

Neural Architecture Search con Aprendizaje por Refuerzo

Autor: Esp. Lic. Joaquín Sebastian Tschopp
Docentes: Dr. Ing. Cesar Caiafa

Motivación

La Evolución del Diseño Manual: De 8 capas (AlexNet) a +100 capas (ResNet/DenseNet).

- *Problema:* La complejidad arquitectural supera la intuición humana. El diseño se vuelve "prueba y error" costoso.

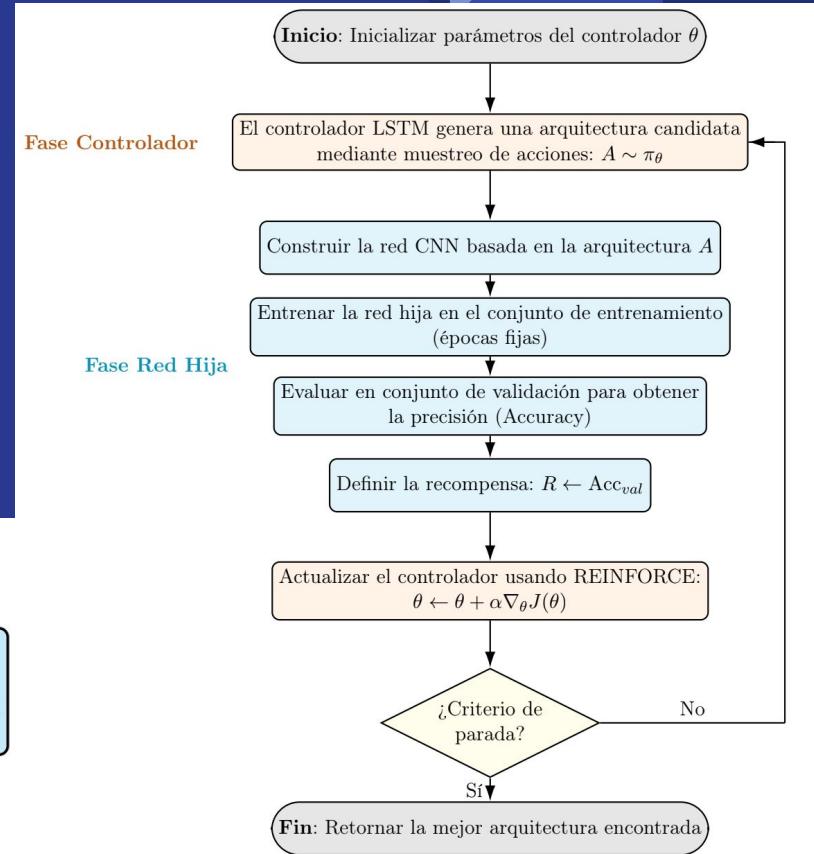
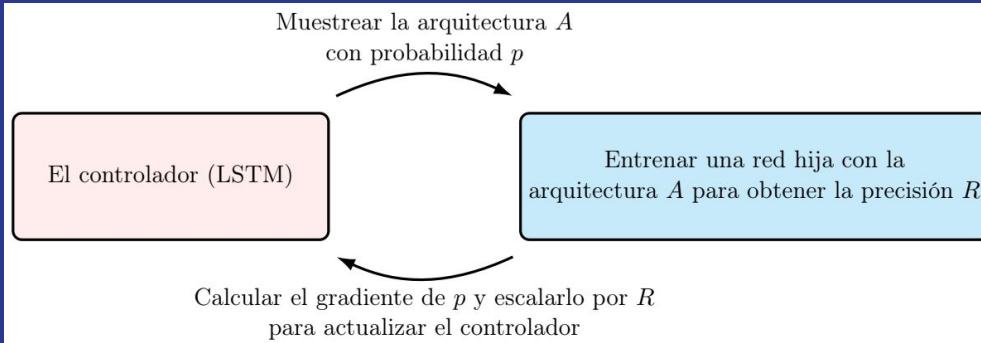


El Foco de esta Monografía

- **Validación Empírica:** ¿Es reproducible este hito científico?
- **Democratización:** ¿Es viable aplicar estos métodos sin las 800 GPUs de Google?
- **Caja Negra:** Reconstruir la arquitectura *NASCNN15* sin código fuente disponible.

El Método: NAS con RL

1. Tenemos un **Controlador** (una RNN) que propone una arquitectura.
2. Construimos y entrenamos esa '**Red Hija**'.
3. Su precisión en validación se convierte en la **recompensa**.
4. El controlador aprende qué diseños funcionan y cuáles no.



El Cerebro: Controlador LSTM

El Rol del Controlador (LSTM)

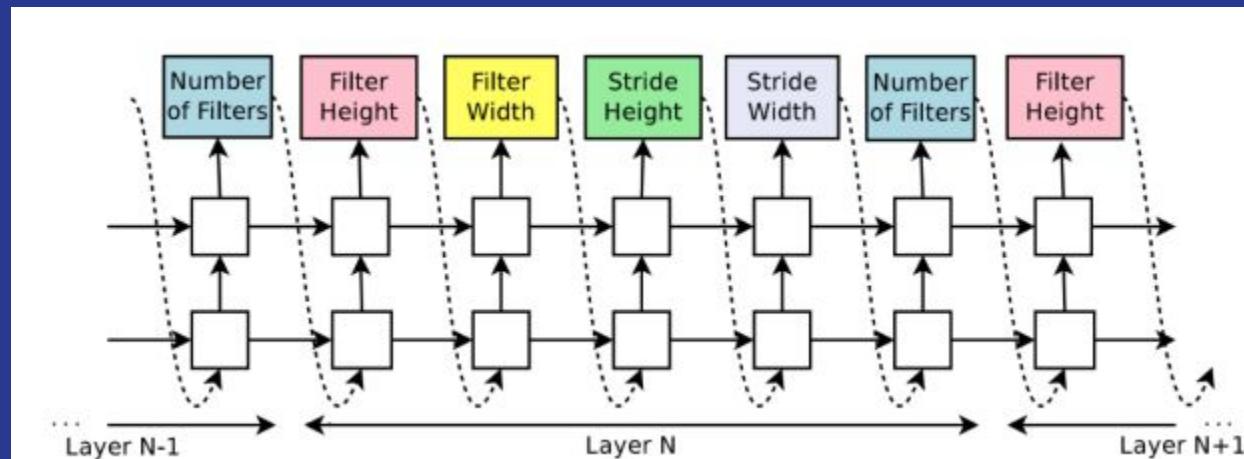
- Funciona como un arquitecto escribiendo planos capa por capa.

Generación del "ADN" (Acciones Discretas)

- En cada paso t , elige una configuración específica mediante muestreo probabilístico ($a_t \sim \pi_\theta$)
Decisiones: Tamaño de Kernel (1, 3, 5...), Número de Filtros, Stride.

Aprendizaje (REINFORCE)

- Si la red hija funciona bien (alta Accuracy), el controlador aumenta la probabilidad de repetir esa secuencia de decisiones.



Estrategia Experimental

El Desafío Computacional: Paper original: Cluster de 800 GPUs vs. Hardware modesto.

- Imposibilidad de realizar la búsqueda completa (Search Space) en tiempos razonables.

2. Fase A: Validación del AGENTE. Implementación: ‘demo’

- **Objetivo:** ¿Aprende el controlador?
- **Escala:** Búsqueda acotada a 160 arquitecturas (vs 12,800).
- **Foco:** Observar la *pendiente* de mejora de la recompensa.

3. Fase B: Validación de la ARQUITECTURA (NASCNN15)

- **Objetivo:** ¿Es la red resultante realmente competitiva?
- **Método:** Ingeniería inversa del diagrama original y entrenamiento desde cero (Stand-alone training).



Fase A - La Búsqueda 'demo'

Dimensiones de la Búsqueda

Total Evaluado: 160 Arquitecturas únicas. * **Ciclos:** 10 Episodios (Batch size = 16 redes por episodio).

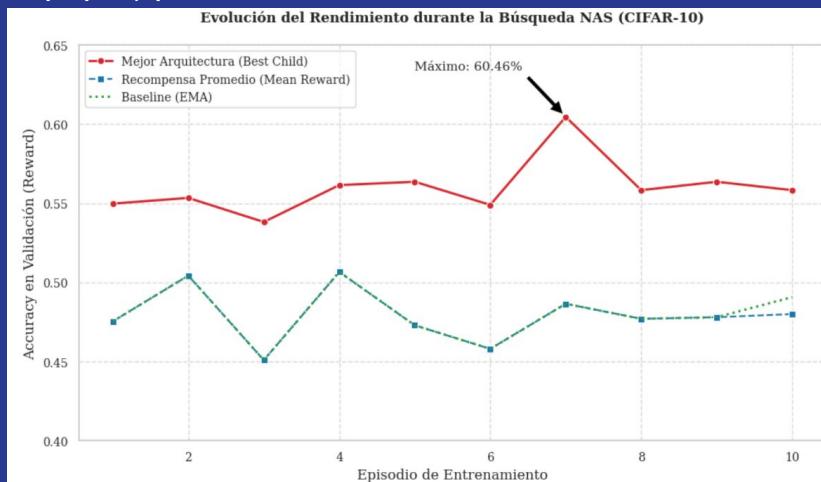
Adaptaciones de Viabilidad ("Trade-offs")

Entrenamiento Redes Hijas: Reducido a 20 épocas (vs 50 paper) para acelerar feedback.

Layer Schedule (Profundidad Progresiva):

- Inicio: Redes de 6 capas.
- Incremento: Aumenta la profundidad cada 16 arquitecturas (1 episodio).

Resultado: Convergencia y máximo global alcanzado en **Episodio 7**



Resultado Fase A

Una Arquitectura "No Convencional"

"El Hallazgo (Episodio 7) El controlador encontró una arquitectura de **13 capas**.

Rendimiento: 60.46% de Accuracy en Validación (limitado por las 20 épocas de entrenamiento)

ADN" encontrado, con formato $[k, f, s, p]$.

```
[  
    [1, 2, 11, 1],    % Capa 1  
    [1, 4, 8, 1],    % Capa 2  
    [10, 2, 2, 8],   % Capa 3  
    [2, 13, 13, 5],  % Capa 4  
    [15, 4, 2, 11],  % Capa 5  
    [24, 10, 7, 22], % Capa 6  
    [7, 13, 4, 25],  % Capa 7  
    [4, 1, 3, 17],   % Capa 8  
    [8, 10, 11, 17], % Capa 9  
    [11, 5, 9, 6],   % Capa 10  
    [19, 1, 4, 5],   % Capa 11  
    [17, 2, 3, 2],   % Capa 12  
    [18, 12, 16, 11] % Capa 13  
]
```

Capa	Vector ADN	Kernel (K)	Filtros (F)	Stride (S)	Padding (P)
1	[1, 2, 11, 1]	1×1	2	11	1×1
2	[1, 4, 8, 1]	1×1	4	8	1×1
3	[10, 2, 2, 8]	10×10	2	2	8×8
4	[2, 13, 13, 5]	2×2	13	13	5×5
5	[15, 4, 2, 11]	15×15	4	2	11×11
6	[24, 10, 7, 22]	24×24	10	7	22×22
7	[7, 13, 4, 25]	7×7	13	4	25×25
8	[4, 1, 3, 17]	4×4	1	3	17×17
9	[8, 10, 11, 17]	8×8	10	11	17×17
10	[11, 5, 9, 6]	11×11	5	9	6×6
11	[19, 1, 4, 5]	19×19	1	4	5×5
12	[17, 2, 3, 2]	17×17	2	3	2×2
13	[18, 12, 16, 11]	18×18	12	16	11×11

Fase 2 - NASCNN15 (Ingeniería Inversa)

El Desafío de Reproducibilidad

- La arquitectura no fue publicada en código, solo como diagrama visual.
Reconstrucción "forense" manual en PyTorch basada en la topología del paper.

Anatomía de NASCNN15

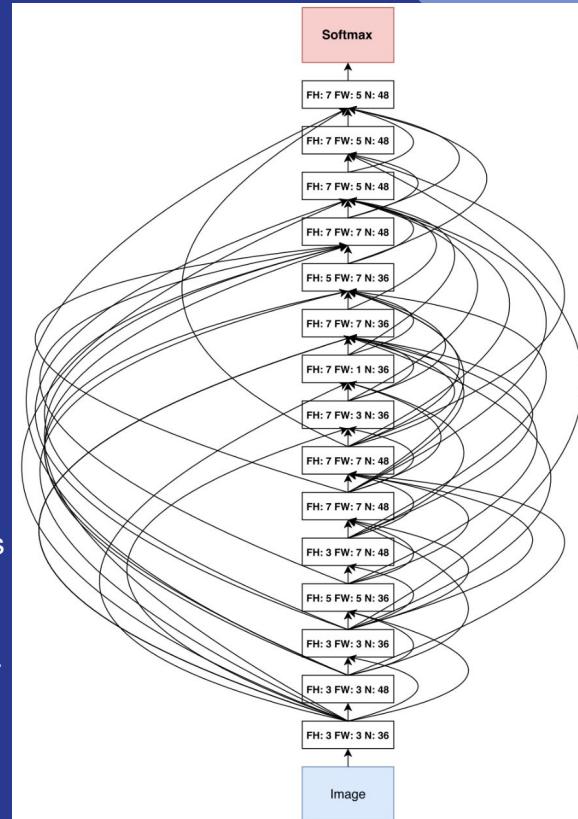
Profundidad: 15 capas convolucionales + Clasificación.

Conectividad Densa (Skip-Connections)

Topología no lineal: Una capa puede recibir inputs de hasta 9 capas anteriores concatenadas.

Kernels Híbridos: Mezcla inusual de filtros cuadrados (3×3 , 5×5) y rectangulares (1×7 , 7×1).

Preservación Espacial: Stride 1 fijo sin Max Pooling (mantiene resolución 32×32).



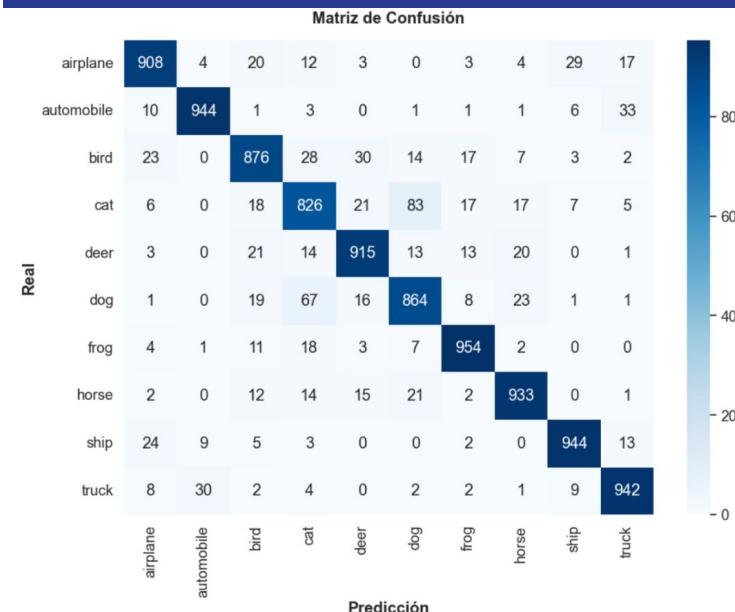
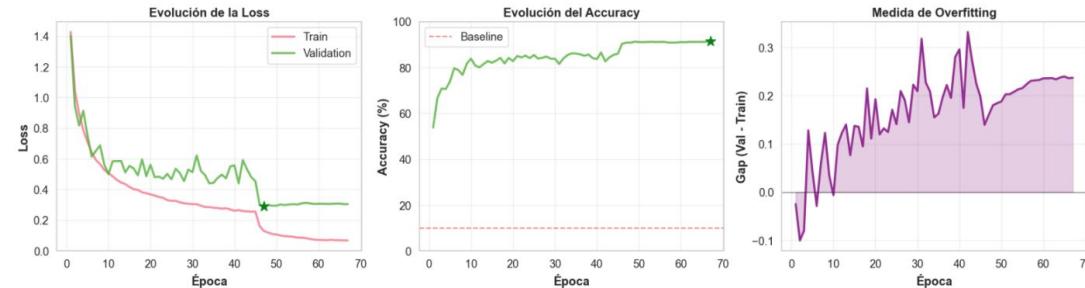
Resultados Fase B - Comparativa

Accuracy en Test: 91.06% (vs 94.50% reportado en el paper).

Parámetros: ~4.2 Millones (Coincidencia exacta con el original).

Entrenamiento: Detenido en época 87 por *Early Stopping* para evitar sobreajuste

Métrica	Esta reproducción	Paper original Zoph y Le, 2016
Accuracy validación	91.30 %	—
Accuracy test	91.06 %	94.50 %
Loss final (validation)	0.3041	—
Parámetros	~4.2 M	4.2 M
Épocas entrenadas	87 / 300	—
Early stopping	Sí	—



Conclusiones



Validación del Método (NAS + RL): Se confirmó empíricamente que un agente autónomo puede descubrir topologías eficientes (kernels gigantes, conexiones densas) que superan la intuición humana, incluso en escenarios de recursos limitados.



La Barrera de la Reproducibilidad: La reconstrucción de NASCNN15 evidenció la dificultad de replicar ciencia de vanguardia cuando faltan detalles de implementación. Se logró recuperar un rendimiento competitivo (91.06%), validando la solidez estructural del diseño original.



Limitaciones y Vigencia: Generalización: Arquitecturas altamente optimizadas tienden a sobreajustarse al dataset de validación original (caída de rendimiento en CIFAR-10.1). *

Costo: A pesar de la eficiencia del modelo final, el costo computacional de la *búsqueda* sigue siendo prohibitivo para la mayoría de los investigadores.

Muchas
Gracias!!