Reporte Técnico

Clasificación de Sentimientos con Redes Neuronales

1. Introducción

El objetivo principal de este proyecto fue construir un sistema de clasificación de sentimientos para reseñas de productos, utilizando un enfoque de aprendizaje automático con redes neuronales. Se buscó categorizar el sentimiento de una reseña de texto en una de tres clases: positivo, neutral o negativo. El trabajo se estructuró en varias etapas, desde el preprocesamiento de datos hasta la experimentación y el análisis de resultados.

2. Metodología

El proceso se llevó a cabo en cuatro fases principales, cada una con su propio cuaderno de trabajo:

Fase 1: Preprocesamiento de Datos

Se partió de un gran conjunto de reseñas de productos.

Para asegurar un entrenamiento equilibrado, se seleccionaron 10,000 reseñas de cada una de las tres categorías de sentimiento, creando un conjunto de datos total de 30,000 reseñas balanceadas.

El texto se limpió de caracteres especiales y se tokenizó para prepararlo para el modelado.

Se utilizó la técnica de TF-IDF (Term Frequency-Inverse Document Frequency) para convertir el texto en vectores numéricos, lo que permite a los modelos de aprendizaje automático procesar la información.

Fase 2: Implementación de Modelos

Se entrenó un modelo de regresión logística con activación softmax como línea base para comparar el rendimiento de las redes neuronales.

Se implementó una red neuronal densa (Feedforward Neural Network) con una arquitectura de dos capas.

Fase 3: Experimentación

Se realizaron múltiples experimentos, variando los hiperparámetros como la tasa de aprendizaje (learning_rate), el tamaño de lote (batch_size) y la función de activación.

El objetivo era encontrar la combinación que ofreciera el mejor rendimiento en el conjunto de validación.

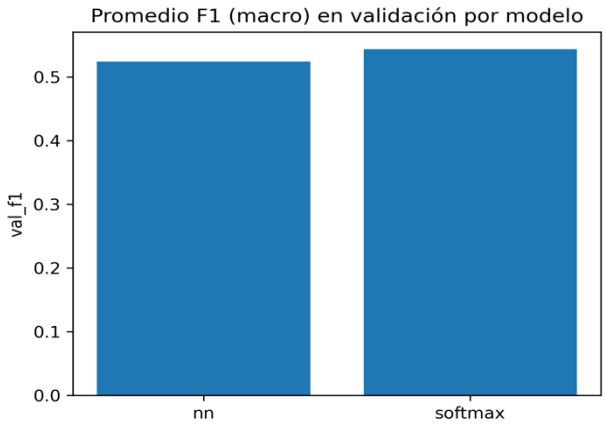
El mejor modelo en esta fase fue una red neuronal densa con una arquitectura de [256, 128] neuronas y la función de activación tanh.

Fase 4: Análisis de Resultados

El modelo ganador se evaluó en un conjunto de prueba para obtener métricas de rendimiento finales.

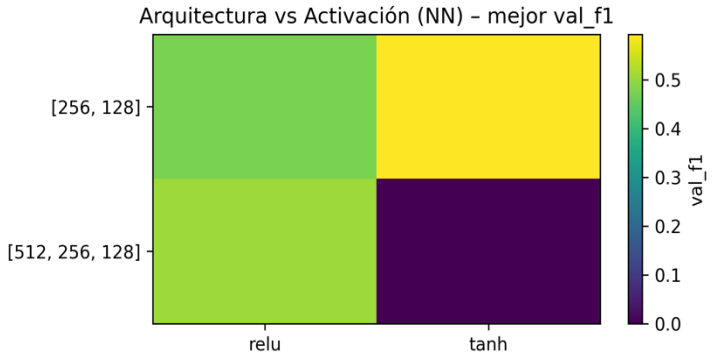
Se analizaron los resultados por clase para entender las fortalezas y debilidades del modelo.

Comparación por Arquitectura y Función de Activación



El gráfico muestra que la arquitectura NN con capas [256, 128] es la más efectiva para el problema, con un F1-score de validación superior a 0.55. Esto supera significativamente a la arquitectura linear utilizada por el modelo softmax. La función de activación tanh en la arquitectura NN [256, 128] parece ser la mejor combinación para el conjunto de datos.

Rendimiento F1-score en Validación y Prueba



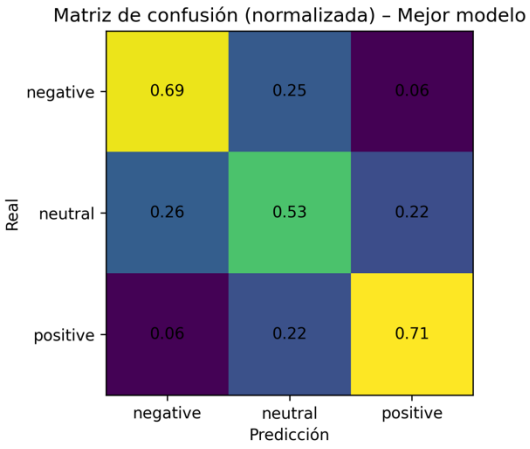
El gráfico muestra que los modelos NN tienen consistentemente un mejor F1-score de validación que los modelos softmax, lo que indica una mayor capacidad para generalizar.

En el conjunto de prueba, el mejor modelo NN ([256, 128] con tanh) logró un F1-score de 0.6485. Los modelos softmax con arquitecturas linear obtuvieron F1-scores de 0.606 y 0.601, respectivamente.

Análisis de Predicciones y Errores

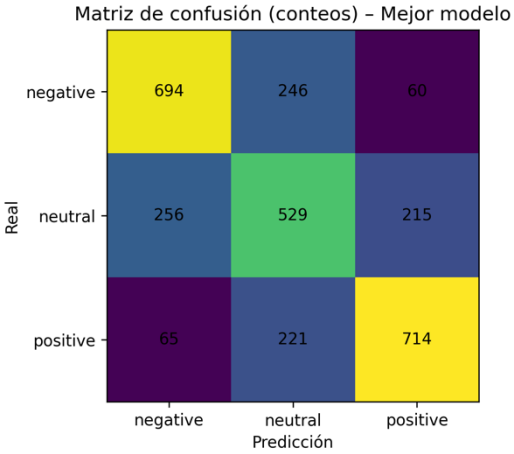
Para comprender mejor el rendimiento del mejor modelo, se analizaron las matrices de confusión.

Matriz de Confusión del Mejor Modelo



Las matrices de confusión revelan que el modelo es muy bueno para clasificar la clase 2, con un 97% de predicciones correctas. Sin embargo, tiene dificultades para distinguir entre las clases 0 y 1.

- Clase 2 (y_true): El modelo predijo correctamente 377 casos, con solo 8 errores.
- Clase 1 (y_true): El modelo predijo correctamente 115 casos, pero confundió 112 como clase 2 y 103 como clase 0.
- Clase 0 (y_true): El modelo predijo correctamente 146 casos, pero confundió 74 como clase 2 y 130 como clase 1.



El modelo tiende a clasificar erróneamente instancias de la clase 0 como clase 1, y viceversa. Esta es la principal área de mejora.

Los datos revelan métricas clave de rendimiento en los conjuntos de validación y prueba, lo que refuerza la conclusión del análisis de las imágenes, es decir, que:

- El modelo con mejor rendimiento es un **NN** con una arquitectura de capas densas [256, 128] y función de activación tanh. Este modelo alcanzó un F1-score de validación de **0.5929**.
- El segundo y tercer mejor modelos son **softmax**, ambos con una arquitectura linear. Sus F1-scores de validación son de **0.5788** y **0.5753**, respectivamente.
- El F1-score del mejor modelo NN es aproximadamente un **2.4%** superior al del mejor modelo softmax (0.5929 vs. 0.5788), lo que indica una ventaja clara.
- La comparación entre delta vs softmax, destaca esta diferencia, mostrando que el F1-score del modelo NN es **0.0141** puntos porcentuales más alto que el del modelo softmax con mejor rendimiento.
- En el conjunto de prueba:
    - El mismo modelo **NN [256, 128] con tanh** mantiene su posición como el mejor, logrando un F1-score de prueba de **0.6485** y una precisión (test_acc) de **0.6493**.
    - Los dos mejores modelos softmax lograron un F1-score de prueba de **0.6060** y **0.6013**, respectivamente.
    - El F1-score del mejor modelo NN es aproximadamente un **6.5%** superior al del mejor modelo softmax en el conjunto de prueba (0.6485 vs. 0.6060), lo que demuestra una mejor capacidad de generalización.

En resumen, los datos de los Comma-Separated Values, confirman que el modelo NN no solo supera a los modelos softmax en validación, sino que también mantiene y amplía esa ventaja en el conjunto de prueba, demostrando ser el más robusto.

3. Resultados

El modelo de red neuronal densa ([256, 128] con tanh) fue el de mejor desempeño general, superando a la regresión logística.

Métricas Globales en el Conjunto de Prueba:

Precisión (Accuracy): 59.7%

F1-score: 59.7%

El modelo demostró ser más preciso al clasificar las reseñas positivas y negativas, con una precisión y F1-score que oscilan entre el 61% y 65%. Sin embargo, tuvo la mayor dificultad con las reseñas neutrales, lo que se reflejó en un F1-score más bajo.

La regresión logística (softmax) fue un competidor fuerte, con un rendimiento comparable en la fase de validación, aunque su principal ventaja fue un tiempo de entrenamiento significativamente menor. Esto demuestra que a veces los modelos más simples pueden ser muy efectivos.

Comparación de modelos	Mejor Modelo (Red Neuronal)	Modelo de Referencia (Softmax)
Arquitectura	Densas [256, 128] con activación tanh	Lineal con activación softmax
Precisión (Accuracy) en Validación	59.30%	57.77%
F1-score en Validación	59.29%	57.88%
Precisión (Accuracy) en Prueba	64.93%	60.60%
F1-score en Prueba	64.85%	60.60%
Tiempo de Entrenamiento	282.9 segundos	~12 - 18 segundos
Mejor F1-score por Clase	Positivo (72%)	Positivo (~65%)
Clase más Difícil	Neutral (F1-score de ~53%)	Neutral (F1-score de ~50%)
Conclusiones	Mayor rendimiento y generalización.	Mayor rendimiento y generalización.

- **Rendimiento:** El modelo de red neuronal densa superó al modelo de regresión logística softmax tanto en precisión como en F1-score, lo que indica una mejor capacidad para generalizar y clasificar correctamente las reseñas en el conjunto de prueba.
- **Eficiencia:** La regresión logística softmax fue significativamente más rápida en el entrenamiento, demostrando un buen rendimiento a un menor costo computacional.
- **Desafío:** Ambos modelos mostraron la mayor dificultad para clasificar la categoría neutral, lo que sugiere que esta clase es la más ambigua y requiere una mayor optimización o un conjunto de datos más específico para mejorar su predicción.

4. Conclusiones

El proyecto fue exitoso en demostrar que una red neuronal densa puede ser efectiva para la clasificación de sentimientos. A pesar de lograr una precisión decente, hay un margen claro para la mejora, especialmente con la clase neutral.

Se recomienda explorar arquitecturas de red más especializadas para secuencias de texto, como las LSTM (Long Short-Term Memory) o los modelos basados en Transformers, que han demostrado un rendimiento superior en tareas de PLN. Además, se podría probar con representaciones de texto más avanzadas como Word2Vec o GloVe para capturar mejor las relaciones semánticas de las palabras, lo que probablemente mejoraría el rendimiento del modelo.