

卷积神经网络中的部分数学理论证明

李勤丰

(金陵科技学院理学院, 江苏 南京 211169)

摘要:卷积神经网络是当今深度学习领域的重要网络模型。基于最基本的三层、五层的卷积神经网络模型,在零散的相关数学理论的基础上对卷积神经网络参数调整的有效性和方法进行了较详细的理论证明,并对网络构建时的部分实现细节加以理论说明。突破了传统的基于实验结果的有效性说明,为卷积神经网络模型的理论研究奠定了一定的基础,并为提高卷积神经网络模型的效率提供了一定的参考。

关键词:卷积神经网络;数学理论;证明;参数调整;有效性

中图分类号:O29;TP301

文献标识码:A

文章编号:1672-755X(2021)02-0025-08

Some Proofs of Mathematical Theory about Convolutional Neural Network

LI Qin-feng

(Jinling Institute of Technology, Nanjing 211169, China)

Abstract: Convolutional neural network is an important network model in the field of deep learning. Based on the basic three-layer and five-layer convolutional neural network models, this paper provides a detailed theoretical proof about the validity of convolutional neural network parameter adjustment and the method of parameter adjustment, employing piecemeal mathematical theories. Part of the implementation details during construction are explained theoretically. It breaks through the traditional validity description based on experimental results, lays the theoretical foundation to a certain extent and provides a certain reference for the efficiency improvement of the convolutional neural network model.

Key words: convolutional neural network; mathematical theory; proof; parameter adjustment; validity

卷积神经网络^[1]自提出以来,许多学者对该网络模型进行了改进^[2-5],使卷积神经网络模型在图像分类、视频分类、模式识别等智能领域有了广泛的应用,成为当今最流行的深度学习模型之一。然而卷积神经网络模型的有效性都是通过实验结果的对比来说明,卷积神经网络的数学理论少之又少。2019年有文献^[6-7]开始从实验角度关注卷积神经网络模型的学习原理。由于缺乏完整的数学解释和严谨的数学证明,卷积神经网络模型的有效性受到一定的质疑,模型的发展与改进也受到制约。本文在文献^[8-9]给出的卷积神经网络模型部分理论推导的基础上,作了更详细的数学解释和证明。从最基本的三层卷积神经网络模型入手,首先证明了调整卷积神经网络的效率只需调整该模型的参数,然后用数学理论分析了模型参数调整的原理和方法,最后对参数的更新顺序、多层卷积等网络实现细节作了说明。

收稿日期:2021-04-01

基金项目:国家自然科学基金青年项目(11601204)

作者简介:李勤丰(1979—),女,江苏无锡人,副教授,硕士,主要从事智能算法网络服务研究。

1 基本卷积神经网络模型

为便于对卷积神经网络模型的图像分类有效性作详细的理论证明,这里以最基本三层卷积神经网络模型(图 1)为例。其中输入层输入图像数据,不计层数,实线矩形框内包含部分共用一层,表示该模型的核心层级。箭头方向表示数据初始传播方向。

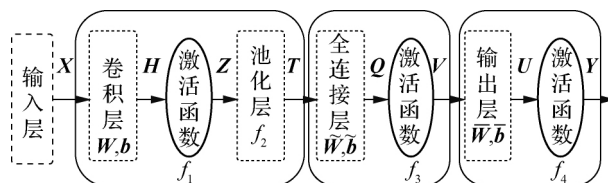


图 1 最基本的三层卷积神经网络模型示意图

为便于描述,现假设该卷积神经网络模型的输入有 100 个图像数据向量 X^n , 每个输入向量实际为 24×24 的像素矩阵,相应的类别向量记为 $y^n, n = 1, 2, \dots, 100$ 。记卷积层的输入 X 为 z^1 。卷积层有 32 个 3×3 的小卷积核, $W^k = (\omega_{ij}^k), k = 1, 2, \dots, 32$ 为其参数,滑动时步长为 1,每个卷积核有一个对应偏置 $b^k, k = 1, 2, \dots, 32$,则输入 X^n 经过卷积层后得规模为 22×22 的输出矩阵 $H^{nk} = (h_{ij}^k)_n, k = 1, 2, \dots, 32$,记为 a^1 。经过激活函数 f_1 后得池化层的输入矩阵 Z ,记为 z^2 。设池化层采用 2×2 池化核进行最大池化,于是得规模为 11×11 的输出矩阵 $T^{nk} = (t_{ij}^k)_n, k = 1, 2, \dots, 32$,记为 a^2 。 $T^{nk} (k = 1, 2, \dots, 32)$ 就是全连接层的输入,记为 z^3 。设全连接层有 2 048 个神经元,则每个神经元有 1 个偏置和 3 872 ($3\ 872 = 32 \times 11 \times 11$) 个权值,分别记为 \tilde{b}^s 和 $\tilde{W}^s = (\omega_{ij}^s), s = 1, 2, \dots, 2\ 048$,因此权值矩阵 \tilde{W} 的规模为 $2\ 048 \times 3\ 872$,可以看成块矩阵 \tilde{W}^k ,偏置向量为 \tilde{b} 。则经过运算后得到 2 048 个神经元的值 $Q^n = (q_s)_n, s = 1, 2, \dots, 2\ 048$ 。因此 Q 是 2 048 维的向量,记为 a^3 。则经过激活函数 f_3 后得到向量 V^n ,记为 z^4 ,也是输出层的输入。输出层包含 10 个神经元,每个神经元有 1 个偏置和 2 048 个权值,故权值矩阵大小是 $10 \times 2\ 048$,记为 $\bar{W} = (\bar{\omega}_{ij}),$ 相应的偏置记为 $\bar{b} = (\bar{b}^j), j = 1, 2, \dots, 10$ 。则经过输出层运算后得到一个 10 维的向量 U^n ,记为 a^4 ,再经过激活函数 f_4 后得到整个网络的输出向量 $y^n, n = 1, 2, \dots, 100$ 。的元素记为 $y_k^n, k = 1, 2, \dots, 10, y^n$ 是判断向量类别的依据。

2 卷积神经网络模型的效率调整

对任何分类而言,要使卷积神经网络模型的分效率最高,就要使模型对样本的分类准确率最高,也就是要使初始类别向量 y^n 和系统输出向量 \hat{y}^n 的差距尽量小,这个差距用系统损失函数 E 来描述。整个系统的损失函数 E 是每个输入产生的损失之和,则

$$E = \sum_{n=1}^{100} E^n = \frac{1}{2} \sum_{n=1}^{100} \|y^n - \hat{y}^n\|_2^2 = \frac{1}{2} \sum_{n=1}^{100} \sum_{k=1}^{10} (y_k^n - \hat{y}_k^n)^2 \quad (1)$$

要使目标函数 E 的值最小,如果 E 是输入 X^n 的凸函数,那么求 E 的最小值就转变为求 E 的极小值。先求 E 的驻点。由上一节可知卷积神经网络模型各层之间的函数关系为

$$\begin{aligned} y^n &= f_4(a^4) = f_4(U) = f_4(V \circ \bar{W} + \bar{b}) = f_4(z^4 \circ \bar{W} + \bar{b}) \\ V &= f_3(a^3) = f_3(Q) = f_3(T \circ \tilde{W} + \tilde{b}) = f_3(z^3 \circ \tilde{W} + \tilde{b}) \\ T &= f_2(a^2) = \text{down}(Z) = \text{down}(z^2) \\ Z &= f_1(a^1) = f_1(A) = f_1(X * W + b) = f_1(z^1 * W + b) \end{aligned} \quad (2)$$

其中 $\text{down}(Z)$ 表示对矩阵 Z 进行最大池化操作, \circ 表示矩阵的每个对应元素相乘, $*$ 表示矩阵的卷积。故 E 的驻点满足

$$\begin{aligned} \frac{\partial E}{\partial X} &= \frac{\partial E}{\partial Y} \frac{\partial Y}{\partial U} \frac{\partial U}{\partial X} = \frac{\partial E}{\partial Y} \cdot f_4'(U) \cdot (\bar{W}) \cdot \frac{\partial V}{\partial X} = \frac{\partial E}{\partial Y} \cdot f_4'(U) \cdot (\bar{W}) \cdot f_3'(Q) \cdot \frac{\partial Q}{\partial X} \\ &= \frac{\partial E}{\partial Y} \cdot f_4'(U) \cdot (\bar{W}) \cdot f_3'(Q) \cdot (\tilde{W}) \cdot \frac{\partial T}{\partial X} = \frac{\partial E}{\partial Y} \cdot f_4'(U) \cdot (\bar{W}) \cdot f_3'(Q) \cdot (\tilde{W}) \cdot \frac{\partial \text{down}(Z)}{\partial X} \end{aligned}$$

$$\begin{aligned}
&= \frac{\partial E}{\partial \mathbf{Y}} \circ f'_4(\mathbf{U}) \circ (\bar{\mathbf{W}}) \circ f'_3(\mathbf{Q}) \circ (\tilde{\mathbf{W}}) \circ \frac{\partial \text{down}(\mathbf{Z})}{\partial \mathbf{A}} \circ \frac{\partial \mathbf{A}}{\partial \mathbf{X}} \quad (3) \\
&= - \sum_{n=1}^{100} (\mathbf{y}^n - \hat{\mathbf{y}}^n) \circ f'_4(\mathbf{U}) \circ (\bar{\mathbf{W}}) \circ f'_3(\mathbf{Q}) \circ (\tilde{\mathbf{W}}) \circ \frac{\partial \text{down}(\mathbf{Z})}{\partial \mathbf{A}} \circ (\mathbf{W})
\end{aligned}$$

其中 $(\bar{\mathbf{W}})$ 表示由 \bar{W} 组成的表达式, 符号 $(\tilde{\mathbf{W}})$, (\mathbf{W}) 类似。由式(3)知 $\frac{\partial E}{\partial \mathbf{X}}$ 仅与 $\bar{W}, \tilde{W}, W, f_4, f_3, f_1, \text{down}(\cdot)$ 有关, 由式(2)知 $f_4, f_3, f_1, \text{down}(\cdot)$ (看成是一种特殊函数) 的相关结果不仅与激活函数的选择有关, 也与 $\bar{W}, \tilde{W}, W, \bar{b}, \tilde{b}, b$ 有关。因此当网络系统的超参数确定以后, 只需调整网络的偏置、权值等参数就能调整整个网络模型的效率。

3 卷积神经网络模型参数调整的方法

系统损失函数 E 达到最小值时模型输出标签和真实标签最接近, 模型的准确率最高, 为使 E 最小, 对网络分层讨论。

3.1 输出层

输出层的输出向量 $\hat{\mathbf{y}}^n$ 直接与系统损失函数 E 关联, 由于 \mathbf{y}^n 是常向量, 只需要先找到合适的参数 \bar{W}, \bar{b} , 使得输出层的输入固定时 $\hat{\mathbf{y}}^n$ 与 \mathbf{y}^n 尽量接近, 从而 E 尽量小。

由于 E 是 \bar{W} 的凸函数, 因此用梯度下降法能快速找到合适的 \bar{W}, \bar{b} 。梯度下降法的迭代公式是

$$\bar{\mathbf{W}}^{\text{new}} = \bar{\mathbf{W}}^{\text{old}} - \eta \frac{\partial E}{\partial \bar{\mathbf{W}}^{\text{old}}}, \quad \bar{\mathbf{b}}^{\text{new}} = \bar{\mathbf{b}}^{\text{old}} - \eta \frac{\partial E}{\partial \bar{\mathbf{b}}^{\text{old}}} \quad (4)$$

求 $\frac{\partial E}{\partial \bar{\mathbf{W}}}, \frac{\partial E}{\partial \bar{\mathbf{b}}}$ 。先求 $\frac{\partial E}{\partial \mathbf{b}}$:

$$\frac{\partial E}{\partial \mathbf{b}} = \frac{\partial E}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \mathbf{b}} = \frac{\partial E}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{b}} = \frac{\partial E}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \mathbf{U}} = \frac{\partial E}{\partial \mathbf{U}} \quad (5)$$

事实上由式(1)知

$$\frac{\partial E}{\partial b_1} = - \sum_{n=1}^{100} (\mathbf{y}_1^n - \hat{\mathbf{y}}_1^n) \frac{\partial \hat{\mathbf{y}}_1^n}{\partial b_1}$$

由于 \mathbf{V}^n 是一个 2 048 维的向量, 不妨设为 $(v_1, v_2, \dots, v_{2048})_n^T$, 则矩阵

$$\mathbf{U}^n = \bar{\mathbf{W}}\mathbf{V}^n + \bar{\mathbf{b}} = \left[\sum_{j=1}^{2048} (\bar{w}_{1j} v_j + \bar{b}_1) \right]_{10 \times 1} \quad (6)$$

故有

$$\begin{aligned}
\hat{\mathbf{y}}_1^n &= f_4 \left(\sum_{j=1}^{2048} \bar{w}_{1j} v_j + \bar{b}_1 \right) \\
\frac{\partial \hat{\mathbf{y}}_1^n}{\partial b_1} &= f'_4(u_1) \times 1 = f'_4(u_1) \\
\therefore \frac{\partial E}{\partial b_1} &= - \sum_{n=1}^{100} (\mathbf{y}_1^n - \hat{\mathbf{y}}_1^n) f'_4(u_1)
\end{aligned}$$

类似地, 有

$$\begin{aligned}
\frac{\partial E}{\partial b_k} &= - \sum_{n=1}^{100} (\mathbf{y}_k^n - \hat{\mathbf{y}}_k^n) f'_4(u_k) \quad k = 1, 2, \dots, 10 \\
\therefore \frac{\partial E}{\partial \mathbf{b}} &= \frac{\partial E}{\partial \mathbf{U}} = \left[- \sum_{n=1}^{100} (\mathbf{y}_i^n - \hat{\mathbf{y}}_i^n) f'_4(u_i) \right]_{10 \times 1} \triangleq \delta_4
\end{aligned} \quad (7)$$

这里定义误差对输出层的输出 \mathbf{U} 的变化率为输出层的误差灵敏度, 记为 δ_4 。同样的方法求 $\frac{\partial E}{\partial \bar{\mathbf{W}}}$:

$$\frac{\partial E}{\partial \bar{\mathbf{W}}} = \frac{\partial E}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \bar{\mathbf{W}}} = \frac{\partial E}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \bar{\mathbf{W}}}$$

由式(1)知

$$\frac{\partial E}{\partial \bar{w}_{kj}} = - \sum_{n=1}^{100} (y_k^n - \hat{y}_k^n) \frac{\partial \hat{y}_k^n}{\partial \bar{w}_{kj}} \quad j = 1, 2, \dots, 2048; k = 1, 2, \dots, 10$$

由式(6)知

$$\begin{aligned} \hat{y}_k^n &= f_4 \left(\sum_{j=1}^{2048} \bar{w}_{kj} v_j + \bar{b}_k \right) \\ \therefore \frac{\partial \hat{y}_i^n}{\partial \bar{w}_{ij}} &= f_4'(u_i)_n (v_j)_n, \quad i = 1, 2, \dots, 10; j = 1, 2, \dots, 2048 \\ \therefore \frac{\partial E}{\partial \bar{\mathbf{W}}} &= \left[- \sum_{n=1}^{100} (y_i^n - \hat{y}_i^n) f_4'(u_i)_n (v_j)_n \right]_{10 \times 2048} \triangleq \delta_4 \circ \mathbf{V}^n \end{aligned} \quad (8)$$

3.2 全连接层

全连接层的输出矩阵就是输出层的输入矩阵 \mathbf{V}^n , 而当该层的输入固定时输出矩阵的元素值只与该层的偏置向量 $\tilde{\mathbf{b}}$ 和权值矩阵 $\tilde{\mathbf{W}}$ 有关, 因此需要先找到合适的 $\tilde{\mathbf{W}}, \tilde{\mathbf{b}}$ 使全连接层的输出 \mathbf{V}^n 和该层对应于真实标签的输出 $\bar{\mathbf{V}}^n$ 尽量接近。虽然真实的 $\bar{\mathbf{V}}^n$ 未知, 但它与真实标签 \mathbf{y}^n 有关, 因此考虑用系统损失函数 E 间接表示 \mathbf{V}^n 和 $\bar{\mathbf{V}}^n$ 的差距。

寻找 $\tilde{\mathbf{W}}, \tilde{\mathbf{b}}$ 同样使用梯度下降法, 参数迭代公式与输出层类似, 不同的是该层的真实输出不可知, 需要用复合函数描述梯度下降, 迭代公式为

$$\tilde{\mathbf{W}}^{\text{new}} = \tilde{\mathbf{W}}^{\text{old}} - \alpha \frac{\partial E}{\partial \tilde{\mathbf{W}}^{\text{old}}}, \quad \tilde{\mathbf{b}}^{\text{new}} = \tilde{\mathbf{b}}^{\text{old}} - \alpha \frac{\partial E}{\partial \tilde{\mathbf{b}}^{\text{old}}} \quad (9)$$

求 $\frac{\partial E}{\partial \tilde{\mathbf{W}}}, \frac{\partial E}{\partial \tilde{\mathbf{b}}}$:

$$\frac{\partial E}{\partial \tilde{\mathbf{b}}} = \frac{\partial E}{\partial \mathbf{V}} \frac{\partial \mathbf{V}}{\partial \tilde{\mathbf{b}}} = \frac{\partial E}{\partial \mathbf{V}} \frac{\partial \mathbf{V}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \tilde{\mathbf{b}}} = \frac{\partial E}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{V}} \frac{\partial \mathbf{V}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \tilde{\mathbf{b}}} \quad (10)$$

$$\frac{\partial E}{\partial \tilde{\mathbf{W}}} = \frac{\partial E}{\partial \mathbf{V}} \frac{\partial \mathbf{V}}{\partial \tilde{\mathbf{W}}} = \frac{\partial E}{\partial \mathbf{V}} \frac{\partial \mathbf{V}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \tilde{\mathbf{W}}} = \frac{\partial E}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{V}} \frac{\partial \mathbf{V}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \tilde{\mathbf{W}}} \quad (11)$$

由式(6)知

$$\begin{aligned} \frac{\partial \mathbf{U}^n}{\partial (v_j)_n} &= \begin{bmatrix} \bar{w}_{1j} \\ \vdots \\ \bar{w}_{10,j} \end{bmatrix}_n \quad j = 1, 2, \dots, 2048 \\ \therefore \frac{\partial \mathbf{U}}{\partial \mathbf{V}} &= (\bar{w}_{ij})_{10 \times 2048} = \bar{\mathbf{W}} \end{aligned} \quad (12)$$

又

$$\begin{aligned} \mathbf{Q}^n &= (q_s)_{2048 \times 1} = \mathbf{T} \tilde{\mathbf{W}} + \tilde{\mathbf{b}} = (t_{ij}^k)_{11 \times 11} \circ (\tilde{w}_{ij})_{2048 \times 3872} + (\tilde{b}_s)_{2048 \times 1} \\ &= \left[\sum_{k=1}^{32} \sum_{i=1}^{11} \sum_{j=1}^{11} t_{ij}^k \tilde{w}_{s, j+11(i-1)+121(k-1)} + \tilde{b}_s \right]_{2048 \times 1} \\ \therefore \frac{\partial (q_1)_n}{\partial b_1} &= 1, \quad \therefore \frac{\partial \mathbf{V}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial b_1} = \frac{\partial \mathbf{V}}{\partial q_1} \frac{\partial q_1}{\partial b_1} = \frac{\partial \mathbf{V}}{\partial q_1} = f_3'(q_1) \end{aligned} \quad (13)$$

同理可得

$$\frac{\partial \mathbf{V}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial b_s} = \frac{\partial \mathbf{V}}{\partial (q_s)_n} \frac{\partial (q_s)_n}{\partial b_s} = \frac{\partial \mathbf{V}}{\partial (q_s)_n} = f_3'(q_s), \quad s = 1, 2, \dots, 2048$$

再根据式(10)得

$$\frac{\partial E}{\partial \tilde{\mathbf{b}}} = \frac{\partial E}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{V}} \frac{\partial \mathbf{V}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \tilde{\mathbf{b}}} = \delta_4 \bar{\mathbf{W}} f_3'(\mathbf{Q}) = \frac{\partial E}{\partial \mathbf{Q}} \triangleq \delta_3 \quad (14)$$

又

$$\frac{\partial Q}{\partial \tilde{w}_{s,j+11(i-1)+121(k-1)}} = (t_{ij}^k)_n$$

$$\therefore \frac{\partial V}{\partial Q} \frac{\partial Q}{\partial \tilde{w}_{s,j+11(i-1)+121(k-1)}} = \frac{\partial V}{\partial (q_s)_n} \frac{\partial (q_s)_n}{\partial \tilde{w}_{s,j+11(i-1)+121(k-1)}} = f'_3(q_s)_n (t_{ij}^k)_n$$

结合式(11)得

$$\frac{\partial E}{\partial \tilde{W}} = \frac{\partial E}{\partial U} \frac{\partial U}{\partial V} \frac{\partial V}{\partial Q} \frac{\partial Q}{\partial \tilde{W}} = \delta_4 \bar{W} [f'_3(q_s) t_{ij}^k]_{2 \ 048 \times 3 \ 872} = \delta_4 \bar{W} f'_3(Q) \circ \mathbf{T}^{nk} = \delta_3 \circ \mathbf{T}^{nk} \quad (15)$$

3.3 池化层

池化层的输出就是全连接层的输入 \mathbf{T}^{nk} 。池化方法确定时,池化层的输出只与池化层的输入矩阵 \mathbf{Z}^{nk} 有关,根据 \mathbf{T}^{nk} 就能还原 \mathbf{Z}^{nk} ,若用函数 $up(\cdot)$ 表示还原矩阵的过程,则 $\mathbf{Z}^{nk} = up(\mathbf{T}^{nk})$ 。以最大池化为例,设

$$\mathbf{Z}^{n1} = up(\mathbf{T}^{n1}) = \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & t_{11}^1 \end{bmatrix} & \dots & \begin{bmatrix} t_{1,11}^1 & 0 \\ 0 & 0 \end{bmatrix} \\ \vdots & \dots & \vdots \\ \begin{bmatrix} 0 & 0 \\ t_{11,1}^1 & 0 \end{bmatrix} & \dots & \begin{bmatrix} 0 & t_{11,11}^1 \\ 0 & 0 \end{bmatrix} \end{bmatrix}_n \quad (16)$$

其中元素 t_{ij}^1 所在的位置由函数 $down(\cdot)$ 运算时的记录而定。如果记录为左上,则位置形如 $t_{1,11}^1$,其他依此类推。由于池化层没有激活函数,为方便起见,令激活函数是 $f_2(x) = x$,误差在池化层的灵敏度 δ_2 可描述为

$$\delta_2 = \frac{\partial E}{\partial down(\mathbf{Z})} = \frac{\partial E}{\partial Q} \frac{\partial Q}{\partial \mathbf{T}} \frac{\partial \mathbf{T}}{\partial down(\mathbf{Z})}$$

又由式(13)知

$$\frac{\partial Q}{\partial (t_{ij}^k)_n} = [\tilde{w}_{s,j+11(i-1)+121(k-1)}]_{2 \ 048 \times 1}$$

$$\therefore \frac{\partial Q}{\partial \mathbf{T}} = \tilde{W}, \text{ 而 } \frac{\partial \mathbf{T}}{\partial down(\mathbf{Z})} = f'_2[down(\mathbf{Z})] = down(\mathbf{Z})$$

$$\therefore \delta_2 = \delta_3 \tilde{W} down(\mathbf{Z}) \quad (17)$$

3.4 卷积层

卷积层的输出就是池化层的输入 \mathbf{Z}^{nk} ,当卷积层的输入固定时, \mathbf{Z}^{nk} 只与该层的偏置 b_k 和权值矩阵 \mathbf{W}^k 有关。同样要找到合适的 \mathbf{W}^k, b_k 使卷积层的输出 \mathbf{Z}^{nk} 和该层真实标签对应的输出 $\bar{\mathbf{Z}}^{nk}$ 尽量接近,但真实的 $\bar{\mathbf{Z}}^{nk}$ 未知。这里依然用 E 间接表示 \mathbf{Z}^{nk} 和 $\bar{\mathbf{Z}}^{nk}$ 的差距。

寻找 \mathbf{W}^k, b_k 时依然使用梯度下降法,梯度迭代公式与全连接层、输出层类似。不同的是,该层的真实输出未知,而且该层的参数矩阵不止一个,参数矩阵数量与卷积核数量一致。迭代公式为

$$\mathbf{W}^{knew} = \mathbf{W}^{kold} - \beta \frac{\partial E}{\partial \mathbf{W}^{kold}}, \quad b_k^{new} = b_k^{old} - \beta \frac{\partial E}{\partial b_k^{old}}$$

求 $\frac{\partial E}{\partial \mathbf{W}^k}, \frac{\partial E}{\partial b_k}$:

$$\frac{\partial E}{\partial b_k} = \frac{\partial E}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial b_k} = \frac{\partial E}{\partial down(\mathbf{Z})} \frac{\partial down(\mathbf{Z})}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial b_k} \quad (18)$$

$$\frac{\partial E}{\partial \mathbf{W}^k} = \frac{\partial E}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial \mathbf{W}^k} = \frac{\partial E}{\partial down(\mathbf{Z})} \frac{\partial down(\mathbf{Z})}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial \mathbf{W}^k} \quad (19)$$

其中 $down(\mathbf{Z})$ 是对 \mathbf{Z} 中的相应位置上的 4 个元素选取最大值,即最大池化过程中没有对元素作实质性的运算,认为 $\frac{\partial down(\mathbf{Z})}{\partial \mathbf{Z}} = 1$ 。

又 $\mathbf{Z}^{nk} = f_1(\mathbf{A}^{nk}), \mathbf{A}^{nk} = \mathbf{X}^n * \mathbf{W}^k + b_k$,故

$$\mathbf{A}^{nk} = (a_q)_{22 \times 22} = \left(\sum_{i=0}^2 \sum_{j=0}^2 x_{s+i,t+j} w_{3-i,3-j}^k + b_k \right)_{22 \times 22} \quad (20)$$

$$\therefore \frac{\partial Z}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial b_1} = \frac{\partial Z}{\partial \mathbf{A}^{n1}} \frac{\partial \mathbf{A}^{n1}}{\partial b_1} = f'_1(\mathbf{A}^{n1}) \frac{\partial \mathbf{A}^{n1}}{\partial b_1}$$

又

$$\frac{\partial \mathbf{A}^{n1}}{\partial b_1} = 1, \therefore \frac{\partial Z}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial b_1} = f'_1(\mathbf{A}^{n1})$$

同理知 $\frac{\partial Z}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial b_k}, k = 1, 2, \dots, 32$, 当步长变化时 \mathbf{A}^{nk} 的规模会变化。又由式(18)知

$$\frac{\partial E}{\partial \mathbf{b}} = \delta_2 \frac{\partial \text{down}(\mathbf{Z})}{\partial \mathbf{Z}} f'_1(\mathbf{A}) = \frac{\partial E}{\partial \mathbf{A}} \triangleq \delta_1 \tag{21}$$

类似地,

$$\frac{\partial \mathbf{A}}{\partial \omega_{ij}^k} = \frac{\partial \mathbf{A}^{nk}}{\partial \omega_{ij}^k}$$

则根据式(20)有 $\frac{\partial \mathbf{A}}{\partial \mathbf{W}^1} = \left(\frac{\partial \mathbf{A}^{n1}}{\partial \omega_{ij}^1} \right)_{3 \times 3}$,

$$\therefore \text{rot}180 \left[\frac{\partial \mathbf{A}}{\partial (\mathbf{W}^1)} \right] = \left[\frac{\partial \mathbf{A}^{n1}}{\partial \omega_{(4-i),(4-j)}^1} \right]_{3 \times 3} = (\mathbf{A}_{ij})^1 \quad i, j = 1, 2, 3$$

其中 $(\mathbf{A}_{ij})^1 = \begin{bmatrix} x_{ij} & \dots & x_{i,j+21} \\ \vdots & \dots & \vdots \\ x_{i+21,j} & \dots & x_{i+21,j+21} \end{bmatrix}$, 此时

$$\delta_1 \circ \text{rot}180 \frac{\partial \mathbf{A}}{\partial \mathbf{W}^1} = \delta_1 * (\mathbf{A}_{ij})^1$$

同理 $\frac{\partial E}{\partial \mathbf{W}^k} = \frac{\partial E}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial \mathbf{W}^k} = \delta_1 * \text{rot}180(\mathbf{A}_{ij})^k$, 对该卷积层有

$$\frac{\partial E}{\partial \mathbf{W}} = \frac{\partial E}{\partial \mathbf{A}} \frac{\partial \mathbf{A}}{\partial \mathbf{W}} = \delta_1 * \text{rot}180[(\mathbf{A}_{ij})^k] \tag{22}$$

由上述分层讨论可以看出, 参数更新就是依据反向传播计算参数的相应变化。由于中间层的很多真实结果未知, 因此中间层的输出误差就用整个系统的误差间接表示, 即通过计算传递到中间层的系统误差来代替原中间层的真实误差。

4 卷积神经网络模型参数调整中的一些细节

4.1 参数更新顺序

由上一节知, 用梯度下降法进行参数更新时, 涉及的参数量大, 且参数出现在不同的层级, 此时需要考虑参数更新的顺序。以参数 \bar{W}, \bar{b} 为例, 并非利用公式(4)找到最合适的 \bar{W}, \bar{b} 后再考虑全连接层的参数 \tilde{W}, \tilde{b} 的更新, 而应该利用公式(4)更新一次 \bar{W}, \bar{b} 后马上考虑全连接层 \tilde{W}, \tilde{b} 的一次更新。因为利用公式(4)找到的最合适的 \bar{W}, \bar{b} 是针对该层的输入向量 \mathbf{V}^n 来说的, 但是输入向量 \mathbf{V}^n 与网络在该层的真实输入 $\bar{\mathbf{V}}^n$ 可能相差很大, 此时最合适的参数 \bar{W}, \bar{b} 并不是真正意义上该层最合适的参数, 调整所谓最合适的参数 \bar{W}, \bar{b} 过程中的绝大部分工作都是无用功, 甚至可能使整个更新过程变慢, 造成软硬件资源的浪费。

4.2 多层卷积的问题

上文的参数更新是基于图1的最基本的三层卷积神经网络模型。下面以五层卷积神经网络模型(图2)为例, 说明模型包含多个卷积层或多个连接层时参数的调整情况。

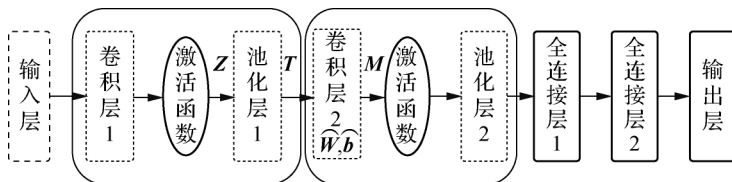


图 2 五层卷积神经网络模型示意图

从图 2 可以看出,五层卷积神经网络的正向传播过程与三层基本卷积神经网络模型完全一致,反向传播时除了池化层 1 至卷积层 2 的反向传播在上节没有讨论,其他层之间的反向传播与三层卷积模型的反向传播一致,因此这里仅讨论池化层 1 至卷积层 2 的反向传播。

假设卷积层 2 的误差灵敏度为 $\bar{\delta}$,池化层 1 的输入为 \mathbf{Z} ,它包含 32 个 11×11 的矩阵 $\mathbf{Z}^k, k = 1, 2, \dots, 32$,池化时采用尺寸为 2×2 的池化核进行最大池化,记输出矩阵为 \mathbf{T} ,则它包含 32 个 6×6 的矩阵 \mathbf{T}^k 。此时 \mathbf{T} 也是卷积层 2 的输入,假设卷积层 2 采用 64 个 2×2 的卷积核 $\widehat{\mathbf{W}}^s$,每个卷积核对应一个偏置 \widehat{b}_s ,则卷积层 2 的输出是 64 个 5×5 的矩阵 $\mathbf{M}^s, s = 1, 2, \dots, 64$,记为 \mathbf{M} 。

此时函数关系 $\mathbf{T}^k = \text{down}(\mathbf{Z}^k)$ 成立。需要注意的是,由于 11 不能被 2 整除,因此具体实现时用加边法(padding)扩充输入矩阵。

由卷积知 $\mathbf{M}^1 = \sum_{k=1}^{32} \mathbf{T}^k \star \widehat{\mathbf{W}}^1 + \widehat{b}_1$,则有 $\mathbf{M}^1 = (\mathbf{M}_{ij})_{5 \times 5}$,且

$$\mathbf{M}_{ij} = \sum_{k=1}^{32} (t_{ij}^k \widehat{w}_{22}^1 + t_{i,j+1}^k \widehat{w}_{21}^1 + t_{i+1,j}^k \widehat{w}_{12}^1 + t_{i+1,j+1}^k \widehat{w}_{11}^1) + \widehat{b}_1$$

其中 t_{ij}^k 表示 \mathbf{T}^k 中第 i 行第 j 列的元素, \widehat{w}_{ij}^s 表示 $\widehat{\mathbf{W}}^s$ 中第 i 行第 j 列的元素。

$$\therefore \frac{\partial \mathbf{M}^1}{\partial t_{ij}^k} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \widehat{w}_{11}^1 & \widehat{w}_{12}^1 & 0 \\ 0 & \widehat{w}_{21}^1 & \widehat{w}_{22}^1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad i, j = 1, 2, \dots, 6$$

其中 \widehat{w}_{22}^1 的位置在 $\frac{\partial \mathbf{M}^1}{\partial t_{ij}^k}$ 的第 i 行第 j 列。

$$\therefore \frac{\partial \mathbf{M}^1}{\partial \mathbf{T}^k} = \left(\frac{\partial \mathbf{M}^1}{\partial t_{ij}^k} \right)_{6 \times 6} \quad (23)$$

此时若有矩阵 Δ ,则

$$\begin{aligned} \Delta \circ \frac{\partial \mathbf{M}^1}{\partial \mathbf{T}^k} &= \begin{bmatrix} \delta_{11} & \delta_{12} & \delta_{13} \\ \delta_{21} & \delta_{22} & \delta_{23} \\ \delta_{31} & \delta_{32} & \delta_{33} \end{bmatrix} \circ \frac{\partial \mathbf{M}^1}{\partial \mathbf{T}^k} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \delta_{11} & \delta_{12} & \delta_{13} \\ 0 & \delta_{21} & \delta_{22} & \delta_{23} \\ 0 & \delta_{31} & \delta_{32} & \delta_{33} \end{bmatrix} \otimes \begin{bmatrix} \widehat{w}_{11}^k & \widehat{w}_{12}^k \\ \widehat{w}_{21}^k & \widehat{w}_{22}^k \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \delta_{11} & \delta_{12} & \delta_{13} \\ 0 & \delta_{21} & \delta_{22} & \delta_{23} \\ 0 & \delta_{31} & \delta_{32} & \delta_{33} \end{bmatrix} \star \text{rot}180(\widehat{\mathbf{W}}^k) \quad (24) \end{aligned}$$

其中左上角的“0”边是由上述加边法在右下角加“0”边产生的。又由于卷积层 2 的误差灵敏度为 $\bar{\delta}$,故 $\frac{\partial E}{\partial \mathbf{M}} = \bar{\delta}$,且

$$\delta = \frac{\partial E}{\partial \text{down}(\mathbf{Z})} = \frac{\partial E}{\partial \mathbf{M}} \frac{\partial \mathbf{M}}{\partial \mathbf{T}} \frac{\partial \mathbf{T}}{\partial \text{down}(\mathbf{Z})}$$

故池化层 1 的灵敏度为

$$\delta = \bar{\delta} \circ \frac{\partial \mathbf{M}}{\partial \mathbf{T}} \text{down}(\mathbf{Z}) = \begin{bmatrix} 0 & 0 \\ 0 & \bar{\delta} \end{bmatrix} \star \text{rot}180(\widehat{\mathbf{W}}) \text{down}(\mathbf{Z}) \quad (25)$$

5 结 语

本文基于最基本的三层和五层卷积神经网络模型,对模型的有效性和部分实现细节进行了理论基础证明。结果表明,调整卷积神经网络模型的效率只需进行参数调整,即要获得较高效率的网络模型,只需

找到较合适的模型参数,这与传统的通过数据实验确定模型参数的结论一致。这里利用梯度下降法对参数调整方法进行了详细说明,并对参数调整的顺序和多层卷积作了说明,为基于多层卷积的复杂网络模型的有效性证明奠定基础。由于复杂卷积神经网络模型的结构与基本卷积模型类似,只要损失函数 E 是输入 X^n 的凸函数,本文的证明就依然适用。但是复杂卷积神经网络的损失函数 E 经过多层运算后,通常不再是输入 X^n 的凸函数,此时需要利用优化方法的有关数学理论将损失函数转换成 X^n 的凸函数,而如何将非凸函数转化成凸函数有待进一步研究。

参考文献:

- [1] LECUN Y, BOSE B, DENKER J S, et al. Backpropagation applied to handwritten zip code recognition[J]. *Neural Computation*, 1989, 11(4): 541 - 551
- [2] LECUN Y, BOTTOU I, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. *Proceedings of the IEEE*, 1998, 86(11): 2278 - 2324
- [3] KRIZHEVSKY A, SUTSKEVER I, HINTON G. ImageNet classification with deep convolutional neural networks[C]. Cambridge MA: MIT Press, 2012: 1106 - 1114
- [4] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition*. Boston: IEEE, 2015: 1 - 9
- [5] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition*. Las Vegas: IEEE, 2016: 770 - 778
- [6] MCNEELY-WHITE D, BEVERIDGE J R, DRAPER B A. Inception and ResNet: same training, same features[J]. *Springer Cham*, 2019, 48: 352 - 357
- [7] MCNEELY-WHITE D, BEVERIDGE J R, DRAPER B A. Inception and ResNet features are (almost) equivalent[J]. *Cognitive Systems Research*, 2020, 59: 312 - 318
- [8] VIDAL R, BRUNA J, GIRYES R, et al. Mathematics of deep learning[J]. *Machine Learning*, 2017, arXiv: 1712. 04741
- [9] SNOEK J, LAROCHELLE H, ADAMS R P. Practical bayesian optimization of machine learning algorithms[C]. Lake Tahoe: Curran Associates Inc, 2012: 2951 - 2959

(责任编辑: 湛 江)

(上接第 11 页)处理、存储、查询、显示等功能,还实现了对各子系统故障数据的正确解析和挖掘,而且可以判定子系统与子系统之间故障的关联关系。在线路上的实际应用结果表明,引入改进算法后可以有效识别单个子系统的异常情况,并进一步挖掘出子系统间的关联故障,为轨道交通专用通信系统的运维提供支持。

参考文献:

- [1] 彭飞, 张尧. 轨道交通轴承故障诊断与寿命预测技术综述[J]. *城市轨道交通研究*, 2020, 23(12): 162 - 168
- [2] 段亚美, 施聪, 黄晓荣. 基于故障预测与健康管理技术的城市轨道交通信号系统健康管理体[J]. *城市轨道交通研究*, 2020, 23(12): 177 - 181
- [3] 周勇. 城市轨道交通智慧车站技术方案研究与实现[J]. *铁道建筑*, 2020, 60(12): 117 - 120
- [4] 刘彦军, 杨涛存, 武威, 等. 基于大数据技术的高铁运营安全规律分析系统设计与应用[J]. *中国铁路*, 2020(9): 28 - 33
- [5] 刘丙林, 朱佳, 李翔宇. 城市轨道交通车辆智能运维系统探索与研究[J]. *现代城市轨道交通*, 2019(6): 16 - 21
- [6] 白华. 城市轨道交通车辆智能运维系统研究[J]. *科学与技术*, 2020(35): 1 - 2
- [7] 杨莎, 张振山. 基于关联关系的城市轨道交通智能报警系统[J]. *城市轨道交通研究*, 2017, 20(12): 70 - 72
- [8] 刘雅娟. 基于故障诊断与健康专家系统的装备体系架构[J]. *无线电通信技术*, 2021, 47(3): 259 - 268
- [9] 张铭, 王富章, 程超. 城市轨道交通设备故障聚类与贝叶斯网络预警[J]. *计算机工程与应用*, 2016, 52(11): 259 - 264
- [10] AGRAWAL R, IMICLINSKI T, SWAMI A. Mining association rules between sets of items in large database[C]. Washington D C: Proceeding of the ACM SIG-MOD Conference on Management of Data(SIGMOD 93), 1993: 207 - 216

(责任编辑: 湛 江)