CS4347

Sound and Music Computing

L2b: Machine Learning for SMC

Wang Ye

www.comp.nus.edu.sg/~wangye

wangye@comp.nus.edu.sg

Office: AS6-04-08

Topics to Cover (*selective approach*)

Part A: The Core

- Introduction
- Review of DFT, Audio Representation, and Machine Learning
- Music Representation, Analysis and Transcription
- Automatic Music Transcription (AMT)
- Automatic Speech Recognition (ASR)
- Generative Models for Text-to-Speech (TTS) & Singing Voice Synthesis (SVS)

Midterm break

Part B: The Breadth

- Singing voice processing
- Music production audio effects
- Automatic Music Generation
- Synthesis of sound & music – a DSP approach
- Project presentations/demo

Some Big Ideas

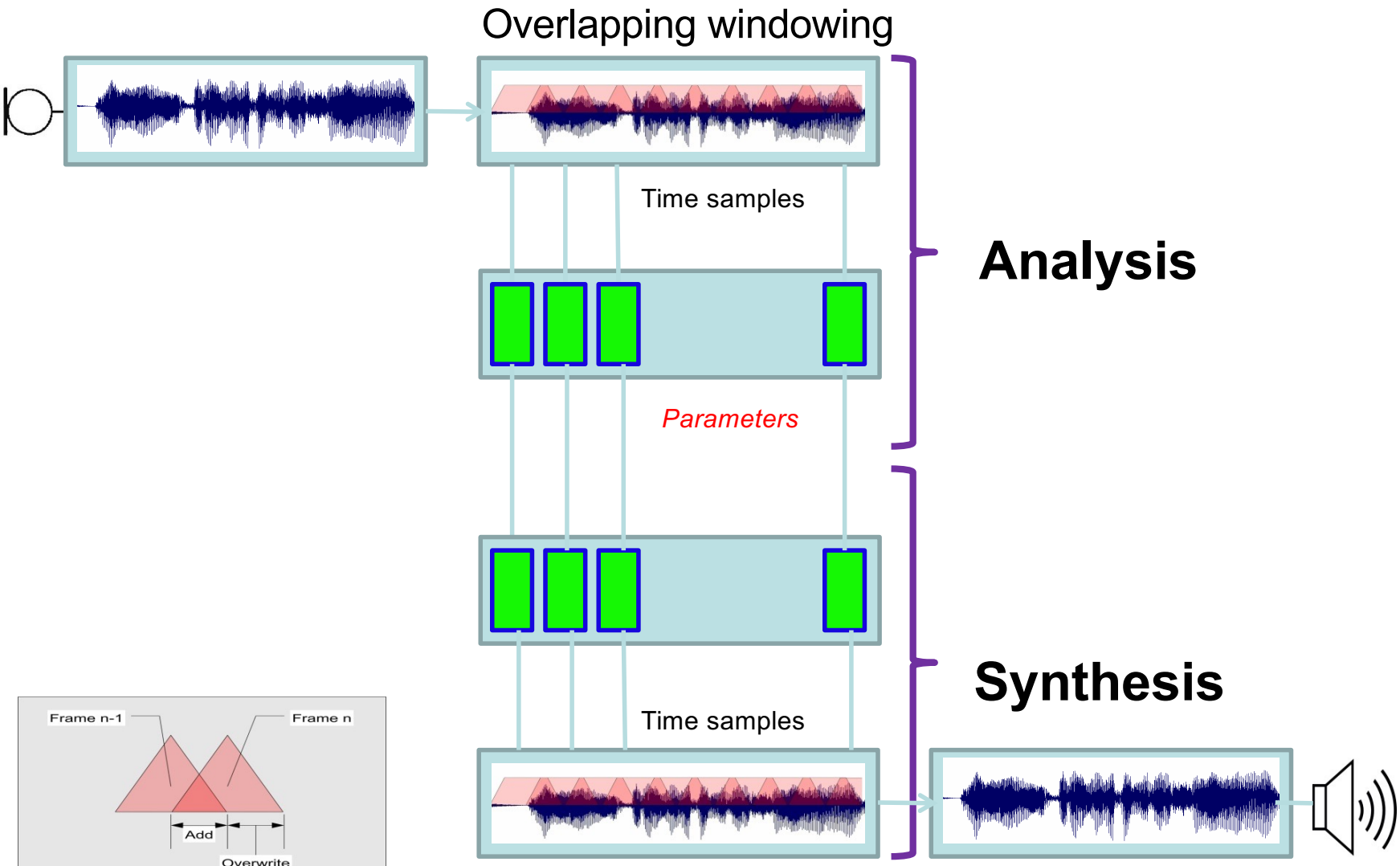
Simplicity plays a key role in science, including computer science.

One simple yet fundamental principle is to decompose a complex problem to something much simpler.

For example, DFT is to decompose an unknown signal (can be very complex) into sum of sinusoids of different frequencies.

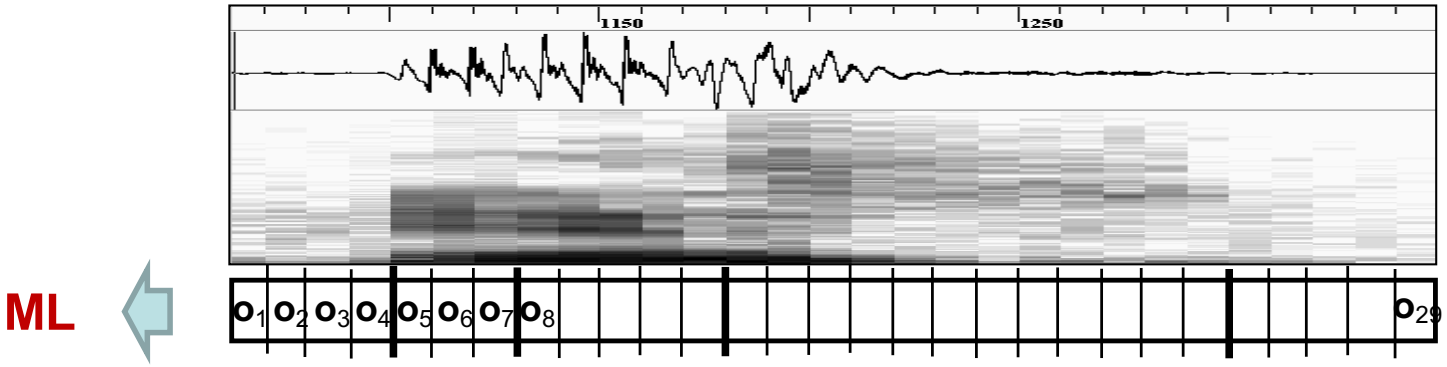
In machine learning, lines and planes are used to separate data into different classes (classification).

Audio Analysis and Synthesis Framework
from a DSP Perspective



Audio Signal Analysis Framework

to answer more interesting questions that DSP alone cannot answer



1) Is it speech, music or something else (audio classification)?

2) Who spoke at the meeting and when (diarization)?

3) What was spoken (ASR)?

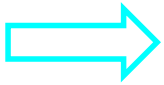
4) What instruments are in this piece of music (instrument recognition)?

5) How to convert this musical signal to a MIDI file (AMT)?

6) Who is the best singer/presenter in class based on the recordings?

⋮

Topics Today



Part A: Machine Learning Fundamentals

- A challenging task due to diverse student background!
- Overview + big ideas

Part B: Practical issues

Part C: Classification libraries

Watch videos of 3 big ideas in AI/ML (if you don't know them):

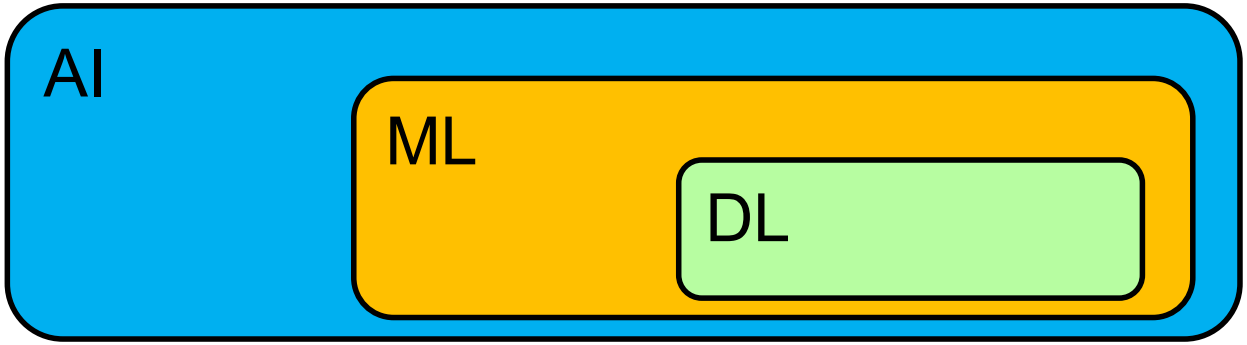
- Perceptron (Frank Rosenblatt, 1958)
- Gradient descent
- Backpropagation

AI vs. ML vs. DL

Artificial Intelligence (AI): A vague term meaning many thing (no definition exists).

Machine Learning (ML): Techniques that learn to solve problems from data.

Deep Learning (DL): A branch of ML using deep neural networks.

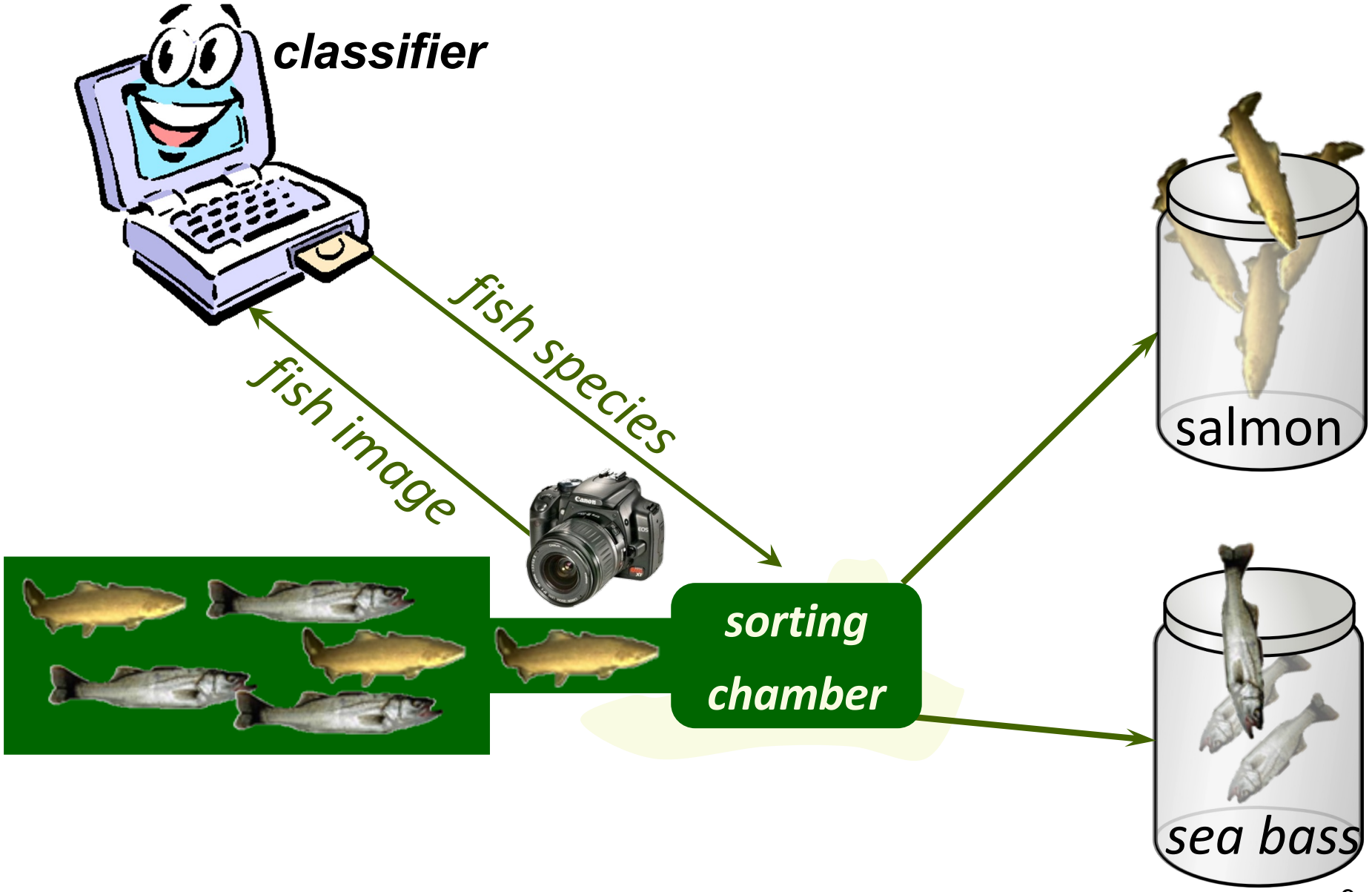


Case Studies

Let's have a look of a few examples

- Computer vision,
- Natural language processing,
- Music information retrieval,
- and gesture recognition (e.g., dance moves).

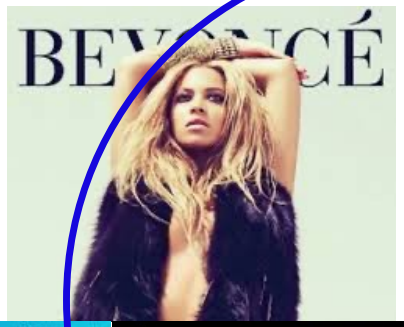
Toy Application: fish sorting

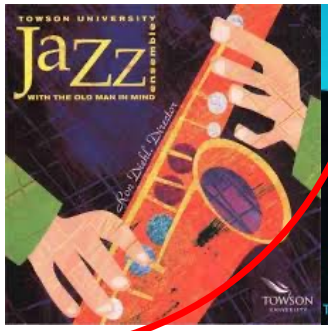


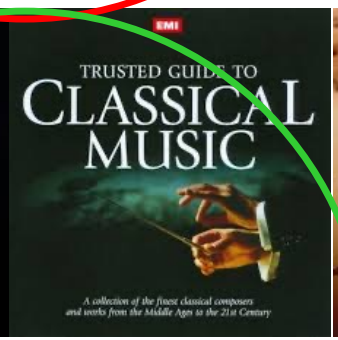
Examples of Text Categorization

- LABELS=BINARY
  - “spam” / “not spam”
- LABELS=TOPICS
  - “finance” / “sports” / “asia”
- LABELS=OPINION
  - “like” / “hate” / “neutral”
- LABELS=AUTHOR
  - “Shakespeare” / “Marlowe” / “Ben Jonson”
  - The Federalist papers

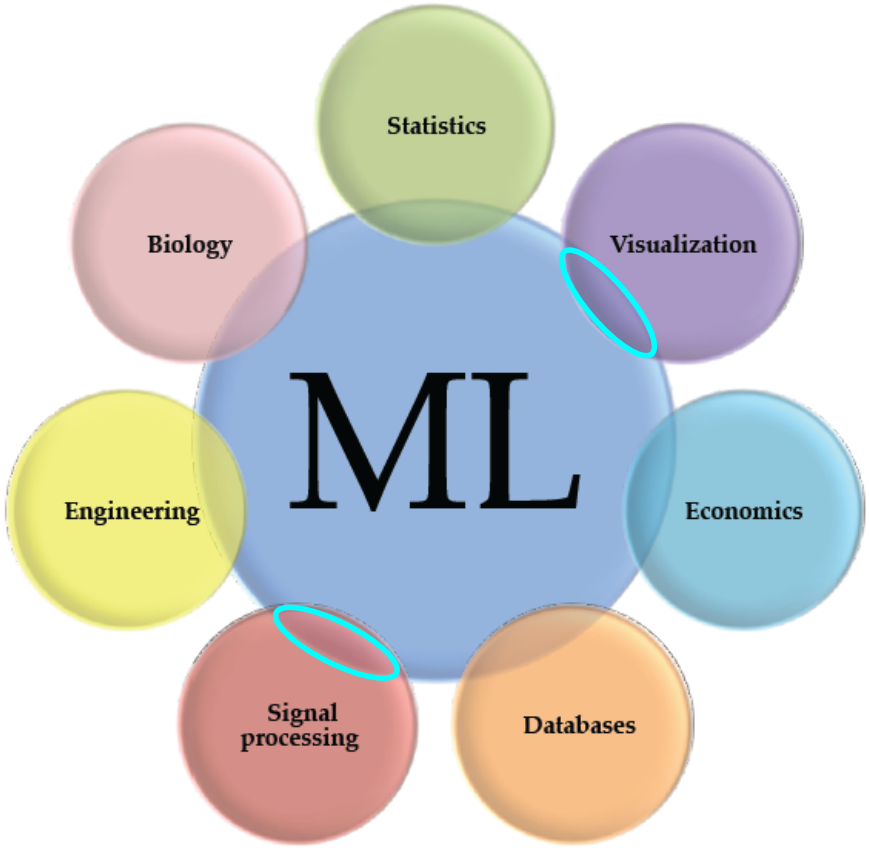
Music genre classification







Applications of ML Problems



DL

(Green AI)

Classic ML

An excursion from classic ML to DL for SMC!

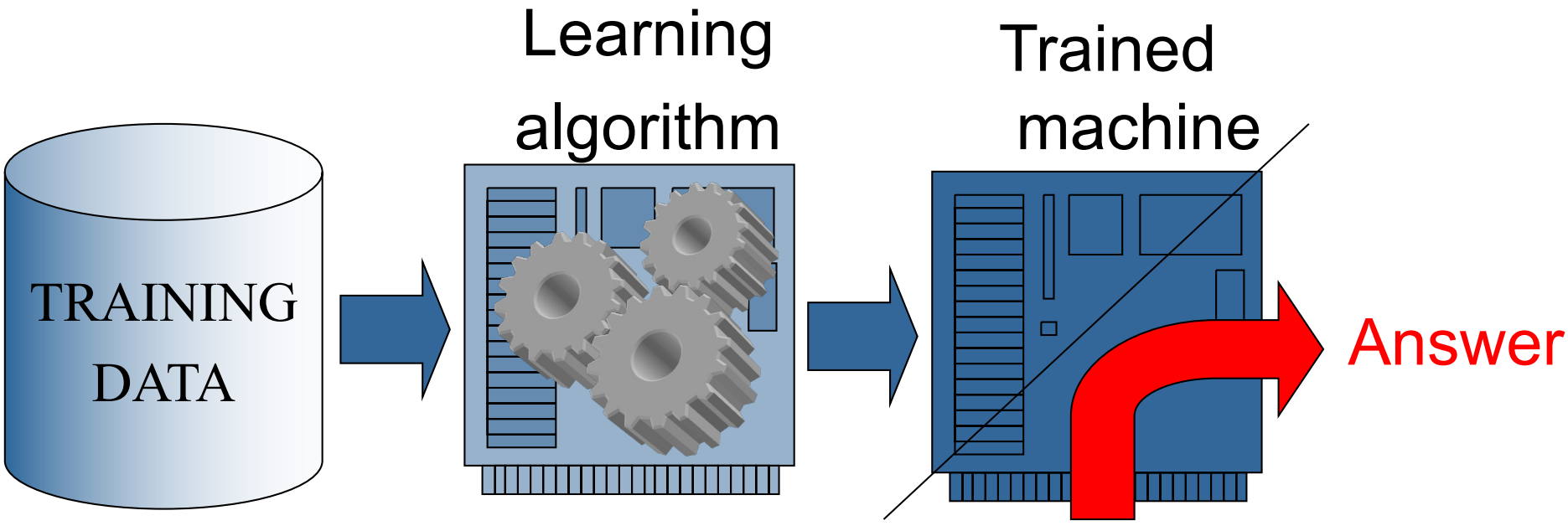
The Essence of Learning Problems

1. A pattern exists

2. Difficult to pin down formally/analytically/mathematically

3. We have data for it (e.g., in our memory)

A Machine Learning Framework
(general)



1. How to learn from experience/data?
2. How to make prediction?

Problem Formalism

Formalization

- Input: $\boldsymbol{x}$ *(music data)*
- Output: $\boldsymbol{y}$ *(genre label)*
- Target Function: $f : \mathcal{X} \rightarrow \mathcal{Y}$ *(model)*
- Data: $\mathcal{D} = (\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)$ *(annotated training data)*

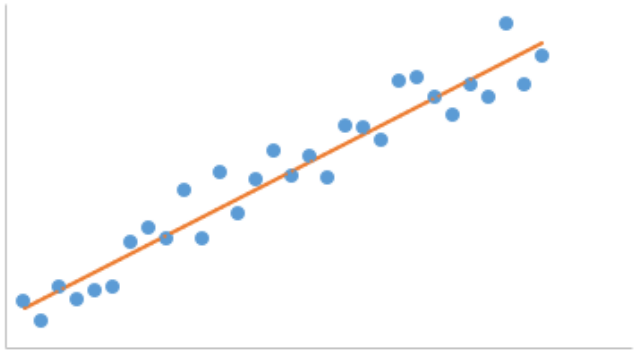
$\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{D}$ are given.

The target f is unknown (to be learned).

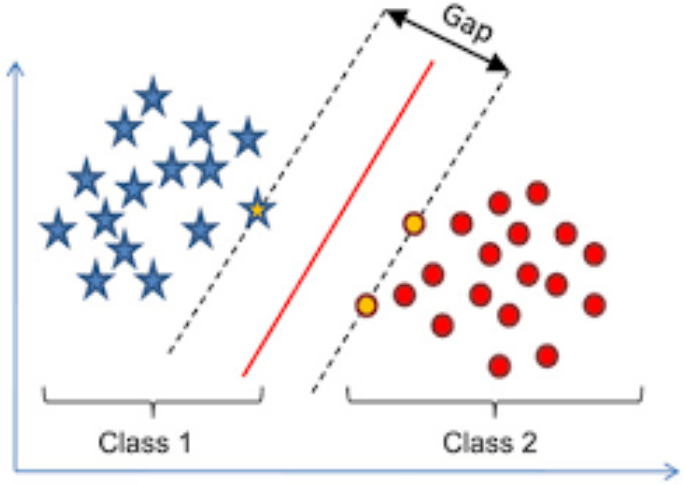
We learn the function f from the data $\mathcal{D}$.

Terminology

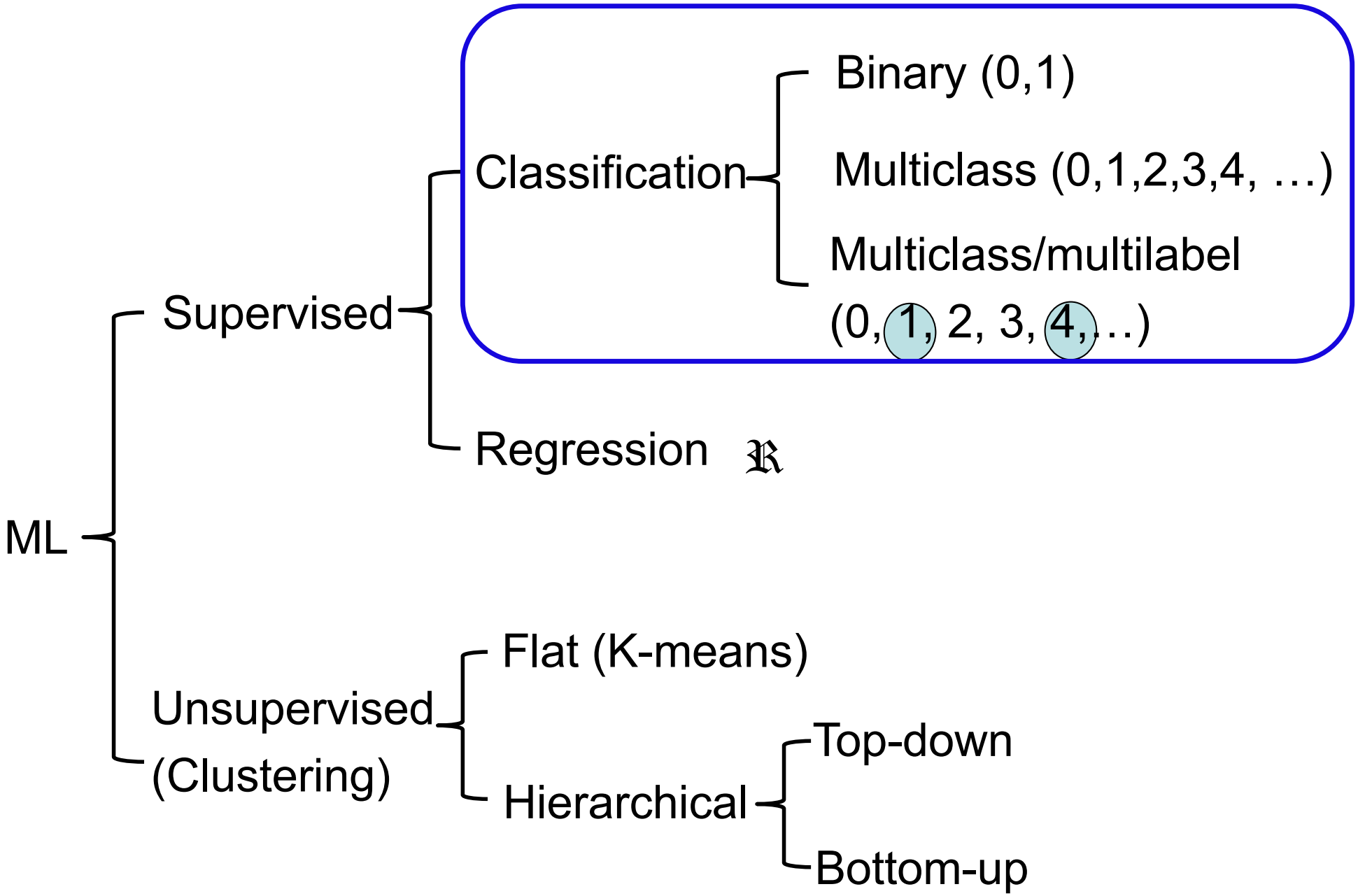
Statistical modeling: In simple terms, statistical modeling is a simplified, mathematically-formalized way to approximate reality (i.e., what generates your data) and optionally to make predictions from this approximation. The statistical model is the mathematical equation that is used.



Regression (curve fitting)

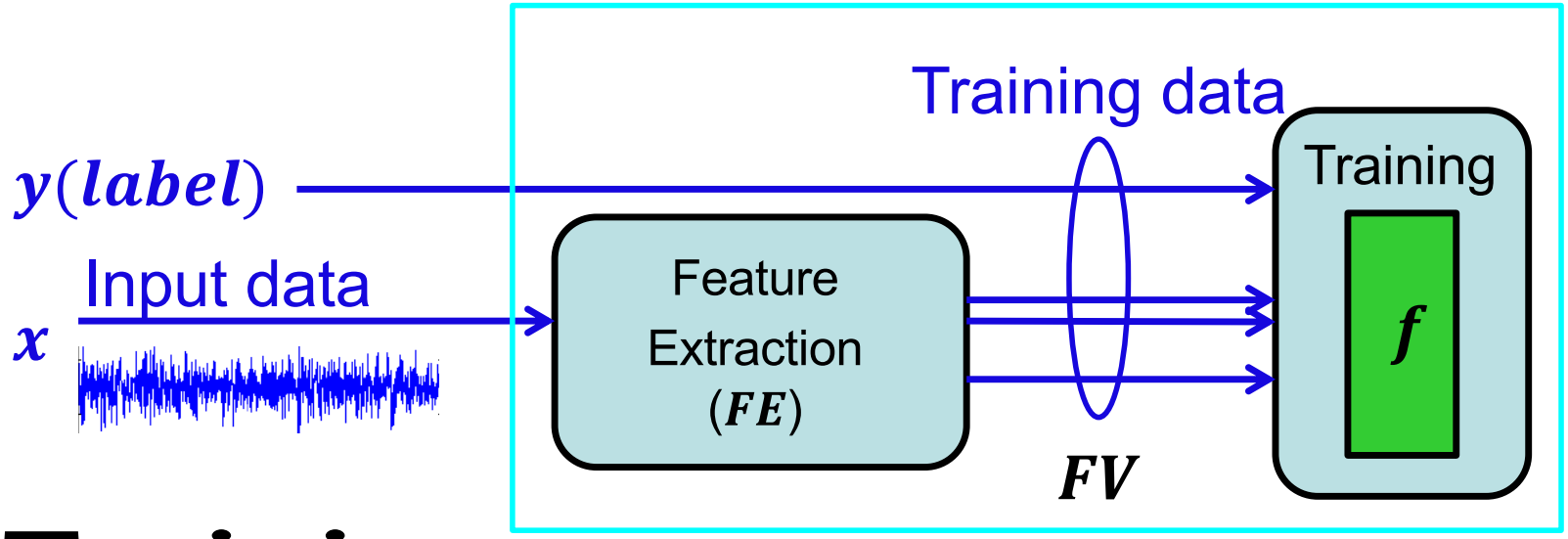


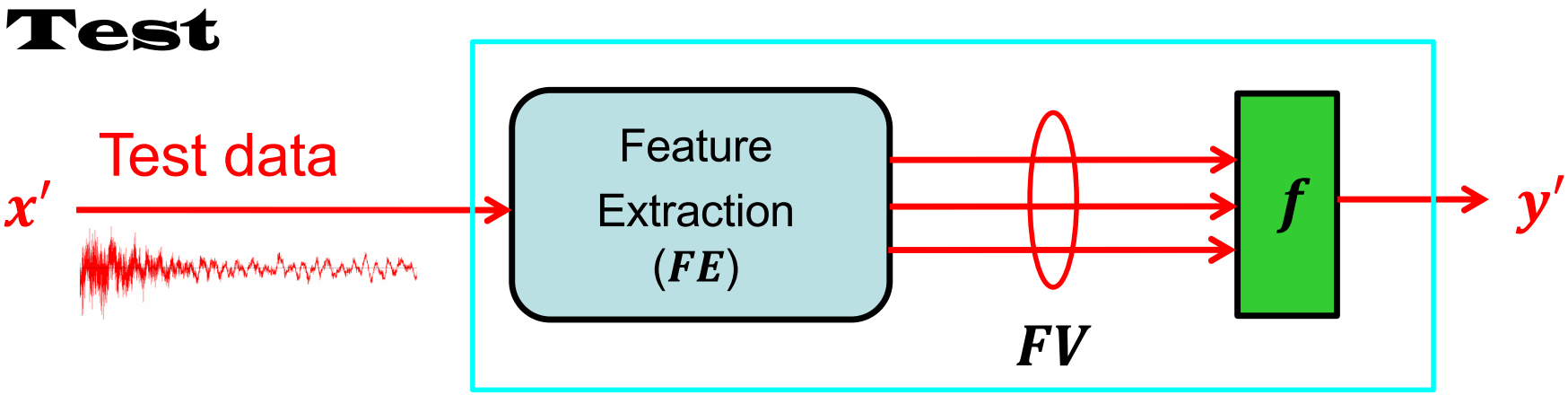
Classification



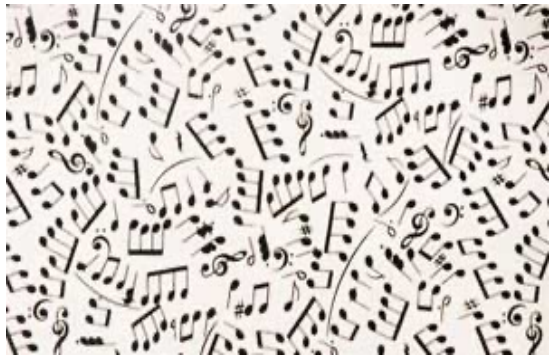
(supervised learning is when you have labelled training data)

ML Procedure for Classification



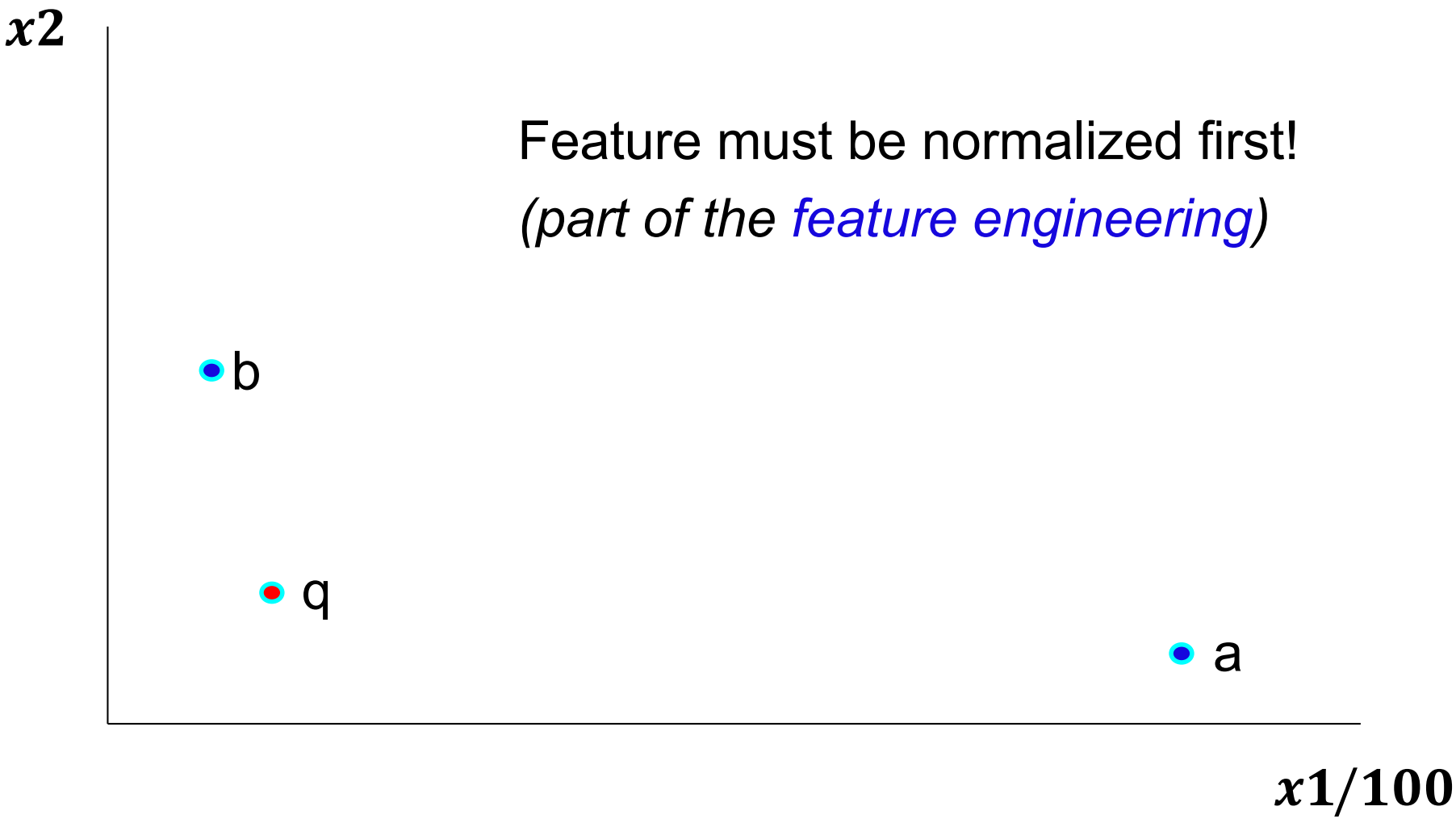


Understand the Music Dataset!

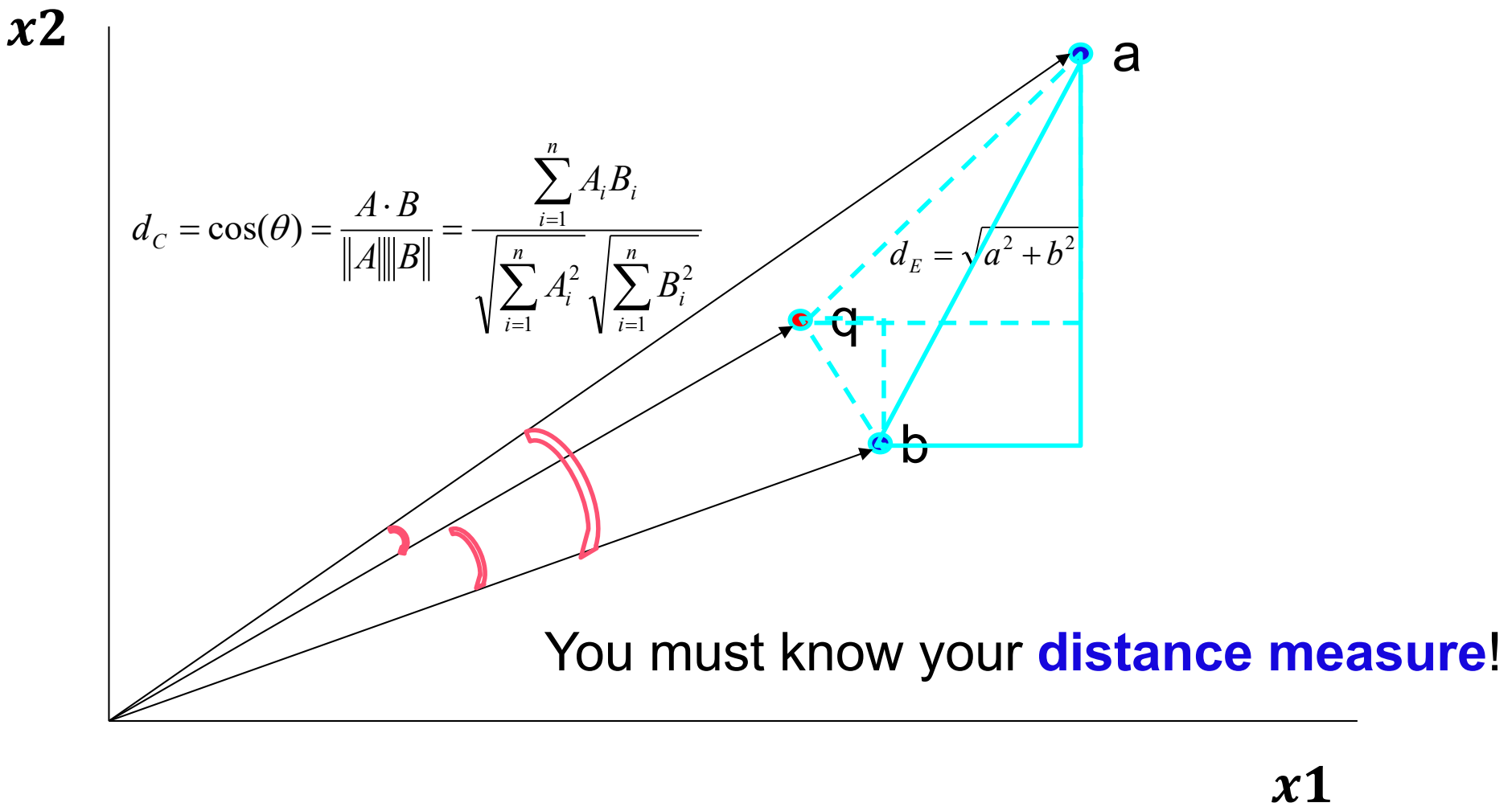


https://labrosa.ee.columbia.edu/millionsong/pages/example-track-description

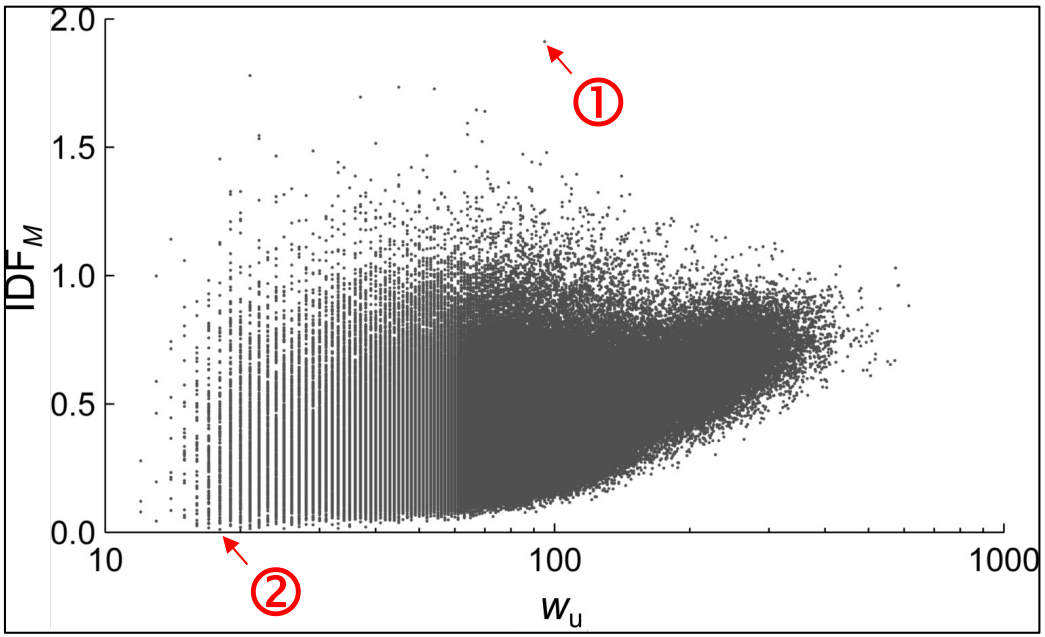
1. Which song is closer to my query?



2. Which song is closer to my query?



Lyrics feature example: IDF_M



Distance
&
Similarity

① There's Syria, Lebanon, Israel, Jordan
Both Yemens, Kuwait, and Bahrain
The Netherlands, Luxembourg, Belgium, and Portugal
France, England, Denmark, and Spain
— "Yakko's World" from *Anamaniacs* (1993)

② I can't think straight
Help me now before it's too late
Now what do I care?
'Cause we're going nowhere
— "Going Nowhere" by Cut Copy (2004)

Feature Selection/Learning

Selecting good features which distinguish well between the genres and increase the accuracy of the algorithm.

Features are classified into:

DL based
feature learning or representation learning
(e.g., Wav2vec)

Supervised Learning

Prediction Problem: Given a training data set ($\boldsymbol{x_i}$, $\boldsymbol{y_i}$) i=1,2,…,N, learn a function that predicts $\boldsymbol{y}$' given $\boldsymbol{x}$' with as few misclassifications as possible.

For example:

$\boldsymbol{x_i}$ – a song

$\boldsymbol{y_i}$ – a class label (e.g., 0 for rock, 1 for jazz, …)

Potential Models

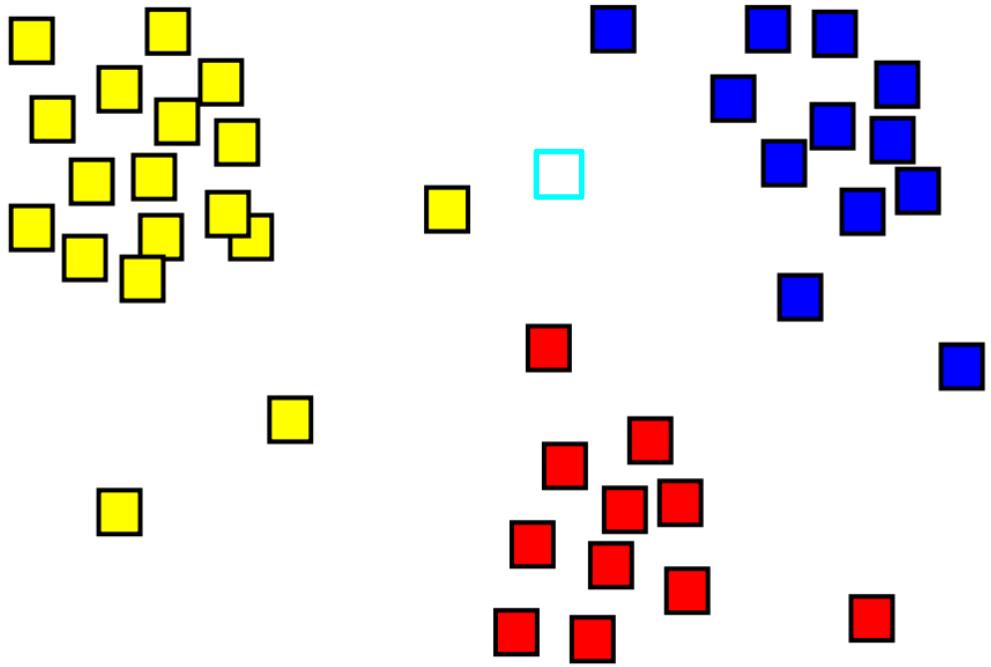
1-Nearest Neighbours (NN)

K-Nearest Neighbours (kNN)

Support Vector Machine (SVM)

Artificial Neural Network (ANN)

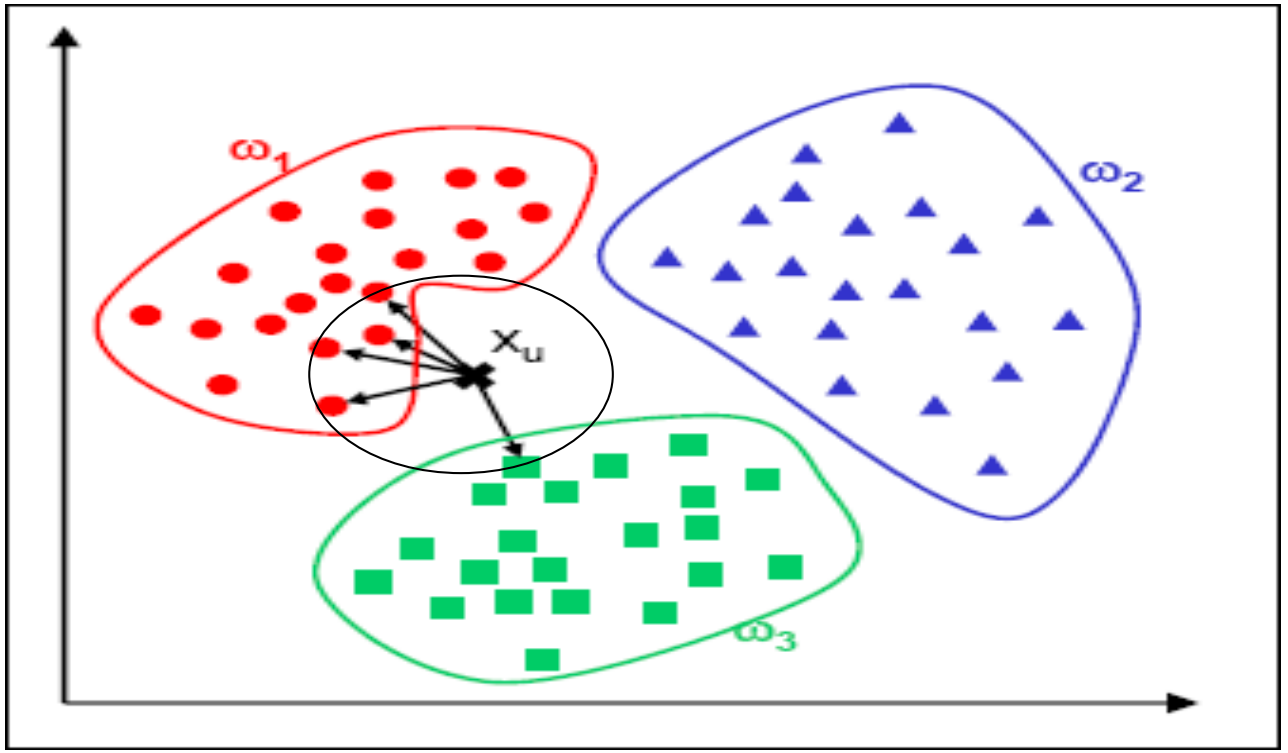
1-NN



Which class label should we assign to the new sample?

K-Nearest Neighbor Classifier

Classify unlabeled data according to the dominant class of the k nearest neighbors in the training data set.



K-Nearest Neighbours

- KNN is a classification algorithm.
- Classifies datasets based on a distance measure with its closest K neighbours.
- Euclidean or cosine distance (similarity measure) are used frequently
- Training consists of storing the feature vectors and class labels of the training samples.
- Testing consists of assigning a label to test object which is most frequent among the k training samples nearest to that point.

K-NN: Advantages and Disadvantages

Advantages: easy to implement

Disadvantages:

- Large storage requirements
- Computationally intensive recall
- How do we pick *k*?

In short, it is a data dependent method which is particularly challenging for embedded systems!

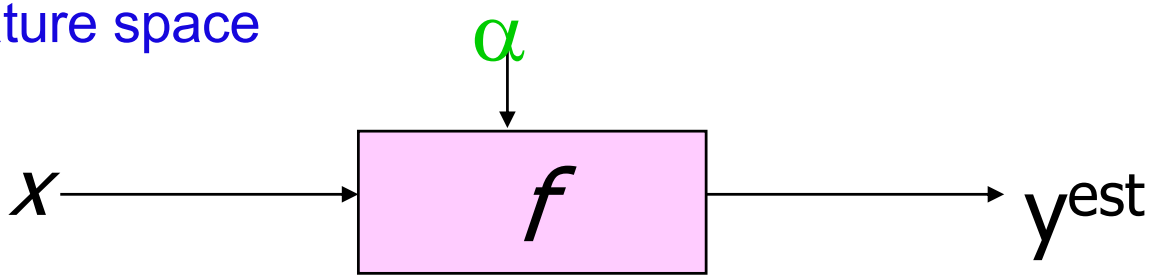
Can you think of a better method
to address above disadvantages of k-NN?

Let's take Voice Activity Detection (VAD)
as an example to see how to solve the problem
with much less computational workload!

VAD example

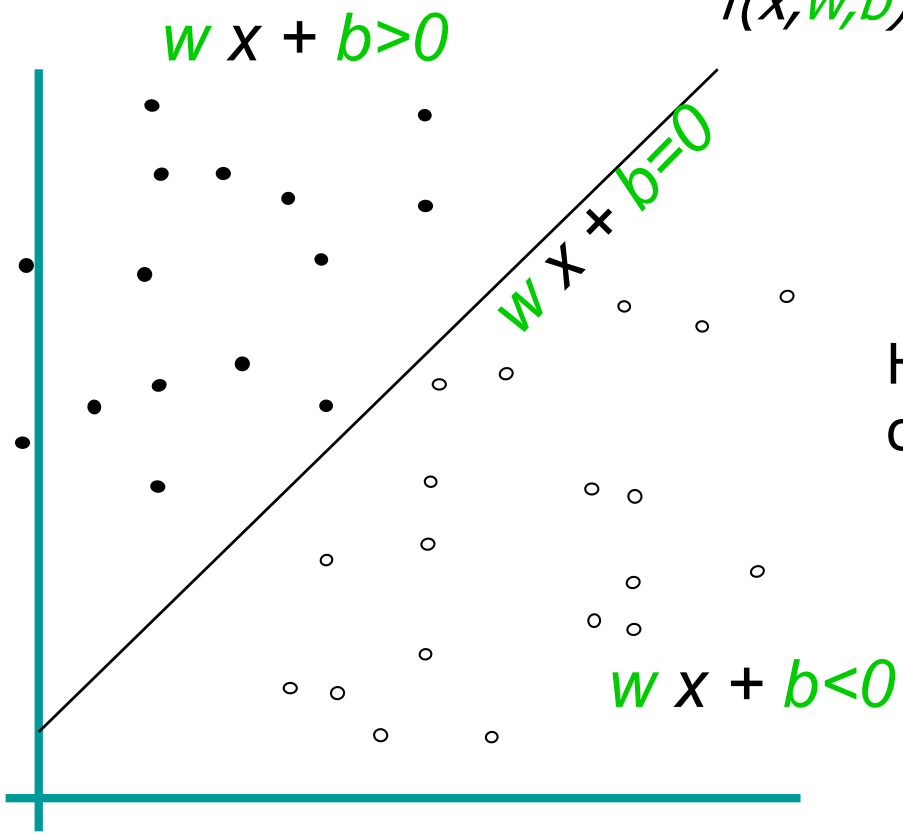
Linear Classifiers

Visualize the data to 2D feature space



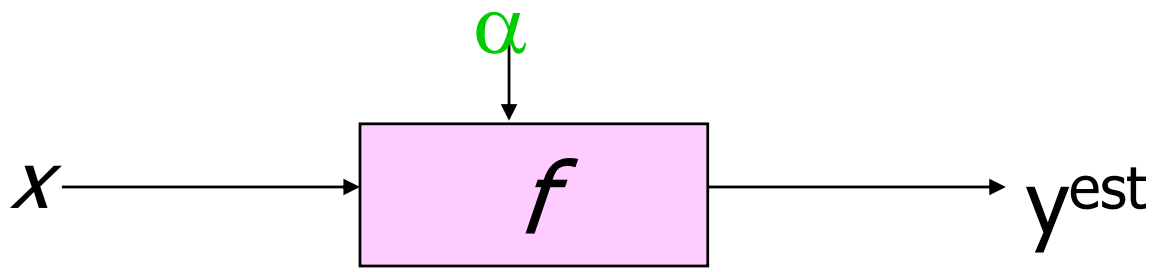
$f(x,w,b) = sign(w\,x + b)$

- denotes +1
- ◦ denotes -1



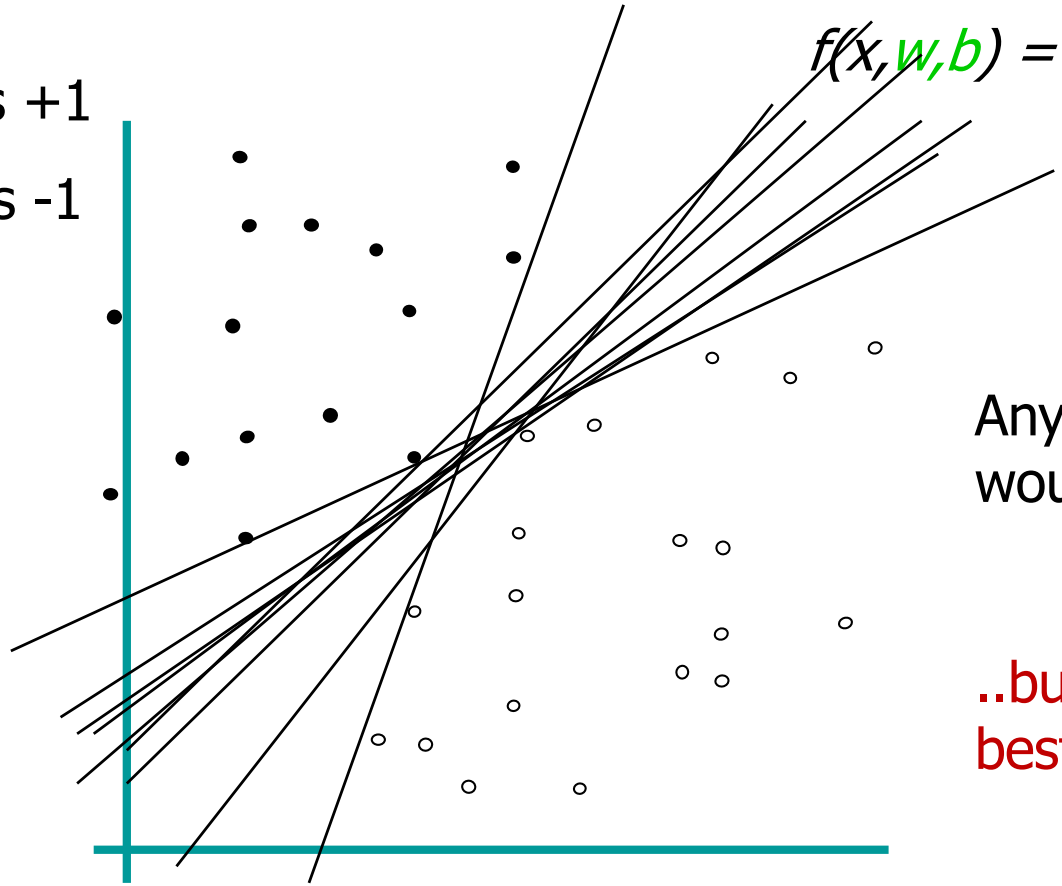
How would you classify this data?

Linear Classifiers



$f(x,w,b) = sign(w\,x + b)$

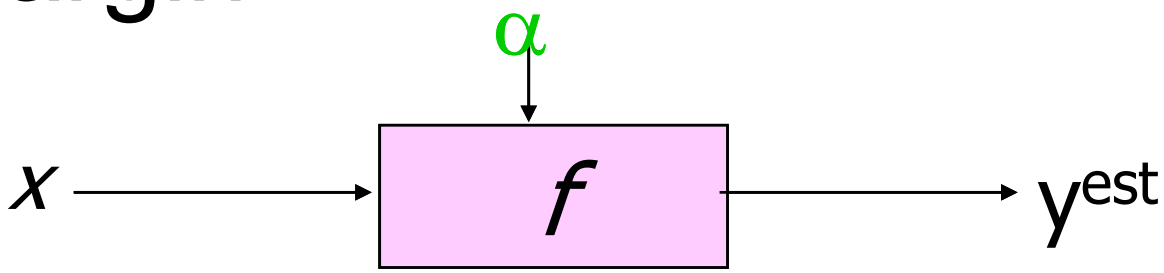
- denotes +1
- denotes -1



Any of these would be fine..

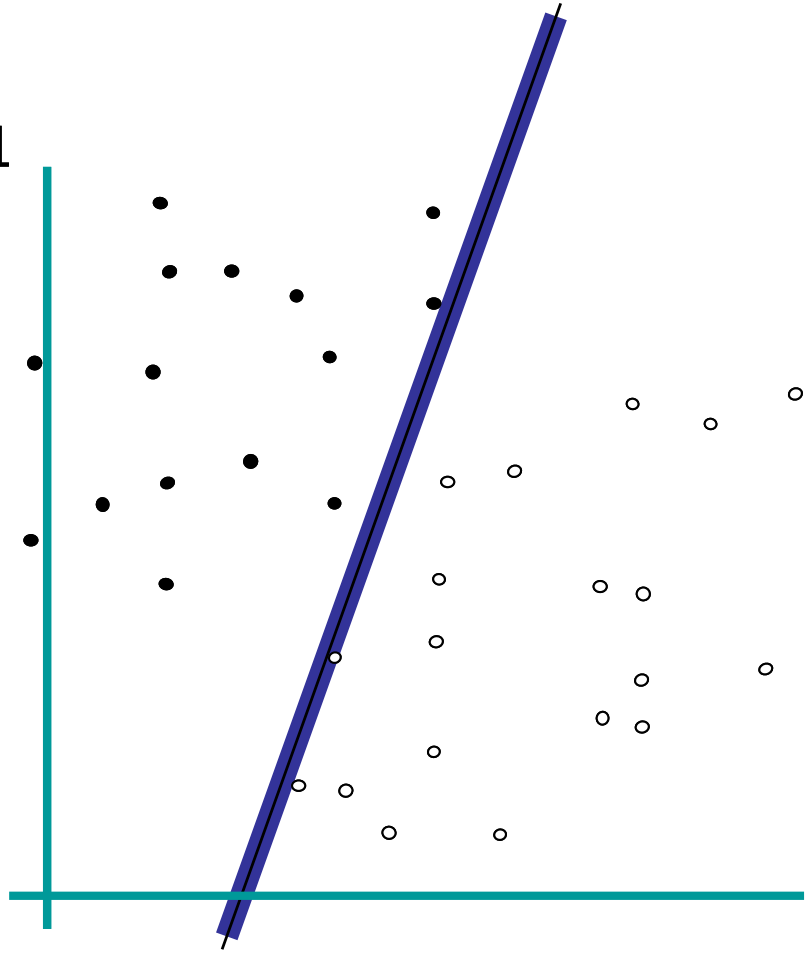
..but which is best?

Classifier Margin



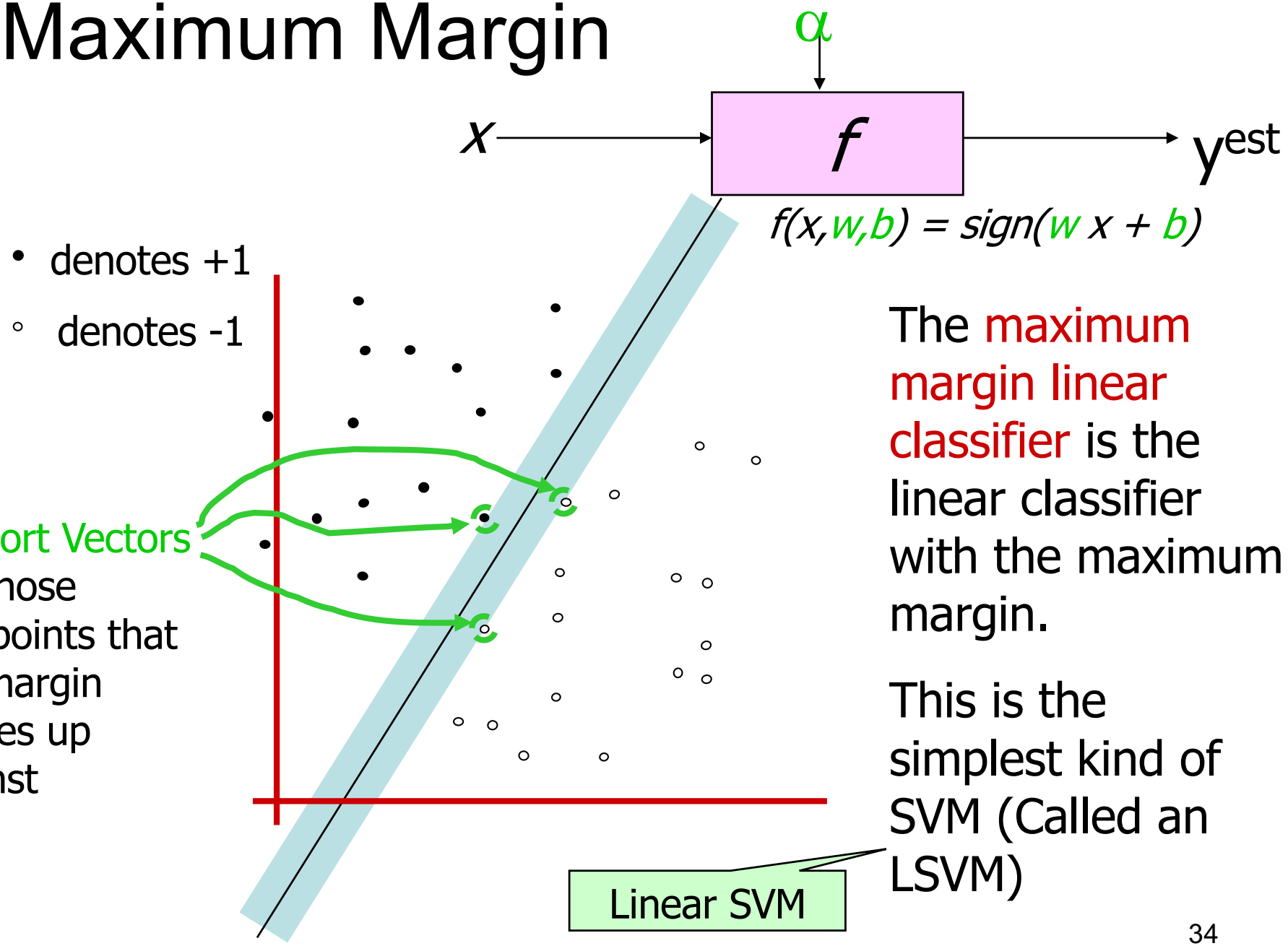
f(x,w,b) = sign(w x + b)

- denotes +1
- denotes -1



Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin



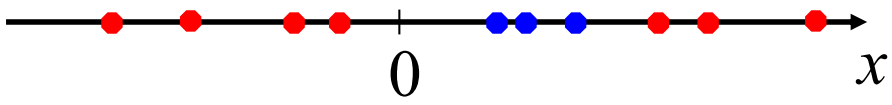
The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

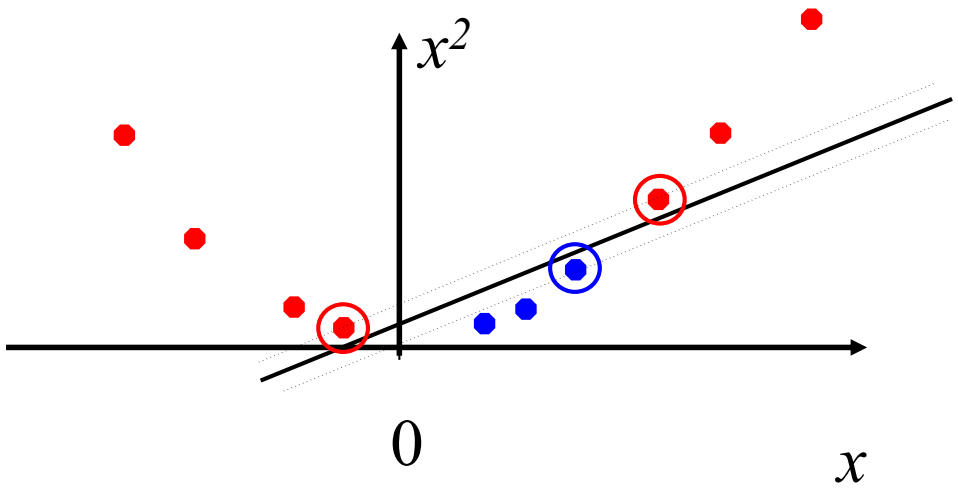
Why is it a good idea to maximize the margin?

Kernel SVM

- But what are we going to do if the dataset is just too hard?

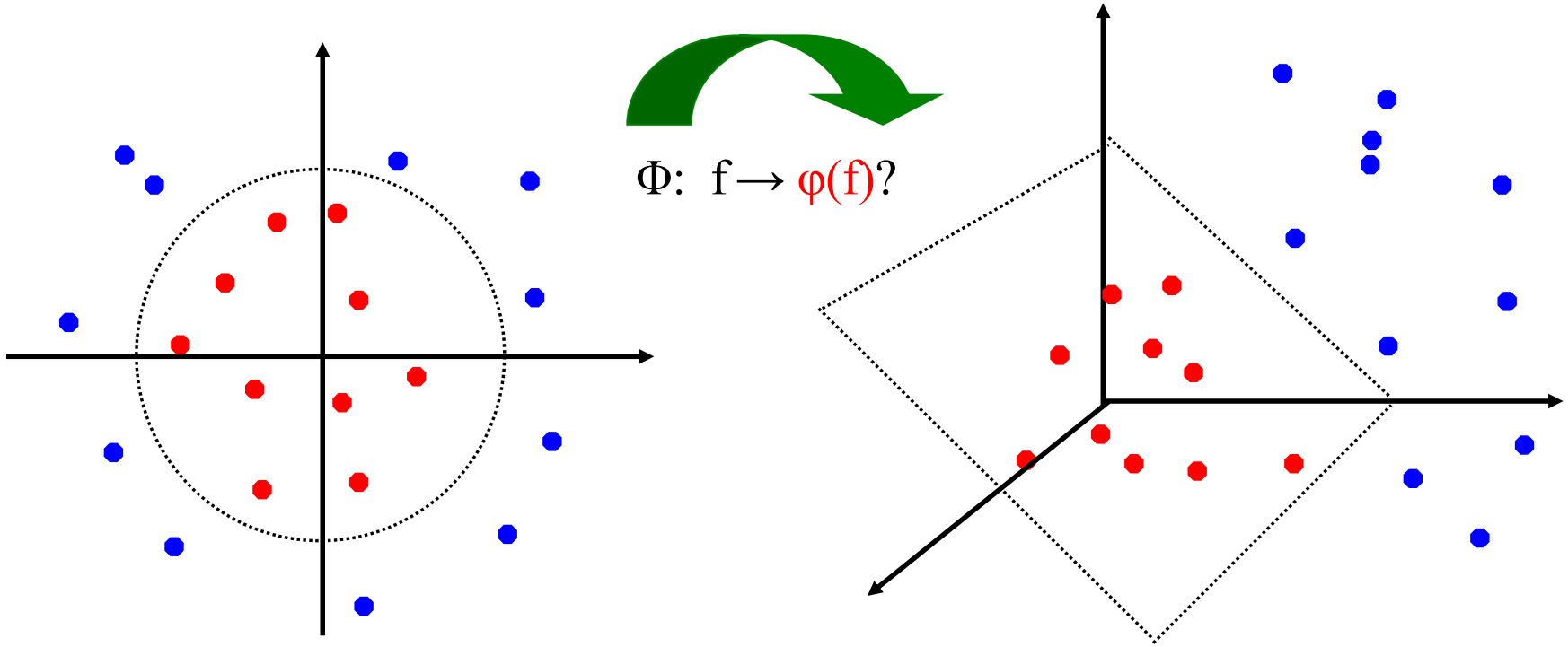


- How about… mapping data to a higher-dimensional space:



Kernel SVMs: Feature Space (2D example)

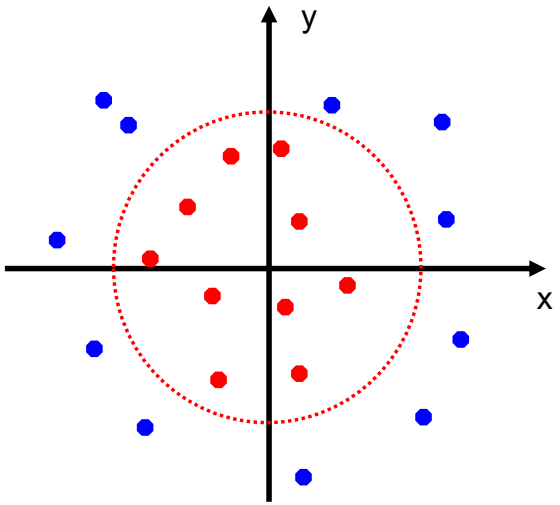
- General idea:
  - the original input space can always be mapped to some higher-dimensional feature space where the training set is separable.



Can you suggest a solution without increasing the dimensionality?

Convert a function

$F(x,y)$: $X^2 + Y^2 = 4$

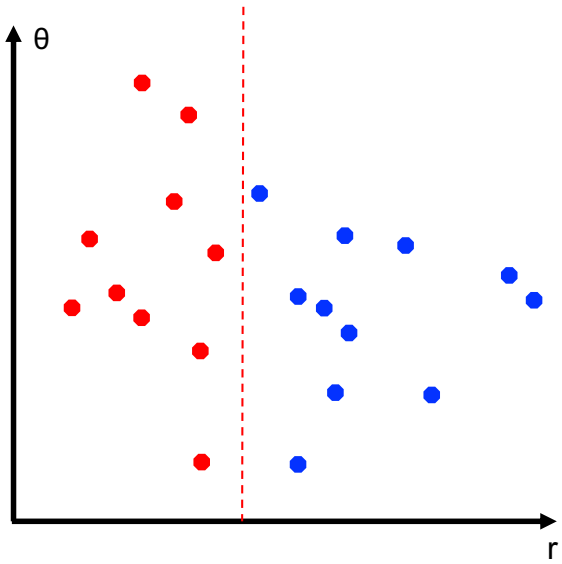


Rectangular (Cartesian) coordinate system

$F(r, \theta)$: $r^2\cos^2\theta + r^2\sin^2\theta = 4$

$r^2 = 4$

$r = 2$



polar coordinate system

SVM: Advantages and Disadvantages

Advantages:

Quick to evaluate

Very accurate and powerful – non-linear kernels

Disadvantages:

Binary classification – can be heuristically use for multi-class

Non-trivial to pick parameters and kernel function

SOTA TOPIC:

Deep Learning and Neural Networks (NNs)

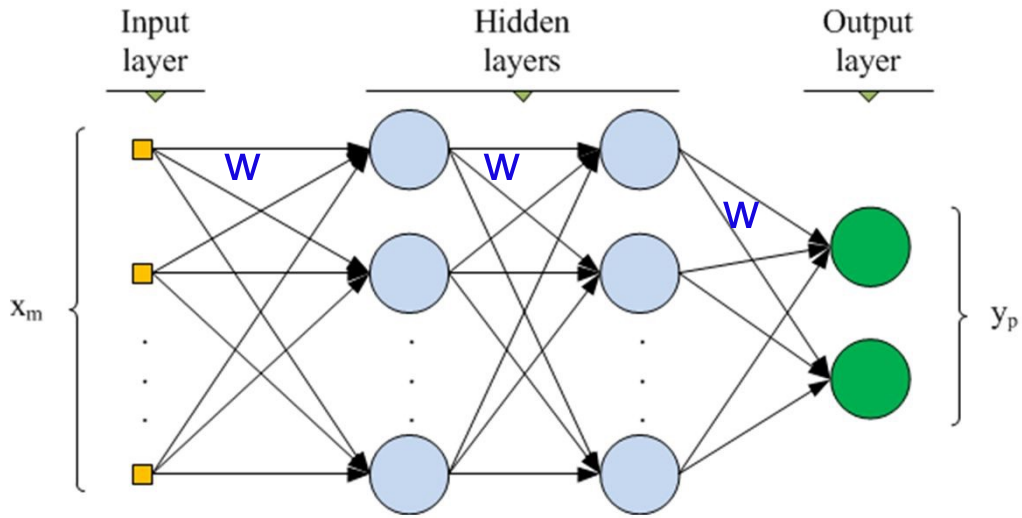
A NN tries to model human neural network in the brain!

Common NNs:

- Multi-Layer Perceptrons (MLPs)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Attention Neural Networks (ANNs)

Multi-layer Perceptrons (MLPs)

A MLP is a multi-layer structure with the first layer as an input and the last layer as outputs, a.k.a. Fully Connected (FC) neural networks.



Each layer could consist of one or multiple neurons.

The middle layers do not connect with the outer world, hence are called hidden layers.

VAD example

Multi-layer Perceptrons (MLPs)

In a fully connected NN, each neuron in one layer is connected with each neuron in the next layer.

Each connection between neurons has a weight (parameter *w*) associated with it.

Learnt knowledge of the model is encoded into the weights of the neurons.

Multi-layer Perceptrons (MLPs)

Advantages:

- Powerful to learn anything, theoretically

Disadvantages:

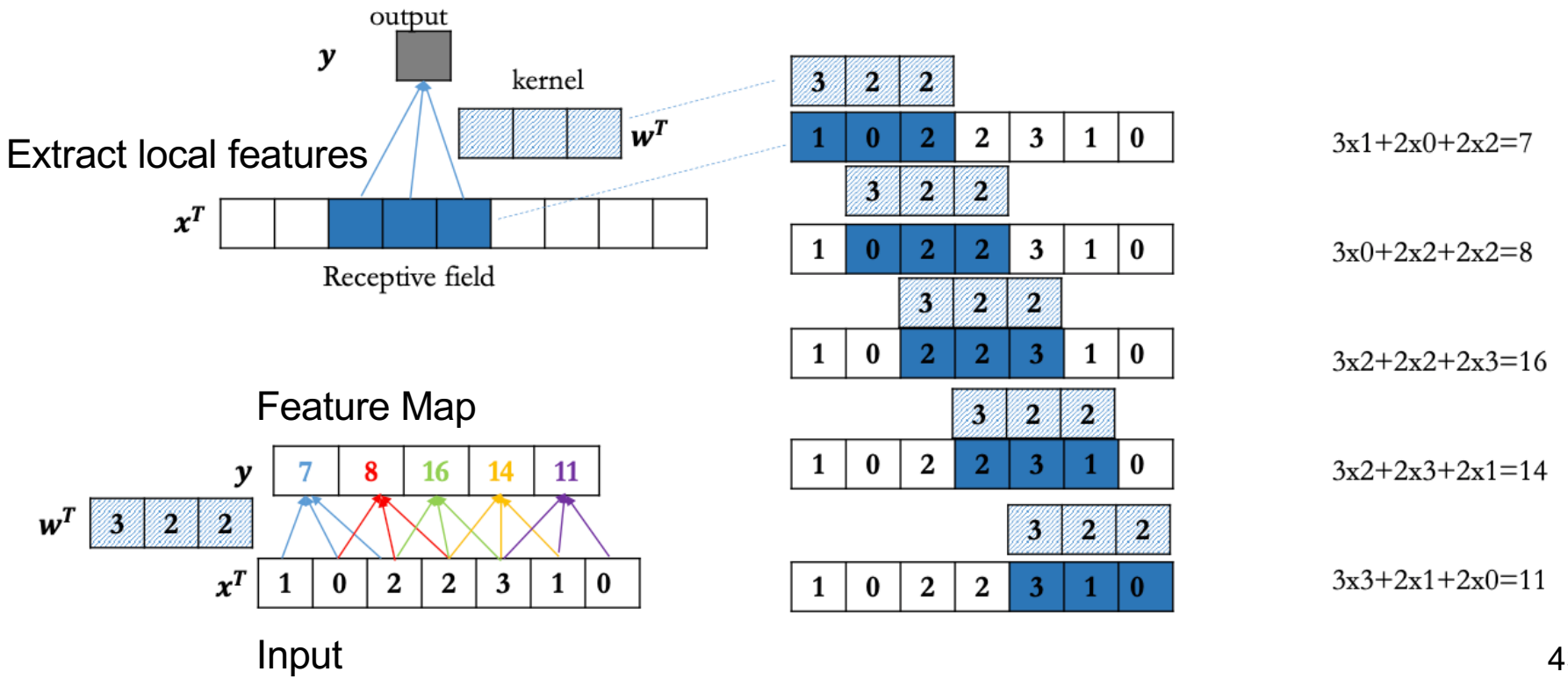
- MLPs do not consider any specific structure of data. Consider data that exhibit special structures, e.g. image, spectrogram

- Too many parameters (weights) to tune which require lots of data

Convolutional Neural Networks (CNNs)

Convolution: A linear transformation

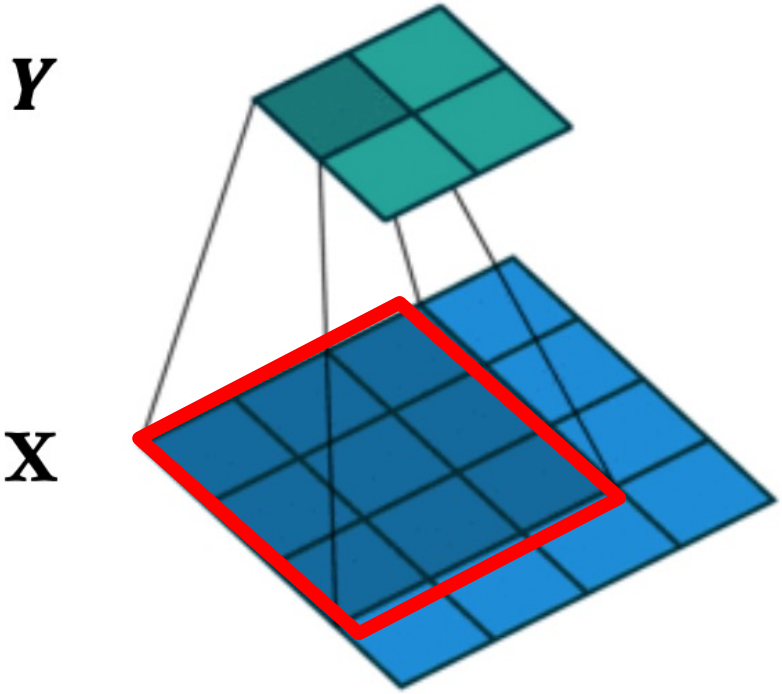
- 1D: Text/Signal processing
- Example:



Convolutional Neural Networks (CNNs)

Convolution: A linear transformation

- 2D: Image processing
- Example



What is the kernel in this 2D example?

CNNs vs. MLPs

Sparse connection

vs.

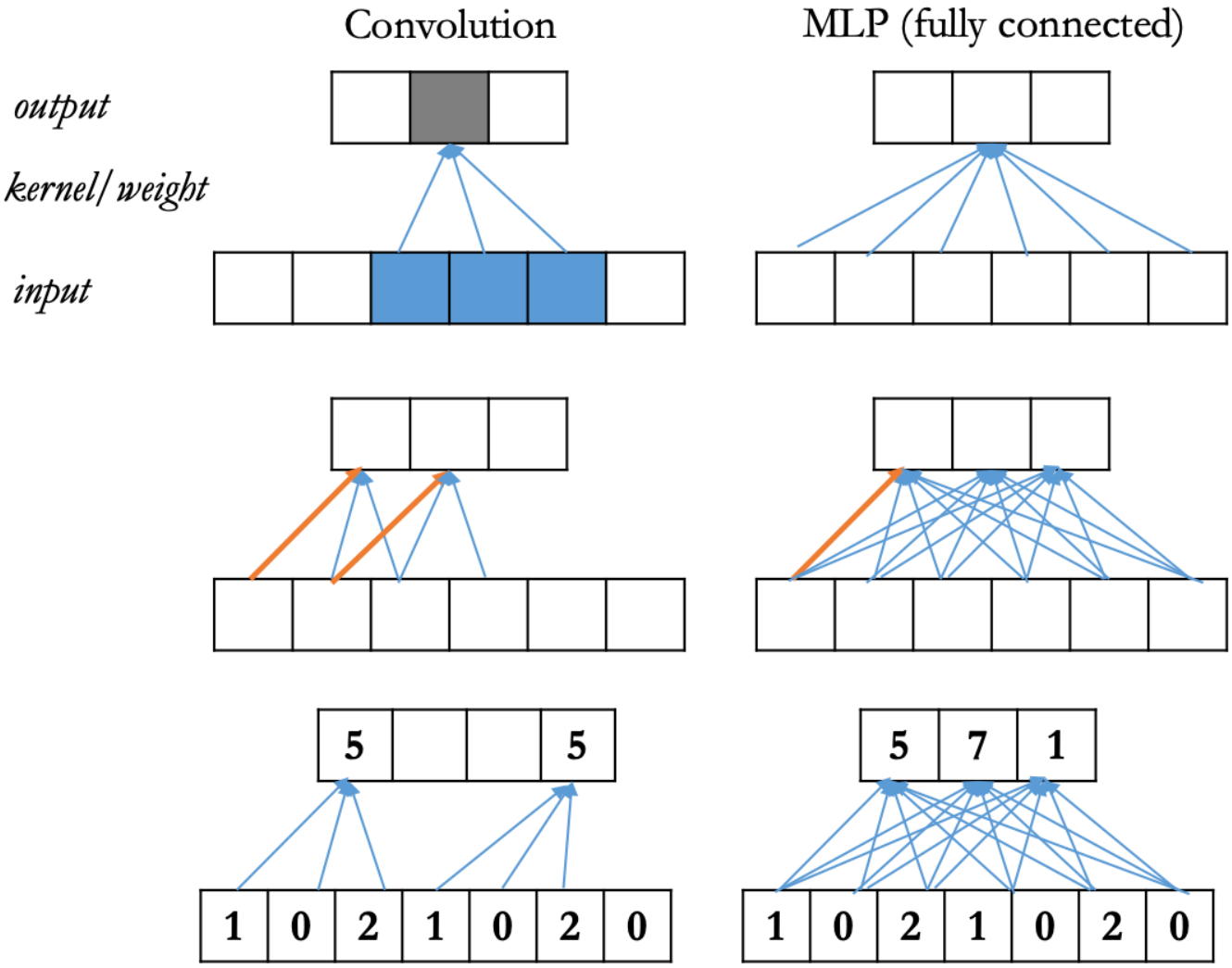
Dense connection

Weight sharing

vs.

Unique weights

Local invariant features



Convolutional Neural Networks (CNNs)

Common CNN architectures/Case Studies:

- LeNet5[1]
- AlexNet[2]
- VGGNet[3]
- ResNet[4]

[1] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[4] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

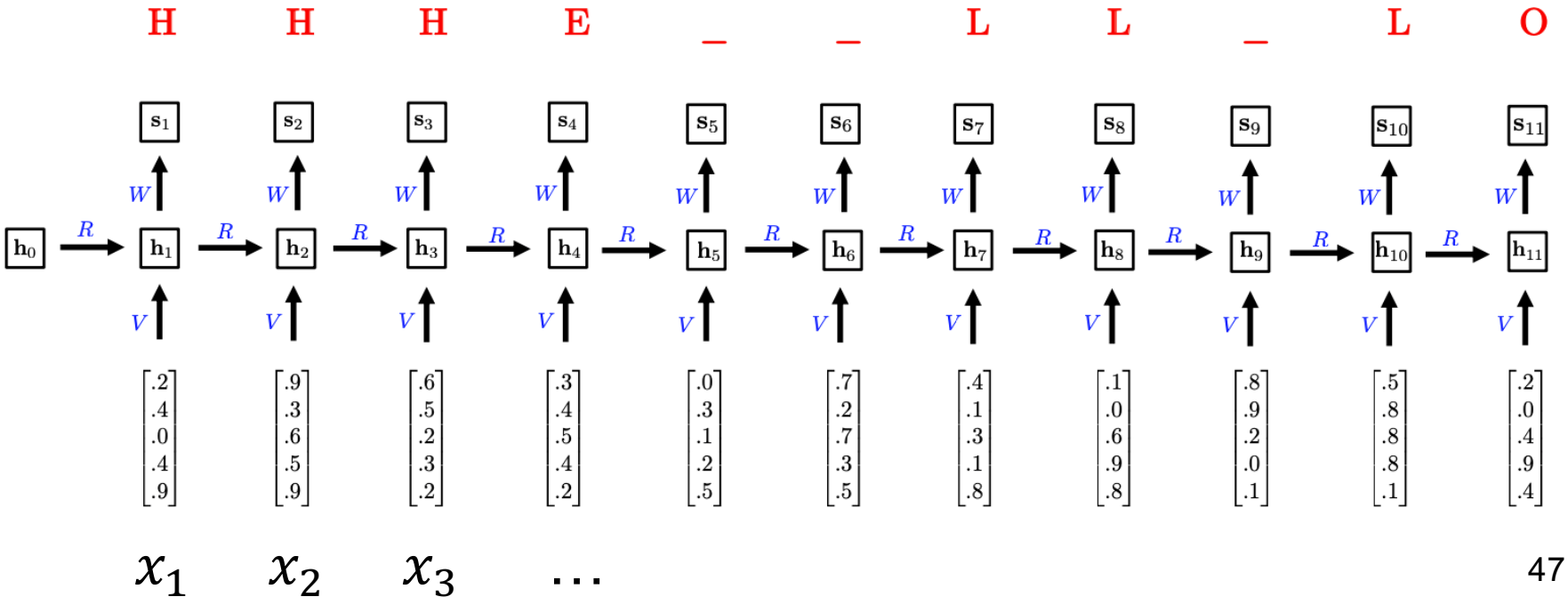
Recurrent Neural Networks (RNNs)

RNNs are good at processing a sequence of data.

Each element in a sequence is processed in the same way recurrently.

Weights(W, R, V) are shared.

Example:



Recurrent Neural Networks (RNNs)

Common RNN architectures/Case Studies:

- LSTM[1]
- GRU[2]

[1] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[2] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Attention Neural Networks (ANNs)

Attention: mimic cognitive attention

Motivation: devote more focus to important parts of data

- Maps a Query (Q) and a set of Key-Value (K, V) pairs to an output
- The output describes how well the pair matches the Query

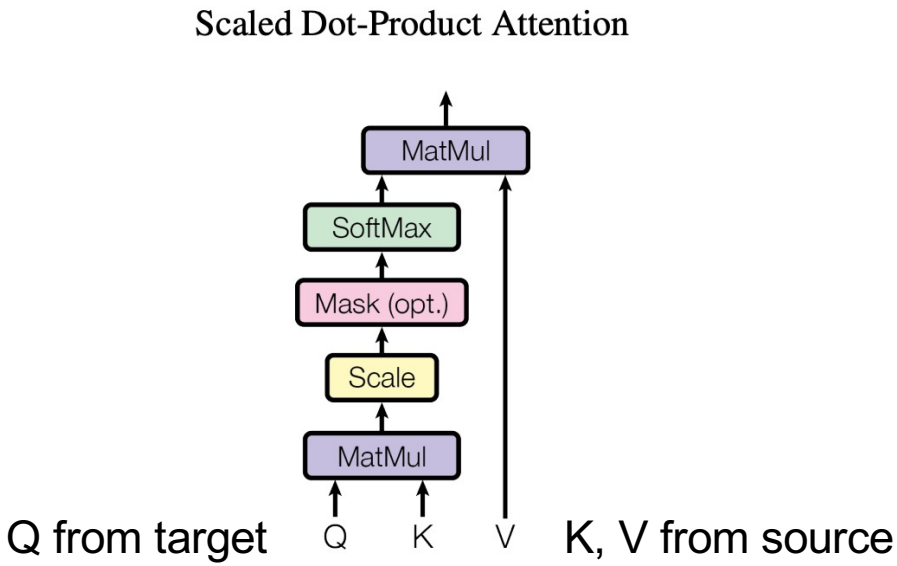
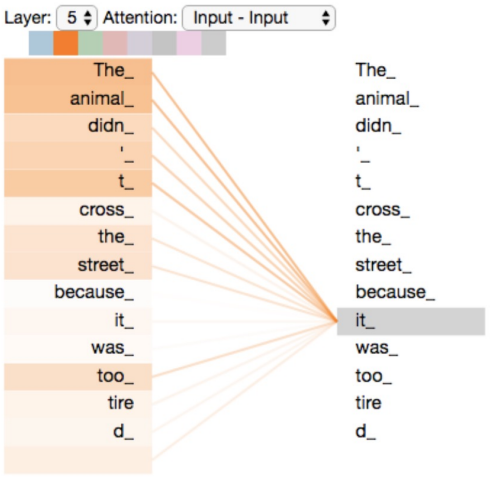


Figure Source: Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Attention Neural Networks (ANNs)

Self-Attention

- Query and Key-Value pairs are from the same sequence
- Represent a word by considering the words in the context



Cross-Attention

- Query and Key-Value pairs are from the different sequence

Multi-Head Attention

- Consists of several attention layers in parallel
- Attend to information from different scopes

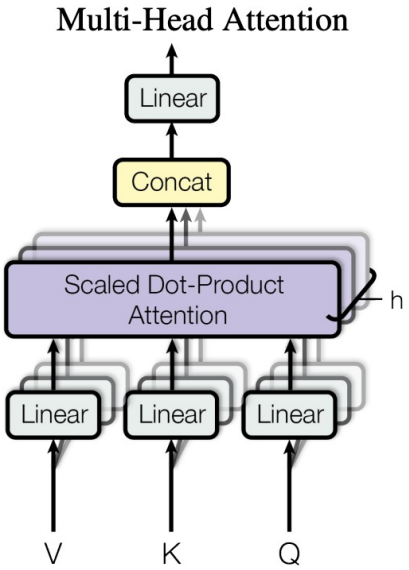


Figure Source: Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Attention Neural Networks (ANNs)

Transformer

- Self-Transformer Encoder
- Self-Transformer Decoder
- Transformer Encoder-Decoder
  - Which one is Query set? Input or Output?

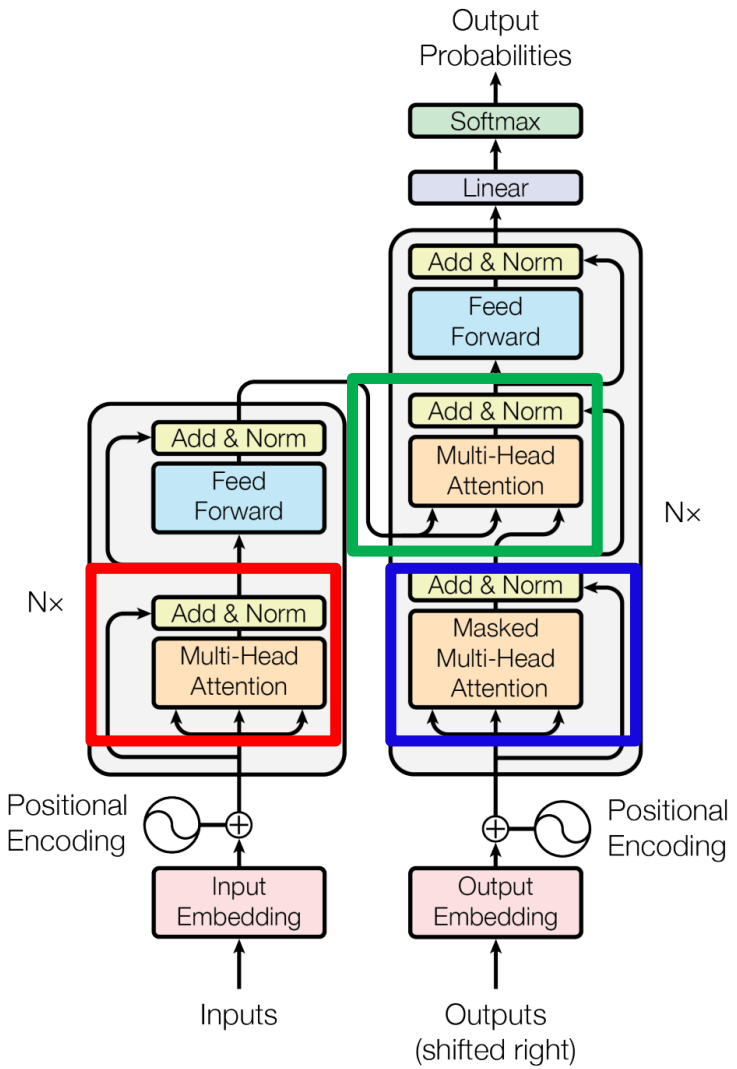


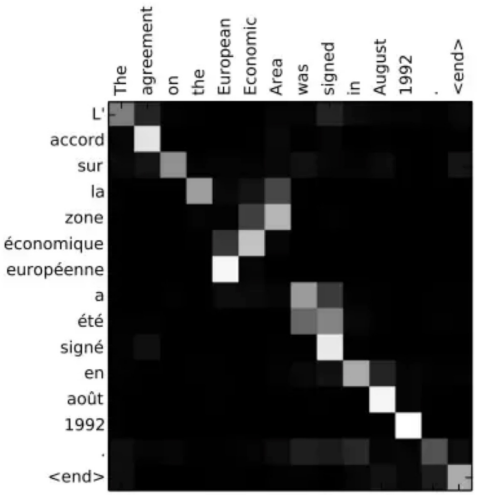
Figure 1: The Transformer - model architecture.

Figure Source: Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Attention Neural Networks (ANNs)

Example:

- Machine Translation
  - Match source and target words



- Sentiment Analysis
  - Focus on relevant words
  - Eg. Decide the book's review good or bad

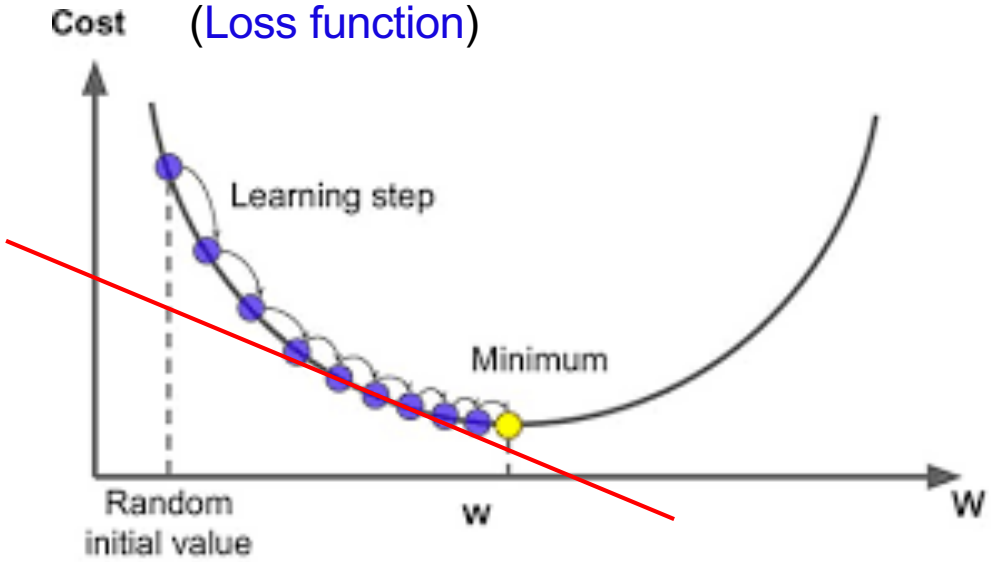
    Steven is arrogant but his book is awesome

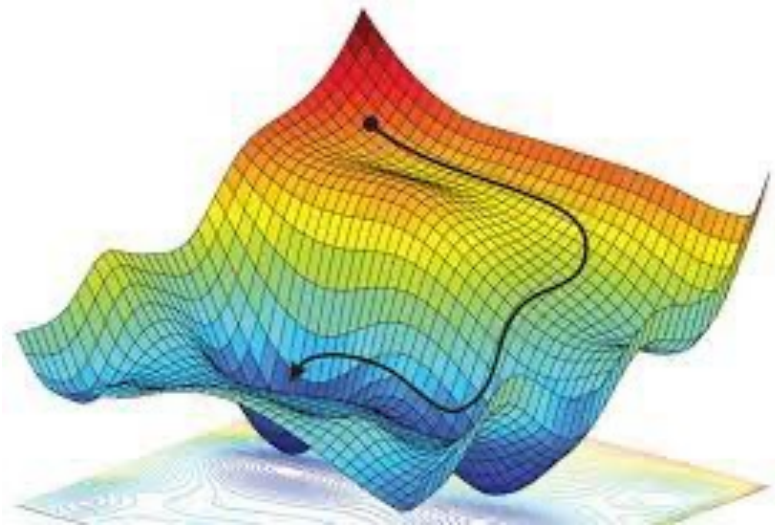
Steps of Training a NN

- Pass data to model via **forward propagation**
- Calculate **loss** on output
- **Gradient descent** minimizes the loss.
  - By calculating the gradient of the loss function and **updating weights**
  - Gradient is calculated via **backpropagation**

$$\frac{d(loss)}{d(weights)}$$

Gradient Descent





e.g.,
CS4347 revamp from 2021
(AMT+ASR+SED)

Expected learning outcomes

Student feedbacks

Will you use DNNs for your project? And Why?

If so, which type of DNNs will you use? And Why?

Refer to the 3 videos to be uploaded to Canvas!

- Perceptron
- Gradient descent
- Backpropagation

Model Assessment

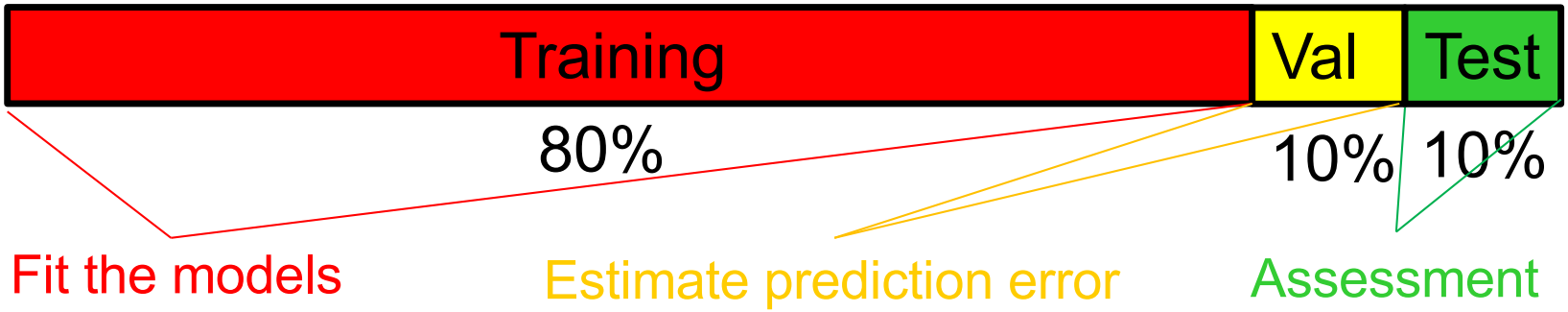
Say we have two classifiers

- kNN vs. SVM
- or SVM vs. Kernel SVM
- or two different DNNs

Which one works better?

Train / Validation / Test

- Train – As much training data as possible
- Validation (Val) – Used to tune hyper parameters (e.g., number of hidden units, layers for NNs)
- Test – Used at very end to describe performance



Training/Validation/Test Error

We compute the error using this equation:

$$E = \frac{1}{n}\sum_{i=1}^{n} L(Y_i, Y_i')$$

Where L is a ***loss function***

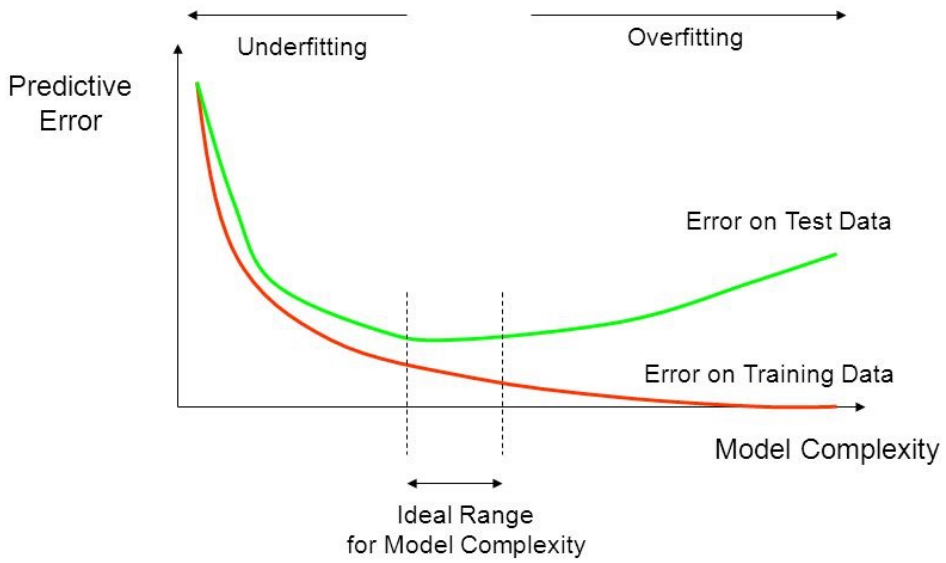
Training Error: The average error over the training sample.

Validation Error: The average error over the validation sample.

Test Error: The average error over the test sample.

Note: The training error rate can dramatically *underestimate* the test error rate.

Underfitting and Overfitting



As the model becomes more and more complex, it uses the training data more and is able to adapt to more complicated underlying structures.

Training error consistently decreases with model complexity but performs worse on test data (higher test error)

Evaluation Metrics

		Actual Label	
		Positive	Negative
Predicted Label	Positive	**True Positive (TP)**	**False Positive (FP)**
	Negative	**False Negative (FN)**	**True Negative (TN)**

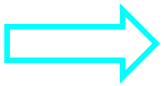
Accuracy	(TP + TN) / (TP + TN + FP + FN)	The percentage of predictions that are correct
Precision	TP / (TP + FP)	The percentage of positive predictions that are correct
Sensitivity (Recall)	TP / (TP + FN)	The percentage of positive cases that were predicted as positive
Specificity	TN / (TN + FP)	The percentage of negative cases that were predicted as negative

Confusion Matrix

	Voice (Predicted)	Non-Voice (Predicted)	Accuracy
Voice (Actual)	27	6	81.81
Non-Voice (Actual)	10	57	85.07
Overall Accuracy			83.44

Topics Today

Part A: Machine Learning Fundamentals



Part B: Practical issues

Part C: Classification libraries

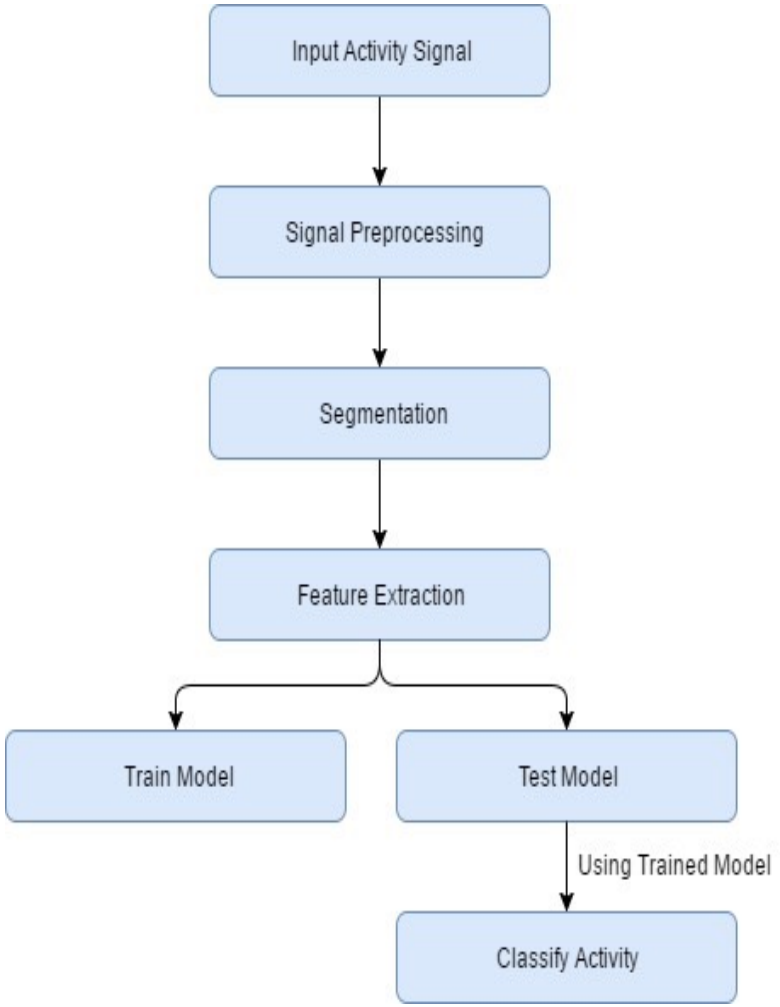
Practical Steps for Activity Recognition

Sensor Inputs

Signal Preprocessing

Segmentation

Feature Extraction



Classification of Activity (e.g., VAD)

Dataset

The devices used to record the participants' singing data.

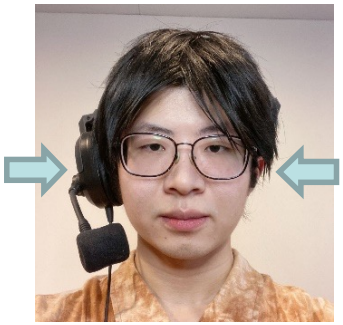
1) Professional condenser microphone
2) Video camera
3) **eSense earbud**

   eSense is an earable device for research purpose, equipped with a speaker, a microphone and an inertial motion unit (IMU).

   An IMU is an electronic device that measures a body's specific force and angular rate, using a combination of accelerometers and gyroscopes.

eSense device

Earphone



eSense

Sensor Inputs

Each sensor provides raw inputs to the system.

Sensors can be of different types e.g., microphone, accelerometers, gyroscopes, proximity sensors, camera etc.

Each sensor may have more than one axis.

One or multiple sensors can be used to capture an activity or event.

Preprocessing

Different preprocessing techniques can be applied to the raw data to smooth the signal or remove the noise.

The signal can be normalized or scaled before being processed.

Different types of filters can be used:

- Low Pass Filters: To remove noise from the raw input signals
- High Pass Filters: To remove the effect of gravity from the raw input signals

Segmentation

Identifying the start and end of the actual activity from the preprocessed signal.

Important for recognizing the activity accurately.

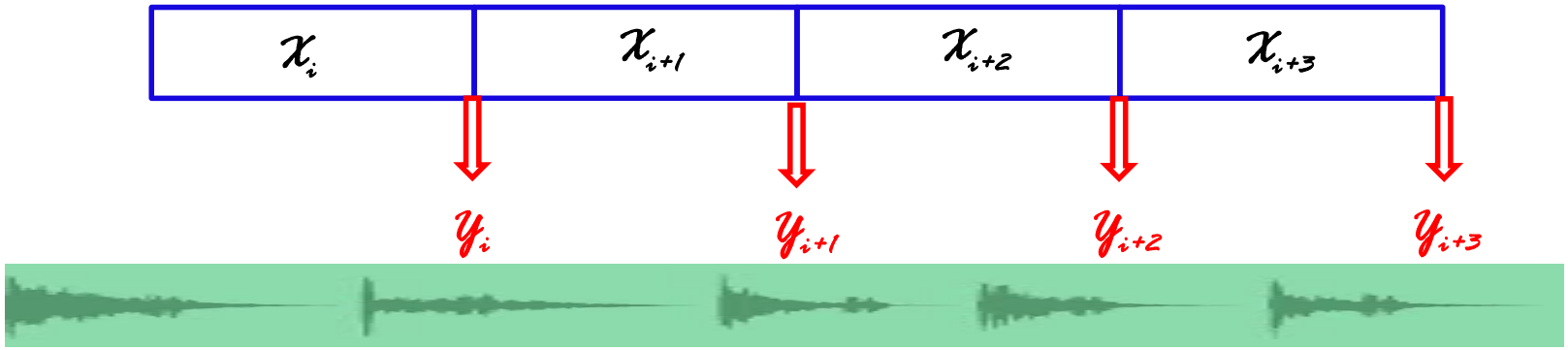
Different segmentation approaches can be used.

Another technique is to identify when the movement starts. It can be marked by a sudden change in accelerometer values.

How do we do segmentation in practice?

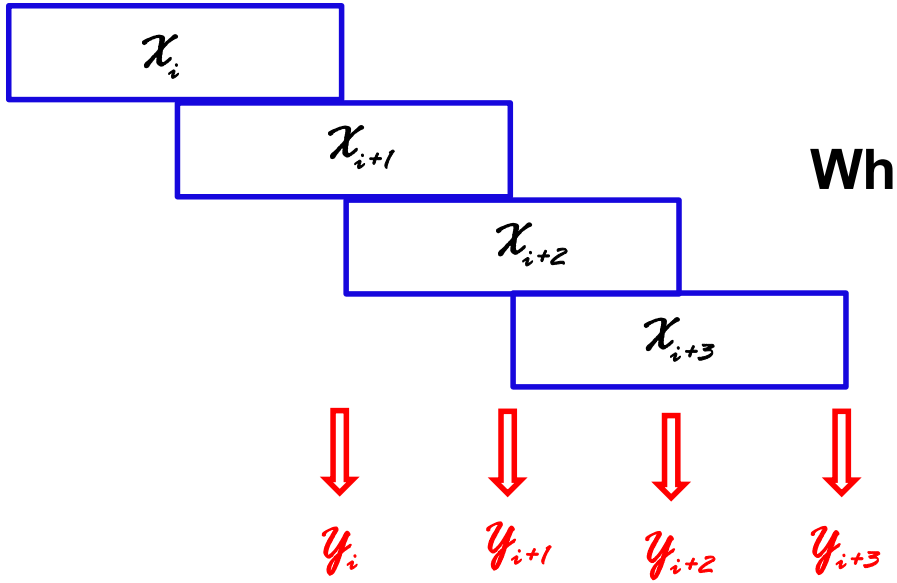
x_i – features extracted from sensors

y_i – a class label (voice or no voice)



1) Non-overlapping frames

Which technique is better? Why?



2) Overlapping frames

Feature Extraction

Different features can be extracted to classify the activities.

It is important to extract good features which distinguish well between the activities and increase the accuracy of the algorithm.

Features can be extracted for each axis e.g. x, y and z.

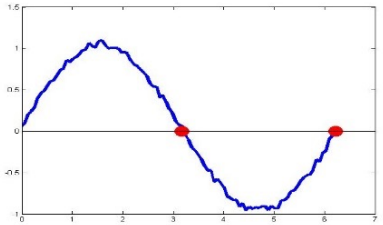
Features are classified into two types:

- Time Domain Features
- Frequency Domain Features

Feature Extraction

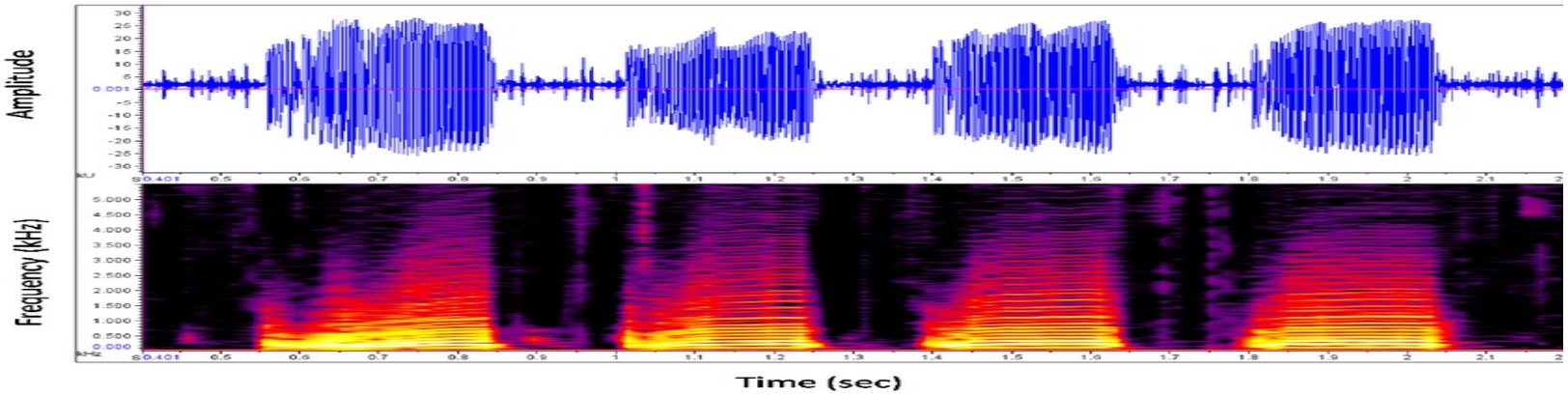
Time Domain Features include

Mean, Variance, Median, Mean Absolute Deviation, Zero-crossing



Frequency Domain Features include

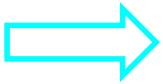
Spectral power, Entropy, Peak Frequency, Sub-band power



Topics Today

Part A: Machine Learning Fundamentals

Part B: Practical issues



Part C: Classification libraries

Commonly Used Libraries

There are many libraries/apis for ML, that support many different types of models

- Scikit-learn (python)
- PyTorch (standalone, python, etc)

Links

We will be looking at programming in more detail. To get ahead you could look into classification examples from scikit-learn:

http://scikit-learn.org/stable/supervised_learning.html#supervised-learning

If interested, you should also have a look into the PyTorch library, which provides more intuition into deep learning models.
https://pytorch.org/tutorials/beginner/pytorch_with_examples.html

Conclusion: 3 take home messages

1) Machine learning is useful for making predictions based on data.

2) Supervised learning predicts some sorts of labels based on data.

3) Evaluating the accuracy of prediction is important. (We have a range of evaluation metrics.)

On the enrollment of the class

I did not expect that the enrollments of CS4347/CS5647 has doubled this year. It is challenging to manage which might affect your learning experience (e.g., I am unable to recruit enough qualified TAs this year. As a result, we might be unable to be as responsible as we wish. It will take longer for us to grade assignments and projects).

Therefore, I advise those of you with zero background in DSP and ML not to take CS4347/CS5647 this year. Take your time to play around with DSP and ML, then enroll into this course next year.

This is to ensure that you all will have a joyful learning experience rather than frustration and bad stress!