OOD-MAML: meta-learning for few-shot out-of-distribution detection and classification

Jiangjiao Xu

31/01/2021

Taewon Jeong, and Heeyoung Kim, “OOD-MAML: meta-learning for few-shot out-of-distribution detection and classification", 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, 2020.

Outline

1. Review of Meta-Learning
2. Introduction of OOD Detection
3. OOD-MAML
4. Conclusion and discussion

Deep Neural Network VS Meta Learning

- **Deep neural networks (DNNs):** Large amounts of training data, small training data, low levels of performance.

- **Meta-learning (ML):** A small amount of data, better performance.

Normal Meta Learning

- In general, in few shot learning, it is assumed that the training and test data are drawn from the same distribution, and the algorithm requires a test sample to be classified into one of the known classes encountered during training.

- However, in real-world application, it is unreasonable to assume the same distribution for the training and test data.

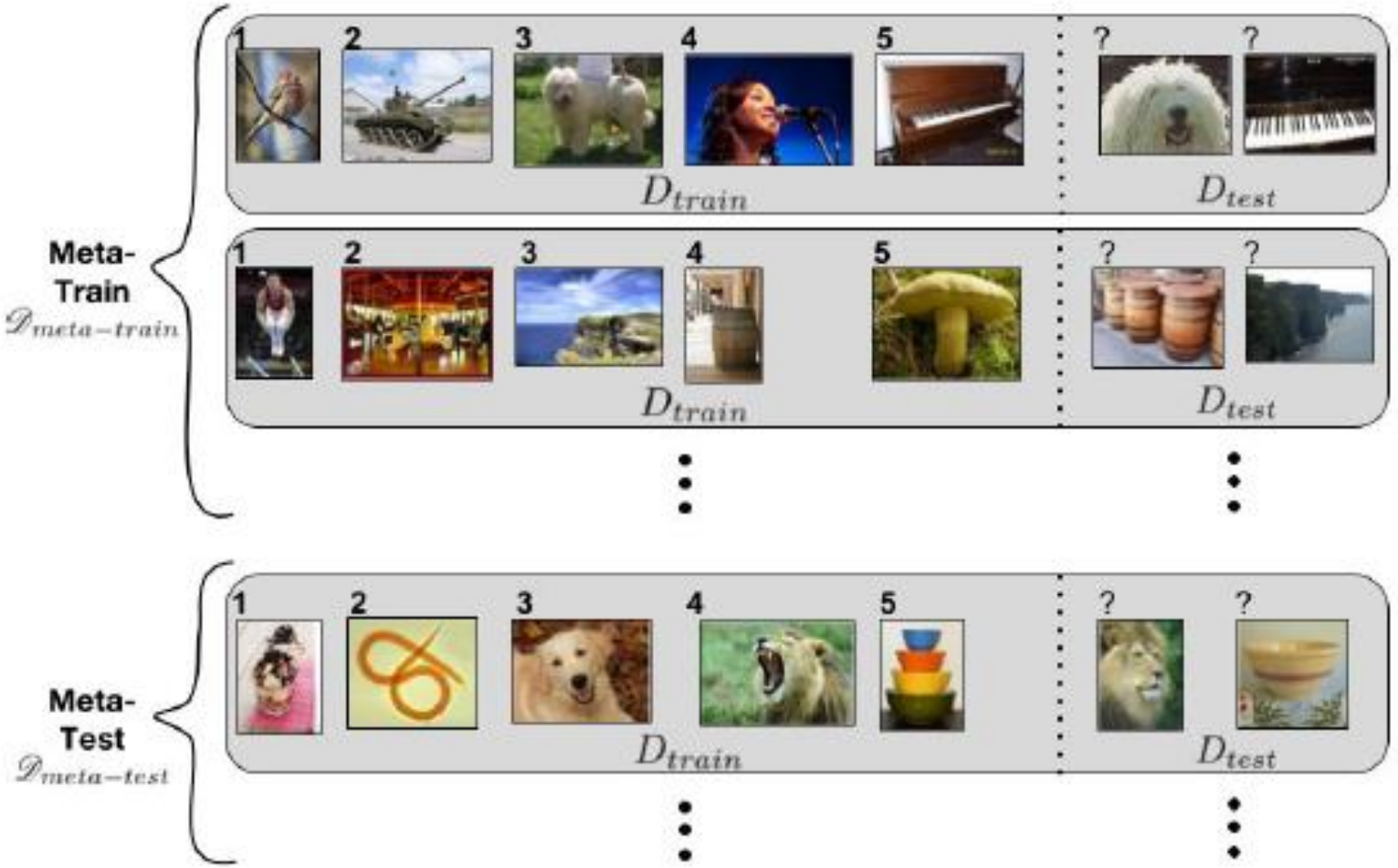
- MAML: Find good initial parameters of a model (such as DNN), update initial parameters via one or a few SGD to obtain a good performance for a new task.

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

1. 模型无关：Optimization-based meta-learning approach。
2. MAML训练：通过MAML提出的训练方式，用少量的数据就可以训练模型。
3. 快速适应：意思是模型经过训练后能够产生一个比较好的初始参数，之后对于新的任务只需要经过较少的迭代次数就可以在新任务上收敛。
4. 多场景：方法可以适用到分类、回归、强化学习等场景。

C. Finn, P. Abbeel, and S. Levine. Model-agnostic metalearning for fast adaptation of deep networks. In International Conference on Machine Learning, 2017.

General MAML



MAML for Few-Shot Supervised Learning伪代码

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
6: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
7: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
8: Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
9: **end for**
10: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
11: **end while**

要求 P(T) 是所有任务服从的分布
要求 α，β 是步长超参数

1.随机初始化网络模型参数 θ
2.while 循环（此行代表 meta train，如迭代次数等）
3.从任务分布 P(T) 中随机抽样 batch 个 tasks
4.遍历所有的任务，对每个任务 T_i 执行
5.在任务 T_i 上抽取K个样本的训练数据集D
6.使用D来计算损失函数L和损失函数对θ的梯度
7.使用梯度下降发更新参数
8.从该任务中抽取测试数据集D'，用于meta的参数更新
9.Batch 个 T_i 都执行完结束本次循环
10.meta 更新模型参数 θ，使用每个任务T_i 的测试集数据D'上计算的loss和对参数θ的梯度来更新模型参数
11.while 结束即完成 meta 训练过程

OOD Detection

1. ODD detection指的是模型能够检测出OOD样本，而OOD样本是相对于in distribution样本来说的。

2. 众所周知，如今使用DNN进行OOD检测是一个具有挑战性的问题，因为DNN会对OOD样本会对一个ODD样本认为是ID样本中的某一类，产生不正确的高置信度预测。

3. 基于深度模型的 OOD detection 首先由 Hendrycks 等人在 17 年提出了一个baseline, also called Outlier Detection or Novelty Detection.

4. 不仅可以找到OOD样本，同时也能够保证模型的performance.

基于深度模型的 OOD Detection 的方法

1. **Softmax-based:** 这类方法利用 pre-trained model 输出的最大 softmax 概率进行统计分析，统计发现 OOD 样本和 ID 样本 softmax 概率的分布情况，试图将二者的分布差距加大，然后选取合适的阈值来判断一个样本属于 OOD 还是 ID。

2. **Uncertainty:** 由于模型的概率输出并不能直接表示模型的置信度(confidence)。因此这类方法让模型学习一个对输入样本的不确定性属性。面对测试数据，如果模型输入为 ID 样本，则不确定性低，相反，如果模型输入为 OOD 样本，则不确定性高。

3. **Generative Model:** 这类方法主要利用 Variational Autoencoder 的 reconstruction error 或者其他度量方式来判断一个样本是否属于 ID 或 OOD 样本。

4. **Classifier:** 这类方法比较直接，使用分类器对提取的特征进行分类来判断是否为OOD样本。N+1分类器。

ODD-MAML

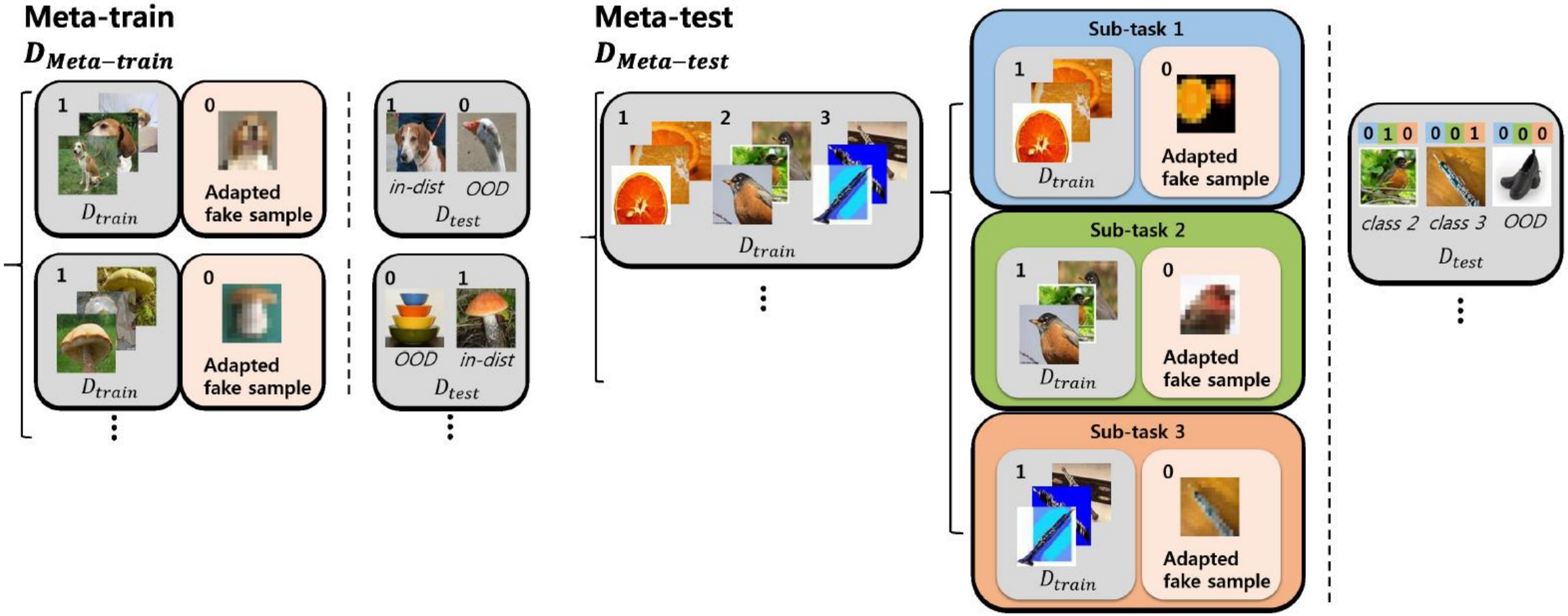
Few-shot OOD Detection with MAML. The training and testing procedures is used in our meta-learning method for few-shot OOD detection and classification.

1. Task formulation in OOD-MAML.

2. Meta-training procedure for OOD-MAML.

3. Meta-testing procedure for OOD-MAML

OOD-MAML


Meta-train
$D_{Meta-train}$
1
0
Adapted fake sample
D_{train}
1
0
in-dist
OOD
D_{test}
1
0
Adapted fake sample
D_{train}
0
1
OOD
in-dist
D_{test}
Meta-test
$D_{Meta-test}$
1
2
3
D_{train}
Sub-task 1
1
0
Adapted fake sample
D_{train}
Sub-task 2
1
0
Adapted fake sample
D_{train}
Sub-task 3
1
0
Adapted fake sample
D_{train}
0 1 0
0 0 1
0 0 0
class 2
class 3
OOD
D_{test}

Task formulation in OOD-MAML

1. $D_{meta-train}$ 和 $D_{meta-\text{test}}$ 分别是训练数据和测试数据.

2. $D_{train}^{i} \in D_{meta-train}$ contain K examples of one know class and one artificially generate OOD example. 通过这种方式先在meta-train里面来训练模型是属于ID还是OOD的样本。

3. During the test time, the algorithm evaluate the meta-trained model for both K-shot N-way classification and OOD detection.

4. $D_{train}^{j} \in D_{meta-\text{test}}$ contain K-shot N-way setting. Split into multiple sub-tasks T_{jn} that only samples of the nth class belong to the seen class for T_{jn}.

$$D_{train}^{j} = \{\{\mathbf{x}_{1k}^{j}\}_{1\leq k\leq K}, \{\mathbf{x}_{2k}^{j}\}_{1\leq k\leq K}, \ldots, \{\mathbf{x}_{Nk}^{j}\}_{1\leq k\leq K}\}$$

Meta-training procedure for OOD-MAML

1. If we adapt θ with a gradient update using MAML, the adapted base parameter would be biased and the adapted base model would become a trivial classifier because all the elements belong to the same class.

2. Introduce a fake-sample parameter vector $\theta_{fake} = (\theta_{fake,1}, \dots \theta_{fake,M})$.

3. Define an additional loss and obtain the new total loss

$$L_{\theta;T_i}(D^i_{train}, \theta_{fake}) = L^{in}_{\theta;T_i} + L^{out}_{\theta;T_i}(\theta_{fake})$$

$$\theta^i = \theta - \alpha \nabla_\theta L_{\theta;T_i}(D^i_{train}, \theta_{fake}).$$

4. Adapting θ_{fake} to each task T_i.

$$\theta^i_{fake} = \theta_{fake} - \beta_{fake} \odot sign(-\nabla_{\theta_{fake}} L_{\theta^i;T_i}(D^i_{train}, \theta_{fake}))$$

5. Reflect the adversarial input into the base model

$$\theta^i_{adapt} = \theta - \alpha \nabla_\theta L_{\theta^i;T_i}(D^i_{train}, (\theta_{fake}, \theta^i_{fake})).$$

$$(\theta, \theta_{fake}, \beta_{fake}) \leftarrow (\theta, \theta_{fake}, \beta_{fake}) - \gamma \nabla_{(\theta, \theta_{fake}, \beta_{fake})} \sum_{T_i \sim P(T)} L(D^i_{test})$$

Meta-testing procedure for OOD-MAML

1. Given sub-task T_{jn} from $D_{train}^{j} \in D_{meta-\text{test}}$, we update the θ_{adapt}^{jn} using the same manner.

2. Validate the adaptation to the samples in D_{test}^{j}.

$$p^{j}(x) = [f_{\theta_{adapt}^{j1}}(x), \ldots, f_{\theta_{adapt}^{jN}}(x)]$$

Simulation Results 1

Table 1: OOD detection results

	ODIN -MAML	ODIN -PN	MAH -MAML	(N+1) -MAML	(N+1) -MAML*	OOD -MAML(M=3)	OOD -MAML(M=5)
Omniglot							
detect.acc $\alpha = 95$	0.8744 (0.0512)	0.8977 (0.0441)	0.8712 (0.0481)	0.9142 (0.0392)	0.9524 (0.0341)	0.9712 (0.0296)	0.9701 (0.0288)
detect.acc $\alpha = 98$	0.8912 (0.0331)	0.9122 (0.0287)	0.8320 (0.0785)			0.9838 (0.0225)	0.9833 (0.0214)
TNR $\alpha = 95$	0.6942 (0.1142)	0.7122 (0.0533)	0.7288 (0.0821)	0.8722 (0.0730)	0.9201 (0.0633)	0.9924 (0.0224)	0.9918 (0.0225)
TNR $\alpha = 98$	0.7124 (0.0988)	0.7369 (0.0629)	0.7544 (0.0233)			0.9831 (0.0342)	0.9839 (0.0359)
CIFAR-FS							
detect.acc $\alpha = 95$	0.5811 (0.1022)	0.5933 (0.1113)	0.5601 (0.0891)	0.5035 (0.1299)	0.5531 (0.1021)	0.6752 (0.0738)	0.6612 (0.0813)
detect.acc $\alpha = 98$	0.6129 (0.1132)	0.6039 (0.1285)	0.5458 (0.0671)			0.6590 (0.0719)	0.6594 (0.0725)
TNR $\alpha = 95$	0.2311 (0.1291)	0.1592 (0.1422)	0.2999 (0.1239)	0.1051 (0.1833)	0.2017 (0.0945)	0.5492 (0.1250)	0.5512 (0.1311)
TNR $\alpha = 98$	0.2401 (0.1087)	0.1439 (0.1027)	0.1862 (0.1244)			0.4317 (0.1287)	0.4198 (0.1249)
***mini*ImageNet**							
detect.acc $\alpha = 95$	0.5124 (0.0742)	0.5491 (0.0981)	0.5111 (0.1124)	0.5019 (0.0712)	0.5422 (0.1101)	0.6207 (0.0736)	0.6199 (0.0744)
detect.acc $\alpha = 98$	0.5641 (0.0411)	0.5669 (0.1003)	0.5229 (0.1174)			0.6125 (0.0750)	0.6024 (0.0688)
TNR $\alpha = 95$	0.1211 (0.1899)	0.1829 (0.1042)	0.1429 (0.1366)	0.0749 (0.0822)	0.1009 (0.1033)	0.6770 (0.1181)	0.6613 (0.1203)
TNR $\alpha = 98$	0.1659 (0.1426)	0.1942 (0.0819)	0.1944 (0.1209)			0.4902 (0.1310)	0.4891 (0.1287)

Simulation Results 2

Table 2: Classification accuracy results. M=3 for OOD-MAML

Method	Omniglot	CIFAR -FS	*mini* Imagenet	Method	Omniglot	CIFAR -FS	*mini* Imagenet
MAML (K=5,N=5)	0.9911 (0.0371)	0.7084 (0.1230)	0.5926 (0.1086)	OOD-MAML (K=5,N=3)	0.9996 (0.0054)	0.7220 (0.1121)	0.6322 (0.0989)
OOD-MAML (K=5,N=5)	0.9989 (0.0071)	0.7158 (0.1129)	0.6044 (0.1098)	OOD-MAML (K=5,N=7)	0.9894 (0.0287)	0.6964 (0.1139)	0.5822 (0.1319)

总结和思考

1. 这篇文章主要是提出通过使用类似MAML的算法来计算一个能够区别OOD样本，同时也能完成它本职分类任务的深度模型。

2. 这个方法主要是运用n+1的分类器来检测OOD样本的同时也能保证算法的对分类任务的有效性。基于这个算法，能不能够使算法能够同时适应于两种不同分布数据之间的分类。

3. 目前的OOD-MAML主要是用于分类任务，对于预测类的任务模型，我如果同时使用两种不同分布的数据训练模型，能不能使用类似的sub-task模型能够快速适应任何一种分布数据的新任务的预测。或者说类似于bi-level方法，快速使用不同分布训练好的模型直接预测？

4. 基于这个OOD检测算法，在智能电网里面，我们也需要用到异常数据的检测，特别是智能电网里面ICT系统的加入，使系统更加容易受到网络攻击，所以AI检测的运用是非常有必要的。