Machine (Deep) Learning
for 3D Understanding

Shenglan Du
26-03-2021

TU Delft

Source: https://www.searchenginejournal.com/machine-learning-seo/208415/

Today’s Agenda

- Machine Learning basics
  - Definition & Scope of machine learning
  - Bayes classifier
  - Linear classifier (Fisher, SVM)

- Deep Learning for 3D urban applications
  - Deep learning intuition
  - Deep neural networks for 3D classification and segmentation of point clouds

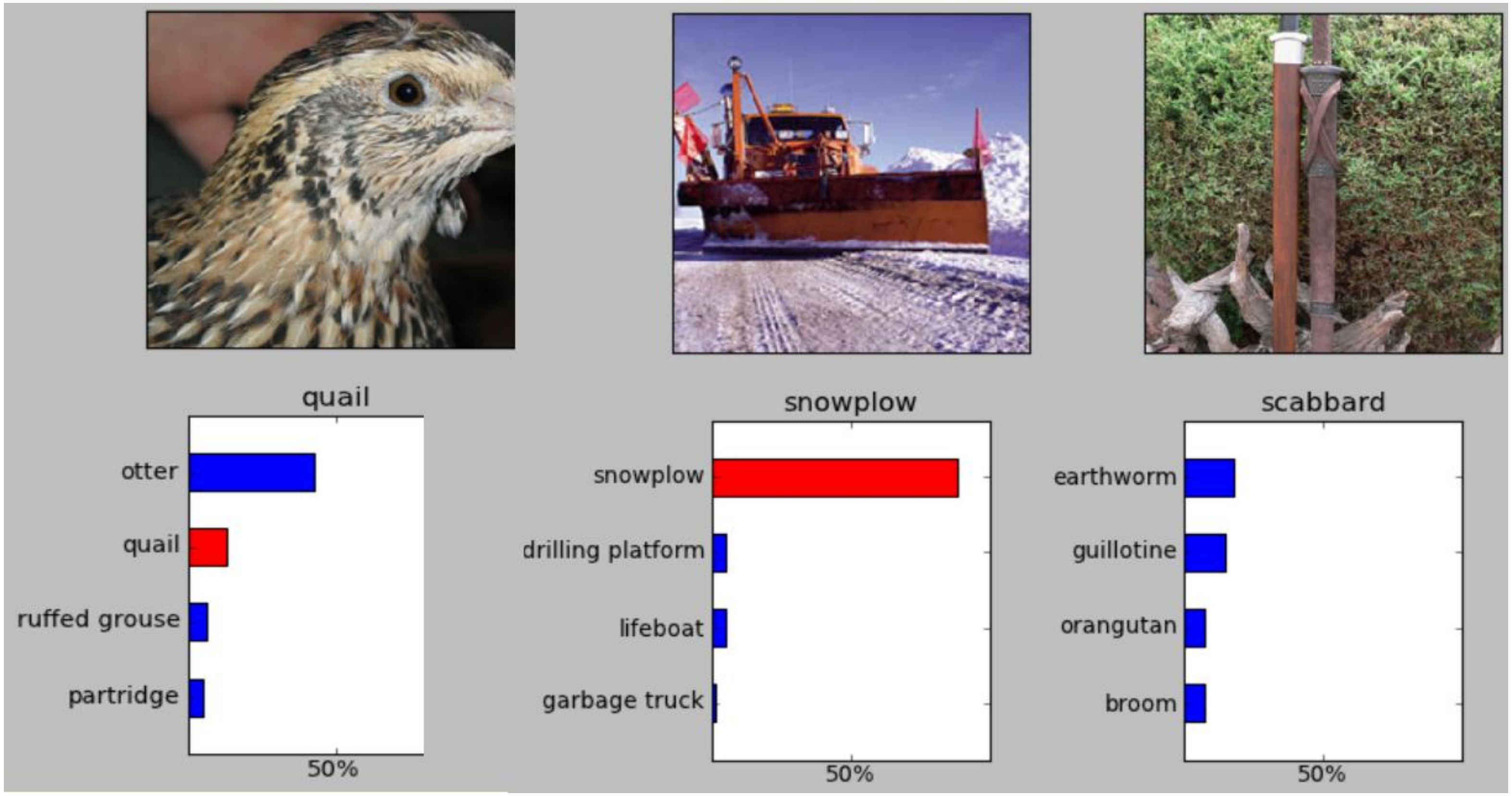
Today's Agenda

- Machine Learning basics
  - Definition & Scope of machine learning
  - Bayes classifier
  - Linear classifier (Fisher, SVM)

- Deep Learning for 3D urban applications
  - Deep learning intuition
  - Deep neural networks for 3D classification and segmentation of point clouds

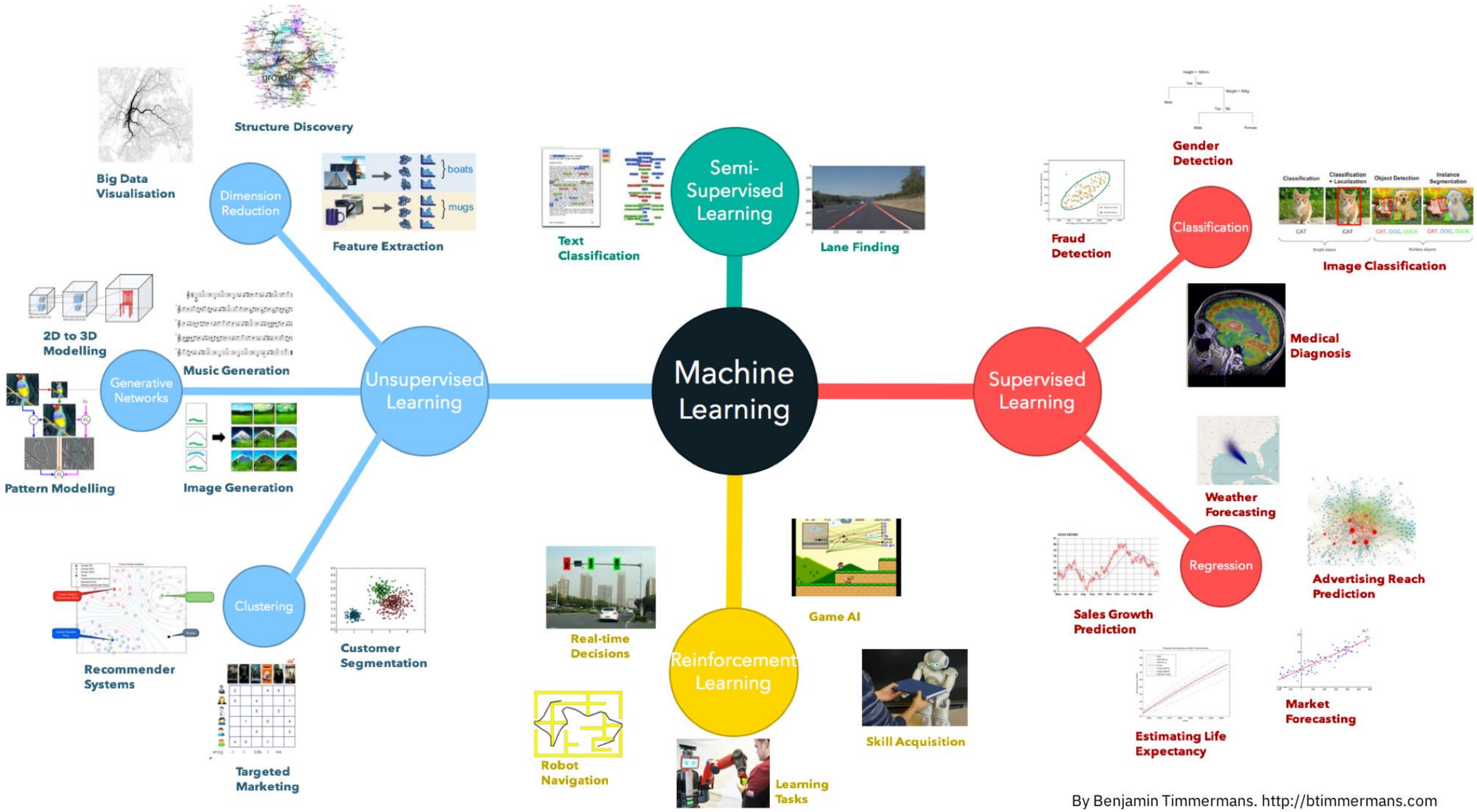
Machine Learning

- Learning from examples



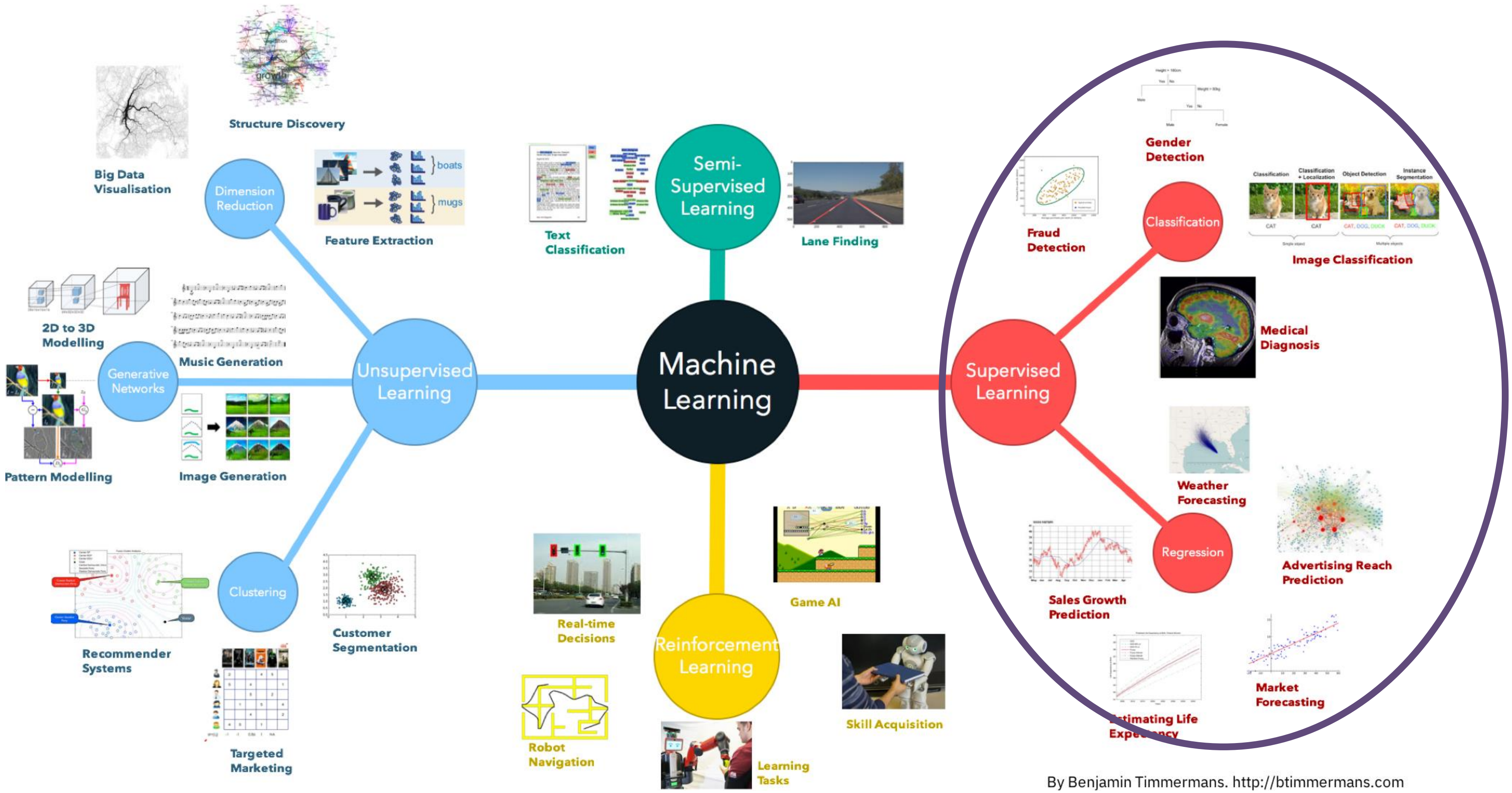
Source: https://www.cs.toronto.edu/~hinton/coursera_slides.html

Machine Learning Scope



TU Delft

Machine Learning Scope



By Benjamin Timmermans. http://btimmermans.com

TU Delft

Today’s Agenda

- Machine Learning basics
  - Definition & Scope of machine learning
  - Bayes classifier
  - Linear classifier (Fisher, SVM)

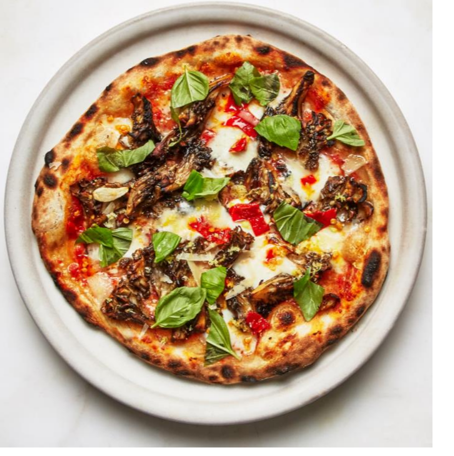
- Deep Learning for 3D urban applications
  - Deep learning intuition
  - Deep neural networks for 3D classification and segmentation of point clouds

Bayes Classifier

Source: Google image

Bayes Classifier

- A simple scenario:
  - If the day is **sunny**, I go for **pizza**
  - If it is **raining** that day, I go for **hotpot**



- One day you observed that I had **hotpot** as dinner. What the weather of that day do you guess to be?

TU Delft

Bayes Classifier

- It's very likely to be rainy!
- But how do machines interpretate the word "likely"?

Bayes Classifier: Probability basics

- Product rule:

$$P(A, B) = P(A)\,P(B|A)$$

- Bayes rule:

$$P(B)\,P(A|B) = P(A)\,P(B|A)$$

$$P(A|B) = \frac{P(A)\,P(B|A)}{P(B)}$$

Bayes Classifier: Probability basics

- We have an observation, assuming x is the measurement, and ω is the class label

$$P(\omega|x) = \frac{P(\omega)\,P(x|\omega)}{P(x)}$$

- P(x | ω): class conditional probability
- P(ω): class prior probability
- P(ω | x): class posterior probability

TU Delft

Bayes Classifier: Probability basics

- Recap the problem of weather prediction by food

- Assume equal priors in both sunny days and rainy days

$$P(\omega = s) = P(\omega = r) = 0.5$$

TU Delft

Bayes Classifier: Probability basics

- Assume we have the class conditional probabilities as follows

$$P(hotpot|\omega = s) = 0.2$$
$$P(pizza|\omega = s) = 0.8$$

$$P(hotpot|\omega = r) = 0.75$$
$$P(pizza|\omega = r) = 0.25$$

TU Delft

Bayes Classifier: Probability basics

- With the observation of “hotpot”, what is the posterior probability of the weather?

Bayes Classifier: Probability basics

Sunny:

$$P(\omega = s|hotpot) = \frac{P(\omega = s)\,P(hotpot|\omega = s)}{P(hotpot)}$$

$$= \frac{0.5 * 0.2}{P(hotpot)}$$

Rainy:

$$P(\omega = r|hotpot) = \frac{P(\omega = r)\,P(hotpot|\omega = r)}{P(hotpot)}$$

$$= \frac{0.5 * 0.75}{P(hotpot)}$$

Bayes Classifier: Probability basics

Sunny:

$$P(\omega = s|hotpot) = \frac{P(\omega = s)\, P(hotpot|\omega = s)}{P(hotpot)}$$

$$= \frac{0.5 * 0.2}{P(hotpot)}$$

Rainy:

$$P(\omega = r|hotpot) = \frac{P(\omega = r)\, P(hotpot|\omega = r)}{P(hotpot)}$$

$$= \frac{0.5 * 0.75}{P(hotpot)}$$

Bayes Classifier

- Prior:

$$P(\omega = s) = P(\omega = r)$$

- Posterior:

$$P(\omega = s|hotpot) \ll P(\omega = r|hotpot)$$

Bayes Classifier

- The Bayes rule provides an approach of describing the uncertainty quantitatively, allowing for the optimal prediction given the observations present.

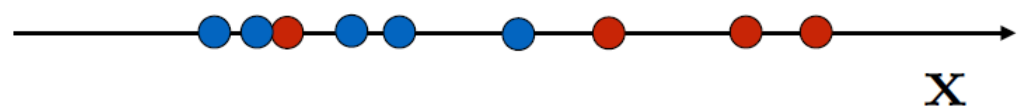
- Bayes serves as the foundation for the modern machine learning

Bayes Classifier

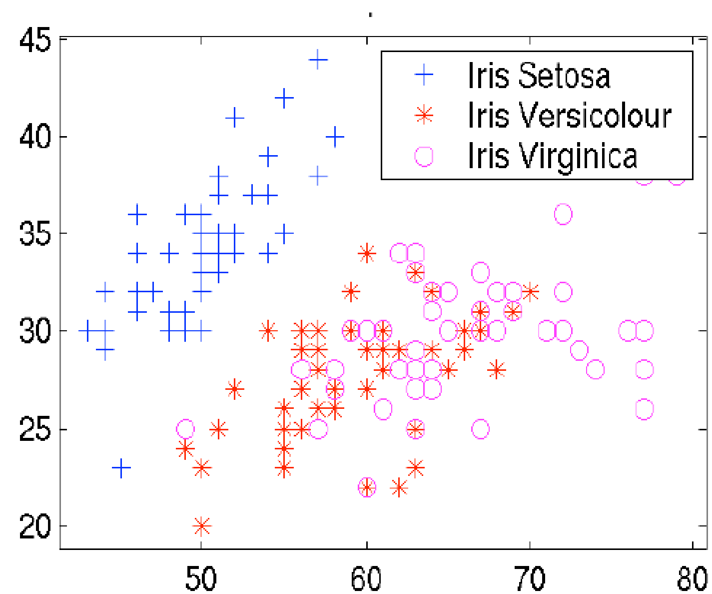
- We can interpretate the observations as a vector:

$$\boldsymbol{x} = (x_1, x_2, x_3 \dots x_p)^T$$

1D feature space:



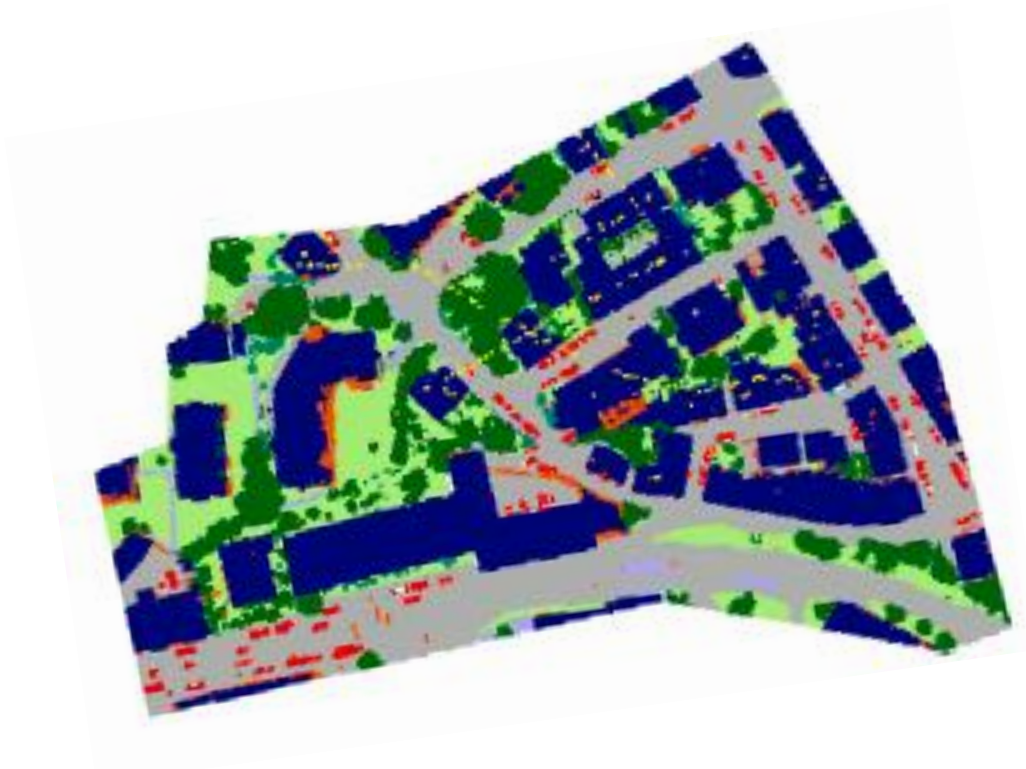
2D feature space:



Bayes Classifier

- Regarding point clouds:

$$\boldsymbol{x} = (x, y, z, r, g, b, intensity \dots)^T$$



Source: https://www.sciencedirect.com/science/article/pii/S0924271620300605

Bayes Classifier

- Each observation is linked with a label ω
- Bayes aims to model the joint distribution:

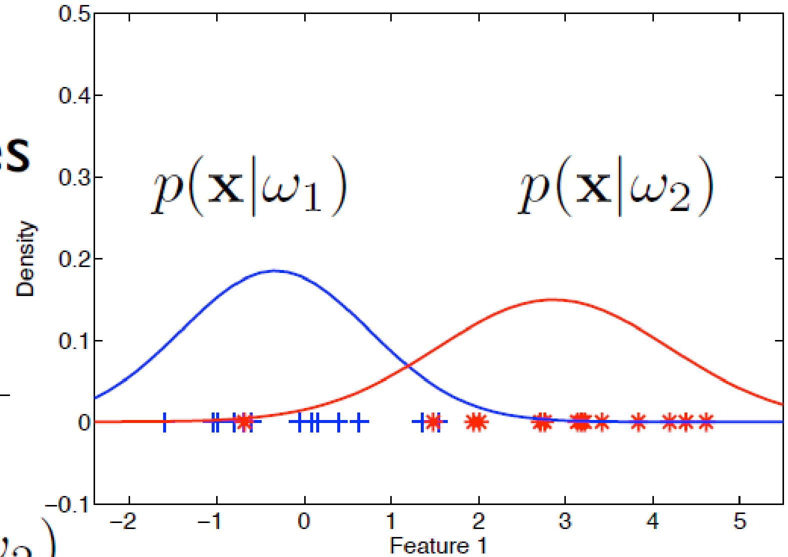
$$P(\boldsymbol{x}, \omega)$$

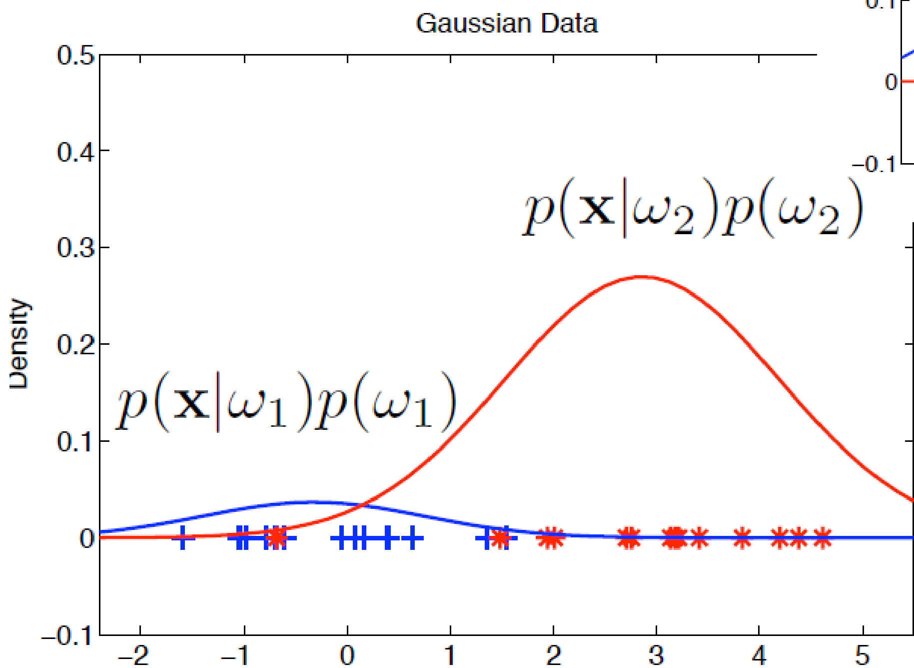
Or

$$P(\omega)\, P(\boldsymbol{x}|\omega)$$

Bayes Classifier

1. Estimate the class conditional probabilities





2. Multiply with the class priors

27

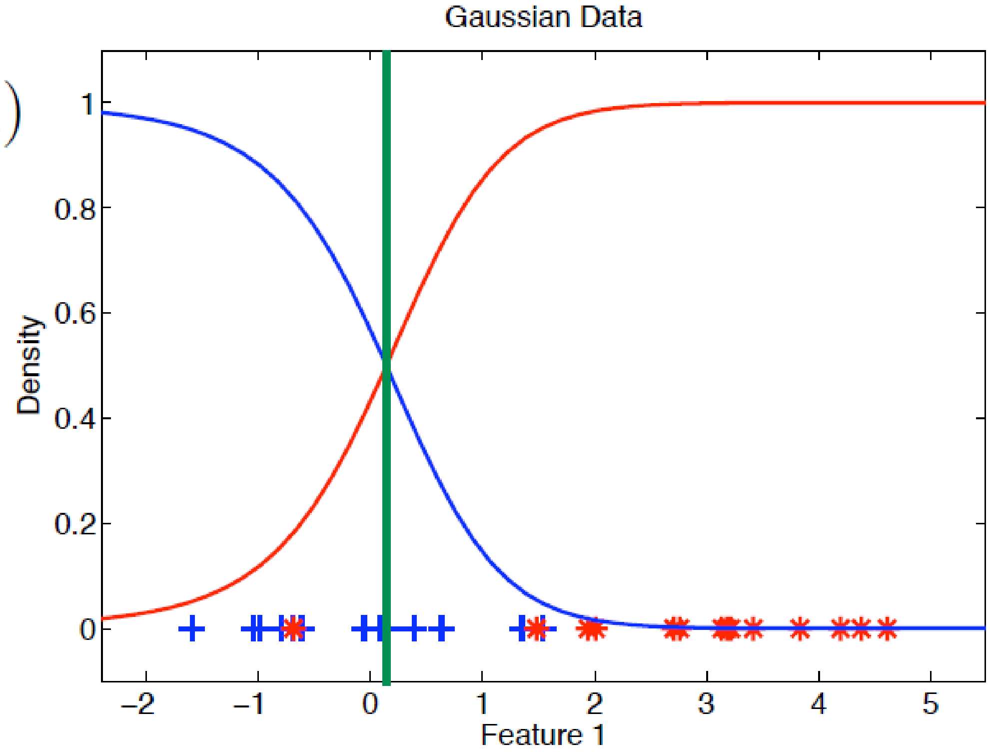
Source: CS4220 TU Delft

TU Delft

Bayes Classifier

3. Compute the class posterior probabilities
4. Assign objects to the class with the highest posterior probability

$p(\omega_1|\mathbf{x})$



$p(\omega_2|\mathbf{x})$

Source: CS4220 TU Delft

Bayes Classifier

- Approaches modeling P($\mathbf{x}$, ω) are called Generative approaches
- The distribution of P($\mathbf{x}$, ω) can be complex and hard to model
- It's impossible to obtain the true distributions in real world

An Alternative

- Distinctive approaches: Model a function that directly map from input **x** to the output label ω :

$$\omega = f(\boldsymbol{x})$$
$$\boldsymbol{x} = (x_1, x_2, x_3 \ldots x_p)^T$$

  – Linear mappings (Fisher, SVM)
  – Non-linear mappings (decision trees, neural networks, deep learning)

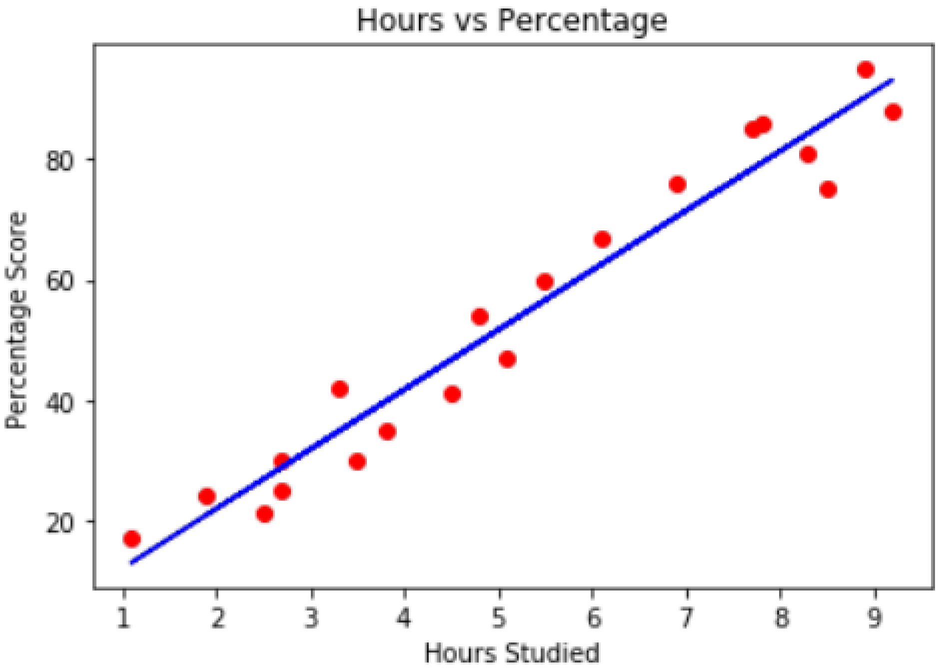
Today’s Agenda

- Machine Learning basics
  - Definition & Scope of machine learning
  - Bayes classifier
  - Linear classifier (Fisher, SVM)

- Deep Learning for 3D urban applications
  - Deep learning intuition
  - Deep neural networks for 3D classification and segmentation of point clouds

Linear Regression Recap

- $y_i = \boldsymbol{w}^T \boldsymbol{x_i} + b$
  - $\boldsymbol{x_i} = (x_1, x_2, x_3 \dots x_p)^T$
  - $\boldsymbol{w} = (w_1, w_2, w_3 \dots w_p)$
  - b is the bias scalar
  - y_i is the output scalar which should be continuous

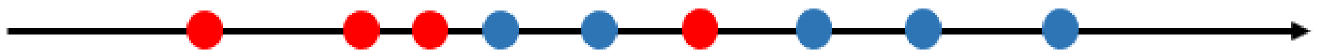


Source: https://stackabuse.com/linear-regression-in-python-with-scikit-learn/

TU Delft

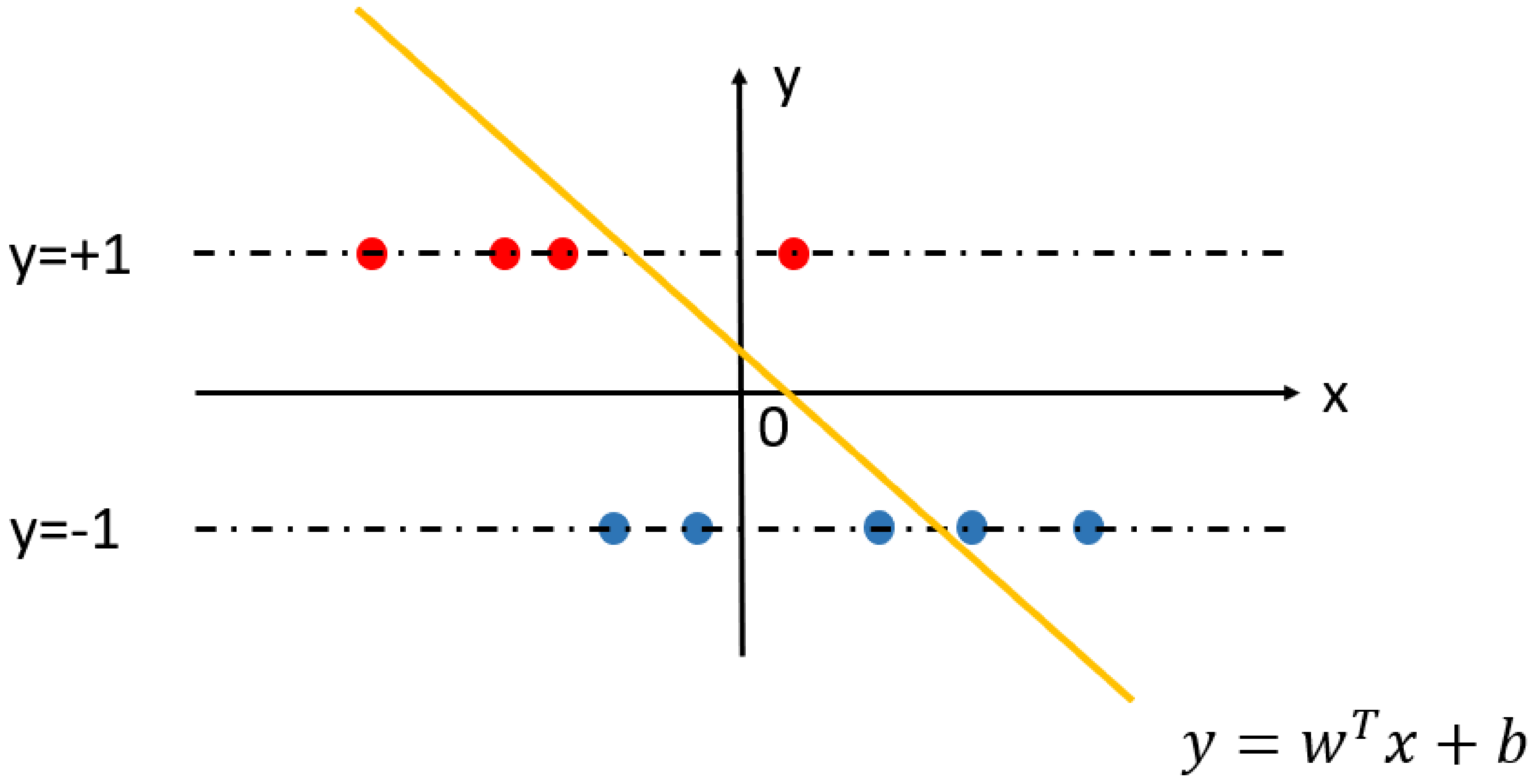
Fisher Classifier

- $y_i = \boldsymbol{w}^T \boldsymbol{x_i} + b$
  - In classification, y is the output labels which are discrete
  - In a 1D feature, 2 class problems, we assume y_i=+1 for positive class, y_i=-1 for negative class

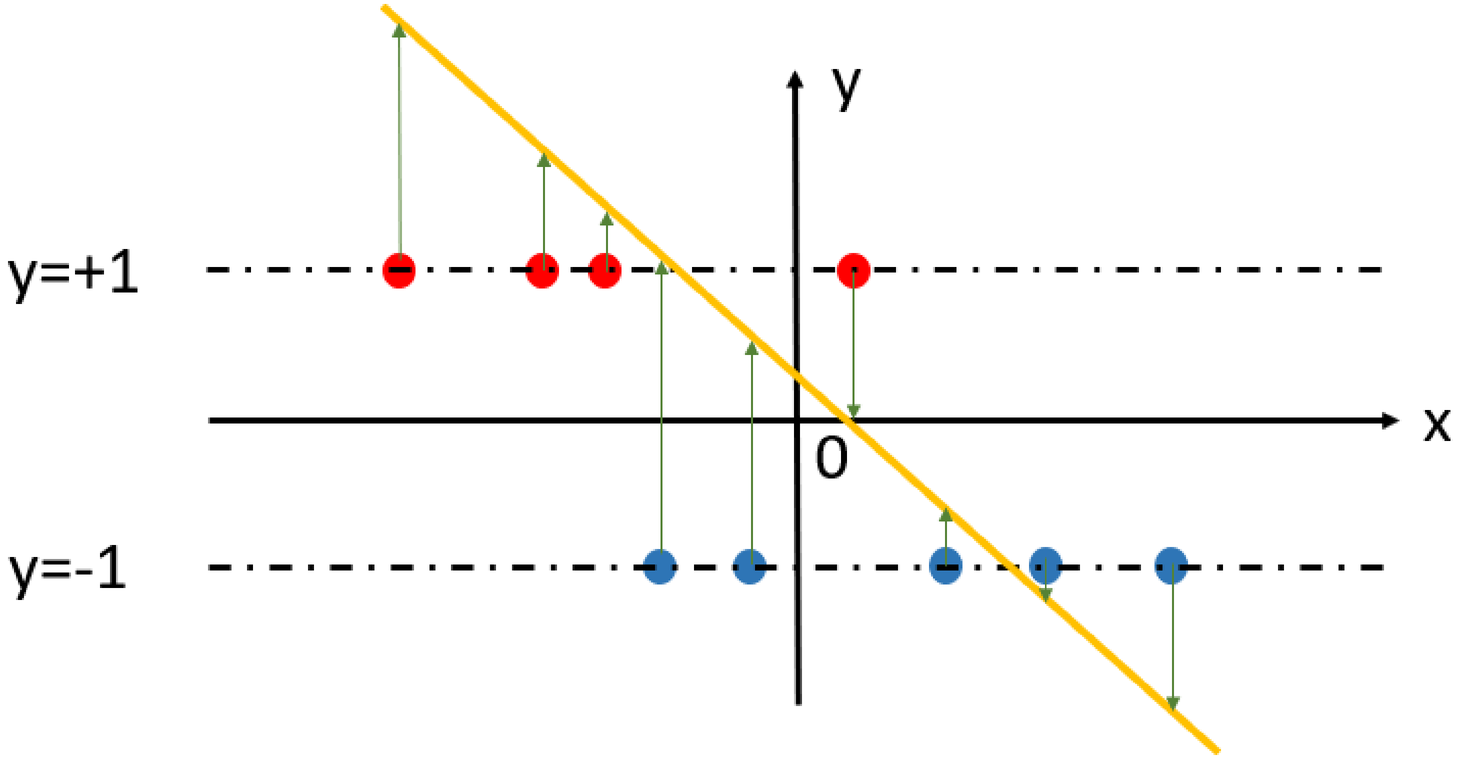


Fisher also represent a feature mapping method, but here we refer “Fisher” only to the standard linear classification

Fisher Classifier



Fisher Classifier



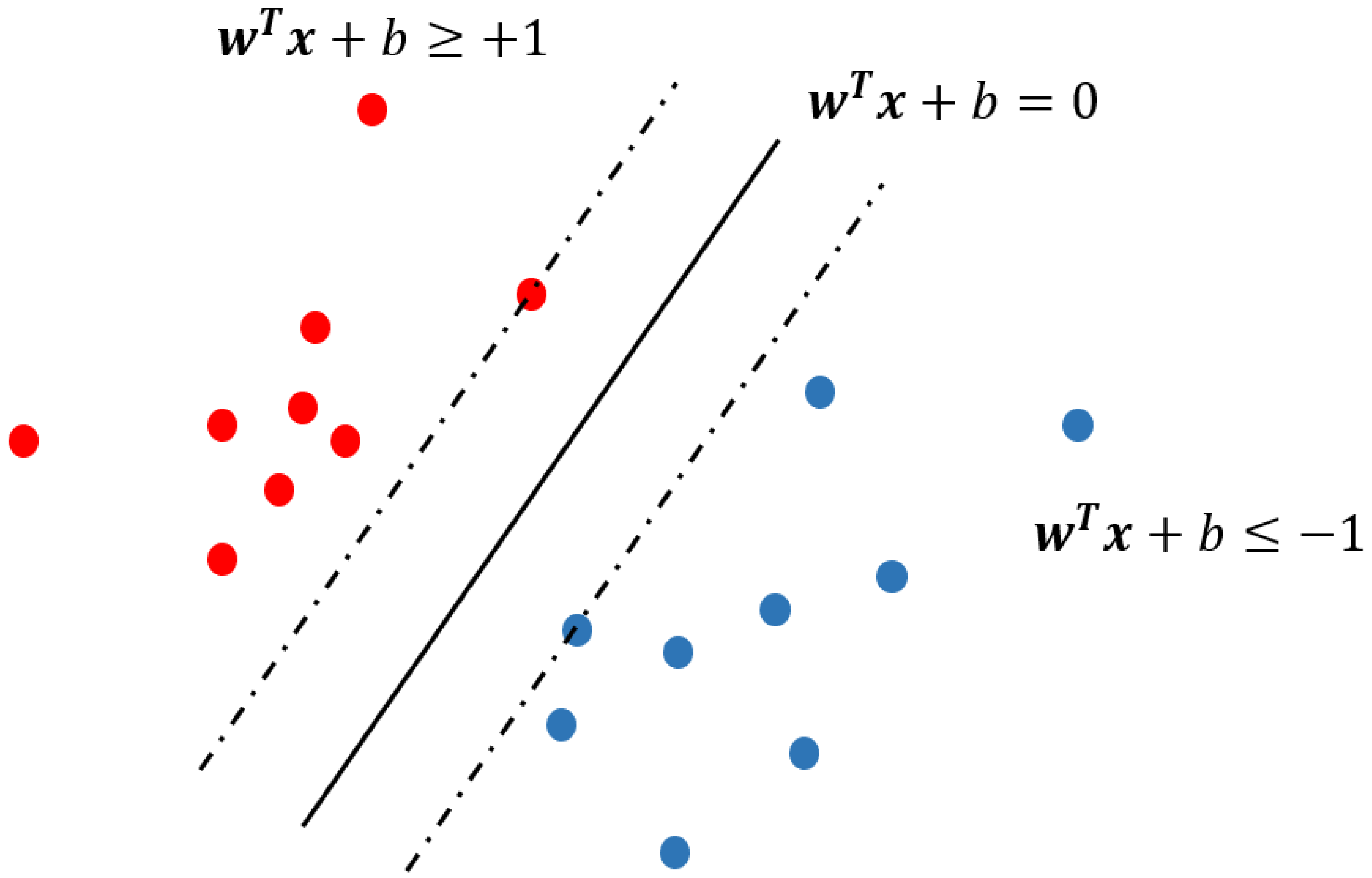
$$y = w^T x + b$$

Support Vector Machine

- Assume the dataset is linearly separable
- Constrain the weights so that the output is always larger than 1 or smaller than -1

$$\begin{cases} \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b \geq +1 & if \quad y_i = +1 \\ \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b \leq -1 & if \quad y_i = -1 \end{cases}$$

Support Vector Machine

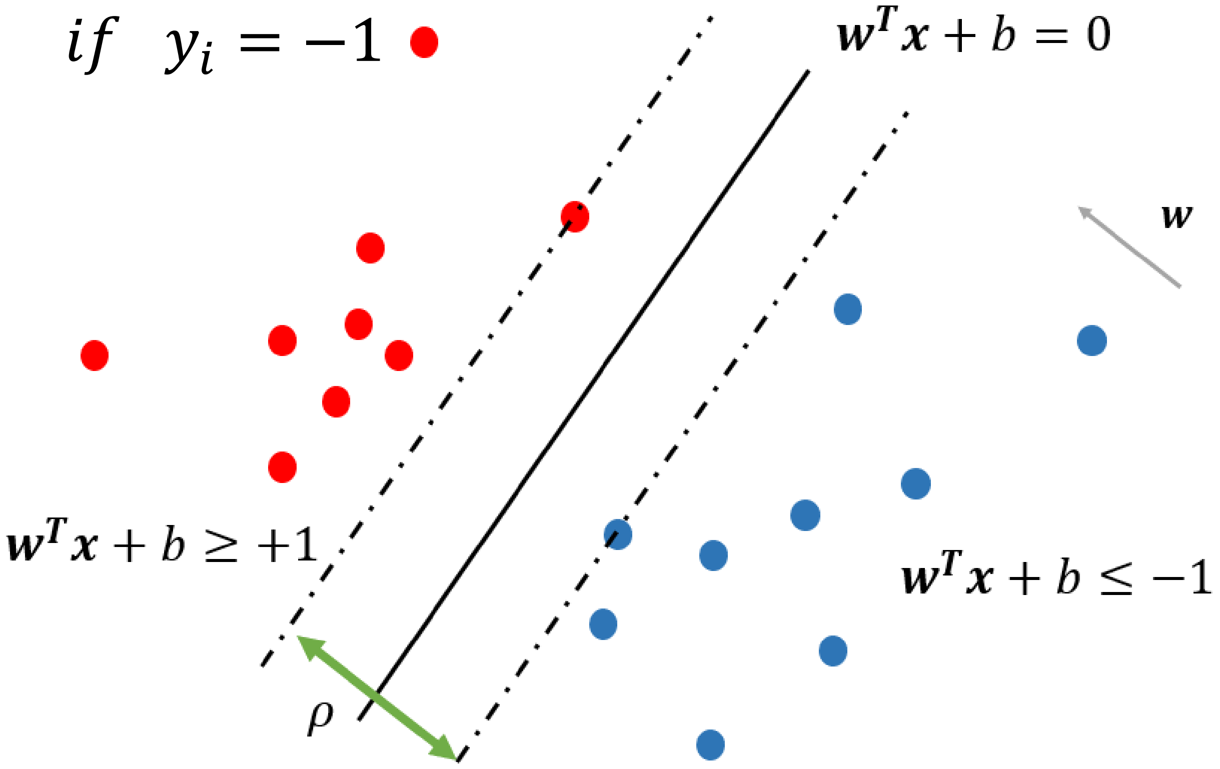


TU Delft

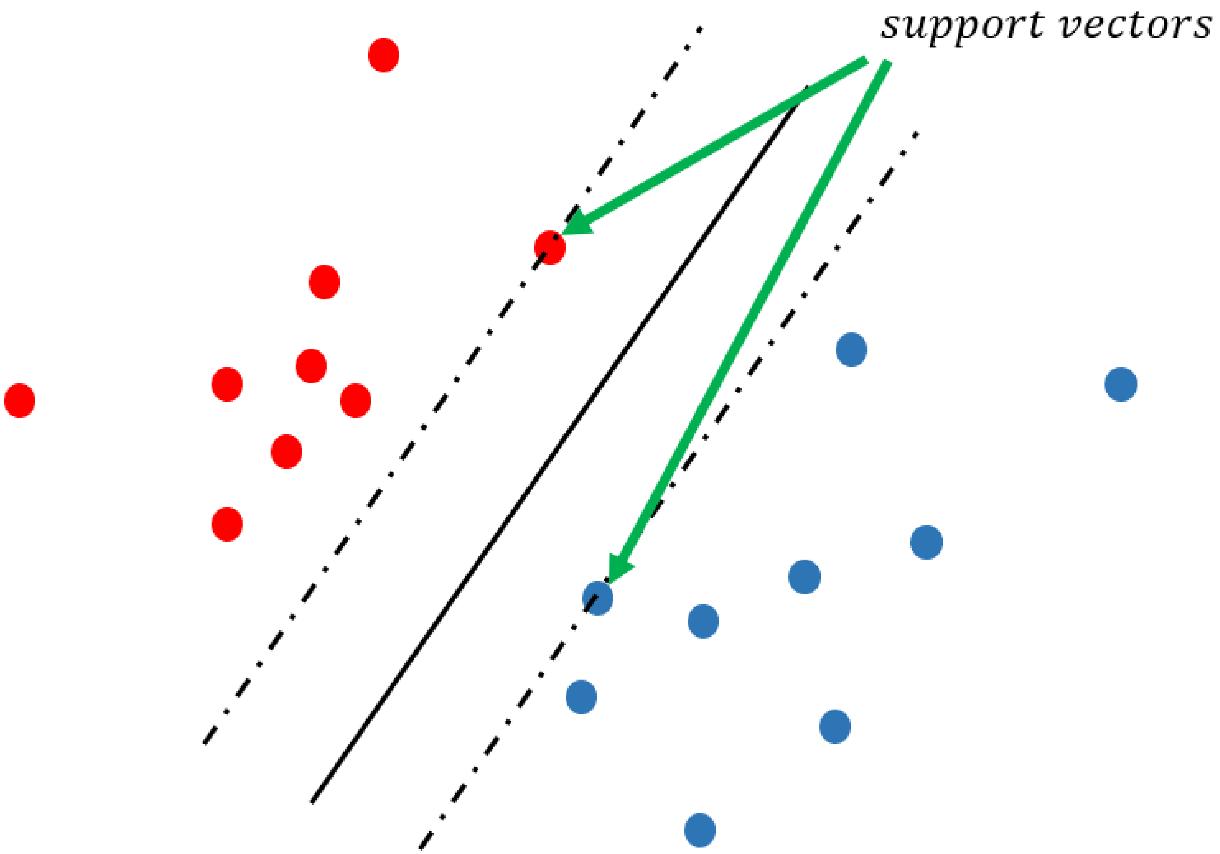
Support Vector Machine

- $max\ \rho$

$$s.t. \begin{cases} \boldsymbol{w}^T \boldsymbol{x_i} + b \geq +1 & if \quad y_i = +1 \\ \boldsymbol{w}^T \boldsymbol{x_i} + b \leq -1 & if \quad y_i = -1 \end{cases}$$



Support Vector Machine



References

- Pattern Recognition and Machine Learning

  https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf
  - Probabilities and Bayes: Section 1.2-1.2.4
  - Linear models: Section 3.1-3.1.1

- Pattern recognition

  https://darmanto.akakom.ac.id/pengenalanpola/Pattern%20Recognition%204th%20Ed.%20(2009).pdf
  - SVM: Section 3.7

Today's Agenda

- Machine Learning basics
  - Definition & Scope of machine learning
  - Bayes classifier
  - Linear classifier (Fisher, SVM)

- Deep Learning for 3D urban applications
  - Deep learning intuition
  - Deep neural networks for 3D classification and segmentation of point clouds

Perceptron

- Consider a 2-class problem, the perceptron is formalized as:

$$y = f(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b),$$

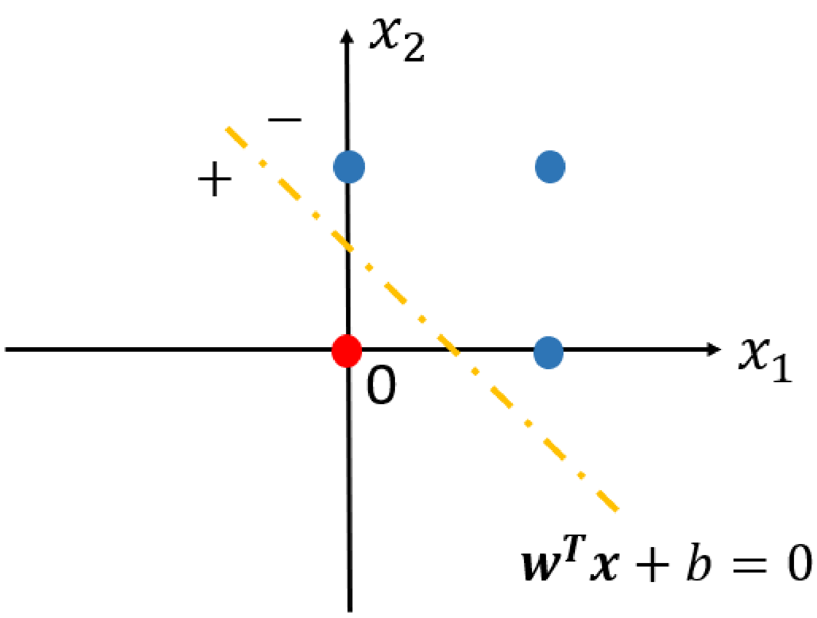
where $f(\cdot)$ defines an activation function:

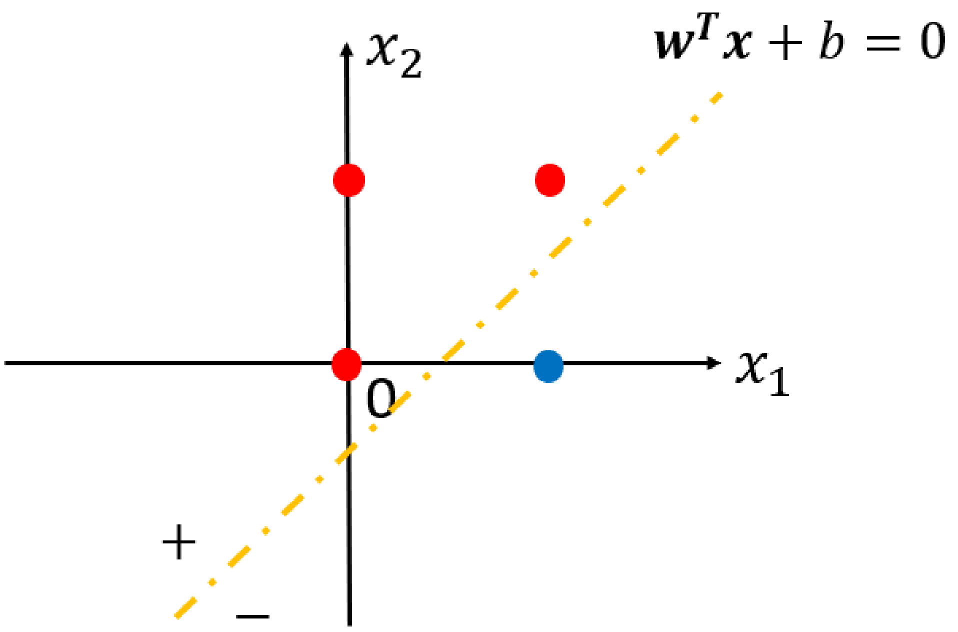
$$f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases}$$

TU Delft

Perceptron

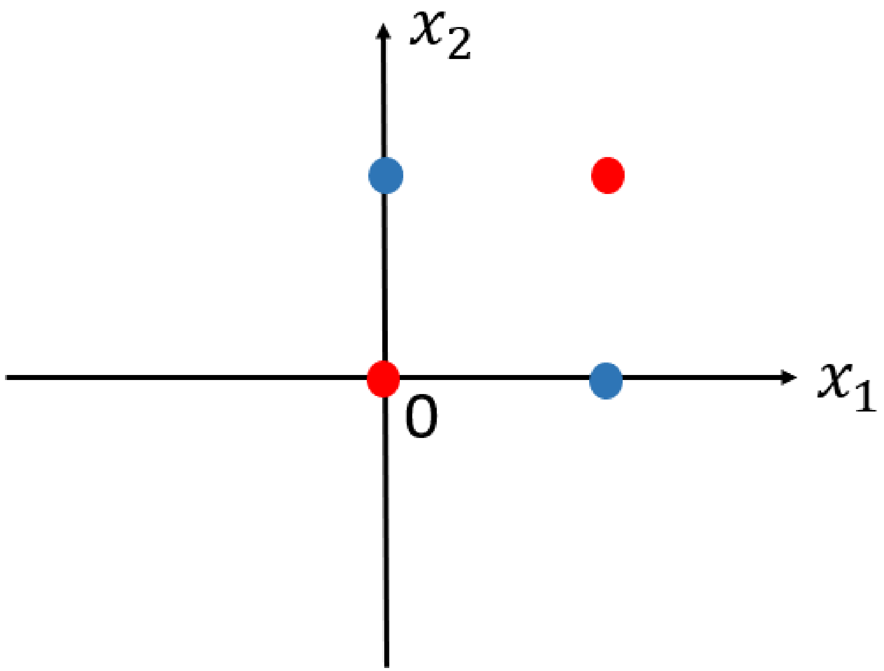
- $f(\cdot)$ defines a linear boundary





Perceptron

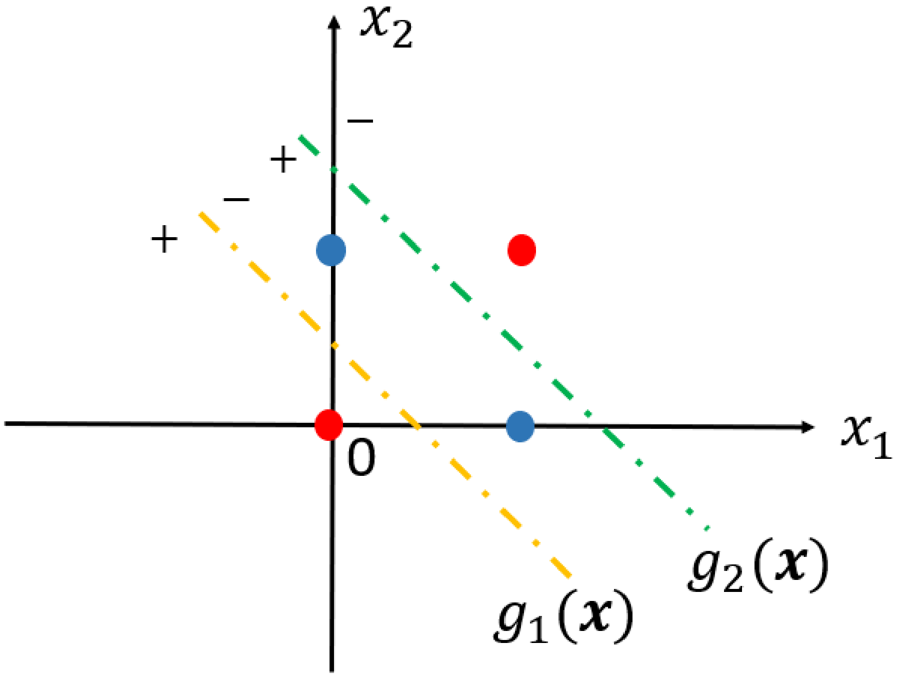
- How to define a boundary for the following case?



x1	x2	y
0	0	+1
0	1	-1
1	0	-1
1	1	+1

TU Delft

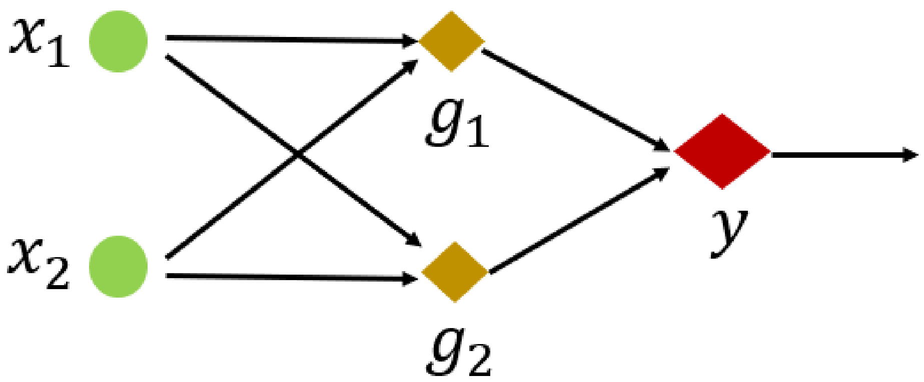
Multi-Layer Perceptron (MLP)



x1	x2	g1	g2	y
0	0	+1	+1	+1
0	1	-1	+1	-1
1	0	-1	+1	-1
1	1	-1	-1	+1

TU Delft

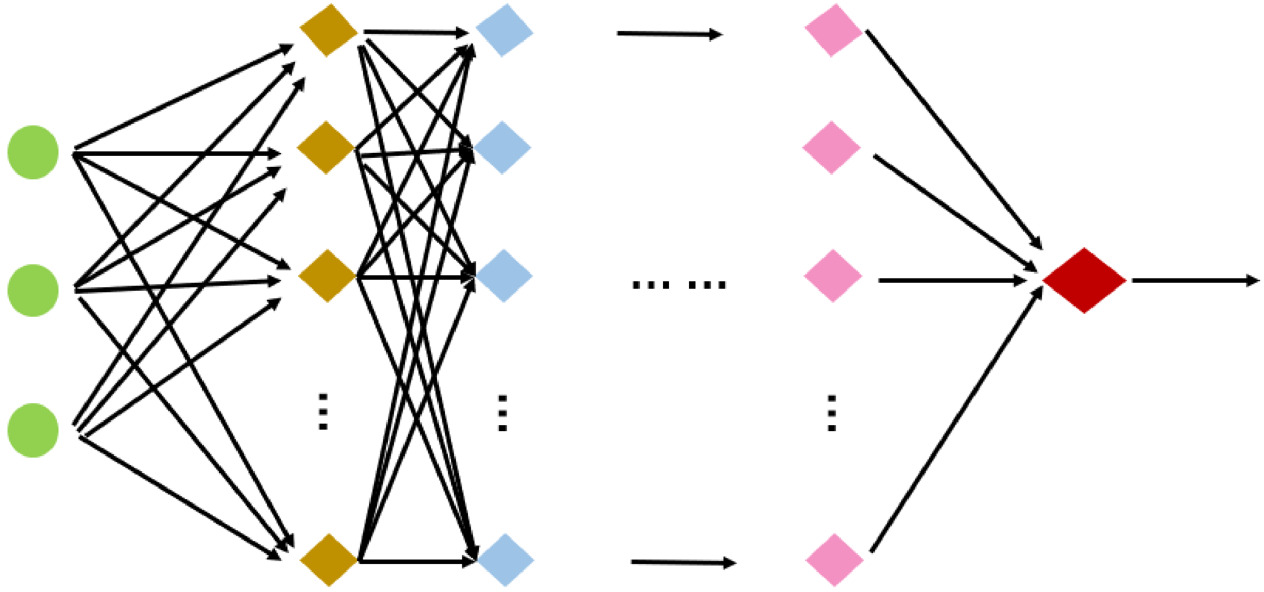
Multi-Layer Perceptron (MLP)



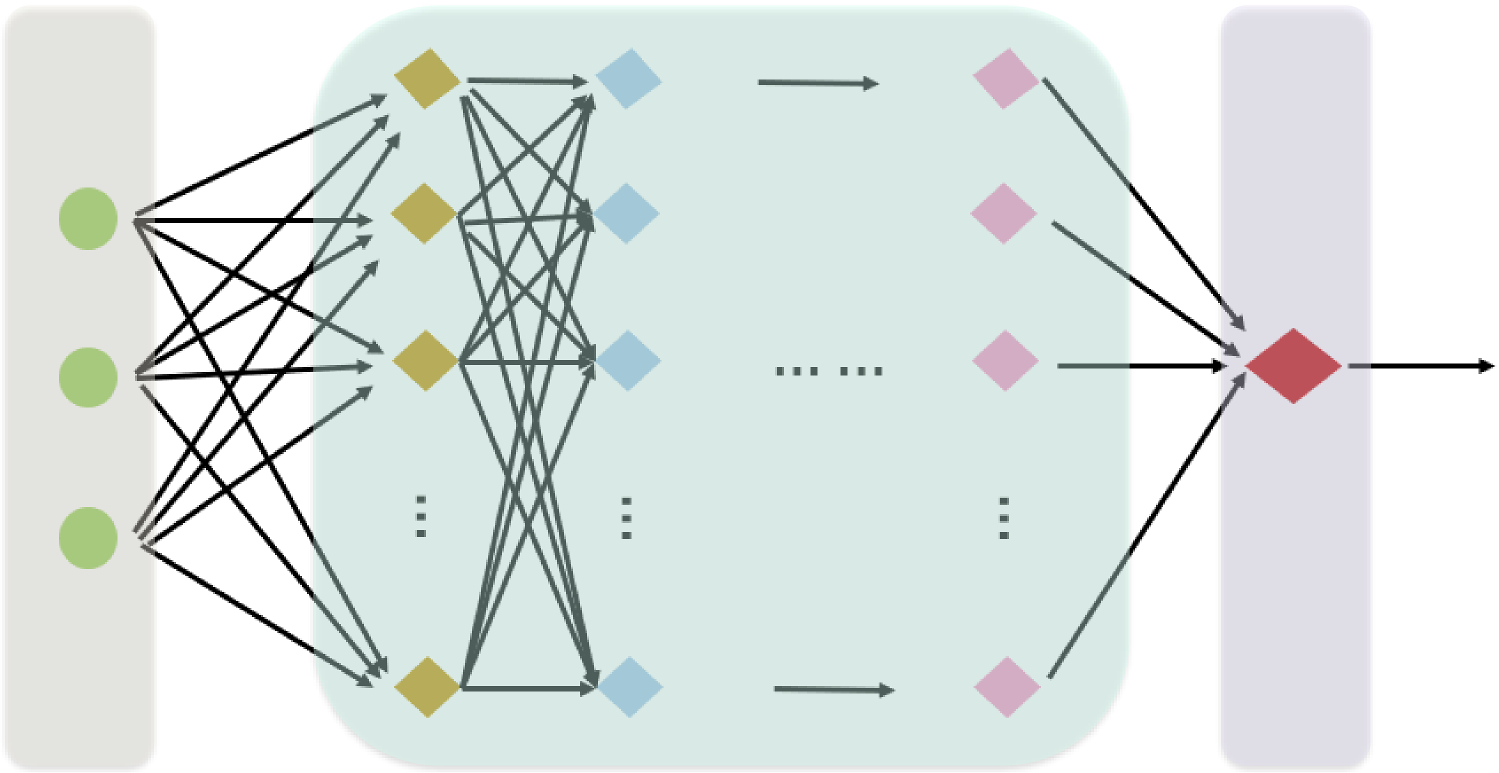
x1	x2	g1	g2	y
0	0	+1	+1	+1
0	1	-1	+1	-1
1	0	-1	+1	-1
1	1	-1	-1	+1

Multi-Layer Perceptron (MLP)

- The more layers and perceptron units we have, the more capable of the machine to model complex non-linear distributions



Deep Learning: Artificial Neural Network



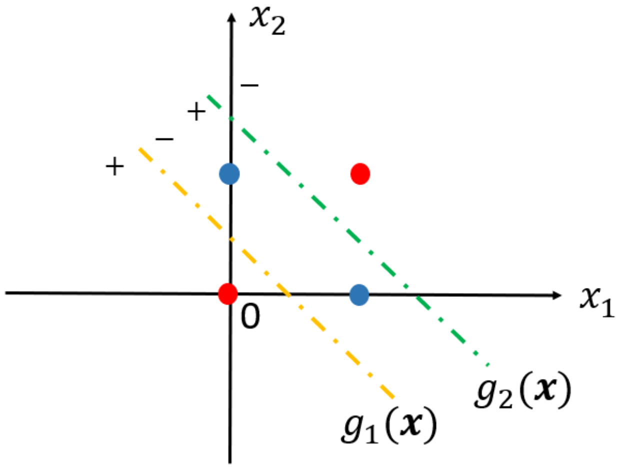
Deep Learning

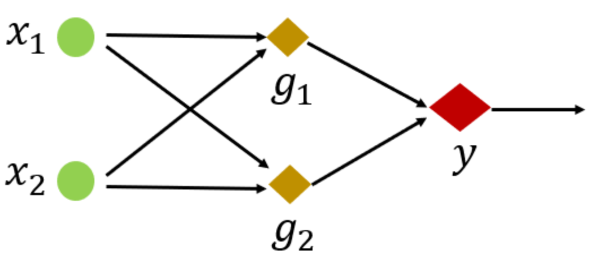
- **Capability**: MLP has the ability of modeling any finite input distributions in theory.

http://www.vision.jhu.edu/teaching/learning/deeplearning18/assets/Hornik-91.pdf

- **Flexibility**: learning to represent the world as a nested hierarchy of concepts, first with simple concepts and then building more complex concepts upon them.

Deep Learning



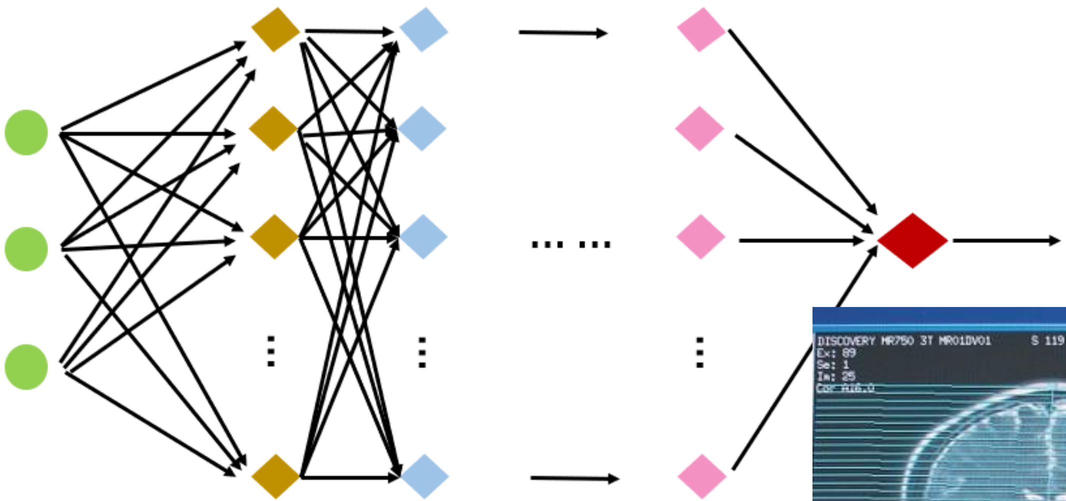


Source: Deep Learning by Ian Goodfellow (https://www.deeplearningbook.org/)

Deep Learning

- ***1940s – 1960s***: Perceptron concept developed
- ***1980s – 1990s***: Connectionism; Back propagation
- ***1990s – 2006s***: Decline of the wave
- ***2006s – Now***: Breakthrough and Prosperity
- ***2015s – Now***: Widely applied in 3D data (point clouds)

Deep Learning vs. Neuro-Science



Source: https://www.nottingham.ac.uk/psychology/research/computational-neuroscience.aspx

References

- Pattern recognition

  https://darmanto.akakom.ac.id/pengenalanpola/Pattern%20Recognition%204th%20Ed.%20(2009).pdf

  - Multi-Layer Perceptron: Section 4.1-4.4

- Deep Learning

  https://www.deeplearningbook.org/

  - DL introduction: Chapter 1

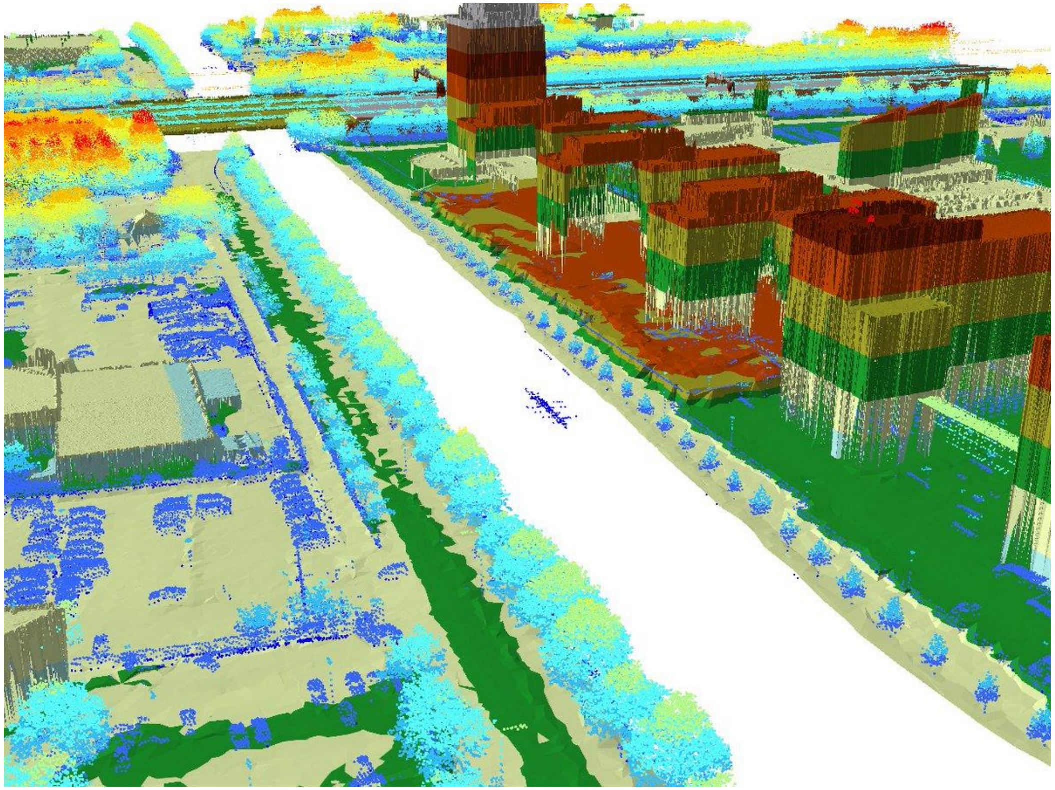
Today's Agenda

- Machine Learning basics
  - Definition & Scope of machine learning
  - Bayes classifier
  - Linear classifier (Fisher, SVM)

- Deep Learning for 3D urban applications
  - Deep learning intuition
  - Deep neural networks for 3D classification and segmentation of point clouds

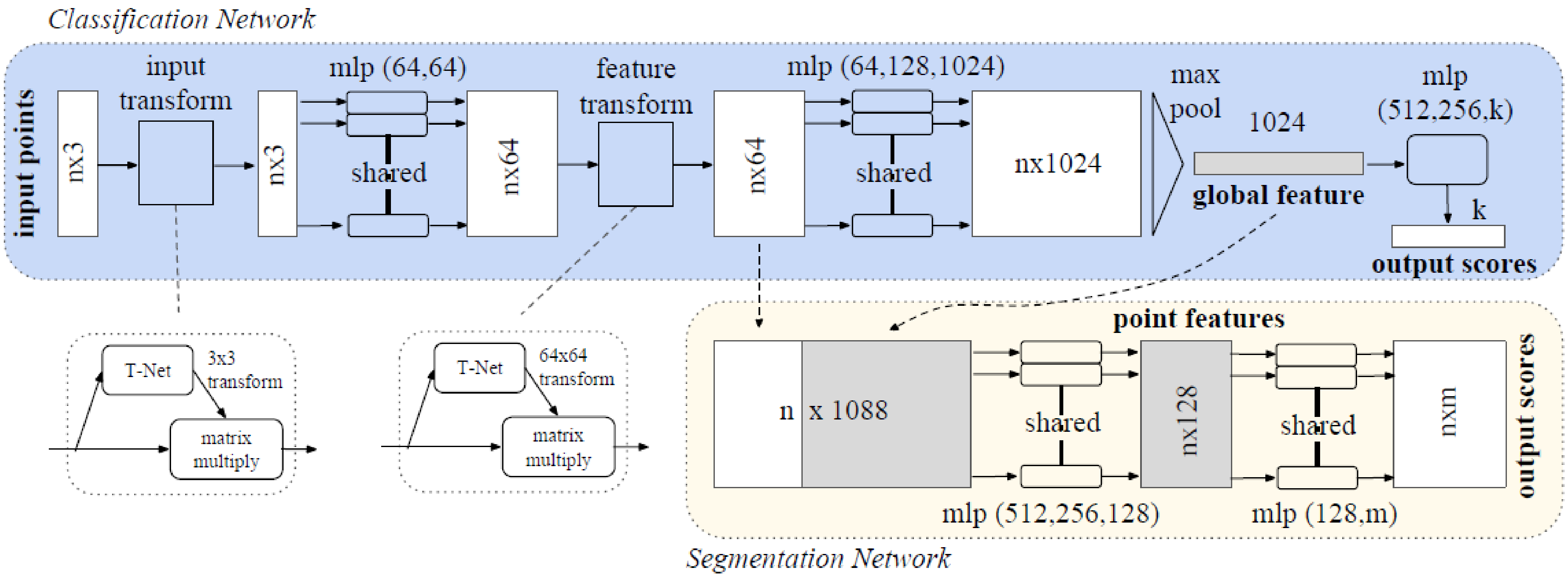
Motivation

- 20% - 30% cost of the AHN project goes to classification
- Manual labelling is the prior choice



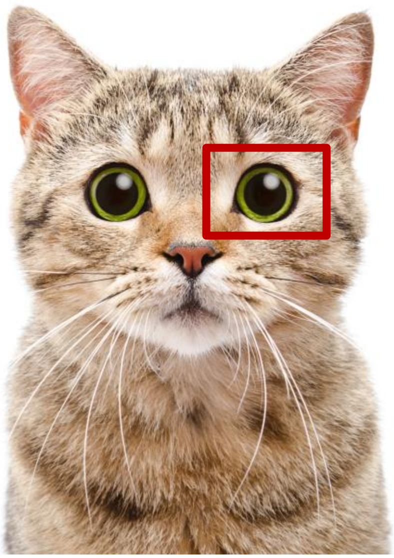
Source: https://twitter.com/hashtag/ahn3?lang=ga

PointNet (Qi et al, 2016)



PointNet (Qi et al, 2016)

- Max Pooling



Source: https://timesofindia.indiatimes.com/life-style/relationships/pets/5-things-that-scare-and-stress-your-cat/articleshow/67586673.cms; https://blog.bricsys.com/point-cloud-to-a-bim-model-modeling-a-church-1-the-outside/

PointNet (Qi et al, 2016)

- The pioneer work to first apply deep learning in 3D point clouds
- Simple, clearly-structured, elegant
- Taking per individual point into computation, lacking consideration for the context information
- Focused on global rather than local features

PointNet++ (Qi et al, 2017)

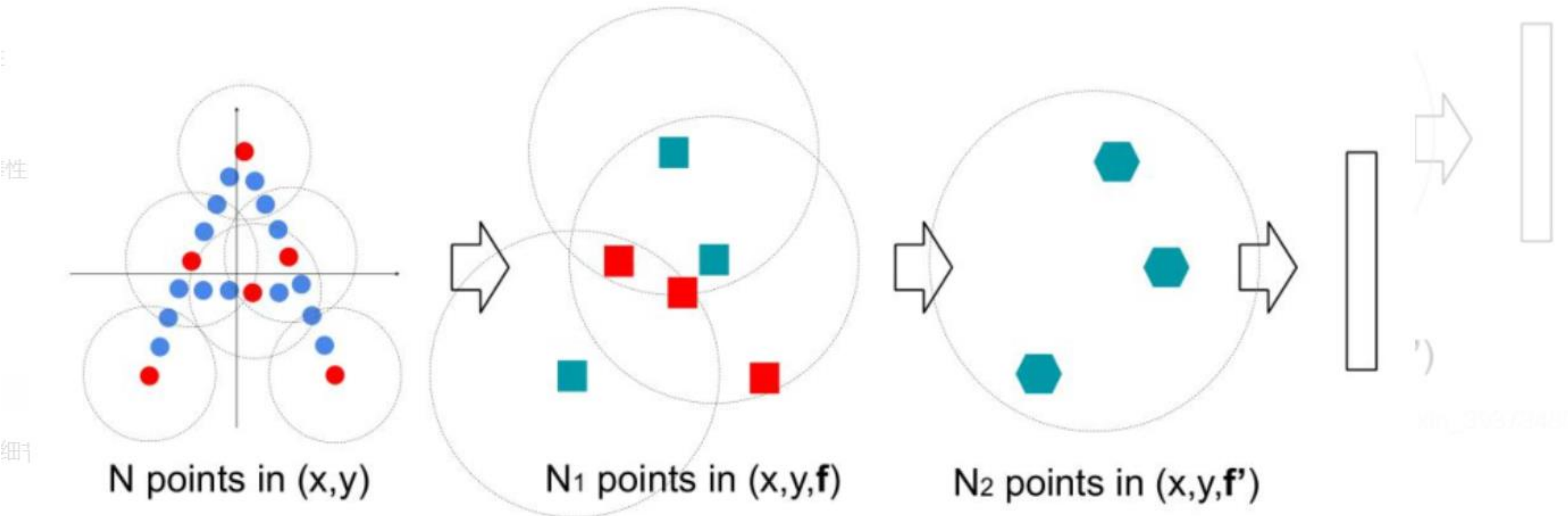
sampling + grouping

pointnet

2nd sampling + grouping

pointnet

3rd sampling + grouping

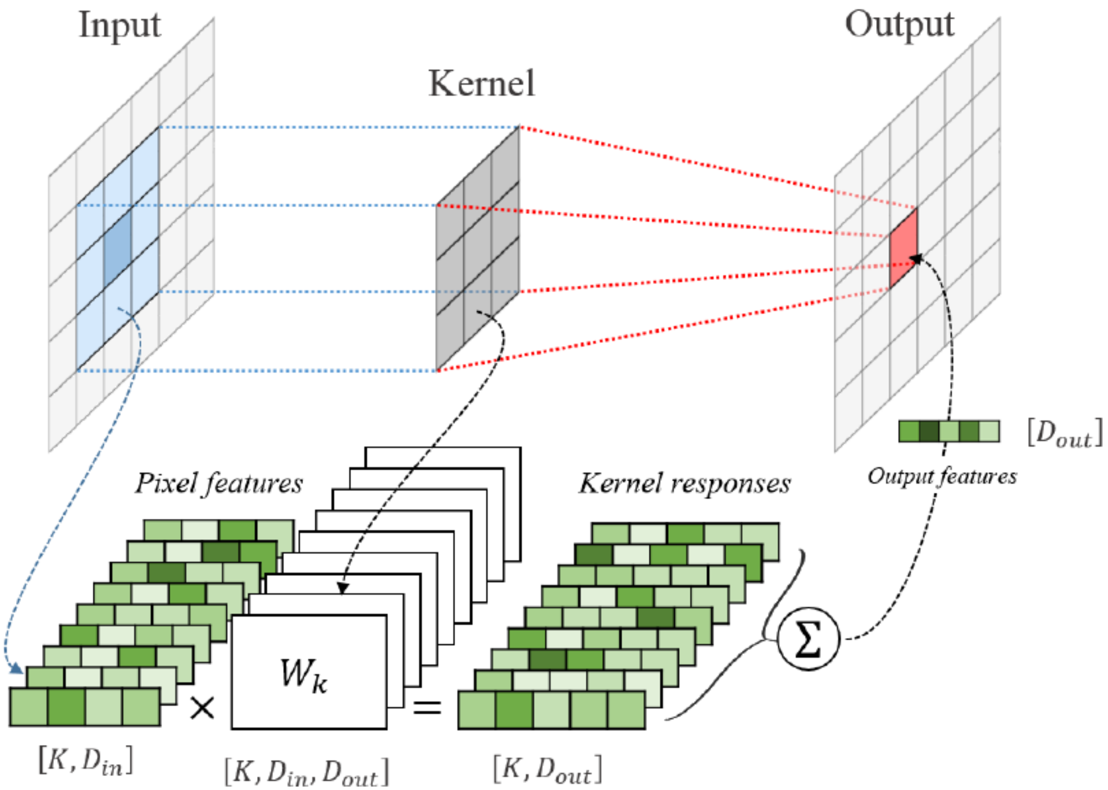


PointNet++ (Qi et al, 2017)

- Involving local features by aggregating features in the neighborhoods
- Making efforts to achieve hierarchical learning
- Still considering per point in its local region independently

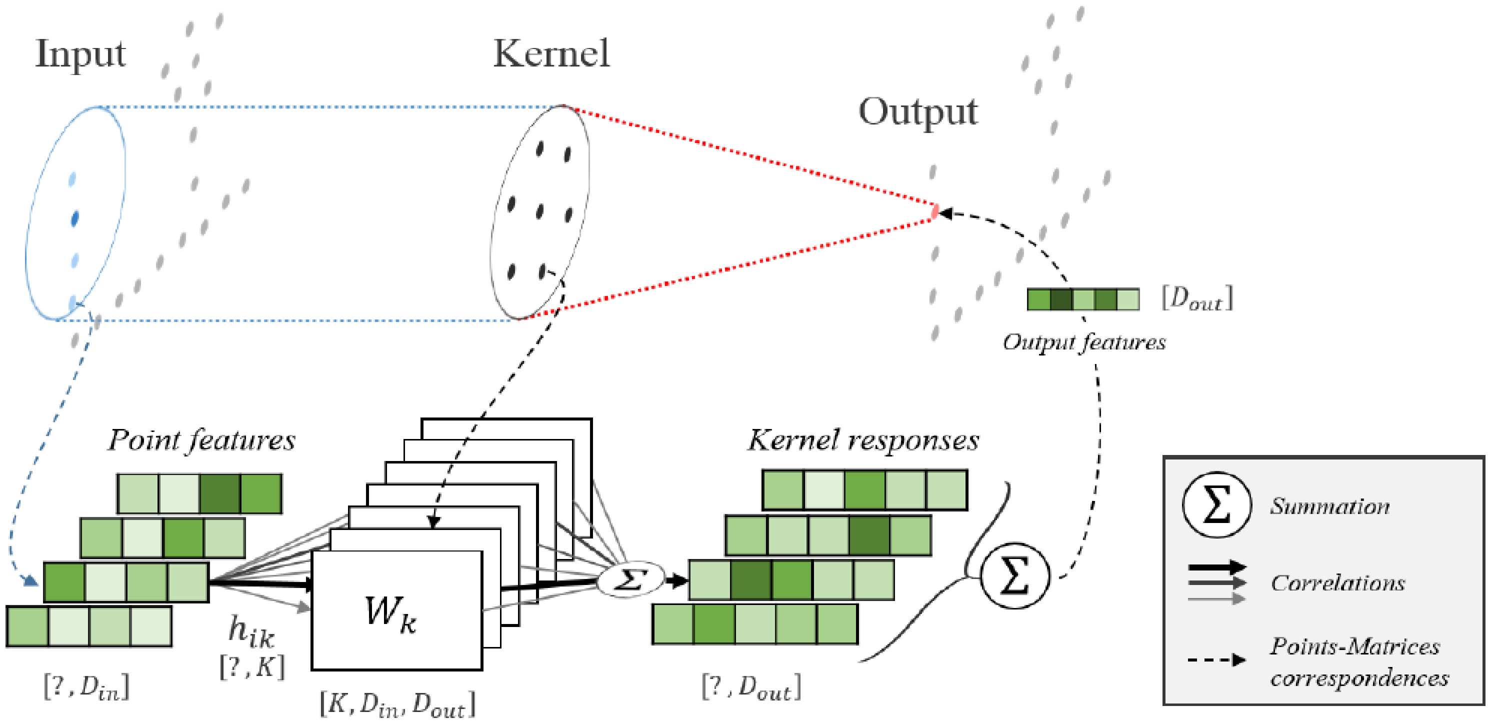
KP-Conv (Thomas et al, 2019)

- Traditional 2D convolution

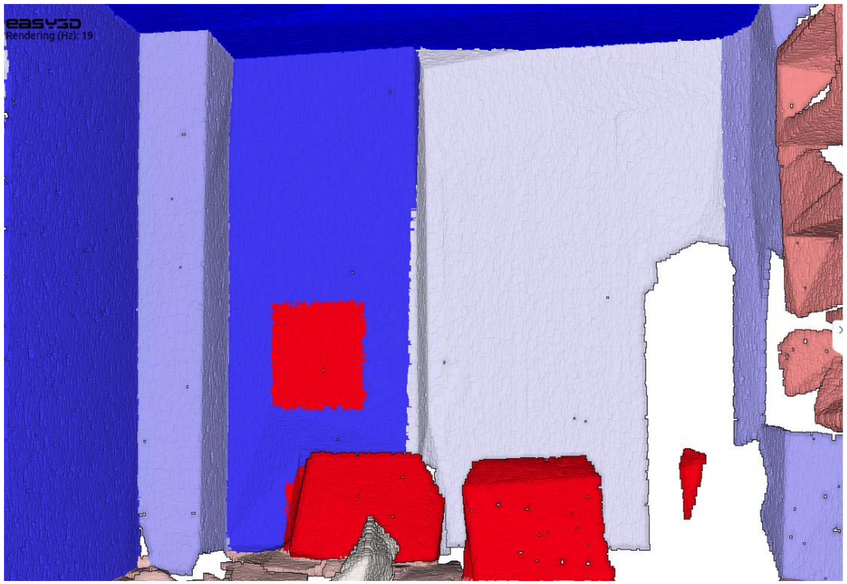


KP-Conv (Thomas et al, 2019)

- 3D convolution using kernel points



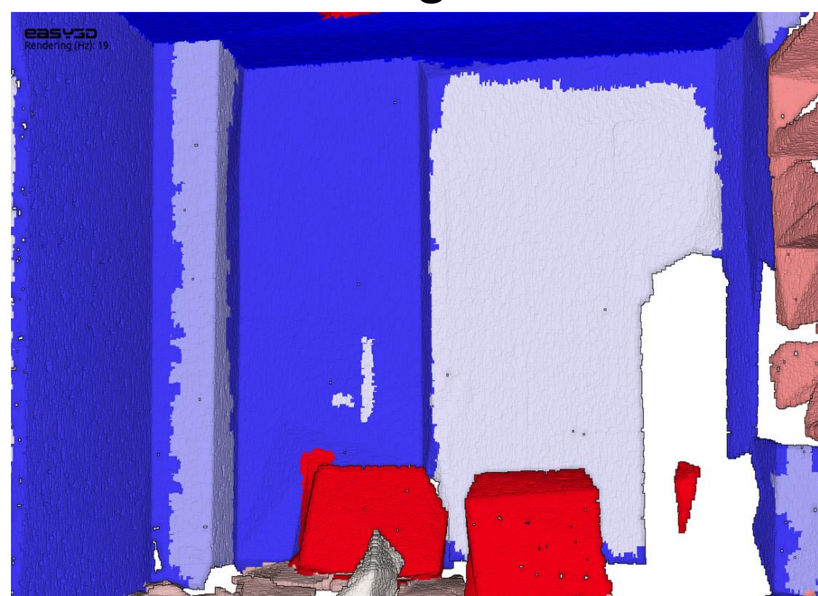
Visualizing the Segmentation (KP-Conv)



Ground truth

- window
- wall
- clutter
- bookcase
- column
- table
- chair

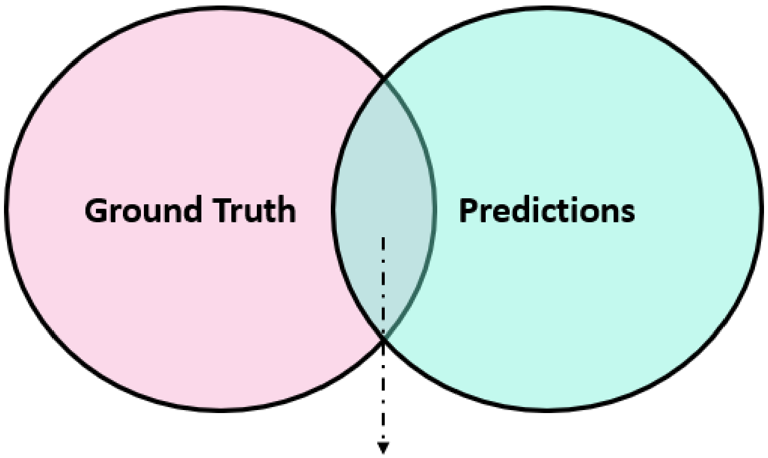
KP-Conv Segmentation



TU Delft

Segmenting Evaluation

- Evaluation is made via mIoU (Mean Intersection over Union):



Methods	Scannet	Sem3D	S3DIS	PL3D
Pointnet [26]	-	-	41.1	-
Pointnet++ [27]	33.9	-	-	-
SnapNet [4]	-	59.1	-	-
SPLATNet [34]	39.3	-	-	-
SegCloud [37]	-	61.3	48.9	-
RF_MSSF [38]	-	62.7	49.8	56.3
Eff3DConv [50]	-	-	51.8	-
TangentConv [36]	43.8	-	52.6	-
MSDVN [30]	-	65.3	54.7	66.9
RSNet [15]	-	-	56.5	-
FCPN [28]	44.7	-	-	-
PointCNN [20]	45.8	-	57.3	-
PCNN [2]	49.8	-	-	-
SPGraph [17]	-	73.2	58.0	-
ParamConv [41]	-	-	58.3	-
SubSparseCNN [9]	**72.5**	-	-	-
KPConv *rigid*	68.6	**74.6**	65.4	72.3
KPConv *deform*	68.4	73.1	**67.1**	**75.9**

Project Resources

- **Pointnet**
  - Tensorflow: https://github.com/charlesq34/pointnet
  - Pytorch: https://github.com/fxia22/pointnet.pytorch

- **Pointnet++**
  - Tensorflow: https://github.com/charlesq34/pointnet2
  - Pytorch: https://github.com/yanx27/Pointnet_Pointnet2_pytorch

- **KP-Conv**
  - Tensorflow: https://github.com/HuguesTHOMAS/KPConv
  - Pytorch: https://github.com/HuguesTHOMAS/KPConv-PyTorch

A Small Challenge (Optional)

- The exercise is about Bayes classifier
- 2-classes, 1D feature space
- Questions together with answers will be shared

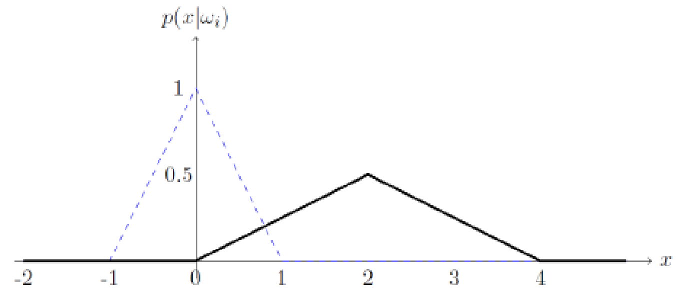


Figure 1.1: The class-conditional probabilities of two classes $p(x|\omega_1)$ (dashed blue line) and $p(x|\omega_2)$ (solid black line) in a 1-dimensional feature space.

1. Two class conditional probability density functions are given in the figure above. The first class ω_1 is represented by a dashed blue line and the second class ω_1 is represented with a solid black line. Two classes have equal prior probability:

$$P(\omega_1) = P(\omega_2) = 0.5$$

(a) Use the Bayes' rule to derive the class posterior probabilities of the following objects:

- x = -0.5;
- x = +0.5;
- x = 3;

To which class are the objects therefore assigned?

(b) What is the decision boundary of the Bayes classifier?

2. Revisit the question 1, assume the prior probabilities of two classes have changed:

$$P(\omega_1) = \frac{1}{3}$$

$$P(\omega_2) = \frac{2}{3}$$

Again, what is the decision boundary of the Bayes classifier?

TU Delft

Thank you! Questions?

TUDelft

Source: https://www.searchenginejournal.com/machine-learning-seo/208415/