高级机器学习
作业二

brooksj

2019 年 8 月 24 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 $h_r(x) = \mathbb{I}(\parallel x \parallel \leq r)$, 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) 最近邻分类器的 Rademachar 复杂度为无穷大。
证: 最近邻分类器的模型由训练集样本决定, 可以通过构建训练样本来构建一个分类器。对于任意数量为 m 的样本集合 D 中的每一个样本 x_i, 它的最近邻距离为 $d = \min_{x_j \in D \setminus x_i} dist(x_i,\ x_j)$, 在样本附近放置 1 个训练样本 $x_i^{'}$ 满足 $dist(x_i,\ x_i^{'}) < d$, 而最近样本点的类别决定了 x_i 的分类，于是样本分类有 2^m 种，因此最近邻分类器假设空间的 VC 维大小为无穷大。

(2) 证：由课本上 chapter12 的公式 (12.40) 可知常数函数 c 的 Rademacher 复杂度为

$$
\begin{aligned}
\hat{R}_Z(\mathcal{F}) &= \mathbb{E}_\epsilon[\sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(z_i)] \\
&= \mathbb{E}_\epsilon[\frac{1}{m} \sum_{i=1}^{m} \sigma_i c] \\
&= \frac{c}{m} \sum_{i=1}^{m} \mathbb{E}(\sigma_i)
\end{aligned}
$$

又因为 σ_i 是随机变量，以 *0.5* 的概率取 *1*，*0.5* 的概率取*-1*，所以 $\mathbb{E}(\sigma_i) = 0$，由此可得

$$\hat{R}_Z(\mathcal{F}) \;=\; \frac{c}{m}\sum_{i=1}^{m}\mathbb{E}(\sigma_i) = 0$$

同时由课本上 *chapter12* 的公式 *(12.41)* 可得*:*

$$R_m(\mathcal{F}) \;=\; \mathbb{E}_{Z\subseteq\mathcal{Z}:|Z|=m}[\hat{R}_Z(\mathcal{F})] \;=\; 0$$

(3) 证：引用一下 *PAC* 可学习的定义：令 m 表示从分布 $\mathcal{D}$ 中独立同分布采样得到的样例数目，$0 < \epsilon, \delta < 1$，对所有分布 $\mathcal{D}$，若存在学习算法 $\mathcal{L}$ 和多项式 $poly(.,.,.,.)$，使得对于任何 $m \geq poly(1/\epsilon, 1/\delta, size(\mathbf{x}), size(c))$，$\mathcal{L}$ 能从假设空间 $\mathcal{H}$ 中 *PAC* 辨识概念类 $\mathcal{C}$，则称概念类 $\mathcal{C}$ 对假设空间 $\mathcal{H}$ 而言是 *PAC* 可学习的。
本题中 $size(x) = 2, size(c) = 1$，所以只需证明对于任意 $m \geq poly(1/\epsilon, 1/\delta)$，$\mathcal{L}$ 可从假设空间 $\mathcal{H}$ 中 *PAC* 辨识概念类 $\mathcal{C}$ 即证明

$$P(E(h) \leq \epsilon) \geq 1 - \delta \tag{1.1}$$

因为假设空间是可分的，所以目标概念存在于假设空间中，设目标概念 c 为 $c(x) = \mathbb{I}(\parallel x \parallel \leq r_c)$。目标算法为返回与训练集大小一致的最小圆算法，设 r 为训练集的正样本中离原点距离最远的距离，那么 $r \leq r_c$。
假设最终学得的最小圆算法半径为 r_ϵ，设点落在半径为 r_c 的圆和半径为 r_ϵ 之间的概率为 ϵ，采样点在圆环内的概率为 ϵ，不在圆环内的概率为 $1-\epsilon$，保证每次采样都是独立同分布的，则

$$\begin{aligned} P(E(h) \;\leq\; \epsilon) \;&=\; P(min(r_\epsilon,\; r_c) \;\leq\; r \;\leq\; max(r_\epsilon,\; r_c)) \\ &=\; 1 \;-\; (1-\epsilon)^m \;>\; 1 \;-\; e^{-m\epsilon} \end{aligned}$$

结合式 *(1.1)* 只需使 $1 \;-\; e^{-m\epsilon} >= 1 - \delta$，可得 $m \;\geq\; \frac{ln(1/\delta)}{\epsilon}$，显然符合 *PAC* 可学习的定义，因此 $\mathcal{H}$ 是 *PAC* 可学习的，并且样本复杂度为 $\frac{ln(1/\delta)}{\epsilon}$。
证毕!

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.

(1) 解：

k=5 个主题时

表 1: k = 5 个主题时的结果 (sklearn)

	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$	$w7$	$w8$	$w9$	$w10$
t1	executive 0.40%	business 0.39%	percent 0.85%	company 1.29%	student 0.38%	school 0.47%	people 0.45%	market 0.38%	report 0.35%	pay 0.41%
t2	restaurant 0.29%	building 0.40%	people 0.36%	water 0.38%	house 0.31%	time 0.33%	food 0.35%	city 0.71%	day 0.43%	car 0.34%
t3	player 0.92%	season 0.84%	score 0.47%	game 1.38%	team 1.20%	lead 0.53%	play 1.05%	time 0.60%	win 1.01%	hit 0.50%
t4	people 0.42%	write 0.31%	woman 0.33%	York 0.47%	time 0.46%	life 0.29%	book 0.28%	play 0.28%	film 0.26%	New 0.60%
t5	government 0.48%	campaign 0.51%	official 0.44%	Clinton 0.53%	country 0.45%	people 0.57%	United 0.47%	police 0.45%	Trump 1.09%	Obama 0.39%

k=10 个主题时

表 2: k = 10 个主题时的结果 (sklearn)

	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$	$w7$	$w8$	$w9$	$w10$
t1	student 0.71%	federal 0.46%	public 0.48%	school 0.87%	people 0.71%	health 0.41%	court 0.53%	drug 0.39%	law 0.73%	New 0.46%
t2	Warriors 0.48%	driver 0.35%	night 0.37%	James 0.37%	Curry 0.39%	leave 0.37%	time 0.63%	day 0.56%	dog 0.37%	car 0.46%
t3	University 0.61%	graduate 0.73%	receive 0.58%	couple 0.70%	father 0.69%	mother 0.57%	marry 0.57%	York 1.27%	New 1.63%	son 0.61%
t4	artist 0.43%	people 0.32%	music 0.44%	woman 0.54%	time 0.47%	York 0.39%	play 0.39%	wear 0.31%	art 0.46%	New 0.50%
t5	Republican 0.95%	candidate 0.57%	campaign 1.25%	election 0.58%	Clinton 1.37%	Obama 0.80%	party 0.77%	Trump 2.81%	voter 0.64%	vote 0.81%
t6	government 0.88%	official 0.77%	military 0.46%	country 0.63%	officer 0.47%	people 0.58%	attack 0.54%	police 0.91%	United 0.77%	States 0.53%
t7	restaurant 0.44%	building 0.56%	people 0.38%	Street 0.33%	space 0.33%	water 0.36%	house 0.47%	build 0.33%	food 0.45%	city 0.67%
t8	executive 0.55%	financial 0.45%	business 0.59%	percent 1.18%	company 2.16%	market 0.64%	price 0.51%	chief 0.42%	bank 0.46%	pay 0.45%
t9	player 1.13%	season 0.96%	score 0.58%	play 1.17%	game 1.70%	team 1.42%	time 0.64%	lead 0.54%	win 1.18%	hit 0.57%
t10	European 0.76%	Britain 0.61%	people 0.77%	write 0.72%	story 0.60%	Times 0.56%	Union 0.45%	time 0.52%	film 0.51%	book 0.47%

k=20 个主题时

表 3: k = 20 个主题时的结果 (sklearn)

	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$	$w7$	$w8$	$w9$	$w10$
t1	transgender 1.52%	bathroom 0.65%	Carolina 0.48%	lesbian 0.51%	people 0.48%	gender 0.53%	right 0.63%	North 0.50%	bar 0.58%	gay 2.69%
t2	people 0.47%	travel 0.39%	water 0.87%	city 0.66%	time 0.48%	mile 0.43%	hour 0.41%	park 0.40%	day 0.63%	car 0.45%
t3	University 0.80%	graduate 1.02%	daughter 0.80%	father 0.96%	couple 1.02%	mother 0.87%	marry 0.83%	York 1.39%	New 1.82%	son 0.88%
t4	director 0.38%	include 0.46%	fashion 0.37%	design 0.50%	museum 0.38%	artist 0.74%	wear 0.43%	York 0.60%	New 0.73%	art 0.85%
t5	Republican 1.08%	candidate 0.79%	campaign 1.57%	Clinton 1.90%	Sanders 0.77%	Trump 3.89%	voter 0.78%	party 0.87%	Obama 0.71%	win 0.69%
t6	government 1.17%	official 0.79%	military 0.84%	American 0.74%	country 0.76%	Islamic 0.62%	United 1.15%	States 0.76%	attack 0.73%	State 0.62%
t7	neighborhood 0.59%	apartment 0.81%	property 0.62%	building 1.34%	Street 0.70%	house 0.92%	build 0.57%	space 0.56%	city 1.27%	New 0.57%
t8	Broadway 0.79%	Hamilton 0.72%	Russian 0.81%	athlete 0.98%	Olympic 0.83%	Russia 0.86%	North 1.07%	sport 0.89%	Korea 0.79%	test 0.93%
t9	Syndergaard 0.63%	mosquito 0.49%	Collins 1.67%	Harvey 0.92%	Wright 0.85%	virus 1.02%	Mets 2.84%	Zika 1.30%	Kong 0.77%	Hong 0.82%
t10	European 2.68%	Britain 1.74%	British 1.30%	country 0.82%	Europe 1.27%	France 0.84%	Union 1.42%	leave 0.84%	vote 0.94%	Ali 1.04%
t11	Warriors 0.88%	season 0.77%	player 0.69%	James 0.73%	Curry 0.72%	game 1.28%	team 1.16%	Game 0.75%	play 0.76%	win 0.77%
t12	investigation 0.84%	Redstone 0.62%	federal 0.69%	charge 0.75%	lawyer 1.34%	judge 0.84%	court 1.70%	legal 0.66%	file 0.64%	law 0.71%
t13	executive 0.92%	business 0.95%	company 3.61%	percent 0.62%	service 0.57%	chief 0.71%	sell 0.66%	sale 0.70%	deal 0.55%	pay 0.63%
t14	player 1.15%	season 0.93%	score 0.61%	game 1.65%	team 1.36%	play 1.19%	time 0.66%	goal 0.57%	win 1.17%	hit 0.69%

t15	University 0.62%	student 1.38%	program 0.55%	percent 0.51%	school 1.69%	people 0.81%	health 0.75%	study 0.71%	child 0.52%	drug 0.67%
t16	government 0.71%	economic 0.73%	Chinese 0.83%	percent 2.06%	economy 0.77%	market 0.87%	China 1.41%	rise 0.77%	bank 0.71%	rate 0.70%
t17	people 0.90%	woman 0.58%	story 0.58%	write 0.71%	play 0.61%	time 0.93%	film 0.56%	life 0.66%	book 0.52%	don 0.65%
t18	article 0.79%	public 0.72%	Senate 0.53%	House 0.68%	Times 0.87%	write 0.55%	York 0.85%	news 0.50%	law 0.91%	New 1.05%
t19	shooting 0.55%	officer 1.24%	people 1.16%	arrest 0.58%	victim 0.60%	police 2.34%	death 0.55%	kill 0.65%	city 0.62%	gun 0.76%
t20	restaurant 0.86%	recipe 0.62%	album 0.74%	music 1.24%	food 1.11%	song 0.87%	wine 0.55%	cook 0.55%	chef 0.47%	eat 0.47%

(2) 解：

$k = 5$ 个主题时

表 4: k = 5 个主题时的结果

	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$	$w7$	$w8$	$w9$	$w10$
t1	executive 0.34%	campaign 0.48%	business 0.35%	company 1.10%	percent 0.79%	Clinton 0.54%	people 0.40%	market 0.32%	Trump 1.11%	vote 0.39%
t2	government 0.52%	official 0.55%	country 0.43%	student 0.34%	people 0.65%	police 0.51%	United 0.51%	school 0.42%	States 0.38%	law 0.38%
t3	building 0.39%	include 0.29%	people 0.36%	house 0.34%	space 0.29%	city 0.51%	time 0.31%	food 0.29%	New 0.39%	day 0.31%
t4	player 0.95%	season 0.79%	score 0.50%	game 1.51%	team 1.23%	play 1.00%	time 0.63%	lead 0.51%	win 1.04%	hit 0.50%
t5	people 0.53%	woman 0.44%	write 0.42%	time 0.53%	play 0.41%	York 0.41%	life 0.36%	book 0.33%	film 0.31%	New 0.53%

k=10 个主题时

表 5: k = 10 个主题时的结果

	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$	$w7$	$w8$	$w9$	$w10$
t1	technology 0.46%	Facebook 0.60%	company 1.48%	service 0.48%	people 0.77%	online 0.52%	Apple 0.43%	media 0.42%	time 0.48%	car 0.47%
t2	University 0.67%	student 1.37%	school 1.80%	family 0.74%	father 0.66%	mother 0.62%	people 0.62%	child 0.81%	York 0.88%	New 1.16%
t3	republican 0.63%	political 0.58%	campaign 1.15%	Clinton 1.25%	support 0.56%	Trump 2.58%	party 0.74%	Obama 0.69%	voter 0.59%	vote 0.85%
t4	financial 0.58%	executive 0.51%	business 0.65%	company 1.74%	percent 1.61%	market 0.77%	price 0.65%	money 0.55%	bank 0.57%	pay 0.65%
t5	player 1.05%	season 0.86%	score 0.56%	game 1.67%	team 1.35%	play 1.11%	time 0.63%	lead 0.52%	win 1.11%	hit 0.55%
t6	people 0.40%	write 0.46%	music 0.40%	woman 0.38%	play 0.59%	time 0.56%	film 0.47%	book 0.42%	life 0.37%	love 0.36%
t7	official 0.45%	officer 0.51%	federal 0.41%	police 0.80%	people 0.56%	lawyer 0.53%	report 0.45%	charge 0.45%	court 0.74%	law 0.63%
t8	restaurant 0.72%	Redstone 0.68%	Britain 0.71%	recipe 0.48%	Union 0.45%	leave 0.44%	food 0.98%	eat 0.49%	day 0.49%	dog 0.48%
t9	government 0.97%	official 0.70%	american 0.54%	military 0.54%	country 0.92%	United 0.96%	States 0.66%	people 0.61%	attack 0.57%	China 0.59%
t10	building 0.64%	include 0.37%	people 0.39%	house 0.54%	space 0.46%	city 0.85%	York 0.39%	live 0.37%	New 0.52%	art 0.45%

k=20 个主题时

表 6: k = 20 个主题时的结果

	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$	$w7$	$w8$	$w9$	$w10$
t1	restaurant 1.18%	chicken 0.42%	cooking 0.41%	recipe 0.70%	serve 0.45%	food 1.43%	wine 0.67%	chef 0.53%	cook 0.50%	eat 0.59%
t2	tournament 0.68%	athlete 0.63%	player 1.30%	United 0.65%	sport 0.90%	match 0.79%	team 1.36%	play 1.10%	game 0.85%	win 1.33%
t3	University 0.93%	graduate 0.94%	daughter 0.75%	receive 0.81%	father 1.07%	couple 0.88%	mother 0.86%	York 2.42%	New 3.21%	son 0.82%
t4	history 0.61%	church 0.67%	family 0.57%	Israel 0.57%	write 0.69%	India 0.50%	black 0.43%	book 0.90%	Ali 1.28%	war 0.43%
t5	hospital 0.66%	patient 0.88%	medical 0.73%	health 1.30%	doctor 0.82%	people 0.74%	cancer 0.72%	study 0.86%	drug 1.26%	care 0.74%
t6	financial 0.70%	increase 0.81%	percent 2.58%	company 0.72%	market 1.12%	price 0.90%	bank 0.86%	rate 0.85%	rise 0.75%	pay 0.71%
t7	Yankees 1.07%	inning 1.03%	season 1.02%	pitch 0.88%	start 0.86%	game 1.48%	Mets 0.98%	hit 1.46%	run 0.85%	win 0.84%
t8	people 1.25%	friend 0.52%	time 1.23%	life 0.79%	tell 0.61%	feel 0.59%	talk 0.53%	call 0.53%	live 0.50%	day 0.88%
t9	performance 0.46%	character 0.46%	musical 0.45%	season 0.56%	music 0.96%	movie 0.67%	film 1.11%	play 1.01%	song 0.62%	star 0.48%
t10	university 0.75%	University 0.68%	education 0.63%	student 2.88%	college 1.05%	program 0.64%	school 3.51%	child 0.67%	class 0.67%	car 0.87%
t11	official 0.61%	decision 0.57%	federal 0.66%	lawyer 0.73%	court 1.08%	legal 0.53%	Court 0.50%	issue 0.49%	rule 0.53%	law 1.21%
t12	republican 0.95%	candidate 0.78%	campaign 1.49%	Clinton 1.88%	Sanders 0.76%	Trump 3.86%	Obama 0.87%	party 0.78%	voter 0.77%	vote 0.73%
t13	technology 0.55%	executive 0.95%	business 0.99%	customer 0.53%	company 3.91%	service 0.66%	chief 0.75%	deal 0.62%	sell 0.60%	sale 0.54%
t14	shooting 0.67%	officer 1.22%	police 2.30%	people 1.03%	charge 0.71%	arrest 0.65%	victim 0.61%	kill 0.66%	gun 0.76%	gay 0.66%

t15	government	European	european	Britain	country	british	Europe	Union	leave	vote
	0.77%	1.08%	0.98%	1.45%	1.35%	0.92%	1.13%	1.20%	0.96%	1.03%
t16	apartment	building	artist	museum	house	space	city	York	art	New
	0.57%	1.08%	0.61%	0.58%	0.81%	0.74%	1.30%	0.62%	0.92%	0.74%
t17	government	military	official	american	country	United	States	attack	China	force
	1.19%	0.90%	0.89%	0.79%	0.80%	1.29%	0.89%	0.61%	0.98%	0.58%
t18	island	animal	travel	people	water	plane	fire	mile	dog	fly
	0.46%	0.42%	0.41%	0.38%	0.91%	0.38%	0.47%	0.39%	0.40%	0.36%
t19	designer	article	fashion	editor	Times	media	woman	wear	news	New
	0.58%	1.11%	0.72%	0.66%	1.26%	0.63%	0.60%	0.82%	0.67%	0.58%
t20	player	season	series	score	coach	game	team	play	lead	win
	1.22%	1.13%	0.82%	0.84%	0.81%	2.48%	1.68%	1.43%	0.81%	0.96%

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

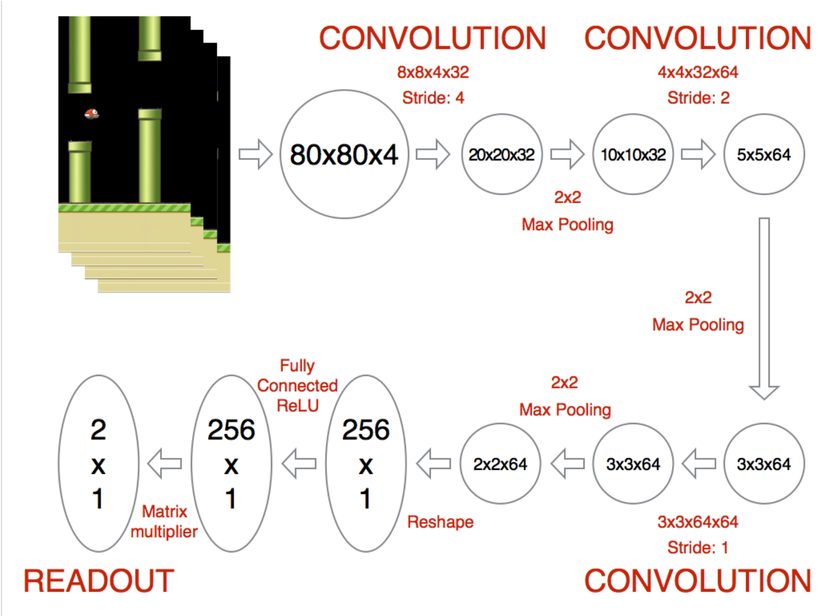

图 1: 网络模型.

- 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 $5-10$s 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)*

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.