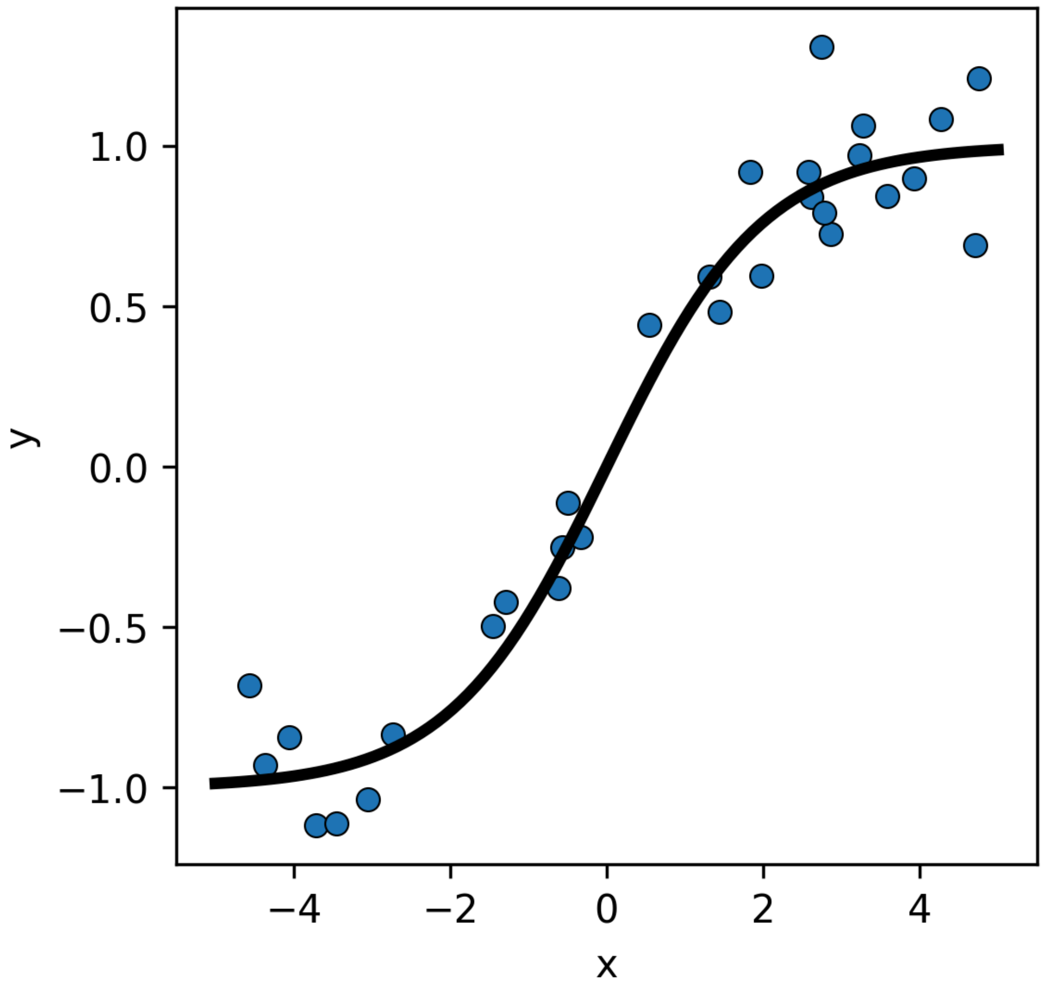


Making it non-linear

We can do **non-linear regression** if we apply a function f:

$$y = f\left(\vec{w} \cdot \vec{x} + b\right)$$

This allows us to fit any function f that we want to.



Recap

We've taken the simplest model in science, **linear regresssion**:

$$y = wx + b$$

We've made it **multi-dimensional:**

$$y = \vec{w} \cdot \vec{x} + b$$

And we made it **non-linear** with a function f that we can choose:

$$y = f\left(\vec{w} \cdot \vec{x} + b\right)$$

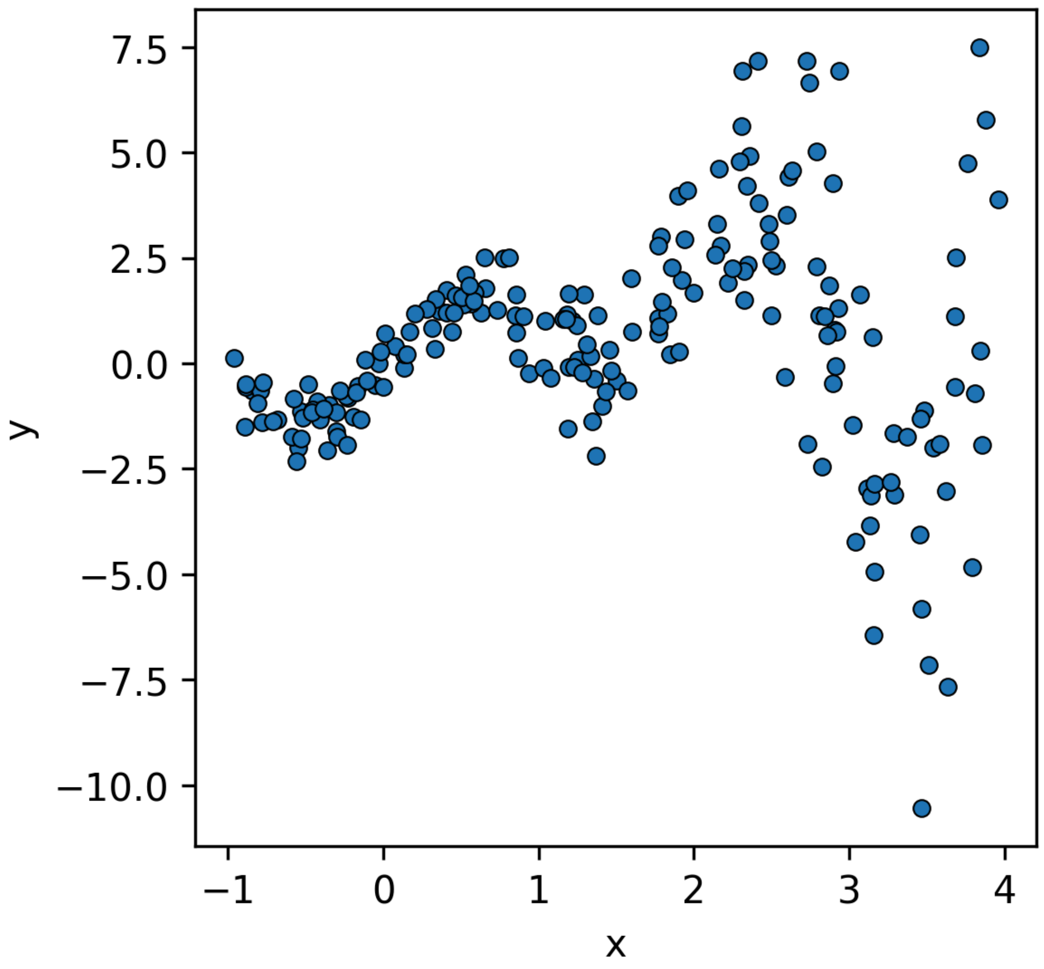
This model is special!

What if I told you that:

$$y = f\left(\vec{w} \cdot \vec{x} + b\right)$$

is a **building block** we can use to approximate **anything.**

How can we extend it further?



Idea 1: connect them together in 'layers'

What happens if we **connect** the blocks together?

Let's define a new **intermediate vector** $\vec{h_j}$ of length n, with elements:

$$\vec{h_j} = \begin{pmatrix} f\left(\vec{w_{1j}} \cdot \vec{x} + b_{1j}\right) \\ \cdots \\ f\left(\vec{w_{nj}} \cdot \vec{x} + b_{nj}\right) \end{pmatrix}$$

We can connect them together **as many times as we want to.**

A universal approximator?

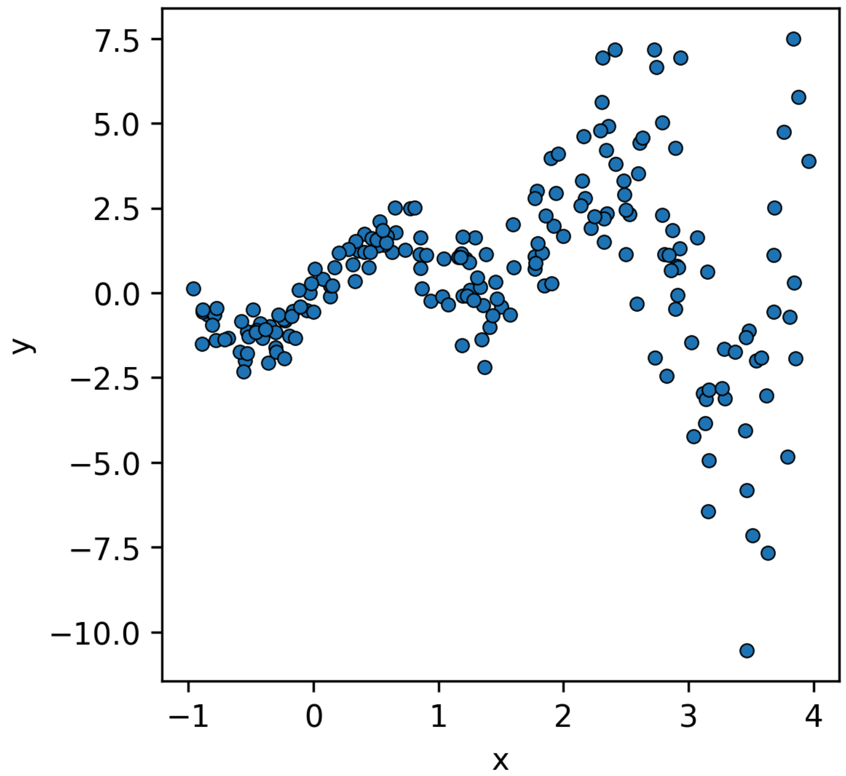
Let's try two $\vec{h}$ vectors of size 20, giving us three **'layers'** of computation:

$$\vec{h_1} = \begin{pmatrix} f\left(\vec{w_{1,1}} \cdot \vec{x} + b_{1,1}\right) \\ \cdots \\ f\left(\vec{w_{20,1}} \cdot \vec{x} + b_{20,1}\right) \end{pmatrix}$$

$$\vec{h_2} = \begin{pmatrix} f\left(\vec{w_{1,2}} \cdot \vec{h_1} + b_{1,2}\right) \\ \cdots \\ f\left(\vec{w_{20,2}} \cdot \vec{h_1} + b_{20,2}\right) \end{pmatrix}$$

$$y = f\left(\vec{w_{1,3}} \cdot \vec{h_2} + b_{1,3}\right)$$

We now have **41** 'layered' nonlinear regressions.



A universal approximator?

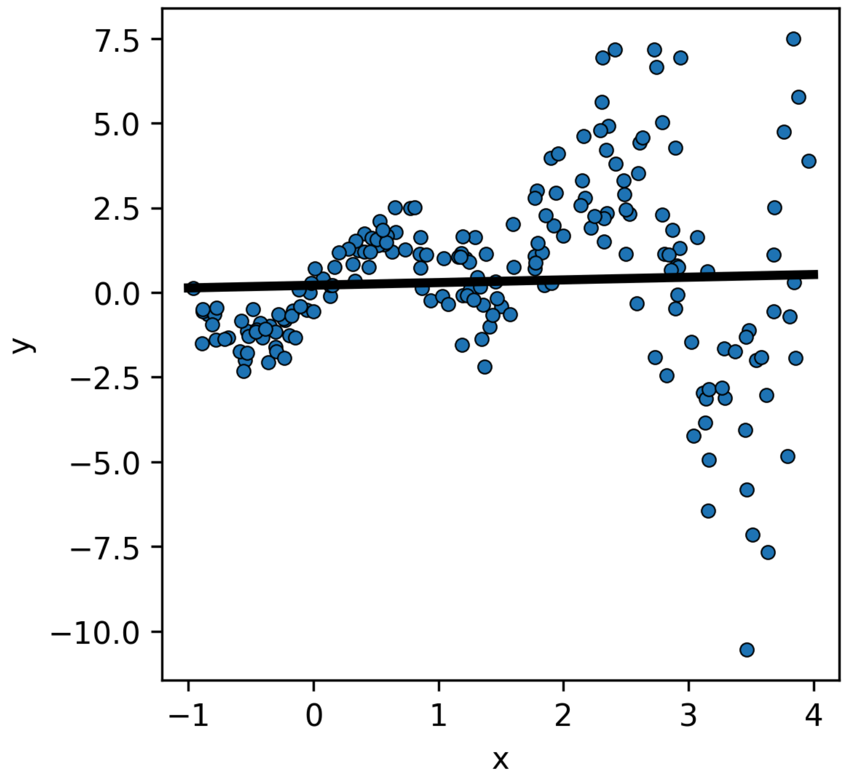
Let's try two $\vec{h}$ vectors of size 20, giving us three **'layers'** of computation:

$$\vec{h_1} = \begin{pmatrix} f\left(\vec{w_{1,1}} \cdot \vec{x} + b_{1,1}\right) \\ \cdots \\ f\left(\vec{w_{20,1}} \cdot \vec{x} + b_{20,1}\right) \end{pmatrix}$$

$$\vec{h_2} = \begin{pmatrix} f\left(\vec{w_{1,2}} \cdot \vec{h_1} + b_{1,2}\right) \\ \cdots \\ f\left(\vec{w_{20,2}} \cdot \vec{h_1} + b_{20,2}\right) \end{pmatrix}$$

$$y = f\left(\vec{w_{1,3}} \cdot \vec{h_2} + b_{1,3}\right)$$

We now have **41** 'layered' nonlinear regressions.



A universal approximator?

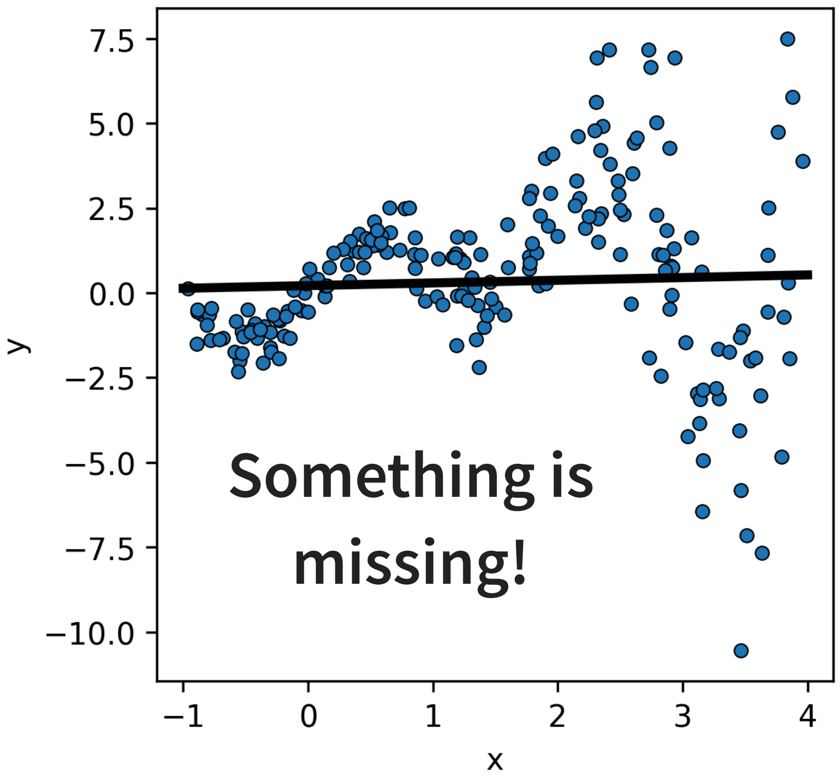
Let's try two $\vec{h}$ vectors of size 20, giving us three **'layers'** of computation:

$$\vec{h_1} = \begin{pmatrix} f\left(\vec{w_{1,1}} \cdot \vec{x} + b_{1,1}\right) \\ \cdots \\ f\left(\vec{w_{20,1}} \cdot \vec{x} + b_{20,1}\right) \end{pmatrix}$$

$$\vec{h_2} = \begin{pmatrix} f\left(\vec{w_{1,2}} \cdot \vec{h_1} + b_{1,2}\right) \\ \cdots \\ f\left(\vec{w_{20,2}} \cdot \vec{h_1} + b_{20,2}\right) \end{pmatrix}$$

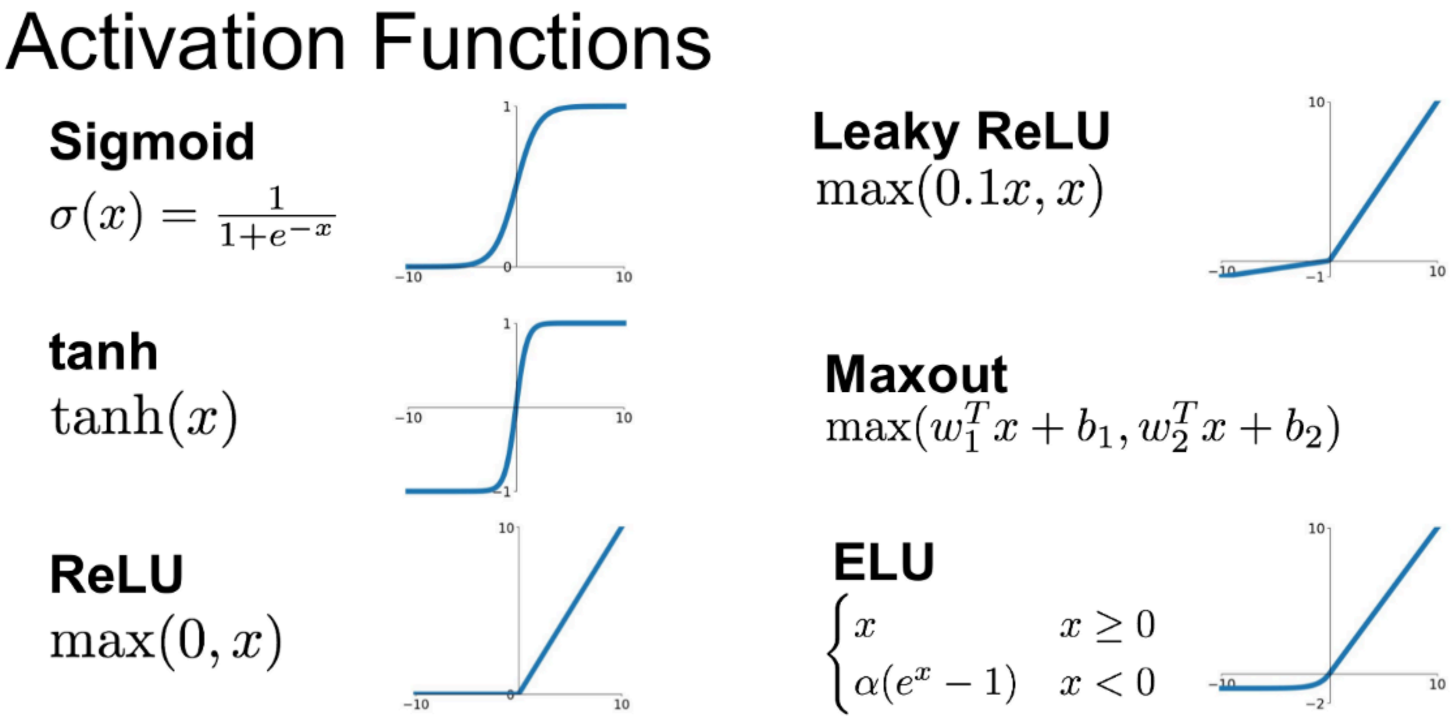
$$y = f\left(\vec{w_{1,3}} \cdot \vec{h_2} + b_{1,3}\right)$$

We now have **41** 'layered' nonlinear regressions.



Idea 2: choosing the right f

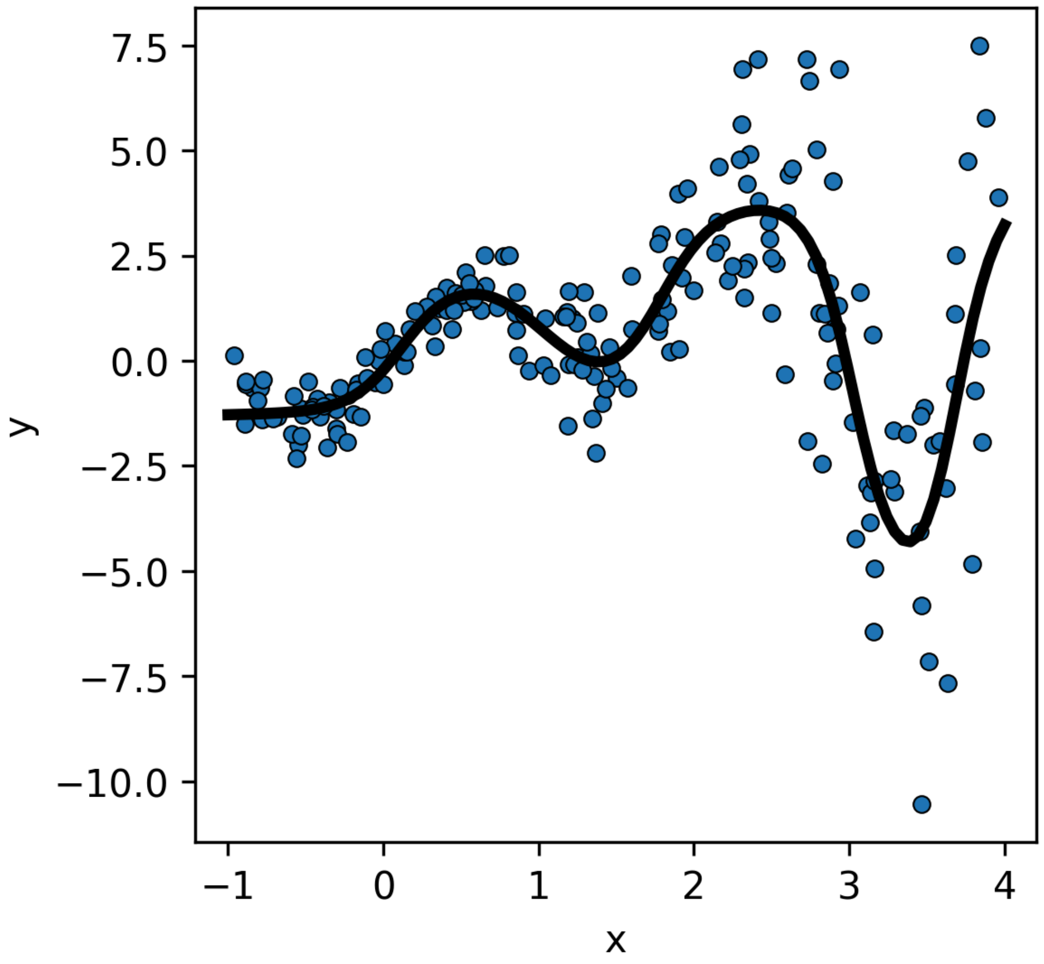
We can choose a function f that has nice properties as a building block



Let's try tanh

Our model of 20 + 20 + 1 nonlinear fitting blocks now fits the curve!

The type of model we made has a special name. **Any guesses?**



What can we learn from this?

What can we learn from this?

1. Machine learning is, at its core, **maths that you can already understand.**

What can we learn from this?

1. Machine learning is, at its core, **maths that you can already understand.**
2. It works because the models are generally **large**, and are set up to work well as **large** models.

What can we learn from this?

1. Machine learning is, at its core, **maths that you can already understand.**
2. It works because the models are generally **large**, and are set up to work well as **large** models.
3. We can make clever choices of things like the activation function f to make them **flexible.**

Things to keep pondering

Things to keep pondering

1. How did I choose the 481 parameters of that model?

Things to keep pondering

1. How did I choose the 481 parameters of that model?

💡 ML only works because of sophisticated ways to train models.

Things to keep pondering

1. How did I choose the 481 parameters of that model?

💡 ML only works because of sophisticated ways to train models.

2. How is it computationally possible to do that?

Things to keep pondering

1. How did I choose the 481 parameters of that model?

💡 ML only works because of sophisticated ways to train models.

2. How is it computationally possible to do that?

💡 Advances in computers over the last 20 years.

Things to keep pondering

1. How did I choose the 481 parameters of that model?

💡 ML only works because of sophisticated ways to train models.

2. How is it computationally possible to do that?

💡 Advances in computers over the last 20 years.

3. Did I have to write the code for that myself?

Things to keep pondering

1. How did I choose the 481 parameters of that model?

💡 ML only works because of sophisticated ways to train models.

2. How is it computationally possible to do that?

💡 Advances in computers over the last 20 years.

3. Did I have to write the code for that myself?

💡 No! There is a **huge** ecosystem of open source software for ML

Recap

We discussed...

- What **machine learning** is and why to use it
- **Classification** vs. **regression**
- **Supervised** vs. **unsupervised** learning
- The **universal approximation theorem**
- ... & that the maths of ML doesn't have to be scary =)