

## Práctica 1-Recorridos

---

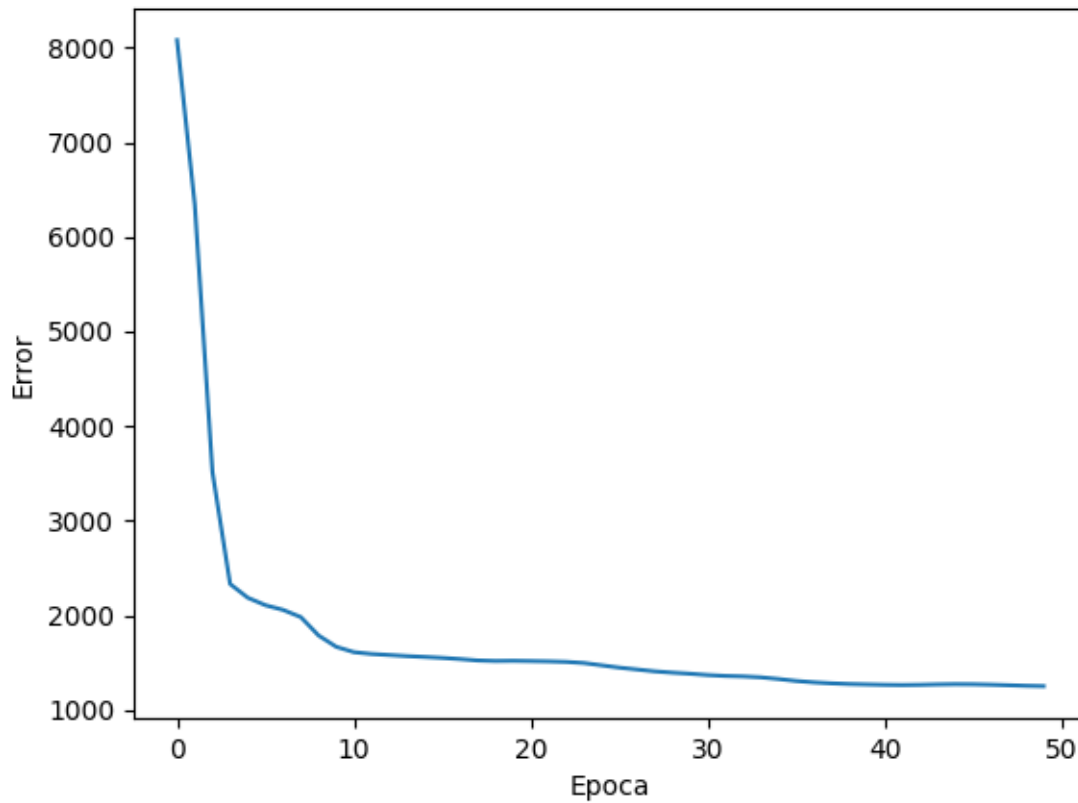
Se agregó un graph\_search sin la lista cerrada, se contaron los nodos expandidos, y se agregaron 2 nuevas interfaces de colas ramificacionacotacion y ramificacionacotacionconsubestimacion que usaron las funciones path\_cost y path\_cost y h() respectivamente para ordenar elementos en la cola

Nodos Expandidos	A → B	O → N	L → R	H → Z	D → O
nivel-anchura	19	45	14	36	23
profundidad-altura	15	27	29	15	9
ramificación y acotación	53	2180	36	304	69
ramificación y acotación con subestimación	6	237	15	19	15

Vemos como el método de ramificación y acotación, sin una lista cerrada, entra en un bucle del que les cuesta salir. Al agregarle la heurística consigue disminuir mucho el número de nodos expandidos.

---

Hemos configurado la red neuronal para que las entradas sean cada uno de los pixels que conforman cada número y una salida por cada uno de los números del 0 al 9. Entrenamos la red hasta un máximo de 50 épocas o cuando la diferencia entre épocas sea muy baja.



En esta ocasión, los valores random escogidos ha hecho que se lleguen a las casi 50 épocas de entrenamiento. Vemos como al principio el error desciende rápidamente y posterior mente se va estabilizando.

Con el conjunto test hemos obtenido un 0.9138 % de acierto

---

### Practica 3-Qlearning

---

En esta práctica hemos implementado algoritmos de explotación y exploración+explotación (Greedy y eGreedy) y comparamos con la selección aleatoria. Contaremos el número de acciones que se toman en cada método.

Para el caso eGreedy explotaremos el 80% de las veces y el otro 20% exploraremos.

Pasaremos unas cuantas veces y obtendremos la media

	Numero de acciones
Aleatorio	213,4
Greedy	19
eGreedy	21,6

Al tener un problema pequeño el eGreedy tiene un número de acciones superior al Greedy

---