> 译者注：
>
> - 本文翻译自 Lars Hulstaert 于 2017 年 11 月 4 日发表的文章 Understanding objective functions in neural networks。
> - 文中括号或者引用块中的 *斜体字* 为对应的英文原文或者我自己注释的话（会标明「译者注」），否则为原文中本来就有的话。
> - 文中的「我」均指原作者 Lars Hulstaert 。
> - 目录保留英文原文。
> - 本人水平有限，如有错误欢迎指出。

这篇博客主要面向拥有一定机器学习经验的人，会帮助你直观理解在训练神经网络时所用到的各种不同的目标函数。

# Introduction

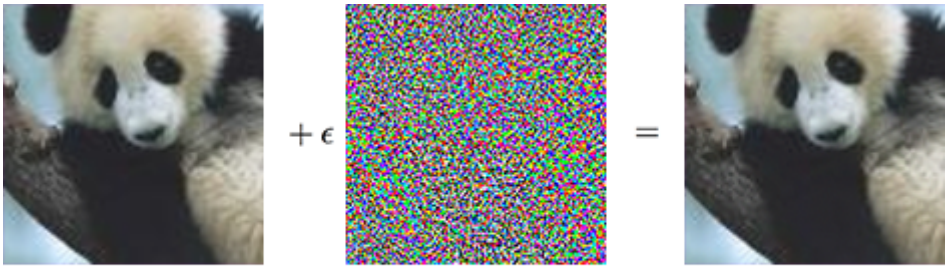
我写这篇博客的原因主要有 3 个：

- 其他博客中经常会解释优化算法，例如 SGD（*stochastic gradient descent*）及其变种，但是少有介绍神经网络中的目标函数是如何构建的。为什么回归和分类任务中的目标函数是均方误差（MSE）（*译者注：式 (2)*）和对数交叉熵损失（*译者注：式 (1)*）？为什么添加正则项说得通？一般来说，通过深入理解目标函数，我们可以了解为什么神经网络在一些情况下表现较好，而在一些情况下表现较差。

$$\mathrm{CE} = -\sum_{x} p(x) \log q(x) \tag{1}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

- 神经网络提供的是一种比较差的概率估计，并且一直遭受对抗样本（*adversarial example*）的困扰（*译者注：简单来说，就是机器学习模型可以很容易被一些特别设计的输入误导，例如在一个图像分类模型中，一个熊猫的图片稍微加一些噪声就可以导致被模型以高置信度识别成长臂猿，如下图所示，图自 Attacking Machine Learning with Adversarial Examples*）。简单来说，神经网络即使错了她也非常自信。如果在现实生活中应用（例如自动驾驶汽车），那么这就是一个问题了。一个自动驾驶汽车在以 90 迈速度行驶的时候做的决定必须是确定的。我们如果要部署深度学习模型，那么必须要了解他们的优点和弱点。




- 我经常想如何从概率角度去解释神经网络和如何扩展到更广泛的机器学习模型框架。人们倾向于把神经网络的输出视为概率，那么在神经网络的概率解释和他们的目标函数之间有没有什么联系?

这篇博文的主要灵感来源于我在剑桥大学计算和生物学习实验室（*Computational and Biological Learning Lab*）与我的朋友 Brian Trippe 就贝叶斯神经网络所做的工作。我强烈建议你们去阅读 Brian 关于神经网络中的变分推理的论文。

# Supervised machine learning

在监督机器学习问题中，我们通常有一个由许多 $(x, y)$ 数据对组成的数据集 $D$，然后试图去为下面的分布建模：

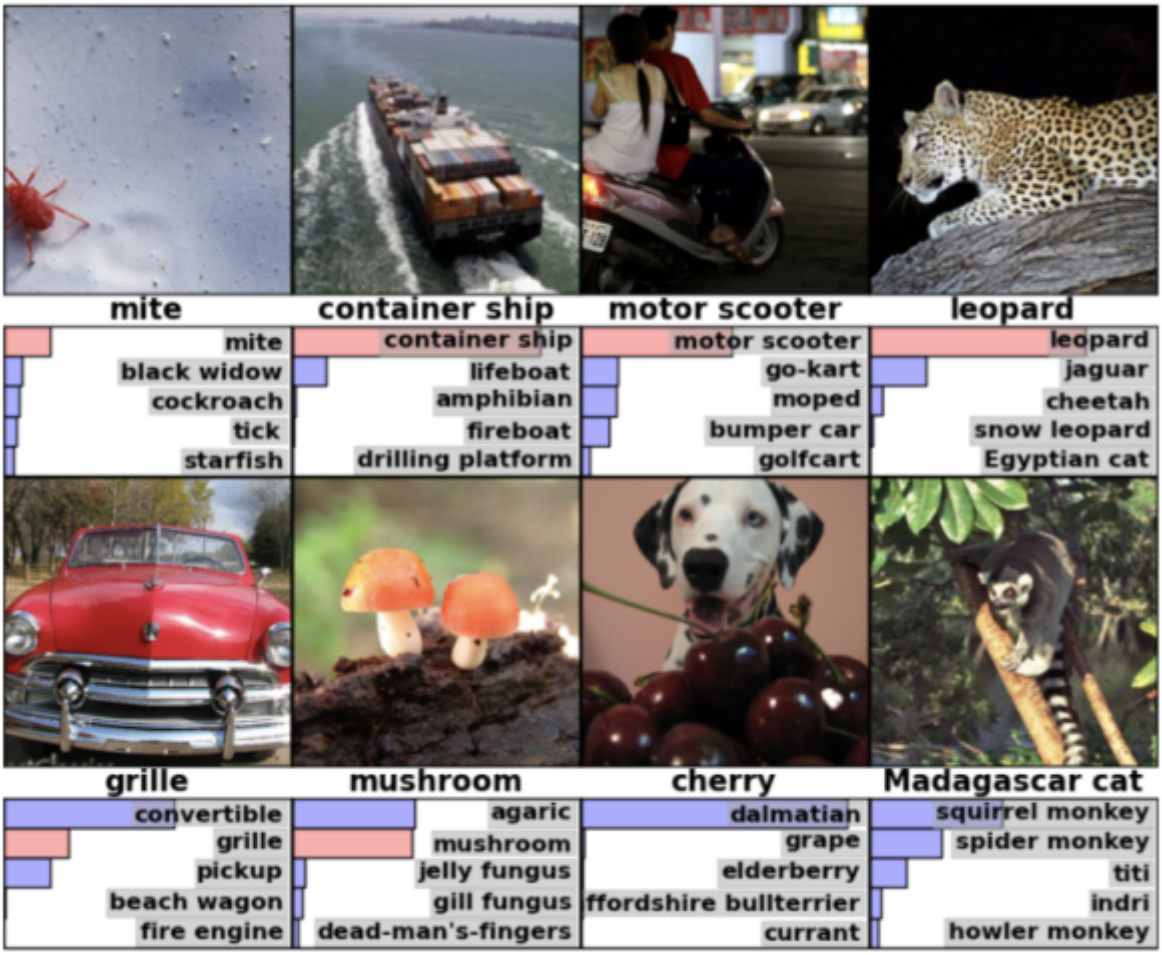
$$p(y|x, \theta)$$

例如在图像分类中，$x$ 表示一幅图像，$y$ 表示对应的图像标签，$p(y|x, \theta)$ 就表示图像 $x$ 的标签是 $y$ 的概率，该模型由参数 $\theta$ 定义。

这样的模型称为判别模型（discriminative models）。在判别或者条件模型中，条件概率分布函数 $p(y|x, \theta)$ 中的参数 $\theta$ 由训练数据推断得到。

基于观察数据（输入数据或者特征值），模型输出一个概率分布，然后据此分布来预测 $y$（类别或者真实值）。不同的模型需要估计的参数不同。线性模型（例如逻辑斯蒂回归，由一个大小和特征个数相同的权重集来定义）和非线性模型（例如神经网络，由每层的权重来定义）都可以用来近似这个条件概率分布。

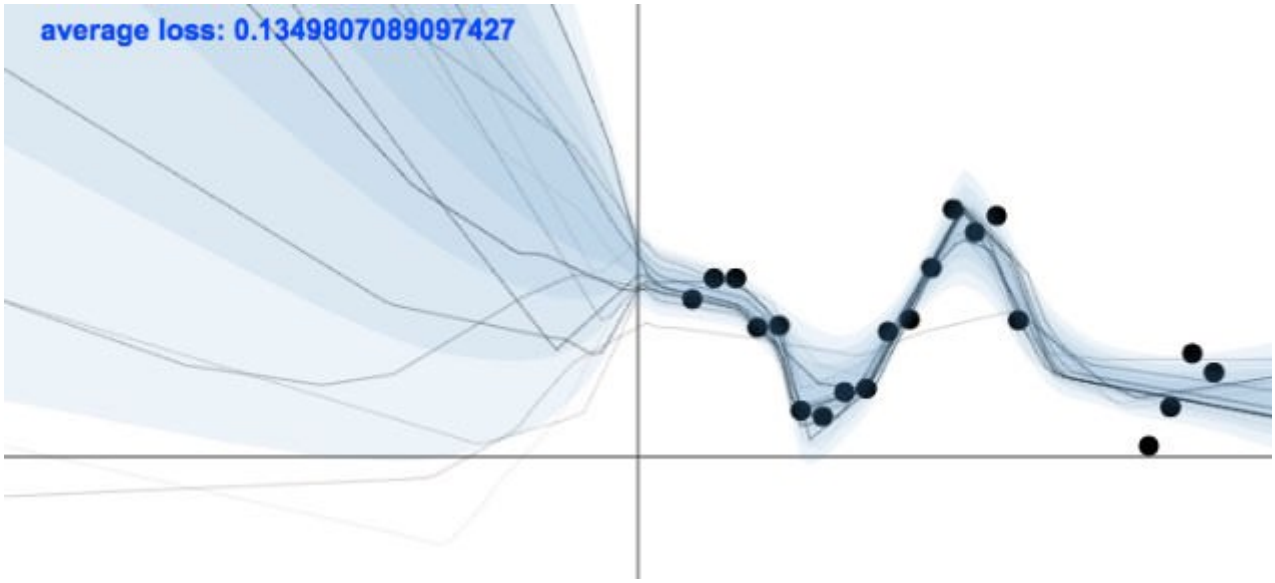
对于典型的分类问题来说，使用可学习的参数集 $\theta$ 来定义从 $x$ 到一个在不同标签上的范畴分布（*categorical distribution*） 的映射。一个判别式分类模型会输出 $N$ 个概率值，$N$ 就是类别数目。每一个 $x$ 属于一个单独的类别，但是模型最终输出的是在多个类别上的概率分布反映了其不确定性。一般来说，选择具有最大概率的类别作为最终结果。




需要注意的是判别式回归模型输出的是一个单独的预测值，而不是一个在所有实数上的分布。这和判别式分类模型有所不同，在分类模型中输出的是在所有类别上的分布。这是否意味着回归模型不是判别式模型？（*Does this mean discriminative models fall apart for regression?*）模型的输出不是应该告诉我们哪个值比其他值具有更高的概率吗？

判别式回归模型的输出虽然比较有误导性（*misleading*），但是确实和一个众所周知的分布有关，高斯分布。事实证明，判别式回归模型的输出是一个高斯分布（一个高斯分布完全由平均值和标准差定义）的平均值。在这个前提下，你可以确定每个实数在给定输入 $x$ 下的概率。

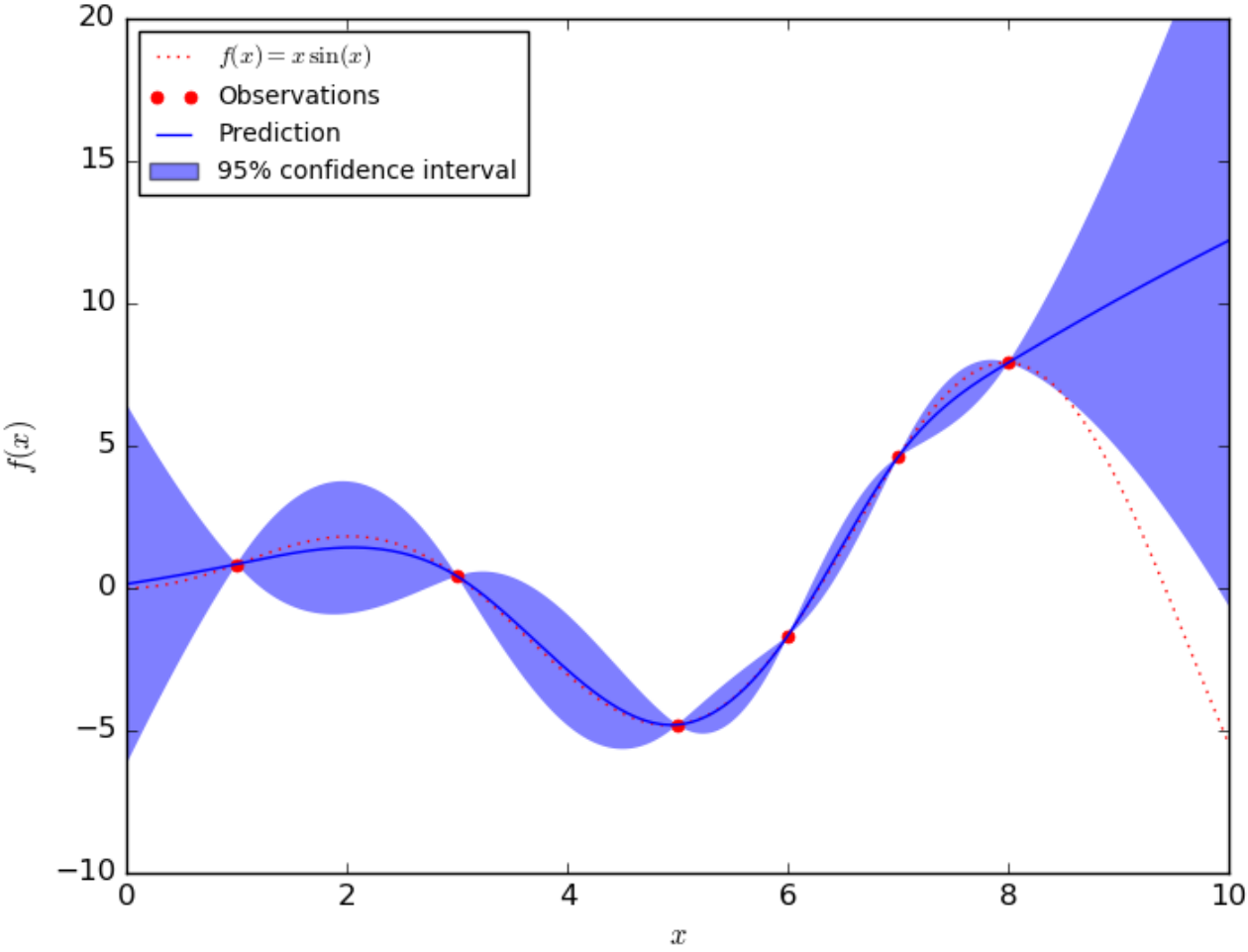
实际上只有这个高斯分布的平均值被建模了，而对应的标准差要不就是没有被建模，要不就是在所有的 $x$ 上都被设定成一个相同的常数。因此再判别式回归模型中，参数 $\theta$ 定义的是一个从 $x$ 到 $y$ 的高斯分布均值的映射。在决策时总是选择均值作为输出。同时输出均值和标准差的模型能传达出更多的信息，此时我们可以知道哪个 $x$ 是不确定的（判断标准差的大小）。




一个模型在没有训练数据的地方需要是不确定的，而在由训练数据的地方需要是确定的。上图就是显示了这样一个模型，图自Yarin Gal 的博文。

其他概率模型（如高斯过程）在回归问题的不确定性建模方面做得更好，而在同时建模均值和标准差时，判别式回归模型往往过于自信。

一个高斯过程可以明确地对标准差进行建模来量化不确定性。高斯过程唯一的不足之处是在大数据集上表现欠佳。在下图中可以看到在有很多数据的区域，高斯过程模型的置信区间（由标准差决定）很小，而在数据比较少的时候，置信区间变得非常大。




高斯模型在有数据点的地方是确定的，而在其他地方则是不确定的，图自 sklearn

判别模型在训练数据集上进行训练，以便学习数据中表示类别或实际值的属性。如果模型能够在测试集中为正确的样本类别分配高概率，或者接近于真实值的平均值，则模型表现良好。

# Link with neural networks

当训练神经网络用于分类或者回归任务时，上述分布（范畴分布和高斯分布）的参数会使用神经网络来建模。

当我们试图使用极大似然估计（MLE）来确定神经网络的参数 $\theta$ 时，这就变得清楚了。MLE 用于找到能够使得训练数据似然（或者对数似然）最大的参数 $\theta$。更具体来说，就是使得下式最大化：

$$\begin{aligned}\theta^{\text{MLE}} &= \text{argmax}_\theta \log p(Y|X,\theta) \\ &= \text{argmax}_\theta \log \prod_{i=1}^{N} p(y_i|x_i,\theta) \\ &= \text{argmax}_\theta \sum_{i=1}^{N} \log p(y_i|x_i,\theta)\end{aligned} \tag{3}$$

其中 $p(Y|X,\theta)$ 表示训练集中真实标签的分布。当 $p(Y|X,\theta)$ 接近于 1 时，就表示模型可以确定训练集中的真实标签或者均值。考虑到训练数据 $(X,Y)$ 是由 $N$ 个观察数据对组成的，训练数据的似然可以被重写为对数似然的和。

在进行分类和回归时，$p(y|x,\theta)$，即单个数据对 $(x,y)$ 的后验概率，可以被重写为范畴分布和高斯分布。在优化神经网络时，目标是不断地更新参数（*译者注：这里应该指的是神经网络的参数*）以至于给定一个输入集 $X$，可以正确输出概率分布 $Y$ 的参数（回归值或者类别）。这通常通过梯度下降算法及其变种来实现。所以为了获得一个 MLE 估计，目标就是根据真实输出来优化模型输出：

- **最大化一个范畴分布的对数也就相当于最小化近似分布和真实分布之间的交叉熵。**
- 最大化一个高斯分布的对数也就相当于最小化近似均值和真实均值之间的均方误差。

所以上式也就可以分别写成交叉熵损失和均方误差，也就是神经网络用于分类和回归时的目标函数。

相比于更传统的概率模型来说，神经网络通过从输入到概率或者均值学习到的非线性函数比较难以解释。虽然这是一个神经网络的重要缺点，但是神经网络可以对复杂函数进行建模，这也是一个巨大的优势。根据本节的推导，在神经网络中，很明显用于确定参数的极大似然估计的目标函数可以从概率角度进行解释。

一个关于神经网络的有趣解释是她们与广义线性模型（线性回归、逻辑斯蒂回归……）的关系。神经网络不是将特征进行线性组合（像 GLM 中那样），而是高度的非线性组合。

# Maximum-a-posteriori

但是，神经网络如果可以被解释为概率模型，那么为什么她们产生的是一个比较差的概率估计，而且遭受着对抗样本的困扰？为什么她们需要如此多的数据？

我喜欢将不同的模型（逻辑斯蒂回归、神经网络……）想象为在不同的搜索空间中寻找好的函数逼近器（*function approximators*）。虽然具有非常大的搜索空间意味着在建模后验概率时有很大的灵活性，但它也是有代价的。例如神经网络被证明是通用函数逼近器，，这意味着只要有足够的参数她们可以逼近任何函数（awesome！）。然而为了保证函数能够在整个数据空间内得到校准（calibrated），这就需要非常大的数据集了（expensive！）。

首先要指出的是标准的神经网络是用 MLE 来进行优化的。这样会导致过度拟合训练数据和需要很多数据才能得到一个比较好的结果。机器学习的目的不是找到一个可以很好解释训练数据的模型。你会宁愿去找一个这样的模型：可以很好处理新数据（*译者注：泛化能力强*），而且对和训练数据明显不同的的数据保持谨慎（*unsure for data that is significantly different from the train data.*）。

使用最大后验概率（*maximum-a-posteriori*，MAP）方法是一个有效的选择，当模型有过拟合问题时可以试图使用这种方法。那么 MAP 在神经网络中对应于什么？其对目标函数有何影响？

和 MLE 类似，在神经网络中 MAP 同样可以被重写为一个目标函数。实际上在使用 MAP 的时候，假设 $\theta$ 服从一个先验分布，在给定数据的情况下，你是在最大化参数集 $\theta$ 的概率：

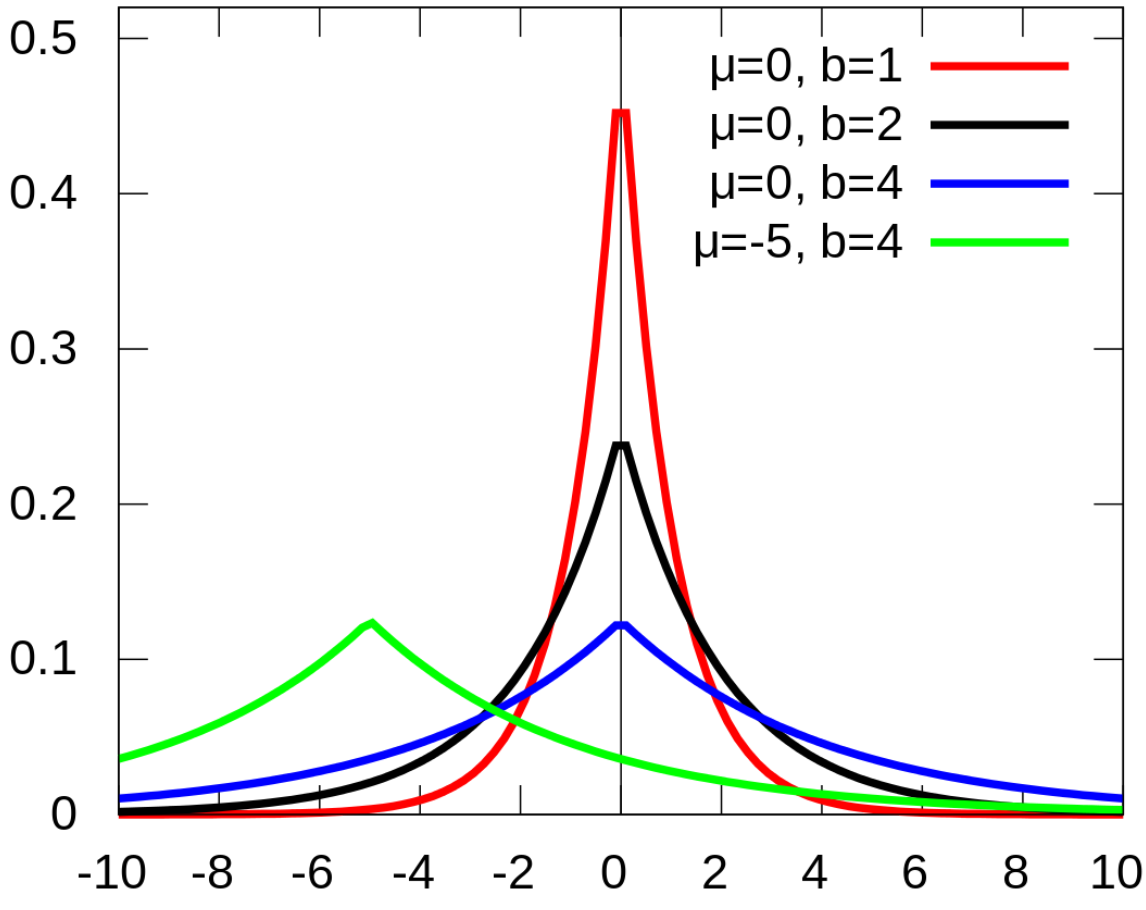
$$\begin{aligned}\theta^{\text{MAP}} &= \text{argmax}_\theta \log p(\theta|X,Y) \\ &\approx \text{argmax}_\theta \log p(Y|X,\theta) \cdot p(\theta)\end{aligned} \tag{4}$$

在使用 MLE 的使用，只有式上式中的第一项被考虑到了（模型对训练数据解释地有多好）（*译者注：参见式 (3)*）。而在 MAP 中，模型满足先前的假设（$\theta$ 符合先验分布的程度）以减少过拟合也很重要。

让 $\theta$ 服从一个期望为 0 的高斯分布相当于在目标函数后加上一个 L2 正则项（保证许多权重都比较小）（*译者注：式 (5)*），而让 $\theta$ 服从一个拉普拉斯分布（*Laplace distribution*）则相当于在目标函数后加上一个 L1 正则项（保证许多权重都为 0）（*译者注：式 (6)*）。

$$\lambda \sum_{i=1}^{N} |w_i| \tag{5}$$

$$\lambda \sum_{i=1}^{N} (w_i)^2 \tag{6}$$

> 译者注：为了便于阅读，我将拉普拉斯分布的部分相关信息放在这里，参考自 Laplace distribution - Wikipedia：
>
> - 概率密度函数图像：
>
> 
> 

> - 概率密度函数（PDF）：$\frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$
> - 期望、中位数、众数：$\mu$
> - 方差：$2b^2$

# A full Bayesian approach

在 MLE 和 MAP 中都是使用一个模型（只有一组参数）。特别是在图像这种复杂数据中，数据空间中的某些特定区域可能覆盖不到。模型在这些区域的输出取决于模型的随机初始化和训练过程，导致这些区域的概率估计很差。

尽管 MAP 能够保证模型不会在这些区域过拟合，但是这仍然会导致模型过度自信。在一个全贝叶斯方法中，这可以通过在多个模型上进行平均来解决，获得更好的不确定性估计。该方法的目的不是学习一个单独的参数集，而是学习到一个参数分布。如果所有的模型在未覆盖区域提供的是不同的估计，那么这就表明这些区域存在很大的不确定性。通过平均这些模型，最终的结果是一个在这些区域不确定的模型。这就是我们想要的！

在下篇博文中我将讨论贝叶斯神经网络（*Bayesian Neural Networks*）以及她们是如何解决传统神经网络中存在的上述问题的。贝叶斯神经网络（BNN's）仍然是一个进行中的研究工作，还没有很好地方法来训练她们。

我强烈推荐 Yarin Gal 的博文 Uncertainty in Deep Learning。