

Herramientas en GNU/Linux para estudiantes universitarios

Redes Neuronales con GNU/Linux

Francisco José Palacios Burgos

Herramientas en GNU/Linux para estudiantes universitarios: Redes Neuronales con GNU/Linux
por Francisco José Palacios Burgos

Copyright (c) 2003 Francisco J. Palacios Burgos.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Tabla de contenidos

1. Historia de las Redes Neuronales	1
1.1. Inteligencia Artificial.....	1
1.2. Redes Neuronales	1
2. Conceptos Básicos sobre RNA	3
2.1. La neurona artificial	3
2.2. Arquitectura de una RNA.....	4
3. Tipos de Redes Neuronales.....	6
3.1. Clasificación de las RNA	6
3.2. El perceptron multicapa (MLP).....	7
3.3. Redes Autoorganizadas. Redes SOFM	9
3.4. Redes de funcion de base radial (RBF).....	10
4. Ejemplo de entrenamiento de una red neuronal. Caso RBF	13
4.1. Ejemplo de aprendizaje para un problema de clasificacion por medio de una red RBF.....	13
A. Referencias interesantes	18
B. GNU Free Documentation License	19
B.1. PREAMBLE	19
B.2. APPLICABILITY AND DEFINITIONS	19
B.3. VERBATIM COPYING	20
B.4. COPYING IN QUANTITY	21
B.5. MODIFICATIONS	21
B.6. COMBINING DOCUMENTS	22
B.7. COLLECTIONS OF DOCUMENTS	23
B.8. AGGREGATION WITH INDEPENDENT WORKS	23
B.9. TRANSLATION.....	23
B.10. TERMINATION	24
B.11. FUTURE REVISIONS OF THIS LICENSE	24
B.12. ADDENDUM: How to use this License for your documents	24

Lista de tablas

4-1. Datos de entrenamiento	13
-----------------------------------	----

Lista de figuras

1-1. Representación de una neurona biológica	1
2-1. Representación de una neurona artificial tipo McCulloch-Pitts	3
2-2. Funciones de activación más usuales	4
3-1. Representación de redes unidireccionales y realimentadas	6
3-2. Representación de un Perceptrón Multicapa (MLP)	7
3-3. Forma funcional de una sigmoide	7
3-4. Arquitectura típica de un mapa SOM	9
3-5. Arquitectura típica de una red de tipo RBF	11
3-6. Forma funcional de una función tipo Gaussiana	11
3-7. Centros en el espacio de las entradas	12

Lista de ecuaciones

2-1. Suma ponderada	3
3-1. Error cuadrático medio	8
3-2. Términos delta	8
3-3. Distancia Euclídea entre el vector sináptico y la entrada	10
3-4. Actualización de los pesos en una red SOM	10
3-5. Actualización de los pesos de la capa de salida en una red RBF	12

Capítulo 1. Historia de las Redes Neuronales

1.1. Inteligencia Artificial

La historia de la informática tal y como la conocemos hoy en día es reciente. Hacia finales de los años 30 y durante la década de los 40, los trabajos de gente como Alan Turing o von Neumann asientan las bases de la informática moderna. En un principio se orienta hacia la computación algorítmica, es decir, la resolución de un determinado problema obteniendo un algoritmo que manipula los datos relativos al problema. El binomio hardware + software se muestra como una potente herramienta para la resolución de problemas que el ser humano no podría resolver o que tardaría mucho tiempo en hacerlo.

La evolución del hardware ha hecho que la potencia de cálculo haya crecido de tal forma que los ordenadores hoy en día son indispensables en muchas áreas de actividad del ser humano. La computación algorítmica sin embargo no es suficiente cuando nos enfrentamos a ciertas tareas. Por ejemplo, algo tan sencillo para el ser humano como reconocer una cara de otra persona es el tipo de problema que no es tan fácil ser resuelto por la vía algorítmica.

Debido a este tipo de problemas desde finales de los 50 se ha venido investigando en un conjunto de técnicas que utilizan un enfoque diferente para resolver los problemas. Este conjunto de técnicas y herramientas se bautizó con el nombre de Inteligencia Artificial (IA), porque lo que se pretendía era que los ordenadores presentaran un comportamiento inteligente, entendiendo por esto que supieran hacer frente a ciertos problemas de una manera similar a como lo hacen los seres humanos.

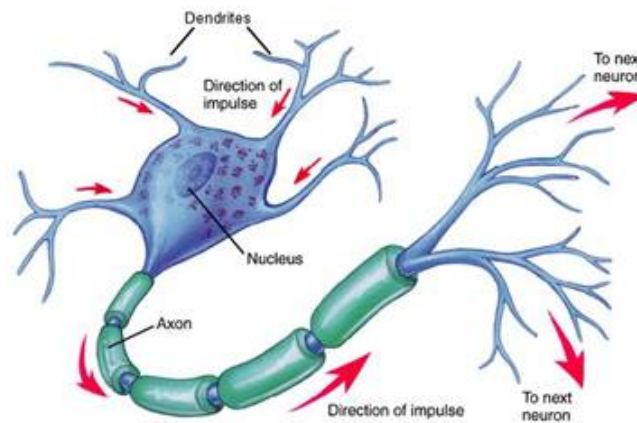
Dentro de la IA, se trabajó en dos enfoques distintos. Por un lado, se desarrolló lo que se conoce como el enfoque simbólico. Este enfoque asienta sus bases en la manipulación de símbolos en vez del mero cálculo numérico, tradicional de la computación algorítmica. La realidad se plasma por medio de una serie de reglas. Herramientas como la lógica de predicados, nos permiten manipular los símbolos y las reglas para obtener nuevas reglas. Este enfoque se presta a ser muy útil en ciertos tipos de problemas aunque en general tiene la desventaja de que a la hora de buscar la solución a un determinado problema los métodos de deducción presentan una explosión combinatoria que hace que requiera bastante tiempo de cálculo.

El otro enfoque tradicional es el enfoque conexionista y es donde se encuadran las redes neuronales. La idea aquí es desarrollar un sistema formado por pequeñas unidades de cálculo en cierta medida muy simples y hacer mediante conexiones entre ellas, que todo el conjunto sea capaz de resolver cierta clase de problemas.

1.2. Redes Neuronales

La idea que animó el modelo conexionista fue la de imitar el sistema de computación más complejo de los que se conocen hasta ahora, que es el cerebro. El cerebro está formado por millones de células llamadas neuronas. Estas neuronas son unos procesadores de información muy sencillos con un canal de entrada de información (dendritas), un órgano de cómputo (soma) y un canal de salida de información (axón).

Figura 1-1. Representación de una neurona biológica



De esta forma, las RNA imitan en cierto modo la estructura y física y el modo de operación de un cerebro. Teniendo en cuenta que el cerebro presenta las cualidades de procesamiento paralelo, procesamiento distribuido y adaptabilidad, un sistema RNA tiene también estas características.

El sistema resulta ser intrínsecamente paralelo porque está formado por unidades elementales de procesamiento llamadas neuronas. Cada neurona realiza un tipo de procesamiento muy simple.

El sistema es distribuido. Esto quiere decir que la información no se almacena localmente en ciertas zonas concretas de la RNA sino que se halla presente por toda ella, en concreto, se almacena en la sinapsis entre las neuronas. De igual forma, la computación es también distribuida. Al calcular la respuesta de la red neuronal, intervienen todos y cada uno de los procesadores elementales, los cuales se hallan distribuidos por toda la arquitectura de la red. Además, este carácter distribuido hace que la red presente tolerancia a fallos (si se pierde una parte de las neuronas no se pierde toda la información)

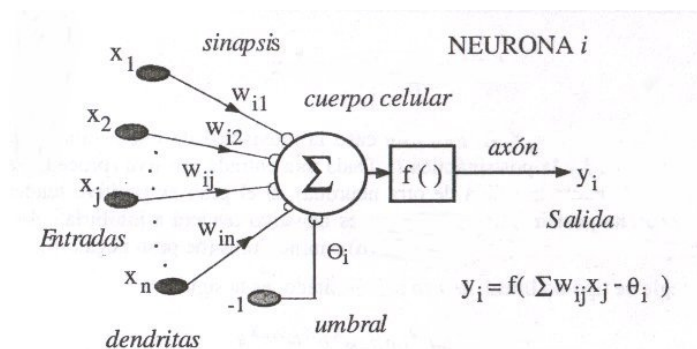
Una red neuronal presenta además un grado de adaptabilidad que se concreta en las capacidades de aprendizaje y generalización. Por aprendizaje entendemos la capacidad para recoger información de las experiencias y utilizarlas para actuar ante situaciones futuras. Íntimamente relacionada con el aprendizaje está la generalización, que podría definirse como la capacidad para abstraer la información útil, más allá de los casos particulares. De esta manera, la RNA es capaz de responder ante casos desconocidos.

Capítulo 2. Conceptos Básicos sobre RNA

2.1. La neurona artificial

La unidad básica de una RNA es la neurona. Aunque hay varios tipos de neuronas diferentes, la mas comun es la de tipo McCulloch-Pitts. En la siguiente figura puede verse una representación de la misma

Figura 2-1. Representación de una neurona artificial tipo McCulloch-Pitts



Una neurona artificial es un procesador elemental, en el sentido de que procesa un vector $\mathbf{x} (x_1, x_2, \dots, x_N)$ de entradas y produce un respuesta o salida única. Los elementos clave de una neurona artificial los podemos ver en la figura anterior y son los siguientes:

- Las entradas que reciben los datos de otras neuronas. En una neurona biológica corresponderían a las dendritas
- Los pesos sinapticos w_{ij} . Al igual que en una neurona biológica se establecen sinápsis entre las dendritas de una neurona y el axón de otra, en una neurona artificial a las entradas que vienen de otras neuronas se les asigna un peso, un factor de importancia. Este peso, que es un número, se modifica durante el entrenamiento de la red neuronal, y es aquí por tanto donde se almacena la información que hara que la red sirva para un propósito u otro.
- Una regla de propagación. Con esas entradas y los pesos sinapticos, se suele hacer algun tipo de operación para obtener el valor del potencial postsinaptico (valor que es funcion de las entradas y los pesos y que es el que se utiliza en último término para realizar el procesamiento). Una de las operaciones mas comunes es sumar las entradas, pero teniendo en cuenta la importancia de cada una (el peso sináptico asociado a cada entrada). Es lo que se llama suma ponderada, aunque otras operaciones también son posibles.


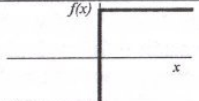
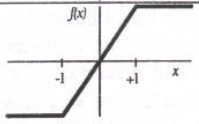
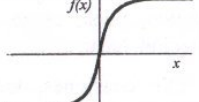
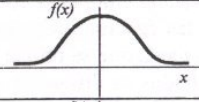
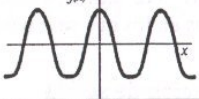
Ecuación 2-1. Suma ponderada

$$h_i(t) = \sum_j w_{ij}x_j$$

La otra regla de propagación mas habitual es la distancia euclídea. Este es el tipo de regla que tienen redes como el SOM o las RBF.

- Una función de activación. El valor obtenido con la regla de propagación, se filtra a través de una función conocida como función de activación y es la que nos da la salida de la neurona. Según para lo que se desee entrenar la red neuronal, se suele escoger una función de activación u otra en ciertas neuronas de la red. En la siguiente tabla se muestran las funciones de activación mas usuales

Figura 2-2. Funciones de activación más usuales

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1+e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

En muchas ocasiones la razón para la aplicación de una función de activación distinta de la identidad surge de la necesidad de que las neuronas produzcan una salida acotada. Esto desde un punto de vista de similitud con el sistema biológico, no es tan descabellado, ya que las respuestas de las neuronas biológicas están acotadas en amplitud. Además cada neurona tiene asociado un número denominado bias o umbral, que puede verse como un número que indica a partir de que valor del potencial postsináptico la neurona produce una salida significativa.

2.2. Arquitectura de una RNA

Desde un punto de vista matemático, se puede ver una red neuronal como un grafo dirigido y ponderado donde cada uno de los nodos son neuronas artificiales y los arcos que unen los nodos son las conexiones sinápticas. Al ser dirigido, los arcos son unidireccionales. ¿Que quiere decir esto? En el lenguaje de neuronas y conexiones significa que la información se propaga en un unico sentido, desde una neurona presinaptica (neurona origen) a una neurona postsináptica (neurona destino)

Por otra parte es ponderado, lo que significa que las conexiones tienen asociado un número real, un peso, que indica la importancia de esa conexión con respecto al resto de las conexiones. Si dicho peso es positivo la conexión se dice que es excitadora, mientras que si es negativa se dice que es inhibidora.

Lo usual es que las neuronas se agrupen en capas de manera que una RNA esta formada por varias capas de neuronas. Aunque todas las capas son conjuntos de neuronas, según la funcion que desempeñan, suelen recibir un nombre específico. Las mas comunes son las siguientes:

- Capa de entrada: las neuronas de la capa de entrada, reciben los datos que se proporcionan a la RNA para que los procese.
- Capas ocultas: estas capas introducen grados de libertad adicionales en la RNA. El número de ellas puede depender del tipo de red que estemos considerando. Este tipo de capas realiza gran parte del procesamiento.
- Capa de salida: Esta capa proporciona la respuesta de la red neuronal. Normalmente también realiza parte del procesamiento.

Capítulo 3. Tipos de Redes Neuronales

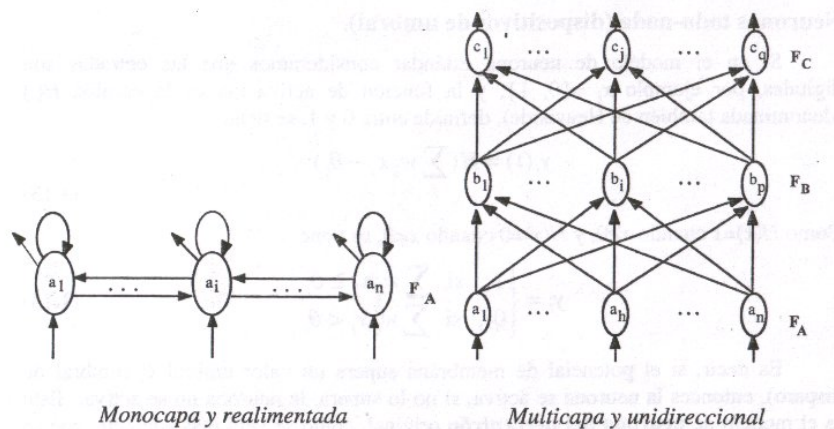
3.1. Clasificación de las RNA

Según el criterio que escojamos para clasificar las RNA tendremos una clasificación u otra. Lo más común es usar la arquitectura y el tipo de aprendizaje como criterios de clasificación.

Si nos fijamos en la arquitectura podemos tener dos posibilidades distintas. Si la arquitectura de la red no presenta ciclos, es decir, no se puede trazar un camino de una neurona a sí misma, la red se llama unidireccional (feedforward).

Por el contrario, si podemos trazar un camino de una neurona a sí misma la arquitectura presenta ciclos. Este tipo de redes se denominan recurrentes o realimentadas (recurrent).

Figura 3-1. Representación de redes unidireccionales y realimentadas



El otro criterio más habitual para clasificar las redes neuronales es el tipo de aprendizaje que se utilice. Hay cuatro clases de aprendizaje distintos:

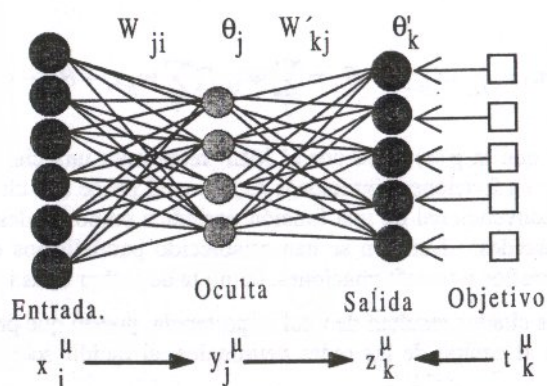
- Aprendizaje supervisado: En este tipo de aprendizaje se le proporciona a la RNA una serie de ejemplos consistentes en unos patrones de entrada, junto con la salida que debería dar la red. El proceso de entrenamiento consiste en el ajuste de los pesos para que la salida de la red sea lo más parecida posible a la salida deseada. Es por ello que en cada iteración se use alguna función que nos de cuenta del error o el grado de acierto que esta cometiendo la red.

- b. Aprendizaje no supervisado o autoorganizado: En este tipo de aprendizaje se presenta a la red una serie de ejemplos pero no se presenta la respuesta deseada. Lo que hace la RNA es reconocer regularidades en el conjunto de entradas, es decir, estimar una función densidad de probabilidad $p(\mathbf{x})$ que describe la distribución de patrones \mathbf{x} en el espacio de entrada R^n .
- c. Aprendizaje Híbrido: Es una mezcla de los anteriores. Unas capas de la red tienen un aprendizaje supervisado y otras capas de la red tienen un aprendizaje de tipo no supervisado. Este tipo de entrenamiento es el que tienen redes como las RBF.
- d. Aprendizaje reforzado (reinforcement learning): Es un aprendizaje con características del supervisado y con características del autoorganizado. No se proporciona una salida deseada, pero sí que se le indica a la red en cierta medida el error que comete, aunque es un error global.

3.2. El perceptrón multicapa (MLP)

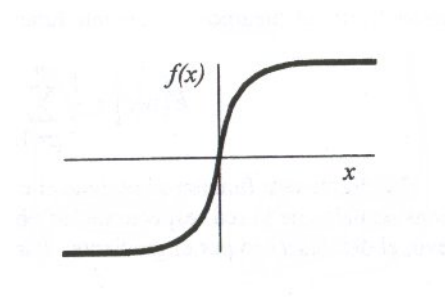
Este es uno de los tipos de redes más comunes. Se basa en otra red más simple llamada perceptrón simple solo que el número de capas ocultas puede ser mayor o igual que una. Es una red unidireccional (feedforward). La arquitectura típica de esta red es la siguiente:

Figura 3-2. Representación de un Perceptrón Multicapa (MLP)



Las neuronas de la capa oculta usan como regla de propagación la suma ponderada de las entradas con los pesos sinápticos w_{ij} y sobre esa suma ponderada se aplica una función de transferencia de tipo sigmoide, que es acotada en respuesta.

Figura 3-3. Forma funcional de una sigmoide



El aprendizaje que se suele usar en este tipo de redes recibe el nombre de retropropagación del error (backpropagation). Como función de coste global, se usa el error cuadrático medio. Es decir, que dado un par $(\mathbf{x}_k, \mathbf{d}_k)$ correspondiente a la entrada k de los datos de entrenamiento y salida deseada asociada se calcula la cantidad:

Ecuación 3-1. Error cuadrático medio

$$E(w_{ij}, \theta_j, w'_{kj}, \theta'_k) = \frac{1}{2} \sum_p \sum_k \left[d_k^p - f \left(\sum_j w'_{kj} y_j^p - \theta'_k \right) \right]^2$$

que vemos que es la suma de los errores parciales debido a cada patrón (índice p), resultantes de la diferencia entre la salida deseada d_p y la salida que da la red $f(\cdot)$ ante el vector de entrada x_k . Si estas salidas son muy diferentes de las salidas deseadas, el error cuadrático medio será grande. f es la función de activación de las neuronas de la capa de salida e y la salida que proporcionan las neuronas de la última capa oculta.

Sobre esta función de coste global se aplica algún procedimiento de minimización. En el caso del MLP se hace mediante un descenso por gradiente. Las expresiones que resultan aplicando la regla de la cadena son las siguientes:

Ecuación 3-2. Términos delta

$$\begin{aligned} \delta w'_{kj} &= -\epsilon \frac{\partial E}{\partial w'_{kj}} \\ \delta w'_{ji} &= -\epsilon \frac{\partial E}{\partial w'_{ji}} \\ \delta w'_{kj} &= \epsilon \sum_p \Delta_k^p y_j^p \quad \text{con} \quad \Delta_k^p = [d_k^p - f(v_k^p)] \frac{\partial f(v_k^p)}{\partial v_k^p} \\ \delta w_{ij} &= \epsilon \sum_p \Delta_j^p x_i^p \quad \text{con} \quad \Delta_j^p = \left(\sum_k \Delta_k^p w'_{kj} \right) \frac{\partial f(v_j^p)}{\partial v_j^p} \end{aligned}$$

Siendo y_k las salidas de la capa oculta.

El aprendizaje por backpropagation queda como sigue:

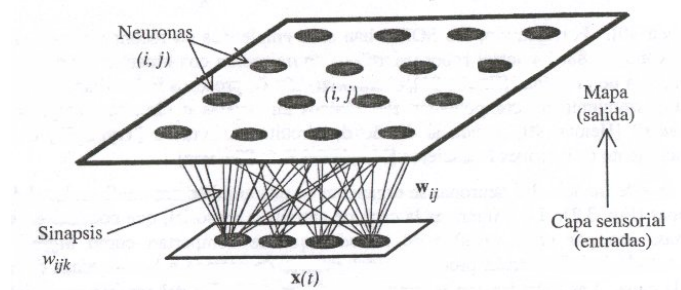
1. Inicializar los pesos y los umbrales iniciales de cada neurona. Hay varias posibilidades de inicialización siendo las mas comunes las que introducen valores aleatorios pequeños.
2. Para cada patrón del conjunto de los datos de entrenamiento
 - a. Obtener la respuesta de la red ante ese patrón. Esta parte se consigue propagando la entrada hacia adelante, ya que este tipo de red es feedforward. Las salidas de una capa sirven como entrada a las neuronas de la capa siguiente, procesandolas de acuerdo a la regla de propagación y la función de activación correspondientes.
 - b. Calcular los errores asociados según la ecuación 3-2
 - c. Calcular los incrementos parciales (sumandos de los sumatorios). Estos incrementos dependen de los errores calculados en 2.b
3. Calcular el incremento total ,para todos los patrones, de los pesos y los umbrales según las expresiones en la ecuación 3-2
4. Actualizar pesos y umbrales
5. Calcular el error actual y volver al paso 2 si no es satisfactorio.

3.3. Redes Autoorganizadas. Redes SOFM

En este tipo de redes el entrenamiento o aprendizaje es diferente al de las redes con entrenamiento supervisado. A la red no se le suministra junto a los patrones de entrenamiento, una salida deseada. Lo que hará la red es encontrar regularidades o clases en los datos de entrada, y modificar sus pesos para ser capaz de reconocer estas regularidades o clases.

Uno de los tipos de redes que pertenece a esta familia y que se ha usado bastante son los mapas autoorganizados, SOM (Self-Organizing Maps). La arquitectura típica de este tipo de mapas es la siguiente:

Figura 3-4. Arquitectura típica de un mapa SOM



Como se puede apreciar es una red de tipo unidireccional. La red se organiza en dos capas, siendo la primera capa la formada por las neuronas de entrada. La segunda capa consiste en un array de neuronas de dos dimensiones. Como se necesitan dos índices para etiquetar cada neurona, los pesos sinápticos asociados a cada neurona tendrán tres índices (i,j,k) donde (i,j) indican la posición de la neurona en la capa y k, la componente o conexión con cierta neurona de entrada.

En cuanto al entrenamiento, este es un ejemplo de red que utiliza un aprendizaje de tipo no supervisado. Además, cada neurona utiliza como regla de propagación una distancia de su vector de pesos sinápticos al patrón de entrada. Otros conceptos importantes que intervienen en el proceso de aprendizaje son los conceptos de neurona ganadora y vecindad de la misma. Un algoritmo de aprendizaje muy usado con este tipo de redes es el algoritmo de Kohonen que se describe como sigue:

1. Inicialización de los pesos w_{ijk} . Hay varias opciones posibles
2. Elección de un patrón de entre el conjunto de patrones de entrenamiento
3. Para cada neurona del mapa, calcular la distancia euclídea entre el patrón de entrada \mathbf{x} y el vector de pesos sinápticos

Ecuación 3-3. Distancia Euclídea entre el vector sináptico y la entrada

$$d^2(\mathbf{w}_{ij}, \mathbf{x}) = \sum_k (w_{ijk} - x_k)^2$$

4. Evaluar la neurona ganadora, es decir aquella cuya distancia es la menor de todas
5. Actualizar los pesos sinápticos de la neurona ganadora y de sus vecinas según la regla:

Ecuación 3-4. Actualización de los pesos en una red SOM

$$\delta w_{ijk}(t) = \alpha(t) \cdot h(|\mathbf{i} - \mathbf{g}|, t) \cdot (x_k(t) - w_{ijk}(t))$$

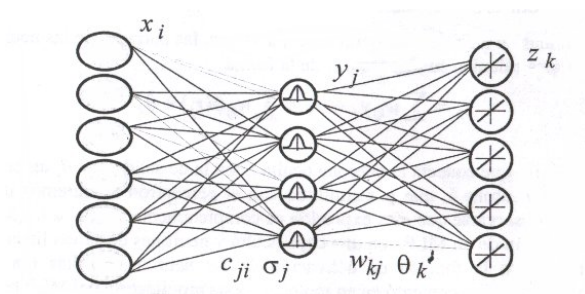
$\alpha(t)$ es un factor llamado ritmo de aprendizaje que da cuenta de la importancia que la diferencia entre el patrón y los pesos tiene en el ajuste de los mismos a lo largo del proceso de aprendizaje. Hay varias posibilidades para esta función, desde un constante hasta algún tipo de función monótona decreciente con el tiempo. h es una función de vecindad que nos indica en que medida se modifican los pesos de las neuronas vecinas. Con esto quiere decir que cuando la neurona ganadora modifica sus pesos, la vecindad de esta neurona lo hace también, en mayor o menor medida según sea la forma funcional de h . En general, las funciones empleadas para h tienen un máximo en $|\mathbf{i}-\mathbf{j}|=0$ y decrecen más o menos rápido a medida que esta distancia aumenta.

6. Lo usual es fijar un número de iteraciones antes de comenzar el aprendizaje. Si no se llegó al número de iteraciones establecido previamente, se vuelve al paso 2. Sobre este número de iteraciones necesario, se suelen tomar criterios como el número de neuronas en el mapa.

3.4. Redes de funcion de base radial (RBF)

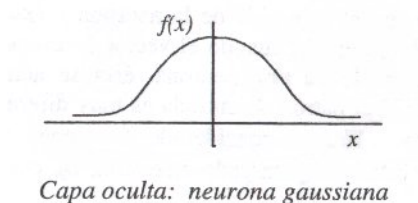
Este tipo de redes se caracteriza por tener un aprendizaje o entrenamiento híbrido. La arquitectura de estas redes se caracteriza por la presencia de tres capas: una de entrada, una única capa oculta y una capa de salida.

Figura 3-5. Arquitectura típica de una red de tipo RBF



Aunque la arquitectura pueda recordar a la de un MLP, la diferencia fundamental está en que las neuronas de la capa oculta en vez de de calcular una suma ponderada de las entradas y aplicar una sigmoide, estas neuronas calculan la distancia euclídea entre el vector de pesos sinápticos (que recibe el nombre en este tipo de redes de centro o centroide) y la entrada (de manera casi analoga a como se hacia con los mapas SOM) y sobre esa distancia se aplica una función de tipo radial con forma gaussiana.

Figura 3-6. Forma funcional de una función tipo Gaussiana



Para el aprendizaje de la capa oculta, hay varios métodos, siendo uno de los más conocidos el algoritmo denominado k-medias (k-means) que es un algoritmo no supervisado de clustering. k es el número de grupos que se desea encontrar, y se corresponde con el número de neuronas de la capa oculta, que es un parámetro que hay que decidir de antemano. El algoritmo se plantea como sigue:

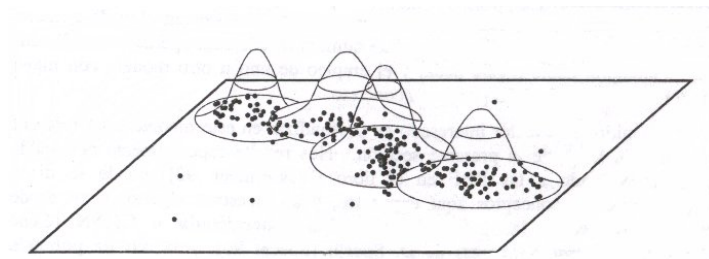
1. Inicializar los pesos (los centros) en el instante inicial. Una inicialización típica es la denominada k-primeras mediante la cual los k centros se hacen iguales a las k primeras muestras del conjunto de datos de entrenamiento $\{\mathbf{x}_p\}_{p=1..N}$

$$\mathbf{c}_1 = \mathbf{x}_1, \mathbf{c}_2 = \mathbf{x}_2, \dots, \mathbf{c}_N = \mathbf{x}_N,$$

2. En cada iteración, se calculan los dominios, es decir, se reparten las muestras entre los k centros. Esto se hace de la siguiente manera: Dada una muestra \mathbf{x}_j se calcula las distancias a cada uno de los centros \mathbf{c}_k . La muestra pertenecerá al dominio del centro cuya distancia calculada sea la menor
3. Se calculan los nuevos centros como los promedios de los patrones de aprendizaje pertenecientes a sus dominios. Viene a ser como calcular el centro de masas de la distribución de patrones, tomando que todos pesan igual.
4. Si los valores de los centros varían respecto a la iteración anterior se vuelve al paso 2, si no, es que se alcanzó la convergencia y se finaliza el aprendizaje

Una vez fijados los valores de los centros, sólo resta ajustar las anchuras de cada neurona. Las anchuras son los parámetros sigma que aparecen en cada una de las funciones gaussianas y reciben ese nombre por su interpretación geométrica, dan una medida de cuando una muestra activa una neurona oculta para que de una salida significativa

Figura 3-7. Centros en el espacio de las entradas



normalmente se toma el criterio de que para cada neurona se toma como valor sigma la distancia al centro más cercano.

Finalmente, se entrena la capa de salida. El entrenamiento de esta capa se suele usar un algoritmo parecido al que se usa para la capa de salida del MLP. La actualización de los pesos viene dada por la expresión:

Ecuación 3-5. Actualización de los pesos de la capa de salida en una red RBF

$$z_k = \sum_j w_{kj} \phi(r_j) + \theta_k$$

$$\delta w_{kj}(t) = \epsilon (d_k - z_k) \phi(r_j)$$

Con este fin se suele presentar todos los patrones de la muestra de entrenamiento varias veces. Cada una de estas veces recibe el nombre de época.

Capítulo 4. Ejemplo de entrenamiento de una red neuronal. Caso RBF

4.1. Ejemplo de aprendizaje para un problema de clasificacion por medio de una red RBF

Como ejemplo de entrenamiento vamos a escoger un problema de clasificacion sencillo y utilizaremos una red de tipo RBF. El problema en cuestion se plantea como sigue: Imaginemos que tenemos puntos en el plano. Los puntos se distribuyen en ciertas regiones, donde la densidad de puntos es mayor. Queremos que la red neuronal sirva como un clasificador de esas regiones de manera que si preguntamos por nuevos puntos nos diga si esta en alguna de esas tres regiones.

Como tendremos tres regiones, escogeremos tres neuronas para la capa oculta. En realidad podiamos haber escogido más, pero con tres sirve para ilustrar el ejemplo. La capa de salida tendra tambien 3 neuronas y estara acotada entre 0 y 1. Si la salida i-esima toma un valor cercano a 1 y el resto de las salidas toman valores cercanos a 0 eso nos dira que la muestra de entrenamiento pertenece a la zona i-esima. Además la funcion de activacion de las neuronas de la capa de salida sera una sigmoide, que nos acota la salida entre 0 y 1.

Nuestro conjunto de muestras de entrenamiento sera el siguiente:

Tabla 4-1. Datos de entrenamiento

0.8	0.8	
0.9	0.1	0.1
0.9	1.3	
0.9	0.1	0.1
1.1	0.7	
0.9	0.1	0.1
1.2	1.1	
0.9	0.1	0.1
6.7	8.2	
0.1	0.9	0.1
6.9	7.8	
0.1	0.9	0.1
7.2	8.1	
0.1	0.9	0.1
7.5	7.5	
0.1	0.9	0.1
14.0	3.0	
0.1	0.1	0.9
14.8	4.1	

0.1	0.1	0.9
15.3	4.2	
0.1	0.1	0.9
16.0	3.0	
0.1	0.1	0.9

Las filas corresponden a la entrada (un punto en el plano se representa con dos coordenadas reales) y su correspondiente salida deseada (tres numeros entre 0 y 1. No usaremos exactamente el valor 1 ni el valor 0 pero con esos valores nos bastara).

Para entrenar la capa oculta elegiremos el algoritmo de las k-medias. El desarrollo de dicho algoritmo es el siguiente:

```

-----
Iteracion: 1
-----
Muestra 0
dc0 = 0, dc1 = 0.509902, dc2 = 0.316228,
Muestra 1
dc0 = 0.509902, dc1 = 0, dc2 = 0.632456,
Muestra 2
dc0 = 0.316228, dc1 = 0.632456, dc2 = 0,
Muestra 3
dc0 = 0.5, dc1 = 0.360555, dc2 = 0.412311,
Muestra 4
dc0 = 9.46414, dc1 = 9.01388, dc2 = 9.36002,
Muestra 5
dc0 = 9.28493, dc1 = 8.8459, dc2 = 9.16788,
Muestra 6
dc0 = 9.70824, dc1 = 9.26984, dc2 = 9.5901,
Muestra 7
dc0 = 9.47523, dc1 = 9.05539, dc2 = 9.33809,
Muestra 8
dc0 = 13.3821, dc1 = 13.2098, dc2 = 13.1034,
Muestra 9
dc0 = 14.3837, dc1 = 14.1792, dc2 = 14.1156,
Muestra 10
dc0 = 14.8933, dc1 = 14.6891, dc2 = 14.625,
Muestra 11
dc0 = 15.3584, dc1 = 15.1954, dc2 = 15.0765,
Dominios: 0 1 2 1 1 1 1 1 2 2 2 2

Centro 1: 0.8 0.8
Centro 2: 5.06667 5.66667
Centro 3: 12.24 3

```

```

-----
Iteracion: 2
-----
Muestra 0
dc0 = 0, dc1 = 6.47216, dc2 = 11.6496,
Muestra 1
dc0 = 0.509902, dc1 = 6.03563, dc2 = 11.4667,

```

Capítulo 4. Ejemplo de entrenamiento de una red neuronal. Caso RBF

Muestra 2
dc0 = 0.316228, dc1 = 6.35627, dc2 = 11.375,
Muestra 3
dc0 = 0.5, dc1 = 5.98377, dc2 = 11.2023,
Muestra 4
dc0 = 9.46414, dc1 = 3.01423, dc2 = 7.59813,
Muestra 5
dc0 = 9.28493, dc1 = 2.81287, dc2 = 7.18022,
Muestra 6
dc0 = 9.70824, dc1 = 3.23608, dc2 = 7.17019,
Muestra 7
dc0 = 9.47523, dc1 = 3.04667, dc2 = 6.53587,
Muestra 8
dc0 = 13.3821, dc1 = 9.32285, dc2 = 1.76,
Muestra 9
dc0 = 14.3837, dc1 = 9.85861, dc2 = 2.78632,
Muestra 10
dc0 = 14.8933, dc1 = 10.3379, dc2 = 3.28688,
Muestra 11
dc0 = 15.3584, dc1 = 11.2538, dc2 = 3.76,
Dominios: 0 0 0 0 1 1 1 1 2 2 2 2

Centro 1: 1 0.975
Centro 2: 7.075 7.9
Centro 3: 15.025 3.575

Iteracion: 3

Muestra 0
dc0 = 0.265754, dc1 = 9.47553, dc2 = 14.4931,
Muestra 1
dc0 = 0.340037, dc1 = 9.03829, dc2 = 14.307,
Muestra 2
dc0 = 0.292618, dc1 = 9.35631, dc2 = 14.2187,
Muestra 3
dc0 = 0.23585, dc1 = 8.98641, dc2 = 14.0448,
Muestra 4
dc0 = 9.20275, dc1 = 0.480234, dc2 = 9.52346,
Muestra 5
dc0 = 9.02168, dc1 = 0.201556, dc2 = 9.15785,
Muestra 6
dc0 = 9.44487, dc1 = 0.23585, dc2 = 9.03915,
Muestra 7
dc0 = 9.21008, dc1 = 0.583631, dc2 = 8.48712,
Muestra 8
dc0 = 13.1568, dc1 = 8.48326, dc2 = 1.17527,
Muestra 9
dc0 = 14.1494, dc1 = 8.60904, dc2 = 0.571183,
Muestra 10
dc0 = 14.6591, dc1 = 9.0189, dc2 = 0.682825,
Muestra 11
dc0 = 15.1361, dc1 = 10.1816, dc2 = 1.13192,
Dominios: 0 0 0 0 1 1 1 1 2 2 2 2

Centro 1: 1 0.975
 Centro 2: 7.075 7.9
 Centro 3: 15.025 3.575

Como vemos, en apenas 3 iteraciones el algoritmo ha convergido totalmente. Se aprecia como los dominos ya reflejan los tres grupos claramente (de hecho estos grupos ya aparecen en la 2 iteracion). Cabe resaltar que en este entrenamiento no se empleo la salida deseada para nada, y esto es debido a que se trata de un algoritmo de aprendizaje no supervisado.

Una vez entrenada la capa oculta, le tocaría el turno a la capa de salida. En ésta lo que se hace es pasar los patrones de entrenamiento durante un cierto numero de épocas (en este entrenamiento se escogio 60 épocas) y en cada época con todos los patrones de entrenamiento se calcula la variación de los pesos de la capa de salida según lo explicado en el apartado 3.4 Al final, los pesos que resultan para toda la red son los siguientes:

```

-----
Topología de la red : 0
-----
Número de neuronas de entrada: 2
Número de neuronas ocultas   : 3
Número de neuronas de salida : 3
Función de la capa oculta    : Gaussiana
Función de la capa de salida : Sigmoide
Red Híbrida MLP              : false
-----
Datos del entrenamiento
-----
Algoritmo de la capa oculta : K-medias
  Tipo de inicialización: K primeras
  Epsilon : 0.0010
Algoritmo de la capa de salida : Regla delta
  Ritmo de aprendizaje : 0.1000
  Número de épocas : 60
-----
Pesos de la red RBF
-----
Pesos de la capa oculta
-----
Neurona      0          1          2
-----
0             1.0000  7.0750  15.0250
1             0.9750  7.9000  3.5750
-----
Anchuras de la capa oculta
-----
          9.2120  9.0503  9.0503
-----
Pesos de la capa de salida
-----
Neurona      0          1          2
-----
0             1.6047 -0.9334 -1.1720
1            -1.5427  2.6404 -1.2721
    
```

```
2          -1.3295  -1.1118  1.9833
-----
Bias de la capa de salida
-----
          0.5470  -0.9427  0.0257
```

Con esto quedaría realizada la etapa de aprendizaje de la red neuronal. Si ahora presentamos otros datos para obtener su clasificación, estaríamos en la etapa de recuperación. Los datos que presentaremos serán los siguientes:

```
0.0  0.0  --> Region A
2.0  0.0  --> Region A
9.0  9.0  --> Region B
13.0 13.0  --> Entre Region B y Region C
```

como podemos apreciar, esta vez ya no se le presentan a la red las salidas deseadas. El resultado que da la red ante estas entradas es el siguiente:

```
0.8903  0.1692  0.0678 --> Region A
0.8309  0.2013  0.0795 --> Region A
0.1021  0.7057  0.2969 --> Region B
0.1243  0.5372  0.5301 --> ???
```

La tasa de aciertos fue del 75 %

Apéndice A. Referencias interesantes

- B. Martín del Brio y A. Sanz Molina, Redes Neuronales y Sistemas Borrosos, de. Ra-Ma, 1997

Excelente libro en Español, con un lenguaje muy claro y sin demasiados formalismos matemáticos más que los estrictamente necesarios

- Haykin S., Neural Networks , 2nd Edition, Prentice Hall, 1999, ISBN 0 13 273350 1

Segundo excelente libro de referencia. Está en inglés y tiene más formalismo matemático que el anterior.

- IEEE Transactions on Neural Networks (<http://iee-nns.org/>)

Esta revista contiene artículos de investigación avanzada sobre redes neuronales. Suele exigir un alto grado de matemáticas.

- <http://www.gc.ssr.upm.es/inves/neural/ann2/anntutorial.html>

Buen tutorial sobre redes neuronales en español

- <http://rfhs8012.-regensburg.de/~saj39122/jfroehl/diplom/e-sample.html>

Una página donde se puede ver en un applet de Java cómo funciona un SOM con el algoritmo de Kohonen

- http://www.infor.uva.es/biometria/Documentos/informes_uva/EstadoArte/EstadoArte/EstadoArte.html

Muy buen documento realizado por dos profesores de la Universidad de Valladolid

Apéndice B. GNU Free Documentation License

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

B.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

B.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the

above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

B.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute.

However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

B.4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

B.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

B.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

B.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

B.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

B.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

B.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

B.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

B.12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.