



UNIVERSIDAD
DE GRANADA

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

Algoritmos meméticos para reducir datos de entrenamiento en modelos de aprendizaje profundo convolucionales

Autor

José Ruiz López (alumno)

Directores

Daniel Molina Cabrera (tutor)



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, Noviembre de 2024

Algoritmos meméticos para reducir datos de entrenamiento en modelos de aprendizaje profundo convolucionales

José Ruiz López (alumno)

Palabras clave: Algoritmos meméticos, Imágenes, Modelos de Aprendizaje profundo convolucionales

Resumen

Los modelos de **Aprendizaje Profundo** (Deep Learning) han supuesto un hito en la Inteligencia Artificial al ser capaz de procesar y ser capaces de reconocer patrones complejos. Dentro de estos, los modelos convolucionales se han mostrado muy capaces de identificar todo tipo de objetos/ características en imágenes.

Sin embargo, a diferencia de las personas, requieren un número muy alto de datos de entrenamiento para cada categoría que debe aprender. Eso implica, además de entrenamiento más largo, una recogida de datos de entrenamiento que, según lo que se desea que aprenda, puede ser problemático de obtener. Además de la obtención de los datos, la nueva ley europea sobre IA, **IA Act** requerirá sobre aplicaciones de IA con datos sensibles, una auditoría no solo del propio modelo, sino también de los datos utilizados para entrenarla. Auditoría que crecerá en complejidad conforme aumente en número el conjunto de entrenamiento. Por tanto, se hace conveniente poder reducir el conjunto de entrenamiento.

Ya se ha confirmado que incrementar el número de imágenes de entrenamiento puede mejorar el proceso o no, según si las imágenes realmente contribuyan al proceso de entrenamiento. Es más, gracias a las técnicas de aumento de datos (**Data Augmentation**) la posible necesidad de imágenes muy similares entre sí se reduce al ser capaz de construirse de forma automática más imágenes de entrenamiento (imágenes que no suponen un problema de cara a una auditoría).

En este trabajo planteamos el uso de estrategias avanzadas, como **algoritmos metaheurísticos**, y métricas de similitud entre imágenes, para establecer un proceso de reducción de imágenes de entrenamiento (selección de instancias) para poder reducir el conjunto de entrenamiento. De esta manera, se seleccionarían solo un conjunto reducido de imágenes representativas que, gracias a las **técnicas de aumento de datos**, puedan entrenar

modelos con una calidad suficiente. Por lo tanto, se podría reducir muy significativamente el conjunto de entrenamiento.

Memetic Algorithms for Reducing Training Data in Convolutional Deep Learning Models

José, Ruiz López (student)

Keywords: Memetic Algorithms, Images, Convolutional Deep Learning Models

Abstract

Deep Learning models have marked a milestone in Artificial Intelligence by being able to process and recognize complex patterns. Among these, convolutional models have proven very capable of identifying all kinds of objects/features in images.

However, unlike humans, they require a very high number of training data for each category they need to learn. This implies not only longer training times but also a data collection process that can be problematic to obtain, depending on what is desired for the model to learn. In addition to data acquisition, the new European law on AI, the **AI Act**, will require an audit not only of the model itself but also of the data used to train it, especially when dealing with sensitive data. The complexity of this audit will grow as the size of the training dataset increases. Therefore, it becomes necessary to reduce the training dataset.

It has been confirmed that increasing the number of training images may or may not improve the process, depending on whether the images truly contribute to the training process. Moreover, thanks to **data augmentation techniques**, the potential need for very similar images is reduced, as it is possible to automatically generate more training images that do not pose authorship issues.

In this work, we propose the use of advanced strategies, such as **meta-heuristic algorithms** and image similarity metrics, to establish a process for reducing training images (instance selection) in order to minimize the training dataset. This way, only a reduced set of representative images would be selected, which, thanks to **data augmentation techniques**, could sufficiently train models. Thus, the training dataset could be significantly reduced.

Yo, **José Ruiz López**, alumno de la titulación **TITULACIÓN** de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77964364E, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: José Ruiz López

Granada a X de mes de 201 .

D. **Daniel Molina Cabrera (tutor)**, Profesor del Departamento Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Algoritmos meméticos para reducir datos de entrenamiento en modelos de aprendizaje profundo convolucionales***, ha sido realizado bajo su supervisión por **José Ruiz López (alumno)**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a X de mes de 201 .

Los directores:

Daniel Molina Cabrera (tutor)

Agradecimientos

Poner aquí agradecimientos...

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Objetivos	2
1.3. Metodología	2
2. Marco Teórico	3
2.1. Fundamentos de Aprendizaje Profundo	3
2.2. Estado del Arte	3
3. Descripción de los Algoritmos	5
3.1. Algoritmos Meméticos	5
4. Metodología	7
4.1. Descripción del Sistema	7
4.2. Herramientas y Lenguajes de Programación	8
4.3. Proceso de Desarrollo	8
5. Implementación	9
5.1. Desarrollo del Código	9
5.2. Pruebas Iniciales	9
5.3. Ajustes Realizados	9
6. Resultados	11
7. Conclusiones	13

Capítulo 1

Introducción

1.1. Contexto

En la era actual, donde los datos se generan a un ritmo vertiginoso, la necesidad de procesar y analizar grandes volúmenes de información se ha vuelto crucial. Los **modelos de aprendizaje profundo**, y en particular las **redes neuronales convolucionales**, han demostrado su capacidad para alcanzar niveles sin precedentes de precisión en tareas como la clasificación de imágenes y el reconocimiento de patrones. Sin embargo, estos modelos a menudo requieren enormes cantidades de datos de entrenamiento, lo que plantea desafíos significativos en términos de tiempo, costo y recursos computacionales.

Es aquí donde entra en juego la importancia de los **algoritmos meméticos**. Combinando las fortalezas de las técnicas evolutivas con **métodos de búsqueda local**, estos algoritmos ofrecen un enfoque innovador y eficiente para la reducción de datos. La optimización de conjuntos de datos no solo puede mejorar el rendimiento de los modelos, sino que también permite una capacitación más rápida y menos costosa, facilitando así la investigación y el desarrollo en diversas aplicaciones.

Realizar un TFG sobre este tema no solo representa una oportunidad para explorar una frontera apasionante de la inteligencia artificial. La investigación en algoritmos meméticos para la reducción de datos puede ser clave para hacer más accesible el **aprendizaje profundo** a aquellos que enfrentan limitaciones de datos, permitiendo así un futuro donde la inteligencia artificial sea más inclusiva y eficiente.

1.2. Objetivos

1.3. Metodología

Capítulo 2

Marco Teórico

2.1. Fundamentos de Aprendizaje Profundo

2.2. Estado del Arte

Capítulo 3

Descripción de los Algoritmos

3.1. Algoritmos Meméticos

Capítulo 4

Metodología

En este capítulo se detallan la arquitectura del sistema que se ha implementado, como la tecnología usada, diseños, etc.

El repositorio ha sido almacenado en la plataforma **Github** desde el comienzo del proyecto. De forma obvia, la herramienta usada para el control de las versiones es **Git** [?].

4.1. Descripción del Sistema

La estructura del proyecto es la siguiente:

- **data** – Almacena los dataset utilizados por el proyecto.
- **docs** – Documentación del proyecto en latex.
- **img** – Imágenes generadas en el proyecto.
- **LICENSE** – Términos de distribución del proyecto.
- **README.md** – Descripción general.
- **requirements.txt** – Dependencias del proyecto.
- **results** – Resultados generados por el proyecto (fitness, tiempos, etc.).
- **scripts** – Scripts y programas secundarios para ejecutarse en el servidor GPU.
- **src** – Código fuente del proyecto.
- **utils** – Módulos y scripts de utilidad.

4.2. Herramientas y Lenguajes de Programación

El proyecto ha sido programado usando el lenguaje de programación **Python 3.10**. Junto a las librerías de **PyTorch 2.3.1**, **scikit-learn1.5.2** y **numpy 2.0.0**.

4.3. Proceso de Desarrollo

Al plantear el TFG, se pensó en como desarrollarlo.

Para ello se definieron unos objetivos iniciales y un plan temporal de cuyo desarrollo. Tras el planteamiento inicial, se realizó una investigación preliminar para tener contexto del Estado del arte y de que herramientas y tecnologías usar.

Una vez planeado todo, se empezó a desarrollar código, junto a pruebas iniciales. Después de una implementación inicial, se desarrollaría una experimentación iterativa para añadir nuevos experimentos y realizar ajustes continuos. Y finalmente realizar la documentación del TFG, junto a la presentación y defensa del proyecto final.

Capítulo 5

Implementación

5.1. Desarrollo del Código

5.2. Pruebas Iniciales

Las pruebas iniciales que se plantearon fueron tomar un dataset simple para realizar las primeras pruebas, y ya cuando funcionase correctamente, probar con otro dataset más complejo o realista. Para ello se decidió usar el dataset de **RPS**, ya que tiene un número pequeño de imágenes, 2520 en el **training set** y 3 clases (840 imágenes por clase), y 372 imágenes para el **test set** (124 por clase).

Para obtener unos primeros resultados con este dataset, se planteó usar el modelo de **Resnet50**

5.3. Ajustes Realizados

Capítulo 6

Resultados

Capítulo 7

Conclusiones

