Author：Liu Jian

Time：2020-01-25

机器学习6-感知机与神经网络

1 感知机

感知机是神经网络与支持向量机的基础，是用于二分类问题的线性模型： $f(x) = sign(w^T x + b)$ 。

学习策略/损失函数：最小化**误分类点**到超平面的总距离。线性可分数据集 T 中误分类点的集合为 M ，则可将总距离最小化至 0 :

$$\min_{w,b} L(w,b) = -\sum_{x_i \in M} y_i (w^T x_i + b)$$

随机梯度下降法:

1. 选择初值 w_0 、 b_0 ；
2. 在训练集中随机选取数据点 (x_i, y_i) ，若 $y_i(w^T x_i + b) \leqslant 0$ ，则:

$$w \leftarrow w + \eta y_i x_i$$
$$b \leftarrow b + \eta y_i$$

其中， $0 < \eta \leqslant 1$ 为给定的学习率。迭代直至没有误分类点。

事实上，损失函数关于参数 w 和 b 的梯度如下:

$$\frac{\partial L}{\partial w} = -\sum_{x_i \in M} y_i x_i$$
$$\frac{\partial L}{\partial b} = -\sum_{x_i \in M} y_i$$

因此按照梯度下降法的标准流程，参数更新公式如下:

$$w \leftarrow w + \eta \sum_{x_i \in M} y_i x_i$$
$$b \leftarrow b + \eta \sum_{x_i \in M} y_i$$

标准梯度下降参数更新公式在一次更新中考察了所有的误分类点，而我们实际使用的更新公式在一次更新中只考察了一个误分类点，这可以看做是标准更新公式中的一步，若我们依次考察完所有的误分类点，得到的就是标准梯度下降参数更新公式了。但还有一点不同的是，实际使用的更新公式中，每次更新中误分类点的选取是随机的，因此这里的学习算法被称为**随机梯度下降算法 (stochastic gradient descent, SGD)。事实上，随机梯度下降算法每次只试图调整一个误分类点为正确分类点，也就是每次只试图最小化一个误分类点的损失函数，而不是总的损失函数。**

显然，感知机学习算法存在许多解，这些解既依赖于初值的选择，也依赖于迭代过程中误分类点的选择顺序。为了得到唯一的超平面，需要对分离超平面增加约束条件，这就是支持向量机的想法。

算法收敛性：对于线性可分数据集，感知机学习算法收敛，即经过有限次迭代可以得到一个将训练数据集完全正确划分的分离超平面。

对偶形式：感知机学习算法的对偶形式并不是像 SVM 那样基于拉格朗日函数推得，而是基于具体的优化算法即随机梯度算法的特点得到：**基于上述随机梯度下降算法**，假设样本点在参数更新过程中被使用了 n_i 次，假设初始值 w_0 和 b_0 均取 0 ，则参数最终为：

$$w = \sum_{i=1}^{N} n_i \eta y_i x_i$$

$$b = \sum_{i=1}^{N} n_i \eta y_i$$

代入感知机模型中：

$$f(x) = sign(\sum_{i=1}^{N} n_i \eta y_i \langle x_i, x \rangle + \sum_{i=1}^{N} n_i \eta y_i)$$

学习的目标就从 (w, b) 变为 n_i ，训练算法如下：

1. 初始化： $n_i = 0,\ i = 1, 2, \cdots, N$
2. 在训练集中随机选取数据点 (x_j, y_j) ，若 $y_j(\sum_{i=1}^{N} n_i \eta y_i \langle x_i, x_j \rangle + \sum_{i=1}^{N} n_i \eta y_i) \leqslant 0$ ，则：

$$n_j \leftarrow n_j + 1$$

其中， $0 < \eta \leqslant 1$ 为给定的学习率。迭代直至没有误分类点。

可以看到，此时的学习算法几乎与前面的随机梯度下降算法一模一样，只是学习的对象从 (w, b) 变为 n_i ，判断误分类点时因存在内积 $\langle x_i, x_j \rangle$ 的形式，从而可提前计算出所有内积即 Gram 矩阵，在需要用到内积结果时查询 Gram 矩阵即可。显然，基于这种内积形式可提高计算效率，并可以引入核函数构造非线性感知机。

事实上，对偶问题的标准推导过程从构造拉格朗日函数开始，接着以待定参数为变量最小化拉格朗日函数，从而得到待定参数与拉格朗日乘子之间的关系，再代入到拉格朗日函数中消去待定参数，将问题转化为以拉格朗日乘子为变量的最大化拉格朗日函数的问题。**可以看到对偶问题的标准推导过程并不涉及求解优化问题的具体算法。结合所谓感知机对偶形式的推导过程，个人认为，感知机学习算法的对偶形式并非标准意义上的对偶问题，而只是基于变量替换在对原问题进行等价变形，即将 (w, b) 替换为 n_i 将原问题等价变换为一种存在内积 $\langle x_i, x_j \rangle$ 的形式，不存在引入拉格朗日乘子构造拉个朗日函数，得到的等价问题只是在形式上与 SVM 的对偶问题相似，并没有对偶问题的内核。**

2 神经网络算法

2.1 神经元模型与感知机

功能神经元模型：

$$y = f\left(\sum_{i=1}^{n} w_i x^{(i)} - \theta\right)$$

其中，x^i 表示输入的第 i 个分量，w_i 表示神经元的第 i 个连接权重，θ 为阈值，$f(\cdot)$ 为激活函数。

常用的激活函数：

- 阶跃函数：$f(x) = sign(x)$，理想中的激活函数，但其不连续、不光滑；
- Sigmoid 函数，典型的有对数几率函数：$f(x) = \dfrac{1}{1+e^{-x}}$

可以看到，感知机=输入层(接受输入信号)+功能神经元 (处理输入信号，输出标记)，虽然由两层神经元组成，但只拥有一层功能神经元。感知机可解决线性可分问题，如“与”、“或”、“非”问题，但无法解决非线性可分问题，如“异或”问题。若要处理非线性可分问题，则需要使用多层功能神经元。多层神经网络中输入层与输出层之间的每个神经元层被称为隐层，输入层神经元仅接受输入，不进行函数处理，隐层与输出层则包含功能神经元。此外，虽然感知机的输出层只含有一个功能神经元，但**神经网络的输出层可含有多个功能神经元，对应的就是多维标记。给定神经网络结构，神经网络的学习对象就是连接权和阈值。**

2.2 典型神经网络

2.2.1 多层前馈神经网络

多层前馈神经网络 (multi-layer feedforward neural networks)：输入层+一个隐层+输出层，共三层。只要隐层包含足够多的神经元，多层前馈网络就能以任意精度逼近任意复杂度的连续函数，如何设置隐层神经元的个数实际中通常采用试错法 (trial-by-error) 调整。对于输出层，**离散属性需先进行处理：若属性间存在“序”关系则可进行连续化；否则通常转化为 k 维向量，k 为属性值数。比如，序属性“高度”的取值“高”、“中”、“矮”可转化为 $\{1.0, 0.5, 0.0\}$，无序属性“瓜类”的取值“西瓜”、“南瓜”、“黄瓜”可转化为 $(0,0,1)$、$(0,1,0)$、$(1,0,0)$。**

标准误差逆传播 (error backpropagation, BP) 算法每次试图使用梯度下降法最小化一个数据点的均方误差来进行权值和阈值的更新，类似于感知机中的随机梯度下降算法。而类似于标注梯度下降算法，神经网络的**累积误差逆传播 (accumulated error backpropagation) 算法则试图使用梯度下降法最小化累积误差，即所有样本点的误差之和。**标准 BP 算法参数更新频繁，迭代次数多；累积 BP 算法参数更新频率低，但累积误差下降到一定程度后进一步下降会非常缓慢，这时可考虑采用标准 BP 算法。

防止过拟合：(1) 早停 (early stopping)：若训练集误差降低但验证集误差升高，则停止训练；(2) 正则化 (regularization)。

2.2.2 其他神经网络

1. 径向基函数 (radial basis function, RBF) 网络：拓扑结构和多层前馈神经网络一样是单隐层前馈神经网络，只不过隐层神经元激活函数为径向基函数，输出层则是对隐层神经元输出的线性组合。同样地，具有足够多隐层神经元的 RBF 网络能以任意精度逼近任意连续函数。
2. 自适应谐振理论 (adaptive resonance theory, ART) 网络：基于竞争型学习 (competitive learning) 这样一种常用的无监督学习策略。
    - 竞争型学习基于胜者通吃 (winner-take-all) 原则。
    - ART 比较好地缓解了竞争型学习中的可塑性-稳定性窘境 (stability-plasticity dilemma, 可塑性是指神经网络要有学习新知识的能力，稳定性是指神经网络在学习新知识时要保持对旧知识的记忆)，从而能够进行增量学习 (incremental learning) 或在线学习 (online learning)。
    - 增量学习是指在学得模型后，再接收到训练样本时，仅需根据新样本对模型进行更新，不必重新训练整个模型，并且先前学得的有效信息不会被冲掉。在线学习是指每获得一个新样本点就进行一次模型更新。显然，在线学习是增量学习的特例，而增学习可视为批模式 (batch mode) 的在线学习；
    - ART 网络也是一种结构自适应神经网络。

3. 自组织映射 (self-organizing map, SOM) 网络/自组织映射 (self-organizing feature map) 网络/Kchonen 网络：是一种竞争学习型的无监督神经网络，能将高维输入数据映射到低维空间 (通常为二维)，同时保持输入数据在高维空间的拓扑结构，即将高维空间中相似的样本点映射到网络输出层中的邻近神经元。SMO 网络的训练过程类似于原型聚类算法中的 LVQ 算法。
4. 结构自适应神经网络/构造性神经网络：将网络结构也作为学习的目标之一，典型的有级联相关 (cascade-correlation) 网络。
5. 递归神经网络 (recurrent neural networks)：允许网络中出现环形结构，从而可让一些神经元的输出反馈回来作为输入信号，从而能处理与时间有关的动态变化，典型的有 Elan 网络、Boltzmann 机。
6. Boltzmann 机：一种基于能量的模型。现实中常采用受限 Boltzmann 机 (对网络结构的一种简化，类似于朴素贝叶斯)，其训练算法常用对比散度 (contrastive divergence, CD) 算法 (迭代计算最优，类似于坐标上升法)。

2.3 深度学习

提高神经网络模型容量有两种途径：(1) 增加隐层神经元的数目；(2) 增加隐层的数目。单隐层的多层前馈网络已具有很强大的学习能力；但从增加模型复杂度的角度来看，增加隐层的数目显然比增加隐层神经元的数目更有效。因为增加隐层数不仅增加了拥有激活函数的神经元数目，还增加了激活函数嵌套的层数．然而，多隐层神经网络难以直接用经典算法 (例如标准BP算法) 进行训练。因为误差在多隐层内逆传播时往往会“发散” (diverge) 而不能收敛到稳定状态。所谓的深度学习模型就是网络结构很深的神经网络。

高效的训练策略：

- 无监督逐层训练 (unsupervised layer-wise training)：逐层预训练 (pre-training, 相当于进行局部寻优) + 全网微调 (fine-tuning, 如采用 BP 算法，相当于进行全局寻优)。代表应用：深度信念网络 (deep belief network, DBN)。
- 权共享 (weight sharing)：让一组神经元使用相同的连接权。代表应用：卷积神经网络 (convolutional neural network, CNN)。

深度学习中多隐层堆叠、每层对上一层的输出进行处理的机制，可看作是在对输入信号进行逐层加工，从而把初始的、与输出目标之间联系不太密切的输入表示，转化成与输出目标联系更密切的表示，使得原来仅基于最后一层输出映射难以完成的任务成为可能。换言之，**通过多层处理，逐渐将初始的“低层”特征表示转化为“高层”特征表示后，用“简单模型”即可完成复杂的分类等学习任务。由此可将深度学习理解为进行“特征学习” (feature learning) 或“表示学习” (representation learning)**。以往在机器学习用于现实任务时，描述样本的特征通常需由人类专家来设计，这称为特征工程 (feature engineering)。但人类专家设计出好特征也并非易事，而特征学习则可通过机器学习技术自身来产生好特征。

3 附录 - 局部极小跳出策略

1. 以多组不同参数值初始化神经网，选取其中误差最小的；
2. 模拟退火 (simulated annealing) 技术：在每一步都以一定的概率接受比当前解更差的结果，从而有助于跳出局部极小，但在每步迭代过程中，接受“次优解”的概率要随着时间的推移逐渐降低，从而保证算法稳定；
3. 随机梯度下降；
4. 遗传算法。

上述方法大多是启发式的方法，理论上尚缺乏保障。
