9.641 Redes neuronales
Boletín de problemas 4

1. Considere la clase de funciones definidas sobre R^2, que comprende la zona interior de los triángulos. Dicho de otro modo, a cada triángulo T le corresponde una función f_T, de modo que $f_T(x)=1$ para $x \in T$, a menos que $f_T(x)=0$.
   a. Demuestre que la dimensión VC de la clase de función es al menos 7. Consejo: tenga en cuenta el caso en el que todos los puntos se disponen en círculo.
   b. (Punto adicional). Demuestre que la dimensión VC de la clase de función es a lo sumo 7.

   Recuerde que la dimensión VC se define como el tamaño del conjunto más grande fragmentado por la clase de función. Con el fin de demostrar que la dimensión VC es al menos *d*, basta con mostrar sólo un conjunto fragmentado de tamaño *d*. Para mostrar que la dimensión VC es a lo sumo *d*, es necesario exponer que ningún conjunto de tamaño d + 1 puede fragmentarse. Por tanto, los límites superiores en la dimensión VC son generalmente más difíciles de verificar que los inferiores.

2. Considere la clase de función parametrizada,
$$f_w(x) = \Theta(\sin(wx)) \tag{1}$$
para $x \in R$, donde la función escalón de Heaviside $\Theta(z)=1$ para z > 0, y 0 para $z \le 0$. El parámetro w puede adoptar todos los valores reales.
Muestre que los puntos:
$$x_1 = 10^{-1}, ..., x_m = 10^{-m} \tag{2}$$
pueden fragmentarse mediante esta clase de función. Consejo: codifique el etiquetado deseado en la expresión base 10 para *w*. Dado que esta construcción funciona para un *m* arbitrariamente grande, la dimensión VC es infinita. Este ejemplo demuestra que la dimensión VC no es siempre equivalente al número de parámetros.

3. La función:
$$C(m,d) = 2\sum_{i=0}^{d-1}\binom{m-1}{i}$$
es el número de dicotomías inducidas por un perceptrón homogéneo de dimensión *d* sobre un conjunto de vectores *m* en posición general.
   a. Escriba código MATLAB para calcular $C(m,d)$. (Consejo: si lo desea, puede utilizar la función `nchoosek`).

b. Trace $C(m,d)/2^m$ como una función de *m*/*d* para *d* = 5, 10 y 20. Supongamos que *m*/*d* oscila entre 0 a 4 y se superpone a todos los gráficos. Debería observar que el gradiente en *m*/*d* = 2 se hace más pronunciado a medida que *d* aumenta.

En el límite $d \to \infty$, la transición en *m*/*d* = 2 se hace marcada. Esta es la razón por la que *m* = 2*d* se contempla normalmente como la capacidad del perceptrón.

4. Facilitamos en clase la fórmula para $C(m,d)$ sin prueba. En este ejercicio, usted tendrá que derivarla para el caso especial *d* = 2.

a. Demuestre que la fórmula anterior implica $C(m,2) = 2m$.
b. Imagine que no conoce la fórmula para $C(m,d)$. Demuestre que $C(m,2) = 2m$, argumentándolo a partir de los principios primeros.

Este es un caso especial de un hecho que se discutió en clase, en el que se exponía que $C(m,d)$ aumenta como m^d para *m* grande. Según la teoría VC, dicho crecimiento subexponencial es importante para la generalización.

5. La probabilidad de que los ejemplos aleatorios *m* puedan ser separados por un perceptrón homogéneo, viene dada por $C(m,d)/2^m$. Compare esta fórmula con la simulación numérica del aprendizaje del perceptrón para *d* = 5.

a. Escriba el código MATLAB para simular la regla de aprendizaje del perceptrón. Supongamos que éste está determinado por sign $(w \cdot x)$, de modo que su salida es ± 1. La entrada al código debería ser un conjunto de ejemplos *m*, donde cada ejemplo es un vector de entrada *x* de dimensión *d*, y una salida deseada $y = \pm 1$. La regla de aprendizaje es $\Delta w = yx$ cuando el perceptrón comete un error en el ejemplo, y $\Delta w = 0$ cuando el perceptrón es correcto.
Escriba su código como dos bucles anidados. El bucle interior completa un ciclo al dar una vuelta por todos los ejemplos *m* del grupo de entrenamiento. Este bucle interior debería hacer un seguimiento del número de errores cometidos durante dicho ciclo. El bucle exterior gira alrededor de los ciclos. Finaliza si en un ciclo se obtiene error cero. También acaba si no se obtiene error cero en 1000 ciclos.

b. Genere vectores de entrada *m* en posición general, trazándolos aleatoriamente. A continuación, haga una estimación de $C(m,d)$, ejecutando el código anterior sobre todos los posibles etiquetados 2^m de estos vectores. Si el código finaliza pronto, el etiquetado se realiza mediante un perceptrón. Si el código no acaba hasta pasados 1000 ciclos, se concluye sin llevar a cabo el etiquetado. El cómputo de los etiquetados que puede realizarse es $C(m,d)$.

Calcule $C(m,d)\big/2^m$ para m = 6, 8 y 10, y trace los resultados junto con el grafo de $C(m,d)\big/2^m$. Tendría que observar una óptima concordancia. ¿Espera que esto sea una sobreestimación o una subestimación? Razone su respuesta.