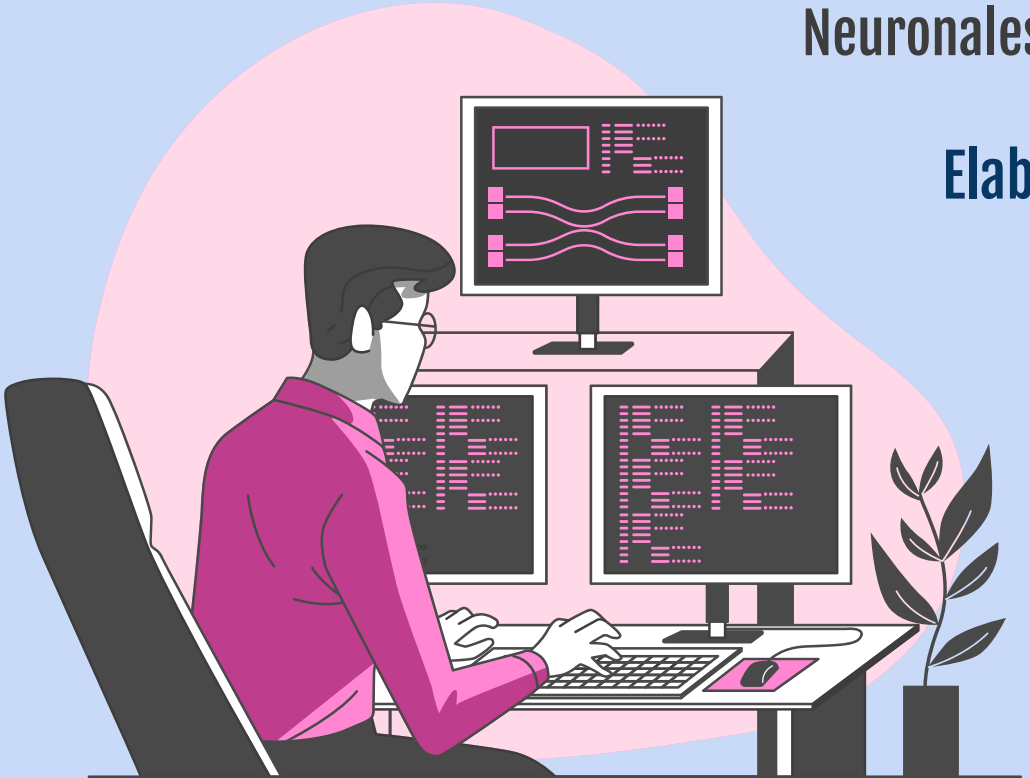
Centro de Investigación en Matemáticas, A.C.

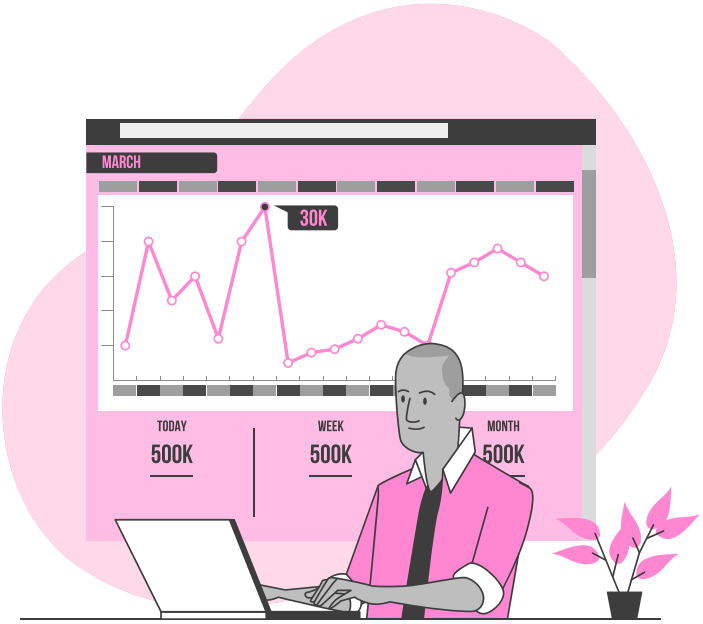
CIMAT

Búsqueda Automática de Arquitecturas Neuronales para clasificación mediante NNI

Elaborado por: Alondra Matos



Introducción



En la actualidad, la Búsqueda de Arquitecturas Neuronales (NAS) se realiza de forma automática en Python a través de varias bibliotecas de AutoML

AutoKeras proporciona una interfaz fácil de usar, pero **tiene ciertas limitaciones en la flexibilidad y personalización del experimento NAS.**

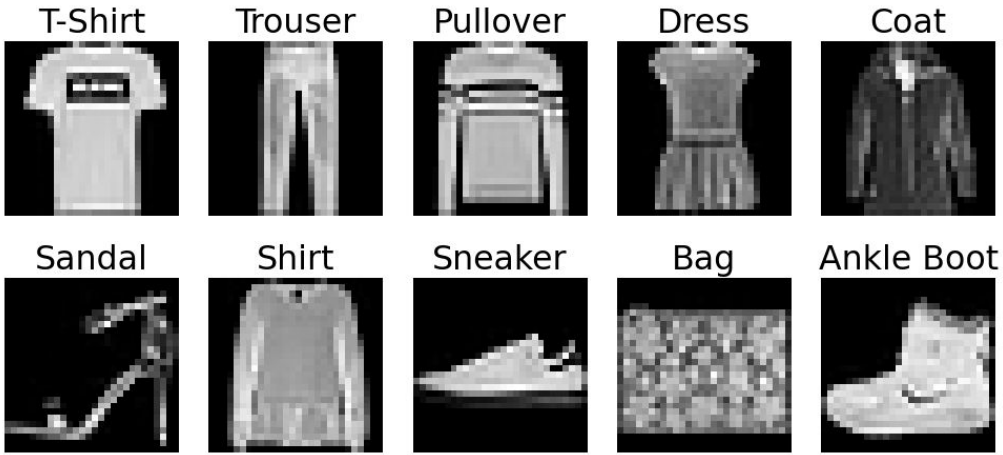
Objetivo

Explorar las características y capacidades de la paquetería **NNI de Microsoft para NAS**, con el fin de encontrar un buen algoritmo de clasificación de imágenes basado en redes convolucionales.

Base de datos

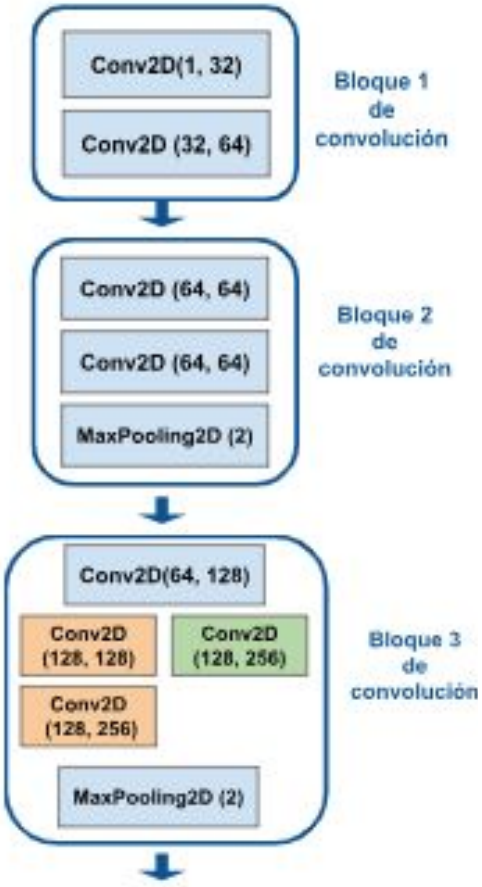
Características

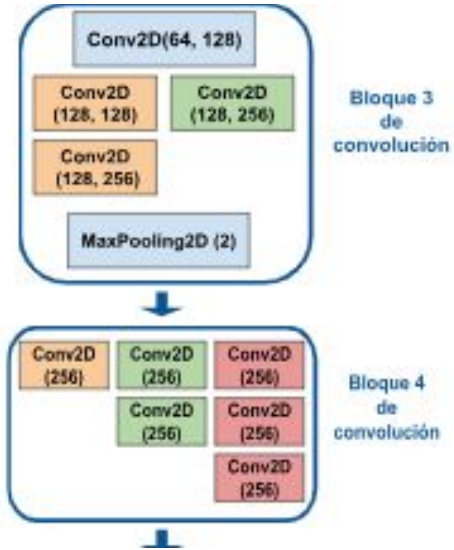
- Imágenes en escala de grises de 28x28 pixeles.
- Total de datos de entrenamiento: 60,000 imágenes.
- Total de datos de prueba: 10,000 imágenes.
- Base de datos balanceada.

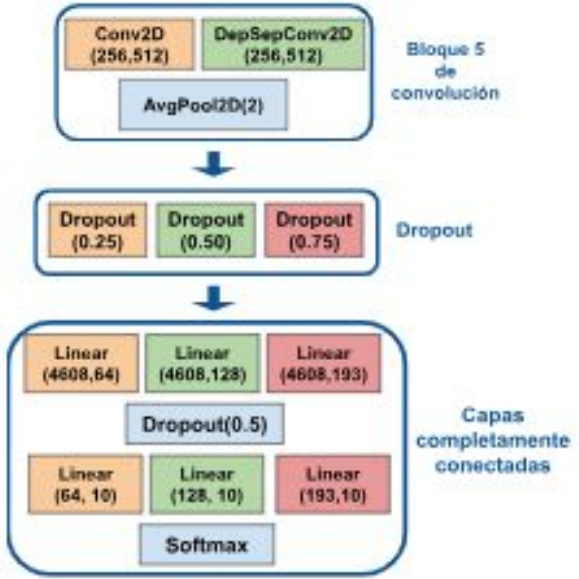


Experimento NAS

Espacio de búsqueda:
Personalizado

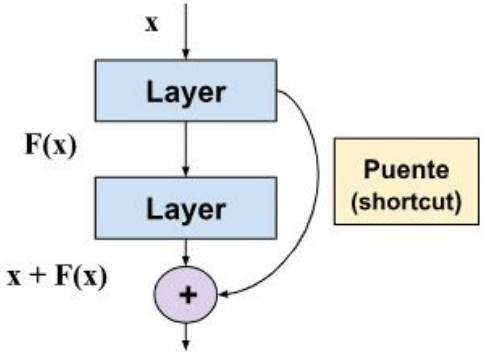


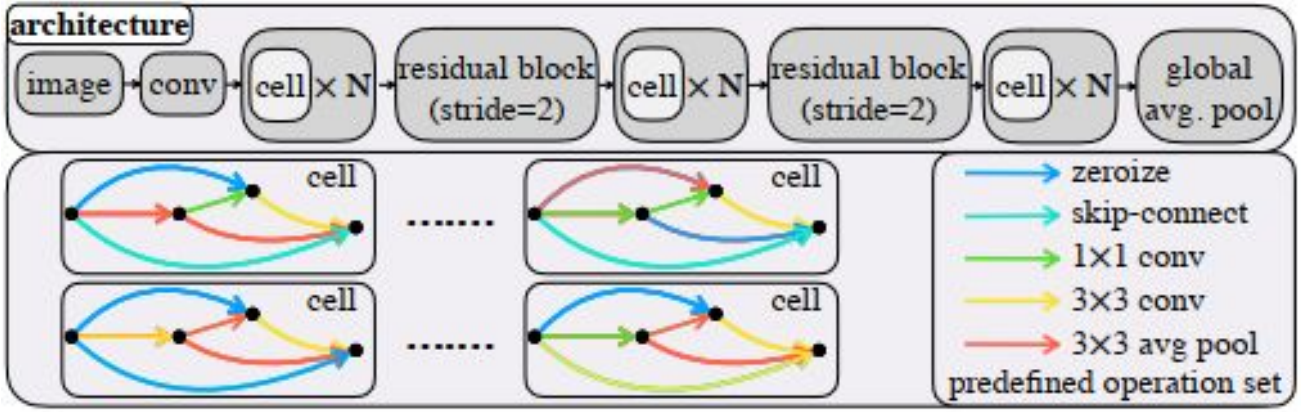




Experimento NAS

Espacio de búsqueda:
NasBench201





Algorithm 1 Aging Evolution

```
population ← empty queue                          ▷ The population.
history ← ∅                                       ▷ Will contain all models.
while |population| < P do                         ▷ Initialize population.
    model.arch ← RANDOMARCHITECTURE()
    model.accuracy ← TRAINANDEVAL(model.arch)
    add model to right of population
    add model to history
end while
while |history| < C do                            ▷ Evolve for C cycles.
    sample ← ∅                                    ▷ Parent candidates.
    while |sample| < S do
        candidate ← random element from population
                                                  ▷ The element stays in the population.
        add candidate to sample
    end while
    parent ← highest-accuracy model in sample
    child.arch ← MUTATE(parent.arch)
    child.accuracy ← TRAINANDEVAL(child.arch)
    add child to right of population
    add child to history
    remove dead from left of population           ▷ Oldest.
    discard dead
end while
return highest-accuracy model in history
```

Experimento NAS

Estrategia de búsqueda:

Algoritmo evolutivo regularizado

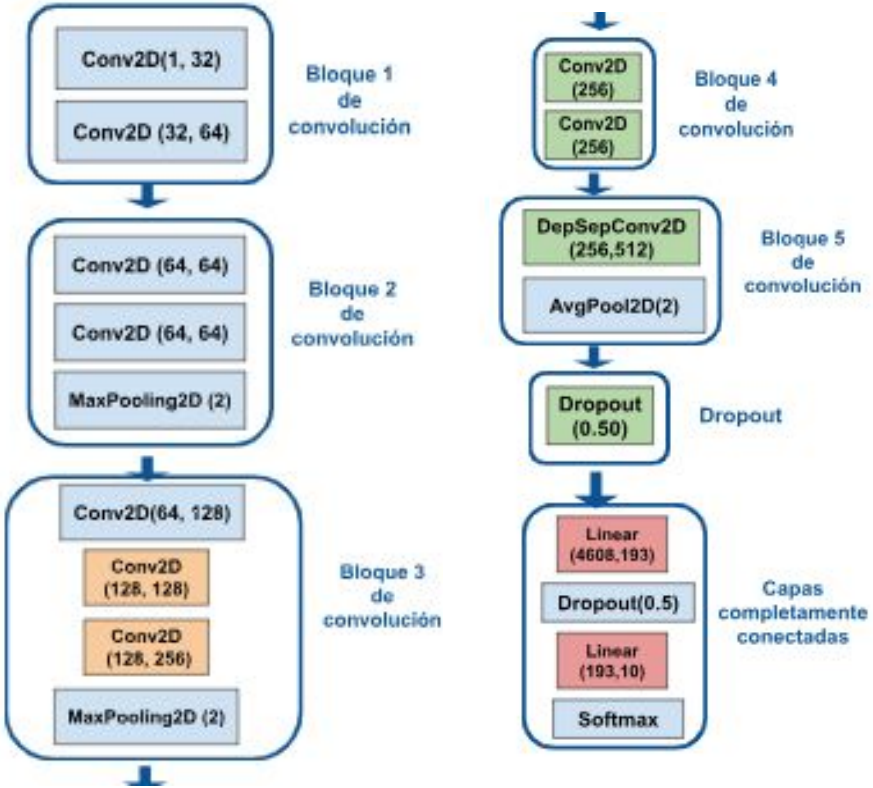
Evaluación del desempeño:

Exactitud

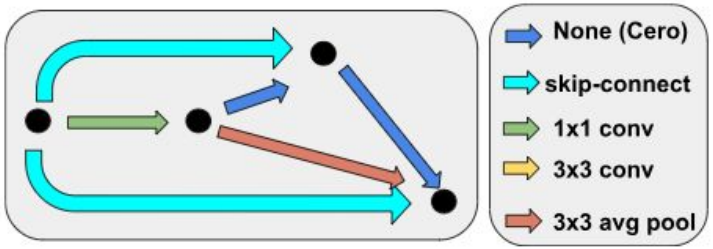
Resultados

	Experimento 1	Experiemento 2	Experiemento 3
Espacio de búsqueda	Personalizado	NasBench201	ResNet (Bloques residuales)
Estrategia de búsqueda	Algoritmo evolutivo regularizado	Algoritmo evolutivo regularizado	Optimización bayesiana "Optuna"
Número máximo de pruebas	6	6	3
Número de épocas por prueba	3	3	3
Tiempo total de ejecución (s)	11940	15882	15497
Tiempo promedio por prueba (s)	1990	2647	5165
Exactitud del mejor modelo (%)	91.34	88.96	83.96

Mejor resultado encontrado en el espacio de búsqueda personalizado



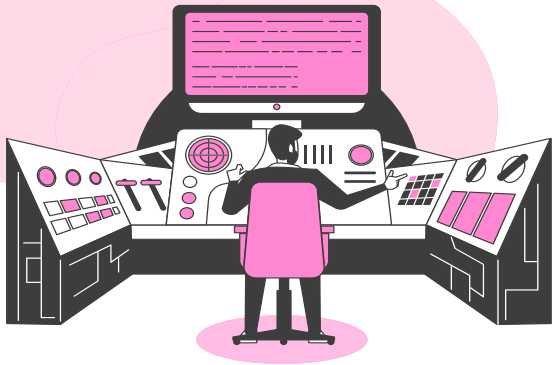
Mejor configuración de celda



Mejor resultado encontrado en el espacio de búsqueda NasBench201

Conclusión

La biblioteca NNI es una **buena alternativa** para llevar a cabo la Búsqueda Automática de Arquitecturas Neuronales (NAS), ya que proporciona una amplia variedad de implementaciones para crear experimentos NAS, incluyendo la capacidad de definir tu propio espacio de búsqueda. Además, la paquetería ya incorpora muchos de los espacios de búsqueda más populares en la actualidad.



REFERENCIAS

[1] MICROSOFT(2020). *"NNI Documentation"*. https://nni.readthedocs.io/en/latest/

[2] REAL, E., AGGARWAL, A., HUANG, Y., & LE, Q. V.(2019, July). *Regularized evolution for image classifier architecture search"*.In Proceedings of the aaai conference on artificial intelligence (Vol. 33, No. 01, pp. 4780-4789).

[3] DONG, X., & YANG, Y.(2020). *"Nas-bench-201: Extending the scope of reproducible neural architecture search"*.arXiv preprint arXiv:2001.00326.