理解深度学习

许志钦

2023 年 11 月 28 日

目录

Chapter 1

频率原则

为了研究一个方法的泛化性，我们有必要把研究分成两个部分，一个是算法的特征，另一个是数据的特征。如果算法和数据的特征是一致的，那算法的泛化性能就会好，若不一致，则泛化不好。在这一章中，我们将研究神经网络在频域的训练过程来理解神经网络算法的一种特征，也就是

频率原则：神经网络常常会以从低频到高频的顺序拟合训练数据。

本章节的实验来源于频率原则的代表作：Xu et al. (2020)。最初始的想法来源于频率原则的第一篇文章：Xu et al. (2019)。

1.1 为什么要研究训练过程

科学问题很多是相通的，科学研究方法往往也是可以相互借鉴的。为了研究一个现象，我们通常可以从这个现象的形成过程入手，理解这个现象是如何一步一步形成的。举个例子，假设我们要研究行星的运动，为什么行星会绕着恒星转 (图1.1)。以前，探测行星和恒星是非常困难的，我们要做实验也是很难的，因为我们很难操作这么大的星球，但是我们可以把这样一个问题转化成另一个相对容易操作的问题，比如我们要给地球发射卫星。卫星绕着地球转和行星绕着恒星转是类似的。那么现在我们的问题是我们怎么把卫星打到天上去。

为了把卫星打到天上去，我们可以考虑这么一个实验 (图1.2)，假设我们手上有一个石头，我们用力的往前扔，当我们的力气比较小的时候，石头会落在离我们比较近的地方。当我们更用力把石头往远的地方扔的时候，它会掉在地表更远的地方。考虑到地球是一个球形状的物体，如果我们用的力气足够大的话，那么它会不会转了一圈都还没掉下来？假设能量是守恒的，那当它又回到初始被扔出去的地方的时候，相当于说，它又以相同的能量被扔出去了。显然，这个石头就会不断的围绕地球在转。所以我们要研究怎么把卫星发射上去可以转化成研究扔石头

的过程。通过这个例子，我们可以看到研究一个现象的形成过程对理解这个现象背后的本质是非常重要的。在机器学习当中也是类似的，当我们发现一个很重要的现象的时候，我们要理解这个现象，我们就必须知道形成这个现象的过程是什么样子。

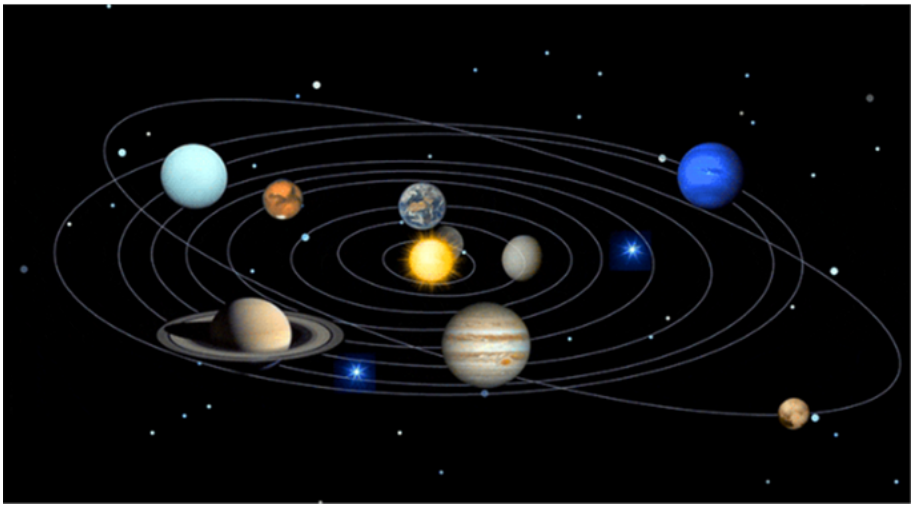


图 1.1

1.2 为什么从频率角度研究训练过程

神经网络这个领域受到工业的影响非常明显，它的几次起落都和它在实际问题的性能有关。前两次迭入低谷的原因很大程度上是因为神经网络在实际问题中并没有取得人们所预期的结果。而最近这一次神经网络能够获得大量关注的原因是它在图像识别、语音处理、机器翻译等应用上比传统方法有非常大的提升。正是因为神经网络的实际问题的关系非常紧密，涉及神经网络的研究也自然地紧紧围绕实际问题。很多算法能不能发表的潜在标准（甚至是公开且唯一的标准）是它能不能在公开的常用的标准数据集上取得最好的结果。甚至研究神经网络本身的现象和理论也被要求需要在真实数据上验证。注意，在真实数据上验证本身并不是不对的事。但一切唯真实数据论的想法会带来研究上的困难。比如，MNIST 通常被认为是分类问题中最简单的数据集。但即使是 MNIST，它的输入维度也是 784 维。对于科学研究来说，这是极其高维的问题。一般的科学计算的方法很少能处理这么高维的数据。另一方面，对于如此高维的数据，我们要去挖掘与它相关的现象时，我们既不能对数据做精确的可视化，也很难寻找合适地量去刻画学习过程。有时为了刻画高维现象而定义的量，也因为理论指导而难以分析。下面我们举两个例子。

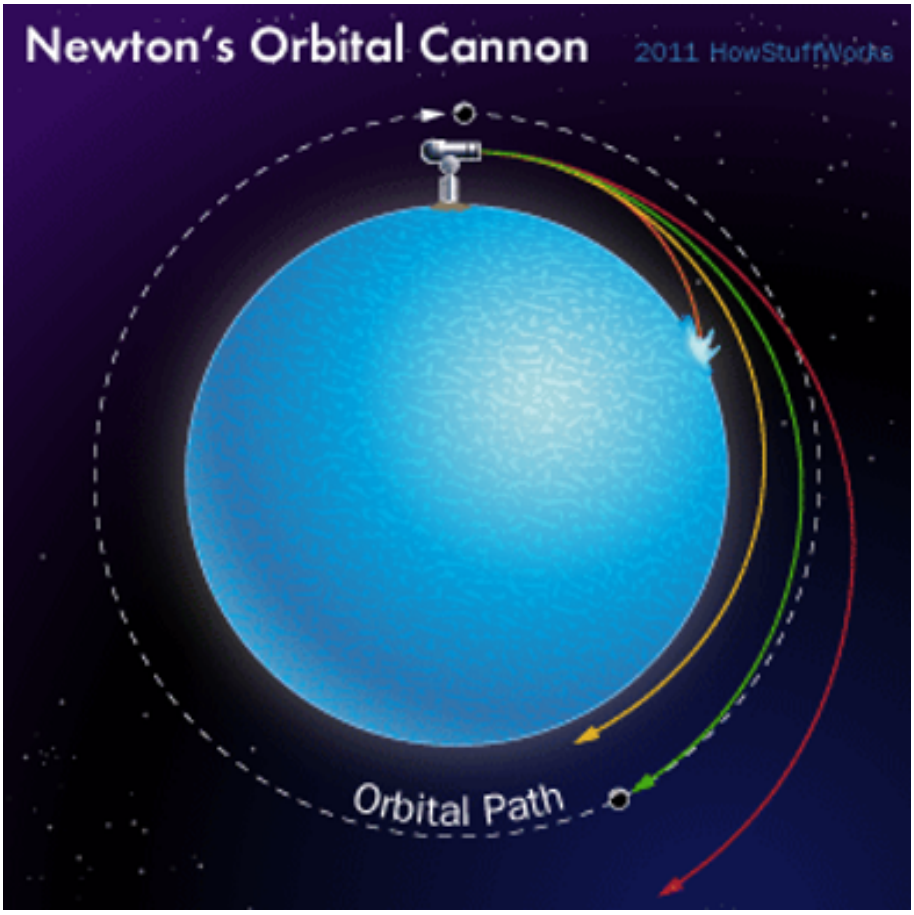


图 1.2

Arpit et al. (2017) 为了刻画神经网络的输出函数的复杂度随着训练进行是怎么变化的，他们定义了一个量叫临界样本数量。一个样本是临界的如果在它一个小的邻域内有一个和它类别不一样的样本。他们的实验发现临界样本数量随着训练步数增加会越来越多。基于此，他们推断神经网络的输出函数的复杂度会随着训练而增加。临界样本数量是一个很难分析的量。另一个研究Nakkiran et al. (2019) 用熵来衡量神经网络输出与其它函数的相关性来推断神经网络输出的复杂度。他们发现在刚开始训练的时候，神经网络的输出函数与线性函数的相关性很高，再经过一段训练后，该网络与浅层神经网络的充分训练结果相关性很高，更多训练后，该网络与更深网络的充分训练结果相关性很高。因此可以类似推荐神经网络的输出函数的复杂度会随着训练而增加。然而，这个研究用神经网络来刻画另一个神经网络，尽管在直观上确实能得到复杂度增加的结论，但在理论上，很难用黑箱的模型再分析另一个黑箱的模型。

正如很多科学研究的问题一样，比如计算数学 [cite E et al., CSIAM 2020]，或者比如研究苹果从树上掉到地上的机理来代替研究行星的运动，我们可以先从一些非常简单的问题中抓紧关键结论，再将这些结论推广到复杂的问题中。我们选择的这些问题应该是有一定的复杂性能够保存我们关心的现象，也应该足够的简单能够允许我们有比较深入的分析。对于神经网络的学习问题，本质上它是在做函数的拟合问题。尽管现实问题通常是高维的，但我们关心的问题在比较低维的问题中也经常存在。比如神经网络的泛化研究中受到广泛关注的泛化迷团 (generalization puzzle)，那就是为什么过参数化 (over-parameterized) 的神经网络（参数数目大于样本数目）在实际问题中能够取得比较好的泛化性能 [cite]。泛化迷团有趣的地方在于它表面上违背上传统学习理论的直观。在传统学习理论中，一个函数空间的复杂度通常和它的自由参数数目正相关。过参数化的神经网络所构成的函数空间直观上就有很大的复杂度。传统学习理论指出一个机器学习方法的泛化误差会随函数空间的复杂度增加而变大。因此，直观上，过参数化的神经网络拟合的泛化误差应该会很大，这与实际效果有明显的矛盾。泛化迷团不仅存在于实际问题，也存在于一维的拟合问题。传统的复杂度分析并没有考虑算法的训练过程，这使得其在考虑最差情况时很难结合算法的真实情况，而导致误差界过于宽松。

我们来看一个 MNIST 手写数据识别的例子。【详细说明这个例子：用不同的全连接网络，发现有一些简单样子总是在训练早期就分类好，有一些困难样本总是在训练晚期才分类好。】

对于这些简单或者困难的样本，要找到一个合适的量刻画或者发现其中的规律并不容易。但我们可以在一维函数中看看什么样本是简单或者困难的。比如图 1.3，用神经网络拟合 $\sin(x) + \sin(5x)$，在 2000 步的时候，那些恰好在 $\sin(x)$ 上的点已经简单拟合好了，而那些很振荡的部分还远未被拟合好。从这个简单的实验，我们可以推测，在一维实验里，简单的样本应该是那些在低频轮廓上的点，而复杂样本应该是那些在振荡剧烈位置的样本。因为我们可视化神经网络在拟合一维函数的训练过程，所以很容易发现，神经网络会先抓住训练集的轮廓，然后经过很长时间后，开始抓住更多细节。

在前一章中，对于一维问题，我们用一个多层神经网络来拟合一个仅有少数几个点的数据集 [引图]，我们发现尽管神经网络的参数远大于数据集的点数，神经网络学习到的函数是比较平坦的而非是振荡厉害的函数。通常平坦的函数不容易带来过拟合，而振荡的函数容易带来过拟合。因此，一维问题保存了泛化迷团这个现象，同时一维问题是非常容易可视化的，因为对于深入研究非常有利。图1.3中的例子也显示了一维问题保存了简单和困难样本的特点，我们也能对这些样本做简单的分析。

从一维实验，我们有两个线索，一是在平坦与振荡中，神经网络似乎偏好平坦；二是学习过程中，先学习轮廓再学习细节。平坦与振荡和轮廓与细节共同指向一个常用且相对容易分析的量，即频率。因此，我们将从频率的角度来研究神经网络的训练过程。

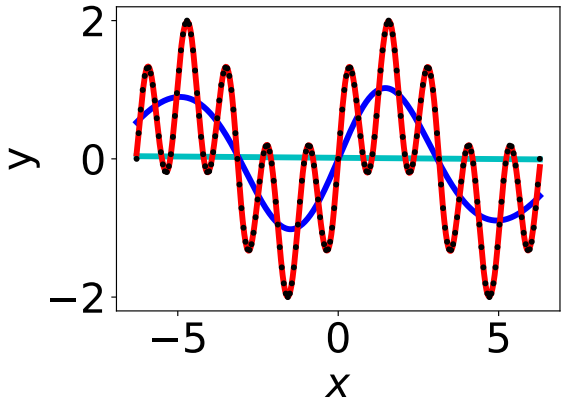


图 1.3: 深度神经网络（DNN）训练过程的示意图。黑色点表示从目标函数 $\sin(x)+\sin(5x)$ 中采样的训练数据。青色、蓝色和红色曲线分别表示在训练轮次 $t=0,2000,17000$ 时 DNN 的输出。

1.3 频率原则的一维实验

神经网络经常被用在分类问题中，因此，我们构造了一种看起来有点像分类问题的目标函数，如图1.4中的红色实线所示，它是一个周期的跳跃函数，它的输出只有三个值，类比三个类别。但为了简单，我们这里用的损失函数是均方误差函数，使用的神经网络是全连接的网络，按我们前面讨论的，把这个问题当成回归问题来做了。为了单纯研究训练过程，我们采了非常密的数据，避免由于数据稀疏带来其它问题。如图所示，可以发现，神经网络首先学习了这组数据的轮廓，然后再去学习了这组数据的细节，最后逼近这个真实函数。那么数学上刻画“轮

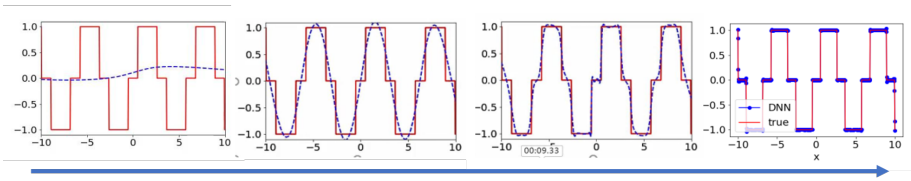


图 1.4: DNN 输出随训练步数的变化：从左到右训练步数增多，可以发现 DNN 首先捕捉了低频成分，得到了形如三角函数的输出，然后开始捕捉到目标函数的高频特性，逐渐学到目标函数中的“台阶”部分。

廓”与“细节”的方法，就是频率。这里简单介绍傅里叶变换（更多的介绍请见附录）：

连续傅里叶变换 (Continuous FT, CFT)：考虑函数 $g(x)$ 从实域 x 空间到频率 ξ 空间的变

换：

$$\mathcal{F}[g(x)](\xi)=\int_{-\infty}^{\infty}g(x)\mathrm{e}^{-2\pi\mathrm{i}\xi x}\mathrm{d}x,\quad \mathcal{F}^{-1}[\hat{g}(\xi)](x)=\int_{-\infty}^{\infty}\hat{g}(\xi)\mathrm{e}^{2\pi\mathrm{i}\xi x}\mathrm{d}\xi \tag{1.1}$$

它其实是一个内积的形式，把 $g(x)$ 投影到 $\mathrm{e}^{2\pi\mathrm{i}\xi}$ 的方向，也就是投影到了 sin 和 cos 的函数上，如果 $g(x)$ 是比较平坦的函数，那么就会在比较平坦的方向上保留的量比较大；反之如果 $g(x)$ 是高频函数，那么就会在高频的方向上有很多投影留下来的值。

傅里叶级数 (Fourier Series,FS): 由于我们使用有限区间采样的方法来获取数据，所以傅里叶变换就从连续退化成了傅里叶级数：

$$\mathcal{F}_{FS,T}[g(x)](k)=c_k=\frac{1}{T}\cdot\int_{-\frac{T}{2}}^{\frac{T}{2}}g(x)\mathrm{e}^{-2\pi itk/T}\mathrm{d}x\mathcal{F}_{FS,T}^{-1}\left[c_k\right](x)=\sum_{k=-\infty}^{\infty}c_k\mathrm{e}^{2\pi ixk/T} \tag{1.2}$$

其中 $\left(-\frac{T}{2}\leq x\leq\frac{T}{2},\xi=\frac{k}{T},k\in\mathbb{Z}\right)$，$\{c_k\}_{k\in\mathbb{Z}}$ 被称作傅里叶系数，它在频域下只能取离散的成分，最平坦的数据对应到频域就有最小的频率。

离散傅里叶变换（DFT）：然而我们不仅周期是有限的，而且采样也是有精度的，所以最后就再退化为离散傅里叶变换的形式：

$$\mathcal{F}_{DFT}\left[\{a_j\}_{j=0}^{N-1}\right](k)=\sum_{j=0}^{N-1}a_j\mathrm{e}^{-2\pi\mathrm{i}kj/N}\mathcal{F}_{DFT}^{-1}\left[\{b_k\}_{k=0}^{N-1}\right](j)=\frac{1}{N}\sum_{k=0}^{N-1}b_k\mathrm{e}^{2\pi\mathrm{i}kj/N} \tag{1.3}$$

其中 $(j,k\in0,1,\ldots N-1)$。可以发现从连续傅里叶变换到离散傅里叶变换，我们的积分形式变成了离散形式，而且在时域和频域上都只有有限个点，所以我们可以通过快速傅里叶变换来监测在频域上的拟合过程，如图1.5所示 从这个实验中可以发现：DNN 在拟合函数的时候，首

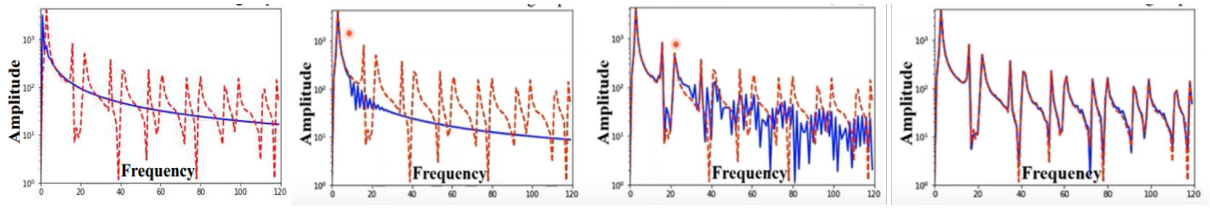


图 1.5: DNN 输出的 FFT 随时间的变化过程，红色的虚线为目标函数的 FFT，蓝色的实现为这个时刻 DNN 拟合结果的 FFT，从左到右训练步数逐渐增加。从频域上的结果可以更直观地看出 DNN 首先学习了数据的低频成分，然后随着训练步数的增多，学到了数据的高频成分。

先学习到了低频的成分，而高频的成分学到的很少；然后逐渐得学到高频的成分，我们将这个现象总结为一个频率原则。当目标函数中所有高频成分都被学习完了以后，误差就变为 0 了，也就是没有动力继续训练了，也就是说 DNN 的拟合结果的频率不会带来超过数据集中最高的频率。

细心的读者会发现这个实验存在缺陷，不足以说明频率原则。从图中可以看出，随着频率的增加，相应的幅值也下降了，所以到底是因为低频使神经网络的学习变快还是因为大幅值才让它变快了呢？下面这个实验控制不同频率成分的幅值相同，并进行了实验。目标函数选为 $f(x)=\sin(x)+\sin(3x)+\sin(5x)$，结果如图1.6所示，左边是目标函数和它的傅里叶变换，右边是不同频率成分下 DNN 拟合结果的相对误差 $\Delta F(k)=\left|\hat{h}_k-\hat{f}_k\right|/\left|\hat{f}_k\right|$ 随训练步数的演化。可以看出，在幅值相同的情况下，依旧是存在低频先收敛的现象。幅值在收敛中起到什么作用，

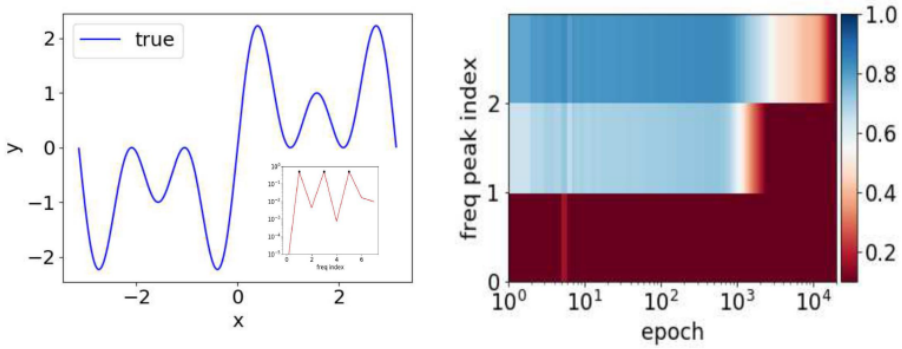


图 1.6: 控制目标函数幅值相同的一维数据实验，热力图中红色代表 DNN 拟合结果的相对误差很小，蓝色代表相对误差较大。可以发现目标函数有三个主要的频率成分且对应的幅值都相等。在训练的过程中频率最低的成分收敛地最快，没几个 epoch 就已经拟合地很好，而高频成分却用了 10^4 个 epoch 才达到了较小的相对误差

我们将在理论的部分重新回来看。

基于一维实验的启发，我们可以设计更多针对不同的网络结构、不同的激活函数、训练方法、参数初始化方法、真实数据等等的实验。大量的实验验证频率原则是广泛存在的。下面介绍二维的实验和高维的实验。

1.4 二维图片实验：回归问题

二维函数的拟合可以用神经网络记忆一张二维图片为例。一张图片，以黑白照片为例，实际上是一个二维的函数，自变量是位置坐标，应变量就是它的灰度值，它是一个从二维位置坐标到一维灰度值输出的函数。假设一个神经网络完全记住了这张图片，我们可以将位置坐标逐个输入到神经网络中，然后得到每个位置坐标对应的灰度值，最后把它们拼在一起就可以得到这张图片。

若是彩色图片，它有三个通道，分别是红色、绿色和蓝色（RGB），我们可以用一个具有两维输入和三维输出的神经网络来拟合它。

图1.7展示了用一个全连接网络拟合一张图片的训练过程。可以发现神经网络拟合这个二维函数的时候，当训练步数在 4000 步的时候，对眉毛、面部、头发等的低频成分已经有了一定的捕捉；在 8000 步的时候，对面部这种低频的信息还原已经比较清晰；直到 36000 步的时候 DNN 才捕捉到了而头发中的高频成分，对整个面部有了一个比较好的还原。在这个训练过程的展示中，我们发现，神经网络并不是先记住某一部分信息再去记另一部分信息，而是先学习到了一个模糊的图像，从频率的角度，这就低频的成分。神经网络的学习并非是简单的记忆过程，而是一种特征提取的过程。

进一步，我们也可以用神经网络来重构三维物体。比如给一堆点云数据，也就是一个物体的三维坐标和强度值，但神经网络重构的缺点在于它很难重构细节部分。如果克服这个缺点是神经网络算法设计的一个重要方向，我们将在多尺度神经网络这一章节中深入。

这个训练过程与我们人类在认识一个陌生的事物是类似的，一般是先记住大致的轮廓信息，然后再是细节信息。

图 1.7: DNN 拟合图像的过程，当训练步数在 4000 步的时候，对眉毛、面部、头发等的低频成分已经有了一定的捕捉；在 8000 步的时候，对面部这种低频的信息还原已经比较清晰；直到 36000 步的时候 DNN 才捕捉到了而头发中的高频成分，对整个面部有了一个比较好的还原。

1.5 在图像分类问题中验证频率原则

图像识别是非常高维的问题，比如在 CIFAR10 的分类中，图片的大小是 $32 \times 32 \times 32 \times 3$, 最后的 “3” 是 RGB 三个通道。将一张图片输入到神经网络中，我们需要输入 3072 个数，输出是 10 个类别中每个类别的概率，也就是在 CIFAR10 的分类问题中，神经网络需要学习一

个从 3072 维的输入空间映射到 10 维空间的高维函数。在如此高维的问题中，频率的概念和傅里叶变换都是极难想象的。我们首先区分两种常见的频率概念，然后介绍高维傅里叶变换的几种替代方式。

频率的基本理解是输入变化对输出的影响。低频指的是当输入发生变化时，输出的变化量很小。高频指的是当输入有较小的变化时，输出会有较大的变化。对于一个函数，不同位置的主要频率成分可以很不一样。

1.5.1 图像中的频率

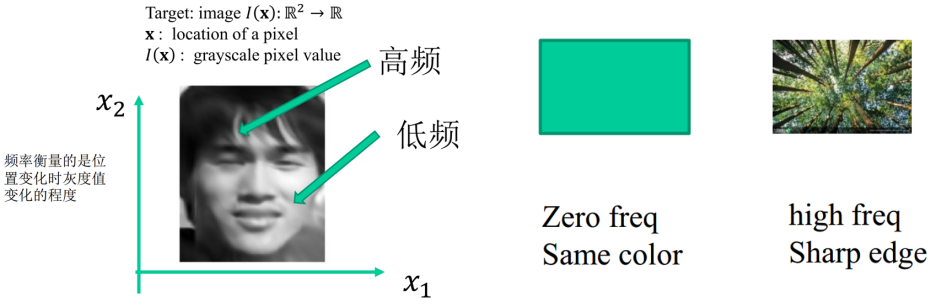


图 1.8: Image Frequency 定义图像拟合中的高频与低频成分：当某个像素点与周围像素点灰度 (RGB) 值接近的时候，认为是低频。比如对于纯色块的图片，它的频率就是 0；当像素点的灰度值 (RGB 值) 变化剧烈的时候，认为是图像中的高频成分，比如图中色彩非常鲜艳的照片就包含许多高频成分。

在图像拟合的问题中，我们考虑的是图片中的频率。比如图1.8这张照片中，头发边缘这种变化非常快的地方是高频成分主导的部分；而对于面部这种灰度值变化不剧烈的地方，它就是低频成为主导的。更为极端的，如果一张图片只有一种纯色，那么它就是一个只含有零频的常值函数。

1.5.2 分类问题中的响应频率

高维函数的频率是相对难想象的。我们考虑的是从输入到输出的响应频率，而不是输入空间内部的频率。我们可以用基本的频率的概念来理解。对高维函数中的某一点，沿不同方向，它主导的频率成分差别可以很大。

在图像分类识别中，对抗样本是很著名的例子。比如一张熊猫的图片，图1.9，它是高维空间中的一点，在这点沿某个方向做一点扰动后，可以得到一张与原始图片在肉眼上看几乎一模

一样的图片，但此时，神经网络输出的标签已经从熊猫变成了长臂猿，也就是微小的输入变化引起了巨大的输出变化，这个方向上有高强度的高频成分。

要注意，分类问题中，我们这里并不是考虑图片中的频率。我想再强调这一点，因为很多人在考虑分类问题时，仍然局限于图片本向的频率，很难想象分类函数的频率。另一点需要注意的是，考虑图片中的低频和高频成分对分类的准确度和稳定性的影响也是一个有趣的研究方向。我们这里在分类问题中不考虑图片频率，并不代表它对分类问题没有影响。

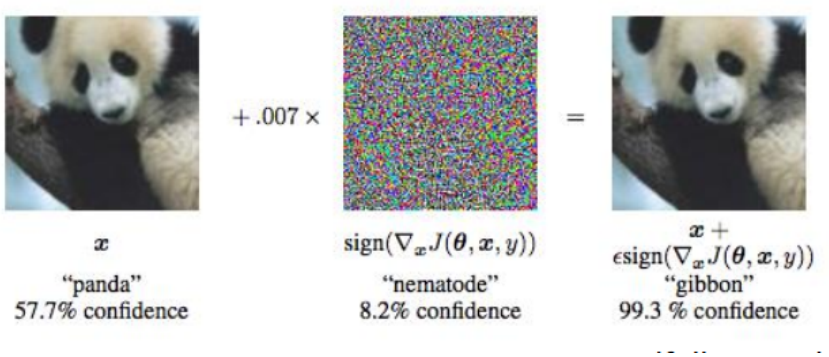


图 1.9: 某张图片的分类结果为 57.7% 的可能是熊猫，但是对图像加上一些随机噪声，加上噪声之后的结果在 Image frequency 的范畴内没什么变化，这个噪声可以被视作低频信息，但是却使得网络的输出变为 99.3% 的可能为长臂猿。所以在响应频率的范畴下，这个噪声是一个高频信息。

1.5.3 高维傅里叶变换的困难

对于一个高维数据集的傅里叶变换，可以看成对连续傅里叶变换的离散蒙特卡洛离散形式，比如对训练集 $\{(\mathbf{x}_i, y_i)\}_{i=1}^n : x_i \in R^d, y_i \in R^{d_o}$，我们用它的 NUDFT(Nonuniform Discrete Fourier transform) 来定义图片的响应频率：

$$\hat{y}_{\mathbf{k}} = \frac{1}{n} \sum_{i=1}^{n} y_i \mathrm{e}^{-\mathrm{i}2\pi \mathbf{k} \cdot \mathbf{x}_i}, \tag{1.4}$$

其中 $\mathbf{k}$ 就是我们定义的图片的响应频率，它维度与图片输入维度相等。比如对于 MNIST 数据集，输入 x 是一个 784 维的灰度值，那么这个 k 相应地也是 784 维。注意到公式中的求和是对所有样本，也就是这里的频率是考虑了所有样本和所有输出的整体频率。

高维的傅里叶变换存在维数灾难的问题。假如对于 MNIST 数据集，输入 x 是一个 784 维的向量，我在每个维度上观察两个频率 $\mathbf{k}$，那么总共就要观察 2^{784} 个数据点，约为 10^{200} 个数

据点，这对于现阶段的计算能力来说，是不可能实验验证的。因此，我们需要做一些折中来研究一个高维函数在频域空间的特征。

1.5.4 投影法

投影法是只计算函数在高维频率空间中一个方向的数值。首先要选定一个方向 $\mathbf{p_1}$，在这个方向的频率可以写成 $\mathbf{k} = k\mathbf{p}_1$，将其代入 NUDFT 中，

$$\hat{y}_{\mathbf{k}} = \frac{1}{n}\sum_{i=1}^{n} y_i \mathrm{e}^{-\mathrm{i}2\pi k(\mathbf{p}_1 \cdot \mathbf{x}_i)}. \tag{1.5}$$

原则上, 我们可以随意选取 $\mathbf{p_1}$ 的方向。下面的实验中，我们选取了主成分方向进行研究。若将数据在这个方向上的投影记为 v_i，即 $v_i = \mathbf{p}_1 \cdot \mathbf{x}_i \in \boldsymbol{R}$，那么 NUDFT 进一步化简为

$$\hat{y}_{\mathbf{k}} = \frac{1}{n}\sum_{i=1}^{n} y_i \mathrm{e}^{-\mathrm{i}2\pi k v_i}. \tag{1.6}$$

上面的表达式实际上变成了一维的傅里叶变换。下面来看投影方法的实验结果：图1.10(a,c) 分

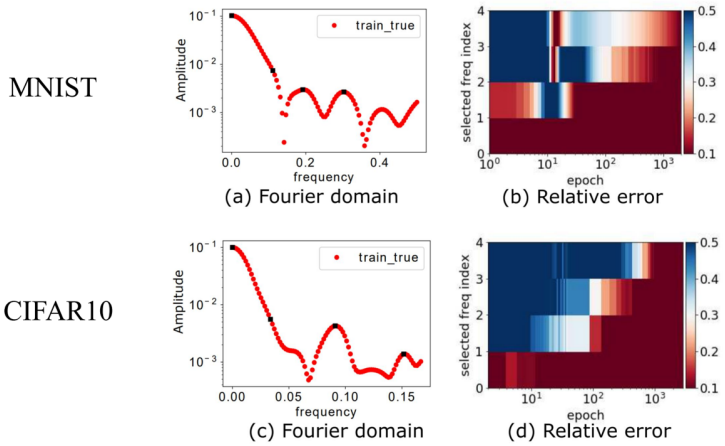


图 1.10: MNIST 与 CIFAR10 在主成分方向的傅里叶变换及其主导频率下训练的收敛速度。

别为 MNIST 数据集和 CIFAR10 数据集的主成分方向的傅里叶变换结果，可以发现对于真实数据中往往也是低频占优的，也就是低频的幅值往往更大一些。选取其中几个峰值（为什么选择峰值？附录里我们会简单说明）对应的频率进行考察，结果如图1.10(b,d) 所示。颜色越偏向深红，表示在某个频率上的相对误差 $\Delta_F(k) = \left|\hat{h}_k - \hat{y}_k\right| / |\hat{y}_k|$ 越小。从单个方向的角度可以看出，在 MNIST 和 CIFAR10 上，的确有低频比高频先收敛的现象。

1.5.5 滤波法

虽然从单个方向的角度的确看到了低频先收敛的现象，但是我们有时候并不知道我们想研究的信号处于哪个方向。滤波的方法把频率空间分成低频和高频两部分 (图1.11(a))，然后检验低频部分和高频部分的收敛。要获得低频部分的方法，理论上我们可以把要检验的函数乘以一个示性函数 (图1.11(b))，这个示性函数在频率强度小于 k_0 的范围内为 1，其余为 0，

$$\mathbf{1}_{|\boldsymbol{k}|\leq k_0} = \begin{cases} 1, & |\boldsymbol{k}| \leq k_0 \\ 0, & |\boldsymbol{k}| > k_0 \end{cases} \tag{1.7}$$

这个示性函数将整个傅里叶空间截断为一个球来进行研究。但是这个示性函数是频域空间上的函数，需要先把考虑的函数先变换到频域空间，显然不合适。这时候，我们想到卷积定理，考虑频域空间的两个函数，$\hat{g}_1$ 和 $\hat{g}_2$，它们在频域空间的乘积做傅里叶逆变换，等价于这两个函数的逆变换在实域空间做卷积，

$$F^{-1}\left[\hat{g}_1 \cdot \hat{g}_2\right](x) = F^{-1}\left(\hat{g}_1\right) * F^{-1}\left(\hat{g}_2\right). \tag{1.8}$$

即使用了卷积定理，我们仍然需要计算一次高维示性函数的傅里叶逆变换。为了克服这个困难，我们采用高斯函数 (图1.11(c)) 来近似地作为示性函数，因为 1) 它指数衰减，可以作为理想的示性函数的近似, 2) 它的傅里叶逆变换仍然是高斯函数。举个例子，考虑一维高斯函数 $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$，对它做傅里叶变换：

$$F(\omega) = \int_{-\infty}^{+\infty} e^{-ax^2} e^{-j\omega x} dx = \int_{-\infty}^{+\infty} e^{-ax^2 - j\omega x} dx = e^{-\frac{\omega^2}{4a}} \int_{-\infty}^{+\infty} e^{-\left(\sqrt{a}x + \frac{j\omega}{2\sqrt{a}}\right)^2} dx = \sqrt{\frac{\pi}{a}} e^{-\frac{\omega^2}{4a}} \tag{1.9}$$

也即是说高斯函数在傅里叶变换前后保持了 $f(x) = e^{-ax^2}$ 的形式，高斯函数的傅里叶（逆）变换还是一个高斯函数。在实域上方差为 δ 的高斯函数，在频域上其标准差为 $1/\delta$，这个标准差也就是近似为示性函数的宽度。那么在频域上的截断就转化为在时域上的函数和一个高斯函数的卷积，就把函数的高频成分和低频成分近似地分开了。低频成分可以近似成

$$\boldsymbol{y}_i^{\text{low},\delta} = \frac{1}{C_i} \sum_{j=0}^{n-1} \boldsymbol{y}_j G^{\delta}(\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{1.10}$$

其中 $C_i = \sum_{j=0}^{n-1} G^{\delta}(\boldsymbol{x}_i - \boldsymbol{x}_j)$ 是一个归一化因子，

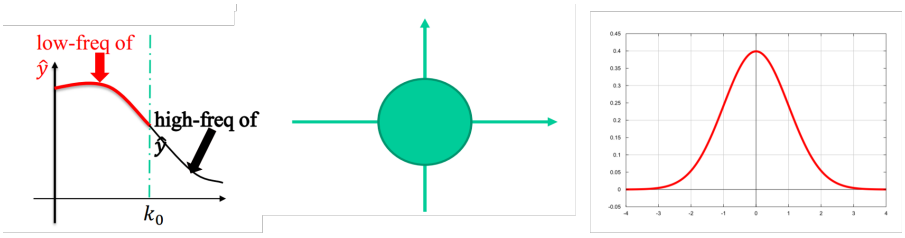
$$G^{\delta}(\boldsymbol{x}_i - \boldsymbol{x}_j) = \exp\left(-|\boldsymbol{x}_i - \boldsymbol{x}_j|^2/(2\delta)\right). \tag{1.11}$$

高频部分可以近似为 $\boldsymbol{y}_i^{\text{high},\delta} \triangleq \boldsymbol{y}_i - \boldsymbol{y}_i^{\text{low},\delta}$. 对于 DNN 每个输出 $\boldsymbol{h}_i$，我们也可以类似地做，$\boldsymbol{h}_i^{\text{low},\delta}$ and $\boldsymbol{h}_i^{\text{high},\delta}$。然后，我们可计算低频和高频的相对误差

$$e_{\text{low}} = \left(\frac{\sum_i |\boldsymbol{y}_i^{\text{low},\delta} - \boldsymbol{h}_i^{\text{low},\delta}|^2}{\sum_i |\boldsymbol{y}_i^{\text{low},\delta}|^2} \right)^{\frac{1}{2}}, \tag{1.12}$$

$$e_{\text{high}} = \left(\frac{\sum_i |\boldsymbol{y}_i^{\text{high},\delta} - \boldsymbol{h}_i^{\text{high},\delta}|^2}{\sum_i |\boldsymbol{y}_i^{\text{high},\delta}|^2} \right)^{\frac{1}{2}}, \tag{1.13}$$

这里 h 和 y 分别代表神经网络输出和真实标签。



(a) 频域空间的高低频成分　　(b) 示性函数的在频域空间截断　　(c) 高斯函数

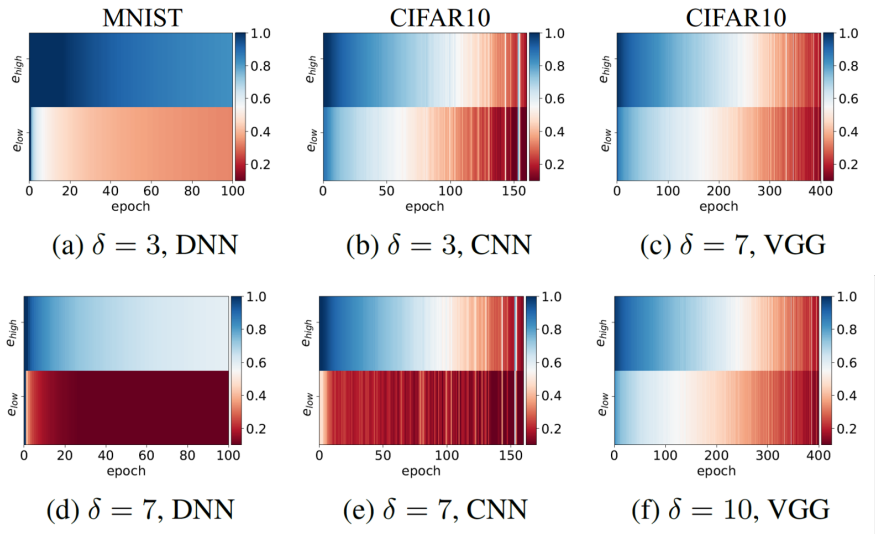
图 1.11: (a) 表示了一个函数在频率空间中的高频和低频成分，(b) 表示滤波研究的基本方法是通过示性函数将频率空间截断为一个球和剩余部分进行研究，(c) 是一个一维高斯函数的可视化，它指数衰减、傅里叶逆变换很容易做的性质，使得它成为了理想示性函数的近似。

我们对 MNIST 和 CIFAT10 使用全连接网络 DNN、卷积网络 CNN、一个常用的中等规模网络 VGG16 Simonyan & Zisserman (2014) 三种不同的网络用交叉熵进行训练，其实验结果如下：从图1.12中可以看出，通过对图片的横向比较，可以发现对于 DNN,CNN,VGG 三种网络，都呈现出了低频成分先收敛的现象。在图中纵向进行比较，可以发现当高斯函数的标准差变化时，也呈现出了一致的结论。

所以以上的多种实验都验证了频率原则是广泛存在的一种现象，它和神经网络的结构、使用的激活函数、使用的损失函数等等都是无关的。所以实验上可以说明，神经网络都是偏好低频信息的。下面就用更严格的数学表示，来描述这一现象，探究它产生的原因，并对神经网络的泛化性提出新的理解。

1.6 在非梯度下降的算法中验证频率原则

在Ma et al. (2021) 中，许多用非梯度下降的算法也能看到明显的频率原则，包括拟牛顿法，以及一些完全没有用梯度信息的算法，比如 Powell’ s method 和 Particle Swarm Optimization。这些实验验证了频率原则的广泛性。



(a) $\delta = 3$, DNN (b) $\delta = 3$, CNN (c) $\delta = 7$, VGG

(d) $\delta = 7$, DNN (e) $\delta = 7$, CNN (f) $\delta = 10$, VGG

图 1.12: 滤波方法的实验结果，其中 δ 表征高斯函数的宽度，也就是研究的低频到底有多低，高频到底有多高。σ 越大表示高斯函数在频率空间的带宽越窄，低频成分的范围越大。【介绍更多的实验细节】

参考文献

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y. et al. (2017), 'A closer look at memorization in deep networks', *arXiv preprint arXiv:1706.05394* .

Ma, Y., Xu, Z.-Q. J. & Zhang, J. (2021), 'Frequency principle in deep learning beyond gradient-descent-based training', *arXiv preprint arXiv:2101.00747* .

Nakkiran, P., Kaplun, G., Kalimeris, D., Yang, T., Edelman, B. L., Zhang, F. & Barak, B. (2019), 'Sgd on neural networks learns functions of increasing complexity', *arXiv preprint arXiv:1905.11604* .

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* .

Xu, Z.-Q. J., Zhang, Y., Luo, T., Xiao, Y. & Ma, Z. (2020), 'Frequency principle: Fourier analysis sheds light on deep neural networks', *Communications in Computational Physics* **28**(5), 1746–1767.

Xu, Z.-Q. J., Zhang, Y. & Xiao, Y. (2019), 'Training behavior of deep neural network in frequency domain', *International Conference on Neural Information Processing* pp. 264–274.