

PRAC 6

MULTILAYER NEURAL NETWORKS, TREE CLASSIFIERS

CLP: Clasificación de Patrones.
ETSETB
UPC

Noviembre 2016

1	Objetivos	2
2	Bases de Datos	2
3	Trabajo en el laboratorio	3
3.1	Preparación de la base de datos	3
3.2	Neural networks	3
3.3	Decision trees.....	4
3.4	Clasificación de una imagen nueva.....	4

1 Objetivos

Aplicación de las técnicas de clasificación trabajadas en el tema 4:

- Redes neuronales
- Árboles de decisión

La práctica se realiza en dos sesiones de laboratorio.

2 Bases de Datos

Se utilizarán bases de datos de tipo BRAIN, análogas a la que se utilizó en la práctica 0. Se facilitan dos bases de datos correspondientes a dos cortes horizontales del cerebro diferentes: “PRBB_Brain” y “PRBB_Brain2”. Para cada una de las dos bases de datos se tienen 8 imágenes tal como se explica a continuación.

The BRAIN Database has been provided by Dr Joan Domingo Gispert from the “Institut d’Alta Tecnologia PRBB, Hospital del Mar, Barcelona”: www.prbb.org .

Each database is formed by 8 images. All of them correspond to a single Human Brain Cut.

Images 1,2,3,4: Obtained by Magnetic Resonance (Structural Analysis).

Images 5: Positron emission tomography (PET) (Functional Analysis).

Images 6:8: Probabilities for each pixel to be in one of the three classes:

- White (Materia Blanca) Class 2
- Grey (Materia Gris) Class 1
- LCR (Líquido Cefaloraquídeo) Class 3
- Background, Class 4

Each Image has 256×256 pixels = 65536 = N patterns

Feature vectors are formed taking $d=5$ pixels. Each j -th feature in a vector is obtained from the same horizontal-vertical pixel at the j -th image.

3 Trabajo en el laboratorio

Utilice el fichero prac6_Brain.m.

3.1 Preparación de la base de datos

1. Dado que las imágenes 6,7,8 corresponden a los mapas de probabilidad de cada una de las clases 1, 2 y 3, se utilizan estas imágenes para etiquetar la base de datos y presentar una imagen correspondiente a la base de datos etiquetada. A través de la variable *Pr_min* se decide el nivel de fiabilidad de las etiquetas de la Base de datos etiquetada. Pruebe diferentes valores de *Pr_min*, (por ejemplo 3 casos) entre 0.75 y 1 y observe las gráficas obtenidas comentando la ventajas e inconvenientes que puede generar el utilizar como base de datos de *train* la base obtenida en cada caso. Nótese que puede haber píxeles que no pertenezcan a ninguna de las clases.
2. Elija ***Pr_min* = 0.9** para el resto de la práctica. Se recomienda además desactivar la variable '*i_dib*' para el resto de la práctica.

3.2 Neural networks

1. Mediante la activación de la variable '*i_clas_NN*', realice una clasificación mediante **redes neuronales** con el método de Back-Propagation (BP) (opción `net.trainFcn='traingd'`) generada por defecto. Asegúrese de iterar tantas épocas como sea necesario para permitir si es posible que el algoritmo pare por validación, incrementando si es necesario la variable '`net.trainParam.epochs`' (hasta un máximo de 100000). Importe las figuras de '*performance*' y de '*confusion*' que puede obtener a partir de la '*nntool*' y comente los resultados.
2. Repita el punto anterior mediante el algoritmo Levenberg-Marquadt (opción `net.trainFcn='trainlm'`). Compruebe que este método siempre converge y más rápidamente que el anterior.
3. Compare las imágenes clasificadas en los dos apartados anteriores, así como el número de épocas necesario para alcanzar el mínimo por validación y comente los resultados.
4. A continuación, y con Levenberg-Marquadt se propone que incluya código en el fichero en el apartado preparado mediante la variable de activación '*i_valida_hidden*' para realizar una validación del número de neuronas a utilizar en la capa oculta mediante las etapas indicadas en el propio código. Elija el número de neuronas óptimo a partir del menor error de clasificación medido sobre la BD de validación. Se recomienda hacer un barrido sobre el número de neuronas entre 1 y el doble del número de características, es decir, entre 1 y 10 en este caso. Anote para este caso el error obtenido sobre la BD de test. Incluya el código generado en el documento a entregar.
5. OPCIONAL: Pruebe alguna estructura de doble capa oculta y comente los resultados.

3.3 Decision trees

1. Mediante la activación de la variable 'i_clas_Tree', realice una clasificación mediante **árboles**. Anote errores de clasificación, así como matrices de confusión obtenidas. Importe las figuras de 'árbol obtenido' y comente los resultados.
2. Compare la imagen clasificada con las obtenidas mediante redes neuronales.
3. A continuación, se propone que incluya código en el fichero en el apartado preparado mediante la variable de activación 'i_valida_split' para realizar una validación del número máximo de splits a utilizar al generar el árbol de decisión. La correspondiente variable de entrada se denomina 'MaxNumSplits'. Elija el número máximo de splits óptimo a partir del menor error de clasificación medido sobre la BD de validación. Anote para este caso el error obtenido sobre la BD de test. Incluya el código generado en el documento a entregar.
4. Compare el error de test seleccionado en el apartado anterior con el medido con redes neuronales en el apartado 4 y a la vista de los resultados decida el mejor clasificador a utilizar para la presente aplicación.

3.4 Clasificación de una imagen nueva

Con los dos mejores clasificadores (redes neuronales y árboles de decisión) obtenidos previamente, clasifique la otra base de datos disponible en el laboratorio (*PRBB_Brain2*) y genere la imagen que contenga los píxeles etiquetados. Compare los resultados obtenidos mediante el clasificador de tipo red neuronal y mediante el clasificador de tipo árbol de decisión.

NOTA IMPORTANTE

Entregue un único documento en Atenea G***_apellido1_apellido2_prac*.pdf. El documento debe incluir todos los resultados gráficos y numéricos y explicaciones que se consideren convenientes, es decir, aquellos que aporten información adicional y relevante. Las explicaciones deben apoyarse en los conceptos teóricos vistos en clase.