

ICLR2020

背景：持续学习

持续学习的问题在于如何学习既定模型的情况下，加入新的模型（甚至，当我们不知道原始训练数据时，如何保留原有的模型）。

灾难性的遗忘：当学习者从一个训练过的任务转换到一个新的任务时，其表现显著下降。这种现象的发生是因为在最初的任务中的训练参数的改变有利于学习新的目标（而不是旧的）。

背景：贝叶斯网络

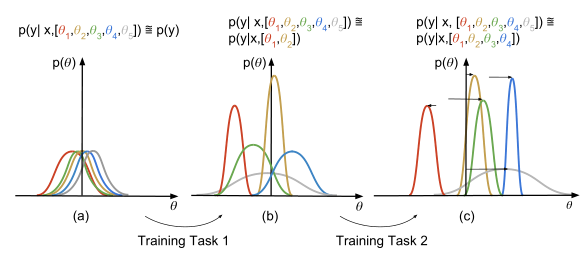
原生考虑了不确定性。这些网络将每个参数定义成了一个分布（本文均假设为了一个高斯分布），有均值和方差，这个分布即是对参数估计的可能值的所有情况。

如何求解每个参数的可能的概率分布呢？变分推断（Variational inference）

如何求解梯度的估计值呢？蒙特卡洛采样

本文目标：

提出UCB网络，该网络中，学习率是按照网络中的权重的概率分布的不确定性来设置的。



上图中，a有五个不确定变量。经过task1，theta1和theta2的不确定性降低，其余的还是很高（用于学习新的task）。再经过task2，theta1-4不确定性降低（在这里，theta1-2的学习率改变很低，主要改变theta3-4）。以此类推，继续学习…

问题：如何确定哪个theta是确定的和不确定的？

Bayes by backprop(BBB):

在贝叶斯模型中，潜在变量从先验密度p(w)中提取，先验密度p(w)通过概率p(x|w)与观测值相关。在推断过程中，根据给定的输入数据计算后验分布p(w|x)。然而，在实践中，这种概率分布是很难的，往往是通过近似推理估计。

* 蒙特卡罗抽样是其中一种估计方式。然而，该算法尽管提供了从目标分布中寻找渐近精确样本的保证，但并不适合于大型数据集和模型，因为它们受到速度和可伸缩性问题的限制。
* 于是，变分推理提供了一个更快的解决方案，其中后验是使用优化近似而不是从链中取样。变分推理利用优化方法，如随机方法或分布方法，使其能够快速探索数据模型。

是观测到的变量，w是因变量。

神经网络（即概率模型）：在给定数据集的情况下，找到最佳w使得P最接近真实的P。

如何度量概率分布？不能使用范数（点到点），应当使用KL散度或JS距离。

假设一系列的高斯分布，他们的均值和（准）方差为。

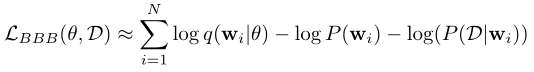
由于目标是，因此优化问题：

**

该argmin函数等价于：



使用蒙特卡洛采样w（N个），进行变分后：



也就是说，目标loss即上述表达式。

考虑：

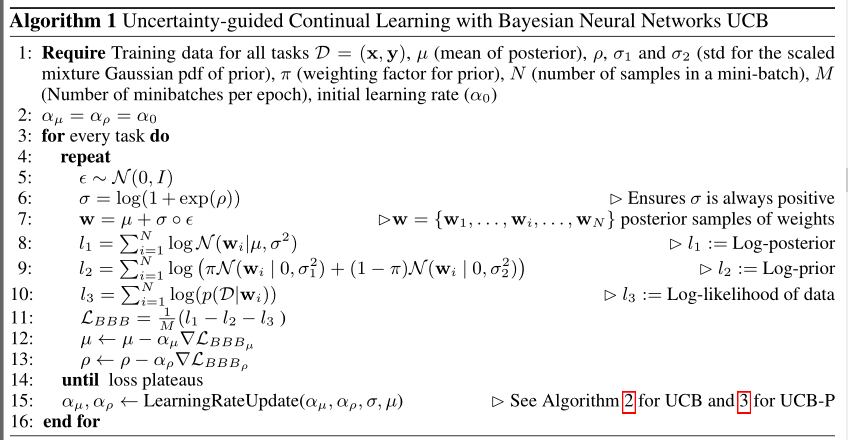
为了确保方差大于0，标准差由以下式子得到：



由此，w可以由高斯分布随机获取N个（是噪声）：



UCB算法：



算法解释：根据高斯分布随机获取若干w，计算的loss，并以此作为梯度下降的依据，拟合到合适的散度。

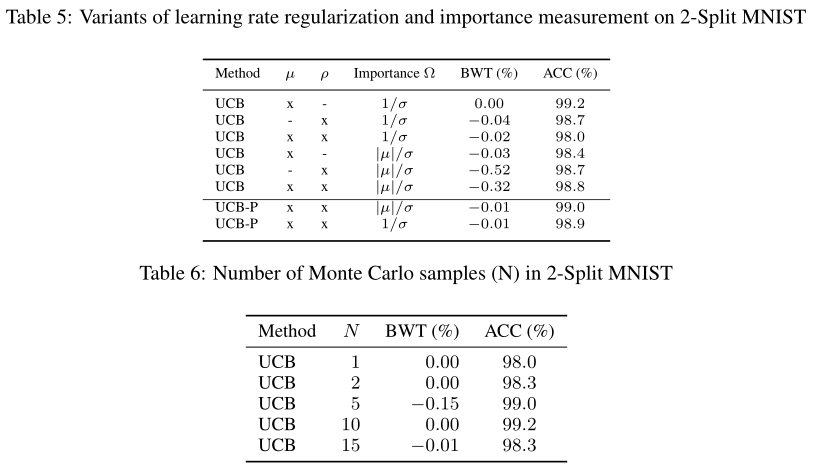
参数更新方法1：学习率作为重要性的更新

在UCB中，正则化是按照（每个参数的）学习率来执行的，因此参数的梯度更新就变成了它的重要性的函数。对每个采样的和，学习速率按其重要性成反比。由此，可以减少重要参数的变化，同时允许不太重要的参数进行更多的改变，以有利于学习新的任务。

其中，与成反比，即：



具体的公式是通过消融实验得到的，即：



优势：

以学习率作为正则化的UCB的关键好处是，它不需要额外的内存，而不是剪枝技术，也不需要跟踪与之前学习任务相关的参数变化，这在常用的权重正则化方法中是需要的。更重要的是，该方法不需要意识到任务切换，只需要根据后态分布中的均值当前的不确定性调整其学习速率。

参数更新方法2：剪枝（weight pruning）

使用binary mask标记是否被剪枝，以避免遗忘。

重要性依据：贝叶斯神经网络中定义的统计基础不确定性

定义信噪比SNR为：



SNR是信号处理中用来区分信号中“有用的”信息和不需要的噪声的常用量度。在神经模型的背景下，信噪比可以被认为是参数重要性的重要指标：信噪比越高，表明该参数对模型预测的有效性或重要性。

算法：对于每一层，无论是卷积的还是全连通的，参数按其信噪比值排序，重要性最低的参数被修剪(设为零)。修剪后的参数用二进制掩码进行标记，以便以后学习新任务时使用，而重要参数在未来任务的训练中始终保持不变。一旦学习了一个新任务，就会保存一个相关的二进制掩码，在推理期间将使用该掩码恢复关键参数，从而获得所需任务的确切性能。

在编码掩码以及将其保存到磁盘上时，每个参数的开销内存如下所示。假设我们使用一个网络学习n个任务，为一个参数编码一个累积的掩码所需的总位数是最大位。

实验：

数据集：5分MNIST和permuted MNIST

以及其它数据集FaceScrub, MNIST, CIFAR100, NotMNIST, SVHN, CIFAR10, TrafficSigns FashionMNIST作为补充。

Baseline1：三个不考虑参数重要性的模型，即微调、特征提取和联合训练

在微调(BBB-FT)中，在没有避免遗忘的情况下，训练在新任务到达后继续进行。

在特征提取(BBB-FE)中，在训练第一个任务后冻结网络中的所有层，对剩余的任务只训练最后一层。

在联合训练(BBB-JT)中，我们以多任务学习方式联合学习所有任务，这种多任务学习方式作为所有任务平均准确率的上界，因为它不遵循连续学习场景。

Baseline2：使用FT、FE和JT模型中的普通神经网络（即不包括上述特性的）ORD-FT、ORDFE和ORD-JT。

Baseline3：和目前主流方法

Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017),

Incremental Moment Matching (IMM) (Lee et al., 2017),

Learning Without Forgetting (LWF) (Li & Hoiem, 2016),

Less-Forgetting Learning (LFL) (Jung et al., 2016),

PathNet (Fernando et al., 2017),

Progressive neural networks (PNNs) (Rusu et al., 2016)

Hard Attention Mask (HAT) (Serra et al., 2018)

Baeline4：其余论文中报告的精度（本文没实现）

Permuted MNIST：SI (Zenke et al., 2017) from (Serra et al., 2018).

Split and Permuted MNIST：VCL (Nguyen et al., 2018)、VCL-GNG (Chen et al., 2019)、VCL-V adam (Tseran et al., 2018)

因为我们的方法是基于regularization的，所以我们只比较那些没有episodic或coreset的baseline。

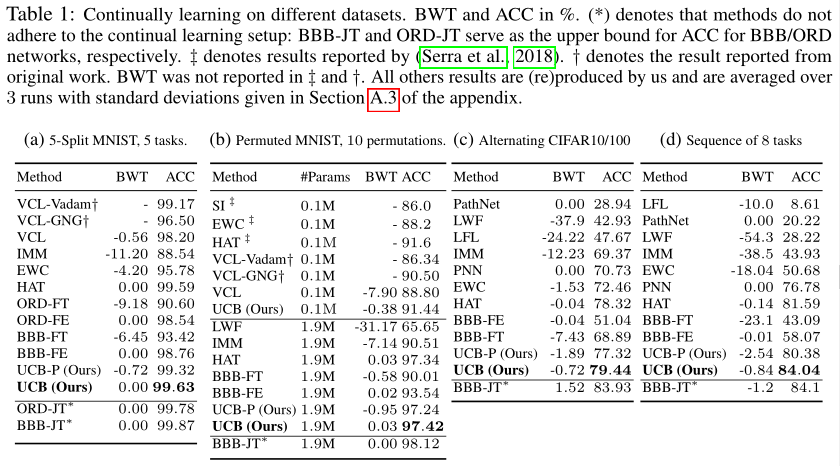
评估方式：

BWT 大于0表示学习新任务对旧任务有帮助，小于0有负影响。

ACC 所有任务的平均acc



数据（UCB效果更好）：

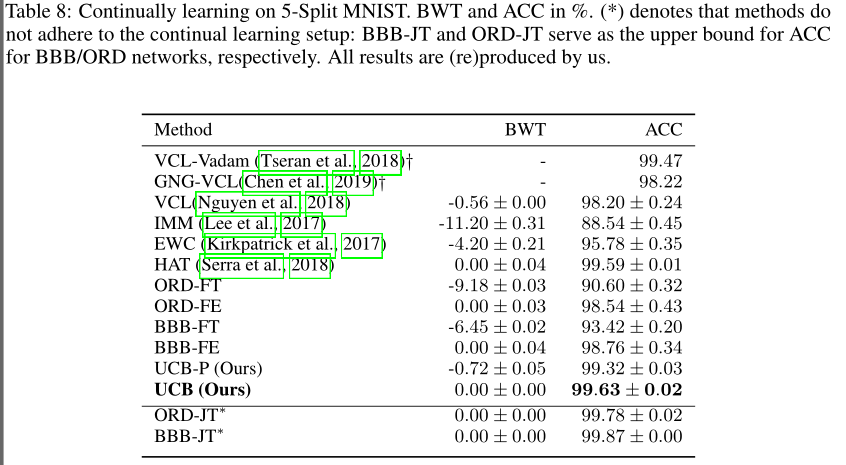


Permuted mnist：将MNIST里面每张图片向量化后进行重排列，其实就是用一组随机索引来打乱向量里面每个元素的位置，不同的随机索引打乱后产生不同的任务。

Alternatingcifar10-100：每个数据集分成5个任务，交替10个数据集，每个数据集有2个或20个标签。

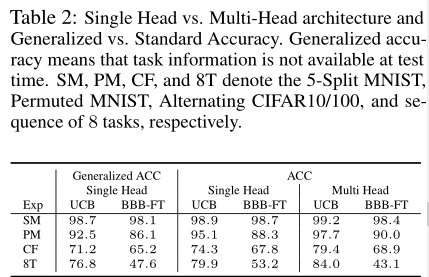
多数据集：学习8个任务

5-split mnist：每次学习2个数字



泛化acc：

即使在测试时没有给出任务信息，也可以使用UCB。为此目的，在训练时，我们不是为每个任务使用一个单独的完全连接的分类头，而是使用一个包含所有任务输出总数的分类头。例如，在8个数据集的实验中，在训练和推断期间，我们只使用一个有293个输出类的head，而不是使用8个单独的head。表2后4列是使用单头训练的UCB和BBB-FT与使用多头架构的结果。UCB从多头训练到单头训练的性能下降很小（0.3%、2.6%、5.1%和4.1%）。



前两列的数据是覆盖了所有任务中的类后得到的结果，有下降0.2%、2.6%、3.1%和3.1%，可能是在测试时出现更多的类导致了不明显的混淆。

这表明UCB可以在更复杂的条件下（比如测试时任务信息未知时）使用。

可能的原因：直接使用预测的权重不确定性来寻找重要参数，而不是计算通常依赖于任务、输入或输出的评价指标。