智能计算技术大作业

课程论文

Intelligent computing technology

姓名：李广通

学号：919106840421

专业：计算机科学与技术专业

学院：计算机科学与工程学院

南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

目录

1.FF 算法具体方法步骤

1.1 背景

杰弗里·辛顿(Geoffrey Hinton)，谷歌副总裁兼工程研究员、Vector 研究所首席科学顾问、多伦多大学名誉教授，亦是伦敦大学学院（UCL）盖茨比计算神经科学中心的创立者，是神经网络泰斗。

机器学习领域伴随着硬件 GPU 的发展和支持逐渐壮大，模型方法也多种多样，一系列包括逻辑回归、神经网络、Transformer 等技术被学者和企业专家所提出。其中最为常用的**神经网络**发展最为迅猛，业界和各种计算机学术会议上学者们也对其进行了各种各样的优化。

但神经网络发展至今，权重和参数更新方式大同小异，慢慢的模型规模逐渐增大，大体量模型的发展大行其道。**要学习 FF 算法的理念，首先要了解其 motivation。**

通读作者的论文，我总结出当前的神经网络存在以下问题，并列出当前的发展需求：

1. 网络规模增大，复杂网络对算力的高要求使得无法在各异的设备上部署运行（例如移动端处理器中 GPU 的能效完全不足以支撑大体量模型的训练和执行开销）

——>需要高能效比的软硬件体系结构和适当的模型算法调整来对更加多样的设备提供深度学习支持

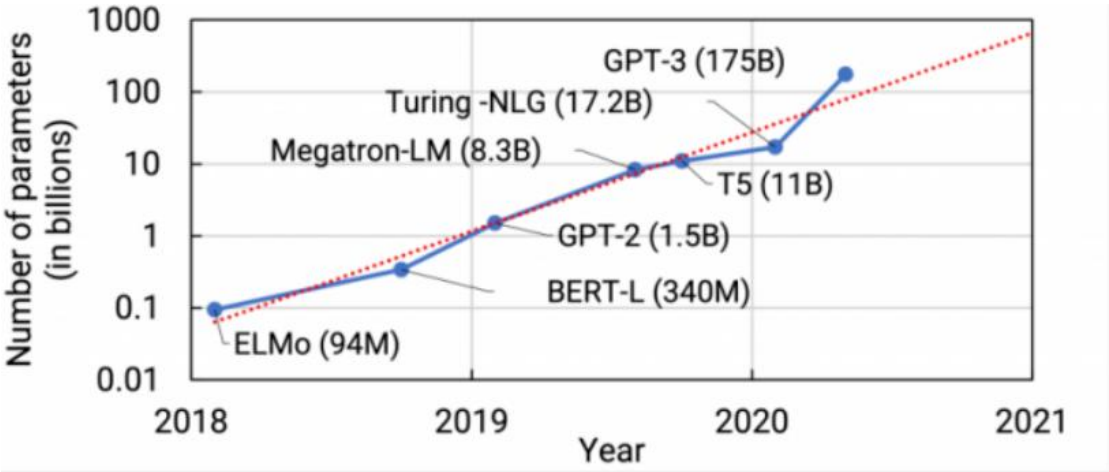
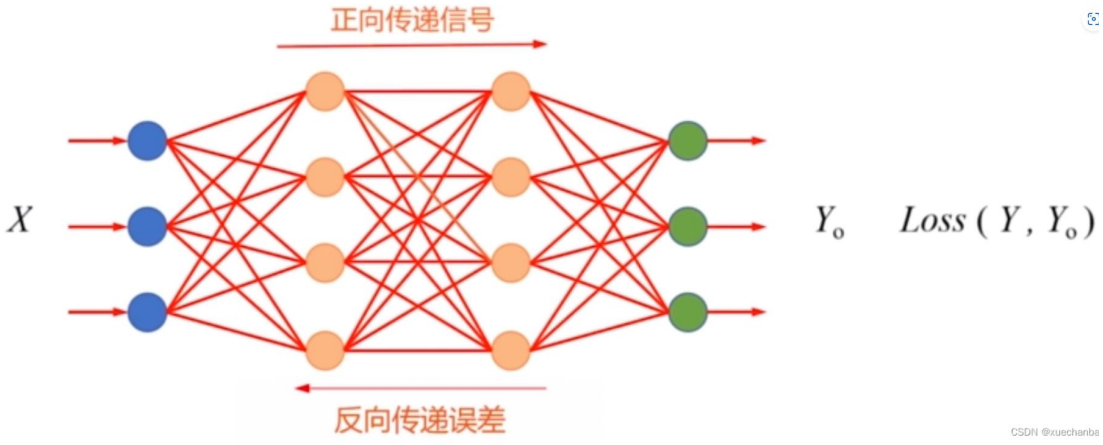


Figure. Trend of state-of-the-art NLP model sizes with time.

2. 反向传播算法本身带来一系列问题和思考。

2.1 反向传播算法本身并不符合类脑学习的生物实际。反向传播算法的具体思路是正向转递信号，通过隐藏层和输出层的误差反向传递，对每一层的参数和权重进行更新调整。然而，现实是：**大脑学习序列不能使用基于时间的反向传播机制。**为了在**不频繁暂停**的情况下处理感知输入流，大脑需要通过感觉**处理不同阶段的感官数据**，也需要一个**即时**学习过程。人们通过各种生物学验证也发现神经元并没有刺激反向传播的机制，而是不断地进行正向循环。

——>可以考虑使用正向迭代来更好的拟合生物学习方式



2.2 反向传播算法对函数的要求相对苛刻，**变相牺牲了一部分可行解的空间**。反向传播的另一个严重限制是，它需要完全了解正向传递中执行的计算过程**（函数必须可导）**，才能计算正确的导数。这是因为后向传播的前提是对前向计算的导数模型要非常明细

——>需要考虑如何使用鲁棒性更强的算法来扩大可行解空间

为了解决上述反向传播的问题，使用正向迭代来取缔反向传播，也有学者尝试使用强化学习来模拟类脑计算的流程方法。这个想法是对权重或神经活动进行随机扰动，并将这些扰动与由此产生的收益函数变化相关联。但顾此失彼，消除了后向传播，强化学习又带来了高方差问题，在大型网络上的扩展性差，对大量参数情况下的处理有失偏颇。

在以上背景下，Forward-Forward 算法（FF）被提出。

1.2 总体思路

本文的主要目的是证明包含未知非线性的神经网络不需要求助于强化学习就可以消除反向传播的形式。

总体思路如下：

将**反向传播的前向和后向传递替换为两个前向传递，一个具有正（真实/源）数据，一个具有网络自身生成的负数据**，每一层都有自己的目标函数，即对正数据具有较高的优度，对负数据具有较低的优度。（优度指匹配程度），层内活动平方和可以用作优度，当然还有许多其他可能性，包括减去活动平方和。

如果正传递和负传递能够及时分离，则负传递可以离线完成，这使得正传递中的学习更加简单，并允许视频在网络中流水线化，而无需存储活动或停止传播导数（梯度），这使得模型的学习过程满足了类脑计算中的**即时性**。

1.3 具体步骤

生 Forward-Forward 算法是一种贪婪的多层学习程序，其**灵感来自玻尔兹曼机和噪声对比估计**

其思想是用两个正向传递来代替反向传播的正向传递和反向传递，这两个正向传播以彼此完全相同的方式操作，但在不同的数据上操作，目标相反。

迭代方式：

1. 正传递（Positive pass）对真实数据（正样本）进行操作，并调整权重以增加每个隐藏层的优度（完成权重的增益操作）

2. 负传递（Negative pass）对“负数据”（负样本）进行操作，并调整权重以降低每个隐藏层中的优度（完成权重的减益操作）

本文探讨了两种不同的优度——平方神经活动的和和平方活动的负和，其他度量标准也是可以的，但这里为了简便，使用这两者作为增益和减益的基础。

即假设一层的**优度函数是该层中整流线性神经元活动的平方和（好处有两点：1.有相对简单的导数 2.层归一化可以消除所有的优度）**

模型训练目标及方法：

学习的目的是使正样本的优度远远高于某个阈值，而负样本的优度远远低于该阈值，即将输入向量正确分类为正数据或负数据（这样就可以通过 relu 进行判断，例如可以使用 sigmoid 函数）

基于阈值的分类模型入下所示：

$$p(positive) = \sigma\left(\sum_j y_j^2 - \theta\right)$$

此图表示为正数据的概率，就是基于上述训练目标和方法设计的。特别要说明的是负数据可以由神经网络使用自上而下的连接进行预测，也可以由外部提供。

很容易看出，通过使隐藏单元的和平方活动对于正数据为高而对于负数据为低，对单个隐藏层的神经网络推理非常简单，但是如果推广到多隐藏层，即第一个隐藏层的活动随后被用作第二个隐藏层的输入，那么通过简单地使用第一个隐藏层中活动向量的长度来区分正负数据是没有意义的，**相当于重复分析同一特征，学习陷入停滞。为了解决学习陷入停滞的问题，可以在向下一隐藏层传输前使用层归一化对当前层已经学习过的优度进行屏蔽，这样可以保证每一层进行学习时的特征不重复**。换句话说，第一个隐藏层中的活动向量具有长度和方向。长度用于定义该层的优度，只有方向被传递到下一层。

关于负样本的创建：

为了迫使 FF 专注于表征形状的图像中的长期相关性，我们需要创建具有非常不同的长期相关性但非常相似的短期相关性的负数据。这可以通过创建一个包含相当大的 1 和 0 区域的掩码来完成。然后，我们通过将一个数字图像乘以掩码和一个不同的数字图像乘以掩码的反转（取逆）相加来为负数据创建混合图像，可以从随机位图开始创建这样的掩码，然后然后在水平和垂直方向上使用［1/4, 1/2,1/4］形式的过滤器重复模糊图像。反复模糊后，图像的阈值设为 0.5。

下图就是 7 和 6 图片样本创建负样本的过程：使用掩码和掩码取逆进行矩阵乘法后混合。

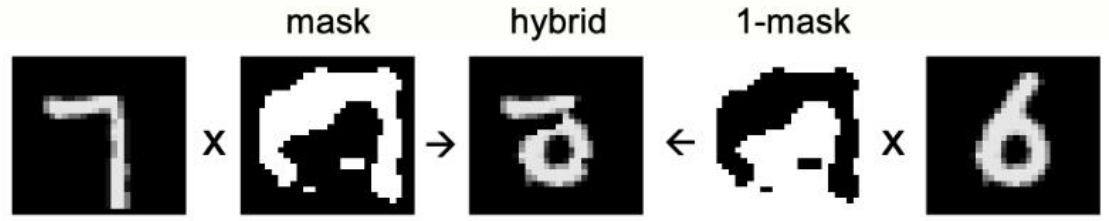


Figure 1: A hybrid image used as negative data

1.4 实验效果

实验发现针对**大型网络，FF 模型收敛速度相比反向传播算法要慢**；而功率上由于消除了求导计算过程，**FF 模型功耗水平优势更大**，FF 算法的可行性验证有助于将大体量模型的训练过程移植到算力相对羸弱的移动端和嵌入式设备上。

论文中描述的大部分实验都使用手写数字的 MNIST 数据集，MNIST 的反向方向传播模型非常广泛，这使得对新算法的检验具有很好的参照性。**基准选择**

的是具有多隐藏层的前馈神经网络在不使用复杂正则化器的情况下，置换不变任务版本的测试误差是 **1.4%**

无监督训练版本 FF 算法在用四个隐藏层（每个隐藏层包含 2000 个 ReLU）训练 100 个时期后，如果我们使用最后三个隐藏层的归一化活动向量作为 softmax 的输入，我们得到 **1.37%** 的测试错误率，该 softmax 被训练来预测标签。使用第一个隐藏层作为线性分类器输入的一部分会使测试性能变差。我们可以使用局部感受野（没有权重共享）而不是使用完全连接的层，这可以提高性能。经过 60 个 epoch 的训练后，它给出了 **1.16%** 的测试错误。它使用隐藏活动的"对等标准化"来防止任何隐藏单元极度活跃或永久关闭

监督学习版本的 FF 算法： MNIST 图像包含黑色边框，使卷积神经网络的工作变得轻松。如果我们用标签的 N 个表示中的一个替换前 10 个像素，则很容易显示第一个隐藏层学到的内容。一个有 4 个隐藏层的网络，每个隐藏层包含 2000 个 ReLU，层与层之间的完全连接在 60 个 epoch 后在 MNIST 上得到 1.36% 的测试错误。反向传播需要大约 20 个 epoch 才能获得类似的测试性能。将 FF 的学习率加倍并训练 40 个 epochs 而不是 60 个 epoch 会得到稍差的测试误差，即 1.46% 而不是 1.36%。

可以对比得知 FF 算法在轻量级模型的精准度上相比反向传播不会有折损，可行性得到了验证。

作者还使用了 CIFAR-10 进行验证

learning procedure	testing procedure	number of hidden layers	training % error rate	test % error rate
BP		2	0	37
FF min ssq	compute goodness for every label	2	20	41
FF min ssq	one-pass softmax	2	31	45
FF max ssq	compute goodness for every label	2	25	44
FF max ssq	one-pass softmax	2	33	46
BP		3	2	39
FF min ssq	compute goodness for every label	3	24	41
FF min ssq	one-pass softmax	3	32	44
FF max ssq	compute goodness for every label	3	21	44
FF max ssq	one-pass softmax	3	31	46

2.与一般神经网络的异同

2.1 与神经网络（感知机）的异同

部分异同点在背景部分就已经做过讲解并进行了图示，这里简单重复归纳一下

相同点：

信号传播方向相同，都是正向传播

不同点：

1. 迭代方法不同。FF 使用**两个前向传递，而感知机的信号前向传播，误差后向传播。**

2. 样本分类不同。FF 分为正样本和负样本，而感知机没有样本分类

3. 隐藏层要求不同。，感知机需要完全了解正向传递中执行的计算过程**（函数必须可导/可微）**，而 FF 没有这样的要求

4. 功耗水平不同。FF 功耗更占优势，而感知机功耗表现糟糕

5. 算法可扩放性不同。FF 在大型网络上的实用性还有待后续验证，而感知机的应用非常全面。

2.2 和玻尔兹曼机的关系

玻尔兹曼机可以看作是两种思想的结合：

1.通过最小化真实数据上的自由能和最大化网络本身生成的负数据上的自由能来学习。

2.使用 Hopfield 能量作为能量函数，并使用重复随机更新从能量函数定义的 Boltzmann 分布中采样全局配置。

FF 将玻尔兹曼机的对比学习与简单的局部优度函数相结合，该函数比二元随机神经元网络的自由能更易于处理。

2.3 和生成对抗网络的关系

FF 可以看作是 GAN 的一个特例，其中判别网络的每个隐藏层都自己贪婪地决定输入是正还是负，因此不需要反向传播来学习判别模型。也不需要反向传播来学习生成模型，因为它不是学习自己的隐藏表示，而是重用判别模型学到的

表示。生成模型唯一需要学习的是如何将这些隐藏表示转换为生成数据，如果使用线性变换来计算 softmax 的对数，则不需要反向传播。对两个模型使用相同的隐藏表示的一个优点是它消除了当一个模型相对于另一个模型学习太快时出现的问题。它还消除了模式崩溃。

2.4 非永生计算（新概念）

文中提到了一个全新的概念：mortal computation，这一概念目前还没有官方的翻译，根绝个人阅读论文后理解，可以称之为非永生计算。这一概念的提出是对计算机体系结构设计的挑战——从软硬件分离，到现在 FF 计算带来重新耦合的可能。使用快速权重方法来在特质化的硬件之间进行模型中间运行结果的迁移（用来替代之前的上下文保存在不同硬件之间的特质化问题）

目前体系结构坚持认为软件应该与硬件分离，这样相同的程序或相同的权重集就可以在硬件的不同物理副本上运行。**（为了可移植性要牺牲更好的针对性优化）**这使得包含在程序或权重中的知识是不朽的:当硬件死亡时，知识不会死亡**（软硬件解耦合带来的一点优势）**

软硬件分离的优势：

1. 它使研究程序的性质而不用担心硬件电路成为可能。（软件开发和硬件电路解耦合，对纯算法的研发带来更大遍历）

2. 它使编写一次程序并将其复制到数百万台计算机上成为可能（兼容性提升，统一编译器使得移植性大幅提升）

然而，**然而，如果我们愿意放弃软件体系的永生（高扩展和可移植性），就有可能在执行计算所需的能量和制造执行计算的硬件成本方面实现巨大的节省（通过牺牲通用性来节约单体成本）**

这些参数值只对特定的硬件实例有用，因此它们执行的计算是非永生的:它会随着硬件的消失而消失**（运算的硬件鲁棒性高，这是通过改变模型结构和计算方式来实现的，因此中间变量是硬件相关的而不是硬件无关的）**因此将学习到的参数值和权重直接复制到另一个工作方式不同的硬件上没有意义，但有一种更生物学的方式可以将一个硬件学到的东西转移到另一个硬件上。通过使用蒸馏，功能本身可以转移到不同的硬件上。新硬件经过训练，不仅能给出与旧硬件相同的答案，还能输出与旧硬件相同的错误答案的概率。因此，通过训练新模型来匹配错误答案的概率，我们训练它以与旧模型相同的方式进行泛化**（模型的计算方式在不同硬件之间转移需要重新进行拟合）**

通过软硬件的重新拟合，我们可以在特质化很强的设备上也取得很好的效果，不过是以牺牲不同硬件之间的通用性为代价。而上下文切换和模型在不同硬件之间的转移变得困难，但可以通过快速权重来进行粗略的基础拟合，后面再对新硬件进行特定的拟合过程。

2.5 总结（FF 优劣）

优势：

1. FF 使得人类再类脑学习领域的模拟更加贴近真实情况
2. FF 使得神经网络的可行解空间得到扩大，模型的设计灵活性提升（不再受限于感知机的函数可微条件限制）
3. FF 可能带来软件和硬件的深度融合，创造非永生计算下新的计算机体系结构设计
4. 功耗水平优秀，相比反向传播能耗更少，使得弱算力设备上运行更加大型的神经网络成为可能。

劣势：

1. 验证较少，针对大型网络可扩放性的验证还没有进行
2. Negative pass 和 positive pass 如何分离并将 negative pass 离线运行还没有非常具体的方法

3.优化改进思路

关于改进思路，有以下三个思路，从不同角度入手

3.1 进化算法

智能计算课上学习过蚁群算法和遗传算法，这里作者在论文中表述多隐藏层情况下的信息传递问题可以使用层归一化来处理，消除已经提取过的信息内容，而且负样本的采集和创建方法也比较单一和欠缺。

我认为可以引入遗传算法来解决上述问题。首先针对多隐藏层，遗传算法对目标函数的要求低，对前层屏蔽效果可能更好，而 negative pass 更多依赖负样本的质量，遗传算法的突变随机特性可能能够带来更好的负样本空间，从而使得负增益更加有效。

也就是说，现在的计算机硬件追求绝对的精确，没有给随机性保留空间，这点在自建的负样本中会成为瓶颈，使用遗传算法适当在可控范围内加强随机性可以缓解上述问题。

3.2 个人想法

个人认为可以从以下两个角度进行优化：

1. 引入混合精度计算（便于快速权重），**提升模型收敛速度**。我研究生预计会主攻模型压缩和模型适应方向。最近看了关于混合精度计算的论文 Campo: Cost-Aware Performance Optimization for Mixed-Precision Neural Network Training（ACT'22），对本文章有了一定的启发，我认为可以考虑在 FF 计算中也是用混合精度计算，对静态的模型图进行更加激进的精度调整，针对动态流图就相对保守，避免为了加快收敛速度对模型精度产生损失。**文章中说 FF 算法在大型网络下收敛速度相比反向传播要慢，我认为更加细粒度的混合精度计算或许是一种可行的解决方案。**

2. 使用模拟硬件，改进梯度表示方法，**降低能耗**。将活动向量乘以权重矩阵的一种节能方法是将活动实现为电压，将权重实现为电导**（利用硬件的更多物理学特性，感觉和三进制计算机对电压数值的利用有异曲同工之妙）**这似乎比在高功率下驱动晶体管对一个数字的数字表示中的单个位进行建模，然后执行 O(n2)个单位操作将两个 n 位数字相乘要合理得多。（通过更多得利用线性变化的物理学定律，可以减少使用常规二值计算带来的低效计

算功耗，但个人认为要考虑很多损失，同时需要在体系结构的层面进行优化）这样可以提高计算效率，更近一步发挥 FF 的能耗优势。

3.3 非永生计算及相关总结

FF 可以将硬件和软件重新耦合，充分利用特质化硬件计算资源。现在的计算机，特点是硬件 非常赢，软件非常软，软件与硬件是可分离的，可以无限拷贝，一次编码无限使用。

现在的常规软件是 immortal 的，也就是说，就算这个硬件坏了，软件仍然可以在其他地方运行。

FFA 与现有的数字计算机系统有根本的区别：

1. 它的硬件不是现在这种用不犯错的数字芯片，而是模拟器件或者忆阻器等，模拟器件的运行是收随机噪声影响的。

2．AI 程序是生长在硬件上的，程序的参数的学习过程会受到硬件中的随机性的影响。软件是 mortal 的，是会死的，也就是说，如果硬件坏了，软件也就失效了，因为软件完全与硬件绑定。反过来，也可以认为它是活的，因为只有活的才会死。

可能是人的直觉，觉得这种符合生物学逻辑的仿生计算方式才是未来 AI 计算中软硬件分配的最终选择——在特质化硬件的基础上既尊重随机性和噪声，也追求模型的特性。