

# Reporte Ejercicio 1

## Modelos utilizados

---

1. **Modelo 1:**

1 capa oculta de 16 neuronas, con función de activación **ReLU** y regularización **L2**. Salida compuesta por una capa de 1 neurona con activación **ReLU** y regularización **L2**.

2. **Modelo 2:**

1 capa oculta de 32 neuronas, con función de activación **ReLU** y regularización **L2**. Salida compuesta por una capa de 1 neurona con activación **ReLU** y regularización **L2**.

3. **Modelo 3:**

1 capa oculta de 16 neuronas, con función de activación **ReLU** y regularización **L2**. Una capa de Dropout, una capa oculta de 8 neuronas con activación **ReLU** y regularización **L2**. Luego otra capa de Dropout y finalmente la capa de salida compuesta por una capa de 1 neurona con activación **ReLU** y regularización **L2**.

4. **Modelo 4:**

1 capa oculta de 64 neuronas, con función de activación **ReLU**. Salida compuesta por una capa de 1 neurona con activación **ReLU**.

5. **Modelo 5:**

1 capa oculta de 16 neuronas, con función de activación **ReLU** y regularización **L2**. Una capa oculta de 16 neuronas con activación **ReLU** y regularización **L2**. Finalmente la capa de salida compuesta por una capa de 1 neurona con activación **ReLU** y regularización **L2**.

## Procesado del Dataset mediante una representación TfIDf:

---

Parámetros del TfIDf:

1. **Binary:** True
2. **Ngram\_range:** (1,2)
3. **Stop\_words:** English
4. **Max\_df:** 0.7
5. **Norm:** L2
6. **Vocabulary:** None

## Decisiones de los Modelos

---

1. **Modelo 1:** Se busco generar un modelo simple, con poco procesamiento para poder tomarlo como

referencia para los otros dos modelos (Baseline).

- **Cantidad de capas:** 1 de entrada, 1 oculta y 1 de salida.
- **Tamaño de las capas:** 416701, 16 y 1.
- **Activación:** ReLU
- **Regularización:** L2.
- **Dropout:** No.
- **Función de costo:** binary\_crossentropy.
- **Optimizador:** Adam.
- **Metrica:** Accuracy.

2. **Modelo 2:** Se mantuvo la cantidad de capas, sin embargo se aumentó la cantidad de neuronas (Complejidad Media).

- **Cantidad de capas:** 1 de entrada, 1 oculta y 1 de salida.
- **Tamaño de las capas:** 416701, 32 y 1.
- **Activación:** ReLU.
- **Regularización:** L2.
- **Dropout:** No.
- **Función de costo:** binary\_crossentropy.
- **Optimizador:** Adam.
- **Metrica:** Accuracy.

3. **Modelo 3:** El modelo más complejo, se decidió utilizar una mayor cantidad de capas ocultas, junto con capas de Dropout.

- **Cantidad de capas:** 1 de entrada, 4 ocultas y 1 de salida.
- **Tamaño de las capas:** 416701, 16, dropout=0.5, 8, dropout=0.5 y 1.
- **Activación:** ReLU.
- **Regularización:** L2.
- **Dropout:** Si, a la salida de la capa de 32 neuronas y a la salida de la capa de 16 neuronas.
- **Función de costo:** binary\_crossentropy.
- **Optimizador:** Adam.
- **Metrica:** Accuracy.

4. **Modelo 4:** El modelo con más neuronas y sin regularización.

- **Cantidad de capas:** 1 de entrada, 1 oculta y 1 de salida.
- **Tamaño de las capas:** 416701, 64 y 1.
- **Activación:** ReLU.
- **Regularización:** None.
- **Dropout:** No.
- **Función de costo:** binary\_crossentropy.
- **Optimizador:** Adam.
- **Metrica:** Accuracy.

5. **Modelo 5:** El modelo con 2 capas ocultas iguales y regularización.

- **Cantidad de capas:** 1 de entrada, 2 ocultas y 1 de salida.
- **Tamaño de las capas:** 416701, 16, 16 y 1.
- **Activación:** ReLU.
- **Regularización:** L2.
- **Dropout:** No.

- **Función de costo:** binary\_crossentropy.
- **Optimizador:** Adam.
- **Métrica:** Accuracy.

## Proceso de Entrenamiento

---

1. **División de Train y Test:** 75% para Entrenamiento y 25% para Test.
2. **Tamaño de Batch:** 50.
3. **Número de Épocas:** 50.
4. **Métricas de evaluación:** Accuracy.

## Overfitting

---

Para evitar el **overfitting** se utilizó la técnica de **EarlyStopping** de Keras, la cual detiene el proceso de entrenamiento cuando una cantidad monitoreada ha dejado de mejorar, lo que evita que el modelo se **memorice** los datos de entrenamiento.

Descripción de los parámetros del **EarlyStopping**:

1. **Monitor:** Cantidad a monitorear.
2. **Min\_delta:** Cambio mínimo en la cantidad supervisada para calificar como una mejora, es decir, un cambio absoluto de menos de min\_delta, contará como ninguna mejora.
3. **Patience:** Número de épocas sin mejoría después del cual se detendrá el entrenamiento.
4. **Verbose:** Modo de verbosidad.
5. **Mode:** Uno de {auto, min, max}. En el modo mínimo, el entrenamiento se detendrá cuando la cantidad monitoreada haya dejado de disminuir; en modo máximo se detendrá cuando la cantidad monitoreada haya dejado de aumentar; en el modo automático, la dirección se deduce automáticamente del nombre de la cantidad supervisada.
6. **Baseline:** Valor de referencia para alcanzar la cantidad monitoreada. El entrenamiento se detendrá si el modelo no muestra mejoras sobre la línea de base.

Parámetros de **EarlyStopping** usados:

1. **Monitor:** val\_loss
2. **Min\_delta:** 0
3. **Patience:** 15
4. **Verbose:** 1
5. **Mode:** auto
6. **Baseline:** None

Además, se realizaron dos gráficas: **Accuracy vs Epoch** y **Loss vs. Epoch**, para poder observar de mejor

manera la evolución del aprendizaje de los modelos propuestos.

## Archivos Generados

---

1. **Modelos:** Figuras de los modelos y los modelos en sí (Carpeta **Modelos**).
2. **Precisión y Otros:** Reporte de los parámetros principales usados y precisiones logradas (Carpeta **Precision**).
3. **Resultados:** Predicciones realizadas y gráficas de **Accuray vs. Epoch** y **Loss vs. Epoch** (Carpeta **Resultados**)