CS449/549 Computational Learning Quiz 2 **Solution** (Fall 2023).

Part 1: True/False Circle one. Provide short explanations to your answers.

1. (T/F) When a mistake is made, the single chosen expert in Randomized Weighted Majority is one of the mistaken experts in standard Weighted Majority.
   TRUE: the chosen expert made a mistake so it has to be in the majority.

2. (T/F) If there is only one best expert for the entire duration, Drifting Weighted Majority algorithm has the same mistake bound guarantee as the standard Weighted Majority.
   FALSE: DWM imposes a threshold minimum for the weight updates which alters the mistake bound.

3. (T/F) The AdaBoost algorithm will crash if the weak learner is perfect.
   TRUE: if $\epsilon_t = 0$ then $\alpha_t = \infty$.

4. On a sample $S \subseteq \mathbb{R}^d \times \{-1, +1\}$ of size M, the dual Perceptron algorithm finds an integer vector $\vec{m}$ so that

$$\sum_{j=1}^{M} m_j y_i y_j \langle \vec{x}_i, \vec{x}_j \rangle \geq 0, \quad \text{for all } i = 1, \ldots, d$$

   (T/F) The integer vector $\vec{m}$ only contains positive entries.
   FALSE: it may contain zero entries.

5. Consider a rectangular neural network (NN) over n inputs with m layers where each layer has n units (including the output layer). Each layer k is a matrix $M_k \in \mathbb{R}^{n \times n}$ whose jth row represents the weight vector of unit j (in layer k).
   (T/F) The forward propagation be expressed as a sequence of matrix-vector $\vec{z} = M_k \vec{y}$ products interspersed with entry-wise application of some activation function $\vec{y} = \mathtt{map}(\sigma, \vec{z})$.
   TRUE: The inner product performed by each unit can be viewed as the inner product of a row vector (of weights) with a column vector (of inputs). If we group all the row weight vectors into a matrix, this yields a matrix-vector product between a matrix of row weights with a column vector (of inputs from the previous layer). To produce the outputs of the current layer, we need to map the activation function to each entry of the matrix-vector product.

Part 2: Boosting

1. (T/F) After T rounds, AdaBoost returns a final hypothesis

$$H(x) = \mathtt{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

   where $h_t(x)$ is the weak hypothesis obtained in round t and $\alpha_t \in \mathbb{R}$ are real coefficients.
   TRUE:

2. (T/F) Assuming ± 1 classification, the coefficients α_t satisfy

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

   where $\epsilon_t = \Pr_{D_t}[h_t(x) \neq f(x)]$ is the error of the weak hypothesis h_t with respect to distribution D_t over the training sample.
   TRUE: if $\gamma_t > 0$ is the advantage of the weak hypothesis from random guessing, then $\epsilon_t = \frac{1}{2} - \gamma_t$, and thus

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 + 2\gamma_t}{1 - 2\gamma_t} \right) = \frac{1}{2} \ln \left(\frac{\frac{1}{2} + \gamma_t}{\frac{1}{2} - \gamma_t} \right) = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

Part 3: Dual Perceptron Consider the following implementation of Vapnik's trick.

```
/* x[i] = i-th feature vector, y[i] = (-1/+1) label of x[i], k(x,z) is a kernel map */
m = zero_vector(num_points)
Oops = True /* mistake alert */
while Oops:
  Oops = False
  for i in range(num_points):
    if y[i] * sum(m[j]*y[j]*k(x[i],x[j]) for j in range(num_points)) < 0:
      m[i] = m[i] + 1
      Oops = True
return h(z) = (lambda x: sum(m[j]*y[j]*k(z,x[j]) for j in range(num_points)))
```

1. (T/F) The above algorithm correctly generalizes the dual Perceptron algorithm.
   FALSE: variable x should be z. Missing `sign` *function within the lambda expression.*

2. (T/F) If $k(x, z) = \langle x, z\rangle$, we recover the Perceptron algorithm.
   TRUE: assuming the algorithm is correct.