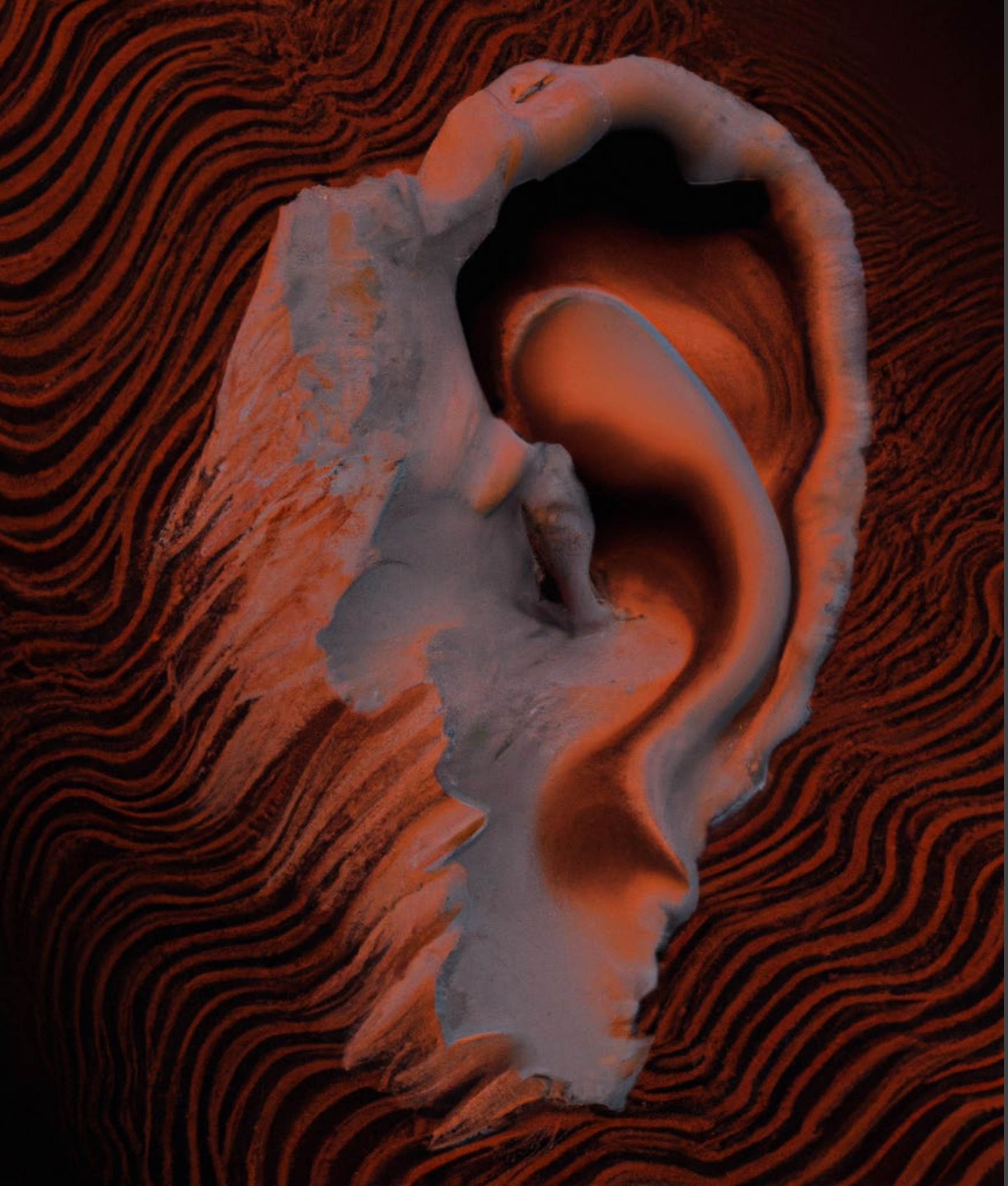


1D Signals and Audio

Applied Machine Learning
Derek Hoiem

This class: Audio and 1D signals

- Representing sound with frequencies
- Deep networks for audio
- Other 1D time series

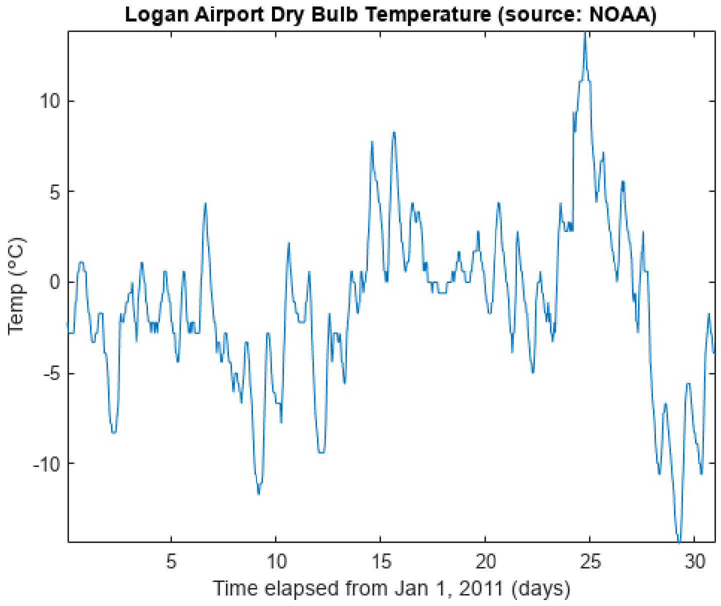
1D time series problems are common

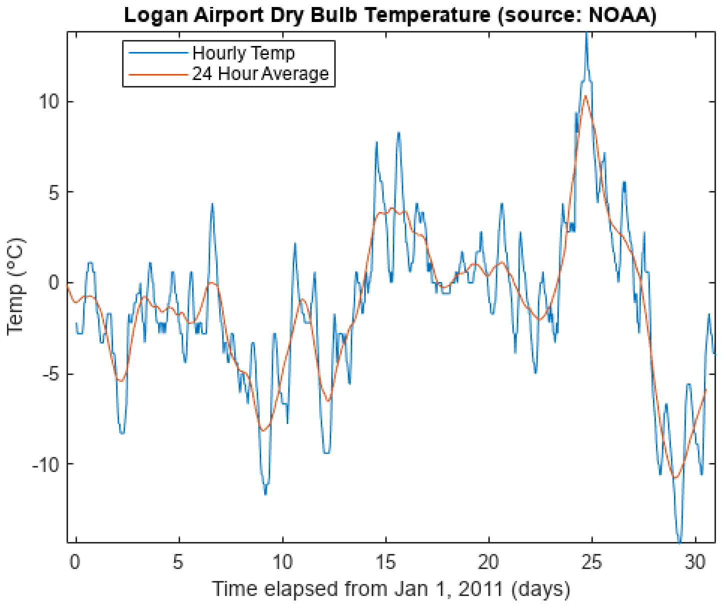
- Stock prices, vibrations, popularity trends, temperature, …

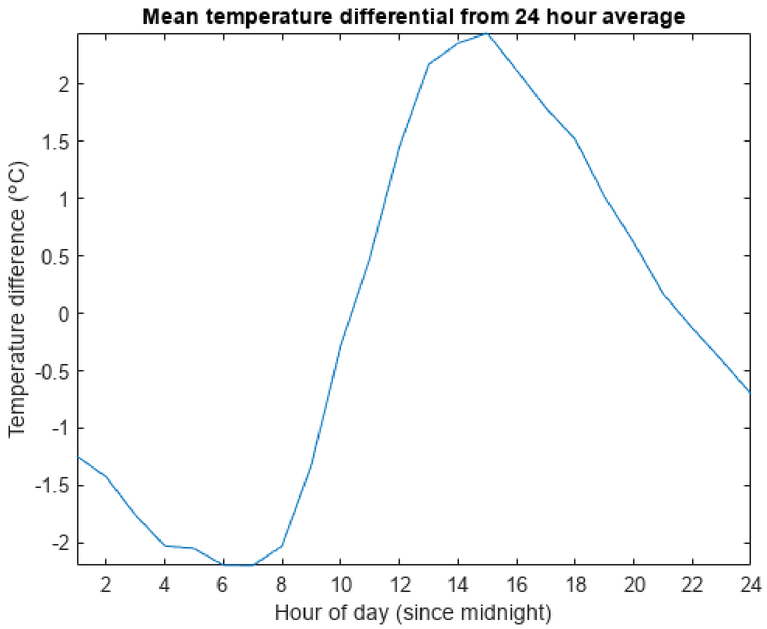
- Audio
  - Speech recognition, sound classification, source separation

Example: Temperature at Logan Airport

- Trends can occur at multiple time scales
- Smoothing and differencing are important forms of analysis







https://www.mathworks.com/help/signal/ug/signal-smoothing.html

How are 1D signals different from other problems?

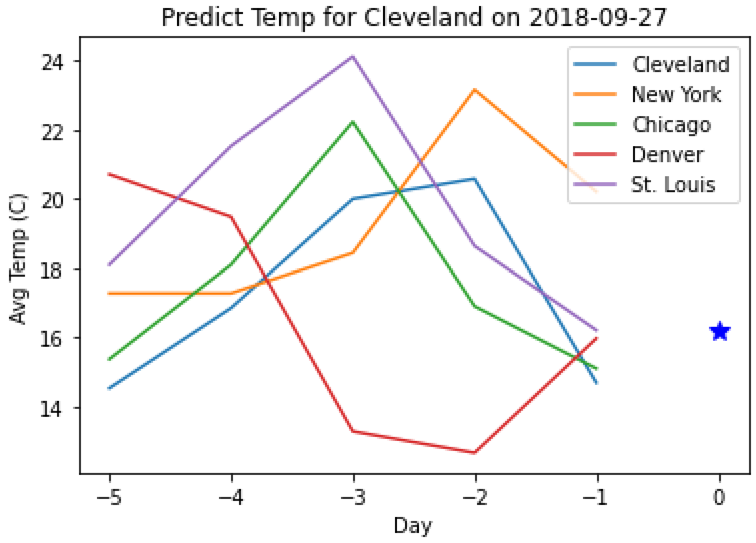
1. Signals are often **periodic**, and can be analyzed in terms of slow and fast moving changes

2. Signals can have **arbitrary length**, and can be analyzed in terms of recent data points (windows) or some cumulative representation

Example: city temperature prediction

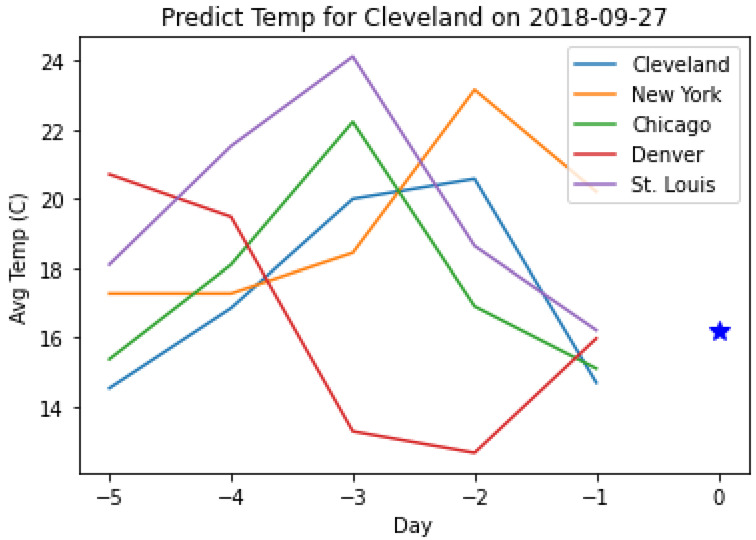
Recall HW 1: predict temperature in Cleveland based on temperatures from US cities in previous days

What did we do to handle this time series?



Example: city temperature prediction

Recall HW 1: predict temperature in Cleveland based on temperatures from US cities in previous days



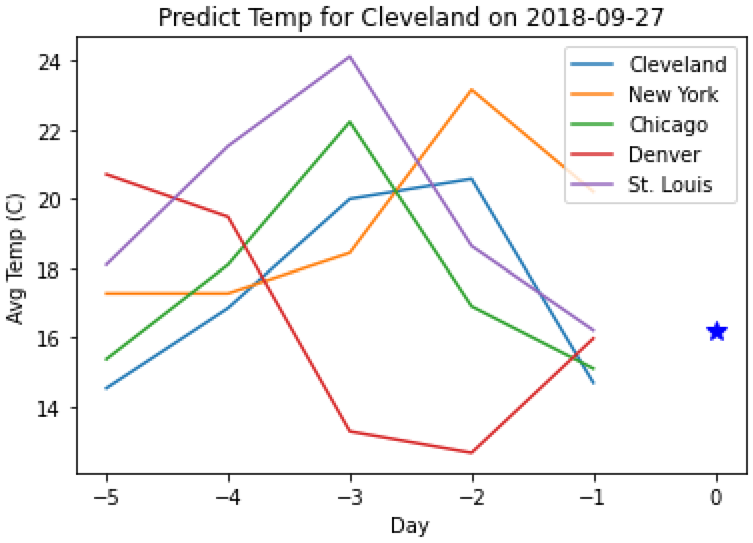
What did we do to handle this time series?

1. Windowed prediction: Consider temperature given five preceding days
2. Various prediction methods
    1. Linear regression
    2. Nearest neighbor
    3. Random forest

Example: city temperature prediction

Recall HW 1: predict temperature in Cleveland based on temperatures from US cities in previous days

What could we do to improve prediction?

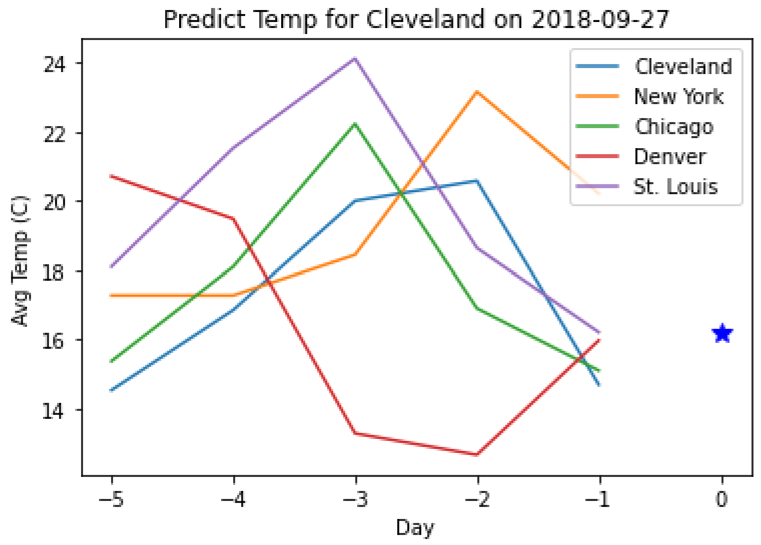


1D Time Series more generally

Recall HW 1: predict temperature in Cleveland based on temperatures from US cities in previous days

What could we do to improve prediction?

1. Take into account other features: time of year, other atmospheric effects
2. Learn better representations, e.g. use multitask learning to regress difference of temperature for each city from previous day in a deep neural net
3. Reframe in terms of differences from recent days and same day in recent years



1D Series Prediction

- Common to apply 1-D filter operations to smooth (e.g. 1D Gaussian) or highlight differences (e.g. difference filter [-1 1 0])
- Signal is often converted into fixed length by windowing and/or padding
- Predictions by dynamic models can also be used as features, e.g. accounting for position, velocity, and acceleration
- Continuity of features can be achieved using convolutional networks, LSTMs, or other recurrent networks

ARIMA: autoregressive integrated moving average

ARIMA is statistical time series model that predicts the *difference* from the last value based on the recent history of consecutive differences

$$x_t = \Delta^d y_t \qquad x_t = \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}$$

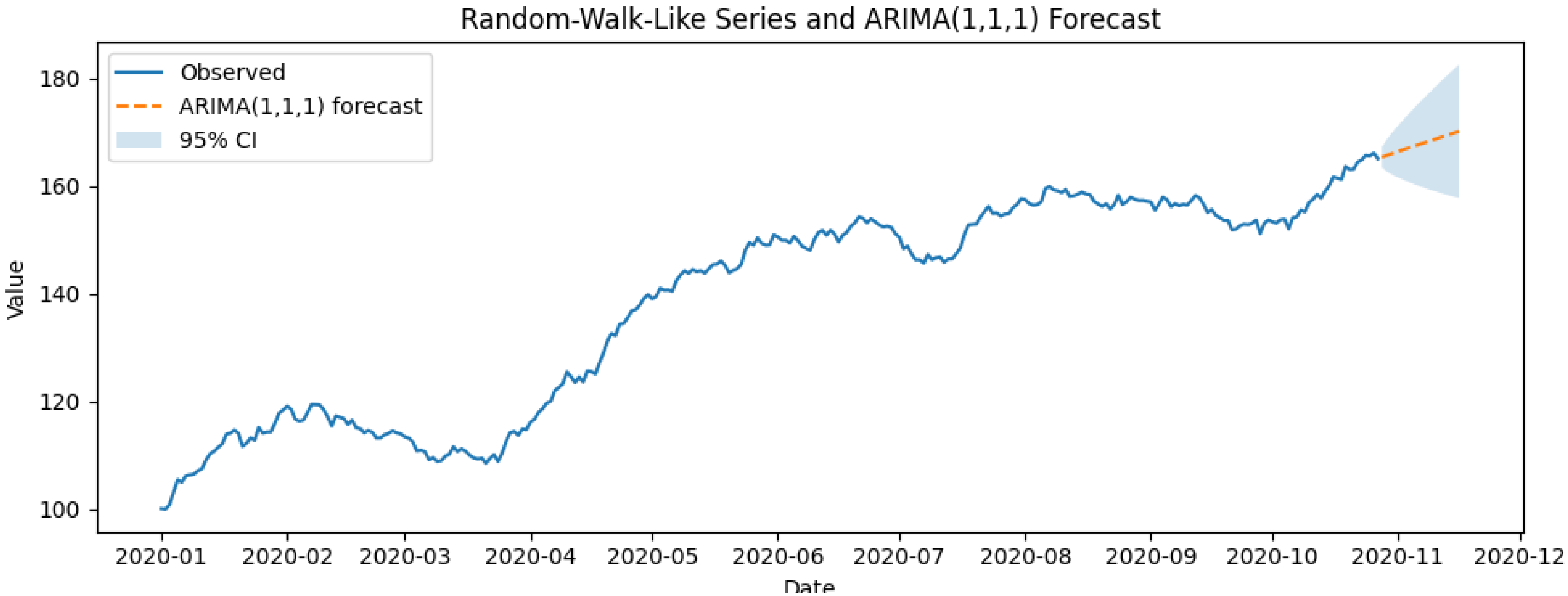
d^{th} order difference of time series values

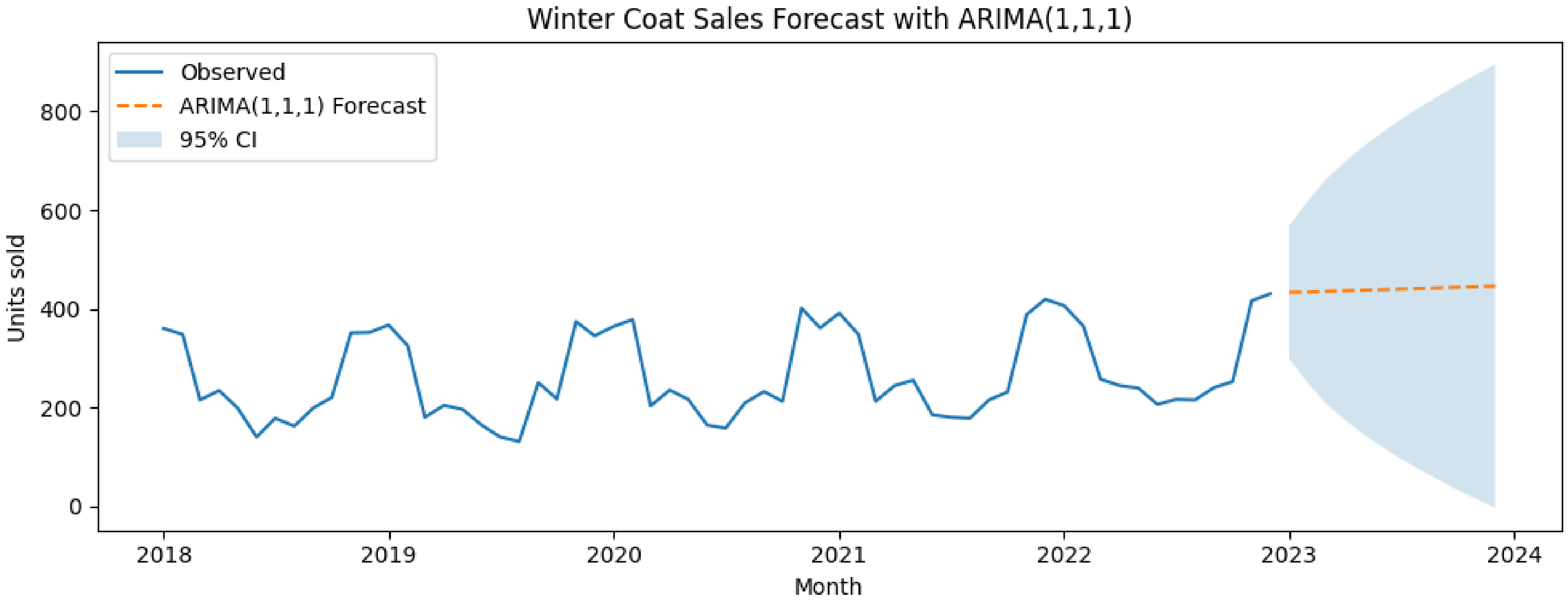
Linear predictor from recent differences

Linear predictor from recent forecasting errors

Parameterized with d (order of difference), p (series history length), q (error history length)

ARIMA Predictions





SARIMA: **seaonal** autoregressive integrated moving average

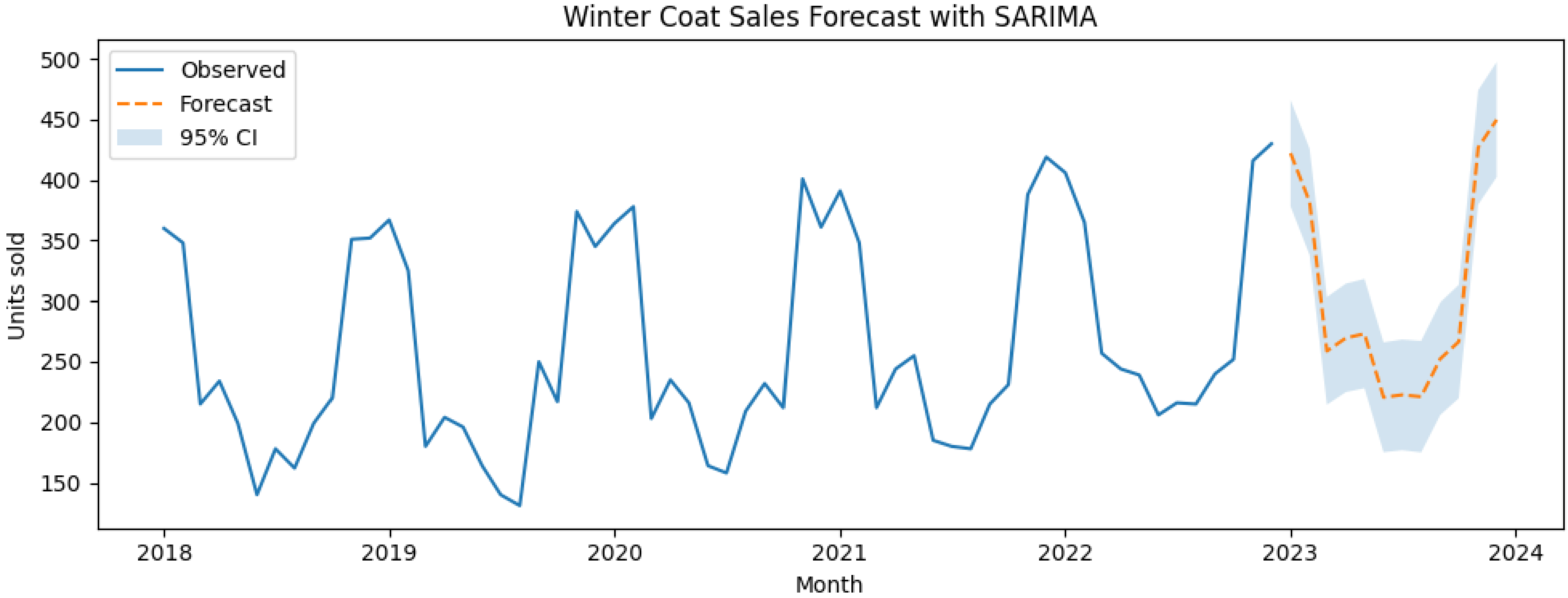
SARIMA extends ARIMA by adding a seasonal term, e.g. also considering differences from a previous cycle, e.g.

- Coat sales in December 2025 based on recent months *and* December sales in previous years

$$x_t = c + \sum_{i=1}^{p} \phi_i \, x_{t-i} + \sum_{k=1}^{P} \Phi_k \, x_{t-ks} + \epsilon_t + \sum_{j=1}^{q} \theta_j \, \epsilon_{t-j} + \sum_{\ell=1}^{Q} \Theta_\ell \, \epsilon_{t-\ell s}$$

Similar terms to ARIMA except based on differences and forecasting errors at interval of s

SARIMA Prediction



Deep networks for time series forecasting

- The success of deep networks in vision, language, and audio is due to ability to train pre-trained models on large corpora and transfer representations to target tasks

- But time series problems seem quite different, e.g. predicting the price of commodities, the temperature, or the favorability polling of a political candidate

- Can we pre-train a model that adapts well to target time series forecasting tasks?

Yes, we can train deep models to forecast, but tree-based and statistical models are usually more practical

- Methods like Temporal Fusion Transformer (TFT), Informer, and N-BEATs are:
  - Interesting and innovate
  - Perform well in academic benchmarks and used widely in research

- Tree-based models (e.g. XGBoost, random forests) and ARIMA/SARIMA and other statistical models are:
  - More interpretable (easier to debug and explain)
  - Lower computation
  - Much more widely used

- In practice, raw prediction accuracy is only one factor – also important are
  - Interpretability
  - Stability
  - Controllability
  - Ease of incorporating expert knowledge
  - Ease of monitoring & debugging
  - Low maintenance cost

Q1-3

https://tinyurl.com/AML441-L23

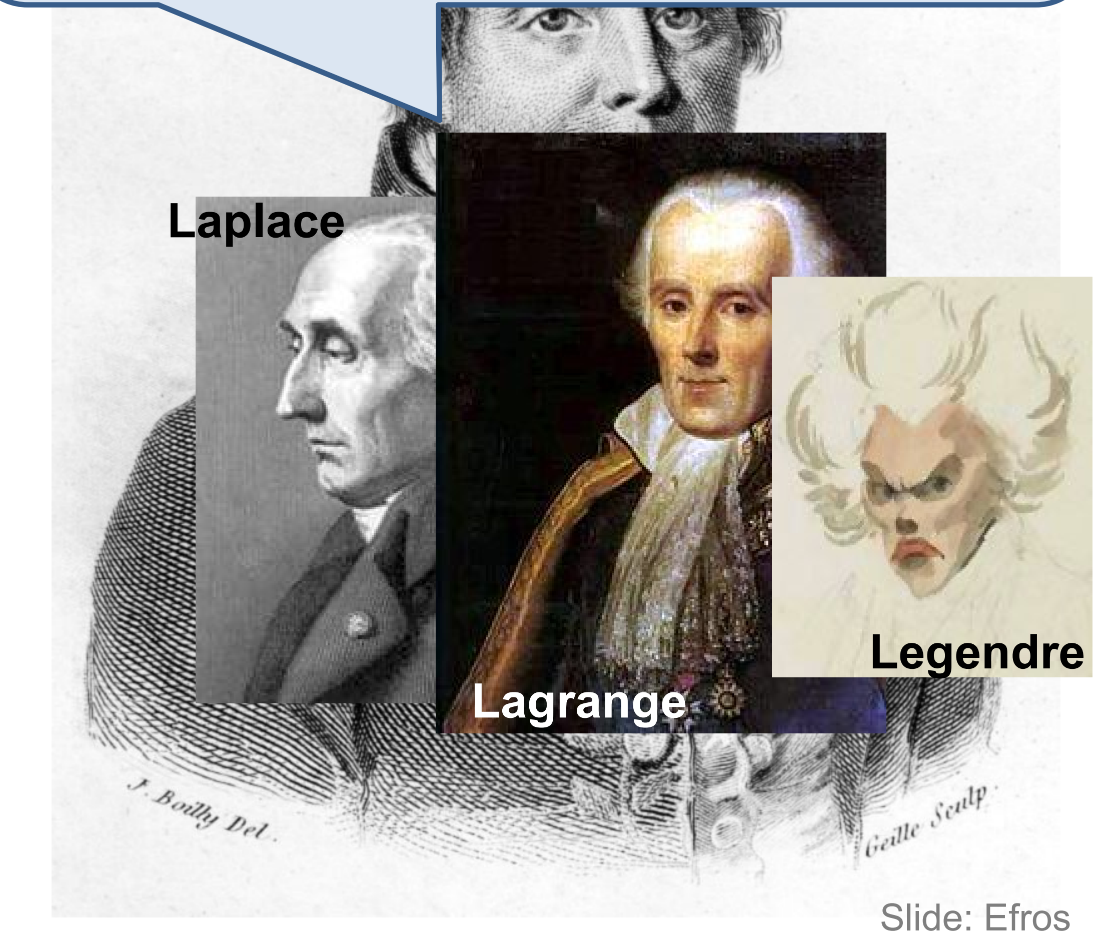
To understand audio, we need better ways to represent frequency

Jean Baptiste Joseph Fourier had a crazy idea in 1807

***Any** univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.*

- Don’t believe it?
  - Neither did Lagrange, Laplace, Poisson and other big wigs
  - Not translated into English until 1878!
- But it’s (mostly) true
  - called Fourier Series
  - there are some subtle restrictions

...the manner in which the author arrives at these equations is not exempt of difficulties and...his analysis to integrate them still leaves something to be desired on the score of generality and even rigour.



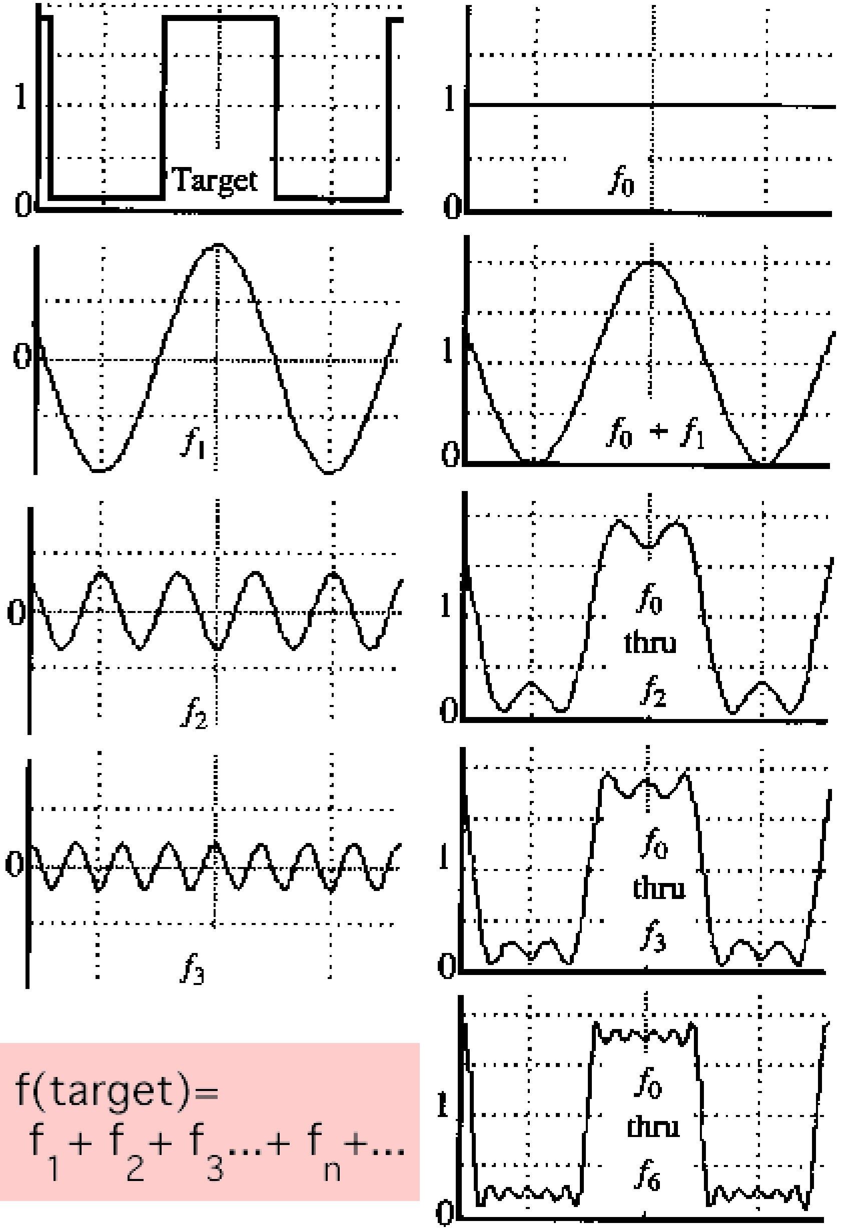
Slide: Efros

A sum of sines

Our building block:

$$A\sin(\omega x+\phi)$$

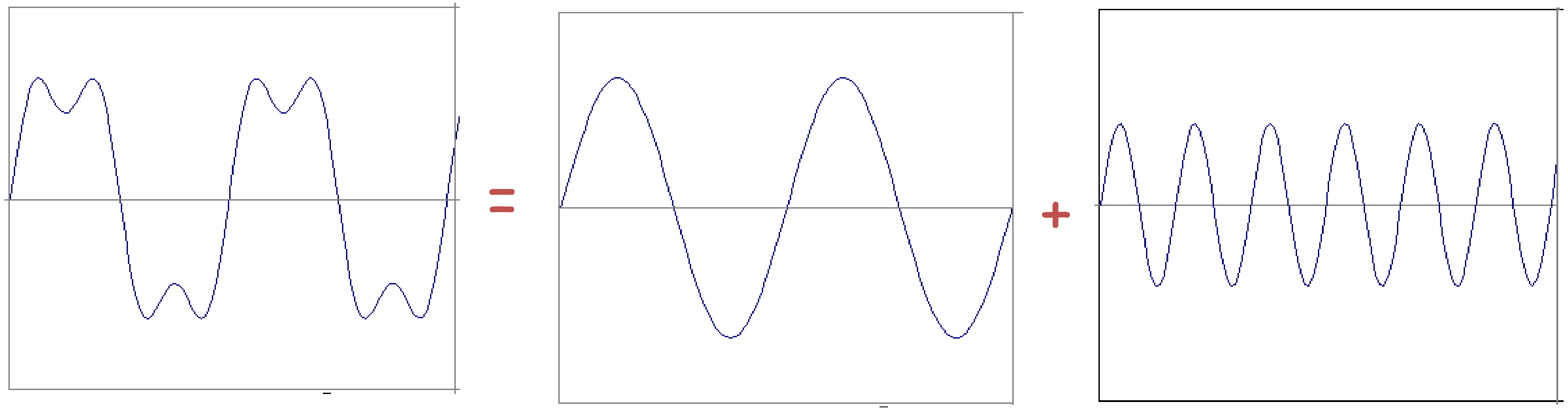
Add enough of them to get any signal *f(x)* you want!

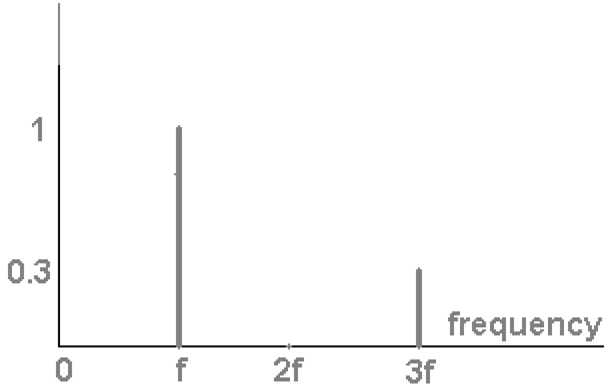


f(target)=
$f_1 + f_2 + f_3 ... + f_n + ...$

Frequency Spectra

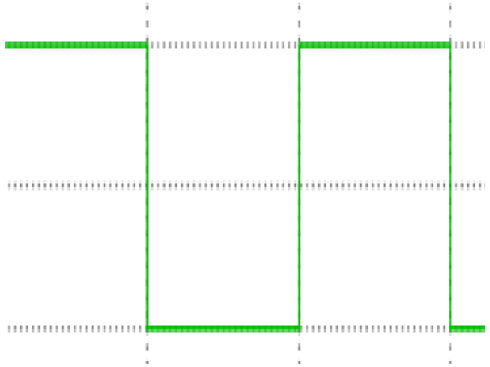
- example : $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$



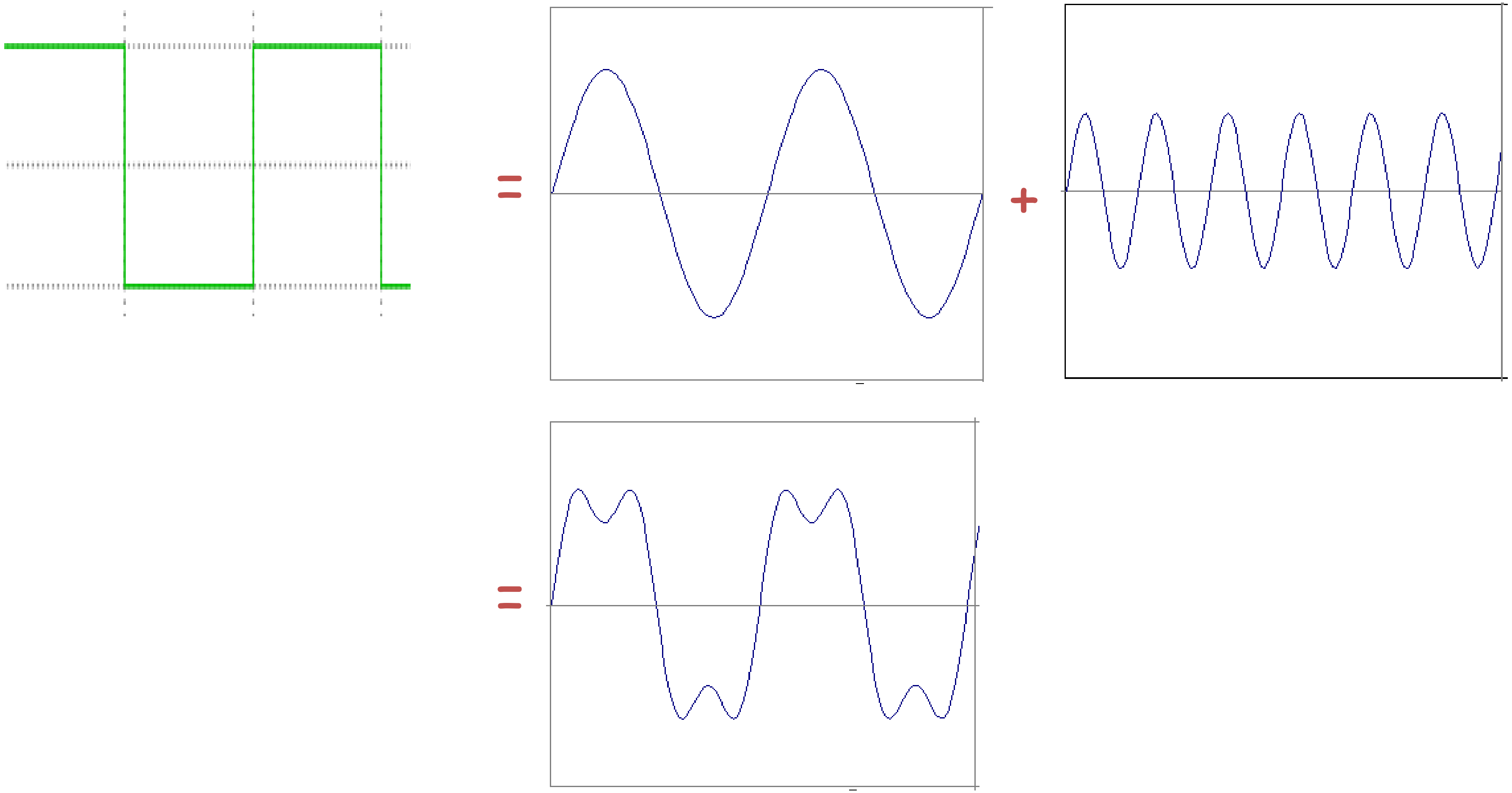


Slides: Efros

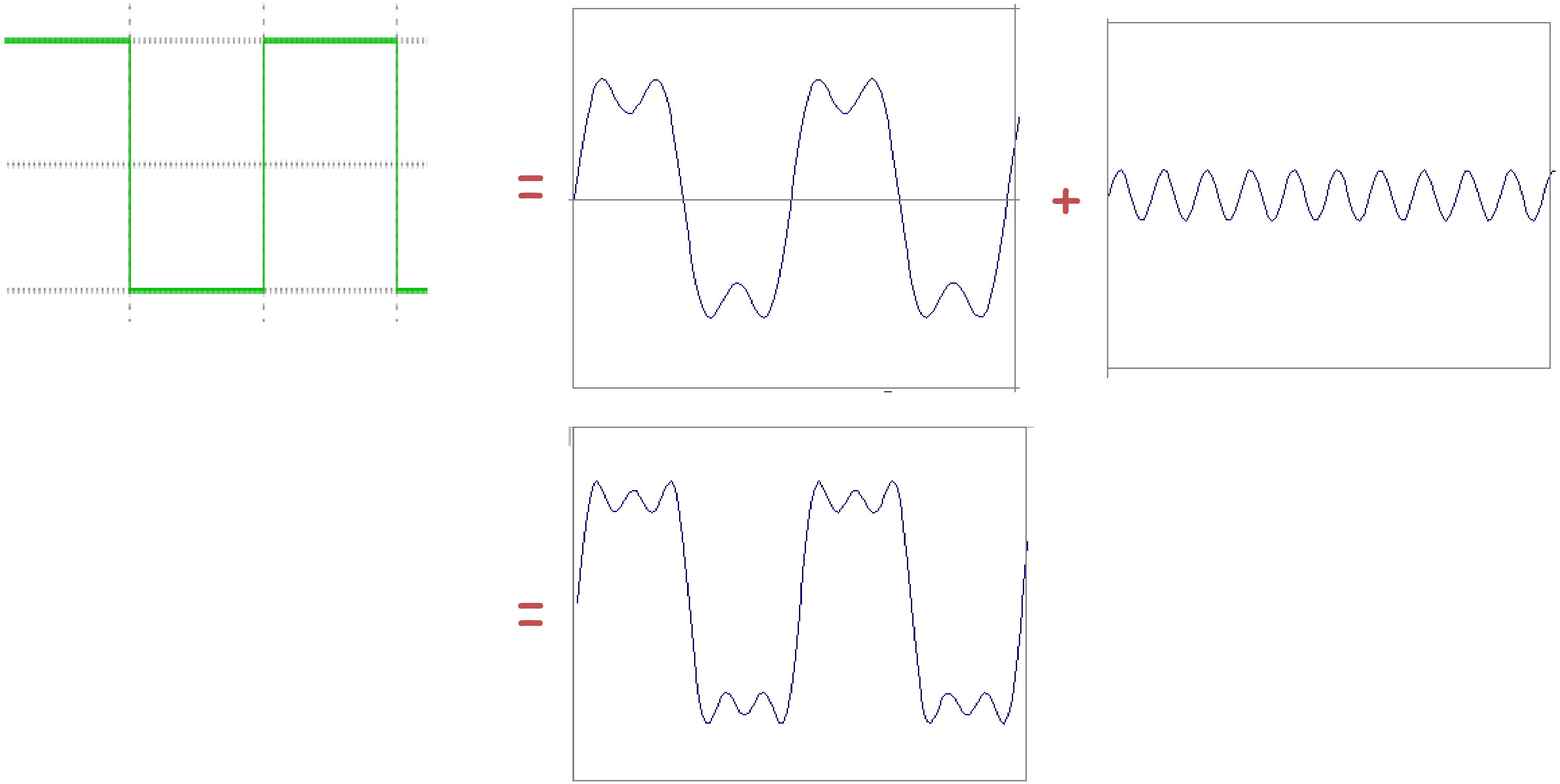
Frequency Spectra



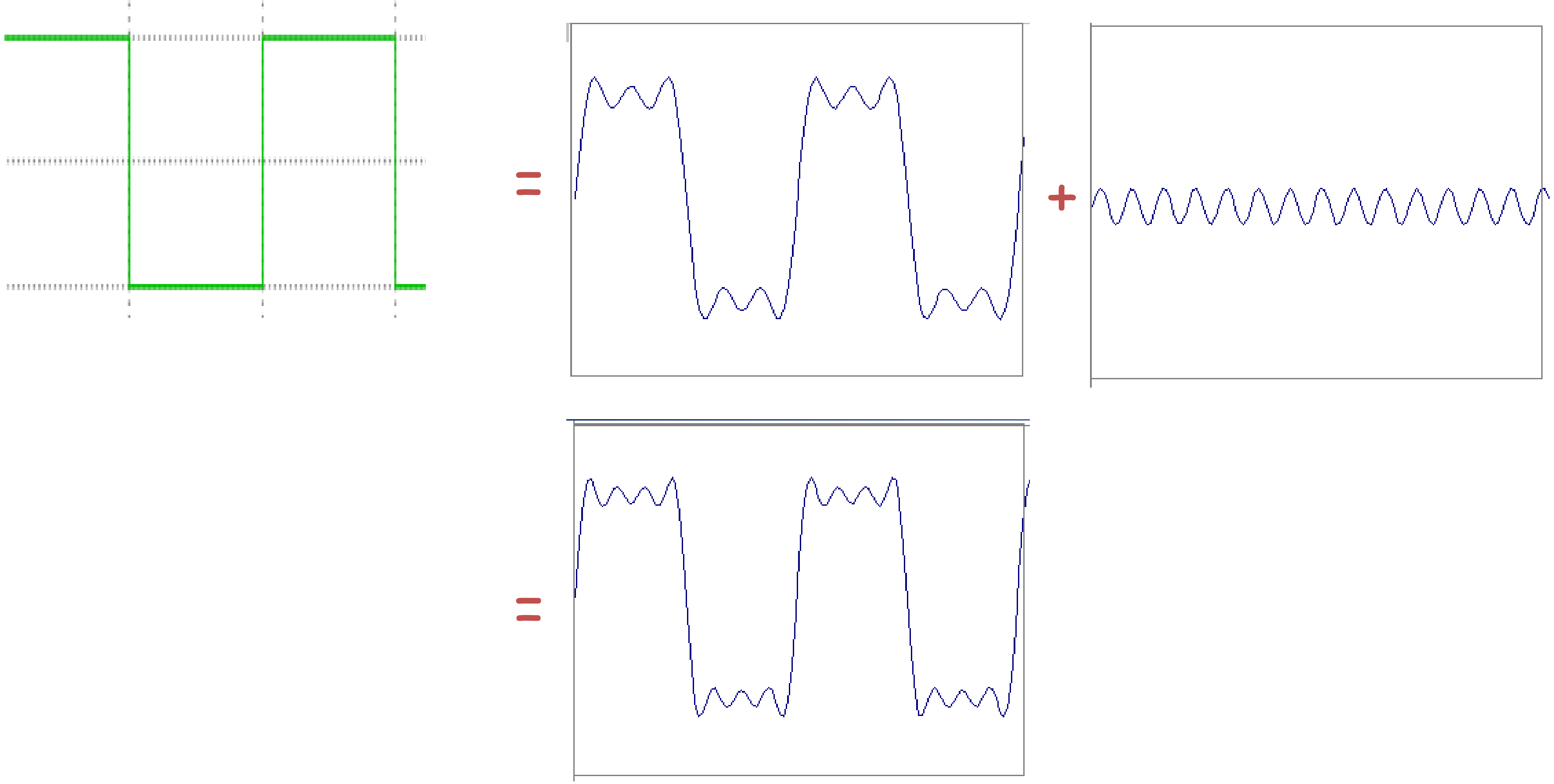
Frequency Spectra



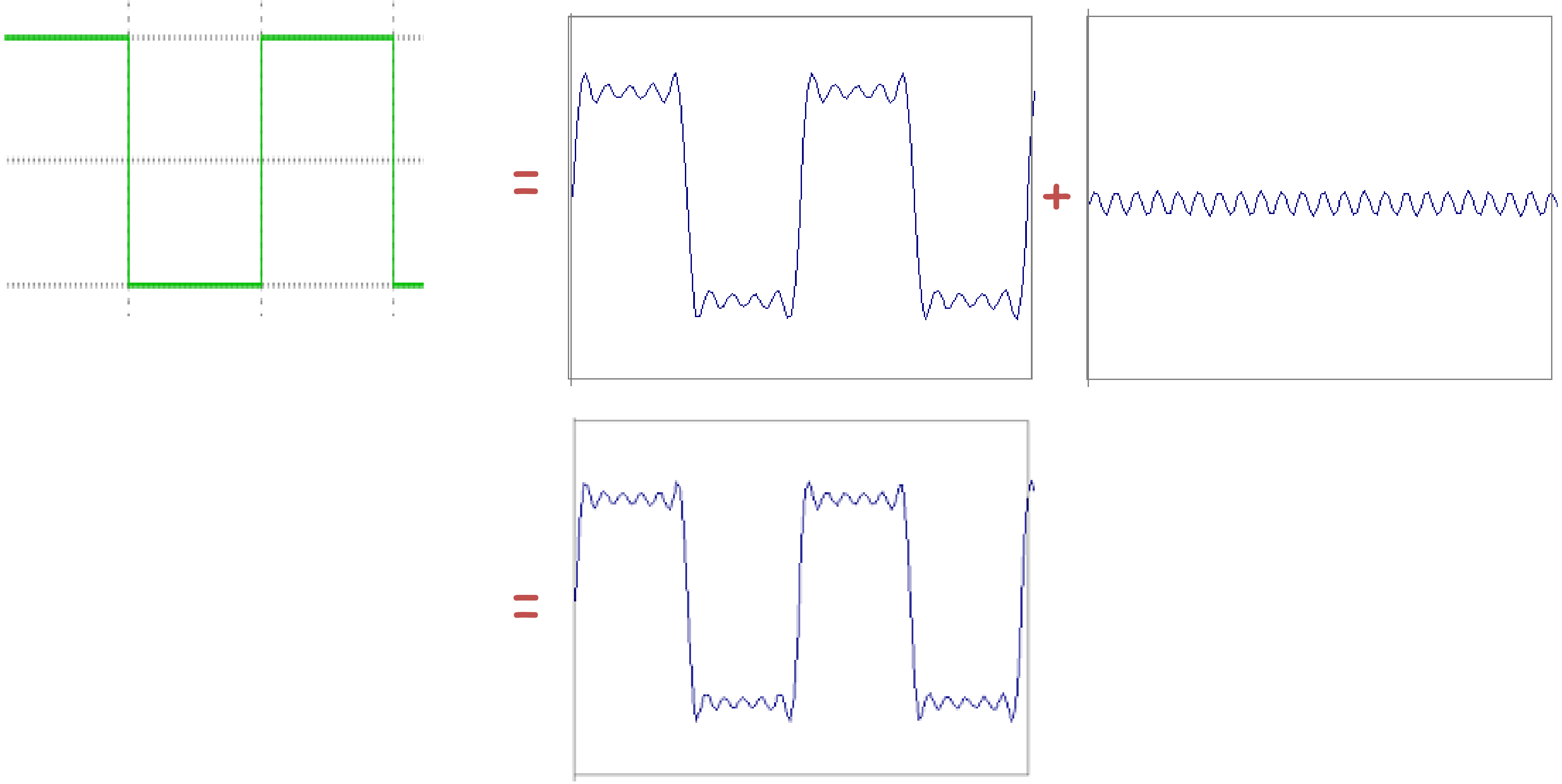
Frequency Spectra



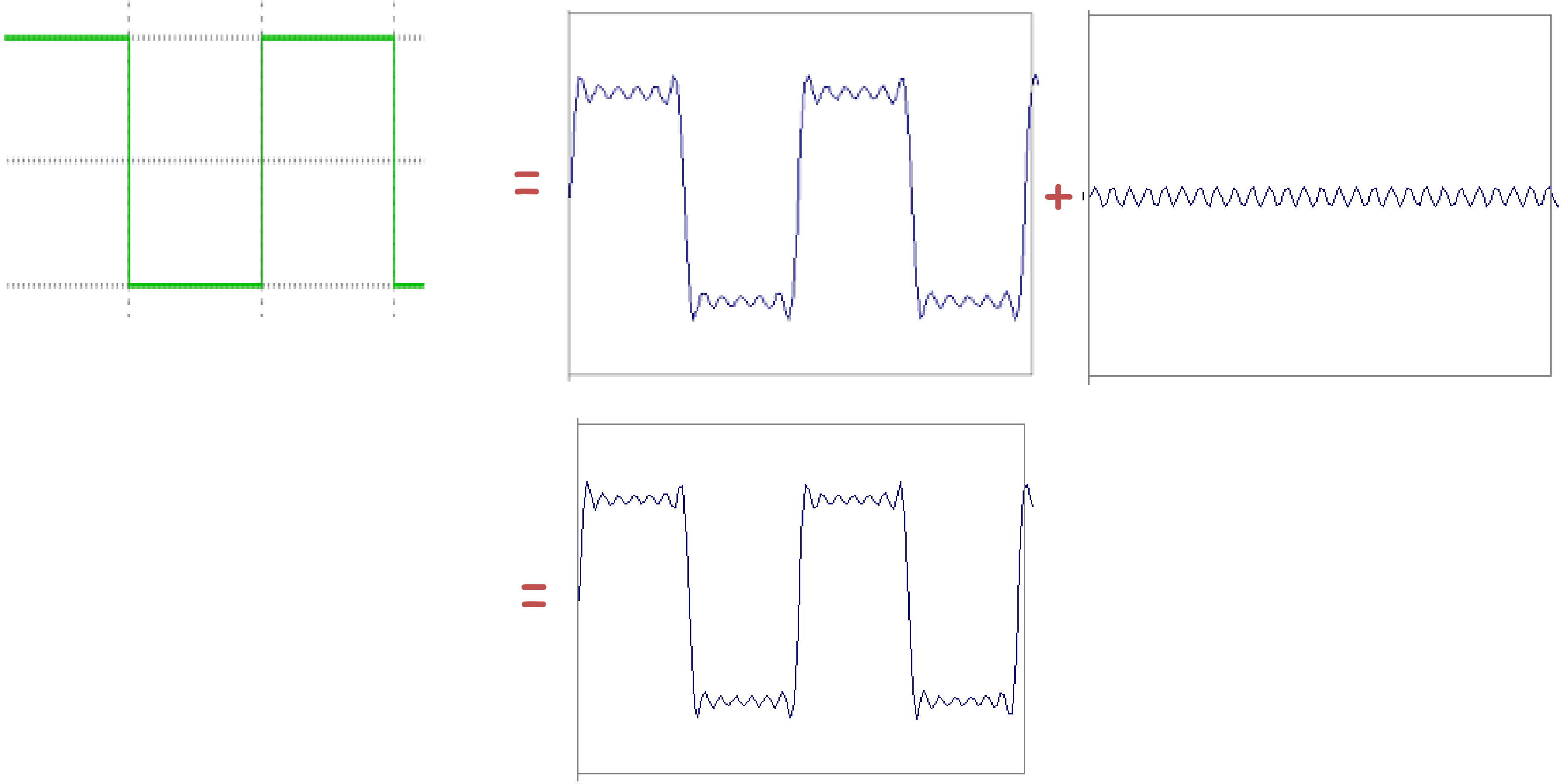
Frequency Spectra



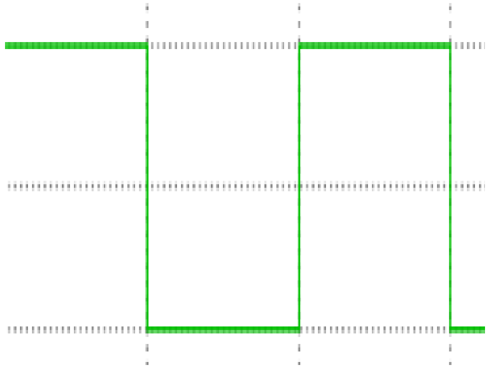
Frequency Spectra



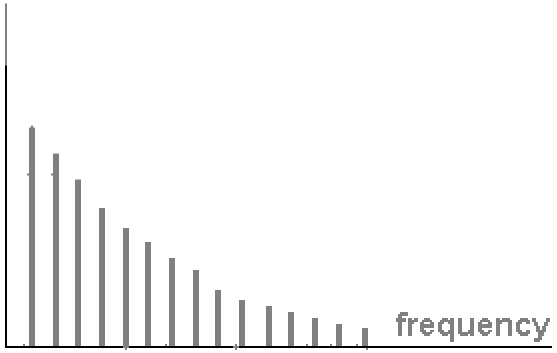
Frequency Spectra



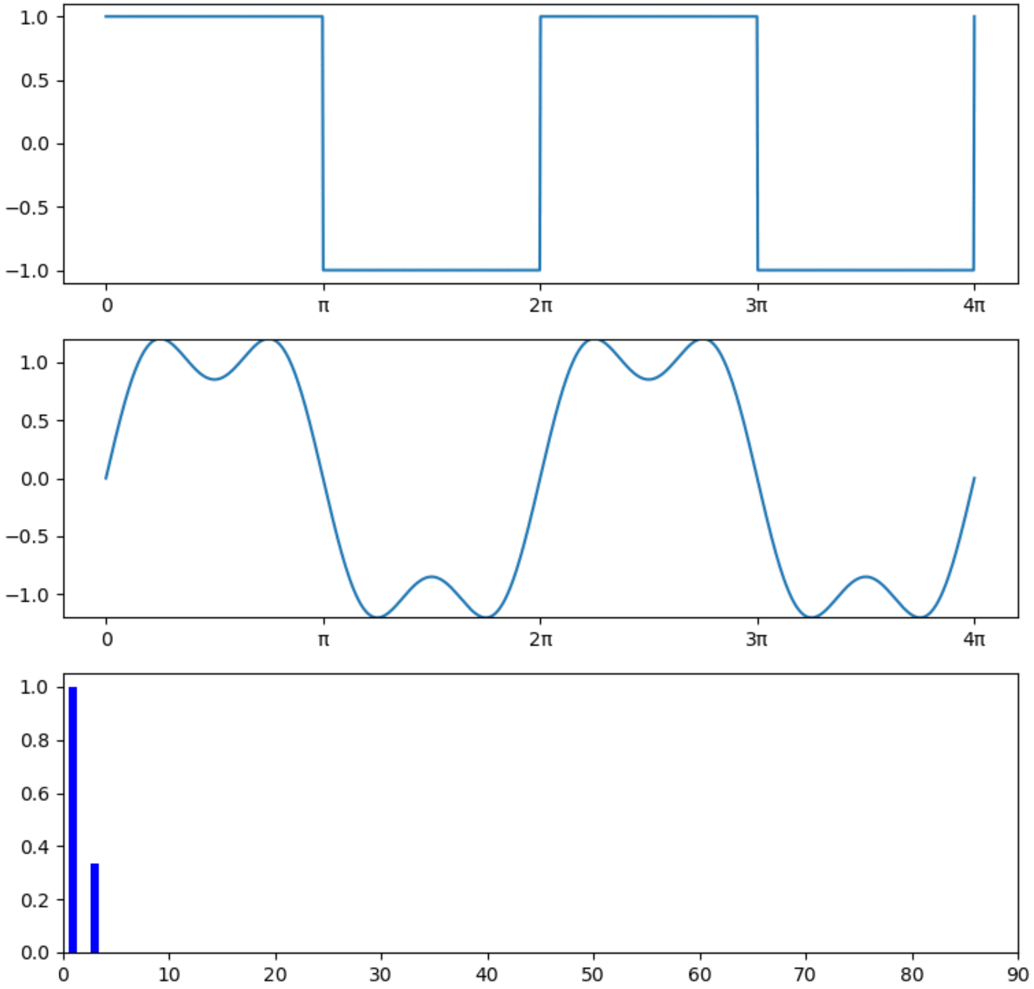
Frequency Spectra



$$= \boxed{A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)}$$



Frequency Spectra



Sound

- Sound is composed of overlaid sinusoidal functions with varying frequency and amplitude
- Digital sound is recorded by sampling the sound signal with high sampling frequency

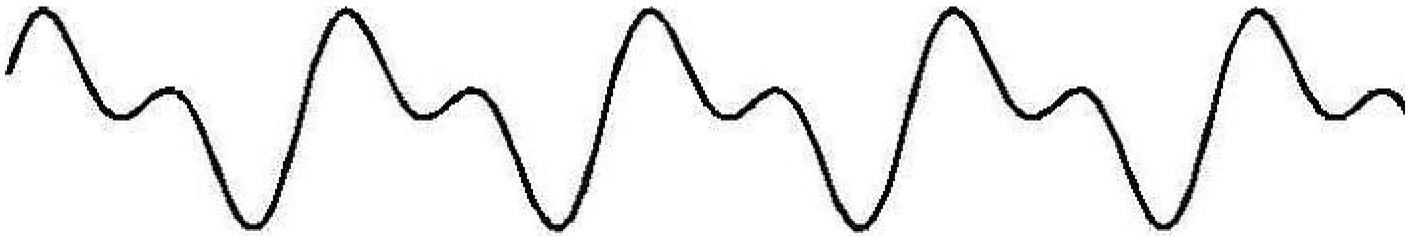


Fig src

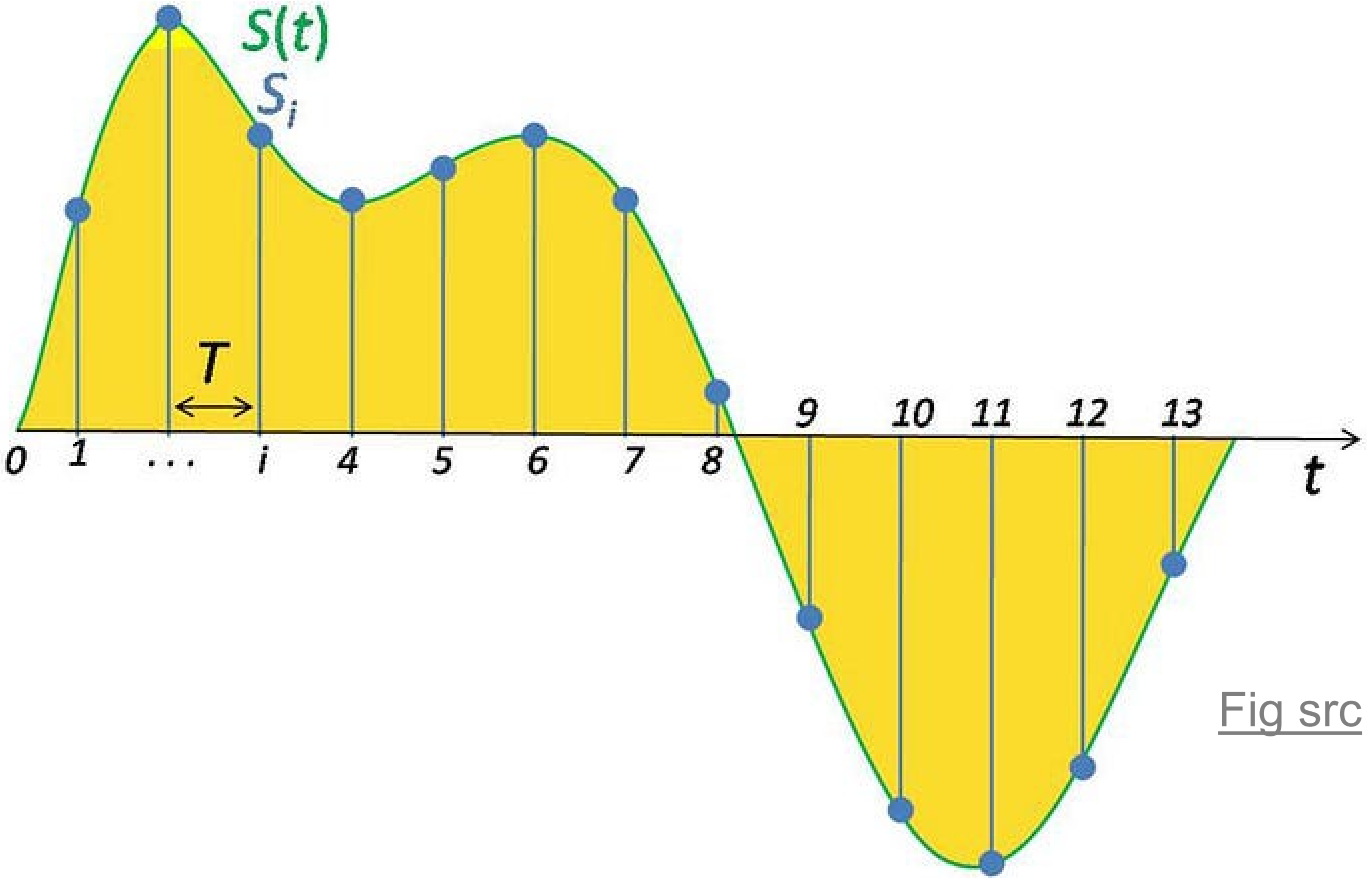


Fig src

Slide content source: TDS

Spectrogram

- Sound is a series of sinusoids with varying amplitudes, frequencies, and phases

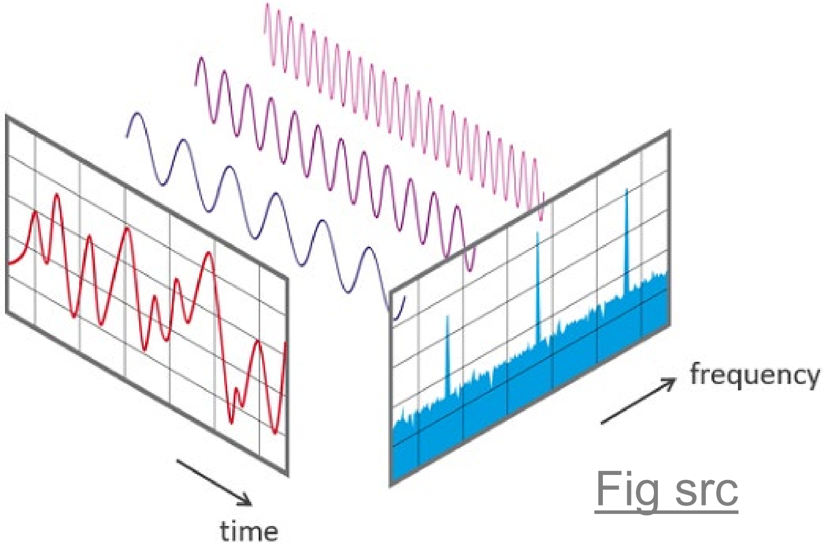
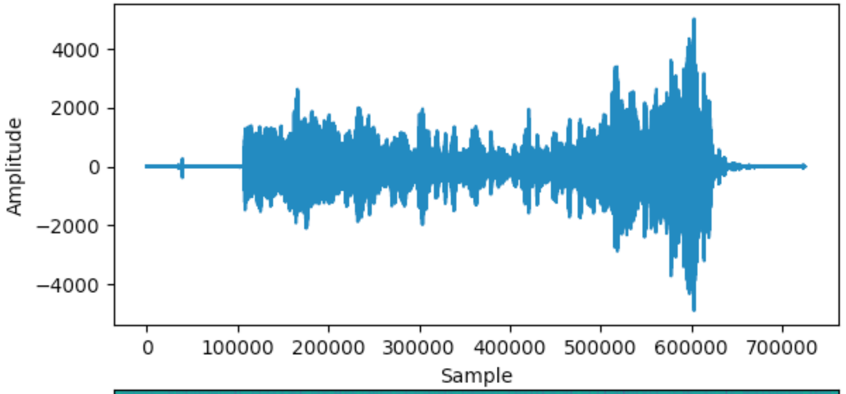
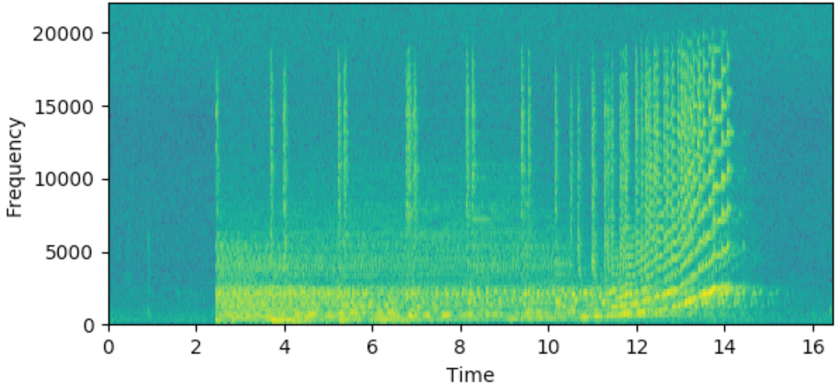


Fig src

- Total amplitude tells us how loud the sound is at some point, but that's not very informative



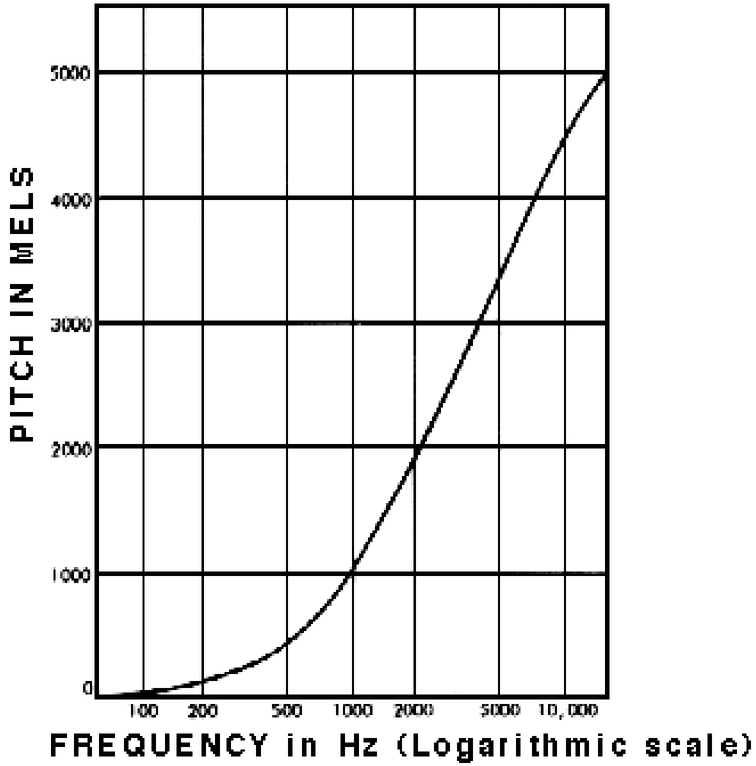
- The spectrogram tells us the power in each frequency over some time window



Slide content source: TDS

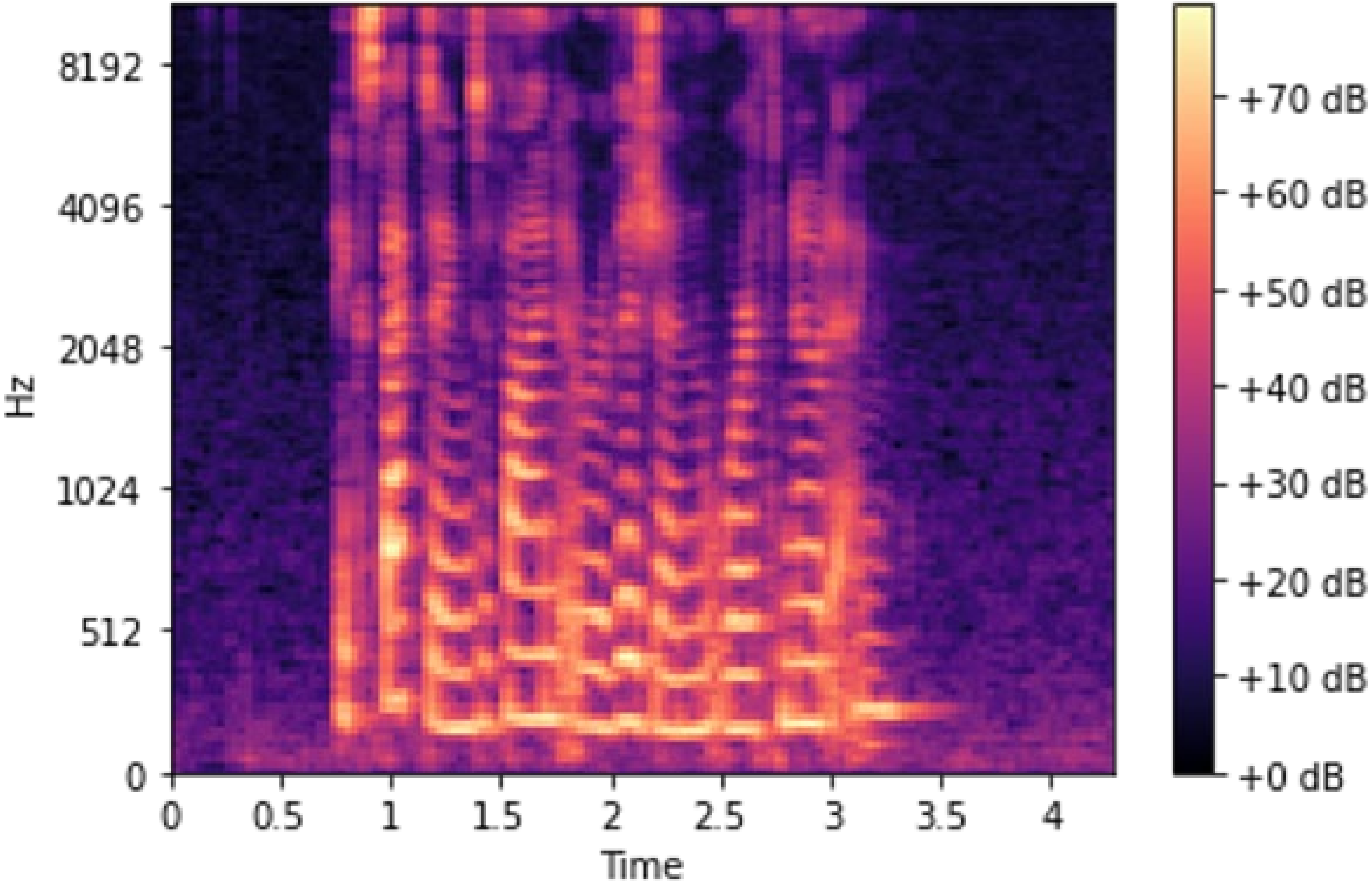
Mel Spectrograms

- Humans perceive frequency on a logarithmic scale, e.g. each octave in music doubles the frequency
- Mel scale maps frequency to human perception of pitch
- Mel scale records amplitude in decibels (logarithmic base 10)



Slide content source: TDS

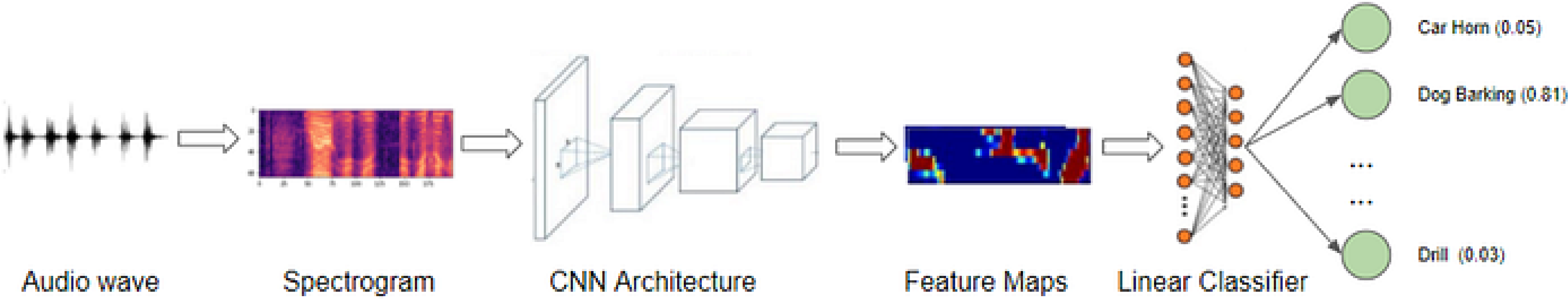
Mel Spectrogram



Slide content source: TDS See this link for code to generate Mel Spectrogram

Audio Classification

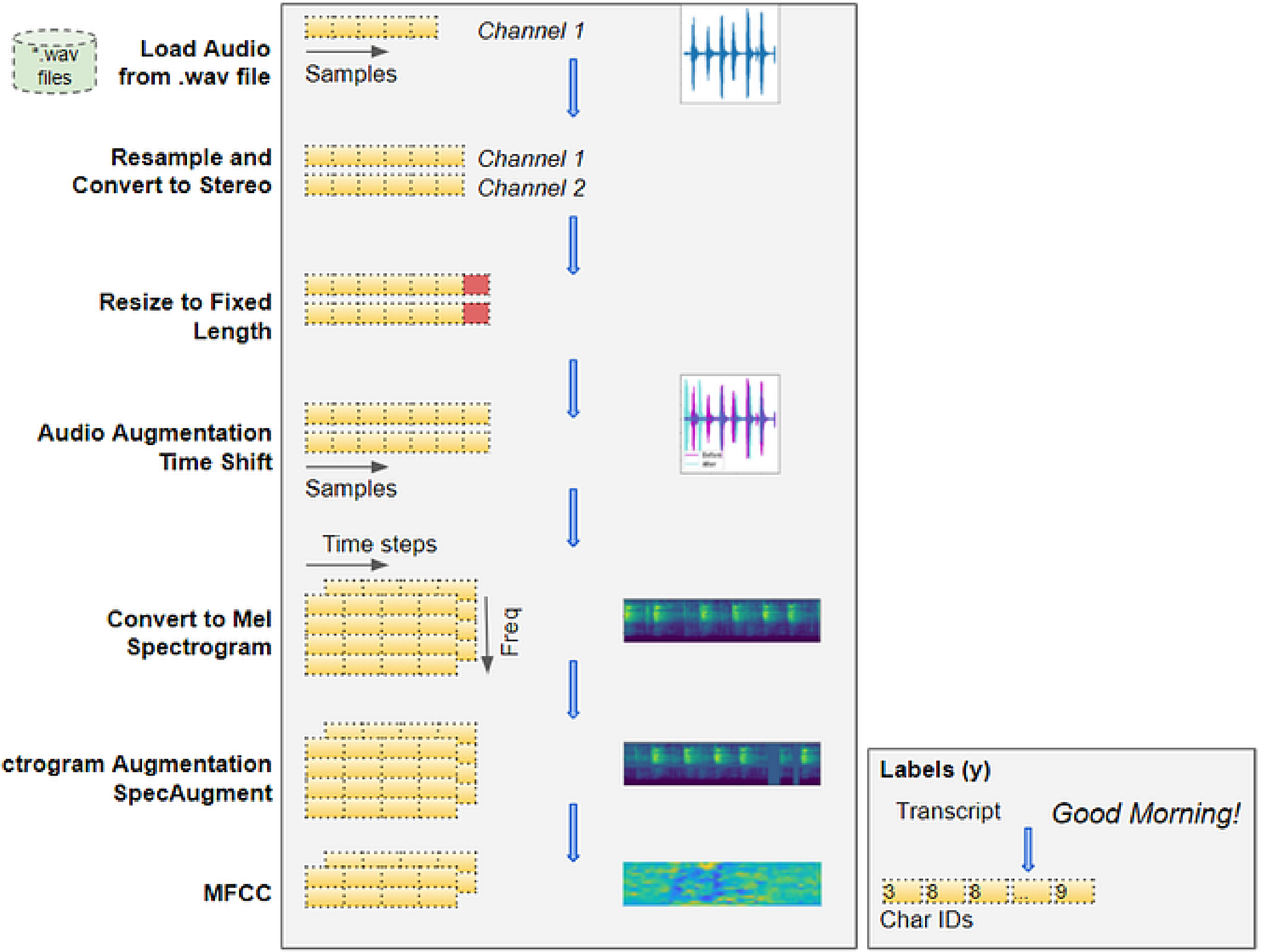
1. Pre-process into Mel Spectogram Image
2. Apply vision-based architectures to classify
   - Data augmentation can include time shift on audio wave and time/frequency masking on spectrogram



Details and code here: https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5

Audio to Speech (ASR)

- Process data, similar to audio classification
- MFCC processes Mel Spectrogram with DCT to get a compressed representation that focuses on speech-related frequencies



*.wav files

Load Audio from .wav file

Channel 1

Samples

Resample and Convert to Stereo

Channel 1

Channel 2

Resize to Fixed Length

Audio Augmentation Time Shift

Samples

Time steps

Convert to Mel Spectrogram

Freq

Spectrogram Augmentation SpecAugment

MFCC

Labels (y)

Transcript

Good Morning!

3 8 8 ... 9

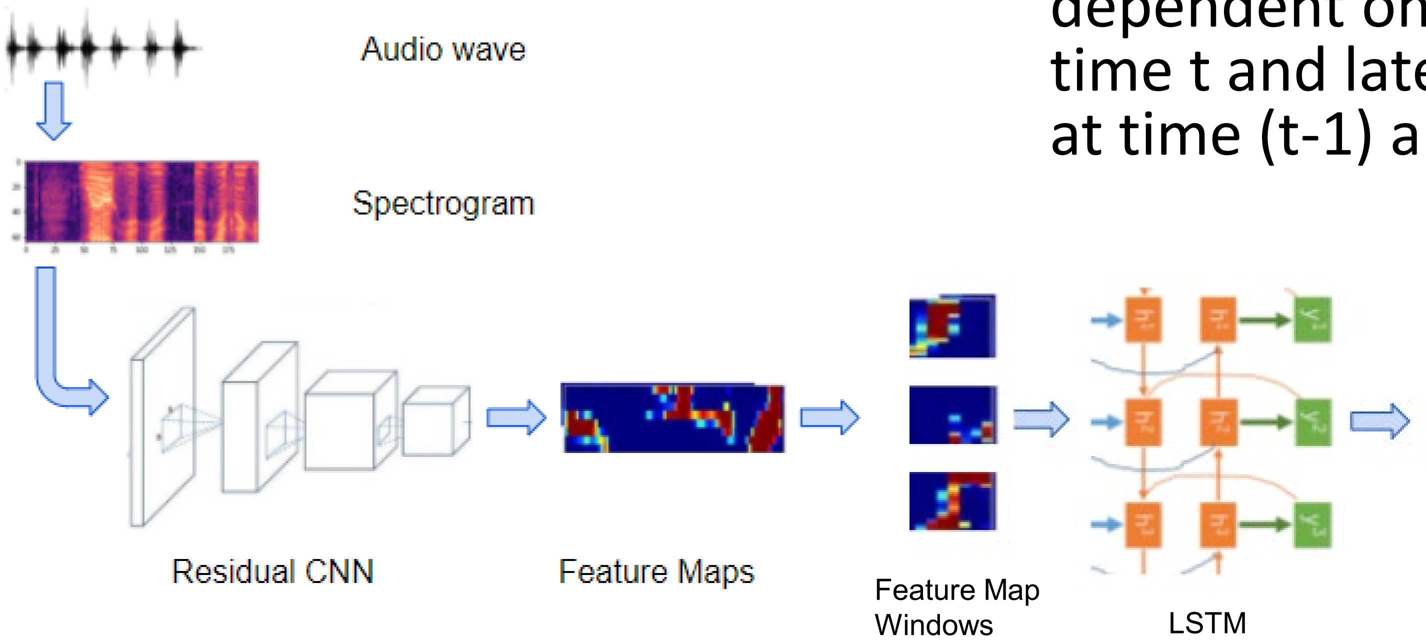
Char IDs

Content src

ASR

Deep network with vision architecture extracts features

Recurrent network (e.g. LSTM) models each output at time t as dependent on the features at time t and latent representations at time (t-1) and/or (t+1)

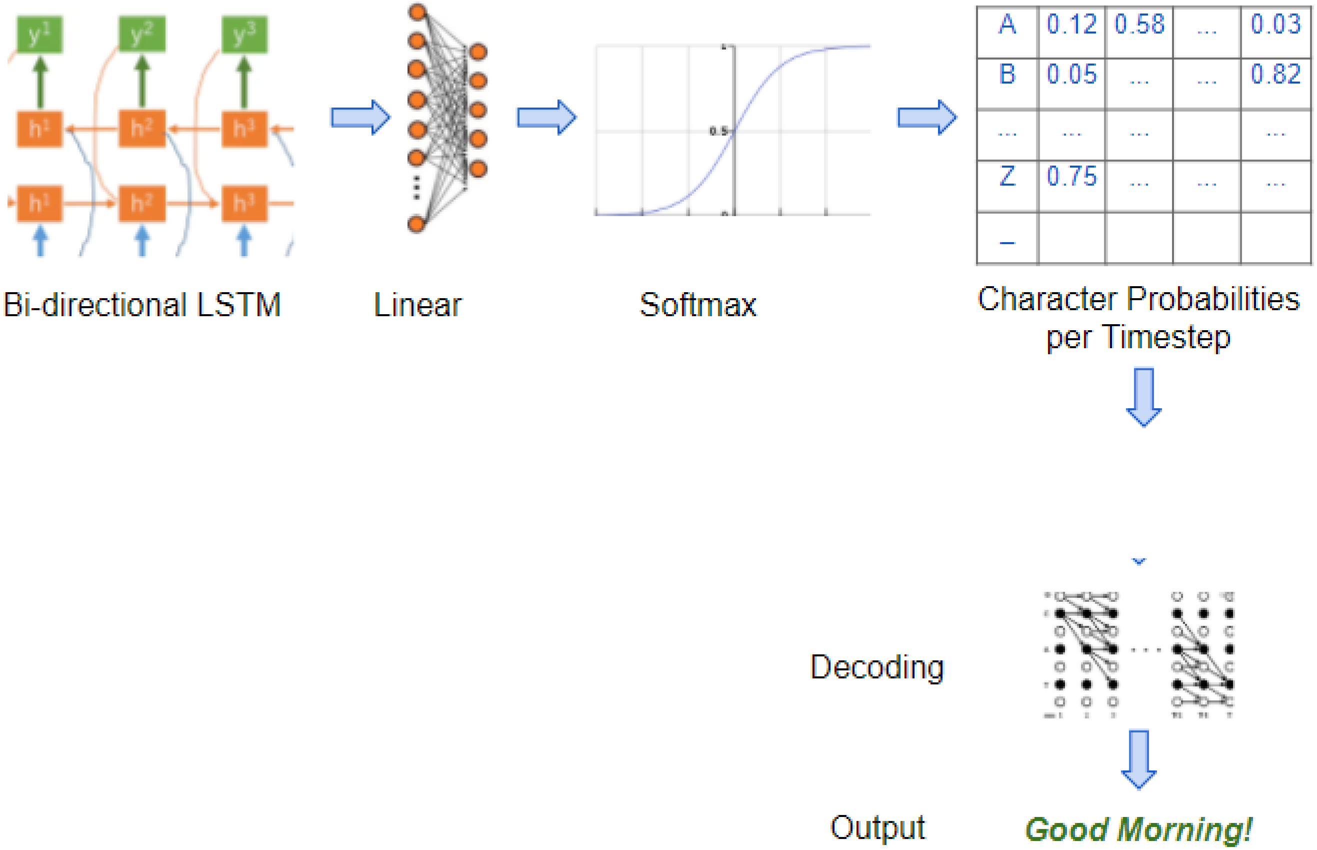


Based on Baidu’s Deep Speech model, as described here

ASR

- Each time step predicts a probability for each character

- Character probabilities are decoded into text output



Bi-directional LSTM

Linear

Softmax

A	0.12	0.58	...	0.03
B	0.05	...	...	0.82
...	...	...		...
Z	0.75	...	...	...
–				

Character Probabilities per Timestep

Decoding

Output

Good Morning!

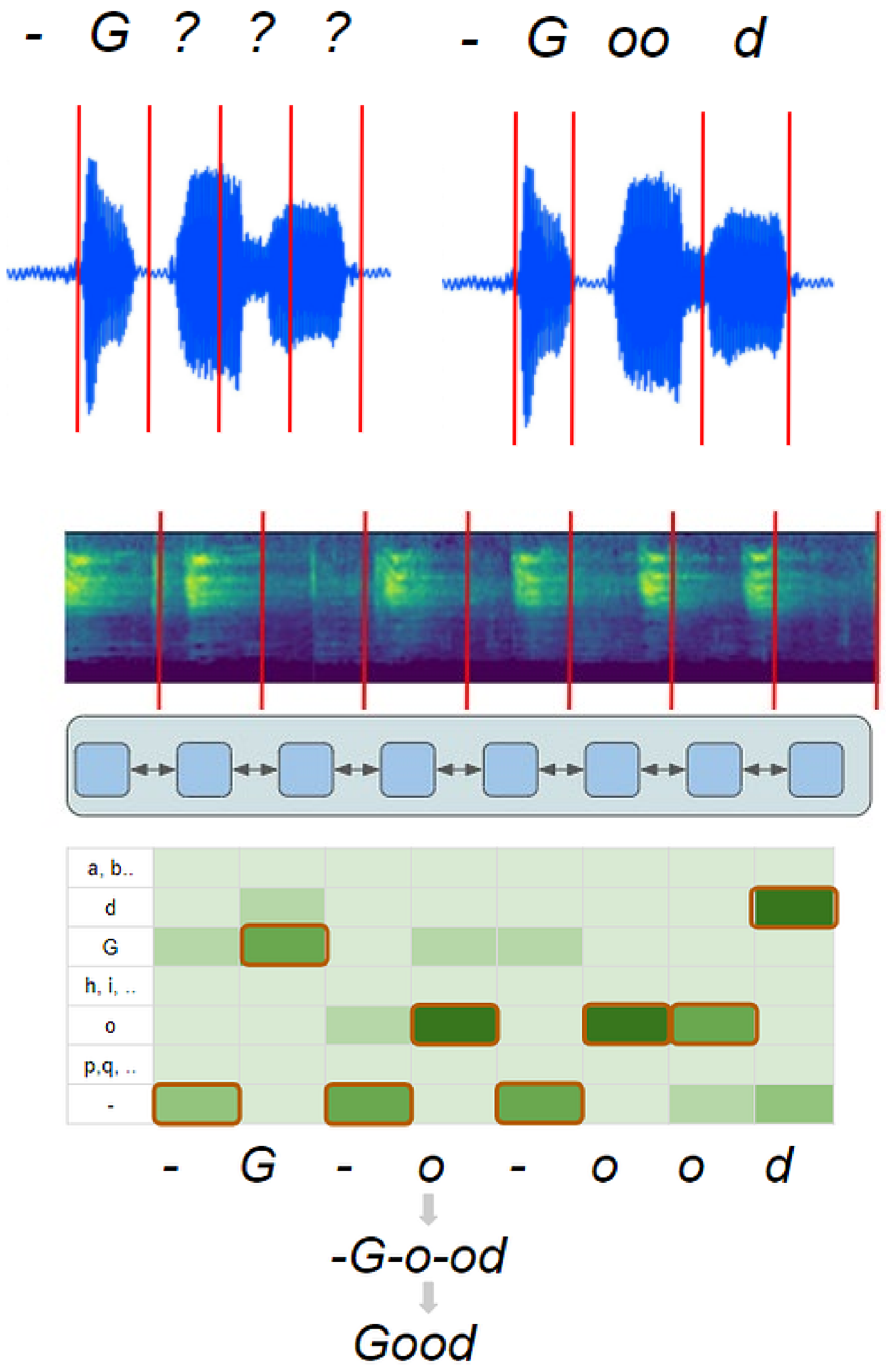
Based on Baidu’s Deep Speech model, as described here

ASR

- A challenge of ASR is temporal spacing between characters

- Audio is sliced uniformly and fed into RNN

- RNN predicts character probabilities, merges repeated characters, and removes blanks

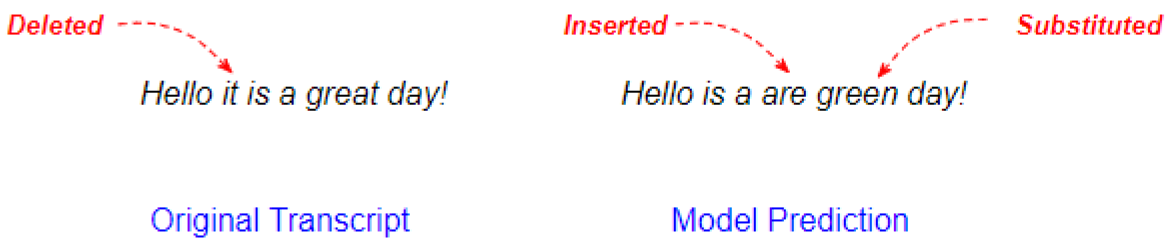


Based on Baidu’s Deep Speech model, as described here

Other details

- Loss based on likelihood of true character sequence

- Error is often reported as Word Error Rate or Character Error Rate



- A language model and/or beam search can be used to improve output

As described here

Popular libraries for audio processing

- Librosa: https://librosa.org/doc/latest/index.html
- Torch Audio: https://pytorch.org/audio/stable/index.html
- Others: https://wiki.python.org/moin/Audio/
- HuggingFace Models: https://huggingface.co/docs/transformers/tasks/audio_classification

Audio Foundation Models

Domain	Model (with URL)	Typical Tasks
Speech	Metis https://metis-demo.github.io/	ASR, TTS, voice conversion, speech enhancement
Speech	WhisperX https://github.com/m-bain/whisperX	Speech transcription, translation
Music	LLark https://github.com/spotify-research/llark	Music understanding, captioning, metadata extraction
Music	MusicGen https://audiocraft.metademolab.com/musicgen.html	Music generation
Universal / General Audio	UniAudio https://github.com/yangdongchao/UniAudio	Speech + audio + music generation
Universal / General Audio	Qwen-Audio https://huggingface.co/Qwen/Qwen-Audio	Audio reasoning, classification, ASR
Universal / General Audio	Kimi-Audio https://github.com/MoonshotAI/Kimi-Audio	Audio understanding, generation, conversation
Domain-Specific (Bioacoustics)	BirdAVES https://github.com/earthspecies/aves	Birdsong modeling, species identification
Domain-Specific (Bioacoustics)	NatureLM-Audio https://earthspecies.github.io/naturelm-audio-demo/	Wildlife and ambient acoustic modeling

Q4

https://tinyurl.com/AML441-L23

And now, this…

Smarter Better Faster (my parody lyrics)

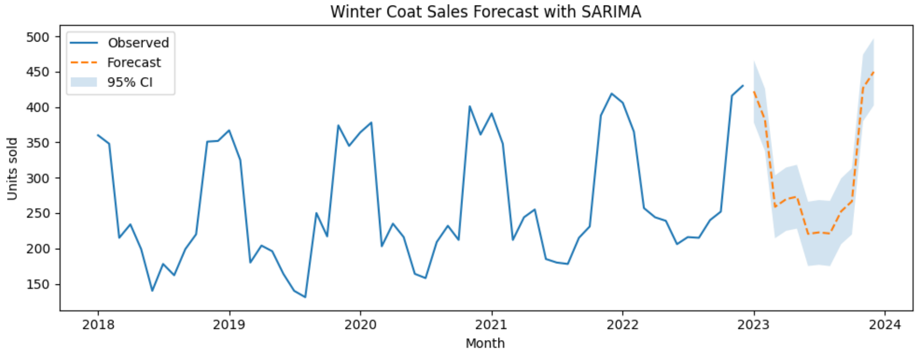
https://suno.com/song/eb2a33d3-aa01-466f-8d4b-a8ebcb49ced4

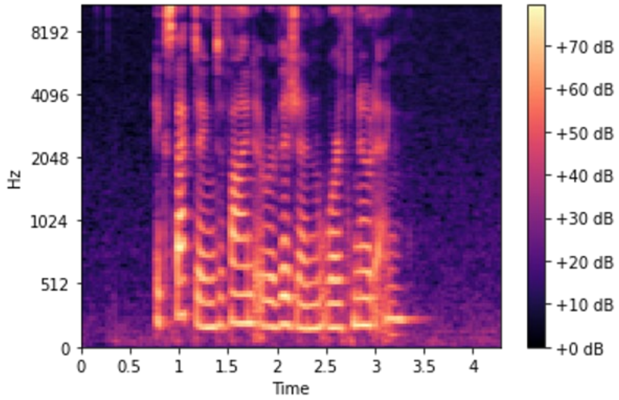
It’s my loss (my lyrics)

https://suno.com/song/2cdf6a66-7a1a-4b2f-afff-507d8554b610

Things to remember

- 1D Time classification or forecasting methods often take a windowed approach, making a prediction based on data from a fixed length of time
- Statistical models like (S)ARIMA are often preferred for time series forecasting due to their interpretability and controllability
- Audio is best represented in terms of amplitudes of frequency ranges over time
- Audio models use vision-based architectures on Mel Spectrograms





Coming up

- Submit your final project form before Thursday
- Guest Lecture -- Chenxi Yu from State Farm on Thursday
- Fall Break next week (no office hours 11/22 to 11/30)