Estudio del conocimiento actual en la compresión de datos para el aprendizaje automático

Ricardo Peralta Egea

Trabajo Fin de Grado

Supervisado por
Rocío González Díaz – Eduardo Paluzo Hidalgo

Universidad de Sevilla

junio 2023

Publicado en junio 2023 por
Ricardo Peralta Egea

ricardoperalta00@gmail.com

.

Yo, D. Ricardo Peralta Egea con NIF número 26514202D,

DECLARO

mi autoría del trabajo que se presenta en la memoria de este trabajo fin de grado que tiene por título:

Estudio del conocimiento actual en la compresión de datos para el aprendizaje automático

Lo cual firmo,

Fdo. D. Ricardo Peralta Egea
en la Universidad de Sevilla
28/06/2023

Quiero dedicar este trabajo a mi familia, cuyo apoyo ha sido fundamental en mi crecimiento tanto personal como académico al guiarme hacia el logro de mis metas.

Asimismo, quiero extender mi dedicatoria a mis amigos, que se han convertido en familia durante mi etapa universitaria. En particular, quiero hacer una especial mención a Unai, quien ha compartido valiosos consejos y enseñanzas que han enriquecido mi desarrollo. A Cristian le agradezco transmitirme la alegría de vivir, dándome una energía positiva que ha impulsado mi motivación y perseverancia. Finalmente a María por su apoyo diario y estar presente en cada paso de mi trayectoria, brindándome amistad y compañía.

Espero que este gesto de reconocimiento refleje mi gratitud hacia aquellos que han estado a mi lado. Sin su apoyo, este camino no hubiera sido el mismo.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas aquellas personas que contribuyeron de manera significativa al desarrollo de este trabajo.

En primer lugar, deseo agradecer a mis tutores Rocío González Díaz y Eduardo Paluzo Hidalgo por su orientación y dedicación a lo largo del proyecto. Sus conocimientos y consejos fueron fundamentales para el desarrollo del mismo.

Del mismo modo agradecer a todos los investigadores e investigadoras de todas partes de mundo que generosamente dedicaron su tiempo y colaboraron con sus trabajos y experiencias. Sin su participación este trabajo no hubiese sido posible.

También quiero agradecer a mi familia y amigos, quienes siempre estuvieron a mi lado, brindándome su incondicional apoyo emocional y motivación. Su confianza en mi fueron cruciales para superar desafíos y obstáculos.

Por último, quiero reconocer a mis compañeros y compañeras, quienes compartieron conmigo sus conocimientos y experiencias, enriqueciendo mi experiencia académica.

A todas las personas mencionadas y a aquellas que, de alguna manera, han dejado su huella en este proyecto, les expreso mi profundo agradecimiento.

Gracias.

Abstract

In this final degree proyect, both the current methods and the context in the field of companies, researchs, and organizations surrounding the term *Green AI* are studied, providing a deeper focus on data reduction in the training phase owing to the fact that if we reduce the data quantity, the model will performance less operations obtaining a faster and greener process.

An analysis of related works is presented, covering from companies to research papers, while previously introducing the necessary concepts for their understanding. I have designed and implemented two methods, the code can be found on GitHub. One of the methods I have implemented deals with representative data splitting, while the other involves selecting data with more information content. Finally, I made a comparison between both methods.

Resumen

En este trabajo fin de grado se estudian tantos los métodos actuales como el contexto en el ámbito de empresas, investigaciones y organizaciones que se cierne sobre el término Inteligencia Artificial Sostenible (*Green AI*) dando un enfoque más profundo a la reducción de datos en la fase de entrenamiento, ya que, al tener menos datos se realizarán menos operaciones y el entrenamiento será a la vez que más rápido, más sostenible.

El documento presenta un análisis de trabajos relacionados abarcando desde empresas hasta artículos de investigación, pero introduciendo anteriormente los conceptos necesarios para su entendimiento. He realizado el diseño e implementación de dos métodos, el código se puede encontrar en GitHub, uno de los métodos que he implementado trata sobre cómo dividir el conjunto de manera representativa y el otro sobre cómo escoger datos con mayor información. Finalmente, hice una comparativa entre ambos métodos.

Índice general

Índice de figuras

Índice de tablas

Introducción

Con el problema de la creciente tendencia del uso de grandes conjuntos de datos en la fase de entrenamiento de los modelos de inteligencia artificial (IA), también conocido como la naturaleza *Data-Hungry* de la IA, se ha evidenciado el coste computacional que este posee y la huella de carbono que llega a liberar. Un ejemplo de ello se puede ver con el reciente y famoso chat GPT-3 con el que se ha estimado que en su fase de entrenamiento ha consumido 1.287 MWh traduciéndose en una emisión de 552 toneladas de CO^2 [14]. Debido a las cantidades exagerada de contaminación, ha nacido una corriente dentro de la inteligencia artificial conocida como inteligencia artificial sostenible (*Green AI*). Esta nueva corriente busca desarrollar métodos y enfoques que minimicen la huella de carbono de los modelos de inteligencia artificial y promuevan la adopción de prácticas más responsables desde el punto de vista ambiental. A medida que la IA se vuelve más presente en nuestra sociedad, también aumenta su consumo de energía y por tanto, sus emisiones de carbono asociadas. Esto plantea una cuestión crucial : ¿Cómo podemos aprovechar los beneficios de la IA sin comprometer el medio ambiente?

La respuesta se encuentra en la adopción de prácticas y políticas que promuevan la sostenibilidad en todas las etapas del ciclo de vida de los modelos de IA. Esto implica desde el desarrollo de algoritmos que mejoren la eficiencia del modelo hasta generar conciencia y responsabilidad en la comunidad de investigadores, desarrolladores y usuarios. Se trata de fomentar un ambiente colaborativo entre todos, incluyendo investigadores, empresas, gobiernos... para encontrar soluciones innovadoras y promover un uso de la inteligencia artificial más respetuosa con el medio ambiente.

Este trabajo fin de grado nace con el objetivo de contrarrestar la naturaleza *Data-Hungry* de los modelos de IA desde un enfoque posicionado en la reducción de datos y servir de aportación para la nueva corriente de la *Green AI*. Se dice que un modelo posee esta naturaleza cuando hace uso de una cantidad exagerada de datos en su fase de entrenamiento para poder realizar predicciones precisas. En el presente estudio, se ha realizado un trabajo bibliográfico en el campo de la reducción de datos con el objetivo de analizar las metodologías existentes y explorar las diferentes estrategias utilizadas

para dicha reducción. Se han revisado numerosas fuentes bibliográficas, incluyendo investigaciones académicas, organizaciones y trabajos previos relacionados con el tema. Este análisis ha permitido identificar las técnicas más relevantes en este ámbito. Además, se han realizado experimentaciones que se pueden consultar en mi repositorio de GitHub, donde la reducción de datos ha demostrado ser efectiva en mejorar la eficiencia, el rendimiento y la sostenibilidad de los modelos de inteligencia artificial.

Este trabajo está divido por secciones que serán comentadas a continuación. La sección §2 novela la metodología que he llevado a cabo durante este proyecto, seguidamente se encuentra la sección §3 donde se detalla la planificación del proyecto y sus costes. Después la sección §4 que va a servir de herramienta para la comprensión de conocimientos necesarios para la lectura del documento. Para tener una visión de los trabajos relacionados con este tema, la sección §5 estudia las organizaciones, centros de datos, empresas e investigaciones que se enmarcan en esta corriente. Quise profundizar en dos técnicas haciendo una implementación de las dos y experimentando con diferentes conjuntos, por lo que, en la sección §6.1 y la sección §7.1 se pueden encontrar la descripción, implementación y experimentación de las mismas. Finalmente, hago una comparativa de las técnicas mencionadas en §8 y una conclusión final en la sección §9.

Metodología

Para la realización de este Trabajo de Fin de Grado, he llevado a cabo un trabajo bibliográfico y de investigación que me ha llevado a contactar con diferentes investigadores de todo el mundo. La metodología que he seguido empieza por la búsqueda de artículos, empresas y organizaciones que tuvieran relación con la reducción de datos o la creación de subconjuntos representativos con un enfoque ecológico. Después de mi pesquisa en solitario, mis tutores me propusieron contactar con otros investigadores para tener diferentes puntos de vista y poder desarrollar de una mejor manera mi trabajo. Por consiguiente, contacté con los autores de los artículos de investigación que me parecieron que se ajustaban mejor a mi proyecto. Obtuve respuesta de varios investigadores, empezando por Manuel Fernández, investigador de CiTIUS de la Universidad de Santiago, que me dio una serie de pautas sobre dónde empezar a buscar. Por otro lado, contacté con Jens Groger, investigador en el *Instituto Oko* en Alemania, que no poseía en esos momentos información sobre como contrarrestar el *Data-Hungry* de los modelos. Sin embargo, su equipo y él están trabajando en desarrollar un modelo de referencia para evaluar las aplicaciones de la inteligencia artificial, incluyendo el esfuerzo del hardware y los efectos externos. Además Jens Groger me puso en contacto con compañeros que podrían ayudarme. Uno de ellos fue Achim Guldner, investigador en la Universidad de Birkenfeld en Alemania. Le comenté lo que estaba realizando, así como mis intenciones, y le agradó la idea. Me envió algunas de las investigaciones realizadas por él y otras realizados por compañeros/as que ayudaron al desarrollo de este TFG. También como propuesta futura, Achim Guldner sugirió hacer una colaboración. Tras esto, me comuniqué con un integrante del grupo de investigación portugués *Green Software Lab*, Rui Maranhão, que me comentó que tenía un estudiante realizando un estudio parecido y me puso en contacto con él. Era un estudiante de la Universidad de Delft, llamado Abel Van Steenweghen. Tras acordar una videollamada, comentamos en ella nuestras investigaciones, por su parte, me explicó cómo comprimir modelos de aprendizaje profundo. Esta videollamada fue beneficiosa para ambos ya que pudimos obtener diferentes puntos de vista sobre cómo hacer un modelo más rápido a la vez que verde.

De igual forma, me quise poner en contacto con asociaciones y empresas que estaban interesadas en el software sostenible. La primera de ellas fue *Green Software Foundation*, a la que pregunté si tenían algún proyecto relacionado con el tema del TFG y, si fuese el caso, le solicité asistir a alguna conferencia. No obstante, en esos momentos, no estaban enfocados en la reducción de datos, ya que estaban trabajando en varios proyectos sobre cómo medir la cantidad de carbono de un proceso y en la construcción de un ecosistema compuesto de personas, estándares, herramientas y mejores prácticas para crear y construir software sostenible. Por otra parte, realicé la misma consulta a *Green Code Berlin* y me mostraron que estaban trabajando en métricas para poder cuantificar cómo de sostenible es una arquitectura software. Todo esto me llevó a concluir que las empresas que ubiqué se centran más en cómo medir el efecto ambiental del software y dar pautas sobre cómo hacer un código más sostenible en vez de innovación de métodos para reducir recursos.

Respecto al desarrollo del trabajo, se ha mantenido un seguimiento continuo mediante reuniones cada dos semanas con Rocío González y Eduardo Paluzo, tutores de este TFG.

La primera fase fue de investigación. Estuve febrero y marzo documentándome sobre el contexto y las investigaciones actuales que tenían un enfoque sostenible de la inteligencia artificial. La segunda fase, que fue entre marzo y abril, consistió en la redacción, implementación y experimentación de algunas de las técnicas encontradas. Finalmente, la última fase, en mayo y junio, siguió con la experimentación de dichas técnicas y mejora del documento.

Análisis temporal y costes de desarrollo

En esta sección se exponen la planificación y el coste estimado para la realización de este proyecto. La planificación se encuentra en primer lugar y detalla las actividades clave con horas previstas, haciendo una comparación con las horas estimadas y el tiempo real que se ha dedicado. Esta planificación se fue modificando a lo largo del proyecto.

En segundo lugar, se encuentran los coste que proporcionan una estimación de los recursos financieros necesarios para llevar a cabo el proyecto. Es importante tener en cuenta que el coste estimado es una aproximación y puede estar sujeto a cambios.

Planificación

La planificación se realizó para hacer la entrega en la primera convocatoria. Sin embargo, se realizó finalmente para la segunda, prolongando el proyecto un mes más. Las fases en las que se divide un proyecto son: iniciación, planificación, ejecución y control y seguimiento. La tabla §3.1 presenta las fechas estimadas y reales de estas fases.

-	Iniciación	Planificación	Ejecución	Control y Seguimiento
Inicio	20/01/2023	08/02/2023	15/02/2023	20/01/2023
Final estimado	02/02/2023	15/02/2023	30/05/2023	30/05/2023
Final real	02/02/2023	15/02/2023	29/06/2023	29/06/2023

Tabla 3.1: Organización de tiempos por fases

Al ser este TFG un estudio sobre un campo concreto de la informática, las tareas durante la ejecución del proyecto son similares. Estas tareas son: corrección de errores, realizar lecturas analíticas, seleccionar información y hacer construcciones teóricas. Cada dos semanas se han llevado a cabo reuniones de control y seguimiento. Finalmente también he considerado añadir el tiempo para preparación de la presentación. La tabla §3.2 muestra la planificación inicial por tareas.

Tarea	Horas estimadas
Planificación	12:00h
Lecturas	73:00h
Selección e Interpretación	52:00h
Construcción teórica	95:00h
Corrección de errores	47:00h
Reuniones	8:00h
Slides y presentación	13:00h
Total	300:00h

Tabla 3.2: Tiempos de planificación inicial

En un principio no iba a añadir experimentación, por lo que, la planificación inicial se vio modificada durante el proyecto. También, al prolongar el proyecto un mes más, se realizaron más reuniones. La tabla muestra las horas previstas con las horas reales.

Tarea	Horas estimadas	Horas reales
Planificación	12:00h	12:00h
Lecturas	73:00h	63:45h
Selección e Interpretación	52:00h	35:30h
Construcción teórica	95:00h	87:00h
Corrección de errores	47:00h	42:00h
Implementación	-	37:00h
Experimentación	-	9:15h
Reuniones	8:00h	9:30h
Slides y presentación	13:00h	13:00h
Total	300:00h	309:00h

Tabla 3.3: Tiempos de planificación inicial

Costes

Otros elementos a considerar en el desarrollo del proyecto son los gastos asociados al personal, los gastos relacionados con servicios externos y las amortizaciones. Los factores financieros son fundamentales para evaluar la rentabilidad y elaborar un presupuesto completo que cubra todas las áreas para tener un proyecto viable y exitoso.

Al calcular el coste de personal para este proyecto, nos enfocamos únicamente en el coste directo asociado a la persona involucrada. En el caso de un ingeniero informático con un título recién obtenido, podemos determinar el coste basándonos en las horas trabajadas.

Puesto	Sueldo	Horas	Total
Ingeniero informático	13.85 €/h	309h	4279.65 €

Tabla 3.4: Costes de personal para el proyecto

Al considerar los costes de servicio en este proyecto, me centro exclusivamente en los servicios básicos consumidos, como el acceso a Internet y el suministro de energía eléctrica. Para calcular el coste de la luz, haré una aproximación utilizando el promedio mensual de consumo.

Servicio	€/mes	Meses de consumo	Total
Luz	45 €	5	225 €
Internet	36 €	5	180 €
Total			405 €

Tabla 3.5: Costes de servicio para el proyecto

El último paso consiste en calcular las amortizaciones del proyecto, que se refieren específicamente a los equipos utilizados en su desarrollo. Es importante destacar que los sistemas y programas utilizados en la ejecución del proyecto no han incurrido en ningún coste adicional. Para determinar la amortización, se utilizará un porcentaje anual del 20 %, tal como lo establece la Agencia Tributaria según las directrices del Gobierno de España, en su Ministerio de Hacienda y Función Pública.

Equipo	Coste	Meses de uso	Total
Xiaomi mi air 13.3	699 €	5	139.8 €

Tabla 3.6: Amortizaciones del proyecto

A continuación, se presenta una tabla que resume los costes totales del proyecto después de considerar todos los elementos mencionados anteriormente: costes de personal, costes de servicio y amortizaciones.

Tipo	Coste
Coste de personal	4279.65 €
Coste de servicios	405 €
Amortizaciones	139.8 €
Total	4824.45 €

Tabla 3.7: Costes totales asociados al proyecto

Conceptos Previos

Para poder tener un mayor entendimiento del trabajo, es necesario tener presente algunos conocimientos previos que van a ser mencionados a lo largo de todo el proyecto. Por ello, esta sección se ha diseñado para proporcionar una comprensión de los conceptos clave presentes en el trabajo. Esta recopilación tiene como objetivo facilitar y que sirva de herramienta de referencia especialmente para lectores menos familiarizados con los términos más técnicos.

4.1 Sobre el contexto

Uno de los objetivos del trabajo es buscar la reducción de datos para que la contaminación que producen los modelos de aprendizaje automático disminuya. Por tanto, para entender el contexto en el que se enmarca este proyecto, es necesario hablar sobre *Green software*, que es una disciplina cuyo objetivo consiste en lograr un software eficiente de forma que tenga un impacto medioambiental mínimo o nulo. En la misma línea, *Data-Hungry* se refiere a cuando un modelo hace uso de una cantidad exagerada de datos para un entrenamiento efectivo y un rendimiento óptimo. Cuantas más datos, más operaciones y por lo tanto más *huella de carbono*, métrica ambiental, la cual mide la totalidad de gases de efecto invernadero emitidos por un evento, persona o acción. Todas estas operaciones se realizan en *centros de datos* que son instalaciones físicas que albergan gran cantidad de infraestructura tecnológica como servidores, equipos de almacenamiento... Estos están diseñados para almacenar, procesar, administrar y distribuir grandes cantidades de datos.

4.2 Modelos de aprendizaje automático

Para hablar de modelos de aprendizaje automático, debemos introducir el término *inteligencia artificial*. Este es un campo de estudio en la informática que combina algoritmos y técnicas con el propósito de crear modelos que simulen y repliquen ciertas capacidades cognitivas y de aprendizaje vistas en la naturaleza como la inteligencia de

enjambre que replica el comportamiento colectivo de especies como la hormiga. Hoy en día, la inteligencia artificial posee varios enfoques como el procesamiento de lenguaje natural, la robótica, la planificación y toma de decisiones entre otras. Una de las ramas de la inteligencia artificial es el *aprendizaje automático* o también conocido en inglés como *machine learning* que se centra en el desarrollo de algoritmos y modelos que pueden aprender de los datos y mejorar su rendimiento sin ser explícitamente programados. El aprendizaje automático utiliza técnicas de estadística y matemáticas para desarrollar modelos que pueden hacer tareas específicas como clasificación, regresión, clustering, reconocimiento de patrones... Dentro de esta rama encontramos diferentes visiones:

- Aprendizaje supervisado: los modelos se entrenan utilizando datos ya etiquetados. Por ejemplo, se podría entrenar un modelo para clasificar fotos de números dándole fotos de números e indicándole que número hay en cada foto.
- Aprendizaje no supervisado: Los modelos se encargan de buscar patrones entre gran cantidad de datos. Estos modelos podrán agrupar los datos en clústeres o encontrar relaciones entre las variables. Por ejemplo, se puede utilizar este modelo para segmentar clientes en diferentes grupos basados en sus preferencias de compra.
- Aprendizaje por refuerzo: Los modelos aprenden a través del entorno donde se encuentran y depende de su acción reciben castigos o premios. El objetivo es encontrar la mejor cadena de acciones y conseguir el premio más alto.

Tras definir estos términos voy a presentar los modelos que van a ser introducidos en este documento. El primero de ellos es el *clustering*. Su objetivo es agrupar datos similares en grupos o clústeres, de manera que las muestras dentro de un mismo grupo sean más similares entre si que con las muestras de otro grupo. En este proceso, el algoritmo examina las características de los datos y busca similitudes y patrones que permitan agruparlos de la manera más coherente. Los clústeres pueden ser interpretados como categorías, lo que puede ayudar a la comprensión de los datos y descubrir patrones o relaciones. Existen varios algoritmos de clustering como son k-medias, k vecinos cercanos, o DBSCAN. El algoritmo k vecinos cercanos (KNN) utiliza la proximidad para clasificar un punto de interés basado en la mayoría de datos que le rodean. KNN es un algoritmo supervisado, se necesita un conjunto de datos de entrenamiento, y es basado en instancia, quiere decir que el algoritmo no aprende un modelo, pero memoriza las instancias de entrenamiento para realizar la predicción.

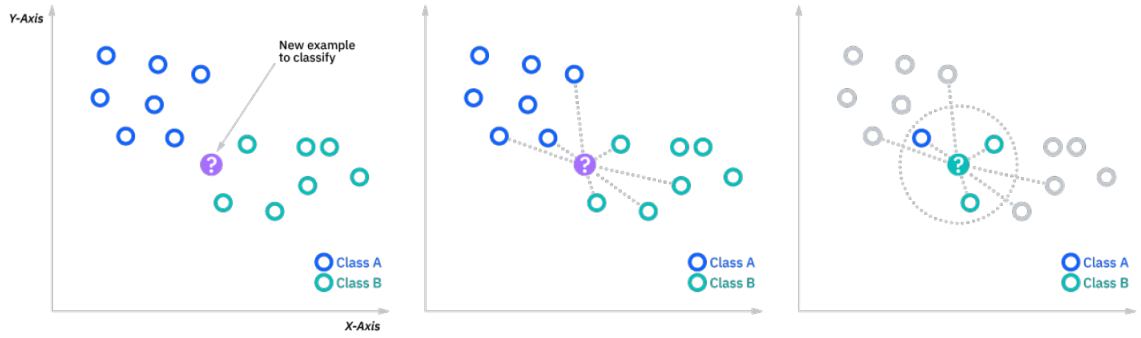


Figura 4.1: Procedimiento del algoritmo KNN
Tomada de Cloud2data

La manera que tiene KNN de funcionar es la siguiente: primero calcula la distancia entre la muestra a clasificar y el resto de items del dataset, luego selecciona los k elementos más cercanos y finalmente se realiza una "votación" entre los k vecinos más cercanos para determinar la etiqueta de la muestra a clasificar, la etiqueta más común entre los vecinos se asigna como la etiqueta de la muestra. La figura §4.1 hace una representación de cómo sería el proceso con $k = 3$. Se pueden utilizar diferentes medidas para calcular la distancia entre puntos, las más habituales son:

- Distancia euclídea: se usa como medida de distancia entre dos puntos en un espacio euclídeo. Es una medida comúnmente utilizada en análisis de datos para calcular la similitud entre puntos en un espacio multidimensional. Para calcular la distancia se usa la fórmula:

$$d(A,B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + \ldots + (z_n - z_1)^2} \tag{4.1}$$

  siendo $A = (x_1, x_2, \ldots, x_n)$ y $B = (y_1, y_2, \ldots, y_n)$.

- Distancia Manhattan: Representa la distancia "rectilínea" entre los puntos en un plano o en un espacio de mayor dimensión. A diferencia de la distancia euclídea, la distancia Manhattan solo considera desplazamientos horizontales y verticales, sin tomar en cuenta la diagonal.

$$\text{Manhattan}(A,B) = \sum_{i=1}^{n} |x_i - y_i| \tag{4.2}$$

- Distancia Minkowski: Es una generalización de la distancia Manhattan y la distancia euclídea. Permite ajustar la sensibilidad de la distancia a través del parámetro p, es decir, un valor de p mayor que 2 dará más importancia a las grandes diferencias en las coordenadas, mientras que un valor de p menor que 2 dará más importancia a las diferencias más pequeñas

$$\text{Minkowski}(A,B) = \left(|x_1 - y_1|^p + |x_2 - y_2|^p + \cdots + |x_n - y_n|^p\right)^{1/p} \tag{4.3}$$

Otro modelo a tener en cuenta es la *máquina de vectores de soporte* (SVM). Este modelo está enmarcado bajo una visión de aprendizaje automático supervisado y es utilizado principalmente para tareas de clasificación y regresión. Su objetivo principal es encontrar un hiperplano que separe las muestras de la manera más óptima. Un hiperplano es un subespacio de una dimensión menor que su espacio ambiente. Una máquina de vectores de soporte toma en cuenta el conjunto de datos del entrenamiento que es etiquetado y genera un hiperplano (que en dos dimensiones sería una línea) que separa las etiquetas, es decir, todo lo que se encuentre en una parte del hiperplano es una categoría y lo que se encuentre en la otra parte será clasificado con otra categoría. En la figura §4.2 se puede ver claramente esta separación.

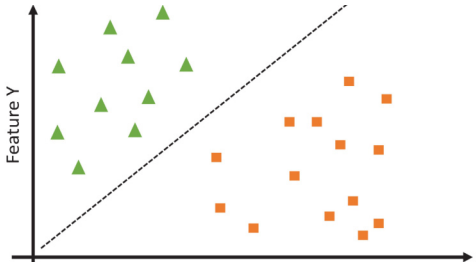


Figura 4.2: Representación de SVM. La línea representa el hiperplano, los triángulos una etiqueta y los cuadrados otra.

Tomada de Shutterstock

Para elegir el hiperplano más óptimo en el caso de datos linealmente separables, se busca el hiperplano que maximice la distancia entre los vectores de soporte de diferentes clases (margen máximo).

En ocasiones, existen casos en los que es imposible encontrar un plano que pueda separar dos clases. A esto se le conoce como falta de linealidad para dividir las clases. No obstante, existe una técnica llamada "truco del kernel". En la figura §4.3 se ve la aplicación de esta técnica. Este truco se basa en la idea de crear una dimensión adicional en la que si que podamos encontrar un plano de separación para las clases. Se utiliza una función especial llamada "función del kernel" que mapea los datos a esta dimensión adicional. A través de esta transformación, podemos encontrar una superficie de decisión que separa claramente las dos clases.

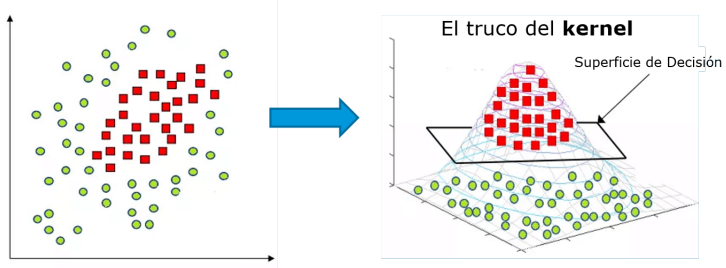


Figura 4.3: Aplicación del "truco del kernel".
Tomada de IArtificial

Las SVM tienen varias ventajas, como la capacidad de manejar conjuntos de datos de alta dimensionalidad, su efectividad en conjuntos de datos pequeños y la capacidad de manejar datos no linealmente separables utilizando funciones de kernel. Sin embargo, pueden ser sensibles a la selección de parámetros y pueden requerir tiempo computacionalmente costoso para conjuntos de datos grandes.

Para finalizar, el último modelo que presento son las *redes neuronales artificiales* que se inspira en el funcionamiento del cerebro. Es una estructura compuesta por un conjunto de unidades interconectadas llamadas neuronas artificiales o nodos, que trabajan en conjunto para realizar tareas de procesamiento de información. Cada neurona posee una o varias entradas, a las que se le asigna un peso numérico. Estas entradas se combinan linealmente para aplicarle una función de activación y producir su salida. La combinación es de la forma $z = \sum_{i=1}^{n} w_i \cdot x_i + b$ donde w_i es el peso i, x_i la entrada i y b una bias. Esta salida puede ser la entrada para otras neuronas en la red, formando así una red de conexiones. Se puede apreciar el proceso en la figura §4.4.

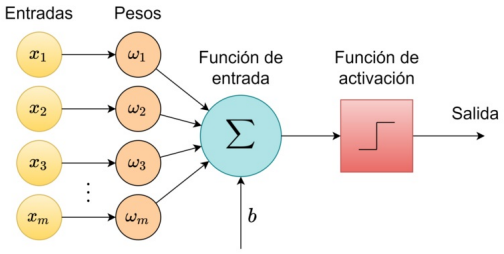
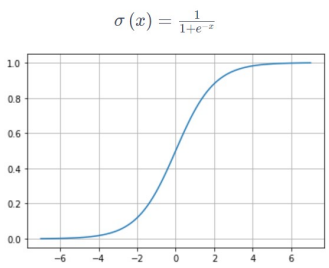


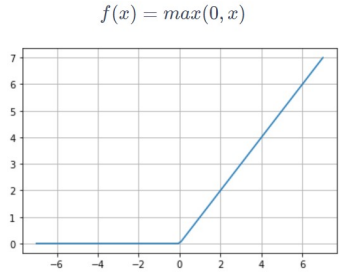
Figura 4.4: Esquema de una neurona artificial
Tomada de The Code-it List

Una función de activación es una función matemática que permite la representación de funciones complejas. Hay diversos tipos de funciones de activación, pero los más usados son (en la figura §4.5 se pueden ver varias):

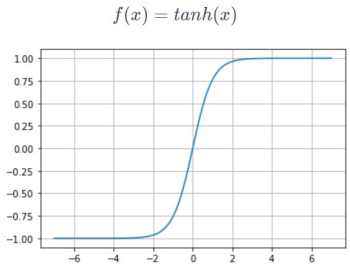
- Función Sigmoide: mapea los valores de entrada entre 0 y 1 y comúnmente se utiliza en problemas de clasificación binaria.
- Función ReLU: transforma los valores introducidos anulando los negativos y dejando los positivos tal y como entran. Obtiene una mejora de rendimiento en comparación con la función sigmoide.
- Función Tangente Hiperbólica (tanh): es similar a la función sigmoide pero mapea los valores de entrada a un rango entre -1 y 1. Al igual que la función sigmoide, es útil en problemas de clasificación binaria.
- Función Softmax: se utiliza en las capas de salida de la red para realizar una clasificación multiclase. Transforma vectores de entrada en vectores de probabilidad que suman uno, asignando así una probabilidad a cada clase.



(a) Representación de la función sigmoide



(b) Representación de la función ReLU



(c) Representación de la función tanh

Figura 4.5: Representaciones de las funciones de activación

La estructura de estas redes se organiza en capas. La capa de entrada recibe los datos de entrada y las capas intermedias, si las hay, procesan la información de manera sucesiva. Finalmente la capa de salida genera la respuesta o predicción deseada según el tipo de tarea que la red esté diseñada para realizar.

La fase de entrenamiento de una red neuronal consiste en el ajuste de pesos de las conexiones entre las neuronas para que la red pueda aprender a realizar tareas específicas. Esto se logra mediante algoritmos de aprendizaje, como el descenso del gradiente, que buscan minimizar una función de pérdida o error que cuantifica la discrepancia entre las salidas predichas por la red y las salidas deseadas. Debido a su gran capacidad de aprendizaje y adaptación a los datos, las redes pueden ser utilizadas en una amplia variedad de aplicaciones como, reconocimiento de imágenes digitales, análisis de datos, predicciones...

Un tipo de red neuronal artificial es la *red convolucional* (CNN), especializada para tratar datos estructurados en forma de matriz, como las imágenes [2]. Esta red procesa sus capas imitando al córtex visual del ojo para identificar características en imágenes y poder clasificarlas. Para ello, la CNN contiene varias capas ocultas especializadas y una jerarquía. Las primeras capas detectan líneas, curvas... hasta llegar a la capa más profunda que puede reconocer formas más complejas como un animal.

Para comenzar, la red toma como entrada los píxeles de una imagen, es decir, si el input consiste en imágenes en escala de grises de 28×28 píxeles de alto y ancho, la primera capa de la red va a contener 784 neuronas. Si tuviésemos una imagen a color con 3 canales (rojo, azul y verde), la primera capa contendría el triple ($3 \times 28 \times 28$) de neuronas. Internamente, la red va a realizar operaciones llamadas "convoluciones". Estas consisten en tomar "grupos de píxeles cercanos" e ir operando matemáticamente con otra matriz que llamaremos kernel. Este kernel recorrerá cada píxel de la imagen y generará una nueva matriz de salida, que será una nueva capa de neuronas. Realmente no solo se aplica un kernel en cada capa, sino que se aplican varias, dando resultado a varias salidas. Es decir, si aplicamos 10 kernels, obtendremos 10 salidas. Al conjunto de esta salidas se le conoce como *mapeo de características* (feature mapping) y cada salida será del mismo tamaño de la entrada. Tras esta convolución, se aplica una función de activación como se observa en la figura §4.6.

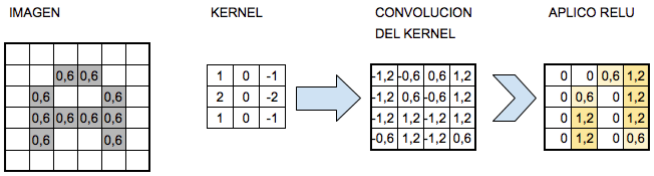


Figura 4.6: Ejemplo de convolución y activación
Tomada de Aprende Machine Learning

Tras toda la convolución, viene el momento de reducir la cantidad de neuronas. Como he dicho anteriormente, al aplicar 10 kernels a una imagen, se obtienen 10 salidas. Por lo tanto, si tenemos una imagen $28 \times 28 \times 1$, tendremos diez veces más que realmente son 10 mapas de características 28x28. De esta forma, para no aumentar el tamaño de la muestra en la siguiente convolución se realiza un proceso llamado "sub-sampling". Este consiste en reducir el tamaño de imágenes computadas con el kernel pero intentando mantener las características más importantes. Un tipo de este proceso se llama "max-pooling" que consiste en pasar una matriz por cada salida obtenida y seleccionar el máximo valor. Por ejemplo, si tenemos una matriz 28×28 y pasamos un "max-pooling" de 2×2 la matriz original, se verá reducida a la mitad, es decir a una 14×14, reduciendo drásticamente el número de neuronas. Para entenderlo de mejor forma, se puede ver el proceso en la figura §4.7.

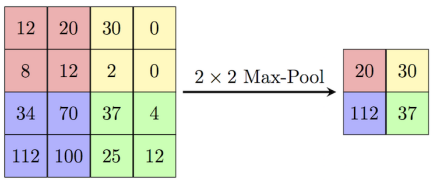


Figura 4.7: Ejemplo de *Max-pooling*.
Tomada de Computer Science Wiki

Esto solo ha sido la primera convolución, pero podremos añadir cuantas se quieran. A medida que hagamos más capas, los mapas de características serán capaces de de reconocer formas más complejas. Al terminar todas las convoluciones, podemos conectar la última salida a una red neuronal clásica que tomará estas características y clasificarás en función de ellas.

4.2.1 Entrenamiento de un modelo

Para que estos modelos funcionen y puedan producir salidas es necesario que realicen un proceso de "aprendizaje". Por ello experimentan una *fase de entrenamiento*, en

la fase el modelo aprende a partir de unos datos de entrada que llamaremos *conjunto de entrenamiento* y este junto con las funciones del modelo van a ajustar los parámetros internos del mismo, para poder hacer predicciones o tomar decisiones precisas en función de entradas futuras. Estos datos no van a ser introducidos al modelo de uno en uno, sino en un lote de datos elegido por el desarrollador llamado *batch*. Cada modelo posee una *tasa de aprendizaje* que es un hiperparámetro que determina la magnitud con los que se realizan los ajustes de parámetros durante un entrenamiento, es decir, controla que tan rápido o lento se actualizan los pesos en del modelo. Una tasa alta permite actualizaciones más grandes en cada iteración. Esto puede llevar a obtener convergencias más rápidamente, pero también puede hacer que el modelo se estanque en mínimos locales. En la práctica, es común realizar ajustes del mismo durante el entrenamiento. Asimismo, hacen uso de una *función de pérdida* técnica para cuantificar las discrepancias entre las predicciones de un modelo y los valores reales de los datos de entrenamiento de un modelo. Al calcular esta pérdida, el modelo puede ajustar sus parámetros internos de manera que minimice esas discrepancias. Existen diferentes tipos de funciones de pérdida que se utilizan según el tipo de problema de aprendizaje automático. Algunas de las funciones de pérdida comunes incluyen:

- Error cuadrático medio (MSE): mide el promedio de los cuadrados de las diferencias entre las predicciones y los valores reales.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.4}$$

  En esta fórmula, y_i representa el valor real de la observación i, mientras que $\hat{y}_i$ es la predicción realizada por el modelo para esa misma observación. La suma se realiza para todas las n observaciones y luego se divide por n para obtener el promedio.

- Entropía cruzada (cross-entropy): se usa comúnmente en problemas de clasificación y mide la discrepancia entre la distribución probabilística predicha por el modelo y la real.

$$\text{CrossEntropy} = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)) \tag{4.5}$$

  En esta fórmula, y_i representa el valor real de la clase para la observación i, mientras que $\hat{y}_i$ es la predicción de probabilidad realizada por el modelo para la clase positiva. La suma se realiza para todas las n observaciones y luego se divide por n para obtener el promedio.

4.3 MÉTRICAS

Para tener una evaluación rigurosa es necesario tener en cuenta métricas para cuantificar el desempeño de nuestro modelo. En este documento se van a nombrar varias como el *tiempo de ejecución* que se refiere a la cantidad de tiempo que tarda un algoritmo en completar su ejecución, es decir, el tiempo necesario para realizar todas las operaciones. *La confianza de un modelo* representa una medida de cuánto se puede confiar en las predicciones de un modelo. Se puede expresar a través de probabilidades. Asimismo, *la precisión de un modelo* indica la proporción de predicciones correctas que el modelo obtuvo sobre el total de predicciones realizadas. Es una medida de exactitud del modelo y se mide con la siguiente ecuación:

$$\text{Precisión} = \frac{\text{Número de predicciones correctas}}{\text{Total de predicciones}} \times 100\,\% \tag{4.6}$$

Esta última será la que más importancia va a tener a la hora de realizar pruebas y llegar a conclusiones. Finalmente cada algoritmo tiene un *coste computacional* definido como la cantidad de recursos computacionales que se requieren al ejecutar un algoritmo como tiempo o memoria. Generalmente, el coste computacional mide la cantidad de operaciones o pasos que se necesitan para realizar una tarea. Este análisis es fundamental porque permite evaluar la eficiencia y escalabilidad de los algoritmos y seleccionar la mejor solución.

4.4 OTROS CONCEPTOS

Para finalizar voy a presentar el resto de conceptos que no tienen una clasificación en concreto y van a ir apareciendo a lo largo del trabajo.

El primero es *gating modules* que son elementos utilizados en modelos de aprendizaje automático para controlar el flujo de información dentro de la red. Esto es útil en modelos que tienen conexiones recurrentes o necesitan adaptar la red a diferentes entradas o contexto. Para abrir o cerrar estas compuertas, se puede usar *binarización* que es el proceso de convertir datos en una representación binaria, es decir, transformalos a 0 o 1. Este proceso generalmente sigue un umbral o regla predeterminado. Si la variable supera el umbral, se le asigna el valor 1, y en caso contrario, el 0. *La función rectangular* es una función matemática que toma valor 1 en un intervalo específico y en el resto 0.

Estudio de trabajos relacionados

Es sabido que las técnicas de aprendizaje automático están muy presentes en la actualidad, la mayoría de estos modelos son entrenados con grandes cantidades de datos haciendo que lleguen a ser lentos y contaminantes.

Debido a la gran huella de carbono producida por estos modelos, se ha visto un creciente interés en la creación de una inteligencia artificial más sostenible. La parte hardware ha tenido más peso en el estudio, realizado por la comunidad científica, para lograr este objetivo dejando la parte software no tan desarrollada. En este capítulo, se hace un estudio desde diferentes perspectivas y ámbitos sobre cómo se trabaja en la actualidad para conseguir esta meta.

Todas las organizaciones, investigaciones y empresas que van a ser citadas en este apartado han sido clasificadas por diferentes criterios y se puede ver una taxonomía tanto de los artículos como de las organizaciones.

5.1 Organizaciones

Debido al auge del *green software* estos últimos años, son varias las organizaciones que trabajan para conseguir un software más sostenible; algunas de ellas no solo se quedan en la perspectiva del software, sino que buscan también soluciones hardware. Además de las fundaciones, existen varios centros de investigación concienciados con este problema en los que se desarrollan proyectos y estudios para solventar esta cuestión.

En primer lugar, Software Foundation (GSF) [1] busca la manera de crear una comunidad de personas, estándares, herramientas, y buenas prácticas para el desarrollo de un software sostenible. Quieren cambiar la cultura de desarrollo de software en la industria de la tecnología, para que una de las prioridades sea la sostenibilidad en los equipos de software. La misión de GSF es reducir el total de emisiones de carbono asociadas con el software defendiendo la reducción (disminución de emisiones) ante la

[1] https://greensoftware.foundation/

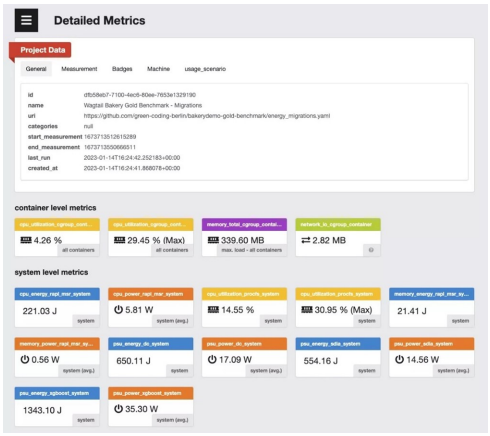
neutralización (compensación de emisiones). Es una organización sin ánimo de lucro formada bajo Linux Foundation, que actualmente cuenta con 42 organizaciones miembro como pueden ser NTT data o Avanade y 970 miembros individuales. Todos ellos participan en diversos proyectos para promover su misión.

Algunas herramientas que la fundación comparte son, principalmente, el cálculo de la huella de carbono que puede producir nuestro software en diferentes contextos como puede ser el coste de carbono en Machine Learning, en páginas webs, en nubes o incluso en tu propio sistema. También ofrece artículos, libros e investigaciones sobre como hacer un código más sostenible. Finalmente, esta organización lleva a cabo diversos proyectos para logar su misión. El más destacable es el llamado "Awesome Green Software", el cual es un conjunto con todas las herramientas, prácticas, librerías... desarrolladas por el equipo de trabajo. Otro de su proyecto es el "Software Carbon Intensity (SCI) Specification", que describe cómo calcular la intensidad de carbono de una aplicación software y define unos criterios de selección sobre herramientas, arquitectura, enfoques y servicios para usar en un futuro. Su propósito es ayudar a usuarios y desarrolladores a alcanzar un buen resultado en la reducción de emisión de carbono. "Green Software Patterns" es otro proyecto realizado por ellos. Es una base de datos de código abierto en la que encontraremos guías para hacer nuestro software más sostenible. Dispone de un catálogo dividido en tres categorías principales: inteligencia artificial, Cloud y Web. Sin embargo, en el apartado de optimizar el tamaño en modelos de aprendizaje automático no se encuentra ninguna propuesta. Para finalizar dispone de otros proyectos como "Carbon Aware SDK" que propone ejecutar tu programa en diferentes partes del mundo y horarios para hacerlo más sostenible y otros como "SCI Open Data" o "State of Green Software" que aún no están publicados.

Siguiendo una línea similar a la anterior, Green Code Berlin [2] tiene como objetivo hacer visible el consumo de energía y la emisión de carbono producida por el uso del software en general. Por ello, crean unas herramientas de código abierto para medir la huella de carbono durante el ciclo de vida de la aplicación siendo este ciclo la "vida útil" completa de una aplicación software desde el momento en el que se concibe hasta el su final. Normalmente este ciclo sigue las fases: Planificación, Desarrollo, Test, Lanzamiento y Monitorización. Está dirigida para usuarios que usan aplicaciones en ordenador, web y móvil. Uno de los proyectos que lleva a cabo es "Green Metrics Tool", que es una herramienta gratis de código abierto diseñada para facilitar a desarrolladores la medición del consumo de energía/CO^2 de arquitecturas software. En la Figura §5.1 podemos ver cómo es la interfaz de esta herramienta. Green Code Berlin

[2] https://www.green-coding.berlin/

trabaja en tres proyectos más: (1) "ECO CI" donde crean pequeñas herramientas para hacer un canal de integración continua más transparente en términos de uso de energía y aparte que puedan ahorrar energía. Actualmente este proyecto está enfocado en GitHub, ya que, es uno de las mayores plataformas de integración continua. (2) "OpenEnergyBadge" es un proyecto que propone la creación y uso de una nueva insignia en el repositorio de GitHub y que va a informar sobre el coste energético que va a producir el proyecto. Finalmente, (3) en el proyecto "Cloud Energy" se propone un modelo de aprendizaje automático estimado, basado en datos recolectados de SPECpower[3] para conseguir métricas sobre el consumo de energía directamente en la nube, debido a que normalmente en la nube no es posible acceder a estos indicadores. El modelo permite la medición en vatios o en julios del coste, además está construido con variables dinámicas para funcionar en diferentes entornos de la nube en los que puede que no esté disponible cierta información.



(a) Demo green metric tool-1

(b) Demo Green metric tool-2

Figura 5.1: Muestra de Green Metrics Tool para el usuario.
Tomada de Green Code Berlin

"The Energy-Efficient Multimedia Systems Group"[4] es un grupo de investigación que se enfoca en el desarrollo e implementación de sistemas energéticamente eficientes y de alto rendimiento para aplicaciones multimedia como aprendizaje automático, compresión de vídeos e imágenes... Trabajan en el diseño de algoritmos, arquitecturas, circuitos y sistemas para buscar una compensación entre potencia, velocidad y calidad del resultado. Tienen como objetivo reducir drásticamente el consumo de energía de

[3] https://www.spec.org/power
[4] https://www.rle.mit.edu/eems/

algoritmos computacionalmente complejos. Desarrollan métodos computacionalmente eficientes investigando tanto en hardware como en software.

Un proyecto destacable es Eyeriss [8] que busca el co-diseño entre hardware y software para la ejecución energéticamente eficiente de algoritmos de aprendizaje profundo (Deep Learning Algorithms), haciendo una arquitectura que explota la reutilización de los datos. Esto es evaluado en un test chip real.

"The Green Software Engineering Work Group" [5] es un grupo de investigación de la universidad de Birkenfeld, Alemania. Desde 2008 trabajan en proyectos de investigación en el área de software sostenible. Una de las investigaciones que actualmente lleva a cabo este grupo es "Green Coding", en el que buscan hacer un enfoque integral, que considera todos los elementos, sobre las interrelaciones entre los aspectos ambientales, económicos, técnicos de las tecnologías de la información y comunicación. Además, se dirigen a desarrolladores de software, empresas y estudiantes de ciencias de la computación. Sus metas centrales se enfocan en analizar las actuales prácticas de código verde y entrevistar expertos e interesados para determinar qué conceptos pueden ser llevados a cabo, encuestas online para analizar qué terminos de la ingeniera del software son amigables con el medio ambiente y que ya existen en la industria y, finalmente, identificar de qué manera se pueden incluir estos conceptos de código sostenible a grados universitarios.

La universidad de Radboud [6] en Holanda, investiga sobre como hacer más *verde* la tecnología de la información buscando nuevas maneras de almacenar y procesar datos. También investigan sobre soluciones software que contribuyan a los objetivos de un desarrollo sostenible. Sin embargo, no nos presenta ninguna investigación sobre como hacer un código más verde.

Por último, "Green Software Lab" [7] es un grupo de profesores, PostDocs, PhD y MSc investigadores informáticos de varias universidades de Portugal como la de Coímbra o la de Oporto. Trabajan en la reducción del consumo de energía en varios sistemas informáticos como bases de datos, móviles... Se enfocan más en la parte software donde su misión es aplicar técnicas de análisis y transformación para detectar anomalías en el consumo de energía e implementar optimizaciones para reducir dicho consumo.

[5] https://www.umwelt-campus.de/en/green-software-engineering
[6] https://www.ru.nl/en/about-us/organisation/faculties/science
[7] https://greenlab.di.uminho.pt/

Conclusiones

Tras hablar de algunas de las organizaciones que trabajan para conseguir un software más sostenible, y reducir la huella de carbono de este, se observa que la mayoría busca la manera de medir este consumo, pero faltan propuestas metodológicas a nivel de código para poder conseguirlo. Además, los que sí realizan propuestas, suelen ser a nivel de hardware.

5.2 EMPRESAS

El sector de la empresa también es uno de los grandes interesados en el software sostenible, y buscan la manera de tener un aprendizaje automático más rápido y con datos de mayor calidad frente a gran cantidad de datos.

Una de las empresas que trabaja sobre la calidad del dato es Nextlytics[8]. Esta trabaja para crear soluciones de bussiness intelligence y aprendizaje automático para lograr una transformación exitosa en empresas digitales. En una de sus publicaciones, apuntan que el funcionamiento de los modelos es un gran impulsor de una mayor sostenibilidad ambiental. Por ello, es importante trabajar la calidad del dato y luchar contra el "Data Hungry". No obstante, solo hacen alusión a estas prácticas.

Otro ejemplo parecido lo encuentro con Niu Solutions [9]. Señala que la calidad de los datos tiene prioridad, ya que, un conjunto de datos representativo es inseparable de un entrenamiento exitoso, obteniendo un modelo de alto rendimiento.

Green Software Design[10] es una consultora que trae sostenibilidad a las tecnologías de la información del cliente. La organización integra aspectos de software de diseño sostenible: alcance, especificación, arquitectura, diseño e implementación. También toma en cuenta la mantenibilidad, la reducción de emisiones.

Syniti[11] es una empresa especializada en el tratamiento de los datos, desde la calidad del dato hasta su replicación pasando por migración de datos, enlace de registros... Syniti nos presenta The Syniti Knowledge Platform (SKP), desarrollada por dicha empresa, es una plataforma para manejar las necesidades de optimización de datos, transformación y gobernanza desde una única aplicación.

[8] https://www.nextlytics.com/
[9] https://niu.solutions/
[10] https://www.greensoftwaredesign.com/en/
[11] https://www.syniti.com/

Por otra parte, Decile [12], "Data Efficient Machine Learnig" es una empresa que intenta resolver la naturaleza "Data-Hungry" de los modelos de aprendizaje planteándose preguntas como si es posible entrenar modelos con una sola muestra de entre el 5% y el 10% de los datos totales sin que esto influya de manera significativa en la precisión ahorrando así tiempo y coste. Según ellos, las necesidades actuales que quieren combatir son: los grandes costes del entrenamiento de modelos en aprendizaje profundo, el coste de etiquetar grandes conjuntos de datos, el ruido y desequilibrio en conjuntos y finalmente el consumo humano y la sobrecarga de datos. Por consiguiente, para lograr lo anteriormente mencionado realizan varias investigaciones como DISTIL [13] (Deep Diversified Interactive Learning) que es una librería construida sobre PyTorch [13] que implementa una serie de estrategias de aprendizaje activo con un enfoque particular al aprendizaje profundo. DISTIL elimina la iteratividad del aprendizaje activo dándole al usuario flexibilidad para controlar el procedimiento de entrenamiento. Exhibe una gran variedad de técnicas y ofrece soporte para incorporar este aprendizaje con datos personalizados y experimentar con datos ya conocidos. Como resultado obtuvieron un entrenamiento entre 3 y 5 veces más rápido. Otra investigación es SUBMODLIB [14] (Submodular Optimization Library), una librería para optimización submodular, en resumen, encuentra su aplicación en la selección de subconjuntos, ajuste de hiperparámetros, entrenamiento eficiente... a través de una generosa API que ofrece gran flexibilidad a la hora de usarla. Por último, uno de sus proyectos más recientes es CORDS [15] (Coreset and data selection library), que tiene como objetivo principal seleccionar los subconjuntos de datos representativos adecuados a partir de conjuntos de datos masivos, todo esto desde la visión de técnicas coreset. Las técnicas coreset son herramientas utilizadas para reducir el tamaño de conjuntos de datos de manera eficiente sin perder información crítica o características importantes. CORDS aplica estos algoritmos de selección para lograr un aprendizaje supervisado y semisupervisado eficiente.

Seguidamente se introduce una taxonomía realizada por mi para las organizaciones y empresas descritas anteriormente que analiza trabajos relacionados. Voy a clasificarlos en si realizan mediciones y recomendaciones o presentan investigaciones.

[12] https://decile.org

[13] https://github.com/decile-team/distil

[14]

[15] https://github.com/decile-team/cords

Mediciones y recomendaciones					
-	Año	Activo	Localización	Implementación	Destacado
Green Software Foundation	2021	Si	Mundial	Software	medidor de carbono
Green Code Berlin	2022	Si	Alemania	Software	medidor de carbono
The Green Software Engineering Work Group	2008	Si	Alemania	Software	Medición y Evaluación de eficiencia energética en algoritmos
Universidad Libre de Amsterdam (Research unit)	2014	Si	Holanda	Software	enfoque sobre como incluir estratégias verdes en procesos de diseño de servicios

Tabla 5.1: Organizaciones dedicadas a *Green IT*.

Investigación más allá de métricas					
-	Año	Activo	Localización	Implementación	Destacado
The Energy-Efficient Multimedia Systems Group	2011	Si	EEUU	Ambos	Procesamiento Eficiente de DNN
Decile	2008	Si	India	Software	Librerías para la reducción de datos o creación de subconjuntos. Su proyecto más destacable es CORDS
La universidad de Radboud	-	Si	Holanda	Ambos	Investigación sobre como almacenar y procesar datos
Green Software Lab	2014	No	Portugal	Software	Varias investigaciones sobre como reducir el consumo de energía

Tabla 5.2: Organizaciones dedicadas a software

Conclusión

Tras ver algunos ejemplos de empresas que buscan reducir la huella de carbono o tratar con datos, observo una tendencia en la que las empresas buscan conseguir tener un software más verde o hacer frente a grandes cantidades de datos. No obstante, ninguna llega al desarrollo de estas técnicas aunque sí existen grupos como Decile que sí llegan a desarrollarlas pero desde un punto de vista de la eficiencia.

5.3 Centros de datos

Otra perspectiva desde la que se combaten las emisiones de carbono es desde el mantenimiento y uso de las infraestructuras hardware, sobre todo en centros de datos. Segun ICREA [16] estas salas llegan a emitir el 2% de la huella de carbono a nivel mundial y desperdician alrededor del 90% de su energía eléctrica. Es necesario el uso de técnicas sostenibles para reducir su impacto. Cada día, estas infraestructuras persiguen la manera de solucionarlo y en Europa se llegó al pacto verde europeo por el cual los centros de datos se comprometen a ser climáticamente neutrales para 2030.

Leafdal Mine Datacenter[17] se presenta como uno de los centros de datos más sostenibles del mundo. Gracias a su ubicación en Noruega, puede producir grandes cantidades de energía verde y a bajo coste. Esto se debe a que Noruega está situado como número uno en energía segura, accesible y diversificada. Asimismo, la pérdida de red eléctrica en este centro es mínima. La red nacional de Noruega que está conectada a varias centrales hidroeléctricas, puede proveer la energía suficiente para cualquier futuro imprevisible. Mas aún, la fuente de alimentación actual tiene una fiabilidad del 99,97%. El sistema de enfriamiento que usa este centro utiliza agua de mar fría como fuente de refrigeración a través de un intercambiador de calor que usa agua dulce para enfriar los ordenadores. Su proximidad con un fiordo garantiza el acceso a agua de mar fría durante todo el año. Este agua hará que el agua dulce que enfría los equipos pase de 30 grados Celsius a 18 grados Celsius. Llevan 10 años funcionando así. Gracias a este método, se consigue que la instalación funcione con un PUE (Power usage effectiveness) del 1.15, siendo 1.0 un PUE ideal. PUE es un indicador para medir la eficiencia energética de un centro de datos. Este centro de datos está diseñado para neutralizar las emisiones de carbono, ya que la energía suministrada es 100% renovable y la emisiones de CO^2 generadas por vehículos se verá reducidos por la incorporación de vehículos eléctricos. Posee el certificado NS-EN ISO 14001:2015 - Normativa medioambiental.

Otro centro a destacar es TGG o "The Green Grid" [18] que no es un centro de datos, sino una organización que trabaja para proporcionar herramientas, conocimientos técnicos y aboga por la optimización de la energía y la eficiencia de los recursos para centros de datos. Está afiliada a ITI (Information Technology Industry Council) que es una asociación que trabaja para hacer avanzar en la políticas públicas del sector tecnológico. Una de las herramientas que TGG ofrece es una calculadora del coste total de

[16] International Computer Room Experts Association
[17] https://www.lefdalmine.com/
[18] https://www.thegreengrid.org/

operaciones del sistema de refrigeración que puede ayudar a evaluar las opciones de enfriamiento y ver las ventajas y desventajas de otras soluciones.

Conclusión

Poseer centros de datos verdes ayuda a lograr una tecnología más sostenible aprovechando la energía de sus instalaciones y disminuyendo la huella de carbono. Sin embargo, todas estas prácticas se realizan a nivel de hardware.

5.4 ARTÍCULOS DE INVESTIGACIÓN

Para definir el término *Green AI* [17] varios autores se apoyaron en que es el conjunto de técnicas dentro del campo de la inteligencia artificial. Con la finalidad de producir buenos resultados minimizando el coste computacional, diferenciándose de *Red AI* que hace uso de técnicas que presentan grandes costes computacionales y, por tanto, mayor emisión de carbono. Muchas de las investigaciones abogan por la creación de métricas de eficiencia energética, para que sean aceptadas como una evaluación importante para la investigación, y así poder centrarse en la eficiencia de los modelos. Al medir el gasto producido por un modelo, se busca una comparación justa entre los diferentes modelos. De esta forma se presentan varias formas de medir la actividad.

La primera consiste en medir la emisión de carbono que es una métrica atractiva porque es lo que se quiere reducir directamente. Sin embargo, es poco práctico medir la cantidad exacta al ejecutar un modelo ya que esta depende en gran medida de la infraestructura eléctrica local. Es decir, no es útil usarla como comparación entre diferentes investigaciones que se encuentran en diferentes partes del mundo. De igual forma pasa con medir el consumo de energía eléctrica.

Otra manera de medir esto podría ser el tiempo total de ejecución de un modelo, esta es una medida natural de eficiencia ya que, en igualdad de condiciones, un modelo más rápido tiene menos coste. No obstante, esta medida está influida por factores como el hardware o si otros procesos se están ejecutando en la misma máquina. Estos factores dificultan la comparación entre diferentes modelos.

Como medida concreta, los autores proponen informar sobre el número total de operaciones (FPO) realizadas para generar un resultado. Este proporciona una estimación de la cantidad de trabajo realizada por un proceso computacional. Se calcula definiendo el coste para dos operaciones básicas *ADD* y *MUL*. A partir de estas operaciones el coste FPO puede calcularse como una función recursiva de las mismas. *FPO*

se ha usado en el pasado para cuantificar la huella energética de un modelo, pero no se ha adoptado para la inteligencia artificial. Esta es una métrica atractiva porque calcula directamente la cantidad de trabajo realizado y porque es independiente del hardware en el que se ejecuta el modelo, facilitando así la comparación entre distintos enfoques.

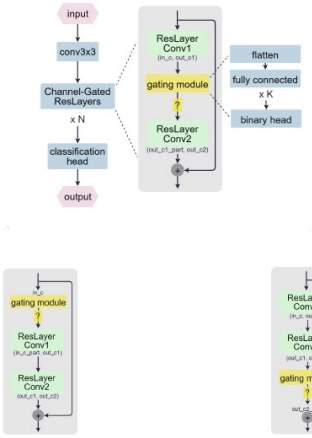
Para reducir la cantidad de datos es importante hacer un ajuste de los datos antes del entrenamiento. Este reciente artículo [16] nos muestra un estudio empírico de varias opciones de diseño en URBL (Unsupervised Reinforcement Learning Benchmark[19]) donde mostraron robustez frente a perturbaciones del entorno en el RWRL Benchmark. El enfoque que le dan se basa en tres principales hallazgos:

- Aprendizaje de un agente basado en modelos con datos recopilados mediante URL
- Configuración del modelo global pre entrenado
- Adoptar un modelo híbrido

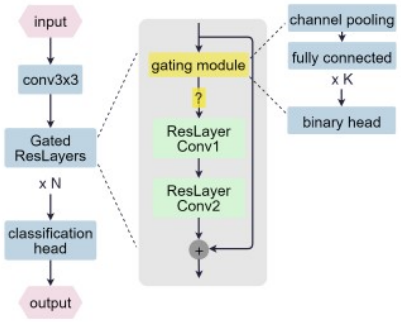
Otras investigaciones se enfocan en marcar nuevas referencias en el campo de la inteligencia artificial *verde*. Este artículo [26] propone un esquema de poda orientada a tener un modelo más sostenible, pero con un desempeño igualable al modelo sin realizar la poda. Consiste en la búsqueda de una subredes óptimas, sin embargo, en su proceso solo realiza una iteración con técnicas de poda dinámicas y realiza esta búsqueda a la vez que realiza el entrenamiento del modelo con el fin de minimizar la cantidad de *FPOs* y reducir la emisión de carbono. Este esquema usa *gating modules* diferenciables y binarizados, además hace uso de funciones de pérdida para encontrar subredes con dispersión definida por el usuario. Generalmente al hablar de podas se encuentran dos tipos de las mismas. Por un lado, la poda estática genera una subred unificada para todos los datos [20] y poda dinámica que calcula diferentes subredes para diferentes muestras de datos. La poda estática normalmente necesita una medida sobre la "importancia" de la neurona ya predefinida que servirá como umbral para saber que capas hay que podar. Por otro lado, la poda dinámica calcula la "importancia" de la neurona mientras se realiza el proceso mediante el uso de una función rectangular paremetrizable y aprendible que llevará a tener un esquema de poda diferente para cada muestra.[6] Desde una perspectiva de la inteligencia artificial ambientalista ninguna de estos dos enfoques son ideales. La poda dinámica presenta una sobrecarga de cálculo debido a la medida de la "importancia" de las neuronas mientras que la

[19]Es una técnica utilizada para medir el rendimiento de un sistema o uno de sus componentes

poda estática puede reducir esta sobrecarga, pero el proceso iterativo de poda y ajuste de parámetros consume más recursos computacionales y tiempo durante la fase de entrenamiento. Por tanto, los autores crearon un método para la poda que es capaz de realizarse para una red pequeña sin tener un coste significativo en el entrenamiento optimizando simultáneamente la estructura de la red y sus parámetros mientras se entrena. Esta optimización está efectuada con el uso de una función *gate* binarizada y entrenable, además cuenta con una regularización de polarización para fomentar una función de activación unificada de las aristas ya que la red, está definida por la forma de un grafo, que representa las conexiones entre capas $\Phi := (V, E)$ donde V es un conjunto de vértices y E es un conjunto de aristas $E := e^{(x,y)}, \forall x, y \in V$ y donde todas las aristas van a estar asociadas a un peso. Para obtener una estructura de la subred estable, también definida por un grafo, mediante una optimización por gradiente, por ejemplo con una activación binaria para cada conexión, es necesario un *módulo de compuerta* o en inglés *gating module*. El enfoque que le han dado es con el uso de *straight-through estimator (STE)* [3] que binariza la salida estocástica con un umbral basado en la siguiente capa, además copia heurísticamente el gradiente de la siguiente capa. Los autores eligieron esta *puerta* dado que es la que daba un ratio de error más bajo. Para comprobar la eficacia de este método los autores realizaron una experimentación con ResNet 110 como red. Los datasets que usaron fueron CIFAR-10 y CIFAR-100 que contienen 50,000 imágenes para el entrenamiento y 10,000 para testeo. La dimensión de estas imágenes es 32×32 y todas ellas en color y CIFAR-10 presenta 10 clases mientras que CIFAR-100 contiene 100 clases. Realizaron una poda de capas y por otra parte una poda de canales. La red ResNet-110 presenta 54 capas residuales y 2 capas convolucionales. Para la poda por capas, el *gating module* está colocado al principio de cada capa residual para decidir si se computa o no (5.2b). Por otro lado, para la poda de canal, el *gating module* puede estar colocado en 3 posiciones diferentes: una antes de la primera capa de convolución, otra entre las dos capas y finalmente después de la segunda capa convolucional. La 5.2a muestra estas 3 posiciones.



(a) posición del *module gating* en la poda por canal



(b) posición del *module gating* en la poda de capas

Figura 5.2: Posiciones del *module gating*
Tomada de [26]

Tras experimentar con los datasets, en la tabla §5.3 se aprecia la precisión, el ratio de veces que se ha abierto la compuerta y el número de *FLOPs* para el modelo sin poda y después para el modelo tras aplicar la poda por capas y seguidamente para la poda por canal tanto para el dataset CIFAR-10 como para CIFAR-100. Para las pruebas los autores usaron un *learning rate* inicial que para ambos datasets fue del 0.1 y en cada etapa posterior se disminuía un 10%.

Tabla 5.3: Resultados tras la poda

Dataset	Model	Top-1 acc (%)	Gate open RI (%)	FLOPs (M)(rel)
CIFAR-10	baseline	93.68 (0)	100.00	255.3
-	layer pruned	92.82 (-0.86)	53.70	137.7
-	channel pruned	91.01 (-2.67)	48.76	189.9
CIFAR-100	baseline	71.85 (0)	100.00	255.3
-	layer pruned	70.01 (-1.84)	66.67	171.1
-	channel pruned	66.91 (-4.94)	51.14	135.41

Los resultados arrojaron que se puede eliminar aproximadamente un 50% de las conexiones en redes profundas en los *datasets* CIFAR-10 y CIFAR-100 pero con una reducción de menos del 1% en la precisión. En comparación con otros métodos de poda relacionados, su método tiene una menor caída de precisión para reducciones equivalentes en los costes computacionales.

La mayoría de las investigaciones tienen en cuenta cómo hacer un código más energéticamente eficiente. Estas no son de gran ayuda para este proyecto, ya que, la manera en la que intentan resolver no pasa por combatir el "Data-Hungry" del aprendizaje automático, aunque existen algunos estudios que no tienen como objetivo esto pero nos pueden servir de apoyo. En el Centro Singular de Investigación en Tecnologías Inteligentes (CiTIUS) de la universidad de Santiago de Compostela, llevaron a cabo una investigación obteniendo como resultado un algoritmo de aprendizaje automático y de bajo consumo para el procesamiento de grandes datos [1]. Han desarrollado un nuevo modelo de Máquinas de Vectores de Soporte (SVM) que ha permitido superar limitaciones que este poseía, ya que, era considerablemente lento a la hora de abordar problemas en los que el número de datos era considerablemente alto. Este modelo obtiene resultados entre 10 y 100 veces más rápido.

Han llamado al nuevo modelo desarrollado como Fast Support Vector Classification, pero también han propuesto un entrenamiento eficiente inspirado en las ideas SVC, pero de manera más rápida, directa

Siguiendo un enfoque centrado en los datos, en la conferencia para la sostenibilidad TIC, varias universidades presentaron un estudio empírico sobre este enfoque en la inteligencia artificial [24]. Experimentaron basándose en varias variables como el tipo de modelo de datos, el número de datos y el número de características. Se puede observar más en la sección §6.1.

Dataset distilation

Asimismo, existen métodos de destilación de datos. Estos consisten en una transformación de grandes conjuntos de datos en conjuntos significativamente más pequeños pero de alto rendimiento. Google junto con DeepMind pusieron en práctica una novedosa aplicación de este método [12] usando solo 10 imágenes del dataset CIFAR-10 obteniendo un 65% de precisión. El enfoque que tiene la "destilación de datos" (data destillation) es sintetizar conjuntos de datos que sean más informativos que la naturaleza del conjunto inicial. Asimismo, estos conjuntos destilados no serán como las imágenes naturales pero captarán características útiles para una red neuronal que según ellos, es una capacidad con cierto misterio que no ha sido comprendida del todo. En esta investigación, ponen en práctica una extensión a gran escala de los métodos KIP (Kernel Inducing Points) y LS (Label Solve) [11] usando una función de pérdida derivada de una regresión del kernel para obtener un nuevo estado del arte. En la figura §5.3, se observan las imágenes sin modificar y las imágenes tras ser destiladas.

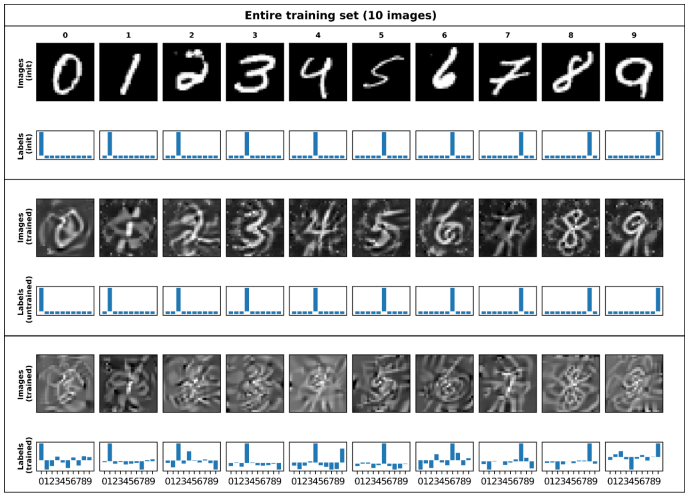


Figura 5.3: Comparación de la imagen sin destilar y la imagen ya destilada. Tomada de [12]

Para abordar la "destilación de datos", se apoyan en la teoría del límite de ancho infinito de las redes neuronales. Esta teoría se refiere a un marco teórico utilizado para analizar el comportamiento de las redes neuronales con una arquitectura de ancho infinito. En este límite, se examinan las propiedades y características asintóticas de las redes neuronales, lo que permite comprender mejor su capacidad de representación, generalización y otros aspectos fundamentales de su funcionamiento. El proceso de destilación puede ser formulado en un proceso de optimización de dos pasos: un bucle interno que entrena los modelos con datos aprendidos, y un bucle externo que optimiza los datos aprendidos para la ejecución con datos sin modificar. La teoría mencionada cambia el bucle interno de entrenamiento con una regresión simple del kernel que, añadiendo una regularización la regresión, se convierte en un problema de regresión Ridge del Kernel (KRR) lo que es bastante beneficioso porque esta regresión posee una fórmula relaciona con los datos de entrenamiento, lo que significa que su función de pérdida va a poder ser optimizada durante el bucle externo. El algoritmo KIP optimiza los datos del conjunto de entrenamiento minimizando la función de pérdida del KRR a través de métodos basados en el gradiente. Por otro lado, LS computa directamente el conjunto de etiquetas que minimicen la función de pérdida, generando así un conjunto de etiquetas único para cada imagen.

En la primera etapa, en la cual se aplica el KRR, se enfoca en redes completamente conectadas donde sus elementos de kernel son poco costosos de calcular. Sin embargo, en el segundo paso, necesitan distribuir el cálculo de los elementos del kernel y sus gradientes en diferentes dispositivos, por lo que invocan un modelo cliente-servidor. El servidor va a distribuir cargas de trabajo independientes hacia los clientes y haciendo

que el paso de propagación hacía atrás sea lo más eficiente posible. Un esquema de este modelo es que vemos en la figura §5.4.

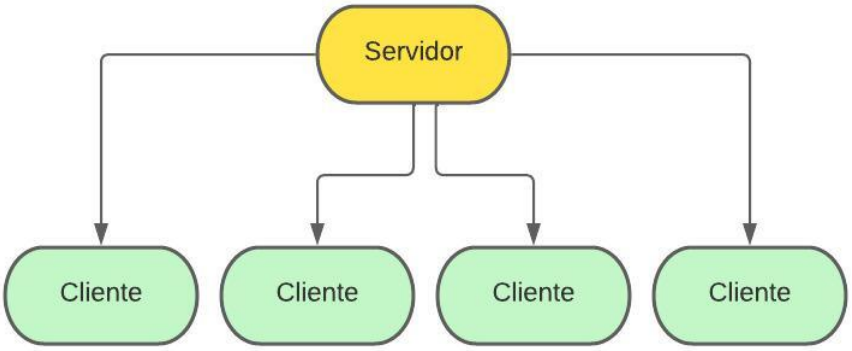


Figura 5.4: Modelo cliente servidor

Para comprobar las efectividad de estos datos, experimentaron en la conocida red "ConvNet" porque ya ha sido utilizada en otras investigaciones de destilación. Hicieron una variación de esta red y usaron una arquitectura simple que constaba de 3 bloques de convolución, una función RELU y 2x2 pooling promedio. Luego, agregaron una capa de lectura lineal final, con una capa adicional de convolución 3x3 y ReLu antepuesta. Realizaron la experimentación en cinco datasets: MNIST, Fashion-MNIST, SHVN, CIFAR-10 y CIFAR-100. Todos los resultados se pueden ver en la tabla §5.4.

Como resultado obtienen una actuación que se enmarca en el cuadro de la clasificación de imágenes. Destacando en CIFAR-10 que se consigue un 64% de precisión usando solo un una imagen por clase, es decir, un 0.02% del total del dataset y obteniendo un 80% cuando usa 50 imágenes. Para concluir se observa que con LS y KIP la optimización de etiquetas es una herramienta muy poderosa para la destilación de datos, obtienen unas etiquetas bastante alejadas de las etiquetas estándares y animan a que el estudio de optimización de etiquetas crezca.

Dataset	Img por clase	LS ConvNet	KIP ConvNet
MNIST	1	73.4	97.3
	50	96.4	99.1
	100	98.3	99.4
Fashion-MNIST	1	65.3	82.9
	50	80.8	91.0
	100	86.9	92.4
SVHN	1	23.9	62.4
	50	52.8	79.3
	100	76.8	82.0
CIFAR-10	1	26.1	64.7
	50	53.6	75.6
	100	65.9	78.2
CIFAR-100	1	23.8	34.9
	50	39.2	47.9

Tabla 5.4: Resultados obtenidos después de usar las imágenes obtenidas por la red neuronal. Los números de la columna LS y KIP corresponde con su precisión

Indicador distancia-entropía

Una forma de reducir el número de datos del conjunto de entrenamiento es usar un indicador distancia-entropía como el que proponen unos investigadores de la Universidad de Tainjin en China [10]. El propósito es distinguir los datos de calidad desde la perspectiva de la información. Este indicador es calculado basado en las relaciones entre la muestra nueva y todas las categorías.

Diferencian así muestras con gran calidad de información que se van a dar cuando este indicador sea grande y pueden proveer información que los datos ya recogidos no incluyen. Por otro lado, se tienen datos redundantes con poca calidad de información que no incluye una mejora en el rendimiento. Este se explicará más detalladamente en la sección §7.1

Selección de conjuntos de datos

Existen métodos de selección de conjuntos de datos como Core-Set selection o Active Learning. Active Learning selecciona puntos para etiquetar un gran grupo de datos sin etiquetas entrenando repetidamente un modelo usando pequeños conjuntos de datos ya etiquetados y seleccionando ejemplos adicionales para etiquetar basándose en

la incertidumbre del mismo. [4] Active learning comienza con una gran "piscina" de datos sin etiquetar. Inicialmente estos métodos etiquetan una pequeña porción de puntos elegidas uniformemente al azar [19]. Teniendo una función de pérdida, el conjunto completo sin etiquetar y las etiquetas del subconjunto aleatorio inicial, la meta es seleccionar hasta un límite b de puntos para etiquetar y que produzca un modelo con bajo error. Dos estrategias que cubren el espectro de modelos basados en la incertidumbre y la representatividad para el aprendizaje automático son "least confidence uncertainty sampling" y "greedy k-centers". Para el primero calculan la "confianza" del modelo usando:

$$f_{confidence}(x; A_k^T) = 1 - max_{\hat{y}} P(\hat{y}|x; A_k^T)$$

y seleccionan los ejemplos con menor confianza [21] y para el segundo método extraen una representación de características de la última capa oculta del modelo y calculan distancia entre los ejemplos para seleccionar puntos [18]. Finalmente el mismo modelo es entrenado con los puntos b etiquetados para obtener el modelo final, que se evalúa con un conjunto de datos no utilizados para calcular el error y cuantificar la calidad de los datos seleccionados.

Por otro lado, las técnicas de Core-Set Selection comienzan con un gran conjunto de datos etiquetado o sin etiquetar y tienen como meta encontrar un subconjunto que se aproxime en calidad de información con el conjunto de datos completo seleccionando ejemplos representativos [18]. Empiezan con un conjunto de datos ya etiquetado $L = \{x_i, y_i\}_{i \in [n]}$ con objetivo de encontrar un subconjunto de $m \leq n$ de puntos $s = \{s_j \in [n]\}_{j \in [n]}$ que logra una calidad comparable con el dataset inicial:

$$\min_{s:|s|=m} E_{x,y \sim p_z}[\ell(x, y; A_s)] - E_{x,y \sim p_z}[\ell(x, y; A_{[n]})].$$

Aquí se puede destacar tres estrategias "greedy k-centers" [18] ya descrita anteriormente. "Forgetting events" [23] que hace referencia al número de veces que un ejemplo es clasificado incorrectamente después de haber sido clasificado de manera correcta anteriormente. Aquí para seleccionar puntos, se seleccionan los que tienen el mayor número de "Forgetting events" y los que nunca fueron clasificados correctamente se les trató como que su número de "Forgetting events" es infinito. De manera similar, la tercera técnica clasifica los ejemplos basándose en la entropía de un modelo objetivo como:

$$f_{entropy}(x; A_{[n]}^T) = -\sum P(\hat{y}|x; A_{[n]}^T) \log P(\hat{y}|x; A_{[n]}^T)$$

y quedarse con los m ejemplos con mayor entropía.

Para evaluar la calidad de la estrategia, comparan el rendimiento del modelo primero entrenando con el subconjunto representativo obtenido y luego con el conjunto de datos completo

Sin embargo, estas técnicas suelen ser excesivamente costosas para aplicarlas a modelos de aprendizaje automático. Por ejemplo, muchas técnicas de Active Learning requieren alguna representación de características antes de que puedan identificar con precisión los puntos con más información. Consecuentemente, estos métodos de aprendizaje solicitan etiquetas en grandes batches para evitar reentrenar el modelo demasiadas veces, aunque para cada batch requieren entrenar un modelo completo lo que es demasiado costoso para modelos grandes. Para solventar esta cuestión de coste, los autores de [4] proponen una selección vía proxy. Para crear un modelo proxy "barato" plantean escalar hacia abajo eliminando capas, usando una arquitectura más pequeña y entrenándola con pocos epochs obteniendo sorprendentemente representaciones útiles. Lo que aplicaron a cada técnica fue lo siguiente:

- Para Active Learning se reemplazó el modelo entrenado en cada batch por el proxy, pero luego entrenando el mismo modelo una vez que el límite b se alcanzó
- Para Core-Set se usó un proxy en vez de un trained target para calcular las métricas y seleccionar s

Aplicando las dos técnicas de Active Learning con el proxy consiguieron hasta un 41,9% y 3,8% de mejora en el tiempo de ejecución, respectivamente, sin un aumento significativo del error Para Core-Set Selectión. Esta técnica puede eliminar hasta el 50% de los datos de CIFAR10 en un 10% menos de lo que se tarda en entrenar el modelo objetivo, lo que supone una mejora de 1,6% en el entrenamiento de principio a fin.

Elección de datos de fenómenos sísmicos

Para el tratamiento sísmicos se encuentra un estudio sobre la elección de datos para el entrenamiento de un modelo [9]. Utilizan dos enfoques el primero de ellos utiliza un método de clustering basado en horizontes geológicos donde se utiliza la distancia a los centroides en el clustering y no se usa ninguna restricción en la ubicación de datos seleccionados. El segundo método parte de un conjunto predefinido y, a continuación, escanea sobre el conjunto completo de datos para identificar muestras de entrenamiento adicionales para aumentar el conjunto de datos inicial.

Una investigación que estudia la respuesta de diferentes estructuras a fenómenos sísmicos que rara vez ocurren plantean un método de selección de datos para realizar el

entrenamiento [22]. En el contexto de la ingeniería de terremotos, se han llevado a cabo investigaciones para caracterizar el movimiento sísmico en términos de magnitudes, como las intensidades sísmicas. Estas medidas funcionarán como características en los parámetros de entrada en la red neuronal.

Su objetivo es estudiar los casos de terremotos menos frecuentes y para ello quieren un muestra representativa que contengan estos casos. Primero generan muestras artificiales de terremotos para después seleccionar un dataset basado en las intensidades de los sismos generados. Para ello calculan las intensidades de los terremotos generados y escogen diferentes intensidades dependiendo de lo que quieran estudiar. Realizan una distribución probabilista en base a las intensidades y escogen un tamaño de cuadrícula para dividir el dominio. A continuación se selecciona una muestra por cuadrícula obteniendo un subconjunto.

Dependiendo de cuantas intensidades sísmicas se escojan para el proceso de selección se obtienen diferentes cantidades de muestras en el conjunto de entrenamiento seleccionado. De un conjunto completo de 10,000 muestras y usando dos intensidades se consigue seleccionar un set de 438 muestras. La figura §5.5 muestra el proceso para obtener el subconjunto

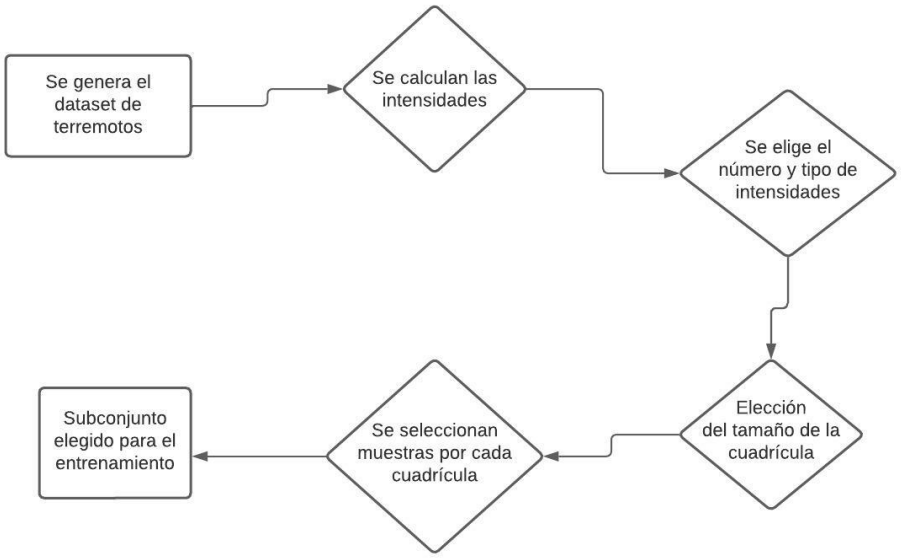


Figura 5.5: Diagrama del proceso para subconjunto de terremotos

Como resultado van a obtener un conjunto de datos con representatividad, ya que, su objetivo es también analizar los fenómenos que son poco comunes, por ello quieren un subconjunto que existan esos datos. Si en vez de utilizar este método usasen una selección aleatoria probablemente los eventos más extraños no aparecerían, ya que, hay menos.

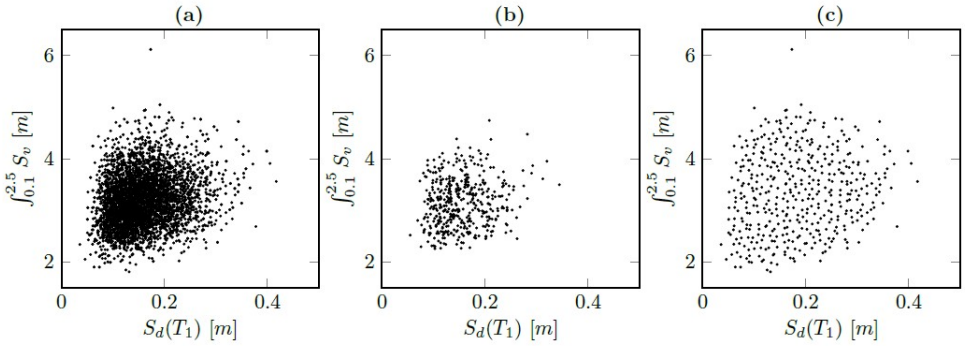


Figura 5.6: Estrategia de selección de datos.
Tomada de [22]

La figura §5.6 nos muestra tres representaciones del conjunto de datos usando dos intensidades, la primera de ellas (a) sería el conjunto de datos completo, la segunda (b) sería un subconjunto que ha usado como método de selección la aleatoriedad y por último (c) el subconjunto pero con el método usado Como se observa, el subconjunto que obtienen realizando su método presenta mas diversidad de datos e incluye sucesos poco comunes.

Este método consigue obtener un conjunto representativo que permite hacer una predicción de todo el dominio estadístico al mismo tiempo que presenta un buen nivel de eficiencia computacional.

De la misma forma hice una clasificación para estas investigaciones. Las he dividido con las que hacen una implementación para conseguir una reducción de datos, las que consiguen mejorar la eficiencia de otras formas y finalmente los artículos que dan recomendaciones o son un estudio sin implementación.

Artículos informativos					
-	Año	Publicado	Localización	Implementación	uso
Energy-efficient machine learning on the edges	2020	IEEE	EEUU	Ambos	Estudio del arte sobre técnicas de eficiencia energética en aprendizaje automático y presenta un plugin que ayuda en la optimización del aprendizaje y mide el consumo de energía
The carbon footprint of machine learning training will plateau, then shrink	2022	arxiv	EEUU	-	Aconseja y describe prácticas para reducir el consumo de energía en el entrenamiento de modelos de aprendizaje automático.
Green ai	2019	arxiv	EEUU	-	Describe el término "Green AI" y anima a investigadores a usar la eficiencia como un criterio determinante

Tabla 5.5: Tabla de artículos a *Green AI* en un marco teórico

Artículos dedicados a la eficiencia del modelo					
-	Año	Publicado	Localización	Implementación	uso
Fast support vector classification for large-scale problems	2022	IEEE	España	Software	propone una mejora de SVM[20] haciéndolo mucho más rápido y eficiente con un entramiento eficiente y creación de prototipos de clase
Unsupervised model-based pre-training for data-efficient control from pixels	2022	arxiv	Canada	Software	Diseñan una estratégia de RL no supervisado, primero estudia los modelos pre-entrenados para luego analizar varias opciones de diseño reutilizando componentes pre-entrenados obteniendo un modelo que necesita 20 veces menos datos para igualar métodos supervisados
A new baseline for green ai: Finding the optimal sub-network via layer and channel pruning	2023	arxiv	Dublín	Software	proponen un esquema de poda orientado ecologicamente. El esquema de poda consiste en un módulo ligero, diferenciable y binarizado y en nuevas funciones de pérdida para descubrir subredes con una dispersión definida por el usuario.

Tabla 5.6: Tabla de artículos dedicados a *Green IT*.

Artículos dedicados a la reducción de datos					
-	Año	Publicado	Localización	Implementación	Uso
Distance-entropy: An effective indicator for selecting informati-ve data	2022	Frontiers	China	Software	proponen un indicador entrópico-distancia para distinguir los datos con calidad de información en el ámbito de smart agriculture
Data-driven method for training data selection for deep learning	2021	EAGE	Holanda	Software	Selección de un conjuto representativo para datos sísmicos
Training data selection for machine learning-enhanced monte carlo simulations in structural dynamics	2022	Applied Science	Alemania	Software	Manera de seleccionar un conjunto de datos de entrenamiento para predicir el comportamiento de estructuras a fenómenos sísmicos poco probables
selection via proxy: efficient data selection for deep learning	2020	ICRL 2020	EEUU	Software	Mejora de la eficiencia computacional de métodos de selección de conjuntos de datos como Core set o Active Learning
Data-Centric Green AI An Exploratory Empirical Study	2022	IEEE	Europa	Software	Realiza un estudio empírico sobre como afecta al rendimiento y por tanto, a la huella de carbono usar un conjunto de datos reducido.
Dataset distillation with infinitely wide convolutional networks	2021	NeurIPS 2021	-	Software	implementa dos algoritmos (KIP y LS) en imágenes para conseguir nuevas imágenes modificadas con más información de manera que con menos muestras se obtienen buenos resultados

Tabla 5.7: Tabla de artículos dedicadas a la reducción de datos

Conclusión

Gracias a todo lo anterior, se puede interpretar que hay una tendencia tanto en investigaciones como organizaciones que se preocupa por tener un software más verde y eficiente. La mayoría de las organizaciones colabora reuniendo y recomendando una serie de prácticas para hacer un código más verde, aunque se orienta sobre todo a calcular la emisión producida por un código. De igual forma, el desarrollo de un hardware verde ha sido el que ha tenido mucho más peso a lo largo de los últimos años prevaleciendo este sobre hacer código verde, de ahí, que los centros de datos trabajen para ser neutralmente climáticos. Por otro lado, en las investigaciones llevadas a cabo hasta la fecha si bien ha habido un auge sobre el estudio en el término de green IT, más concretamente *Green AI*, muy pocas tratan de solucionar la naturaleza *Data-Hungry* de los modelos de aprendizaje automático dejándonos por delante un campo abierto a la investigación del mismo

6

Método por muestreo estratificado

Esta sección muestra uno de los métodos presentados en el apartado anterior §5. Esta técnica se basa en el artículo [24]. Primero se encuentra una descripción del método para después exponer su implementación y experimentación realizadas por mi.

6.1 Descripción

Este es un método para dividir un conjunto completo de datos en conjunto de entrenamiento y conjunto de test pero de manera que se mantenga la proporción de clases en ambas partes. Esta garantiza que cada conjunto contenga una distribución similar de las clases presentes en el conjunto de datos original. Para la realización de estas pruebas me inspiré en un trabajo que estudia un enfoque en los datos más que en el modelo para tener una inteligencia artificial más sostenible [24].

En ella realizan un estudio empírico usando 6 diferentes modelos de aprendizaje, un dataset de 5,574 muestras y dos modificaciones: número de datos y número de características motivados por descubrir el potencial impacto que se obtendría al modificar el conjunto de datos en el consumo de energía en modelos de inteligencia artificial. Para su experimentación, quisieron probar con diferentes variables. En concreto usaron 3 variables independientes que eran: (1) El tipo de algoritmo, en este caso, usaron los modelos más famosos y que se encontraban en la librería *Scikit-learn* [15] como eran: Arbol de decisión, maquina de soporte de vector... (2) La segunda variable fue el número de datos para el conjunto del entrenamiento que los seleccionaron de manera estratificada y en porcentajes de 10 %, 20 %, 30 %... (3) Finalmente la última de las variables independientes fue el número de características. Igualmente fue por porcentajes y para seleccionarlas usaron el test chi-cuadrado (Chi2). Como métricas usaron las variables dependientes estas eran el consumo de energía y la métrica *F1-score*. Tras la realización de su experimentación, obtuvieron que pueden llegar a reducir el consumo de energía hasta en un 76 % mientras se mantiene el rendimiento del modelo sobre todo cuando se hace una reducción de datos. En el caso del algoritmo *Random Forest* se reduce hasta un 92 %. Concluyeron que la precisión de los modelos podía ver-

se afectada negativamente por las estrategias, pero que, en la mayoría de los casos era insignificante demostrando que a menudo "menos es más" ya que, procesar un conjunto de datos más pequeño puede reducir drásticamente el consumo de energía sin sacrificar la precisión.

6.2 Implementación y experimentación realizadas

En mi caso, mi experimentación se va a basar en cómo afecta a la precisión de un modelo solo modificando el número de datos con un conjunto de datos de uso libre que se detallará en la siguiente sección §6.2 y midiendo el tiempo.

Para ver más clara esta separación, propongo un ejemplo práctico: imaginemos que tengo un conjunto de datos con 1000 muestras y dos clases: Clase A y Clase B la distribución es la siguiente:

- Clase A: 800 datos
- Clase B: 200 datos

Ahora se quiere dividir de manera estratificada este conjunto entero, la proporción de datos entre conjunto de entrenamiento y conjunto de prueba la elige el usuario, en esta caso va a ser del 80% para entrenamiento y 20% para test. Como debemos mantener las proporciones de las clases en ambos conjuntos nos quedaría de la siguiente forma:

- Conjunto de entrenamiento (80%):
  - Clase A: 640 muestras (80% de 800)
  - Clase B: 160 muestra (80% de 200)
- Conjunto de Prueba (20%):
  - Clase A: 160 muestras (20% de 800)
  - Clase B: 40 muestras (20% de 200)

Como resultado, tanto el conjunto de entrenamiento como el de prueba mantiene la proporción original de las clases, asegurando una representación equilibrada de todas las clases.

Para la realización de las pruebas he usado el DataSet de uso libre *fashion-mnist* [25]. Este contiene 70,000 imágenes con 10 categorías de productos de moda en una escala de grises y con dimensiones 28x28. El conjunto de entrenamiento está formado por 60,000 muestras y el de test por 10,000. El código lo he realizado con python 3.10.9 en Jupyter Notebook [7] con las librerías de *Scikit-learn* en la versión 1.2.1 [15] y *Tensorflow* con versión 2.12.0 [5]. Todo el código creado por mí, se puede encontrar en mi GitHub[1].

Las pruebas las he realizado en un equipo con procesador Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz y una memoria RAM de 8.00 GB.

Para la experimentación de este método he separado el conjunto de entrenamiento y el de test con diferentes porcentajes pero siempre haciéndolo de una manera estratificada.

He separado el conjunto de entrenamiento en 10 % del total y he entrenado diferentes modelos para ver cuanto tardaban y su precisión. Inicialmente tenía 70,000 datos, por lo tanto, con 10 % uso 7,000 datos. Así he repetido el proceso para 25 %, 30 %, 40 %, 50 % y finalmente 80 % que sería la línea base con la que normalmente se entrena. Todos los resultados se pueden apreciar en la tabla §6.1.

Algorithm	**10 %**	**25 %**	**30 %**	**40 %**	**50 %**	**80 %**
			Precisión			
SVM	0.867	0.881	0.883	0.889	0.891	0.904
KNN	0.810	0.832	0.836	0.842	0.844	0.858
DecTree	0.692	0.732	0.744	0.705	0.751	0.762
RandForest	0.833	0.849	0.851	0.855	0.858	0.868
AdaBoost	0.466	0.525	0.553	0.514	0.495	0.524
BaggingClas	0.852	0.869	0.871	0.876	0.880	0.891
			Tiempo (s)			
SVM	10.080	21.815	30.869	55.690	93.573	251.953
KNN	0.0209	0.0038	0.0155	0.0156	0.0167	0.0468
DecTree	1.254	3.445	4.802	6.294	8.763	15.948
RandForest	18.435	52.818	63.794	88.293	111.13	191.68
AdaBoost	13.425	36.699	44.324	57.983	76.208	122.93
BaggingClas	4.473	20.107	28.645	53.761	81.510	267.36

Tabla 6.1: Fashion Mnist - Estratificado

[1] https://github.com/ricardoperalta00/TFG-.git

Podemos observar que se obtiene una caída de alrededor de un 4% entre usar un 10% de los datos frente a usar un 80%. Sin embargo, se obtiene un tiempo de ejecución mucho menor. Los que peores resultados presentan son el árbol de decisión y AdaBoost. Finalmente se aprecia que con muchos menos datos, por lo tanto, menos operaciones, un modelo más rápido y "verde" es obtenido sin sacrificar el rendimiento del mismo. También quise añadir como modelo de aprendizaje una red convolucional con la arquitectura que se muestra en la tabla §6.2.

Arquitectura modelo secuencial		
Layer (type)	**Output Shape**	**Param**
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d	(None, 13, 13, 32)	0
flatten (Flatten)	(None, 5408)	0
dense (Relu)	(None, 100)	540900
dense_1 (Softmax)	(None, 10)	1010
Total params: 542,230		
Trainable params: 542,230		
Non-trainable params: 0		

Tabla 6.2: Arquitectura de la red para la experimentación del medidor

Esto también servirá para poder comparar con el método siguiente §7.1. Los resultados obtenidos son los que se muestran en la tabla §6.3. Con esta tabla aprecio las mismas conclusiones que antes.

porcentaje	**10%**	**25%**	**30%**	**40%**	**50%**	**80%**
Num muestras	6000	15000	18000	24000	30000	48000
Precisión	0.87	0.89	0.90	0.90	0.90	0.92
Tiempo (segundos)	57.01	112.87	137.33	165.90	205.81	318.88

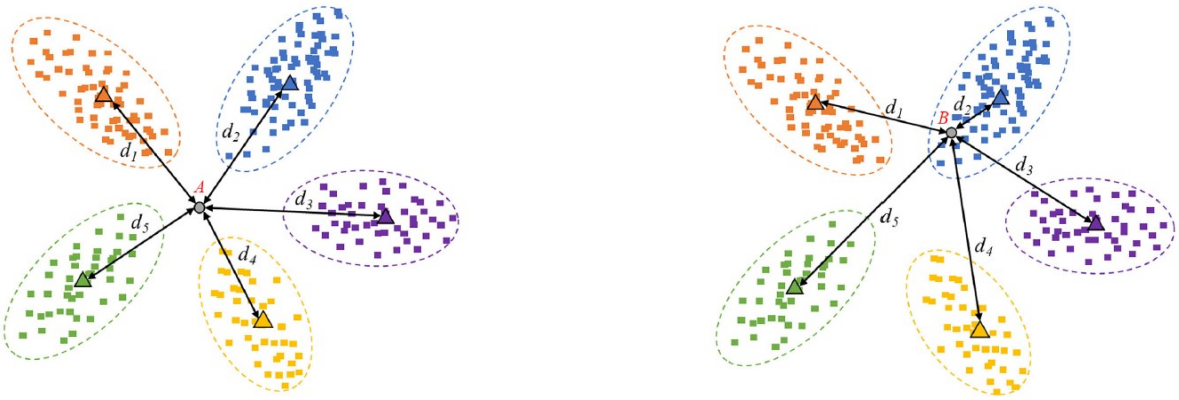
Tabla 6.3: Tabla de resultados para CNN.

Método del indicador distancia-entropía

En esta sección se estudia con más profundidad el método descrito en el artículo [10]. En primer lugar, hay una descripción de la técnica y los resultados que obtuvieron. Seguidamente la implementación y después la experimentación realizada por mi

7.1 Descripción

Este método [10] se basa en el uso de una pequeña parte de los datos del conjunto de datos total a los que voy a llamar datos base y, sobre ellos, se van tomando elementos del conjunto total que no están en la base para calcular su distancia con los puntos que existen en el conjunto de la base y su entropía. Finalmente, se añade a los datos base los elementos del conjunto total con mejor entropía, es decir, los datos que han obtenido una entropía más alta.



(a) Dato con gran calidad de información

(b) Dato con poca calidad de información

Figura 7.1: Calidades de datos
Tomada de [10]

El indicador distancia-entropía usa la distancia para medir similitudes y la entropía para evaluar la información. Para saber que muestras son válidas y cuales no, primero dividen el conjunto de entrenamiento en dos : conjunto base con una cantidad pequeña

de datos entorno al 10% - 20% y un conjunto con el resto de datos que lo denominan como conjunto "Pool". En la figura §7.1 se encuentra una representación del conjunto base, en este caso, con cinco categorías diferenciadas por colores y grupos.

Una vez que lo tienen dividido, seleccionan los datos uno a uno del conjunto "Pool" para medir su distancia y entropía. Si se observa la figura 7.1a se aprecia que el punto *A* tiene una distancia similar a todos los centroides, luego, tendrá una entropía alta y será un dato con calidad de información. No obstante, la figura 7.1b nos presenta el dato *B* en él vemos que muestra una distancia al centro del conjunto azul pequeña y de facto "está dentro" de esa agrupación, por tanto, será un dato redundante y con entropía baja, luego, un dato con poca calidad.

Para corroborar la eficiencia de este método,los autores efectuaron diversas experimentaciones relacionadas con la cantidad de datos en base y la cantidad de características. La experimentación se realizó sobre un dataset sobre plagas en plantas llamado *plant pest dataset* consta de 6 clases diferentes y cada una de las clases presenta 1,000 muestras dando un total de 6,000. Cada imagen tiene una dimensión uniforme de 224x224x3 a color.

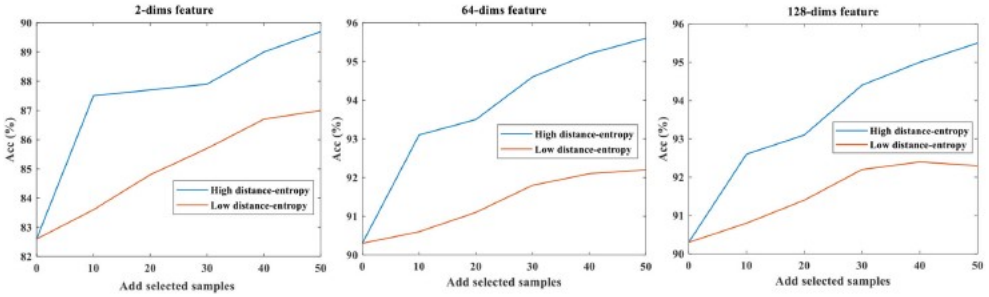


Figura 7.2: Gráfica sobre las dimensiones escogidas para el indicador.
Tomada de [10]

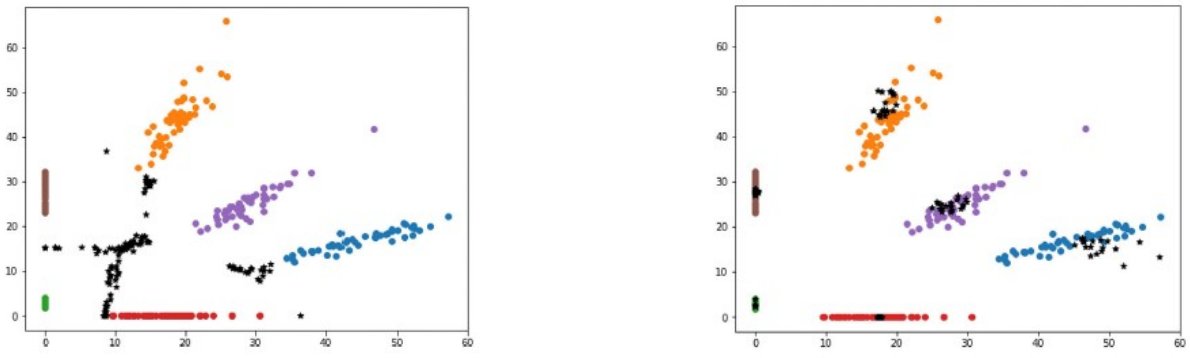
Efectuaron pruebas con transformaciones en los datos con el fin de tener el conjunto con 2 dimensiones, 64 dimensiones y 128 dimensiones. Los datos que estaban en el conjunto base eran 50 por clase, es decir, como hay 6 clases, 300 datos en la figura §7.2 la línea azul representa el conjunto base con los datos con mejor entropía y la naranja el conjunto base con los datos con una entropía más baja. Los resultados que obtuvieron les llevó a la conclusión que independientemente de la capacidad de representación del modelo el indicador que proponen es confiable y eficiente para seleccionar muestras con calidad de información.

		50 samples per class	100 samples per class
Base data	High d-e	82.6	87.7
	Low d-e	82.6	87.7
Add 10	High d-e	87.5	90.5
	Low d-e	83.6	88.4
Add 20	High d-e	87.7	92.1
	Low d-e	84.8	91.2
Add 30	High d-e	87.9	92.8
	Low d-e	85.7	91.5
Add 40	High d-e	89.0	93.0
	Low d-e	86.7	91.8
Add 50	High d-e	89.7	93.3
	Low d-e	87.0	92.0

Tabla 7.1: Resultados sobre la cantidad de muestras por clase en el conjunto

De igual manera hicieron experimentos para ver si influía la cantidad de muestras escogidas para la base y concluyeron que el efecto era similar al de las dimensiones, dicho de otro modo, el indicador es efectivo independientemente del número de datos en el conjunto base. Agregado a lo anterior obtuvieron que el uso de datos con mayor calidad aumenta la robustez y precisión del modelo hasta el 93.3% frente a datos con baja calidad y redundante que no ofrecían una mejora considerable.

Para finalizar hicieron una representación §7.3 de cuales fueron los puntos con mayor entropía y cuales los que menor tenían. Los puntos en negro son los que se añadieron



(a) Datos con mayor entropía

(b) Datos con menor entropía

Figura 7.3: Dos tipos de conjuntos finales
Tomada de [10]

7.2 Implementación y experimentación realizadas

Para la realización de las pruebas, igual que en el anterior apartado, he usado el dataset de uso libre *fashion-mnist* [25]. Este contiene 70,000 imágenes con 10 categorías de productos de moda en una escala de grises y con dimensiones 28x28. El conjunto de entrenamiento está formado por 60,000 muestras y el de test por 10,000. El código lo he realizado con python 3.10.9 en Jupyter Notebook [7] con las librerías de *Scikit-learn* en la versión 1.2.1 [15] y *Tensorflow* con versión 2.12.0 [5]. Todo el código creado por mí se puede encontrar en mi GitHub[1]

Para la implementación de esta técnica, lo primero que hago a hacer es dividir todo el conjunto de datos en dos partes, una para el entrenamiento y otra para el test.

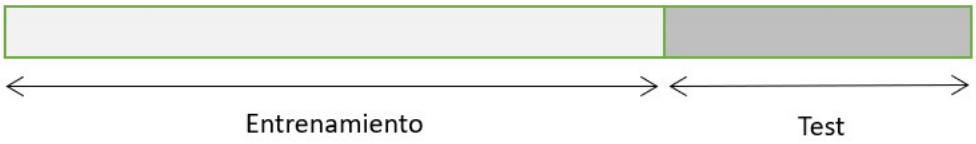


Figura 7.4: Ilustración de la primera división en subconjuntos para el medidor

A continuación, se vuelve a dividir el subconjunto de entrenamiento en dos, datos de base (Base Data) y el resto de los datos (Pool Data).

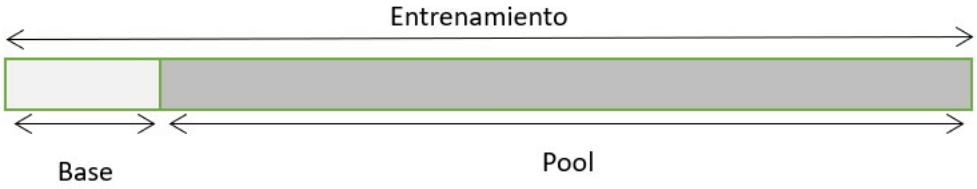


Figura 7.5: Ilustración de la segunda división en subconjuntos para el medidor

Para tener una visión más clara, la figura §7.6 muestra una representación de como quedaría el conjunto de datos que he obtenido llamado: "Conjunto base". Se aprecian diferentes agrupaciones con colores diferentes, cada color representa una clase teniendo en este caso tres clases. A partir de aquí llamaremos a esto "clases que se encuentran en la base" o para acortarlo "clases de la base".

[1] https://github.com/ricardoperalta00/TFG-.git

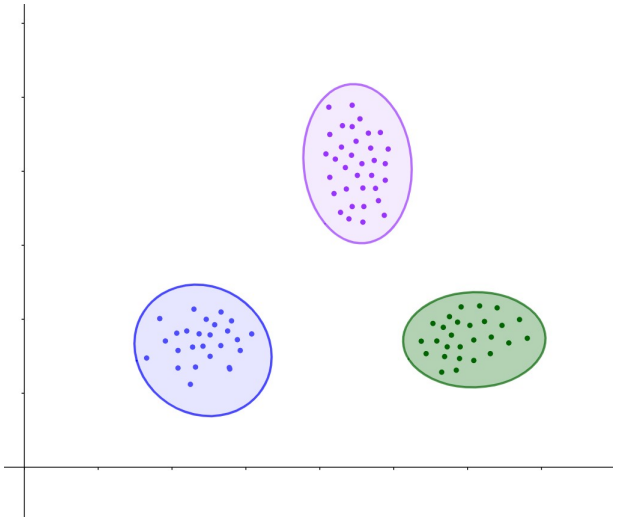


Figura 7.6: Representación del conjunto base

Una vez que tengo estás divisiones realizadas y el conjunto base, voy calcular la distancia euclídea que existe entre el centro de las clases de la base y cada uno de los puntos que se encuentran en el conjunto que he denominado anteriormente como "Pool". Para el cálculo del centro de las clases de la base simplemente se suman todos los puntos por componentes y se calcula la media. Imaginemos que una de las clases está constituida por tres puntos y son: x1 = [1, 2, 3] , x2 = [1.3, 1.5, 2] y x3 = [4, 2.3, 4]. Para calcular el centro hago lo siguiente: $[\frac{1+1,3+4}{3} + \frac{2+1,5+2,3}{3} + \frac{3+2+4}{3}]$.

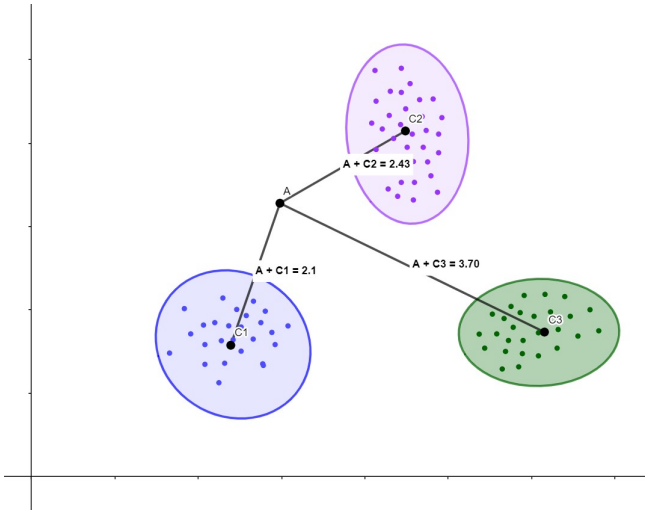


Figura 7.7: Representación de un punto del conjunto "Pool" y las distancias al centro de las clases

Una lista es obtenida con las distancias donde cada posición de la lista indica la distancia que hay entre el punto y la clase i, siendo i el índice. Siguiendo la figura §7.7 para el punto A se tendrá una lista de la forma: [2.1, 2.43, 3.70]. Esta lista de distancias va a ser pasada por una función *softmax* como (6.1) para obtener una distribución probabilística de las distancias uso la siguiente fórmula:

$$S(d_i) = \frac{e^{d_i}}{\sum_i e^{d_j}} \tag{7.1}$$

Seguidamente calculo la entropía del punto con la ecuación:

$$E = -\sum_i S(d_i) \cdot \log_2 S(d_i) \tag{7.2}$$

La distancia más corta va a representar una similitud más grande, por lo tanto, debería tener mayor peso en la distribución basada en la función *Softmax*. Como es sabido, el principio de la máxima entropía nos dice que cuando todas las probabilidades son iguales la entropía de ese evento será máxima.

Repetimos este proceso con todos los puntos del conjunto "Pool" y se almacenan sus entropías. Al finalizar todo el proceso se añade al conjunto base las n muestras que tengan mayor entropía, siendo n un número elegido por el usuario. El conjunto base con los puntos añadidos será el que se usará en la fase de entrenamiento.

Para la experimentación de esta técnica he optado por usar la red convolucional §6.2 que es igual a la del apartado anterior y que es entrenada con el optimizador descenso por gradiente estocástico. Como función de pérdida he usado "categorical cross entropy". He realizado varias pruebas con distintos tamaños en la base y en la cantidad de muestras añadidas.§7.2 y otra tabla con resultados sobre la precisión indicando los modos con los que se han obtenido los datos §7.3

Base	Añadidas	Precisión	Tiempo (s)
6000	1200	0.88	68.26
6000	2400	0.89	77.38
6000	3600	0.89	83.64
6000	4800	0.90	89.27
6000	6000	0.90	100.14

Tabla 7.2: Resultados del modelo CNN con medidor

Modo	Muestras	Precisión	Tiempo (s)
Random	6000	0.86	30.26
Random	12000	0.88	101.13
Estratificado	6000	0.88	31.70
Completo	60000	0.92	388.753

Tabla 7.3: Resultados del modelo CNN sin medidor

Para concluir con este apartado, se aprecia que usando este método podemos aumentar la precisión incluso con menos cantidad de muestras, este es el caso de usar 6000 ejemplos en la base y sumarle 4800 que nos daría un total de 10800 datos con una precisión del 90 % a diferencia de escoger 12000 datos de manera aleatoria con los que obtenemos un 88 %

Comparativa de métodos

En esta sección voy a realizar una comparativa de los métodos descritos anteriormente para poder tener una visión más acertada y finalmente cuál de los dos métodos funciona mejor.

Para tener una comparación justa con ambos métodos utilicé el mismo conjunto de datos inicial *Fashion-MNIST* [25], ya descrito en anteriores capítulos y en el mismo equipo y mismo lenguaje de programación.

La red convolucional con la que se realizaron ambos experimentos tienen la misma arquitectura y fue realizada con la librería *Tensorflow* de python [5]

Los resultados que se obtuvieron con el método estratificado fueron:

porcentaje	**10 %**	**25 %**	**30 %**	**40 %**	**50 %**	**80 %**
Num muestras	6000	15000	18000	24000	30000	48000
Precisión	0.87	0.89	0.90	0.90	0.90	0.92
Tiempo (segundos)	57.01	112.87	137.33	165.90	205.81	318.88

Tabla 8.1: Tabla de resultados para CNN usando muestreo estratificado

Por otra parte las que se obtuvieron con el método del medidor distancia entropía las contrastamos aquí §8.2:

Base	Añadidas	Precisión	Tiempo (s)
6000	1200	0.88	68.26
6000	2400	0.89	77.38
6000	3600	0.89	83.64
6000	4800	0.90	89.27
6000	6000	0.90	100.14

Tabla 8.2: Resultados del modelo CNN con medidor

Se evidencia que escogiendo muestras de calidad se puede conseguir una mejora en el rendimiento del modelo. Cuando tenemos con el muestreo estratificado un 25% de los datos, es decir, 15,000 muestras obtenemos un 89% de precisión en un tiempo de 112.87 segundos. Al comparar con la segunda tabla §8.2 comprobamos que con 8400 muestras (6000 base + 2400 añadidas) ya alcanzamos un 89% de precisión y en tiempo menor de 77.38 segundos. De igual manera pasa con usar un 30% o un 50% de los datos separados de manera estratificada, obteniendo una precisión del 90% mientras que con el método del indicador con muchas menos muestras obtenemos el mismo resultado y con menos operaciones realizadas.

Conclusión

Este trabajo realiza una revisión bibliográfica sobre cómo reducir los datos de entrenamiento en modelos de aprendizaje automático. Sin embargo, también posee una parte de implementación propia y experimentación. En primer lugar, para una mejor lectura del documento, se realiza un resumen de conceptos técnicos que van a ser introducidos a lo largo del trabajo clasificados por categorías. A continuación, se estudian los trabajos actuales que existen relacionados con la inteligencia artificial sostenible tanto en el entorno de organizaciones y empresas como en artículos de investigación. Por otra parte, he realizado una implementación y experimentación de dos métodos que trataban sobre la reducción del conjunto de datos en la fase de entrenamiento y finalmente una comparativa de ambos.

Tras analizar lo anteriormente expuesto, podemos concluir que las empresas se dedican más a crear una comunidad enfocada a la obtención de una inteligencia artificial sostenible poniendo herramientas como medidores de huella de carbono o recomendaciones a seguir sin darle una gran importancia al dato. Por otra parte, las investigaciones que se han realizado se observa que hay un aumento en el interés por conseguir esta sostenibilidad. Los enfoques más comunes que se encuentran son relacionados con implementaciones software, no obstante, pocas eran las que tenían una perspectiva en la reducción de datos.

Entre las que sí se dedican a la reducción de datos, pocas veces dan una mirada a la sostenibilidad, sino al rendimiento del mismo y en particular a modelos de aprendizaje automático y profundo. De manera habitual, tanto empresas como investigaciones tienen objetivos comunes en la obtención de datos de calidad sin ruido y balanceados y en mantener el rendimiento del proceso. Todas coinciden en que un entrenamiento exitoso va de la mano de un buen uso de datos. Como resultado, la mayor parte concuerda que con una buena elección de datos se puede reducir un conjunto de entrenamiento hasta en un 90 % y aún así tener un buen rendimiento total en la ejecución del modelo.

Luego de analizar las técnicas, he realizado una experimentación de dos métodos. En primer lugar inspirándome en un trabajo con una visión centrada en los datos [24]. He realizado una variante de su método haciendo un muestreo estratificado del conjunto de datos usados para el entrenamiento y obteniendo resultados que no comprometían el rendimiento del modelo. Por otra parte, para seleccionar datos de calidad me he inspirado en un indicador que hace uso de la distancia y la entropía [10]. He llevado a cabo una implementación y experimentación de su método que comparándolo con el anterior se saca como conclusión que la elección de datos con calidad informativa nos lleva a tener modelos más eficientes sin sacrificar su precisión.

Este trabajo me ha acercado al mundo de la investigación y he podido vivir de primera mano cómo son las relaciones entre investigadores e investigadoras del campo y las colaboraciones que pueden surgir valorando el trabajo que realizan y la importancia de la colaboración en el ámbito de la investigación. De igual forma, el proyecto me ha servido para profundizar mis conocimientos en el campo de la inteligencia artificial y estimar la gran cantidad de enfoques que existen.

Para finalizar, se podría profundizar más este estudio probando diferentes conjuntos de datos e implementando otras técnicas descritas como la destilación de datos para tener un mejor análisis del tema. Asimismo, espero que este documento sirva como guía para futuros proyectos y animo no solo a la investigación en la reducción de datos, sino también a fomentar una inteligencia artificial sostenible. Al hacerlo se reconoce que el progreso tecnológico no debe de tener consecuencias negativas sobre el medio ambiente, y que es posible alcanzar un equilibrio entre innovación y sostenibilidad. Con un enfoque responsable, podemos utilizar la IA como una poderosa herramienta para abordar desafíos globales, al tiempo que preservamos nuestro entorno natural.

BIBLIOGRAFÍA

[1] Z. Akram-Ali-Hammouri, M. Fernández-Delgado, E. Cernadas, and S. Barro. Fast support vector classification for large-scale problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6184–6195, 2022. doi: 10.1109/TPAMI.2021.3085969. (page 31).

[2] J. I. Bagnato. ¿cómo funcionan las convolutional neural networks? visión por ordenador, Jun 2020. URL https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/. (page 15).

[3] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. (page 29).

[4] C. Coleman, C. Yeh, S. Mussmann, B. Mirzasoleiman, P. Bailis, P. Liang, J. Leskovec, and M. Zaharia. Selection via proxy: Efficient data selection for deep learning. *CoRR*, abs/1906.11829, 2019. URL http://arxiv.org/abs/1906.11829. (pages 35 y 36).

[5] T. Developers. Tensorflow, Mar. 2023. URL https://doi.org/10.5281/zenodo.7764425. Specific TensorFlow versions can be found in the "Versions"list on the right side of this page.
See the full list of authors <a href="htt ps://github.com/tensorflow/tensorflow/graphs/contr ibutors»on GitHub. (pages 45, 50 y 54).

[6] Z. Hou, M. Qin, F. Sun, X. Ma, K. Yuan, Y. Xu, Y.-K. Chen, R. Jin, Y. Xie, and S.-Y. Kung. Chex: Channel exploration for cnn model compression, 2022. (page 28).

[7] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016. (pages 45 y 50).

[8] M. Kumar, X. Zhang, L. Liu, Y. Wang, and W. Shi. Energy-efficient machine learning on the edges. In *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 912–921, 2020. doi: 10.1109/IPDPSW50202.2020.00153. (page 22).

[9] C. Lacombe, I. Hammoud, J. Messud, H. Peng, T. Lesieur, and P. Jeunesse. Data-driven method for training data selection for deep learning. 2021(1):1–5, 2021. ISSN 2214-4609. doi: https://doi.org/10.3997/2214-4609.202112817. URL https://www.earthdoc.org/content/papers/10.3997/2214-4609.202112817. (page 36).

[10] Y. Li and X. Chao. Distance-entropy: An effective indicator for selecting informative data. *Frontiers in Plant Science*, 12, 2022. ISSN 1664-462X. doi: 10.3389/fpls.2021.818895. URL https://www.frontiersin.org/articles/10.3389/fpls.2021.818895. (pages 34, 47, 48, 49 y 57).

[11] T. Nguyen, Z. Chen, and J. Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=l-PrrQrK0QR. (page 31).

[12] T. Nguyen, R. Novak, L. Xiao, and J. Lee. Dataset distillation with infinitely wide convolutional networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=hXWPpJedrVP. (pages 31 y 32).

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. (page 24).

[14] D. Patterson, J. Gonzalez, U. Hölzle, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean. The carbon footprint of machine learning training will plateau, then shrink, 2022. URL https://arxiv.org/abs/2204.05149. (page 1).

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. (pages 43, 45 y 50).

[16] S. Rajeswar, P. Mazzaglia, T. Verbelen, A. Piché, B. Dhoedt, A. Courville, and A. Lacoste. Unsupervised model-based pre-training for data-efficient control from pixels, 2022. URL https://arxiv.org/abs/2209.12016. (page 28).

[17] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. Green ai, 2019. URL https://arxiv.org/abs/1907.10597. (page 27).

[18] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW. (page 35).

[19] B. Settles. From theories to queries: Active learning in practice. In I. Guyon, G. Cawley, G. Dror, V. Lemaire, and A. Statnikov, editors, *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pages 1–18, Sardinia, Italy, 16 May 2011. PMLR. URL https://proceedings.mlr.press/v16/settles11a.html. (page 35).

[20] M. S. Shafiee, M. J. Shafiee, and A. Wong. Efficient inference on deep neural networks by dynamic representations and decision gates. *CoRR*, abs/1811.01476, 2018. URL http://arxiv.org/abs/1811.01476. (page 28).

[21] Y. Shen, H. Yun, Z. Lipton, Y. Kronrod, and A. Anandkumar. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2630. URL https://aclanthology.org/W17-2630. (page 35).

[22] D. Thaler, L. Elezaj, F. Bamer, and B. Markert. Training data selection for machine learning-enhanced monte carlo simulations in structural dynamics. *Applied Sciences*, 12:581, 01 2022. doi: 10.3390/app12020581. (pages 37 y 38).

[23] M. Toneva, A. Sordoni, R. T. des Combes, A. Trischler, Y. Bengio, and G. J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm. (page 35).

[24] R. Verdecchia, L. Cruz, J. Sallou, M. Lin, J. Wickenden, and E. Hotellier. Data-centric green ai an exploratory empirical study. In *2022 International Conference on ICT for Sustainability (ICT4S)*, pages 35–45, June 2022. doi: 10.1109/ICT4S55073.2022.00015. (pages 31, 43 y 57).

[25] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. (pages 45, 50 y 54).

[26] X. Zhi, V. Babbar, P. Sun, F. Silavong, R. Shi, and S. Moran. A new baseline for greenai: Finding the optimal sub-network via layer and channel pruning, 2023. URL https://arxiv.org/abs/2302.10798. (pages 28 y 30).