

General



General

- Everything is on dl4cv.github.io (and Moodle)



General

- Everything is on dl4cv.github.io (and Moodle)
- 4 credit points

General

- Everything is on dl4cv.github.io (and Moodle)
- 4 credit points
- 2 hrs. lecture, ~2 hrs. tutorial (Tutorial covers new material)

General

- Everything is on dl4cv.github.io (and Moodle)
- 4 credit points
- 2 hrs. lecture, ~2 hrs. tutorial (Tutorial covers new material)
- Communication through Moodle

General

- Everything is on dl4cv.github.io (and Moodle)
- 4 credit points
- 2 hrs. lecture, ~2 hrs. tutorial (Tutorial covers new material)
- Communication through Moodle
- 3 homework assignments + Final Project
Homework is demanding, but worth it



General

- Everything is on dl4cv.github.io (and Moodle)
- 4 credit points
- 2 hrs. lecture, ~2 hrs. tutorial (Tutorial covers new material)
- Communication through Moodle
- 3 homework assignments + Final Project
Homework is demanding, but worth it
- Reserve duty etc.: We will support!

”If we have seen further, it is by standing on the shoulders of Giants”



”If we have seen further, it is by standing on the shoulders of Giants”

- From basic to most recent SotA



”If we have seen further, it is by standing on the shoulders of Giants”

- From basic to most recent SotA
- Slightly biased towards Weizmann research



”If we have seen further, it is by standing on the shoulders of Giants”

- From basic to most recent SotA
- Slightly biased towards Weizmann research
- Intuition



”If we have seen further, it is by standing on the shoulders of Giants”

- From basic to most recent SotA
- Slightly biased towards Weizmann research
- Intuition
- Hands-on



”If we have seen further, it is by standing on the shoulders of Giants”

- From basic to most recent SotA
- Slightly biased towards Weizmann research
- Intuition
- Hands-on
- Openness



We Assume you...



We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).

We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).

We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).
- Written code before (preferably Python).

We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).
- Written code before (preferably Python).
- Bumped into Machine Learning (e.g. heard the term “Overfitting”).

We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).
- Written code before (preferably Python).
- Bumped into Machine Learning (e.g. heard the term “Overfitting”).

Homework

We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).
- Written code before (preferably Python).
- Bumped into Machine Learning (e.g. heard the term “Overfitting”).

Homework



Theory

We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).
- Written code before (preferably Python).
- Bumped into Machine Learning (e.g. heard the term “Overfitting”).

Homework

Theory

From Scratch

We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).
- Written code before (preferably Python).
- Bumped into Machine Learning (e.g. heard the term “Overfitting”).

Homework

Theory

From Scratch

Applied

We Assume you...

- Know Basic Calculus (e.g. know what is a Gradient).
- Know Basic Algebra (e.g. Vector spaces, Matrix multiplication, Eigen decomposition).
- Written code before (preferably Python).
- Bumped into Machine Learning (e.g. heard the term “Overfitting”).

Homework

Theory

From Scratch

Applied

**HW1 is
online!**

Road map

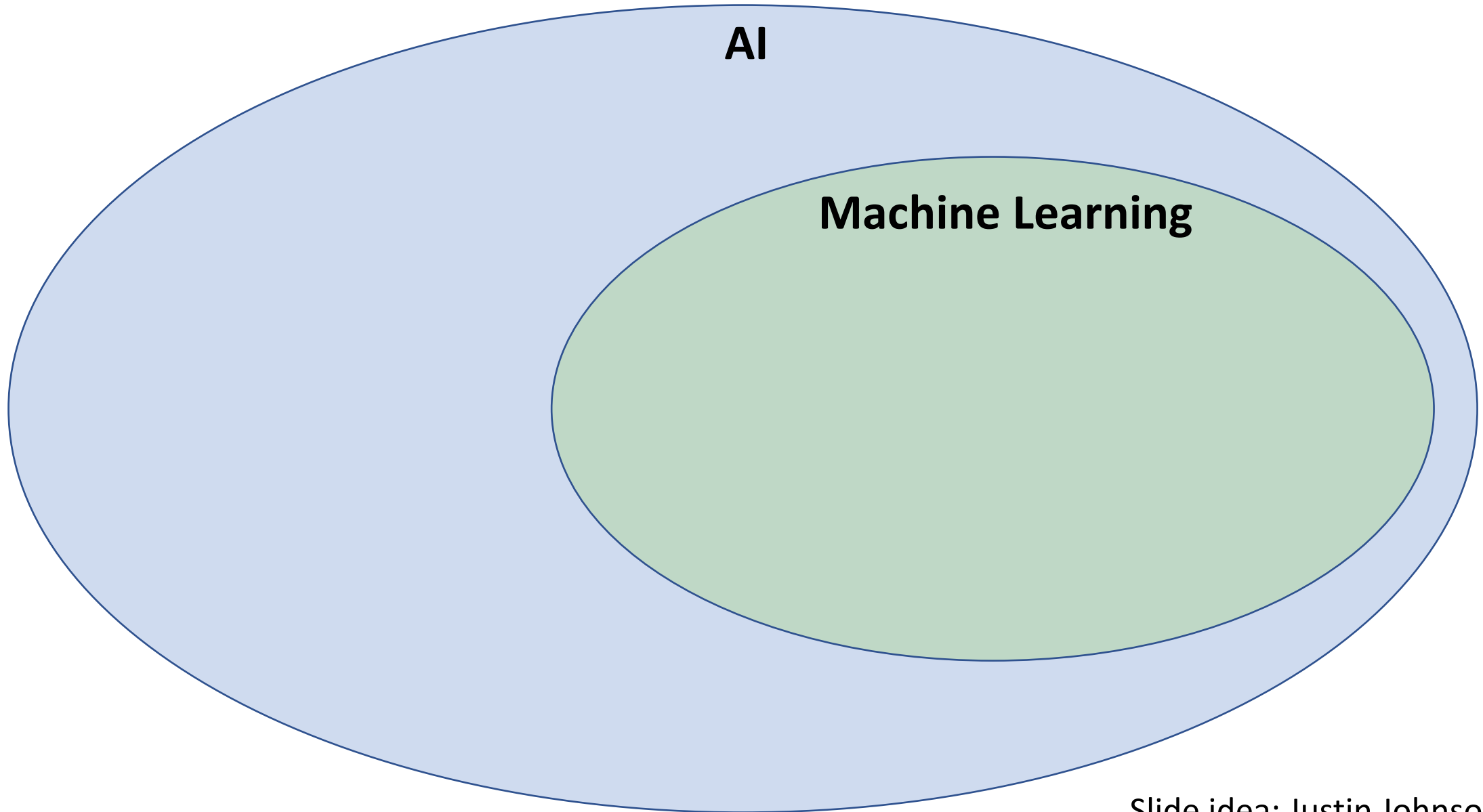


Road map

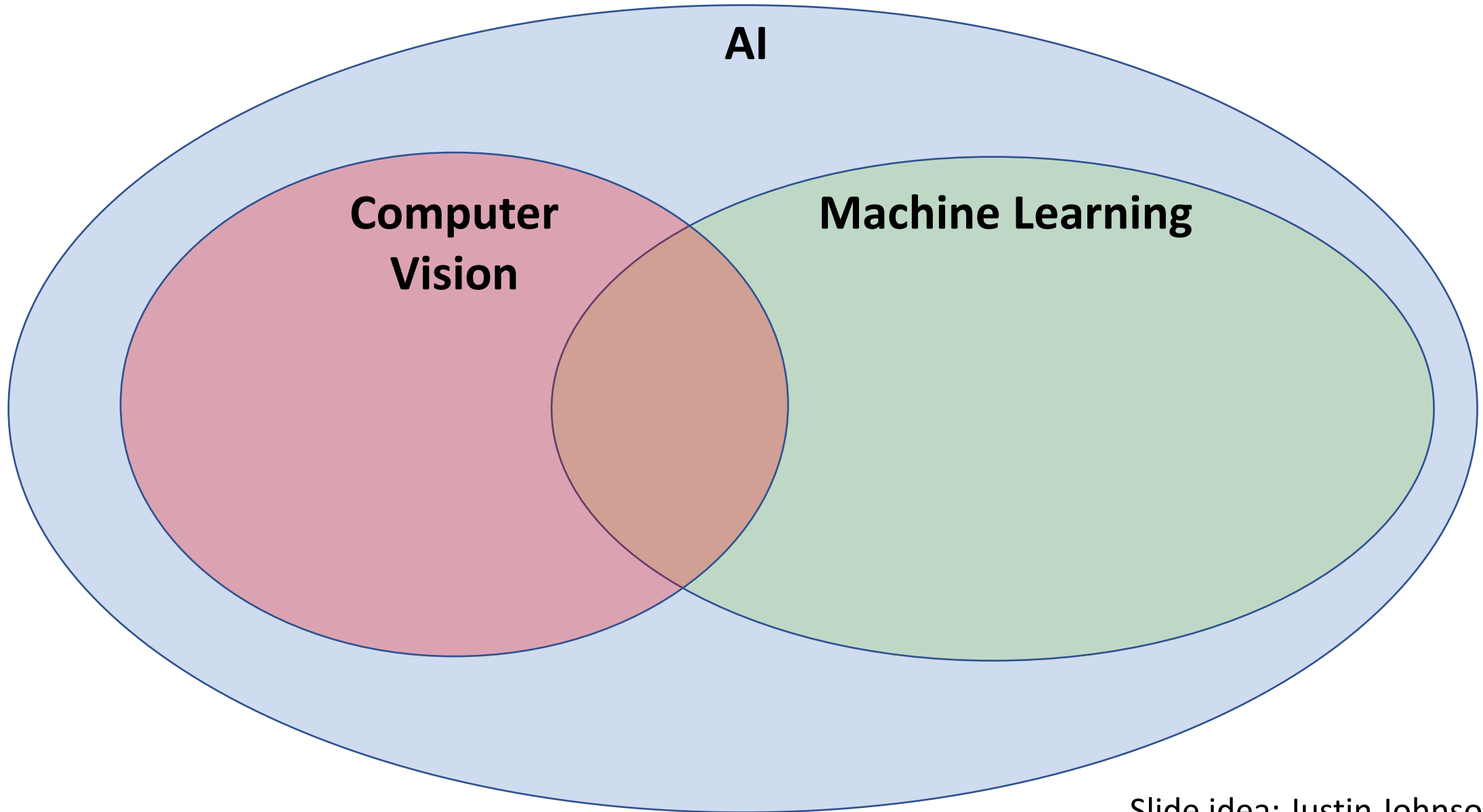
AI



Road map



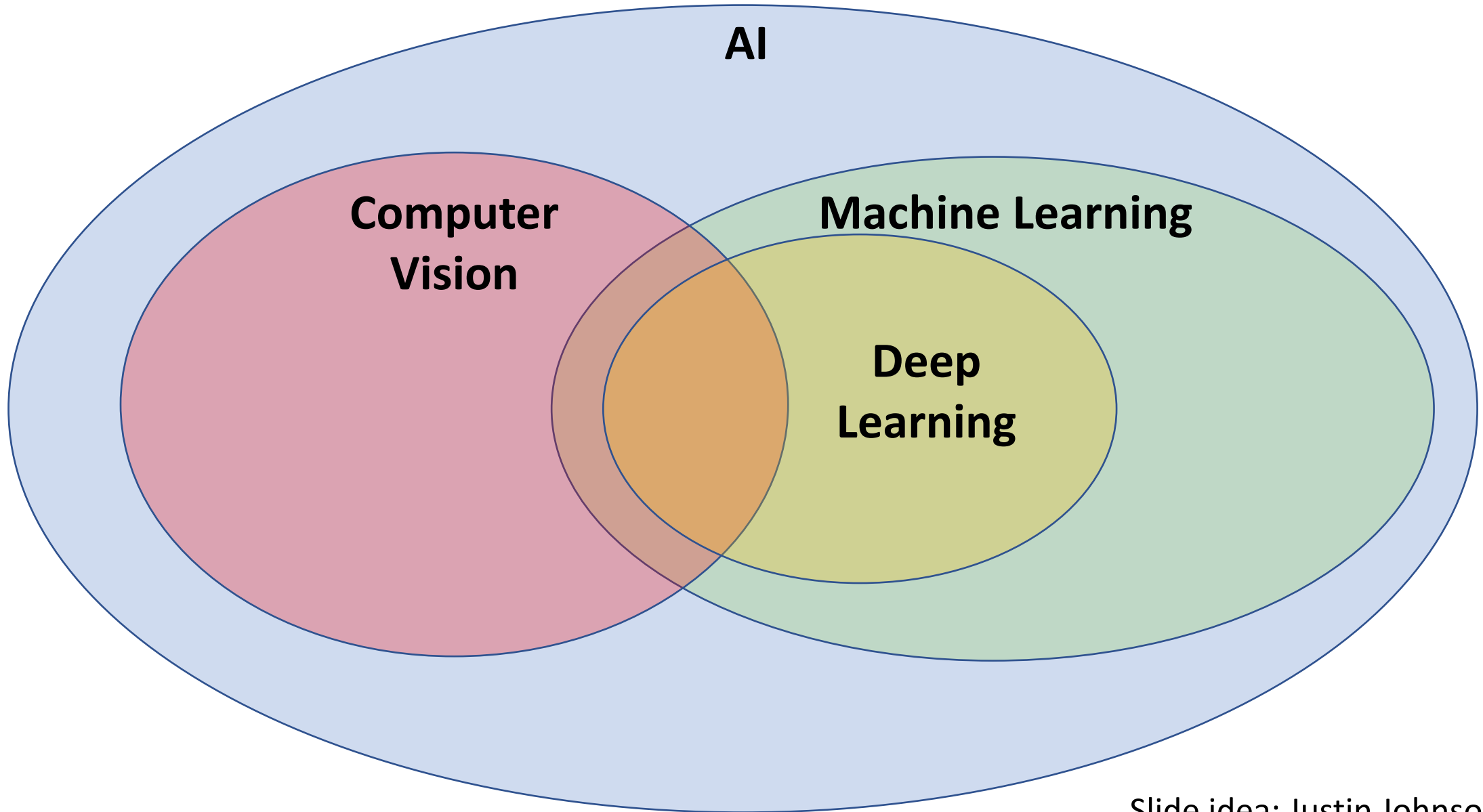
Road map



Slide idea: Justin Johnson



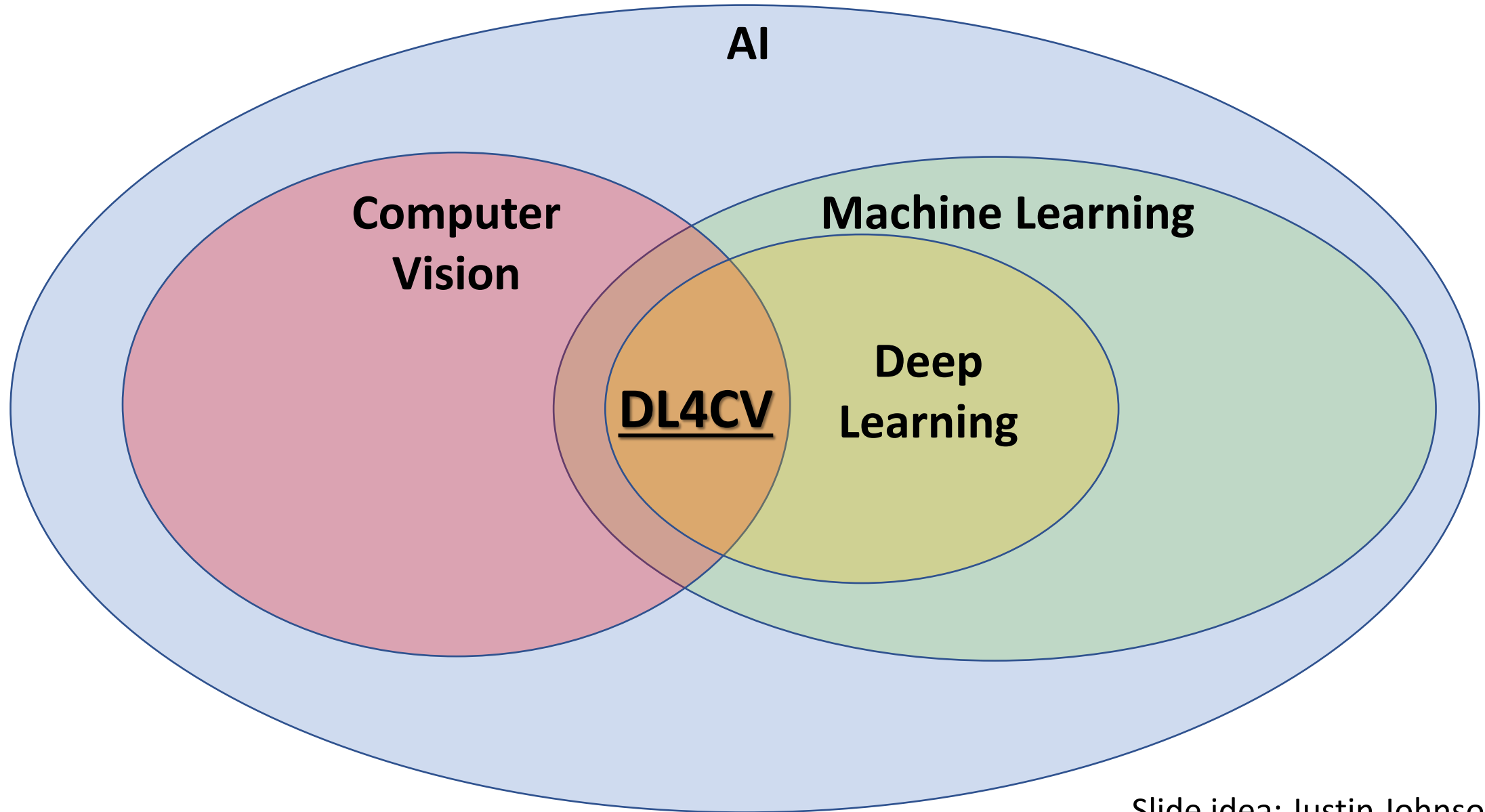
Road map



Slide idea: Justin Johnson



Road map



Today:



Today:

- Motivation and history (15%)

Today:

- Motivation and history (15%)
- Supervised learning (25%)

Today:

- Motivation and history (15%)
- Supervised learning (25%)
- Linear regression (20%)

Today:

- Motivation and history (15%)
- Supervised learning (25%)
- Linear regression (20%)
- Gradient descent (25%)



Today:

- Motivation and history (15%)
- Supervised learning (25%)
- Linear regression (20%)
- Gradient descent (25%)
- Feature transform (15%)

Deep Learning is powerful



"girl in pink dress is jumping in air."



"black and white dog jumps over bar."



"young girl in pink shirt is swinging on swing."

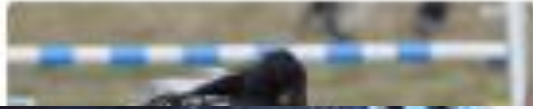


"man in blue wetsuit is surfing on wave."

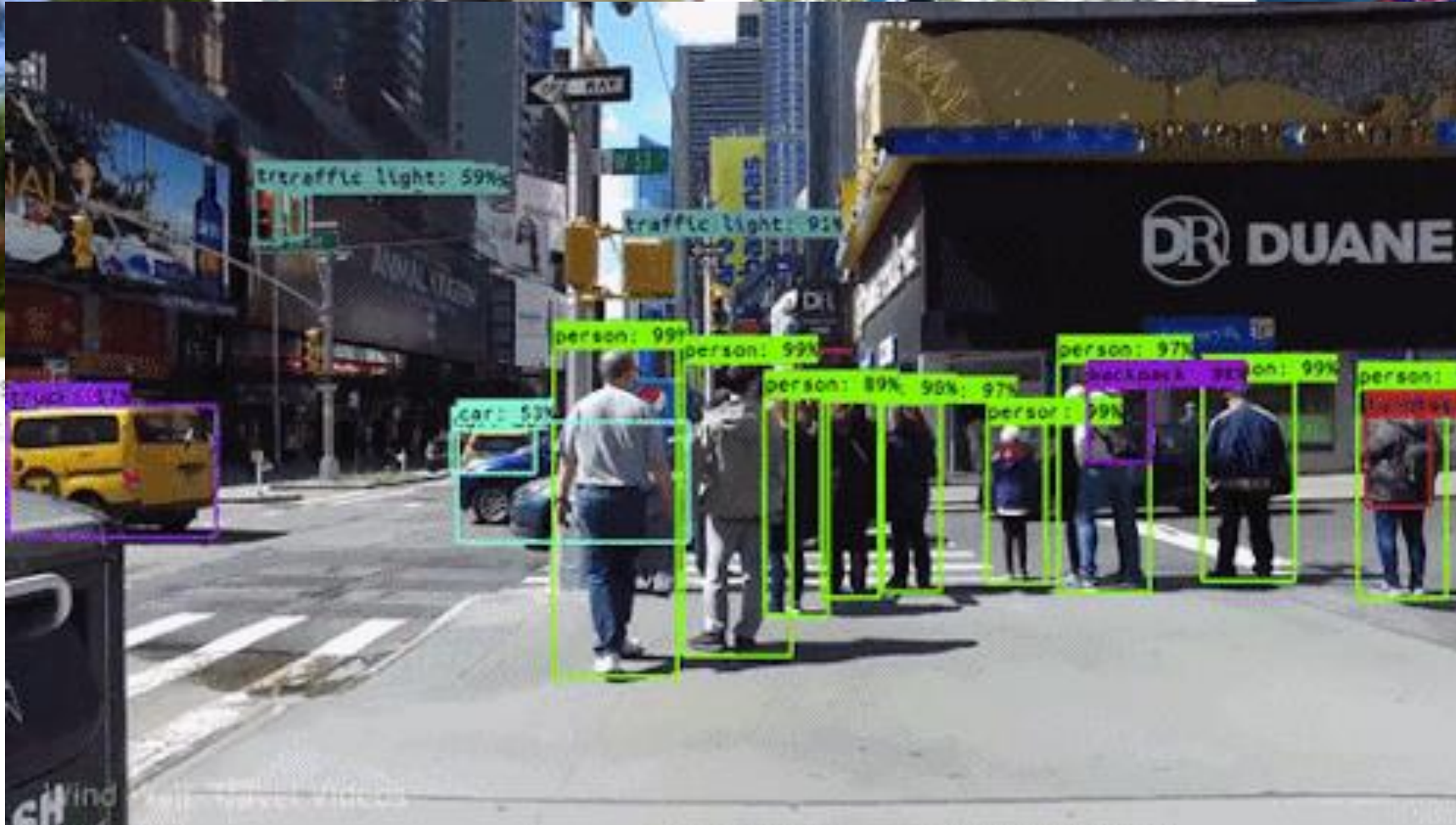
Deep Learning is powerful



"girl in pink dress
air"



suit is surfing on
ve."



Andrej Karpathy, Li Fei-Fei, CVPR 2015 Deep Visual-Semantic
Alignments for Generating Image Descriptions

Abhishek Bansal- DetectMe (GitHub)





You

generate the picture depicting the joke: a snail on the back of a turtle saying "woohoo"





You

generate the picture depicting the joke: a snail on the back of a turtle saying "woohoo"



ChatGPT



Here's the image depicting the joke: a snail on the back of a turtle, exclaiming "woohoo!" The scene captures the snail's enthusiasm contrasted with the turtle's calm and steady pace. I hope you find it amusing! Let me know if there's anything else I can assist you with!





You

generate the picture depicting the joke: a snail on the back of a turtle saying "woohoo"



ChatGPT



Here's the image depicting the joke: a snail on the back of a turtle, exclaiming "woohoo!" The scene captures the snail's enthusiasm contrasted with the turtle's calm and steady pace. I hope you find it amusing! Let me know if there's anything else I can assist you with!



You

thats very good, but I want it to look like the snail feels the speed. maybe some wind on his skin or anything else that indicates that he feels he is going fast



You

generate the picture depicting the joke: a snail on the back of a turtle saying "woohoo"



ChatGPT



Here's the image depicting the joke: a snail on the back of a turtle, exclaiming "woohoo!" The scene captures the snail's enthusiasm contrasted with the turtle's calm and steady pace. I hope you find it amusing! Let me know if there's anything else I can assist you with!

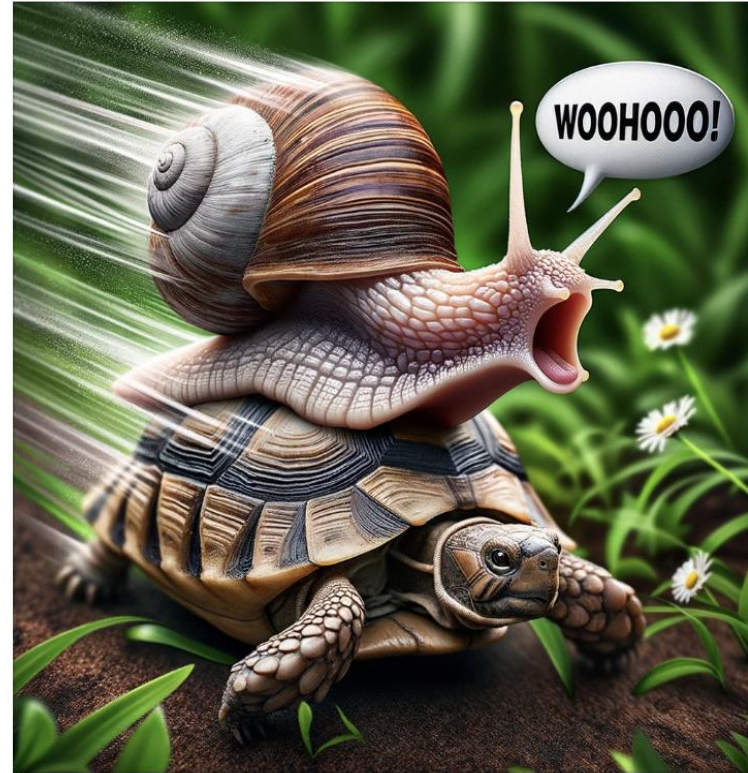


You

thats very good, but I want it to look like the snail feels the speed. maybe some wind on his skin or anything else that indicates that he feels he is going fast



ChatGPT



Here's the revised image where the snail truly feels the thrill of speed! The motion blur and the wind effects emphasize the snail's perceived rapid movement, even though the turtle remains at its usual slow pace. I hope this captures the humor and essence of the joke better. Let me know if there's anything else I can assist you with!





History of:

- Computer Science
- Mathematics
- Neuroscience

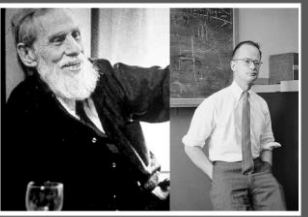


- Computer vision
- Machine learning





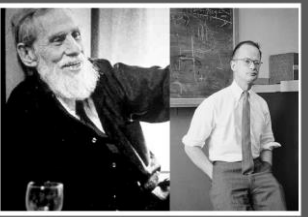
McCulloch Pitts
Non learned
Neuron



1943

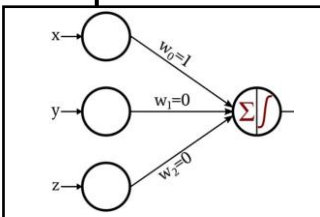


McCulloch Pitts
Non learned
Neuron



1957

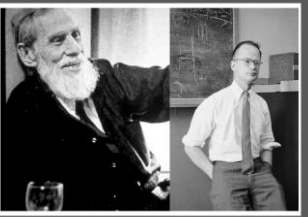
1943



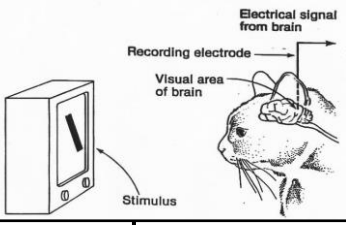
Perceptron
(Rosenblatt)
learned weights



McCulloch Pitts
Non learned
Neuron

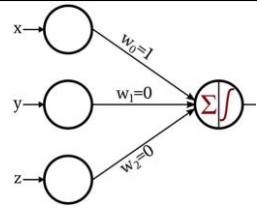


Hubel & Wiesel



1957

1943 1959

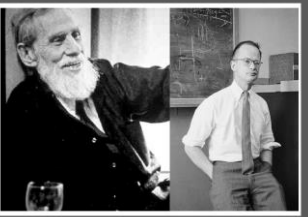


Perceptron
(Rosenblatt)
learned weights

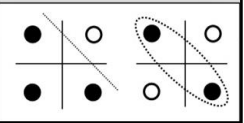


McCulloch Pitts
Non learned
Neuron

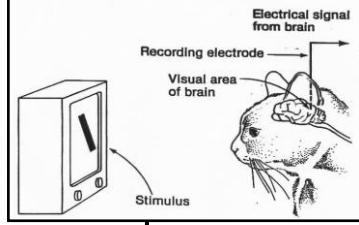
XOR kills
perceptron



M. Minsky - S. Papert



Hubel & Wiesel



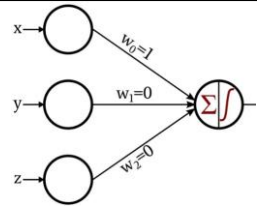
1957



1943

1959

1969



Perceptron
(Rosenblatt)
learned weights

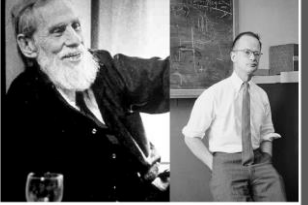
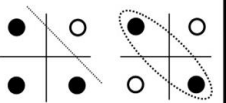


McCulloch Pitts
Non learned
Neuron

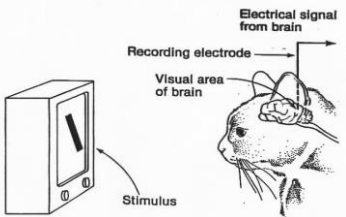
XOR kills
perceptron



M. Minsky - S. Papert



Hubel & Wiesel

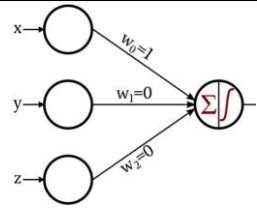


1957

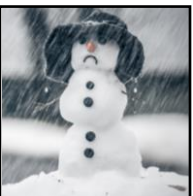
1969

1943

1959



Perceptron
(Rosenblatt)
learned weights



1st
AI Winter

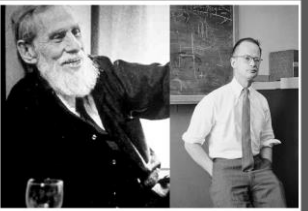
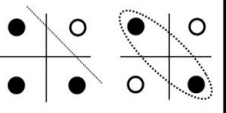


McCulloch Pitts
Non learned
Neuron

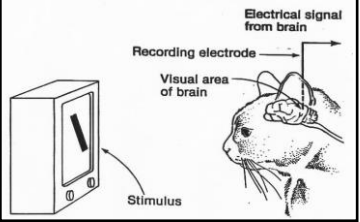
XOR kills
perceptron



M. Minsky - S. Papert



Hubel & Wiesel



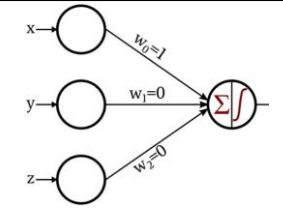
1957

1970

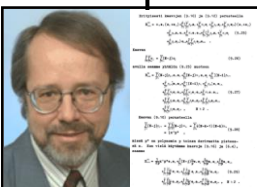
1943

1959

1969



Perceptron
(Rosenblatt)
learned weights



Back
Propagation
(Linnainmaa)



1st
AI Winter

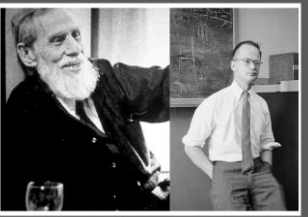
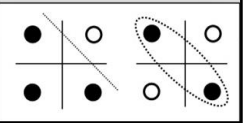


McCulloch Pitts
Non learned
Neuron

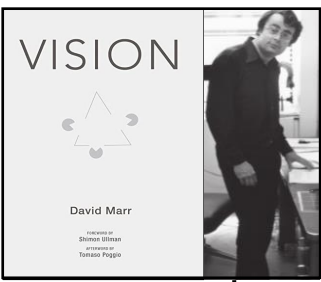
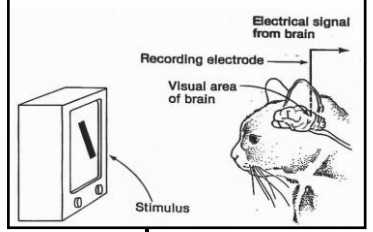
XOR kills
perceptron



M. Minsky - S. Papert



Hubel & Wiesel



1957

1970

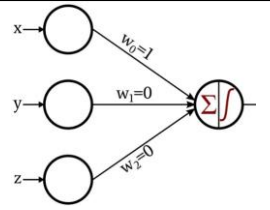


1943

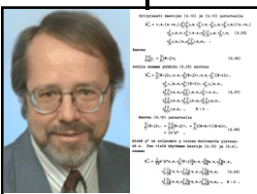
1959

1969

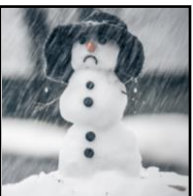
1979



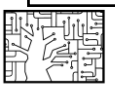
Perceptron
(Rosenblatt)
learned weights



Back
Propagation
(Linnainmaa)



1st
AI Winter



VISION

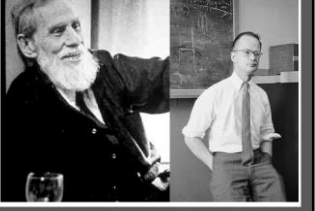
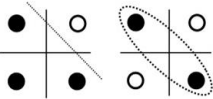


McCulloch Pitts
Non learned
Neuron

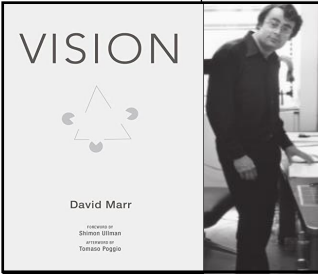
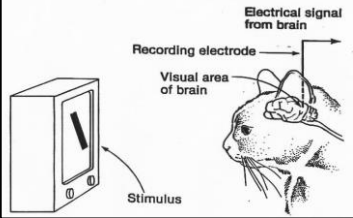
XOR kills
perceptron



M. Minsky - S. Papert



Hubel & Wiesel



1957

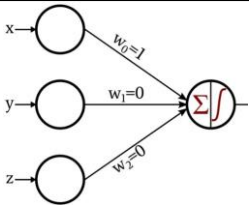
1970

1943

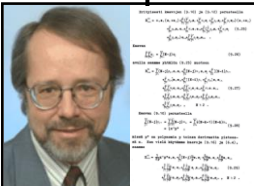
1959

1969

1979



Perceptron
(Rosenblatt)
learned weights



Back
Propagation
(Linnainmaa)



1st
AI Winter



David Marr

FOREWORD BY
Shimon Ullman

AFTERWORD BY
Tomaso Poggio

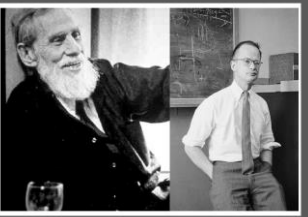
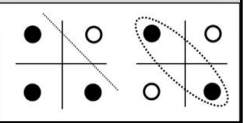


McCulloch Pitts
Non learned
Neuron

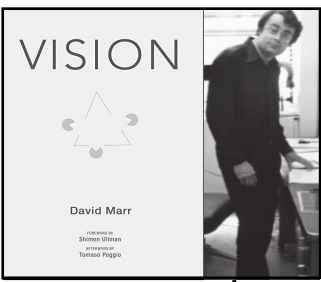
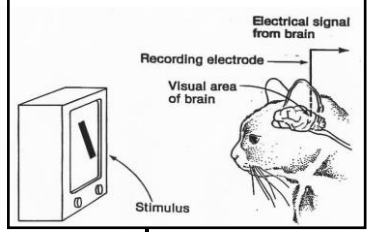
XOR kills
perceptron



M. Minsky - S. Papert



Hubel & Wiesel



1957

1970

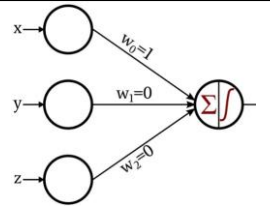


1943

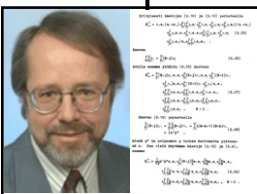
1959

1969

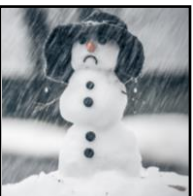
1979



Perceptron
(Rosenblatt)
learned weights



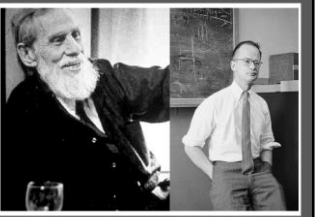
Back
Propagation
(Linnainmaa)



1st
AI Winter



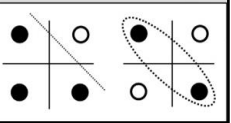
McCulloch Pitts
Non learned
Neuron



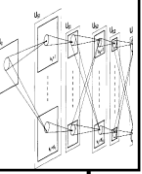
XOR kills
perceptron



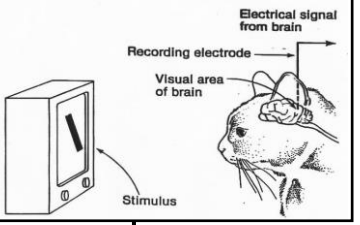
M. Minsky - S. Papert



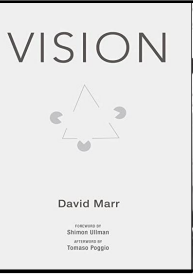
Neocognitron
(Fukushima)
First CNN?
No backprop



Hubel & Wiesel



1957



1970

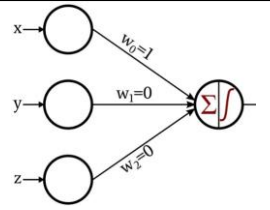
1979

1980

1943

1959

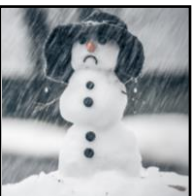
1969



Perceptron
(Rosenblatt)
learned weights



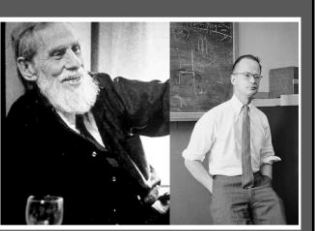
Back
Propagation
(Linnainmaa)



1st
AI Winter



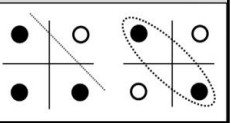
McCulloch Pitts
Non learned
Neuron



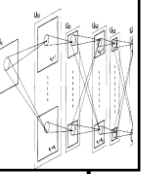
XOR kills
perceptron



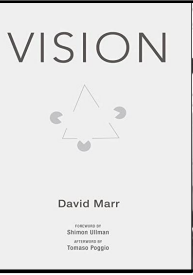
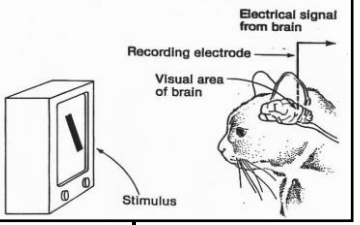
M. Minsky - S. Papert



Neocognitron
(Fukushima)
First CNN?
No backprop



Hubel & Wiesel



David Marr

1957

1970

1986



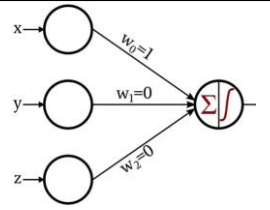
1943

1959

1969

1979

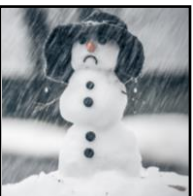
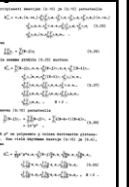
1980



Perceptron
(Rosenblatt)
learned weights



Back
Propagation
(Linnainmaa)



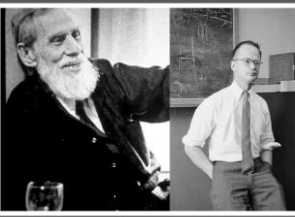
1st
AI Winter




Backprop in
Neural Nets
demonstrated
(Hinton et al.)



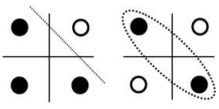
McCulloch Pitts
Non learned
Neuron



XOR kills
perceptron



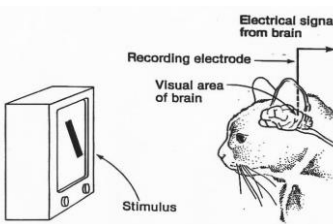
M. Minsky - S. Papert



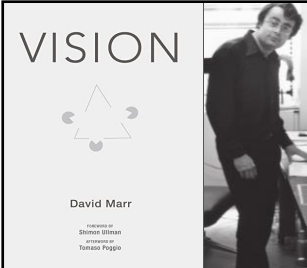
Neocognitron
(Fukushima)
First CNN?
No backprop




Hubel & Wiesel



VISION



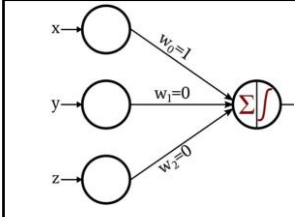
David Marr





2nd AI Winter



Perceptron
(Rosenblatt)
learned weights



Back
Propagation
(Linnainmaa)

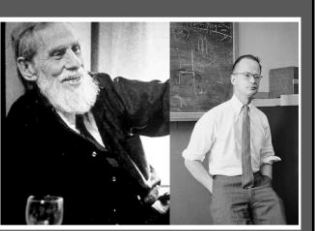
1st
AI Winter



Backprop in
Neural Nets
demonstrated
(Hinton et al.)



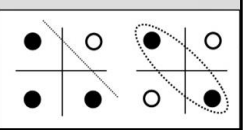

McCulloch Pitts
Non learned
Neuron



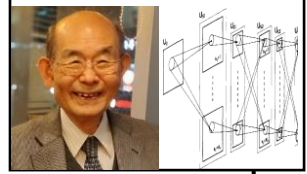
XOR kills
perceptron



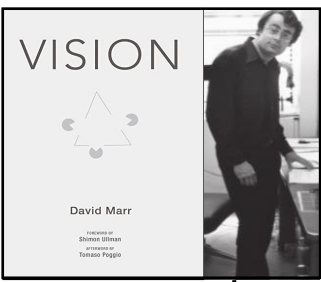
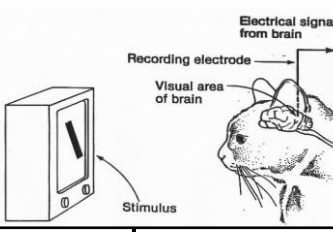
M. Minsky - S. Papert



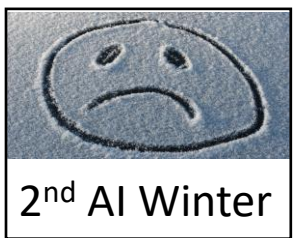
Neocognitron
(Fukushima)
First CNN?
No backprop



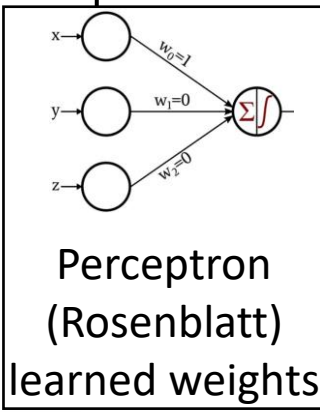
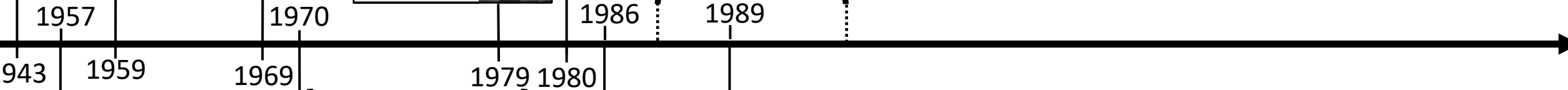
Hubel & Wiesel



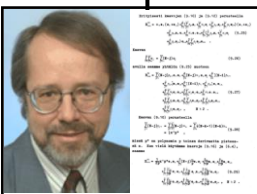
David Marr



2nd AI Winter



Perceptron
(Rosenblatt)
learned weights



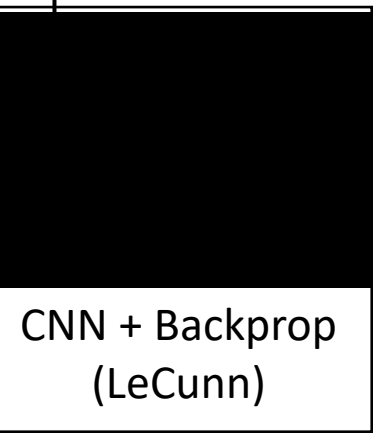
Back
Propagation
(Linnainmaa)



1st
AI Winter

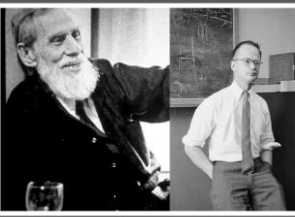


Backprop in
Neural Nets
demonstrated
(Hinton et al.)




CNN + Backprop
(LeCun)

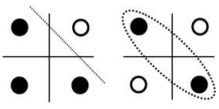
McCulloch Pitts
Non learned
Neuron



XOR kills
perceptron



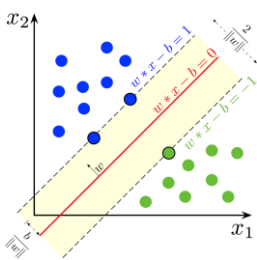
M. Minsky - S. Papert



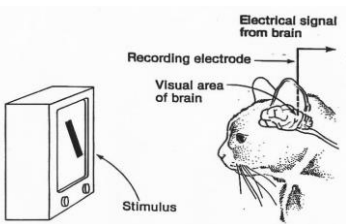
Neocognitron
(Fukushima)
First CNN?
No backprop



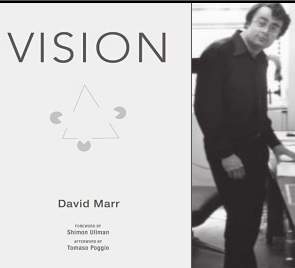

Kernel-SVM
Vapnik&Cortes




Hubel & Wiesel



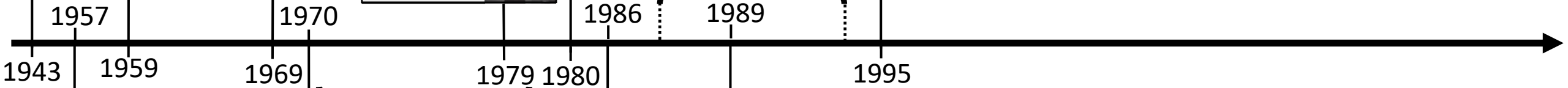
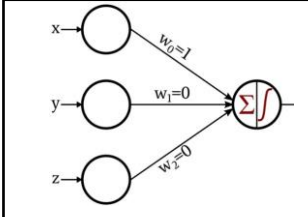
VISION




David Marr



2nd AI Winter


Perceptron
(Rosenblatt)
learned weights



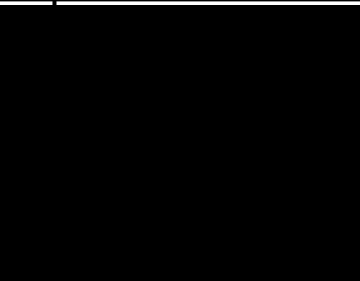
Back
Propagation
(Linnainmaa)



1st
AI Winter



Backprop in
Neural Nets
demonstrated
(Hinton et al.)



CNN + Backprop
(LeCun)

McCulloch Pitts
Non learned
Neuron

XOR kills
perceptron

M. Minsky - S. Papert

Neocognitron
(Fukushima)
First CNN?
No backprop

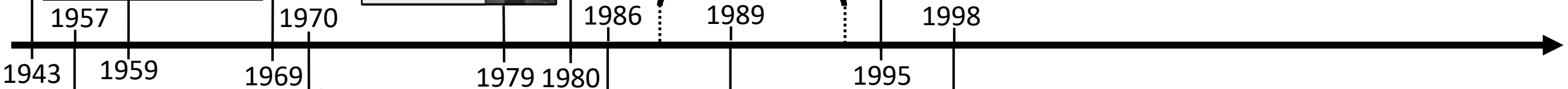
Kernel-SVM
Vapnik&Cortes

Hubel & Wiesel

VISION

David Marr

2nd AI Winter



Perceptron
(Rosenblatt)
learned weights

Back
Propagation
(Linnainmaa)

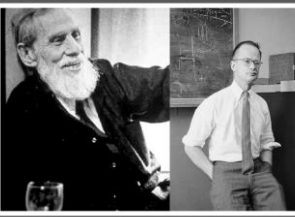
1st
AI Winter

Backprop in
Neural Nets
demonstrated
(Hinton et al.)


CNN + Backprop
(LeCun)

LeNet-5

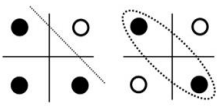
McCulloch Pitts
Non learned
Neuron



XOR kills
perceptron



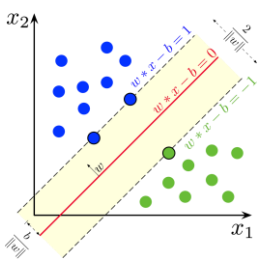
M. Minsky - S. Papert



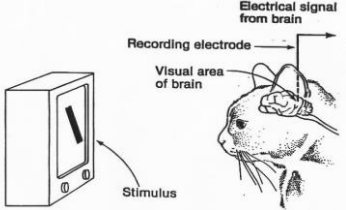
Neocognitron
(Fukushima)
First CNN?
No backprop



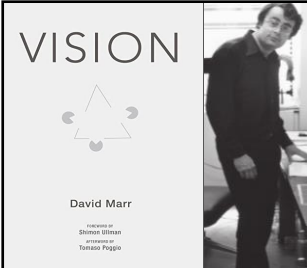

Kernel-SVM
Vapnik&Cortes



Hubel & Wiesel


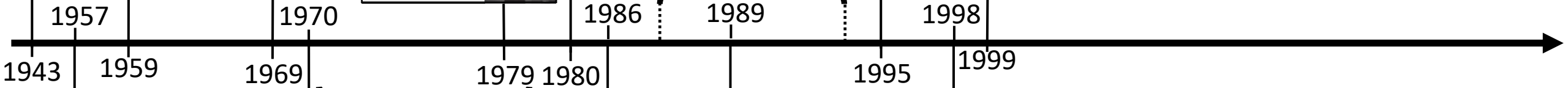


VISION

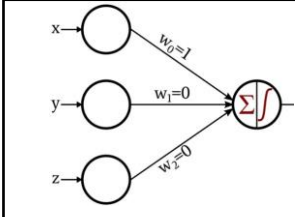


David Marr


2nd AI Winter

Perceptron
(Rosenblatt)
learned weights



Back
Propagation
(Linnainmaa)



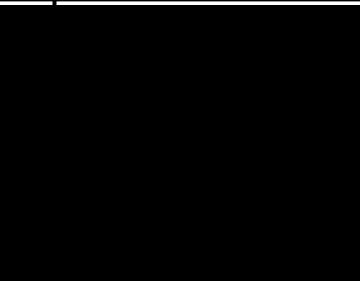
1st
AI Winter



Backprop in
Neural Nets
demonstrated
(Hinton et al.)



CNN + Backprop
(LeCun)



SIFT

LeNet-5

McCulloch Pitts
Non learned
Neuron

XOR kills
perceptron

M. Minsky - S. Papert

Neocognitron
(Fukushima)
First CNN?
No backprop

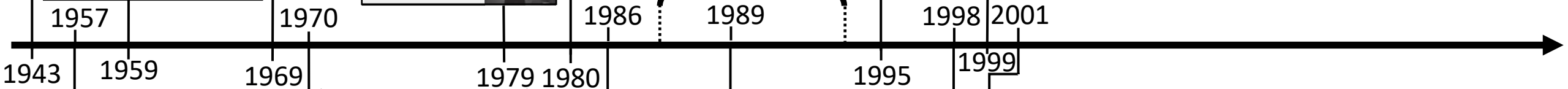
Kernel-SVM
Vapnik&Cortes

Hubel & Wiesel

VISION

David Marr

2nd AI Winter



Perceptron
(Rosenblatt)
learned weights

Back
Propagation
(Linnainmaa)

1st
AI Winter

Backprop in
Neural Nets
demonstrated
(Hinton et al.)

CNN + Backprop
(LeCun)

SIFT

LeNet-5

Rand. Forest

McCulloch Pitts
Non learned
Neuron

XOR kills
perceptron

M. Minsky - S. Papert

Neocognitron
(Fukushima)
First CNN?
No backprop

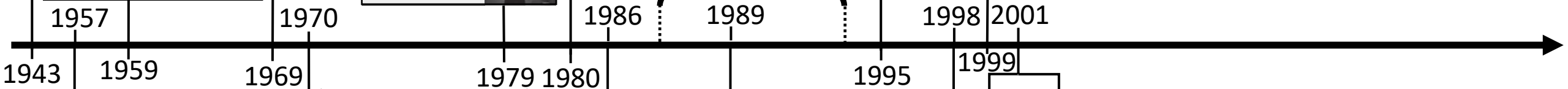
Kernel-SVM
Vapnik&Cortes

Hubel & Wiesel

VISION

David Marr

2nd AI Winter



Perceptron
(Rosenblatt)
learned weights

Back
Propagation
(Linnainmaa)

1st
AI Winter

Backprop in
Neural Nets
demonstrated
(Hinton et al.)

CNN + Backprop
(LeCun)

LeNet-5

Rand. Forest

Viola & Jones

SIFT

McCulloch Pitts
Non learned
Neuron

XOR kills
perceptron

M. Minsky - S. Papert

Neocognitron
(Fukushima)
First CNN?
No backprop

Kernel-SVM
Vapnik&Cortes

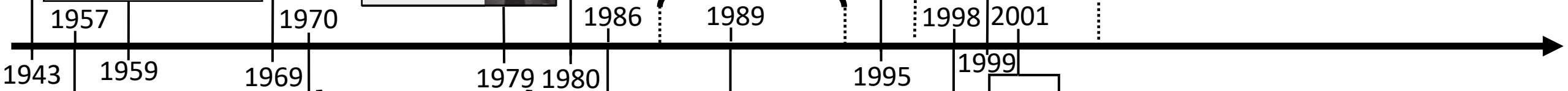
Hubel & Wiesel

VISION

David Marr

2nd AI Winter

Neural
Winter?



Perceptron
(Rosenblatt)
learned weights

Back
Propagation
(Linnainmaa)

1st
AI Winter

Backprop in
Neural Nets
demonstrated
(Hinton et al.)

CNN + Backprop
(LeCun)

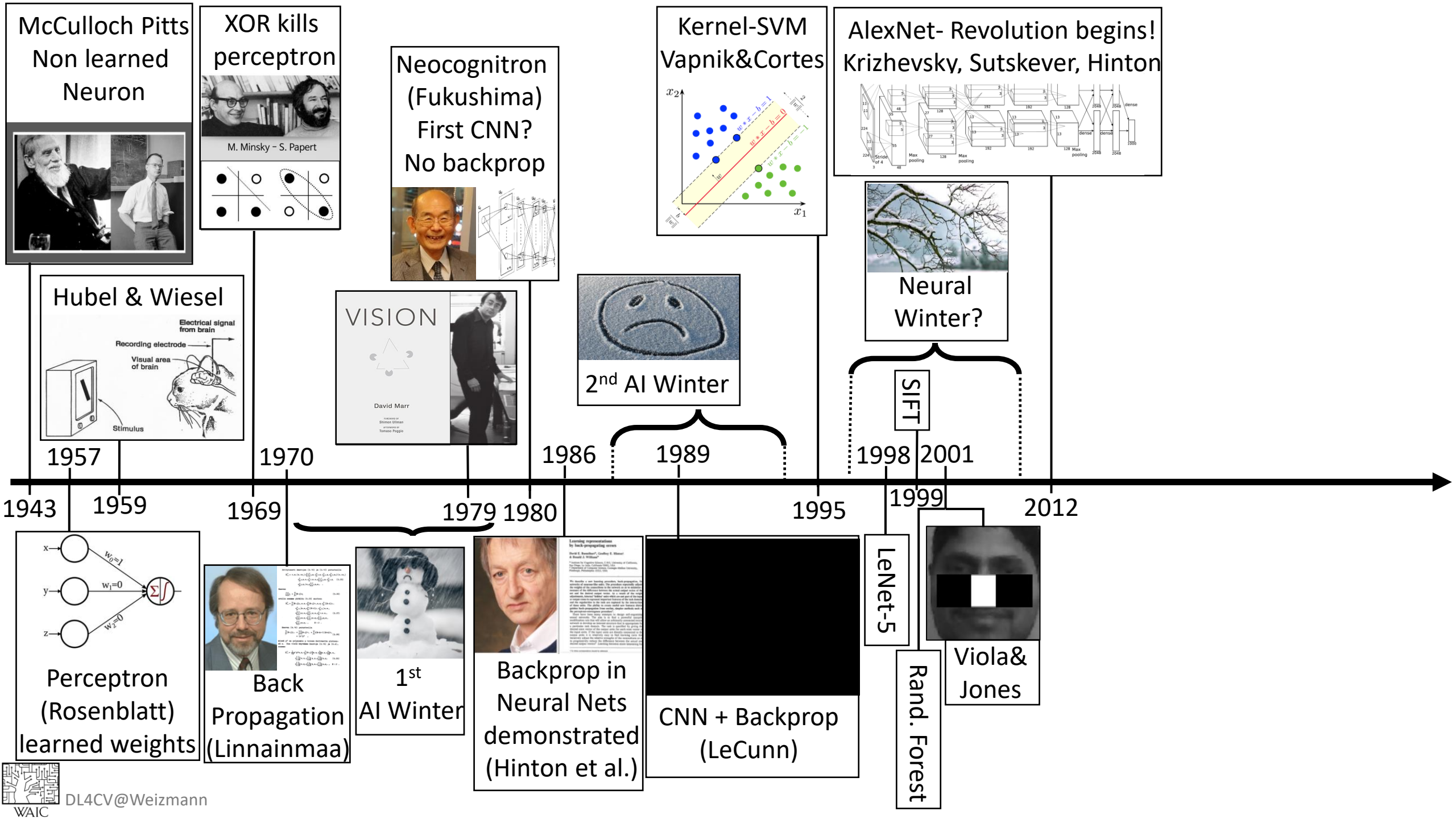
LeNet-5

Rand. Forest

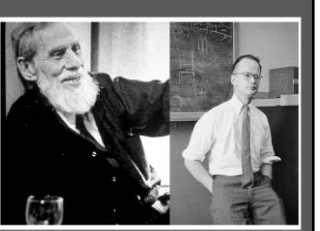
Viola &
Jones

SIFT

1998 2001



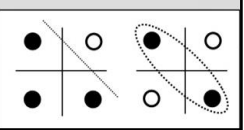
McCulloch Pitts
Non learned
Neuron



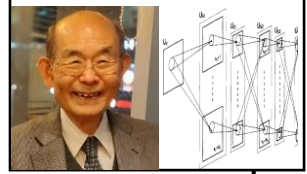
XOR kills
perceptron



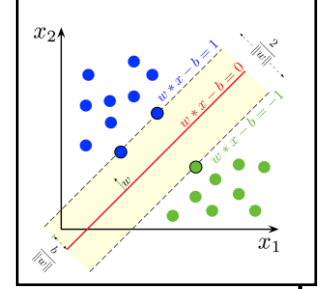
M. Minsky - S. Papert



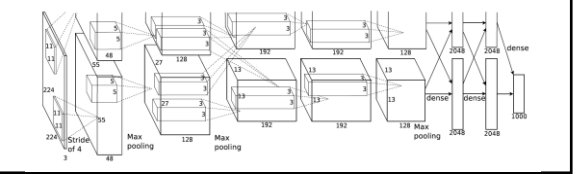
Neocognitron
(Fukushima)
First CNN?
No backprop



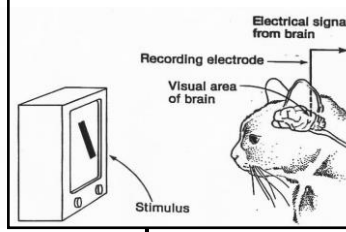
Kernel-SVM
Vapnik&Cortes



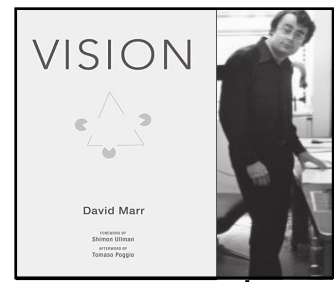
AlexNet- Revolution begins!
Krizhevsky, Sutskever, Hinton



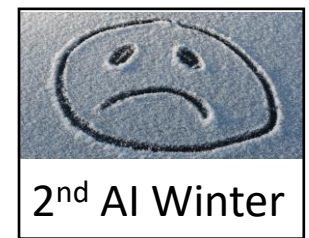
Hubel & Wiesel



1957



1970



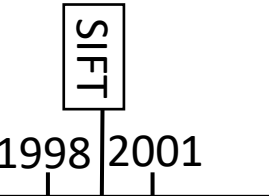
2nd AI Winter

1986

1989

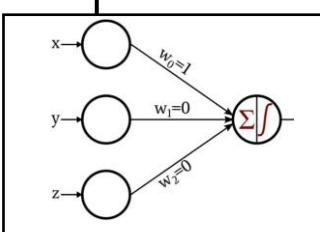


Neural
Winter?

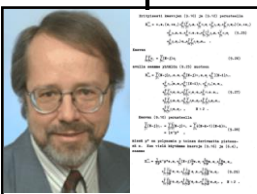


1998 2001

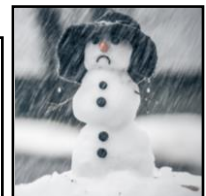
1943 1959



Perceptron
(Rosenblatt)
learned weights



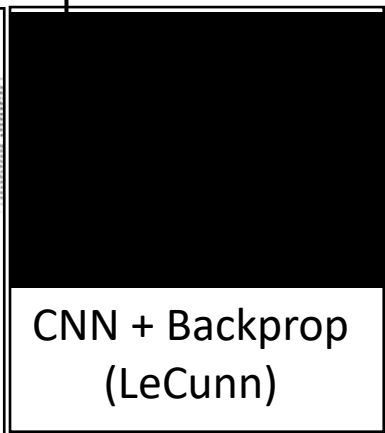
Back
Propagation
(Linnainmaa)



1st
AI Winter

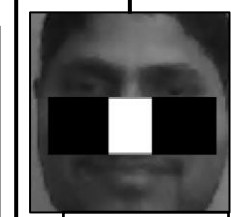


Backprop in
Neural Nets
demonstrated
(Hinton et al.)



CNN + Backprop
(LeCun)

LeNet-5



Viola &
Jones

Rand. Forest

2012



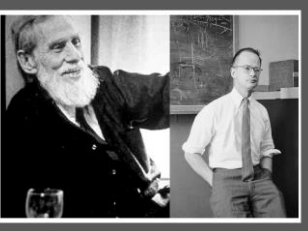
McCulloch Pitts
Non learned
Neuron

XOR kills
perceptron

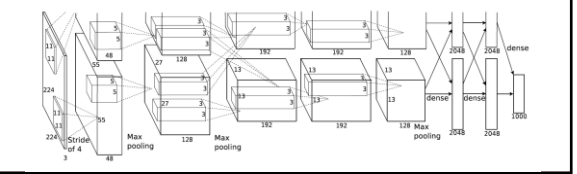
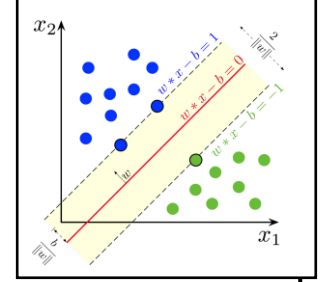
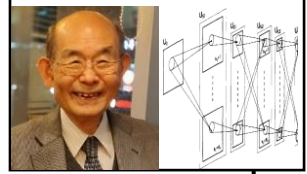
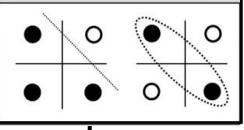
Neocognitron
(Fukushima)
First CNN?
No backprop

Kernel-SVM
Vapnik&Cortes

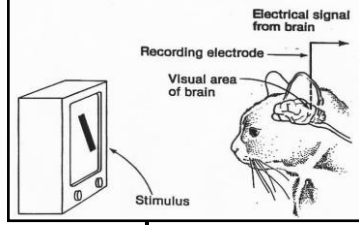
AlexNet- Revolution begins!
Krizhevsky, Sutskever, Hinton



M. Minsky - S. Papert

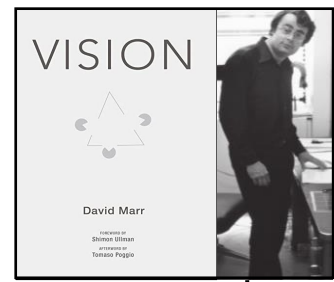


Hubel & Wiesel



1957

1970



David Marr



2nd AI Winter

1986

1989



Neural
Winter?

SIFT

1998

2001

1943 1959

1969

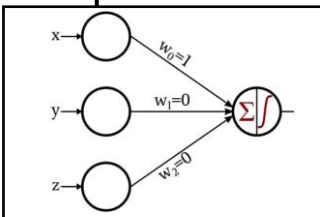
1979

1980

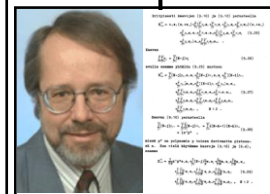
1995

1999

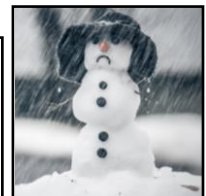
2012



Perceptron
(Rosenblatt)
learned weights



Back
Propagation
(Linnainmaa)



1st
AI Winter

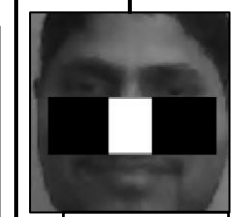


Backprop in
Neural Nets
demonstrated
(Hinton et al.)



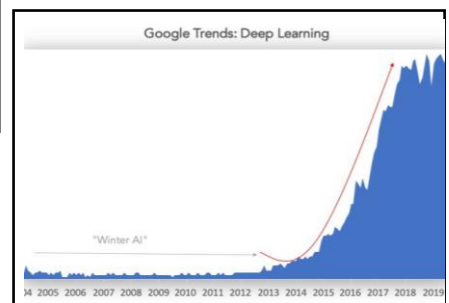
CNN + Backprop
(LeCun)

LeNet-5



Viola &
Jones

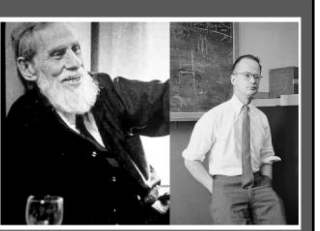
Rand. Forest



Deep Learning
revolution



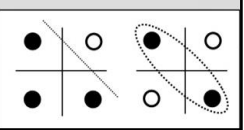
McCulloch Pitts
Non learned
Neuron



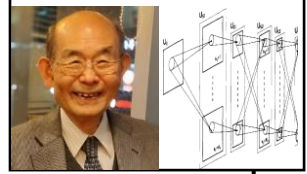
XOR kills
perceptron



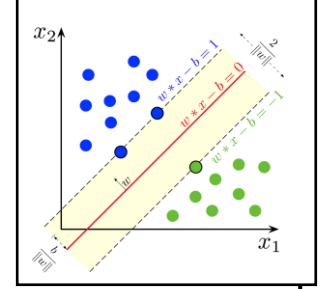
M. Minsky - S. Papert



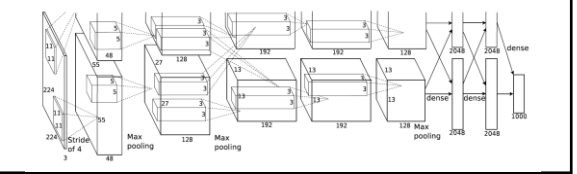
Neocognitron
(Fukushima)
First CNN?
No backprop



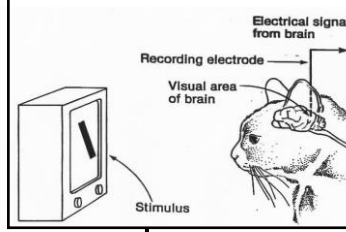
Kernel-SVM
Vapnik&Cortes



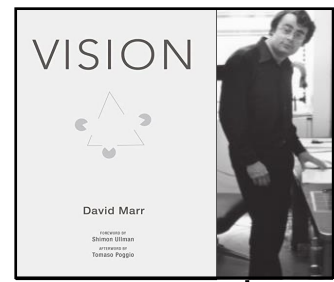
AlexNet- Revolution begins!
Krizhevsky, Sutskever, Hinton



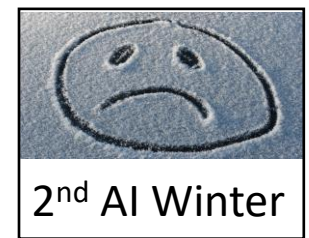
Hubel & Wiesel



1957



1970



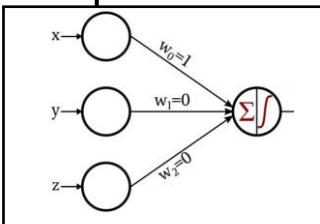
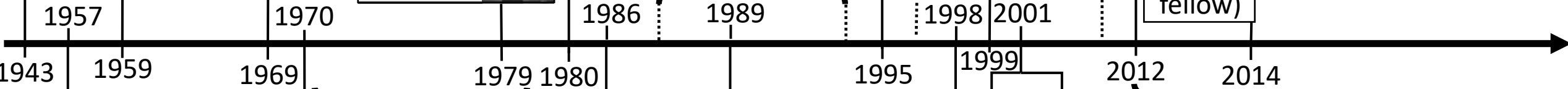
2nd AI Winter



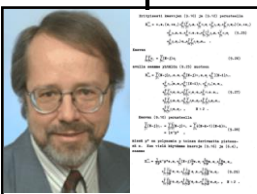
Neural Winter?



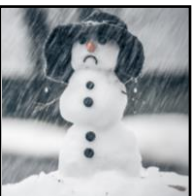
GANs
(Good fellow)



Perceptron
(Rosenblatt)
learned weights



Back
Propagation
(Linnainmaa)



1st
AI Winter



Backprop in
Neural Nets
demonstrated
(Hinton et al.)



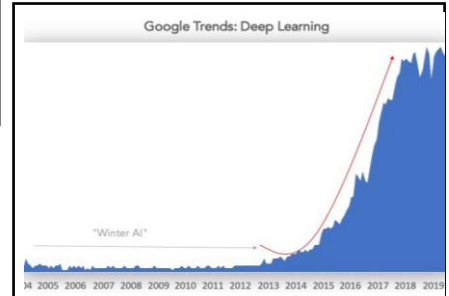
CNN + Backprop
(LeCun)

LeNet-5



Viola &
Jones

Rand. Forest



Deep Learning
revolution

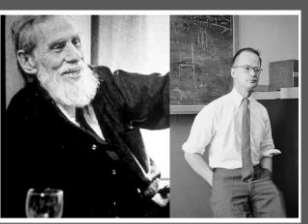
McCulloch Pitts
Non learned
Neuron

XOR kills
perceptron

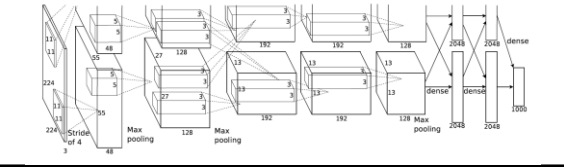
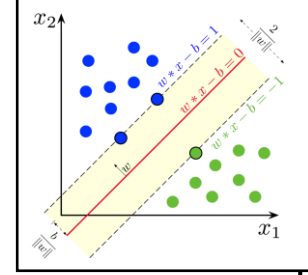
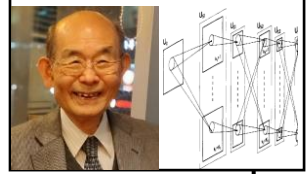
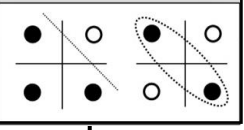
Neocognitron
(Fukushima)
First CNN?
No backprop

Kernel-SVM
Vapnik&Cortes

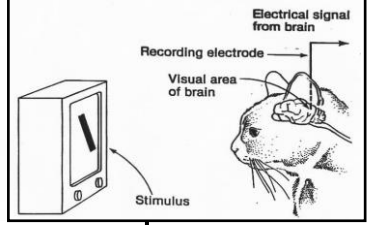
AlexNet- Revolution begins!
Krizhevsky, Sutskever, Hinton



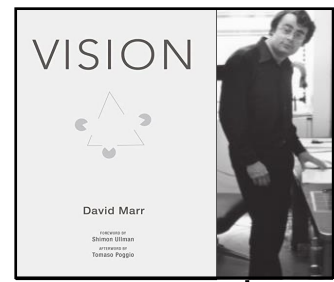
M. Minsky - S. Papert



Hubel & Wiesel



1957



David Marr



2nd AI Winter

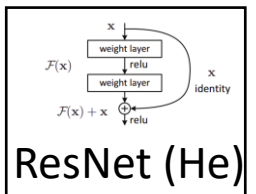


Neural
Winter?

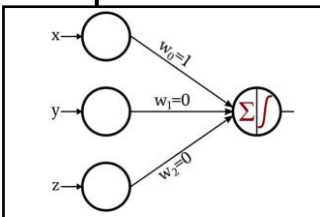
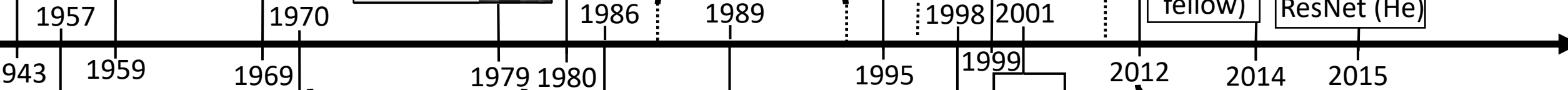
SIFT



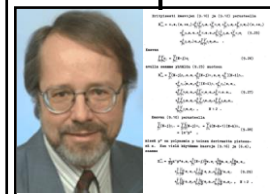
GANs
(Good
fellow)



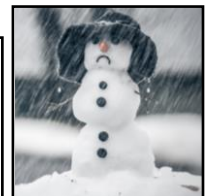
ResNet (He)



Perceptron
(Rosenblatt)
learned weights



Back
Propagation
(Linnainmaa)



1st
AI Winter



Backprop in
Neural Nets
demonstrated
(Hinton et al.)



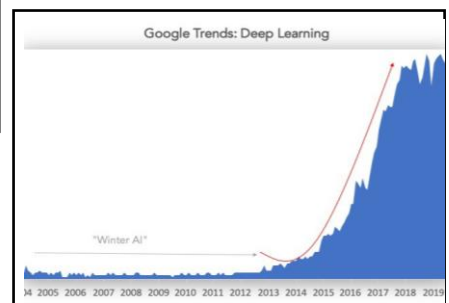
CNN + Backprop
(LeCun)

LeNet-5



Viola &
Jones

Rand. Forest



Deep Learning
revolution



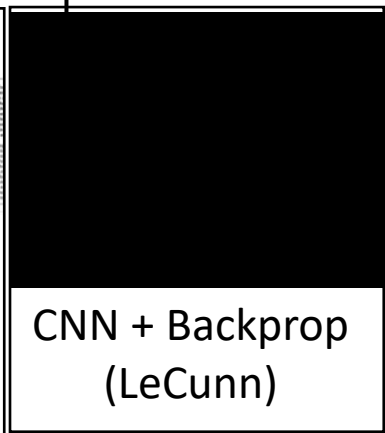
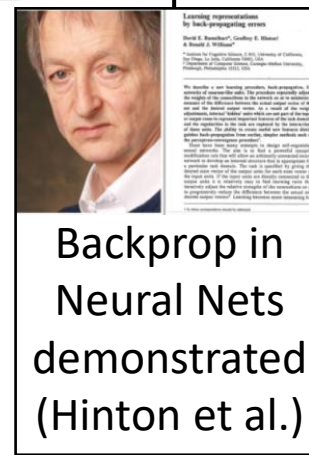
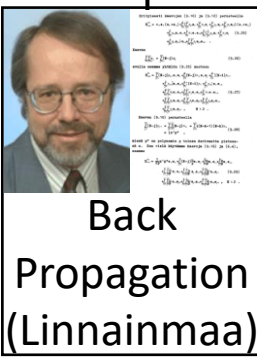
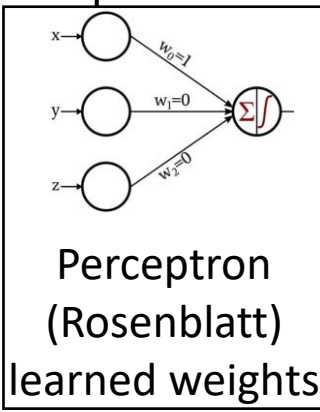
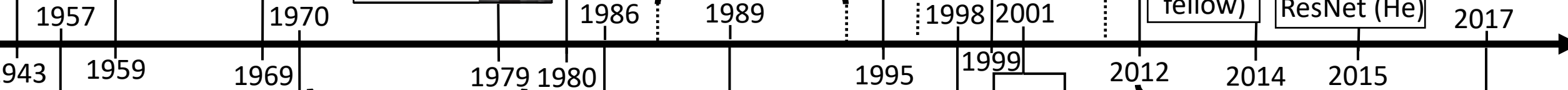
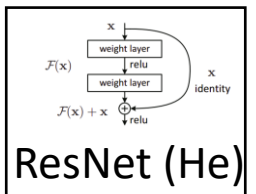
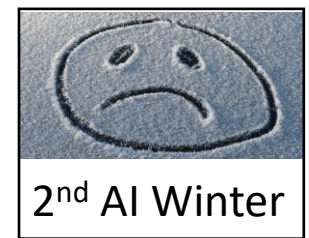
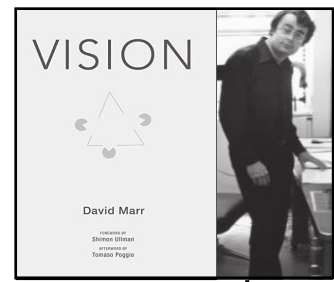
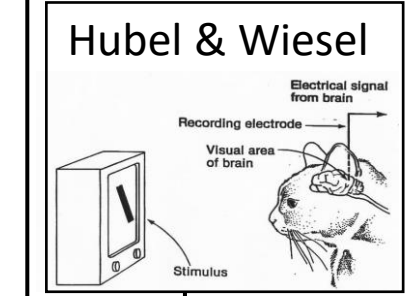
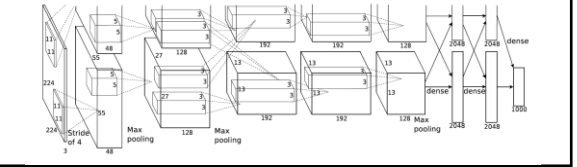
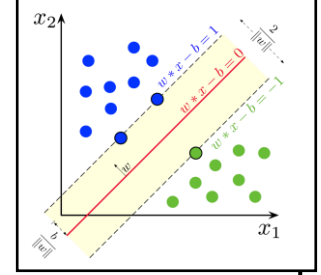
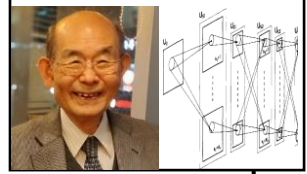
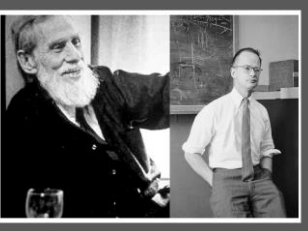
McCulloch Pitts
Non learned
Neuron

XOR kills
perceptron

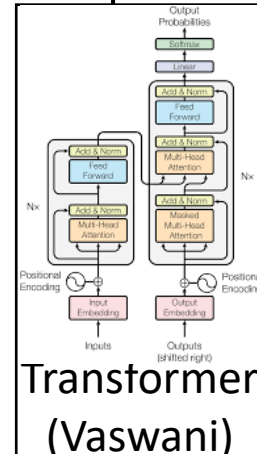
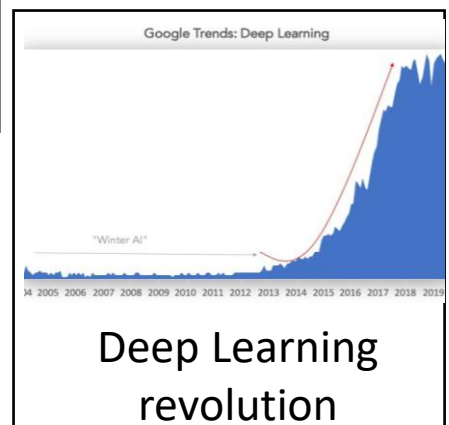
Neocognitron
(Fukushima)
First CNN?
No backprop

Kernel-SVM
Vapnik&Cortes

AlexNet- Revolution begins!
Krizhevsky, Sutskever, Hinton



LeNet-5
Rand. Forest



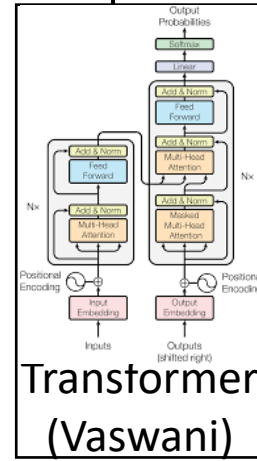
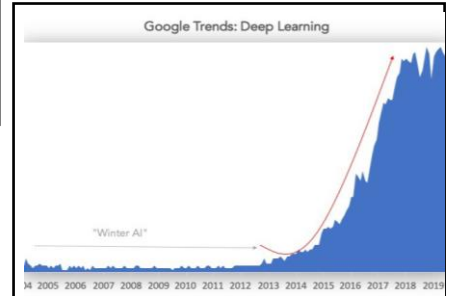
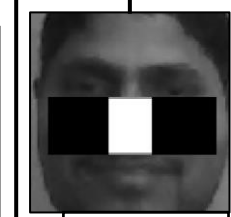
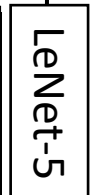
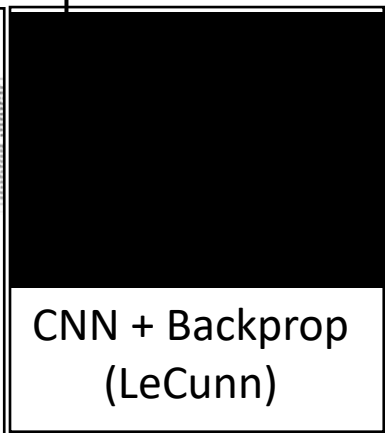
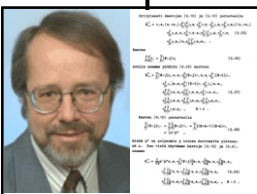
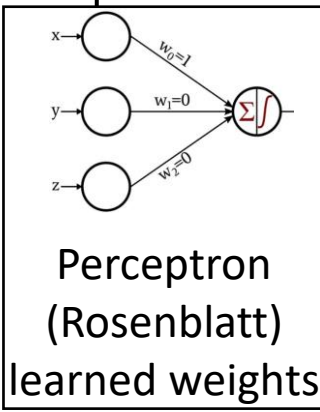
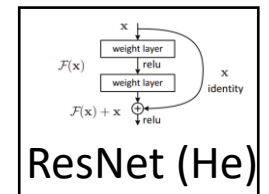
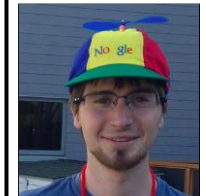
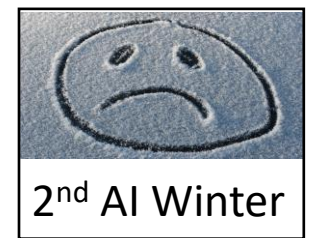
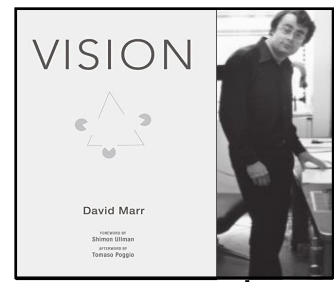
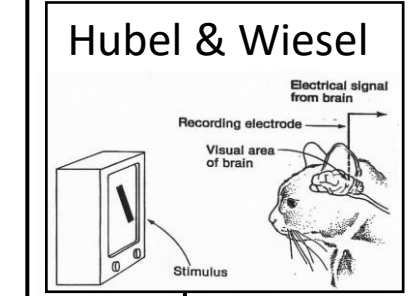
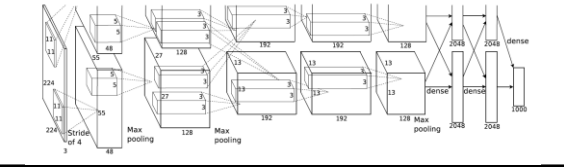
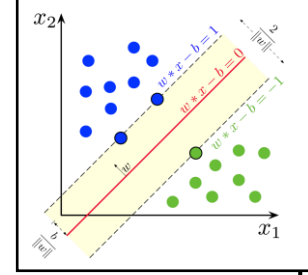
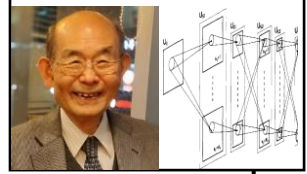
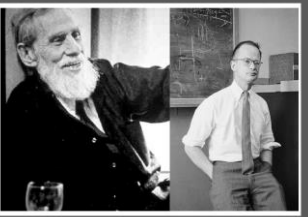
McCulloch Pitts
Non learned
Neuron

XOR kills
perceptron

Neocognitron
(Fukushima)
First CNN?
No backprop

Kernel-SVM
Vapnik&Cortes

AlexNet- Revolution begins!
Krizhevsky, Sutskever, Hinton



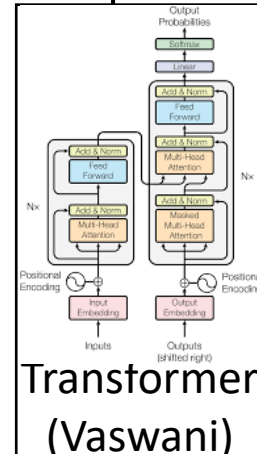
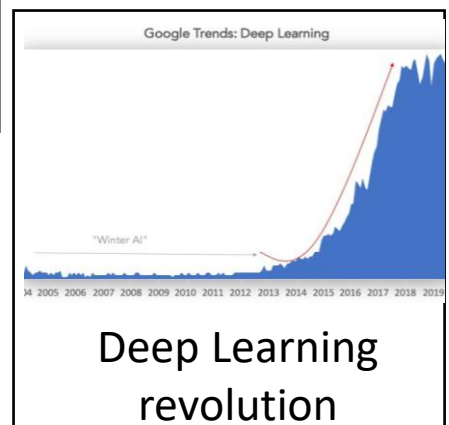
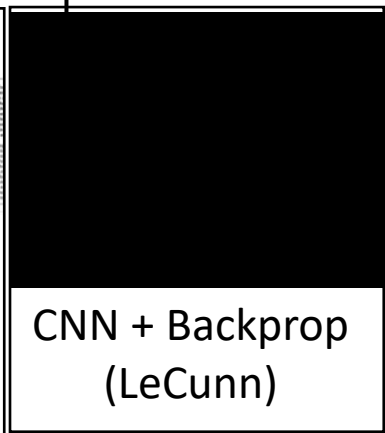
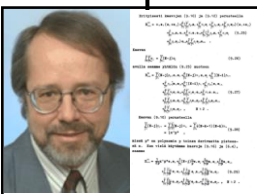
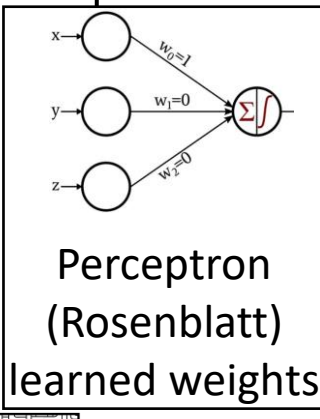
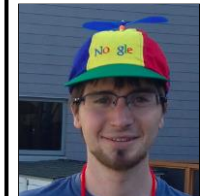
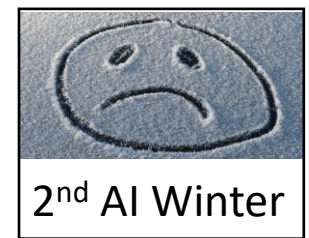
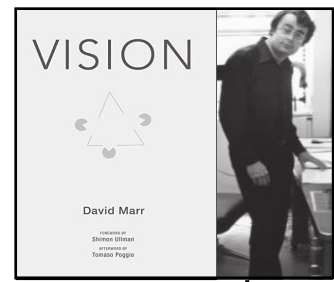
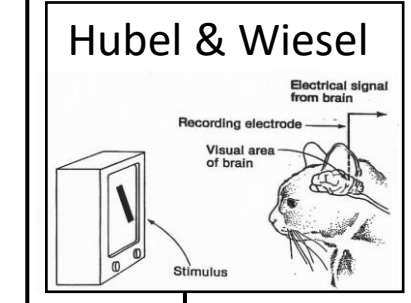
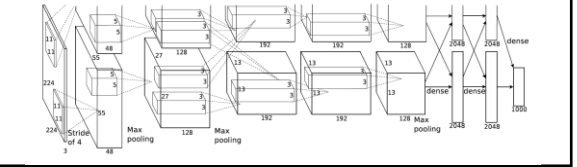
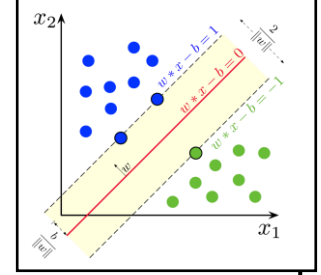
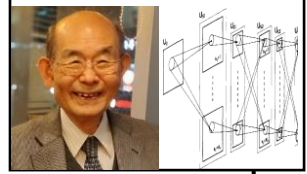
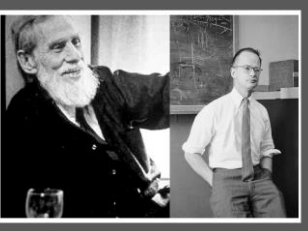
McCulloch Pitts
Non learned
Neuron

XOR kills
perceptron

Neocognitron
(Fukushima)
First CNN?
No backprop

Kernel-SVM
Vapnik&Cortes

AlexNet- Revolution begins!
Krizhevsky, Sutskever, Hinton



Supervised Learning



Supervised Learning

| | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|---|--------------------------|------------------|--------------|---------------|------------|
|  | 0 | 2 | Germany | Gold | Ham |
|  | 1 | 4 | Germany | Silver | Ham |
|  | 5 | 2 | Nigeria | Bronze | Spam |
|  | 2 | 4 | Russia | Bronze | Spam |
|  | 3 | 4 | Germany | Bronze | Ham |
|  | 0 | 1 | USA | Silver | Ham |
|  | 4 | 2 | USA | Silver | Spam |

Numeric Nominal Ordinal



Supervised Learning

Instances

| Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|--------------------------|------------------|--------------|---------------|------------|
| 0 | 2 | Germany | Gold | Ham |
| 1 | 4 | Germany | Silver | Ham |
| 5 | 2 | Nigeria | Bronze | Spam |
| 2 | 4 | Russia | Bronze | Spam |
| 3 | 4 | Germany | Bronze | Ham |
| 0 | 1 | USA | Silver | Ham |
| 4 | 2 | USA | Silver | Spam |

Numeric Nominal Ordinal

Supervised Learning

Features

| Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|--------------------------|------------------|--------------|---------------|------------|
| 0 | 2 | Germany | Gold | Ham |
| 1 | 4 | Germany | Silver | Ham |
| 5 | 2 | Nigeria | Bronze | Spam |
| 2 | 4 | Russia | Bronze | Spam |
| 3 | 4 | Germany | Bronze | Ham |
| 0 | 1 | USA | Silver | Ham |
| 4 | 2 | USA | Silver | Spam |

Numeric Nominal Ordinal



Supervised Learning

Features

Labels

Instances

| | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|--|--------------------------|------------------|--------------|---------------|------------|
| | 0 | 2 | Germany | Gold | Ham |
| | 1 | 4 | Germany | Silver | Ham |
| | 5 | 2 | Nigeria | Bronze | Spam |
| | 2 | 4 | Russia | Bronze | Spam |
| | 3 | 4 | Germany | Bronze | Ham |
| | 0 | 1 | USA | Silver | Ham |
| | 4 | 2 | USA | Silver | Spam |

Numeric Nominal Ordinal

Supervised Learning

Features

Labels

Instances

| | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|--|--------------------------|------------------|--------------|---------------|------------|
| | 0 | 2 | Germany | Gold | Ham |
| | 1 | 4 | Germany | Silver | Ham |
| | 5 | 2 | Nigeria | Bronze | Spam |
| | 2 | 4 | Russia | Bronze | Spam |
| | 3 | 4 | Germany | Bronze | Ham |
| | 0 | 1 | USA | Silver | Ham |
| | 4 | 2 | USA | Silver | Spam |

Numeric Nominal Ordinal

Supervised Learning

Features

Labels








Instances

| | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|--|--------------------------|------------------|--------------|---------------|------------|
| | 0 | 2 | Germany | Gold | Ham |
| | 1 | 4 | Germany | Silver | Ham |
| | 5 | 2 | Nigeria | Bronze | Spam |
| | 2 | 4 | Russia | Bronze | Spam |
| | 3 | 4 | Germany | Bronze | Ham |
| | 0 | 1 | USA | Silver | Ham |
| | 4 | 2 | USA | Silver | Spam |

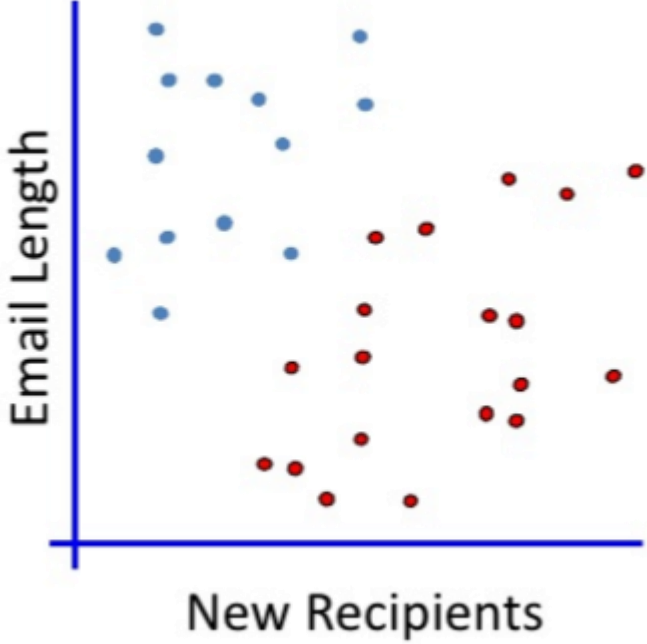
Numeric Nominal Ordinal

Supervised Learning

Features Labels

| | | | | | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|-----------|---|---|---|---------|--------------------------|------------------|--------------|---------------|------------|
| Instances |  | 0 | 2 | Germany | Gold | Ham | | | |
| |  | 1 | 4 | Germany | Silver | Ham | | | |
| |  | 5 | 2 | Nigeria | Bronze | Spam | | | |
| |  | 2 | 4 | Russia | Bronze | Spam | | | |
| |  | 3 | 4 | Germany | Bronze | Ham | | | |
| |  | 0 | 1 | USA | Silver | Ham | | | |
| |  | 4 | 2 | USA | Silver | Spam | | | |

Numeric Nominal Ordinal



Supervised Learning

Features

Labels

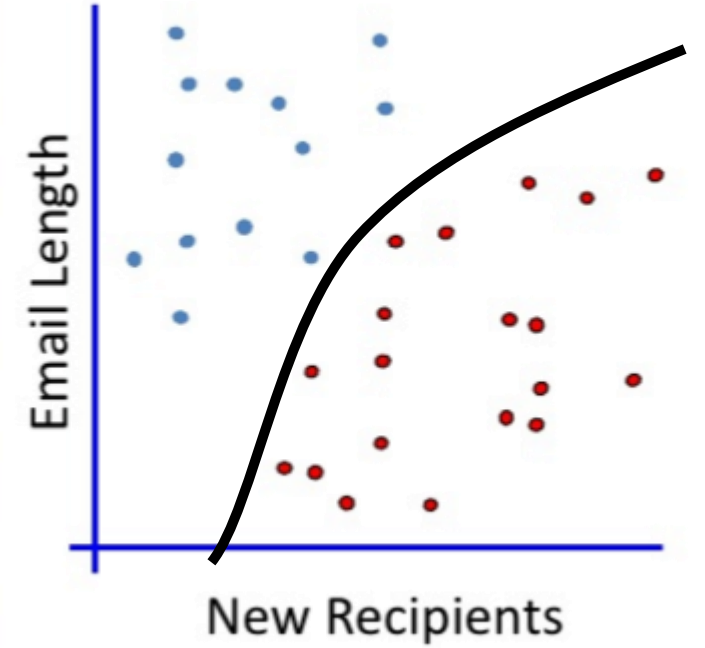
Instances

| | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|--|--------------------------|------------------|--------------|---------------|------------|
| | 0 | 2 | Germany | Gold | Ham |
| | 1 | 4 | Germany | Silver | Ham |
| | 5 | 2 | Nigeria | Bronze | Spam |
| | 2 | 4 | Russia | Bronze | Spam |
| | 3 | 4 | Germany | Bronze | Ham |
| | 0 | 1 | USA | Silver | Ham |
| | 4 | 2 | USA | Silver | Spam |

Numeric

Nominal

Ordinal



Supervised Learning

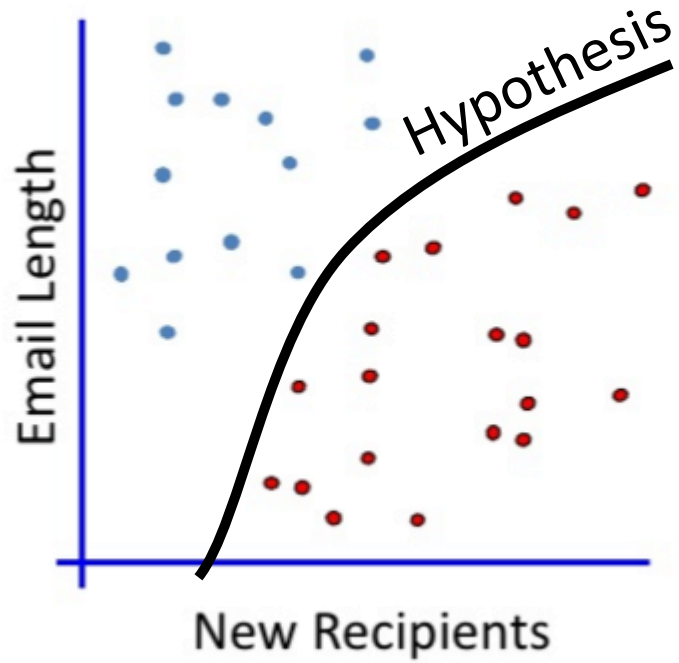
Features

Labels

Instances








| | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|---|--------------------------|------------------|--------------|---------------|------------|
| Ⓜ | 0 | 2 | Germany | Gold | Ham |
| Ⓜ | 1 | 4 | Germany | Silver | Ham |
| Ⓜ | 5 | 2 | Nigeria | Bronze | Spam |
| Ⓜ | 2 | 4 | Russia | Bronze | Spam |
| Ⓜ | 3 | 4 | Germany | Bronze | Ham |
| Ⓜ | 0 | 1 | USA | Silver | Ham |
| Ⓜ | 4 | 2 | USA | Silver | Spam |

Numeric Nominal Ordinal

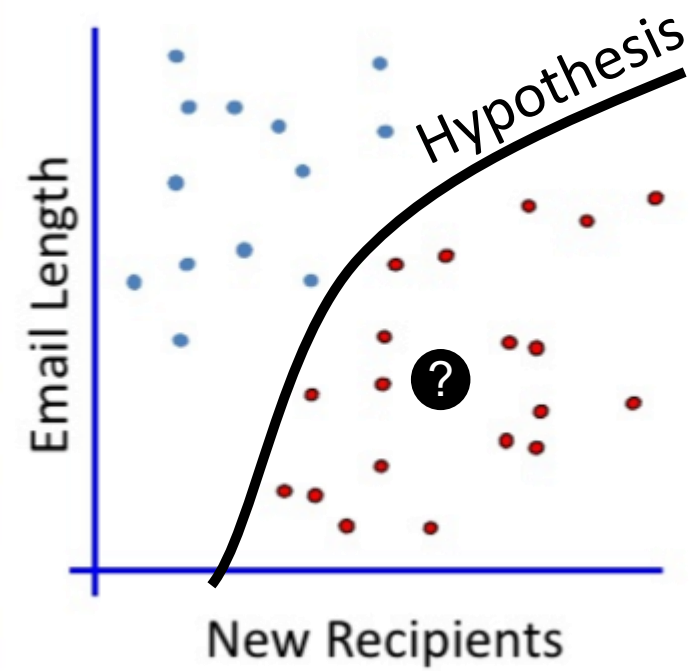


Supervised Learning

Features Labels

| | | | | | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|-----------|---|---|---|---------|--------------------------|------------------|--------------|---------------|------------|
| Instances |  | 0 | 2 | Germany | Gold | Ham | | | |
| |  | 1 | 4 | Germany | Silver | Ham | | | |
| |  | 5 | 2 | Nigeria | Bronze | Spam | | | |
| |  | 2 | 4 | Russia | Bronze | Spam | | | |
| |  | 3 | 4 | Germany | Bronze | Ham | | | |
| |  | 0 | 1 | USA | Silver | Ham | | | |
| |  | 4 | 2 | USA | Silver | Spam | | | |

Numeric Nominal Ordinal



Supervised Learning

Features

Labels

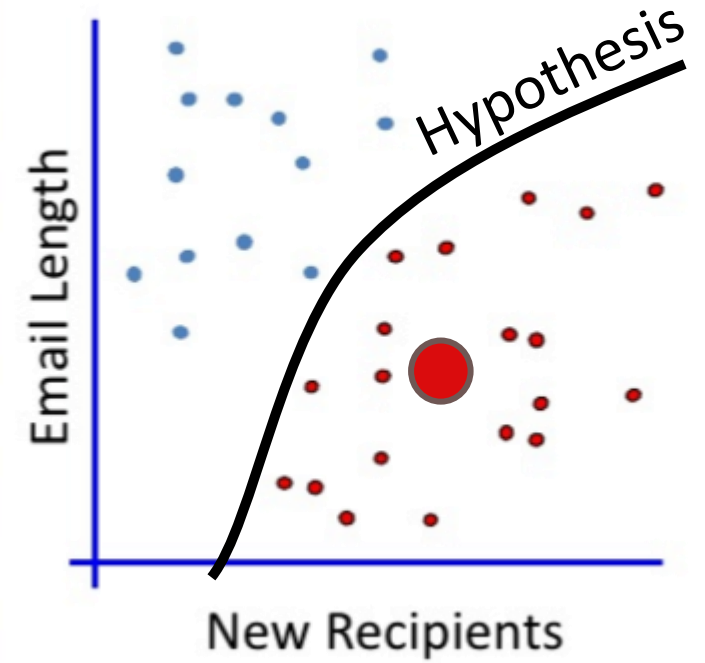
Instances

| | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|--|--------------------------|------------------|--------------|---------------|------------|
| | 0 | 2 | Germany | Gold | Ham |
| | 1 | 4 | Germany | Silver | Ham |
| | 5 | 2 | Nigeria | Bronze | Spam |
| | 2 | 4 | Russia | Bronze | Spam |
| | 3 | 4 | Germany | Bronze | Ham |
| | 0 | 1 | USA | Silver | Ham |
| | 4 | 2 | USA | Silver | Spam |

Numeric

Nominal

Ordinal

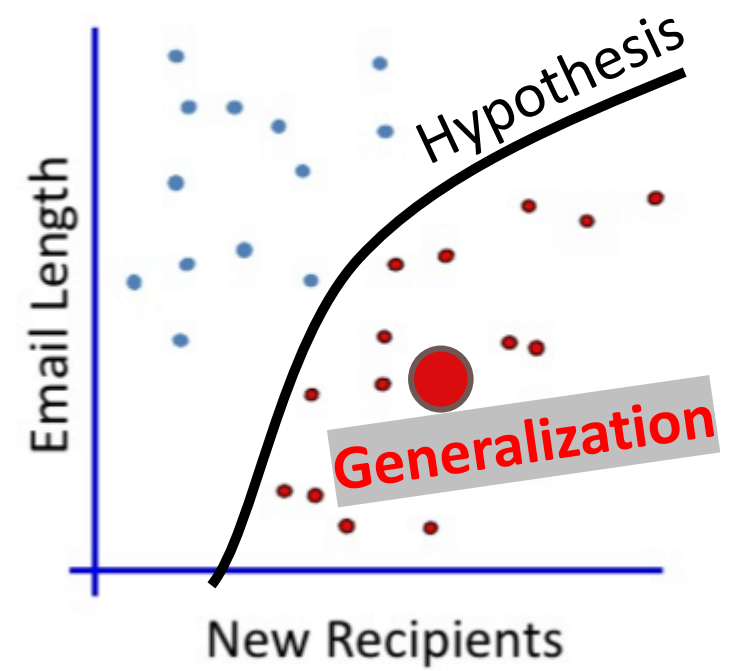


Supervised Learning

Features Labels

| | | | | | Number of new Recipients | Email Length (K) | Country (IP) | Customer Type | Email Type |
|-----------|--|---|---|---------|--------------------------|------------------|--------------|---------------|------------|
| Instances | | 0 | 2 | Germany | Gold | Ham | | | |
| | | 1 | 4 | Germany | Silver | Ham | | | |
| | | 5 | 2 | Nigeria | Bronze | Spam | | | |
| | | 2 | 4 | Russia | Bronze | Spam | | | |
| | | 3 | 4 | Germany | Bronze | Ham | | | |
| | | 0 | 1 | USA | Silver | Ham | | | |
| | | 4 | 2 | USA | Silver | Spam | | | |

Numeric Nominal Ordinal



Supervised Learning



Supervised Learning

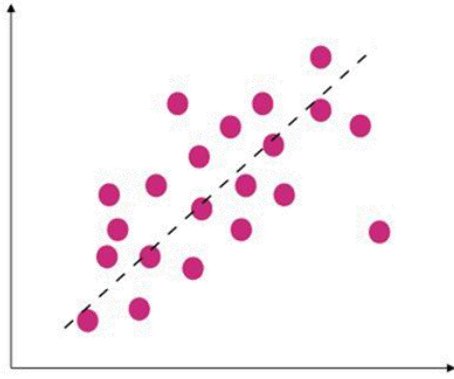
Regression

Classification



Supervised Learning

Regression

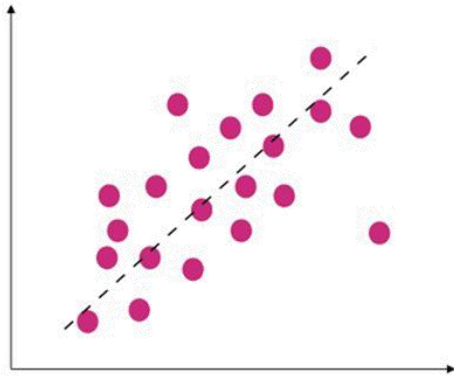


Classification

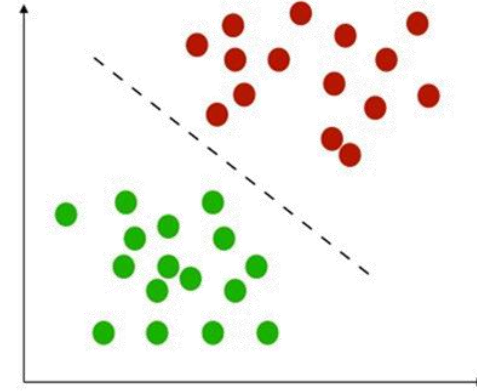


Supervised Learning

Regression

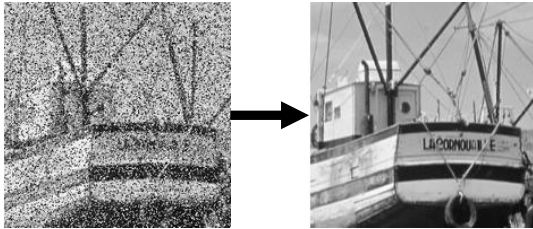
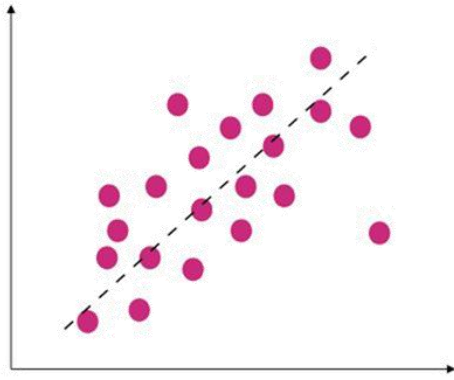


Classification



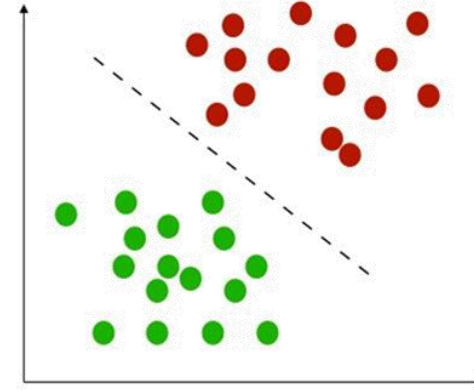
Supervised Learning

Regression



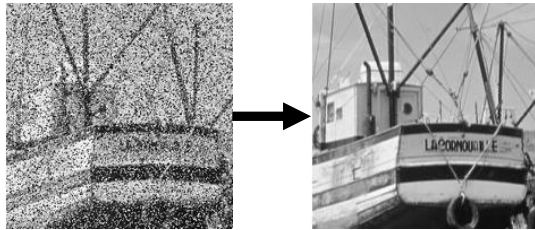
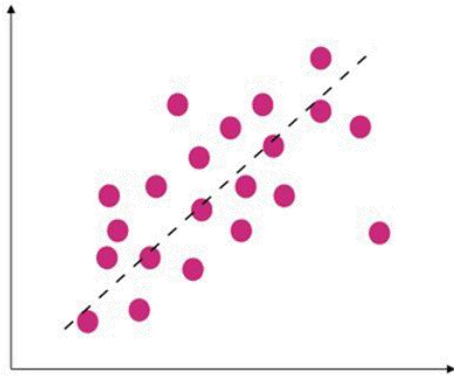
E.g. Image denoising

Classification



Supervised Learning

Regression

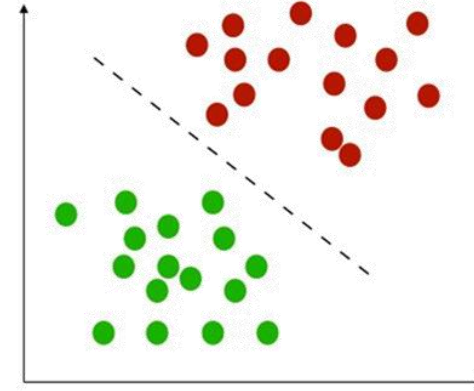


E.g. Image denoising



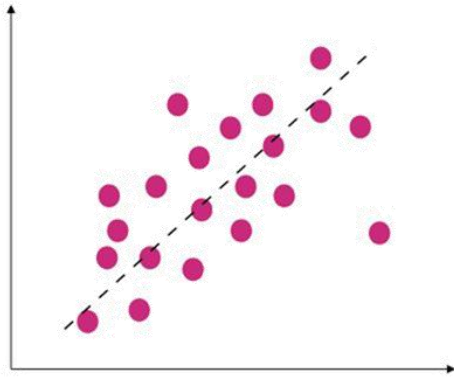
E.g. Object localization

Classification

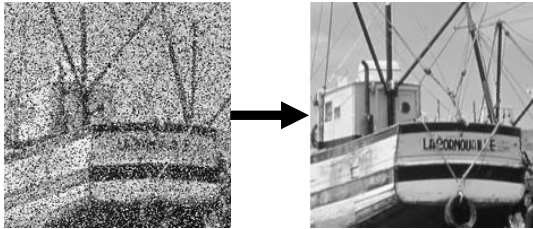


Supervised Learning

Regression



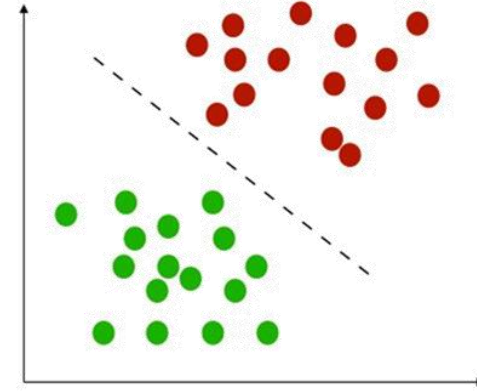
E.g. Image denoising



E.g. Object localization



Classification

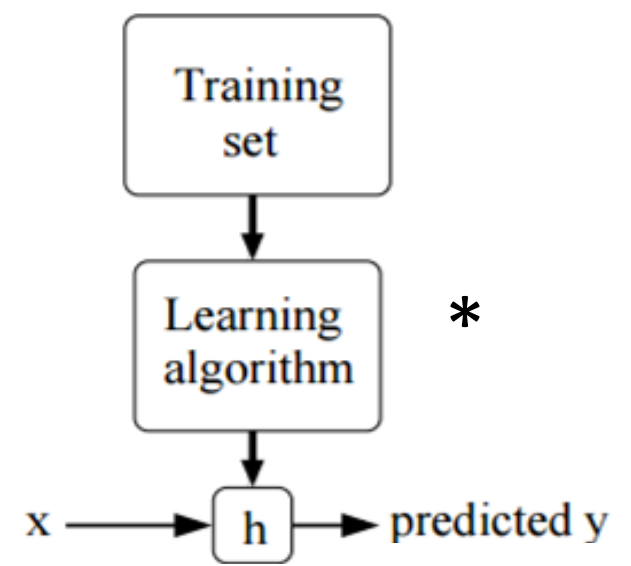


E.g. Image classification



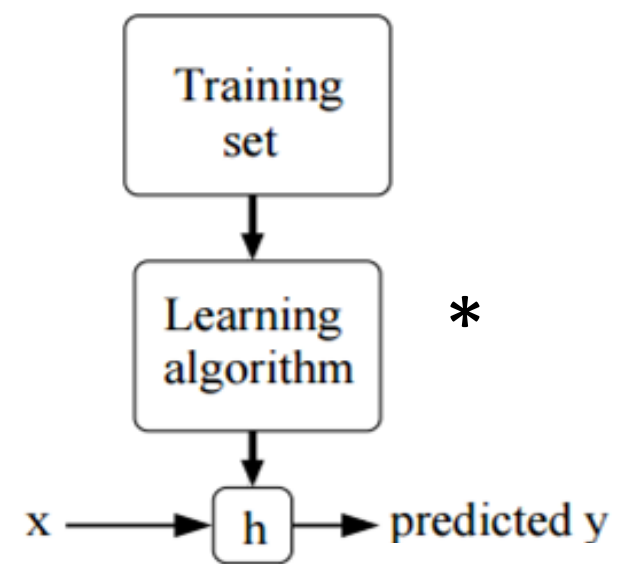
Supervised Learning

$$A(S) = h$$



Supervised Learning

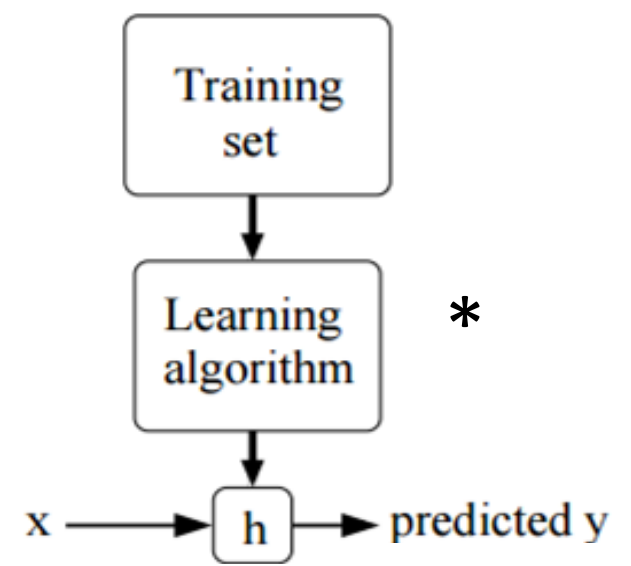
$$A(\overset{\text{Training set}}{\boxed{S}}) = h$$



Supervised Learning

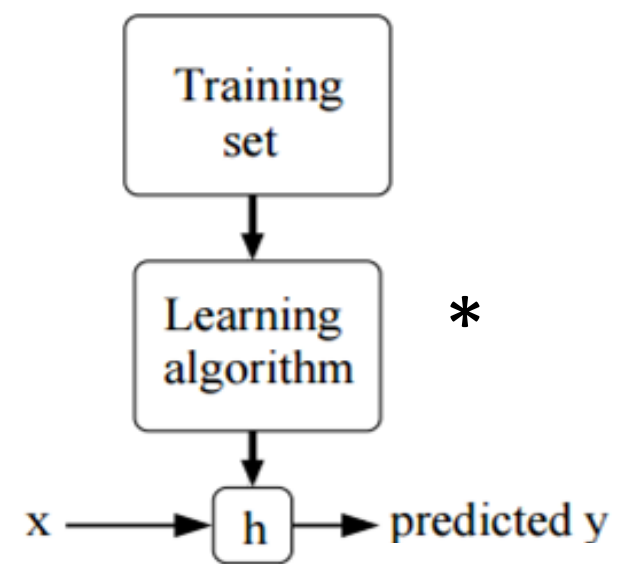
Learning Algorithm A Training set S

$$A(S) = h$$



Supervised Learning

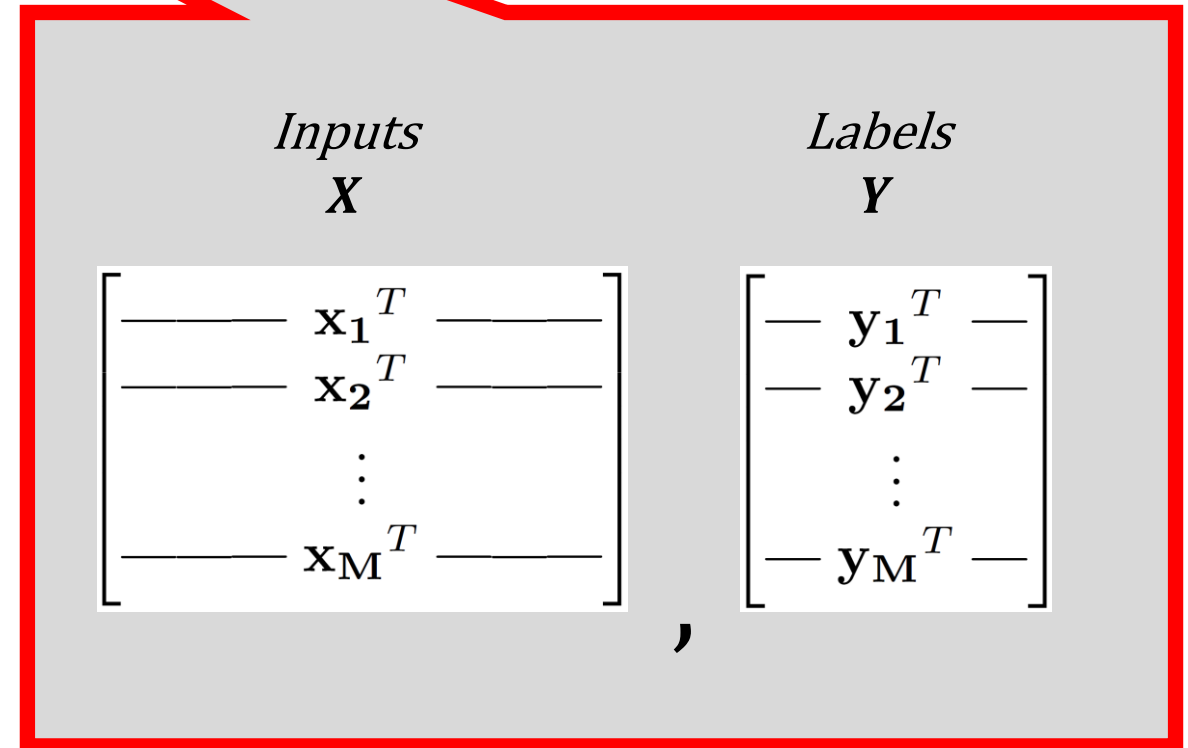
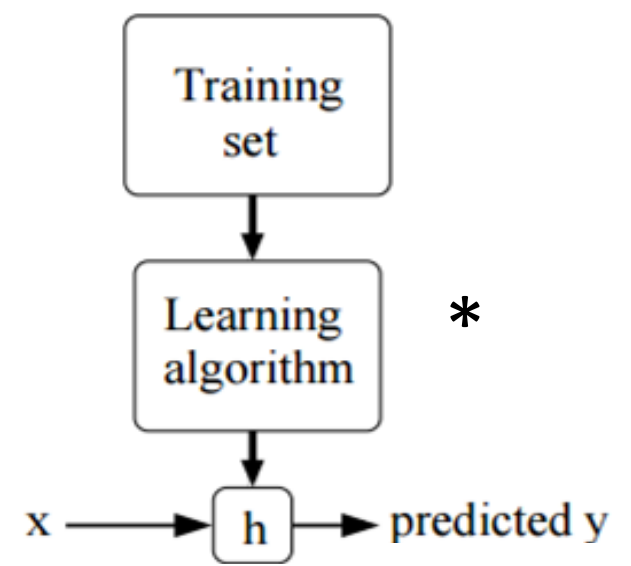
Learning Algorithm A (Training set S) = Hypothesis h



Supervised Learning

Learning Algorithm A Training set S Hypothesis h

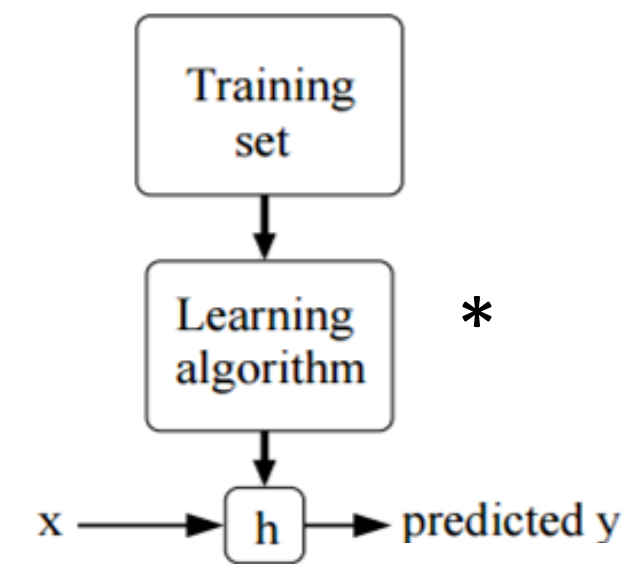
$$A(S) = h$$



Supervised Learning

Learning Algorithm A (Training set S) = Hypothesis h

Hypothesis class
 $\mathcal{H} = \{h_1, h_2 \dots\}$



Inputs
 X

| | | |
|---|------------------|---|
| — | \mathbf{x}_1^T | — |
| — | \mathbf{x}_2^T | — |
| | \vdots | |
| — | \mathbf{x}_M^T | — |

,

Labels
 Y

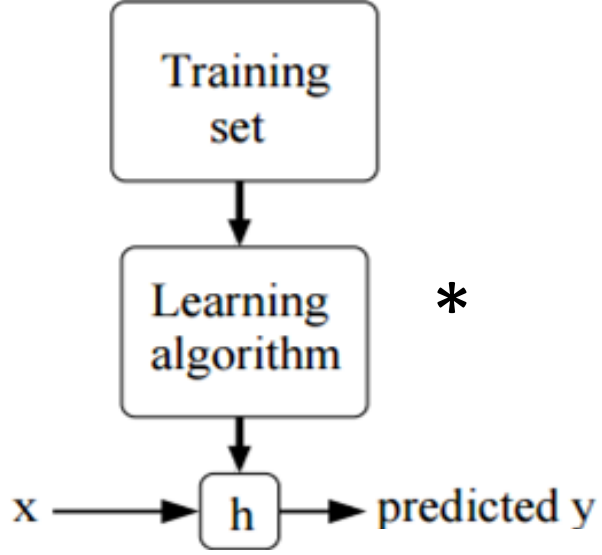
| | | |
|---|------------------|---|
| — | \mathbf{y}_1^T | — |
| — | \mathbf{y}_2^T | — |
| | \vdots | |
| — | \mathbf{y}_M^T | — |



Supervised Learning

Learning Algorithm A (Training set S) = Hypothesis h

Hypothesis class
 $\mathcal{H} = \{h_1, h_2 \dots\}$



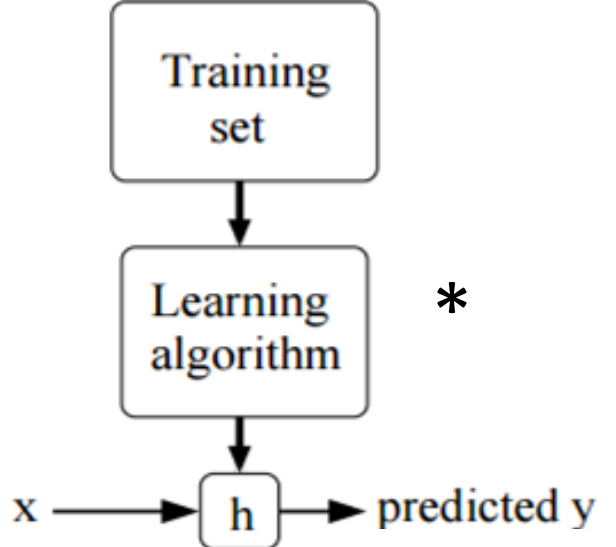
| Inputs X | Labels Y |
|--|--|
| $\begin{bmatrix} \text{---} & \mathbf{x}_1^T & \text{---} \\ \text{---} & \mathbf{x}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{x}_M^T & \text{---} \end{bmatrix}$ | $\begin{bmatrix} \text{---} & \mathbf{y}_1^T & \text{---} \\ \text{---} & \mathbf{y}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{y}_M^T & \text{---} \end{bmatrix}$ |

,



Supervised Learning

Learning Algorithm A (Training set S) = Hypothesis h



Hypothesis class
 $\mathcal{H} = \{h_1, h_2 \dots\}$

$h(x) \approx y$

| Inputs X | Labels Y |
|----------------------|---------------|
| — \mathbf{x}_1^T — | — y_1^T — |
| — \mathbf{x}_2^T — | — y_2^T — |
| ⋮ | ⋮ |
| — \mathbf{x}_M^T — | — y_M^T — |

,



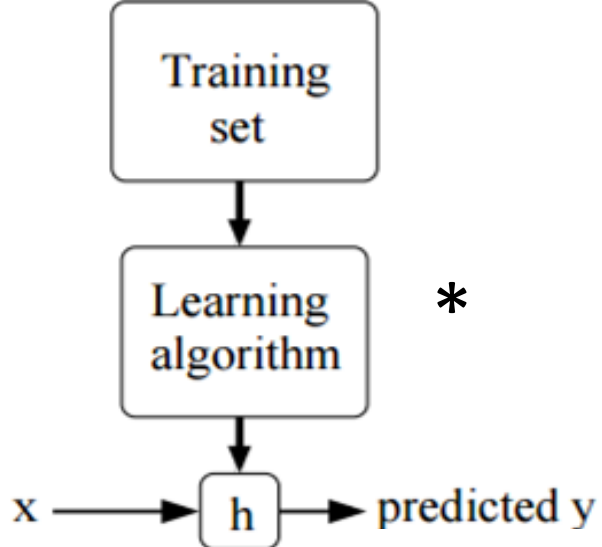
Supervised Learning

Learning Algorithm A (Training set S) = Hypothesis h

Hypothesis class
 $\mathcal{H} = \{h_1, h_2 \dots\}$

Loss
 \mathcal{L}

$h(x) \approx y$



| <i>Inputs</i> X | <i>Labels</i> Y |
|--|--|
| $\begin{bmatrix} \text{---} & \mathbf{x}_1^T & \text{---} \\ \text{---} & \mathbf{x}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{x}_M^T & \text{---} \end{bmatrix}$ | $\begin{bmatrix} \text{---} & \mathbf{y}_1^T & \text{---} \\ \text{---} & \mathbf{y}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{y}_M^T & \text{---} \end{bmatrix}$ |
| , | |



Supervised Learning

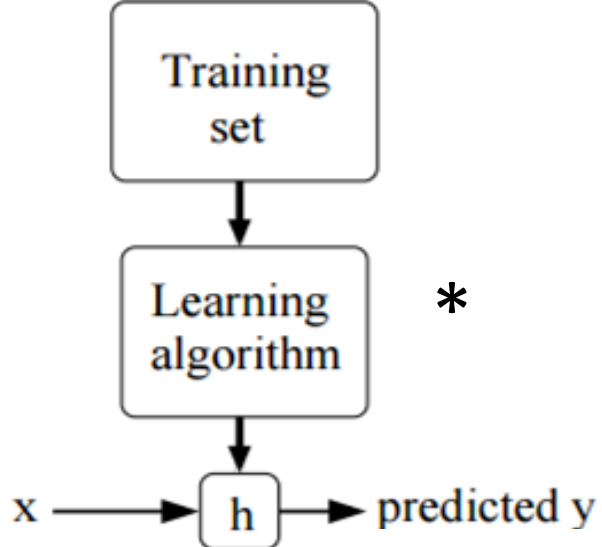
Learning Algorithm A (Training set S) = Hypothesis h

Hypothesis class
 $\mathcal{H} = \{h_1, h_2 \dots\}$

Loss
 \mathcal{L}

Optimization method

$h(x) \approx y$



| <i>Inputs</i> X | <i>Labels</i> Y |
|--|--|
| $\begin{bmatrix} \text{---} & \mathbf{x}_1^T & \text{---} \\ \text{---} & \mathbf{x}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{x}_M^T & \text{---} \end{bmatrix}$ | $\begin{bmatrix} \text{---} & \mathbf{y}_1^T & \text{---} \\ \text{---} & \mathbf{y}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{y}_M^T & \text{---} \end{bmatrix}$ |
| , | |



Supervised Learning

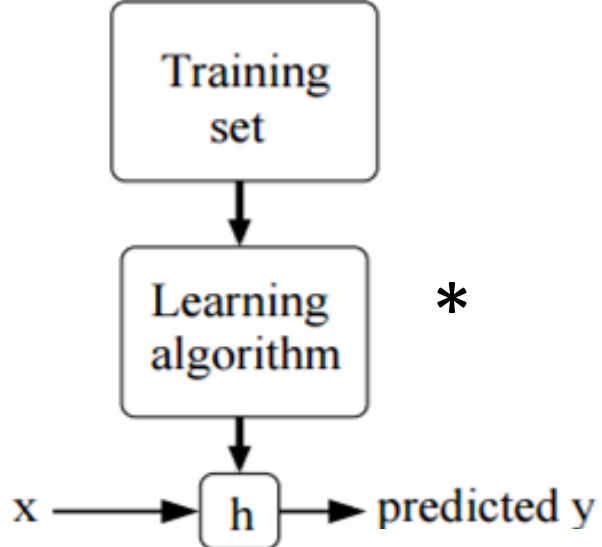
Learning Algorithm A (Training set S) = Hypothesis h

Hypothesis class
 $\mathcal{H} = \{h_1, h_2 \dots\}$

Loss
 \mathcal{L}

Optimization method

$h(x) \approx y$



| <i>Inputs</i> X | <i>Labels</i> Y |
|--|--|
| $\begin{bmatrix} \text{---} & \mathbf{x}_1^T & \text{---} \\ \text{---} & \mathbf{x}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{x}_M^T & \text{---} \end{bmatrix}$ | $\begin{bmatrix} \text{---} & \mathbf{y}_1^T & \text{---} \\ \text{---} & \mathbf{y}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{y}_M^T & \text{---} \end{bmatrix}$ |
| , | |



Supervised Learning

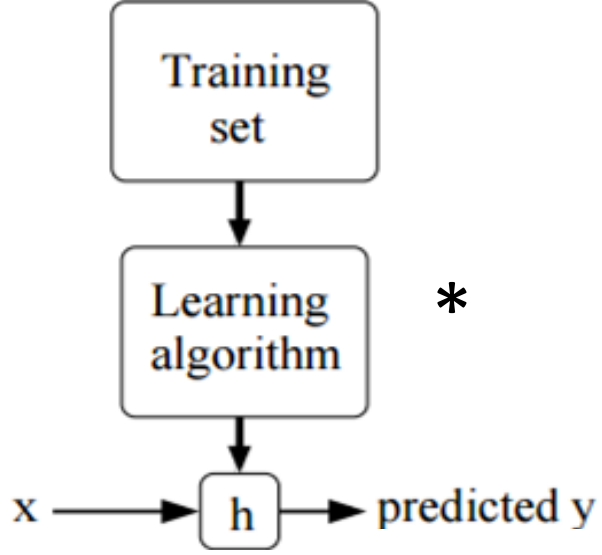
Learning Algorithm A (Training set S) = Hypothesis h

Hypothesis class
 $\mathcal{H} = \{h_1, h_2 \dots\}$

Loss
 \mathcal{L}

Optimization method

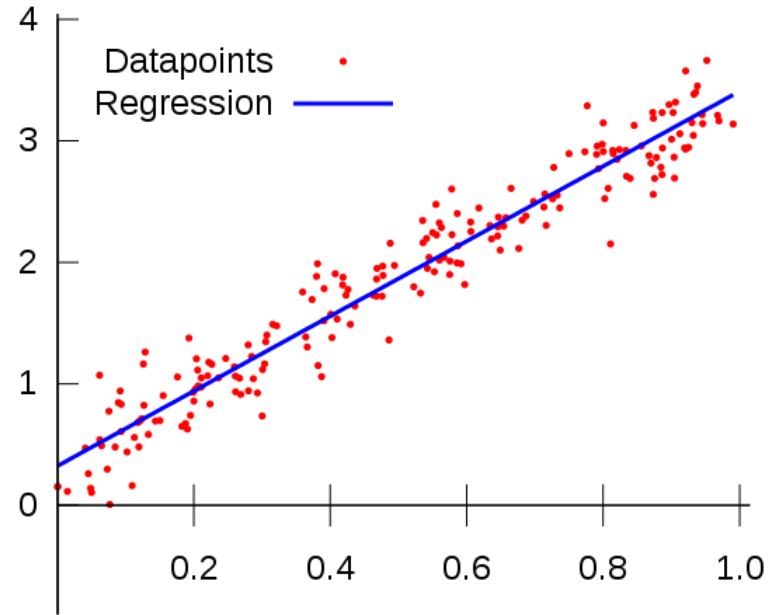
$h(x) \approx y$



| <i>Inputs</i> X | <i>Labels</i> Y |
|--|--|
| $\begin{bmatrix} \text{---} & \mathbf{x}_1^T & \text{---} \\ \text{---} & \mathbf{x}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{x}_M^T & \text{---} \end{bmatrix}$ | $\begin{bmatrix} \text{---} & \mathbf{y}_1^T & \text{---} \\ \text{---} & \mathbf{y}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{y}_M^T & \text{---} \end{bmatrix}$ |
| , | |



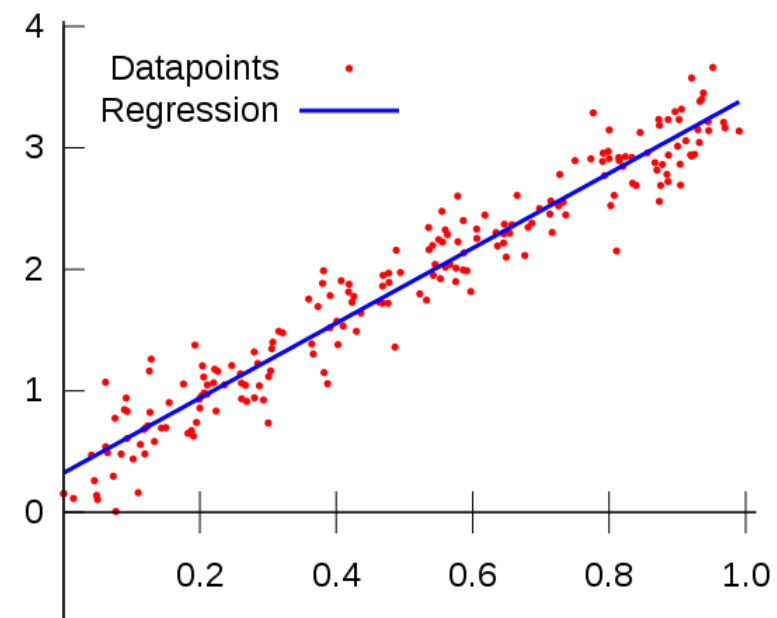
Linear Regression



Linear Regression

Hypothesis class: Linear \rightarrow

$$\mathcal{H} = \{h_{\theta} \mid \theta \in \mathbb{R}^{N+1}\}, \quad h_{\theta}(\mathbf{x}) = \theta_0 + \sum_{j=1}^N \theta_j x_j = \theta_0 + \tilde{\theta}^T \mathbf{x}$$

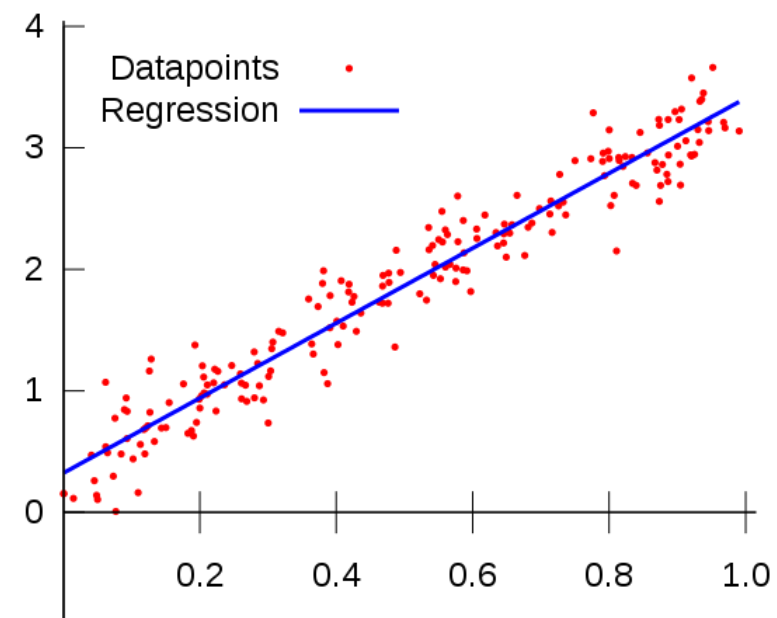


Linear Regression

Hypothesis class: Linear \rightarrow

$$\mathcal{H} = \{h_{\theta} \mid \theta \in \mathbb{R}^{N+1}\}, \quad h_{\theta}(\mathbf{x}) = \theta_0 + \sum_{j=1}^N \theta_j x_j = \theta_0 + \tilde{\theta}^T \mathbf{x} = \boldsymbol{\theta}^T \begin{pmatrix} 1 \\ | \\ \mathbf{x} \\ | \end{pmatrix}$$

Bias = Just add 1 at top of the input vec!



Linear Regression

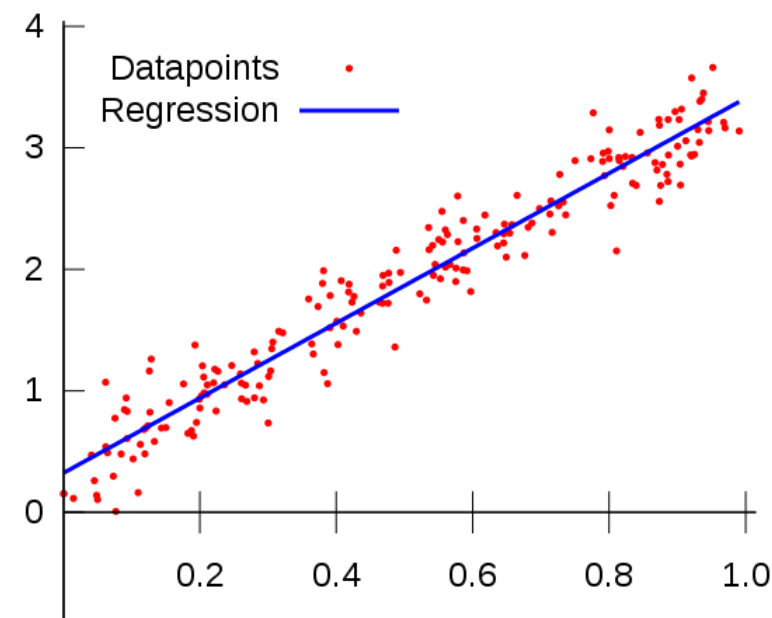
Hypothesis class: Linear \rightarrow

$$\mathcal{H} = \{h_{\theta} \mid \theta \in \mathbb{R}^{N+1}\}, \quad h_{\theta}(\mathbf{x}) = \theta_0 + \sum_{j=1}^N \theta_j x_j = \theta_0 + \tilde{\theta}^T \mathbf{x} = \boldsymbol{\theta}^T \begin{pmatrix} 1 \\ | \\ \mathbf{x} \\ | \end{pmatrix}$$

Bias = Just add 1 at top of the input vec!

Loss: Mean Squared Error \rightarrow

$$\mathcal{L} = \frac{1}{2M} \sum_{i=1}^M (h_{\theta}(\mathbf{x}_i) - y_i)^2 = \frac{1}{2M} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|^2$$



Linear Regression

Hypothesis class: Linear \rightarrow

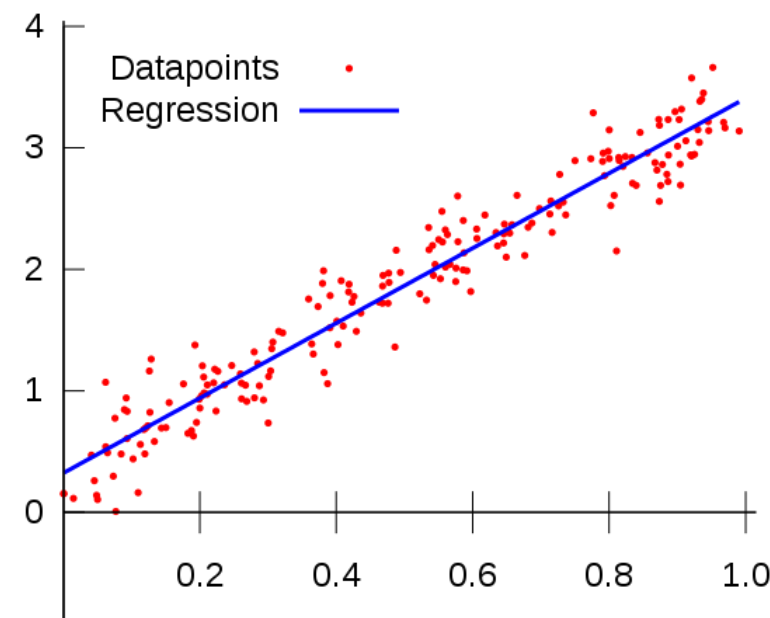
$$\mathcal{H} = \{h_{\theta} \mid \theta \in \mathbb{R}^{N+1}\}, \quad h_{\theta}(\mathbf{x}) = \theta_0 + \sum_{j=1}^N \theta_j x_j = \theta_0 + \tilde{\theta}^T \mathbf{x} = \theta^T \begin{pmatrix} 1 \\ | \\ \mathbf{x} \\ | \end{pmatrix}$$

Bias = Just add 1 at top of the input vec!

Loss: Mean Squared Error \rightarrow

$$\mathcal{L} = \frac{1}{2M} \sum_{i=1}^M (h_{\theta}(\mathbf{x}_i) - y_i)^2 = \frac{1}{2M} \|\mathbf{X}\theta - \mathbf{y}\|^2$$

Optimization method: Normal equations / Gradient Descent

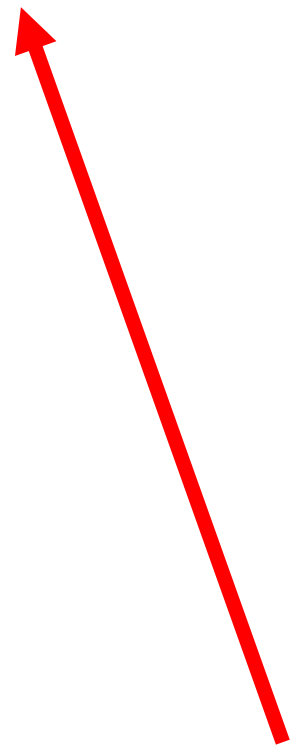


Normal Equations (intuition)

$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - \mathbf{X}\theta\|^2$$

Normal Equations (intuition)

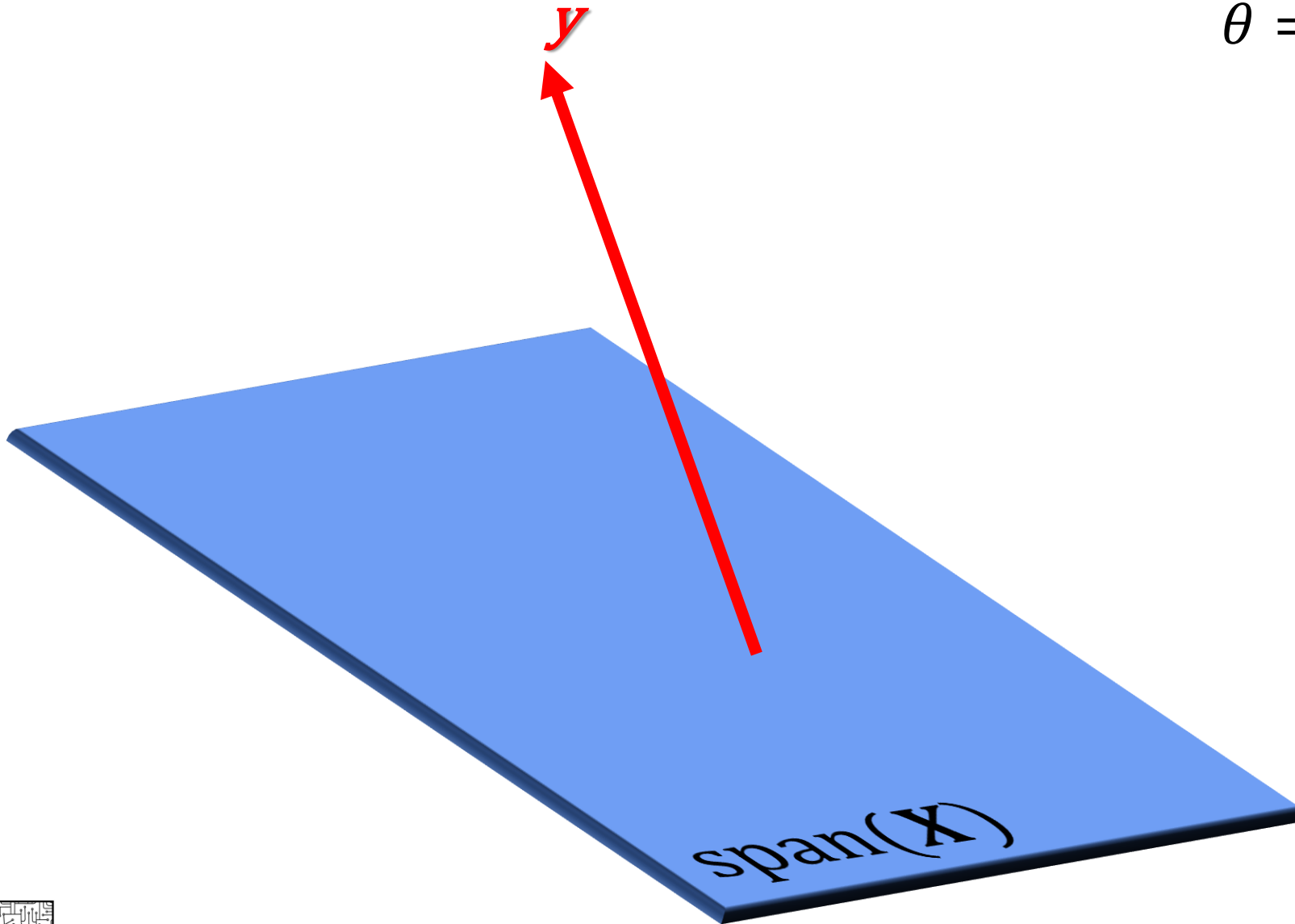
y



$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$

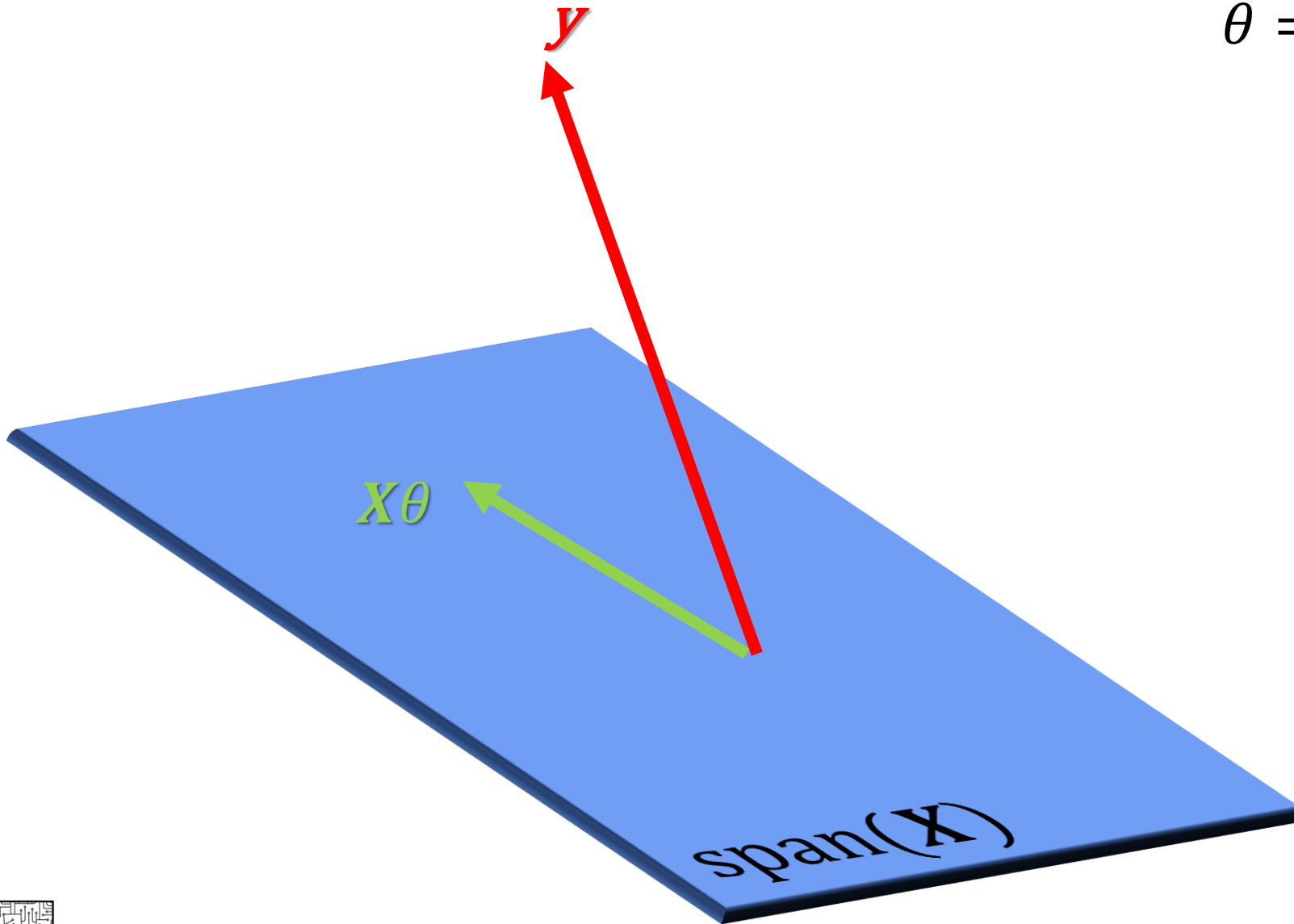
Normal Equations (intuition)

$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$



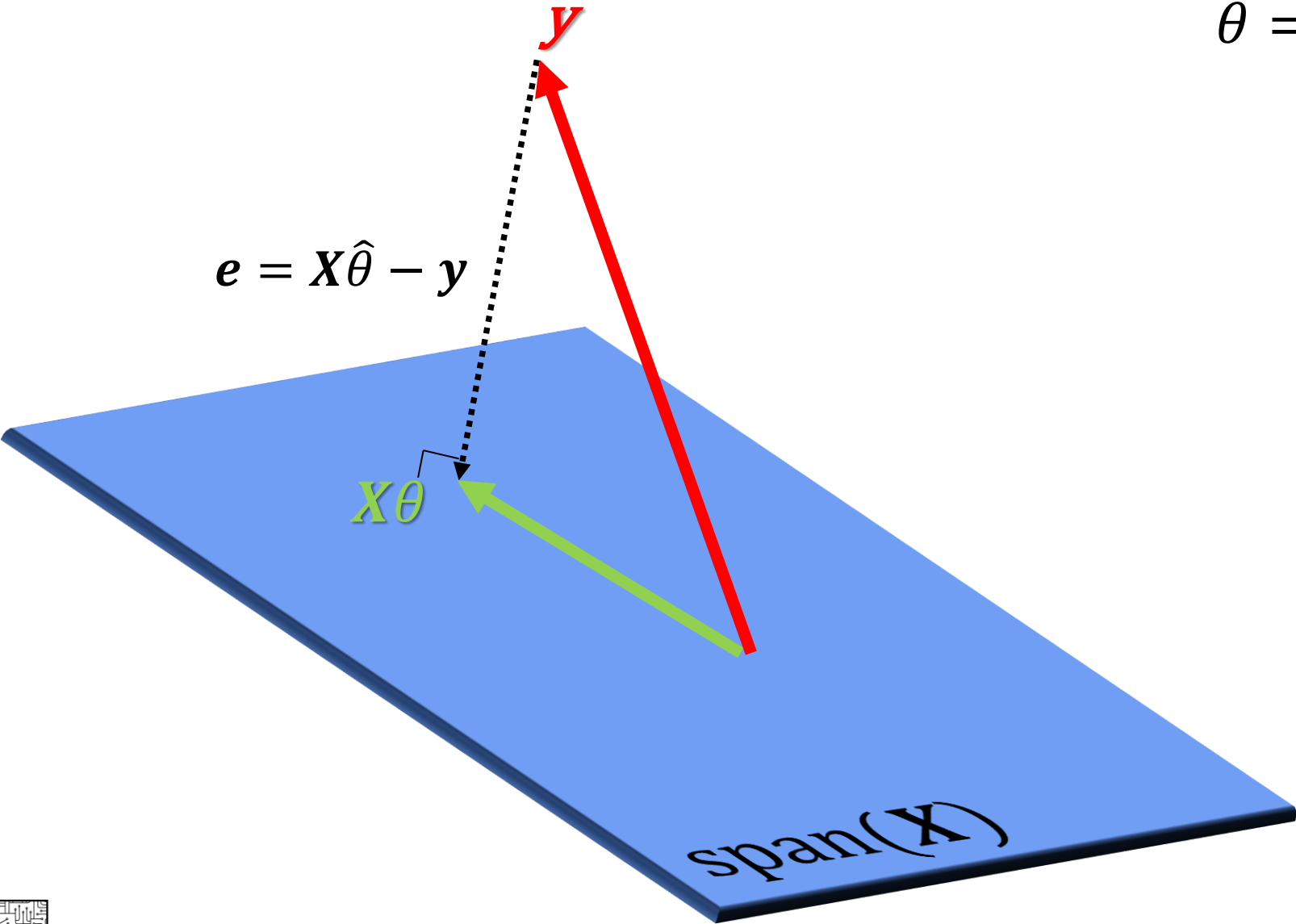
Normal Equations (intuition)

$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$

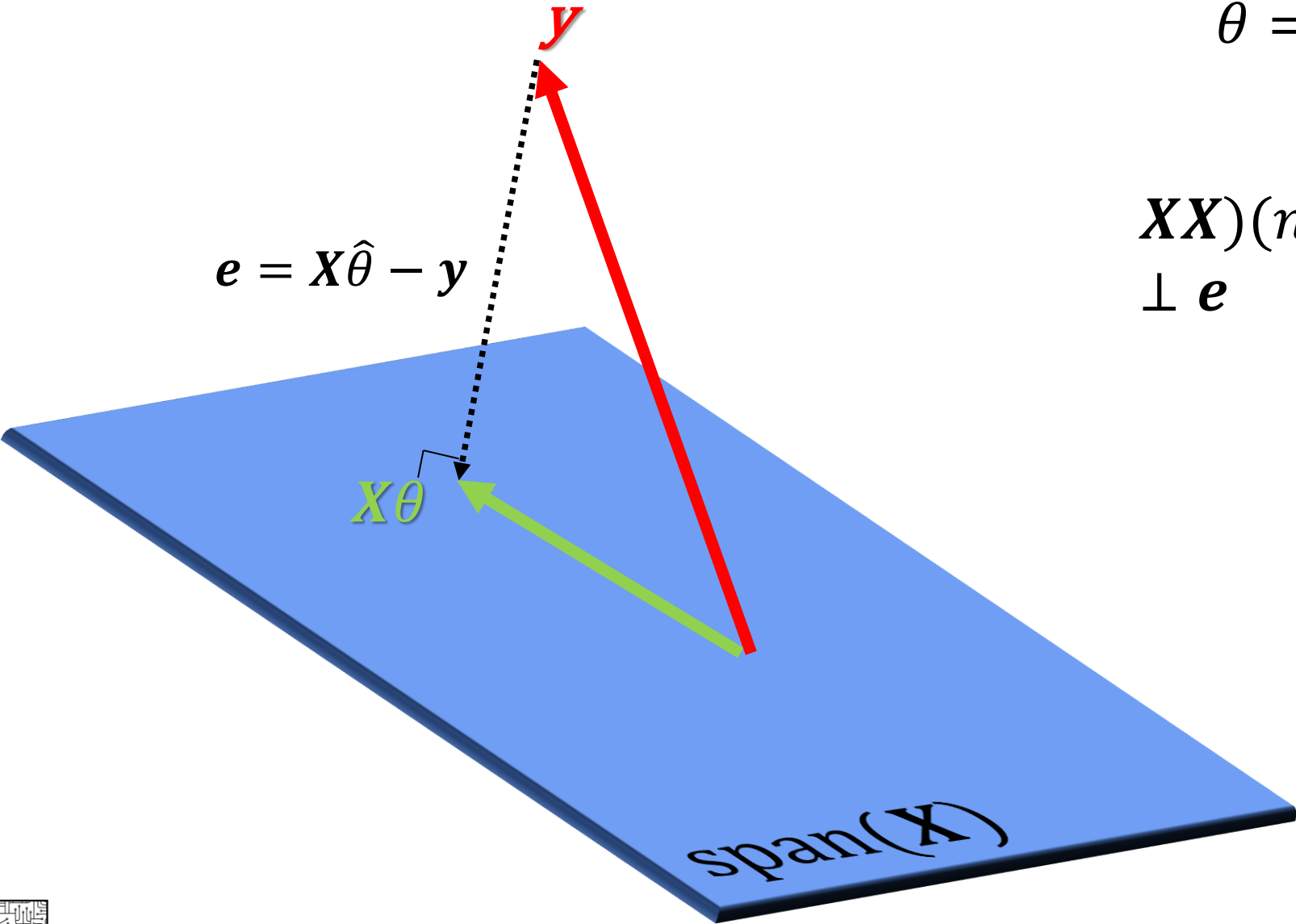


Normal Equations (intuition)

$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$



Normal Equations (intuition)

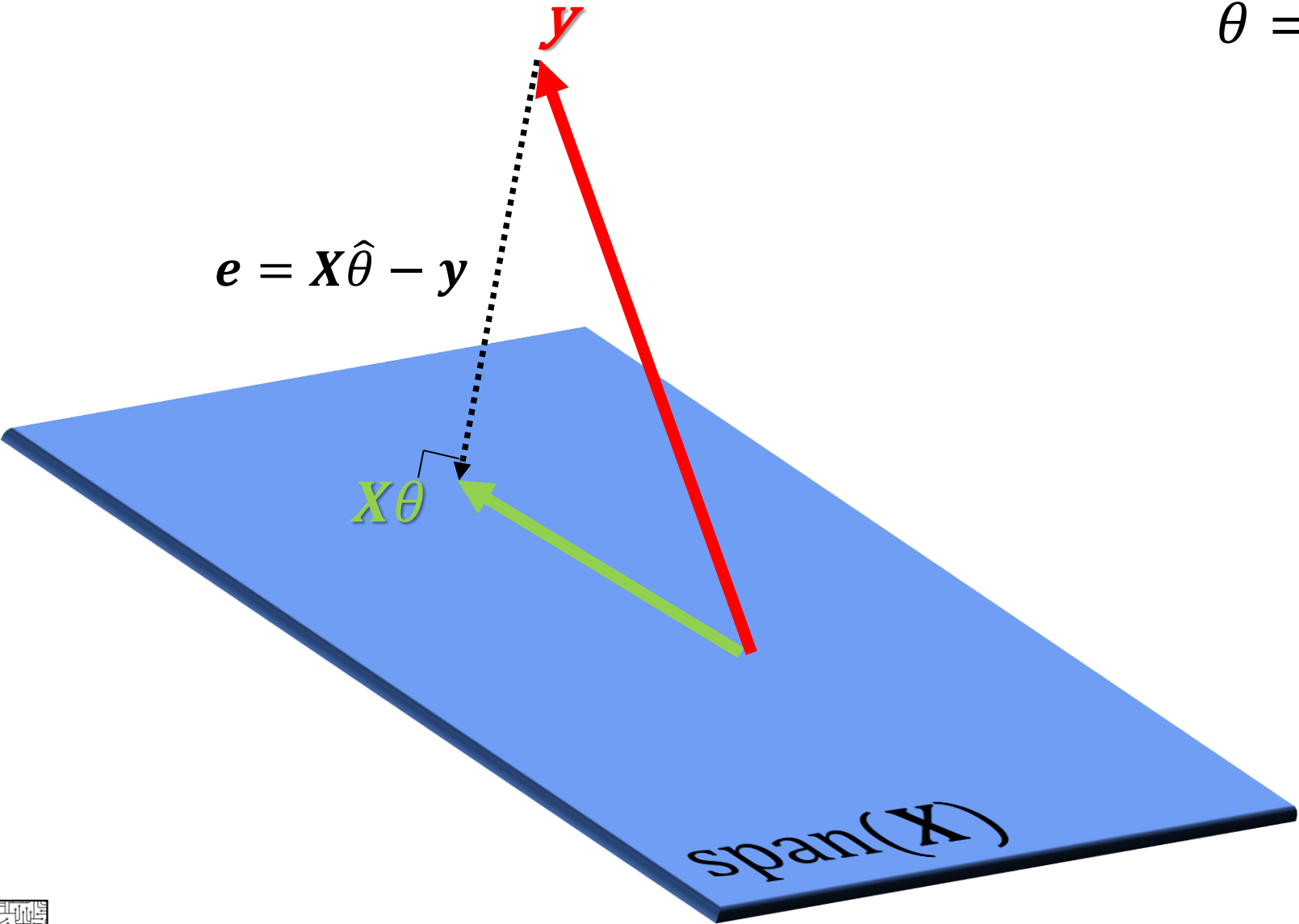


$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$

$$X^T e = 0 \quad \forall x \in \text{span}(X)$$

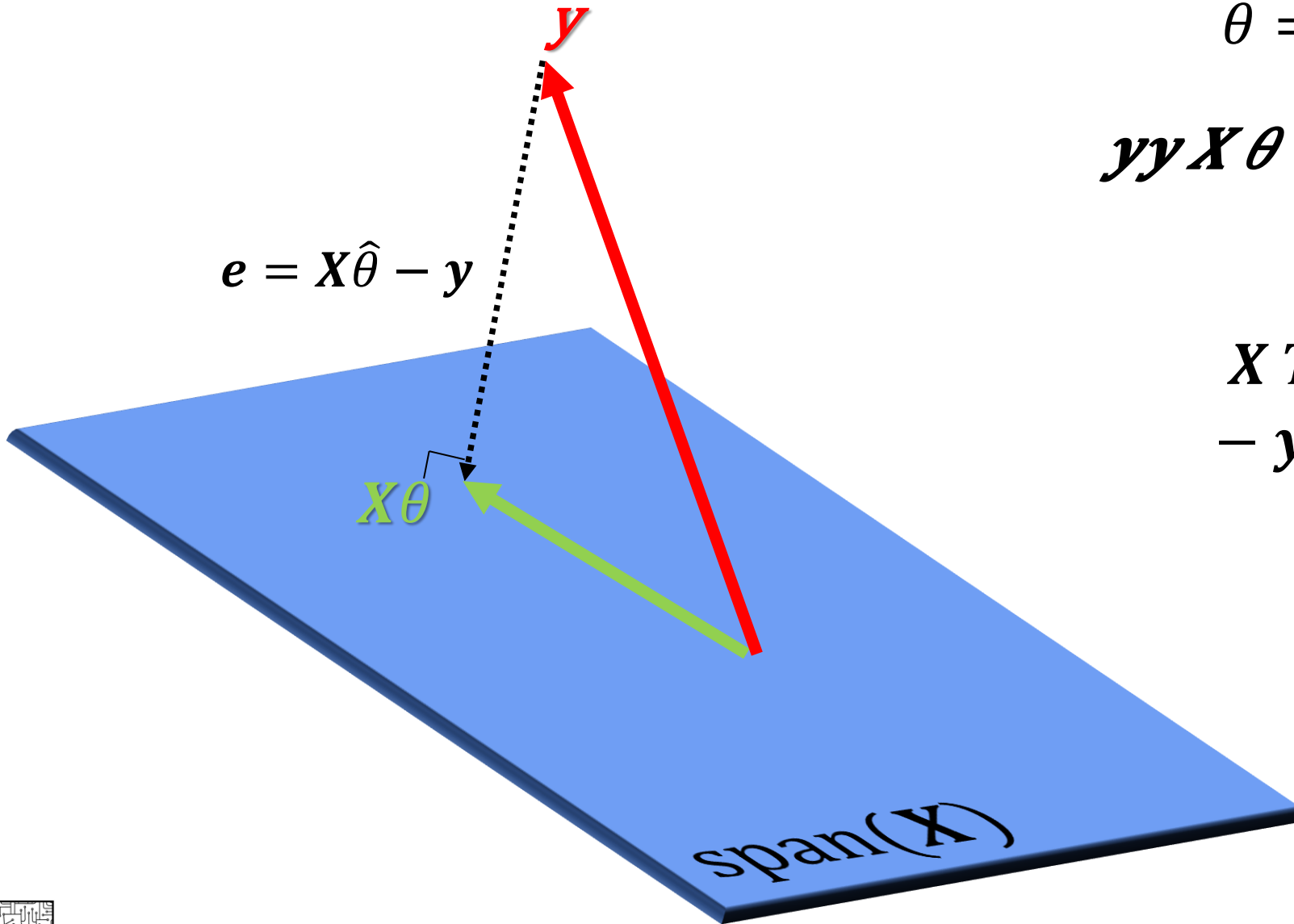
Normal Equations (intuition)

$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$



⇓

Normal Equations (intuition)

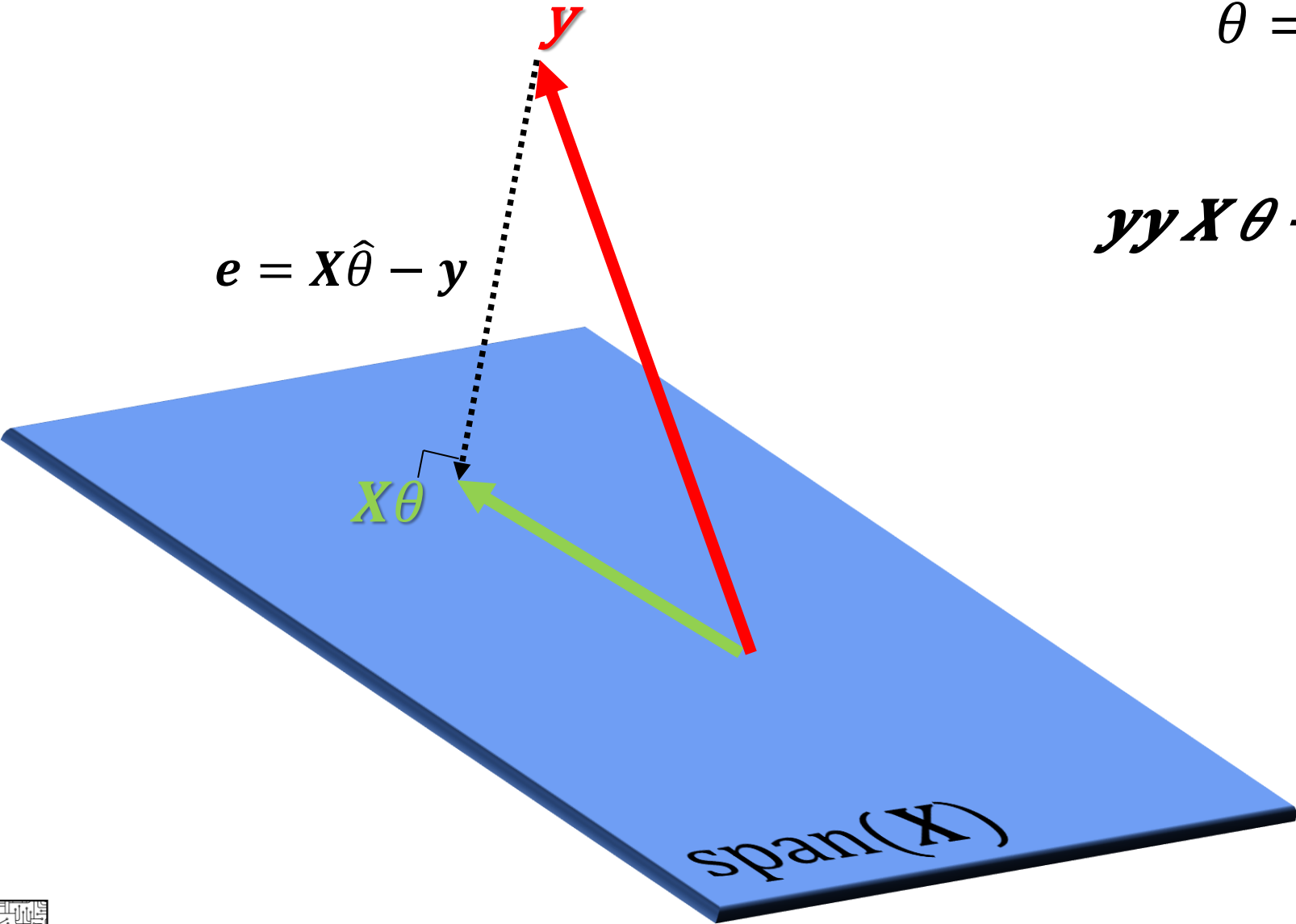


$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|y - X\theta\|^2$$

$$y - X\hat{\theta} = 0$$

$$X^T X \hat{\theta} - X^T y = 0$$

Normal Equations (intuition)

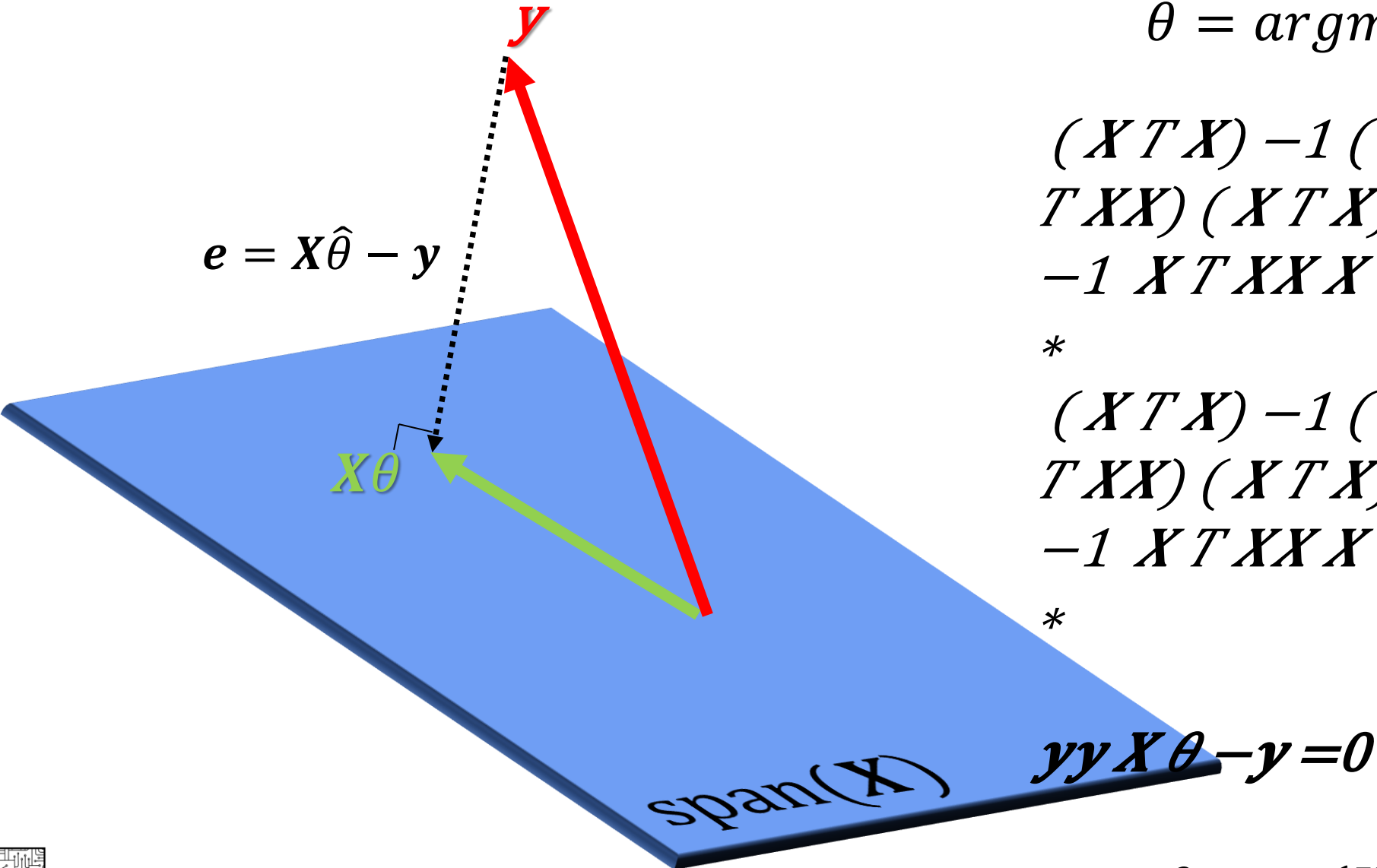


$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$

$$y - X\theta = 0$$

⇓

Normal Equations (intuition)



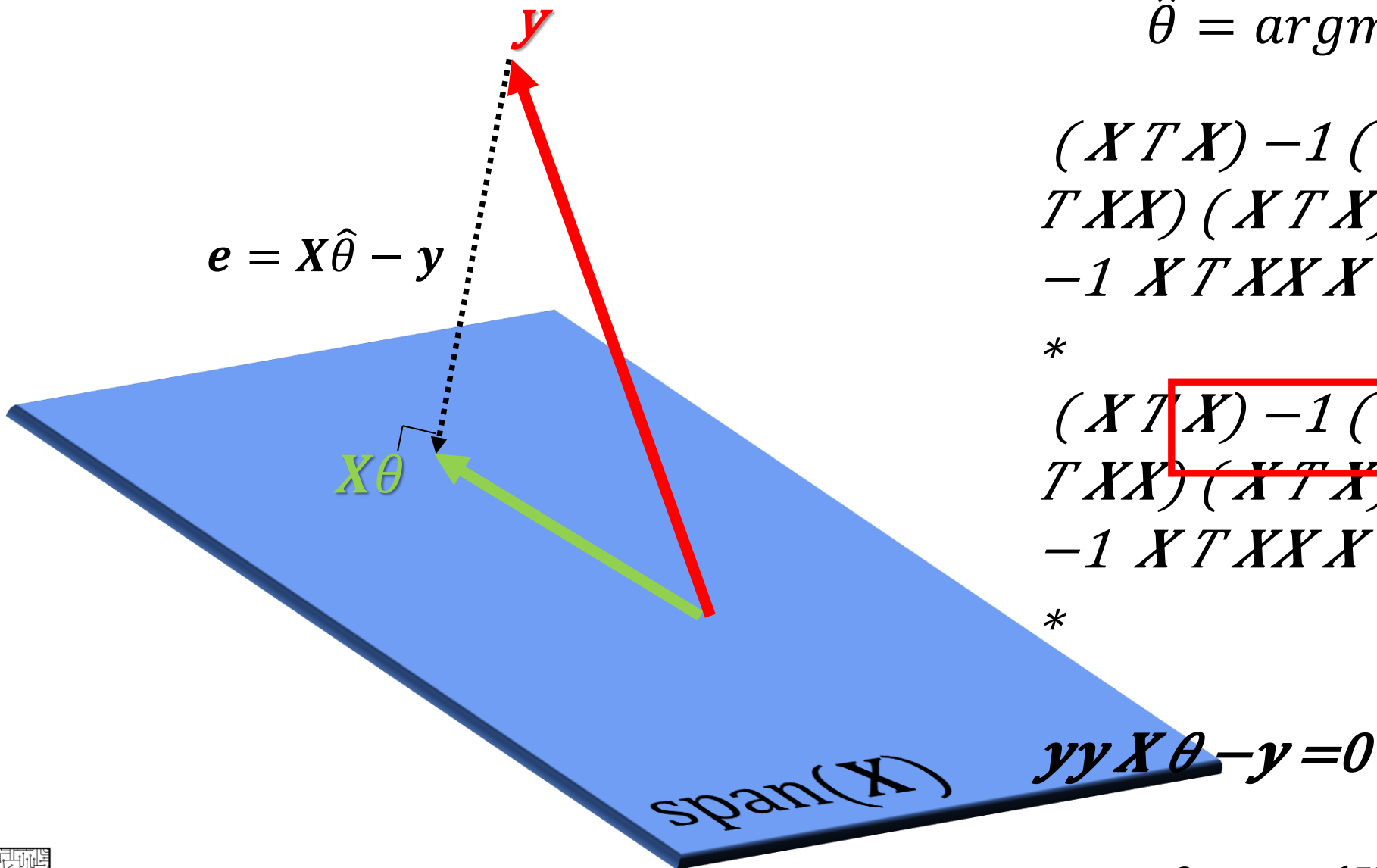
$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$

$$\begin{aligned} & (X^T X)^{-1} (X^T X X^T T T X \\ & T X X) (X^T X)^{-1} - 1 (X^T X) \\ & - 1 X^T X X X^T T T X^T y y * * * \\ & * \\ & (X^T X)^{-1} (X^T X X X^T T T X \\ & T X X) (X^T X)^{-1} - 1 (X^T X) \\ & - 1 X^T X X X^T T T X^T y y * * * \\ & * \end{aligned}$$

$$y y X \theta - y = 0$$

$$\theta = = (X^T X)^{-1} X^T y$$

Normal Equations (intuition)



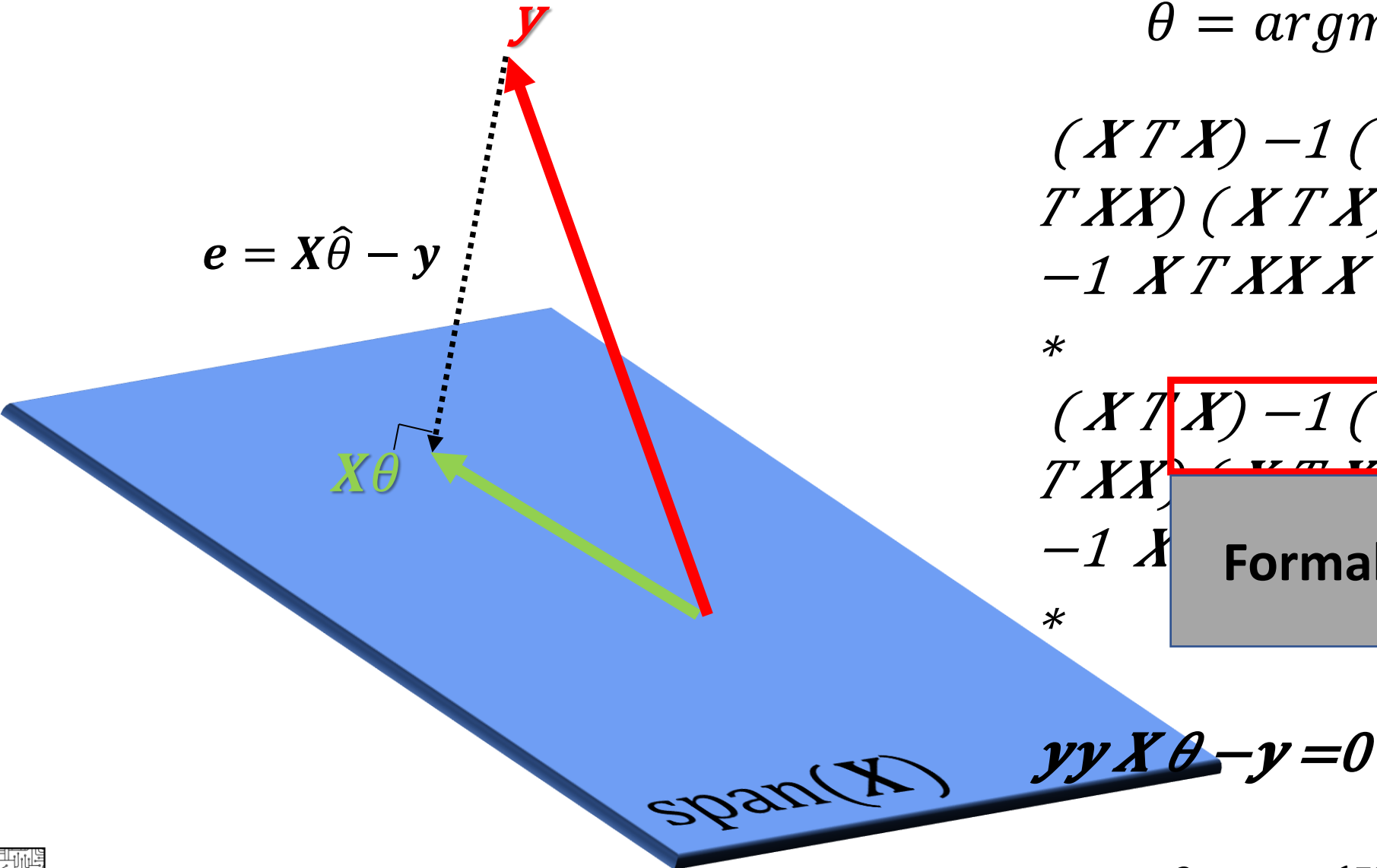
$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$

$$\begin{aligned} & (X^T X)^{-1} (X^T X X^T T T X \\ & T X X) (X^T X)^{-1} - 1 (X^T X) \\ & - 1 X^T X X X^T T T X^T y y * * * \\ & * \\ & (X^T X)^{-1} (X^T X X X^T T T X \\ & T X X) (X^T X)^{-1} - 1 (X^T X) \\ & - 1 X^T X X X^T T T X^T y y * * * \\ & * \end{aligned}$$

$$\theta = = (X^T X)^{-1} X^T y$$

(Note: The original image contains some garbled text and symbols in this block, including 'invertible' and 'X^T X'.)

Normal Equations (intuition)



$$\hat{\theta} = \operatorname{argmin}_{\theta} \|y - X\theta\|^2$$

$$(X^T X)^{-1} (X^T X X^T T T X$$

$$T X X) (X^T X)^{-1} - 1 (X^T X)$$

$$- 1 X^T X X X^T T T X^T y y * * *$$

$$*$$

$$(X^T X)^{-1} (X^T X X X^T T T X$$

$$T X X) (X^T X)^{-1} - 1 (X^T X)$$

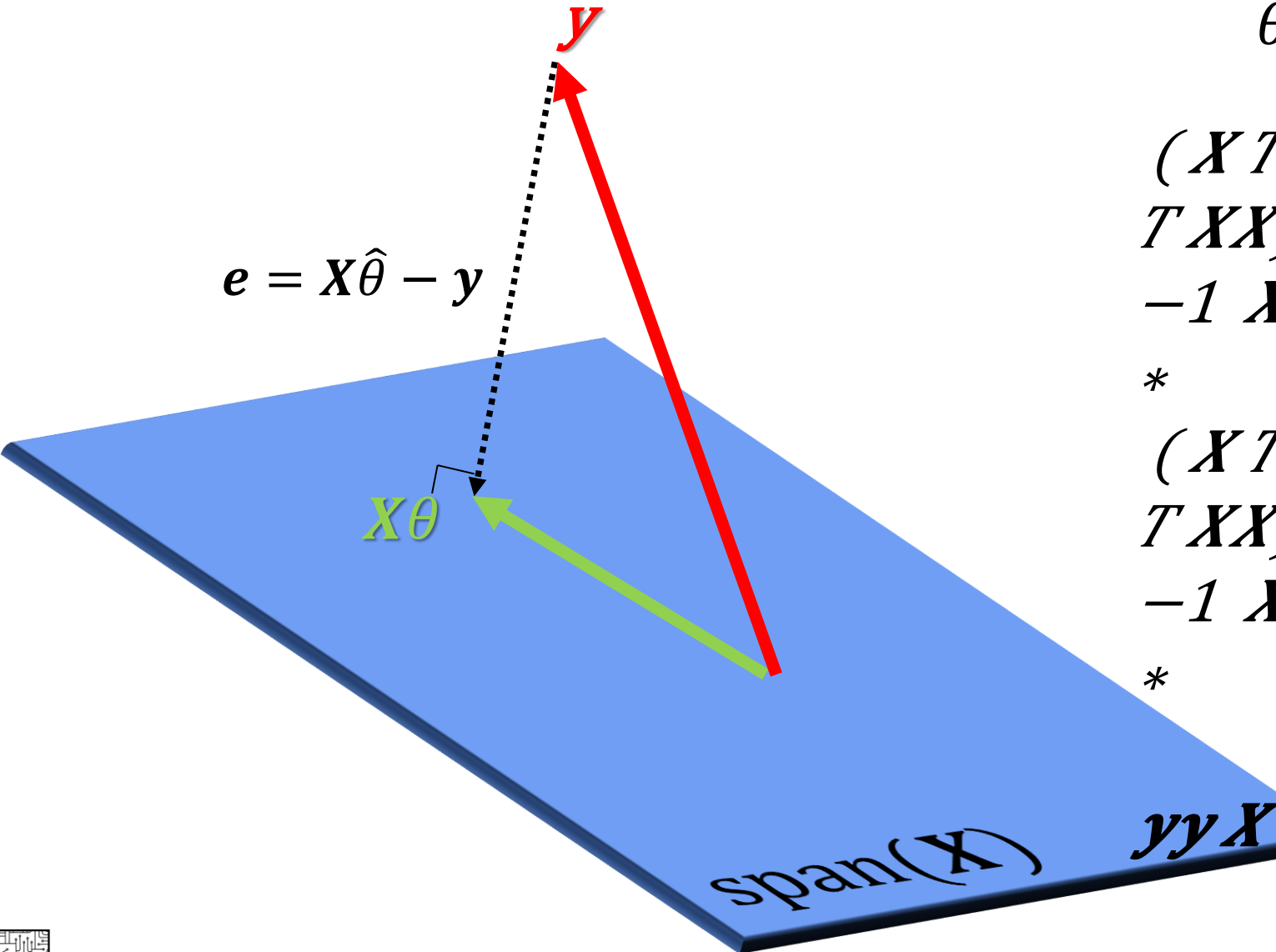
$$- 1 X * * *$$

$$*$$

Formal proof: HW

$$\theta = (X^T X)^{-1} X^T y$$

Normal Equations (intuition)



$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|y - X\theta\|^2$$

$$(X^T X)^{-1} (X^T X X^T T T X$$

$$T X X) (X^T X)^{-1} - 1 (X^T X)$$

$$- 1 X^T X X X^T T T X^T y y * * *$$

$$* (X^T X)^{-1} (X^T X X X^T T T X$$

$$T X X) (X^T X)^{-1} - 1 (X^T X)$$

$$- 1 X * * *$$

Formal proof: HW

Also in HW:
is $X^T X$ invertible?

$$\theta = = (X^T X)^{-1} X^T y$$

Normal Equations

X may cost unreasonable memory / time

Normal Equations

Q: Will normal equations always be practical?

X may cost unreasonable memory / time

Normal Equations

Q: Will normal equations always be practical?

A: No;

1. may cost unreasonable memory / time
2. Inverting $X^T X T T T X$ may cost unreasonable memory / time



Normal Equations

Q: Will normal equations always be practical?

A: No;

1. may cost unreasonable memory / time
2. Sometimes not applicable: Regularization? Different loss? Non-linearity?



Gradient descent



Gradient descent

Possible solution: Iteratively reduce loss



Gradient descent

Possible solution: Iteratively reduce loss



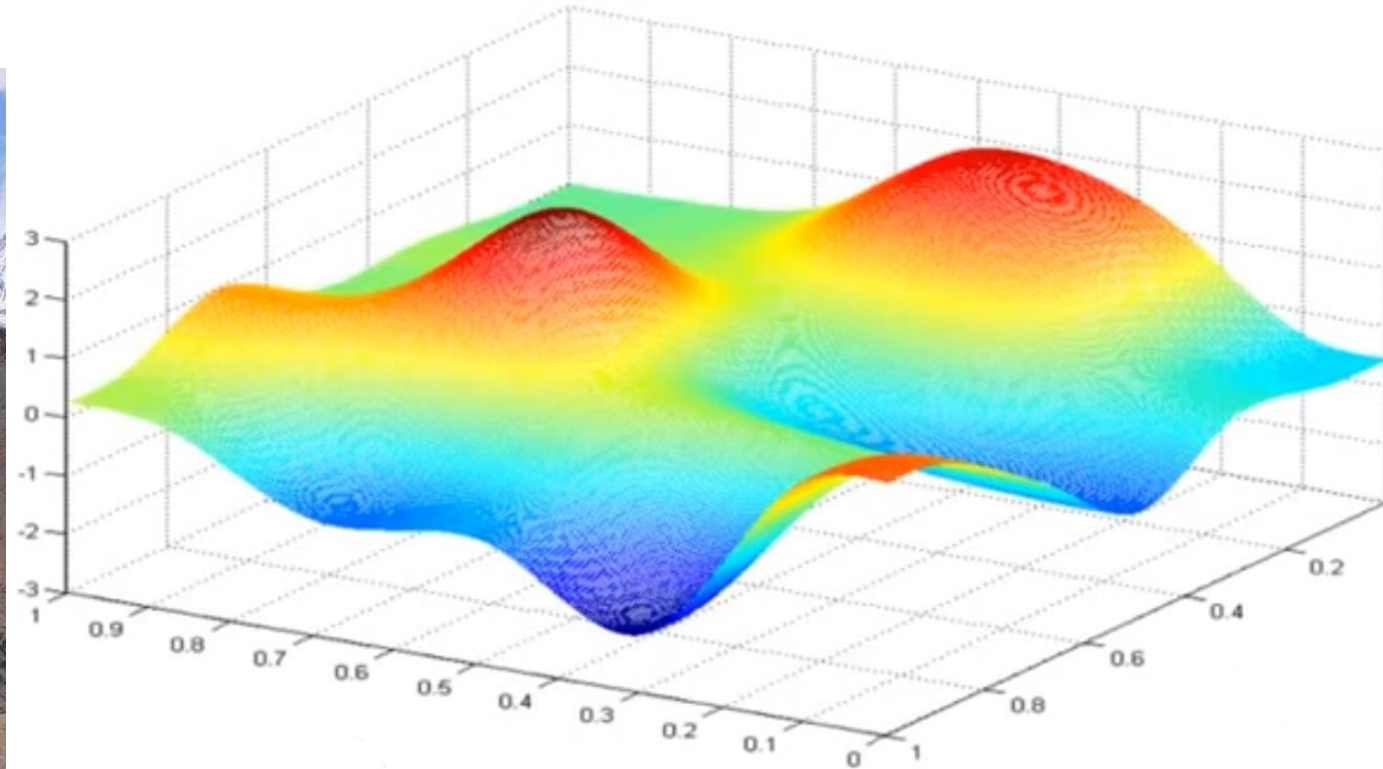
Gradient descent

Possible solution: Iteratively reduce loss



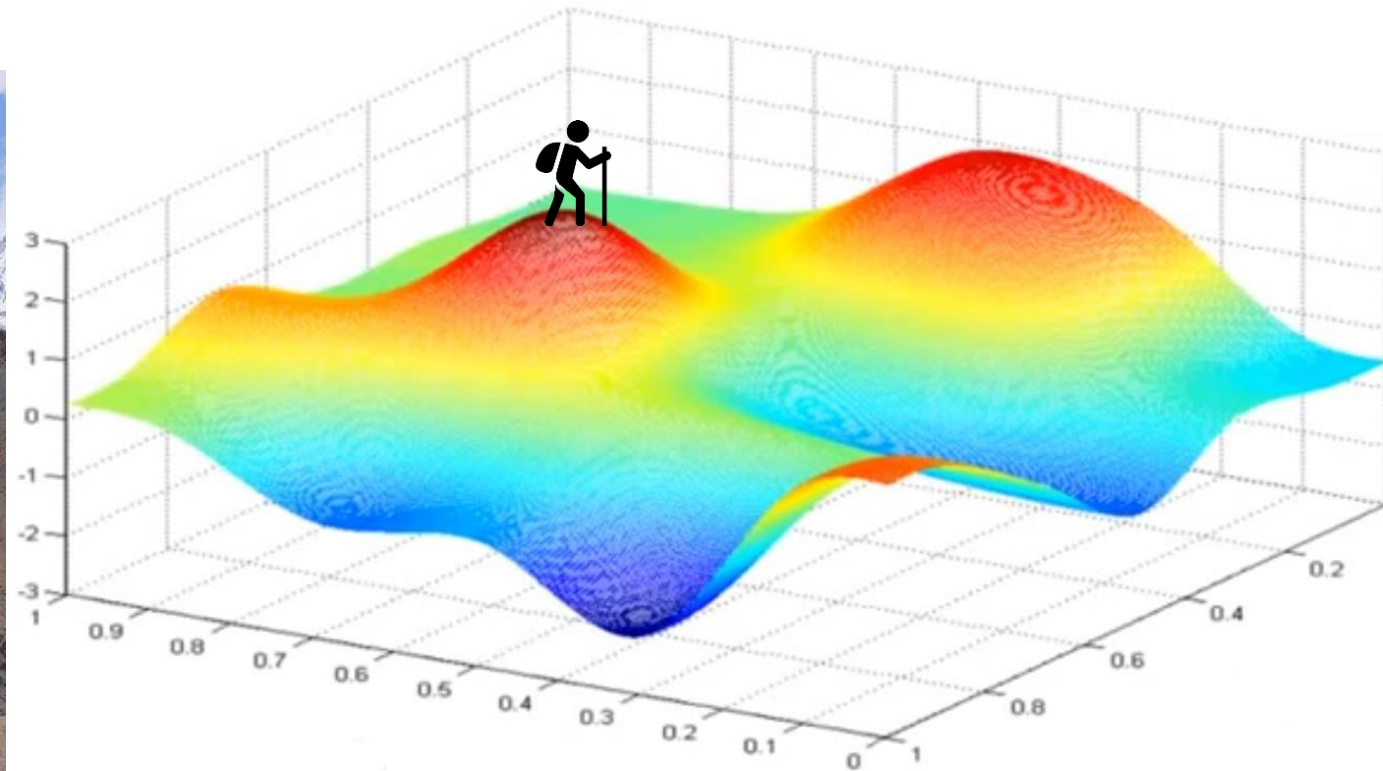
Gradient descent

Possible solution: Iteratively reduce loss



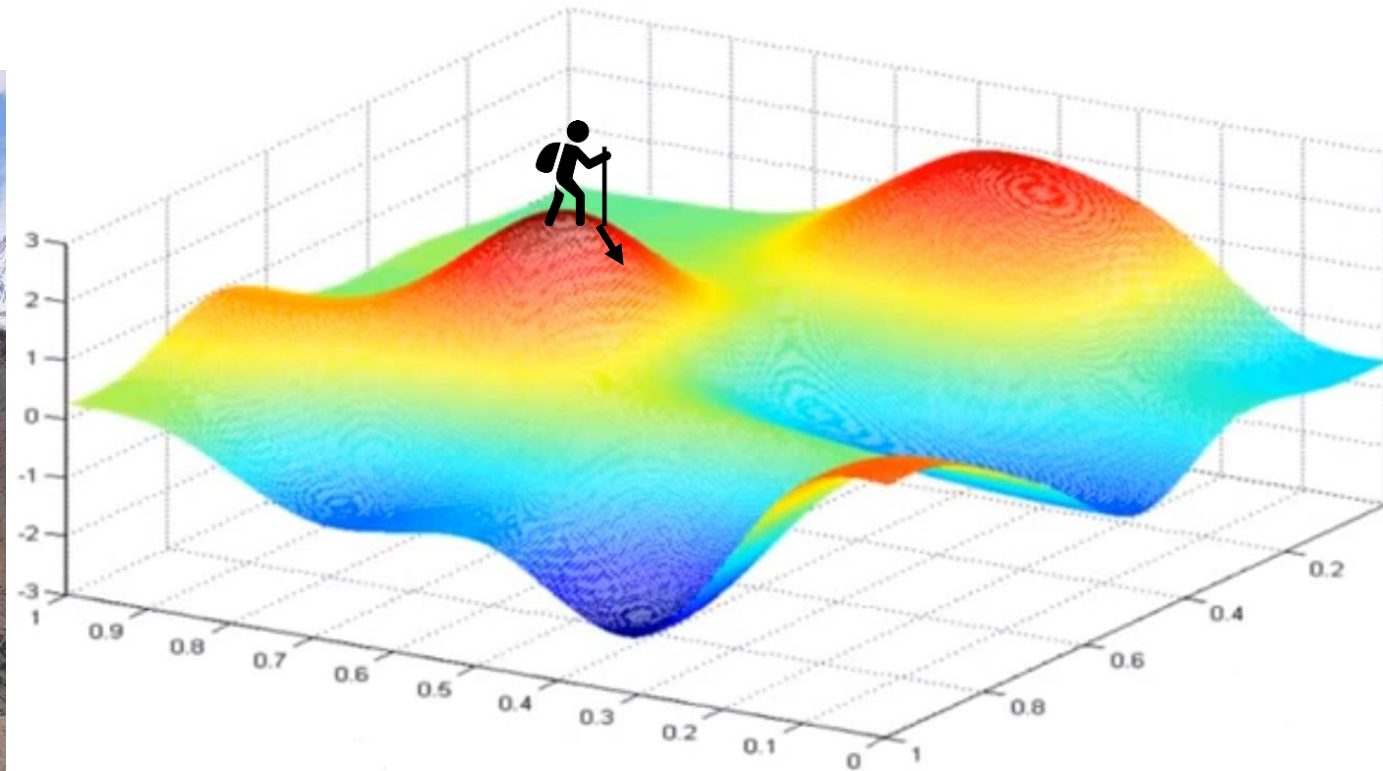
Gradient descent

Possible solution: Iteratively reduce loss



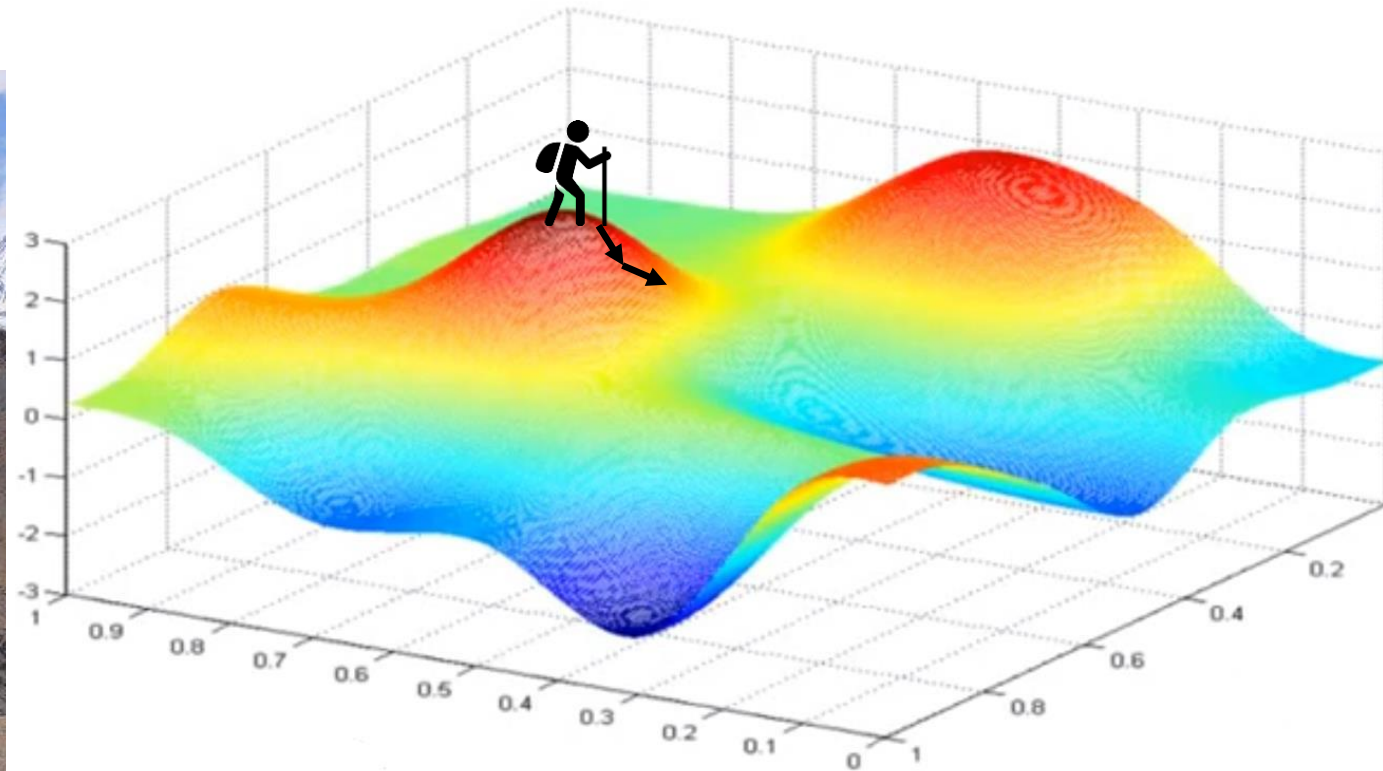
Gradient descent

Possible solution: Iteratively reduce loss



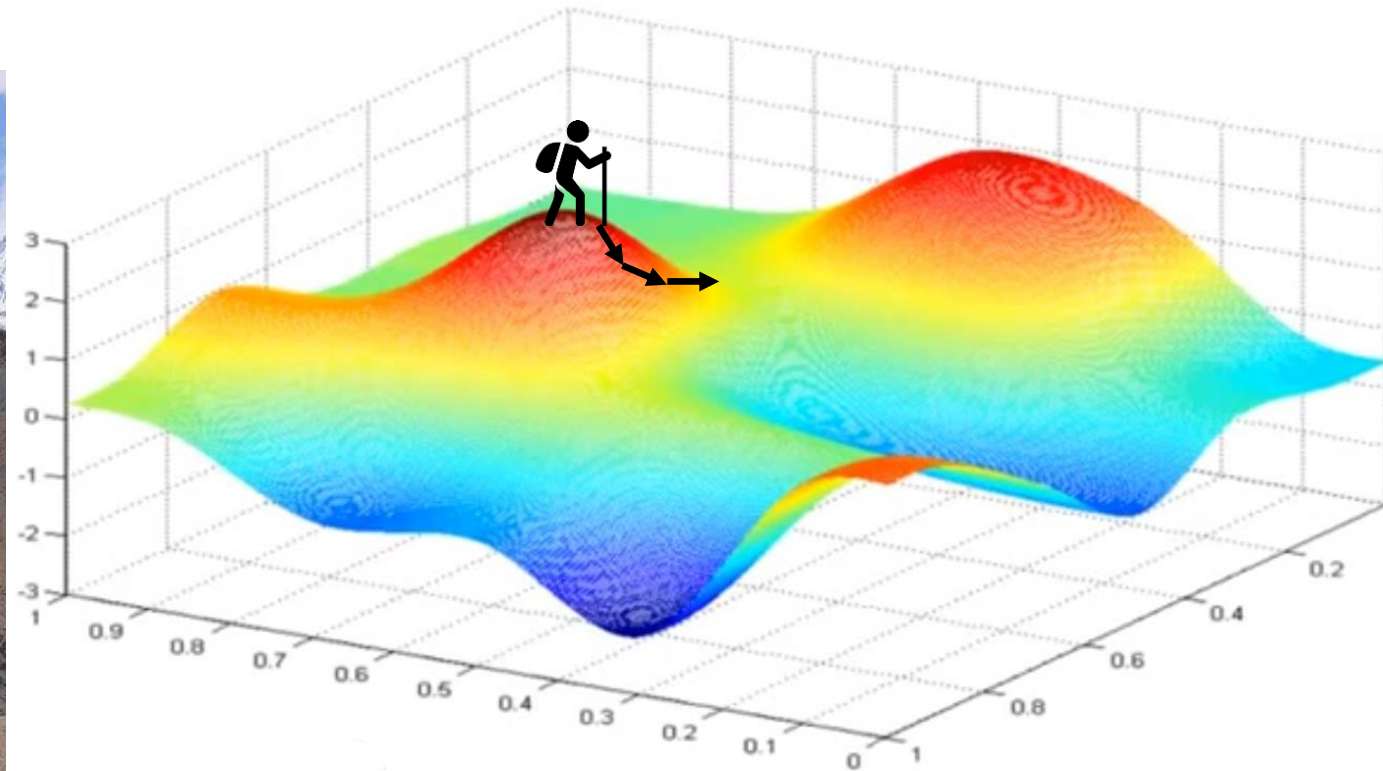
Gradient descent

Possible solution: Iteratively reduce loss



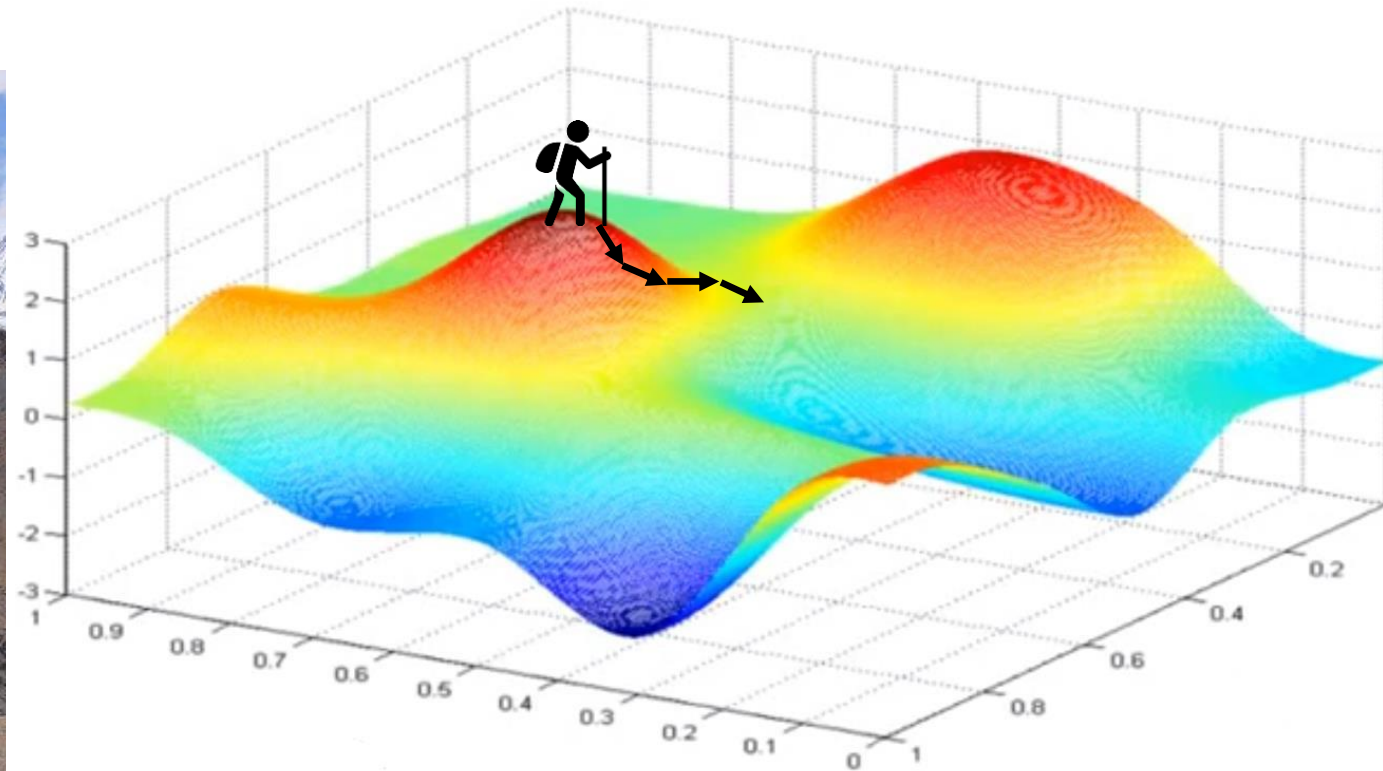
Gradient descent

Possible solution: Iteratively reduce loss



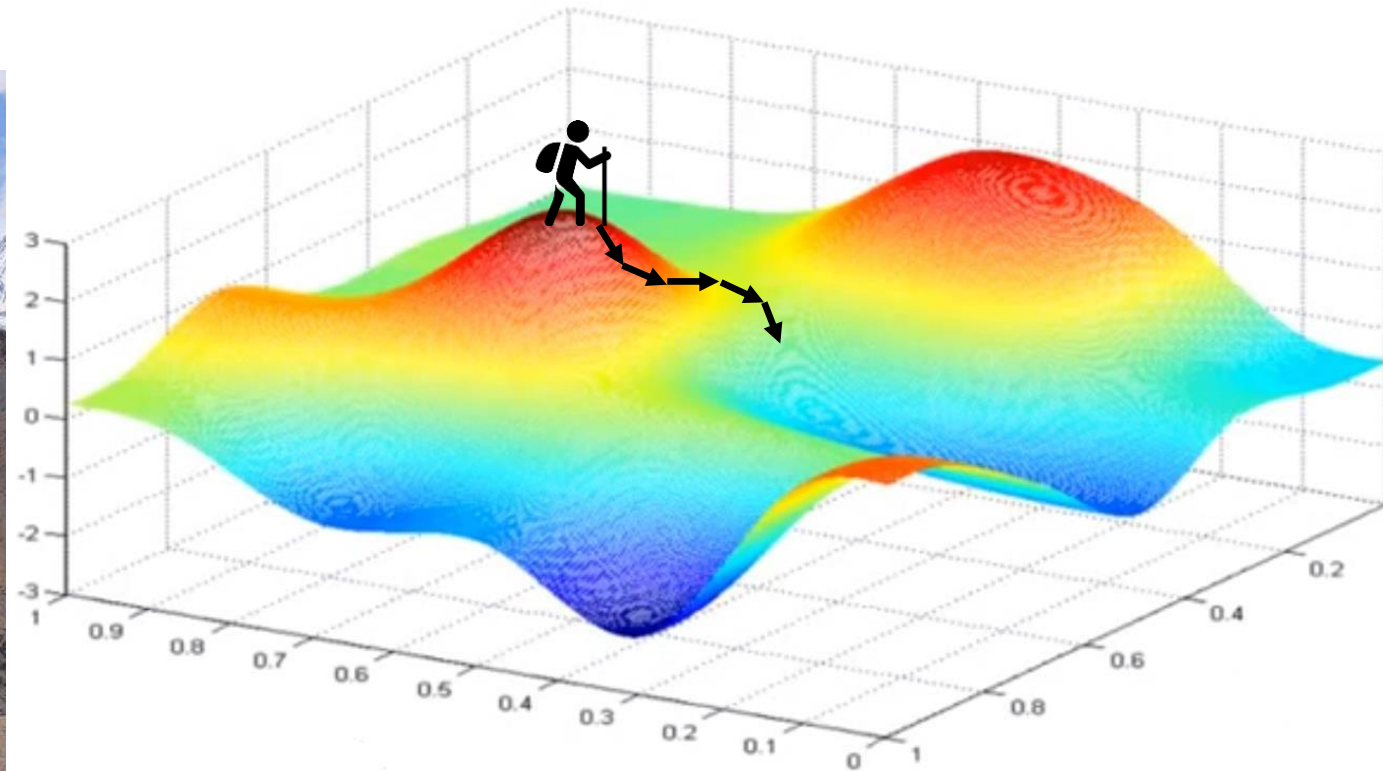
Gradient descent

Possible solution: Iteratively reduce loss



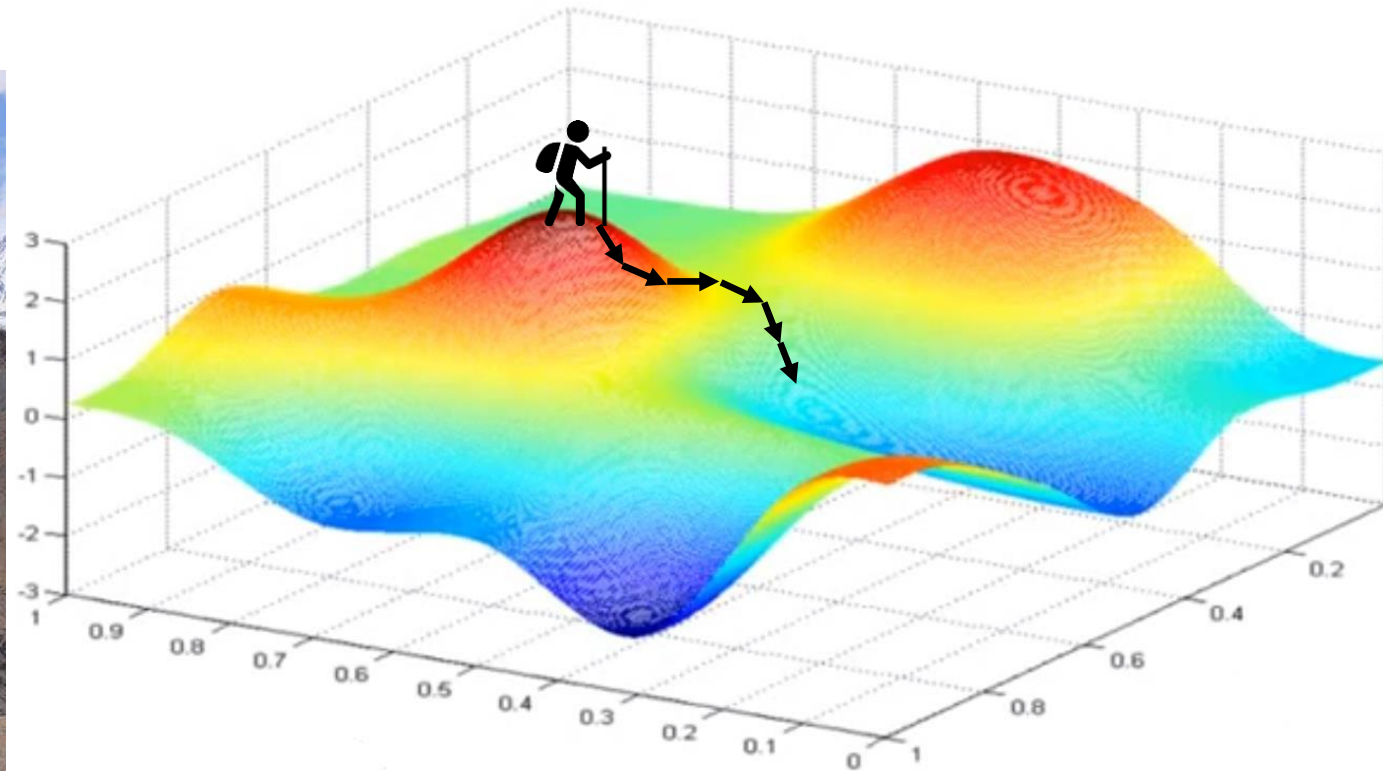
Gradient descent

Possible solution: Iteratively reduce loss



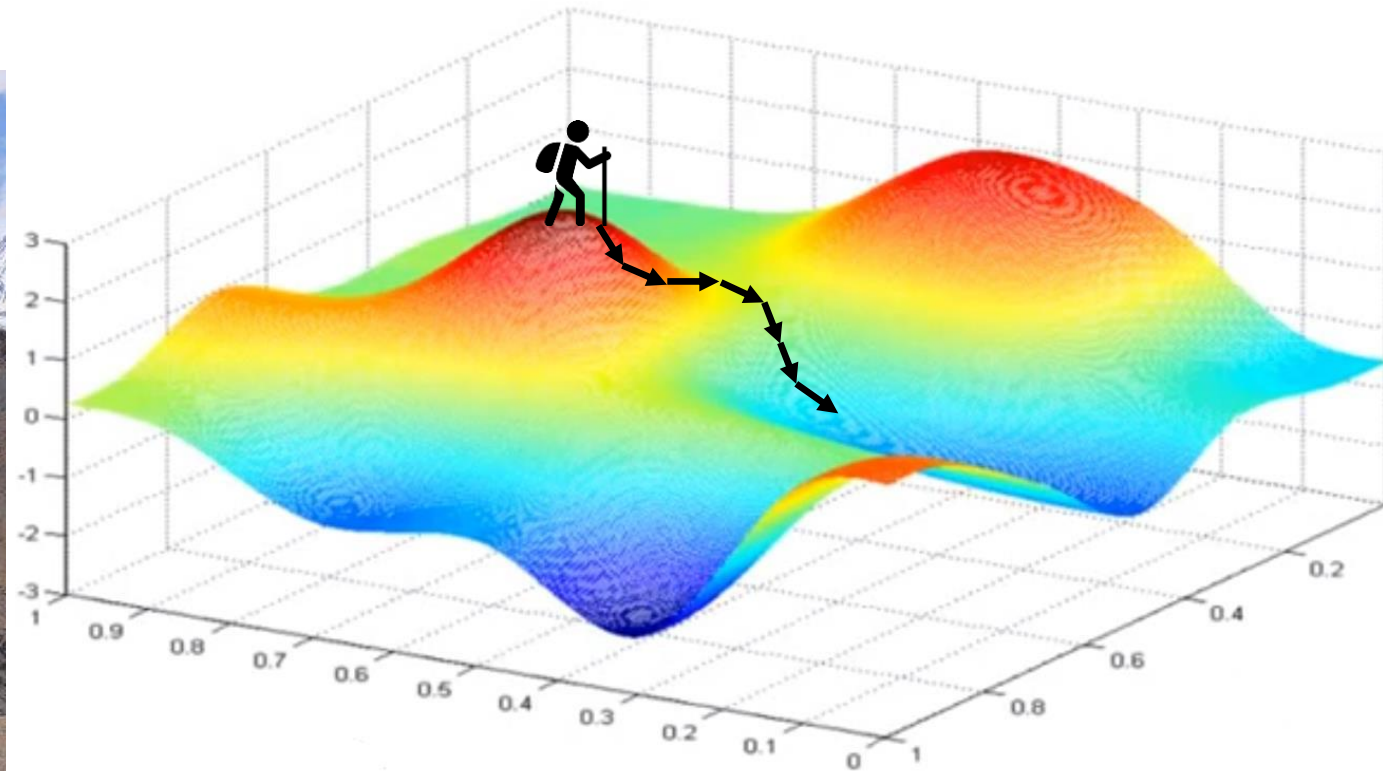
Gradient descent

Possible solution: Iteratively reduce loss



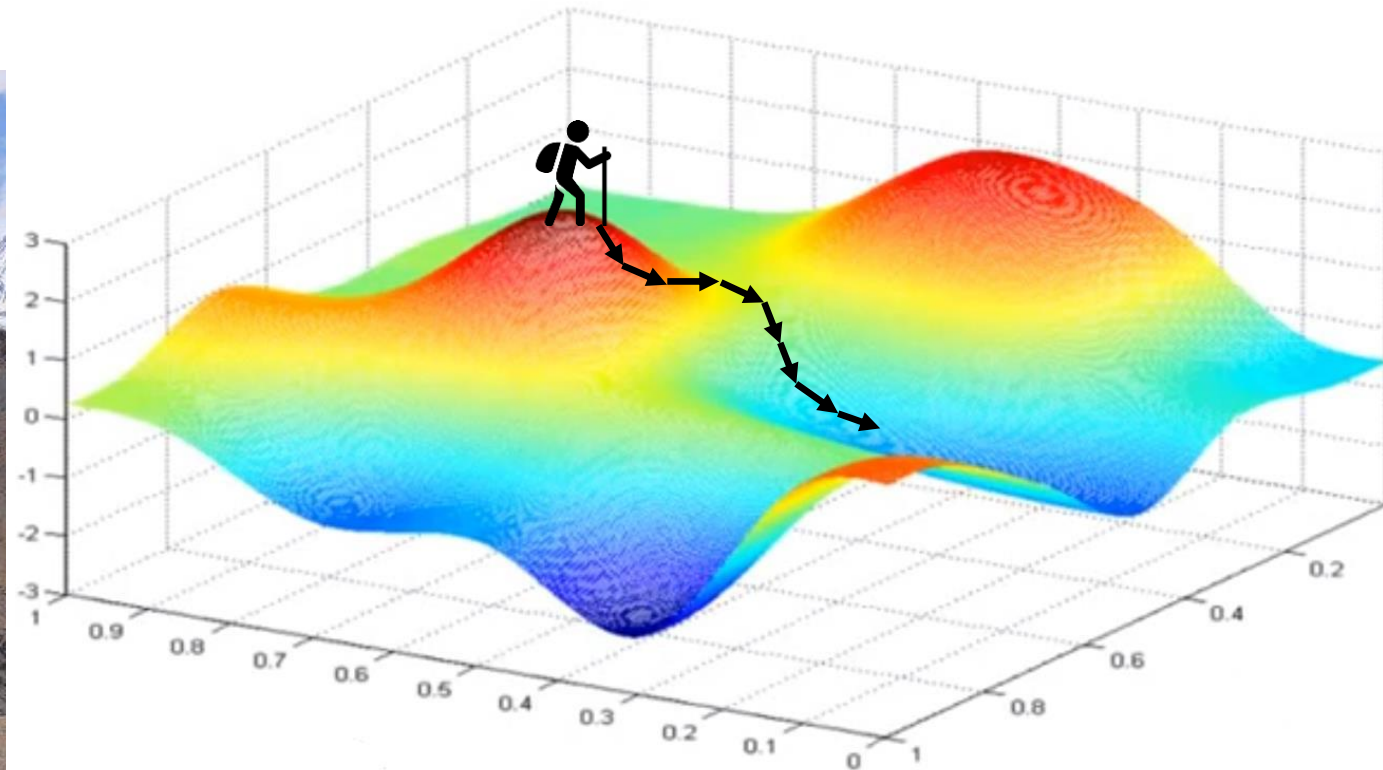
Gradient descent

Possible solution: Iteratively reduce loss



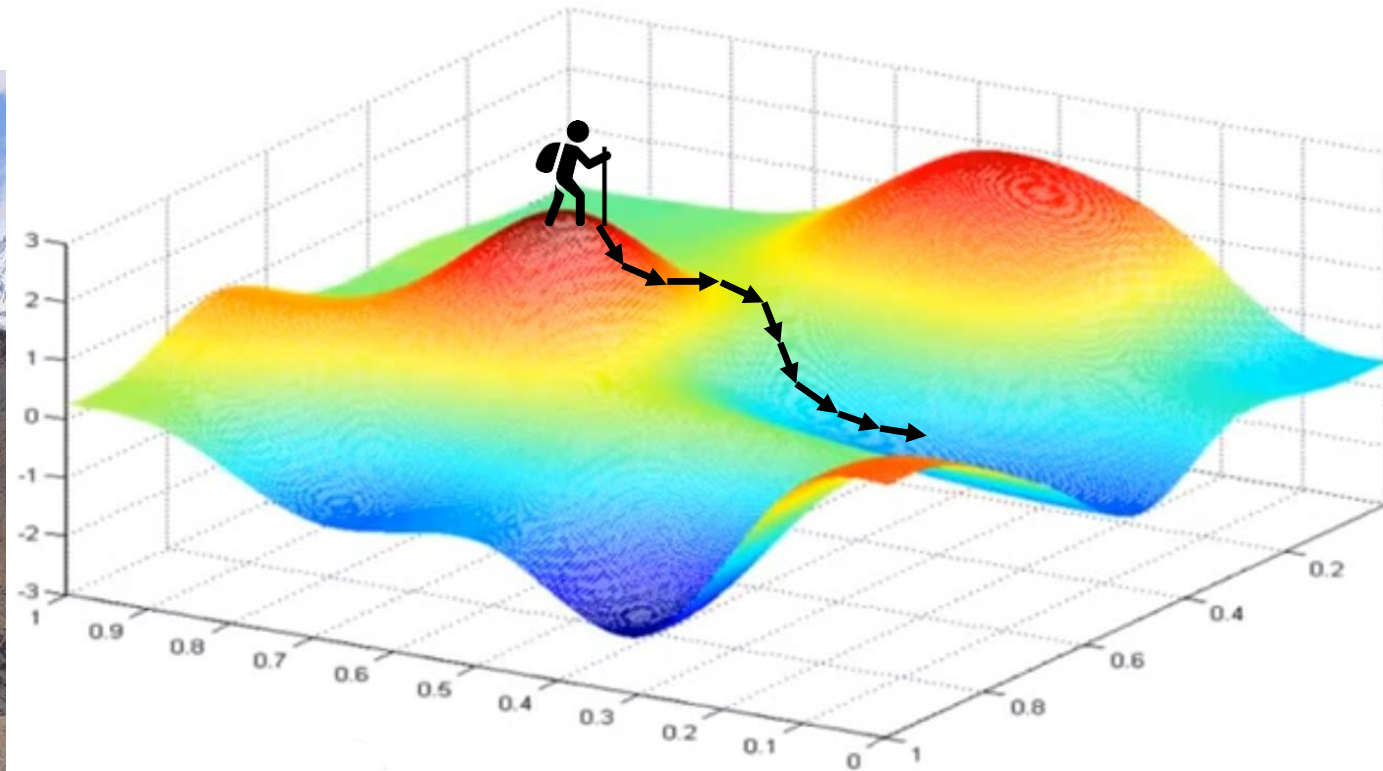
Gradient descent

Possible solution: Iteratively reduce loss



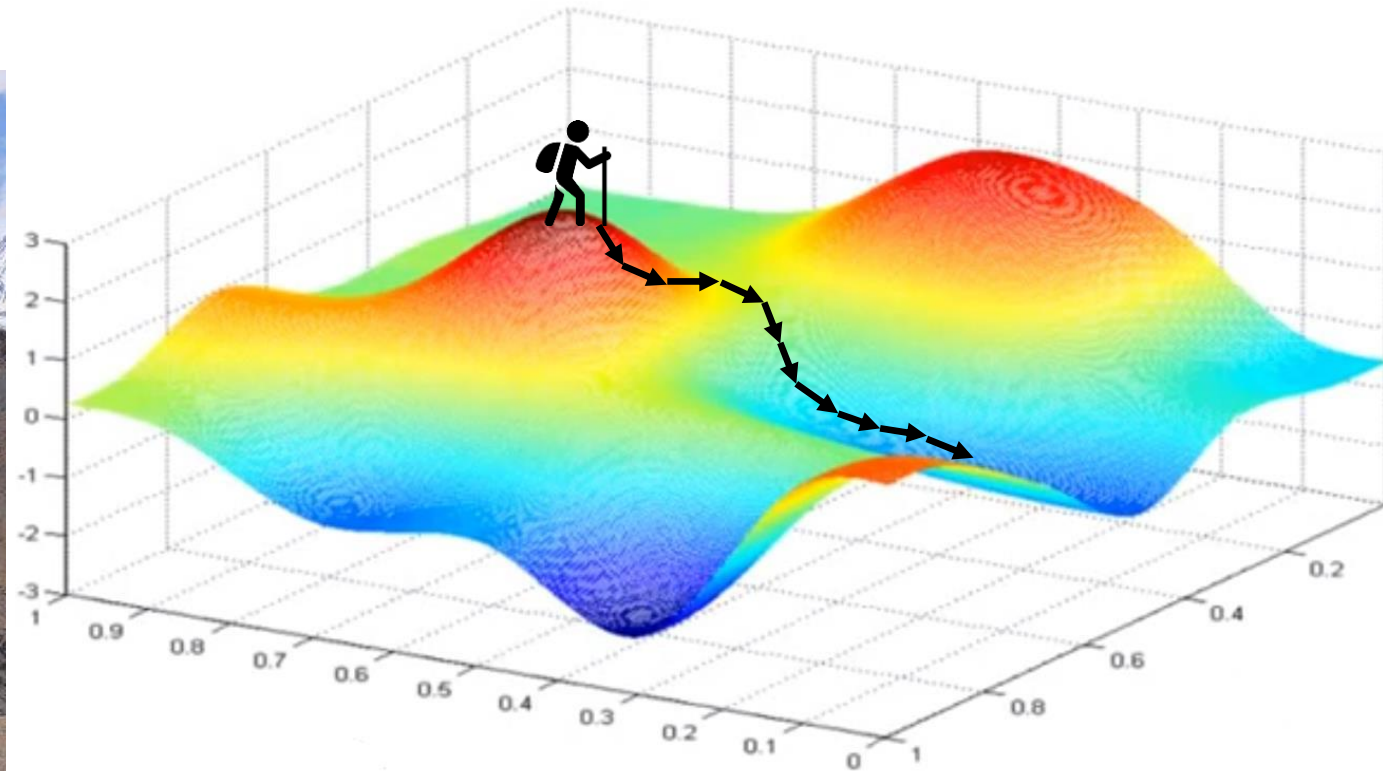
Gradient descent

Possible solution: Iteratively reduce loss



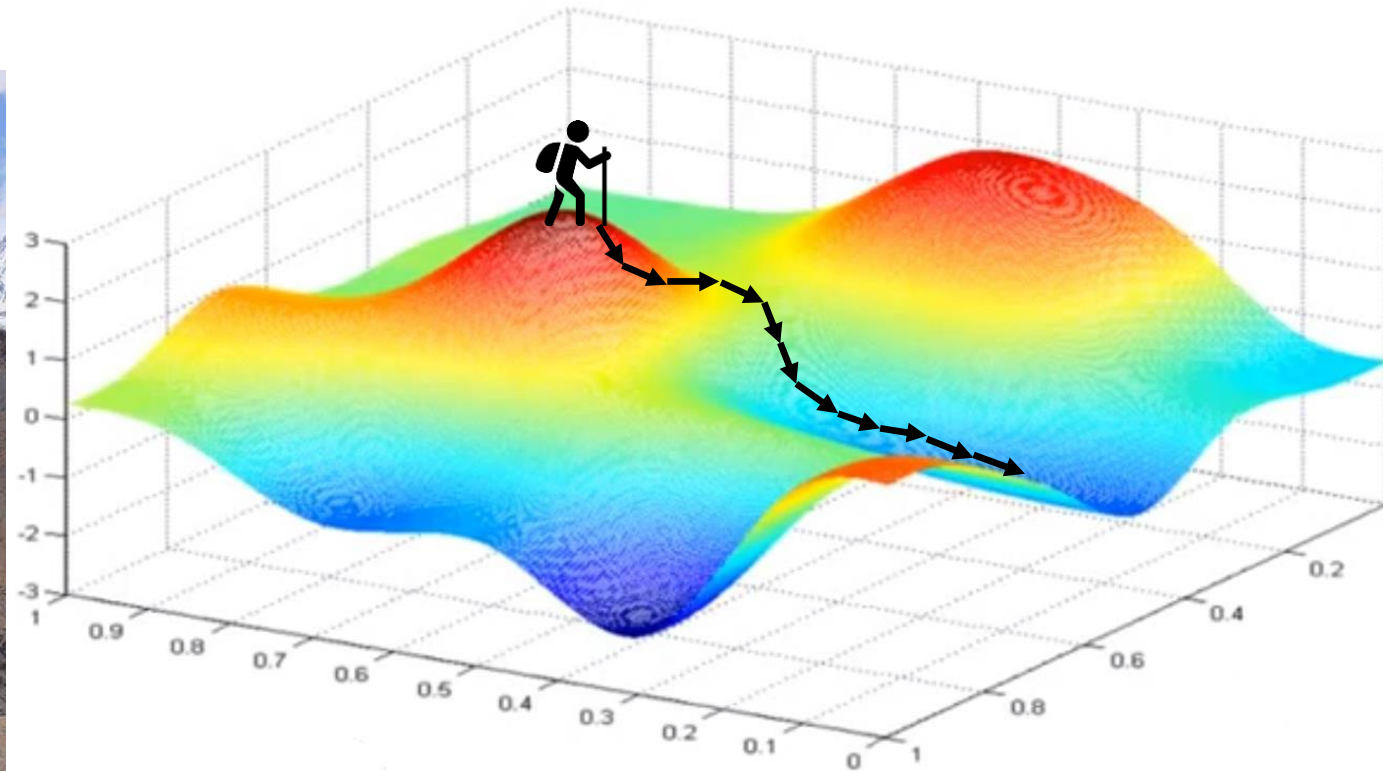
Gradient descent

Possible solution: Iteratively reduce loss



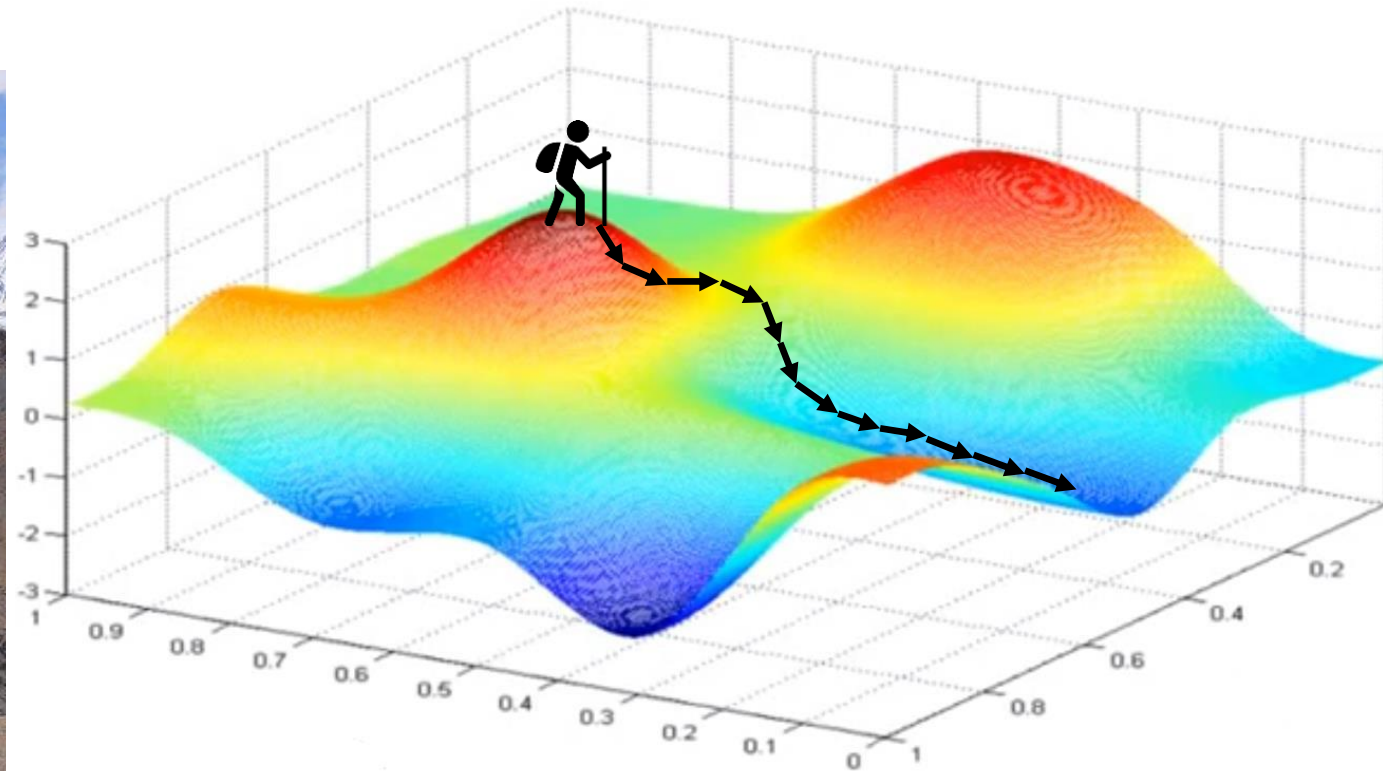
Gradient descent

Possible solution: Iteratively reduce loss



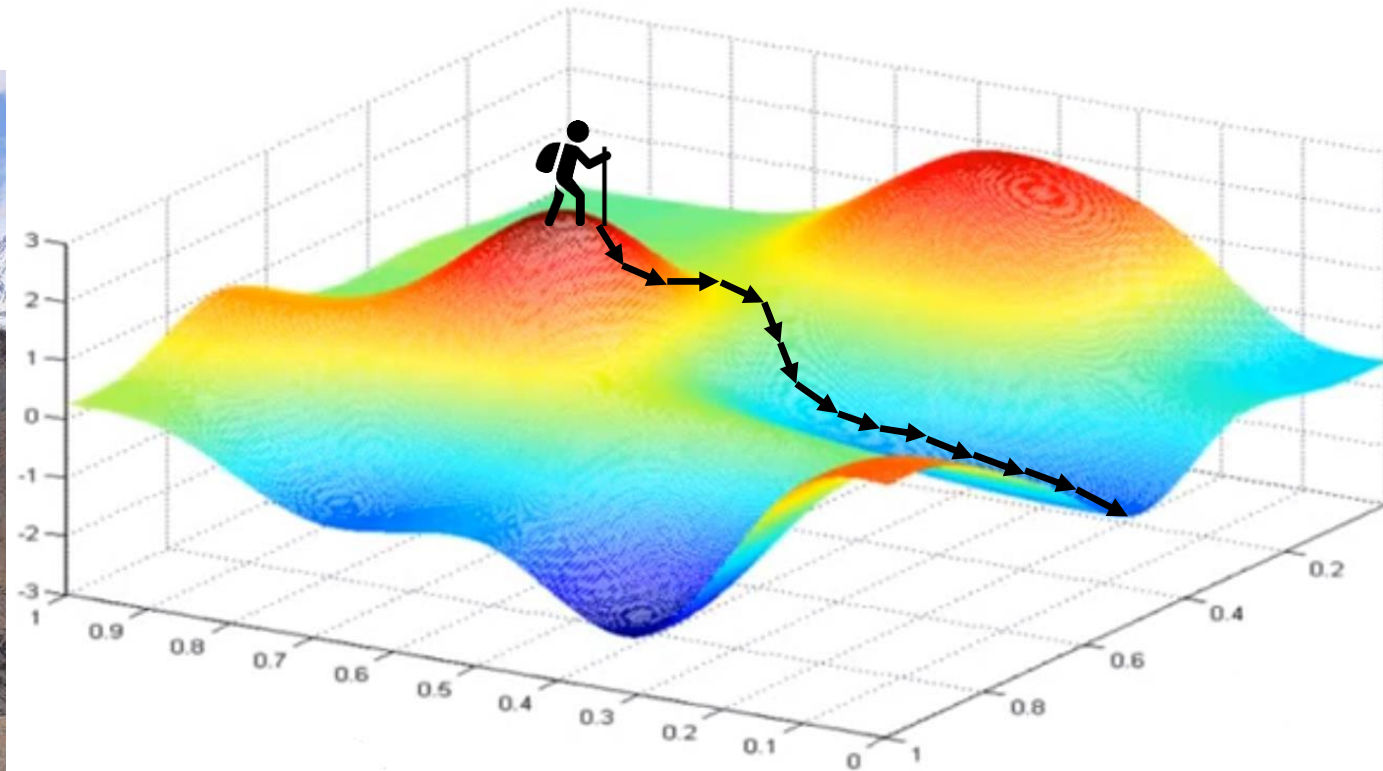
Gradient descent

Possible solution: Iteratively reduce loss



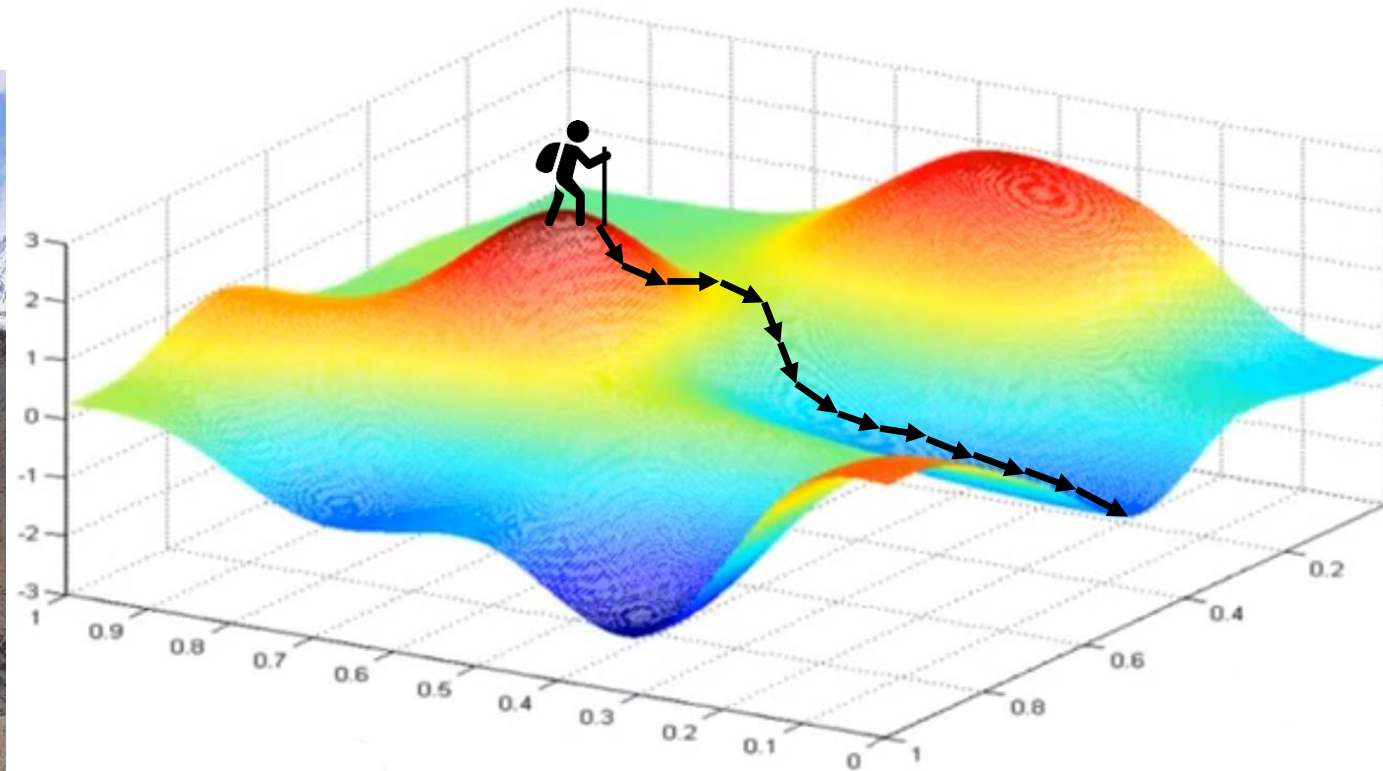
Gradient descent

Possible solution: Iteratively reduce loss



Gradient descent

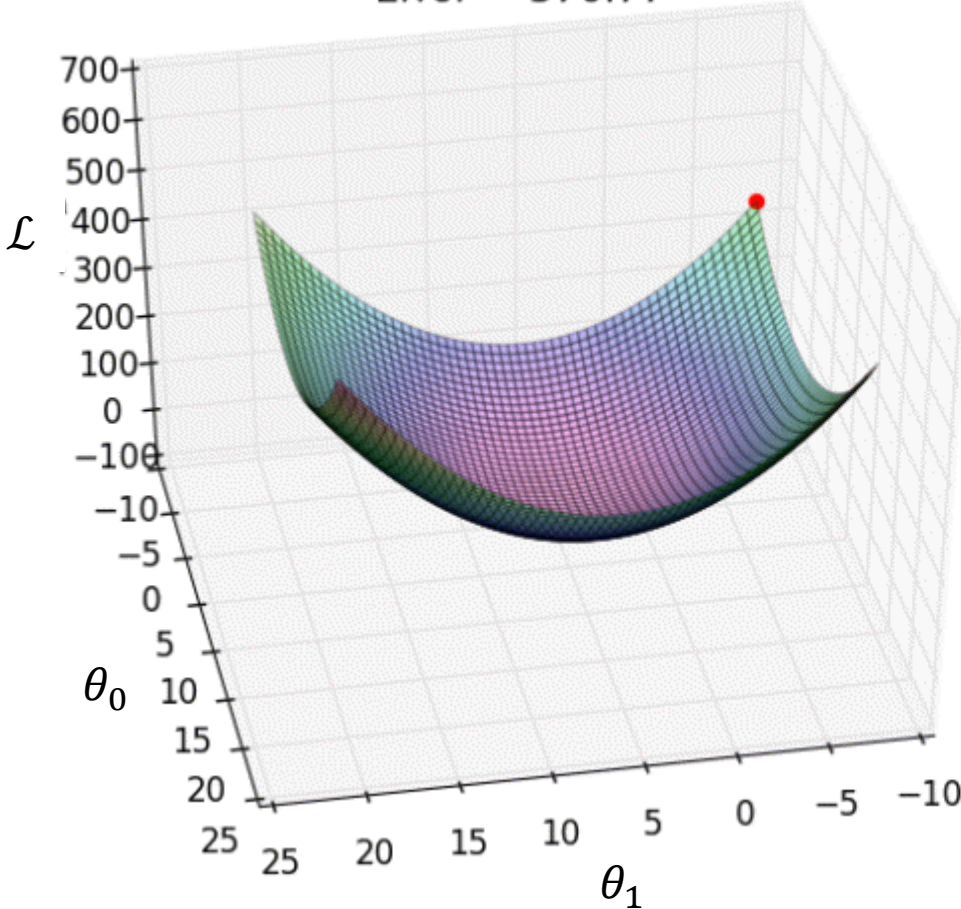
Possible solution: Iteratively reduce loss



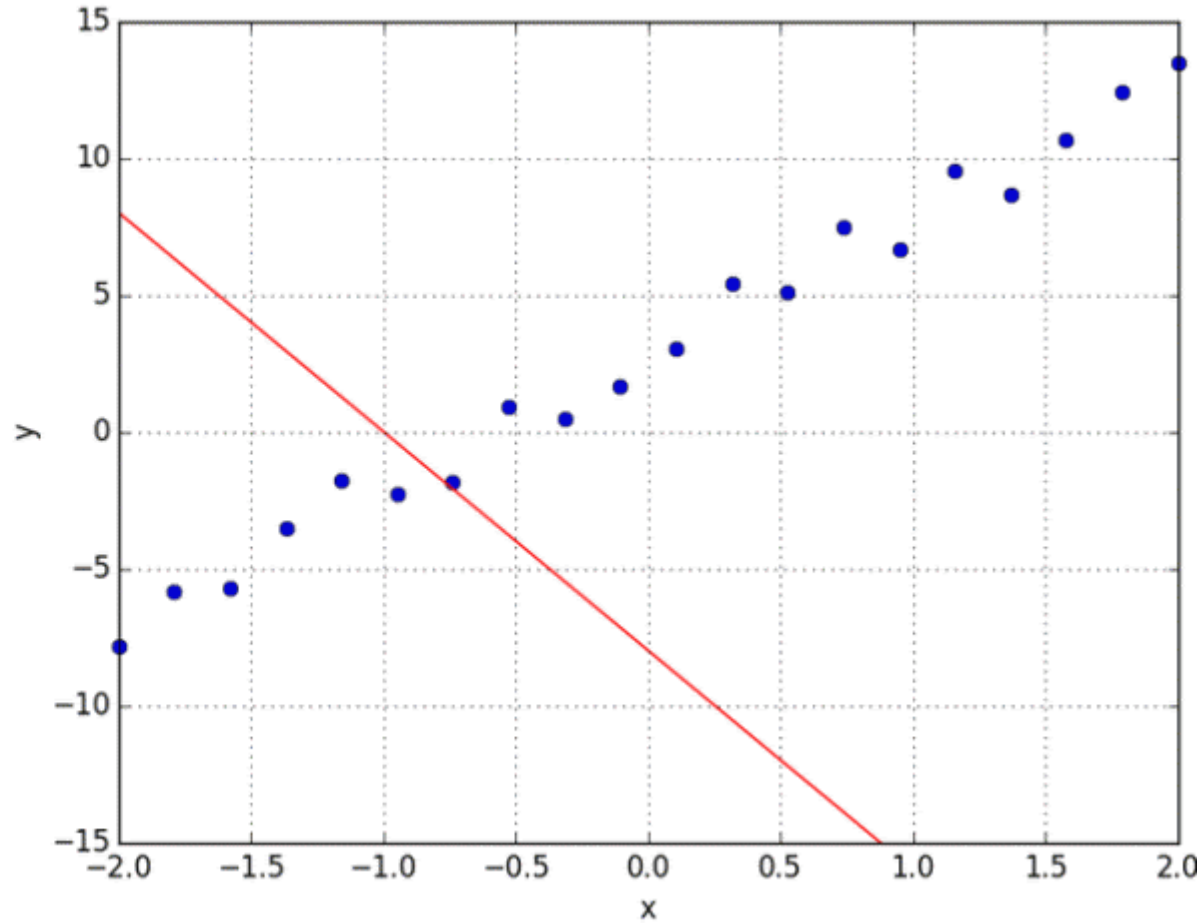
Can we guarantee global min?

Gradient descent

Error = 370.77



Parameter space: $\mathcal{L}(\boldsymbol{\theta}; S)$

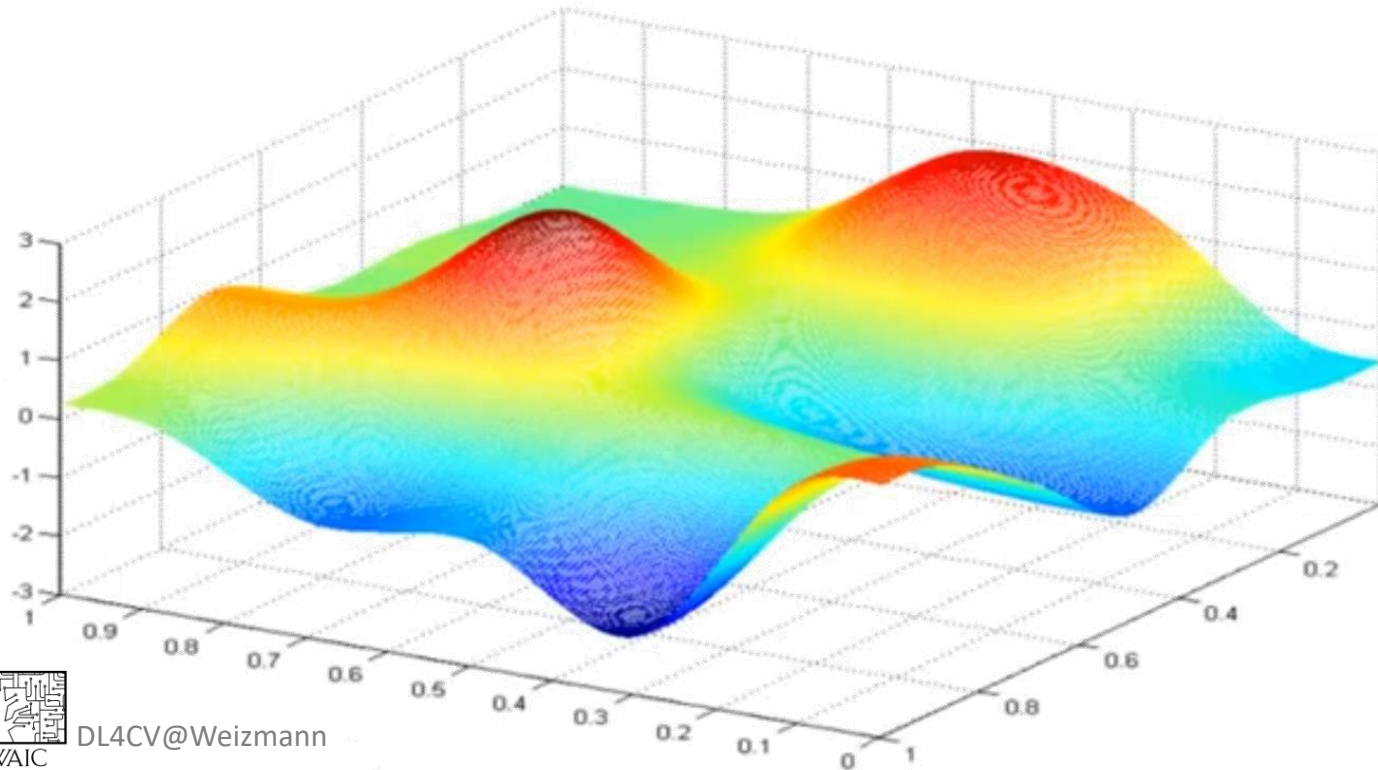


Data space: $h_{\boldsymbol{\theta}}(x)$

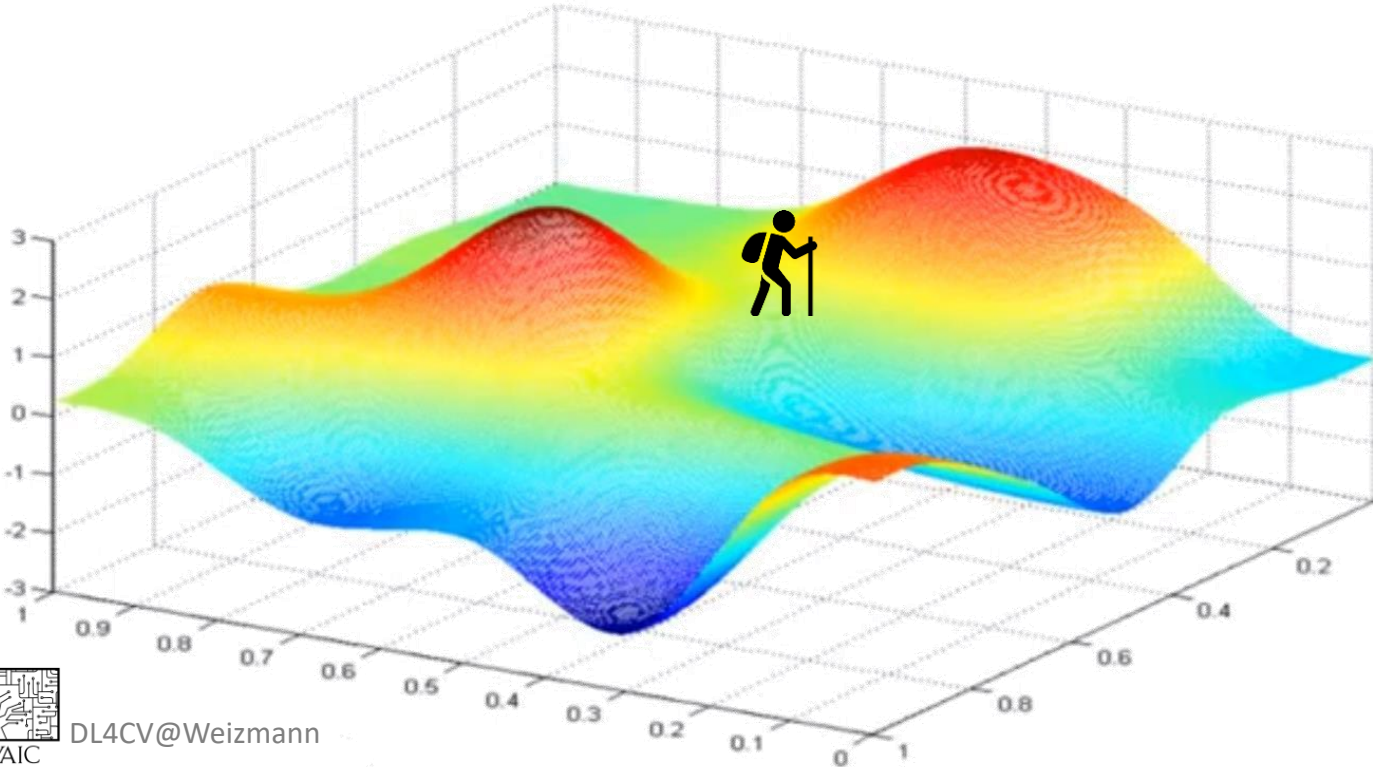
Calculus reminder: Directional derivative



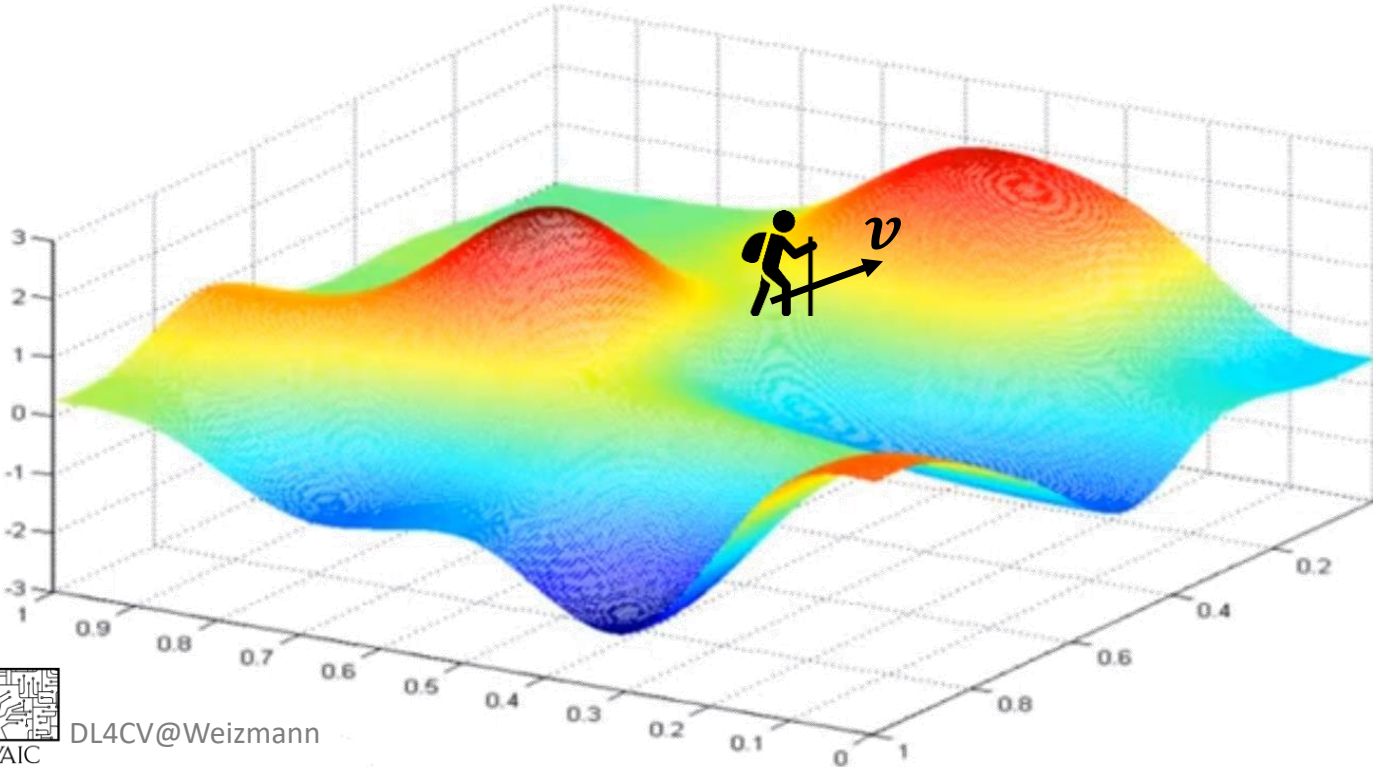
Calculus reminder: Directional derivative



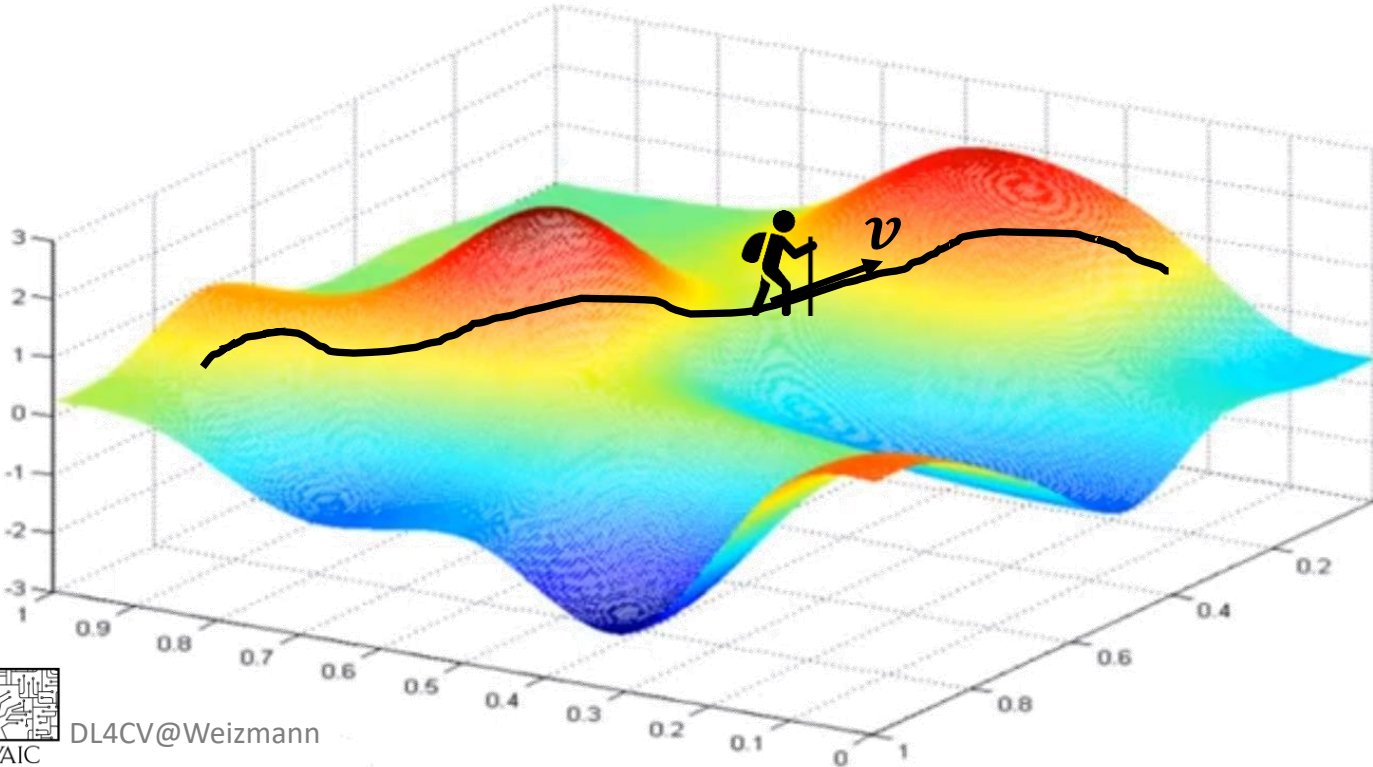
Calculus reminder: Directional derivative



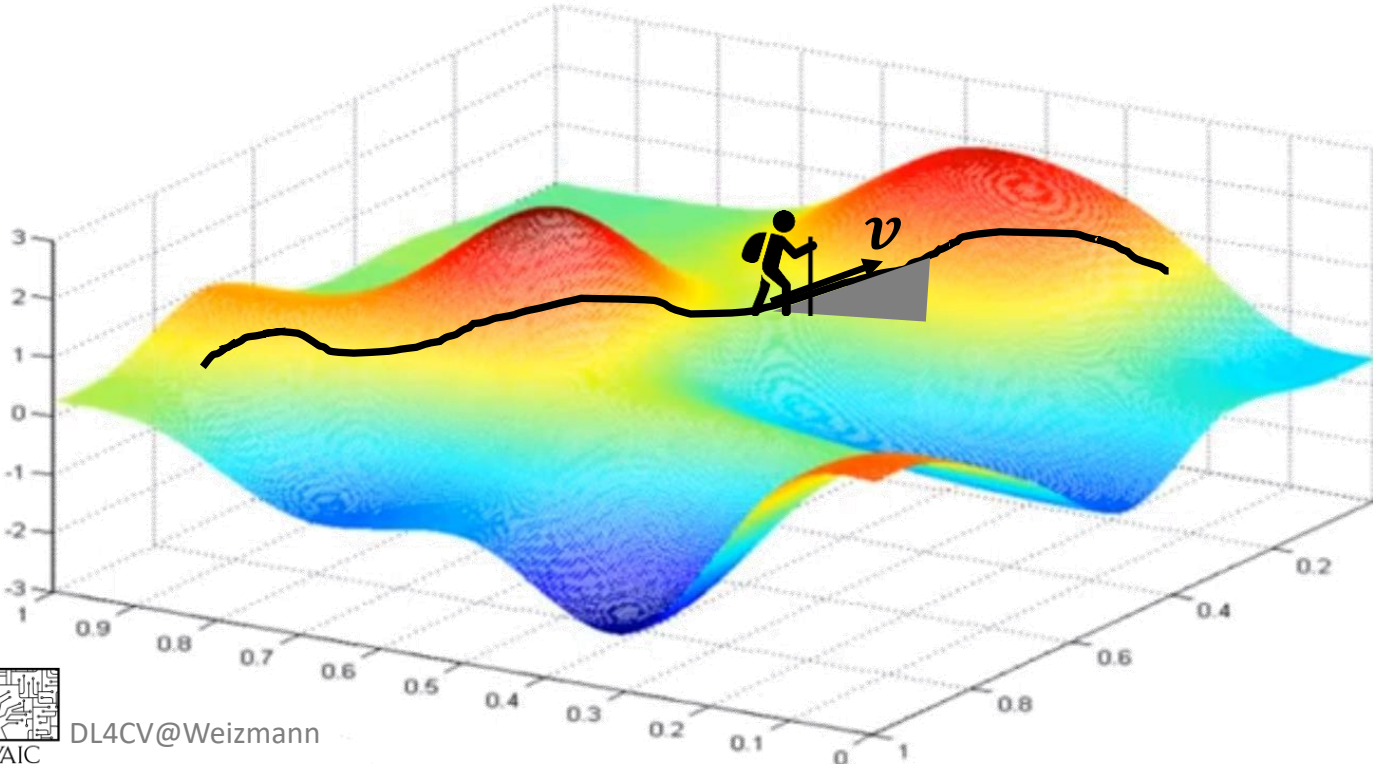
Calculus reminder: Directional derivative



Calculus reminder: Directional derivative

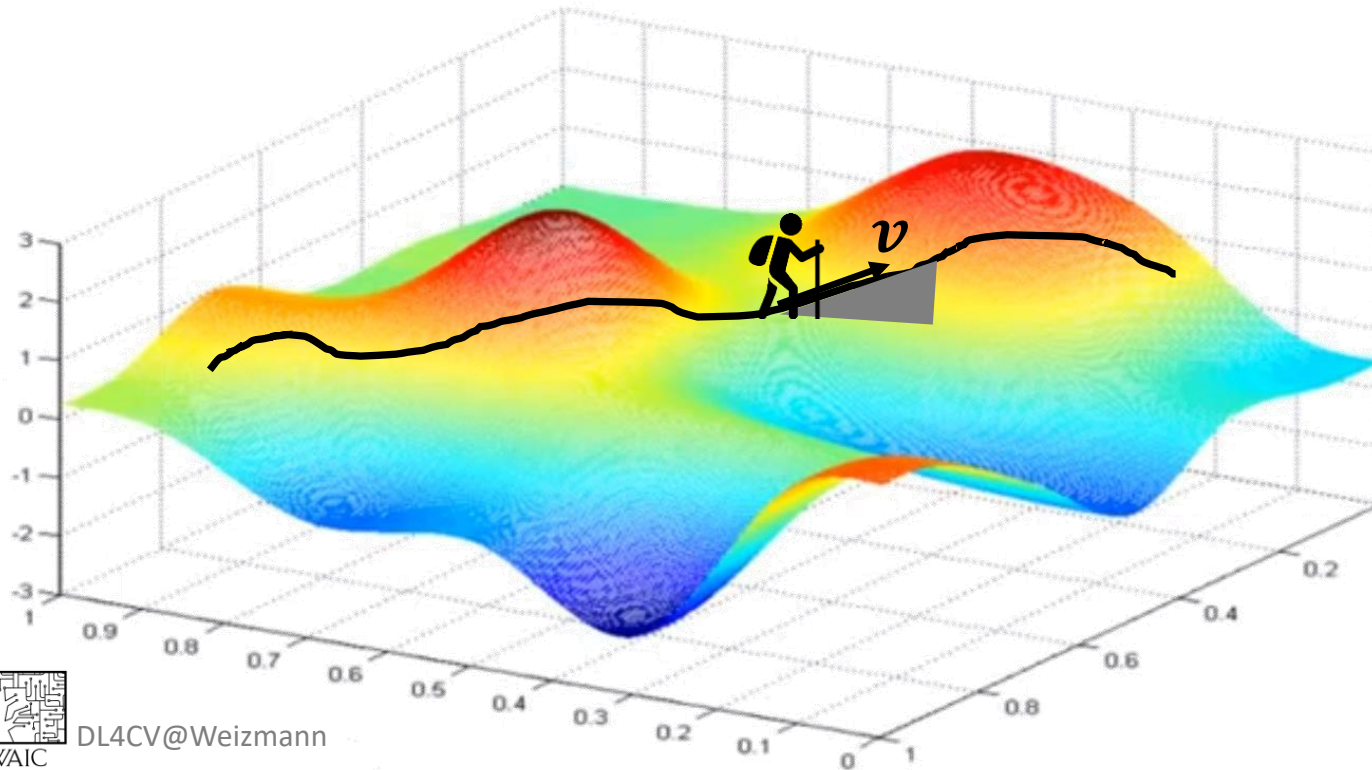


Calculus reminder: Directional derivative



Calculus reminder: Directional derivative

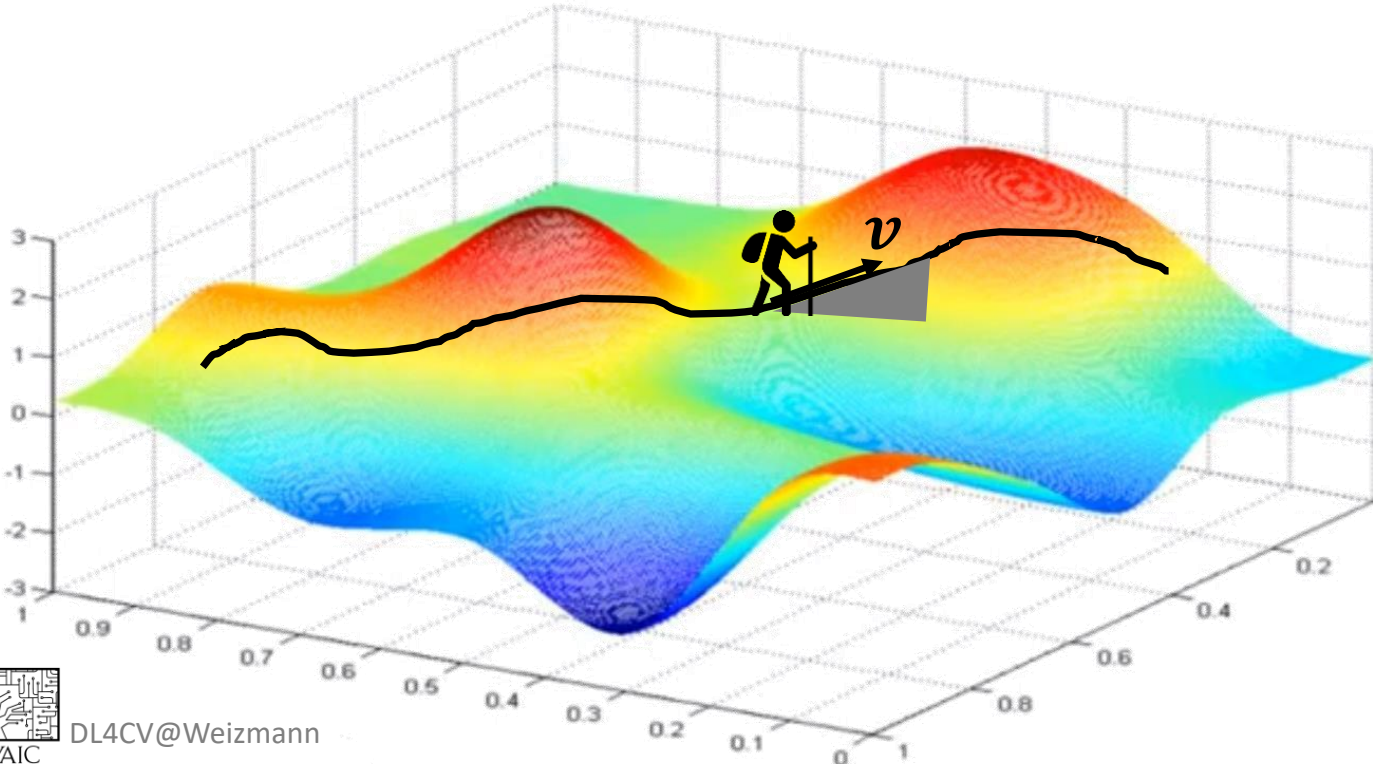
$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|}$$



Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} = \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i}$$

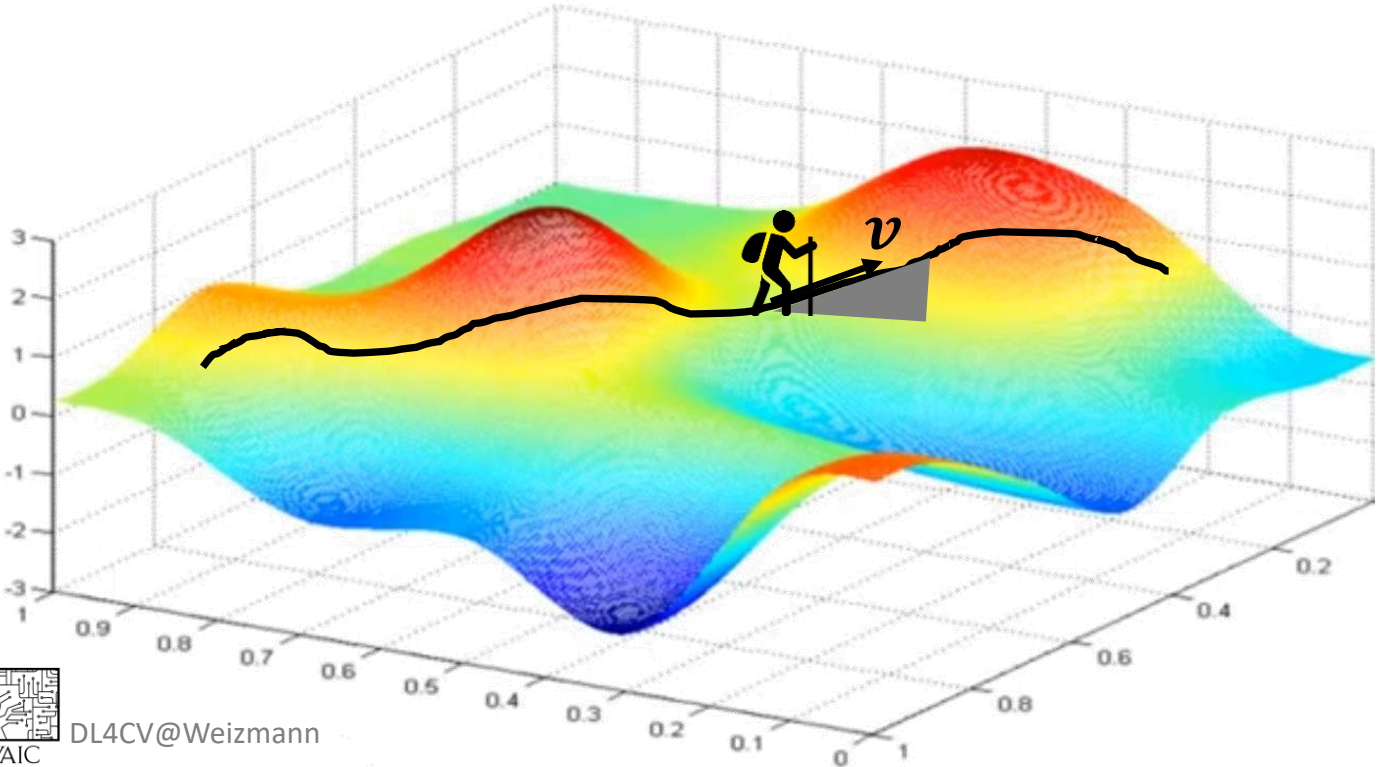
If differentiable



Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} \stackrel{\text{If differentiable}}{=} \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix}$$

If differentiable

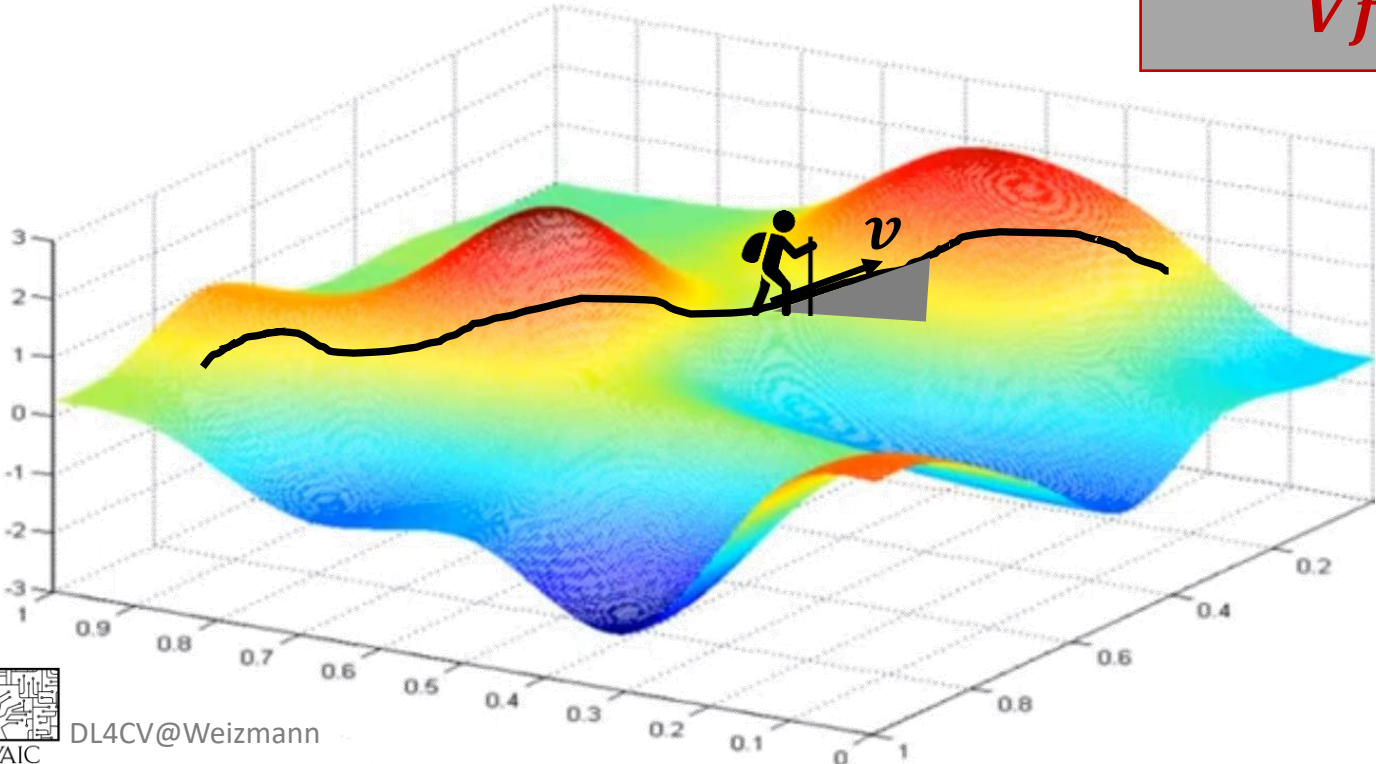


Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} = \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix}$$

If differentiable

Gradient!
 $\vec{\nabla} f$

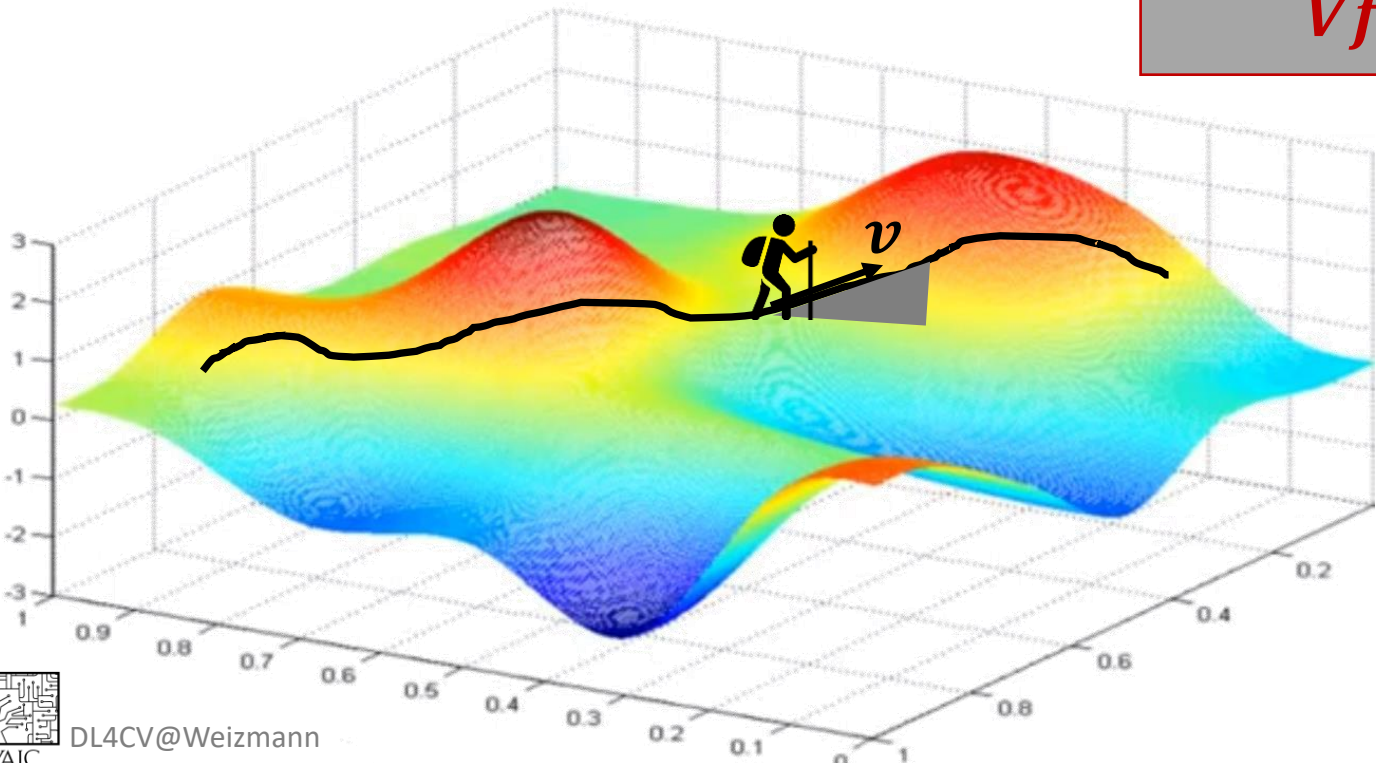


Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} = \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \frac{1}{\|\mathbf{v}\|} \langle \mathbf{v}, \nabla f \rangle$$

If differentiable

Gradient!
 $\vec{\nabla} f$

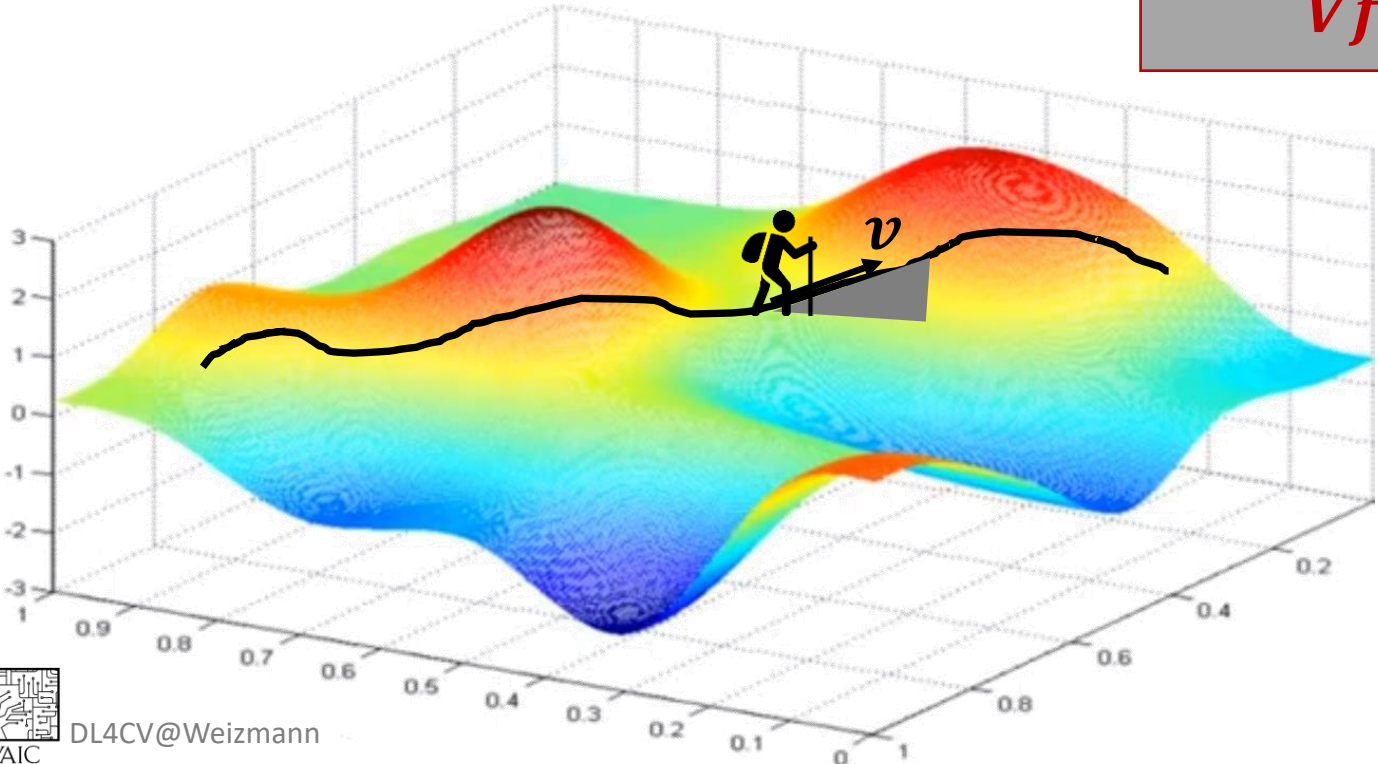


Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} = \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \frac{1}{\|\mathbf{v}\|} \langle \mathbf{v}, \nabla f \rangle$$

If differentiable

Gradient!
 $\vec{\nabla} f$



Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} \stackrel{\text{If differentiable}}{=} \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \frac{1}{\|\mathbf{v}\|} \langle \mathbf{v}, \nabla f \rangle$$

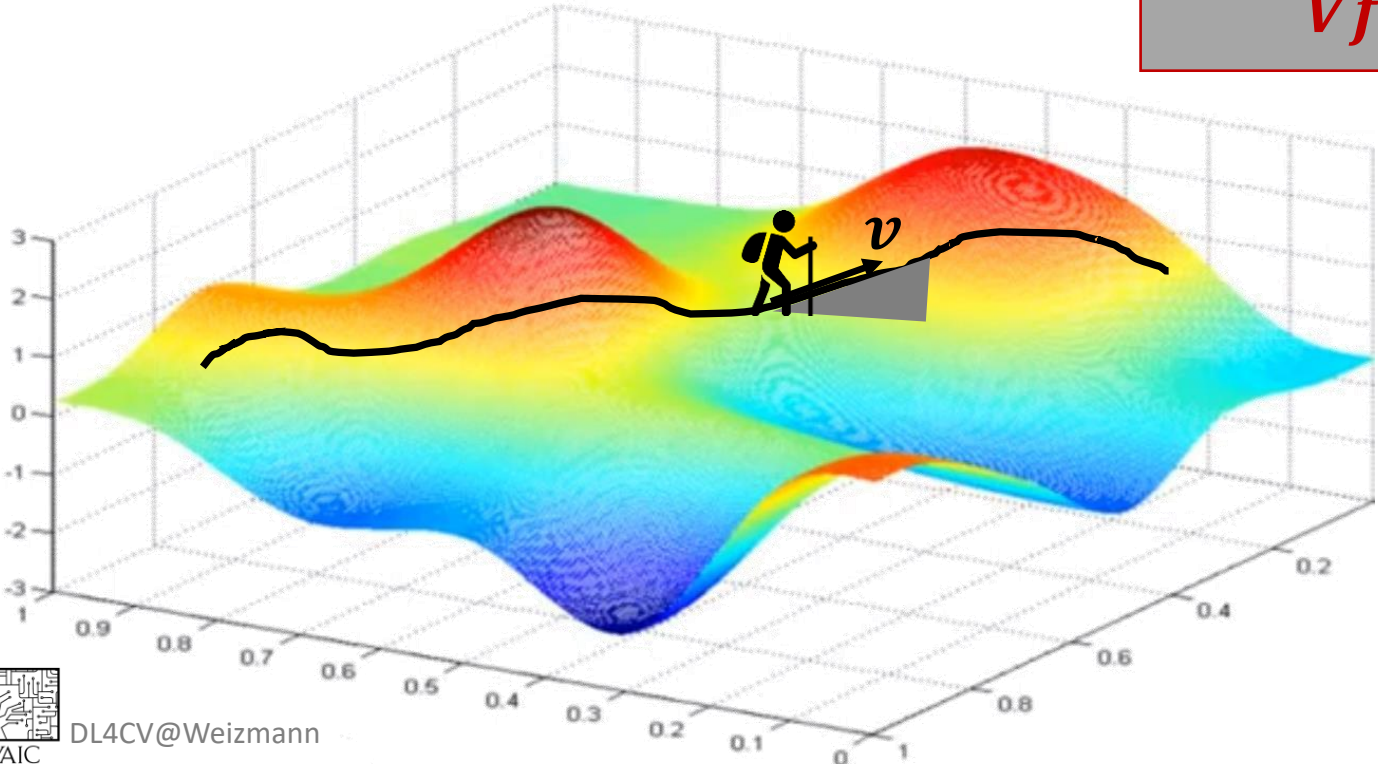
If differentiable

Gradient!
 $\vec{\nabla} f$



According to Cauchy-Schwarz inequality:

- Max value is $\|\nabla f\|$
- Obtained when \mathbf{v} is parallel to ∇f



Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} \stackrel{\text{If differentiable}}{=} \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \frac{1}{\|\mathbf{v}\|} \langle \mathbf{v}, \nabla f \rangle$$

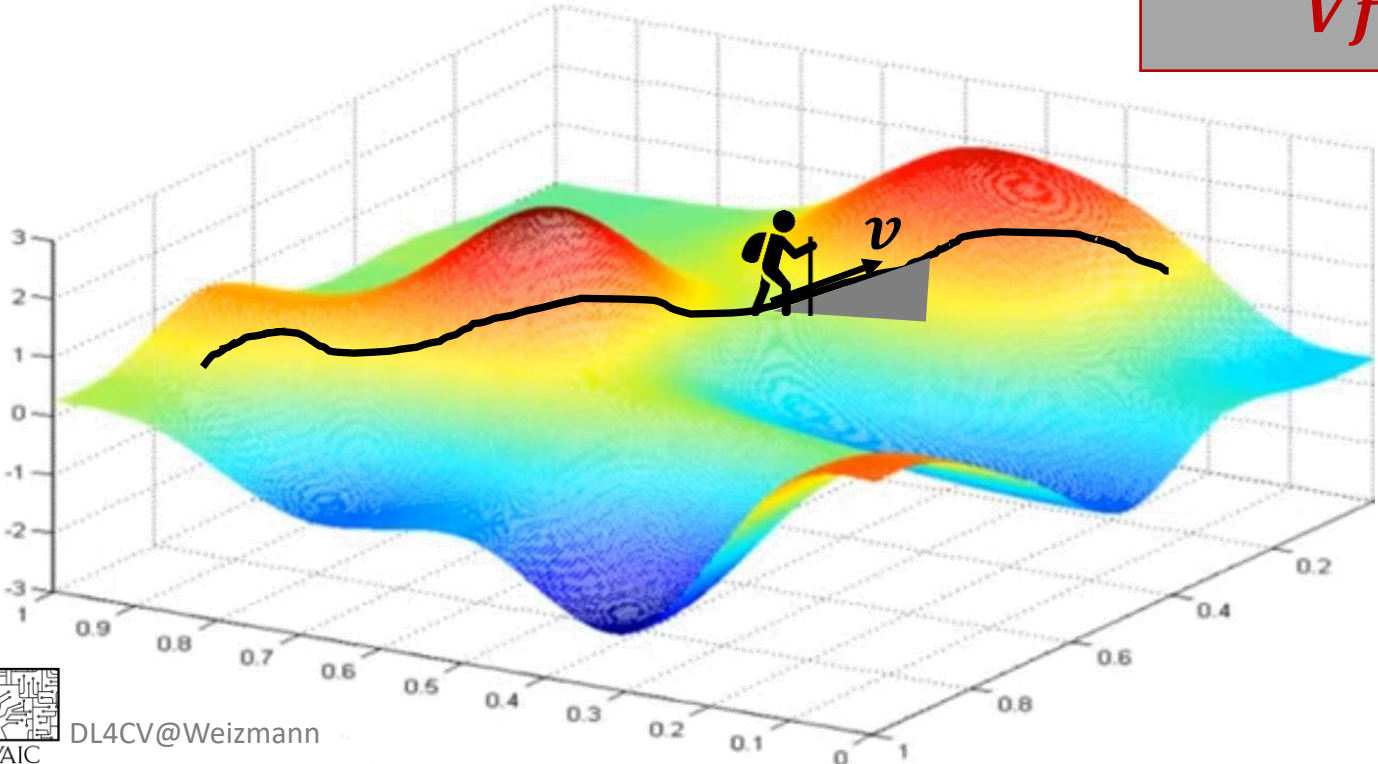
If differentiable

Gradient!
 $\vec{\nabla} f$



According to Cauchy-Schwarz inequality:

- Max value is $\|\nabla f\|$
- Obtained when \mathbf{v} is parallel to ∇f

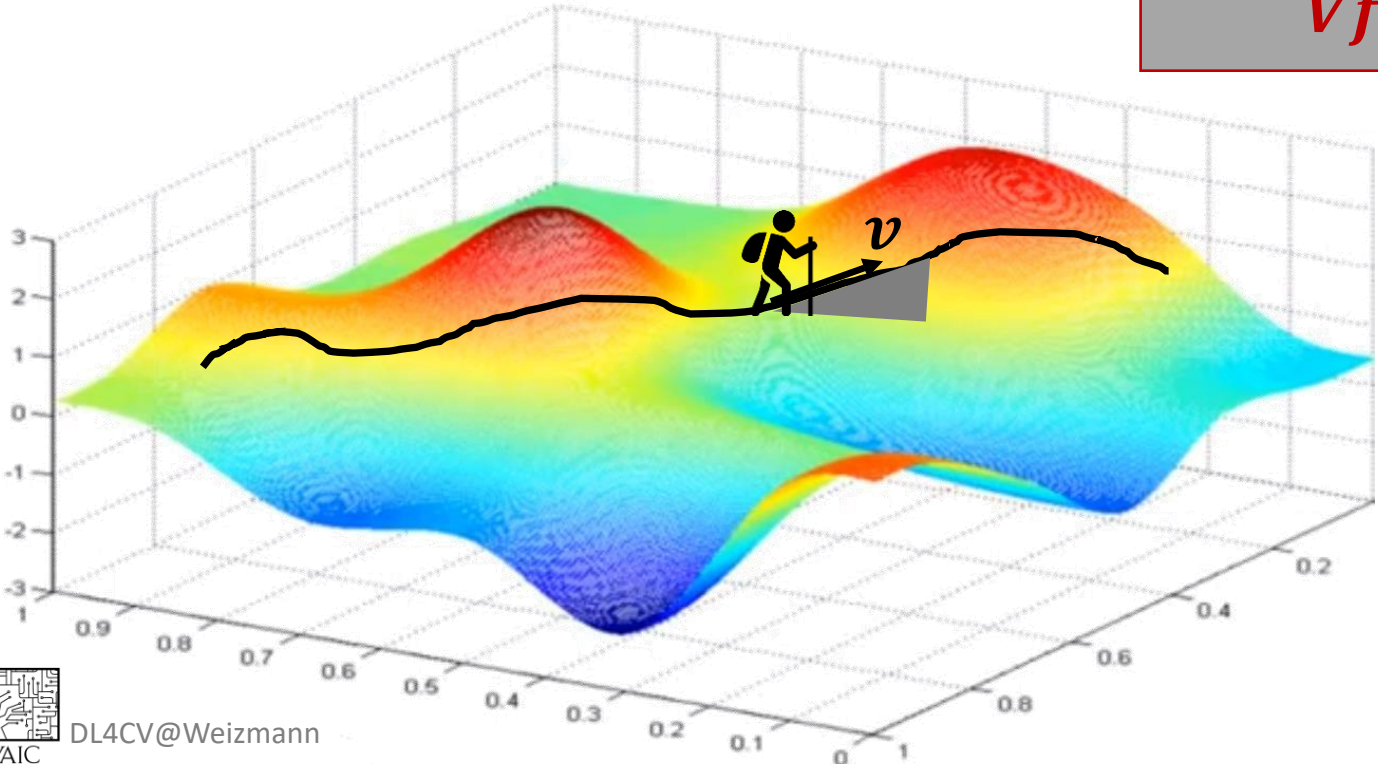


Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} \stackrel{\text{If differentiable}}{=} \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \frac{1}{\|\mathbf{v}\|} \langle \mathbf{v}, \nabla f \rangle$$

If differentiable

Gradient!
 $\vec{\nabla} f$



According to Cauchy-Schwarz inequality:

- Max value is $\|\nabla f\|$
- Obtained when \mathbf{v} is parallel to ∇f

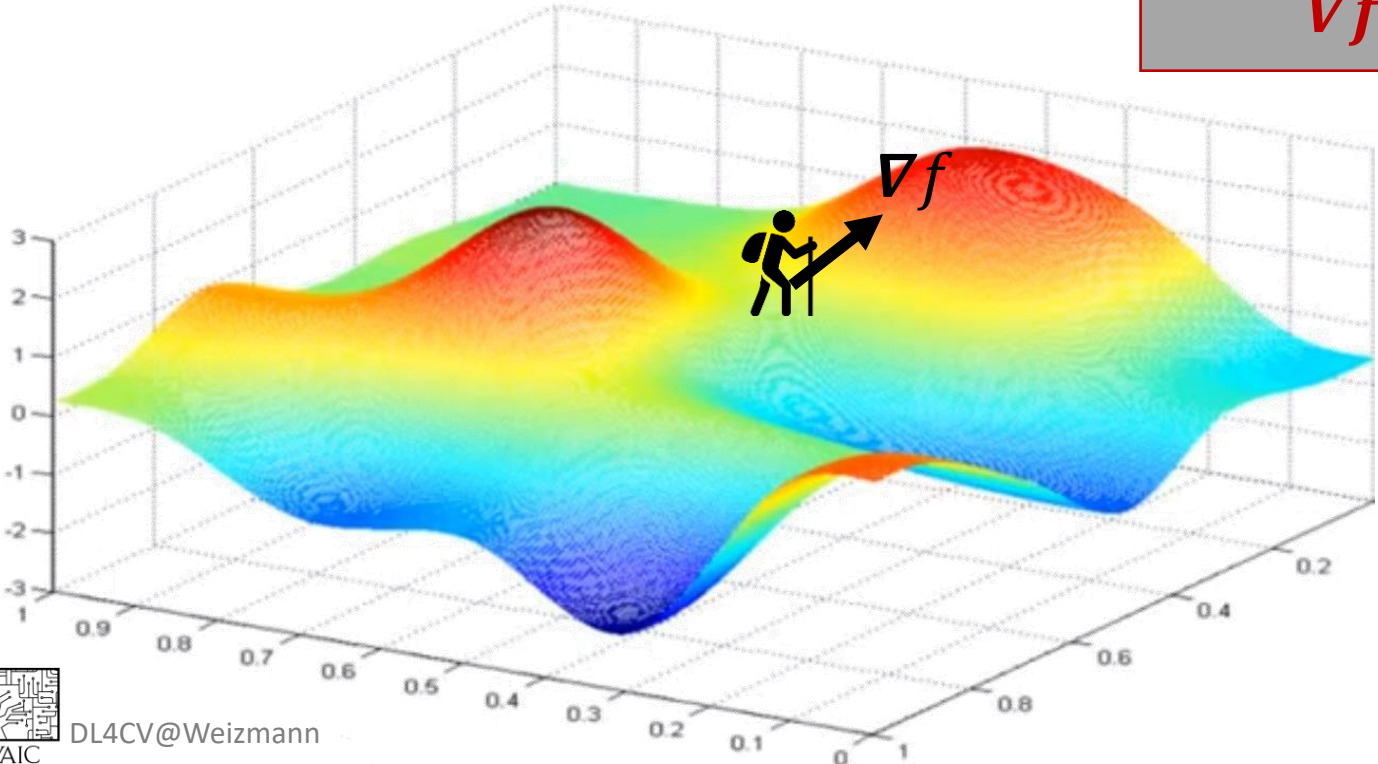
• Gradient directs to steepest ascent.
 • It's size is the max steepness.

Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} = \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \frac{1}{\|\mathbf{v}\|} \langle \mathbf{v}, \nabla f \rangle$$

If differentiable

Gradient!
 $\vec{\nabla} f$



According to Cauchy-Schwarz inequality:

- Max value is $\|\nabla f\|$
- Obtained when \mathbf{v} is parallel to ∇f

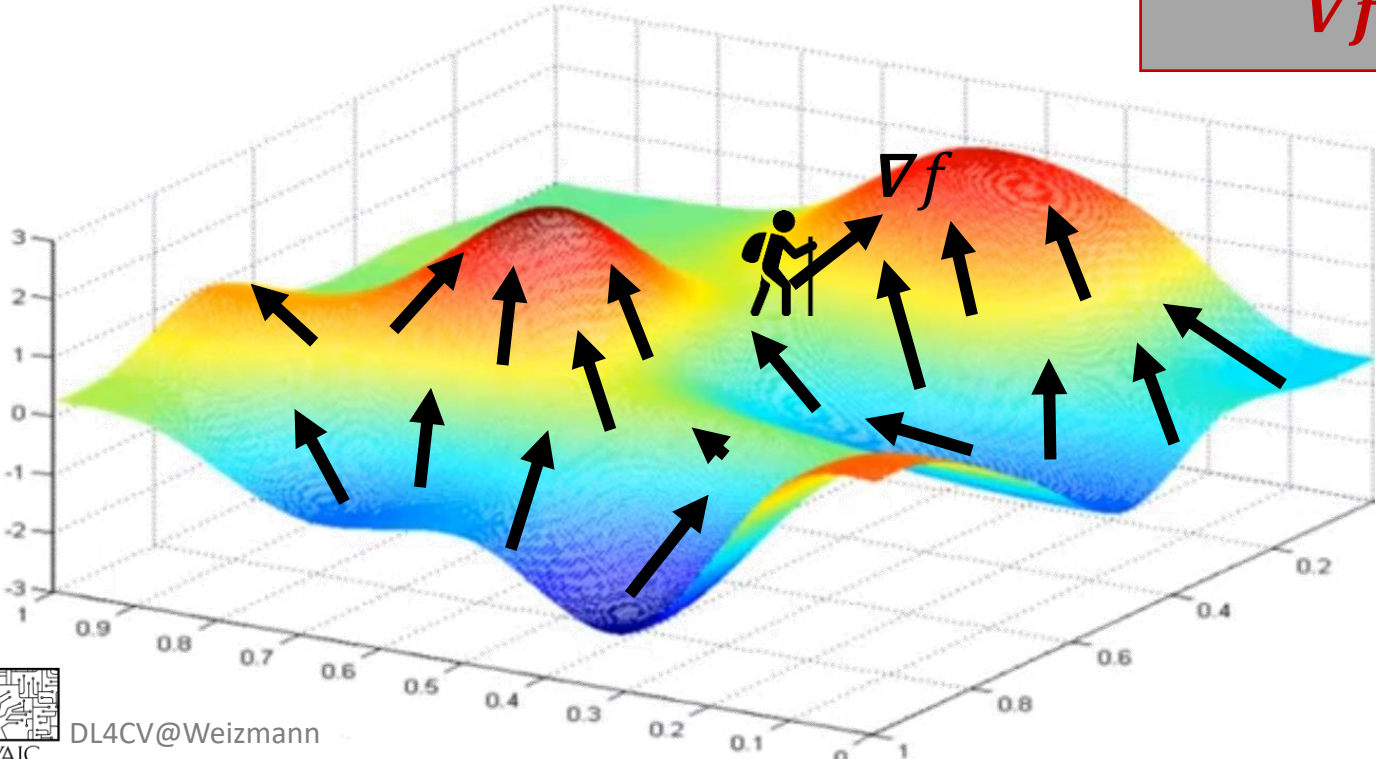
• Gradient directs to steepest ascent.
 • It's size is the max steepness.

Calculus reminder: Directional derivative

$$\lim_{\varepsilon \rightarrow 0} \frac{f(\mathbf{x} + \varepsilon \mathbf{v}) - f(\mathbf{x})}{\varepsilon \|\mathbf{v}\|} \stackrel{\text{If differentiable}}{=} \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \frac{1}{\|\mathbf{v}\|} \langle \mathbf{v}, \nabla f \rangle$$

If differentiable

Gradient!
 $\vec{\nabla} f$



According to Cauchy-Schwarz inequality:

- Max value is $\|\nabla f\|$
- Obtained when \mathbf{v} is parallel to ∇f

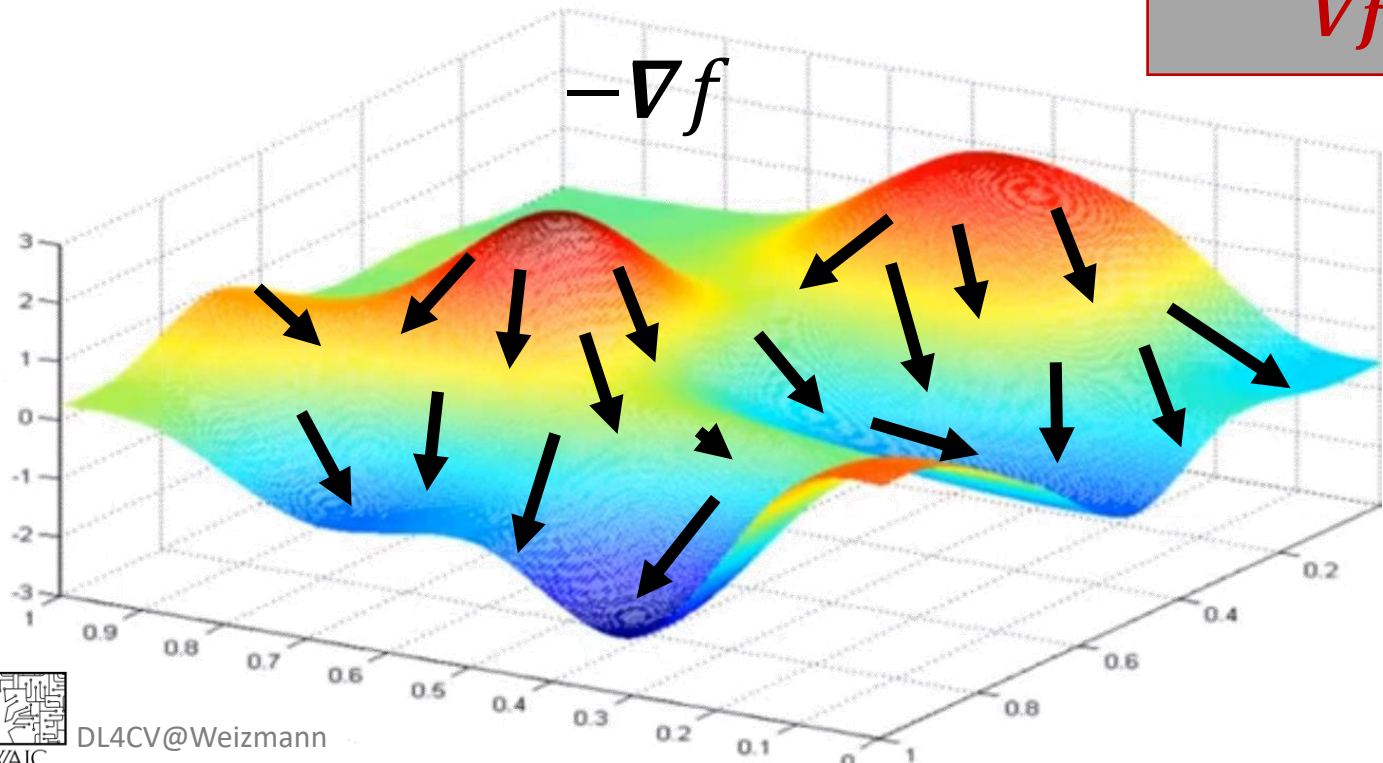
• Gradient directs to steepest ascent.
 • It's size is the max steepness.

Calculus reminder: Directional derivative

$$\lim_{\epsilon \rightarrow 0} \frac{f(\mathbf{x} + \epsilon \mathbf{v}) - f(\mathbf{x})}{\epsilon \|\mathbf{v}\|} \stackrel{\text{If differentiable}}{=} \frac{1}{\|\mathbf{v}\|} \sum v_i \frac{\partial f}{\partial x_i} = \frac{\mathbf{v}^T}{\|\mathbf{v}\|} \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{pmatrix} = \frac{1}{\|\mathbf{v}\|} \langle \mathbf{v}, \nabla f \rangle$$

If differentiable

Gradient!
 $\vec{\nabla} f$



According to Cauchy-Schwarz inequality:

- Max value is $\|\nabla f\|$
- Obtained when \mathbf{v} is parallel to ∇f

• Gradient directs to steepest ascent.
 • It's size is the max steepness.

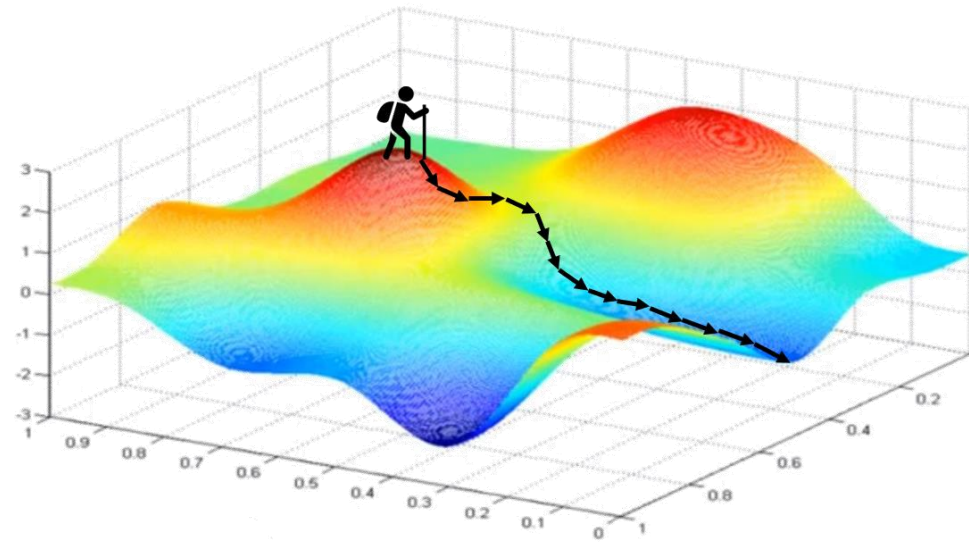
Gradient descent

$$\nabla \mathcal{L}(\theta_0, \theta_1 \dots \theta_N) = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \theta_0} \\ \frac{\partial \mathcal{L}}{\partial \theta_1} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \theta_N} \end{pmatrix}$$



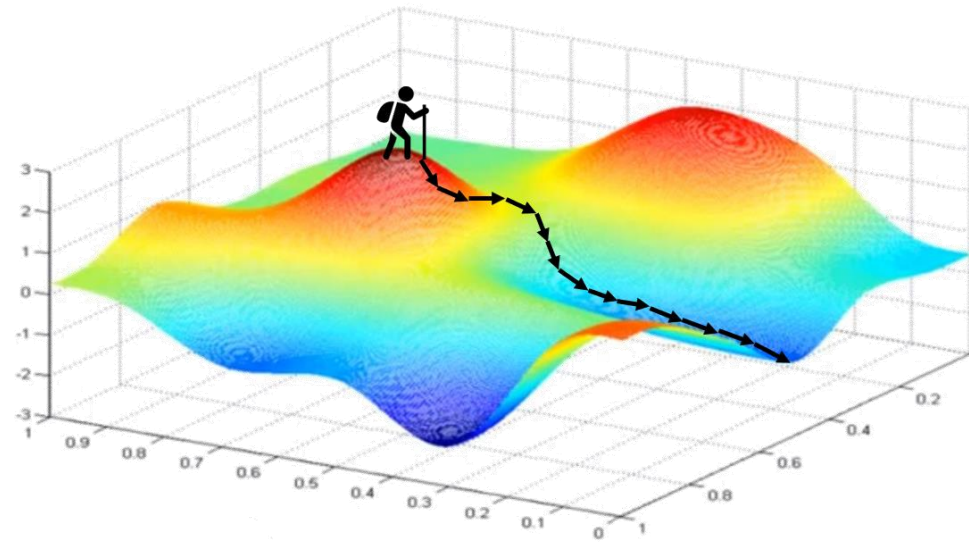
Gradient descent

$$\nabla \mathcal{L}(\theta_0, \theta_1 \dots \theta_N) = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \theta_0} \\ \frac{\partial \mathcal{L}}{\partial \theta_1} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \theta_N} \end{pmatrix}$$



Gradient descent

$$\nabla \mathcal{L}(\theta_0, \theta_1 \dots \theta_N) = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \theta_0} \\ \frac{\partial \mathcal{L}}{\partial \theta_1} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \theta_N} \end{pmatrix}$$



Augustin
Louis
Cauchy

1. Initialize $\theta \sim \text{Random}$
2. Repeat until convergence:
{
$$\theta := \theta - \alpha \nabla \mathcal{L}(\theta; S)$$

}

α : Learning rate

Gradient descent



Full batch Gradient Descent



Gradient descent



Full batch Gradient Descent

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \alpha \nabla \mathcal{L}(\boldsymbol{\theta}; S)$$

Gradient descent



Full batch Gradient Descent

$$\theta := \theta - \alpha \nabla \mathcal{L}(\theta; S)$$

Gradient descent



Full batch Gradient Descent



Stochastic Gradient Descent

$$\theta := \theta - \alpha \nabla \mathcal{L}(\theta; S)$$

Gradient descent



Full batch Gradient Descent

$$\theta := \theta - \alpha \nabla \mathcal{L}(\theta; S)$$



Stochastic Gradient Descent

$$\theta := \theta - \alpha \nabla \mathcal{L}(\theta; x_i, y_i)$$

$$i \sim U[1, M]$$

Gradient descent



Full batch Gradient Descent

$$\theta := \theta - \alpha \nabla \mathcal{L}(\theta; S)$$



Stochastic Gradient Descent

$$\theta := \theta - \alpha \nabla \mathcal{L}(\theta; x_i, y_i)$$

$$i \sim U[1, M]$$



Stochastic Mini-Batch
Gradient Descent

Gradient descent



Full batch Gradient Descent

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \alpha \nabla \mathcal{L}(\boldsymbol{\theta}; \mathbf{S})$$



Stochastic Gradient Descent

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \alpha \nabla \mathcal{L}(\boldsymbol{\theta}; x_i, y_i)$$

$$i \sim U[1, M]$$



Stochastic Mini-Batch
Gradient Descent

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \alpha \nabla \mathcal{L}(\boldsymbol{\theta}; \mathbf{B})$$

$$\mathbf{B} = \{(x_i, y_i) ; i \sim U^m[1, M]\}$$

Gradient descent



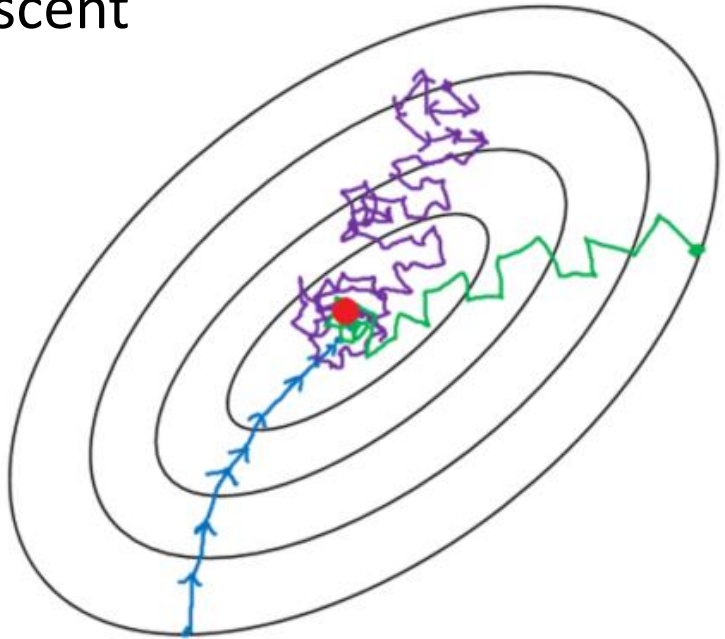
Full batch Gradient Descent



Stochastic Gradient Descent



Stochastic Mini-Batch Gradient Descent



- Batch gradient descent
- Mini-batch gradient Descent
- Stochastic gradient descent

Figure by Z² Little on Medium

Gradient descent for Linear Regression

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i)^2$$

Gradient descent for Linear Regression

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i)^2$$

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L} = \frac{1}{m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i) \mathbf{x}_i$$

Gradient descent for Linear Regression

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i)^2$$

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L} = \frac{1}{m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i) \mathbf{x}_i = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})_i$$

Gradient descent for Linear Regression

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i)^2$$

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L} = \frac{1}{m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i) \mathbf{x}_i = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})_i = \mathbf{X}^T (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$$

Gradient descent for Linear Regression

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i)^2$$

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L} = \frac{1}{m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i) \mathbf{x}_i = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})_i = \mathbf{X}^T \overbrace{(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})}$$

Gradient descent for Linear Regression

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i)^2$$

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L} = \frac{1}{m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i) \mathbf{x}_i = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})_i = \mathbf{X}^T \underbrace{(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})}_e$$

e

Gradient descent for Linear Regression

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i)^2$$

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L} = \frac{1}{m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i) \mathbf{x}_i = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})_i = \mathbf{X}^T \underbrace{(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})}_{\mathbf{e}}$$

\mathbf{e}

Repeat until convergence:

{

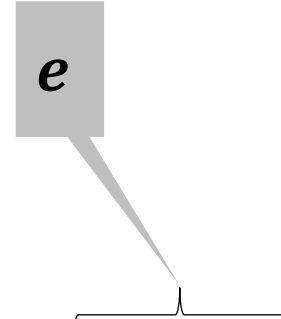
$$\boldsymbol{\theta} := \boldsymbol{\theta} - \frac{\alpha}{m} \mathbf{X}^T \mathbf{e}$$

}

Gradient descent for Linear Regression

$$\mathcal{L} = \frac{1}{2m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i)^2$$

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L} = \frac{1}{m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}_i - y_i) \mathbf{x}_i = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})_i = \mathbf{X}^T \underbrace{(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})}_{\mathbf{e}}$$



Repeat until convergence:

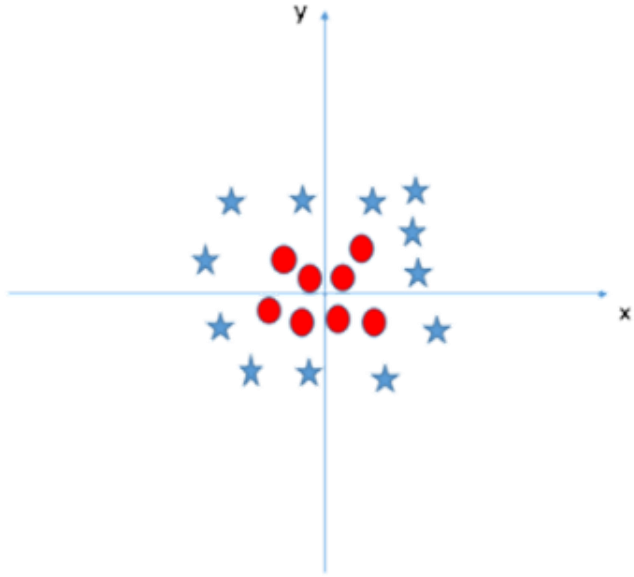
{

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \frac{\alpha}{m} \mathbf{X}^T \mathbf{e}$$

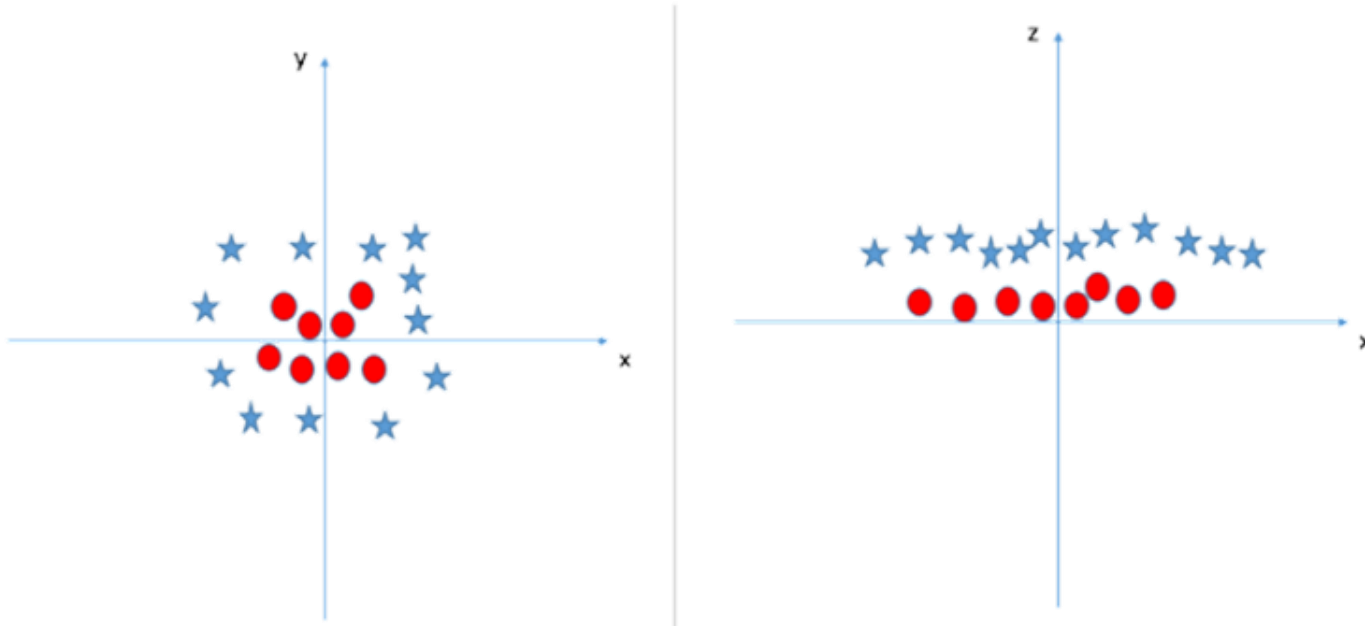
}

Q: Find the relation between convergence and Normal Equations

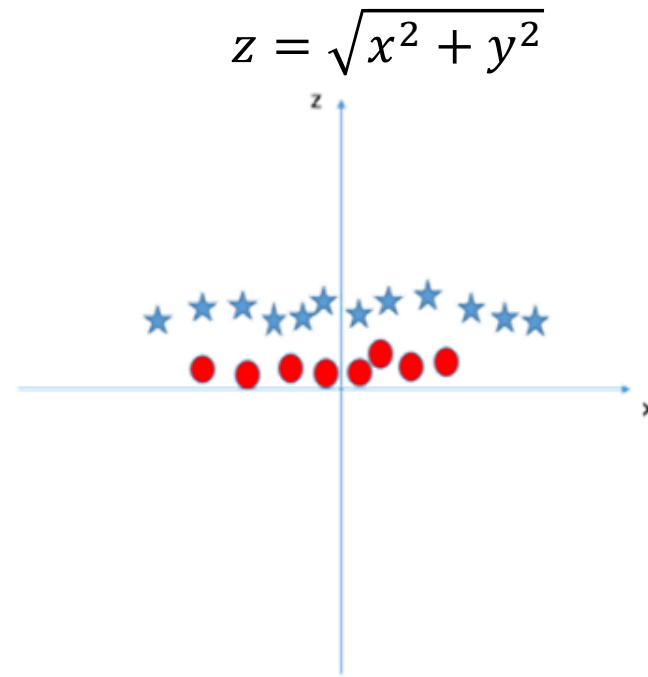
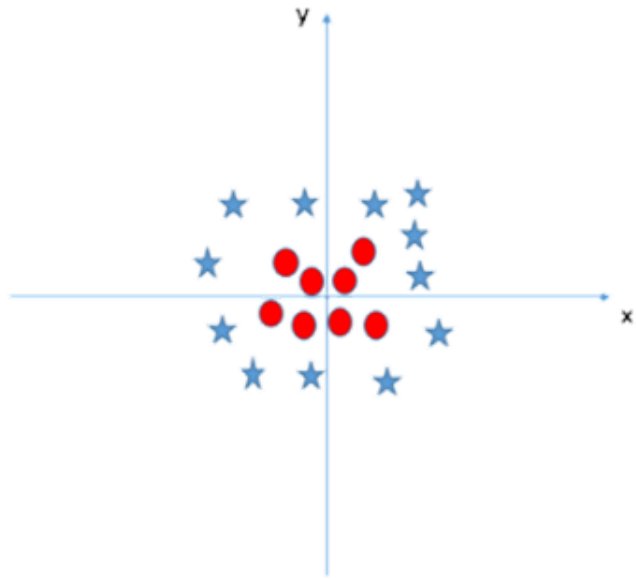
Feature transform



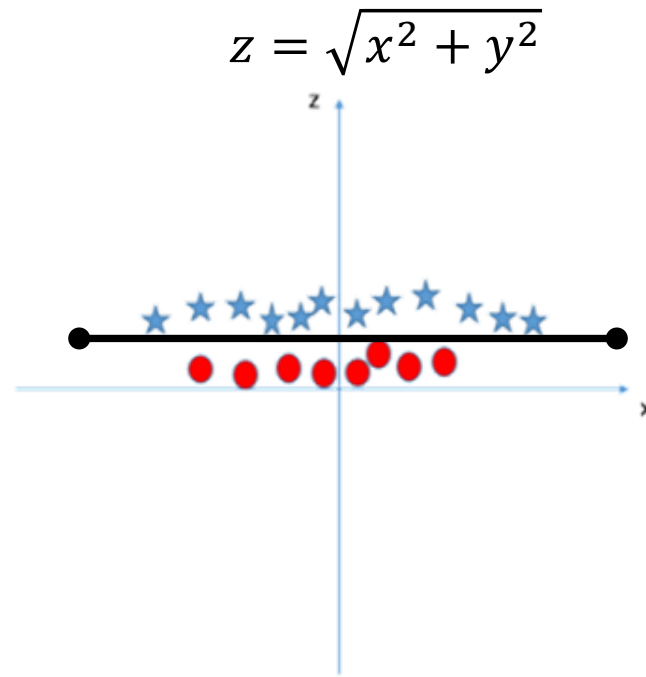
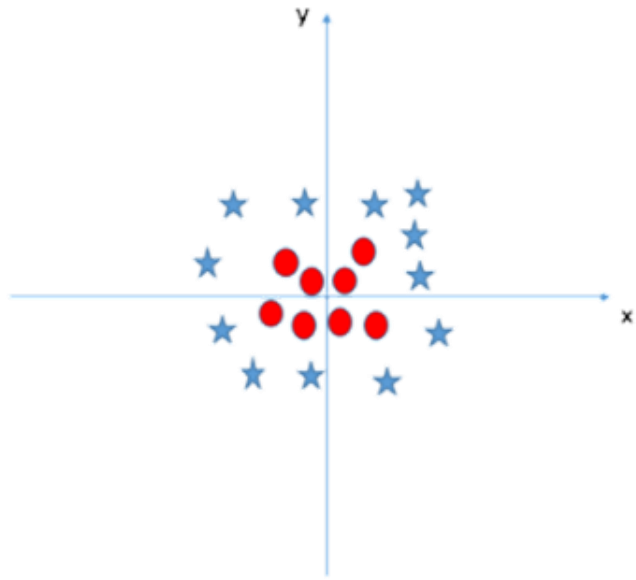
Feature transform



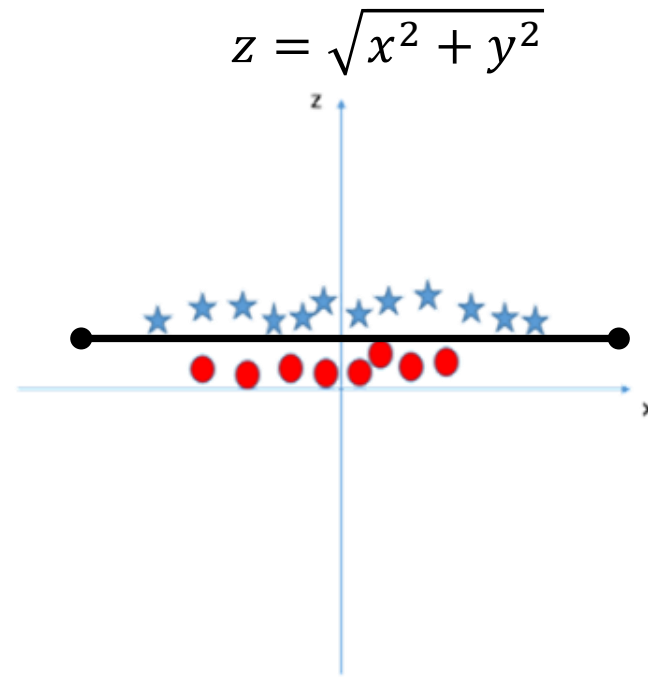
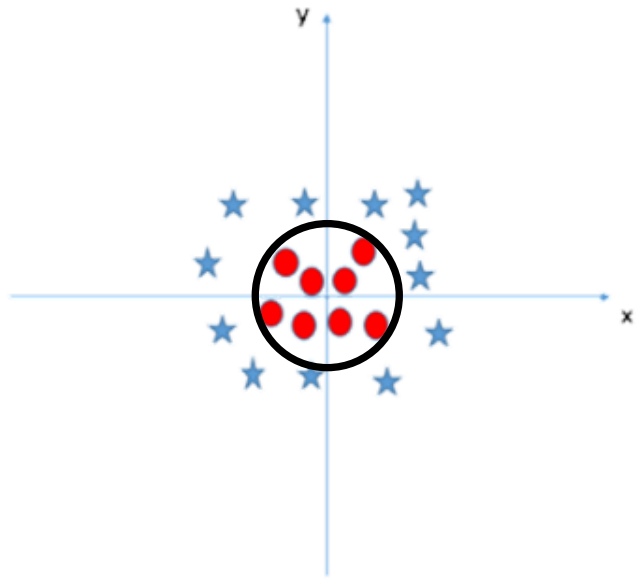
Feature transform



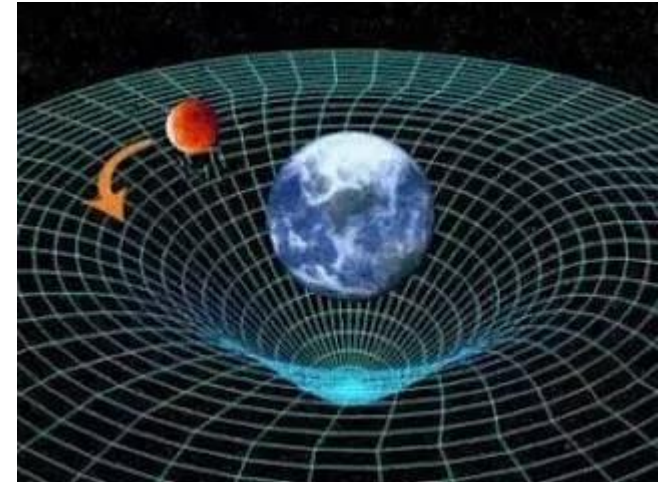
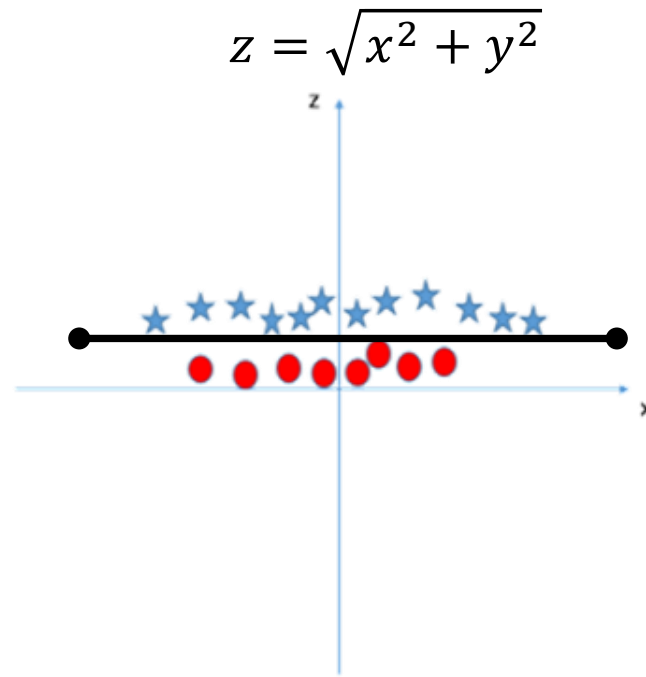
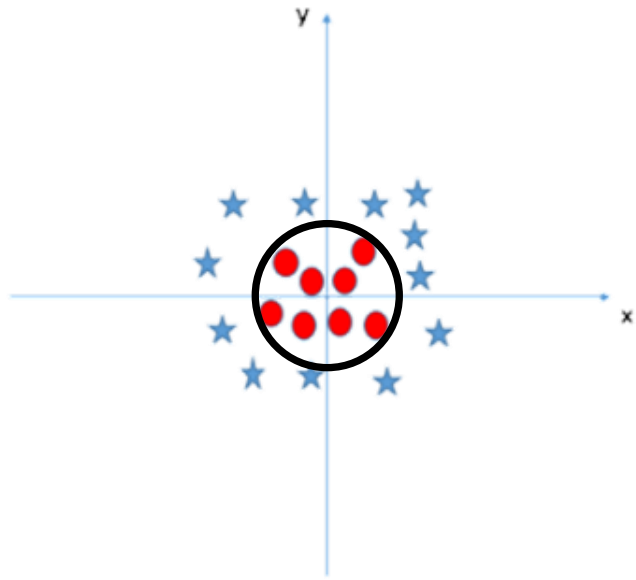
Feature transform



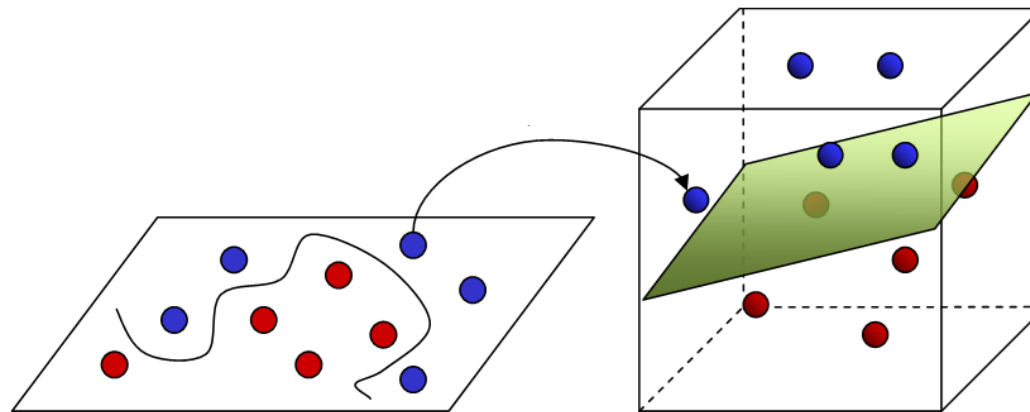
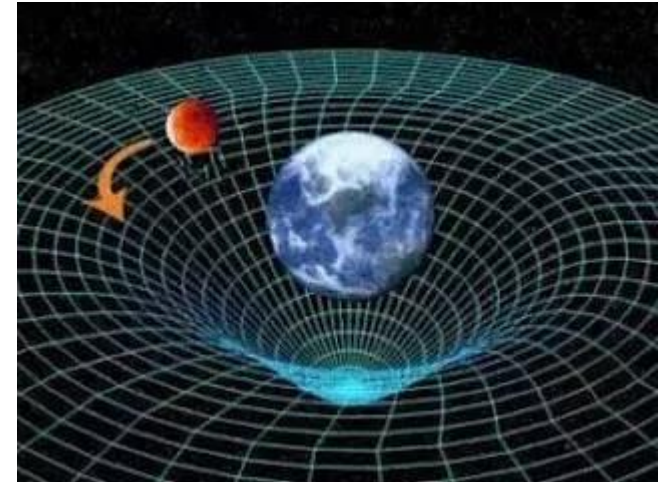
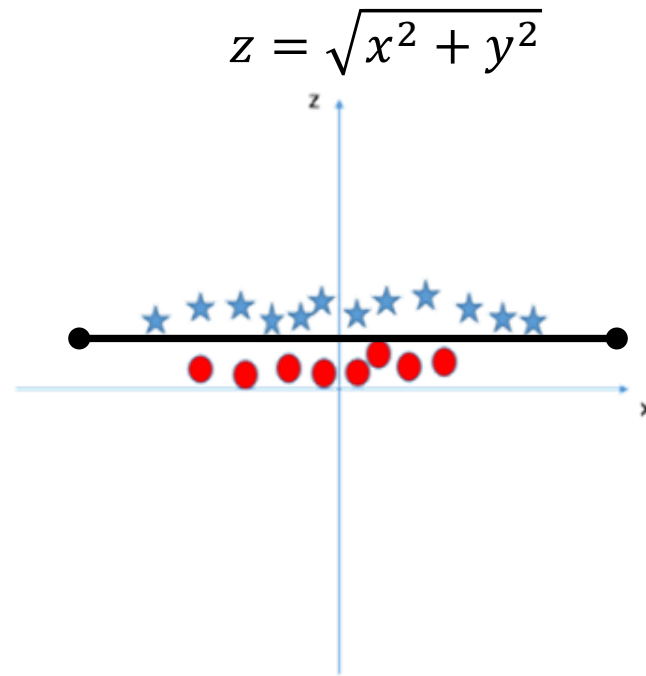
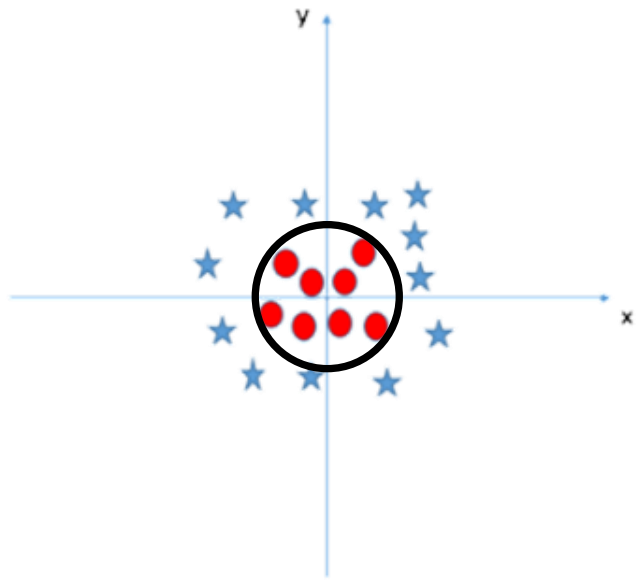
Feature transform



Feature transform



Feature transform



Input Space

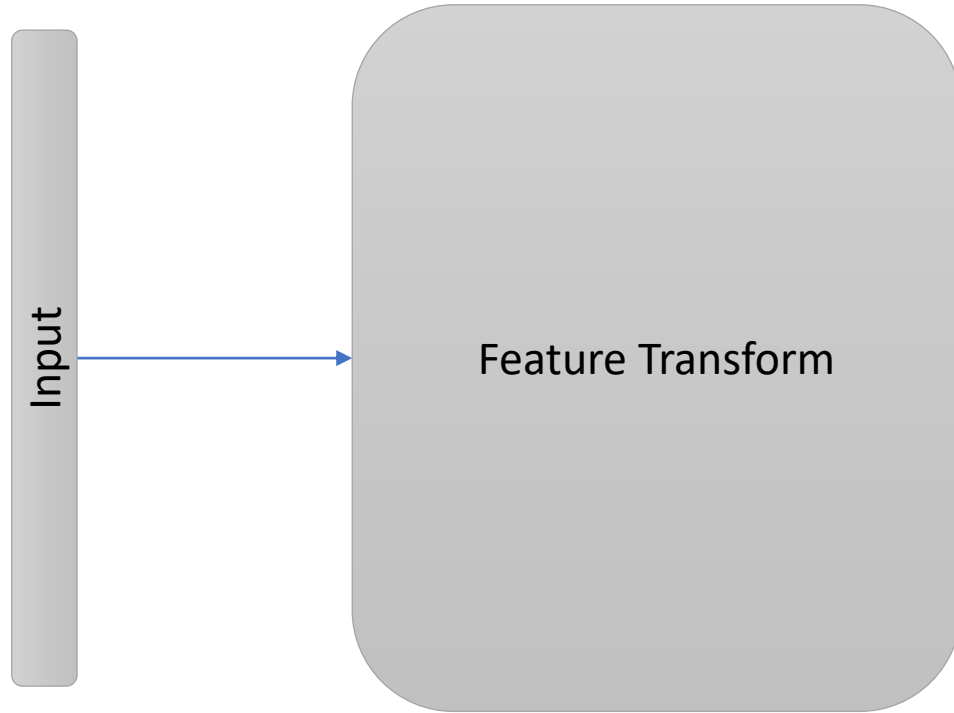
Feature Space

Feature transform

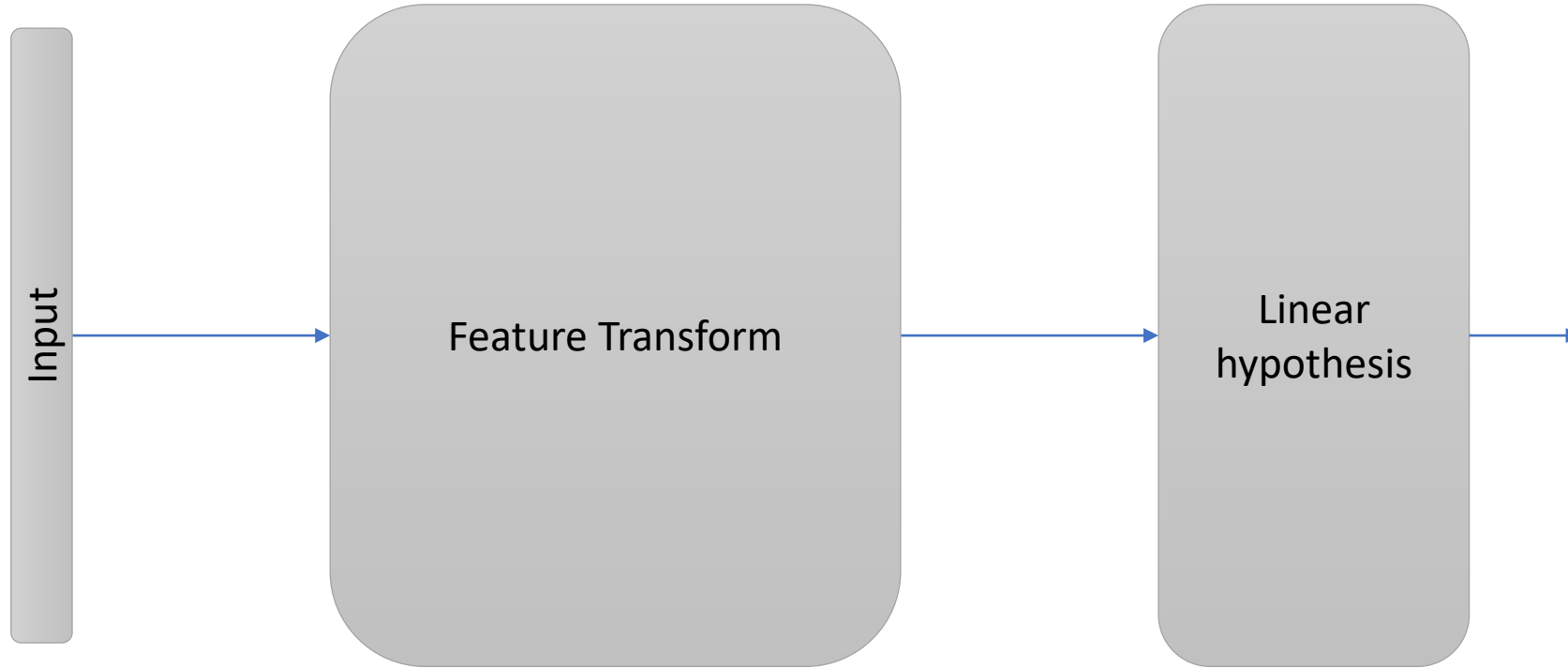
Input



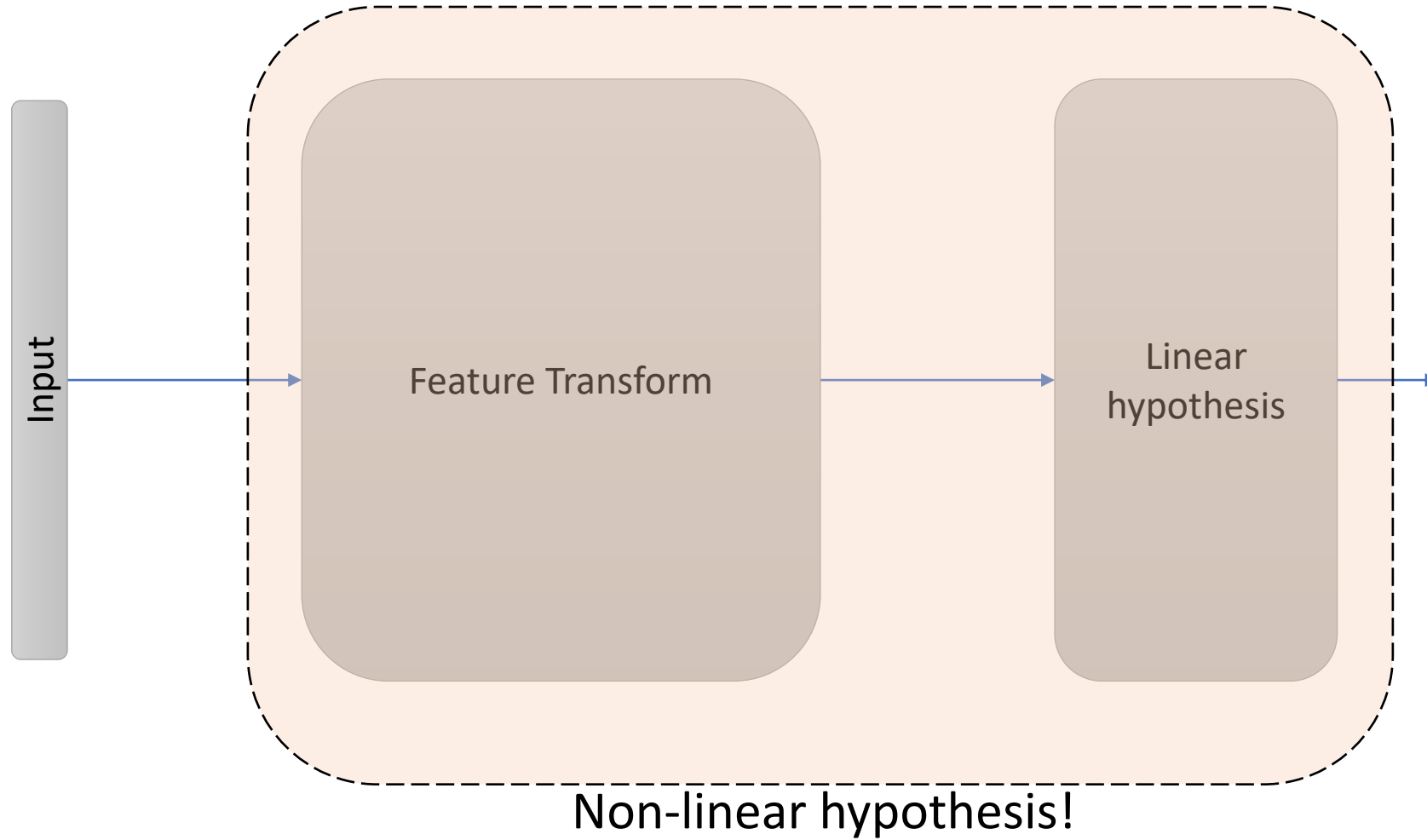
Feature transform



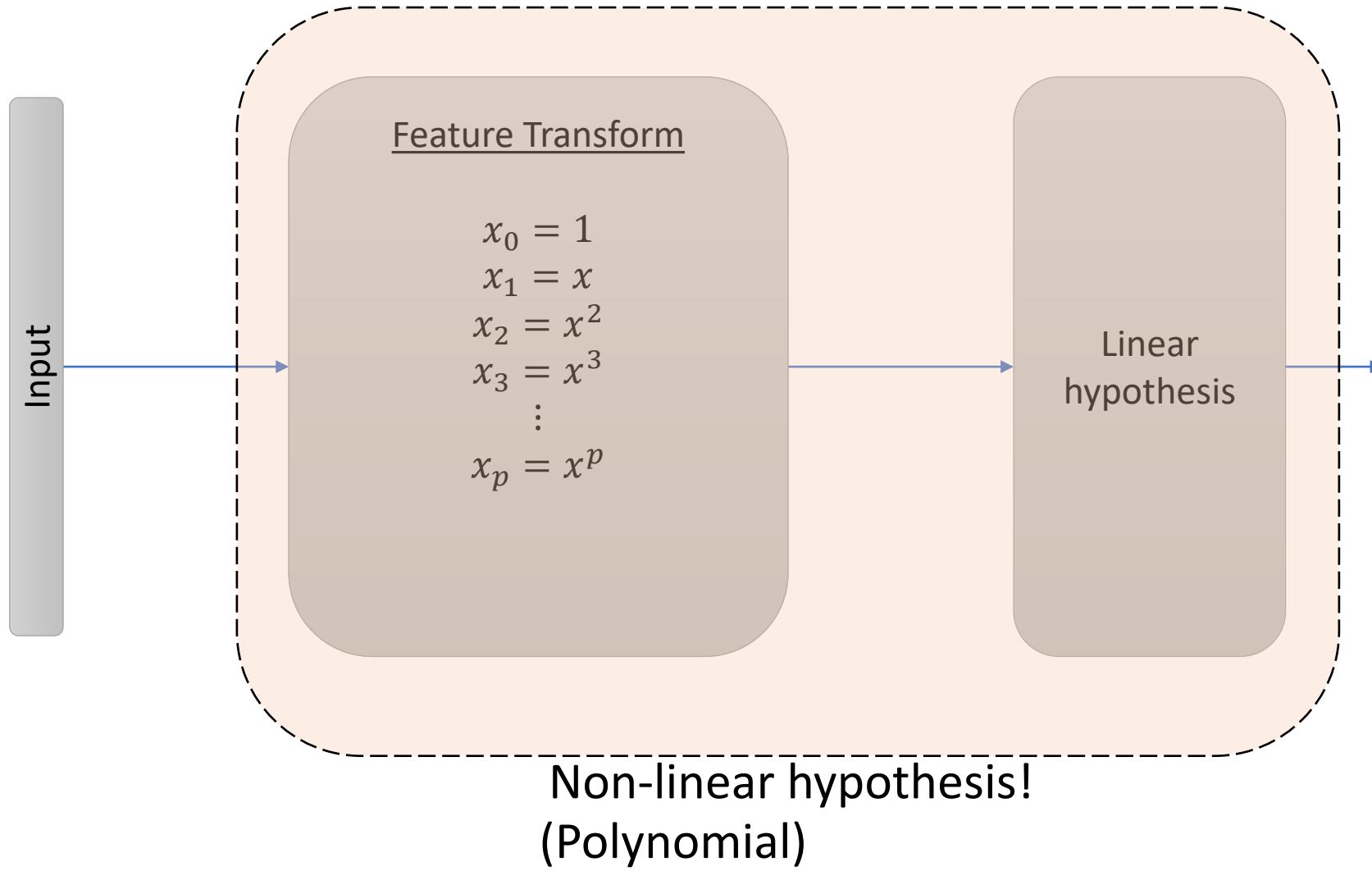
Feature transform



Feature transform




Feature transform



Error decomposition

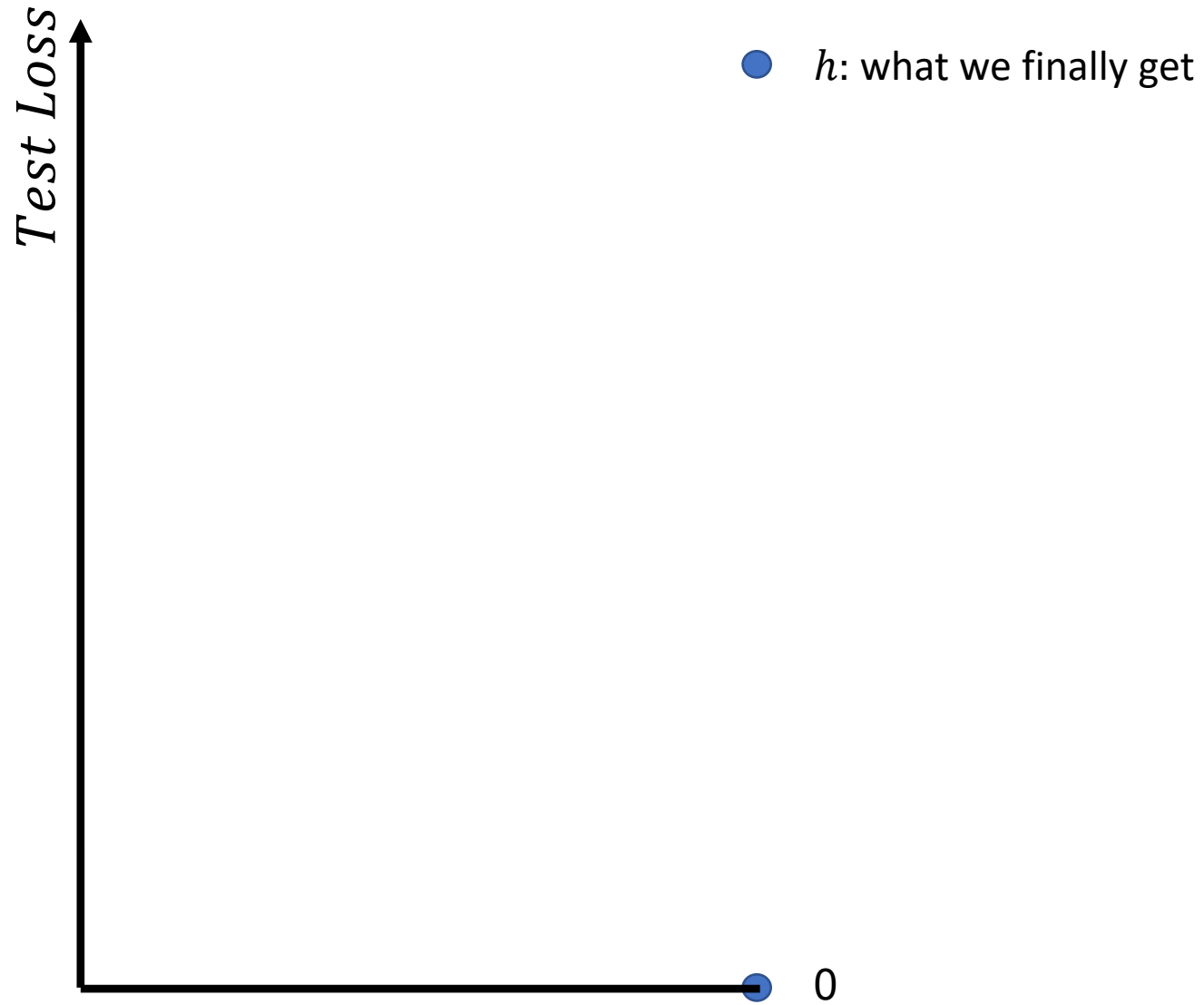
Test Loss



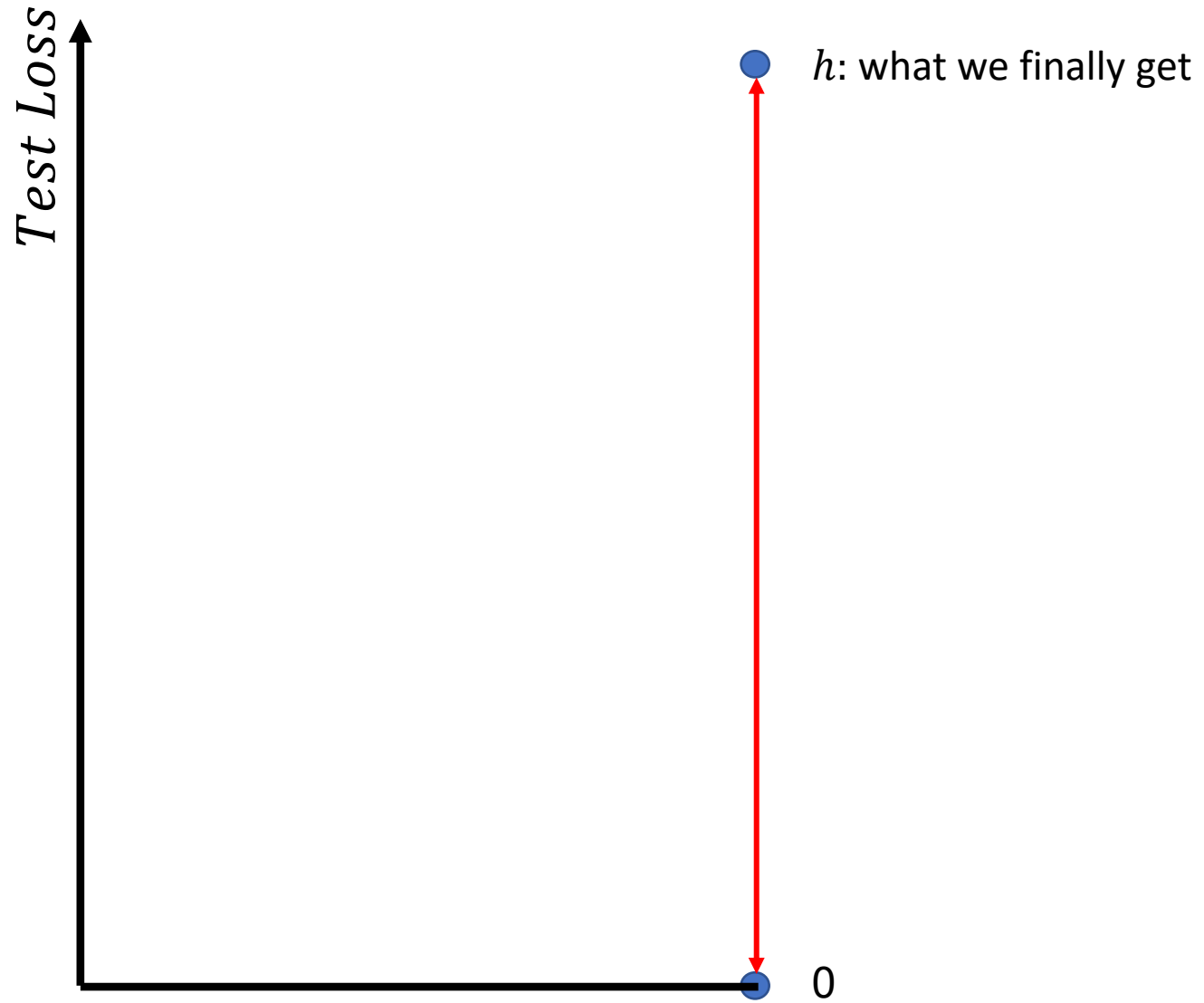
● *h*: what we finally get



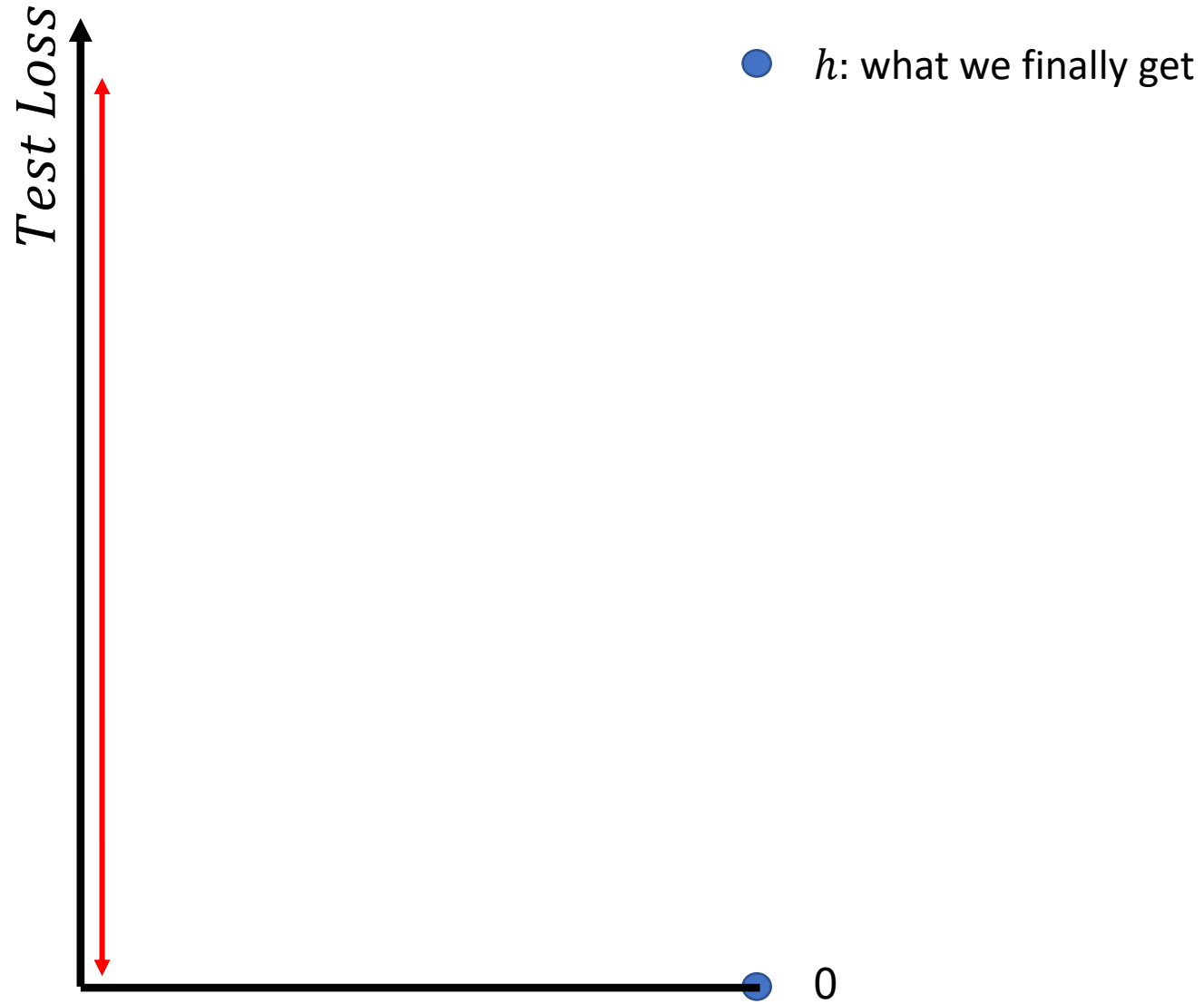
Error decomposition



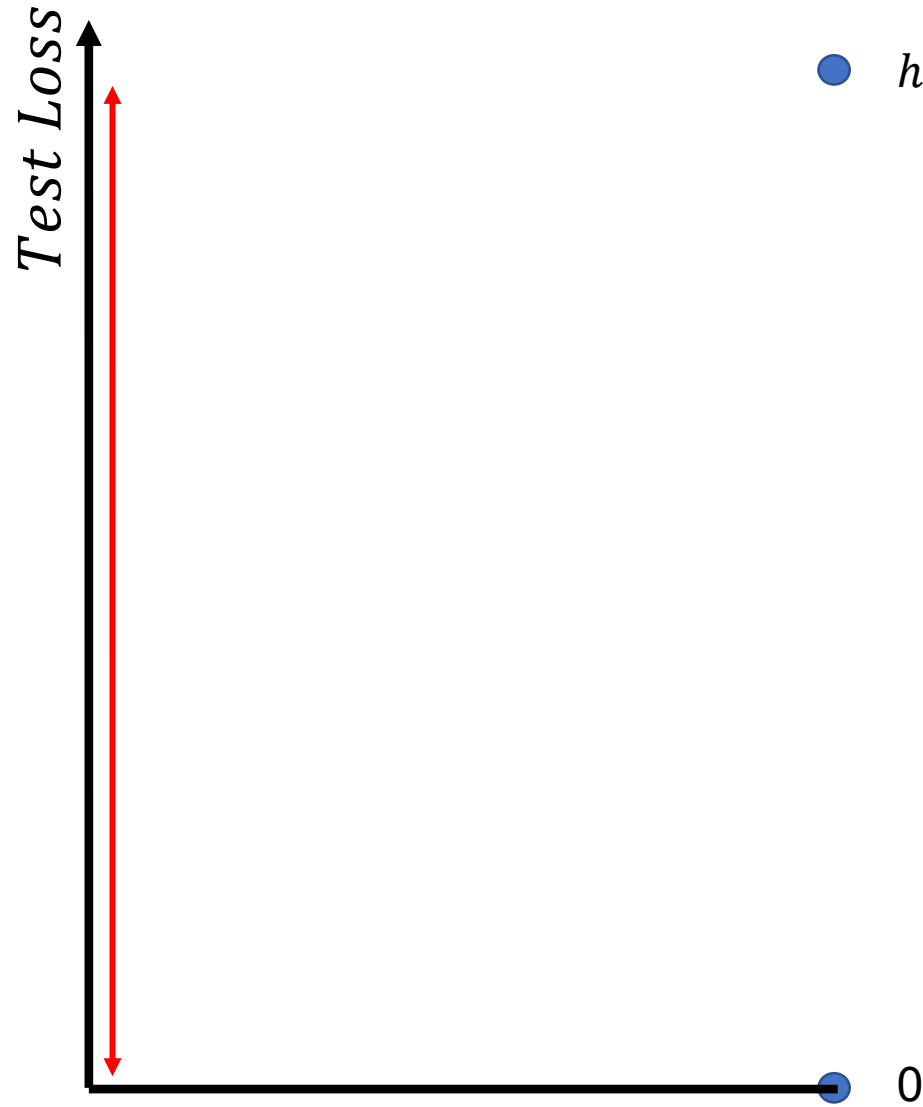
Error decomposition



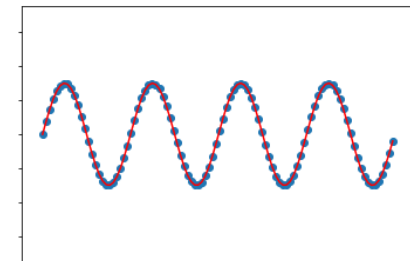
Error decomposition



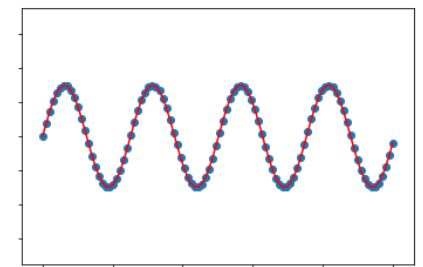
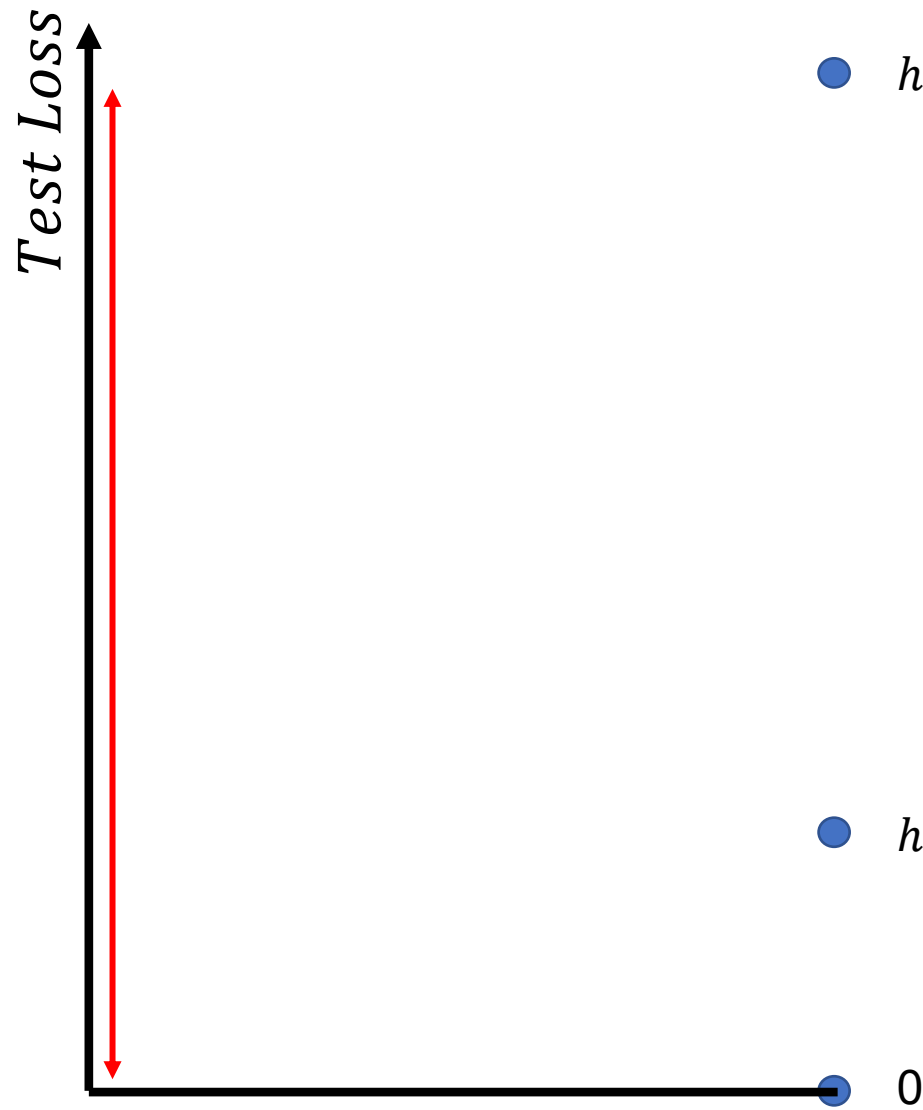
Error decomposition



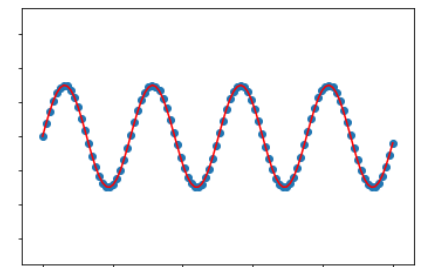
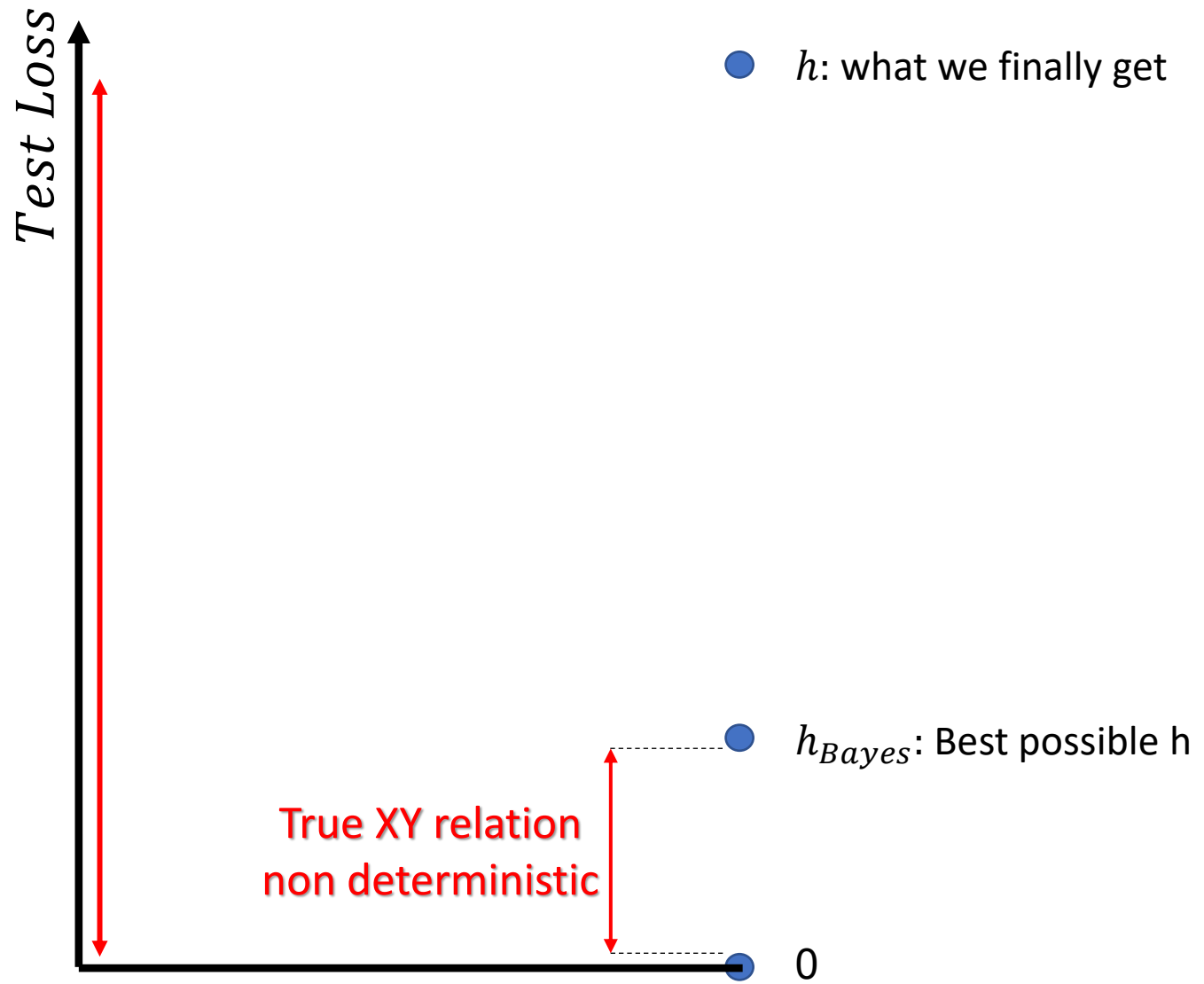
● h : what we finally get



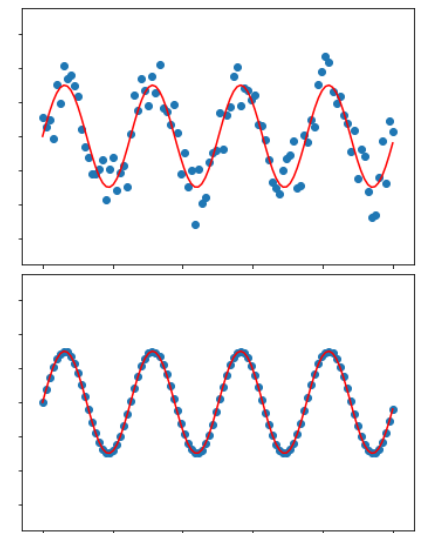
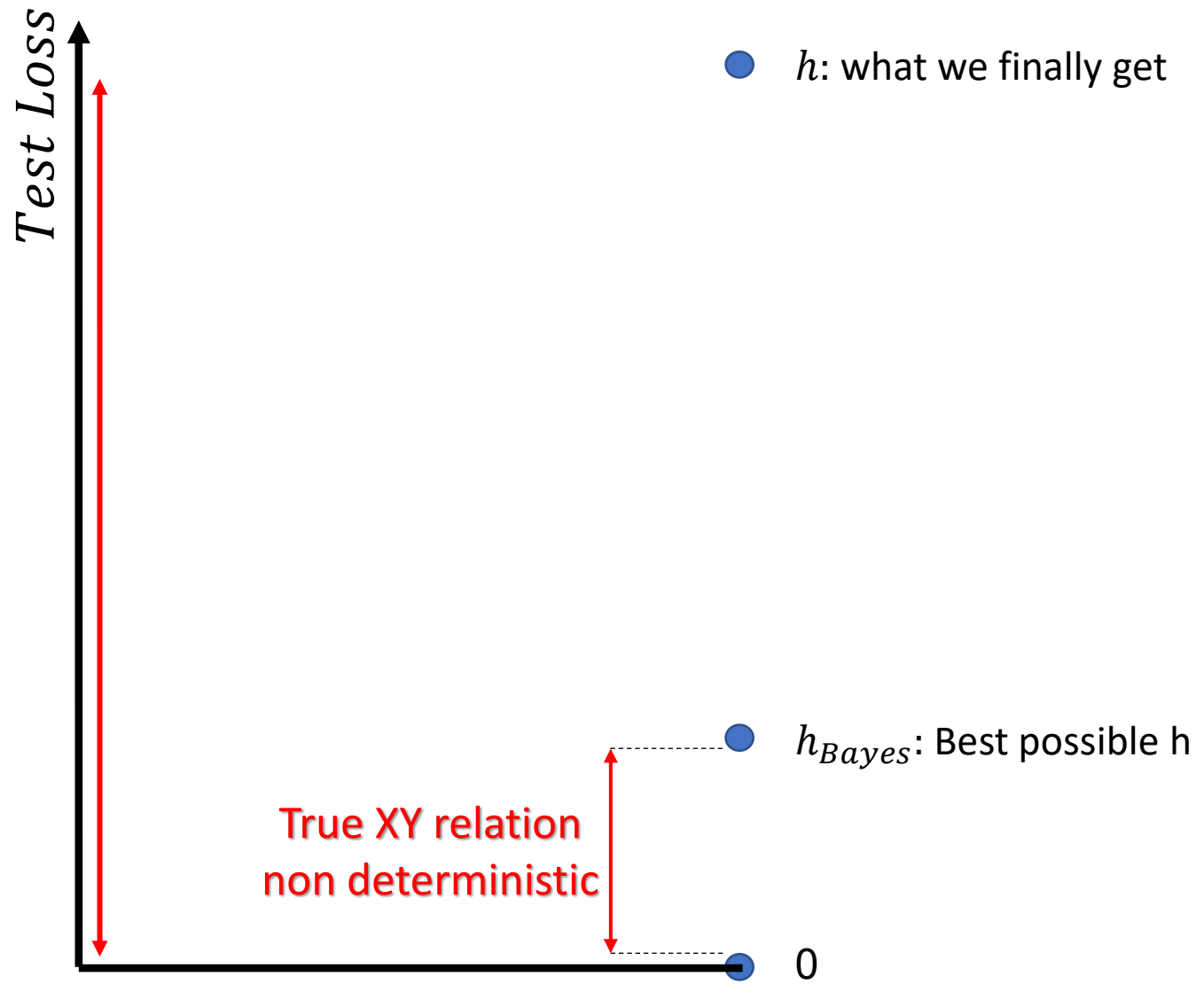
Error decomposition



Error decomposition

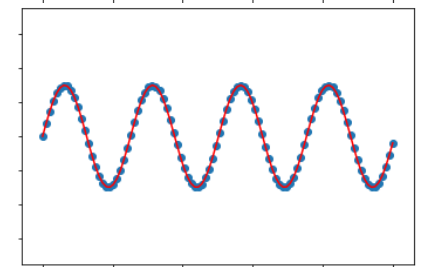
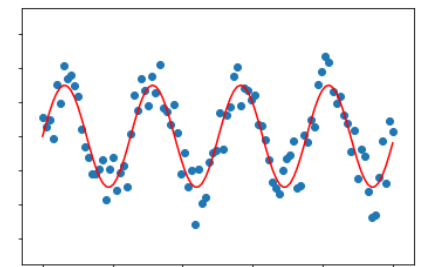
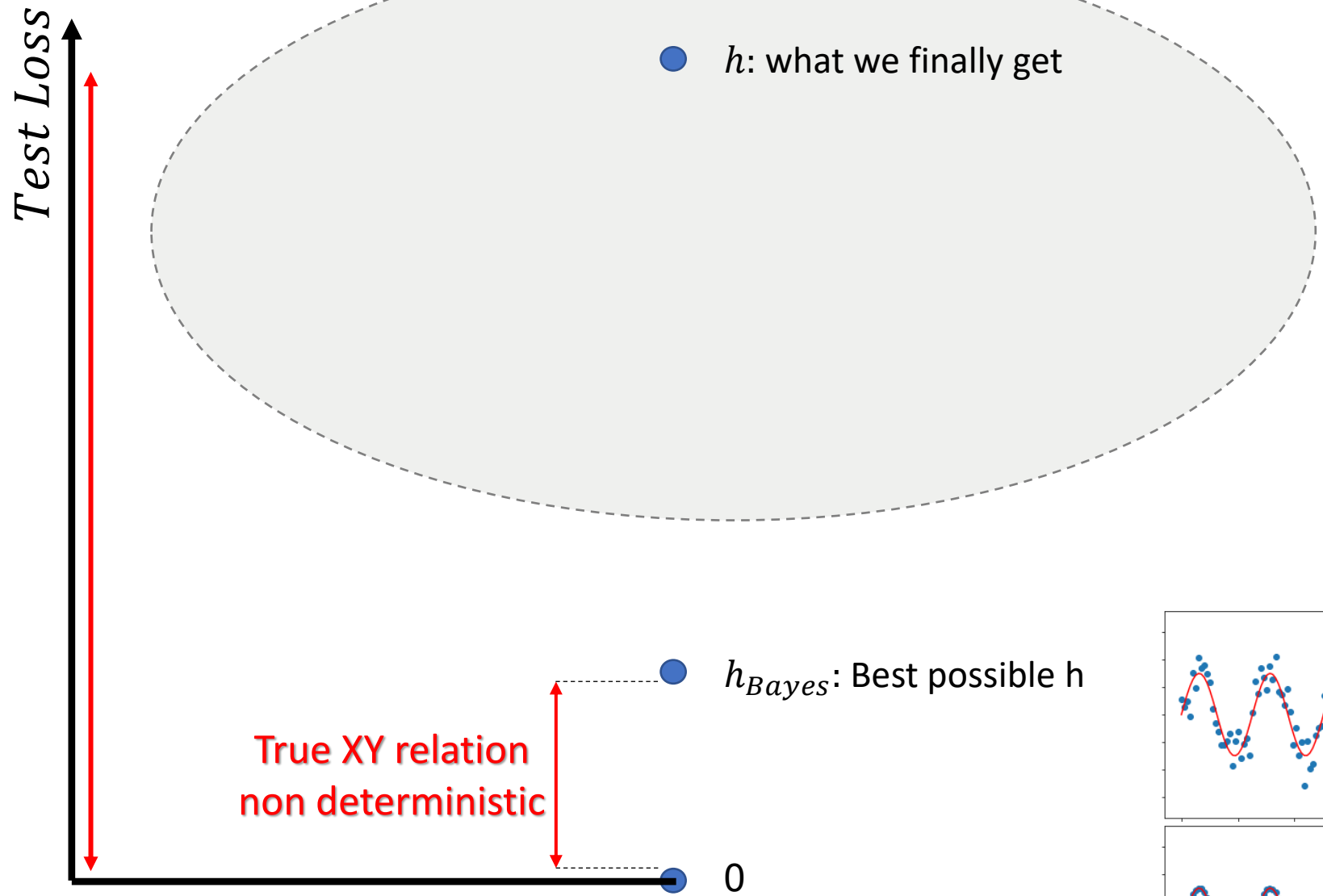


Error decomposition

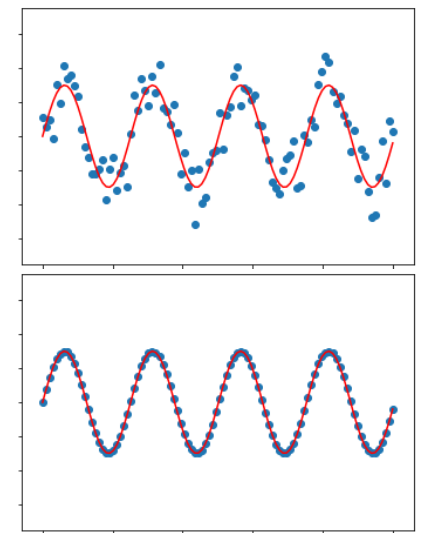
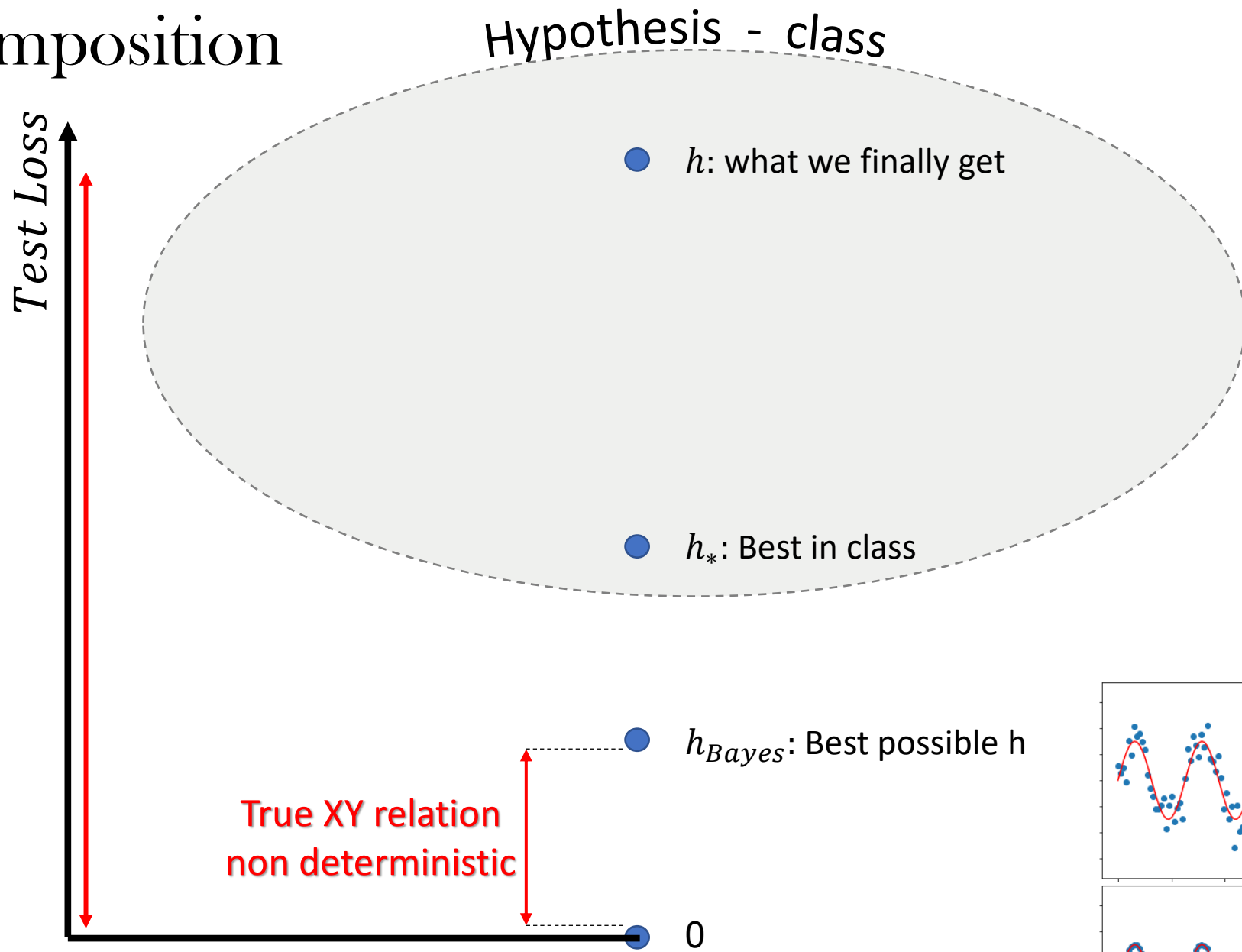


Error decomposition

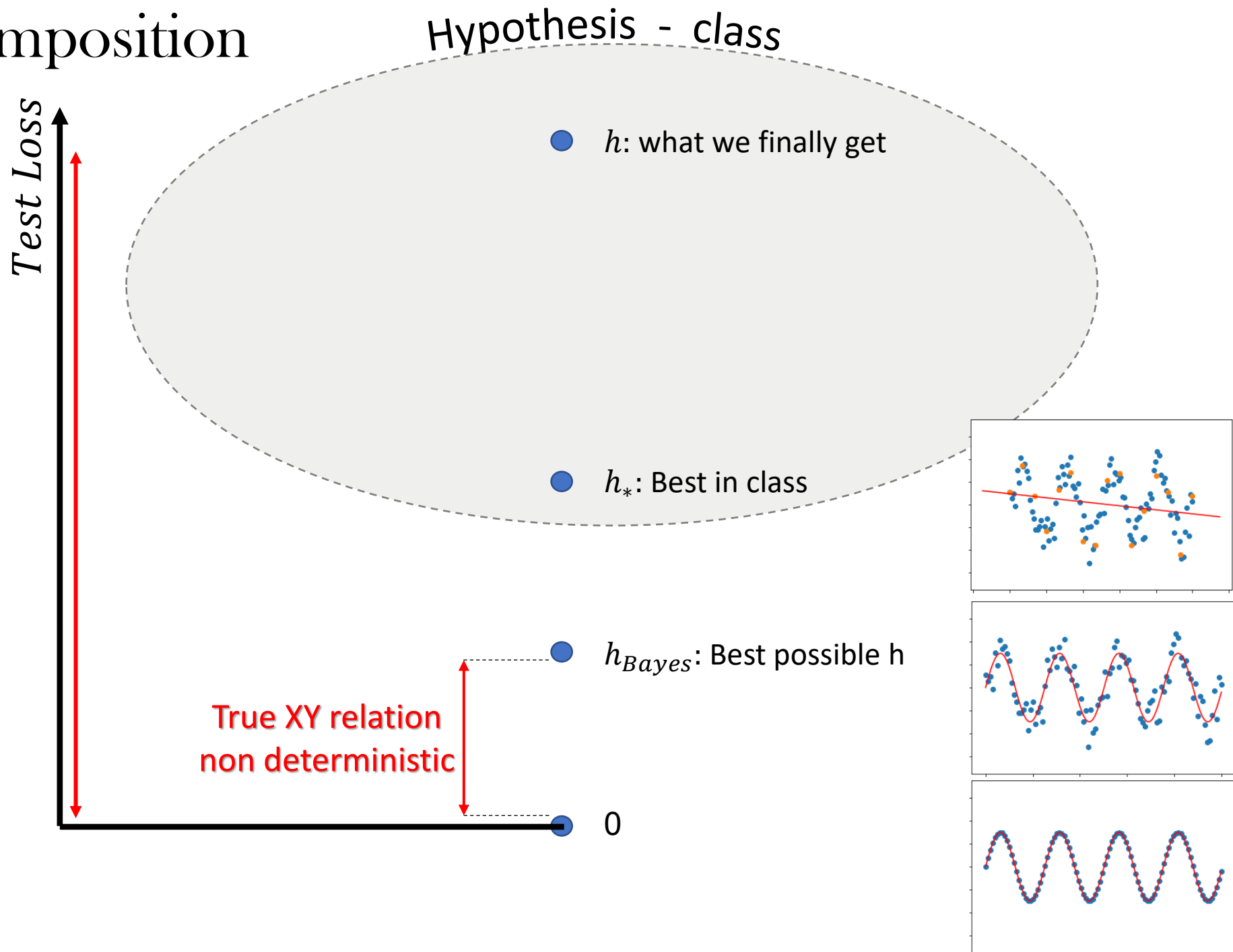
Hypothesis - class



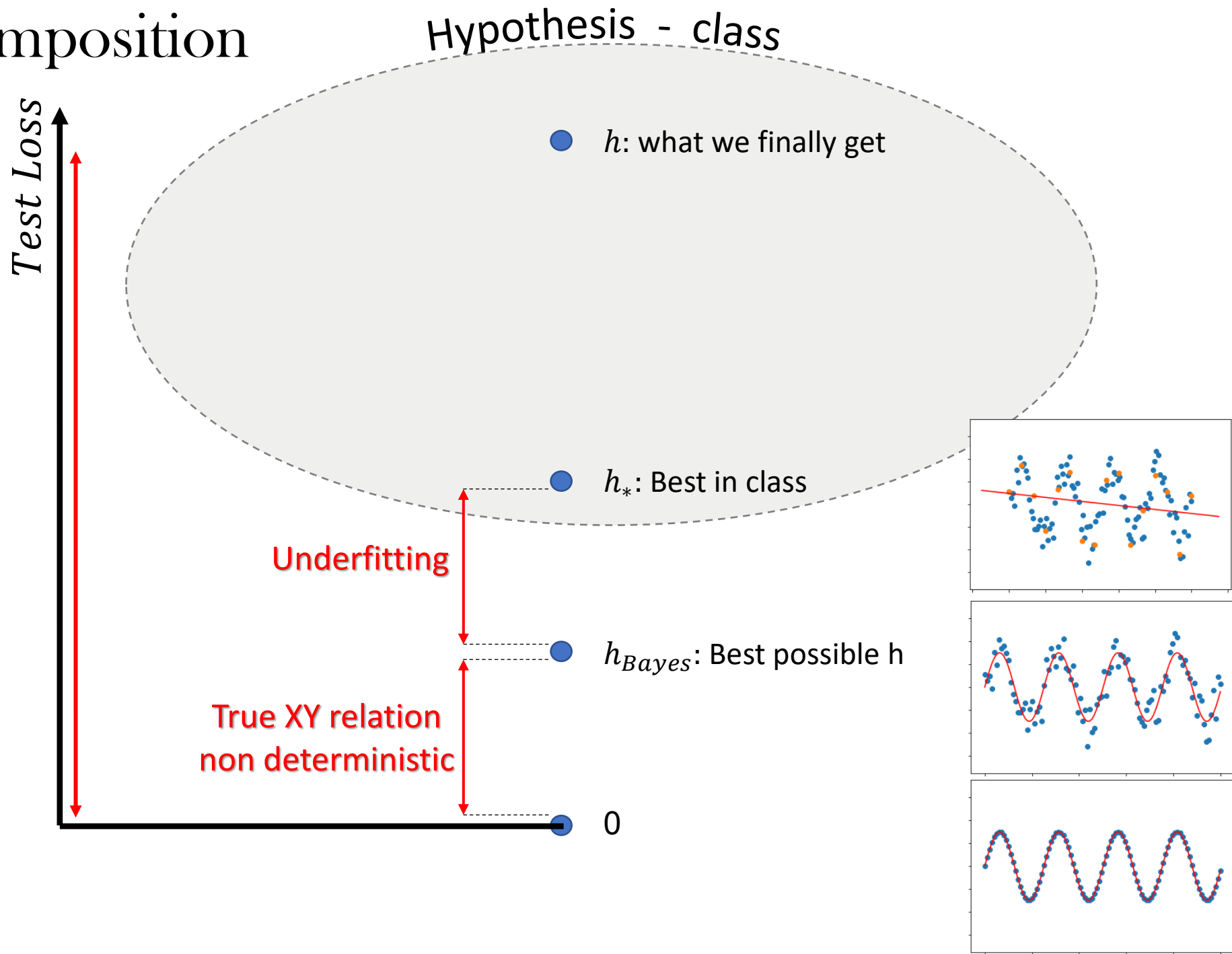
Error decomposition



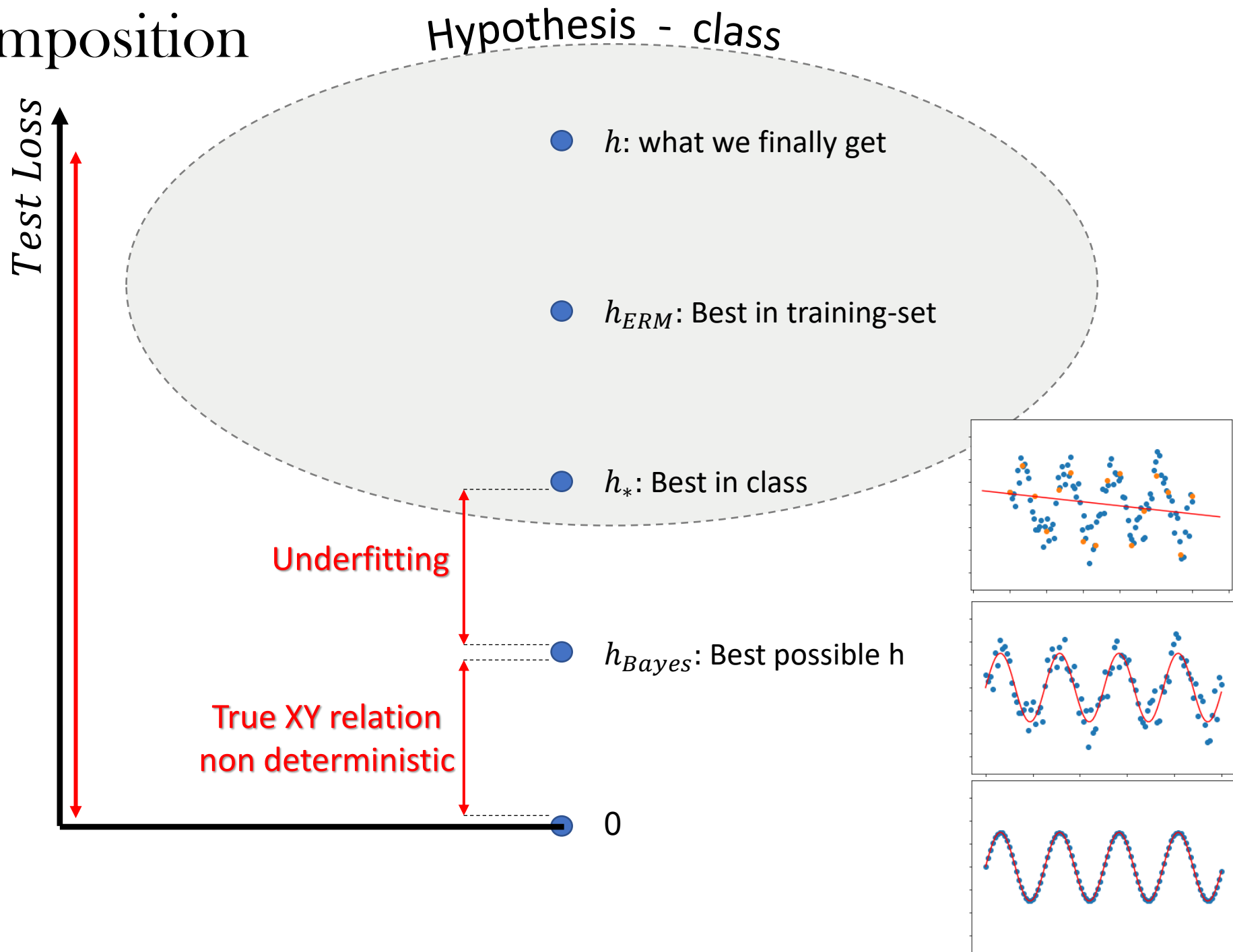
Error decomposition



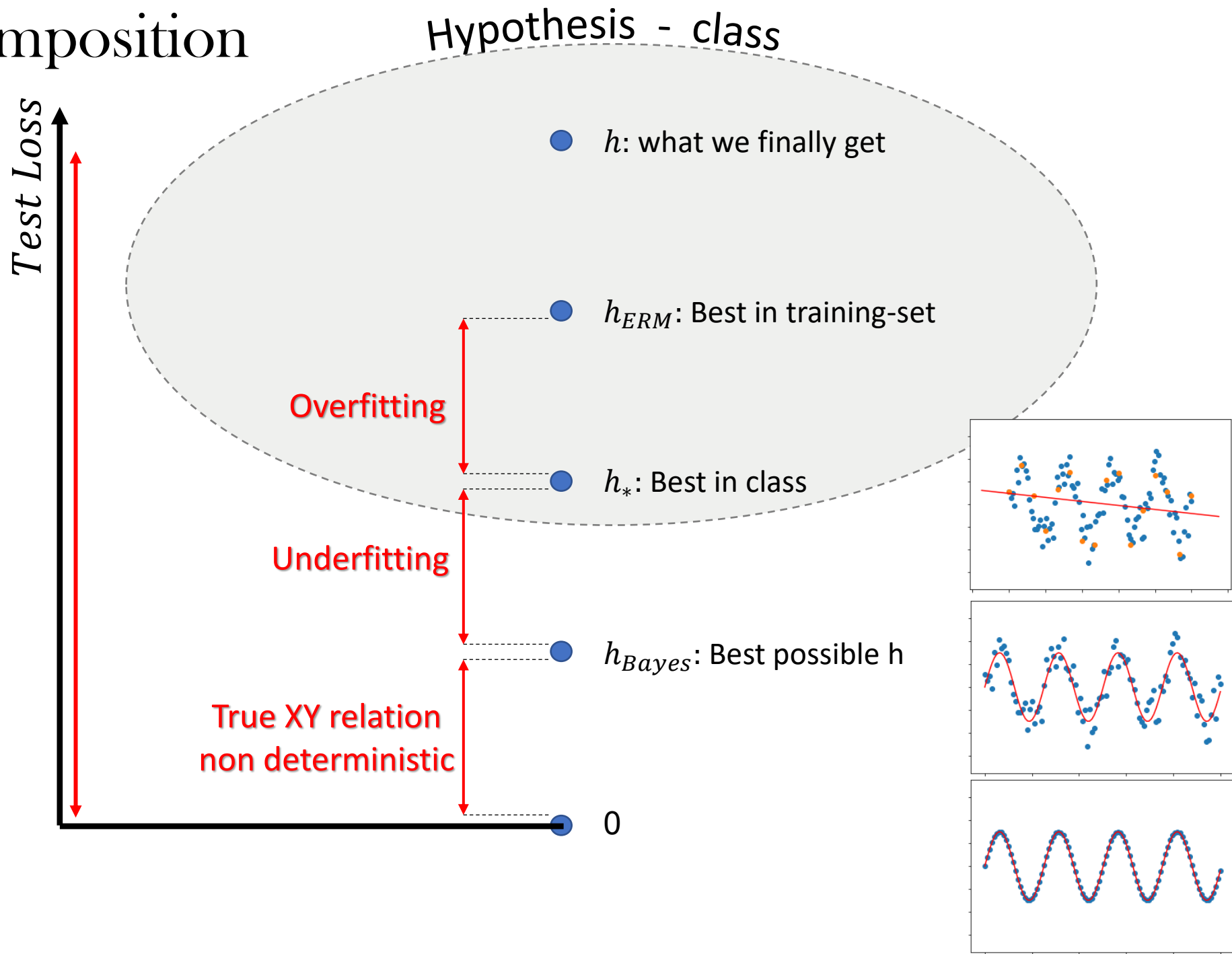
Error decomposition



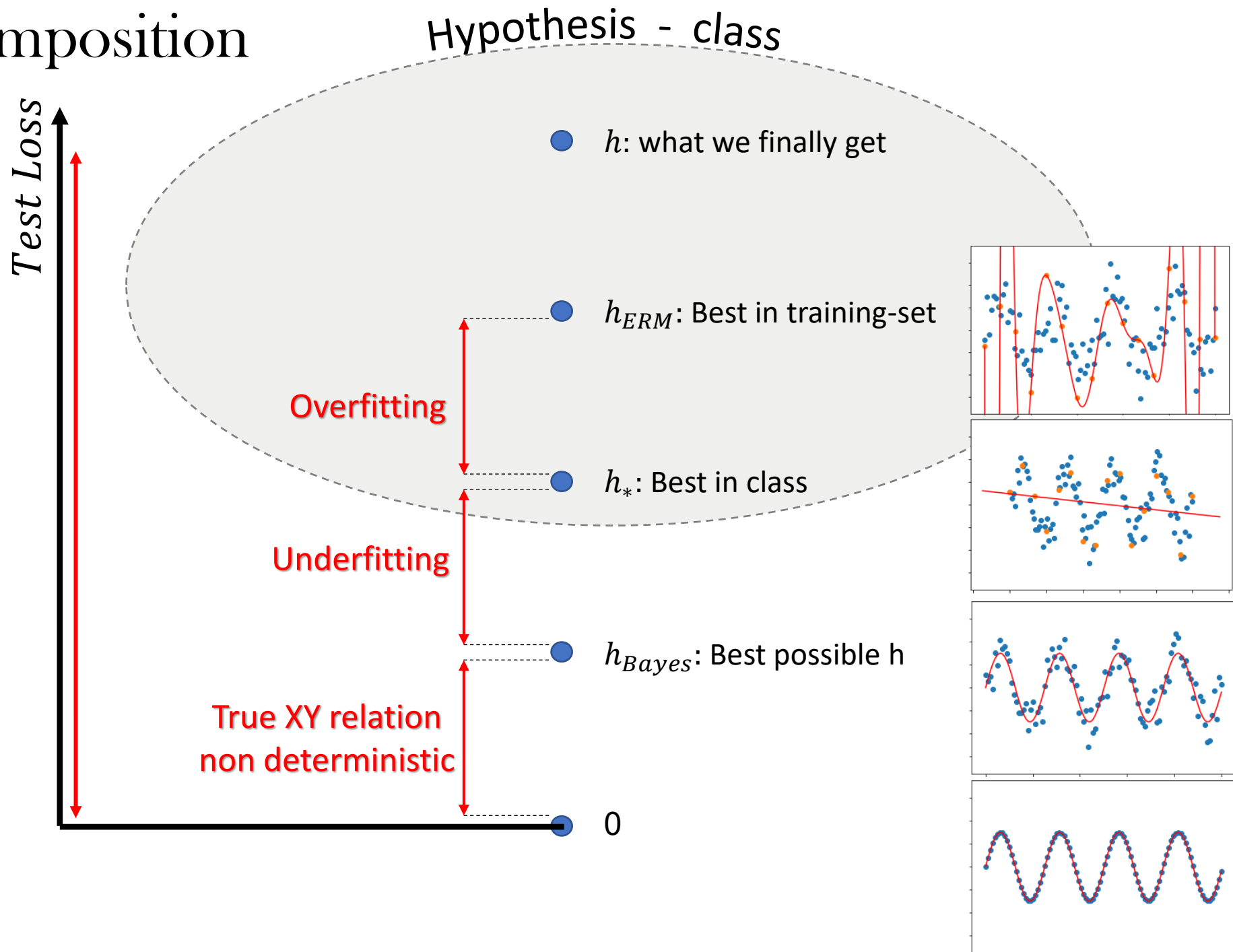
Error decomposition



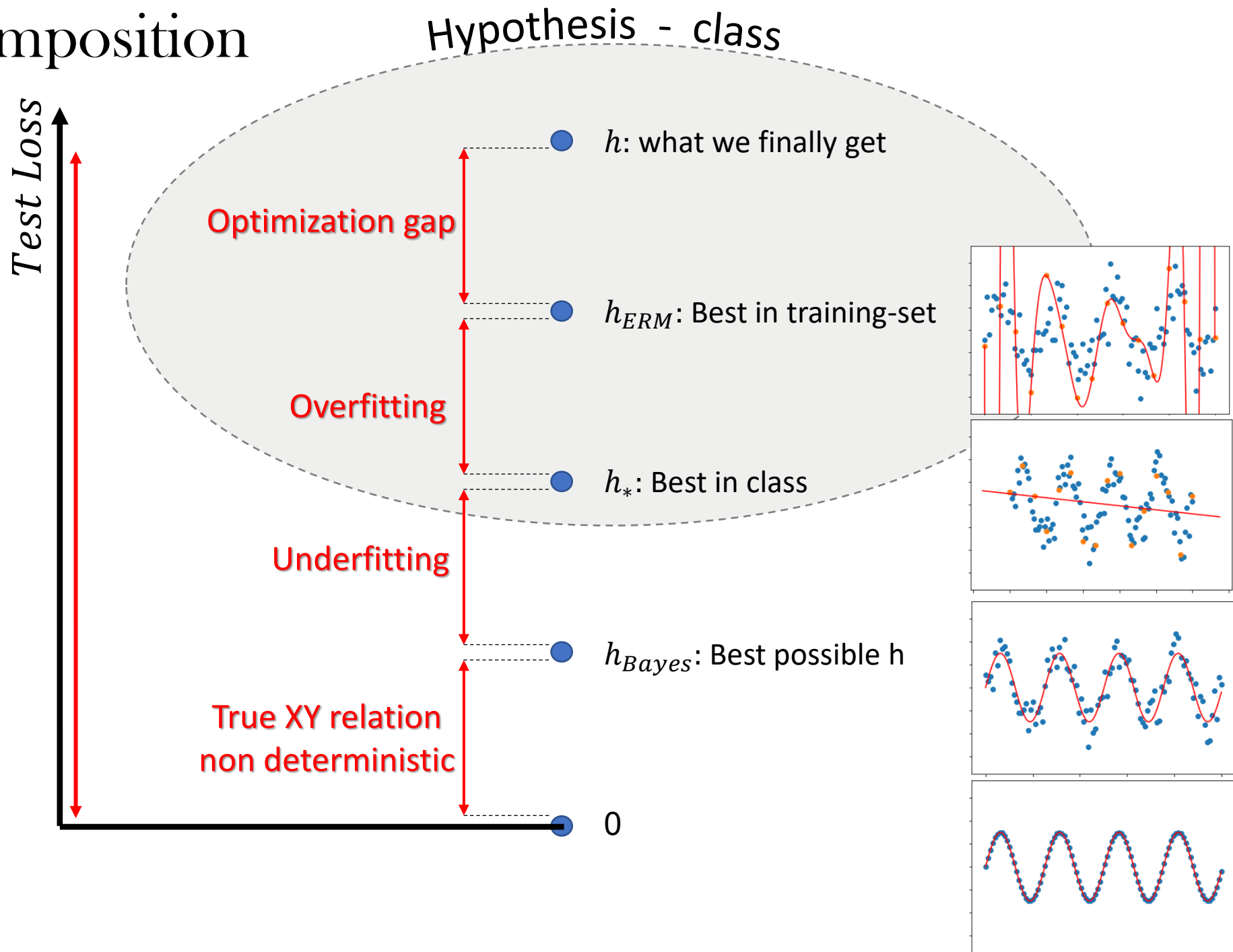
Error decomposition



Error decomposition

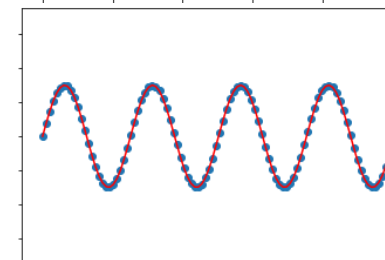
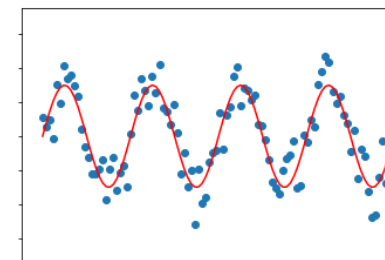
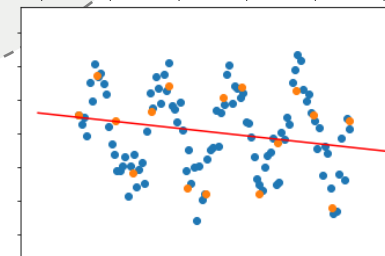
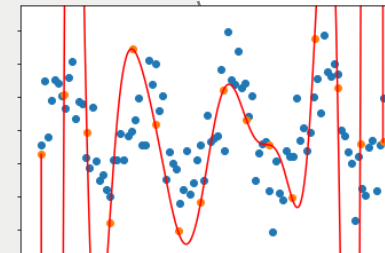
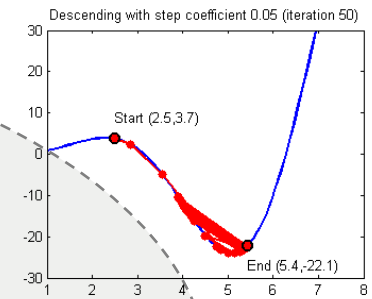
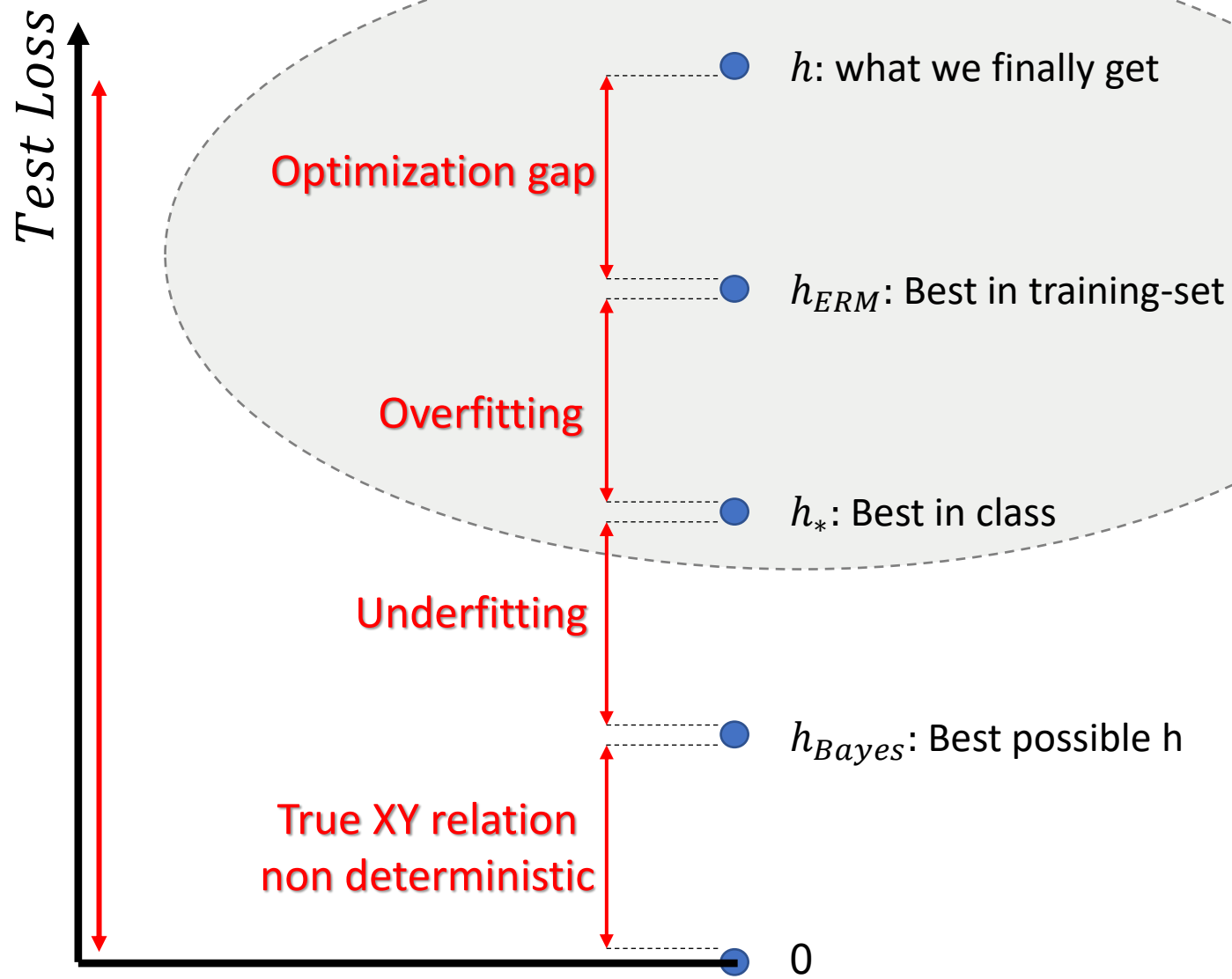


Error decomposition

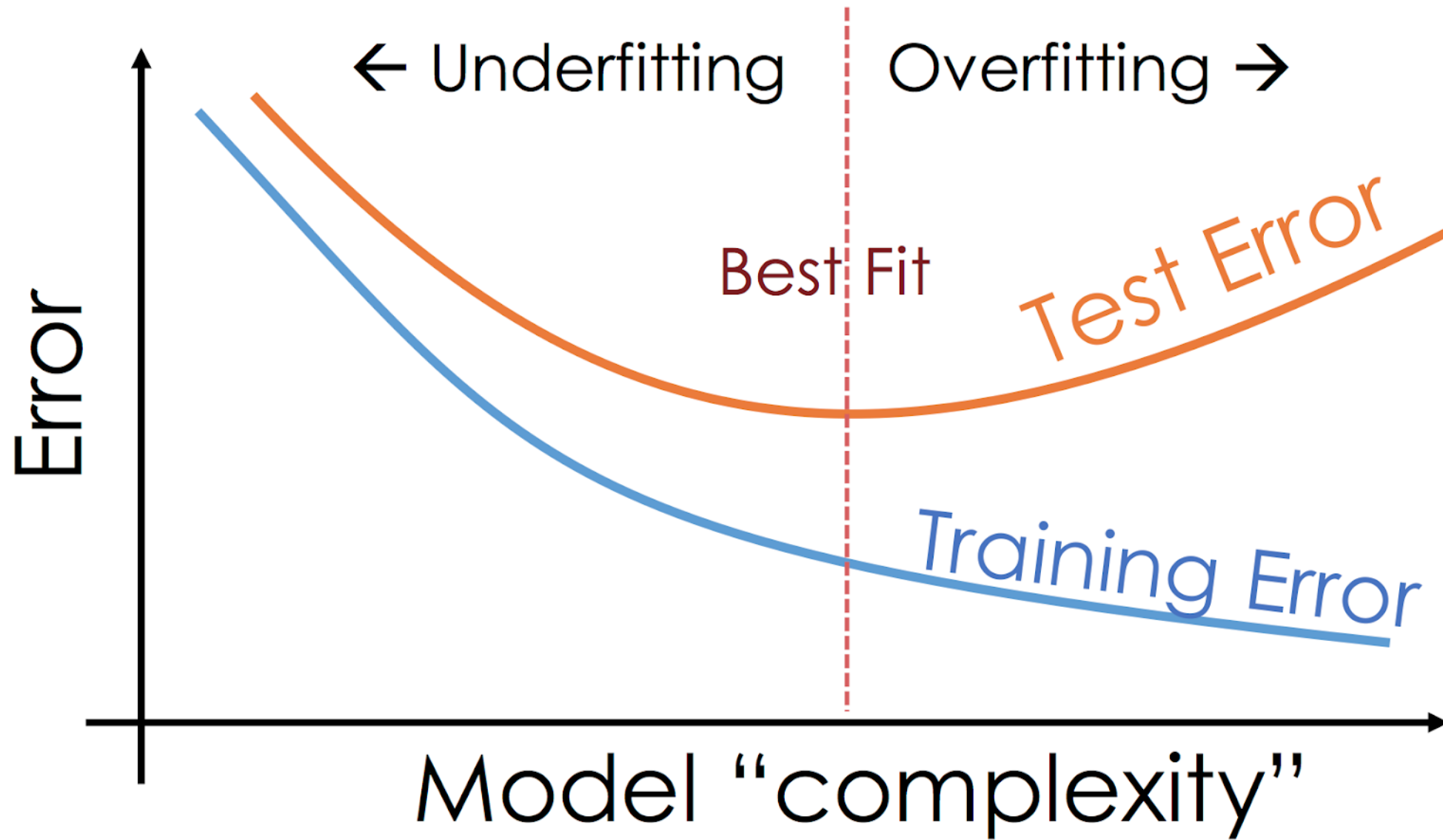


Error decomposition

Hypothesis - class



Generalization



Overfitting- Data influence



Overfitting- Data influence



Matrix A

$$\begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$$

Overfitting- Data influence



Matrix A

$$\begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$$

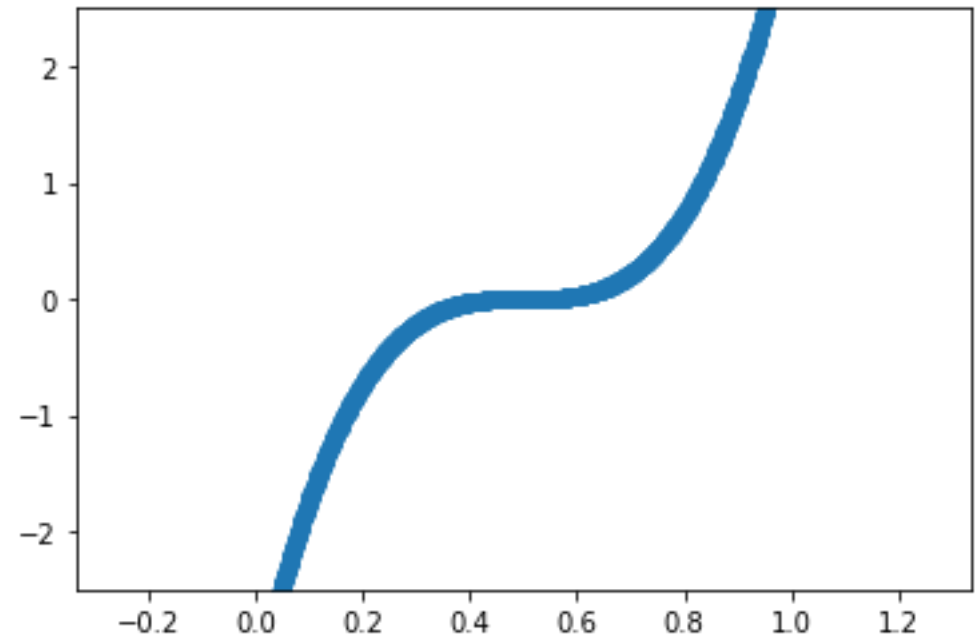
A^2

$$\begin{bmatrix} 33 & 44 \\ 66 & 88 \end{bmatrix}$$

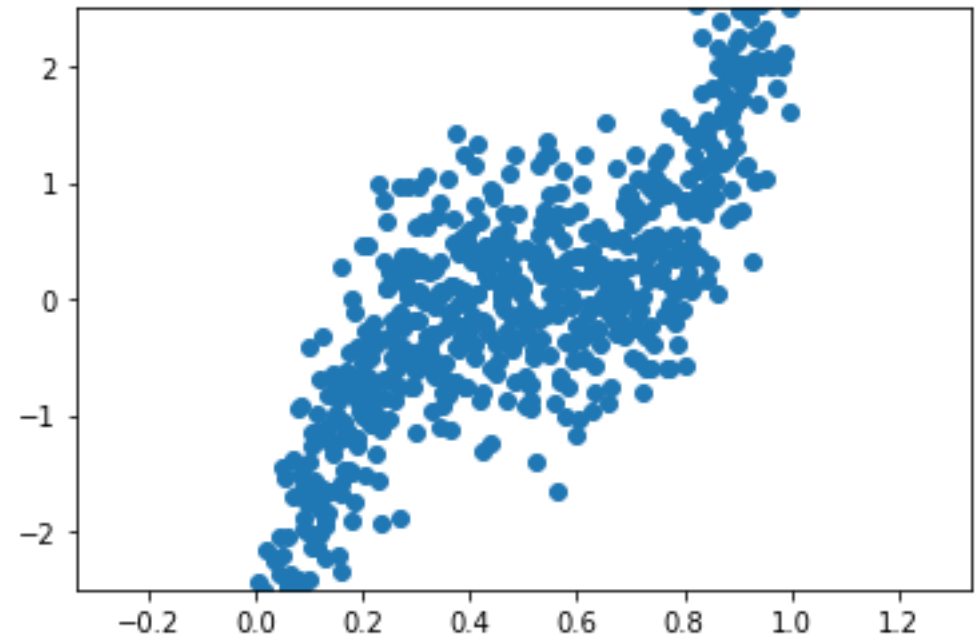
Polynomial fitting



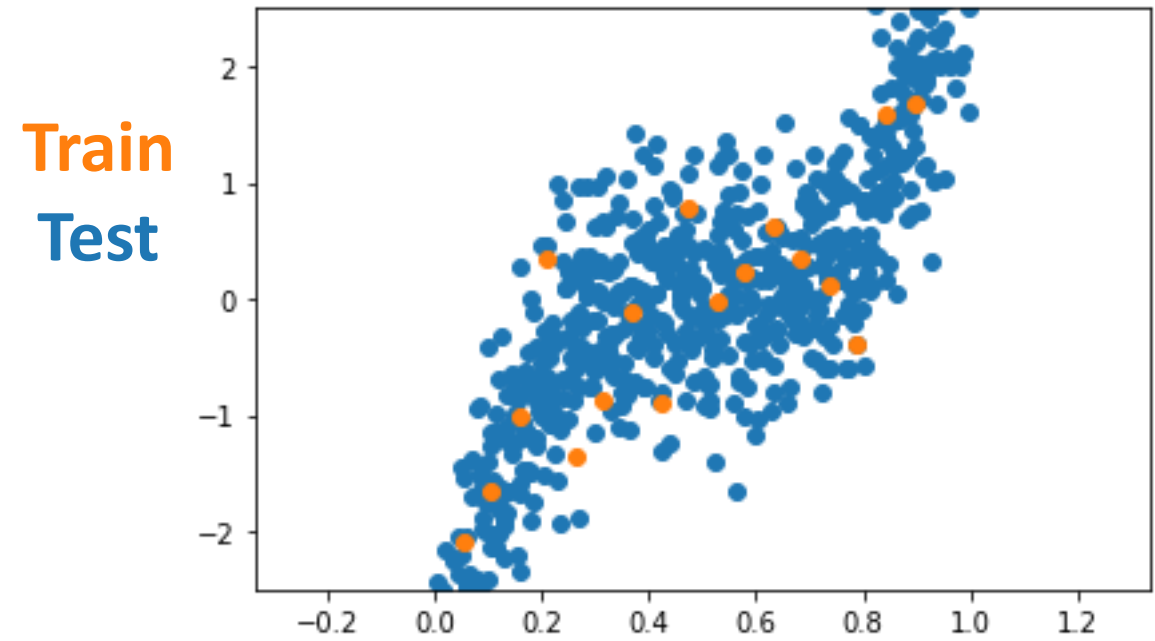
Polynomial fitting



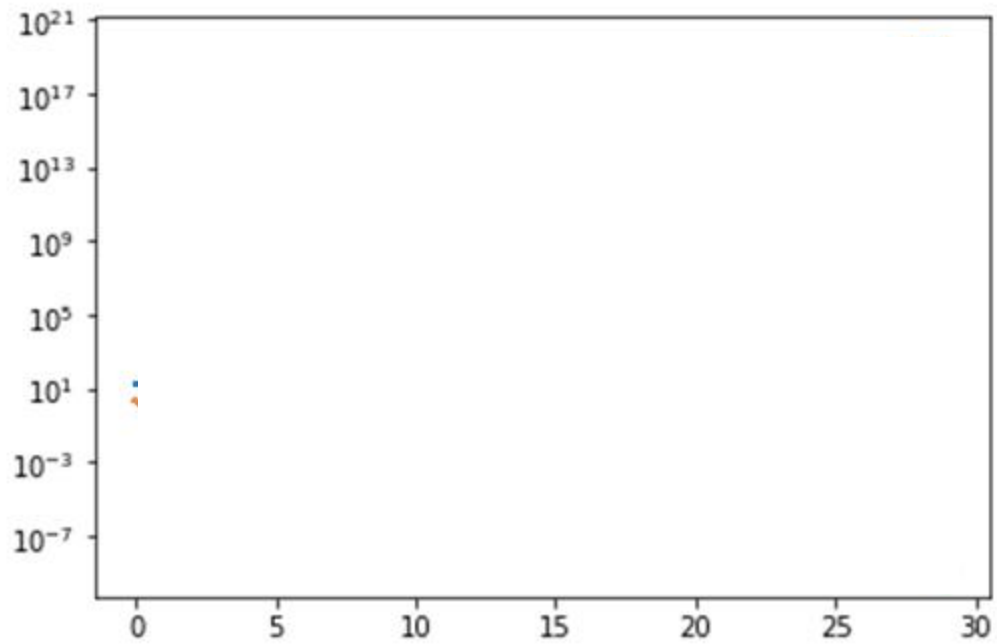
Polynomial fitting



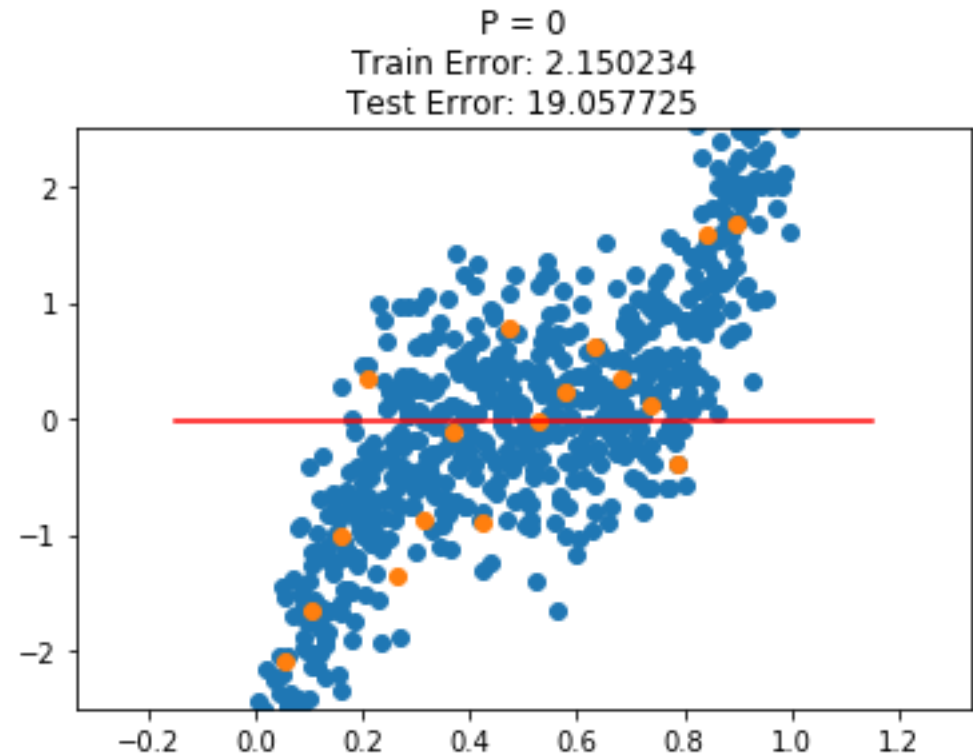
Polynomial fitting



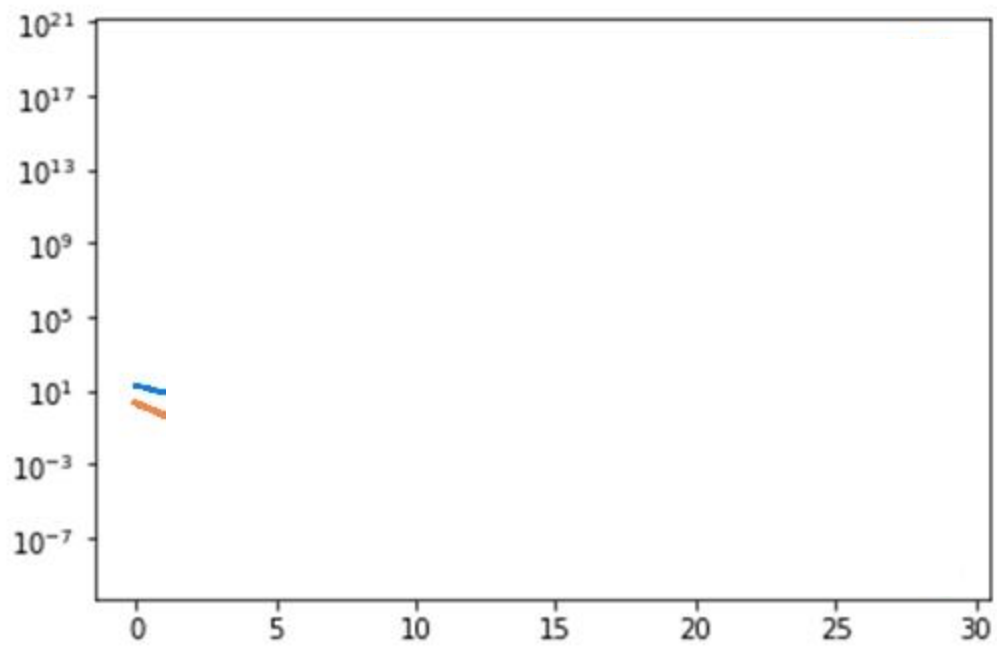
Polynomial fitting



Train
Test



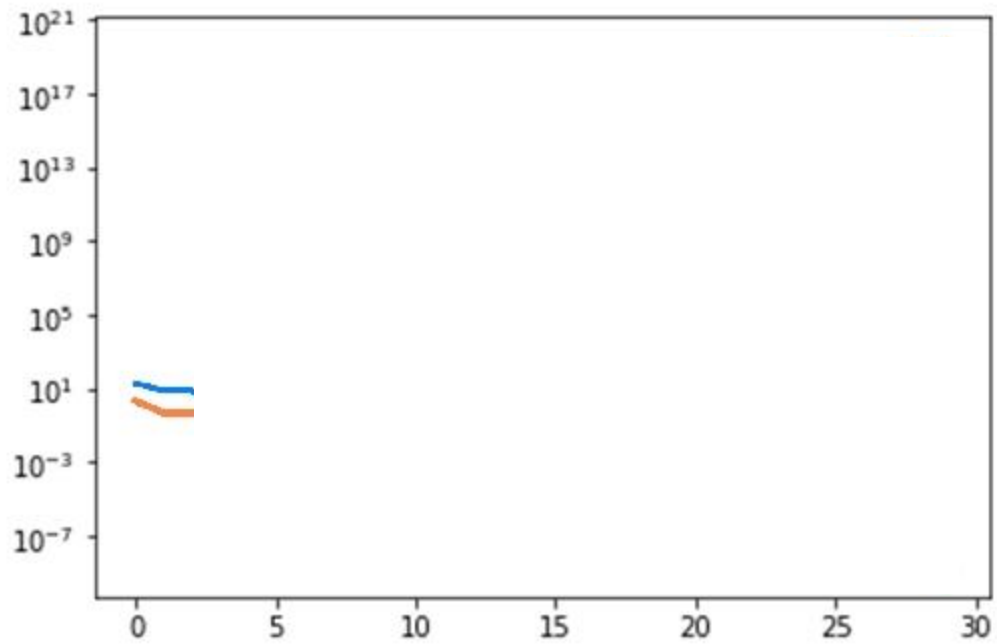
Polynomial fitting



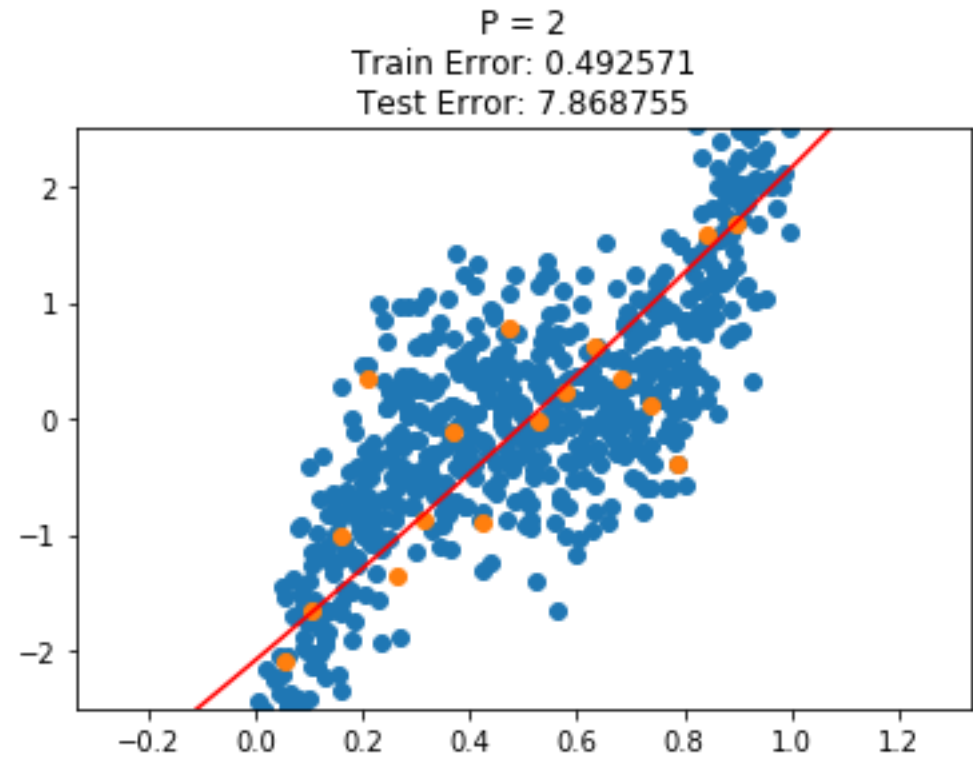
Train
Test



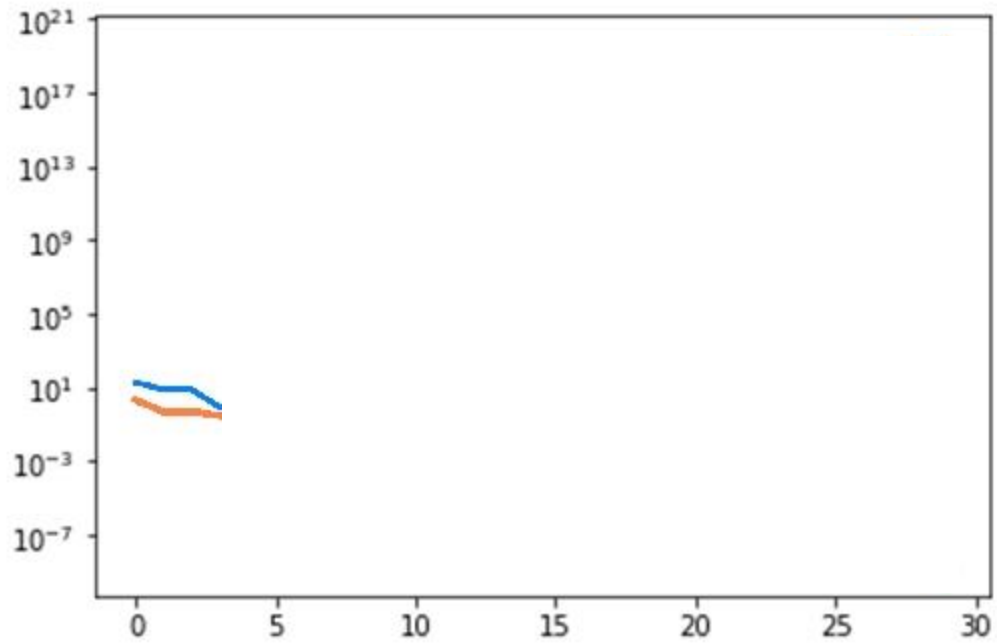
Polynomial fitting



Train
Test



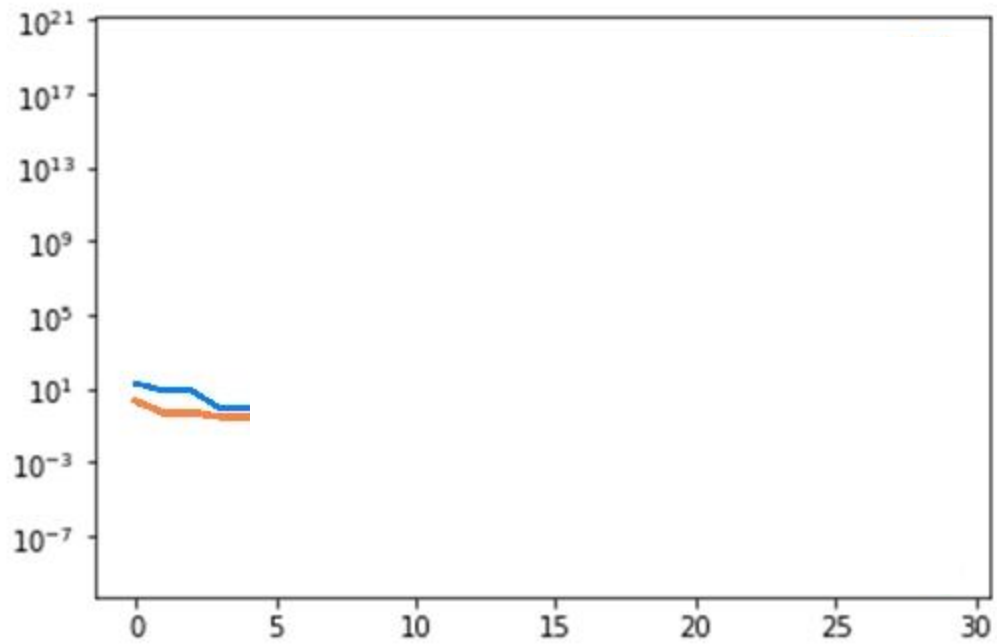
Polynomial fitting



Train
Test



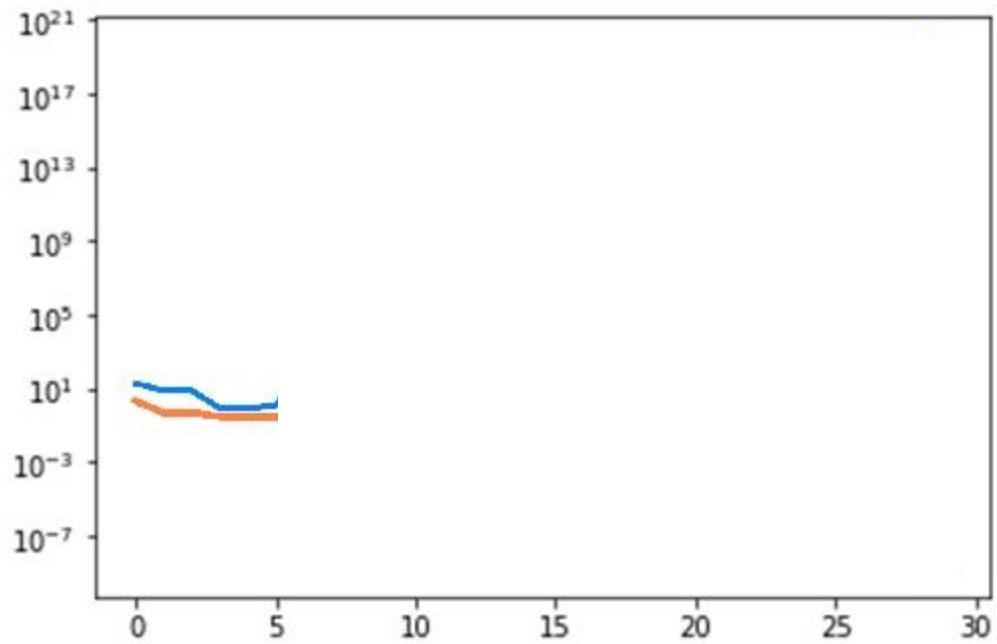
Polynomial fitting



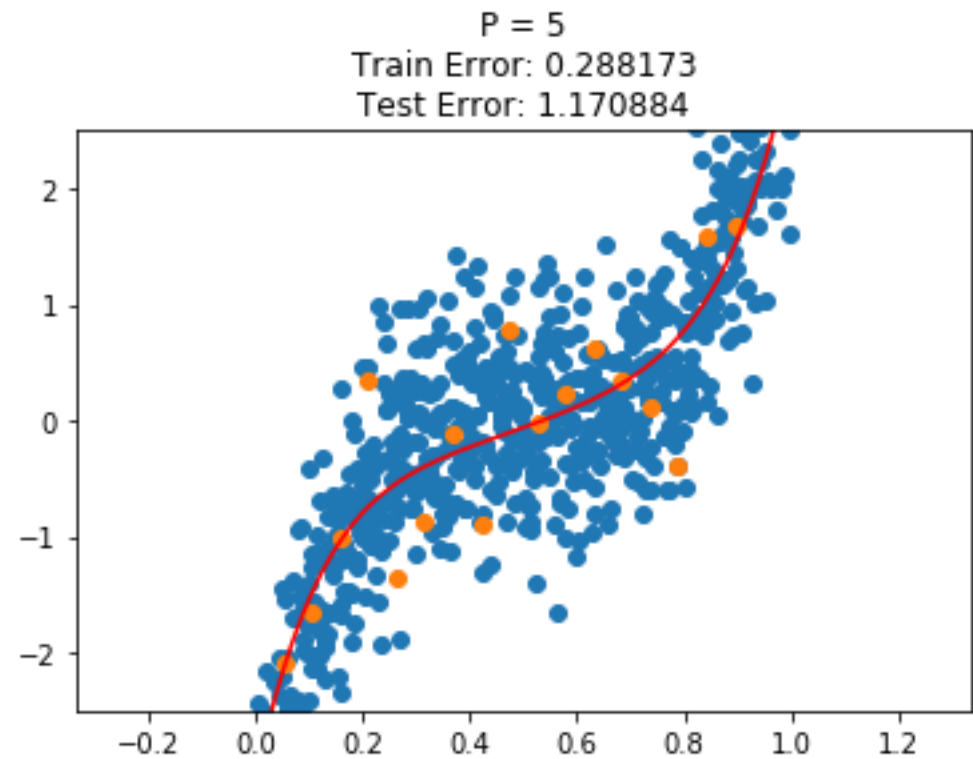
Train
Test



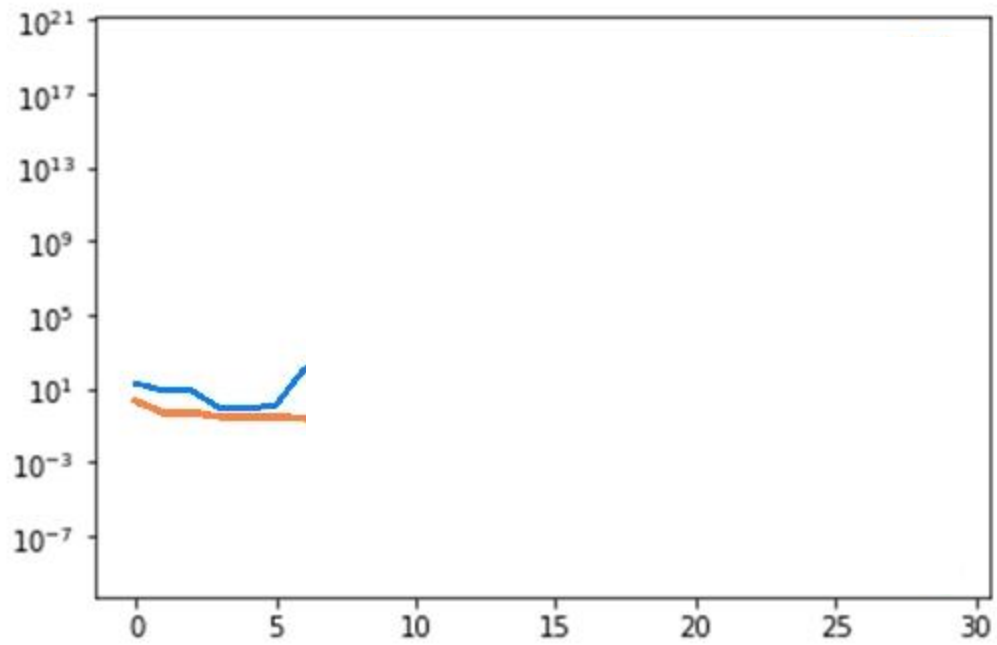
Polynomial fitting



Train
Test



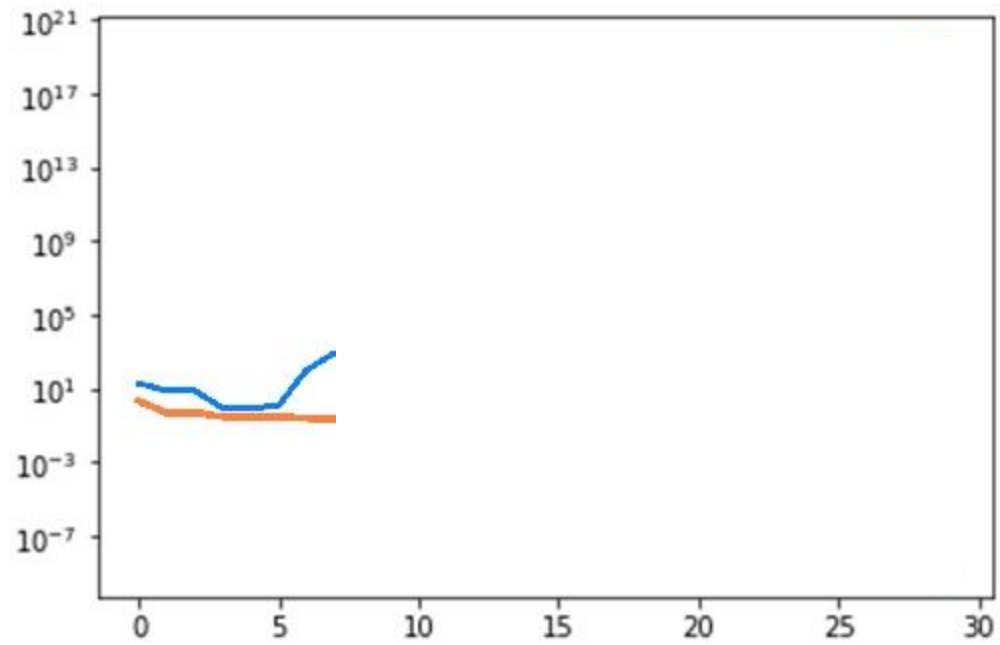
Polynomial fitting



Train
Test



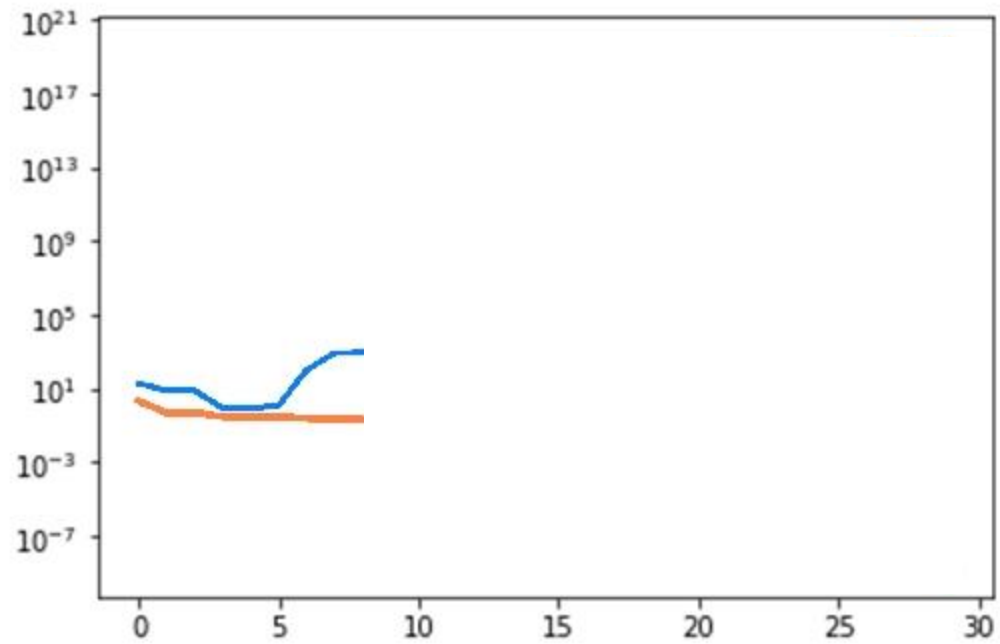
Polynomial fitting



Train
Test



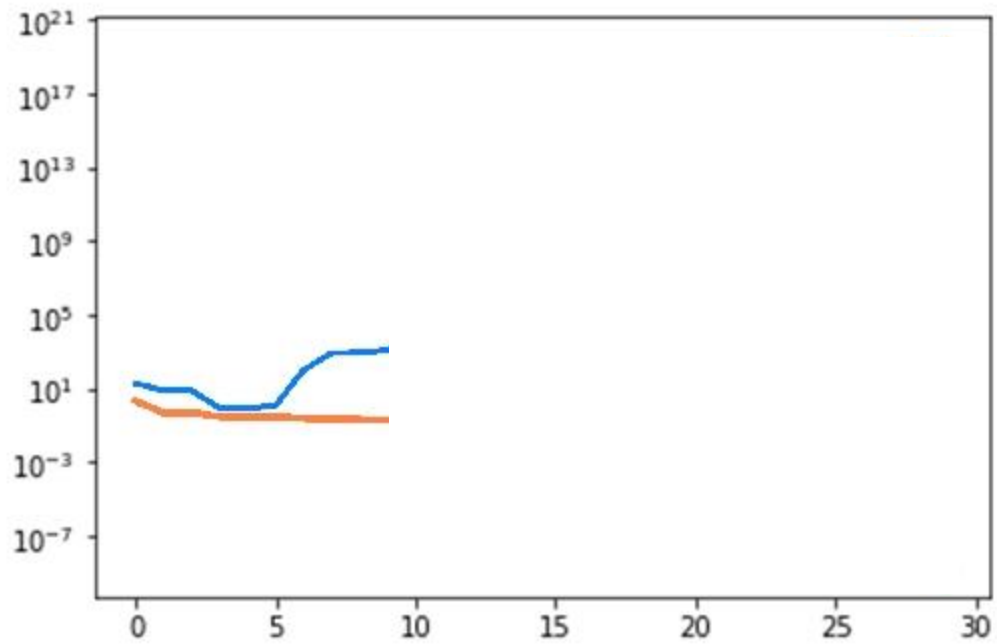
Polynomial fitting



Train
Test



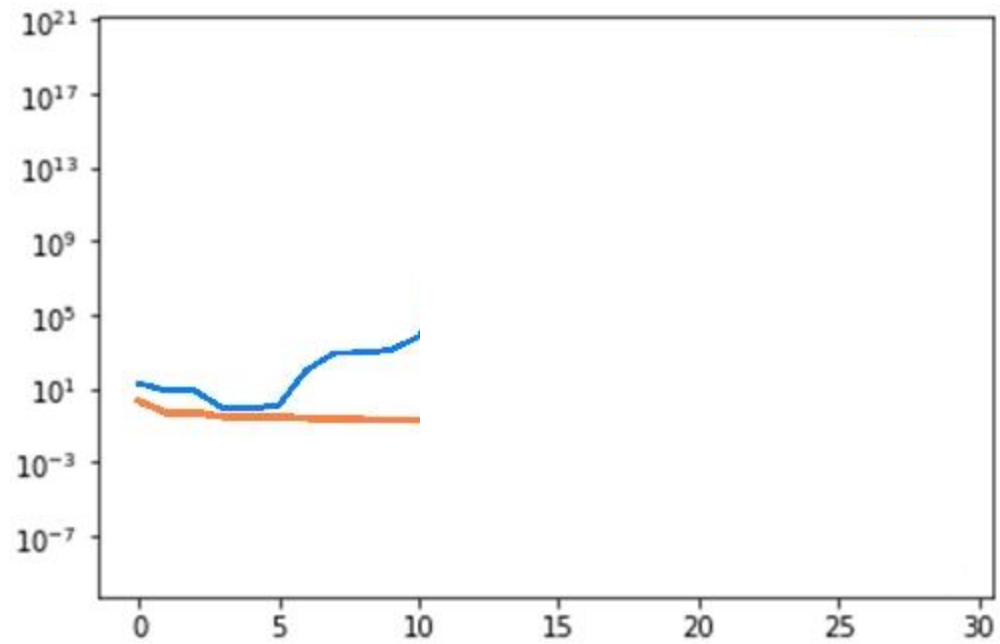
Polynomial fitting



Train
Test



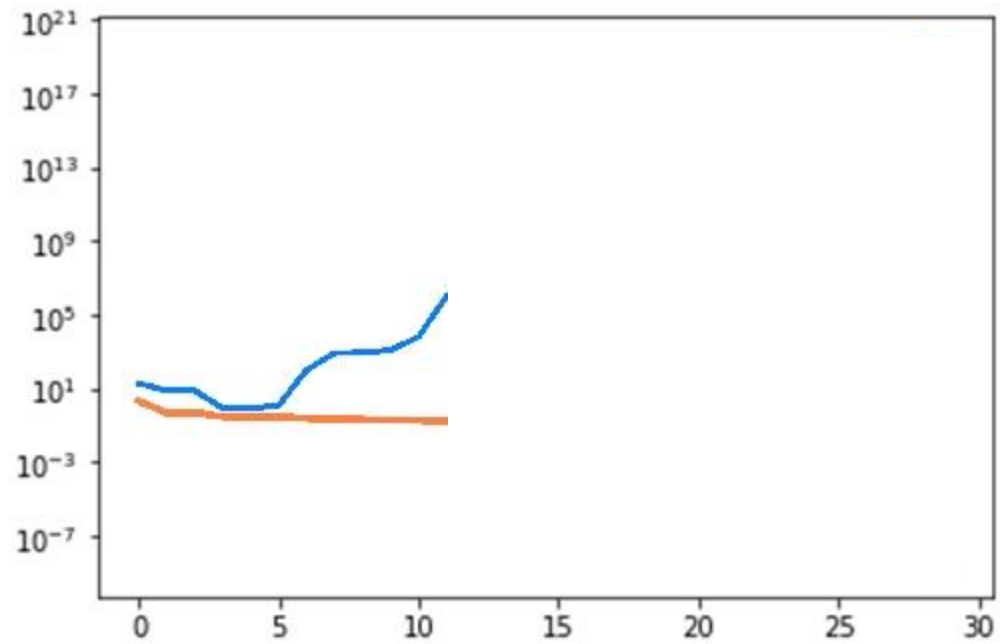
Polynomial fitting



Train
Test



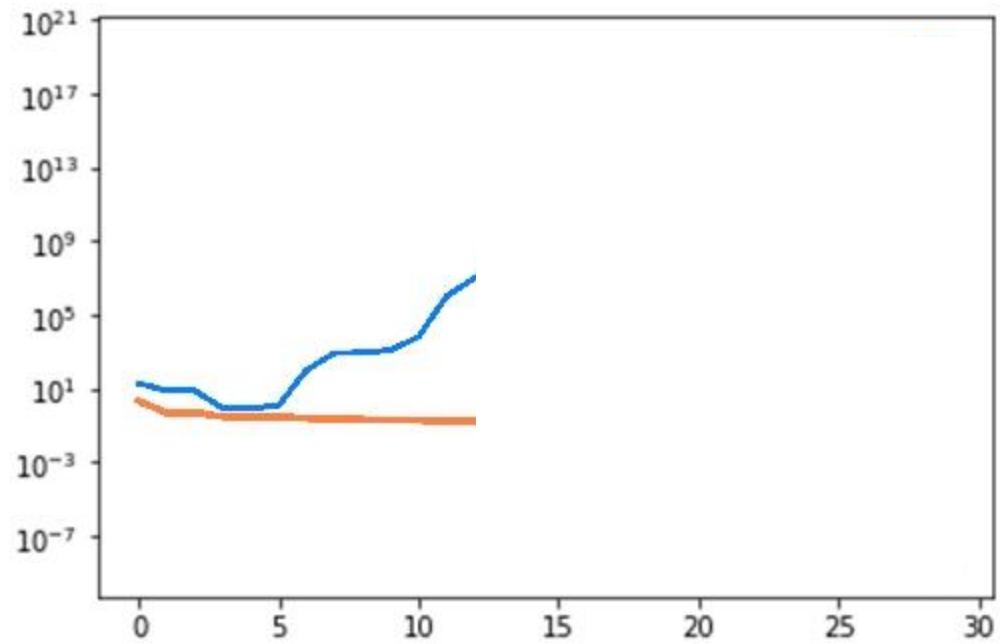
Polynomial fitting



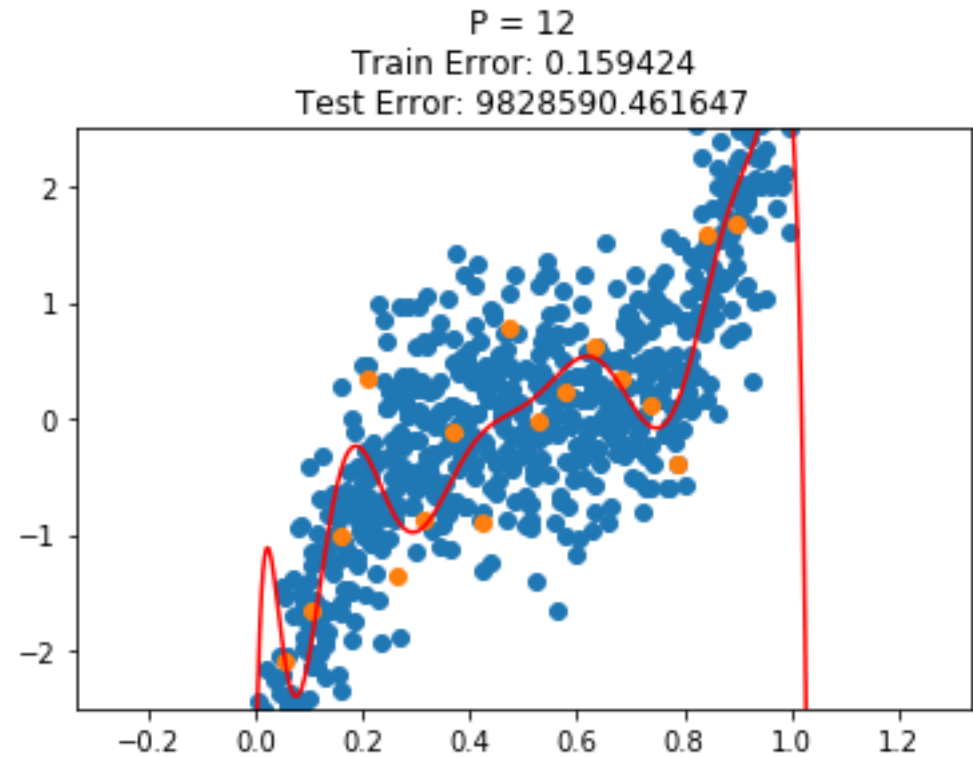
Train
Test



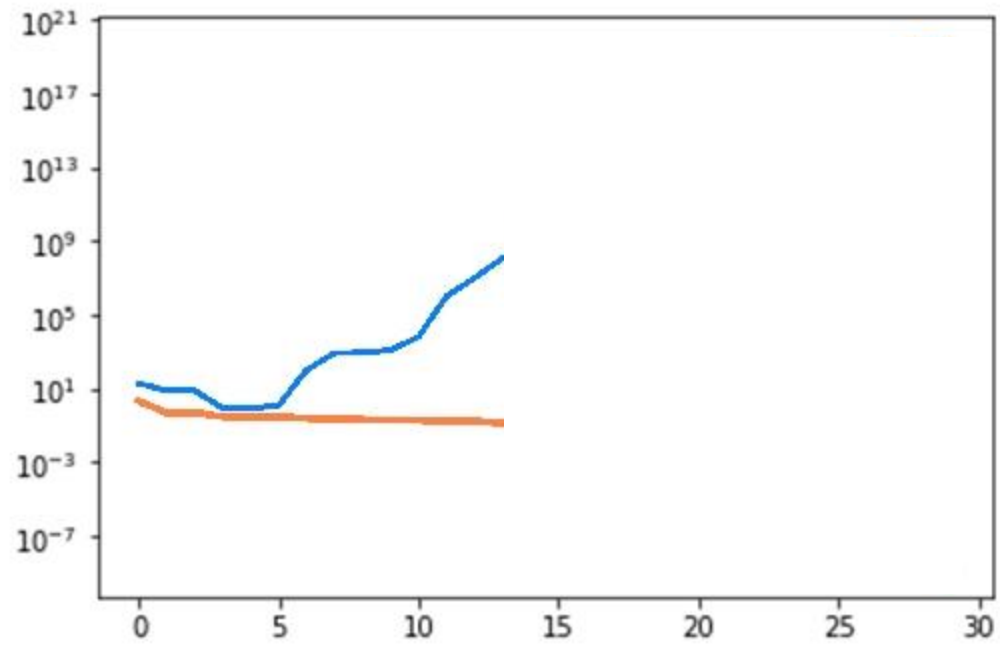
Polynomial fitting



Train
Test



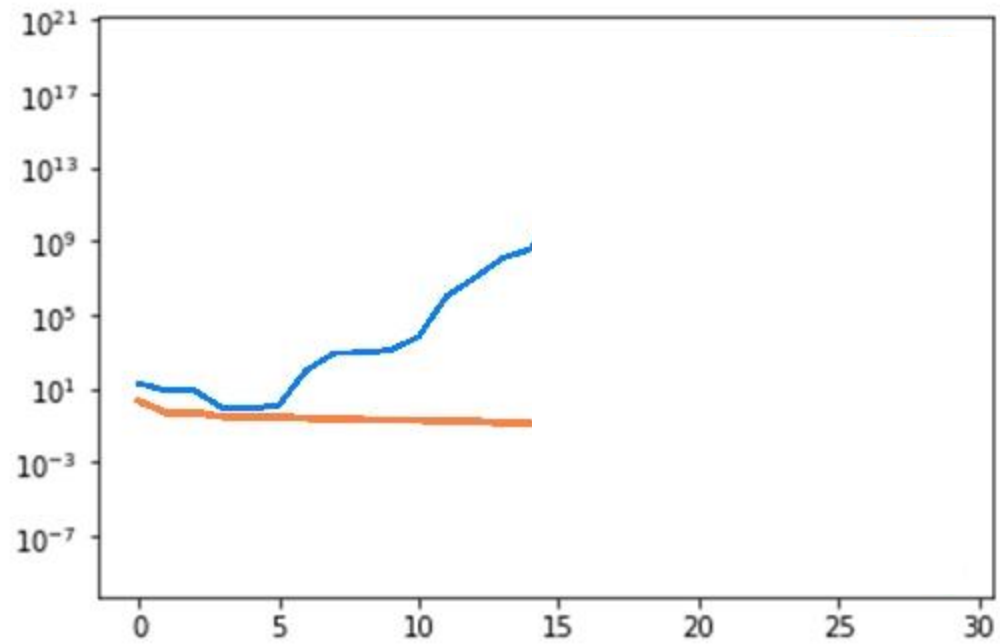
Polynomial fitting



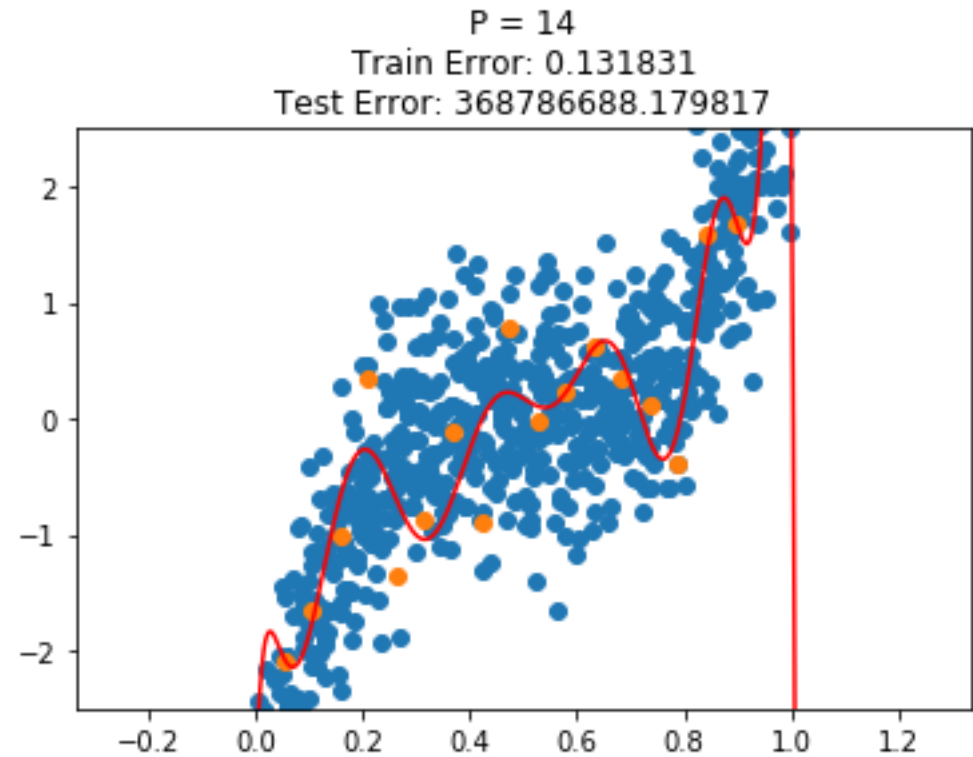
Train
Test



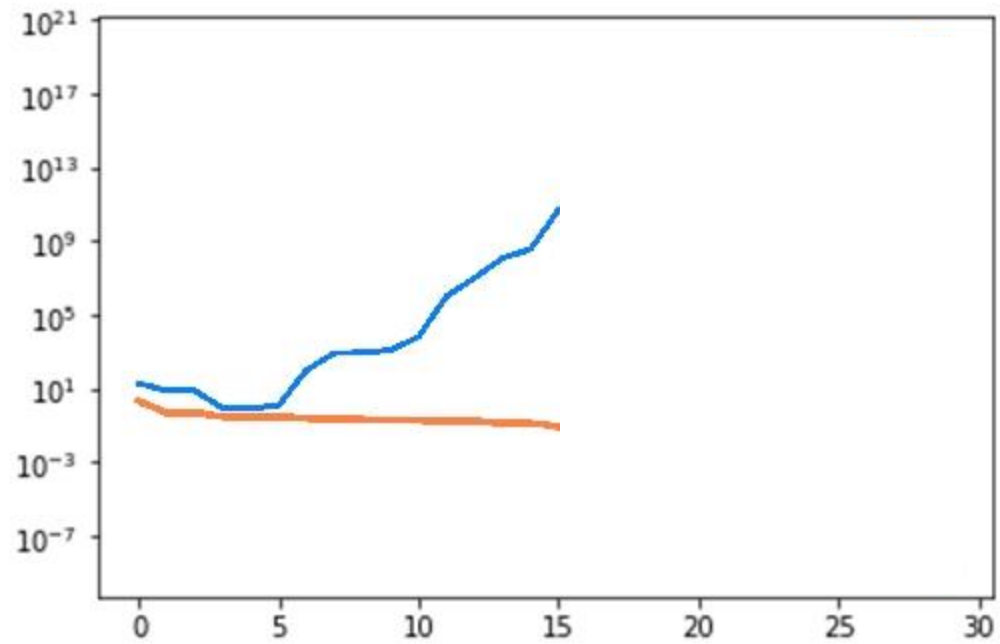
Polynomial fitting



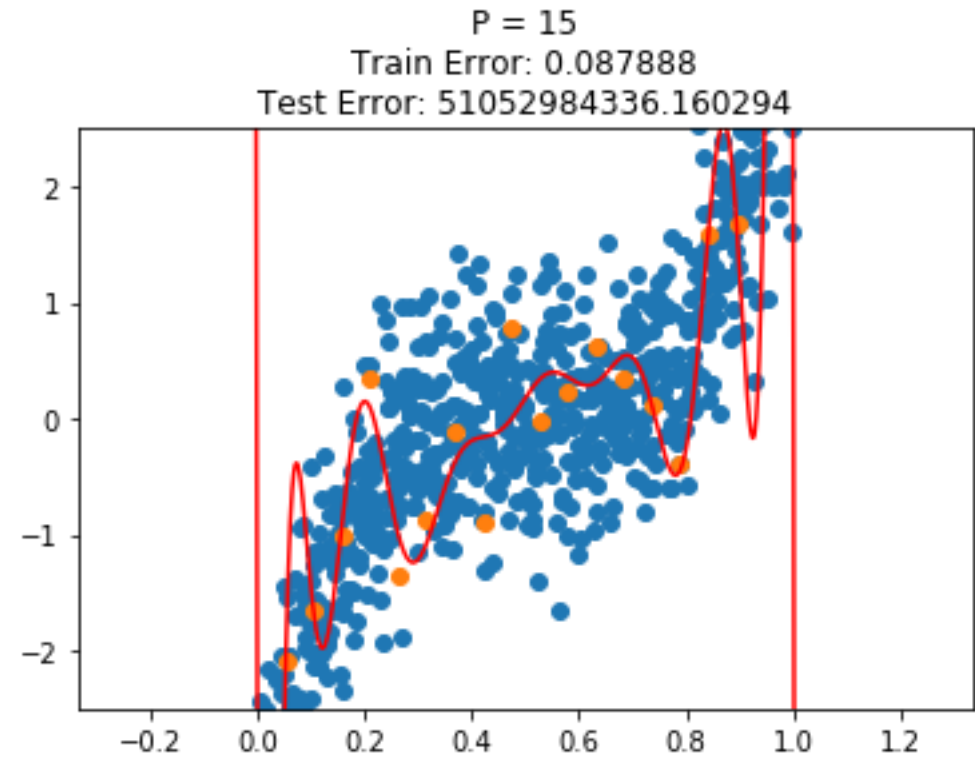
Train
Test



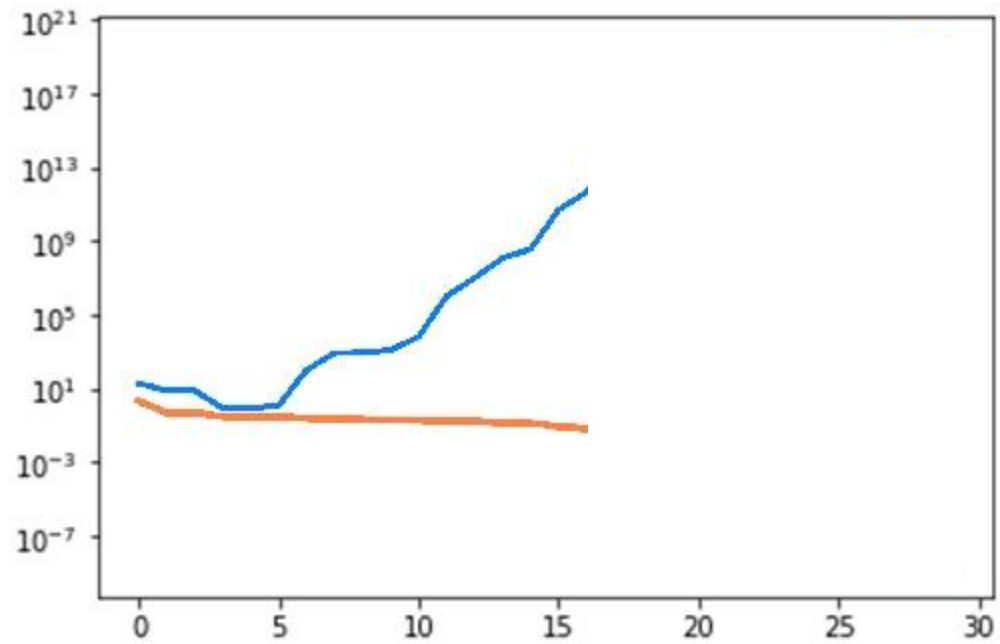
Polynomial fitting



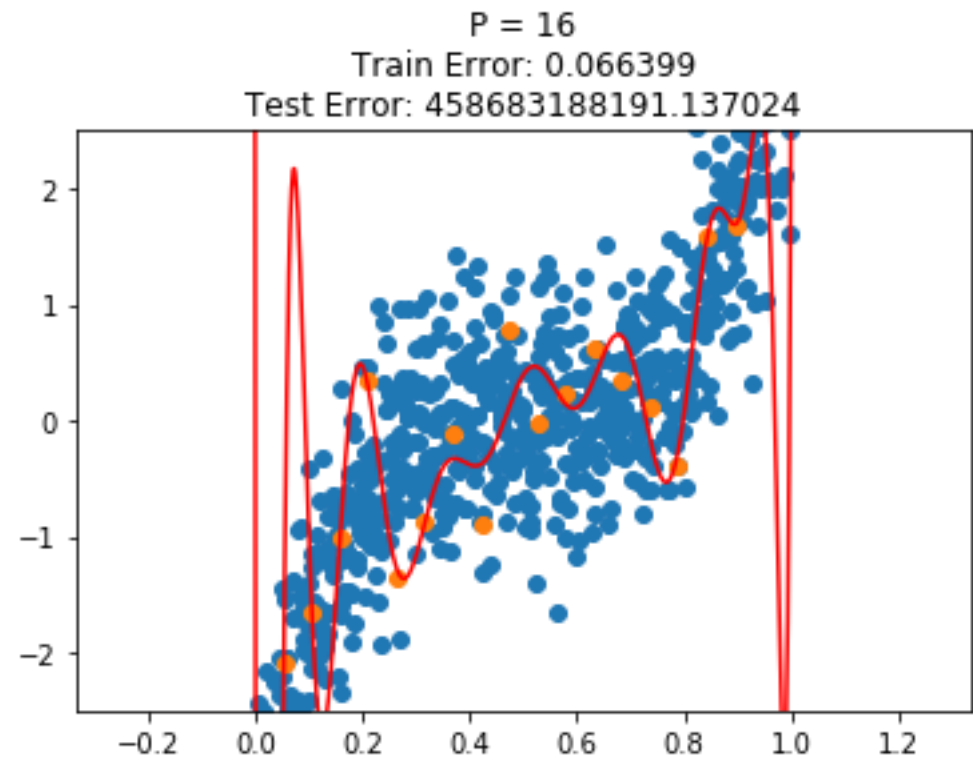
Train
Test



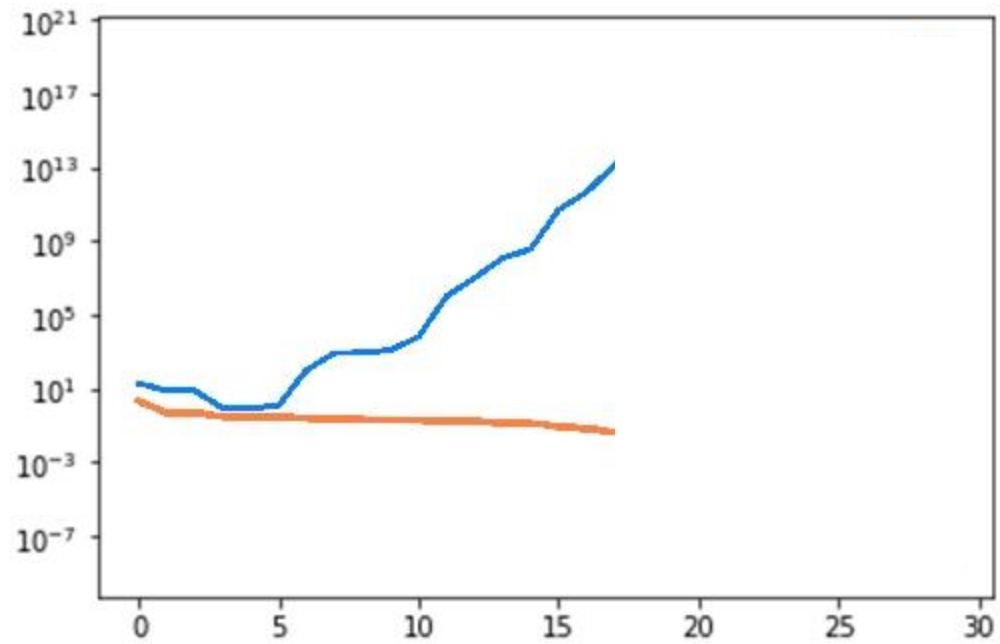
Polynomial fitting



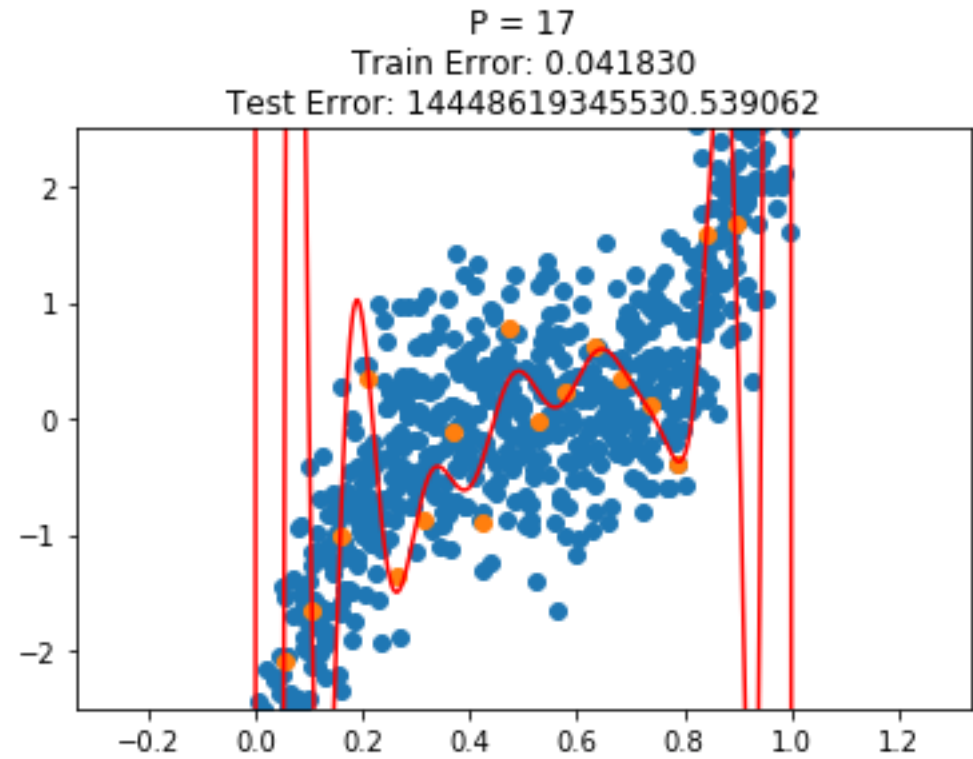
Train
Test



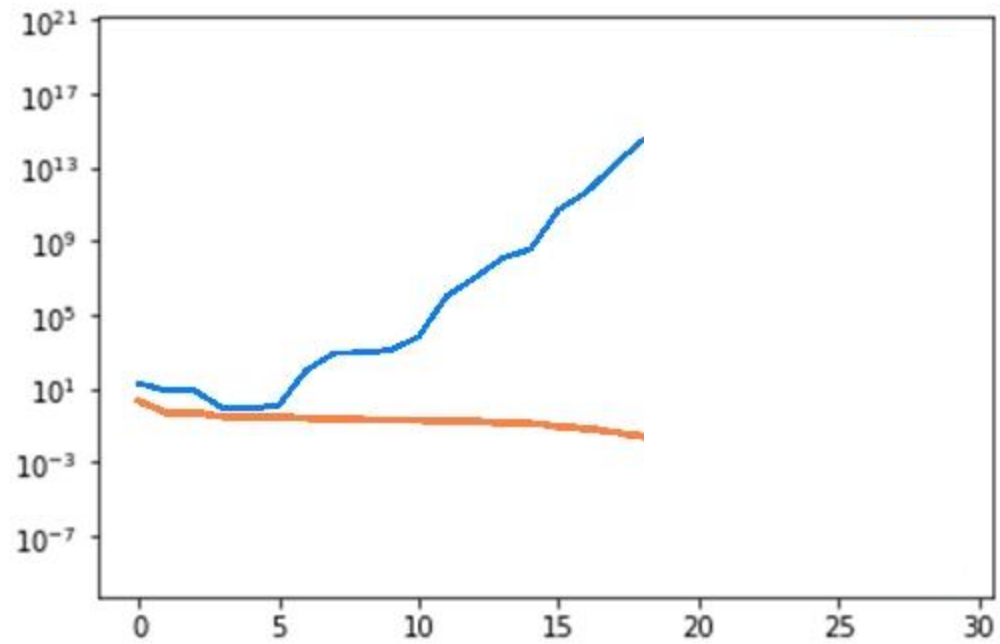
Polynomial fitting



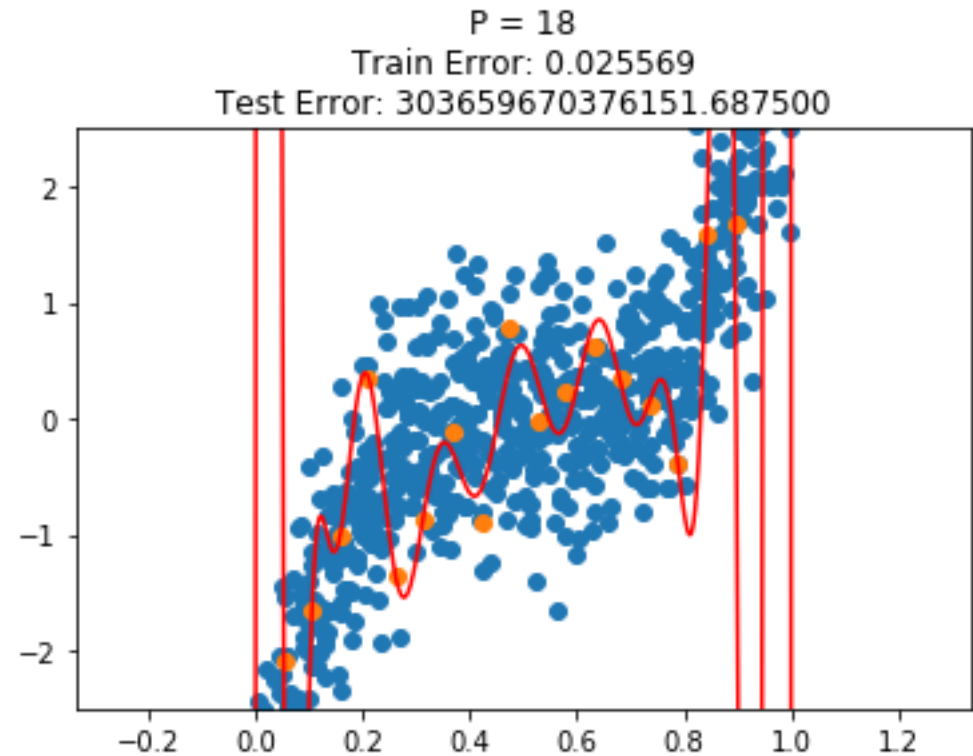
Train
Test



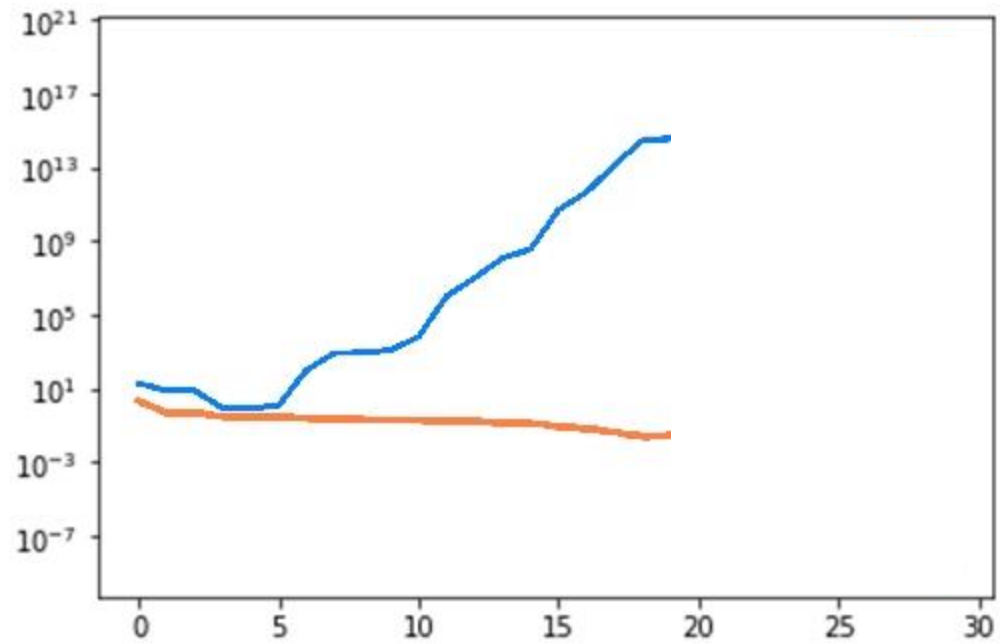
Polynomial fitting



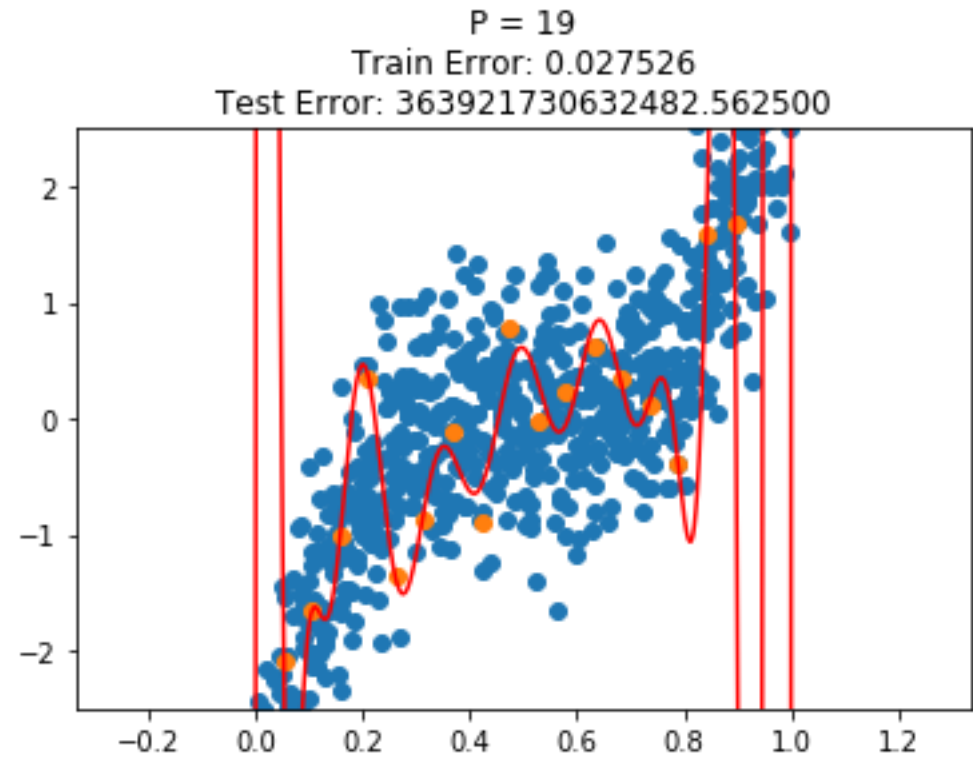
Train
Test



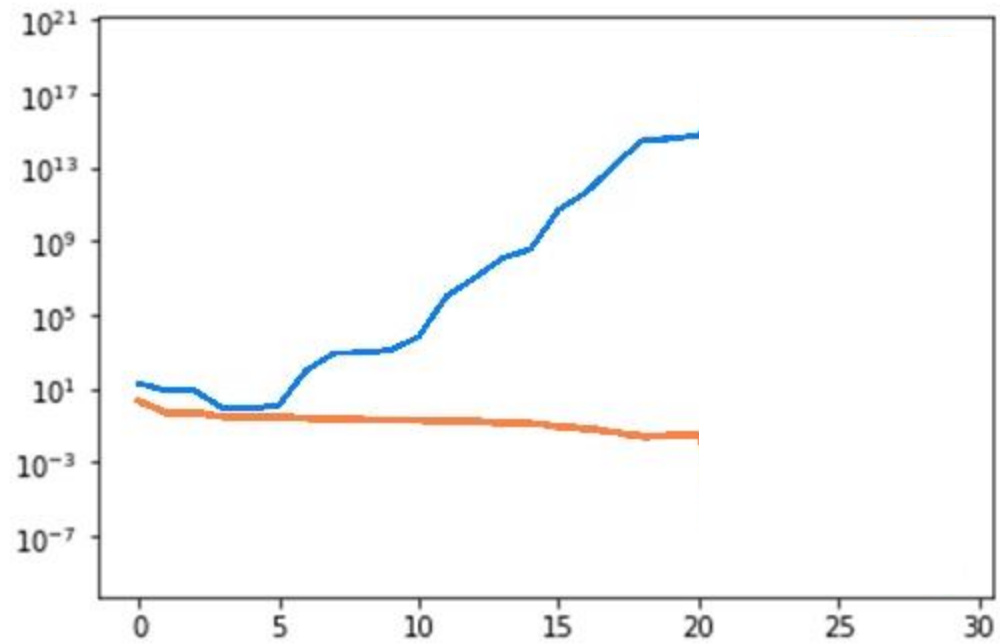
Polynomial fitting



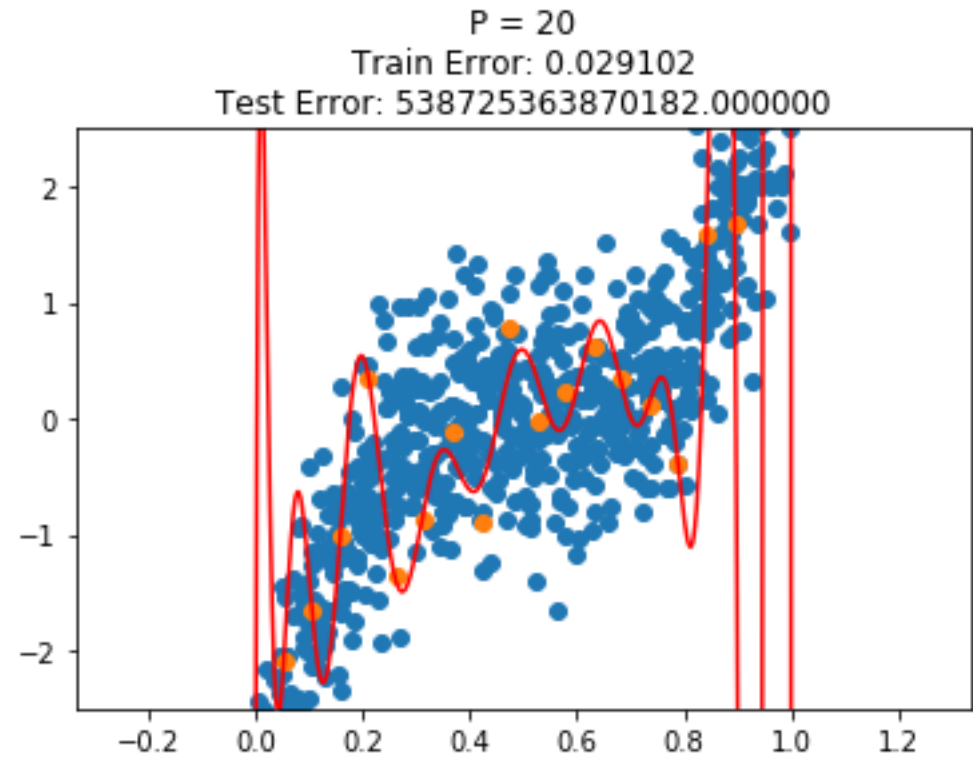
Train
Test



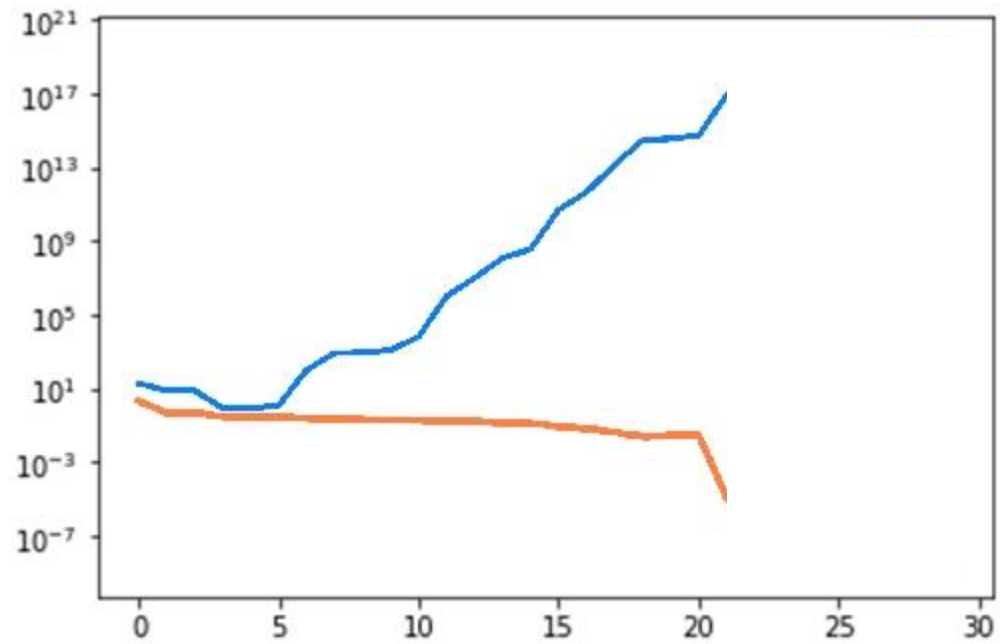
Polynomial fitting



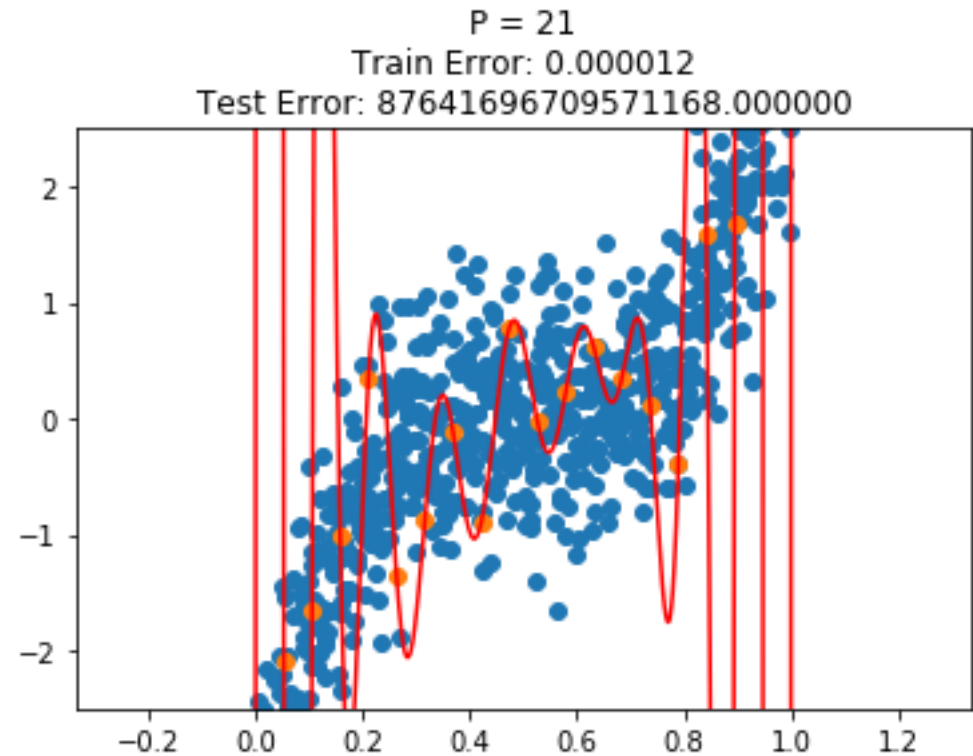
Train
Test



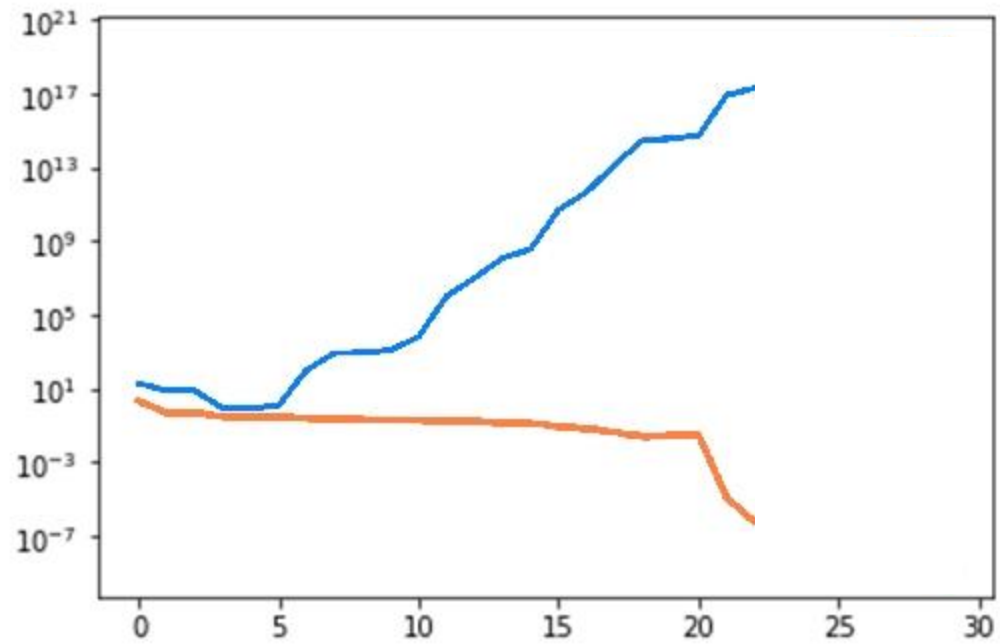
Polynomial fitting



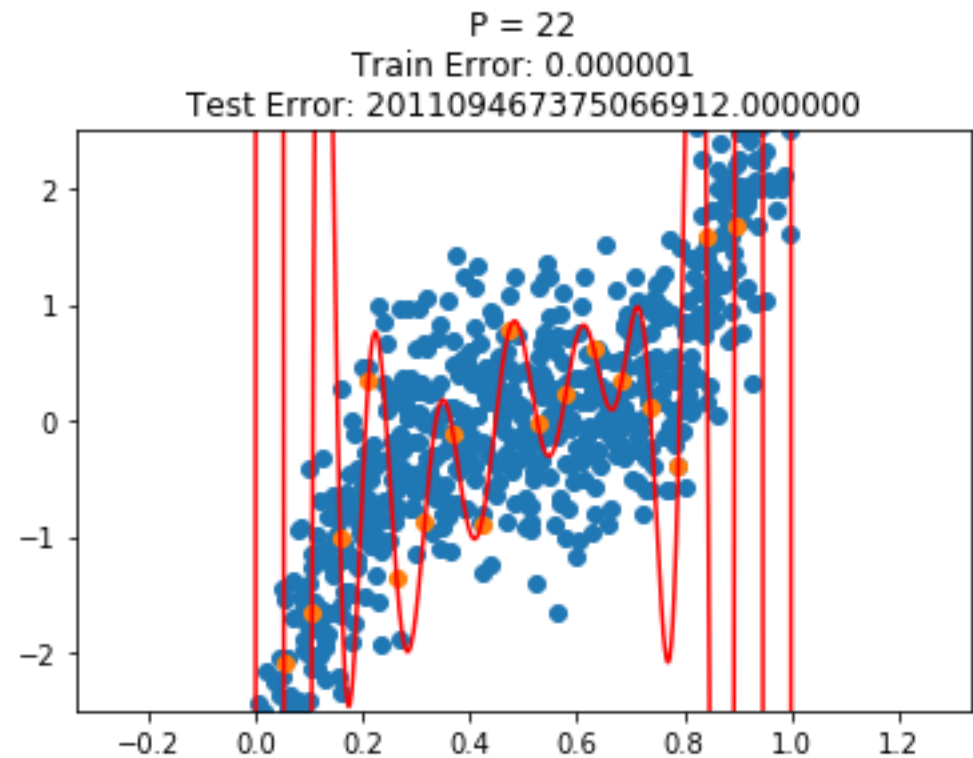
Train
Test



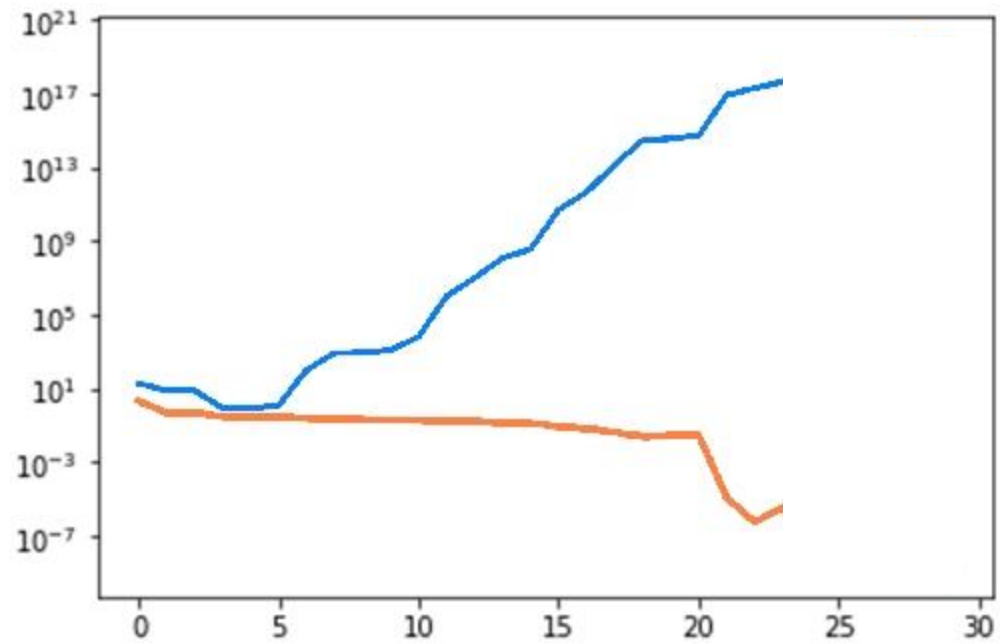
Polynomial fitting



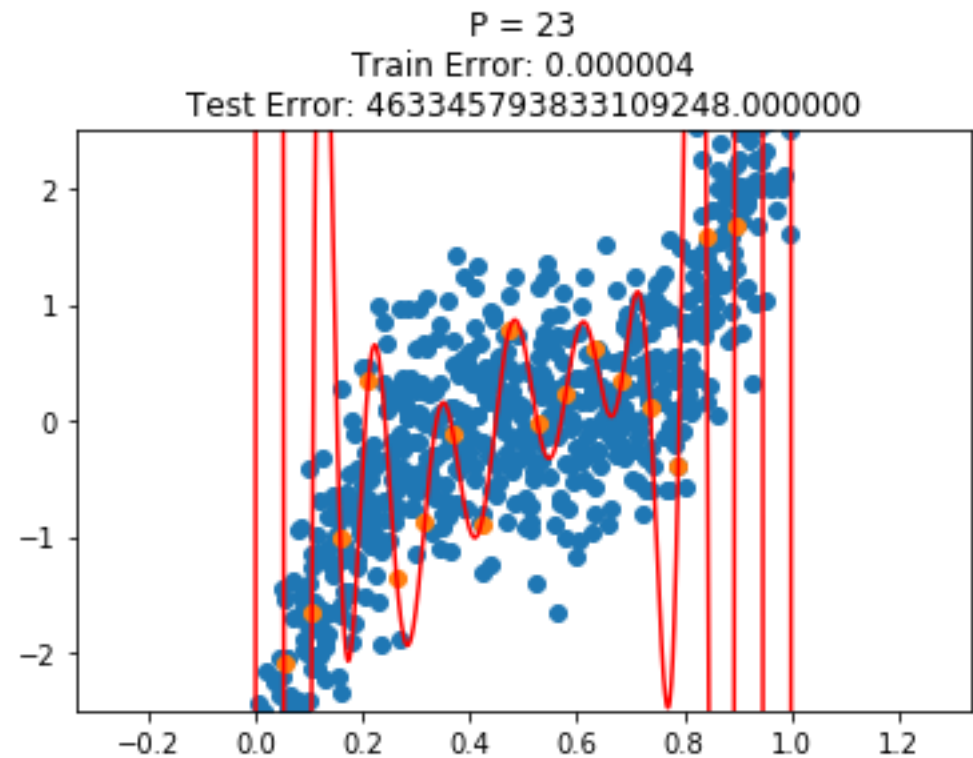
Train
Test



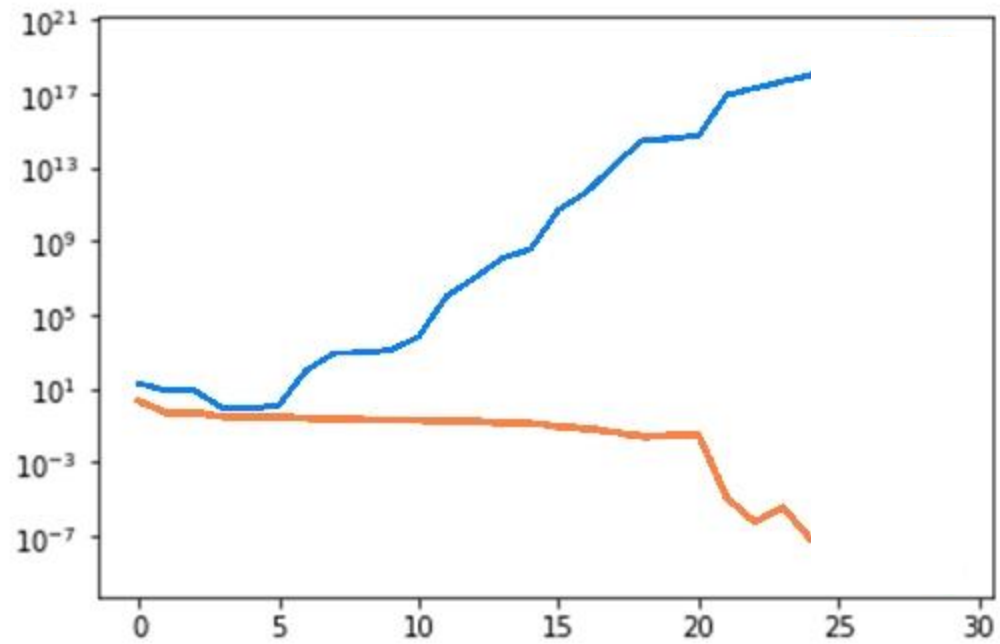
Polynomial fitting



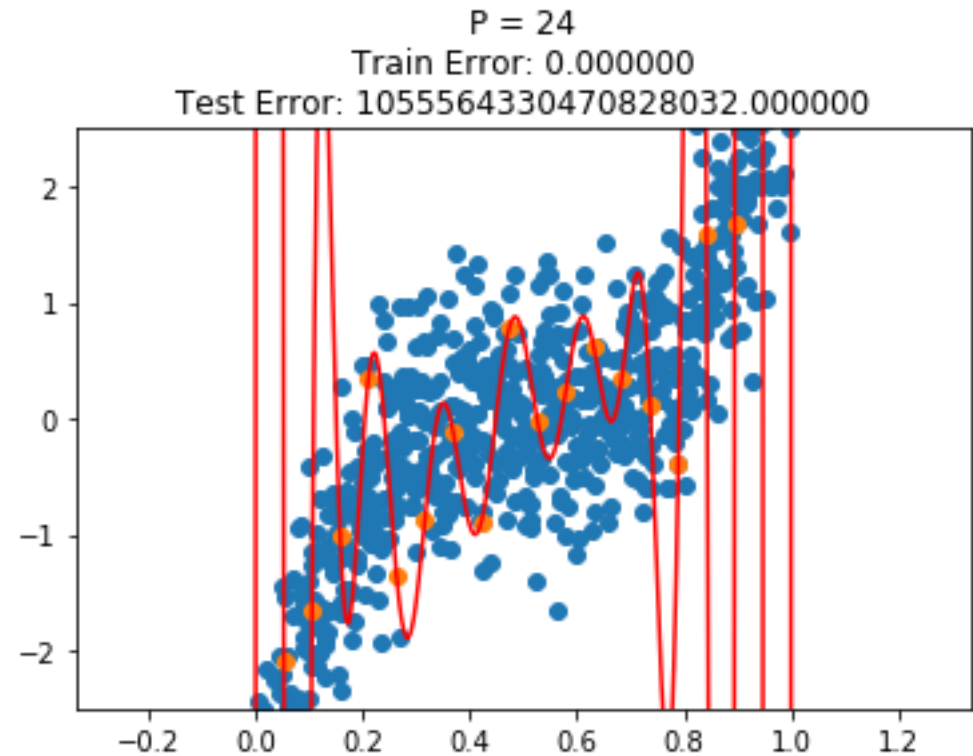
Train
Test



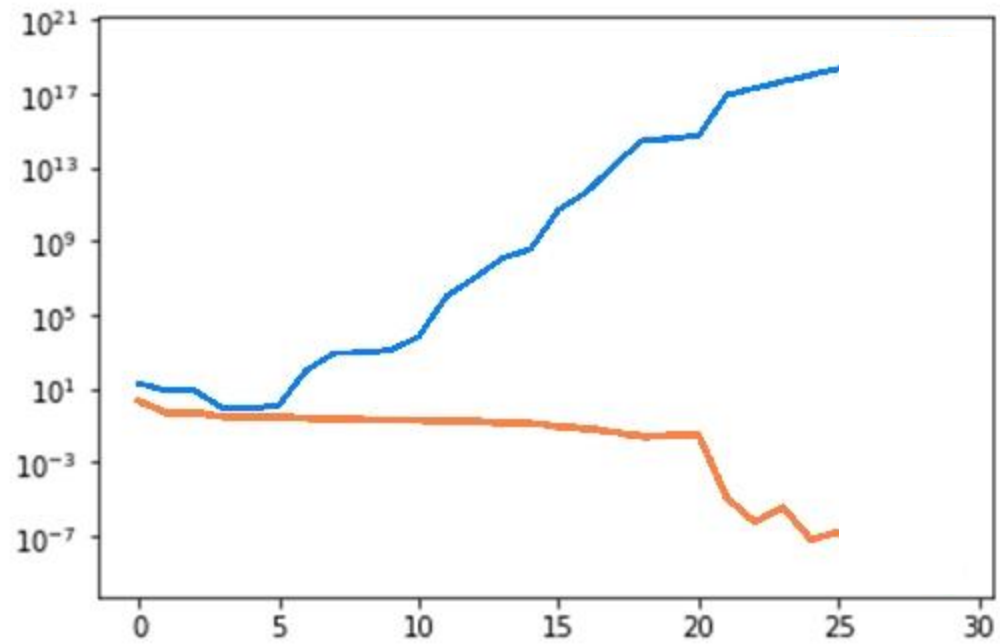
Polynomial fitting



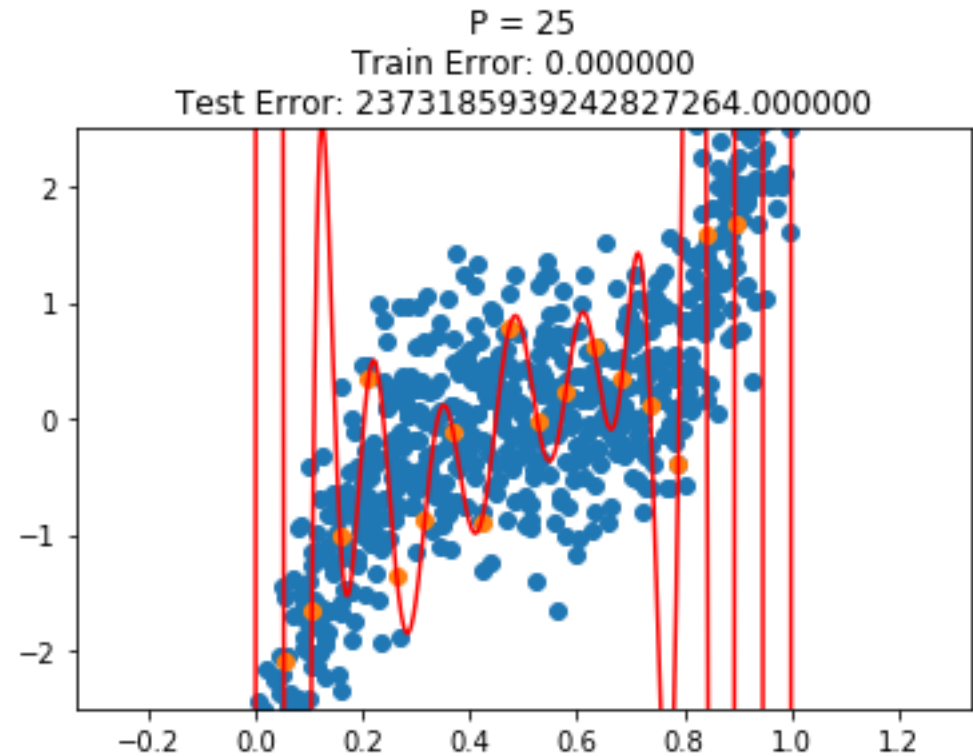
Train
Test



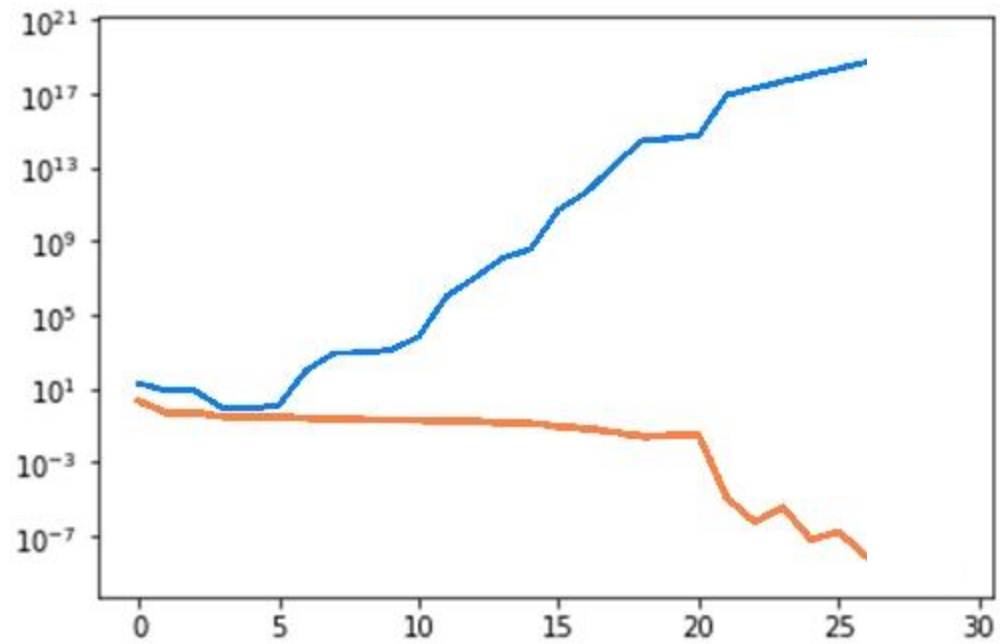
Polynomial fitting



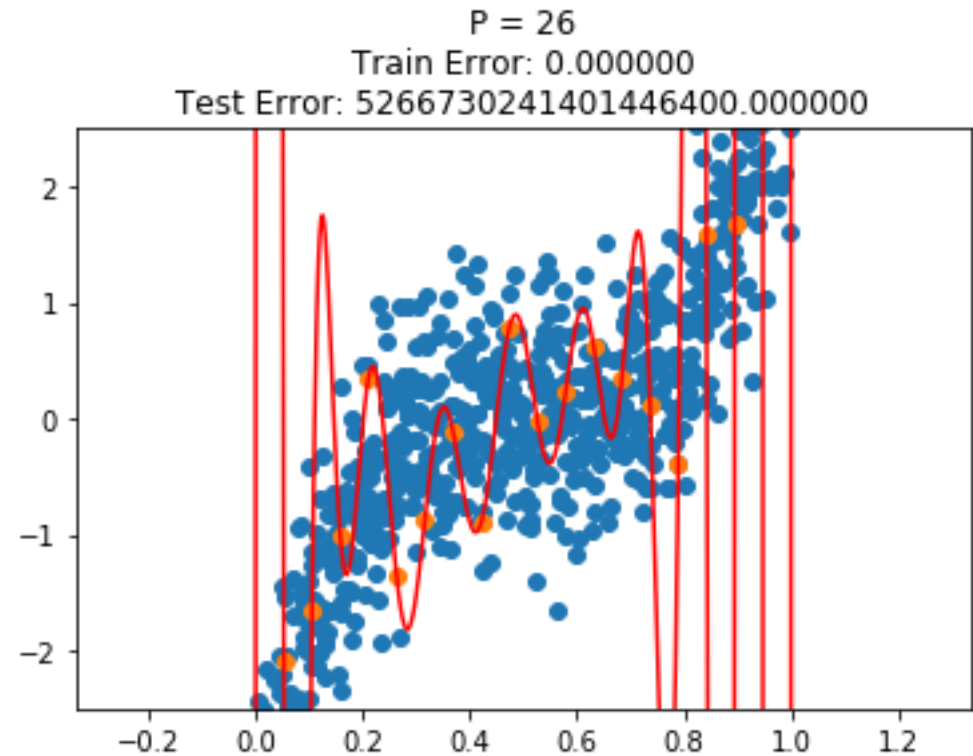
Train
Test



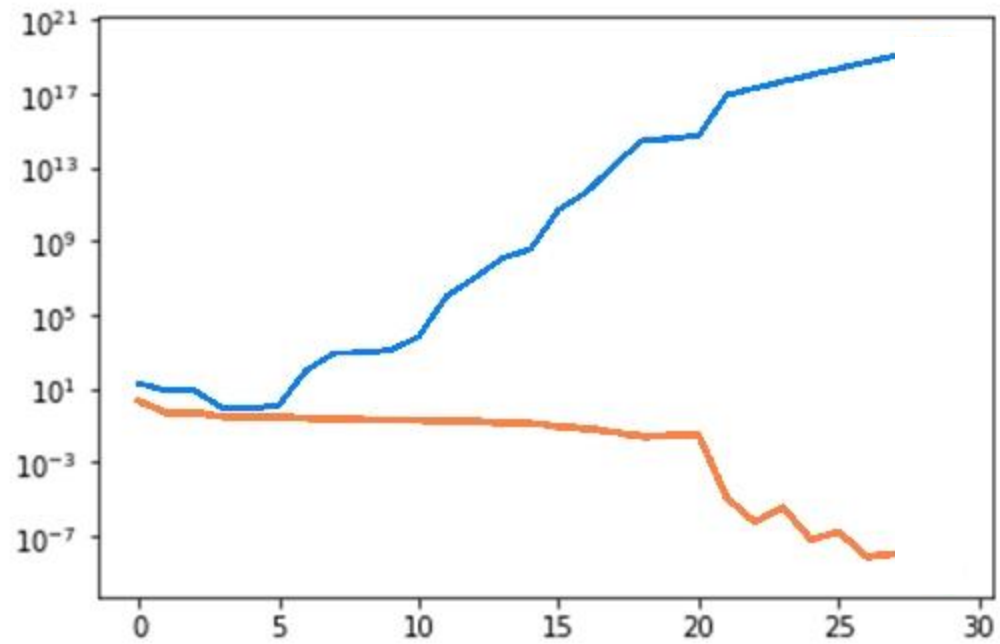
Polynomial fitting



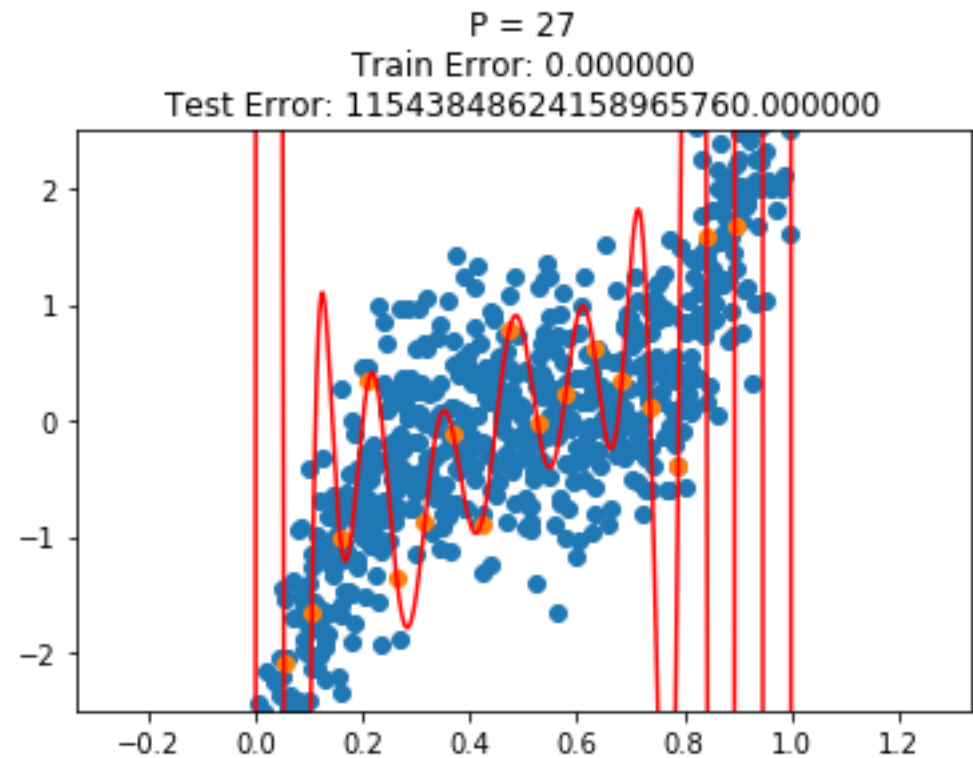
Train
Test



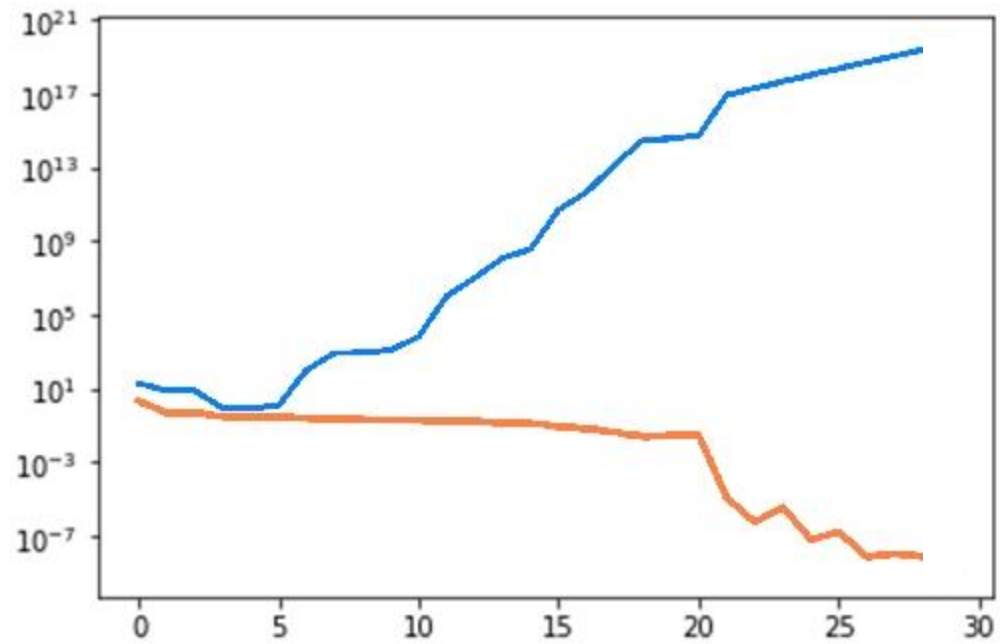
Polynomial fitting



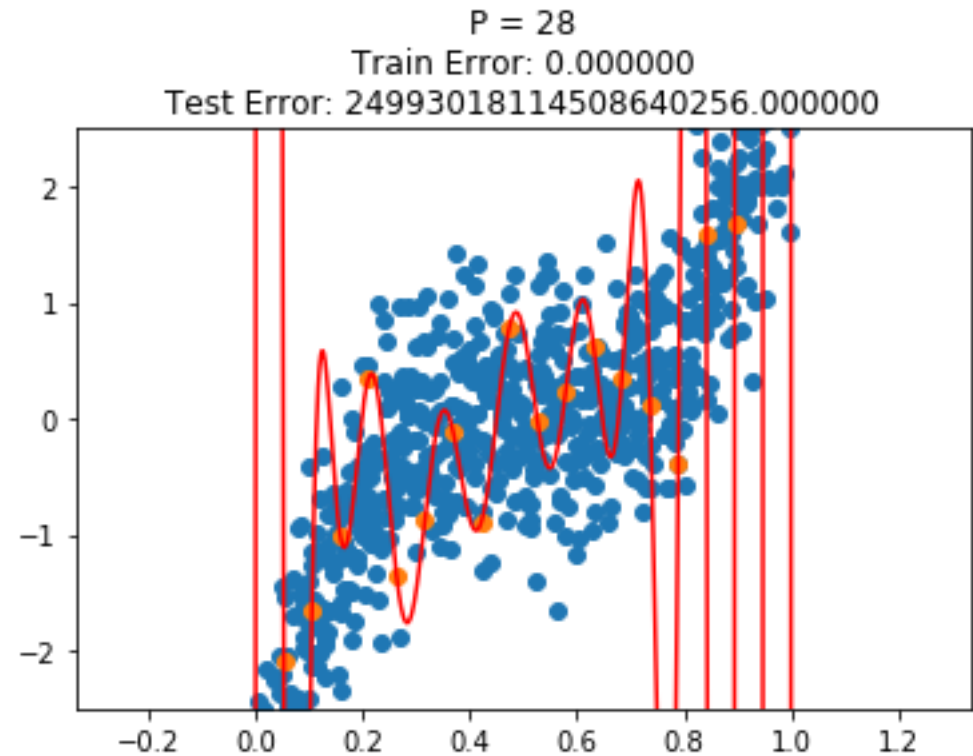
Train
Test



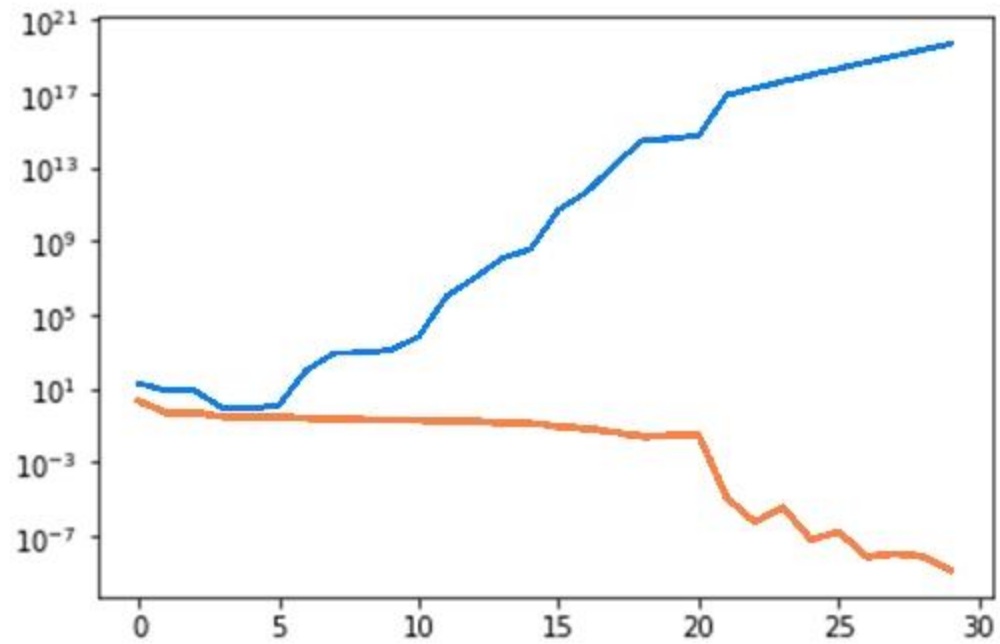
Polynomial fitting



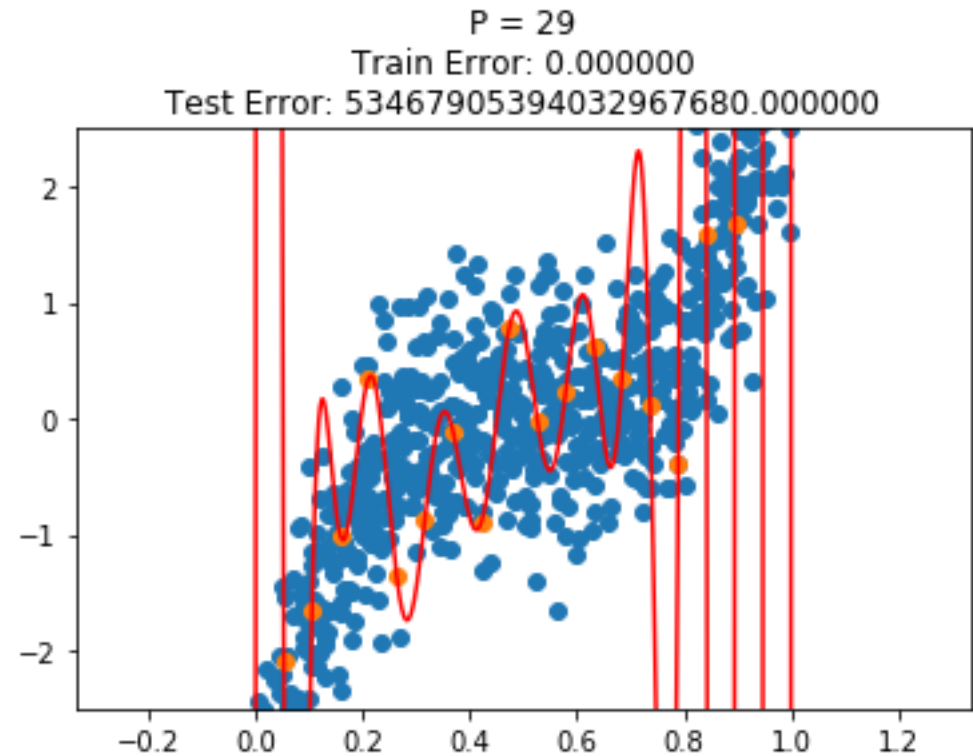
Train
Test



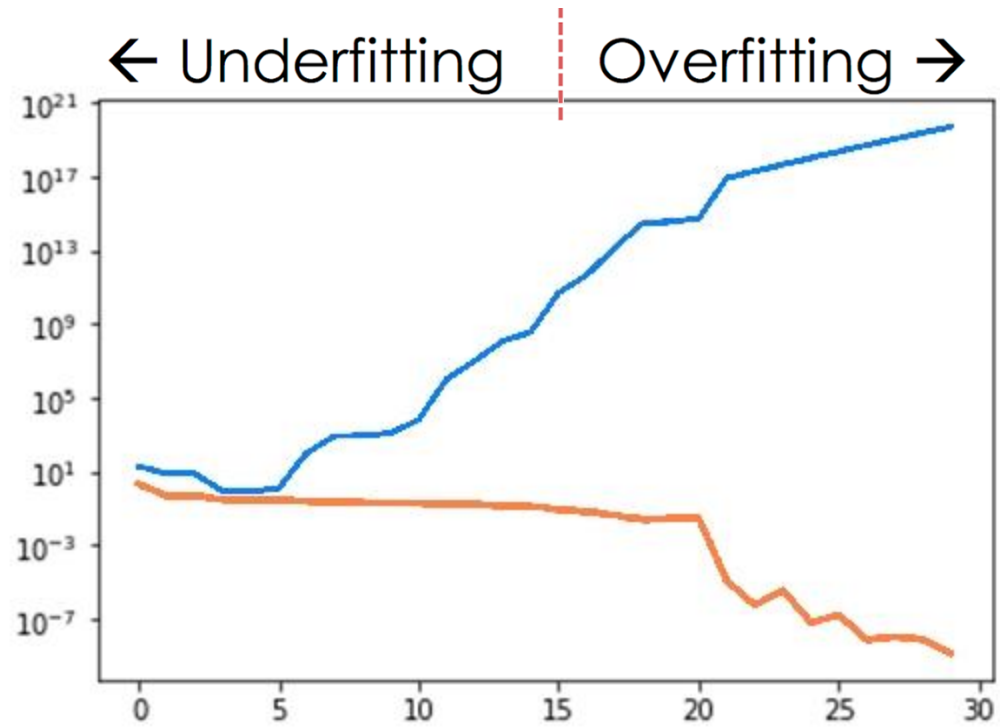
Polynomial fitting



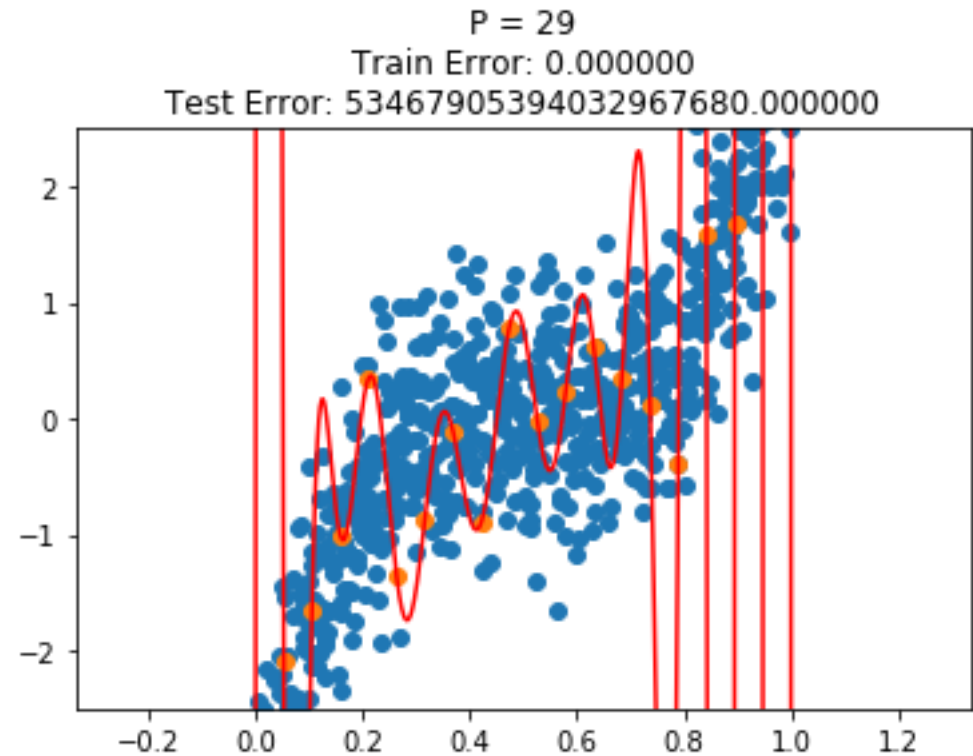
Train
Test



Polynomial fitting



Train
Test





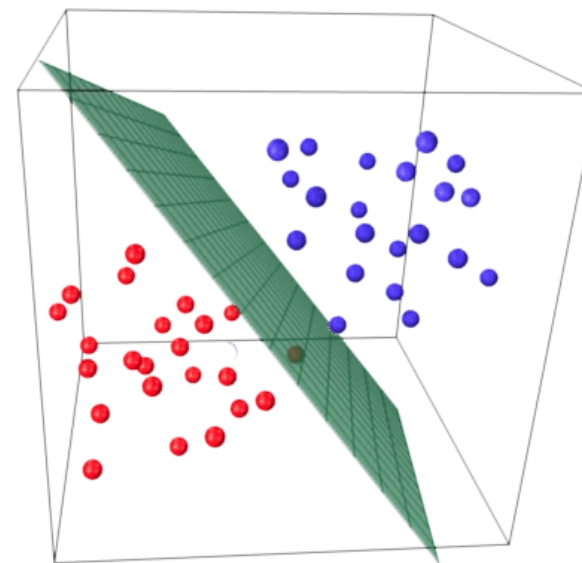
DL4CV@Weizmann

This week's tutorial:



Lior Yariv

Linear classification + some PyTorch

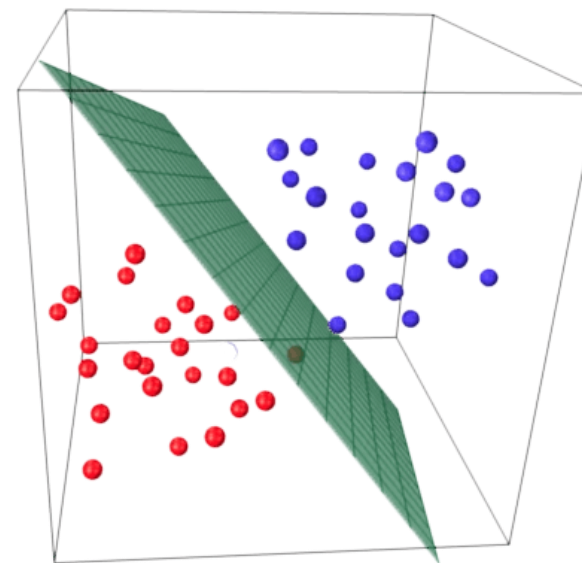


This week's tutorial:



Lior Yariv

Linear classification + some PyTorch



Next week's lecture:

(Me Again) **Neural Networks**

