《深度学习平台与应用》作业一
20251103

一、选择题

1. 基于数据驱动（Data-Driven）实现图像分类不包括：（D）

   A. 收集图像和标签的数据集

   B. 使用机器学习算法训练分类器

   C. 使用分类器对新图像进行分类

   D. 基于专家知识手动编写所有分类规则

2. 对于一个三层神经网络，输入层输入维度4，隐藏层包含5个神经元，输出层包含2个神经元，则网络中可训练参数的总数为（请考虑所有层的权重和偏置项的数量）：(C)

   A. 30　　B. 31　　C. 37　　D. 43

3. 在深度学习模型训练中，引入正则化（Regularization）的主要作用是：(B)

   A. 增大模型容量，使其在训练集上取得更高的准确率

   B. 避免模型过拟合训练数据

   C. 加快梯度下降的收敛速度

   D. 提高模型的非线性表达能力，使其能够拟合任意函数

4. 关于模型优化下列说法正确的是：(D)

   A. 局部最优值处梯度为零并且随着网络参数量的增加，局部最优值问题的影响会越来越大

   B. 鞍点表现为Hessian矩阵全部为正或者全部为负

   C. RMSProp会使沿着“陡峭”方向的优化变快

   D. AdaGrad适合于解决凸优化问题

5. 在一个计算图中，某个节点的“上游梯度”表示：(B)

   A. 该节点对输入的影响

   B. 损失函数对该节点输出的梯度

   C. 该节点对损失函数的直接影响

D. 该节点对参数的偏导数

6. 关于二阶优化方法中的拟牛顿法（如L-BFGS），以下说法正确的是：(C)

   A. 需要显式存储完整的 Hessian 矩阵

   B. 计算复杂度为 $O(n^3)$

   C. 通过维护近似的逆 Hessian 矩阵来避免直接求逆

   D. 不适用于随机优化设置

二、填空题

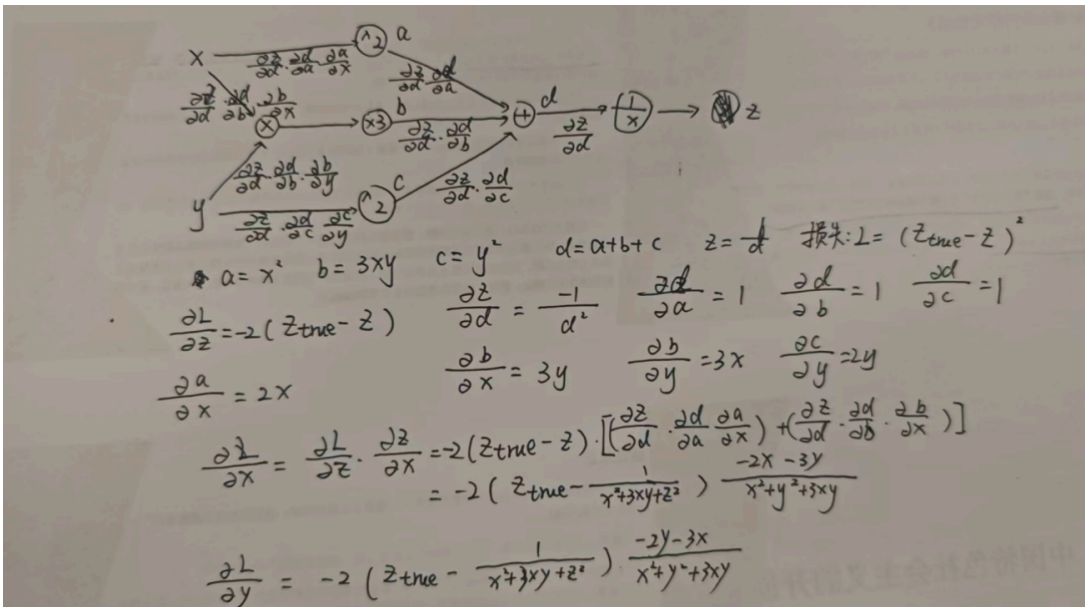
1. 使用 Cosine 学习率调度，设初始学习率 $\alpha_0 = 0.1$，总迭代次数 T = 100，在第 t = 50 次迭代时的学习率 α_{50} = _____(0.05)。
2. 对于输入维度为256， 输出维度为80的线性分类器
   $f(x, W) = Wx + b$，那么完成一次推理需要进行
   $256 \times 80 + 255 \times 80 + 80 = 40960$次运算（包括乘法和加法）
3. 给出L1距离和L2距离的公式($\sum_i |x_i - y_i|$ 和 $\sqrt{\sum_i (x_i - y_i)^2}$

三、计算题:

1. 给定函数$z = f(x, y) = \frac{1}{x^2+3xy+y^2}$

   a) 画出该函数的计算图（画出计算图时，确保标明各个节点的计算过程，并清楚标出每一步的梯度计算）

   b) 结合计算图，使用反向传播求出损失函数$L = (z_{true} - z)^2$对x和y的梯度



2. 四分类的线性分类器$f(x,w)$四个类别$class_set = [1,2,3,4]$的输出分数为$[1.1, 0.7, -0.9, 2.2]$，真实标签$class = 1$，请计算(给出计算过程)：

   a) 多类别svm loss(hinge loss): hinge loss = $\sum_{j \neq y} \max(0, s_j - s_y + 1)$ = $0.6 + 0 + 2.1 = 2.7$

   b) Softmax loss:

   $L = -\log\left(\frac{e^{s_y}}{\sum e^{s_j}}\right) = -s_y + \log(\sum e^{s_j}) = 1.57$

3. 给定一个线性分类器$f(x,w) = w^T x + b$，假设训练数据点集为$D = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$，其中$x_i \in \mathbb{R}^d$为输入特征，$y_i \in \{1, -1\}$为标签。损失函数为：

   $L(w,b) = \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i + b)) + \lambda \|w\|_2^2$

   a) 计算该损失函数关于w和b的梯度。

   $\nabla_w L = \frac{1}{n}\sum_{i=1}^{n} \left[-\mathbb{I}(1 - y_i(w^T x_i + b) > 0) \cdot y_i x_i\right] + 2\lambda w$

   $\nabla_b L = \frac{1}{n}\sum_{i=1}^{n} \left[-\mathbb{I}(1 - y_i(w^T x_i + b) > 0) \cdot y_i\right]$

   b) 讨论L2正则化的作用，并解释如何通过正则化来防止过拟合。

   - $\lambda\|w\|^2$ 项，迫使权重向量 w 的分量尽可能小且分布均匀。避免模型过于复杂，降低过拟合。

4. 给定二维训练集（点与标签）：

   - $x_1 = (1, 2),\ y_1 = A$

- $x_2 = (2, 1),\ y_2 = B$
- $x_3 = (4, 5),\ y_3 = A$
- $x_4 = (7, 8),\ y_4 = B$
- $x_5 = (1, 0),\ y_5 = A$

测试点： $x_{\text{test}} = (2,\ 2.5)$。取 $k = 3$。

a) 计算 x_{test} 到每个训练点的 L1 距离 和 L2 距离。

$$
\begin{aligned}
L_1(x, x_1) &= 1 + 0.5 = 1.5 & L_2(x, x_1) &= \sqrt{1 + (0.5)^2} = \frac{\sqrt{5}}{2} \\
L_1(x, x_2) &= 0 + 1.5 = 1.5 & L_2(x, x_2) &= \sqrt{0 + (1.5)^2} = 1.5 \\
L_1(x, x_3) &= 2 + 2.5 = 4.5 & L_2(x, x_3) &= \sqrt{2^2 + (2.5)^2} = \frac{\sqrt{41}}{2} \\
L_1(x, x_4) &= 5 + 5.5 = 10.5 & L_2(x, x_4) &= \sqrt{5^2 + (5.5)^2} = \frac{\sqrt{221}}{2} \\
L_1(x, x_5) &= 1 + 2.5 = 3.5 & L_2(x, x_5) &= \sqrt{1 + (2.5)^2} = \frac{\sqrt{29}}{2}
\end{aligned} \tag{1}
$$

b) 分别在 L1 与 L2 两种度量下，找出 3 个最近邻的标签，并给出最终预测。

L1：最近的3个点：x_1 (1.5, A), x_2 (1.5, B), x_5 (3.5, A)。投票：2票 A，1票 B。**预测：A**。

L2：最近的3个点：x_1 (1.12, A), x_2 (1.50, B), x_5 (2.69, A)。投票：2票 A，1票 B。**预测：A**。

5. 数据集大小 $N = 50000$，批量大小 $B = 128$，总训练 Epoch为 $E = 100$。初始学习率 $\alpha_0 = 0.1$。

   a) 计算每个 epoch 的迭代次数（steps/epoch），以及总迭代次数（总 steps）

   epoch = $\lceil 50000/128 \rceil = 391$

   b)按课件公式：$\alpha_t = \alpha_0 \cdot \frac{1+\cos\left(\pi\, t/T\right)}{2}$, 其中 t 为 从 0 开始 的全局迭代步数，T 为总迭代步数。计算在 $t = 0,\ t = T/2,\ t = T$ 时的学习率，并解释为何 $t = T$ 时学习率降到 ≈ 0。

$t = 0$: $\alpha = 0.1 \cdot \frac{1+1}{2} = \mathbf{0.1}$

$t = T/2$: $\cos(\pi/2) = 0 \rightarrow \alpha = 0.1 \cdot 0.5 = \mathbf{0.05}$

$t = T$: $\cos(\pi) = -1 \rightarrow \alpha = 0.1 \cdot 0 = \mathbf{0}$

学习率到0，这是为了在训练结束时让模型收敛到局部极小值的底部，不再跳动。

c) 假定加入线性 Warmup（前 5000 次迭代），Warmup 期间学习率从 0 线性上升到 α_0，之后切换到 Cosine 调度（以 T 为总训练步数继续衰减）。写出 Warmup 阶段 的学习率公式，并计算 $t = 2500$ 、$t = 5000$ 时的学习率；切换到 Cosine 后，计算 $t = 5000$ 与 $t = T$ 时的学习率（说明在 $t = 5000$ 处两段是否相等，如果要保证连续性，cosine 学习率策略该如何调整）。

Warmup: $\alpha_t = \frac{t}{5000} \cdot \alpha_0 = \frac{t}{5000} \cdot 0.1$。

- $t = 2500$: $\alpha = 0.5 \times 0.1 = 0.05$。
- $t = 5000$: $\alpha = 1.0 \times 0.1 = 0.1$。

Cosine ($t \geq 5000$):

- 要保证连续性，Cosine 应该从 warmup 结束后的步数开始计数，即定义 $t' = t - 5000$，且分母为 $T_{remain} = T - 5000$。
- **修正公式**：$\alpha_t = \alpha_0 \cdot \frac{1+\cos\left(\pi \frac{t-5000}{T-5000}\right)}{2}$。
- $t = 5000$ **处**：$\cos(0) = 1 \rightarrow \alpha = 0.1$。与 Warmup 结束时的值**相等**（连续）。
- $t = T$ **处**：$\cos(\pi) = -1 \rightarrow \alpha = \mathbf{0}$。
- **调整策略说明**：如果不调整（即直接代入全局 t 到原 Cosine 公式），在 $t = 5000$ 时 $\alpha \approx 0.1 \cdot \frac{1+\cos(5000\pi/39100)}{2} < 0.1$，会导致学习率突降。必须平移时间轴 $t \rightarrow t - T_{warmup}$ 并缩放周期。

6. 考虑同一个二分类模型的某一层参数向量，当前参数为：$w_0 = \begin{bmatrix} 0.50 \\ -0.30 \end{bmatrix}$, 在当前 mini-batch 上，损失函数对该参数的梯度为：$g_1 = \nabla_w L = \begin{bmatrix} 0.40 \\ -0.20 \end{bmatrix}$,现要求你分别用 (1) 普通 SGD、(2) 带动量的 SGD（Momentum）、(3) Adam 三种优化算法，计算“第 1 次更

新”后得到的参数，并比较它们的差异。

给定超参数如下：

- SGD：学习率：$\alpha = 0.01$
- SGD + Momentum：学习率：$\alpha = 0.01$，动量系数：$\rho = 0.9$，初始动量：$v_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- Adam：学习率：$\alpha = 0.01$, 一阶动量衰减：$\beta_1 = 0.9$, 二阶动量衰减：$\beta_2 = 0.999$, 数值稳定项：$\epsilon = 10^{-8}$, 初始一阶动量：$m_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, 初始二阶动量：$v_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

(1) 普通 SGD

$$w_1 = w_0 - \alpha g = \begin{bmatrix} 0.5 \\ -0.3 \end{bmatrix} - 0.01 \begin{bmatrix} 0.4 \\ -0.2 \end{bmatrix} = \begin{bmatrix} 0.5 - 0.004 \\ -0.3 + 0.002 \end{bmatrix} = \begin{bmatrix} 0.496 \\ -0.298 \end{bmatrix}$$

(2) SGD + Momentum

第一次迭代前 $v_0 = 0$。公式：$v_{t+1} = \rho v_t + g_t$, $w_{t+1} = w_t - \alpha v_{t+1}$。

- $v_1 = 0.9 \times 0 + g = \begin{bmatrix} 0.4 \\ -0.2 \end{bmatrix}$
- $w_1 = w_0 - 0.01 v_1 = \begin{bmatrix} 0.5 \\ -0.3 \end{bmatrix} - \begin{bmatrix} 0.004 \\ -0.002 \end{bmatrix} = \begin{bmatrix} 0.496 \\ -0.298 \end{bmatrix}$

  (注：在第1步，由于没有历史动量积累，SGD 和 Momentum 更新结果相同)

(3) Adam

$t = 1$。

- 一阶动量 m_1:

  $$m_1 = \beta_1 m_0 + (1 - \beta_1) g = 0.9(0) + 0.1 \begin{bmatrix} 0.4 \\ -0.2 \end{bmatrix} = \begin{bmatrix} 0.04 \\ -0.02 \end{bmatrix}$$

- 二阶动量 v_1 (平方累积):

  $$v_1 = \beta_2 v_0 + (1 - \beta_2) g^2 = 0.999(0) + 0.001 \begin{bmatrix} 0.4^2 \\ (-0.2)^2 \end{bmatrix} = \begin{bmatrix} 0.00016 \\ 0.00004 \end{bmatrix}$$

- 偏差修正:

  $$\hat{m}_1 = m_1 / (1 - 0.9^1) = m_1 / 0.1 = \begin{bmatrix} 0.4 \\ -0.2 \end{bmatrix}$$

  $$\hat{v}_1 = v_1 / (1 - 0.999^1) = v_1 / 0.001 = \begin{bmatrix} 0.16 \\ 0.04 \end{bmatrix}$$

- 参数更新:

$$\sqrt{\hat{v}_1} = \begin{bmatrix} 0.4 \\ 0.2 \end{bmatrix} \text{ (忽略极小 } \epsilon\text{)}$$

$$Update = \alpha \cdot \frac{\hat{m}_1}{\sqrt{\hat{v}_1}} = 0.01 \cdot \begin{bmatrix} 0.4/0.4 \\ -0.2/0.2 \end{bmatrix} = 0.01 \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$w_1 = \begin{bmatrix} 0.5 \\ -0.3 \end{bmatrix} - \begin{bmatrix} 0.01 \\ -0.01 \end{bmatrix} = \begin{bmatrix} 0.49 \\ -0.29 \end{bmatrix}$$