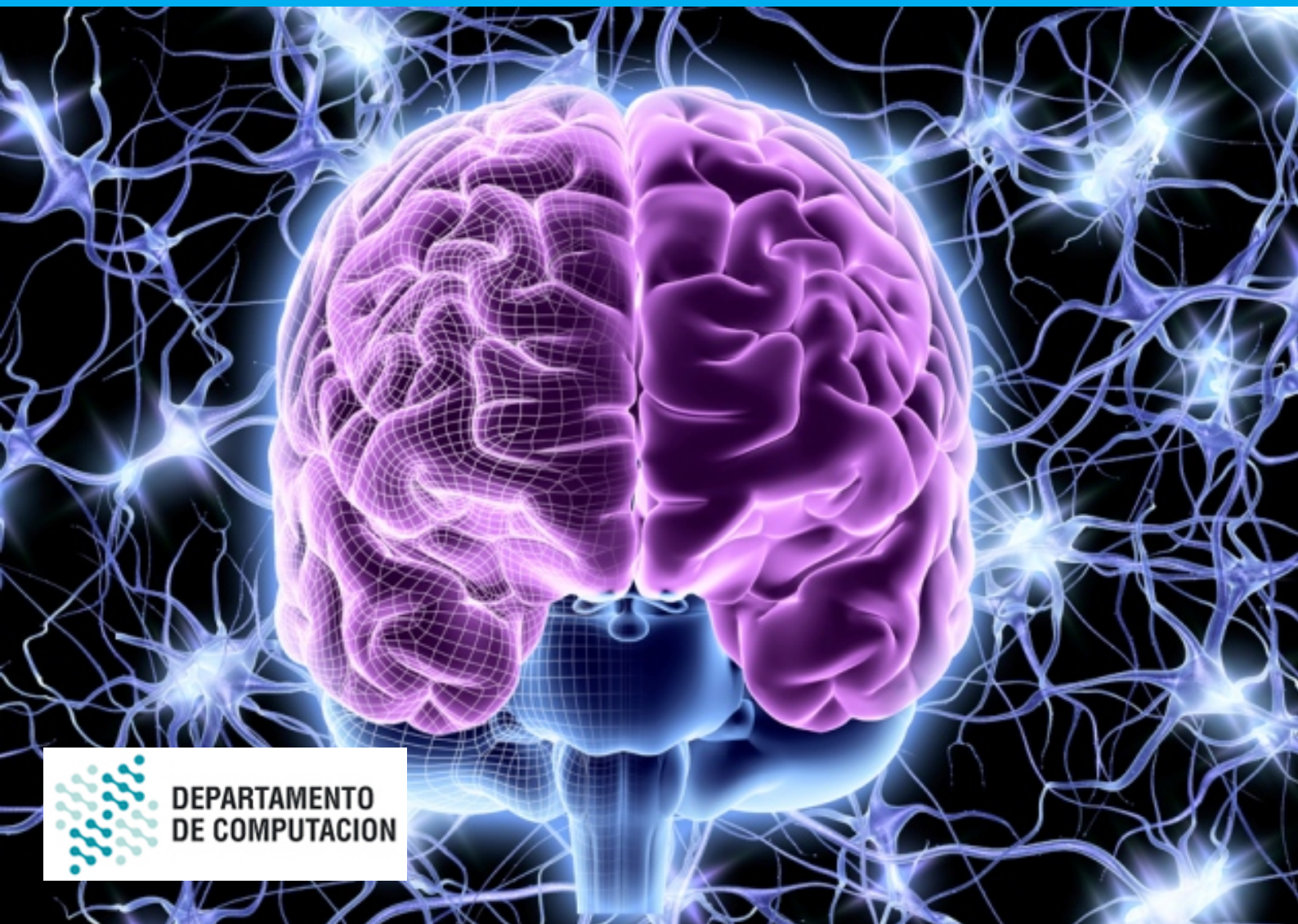


TP 2

Aprendizaje Hebbiano y SOM

Redes Neuronales

Aprendizaje no supervisado



Introducción

1.1. Detalles de la entrega

La entrega deberá consistir al menos de un programa ejecutable y un informe escrito.

El programa ejecutable deberá permitir seleccionar un conjunto de datos y tener la opción de trabajar con un red neuronal nueva o ya entrenada. Si se utiliza una red nueva deberá ser posible entrenarla con los datos seleccionados más los parámetros necesarios. Si se utiliza una red ya entrenada se deberán poder utilizar los datos seleccionados para testarla.

En ambos casos el programa deberá mostrar de forma clara el desempeño del modelo. Si las opciones son pasadas por línea de comando deberán estar claramente especificadas. También es posible entregar un ejecutable distinto para cada problema.

En caso de que existan dificultades en la ejecución del programa para la evaluación, el trabajo puede llegar a ser rechazado. Por esto se recomienda enfáticamente utilizar un lenguaje interpretado multiplataforma y testear su ejecución en distintas máquinas antes de la entrega. En otro caso se deberán especificar instrucciones detalladas para la instalación y/o ejecución del mismo y de las librerías que utilice.

El informe deberá ser conciso y podrá contener los siguientes tópicos además de los que consideren necesarios para que sea claro:

- Breve introducción al problema.
- Descripción de cómo se implementó la solución.
- Ejemplos de instancias de entrenamiento (gráficos de error, datos, pesos, etc).
- Ejemplos de instancias de validación (gráficos de error, datos, pesos, etc).
- Detalle de las opciones que acepta el programa y su modo de uso.
- El código fuente impreso del programa.
- Decisiones tomadas y su justificación.
- Conclusiones si las hubiere.

El informe debe entregarse impreso y además en un archivo comprimido junto al código ejecutable a la dirección:

entregas.redneu@gmail.com

2

Problemas

2.1. Introducción

El objetivo de este trabajo práctico es utilizar dos modelos distintos de redes neuronales artificiales sobre el mismo conjunto de datos. Los modelos estarán basados en aprendizaje hebbiano no supervisado y redes autoorganizadas y serán utilizados para una tarea de clasificación. Los datos pertenecen a un problema real y es posible que contengan errores. Se espera que respetando las consignas se construyan modelos adecuados para cada problema, que puedan aprender todas las instancias de entrenamiento, y que tengan una capacidad aceptable para generalizar. Tanto el diseño de los modelos, el entrenamiento y testeo, y los métodos para presentar los resultados deberán ser documentados y entregados en un informe en donde se deberán justificar las decisiones tomadas y explicados los resultados obtenidos.

2.2. Problema

El conjunto de datos consiste en documentos con descripciones de texto correspondientes a compañías Brasileñas clasificadas en nueve categorías distintas. Los textos originales fueron preprocesados para obtener un tipo de representación conocida como Bolsa de Palabras (Bag-Of-Words / BOW). En este tipo de formato cada documento es representado mediante un vector en donde cada dimensión corresponde a una palabra específica y su valor está dado por la cantidad de apariciones de esa palabra en el documento. Para mejorar la representación las palabras más comunes (artículos, preposiciones, etc) no son tenidas en cuenta. Notar que al representar un documento de esta forma no se está teniendo en cuenta el orden de las palabras, solo su frecuencia de aparición.

El conjunto de datos se encuentra en formato CSV sin encabezado y contiene 900 entradas distribuidas uniformemente entre las 9 categorías. Cada entrada representa un documento y consiste en el número de categoría (1 a 9) más 850 atributos correspondientes a frecuencias de palabras. Tener en cuenta que el conjunto de datos es altamente disperso (más del 99% son ceros) y que el número de categoría corresponde a la actividad principal de la empresa, no necesariamente la única.

El problema a resolver será clasificar automáticamente los documentos utilizando dos modelos distintos de aprendizaje no supervisado. Notar que por tratarse de aprendizaje no supervisado para el entrenamiento solo se deben utilizar los valores de las frecuencias de palabras, la información de la categoría solo será utilizada para verificar la clasificación hecha por los modelos.

2.2.1. Reducción de dimensiones

Construir un modelo de red neuronal artificial que permita reducir la alta dimensionalidad de las entradas en el conjunto de datos a sólo 3 dimensiones. Para esto el modelo debe poder ser entrenado mediante las reglas de aprendizaje de Oja y de Sanger. Una vez realizado el entrenamiento representar gráficamente las posiciones en el espacio R^3 que el modelo le asigna a cada documento.

Para realizar la representación gráfica de los resultados tener en cuenta que:

- Se debe diferenciar claramente cada categoría (por ejemplo con colores).

- Se deben diferenciar claramente los datos de entrenamiento y validación (por ejemplo con distintos tipos de marcadores).
- Para apreciar la distribución espacial pueden ser necesarias varias vistas.
- Se deben verificar los resultados mediante varias instancias de entrenamiento.

¿Qué interpretación le puede dar a estos resultados? ¿Qué tan buenos son estos modelos específicos para hacer una clasificación de los documentos? ¿Cómo los utilizaría?

2.2.2. Mapeo de características

Construir un modelo de mapeo de características auto-organizado que clasifique automáticamente los documentos en un arreglo de dos dimensiones. Tener en cuenta que para poder realizar una buena clasificación se debe contar con suficientes unidades de salida y con suficientes instancias de datos por unidad. Construir otro modelo equivalente que utilice como entrada los datos proyectados en las primeras 3 componentes principales, y otro modelo que utilice como entrada los datos proyectados en las primeras 9 componentes principales.

Una vez realizado el entrenamiento representar gráficamente en un mapa de características los resultados señalando:

- Para cada unidad de salida cuál es la categoría que más la activa (por ejemplo con distintos colores).
- Cuál es la diferencia entre los datos de entrenamiento y validación (por ejemplo en distintas figuras).
- Comparar los resultados variando la cantidad de unidades de salida.
- Comparar los resultados variando los parámetros de entrenamiento.

¿Qué interpretación puede darle a los resultados obtenidos? ¿Cómo se comparan estos resultados con los obtenidos anteriormente? ¿Cuál es la arquitectura de red y parámetros de entrenamiento que utilizaría? ¿Qué resultados esperaba obtener con datos de testeo con los que el modelo no fue entrenado? ¿Hubo diferencia en el tiempo de entrenamiento utilizando los datos de dimensión reducida? ¿Y en la calidad del resultado?