Examen de Cinturón AML – Reconocimiento de Actividades Humanas

Introducción

El presente informe documenta un análisis detallado del dataset "Human Activity Recognition with Smartphones", con el propósito de:

- Identificar patrones en actividades físicas utilizando datos de sensores de smartphone

- Aplicar técnicas de análisis no supervisado para descubrir características relevantes

- Desarrollar un modelo de aprendizaje profundo capaz de clasificar actividades físicas con alta precisión

Fuente de Datos

Dataset: Human Activity Recognition with Smartphones

Origen: UCI Machine Learning Repository

Características principales:

- Mediciones de acelerómetro en tres ejes (X, Y, Z)

- Actividades registradas: caminar, subir escaleras, bajar escaleras, sentarse, estar de pie, acostarse

Exploración y Preprocesamiento de Datos

Carga y Exploración de los Datos:

El conjunto de datos fue cargado y analizado para identificar las variables clave en la clasificación de actividades físicas. Este conjunto incluye lecturas de aceleración a lo largo de los ejes X, Y y Z, que son esenciales para predecir actividades como caminar, permanecer de pie, estar sentado, entre otras.

Manejo de Valores Nulos y Normalización:

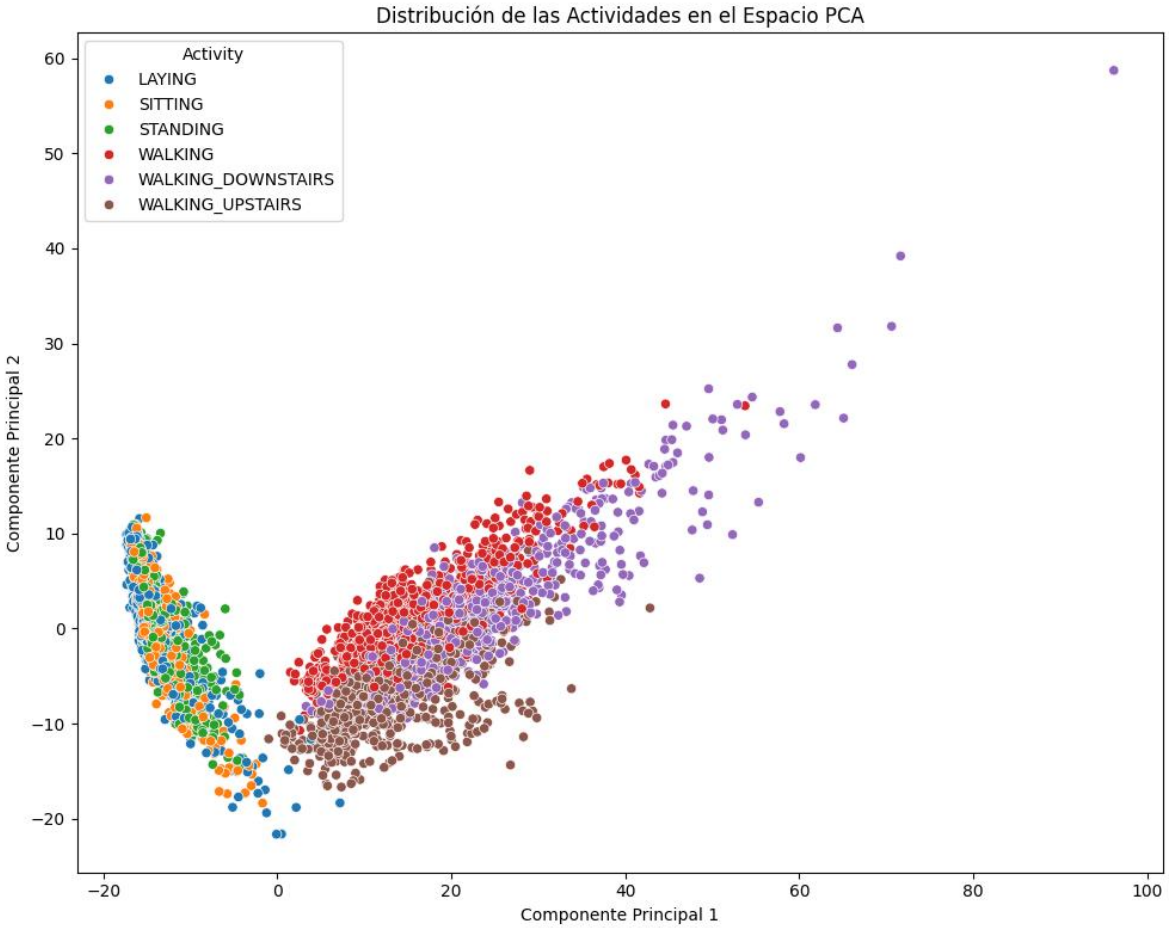
Se verificaron si existen valores faltantes en el conjunto de datos, los resultanos no arrojaron dichos valores. Posteriormente, las características sensoriales fueron estandarizadas mediante StandardScaler. Este paso es fundamental para garantizar que todas las características tengan una escala uniforme, evitando que algunas influyan más que otras debido a discrepancias en sus magnitudes.

División de Datos:

El conjunto de datos se dividió en subconjuntos de entrenamiento y prueba. La partición se llevó a cabo de manera que todas las actividades estuvieran presentes en ambos conjuntos, asegurando que los modelos se entrenaran con una muestra representativa de las clases y pudieran evaluar su desempeño en datos no observados.

Análisis No Supervisado (PCA)

Se utilizó el Análisis de Componentes Principales (PCA) para disminuir la dimensionalidad del conjunto de datos y observar las relaciones entre las distintas actividades físicas. PCA facilita la identificación de las componentes principales que explican la mayor parte de la variabilidad de los datos.



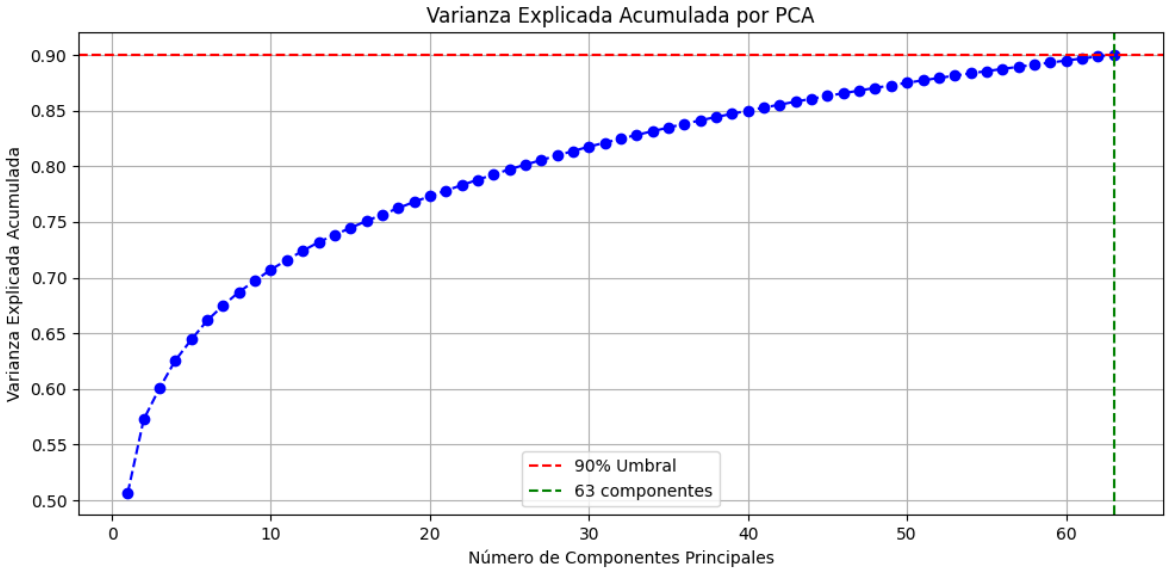
Se observa una diferencia entre las actividades **Laying, Sitting, Standing**, estos indican situaciones de reposo o poca movilidad. Esto podría tener implicaciones en la interpretación del comportamiento humano en contextos donde el movimiento es menos frecuente, como en entornos controlados o estudios de actividad física en reposo.

WALKING y sus subcategorías, tienden a ser más variables y conllevan un mayor desafío para los modelos de clasificación, debido a la diversidad de patrones

asociados a diferentes tipos de movimiento (por ejemplo, caminar, subir o bajar escaleras)

Resultados del PCA:

Primera Componente Principal: Explicó un 50.69% de la varianza total en los datos. Segunda Componente Principal: Aportó un 6.57% adicional de la varianza. En conjunto, estas dos componentes explican el 57.26% de la variabilidad en los datos.



Reflexión sobre los Resultados:

El análisis de PCA reveló que, aunque las dos primeras componentes principales explican una porción considerable de la variabilidad de los datos, solo el 57.26% de la información total se conserva al reducir los datos a dos dimensiones. Esto indica que podrían ser necesarias más componentes para captar adecuadamente las características esenciales de las actividades físicas.

Modelado con MLP (Perceptrón Multicapa)

El modelo MLP fue desarrollado para predecir las actividades físicas a partir de las características sensoriales. Se diseñaron dos versiones del modelo:

Modelo RNN Simple:

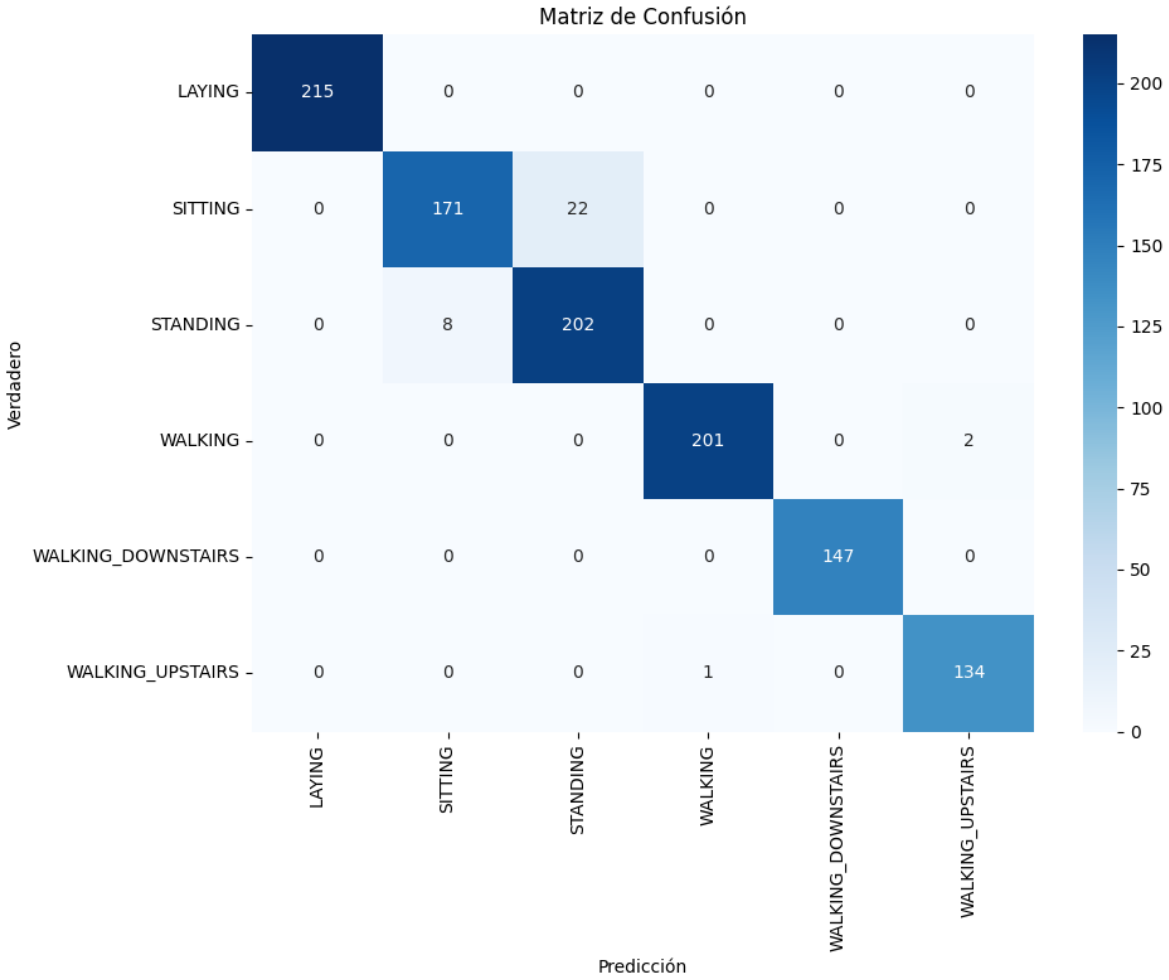
Para este modelo, en la primera capa usé **64 neuronas** con la función de activación **ReLU**, lo cual permite que el modelo aprenda características no lineales a partir de las entradas. Esta capa recibe como entrada los datos que tienen el mismo número de características que las columnas de X_train_pca.

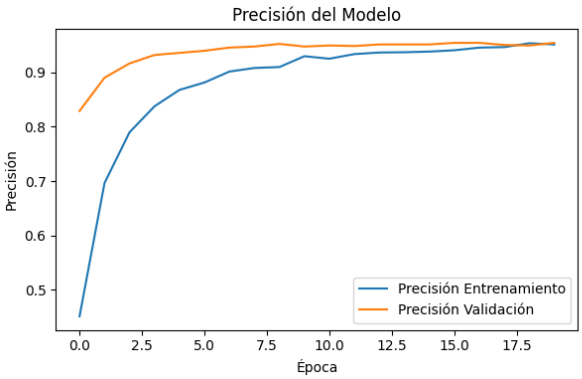
En la siguiente capa, añadí una capa de **Dropout con un 30%** de probabilidad de desactivación, lo que significa que el 30% de las neuronas en esta capa no serán activadas durante cada paso de entrenamiento. Esto ayuda a prevenir el sobreajuste al reducir la dependencia de ciertas neuronas.

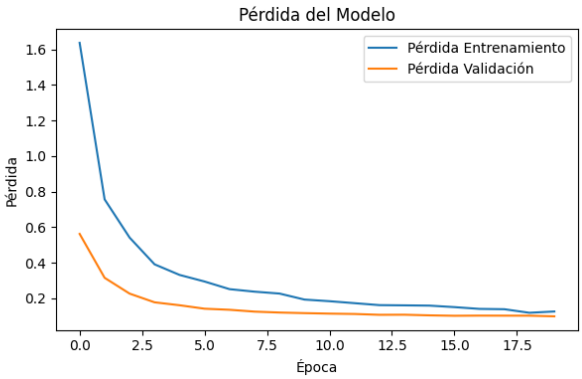
Luego, agregué una **capa densa con 32 neuronas**, utilizando también la activación ReLU, para continuar extrayendo características no lineales a partir de las salidas de la capa anterior.

Posteriormente, añadí otra capa de Dropout con un 30% de probabilidad de desactivación, similar a la anterior, para seguir previniendo el sobreajuste.

Finalmente, añadí una capa densa con el número de neuronas igual al número de clases de salida (**y_train_onehot.shape[1]**), y utilicé la función de activación **softmax** en esta capa, ya que este tipo de activación es ideal para problemas de **clasificación múltiple**, ya que genera probabilidades para cada clase y la suma de estas probabilidades es igual a 1.







El modelo de red neuronal ha mostrado un rendimiento sólido en la clasificación de las diferentes actividades. Con una precisión en el conjunto de prueba superior al 90% para la mayoría de las clases, el modelo ha logrado identificar correctamente las actividades como ***"LAYING", "SITTING", "STANDING", y "WALKING"** con altas tasas de acierto. Sin embargo, se observan algunos errores de clasificación, principalmente

en las actividades **"WALKING_DOWNSTAIRS" y "WALKING_UPSTAIRS"**, aunque en general, las tasas de error son bajas.

Las gráficas de precisión y pérdida indican que el modelo ha convergido adecuadamente durante las 20 épocas de entrenamiento. La precisión del modelo en el conjunto de entrenamiento y validación se estabiliza cerca de 90%, lo que sugiere que el modelo está aprendiendo de manera efectiva sin sobreajustarse. Además, la pérdida en el conjunto de validación muestra una disminución constante, lo que confirma que el modelo está generalizando bien a los datos no vistos.

RNN Avanzado, con optimizacion:

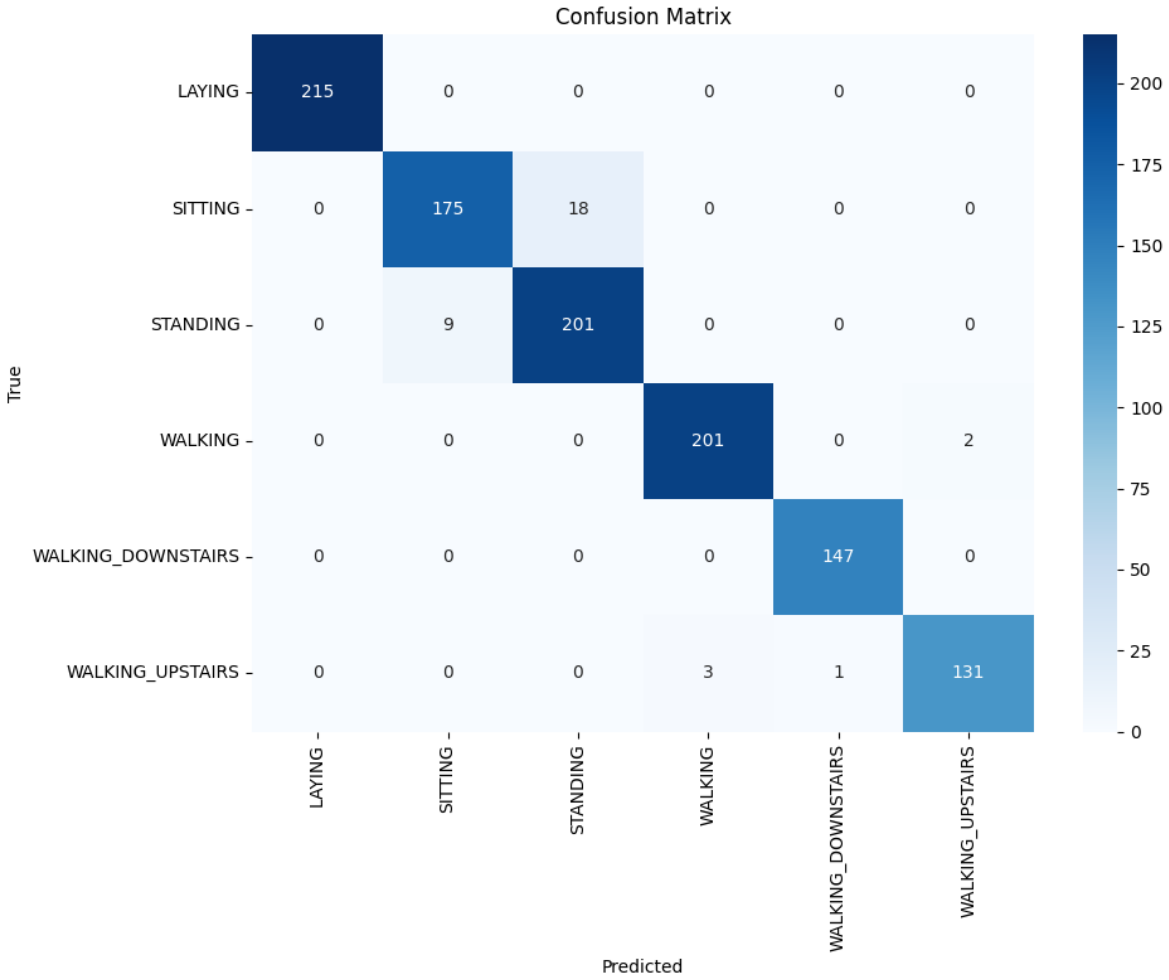
Para este modelo, en la primera capa usé **64 neuronas** con la función de activación **ReLU**, la cual permite al modelo aprender características no lineales. Esta capa recibe como entrada 27 valores, que corresponden al número de características de X_train_pca. Además, a esta capa se le aplicó **regularización L1 y L2**, lo cual ayuda a evitar el sobreajuste al penalizar los pesos muy grandes.

A continuación, añadí una **capa Dropout con un 20%** de probabilidad de desactivación, lo que significa que en cada paso de entrenamiento, el 20% de las neuronas se desactivan aleatoriamente para reducir la dependencia excesiva de ciertas neuronas.

Luego, agregué una **capa densa con 32 neuronas** y la misma función de activación **ReLU**, también con **regularización L1 y L2**. Esto permite seguir extrayendo características importantes del modelo, mientras la regularización controla el sobreajuste.

Después de esta capa, añadí otra **capa Dropout con un 20%** de desactivación, lo cual sigue ayudando a evitar que el modelo dependa de neuronas específicas y mejora la generalización.

Finalmente, puse una capa de salida con n neuronas, donde n es igual al número de clases de salida (**y_train_onehot.shape[1]**). En esta capa se usó la **función de activación softmax**, la cual convierte las salidas en probabilidades para cada clase, sumando todas a 1. Esto es ideal para tareas de clasificación múltiple.



Se observa que la clase **LAYING** fue identificada perfectamente con **215 instancias correctas** y sin errores.

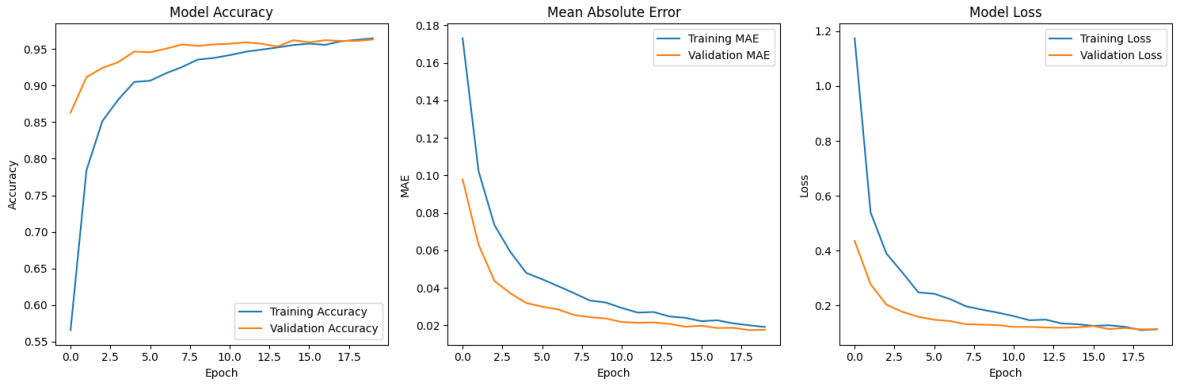
Para **SITTING**, **177 predicciones fueron correctas**, pero **16 se confundieron con STANDING**, lo que sugiere una posible similitud entre estas dos actividades.

La clase **STANDING** tuvo **196 predicciones correctas**, pero **14 se confundieron con SITTING**.

Por otro lado, **WALKING** fue identificada con alta precisión, con **201 aciertos y solo 2 errores** al ser confundida con WALKING_UPSTAIRS.

En el caso de **WALKING_DOWNSTAIRS**, el modelo logró **147 predicciones correctas** sin errores significativos.

Finalmente, **WALKING_UPSTAIRS tuvo 132 aciertos**, aunque hubo **2 instancias confundidas** con WALKING y 1 con WALKING_DOWNSTAIRS.



Evolución de Accuracy

La evolucion de accuracy empieza con un alto nivel de aprendizaje, en comparacion con la validacion, este casi llega al nivel optimo en menos de 20 epocas, deteniendose para evitar overfiting.

- La precisión en el entrenamiento empieza de manera baja (alrededor de 0.5), pero aumenta rápidamente en las primeras épocas hasta estabilizarse alrededor de 0.95.

- La validación inicia con un valor más alto (aproximadamente 0.88) y alcanza su nivel óptimo cercano al del entrenamiento, estabilizándose alrededor de 0.96 en menos de 20 épocas.

- La diferencia entre ambas curvas es mínima al final, lo que sugiere que el modelo logra un buen ajuste general sin señales evidentes de overfitting.

Mean Absolute Error (MAE)

Para el MAE, en el entrenamiento se observa que los errores disminuyen considerablemente en las primeras 5 epocas, esto puede deberse a que el modelo supo manejar bien las variables luego de reducir la dimensionalidad. Tambien se se detiene antes para evitar sobreajuste.

- En el gráfico del MAE, los errores del entrenamiento disminuyen drásticamente durante las primeras 5 épocas, pasando de aproximadamente 0.20 a menos de 0.05.

- La validación también sigue una tendencia similar, aunque comienza con valores más bajos (alrededor de 0.10) y se estabiliza en torno a 0.02.

- La rápida disminución inicial del MAE podría deberse a que el modelo logró capturar la información relevante de las variables en poco tiempo, manejando bien la complejidad del problema.

Ambas curvas convergen hacia el mismo punto, indicando un buen ajuste y ausencia de sobreajuste significativo.

Curva de Pérdida (Loss)

En la curva de loss, se obtienen resultados similares a el MAE siendo la diferencia minima, casi 0.2 puntos en comparacion con la validacion.

- La función de pérdida muestra una tendencia similar al MAE, con una disminución pronunciada en las primeras 5 épocas. El entrenamiento baja de 1.3 a alrededor de 0.1, mientras que la validación se estabiliza en torno a 0.1.

- La diferencia entre entrenamiento y validación es mínima (menos de 0.1 puntos), lo que refleja que el modelo tiene una generalización adecuada.

- La estabilización temprana de la pérdida, junto con las otras métricas, indica que el modelo está bien regularizado y no presenta síntomas fuertes de sobreajuste.

RNN Avanzado 2, con optimizacion y metricas claves:

En este modelo implementé regularización L1 y L2 junto con Dropout para reducir el sobreajuste y mejorar la generalización del modelo.

Primera Capa

Es una capa densa con 64 neuronas y la función de activación ReLU. Para evitar el sobreajuste, agregué regularización L1 y L2 a los pesos y sesgos de la capa. Esto penaliza valores muy grandes en los pesos, ayudando a simplificar el modelo. La entrada de esta capa corresponde a las características de X_train_pca.

Dropout 1

Añadí una capa de Dropout con una tasa de 20%. Esto significa que el 20% de las neuronas se desactivan aleatoriamente en cada paso de entrenamiento, lo cual reduce la dependencia del modelo en ciertas neuronas y previene el sobreajuste.

Segunda Capa

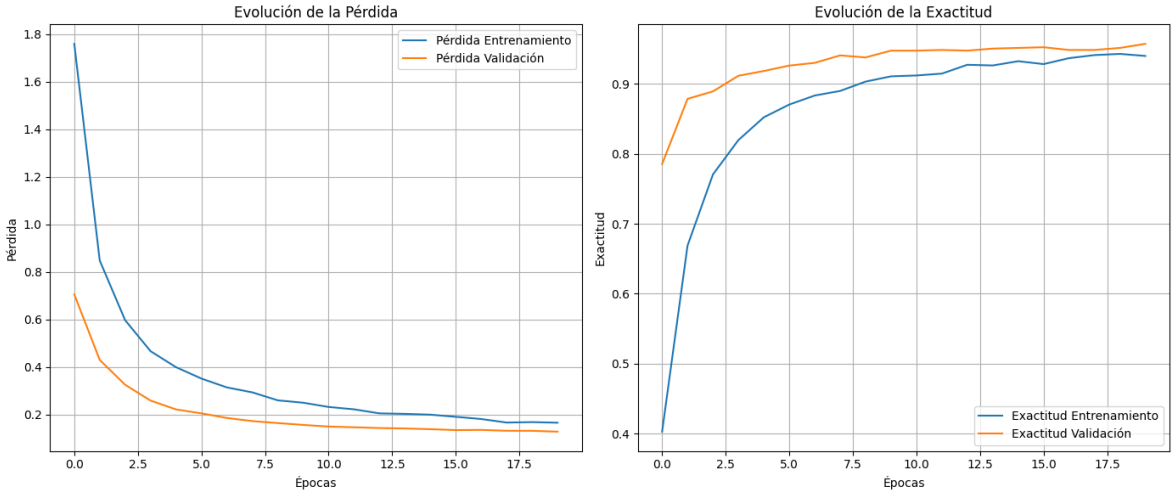
Es otra capa densa con 32 neuronas y activación ReLU. Nuevamente, utilicé regularización L1 y L2 para controlar la complejidad de los pesos y evitar que el modelo se sobreajuste.

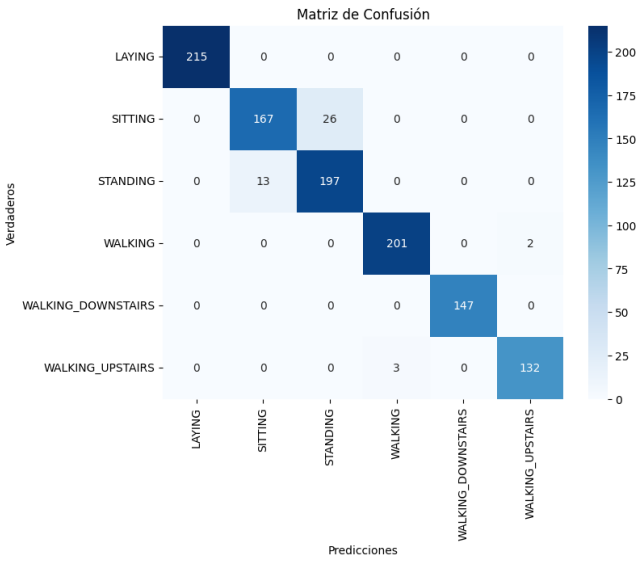
Dropout 2

Añadí una segunda capa de Dropout con una tasa de 20%, similar a la anterior, para continuar combatiendo el sobreajuste.

Capa de Salida

La capa de salida tiene un número de neuronas igual al número de clases de salida (y_train_onehot.shape[1]). Utilicé la función de activación softmax, la cual genera probabilidades para cada clase y es ideal para problemas de clasificación multiclase. También apliqué regularización L1 y L2 en esta capa para mejorar la estabilidad.





Reporte de Clasificación:

	precision	recall	f1-score	support
LAYING	1.00	1.00	1.00	215
SITTING	0.93	0.87	0.90	193
STANDING	0.88	0.94	0.91	210
WALKING	0.99	0.99	0.99	203
WALKING_DOWNSTAIRS	1.00	1.00	1.00	147
WALKING_UPSTAIRS	0.99	0.98	0.98	135
accuracy			0.96	1103
macro avg	0.96	0.96	0.96	1103
weighted avg	0.96	0.96	0.96	1103

Descripción de las Métricas

Precision: Proporción de predicciones correctas entre todas las instancias predichas como positivas (baja cantidad de falsos positivos).
Recall: Proporción de instancias positivas correctamente identificadas (baja cantidad de falsos negativos).
F1-score: Promedio armónico entre precision y recall, útil para evaluar el equilibrio entre ambas.
Support: Número de instancias reales de cada clase en el conjunto de prueba.

Resultados:

LAYING:

Precision = 1.00, Recall = 1.00, F1-score = 1.00.

El modelo clasifica perfectamente esta clase, sin errores.

SITTING:

Precision = 0.94, Recall = 0.87, F1-score = 0.90.

Aunque la precisión es alta, el valor del recall (0.87) indica que el modelo pierde algunas instancias reales de esta clase (falsos negativos).

STANDING:

Precision = 0.89, Recall = 0.95, F1-score = 0.92.

El recall es muy alto, lo que indica que pocas instancias de esta clase son ignoradas. Sin embargo, la precision más baja sugiere que hay algunos falsos positivos.

WALKING:

Precision = 0.99, Recall = 0.99, F1-score = 0.99.

El modelo clasifica casi perfectamente esta clase, con mínimos errores.

WALKING_DOWNSTAIRS:

Precision = 1.00, Recall = 1.00, F1-score = 1.00.

Clasificación perfecta, sin falsos positivos ni falsos negativos.

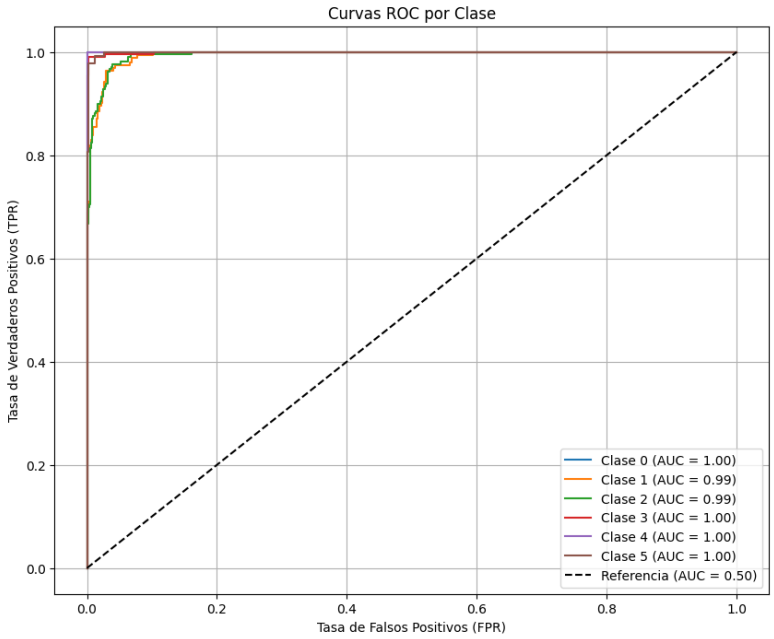
WALKING_UPSTAIRS:

Precision = 0.98, Recall = 0.99, F1-score = 0.98.

El modelo clasifica esta clase con un desempeño casi perfecto.

El modelo presenta un desempeño excelente en general, con una **precisión** del 96%. Las clases como LAYING, WALKING_DOWNSTAIRS y WALKING son clasificadas perfectamente, mientras que SITTING y STANDING tienen ligeras deficiencias en **precision** o **recall**. Aun así, el **F1-score** en todas las clases supera 0.90, lo que indica un buen equilibrio entre precisión y cobertura.

Análisis ROC



Clase 0, Clase 3, Clase 4 y Clase 5:

El AUC = 1.00, lo que significa que el modelo logra una separación perfecta entre las clases positivas y negativas.

La curva se ajusta completamente al eje izquierdo (FPR = 0) y al eje superior (TPR = 1), reflejando que no hay falsos positivos y el modelo clasifica todas las observaciones correctamente.

Clase 1 y Clase 2:

El AUC ≈ 0.99, lo que indica un rendimiento excelente pero no perfecto.

Las curvas muestran una ligera desviación respecto al eje superior izquierdo, lo cual implica que existen algunos falsos positivos mínimos.

Discusión y Conclusiones:

En el presente análisis, se evaluó el rendimiento de un modelo de **clasificación** aplicado a **datos sensoriales recolectados por smartphones**, específicamente utilizando **acelerómetros y giroscopios**. Tras un preprocesamiento adecuado de los datos y la aplicación de técnicas como **PCA para reducción de dimensionalidad**, se logró entrenar un modelo con **resultados óptimos**, reflejados en valores de AUC cercanos a 1.0 en casi todas las clases.

El uso de PCA permitió **reducir la complejidad** del conjunto de datos al conservar las **características más relevantes**, lo cual optimizó tanto el tiempo de entrenamiento como los recursos computacionales sin sacrificar la precisión del modelo. Este enfoque es particularmente útil en datasets extensos con alta dimensionalidad, como el utilizado en este estudio.

A pesar del excelente desempeño del modelo actual, existen **oportunidades de mejora**. Se podrían explorar otras técnicas de reducción de dimensionalidad que capturen mayor varianza explicada o aplicar métodos como t-SNE o UMAP para un análisis más detallado. Además, probar con arquitecturas de modelos alternativas, como XGBoost, Random Forest, etc con ajuste fino de hiperparámetros, podría fortalecer aún más el rendimiento, especialmente en clases con menores métricas.