如何从频域的角度解释CNN（卷积神经网络）？

我觉得这个对我启发最大的是上海交大许志钦的工作。

https://ins.sjtu.edu.cn/people/xuzhiqin/fprinciple/index.html

他的B站演讲

数学学院本科课程：统计计算与机器学习3 Frequency Principle_哔哩哔哩 (゜-゜)つロ 干杯~-bilibili

另外，我大概线下听过他两次演讲，几乎都是关于神经网络与傅立叶变换、傅里叶分析方面的工作。

Training behavior of deep neural network in frequency domain

https://arxiv.org/pdf/1807.01251.pdf

这篇论文，开宗明义就是神经网络的泛化性能来源于它在训练过程，会更多关注低频分量。

CIFAR-10、MNIST的神经网络的拟合过程，感谢 @Jimmy指正，蓝色表示相对误差大，红色表示相对误差小，随着训练的epoch，频率越高（frequency index 大的），收敛越慢（即，对于某个epoch，低频的误差小，颜色偏红，高频部分误差大，颜色偏蓝色）。

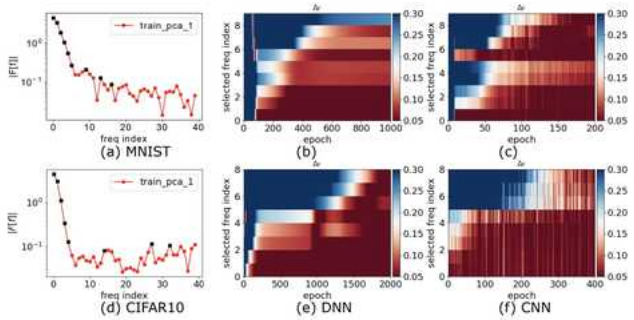


Fig. 1. Frequency analysis of DNN output function along the first principle component during the training. The training datasets for the first and the second row are from MNIST and CIFAR10, respectively. The neural networks for the second column and the third column are fully-connected DNN and CNN, respectively. (a,d) $|\mathcal{F}^n_{PC}[y](\gamma)|$. The selected frequencies are marked by black dots. (b, c, e, f) Δ_F at different recording epochs for different selected frequencies. Δ_F larger than 0.3 (or smaller than 0.05) is represented by blue (or red).

Theory of the frequency principle for general deep neural networks

https://arxiv.org/pdf/1906.09235v2.pdf

做了大量的数学推导证明F-Principle，分成训练的初始阶段、中间阶段、收尾阶段分别证明，对于非数学专业的人，有点繁琐。

Explicitizing an Implicit Bias of the Frequency Principle in Two-layer Neural Networks

https://arxiv.org/pdf/1905.10264.pdf

为什么参数比样本多的深层神经网络（DNNs）通常能很好地泛化，这仍然是个谜。理解这一难题的一个尝试是发现DNNs训练过程中的隐含偏差，例如频率原理（F-Principle），即DNNs通常从低频到高频拟合目标函数。受F-Principle的启发，该论文提出了一个有效的线性F-Principle动力学模型，该模型能准确预测大宽度的两层ReLU神经网络（NNs）的学习结果。这种Linear FP动力学被NNs的线性化Mean Field剩余动力学合理化。重要的是，这种LFP动力学的长时间极限解等价于显式最小化FP范数的约束优化问题的解，其中可行解的高频率受到更严重的惩罚。利用该优化公式，给出了泛化误差界的先验估计，表明目标函数的FP范数越高，泛化误差越大。总的来说，通过将F-Principle的隐式偏差解释为两层NNs的显式惩罚，这个工作朝着定量理解一般DNNs的学习和泛化迈出了一步。

这个是图像类的二维数据的LFP模型示意图。

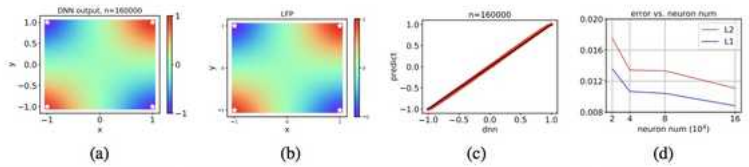


Figure 2: LFP model for 2-d training data of the XOR problem. (a) The final output of the NN. (b) The solution of the corresponding LFP model. The training data are marked by white stars. (c) Each dot represents the final output of NN (abscissa) vs. solution of the LFP model (ordinate) evaluated at one of the 1600 evenly spaced test points. The black line indicates the identity function. (d) Decay of $L^1(h_N, h_{\mathrm{LFP}})$ and $L^2(h_N, h_{\mathrm{LFP}})$ (mean of 10 trials) vs. neuron number.

许教授之前的介绍：

> LFP 模型为神经网络的定量理解提供了全新的思路。首先，LFP 模型用一个简单的微分方程有效地刻画了神经网络这样一个参数极多的系统其训练过程的关键特征，并且能够精确地预测神经网络的学习结果。因此该模型从一个新的角度建立了微分方程和神经网络的关系。由于微分方程是一个非常成熟的研究领域，我们相信该领域的工具可以帮助我们进一步分析神经网络的训练行为。

> 其次，与统计物理类似，LFP 模型只与网络参数的一些宏观统计量有关，而与单个参数的具体行为无关。这种统计刻画可以帮助我们准确理解在参数极多的情况下 DNN 的学习过程，从而解释 DNN 在参数远多于训练样本数时较好的泛化能力。
>
> 在该工作中，我们通过一个等价的优化问题来分析该 LFP 动力学的演化结果，并且给出了网络泛化误差的一个先验估计。我们发现网络的泛化误差能够被目标函数f本身的一种 F-principle 范数（定义为 $||\gamma(\xi)^{-1}\hat{f}(\xi)||_{L^2}$，γ(ξ) 是一个随频率衰减的权重函数）所控制。
>
> 值得注意的是， 我们的误差估计针对神经网络本身的学习过程，并不需要在损失函数中添加额外的正则项。关于该误差估计我们将在之后的介绍文章中作进一步说明。

FREQUENCY PRINCIPLE: FOURIER ANALYSIS SHEDS LIGHT ON DEEP NEURAL NETWORKS

https://arxiv.org/pdf/1901.06523.pdf

Theorem 1. *Considering a DNN of one hidden layer with activation function $\sigma(x) = \tanh(x)$, for any frequencies k_1 and k_2 such that $|\hat{f}(k_1)| > 0$, $|\hat{f}(k_2)| > 0$, and $|k_2| > |k_1| > 0$, there exist positive constants c and C such that for sufficiently small δ, we have*

$$\frac{\mu\left(\left\{W : \left|\frac{\partial L(k_1)}{\partial \theta_{ij}}\right| > \left|\frac{\partial L(k_2)}{\partial \theta_{ij}}\right| \quad \text{for all} \quad l, j\right\} \cap B_\delta\right)}{\mu(B_\delta)} \geq 1 - C\exp(-c/\delta),$$

where $B_\delta \subset \mathbb{R}^m$ is a ball with radius δ centered at the origin and $\mu(\cdot)$ is the Lebesgue measure.

这表明，对于任意两个非收敛频率，在较小的权重下，低频梯度指数性地优于高频梯度。根据Parseval定理，空间域中的MSE损失与Fourier域中的L2损失等效。为了更直观地理解低频损耗函数的高衰减率，我们考虑了在只有两个非零频率的损失函数的Fourier域中的训练。

Theorem 2. *Considering a DNN of one hidden layer with activation function $\sigma(x) = \tanh(x)$. Suppose the target function has only two non-zero frequencies k_1 and k_2, that is, $|\hat{f}(k_1)| > 0$, $|\hat{f}(k_2)| > 0$, $|k_2| > |k_1| > 0$, and $|\hat{f}(k)| = 0$ for $k \neq k_1, k_2$. Consider the loss function of $L = L(k_1) + L(k_2)$ with gradient descent training. Denote*

$$\mathcal{S} = \left\{\frac{\partial L(k_1)}{\partial t} \leq 0, \frac{\partial L(k_1)}{\partial t} \leq \frac{\partial L(k_2)}{\partial t}\right\},$$

that is, $L(k_1)$ decreases faster than $L(k_2)$. There exist positive constants c and C such that for sufficiently small δ, we have

$$\frac{\mu\left(\{W : \mathcal{S} \quad \text{holds}\} \cap B_\delta\right)}{\mu(B_\delta)} \geq 1 - C\exp(-c/\delta),$$

where $B_\delta \subset \mathbb{R}^m$ is a ball with radius δ centered at the origin and $\mu(\cdot)$ is the Lebesgue measure.

解释了ReLU函数为什么Work，因为tanh函数在空间域是光滑的，其导数在傅里叶区域随频率呈指数衰减。

许教授关于F-Principle的几篇科普文：

null

null

null

null

On the Spectral Bias of Deep Neural Networks

Bengio组的工作，之前写过一个比较粗糙的分析札记。

null

1. 利用连续分段线性结构对ReLU网络的傅里叶谱分量进行分析。
2. 发现了谱分量偏差（Spectrum bias）的经验证据，来源于低频分量，然而对低频分量的学习，有助于网络在对抗干扰过程中的鲁棒性。
3. 通过流形理论，给予学习理论框架分析。

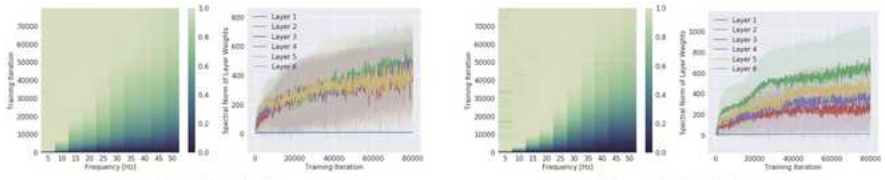
根据拓扑学的Storkes定理，证明ReLU函数紧凑、光滑，有助于训练的收敛，之后的Swish和Mish呢？（狗头）。

这样，在高维空间中，ReLU函数的谱衰减具有强烈的各向异性，ReLU傅立叶变换幅度的上限满足李普希茨约束。

实验：

中心点：低频分量学习优先级高

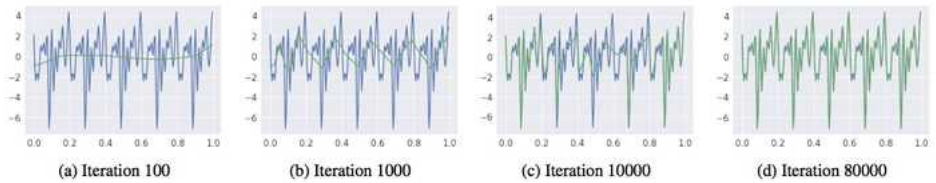
1. 对函数做实验： $\lambda(z)=\sum_i A_i \sin\left(2\pi k_i z+\varphi_i\right)$



(a) Equal Amplitudes (b) Increasing Amplitudes

Figure 1. Left (a, b): Evolution of the spectrum (x-axis for frequency) during training (y-axis). The colors show the measured amplitude of the network spectrum at the corresponding frequency, normalized by the target amplitude at the same frequency (i.e. $|\tilde{f}_{k_i}|/A_i$) and the colorbar is clipped between 0 and 1. Right (a, b): Evolution of the spectral norm (y-axis) of each layer during training (x-axis). Figure-set (a) shows the setting where all frequency components in the target function have the same amplitude, and (b) where higher frequencies have larger amplitudes. **Gist**: We find that even when higher frequencies have larger amplitudes, the model prioritizes learning lower frequencies first. We also find that the spectral norm of weights increases as the model fits higher frequency, which is what we expect from Theorem 1.

傅立叶变换效果



(a) Iteration 100 (b) Iteration 1000 (c) Iteration 10000 (d) Iteration 80000

Figure 2. The learnt function (green) overlayed on the target function (blue) as the training progresses. The target function is a superposition of sinusoids of frequencies $\kappa=(5,10,\ldots,45,50)$, equal amplitudes and randomly sampled phases.

迭代过程对函数的学习

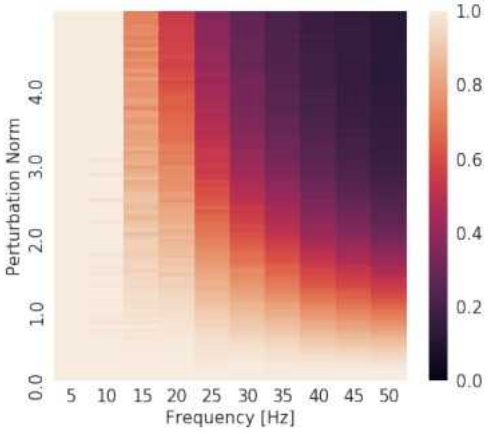
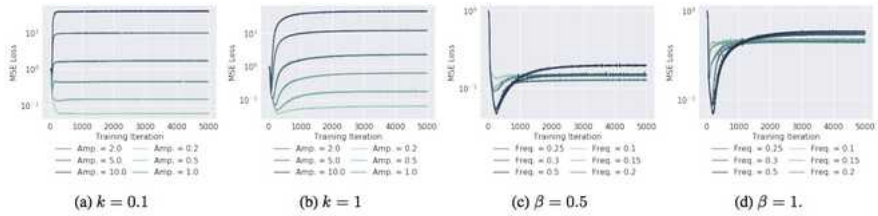


Figure 3. Normalized spectrum of the model (x-axis for frequency, colorbar for magnitude) with perturbed parameters as a function of parameter perturbation (y-axis). The colormap is clipped between 0 and 1. We observe that the lower frequencies are more robust to parameter perturbations than the higher frequencies.

模型的标准化谱分量

2. 带噪环境学习MNIST数据



(a) $k = 0.1$ (b) $k = 1$ (c) $\beta = 0.5$ (d) $\beta = 1.$

Figure 4. (a,b,c,d): Validation curves for various settings of noise amplitude β and frequency k. Corresponding training curves can be found in Figure 11 in appendix A.3. **Gist**: Low frequency noise affects the network more than their high-frequency counterparts. Further, for high-frequency noise, one finds that the validation loss dips early in the training. Both these observations are explained by the fact that network readily fit lower frequencies, but learn higher frequencies later in the training.

不同的验证损失

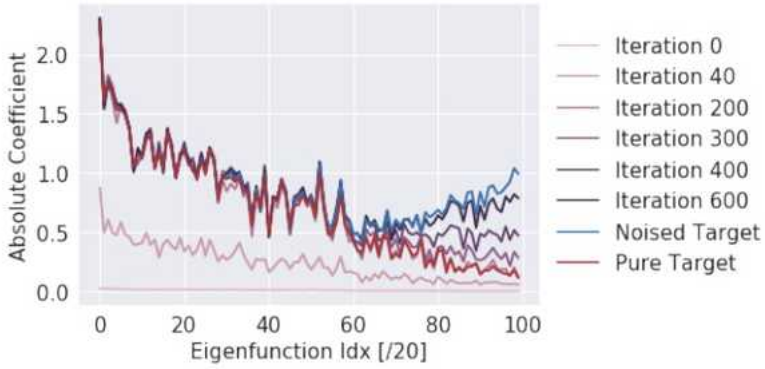


Figure 5. Spectrum of the network as it is trained on MNIST target with high-frequency noise (*Noised Target*). We see that the network fits the true target at around the 200th iteration, which is when the validation score dips (Figure 13 in appendix).

MNIST数据拟合的频率分量

神经网络可以近似任意值功能，但研究人员发现他们更喜欢低频的分量，也因此，它们表现出对平滑函数的偏倚——被称之为谱偏移（spectral bias）的现象。

流形假设

Manifold hypothesis. We consider the case where the data lies on a lower dimensional *data manifold* $\mathcal{M} \subset \mathbb{R}^d$ embedded in input space (Goodfellow et al., 2016), which we assume to be the image $\gamma([0,1]^m)$ of some injective mapping $\gamma : [0,1]^m \to \mathbb{R}^d$ defined on a lower dimensional latent space $[0,1]^m$. Under this hypothesis and in the context of the standard regression problem, a target function $\tau : \mathcal{M} \to \mathbb{R}$ defined on the data manifold can be identified with a function $\lambda = \tau \circ \gamma$ defined on the latent space. Regressing τ is therefore equivalent to finding $f : \mathbb{R}^d \to \mathbb{R}$ such that $f \circ \gamma$ matches λ. Further, assuming that the data probability distribution μ supported on $\mathcal{M}$ is induced by γ from the uniform distribution U in the latent space $[0,1]^m$, the mean square error can be expressed as:

$$\mathrm{MSE}_{\mu}^{(\mathbf{x})}[f,\tau] = \mathbb{E}_{\mathbf{x}\sim\mu}|f(\mathbf{x}) - \tau(\mathbf{x})|^2 = \mathbb{E}_{\mathbf{z}\sim U}|(f(\gamma(\mathbf{z})) - \lambda(\mathbf{z})|^2 = \mathrm{MSE}_{U}^{(\mathbf{z})}[f\circ\gamma,\lambda] \quad (12)$$

on $\mathbb{R}^d$ that yield the same function when restricted to the data manifold $\mathcal{M}$.

Our findings from the previous section suggest that neural networks are biased towards expressing a particular subset of such solutions, namely those that are low frequency. It is also worth noting that there exist methods that restrict the space of solutions: notably adversarial training (Goodfellow et al., 2014) and Mixup (Zhang et al., 2017b).

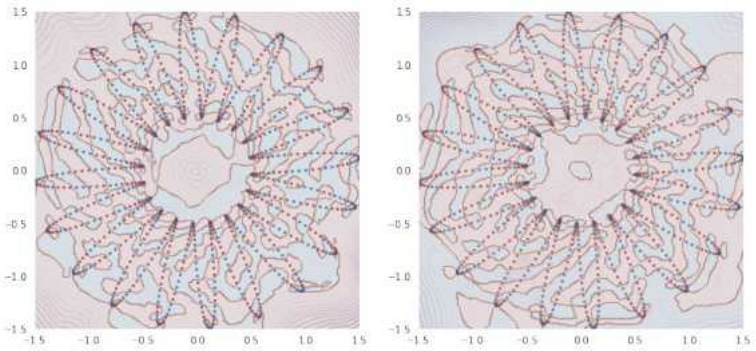


Figure 7. Functions learned by two identical networks (up to initialization) to classify the binarized value of a sine wave of frequency $k = 200$ defined on a $\gamma_{L=20}$ manifold. Both yield close to perfect accuracy for the samples defined on the manifold (scatter plot), yet they differ significantly elsewhere. The shaded regions show the predicted class (Red or Blue) whereas contours show the confidence (absolute value of logits).

流形越复杂，然后学习过程越容易，这个假设会Break“结构风险最小化”假设，有可能会出现“过拟合”。

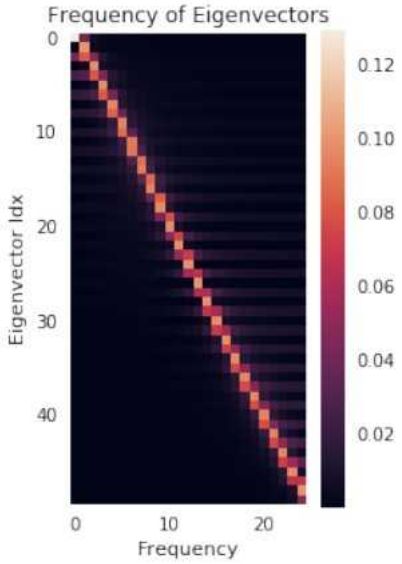


Figure 6. Spectrum (x-axis for frequency, colorbar for magnitude) of the n-th (y-axis) eigenvector of the Gaussian RBF kernel matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, where the sample set is $\{x_i \in [0, 1]\}_{i=1}^{50}$ is $N = 50$ uniformly spaced points between 0 and 1 and k is the Gaussian RBF kernel function. **Gist:** The eigenfunctions with increasing n roughly correspond to sinusoids of increasing frequency. Refer to Appendix A.4 for more details.

如果有复杂的数据集（ImageNet），搜索空间比较大，也要通过一定的方法，使其“work in harmony”，调谐地工作。

感觉Bengio认为其对深度学习的正则化有启发意义。

Machine Learning from a Continuous Viewpoint

https://arxiv.org/pdf/1912.12777.pdf

数学家Wienan.E的争鸣，频率原则并不总是Work的。

假设某个函数：

$$f(x)=\int_0^{2\pi}\varphi(x,w)\rho(dw), \tag{148}$$

with the feature $\varphi(\boldsymbol{x},w)$ given by

$$\varphi(x,w)=\sum_{k=-\infty}^{\infty}e^{-\frac{(x-w-2k\pi)^2}{h^2}}, \tag{149}$$

$\rho(dw)$概率测度

基于核函数对其求导：

$$\begin{aligned}\frac{d}{dt}f_t(x)&=\int\varphi(x,w)\partial_t\rho_t(w)dw\\&=\int\varphi(x,w)\nabla\cdot\left(\rho_t(w)\int\nabla K\left(w,w'\right)\left(\rho_t\left(w'\right)-\rho^*\left(w'\right)\right)dw'\right)dw\\&=\int\varphi(x,w)\nabla\cdot\left(\rho_t(w)\int\nabla\varphi\left(x',w\right)\left(f_t\left(x'\right)-f^*\left(x'\right)\right)dx'\right)dw\\&=-\int\left(\int\nabla\varphi(x,w)\nabla\varphi\left(x',w\right)\rho_t(w)dw\right)\left(f_t\left(x'\right)-f^*\left(x'\right)\right)dx'\end{aligned}$$

其中：

$$\frac{d}{dt}f_t(x)=-\int\tilde{K}\left(x,x'\right)\left(f_t\left(x'\right)-f^*\left(x'\right)\right)dx'$$

$$\tilde{K}\left(x,x'\right)=\int\nabla_w\varphi(x,w)\nabla_w\varphi\left(x',w\right)\rho_0(w)dw$$

$$\begin{aligned}\tilde{K}\left(x,x'\right)&=\partial_x\partial_{x'}\int\varphi(x,w)\varphi\left(x',w\right)\rho_0(w)dw\\&=\partial_x\partial_{x'}K\left(x,x'\right)\\&=\frac{h}{\sqrt{8\pi}}\sum_{k=-\infty}^{\infty}\left[\frac{1}{h^2}-\frac{\left(x-x'+2k\pi\right)^2}{h^4}\right]e^{-\frac{\left(x-x'+2k\pi\right)^2}{2h^2}}\end{aligned}$$

进行傅立叶系数的分解：

$$\begin{aligned}\tilde{K}\left(x, x'\right) =& c_0+\sum_{k=1}^{\infty} b_k \sin\left(k\left(x-x'\right)\right)+c_k \cos\left(k\left(x-x'\right)\right) \\ =& c_0+\sum_{k=1}^{\infty} b_k\left(\sin(kx) \cos\left(kx'\right)-\cos(kx) \sin\left(kx'\right)\right) \\ &+\sum_{k=1}^{\infty} c_k\left(\cos(kx) \cos\left(kx'\right)+\sin(kx) \sin\left(kx'\right)\right)\end{aligned}$$

推导得到：

$$\int \tilde{K}\left(x, x'\right)\left(u \sin\left(kx'\right)+v \cos\left(kx'\right)\right) dx'=\pi\left(\left(c_k u+b_k v\right) \sin(kx)+\left(c_k v-b_k u\right.\right.$$

特征函数：

$$\left\{u \sin(kx)+v \cos(kx) : (u, v)^T \text{ is the eigenvector of } \begin{bmatrix} c_k & b_k \\ -b_k & c_k \end{bmatrix}\right\}$$

然后给出了频域原则work的边界。

The eigenvalues are $\pi\lambda_k$, where $\lambda_k = c_k + ib_k, c_k - ib_k$. Using (156), we can explicitly compute the Fourier coefficients of $\tilde{K}$ and obtain $c_0 = 0$, $b_k = 0$, and $c_k = \frac{h^2k^2}{2}e^{-h^2k^2/2}$. Hence, the eigenvalues of the operator $\tilde{K}$ are $\{\frac{\pi h^2k^2}{2}e^{-h^2k^2/2}\}$, and the eigenfunctions are simply the Fourier basis functions. We see that the eigenvalues decrease with the frequency k when $k \geq 2/h$. This implies that the frequency principle should hold for $k \geq 2/h$.

work的情况：

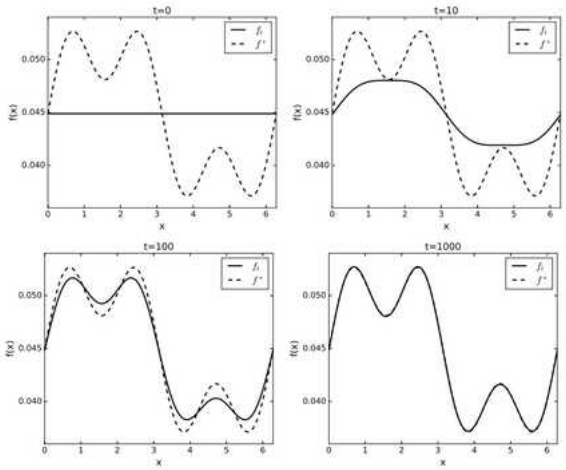


Figure 5: The example that demonstrates the frequency principle. The four plots correspond to the function f_t at $t = 0,\ 10,\ 100,\ 1000$, compared to the target function.

不work的情况：

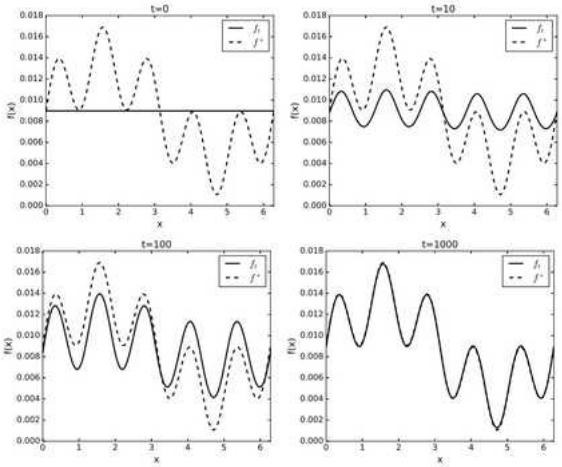


Figure 6: The example that demonstrates when the frequency principle does not hold. The four plots correspond to the function f_t at $t = 0,\ 10,\ 100,\ 1000$, compared to the target function.

如果说Wienan. E是从数学家的角度给出了Frequency Principle的边界的话，那么做工程的小伙伴一定要看看这篇论文

A Fourier Perspective on Model Robustness in Computer Vision

代码也已经开源了：

https://arxiv.org/pdf/1906.08988.pdf

https://github.com/google-research/google-research/tree/master/frequency_analysis

作者的意思是关注鲁棒性，不能完全丢掉高频特征。

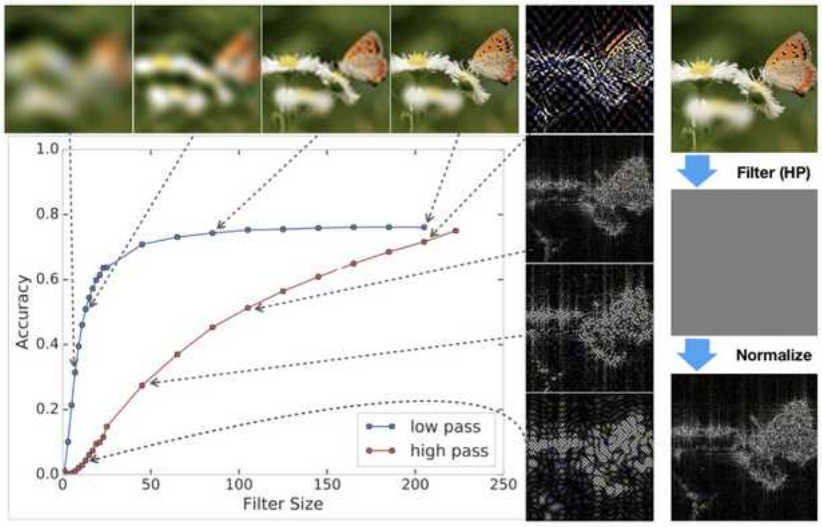


Figure 1: Models can achieve high accuracy using information from the input that would be unrecognizable to humans. Shown above are models trained and tested with aggressive high and low pass filtering applied to the inputs. With aggressive low-pass filtering, the model is still above 30% on ImageNet when the images appear to be simple globs of color. In the case of high-pass (HP) filtering, models can achieve above 50% accuracy using features in the input that are nearly invisible to humans. As shown on the right hand side, the high pass filtered images needed be normalized in order to properly visualize the high frequency features (the method that we use to visualize the high pass filtered images is provided in the appendix).

图片说明翻译：使用人类无法识别的输入信息，模型可以实现高精度。上面显示的是经过训练和测试的模型，这些模型在输入端应用了严格的高通和低通滤波。通过积极的低通滤波，当图像看起来是简单的彩色球体时，该模型在ImageNet上仍然高于30%。在高通（HP）过滤的情况下，使用人类几乎看不见的输入特征，模型可以达到50%以上的精度。如右图所示，需要对高通滤波图像进行归一化处理，以便正确地可视化高频特征（我们用附录中提供的可视化高通滤波图像的方法）。

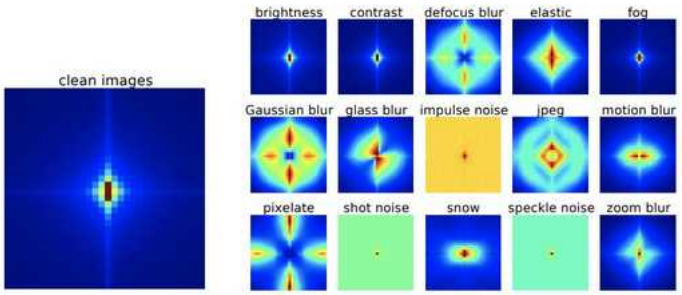


Figure 2: Left: Fourier spectrum of natural images; we estimate $\mathbb{E}[|\mathcal{F}(X)[i,j]|]$ by averaging all the CIFAR-10 validation images. Right: Fourier spectrum of the corruptions in CIFAR-10-C at severity 3. For each corruption, we estimate $\mathbb{E}[|\mathcal{F}(C(X)-X)[i,j]|]$ by averaging over all the validation images. Additive noise has relatively high concentrations in high frequencies while some corruptions such as fog and contrast are concentrated in low frequencies.

图片说明翻译：左：自然图像的傅里叶谱；我们通过平均所有CIFAR-10验证图像来估计E[|F(X)[i，j]|]。右：CIFAR-10-C中严重程度为3的被腐蚀的傅里叶谱。对于每个腐蚀点，我们通过平均所有验证图像来估计E[|F(C(X)−X)[i，j]|]。加性噪声在高频段具有较高的浓度，而雾、对比度等污染集中在低频段。

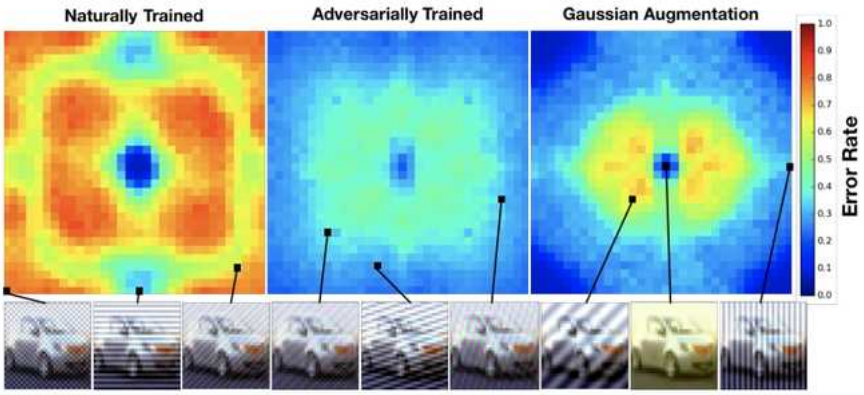


Figure 3: Model sensitivity to additive noise aligned with different Fourier basis vectors on CIFAR-10. We fix the additive noise to have ℓ_2 norm 4 and evaluate three models: a naturally trained model, an adversarially trained model, and a model trained with Gaussian data augmentation. Error rates are averaged over 1000 randomly sampled images from the test set. In the bottom row we show images perturbed with noise along the corresponding Fourier basis vector. The naturally trained model is highly sensitive to additive noise in all but the lowest frequencies. Both adversarial training and Gaussian data augmentation dramatically improve robustness in the higher frequencies while sacrificing the robustness of the naturally trained model in the lowest frequencies (i.e. in both models, blue area in the middle is smaller compared to that of the naturally trained model).

图片翻译说明：CIFAR-10上不同傅立叶基向量对加性噪声的模型灵敏度。我们将加性噪声固定为“L2范数为4”，并评估了三个模型：自然训练模型、对抗训练模型和高斯数据增强训练模型。对来自测试集中的1000个随机采样的图像进行平均错误率。在最下面的一行中，我们显示了沿着相应的傅立叶基向量受到噪声干扰的图像。自然训练的模型对除最低频率以外的所有加性噪声都高度敏感。对抗性训练和高斯数据增强都极大地提高了高频下的鲁棒性，而牺牲了自然训练模型在低频率下的鲁棒性(即，在这两个模型中，中间的蓝色区域比自然训练模型的小)。

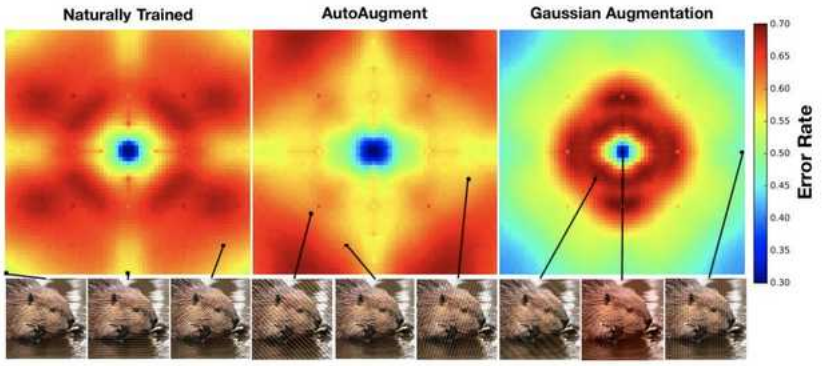


Figure 4: Model sensitivity to additive noise aligned with different Fourier basis vectors on ImageNet validation images. We fix the basis vectors to have ℓ_2 norm 15.7. Error rates are averaged over the entire ImageNet validation set. We present the 63×63 square centered at the lowest frequency in the Fourier domain. Again, the naturally trained model is highly sensitive to additive noise in all but the lowest frequencies. On the other hand, Gaussian data augmentation improves robustness in the higher frequencies while sacrificing the robustness to low frequency perturbations. For AutoAugment, we observe that its Fourier heat map has the largest blue/yellow area around the center, indicating that AutoAugment is relatively robust to low to mid frequency perturbations.

图片翻译说明：ImageNet验证图像上的不同傅立叶基向量对加性噪声的模型敏感度。我们将基向量固定为L2范数的值等于15.7。错误率是整个ImageNet验证集的平均错误率。给出了以傅里叶域最低频率为中心的63×63平方。同样，自然训练的模型对除最低频率之外的所有加性噪声都高度敏感。另一方面，高斯数据增强提高了高频下的鲁棒性，同时牺牲了对低频扰动的鲁棒性。对于AutoAugment，我们观察到它的傅立叶热图在中心周围有最大的蓝色/黄色区域，这表明

AutoAugment对低频到中频的破坏是相对健壮的。

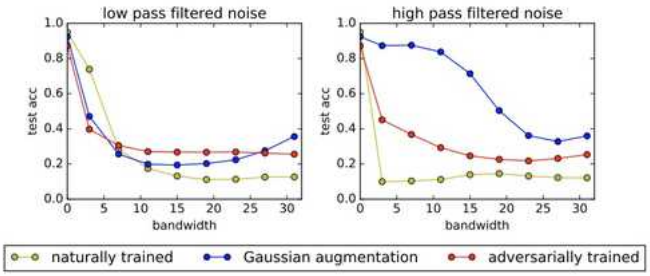


Figure 5: Robustness of models under additive noise with fixed norm and different frequency distribution. For each channel in each CIFAR-10 test image, we sample i.i.d Gaussian noise, apply a low/high pass filter, and normalize the filtered noise to have ℓ_2 norm 8, before applying to the image. We vary the bandwidth of the low/high pass filter and generate the two plots. The naturally trained model is more robust to the low frequency noise with bandwidth 3, while Gaussian data augmentation and adversarial training make the model more robust to high frequency noise.

图片翻译说明：固定范数和不同频率分布的加性噪声下模型的稳健性。对于每个CIFAR-10测试图像中的每个通道，在应用到图像之前，我们对独立同分布高斯噪声进行采样，应用低/高通滤波器，并将滤波后的噪声归一化为L2范数值为8。我们改变低/高通滤波器的带宽，生成两个曲线图。自然训练的模型对带宽为3的低频噪声具有更强的鲁棒性，而高斯数据增强和对抗性训练使模型对高频噪声具有更强的鲁棒性。

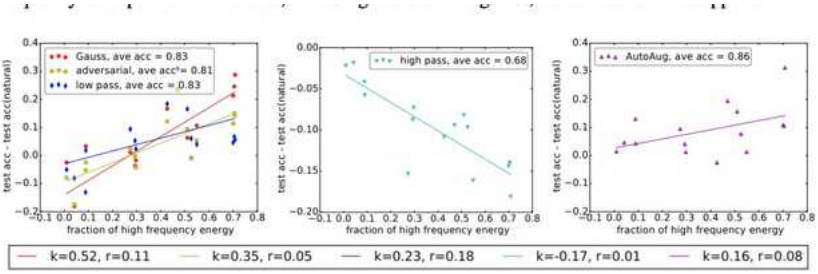


Figure 6: Relationship between test accuracy and fraction of high frequency energy of the CIFAR-10-C corruptions. Each scatter point in the plot represents the evaluation result of a particular model on a particular corruption type. The x-axis represents the fraction of high frequency energy of the corruption type, and the y-axis represents change in test accuracy compared to a naturally trained model. Overall, Gaussian data augmentation, adversarial training, and adding low pass filter improve robustness to high frequency corruptions, and degrade robustness to low frequency corruptions. Applying a high pass filter front end yields a more significant accuracy drop on high frequency corruptions compared to low frequency corruptions. AutoAugment improves robustness on nearly all corruptions, and achieves the best overall performance. The legend at the bottom shows the slope (k) and residual (r) of each fitted line.

图片翻译说明：CIFAR-10-C腐蚀高频能量分数与测试精度的关系。绘图中的每个散布点代表特定模型对特定损坏类型的评估结果。X轴表示损坏类型的高频能量的分数，y轴表示与自然训练的模型相比测试精度的变化。总体而言，高斯数据增强、对抗性训练和添加低通滤波器提高了对高频破坏的鲁棒性，降低了对低频破坏的鲁棒性。与低频损坏相比，应用高通滤波器前端对高频损坏产生更显著的精度下降。AutoAugment提高了对几乎所有损坏的健壮性，并实现了最佳的整体性能。底部的图例显示了每条拟合线的斜率(K)和残差(r）。

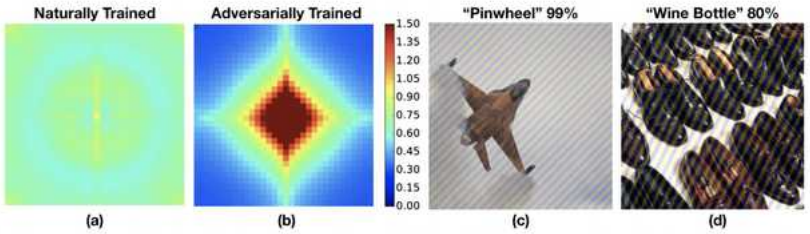


Figure 7: (a) and (b): Fourier spectrum of adversarial perturbations. For any image X, we run the PGD attack [22] to generate an adversarial example $C(X)$. We estimate the Fourier spectrum of the adversarial perturbation, i.e., $\mathbb{E}[|\mathcal{F}(C(X) - X)[i, j]|]$, where the expectation is taken over the perturbed images which are incorrectly classified. (a) naturally trained; (b) adversarially trained. The adversarial perturbations for the naturally trained model are uniformly distributed across frequency components. In comparison, adversarial training biases these perturbations towards the lower frequencies. (c) and (d): Adding Fourier basis vectors with large norm to images is a simple method for generating content-preserving black box adversarial examples.

图片翻译说明：(a)和(b)：对抗扰动的傅立叶频谱，给定图片X，发起PGD攻击，得到对抗样本C(X)，估算对抗扰动的傅立叶频谱，会使得图片错误分类；(a) 是自然训练得到的频谱；(b)是对抗训练得到的频谱。自然训练模型的对抗性扰动均匀分布在频率分量上。相比之下，对抗性的训练使这些扰动偏向较低的频率。(C)和(D)：将范数大的傅立叶基向量加到图像上是一种生成内容保持黑盒对抗性示例的简单方法。

几点结论：

1．对抗训练会关注到一些高频分量，而非一味执迷于低频分量。

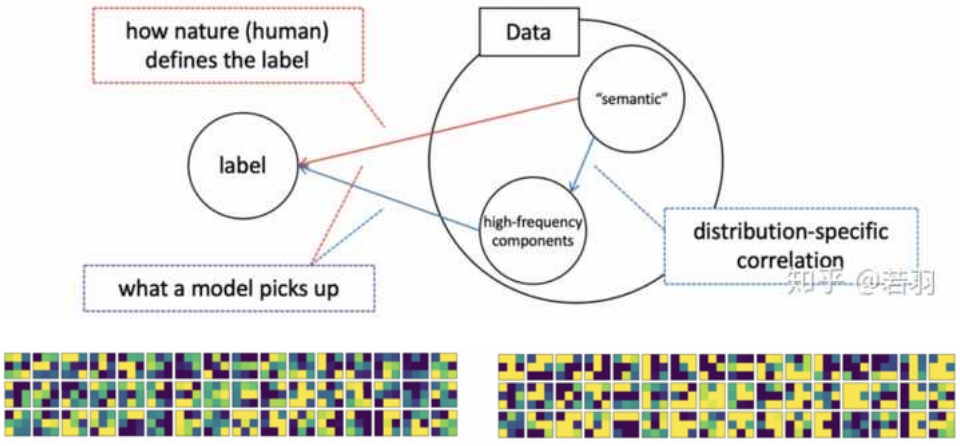
2）AutoAugment有助于提高鲁棒性

开源代码主要教人画出论文中类似的示意图。

另外一篇论文Eric Xing组里的，知乎的自媒体之前发过了：

High-frequency Component Helps Explain the Generalization of Convolutional Neural Networks

https://arxiv.org/pdf/1905.13545.pdf



自然训练的卷积的可视化与对抗训练的卷积的可视化

该论文实验了几个方法：

- 对于一个训练好的模型，我们调整其权重，使卷积核变得更加平滑；
- 直接在训练好的卷积核上将高频信息过滤掉；
- 在训练卷积神经网络的过程中增加正则化，使得相邻位置的权重更加接近。

然后得出结论：

关注低频信息，有助于提高泛化性，高频分量可能与对抗攻击有联系，但不能太武断。

Contribution是用详细的实验证明Batch Normalization对于拟合高频分量，提高泛化性是有用的。

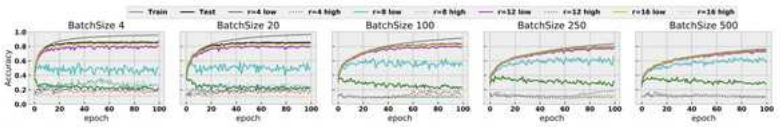


Figure 4. Plots of accuracy of different epoch sizes along the epoches for train, test data, as well as LFC and HFC with different radii.

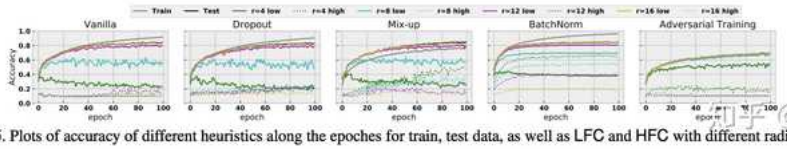


Figure 5. Plots of accuracy of different heuristics along the epoches for train, test data, as well as LFC and HFC with different radii.

最后，就是全凭一张嘴了。

这边厢，许教授证明ReLU的光滑性有助于函数优化；那边厢，近期的一个工作叫 *Bandlimiting Neural networks against adversarial attacks*

https://arxiv.org/pdf/1905.12797.pdf

ReLU函数得到一种piecewise的linear function

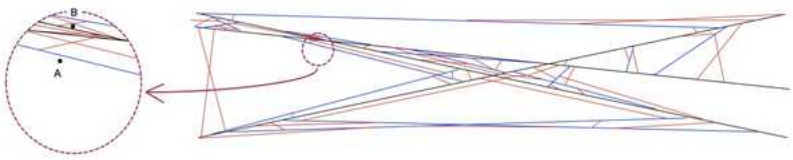


Figure 1: Illustration of input space divided into sub-regions by a biased neural network. The black lines are the hyperplanes for the first network layer, while the blue lines are for the second layer and the red lines are for the third layer. A small perturbation from point A to point B may possibly cross many hyperplanes.

可以分解为众多的频率分量。

> 对于N=1000个节点的隐藏层，并且输入维度为n=200时，区域的最大数目大致等于10^200。换言之，即使是一个中等规模的神经网络也可以将输入空间划分为大量的子区域，这很容易超过宇宙中的原子总数。当我们学习神经网络时，我们不能期望每个区域内至少有一个样本。对于那些没有任何训练样本的区域，其中的结果线性函数可以是任意的，因为它们根本不对训练目标函数有贡献。当然，这些地区中的大多数都非常小。当我们测量整个空间的预期损失函数时，它们的贡献可以忽略不计，因为随机抽样点落入这些微小区域的机会非常小。然而，对抗性攻击带来了新的挑战，因为对抗性样本不是自然抽样的。考虑到区域的总数是巨大的，那么这些微小的区域在输入空间中几乎无处不在。对于输入空间中的任何一个数据点，我们几乎肯定可以找到这样一个微小的区域，其中线性函数是任意的。如果选择了这个微小区域内的一个点，神经网络的输出可能会出乎意料。这些微小的区域是神经网络易受敌意攻击的根本原因。

然后，提出了一种对抗防御的方法，表示没看懂，看官自己读论文，欢迎读完在评论区点拨我。

3 The proposed defence approach: post-averaging

3.1 Post-averaging

In this paper, we propose a simple post-processing method to smooth out those high frequency components as much as possible, which relies on a simple idea similar to moving-average in one-dimensional sequential data. Instead of generating prediction merely from one data point, we use the averaged value within a small neighborhood around the data point, which is called *post-averaging* here. Mathematically, the post-averaging is computed as an integral over a small neighborhood centered at the input:

$$f_C(\mathbf{x}) = \frac{1}{\mathbf{V}_C} \int \cdots \int_{\mathbf{x}' \in C} f(\mathbf{x} - \mathbf{x}')\, \mathrm{d}\mathbf{x}' \tag{5}$$

where $\mathbf{x}$ is the input and $f(\mathbf{x})$ represents the output of the neural network, and C denotes a small neighborhood centered at the origin and $\mathbf{V}_C$ denotes its volume. When we choose C to be an n-sphere in $\mathbb{R}^n$ of radius r, we may simply derive the Fourier transform of $f_C(\mathbf{x})$ as follows:

$$F_C(\boldsymbol{\omega}) = F(\boldsymbol{\omega}) \frac{1}{\mathbf{V}_C} \int \cdots \int_{\mathbf{x}' \in C} e^{-\mathrm{i}\mathbf{x}' \cdot \boldsymbol{\omega}}\, \mathrm{d}\mathbf{x}' = F(\boldsymbol{\omega}) \frac{\Gamma(\frac{n}{2}+1)}{\pi^{\frac{n}{2}}} \frac{J_{\frac{n}{2}}(r|\boldsymbol{\omega}|)}{(r|\boldsymbol{\omega}|)^{\frac{n}{2}}} \tag{6}$$

where $J_{\frac{n}{2}}(\cdot)$ is the first kind Bessel function of order $n/2$. Since the Bessel functions, $J_\nu(\omega)$, decay with rate $1/\sqrt{\omega}$ as $|\omega| \to \infty$ (Watson, 1995), we have $F_C(\boldsymbol{\omega}) \sim \frac{F(\boldsymbol{\omega})}{(r|\boldsymbol{\omega}|)^{\frac{n+1}{2}}}$ as $|\boldsymbol{\omega}| \to \infty$. Therefore, if r is chosen properly, the post-averaging operation can significantly bandlimit neural networks by smoothing out high frequency components. Note that the similar ideas have been used in (Jiang et al., 1999; Jiang and Lee, 2003) to improve robustness in speech recognition.

3.2 Sampling methods

However, it is intractable to compute the above integral for any meaningful neural network used in practical applications. In this work, we propose to use a simple numerical method to approximate it. For any input $\mathbf{x}$, we select K points in the neighborhood C centered at $\mathbf{x}$, i.e. $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_K\}$, the integral is approximately computed as

$$f_C(\mathbf{x}) \approx \frac{1}{K} \sum_{k=1}^{K} f(\mathbf{x}_k). \tag{7}$$

Obviously, in order to defend against adversarial samples, it is important to have samples outside the current unlearned tiny region. In the following, we use a simple sampling strategy based on directional vectors. To generate a relatively even set of samples for eq.(7), we first determine some directional vectors $\hat{\mathbf{v}}$, and then move the input $\mathbf{x}$ along these directions using several step sizes within the sphere of radius r:

$$\mathbf{x}' = \mathbf{x} + \lambda \cdot \hat{\mathbf{v}} \tag{8}$$

where $\lambda = [\pm\frac{r}{3}, \pm\frac{2r}{3}, \pm r]$, and $\hat{\mathbf{v}}$ is a selected unit-length directional vector. For each selected direction, we generate six samples within C along both the positive and the negative directions to ensure efficiency and even sampling. Here, we propose two different methods to sample directional vectors:

- **random**: Random sampling is the simplest and most efficient method that one can come up with. We fill the directional vectors with random numbers generated from a standard normal distribution, and then normalize them to have unit length.

- **approx**: Instead of using random directions, it would be much more efficient to move out of the original region if we use the normal directions of the closest hyperplanes. In ReLU neural networks, each hidden node represents a hyperplane in the input space. For any input $\mathbf{x}$, the distance to each hyperplane may be computed as $d_k^{(n)} = \frac{a_k^{(n)}}{\|\hat{\mathbf{v}}_k^{(n)}\|}$, where $a_k^{(n)}$ denotes the output of the corresponding hidden node and $\hat{\mathbf{v}}_k^{(n)} = \nabla_{\mathbf{x}} a_k^{(n)}$. Based on all distances computed for all hidden nodes, we can select the normal directions for the K closest hyperplanes. However, computing the exact distances is computationally expensive as it requires back-propagation for all hidden nodes. In implementation, we simply estimate relative distances among all hidden units in the same layer using the weights matrix of this layer and select some closest hidden units in each layer based on the relative distances. In this way, we only need to back-propagate for the selected units. We refer this implementation as "*approx*" in the experimental results.

虽然有拖延症，但其他一些相关的、有趣的论文，我看到后也会在这个区分享的。