Pontificia Universidad Católica de Chile
Escuela de Ingeniería, Depto. de Ciencia de la Computación
IIC-2612 Inteligencia Artificial

Noviembre 2012

Interrogación 3

Tiempo: 100 minutos, SIN APUNTES

Nombre:_____

1. (*20 puntos*) Responda las siguientes preguntas fundamentando brevemente su respuesta.

a. En el caso de un problema de clasificación con una variable de salida binaria, ¿Cuál es el mayor error que se puede obtener en el set de entrenamiento independientemente de cual sea la complejidad de este set?.

SOL: 50 %.

b. Construya un set de datos que, no importando el tipo de clasificador, siempre alcance el error máximo indicado anteriormente. Por simplicidad, limite su respuesta a un set de datos con a lo más 5 registros de entrenamiento y no más de 2 atributos.

SOL: Cualquier set ruidoso, en el cual para cada registro exista otro registro con el mismo valor en los atributos pero distinta clase. Por ejemplo, acá 1 posible caso:

A	B	Class
0	0	0
0	0	1
1	1	0
1	1	1

c. ¿Cuál es la diferencia principal entre clasificación y predicción (regresión)?

SOL: En clasificación la variable objetivo es categórica o discreta, mientras que en predicción es continua.

d. Para un problema de clasificación, fundamente si la superficie de error que desciende el algoritmo de backpropagation cambia o no cambia si se modifica el set de entrenamiento.

SOL: cambia pues la forma de la función de error a optimizar se ve afectada.

e. Indique algún caso en que el método batch es preferible al método incremental al entrenar una red neuronal.

SOL: cuando hay pocos registros es preferible seguir en forma exacta la dirección del gradiente de la función de error, por tanto, es preferible usar el método batch (pasos no tan rápidos como el método incremental pero en la dirección localmente óptima).

f. Indique algún caso en que el método incremental es preferible al método batch al entrenar una red neuronal.

SOL: cuando el set de entrenamiento es muy grande, usar el método batch puede tomar mucho tiempo en cada actualización de los pesos, en esos casos el método incremental es más efectivo (pasos rápidos pero no necesariamente en la dirección localmente óptima).

g. Si un set de datos tiene R registros entonces la máxima profundidad alcanzada por un árbol de decisión debe ser siempre menor a $(1 + log_2 R)$.

SOL: No pues el árbol puede estar desbalanceado y tener ramas de mayor profundidad.

h. En una red neuronal al aumentar el número de neuronas de la capa oculta disminuye la posibilidad de sufrir problemas de sobreajuste.

SOL: Falso, aumenta el número de parámetros, por tanto, el espacio de hipótesis y la posibilidad de sufrir problemas de sobreajuste.

i. En un árbol de decisión cada rama del árbol debe clasificar correctamente al menos 1 registro del set de entrenamiento.

SOL: No necesariamente, por ejemplo, considere un árbol de un sólo nodo en que el atributo correspondiente no toma alguno de los valores posibles. Claramente la rama correspondiente a ese valor no recibirá registros de entrenamiento.

j. Explique por qué se considera a un árbol de decisión como una técnica de subespacios de clasificación.

SOL: La clasificación de cada registro depende de los atributos usados en la rama correspondiente. Estos atributos conforman un subconjunto o subespacio de los atributos disponible. Por tanto, cada rama define un subespacio de clasificación.

k. Dada cierta estructura con capa oculta y cierto set de entrenamiento, el algoritmo de backpropagation permite siempre encontrar el mejor valor posible para los pesos de la red neuronal resultante.

SOL: Falso, backpropagation sólo garantiza encontrar un óptimo local de la función objetivo.

l. Explique por qué es importante usar un set de validación al podar un árbol de decisión.

SOL: El podaje debe ser con un set distinto al de entrenamiento, pues el árbol ya esta optimizado para este último (nada es podable). También debe ser con un set distinto al de test, pues para medir capacidad de generalización el set de test debe contener datos no usados en el ajuste del modelo. De esta forma, se necesita podar con un tercer set de datos, que como vimos en clase se denomida set de validación.

m. Explique en que caso sería relevante usar la métrica SplitInInformation en lugar de ganancia de información para construir un árbol de decisión.

SOL: En ninguno pues el SplitInInformation no mide homogeneidad de los subgrupos formados en cada nodo de decisión.

n. Explique por qué es más recomendable aplicar un factor de momemtum al entrenar una red neuronal con el método incremental que con el método batch.

SOL: Para cada actualización de pesos, el método incremental no garantiza seguir la dirección del gradiente de la función de error , por tanto, un factor de momemtum es más relevante.

ñ. Indique ¿Cómo podría usar un set de validación para evitar el overfitting en una red neuronal?

SOL: Este set puede ser usado para determinar el momento en que el error de predicción comienza a aumentar en un set distinto al de entrenamiento.

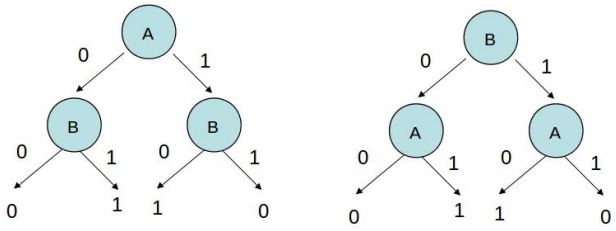
2. (*16 puntos*) Árboles de decisión

Considere el siguiente set de entrenamiento:

A	B	C	Class
1	1	0	0
1	0	1	1
0	1	1	1
0	0	1	0

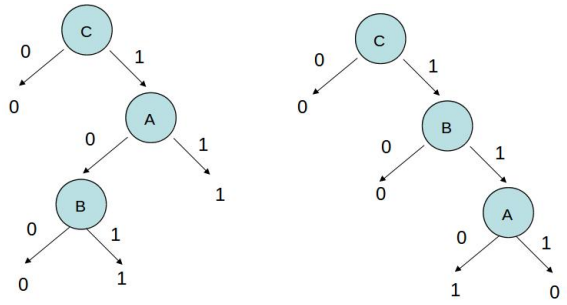
a. (*4 pts*) Encuentre un árbol de decisión de profundidad mínima que permita clasificar estos registros sin error.

SOL: el concepto subyacente a estos datos en un or-exlusivo (A or-ex B). Cualquiera de los siguientes árboles representa un solución de mínima profundidad.



b. (*4 pts*) Para este set de entrenamiento determine el árbol de decisión encontrado por el algoritmo ID3 visto en clases.

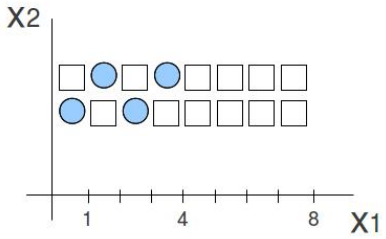
SOL: a profundidad 2 atributos A y B tienen la misma GI, por tanto, cualquiera de los siguientes árboles representa la solución de ID3.



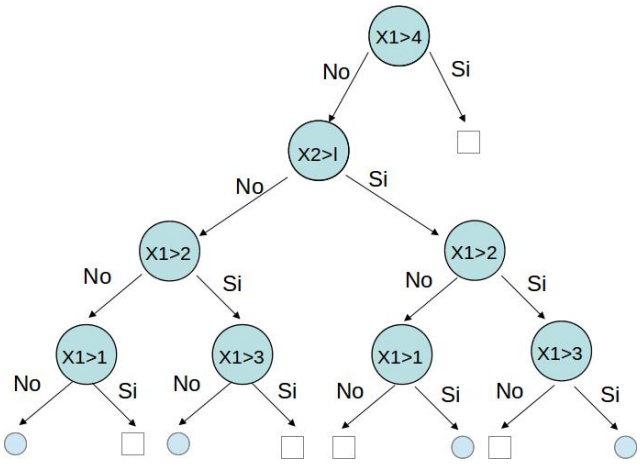
c. (*4 pts*) Comente la principal razón por la cual los árboles resultantes en las preguntas anteriores son iguales o distintos.

SOL: El algoritmo ID3 utiliza una búsqueda codiciosa (tipo greedy) que no garantiza soluciones de menor profundidad. Por ejemplo, en el caso anterior ID3 determina una rama en que se realiza la conjunción de 3 atributos, aunque según el árbol de la parte a) es posible encontrar una solución en que conjunciones de sólo 2 atributos son suficientes para clasificar correctamente todos los datos.

d. (*4 pts*) Indique si es posible construir un árbol de decisión que permita clasificar sin errores el siguiente set de datos. Indique la forma de este árbol o en caso contrario fundamente por qué no es posible lograr este objetivo.



SOL: existen varias posibles soluciones, lo importante es que el árbol clasifique correctamente todos los registros. Lo más directo es dividir primero según eje x1=4, lo cual permite clasificar correctamente la mitad de los puntos. Para la mitad restante es necesario ir generando subnodos hasta llegar a nodos 8 hoja, tal como en el caso del siguiente árbol (l indica el valor de x2 que divide los datos en 2 grupos de igual número de datos):



3. (*16 puntos*) Redes Neuronales

a. (*4 pts*) En un perceptron se tiene una unidad cuya salida es $w_0 + w_1(x_1 + 1) + w_2(x_2^2)$. Indique las expresiones para actualización de pesos de esta unidad según la técnica de descenso del gradiente y el método de actualización tipo batch. Asuma que el set de entradas a esta unidad está dado por n ejemplos (registros) de entrenamiento (x_1^i, x_2^i, t_i), donde t_i indica la salida deseada (clase) e $i \in [1, ..., n]$.

SOL:
El error esta dado por:

$$E = \frac{1}{2}\sum_i (t_i - o_i)^2, \quad (1)$$

donde: (2)

$$o_i = w_0 + w_1(x_1^i + 1) + w_2(x_2^i)^2 \quad (3)$$

Por tanto las expresiones de actualización de pesos son:

$$w_0^t = w_0^{t-1} + \eta \sum_i (t_i - o_i) \quad (4)$$

$$w_1^t = w_1^{t-1} + \eta \sum_i (t_i - o_i)(x_1^i + 1) \quad (5)$$

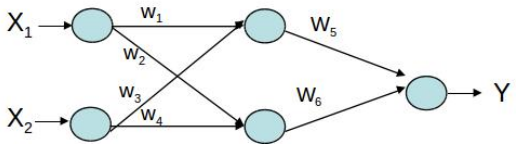
$$w_2^t = w_2^{t-1} + \eta \sum_i (t_i - o_i)(x_2^i)^2 \quad (6)$$

(7)

b. (*4 pts*) ¿ Cuál es la complejidad computacional de la actualización de los pesos de esta unidad usando el método batch ?

SOL: En términos de complejidad computacional, para todos los pesos el término principal es la sumatoria, la cual depende linealmente en el número de ejemplos de entrenamiento. Por tanto la complejidad es O(n).

c. (*4 pts*) Considere la siguiente red neuronal cuyas unidades tienen una función de activación lineal, o sea la salida de cada neurona es una combinación lineal de sus entradas.

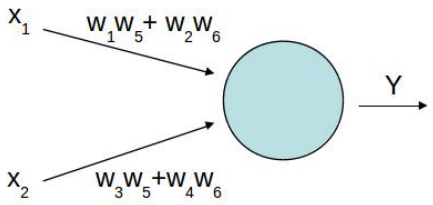


Indique si posible que cada hipótesis de esta red pueda ser equivalentemente representada por una red compuesta de sólo una unidad (perceptron). Si es así, indique una posible configuración de esta unidad (perceptron equivalente), detallando los pesos y función de activación equivalentes. Por el contrario, si la equivalencia no es posible, indique la razón.

SOL: Sí, pues una combinación lineal de funciones lineales es una función lineal. En este caso, la red original es equivalente a

$Y = w_5(w_1x_1 + w_3x_2) + w_6(w_2x_1 + w_4x_2)$

Una unidad equivalente es:
$Y = (w_1w_5 + w_2w_6)x_1 + (w_3w_5 + w_4w_6)x_2$



d. (*4 pts*) Considere el caso de redes neuronales con funciones de activación lineal o tipo switch (función signo), es decir:

- Lineal : $y = w_0 + \sum_i w_i x_i$
- Switch :

$$sng(x) = \begin{cases} 1 \text{ if } w_0 + \sum_i w_i x_i \geq 0 \\ 0 \text{ en caso contrario} \end{cases}$$

¿ Cuál(es) de las siguientes funciones puede(n) ser representada(s) en forma exacta por una red neuronal compuesta de una capa oculta y unidades con función de activación lineal y/o tipo switch ?

- Polinomios de grado 1.

  SOL: Una combinación lineal de funciones lineales es también una función lineal por ende cualquier función lineal puede ser representada con una red de capa oculta y funciones lineales de activación.

- Polinomios de grado 2.

  SOL: No es posible pues el término cuadrático no se puede modelar en forma exacta con funciones lineales o de tipo switch.

- Funciones constantes a trozos (piecewise constant functions).

  SOL: Si, una posibilidad es usar funciones tipo switch en la capa interna y una combinación lineal de estas salidas en la capa externa.