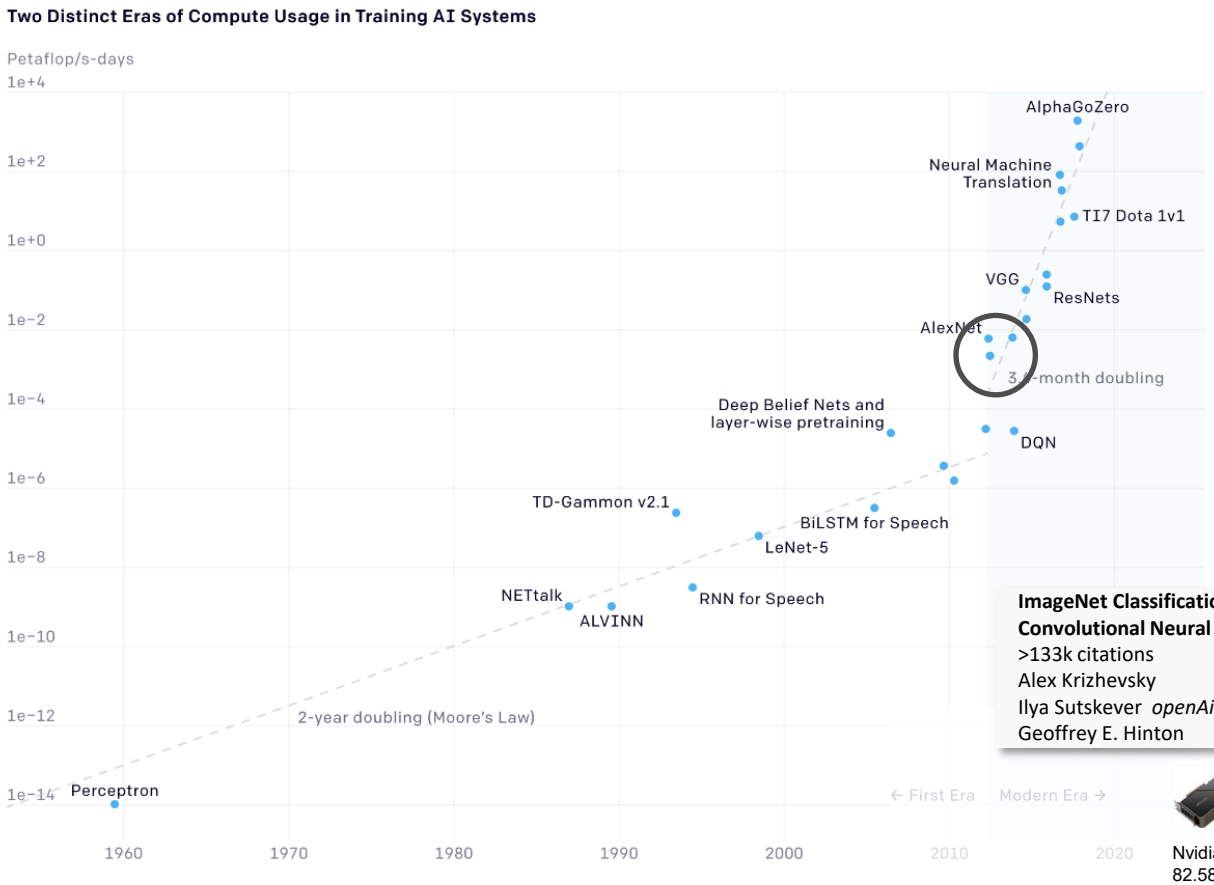
Deep Learning models for ASR

1. Feature extraction

Summary

- 1 Feature extraction
  - **Introduction**
  - Signal processing
    - Spectrogram
    - Melfilter bank
    - Cepstrum
  - Unsupervised learning
  - Robustness

Background



ImageNet Classification with Deep Convolutional Neural Networks
>133k citations
Alex Krizhevsky
Ilya Sutskever *openAi Chief Scientist (GPT)*
Geoffrey E. Hinton

Nvidia RTX 4090
82.58 TFLOPS

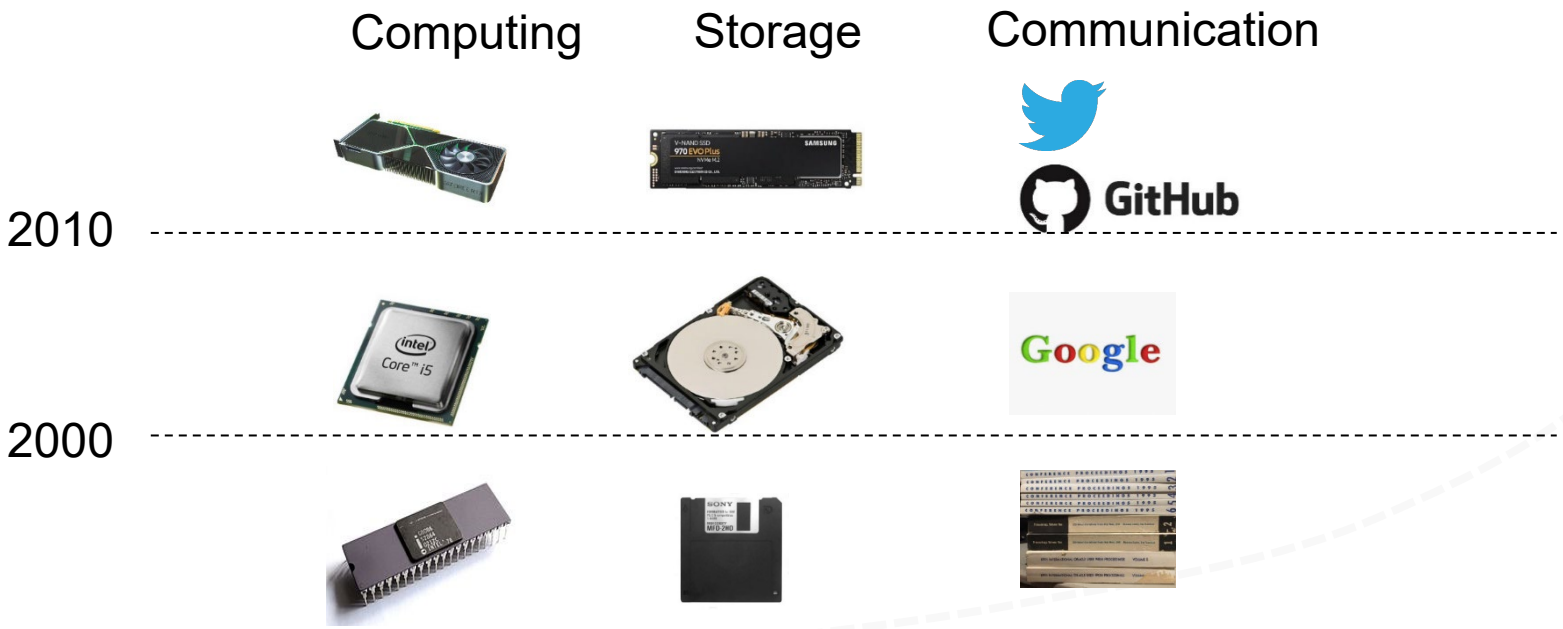
50 x 

10k x 

https://openai.com/research/ai-and-compute

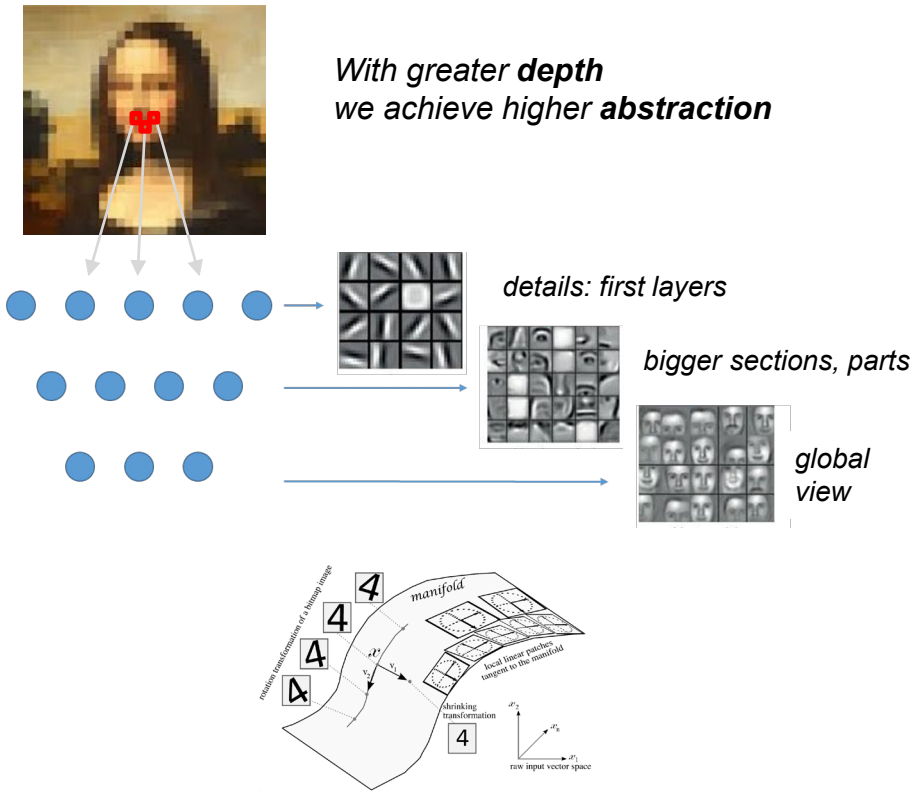
Background

- Evolution: technology and communication



Introduction

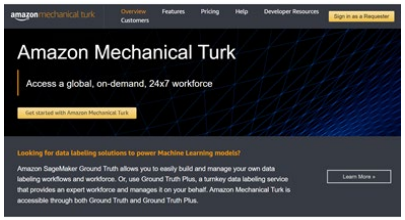
- Deep Learning



Bengio, Y. (2009). Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2(1), 1-127.

- Data and Labels
  - **Number of updates**
    - SGD is slow: depends on the difficulty of the task may be **millions of updates**
  - **Data and answers/labels: high cost**
    - Corpus, databases
    - Billions or millions of examples with their label
  - **Data bias problem**
    - If we show more times an example and the answer than other examples there will appear a **bias** in the system

We all label at some time, but there are specialized facilities and services like amazon mechanical turk

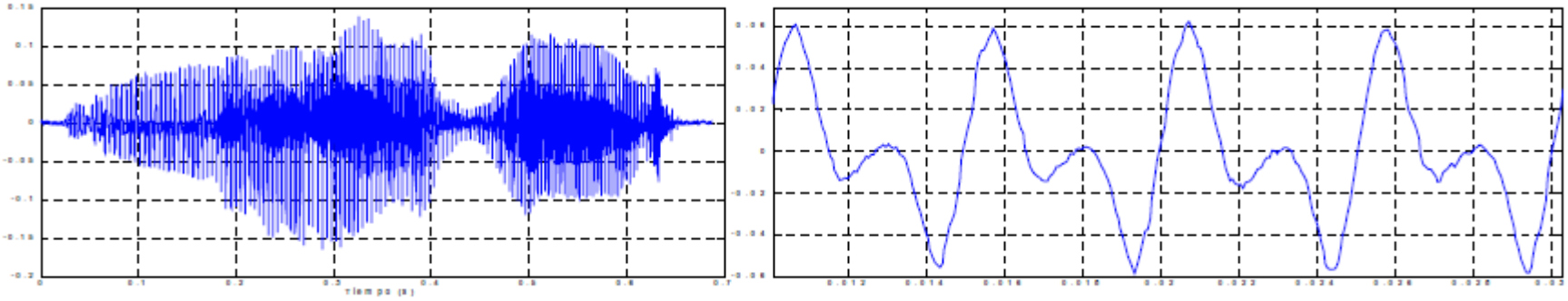


Summary

- 1 Feature extraction
  - Introduction
  - **Signal processing**
    - Spectrogram
    - Melfilter bank
    - Cepstrum
  - Unsupervised learning

Feature Extraction

- **Speech signals**
  - Non stationary signals. Examples: music, speech
  - Short-term stationary: Segments of miliseconds (20-30 ms)



  - Short-term analysis

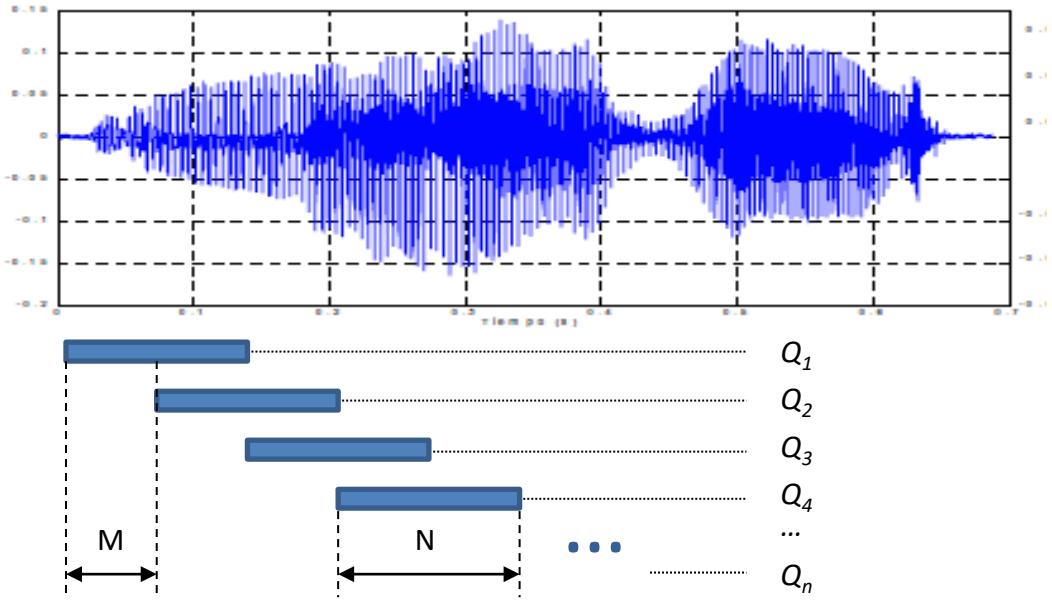
$$Q_n = \sum_{m=-\infty}^{\infty} T\{x[m]w[n-m]\}$$

Transform

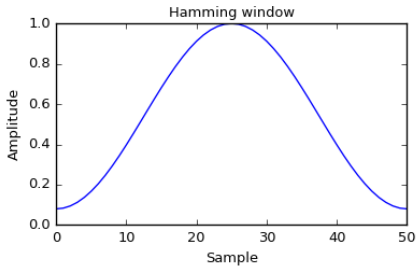
window

Short-term analysis

- **Basic configuration**



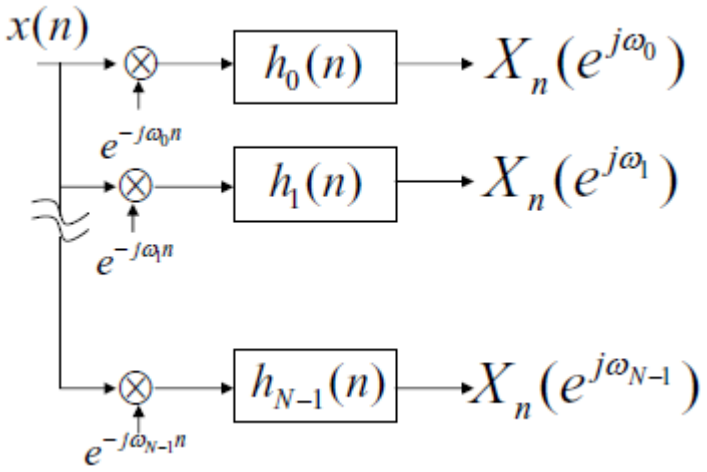
  - Short-term analysis
    - N: window length, (speech/music [25-30]ms)
    - M: hop length , (~10ms)
    - Window type w[n]: Rectangular, **Hamming**, Hanning,…

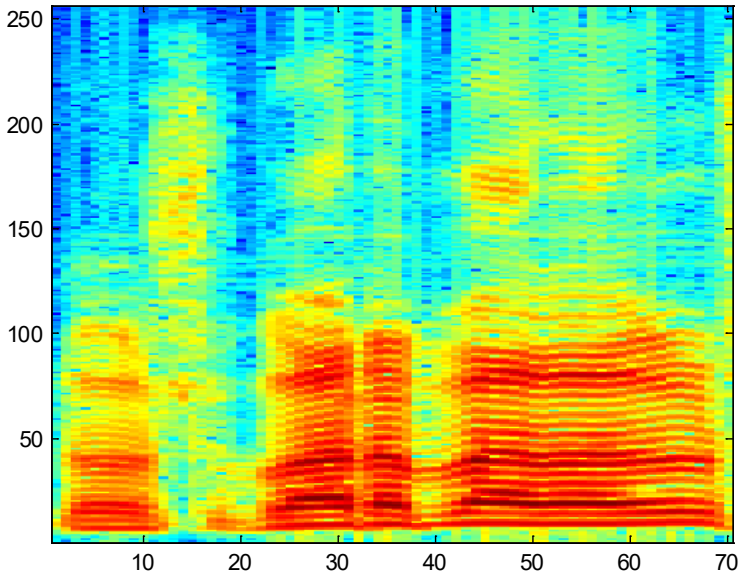


Short-term analysis

- **Short-term Fourier transform**
  - Also called *short-time frequency analysis*
    - *T{} is the FT, for fast implementations FFT*

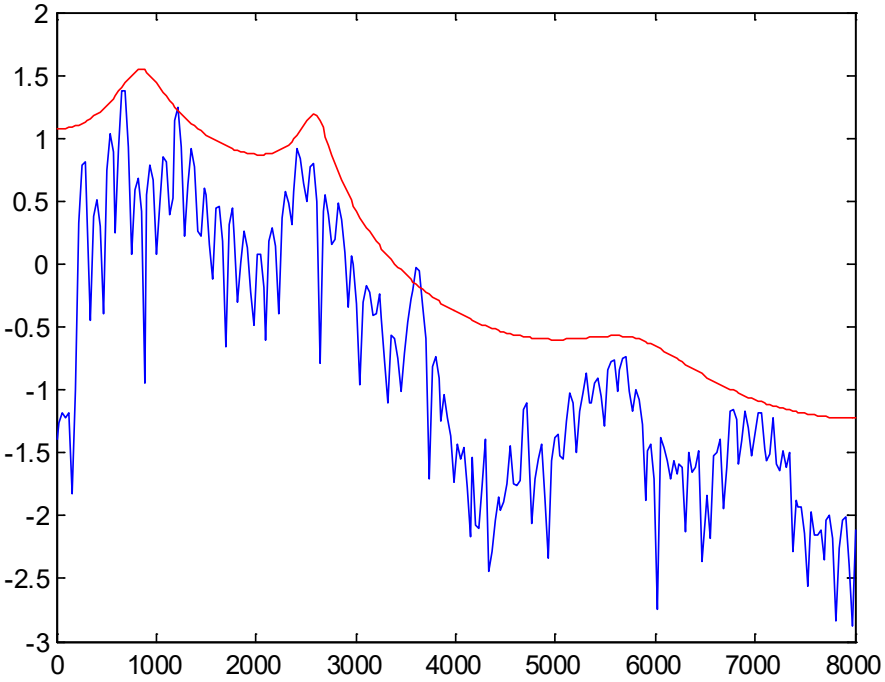
$$X_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]\,e^{-j\omega m}$$

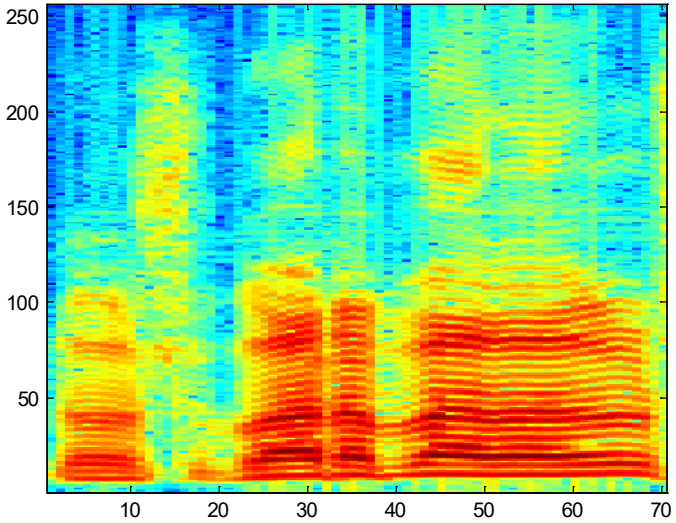


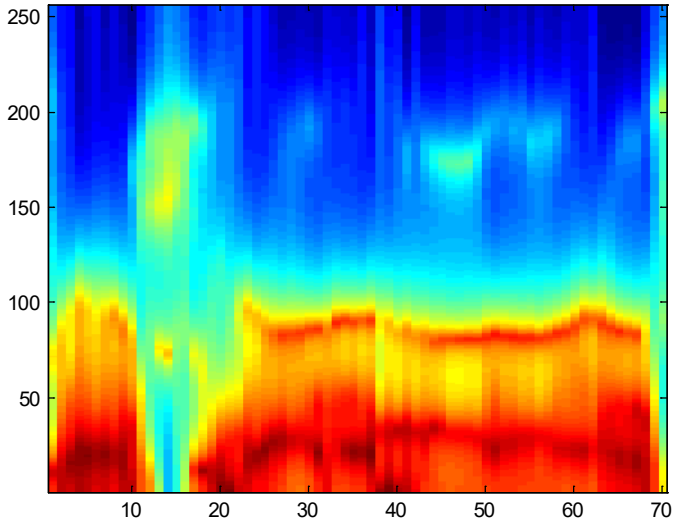


Short-term analysis

- **Linear prediction**
  - Model captures envelope of spectrogram

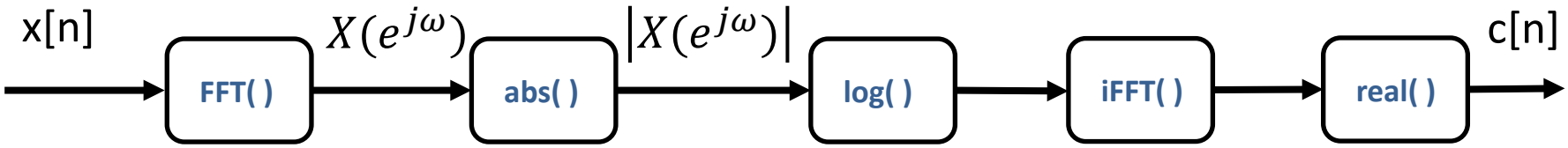






Short-term analysis

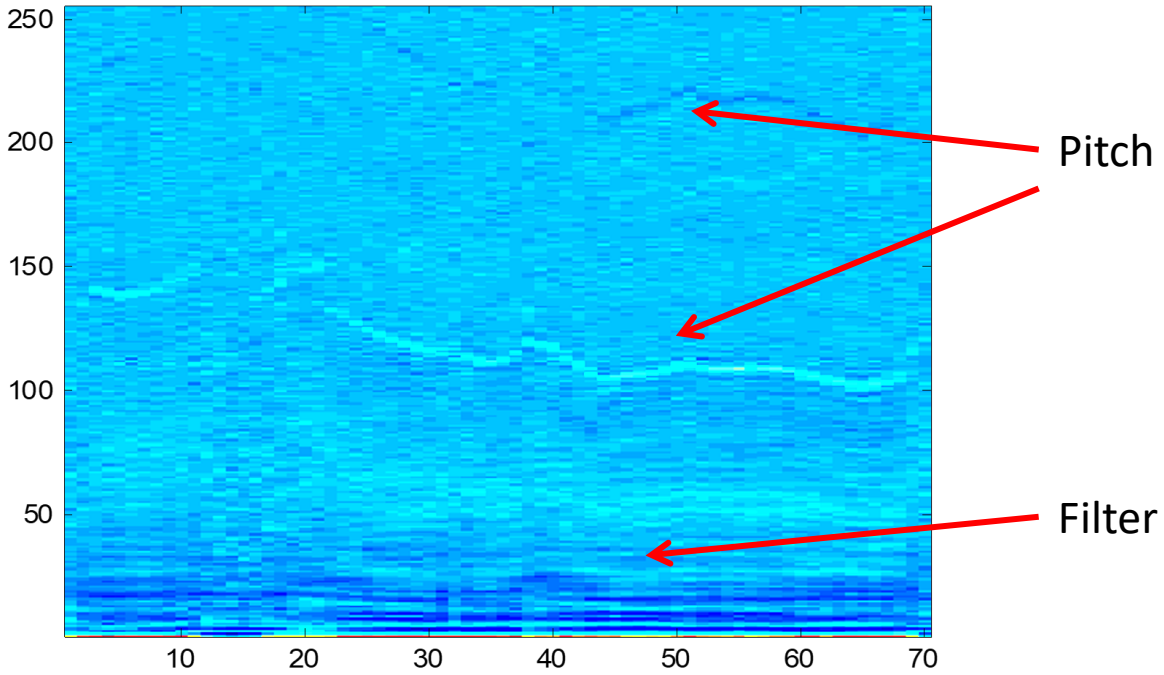
- **Cepstrum**
  - It is opbtained by applying log() and inverse FT to spectrogram
  - Properties:
    - Convolutions -> additions
    - Speech signal: filter + excitation (pitch)
    - Simple channel compensation > **cepstrum mean substraction**



  - Cepstrum related terms:
    - spectrum -> cepstrum
    - filter -> lifter
    - frequency -> quefrency (or cepstrum coefficient)

Short-term analysis

- **Cepstrum**

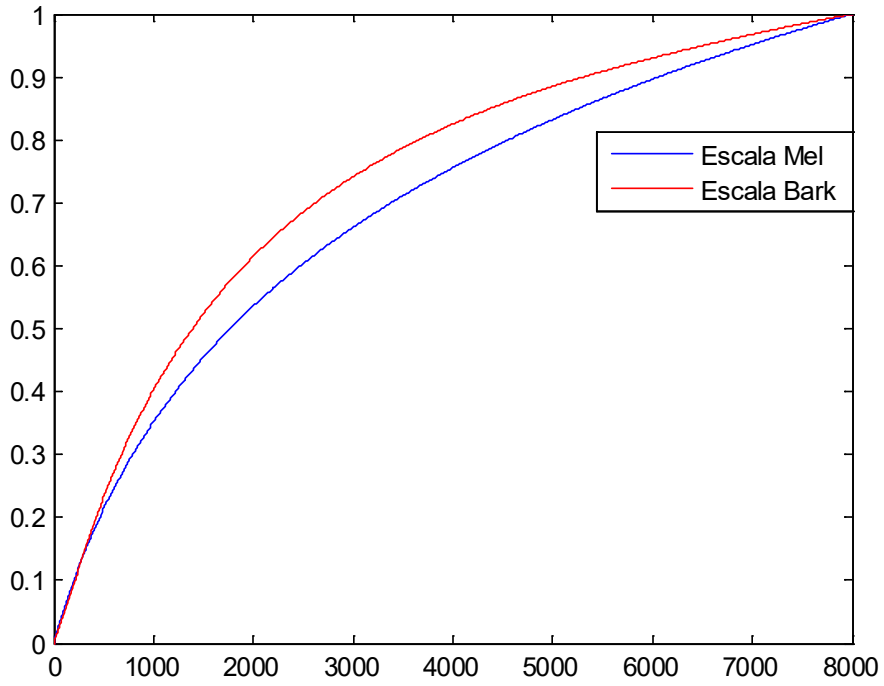


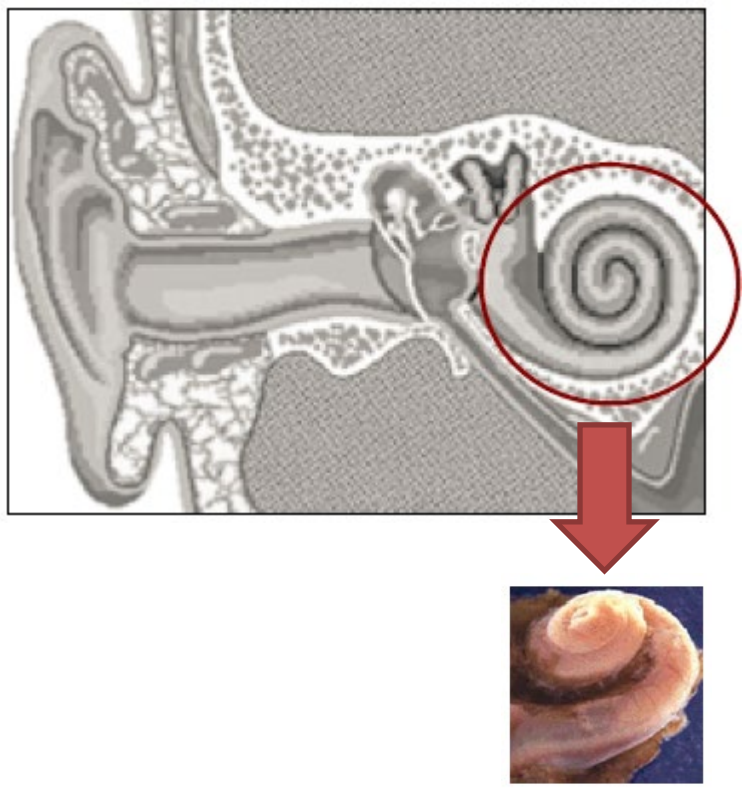
- Filter information (vocal tract): lower values

Short-term analysis

- **Mel-Cepstrum localizado**
  - Perceptual scale: Mel, Bark

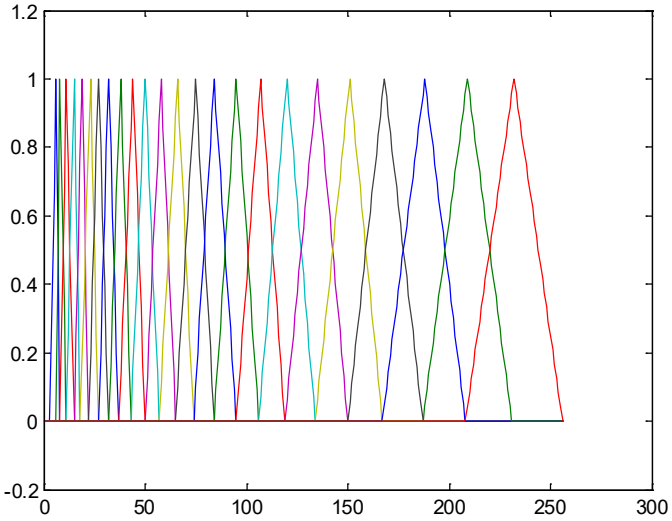
$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

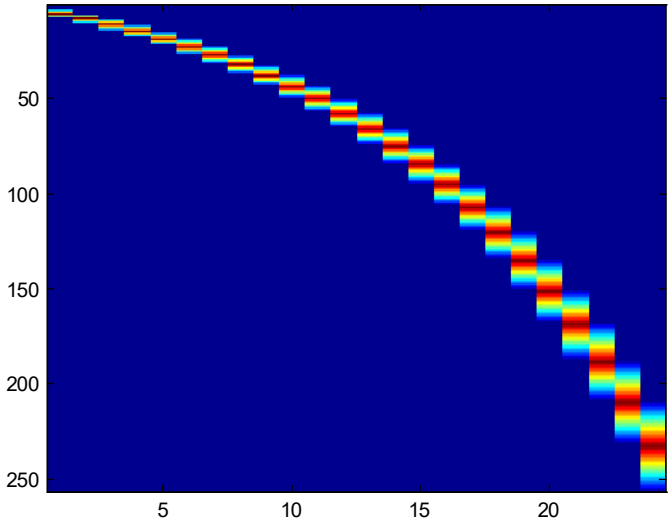




Short-term analysis

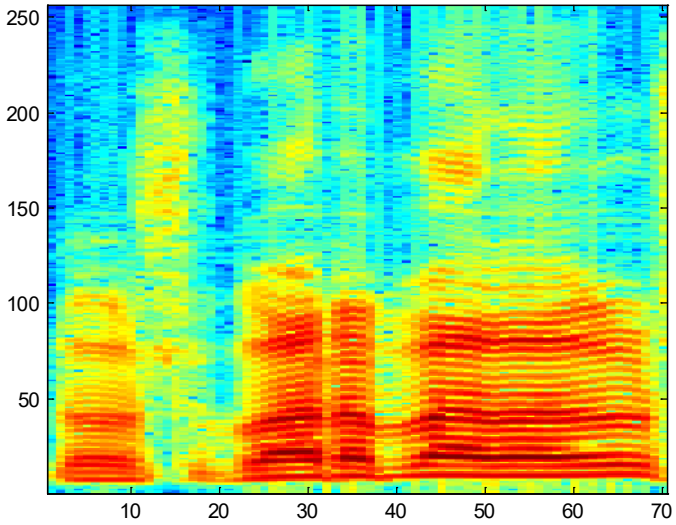
- **Mel-Cepstrum**
  - Mel filterbank
    - Can be calculated as a matrix multiplication
      - size F x B
      - F = NFFT/2
      - B = Filters / Critical bands: ~ 24



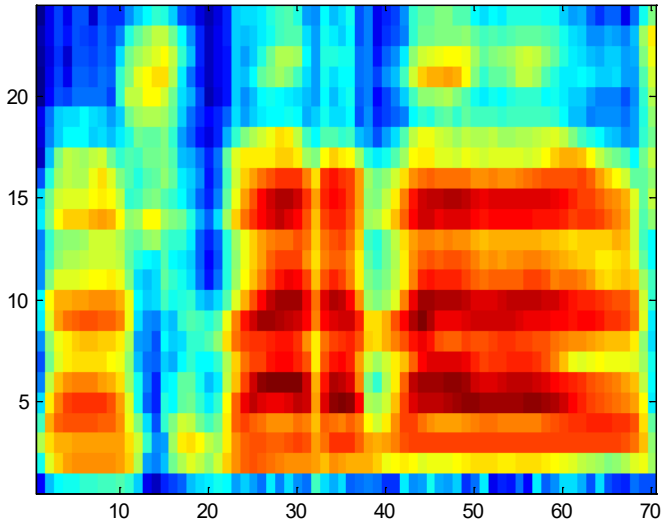


Short-term analysis

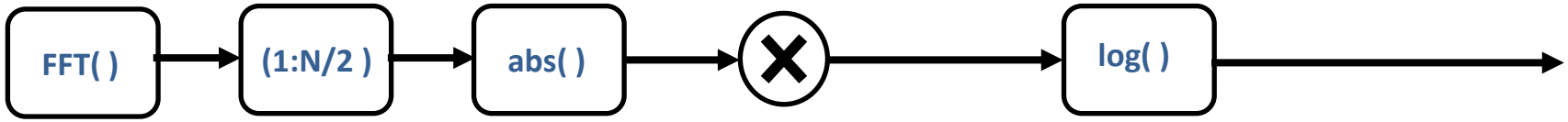
- **Mel-Cepstrum**

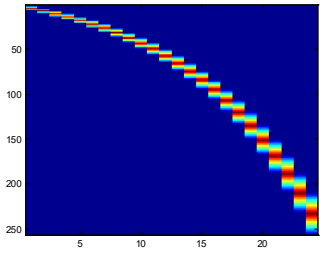


(absolute FT)



(log mel fb)

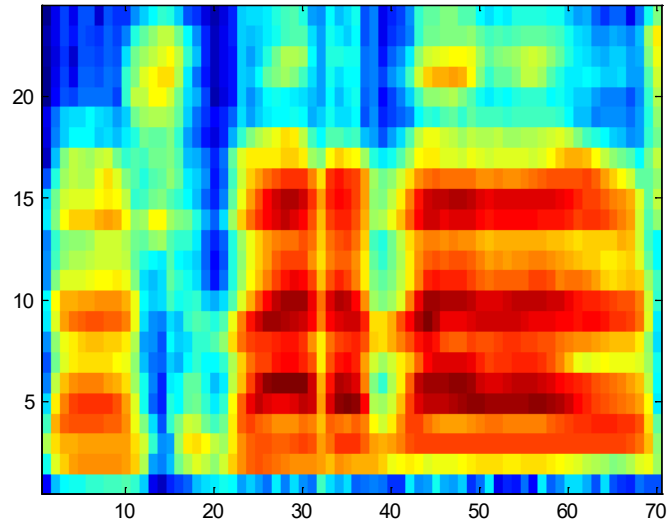




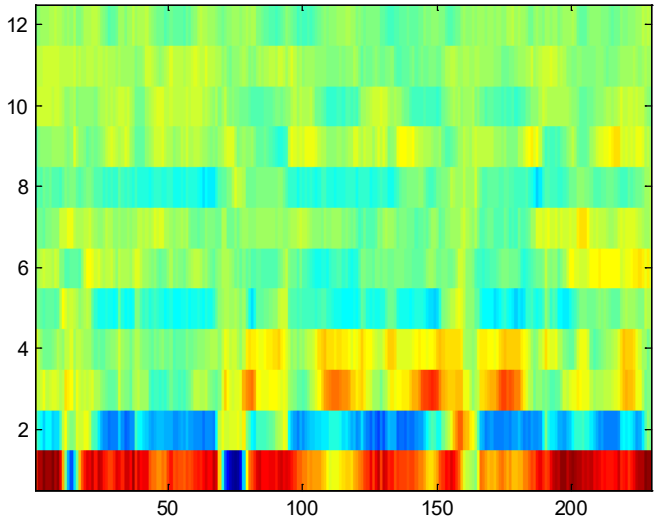
filterbank

Short-term analysis

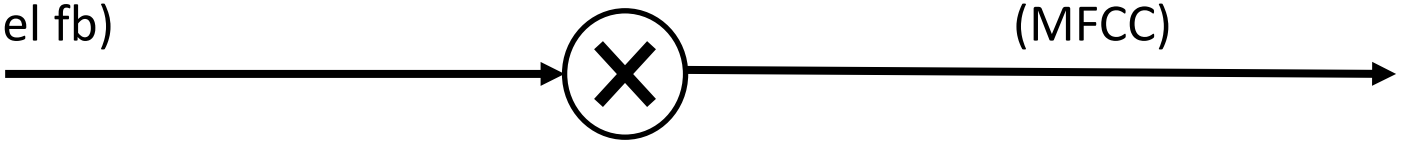
- **Mel-Cepstrum localizado**

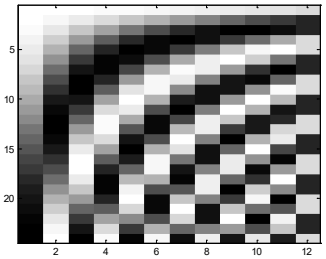


(log mel fb)



(MFCC)





DCT (truncated)

Summary

- 1 Feature extraction
  - Introduction
  - Signal processing
    - Spectrogram
    - Melfilter bank
    - Cepstrum
  - **Unsupervised learning**
  - Robustness

Unsupervised learning

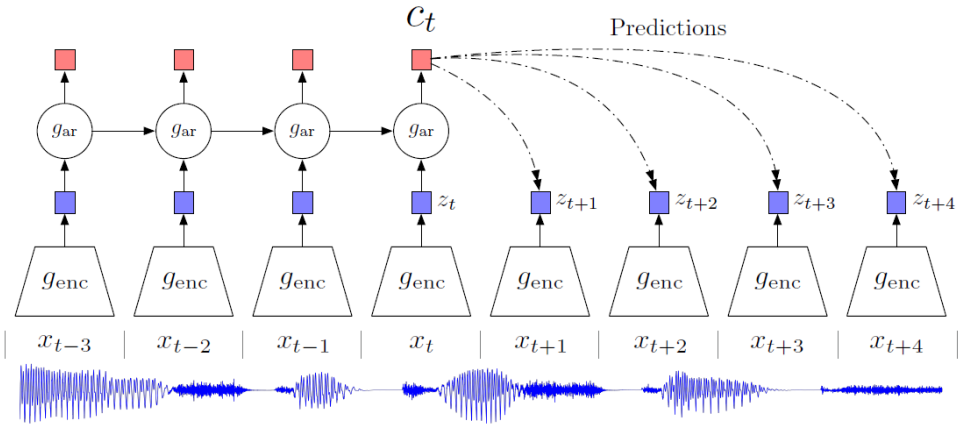
- **Contrastive loss**
- **CPC** (Oord **2018**)

Oord, A. V. D., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748

- **Wav2vec** (Schneider **2019**)

Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. arXiv preprint arXiv:1904.05862.

- Two stages are combined to learn a feature extractor, input **raw samples** at 16kHz
  - CNN: convolutional network designed to process raw simples
    - The strides are: 5, 4, 2, 2, 2 (product = 160 global downsampling size for 16kHz)
    - The kernel sizes: 10, 8, 4, 4, 4 to approximate 30ms of equivalent reception field
  - RNN: a GRU is used to generate a context embedding, that will be used to make predictions



Unsupervised learning

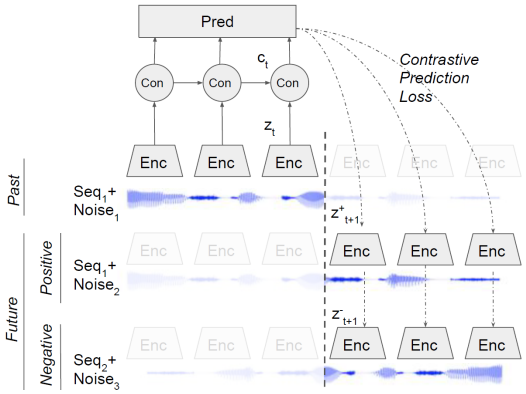
- **Contrastive loss**
- **CPC** (Oord **2018**)

Oord, A. V. D., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748

- **Wav2vec** (Schneider **2019**)

Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. arXiv preprint arXiv:1904.05862.

- Contrastive loss, learns to select ground truth continuation of the audio, among many alternatives (distractors)
  - In theory any audio segment, for efficiency in the same minibatch
    - number of distractors-> batch_size (limited in GPU memory)
  - Augmentations to make the task more difficult: positive match original audio vs. augmented audio

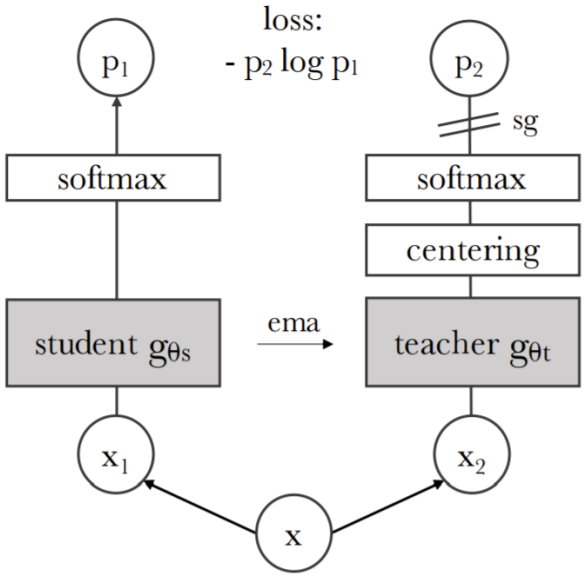


- Pretext task: **correct continuation**
  - Other pretext tasks, from image or text: same file? Correct order of fragments ?

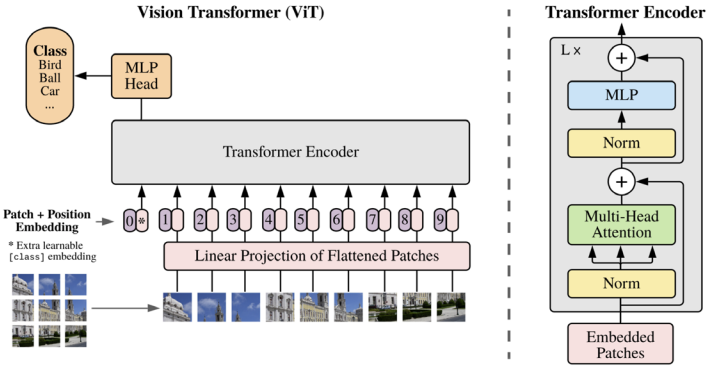
Unsupervised learning

- **Other methods**
- Compare pairs of examples and update a loss function to improve generalization of common representation
- **DINO**(Caron **2021**)

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. arXiv preprint arXiv:2104.14294..



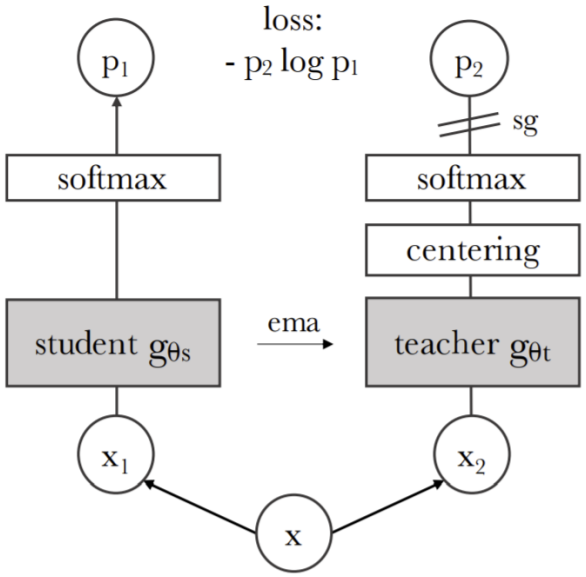
- Two networks teacher student:
  - T(x), S(x)
- ViT architecture, class token



Unsupervised learning

- **Other methods**
- Compare pairs of examples and update a loss function to improve generalization of common representation
- **DINO**(Caron **2021**)

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. arXiv preprint arXiv:2104.14294..



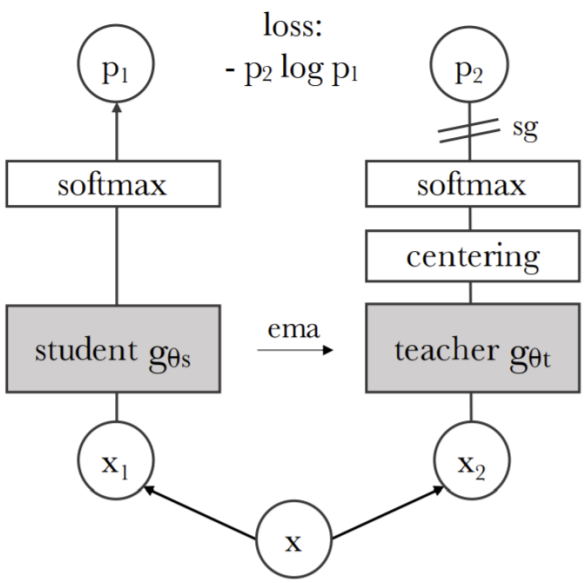
- Two networks T(x), S(x)
  - same arch. (ViT) diff. params
- Two augmentations: x1, x2

- The output of the teacher network T(x2)
- is centered with a mean computed over the batch.
- Each networks outputs a K dim
  - **softmax** (discrete distrib)
- Loss: cross-entropy loss, measure similarity distrib
- The teacher is not updated: stop-gradient (sg)
- The teacher exponential moving average (ema) of the student net

Unsupervised learning

- **DINO**(Caron **2021**)

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. arXiv preprint arXiv:2104.14294..



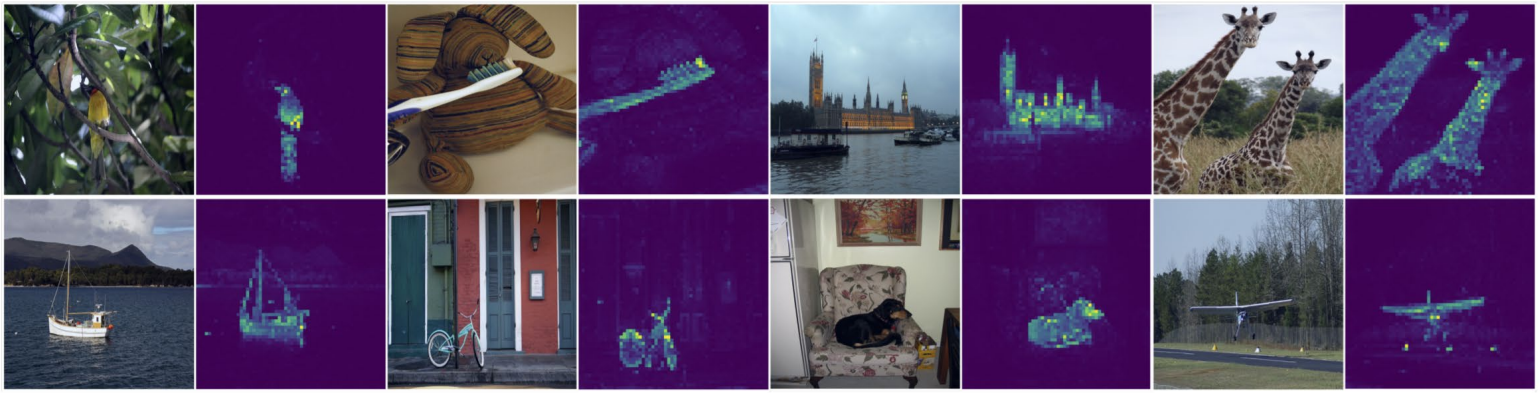
Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

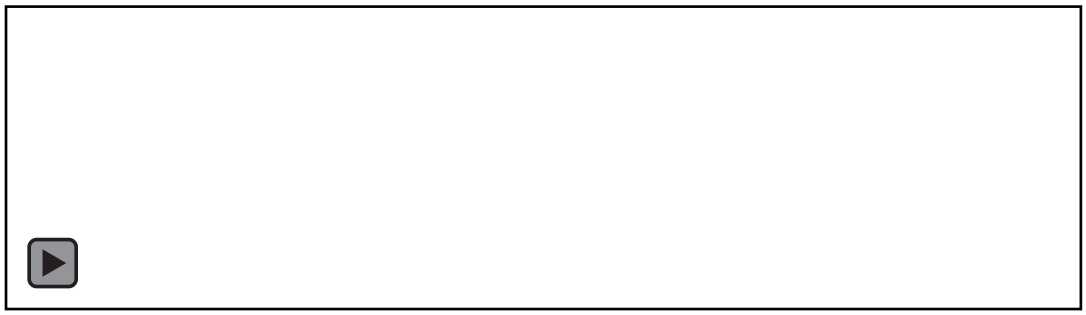
Unsupervised learning

- **Other methods**
- Compare pairs of examples and update a loss function to improve generalization of common representation
- **DINO**(Caron **2021**)

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. arXiv preprint arXiv:2104.14294.



Method	Arch.	Param.	im/s	Linear	k-NN
Comparison across architectures					
SCLR [12]	RN50w4	375	117	76.8	69.3
SwAV [10]	RN50w2	93	384	77.3	67.3
BYOL [30]	RN50w2	93	384	77.4	–
DINO	ViT-B/16	85	312	78.2	76.1
SwAV [10]	RN50w5	586	76	78.5	67.1
BYOL [30]	RN50w4	375	117	78.6	–
BYOL [30]	RN200w2	250	123	79.6	73.9
DINO	ViT-S/8	21	180	79.7	**78.3**
SCLRv2 [13]	RN152w3+SK	794	46	79.8	73.1
DINO	ViT-B/8	85	63	**80.1**	77.4



Unsupervised learning

- **Reconstruction methods**
- Recent uses of reconstruction methods for unsupervised representation learning

- **Decoar** (Ling **2020**)

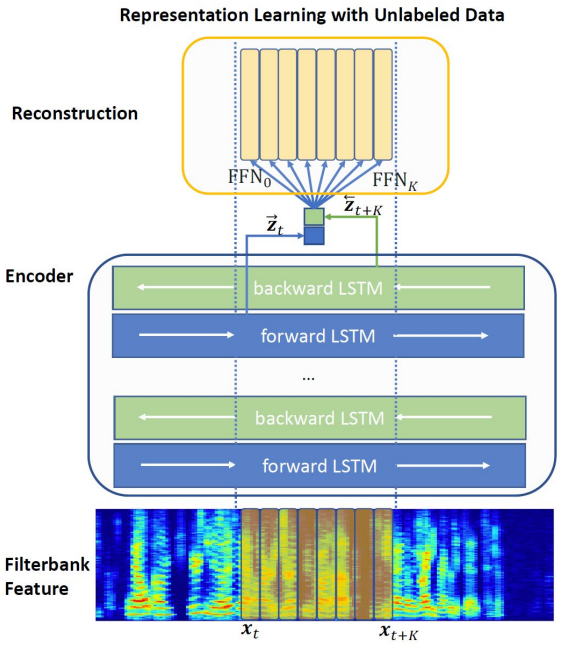
Ling, S., Liu, Y., Salazar, J., & Kirchhoff, K. (2020, May). Deep contextualized acoustic representations for semi-supervised speech recognition. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6429-6433). IEEE

- It also uses a L1 reconstruction to predict missing parts of spectrogram
- Bidirectional RNNs are used:
  - Direct and reverse representations are used to predict
  - **Forward** is used to predict K **frames** from: x_t
  - **Backward** is used to predict K **frames** before $x_t + K$
  - The loss is the sum of the prediction loss of future K frames

$$\mathcal{L}_t = \sum_{i=0}^{K} |\mathbf{x}_{t+i} - \mathrm{FFN}_i([\overrightarrow{\mathbf{z}}_t; \overleftarrow{\mathbf{z}}_{t+K}])|$$

  - A small network (different) is used to predict each future frame
    - (same embedding)

$$\mathrm{FFN}_i(\mathbf{v}) = \mathbf{W}_{i,2}\mathrm{ReLU}(\mathbf{W}_{i,1}\mathbf{v} + \mathbf{b}_{i,1}) + \mathbf{b}_{i,2}$$



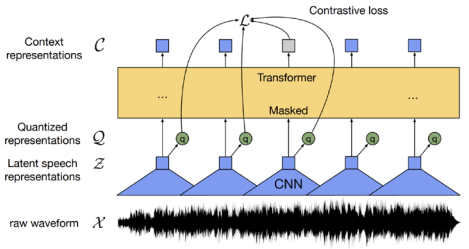
Unsupervised learning

- **Prediction of clusters/vq centroids**
- The signal is assigned to discrete variables: cluster ids, centroids

- **Wav2vec2.0** (Ling **2020**)

Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. arXiv preprint arXiv:2006.11477.

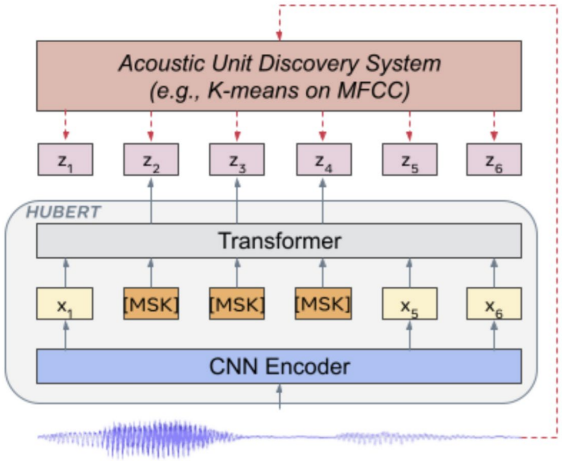
- The centroids of a quantification process are the objective (discrete)
- The training objective requires identifying:
  - the **correct quantized latent audio representation** in a set of distractors
- Improve ASR



- **Hubert** (Hsu **2021**), **Wavlm** (Hsu **2021**)

Hsu, W. N., Bolte, B., Tsai, Y. H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). HuBE Representation Learning by Masked Prediction of Hidden Units. arXiv preprint arXiv:2106.07447.
Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., ... & Wei, F. (2022). Wavlm: Large-scale s stack speech processing. IEEE Journal of Selected Topics in Signal Processing, 16(6), 1505-1518

- Iterative clustering -> index of clusters
- Prediction with large transformed (masked), similar to BERT style
  - First iteration: kmeans of standard features, Mel cepstrum
  - Folloing iterations: kmeans of last iteration embeddings
- Improvements on many tasks: ASR, speaker id, emotion recognition
- All these models can be downloaded

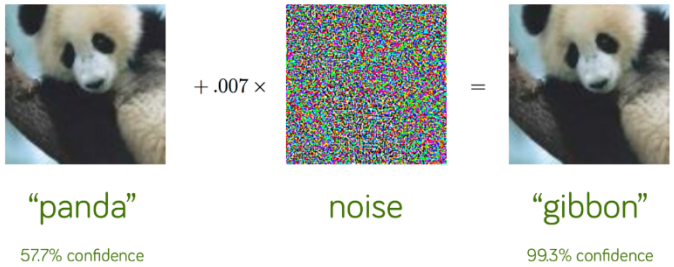


Summary

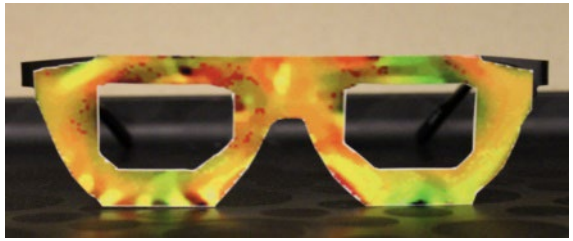
- Feature extraction
  - Signal processing
    - Spectrogram
    - Melfilter bank
    - Cepstrum
  - Unsupervised learning
  - **Robustness**

Robustness

- **Adversarial attacks**
- **A subtle change in an image can make the network change the prediction**



Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.



Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition
Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter ACM Conference on Computer and Communications Security (CCS 2016)

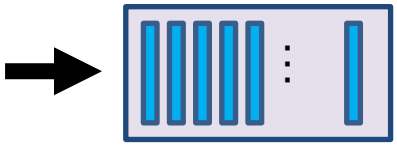
Speed Limit 80
(88% confidence)

Robust physical-world attacks on deep learning visual classification.
Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., ... & Song, D. (2018). In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1625-1634).

Augmentation

- **Data augmentation**
- **Increase artificially the dataset by applying transformations to the inputs**
  - Transformations depend on the task:
    - Image: rotations, occlusions, scaling, random crops, color transformations
    - Audio: adding noise, reverberation
- **The system becomes more robust**
  - It is exposed to more variability during training
  - The augmentations can also be applied to predictions (TTA test time augmentations)

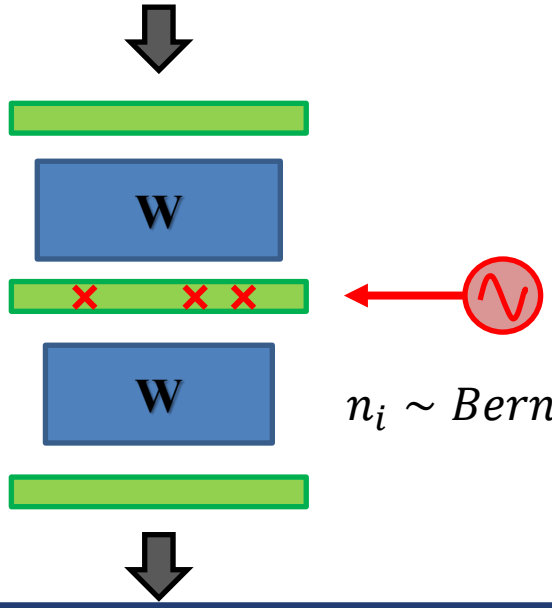
paperswithcode.com



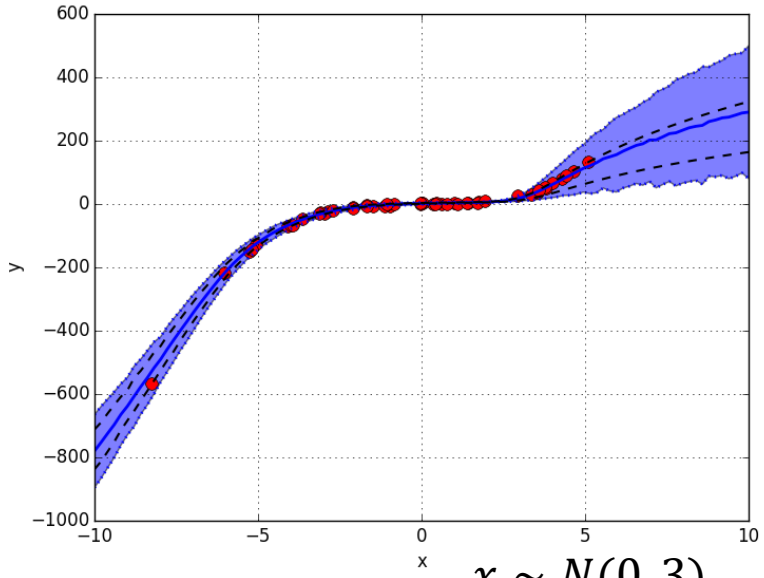
Train/Predict

Dropout

- **Dropout (Hinton 2012)**
  - **One of the many recent techniques to avoid overfitting in neural networks**
  - **Deactivates randomly some neurons forcing the remaining to solve the problem**
  - **Can also be applied training/predictions (TTA)**



$$n_i \sim Berno(1-p)$$



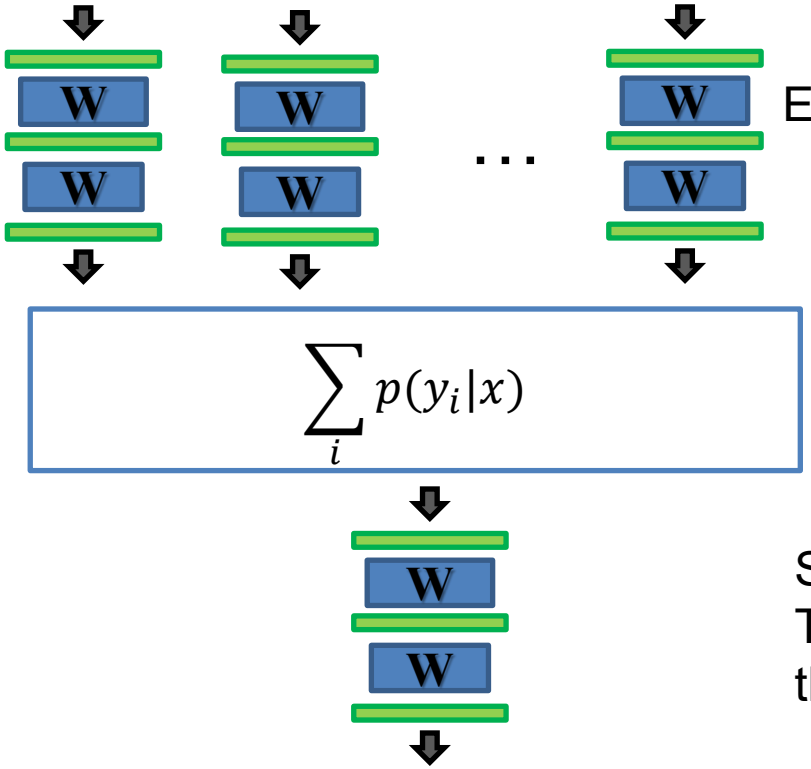
Artificial data example:
1D regression

$$x \sim N(0,3)$$
$$\varepsilon \sim N(0,3)$$
$$y = x^3 + \varepsilon$$

Bayesian dark knowledge

- **Bayesian dark knowledge(Korattikara 2015)**

Korattikara Balan, A., Rathod, V., Murphy, K. P., & Welling, M. (2015). Bayesian dark knowledge. Advances in Neural Information Processing Systems, 28, 3438-3446..



Ensemble of teacher networks

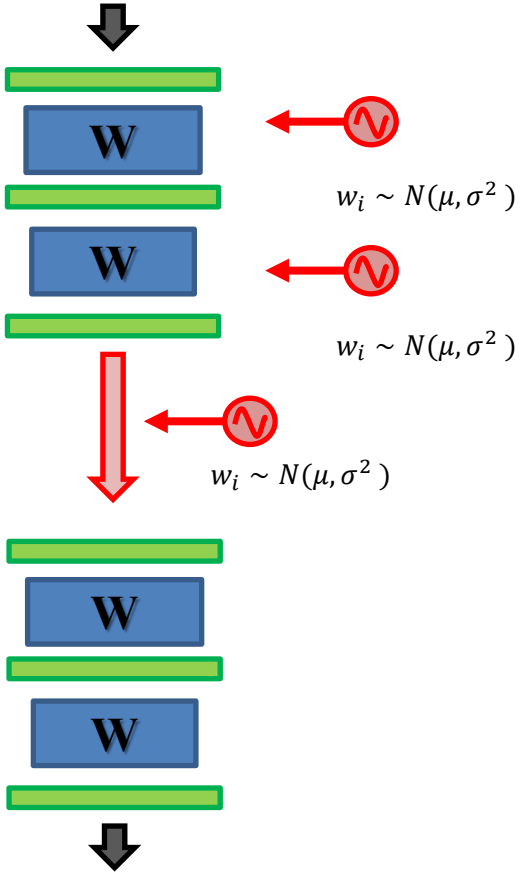
Student network
The student network learned
the ensemble of networks

Bayesian dark knowledge

- **Bayesian dark knowledge(Korattikara 2015)**

Korattikara Balan, A., Rathod, V., Murphy, K. P., & Welling, M. (2015). Bayesian dark knowledge. Advances in Neural Information Processing Systems, 28, 3438-3446..

Teacher network, sampling

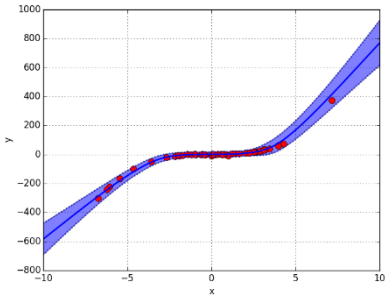


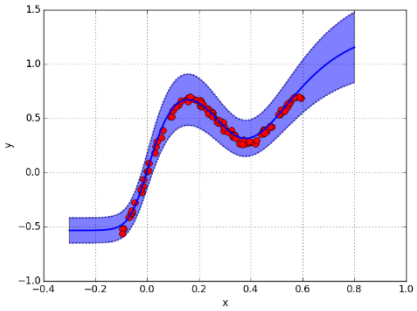
Student network, learns the variability

Noisy student in ASR (2020)

Park, D. S., Zhang, Y., Jia, Y., Han, W., Chiu, C. C., Li, B., ... & Le, Q. V. (2020). Improved noisy student training for automatic speech recognition. *arXiv preprint arXiv:2005.09629*.

Examples regression



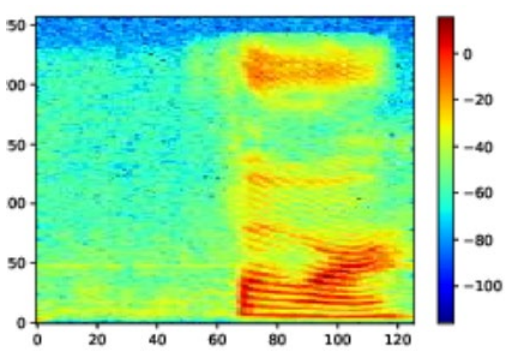


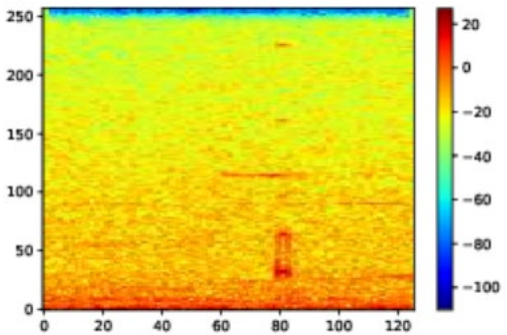
Speech tasks augmentations

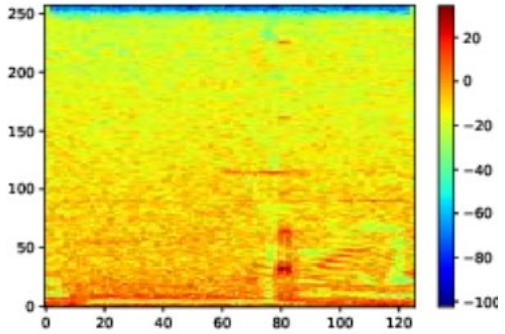
- **Additive Noise**
- **Augmentation**
  - MUSAN, noises and music from Internet: https://www.openslr.org/17/

Snyder, D., Chen, G., & Povey, D. (2015). Musan: A music, speech, and noise corpus. arXiv preprint arXiv:1510.08484.

  - A collection of noises, the user decides the SNR randomly to add the noise.
  - Steps:
    - Calculate the power of the signal: Ps = x.std()**2
    - Calculate the power of the noise: Pn = n.std()**2
    - Sample a random SNR in dBs for example in the range [-5, 20] uniformly
    - Scale the noise so that the SNR is the desired : SNRdb = 10*log10(Px / Pn)



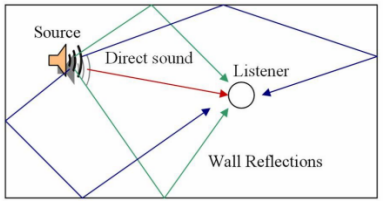


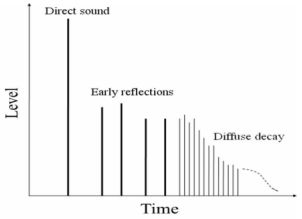


Importantaug: A Data Augmentation Agent for Speech

Speech tasks augmentations

- **RIR**
  - **Databases available: RIR_noises https://www.openslr.org/28/**
  - **Simulated rooms**





Auditory Room Size Perception for Real Rooms

Impulse response

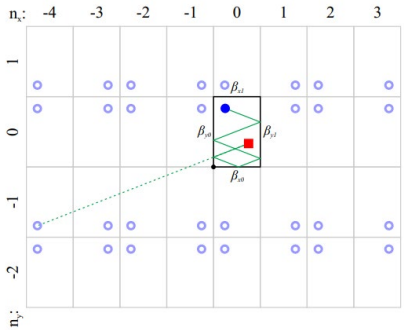
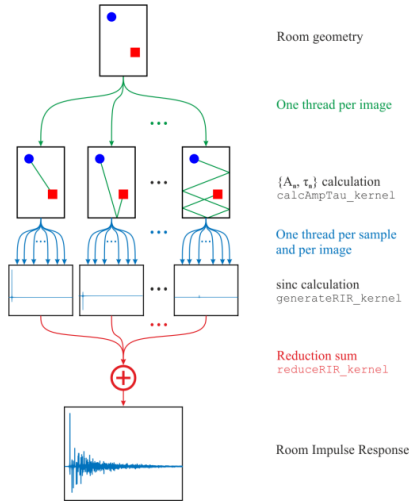


Image method



GPU implementation

- **Libraries:**
  - **GPURIR**

    Diaz-Guerra, D., Miguel, A., & Beltran, J. R. (2021). gpuRIR: A python library for room impulse response simulation with GPU acceleration. Multimedia Tools and Applications, 80, 5653-5671.

    https://github.com/DavidDiazGuerra/gpuRIR

  - **pyroomacoustics**

    https://github.com/LCAV/pyroomacoustics

  - **RIR-generator**

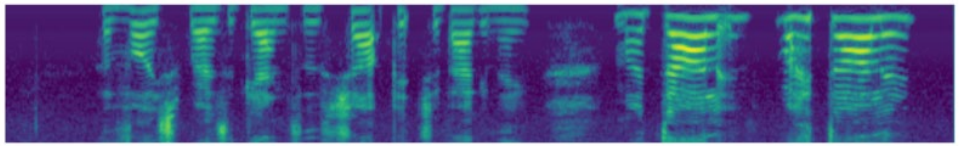
    https://github.com/ty274/rir-generator

SpecAugment

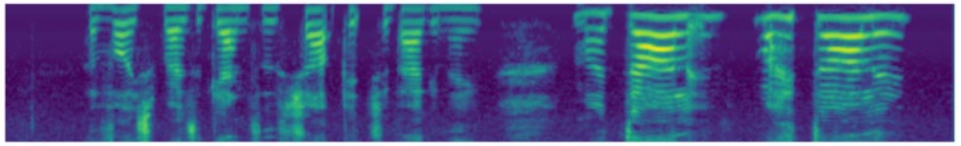
- **SpecAugment (Park 2020)**

Park, D. S., Zhang, Y., Chiu, C. C., Chen, Y., Li, B., Chan, W., ... & Wu, Y. (2020, May). Specaugment on large scale datasets. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6879-6883). IEEE..

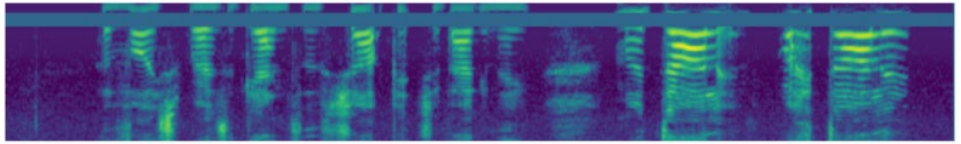
Original log-mel fb



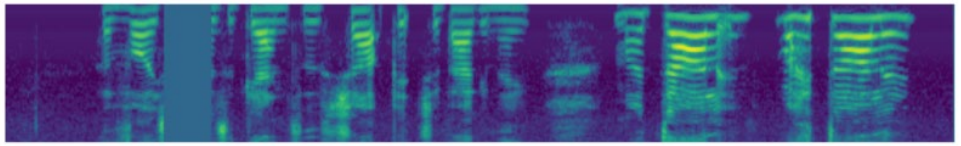
Time warp



Freq mask



Time mask



Masks applied to image/sequences:

cutout *DeVries, T., & Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552.*

BERT *Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805..*