Can neural networks be explained using polynomial regression and Taylor series?

Pablo Morala, Jenny Alexandra Cifuentes, Rosa E. Lillo, Iñaki Úcar

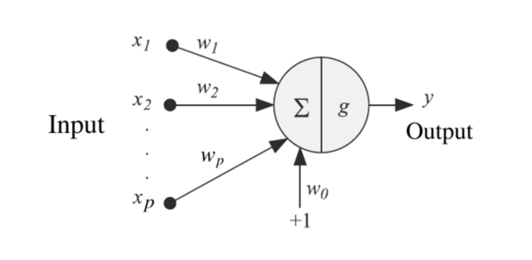
New Bridges between Mathematics and Data Science (NBMDS)
Nov 10th, 2021, Valladolid, Spain

Table of contents

Motivation

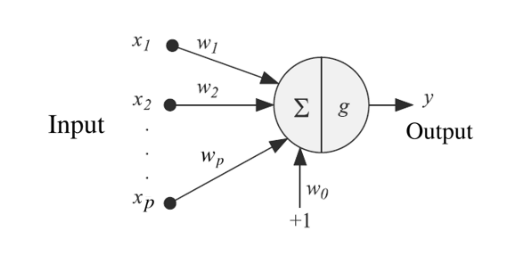
Motivation

- Neural Networks (NN) are one of the most widely used tools in Machine Learning.



Motivation

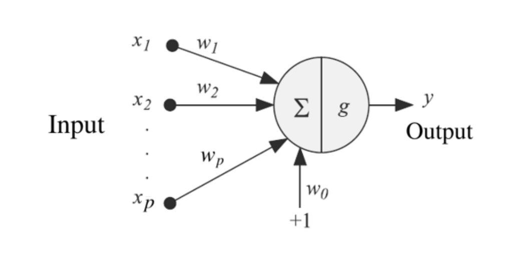
- Neural Networks (NN) are one of the most widely used tools in Machine Learning.



- However, NNs present some limitations:

Motivation

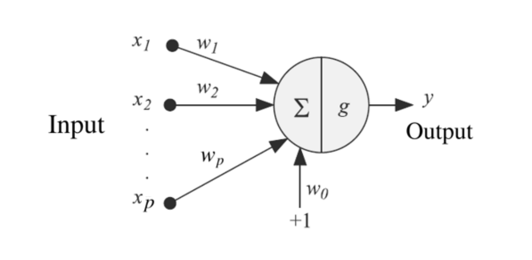
- Neural Networks (NN) are one of the most widely used tools in Machine Learning.



- However, NNs present some limitations:
  - Considered "black boxes".

Motivation

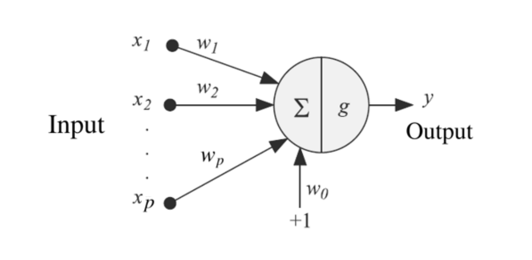
- Neural Networks (NN) are one of the most widely used tools in Machine Learning.



- However, NNs present some limitations:
  - Considered "black boxes".
  - Hyperparameter tuning.

Motivation

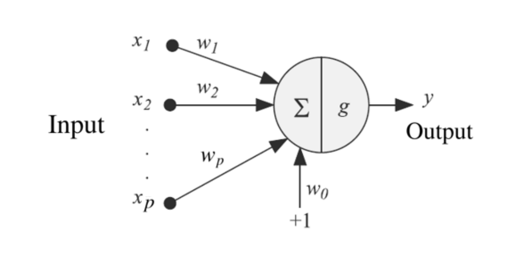
- Neural Networks (NN) are one of the most widely used tools in Machine Learning.



- However, NNs present some limitations:
  - Considered "black boxes".
  - Hyperparameter tuning.
  - Assessing the uncertainty and error in their predictions.

Motivation

- Neural Networks (NN) are one of the most widely used tools in Machine Learning.



- However, NNs present some limitations:
  - Considered "black boxes".
  - Hyperparameter tuning.
  - Assessing the uncertainty and error in their predictions.
- A new perspective: the authors of Cheng et al., 2019 conjectured that NNs are equivalent to Polynomial Regression (PR).

Motivation: Proposed equivalence in Cheng et al., 2019

Publication

Cheng, X., Khomtchouk, B., Matloff, N., & Mohanty, P. (2019). Polynomial Regression As an Alternative to Neural Nets. *arXiv:1806.06850 [cs, stat]*

They present the following ideas:

- NNs are a form of PR.
- The degree of the polynomial increases with each hidden layer.
- PR properties can be used to study and even solve NN problems.
- Experimental results: PR performs as good as NNs in the used datasets.
- Using PR instead of NNs: R package (polyreg).

Motivation: Proposed equivalence in Cheng et al., 2019

Publication

Cheng, X., Khomtchouk, B., Matloff, N., & Mohanty, P. (2019). Polynomial Regression As an Alternative to Neural Nets. *arXiv:1806.06850 [cs, stat]*

They present the following ideas:

- NNs are a form of PR.
- The degree of the polynomial increases with each hidden layer.
- PR properties can be used to study and even solve NN problems.
- Experimental results: PR performs as good as NNs in the used datasets.
- Using PR instead of NNs: R package (polyreg).

However...

The relation between PR and NNs is not explicitly proven.

Our contribution

Publication

Morala, P., Cifuentes, J. A., Lillo, R. E., & Ucar, I. (2021). Towards a mathematical framework to inform neural network modelling via polynomial regression. *Neural Networks*, *142*, 57–72. https://doi.org/10.1016/j.neunet.2021.04.036

Our contribution

Publication

Morala, P., Cifuentes, J. A., Lillo, R. E., & Ucar, I. (2021). Towards a mathematical framework to inform neural network modelling via polynomial regression. *Neural Networks*, *142*, 57–72. https://doi.org/10.1016/j.neunet.2021.04.036

Main objectives:

- Find an explicit expression to build a PR from the weights of a given NN, using Taylor expansion.
  - First, for a single hidden layer NN, then extending it to deeper layers.
- Study through simulations the validity of the proposed method.

Proposed method

Notation: Polynomial Regression

- Polynomial Regression:

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \cdots + \\ \beta_{11} x_1^2 + \beta_{12} x_1 x_2 + \cdots + \beta_{1p\ldots p} x_1 x_p^{k-1} + \beta_{p\ldots p} x_p^k.$$

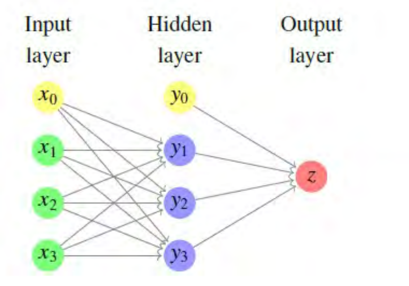
Notation: Neural Network

- Hidden layer neurons:

$$y_j = g\left(\sum_{i=0}^{p} w_{i,j} x_i\right).$$

- Final output:

$$z = \sum_{j=0}^{h_1} v_j y_j = \sum_{j=0}^{h_1} v_j g\left(\sum_{i=0}^{p} w_{i,j} x_i\right).$$



Proposed method: Necessary tools

Taylor expansion

$$y_j = g\left(\sum_{i=0}^{p}(w_{i,j}x_i)\right) = \sum_{n=0}^{\infty}\frac{g^{(n)}(a)}{n!}\left(\sum_{i=0}^{p}(w_{i,j}x_i) - a\right)^n$$

↓

Binomial Theorem

$$\left(\sum_{i=0}^{p} w_{i,j}x_i - a\right)^n = \sum_{k=0}^{n}\binom{n}{k}(-a)^{n-k}\left(\sum_{i=0}^{p}(w_{i,j}x_i)\right)^k$$

↓

Multinomial Theorem

$$\left(\sum_{i=0}^{p} w_{i,j}x_i\right)^k = \sum_{m_0+\cdots+m_p=k}\binom{k}{m_0,\ldots,m_p}(w_{0,j}x_0)^{m_0}\cdots(w_{p,j}x_p)^{m_p}$$

Proposed Method: Building the coefficients formula

- Combining all of the previous steps to obtain the output of the NN, setting the Taylor expansion at $a = 0$ and truncating the series at a given degree q:

$$z = v_0 + \sum_{j=1}^{h_1} v_j \sum_{n=0}^{q} \frac{g^{(n)}(0)}{n!} \times$$

$$\times \left[\sum_{m_0+\cdots+m_p=n} \binom{n}{m_0, \ldots, m_p} (w_{0,j}x_0)^{m_0} \cdots (w_{p,j}x_p)^{m_p} \right]$$

Proposed Method: Building the coefficients formula

The following PR coefficients are obtained:

- Intercept:

$$\beta_0 = v_0 + \sum_{j=1}^{h_1} v_j \left(\sum_{n=0}^{q} \frac{g^{(n)}(0)}{n!} (w_{0,j})^n \right)$$

- Rest of the coefficients:

$$\beta_{l_1 l_2 \ldots l_t} = \sum_{j=1}^{h_1} v_j \left(\sum_{n=t}^{q} \frac{g^{(n)}(0)}{(n-t)! \cdot m_1! \cdots m_p!} (w_{0,j})^{n-t} (w_{1,j})^{m_1} \ldots (w_{p,j})^{m_p} \right)$$

Taylor expansion validity for some activation functions

Activation Functions considered:

We consider the three following activation functions:

Activation Functions considered:

We consider the three following activation functions:

- **Softplus or SmoothRelu:**

$$g(x) = \ln\left(1 + e^x\right)$$

Activation Functions considered:

We consider the three following activation functions:

- **Softplus or SmoothRelu:**

$$g(x) = \ln(1 + e^x)$$

- **Hyperbolic tangent:**

$$g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Activation Functions considered:

We consider the three following activation functions:

- **Softplus or SmoothRelu:**

$$g(x) = \ln\left(1 + e^{x}\right)$$

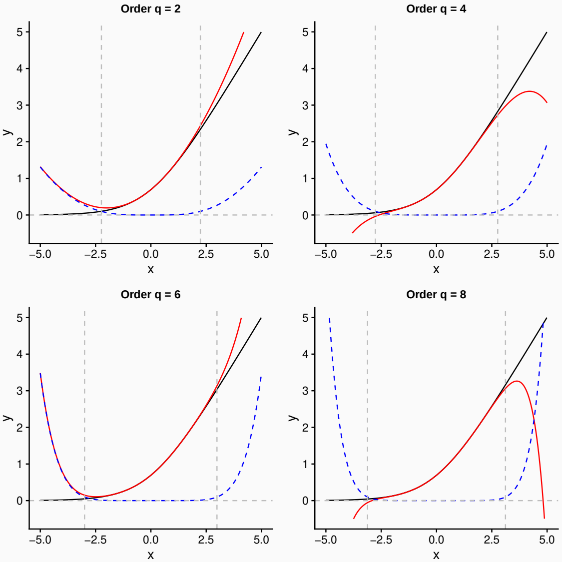
- **Hyperbolic tangent:**

$$g(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

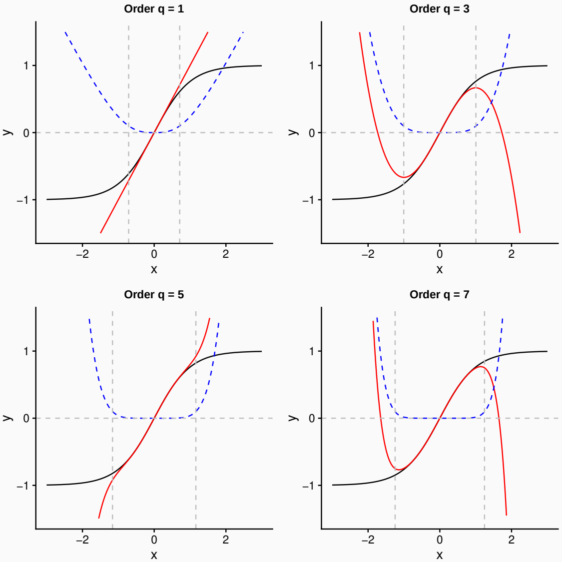
- **Sigmoid:**

$$g(x) = \frac{1}{1 + e^{-x}}$$

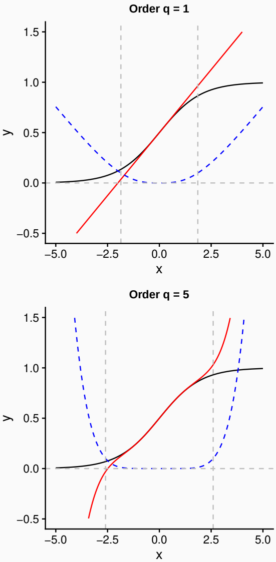
Taylor expansion: softplus



Taylor expansion: hyperbolic tangent

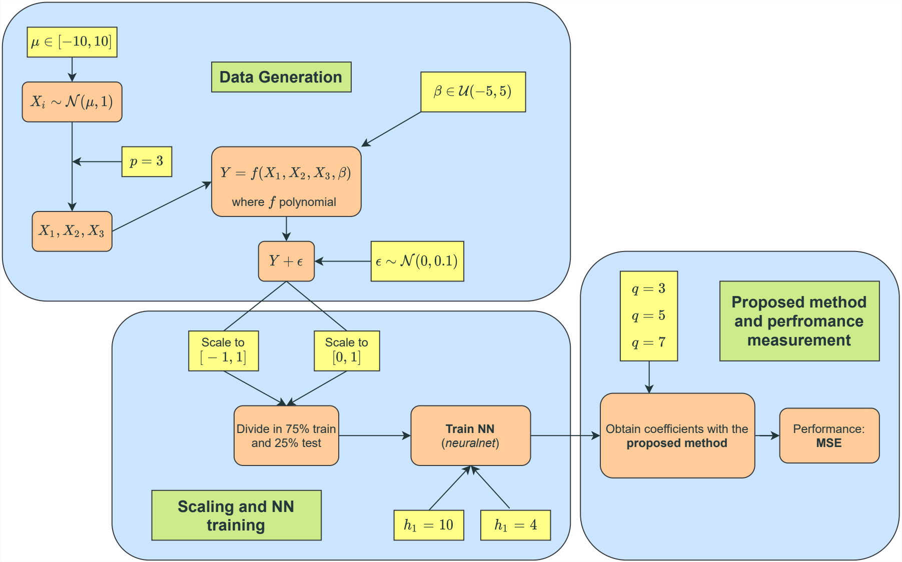


Taylor expansion: sigmoid

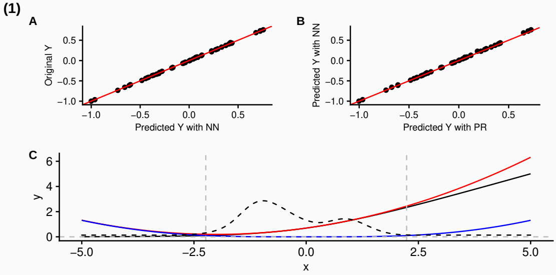


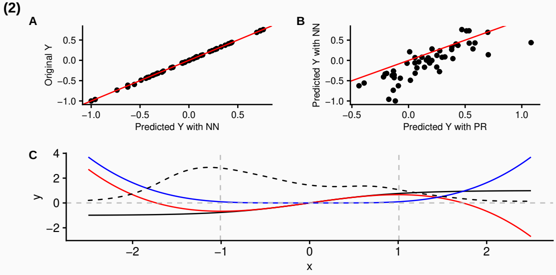
Simulation study (without restrictions)

Data generation

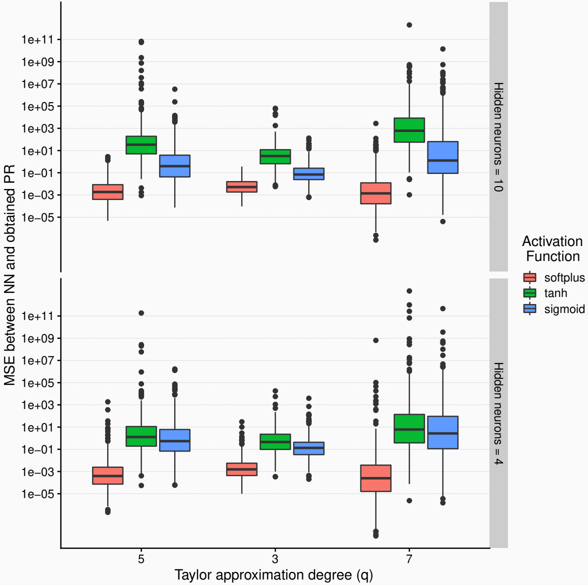


Performance examples

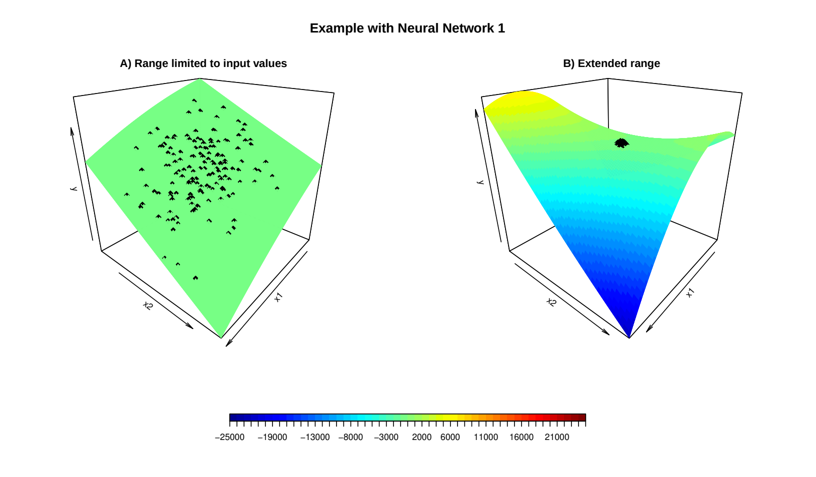




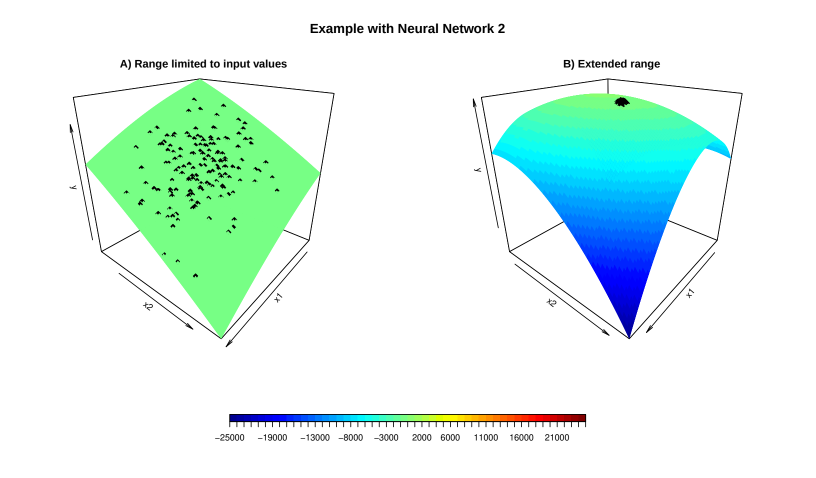
500 MSE simulations scaling to [-1,1]



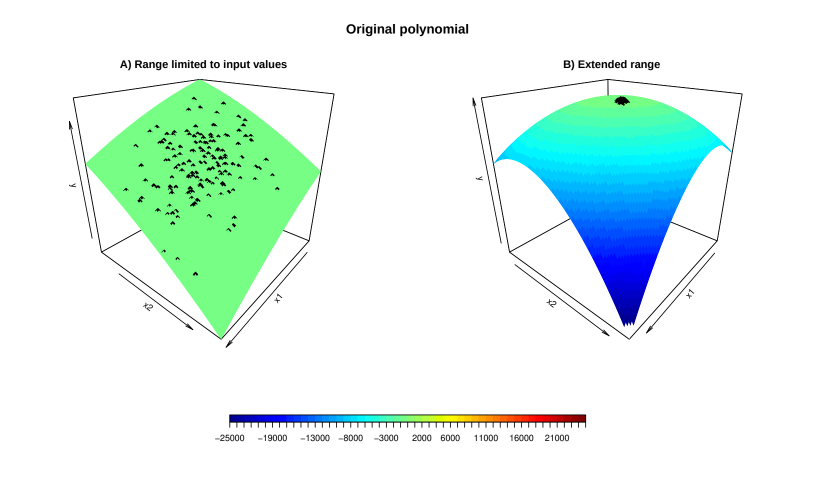
Same data, different NNs (1)



Same data, different NNs (2)



Same data, 4 different NNs: Original surface



Simulation study (with restriction)

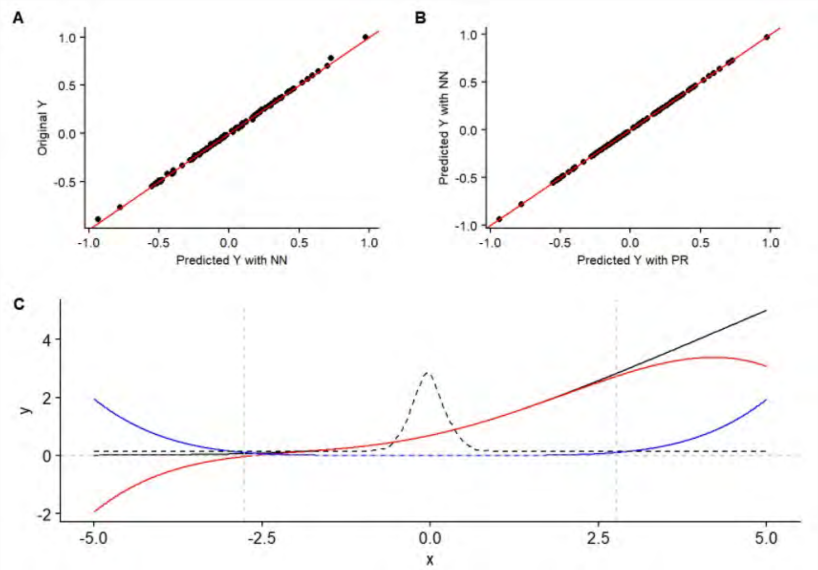
Constraining the weights in the hidden layers

- Limited region where Taylor expansion is accurate.
  - Impose an ℓ_1-norm equal to one for the hidden layer weights: $||\vec{w}_j||_1 = \sum_{i=0}^{p} |w_{i,j}| = 1$ for all j.
  - Then the synaptic potentials u_j are also constrained by 1 in absolute value:

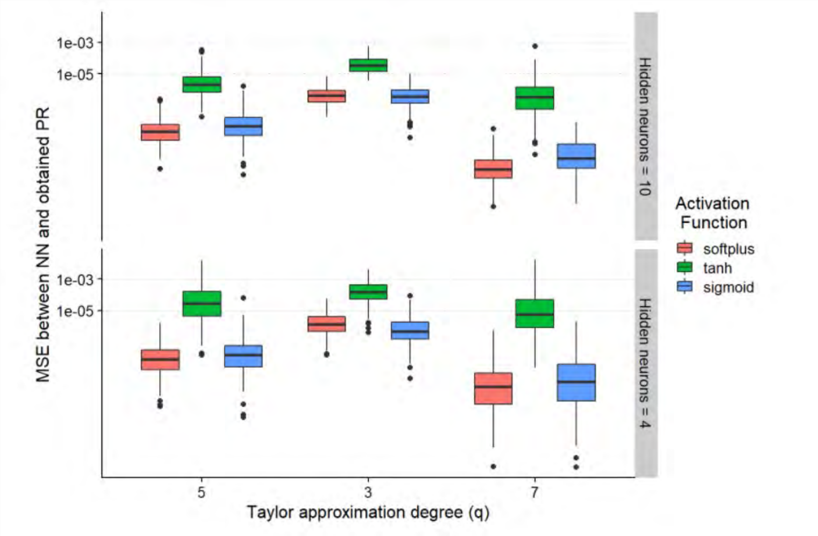
$$|u_j| = \left|\sum_{i=0}^{p} w_{i,j} x_i\right| \leq \sum_{i=0}^{p} |w_{i,j} x_i| \leq \sum_{i=0}^{p} |w_{i,j}| = ||\vec{w}_j||_1 = 1,$$

where $|x_i| \leq 1$ because of the $[-1, 1]$ scaling. Therefore, $|u_j| \leq 1$.

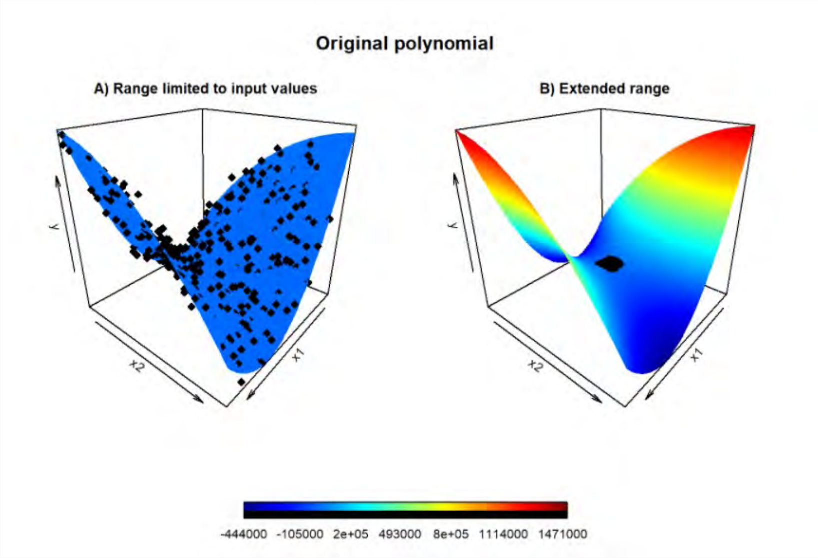
Example using weight constraints



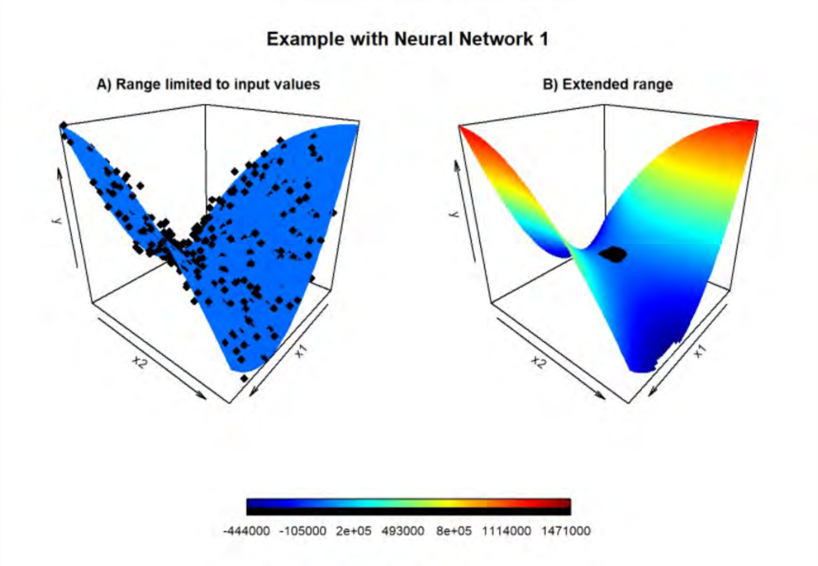
MSE simulations (restricted weights)



Surface example: Original Polynomial



Surface example: PR obtained from a NN



Conclusions and future work

Limitations and future work

Limitations and future work

- Limited to a single hidden layer regression:

Limitations and future work

- Limited to a single hidden layer regression:
  - Extension to deeper NN's.

Limitations and future work

- Limited to a single hidden layer regression:
  - Extension to deeper NN's.
  - Extension to classification problems, with more than one output and with a non linear AF in the output layer.

Limitations and future work

- Limited to a single hidden layer regression:
    - Extension to deeper NN's.
    - Extension to classification problems, with more than one output and with a non linear AF in the output layer.
- The validity of the method highly depends on the weights obtained in the NN and how they affect the Taylor expansion.

Limitations and future work

- Limited to a single hidden layer regression:
    - Extension to deeper NN's.
    - Extension to classification problems, with more than one output and with a non linear AF in the output layer.
- The validity of the method highly depends on the weights obtained in the NN and how they affect the Taylor expansion.
- Further extend the simulation study to different situations: higher dimension problems, correlation between variables or even real data examples, focusing on how this improves interpretability.

Limitations and future work

- Limited to a single hidden layer regression:
  - Extension to deeper NN's.
  - Extension to classification problems, with more than one output and with a non linear AF in the output layer.
- The validity of the method highly depends on the weights obtained in the NN and how they affect the Taylor expansion.
- Further extend the simulation study to different situations: higher dimension problems, correlation between variables or even real data examples, focusing on how this improves interpretability.
- Implement the proposed method in a package (R/Python).

Conclusions

- Our main contribution: Explicit formula to obtain a PR from a single hidden layer NN.

Conclusions

- Our main contribution: Explicit formula to obtain a PR from a single hidden layer NN.
- The work developed shows that it is possible to explore a mathematical framework relating PR with NN as was conjectured in Cheng et al., 2019.

Conclusions

- Our main contribution: Explicit formula to obtain a PR from a single hidden layer NN.
- The work developed shows that it is possible to explore a mathematical framework relating PR with NN as was conjectured in Cheng et al., 2019.
- Possible applications:

Conclusions

- Our main contribution: Explicit formula to obtain a PR from a single hidden layer NN.
- The work developed shows that it is possible to explore a mathematical framework relating PR with NN as was conjectured in Cheng et al., 2019.
- Possible applications:
  - NNs interpretability by means of the PR coefficients.

Conclusions

- Our main contribution: Explicit formula to obtain a PR from a single hidden layer NN.
- The work developed shows that it is possible to explore a mathematical framework relating PR with NN as was conjectured in Cheng et al., 2019.
- Possible applications:
    - NNs interpretability by means of the PR coefficients.
    - Exploring NNs hyperparameter and structure tuning using PR properties.

Conclusions

- Our main contribution: Explicit formula to obtain a PR from a single hidden layer NN.
- The work developed shows that it is possible to explore a mathematical framework relating PR with NN as was conjectured in Cheng et al., 2019.
- Possible applications:
  - NNs interpretability by means of the PR coefficients.
  - Exploring NNs hyperparameter and structure tuning using PR properties.
  - Model NNs error and uncertainty using PR.

Main References

1. Cheng, X., Khomtchouk, B., Matloff, N., & Mohanty, P. (2019). Polynomial Regression As an Alternative to Neural Nets. *arXiv:1806.06850 [cs, stat]*
2. Morala, P., Cifuentes, J. A., Lillo, R. E., & Ucar, I. (2021). Towards a mathematical framework to inform neural network modelling via polynomial regression. *Neural Networks*, *142*, 57–72. https://doi.org/10.1016/j.neunet.2021.04.036

Thanks for your attention!

Contact: pablo.morala@uc3m.es