Random Vector Functional Link Nets

and Extreme Learning Machines.

POLISH NATIONAL AGENCY FOR ACADEMIC EXCHANGE

Marcos Eduardo Valle
Universidade Estadual de Campinas (Unicamp)
Campinas, Brazil.

Introduction

Huang et al. (2006) has coined the name extreme learning machines (ELMs) to a class of single hidden-layer networks.

An ELM is designed by randomly initializing the parameters of the hidden layer and adjusting the output layer using least squares.

Despite having more than 6,000 citations in the Web of Science, the main idea behind ELMs has been introduced and formalized by Igelnik and Pao (1995); Pao et al. (1994), known as random vector functional link networks (RVFL nets).

The RVFL nets are based on two concepts: An integral representation of a function and the Monte Carlo method.

The following is based on Husmeier (1999).

Monte Carlo Method

Monte Carlo methods aim to approximate the solution of problems using randomness.

They are handy for high-dimensional numerical integration as follows.

Consider the problem of estimating the volume of an m-dimensional hypersphere by numerical integration.

Let $\chi_S : \mathbb{R}^m \to \{0, 1\}$ be the indicator function of the hypersphere of radius $R > 0$. Formally, we have

$$\chi_S(\mathbf{x}) = \begin{cases} 1, & \|\mathbf{x}\| \leq R, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The volume of the hypersphere is

$$V_S = \int_{\mathbb{R}^m} \chi_S(\mathbf{x}) d\mathbf{x} = \int_K \chi_S(\mathbf{x}) dx = \frac{2\pi^{m/2} R^m}{m\Gamma(m/2)}, \tag{2}$$

where $K = [-R, R]^m$ is the smallest hypercube that contains the hypersphere.

Using a standard numerical method based on the Riemann integral, each side of the hypercube K is divided into k intervals of length $\ell = (2R)/k$.

As a consequence, the hypercube K is divided into $n = k^m$ equally-sized sub-cubes, each one with volume

$$\ell^m = \left(\frac{2R}{k}\right)^m = \frac{(2R)^m}{n}. \tag{3}$$

Finally, the volume of the hypersphere is approximated by

$$V_S^G = \sum_{i=1}^{n} \chi_S(\mathbf{x}_i) \frac{(2R)^m}{n} = (2R)^m \left(\frac{1}{n} \sum_{i=1}^{n} \chi_S(\mathbf{x}_i) \right), \tag{4}$$

where $\mathbf{x}_i$ is the center of the ith sub-cube.

The Monte Carlo method approximates V_s using random samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ instead of a regular grid.

Precisely, consider a uniform distribution on K given by

$$\mathcal{P}(\mathbf{x}) = \frac{1}{(2R)^m} \chi_K(\mathbf{x}), \tag{5}$$

where $\chi_K : \mathbb{R}^m \to \mathbb{R}$ is the indicator function of the hypercube K.

The volume of the hypersphere is approximated by

$$V_S^{MC} = (2R)^m \left(\frac{1}{n} \sum_{i=1}^{n} \chi_S(\mathbf{x}_i) \right), \tag{6}$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are selected independently using the uniform distribution.

Note that V_S^G and V_S^{MC} have the same expression and differ only on the samples $\mathbf{x}_i$'s (grid versus random sample).

However, the two approximation methods differ significantly as m and n increases.

The following table contains the relative error given by

$$\mathcal{E}_r = \frac{100}{V_s} |V_S^X - V_S|, \quad V_S^X \in \{V_S^G, V_S^{MC}\}. \tag{7}$$

Dimension (m)	# Samples (n)	Grid $\mathcal{E}_r$	MC $\mathcal{E}_r$
3	$27 (= 3^3)$	34.4	12.8
3	$125 (= 5^3)$	23.8	7.8
3	$1000 (= 10^3)$	5.4	2.8
10	$1024 (= 2^{10})$	100.0	38.5
10	$59049 (= 3^{10})$	36.7	4.9

Source: Husmeier (1999).

Random Vector Functional Link (RVFL) Networks

A single hidden-layer network defines a function $\tilde{f}_n : \mathbb{R}^m \to \mathbb{R}$ using the equation

$$\tilde{f}_n(\mathbf{x}) = \sum_{i=1}^{n} w_i g(\mathbf{u}_i^T \mathbf{x} - b_i), \tag{8}$$

where $g : \mathbb{R} \to \mathbb{R}$ is the activation or transfer function and n is the number of neurons in the hidden layer.

In an RVFL network, the parameters of the hidden layer – the weights $\mathbf{u}_i \in \mathbb{R}^m$ and the bias $b_i \in \mathbb{R}$ – are selected randomly and independently in advance.

The weights w_i's are determined using least squares (or, eventually, using logistic regression or softmax regression in classification problems).

Let $K \subset \mathbb{R}^m$ be a compact set, and let $f : K \to \mathbb{R}$ be a continuous function on K.

Let the transfer function $g : \mathbb{R} \to \mathbb{R}$ be a bounded (for convenience, we assume that $|g(t)| \leq 1$ for all $t \in \mathbb{R}$) and differentiable function whose derivative is square integrable, that is,

$$\int_{\mathbb{R}} (g'(t))^2 dt < +\infty. \tag{9}$$

Note that the logistic function $\sigma(t) = 1/(1 + e^{-t})$ and `tanh` satisfies these conditions.

Integral Representation f

The function $f : K \subset \mathbb{R}^m \to \mathbb{R}$ satisfies the following identity (Murata, 1996)

$$f(\mathbf{x}) = \int_{\mathbb{R}^{m+1}} T(\mathbf{u}, b) g(\mathbf{u}^T \mathbf{x} + b) d\mathbf{u} db, \quad \forall \mathbf{x} \in K, \tag{10}$$

where the transform $T : \mathbb{R}^{m+1} \to \mathbb{R}$ is given by

$$T(\mathbf{u}, b) \propto \int_{\mathbb{R}^m} \breve{g}(\mathbf{u}^T \mathbf{x} - b) f(\mathbf{x}) d\mathbf{x}, \quad \forall \mathbf{u} \in \mathbb{R}^m \quad \text{and} \quad b \in \mathbb{R}. \tag{11}$$

Here, the symbol "$\propto$" means that $T(\mathbf{u}, b)$ is proportional to the integral on the right, and $\breve{g}$ is a kind of conjugate of g.

Formally, g and $\breve{g}$ must satisfy the conditions

$$\breve{G}^*(-w)G(-w) = \breve{G}^*(w)G(w), \tag{12}$$

$$\int_0^\infty \frac{1}{w^m} |\breve{G}^*(w)G(w)| dw < \infty, \tag{13}$$

and

$$\int_0^\infty \frac{1}{w^m} \breve{G}^*(w)G(w) dw \neq 0, \tag{14}$$

where $G = \mathcal{F}\{g\}$ and $\breve{G} = \mathcal{F}\{\breve{g}\}$ denote the Fourier transform of g and $\breve{g}$, respectively, and $\breve{G}^*(w)$ denotes the complex conjugate of $\breve{G}(w)$.

Remark: The integral in (14) appears multiplying the integral on the right-hand side of (11).

Approximation of T

First, let us constrain the domain of the integral in (11) from $\mathbb{R}^{m+1}$ to the hypercube $H = [-R, R]^{m+1}$.

Precisely, let f_R be the function defined by

$$f_R(\mathbf{x}) := \int_H T(\mathbf{u}, b) g(\mathbf{u}^T \mathbf{x} - b) d\mathbf{u} db. \tag{15}$$

Note that

$$f(\mathbf{x}) = \lim_{R \to \infty} f_R(\mathbf{x}). \tag{16}$$

Furthermore, let us approximate the integral in (15) using the Monte Carlo method.

Formally, define the function $\tilde{f}_n$ by means of the equation

$$\tilde{f}_n(\mathbf{x}) = \frac{(2R)^{m+1}}{n} \sum_{i=1}^{n} T(\mathbf{u}_i, b_i) g(\mathbf{u}_i^T \mathbf{x} - b_i), \tag{17}$$

where $(\mathbf{u}_1, b_1), \ldots, (\mathbf{u}_n, b_n)$ is a sample of size n drawn independently from a uniform distribution in $H = [-R, R]^{m+1}$.

Note that we obtain a single hidden-layer network

$$\tilde{f}_n(\mathbf{x}) = \sum_{i=1}^{n} w_i g(\mathbf{u}_i^T \mathbf{x} - b_i), \tag{18}$$

by setting

$$w_i = \frac{(2R)^{m+1}}{n} T(\mathbf{u}_i, b_i). \tag{19}$$

Let us define

$$d[f, \tilde{f}_n] = \sqrt{\frac{1}{|K|}\mathbb{E}\left\{\int_K (f(\mathbf{x}) - \tilde{f}_n(\mathbf{x}))^2 d\mathbf{x}\right\}}, \tag{20}$$

where $|K|$ denotes the volume of K and $\mathbb{E}\{\cdot\}$ denotes the expectation value with respect to the uniform probability distribution in $H = [-R, R]^{m+1}$.

Using the distance d, we obtain the inequality (Husmeier, 1999)

$$d[f, f_n] \leq \sup_{\mathbf{x}\in K} |f(\mathbf{x}) - f_R(\mathbf{x})| + d[f_R, f_n]. \tag{21}$$

On the one hand, the first term on the right rand-side of (21) can be made arbitrarily small by choosing large enough R. On the other hand, the second term becomes very large as $R \to \infty$.

We can overcome this dilemma by assuming f is Lipshitz continuous, that is,

$$\exists \kappa > 0 : |f(\mathbf{x}) - f(\mathbf{y})| \leq \kappa \sum_{i=1}^{m} |x_i - y_i|, \quad \forall \mathbf{x}, \mathbf{y} \in K. \tag{22}$$

In this case, the first term becomes negligibly for finite R, and we obtain the inequality

$$d[f, \tilde{f}_n] \leq \frac{C_{RVFL}}{\sqrt{n}}, \quad C_{RVFL}^2 = |H| \int_H T^2(\mathbf{u}, b) d\mathbf{u} db, \tag{23}$$

where $|H| = (2R)^{m+1}$ is the volume of the hypercube H.

In a similar fashion but considering a probability distribution that considers information about the function f (we can think of fine-tuning the parameters $\mathbf{u}_i$'s and b_i's), we obtain

$$d[f, \tilde{f}_n] \leq \frac{C_{MLP}}{\sqrt{n}}, \quad C_{MLP} = \int_{\mathbb{R}^{m+1}} T(\mathbf{u}, b) d\mathbf{u} db. \tag{24}$$

Furthermore, we have

$$C_{RVFL}^2 - C_{MLP}^2 = |H|^2 \mathrm{Var}|T(\mathbf{u}, b)| \geq 0, \tag{25}$$

which implies that

$$C_{RVFL} \geq C_{MLP}. \tag{26}$$

Thus, fine-tuning the hidden layer parameters gives a closer approximation to f than the RVFL model for a given n.

However, the approximation error is $\propto 1/\sqrt{n}$ in both cases.

Concluding

Extreme learning machines (ELMs) are equivalent to random vector functional link (RVFL) networks.

RVFL networks are obtained first using an integral approximation of function f and then using the Monte Carlo method to approximate the integral.

As expected, fine-tuning the hidden layer parameters gives a closer approximation to f than the RVFL model for a given n.

However, the approximation error is $\propto 1/\sqrt{n}$, where n is the number of hidden units, either finite-tuning or using a random initialization of the hidden layer parameters.

Thanks for your attention!

Acknowledge

These slides are part of a mini-course given during the workshop on **hypercomplex-valued neural networks**, which took place at the **Cracow University of Technology**, Cracow, Poland, *25–30 September 2023*, with the support of

NAWA POLISH NATIONAL AGENCY FOR ACADEMIC EXCHANGE

My research on hypercomplex-valued neural networks has been partially supported by:

FAPESP and CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

References (1)

G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

D. Husmeier. *Neural Networks for Conditional Probability Estimation*. Springer London, London, 1999. ISBN 978-1-85233-095-8. doi: 10.1007/978-1-4471-0847-4.

B. Igelnik and Y.-H. Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Transactions on Neural Networks*, 6(6):1320–1329, 1995. ISSN 10459227. doi: 10.1109/72.471375.

N. Murata. An Integral Representation of Functions Using Three-layered Networks and Their Approximation Bounds. *Neural Networks*, 9(6):947–956, 8 1996. ISSN 0893-6080. doi: 10.1016/0893-6080(96)00000-7.

References (2)

Y. H. Pao, G. H. Park, and D. J. Sobajic. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2):163–180, 4 1994. ISSN 0925-2312. doi: 10.1016/0925-2312(94)90053-1.