

Memoria de las prácticas de fundamentos de los sistemas inteligentes.

ALBERTO CÁRDENES PÉREZ

Contenido

Práctica 1: Búsqueda	2
Practica 2: Redes neuronales	2
Práctica 3: Aprendizaje por refuerzo	4

Práctica 1: Búsqueda

En la primera parte de la práctica se planteó el problema de la búsqueda por el método de la ramificación y salto no informado. Se creó una nueva clase a modo de 'fringe' que se llamó `no_informed_list`, esta clase contiene los siguientes métodos:

- `append`: introduce el primer elemento del problema en la lista
- `extend`: dado un conjunto de nodos los añade a la lista
- `Pop`: obtiene el elemento con el menor `path_cost` de la lista, esto se hace mediante un bucle que mira todos los `path_cost` de los elementos de la lista y devuelve el menor

El funcionamiento es idéntico al del código original, solo se cambia la clase que actúa como 'fringe' que ahora es la que nosotros creamos.

En la segunda parte de esta misma práctica se creó otra clase 'fringe' llamada `informed_list` que contiene los métodos `append` y `extend` iguales a la clase anteriormente nombrada y el método `pop` devuelve el elemento con un menor `path-cost` sumado con su `heuristic-cost`.

Para esta parte de la práctica hubo que modificar la clase `Node` para que, cuando se creara, incluyera el coste heurístico a sus atributos. Esta modificación solo se aplica a los nodos creados mediante el método `expand` de la clase `Node`, por lo que el nodo inicial del problema tendrá coste heurístico 0, esto se puede solucionar añadiendo el coste heurístico cuando se añade el nodo inicial a la lista, pero es innecesario ya que el nodo inicial, en la primera vuelta, será el único de la lista y la función `pop` lo devolverá sin importar la heurística.

Los resultados obtenidos son los siguientes, en nodos visitados:

	A → B	B → F	C → H	E → G	B → A
Anchura	19	5	23	10	25
Profundidad	15	2	26	42	9
BB Informada	6	3	5	5	5
BB no informada	24	10	26	9	29

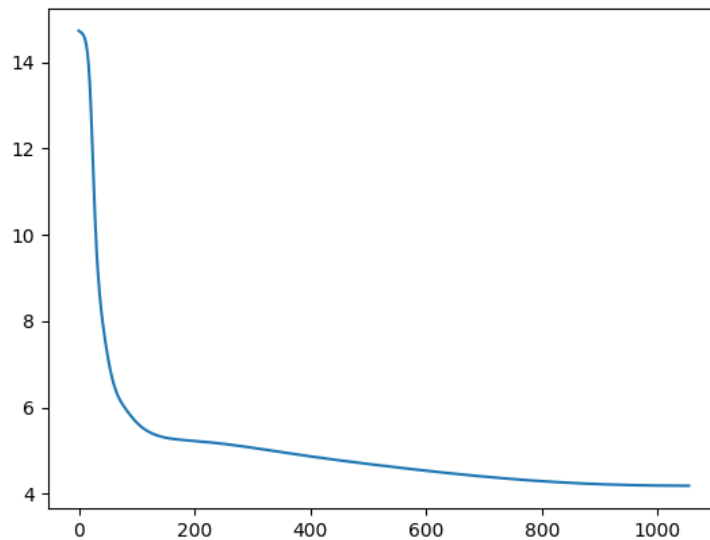
Por lo que se puede observar, la ramificación y poda con heurística es la que mejor resultados obtiene.

Practica 2: Redes neuronales

Esta práctica también se divide en dos partes, el conjunto de iris y el conjunto mnist

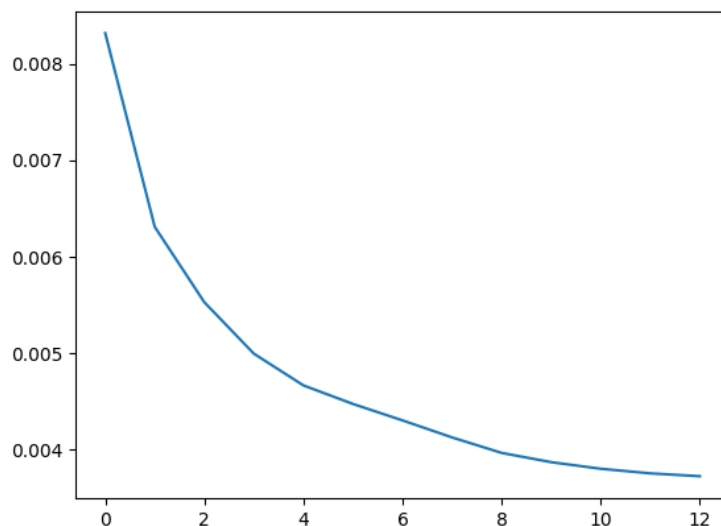
En la primera parte se planteaba un modelo basado en cinco neuronas en la capa intermedia y tres neuronas en la capa final, cada neurona de la capa final se enciende a un determinado tipo de flor. Esta red neuronal realiza un descenso por el gradiente del error cuadrado con un factor de aprendizaje de 0,01. Con esta configuración la red es capaz de llegar a un 100% de acierto con el conjunto de test.

Ya que solo tenemos un conjunto se ha separado un 70% para el conjunto de aprendizaje, un 15% para el conjunto de validación y un 15% para el conjunto de test. La gráfica de error es la siguiente:



Como se puede ver, se necesita un gran número de épocas (>1000) para que la variación del error sea mínima. Después de este entrenamiento se obtiene un 100% de acierto en el conjunto de test.

La segunda parte de la practica requirió una búsqueda sobre configuraciones para resolver el problema del mnist. La mejor configuración que encontré fueron cien neuronas en la capa intermedia y una función de error que es la media de la multiplicación entre el valor esperado y el logaritmo del valor predicho, con una reducción de índices de 1. Esta configuración consigue un porcentaje de acierto del 97% en el conjunto de test en 13 épocas. La gráfica del error es la siguiente:



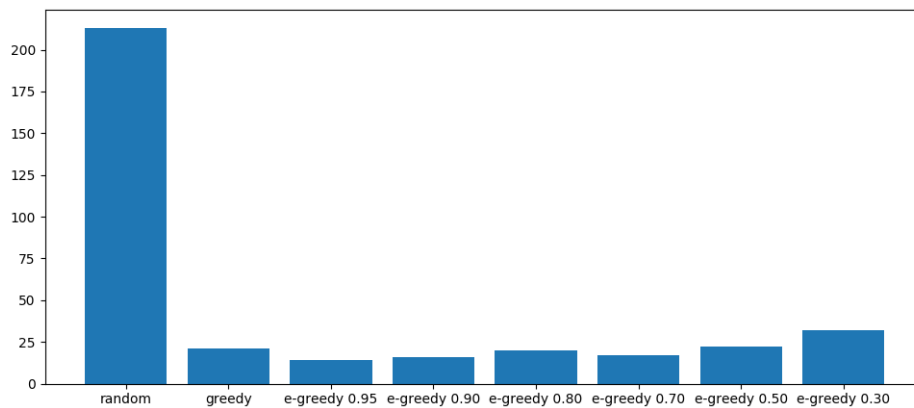
Otra posible función de error es la de entropía cruzada con logits, sin embargo, esta configuración consigue un mayor porcentaje de acierto, pero un mayor error cuadrado medio, es decir, acierta más veces, pero los fallos están más alejados de valores correctos.

Práctica 3: Aprendizaje por refuerzo

En esta última práctica de la asignatura se plantea el aprendizaje por refuerzo en un tablero que tiene una serie de casillas prohibidas y una casilla objetivo (recompensa máxima), para ello se siguen la política de siempre exploración, la de siempre explotación y unas mezclas entre ambas con probabilidades del 80%, 50% y 70% de explotación.

Para la realización de esta práctica se modificó el código dado y se introdujo un bucle que iba cambiando de política y un método que, pasándole la política, devolvía la acción a tomar.

Tras 1000 repeticiones el número de acciones promedio de cada política es la siguiente:



Por los datos obtenidos se puede pensar que para este problema determinado la explotación al 95% es la mejor opción.