

Cuestionario 2

Luis Suárez Lloréns

Cuestión 1.

Respuesta. a) La clase \mathcal{H}^1 tiene un sólo punto que vale 1, y el resto valen 0. Por tanto, es la elección de dicho punto es el parámetro a seleccionar. Por tanto, el número de funciones en \mathcal{H}^1 es igual al número de puntos. El número de puntos es $2^{10} = 1024$.

La clase \mathcal{H}^{100} tiene por parámetros 100 puntos del espacio. Tenemos que elegir 100 puntos, sin repetición, y sin importar el orden. Por tanto, el número de datos es $\binom{1024}{100}$. Por tanto, tenemos 7.75×10^{140} funciones distintas.

b) Simplemente necesitamos almacenar el punto en el que vale 1. Por tanto son 10 bits.

c) De igual manera, necesitamos almacenar los 100 puntos. Cada punto necesita para ser almacenado 10 bits, luego necesitamos 1000 bits.

Podemos ver que un espacio de funciones más complejo, no sólo el número de funciones disponibles es mayor, si no que también el de sus componentes. Esto es lógico, pues el número de parámetros crece, y esto hace crecer tanto el número de posibilidades de la clase de funciones como la complejidad de representación de una posible solución, pues necesita dar valor a una variable más.

Cuestión 2.

Respuesta. a) Hay 5 partidos, y para cada partido hay dos resultados posibles, ganar o perder. Luego tenemos 2^5 opciones, es decir 32 predicciones distintas.

b) Evidentemente, si hay 32 opciones, puede enviar a 32 personas distintas una predicción distinta a cada una. Así, al terminar el proceso, una y sólo una de las 32 personas habrá recibido las 5 predicciones correctas.

c) Tendrá que enviar la carta a la mitad de las personas, pues habrá enviado 16 "gana el equipo 1" y 16 "gana el equipo 2".

d) Habrá enviado 32 cartas la primera semana, 16 la segunda, 8 la tercera, 4 la cuarta y 2 la última semana. En total 62 cartas distintas.

e) Simplemente tenemos que realizar la siguiente operación:

$$50000 - 0.5 \times 62 = 49969$$

Se obtendrían 49969 €.

f) TODO

Cuestión 3.

Respuesta. TODO

Cuestión 4.

Respuesta. Este procedimiento tiene, de base, un gran problema. Observar los datos contamina de manera total el procedimiento, pues nos lleva directamente a elegir un procedimiento y un conjunto de funciones, en este caso lineales. Esto es incorrecto, pues ya estamos introduciendo conocimiento del problema.

Dado esto, hace que no tenga sentido la parte de obtener una cota del error, pues el error en el procedimiento cometido al seleccionar la función o el conjunto de funciones también rompe el concepto de la cota.

Cuestión 5.

Respuesta. TODO

Cuestión 6.

Respuesta. Este procedimiento tiene varios errores.

Por un lado, seleccionar el k para el que el error en la muestra E_{in} es mínima, va a seleccionar siempre $k = 1$. Esto es lógico, pues con $k = 1$ seleccionamos siempre como clase el más cercano, lo que dentro de la muestra es siempre es el mismo punto que estás evaluando, luego el error E_{in} va a ser 0.

Podría pasar que de verdad $k = 1$ sea la mejor elección, pero sabemos que no tiene porque serlo. Es más, $k = 1$ realiza un gran sobreaprendizaje de la muestra lo que, como sabemos, reduce enormemente la capacidad de generalización.

Por otro lado, tenemos que no es adecuado de todas formas seleccionar la cota para N funciones. Esto se debe a que cada una de las hipótesis —cada uno de los distintos k — hay infinitas funciones en esa clase, y no una sola como indica el ejercicio. Es más, para $k = 1$, podemos clasificar perfectamente cualquier conjunto de puntos, por lo que hemos visto antes. Esto nos indica que la cota de Vapnik-Chervonenskis es infinita. Es claro que una sola función no puede hacer eso. Luego cada k nos da un conjunto de funciones, y no se puede realizar la cota que se nos indica.

Cuestión 7.

Respuesta. TODO

Cuestión 8.

Respuesta. Sabemos que la tangente hiperbólica lleva los valores muy grandes en valores que son casi 1 o -1. Por tanto, la idea que perseguimos es convertir el número que obtengamos en la operación $x_n^T w_1$ en un número lo suficientemente grande como para quedarse a un ϵ del valor correspondiente, 1 o -1.

Por tanto, tomamos $w_2 = w_1 \alpha$. Por simplificar la notación, llamamos β a $x_n^T w_1$. Entonces la ecuación queda de la siguiente forma:

$$|\text{sign}(\beta) - \tanh(\beta \alpha)| \leq \epsilon$$

Tanto si $\text{sign}(\beta)$ es positivo como si es negativo, el problema es el mismo. Tenemos que encontrar un valor de α que haga el producto lo suficientemente grande. Pero β es una constante, luego siempre podemos encontrar un número α que haga el producto lo suficientemente grande para que se cumpla la inecuación.

Cuestión 9.

Respuesta. TODO

Cuestión 10.

Respuesta. TODO: Pasar la derivada.

Ahora que tenemos calculado el gradiente, podemos seguir con el siguiente apartado. En él, se nos pregunta que pasa con la expresión del gradiente si $w \rightarrow \infty$.

Obtenemos que se calcula la tangente hiperbólica de un número que también tiende a infinito, y esa tangente hiperbólica tiende a 1 o a -1. Entonces el término $1 - \tanh(x_n^T w)^2$ tiende a 0. Entonces el producto tiende a 0 y la sumatoria también.

Obtenemos entonces lo siguiente:

$$\text{Si } w \rightarrow \infty, \nabla E_{in} \rightarrow 0$$

Esto es un gran problema. Las redes neuronales utilizan el descenso del gradiente o alguna variante del mismo para aprender los pesos. Pero si el gradiente tiende a 0, la variación de los pesos también tendería a 0, y el método se atascaría y no podría aprender.

Si a esto le añadimos que, dado al "back-propagation", el resto de las derivadas que dependen de esta, irán a 0 también. Luego no sólo una capa queda parada en el aprendizaje, si no todas. Por tanto, este problema puede bloquear totalmente el aprendizaje de una red neuronal.