

# Tree of Nets

## 1 Preliminaries

This section is devoted to the description of the considered problem, namely the predecessor Search and of some preliminary definitions about Neural Networks, the model on which is based the solution to the problem proposed in this study.

L'idea principale di questo studio è di trovare un modello efficiente in tempo e spazio che possa accedere ad un elemento di una lista ordinata in tempo e con una struttura dati che occupa spazio  $O(1)$ . Solitamente per questo tipo di problema vengono usati i B-Tree che invece occupano spazio  $O(n)$  e effettuano la ricerca in  $O(\log n)$

### 1.1 Predecessor Search

Let  $X$  be a subset of part of a universe  $U$ , sorted according to the  $\leq$  order relation defined on  $U$ . We assume that each element of  $U$  can be represented by  $d$  bits. Moreover, we assume that if the binary format is such to represent an integer or a floating point, the arithmetic will be done accordingly. We also assume that lexicographic ordering of the binary representation of each element in  $U$  preserve the order relation  $\leq$ . That is, a comparison between two elements of  $U$  can be done using either the binary representation as a string or a primitive comparison operation. For the special case of float IEEE 754 representation the lexicographic ordering preserve the order relation  $\leq$  by flipping the sign bit of the representation. Moreover, we will consider each element of  $U$  either as a "number" in decimal notation, i.e.,  $x$ , or as a vector in binary notation denoted as  $\vec{x}$ , according to its "internal format", i.e., integer or float. An analogous notation is followed for  $X$ . Our model is a unit-cost RAM with wordsize  $d$ .

The *predecessor search problem* consist in finding the position of the largest key in  $X$  not greater than a given input key  $x$ .

Example 1: Let  $X$  be

$$X = \{2,3,4,5,12,15,18\}$$

Let us assume that 7 is the key to search. The predecessor search should return the position  $i = 4$ .

**Neural Nets Solution** Consider a sorted set of keys  $X$ , with  $|X| = n$ . Let  $F$  be the *Empirical Cumulative distribution* of elements of  $U$  with respect to

(sample)  $X$ . That is, for each  $z \in U$ ,  $\frac{|y \in X|y \leq z|}{n}$ . Notice that knowledge of  $F$  provides a solution to our problem since, given an element  $z$  of  $U$ , only one evaluation of  $F$  provides  $p(z) = F(z) * n$  ceil. Our task is find to a function  $\tilde{F}$  which gives a good approximation of  $F$ .

We concentrate on *feed forward network* to determinate  $\tilde{F}$ . We consider 3 model with some common settings:

- The loss function is the most common choice, mean square error  $E(\vec{A}, \vec{B}) = \frac{1}{m} \sum_{i=1}^m (a_i - b_i)^2$ .
- All no inputs neurons have Leaky-ReLU activation function ( $\alpha = 0.05$ )
- learning rate = 0.1
- Batch-size = 64
- Optimizer = SGD with Momentum ( $\mu = 0.9$ )
- Initializer for weights connection and bias is Random Normal with  $\mu = 0$  and  $\sigma = 0.05$
- Stopping Criteria: the loss don't improve with patience 10 or number of epochs is 20000

NN1K model have 64 inputs neurons and 1 outputs neurons.

NN2K model have 64 inputs neurons, 256 hiddens neurons and 1 outputs neurons.

NN3K model have 64 inputs neurons, 2 layers of 256 hiddens neurons and 1 outputs neurons.

**Tree with NN0 leaf** The idea is to split the dataset in  $s$  parts. So there are  $s$  Neural nets like previous paragraph with 64 input neurons and 1 output neurons.

NN0 model.

- The loss function is the most common choice, mean square error  $E(\vec{A}, \vec{B}) = \frac{1}{m} \sum_{i=1}^m (a_i - b_i)^2$ .
- The activation function of output neurons is Leaky-ReLU ( $\alpha = 0.05$ )
- Learning rate is a schedule decay learning rate which start from 0.005.  
 $lr_t = 0.005 * \frac{1}{1+0.001*t}$
- Batch size = 32
- Optimizer = Nesterov Momentum ( $\mu = 0.9$ )
- Initializer for weights connection and bias is Random Normal with  $\mu = 0$  and  $\sigma = 0.05$
- Stopping Criteria: the mean absolute error don't improve with patience 10 or number of epochs is 20000

## 2 Results

In our test we used 3 different datasets, all with the uniform distribution:

- file3  $2^9$  elements
- file7  $2^{13}$  elements
- file10  $2^{20}$  elements

### Tables of NN1K, NN2K, NN3K model

- $\epsilon$  = Max error
- SpaceOVH = space of the model in KB with respect to dataset

NNxK file 3			NNxK file 7			NNxK file 10		
Model	$\epsilon$	SpaceOVH	Model	$\epsilon$	SpaceOVH	Model	$\epsilon$	SpaceOVH
NN1K	9	$8.93 \times 10^{-2}$	NN1K	53	$3.09 \times 10^{-3}$	NN1K	905	$2.42 \times 10^{-5}$
NN2K	4	$1.29 \times 10^1$	NN2K	33	$8.06 \times 10^{-1}$	NN2K	1031	$6.29 \times 10^{-3}$
NN3K	4	$6.31 \times 10^1$	NN3K	40	3.94	NN3K	1270	$3.08 \times 10^{-2}$

### Tables of Tree-NN0 model

- Split = Number of splits on dataset (and number of NN0 used)
- $\epsilon$  = Max of error max of each NN0
- $\mu$  = Mean of max error of each NN0
- SpaceOVH = space of the model in KB with respect to dataset

NN0 file 3

Split	$\epsilon$	$\mu$	SpaceOVH
1	8	8.0	$4.96 \times 10^{-2}$
2	8	6.5	$9.92 \times 10^{-2}$
3	7	5.0	$1.49 \times 10^{-1}$
4	9	6.0	$1.98 \times 10^{-1}$
5	6	4.4	$2.48 \times 10^{-1}$
6	5	3.5	$2.98 \times 10^{-1}$
7	5	3.14	$3.47 \times 10^{-1}$
8	5	3.25	$3.97 \times 10^{-1}$
9	3	2.33	$4.46 \times 10^{-1}$
10	4	2.8	$4.96 \times 10^{-1}$
11	4	2.45	$5.45 \times 10^{-1}$
12	3	1.92	$5.95 \times 10^{-1}$
13	3	1.85	$6.45 \times 10^{-1}$
14	2	1.57	$6.94 \times 10^{-1}$
15	4	1.67	$7.44 \times 10^{-1}$
16	2	1.19	$7.93 \times 10^{-1}$
17	2	1.12	$8.43 \times 10^{-1}$
18	2	1.17	$8.93 \times 10^{-1}$
19	4	1.16	$9.42 \times 10^{-1}$
20	1	1.0	$9.92 \times 10^{-1}$
21	2	1.1	1.04
22	2	1.05	1.09
26	1	1.0	1.29
30	2	1.07	1.49

NN0 file 7

Split	$\epsilon$	$\mu$	SpaceOVH
1	41	41.0	$3.1 \times 10^{-3}$
2	36	29.0	$6.2 \times 10^{-3}$
3	33	27.67	$9.3 \times 10^{-3}$
4	43	30.5	$1.24 \times 10^{-2}$
5	27	20.2	$1.55 \times 10^{-2}$
6	31	22.17	$1.86 \times 10^{-2}$
7	32	19.86	$2.17 \times 10^{-2}$
8	29	21.5	$2.48 \times 10^{-2}$
9	33	19.0	$2.79 \times 10^{-2}$
10	24	15.6	$3.1 \times 10^{-2}$
11	25	15.09	$3.41 \times 10^{-2}$
12	28	14.92	$3.72 \times 10^{-2}$
13	24	14.46	$4.03 \times 10^{-2}$
14	26	13.64	$4.34 \times 10^{-2}$
15	24	12.4	$4.65 \times 10^{-2}$
16	21	13.25	$4.96 \times 10^{-2}$
17	23	12.65	$5.27 \times 10^{-2}$
18	21	12.44	$5.58 \times 10^{-2}$
19	19	10.79	$5.89 \times 10^{-2}$
20	27	11.55	$6.2 \times 10^{-2}$
21	20	11.05	$6.51 \times 10^{-2}$
22	20	10.77	$6.82 \times 10^{-2}$
26	13	9.35	$8.06 \times 10^{-2}$
30	17	9.3	$9.3 \times 10^{-2}$
34	16	8.53	$1.05 \times 10^{-1}$
38	13	7.39	$1.18 \times 10^{-1}$
42	11	7.14	$1.3 \times 10^{-1}$
46	13	7.13	$1.43 \times 10^{-1}$
50	12	6.8	$1.55 \times 10^{-1}$
54	13	6.57	$1.67 \times 10^{-1}$
58	12	6.24	$1.8 \times 10^{-1}$
62	11	6.02	$1.92 \times 10^{-1}$
64	21	6.0	$1.98 \times 10^{-1}$
72	9	5.24	$2.23 \times 10^{-1}$
80	9	4.92	$2.48 \times 10^{-1}$
88	10	4.65	$2.73 \times 10^{-1}$
96	9	4.38	$2.98 \times 10^{-1}$
104	8	4.13	$3.22 \times 10^{-1}$
112	7	3.83	$3.47 \times 10^{-1}$
120	10	3.75	$3.72 \times 10^{-1}$
128	6	3.52	$3.97 \times 10^{-1}$

NN0 file 10

Split	$\epsilon$	$\mu$	SpaceOVH
1	714	714.0	$2.42 \times 10^{-5}$
2	593	557.5	$4.84 \times 10^{-5}$
3	494	420.33	$7.26 \times 10^{-5}$
4	515	428.75	$9.69 \times 10^{-5}$
5	545	360.0	$1.21 \times 10^{-4}$
6	561	319.17	$1.45 \times 10^{-4}$
7	538	292.57	$1.69 \times 10^{-4}$
8	465	298.5	$1.94 \times 10^{-4}$
9	453	240.33	$2.18 \times 10^{-4}$
10	425	228.6	$2.42 \times 10^{-4}$
11	338	222.64	$2.66 \times 10^{-4}$
12	311	200.08	$2.91 \times 10^{-4}$
13	207	183.85	$3.15 \times 10^{-4}$
14	238	175.07	$3.39 \times 10^{-4}$
15	254	176.53	$3.63 \times 10^{-4}$
16	305	196.75	$3.87 \times 10^{-4}$
17	266	169.53	$4.12 \times 10^{-4}$
18	246	154.06	$4.36 \times 10^{-4}$
19	193	147.79	$4.6 \times 10^{-4}$
20	233	144.3	$4.84 \times 10^{-4}$
21	277	150.67	$5.08 \times 10^{-4}$
22	357	143.0	$5.33 \times 10^{-4}$
26	209	131.85	$6.3 \times 10^{-4}$
30	233	116.7	$7.26 \times 10^{-4}$
34	196	106.15	$8.23 \times 10^{-4}$
38	177	104.55	$9.2 \times 10^{-4}$
42	214	101.38	$1.02 \times 10^{-3}$
46	153	88.57	$1.11 \times 10^{-3}$
50	143	84.88	$1.21 \times 10^{-3}$
54	145	85.07	$1.31 \times 10^{-3}$
58	133	82.4	$1.4 \times 10^{-3}$
62	143	80.34	$1.5 \times 10^{-3}$
64	142	79.77	$1.55 \times 10^{-3}$
72	121	69.71	$1.74 \times 10^{-3}$
80	134	66.75	$1.94 \times 10^{-3}$
88	153	65.88	$2.13 \times 10^{-3}$
96	109	60.6	$2.32 \times 10^{-3}$
104	107	59.13	$2.52 \times 10^{-3}$
112	99	56.11	$2.71 \times 10^{-3}$
120	118	54.53	$2.91 \times 10^{-3}$
128	302	56.22	$3.1 \times 10^{-3}$