Ejercicio 11 & 12 - CC50Q
Teoría de la Información y Redes Neuronales

Prof: Pedro Ortega <peortega@dcc.uchile.cl>
Aux: Francisco Claude <fclaude@dcc.uchile.cl>

13 de noviembre de 2005

Entrega: Jueves 26 de Noviembre, por U-Cursos. La tarea es individual.

En esta tarea, Ud. debe implementar una red neuronal de dos capas y utilizarla para construir un clasificador que sea capaz de distinguir entre vocales nasales y orales.

Introducción

La mayoría de los sistemas de reconocimiento de voz se basa en el procesamiento global de las señales, y no aprovechan realmente las características particulares de la voz. En contraste, los sistemas analíticos consideran los procesos de articulación que dan origen a los diferentes fonemas del lenguaje, a partir de los cuales deducen características globales. En esta tarea se construirá un detector de vocales nasales y orales que formaría parte de un sistema analítico.

Para esto, se han registrado muchas sílabas habladas, a partir de las cuales se extrajeron las ventanas de tiempo que contenían vocales. Estas fueron procesadas, calculando un total de cinco características para caracterizarlas.

Descripción de los Datos

La tarea consta de dos archivos: `basededatos.txt/xls` y `casosnuevos.txt/xls`. Ambos vienen en dos formatos: texto plano (`.txt`) y archivo Excel (`.xls`). El archivo `basededatos` contiene 5000 ejemplos de entrenamiento $[x_1, x_2, \ldots, x_5]$ con su respectivo diagnóstico d, uno por fila. El archivo `casosnuevos` cuenta con 404 casos nuevos, para los cuales se poseen los indicadores de la señal pero no la clasificación.

Instrucciones

Estos son los pasos que Ud. debe seguir para resolver la tarea.

Implementar una red neuronal (8 horas): Implemente una red neuronal de dos capas de pesos. Todas las unidades deben poseer la función de activación *sigmoide logística*. La red neuronal debe ser entrenable, por ej. por medio del algoritmo de optimización descenso por el gradiente, minimizando *la entropía cruzada*. El número de neuronas en cada una de las capas debe ser ajustable. Se recomienda diseñar su programa en base a capas de neuronas como visto en clase, y no en base a neuronas individuales. Observe que su red debe ser capaz de:

- Inicializar la red neuronal con pesos aleatorios (se aconseja un valor aleatorio en [-1;1]).
- Entrenar una red neuronal con arquitectura *in-mid-out*, dado un conjunto de entrenamiento (entradas y salidas deseadas), un conjunto de validación, un factor de aprendizaje y un número de iteraciones.
- Imprimir el *error de validación* y el *error de entrenamiento* en cada iteración.
- Finalizar guardando los pesos, y retomar el entrenamiento en una siguiente ejecución (cargando los pesos grabados).
- Leer luego un conjunto de datos de entrada y clasificarlo utilizando los pesos optimizados. Las clasificaciones deben escribirse dentro de un archivo.

¡Haga un implementación sencilla!

Normalice la BD (10 minutos): Para no favorecer una variable frente a otra, normalice la base de entrenamiento. Esto se hace aplicando una transformación

sobre cada variable x_i, de manera que cada una quede con media $\overline{x}_i = 0$ y varianza muestral $s_i^2 = 1$. Si d_i^j es el valor en la i-ésima variable (columna) del j-ésimo registro, entonces

$$z_i^j = \frac{d_i^j - \overline{x}_i}{s_i^2}$$

es el valor del dato normalizado. Con esto ud. elimina el sesgo debido a las escalas distintas entre las variables.

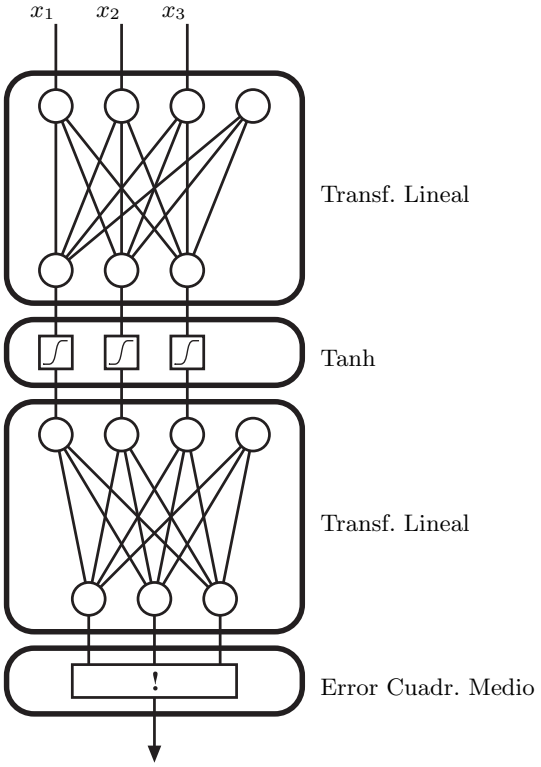
Prepare la BD (10 minutos): Divida la base de datos en una parte de entrenamiento y otra de validación. Construya éstos mezclando los registros aleatoriamente y destinando 70 % de los registros para el conjunto de entrenamiento y el 30 % restante para validación.

Entrene la red neuronal (4 horas): Ejecute el algoritmo de entrenamiento de la red neuronal. Pruebe con distinto número de neuronas en la capa oculta y varios factores de aprendizaje (por ej. 5-5-2 con $\mu = 0{,}1$, $\mu = 0{,}01$). Si el error de validación y de entrenamiento no decrecen entonces *hay algo que está mal*. Finalmente, clasifique los registros en el archivo `casosnuevos`.

Elabore un breve informe (4 horas): El informe debe contener: introducción breve, descripción del problema, metodología y resultado. Grafique ambas curvas de error versus iteración de entrenamiento. Haga una *matriz de confusión* y analice los resultados. El informe no debe tener una extensión mayor a 10 páginas. Sea breve y conciso.

Esta tarea vale por dos ejercicios. Entregue por U-Cursos un archivo `.zip` con: código fuente, ejecutable (Anakena ó Windows) informe y clasificación de los registros del archivo `casosnuevos`.

Datos:
$X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$
$D = \{\mathbf{d}^{(1)}, \mathbf{d}^{(2)}, \ldots, \mathbf{d}^{(N)}\}$
Arquitectura:
Entradas: N
Unidades Ocultas: M
Salidas: L



ENTRENAR(X, D, μ, ϵ):
Inicializar Parámetros:
$\mathbf{w} \leftarrow$ VECTORALEATORIO$(NM + M)$
$\widetilde{\mathbf{w}} \leftarrow$ VECTORALEATORIO$(ML + L)$
Iterar hasta que E sea inferior a ϵ:
Do {
  Inicializar e, $\dot{\mathbf{w}}$, $\dot{\widetilde{\mathbf{w}}}$:
  $E \leftarrow 0$, $\dot{\mathbf{w}} \leftarrow \mathbf{0}$, $\dot{\widetilde{\mathbf{w}}} \leftarrow \mathbf{0}$,
  Calcular e, $\dot{\mathbf{w}}$, $\dot{\widetilde{\mathbf{w}}}$:
  For $n = 1, \ldots, N$ {
    $\mathbf{x} \leftarrow \mathbf{x}^{(n)}$, $\mathbf{d} \leftarrow \mathbf{d}^{(n)}$
    Calcular e:
    $\forall j, \quad a_j \leftarrow \sum_{i=0}^{N} w_{ji} x_i$
    $\forall j, \quad z_j \leftarrow 1/(1 + \exp -a_j)$
    $\forall k, \quad y_k \leftarrow \sum_{j=0}^{M} \tilde{w}_{kj} z_j$
    $e \leftarrow -\frac{1}{N} \sum d_k \ln y_k + (1 - d_k) \ln(1 - y_k)$
    $E \leftarrow E + e$
    Calcular $\dot{\mathbf{w}}$, $\dot{\widetilde{\mathbf{w}}}$:
    $\dot{y}_k \leftarrow \frac{y_k - d_k}{N y_k (1 - y_k)}$
    $\forall k, j, \quad \dot{\tilde{w}}_{kj} \leftarrow \dot{\tilde{w}}_{kj} + z_j \dot{y}_k$
    $\forall j, \quad \dot{z}_j \leftarrow \sum_{k=1}^{L} \tilde{w}_{kj} \dot{y}_k$
    $\forall j, \quad \dot{a}_j \leftarrow z_j (1 - z_j) \dot{z}_j$
    $\forall j, i, \quad \dot{w}_{ji} \leftarrow \dot{w}_{ji} + x_i \dot{a}_j$
  }
  Ajustar Parámetros:
  $[\mathbf{w}, \tilde{\mathbf{w}}] \leftarrow [\mathbf{w}, \tilde{\mathbf{w}}] - \mu [\dot{\mathbf{w}}, \dot{\tilde{\mathbf{w}}}]$
} While $(E > \epsilon)$