University of Salford MANCHESTER

Artificial Intelligent – Week 4

Dr Salem Ameen

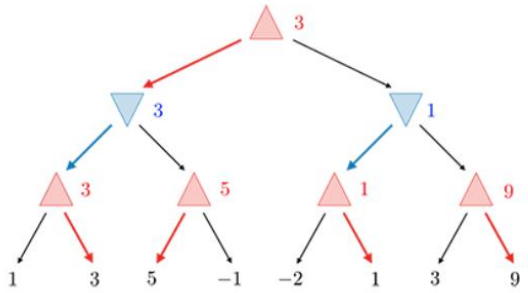
S.A.AMEEN1@SALFORD.AC.UK

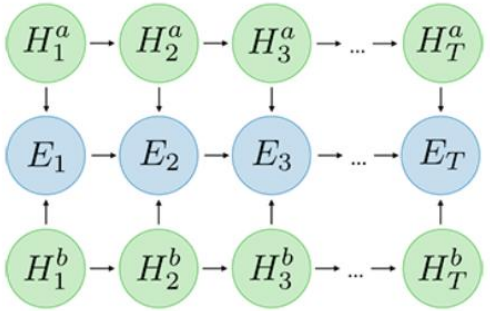
Lecture - Week 4: Optimization and Uncertainty

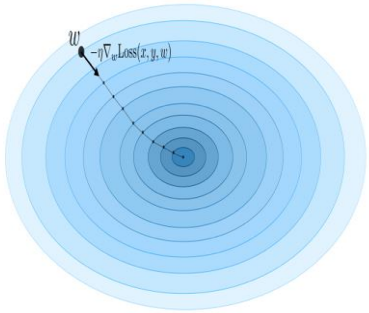
University of Salford MANCHESTER

- Optimization
- Uncertainty
  - Estimation

Lecture - Week 4: Agent Types







Search problems | Constraint satisfaction problems

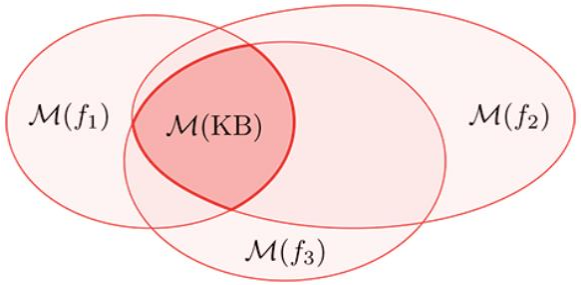
Markov decision processes | Markov networks

Adversarial games | Bayesian networks

Reflex **States** **Variables** **Logic**

"Low-level intelligence" "High-level intelligence"

Machine learning



Lecture - Week 4 - Machine Learning

University of Salford MANCHESTER

❑ What is Machine Learning (ML) :

- **The study of computer programs (algorithms) that can learn by example**
- **ML algorithms can generalize from existing examples of a task**
- *– e.g. after seeing a training set of labelled images, an image classifier can figure out how to apply labels accurately to new, previously unseen images*

Lecture - Week 4 - Machine Learning

University of Salford MANCHESTER

❑ What is Machine Learning (ML) :

Two definitions of Machine Learning are offered.

- Arthur Samuel described it as: "the field of study that gives computers the ability to learn without being explicitly programmed." This is an older, informal definition.
- Tom Mitchell provides a more modern definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Lecture - Week 4 - Machine Learning

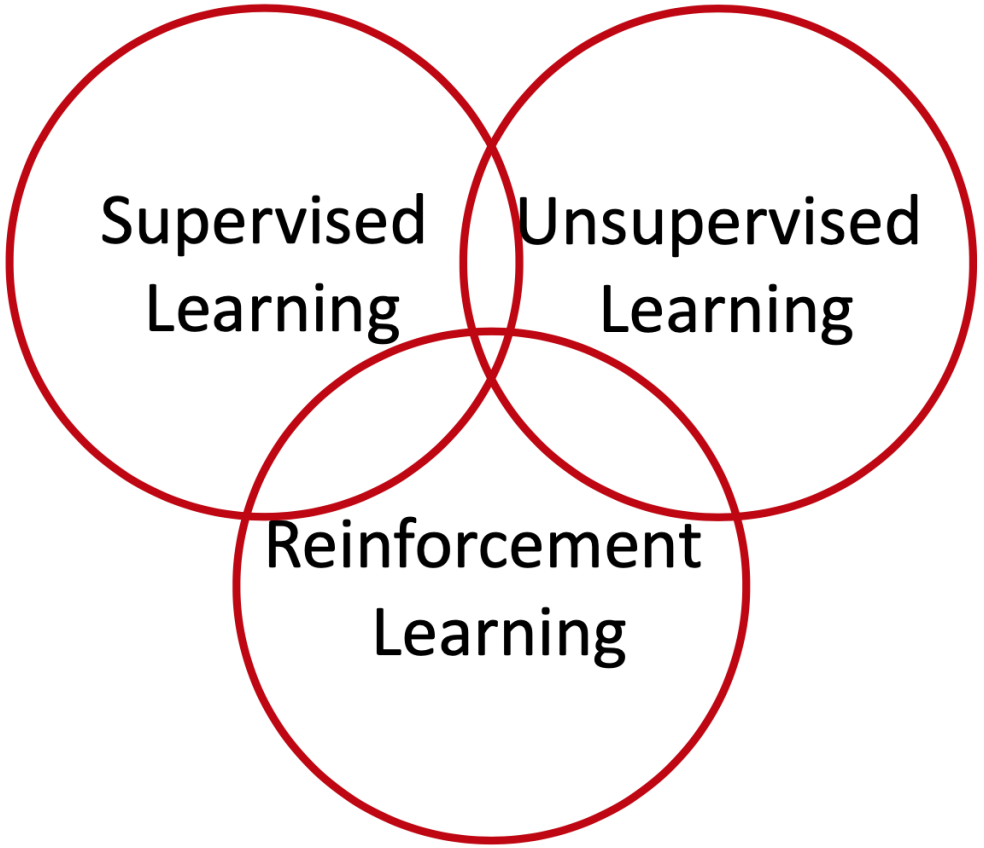
University of Salford MANCHESTER

- **What is Machine Learning (ML)** :

  Learning from the Data

  - Data
  - Pattern
  - Model

Lecture - Week 4 - Machine Learning



Lecture - Week 4 - Machine Learning

University of Salford MANCHESTER

❑ Key types of Machine Learning problems :

- **Supervised machine learning**: Learn to predict target values from labelled data.
- **Unsupervised machine learning: Find structure in *unlabelled data***
  - Find groups of similar instances in the data (clustering)
  - Finding unusual patterns (outlier detection)
- **Reinforcement Learning: RL** is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.

Lecture - Week 4 - Supervised machine learning

University of Salford MANCHESTER

Given a set of data points $\{x^{(1)}, ..., x^{(m)}\}$ associated to a set of outcomes $\{y^{(1)}, ..., y^{(m)}\}$, we want to build a classifier that learns how to predict y from x.

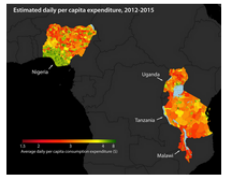
□ **Type of prediction** — The different types of predictive models are summed up in the table below:

	Regression	Classification
Outcome	Continuous	Class
Examples	Linear regression	Logistic regression, SVM, Naive Bayes

Lecture - Week 4 - Supervised ML Regression

University of Salford MANCHESTER

$$x \longrightarrow \boxed{f} \longrightarrow y \in \mathbb{R} \text{ response}$$

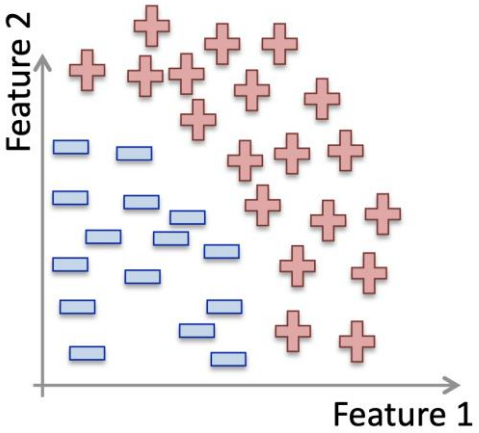


Poverty mapping: satellite image $\longrightarrow$ asset wealth index

Housing: information about house $\longrightarrow$ price

Arrival times: destination, weather, time $\longrightarrow$ time of arrival

Lecture - Week 4 - Supervised ML
Binary classification

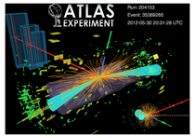


classifier

$$x \longrightarrow \boxed{f} \longrightarrow y \in \{+1, -1\}$$ label

Fraud detection: credit card transaction $\longrightarrow$ fraud or no fraud

Toxic comments: online comment $\longrightarrow$ toxic or not toxic



Higgs boson: measurements of event $\longrightarrow$ decay event or background

Lecture - Week 4 - Supervised ML Binary classification

University of Salford MANCHESTER

Hello Students,

To ESTIA group

Tomorrow, we have online session from 13:00 to 15:00 UK time (14:00 to 17:00 French time).

Regards,

Salem

Spam
Not Spam

MICROSOFT CORPERATION!
MICROSOFT E-MAIL SELECT AWARD HELD IN JOHANNESBURG SOUTH AFRICA

Congratulations!

Your active email address has just won you the sum of $1, 000, 000.00 USD (ONE MILLION UNITED STATES DOLLAR) in our annual international Microsoft Promotion Award held in Johannesburg South Africa, Award winners emerged through random selection of thousands of active email subscribers online. Three out of thousands of emails benefit from this promotion annually.

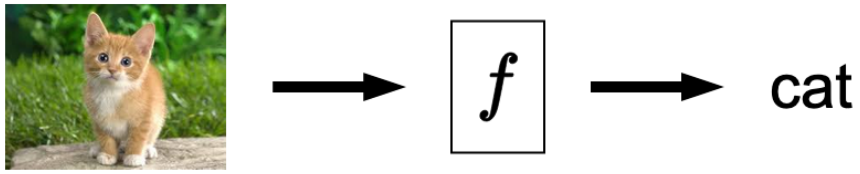
Winners are to be paid in accordance with his/her Settlement Centre. This Prize Award must be claimed in not less than 60 days from date of draw and notification, after which unclaimed prizes will be cancelled.

YOU ARE A WINNER NO: 2

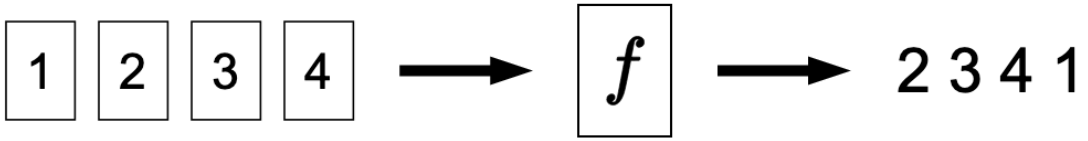
Lecture - Week 4 - Supervised ML
Multiclass classification

University of Salford MANCHESTER

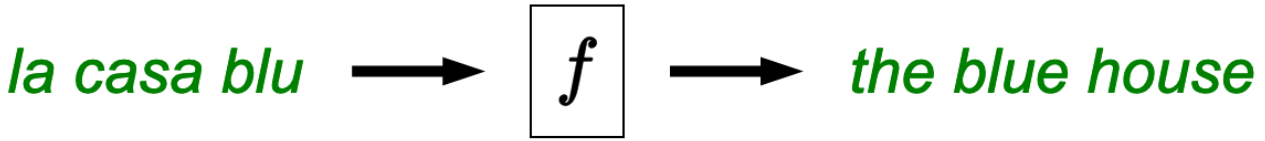
Multiclass classification: y is a category



Ranking: y is a permutation



Structured prediction: y is an object which is built from parts



Lecture - Week 4 - Supervised ML Multiclass classification

University of Salford MANCHESTER

sepal_length	sepal_width	petal_length	petal_width
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.6	3.4	1.4	0.3
5.0	3.4	1.5	0.2
4.4	2.9	1.4	0.2
4.9	3.1	1.5	0.1
5.4	3.7	1.5	0.2
4.8	3.4	1.6	0.2
4.8	3.0	1.4	0.1
4.3	3.0	1.1	0.1
5.8	4.0	1.2	0.2

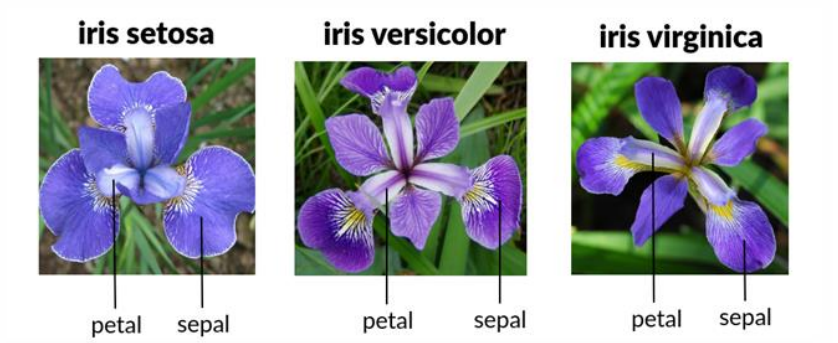
species
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa

Classifier
Model.fit(X,y)

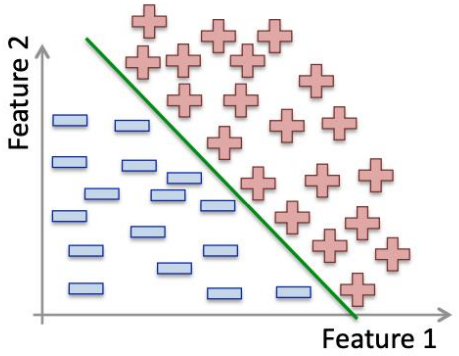

Iris setosa

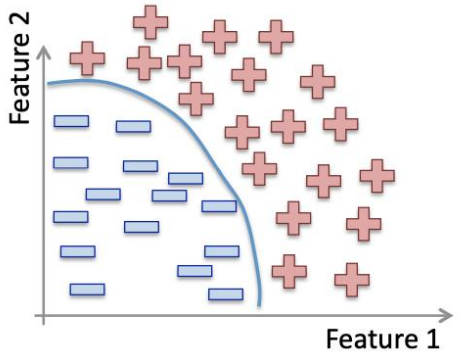

Iris versicolor

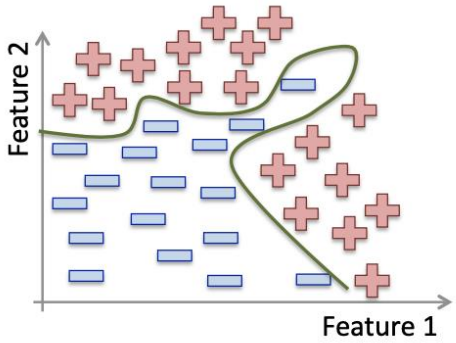

Iris virginica

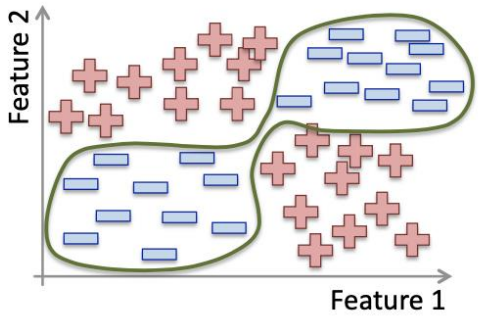


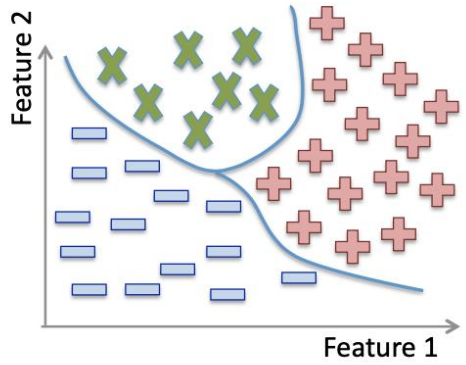
Lecture - Week 4 - Supervised ML Multiclass classification











Lecture - Week 4 - Supervised ML
Structured prediction

$x \longrightarrow \boxed{f} \longrightarrow y$ is a complex object

Machine translation: English sentence → Japanese sentence

Dialogue: conversational history → next utterance

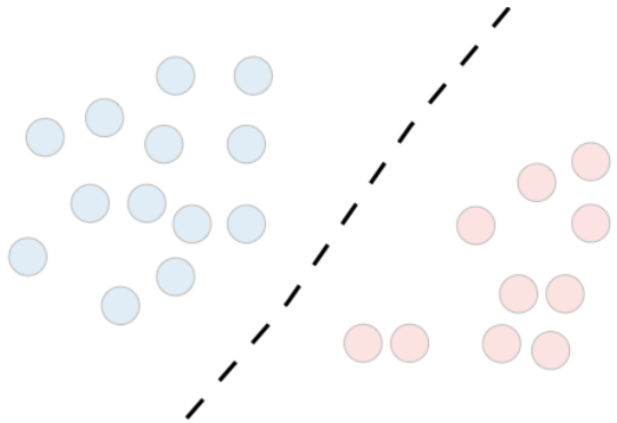
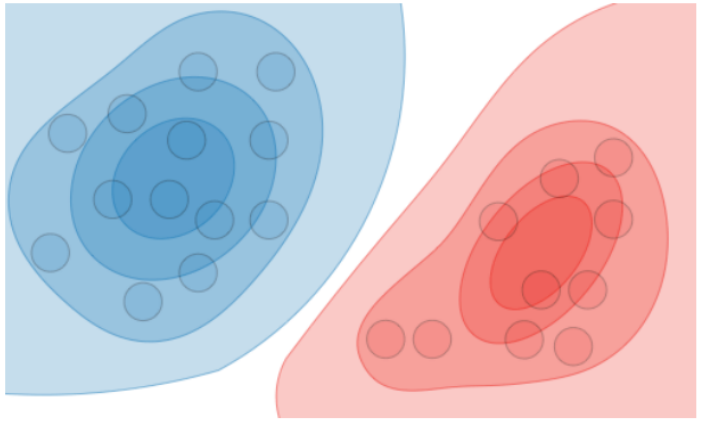
Image captioning: image → sentence describing image

Image segmentation: image → segmentation

Lecture - Week 4 - Supervised machine learning

University of Salford MANCHESTER

□ **Type of model** — The different models are summed up in the table below:

	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

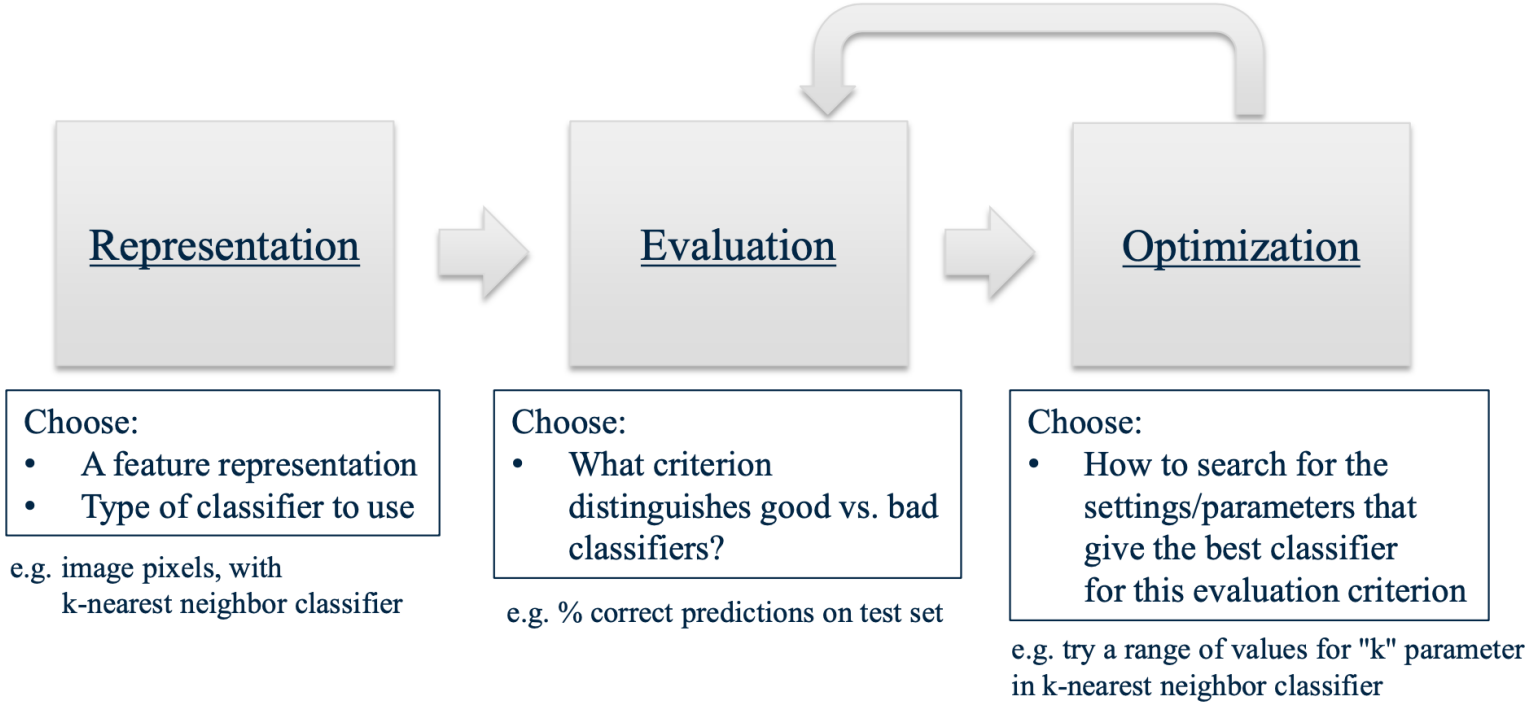
Lecture - Week 4 - Outline

University of Salford MANCHESTER

- Task
  - Supervised Learning
  - Linear regression
  - Polynomial regression
  - Linear classification
- Algorithm
  - Gradient descent

Lecture - Week 4 - Supervised ML

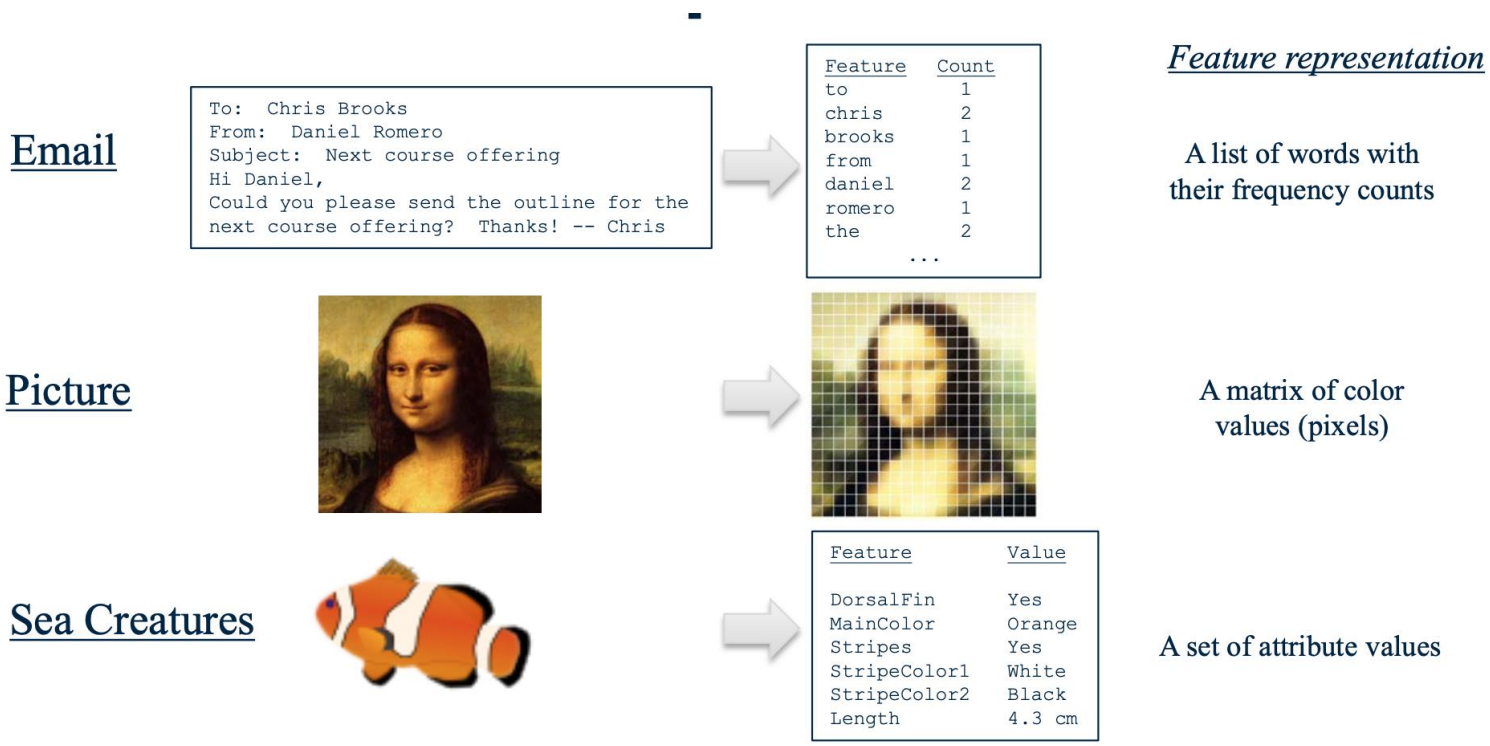
❑Machine Learning Workflow :



Lecture - Week 4 - Supervised ML

❑Machine Learning Workflow / Feature Representation :

-



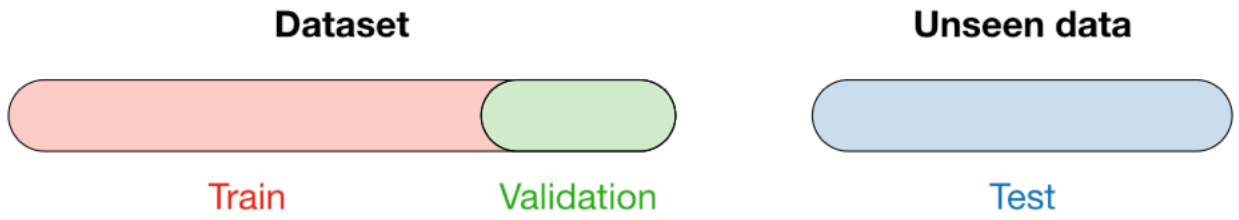
Lecture - Week 4 - Supervised ML

University of Salford MANCHESTER

- **Dataset**: (More the following lectures)

Training set	Validation set	Testing set
• Model is trained • Usually 80% of the dataset	• Model is assessed • Usually 20% of the dataset • Also called hold-out or development set	• Model gives predictions • Unseen data

Once the model has been chosen, it is trained on the entire dataset and tested on the unseen test set. These are represented in the figure below:



Lecture - Week 4 - Supervised ML

University of Salford MANCHESTER

❑Dataset:

https://archive.ics.uci.edu/ml/datasets/Container+Crane+Controller+Data+Set

```
import pandas as pd
df = pd.read_csv("dataset.csv")
df.head()
```

	Speed	Angle	Power
0	1	-5	0.3
1	2	5	0.3
2	3	-2	0.5
3	1	2	0.5
4	2	0	0.7

Lecture - Week 4 - Supervised ML

University of Salford MANCHESTER

- **Dataset**:

```
df.head()
```

	Speed	Angle	Power
0	1	-5	0.3
1	2	5	0.3
2	3	-2	0.5
3	1	2	0.5
4	2	0	0.7

```
X = df[["Speed","Angle"]]
Y = df["Power"]
```

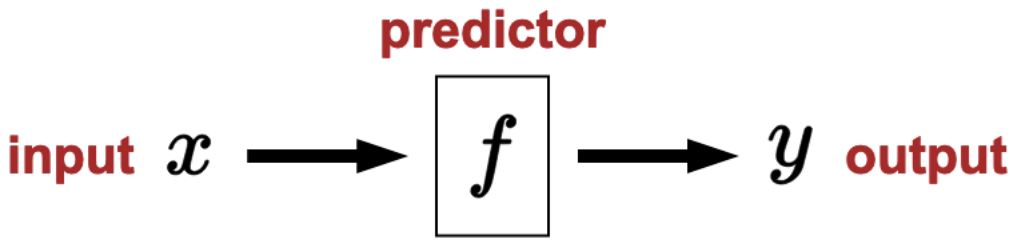
Lecture - Week 4 - Supervised ML

University of Salford MANCHESTER

- **Dataset**:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, Y, test_size=0.4,
random_state=0)
```

Lecture - Week 4 - Machine Learning
AI Paradigm



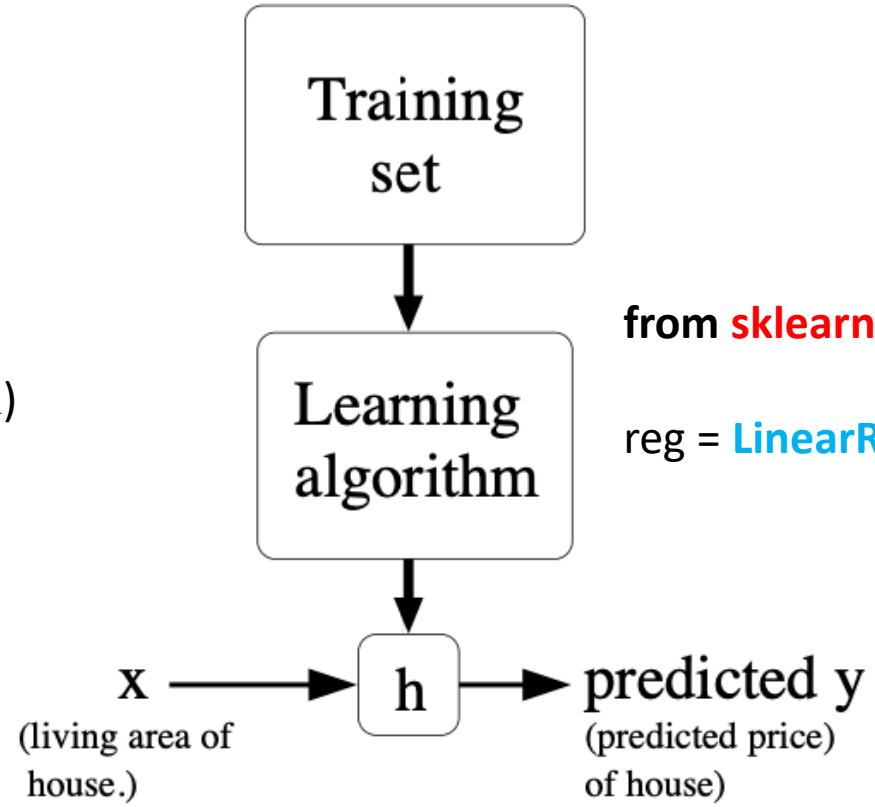
Modeling

Inference

Learning

Lecture - Week 4 - Machine Learning
Linear regression

```
X_train, X_test, y_train, y_test =
train_test_split( X, Y, test_size=0.4,
random_state=0)
```

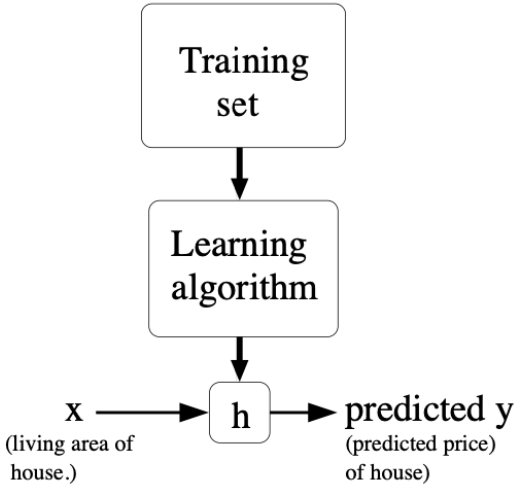


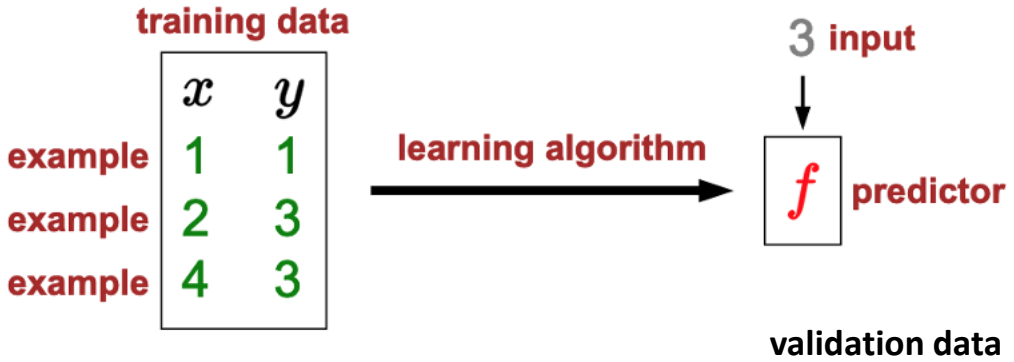

```
from sklearn.linear_model import LinearRegression

reg = LinearRegression().fit(X_train, y_train)
```

```
reg().score(X_train, y_train)

reg().score(X_test, y_test)
```

```
Y = reg().predict(X_new)
```

Lecture - Week 4 - Machine Learning
Linear regression

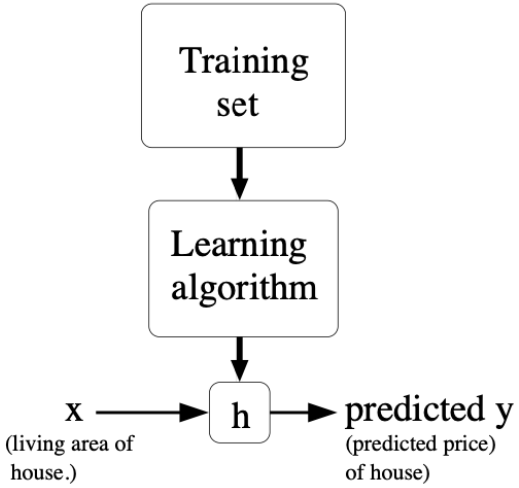




validation data

3 4

Lecture - Week 4 - Machine Learning
Linear regression



Line equation

$y = 2x - 1$

Slope

2

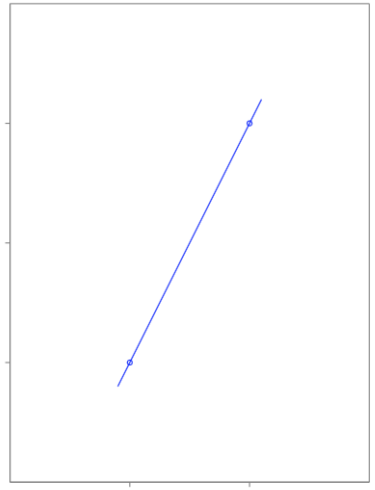
Intercept

-1

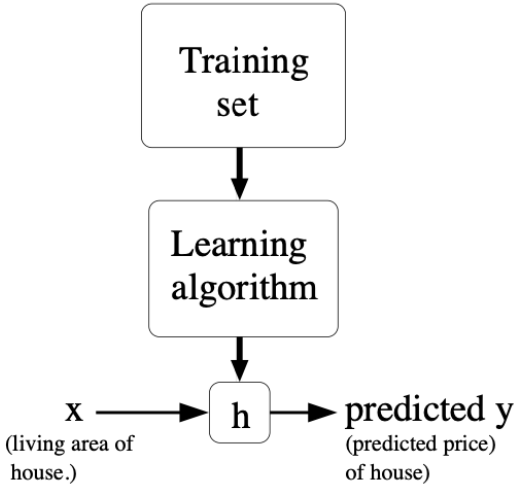
training data

	x	y
example	1	1
example	2	3
example	4	3

learning algorithm → f predictor



Lecture - Week 4 - Machine Learning
Linear regression



Line equation

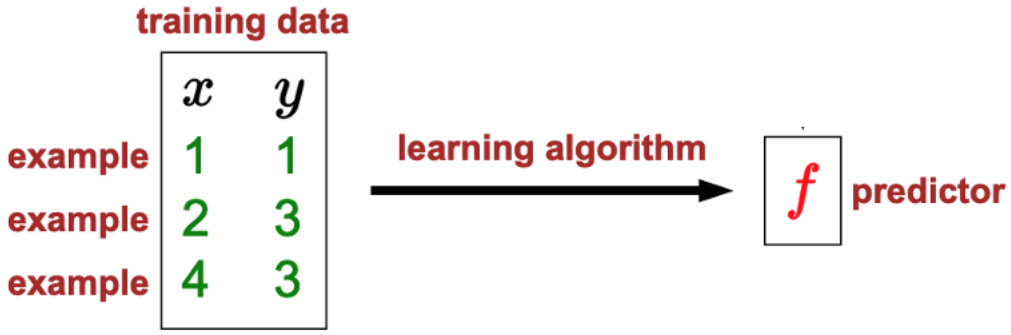
$$y = 0.6666666666666666x + 0.3333333333333335$$

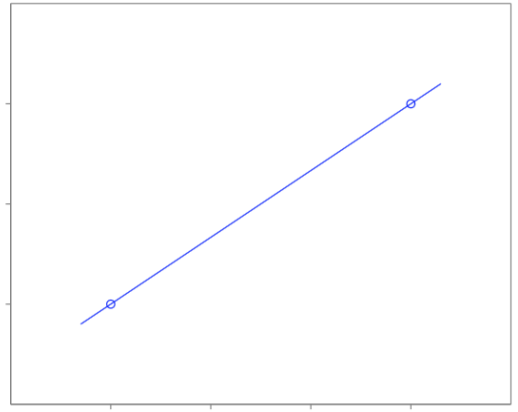
Slope

0.7

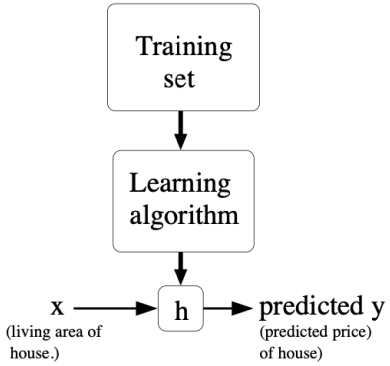
Intercept

0.3





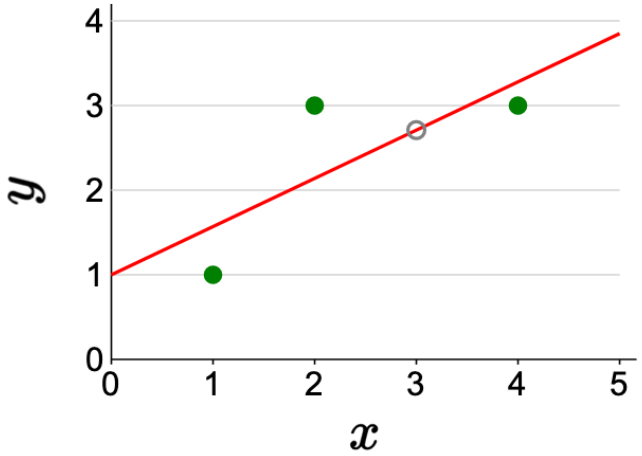
Lecture - Week 4 - Machine Learning
Linear regression



training data

	x	y
example	1	1
example	2	3
example	4	3

learning algorithm → f predictor



Design decisions:

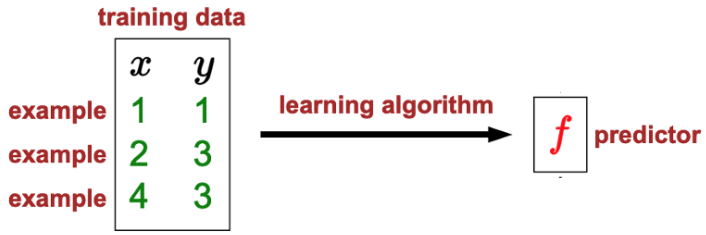
Which predictors are possible? **hypothesis class**

How good is a predictor? **loss function**

How do we compute the best predictor? **optimization algorithm**

Which predictors

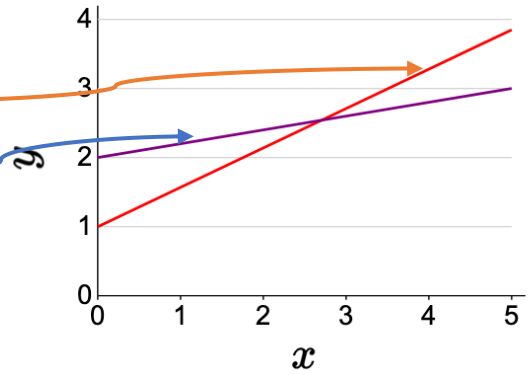
Lecture - Week 4 – ML - Linear regression
Hypothesis class: which predictors



$$f(x) = 1 + 0.57x$$

$$f(x) = 2 + 0.2x$$

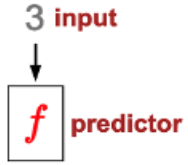
$$f(x) = w_1 + w_2 x$$



Vector notation:

weight vector $\mathbf{w} = [w_1, w_2]$ feature extractor $\phi(x) = [1, x]$ feature vector

$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x) \text{ score}$$



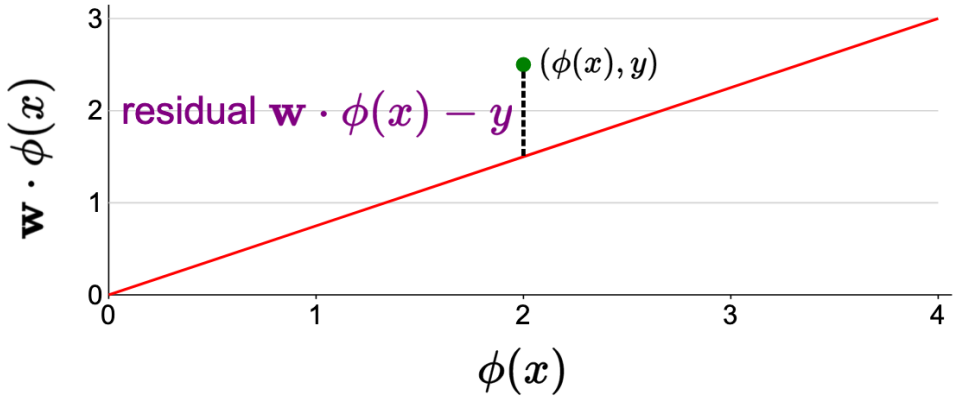
validation data

3 4

$$f_{\mathbf{w}}(3) = [1, 0.57] \cdot [1, 3] = 2.71$$

Lecture - Week 4 – ML - Linear regression
Hypothesis class: which predictors

$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$$



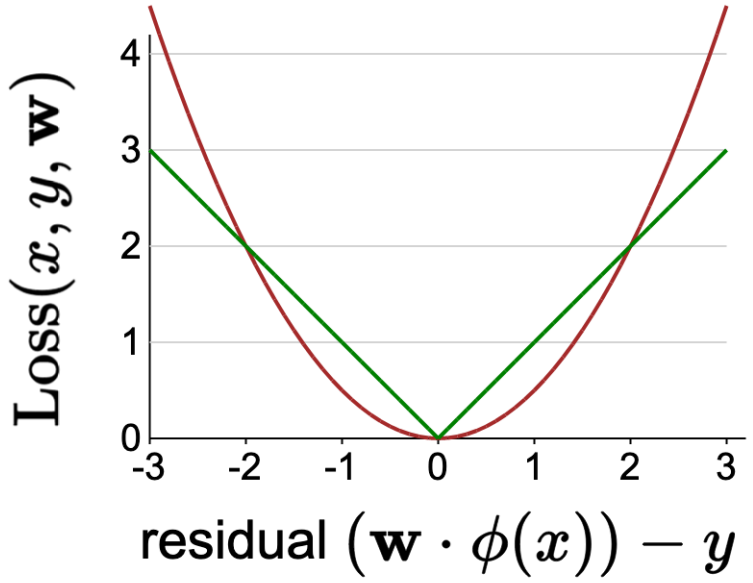
$$\text{Loss}_{\text{squared}}(x, y, \mathbf{w}) = (\underbrace{f_{\mathbf{w}}(x) - y}_{\text{residual}})^2$$

Definition: residual

The **residual** is $(\mathbf{w} \cdot \phi(x)) - y$, the amount by which prediction $f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$ overshoots the target y.

how good is a predictor?

Lecture - Week 4 – ML - Linear regression
Loss function: how good is a predictor?



$$\text{Loss}_{\text{squared}}(x, y, \mathbf{w}) = (\mathbf{w} \cdot \phi(x) - y)^2$$

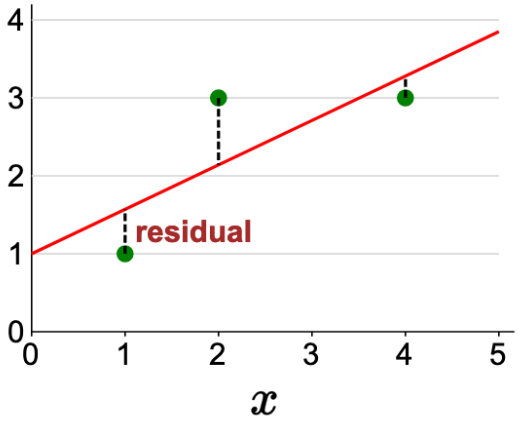
$$\text{Loss}_{\text{absdev}}(x, y, \mathbf{w}) = |\mathbf{w} \cdot \phi(x) - y|$$

Lecture - Week 4 – ML - Linear regression
Loss function: how good is a predictor?

$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$$
$$\mathbf{w} = [1, 0.57]$$
$$\phi(x) = [1, x]$$

training data $\mathcal{D}_{\text{train}}$

x	y
1	1
2	3
4	3



$$\text{Loss}(x, y, \mathbf{w}) = (f_{\mathbf{w}}(x) - y)^2$$ squared loss

$$\text{Loss}(1, 1, [1, 0.57]) = ([1, 0.57] \cdot [1, 1] - 1)^2 = 0.32$$
$$\text{Loss}(2, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 2] - 3)^2 = 0.74$$
$$\text{Loss}(4, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 4] - 3)^2 = 0.08$$

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \text{Loss}(x, y, \mathbf{w})$$

$$\text{TrainLoss}([1, 0.57]) = 0.38$$

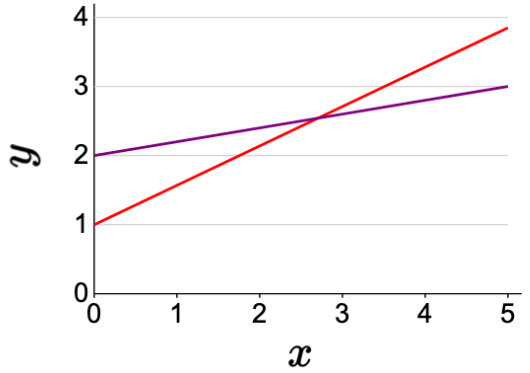
Lecture - Week 4 – ML - Linear regression
Loss function: how good is a predictor?

University of Salford MANCHESTER

$$f(x) = 1 + 0.57x$$

$$f(x) = 2 + 0.2x$$

$$f(x) = w_1 + w_2 x$$



University of Salford MANCHESTER

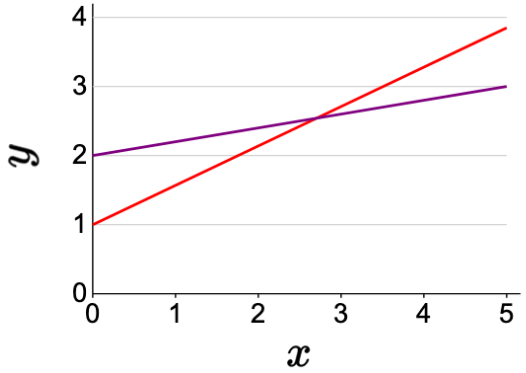
how to compute the best?

Lecture - Week 4 – ML - Linear regression how to compute the best?

training data

	x	y
example	1	1
example	2	3
example	4	3

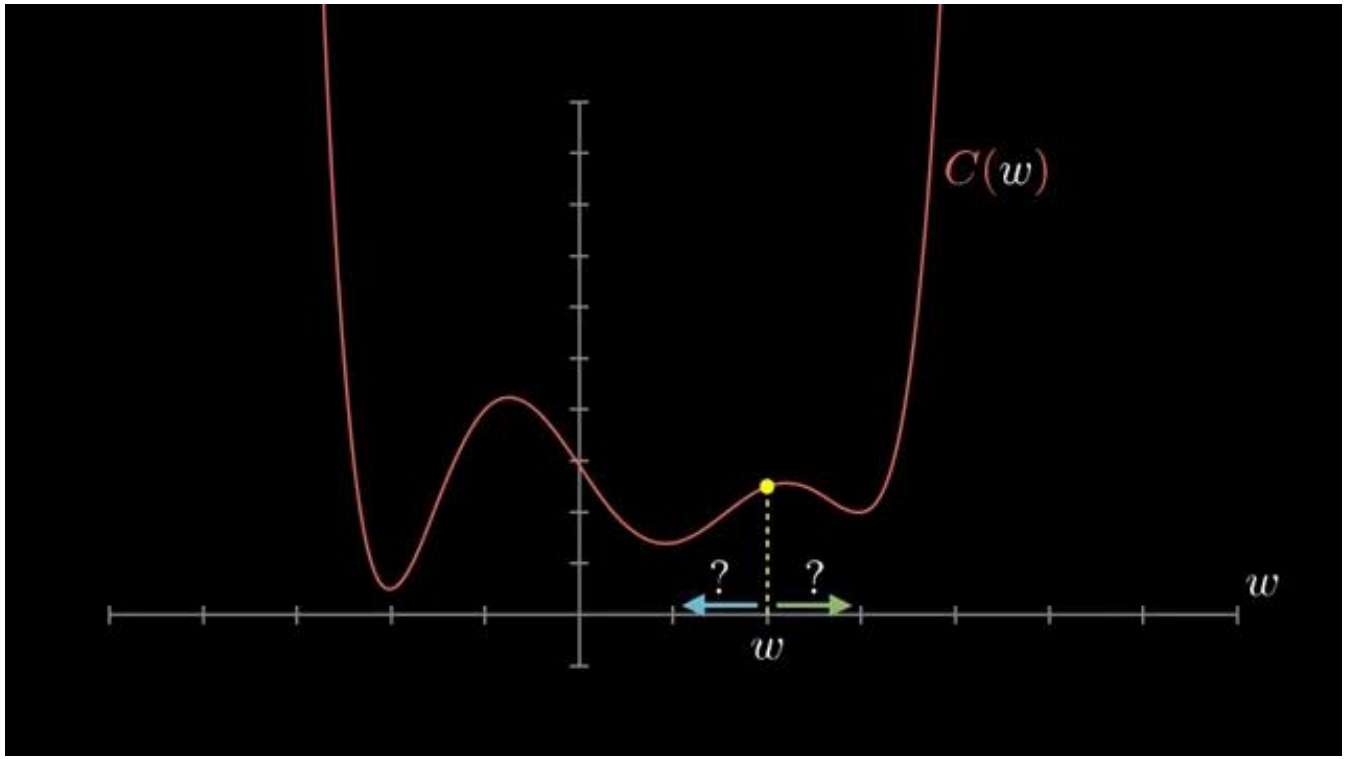
$$f(x) = w_1 + w_2 x$$



Lecture - Week 4 – ML - Linear regression
Optimization algorithm: how to compute the best?

University of Salford MANCHESTER

Gradient Descent (GD)

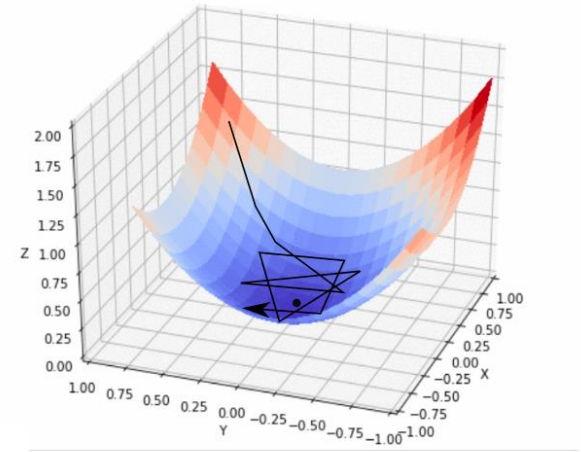


Lecture - Week 4 – ML - Linear regression
Optimization algorithm: GD

Goal: $\min_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$

Definition: gradient

The gradient $\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$ is the direction that increases the training loss the most.



Algorithm: gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$

For $t = 1, \ldots, T$: epochs

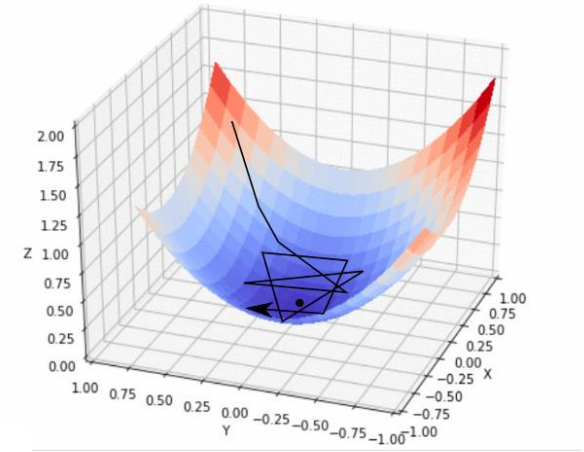
$$\mathbf{w} \leftarrow \mathbf{w} - \underbrace{\eta}_{\text{step size}} \underbrace{\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})}_{\text{gradient}}$$

Lecture - Week 4 – ML - Linear regression
Optimization algorithm: SGD

Goal: $\min_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$

Definition: gradient

The gradient $\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$ is the direction that increases the training loss the most.



Algorithm: stochastic gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$
For $t = 1, \ldots, T$:
  For $(x, y) \in \mathcal{D}_{\text{train}}$:
    $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \text{Loss}(x, y, \mathbf{w})$

Lecture - Week 4 – ML - Linear regression
Optimization algorithm: GD

University of Salford MANCHESTER

$$f(x) = 1 + 0.57x$$

$$\text{Loss}(1, 1, [1, 0.57]) = ([1, 0.57] \cdot [1, 1] - 1)^2 = 0.32$$
$$\text{Loss}(2, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 2] - 3)^2 = 0.74$$
$$\text{Loss}(4, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 4] - 3)^2 = 0.08$$

Algorithm: gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$

For $t = 1, \ldots, T$: epochs

$$\mathbf{w} \leftarrow \mathbf{w} - \underbrace{\eta}_{\text{step size}} \underbrace{\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})}_{\text{gradient}}$$

Objective function:

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} (\mathbf{w} \cdot \phi(x) - y)^2 = \frac{1}{3} (0.32 + 0.74 + 0.08) = 0.38$$

Lecture - Week 4 – ML - Linear regression
Optimization algorithm: GD

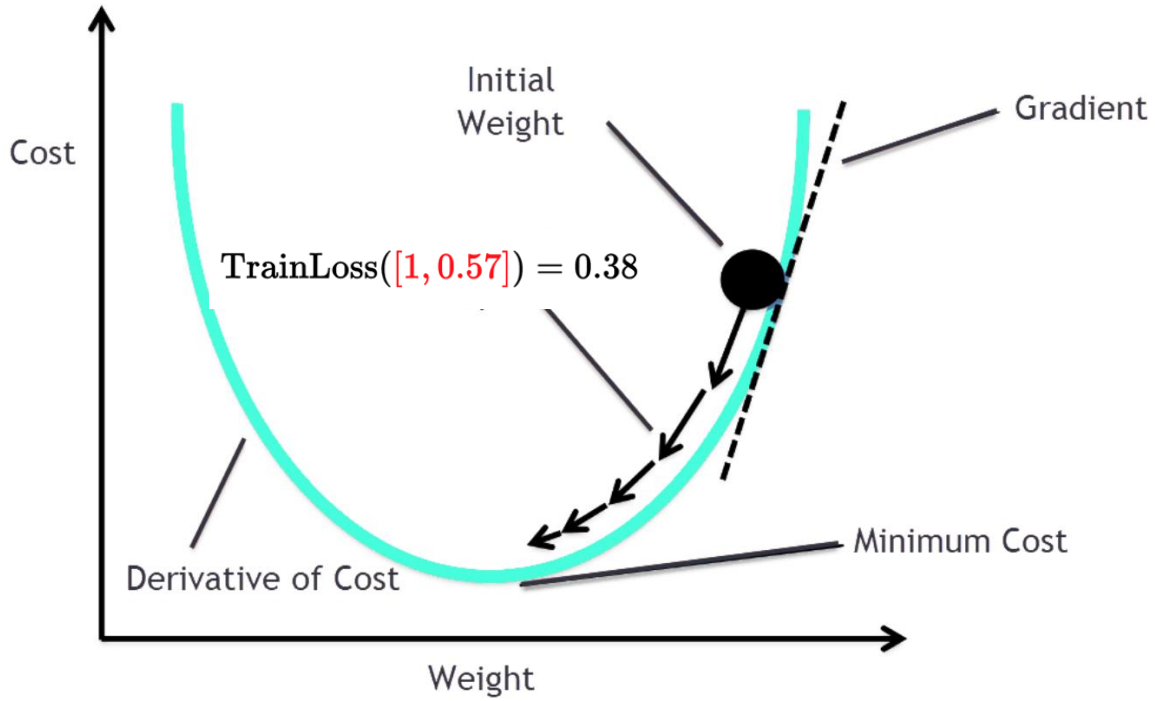
$$f(x) = 1 + 0.57x$$

$$\text{Loss}(1, 1, [1, 0.57]) = ([1, 0.57] \cdot [1, 1] - 1)^2 = 0.32$$
$$\text{Loss}(2, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 2] - 3)^2 = 0.74$$
$$\text{Loss}(4, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 4] - 3)^2 = 0.08$$

Objective function:

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} (\mathbf{w} \cdot \phi(x) - y)^2$$

$$\text{TrainLoss}([1, 0.57]) = 0.38$$



Lecture - Week 4 – ML - Linear regression
Optimization algorithm: GD

University of Salford MANCHESTER

$$f(x) = 1 + 0.57x$$

$$\text{Loss}(1, 1, [1, 0.57]) = ([1, 0.57] \cdot [1, 1] - 1)^2 = 0.32$$
$$\text{Loss}(2, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 2] - 3)^2 = 0.74$$
$$\text{Loss}(4, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 4] - 3)^2 = 0.08$$

Algorithm: gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$

For $t = 1, \ldots, T$: epochs

$$\mathbf{w} \leftarrow \mathbf{w} - \underbrace{\eta}_{\text{step size}} \underbrace{\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})}_{\text{gradient}}$$

Objective function:

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} (\mathbf{w} \cdot \phi(x) - y)^2$$

$$\text{TrainLoss}([1, 0.57]) = 0.38$$

Gradient (use chain rule):

$$\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} 2(\underbrace{\mathbf{w} \cdot \phi(x) - y}_{\text{prediction-target}})\phi(x)$$

How many Ws

$[w_1, w_2]$

Lecture - Week 4 – ML - Linear regression
Optimization algorithm: GD

University of Salford MANCHESTER

$$f(x) = 1 + 0.57x$$

$$\text{Loss}(1, 1, [1, 0.57]) = ([1, 0.57] \cdot [1, 1] - 1)^2 = 0.32$$
$$\text{Loss}(2, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 2] - 3)^2 = 0.74$$
$$\text{Loss}(4, 3, [1, 0.57]) = ([1, 0.57] \cdot [1, 4] - 3)^2 = 0.08$$

Algorithm: gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$

For $t = 1, \ldots, T$: epochs

$$\mathbf{w} \leftarrow \mathbf{w} - \underbrace{\eta}_{\text{step size}} \underbrace{\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})}_{\text{gradient}}$$

Objective function:

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} (\mathbf{w} \cdot \phi(x) - y)^2$$

$Cost$ = TrianLoss(w)

$$\text{TrainLoss}([1, 0.57]) = 0.38$$

$$[w_1, w_2] = [1,0.57] - \eta[1.14 - 2.28]$$

η = 0.1

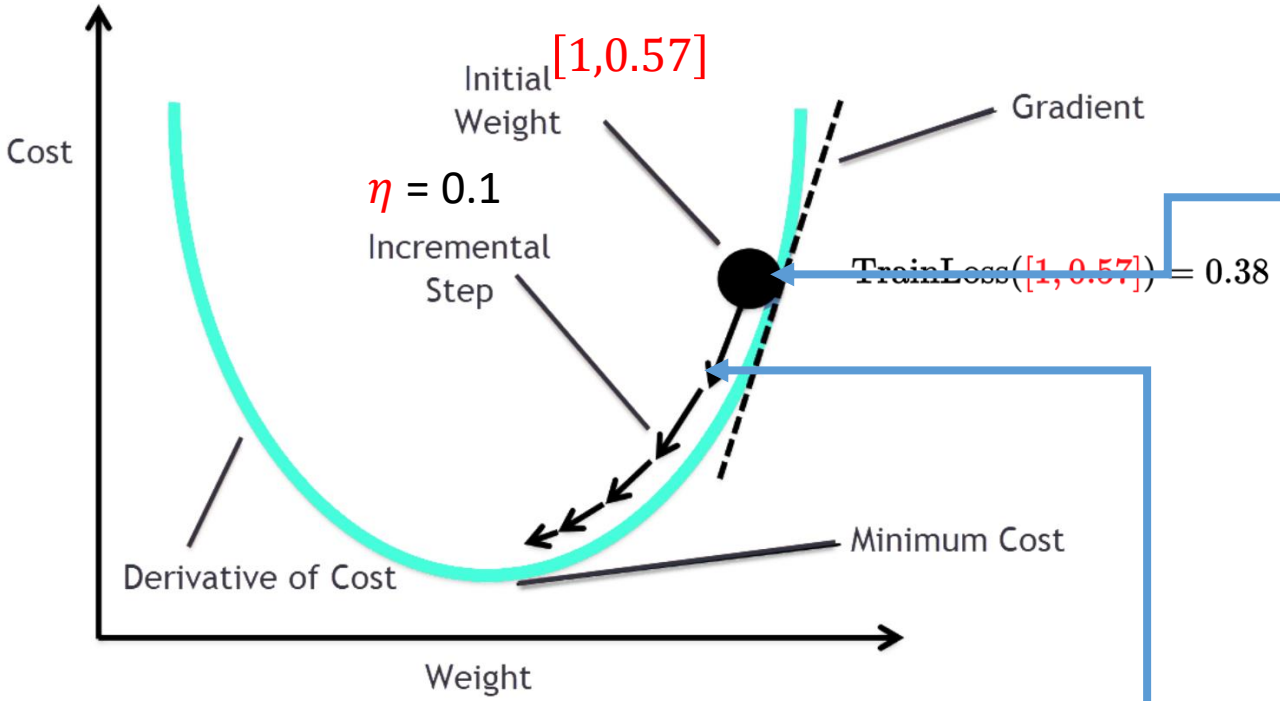
Gradient (use chain rule):

$$\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} 2(\underbrace{\mathbf{w} \cdot \phi(x) - y}_{\text{prediction-target}})\phi(x)$$

$$\frac{d\ Cost}{dw_1} = 1.14$$

$$\frac{d\ Cost}{dw_2} = 2.28$$

Lecture - Week 4 – ML - Linear regression
Optimization algorithm: GD



Algorithm: gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$

For $t = 1, \ldots, T$: epochs

$$\mathbf{w} \leftarrow \mathbf{w} - \underbrace{\eta}_{\text{step size}} \underbrace{\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})}_{\text{gradient}}$$

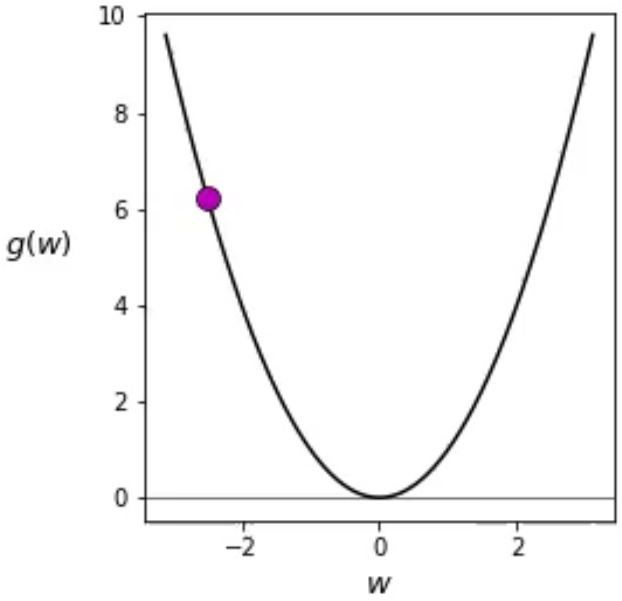
$$[w_1, w_2] = [1,0.57] - \eta[1.14 - 2.28]$$

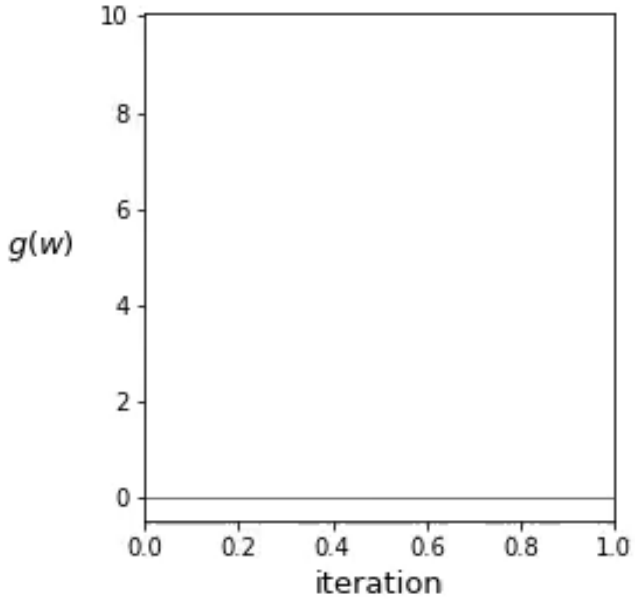
η = 0.1

$$[w_1, w_2] = [1,0.57] - 0.1[1.14 - 2.28]$$

$[w_1, w_2]$ = [0.89, 0.34]

Lecture - Week 4 – ML - Linear regression
Optimization algorithm: GD





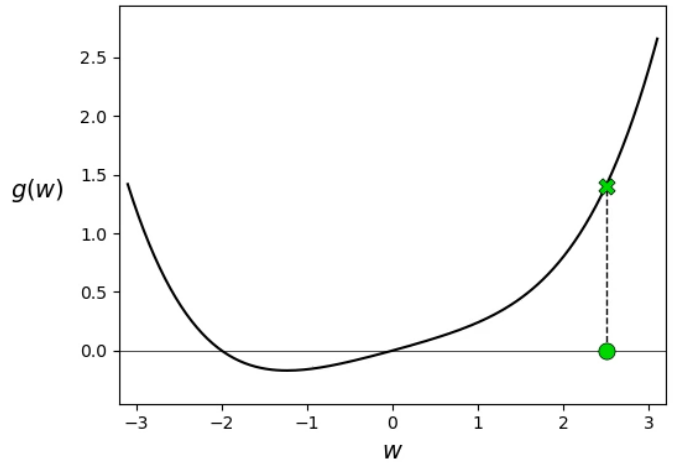
Algorithm: gradient descent

Initialize $\mathbf{w} = [0, \ldots, 0]$

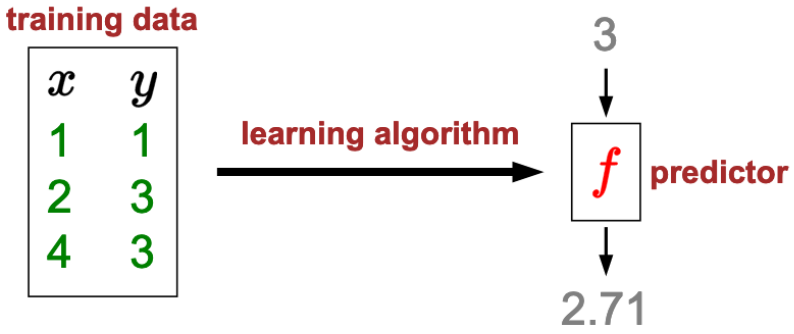
For $t = 1, \ldots, T$: epochs

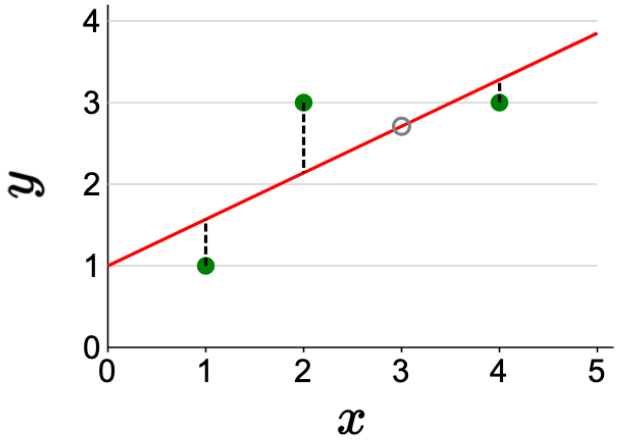
$$\mathbf{w} \leftarrow \mathbf{w} - \underbrace{\eta}_{\text{step size}} \underbrace{\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})}_{\text{gradient}}$$

$[w_1, w_2]$



Lecture - Week 4 – ML - Linear regression Quick Summary





Which predictors are possible?
Hypothesis class

How good is a predictor?
Loss function

How to compute best predictor?
Optimization algorithm

Linear functions
$$\mathcal{F} = \{f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)\}, \phi(x) = [1, x]$$

Squared loss
$$\text{Loss}(x, y, \mathbf{w}) = (f_{\mathbf{w}}(x) - y)^2$$

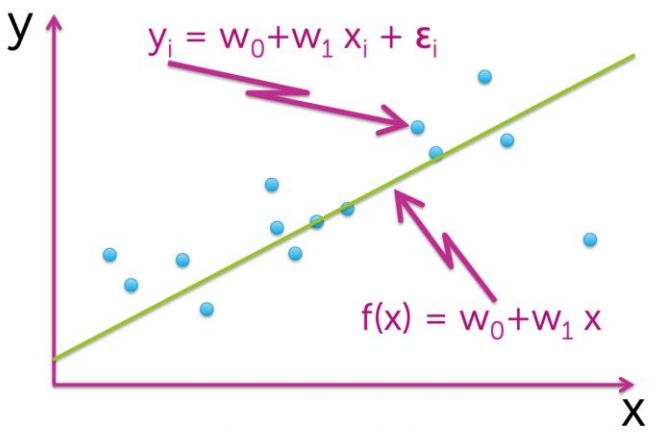
Gradient descent
$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla \text{TrainLoss}(\mathbf{w})$$

Lecture - Week 4 –Linear regression Least Squares in Matrix Form (Alternative method)

Our data consists of n paired observations of the predictor variable X and the response variable Y, i.e., $(x_1, y_1), \ldots (x_n, y_n)$. We wish to fit the model

$$Y = \beta_0 + \beta_1 X + \epsilon$$

$[w_0, w_1]$



Lecture - Week 4 –Linear regression Least Squares in Matrix Form

University of Salford MANCHESTER

Our data consists of n paired observations of the predictor variable X and the response variable Y, i.e., $(x_1, y_1), \ldots (x_n, y_n)$. We wish to fit the model

$$Y = \beta_0 + \beta_1 X + \epsilon$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \qquad \mathbf{x}\beta = \begin{bmatrix} \beta_0 + \beta_1 x_1 \\ \beta_0 + \beta_1 x_2 \\ \vdots \\ \beta_0 + \beta_1 x_n \end{bmatrix}$$

Lecture - Week 4 –Linear regression
Least Squares - Mean Squared Error

University of Salford MANCHESTER

$$\text{Loss}_{\text{squared}}(x, y, \mathbf{w}) = \underbrace{(f_{\mathbf{w}}(x) - y)}_{\text{residual}}^2$$

Objective function:

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} (\mathbf{w} \cdot \phi(x) - y)^2$$

$$\mathbf{e}(\beta) = \mathbf{y} - \mathbf{x}\beta$$

$$MSE(\beta) = \frac{1}{n} \sum_{i=1}^{n} e_i^2(\beta)$$

Lecture - Week 4 –Linear regression
Least Squares - Mean Squared Error

University of Salford MANCHESTER

$$\text{Loss}_{\text{squared}}(x, y, \mathbf{w}) = (\underbrace{f_{\mathbf{w}}(x) - y}_{\text{residual}})^2$$

Objective function:

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} (\mathbf{w} \cdot \phi(x) - y)^2$$

$$\begin{aligned} MSE(\beta) &= \frac{1}{n}\mathbf{e}^T\mathbf{e} \\ &= \frac{1}{n}(\mathbf{y} - \mathbf{x}\beta)^T(\mathbf{y} - \mathbf{x}\beta) \\ &= \frac{1}{n}(\mathbf{y}^T - \beta^T\mathbf{x}^T)(\mathbf{y} - \mathbf{x}\beta) \\ &= \frac{1}{n}\left(\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{x}\beta - \beta^T\mathbf{x}^T\mathbf{y} + \beta^T\mathbf{x}^T\mathbf{x}\beta\right) \end{aligned}$$

$$MSE(\beta) = \frac{1}{n}\left(\mathbf{y}^T\mathbf{y} - 2\beta^T\mathbf{x}^T\mathbf{y} + \beta^T\mathbf{x}^T\mathbf{x}\beta\right)$$

Lecture - Week 4 –Linear regression Minimizing the MSE

University of Salford MANCHESTER

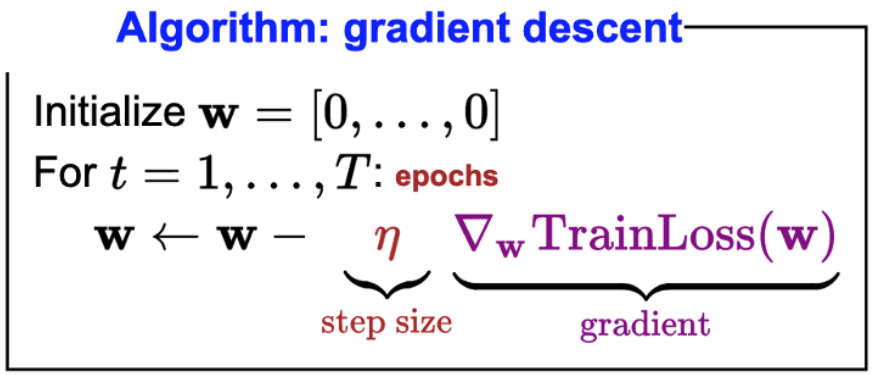
$$\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} 2(\underbrace{\mathbf{w}\cdot\phi(x) - y}_{\text{prediction}-\text{target}})\phi(x)$$

$$\frac{d\ Cost}{dw_0} =$$

$$\frac{d\ Cost}{dw_1} =$$

$$\begin{aligned} \nabla MSE(\beta &= \frac{1}{n}\left(\nabla \mathbf{y}^T\mathbf{y} - 2\nabla\beta^T\mathbf{x}^T\mathbf{y} + \nabla\beta^T\mathbf{x}^T\mathbf{x}\beta\right) \\ &= \frac{1}{n}\left(0 - 2\mathbf{x}^T\mathbf{y} + 2\mathbf{x}^T\mathbf{x}\beta\right) \\ &= \frac{2}{n}\left(\mathbf{x}^T\mathbf{x}\beta - \mathbf{x}^T\mathbf{y}\right) \end{aligned}$$

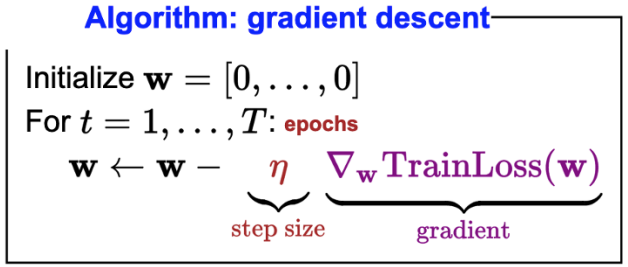
Lecture - Week 4 –Linear regression Minimizing the MSE



$$\mathbf{x}^T\mathbf{x}\widehat{\beta} - \mathbf{x}^T\mathbf{y} = 0$$

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}$$

Lecture - Week 4 –Linear regression Minimizing the MSE



$$\mathbf{x}^T\mathbf{x}\hat{\beta} - \mathbf{x}^T\mathbf{y} = 0$$

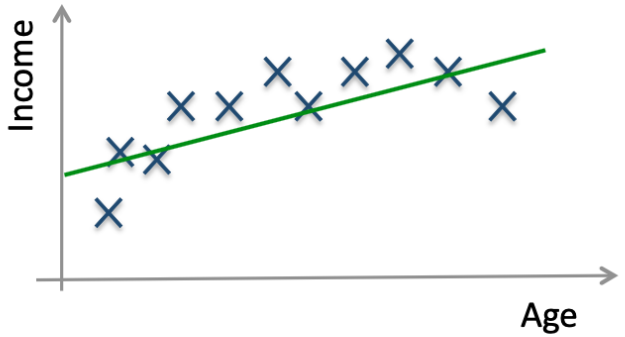
$$\hat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}$$

```
from sklearn.linear_model import SGDRegressor

reg = SGDRegressor(learning_rate=0.01, loss= 'squared_error',
max_iter=1000)
Reg.fit(X_train, y_train)
```

max_iter = epochs

```
from sklearn.linear_model import LinearRegression

reg = LinearRegression().fit(X_train, y_train)
Reg.fit(X_train, y_train)
```

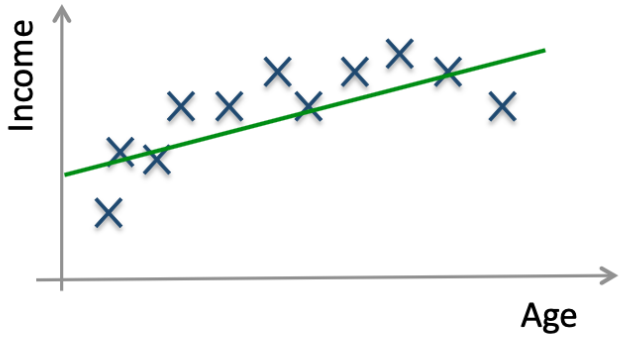
Lecture - Week 4 –Linear regression

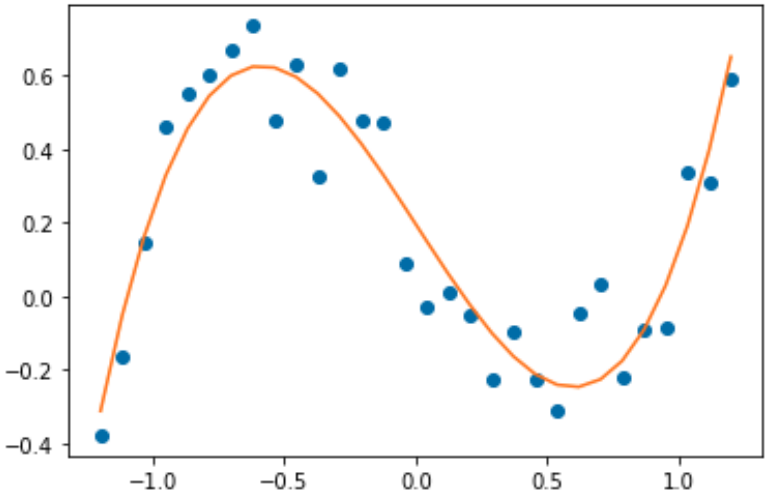


Linear !!!

Yes
No

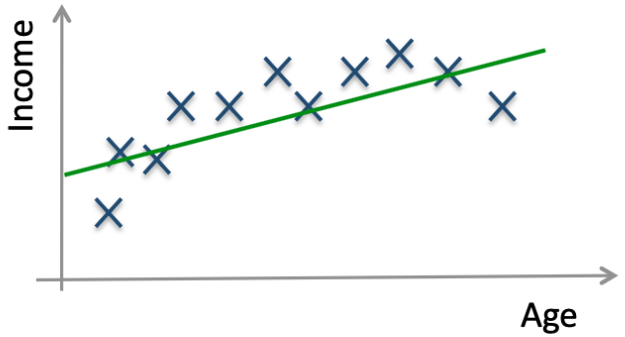
Lecture - Week 4 –Linear regression

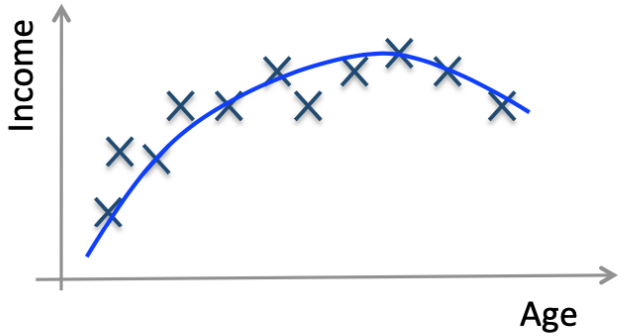




No

Lecture - Week 4 –Linear regression
Polynomial regression

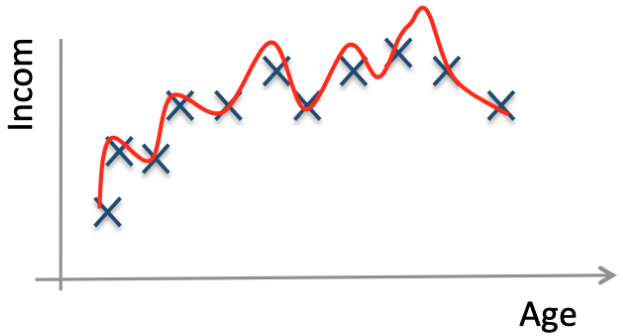


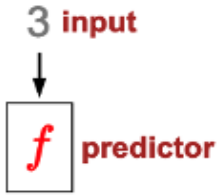


Better!

$$f(x) = w_0 + w_1 x$$

$$f(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_p x^p$$





validation data

3 4

Lecture - Week 4 – Regression

University of Salford MANCHESTER

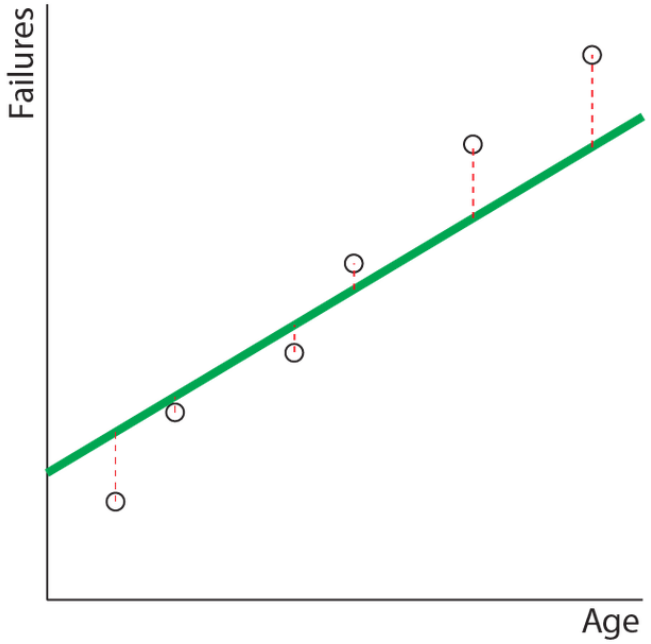
- **Mean Absolute Error (MAE)**

Lecture - Week 4 – Regression

University of Salford MANCHESTER

❑Mean Absolute Error (MAE)

$$MAE = \frac{\Sigma|y - \hat{y}|}{N}$$



Age	Failures	Prediction	Error	abs(Error)
10	15	26	11	11
20	30	32	2	2
40	40	44	4	4
50	55	50	-5	5
70	75	62	-13	13
90	90	74	-16	16

Mean abs(Error)	8.5

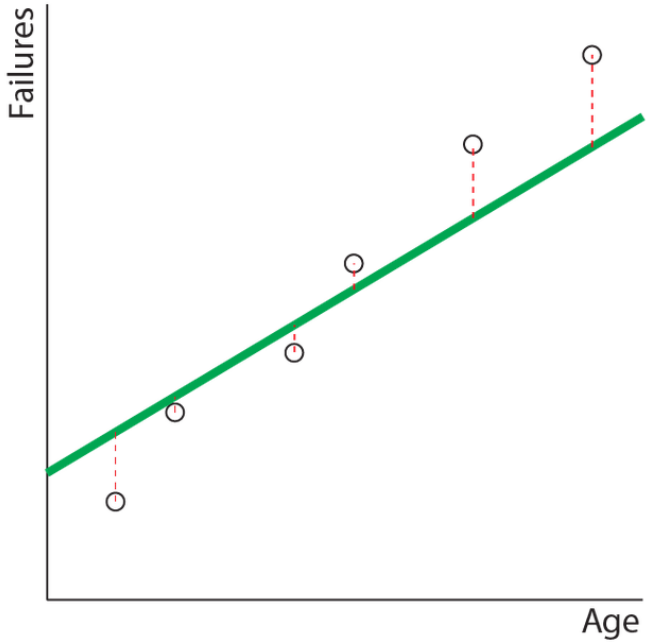
Lecture - Week 4 – Regression

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)

Lecture - Week 4 – Regression

❑Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{\Sigma(y - \hat{y})^2}{N}}$$



Here is the calculation for RMSE on our example scenario:

Age	Failures (y)	Prediction ($\hat{y}$)	Error ($y-\hat{y}$)	Error^2 ($(y-\hat{y})^2$)
10	15	26	11	121
20	30	32	2	4
40	40	44	4	16
50	55	50	-5	25
70	75	62	-13	169
90	90	74	-16	256

Mean of Error^2	$\frac{\Sigma(y-\hat{y})^2}{N}$	98.5
Square root of Mean of Error^2	$\sqrt{\frac{\Sigma(y-\hat{y})^2}{N}}$	**9.9**

Lecture - Week 4 – Regression

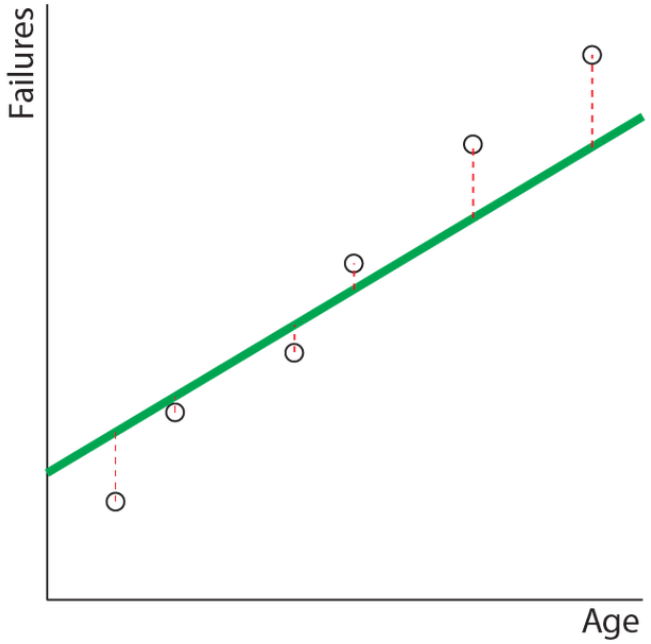
University of Salford MANCHESTER

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)
- R-Squared

Lecture - Week 4 – Regression

❑R-Squared

$$R^2 = \frac{\Sigma(y-\bar{y})^2 - \Sigma(y-\hat{y})^2}{\Sigma(y-\bar{y})^2} \qquad R^2 = \frac{var(mean) - var(line)}{var(mean)}$$



			Regression Line	Mean Line	Regression Line	Mean Line
	y	$\hat{y}$	$y-\hat{y}$	$y-\bar{y}$	$(y-\hat{y})^2$	$(y-\bar{y})^2$
Age	Failures	Prediction	Error	Error	Error^2	Error^2
10	15	26	11	-35.8	121	1281.6
20	30	32	2	-20.8	4	432.6
40	40	44	4	-10.8	16	116.6
50	55	50	-5	4.2	25	17.6
70	75	62	-13	24.2	169	585.6
90	90	74	-16	39.2	256	1536.6
Mean of Error^2					$\frac{\Sigma(y-\hat{y})^2}{N}$ 98.5	$\frac{\Sigma(y-\bar{y})^2}{N}$ 661.8
R^2					$\frac{\Sigma(y-\bar{y})^2-\Sigma(y-\hat{y})^2}{\Sigma(y-\bar{y})^2}$	**0.85**

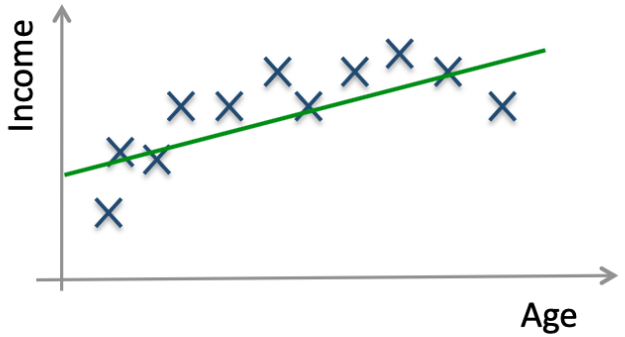
Lecture - Week 4 – Regression

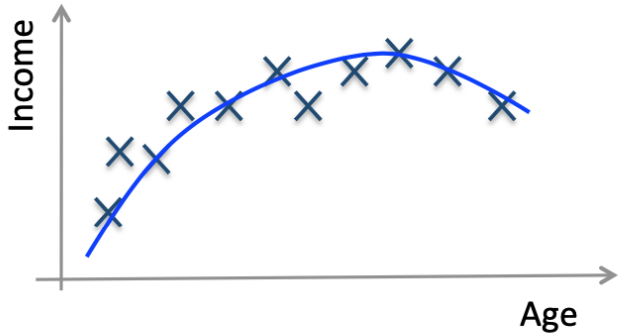
University of Salford MANCHESTER

Regression	
'explained_variance'	metrics.explained_variance_score
'max_error'	metrics.max_error
'neg_mean_absolute_error'	metrics.mean_absolute_error
'neg_mean_squared_error'	metrics.mean_squared_error
'neg_root_mean_squared_error'	metrics.mean_squared_error
'neg_mean_squared_log_error'	metrics.mean_squared_log_error
'neg_median_absolute_error'	metrics.median_absolute_error
'r2'	metrics.r2_score
'neg_mean_poisson_deviance'	metrics.mean_poisson_deviance
'neg_mean_gamma_deviance'	metrics.mean_gamma_deviance
'neg_mean_absolute_percentage_error'	metrics.mean_absolute_percentage_error

Lecture - Week 4 – Linear regression
More Next lectures

$R^2 = 0.82$
$R^2 = 0.79$

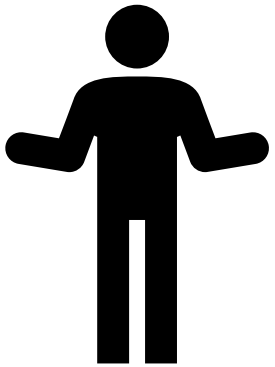


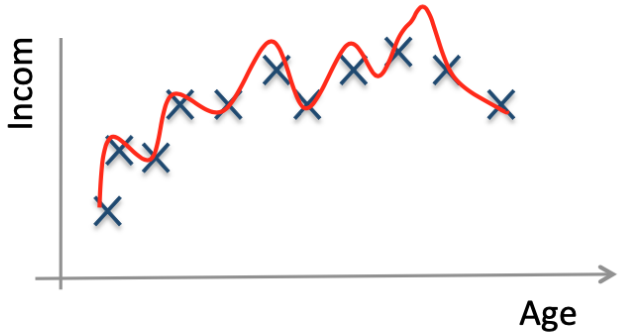


$R^2 = 0.92$
$R^2 = 0.90$

$$f(x) = w_0 + w_1 x$$

$$f(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_p x^p$$

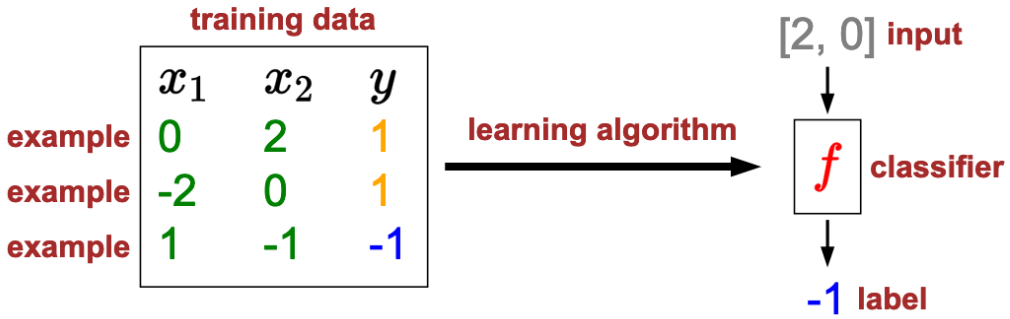


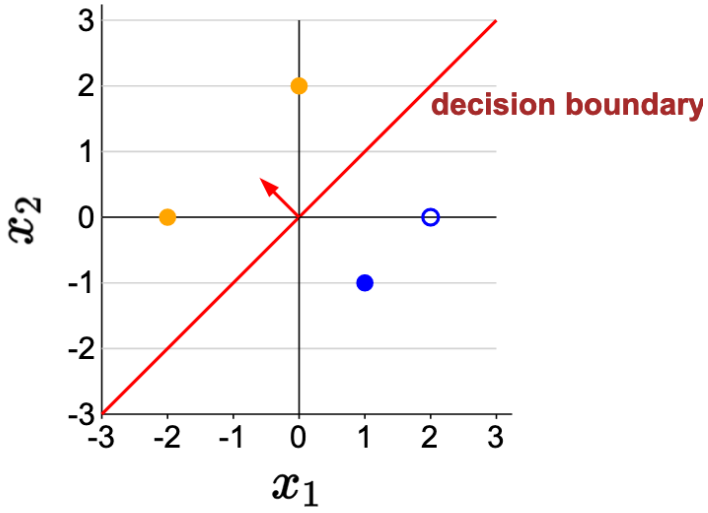


$R^2 = 1.0$
$R^2 = 0.6$

Classification Model

Lecture - Week 4 – ML
Linear classification framework





Design decisions:

Which classifiers are possible? **hypothesis class**

How good is a classifier? **loss function**

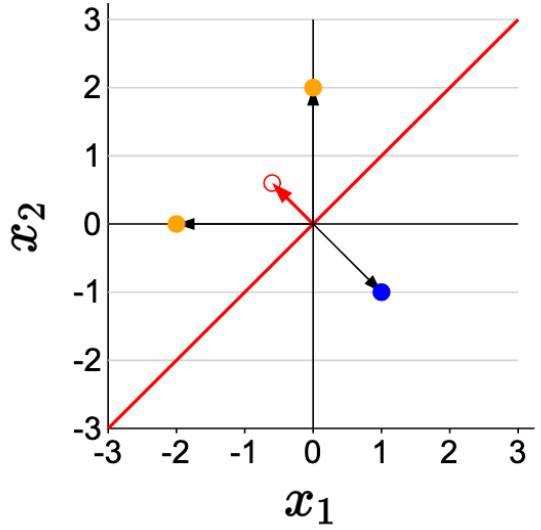
How do we compute the best classifier? **optimization algorithm**

Lecture - Week 4 – ML
Linear classification framework

$$f(x) = \text{sign}(\overbrace{[-0.6, 0.6]}^{\mathbf{w}} \cdot \overbrace{[x_1, x_2]}^{\phi(x)})$$

$$\text{sign}(z) = \begin{cases} +1 & \text{if } z > 0 \\ -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \end{cases}$$

x_1	x_2	$f(x)$
0	2	1
-2	0	1
1	-1	-1



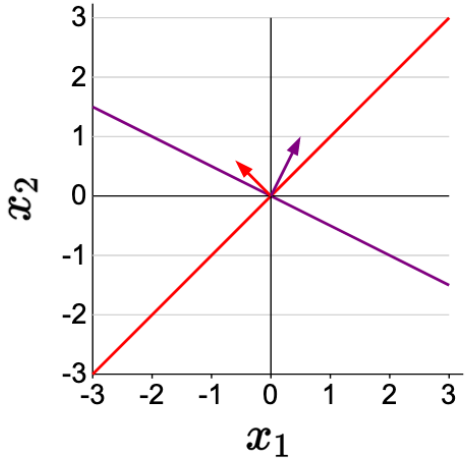
$$f([0, 2]) = \text{sign}([-0.6, 0.6] \cdot [0, 2]) = \text{sign}(1.2) = 1$$
$$f([-2, 0]) = \text{sign}([-0.6, 0.6] \cdot [-2, 0]) = \text{sign}(1.2) = 1$$
$$f([1, -1]) = \text{sign}([-0.6, 0.6] \cdot [1, -1]) = \text{sign}(-1.2) = -1$$

Decision boundary: x such that $\mathbf{w} \cdot \phi(x) = 0$

Lecture - Week 4 – ML
Hypothesis class: which classifiers?

University of Salford MANCHESTER

$$\phi(x) = [x_1, x_2]$$
$$f(x) = \text{sign}([-0.6, 0.6] \cdot \phi(x))$$
$$f(x) = \text{sign}([0.5, 1] \cdot \phi(x))$$



General binary classifier:

$$f_{\mathbf{w}}(x) = \text{sign}(\mathbf{w} \cdot \phi(x))$$

Hypothesis class:

$$\mathcal{F} = \{f_{\mathbf{w}} : \mathbf{w} \in \mathbb{R}^2\}$$

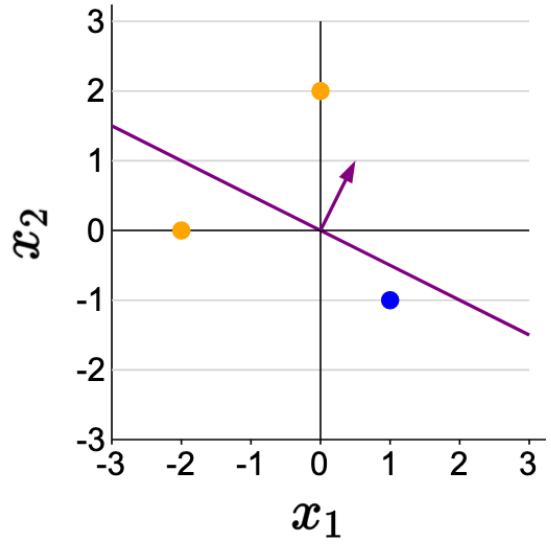
Lecture - Week 4 – ML
Loss function: how good is a classifier?

University of Salford MANCHESTER

$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$$
$\mathbf{w} = [0.5, 1]$
$\phi(x) = [x_1, x_2]$

training data $\mathcal{D}_{\text{train}}$

x_1	x_2	y
0	2	1
-2	0	1
1	-1	-1



$$\text{Loss}_{\text{0-1}}(x, y, \mathbf{w}) = \mathbf{1}[f_{\mathbf{w}}(x) \neq y]$$ zero-one loss

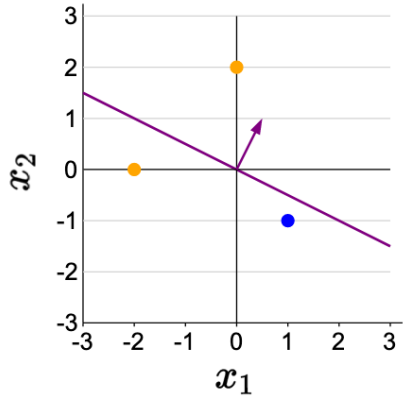
$$\text{Loss}([0, 2], 1, [0.5, 1]) = \mathbf{1}[\text{sign}([0.5, 1] \cdot [0, 2]) \neq 1] = 0$$
$$\text{Loss}([-2, 0], 1, [0.5, 1]) = \mathbf{1}[\text{sign}([0.5, 1] \cdot [-2, 0]) \neq 1] = 1$$
$$\text{Loss}([1, -1], -1, [0.5, 1]) = \mathbf{1}[\text{sign}([0.5, 1] \cdot [1, -1]) \neq -1] = 0$$

$$\text{TrainLoss}([0.5, 1]) = 0.33$$

Lecture - Week 4 – ML
Score and margin

Predicted label: $f_{\mathbf{w}}(x) = \text{sign}(\mathbf{w} \cdot \phi(x))$

Target label: y



Definition: score

The score on an example (x, y) is $\mathbf{w} \cdot \phi(x)$, how **confident** we are in predicting +1.

Definition: margin

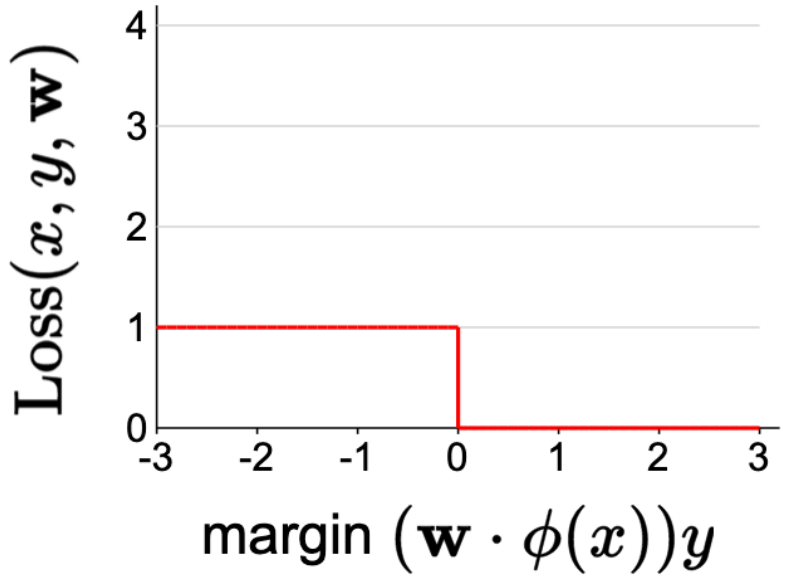
The margin on an example (x, y) is $(\mathbf{w} \cdot \phi(x))y$, how **correct** we are.

Lecture - Week 4 – ML
Zero-one loss: how good is a classifier?

University of Salford MANCHESTER

Definition: zero-one loss

$$\begin{aligned}\text{Loss}_{0\text{-}1}(x, y, \mathbf{w}) &= \mathbf{1}[f_{\mathbf{w}}(x) \neq y] \\ &= \mathbf{1}[\underbrace{(\mathbf{w} \cdot \phi(x))y}_{\text{margin}} \leq 0]\end{aligned}$$



Lecture - Week 4 – ML
Optimization algorithm: how to compute the best?

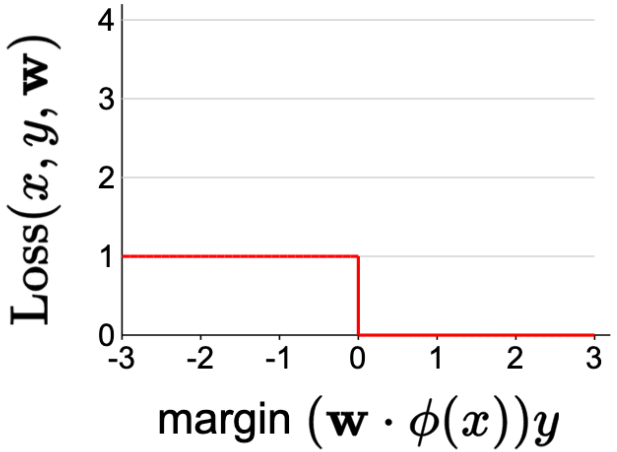
University of Salford MANCHESTER

Goal: $\min_{\mathbf{w}} \text{TrainLoss}(\mathbf{w})$

To run gradient descent, compute the gradient:

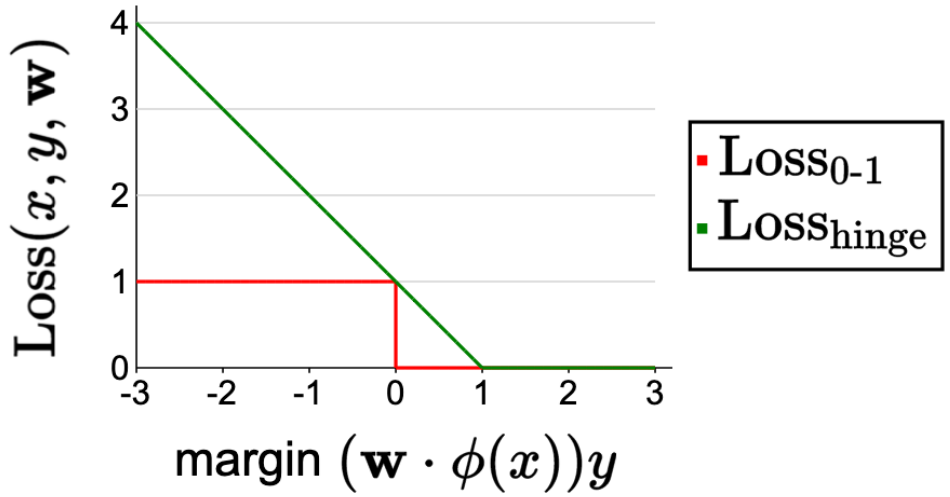
$$\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y)\in\mathcal{D}_{\text{train}}} \nabla \text{Loss}_{0\text{-}1}(x, y, \mathbf{w})$$

$$\nabla_{\mathbf{w}} \text{Loss}_{0\text{-}1}(x, y, \mathbf{w}) = \nabla \mathbf{1}[(\mathbf{w} \cdot \phi(x))y \leq 0]$$



What is the Problem!

Lecture - Week 4 – ML
Optimization algorithm: how to compute the best?
Hinge loss

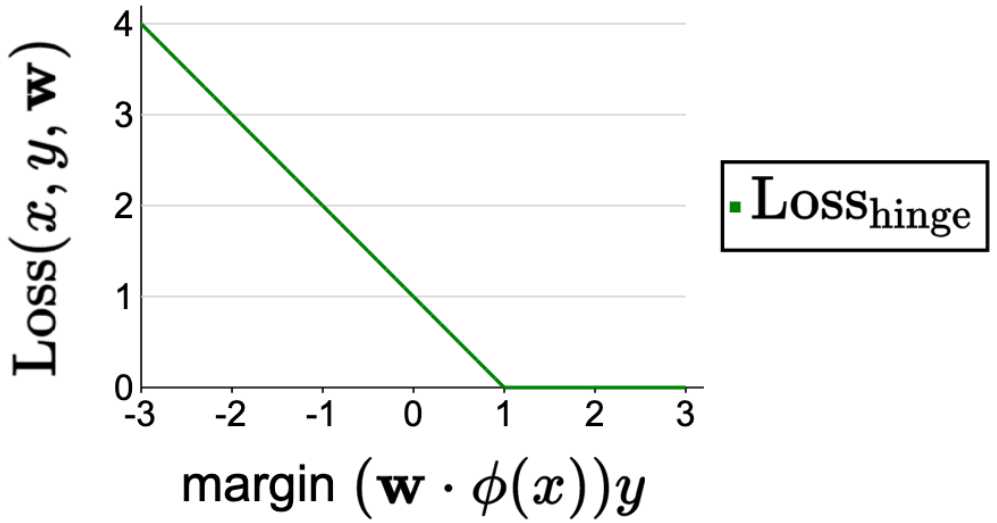


$$\text{Loss}_{\text{hinge}}(x, y, \mathbf{w}) = \max\{1 - (\mathbf{w} \cdot \phi(x))y, 0\}$$

A point has zero loss if it is confidently classified correctly (margin > 1)

Confidently misclassifying a point incurs a linear penalty

Lecture - Week 4 – ML
Optimization algorithm: how to compute the best?
Hinge loss/Gradient



$$\text{Loss}_{\text{hinge}}(x, y, \mathbf{w}) = \max\{1 - (\mathbf{w} \cdot \phi(x))y, 0\}$$

$$\nabla \text{Loss}_{\text{hinge}}(x, y, \mathbf{w}) = \begin{cases} -\phi(x)y & \text{if } 1 > (\mathbf{w} \cdot \phi(x))y \\ 0 & \text{otherwise} \end{cases}$$

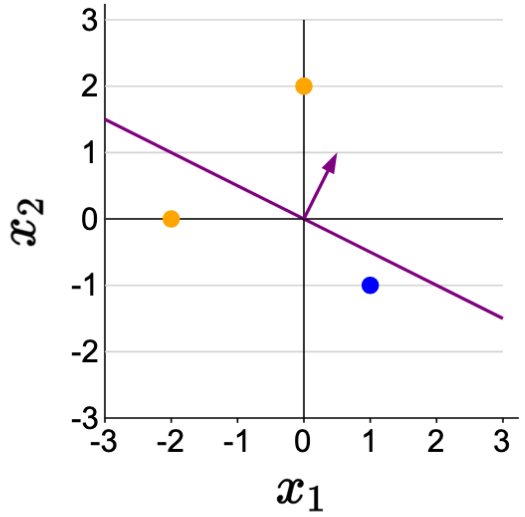
Lecture - Week 4 – ML
Optimization algorithm: how to compute the best?
Hinge loss on training data

University of Salford MANCHESTER

$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x)$$
$$\mathbf{w} = [0.5, 1]$$
$$\phi(x) = [x_1, x_2]$$

training data $\mathcal{D}_{\text{train}}$

x_1	x_2	y
0	2	1
-2	0	1
1	-1	-1



$$\text{Loss}_{\text{hinge}}(x, y, \mathbf{w}) = \max\{1 - (\mathbf{w} \cdot \phi(x))y, 0\}$$

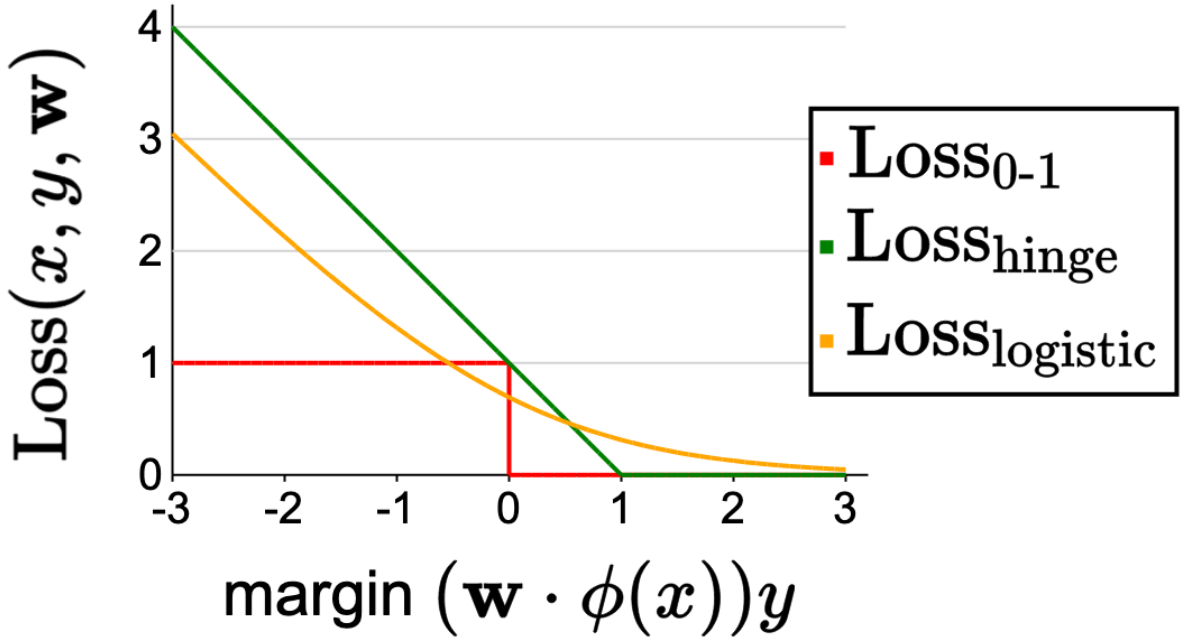
$$\text{Loss}([0,2], 1, [0.5,1]) = \max\{1 - [0.5,1] \cdot [0,2](1), 0\} = 0$$
$$\text{Loss}([-2,0], 1, [0.5,1]) = \max\{1 - [0.5,1] \cdot [-2,0](1), 0\} = 2$$
$$\text{Loss}([1,-1], -1, [0.5,1]) = \max\{1 - [0.5,1] \cdot [1,-1](-1), 0\} = 0.5$$
$$\text{TrainLoss}([0.5,1]) = 0.83$$

$$\nabla\text{Loss}([0,2], 1, [0.5,1]) = [0,0]$$
$$\nabla\text{Loss}([-2,0], 1, [0.5,1]) = [2,0]$$
$$\nabla\text{Loss}([1,-1], -1, [0.5,1]) = [1,-1]$$
$$\nabla\text{TrainLoss}([0.5,1]) = [1,-0.33]$$

Lecture - Week 4 – ML
Optimization algorithm: how to compute the best? logistic regression (Softmax)

University of Salford MANCHESTER

$$\text{Loss}_{\text{logistic}}(x, y, \mathbf{w}) = \log(1 + e^{-(\mathbf{w}\cdot\phi(x))y})$$

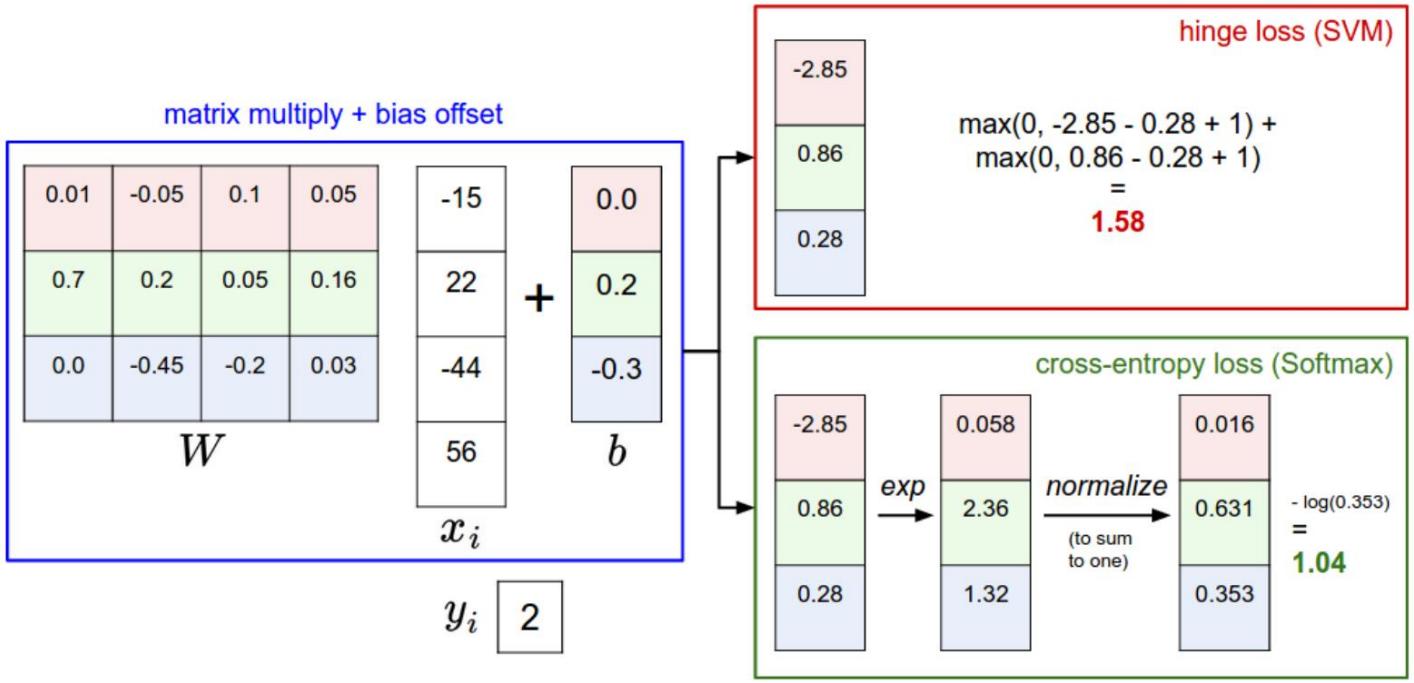


Intuition: Try to increase margin even when it already exceeds 1

Lecture - Week 4 – ML
Optimization algorithm: how to compute the best?
SVM vs. Softmax

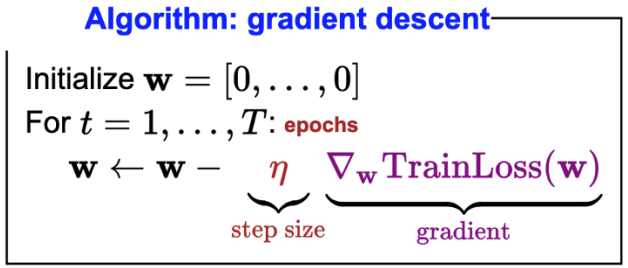
Correct class (class 2)

linear SVM



logistic regression

Lecture - Week 4 –Linear classifier




```
from sklearn.linear_model import SGDClassifier

sgd = SGDClassifier(learning_rate=0.01, loss= 'hinge', max_iter=1000)
sgd.fit(X_train, y_train)
```

Loss = log ➔ Logistic Regression

Lecture - Week 4 –Linear classifier

```
from sklearn.linear_model import SGDClassifier

sgd = SGDClassifier(learning_rate=0.01, loss= 'log',
max_iter=1000)
sgd.fit(X_train, y_train)
```

```
from sklearn.linear_model import LogisticRegression

sgd = LogisticRegression(solver='lbfgs',)
sgd.fit(X_train, y_train)
```

- 'newton-cg' - ['l2', 'none']
- 'lbfgs' - ['l2', 'none']
- 'liblinear' - ['l1', 'l2']
- 'sag' - ['l2', 'none']
- 'saga' - ['elasticnet', 'l1', 'l2', 'none']

Lecture - Week 4 Unsupervised Summary

University of Salford MANCHESTER

Model complexity
More Models