INFORME PROYECTO FINAL REDES NEURONALES: FAKE NEWS

Proyecto realizado por: Lucía Aranda, Ana Martínez y Marta Ripio

1) PROBLEMA A RESOLVER

La desinformación es uno de los principales desafíos de la era digital. Las noticias falsas se propagan rápidamente, generando confusión y afectando negativamente a la sociedad. En este proyecto, desarrollamos un modelo capaz de clasificar si una noticia es verdadera o falsa a partir de su título. Además, si la noticia es clasificada como verdadera y se encuentra en nuestra base de datos, se mostrará su contenido completo.

Para lograrlo, trabajamos con dos conjuntos de datos: **True.csv** (noticias verdaderas) y **Fake.csv** (noticias falsas), ambos obtenidos de kaggle, una plataforma que recopila datasets públicos. Después, los combinamos en un único dataset, al que añadimos una columna ('label') que indica si la noticia es verdadera (1) o falsa (0).

2- SOLUCIÓN PROPUESTA

1) Transformación del Input:

Para que nuestra red entienda el input necesitamos convertir los títulos de las noticias en números ya que las redes solo entienden datos numéricos, para esto usamos el Bag of Words. Esto lo que hace es que cuenta cuántas veces aparece cada palabra así cada título se convierte en un vector de números que representan si aparece o no una palabra, y las que aparecen indica cuántas veces están.
El Bag of Words no normaliza, solo cuenta las palabras, por lo que si una palabra aparece mucho influirá más que otras en el resultado, aunque no sea importante (como pasa con palabras como "the", "is", "are", etc.). Para que esto no suponga un problema en nuestro modelo, utilizamos "stop_words='english'" que se encarga de eliminar estas palabras irrelevantes. Existe otra técnica parecida llamada TF-IDF que sí normaliza, ajustando el peso de las palabras según lo comunes o raras que sean. A pesar de que TF-IDF si que normaliza, hemos preferido usar CountVectorizer porque es más simple, rápido de implementar y suficiente para nuestro caso (los títulos son cortos y las palabras relevantes se distinguen fácilmente).

2) ¿Cómo hemos llegado a la solución final?

Queríamos crear una red simple, que fuese fácil de utilizar, pero a su vez que fuese precisa y obtuviese buenas predicciones. Para ello, hemos decidido hacer una comparación entre el modelo de perceptrón simple y multicapa.
Hemos calculado 5 modelos diferentes. Todos ellos entrenados en 5 épocas con BCE ponderado.
Todos los modelos comparten:

- **Mismo input:** 1 vector de entrada de 5000 características
- **5 épocas** es suficiente para que el modelo aprenda sin sobreajuste
- **Mini-batch Gradient Descent:** Más ruido y acelera convergencia. Entrena usando grupos pequeños de ejemplos, de esta manera balancea velocidad y estabilidad, permite buen rendimiento.
- **BCE ponderado** calcula que cerca están las predicciones de los valores reales (False o True) y aplica un peso ponderado que corrige el desequilibrio de clases. Compensamos que: Nuestro dataset de True (21418) tiene menos filas que el false (23482)
- **Función de las capas ocultas -> RELU:** las capas ocultas ayudan al modelo a descubrir ciertas relaciones más complejas entre los datos (como si una noticia tiene ciertas palabras juntas será más probable que sea falsa). Utilizamos ReLU ya que solo deja pasar valores positivos ignorando los valores inútiles, es muy rápida y ayuda a la red a que aprenda mejor sin atascarse. Con esto conseguimos introducir en nuestro modelo inteligencia y no-linealidad.

- **Función de salida -> SIGMOIDE:** Usamos la función sigmoide en la capa de salida porque convierte el resultado en un valor entre 0 y 1, que se interpreta como la probabilidad de que una noticia sea

verdadera. Si la probabilidad es mayor o igual a 0.5, clasificamos como true; si es menor, como fake.

Técnicas avanzadas de ML:

- **Dropout** de 30%. Cada neurona tiene una probabilidad del 30% de apagarse y sirve para añadir aleatoriedad y reducir el sobreajuste.
- **Regularización.** Utilizada en el optimizador para penalizar automáticamente los pesos grandes del modelo de entrenamiento

Modelos perceptrón simple:
Métricas utilizadas:

- **BCE:** Cuánto más cercana sea la predicción a la etiqueta correcta, menor será el BCE.
- **Accuracy:** Medida de precisión global. Mide el % de predicciones correctas.
- **F1 Score:** Para asegurarnos que no hay clases desbalanceadas (Penaliza FP como los FN). Cuanto más cercano a 1 mejor.

Modelo A: 1 capa oculta de 64 neuronas

- Dropout y regularización
- Métricas: **BCE**-> 0,1193; **ACCURACY** -> 0,9472; F1- **SCORE**:0,95

Modelo B: 0 capas. Implica un modelo lineal sin capas ocultas

- Métricas: **BCE**-> 0,2222; **ACCURACY** -> 0,9386; **F1- SCORE**: 0,9376

El Modelo A, gracias a su capa oculta, dropout y regularización, aprende mejor y reduciendo el error (BCE), mejorando la precisión general (accuracy), y logrando un mejor equilibrio entre precisión y F1-score. El Modelo B, aunque es simple y rápido, tiene menos capacidad y por tanto peores métricas ya que funciona como un modelo lineal.

Modelos perceptrón múltiple:
Modelo C:

- 2 capas ocultas de 64 neuronas
- Dropout y regularización
- Métricas: **BCE**-> 0,1694; **ACCURACY** -> 0,9504; **F1- SCORE**: 0,9482

Modelo D:

- 2 capas ocultas de 64 neuronas
- NO dropout y sí regularización
- Métricas: **BCE**-> 0,1654; **ACCURACY** -> 0,9500; **F1- SCORE**: 0,9479

Modelo E:

- 2 capas ocultas de 256 neuronas
- Dropout y regularización
- Métricas: **BCE**-> 0,2815; **ACCURACY** -> 0,9546; **F1- SCORE**: 0,9529

Modelo C y modelo D para ver los efectos de las técnicas avanzadas del Deep learning:Las métricas son prácticamente iguales, con una ligerísima mejora en BCE en el modelo D. Esto indica que en este caso, el modelo no estaba sobreajustando, por lo tanto, el dropout no aporta mejora clara. Aún así modelo C tendrá menos riesgo de overfitting.
El modelo E: Se entrenan más neuronas, por lo que, aumenta un poco la accuracy y F1-score, lo que indica que el modelo aprende patrones más complejos. Sin embargo, el coste aumenta.

Decisión final:
En general, más capas y neuronas mejoran el rendimiento**,** pero solo si están acompañadas de buenas prácticas de regularización**.** Sin embargo, esa potencia debe controlarse con técnicas de regularización y dropout para evitar sobreajuste.

El Modelo E es el mejor clasificador en términos de rendimiento (mayor F1, Accuracy)**.** Pero, tiene mayor error, lo que implica que tiene mayor coste computacional y riesgo de sobreajuste. Por tanto, hemos decidido elegir el modelo C como nuestro modelo final ya que tiene un accuracy menor pero muy parecido y un BCE menor notable.

3) Comparación del modelo C con scklearn

Como hemos explicado previamente el modelo C es una red neuronal con 2 capas ocultas, 64 neuronas, funciones ReLU y dropout para evitar que el modelo memorice datos en vez de aprender a generalizar.

En machine learning existen modelos listos para usar como librerías de sklearn que hacen la misma función, predecir si son fake o true news aprendiendo de los ejemplos. Pero se diferencian en que el modelo de sklearn (LogisticRegression) tiene menos control además de tener una personalización limitada ya que es un modelo preparado y con este no podríamos experimentar con diferentes capas o funciones para evaluar diferentes resultados.

4) CONCLUSIÓN

En este proyecto hemos demostrado cómo una red neuronal, partiendo de datasets y aplicando técnicas como Bag of Words, regularización, dropout y funciones de activación adecuadas, se consigue clasificar eficazmente noticias verdaderas y falsas a partir de su título.

Podemos concluir que, según los modelos estudiados, en general:

- Más capas y neuronas implican mejor modelo (Pero pueden incluir un mayor coste también)
- Es importante utilizar técnicas de regularización: Para evitar gradient vanisihing
- Es recomendable usar dropout para evitar overfitting. Aunque es verdad que en nuestros modelos no se ha mostrado una notable diferencia en usar dropout o no, esto se puede deber a que los datos introducidos ya estaban manipulados.

Tras comparar distintas arquitecturas, seleccionamos el **Modelo C por** ofrecer un equilibrio óptimo entre rendimiento y complejidad.

5-ESTADO DEL ARTE

Actualmente, la detección de noticias falsas resulta fundamental, dado el gran volumen de información que circula en redes sociales y medios digitales. Antes del auge de las nuevas tecnologías y el machine learning, la detección de fake news se realizaba principalmente mediante estrategias manuales de verificación conocidas como fact-checking. Estas eran realizadas por periodistas y profesionales especializados, que contrastaban datos con fuentes oficiales, analizaban el contexto y evaluaban la credibilidad de los emisores. Actualmente, esto se han ido complementado con la tecnología, con algoritmos de clasificación o redes neuronales, aunque en muchos casos el razonamiento humano sigue siendo clave, especialmente para interpretar ironía o sarcasmo.

Referencia: https://revistas.usfq.edu.ec/index.php/perdebate/article/view/1558/1706