

Deep Neural Networks for Learned Indexes Data Structures

Ettore Tancredi Galante

January 2019

Introduction

Neural Network for a Uniform Distribution

Bozza 1

Sia X una variabile aleatoria distribuita uniformemente su U , possiamo dire che:

$$P\{X = x\} = \frac{1}{|X|} = \frac{1}{n}; \text{ con } x \in X$$
$$P\{X \leq x\} = \hat{F}(x) = \frac{I(x)}{n}$$

Possiamo quindi dire che: $P\{X \leq x\} = F(x)$ e che tale $F(x)$ è stimata da $\hat{F}(x)$.

0.1 Esperimento 1

Generiamo $X \subseteq Z$ tale che X sia uniformemente distribuito e che i suoi elementi siano equispaziati tra loro di un intervallo $l \in \mathbb{N}$.

Per stimare $F(x)$, necessitiamo di apprendere una rete feed forward il cui output sia $\hat{F}(x)$.

La rete avrà uno strato di input composto da k neuroni, con:

$$k = \log_2 n;$$

$$n = \max(X).$$

Lo strato nascosto di neuroni ha dimensione arbitraria, tipo Dense e funzione di attivazione Sigmoidale, come nel singolo neurone di output.

Questo neurone restituirà $\hat{F}(x)$.

La funzione di loss è definita dal Mean Squared Error.

La funzione sigmoide, come la funzione tangente iperbolica, dimostra di restituire valori troppo compressi, impedendo di ottenere predizioni accurate. N.B.: Allega grafici con sigmoide e tanh. L'idea adesso è definire una funzione sigmoide custom e sperimentare con valori di beta differenti da 1, minori o maggiori che

siano. N.B.: Allegare grafici con tale funzione con valori di beta diversi. N.B.: Allegare grafici del mae e della loss della rete. Modificare gli strati di neuroni nascosti.