Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

对抗环境强化学习算法

硕士研究生　王逸洲

2019年12月08日

内容提要

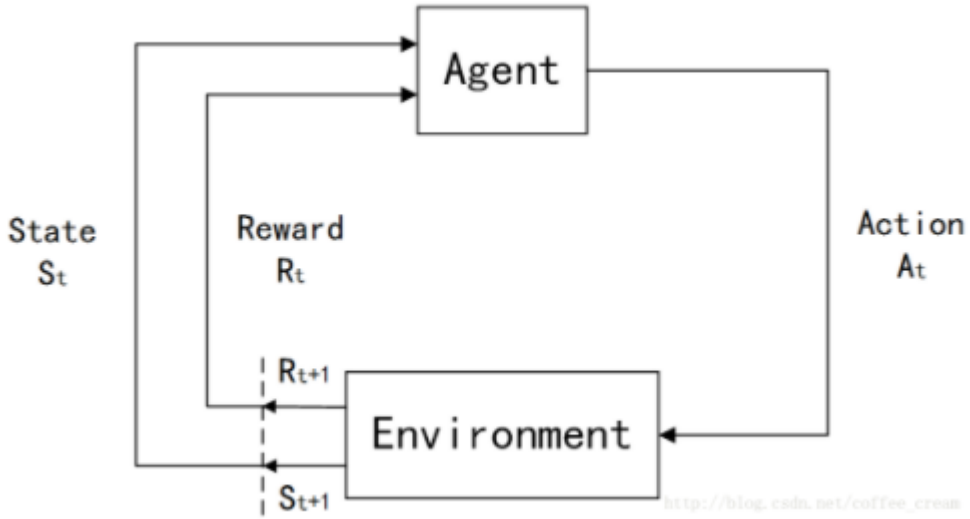
- 背景简介
- 基本概念
  - 强化学习
  - Q-Learning
  - 深度Q网络
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

背景简介

- 预期收获
  - 1. 了解强化学习相关基础知识
  - 2. 理解Q-Learning、了解深度Q网络算法原理
  - 3. 了解对抗环境强化学习对不平衡数据学习的改进效果

基本概念

- 强化学习
  - 主要包含四个元素，智能体(Agent)，环境状态(State)，行动(Action)，奖励(Reward)
  - 本质是解决 decision making 问题，即自动进行决策，并且可以做连续决策



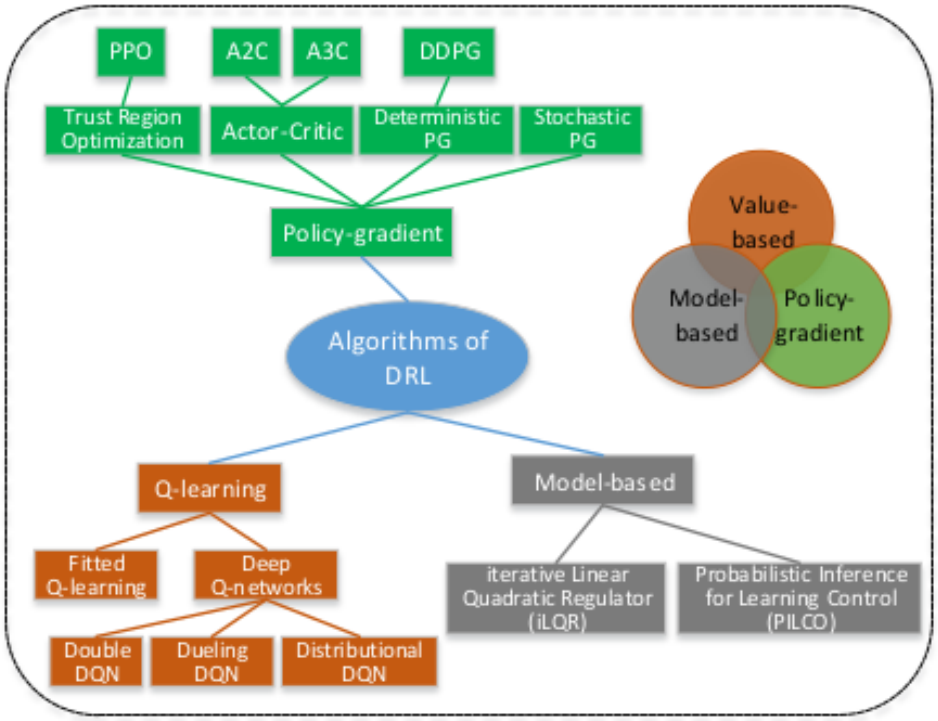
- 在每个时间步(timestep)，智能体在当前状态根据观察来确定下一步的动作。
- 策略：状态和动作之间的映射关系，记作：$A=\pi(S)$

基本概念

- 强化学习与监督学习、无监督学习
  - 监督学习与强化学习
    - 有监督学习从已有标记的训练集中学习
    - 样本特征-state、标签-action
    - 有监督学习不能学习交互的情景
  - 无监督学习与强化学习
    - 无监督学习从未标记样本中发现隐藏结构
    - 强化学习的目的是最大化奖励信号
  - 强化学习特点
    - 其中没有监督者，只有一个reward信号
    - 反馈是延迟的，不是立即生成的
    - agent的行为会影响之后一系列的data

基本概念

- Value Based
  - 关注点是找到最优奖励总和
  - 输出的是所有动作的价值，根据最高价值来选动作，不能选取连续的动作
- Policy Based
  - 关注点是找到最优策略
  - 通过分析所处的环境，直接输出下一步要采取的各种动作的概率，然后根据概率采取行动



基本概念

- 回报(Return)
  - 为最大化长期累积奖赏，定义当前时刻后的累积奖赏为回报(Return)
  - 考虑折扣因子(避免时间过长时，总回报无穷大):

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

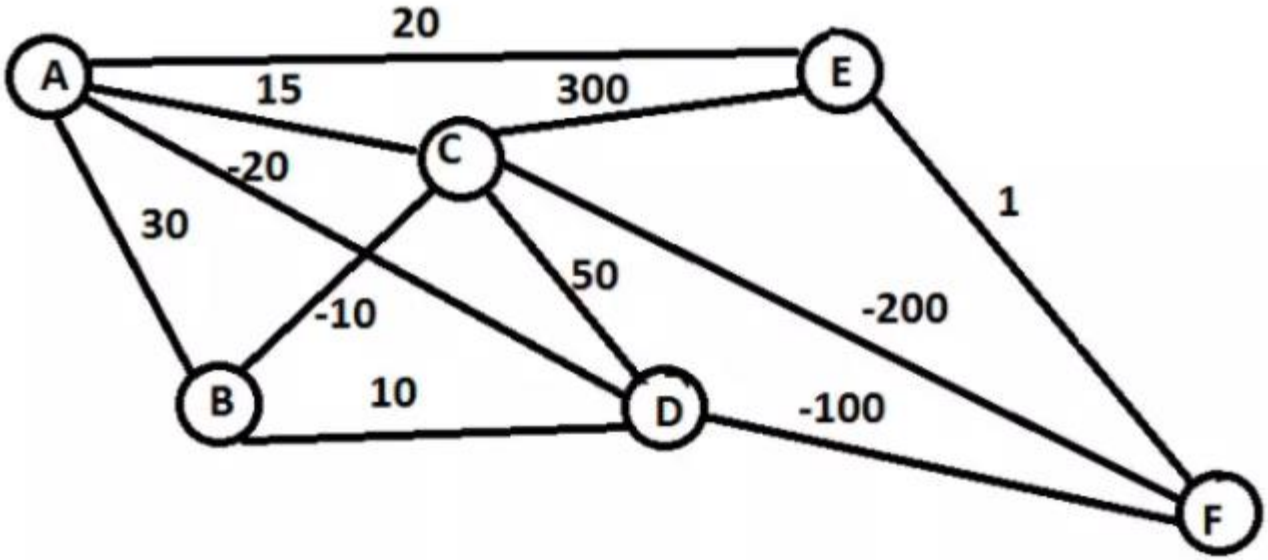
- 值函数
  - 状态值函数：在状态S下获得的期望回报。

$$V^{\pi}(s) = E_{\pi}\left[R_t | s_t = s\right] = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right]$$

  - 状态-动作值函数：在状态S下执行动作A后获得的期望回报

$$Q^{\pi}(s,a) = E_{\pi}\left[R_t | s_t = s, a_t = a\right] = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right]$$

基本概念



- states **就是节点** {A, B, C, D, E, F}
- action **就是从一点走到下一点** {A -> B, C -> D, …}
- reward **就是边上的** cost
- policy**就是完成任务的整条路径** {A -> C -> F}

基本概念

- Q-Learning
  - 强化学习算法中Value-based的算法，Q即为Q（s,a）就是在某一时刻的 s 状态下(s∈S)，采取 动作a (a∈A)动作能够获得收益的期望
  - 主要思想就是将State与Action构建成一张Q-table来存储Q值，然后根据Q值来选取能够获得最大的收益的动作

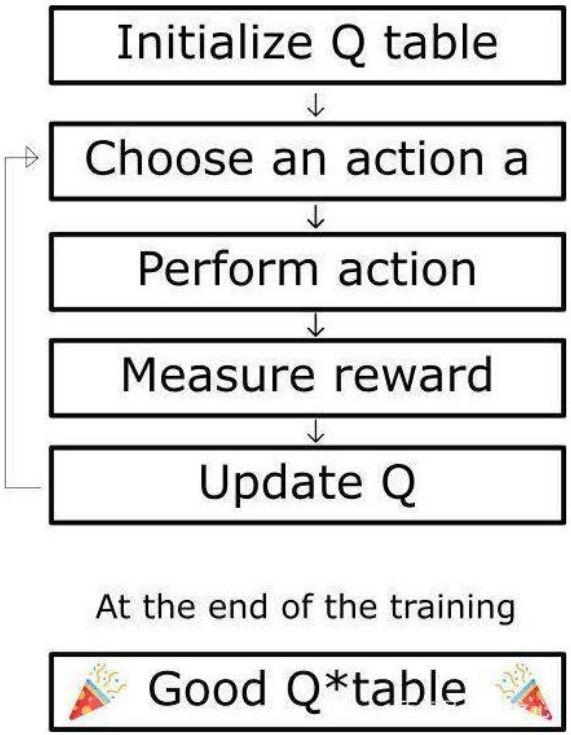
基本概念

- 算法流程

算法 14.4: Q学习：一种异策略的时序差分学习算法

输入: 状态空间 $\mathcal{S}$，动作空间 $\mathcal{A}$，折扣率 γ，学习率 α

随机初始化 $Q(s,a)$;
$\forall s, \forall a, \pi(a|s) = \frac{1}{|\mathcal{A}|}$;
repeat
  初始化起始状态 s;
  repeat
    在状态 s，选择动作 $a = \pi^{\epsilon}(s)$;
    执行动作 a，得到即时奖励 r 和新状态 s';
    $Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$;
    $s \leftarrow s'$;
  until s 为终止状态;
until $\forall s, a$，$Q(s,a)$ 收敛;
输出: 策略 $\pi(s) = \arg\max_{a \in |\mathcal{A}|} Q(s,a)$

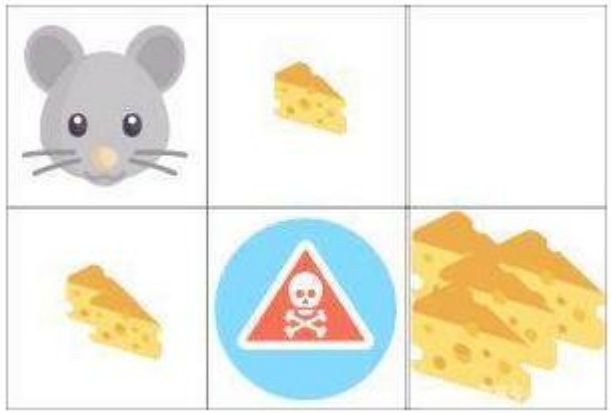


基于时序差分更新Q值：

$$Q(S,A) \leftarrow Q(S,A) + \alpha[r + \gamma \max_{a'} Q(S',A') - Q(S,A)]$$

- α：学习率
- γ：奖励衰变系数

基本概念



- 一块奶酪 = +1
- 两块奶酪 = +2
- 一堆奶酪 = +10（训练结束）
- 毒药 = −10（训练结束）

	左	右	上	下
开始	0	0	0	0
一块奶酪	0	0	0	0
无	0	0	0	0
两块奶酪	0	0	0	0
死亡	0	0	0	0
一堆奶酪	0	0	0	0

基本概念

	左	右	上	下
开始	0	0	0	0
一块奶酪	0	0	0	0
无	0	0	0	0
两块奶酪	0	0	0	0
死亡	0	0	0	0
一堆奶酪	0	0	0	0

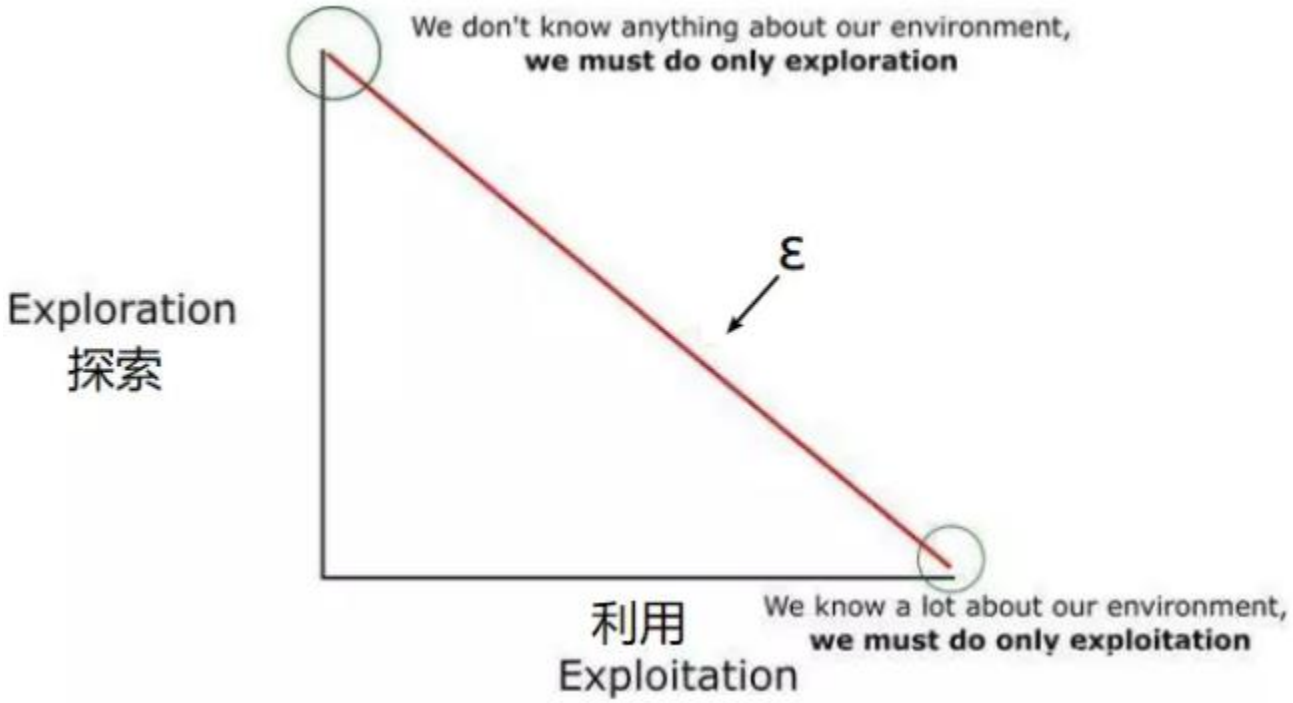
	左	右	上	下
开始	0	1	0	0
一块奶酪	0	0	0	0
无	0	0	0	0
两块奶酪	0	0	0	0
死亡	0	0	0	0
一堆奶酪	0	0	0	0

基本概念

- 探索-利用
  - 探索(Exploration)
    - 做出新的决策来收集更多信息
    - 过多的探索会阻碍智能体最大限度地获得短期奖励
  - 利用(Exploition)
    - 根据当前信息做出最佳决策
    - 靠不完全知识来利用环境会阻碍长期奖励的最大化

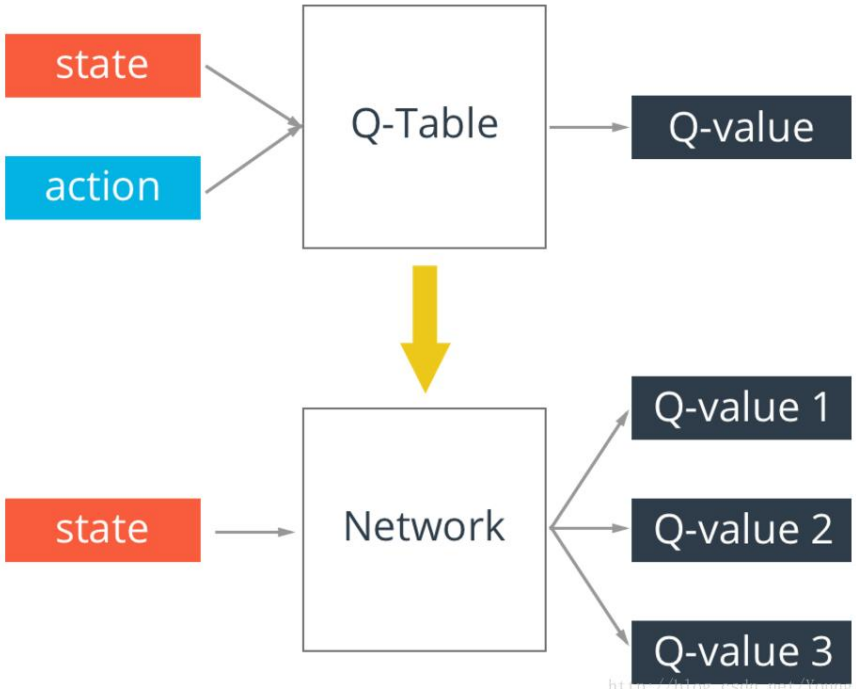
基本概念

- ε -greedy
  - 每次有 ε 概率进行探索，有(1- ε)的概率利用已学习的数据



基本概念

- 深度Q网络（Deep Q Network,DQN）
  - 传统的强化学习局限于动作空间和状态空间都很小，且一般是离散的情境下。
  - 比较复杂的、更加接近实际情况的任务则往往有着很大的状态空间和连续的动作空间。



基本概念

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    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 $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

- 相比Q-Learning的主要改进
  - 利用深度卷积网络(CNN)来逼近值函数；
  - 利用经验回放训练强化学习的学习过程；
  - 独立设置了目标网络来单独处理时序差分中的偏差

算法原理

T	改善分类器对不平衡数据的分类结果
I	带标记的入侵检测样本数据
P	1. 初始化Q函数、初始状态S0 2. 环境根据当前状态和其策略选择一个动作Aet(入侵样本的种类标签) 3. 环境从数据集中随机选取一个状态S(Aet)，其提供一个 特征-标签对（St，Aet） 4. 分类器分类并分配动作Act 5. 环境响应分类动作并返回奖励 6. 环境再次基于其动作值Aet+1和其策略提供下一组 特征-标签对(St+1,Aet+1) 7. 分类器和环境主机依据DQN更新规则更新策略函数
O	高准确率的分类模型

P	强化学习算法存在对不平衡数据敏感的问题
C	对抗环境强化学习(AE-RL)
D	从不平衡样本中选取样本数少的样本进行训练
L	CCFB类

算法原理

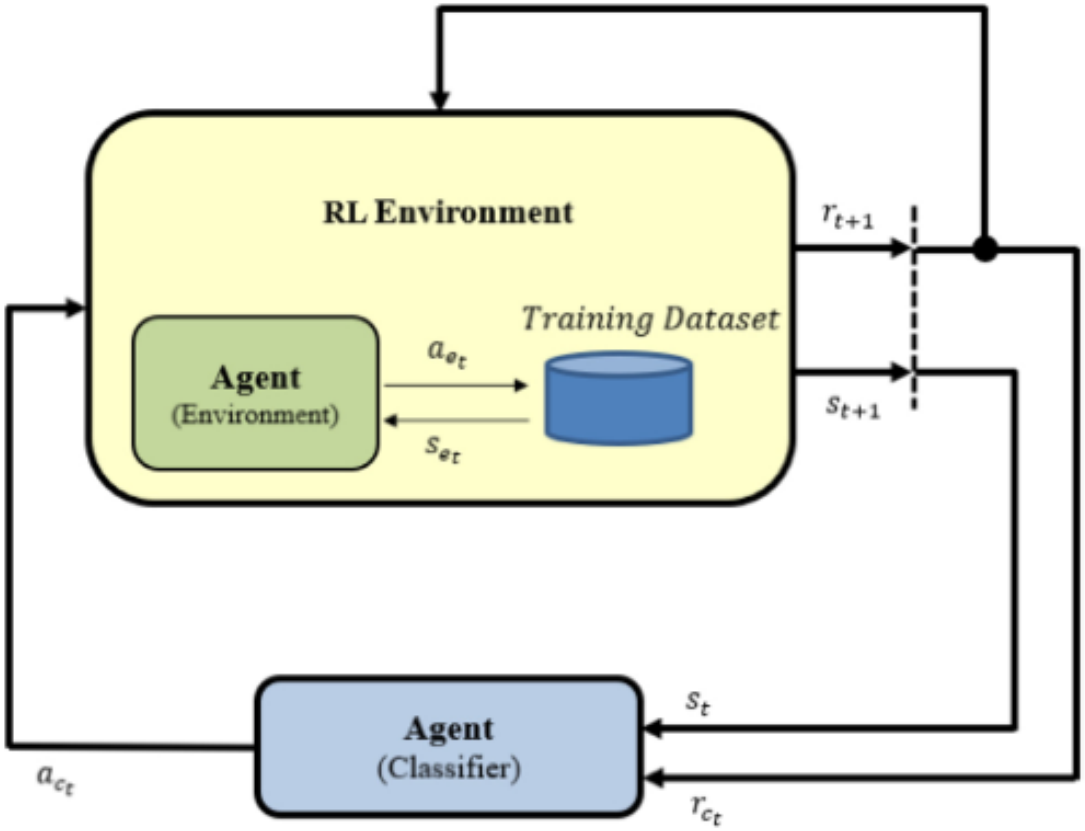


Fig. 3. Reinforcement learning interaction between the agent and its intelligent environment.

- 攻击选择器
  - 选择使分类器产生错误分类的样本
  - 状态S_{et}–样本特征
  - 动作A_{et}–种类标签
- 分类器
  - 根据样本特征进行正确分类
  - 状态S_{ct}–样本特征
  - 动作A_{ct}–种类标签

算法原理

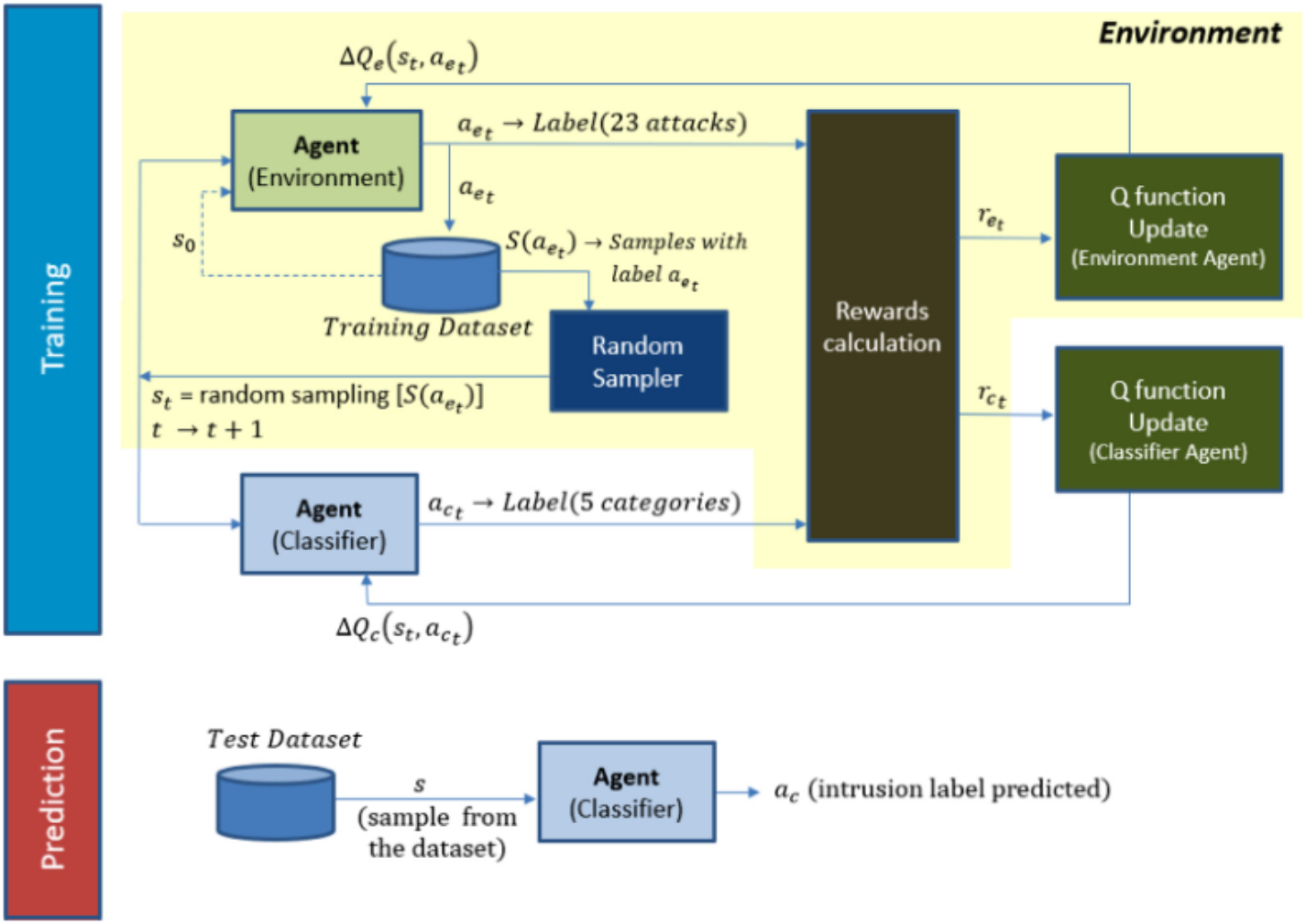


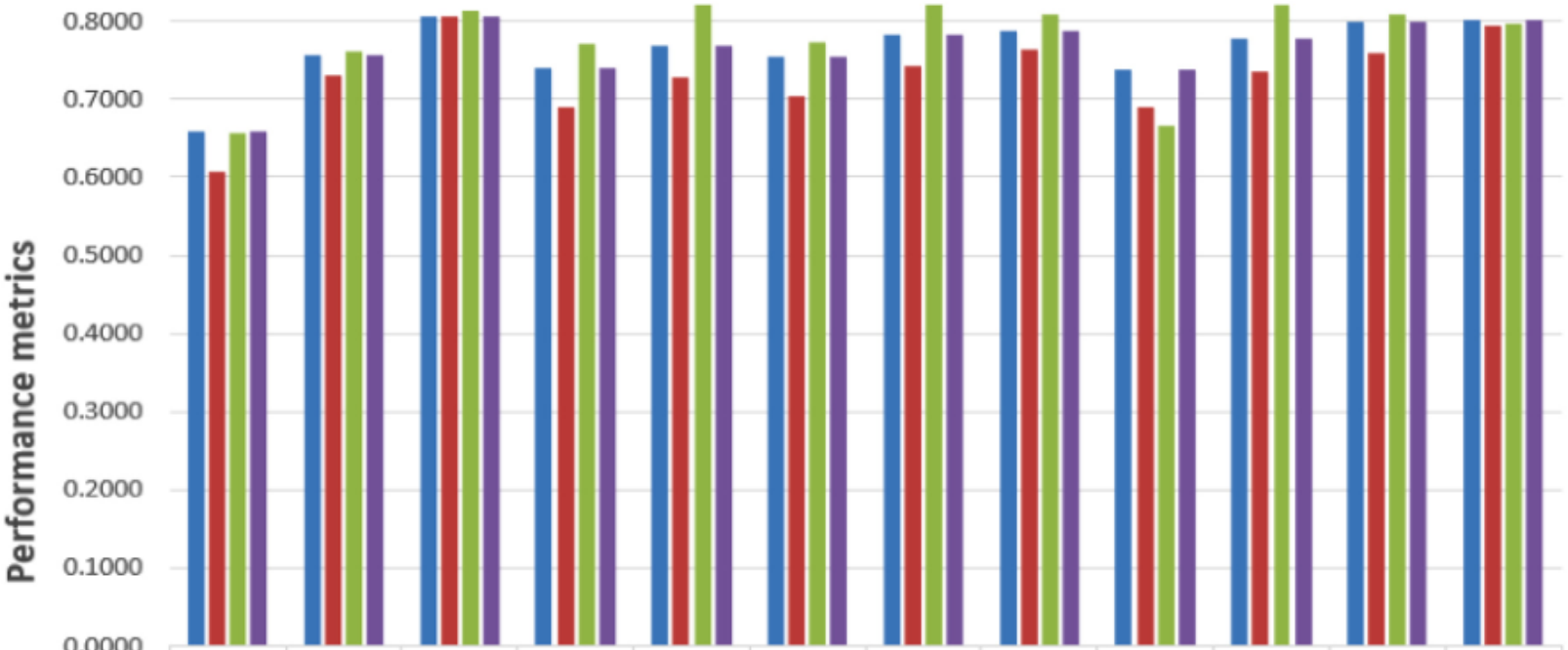
Fig. 4. Details of the AE-RL algorithm for the training and prediction phases.

算法原理

- AE-RL与bagging、boost算法的区别
  - AE-RL根据每个样本的训练误差修改训练频率，处理不平衡并提供优秀的泛化误差。
  - Boost是改变每个样本的权值，增加一系列加性模型(集成模型)中误差较大的样本的权值。
  - Bagging是利用原始数据集的随机抽样(bootstrap)对一系列模型进行训练，最终得到所有模型的平均结果。数据集的连续重采样是基于随机抽样，而不是基于训练结果。

算法原理

- 实验结果



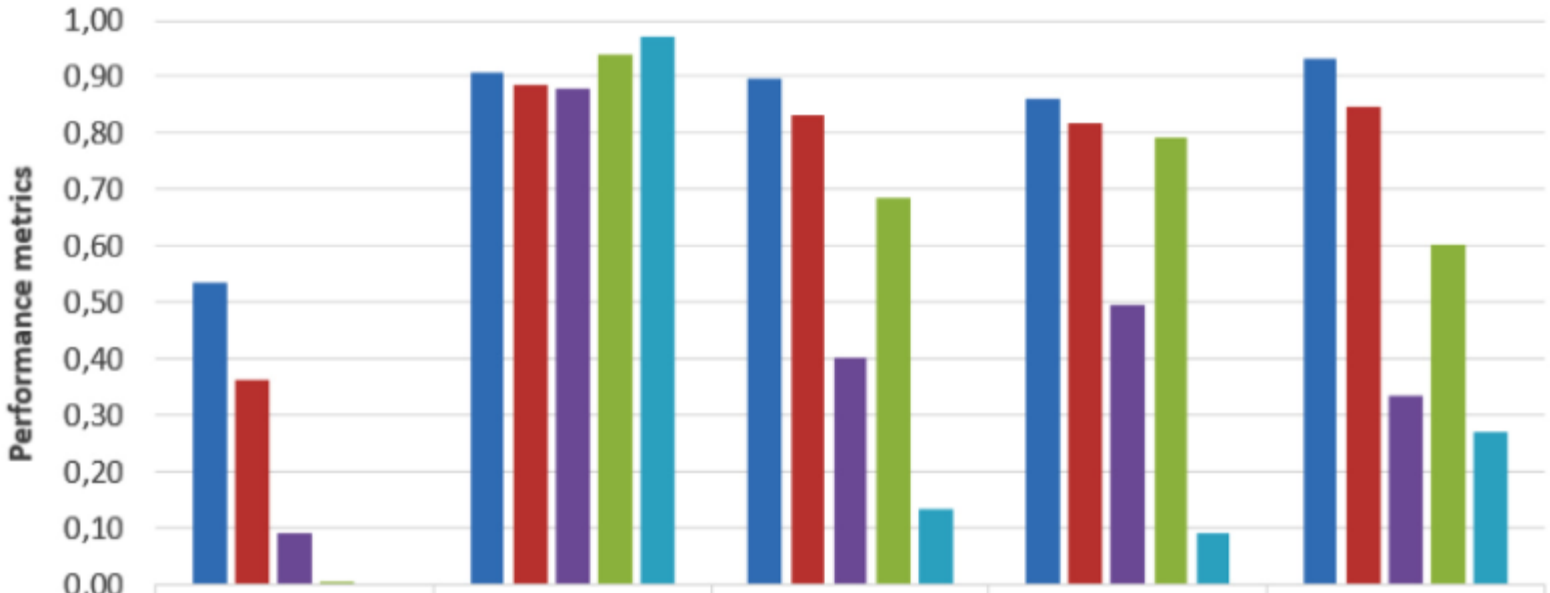
	Log. Regre.	SVM: Linear Kernel	SVM: RBF Kernel	RF	GBM	Ada-Boost	MLP	CNN-1D	DRL: DDQN	DRL: Dueling	DRL: A3C	AE-RL
Accuracy	0.6602	0.7560	0.8065	0.7391	0.7676	0.7531	0.7831	0.7875	0.7372	0.7788	0.8000	0.8016
F1	0.6066	0.7295	0.8056	0.6909	0.7284	0.7044	0.7429	0.7633	0.6898	0.7358	0.7600	0.7940
Precision	0.6570	0.7622	0.8130	0.7708	0.8252	0.7728	0.8262	0.8094	0.6661	0.8206	0.8100	0.7974
Recall	0.6602	0.7560	0.8065	0.7391	0.7676	0.7531	0.7831	0.7875	0.7372	0.7788	0.8000	0.8016

Fig. 7. Aggregated performance scores for all models (NSL-KDD dataset).

算法原理

- 实验结果

G. Caminero, M. Lopez-Martin and B. Carro / Computer Networks 159 (2019) 96–109

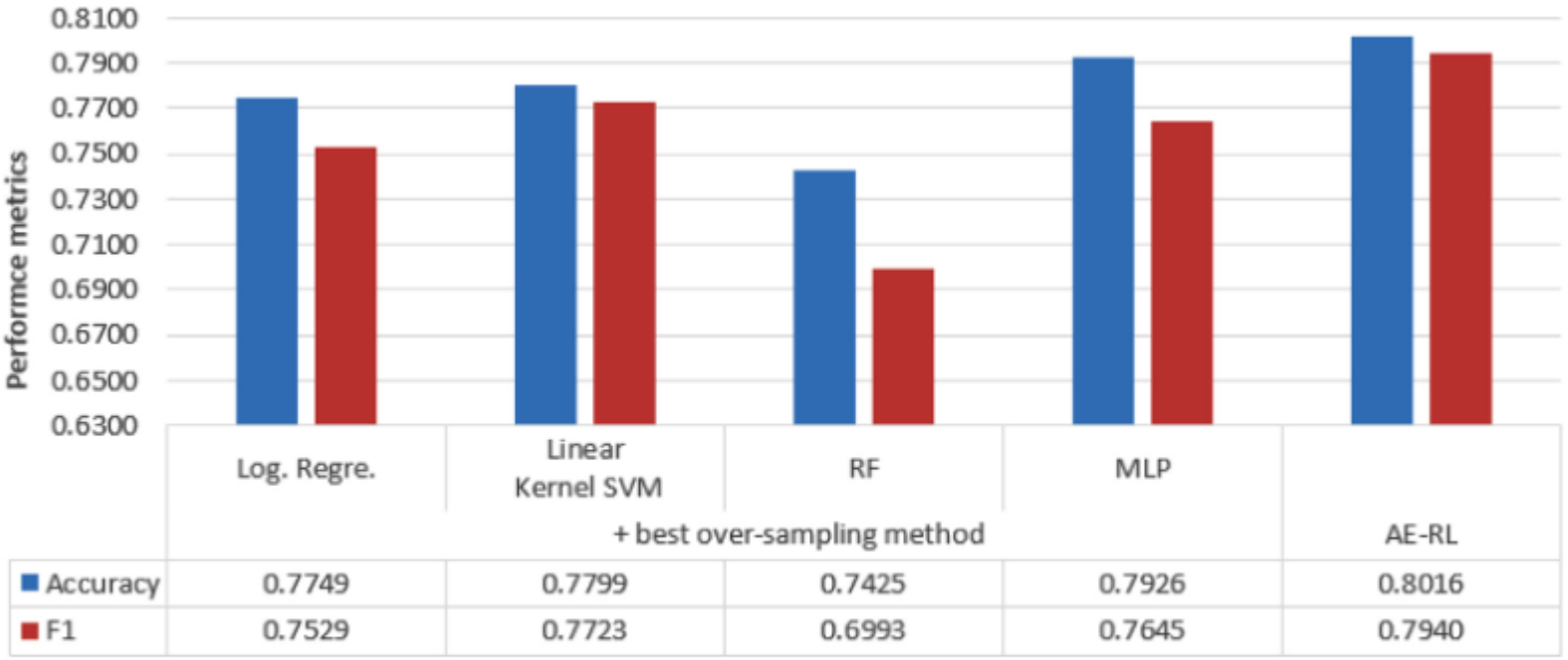


	Frequency	Accuracy	F1	Precision	Recall
NORMAL	53,46%	0,9056	0,8948	0,8603	0,9322
DOS	36,46%	0,8873	0,8326	0,8183	0,8474
PROBE	9,25%	0,8775	0,4019	0,4976	0,3371
R2L	0,79%	0,9405	0,6851	0,7930	0,6031
U2R	0,04%	0,9697	0,1365	0,0914	0,2700

Fig. 8. One vs. Rest performance scores for the AE-RL model applied to the NSL-KDD dataset.

算法原理

- 实验结果



	Log. Regre.	Linear Kernel SVM	RF	MLP	AE-RL
	+ best over-sampling method				
Accuracy	0.7749	0.7799	0.7425	0.7926	0.8016
F1	0.7529	0.7723	0.6993	0.7645	0.7940

Fig. 9. Performance metrics of AE-RL compared to the performance metrics of other ML algorithms + over-sampling method (NSL-KDD).

应用总结

- 算法的应用领域
  - 入侵检测等分类任务
  - 对不平衡数据的一种过采样方法

参考文献

[1] Dpto. TSyCeIT, et alAdversarial environment reinforcement learning algorithm for intrusion detection [J]. arXiv preprint arXiv:1904.12848, 2019.
[2] https://www.jianshu.com/p/f4409a8b7f71
[3] https://www.jianshu.com/p/6436b99dcaef

谢谢！

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。