高级机器学习
作业二

MG1833074, 王易木, wangym@lamda.nju.edu.cn

2018 年 12 月 23 日

1 [30pts] Learning Theory

(1) **[10pts] VC 维**

试讨论最近邻分类器假设空间的 VC 维大小，并给出证明.

(2) **[10pts] Rademaher 复杂度**

试证明: 常数函数 c 的 Rademaher 复杂度为 0.

(3) **[10pts] PAC**

$\mathcal{X} = \mathbb{R}^2, \mathcal{Y} = 0, 1$. 假设空间 $\mathcal{H}$ 定义如下: $\mathcal{H} = \{h_r : r \in \mathbb{R}_+\}$, 其中 $h_r(x) = \mathbb{I}(\| x \| \leq r)$, 假定假设空间是可分的，证明 $\mathcal{H}$ 是 PAC 可学习的，并且样本复杂度为 $\frac{log(1/\delta)}{\epsilon}$
(提示: 可考虑返回与训练集一致的最小圆的算法)

Proof. 此处用于写证明 *(中英文均可)*

(1) ***[10pts] VC 维***

很明显 *1-NN* 对于任意数量的集合，都可以将其打散（因为不管集合有多大，总能找到一个离目标点最近的训练集中的点，并对其分类)。

$$H(x) = argmin_{y \in D_{train}} ||x - y||_2 \tag{1.1}$$

故 *1-NN* 最近邻分类器的假设空间的 *VC* 维大小为无限。

(2) ***[10pts] Rademaher 复杂度***

首先，函数空间 $\mathcal{F}$ 关于训练样本 $\mathcal{Z}$ 的经验 *Rademaher* 复杂度为:

$$\widehat{R}_Z(\mathcal{F}) = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f\left(\boldsymbol{z}_i\right) \right] \tag{1.2}$$

因为 $|\mathcal{F}| = 1$，即函数空间只有一个假设，常数函数。

经验 *Rademaher* 复杂度为

$$\widehat{R}_Z(\mathcal{F}) = 0 \tag{1.3}$$

函数空间 $\mathcal{F}$ 关于训练样本 $\mathcal{Z}$ 上分布 *Rademaher* 复杂度为:

$$\begin{aligned} R_m(\mathcal{F}) &= \mathbb{E}_{Z\subseteq\mathcal{Z}:|Z|=m}\left[\widehat{R}_Z(\mathcal{F})\right] \\ &= \mathbb{E}_{Z\subseteq\mathcal{Z}:|Z|=m}\left[0\right] \\ &= 0 \end{aligned} \tag{1.4}$$

(3) ***[10pts] PAC***

假设 h 的泛化误差大于 ϵ，对分布 $\mathcal{D}$ 上随机采样而得到的任何样例 $(\boldsymbol{x}, y)$

$$\begin{aligned} P(h(\boldsymbol{x}) = y) &= 1 - P(h(\boldsymbol{x}) \neq y) \\ &= 1 - E(h) \\ &< 1 - \epsilon \end{aligned} \tag{1.5}$$

由于 $\mathcal{D}$ 包含 m 个从 $\mathcal{D}$ 中独立同分布采样得到的样例，因此 h 与 $\mathcal{D}$ 表现一直的概率为

$$\begin{aligned} P\left(\left(h\left(\boldsymbol{x}_1\right) = y_1\right) \wedge \ldots \wedge \left(h\left(\boldsymbol{x}_m\right) = y_m\right)\right) &= \left(1 - P(h(\boldsymbol{x}) \neq y)\right)^m \\ &< (1-\epsilon)^m \end{aligned} \tag{1.6}$$

由于 $\mathcal{H}$ 中只有一个假设，我们需要保证泛化误差大于 ϵ，且在训练集上表现完美的所有假设概率之和不大于 δ:

$$\begin{aligned} P(E(h_r) > \epsilon \wedge \widehat{E}(h_r) = 0) &< (1-\epsilon)^m \\ &< e^{-m\epsilon} \\ e^{-m\epsilon} &\leqslant \delta \\ m &\geqslant \frac{log(1/\delta)}{\epsilon} \end{aligned} \tag{1.7}$$

故 $\mathcal{H}$ 是 *PAC* 可学习的，并且样本复杂度为 $\frac{log(1/\delta)}{\epsilon}$。

2 [30pts] 文档主题模型

在一个新闻数据集上实现文档主题模型 (Latent Dirichlet Allocation (LDA)) [1].

我们提供了一个包含 8,888 条新闻的数据集，请在该数据集上完成 LDA 算法的使用及实现。

- 数据集下载：新闻数据集.
- 格式：每行是一条新闻.

数据预处理提示：你可能需要完成分词及去掉一些停用词等预处理工作.

(1) **[10pts] 任务 #1：使用 LDA 模型**

A. 选择开源的 LDA 库（例如：scikit-learn），并在提供的数据集上学习使用.

B. 给出 $K = \{5, 10, 20\}$ 个主题时，每个主题下概率最大的 $M = 10$ 个词及其概率.

(2) **[20pts] 任务 #2：实现 LDA 模型**

A. 不借助开源库，自己完成 LDA 算法.

B. 给出 $K = \{5, 10, 20\}$ 个主题时，每个主题下概率最大的 $M = 10$ 个词及其概率.

Solution. *代码附在 lda_sklearn.py 和 lda.py 中。*

(1) [10pts] 任务 #1：使用 LDA 模型

$K = 5$	*Word1*	*Word2*	*Word3*	*Word4*	*Word5*	*Word6*	*Word7*	*Word8*	*Word9*	*Word10*
Topic1	*game*	*team*	*player*	*season*	*year*	*time*	*point*	*league*	*seri*	*goal*
	3.0%	*1.7%*	*1.4%*	*1.4%*	*1.3%*	*1.0%*	*0.78%*	*0.65%*	*0.61%*	*0.6%*
Topic2	*mr*	*year*	*time*	*people*	*ms*	*work*	*day*	*york*	*way*	*citi*
	1.1%	*1.0%*	*0.93%*	*0.67%*	*0.65%*	*0.51%*	*0.51%*	*0.5%*	*0.49%*	*0.45%*
Topic3	*mr*	*companies*	*trump*	*year*	*percent*	*state*	*clinton*	*campaign*	*parti*	*people*
	2.3%	*1.7%*	*1.4%*	*1.2%*	*1.0%*	*0.95%*	*0.74%*	*0.71%*	*0.69%*	*0.63%*
Topic4	*state*	*year*	*people*	*govern*	*countries*	*offici*	*mr*	*time*	*group*	*attack*
	1.3%	*1.0%*	*1.0%*	*0.92%*	*0.85%*	*0.77%*	*0.76%*	*0.61%*	*0.61%*	*0.5%*
Topic5	*mr*	*school*	*court*	*year*	*office*	*law*	*state*	*case*	*polic*	*student*
	2.5%	*1.3%*	*1.3%*	*1.1%*	*1.1%*	*1.1%*	*1.1%*	*1.0%*	*0.92%*	*0.88%*

$K = 10$	*Word1*	*Word2*	*Word3*	*Word4*	*Word5*	*Word6*	*Word7*	*Word8*	*Word9*	*Word10*
Topic1	*music*	*year*	*song*	*perform*	*time*	*mr*	*prince*	*theater*	*dance*	*day*
	1.5%	*0.91%*	*0.75%*	*0.75%*	*0.71%*	*0.67%*	*0.54%*	*0.54%*	*0.49%*	*0.48%*
Topic2	*time*	*mr*	*ms*	*year*	*people*	*work*	*book*	*art*	*life*	*thing*
	1.5%	*1.3%*	*1.1%*	*1.0%*	*0.96%*	*0.9%*	*0.82%*	*0.76%*	*0.74%*	*0.72%*
Topic3	*game*	*team*	*player*	*season*	*year*	*time*	*point*	*league*	*goal*	*seri*
	4.3%	*2.4%*	*2.0%*	*1.9%*	*1.5%*	*1.1%*	*1.1%*	*0.92%*	*0.85%*	*0.82%*
Topic4	*state*	*govern*	*mr*	*countries*	*preside*	*year*	*offici*	*union*	*people*	*nation*
	1.8%	*1.6%*	*1.4%*	*1.4%*	*0.94%*	*0.86%*	*0.85%*	*0.8%*	*0.71%*	*0.68%*
Topic5	*school*	*people*	*year*	*mr*	*polic*	*office*	*student*	*citi*	*famili*	*state*
	1.8%	*1.6%*	*1.5%*	*1.5%*	*1.5%*	*1.3%*	*1.3%*	*0.95%*	*0.84%*	*0.79%*
Topic6	*citi*	*house*	*home*	*york*	*mr*	*year*	*street*	*room*	*build*	*park*
	1.5%	*1.0%*	*0.9%*	*0.89%*	*0.88%*	*0.86%*	*0.78%*	*0.73%*	*0.67%*	*0.67%*
Topic7	*trump*	*mr*	*clinton*	*campaign*	*state*	*parti*	*voter*	*elect*	*preside*	*sander*
	3.7%	*3.4%*	*2.0%*	*1.6%*	*1.2%*	*1.2%*	*0.94%*	*0.92%*	*0.87%*	*0.87%*
Topic8	*film*	*mr*	*movi*	*time*	*people*	*media*	*facebook*	*season*	*network*	*televis*
	1.7%	*1.4%*	*1.2%*	*1.1%*	*0.95%*	*0.94%*	*0.92%*	*0.81%*	*0.77%*	*0.75%*
Topic9	*mr*	*court*	*state*	*law*	*case*	*lawyer*	*judge*	*year*	*companies*	*justice*
	3.6%	*1.9%*	*1.6%*	*1.4%*	*1.4%*	*0.92%*	*0.89%*	*0.85%*	*0.79%*	*0.75%*
Topic10	*companies*	*year*	*percent*	*market*	*people*	*busi*	*price*	*rate*	*research*	*health*
	2.5%	*2.0%*	*1.7%*	*1.0%*	*0.79%*	*0.77%*	*0.73%*	*0.67%*	*0.59%*	*0.59%*

$K = 20$	*Word1*	*Word2*	*Word3*	*Word4*	*Word5*	*Word6*	*Word7*	*Word8*	*Word9*	*Word10*
Topic1	*company*	*bank*	*year*	*money*	*fund*	*busi*	*tax*	*mr*	*invest*	*market*
	2.9%	*2.3%*	*1.8%*	*1.5%*	*1.4%*	*1.3%*	*1.1%*	*1.1%*	*1.0%*	*1.0%*
Topic2	*art*	*design*	*museum*	*artist*	*fashion*	*work*	*collect*	*exhibit*	*galleri*	*mr*
	2.9%	*2.4%*	*2.4%*	*1.7%*	*1.7%*	*1.5%*	*1.4%*	*1.2%*	*1.2%*	*1.0%*
Topic3	*game*	*run*	*season*	*yanke*	*met*	*inning*	*home*	*team*	*league*	*time*
	2.8%	*1.8%*	*1.8%*	*1.8%*	*1.7%*	*1.3%*	*1.1%*	*1.1%*	*0.96%*	*0.95%*
Topic4	*mr*	*trump*	*clinton*	*campaign*	*state*	*parti*	*preside*	*elect*	*senat*	*voter*
	5.4%	*3.7%*	*1.9%*	*1.8%*	*1.6%*	*1.6%*	*1.3%*	*1.1%*	*1.1%*	*0.98%*
Topic5	*court*	*mr*	*case*	*lawyer*	*law*	*judge*	*state*	*justice*	*investing*	*deci*
	4.0%	*3.5%*	*2.9%*	*2.0%*	*1.9%*	*1.9%*	*1.8%*	*1.6%*	*1.2%*	*0.95%*
Topic6	*polic*	*mr*	*office*	*people*	*citi*	*year*	*man*	*attack*	*investing*	*victim*
	2.9%	*2.6%*	*2.4%*	*1.7%*	*1.4%*	*1.4%*	*0.9%*	*0.88%*	*0.85%*	*0.76%*
Topic7	*health*	*drug*	*dr*	*care*	*year*	*doctor*	*cancer*	*patient*	*research*	*hospital*
	2.8%	*2.3%*	*2.1%*	*1.5%*	*1.4%*	*1.3%*	*1.3%*	*1.3%*	*1.1%*	*1.0%*
Topic8	*food*	*redston*	*restore*	*recipe*	*mr*	*day*	*chef*	*wine*	*cook*	*Viacom*
	2.8%	*1.6%*	*1.4%*	*1.0%*	*0.98%*	*0.85%*	*0.79%*	*0.78%*	*0.73%*	*0.59%*
Topic9	*govern*	*state*	*country*	*mr*	*year*	*offici*	*union*	*preside*	*nation*	*group*
	2.1%	*2.1%*	*1.8%*	*1.5%*	*1.1%*	*1.1%*	*0.99%*	*0.91%*	*0.82%*	*0.8%*
Topic10	*article*	*film*	*time*	*editor*	*news*	*mr*	*york*	*reader*	*stori*	*newspaper*
	2.6%	*1.9%*	*1.8%*	*1.3%*	*1.1%*	*1.1%*	*0.89%*	*0.87%*	*0.82%*	*0.82%*
Topic11	*companies*	*percent*	*year*	*market*	*product*	*price*	*busi*	*technolog*	*industri*	*apple*
	4.1%	*2.2%*	*1.7%*	*1.3%*	*1.1%*	*1.0%*	*0.91%*	*0.88%*	*0.81%*	*0.8%*
Topic12	*prince*	*gawker*	*thieve*	*denton*	*valley*	*media*	*yida*	*yidda*	*com*	*philip*
	4.6%	*2.0%*	*0.98%*	*0.57%*	*0.57%*	*0.51%*	*0.48%*	*0.46%*	*0.44%*	*0.4%*
Topic13	*game*	*goal*	*season*	*team*	*year*	*seri*	*time*	*race*	*player*	*playoff*
	3.2%	*2.0%*	*1.7%*	*1.7%*	*1.5%*	*1.1%*	*1.1%*	*1.0%*	*0.87%*	*0.84%*
Topic14	*time*	*people*	*ms*	*year*	*way*	*thing*	*life*	*day*	*somethe*	*book*
	1.7%	*1.6%*	*1.3%*	*1.1%*	*1.0%*	*1.0%*	*0.96%*	*0.85%*	*0.74%*	*0.7%*
Topic15	*water*	*year*	*people*	*time*	*mile*	*day*	*flight*	*plane*	*air*	*area*
	1.4%	*1.1%*	*0.76%*	*0.7%*	*0.69%*	*0.67%*	*0.59%*	*0.58%*	*0.55%*	*0.54%*
Topic16	*mr*	*music*	*year*	*perform*	*theater*	*film*	*season*	*song*	*movi*	*time*
	2.5%	*1.9%*	*1.4%*	*1.1%*	*1.0%*	*1.0%*	*0.84%*	*0.8%*	*0.75%*	*0.74%*
Topic17	*citi*	*house*	*street*	*home*	*room*	*build*	*york*	*park*	*space*	*mr*
	2.8%	*1.8%*	*1.5%*	*1.4%*	*1.4%*	*1.3%*	*1.3%*	*1.2%*	*1.1%*	*1.0%*
Topic18	*world*	*sport*	*team*	*olymp*	*game*	*athlete*	*year*	*soccer*	*state*	*time*
	2.2%	*2.0%*	*1.9%*	*1.6%*	*1.4%*	*1.2%*	*1.2%*	*1.2%*	*1.1%*	*1.1%*
Topic19	*game*	*player*	*team*	*point*	*season*	*year*	*warrior*	*time*	*league*	*coach*
	4.0%	*2.8%*	*1.9%*	*1.9%*	*1.5%*	*1.4%*	*0.98%*	*0.98%*	*0.92%*	*0.88%*
Topic20	*school*	*student*	*year*	*univers*	*college*	*law*	*state*	*people*	*educate*	*citi*
	4.1%	*2.8%*	*2.0%*	*1.8%*	*1.4%*	*1.4%*	*1.4%*	*1.1%*	*0.99%*	*0.87%*

(2) ***[20pts]*** **任务 *#2*: 实现 *LDA* 模型**

$K=5$	*Word1*	*Word2*	*Word3*	*Word4*	*Word5*	*Word6*	*Word7*	*Word8*	*Word9*	*Word10*
Topic1	*mr*	*state*	*trump*	*preside*	*parti*	*countries*	*campaign*	*people*	*govern*	*clinton*
	2.8%	*1.7%*	*1.5%*	*0.97%*	*0.87%*	*0.83%*	*0.82%*	*0.82%*	*0.82%*	*0.81%*
Topic2	*companies*	*year*	*percent*	*market*	*busi*	*bank*	*price*	*people*	*time*	*execute*
	2.3%	*1.6%*	*1.2%*	*0.83%*	*0.71%*	*0.67%*	*0.62%*	*0.61%*	*0.56%*	*0.52%*
Topic3	*mr*	*year*	*office*	*court*	*citi*	*case*	*polic*	*people*	*state*	*law*
	2.4%	*1.2%*	*1.1%*	*1.1%*	*1.1%*	*1.0%*	*1.0%*	*0.88%*	*0.8%*	*0.7%*
Topic4	*mr*	*time*	*year*	*ms*	*people*	*day*	*work*	*way*	*york*	*thing*
	1.0%	*0.97%*	*0.92%*	*0.66%*	*0.64%*	*0.52%*	*0.51%*	*0.5%*	*0.49%*	*0.43%*
Topic5	*game*	*team*	*season*	*player*	*year*	*time*	*point*	*league*	*seri*	*goal*
	2.9%	*1.6%*	*1.4%*	*1.4%*	*1.3%*	*1.0%*	*0.75%*	*0.62%*	*0.58%*	*0.57%*

$K=10$	*Word1*	*Word2*	*Word3*	*Word4*	*Word5*	*Word6*	*Word7*	*Word8*	*Word9*	*Word10*
Topic1	*game*	*point*	*season*	*redston*	*warrior*	*seri*	*year*	*time*	*sport*	*team*
	2.5%	*1.4%*	*1.1%*	*0.95%*	*0.95%*	*0.9%*	*0.87%*	*0.86%*	*0.84%*	*0.74%*
Topic2	*school*	*student*	*year*	*univers*	*people*	*ms*	*famili*	*college*	*women*	*children*
	2.9%	*1.8%*	*1.6%*	*1.3%*	*1.1%*	*1.1%*	*1.1%*	*1.0%*	*0.92%*	*0.88%*
Topic3	*companies*	*year*	*percent*	*market*	*busi*	*bank*	*mr*	*price*	*execute*	*sale*
	3.4%	*1.8%*	*1.5%*	*1.2%*	*1.0%*	*0.99%*	*0.88%*	*0.84%*	*0.76%*	*0.65%*
Topic4	*mr*	*court*	*office*	*case*	*polic*	*investing*	*law*	*lawyer*	*year*	*judge*
	3.4%	*2.3%*	*1.9%*	*1.9%*	*1.8%*	*1.2%*	*1.2%*	*1.2%*	*1.1%*	*1.1%*
Topic5	*music*	*mr*	*year*	*perform*	*theater*	*song*	*ms*	*art*	*york*	*time*
	1.7%	*1.3%*	*1.2%*	*0.86%*	*0.84%*	*0.79%*	*0.74%*	*0.65%*	*0.61%*	*0.6%*
Topic6	*mr*	*trump*	*time*	*people*	*way*	*book*	*thing*	*film*	*year*	*work*
	2.2%	*1.3%*	*1.3%*	*0.88%*	*0.71%*	*0.69%*	*0.66%*	*0.66%*	*0.66%*	*0.59%*
Topic7	*game*	*team*	*player*	*season*	*year*	*time*	*league*	*goal*	*run*	*day*
	2.8%	*1.9%*	*1.7%*	*1.5%*	*1.4%*	*1.1%*	*0.79%*	*0.76%*	*0.69%*	*0.63%*
Topic8	*citi*	*home*	*year*	*house*	*day*	*street*	*room*	*food*	*build*	*people*
	1.5%	*0.95%*	*0.94%*	*0.88%*	*0.79%*	*0.78%*	*0.76%*	*0.69%*	*0.66%*	*0.64%*
Topic9	*state*	*year*	*health*	*people*	*drug*	*percent*	*research*	*dr*	*senat*	*care*
	2.0%	*1.2%*	*1.0%*	*0.84%*	*0.82%*	*0.74%*	*0.73%*	*0.66%*	*0.62%*	*0.53%*
Topic10	*mr*	*state*	*preside*	*parti*	*clinton*	*govern*	*countries*	*trump*	*campaign*	*people*
	3.3%	*1.9%*	*1.2%*	*1.2%*	*1.2%*	*1.2%*	*1.2%*	*1.1%*	*1.0%*	*0.96%*

$K = 20$	*Word1*	*Word2*	*Word3*	*Word4*	*Word5*	*Word6*	*Word7*	*Word8*	*Word9*	*Word10*
Topic1	*time*	*companies*	*people*	*media*	*apply*	*news*	*facebook*	*year*	*technolog*	*service*
	2.0%	*1.6%*	*1.5%*	*1.0%*	*1.0%*	*0.98%*	*0.96%*	*0.86%*	*0.84%*	*0.79%*
Topic2	*art*	*work*	*music*	*book*	*time*	*artist*	*mr*	*museum*	*year*	*way*
	1.2%	*1.0%*	*0.97%*	*0.87%*	*0.87%*	*0.85%*	*0.85%*	*0.81%*	*0.8%*	*0.7%*
Topic3	*game*	*team*	*player*	*season*	*year*	*time*	*point*	*league*	*goal*	*seri*
	3.6%	*2.0%*	*1.6%*	*1.6%*	*1.3%*	*1.1%*	*0.91%*	*0.78%*	*0.71%*	*0.69%*
Topic4	*house*	*room*	*bedroom*	*home*	*floor*	*oil*	*properties*	*build*	*price*	*design*
	2.0%	*2.0%*	*1.5%*	*1.3%*	*1.1%*	*1.1%*	*1.1%*	*0.99%*	*0.99%*	*0.97%*
Topic5	*court*	*case*	*law*	*mr*	*lawyer*	*state*	*judge*	*justice*	*investing*	*office*
	3.9%	*3.0%*	*2.5%*	*2.1%*	*1.9%*	*1.9%*	*1.8%*	*1.7%*	*1.3%*	*1.2%*
Topic6	*mr*	*trump*	*clinton*	*campaign*	*parti*	*state*	*preside*	*senat*	*elect*	*sander*
	5.8%	*4.2%*	*2.2%*	*1.9%*	*1.6%*	*1.5%*	*1.2%*	*1.1%*	*1.1%*	*0.99%*
Topic7	*york*	*match*	*couple*	*son*	*univers*	*william*	*father*	*tenni*	*groom*	*court*
	1.5%	*1.4%*	*1.3%*	*1.2%*	*1.2%*	*1.1%*	*1.1%*	*1.1%*	*1.0%*	*1.0%*
Topic8	*article*	*ticket*	*plane*	*flight*	*time*	*travel*	*airline*	*airport*	*york*	*passeng*
	2.4%	*1.5%*	*1.4%*	*1.4%*	*1.4%*	*1.3%*	*1.1%*	*1.1%*	*0.89%*	*0.87%*
Topic9	*year*	*state*	*health*	*percent*	*rate*	*worker*	*govern*	*people*	*countries*	*economi*
	2.1%	*1.5%*	*1.3%*	*1.2%*	*1.1%*	*1.0%*	*0.99%*	*0.93%*	*0.87%*	*0.84%*
Topic10	*union*	*govern*	*britain*	*countries*	*mr*	*parti*	*state*	*vote*	*minist*	*europ*
	2.5%	*1.5%*	*1.5%*	*1.4%*	*1.2%*	*1.2%*	*1.2%*	*1.1%*	*1.0%*	*0.84%*
Topic11	*school*	*student*	*year*	*univers*	*college*	*women*	*children*	*people*	*famili*	*ms*
	4.0%	*2.6%*	*2.1%*	*1.6%*	*1.4%*	*1.2%*	*1.2%*	*1.1%*	*1.0%*	*1.0%*
Topic12	*polic*	*mr*	*people*	*office*	*year*	*citi*	*man*	*day*	*famili*	*time*
	2.4%	*1.9%*	*1.7%*	*1.7%*	*1.3%*	*1.1%*	*0.95%*	*0.87%*	*0.83%*	*0.81%*
Topic13	*mr*	*ms*	*year*	*film*	*time*	*movi*	*theater*	*season*	*director*	*redston*
	2.2%	*1.5%*	*1.3%*	*1.2%*	*1.1%*	*0.85%*	*0.78%*	*0.76%*	*0.67%*	*0.64%*
Topic14	*citi*	*year*	*street*	*food*	*day*	*park*	*home*	*restauraunt*	*york*	*mr*
	1.9%	*1.0%*	*0.97%*	*0.93%*	*0.81%*	*0.81%*	*0.78%*	*0.76%*	*0.73%*	*0.73%*
Topic15	*dr*	*research*	*scientist*	*studi*	*dog*	*year*	*anim*	*people*	*science*	*brain*
	2.4%	*1.8%*	*1.1%*	*1.1%*	*0.91%*	*0.9%*	*0.89%*	*0.88%*	*0.75%*	*0.73%*
Topic16	*state*	*mr*	*govern*	*countries*	*preside*	*offici*	*year*	*group*	*war*	*people*
	2.5%	*1.9%*	*1.6%*	*1.6%*	*1.3%*	*1.2%*	*1.0%*	*0.92%*	*0.9%*	*0.9%*
Topic17	*mr*	*year*	*democrat*	*house*	*ms*	*senat*	*leader*	*jew*	*van*	*time*
	6.3%	*0.98%*	*0.59%*	*0.57%*	*0.47%*	*0.43%*	*0.41%*	*0.4%*	*0.39%*	*0.38%*
Topic18	*company*	*percent*	*year*	*bank*	*market*	*busi*	*execute*	*sale*	*mr*	*investor*
	5.1%	*2.0%*	*2.0%*	*1.8%*	*1.7%*	*1.6%*	*1.2%*	*1.2%*	*1.2%*	*1.1%*
Topic19	*drug*	*cancer*	*patient*	*ali*	*doctor*	*treatment*	*hospital*	*cell*	*pain*	*medicin*
	4.8%	*3.7%*	*3.2%*	*2.4%*	*2.3%*	*1.4%*	*1.4%*	*0.93%*	*0.79%*	*0.73%*
Topic20	*car*	*vehicle*	*driver*	*volkswagen*	*race*	*safety*	*model*	*engine*	*owner*	*state*
	3.9%	*2.5%*	*2.3%*	*1.7%*	*1.2%*	*0.97%*	*0.86%*	*0.8%*	*0.79%*	*0.76%*

3 [40pts] 强化学习实验

用 DQN (deep Q Networks) 训练 Flappy Bird. 请各位同学根据 DQN 算法流程，补全提供的代码包中deep_q_networkd.py文件中“# TODO”部分代码 (补全 epsilon-greedy action selection 以及 Q learning updating)，了解 DQN 算法，并进行训练，本实验时间相对较久.

本次实验所需要的依赖如下：

- python2.7 or python3;
- pygame;
- OpenCV-python;
- TensorFlow (建议使用 1.1-1.6).

强化学习中经典的 off-policy 算法 Q-Learning 的原始版本采用表格形式来记录 Q 函数，显然只能应用于有限离散状态、有限离散动作且状态、动作数量较少的情况下，即有维度灾难问题 (表格大小正比于 $|S| * |A|$). 采用函数近似法，假定 Q 函数可由状态特征经过某个函数的映射到对应动作的评价值上，可扩大 Q-Learning 使用范围. 近年来，DeepMind 结合深度模型强大的表达能力，用深度神经网络作为近似函数来表达强化学习中的 Q 函数，进一步扩大了 Q-Learning 可用范围. DQN 中采用 experience replay 和 target network 两种技术，使 DQN 的训练更加高效且鲁棒，并在 atari 的部分游戏上取得了人类水平的表现.

DQN 的流程大致如下 1：

上图是 15 年 DeepMind 发表在 Nature 上文章中所采用的算法流程，包含了 experience replay 和 target network 技术，本次实验不要实现 target network ，仅需要实现 experience replay 即可 (实现 target network 可额外获得 5pts bonus). 感兴趣的同学可参阅 DQN 相关教程或文章，进一步了解两种技术.

本次实验中状态太输入为 raw pixel，转为 $80 * 80$ 的灰度图 (采用 openCV 转换)，并将历史最近 3 个 frame 叠加到当前 frame 中作为状态输入，即每一步输入状态为 $4 * 80 * 80$，动作为 2 维离散动作 (上、下，action 为 2 维 one-hot 编码). 网络模型已经搭建好 (采用 TensorFlow 搭建)，输入为 $4 * 80 * 80$，输出为 2，对应每个动作对应的 Q 值。如下图所示 1.

游戏环境中，单步奖励为 0.1，越过一个管道 $+1$，死亡得到 -1 的惩罚. 可采用其他深度学习框架，如 pytorch、keras 等搭建模型并完成训练代码. DQN 算法设置可采用如下配置：

- GAMMA = 0.99 # decay rate of past observations;
- OBSERVE = 10000. # timesteps to observe before training;
- EXPLORE = 2000000. # frames over which to anneal epsilon;
- FINAL_EPSILON = 0.0001 # final value of epsilon;
- INITIAL_EPSILON = 0.1 0.2 # starting value of epsilon;
- REPLAY_MEMORY = 50000 # number of previous transitions to remember;
- BATCH = 32 # size of minibatch;

Algorithm 1 DQN with experience replay

Initialize replay memory D to capacity N

Initialize action-value function Q with random weights θ

Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$

for *episode* $= 1, M$ **do**

Initialize sequence $s_1 = x_1$ and preprocessed sequence $\phi_1 = \phi(s_1)$

for $t = 1, T$ **do**

With probability ϵ select a random action a_t

otherwise select $a_t = \arg\max_a Q(\phi(s_t), a; \theta)$

Execute action a_t in emulator and observe reward r_t and image x_{t+1}

Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$

Store transition $(\phi_t, a_t, r_t, \phi_t)$ in D

Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from D

Set

$$f(x) = \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j=1}, a'; \theta^-) & \text{otherwise} \end{cases} \tag{3.1}$$

Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the network parameters θ

Every C steps reset $\hat{Q} = Q$

end for

end for

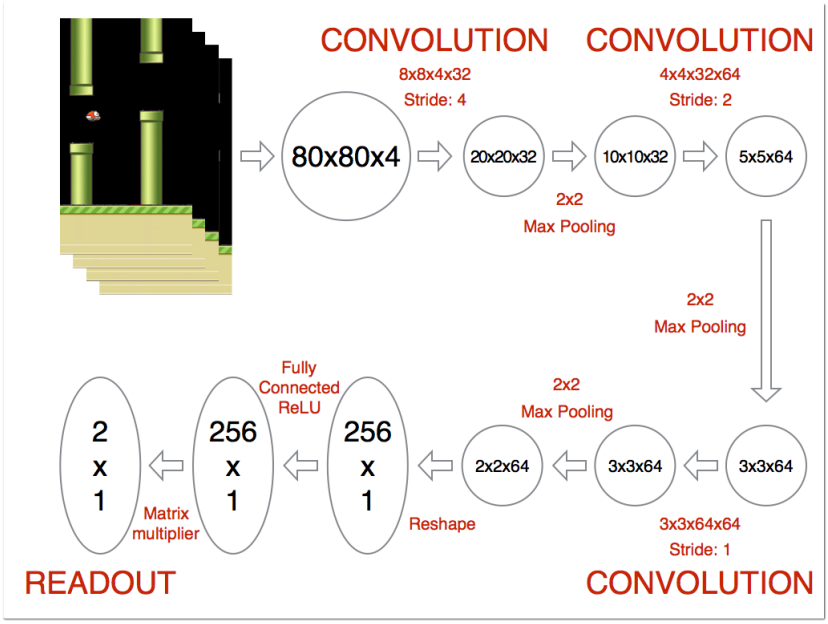

图 1: 网络模型.

- FRAME_PER_ACTION = 1.

默认一直训练不会终止，每 10,000 frames 保存一个模型，默认最大保存 5 个，保存的模型可恢复用来测试，默认保存在 save_model 目录下. 采用 GPU 可加速训练，仅使用双核 CPU 训练时，采用如上配置，总样本量到 1M (1,000,000 个 state) 需要时间为 20 24h，大概 3M 可训练出相当不错的策略，考虑到计算咨询和时间，可自行选择训练量.

采用其他深度学习框架时，只需要保持从环境中获得返回的状态、奖励信息，以及是否终止，并可在环境中执行 action (再次注意，action 为 2 维 one-hot 编码). Agent 与环境交互过程如下所示：

- sys.path.append("game/");
- import wrapped_flappy_bird as game # import game environment;
- game_state = game.GameState() # initialize;
- # execute an action and get info from the environment;
- x_t, r_0, terminal = game_state.frame_step(action).

本实验提交要求：

仅需提供补全后 deep_q_network.py 文件，以及训练后的短视频 (连续飞行 5 – 10s 即可) 或图片或 gif 动图等辅助证明材料，并说明训练使用样本量. 如果有任何修改或补充说明，请一并说明. (建议写 Readme 文件或报告)

Solution. *此处用于写解答 (中英文均可)*

运行了 4080000 次保存的结果。

参考文献

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.