Resolución Hoja de Ejercicios y Problemas tipo examen

Ismael Sagredo Olivenza

Problema 01

del dia del año (entero), la humedad (real), la cantidad de nubosidad (baja, media, alta), la temperatura (real) y la presión atmosférica (real).

- KNN: K=5, 88% distancia euclídea.
- Decision Tree: 90%, profundidad del árbol 5.
- Random Forest: 98%
- Perceptrón Multicapa: 10 neuronas ocultas en la primera capa y 5 en la segunda, factor de aprendiszaje de 0,2, valores normalizados 0-1: 90%

1. ¿ Qué podriamos hacer para mejorar los resultados del perceptrón?

El factor de aprendizaje quizás esté muy alto, podríamos reentrenar la red con otro factor de aprendizaje. Tambien podríamos modificar la arquitectura de la red o seleccionar los atributos

2. Con los datos del ejercicio, razona cual sería el modelo que elegirías.

La respuesta peude ser variada y si está argumentada sería correcta. Por un lado podemos pensar que un 90% es suficiente y elegir arboles de decisión por ser un algoritmo de caja blanca. Tenemos que tener en cuenta que saber si va a llover o no es algo que tiene importancia para hoteles o para saber si te vas de vacaciones o no, pero si 9 de cada 10 veces se va a acertar, no es un mal rendimiento. Si lo que queremos es maximizar la precisión usaríamos random forest, pero tenemos que ser conscientes que perdemos explicabilidad.

3. En KNN el valor categórico ha sido codificado como 0, 1, 2 y el resto de atributos para calcular la diferencia se realizan restando sus valores sin ninguna transformación. ¿Cómo podríamos mejorar KNN para conseguir mejores resultados?

Podriamos aplicar one-hot-enconding para codificar los valores categóricos. Tambien estaria bien normalizar el resto de valores. Bien aplicando una resta de la normal y dividiendo por la desviación típica o bien haciendo que todos los datos esten entre 0 y 1. Esto haría que la distancia euclídea represente mejor al individuo.

Problema 02

La mejor forma de abordar este problema es usando cross validation. Con esta técnica calculamos el error medio entre diferentes particiones del dataset entre datos de entrenamiento y de validación, lo que minimiza la desviación que puede producir una mala partición del dataset entre entrenamiento y validación.

Problema 03

1. Explica los pasos que harías para preparar los datos

Las imágenes son demasiado grandes habría que escalarlas a un tamaó más manejable, como por ejemplo 4x 64. Podemos escalarlas a mano usando alguna algoritmos de escaldo, o podriamos intantar extrar las caracteristicas de una imagen como etapa de proprocesado inicial.Para ello podemos utilizar un Autoencoder o un algortimo de PCA. Por otro lado usando PCA dibujaría las diferentes clases para ver que pinta tiene. Tambien se podría hacer un análisis de las imágenes y detectar que pixeles aportan ma´s información y hasta que punto podemos recortar las imágenes.

2. A priori, cual crees que sería el mejor modelo y porqué.

A priori el mejor modelo para procesar imágenes de lso que hemos estudiado son las redes de neuronas convolucionales y lso autoencoders. Tambien a priori el perceptrón multicapa se comporta mejor en este tipo de problemas que otros algoritmos. Pero no podemos asegurarlo a ciencia cierta hasta hacer pruebas.

3. Suponiendo que se han creado dos modelos y con 2000 datos de validación, el modelo A y el modelo B

El modelo B ya que es más importante que no haya falsos positivos, es decir que habiendo un tumor se clasifique como no tumor. Es peor este resultado que clasificar como tumor algo que no lo tiene, ya que normalmente se le realizaran otras pruebas antes de darle un diagnostico final.

Problema 04

1. ¿Como codificarías los datos para que KNN pueda funcionar correctamente y que medida de distancia usarías?

- La cuadrícula con piezas seria una matriz de entrada donde valdría 1 la casilla X,Y esta ocupada por una pieza y 0 en caso contrario
- La posición de la pieza actual se guardará como un valor real con el origen de coordinadas de (0,0) en la esquina inferior izquierda de la parte jugable y que ira de 0 a 1.
- Usamos One-hot-encoding para guardar el identificador de la pieza guardada, la actual y la próxima ya que la categoria no es ordinal
- Para saber el nivel tambien codificaremos este número como número real entre 0-1 seindo 0 el nivel 0 y 1 el nivel más alto posible.
- como medida de similitud usaría la distancia de Manhattan ya que en este tipo de problemas las diagonales no son importantes a priori (en el tetris no te puedes mover en diagonal)
- Tendría una ventana temporal de acciones realizadas previamente ya que hay una dependencia entre las acciones realizadas anteriormente y las nuevas.

2. Para saber si el agente imita correctamente al jugador, comprobamos que la secuencia de acciones realizadas por el agente sea similar a la del jugador para un

estado de juego dado (El Tetris es determinista asumiendo la misma semilla aleatoria) De esta forma tendremos la siguiente secuencia de acciones:

MI|MD|R|B|MI|MI|B|G|MD|MI|MD|R|R|B

¿Que medida de distancia utilizarías para calcular el error?

Usaría distancia de edición. La similitud sería el número de cambios que hacen que una cadena sea igual que la otra. Esta medida es más representativa que una distancia euclídea, primero porque el valor de la clase no es ordinal y por otro lado, porque hay una dependencia temporal de las acciones realizadas. Si queremos imitar a la percepción el comportamiento del humano, debemos realizar las mismas acciones en le mismo orden más allá de que estas sean buenas o malas.

Problema 05

¿Qué atributo elegiría primero el árbol de decisión?

A1	A2	A3	A4	Clase
2	0	3	A	Bueno
1	1	1	B	Malo
1	2	0	B	Bueno
0	2	1	B	Bueno
2	1	0	A	Malo
3	0	1	B	Malo

Para calcular log en base 2 usar esta fórmula

$$Log_a N = \frac{Log_b N}{Log_b a} \forall b$$

por ejemplo Log_{10}

A1= {0,1,2,3} = $\frac{1}{6} I_{10} + \frac{2}{6} I_{11} + \frac{2}{6} I_{12} + \frac{1}{6} I_{13}$

Para el atributo A1, para la clase (Bueno) tenemos un 1. Para la clase (malo) tenemos 0

$$I_{10} = -\frac{1}{1} log_2 \frac{1}{1} - \frac{0}{1} log_2 \frac{0}{1} = -1 * 0 + 0 = 0$$

Para el valor de 1 tenemos uno de la clase malo y otro de la clase bueno $I_{11} = -\frac{1}{2} log_2 \frac{1}{2} - \frac{1}{2} log_2 \frac{1}{2} = -0.5 * -1 - 0.5 * -1 = -0.5 * -1 - 0.5 * -1 = 0.5 + 0.5 = 1$

Para el valor de 2 tenemos uno de la clase malo y otro de la clase bueno $I_{12} = -\frac{1}{2} log_2 \frac{1}{2} - \frac{1}{2} log_2 \frac{1}{2} = -0.5 * -1 - 0.5 * -1 = -0.5 * -1 - 0.5 * -1 = 0.5 + 0.5 = 1$

$$I_{13} = -\frac{1}{1} log_2 \frac{1}{1} - \frac{0}{1} log_2 \frac{0}{1} = -1 * 0 + 0 = 0$$

$$\text{A1} = \frac{1}{6}(0) + \frac{2}{6}1 + \frac{2}{6}1 + \frac{1}{6}0 = \frac{2}{6} + \frac{2}{6} = \frac{4}{6} = 0.66$$

$$\text{A2} = 0.33 = \frac{2}{6} I_{20} + \frac{2}{6} I_{21} + \frac{2}{6} I_{22} = \frac{2}{6}(1) + \frac{2}{6}(0) + \frac{2}{6}(0)$$

$$\text{A3} = 0.795 = \frac{2}{6} I_{30} + \frac{3}{6} I_{31} + \frac{1}{6} I_{33} = \frac{2}{6}(1) + \frac{3}{6}(0.924) + \frac{1}{6}(0) = 0.33 + 0.462 =$$

$$\text{A4} = 1 = \frac{2}{6} I_{4A} + \frac{4}{6} I_{4B} = \frac{2}{6}(1) + \frac{4}{6}(1) = 0.5 + 0.5 = 1$$

El mejor sería A2

Problema 06

- Escalar la imagen usando un algoritmo de escalado: método conocido y rápido desventaja, no permite extraer información más allá de concentrar la información en menos pixeles
- PCA:
- Autoencoder: Permite extraer un resumen de la red más complejo y más rico, pero se necesita un gran volumen de datos para entrenar
- Convolutional networks: permite detectar características concretas dentro de una imagen. Solo detecta una característica concreta por cada capa convolucional. no reduce la dimensionalidad tanto como autoenconder y pca

problema 07

Un Perceptrón multicapa con función de activación ReLu ya que no es una función de activación lineal

problema 08

para 5 salidas tendríamos 5 neuronas y la función de activación sería softmax.