Tarea 2: Clasificación de Imágenes y Estimación de Incertidumbre

Fernando Antonio Quiroz Escobar,
Astronomía, Universidad de Concepción. Matrícula 2020048037

1 Introducción

Como en la tarea uno, trabajaremos con datos obtenidos por High Cadence Transient Survey (HiTS), no obstante, este trabajo estará dirigido mayormente a la estimación de incertidumbre en las predicciones usando métodos como Monte Carlo Dropout, además de la calibración del modelo donde aplicamos regresión Sigmoide, donde en conjunto se puede evidenciar la clara mejora del modelo.

Monte Carlo Dropout es una técnica utilizada en redes neuronales para estimar la incertidumbre en las predicciones, implementando Dropout tanto en el entrenamiento como en la inferencia. Introducida por Yarin Gal y Zoubin Ghahramani [1], esta técnica consiste en realizar múltiples predicciones estocásticas del mismo input con Dropout activado durante la inferencia, y luego promediar estos resultados para obtener la predicción final. La variabilidad entre estas predicciones se utiliza para cuantificar la incertidumbre. Este método proporciona una aproximación práctica a la inferencia bayesiana, permitiendo modelar la incertidumbre sin necesidad de métodos bayesianos complejos, y mejora la generalización del modelo al mantener las propiedades de regularización de Dropout.

La calibración de Platt, también conocida como Platt Scaling, es una técnica introducida por John Platt en 1999 que se utiliza para convertir las puntuaciones sin calibrar de un clasificador binario en probabilidades calibradas. Este método ajusta una función sigmoide a las puntuaciones del modelo, transformándolas en probabilidades bien calibradas mediante una regresión logística. La fórmula utilizada es $P(y = 1|f) = \frac{1}{1+\exp(Af+B)}$, donde f es la puntuación sin calibrar y A y B son parámetros ajustados para minimizar la discrepancia entre las probabilidades predichas y las etiquetas verdaderas. La calibración de Platt mejora la precisión de las predicciones probabilísticas, haciéndolas más representativas de las probabilidades verdaderas, y es aplicable a diversos clasificadores, incluyendo SVMs y redes neuronales ([3]).

2 Materiales y Métodos

2.1 Carga y Preprocesamiento de Datos

Se utilizó un conjunto de datos que contiene imágenes y etiquetas, almacenado en un archivo PKL. Las imágenes fueron reestructuradas para tener dimensiones específicas (21x21 píxeles con una sola canal), y tanto las imágenes como las etiquetas fueron convertidas a tensores para su procesamiento con PyTorch. se ha de mecnionar que en este algoritmo solo se consideró usar la imágenes de ciencia, ya que el uso de múltiples tipos de imágenes concatenadas podría aumentar la complejidad del modelo y del proceso de entrenamiento sin agregar un beneficio significativo. Estos datos fueron divididos en tres conjuntos: entrenamiento con 2415 datos (60%), validación con 805 datos (20%) y prueba también con 805 datos (20%). Esta división permite entrenar el modelo y evaluar su rendimiento de manera objetiva.

2.2 Entrenamiento del Modelo

El entrenamiento del modelo se realiza utilizando la función de pérdida de entropía cruzada, adecuada para tareas de clasificación. El optimizador Adam ajusta los pesos del modelo durante el entrenamiento, conocido por su capacidad de manejar problemas de gradiente y ajustar la tasa de aprendizaje adaptativamente. Durante cada época de entrenamiento, se monitoriza el rendimiento en el conjunto de validación para ajustar hiperparámetros y evitar el sobreajuste. Las pérdidas y precisiones en los conjuntos de entrenamiento y validación se registran para evaluar la convergencia y desempeño del modelo .

Se consideraron varios hiperparámetros, incluyendo el "learning rate" (lr) establecido en 0.001, el tamaño del "batch" fijado en 32, con 50 épocas, y un filtro de dimensiones 2x2 en la capa de convolución. Además, se aplicó un "dropout" del 50%.

Hiperparámetros	Valor
Learning Rate (lr)	0.001
Tamaño del Batch	32
Épocas	50
Filtro	2x2
Capas	9
Dropout	50%

Table 1: Hiperparámetros con sus respectivos valores

2.3 Arquitectura

Capa	Tamaño de Entrada	Tamaño de Salida	Filtro	Función de Activación
Input	$B \times 1 \times 21 \times 21$	-	-	-
Conv1	$B \times 1 \times 21 \times 21$	$B \times 45 \times 20 \times 20$	2×2	ReLU
MaxPool1	$B \times 45 \times 20 \times 20$	$B \times 45 \times 19 \times 19$	2×2	-
Dropout	$B \times 45 \times 19 \times 19$	$B \times 45 \times 19 \times 19$	-	-
Conv2	$B \times 45 \times 19 \times 19$	$B \times 76 \times 18 \times 18$	2×2	ReLU
MaxPool2	$B \times 76 \times 18 \times 18$	$B \times 76 \times 17 \times 17$	2×2	-
Dropout	$B \times 76 \times 17 \times 17$	$B \times 76 \times 17 \times 17$	-	-
FC1	$B \times (17 \times 17 \times 76)$	$B \times 128$	-	ReLU
FC2	$B \times 128$	$B \times 2$	-	-

Table 2: Arquitectura de la Red Neuronal Convolucional

Capa de Convolución (Conv2d): En la arquitectura de la CNN, se utilizan capas de convolución (nn.Conv2d) para extraer características relevantes de las imágenes. En la primera capa de convolución (Conv1), se especifica un canal de entrada de tamaño 1 (escala de grises), 45 canales de salida y un kernel (filtro) de tamaño 2x2. Esto implica que la capa convolucional realiza 45 operaciones de convolución con un kernel de 2x2 en cada canal de la imagen de entrada. De manera similar, en la segunda capa de convolución (Conv2), se tienen 45 canales de entrada, 76 canales de salida y un kernel de tamaño 2x2.

Función de Activación ReLU: Después de cada capa de convolución, se aplica una función de activación ReLU. La función ReLU (Rectified Linear Unit) introduce no linealidades en la red, permitiendo que la CNN aprenda características no lineales en los datos. Esto es crucial para capturar la complejidad de las imágenes científicas, ya que muchas relaciones en los datos visuales no son lineales.

Capa de Max Pooling (MaxPool2d): Después de aplicar la función ReLU, se utiliza una capa de max pooling (nn.MaxPool2d) con un kernel de tamaño 2x2. El max pooling reduce la dimensionalidad de la salida de las capas de convolución, conservando las características más importantes y reduciendo el número de parámetros en la red. Esto ayuda a evitar el sobreajuste y mejora la eficiencia computacional durante el entrenamiento. En la primera capa de max pooling (MaxPool1), el tamaño de salida se reduce de $B \times 45 \times 20 \times 20$ a $B \times 45 \times 19 \times 19$. De manera similar, en la segunda capa de max pooling (MaxPool2), el tamaño de salida se reduce de $B \times 76 \times 18 \times 18$ a $B \times 76 \times 17 \times 17$.

Capa de Dropout: Para regularizar la red y prevenir el sobreajuste, se aplica una capa de dropout (nn.Dropout) después de cada capa de max pooling. El dropout apaga aleatoriamente unidades (neuronas) durante el entrenamiento, lo que obliga a la red a aprender características más robustas y generalizables.

Capas Completamente Conectadas (Linear): Finalmente, después de aplanar la salida de la última capa de dropout, se utilizan dos capas completamente conectadas (nn.Linear) para la clasificación binaria. La primera capa completamente conectada (FC1) tiene $17 \times 17 \times 76$ neuronas de entrada y 128 neuronas de salida, mientras que la segunda capa completamente conectada (FC2) tiene 128 neuronas de entrada y 2 neuronas de salida, que representan las clases binarias a predecir.

2.4 Evaluación del Modelo y Curvas de Aprendizaje

Al final del entrenamiento, se graficaron las curvas de aprendizaje para visualizar la evolución de la pérdida y precisión a lo largo de las épocas. Esto permite identificar si el modelo está aprendiendo correctamente y si hay problemas de sobreajuste o subajuste. Las curvas de pérdida muestran cómo disminuye el error del modelo, mientras que las curvas de precisión indican la proporción de predicciones correctas en cada época.

2.5 Estimación de Incertidumbre

El objetivo de esta sección es identificar y evaluar cuando el modelo clasifica incorrectamente y detectar si presenta alta incertidumbre en sus predicciones.

Para lograr esto, utilizaremos un modelo con técnicas de estimación de incertidumbre. Algunas de las técnicas más comunes son el Autoencoder Variacional, Monte Carlo Dropout y redes con enfoque Bayesiano. Para este informe, hemos seleccionado Monte Carlo Dropout.

El modelo seleccionado es un clasificador multiclase que, para evaluar la incertidumbre asociada a la predicción, se transforma en una tarea binaria durante la inferencia de los datos de test. La nueva etiqueta

$$\tilde{e} = \begin{cases} 1 & si \quad y_i \neq \hat{y}_i \\ 0 & si \quad y_i = \hat{y}_i \end{cases} \tag{1}$$

Donde y_i es la etiqueta verdadera y es la etiqueta predicha. Estas nuevas instancias indican si el modelo cometió un error al predecir la etiqueta.

Como bien se mencionó en la introducción, durante la inferencia, se utiliza Monte Carlo Dropout para obtener múltiples predicciones para cada muestra de prueba. Esto se logra habilitando dropout incluso en modo de evaluación. Se realizan múltiples pases hacia adelante (T veces) a través del modelo con dropout activado, obteniendo diferentes vectores de probabilidad para cada clase. A partir de estas predicciones, se calcula la media y la desviación estándar de las probabilidades predichas. El grado de confianza u se define como

$$u = 1 - \max\left[\frac{1}{T}\sum_{t=1}^{T} p_t(y = c \mid x)\right] \tag{2}$$

donde T es el número de pases hacia adelante, y p_t es la probabilidad predicha en el t-ésimo pase. Se comparan las predicciones promedio con las etiquetas verdaderas para determinar si la predicción fue correcta o incorrecta.

Es importante recatar que en el código se realizaron 100 pases hacia adelante (T = 100) para estimar la incertidumbre, siguiendo las recomendaciones de estudios previos que sugieren que un

número alto de pases mejora la estimación de la incertidumbre [1].

Para evaluar la capacidad del modelo para distinguir entre predicciones correctas e incorrectas, se calcula el área bajo la curva ROC (ROC-AUC). Se grafica la curva ROC para visualizar el rendimiento del modelo en términos de tasa de verdaderos positivos y tasa de falsos positivos. Además, se calculan métricas adicionales como precisión, recall, F1 Score y exactitud para evaluar el rendimiento del modelo de manera integral.

2.6 Calibración del Modelo

Antes de empezar se ha de mencionar que la calibración del modelo, y su método de calibración fue obtenido de [2] en el cual describen tanto como el método "Isotonic Regression" y "Platt Scaling", se probaron ambos, no obstante en este caso, "Platt scaling" tuvo mejores resultados en su gráfica ROC.

Primero, se extraen características del modelo de PyTorch utilizando la función "*extract_features*". El propósito de esta función es cambiar el modelo al modo de evaluación y desactivar el cálculo del gradiente para mejorar la eficiencia. Luego, se recorre el conjunto de datos (loader) y se obtienen las salidas de las capas convolucionales del modelo (model.convs). Estas salidas se remodelan en un vector plano y se agregan a una lista de características (features). Simultáneamente, se guardan las etiquetas correspondientes en una lista separada (labels). Al final, estas listas se concatenan para formar un tensor de características y otro de etiquetas.

Se prosigue con la conversión a Numpy de las características y etiquetas extraídas se convierten a arreglos de Numpy. Esta conversión es necesaria porque los clasificadores de Scikit-learn, como el que se utiliza posteriormente, trabajan mejor con datos en formato Numpy.

Luego viene el entrenamiento del clasificador base, en el cual se crea una instancia del clasificador de regresión logística (LogisticRegression) y se entrena (ajusta) con las características y etiquetas del conjunto de entrenamiento. Este clasificador servirá como modelo base antes de realizar la calibración.

Siguiendo va la calibración del clasificador, de forma que se utiliza CalibratedClassifierCV con el método de calibración **sigmoid** para calibrar las probabilidades predichas por el clasificador base. Este método ajusta las probabilidades predichas para que se correspondan mejor con las probabilidades verdaderas observadas en los datos de entrenamiento. La calibración es crucial para mejorar la confiabilidad de las predicciones de probabilidad del modelo. Aquí, cv='prefit' indica que el clasificador base ya está entrenado.

Luego de la calibración del clasificador, sigue la obtención de probabilidades de validación, se usan las características del conjunto de validación para obtener las probabilidades calibradas. La función *predict_proba* del clasificador calibrado genera las probabilidades predichas de la clase positiva (asumiendo un problema binario).

Se calcula el área bajo la curva ROC (ROC-AUC) usando *roc_auc_score*. Esta métrica evalúa la capacidad del modelo para discriminar entre clases. Un valor más alto de ROC-AUC indica un mejor desempeño del modelo.

Finalmente, se genera y muestra la curva ROC. Esta curva ilustra el desempeño del modelo en términos de la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR) a varios umbrales de clasificación. La línea diagonal representa un clasificador aleatorio. La distancia de la curva ROC a esta línea diagonal indica el poder discriminante del modelo.

La calibración del modelo se evalúa utilizando la puntuación de Brier y las curvas de calibración. La puntuación de Brier mide la precisión de las probabilidades predichas (siendo un valor cercano a 0 mejor que un valor cercano a 1), mientras que las curvas de calibración muestran la relación entre las probabilidades predichas y la proporción real de aciertos. Un modelo bien calibrado tendrá una curva de calibración cercana a la diagonal. Se utiliza la regresión sigmoide para mejorar la calibración del modelo. Se ajusta las probabilidades predichas de manera que se alineen mejor con las frecuencias observadas.

3 Resultados y Discusión

3.1 Curvas de Aprendizaje

Como podemos ver en 1 el gráfico de **Training and Validation Loss**, se observa cómo la pérdida disminuye rápidamente durante las primeras épocas tanto para los datos de entrenamiento como de validación, lo cual indica que el modelo está aprendiendo de manera efectiva. Sin embargo, mientras que la pérdida del entrenamiento sigue disminuyendo y se mantiene baja, la pérdida de validación presenta fluctuaciones ligeras después de estabilizarse. Estas fluctuaciones pueden sugerir cierto grado de sobreajuste, pero no parecen ser significativas, lo que sugiere que el modelo está manejando bien el aprendizaje sin ajustarse demasiado a los datos de entrenamiento.

En el gráfico de **Training and Validation Accuracy**, también en 1, la precisión aumenta rápidamente durante las primeras épocas y se estabiliza en un nivel alto cercano al 98% para el entrenamiento y al 96% para la validación. La pequeña diferencia entre estas dos curvas indica que el modelo está generalizando bien a los datos de validación y no está sobreajustado a los datos de entrenamiento. La precisión estable en la validación sugiere que el modelo puede ser robusto y confiable cuando se aplican nuevos datos.

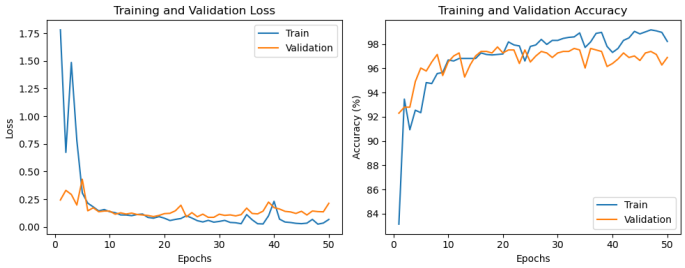

Figure 1: Este gráficos muestra la pérdida y la precisión durante el entrenamiento y la validación a lo largo de 50 épocas.

Métricas modelo sin calibrar	
Accuracy	0.9639751552795031
Precision	0.9642994111243072
Recall	0.9636970730321114
F1 Score	0.9639250531195673

Table 3: metricas obtenidas antes de calibrar

3.2 ROC-AUC sin calibrar

La ROC Curve (2) muestra la capacidad del modelo para distinguir entre las dos clases posibles. La curva ROC se

eleva significativamente por encima de la línea diagonal de no discriminación (AUC=0.5), lo que indica un buen desempeño. Con un área bajo la curva (ROC-AUC) de 0.8724, el modelo demuestra una alta capacidad de discriminación, siendo capaz de distinguir correctamente entre clases positivas y negativas la mayor parte del tiempo. Este valor de AUC es un fuerte indicador de la eficacia del modelo en términos de su capacidad para predecir correctamente la clase de las instancias.

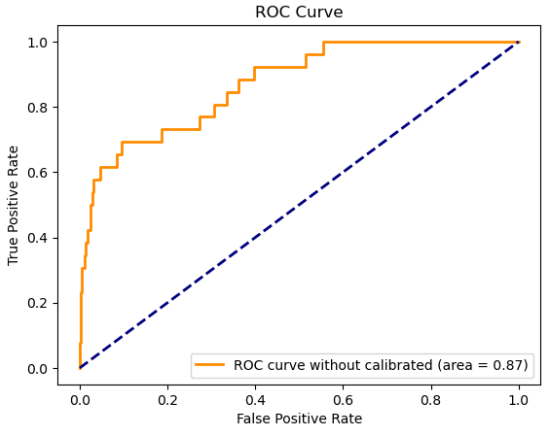

Figure 2: Curva ROC sin calibrar

3.3 ROC-AUC calibrado

El gráfico de la ROC Curve Calibrated (3) revela una mejora significativa en la capacidad de discriminación del modelo después de la calibración. La curva ROC se eleva casi verticalmente y se mantiene cerca del eje de las ordenadas antes de alcanzar el punto (1,1), lo que indica una excelente capacidad de discriminación. Con un área bajo la curva (ROC-AUC) de 0.99, el modelo calibrado muestra una capacidad casi perfecta para distinguir entre las clases positivas y negativas, mejorando notablemente respecto al modelo sin calibrar. Esto sugiere que la calibración ha mejorado la precisión de las predicciones probabilísticas del modelo, haciéndolo más robusto y confiable para la clasificación.

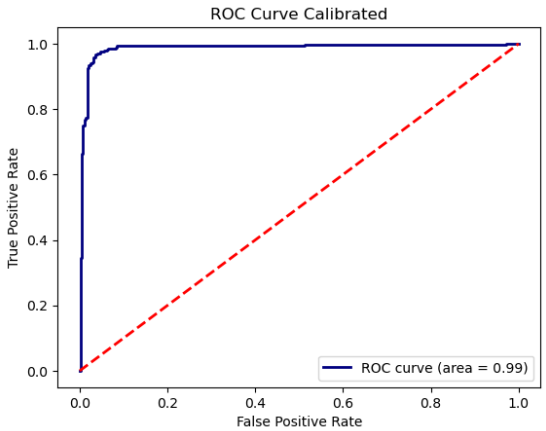

Figure 3: Curva ROC calibrado mediante el método sigmoid

Metricas modelo calibrado	
Accuracy	0.968944099378882
Precision	0.9694342331637642
Recall	0.0.9685990338164252
F1 Score	0.9688949476121367

Table 4: métricas obtenidas despues de calibrar

3.4 Curva de confiabilidad

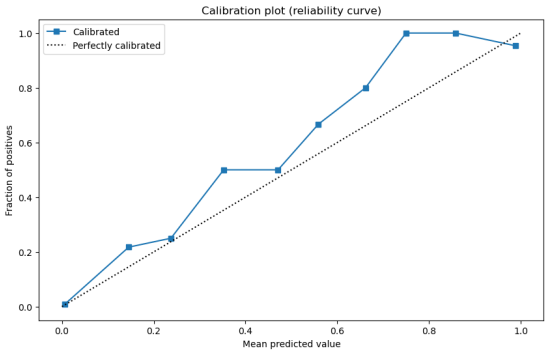

Figure 4: Curva de calibración para nuestro modelo

El gráfico que se muestra (4) es una curva de calibración, también conocida como curva de confiabilidad (reliability curve). Este gráfico se utiliza para evaluar la calidad de las probabilidades predichas por un modelo de clasificación. La línea discontinua negra representa una calibración perfecta, donde la fracción de positivos coincide exactamente con el valor medio predicho. La línea azul, por otro lado, muestra la calibración real del modelo después de ser ajustado usando regresión sigmoide.

Podemos extraer información de esta figura en tres rangos:

Rango de 0.0 a 0.25:

En este rango, el modelo sobrestima ligeramente la probabilidad de positivos. Observamos que la fracción de positivos (eje y) es menor que el valor medio predicho (eje x), lo que indica que el modelo predice una probabilidad de positivos más alta de lo que realmente se observa en los datos. Esta sobrestimación no es muy grande, pero es notable en este rango de bajas probabilidades.

Rango de 0.25 a 0.75:

En este intervalo, el modelo muestra cierta inconsistencia. En la franja de 0.25 a 0.5, el modelo subestima considerablemente la probabilidad de obtener positivos, ya que la fracción de positivos es mucho mayor que el valor medio predicho. Esto indica una calibración deficiente en esta parte del rango. A medida que nos acercamos al valor de 0.75, la calibración mejora y la línea azul se aproxima más a la diagonal negra, indicando una mejor correspondencia entre las predicciones y la realidad.

Rango de 0.75 a 1.0:

En el rango alto de probabilidades, el modelo tiende a sobrestimar la probabilidad de positivos. La línea azul está por encima de la línea negra, especialmente cerca de 0.9, lo que sugiere que el modelo predice una probabilidad de positivos más alta de lo que realmente se observa. Sin embargo, cerca del valor 1.0, el modelo se aproxima nuevamente a una buena calibración. De manera general, podemos decir que el modelo tiene problemas de calibración, sobrestimando las probabilidades en los rangos bajos y altos, y subestimando en el rango medio. Esto puede ocurrir debido a varias razones, como un desbalanceo en las etiquetas, es decir, hay más etiquetas de una clase que de otra, o que los hiperparámetros utilizados no fueron los ideales para la calibración. Para mejorar la calibración, se podrían considerar ajustes adicionales, como añadir más datos, aplicar técnicas de balanceo de clases, o realizar una validación cruzada más exhaustiva para optimizar los hiperparámetros.

4 Conclusiones

En este informe, se ha desarrollado un modelo de red neuronal convolucional para la clasificación de imágenes y la estimación de incertidumbre utilizando Monte Carlo Dropout. Además, hemos implementado una calibración de probabilidad mediante Platt Scaling para mejorar la confiabilidad de las predicciones del modelo.

Los resultados indican que el modelo sin calibrar ya presenta un buen desempeño con una alta precisión y un ROC-AUC significativo. Sin embargo, la implementación de Monte Carlo Dropout nos permitió identificar áreas de alta incertidumbre, lo cual es crucial para aplicaciones prácticas.

La calibración del modelo mostró un buen nivel en la calibración de las probabilidades predichas, como lo demuestra la curva de calibración y el Brier score (0.0405). Aunque las métricas de precisión, recall y F1 score no mostraron cambios drásticos, la exactitud global del modelo mejoró ligeramente.

En este caso si se tuviese que elegir entre el modelo calibrado y el no calibrado, la mejor opción podría ser el modelo calibrado. La principal razón es la mejora en la confiabilidad de las probabilidades predichas. En aplicaciones críticas, es vital que las probabilidades predichas reflejen fervientemente la realidad. La calibración del modelo asegura que las decisiones basadas en estas probabilidades sean más informadas y confiables. Además, la mejora en la curva ROC refuerza la elección del modelo calibrado, ya que proporciona una excelente representación de la incertidumbre en las predicciones.

En conclusión, mientras que ambos modelos muestran un rendimiento robusto, la calibración adicional aporta un valor significativo en términos de confiabilidad y precisión de las predicciones probabilísticas, haciendo que el modelo calibrado sea la opción preferida para aplicaciones prácticas que requieren alta confianza en las predicciones.

References

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 625–632, New York, NY, USA. Association for Computing Machinery.

Platt, J. (2000). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10.