

Assessing Difficulty with Information-Theoretic measures

Maximiliano Westerhout Aliste, Álvaro Gallardo Alvarado

Abstract—El proyecto se enfoca en explorar el impacto de la dificultad de los datos en el rendimiento de los modelos de aprendizaje automático. La dificultad de los datos es un concepto muy importante en el campo del aprendizaje automático, abarcando la complejidad asociada a tareas específicas, como la clasificación de imágenes o la generación de texto. Sin embargo, su naturaleza subjetiva dificulta su cuantificación precisa, lo que limita la comprensión profunda de los elementos que componen un problema dado. Para abordar esta limitación, se propone la utilización de métricas que cuantifiquen la dificultad de los datos, lo que podría proporcionar una herramienta valiosa para priorizar tareas y mejorar el desempeño de los modelos.

En este proyecto, se exploran dos métricas específicas: 'Prediction Depth' y 'CG-Score'. La primera métrica, propuesta en [1], relaciona la dificultad con la clasificación del algoritmo K-Nearest Neighbor (KNN) en las capas internas de modelos como ResNet18 y VGG16. Por otro lado, la segunda métrica, mencionada en [2], evalúa la irregularidad de las instancias dentro de cada clase al eliminar determinadas instancias del conjunto de datos.

A través de experimentos con el conjunto de datos CIFAR-10, se llevaron a cabo pruebas detalladas con ambos métodos. Para 'Prediction Depth', se examinó el rendimiento del KNN por capa, se compararon las diferencias entre las 'Prediction Depths' más pequeñas y más grandes, y se generaron histogramas para visualizar la distribución de la 'Prediction Depth' por clases. En cuanto al 'CG-Score', se calculó para el subconjunto completo y se analizó su variación por clase, explorando promedios, desviaciones estándar y comparaciones entre instancias de alto y bajo 'CG-Score' en relación con las imágenes correspondientes.

Los resultados obtenidos sugieren una relación entre la dificultad de los datos y las métricas ya que mantienen una consistencia en los patrones encontrados, respaldando la hipótesis de que medir la dificultad de los datos puede ser crucial para comprender y mejorar el desempeño de los modelos de aprendizaje automático.

Index Terms—Dificultad, CG Score, Prediction Depth, Algoritmo KNN, Generalización

I. INTRODUCCIÓN

La dificultad corresponde a una palabra que se menciona constantemente en el área del aprendizaje de máquinas, haciendo referencia a lo complicado que puede llegar a ser una tarea específica de un modelo, ya sea clasificación de imágenes, generación de texto, entrenamiento de un modelo, importancia de algunos elementos en el dataset, etc. La problemática de dicho concepto radica en que la

Este documento corresponde a una entrega final para el curso de Teoría de la Información (EL7024) de la Universidad de Chile con fecha 11-12-2023.

dificultad está asociada a algo subjetivo, siendo un concepto ampliamente utilizado pero no enumerable, generando incapaz a los investigadores de poder determinar de forma concreta que procesos son más difíciles, lo cual evita que sean capaces de poder entender en profundidad los distintos elementos que componen el problema tratado, siendo este cualquiera de los anteriores. Debido a esto, poder generar métricas que enumeren la dificultad permitiría ser una herramienta útil al momento de poder priorizar algunas tareas, lo cual se puede traducir en una mejora significativa en el desempeño de algunos modelos. En el presente proyecto se estudia el concepto de dificultad como una métrica asociada a la data, específicamente a la utilizada para el entrenamiento o evaluación de un modelo de machine learning, para ello se estudiaron dos métodos capaces de evaluar dicha métrica, siendo la primera 'Prediction Depth' y la segunda 'CG-Score'.

El primer método corresponde a un término acuñado en el paper [1] siendo una métrica que relaciona la dificultad con la clasificación del algoritmo KNN (K-Nearest-Neighbor) durante los resultados tempranos de las capas internas del modelo. En términos detallados, se utiliza un modelo alterado que retorna los resultados de cada capa del modelo los cuales pasan por un clasificador KNN, el cual retorna una etiqueta considerando como datos de entrada la salida de dicha capa. Dicho proceso busca comparar el resultado final de la clasificación del algoritmo, en relación a los resultados de cada capa, calculando la 'Prediction depth' como la diferencia entre la última capa del modelo y el resultado más temprano de clasificación KNN similar a la clasificación final, en otras palabras, si el resultado final de clasificación corresponde a la etiqueta 5 (correspondiente a la última capa), se busca la primera capa en donde el algoritmo KNN retorne la etiqueta 5, en donde si dicho valor se obtiene en la capa cuatro se considera la prediction depth como dicho valor.

El segundo método corresponde al término definido en el paper [2] siendo una métrica que relaciona la dificultad de los datos cuando una determinada instancia de datos se elimina del conjunto de datos completo. Es decir, dentro de un subdataset el modelo evalúa y genera una puntuación que cuantifica la "irregularidad" de las instancias dentro de cada clase,

Durante el presente informe, se detallan experimentos asociados a estas métricas con tal de respaldar las observaciones hechas por los investigadores, se recrean algunos de los experimentos al igual que se proponen otros, siendo todos trabajados con el dataset CIFAR-10. Para la primera métrica se realizaron

distintas pruebas con dos modelos distintos siendo el primero RESNET-18 y el segundo VGG-16, en ambos se observo una consistencia en los resultados llegando a la conclusión de que la prediction depth permite en efecto evaluar la dificultad del dataset en cuestión. Para el segundo caso se calculó el cg score para la muestra del dataset completo, utilizando el modelo RESNET-18, con ello se analizó su comportamiento por dato y por clases, junto con la dispersión y comportamiento.

II. PRELIMINARES O MARCO TEÓRICO

El método de prediction depth, utiliza el algoritmo K-Nearest-Neighbor el cual consiste en un algoritmo de clasificación que utiliza la proximidad para clasificar un punto determinado de la data. Lo que hace específicamente es que toma el punto analizado y calcula un radio de distancia, en donde dependiendo del numero de clases cercanas determina que clase pertenece el punto en cuestión, por ejemplo, si son dos clases y en el radio están presentes dos puntos de la clase A y uno solo de la B, entonces el punto se clasifica como A. La métrica asociada para calcular el radio es variable, del mismo modo que el porcentaje mínimo para determinar un punto desconocido siendo el parámetro K el valor asociado a dicha cantidad mínima, en donde si $k=1$, se considera únicamente al vecino mas cercano para tomar la decisión de clasificación. En este caso, el algoritmo KNN se aplica por medio de la creación de índices de la librería 'faiss', los cuales generan diccionarios de la información por capas generando índices que permiten calcular el algoritmo KNN, de este modo por medio de faiss se entrena el clasificador y se desarrollan las clasificaciones, siendo este proceso aplicable para todo modelo. La métrica utilizada para medir la distancia corresponde a la Euclidiana cuadrática (L2) siendo su formula la siguiente:

Sean dos puntos p y q , de n -dimensiones se tiene $p = (p_1, p_2, \dots, p_n)$ y $q = (q_1, q_2, \dots, q_n)$, la distancia L2 asociada a dichos puntos es la siguiente:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

Para el caso de los índices faiss, la distancia esta almacenada como una variable denominada 'D' y los id's como 'I', de este modo para el tensor consultado (imagen) se buscan el vector de distancias e índices mas cercanos. El valor del parámetro k corresponde a una proporción del numero de datos del dataste, siendo este caso:

$$K = 8 * \sqrt{L} \in \mathbb{N}$$

Siendo L el largo de los datos. Dicho valor es un hiperparametro escogido para la realización de todos los experimentos, corresponde a un porcentaje de aproximadamente el 25% de los datos, teniendo dicha formula para castigar en mayor medida el numero de los datos, aumentando progresivamente si el conjunto de datos es mayor. Esto ultimo se debe a que valores bajos de k por lo general presentan malas predicciones.

El método cg score, se establece como una métrica que se refiere a la diferencia entre la medida de complejidad de los datos cuando (x_i, y_i) se elimina de un conjunto de datos determinado (x_i, y_i)

$$CG(i) = y^T (H^\infty)^{-1} y - y_{-i}^T (H_{-i}^\infty) y_{-i}$$

con y^T el vector de etiqueta H^∞ la gran matrix asociada a la activación de ReLU, es decir con el entrenamiento de la red neuronal, dicho modelo no requiere tener una red neuronal para poder calcular la métrica pero esta mantiene dos propiedades importantes en el entrenamiento y generalización de instancias de datos,

- Esta métrica indica que un cg score grande se cataloga como un ejemplo "difícil", en el sentido de que eliminar dicha imagen del conjunto de datos reduce en gran medida el error de generalización, lo que implica que el conjunto de datos sin (x_i, y_i) es mucho más fácil de aprender y generalizar.
- Una instancia (x_i, y_i) con un cg score mayor contribuye más a la optimización y a las unidades del movimiento total de neuronas en una cantidad mayor, medida por

$$\|W(k) - W(0)\|_F.$$

esta se refiere a la norma del peso de la instancia hasta el peso inicial, es decir la distancia entre ellos.

Otra forma de caracterizar el cg score es debido a la interpretación geométrica que tiene esta métrica, esto quiere decir que se puede tener una muestra irregular o regular x_i, y_i en el sentido de que tenga una mayor o menor similitud a muestras de la misma clase que a muestras de una clase diferente, en este punto, para una muestra regular tendrá un valor bajo de cg score ya que este tiene gran similitud a las muestras de la misma clase (poca distancia entre ellas), por el contrario, tendría un cg score alto si no tiene similitud (muchas distancias entre ellas) a las muestras de la clase pero si a otras clases.

III. METODOLOGÍA O CONFIGURACIÓN EXPERIMENTAL

Como solución propuesta al problema se tiene que evaluar el impacto de la dificultad de los datos en el rendimiento de los modelos de aprendizaje automático. Para ello, se realizaron diferentes pasos, dentro de los cuales se encuentran:

- Selección de métricas de Dificultad de Datos:
Se implementaron diferentes métricas basadas en fundamentos teóricos de la información, como la predicción depth y cg score
- Generación de Modelos:
Se implementaron modelos de aprendizaje automático, en el caso de la primera métrica se utilizaron modelos de redes neuronales, específicamente ResNet18 y VGG16, mientras que para la segunda métrica se utilizó solamente el modelo ResNet18.
- Pruebas:
Las pruebas fueron utilizando el dataset CIFAR-10 para ambas métricas, al momento de realizar las pruebas se equilibraron las clases para el subconjunto que se utilizó,

para el caso de la segunda métrica además se dividió el subconjunto por clase que contenía el dataset.

Dentro de las pruebas que se realizaron para la primera métrica se encuentran verificar el accuracy del KNN por capa, comparar smallest prediction depth con largest prediction depth, comparar el accuracy del KNN con prediction depth y generar histogramas del predicción depth por las clases.

Por otro lado las pruebas que se realizaron para la segunda métrica fueron el cálculo del cg score para el subdataset completo junto con su visualización, dividir los cg scores por cada clase del dataset, calcular métricas como el promedio y la desviación estándar del cg score por clase (graficar la desviación estándar vs el promedio por clase), generar un gráfico de los cg score para cada clase y determinar un umbral para poder determinar cg score alto y bajo el promedio, comparar los cg scores altos y bajos de cada clase respecto a las imágenes que corresponden.

IV. RESULTADOS

Los experimentos realizados contemplan las limitaciones físicas de los equipos utilizados para realizar los experimentos, utilizando subsets del datasets CIFAR-10 y plataformas de programación tal como google colab, los recursos en específico corresponden a 12.7 GB de RAM, 17.8 GB de GPU Tesla 4 y 30 GB de disco duro.

Para el primer y segundo experimento, se utiliza un subset de la data CIFAR-10 específicamente 1024 datos, para el primer caso no se aplica ningún tipo de equilibrio de clases y para el segundo si, teniendo 103 imágenes para las primeras cuatro y 102 para el resto (1024 en total). El tamaño de los datos corresponde al numero máximo de datos capaz de procesar en la implementación del algoritmo KNN, aplicando la búsqueda de índices para las mismas imágenes, se destaca que no se aplica entrenamiento para el clasificador KNN y que únicamente se lleva acabo la creación de los índices y la búsqueda exhaustiva en todo el dataset observado. El modelo aplicado corresponde a una RESNET-18 pre-entrenada con los pesos de la librería pytorch, obteniendo los siguientes resultados.

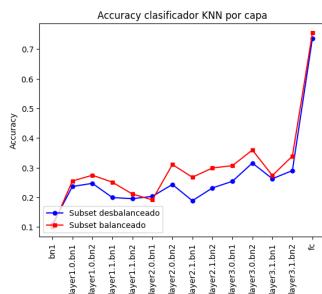


Fig. 1. Accuracy del clasificador KNN para las distintas capas.

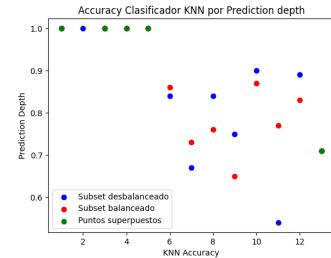


Fig. 2. Accuracy del clasificador KNN para las distintas Prediction Depths.

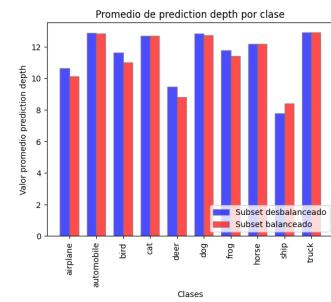


Fig. 3. Promedio prediction depth por clase para ambos subsets.

El accuracy de cada etiqueta para ambos experimentos esta en la tabla 1 y 2:

TABLE I
ACCURACY DE CLASES PARA EL SUBSET DESBALANCEADO.

Label	N Correctas	N Incorrectas	Accuracy (%)
0	79	24	76.699
1	112	3	97.391
2	39	63	38.235
3	70	24	74.468
4	84	21	80.0
5	41	45	47.674
6	95	16	85.586
7	63	40	61.165
8	80	21	79.208
9	90	14	86.538

TABLE II
ACCURACY DE LAS CLASES PARA EL SUBSET BALANCEADO.

Label	N Correctas	N Incorrectas	Accuracy (%)
0	80	23	77.67
1	100	3	97.087
2	54	49	52.427
3	80	23	77.67
4	82	20	80.392
5	50	52	49.02
6	85	17	83.333
7	65	37	63.725
8	93	9	91.176
9	83	19	81.373

El experimento tres considero el uso nuevamente de un modelo RESNET-18 pre-entrenado, pero en este caso se utilizaron 640 imágenes con un equilibrio entre clases (64 por clase). Dicho valor se utilizo debido que para este caso se

aplico un entrenamiento al clasificador KNN y se realiza la búsqueda de índices en imágenes aleatoria de todo el dataset (no necesariamente de las 640 presentes) en un total de 1024 por medio de la función np.random.choice (la cual mantiene cierto grado de representatividad), obteniendo los siguientes resultados en comparación al modelo con subsets balanceados.

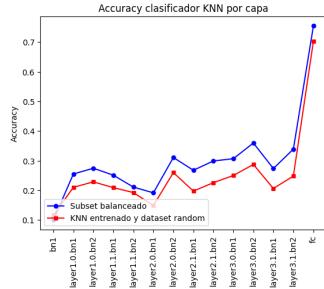


Fig. 4. Accuracy del clasificador KNN para las distintas capas.

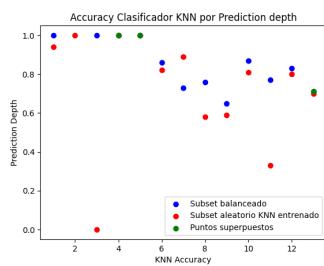


Fig. 5. Accuracy del clasificador KNN para las distintas Prediction Depths.

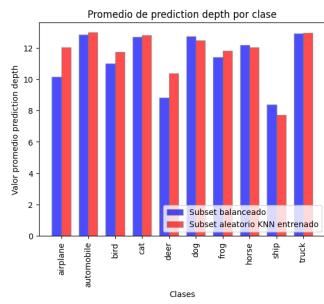


Fig. 6. Promedio prediction depth por clase para ambos subsets.

Siendo los accuracy por clase para este nuevo modelo los siguientes:

TABLE III
ACCURACY PARA EL MODELO KNN ENTRENADO.

Label	N Correctas	N Incorrectas	Accuracy (%)
0	65	34	65.657
1	92	8	92.0
2	31	49	38.75
3	81	30	72.973
4	69	27	71.875
5	50	58	46.296
6	89	18	83.178
7	60	39	60.606
8	88	18	83.019
9	94	24	79.661

Finalmente, se probó esta métrica con otra arquitectura siendo esta una VGG-16, se realizó un solo experimento debido a que el resto de los intentos falló en términos computacionales. Para dicho caso se entreno con 240 datos equilibrados y no se entreno el clasificador KNN, generando una busca exhaustiva para los 240 datos. Dichos resultados son los siguientes.

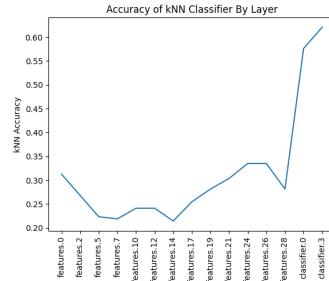


Fig. 7. Accuracy del clasificador KNN para las distintas capas.

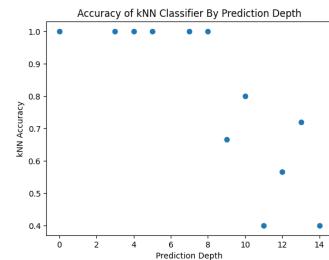


Fig. 8. Accuracy del clasificador KNN para las distintas Prediction Depths.

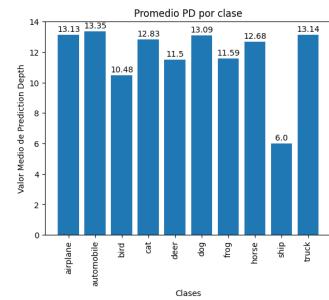


Fig. 9. Promedio prediction depth por el modelo VGG-16.

Finalmente el accuracy por clase para este modelo corresponde al presente en la tabla 4.

TABLE IV
PRECISIONES PARA DISTINTAS CLASES VGG-16.

Label	N Correctas	N Incorrectas	Accuracy (%)
0	8	15	34.783
1	14	9	60.87
2	10	13	43.478
3	7	16	30.435
4	19	3	86.364
5	4	18	18.182
6	20	2	90.909
7	7	15	31.818
8	21	1	95.455
9	19	3	86.364

La siguiente métrica, se utilizó un subset de datos de CIFAR-10 específicamente 1024 datos, dicho subconjunto contempla una representatividad ya que se utiliza clases equilibradas. El tamaño de los datos corresponde al numero máximo de datos capaz de procesar en la implementación del cg score en un computador de forma local.

En primer lugar se calculo el promedio que se tiene por clase de los cg scores calculados, en el cual se puede ver que el promedio mayor es con la clase 3 y el menor con la clase 6.

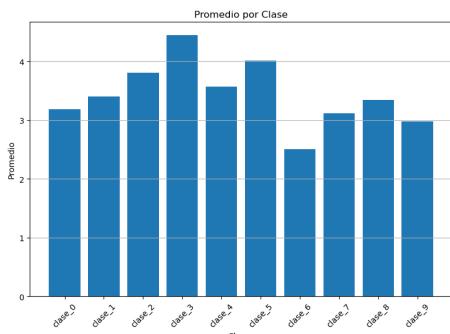


Fig. 10. Promedio de cg score por clases para CIFAR-10.

Luego se calcula la dispersión de los datos respecto al promedio y se gráfica el comportamiento del cg score para las clases 3 y 6 manteniendo una franja roja que corresponde al promedio de dicha métrica para tal clase.

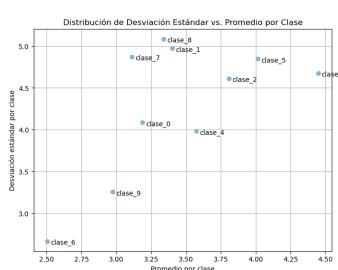


Fig. 11. Promedio de cg score versus desviación estándar para CIFAR-10.

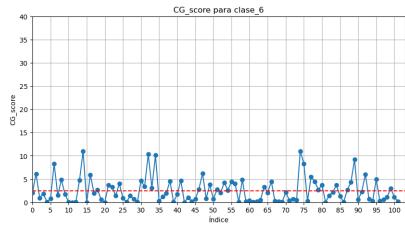


Fig. 12. cg score de la clase 6 para CIFAR-10.

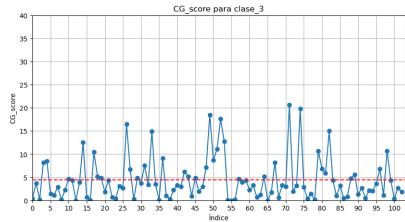


Fig. 13. cg score de la clase 3 para CIFAR-10.

Como segundo experimento se visualizó las imágenes que corresponden a los 8 cg scores más altos y bajos para las clases 3 y 6 respectivamente.



Fig. 14. fotos que corresponden al mayor valor de cg score de la clase 3 para CIFAR-10.



Fig. 15. fotos que corresponden al menor valor de cg score de la clase 3 para CIFAR-10.



Fig. 16. fotos que corresponden al mayor valor de cg score de la clase 6 para CIFAR-10.



Fig. 17. fotos que corresponden al menor valor de cg score de la clase 6 para CIFAR-10.

V. DISCUSIÓN

Para la primera métrica prediction depth, se observan distintos resultados dependientes del modelo, tipo de algoritmo KNN implementado (entrenado o no) y tipo de dataset. Para los primeros tres resultados, se hacen comparaciones entre tener un subset de datos equilibrados y uno desequilibrado, notando que en efecto existen pequeñas variaciones del accuracy del KNN. Este resultado tiene una clara relación debido a que poseer menos elementos de una clase genera que tener menos aciertos afecte de mayor forma la métrica de accuracy, dado que el porcentaje de dicho error es mayor si es que el total de la clase es un numero pequeño, por ello es esperable notar variaciones en dicha métrica que es en efecto lo que se genera pero independiente de ello se nota una consistencia en las clasificaciones del KNN y el valor de prediction depth, en donde independiente de las fluctuaciones del accuracy los promedios de PD y los valores captados, son prácticamente los mismos. Este resultado es interesante, dado que la PD depende directamente del conjunto del dataset analizado en donde uno esperaría que un desbalance en las clases altere de manera drástica los valores promedios, pero esto no sucede, dicho resultado se debe a que el PD mantiene una consistencia mas relacionada a las características de la clase mas que cada dato en si.

Referido a los resultados de entrenar o no el clasificador KNN, primero se debe entender que implica no entrenar dicho clasificador. El algoritmo crea índices relacionados a cada distancia entre imágenes en comparación al resto de los datos del conjunto estudiado, en donde si no se genera un entrenamiento pero se aplica una búsqueda exhaustiva a lo largo de todo el conjunto utilizado para crear los índices (no debe ser todo el dataset), dicha búsqueda es similar a entrenar el modelo y buscar en datos externos a los utilizados. Básicamente, el overfitting del clasificador de la data genera resultados similares a los del entrenamiento, siendo este resultado producto de reiteradas iteraciones experimentales. Una de las principales razones de por que esto ocurre se debe a que el análisis de esta métrica no tiene un propósito predictivo, sino mas bien de análisis del dataset, en donde se busca medir la dificultad por medio de la consistencia de un solo clasificador sin ser el objetivo principal el accuracy de dichas clasificaciones. En otras palabras, se busca observar como opera el clasificador independiente de si la predicciones es correcta o no, saber si un conjunto de datos aparece en predicciones tempranas permite saber si el modelo considera

dicha clasificación como una operación 'fácil' o 'difícil' en donde el accuracy de la clasificación no es tan relevante como conocer dicha información, por lo cual si el clasificador esta overfitting tendrá a tener un mejor accuracy en la predicción pero las clases presentes en la capa serán las mismas para el algoritmo entrenado, que es en efecto lo que se observa en las figuras 4 y 5. Esta observación es bastante interesante dado que considera una solución viable para análisis de bajo costo, en donde implementar este código sin entrenamiento del clasificador aporta prácticamente la misma información que llevar acabo un entrenamiento costoso (en termino computacionales) del algoritmo. Cabe destacar, que dentro del código anexado están presentes tablas de las clases presentes en las capas tempranas (de la 0 a la 5) en donde se mantiene que para ambos modelos las clases presentes en dichas capas son siempre las 0,2 y 8, del mismo modo que los promedios de PD por clase son básicamente los mismos, además se destaca que para algunas capas existe el mismo porcentaje de accuracy por clase como es por ejemplo para PD igual a 1, en donde ambos modelos presentan 100% de accuracy de la clase 8 para 9 y 16 imágenes.

Por ultimo, referido a la diferencia entre modelos se tiene que los resultados arrojan una consistencia respecto a los anteriores, en donde inclusive para el nuevo modelo que posee un mayor numero de capas el ranking de PD es básicamente el mismo, manteniéndose que la clase 8 posee el menor valor de PD seguido por la clase 2 y 4 respectivamente, nuevamente este resultado se debe a que esta métrica esta asociada a las datos específicamente a las clases y no al modelo o al numero, por ello independiente de que este modelo haya sido ejecutado con un menor numero de datos e independiente del numero de capas del modelo, los resultados fueron practicamente los mismos. Este resultado es mencionada en el paper [1], en donde probaron múltiples modelos con distintas arquitecturas llegando a lo mismo, es interesante dicha conclusión debido a que permite a los investigadorxs optimizar el uso de sus recursos utilizando modelos mas sencillos para generar análisis de dificultad, y evitar utilizar modelos con gran cantidad de capas que utilicen mayores recursos computacionales. Cabe destacar que incluso para este modelo, las clases 2 y 8 estuvieron presentes en las capas tempranas, la clase cero no aparece pero se debe principalmente a la poca cantidad de datos presentes (en general la clase 0 estaba en menor proporción que la 2 y 8 para los otros modelos).

En base a los resultados obtenidos en la segunda métrica analizada sobre el cg score por clase y sus implicaciones para abordar la dificultad en los datos en términos de aprendizaje automático podemos decir que para el promedio de cg score por clase, la clase 3 muestra que tiene un promedio más alto en comparación con otras clases, lo que sugiere que estas instancias son mas difíciles para el modelo. Por otro lado, la clase 6 tiene el cg score promedio más bajo, lo que indica que estas instancias son relativamente más fáciles de aprender y generalizar.

En relación entre el promedio y desviación estándar por clase se puede decir que la alta desviación estándar junto

con el mayor promedio de la clase 3 indican que esta clase no solo es difícil, sino que también es más dispersa en términos de dificultad. Mientras que la clase 6 muestra tanto un promedio bajo como una baja dispersión, lo que indica una consistencia en su nivel de dificultad.

Para el análisis de las imágenes con cg score alto y bajo, se puede ver que para la clase 3, las imágenes con cg score alto (gatos de cuerpo completo y fondos coloridos) son más desafiantes, mientras que las de cg score bajo que corresponden más específicamente a la cara de los gatos son menos complejas. Para la clase 6, las imágenes con cg score alto que corresponden a sapos en entornos diferentes, representan instancias más difíciles, mientras que las que tienen cg score bajo, es decir las imágenes que son por lo general de sapos en fondos verdes y más uniformes son menos desafiantes.

En base a esto podemos decir que el cg score representa la diferencia en la medida de complejidad de los datos al eliminar una instancia en particular, por lo que un cg score alto indica que esa instancia es más compleja (difícil) para el modelo. A su vez, una clase que contiene instancias más complejas para el modelo, es decir que tiene mayor complejidad se puede traducir en una mayor cantidad de información o incertidumbre.

Métricas como la desviación estándar y dispersión nos indican que la complejidad también se puede caracterizar como una instancia con una distribución más dispersa.

En síntesis, podemos decir que las imágenes con cg score alto en la clase 3 y 6 pueden interpretarse como aquellas que aportan más información única o más difícil de clasificar para el modelo, también se puede ver que tienen una menor similitud a muestras de la misma clase. Mientras que las imágenes con cg score bajo pueden ser más predecibles o tener menos información adicional en términos de clasificación y tienen una mayor similitud a muestras de la misma clase. Por lo que si se quiere simplificar el problema para reducir la cantidad de información difícil o complejidad para un modelo se puede eliminar la clase 3 con el fin de generalizarlo.

VI. CONCLUSIONES

En este proyecto, se abordó la compleja noción de la dificultad de los datos en el contexto del aprendizaje automático. La exploración de métricas como 'Prediction Depth' y 'CG score' permitió una comprensión más profunda de cómo la dificultad de los datos puede impactar en el rendimiento de algún modelo que utilice el dataset. Los experimentos realizados con modelos como ResNet18 y VGG16 utilizando el conjunto de datos CIFAR-10, revelaron patrones consistentes que respaldan la idea de que la dificultad de los datos juega un importante rol en la capacidad de generalización y desempeño de los modelos de aprendizaje automático.

La capacidad de cuantificar la dificultad de los datos proporciona una herramienta valiosa para futuras investigaciones sobre el aprendizaje automático. Estas métricas pueden servir como guía para priorizar tareas, datos, mejorar la comprensión del comportamiento del modelo y potencialmente, desarrollar estrategias más efectivas de entrenamiento y evaluación de modelos.

Sin embargo, se puede concluir que este estudio representa solo un primer paso en la comprensión de la dificultad de los datos. Se requiere una validación más extensa en diversos conjuntos de datos y entornos para confirmar la generalización de estas conclusiones.

En conclusión, este proyecto ha demostrado la relevancia y el potencial de comprender la dificultad de los datos en el diseño y mejora de modelos de aprendizaje automático, abriendo nuevas puertas para investigaciones futuras y aplicaciones prácticas en esta área que mantiene un constante desarrollo.

Nota: Esta sección es requerida solamente para la entrega final. Discutir el aporte e impacto de su trabajo, además de posibles avenidas para trabajo futuro.

REFERENCES

[1] R. J. N. Baldock, H. Maennel, and B. Neyshabur, "Deep learning through the lens of example difficulty," in Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual (M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, eds.), pp. 10876–10889, 2021.

[2] N. Ki, H. Choi, and H. W. Chung, "Data valuation without training of a model," in The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net, 2023.

[3] Pikus16, "ExampleDifficultyScale", in <https://github.com/Pikus16/ExampleDifficultyScale/tree/main>

[4] JJchy, "CG_score", in <https://github.com/JJchy/CGscore>

Maximiliano Westerhout cg score, Prediction depth, informe. Análisis general

Álvaro Gallardo cg score, Prediction depth, informe. Análisis general