Boulder

Speech & Music; Modeling

YouTube Playlist

Maziar Raissi

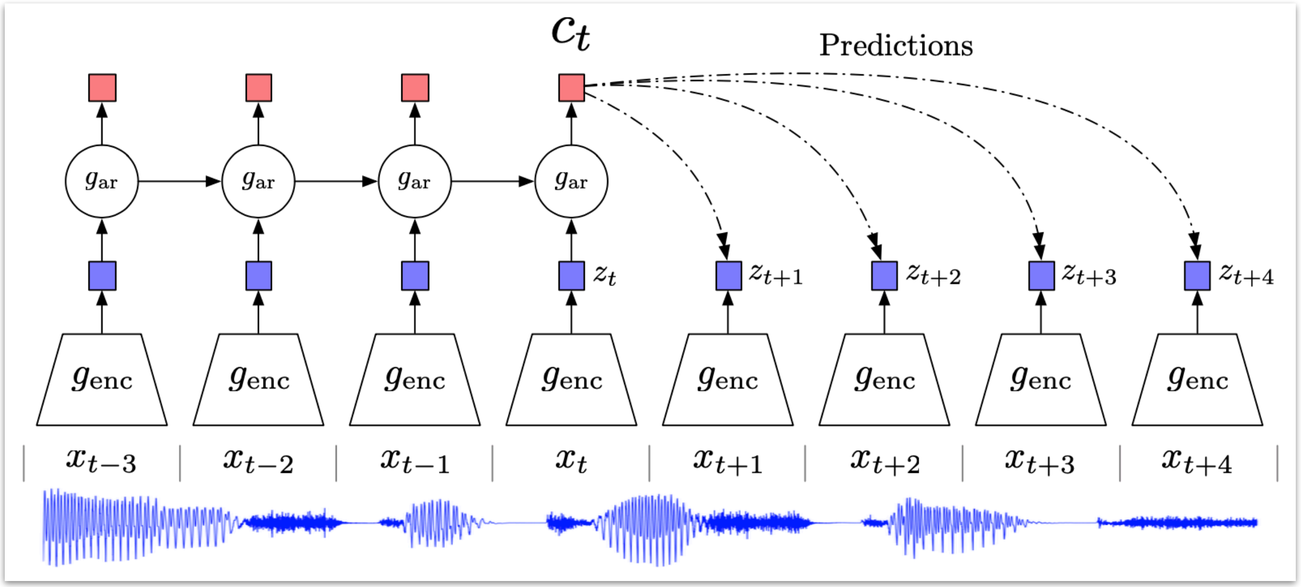
Assistant Professor

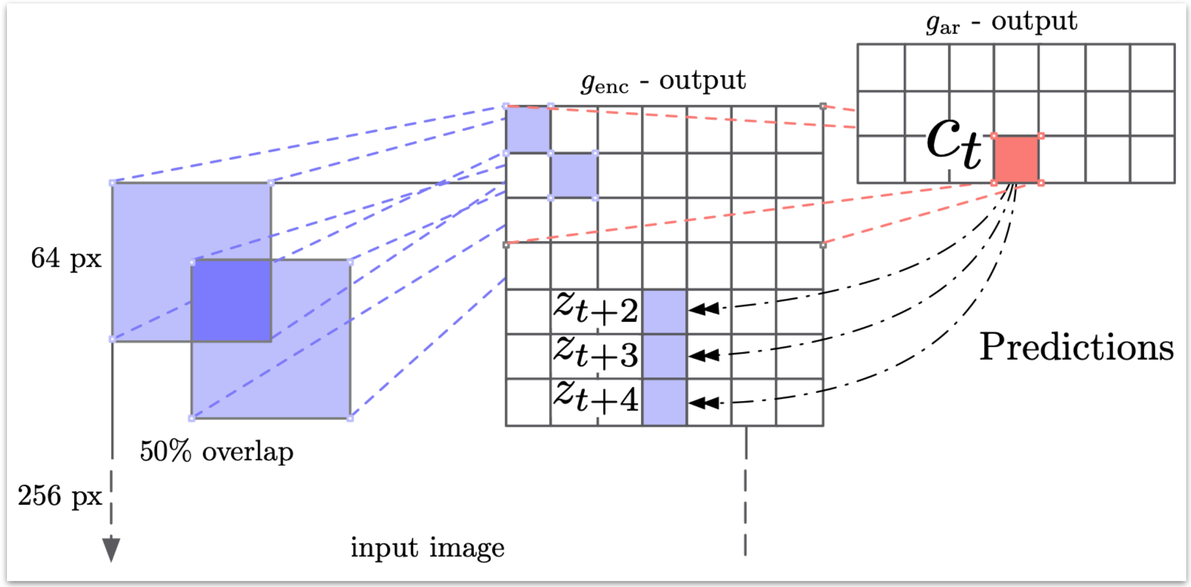
Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

Representation Learning with Contrastive Predictive Coding



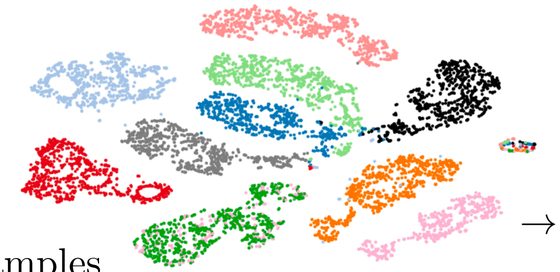


$g_{enc} \rightarrow$ non-linear encoder (e.g., strided convolutional layers with resnet blocks)
$x_t \rightarrow$ input observation
$z_t = g_{enc}(x_t) \rightarrow$ latent representation
$g_{ar} \rightarrow$ autoregressive model (e.g., GRUs)
$c_t = g_{ar}(z_{\leq t}) \rightarrow$ context latent representation (summarizing all $z_{\leq t}$ in the latent space)
$f_k(x_{t+k}, c_t) := \exp(z_{t+k}^T \underbrace{W_k c_t}_{\hat{z}_{t+k}})$

InfoNCE Loss

$$\mathcal{L}_N = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

$X = \{x_1, x_2, \ldots, x_N\} \rightarrow$ set of N random samples



$\rightarrow$ t-SNE visualization of speech (each color represents a different speaker)
Speech, images, text and reinforcement learning!

Containing one positive sample from $p(x_{t+k}|c_t)$ and $N-1$ negative samples from the proposal distribution $p(x_{t+k})$

Mutual Information Estimation

The optimal value for $f(x_{t+k}, c_t)$ is proportional to $\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$!

No need to predict future observations x_{t+k} directly with a generative model $p(x_{t+k}|c_t)$. InfoNCE (Noise Contrastive Estimation) relieves the model from modeling the high dimensional distributions x_{t+k}.

$I(x_{t+k}; c_t) \geq \log N - \mathcal{L}_N$

$$I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x|c)}{p(x)} \rightarrow \text{mutual information}$$

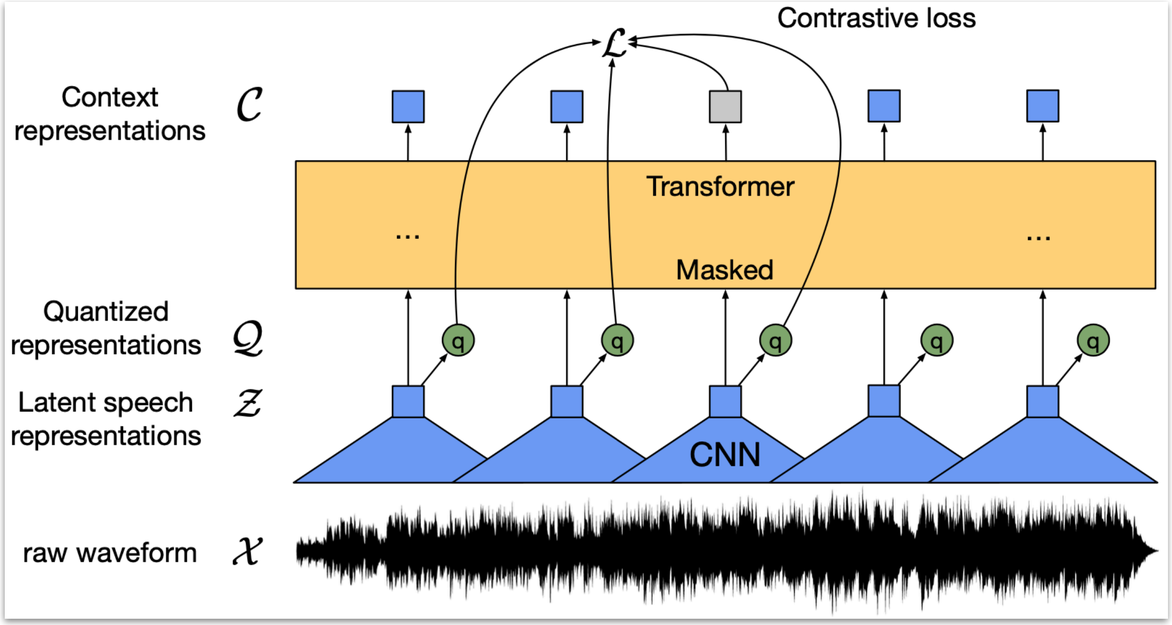
Minimizing $\mathcal{L}_N$ implies maximizing $I(x_{t+k}; c_t)$.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv preprint arXiv:1807.03748* (2018).

wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations

Boulder

YouTube Video



$f : \mathcal{X} \mapsto \mathcal{Z}$
↳ multi-layer convolutional feature encoder
$\mathcal{X} \rightarrow$ input raw audio
$\mathcal{Z} = (z_1, z_2, \ldots, z_T) \rightarrow$ latent speech representations
$g : \mathcal{Z} \mapsto \mathcal{C}$
↳ transformer
$\mathcal{C} = (c_1, c_2, \ldots, c_T) \rightarrow$ representations capturing information from the entire sequence

Instead of fixed positional embeddings which encode absolute positional information, use a convolutional layer which acts as relative positional embedding.

$\mathcal{Z} \mapsto \mathcal{Q}$
↳ quantization module
$\mathcal{Q} = (q_1, q_2, \ldots, q_T)$

diversity loss: encourage the model to use the codebook entries equally often $\leftarrow$

Quantization Module

$G \rightarrow$ number of codebooks/groups
$V \rightarrow$ number of entries
$e \in \mathbb{R}^{V \times d/G}$

Choose one entry/row from each codebook e and concatenate the resulting vectors $e_1, \ldots, e_G$ and apply a linear transformation $\mathbb{R}^d \rightarrow \mathbb{R}^f$ to obtain $q \in \mathbb{R}^f$.
The Gumbel softmax enables choosing discrete codebook entries in a fully differentiable way!
$z \mapsto l$
$z \rightarrow$ feature encoder output
$l \in \mathbb{R}^{G \times V} \rightarrow$ logits

$$p_{g,v} = \frac{\exp(l_{g,v} + n_v)/\tau}{\sum_{k=1}^{V} \exp(l_{g,k} + n_k)/\tau}$$

↳ probability of choosing the v-th codebook entry for group g
$\tau \rightarrow$ non-negative temperature
$n = -\log(-\log(u)) \rightarrow$ Gumbel noise $\quad u \sim U(0,1)$
Forward pass: $i = \arg\max_j p_{g,j} \rightarrow$ codeword i
Backward pass: true gradient of the Gumbel softmax outputs

	avg. WER	std.
Continuous inputs, quantized targets (Baseline)	7.97	0.02
Quantized inputs, quantized targets	12.18	0.41
Quantized inputs, continuous targets	11.18	0.16
Continuous inputs, continuous targets	8.58	0.08

TIMIT phoneme recognition accuracy in terms of phoneme error rate (PER).

	dev PER	test PER
CNN + TD-filterbanks [59]	15.6	18.0
PASE+ [47]	-	17.2
Li-GRU + fMLLR [46]	–	14.9
wav2vec [49]	12.9	14.7
vq-wav2vec [5]	9.6	11.6
This work (no LM) Connectionist Temporal Classification (CTC) loss		
LARGE (LS-960)	7.4	8.3

Pre-training

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)} \qquad sim(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T\mathbf{b}/\|\mathbf{a}\|\|\mathbf{b}\|$$

↳ contrastive loss: identify the true quantized latent speech representation for a masked time step within a set of distractors.

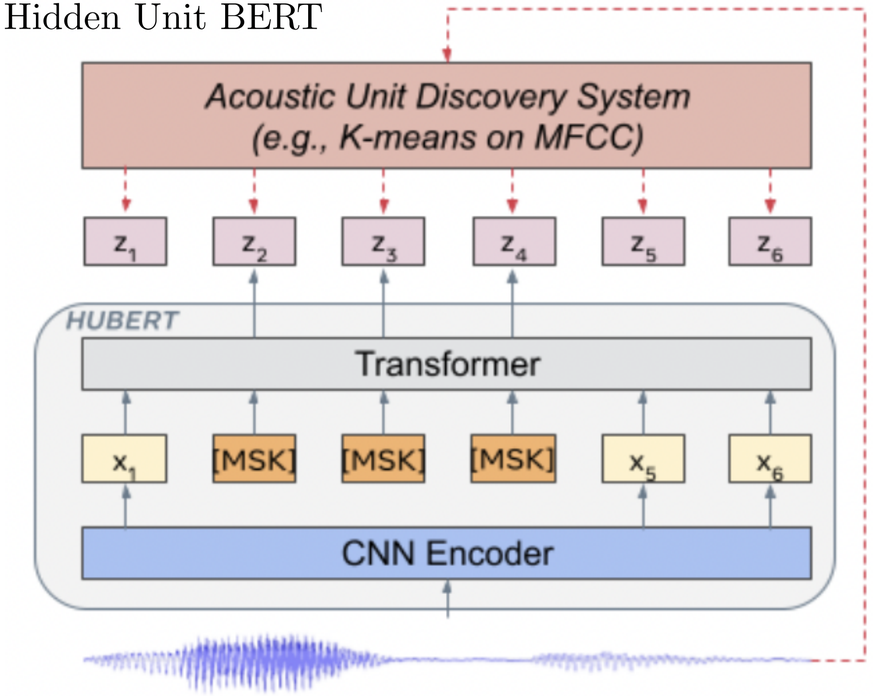
$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v} \qquad \mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d$$

Baevski, Alexei, et al. "wav2vec 2.0: A framework for self-supervised learning of speech representations." *arXiv preprint arXiv:2006.11477* (2020).

HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units

Boulder

Hidden Unit BERT



Acoustic unit discovery model

$X = [x_1, x_2, \ldots, x_T] \rightarrow$ speech utterance of T frames

MFCC features or latent features extracted from the HuBERT model pre-trained in the previous iteration (not fine-tuned) at some intermediate transformer layer

$h \rightarrow$ clustering model (e.g., k-means)

$h(X) = Z = [z_1, \ldots, z_T] \rightarrow$ discovered hidden units

$z_t \in \{1, \ldots, C\} \rightarrow C$-class categorical variable

Representation learning via masked prediction

$M \subset \{1, \ldots, T\} \rightarrow$ set of indices to be masked

$\tilde{X} = r(X, M) \rightarrow$ corrupted version of X

x_t is replaced with a mask $\tilde{x}$ if $t \in M$

$f : \tilde{X} \mapsto p_f(\cdot|\tilde{X}, t)$

f: masked prediction model; $p_f(\cdot|\tilde{X}, t)$: distribution over the target indices at each time step t

All 960 Hours of Labeled Librispeech Data

Model	Unlabeled Data	LM	dev-clean	dev-other	test-clean	test-other
Pre-Training						
wav2vec 2.0 LARGE [7]	LL-60k	Transformer	1.6	3.0	1.8	3.3
pre-trained Conformer XXL [41]	LL-60k	LSTM	1.5	3.0	1.5	3.1
This work (Pre-Training)						
HUBERT LARGE	LL-60k	Transformer	1.5	3.0	1.9	3.3
HUBERT X-LARGE	LL-60k	Transformer	1.5	2.5	1.8	2.9

How to mask? (similar to SpanBERT & wav2vec 2.0)

$p\%$ (e.g. 8%) of the time steps are randomly selected as start indices and spans of ℓ (e.g., 10%) steps are masked

Where to apply the prediction loss?

$$L_m(f; X, M, Z) = \sum_{t \in M} \log p_f(z_t|\tilde{X}, t) \rightarrow \text{cross-entropy loss computed over masked timesteps}$$

$$L_u(f; X, M, Z) = \sum_{t \notin M} \log p_f(z_t|\tilde{X}, t) \rightarrow \text{cross-entropy loss computed over unmasked timesteps}$$

$L = \alpha L_m + (1 - \alpha) L_u, \alpha = 1$

$[o_1, \ldots, o_T] \rightarrow$ feature sequence output by the BERT encoder

$$p_f^{(k)}(c \mid \tilde{X}, t) = \frac{\exp(\text{sim}(A^{(k)} o_t, e_c)/\tau)}{\sum_{c'=1}^{C} \exp(\text{sim}(A^{(k)} o_t, e_{c'})/\tau)} \rightarrow \text{distribution over codewords}$$

$A^{(k)} \rightarrow$ clustering model k (ensemble of clustering algorithms)

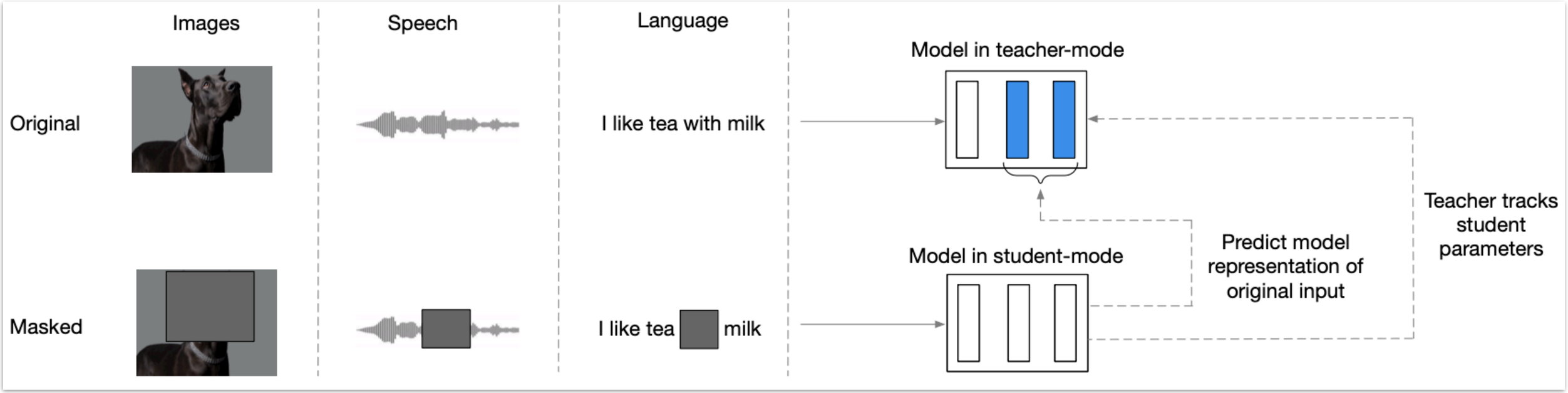
$e_c \rightarrow$ embedding for codeword c

Supervised Fine-Tuning and Decoding wav2letter++ beam search decoder wrapped in Fairseq

$\log p_{CTC}(Y \mid X) + w_1 \log P_{LM}(Y) + w_2|Y|$ $Y \rightarrow$ predicted text $|Y| \rightarrow$ length of Y

Hsu, Wei-Ning, et al. "Hubert: Self-supervised speech representation learning by masked prediction of hidden units." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021): 3451-3460.

data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language



Modality-specific encoding of the input data

Computer Vision: sequence of patches (each spanning 16×16 pixels)

Speech: encoded using a multi-layer 1D CNN mapping 16kHz waveform to 50Hz representations

Text: byte-pair encoded sub-word units (learned embedding vectors)

Teacher Parametrization

exponentially moving average (EMA)

$\Delta \leftarrow \tau\Delta + (1-\tau)\,\theta$

Targets

$y_t = \frac{1}{K}\sum_{l=L-K+1}^{L} \hat{a}_t^l$

$\hat{a}_t^l \rightarrow$ normalized output of block l at time-step t

Averaging top K blocks

Objective

$$\mathcal{L}(y_t, f_t(x)) = \begin{cases} \frac{1}{2}(y_t - f_t(x))^2/\beta & |y_t - f_t(x)| \le \beta \\ (|y_t - f_t(x)| - \frac{1}{2}\beta) & \text{otherwise} \end{cases}$$

Smooth L1 loss

	MNLI	QNLI	RTE	MRPC	QQP	STS-B	CoLA	SST	Avg.
Base models									
BERT (Devlin et al., 2019)	84.0/84.4	89.0	61.0	86.3	89.1	89.5	57.3	93.0	80.7
Baseline (Liu et al., 2019)	84.1/83.9	90.4	69.3	89.0	89.3	88.9	56.8	92.3	82.5
data2vec	83.2/83.0	90.9	67.0	90.2	89.1	87.2	62.2	91.8	82.7
+ wav2vec 2.0 masking	82.8/83.4	91.1	69.9	90.0	89.0	87.7	60.3	92.4	82.9

	ViT-B	ViT-L
MoCo v3 (Chen et al., 2021b)	83.2	84.1
DINO (Caron et al., 2021)	82.8	-
BEiT (Bao et al., 2021)	83.2	85.2
MAE (He et al., 2021)	83.6	85.9
SimMIM (Xie et al., 2021)	83.8	-
MaskFeat (Wei et al., 2021)	84.0	85.7
data2vec	84.2	86.2

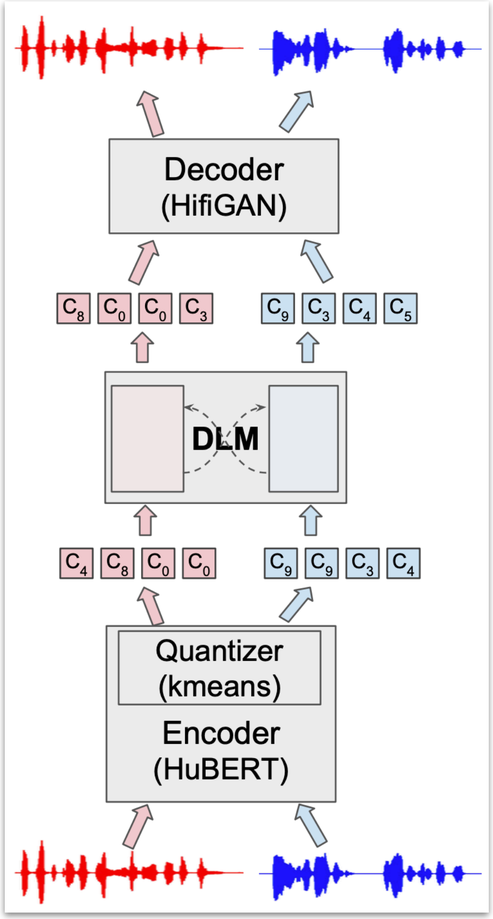
	Unlabeled data	LM	Amount of labeled data 10m	1h	10h	100h	960h
Base models							
wav2vec 2.0 (Baevski et al., 2020b)	LS-960	4-gram	15.6	11.3	9.5	8.0	6.1
HuBERT (Hsu et al., 2021)	LS-960	4-gram	15.3	11.3	9.4	8.1	-
WavLM (Chen et al., 2021a)	LS-960	4-gram	-	10.8	9.2	7.7	-
data2vec	LS-960	4-gram	12.3	9.1	8.1	6.8	5.5

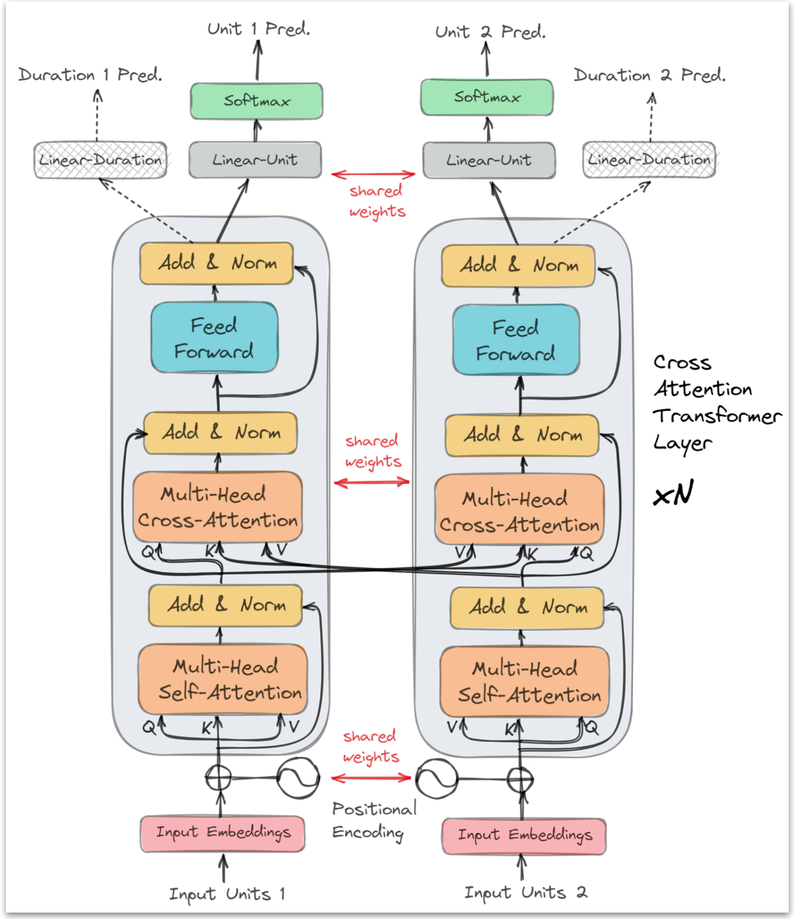
Baevski, Alexei, et al. "data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language." (2022).

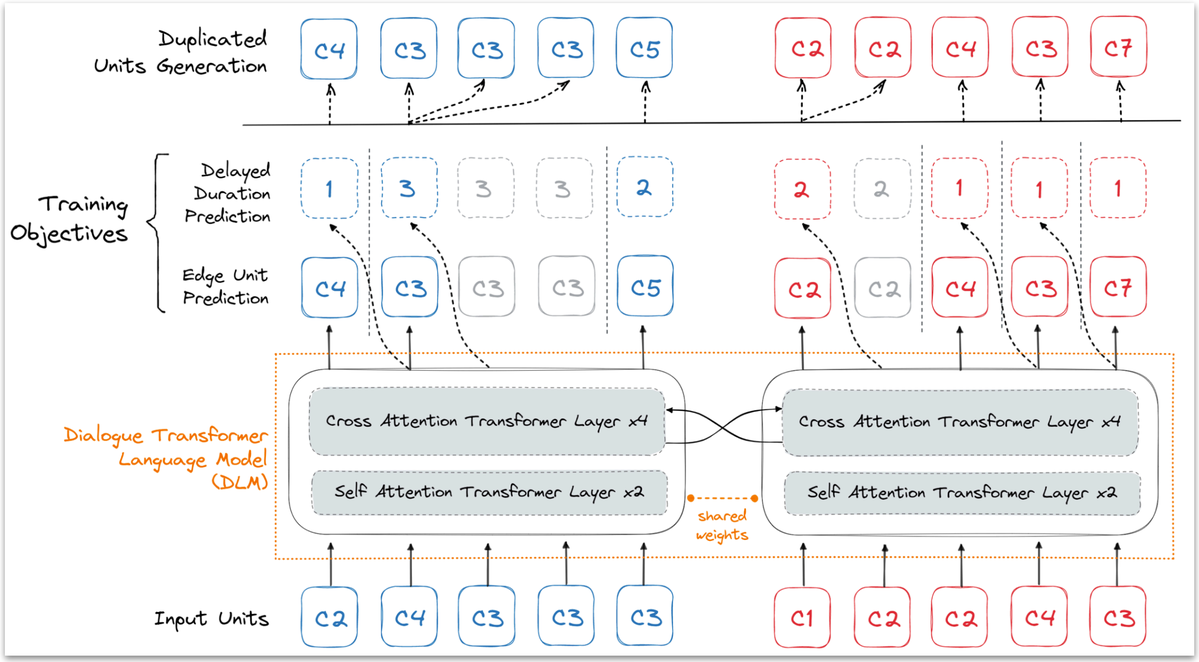
Generative Spoken Dialogue Language Modeling

dGSLM → a "textless" model able to generate audio samples of naturalistic spoken dialogues
2000 hours of two-channel raw conversational audio (Fisher dataset)
– Speech-to-Units encoder (HuBERT followed by kmean clustering)
– Units-to-Units language model (new Dialogue Transformer Language Model, or DLM)
– Units-to-Speech decoder (modified Hifi-GAN neural vocoder)

https://speechbot.github.io/dgslm
– casual expressions (filler words like 'hmm')
– a variety of non verbal sounds (e.g., laughter)
– turn taking







$$\mathcal{L}_{EU} = \sum_{c=1}^{2} \sum_{\substack{t \\ u_t^{(c)} \neq u_{t-1}^{(c)}}} \log p(u_t^{(c)} \mid u_{1:t-1}^{(1,2)}; \theta) \rightarrow \text{edge unit prediction loss}$$

$\Delta \rightarrow$ delay factor

$$\mathcal{L}_{ED} = \sum_{c=1}^{2} \sum_{\substack{t \\ u_t^{(c)} \neq u_{t-1}^{(c)}}} \left| d_t^{(c)} - \hat{d}_t^{(c)} \left(u_{1:t-1+\Delta}^{(1,2)}; \theta \right) \right| \rightarrow \text{delayed duration prediction loss}$$

disentangles the content modeling problem from the duration modeling problem by training the language model on deduplicated discrete units and the corresponding unit durations with different objectives

Nguyen, Tu Anh, et al. "Generative Spoken Dialogue Language Modeling." *arXiv preprint arXiv:2203.16502* (2022).

Boulder

Questions?

YouTube Playlist