深度学习（Deep Learning）

关注者 415 被浏览 33,933

如果在前向传播的过程中使用了不可导的函数，是不是就不能进行反向传播了？

Houye
北京邮电大学 计算机科学与技术博士在读

专业 已有 2 人赠与了专业徽章

740 人赞同了该回答

这个问题非常有意思,我在刚接触深度学习的时候也疑惑过.当时主要是对ReLU激活函数在x=0的求导比较困惑,后来发现除了不可导的函数之外,深度学习中还有很多不可导的操作.

下面简单的梳理一下

Houye：盘点深度学习中的不可导操作(次梯度和重参数化)
zhuanlan.zhihu.com

深度学习中的不可导操作(次梯度和重参数化).

主要包括两大类

[TOC]

次梯度

深度学习算法通常需要反向传播来进行优化,这就涉及到求导的问题. 激活函数需要满足单调,处处可导,有界等条件. 如传统的sigmoid函数,但是现在很多激活函数并不是处处可导的.

如ReLU函数

$$ReLU(x) = max(0, x)$$

其图像如下

关于作者

Houye
公众号:【图与推荐】
北京邮电大学 计算机科学与技术博士在读

回答 85 文章 19 关注者 1,117

关注他 发私信

被收藏 733 次

机器学习/深度学习 444 人关注
凌景冰 创建

机器学习 381 人关注
黄锦华 创建

机器学习 206 人关注
tom pareto 创建

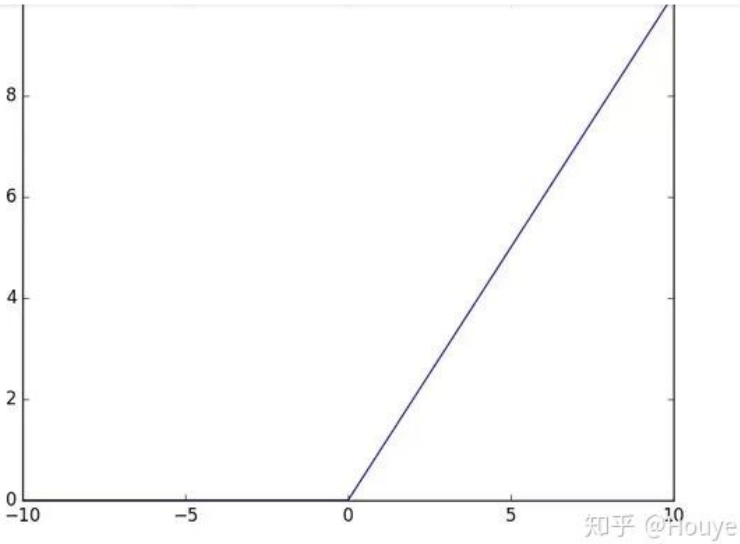
机器学习相关 52 人关注
魏天闻 创建

机器学习 6 人关注
浣熊侠 创建

相关问题

为什么代价函数要非负？ 9 个回答

在神经网络中，先进行BatchNorm还是先运行激活函数？ 16 个回答



很明显在 $x=0$ 处不可导,那么如何实现反向传播和模型优化呢? 答案就是:次梯度

次梯度

$$c<=\frac{f(x)-f(x_0)}{x-x_0}$$

对于ReLU函数, 当x>0的时候,其导数为1; 当x<0时,其导数为0. 则ReLU函数在x=0的次梯度是 $c \in [0,1]$,这里是次梯度有多个,可以取0,1之间的任意值. 工程上为了方便取c=0即可.

重参数技巧

VAE中对高斯分布的重参数

这里是对连续分布的重参数.

VAE中隐变量z一般取高斯分布,即 $z=\mathcal{N}(\mu,\sigma^2)$,然后从这个分布中采样.但是这个采样操作是**不可导**的,进而导致整个模型无法BP. 解决方法就是Reparametrization tricks重参数技巧.

我们首先从从均值为0,标准差为1的高斯分布中采样,再放缩平移得到Z.

$$\mathbf{z}_i=\mu_i+\sigma_i*\epsilon,\epsilon\sim\mathcal{N}(0,\mathbf{I})$$

这样从 ϵ 到 $\mathbf{z}$ 只涉及了线性操作(平移缩放),采样操作在NN计算图之外,而 ϵ 对于NN来说只是一个常数.

离散分布的采样Gumbel-softmax

Gumbel-Softmax Trick

VAE的例子是一个连续分布(正态分布)的重参数，离散分布的情况也一样，首先需要可以采样，使得离散的概率分布有意义而不是只取概率最大的值，其次需要可以计算梯度。那么怎么做到的，具体操作如下：

对于n维概率向量 π ,对 π 对应的离散随机变量 x_π 添加Gumbel噪声，再取样

$$x_\pi=\arg\max(\log(\pi_i)+G_i)$$

其中 G_i 是是独立同分布的标准Gumbel分布的随机变量，标准Gumbel分布的CDF为 $F(x)=e^{-e^{-x}},F^{-1}(x)=-\log(-\log(x))$.这就是**Gumbel-Max trick**。可以看到由于这中间有一

上述的 **argmax操作是不可导的**. 所以尝试用softmax来代替, 即**Gumbel-Softmax Trick**. 这里我们假设argmax返回的是一个one-hot向量,那么我们需要找到argmax的一个显式且光滑的逼近. 这里的 G_i 可以利用 $F_{-1}(x)$ 从均匀分布中采样得到,即 $G_i = -\log(-\log(U_i)), U_i \sim U(0,1)$.

综上总体思路:

1. 基于Gumbel Distribution采样来避免不可导问题
2. 在1中引入了argmax又导致了不可导(Gumbel max)
3. 又引入softmax函数来对argmax进行光滑近似,使得可导(Gumbel softmax)

具体步骤如下:

- 对于网络输出的一个n维向量v, 生成n个服从均匀分布U(0,1)的独立样本 $\epsilon_1, \ldots, \epsilon_n$
- 通过 $G_i = -\log(-\log(\epsilon_i))$ 计算得到 G_i
- 对应相加得到新的值向量 $v' = [v_1 + G_1, v_2 + G_2, \ldots, v_n + G_n]$
- 通过softmax函数

$$\sigma_\tau(v'_i) = \frac{e^{v'_i/\tau}}{\sum\limits_{j=1}^{n} e^{v'_j/\tau}}$$

这里 $\sigma_\tau(v'_i)$ 就可以实现对argmax的显式且光滑的逼近

$$\lim_{\tau \to 0} \sigma_\tau(v'_i) = argmax$$

温度参数 τ 的影响: τ 越小(趋近于0), 越接近categorical分布; τ 越大(趋近于无穷), 越接近均匀分布

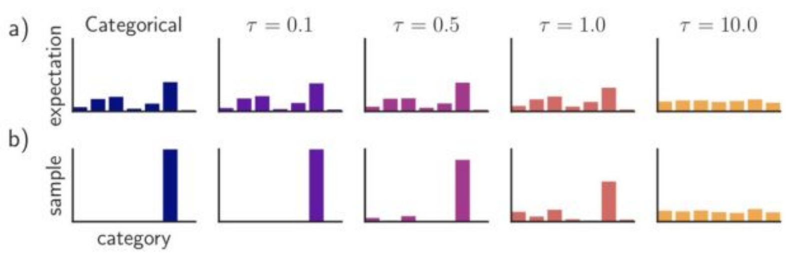


Figure 1: The Gumbel-Softmax distribution interpolates between discrete one-hot-encoded categorical distributions and continuous categorical densities. (a) For low temperatures ($\tau = 0.1, \tau = 0.5$), the expected value of a Gumbel-Softmax random variable approaches the expected value of a categorical random variable with the same logits. As the temperature increases ($\tau = 1.0, \tau = 10.0$), the expected value converges to a uniform distribution over the categories. (b) Samples from Gumbel-Softmax distributions are identical to samples from a categorical distribution as $\tau \to 0$. At higher temperatures, Gumbel-Softmax samples are no longer one-hot, and become uniform as $\tau \to \infty$.

证明

常规的softmax形式为

$$\pi_k = \frac{e^{x_k}}{\sum_{k'=1}^{K} e^{x'_k}}$$

其中, π_k 是softmax之后得到一个概率密度函数. 那么有没有某个分布能够等价于上述的分布呢?

如果对每个 x_k 添加独立标准的gumbel噪声(位置为0,尺度为1),并选择值最大的维度输出,每次的输出结果有一个概率密度函数.这样一个概率密度同样为 π_k .

化简

$$
\begin{aligned}
&= \int e^{-\sum_{k'\neq k} e^{-(z_k - x_{k'})} - (z_k - x_k) - e^{-(z_k - x_k)}}\, dz_k \\
&= \int e^{-\sum_{k'=1}^{K} e^{-(z_k - x_{k'})} - (z_k - x_k)}\, dz_k \\
&= \int e^{-(\sum_{k'=1}^{K} e^{x_{k'}}) e^{-z_k} - z_k + x_k}\, dz_k \\
&= \int e^{-e^{-z_k + \ln(\sum_{k'=1}^{K} e^{x_{k'}})} - z_k + x_k}\, dz_k \\
&= \int e^{-e^{-(z_k - \ln(\sum_{k'=1}^{K} e^{x_{k'}}))} - (z_k - \ln(\sum_{k'=1}^{K} e^{x_{k'}})) - \ln(\sum_{k'=1}^{K} e^{x_{k'}}) + x_k}\, dz_k \\
&= e^{-\ln(\sum_{k'=1}^{K} e^{x_{k'}}) + x_k} \int e^{-e^{-(z_k - \ln(\sum_{k'=1}^{K} e^{x_{k'}}))} - (z_k - \ln(\sum_{k'=1}^{K} e^{x_{k'}}))}\, dz_k \\
&= \frac{e^{x_k}}{\sum_{k'=1}^{K} e^{x_{k'}}} \int e^{-e^{-(z_k - \ln(\sum_{k'=1}^{K} e^{x_{k'}}))} - (z_k - \ln(\sum_{k'=1}^{K} e^{x_{k'}}))}\, dz_k \\
&= \frac{e^{x_k}}{\sum_{k'=1}^{K} e^{x_{k'}}} \int e^{-(z_k - \ln(\sum_{k'=1}^{K} e^{x_{k'}})) - e^{-(z_k - \ln(\sum_{k'=1}^{K} e^{x_{k'}}))}}\, dz_k
\end{aligned}
$$

积分里面是 $\mu = \ln(\sum_{k'=1}^{K} e^{x_{k'}})$ 的gumbel分布,整个积分为1,则

$$P(z_k \geq z_{k'}; \forall k' \neq k | \{x_{k'}\}_{k'=1}^{K}) = \frac{e^{x_k}}{\sum_{k'=1}^{K} e^{x_{k'}}}$$

结果与softmax的分布一致.

为什么需要gumbel-softmax

乍看起来，gumbel-softmax 的用处令人费解。比如上面的代码示例，直接使用 softmax，也可以达到类似的参数训练效果。但两者有着根本的区别。 原理上，常规的 softmax 直接建模了一个概率分布（多项分布），基于交叉熵的训练准则使分布尽可能靠近目标分布；而 gumbel-softmax 则是对多项分布采样的一个近似。使用上，常规的有监督学习任务（分类器训练）中，直接学习输出的概率分布是自然的选择；而对于涉及采样的学习任务（VAE 隐变量采样、强化学习中对actions 集合进行采样以确定下一步的操作），gumbel-softmax 提供了一种再参数化的方法，使得模型可以以端到端的方式进行训练。

Ref

CATEGORICAL REPARAMETERIZATION WITH GUMBEL-SOFTMAX
arxiv.org

救命稻草人：Reparametrization tricks重参数技巧（在VAE、Gumbel-softmax G...
zhuanlan.zhihu.com

The Gumbel-Softmax Trick for Inference of Discrete Variables
casmls.github.io

http://lips.cs.princeton.edu/the-gumbel-max-trick-for-discrete-distributions/
lips.cs.princeton.edu

https://blog.csdn.net/jackytintin/article/details/53641885
blog.csdn.net

大量tf代码实例
amid.fish

www.quora.com

https://towardsdatascience.com/beyond-the-derivative-subderivatives-...
towardsdatascience.com

编辑于 2019-12-15

14 条评论 切换为时间排序

精选评论（1）

Houye (作者) 1 个月前
@知乎小管家
1

评论（14）

Sherman Wong 1 个月前
reparametric trick的解释有点问题，可以reparametric是因为优化目标是概率分布的期望值
1

Houye (作者) 1 个月前
@知乎小管家
1

马卡斯·扬 1 个月前
真的是好文章,卧槽
1

李珂 1 个月前
看了之后我才知道VAE重参数的原因 看来我组会讲漏了[飙泪笑] 感谢
赞

Houye (作者) 回复 李珂 1 个月前
🤣🤣🤣 我也在组会讲过 不过讲的是后面的Gumbel

赞

讨厌吃酸的why 1 个月前
说白了就是用连续可导的函数近似离散的、不可导的吧
1

1965 1 个月前

Houye (作者) 回复 1965 1 个月前

没听过[捂脸][捂脸][捂脸]

赞

ShameOfL 1 个月前

虽然采样操作不可导，但是采样结果是可导的...这样说比较完整的感觉

1

陈茕 1 个月前

看到这个就想起VAE的Reparameter

1

秦睿 1 个月前

还差一个手段，叫直通梯度

赞

Houye (作者) 回复 秦睿 1 个月前

求个链接，我去研究下

赞

秦睿 回复 Houye (作者) 1 个月前

VQVAE中使用到了

赞

展开其他 1 条回复

写下你的评论...

更多回答

Young

放弃不难,但坚持一定很酷.

107 人赞同了该回答

传统的神经网络无论是隐层还是激活函数的导数都是可导，可以直接计算出导数函数，但是在CNN网络中存在一些不可导的特殊环节，比如Relu等不可导的激活函数、造成维数变化的池化采样、已经参数共享的卷积环节。

神经网络的反向传播本质就是梯度，理解了这一点就可以理解不可导的函数是怎么进行反向传播的了。

Relu函数在数学上的定义为连续不可微的函数，它的定义为：

当x>0时，Relu(x) = x, 当x<=0时，Relu(x) = 0。

Relu函数在x=0处是不可微的，但是在深度学习框架的代码中为了解决这个直接将其在x=0处的导数置为1，所以它的导数也就变为了，即:

当x>0时，Relu'(x) = 1, 当x<=0时，Relu'(x) = 0。

展开阅读全文

赞同 107 6 条评论 分享 收藏 喜欢 ···

萧潇

PhD Student in Computational Math @ UChicago estimator，很常见，比如VQ-VAE用的)，也可以前传的时候用argmax，反传的时候用argmax的某种连续的近似值(Gumbel-softmax)，还可以用direct optimization的方法直接优化argmax，把梯度写成

展开阅读全文