

CE 395 Special Topics in Machine Learning

Assoc. Prof. Dr. Yuriy Mishchenko

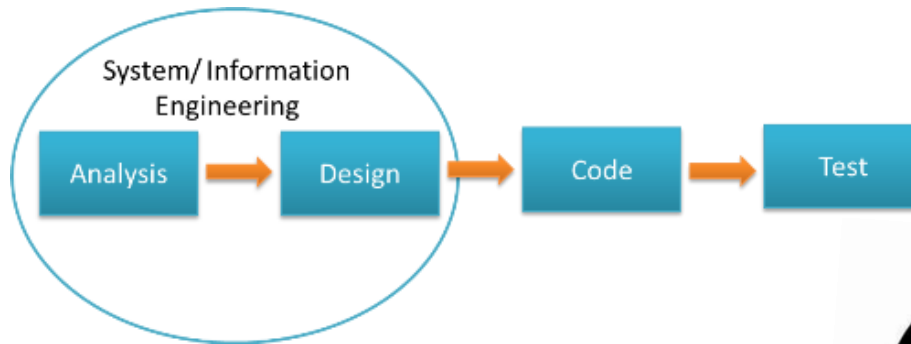
Fall 2017

INTRODUCTION

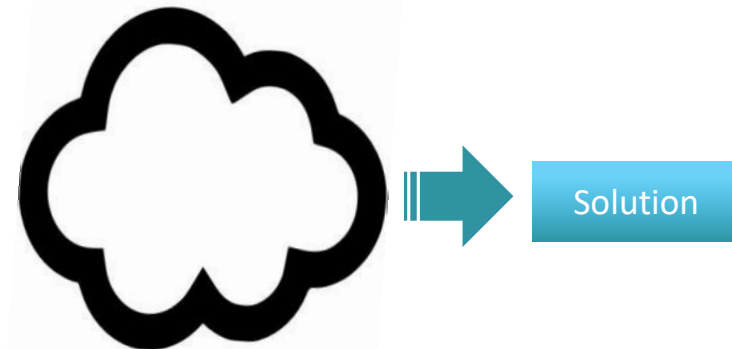
What is Machine Learning

- Machine Learning (ML) – branch of CS that studies algorithms that allow computers to learn
- A new model for software development - algorithms that learn themselves instead of being manually created by humans

What is Machine Learning



instead:



$\{(x \Rightarrow y)\}$

Blackboard software design pattern

What is Machine Learning

Google tries to learn to grasp with Deep Learning



What is Machine Learning

Learning in Machine Learning refers to algorithms that let computers *gain ability* to perform tasks *without being specifically programmed* for them

What is Machine Learning

“Learning” in ML learns a functional relationship $y=f(x)$ from a set of examples of that functional relationship (x,y)

Examples:

- Speech recognition (speech to text)
- Object recognition (image to objects)
- Robot control (pick up physical objects)
- Games (BlueMind and Go)

What is Machine Learning

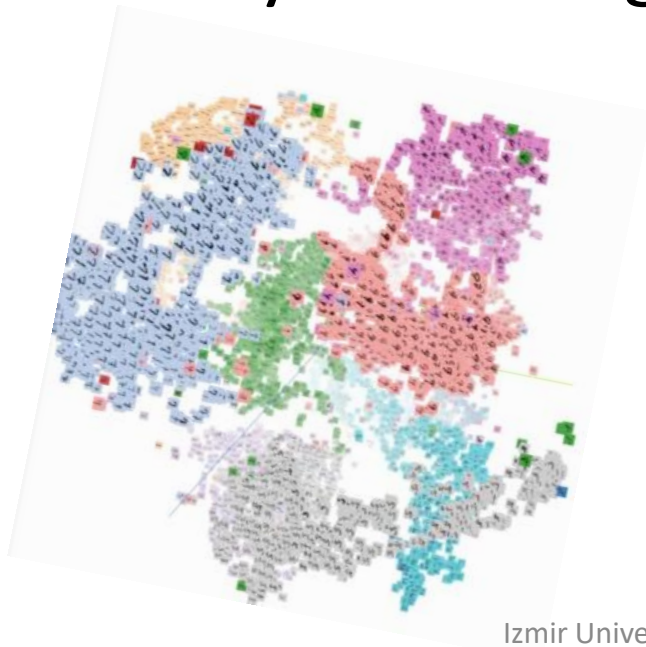
Examples of learning in ML:

$$\{f(x) \rightarrow y\}$$

- Recognizing speech:
x – sounds, y – words from dictionary
- Recognizing written letters:
x – images of written letters,
y – letter from alphabet
- Deciding credit: x – customer, y – yes/no decision
- Playing Go: x – board configuration + move, y – quality of the move

What is Machine Learning

- ML is substantially a branch of ***statistical inference*** that deals with ***function estimation*** in ***very high dimensional*** spaces, even though historically ML emerged from AI

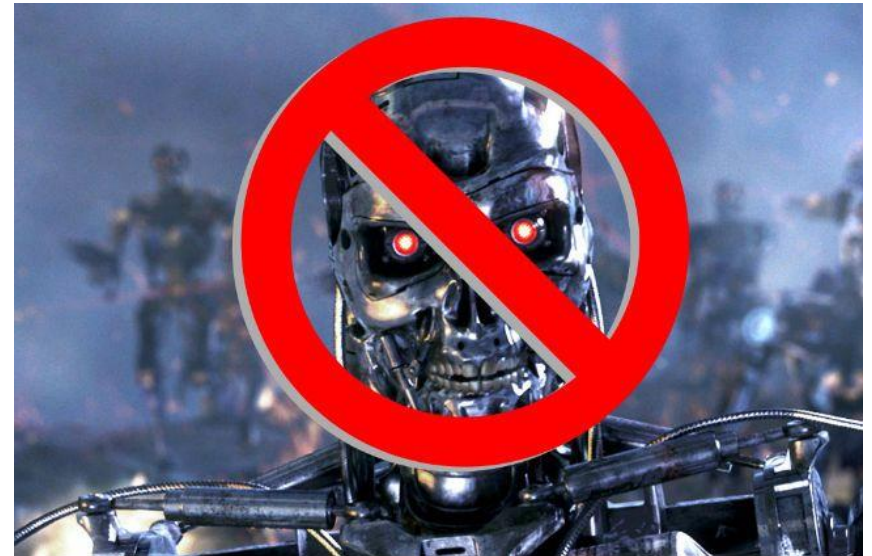


$$y = f(x_1, x_2, \dots, x_p)$$

ML and AI

Machine Learning IS NOT Artificial Intelligence

- **AI** is the branch of computer science that attempts to develop computer systems with the **intellectual capabilities of humans** (*whatever that means*)
- **ML** is the study and development of algorithms that enable computers to automatically acquire new skills



Learning is a *part* of intelligence, but not intelligence self

Types of (Human) Intelligence	Elements of Intelligence
Linguistic – ability to speak and use language	Reasoning – ability to make decisions
Musical – ability to create and understand music and rhythm	Learning – ability to gain knowledge and skills (auditory learning, episodic, motor, observational, etc.)
Logical/mathematical – ability to understand relationships and abstract ideas	Problem solving – ability to devise actions to arrive at a desired solution from a present situation
Spatial – ability to perceive and understand visual and spatial information including images, objects, and motion	Perception – ability to process sensory information
Bodily-kinesthetic – ability to use body or its parts including motion and control of other objects	Linguistic – ability to understand and use language
Intra-personal – ability to understand own emotions, intentions, and motivation	
Inter-personal – ability to understand other individuals' emotions, intentions, and motivation	

ML and AI

- **FIRST GRADED ASSIGNMENT: (essay)** Imagine that you knew an alien who's only ability was to learn things (as in "if **A** happens then do **B**"). Can that alien become **truly intelligent**? That is, **is it possible to learn to be intelligent**? If yes, explain how you think the alien could do that. If not, explain why not.



About is this course

- **This course is the prerequisite for the advanced courses in the ML option of CE/SE departments**
- The objective is to:
 - Prepare the mathematical background necessary for effective progress in the other ML option courses
 - Create a frame of reference to help one understand the place of different parts of ML
 - Provide a general overview of the principles making the foundation of modern ML

About is this course

- **CE 395** Special Topics in Machine Learning (math basics and foundation)
- **CE 344** Advanced Machine Learning (applied take on ML)
- **CE 455** Deep Neural Networks (deep neural networks)
- A Related Elective
 - MATH 485, CE 490, CE 462, CE 420, ...

About is this course

If not planning to continue with ML option:

- Take this course to gain understanding of the issues forming the foundation of ML methodology
- Take on an out-of-classroom study using one of the ML packages out there



About is this course

- Week 1 – Introduction to signals and sampling
- Week 2 – Introduction to filters
- Week 3 – Introduction to Fourier transform
- Week 4 – Review of matrices and vectors
- Week 5 – Introduction to numerical optimization
- Week 6 – Numerical optimization at very large scales
- Week 7 – Midterm exam
- Week 8 – Review of probability
- Week 9 – Introduction to the theory of statistical learning
- Week 10 – Statistical decision theory
- Week 11 – Introduction to model selection
- Week 12 – Cross-validation
- Week 13/14 – Review or bonus topics

About is this course

Evaluation:

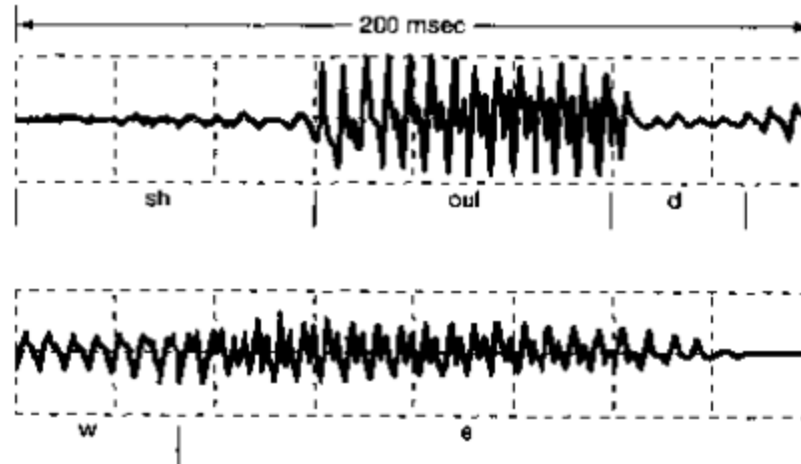
- 20% - 5 homework assignments
- 30% - 1 midterm (concepts + problems)
- 50% - 1 final exam (concepts + problems)

SIGNALS

Why signals

- Digital signal processing is big part of ML
- Much of ML deals with analysis of real world signals – speech, vision, etc.
- Methods from signal processing such as Fourier transform are parts of many ML methods; another example is ***convolution*** in Convolutional Neural Networks

What is signal



0:13



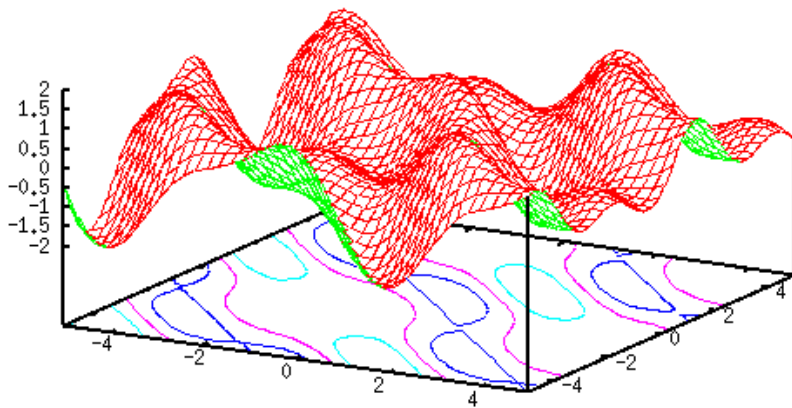
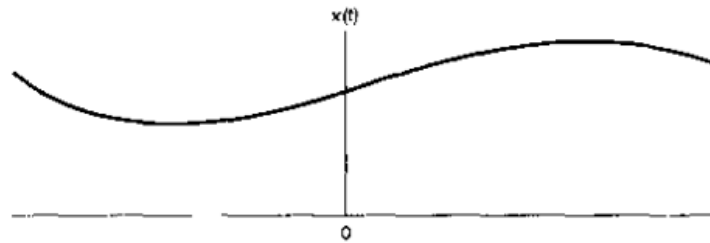
0:14



0:15

Signals abstraction

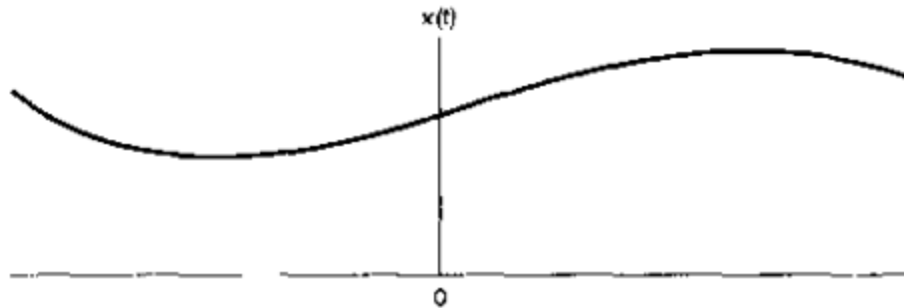
$$x(t)$$



$$f(x, y)$$

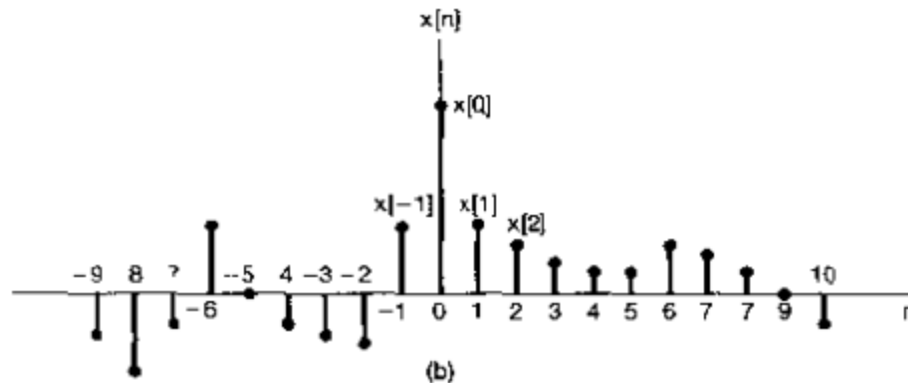
Continuous and discrete

$x(t)$



continuous

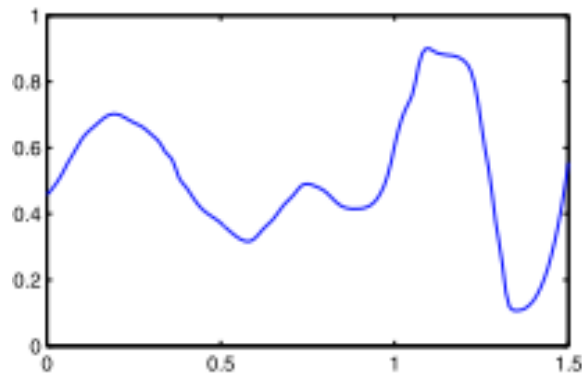
$x(n)$



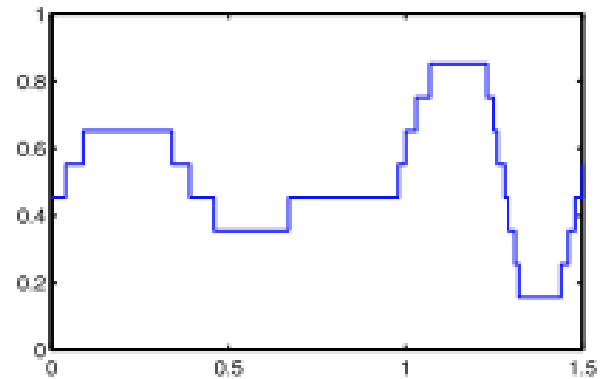
discrete

Analog and digital

Analog

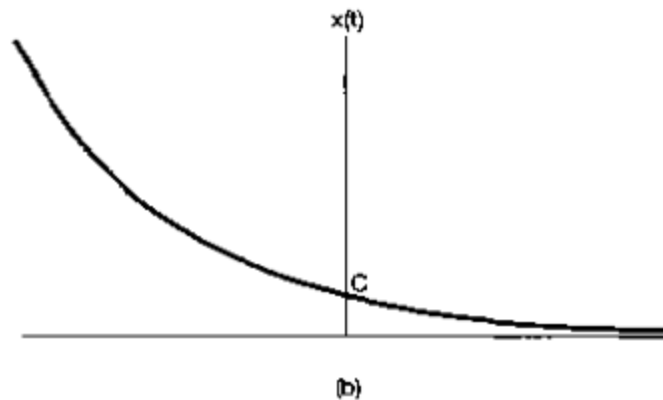
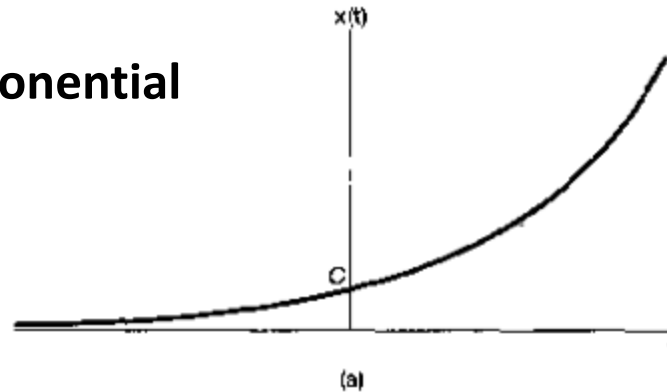


Digital (quantized)



Some basic signals

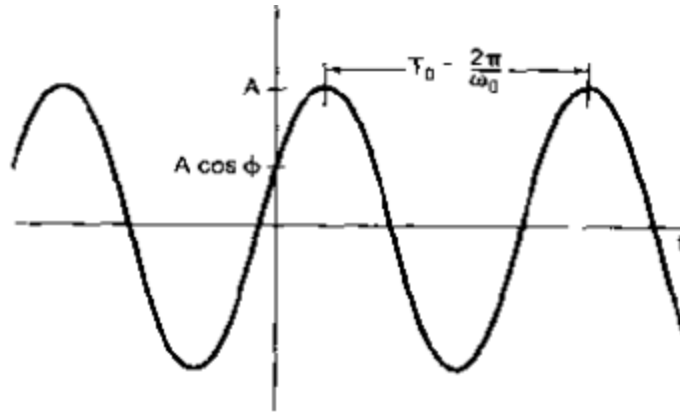
Exponential



$$x(t) = \exp(t)$$

Some basic signals

$$x(t) = A \sin(\omega_0 t + \phi)$$

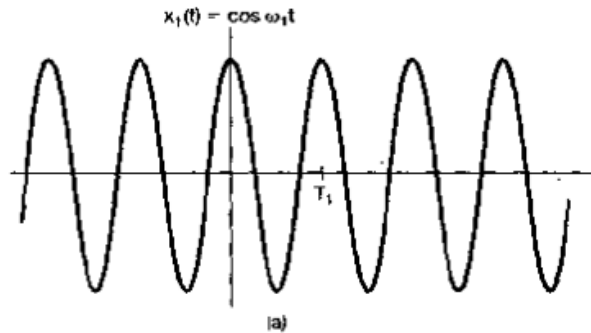


Sinusoidal

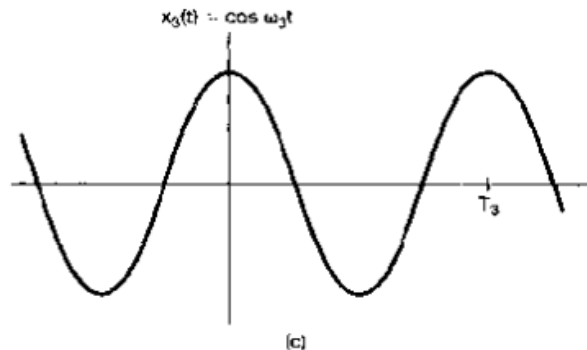
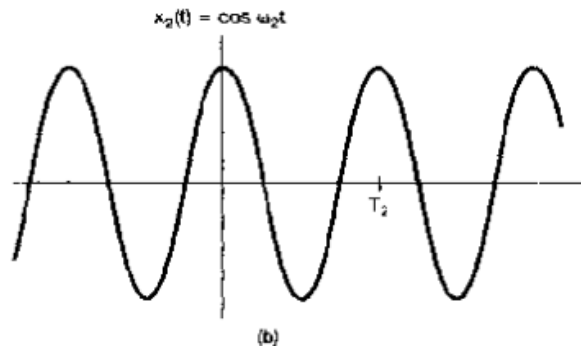
Some basic signals

Frequency parameter in sinusoid controls oscillation speed – the higher the frequency, the faster is the oscillation

Frequency defines how many cycles will happen per unit of time, and is described in Hz – 100 Hz means 100 cycles happen every second



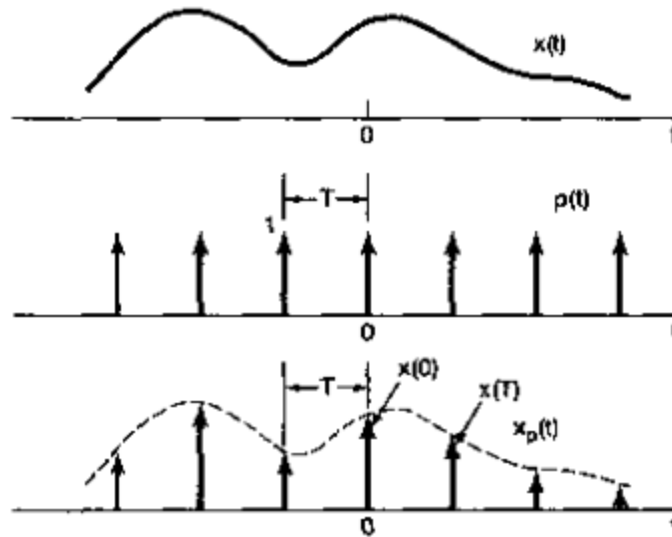
High frequency



Low frequency

Sampling

Sampling is the process of converting a signal into discrete form – computers cannot keep track of continuous signals; for digital processing signal needs to be broken into pieces and stored in step-step form – this is sampling.



Continuous signal (eg electric current in microphone)

Sampling pulse train

Let through values form discrete or sampled signal version of the original at the top

Sampling

Definition: Sampling is the process of converting a *continuous* signal into a *discrete* form

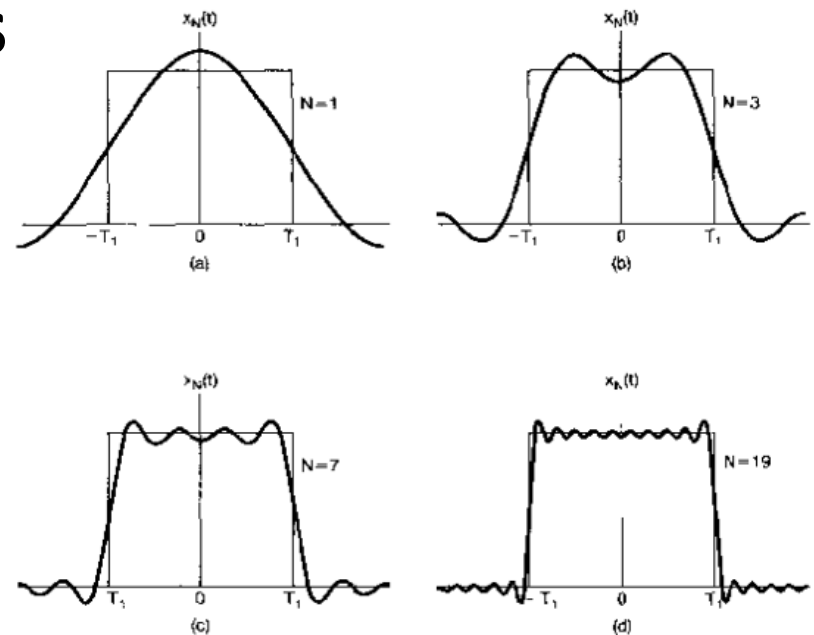
One sample is a single measurement of a signal (usually at an instant or a point)

Sampling

- Sampling frequency or sampling rate F_s is the number of samples recorded every second
- Sampling rate is usually measured as Hz, $F_s=1000$ Hz means 1000 samples of signal are recorded every second
- How many samples will be recorded from signal after 1 s, 10 s, 1000 s ?

The Sampling Theorem

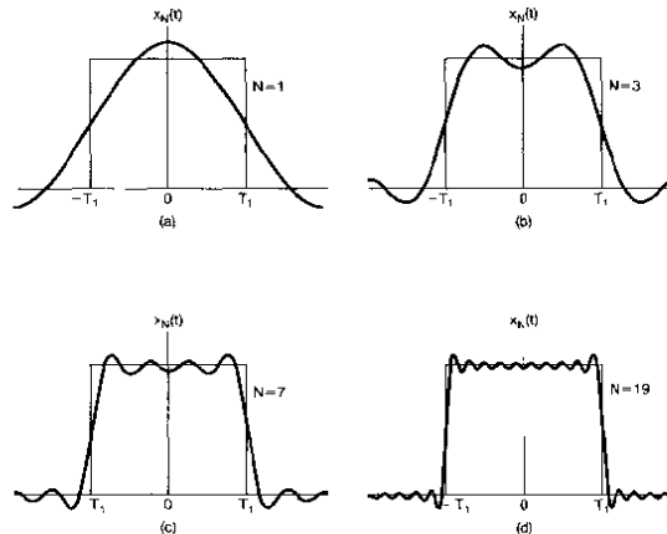
- Problem: play back a signal stored on a computer as a sample.
- **When can this be done without quality losses or noticeable distortions**



The Sampling Theorem

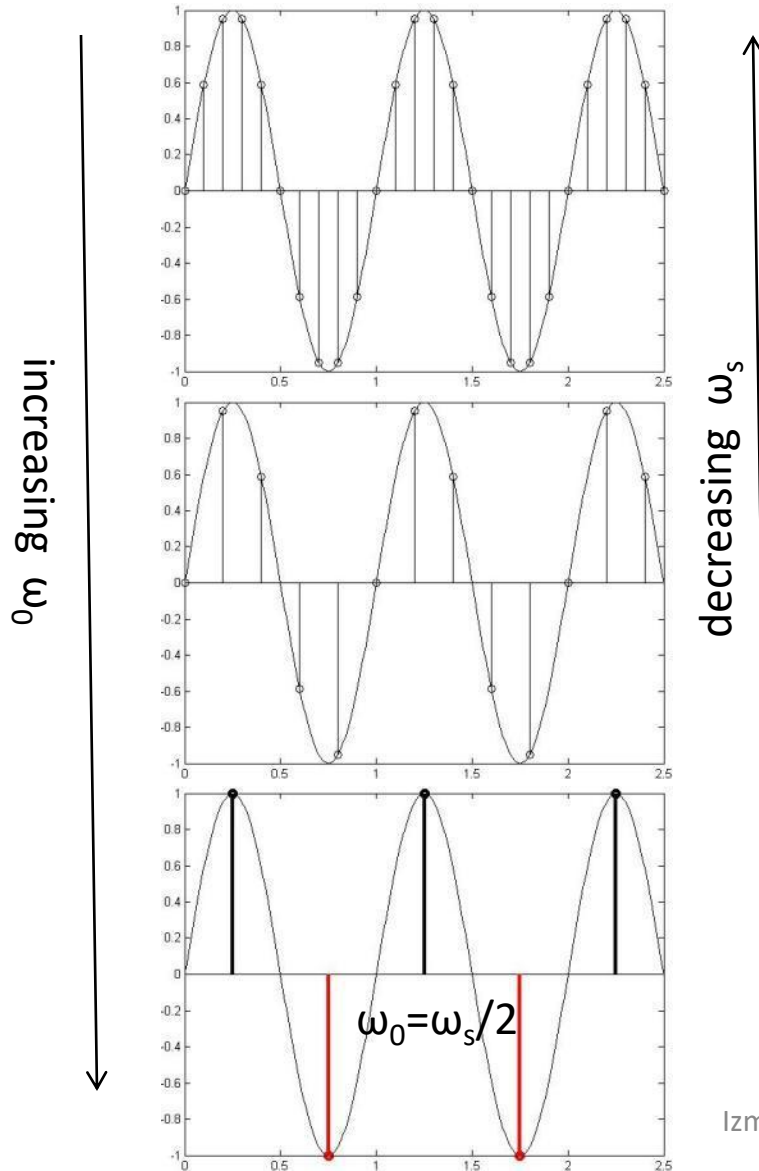
- Frequency-picture of signals = make up signals by adding together sinusoids of different frequencies, amplitudes, and phases

Making up a square
from sinusoids:

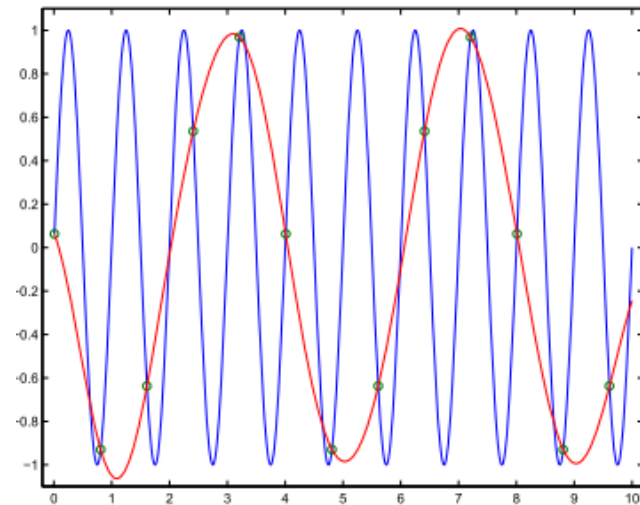


The Sampling Theorem

When a sinusoid can be got back correctly from only its samples?



$$\omega_0 > \omega_s/2$$



When sampling rate is smaller than twice the signal's frequency, more than one sinusoid fits the same samples !!!

The Sampling Theorem

Sampling Theorem: A signal containing in it sinusoids of a maximum frequency F_0 can be completely reconstructed from its samples only if the sampling rate is greater than twice the maximum frequency, $F_s > 2F_0$

The Sampling Theorem

Nyquist rate is the minimum rate at which a signal can be sampled without reconstruction errors

(By the Sampling Theorem,
 $F_{\text{Nyquist}} = 2F_{\text{signal,max}}$)

The Sampling Theorem

What happens to the signal's frequencies that above the Nyquist limit?

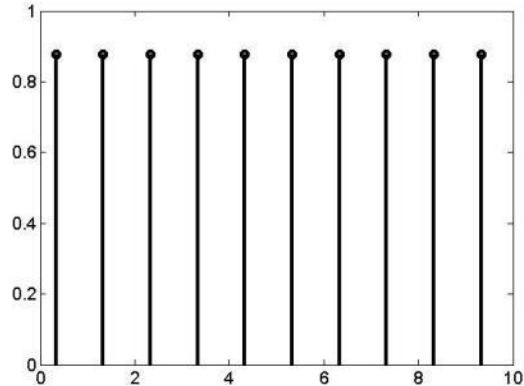
- For frequencies in signal such that $F > F_s/2$, two possibilities exist that equally well match every series of samples: $F_s - F$ and F (see slide 32). The frequency $F_s - F$ is called **alias of F** , and for $F > F_s/2$ the alias has a lower frequency than the original signal, $F_s - F < F$. During signal play-back (reconstruction), the alias gets reconstructed instead of F and the signal becomes corrupted!

The Sampling Theorem

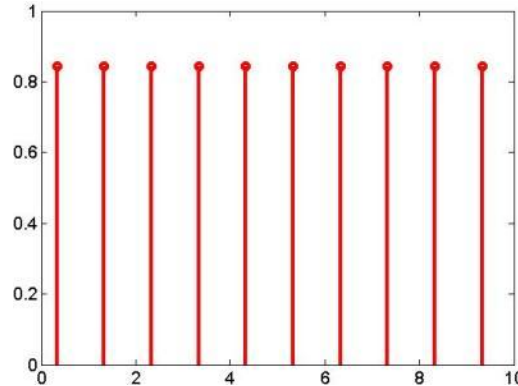
How is this important for ML:

- Example: detect tones in audio recording with frequencies 10kHz and 20kHz, if the audio was recorded by hardware at $F_s=10\text{kHz}$ sampling rate?
- Example: use a security video camera with effective pixel size of 10cm for face recognition that uses face details at 1-2cm scale?

The Sampling Theorem



10kHz signal @ 10kHz sampl. rate,
equivalent to alias 0 Hz !



20kHz signal @ 10kHz sampl. rate,
equivalent to alias 0 Hz !

Can we use ML
to distinguish
between these
signals?



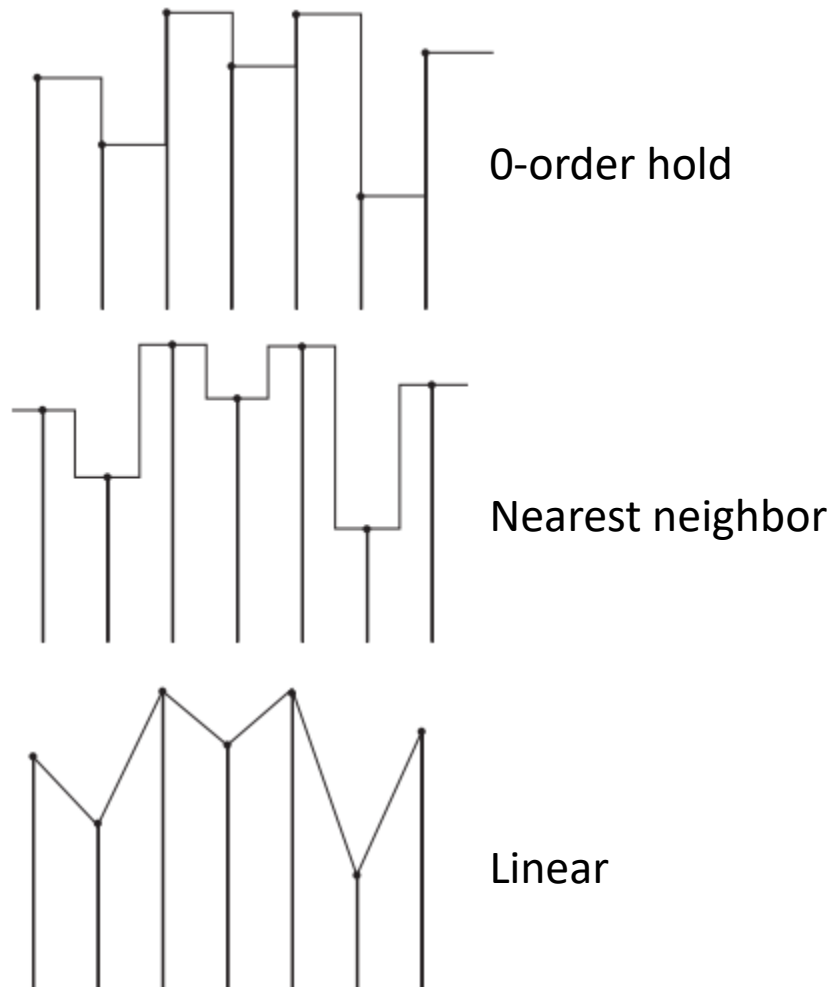
Can we use ML to
ID this person?

Signal reconstruction

- **Problem:** Assuming Sampling Theorem and Nyquist limit is OK, how can we replay a sampled signal back?
- Related questions:
 - Increase sampling rate of an already sampled and recorded signal digitally (**upsampling**).
 - Decrease sampling rate of an already sampled and recorded signal digitally (**downsampling**).
 - Produce new samples of same signal at different time points, digitally (**resampling**).

Signal reconstruction

Interpolation



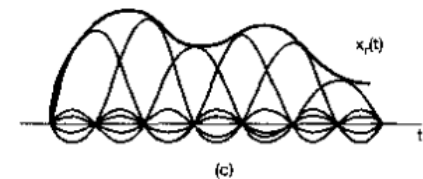
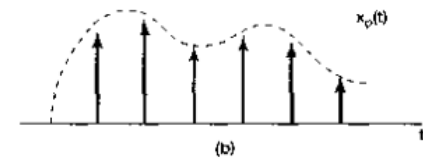
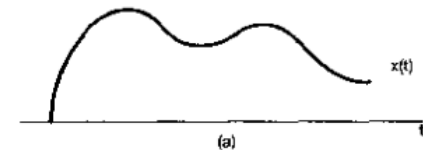
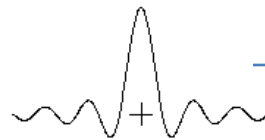
Simple signal interpolation:
Calculate the values of
signal $x(t)$ at any times by
using interpolation
algorithms such as 0-order
hold, nearest neighbor or
linear interpolation

Signal reconstruction

Exact signal reconstruction can be performed by **ideal low-pass filter** or **sinc(t)** function

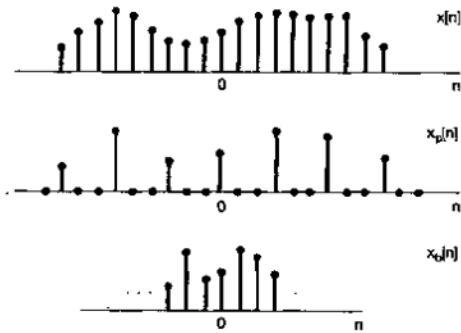
$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) \text{sinc}(t - nT)$$

$$\text{sinc}(t) = \frac{\omega_0 T \sin(\omega_0 t)}{\pi \omega_0 t}$$

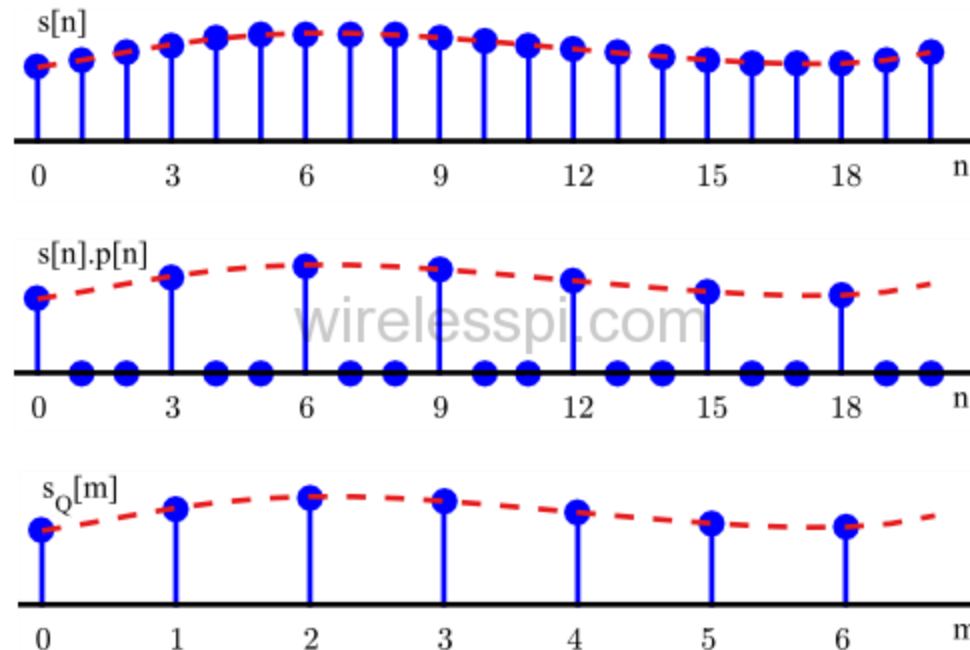


Sinc function is put onto each sample, multiplied by that sample height, and added all together

Signal resampling

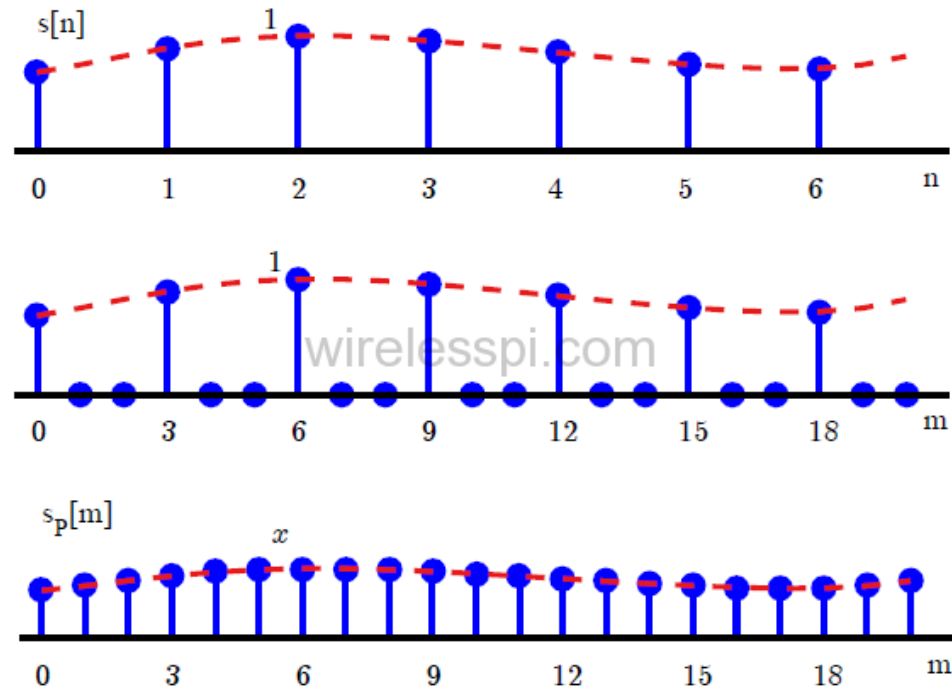


Downsampling by **decimation** is removing every $K-1$ out of K^{th} samples from the signal – the resulting signal has sampling frequency F_s/K and K -times fewer samples !



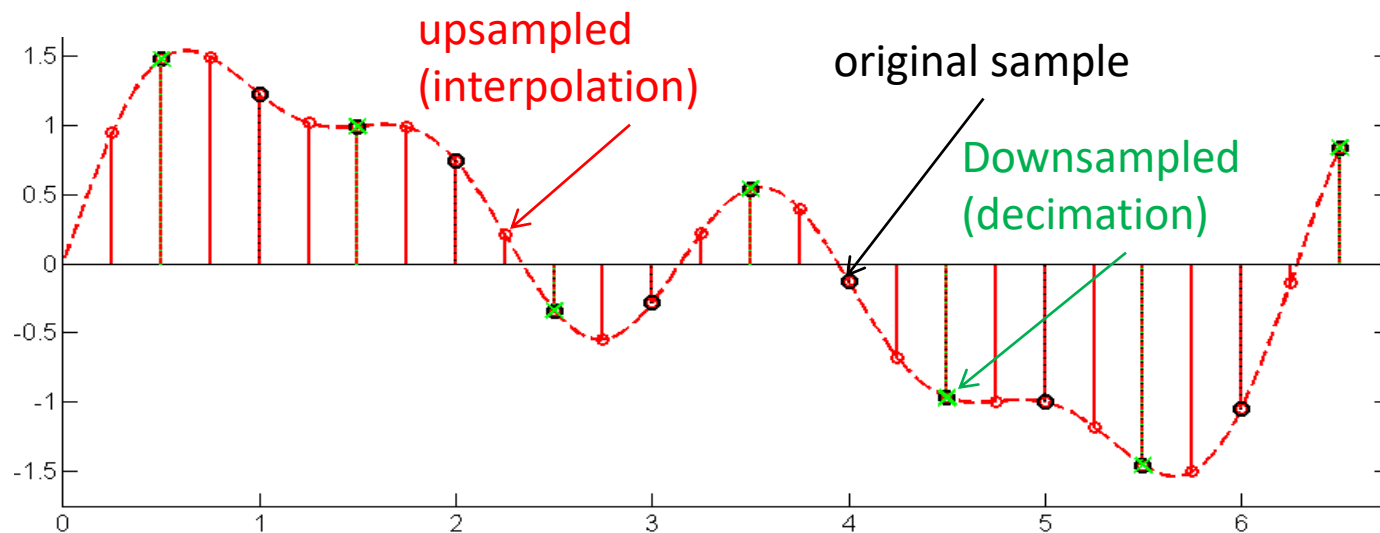
Signal resampling

Upsampling involves adding new samples in between existing samples, calculated digitally via one of the earlier reconstruction methods (simple interpolation or ideal low-pass filter)



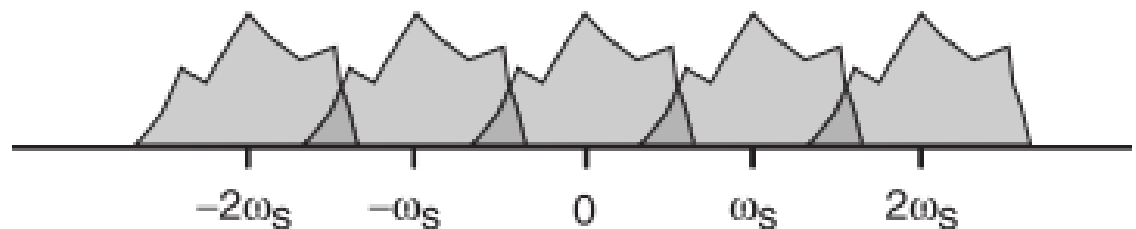
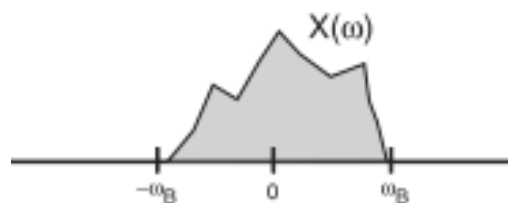
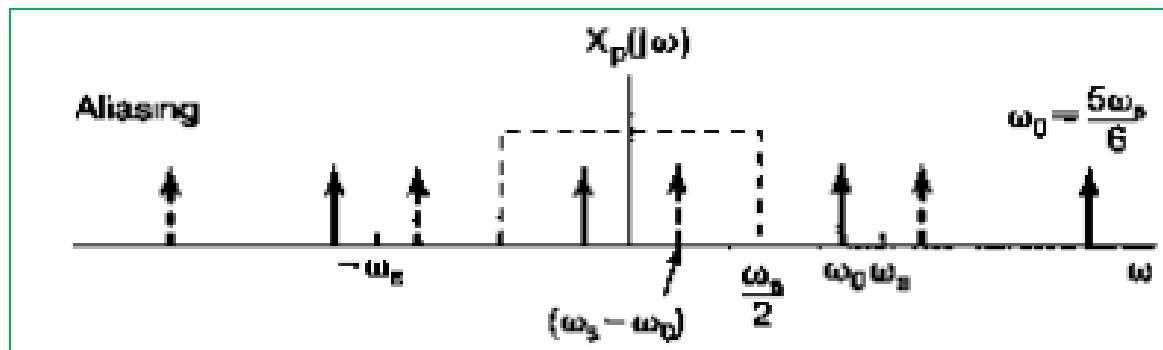
Signal resampling

- Downsampling – reducing the number of samples for example by decimation $x(n) \rightarrow x(2n)$
- Upsampling – increasing the number of samples eg $x(n) \rightarrow x(n/3)$



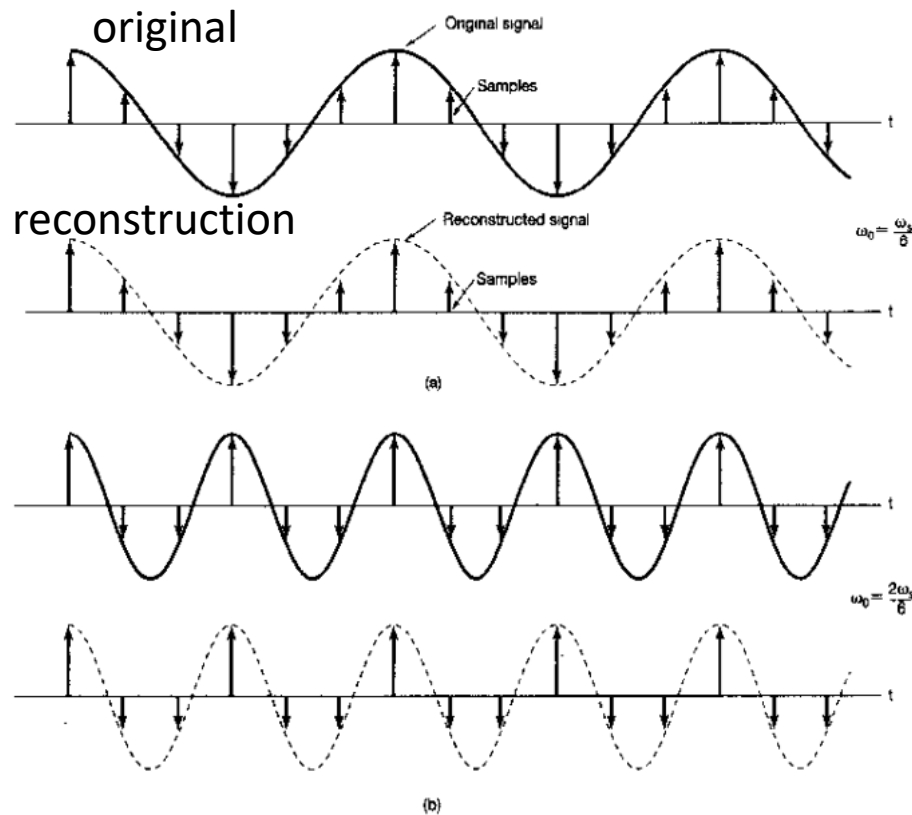
Signal aliasing

If signal has in it sinusoids with frequencies above $F_s/2$, those map during reconstruction onto their low-frequency aliases (see slide 35) creating “ghost” copies of high-frequency part of signal’s spectrum which can significantly distort the reconstruction result.

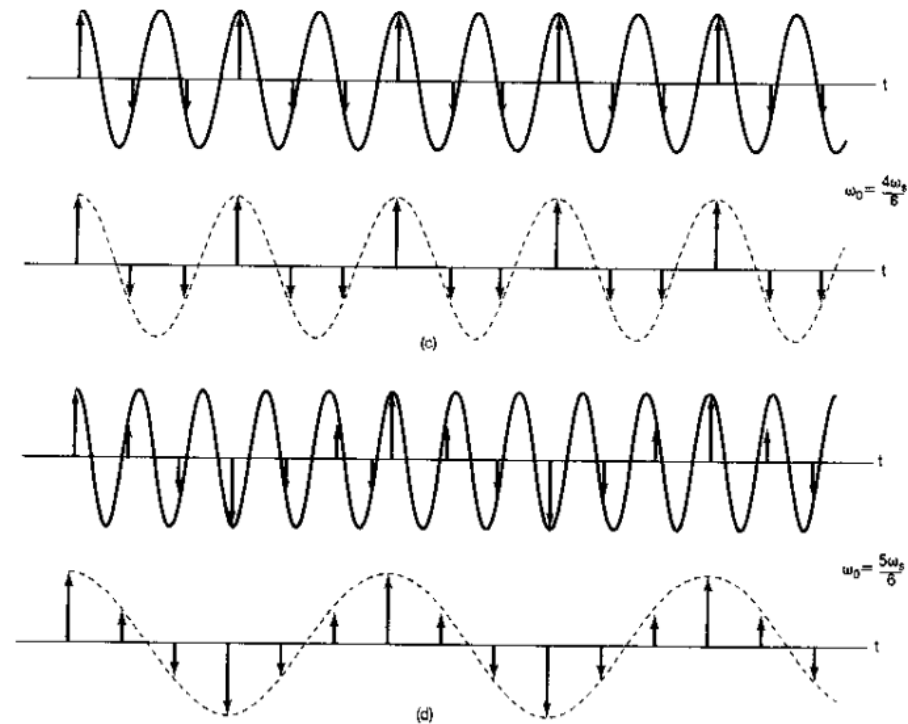


Signal aliasing

Frequency below 1/2 sampling rate (Nyquist limit) – no problems



Frequency above 1/2 sampling rate (Nyquist limit) – maps onto lower-frequency aliases !!



Signal aliasing

Aliasing may become a problem when images are rescaled.

- An image of size X pixel can be said to have sampling rate of $F_s=X$
- The highest (here spatial, in space) frequency present in such an image then will be $F_s/2$, which corresponds to a pattern of alternating black and white pixels
- When image is down-sized to a smaller new size Y , one can consider this as downsampling the image at lower rate $F_s=Y$
- The patterns in the image with spatial scales between 2 and $2Y/X$ pixels (that is frequencies between $Y/2$ and $X/2$) will form aliases distortion in the image by mapping onto low-frequency false textures in the image of size Y

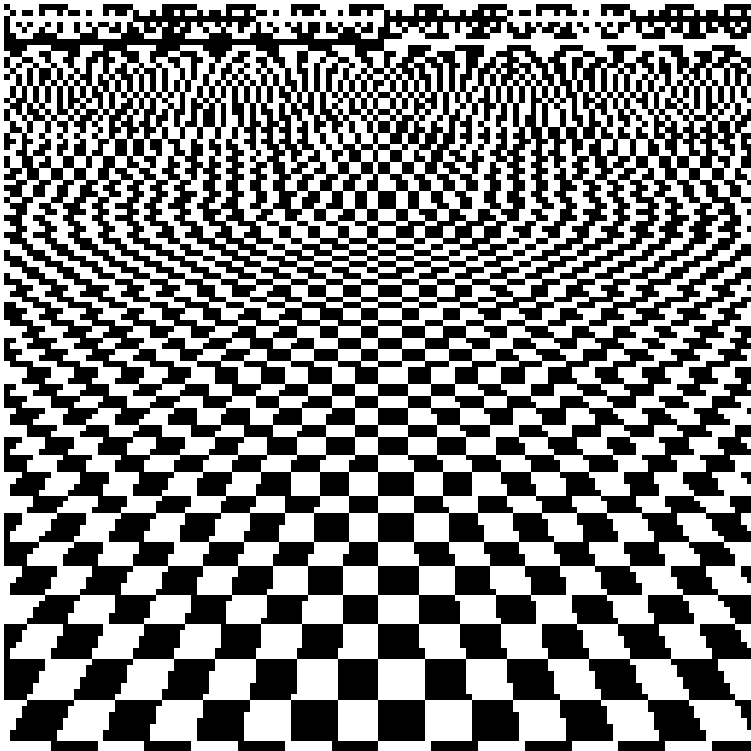
Signal aliasing

For example, consider an image of size $X=2000$ pixels ($F_s=2000\text{dpi}$) having a texture (dark-light pattern) with scale 2.1 pixels ($F_p=950\text{dpi}$).

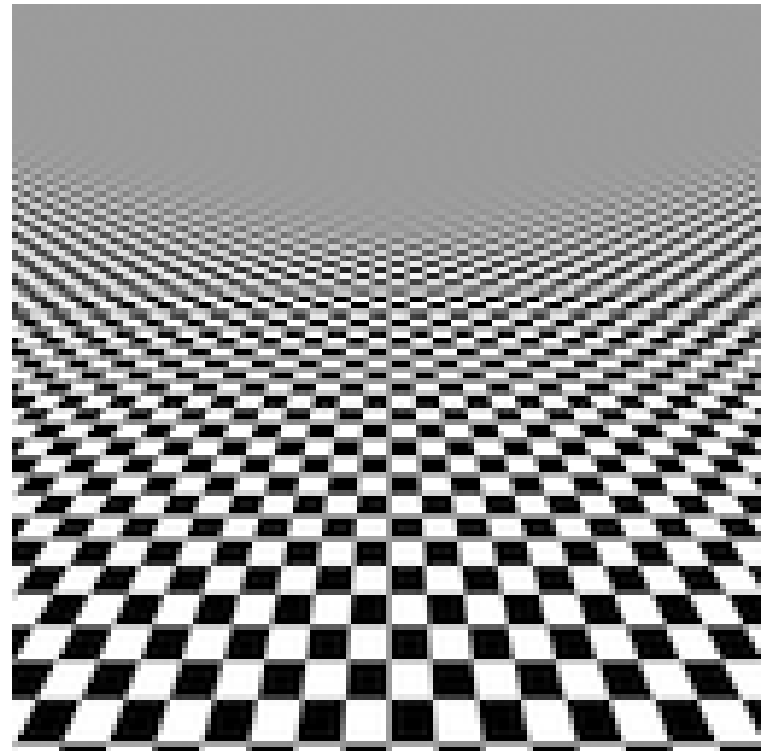
Suppose we resize that image to new size $Y=1000$ pixels ($F_s=1000\text{dpi}$). That texture now is above the $F_s/2$ threshold and will form low-frequency alias with frequency $F_s - F_p = 50\text{dpi}$, which will be perceived by the eye as a not-seen-before dark-light pattern of spatial scale 20 pixels (frequency 50dpi).

Signal aliasing

Aliases formed
during down-sizing:

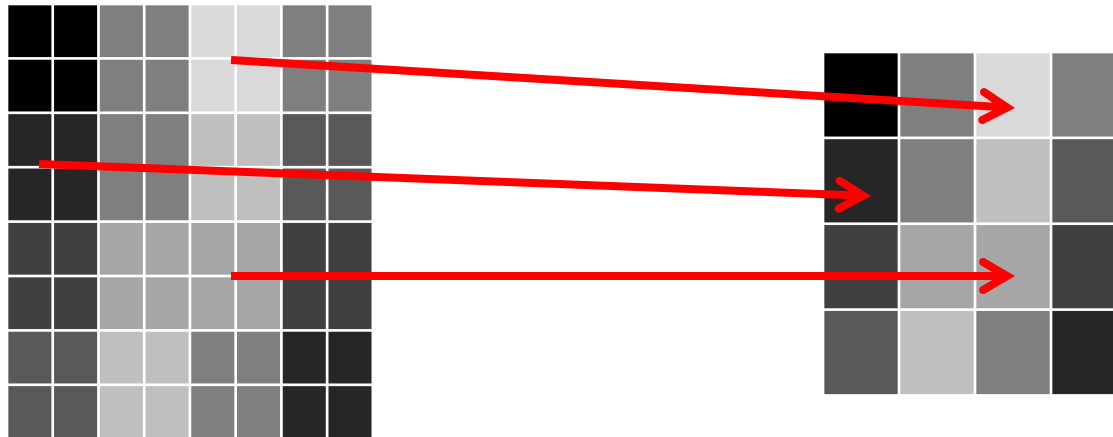


What it should look like
instead:



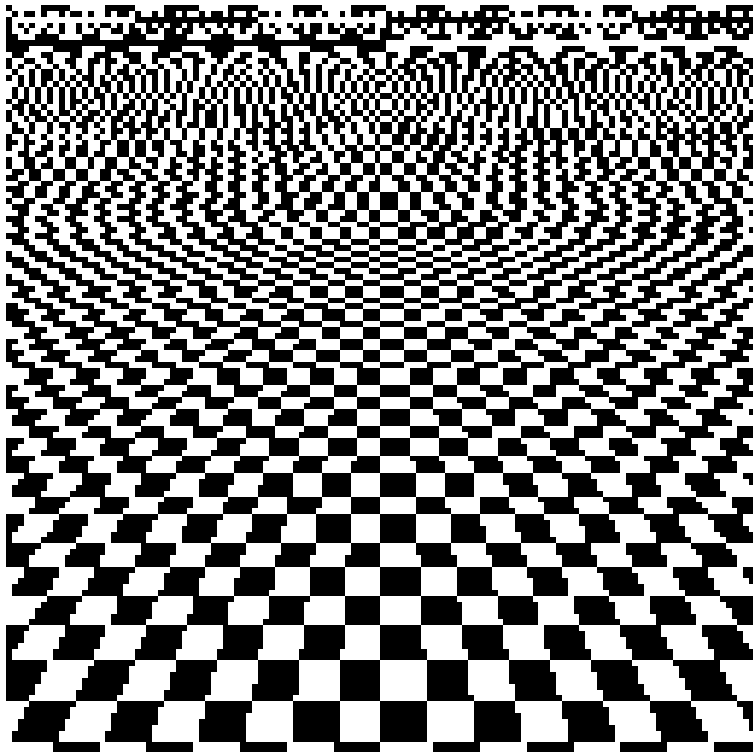
Signal aliasing

Anti-aliasing refers to removing potentially troubling high-frequency patterns from image before down-sizing, generally by means of smoothing such as replacing every $K \times K$ block of image with uniform average value, and then resizing !

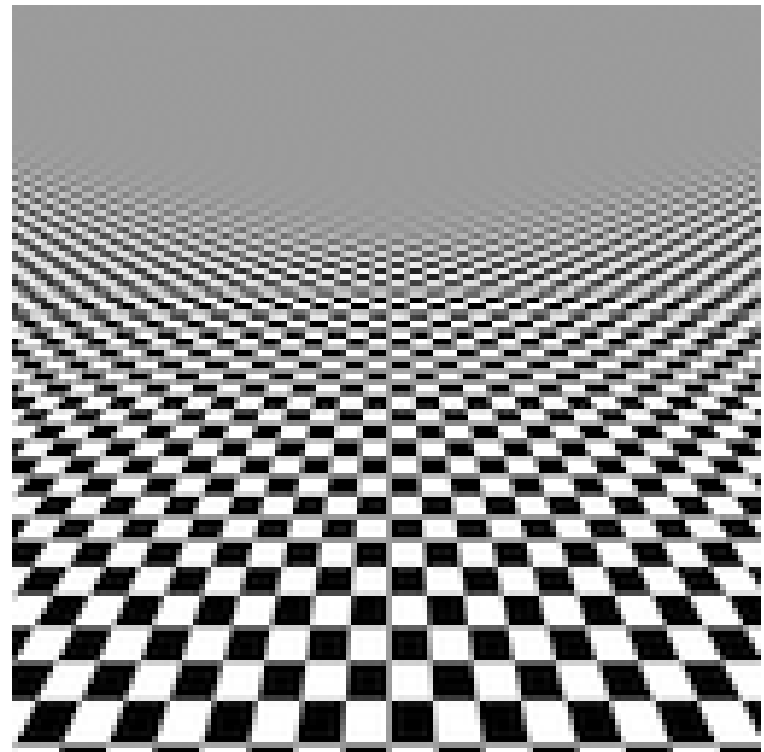


Signal aliasing

Aliased low-res image



Anti-aliased by smoothing



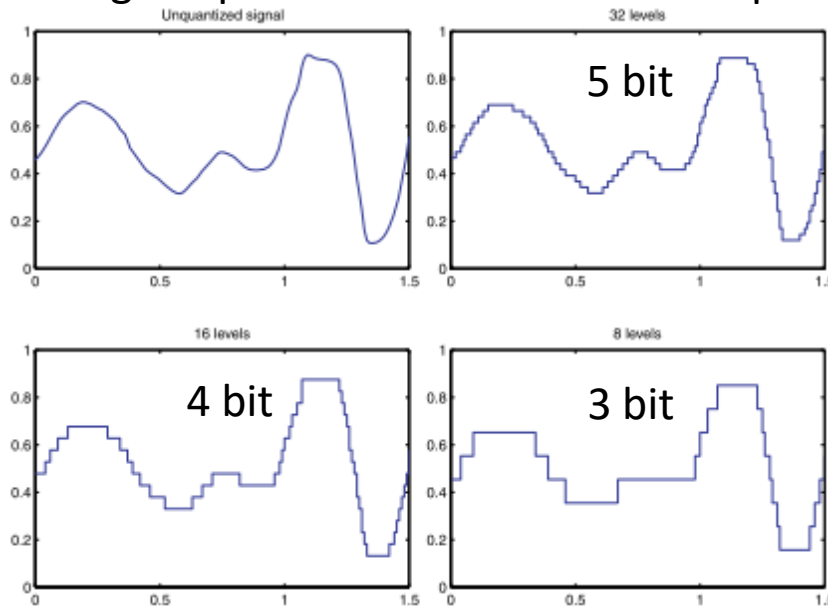
Signal aliasing

- You might have seen anti-aliasing term in computer games: in games original high-res graphics often needs to be resized to lower user screen's resolution – this can cause aliasing as discussed above
- Textures with spatial frequencies close to the size of the down-sized image may create aliases significantly distorting the resulting game graphics

Digital signal quantization

- **Quantization** is the process of transforming an analog signal, represented by continuous numbers, into digital form, represented by binary integers
- Main property of quantization is **bit depth**

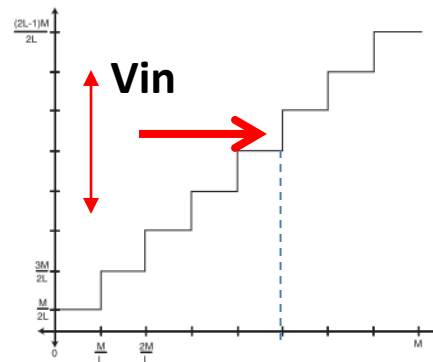
A signal quantized to different bit depth:



$$N_{\text{levels}} = 2^{\{\text{bit depth}\}}$$

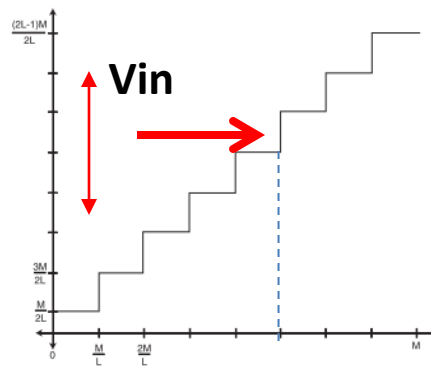
Digital signal quantization

How quantization works: When a continuous signal arrives, its instantaneous value is compared with a set of predefined levels and the largest level smaller than the signal's value is output instead. Thus, for example, incoming signal values $V_{in}=1.555$ and $V_{in}=1.524$ may produce the same quantized output value of 1.5, and the signal value $V_{in}=1.677$ will produce quantized output of 1.6.



Digital signal quantization

- If the series of used quantization levels is uniformly spaced (for example $V_{out} = 1.0, 1.1, 1.2, 1.3, 1.4, \dots$), such quantization is called uniform
- For uniform quantization we can define $V_{out} = \text{floor}(V_{in}/\Delta V)$, where ΔV is the level height also often called quantization **resolution**



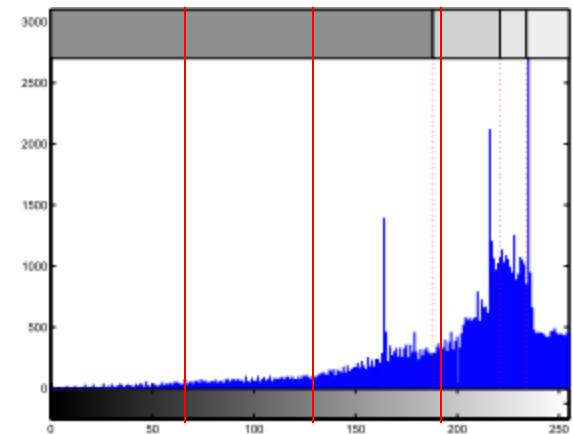
Digital signal quantization

Uniform quantization is the simplest but may not be the best – if signal is extremely nonuniformly distributed, binning it into uniform-width levels may result in a lot of wasted bits!

2 bit (4-levels) uniform quantization of a coin image



Histogram of signal values



These levels are basically wasted –
contain almost no pixels !

Digital signal quantization

Histogram equalization is a quantization method where levels are chosen such that after the quantization each level contains *about the same number of total signal samples*

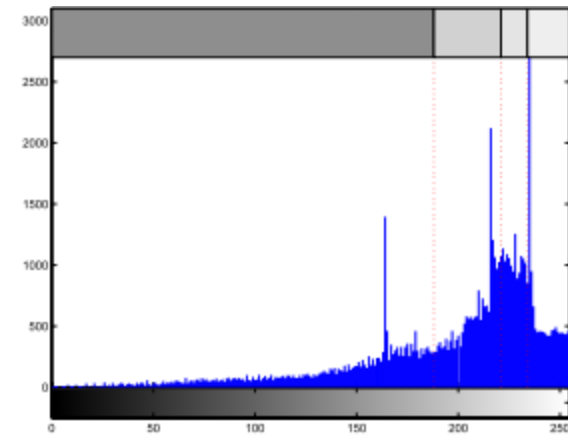
Uniform



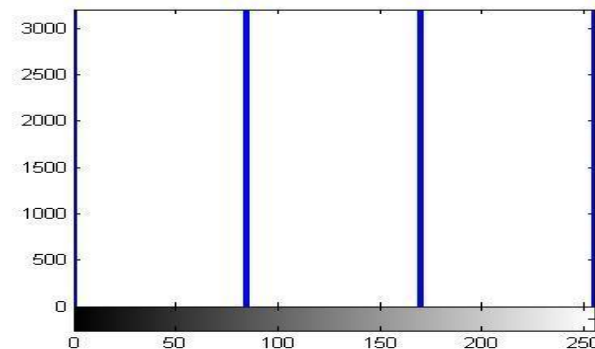
Hist-equalized



Original image histogram



New 2-bit image histogram



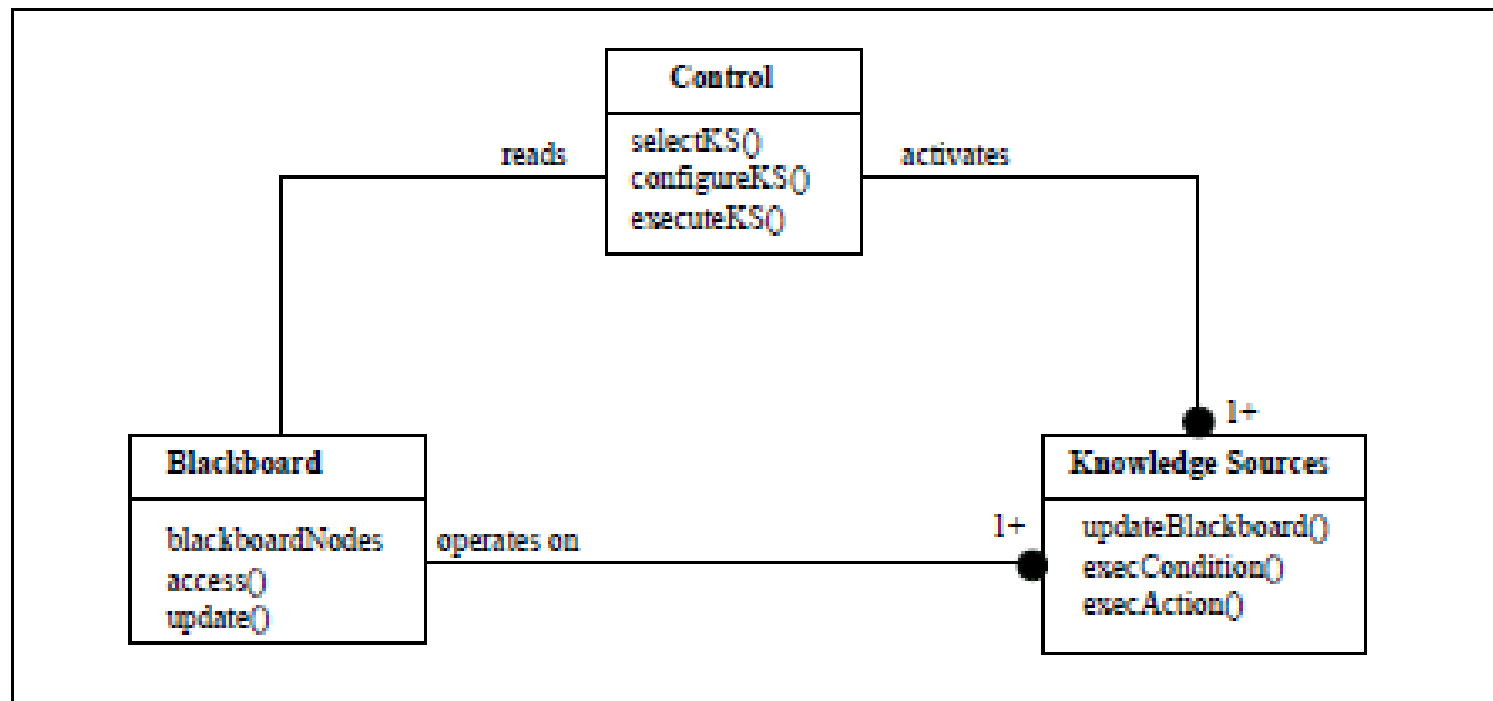
QUESTIONS FOR SELF-CONTROL

- What does *learning* in “Machine Learning” mean?
- In object recognition in image processing, what is x and what is y ?
- How AI and ML are different?
- What are the types and the elements of human intelligence, as we understand it today?
- What are the main types of different signals?
- What is the mathematical abstraction of a signal?
- What effect does changing frequency in a sinusoidal signal have?
- Explain sampling and sampling rate.
- How many samples a 60 seconds audio signal recorded at 44KHz will have?
- What is the statement of the Sampling Theorem?
- What is Nyquist rate and Nyquist limit?
- If you want to distinguish two audio tones on a computer, one having frequency 10KHz and one 20KHz, what minimal sampling rate the recording equipment must have to allow you to do that?

- If two audio signals differ in audio frequency at 10KHz (one is loud and one is not), can such signals be distinguished when recorded at sampling rate of 15KHz? What is the minimal sampling frequency needed to be able to distinguish those signals?
- How can one up- and down-sample a signal by using interpolation and decimation?
- What is frequency alias?
- Why is aliasing a problem when downsizing images?
- Explain what digital signal quantization is.
- How many possible levels a signal quantized at 6 bits has?
- How much space a 60-second audio signal recorded at 44KHz and quantized at 16 bit will require to be stored in a computer?
- What is uniform quantization?
- What is histogram equalization?

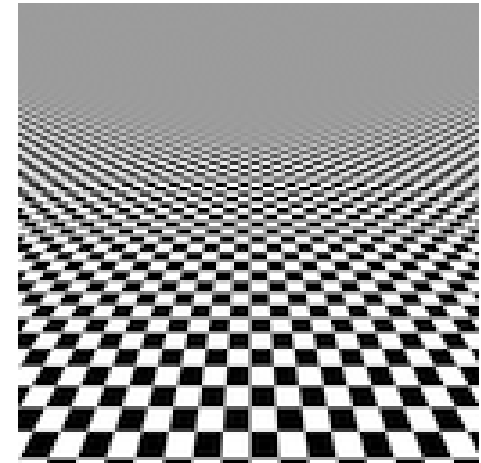
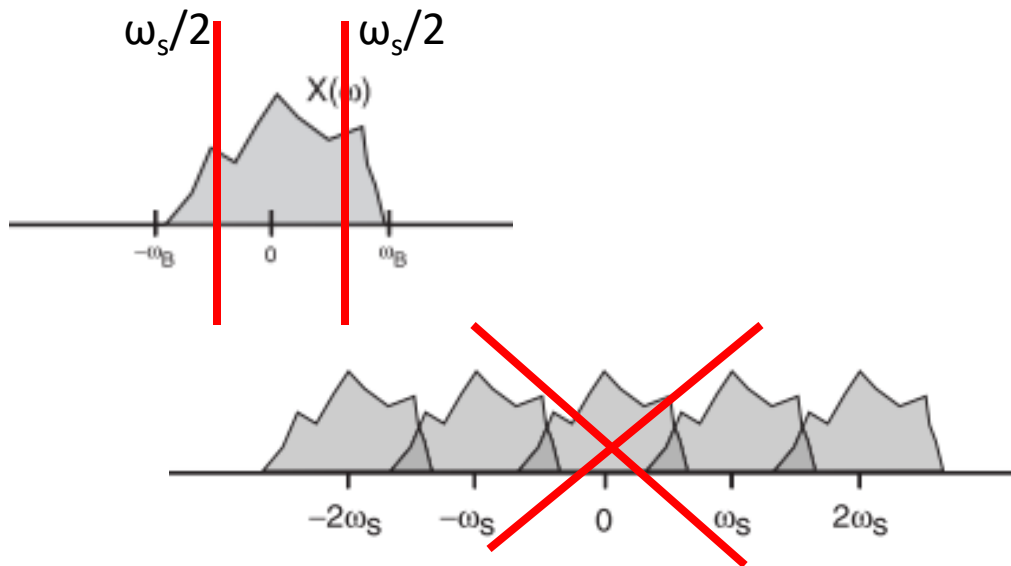
ADVANCED

Blackboard software design pattern



Aliasing & anti-aliasing

Anti-aliasing involves filtering out frequencies above a downsampling-defined threshold before downsampling, preventing aliasing-caused leakage of spectrum from forming



Aliasing in applications

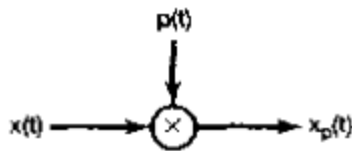
- In digital applications, aliasing appears during downsampling of digitally recorded signals and images
- The higher sampling rate version of a signal can contain frequencies above the Nyquist limit of the downsampled signal, thus producing aliases in the downsampled final signal
- Aliasing is commonly encountered in computer games: high-res original graphic art is resized to match end user's screen resolution thus forming aliases

Aliasing in applications

- In ML, we may encounter aliasing when we need to reduce resolution of an incoming data or image because applying ML algorithm on the original data is prohibitive because of high computing time
- Another widely encountered problem is the multi-scale analysis, which means that ML algorithm that is sensitive to a feature at only a single scale (eg a dark line of size 10 cm) needs to be applied to differently re-scaled image to detect different-scale versions of the same feature (for example detect eyes of variable width 1 cm to 100 cm)

General sampling abstraction

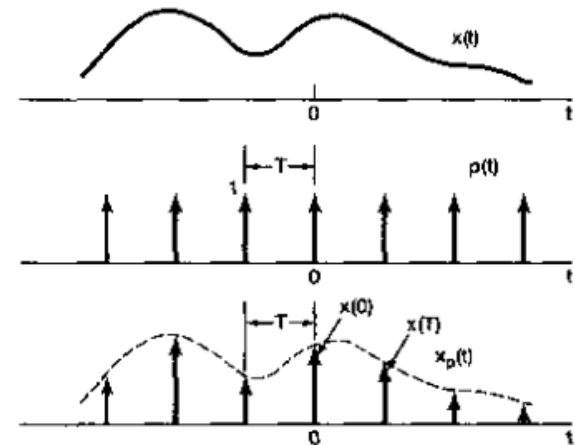
- Impulse-train sampling



delta function

$$x(n) = \sum_t x(t) \delta(t - nT)$$

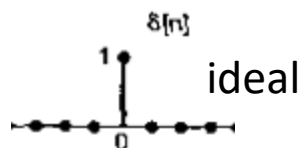
convolution



- General sampling

a sampling function $f(t)$

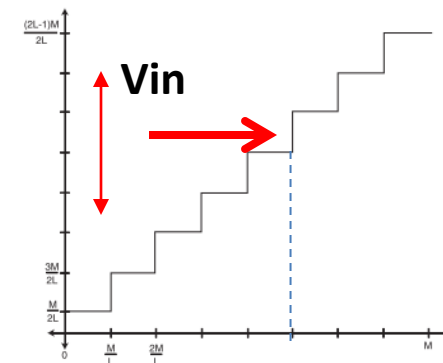
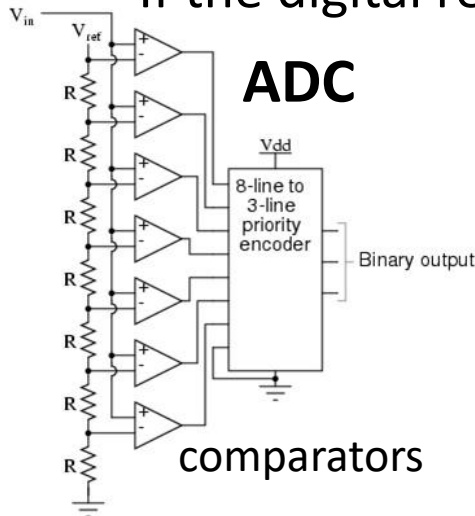
$$x(n) = \sum_t x(t) f(t - nT) = x(t) * f(t)$$



Analog to Digital Converters

Workings of an Analog-to-Digital converter:

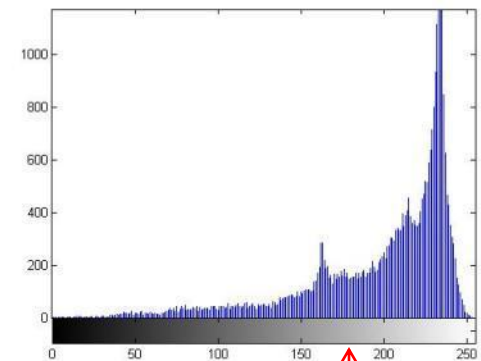
- A chain of electronic comparators is set up with uniformly growing reference voltage (due to the series resistors below) when moving up the chain
- Incoming signal V_0 triggers the lower sub-set of comparators where $V_i < V_0$
- Thus formed linear integer signal is converted into binary integer signal by logical decoder, thus forming a digital representation of incoming signal for a computer
- If the digital representation has b bits there are 2^b possible levels



Otsu's thresholding

BW thresholding can be viewed as a 2-level quantization problem for two Background and Foreground levels. A popular solution for this is Otsu's threshold (eg graythresh in Matlab), which aims to maximize the relative uniformity of the foreground and background image pixel classes

$$\max N_B N_W (m_B - m_W)^2$$



Otsu: 180

Information Theory

Definition: For a signal with a finite alphabet Σ the (Shannon) information H is defined as

$$H = -\sum_{a \in \Sigma} p(a) \log_2 p(a)$$

where $p(a)$ are the frequencies of occurrence of each symbol a from Σ in signal

$$\Sigma = \{a, b, c\}:$$

$$s(t) = "abbababababaa"$$

Information Theory

$$H = -\sum_a p(a) \log_2 p(a)$$

$$s(t) = "0101101001001101" \quad p = [0:0.5, 1:0.5]$$

$$H = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1 \text{ bit per symbol}$$

$$s(t) = "00000000100000100" \quad p = [0:0.9, 1:0.1]$$

$$H = -0.9 \log_2 0.9 - 0.1 \log_2 0.1 = 0.47 \text{ bit per symbol}$$

Information Theory

- Shannon information quantifies the variability of a signal having certain statistical properties (that is the probabilities of observing a symbol $p(a)$) – for that reason Shannon information is also sometimes called **entropy**
- Any signal with Shannon information H can be encoded using binary code of *average length* H
- A signal with Shannon information H cannot be encoded by any binary code to average length smaller than H (it is impossible – a statement in **Information Theory**)

Information Theory

$$H = -\sum_a p(a) \log_2 p(a)$$

$$|\Sigma| = N \Rightarrow H \leq \log_2 N$$

$s(t) = "0101101001001101"$

$H = 1$ bit per symbol

$s(t) = "00000000100000100"$

$H = 0.47$ bit per symbol


This signal can be re-encoded at 50% of its original length \Rightarrow **compressed !**



Information Theory

$H = 0.47$ bit per symbol

$s(t) = "00000000100000100"$



00	0
01	10
10	110
11	111

$\rightarrow s(t) = "0001000100"$

Block-based code of variable length – re-encode every sequence of 2 bits into new code of change of **variable length** from 0 to 3, so that the most frequently seen sequence has new code of length of only 1 ("00" \rightarrow 0)

(VARIABLE LENGTH CODE)

$s(t) = "0001000100"$

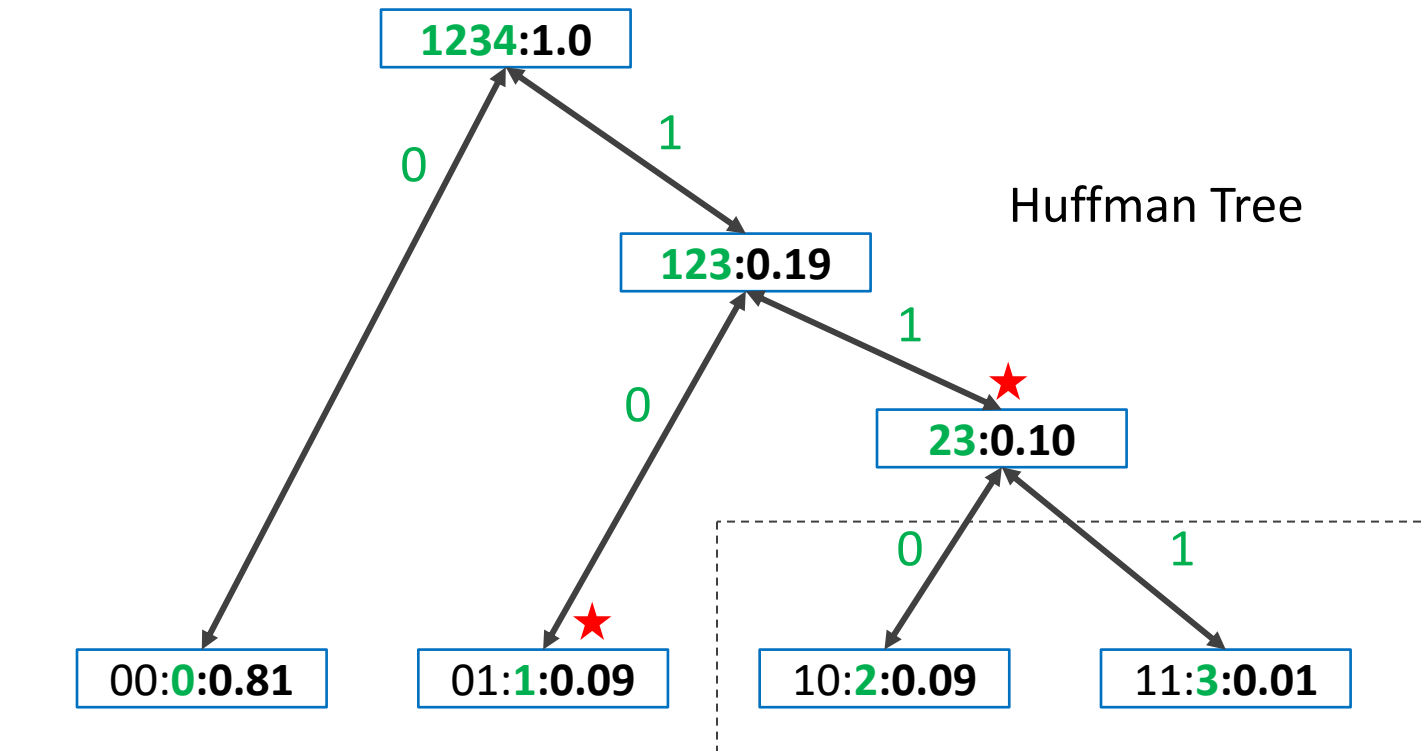
New length:

$$\bar{L} = 1/2 \cdot (0.9^2 \cdot 1 + 0.09 \cdot 2 + 0.09 \cdot 3 + 0.01 \cdot 3) = \underline{0.645}$$

Information Theory

Huffman code is a variable length binary code with the shortest average length for given alphabet (optimal variable length code)

00	0
01	10
10	110
11	111

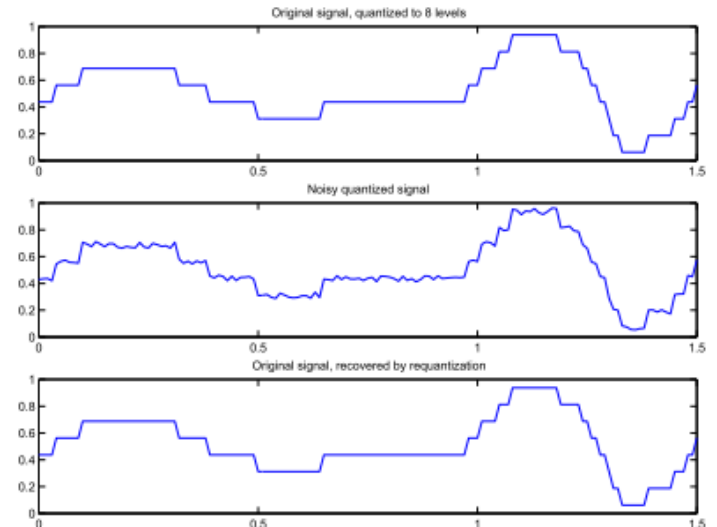
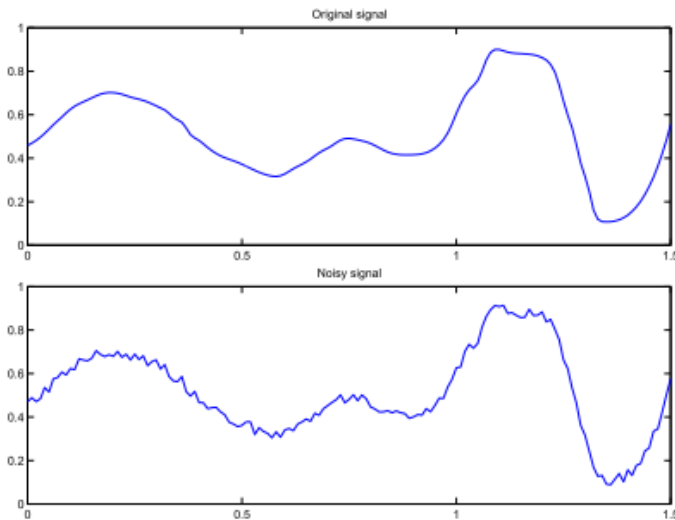


Information Theory

- If Shannon information of a binary signal H is smaller than 1 bps, such signal can be compressed all way down to H
- Huffman code produces a compressed signal with average length strictly greater than H
- Degree of compression in Huffman code can be increased by grouping bits in signal into larger blocks, but this results in a larger **coding table** also
- In the above example (90% 0s and 10% 1s), 3 bit blocks can achieve Huffman compression ratio of 0.533 – much closer to the limit 0.47 than 0.645 with 2 bit blocks

Noise in signals

- Digital signals are less susceptible to noise than analog signals because noise need to exceed one level size to change a digital signal, unlike analog where any noise damages the signal
- Noise is characterized by variance σ^2 – the mean square difference of the noise-corrupted and the original signals

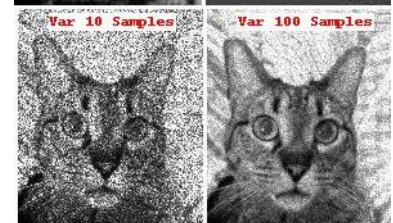
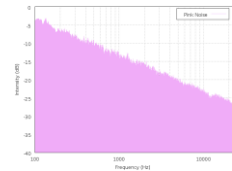
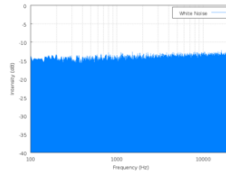
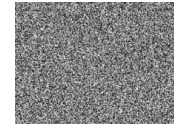


$$\sigma^2 = \left\langle (S_{noisy} - S_{true})^2 \right\rangle$$

Noise in signals

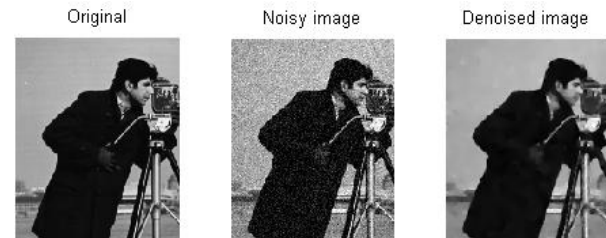
Different types of noise:

- White noise (Gaussian noise or uniform spectral power noise)
- Poisson or shot noise (photon count-related noise)
- Pink noise (more noise at low than high frequencies)
- Salt-and-pepper noise (pixel-based image change)
- Blur (sharpness loss in images)
- Aberration noise (structural change in images)



Noise in signals

- Any noise destroys information, and such information generally can no longer be recovered from signal
- Denoising utilizes prior knowledge about a signal or image to try and restore the information lost to noise



Noise in signals

- Denoising has to be a guess because there are always many possibilities of how image could look like before noise corrupted it (such as many sharp images would match same blurred image). The prior information is used to choose among those possibilities.

