# Chen Data Science & AI for Neuroscience Summer School

TIANQIAO & CHRISSY CHEN INSTITUTE

Caltech

# Introduction to Machine Learning

Sabera Talukder

# What is machine learning?

Machine learning is the study of methods that use data to inform machines on how to solve tasks.

Machine learning is the study of methods that use data to inform machines on how to solve tasks.

The method of choice depends on the task.

Machine learning is the study of methods that use data to inform machines on how to solve tasks.

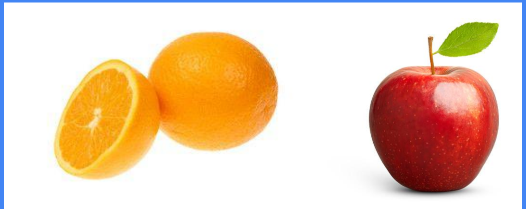
The method of choice depends on the task.

*Deep learning is when the model you use to solve your task is a deep neural network.

# Examples of Tasks

# Examples of Tasks

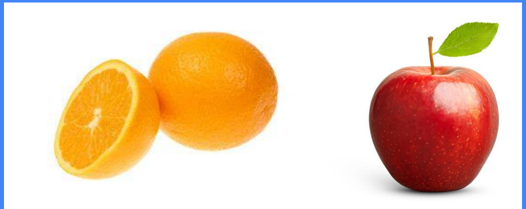
## Classification



Is this the image on the left an apple or an orange?

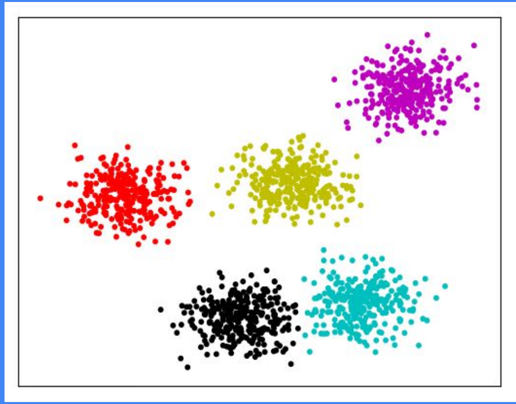
# Examples of Tasks

## Classification



Is this the image on the left an apple or an orange?

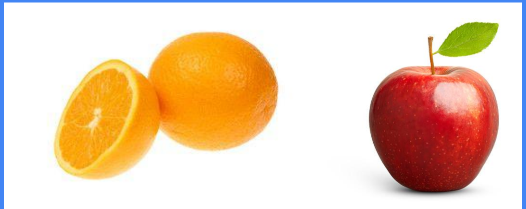
## Clustering



What signals in my dataset are similar or different?

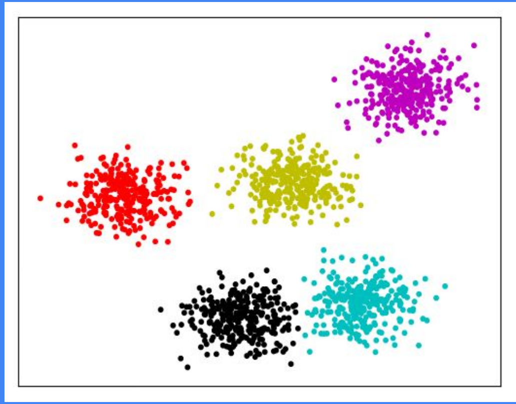
# Examples of Tasks

## Classification



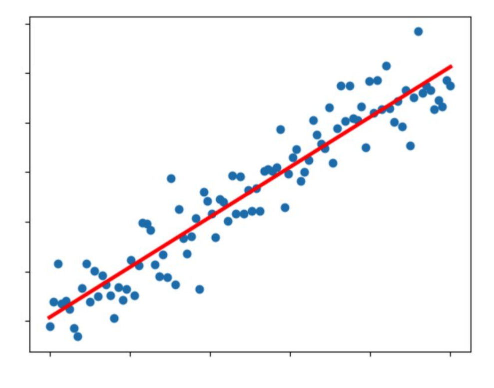
Is this the image on the left an apple or an orange?

## Clustering



What signals in my dataset are similar or different?

## Regression

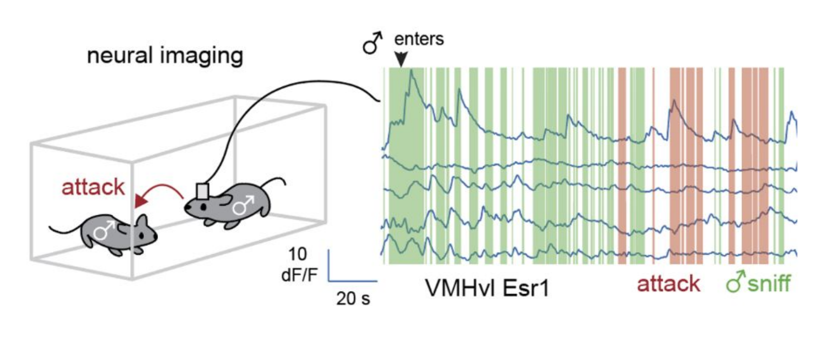


What function describes the relationship between a house’s price and square footage?

# Previous Topics

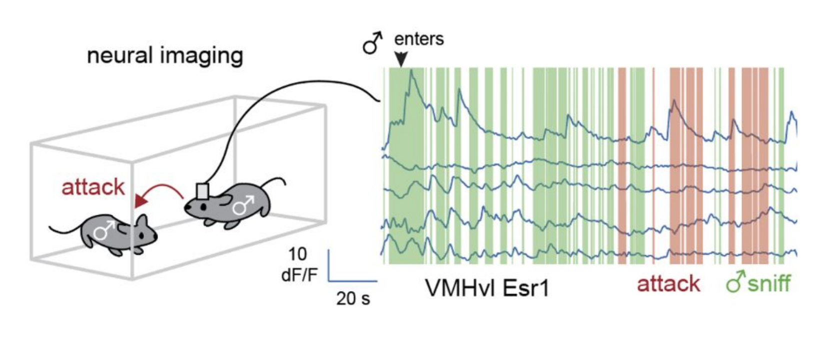
# Previous Topics

- Can we predict mouse attack behavior from linear combinations of neuron values?



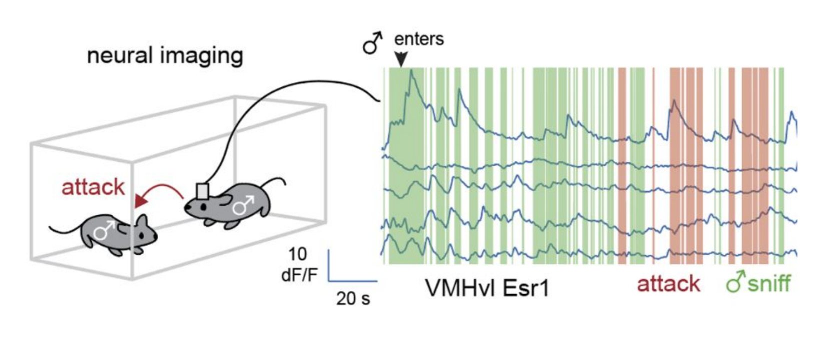
# Previous Topics

- Can we predict mouse attack behavior from linear combinations of neuron values? (linear / logistic regression)

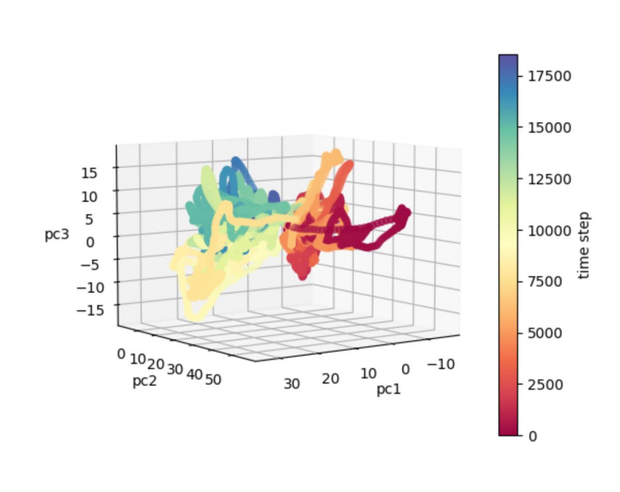


# Previous Topics

- Can we predict mouse attack behavior from linear combinations of neuron values? (linear / logistic regression)

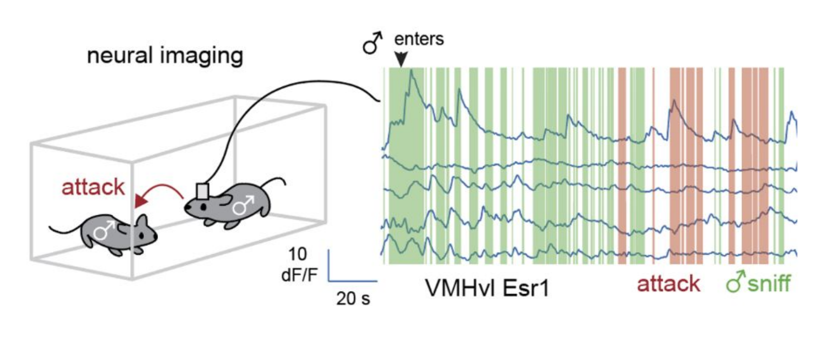



- Can we use principal components analysis to learn a “simple” yet rich representation of high-dimensional neural time-series data?

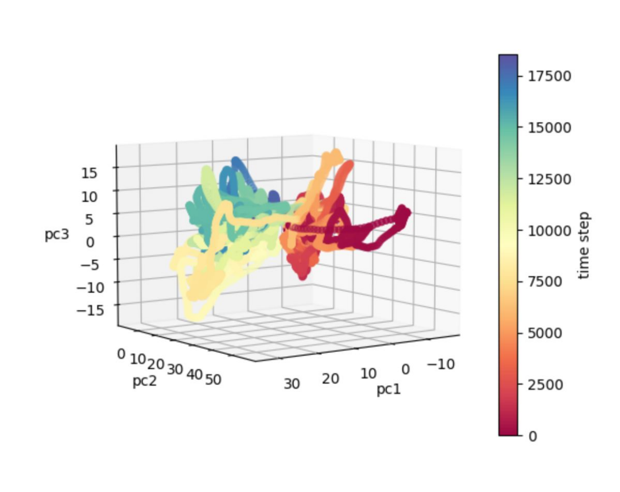


# Previous Topics

- Can we predict mouse attack behavior from linear combinations of neuron values? (linear / logistic regression)




- Can we use principal components analysis to learn a “simple” yet rich representation of high-dimensional neural time-series data? (unsupervised representation learning)



# Examples of Learning Paradigms

# Examples of Learning Paradigms

## Supervised

Learn a function $y = f(x)$ that predicts an output (label) from input (data)

# Examples of Learning Paradigms

## Supervised

Learn a function y = f(x) that predicts an output (label) from input (data)

Data

Label

Apple

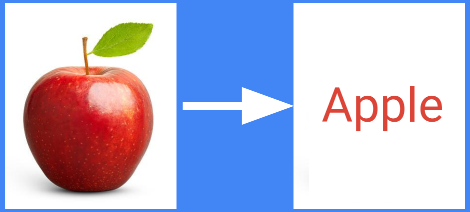
# Examples of Learning Paradigms

## Supervised

Learn a function $y = f(x)$ that predicts an output (label) from input (data)

Data

Label




## Unsupervised

Learn patterns from the data without labels.

# Examples of Learning Paradigms

## Supervised

Learn a function y = f(x) that predicts an output (label) from input (data)

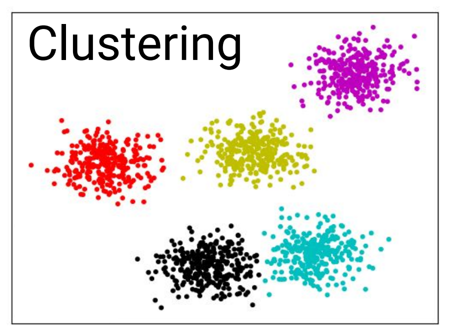
Data

Label

Apple

## Unsupervised

Learn patterns from the data without labels.



# Examples of Learning Paradigms

## Supervised

Learn a function y = f(x) that predicts an output (label) from input (data)

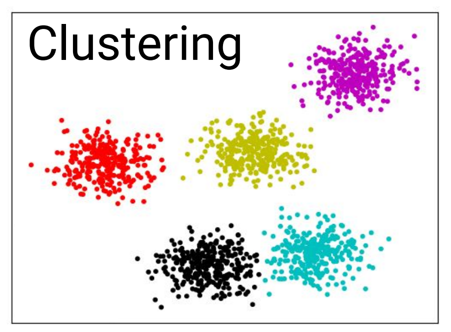
Data

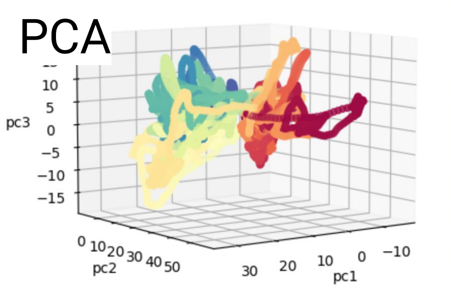
Label

Apple

## Unsupervised

Learn patterns from the data without labels.



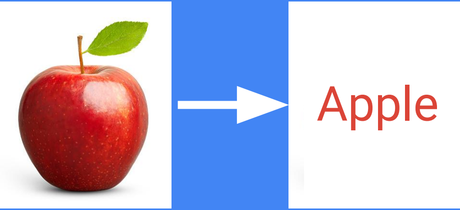
# Examples of Learning Paradigms

## Supervised

Learn a function y = f(x) that predicts an output (label) from input (data)

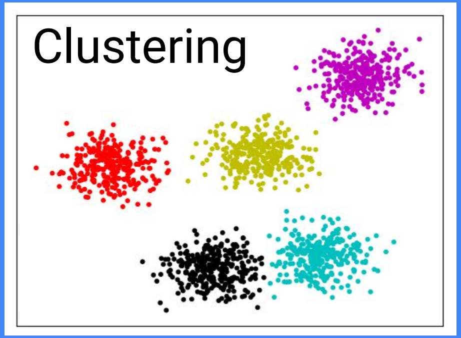
Data

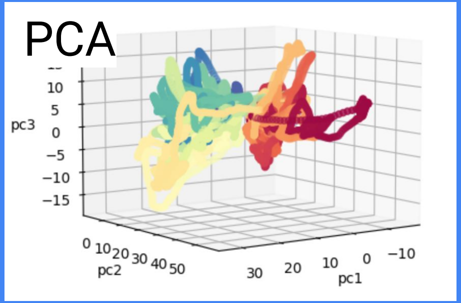
Label




## Unsupervised

Learn patterns from the data without labels.




## Self-Supervised

Learn patterns from the data using the data itself as a label.

# Examples of Learning Paradigms

## Supervised

Learn a function y = f(x) that predicts an output (label) from input (data)

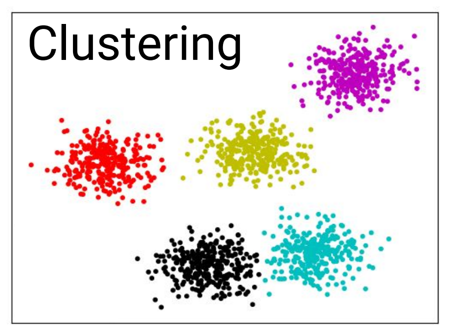
Data

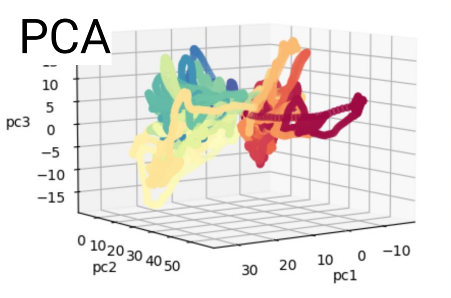
Label

Apple

## Unsupervised

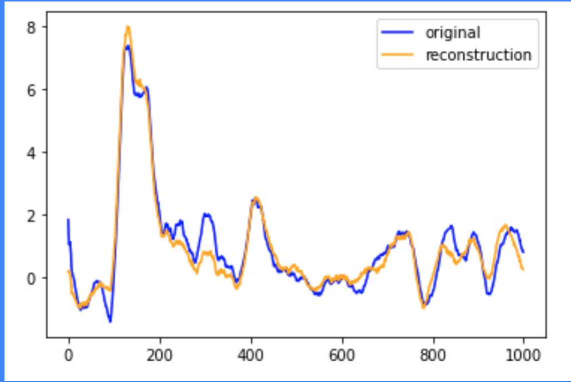
Learn patterns from the data without labels.




## Self-Supervised

Learn patterns from the data using the data itself as a label.



All of the learning paradigms utilize features in the data to accomplish their task.

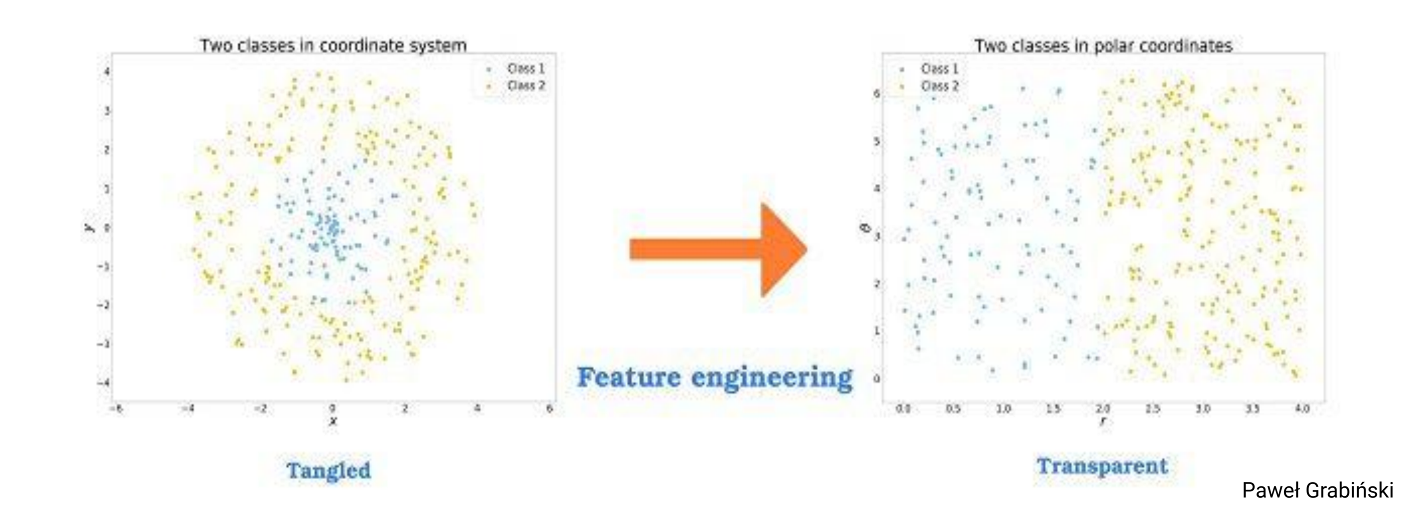
A common theme in machine learning is that the representation of your data matters.

# A common theme in machine learning is that the representation of your data matters.

Sometimes we can hand craft our data features.

# A common theme in machine learning is that the representation of your data matters.

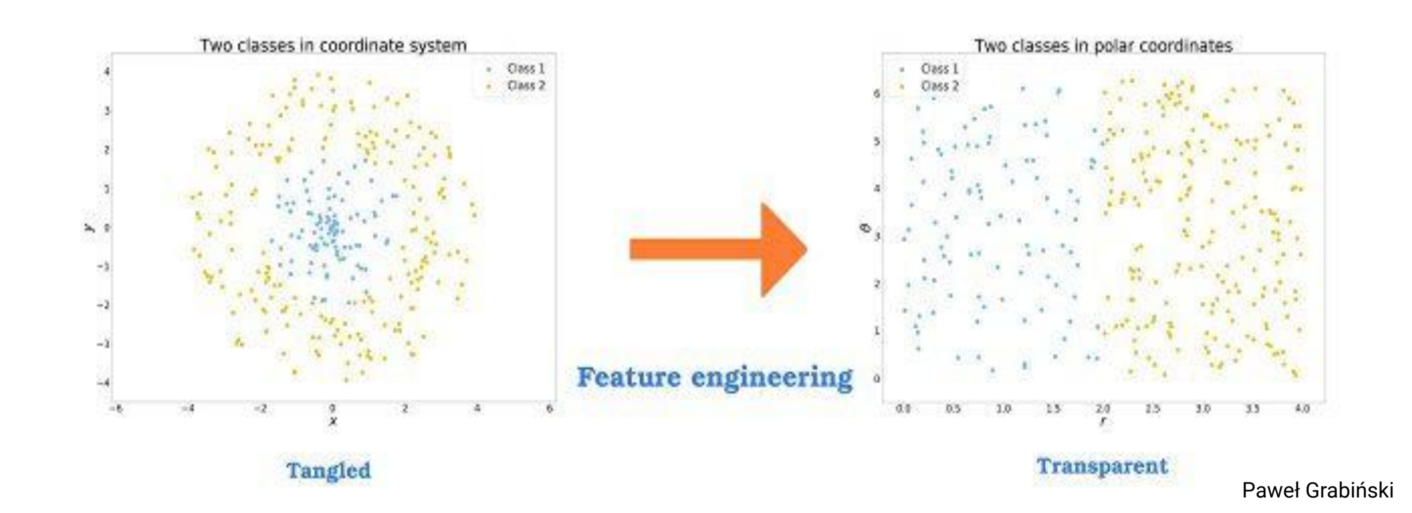
Sometimes we can hand craft our data features.




Paweł Grabiński

# A common theme in machine learning is that the representation of your data matters.

Sometimes we can hand craft our data features.




Other times we let our algorithms learn the important features in our data (aka representation learning).

# But what exactly is a representation?

# But what exactly is a representation?

An encoding of your input data.

But what exactly is a representation?

An encoding of your input data.

What makes a representation good?

## But what exactly is a representation?

An encoding of your input data.

## What makes a representation good?

It depends on the downstream task!

# But what exactly is a representation?

An encoding of your input data.

# What makes a representation good?

It depends on the downstream task!

Ideally your representation is useful for solving many tasks & interpretable (can be very challenging).

Now that we know a little bit more about types of learning & data representations, the let’s jump into the components of learning!

# Example: Supervised Learning

# Example: Supervised Learning

$$\boxed{\{x_i, y_i\}_{i=1}^{N}}$$

Data

# Example: Supervised Learning

$$\boxed{\{x_i, y_i\}_{i=1}^{N}} \longrightarrow \boxed{\hat{y} = f_\theta(x)}$$

Data Model

# Example: Supervised Learning

$$\boxed{\{x_i, y_i\}_{i=1}^{N}} \longrightarrow \boxed{\hat{y} = f_\theta(x)} \longrightarrow \boxed{\ell = \mathcal{L}(y, \hat{y})}$$

Data | Model | Loss

# Example: Supervised Learning

$$\underbrace{\boxed{\{x_i, y_i\}_{i=1}^N}}_{\text{Data}} \longrightarrow \underbrace{\boxed{\hat{y} = f_\theta(x)}}_{\text{Model}} \longrightarrow \underbrace{\boxed{\ell = \mathcal{L}(y, \hat{y})}}_{\text{Loss}}$$

Use loss to update
model parameters
$\theta$ ("learning")

# A little bit more on models!

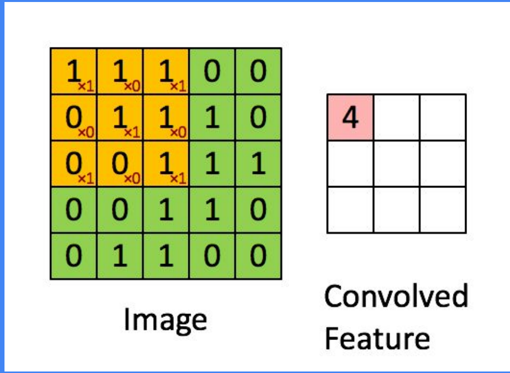
# Models are successful when they exploit domain knowledge in the data

# Models are successful when they exploit domain knowledge in the data

Convolutional Neural Networks → Images

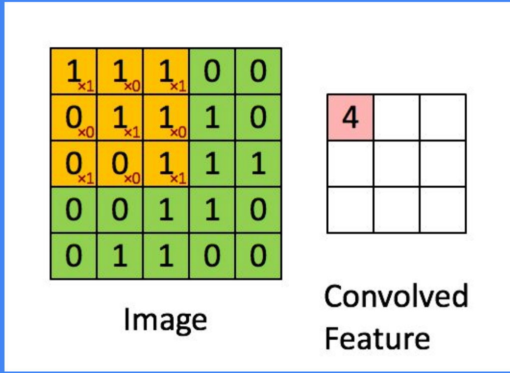
# Models are successful when they exploit domain knowledge in the data

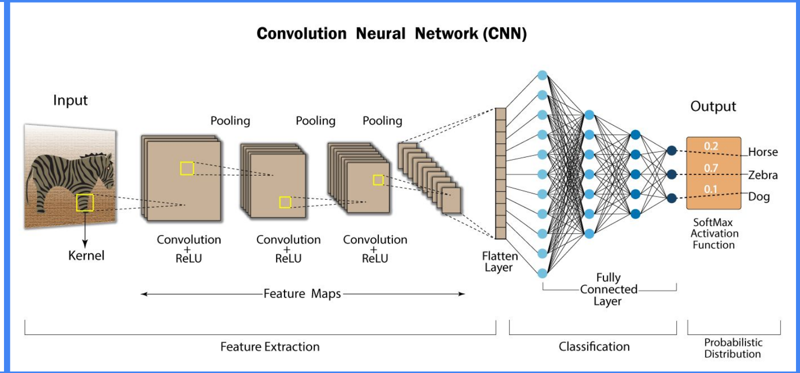
Convolutional Neural Networks → Images



# Models are successful when they exploit domain knowledge in the data

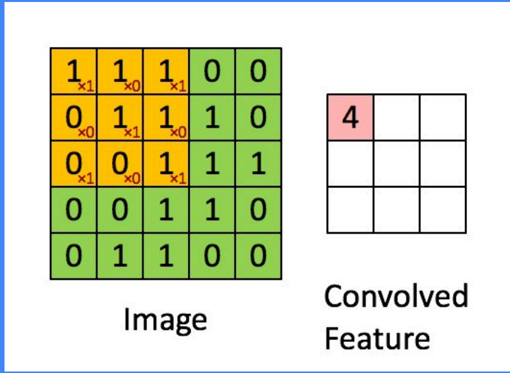
## Convolutional Neural Networks → Images

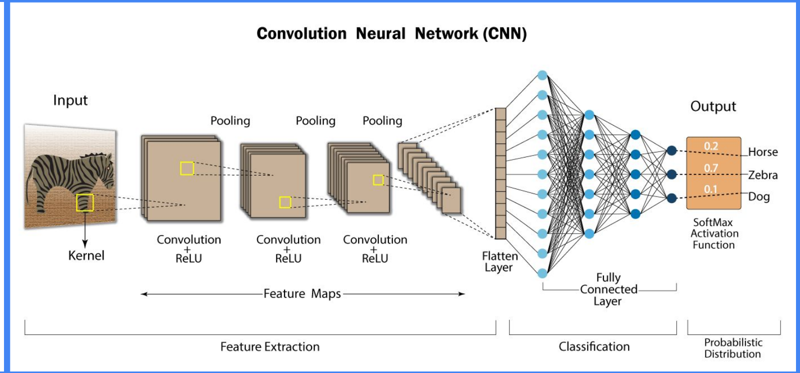


# Models are successful when they exploit domain knowledge in the data

## Convolutional Neural Networks → Images

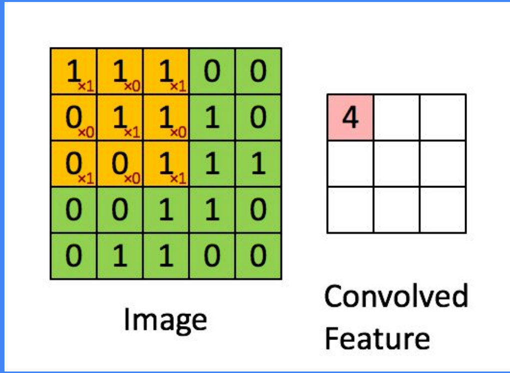


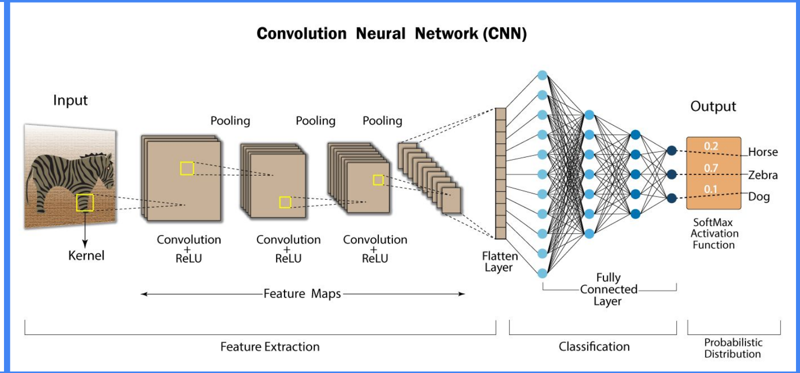




## Recurrent Neural Networks → Sequences

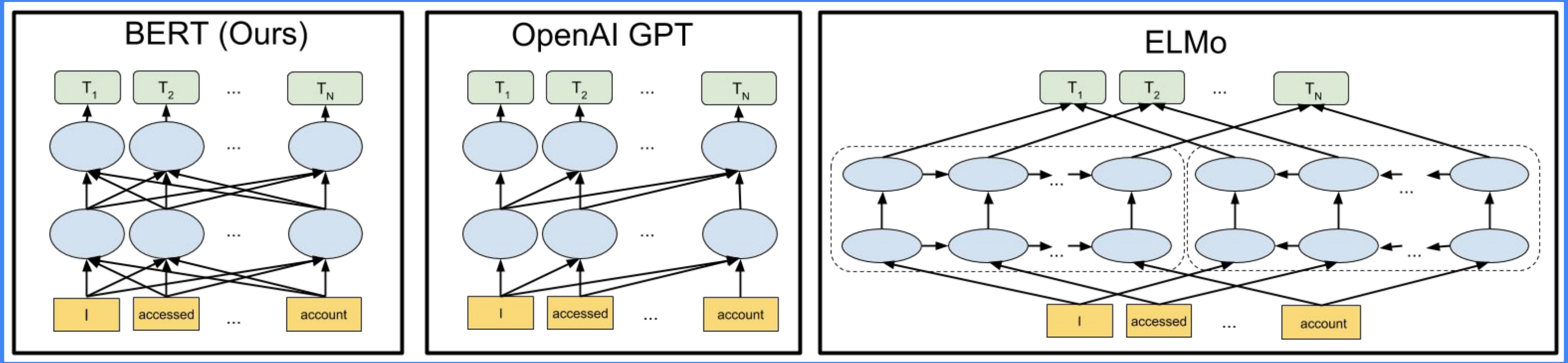
# Models are successful when they exploit domain knowledge in the data

## Convolutional Neural Networks → Images




## Recurrent Neural Networks → Sequences



What are some examples of neuroscience domain knowledge that we could bake into our models?

# A little bit more on losses (with an example)!

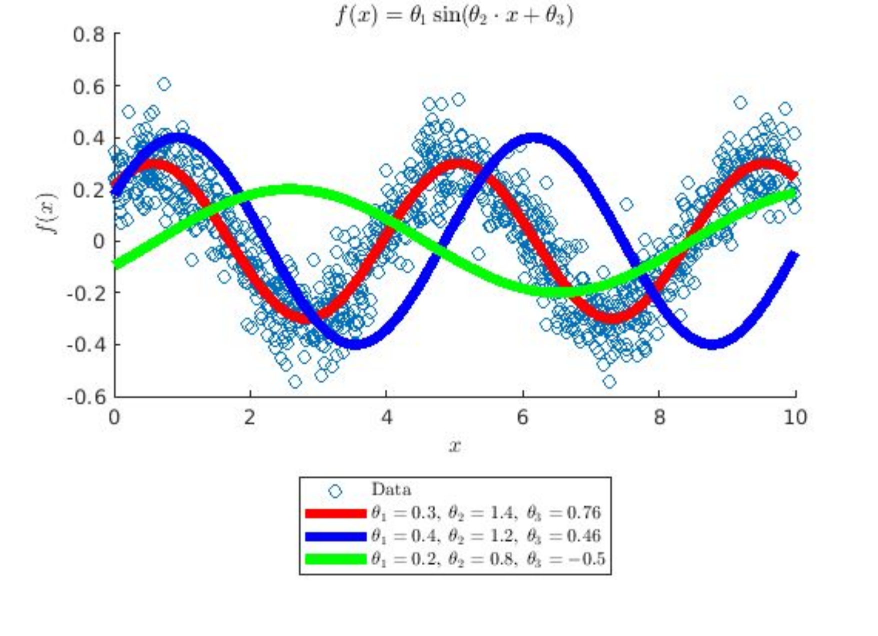
We’re going to explore losses in the context of *parametric models* (models where all information is represented within its parameters).

We’re going to explore losses in the context of *parametric models* (models where all information is represented within its parameters).

Q: For fixed data, what are the best parameters to explain the data?

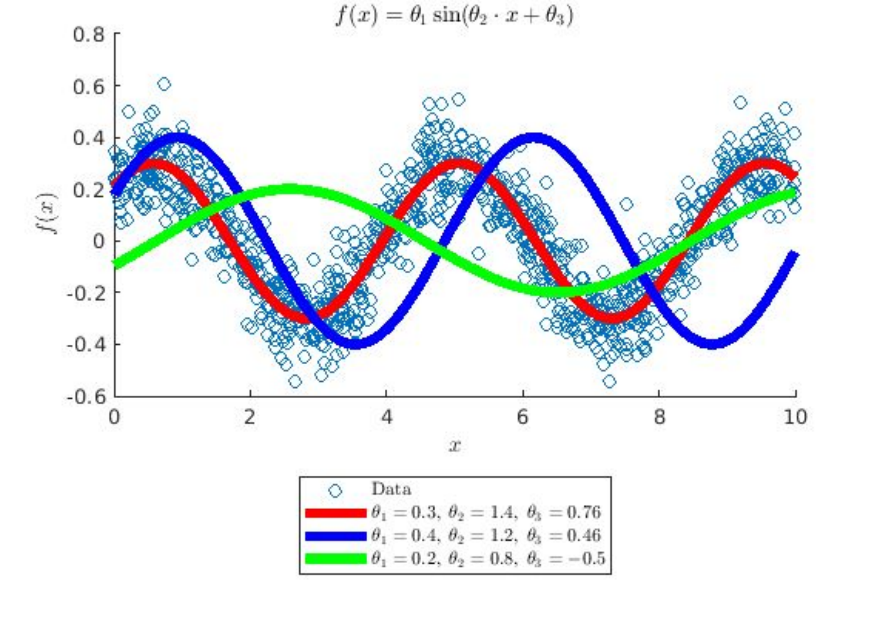
# Ex: 3-Parameter Sinusoidal Model

# Ex: 3-Parameter Sinusoidal Model



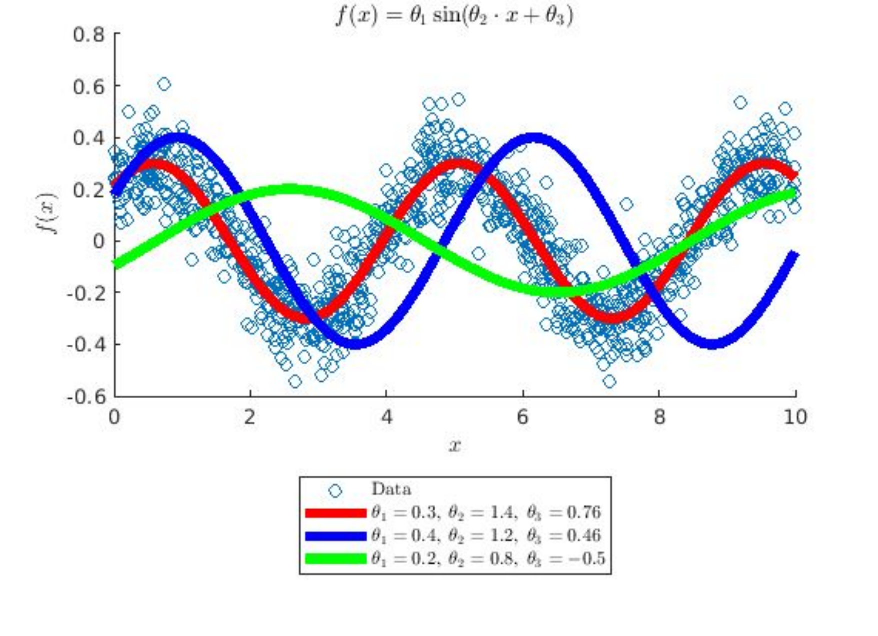

The data are the blue dots. We generated them by sampling from the red line and adding noise.

# Ex: 3-Parameter Sinusoidal Model




The data stays fixed, "learning" this model means picking the three parameter values.

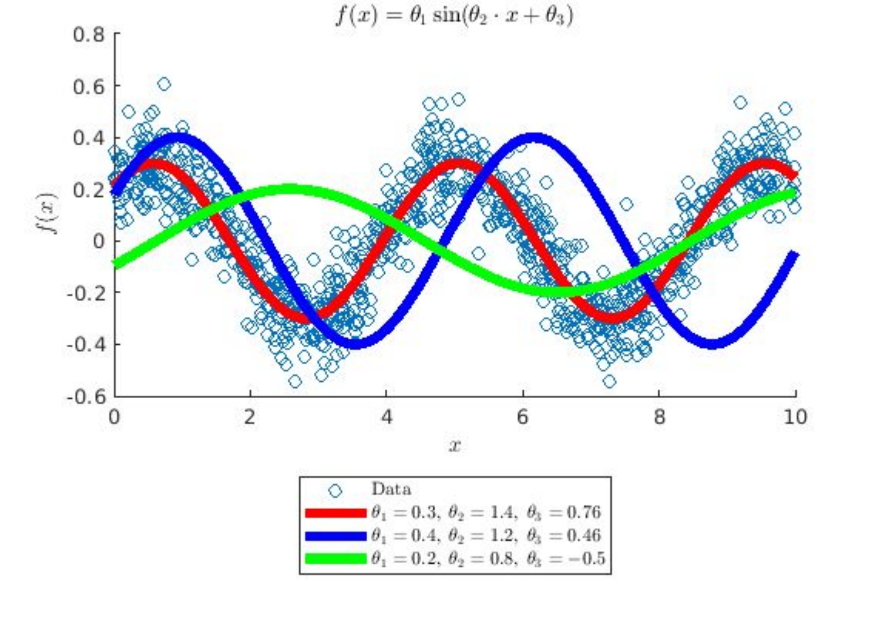
# Ex: 3-Parameter Sinusoidal Model




The data stays fixed, "learning" this model means picking the three parameter values.

Q: How do we pick them?

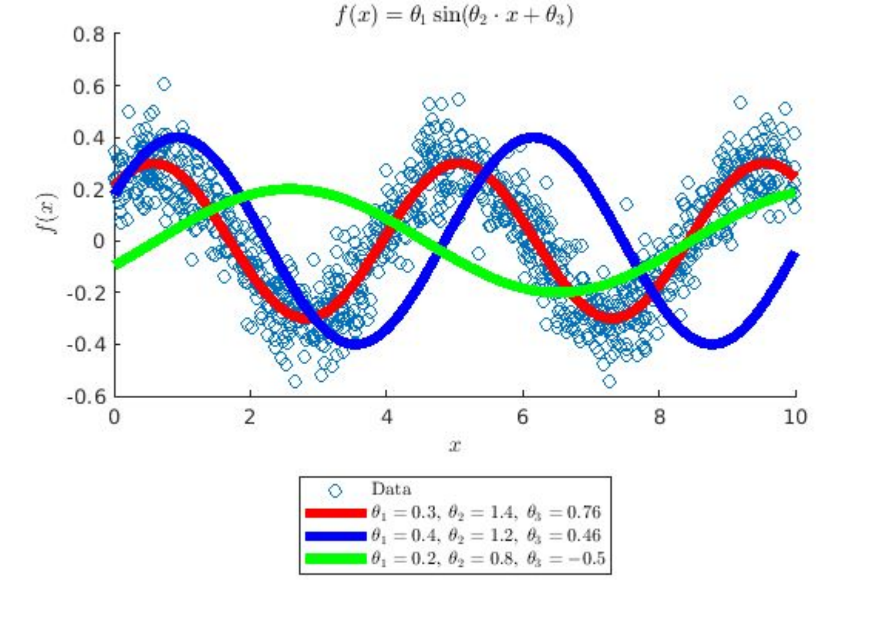
# Ex: 3-Parameter Sinusoidal Model




The data stays fixed, "learning" this model means picking the three parameter values.

Q: How do we pick them?
A: Gradient descent.

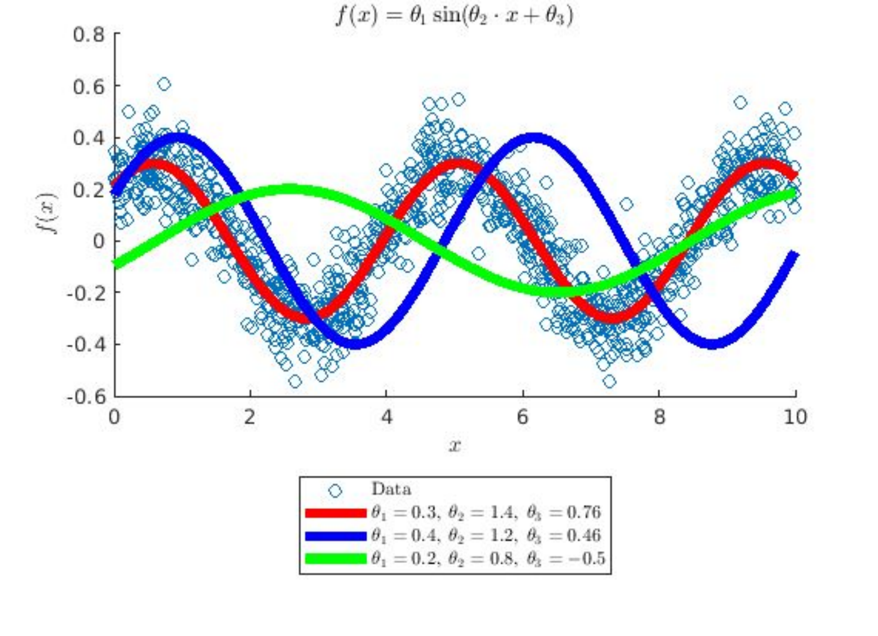
# Ex: 3-Parameter Sinusoidal Model




Loss Values (MSE)

$$\ell = \frac{1}{N} \sum_{i=1}^{N} \|y_i - f_\theta(x_i)\|^2$$

# Ex: 3-Parameter Sinusoidal Model




Loss Values (MSE)

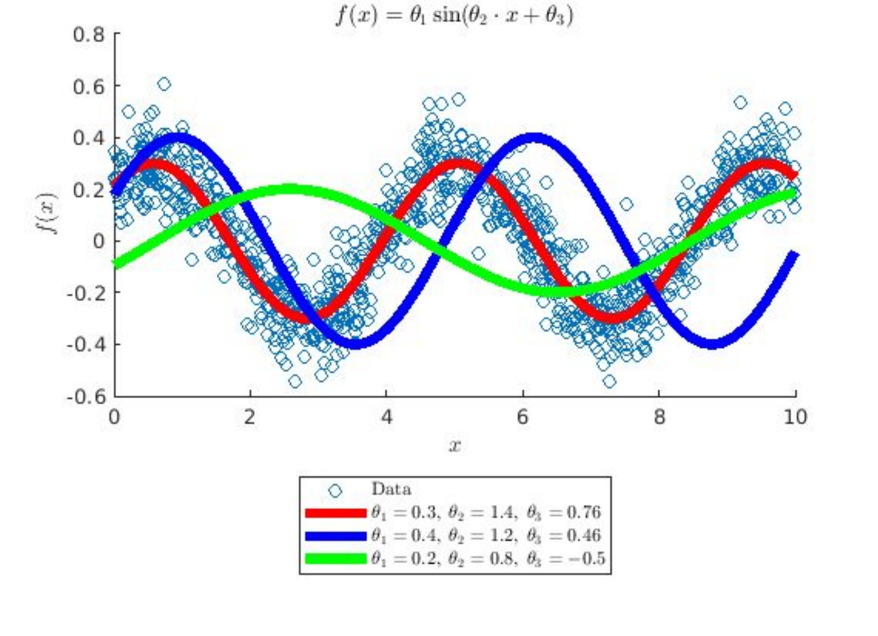
$$\ell = \frac{1}{N}\sum_{i=1}^{N} \|y_i - f_\theta(x_i)\|^2$$

$$\ell_1 = 0.01$$

$$\ell_2 = 0.11$$

$$\ell_3 = 0.08$$

# Ex: 3-Parameter Sinusoidal Model




## Loss Gradient Values

$$\nabla_\theta \ell = \frac{1}{N} \sum_{i=1}^{N} -2(y_i - f_\theta(x_i)) \begin{bmatrix} \sin(\theta_2 \cdot x_i + \theta_3) \\ \theta_1 \cdot x_i \cdot \cos(\theta_2 \cdot x_i + \theta_3) \\ \theta_1 \cdot \cos(\theta_2 \cdot x_i + \theta_3) \end{bmatrix}$$

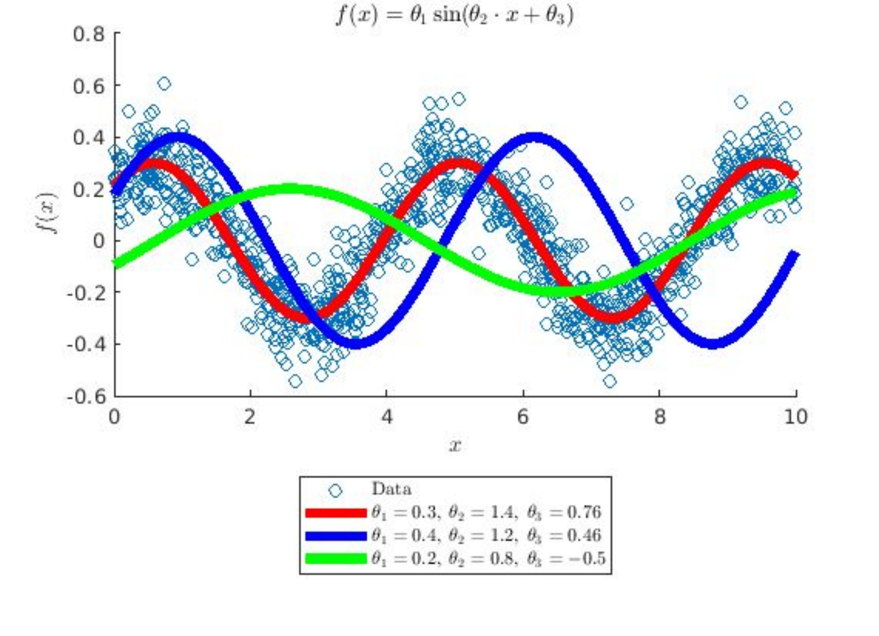
$$\nabla_\theta \ell_1 = \begin{bmatrix} -0.0033 \\ 0.0074 \\ 0.0016 \end{bmatrix}$$

$$\nabla_\theta \ell_2 = \begin{bmatrix} 0.33 \\ -0.66 \\ -0.11 \end{bmatrix}$$

$$\nabla_\theta \ell_3 = \begin{bmatrix} 0.19 \\ 0.049 \\ -0.0004 \end{bmatrix}$$

Element i of the gradient vector is how much the loss changes when changing θi

# Takeaway 1: Gradient values for red model are very small, because it's good!



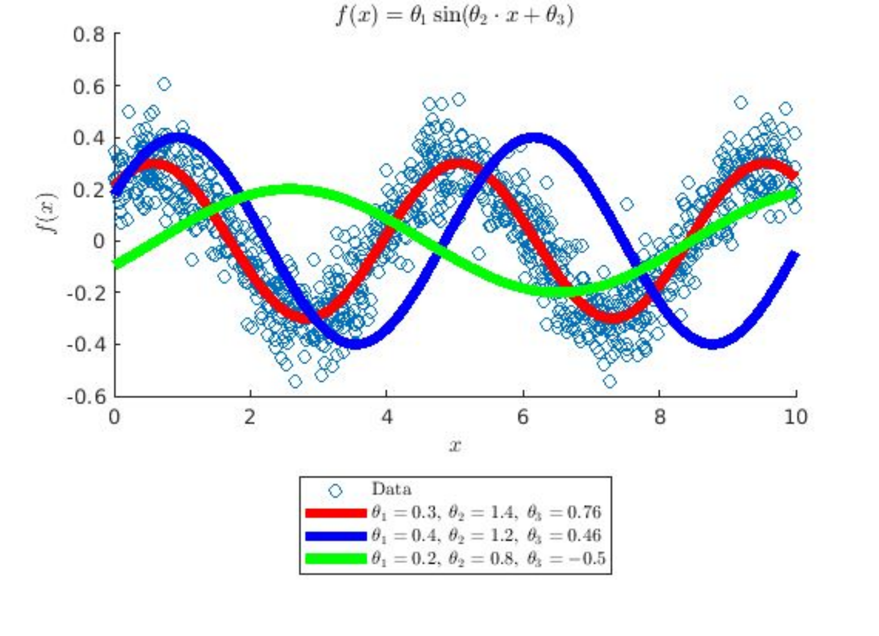

$$\nabla_\theta \ell = \frac{1}{N}\sum_{i=1}^{N} -2(y_i - f_\theta(x_i)) \begin{bmatrix} \sin(\theta_2 \cdot x_i + \theta_3) \\ \theta_1 \cdot x_i \cdot \cos(\theta_2 \cdot x_i + \theta_3) \\ \theta_1 \cdot \cos(\theta_2 \cdot x_i + \theta_3) \end{bmatrix}$$

$$\nabla_\theta \ell_1 = \begin{bmatrix} -0.0033 \\ 0.0074 \\ 0.0016 \end{bmatrix}$$

$$\nabla_\theta \ell_2 = \begin{bmatrix} 0.33 \\ -0.66 \\ -0.11 \end{bmatrix}$$

$$\nabla_\theta \ell_3 = \begin{bmatrix} 0.19 \\ 0.049 \\ -0.0004 \end{bmatrix}$$

# Takeaway 2: Blue and Green models are bad, so the loss changes a lot when we perturb the parameter values.



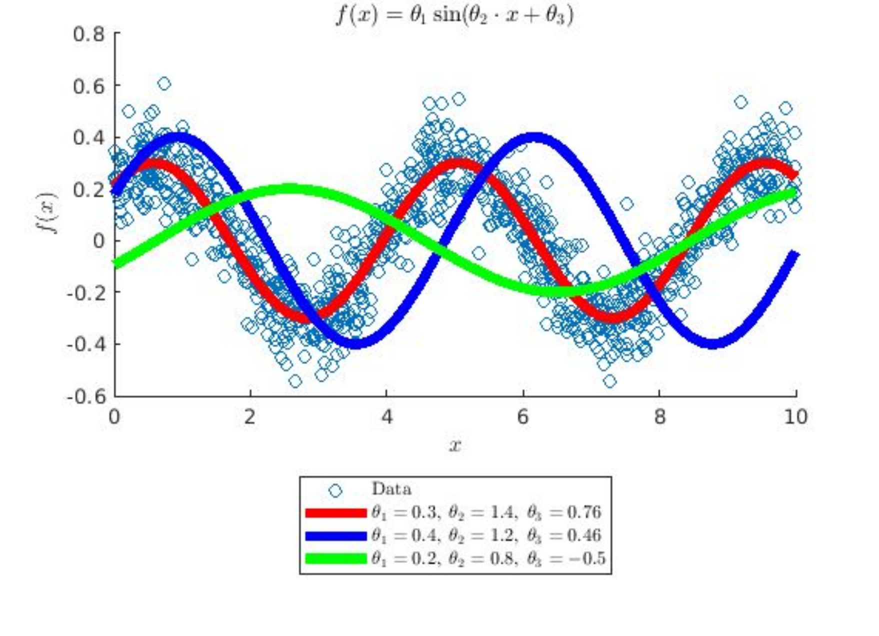

$$\nabla_\theta \ell = \frac{1}{N} \sum_{i=1}^{N} -2(y_i - f_\theta(x_i)) \begin{bmatrix} \sin(\theta_2 \cdot x_i + \theta_3) \\ \theta_1 \cdot x_i \cdot \cos(\theta_2 \cdot x_i + \theta_3) \\ \theta_1 \cdot \cos(\theta_2 \cdot x_i + \theta_3) \end{bmatrix}$$

$$\nabla_\theta \ell_1 = \begin{bmatrix} -0.0033 \\ 0.0074 \\ 0.0016 \end{bmatrix}$$

$$\nabla_\theta \ell_2 = \begin{bmatrix} 0.33 \\ -0.66 \\ -0.11 \end{bmatrix}$$

$$\nabla_\theta \ell_3 = \begin{bmatrix} 0.19 \\ 0.049 \\ -0.0004 \end{bmatrix}$$

Takeaway 3: Gradient information is LOCAL. The third element of the green gradient is very small but that doesn't mean the parameter values are good. It just means that with the current parameters values, changing θ3 slightly isn't really going to change the loss that much.




$$\nabla_\theta \ell = \frac{1}{N}\sum_{i=1}^{N} -2(y_i - f_\theta(x_i)) \begin{bmatrix} \sin(\theta_2 \cdot x_i + \theta_3) \\ \theta_1 \cdot x_i \cdot \cos(\theta_2 \cdot x_i + \theta_3) \\ \theta_1 \cdot \cos(\theta_2 \cdot x_i + \theta_3) \end{bmatrix}$$

$$\nabla_\theta \ell_1 = \begin{bmatrix} -0.0033 \\ 0.0074 \\ 0.0016 \end{bmatrix}$$

$$\nabla_\theta \ell_2 = \begin{bmatrix} 0.33 \\ -0.66 \\ -0.11 \end{bmatrix}$$

$$\nabla_\theta \ell_3 = \begin{bmatrix} 0.19 \\ 0.049 \\ -0.0004 \end{bmatrix}$$

The model family we choose is important! No linear model would ever fit the data well in this regression problem.

We must choose a sufficiently expressive parametric family.

Gradient Descent is the easiest way to optimize our learning objective.

It is a method that changes the weights of your model so you can approach a local minimum.