# Machine Learning Foundation Fall 2020 — Homework 1

Yu-Teng Li, SID B08502093

**Collaborators** (discussion only, solutions written independently): SOME NAMES (or none).

## A. The Learning Problem

**Question 1:** Answer: (d)
(a) No. Lottery numbers are likely uniformly random, and thus it's unlikely to solve this with ML.
(b) No. We don't need ML to solve this, and since average is deterministic, and each student's score is an I.I.D., there's hardly a pattern to approach.
(c) No. A graph's MST does not have to be unique. Therefore, we can have many possible solutions, which is not an ideal phenomenon for ML training.
(d) Yes. This is possible since we have labeled data, and the image pattern does exist.

**Question 2:** Answer: (e)
Option e clearly indicates the "scoring" notion in Machine Learning. Instead of relying on random choices such as coin flips, or manual labeling as in option (b), option (e) offers both spam and non-spam data for the algorithm to learn and evaluate a threshold (or, decision boundary) that it can optimize on.

## B. Perceptron Learning Algorithm

**Question 3:** Answer: (d)
Since the original bound is $\frac{R^2}{\rho^2}$. After substituting each x with 0.25x, the ratio for upper bound did not change.

**Question 4:** Answer: (c)
Reasoning as the follows. Inspiration from professor's PLA convergence proof notes.

$$\begin{aligned} w_f^T w_{t+1} &= w_f^T (w_t + \eta_t y_{n(t)} x_{n(t)}) \\ &= w_f^T w_t + w_f^T \eta_t y_{n(t)} x_{n(t)} \\ &= w_f^T w_t + \frac{|w_f^T \eta_t|}{||x_{n(t)}||} \geq w_f^T w_t + ||w_f||\hat{\rho} \end{aligned} \tag{1}$$

So basically after T equations, $w_f^T w_T \geq w_f^T w_0 + T * ||w_f||\hat{\rho}$. Since every iteration in PLA conducts an update before it converges, we can derive the following given that $\eta_t = \frac{1}{||x_{n(t)}||}$ :

$$\begin{aligned} \parallel w_{t+1} \parallel^2 &= ||w_t + \eta_t y_{n(t)} x_{n(t)}||^2 = ||w_t||^2 + 2y_{n(t)} * \eta_t w_t x_{n(t)} + \eta_t^2 y_{n(t)}^2 ||x_{n(t)}||^2 \\ &\leq ||w_t||^2 + \eta^2 ||x_{n(t)}||^2 = ||w_t||^2 + 1 \end{aligned} \tag{2}$$

We can then expand the equation $\| w_{t+1} \|^2 = ||w_t||^2 + 1$ to $\| w_T \|^2 = ||w_0||^2 + T$. Therefore $\| w_T \| \leq \sqrt{T}$. And since we know that $\frac{w_f^T w_T}{\|w_f\|\|w_T\|} \leq 1 \Longrightarrow \frac{w_f^T w_T}{\|w_f^T\|\|w_T\|} \geq \frac{\|w_f\|\hat{\rho}T}{\|w_f\|\sqrt{T}} = \sqrt{T}\hat{\rho}$ Eventually we get $\Longrightarrow T \leq \frac{1}{\hat{\rho}^2} \Longrightarrow \mathbf{p = 2}$ (Option C) $\square$

**Question 5:** Answer: (d)
Building onto the previous question's update formula, we can derive the following.

$$\begin{aligned} y_{n(t)} w_{t+1}^T x_{n(t)} &= y_{n(t)}(w_t + \eta_t y_{n(t)} x_{n(t)}) x_{n(t)} > 0 \quad \text{(what we hope for)} \\ &= y_{n(t)} w_t^T x_{n(t)} + \eta_t ||y_{n(t)}||^2 ||x_{n(t)}||^2 > 0 \\ &= y_{n(t)} w_t^T x_{n(t)} + \eta_t ||x_{n(t)}||^2 > 0 \end{aligned} \tag{3}$$

We can now learn that $\eta_t > \frac{-y_{n(t)} w_t^T x_{n(t)}}{||x_{n(t)}||^2}$. Thus the answer is (d).

**Question 6:** Answer: (c)
This question demands a reasonable choice of learning rate $\eta_t$, which has to be between 0 and 1 (inclusive) given the normalized inner product result. There are 3 results that fulfills our requirements: 0.1126, option (c)'s, and option (d)'s.

## C. Types of Learning

**Question 7:** Answer: (e)
Reinforcement learning is when the agent trials, and get feedback (both reward of penalty) from the environment. AlphaGo, AlphaStar are all this kind of applications.

**Question 8:** Answer: (b)
This is trivially a structured learning since we want ML algorithm to "imitate" human actions. Raw features for the video data that requires more manual processing before feeding into a ML algorithm. Semi-supervising in a way that 100 hours data are more sufficiently labeled with human action cues, while the other 100 do not. Batch learning is accurate description since data are fed in "batch" rather than sequentially "online". No agent presents nor are we maximizing any reward function so definitely not RL.

## D. Off Training-Set Error

**Question 9:** Answer: (e)
The best case is trivial: it is almost always possible that one gets lucky and comes up with a decision boundary in training set that perfect matches testing data. Thus, the best $E_{ots}(g) = 0$. The worst case happens when PLA selects a bad set of data, in which case the line generated can fail to predict all testing data (i.e. $\mathcal{U} \setminus \mathcal{D}$). The worst $E_{ots}(g)$ is then 1.

## E. Hoeffding Inequality

**Question 10:** Answer: (b)
Fit the options into Hoeffding and the answer is obvious. Hoeffding's inequality states the following.

$$Pr[|\frac{1}{n}\sum_{i=1}^{n} Z_i - E[z]| > \epsilon] \leq 2e^{-2N\epsilon^2}$$

According to the instruction, we want the *LHS* above to be $\delta$. We can then derive as the following. Learning source: *UMass Statistical ML notes in 2010.*

$$\delta \geq 2e^{-2N\epsilon^2}$$

$$\log\frac{\delta}{2} \geq -2N\epsilon^2$$

$$N \geq \frac{1}{2\epsilon^2}\log\frac{2}{\delta}$$

## F. Bad Data

**Question 11:** Answer: (c)
Each example is an i.i.d. random variable. To fulfill the instruction's requirement between $h_2$ and f(x), $x_1$'s sign has to be the same as $x_2$, which has 50% of chance. Five samples will make $\frac{1}{32}$.

**Question 12:** Answer: *None*

**Question 13:** Answer: (b)
The union bound for this question can be calculated with an almost identical approach as in the lecture slide. However, notice that $\mathcal{H}$ set actually contains $d$ complementary pairs, meaning that within each pair, if on hypothesis is correct, the other is guaranteed wrong. The pairs are mutually exclusive, and collectively exhaustive. Thus we now have a probability of $d * 2e^{-2N\epsilon^2}$ to satisfy the inequality.

## G. Multiple Sampling

**Question 14:** Answer: (d)
Given that the dices are drawn with equal probability and each dice is an i.i.d. RV, we discover the following: Option A never happens. Option B happens when Dice C is drawn. Option C happens when Dice A, B, D is drawn. Option D (five green 4's) happens when Dice A and B are drawn, which is the same probability as having green threes.

**Question 15:** Answer: (c)
The cases can actually be simplified into case *With C* and case *Without C*. If C presents, then 6 has to be green, which leaves us with Dice A. If C is not present, 2 is always green for Dice A, B, and D. Thus the solution is:

$$\frac{30}{4^5} + \frac{3^5}{4^5} = \frac{274}{1024}$$

## H. Experiments with Perceptron Learning Algorithm

Answers to question **(16) - (20)** are in the IPython Notebook. In short, the answer is (b), (b), (c), (d), (d) in such order.