## CNN Explainer 试用报告

**姓名：卓威 学号：123106222860**

CNN解释器是 CNN可视化的工具，对于小白而言，CNN可视化对于理解CNN有非常的帮助，因此此篇适用报告将对CNN解释器网站做了一个翻译，在翻译过程中去了解CNN Explainer的使用方法以及CNN的工作原理。

#### 何为卷积网络？

在机器学习中，分类器将类标签分配给数据点。 例如，图像分类器为图像中存在的对象生成类标签（例如，鸟、飞机）。卷积神经网络，简称CNN，是一种分类器，擅长解决这个问题。

CNN 是一种神经网络：一种用于识别数据模式的算法。 神经网络通常由按层组织的一组神经元组成，每个神经元都有自己的可学习权重和偏差。 让我们将 CNN 分解为其基本构建块。

（1）**张量tensor**，可以被认为是一个 n 维矩阵。 在上面的 CNN 中，除输出层外，张量将是 3 维的。

（2）**神经元neuron**，可以被认为是一个接受多个输入并产生单个输出的函数。 神经元的输出在上面表示为红色→蓝色激活图。

（3）**层layer**，是具有相同操作的神经元集合，包括相同的超参数。

（4）**内核权重和偏差**Kernel weights and biases****，对于每个神经元而言，核权重和偏差是独一无二的，但在训练阶段进行了调整，并允许分类器适应所提供的问题和数据集。

（5）CNN 传达了一个可微分函数，它在输出层的可视化中表示为类分数。

CNN 使用一种特殊类型的层， 即卷积层，这使它们能够很好地从图像和类图像数据中学习。 关于图像数据，CNN 可用于许多不同的计算机视觉任务，例如 image processing, classification, segmentation, and object detection.

#### 网络的每一层都做了什么？

###### 2.1 输入层

输入层（最左边的层）代表进入 CNN 的输入图像。 因为我们使用RGB图像作为输入，所以输入层有3个通道，分别对应红、绿、蓝通道。

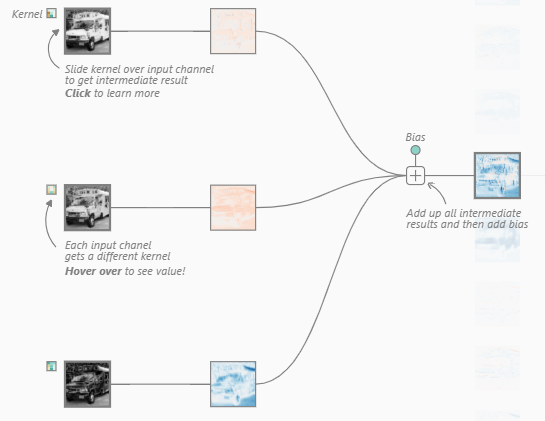
   

###### 2.2卷积层

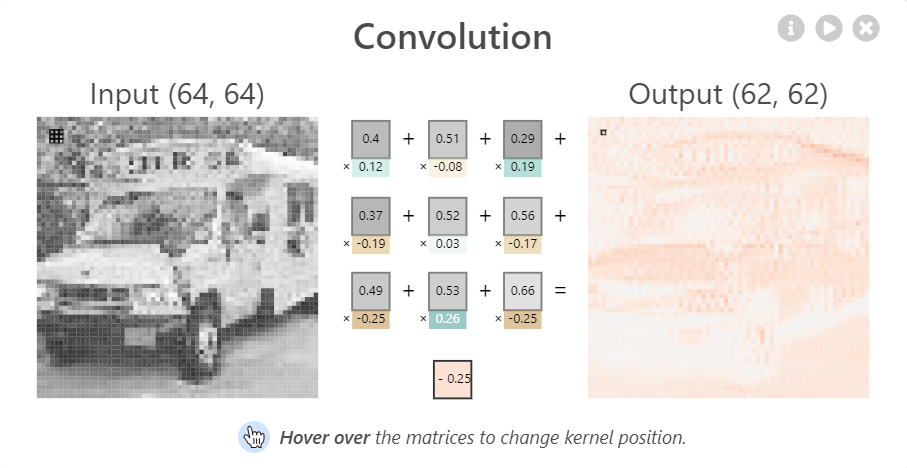
卷积层是 CNN 的基础，因为它们包含学习的内核（权重），它提取区分不同图像的特征——这就是为什么将其成为分类器的原因。当与卷积层交互时，前一层和卷积层之间相互链接。每个链接代表一个唯一的内核，用于卷积运算以产生当前卷积神经元的输出或激活图。

卷积神经元使用独特的内核和前一层相应神经元的输出执行逐元素点积(即互相关运算)。这将产生与唯一内核一样多的中间结果。卷积神经元是所有中间结果与学习偏差相加的结果。

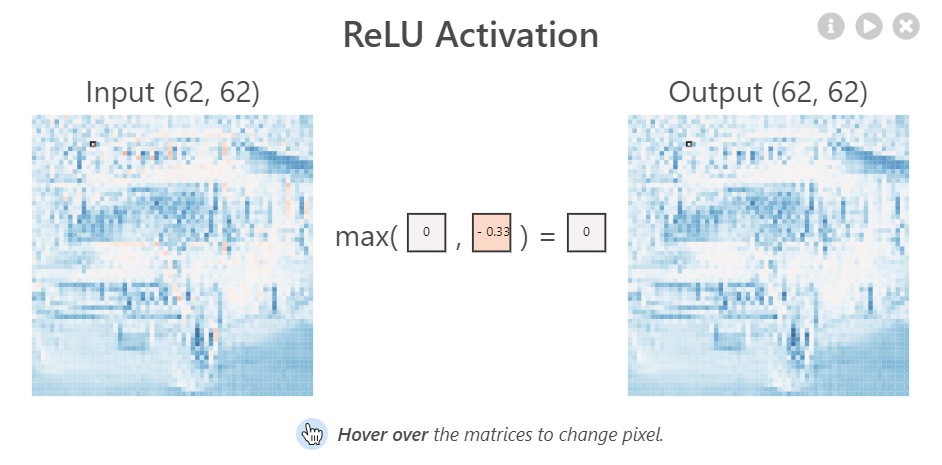
例如，让我们看一下上面 Tiny VGG 架构中的第一个卷积层。请注意，这一层有 10 个神经元，但前一层只有 3 个神经元。在 Tiny VGG 架构中，卷积层是全连接的，这意味着每个神经元都连接到前一层中的每个其他神经元。关注来自第一个卷积层的最顶层卷积神经元的输出，当我们将鼠标悬停在激活图上时，我们看到有 3 个独特的内核。



这些内核的大小是网络架构设计者指定的超参数。 为了产生卷积神经元的输出（激活图），我们必须使用前一层的输出和网络学习到的唯一内核执行元素点积。 在 TinyVGG 中，点积操作使用的步长为 1，这意味着内核每个点积移动 1 个像素以上，但这是网络架构设计者可以调整以更好地适应他们的数据集的超参数。 我们必须对所有 3 个内核执行此操作，这将产生 3 个中间结果。



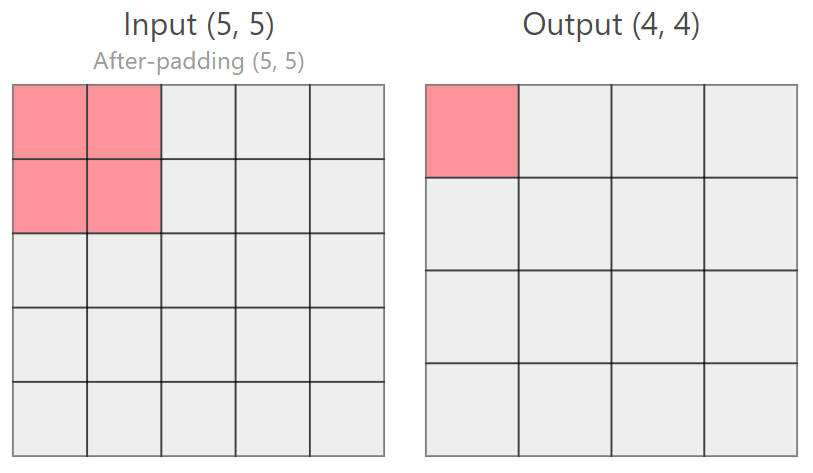
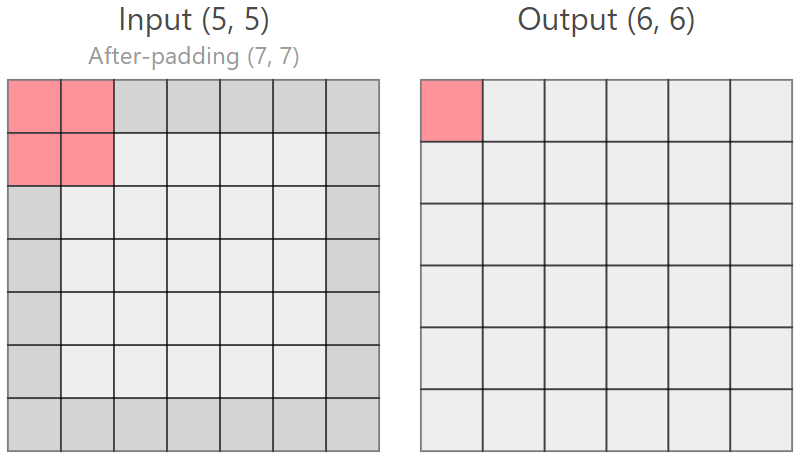
做完卷积处理之后，激活层采用RULU函数，然后，执行包含所有 3 个中间结果以及网络学习的偏差的元素求和。 在此之后，生成的二维张量将是上面界面上第一个卷积层中最顶层神经元的激活图。 必须应用相同的操作来生成每个神经元的激活图。



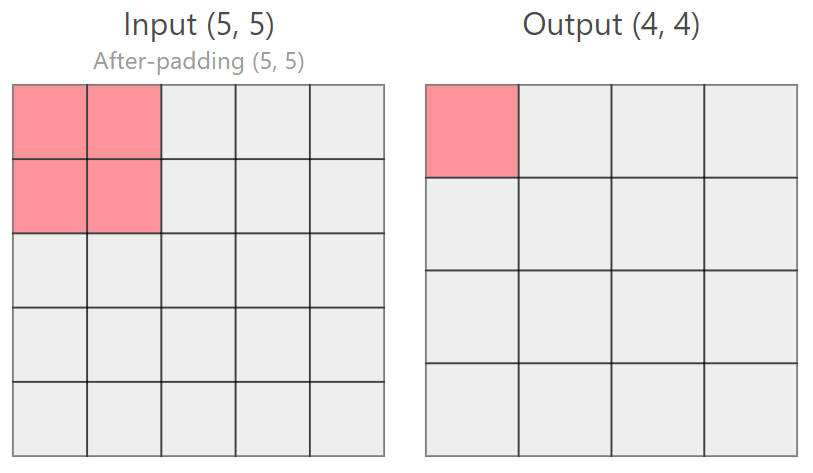
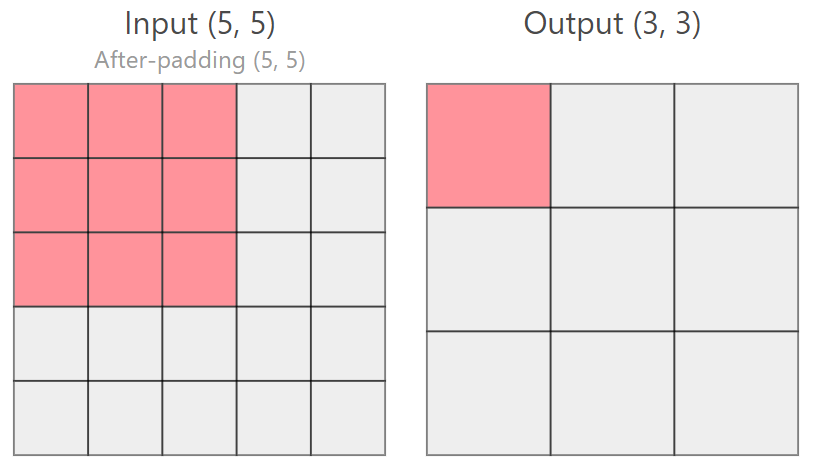
通过一些简单的数学运算，我们能够推断出有 3 x 10 = 30 个独特的内核，每个内核的大小为 3x3，应用于第一个卷积层。 卷积层和前一层之间的连通性是构建网络架构时的一个设计决策，它会影响每个卷积层的内核数。

**了解超参数：**

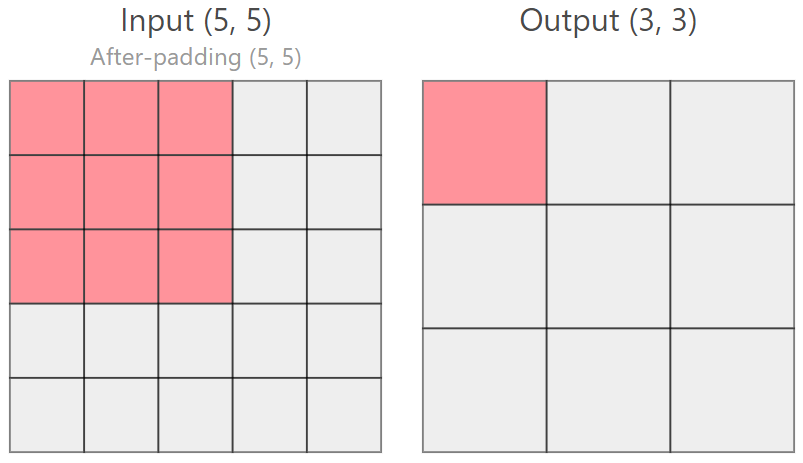
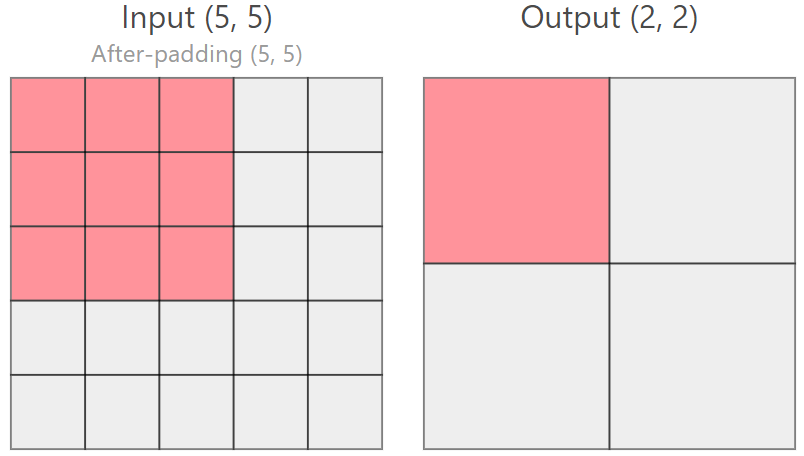
1. **填充padding，**当内核扩展到激活图之外时，通常需要填充。填充可以在激活图的边界保存数据，从而提高性能，并且可以帮助保留输入的空间大小，从而允许架构设计人员构建更简单、性能更高的网络。存在许多填充技术，但最常用的方法是零填充，因为它的性能、简单性和计算效率。该技术涉及在输入的边缘周围对称地添加零。这种方法被许多高性能 CNN 所采用，例如 AlexNet。

1. **内核大小kernel size，**通常也称为过滤器大小，是指输入上滑动窗口的尺寸。选择这个超参数对图像分类任务有巨大的影响。例如，较小的内核尺寸能够从输入中提取包含高度局部特征的大量信息。正如您在上面的可视化中所看到的，较小的内核大小也会导致层维度的减少较小，从而实现更深层次的架构。相反，较大的内核大小提取的信息较少，这会导致层维度的减少更快，通常会导致性能变差。大内核更适合提取更大的特征。归根结底，选择合适的内核大小将取决于您的任务和数据集，但通常，较小的内核大小会为图像分类任务带来更好的性能，因为架构设计师能够将越来越多的层堆叠在一起以学习越来越复杂的功能！

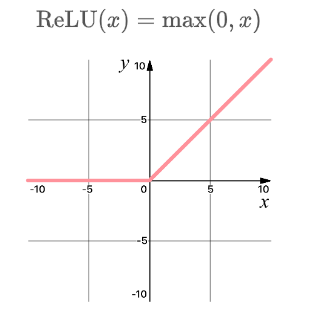
1. **步幅stride，**表示内核应该一次移动多少像素。例如，如上面卷积层示例中所述，Tiny VGG 对其卷积层使用步幅为 1，这意味着在输入的 3x3 窗口上执行内积以产生输出值，然后移至每个后续操作都正确一个像素。步幅对 CNN 的影响类似于内核大小。随着步幅的减小，因为提取了更多的数据，所以学习了更多的特征，这也导致了更大的输出层。相反，随着步幅的增加，这会导致更有限的特征提取和更小的输出层尺寸。架构设计师的职责之一是确保内核在实现 CNN 时对称地滑过输入。使用上面的超参数可视化来改变各种输入/内核维度的步幅来理解这个约束！

###### 2.3激活函数

2.3.1 RELU

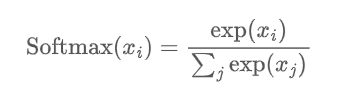
神经网络在现代技术中非常普遍——因为它们非常准确！ 当今性能最高的 CNN 由数量惊人的层组成，这些层能够学习越来越多的特征。 这些开创性的 CNN 能够实现如此巨大的准确性，部分原因在于它们的非线性。 ReLU 将急需的非线性应用到模型中。 非线性是产生非线性决策边界所必需的，因此输出不能写为输入的线性组合。 如果不存在非线性激活函数，深度 CNN 架构将演变为单个等效的卷积层，其性能几乎不会如此。 ReLU 激活函数专门用作非线性激活函数，与 Sigmoid 等其他非线性函数相反，因为根据经验观察到，使用 ReLU 的 CNN 比对应的 CNN 训练速度更快。



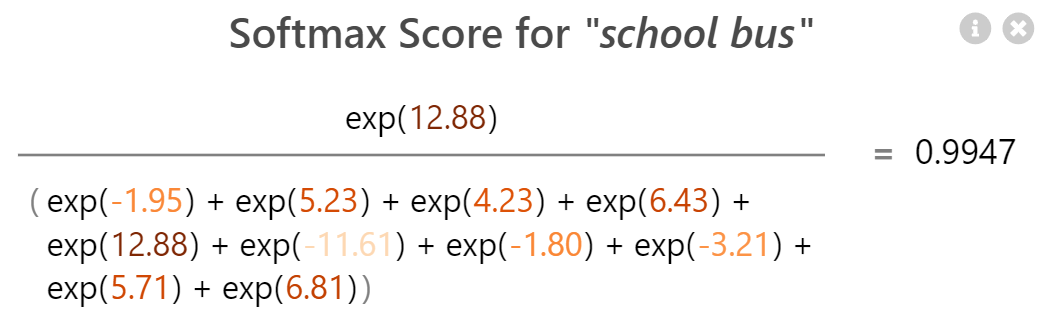
此激活函数逐元素应用于输入张量的每个值。 例如，如果对值 2.24 应用 ReLU，结果将是 2.24，因为 2.24 大于 0。

2.3.2 Softmax

softmax 操作有一个关键目的：确保 CNN 输出总和为 1。因此，softmax 操作可用于将模型输出缩放为概率，他的计算公式如下：



如下图所示，softmax计算校园巴士的得分（概率）：

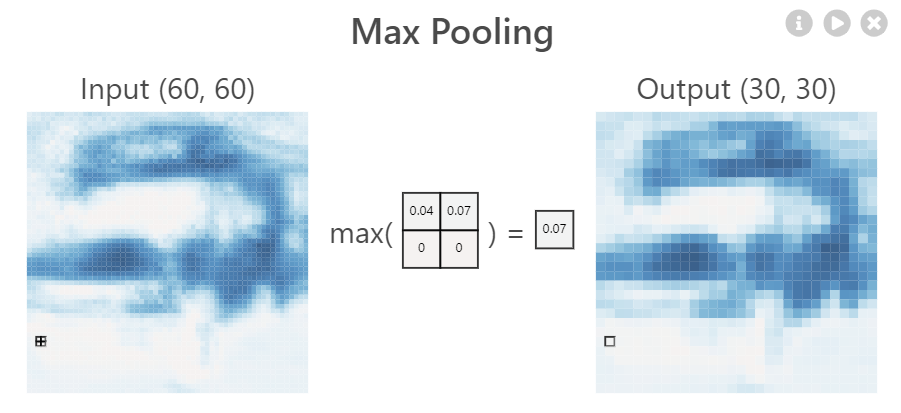


###### 2.4池化层Max Pooling

不同的 CNN 架构中的池化层类型很多，但它们的目的都是逐渐减小网络的空间范围，从而减少网络的参数和整体计算量。 上述 Tiny VGG 架构中使用的池化类型是 Max-Pooling。

Max-Pooling 操作需要在架构设计期间选择内核大小和步长。 选择后，该操作会在输入上滑动具有指定步长的内核，同时仅从输入中选择每个内核切片的最大值以生成输出值。 这个过程可以通过点击上面网络中的一个池化神经元来查看。

在上面的 Tiny VGG 架构中，池化层使用 2x2 内核和 2 步长。具有这些规范的此操作导致丢弃 75% 的激活。 通过丢弃如此多的值，Tiny VGG 的计算效率更高并避免了过度拟合。



###### 2.5展平层

该层将网络中的一个三维层转换为一维向量，以拟合全连接层的输入进行分类。 例如，一个 5x5x2 的张量将被转换为一个大小为 50 的向量。网络的先前卷积层从输入图像中提取特征，但现在是对特征进行分类的时候了。 我们使用 softmax 函数对这些特征进行分类，这需要**一维输入**，这就是为什么展平层是必要的。

