

Inteligencia Artificial (SI404)

CICLO 2020-1

EXAMEN FINAL

TODAS LAS SECCIONES

|  |  |  |
| --- | --- | --- |
| *Proyecto 1*  *20* | *=* | *Nota*  *20* |

Profesores: Heider Ysaias Sanchez, Hugo David Calderon Vilca, Pedro Shiguihara

Secciones: SW71, WX71, WV71 y CC61

**Apellidos y nombres:** Miramira Morales Bryan Antony

**Código:** u201710061

**Fecha:** 08 de julio de 2020, a horas: 11:00 hasta 13:00

**Instrucciones:**

1. El examen consta de 5 preguntas, y tendrá 110 minutos para resolverlas.
2. Las preguntas son tipo respuesta breve, respuesta de archivo y la entrega de su respuesta es a través de respuesta con archivo en blackboard
3. Cada examen cuenta con un equipo académico, el cual estará conectado durante los primeros 20 minutos del examen.
4. El alumno debe dedicar los primeros 20 minutos a revisar las preguntas del examen y de presentarse alguna duda enviar un correo al(los) profesor(es)
5. CALDERON VILCA, HUGO DAVID al correo [pcsihcal@upc.edu.pe](mailto:pcsicdal@upc.edu.pe)
6. De no recibir respuesta del equipo académico, o tener algún inconveniente adicional pasado los primeros 20 minutos, puede comunicarse con los profesores 9322 - GONZALEZ VALENZUELA, RICARDO EUGENIO a [pcsirgon@upc.edu.pe](mailto:pcsicjar@upc.edu.pe)
7. Los profesores en mención, solo recibirán correos provenientes de las cuentas UPC, de ninguna manera se recibirán correos de cuentas públicas.
8. Ante problemas técnicos, debe de forma obligatoria adjuntar evidencias del mismo, como capturas de pantalla, videos, fotos, etc. Siendo requisito fundamental que, en cada evidencia se pueda apreciar claramente la fecha y hora del sistema operativo del computador donde el alumno está rindiendo el examen.
9. Los problemas técnicos se recibirán como máximo 15 minutos culminado el examen.
10. El examen consiste en un proyecto aplicando Redes Neuronales y Procesamiento de Lenguaje Natural.
11. El examen es resuelto en grupo de 2 a 3 máximo.
12. La entrega del examen final es miércoles 8 de julio 13 horas en blackboard, cada uno realiza la entrega.
13. Por la coyuntura la exposición será grabada y puesta en YouTube con una duración de 15 minutos, colocar su nombre completo en el video de forma permanente durante la exposición de cada integrante. En la exposición deberá fundamentar: Problema, etiquetación de la dataset, diseño de las RN, ejecución del aplicativo e interpretación de resultados. Utilizar esquemas, gráficos, diagramas que permitan explicar el procedimiento).
14. Entregable: + Carpeta: EB\_XXXX (XXXX es código de estudiante) con contenido: + Examen EB\_XXXX.docx + Carpeta Proyecto (código fuente del proyecto) + Video.txt (link del video).
15. Comprimir o empaquetar (EB\_XXXX.zip) y adjuntar como actividad en la hora y fecha indicada. Nota. - A pesar de que el trabajo fue realizado de manera grupal, cada uno deberá presentar el examen final, de hecho en el informe abajo están los nombres de los integrantes del grupo.

**Nota. -** A pesar de que el trabajo fue realizado de manera grupal, cada uno deberá presentar el examen final, de hecho en el informe abajo están los nombres de los integrantes del grupo.

**INFORME DE APLICACIÓN**

**TÍTULO DE LA APLICACIÓN: Análisis de sentimientos de fanáticos de básquetbol**

Nombres y apellidos del integrante: Bryan Antony Miramira Morales

Nombres y apellidos del integrante: Fernando Chahua Lopez

Nombres y apellidos del integrante: Diego Eduardo Bustos Bustos

**Enunciado**

Desarrollar una aplicación en modo consola o GUI que clasifique textos a partir de un corpus determinado previamente. Con un problema y objetivo definido por el grupo de trabajo, (ejemplo: detección de spam, análisis de sentimiento, otros), describir los detalles de la aplicación en este informe de acuerdo con las secciones: I, II, III, IV y V. En cuanto a la implementación recomendable utilizar Python como lenguaje de programación.

Pasos o etapas de desarrollo e implementación (recomendable cumplir en ese orden)

1. Describir el problema y objetivo
2. Definir la técnica de PLN a utilizar para representar un texto como un vector de características.
3. Etiquetar al menos 50 oraciones o textos, relacionados al tema definido, manualmente y aplicar la técnica de PLN para representarlo como un vector de características, formando un dataset pequeño.
4. Identificar al menos 1000 oraciones o textos, relacionados al tema definido, no etiquetados y aplicar la técnica de PLN para aproximarlos a los textos etiquetados usando redes neuronales de aprendizaje no supervisado, de esta manera formar un dataset grande.
5. Aplicar al menos tres tipos de arquitecturas de redes multicapa distintas a fin de de clasificar automáticamente los textos y determinar su eficiencia en términos de accuracy o exactitud.

La entrega es código fuente del aplicativo y documento escrito en la siguiente estructura formalmente:

1. **(3 puntos) Descripción del problema, objetivo y dataset.**
2. Describe el problema

El saber lo que piensa y opina la gente es un amplio tema de estudio, muchas empresas dependen de estas opiniones para tomar decisiones ya sea de productos o de su negocio en sí. Ahora entramos un poco en el tema deportivo, para ser más exactos en deportes como el basquetbol, baseball y futbol; los patrocinadores, socios y dueños de los equipos tienen un gran interés en lo que opina la afición, sobre si ven bien el rendimiento equipo, si están jugando mal, si no les agrada un jugador, si toman malas decisiones o si simplemente odian como hacen las cosas. Por ello todas esas opiniones pueden ser fundamentales para tomar decisiones tanto deportiva como económicamente. Somos consientes de como el deporte influye mucho en nosotros y el impacto que tienen, es considerando uno de los negocios que más masas mueve actualmente, así que dentro de todo ese negocio una aplicación que determine automáticamente si los comentarios u opiniones de los aficionados son positivos o negativos sería muy util, facilitando así la toma de decisiones de dichas entidades no tienen el tiempo para preguntar ni revisar la inmensa cantidad de opiniones.

1. Objetivo de la solución

Nuestra propuesta intentara resolver esta necesidad al sugerir un análisis de sentimientos de las opiniones de los aficionados en Twitter, de esta manera se podrán clasificar los buenos y malos comentarios haciendo uso de Procesamiento de Lenguaje Natural, Redes SOM y multicapa BackPropagation, así a los socios, dueños y patrocinadores se les facilitara la toma de decisiones importantes como decidir si seguir apoyando o no a tal equipo.

1. Presentación del dataset o diseño del dataset explicando el método de recolección de textos y su representación como vector de características usando PLN

El dataset que usaremos se encuentra en Kaggle, una página de datasets libres para estudio e investigación, el link del dataset se encuentra en la bibliografía, para nuestra propuesta usaremos: pos\_tweets.txt y neg\_tweets.txt, de los cuales usamos las 1000 entradas etiquetadas, para entrenar la red SOM y otros mil restantes sin etiquetar para probar con la red multicapa BackPropagation.





Nuestro vector de características será un Bag Words que contará la cantidad de repeticiones de cada palabra ya sea positiva o negativa, posee la siguiente estructura:





En total son 40 palabras, 19 positivas y 21 negativas. Por lo que al final se generara un arreglo de 40 números que serán las repeticiones de cada palabra si existen en el texto.

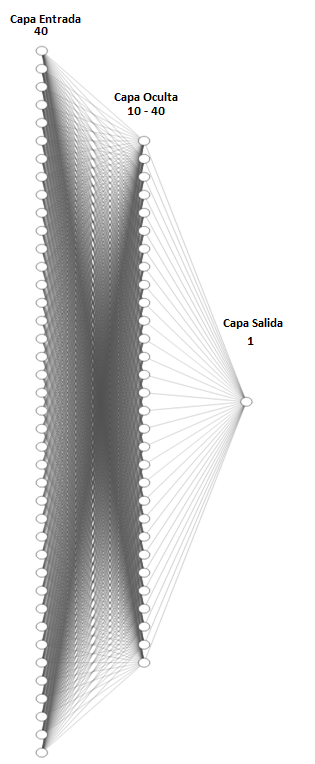
1. **(5 puntos) Metodología de solución:**
2. Diseño de la red neuronal **no supervisada** para etiquetado de textos, descripción de entradas y salidas para el agrupamiento.
3. Diseño de la arquitectura de la Red Neuronal multicapa **supervisada**, descripción de entradas y salidas para la clasificación.

Nuestra red neuronal BackPropagation que usaremos tendrá una **capa de entrada**, una capa oculta y una capa de salida, la cantidad de neuronas en la capa de entrada será 40, la misma cantidad de datos de entrada de la red SOM.

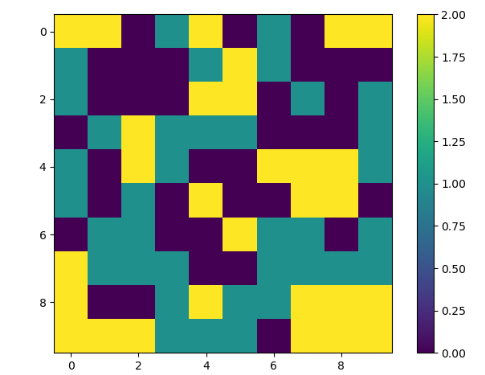
Los datos de entrada para la red neuronal tendrán el siguiente formato: [0,0,0,0,1,0,0,0, 1, ….,0,1,1], un arreglo de frecuencias de rango 40 que fue convertido usando el Bag Words que mencionamos anteriormente. El programa permitirá ingresar por consola una oración cualquiera como “this coach sucks, i prefer the last coach” y la transformará un arreglo de frecuencias para ser procesado por la red neuronal.

Para la **capa oculta** tenemos planeado 4 arquitecturas, una con 10 neuronas, otra con 20 neuronas, otra con 30 neuronas y una con la misma cantidad de neuronas que la capa de entrada, es decir 40 neuronas. Esperamos probar y encontrar la arquitectura mas optima de esas 4.

Finalmente, en la c**apa de salida** que tendrá una única neurona que devolverá como resultado un numero entre 0 y 1, para clasificar estos resultados en el código se implemento un traductor que devolverá si la oración es POSITIVA – NEGATIVA – NEUTRAL.



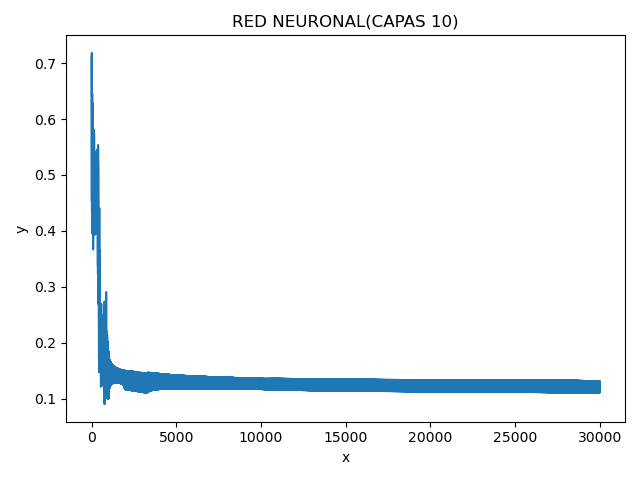
1. **(2 puntos) Describir, interpretar los resultados de las pruebas, gráficos, errores.**
   1. Resultados de la red neuronal no supervisada



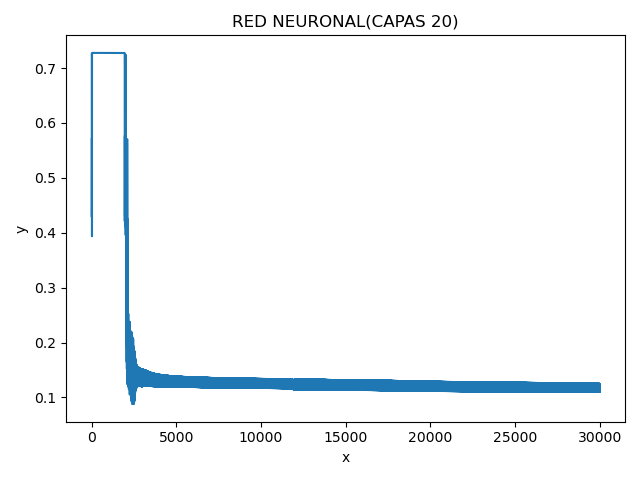
* 1. Resultados de la red neuronal supervisada

Ahora los resultados de la red neuronal que obtuvimos, probamos con las 4 arquitecturas mencionadas anteriormente y se hallo el error cuadrático medio de cada una, para graficarlo posteriormente.

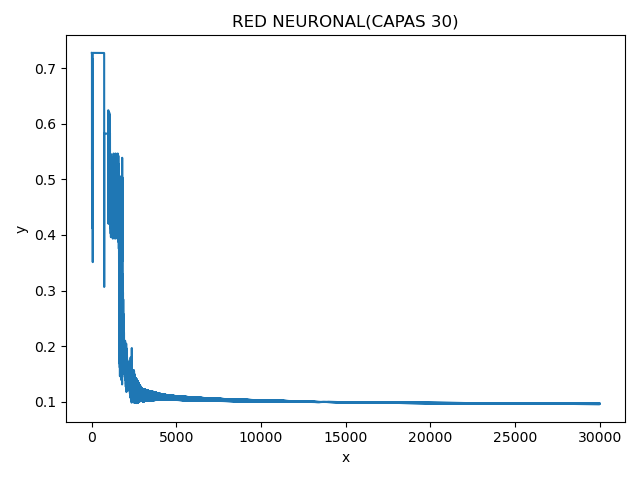
El grafico del error medio con las épocas, en la arquitectura con 10 neuronas en la capa oculta resulto:



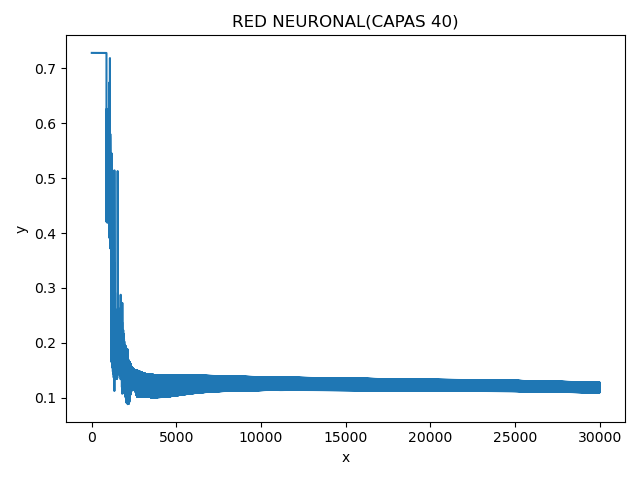
El grafico del error medio con las épocas, en la arquitectura con 20 neuronas en la capa oculta resulto:



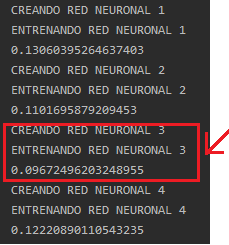
El grafico del error medio con las épocas, en la arquitectura con 30 neuronas en la capa oculta resulto:



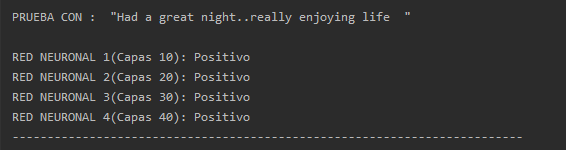
El grafico del error medio con las épocas, en la arquitectura con 40 neuronas en la capa oculta resulto:

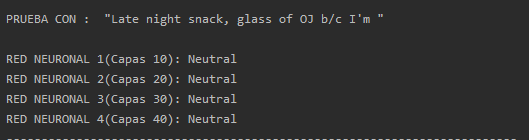


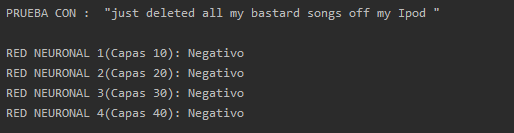
Finalmente comparamos y analizamos los gráficos generados llegando a la conclusión de que la arquitectura con 30 neuronas en su capa oculta normalizo mejor los resultados y pesos de la red.



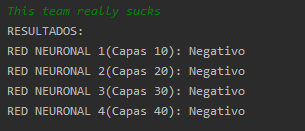
Además, fue la que devolvió el menor error final. Además, tenemos unas comparaciones con los resultados de todas las arquitecturas en algunas oraciones:

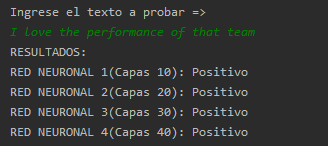






Así que se decidió usar esa arquitectura como predeterminada para clasificar nuevas oraciones que ingresaremos a través de la consola, devolviendo los siguientes resultados:





Finalmente se concluyo que la red es capaz de definir el tipo de texto que se ingresa, la estructura final

1. **Referencias bibliográficas (Lo necesario y lo que ha utilizado)**

* Sentimental Analysis NLP. (2018, 2 diciembre). Recuperado 7 de julio de 2020, de <https://www.kaggle.com/ashishpatel26/sentimental-analysis-nlp>
* R. (2017, 6 noviembre). Los aficionados y la importancia del fútbol. Recuperado 8 de julio de 2020, de <https://codigo.pe/los-aficionados-y-la-importancia-del-futbol-columna/>
* Trullols, J. (2016, 15 diciembre). El cuidado del aficionado y el negocio del cliente. ¿El dilema del deporte actual? Recuperado 8 de julio de 2020, de <http://lajugadafinanciera.com/aficionado-cliente-dilema-del-deporte-actual/>

**RUBRICA DE LA APLICACIÓN**

**(PLN y Redes Neuronales Supervisadas y No supervisadas)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | **Pte.** |
| **Categoría** | **Sobresaliente** | **Aceptable** | **Principiante** |  |
| Describe el problema y proyecta el objetivo de la solución.  Diseña el método para la representación de texto a un vector de características usando PLN. | Describe el problema y proyecta el objetivo de la solución claramente.  Diseña el método para la representación de texto a un vector de características usando PLN, se entiende claramente su descripción. | Describe el problema y proyecta el objetivo de la solución.  Diseña el método para la representación de texto a un vector de características usando PLN. | Describe el problema y proyecta el objetivo de la solución con deficiencias, pero aún se entiende.  Diseña el método para la representación de texto a un vector de características usando PLN con deficiencias, pero aún se entiende. |  |
| **Puntaje** | **3 punto max** | **2 puntos max** | **1 puntos max** |  |
| Diseña una red neuronal de aprendizaje no supervisado para etiquetado y diseña la arquitectura de 3 redes neuronales para aprendizaje supervisado. | Describe claramente la arquitectura, entradas, capas, salida de la red neuronal no supervisada, así como los clústeres.  Describe claramente la arquitectura, entradas, capas, salida de la red neuronal supervisada, así como la clase. | Describe la arquitectura, entradas, capas, salida de la red neuronal no supervisada, así como los clústeres.  Describe la arquitectura, entradas, capas, salida de la red neuronal supervisada, así como la clase. | Describe la arquitectura, entradas, capas, salida de la red neuronal no supervisada, así como los clústeres con deficiencias, pero aún se entiende.  Describe claramente la arquitectura, entradas, capas, salida de la red neuronal supervisada, así como la clase con deficiencias, pero aún se entiende. |  |
| **Puntaje** | **5 puntos max** | **3 punto max** | **2 puntos max** |  |
| Código fuente de la aplicación, claridad y comentarios en el código, realiza las funcionalidades de acuerdo al objetivo. | El código fuente es claro y original o adaptó de acuerdo con el problema, ejecuta la aplicación correctamente. Realiza las funcionalidades de acuerdo al objetivo. | El código fuente es parcialmente claro y original, adaptado al problema, ejecuta la aplicación. Realiza las funcionalidades de acuerdo al objetivo. | No ejecuta la aplicación o no tiene claridad en el código fuente. Pero se entiendo el código y realiza las funcionalidades de acuerdo al objetivo. |  |
| **Puntaje** | **8 puntos max** | **6 puntos max** | **3 punto max** |  |
| Resultados e interpretación | Presenta, describe e interpreta claramente los resultados de las pruebas, de red neuronal no supervisada. así mismo están basados en gráficos los clusters y los errores.  Presenta, describe e interpreta claramente los resultados de las pruebas, de red neuronal supervisada. así mismo están basados en gráficos las clases y los errores). | Presenta, describe e interpreta los resultados de las pruebas, de red neuronal no supervisada. así mismo están basados en gráficos los clusters y los errores.  Presenta, describe e interpreta claramente los resultados de las pruebas, de red neuronal supervisada. así mismo están basados en gráficos las clases y los errores). | Presenta, describe e interpreta con deficiencias los resultados de las pruebas, de red neuronal no supervisada. Pero aun se entiende.  Presenta, describe e interpreta con deficiencias los resultados de las pruebas, de red neuronal supervisada. Pero aún se entiende. |  |
| **Puntaje** | **2 puntos max** | **1 max** | **0.5 puntos max** |  |
| Presenta la exposición en video, explicando claramente: el problema, dataset, diseño de las RNs, ejecución del aplicativo y resultados, Utilizar esquemas, gráficos, diagramas que permitan explicar el procedimiento | Presenta la exposición en video, explicando claramente: el problema, dataset, diseño de las RNs, ejecución del aplicativo y resultados, Utilizar esquemas, gráficos, diagramas que permitan explicar el procedimiento | Presenta la exposición en video, explicando: el problema, dataset, diseño de las RNs, ejecución del aplicativo y resultados, Utilizar esquemas. | Presenta la exposición en video, explicando con deficiencias: el problema, dataset, diseño de las RNs, ejecución del aplicativo y resultados, pero aún se entiende. |  |
| **Puntaje** | **2 puntos max** | **1 punto max** | **0.5 puntos max** |  |

Lima, 8 julio 2020