暑期讲义

作者：有80%信心的DSY

主旨：阿巴阿巴随便写点，慢慢丰富

序言：暂无

参考书籍：《机器学习》（周志华西瓜书），《统计学习方法》（李航），《deep learning book》，《Understanding Machine Learning: From Theory to Algorithms》，《High-dimensional probability : an introduction with applications in data science》，pytorch doc，《点集拓扑讲义》

私货极多欢迎批评，邮箱：madsy@mail.scut.edu.cn

想感谢的人：暂时没想好

献给XXXX

最近前面内容好像有个主旨：本讲义想从机器学习基本问题出发，讲明白对于神经网络甚至更广义的可微模型编程建模逻辑。

1.1机器学习基础概念

1.1集合与模型

1.1.1 事件集，Definition：

真实世界里我们研究对象数值建模化的集合Ω.

eg：如各种各样的图片，声音，文本，信号。

1.1.2 输入空间，Definition：

所有可能的输入元素组成的集合$\mathcal{X}$，**事件集一定是输入空间。**

1.1.3 输出空间，Definition：

所有可能的输出元素组成的集合$\mathcal{Y}$，**事件集可以是输出空间。**

1.1.4联合分布假设:x

对于定义在$\mathcal{X}$和$\mathcal{Y}$上的随机变量X, Y，我们假设在$\mathcal{X} \times \mathcal{Y}$上其存在联合分布。

1.1.5真实映射假设:

对于定义在$\mathcal{X}$和$\mathcal{Y}$上的随机变量X, Y，我们假设存在最优的映射$y = F(x)$,$P(x, y)$可以转换为$P'(x)$.（$P(x, y) = P(x, F(x))$)）

1.1.6数据集（样本点观测集），Definition：

对应集合上定义的随机变量，观察/采样到的独立同分布的样本点集。

eg:比如对于$\mathcal{X} \times \mathcal{Y}$上的数据集$\{(x_i, y_i)\} \sim P(x, y)$

1.1.7模型，Definition：

定义在由$\mathcal{X}$到$\mathcal{Y}$的一个映射f，该模型既可以是概率模型$P(x)$，也可以是非概率模型$f(x)$

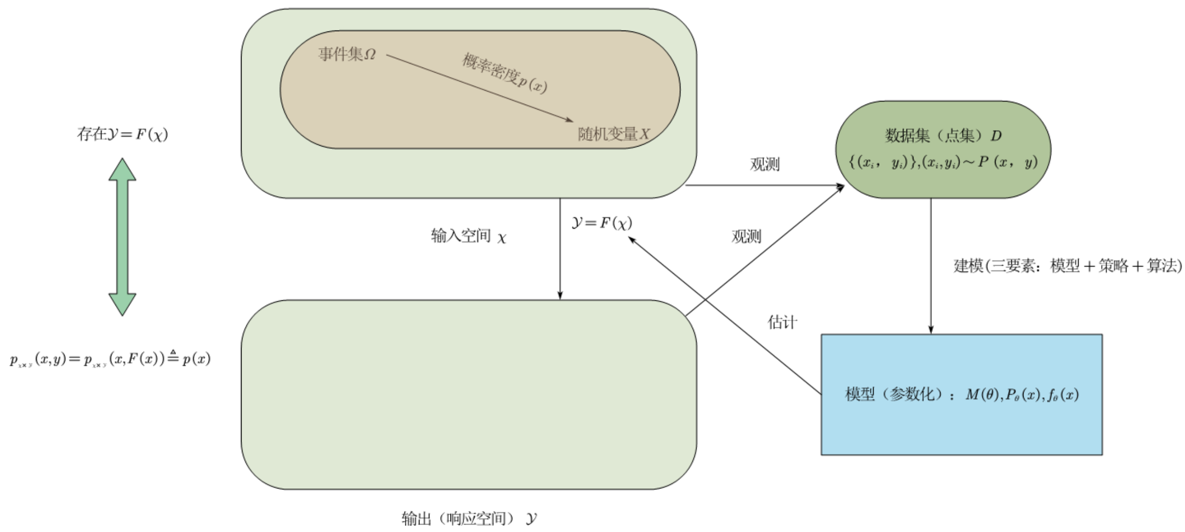
1.1.8假设空间，Definition：

对于我们假设所有可能模型f的集合$\mathcal{H}$.

1.1.9参数化模型及其假设空间，Definition：

定义在由$\mathcal{X}$到$\mathcal{Y}$的映射f由参数θ参数化，$\theta \in \Theta$,我们可以由Θ定义我们的假设空间$\mathcal{H}_\Theta$

eg:$\{f|f(x)=ax+b, a \in R, b \in R\}$定义了所有直线构成的假设空间。



1.2泛化误差及变分问题

1.2.1损失函数，Definition：

对于输出空间$\mathcal{Y}$,$\forall a,b \in \mathcal{Y}$对于二元函数$L(a,b)$,$L$满足：

$$
\begin{aligned}
&L(a,b) \geqslant 0 \\
&L(a,b)=0, \text{if and only if } a=b
\end{aligned}
$$

eg:

如果$\mathcal{Y}$是一个度量空间，则其上任意一个距离d都是损失函数。

1.2.3泛化误差，Definition：

对于一对输入输出空间$\mathcal{X}$和$\mathcal{Y}$上的随机变量X,Y，对任意模型f,我们称：

$$
\begin{aligned}
&R_{\exp}(f) \triangleq E_{P,\mathcal{X}\times\mathcal{Y}}[L(\mathcal{Y},f(X))]=\int_{\mathcal{X}\times\mathcal{Y}} L(\mathcal{Y},f(x))p(x,y)dxdy \\
&or\ R_{\exp}(f) \triangleq E_{P,\mathcal{X}}[L(F(X),f(X))]=\int_{\mathcal{X}} L(F(x),f(x))p'(x)dx
\end{aligned}
$$

What does Machine Learning do?

$$\min_{f \in \mathcal{H}} R_{exp}(f) = R(f)$$

1.2 expand:泛函

1.2.3泛函，Definition：

集合$\mathcal{H}$是一个函数集合，R是实数域，我们定义$\mathcal{H}$到R的一个映射$y = J(f)$为泛函。（本质是向量空间到标量的映射）。泛函一般由积分定义，实际上我们已经接触过很多可以定义泛函的东西。

eg:

$$J(y) = \int_{x0}^{x1} \sqrt{1 + y'^2} dx$$

为二维平面上，过点$x0, x1$的所有可微曲线的泛函，该泛函的意义是这些曲线的长度。

$$J(p) = -\int p(x) log\, p(x) dx$$

这是信息熵，衡量随机变量不确定性的泛函。

$$J(p) = KL(p||q) = \int p(x) log \frac{p(x)}{q(x)}$$

这是关于p和q的一个损失。

当然泛化误差也可以定义一种泛函：

$$J(f) = R_{\exp}(f) \triangleq E_{P, \mathcal{X} \times \mathcal{Y}}[L(\mathcal{Y}, f(X))] = \int_{\mathcal{X} \times \mathcal{Y}} L(\mathcal{Y}, f(x)) p(x, y) dx dy$$

那么机器学习和泛函有什么关系?

1.2.4泛函极值（变分问题）Definition：

我们称对于一个定义在$\mathcal{H}$上的泛函$J(f)$，问题

$$\min_{f \in \mathcal{H}} J(f)$$

称之为泛函极值问题，也可以叫做变分问题。

机器学习的数学抽象本质是变分问题，我们试图要最小化泛化误差这个泛函：

$$\min_{f \in \mathcal{H}} R_{exp}(f) = R(f)$$

对于由损失函数定义的变分问题，最优解是显然的，若真实映射$F \in \mathcal{H}$,F是最优解，不过很可惜，和传统的泛函分析面对的变分问题不同，我们无法解析的求解该变分问题。

原因有一下这么几个：

1.最重要的原因：我们无法解析知道输入空间输出空间确切的分布，和整个集合。我们所能拿到的只有数据集（样本点观测集）。

2.真实映射F是不可知的，我们的假设空间很难包含F

...

那么如何去求解该变分问题？

答案：借助概率论，实分析，测度论的工具进行估计，用统计量建立的学习策略，统计量是可以计算的，再通过参数化模型把变分问题转换成数值优化问题。

1.3PAC Theory

1.3.1经验损失，Definition：

对于可观测到的N个点集——数据集$\{(x_i, y_i)\} \sim P(x, y)$

我们定义一种常见的学习策略$\hat{R}(f)$：

$$\hat{R}(f)=\frac{1}{N} \sum_{i=1}^N L\left(F(x_i), f\left(x_i\right)\right)$$

为经验损失。

变分问题裂解

考虑：

$$R(f)=[R(f)-\hat{R}(f)]+\hat{R}(f)$$

1.3.2Generalization Gap，Definition：

我们定义如上裂解中方括号中的内容$[R(f)-\hat{R}(f)]$叫做**Generalization Gap**，此后记为**GP**。

这是个很重要的分解，可以这么说，这个看起来平常的式子，把一个变分问题拆成两部分，前面交给统计学家，后面交给优化学家。

抽象理解：

对于**GP**问题，统计学家要干的事情就是通过概率论，实分析，测度论等工具，对假设空间$\mathcal{H}$建立如下依概率不等式结论：

$$\forall \varepsilon>0, P(\exists f \in \mathcal{H}, GP \geqslant \varepsilon) \leqslant \frac{m(d_{\mathcal{H}})}{g(N) * h(\varepsilon)} \tag{1}$$

或者等价的

$$\forall \varepsilon>0, P(\forall f \in \mathcal{H}, GP \leqslant \varepsilon) \geqslant 1-\frac{m(d_{\mathcal{H}})}{g(N) * h(\varepsilon)} \tag{2}$$

其中$d_{\mathcal{H}}$的含义是假设空间的一个复杂度度量，如果$\mathcal{H}$是一个有限集合，则d可以是其元素个数，对于无限集合，我们后面会介绍他的一个复杂度度量VC Dimension ，N是观测到的数据量，ε是我们想要GP误差限，其中m，g，f都是一个多项式或者指数级别的函数。

也就是说根据我们的数据量，我们能以$1-\frac{m(d_{\mathcal{H}})}{g(N) * h(\varepsilon)}$的概率把握把**GP**控制在$\varepsilon$以内。

1.3.3 Probability Approximate Correct Learnable（PAC可学习），Definition：

若假设$\mathcal{H}$空间对于真实映射F,存在$\hat{R}(f)$使得不等式（2）成立，则我们称F对于$\mathcal{H}$是**PAC Learnable**的（PAC可学习）。

注意！：

改概念不等价于**GP**依概率收敛到0！！！！！！！！！！！！！（依测度收敛到0只是必要条件）

这里比**GP**依概率收敛到0要求更强！！！！！

PAC Learnable GP 一定依概率收敛到0

GP依据概率收敛到0不一定PAC Learnable

重要的是一定要有一个和$d_{\mathcal{H}}$,N,ε有关的概率Bound，要能**控制**。

从统计的角度理解就是$\hat{R}(f)$是$R(f)$的弱相合统计量是**PAC Learnable**的一个必要条件。

有限假设集定理

定理1.3.4： 有限假设集原理

对于输入空间$\mathcal{X}$,和布尔输出空间$\mathcal{Y}=\{0，1\}$,我们的假设空间$\mathcal{H}$有限，即$\mathcal{H}=\{f_1,f_2,\ldots\ldots,f_d\}$。

我们有：

$$\forall\varepsilon>0,P(\forall f\in\mathcal{H},GP\leqslant\varepsilon)\geqslant1-\frac{d}{exp\{2N\varepsilon^2\}}$$

即有限假设空间对于任意布尔映射是**PAC Learnable**。

该证明需要用到**Hoeffding inequality**：

定理1.3.4.1 Hoeffding inequality（霍夫丁不等式）

$X_1,\cdots,X_n$ 为有界独立随机变量， 即$A_i\leq X_i\leq B_i$,$1\leq i\leq n$. 对于任意$\varepsilon>0$, 和任意$t>0$,

$$P\left(\sum_{i=1}^{n}\left(X_i-E\left[X_i\right]\right)\geq\varepsilon\right)\leq\exp\left(-t\varepsilon+\frac{t^2}{8}\sum_{i=1}^{n}\left(B_i-A_i\right)^2\right)$$

进一步：

$$P\left(\sum_{i=1}^{n}\left(X_i-E\left[X_i\right]\right)\geq\varepsilon\right)\leq\exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n}\left(B_i-A_i\right)^2}\right)$$

而要证明**Hoeffding inequality**我们先要需要证明高维统计如下几个引理：

引理1.3.4a：Markov inequality（马尔可夫不等式）

$$X\geqslant0\,,\forall\varepsilon>0\,,P(x\geqslant\varepsilon)\leqslant\frac{1}{\varepsilon}EX.\quad .$$

proof:

$$EX=\int_0^{+\infty}xdF(x)\geqslant\int_\varepsilon^{+\infty}xdF(x)\geqslant\varepsilon\int_\varepsilon^{+\infty}dF(x)=\varepsilon P(x\geqslant\varepsilon).$$
$$\therefore P(x\geqslant\varepsilon)\leqslant\frac{1}{\varepsilon}EX.\quad .$$

定理1.3.4.1 Hoeffding inequality

proof:

我们考虑：

$$Y_i\triangleq X_i-EX_i\quad,EY_i=0$$

可以看出：

$$a_i=A_i-EX_i\leqslant Y_i\leqslant b_i=B_i-EX_i$$

原问题等价：

$$P\left(\Sigma Y_i\geqslant\varepsilon\right)\leqslant\exp\left(-t\varepsilon+\frac{t^2}{8}\Sigma(b_i-a_i)^2\right)$$

考虑：

$$P\left(\Sigma Y_i\geqslant\varepsilon\right)=P\left(e^{t\Sigma Y_i}\geqslant e^{t\varepsilon}\right)$$

由**Markov inequality**：

$$P\left(e^{t\Sigma Y_i}\geqslant e^{t\varepsilon}\right)\leqslant e^{-t\varepsilon}\cdot E\left[e^{t\Sigma Y_i}\right]$$

因为Y_i独立：

$$P\left(e^{t\Sigma Y_i}\geqslant e^{t\varepsilon}\right)\leqslant e^{-t\varepsilon}\cdot\mathit{\Pi}_i^N E\left[e^{tY_i}\right]$$

又因为$a_i\leqslant Y_i\leqslant b_i$,

$$Y_i=(1-\alpha)a_i+\alpha bi，\alpha=\frac{Y_i-a_i}{b_i-a_i}\in[0.1]$$

因为e^t是凸的：

$$E\left[e^{tY_i}\right]\leqslant E[\frac{b_i-Y_i}{b_i-a_i}e^{ta_i}+\frac{Y_i-a_i}{b_i-a_i}e^{tb_i}]$$
$$E\left[e^{tY_i}\right]\leqslant\frac{b_i}{b_i-a_i}e^{ta_i}-\frac{a_i}{b_i-a_i}e^{tb_i}.$$

令$u=t\left(b_i-a_i\right)>0,\quad\gamma=-\frac{a_i}{b_i-a_i}\Rightarrow\frac{b_i}{b_i-a_i}=1-\gamma$，得到：

$$-\frac{a_i}{b_i-a_i}e^{ta_i}+\frac{b_i}{b_i-a_i}e^{tb_i}=\gamma e^{(1-\gamma)u}+(1-\gamma)e^{-\gamma u}=e^{-\gamma u}\left(\gamma e^u+1-\gamma\right)\qquad(3)$$

令$g(u)\triangleq\log(3)=-\gamma u+\log\left(\gamma e^u+1-\gamma\right)$,考虑0点泰勒展开：

$$g(0)=0.\quad g'(u)=-\gamma+\frac{\gamma e^u}{\gamma e^u+1-\gamma}.\quad g'(0)=0$$

$$g''(u)=\frac{\gamma e^u\left(\gamma e^u+1-\gamma\right)-\gamma e^\mu\left(\gamma e^u\right)}{(\gamma e^u+1-\gamma)^2}=\frac{(1-\gamma)\gamma e^u}{(\gamma e^u+1-\gamma)^2}\leqslant\frac{(1-\gamma)\gamma e^u}{\left(2\sqrt{\gamma e^u(1-\gamma)}\right)^2}=\frac{1}{4}$$

得到：

$$g(u)=g(0)+ug'(0)+\frac{u^2}{2}g''(\xi) \quad 0\leqslant\xi\leqslant u$$

得到：

$$g(u)=\frac{u^2}{2}g''(\xi)\leqslant\frac{u^2}{8}.$$

得到

$$E\left[e^{tY_i}\right]\leqslant e^{g(u)}\leqslant e^{\frac{1}{8}u^2}=e^{\frac{t^2}{8}(b_i-a_i)^2}$$

得到：

$$P\left(\sum Y_i\geqslant\varepsilon\right)\leqslant\exp\left(-t\varepsilon+\frac{t^2}{8}\sum(b_i-a_i)^2\right)$$
$$P\left(\sum Y_i\geqslant\varepsilon\right)\leqslant\exp\left(-\frac{2\varepsilon^2}{\sum(b_i-a_i)^2}\right)$$

得证。

Expend PS：

实际上**Hoeffding**只是更抽象不等式的一个特例，实际上指数型上界的概率不等式不止对有界独立变量成立，对于许多无界的分布也都成立，例如高斯分布，在高维统计中我们有专门研究一类具有可控概率界中心矩的分布（次高斯分布，次指数分布），**Bernstein**不等式等等就是用来描述这类过程的。这类不等式叫做**Concentration inequality**（集中不等式），描述的是样本数量朝着期望的收敛的一类概率bound，这类不等式原理在机器学习的learning theory和很多前沿随机过程问题中非常常见。

定理1.3.4 有限假设集原理

proof：

对任意函数 $f\in\mathcal{H},\hat{R}(f)$ 是 N 个独立的随机变量 $L(Y,f(X))+\varepsilon$的样本均值, $R(f)$是随机变量 $L(Y,f(X))$ 的期望，由**Hoeffding inequality**：

$$P(R(f)-\hat{R}(f)\geqslant\varepsilon)\leqslant\exp\left(-2N\varepsilon^2\right)$$

又因为$\mathcal{H}=\{f_1,f_2,\ldots\ldots,f_d\}$有限：

$$\begin{aligned}P(\exists f\in\mathcal{H}:R(f)-\hat{R}(f)\geqslant\varepsilon)&=P\left(\bigcup_{f\in\mathcal{H}}\{R(f)-\hat{R}(f)\geqslant\varepsilon\}\right)\\&\leqslant\sum_{f\in\mathcal{H}}P(R(f)-\hat{R}(f)\geqslant\varepsilon)\\&\leqslant d\exp\left(-2N\varepsilon^2\right)\end{aligned}$$

1.4VC Dimension（Vapnik-Chervonenkis Dimension）

1.4.3 Empirical processes via VC dimension:

$$E\sup_{f\in\mathcal{H}}\left|\frac{1}{n}\sum_{i=1}^{n}f\left(X_i\right)-Ef(X)\right|\leq C\sqrt{\frac{\operatorname{vc}(\mathcal{H})_0}{n}}$$

$$R(f)-\hat{R}(f) \leq E \sup_{f \in \mathcal{H}}|R(f)-\hat{R}(f)|=E \sup_{f \in \mathcal{H}}\left|\frac{1}{n} \sum_{i}\left(f\left(X_{i}\right)-\mathbb{E} f\left(X_{i}\right)\right)\right| \leq C \sqrt{\frac{vc(\mathcal{H})}{n}}$$

1.5Uniform Convergence

1.6变分问题及参数优化问题

参数优化问题

1.6.1参数化的假设空间,Definition：

我们这里讨论研究的是连续的参数空间Θ,不讨论离散的Θ。

f是由参数θ定义，$\theta \in \Theta$,参数化假设空间：

$$\mathcal{H}=\{f_\theta(x)|\theta \in \Theta\}$$

并定义参数化过程$p: \Theta \rightarrow \mathcal{H}$ 。

1.6.2经验损失变分问题,Definition：

对于可观测到的N个点集——数据集$\{(x_i,y_i)\} \sim P(x,y)$,参数化假设空间$\mathcal{H}=\{f_\theta(x)|\theta \in \Theta\}$,某种建模策略的经验损失$\hat{R}(f)$，定义:

$$\min_{f \in \mathcal{H}} \hat{R}(f)=\frac{1}{N} \sum_{i=1}^{N} L\left(F(x_i), f\left(x_i\right)\right)$$

1.6.3模型参数优化问题,Definition：

对于可观测到的N个点集——数据集$\{(x_i,y_i)\} \sim P(x,y)$,参数化假设空间$\mathcal{H}=\{f_\theta(x)|\theta \in \Theta\}$,某种建模策略的经验损失$\hat{R}(f)$，定义:

$$\min_{\theta \in \Theta} L(\theta)=\hat{R}(f_\theta)=\frac{1}{N} \sum_{i=1}^{N} L\left(F(x_i), f_\theta\left(x_i\right)\right)$$

为参数优化问题。

注意：在我们1.6的讨论场景中$\hat{R}(f)$仍然是一个泛函，而**问题1.6.2**和**问题1.6.3**并不能想当然的认为是等价的，两者等价需要满足一定的充要条件。最为重要的环节在于研究$\mathcal{H}$和Θ之间的关系，要研究这两种之间的关系，我们需要借助一点拓扑工具。

1.6 expand：点集拓扑与同胚

(此处只是稍微展开背后的思想，具体弄明白同胚的概念要学完整本点集拓扑)

1.6.4拓扑，Definition：

令X为一个集合，$\mathcal{T}$是X的一个子集族（族的概念就是集合的集合），如果$\mathcal{T}$满足如下条件：

(1) X，$\varnothing \in \mathcal{T}$

(2)$\forall A, B \in \mathcal{T}$,则$A \cap B \in \mathcal{T}$

(3)若$\mathcal{T}_1 \subset \mathcal{T}$,则$\cup_{A \in \mathcal{T}_1} A \in \mathcal{T}$

则称$\mathcal{T}$是X的一个拓扑。

且称集合X是一个相对于拓扑$\mathcal{T}$而言的拓扑空间，记为偶对$(X, \mathcal{T})$

1.6.5同胚，Definition：

使 $(X, \mathcal{T}_X)$ 和 $(Y, \mathcal{T}_Y)$ 为两个拓扑空间，若存在使 $f:(X, \mathcal{T}_X) \rightarrow (Y, \mathcal{T}_Y)$ 。 f 被称为这两个空间之间的同胚映射当且仅当

(1)f 是一个双射;

(2)f 是连续的;

(3)f^{-1} 是连续的。

我们也称$(X, \mathcal{T}_X)$和$(Y, \mathcal{T}_Y)$是同胚的。

假设空间及参数空间的同胚定理

1.6.6参数冗余，Definition：

我们称参数化过程p是冗余的，当且仅当对于$\forall x \in \mathcal{X}$,$\exists \theta_1, \theta_2 \in \Theta, \theta_1 \neq \theta_2, f_{\theta_1}(x) = f_{\theta_2}(x)$。

eg:$\{f|f(x) = ax + b + c, a \in R, b \in R, c \in R\}$是冗余的。

定理1.6.7 假设空间及参数空间的同胚定理

问题1.6.2和**问题1.6.3**等价：

当且仅当$\mathcal{H}$和Θ同胚，且p为其一个同胚映射。

或者说当且仅当参数化过程p不是冗余的。

proof：

因为$\mathcal{H}$和Θ同胚，且p为其一个同胚映射，所以p是一个双射，所以对于**问题1.6.2**任意局优f^*，和收敛于**问题1.6.2**任意局优的柯西列$\{f_i\}$，都在Θ中存在

θ^*及柯西列$\{\theta_i\}$唯一对应。反过来对**问题1.6.3**同理，因此我们自然得到了等价性的证明。

而由参数化过程p的构造过程，我们知道p一定是Θ到$\mathcal{H}$的连续满射，而参数冗余本质是在说明p还是Θ到$\mathcal{H}$的单射，所以我们可以得到p是Θ到$\mathcal{H}$的同胚映射。

PS:

实际上此处可以更多理解一下泛函这个概念，泛函的定义不是狭隘的函数到标量（实数域）的映射，**他本质是向量空间到实数域的映射**，只不过我们可以把函数当成一种向量。这里的同胚，**本质上是两个向量空间之间的同胚，是最简单的同胚。**

eg：如果我们有n个参数，即$\theta = (\theta_1, \theta_2, \ldots\ldots, \theta_n)$,且每个参数的定义域都是全体实数，即$\Theta = R^n$,那么我们的假设空间$\mathcal{H}$就是同胚于$R^n$的，如此我们在假设空间$\mathcal{H}$上的任意变分问题，**等价于在$R^n$上针对参数$\theta$的无约束数值优化问题。**若$\Theta$不是$R^n$,类似的也可以证明**假设空间$\mathcal{H}$同胚于$R^n$的某个子空间，转换为约束数值优化问题。**

2.1Pytorch基础概念

请大家必备网站：https://pytorch.org/docs/stable/index.html

2.1.1基础数据：

```
torch.Tensor
```

tensor是包含单一数据类型元素的多维矩阵。

Torch 定义了 10 种具有 CPU 和 GPU 变体的张量类型：

数据类型	类型	CPU张量	GPU张量
32 位浮点	`torch.float32` 或者 `torch.float`	`torch.FloatTensor`	`torch.cuda.FloatTensor`
64 位浮点	`torch.float64` 或者 `torch.double`	`torch.DoubleTensor`	`torch.cuda.DoubleTensor`
16 位浮点数[1]	`torch.float16` 或者 `torch.half`	`torch.HalfTensor`	`torch.cuda.HalfTensor`
16 位浮点数[2]	`torch.bfloat16`	`torch.BFloat16Tensor`	`torch.cuda.BFloat16Tensor`
32 位复数	`torch.complex32`		
64 位复数	`torch.complex64`		
128 位复数	`torch.complex128` 或者 `torch.cdouble`		
128 位复数	`torch.complex128` 或者 `torch.cdouble`		
8 位整数（无符号）	`torch.uint8`	`torch.ByteTensor`	`torch.cuda.ByteTensor`
8 位整数（有符号）	`torch.int8`	`torch.CharTensor`	`torch.cuda.CharTensor`
16 位整数（有符号）	`torch.int16` 或者 `torch.short`	`torch.ShortTensor`	`torch.cuda.ShortTensor`
32 位整数（有符号）	`torch.int32` 或者 `torch.int`	`torch.IntTensor`	`torch.cuda.IntTensor`
64 位整数（有符号）	`torch.int64` 或者 `torch.long`	`torch.LongTensor`	`torch.cuda.LongTensor`
布尔值	`torch.bool`	`torch.BoolTensor`	`torch.cuda.BoolTensor`