UNIVERSITY OF TORONTO | Engineering

Incremental Few-shot Learning via Vector Quantization in Deep Embedded Space

Kuilin Chen, Chi-Guhn Lee
University of Toronto

Incremental few-shot learning

- Incremental learning is a learning paradigm that allows the model to continually learn new tasks on novel data, without forgetting how to perform previously learned tasks
- The capability of incrementally learning new tasks without forgetting old ones is challenging due to catastrophic forgetting
- This challenge becomes greater when novel tasks contain very few labelled training samples
- It is desirable to develop algorithms to support incremental learning from very few samples

Unified Framework

- Unified framework of IDLVQ for both classification and regression can be derived from a Gaussian mixture
- A raw input is projected into a feature space by a neural network f_{θ^1}
- Reference vectors $\mathbf{M}^1 = \{\mathbf{m}_1^1, ..., \mathbf{m}_{N^1}^1\}$ are place in feature space
- We will add more reference vectors as we learn novel tasks
- The marginal distribution of a feature vector is a Gaussian mixture
- Assumption: isotropic Gaussian centered at a reference vector with the same covariance

Unified Framework

- Posterior: $p^1(i|\mathbf{x}) = \dfrac{\kappa(f_{\theta^1}(\mathbf{x}), \mathbf{m}_i^1)}{\sum_{j=1}^{N^1} \kappa(f_{\theta^1}(\mathbf{x}), \mathbf{m}_j^1)}$,

- $\kappa(f_{\theta^1}(\mathbf{x}), \mathbf{m}_i^1) = \exp(-\|f_{\theta^1}(\mathbf{x}) - \mathbf{m}_i^1\|^2/\gamma)$ is a Gaussian kernel

- The conditional expectation of output:

$$\hat{y} = \sum_{i=1}^{N^1} p^1(i|\mathbf{x}) q_i^1$$

- q is the reference target
- The model is learned by minimizing an appropriate loss function

Incremental few-shot classification

Algorithm 1 IDLVQ-C

In the base task ($t = 1$)

Initialize θ^1, $\{\mathbf{m}_1^1, ..., \mathbf{m}_{N^1}^1\}$ and γ

Minimize $\mathcal{L} = \mathcal{L}_{CE} + \lambda_{intra}\mathcal{L}_{intra}$ w.r.t. θ^1, $\{\mathbf{m}_1^1, ..., \mathbf{m}_{N^1}^1\}$ and γ

Pick exemplars from $\mathcal{D}^1$ for classes in the base task: $\mathbf{x}_i^{'} = \arg\min_{\mathbf{x}\in\mathcal{D}^1} \left\|f_{\theta^{t-1}}(\mathbf{x}) - \mathbf{m}_i^1\right\|^2$

for novel task $t = 2, 3, ...$ **do**

Initialize $\{\mathbf{m}_{N^{t-1}+1}^t, ..., \mathbf{m}_{N^t}^t\}$

Minimize $\mathcal{L} = \mathcal{L}_M + \lambda_F\mathcal{L}_F + \lambda_{intra}\mathcal{L}_{intra}$ w.r.t. θ^t and $\{\mathbf{m}_{N^{t-1}+1}^t, ..., \mathbf{m}_{N^t}^t\}$

Calibrate old reference vector using $\mathbf{m}_i^t = \mathbf{m}_i^{t-1} + \delta_i^t$

Pick exemplars from $\mathcal{D}^t$ for classes in the novel task t: $\mathbf{x}_i^{'} = \arg\min_{\mathbf{x}\in\mathcal{D}^t} \|f_{\theta^{t-1}}(\mathbf{x}) - \mathbf{m}_i^t\|^2$

end for

Compact intra-class variation

$$\mathcal{L}_{intra} = \sum_{\forall(\mathbf{x},y),y=i} \|f_\theta(\mathbf{x}) - \mathbf{m}_i\|^2$$

Update model when necessary

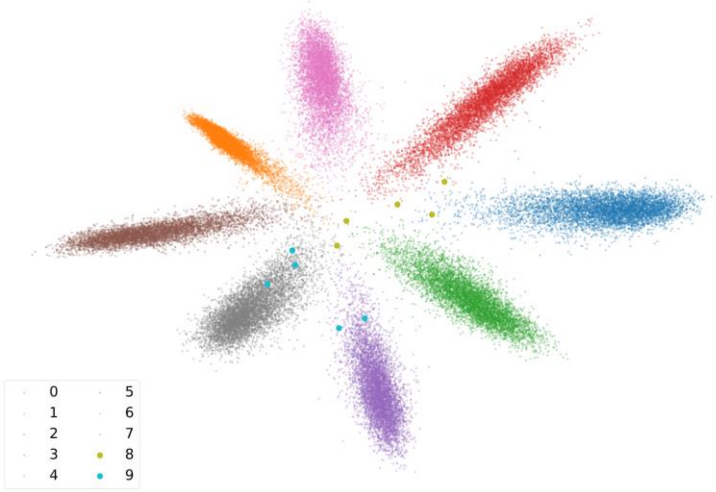
$$\mathcal{L}_M = \mathrm{ReLU}\left(\frac{\left\|f_{\theta^t}(\mathbf{x}) - \mathbf{m}_+^t\right\|^2 - \left\|f_{\theta^t}(\mathbf{x}) - \mathbf{m}_-^t\right\|^2}{\left\|f_{\theta^t}(\mathbf{x}) - \mathbf{m}_+^t\right\|^2 + \left\|f_{\theta^t}(\mathbf{x}) - \mathbf{m}_-^t\right\|^2}\right)$$

Less forgetting

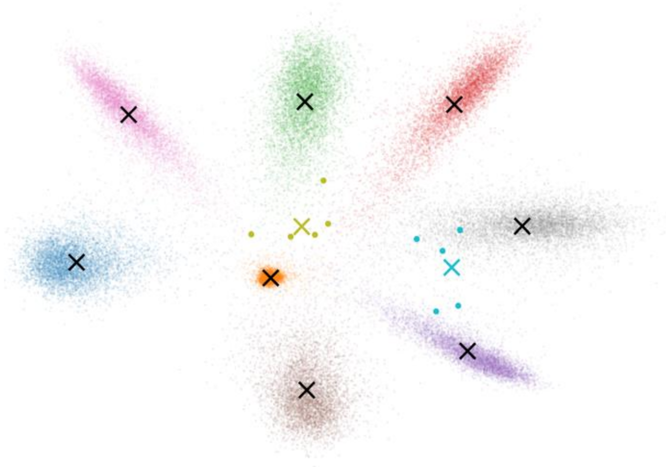
$$\mathcal{L}_F = \sum_{\forall\mathbf{x}_i^{'}} \|f_{\theta^t}(\mathbf{x}_i^{'}) - f_{\theta^{t-1}}(\mathbf{x}_i^{'})\|^2$$

Visualization of feature space

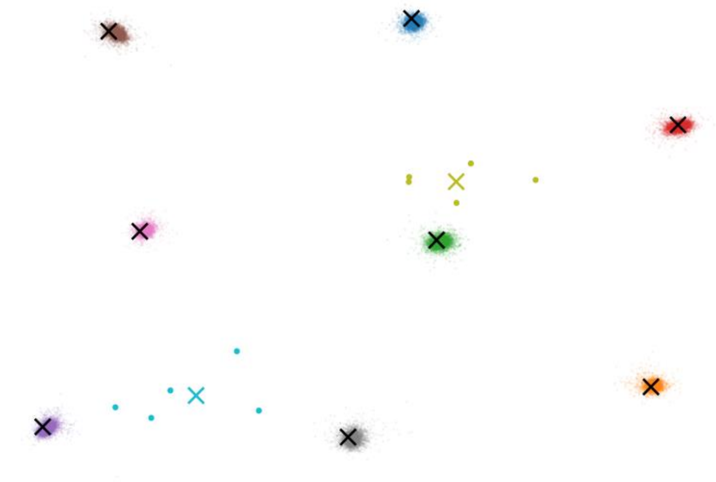
Standard NN



IDLVQ-C w.o. intra loss



IDLVQC



Prediction accuracy on CUB all classes using the 10-way 5-shot incremental setting

Method	sessions										
	1	2	3	4	5	6	7	8	9	10	11
Fine-tune	77.30	46.23	34.71	25.35	23.16	20.65	16.21	13.32	11.98	11.17	10.76
Joint train	77.30	73.28	68.80	65.34	63.75	62.00	60.81	59.71	**59.06**	**58.69**	**58.23**
iCaRL	77.30	57.18	54.67	48.11	40.76	36.85	33.12	30.42	28.22	26.84	25.23
Rebalancing	77.30	64.53	56.14	47.29	38.92	34.39	31.04	27.93	27.12	24.46	23.61
ProtoNet	77.30	69.76	66.01	62.29	59.58	57.10	55.13	54.09	52.40	51.65	50.36
ILVQ	77.30	71.50	66.79	62.71	60.20	57.84	55.27	55.06	52.42	51.72	50.47
SDC	77.34	74.45	69.45	65.27	61.81	58.26	56.14	55.71	53.31	52.79	51.52
Imprint	77.02	73.39	69.50	65.61	62.81	60.74	59.39	58.61	56.85	55.93	54.82
IDLVQ-C	**77.37**	**74.72**	**70.28**	**67.13**	**65.34**	**63.52**	**62.10**	**61.54**	**59.04**	**58.68**	**57.81**

Ablation study

Method	sessions									
	2	3	4	5	6	7	8	9	10	11
No tuning	71.93	67.14	64.21	62.61	60.13	59.04	58.47	55.64	54.25	53.66
w.o. $\mathcal{L}_{intra}$	74.75	70.26	66.89	65.05	63.18	61.84	61.36	58.61	58.14	57.24
w.o. $\mathcal{L}_F$	73.85	69.54	66.21	64.02	62.74	60.28	59.49	56.97	56.38	55.46
w.o. δ_i	74.67	70.01	66.74	64.81	63.90	61.42	60.73	58.16	57.62	56.79
$\mathcal{L}_M \rightarrow \mathcal{L}_{CE}$	73.22	69.41	66.03	63.93	63.07	61.14	60.98	58.67	58.11	57.32
IDLVQ-C	74.72	70.28	67.13	65.34	63.52	62.10	61.54	59.04	58.68	57.81

IDLVQ-R

- For regression tasks, the model is learned by minimizing MSE

$$\mathcal{L} = (y - \hat{y})^2$$

- The learnable parameters in the model are: $\theta, \mathbf{m}, q, \gamma$
- The loss is differentiable w.r.t. all parameters and learning is end-to-end
- It can be interpreted as a sparse kernel smoother

$$\hat{y} = \frac{\sum_{i=1}^{N^t} \kappa(f_\theta(\mathbf{x}), \mathbf{m}_i) q_i}{\sum_{i=1}^{N^t} \kappa(f_\theta(\mathbf{x}), \mathbf{m}_i)}$$

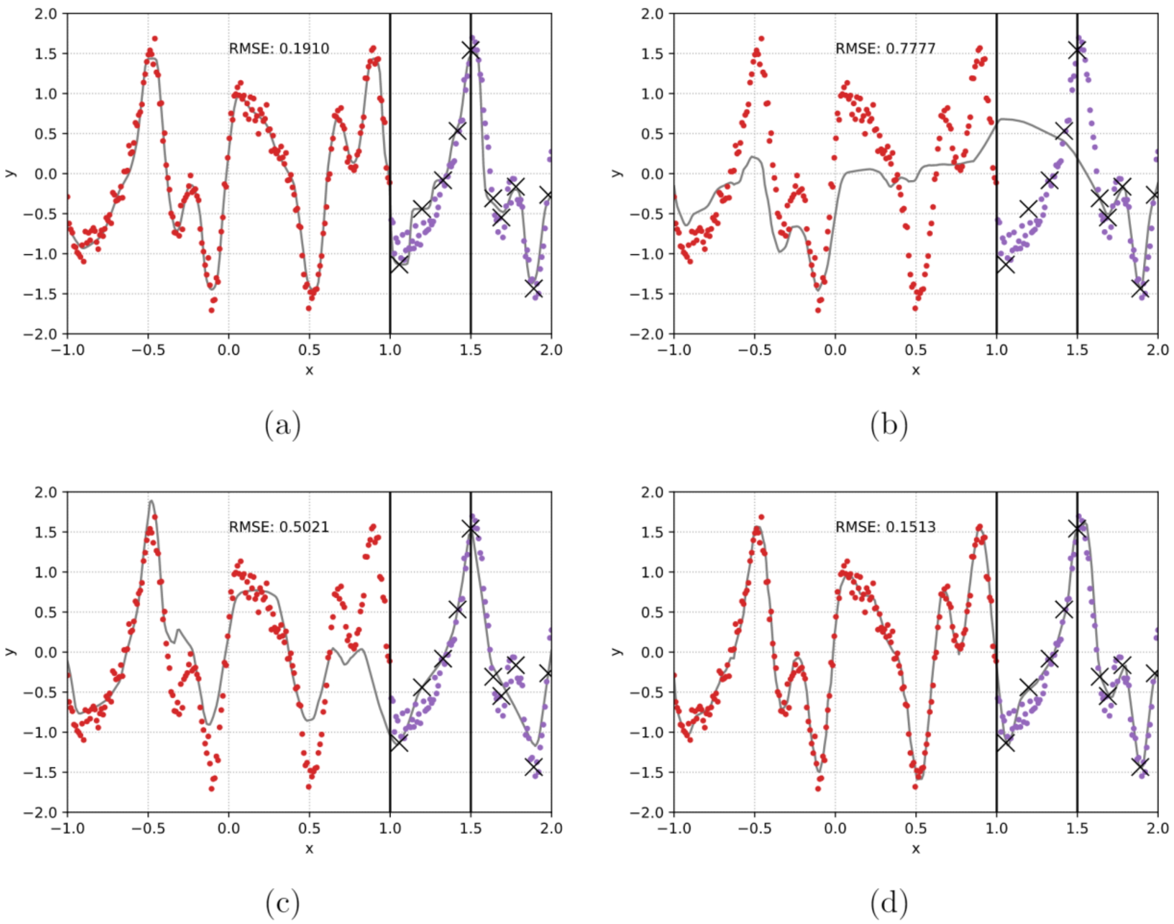
Regression Example

- We generate some nonlinear data

$$y = \sin(3\pi x) + 0.3\cos(9\pi x) + 0.5\sin(7\pi x) + \epsilon$$

- The old data contains 1000 samples generated when $x \in [-1, 1]$
- The model was originally trained on old data
- 1st novel task: 5-shot samples by sampling $x \in [1, 1.5]$
- 2nd novel task: 5-shot samples by sampling $x \in [1.5, 2]$
- Test samples are randomly generated by sampling $x \in [-1, 2]$

Result



- (a) Our method
- (b) Fine-tune using novel data only
- (c) Fine-tune using novel data and saved exemplars
- (d) Offline training using all training samples from all tasks

Conclusions

- We propose a unified framework to handle incremental few-shot classification and regression problems
- The proposed method is based on vector quantization in deep embedded space
- Empirical studies show that the proposed achieves state-of-the-art performance

Thank you!