



ChatGPT文本内容记录

syj2347

目录

1 草坪喷头灌溉问题	1
2 SVM	3
3 CNN相关	5

1 草坪喷头灌溉问题

草坪灌溉通常有两种模式——固定喷头和移动喷头。固定喷头一般用于管理水平较高的运动场上，它的主要特点是可以自由旋转、操作简单、数量较少；而移动喷头通常用于普通绿地，它的主要特点是投入成本低，但在操作上费时费工。对固定喷头，如果其分布不合理，会导致喷水不均匀的情况，也会因喷水到草坪之外，产生水资源的浪费。请建立数学模型，讨论固定喷头的合理布局问题。

问题 1. 对一块矩形草坪，建立固定喷头布局的数学模型，使得喷水尽可能均匀，且浪费的水尽量少。

问题 2. 对问题 1 给出的固定喷头的分布情况，确定喷水频率，使得草坪保持在合理的湿润程度。

问题 3. 对不同形状的草坪，问题 1 和问题 2 的模型应如何进行调整？^a

^a2023年第二十五届华东杯数模竞赛赛题A题

问题1：

为了使得喷水尽可能均匀，我们可以采用网格划分的方法，将矩形草坪划分为若干个小正方形，每个小正方形安装一个固定喷头。

为了使得浪费的水尽量少，我们需要使得每个小正方形的边界恰好处于喷头的喷水范围之内。设每个小正方形的边长为 l ，喷头的喷水半径为 r ，则需要满足 $l \leq 2r$ 。此时，每个小正方形与相邻小正方形的重叠部分就可以被同时喷水，减少浪费。

问题2：

为了确定喷水频率，需要考虑草坪的水分蒸发和渗漏。设草坪面积为 A ，每个小正方形的面积为 $a = l^2$ ，草坪的渗透率为 k ，水的蒸发率为 e ，则草坪的水分平衡方程为：

$$\frac{dW}{dt} = I - E - L - T$$

其中， W 表示草坪的水分含量， I 表示喷水的灌溉量， E 表示草坪的水分蒸发量， L 表示草坪表面水分的流失量， T 表示草坪土壤中的水流

失量。

我们假设固定喷头喷水的速率为 Q , 喷水时间为 t_0 , 则灌溉量 I 可以表示为:

$$I = Q \times t_0$$

草坪的水分蒸发量 E 可以表示为:

$$E = e \times A \times \Delta t$$

其中, Δt 表示时间间隔。

草坪表面水分的流失量 L 可以表示为:

$$L = k \times A \times \Delta W$$

其中, ΔW 表示草坪的水分含量下降量。

草坪土壤中的水分流失量 T 可以表示为:

$$T = k \times A \times (W - W_p) \times \Delta t$$

其中, W_p 表示草坪土壤的极限含水量。

综上, 我们可以得到草坪的水分平衡方程:

$$\frac{dW}{dt} = Q - e \times A - k \times A \times (\Delta W + W - W_p) - k \times A \times W \times \Delta t$$

根据草坪的实际情况, 可以确定合理的初始水分含量 W_0 和灌溉频率 Q , 以及草坪土壤的极限含水量 W_p 和草坪的渗透率 k 。然后可以采用数值模拟的方法, 利用数值方法求解草坪水分平衡方程, 得到草坪水分含量随时间的变化情况。同时, 可以根据实际情况, 调整喷水频率, 使得草坪水分保持在合适的范围内。

2 SVM

学习支持向量机相关内容^a

^a机器学习课程

SVM是支持向量机（Support Vector Machine）的缩写，它是一种用于分类和回归分析的机器学习算法。SVM通过在不同类别之间寻找一个最优的决策边界来进行分类，这个决策边界是由支持向量所定义的，支持向量是离决策边界最近的训练数据点。SVM的主要优势在于它可以有效地处理高维度和复杂的数据集，并且具有很好的泛化性能，能够在新的数据集上表现良好。SVM还有很多的变种，例如核SVM、多类SVM等，它们可以处理更加复杂的问题。SVM被广泛应用于图像识别、自然语言处理、生物信息学、金融预测等领域。

在支持向量机（SVM）中，我们希望找到一个最优的决策边界（超平面），以将数据分为不同的类别。当数据是线性可分的时候，我们可以找到一个硬间隔（hard margin）来完全分隔不同的类别。但是在实际情况下，数据往往不是完全线性可分的，这时候就需要使用软间隔（soft margin）来容忍一定程度的分类错误。

软间隔的基本思想是允许一些样本点出现在决策边界的错误一侧，使得分类器更加鲁棒和泛化能力更强。具体来说，我们引入一个松弛变量（slack variable）来描述每个样本点的分类错误程度，即样本点到正确的决策边界的距离。松弛变量的值越大，表示样本点越靠近错误一侧的区域。

在软间隔SVM中，我们要最小化目标函数：

$$\frac{1}{2}||w||^2 + C \sum_{i=1}^m \xi_i$$

其中， $\frac{1}{2}||w||^2$ 表示模型的复杂度， C 是一个超参数，用来控制模型的惩罚项。 $\sum_{i=1}^m \xi_i$ 表示所有样本的错误程度之和，即松弛变量的总和。我们希望尽可能的减小模型复杂度和错误程度，这相当于是一个权衡问题。当 C 取较大的值时，模型更加关注减小错误程度，这会导致模型的容错能力较低，可能会出现过拟合的情况；当 C 取较小的值时，模型更加关注减小模型复杂度，这会导致模型过于简单，可能会出现欠拟合的情况。

总的来说，软间隔SVM在一定程度上允许了分类错误的存在，使得模型更加健壮，并且可以处理不完全线性可分的数据集。但是，需要注意的

是，软间隔SVM需要对超参数 C 进行适当的调参，以达到最佳的性能。

3 CNN相关

GoogleNet的1*1卷积有什么用? ^a

^a课程提问

GoogleNet是一种深度卷积神经网络，它包含了多个卷积层和池化层来提取图像的特征。在GoogleNet中，1x1卷积被广泛应用于不同的层中，主要有以下两个用途：

1. 增加非线性映射能力

在卷积神经网络中，卷积层用于提取特征，而激活函数用于引入非线性映射能力，从而提高网络的表达能力。在GoogleNet中，1x1卷积的主要作用是增加网络的非线性映射能力。由于1x1卷积的感受野很小，它只能捕获局部特征。但是，在将多个1x1卷积层组合在一起时，可以实现更复杂的非线性变换，从而增强网络的表达能力。

2. 降低计算复杂度

在GoogleNet中，1x1卷积还可以用于降低计算复杂度。由于1x1卷积的感受野很小，它可以将通道数从较大的值减小到较小的值，从而减少计算量。同时，1x1卷积还可以减少过拟合的风险，因为它可以降低特征图的维度，从而减少参数的数量。

总之，1x1卷积在GoogleNet中具有重要的作用，它可以增加网络的非线性映射能力，降低计算复杂度，并减少过拟合的风险。

F.nll_loss和F.cross_entropy的区别

F.nll_loss和F.cross_entropy都是PyTorch中常用的损失函数，它们的主要区别在于输入数据格式和计算方式。

F.nll_loss的输入是一个包含对数概率的张量和一个包含对应标签的张量，它计算的是负对数似然损失（Negative Log Likelihood Loss）。具体来说，F.nll_loss首先将输入的对数概率通过softmax函数转换为概率值，然后取对数并取相反数，最后计算每个样本的损失，并将所有样本的损失求和。因此，F.nll_loss适用于输出层没有经过softmax函数的情况，需要手动

进行softmax操作。

`F.cross_entropy`的输入是一个包含原始输出的张量和一个包含对应标签的张量，它计算的是交叉熵损失（Cross Entropy Loss）。具体来说，`F.cross_entropy`首先对输入进行softmax操作，然后计算每个样本在所有类别上的交叉熵损失，并将所有样本的损失求和。因此，`F.cross_entropy`适用于输出层已经经过softmax函数的情况，无需手动进行softmax操作。

总之，`F.nll_loss`和`F.cross_entropy`的主要区别在于输入数据格式和计算方式。在使用时，需要根据具体的情况选择适合的损失函数。

为什么要这样做：

```
if self.stride != 1 or self.in_channels != self.out_channels:  
    identity = self.downsample(x)
```

这个操作是为了保证在残差块中，输入 x 和输出 out 的维度一致。如果输入 x 的维度和输出 out 的维度不一致，那么在将它们相加时就无法直接相加。因此，在残差块中，当输入 x 的通道数或空间尺寸发生变化时（即当 $self.in_channels \neq self.out_channels$ 或者 $self.stride \neq 1$ 时），就需要将输入 x 通过 `self.downsample` 进行一个变换，使其和输出 out 的维度一致，这个变换操作就是 $identity = self.downsample(x)$ 。

具体来说，`self.downsample` 中包含了一个卷积层和一个批归一化层，通过这个操作可以将输入 x 转化为与输出 out 维度一致的形式。这个操作通常称为“shortcut”或“identity mapping”，是残差网络的核心思想之一，可以有效地提高网络的训练速度和泛化性能。

Sigmoid函数是一个常用的激活函数，它可以将输入值映射到0和1之间的数值，它在神经网络的早期发展中扮演了重要的角色。然而，在卷积神经网络中，Sigmoid函数不再被广泛使用，原因有以下两点：

1. Sigmoid函数饱和的问题

在卷积神经网络中，由于输入数据的范围比较大，Sigmoid函数容易出现饱和的问题，即函数的导数趋近于0，这会导致梯度消失的问题，进而使网络的训练变得困难。

2. Sigmoid函数输出不是零中心的

在卷积神经网络中，使用Sigmoid函数作为激活函数会导致输出不是零

中心的，这会使得网络的训练变得困难。在深度卷积神经网络中，我们通常使用的是ReLU等激活函数，这些激活函数具有零中心性，可以有效地解决这个问题。

因此，在卷积神经网络中，Sigmoid函数不再被广泛使用，取而代之的是ReLU等激活函数，这些激活函数具有良好的性质，可以使网络的训练更加稳定和快速。

在神经网络中，零中心指的是输出的均值接近于0。当输出的均值不为0时，可能会导致梯度更新的不稳定，从而影响网络的收敛速度和性能。在卷积神经网络中，使用ReLU等激活函数可以使得输出的均值接近于0，从而使网络的训练更加稳定和快速。而sigmoid等激活函数则会使得输出不是零中心的，从而影响网络的训练效果。

在机器学习中，我们通常使用loss和accuracy来评估模型的性能。loss是用于衡量模型预测值和真实值之间的差异，而accuracy是用于衡量模型在预测测试数据时的准确率。

当一个模型的average loss更小但是accuracy比较低时，可能存在以下几种情况：

1. 模型过拟合。模型在训练数据上表现很好，但是在测试数据上表现较差。这通常是因为模型过于复杂，过分拟合了训练数据，导致在测试数据上表现不佳。

2. 数据集分布不均匀。可能测试数据集中某些类别的样本比训练数据集中的样本更难预测，导致模型在这些类别上的表现不佳。

3. 模型选择不当。有时候，一个模型可能在某些任务上表现优异，但是在其他任务上表现不佳。因此，需要根据具体的任务和数据集选择合适的模型。

4. 评估指标选择不当。有时候，一个模型的loss更小并不意味着它在测试数据上的表现更好。这是因为不同的任务和数据集可能需要使用不同的评估指标来评估模型的性能。

综上所述，当一个模型的average loss更小但是accuracy比较低时，需要进一步分析原因并针对性地调整模型或数据集，以提高模型的性能。

在卷积神经网络中，每个卷积层通常都会使用激活函数对输出进行非

线性变换，从而增加模型的表达能力。常用的激活函数包括ReLU、sigmoid、tanh等。在这些激活函数中，ReLU是最常用的一种。

使用ReLU等激活函数可以使得输出的均值接近于0的原因如下：

1. ReLU等激活函数本身是“零中心”的：在ReLU等激活函数中，当输入小于0时，输出为0，大于0时，输出等于输入。因此，当数据的均值为0时，ReLU等激活函数的输出的均值也会很接近于0。

2. 通过批量归一化进一步实现零中心化：在卷积神经网络中，通常会在每个卷积层后面添加批量归一化（Batch Normalization，简称BN）层。BN层可以通过对每个批次数据进行归一化，使得输出的均值接近于0。

因此，通过使用ReLU等激活函数，并在每个卷积层后添加BN层，可以使得卷积神经网络的输出的均值接近于0。这一特性可以提高模型的稳定性，减少过拟合的风险，从而提高模型的泛化性能。
