

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/382298196>

# Homotopy Perturbation Neural Networks

**Preprint** · July 2024

DOI: 10.13140/RG.2.2.16035.46885/1

---

CITATIONS

0

---

READS

7

**1 author:**



**Gary Nan Tie**

Mu Risk LLC

**77 PUBLICATIONS** **26 CITATIONS**

[SEE PROFILE](#)

# Homotopy Perturbation Neural Networks

Gary Nan Tie, July 15th 2024

**Abstract:** Inspired by He's homotopy perturbation method, we introduce a new deep learning technique to parsimoniously solve regression problems. Essentially, a homotopy is constructed that deforms a linear problem to our desired non-linear regression problem.

For reference, we recall the homotopy perturbation method [He, 1999] for solving general non-linear differential equations of the form:

$$(1) \quad A(u) + f(r) = 0, \quad r \in \Omega$$

with boundary conditions

$$B(u, \frac{\partial u}{\partial n}) = 0, \quad r \in \Gamma$$

where  $A$  is a general differential operator,

$B$  is a boundary operator,  $f(r)$  is a known analytic function,

$\Gamma$  is the boundary of the domain  $\Omega$ .

Suppose  $A = L + N$ , where  $L$  is linear and  $N$  is non-linear.

So (1) becomes: (2)  $L(u) + N(u) - f(r) = 0$

We construct a homotopy  $v(r, p) : \Omega \times [0, 1] \rightarrow \mathbb{R}$

which satisfies: (3)  $H(v, p) := (1-p)[L(v) - L(u_0)]$

$$+ p[A(v) - f(r)] = 0$$

where  $p \in [0, 1]$ ,  $r \in \Omega$  and  $u_0$  is an initial approximation of (2), which satisfies the boundary conditions.

Note that,  $H(v, 0) = L(v) - L(u_0) = 0$

and  $H(v, 1) = A(v) - f(r) = 0$ .

Now use the embedding parameter  $p$  as a 'small parameter'

and assume that the solution of (3) can be written

as a power series in  $p$ : 
$$v = \sum_{i=0}^{\infty} p^i v_i$$

Setting  $p=1$  results in an approximate solution of (2):

$$(4) \quad u = \sum_{i=0}^{\infty} v_i$$

The series in (4) is convergent in most cases, and the rate of convergence depends on  $A(v)$ .

The homotopy perturbation method motivates the following:

given training data pairs  $(x_i, y_i)$   $i=0, 1, \dots, n$

we want to learn a regression function  $A(x) = y$ .

Let  $L_\varphi$  be a linear  $L$ -layer FFN (feedforward network)

with parameters  $\varphi$  and identity activation function, and

let  $N_\gamma$  be a non-linear  $M$ -layer FFN with

parameters  $\gamma$  and activation function  $\sigma \neq \text{identity}$ .

Suppose  $X = \{x_1, x_2, \dots, x_n\} \subset \mathcal{X} \subseteq \mathbb{R}^m$

define homotopy by: for  $j=0, 1, \dots, k$

$$H_j : X \longrightarrow \mathbb{R}$$

$$x_i \longmapsto (1-p_j) [L_\varphi(x_i) - L_\varphi(x_0)]$$

$$+ p_j [L_\varphi(x_i) + N_\gamma(x_i) - y_i]$$

where  $p_j \triangleq \frac{j}{k}$ ,  $k \geq 2$ ; a discrete deformation

from  $L(x_i) - L(x_0)$  to  $L(x_i) + N(x_i) - y_i$ .

On average we want  $H_j(x_i) = 0$  with small dispersion.

$$\text{So let } V_j \triangleq \sum_{i=1}^n |H_j(x_i)|^2$$

$$\text{and } L(\varphi, \eta) \triangleq \sum_{j=0}^k V_j + \lambda [L(x_0) + N(x_0) - y_0]^2, \lambda > 0$$

$$\text{and } L(\hat{\varphi}, \hat{\eta}) \triangleq \min_{\varphi, \eta} L(\varphi, \eta).$$

$$\text{Then } L_{\hat{\varphi}}(x_i) + N_{\hat{\eta}}(x_i) \approx y_i, \quad i = 1, 2, \dots, n$$

$$\text{since } H_k(x_i) \approx 0.$$

$$\text{Define regression function } A(x) = L_{\hat{\varphi}}(x) + N_{\hat{\eta}}(x), \quad x \in \mathcal{X}.$$

Summary: We learn  $\hat{\varphi}$  and  $\hat{\eta}$  so that deformation

$$H_j \text{ from } L(x_i) - L(x_0) \text{ to } A(x_i) - y_i$$

$$\text{satisfies } H_j(x_i) \approx 0 \text{ for } j = 0, 1, \dots, k.$$

$$\text{In particular, } H_k(x_i) \approx 0, \text{ i.e. } L_{\hat{\varphi}}(x_i) + N_{\hat{\eta}}(x_i) \approx y_i, \quad i = 1, \dots, n$$

so define our regression function to be

$$A(x) \triangleq L_{\hat{\varphi}}(x) + N_{\hat{\eta}}(x) \text{ for } x \in \mathcal{X} \supset X.$$

### Refining the initial guess $x_0$

Let  $X = \{x_0, x_1, \dots, x_n\}$  and  $Y = \{y_0, y_1, \dots, y_n\}$

For  $j = 0, 1, \dots, k$  with  $k \geq 2$ ,

$$\begin{aligned} \text{let } H_j(x_i) \triangleq & (1 - p_j) [L_\varphi(x_i) - L_\varphi(x_0)] \\ & + p_j [L_\varphi(x_i) + N_{\hat{\varphi}}(x_i) - y_i] \end{aligned}$$

where  $p_j \triangleq \frac{j}{k}$  and  $x_i \in X \setminus \{x_0\}$ ,

$$\text{and let } V_j \triangleq \sum_{x_i \in X \setminus \{x_0\}} |H_j(x_i)|^2.$$

$$l(x_0, \varphi, \psi, \lambda) \triangleq \sum_{j=0}^k V_j + \lambda [L_\varphi(x_0) + N_\psi(x_0) - y_0]^2, \quad \lambda > 0$$

$$l(x_0, \hat{\varphi}, \hat{\psi}, \lambda) \triangleq \min_{\varphi, \psi} l(x_0, \varphi, \psi, \lambda)$$

$$A[x_0, \hat{\varphi}, \hat{\psi}, \lambda](x) \triangleq L_{\hat{\varphi}}(x) + N_{\hat{\psi}}(x), \quad x \in \mathcal{X} \supset X$$

$$x_{\hat{i}} \triangleq x_i \in X \text{ that minimizes } |A[x_0, \hat{\varphi}, \hat{\psi}, \lambda](x_i) - y_i|$$

$L(x_0, \hat{\varphi}, \hat{\psi}, \lambda)$  learns  $\hat{\varphi}$  and  $\hat{\psi}$

over  $X$  and  $Y$  with initial guess  $x_0$ .

$L(x_{\hat{i}}, \hat{\varphi}^i, \hat{\psi}^i, \lambda)$  learns  $\hat{\varphi}^i$  and  $\hat{\psi}^i$

over  $X$  and  $Y$  with initial guess  $x_{\hat{i}}$ .

If  $L(x_{\hat{i}}, \hat{\varphi}^i, \hat{\psi}^i, \lambda) < L(x_0, \hat{\varphi}, \hat{\psi}, \lambda)$

update initial guess  $x_0$  to  $x_{\hat{i}}$

and repeat this refinement until  $L(x_{\hat{i}}, \hat{\varphi}^i, \hat{\psi}^i, \lambda)$

is sufficiently small for purpose or stops.

On the final iteration, use  $A[x_{\hat{i}}, \hat{\varphi}^i, \hat{\psi}^i, \lambda]$

as the predictor function.



## References

Ji-Huan He, 'Homotopy perturbation technique'  
Comput. Methods Appl. Mech. Engrg., 178 (1999), pp. 257-262

Ji-Huan He, 'A coupling method of a homotopy technique and a  
perturbation technique for non-linear problems'  
International Journal of Non-Linear Mechanics 35 (2000) 37-43