A Surprising Connection: Neural Networks and Polynomial Regression

Norm Matloff
University of California at Davis

BARUG
presented at GRAIL June 19, 2018

These slides will be available at
`http://heather.cs.ucdavis.edu/polygrail.pdf`

Neural Networks

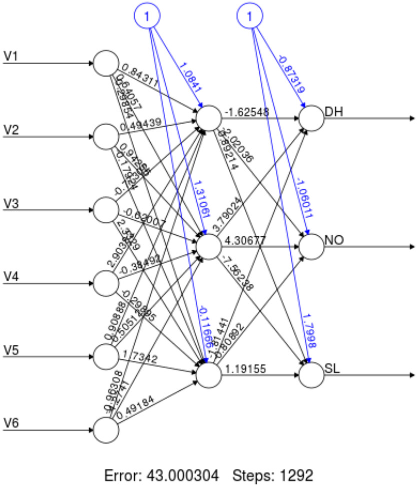
Neural Networks

- Series of *layers*, each consisting of *neurons*.
- First layer consists of the predictor variables.
- Each neuron has inputs from the previous layer.
- Each neuron has output: Linear combination of inputs, then fed through a nonlinear *activation function*.
- Final layer output: The prediction, either regression or classification.

Example

Example

UCI vertebrae data; predict one of 3 classes from 6 predictors.



History of NNs

History of NNs

- Treated largely as a curiosity through the 1990s.
- Then in the 2000s, "NN+" models won a number of major competitions, a huge boost to their popularity.
- But also many dismiss them as hype.
- Some say NNs work poorly on their data; others counter, "You're not using them right."

Contributions of Our Work

Contributions of Our Work

(a) Investigated relation of NNs to polynom. regression (PR).

Contributions of Our Work

(a) Investigated relation of NNs to polynom. regression (PR).

(b) We present an informal argument that NNs, in essence, actually are PR. Acronym: NNAEPR.

Contributions of Our Work

(a) Investigated relation of NNs to polynom. regression (PR).

(b) We present an informal argument that NNs, in essence, actually are PR. Acronym: NNAEPR.

(c) We use this to speculate and then confirm a surprising multicollinearity property of NNs.

Contributions of Our Work

(a) Investigated relation of NNs to polynom. regression (PR).

(b) We present an informal argument that NNs, in essence, actually are PR. Acronym: NNAEPR.

(c) We use this to speculate and then confirm a surprising multicollinearity property of NNs.

(d) NNAEPR suggests that one might simply fit a polynomial model in the first place, bypassing NNs.

Contributions of Our Work

(a) Investigated relation of NNs to polynom. regression (PR).

(b) We present an informal argument that NNs, in essence, actually are PR. Acronym: NNAEPR.

(c) We use this to speculate and then confirm a surprising multicollinearity property of NNs.

(d) NNAEPR suggests that one might simply fit a polynomial model in the first place, bypassing NNs.

(e) Thus avoid NN's problems, e.g. choosing tuning parameters, nonconvergence and so on.

Contributions of Our Work

(a) Investigated relation of NNs to polynom. regression (PR).

(b) We present an informal argument that NNs, in essence, actually are PR. Acronym: NNAEPR.

(c) We use this to speculate and then confirm a surprising multicollinearity property of NNs.

(d) NNAEPR suggests that one might simply fit a polynomial model in the first place, bypassing NNs.

(e) Thus avoid NN's problems, e.g. choosing tuning parameters, nonconvergence and so on.

(f) Tried many datasets. In all cases, **PR meets or beats NNs in predictive accuracy**.

Contributions of Our Work

(a) Investigated relation of NNs to polynom. regression (PR).

(b) We present an informal argument that NNs, in essence, actually are PR. Acronym: NNAEPR.

(c) We use this to speculate and then confirm a surprising multicollinearity property of NNs.

(d) NNAEPR suggests that one might simply fit a polynomial model in the first place, bypassing NNs.

(e) Thus avoid NN's problems, e.g. choosing tuning parameters, nonconvergence and so on.

(f) Tried many datasets. In all cases, **PR meets or beats NNs in predictive accuracy**.

(g) Developed many-featured R pkg., **polyreg**.

Notation and Acronyms

Notation and Acronyms

- n cases; p predictors
- polynomials of degree d
- *PR*: polynomial regression
- *NNAEPR* Neural Networks Are Essentially Polynomial Regression

polyreg

polyreg

- R package.

polyreg

- R package.
- Motivated by NNAEPR — use PR instead of NNs.

polyreg

- R package.
- Motivated by NNAEPR — use PR instead of NNs.
- Generates all possible d-degree polynomials in p variables.

polyreg

- R package.
- Motivated by NNAEPR — use PR instead of NNs.
- Generates all possible d-degree polynomials in p variables.
- Dimension reduction options.

polyreg

- R package.
- Motivated by NNAEPR — use PR instead of NNs.
- Generates all possible d-degree polynomials in p variables.
- Dimension reduction options.
- Functions for cross-validation comparison to various NN implementations.

polyreg

- R package.
- Motivated by NNAEPR — use PR instead of NNs.
- Generates all possible d-degree polynomials in p variables.
- Dimension reduction options.
- Functions for cross-validation comparison to various NN implementations.
- github.com/matloff/polyreg

polyreg

- R package.
- Motivated by NNAEPR — use PR instead of NNs.
- Generates all possible d-degree polynomials in p variables.
- Dimension reduction options.
- Functions for cross-validation comparison to various NN implementations.
- github.com/matloff/polyreg

NNAEPR

NNAEPR

- Consider toy example:

NNAEPR

- Consider toy example:
- Activation function $a(t) = t^2$.

NNAEPR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.

NNAEPR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.

NNAEPR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.
- Output of Layer 2 is all quartic $(d = 4)$ functions of u, v.

NNAEPR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.
- Output of Layer 2 is all quartic ($d = 4$) functions of u, v.
- Etc.

NNAEPR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.
- Output of Layer 2 is all quartic $(d = 4)$ functions of u, v.
- Etc.
- Polynomial regression!

NNAEPR

- Consider toy example:
- Activation function $a(t) = t^2$.
- Say $p = 2$ predictors, u and v.
- Output of Layer 1 is all quadratic functions of u, v.
- Output of Layer 2 is all quartic ($d = 4$) functions of u, v.
- Etc.
- Polynomial regression!
- **Important note:** The degree of the fitted polynomial in NN grows with each layer.

Norm Matloff
University of California at Davis

NNAEPR: General Activation Functions

NNAEPR: General Activation Functions

- Clearly this analysis for the toy activation function $a(t) = t^2$ extends to any polynomial activation function.
- But any reasonable activation function is “close” to a polynomial.

NNAEPR: General Activation Functions

- Clearly this analysis for the toy activation function $a(t) = t^2$ extends to any polynomial activation function.
- But any reasonable activation function is "close" to a polynomial.
  - E.g. Taylor approximation.
  - E.g. Stone-Weierstrass Theorem.
  - Etc.

NNAEPR: General Activation Functions

- Clearly this analysis for the toy activation function $a(t) = t^2$ extends to any polynomial activation function.
- But any reasonable activation function is "close" to a polynomial.
    - E.g. Taylor approximation.
    - E.g. Stone-Weierstrass Theorem.
    - Etc.
- Hence NNAEPR.

Disclaimer

Disclaimer

- We have not (yet) investigated the NNAEPR issue in the contexts of “NN+X”,

Disclaimer

- We have not (yet) investigated the NNAEPR issue in the contexts of "NN+X", e.g. CNNs (X = preprocessing of an image).

Disclaimer

- We have not (yet) investigated the NNAEPR issue in the contexts of "NN+X", e.g. CNNs (X = preprocessing of an image).
- We consider this an orthogonal issue to NNs. E.g. random forests versions of CNNs have been developed.

Disclaimer

- We have not (yet) investigated the NNAEPR issue in the contexts of "NN+X", e.g. CNNs (X = preprocessing of an image).
- We consider this an orthogonal issue to NNs. E.g. random forests versions of CNNs have been developed.
- But it is a topic of future research.

Implications of NNAEPR

Implications of NNAEPR

- Use our understanding of PR to gain insights into NNs.
- Heed the "advice" of NNAEPR, and use PR instead of NNs!

Norm Matloff
University of California at Davis

Multicollinearity in NNs

Multicollinearity in NNs

- Test of a good theory: Does it predict new phenomena?

Multicollinearity in NNs

- Test of a good theory: Does it predict new phenomena? E.g. Einstein "solar eclipse experiment."

Multicollinearity in NNs

- Test of a good theory: Does it predict new phenomena? E.g. Einstein “solar eclipse experiment.”
- PR is well known to be prone to multicollinearity.

Multicollinearity in NNs

- Test of a good theory: Does it predict new phenomena? E.g. Einstein "solar eclipse experiment."
- PR is well known to be prone to multicollinearity.
- The higher the degree in PR, the worse the multicollinearity.

Multicollinearity in NNs

- Test of a good theory: Does it predict new phenomena? E.g. Einstein "solar eclipse experiment."
- PR is well known to be prone to multicollinearity.
- The higher the degree in PR, the worse the multicollinearity.
- Thus NNAEPR predicts that **the outputs of the layers will have multicollinearity**, with each layer having great amounts of multicollinearity.

Multicollinearity in NNs

- Test of a good theory: Does it predict new phenomena? E.g. Einstein "solar eclipse experiment."
- PR is well known to be prone to multicollinearity.
- The higher the degree in PR, the worse the multicollinearity.
- Thus NNAEPR predicts that **the outputs of the layers will have multicollinearity**, with each layer having great amounts of multicollinearity.
- Is it true?

Multicollinearity in NNs

- Test of a good theory: Does it predict new phenomena? E.g. Einstein "solar eclipse experiment."
- PR is well known to be prone to multicollinearity.
- The higher the degree in PR, the worse the multicollinearity.
- Thus NNAEPR predicts that **the outputs of the layers will have multicollinearity**, with each layer having great amounts of multicollinearity.
- Is it true? Yes!

Multicollinearity Example:

Multicollinearity Example:

MNIST data.

Multicollinearity Example:

MNIST data.
Use VIF as measure of multicollinearity.

Multicollinearity Example:

MNIST data.
Use VIF as measure of multicollinearity.

layer	% VIFs > 10	mean VIF
1	0.0078125	4.3537
2	0.9921875	46.84217
3	1	5.196113×10^{13}

Why Use NNs?!

Why Use NNs?!

- NNAEPR suggests that NNs are unnecessary.

Why Use NNs?!

- NNAEPR suggests that NNs are unnecessary. Just use PR.

Why Use NNs?!

- NNAEPR suggests that NNs are unnecessary. Just use PR.
- Advantages of PR:

Why Use NNs?!

- NNAEPR suggests that NNs are unnecessary. Just use PR.
- Advantages of PR:
    - No tuning parameter nightmare. (Just one parameter, d.)

Why Use NNs?!

- NNAEPR suggests that NNs are unnecessary. Just use PR.
- Advantages of PR:
  - No tuning parameter nightmare. (Just one parameter, d.)
  - No convergence problems.

Some of Our Experimental Results

Some of Our Experimental Results

- Compared PR vs. NNs on a wide variety of datasets.

Some of Our Experimental Results

- Compared PR vs. NNs on a wide variety of datasets.
    - PR: plain or with PCA beforehand
    - KF: **kerasformula**, R NN pkg.
    - DN: **deepnet**, R NN pkg.

Some of Our Experimental Results

- Compared PR vs. NNs on a wide variety of datasets.
  - PR: plain or with PCA beforehand
  - KF: **kerasformula**, R NN pkg.
  - DN: **deepnet**, R NN pkg.
- Calculated accuracy (mean abs. prediction error, prop. of correct classification).

Some of Our Experimental Results

- Compared PR vs. NNs on a wide variety of datasets.
  - PR: plain or with PCA beforehand
  - KF: **kerasformula**, R NN pkg.
  - DN: **deepnet**, R NN pkg.
- Calculated accuracy (mean abs. prediction error, prop. of correct classification).
- In every single dataset, **PR matched or exceeded the accuracy of NNs**.

Programmer/Engineer Wages

Programmer/Engineer Wages

setting	accuracy
PR, 1	25595.63
PR, 2	24930.71
PR, 3,2	24586.75
PR, 4,2	24570.04
KF, default	27691.56
KF, layers 5,5	26804.68
KF, layers 2,2,2	27394.35
KF, layers 12,12	27744.56

Prog./Eng. Occupation

Prog./Eng. Occupation

setting	accuracy
PR, 1	0.3741
PR, 2	0.3845
KF, default	0.3378
KF, layers 5,5	0.3398
KF, layers 500	0.3401
KF, layers 5,5; dropout 0.1	0.3399
KF, layers 256,128; dropout 0.8	0.3370

Million Song Data, predict year

Million Song Data, predict year

setting	accuracy
PR, 1, PCA	7.7700
PR, 2, PCA	7.5758
KF, default	8.4300
KF, layers 5,5	7.9381
KF, layers 2,2	8.1719
DN, layers 2,2	7.8809
DN, layers 3,2	7.9458
DN, layers 3,3	7.8060
DN, layers 2,2,2	8.7796

UCI Forest Cover Data, predict type

UCI Forest Cover Data, predict type

setting	accuracy
PR, 1	0.6908
PR, 2	-
KF, layers 5,5	0.7163

UCI Forest Cover Data, predict type

setting	accuracy
PR, 1	0.6908
PR, 2	-
KF, layers 5,5	0.7163

PR,2: out of memory

UCI Concrete Strength

UCI Concrete Strength

method	correlation (pred. vs. actual)
neuralnet	0.608
kerasformula	0.546
PR, 2	**0.869**

MOOCs Data, predict cert.

MOOCs Data, predict cert.

setting	accuracy
PR, 1	0.9871
PR, 2	0.9870
KF, layers 5,5	0.9747
KF, layers 2,2	0.9730
KF, layers 8,8; dropout 0.1	0.9712

Cancer/Genetics, predict Alive

Cancer/Genetics, predict Alive

model	brain cancer	kidney cancer
deepnet	0.6587	0.5387
nnet	0.6592	0.7170
PR (1, 1)	0.6525	0.8288
PR (1, 2)	0.6558	0.8265
PR (PCA, 1, 1)	0.6553	0.8271
PR (PCA, 2, 1)	0.5336	0.7589
PR (PCA, 1, 2)	0.6558	0.8270
PR (PCA, 2, 2)	0.5391	0.7840

Norm Matloff
University of California at Davis

Crossfit Data, predict Rx rank

Crossfit Data, predict Rx rank

model	accuracy	range among 5 runs
KF	0.081	0.164
PR, 1	0.070	0.027
PR, 2	0.071	0.069
PR, 3	0.299	7.08
PR, 4	87.253	3994.5

NYC Taxi Data, predict trip time

NYC Taxi Data, predict trip time

setting	accuracy
PR, 1	**580.6935**
PR, 2	591.1805
DN, layers 5,5	592.2224
DN, layers 5,5,5	623.5437
DN, layers 2,2,2	592.0192

Comments

Comments

- PR needs development of parallel comp. techniques.

Comments

- PR needs development of parallel comp. techniques.
- But $d = 2$ sufficed in almost all cases.

Comments

- PR needs development of parallel comp. techniques.
- But $d = 2$ sufficed in almost all cases.
- "Effective degree" of NN probably much bigger than 2. Hence overfitting.

Comments

- PR needs development of parallel comp. techniques.
- But $d = 2$ sufficed in almost all cases.
- "Effective degree" of NN probably much bigger than 2. Hence overfitting.
- Default values for number of layers etc. in NN software likely much too large.

Comments

- PR needs development of parallel comp. techniques.
- But $d = 2$ sufficed in almost all cases.
- "Effective degree" of NN probably much bigger than 2. Hence overfitting.
- Default values for number of layers etc. in NN software likely much too large.
- All NN software should monitor multicollinearity.

Comments

- PR needs development of parallel comp. techniques.
- But $d = 2$ sufficed in almost all cases.
- "Effective degree" of NN probably much bigger than 2. Hence overfitting.
- Default values for number of layers etc. in NN software likely much too large.
- All NN software should monitor multicollinearity. Likely causes the convergence problems.

Comments

- PR needs development of parallel comp. techniques.
- But $d = 2$ sufficed in almost all cases.
- "Effective degree" of NN probably much bigger than 2. Hence overfitting.
- Default values for number of layers etc. in NN software likely much too large.
- All NN software should monitor multicollinearity. Likely causes the convergence problems.
- See full paper, *https://arxiv.org/abs/1806.06850*.