Lehrbass, Frank

Research Report

Deep Learning Diagnostics − How to Avoid Being Fooled by TensorFlow, PyTorch, or MXNet with the Help of Modern Econometrics

ifes Schriftenreihe, No. 24

Provided in Cooperation with:
ifes Institut für Empirie & Statistik, FOM Hochschule für Oekonomie & Management

Suggested Citation: Lehrbass, Frank (2021) : Deep Learning Diagnostics − How to Avoid Being Fooled by TensorFlow, PyTorch, or MXNet with the Help of Modern Econometrics, ifes Schriftenreihe, No. 24, ISBN 978-3-89275-424-4, MA Akademie Verlags- und Druck-Gesellschaft mbH, Essen

This Version is available at:
https://hdl.handle.net/10419/249987

ifes Schriftenreihe

Band 24

Bianca Krol (Hrsg.)

Deep Learning Diagnostics – How to Avoid Being Fooled by TensorFlow, PyTorch, or MXNet with the Help of Modern Econometrics

~

Frank Lehrbass

FOM Hochschule
ifes
Institut für Empirie & Statistik
der FOM Hochschule
für Oekonomie & Management

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

MA Akademie Verlags- und Druck-Gesellschaft mbH
Leimkugelstraße 6, 45141 Essen
info@mav-verlag.de

Frank Lehrbass*

Deep Learning Diagnostics – How to Avoid Being Fooled by TensorFlow, PyTorch, or MXNet with the Help of Modern Econometrics

ifes Institut für Empirie & Statistik
der FOM Hochschule für Oekonomie & Management

ifes Schriftenreihe
Band 24, 2021

ISBN (Print) 978-3-89275-423-7
ISBN (eBook) 978-3-89275-424-4

ISSN (Print) 2191-3366
ISSN (eBook) 2569-5355

Abstract

Training a Multi-Layer Perceptron (MLP) to achieve a minimum level of MSE is akin to doing Non-Linear Regression (NLR). Therefore, we use available econometric theory and the corresponding tools in R. Only if certain assumptions about the error term in the Data Generating Process are in place, may we enjoy the trained MLP as a consistent estimator. To verify the assumptions, careful diagnostics are necessary.
Using controlled experiments we show that even in an ideal setting, an MLP may fail to learn a relationship whereas NLR performs better. We illustrate how the MLP is outperformed by Non-Linear Quantile Regression in the presence of outliers. A third situation in which the MLP is often led astray is where there is no relationship and the MLP still learns a relationship producing high levels of R^2. We show that circumventing the trap of spurious learning is only possible with the help of diagnostics.

* The views expressed herein are my own and do not necessarily reflect those of FOM University of Applied Sciences. Mail to frank.lehrbass@gmx.de.

Foreword[1]

My esteemed colleague Professor Frank Lehrbass is a dedicated and competent researcher. He gained highly valuable practical experience in finance from exposed positions as a risk manager and financial engineer. Due to his academic and professional background, he combines econometric expertise and relevant practical topics in the field of the financial industry. In this working paper, he discusses deep learning as an upcoming trend.

The application of deep learning is generally known for fields like machine vision or autonomous driving. In recent years, it has also entered the financial industry where digitalization becomes more and more important. One example are roboadvisors, which provide algorithm-based portfolio selection decisions for investors. Deep learning as a sort of machine-based learning applies artificial neuronal networks, where complex problems can be solved. Between input parameters or signals and output coefficients, various hidden layers of neuronal networks are used. Although deep learning has emerged as a successful procedure, shortcomings prevail. The so-called adversarial examples illustrate where procedures lead to misspecifications, for example induced by manipulated signals.

In this study, Professor Frank Lehrbass illustrates shortcomings of deep learning techniques to avoid the "trap of spurious learning" where investors "lose money and … suffer a break-down instead of achieving a breakthrough". He compares the performance of deep learning with the performance of regression specifications that are traditionally used in financial studies. The results show that deep learning can benefit from econometrics and econometric diagnostics in various ways. The valuable insight the reader can learn from this study is that a naïve belief in machine learning has to be handled with caution and how diagnostics can contribute to deal with shortcomings of deep learning techniques.

[1] Prof. Dr. habil. Christoph Schmidhammer is professor at the University of Applied Sciences of the Deutsche Bundesbank. His research focuses on the areas of Financial Econometrics, Market Microstructure, Operational Risk, Banking and Product Pricing.

Table of Contents

Abstract…. III
Foreword… IV
Table of Contents V
List of Figures VI
List of Tables VII
List of Abbreviations VIII
1 Introduction 9
2 Theoretical Background 11
3 Data-Generating Process I 15
4 NLR as Benchmark for MLP Training 17
5 Diagnostics for NLR and MLP 27
6 Data-Generating Process II 35
7 Change of Target Function for MLP and NLR 37
8 New Data and Linear Regression 39
9 New Data and MLP 45
10 Data-Generating Process III 47
11 Outlook: Hypothesis Testing for MLP as for NLR 48
12 Diagnostic Steps and Common Wisdom 50
13 Putting the Findings into Context 51
14 Explainability versus Diagnostics 52
15 Conclusion 53
Literature…… 54

List of Figures

Figure 1: Stylized Chicken-Growth Data 16
Figure 2: Learning Progress of Gauss-Newton Algorithm....18
Figure 3: Learning Progress of SGD....20
Figure 4: Learning Progress of AdaDelta 22
Figure 5: Learning Progress of AdaGrad....23
Figure 6: Learning Progress of Adam....24
Figure 7: Learning Progress of RMSProp 25
Figure 8: Stylized Chicken-Growth Data: NLR and MLP Forecasts 27
Figure 9: Stylized Chicken-Growth Data: NLR Regression 29
Figure 10: Stylized Chicken-Growth Data: Residuals NLR Regression 30
Figure 11: Stylized Chicken-Growth Data: Residuals MLP 31
Figure 12: Stylized Chicken-Growth Data: Residuals MLP 32
Figure 13: Stylized-Chicken Growth Data with Outliers 35
Figure 14: Chicken-Growth Data with Outliers: NLR and MLP Forecasts .36
Figure 15: Data with Outliers: NLR/NLQR and Two MLP Forecasts....37
Figure 16: New Data Containing Two Time Series 39
Figure 17: New Data Visual Mapping x to y....40
Figure 18: New Data: Linear Regression 41
Figure 19: New Data: Regression Residuals 43
Figure 20: New Data: MLP 45
Figure 21: New Data: MLP Residuals 46

List of Tables

Table 1: Parameters of Logistic Function 15
Table 2: Estimates of R-Function 17
Table 3: Set of Weights....21
Table 4: Squared Correlations Between Realized Chicken Weights and Forecasts....26
Table 5: Result from Estimation Resp. Training....34
Table 6: Corresponding Parameters....38
Table 7: Summary and Practical Value....50

List of Abbreviations

ACF	Autocorrelation Function
CNN	Convolutional Neural Networks
DGP	Data Generating Process
LSTM	Long Short-Term Memory Models
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NLQR	Non-Linear Quantile Regression
NLR	Non-Linear Regression
NLS	Non-Linear Least Squares
OLS	Ordinary Least Squares Algorithm
RMSProp	Root Mean Square Propagation

1 Introduction

With the advent of cloud computing, it has become easier to implement Machine Learning (ML) models including big Deep Learning models like LeNet style Convolutional Neural Networks (CNN) for image recognition or Long Short-Term Memory models (LSTM) for time series forecasting. Both models can be subsumed under the term Multi-Layer Perceptron (MLP).

Analytics as a service is provided by Amazon Web Services, Google Cloud Platform, Microsoft Azure, or other machine learning service providers.[2] To stay focussed I explore the motivation of just one service provider. The vision behind Microsoft Azure is as follows:[3]

"Microsoft Azure Machine Learning environment 'democratizes' the mysterious art of data science. It exposes this art to the masses by means of an easy-to-use visual designer interface that requires literally no programming and only a browser to use. We have seen how it enables virtually anyone to create sophisticated and powerful predictive analytic solutions, and how it can then quickly share and expose them for further development, refinement, and implementation. With the option to publish a predictive model right into the Azure Data Marketplace, you can even quickly and easily monetize your next big breakthrough in machine learning" (Barnes, 2015, 93).

But analytics as a service might make it too easy to apply ML. At first glance this might be an unexpected worry – we are all democrats, are we not? – but there are dangers in applying ML without proper background knowledge. In fact, without decent diagnostics and theoretical backing, there is a high chance for "virtually anyone" to lose money and to suffer a breakdown instead of achieving a breakthrough. A "growing body of evidence shows that state-of-the-art models learn to exploit spurious statistical patterns in datasets" (Nie et al, 2019, 1). More specifically, I have recently presented two worked-out examples to underline this (Lehrbass, 2020a). Now I aim to show how to prevent this from happening and provide the required theoretical backing based on White (1992).

Thus, it emerges that MLP can be subsumed under Non-Linear Regression (NLR). The advantage of doing so is that we can reuse the machinery of NLR, especially

[2] For a more comprehensive list of companies see Markets and Markets (2020).

[3] In addition, of course, Microsoft's goal is to make a profit.

theorems concerning the quality of training results. More to the point they clarify whether big data[4] is beneficial for detecting the true structure within our data (i.e. consistency) and under what conditions this is possible at all. The term diagnostics captures all activities for checking the validity of the assumptions of the theorems.

Throughout, this paper exemplifies the theoretical statements with examples written in R using MXNet.[5] The study starts with chicken-growth data and benchmarks an MLP with NLR. Diagnostics are applied to both models. This example also tells us a lot about potential pitfalls in training and the relative weakness of gradient-based methods. As a next step, I add outliers to the data to check the robustness[6] of the learned function. This motivates the use of other target functions.

The final example demonstrates spurious learning and shows how important diagnostics are to avoid being led astray.[7] Diagnostics are the only way to avoid this trap. The reader might be reminded of the modeling approach as in Box and Jenkins (1976), which starts with model identification and selection, parameter estimation (training), and model checking, i.e. diagnostics. It is my conviction that this last step should play a bigger role in ML. I also discuss how far attempts to explain[8] machine learning results can add value to diagnostics. Throughout, I assume some familiarity with ML as a prerequisite.[9]

---

4 Here I simply refer to the size of the samples (=Volume). The other Vs of "Big Data" like Variety, Velocity, and Veracity are not in focus.

5 I could have chosen TensorFlow or PyTorch as well, but for my own teaching at FOM University of Applied Sciences R is the first choice and students should benefit easily from this study. Details on MXNet can be found in Chen et al. (2015) and inMXNet (no year, a).

6 "Despite astonishing progress in the field of Machine Learning (ML), the robustness of high-performance models, especially the ones based on Deep Learning technologies, has been lower than initially predicted. These networks do not generalize as expected, remaining vulnerable to small adversarial perturbations (also known as adversarial attacks). Such shortcomings pose a critical obstacle to implement Deep Learning models for safety-critical scenarios such as autonomous driving, medical imaging, and credit rating" (German Research Center for Artificial Intelligence, no year).

7 I am not against the application of ML and Analytics as a service; after all, I have been applying ML methods since the early 1990s (e.g. Kohonen maps as in Volmer, Lehrbass, 1997, MLP as in Lehrbass, Peter, 1996, recently Lehrbass, 2020b and Lehrbass, Schuster, 2021) and have been teaching them since 2018.

8 Roscher et al. "differentiate between transparency, interpretability, and explainability" (2019, 1).

9 I recommend the book by Ghatak (2019), which is a well-structured introduction into "Deep Learning with R". The first eight chapters in Goodfellow et al. (2016) and Verbeek (2012) introduce the mathematical background.

2 Theoretical Background

The goal of estimating an econometric model or training an MLP is to discover the true process which generates our data. In short: The aim is to uncover the Data Generating Process (DGP).

Assumption 1 (Specification)

Therefore, it is assumed, that – if there were no noise in the world – there would be a true relationship as follows:

$$y_t = f(x_t, W) \qquad (1)$$

For the sake of simplicity, I assume the function *f(x,W)* to be a continuous mapping from the independent variable *x* to the dependent variable *y*. The variable *x* might be a vector of regressors or factors.

The parameters of the function are denoted by a parameter or weight matrix *W*. The function might be linear or not. To allow for gradient- and Hessian-based learning the function needs to be twice differentiable in *W*.[10] Note that this implies continuity in *W*.

With a noisy world the mapping in equation (1) can be expected to hold only on average, which I denote via a conditional expectation:

$$E(y_t|x_t) = f(x_t, W) \qquad (2)$$

Let us put the effects of noise into a random variable u. The full data generating process can then be expressed as follows:

$$y_t = f(x_t, W) + u_t \qquad (3)$$

The data in our samples is disturbed by *u,* which makes us err concerning the true relation between *y* and *x*. Therefore, the u_t are called error terms. Equation (3) is already a strong assumption.

Assumption 2 (Exogeneity)

Due to the linearity of the expectation operator, the following assumption for the distribution of the error term is implied:

$$E(u_t|x_t) = 0 \qquad (4)$$

[10] Goodfellow et al. point out that this assumption can be weakened (2016, 216).

This means that the expected error – given *x* – is zero. In other words: I have exploited all information that is contained in the regressors' forecasting *y*. Often, this assumption is referred to "by saying that the *x* variables are exogenous" (Verbeek, 2012, 14).

Assumption 3 (I.I.D. Error Term)

In the sequel, the error terms *u* are assumed to be independently and identically distributed (i.i.d.) with a finite variance > 0.

Target Function

Training resp. estimation is aimed at minimizing the expected squared error. Formally:

$$\min_{W} E[(y_t - f(x_t, W))^2] \qquad (5)$$

Since the probability law ruling *y* given *x* and *W* is unknown, one can only use observations of *y* and *x* to approximate the expectation operator. Given a training set of size *n* this approximation reads:

$$\min_{W} \frac{1}{n} \sum_{t=1}^{n} (y_t - f(x_t, W))^2 \qquad (6)$$

In brackets, there are the squared residuals, i.e. true *y* minus forecast by the model. Our target can be expressed in other words: We want to minimize the Mean Squared Error (MSE). The solution to this problem is denoted by $\widehat{W}$ and is known under various names:

Linear Regression

When the function *f(x,W)* is linear in *x,* we are doing linear regression using the ordinary least squares algorithm (OLS).

Theorem 1 (Gauss-Markov)

It can be shown that the estimator $\widehat{W}$ is the best linear unbiased estimator for *W* (BLU, Verbeek, 2012, 17).

This is the famous Gauss-Markov Theorem of Econometrics. The U in BLU stands for Unbiasedness, i.e. it is expectable to get estimates centered around the true value. In other words, on average I estimate the true function (Unbiasedness)

and do so with as little variance as possible (Best).[11]

Non-Linear Regression

When the function *f(x,W)* is non-linear in *x* I am doing NLR. Geometrically, the solution to the least-squares problem is the same as in the linear regression model. Both times I "seek the point on the curve generated by *f(x, W)* that is closest in Euclidean distance to *y*" (Davidson, MacKinnon, 2009, 234). The difference is that *f* is a linear function in the first place and non-linear in the second one. Another difference is that from now on I apply numerical methods to solve the problem (6).

Again, what is interesting, are the properties of the resulting non-linear least squares (NLS) estimator (Davidson, MacKinnon, 2009, 224). The specific results of the estimator $\widehat{W}$ depend on the sample and its size *n*. Therefore, the estimator requires a subscript as follows: $\widehat{W}_n$.

From now on I state properties of the estimators in large samples, which are called asymptotic properties. In today's world of big data, we are in a happy situation to have large samples as a rule. Therefore, the following theorems are of practical interest.

Theorem 2 (Jennrich)

It can be shown that the NLS estimator $\widehat{W}_n$ is a strongly consistent estimator for *W* (Jennrich, 1969, 636).

Hence, for increasing sample size *n* the estimator $\widehat{W}_n$ converges to the true *W* with probability one. Note that "consistency and unbiasedness are ... different concepts" and that "estimators may be biased but consistent" (Davidson, MacKinnon, 2009, 96).[12]

Multi-Layer Perceptron

Now let us move to a popular technique from ML: The Multi-Layer-Perceptron

[11] To come full circle, the L in BLU shows that the estimator is a linear function of the observations.

[12] Example (Davidson, MacKinnon, 2009, 97): Estimate the true population mean μ by the estimator $\frac{\sum_{t=1}^{n} x_t}{n+1}$. We see that it comes close to the unbiased estimator "sample mean" in the limit for large n, because then n is very similar to n+1. Hence, it is consistent but misses the true μ always slightly, i.e. is biased.

(MLP). MLP are also called feedforward neural networks and are the "quintessential deep learning models. The goal of a feedforward network is to approximate some function f" (Goodfellow, 2016, 163). Therefore, an MLP can be subsumed in the current framework as follows: The function *f(x,W)* is non-linear in *x,* and training of the MLP – using MXNet or TensorFlow or any other software – is aimed at minimizing MSE (Mean Squared Error) or monotone transformations thereof like RMSE (Root Mean Squared Error).

Assumption 4 (Most Parsimonious Architecture)

We need two more assumptions: There are no "redundant inputs" and there are no "irrelevant hidden units" (White, 1992, 105). Now we can state:

Theorem 3 (White)

It can be shown that the weight matrix of the trained MLP $\widehat{W}_n$ is a strongly consistent estimator for *W* (White, 1992, 123).

Again, for an increasing sample size *n* the MLP, represented by its weights $\widehat{W}_n$, converges to the true MLP with probability one. Note that a trained MLP corresponds to an NLS estimator.[13] To conclude I ask which functions by *f(x, W)* can be learned by an MLP.

Theorem 4 (Hornik, Stinchcombe, White)

An MLP can approximate any continuous function arbitrarily well (Hornik, Stinchcombe, White, 1989). It does not need more than a single hidden layer.

Hence, an MLP is a universal approximator, i.e. *f(x, W)* might be any function as long as it is continuous. Note that it might be the case that a huge number of hidden units is required. In terms of layers, one is sufficient. Goodfellow et al. (2016, 192-195) highlight important aspects of this theorem in more detail.

This theorem has a strong consequence for ML: "Any lack of success in applications must arise from inadequate learning, insufficient number of hidden units or the lack of a deterministic relationship between input" *x* and output *y* (White, 1992, 20).

[13] White (1989) shows that the usually applied backpropagation is capable of successfully solving the problem (6). There is a grain of salt: He merely shows it for an MLP with only one hidden layer.

3 Data-Generating Process I

The theorems above contain strong statements concerning the abilities of ML to uncover the true DGP. To verify these capabilities I set up an experiment. I generate data in a fully controlled manner, i.e. I know the true *f(x, W),* and investigate how close the various methods come i.e. how close $\widehat{W}_n$ gets to W.[14]

The data is generated in order "to mimic … real chicken-growth data" (Riazoshams, Midi, Ghilagaber, 2019, 227), which was recorded of "broiler chicken supply in an area of Marvdasht, Fars province, Iran" (Riazoshams, Midi, Ghilagaber, 2019, 216).

To make this data easily learnable for an MLP with a typical sigmoid[15] activation function I simplify the four-parameter logistic function, which is given by Riazoshams, Midi, and Ghilagaber (2019, 227), to a two-parameter sigmoid function. Specifically, the data are simulated from the following logistic model:

$$y_t = \frac{1}{1+exp(b_0+b_1\, x_t)} + u_t \qquad (7)$$

The u_t are error terms following a normal distribution with a variance of s^2 and an expected value of zero. The independent variable x_t ranges from 3 to 50 and counts the days on which the dependent variable chicken weight y_t is measured. The weight measurement is normalized such that the maximum weight is one. The parameters of the logistic function are set as follows:

Parameter	b_0	b_1	s
True Value	3.5	-0.11	0.05

Table 1: Parameters of Logistic Function

[14] One might call this approach "hide & seek". One hides *f()* in the *x,y* data and seeks for it.

[15] A popular example is the logistic activation function. Throughout I use logistic and sigmoid interchangeably.

Thus, the DGP is specified. Using a sample size of n=100 one finds:

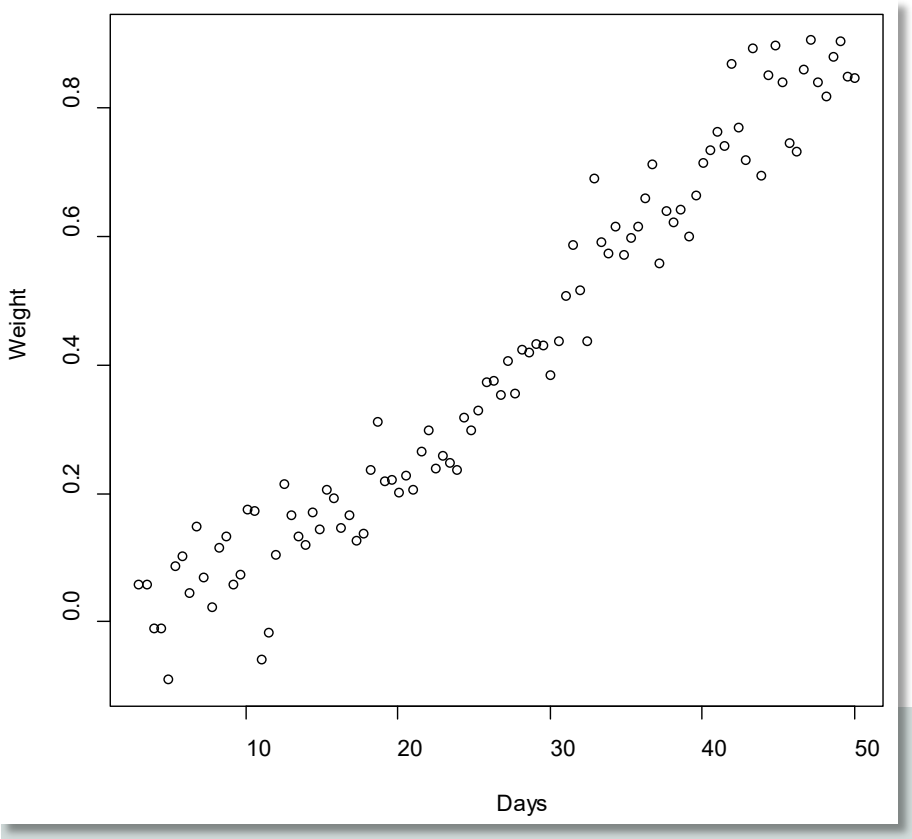

Figure 1: Stylized Chicken-Growth Data

4 NLR as Benchmark for MLP Training

The statistics software R contains the function nls() to carry out NLR. It uses the Gauss-Newton algorithm as the default to solve the NLS problem. The mathematical background can be found in Goodfellow et al. (2016, 87). Note that not only information from the gradient is used but also the Hessian matrix, i.e. the first two derivatives matter.

After five iterations the R-function nls() yields the estimates as given in the last row of the following table:[16]

Parameter	Model / Algorithm	b_0	b_1	s
True Value		3.5	-0.11	0.05
Estimated Value	NLR Gauss-Newton	3.4117	-0.1082	0.0562

Table 2: Estimates of R-Function

Tracing the MSE during estimation one can visualize a kind of learning behavior.

[16] To supply a level playing field for all approaches, I set the starting values via drawing uniform random variables around zero with a radius/max/min of 0.1. This is quite common, e.g. see Liu, Maldonado (2018, 75). "Weight initialization can have a profound impact" on learning behavior (Ghatak, 2019, 87). However, "there is no specific rule for selecting any specific method" (Ghatak, 2019, 99).

As in the sequel, I call each iteration in the numerical search an "Epoch":

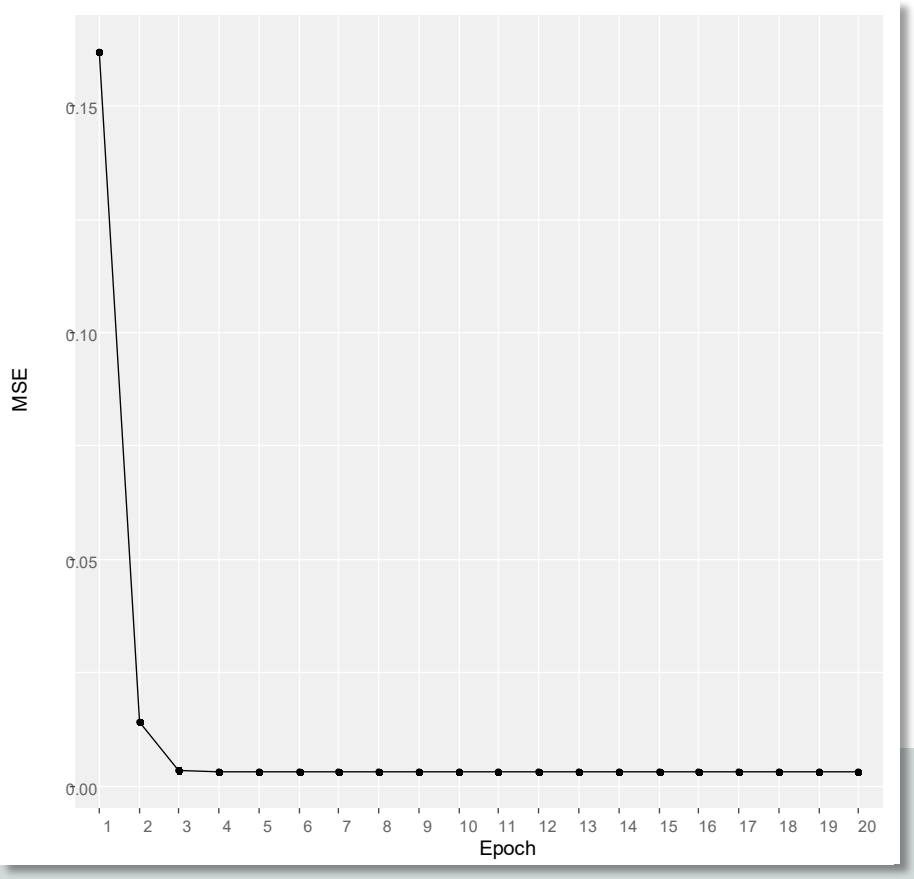

Figure 2: Learning Progress of Gauss-Newton Algorithm

Now an MLP[17] is trained on the same data. Due to the functional form, the MLP should be able to get close to the truth, too. We use the (hyper-) parameters as in the example given in the MXNet online tutorial "Develop a Neural Network with MXNet in Five Minutes".[18] To have a weight/parameter matrix of the same

[17] Strictly spoken this is a single layer perceptron with one neuron. It is used for the sake of exposition. Even in this most simplified example, the trained MLP falls behind NLR. The results can be generalized to richer MLP. The function *f()* is not restricted to one sigmoid function.

[18] See MXNet (no year, b), specifically num.round=20, array.batch.size=15, learning.rate=0.1, momentum=0.9. The meanings are explained in Goodfellow, 2016, ch. 5-8. Similar parameters can be found in Liu, Maldonado (2018, 75). A minibatch stochastic method is applied (Goodfellow

size as before, I apply a single neuron with a logistic activation function.[19] Note that this puts the MLP in a similarly favorable situation as the NLR. In both cases, the functional form coincides with the DGP. Hence, there is no need to worry about misspecification. On top of it, only two weights have to be learned.

On the first try, I use MXNet's default method, stochastic gradient descent (SGD), for optimization (Goodfellow, 2016, 147) using 20 epochs as in the "Five Minutes" example from the web. The learning behavior is surprisingly poor as the evolution of the MSE shows:

et al. 2016, 272). Since "small batches can offer a regularizing effect" we keep the small batch size (Goodfellow et al., 2016, 272).

[19] Aka sigmoid function. See Ghatak (2019, 34-42) for a discussion of the pros and cons of different activations.

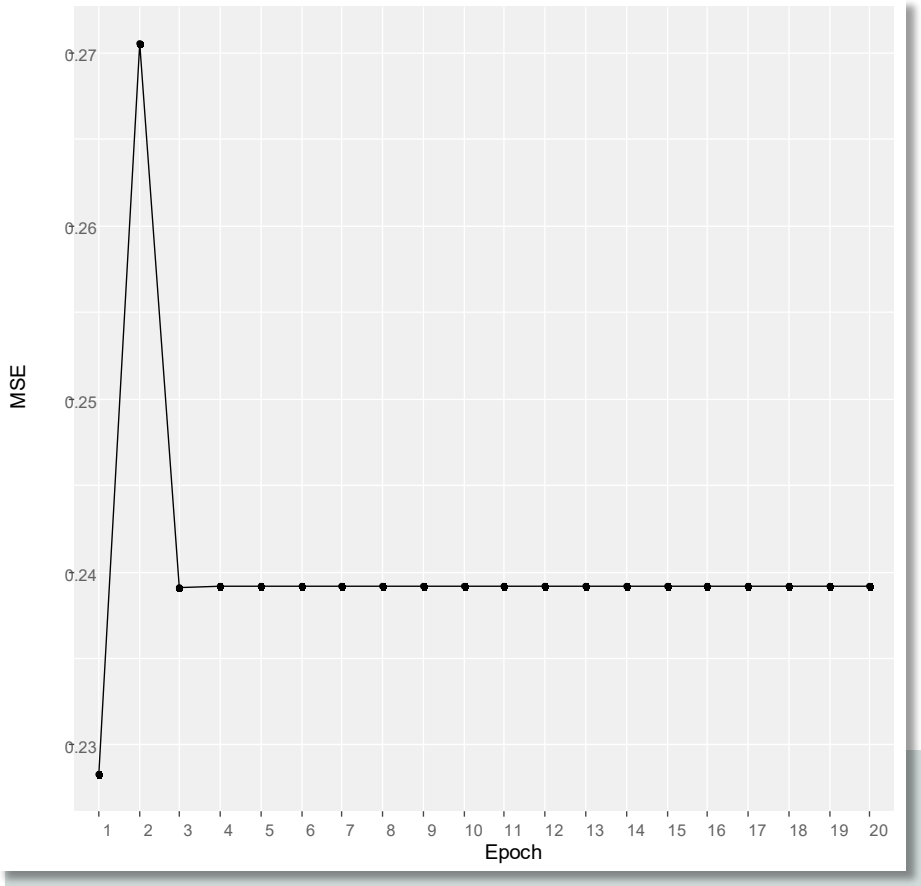

Figure 3: Learning Progress of SGD

The set of weights from training is as follows:[20]

Parameter	Model / Algorithm	b_0	b_1	s
True Value		3.5	-0.11	0.05
Trained Value	MLP SGD	0.3803	1.9534	0.4990

Table 3: Set of Weights

There are three potential reasons for this poor outcome: The reasons a) "insufficient number of hidden units" and b) "the lack of a deterministic relationship" can be discounted from the list, because a logistic relationship has to be learned. Thus, Assumption 1 (Specification) and Assumption 4 (Most Parsimonious Architecture) can be considered given.

Hence, reason c) remains, which is "inadequate learning". Therefore, I change the learning algorithm and apply some of the most popular alternatives,[21] which are ready for use in MXNet together with their default hyperparameters. Since a picture is worth a thousand words, I show the evolution of MSE over the 20 epochs per algorithm in alphabetical order.

[20] The sigmoid function in MXNet takes exp(-x). Hence, one has to multiply the trained weights by minus 1 to make them comparable to the estimates from NLS.

[21] For a description see chapter 5 in Ghatak, 2019.

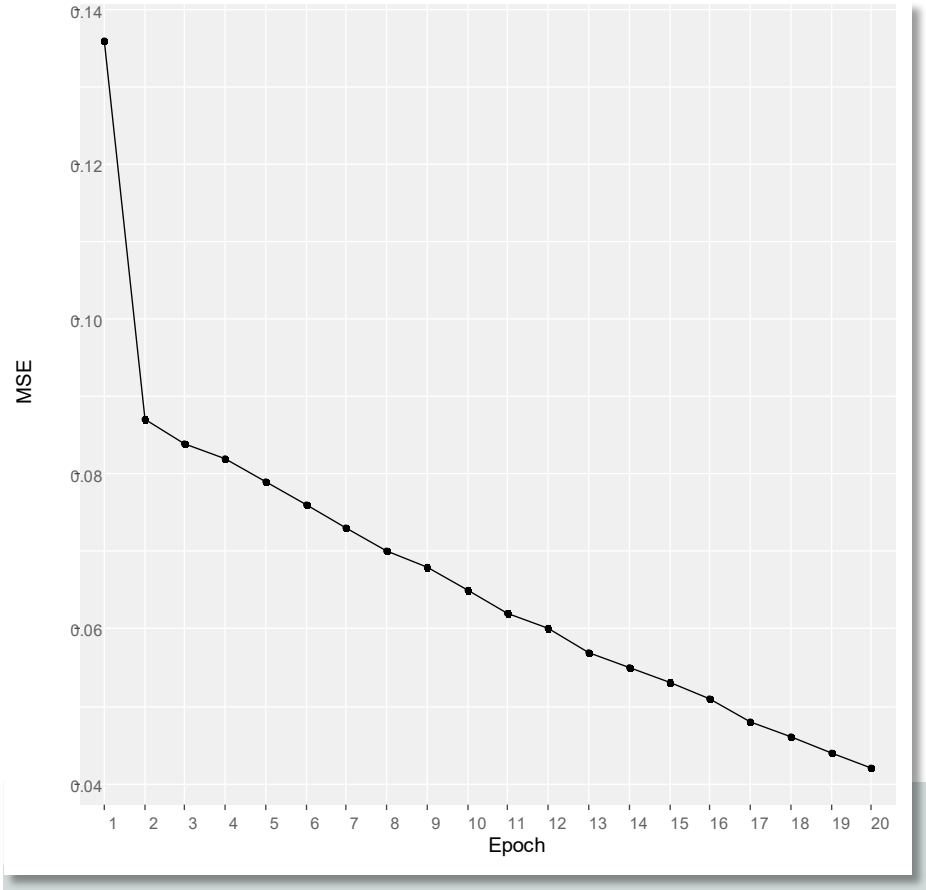

Figure 4: Learning Progress of AdaDelta

AdaDelta achieves a steady improvement because the MSE drops from epoch to epoch.

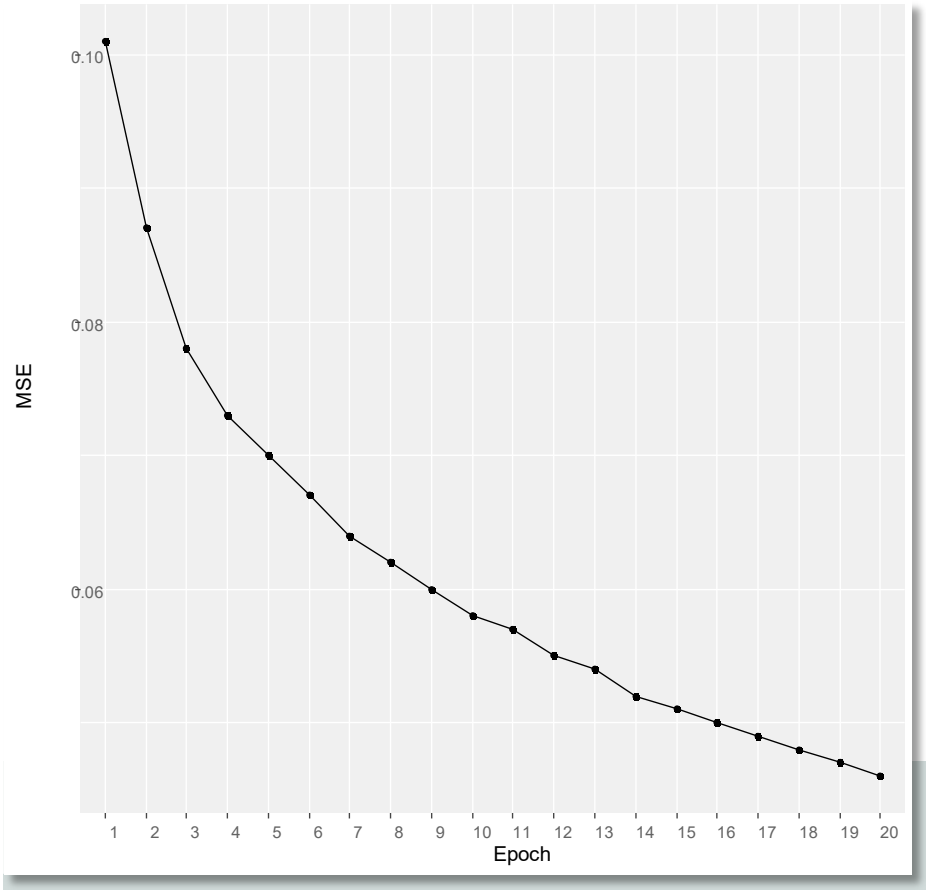

Figure 5: Learning Progress of AdaGrad

AdaGrad achieves a steady improvement, too.

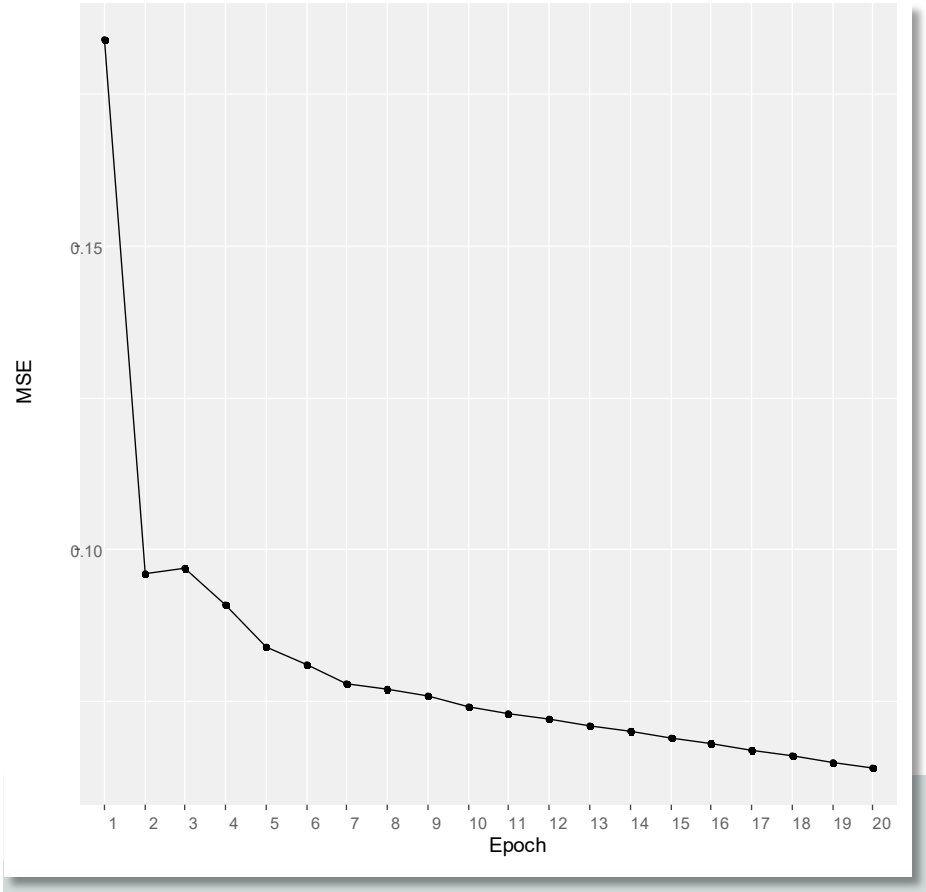

Figure 6: Learning Progress of Adam

Ignoring the initial upwards movement it is safe to say that Adam achieves a steady improvement.

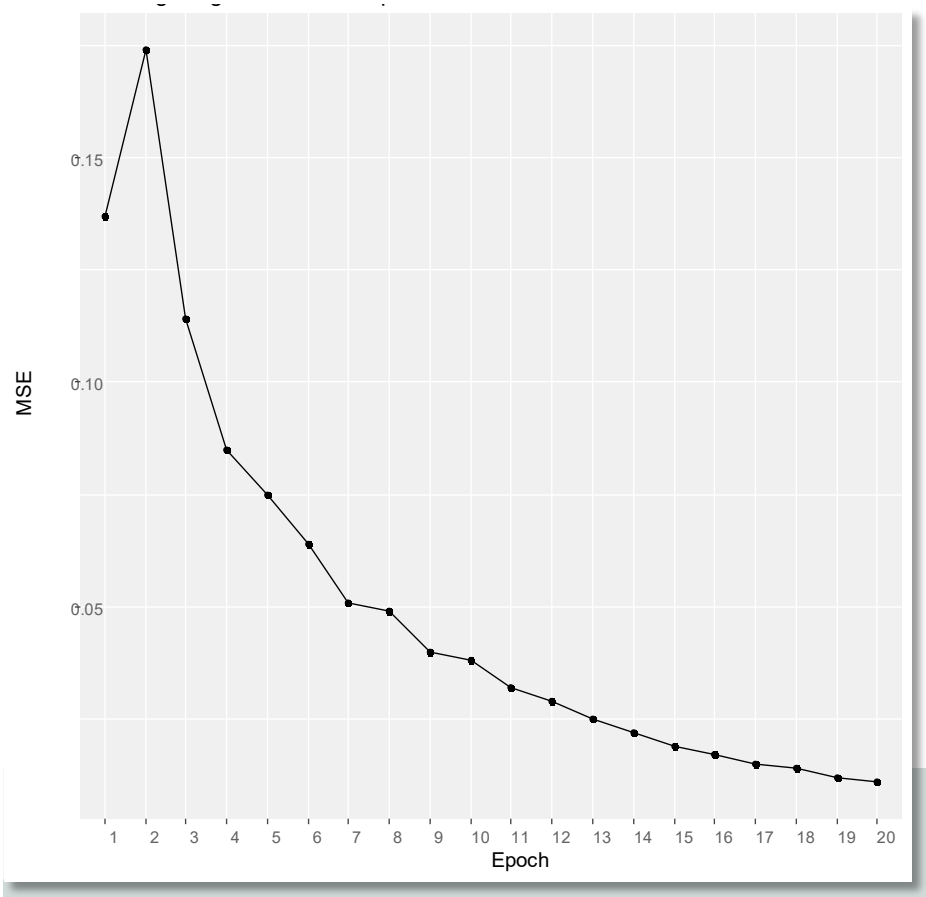

Figure 7: Learning Progress of RMSProp

Ignoring a few kinks one may state again that RMSProp achieves a steady improvement as well.

The visual impression of the learning progress is confirmed by the corresponding R^2. The squared correlations between realized chicken weights *y* and forecasts by each of the models are as follows:

Model	Algorithm	R^2
Non-Linear Regression	Gauss-Newton	0.9620
MLP	SGD	0.0381
	AdaDelta	0.9475
	AdaGrad	0.9466
	Adam	0.9440
	RMSProp	0.9575

Table 4: Squared Correlations Between Realized Chicken Weights and Forecasts

"Currently, the most popular optimization algorithms actively in use include SGD, ..., RMSProp, AdaDelta, and Adam. The choice of which algorithm to use … seems to depend largely on the users' familiarity" (Goodfellow, 2016, 302). Ghatak has chosen Adam in his examples (2019, 159 and 192). Yet I work differently and go for the highest R^2, i.e. I choose RMSProp[22] for the following chapter.

Insights from this section

Even in an ideal setting, an MLP may fail to learn a relationship; NLR performs better. This shows the value of benchmarking MLP with other available techniques. The reason for the inadequate learning of the MLP may be the learning algorithm. A simple gradient-based algorithm may fail, whereas refined versions may perform better. If doable in terms of computer resources, Hessian-based methods may add value.

[22] Ghatak explains the details and shows how RMSProp (Root Mean Square Propagation) can be seen as an improvement of AdaGrad (Ghatak, 2019, 112). In another study my colleague and I have applied AdaGrad (Lehrbass, Schuster, 2021).

5 Diagnostics for NLR and MLP

Assumption 1 (Specification) and Assumption 4 (Most Parsimonious Architecture) have already been confirmed above. As a supporting argument, I chart the forecasting function of the NLR (solid) and the MLP with RMSProp (dashed).

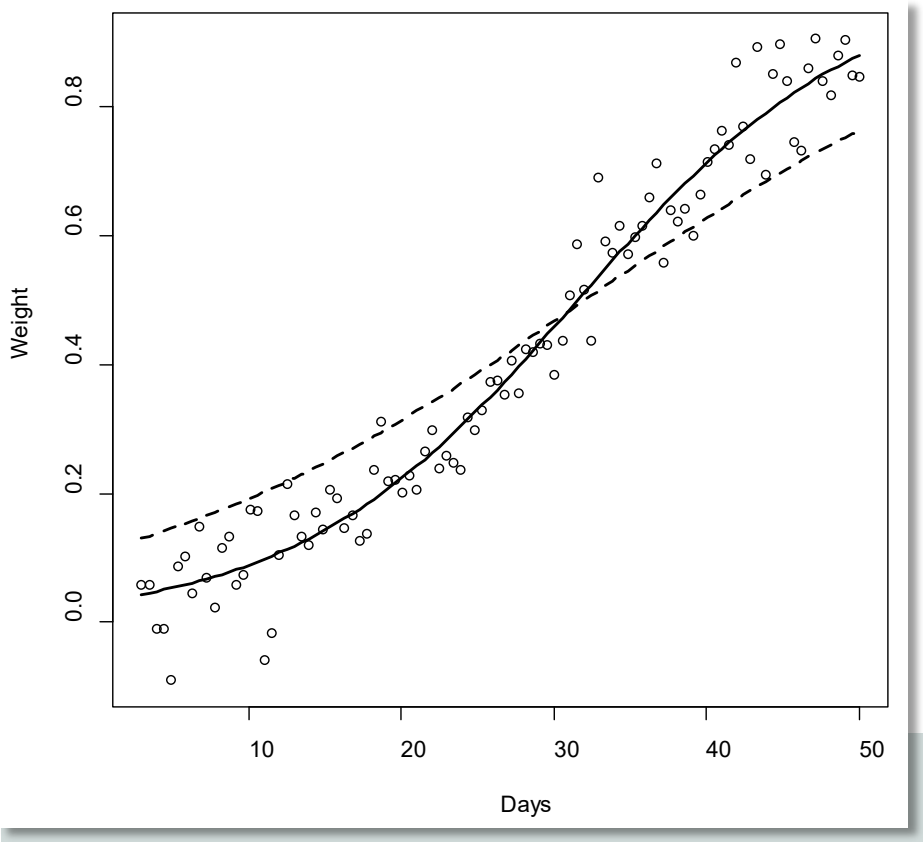

Figure 8: Stylized Chicken-Growth Data: NLR and MLP Forecasts

Assumption 2 (Exogeneity) is given by construction. The time index *x* is exogeneous.[23] What remains to be examined is Assumption 3 (I.I.D. Error Term). A good starting point is to regress the true *y* on the forecasts of the model.

[23] Chickens cannot read the calendar.

The first model under consideration is the NLR and one receives the following:

```
Call:
lm(formula = y ~ fitted(nls1))
Residuals:
      Min        1Q    Median        3Q       Max
-0.157542 -0.034848  0.000378  0.034252  0.153111
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.0003305  0.0099616  -0.033    0.974
fitted(nls1)  1.0004633  0.0200879  49.804   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1
Residual standard error: 0.05616 on 98 degrees of free-
dom
Multiple R-squared:  0.962,   Adjusted R-squared:  0.9616
F-statistic:  2480 on 1 and 98 DF,  p-value: < 2.2e-16
```

For the intercept one may keep the H0 that it is zero, i.e. I do not err systematically. The slope coefficient is close to one. Both findings imply that the mean residual is zero.[24] But what about the other properties of the residuals?

As a next step, this study examines the independence assumption using an autocorrelation function (ACF). Critical values for a confidence level of 99% are shown as dashed lines. Doubting independence is unnecessary:

[24] To see it, deduct slope times forecast from both sides of the equation. Using slope = 1 one receives residual = intercept.

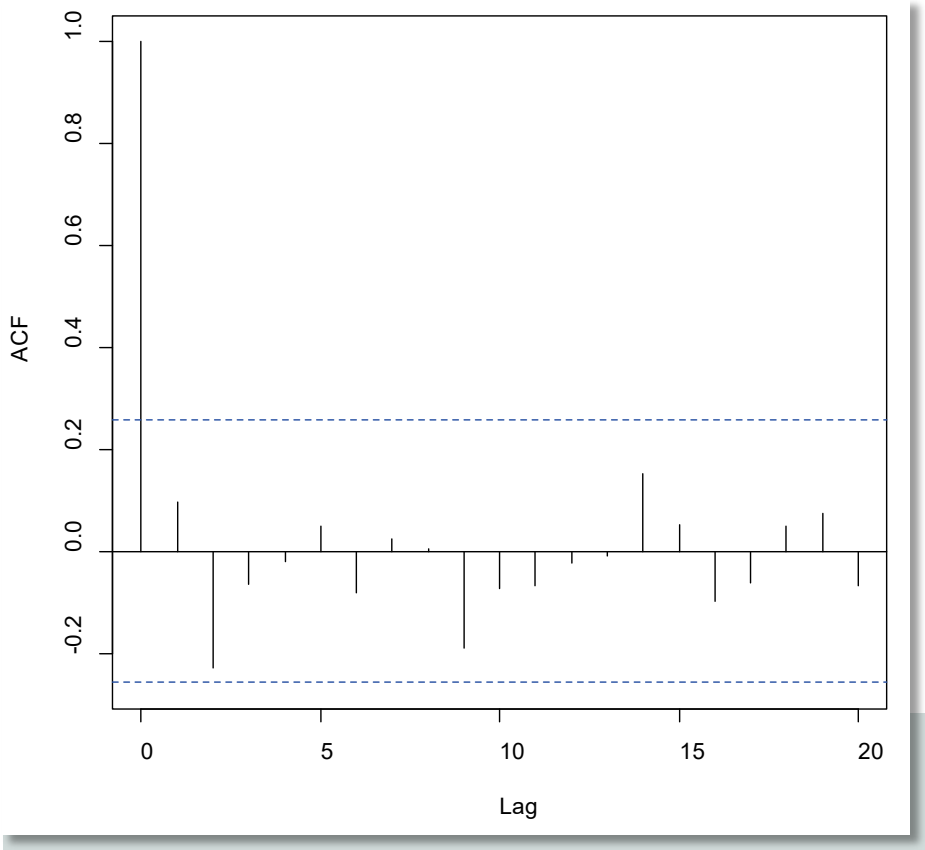

Figure 9: Stylized Chicken-Growth Data: NLR Regression

Concerning the assumption of a constant variance one finds comfort in the following chart:

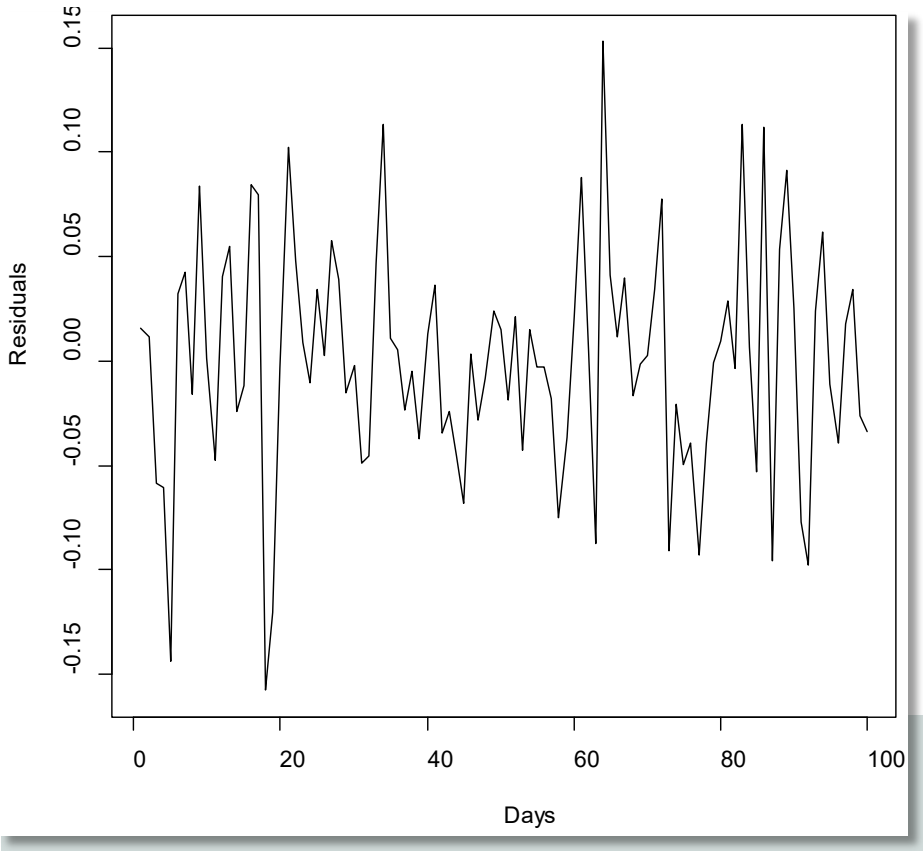

Figure 10: Stylized Chicken-Growth Data: Residuals NLR Regression

One might add tests for homoscedasticity[25] and no-autocorrelation as known from econometrics. But this would not change the message. Hence, Assumption 3 (I.I.D. Error Term) seems to be in place for the NLR.

[25] Aka constant variance.

But for the MLP things look different in terms of autocorrelation:

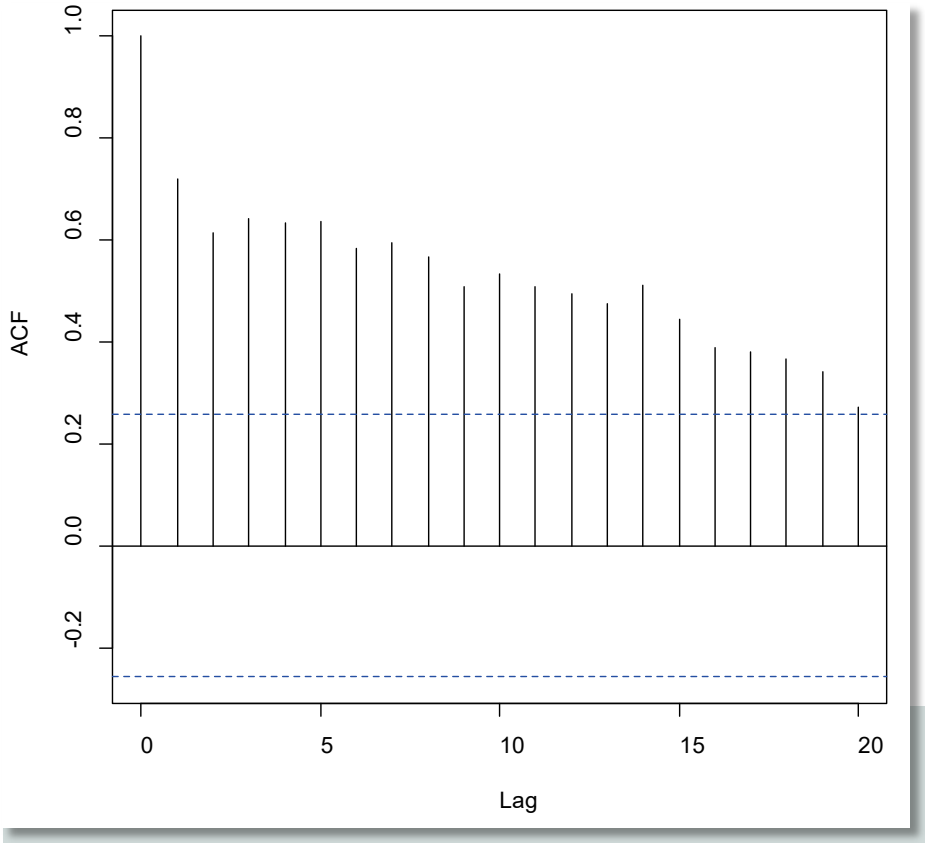

Figure 11: Stylized Chicken-Growth Data: Residuals MLP

Inspection of the time series of residuals reveals a trending behavior:

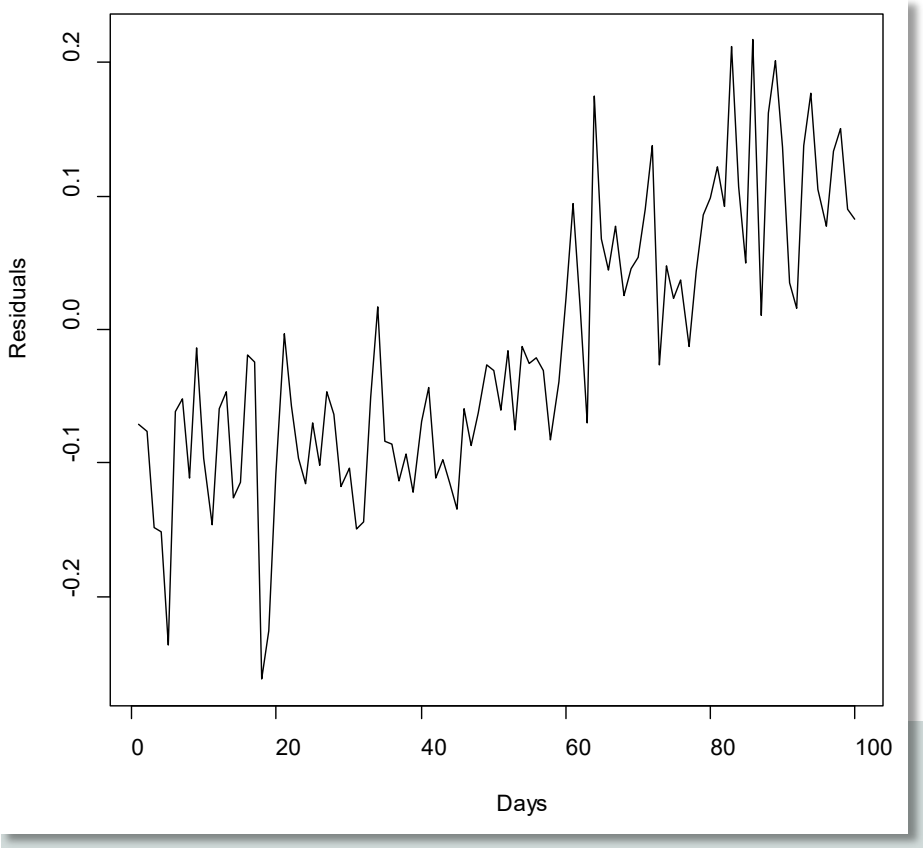

Figure 12: Stylized Chicken-Growth Data: Residuals MLP

To conclude I regress the true *y* on the forecasts of the MLP.

```
Call:
lm(formula = y ~ preds2)
Residuals:
      Min        1Q    Median        3Q       Max
-0.153695 -0.035209 -0.005047  0.036950  0.153340
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.19339    0.01414  -13.68   <2e-16 ***
preds2       1.41786    0.03018   46.98   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1
Residual standard error: 0.05939 on 98 degrees of free-
dom
Multiple R-squared:  0.9575, Adjusted R-squared:  0.9571
F-statistic:  2207 on 1 and 98 DF,  p-value: < 2.2e-16
```

This time I err systematically because the intercept is significantly different from zero. The upshot is that Assumption 3 (I.I.D. Error Term) is violated in the case of the MLP. Hence, increasing the sample size will increase one's proximity to the true parameters of the DGP in the case of NLR.[26] But where this leads me with the MLP is uncertain, because Assumption 3 seems to be violated.

[26] According to Theorem 2.

To illustrate this point the sample size is increased and the result from estimation resp. training is shown below. Note that the values for the MLP differ from the entries in the table above because we refer to the MLP trained with RMSProp, which has the highest R^2:

Parameter	Model / Sample Size	b_0	b_1	s
True Value		3.5	-0.11	0.05
Estimated Value	NLR / 100	3.4117	-0.1082	0.0562
Trained Value	MLP / 100	2.0917	-0.0654	0.1023
Estimated Value	NLR / 1,000	3.5394	-0.1111	0.0519
Trained Value	MLP / 1,000	3.5643	-0.1097	0.0532
Estimated Value	NLR / 10,000	3.5135	-0.1104	0.0499
Trained Value	MLP / 10,000	3.5462	-0.1155	0.0548
Estimated Value	NLR / 100,000	3.5065	-0.1102	0.0499
Trained Value	MLP / 100,000	3.5274	-0.1103	0.0499
Estimated Value	NLR / 1,000,000	3.4984	-0.1100	0.0500
Trained Value	MLP / 1,000,000	3.5148	-0.1159	0.0584

Table 5: Result from Estimation Resp. Training

For the NLR, Theorem 2 can be seen in action. The estimates converge to the true *W* for an increasing sample size. Surprisingly, the training results of the MLP are not as bad. Despite the violation of Assumption 3 the MLP produces learning results, which almost fit the correct order but not entirely. In the following, I show that this finding must not be generalized.

Insights from this section

Diagnostics are a must. If they do not make us reject the assumptions, we can be certain to approach the true relationship with increasing sample size. Thus, big data holds its promise. But if we have to reject the assumptions, we are left without orientation. Although we might be lucky in some cases, it is not a certainty.

6 Data-Generating Process II

I leave the orderly generation of data according to a well-defined DGP behind. With some brute force, certain error terms are amplified to perturb the data in an adversarial way.[27] Thus, the error terms are no longer kept in check. Ten percent of the data is perturbed as follows:

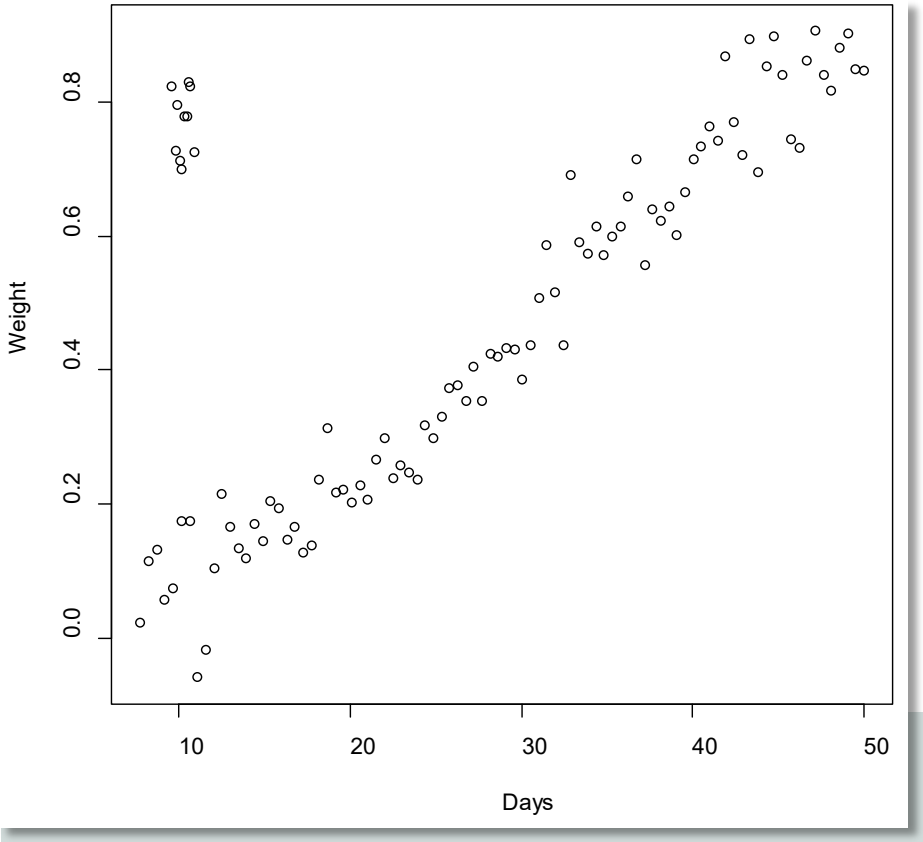

Figure 13: Stylized-Chicken Growth Data with Outliers

One may expect that the MLP and NLS will be confused by the outliers. This is confirmed by the fitted curves shown in the following chart:

[27] In real life this could be due to faulty measurement. Note that this is no simple up-shift of the parameter s. Instead the distribution of the error terms develops in an unusual way.

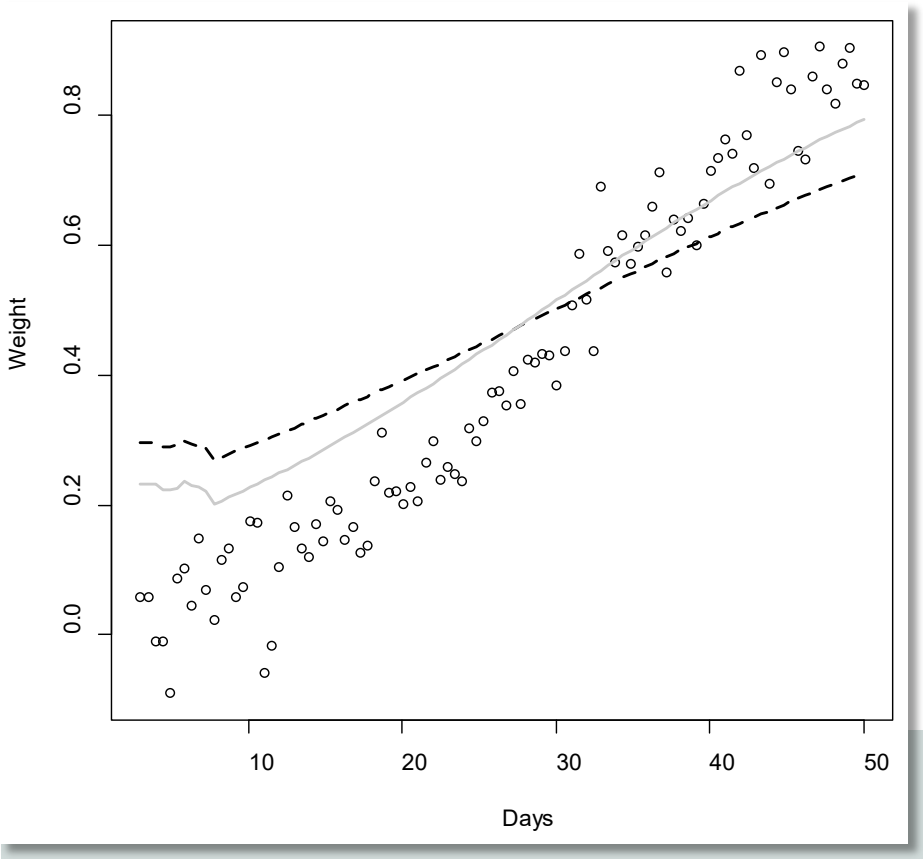

Figure 14: Chicken-Growth Data with Outliers: NLR and MLP Forecasts

Again, the NLR model is shown as a solid line but in grey for reasons soon to become clear. Two of the many approaches to cope with this challenging situation are applied here.

7 Change of Target Function for MLP and NLR

For the MLP the aim is to minimize the Mean Absolute Error (MAE) instead of the MSE. In the case of NLR, I switch to Non-Linear Quantile Regression (NLQR) and focus on the median. The results are shown below (dashed resp. solid lines but lowered).

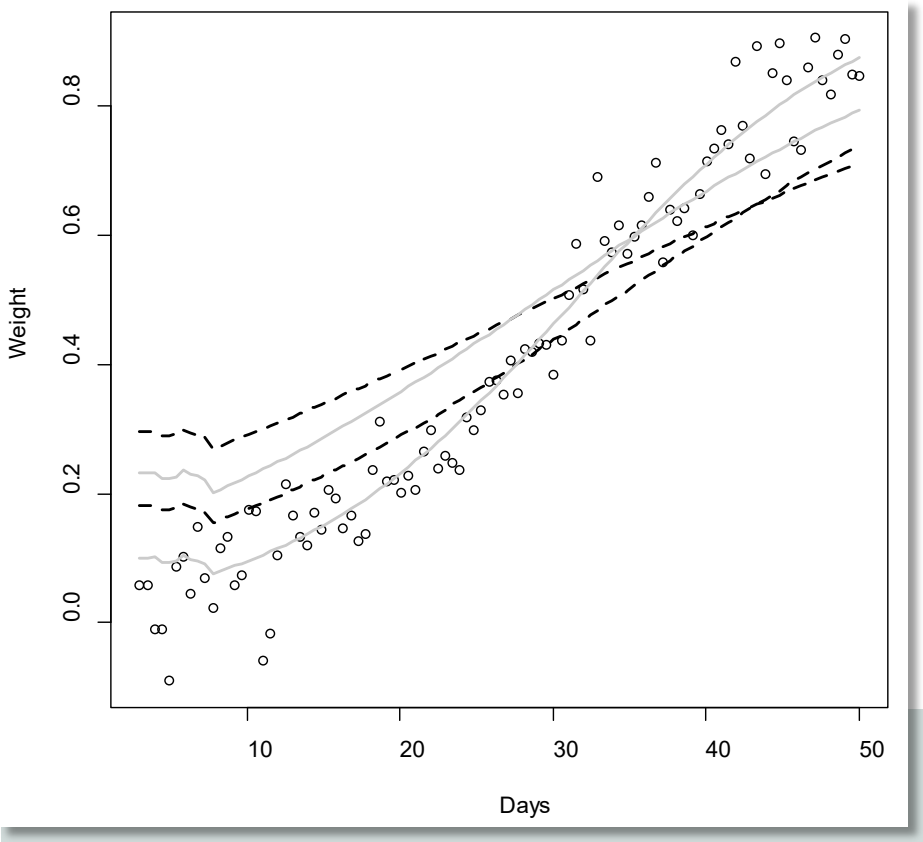

Figure 15: Data with Outliers: NLR/NLQR and Two MLP Forecasts

The corresponding parameters are as follows:

Parameter	Model / Special	b_0	b_1
True Value		3.5	-0.11
Estimated Value	NLR / n.a.	1.8781	-0.0646
Trained Value	MLP / n.a.	1.3428	-0.0451
Estimated Value	NLR / NLQR	3.3010	-0.1050
Trained Value	MLP / MAE	2.1913	-0.0647

Table 6: Corresponding Parameters

The model least confused is NLQR.[28]

Insights from this section

If there are outliers, an MLP may fail to learn a relationship; NLQR performs better. This shows the value of benchmarking MLP with other available techniques. The reason for the inadequate learning of the MLP may be the learning algorithm. The choice of the target function should be pondered in light of the problem to be solved.

[28] Another fruitful application of quantile regression can be found in Lehrbass (2020c).

8 New Data and Linear Regression

This time I postpone the revelation of the DGP to the end for reasons to become clear. Let us assume we are working with a completely new data set containing two time series. They look as follows:

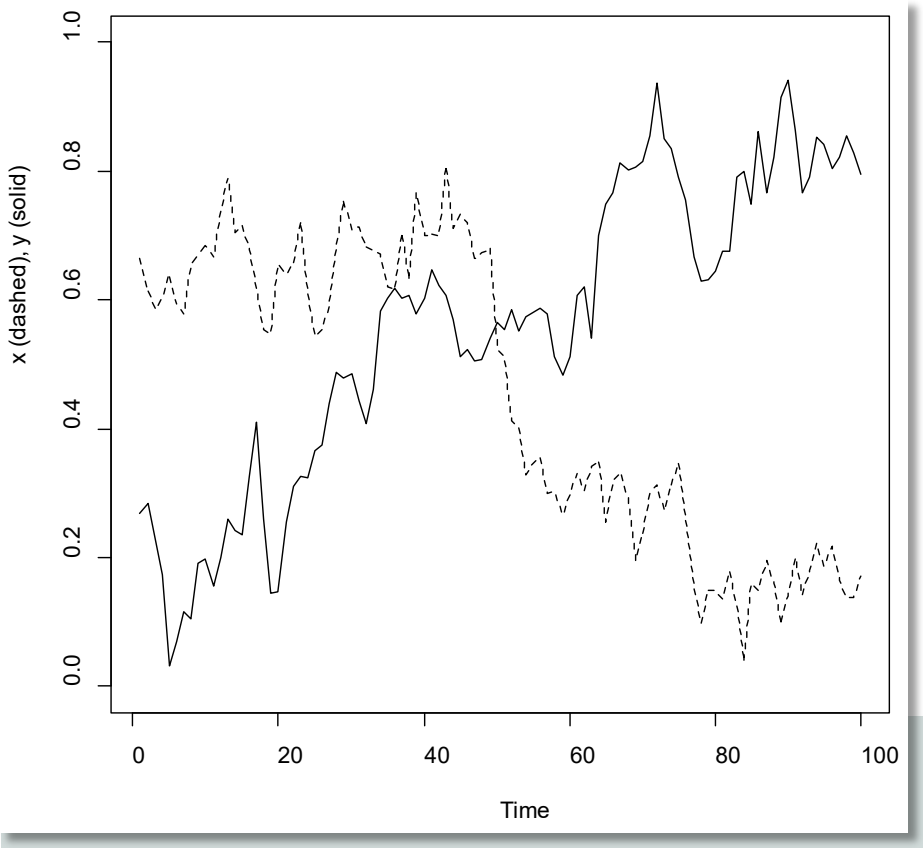

Figure 16: New Data Containing Two Time Series

It seems as if there were a negative relationship between both series. When *x* goes down, *y* goes up.

Visual mapping of *x* to *y* yields the next chart:

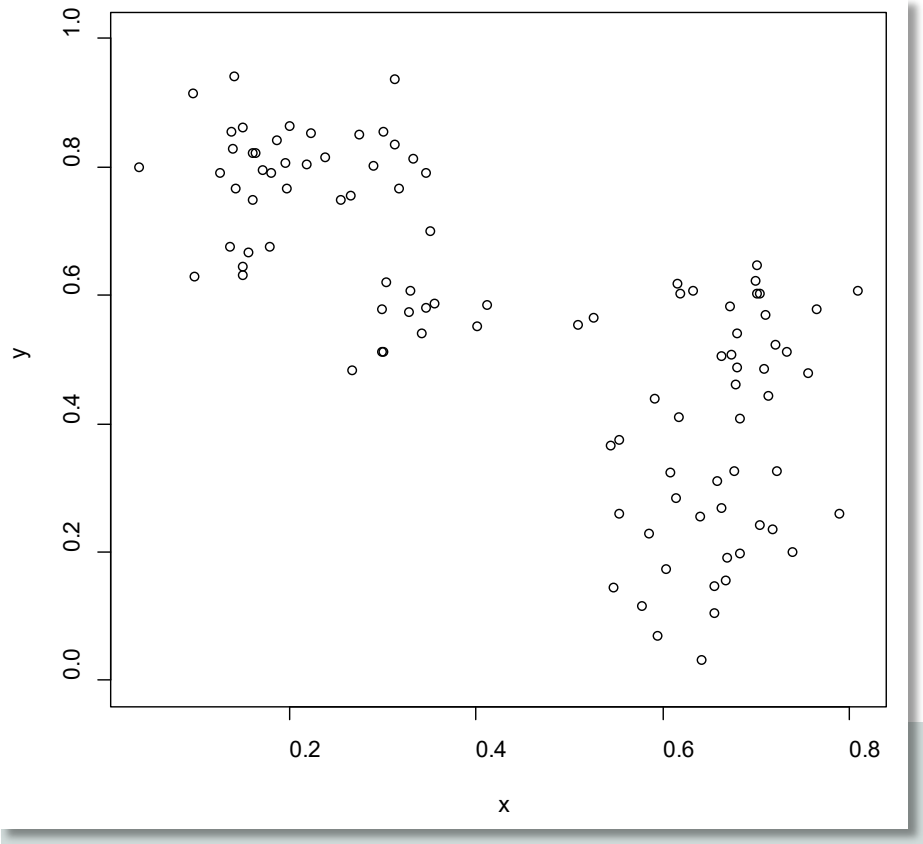

Figure 17: New Data Visual Mapping x to y

It appears sensible to apply linear regression. The estimated function is plotted against the data in the next chart:

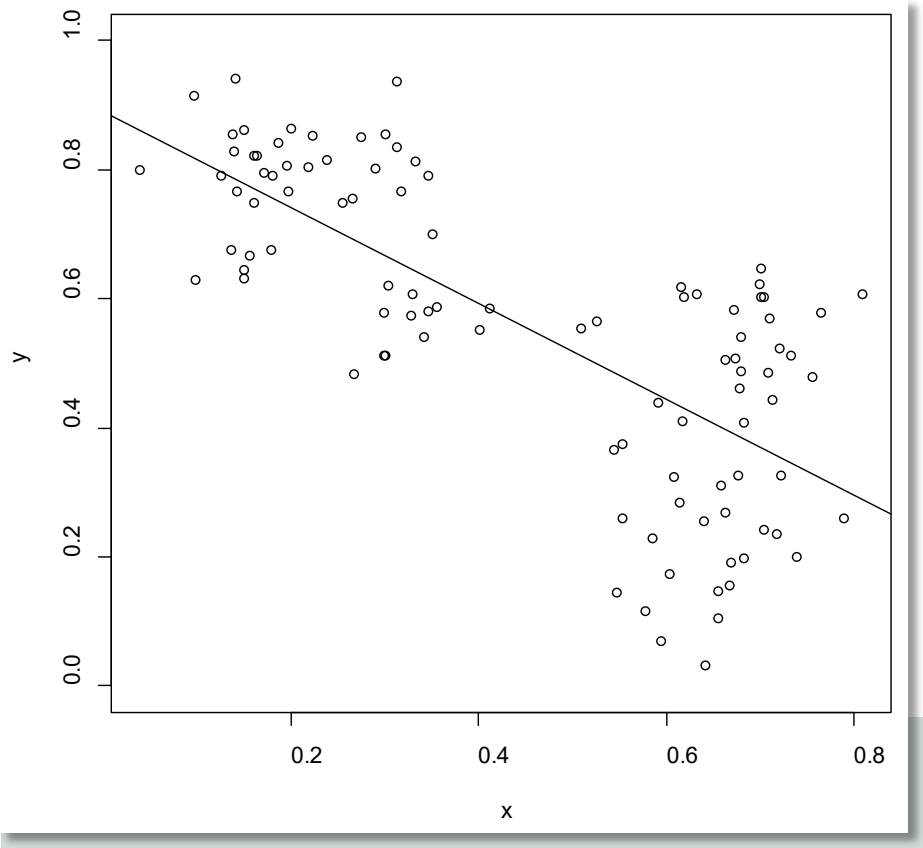

Figure 18: New Data: Linear Regression

The solid line is the regression line. I interpret the output from an ordinary least squares estimation.

```
Call:
lm(formula = y_new ~ x_new)
Residuals:
     Min       1Q   Median       3Q      Max
-0.38134 -0.11633  0.02338  0.12089  0.31747
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.89071    0.03580   24.88   <2e-16 ***
x_new       -0.74340    0.07092  -10.48   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1
Residual standard error: 0.1607 on 98 degrees of freedom
Multiple R-squared:  0.5285, Adjusted R-squared:  0.5237
F-statistic: 109.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

There is an impressively high R^2 of above 52%. Each parameter appears highly significant, i.e. one may reject the H0 that both are zero with high confidence.[29] Of course, one needs to do diagnostics. First, I start to check the autocorrelation of the residuals.

[29] The F-Test does not add value because we face a simple linear regression, i.e. the F-statistic is the square of the t-value of -10.48.

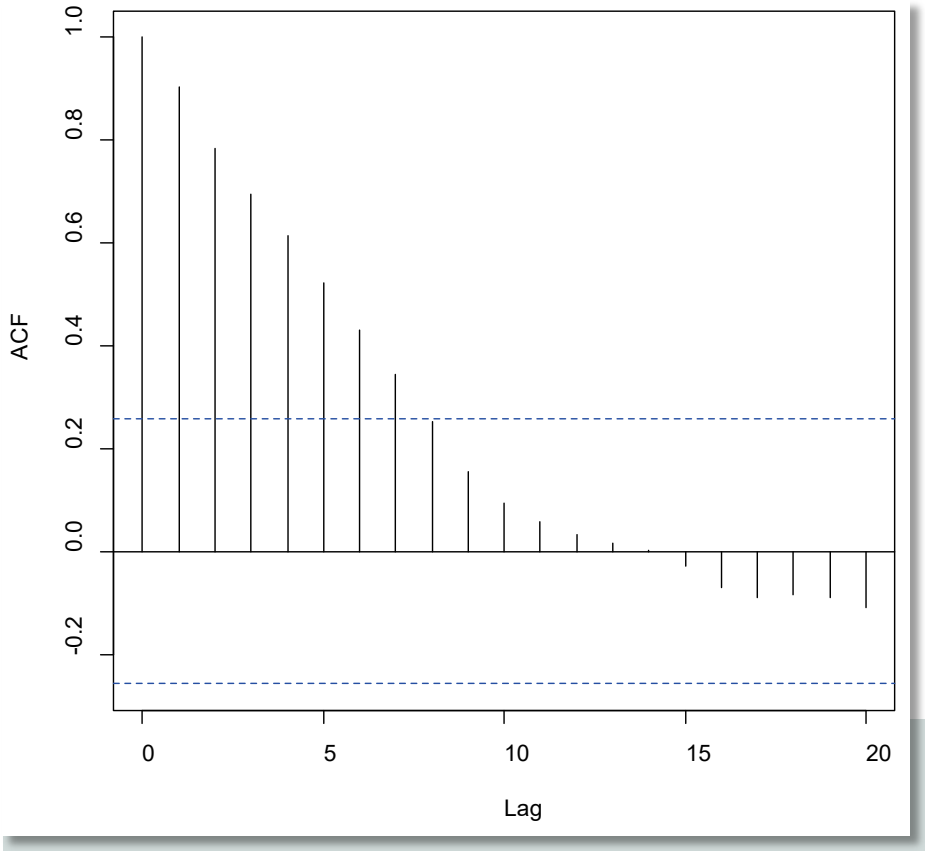

Figure 19: New Data: Regression Residuals

These findings are surprising. Assumption 3 (I.I.D. Error Term) is violated. Since the autocorrelations do not seem to level off, it is no solution to use autocorrelation consistent standard errors by Newey and West.[30] Therefore, the results deserve further consideration. If there were a linear relation, our findings would have been as follows:

$$y_t = a + bx_t \qquad (7)$$

One could also look at the equation for t-1. Deducting the "lagged" equation from (7) leads to the following equation for the differenced variables:

$$\Delta y_t = b\Delta x_t \qquad (8)$$

[30] Newey W. K., West K. D. (1987).

Hence, if there were a linear relation, the regression based on the differenced variables would have to yield the same estimates for the slope coefficient for *x,* which I have labeled *b.* Again, I interpret the output from an ordinary least squares estimation.

```
Call:
lm(formula = diff(y_new) ~ diff(x_new))
Residuals:
      Min        1Q    Median        3Q       Max
-0.158946 -0.033106  0.001167  0.032105  0.155187
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.005166   0.005701   0.906    0.367
diff(x_new) -0.028854   0.101244  -0.285    0.776
Residual standard error: 0.0565 on 97 degrees of freedom
Multiple R-squared:  0.0008367,    Adjusted R-squared:
-0.009464
F-statistic: 0.08122 on 1 and 97 DF,  p-value: 0.7763
```

The R^2 has fallen to zero and the estimated slope is different from -0.7434. All of this is a red flag, which prohibits the assumption of a relationship between input *x* and output *y*.

9 New Data and MLP

The same architecture as before is applied together with RMSProp. The trained function is plotted below:

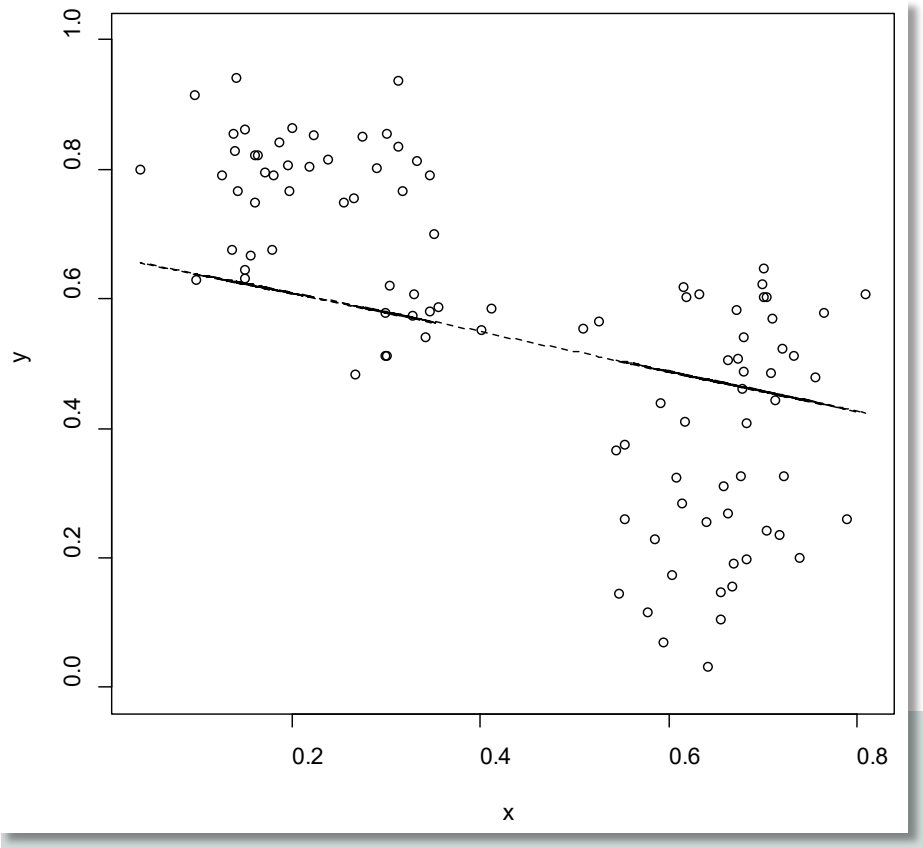

Figure 20: New Data: MLP

What has been learned is something akin to a linear function. The squared correlation (R^2) is even higher as for the linear regression. It amounts to a bit more than 52%. The similarity to the linear regression is no surprise. Often the high R^2 is taken as proof of success. But the very first step of diagnostics points to severe problems:

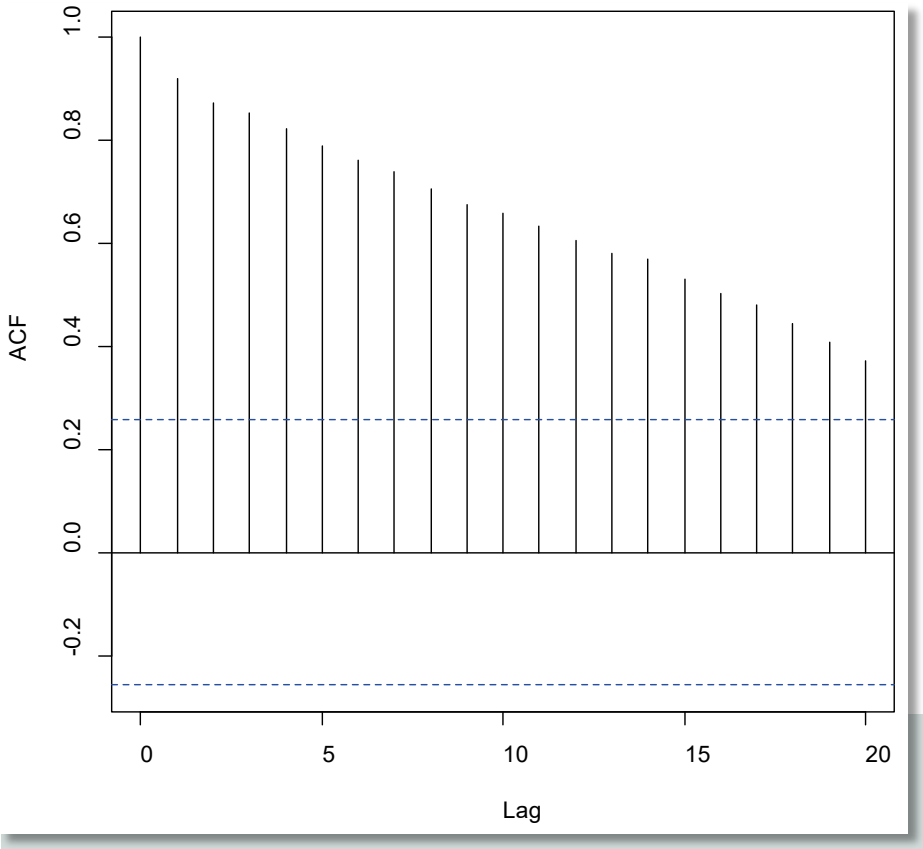

Figure 21: New Data: MLP Residuals

Diagnostics have revealed a problem, which is made transparent in the next section.

10 Data-Generating Process III

The new data has been generated as follows: I chose a sample size of 100 pairs of *y* and *x*. Two random walks (r = *y, x*) were sampled according to the following equation, where *z* is normal white noise and *t* a time index starting at t=1:

$$r_t = r_{t-1} + z_t \qquad (9)$$

Note that by construction there is no dependence between *y* and *x* because each time series is generated independently of the other.[31]

Insights from this section

If there is no relationship, an MLP may still learn one and produce impressive results in terms of R^2. Only diagnostics reveal that one faces spurious learning.

[31] In fact, two stochastic trends are at work and evoke the impression of a relationship.

11 Outlook: Hypothesis Testing for MLP as for NLR

As concerns testing the null hypothesis that a model parameter is equal to zero, one requires more background knowledge. White has shown that if there are no redundant hidden neurons, the weights of the MLP $\widehat{W}$ converge with increasing sample size in distribution to multivariate normal (1989, 1007, Theorem 4.1). As I have chosen the smallest possible MLP I may assume the absence of redundant neurons.

Hence, it is possible to calculate test statistics as before once I have got the standard errors. "Motivated by the desire to obtain distributional results for $\widehat{W}$ that rely neither on large *n* approximations nor on artificial data generating assumptions, statisticians have developed so-called 'resampling techniques' that permit rather accurate estimation of finite sample distributions for $\widehat{W}$" (White, 1992, 110). The test statistics derived by resampling are "in most cases … more accurate than ones based on asymptotic theory" (Davidson, MacKinnon, 2009, 249). With the help of bootstrapping, which is a resampling technique, one can generate standard errors.

I want to illustrate this for the chicken-growth data, which is used to benchmark an MLP by NLR, and use the function nlsBoot() to get bootstrap intervals.[32]

```
Parameter Std. error
b0  0.110419201
b1  0.003443185
```

It is possible to do something similar for the MLP. I recommend applying the "direct method" (Maddala, Lahiri, 2009, 607). Based on each sample one could train the MLP and store the learned weights. Then one could calculate the standard deviation of the bootstrap distribution of the weights.[33] Thus, one could estimate the standard error of each weight by the standard deviation[34] of the bootstrap sample. Since this is just an outlook I do not elaborate further.

[32] I use the default sample size of one thousand.

[33] Another useful recommendation is to use the mean weight as recommended in Maddala, Lahiri (2009, 606).

[34] Using a denominator of 999.

Insights from this section

Hypothesis testing for the learned weights is possible. They are random variables and there are ways to quantify confidence intervals.

12 Diagnostic Steps and Common Wisdom

We summarize the steps of the diagnostics and add comments, why these properties of the residuals are of practical value, too.

Check	Practical Value
The mean of the residuals is zero	There is no systematic forecasting error
No autocorrelation among residuals	There is no unused structure that I have not yet integrated into the model
No variation in the variance of the residuals	One may assume that each pair *y, x* in the sample is equally valid

Table 7: Summary and Practical Value

Insights from this section

Diagnostics encompass the analysis of the residuals, which means "learning from your mistakes". If the diagnostics do not put the assumptions into doubt, one may enjoy the trained MLP as a consistent estimator of the true relationship. This is the main benefit of diagnostics.

13 Putting the Findings into Context

Effectively, an MLP excels "at two things: memorizing training examples, and interpolating within a cloud of points that surround those examples in some cluster of a hyperdimensional space (…) but they generalize poorly outside the training space" (Marcus, 2020, 12). There is the Universal Approximation Theorem (Goodfellow et al., 2016, 192): Every continuous function *y(x)* can be learned by an MLP with one hidden layer. Unfortunately, as has been seen, this also implies learning noise within the training space.

In the last example, the MLP did not indicate that there is no relationship between input x and output y. An educated human being can do so by applying diagnostics. What we have encountered is the classic example of a spurious regression (Granger, Newbold, 1974). Up to the year 1974, such regressions went unnoticed. This means that since the invention of linear regression hundreds of years ago, a large number of regressions were spurious. I hope that it will not take as long to detect spurious MLP.

As of the time of writing, there are few diagnostics in place at Microsoft Azure Machine Learning Studio (classic). Therefore, users have to program diagnostics in RStudio and insert an "Execute R Script" module into their experiment in their workspace in the cloud. This underlines the argument that it is much easier to "drag and drop" another machine learning model into the experiment than to establish decent diagnostics.

With a grain of salt, this is valid for other machine learning service providers such as Amazon Web Services and Google Cloud Platform. It could be one of the reasons for the following statement: A "growing body of evidence shows that state-of-the-art models learn to exploit spurious statistical patterns in datasets" (Nie et al, 2019, 1).

Insights from this section

Be aware that, as a service, "on the shelf" software might need a topping by diagnostics which are not readily available.

14 Explainability versus Diagnostics

Roscher et al. "differentiate between transparency, interpretability, and explainability. Roughly speaking, transparency considers the ML approach, interpretability considers the ML model together with data, and explainability considers the model, the data, and human involvement" (2019, 1).

These concepts help to sharpen our investigation. In the first example above (a) transparency is given, which I explore in the order of the machine learning workflow: The net architecture and the hyper-parameters of the training process can be made explicit as well as the algorithms used for training. The search process during training can be made transparent, e.g. by plotting the weights[35] and corresponding error measures. The resulting network can be made explicit by revealing the optimal weights.

"The aim of (b) interpretability is to present some of the properties of an ML model in understandable terms to a human" (Roscher et al., 2019, 5). The first example is interpretable. It concerns the question of how chickens grow over the first 50 days. One can also work with sensitivities, i.e. changing each input by one percent and measuring the percentage change in the forecast. Combinations of changes might help to illustrate interaction effects.

To proceed towards an explanation we again consider the first example. There are two sides of the coin: One is biology, the other mathematics. Although the first is common knowledge, the second is not. To understand the mathematical background of NLS, one needs to grasp the theory of best approximation from linear algebra (relates to least squares) and to have at least a basic understanding of statistics (relates to statistical inference). It becomes clear that a full explanation often goes beyond available resources.

Insights from this section

Even if one has to make it through transparency, interpretability, and explainability of an MLP, one might still encounter spurious learning. Diagnostics cannot be avoided.

[35] Although I have not done this here, I would recommend it in light of my own experience. Moreover, it is advisable to plot MSE or what is of interest for the validation and training set parallel to the plotting of weights.

15 Conclusion

Training an MLP to achieve a minimum level of MSE is akin to doing NLR with NLS. Therefore, this study has made use of the available econometric theory of NLR and the tools in R. Only if certain assumptions about the error term in the DGP are in place, is it possible to enjoy the trained MLP as a consistent estimator of the true relationship between *x* and *y*.

Using controlled experiments I have shown that even in an ideal setting an MLP may fail to learn a relationship and that NLR performs better. An MLP might also lead to deceiving results when there are outliers in the data. Again the traditional econometric technique of NLQR fared better. Most striking is the case in which there is no relationship but the MLP learns a relationship producing impressive results in terms of R^2 nevertheless. Only with the help of diagnostics is it possible to become aware of spurious learning.

Therefore, deep learning methods should be applied with some degree of caution. One should not be too quick to state that the MLP has learned something. Besides one should be wary of using machine learning without adding necessary diagnostics and considering all the implications. Aiming for transparency, interpretability, and explainability is also important but no substitute for diagnostics.

Literature

Barnes, J. (2015): Azure Machine Learning. Microsoft Press
Beysolow II, T. (2017): Introduction to Deep Learning Using R. Apress Media
Box, G, Jenkins, G. (1976): Time Series Analysis, Forecasting, and Control. Holden-Day
Chen, T. et al. (2015): MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems, https://arxiv.org/abs/1512.01274 (accessed 12 Nov 2020)
Davidson, R., MacKinnon, J. G. (2009): Econometric Theory and Methods, Oxford University Press
German Research Center for Artificial Intelligence (no year): Explainable AI and Neural Networks, https://www.dfki.de/en/web/research/projects-and-publications/projects-overview/projekt/explainn/ (accessed 16 Nov 2020)
Ghatak, A. (2019): Deep Learning with R. Springer Singapore
Goodfellow, I., Bengio, Y., Courville, A. (2016): Deep Learning. Massachusetts Institute of Technology Press
Granger, C. W. J., Newbold, P. (1974): Spurious Regression in Econometrics, in: Journal of Econometrics 2, 111-120
Hornik, K., Stinchcombe, M., White, H. (1989): Multilayer Feedforward Networks are Universal Approximators, Neural Networks 2, 359-366
Jennrich, R. I. (1969): Asymptotic Properties of Non-Linear Least Squares Estimators, in: Ann. Math. Statist. 40, no. 2, 633-643
Lehrbass, F., Peter, M. J. (1996): DAX-Futures-Trading mit künstlichen Neuronalen Netzen, in: Zeitschrift für das gesamte Kreditwesen 4, 4-16
Lehrbass, F. (2020a): Dangerous Deep Learning: How The Machines Can Hit The Wall, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3607952 (accessed 12 Nov 2020)
Lehrbass, F. (2020b): Sales Forecasting: Ein Vergleich von ökonometrischen Methoden und Machine Learning, in: Buchkremer, R. et al. (eds.): Künstliche Intelligenz in Wirtschaft und Gesellschaft. Springer Gabler
Lehrbass, F. (2020c): Analyzing Promotion Effectiveness in Fashion Retailing Using Quantile Regression, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3576434 (accessed 12 Nov 2020)
Lehrbass, F., Schuster, T. (2021): Deviations from Covered Interest Rate Parity: The case of British Pound Sterling versus Euro, in: Journal of Financial Data Science, Volume 3, Issue 1, https://jfds.pm-research.com/content/early/2020/12/17/jfds.2020.1.050 (accessed 15 February 2021)
Liu, Y., Maldonado, P. (2018): R Deep Learning Projects. Packt
Maddala, G. S., Lahiri, K. (2009): Introduction to Econometrics. Wiley

Marcus, G. (2020): The next decade in AI: Four Steps towards robust AI, https://arxiv.org/ftp/arxiv/papers/2002/2002.06177.pdf (accessed 9 March 2020)

Markets and Markets (2020): Analytics as a Service Market by Component, Deployment Mode, Organization Size, Industry Vertical (BFSI, Telecommunications and IT, Healthcare and Life Sciences, and Retail and eCommerce), and Region – Global Forecast to 2024, https:// ://www.marketsandmarkets.com/Market-Reports/analytics-as-a-service-market-159638048.html (accessed 13 Nov 2020)

MXNet (no year, a): API for R, https://mxnet.apache.org/versions/1.6/api/r/docs/api/ (accessed 13 Nov 2020)

MXNet (no year, b): Develop a Neural Network with MXNet in Five Minutes, https://mxnet.apache.org/versions/1.6/api/r/docs/tutorials/fiveMinutesNeuralNetwork.html (accessed 13 Nov 2020)Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., Kiela, D. (2019): Adversarial NLI: A New Benchmark for Natural Language Understanding, https://arxiv.org/pdf/1910.14599 (accessed 5 May 2020)

Newey W. K., West K. D. (1987): A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix, in: Econometrica, 55, 703-708.

Riazoshams, H., Midi, H., Ghilagaber, G. (2019): Robust Nonlinear Regression: with Applications using R, Wiley

Roscher, R., Bohn, B., Duarte, M. F., Garcke, J. (2019): Explainable Machine Learning for Scientific Insights and Discoveries, https://arxiv.org/pdf/1905.08883 (accessed 5 May 2020)

Verbeek, M. (2012): A Guide to Modern Econometrics, Wiley

Volmer, R., Lehrbass, F. (1997): Kohonens selbstorganisierende Karten und der Terminkontrakt auf den DAX, in: WIRTSCHAFTSINFORMATIK, 39, 339-343

White, H. (1989): Some Asymptotic Results for Learning in Single Hidden Layer Feedforward Network Models, in: Journal of the American Statistical Association, 84, 1003-1013

White, H. (1992): Artificial Neural Networks: Approximation and Learning Theory. Blackwell

kostenloser Download
unter **fom-ifes.de**

Lehrbass, F. / Wörndl, F. (2021): Was treibt die Renditen von Hedgefonds? Eine empirische Untersuchung ausgewählter Hedgefonds Strategien, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 23, 2021, ISSN (eBook) 2569-5355, ISBN (eBook) 978-3-89275-422-0

Kladroba, A. / Friz, K. / Buchmann, T. / Wolf, P. (2020): Netzwerk- und Outputmessung – Indikatorik für transformative Technologiefelder (NEO-Indikatorik), in: Krol, B. / Kladroba, A. (Hrsg.), ifes Schriftenreihe, Band 22, 2020, ISSN (eBook) 2569-5355, ISBN (eBook) 978-3-89275-420-6

Bähren, T. / Maasjosthusmann, R. / Walter, A. / Lehrbass, F. (2020): Praktische Umsetzung von Business Analytics im Mediensektor: Predictive Analytics im Filmgeschäft, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 21, 2020, ISSN (eBook) 2569-5355, ISBN (eBook) 978-3-89275-418-3

Kladroba, A. (2019): Der Einfluss mathematischer Methoden auf das Ergebnis von Mannschaftswettkämpfen: Eine Simulationsrechnung, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 20, 2019, ISSN (eBook) 2569-5355, ISBN (eBook) 978-3-89275-416-9

Raasch, A. / Lehrbass, F. (2019): Investmentstrategien im Rahmen von Übernahmen börsennotierter Gesellschaften – Merger Arbitrage und Maschinelles Lernen, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 19, 2019, ISSN 2191-3366, ISBN 978-3-89275-413-8

Hagemann, D. / Lehrbass, F. (2018): Prognosemodelle für Länderrisiken: Logit- und Deep Learning-Methoden im Vergleich, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 18, 2018, ISSN 2191-3366, ISBN 978-3-89275-411-4

Graalmann, M.-P. / Lehrbass, F. (2018): Eignung von Varianz-Kovarianz-Ansätzen und Copula-Modellen zur Risikoaggregation in bankaufsichtlichen Risikotragfähigkeitskonzepten, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 17, 2018, ISSN 2191-3366, ISBN 978-3-89275-409-1

Cox, P. / Lehrbass, F. (2018): Determinanten der Replikationsgüte von Exchange Traded Funds, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 16, 2018, ISSN 2191-3366, ISBN 978-3-89275-407-7

Lehrbass, F. / Scheipers, N. (2017): Determinanten der Höhe von Wirtschaftsprüfungshonoraren am Beispiel von gelisteten Unternehmen im Prime Standard, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 15, 2017, ISSN 2191-3366, ISBN 978-3-89275-406-0

Schwarz, J. (2017): Ergebnisse der Analyse von Studienabbrüchen, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 14, 2017, ISSN 2191-3366, ISBN 978-3-89275-405-3

Lehrbass, F. (2016): Risikomessung für den globalen Kohlehandel: Einfache und fortgeschrittene Verfahren nebst Backtesting sowie ein Vergleich mit IFRS 7, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 13, 2016, ISSN 2191-3366, ISBN 978-3-89275-404-6

Godbersen, H. (2016): Die Means-End Theory of Complex Cognitive Structures – Entwicklung eines Modells zur Repräsentation von verhaltensrelevanten und komplexen Kognitionsstrukturen für die Wirtschafts- und Sozialwissenschaften, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 12, 2016, ISSN 2191-3366, ISBN 978-3-89275-403-9

Seng, A. / Landherr, G. (2015): Vielfalt leben und Vielfalt gestalten – Diversity Management in der Lehre, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 11, 2015, ISSN 2191-3366, ISBN 978-3-89275-402-2

Gansser, O. A. / Schutkin, A. (2014): Studie zur Validierung der Persönlichkeitsmerkmale Abenteuerlust und Routineverhalten, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 10, 2014, ISSN 2191-3366, ISBN 978-3-89275-401-5

Gansser, O. A. (2014): Marketingplanung als Instrument zur Krisenbewältigung, in: Krol, B. (Hrsg.), ifes Schriftenreihe, Band 9, 2014, ISSN 2191-3366, ISBN 978-3-89275-400-8

Runia, P. M. / Wahl, F. / Rüttgers, C. (2013): Das Markenimage von Hersteller- und Handelsmarken: Eine empirische Analyse der Imagekomponenten von Körperpflegemarken auf der Grundlage eines Markenidentitätskonzeptes, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 8, 2013, ISSN 2191-3366

Naskrent, J. / Rüttgers, C. (2013): Sportmonitor Essen 2013: Eine empirische Analyse über das Image regionaler Sportvereine und ihre Sponsoring- und Promotionangebote, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 7, 2013, ISSN 2191-3366

Seng, A. / Fiesel, L. / Rüttgers, C. (2013): Akzeptanz der Frauenquote, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 6, 2013, ISSN 2191-3366

Naskrent, J. / Rüttgers, C. (2012): Wahrnehmung von Werbung mit Sportereignisbezug: Eine empirische Analyse der Einschätzung von Sponsoring und Ambush-Marketing im Rahmen der Fußball-Europameisterschaft und der Olympischen Spiele im Jahr 2012, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 5, 2012, ISSN 2191-3366

Seng, A. / Fiesel, L. / Krol, B. (2012): Erfolgreiche Wege der Rekrutierung in Social Networks, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 4, 2012, ISSN 2191-3366

Heinemann, S. / Krol, B. (2011): Nachhaltige Nachhaltigkeit: Zur Herausforderung der ernsthaften Integration einer angemessenen Ethik in die Managementausbildung, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 2, 2011, ISSN 2191-3366

Hermeier, B. / Rettig, P. / Krol, B. (2010): Marken- und Produktmanagement durch Nutzung von Sportgroßereignissen: Möglichkeiten und Grenzen für Industrie und Handel, in: Krol, B. (Hrsg.), KCS Schriftenreihe, Band 1, 2010, ISSN 2191-3366

ISBN (Print) 978-3-89275-423-7 ISSN (Print) 2191-3366
ISBN (eBook) 978-3-89275-424-4 ISSN (eBook) 2569-5355

Institut für Empirie & Statistik
der FOM Hochschule
für Oekonomie & Management

FOM Hochschule

FOM. Die Hochschule. Für Berufstätige.

Die mit bundesweit über 57.000 Studierenden größte private Hochschule Deutschlands führt seit 1993 Studiengänge für Berufstätige durch, die einen staatlich und international anerkannten Hochschulabschluss (Bachelor/Master) erlangen wollen.

Die FOM ist der anwendungsorientierten Forschung verpflichtet und verfolgt das Ziel, adaptionsfähige Lösungen für betriebliche bzw. wirtschaftsnahe oder gesellschaftliche Problemstellungen zu generieren. Dabei spielt die Verzahnung von Forschung und Lehre eine große Rolle: Kongruent zu den Masterprogrammen sind Institute und KompetenzCentren gegründet worden. Sie geben der Hochschule ein fachliches Profil und eröffnen sowohl Wissenschaftlerinnen und Wissenschaftlern als auch engagierten Studierenden die Gelegenheit, sich aktiv in den Forschungsdiskurs einzubringen.

Weitere Informationen finden Sie unter **fom.de**

ifes

Zunehmende Digitalisierung erfordert und ermöglicht datenbasierten Erkenntnisgewinn und fundiertes unternehmerisches Handeln. Um aus den allgegenwärtigen Daten die richtigen Schlüsse zu ziehen, ist überall eine kritische Methodenkompetenz erforderlich. Der wissenschaftliche Fokus der ifes-Akteure liegt dabei in den Bereichen der empirischen Unternehmens-, Markt- und Konsumentenforschung, der angewandten Statistik, des Data Minings und der Finanzstatistik.

Das ifes verfolgt das Ziel, empirische Kompetenzen an der FOM zu bündeln und die angewandte Forschung im empirischen Bereich der Hochschule weiter voranzutreiben. Damit nimmt das ifes eine zentrale Stellung im Bereich der Entwicklung und Unterstützung der Methodenausbildung in der Lehre der Bachelor- und Masterstudiengänge sowie im Promotionsprogramm der FOM ein.

Weitere Informationen finden Sie unter **fom-ifes.de**

SCIENCE BLOG

Im Forschungsblog werden unter dem Titel „FOM forscht" Beiträge und Interviews rund um aktuelle Forschungsthemen und -aktivitäten der FOM Hochschule veröffentlicht.

Besuchen Sie den Blog unter fom-blog.de