最优化与最优控制 3

zhike chen

September 2021

1 笔记

1.1 拉格朗日乘子法解决等式约束

控制系统常采用微分方程进行建模

$$\dot{x}(t) = f(x(t), u(t), t) \quad t \in [t_0, t_f]$$

具有常微分方程约束的泛函极值问题常采用变分法与拉格朗日乘子法的结合来解决。考虑连续可微的状态变量 $x(t) : [t_0, t_f] \to \mathbf{R}^n$，在初始时刻 t_0，状态为 $x(t_0) = x_0$，在终端时刻 t_f，状态为 $x(t_f) = x_f$。同时满足约束：

$$F(x(t), \dot{x}(t), t) = 0, \quad t \in [t_0, t_f]$$

其中 F 二阶可微，并取值与 $\mathbf{R}^l$，目标为最小化性能指标泛函：

$$J(X) = \int_{t_0}^{t_f} g(x(t), \dot{x}(t), t) dt$$

求解过程引入拉格朗日乘子 $p(t) : [t_0, t_f] \to \mathbf{R}^m$，假定其连续可微。得到新的性能指标：

$$\hat{J}(x, p) = \int_{t_0}^{t_f} [g(x(t), \dot{x}(t), t) + p(t)F(x(t), \dot{x}(t), t)] dt$$

令新的代价函数包括拉格朗日算子：

$$\hat{g}(x(t), \dot{x}(t), t) = g(x(t), \dot{x}(t), t) + p(t)F(x(t), \dot{x}(t), t)$$

利用欧拉-拉格朗日方程，最小化性能指标的必要性条件是：

$$\frac{\partial \hat{g}}{\partial x_i}(x(t), \dot{x}(t), t) - \frac{d}{dt}[\frac{\partial \hat{g}}{\partial \dot{x}_i}(x(t), \dot{x}(t), t)] = 0, \quad i = 1, 2, ...n$$
$$\frac{\partial \hat{g}}{\partial p_l}(x(t), \dot{x}(t), t) - \frac{d}{dt}[\frac{\partial \hat{g}}{\partial \dot{p}_l}(x(t), \dot{x}(t), t)] = 0, \quad l = 1, 2, ...n$$

1.2 Euler-Lagrange 方程求解

Euler-Lagrange Equations:

$$\frac{\partial \hat{g}}{\partial x}(x(t), \dot{x}(t), t) - \frac{d}{dt}[\frac{\partial \hat{g}}{\partial \dot{x}}(x(t), \dot{x}(t), t)] = 0$$

1.2.1 g 不显含 $\dot{x}$

化简可得普通方程而非常微分方程:

$$\frac{\partial \hat{g}}{\partial x}(x(t), \dot{x}(t), t) = 0$$

1.2.2 g 不显含 x

化简可得:

$$\frac{d}{dt}[\frac{\partial \hat{g}}{\partial \dot{x}}(x(t), \dot{x}(t), t)] = 0$$

即:

$$\frac{\partial \hat{g}}{\partial \dot{x}}(x(t), \dot{x}(t), t) = c_1$$

1.2.3 g 不显含 t

化简可得:

$$\frac{d}{dt}[g(x(t), \dot{x}(t)) - \frac{\partial \hat{g}}{\partial \dot{x}}(x(t), \dot{x}(t))] = 0$$

2 作业

2.1 无约束优化的应用

BP 神经网络是无约束优化应用中最广的工具之一，它由输入层，隐藏层和输出层组成。在网络中，只有相邻的网络层的各个单元间存在联系，除了输出层外，每一层都有一个偏置节点如图 1 所示。

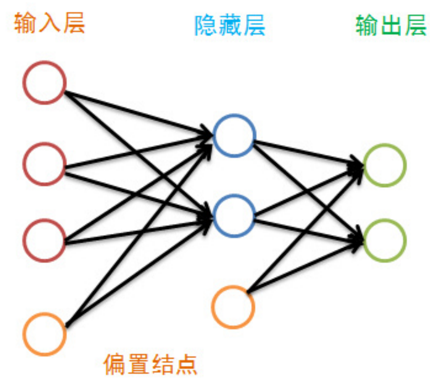

图 1: BP network

2.1.1 神经网络优化问题建模和求解

针对无约束最优化问题求解过程：将目标函数 f 泰勒展开：

$$f(x+\Delta) = f(x) + \nabla f(x)^T \Delta + \Delta^T \nabla^2 f(x) \Delta + O(\Delta^3)$$

只保留一阶项，得到递推式

$$x_{t+1} = x_t - \eta \nabla f(x_t)$$

η 为步长，$\nabla f(x_t)$ 为梯度

```
'''
利用python实现单层神经网络，包含三个输入节点和一个sigmoid隐藏层
'''

from numpy import exp, array, random, dot

Class NN():
    def __init__(self):
        self.weights = random.random((3.1))

    def sigmoid(self, x):
        return 1/(1+exp(-x));
```

```
def sigmoid_derivate(self, x):
    return x*(1-x)

def calculate(self, inputs):
    return self.sigmoid(dot(inputs, self.weights))

def train(self, inputs, output_label, iter_num):
    for iter in range(iter_nums):
        output = self.calculate(inputs)
        err = output_label - output
        delta = dot(inputs.T, err*self.sigmoid_derivate(output))

        self.weights += delta
```

2.2 带不等式约束的优化

KKT 条件：对于形如下式的一般优化问题

$$\min f(x)$$
$$s.t.\ g_j(x) \le 0\ (j = 1, 2, \cdots, n)$$
$$h_k(x) = 0\ (k = 1, 2, \cdots, m)$$

x^* 为最优解的必要条件

$$\begin{cases} \dfrac{\partial f}{\partial x_i} + \Sigma_{j=1}^m \mu_j \dfrac{\partial g_j}{\partial x_i} + \Sigma_{k=1}^m \lambda_k \dfrac{\partial h_k}{\partial x_i} = 0, (i = 1, 2, \cdots, n) \\ h_k(x) = 0,\ (k = 1, 2, \cdots, m) \\ \mu_j g_j(x) = 0,\ (j = 1, 2, \cdots, n) \end{cases} \tag{1}$$

2.3 支持向量机

支持向量机是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器。因此支持向量机可以视为一个求解凸二次规划的最优化算法。

针对一个定义在特征空间上的训练数据集 T 线性可分

$$T = (x_1, y_1), (x_2, y_2), ...(x_N, y_N)$$

其中 $x \in \mathbf{R}^n$，$y \in \{-1, 1\}$，任意超平面可以用线性方程 $w^T x + b = 0$ 来描述，数据集中某个特征点到超平面的距离为 $\frac{|w^T x + b|}{||w||}$，假设特征向量到超平面的距离为 d，其他特征点到超平面的距离大于 d，可以得到:

$$\begin{cases} \dfrac{|w^T x + b|}{||w||} \geq d \quad y = 1 \\ \dfrac{|w^T x + b|}{||w||} \leq -d \quad y = -1 \\ \dfrac{y(w^T x + b)}{||w||} \geq d \end{cases} \tag{2}$$

进而将 SVM 问题的求解简化为:

$$\min \frac{1}{2}||w||^2$$

$$s.t \quad y_i(w^T x_i + b) \geq 1$$

可以利用拉格朗日乘子法得到对偶问题：构造无约束拉格朗日目标函数

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha_i (y_i(w^T x_i + b) - 1)$$

此时原问题转化为:

$$\min_{w,b} \max_{\alpha} L(w, b, \alpha)$$

$$s.t \quad \alpha_i \geq 0$$

利用对偶性转化得到

$$\max_{\alpha} \min_{w,b} L(w, b, \alpha)$$

$$s.t \quad \alpha_i \geq 0$$

对 w、b 求偏导可得:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{N} \alpha_i x_i y_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N} \alpha_i y_i = 0$$

带入可得:

$$\min_{w,b} L(w, b, \alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i * x_j)$$

因此整个问题可以转化为:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i * x_j)$$

$$s.t. \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

进而利用二次规划的求解算法进行求解