CS-C3240 - Machine Learning

Round 1
Three Components of ML: Data, Model, and Loss

Lecture at 13.01.2021

Alexander Jung

Student Questions

- “I’m not sure if I understood the project requirements. On the lecture it was stated that we don’t need to code? It’s little bit confusing if no coding is required. Can the project be like using some algorithm(s) to predict the grade of tumour? For example can I do this project by implementing logistic regression to a dataset and predict the grade of tumour based on just 1 feature (like size) or has it to be more than 1 feature?”

Hints

- can be more than one correct answer in the quiz
- MUST assess example submissions in “Your ML Problem”
- submit “Your ML Problem" until Friday 23:59!
- you can change/modify your ML problem until Friday 23:59!
- student project can be different from “Your ML Problem”
- Quiz 1 is open until 22.01.2021

Three Main Components

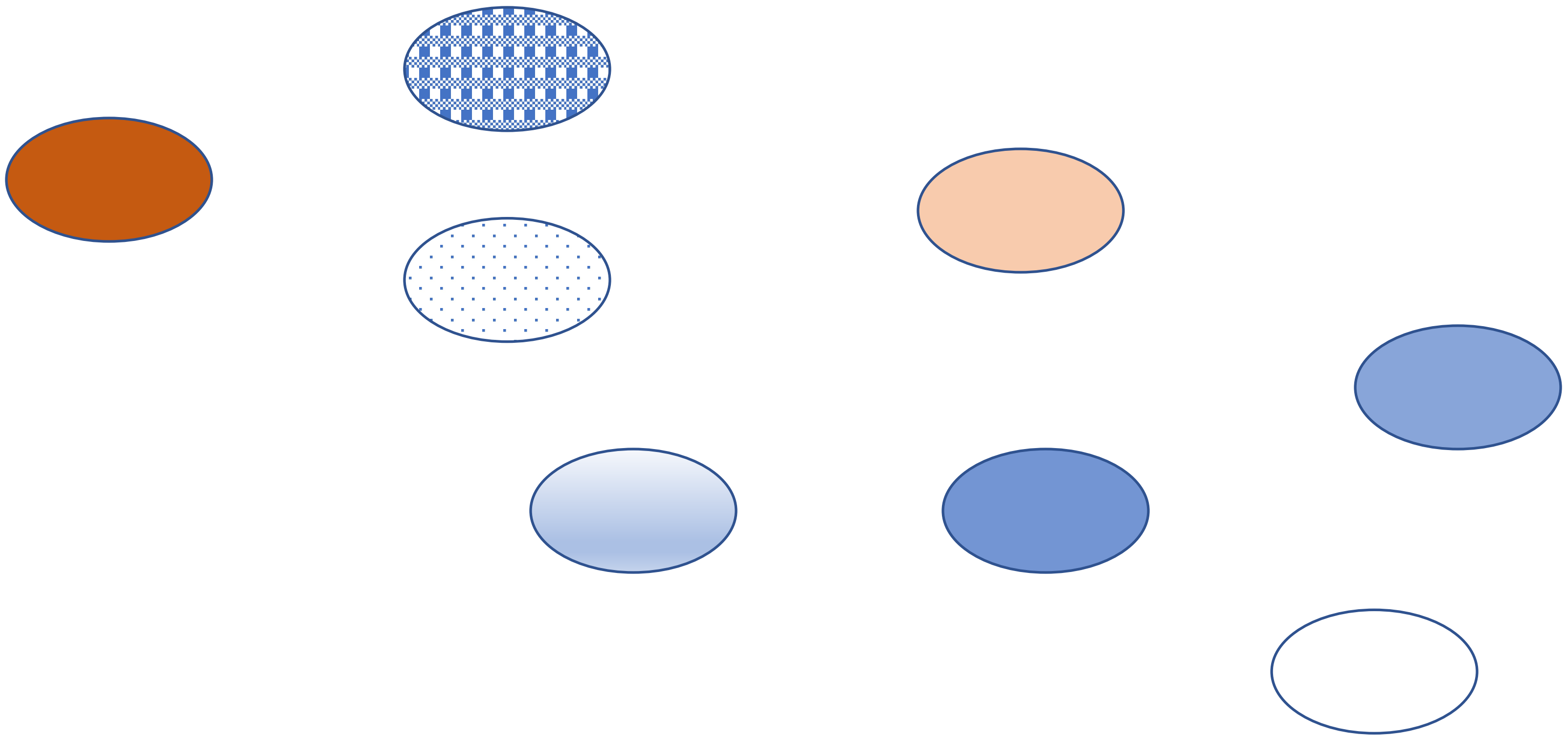
- data (observations)

- model (hypothesis space)

- loss function (performance measure)

Data

Dataset = (Large) Set of “Data Points”



data points are different objects but of similar “type”

Dataset – “Cows”




Dataset – “Forests”

Dataset = “Days During Pandemic”

1/Mar/2020

13/Mar/2020

2/Mar/2020

22/Apr/2020

1/Apr/2020

Data Point provides Atomic Unit of Information

- highly abstract concept
- data points can represent persons
- data points can represent random variables
- data points can represent machine learning problems

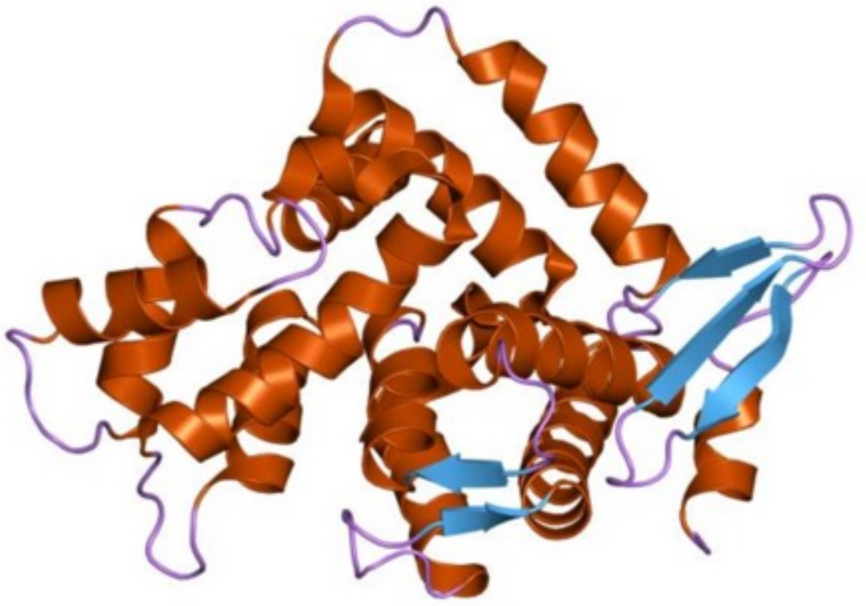
Features and Labels

- data points often have many different properties
- “features”: properties that can be measured/computed easily
- “labels”: properties that are “difficult” to compute
- labels are higher-level facts or quantities of interest
- labels can often be determined only in hindsight
- determining labels might require human (domain) experts

Data Point = “Some Protein”

features:

- protein structure
- physical measurements
- scientific papers about this protein



label:

should this protein be considered for a Covid-19 vaccine?

Data Point = "Some Plant"

features:
- plant species
- RGB image
- multi-spectral image
- ambient temperature

label:
does the plant need more water?

Features and Labels Are Design Choices!

design freedom for defining/choosing features and labels

labels could be defined by humans who provide labeled examples

labels could be subset of features

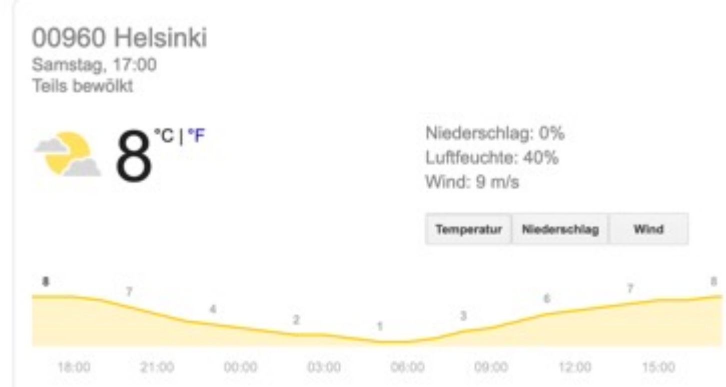
Supervised vs. Unsupervised

“supervised methods”: require labeled data points provided with the help of humans

distinction between supervised and unsupervised methods is blurry

Data Point = “Some Ski-Day Ahead”

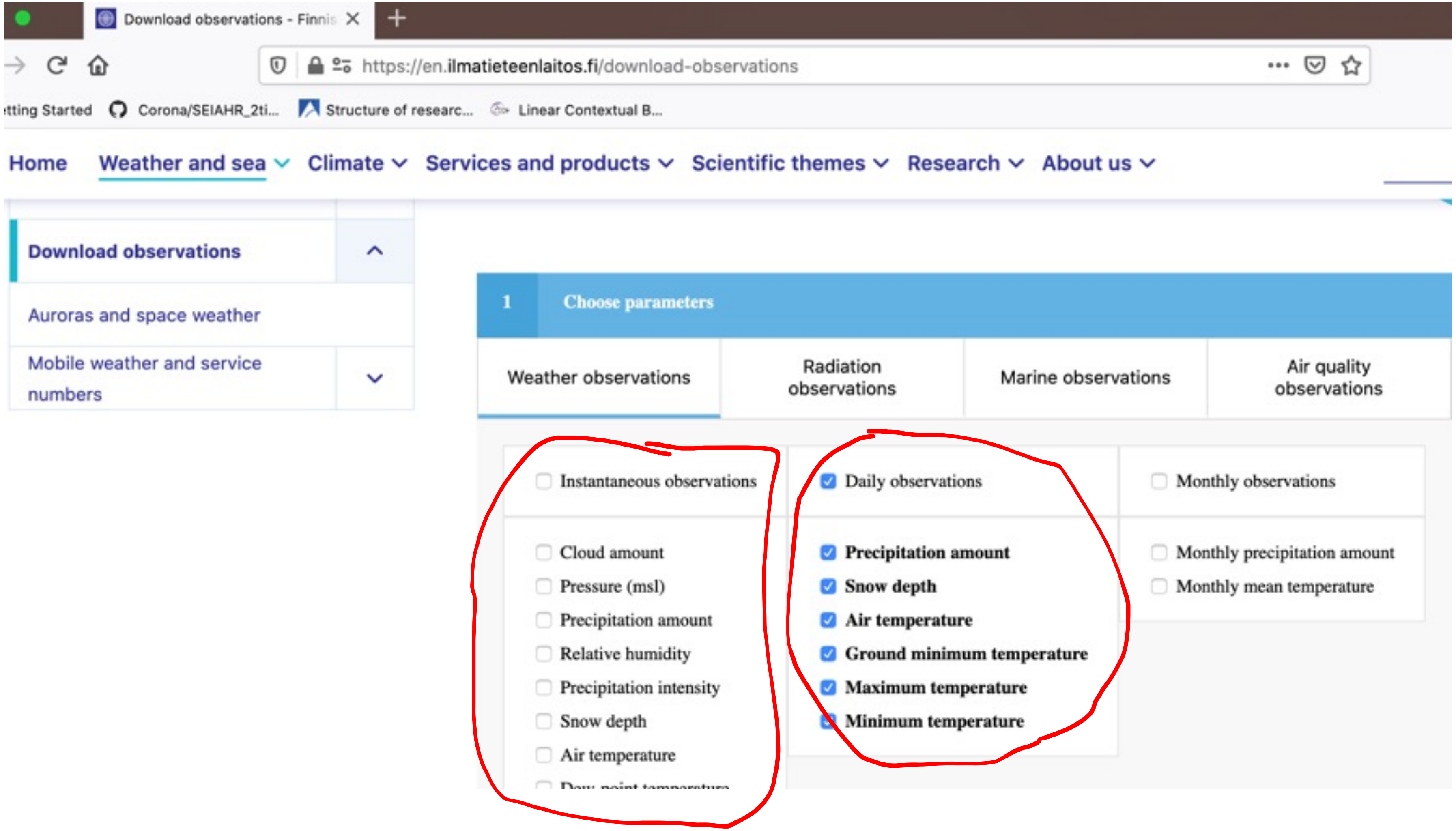
features:
- snapshot in the morning
- morning temperature
- weather history



label:
- maximum daytime temperature (important for ski waxing)

Let’s Get Some Data!

https://en.ilmatieteenlaitos.fi/open-data



features

label

Year	m	d	Time	precp	snow	airtmp	mintmp	maxtmp
2020	1	2	00:00	0,4	55	2,5	-2	4,5
2020	1	3	00:00	1,6	53	0,8	-0,8	4,6
2020	1	4	00:00	0,1	51	-5,8	-11,1	-0,7
2020	1	5	00:00	1,9	52	-13,5	-19,1	-4,6
2020	1	6	00:00	0,6	52	-2,4	-11,4	~~[illegible]~~ ?
2020	1	7	00:00	4,1	52	0,4	-2	1,3
2020	1	8	00:00	4,3	51	0,8	0,1	1,8
2020	1	9	00:00	-1	51	-0,6	-1,9	1,6
2020	1	10	00:00	-1	51	-6,2	-11	-1,4
2020	1	11	00:00	2,8	50	-4,8	-10,7	-2,1
2020	1	12	00:00	-1	53	-1,3	-3,5	0,9
2020	1	13	00:00	-1	53	-6,4	-12,9	-3,1
2020	1	14	00:00	9,7	52	-2,8	-9	-0,7
2020	1	15	00:00	-1	63	0,2	-0,7	0,6
2020	1	16	00:00	0,4	62	-3,9	-5,2	0,1
2020	1	17	00:00	2	62	-5,2	-8,4	-0,7
2020	1	18	00:00	19,6	65	-4,6	-7,3	-4,2
2020	1	19	00:00	0,7	81	-4,4	-8,8	-2,7
2020	1	20	00:00	2,8	79	-1,8	-10,5	1,2

data point

Key Parameters of a Data Set

number n of features

number m of data points "sample size"

Year	m	d	Time	precp	snow	airtmp	mintmp	maxtmp
2020	1	2	00:00	0,4	55	2,5	-2	4,5
2020	1	3	00:00	1,6	53	0,8	-0,8	4,6
2020	1	4	00:00	0,1	51	-5,8	-11,1	-0,7
2020	1	5	00:00	1,9	52	-13,5	-19,1	-4,6
2020	1	6	00:00	0,6	52	-2,4	-11,4	-1
2020	1	7	00:00	4,1	52	0,4	-2	1,3
2020	1	8	00:00	4,3	51	0,8	0,1	1,8
2020	1	9	00:00	-1	51	-0,6	-1,9	1,6
2020	1	10	00:00	-1	51	-6,2	-11	-1,4
2020	1	11	00:00	2,8	50	-4,8	-10,7	-2,1
2020	1	12	00:00	-1	53	-1,3	-3,5	0,9
2020	1	13	00:00	-1	53	-6,4	-12,9	-3,1
2020	1	14	00:00	9,7	52	-2,8	-9	-0,7
2020	1	15	00:00	-1	63	0,2	-0,7	0,6
2020	1	16	00:00	0,4	62	-3,9	-5,2	0,1
2020	1	17	00:00	2	62	-5,2	-8,4	-0,7
2020	1	18	00:00	19,6	65	-4,6	-7,3	-4,2
2020	1	19	00:00	0,7	81	-4,4	-8,8	-2,7
2020	1	20	00:00	2,8	79	-1,8	-10,5	1,2

Alex’s Rule of Thumb

try to use a sample size m which is 10 times the number n of features

$$m >= 10 * n$$

High-Dimensional Data

- can often measure tons of features of data point
- number features n much much larger than m
- for n >> m ML methods tend to overfit (Round 4!)
- cleverly select subset of raw features (Round 6!)

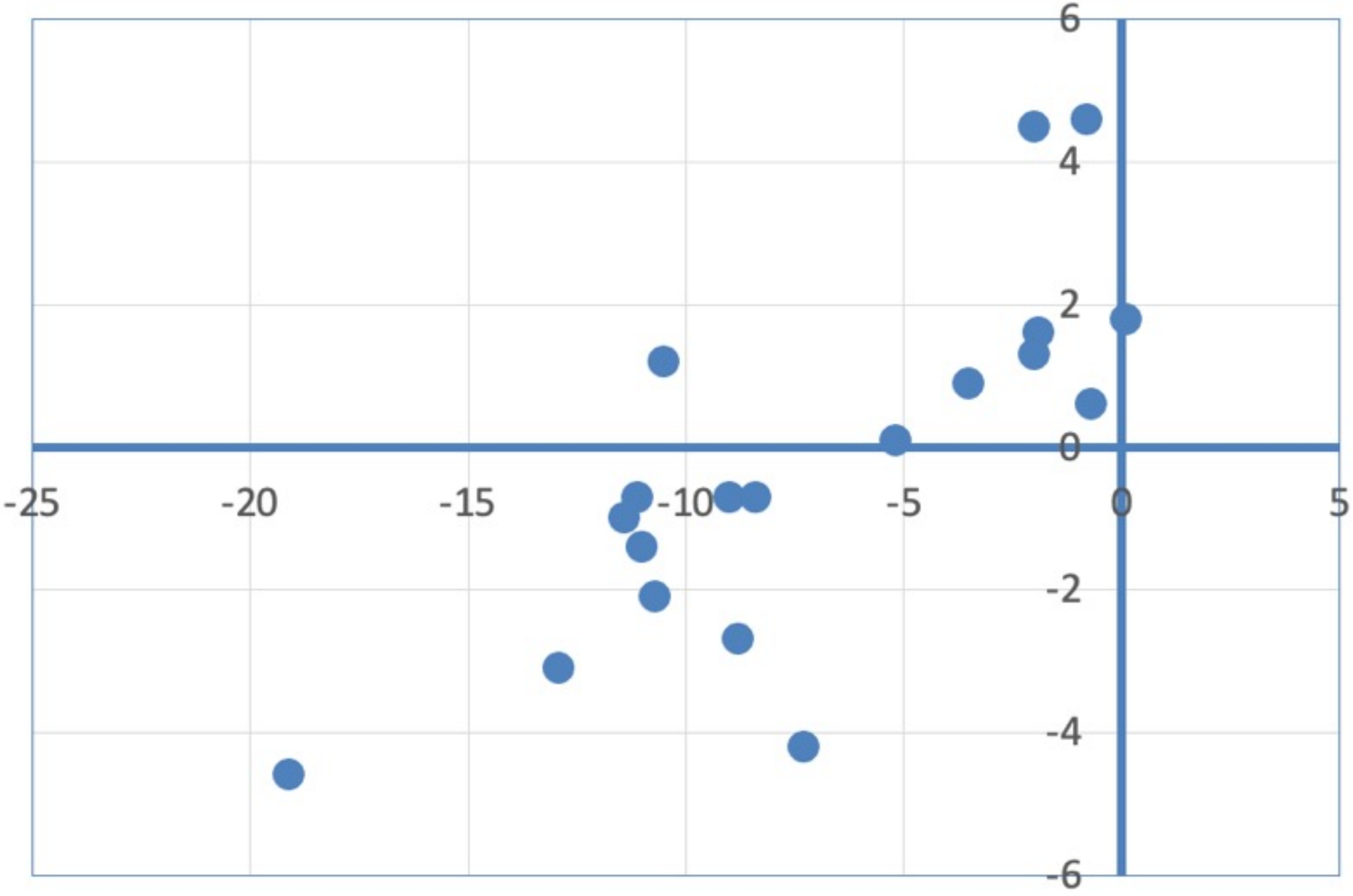
feature label

Year	m	d	Time	precp	snow	airtmp	mintmp	maxtmp
2020	1	2	00:00	0,4	55	2,5	-2	4,5
2020	1	3	00:00	1,6	53	0,8	-0,8	4,6
2020	1	4	00:00	0,1	51	-5,8	-11,1	-0,7
2020	1	5	00:00	1,9	52	-13,5	-19,1	-4,6
2020	1	6	00:00	0,6	52	-2,4	-11,4	-1
2020	1	7	00:00	4,1	52	0,4	-2	1,3
2020	1	8	00:00	4,3	51	0,8	0,1	1,8
2020	1	9	00:00	-1	51	-0,6	-1,9	1,6
2020	1	10	00:00	-1	51	-6,2	-11	-1,4
2020	1	11	00:00	2,8	50	-4,8	-10,7	-2,1
2020	1	12	00:00	-1	53	-1,3	-3,5	0,9
2020	1	13	00:00	-1	53	-6,4	-12,9	-3,1
2020	1	14	00:00	9,7	52	-2,8	-9	-0,7
2020	1	15	00:00	-1	63	0,2	-0,7	0,6
2020	1	16	00:00	0,4	62	-3,9	-5,2	0,1
2020	1	17	00:00	2	62	-5,2	-8,4	-0,7
2020	1	18	00:00	19,6	65	-4,6	-7,3	-4,2
2020	1	19	00:00	0,7	81	-4,4	-8,8	-2,7
2020	1	20	00:00	2,8	79	-1,8	-10,5	1,2

data point

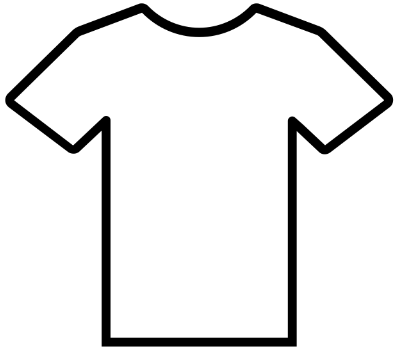
First Look at Data - Scatterplot

label y (max tmp)



feature x
(min tmp)

Model
(Hypothesis Space)



morning temperature = - 10 (minimum daytime temp.)
maximum daytime temperature ?

Hypothesis or Predictor Map

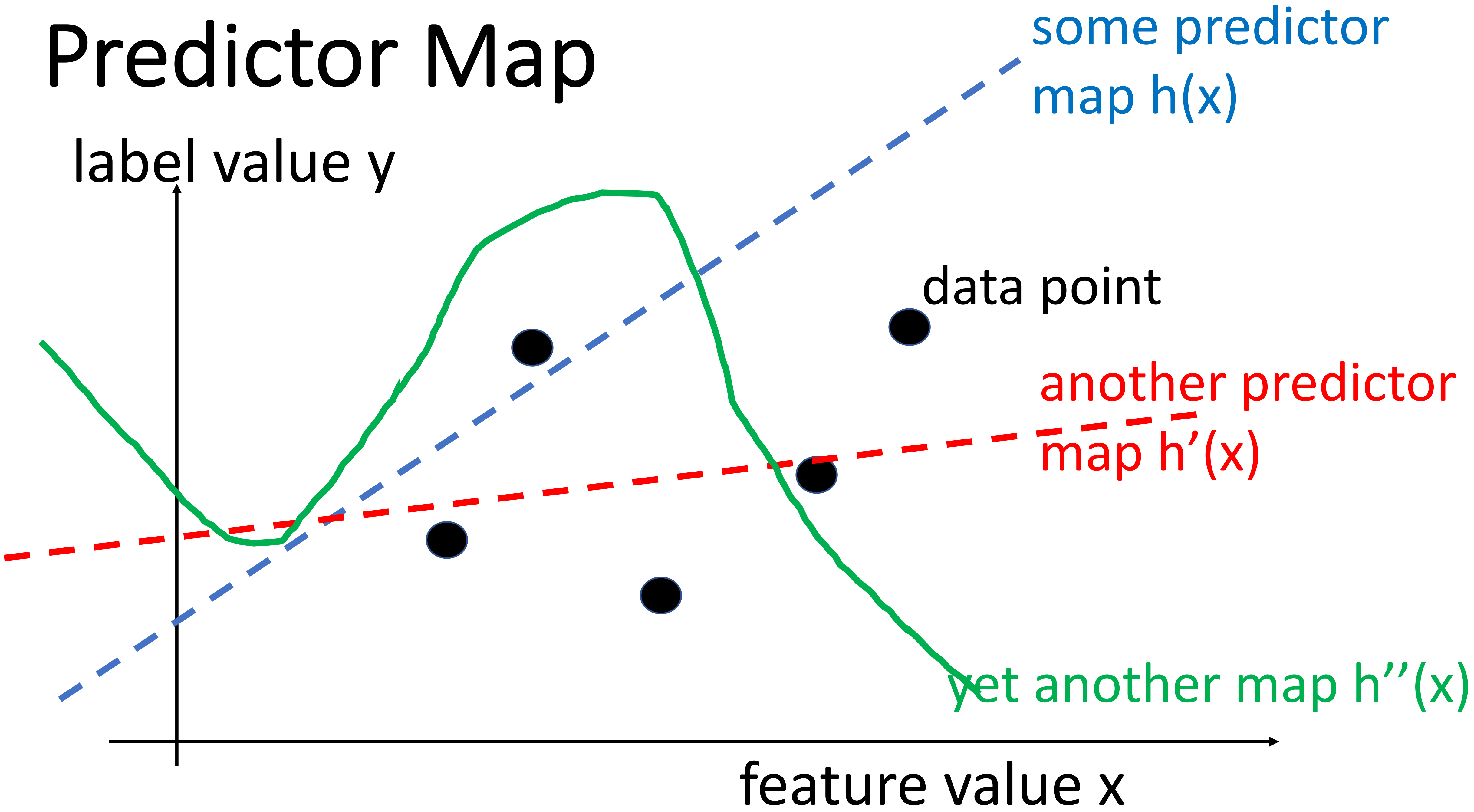
h(x)

$$\hat{y} = h(x) = 2$$

estimate for the label y (max. daytime tmp)

feature x = -10 minimum daytime temperature

Predictor Map



Machine Learning

$\approx$

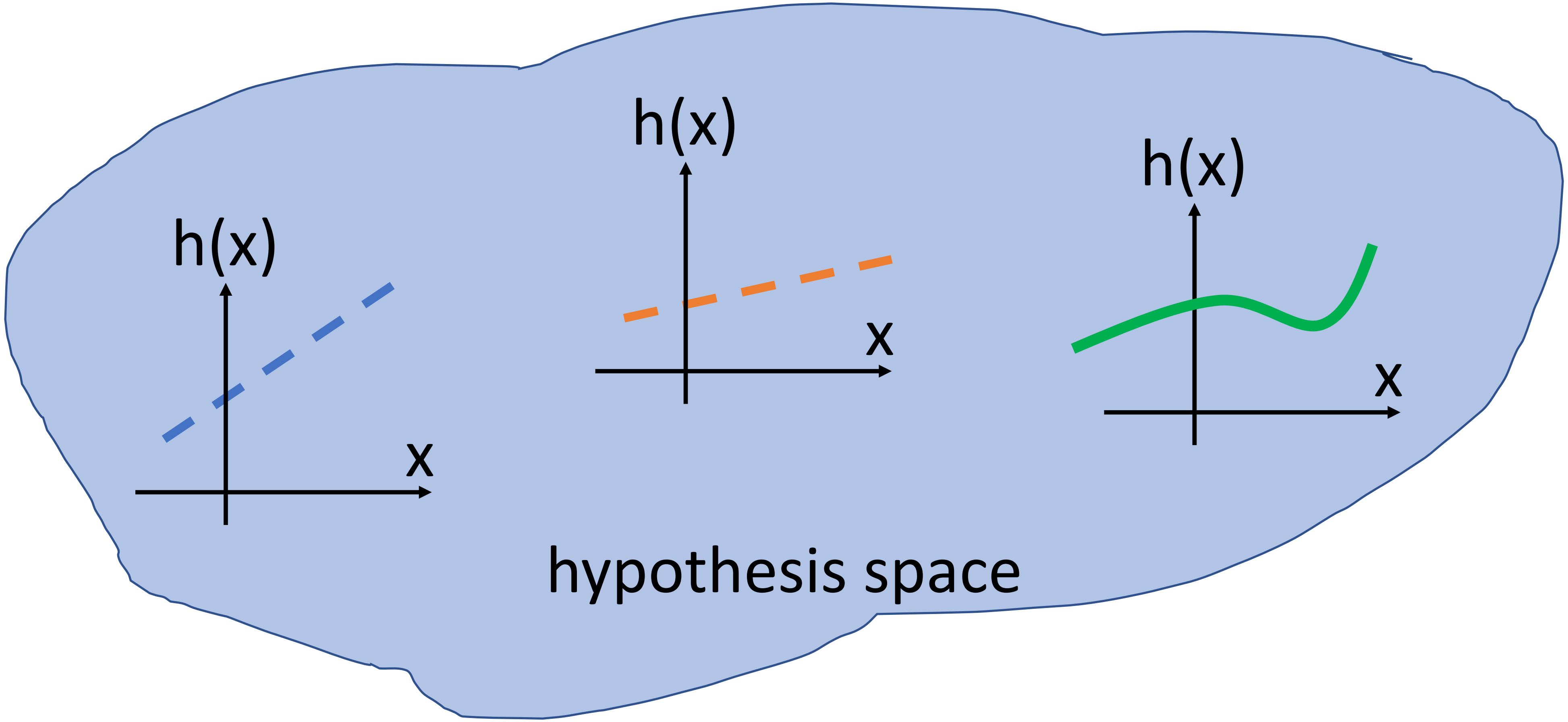
Find Good Predictor Map

data point with single numeric feature x (min tmp) and label y (max tmp)

how many predictor maps h(x) are there ?

have only finite computation time, memory, bandwidth !

a hypothesis space is a computationally tractable subset of predictor maps



machine and deep learning Python libraries provide “fit()” function to search over (huge) hypothesis spaces

```python
# define some hypothesis "hypothesis3"
def hypothesis3(x):
    # perform computation
    tmp = sin(x)
    tmp1 = tmp +cos(x)*4
    tmp1 = abs(tmp1)
    tmp = log(tmp+5)
    hat_y = tmp*tmp1
    return hat_y
```

```python
# define some hypothesis "hypothesis2"
def hypothesis2(x):
    # perform computation
    tmp = sin(x)
    tmp1 = tmp +cos(x)*4
    tmp1 = abs(tmp1)
    hat_y = tmp*tmp1
    return hat_y
```

```python
# define some hypothesis "hypothesis1"
def hypothesis1(x):
    # perform computation
    tmp = x*3
    tmp1 = tmp +x*4
    hat_y = tmp/tmp1
    return hat_y
```

hypothesis space

Two Hypotheses Make a Hypothesis Space

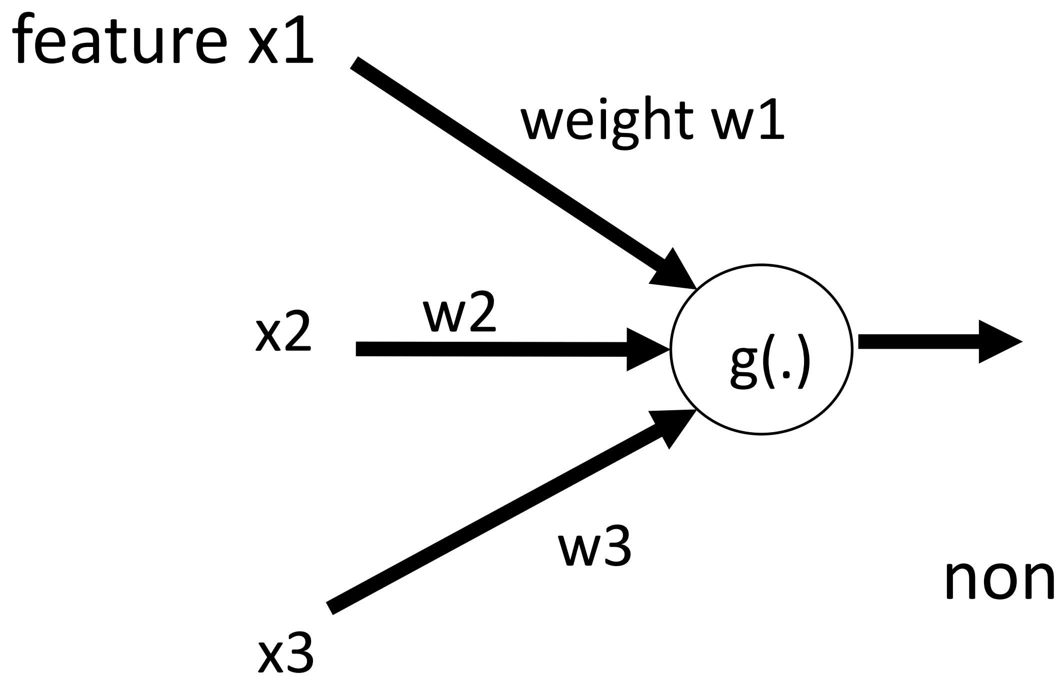
x	h1	h2
-2	1	3
-0,8	2,2	4,2
-11,1	-8,1	-6,1
-19,1	-16	-14
-11,4	-8,4	-6,4
-2	1	3
0,1	3,1	5,1
-1,9	1,1	3,1
-11	-8	-6
-10,7	-7,7	-5,7
-3,5	-0,5	1,5
-12,9	-9,9	-7,9
-9	-6	-4
-0,7	2,3	4,3
-5,2	-2,2	-0,2
-8,4	-5,4	-3,4
-7,3	-4,3	-2,3
-8,8	-5,8	-3,8
-10,5	-7,5	-5,5

Machine Learning

- ML aims at finding/learning a good predictor h(x)
- predictor reads in features x and outputs predicted label
- predictor maps reading in millions of features
- must choose between many different predictor maps
- deep learning uses smart representation for maps

Artificial Neural Networks

- represent predictor map h(x) using network of neurons
- single neuron



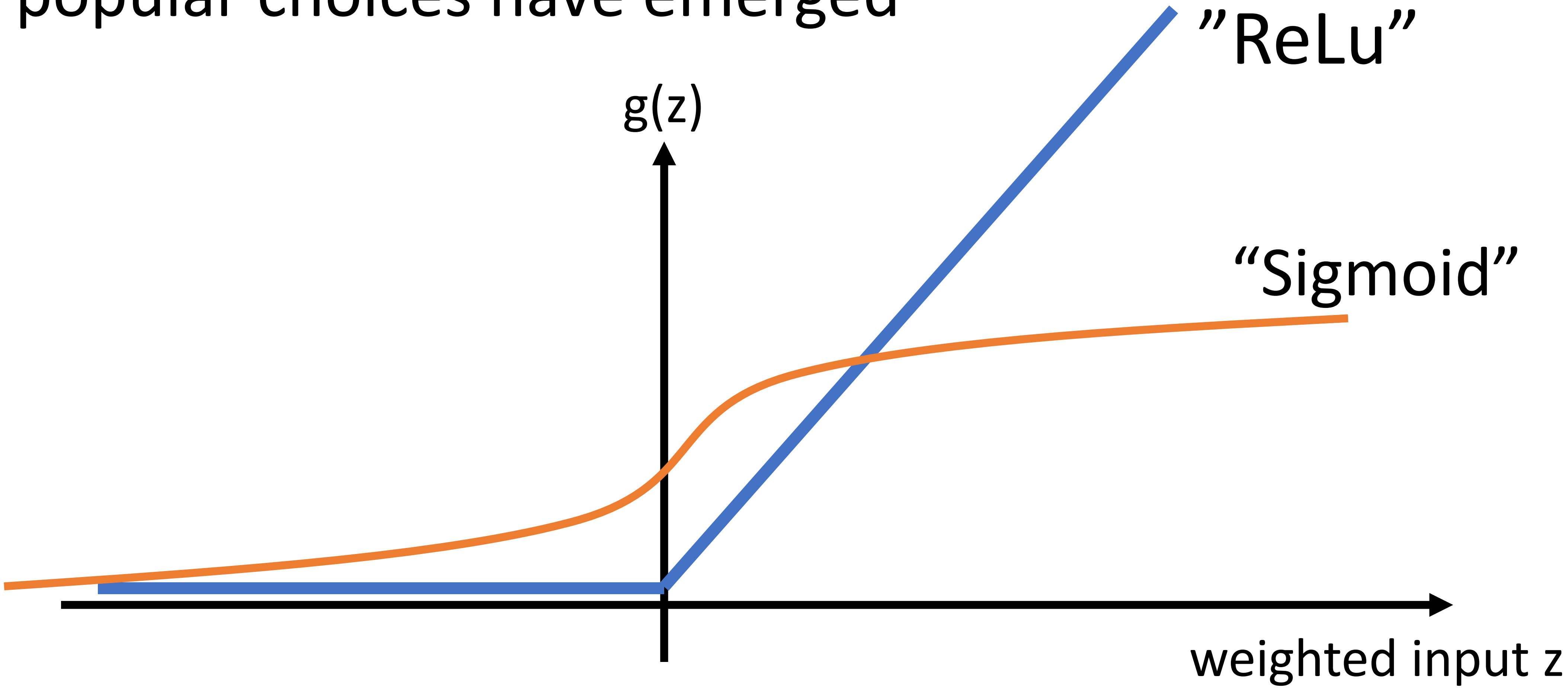
h(x) = g(z)

z = w1*x1+w2*x2+w3*x3

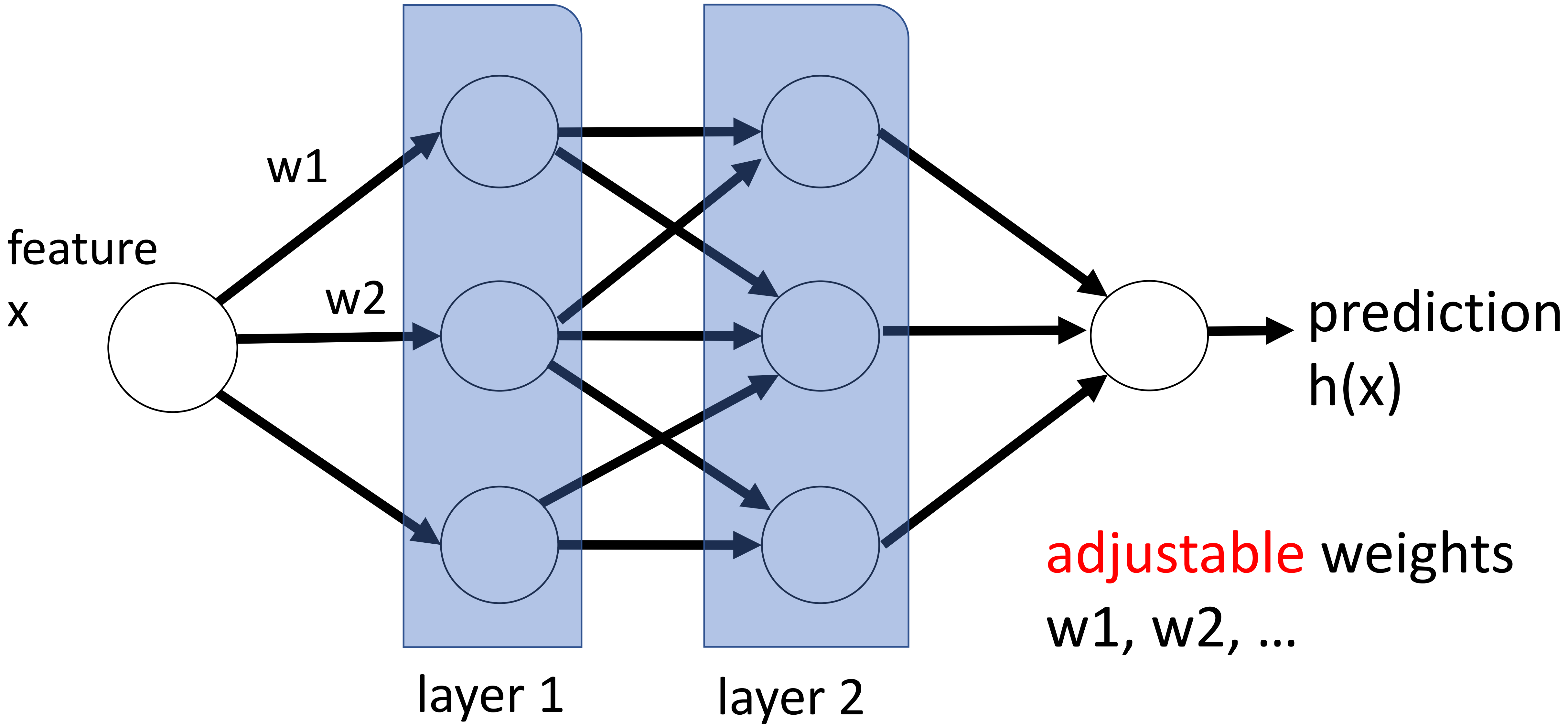
non-linear activation function g(z)

Activation Function

- few popular choices have emerged

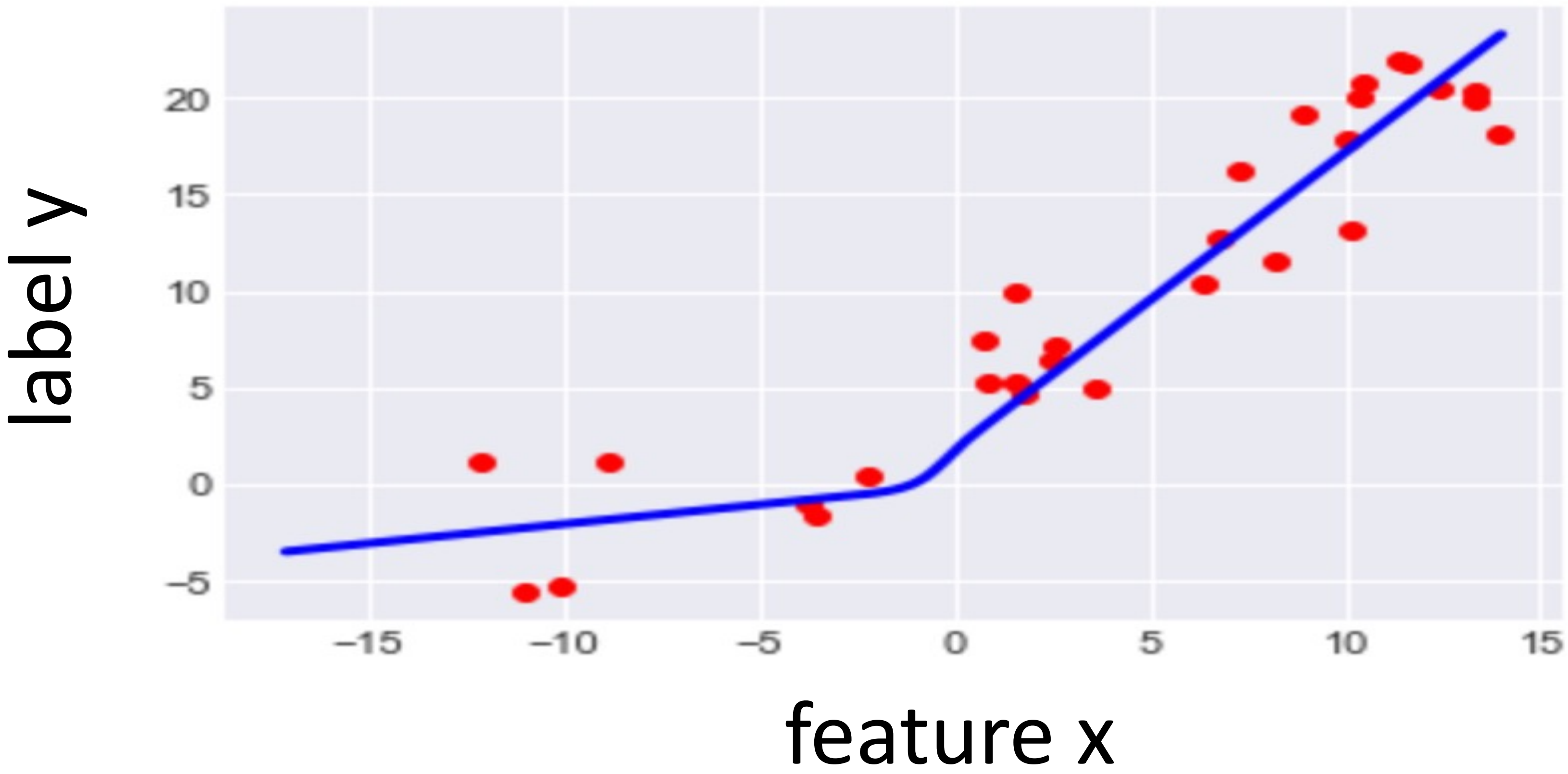


(Deep) Neural Network=(Very) Non-Linear Function



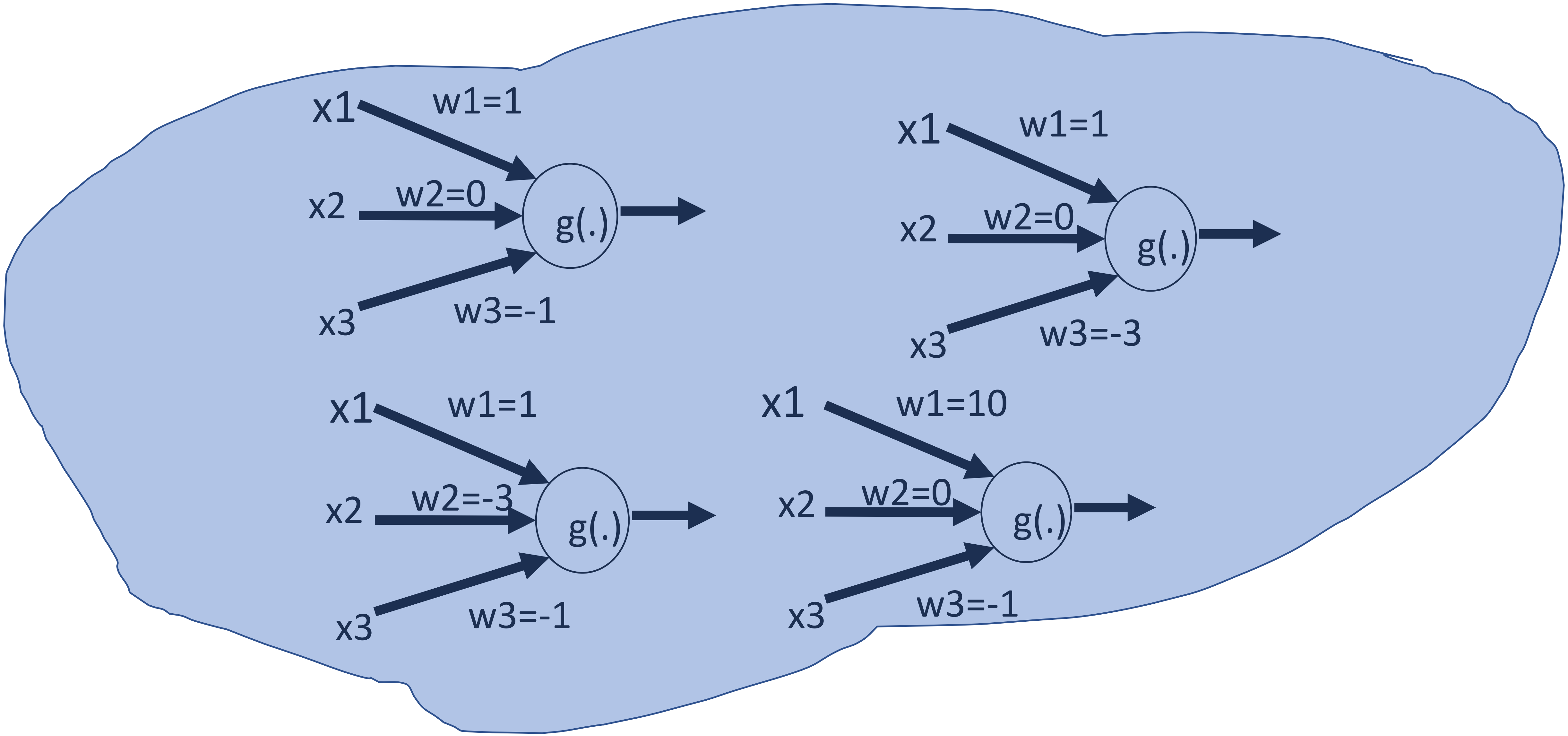
adjustable weights
w1, w2, …

What is Deep Learning ?



deep learning methods fit non-linear maps to large data sets

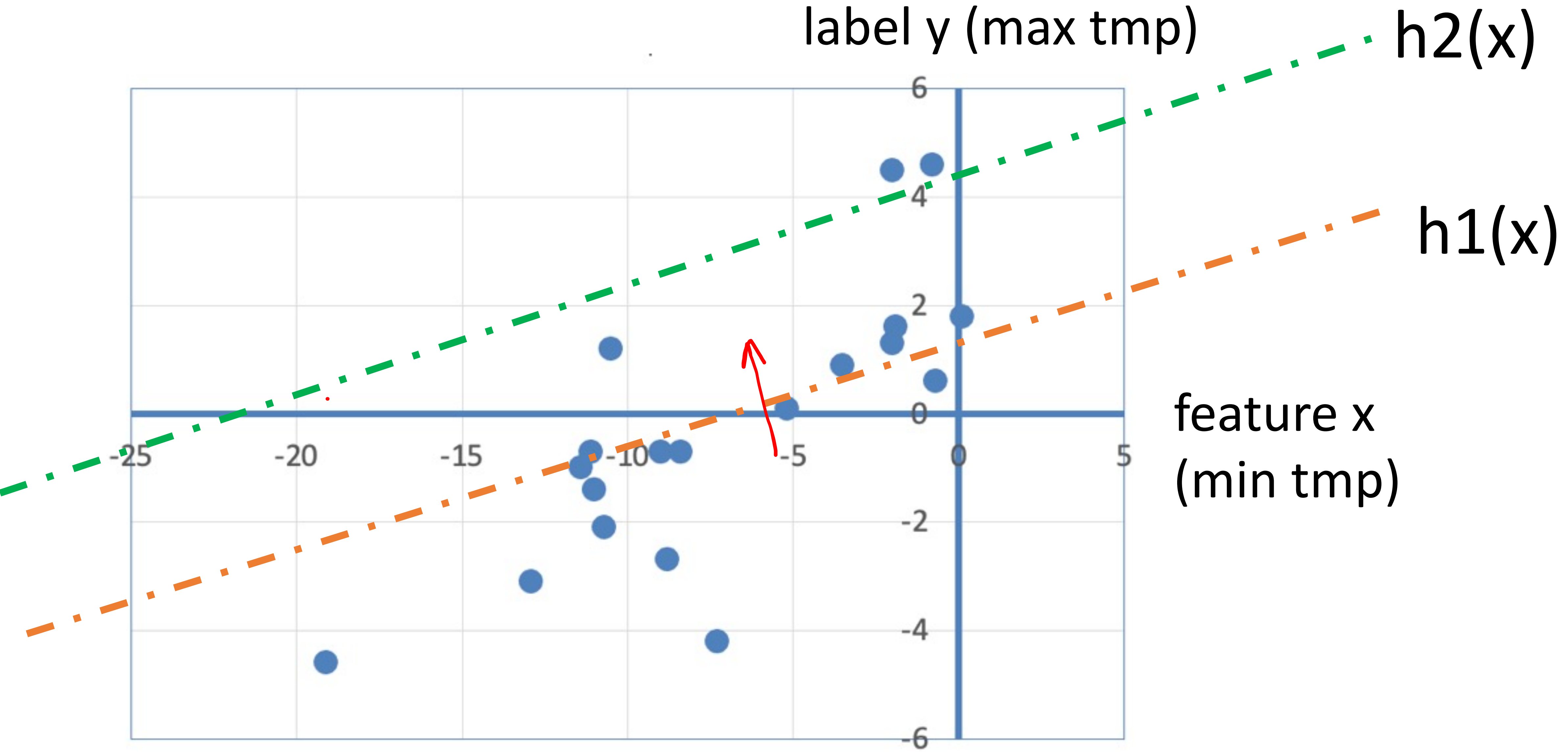
Hypothesis Space of ANN



Start 17:07

Loss Function

which hypothesis is better ?



Loss Function

maps a pair consisting of a data point (x,y) and some hypothesis h to some number

((x,y),h) -> "Loss" denoted L(h,(x,y))

loss function is design choice!

Some Popular Loss Functions

absolute $|y - h(x)|$ "Huber"

squared error loss (numeric labels):

$$L\big(h,(x,y)\big) = (y - h(x))^2$$

$-\infty \leftarrow$

logistic loss (for binary labels, e.g., -1 and 1):

$$L\big(h,(x,y)\big) = \log_e(1+\exp(-yh(x)))$$

note that loss depends on (weights of) predictor map!

Chose Your Favorite Loss Function!

loss function used for adjusting weights

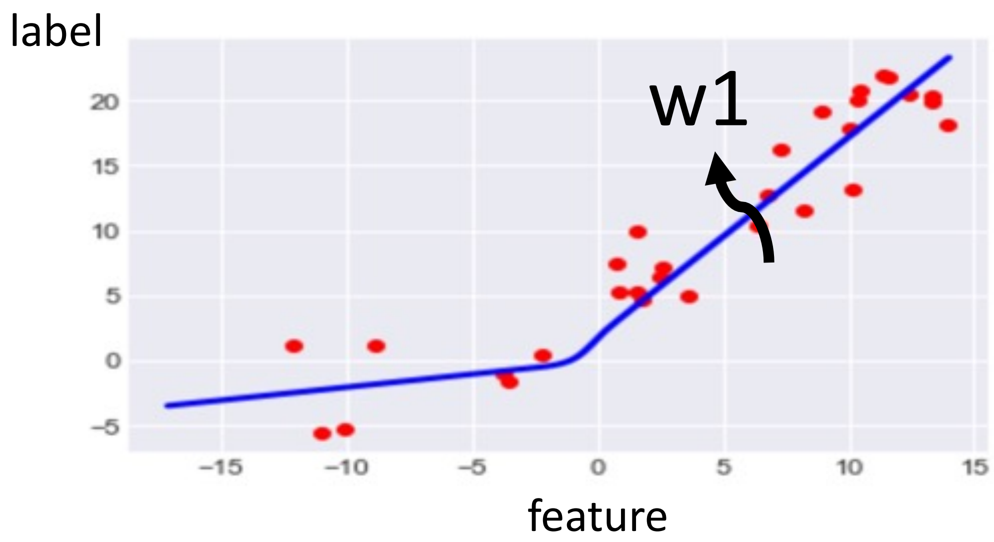
In [8]:
```
### Compile the model
model.compile(optimizer=keras.optimizers.RMSprop(),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

loss function used for final performance evaluation

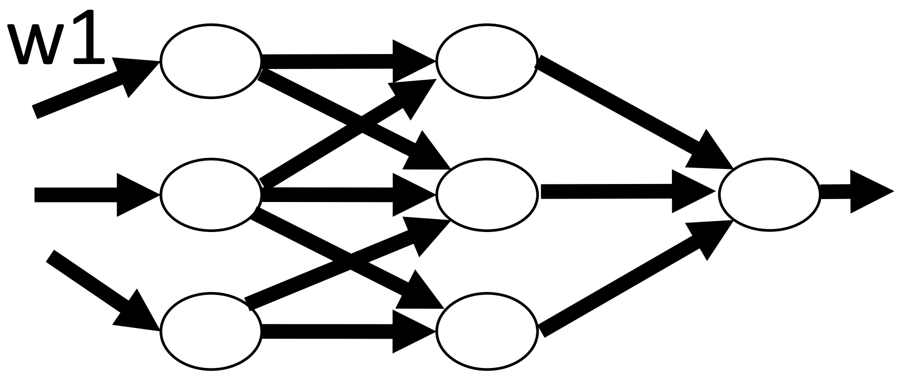
Putting Together the Pieces!

Three Views on Machine Learning

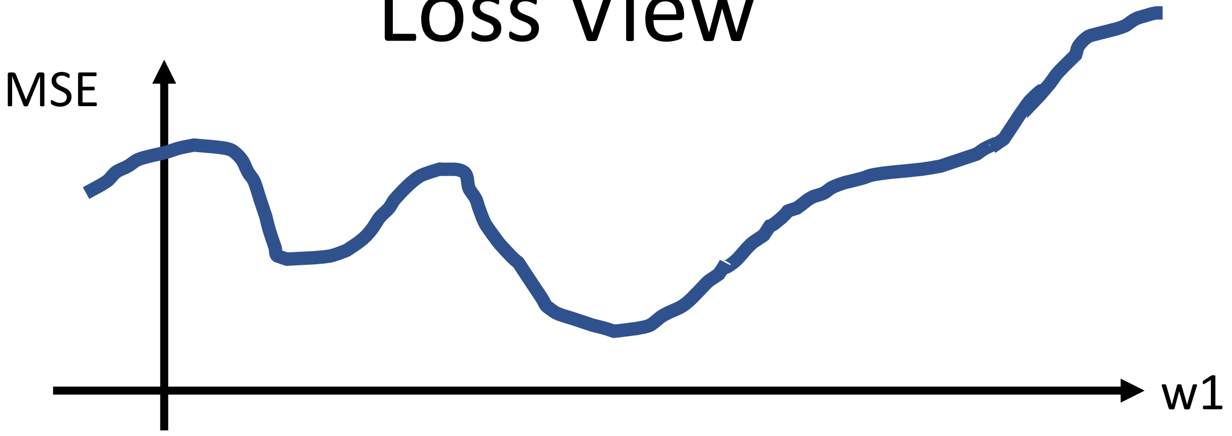
“Data View”



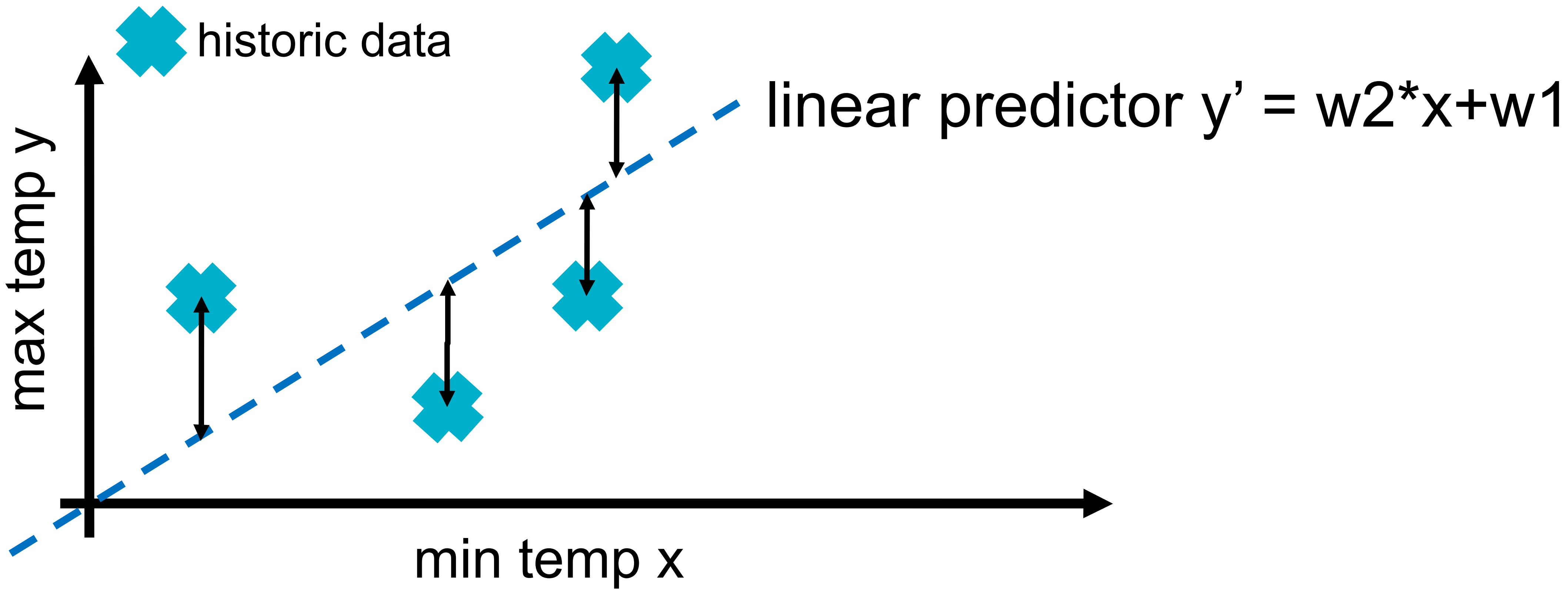
“Model View”



“Loss View”

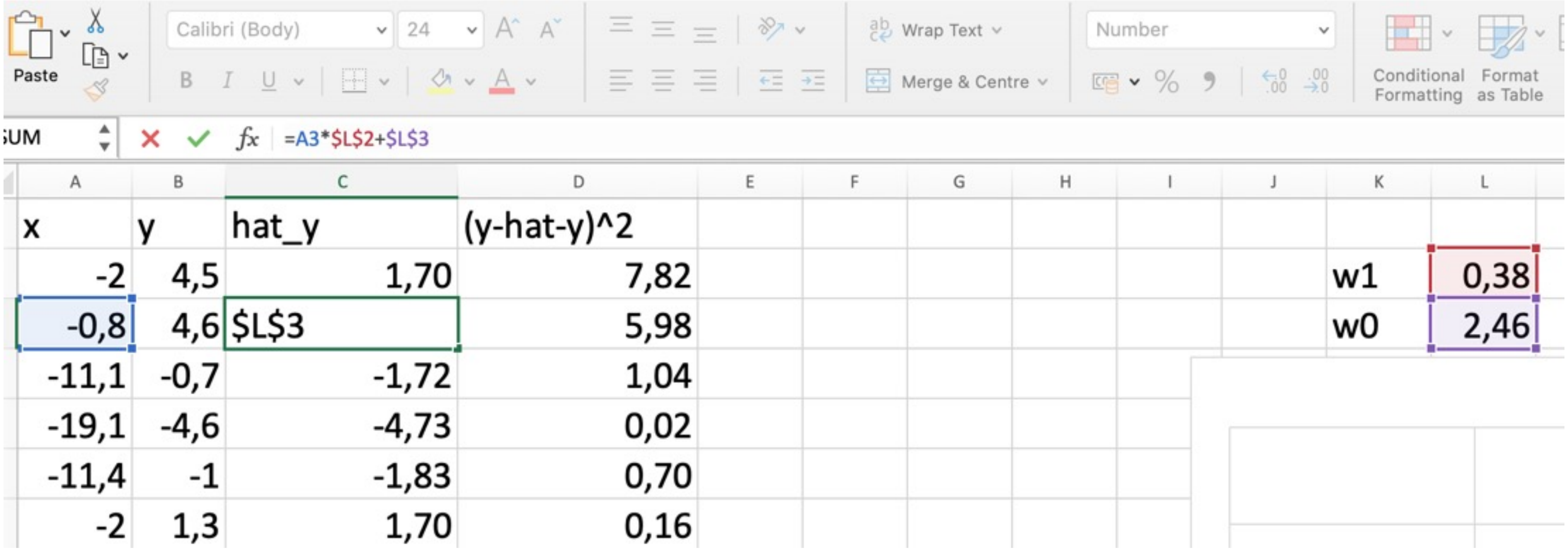


learn predictor y'=h(x) by tuning
weights w2,w1 to minimize average loss



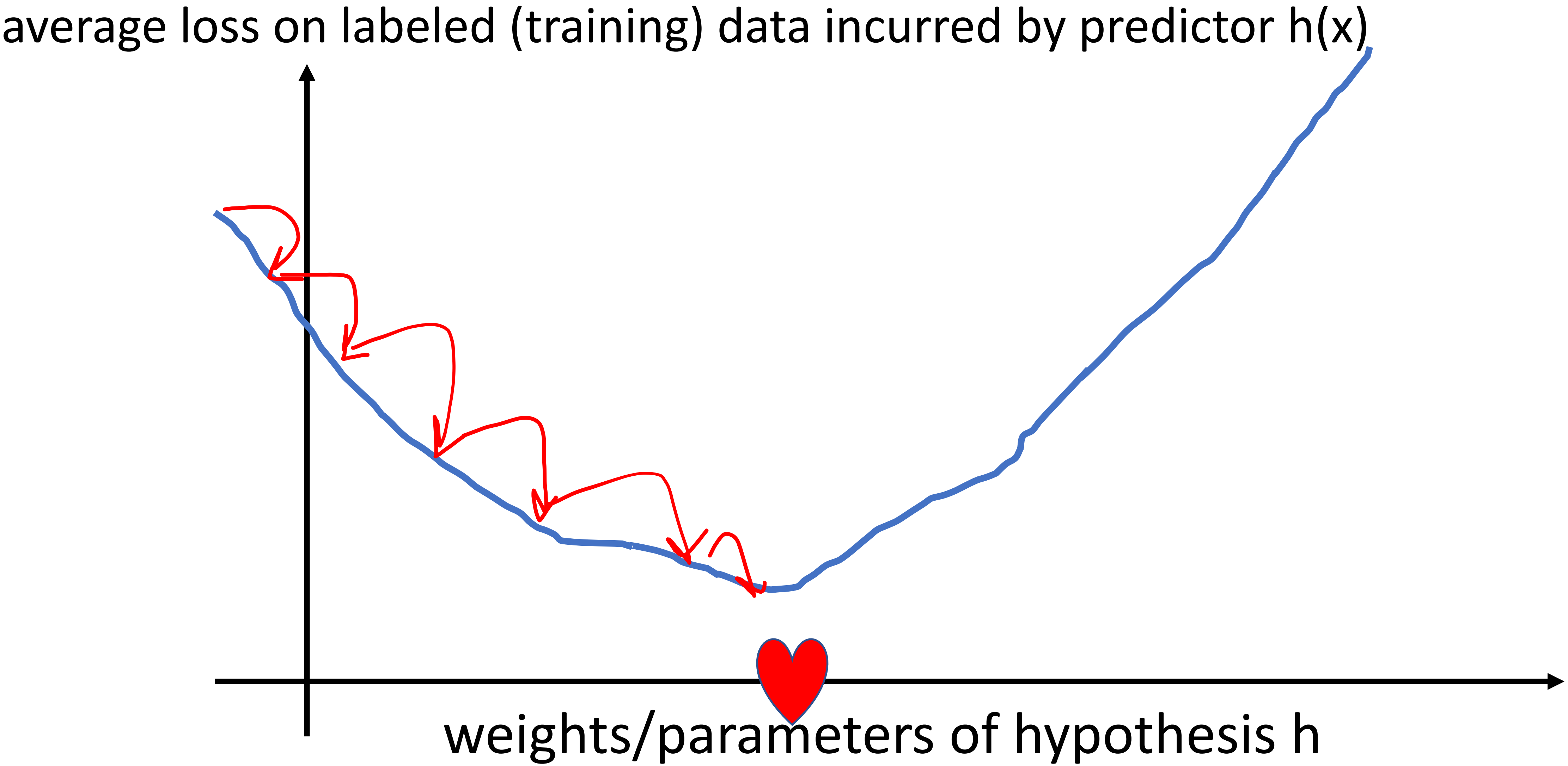
Minimize Average Loss

hat_y = h(x) = w1*x+w0



x	y	hat_y	(y-hat-y)^2	w	
-2	4,5	1,70	7,82	w1	0,38
-0,8	4,6	L3	5,98	w0	2,46
-11,1	-0,7	-1,72	1,04		
-19,1	-4,6	-4,73	0,02		
-11,4	-1	-1,83	0,70		
-2	1,3	1,70	0,16		

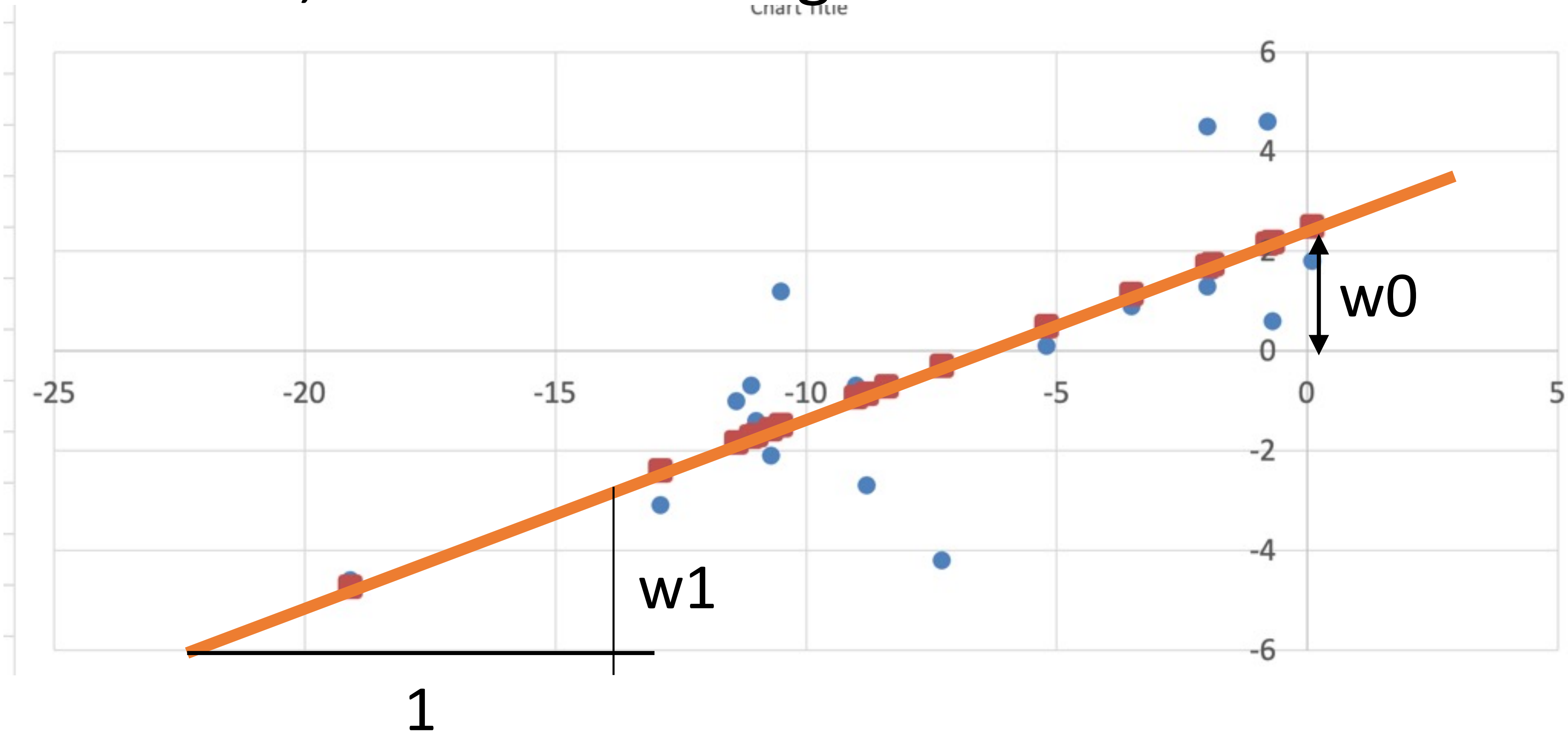
Machine Learning = Optimization



Optimizing using Spreadsheet Software

linear model: h(x) = w1*x+w0

choose w1,w0 to min average loss



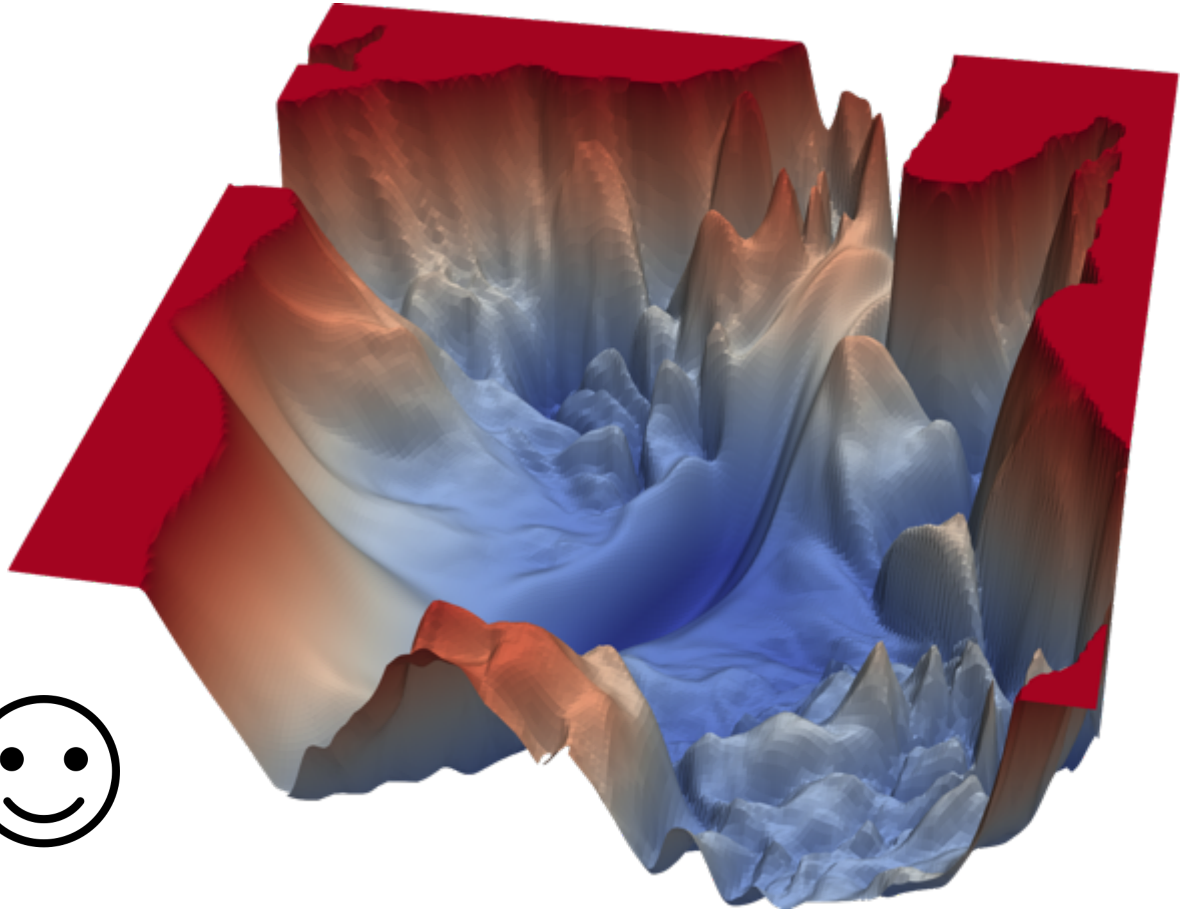
Loss Landscape for Deep Learning

non-convex 😟

non-smooth 😟

high-dimensional 😟

in theory we can evaluate
loss function everywhere ! 🙂



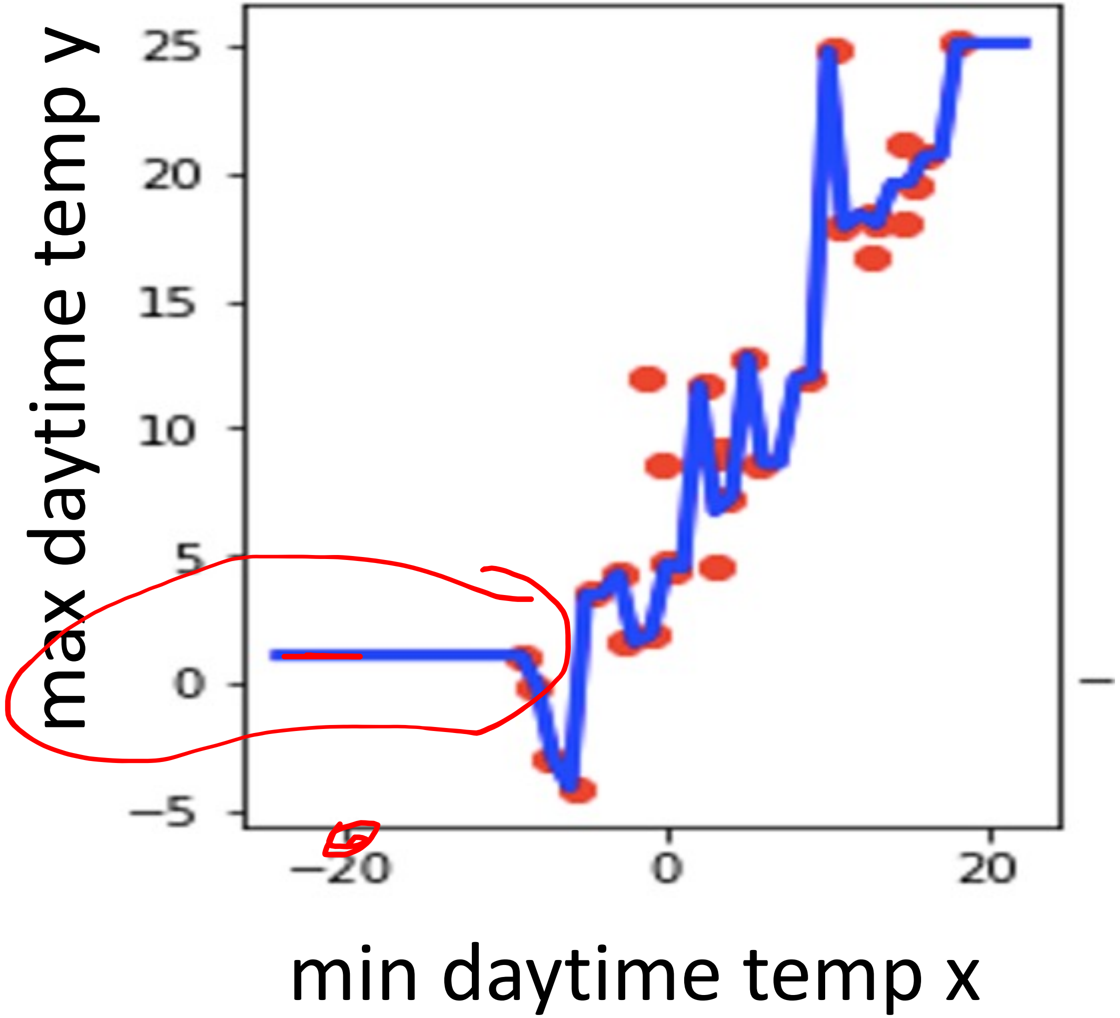
https://www.cs.umd.edu/~tomg/projects/landscapes/

So far so Good!

assume we have found the “best” hypothesis with minimum average loss on training set

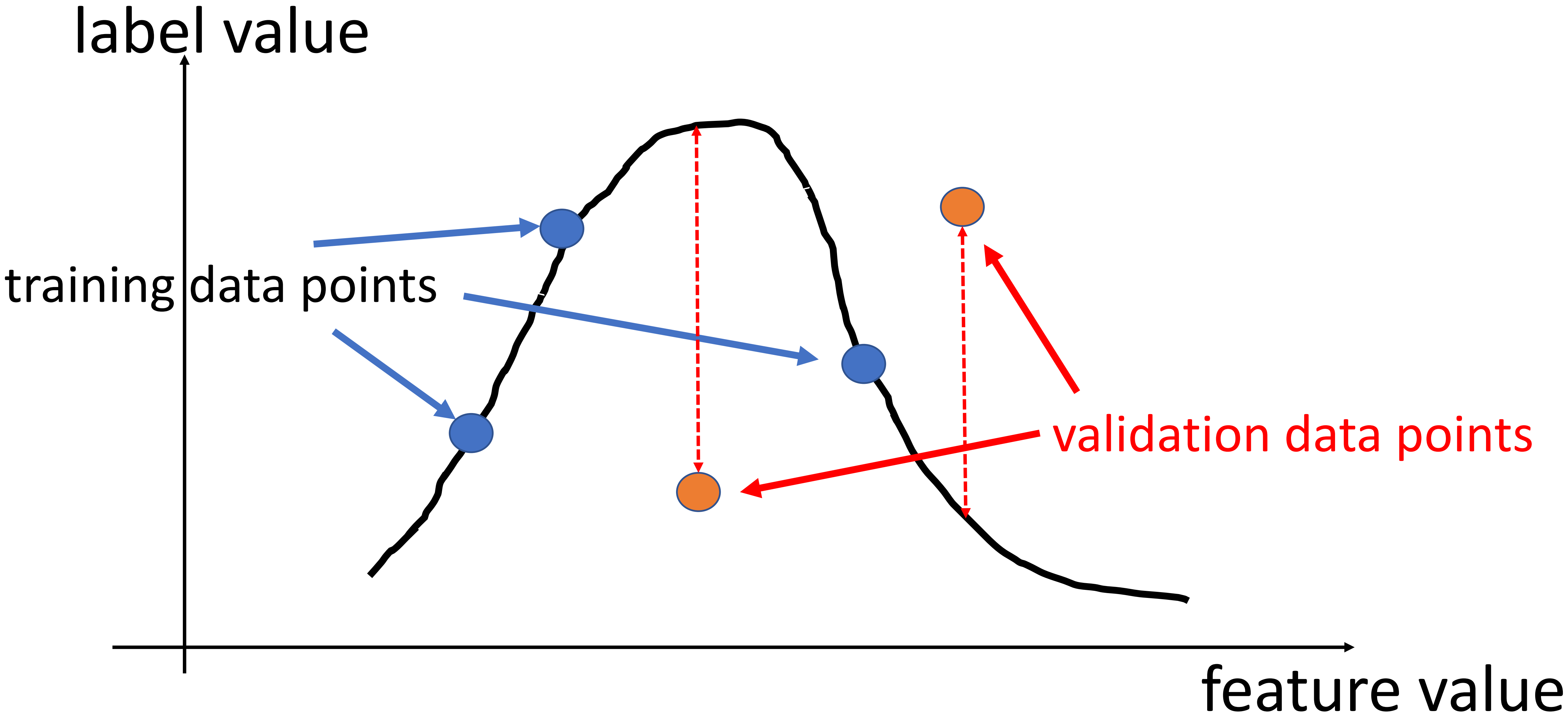
are we done?

Key Challenge in Machine Learning - Overfitting

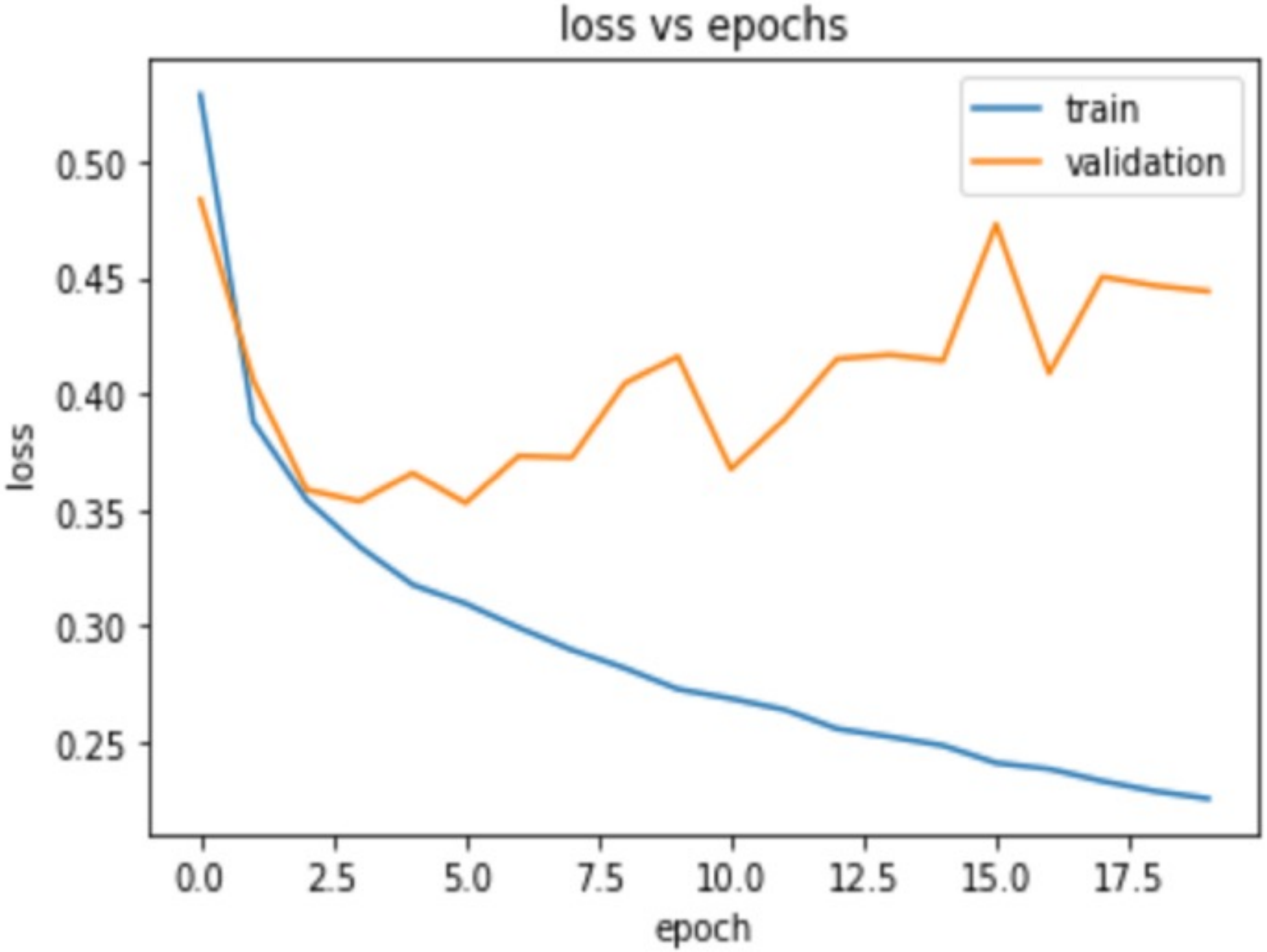


small training error but poor predictor map!

Detecting Overfitting by Validation



Look at the Validation Set !!!



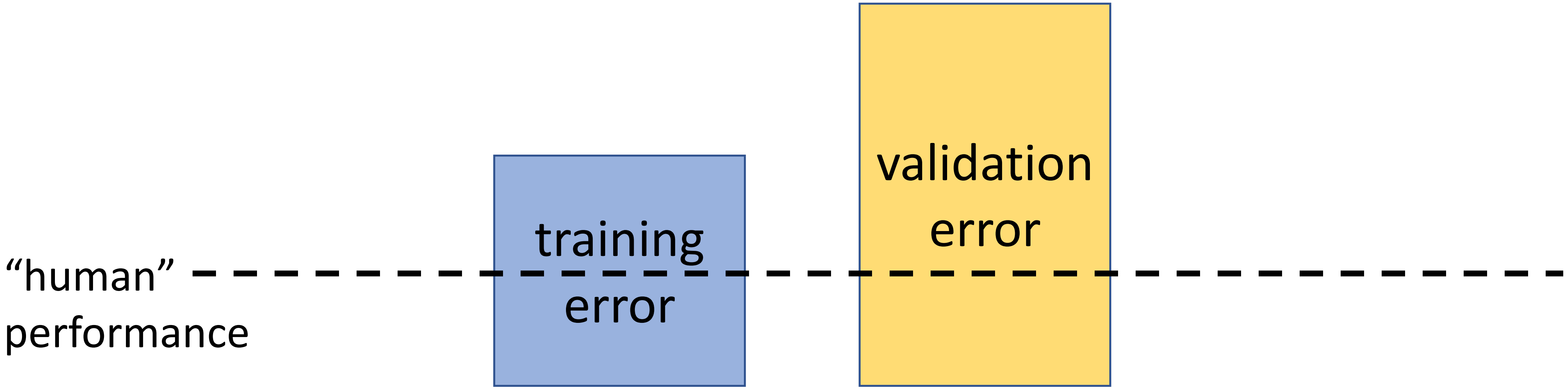
Training, Validation and Test Set

- training set: used to adjust weights
- validation set: used to adjust hyperparameters (number of layers..)
- test set: final performance evaluation

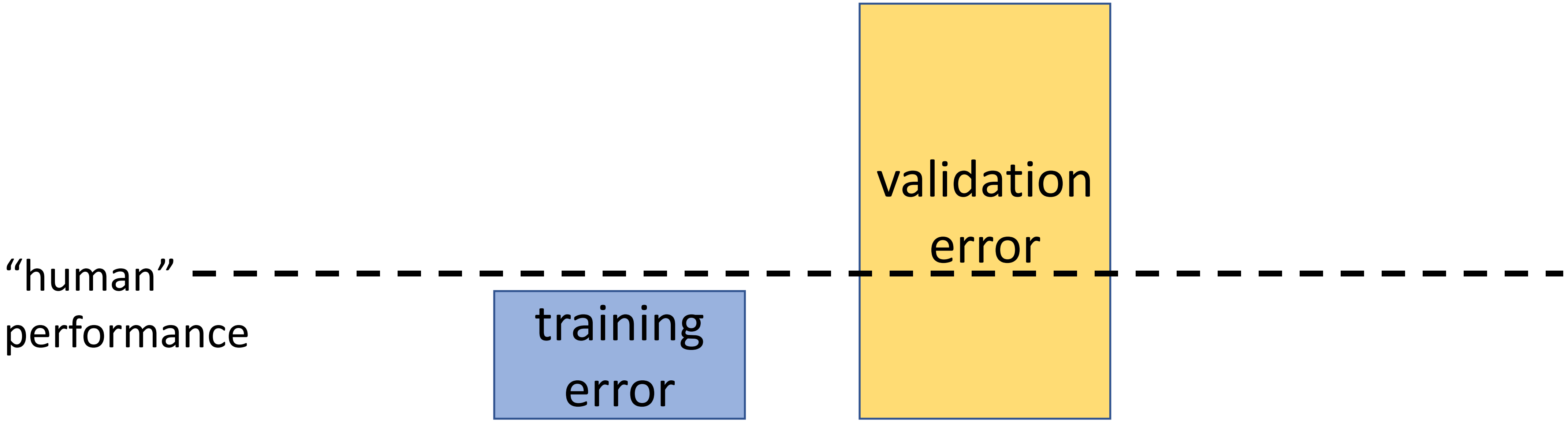
Test Set

- used for final performance evaluation
- results on test set MUST NOT BE used for model adjustment!

Diagnosing ML Methods



Case 1: Overfitting



possible remedies:

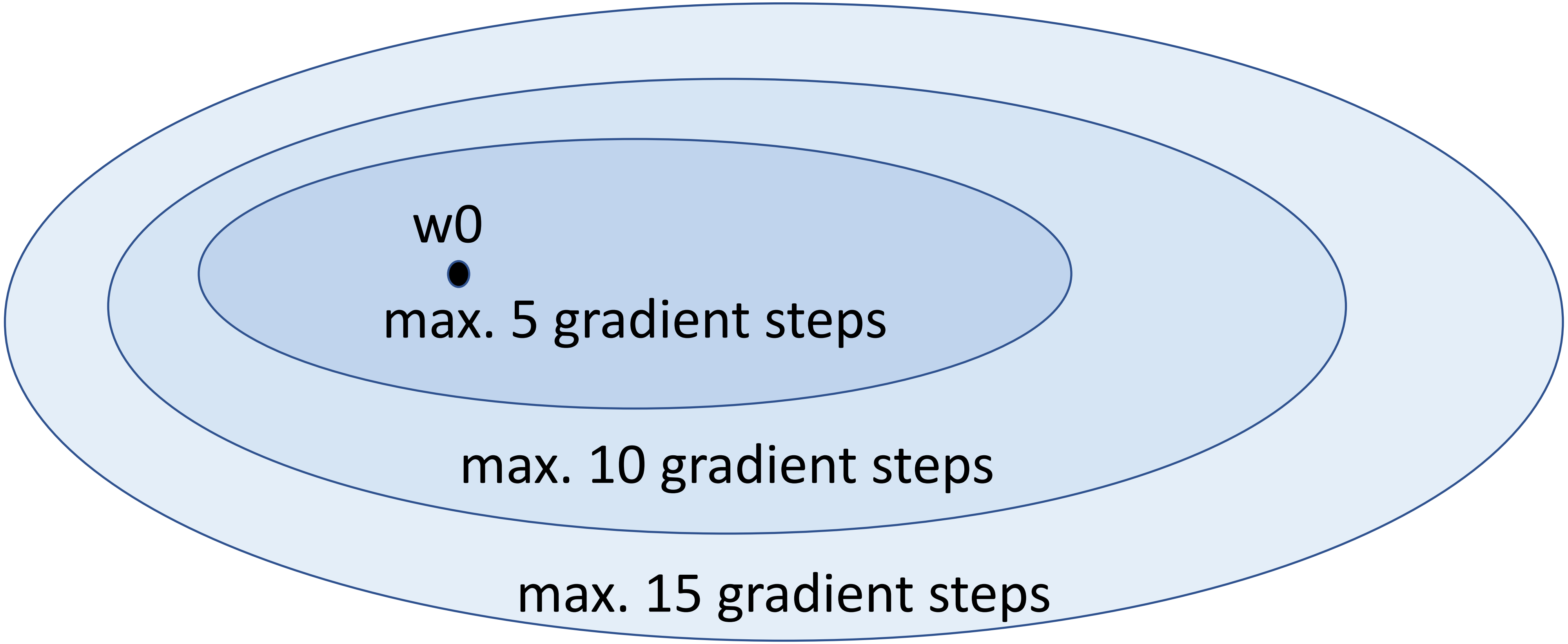
reduce hypothesis space or use more training data

Reducing Hypothesis Space

- use fewer neurons in hidden layers
- use fewer features (manually choose relevant features)
- use fewer layers
- use fewer iterations of gradient descent (search only a smaller subset of the nominal space)

"early stopping"

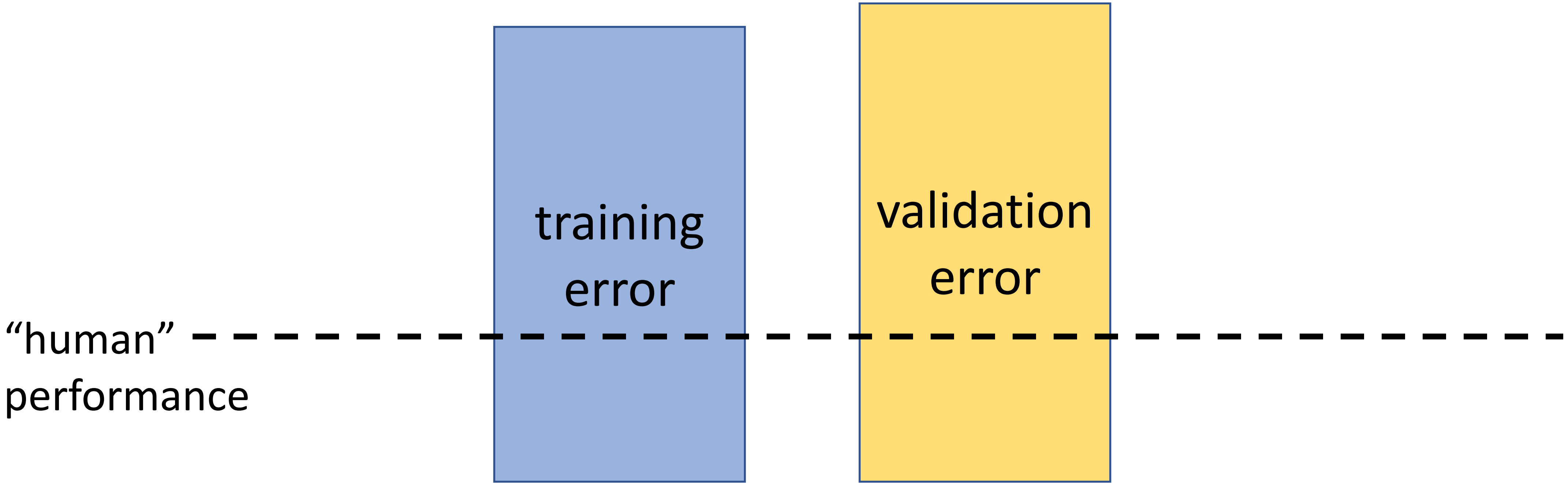
Early Stopping ≈ Hypothesis Space Reduction



Data Augmentation

- enlarge training data artificially
- rotated/flipped/mirrored/blurred/noisy dog image is still dog image

Case 2: Underfitting



possible remedy:
enlarge hypothesis space

Enlarging Hypothesis Space

- use more features
- use larger ANNs
- use deep decision trees
- construct more hypotheses using lookup table
- ….

Wrap Up

- ML combines data, model and loss
- learn/train hypothesis by min. average loss
- after training, validate hypothesis on new data
- diagnose method by comparing train/val error

Submit “Your ML Problem” by Friday 15.01. evening!