ARCADE: A RAPID CONTINUAL ANOMALY DETECTOR

Ahmed Frikha, Denis Krompaß and Volker Tresp

Ludwig Maximilian University of Munich
Siemens AG, Germany

International Conference on Pattern Recognition 2020

Introduction: Continual Anomaly Detection (CAD)

- Most deep learning success stories are in the offline learning setting (i.i.d. data)
- Neural Networks suffer from catastrophic forgetting when sequentially trained on different data distribution, as it is the case in the continual learning setting
- While prior continual learning works consider class-balanced classification, several real-world problems exhibit extreme class-imbalance, e.g. anomaly detection (AD)
- Concrete CAD example: In production plants, products and manufacturing processes are continuously changing, and AD has to be performed in every setting
- Training an anomaly detector from scratch for each setting or storing data and retraining the model every time a new setting is available, is not efficient

→ CAD relaxes this cold start situation by building a central anomaly detector that continuously improves by learning new AD tasks without forgetting or overfitting to the normal class

Contributions & Summary

Motivation

- The intersection of continual learning and anomaly detection is relevant but unexplored
- A learning strategy is needed to cope with the challenges of this novel problem: Catastrophic forgetting and overfitting to the majority class

Contributions

- We introduce and define the novel and relevant continual anomaly detection (CAD) problem
- We propose a first, strong and model-agnostic baseline to tackle CAD
- We validate our method on 3 datasets, where we outperform continual learning and anomaly detection approaches
- We analyze the learning strategy meta-learned by our algorithm

Meta-Learning A Continual Anomaly Detector

- ARCADe uses a bi-level optimization scheme to meta-learn **an initialization** and **a learning rate** for each model parameter to tackle CAD
- Finetuning the meta-learned initialization using the meta-learned learning rates on a sequence of unseen AD tasks yields a model that has high performance on all these tasks
- During meta-training, the parameter-specific initializations and learning rates are updated to increase the adapted model's performance on class-balanced validation sets of all tasks
- Meta-learning a learning rate for each model parameter allows the optimization algorithm to identify the parameters that are most responsible for catastrophic forgetting and/or overfitting to the majority class and reduce their learning rates (negative learning rates are clipped to 0)
- Variants: During adaptation, ARCADe-M finetunes all model parameters and ARCADe-H finetunes only the parameters of the classification head

Algorithm 1 ARCADe Meta-training Procedure

Require: D_{tr}: Set of meta-training task-sequences
Require: β: Learning rate for the meta-update
Require: K: Adaptation set size

Randomly initialize model parameters $\boldsymbol{\theta}$ and parameter-specific learning rates $\boldsymbol{\alpha}$
while not done **do**
  Sample a batch of task-sequences S_i from D_{tr}
  Initialize meta-learning loss $L_{meta} = 0$
  for each sampled S_i **do**
    Adapt model initialization θ to S_i using the train set of each task and the learning rates $\boldsymbol{\alpha}$, yielding $\boldsymbol{\theta}'_{i,J}$
    Evaluate the adapted model $f_{\boldsymbol{\theta}'_{i,J}}$ on S_i using the validation set of each task to compute $L_{s,i}$
    $L_{meta} = L_{meta} + L_{s,i}$
  end for
  Update $(\boldsymbol{\theta}, \boldsymbol{\alpha})$: $(\boldsymbol{\theta}, \boldsymbol{\alpha}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\alpha}) - \beta \nabla_{(\boldsymbol{\theta}, \boldsymbol{\alpha})} L_{meta}$
end while
return Meta-learned parameters $\boldsymbol{\theta}$ and learning rates $\boldsymbol{\alpha}$

Experimental Results

Fig. 1. Retained accuracy on Omniglot

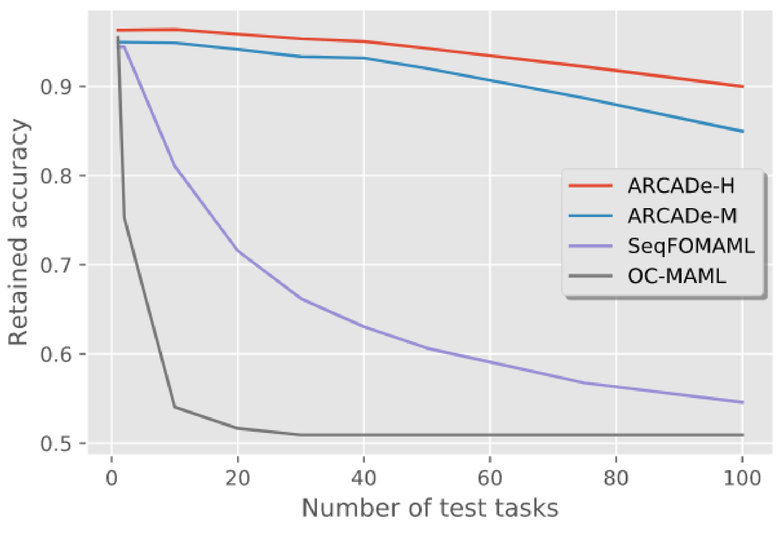


Fig. 2. Retained accuracy on MiniImageNet and CIFAR-FS

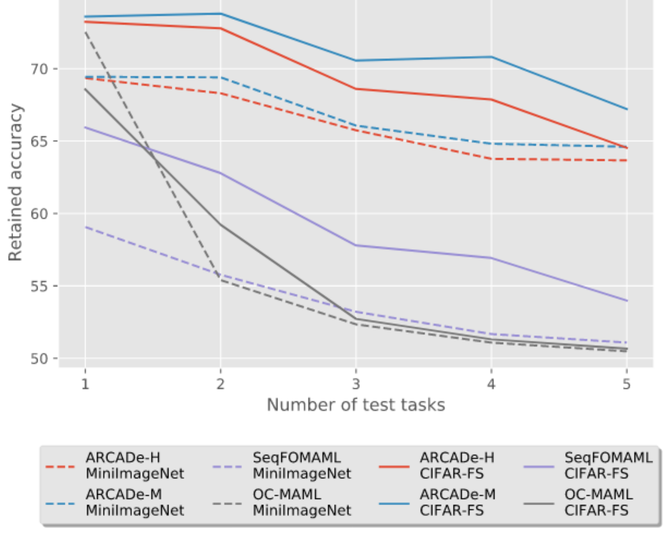
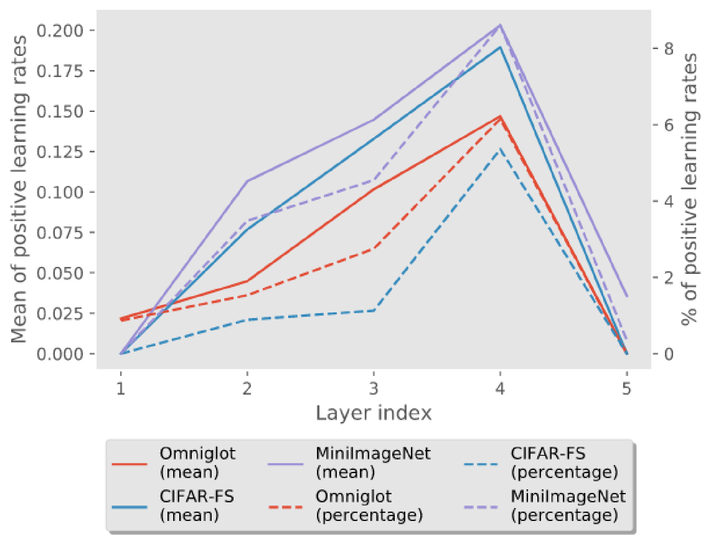


Fig. 3. Layer-wise mean and percentage of positive learning rates meta-learned by ARCADe-M



- The AD tasks are built from the original multi-class datasets by choosing one class to be the normal class and some other classes to be anomalies
- Meta-training and meta-testing tasks contain disjoint classes
- The two ARCADe variants significantly outperform the baselines, and enable learning up to 100 different tasks sequentially, while using only 10 normal examples per task, with minimal forgetting

- The first CNN layer and the classifier are almost frozen
- The last convolutional layer is adapted the most to the task-sequence

→While learning new tasks, ARCADe does not update its normal class boundary to include examples of the new tasks. Instead, it modifies the embeddings of the new examples to fit inside a frozen class boundary