AI第四次作业

连炎杰，519021910706

第一题

解释机器学习模型的过拟合（overfitting）与欠拟合（underfitting）指的是什么现象。我们可以有哪些方法来避免过拟合与欠拟合的现象。

答：

对于深度学习或机器学习模型而言，我们不仅要求它对训练数据集有很好的拟合（训练误差），同时也希望它可以对未知数据集（测试集）有很好的拟合结果（泛化能力），所产生的测试误差被称为**泛化误差**。

欠拟合是指模型不能在训练集上获得足够低的误差。换句换说，就是模型复杂度低，模型在训练集上就表现很差，没法学习到数据背后的规律。

避免欠拟合的方法：

1. 增加网络复杂度，使用更复杂的模型。
2. 训练的时间长一点，train longer。

过拟合是指训练误差和测试误差之间的差距太大。换句换说，就是模型复杂度高于实际问题，**模型在训练集上表现很好，但在测试集上却表现很差**。模型对训练集"死记硬背"（记住了不适用于测试集的训练集性质或特点），没有理解数据背后的规律，**泛化能力差**。

避免过拟合的方法：

1. 正则化，如L1正则化与L2正则化。
2. 早停，即early stopping。
3. Dropout，在训练的时候随机丢掉一些神经元。
4. 使用数据增强，提高泛化能力。

第二题

根据以下的步骤完成神经网络反向传播的推导。假定一个前馈全连接神经网络的结果如下图1所示，其中，$(x_{k-1,1}, x_{k-1,2}, \cdots, x_{k-1,N_{k-1}})$ 为该神经网络第 $k-1$ 层共 $N-1$ 个神经元的输出信号，并被输入第 k 层神经元。对于第 k 层的第 j 个神经元，根据神经网络的前向信号传播规律，我们规定

$$net_{k,j} = [\sum_{i=1}^{N_{k-1}} (m_{k,j,i} \cdot x_{k-1,i}^3 + n_{k,j,i} \cdot x_{k-1,i})] + b_{k,j} \tag{1}$$

（1）假定第 k 层的激活函数是 sigmoid 函数，那么第 k 层第 j 个神经元输出的信号 $x_{k,j}$ 等于什么？（可以用 $net_{k,j}$ 表示结果）。

（2）假定我们计算该网络的输出后，得到其 loss 为 E，我们第 k 层的第 j 个神经元从网络输出处反向传播而来的梯度为

$$\frac{\partial E}{\partial x_{k,j}} = G_{k,j} \tag{2}$$

请计算对于 $net_{k,j}$ 的反向梯度 $\frac{\partial E}{\partial net_{k,j}}$。要求用 $G_{k,j}$ 与 $x_{k,j}$ 表示该反向梯度。

提示：sigmoid 函数求导公式：$\frac{\partial sigmoid(x)}{\partial x} = sigmoid(x) \cdot [1 - sigmoid(x)]$。

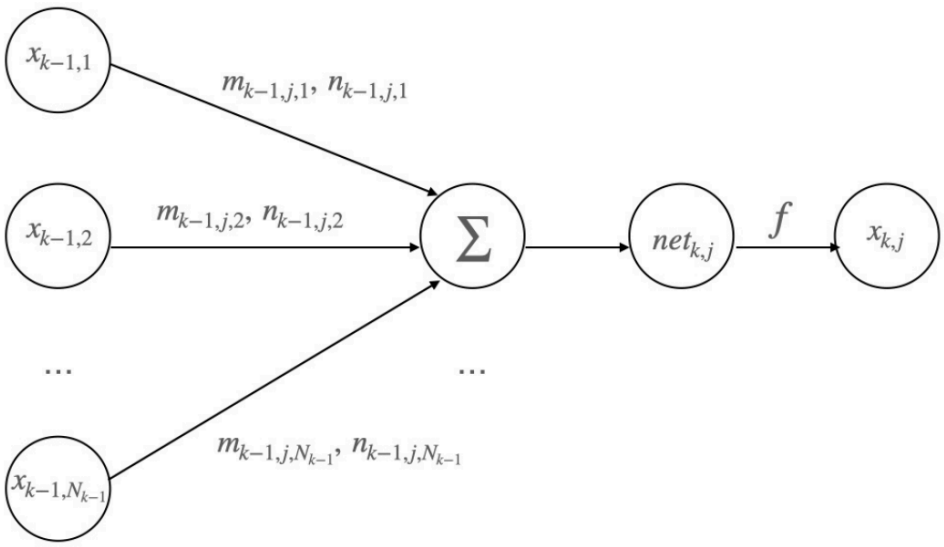

图 1: 第二题的神经网络

（3）利用上方计算好的梯度 $\frac{\partial E}{\partial net_{k,j}}$ 表达式计算参数 $u_{k,j,i}$ 处的梯度 $\frac{\partial E}{\partial m_{k,j,i}}$。要求用 $G_{k,j}$ 与 $x_{k,j}$ 表示该反向梯度。

（4）假如该神经网络的学习率为 η，那么参数 $m_{k,j,i}$ 更新后的值 $m'_{k,j,i}$ 为多少？要求用 $G_{k,j}$，$x_{k,j}$，$m_{k,j,i}$ 与 η 表示更新后的参数。

(1)

$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

$$x_{k,j} = \frac{1}{1+e^{-net_{k,j}}}$$

(2)

$\frac{\partial E}{\partial net_{k,j}} = \frac{\partial E}{\partial x_{k,j}} \frac{\partial x_{k,j}}{\partial net_{k,j}} = G_{k,j} \times \frac{1}{1+e^{-net_{k,j}}} \times [1 - \frac{1}{1+e^{-net_{k,j}}}] = \frac{G_{k,j} e^{-net_{k,j}}}{(1+e^{-net_{k,j}})^2}$（其中$net_{k,j}$比较长，就不代入了）

(3)

$$\frac{\partial E}{\partial m_{k,j,i}} = \frac{\partial E}{\partial net_{k,j}} \frac{\partial net_{k,j}}{\partial m_{k,j,i}} = \frac{G_{k,j} e^{-net_{k,j}} x_{k-1,i}^3}{(1+e^{-net_{k,j}})^2}$$

（其中$net_{k,j}$比较长，就不代入了）

(4)

如果使用SGD（stochastic gradient descent）算法，算法伪代码如下：

- Choose an initial vector of parameters w and learning rate η.
- Repeat until an approximate minimum is obtained:
  - Randomly shuffle examples in the training set.
  - For $i = 1, 2, \ldots, n$, do:
    - $w := w - \eta \nabla Q_i(w)$.

则：

$$m'_{k,j,i} = m_{k,j,k} - \eta \frac{G_{k,j} e^{-net_{k,j}} x_{k-1,i}^3}{(1+e^{-net_{k,j}})^2}$$

（其中$net_{k,j}$比较长，就不代入了）

第三题

我们在 lecture14 中学习了层次化聚类（hierarchical clustering，HAC），k-means，dbscan 密度聚类三种聚类方法，请比较一下这三种算法的时间复杂度和优缺点（假定计算两点之间的距离时间复杂度为 *O*(1)）。

算法	时间复杂度	优点	缺点
HAC	$O(n^2)$或 $O(n^3)$或 $O(n^2\log(n))$，取决于cluster之间的相似度计算，如果是常数级的那就是 $O(n^2)$。	可以构造层次结构。可以使用任意类型的distance function。	1. 不能处理噪声或outlier。2. 时间复杂度比较高。
K-Means	$O(IKn)$，其中I是iteration次数，K是cluster的个数。	1. 比较快。2. 可扩展性比较强，数据多了也可以用。	1. 随机初始化导致可能会无法收敛，或者收敛到suboptimal的结果。2. 需要给定cluster的个数K。3. 对于不是convex的数据形状效果不好。
DBSCAN	最多是$O(n^2)$，可以用一些数据结构加速到 $O(n\log(n))$。	1. 可以找出噪声点，比较鲁棒。2. 不需要指定cluster的数量。	1. 聚类结果非常取决于ϵ的大小，即半径。2. 对于density variance比较大的数据效果不好。