Práctica 9 Introducción al Aprendizaje Automático

Jose Joaquín Rodríguez García y Araceli Teruel Domenech

Universidad Politécnica de Valencia
Master en Big Data Analytics

Octubre 2017

1. Primera Tarea

1.1. Enunciado

Repasad el códgio del siguiente programay ejecutadlo repetidas veces para ver la capacidad de las redes neuronales como clasificadores.

text-ann.py

Este programa Python hace uso de la implmentación de las redes facilitada en la carpeta *ann*. El estudio del código de las redes neuronales es opcional y se recomienda que cada persona lo realice por su cuenta si dispone de tiempo. Tener que implementar las redes excede los objetivos del presente curso.

Sí que se pide en esta tarea que los alumnos prueben diferentes topologías de la red y diferentes tipos de función de activación en cada una de las capas. Los tipos de activación disponibles son: **linear**, **llinear rectified**, **lbinary**, **lsigmoid**, **ltanh** y **lsoftmas**.

Se recomienda utilizr softmax y linear únicamente en la capa de salida. El primero para clasificación y el segundo para regresión. Para el ejemplo de esta tarea, el problema XOR, se recomienda poner a verdadero la variable

xor_example = True plot_examples = True

para que se representen las muestras del training set por pantalla, al menos para una ejecución. De esta manera uno se puede hacer una idea del problema.

Deben probarse distintos tamaños para la capa interna, de 2 a 512 neuronas, se pueden probar las potencias de 2, por ejemplo, y ver qué efecto tiene el número de neuronas

ocultas. Las capas de entrada y salida tienen fijado el tamaño, la de entrada según la dimensionalidad de las muestras. La de salida según el número de clases a distinguir.

Deben probarse distintas funciones de activación para la capa oculta, todas salvo **linear** y **softmax**, y dado que es un problema de clasificación, para la salida pueden probarse las siguientes: **linear**, **softmax**, **tanh** y **sigmoid**.

Se recomienda repasar el código de la función **to_one_hot()** para ver cómo preparar un vector de salida para la red según la clase a la que pertenezca la muestra utilizada para entrenamiento en ese instante.

Como última prueba sobre el problema del XOR, las muestras se han separado mediante con un margen entre cada cuadrante.

margin=1.0

Cambiad el valor del margen a 0 para que las muestras estén juntas, apenas separadas por los ejes de coordenadas, y observad cómo se comportan las redes neuronales y cómo se comporta un clasificador del tipo KDE que se ha estado utilizando como referencia. ¿A qué tipo de clasificador afecta más el hecho de que las muestras no estén separadas por cierto margen?

¿Qué topología de la red y qué funciones de activación se adaptan mejor a cada caso? Variad el número de muestras, por ejemplo **n_samples** igual a 50,100,200,500,1000,2000

¿Qué efecto tiene el número de muestras con respecto a las redes neuronales ya al clasificador KDE cuando las muestras no están separadas vía un margen?

1.2. Resolución

2. Segunda Tarea

2.1. Enunciado

Seguid trabajando con el mismo programa, pero en este caso poned a falso la variable xor_example para que automáticamente se ejecute el código correspondiente a la generación de muestras siguiendo distribuciones normales o de Gauss.

Comprobad el efecto de los cambios propuestos en la tarea anterior para este caso.

Dado que las muestras se general siempre de manera aleatoria, que sirva la precisión obtenida con el clasificador de tipo KDE como referente para ver qué configuraciones de las redes van bien y cuando no es necesario aumentar en número de neuronas de la capa

oculta.

2.2. Resolución

3. Tercera Tarea

3.1. Enunciado

A partir del código anterior más el código utilizado en prácticas anteriores, aplicad las redes neuronales sobre el dataset MNIST.

Por el tamaño del dataset y por la alta dimensionalidad de las muestras el entrenamiento no será lento. Aunque debe probarse sin transformaciones lanzando un proceso que tardará bastante (esto debe de hacerse fuera del laboratorio), se propone, qpara poder probar en el laboratorio, reducir la dimensionalidad mediante PCA y utilizar un conjunto de muestras reducido para la fase de aprendizaje, por ejemplo 1000 y también probad con 2000-

Probad distintas configuraciones de la red para ver cual es la más adecuada a este problema.

3.2. Resolución

4. Cuarta Tarea

4.1. Enunciado

Se trata del dataset utilizado en la tarea 3 de la práctica sobre SVM. Esta tarea consiste en aplicar las redes neuronales sobre el mencionado dataset realizando las mismas pruebas de configuraciones de las redes como se ha hecho en las tareas anteriores. Partid del código disponible en la Web de Scikit Learn y que se utilizó en la anterior práctica.

4.2. Resolución