RED NEURONAL MULTITAREA PARA PROBLEMAS DE DECISIÓN CON INFORMACIÓN INCOMPLETA

Pedro J. García-Laencina, José-Luis Sancho-Gómez
E-mail: {pedroj.garcia, josel.sancho}@upct.es
Dpto. de Tecnologías de la Información y las Comunicaciones.
Universidad Politécnica de Cartagena.

***Abstract*— Missing data is a common problem that appears in many real applications. Handling missing data is a essential requirement for pattern classification because inappropriate treatment of missing data may cause large errors or false results on classification. A classical approach is to estimate and fill the missing values. Up to now, proposed methods are efficient but they do not focus the missing data estimation to pattern classification. In this work, we propose a novel neural network that uses the advantages of Multitask Learning (MTL) to classify patterns with incomplete data. A MTL neural network learns at the same time the classification task and the different task associated to incomplete features. Missing data estimation is guided and oriented by the classification task during the MTL process. Obtained results based on real and artificial classification problems demonstrate the advantages of the proposed algorithm.**

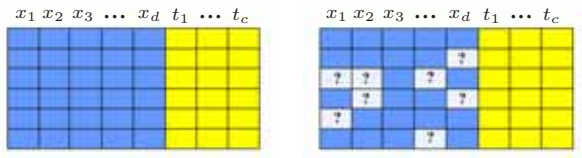
I. INTRODUCCIÓN

Las redes neuronales artificiales (ANNs, *Artificial Neural Networks*) son aplicadas con éxito en una gran variedad de problemas de decisión [1], como la diagnosis médica, teledetección, reconocimiento automático de voz, etc. Un problema muy frecuente en aplicaciones reales es la presencia de información incompleta, lo que se conoce en la literatura como valores perdidos, *Missing Data* [2]. Las razones que provocan la ausencia de información en los patrones de entrada son muy variadas dependiendo del campo de aplicación, desde fallos de un sensor durante el proceso de adquisición de datos hasta señales de voz incompletas debido al ruido [2]. Hasta ahora se han propuesto numerosos trabajos en ANNs que intentan solventar el problema de la información incompleta mediante la estimación de los valores perdidos, pero en ningún caso dicha estimación está orientada a la clasificación de patrones, que es la tarea que realmente se desea resolver.

En este trabajo se propone una nueva red neuronal basada en el aprendizaje multitarea (MTL, *Multitask Learning*) [3] que combina la clasificación de patrones y la imputación (es decir, estimación y asignación) de valores perdidos. En el método propuesto, la imputación de datos es guiada por el proceso de aprendizaje de la tarea de clasificación. El resto del artículo se estructura de la siguiente forma: en la Sección 2, se introduce el problema de los valores perdidos, mostrando posteriormente las distintas alternativas propuestas para solucionar este problema. En la Sección 3 se describe brevemente el concepto de MTL. El método propuesto se analiza en la sección 4. La Sección 5 describe los conjuntos de datos usados para evaluar las prestaciones del método propuesto y muestra los resultados obtenidos. Finalmente, se exponen las conclusiones principales y futuras líneas de trabajo.

II. CLASIFICACIÓN DE PATRONES INCOMPLETOS

Considérese un conjunto de N patrones de entrada, donde el patrón n-ésimo $\mathbf{x}^{(n)}$ está definido por d características o atributos reales, $\mathbf{x}^{(n)} = (x_1^{(n)}, x_2^{(n)}, ..., x_d^{(n)})$. En el caso de aprendizaje supervisado, cada patrón está asociado a una determinada clase, que se conoce como salida deseada $t^{(n)}$. Si existen c posibles clases, se pueden codificar las salidas deseadas usando un codificación $1-de-c$, e.g., si hay cinco clases posibles, $c = 5$, y el patrón n-ésimo pertenece a la tercera, su salida deseada será $\mathbf{t}^{(n)} = (0, 0, 1, 0, 0)$. La figura 1(a) muestra un problema de clasificación donde todas las características son conocidas. Por otro lado, la figura 1(b) muestra un problema de clasificación con patrones incompletos. En este artículo consideraremos que los valores perdidos pueden encontrarse en cualquier característica de cualquier patrón de entrada, y se denotan con el símbolo ?.



(a) Problema de clasificación con patrones completos.

(b) Problema de clasificación con valores incompletos, denotados con el símbolo ?.

Fig. 1. Dos hipotéticos problemas de clasificación de patrones d-dimensionales entre c clases posibles. Primero, en (a), todas las características son conocidas. En cambio, en (b), algunos patrones están incompletos.

La manera más sencilla de tratar con datos perdidos es simplemente eliminarlos y trabajar sólo con los datos completos. Sin embargo, en las bases de datos reales, los valores perdidos pueden aparecer en cualquier característica de cualquier patrón, pudiendo suponer una parte importante del conjunto de datos, por lo que eliminarlos implica una perdida sustancial de información, [2]. Una solución más adecuada es la imputación. La imputación de datos es la etapa en la que los "huecos" correspondientes a los valores perdidos del conjunto de datos son rellenados por valores concretos. Muchos de los trabajos propuestos no orientan la imputación de datos a la clasificación

de patrones, primero obtienen un conjunto completo (mediante imputación a la media, métodos de imputación múltiple,...), y posteriormente, una ANN aprende la tarea de clasificación usando este conjunto [4]. Por otro lado, otras alternativas adaptan la estructura y el aprendizaje de la ANN para ser capaz de tratar con patrones incompletos [5].

III. Aprendizaje Multitarea

Rich Caruana introdujo el concepto de Aprendizaje Multitarea (MTL, *Multitask Learning*), [3]. En este tipo de aprendizaje, la capacidad de generalización de una red entrenada para aprender una determinada tarea (*tarea principal*) mejora al aprenderse simultáneamente junto con distintas tareas relacionadas (*tareas extra o secundarias*), es decir, dicha tarea se aprende mejor que en el caso de aprenderla aislada. Caruana analiza el MTL en perceptrones multicapa (MLP, Multilayer Perceptron). La manera más sencilla de implementarlo consiste en añadir salidas extra para aprender las distintas tareas secundarias, junto con la principal, compartiendo todas las tareas la única capa oculta de la red. El hecho de que un conjunto de neuronas ocultas estén conectadas a las salidas asociadas a las distintas tareas permite que lo que se aprenda en una salida contribuya al aprendizaje del resto.

IV. Red Neuronal Multitarea para Reconocimiento de Patrones Incompletos

Considérese un problema de clasificación entre c clases posibles descrito por N patrones de entrada compuestos por d características. Además, m de las d (con $m \leq d$) presentan valores perdidos, como se muestra en la figura 1(b). Además, definimos el vector $\mathbf{a} = [a_1, a_2, ..., a_m]$, cuyas componentes son los m atributos incompletos. En este problema es necesario resolver dos tipos de tareas: la *tarea principal* de clasificación (problema discreto de decisión), y las *tareas secundarias* de imputación asociadas a las características incompletas (problemas continuos de regresión).

A continuación se presenta una red neuronal multitarea que aprende la tarea de clasificación y las tareas de imputación simultáneamente (ver figura 2). Ésta consta de:

- *Capa de entrada*: Está compuesta por d unidades, asociadas a las características de entrada, y además c entradas extra, asociadas a las salidas deseadas de clasificación. Se emplean entradas lineales.
- *Capa oculta*: Está compuesta por $m + 1$ subredes [6], una subred privada que aprende la tarea principal y m subredes comunes donde cada una de ellas aprenden simultáneamente dos tareas: la tarea principal y la tarea secundaria de imputación asociada. Cada subred puede estar compuesta por un número distinto de neuronas ocultas. El número de neuronas ocultas depende de lo complejo que sea el problema a resolver. La función de activación empleada es la tangente hiperbólica.
- *Capa de salida*: Está compuesta por c unidades de salida para la clasificación, y m unidades asociadas a las distintas características incompletas. Se usan salidas lineales.

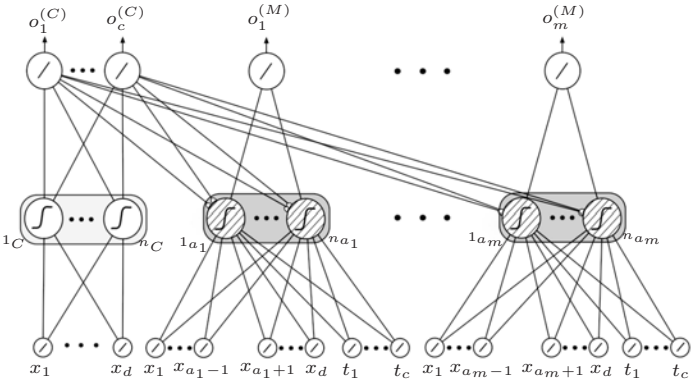
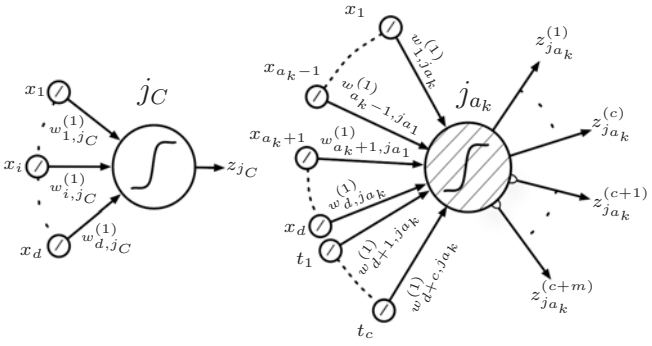


Fig. 2. Red neuronal multitarea que combina la clasificación y la estimación de valores perdidos. Está compuesta por $m+1$ subredes: una subred privada que aprende la tarea principal de clasificación y m subredes comunes, donde cada una de ellas aprende simultáneamente la tarea principal y la tarea de imputación asociada.

En la notación empleada en este trabajo, $w_{i,j}^{(1)}$ denota un peso de la primera capa, que va desde la unidad de entrada i a la neurona oculta j, y $w_{0,j}^{(1)}$ es el sesgo para la neurona j-ésima. La notación es similar en el caso de los pesos de la segunda capa. En la red mostrada en la figura 2 los sesgos no son mostrados con el objetivo de simplificar el esquema. Con respecto a la notación usada para distinguir las neuronas ocultas, en la subred privada que aprende la tarea de clasificación, se usa el subíndice C, y para el resto de subredes comunes, se usa como subíndice la característica incompleta que aprende a_k. En cuanto a las salidas de la red, se emplea un superíndice para distinguirlas: $o^{(C)}$ es una salida de clasificación, y $o^{(M)}$ es una salida asociada al aprendizaje de una tarea de imputación.



(a) Neurona oculta estándar. (b) Neurona oculta MTL.

Fig. 3. Tipos de neuronas ocultas implementadas.

A continuación, se analizan los dos tipos de neuronas ocultas implementadas (ver fígura 3): neuronas estándar, que aprenden una sola tarea, y neuronas MTL, que aprenden varias tareas a la vez. Las neuronas de la subred privada aprenden únicamente la tarea principal, y calculan la suma producto entre sus datos de entrada y los respectivos pesos de entrada. La figura 3(a) muestra una neurona estándar de la subred privada, cuya salida es

$$z_{j_C} = g\left(\sum_{i=1}^{d} w_{i,j_C}^{(1)} x_i + w_{0,j_C}^{(1)}\right) \tag{1}$$

Por otro lado, las neuronas ocultas MTL funcionan de una manera distinta a las propuestas hasta ahora, ya que aprenden en paralelo distintas tareas y producen salidas distintas en función de la unidad de salida a la que va conectada. La figura 3(b) muestra una neurona MTL de la subred común a_k (que aprende la característica a_k, i.e., x_{a_k}). Estas neuronas están conectadas a todas las entradas menos a aquella que tienen que aprender, x_{a_k}. Para explicar esto, supóngase que la subred común a_k estuviese conectada a x_{a_k}. En ese caso existiría conexión directa entrada-salida, y por tanto la salida de imputación $o_k^{(M)}$ sería directamente dependiente de x_{a_k} y el resto de las entradas serían omitidas (los pesos asociados tenderían a ser cero durante el entrenamiento) [3]. Por ello se evitan conexiones directas entrada-salida estableciendo a cero los pesos $w^{(1)}_{a_k,j_{a_k}}$, con $k = 1, 2, ..., m$. En las figuras 2-3(b) no se muestran esas conexiones. Además se usan las salidas deseadas de clasificación como entradas extra, pero esta información sólo la usan las salidas de imputación. De esta manera, las salidas de las neuronas ocultas MTL calculan las siguientes expresiones,

Para $s = 1, ..., c$

$$z^{(s)}_{j_{a_k}} = g\left(\sum_{i=1}^{d} w^{(1)}_{i,j_{a_k}} x_i + w^{(1)}_{0,j_{a_k}}\right) \quad (2)$$

Para $s = c+1, ..., c+m$

$$z^{(s)}_{j_{a_k}} = g\left(\sum_{i=1}^{d+c} w^{(1)}_{i,j_{a_k}} x_i + w^{(1)}_{0,j_{a_k}}\right) \quad (3)$$

Los dos tipos de salidas $z^{(s)}_{j_{a_k}}$, correspondientes a las ecuaciones (2) y (3), se distinguen en las figuras 2-3(b) con una flecha, y una flecha comenzando en circulo, respectivamente.

Finalmente, las salidas finales de la red se obtienen por combinación lineal de las salidas de las neuronas ocultas mediante la segunda capa de pesos $w^{(2)}_{j_{a_k},s}$.

A. Fase de Entrenamiento

Antes de comenzar el entrenamiento, todos los pesos son inicializados aletoriamente con valores dentro del intérvalo $\left[-\frac{1}{2}\sqrt{\frac{3}{d+c}}, +\frac{1}{2}\sqrt{\frac{3}{d+c}}\right]$ [7], el conjunto de entrenamiento es normalizado a media cero y varianza unidad. Además, los valores perdidos son inicializados a cero, ya que una entrada igual a cero no contribuye al aprendizaje.

La función de error E a minimizar está compuesta por dos términos, el error de las salidas de clasificación, $E^{(\mathrm{C})}$, y el error de las salidas de imputación, $E^{(\mathrm{M})}$.

$$E = E^{(\mathrm{C})} + E^{(\mathrm{M})} = E^{(\mathrm{C})} + \sum_{k=1}^{m} E_k^{(\mathrm{M})} \quad (4)$$

En concreto, se emplea la función de error cuadrático medio,

$$E^{(\mathrm{C})} = \frac{1}{2}\sum_{n=1}^{N}\left(\|\mathbf{o}^{(n,\mathrm{C})} - \mathbf{t}^{(n,\mathrm{C})}\|^2\right) \quad (5)$$

$$E_k^{(\mathrm{M})} = \frac{1}{2}\sum_{n=1}^{N}\left(o_k^{(n,\mathrm{M})} - x_{a_k}^{(n)}\right)^2 \quad (6)$$

La optimización de las distintas funciones de error se realiza mediante el método de descenso por gradiente en modo secuencial con tasas adaptativas y termino de momento. Para ello, es necesario calcular las derivadas de E con respecto de los pesos (y sesgos). Esas derivadas dependen de las diferencias entre las salidas obtenidas y los valores deseados [1]. Si existen valores perdidos en el patrón $\mathbf{x}^{(n)}$, no es posible calcular las diferencias para las características incompletas ya que los valores son desconocidos. Por esta razón, las diferencias asociadas a salidas deseadas de imputación desconocidas son establecidas a cero.

Además durante el entrenamiento, los valores perdidos son estimados usando las salidas de imputación. El aprendizaje de la tarea de clasificación afecta a las salidas de imputación, por tanto la imputación se realiza de manera orientada a solucionar la tarea principal. La imputación se realiza cuando el aprendizaje común de las tareas secundarias empieza a detenerse.

B. Fase de Operación

La operación de la red propuesta depende de la presencia de valores perdidos en el patrón a clasificar, $\mathbf{x}^{(n)}$. Si $\mathbf{x}^{(n)}$ no presenta datos perdidos, no es necesario realizar una imputación de datos, y sólo se emplean las salidas de clasificación $\mathbf{o}^{(n,\mathrm{C})}$ para el etiquetado del patrón. En cambio, si $\mathbf{x}^{(n)}$ está incompleto, las salidas de imputación $o_k^{(n,\mathrm{M})}$ se emplean para estimar los valores perdidos en dicho patrón. Las salidas $o_k^{(n,\mathrm{M})}$ son función de las salidas deseadas de clasificación, las cuales no están disponibles durante el modo de operación. Para solucionarlo, se prueban todas las posibles etiquetas y con las correspondientes salidas $o_k^{(n,\mathrm{M})}$ se imputan los datos incompletos obteniendo $\tilde{\mathbf{x}}^{(n)}$. Tras este proceso, se clasifica este patrón empleando las salidas $\mathbf{o}^{(n,\mathrm{C})}$. Para cada una de las clases probadas, se escoge la clase más consistente. La consistencia es una medida de la diferencia entre las etiquetas probadas y las salidas $\mathbf{o}^{(n,\mathrm{C})}$ que produce la red después de que los datos incompletos han sido imputados usando $o_k^{(n,\mathrm{C})}$.

V. Resultados Experimentales

A. Problema Iris

Este problema consta de 150 casos, con cuatro atributos (A1, ..., A4), pertenecientes a tres posibles clases. Empleamos 50 patrones para entrenamiento y los 100 restantes como conjunto de test. Este problema presenta una probabilidad de error en torno a un $4\,\%$. Para evaluar la influencia de los valores perdidos, se eliminarán aleatoriamente un porcentaje de datos en varias de sus características, tanto en el conjunto de entrenamiento como en el de test. Es muy importante tener en cuenta qué características están incompletas y la relación que presentan cada una de ellas con la tarea principal de clasificación. Para medir estas relaciones, usamos la Información Mutua (IM). La IM entre la tarea principal y las características del problema Iris son: 0.877, para A1; 0.511, para A2; 1.446, para A3, y 1.436, para A4.

En la tabla I mostramos los resultados obtenidos para distintas combinaciones de características incompletas con porcentajes de datos perdidos iguales a $30\,\%$ y $40\,\%$. La

primera columna indica qué atributos están incompletos, identificados con un 1, o completos, con un 0. Cuando alguna de las características con mayor IM (A3, A4) están incompletas, los resultados obtenidos son peores que en los casos donde las características incompletas son las qué presentan menor IM (A1, A2).

TABLA I

PROBABILIDADES DE ERROR OBTENIDAS (MEDIA $\pm$ DESV. ESTÁNDAR) EN ENTRENAMIENTO (TRAINING) Y TEST PARA EL PROBLEMA IRIS

Atributos				**Missing (%)**	**Training (%)**	**Test (%)**
A1	A2	A3	A4		Mean $\pm$ SD	Mean $\pm$ SD
1	0	0	0	30 %	$2{,}00 \pm 0{,}10$	$4{,}00 \pm 0{,}60$
1	0	0	0	40 %	$2{,}20 \pm 1{,}00$	$4{,}53 \pm 0{,}88$
0	1	0	0	30 %	$2{,}40 \pm 0{,}80$	$3{,}20 \pm 0{,}88$
0	1	0	0	40 %	$2{,}60 \pm 1{,}00$	$3{,}33 \pm 0{,}67$
0	0	1	0	30 %	$2{,}60 \pm 1{,}20$	$5{,}07 \pm 0{,}53$
0	0	1	0	40 %	$1{,}80 \pm 1{,}08$	$4{,}80 \pm 0{,}88$
0	0	0	1	30 %	$1{,}80 \pm 1{,}08$	$4{,}13 \pm 1{,}11$
0	0	0	1	40 %	$3{,}00 \pm 1{,}34$	$4{,}40 \pm 1{,}04$
1	0	1	0	30 %	$2{,}40 \pm 1{,}50$	$5{,}07 \pm 0{,}53$
1	0	1	0	40 %	$2{,}00 \pm 1{,}26$	$5{,}60 \pm 0{,}80$
1	0	0	1	30 %	$2{,}00 \pm 0{,}89$	$4{,}40 \pm 1{,}20$
1	0	0	1	40 %	$3{,}00 \pm 1{,}84$	$5{,}60 \pm 1{,}16$
0	1	1	0	30 %	$3{,}20 \pm 1{,}33$	$4{,}40 \pm 0{,}85$
0	1	1	0	40 %	$3{,}40 \pm 2{,}01$	$4{,}27 \pm 0{,}53$
1	1	1	0	30 %	$3{,}20 \pm 2{,}04$	$4{,}67 \pm 0{,}89$
1	1	1	0	40 %	$4{,}00 \pm 1{,}26$	$4{,}93 \pm 0{,}85$

Otro aspecto a destacar es que las características menos relacionadas están menos influenciadas por el aprendizaje de la tarea de clasificación, y al contrario para los atributos más relacionados. Para mostrarlo, la figura 4 muestra la evolución del coste de cada cada una de las tareas durante el entrenamiento cuando presentan todos los atributos un 10 % de valores perdidos. Podemos observar cómo las tareas secundarias asociadas a las características A3 y A4 se aprenden mejor que el resto de tareas secundarias. También podemos ver cómo el coste asociado a la tarea principal baja considerablemente cuando se realiza la primera imputación en la época 27.

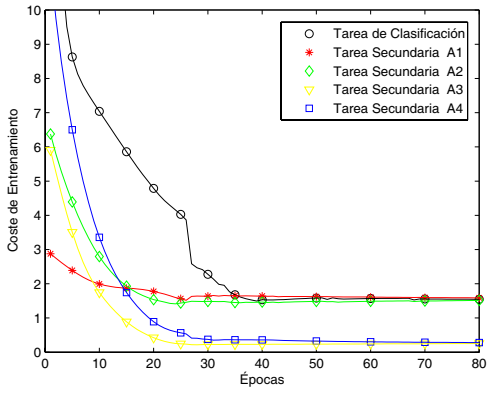


Fig. 4. Evolución del coste de entrenamiento para cada tarea en el problema Iris con 10 % de valores perdidos en todas las características.

B. *Problema Pima*

El objetivo del problema Pima es diagnosticar diabetes a partir de un conjunto de datos obtenido sobre una población de indios pima americanos. Cada patrón tiene ocho entradas (A1, ..., A8). El problema consta de 632 casos divididos previamente en un conjunto de entrenamiento (formado por 200 patrones completos y 100 patrones incompletos) y un conjunto de test (332 patrones completos). Los valores perdidos están presentes en tres de las ocho características de entrada con distintos porcentajes de valores perdidos: A3 (4.33 %), A4 (32.67 %) y A5 (1.00 %).

La probabilidad de error obtenida entrenando únicamente con los casos completos es: $23{,}34 \pm 1{,}63$ %. Sin embargo, los resultados obtenidos con la red propuesta son mejores que en el caso de emplear solamente los casos completos, obteniendo un error de clasificación igual a $19{,}92 \pm 0{,}59$ %. El conjunto de entrenamiento obtenido, compuesto por los casos completos y los casos incompletos con valores imputados producen una mejor generalización, i.e., los valores imputados son los que contribuyen a mejorar el aprendizaje de la tarea principal.

VI. CONCLUSIONES

En este artículo se propone una red neuronal multitarea para problemas de decisión con información incompleta en las características de entrada. Esta arquitectura combina la clasificación y la estimación de valores perdidos usando las ventajas del MTL. Para ello, se emplea la clasificación como tarea principal y las características incompletas como tareas secundarias. La tarea principal de clasificación ayuda a las tareas secundarias de imputación durante el aprendizaje simultáneo de ambas. Además, las salidas asociadas a las tareas secundarias se emplean para estimar la información incompleta. Otro aspecto importante es el uso de las salidas deseadas de clasificación como entradas extra en el aprendizaje de las tareas secundarias. Durante la fase de operación se escoge la clase más consistente. Los resultados obtenidos en problemas reales y artificiales prueban las ventajas del esquema MTL propuesto donde la imputación es dirigida a la resolución de la tarea que realmente se desea aprender.

Son varias las líneas de trabajo futuras que se pretenden abordar. Las principales son el dimensionado automático mediante técnicas de crecimiento y extender el método propuesto a otras máquinas de aprendizaje, como por ejemplo, Máquinas de Vectores Soporte (SVM, Support Vector Machine).

REFERENCIAS

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.

[2] R. J. A. Little, D. B. Rubin, *Statistical Analysis with Missing Data*, Wiley, New Jersey, 2002.

[3] R. Caruana, Multitask learning: a knowledge-based source of inductive bias. *Proceedings of the 10th International Conference of Cognitive Science*, pp. 41-48, 1993.

[4] M. K. Markey, G. D. Tourassi, M. Margolis and D. M. DeLong, "Impact of missing data in evaluating artificial neural networks trained on complete data", *Computers in Biology and Medicine*, pp. 516-525, vol. 36, no. 5, 2006.

[5] H. Ishibuchi, A. Miyazaki, K. Kwon and H. Tanaka, "Learning from incomplete training data with missing values and medical application", *Proceedings of 1993 International Joint Conference on Neural Networks*, pp. 1871-1874. Nagoya, Japan, 1994.

[6] P. J. García-Laencina, A. R. Figueiras-Vidal, Jesús Serrano-García, José-Luis Sancho-Gómez, "Exploiting multitask learning schemes using private subnetworks", *Proceedings of the 8th International Work-Conference on Artificial Neural Networks*, pp. 233-240, Barcelona, Spain, 2005.

[7] L. Bottou and P. Gallinari, "A Framework for the Cooperation of Learning Algorithms". Technical Report, Laboratorie de Recherche en Informatique, Universite de Paris XI, 9145 Orsay Cedex, France, 1991.