Weakly Supervised Learning

Concept, theory, SOTA works,
and application in classification task

23 Jan 2022

Objective: introduce WSL strategy and bring new idea to current task

Contents:

1. Description of problem and concept.
2. Overview, concepts and work flow.
3. SOTA works in CV and NLP areas.
4. Sibling: Self-Supervised Learning
5. Business side prospect.

References:

1. *Weak-shot Fine-grained Classification via Similarity Transfer, NIPS 2021*
   1. *Learning to cluster in order to transfer across domains and tasks. ICLR 2018*
2. *W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training, 2021*
   1. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, NIPS 2021*
      1. *Product quantization for nearest neighbor search, IEEE TPAMI 2011*
      2. *Representation Learning with Contrastive Predictive Coding, 2018*
   2. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NACCL 2019*
3. *ReSSL Relational Self-Supervised Learning with Weak Augmentation, NIPS 2021*

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
   - Appetizer: image classification task
   - Main-dish: speech recognition task
   - Dessert: what about self-supervised learning
4. Summary

Introduction: Weakly Supervised Learning (WSL)

Type of data
- Unlabeled data
- Weakly labeled data (label contains error)
- Labeled data
- Multi-labeled data (instance belongs to multi-category)

Data used in WSL task:
Labeled data (1~30%) + unlabeled data

Problem:
How utilize unlabeled data to train network?

PS: 0% labeled data + 100% unlabeled data convert WSL to self-supervised learning (a task aim to create a better pre-trained network).

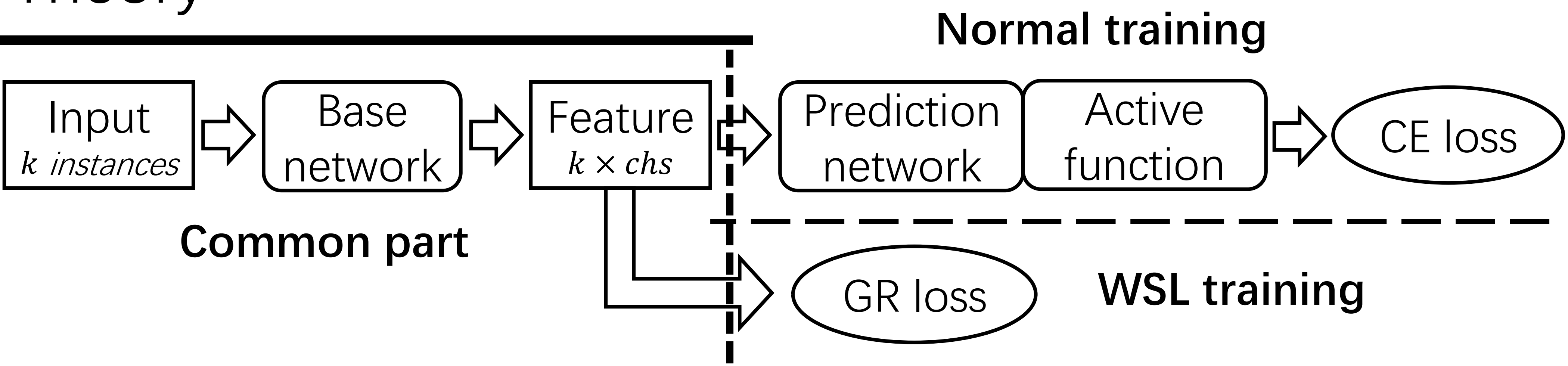
Training via normal:
- Category-distinct
- Minimize difference to target **probability** (or distribution of prob.) (e.g. **C**ross-**E**ntropy loss)

Training via WSL:
- Category-agnostic
- Minimize distance between instances on **feature** space (e.g. **G**raph **R**egularization loss)
- Using auxiliary tasks (e.g. revert X-axis on CV task, masked language model on NLP task)

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
   - Appetizer: image classification task
   - Main-dish: speech recognition task
   - Dessert: what about self-supervised learning
4. Summary

Theory



Example of **G**raph **R**egularization loss:
(loss inside single group)

Distance function, usually L2

$$L_{reg} = \sum_{i,j} \tilde{s}_{i,j} \| h(\mathbf{x}_i) - h(\mathbf{x}_j) \|_2^2,$$

Accumulate all pairs of samples

Weight, calculated by A,B, optional.

Instance A,B, selected by strategy

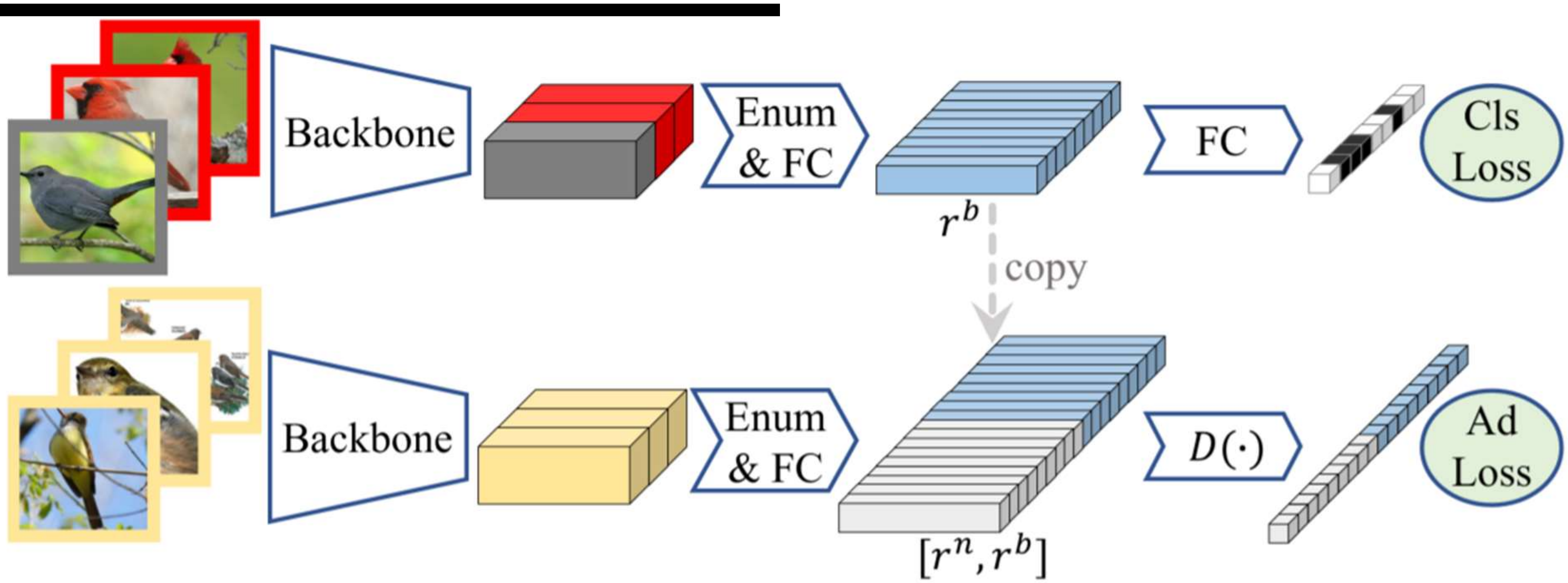
What people focus:

- How to choose instance pairs without label?
- How to add weight on these pairs?
- How to maximize distance between groups?

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
   - Appetizer: image classification task
   - Main-dish: speech recognition task
   - Dessert: what about self-supervised learning
4. Summary

Appetizer: Weak-shot image classification



Novelty:

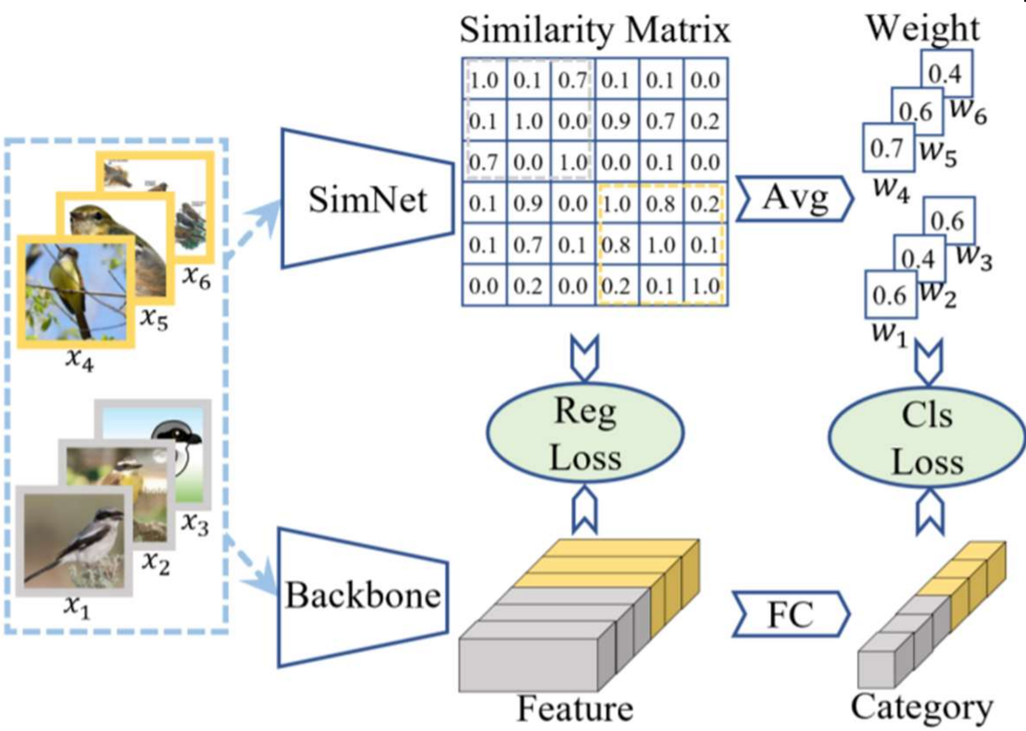
- Use *transferred similarity* + *denoise strategy* to tackle web training data.
- Apply adversarial loss to similarity net.

Network (derived from *ref.1.1*):
Pair-enumerate: $(k, d) \rightarrow (k, k, 2d) \rightarrow (k^2, 2d)$
Fc: $(k^2, 2d) \rightarrow (k^2)$; Similar prediction: Y/N

Feature constraint: cluster unlabeled data based on feature from labeled data.

Weak-shot Fine-grained Classification via Similarity Transfer, NIPS 2021

Appetizer: Weak-shot image classification



$$L_{reg} = \sum_{i,j} \tilde{s}_{i,j} \|h(\mathbf{x}_i) - h(\mathbf{x}_j)\|_2^2, \quad w_{c,i} = \frac{1}{N_c^n} \sum_{j=1}^{N_c^n} \frac{s_{c,i,j} + s_{c,j,i}}{2}.$$

Training steps:

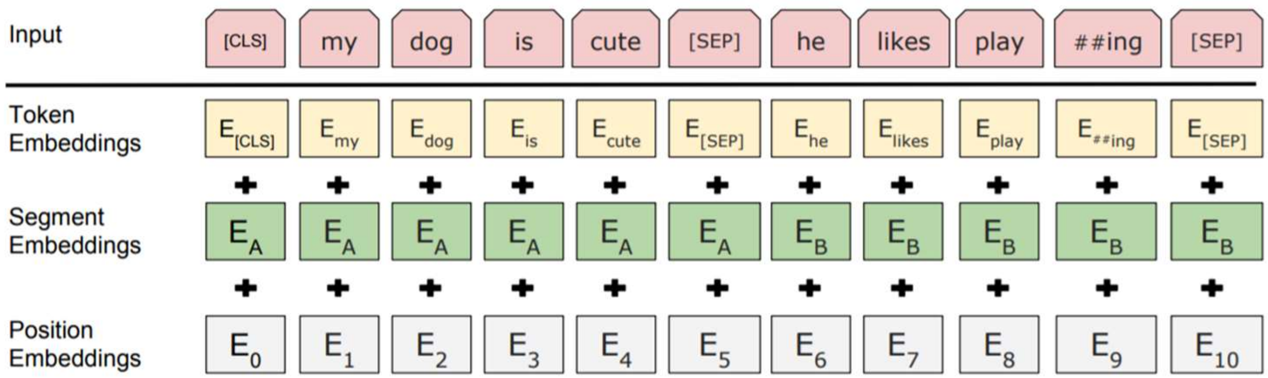
1. Train network on labeled data.
   - Prediction target: binary classification of similar or not.
   - CE loss.
   - Strong constraint: batch size k s.t. k^2 items pre-step.
2. Cluster unlabeled data via pre-trained network.
   - Select k_1 samples from labeled data.
   - Select k_2 samples from unlabeled data.
   - Calculate feature f_1, f_2.
   - Weighted L2 loss by similarity.

Weak-shot Fine-grained Classification via Similarity Transfer, NIPS 2021

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
   - Appetizer: image classification task
   - Main-dish: speech recognition task
   - Dessert: what about self-supervised learning
4. Summary

Main-dish: W2v-BERT



Embedding E_*: codebook consisted by trainable parameters.

Token: vocabulary + start token + end token

Segment: (2, ch) parameters

Position: trainable parameters based on location.

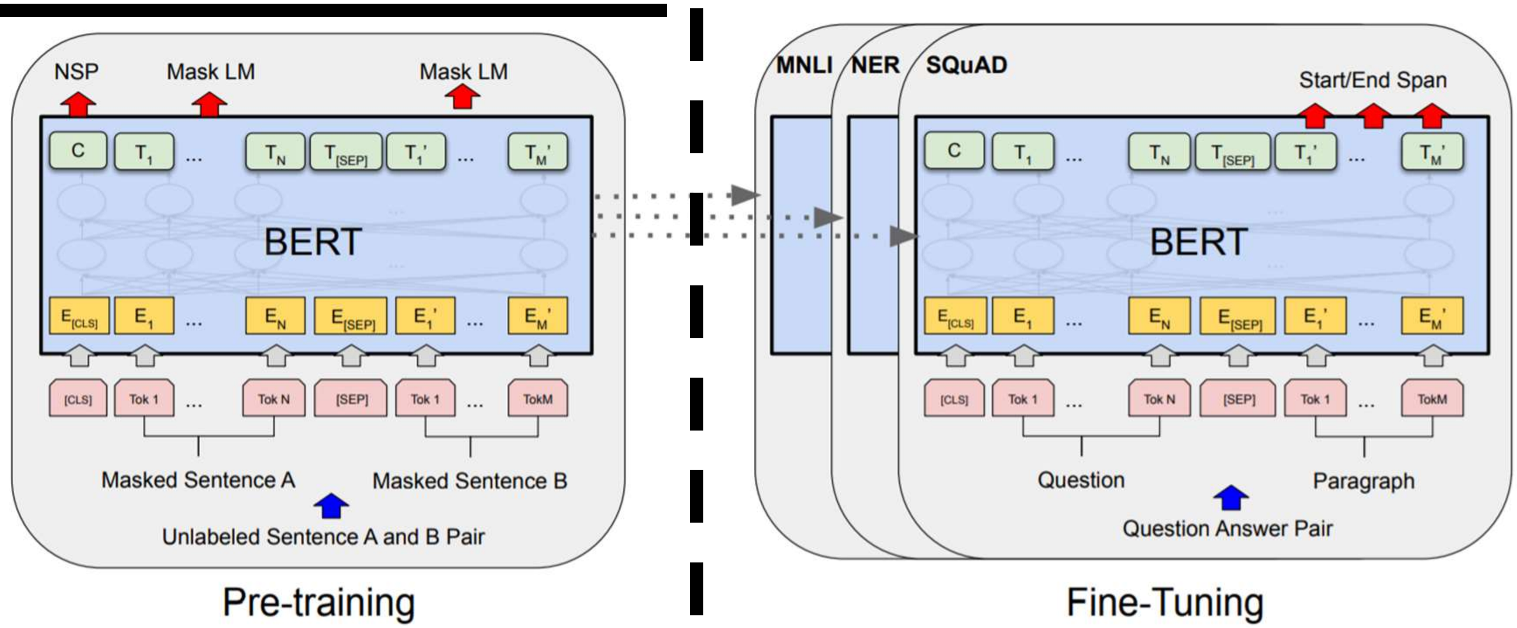
Input: 2 sentences A,B from articles.

Task: randomly mask multi-words (15%) in A and B, predict the words by context (CE loss).

Task: randomly select continuous and discontinuous (50%/50%) sentences A and B, determine whether B is the next sentence of A (CE loss).

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NACCL 2019, Google AI

Main-dish: W2v-BERT

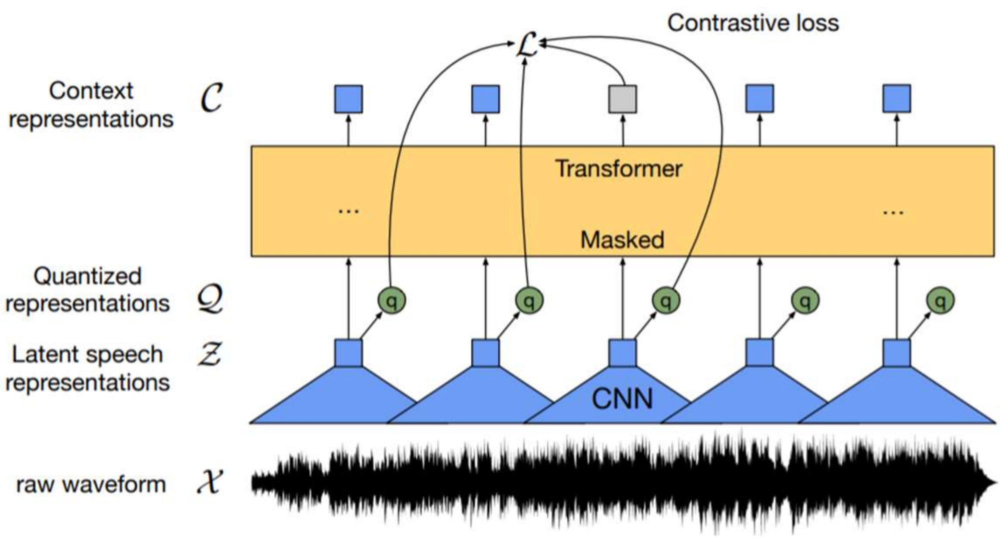


Bidirectional encoder network: transformer with unfixed sequence.

Normal supervised training for task with low amount (e.g. 1/1000 of unlabeled data), labeled, specific data (so called downstream task).

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NACCL 2019, Google AI

Main-dish: W2v-BERT



Auxiliary task: identify the true quantized representation from masked region (*ref2.1.1*).

G codebooks $\in \mathbb{R}^{V\times d}$, product quantization:

$z \in \mathbb{R}^{ch} \xrightarrow{f(z)} I \in \mathbb{R}^{G\times V} \xrightarrow{Gum.Smax(I)} \mathrm{p_{g,v}}$

select G rows $e_1, \dots, e_G$ by $\mathrm{p_{g,v}}$ via $argmax$

concatenate into $e \in \mathbb{R}^{G\times d} \xrightarrow{f(z)} q \in \mathbb{R}^{ch}$

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

τ: non-negative temperature

$n = -\log(-\log(u))$

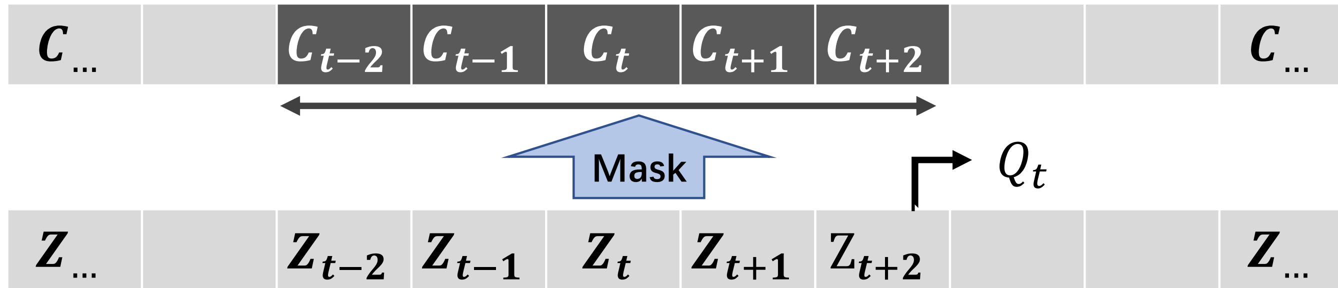
u: uniform sample from (0,1)

Input: speech voice (1d sequence, float)
Task 1: identify the true quantized latent speech representation
Task 2: keep vectors in codebooks used as equal as possible.

Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, NIPS 2020, Facebook AI

Main-dish: W2v-BERT

Random mask:
replace inputs of transformer by shared, trained feature vector



Contrastive Loss (*ref2.1.2*):

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

c: context representation
q: quantized representation
sim: cosine similarity

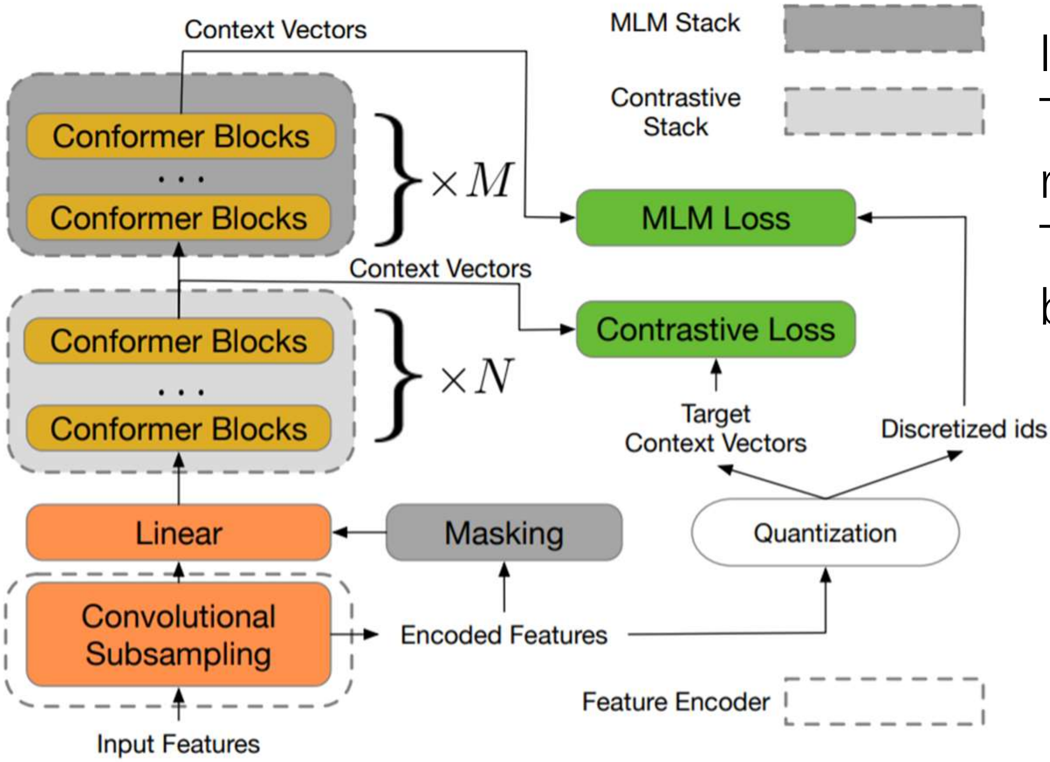
$$sim(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T\mathbf{b}/\|\mathbf{a}\|\|\mathbf{b}\|$$

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

Diversity Loss: encourage the equal use of the V entries in each of the G codebooks by maximizing the entropy of the averaged softmax distribution I

Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, NIPS 2020, Facebook AI

Main-dish: W2v-BERT



Combine W2v and BERT to conduct WSL on unlabeled voice data.

Input: unlabeled voice data
Task: identify the true quantized latent speech representation (contrastive loss)
Task: randomly mask words and predict them by context (**M**asked **L**anguage **M**odeling).

Contrastive stack: transformer-based encoder
MLM stack: transformer-based decoder

W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training, 2021, MIT & Google Brain

Main-dish: W2v-BERT

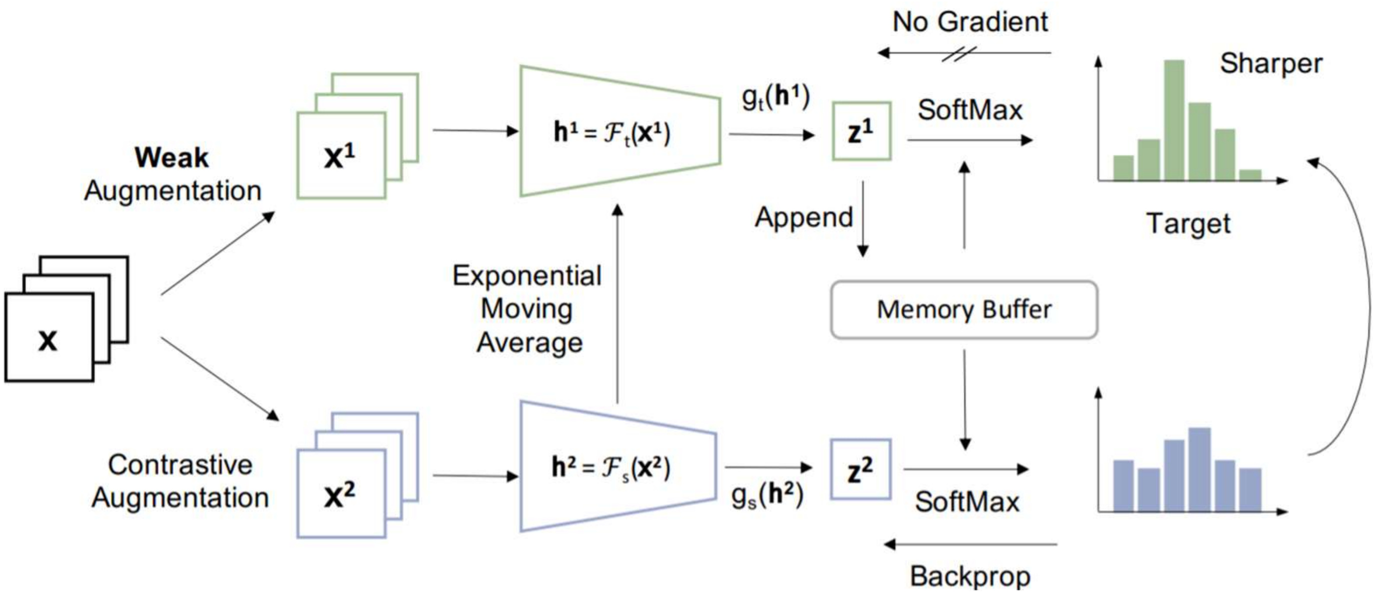
Method	Unlabeled Data (hrs)	AM Size (B)	LM Size (B)	No LM				With LM			
				dev	dev-other	test	test-other	dev	dev-other	test	test-other
Trained from Scratch											
Conformer L [21]*	N/A	0.1	0.1	1.9	4.4	2.1	4.3	–	–	1.9	3.9
Self-training Only											
Conformer L with NST [21]	60k	0.1	0.1	1.6	3.3	1.7	3.5	1.6	3.1	1.7	3.3
Pre-training Only											
wav2vec 2.0 [22]	60k	0.3	$> 0.4^{\dagger}$	2.1	4.5	2.2	4.5	1.6	3.0	1.8	3.3
HuBERT Large [25]	60k	0.3	–	–	–	–	–	1.5	3.0	1.9	3.3
HuBERT X-Large [25]	60k	1.0	–	–	–	–	–	1.5	**2.5**	1.8	2.9
w2v-Conformer XL [21]	60k	0.6	0.1	1.7	3.5	1.7	3.5	1.6	3.2	**1.5**	3.2
w2v-Conformer XXL [21]	60k	1.0	0.1	1.6	3.2	1.6	3.3	1.5	3.0	**1.5**	3.1
w2v-BERT XL (Ours)	60k	0.6	0.1	**1.5**	2.9	**1.5**	2.9	**1.4**	2.8	**1.5**	2.8
w2v-BERT XXL (Ours)	60k	1.0	0.1	**1.5**	**2.7**	**1.5**	**2.8**	**1.4**	2.6	**1.5**	**2.7**
Pre-training + Self-training											
wav2vec 2.0 [22]	60k	0.3	> 0.4	**1.3**	3.1	1.7	3.5	**1.1**	2.7	1.5	3.1
w2v-Conformer XXL [21]	60k	1.0	0.1	**1.3**	2.7	1.5	2.8	1.3	2.6	**1.4**	2.7
w2v-Conformer XXL+ [21]	60k	1.1	0.1	**1.3**	2.7	1.5	2.7	1.3	2.6	**1.4**	2.6
w2v-BERT XL (Ours)	60k	0.6	0.1	**1.3**	2.6	**1.4**	2.7	1.3	2.6	**1.4**	2.6
w2v-BERT XXL (Ours)	60k	1.0	0.1	1.4	**2.4**	**1.4**	**2.5**	1.3	**2.4**	**1.4**	**2.5**

W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training, 2021, MIT & Google Brain

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
   - Appetizer: image classification task
   - Main-dish: speech recognition task
   - Dessert: what about self-supervised learning
4. Summary

Dessert: SSL, strategy purely for pre-training



Teacher-student structure:
x^1: teacher input, weak augmentation from x
x^2: student input, normal augmentation from x

Data augmentation based **S**elf-**S**upervised **L**earning: distribution of predicted classes of instances between two augmentations should be similar.
Calculate the similarity between *single* z_i^1 and *all* z_j^2.

ReSSL: Relational Self-Supervised Learning with Weak Augmentation, NIPS 2021

Dessert: SSL, strategy purely for pre-training

Preliminaries:

Noise **C**ontrastive **E**stimation

Reference (1) Non-zero τ

$$\mathcal{L}_{NCE} = -\log \frac{\exp(sim(\mathbf{z}^1, \mathbf{z}^2)/\tau)}{\exp(sim(\mathbf{z}_i^1, \mathbf{z}_i^2)/\tau) + \sum_{k=1}^{N} \exp(sim(\mathbf{z}_i^1, \mathbf{z}_k)/\tau)}$$

Target term (1)

Local2global term $(1 \times N)$

sim: cosine similarity

$$sim(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v} / \|\mathbf{u}\| \|\mathbf{v}\|$$

Week aug. and normal aug. vs. original image:

1-1 similarity

$$\mathbf{p}_i^1 = \frac{\exp(sim(\mathbf{z}^1, \mathbf{z}_i)/\tau_t)}{\sum_{k=1}^{K} \exp(sim(\mathbf{z}^1, \mathbf{z}_k)/\tau_t,)}. \quad \mathbf{p}_i^2 = \frac{\exp(sim(\mathbf{z}^2, \mathbf{z}_i)/\tau_s)}{\sum_{k=1}^{K} \exp(sim(\mathbf{z}^2, \mathbf{z}_k)/\tau_s,)}$$

1-all similarity

$z^{1,2}$: 1st image from week and normal aug.

z_i: *i-th* original image in batch.

Minimize **K**ullback–**L**eibler divergence

$$\mathcal{L}_{relation} = D_{KL}(\mathbf{p}^1 || \mathbf{p}^2) = H(\mathbf{p}^1, \mathbf{p}^2) - H(\mathbf{p}^1)$$

Because $H(p^1)$ is target distribution so only regress $H(p^1, p^2)$ hence $L = H_{CE}(p^1, p^2)$

ReSSL: Relational Self-Supervised Learning with Weak Augmentation, NIPS 2021

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
   - Appetizer: image classification task
   - Main-dish: speech recognition task
   - Dessert: what about self-supervised learning
4. Summary

WSL: a way to utilize public, unlabeled data

Methodology:

- Clustering on feature space: reduce feature distance, increase similarity, etc.
- Data augmentation: add uncertainty on certain data.

From view of business:

- Convert pre-trained network as asset: one network to multiple downstream tasks.
- Reduce labeling cost.
- Reduce data storage cost: unlabeled data can be removed after producing pre-trained network.

From view of R&D:

- Additional supervision is conducive to regress network more quickly, precisely.
- Graph based supervision positively contribute to class robustness.
- Flexible, dynamic network structure to deliver: FC layer is no longer the only choice.
- Easily used as novelty in research.