

# REDES NEURONALES III

## Práctica VIII - Inteligencia Artificial Avanzada

Abián Santana Ledesma  
Samuel Frías Hernández  
Alexia Sánchez Cabrera  
Javier García Santana

## Introducción

En este trabajo se analizará un corpus de canciones tradicionales y modernas de Canarias aplicando distintos modelos de lenguaje neuronal. Para ello, se utilizarán cuatro modelos (PAD y RNN):

- **BengioNN**: Modelo neuronal feedforward que predice la siguiente palabra a partir de una ventana de contexto fija, utilizando embeddings y capas densas.
- **BengioNN Class**: Extiende BengioNN para realizar clasificación binaria de textos (e.g., estilo tradicional vs moderno), reemplazando la predicción de palabra por una salida de clase.
- **ElmanRNN**: Red neuronal recurrente simple que mantiene memoria del estado anterior, útil para modelar la secuencia y el orden temporal de las palabras.
- **ElmanRNN Class**: Modelo de red neuronal recurrente simple que emplea una arquitectura RNN con embeddings para capturar dependencias secuenciales en letras o versos, y una capa final para clasificar el estilo de la canción.

El objetivo es evaluar el desempeño de cada modelo en términos de generación de texto y capacidad de clasificación, prestando especial atención a la calidad lingüística y coherencia cultural de los resultados generados.

## Resultados

### PAD

#### BengioNN

Tradicional:

**Epoch [10], Loss: 0.2223, Perplexity: 1.2489, n=3**

**Texto Generado:** La veredita EOL Sale La Luna en el Sahara... EOL Va floreciendo la Libertad EOL Cuando Sulema riega el plantal EOL Va floreciendo la Libertad EOL Cuando Sulema amasa el pan EOL Va floreciendo la libertad EOL Cuando Sulema amasa el pan EOL Va floreciendo la Libertad EOL Cuando Sulema baila

**Epoch [15], Loss: 0.1954, Perplexity: 1.2159n=3**

**Texto Generado:** La veredita sonrío EOL Cuando tu pie la acaricia EOL Y la ventana se agita EOL Cuando por esa vereda EOL Tu fina estampa, pasea EOL Fina estampa, caballero EOL Quién te pudiera guardar EOL Fina estampa, caballero EOL Caballero de fina estampa EOL Un lucero EOL Que sonriera bajo un

En este caso vemos que subiendo el número de épocas, la pérdida disminuye junto con la perplejidad, lo que indica que el modelo está aprendiendo a generar texto de manera más coherente y cercana al estilo del corpus de entrenamiento. A medida que la red neuronal BengioNN se entrena por más épocas, mejora su capacidad de predecir la siguiente

palabra en una secuencia, lo que se refleja en una menor pérdida (loss) y una menor perplejidad, ambas métricas clave en tareas de modelado de lenguaje.

La calidad del texto generado también muestra una evolución: en la primera muestra, aunque se perciben repeticiones ("Cuando Sulema amasa el pan"), el estilo empieza a capturar cierta cadencia poética. Para la época 15, el modelo no solo reduce repeticiones, sino que también mantiene una estructura lírica más rica y coherente ("Fina estampa, caballero / Quién te pudiera guardar"), mostrando una mejor comprensión del patrón rítmico y temático del corpus.

Esto evidencia que un mayor entrenamiento permite que BengioNN no solo mejore cuantitativamente, sino también cualitativamente, acercándose a una generación de texto más natural y estilísticamente consistente. Sin embargo, también es importante vigilar el sobreajuste en épocas más altas, para asegurar que el modelo generalice bien y no se limite a memorizar las secuencias del conjunto de entrenamiento.

**Epoch [15], Loss: 0.0874, Perplexity: 1.0914, n=5**

**Texto Generado:** La feliz en su matrimonio EOL Aunque su marido era el mismo demonio EOL Tenía el hombre un poco de mal genio EOL Que ella se quejaba de que nunca fue tierno EOL EOL Desde hace ya más de tres años EOL Recibe cartas de un extraño EOL Cartas llenas de.

Al aumentar la ventana de contexto de 3 a 5, el modelo BengioNN mejora notablemente su rendimiento: la pérdida baja a 0.0874 y la perplejidad a 1.0914, lo que indica predicciones más precisas y seguras. Este mayor contexto permite al modelo captar mejor las dependencias lingüísticas y el estilo del texto, resultando en una generación más coherente y rica, como se refleja en versos con ritmo y sentido más claros. En resumen, el modelo aprovecha la información adicional para aprender patrones más complejos y producir salidas más naturales.

Nueva:

Con el corpus moderno, obtenemos unos resultados similares:

**Epoch [10], Loss: 0.2805, Perplexity: 1.3238, n=3**

**Texto Generado:** La letra completa estará disponible cuando salga la canción EOL EOL Y tú también, pero siempre lo vas a negar EOL Echo de menos las pelea', los grito' y los polvo' EOL No-No sé qué pasó con nosotros' EOL Pasamos del "te quiero" a la que no quería nada EOL EOL.

**Epoch [15], Loss: 0.2371, Perplexity: 1.2676, n=3**

**Texto Generado:** La letra completa estará disponible cuando salga la canción EOL EOL EOL Yo sé que te encanto cuando luego estás con él EOL Tú sabe' que el ego nos mata EOL Mai, te quiero ver EOL Ya me olvidé (Yeh-yeh) de quién es quién EOL Si estoy solo no sé, nadie.

**Epoch [15], Loss: 0.0809, Perplexity: 1.0842, n=5**

**Texto Generado:** La letra completa estará disponible cuando salga la canción EOL EOL EOL Yo sé que lo que hice estuvo mal EOL Pero sabes, baby, me lo vas a perdonar igual EOL No era ni el momento ni el lugar EOL No debimos discutir y nunca debí marchar EOL Ahora, ¿qué es.

A continuación estudiaremos la clasificación binaria de ambos corpus con BengioNN Class.

### **BengioNN Class**

La **precisión** se mantiene constante en **0.9987** en todas las combinaciones de hiperparámetros (tasa de aprendizaje, tamaño de lote y épocas). La **pérdida promedio** cambia ligeramente dependiendo de la configuración, pero la precisión no varía significativamente. A continuación se describen las principales variaciones:

Variaciones en la tasa de aprendizaje (LR):

- **LR=0.01:**
  - **Batch=64, Epochs=10:** Pérdida promedio = 0.0065
  - **Batch=64, Epochs=50:** Pérdida promedio = 0.0063
  - **Batch=128, Epochs=50:** Pérdida promedio = 0.0255 (Notablemente más alta)
- **LR=0.001:**
  - **Batch=64, Epochs=50:** Pérdida promedio = 0.0073
  - **Batch=128, Epochs=10:** Pérdida promedio = 0.0085
  - **Batch=128, Epochs=50:** Pérdida promedio = 0.0073

A medida que disminuye la tasa de aprendizaje (de 0.01 a 0.001), la pérdida generalmente es un poco más alta, pero no afecta significativamente la precisión final. Sin embargo, con una tasa de aprendizaje de **0.01** y **lotes grandes (Batch=128, Epochs=50)**, se observa un aumento en la pérdida (0.0255), lo que podría indicar algo de inestabilidad en esos parámetros.

## Variaciones en el tamaño del lote (Batch Size):

- **Batch=64 vs Batch=128:**

- **Con LR=0.01:** La pérdida para Batch=128 es consistentemente más alta (por ejemplo, 0.0075 vs 0.0063 para Epochs=10).
- **Con LR=0.001:** La pérdida es también más alta para Batch=128 (por ejemplo, 0.0085 vs 0.0073 para Epochs=10).

Un **tamaño de lote más grande (128)** tiende a dar como resultado una pérdida ligeramente más alta en comparación con un **tamaño de lote más pequeño (64)**, aunque el efecto no es extremo. Esto podría deberse a que el modelo no generaliza tan eficazmente con lotes más grandes.

## Variaciones en el número de épocas (Epochs):

- **Epochs=10, 20, 50:**

- A lo largo de las diferentes tasas de aprendizaje, tamaños de lote y valores de épocas, la **precisión de prueba** sigue siendo estable en **0.9987**.
- **Pérdida promedio:** Hay una ligera disminución de la pérdida promedio a medida que aumenta el número de épocas, especialmente para tasas de aprendizaje más pequeñas (por ejemplo, de 0.0065 → 0.0063 para LR=0.01, Batch=64, Epochs=10 a 50).

Aumentar el número de épocas no cambia drásticamente la precisión, pero provoca una ligera reducción en la pérdida, especialmente con tasas de aprendizaje más pequeñas. Sin embargo, la mejora en la reducción de la pérdida tiende a estabilizarse después de 20 o más épocas.

## RNN

### ElmanRNN

Tradicional:

**Epoch: 10, loss:1.3685 perplexity: 3.9293**

**Texto generado:** La Palma, Tanausú EOL Con un momento de la montaña EOL Que el campesino siembre su propio pan EOL El que me querías EOL Y el pescador pueda trabajar sus redes EOL EOL Un aire de mi tierra EOL De Gran Canaria, Doramas EOL Que se amotinen a mi cuerpo EOL

El modelo empieza a aprender estructuras y patrones del lenguaje, pero todavía muestra incertidumbre. La perplejidad de 3.92 indica que el modelo aún se debate entre varias posibles palabras al generar texto, aproximadamente unas 4. El contenido generado es coherente a nivel temático (referencias a La Palma, campesinos, pescadores), pero se nota cierta incoherencia entre las frases.

**Epoch:15 loss: 0.8153 perplexity: 2.599**

**Texto generado:** La guelder de una folía, dejarlo libre volar es la guerra EOL Y el blanco en la tierra que tanto EOL Y el camarero EOL A segar el mar jugaba el cantar es cosa buena EOL Ayúdame compañero EOL Porque el canto ya no suena EOL Corriendo al mar EOL ¡ay!

Aquí el modelo ha aprendido bastante más del corpus. Hay ya una métrica más baja de pérdida y la perplejidad indicando mayor certeza en las elecciones de palabras. Se puede apreciar que aparecen frases más líricas, aunque existen frases sin ningún sentido como "Y el camarero", lo que puede indicar una mezcla entre aprendizaje de estilo y comienzos de memorizar frases específicas del corpus.

**Epoch: 20 loss: 0.4563 perplexity:1.5782**

**Texto generado:** La Laguna donde hay un poema que, tembloroso, cantar es cosa buena EOL Ayúdame compañero que canta EOL Con un mensaje de alma EOL En la sombra perfecta; EOL Y limpieza en la noche también era un grillo, vidala y zamba... EOL Así perdida en los alambres EOL Y los ojos

Se puede ver una caída en la perplejidad y pérdida significa que el modelo "confía" mucho en sus predicciones. Aquí se puede detectar memorización parcial de estructuras concretas y frases típicas del corpus, lo que hace que el texto pierda el aprendizaje del estilo mismo.

### Conclusión:

Tras probar varias instancias con diferentes épocas, vemos que la perplejidad va bajando con respecto a la pérdida, sin embargo, mientras esto la pérdida va disminuyendo la RNN tiende a "aprenderse de memoria" las frases, es por esto que tiene menor perplejidad ya que esto indica la cantidad de palabras que duda para predecir la siguiente palabra.

Nueva:

**Epoch: 20 loss:2.1891 perplexity: 8.9273**

**Texto generado:** La letra completa estará disponible cuando me la enganché EOL Y si te lo que me llegue la mirada EOL EOL No me falta algo y me hace poco EOL EOL Y ahora la última vez que se siente por la mitad EOL Y yo me voy con cualquiera EOL No

Los resultados de la predicción son mucho más ruidosos que el anterior *epoch 20* de la generación anterior. Aquí la pérdida es mayor (2.18 vs. 0.45), y la perplejidad también se dispara (8.92 vs. 1.57), lo que sugiere que el modelo aún tiene dificultades para hacer predicciones.

**Epoch: 30 loss:1.4487 perplexity: 4.2474**

**Texto generado:** La letra completa estará EOL EOL No pu'es decir que yo no te lo niego EOL Y aunque despierto de los raperos a mí EOL EOL Mami, no les mientas, solo cuando estamo' a oscuras EOL EOL No pu'es decir que te encanto cuando luego estás con él EOL EOL Pero.

Aparecen frases con cadencia oral ("no pu'es decir", "estamo'"), lo que apunta a que está aprendiendo de forma correcta el lenguaje coloquial caracterizado por las canciones modernas. Aun así, todavía se nota cierta aleatoriedad. La coherencia mejora frente a epoch 20, pero aún no alcanza la calidad lírica del modelo entrenado con las canciones tradicionales. Mirando la perplejidad se puede llegar a la conclusión de que hay un aprendizaje en curso.

**Epoch: 50 loss:0.6249 perplexity: 1.8681**

**Texto generado:** La letra completa estará EOL EOL No te culpe', papi también era así EOL No le hagas caso a ti EOL Las noches que te fuiste y no se me ponga enfrente de las de Tama EOL Y en la mano para el examen EOL No sé qué pasó con otra

Este modelo tiene más coherencia que el anterior. Dicha inteligencia artificial tiene pocas dudas acerca de la posible siguiente palabra. Sin embargo, hay una mezcla extraña entre frases de la vida cotidiana y frases sin contexto claro. Es posible que haya empezado a memorizar algunas estructuras y por eso tenga esta confianza

### **conclusión:**

Se puede observar que en general todos los textos comienzan con una frase "La letra completa estará" por lo que seguramente en el corpus se haya descargado esta frase en repetidas ocasiones. Además a diferencia del modelo tradicional se puede observar que el lenguaje es claramente más coloquial. Aunque seguramente debido a que el corpus es más extenso, otra forma de abordar esta problemática es a raíz del propio lenguaje ya que el lenguaje coloquial es más complicado de predecir debido a su mezcla de informalismos y abreviaturas.

## ElmanRNN class

- **Épocas cortas (10 épocas):**

- Con learning rate = 0.0001, se observa un promedio de pérdida de 0.0710 y una precisión de prueba del 99.38%.
- Con learning rate = 0.001, el promedio de pérdida es 0.0691 y la precisión de prueba es del 98.55%.

Estos resultados indican que incluso con un número relativamente bajo de épocas, el modelo ya alcanza muy buenas métricas de desempeño, lo que sugiere que la convergencia se da de forma rápida, probablemente debido a la simplicidad relativa del dataset o a la capacidad del modelo para aprender patrones distintivos desde etapas tempranas.

- **Épocas intermedias (20 épocas):**

- Con learning rate = 0.0001, el promedio de pérdida aumenta ligeramente a 0.0990 y la precisión decrece a 98.13%.
- Con learning rate = 0.001, se consigue un promedio de pérdida menor (0.0520) y una precisión de 98.96%.

En estos experimentos se observa cierta variabilidad: en el caso de la tasa baja, el desempeño parece degradarse ligeramente con un incremento en el número de épocas, lo cual podría deberse a efectos de sobreajuste o a una variabilidad inherente en el entrenamiento (especialmente si el dataset es pequeño). En el caso de una tasa de aprendizaje más alta, la pérdida disminuye, sugiriendo que el modelo podría estar refinando mejor los parámetros; sin embargo, la precisión se mantiene en un rango muy similar.

- **Épocas largas (50 épocas):**

- Con learning rate = 0.0001, se registra un promedio de pérdida de 0.0538 y una precisión de 99.17%.
- Con learning rate = 0.001, a pesar de la etiqueta confusa en el mensaje (parece haber un error en el enunciado del experimento, ya que se menciona "Epochs = 20" pero se listan 50 épocas), se obtiene un promedio de pérdida de 0.0412 y una precisión de 98.55%.

Con 50 épocas se observa que la pérdida final baja significativamente en ambos casos, en particular con la tasa de 0.001 se alcanza una pérdida aún menor. La precisión se mantiene muy alta en ambos escenarios, lo que confirma la capacidad del modelo para generalizar correctamente, aunque en un caso se registra una pequeña disminución en la precisión pese a la baja pérdida.



- **Learning Rate = 0.0001:**

- Los resultados muestran alta precisión en las pruebas (entre 98.13% y 99.38%), siendo en el caso de 10 y 50 épocas las ejecuciones con mejores resultados de precisión (más de 99%).
- La pérdida varía en un rango entre 0.0538 y 0.0990, lo que indica un entrenamiento relativamente estable.

- **Learning Rate = 0.001:**

- Con una tasa mayor, el comportamiento es similar en cuanto a precisión (entre 98.55% y 98.96%), y en algunos momentos la pérdida es incluso menor (por ejemplo, 0.0412 en 50 épocas).
- Sin embargo, la diferencia en precisión respecto a una tasa de 0.0001 es mínima; en algunos casos, a pesar de tener una pérdida ligeramente inferior, la precisión se mantiene en valores muy similares o un poco inferiores (98.55% vs. 99.17% en el caso de 50 épocas).

Esto sugiere que el modelo es poco sensible a la variación en la tasa de aprendizaje entre estos dos valores, siempre que se mantengan dentro de un rango adecuado para el problema. En ambos casos, el modelo aprende a clasificar las muestras de forma casi perfecta.

## **Conclusión**

El análisis muestra que el modelo ElmanRNN es robusto y sensible a configuraciones de hiperparámetros en el rango evaluado. Tanto con una tasa de aprendizaje baja (0.0001) como con una tasa ligeramente mayor (0.001) se obtiene una alta precisión en el conjunto de prueba.

- Con 10 y 50 épocas se lograron los mejores resultados de precisión (cerca de o superiores al 99%), mientras que el caso de 20 épocas con LR = 0.0001 muestra una ligera disminución.
- La diferencia en pérdida entre configuraciones sugiere que, a pesar de que la función de pérdida se minimiza mejor con más épocas, la diferencia en precisión es marginal, lo que implica que el modelo ya había aprendido suficientemente bien para el problema con un número de épocas más reducido.

En definitiva, se recomienda ajustar estos hiperparámetros según las restricciones de tiempo computacional y la posible variabilidad del dataset en aplicaciones reales.

## Participación

- **Javier García Santana:** Testeo de modelos BengioNN y BengioNN class para su posterior evaluación de resultados. (33%)
- **Samuel Frías Hernández:** Testeo de modelo Elmann RNN class para su posterior evaluación de resultados. (33%)
- **Abián Santana Ledesma:** Testeo de modelo Elmann RNN para su posterior evaluación de resultados. (33%)