

Neuronale Netze – Was Aktuare schon wissen und was sie noch wissen sollten

Ralf Korn

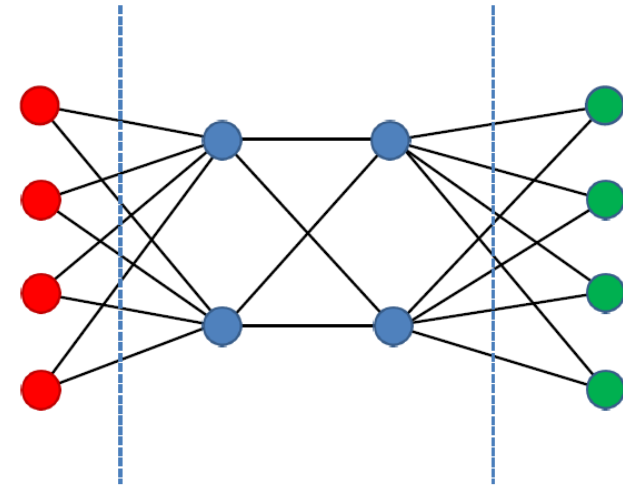
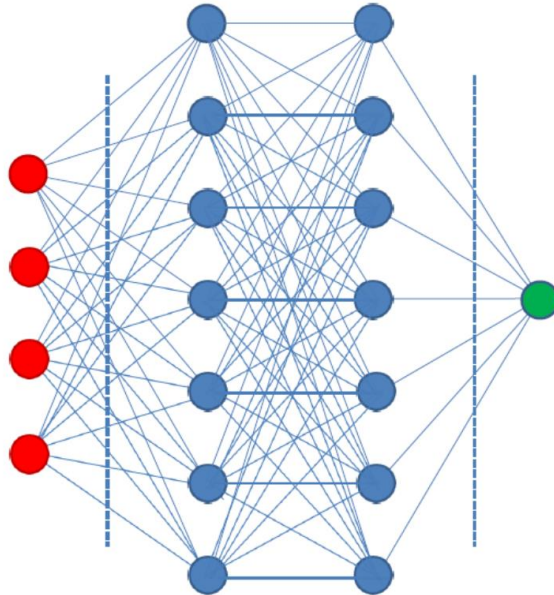
TU Kaiserslautern, Fraunhofer ITWM, EI-QFM

KI, Data Science, Neuronale Netze – Hype, Hilfe oder Gefahr?

- Die Bundesregierung hat ein nationales KI-Programm mit bis zu neuen 100 Professorenstellen aufgelegt
- Versicherer stellen verstärkt Data Scientists ein (wirklich?)
- Neuronale Netze sind in Bildverarbeitung, Schrifterkennung, ... zu Standardwerkzeugen mit ausgezeichneten Performance geworden
- Neuronale Netze werden in Kerngebieten der Versicherungsmathematik eingesetzt z.B.:
 - M. Wüthrich (2018) Neural networks applied to chain ladder reserving. EAJ 8(2), 407-436.
 - D. Hainault (2016) A neural-network analyzer for mortality forecast. ASTIN Bull 48(2), 481-508.

=> **Braucht es noch Aktuare/Versicherungsmathematiker ?**

Neuronale Netze in Bildern



● = Inputvariablen, ● = Neuronen (in den **Hidden Layers**), ● = Outputvariablen

Neuronale Netze –nüchtern ...

Mathematisch ausgedrückt:

Neuronale Netze lösen die Approximationsaufgabe

$$y = f(x) \approx \hat{f}(x; \theta)$$

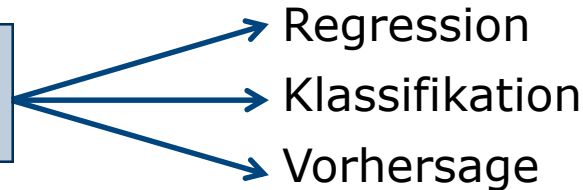
auf eine spezielle Art.

Neuronale Netze –nüchtern ...

Mathematisch ausgedrückt:

Neuronale Netze lösen die Approximationsaufgabe

$$y = f(x) \approx \hat{f}(x; \theta)$$



auf eine spezielle Art.

=>

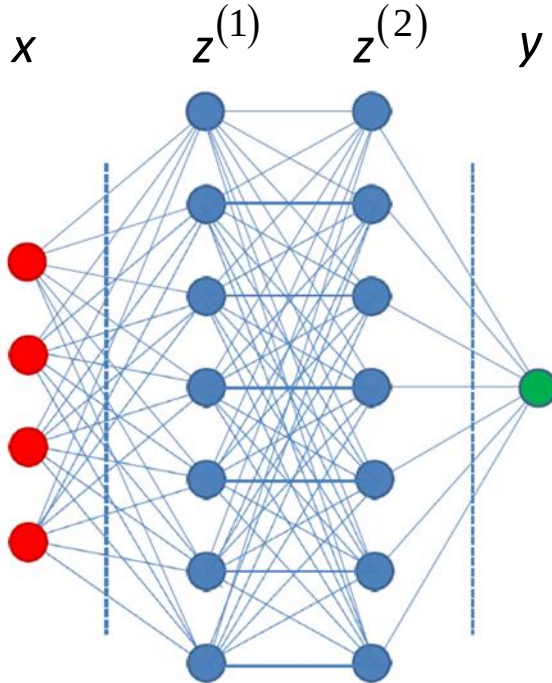
- Neuronale Netze sind nicht-lineare Regressionsverfahren.
- Die beiden wesentlichen Aufgaben sind die Wahl von $\hat{f}(\cdot; \cdot)$ (der *Topologie des Netzes* und der *Aktivierungsfunktion*) und der *Gewichte* θ

Neuronale Netze – Wollt Ihr Mathe oder was?

Mögliche Schwerpunkte des Vortrags:

- Die Mathematik hinter neuronalen Netzen **S1**
- Die Sprache hinter neuronalen Netzen: Klartext für Aktuarien **S2**
- Was macht neuronale Netze attraktiv? **S3**
- Sind neuronale Netze wirklich so gut? **S4**
- Werden Aktuarien im Zeitalter neuronaler Netze überflüssig? **S5**

Neuronale Netze – Wir basteln uns ein Netz



$x \in \mathbb{R}^4$ Inputvariablen

$g : \mathbb{R} \rightarrow \mathbb{R}$ Aktivierungsfunktion

Perzeptron

$W^{(1)} \in \mathbb{R}^{7,4}$ Erste Gewichtsmatrix, $b^{(1)} \in \mathbb{R}^7$

$z_i^{(1)} := g(w_i^{(1)} x + b_i^{(1)})$ Neuronen der ersten Ebene

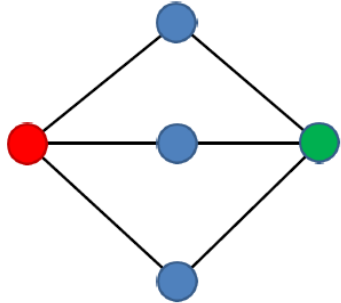
$W^{(2)} \in \mathbb{R}^{7,7}$ Zweite Gewichtsmatrix, $b^{(2)} \in \mathbb{R}^7$

$z_i^{(2)} := g(w_i^{(2)} z^{(1)} + b_i^{(2)})$ Neur. der zweiten Ebene

$w \in \mathbb{R}^7$ Output-Gewichtsvektor, $b \in \mathbb{R}$

$y = w^T z^{(2)} + b$ Outputvariable

Neuronale Netze – Hoch oder tief?



$$y = b + \sum_{i=1}^3 w_i g\left(w_i^{(1)} x + b_i^{(1)}\right) \quad \text{gewichtete Summe von Neuronen}$$

Flaches („Shallow“) Netz



$$y = b + w g\left(w^{(2)} g\left(w^{(1)} x + b^{(1)}\right) + b^{(2)}\right) \quad \text{iterierte Nicht-Linearität}$$

Tiefes („Deep“) Netz

Neuronale Netze – Warum überhaupt?

Satz von Kolmogoroff (1957)

Jede stetige multivariate, reellwertige Funktion besitzt eine Darstellung

$$(1) \quad f(x_1, \dots, x_n) = \sum_{q=0}^{2n} g_q \left(\sum_{p=1}^n h_{q,p}(x_p) \right)$$

mit $g_q(x)$, $h_{q,p}(x)$ stetig, reellwertig, $h_{q,p}(x)$ unabh. von $f(x_1, \dots, x_n)$ wählbar.

⇒ Spricht für Netze mit **2n** Neuronen und nur **einem** Hidden Layer

⇒ Die Funktionen $g_q(x)$ können **alle gleich** gewählt werden (Lorentz(1966))

⇒ Die Fkt. $h_{q,p}(x)$ können (i.W.) **gleich** gewählt werden (Sprecher(1965),...)

Aber: Alle vorkommenden Funktionen haben keine einfache Form!

Neuronale Netze – Warum überhaupt?

2

Also:

Bei einem neuronalen Netz wird in der Kolmogoroff-Darstellung

$$f(x_1, \dots, x_n) = \sum_{q=0}^{2n} g_q \left(\sum_{p=1}^n h_{q,p}(x_p) \right)$$

die innere Summe durch eine **affin-lineare Funktion** in erster Ordnung approximiert und die äußere Funktion durch eine spezielle (iterierte) Nicht-Linearität (**die Aktivierungsfunktion**) angenähert.

Universelle Approximationseigenschaft (Hornik et al. (1989))

Jede Borel-messbare Funktion kann durch ein flaches Netz mit hinreichend vielen Knoten beliebig genau approximiert werden.

Neuronale Netze – Warum überhaupt?

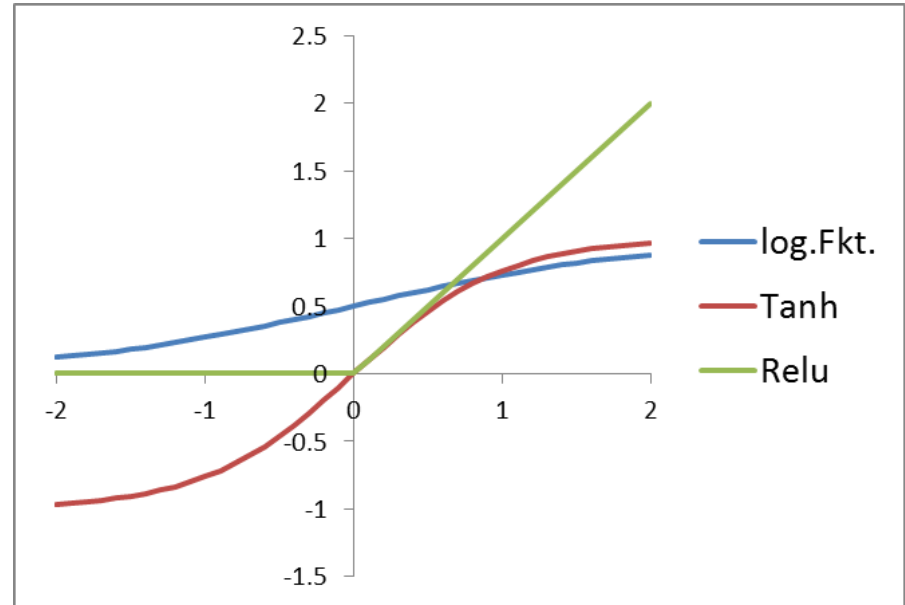
3

Typische Wahlen von $g(x)$

$$g(x) = \frac{1}{1 + e^{-x}} \quad \text{Logistische Funktion}$$

$$g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \text{Tangens hyperbolicus}$$

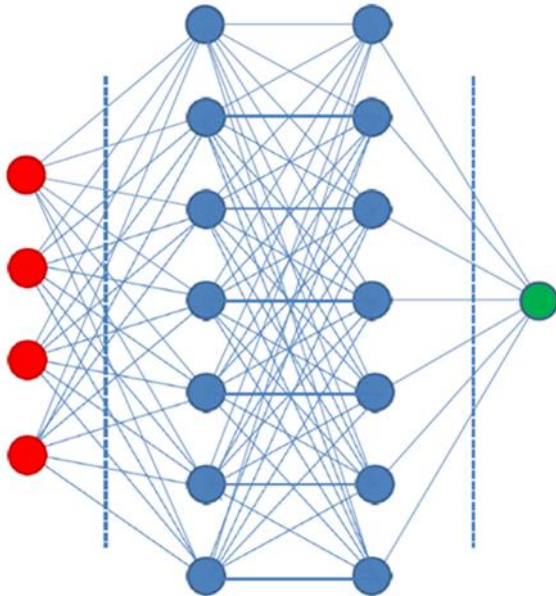
$$g(x) = \max\{x, 0\} \quad \text{Positivteil („RELU“)}$$



Warum gerade diese Art von Aktivierungsfunktionen?

Neuronale Netze – Vielfalt ohne Ende...

- **Feedforward-Netze** als „Regressions- und Klassifikationsersatz“



Für gegebene Dimensionen N_0 des Input und N_L des Output, sowie N_j des j . Hidden-Layer heißt

$$f(x, \theta) = w_L \circ f_{L-1} \circ \dots \circ f_1(x)$$

ein **Feedforward-Netz**. Dabei sind

$$f_j = g \circ w_j, \quad j = 1, \dots, L-1,$$

$$w_j(z) = W^{(j)}z + b^{(j)}, \quad j = 1, \dots, L \text{ mit } W^{(j)} \in \mathbb{R}^{N_j, N_{j-1}},$$

$$\theta = (W^{(1)}, \dots, W^{(L)}, b^{(1)}, \dots, b^{(L)})$$

Neuronale Netze – Vielfalt ohne Ende...

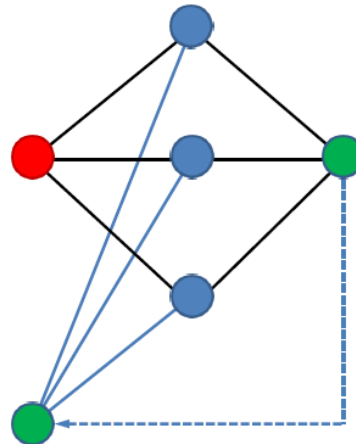
- **Rekurrente Netze** als „Zeitreihenalternative“

Idee:

Imitiere eine Zeitreihe h_t , in dem neben dem aktuellen Input x_t auch der vorangegangene Output h_{t-1} (oder Hidden-Layer des vorangehenden Zeitschritts) als Input für das neuronale Netz zur Erzeugung von h_t verwendet wird.

Ein besonders einfaches Beispiel:

$$h_t = f(x_t, h_{t-1})$$



Neuronale Netze – Vielfalt ohne Ende...

- **Konkurrierende Netze (GAN)** als „Datensimulator“

Generator-Netz:

Erzeuge aus zufälligen Daten $Z(\omega)$ mit Hilfe eines neuronalen Netzes $g_\theta(\cdot)$ eine Zufallsvariable $g_\theta(Z(\omega))$.

Diskriminator-Netz:

Ordne mit Hilfe des Netzes $d_\eta(\cdot)$ der vom Generator-Netz gelieferten Zufallsvariablen $g_\theta(Z(\omega))$ Werte in $[0,1]$ zu.

Idee: Generator will θ so trainieren, dass Diskriminator nicht entscheiden kann, ob es sich um vom Generator erzeugte Daten oder um „echte“ Daten handelt.

Diskriminator will η so trainieren, dass er möglichst richtig über die gelieferten Daten entscheidet.

Ziel: Erzeugung möglichst realistischer Daten durch den Generator.

Neuronale Netze – Vielfalt ohne Ende...

Viele weitere Beispiele:

- Ausgedünnte Netze (**Convolutional Network**), bei denen nicht alle Neuronen miteinander verbunden sind (speziell für Anwendungen in der Bilderkennung) => Effizienz durch Parameterreduktion
- Speziell konstruierte Einzelbestandteile, um z.B. Langzeitabhängigkeiten bei rekurrenten Netzen modellieren zu können (**LSTM** = Long short term memory)
- ...

Machine Learning und Co: Neue Namen braucht das Land (?)

Neu-Deutsch

- Data Science
- Machine Learning
- Feature Extraction
- Neuronale Netze
- Backpropagation
- Lernen
- Lernrate
- ...

Klassische Entsprechung (u.a.)

Statistik

(Algorithmische) Statistik

Hauptkomponentenanalyse

Nichtlineare Regression

Gradientenverfahren

Parameterkalibrierung

Schrittweite im Gradientenverfahren

Neuronale Netze: Warum gerade jetzt?

Hauptgründe:

- Enorme Rechnerkapazitäten (verfügbar lokal oder in der Cloud)
- Datengläubigkeit (durch Google, Amazon, etc. befeuert)
- Verfügbarkeit von vorinstallierter Software (Python, R, ...)
- Verstärkte Forschungsaktivitäten in Neuronalen Netzen mit spektakulären Anwendungen (Bildverarbeitung, Handschrifterkennung, ...)
- Programmierkomfort

Aber: Sind neuronale Netze wirklich so gut?

Neuronale Netze in der Anwendung - 1

- **Können neuronale Netze Optionsbewertung?**

Geht so ...

Einfache Aufgabenstellung:

- Fälligkeit $T = 1$, Volatilität $\sigma = 0,2$ und Zins $r = 0.05$ gegeben (also fest!)
- Trainingsdatensatz: gegebene Black-Scholes-Optionspreise für relative Ausübungspreise $K/S(0)$ in $\{0.4, 0.41, \dots, 1.99, 2\}$

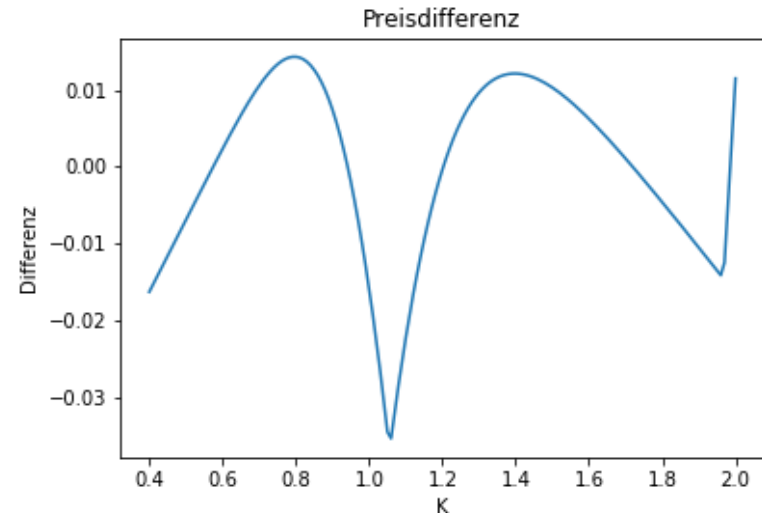
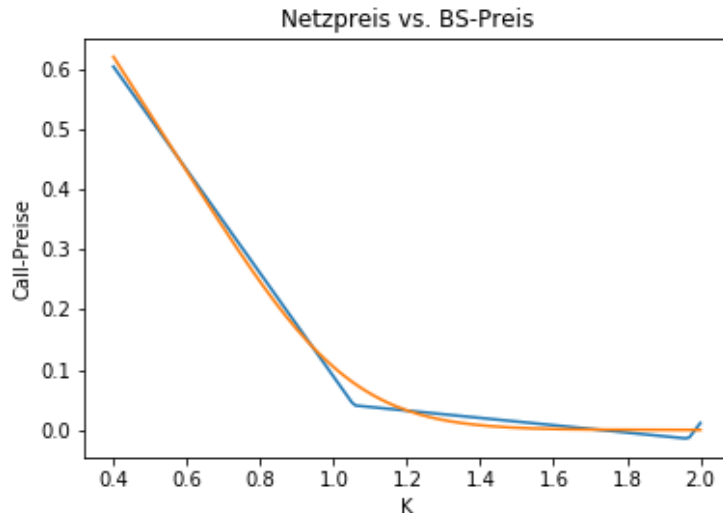
Ziel:

Untersuche die Anpassung an die Preise durch verschiedene neuronale Netze mit der RELU-Aktivierungsfunktion, d.h. $g(x) = \max\{x, 0\}$

Neuronale Netze in der Anwendung - 1

- Können neuronale Netze Optionsbewertung?**

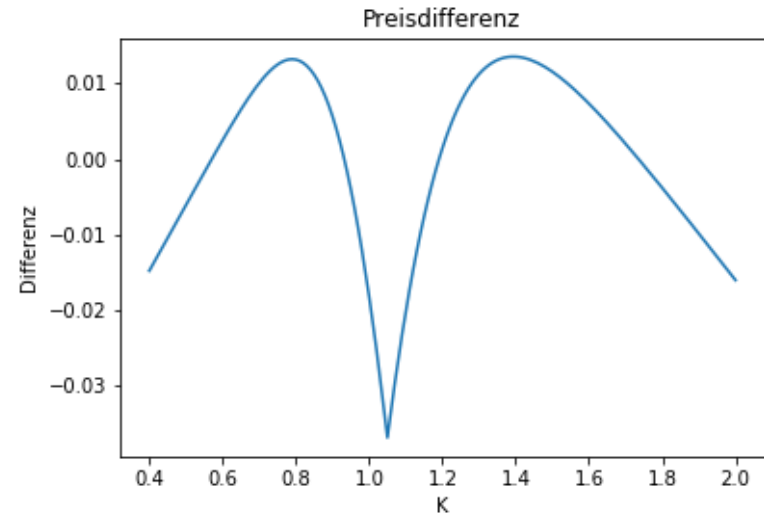
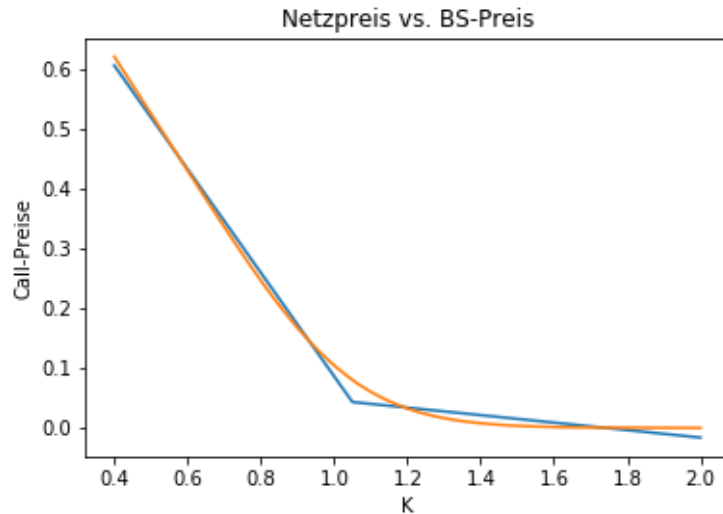
1 Hidden-Layer, 3 Neuronen



Neuronale Netze in der Anwendung - 1

- Können neuronale Netze Optionsbewertung?**

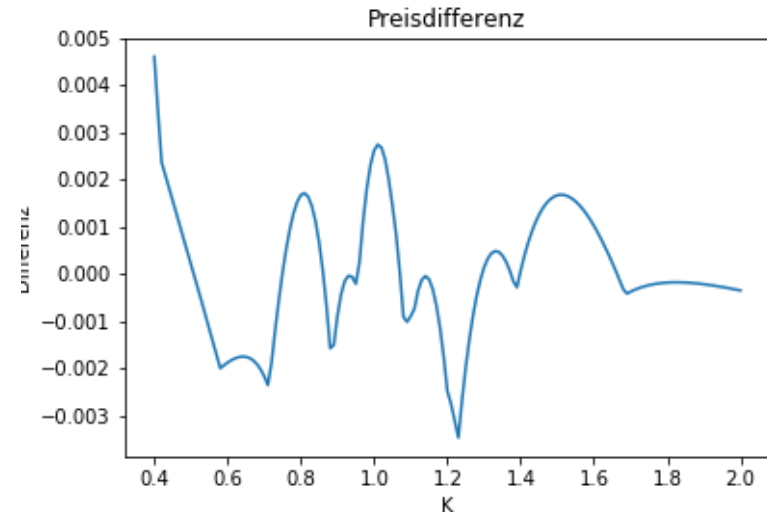
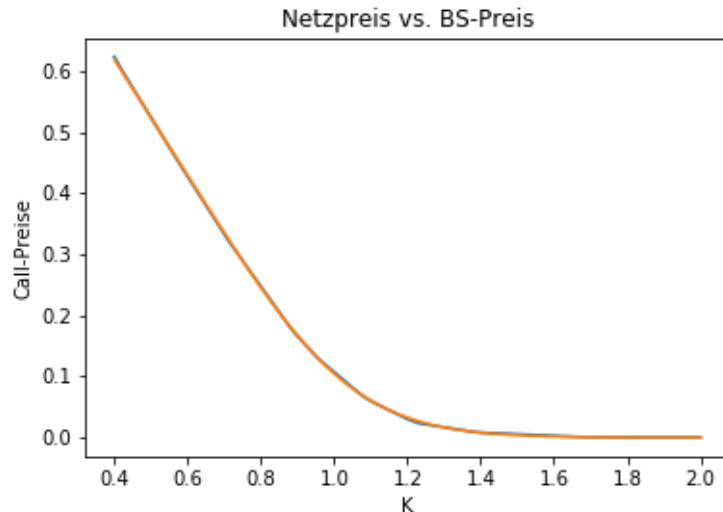
1 Hidden-Layer, 10 Neuronen



Neuronale Netze in der Anwendung - 1

- Können neuronale Netze Optionsbewertung?**

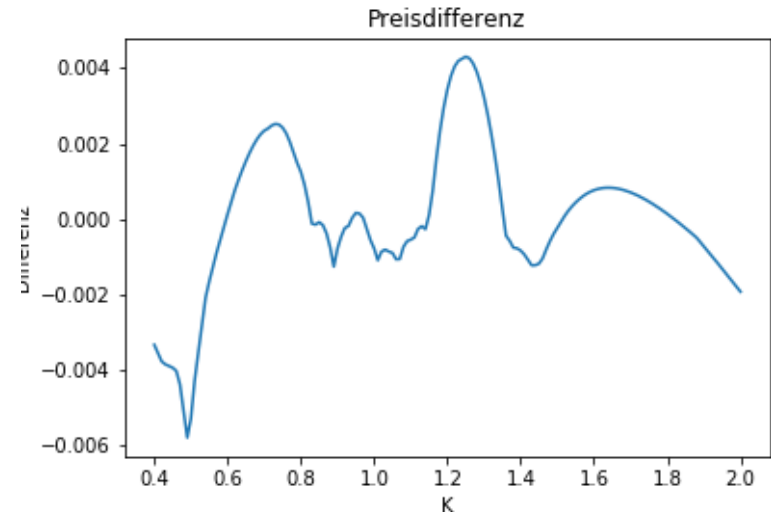
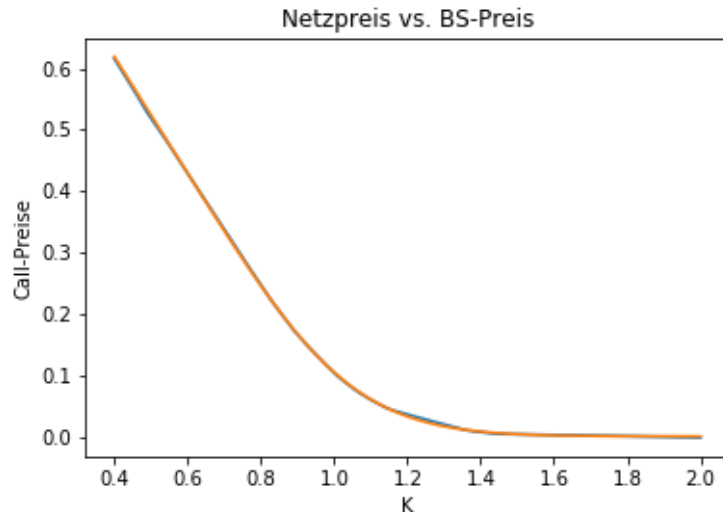
1 Hidden-Layer, 100 Neuronen



Neuronale Netze in der Anwendung - 1

- Können neuronale Netze Optionsbewertung?**

3 Hidden-Layer, (100,100,100) Neuronen



Neuronale Netze in der Anwendung - 2

- **Können neuronale Netze klassifizieren?**

In der Regel sehr gut...

Einfache Aufgabenstellung:

- Jedem Paar (x_1, x_2) , $-1 \leq x_i \leq 1$, wird der Wert 1 zugewiesen, wenn beide Komponenten nicht-negativ sind, 0 sonst
- Trainingsmenge: 1000 Paare

Ziel:

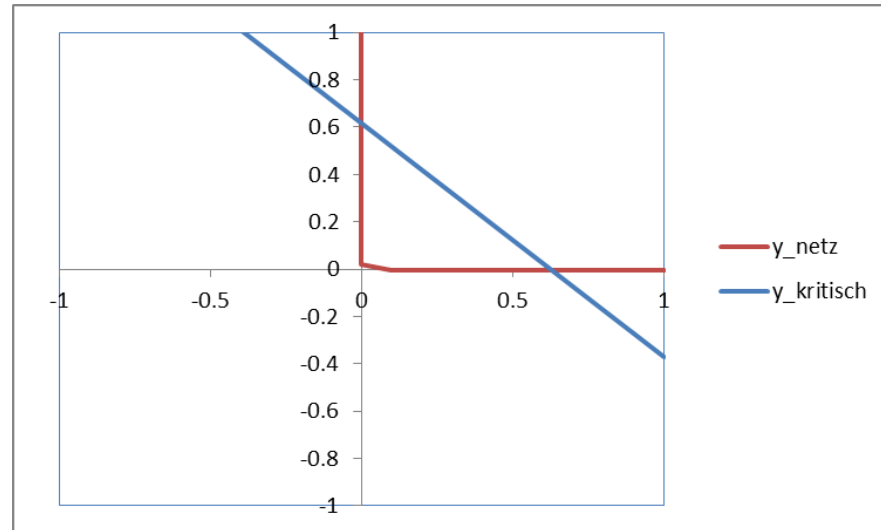
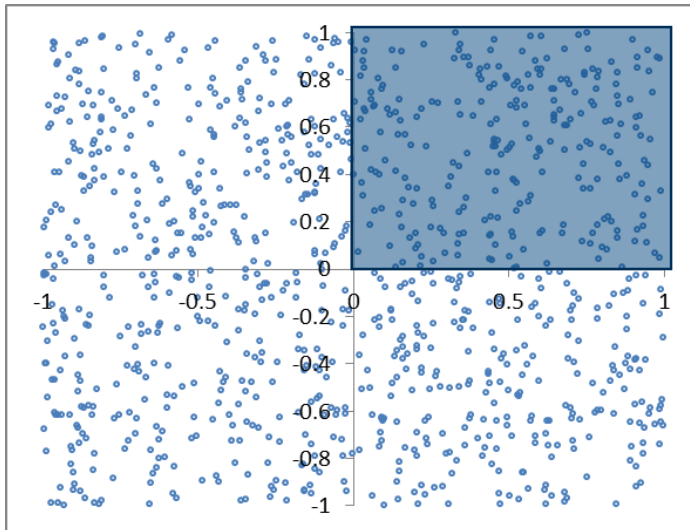
Finde die richtige Klassifikationsregel mit der logistischen Funktion als Aktivierungsfunktion, d.h. $g(x) = 1/(1+\exp(-x))$

Neuronale Netze in der Anwendung - 2

- Können neuronale Netze klassifizieren?**

Neur. Netz mit 1 Hidden-Layer, 2 Neuronen, Vergleich mit logist. Regression

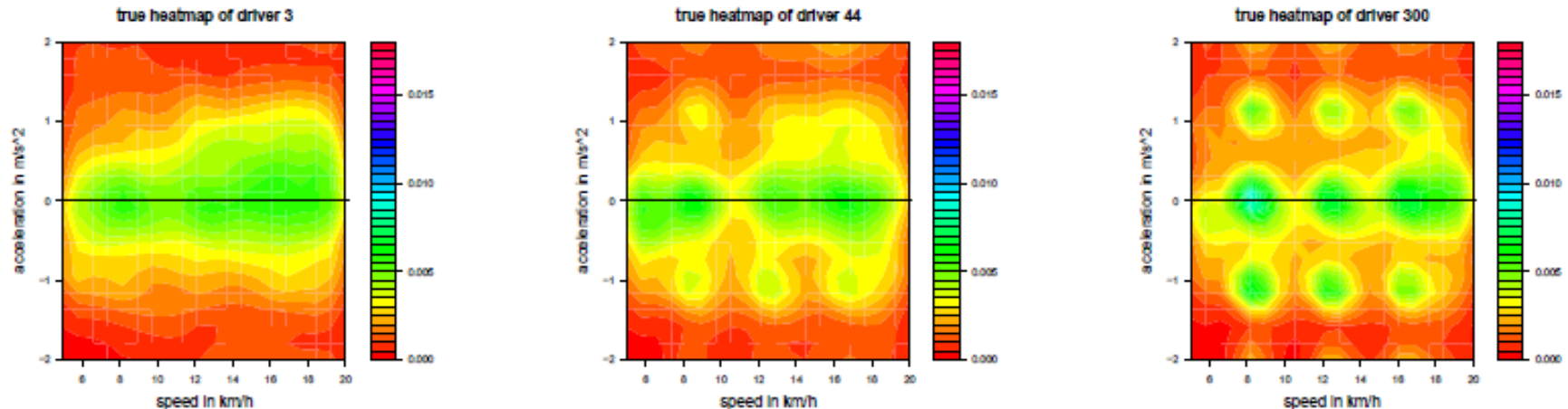
Rot = Trennlinie Neur. Netz, Blau = Trennlinie Logistische Regression



Neuronale Netze – Datenkompression bei Telematik

Guo, Wüthrich (2018) Feature extraction from telematics car driving heat maps, EAJ8(2):383-406.

- Abbildung von Fahrerdaten auf **Heat-Maps** („Wie oft wurde mit welcher Geschwindigkeit und welcher Beschleunigung gefahren?“)
- => **Hochdimensionale Objekte!**



Neuronale Netze – Datenkompression bei Telematik – 2

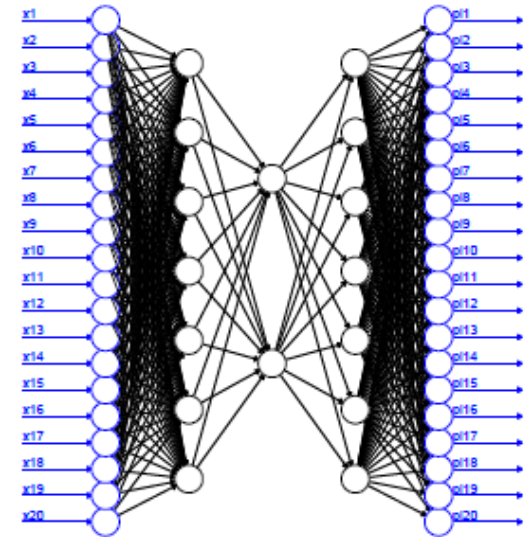
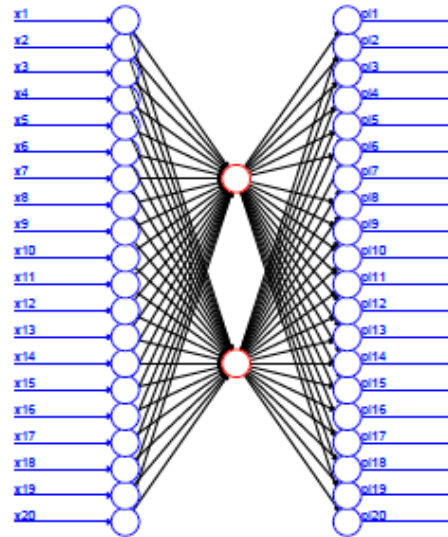
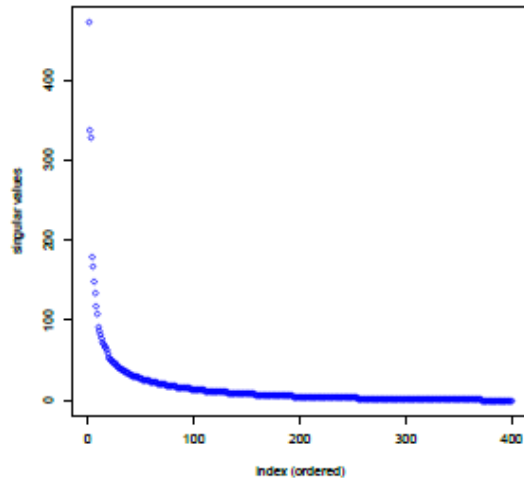
- Diskretisiere beide Achsen in 20 Teile => 400 Werte („Variablen“) pro Fahrer

Datenreduktion über

Hauptkomponentenanalyse, Autoencoder 1,

Autoencoder 2

singular values of SVD



Neuronale Netze – Datenkompression bei Telematik – 3

„Ergebnis“ mit Dimension 2:

- Die Hauptkomponentenanalyse mit den beiden ersten Hauptkomponenten schneiden ähnlich gut ab wie das neuronale Netz (der Autoencoder) mit einem Hidden-Layer (das **flache Netz**)
- Das neuronale Netz mit drei Hidden-Layer (das **tiefe Netz**) schneidet besser ab als die beiden anderen Methoden

Neuronale Netze – Datenkompression bei Telematik – 4

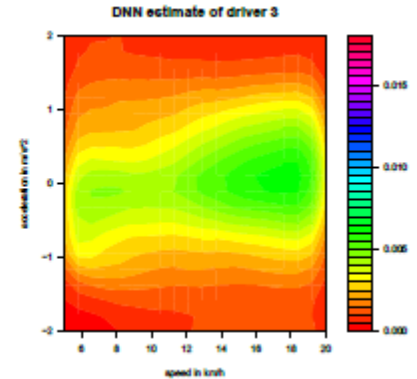
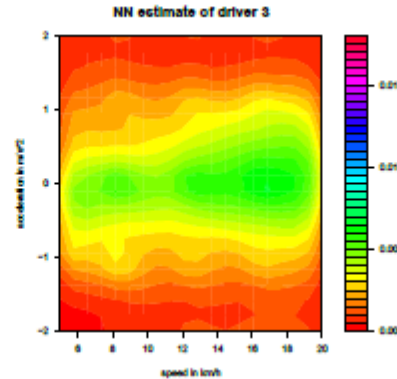
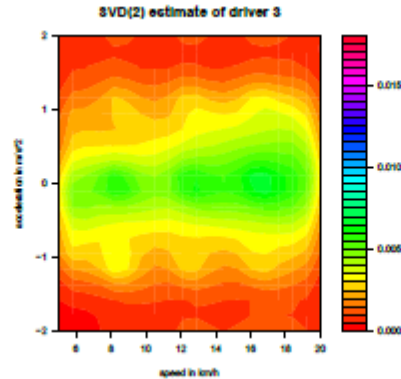
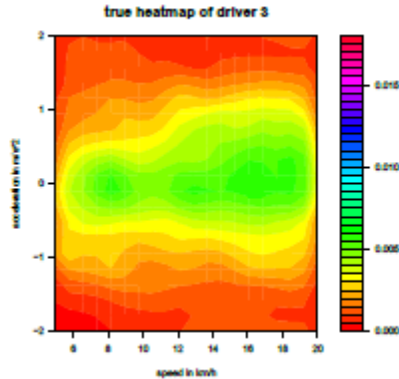
Bilder für Fahrer 3

Original

Hauptkomp.

Flaches Netz

Tiefes Netz



Neuronale Netze – Datenkompression bei Telematik – 4

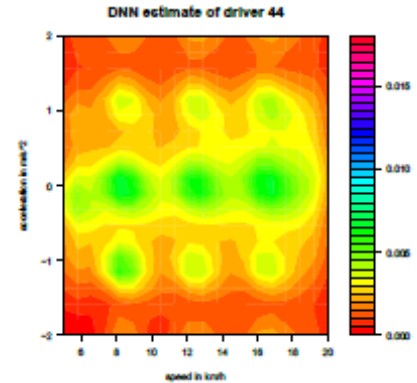
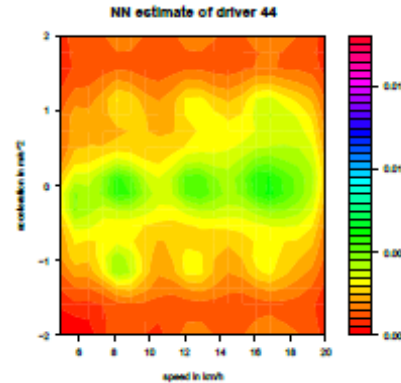
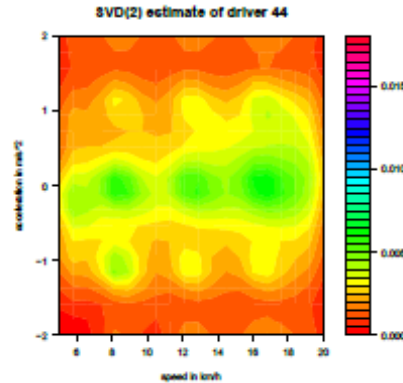
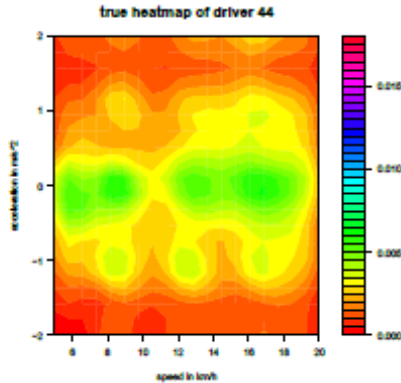
Bilder für Fahrer 44

Original

Hauptkomp.

Flaches Netz

Tiefes Netz



Neuronale Netze – Datenkompression bei Telematik – 4

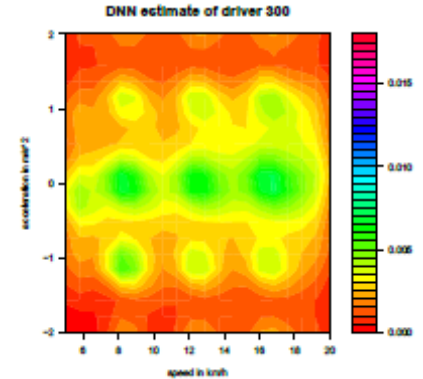
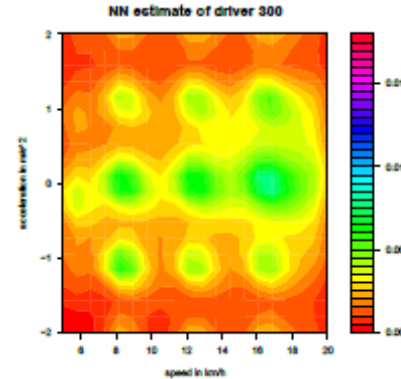
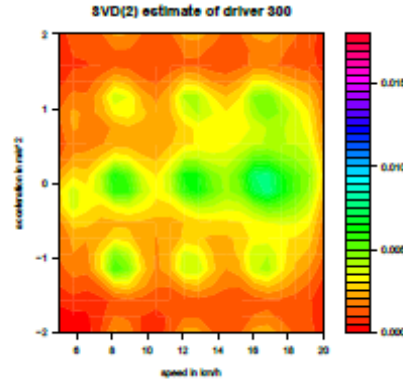
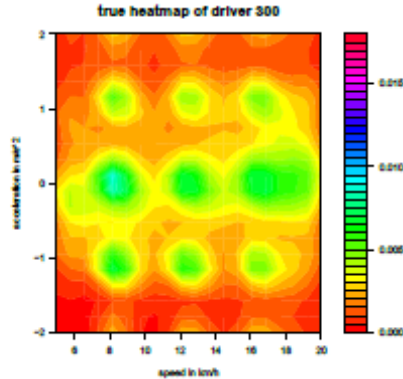
Bilder für Fahrer 300

Original

Hauptkomp.

Flaches Netz

Tiefes Netz



Neuronale Netze – Probleme 1

Mathematische/algorithmische Probleme

- Hauptproblem **Kalibrierung (Parameterbestimmung)**
 - Global optimale Parameter aufgrund der Nichtlinearität der Aktivierungsfunktion $g(\cdot)$ idR nicht bestimmbar
 - Oft nur lokales Minimum erreicht
 - Form der Aktivierungsfunktion führt zu verschwindenden Gradienten ($\Rightarrow$ sehr langsame Konvergenz) oder zu explodierenden Gradienten (evtl. gar keine Konvergenz)
 - Schrittweitenwahl bei gradientenbasierten Verfahren
 - Oft problemabhängige Sonderkonstruktionen nötig (eigentlich eher das Gegenstück von KI ...)

Neuronale Netze – Probleme 2

Mathematische/algorithmische Probleme

- Topologie des Netzes
 - Wahlen der Form des Netzes nicht generisch geregelt
 - Große Flexibilität birgt auch Chancen
- Wahl der Aktivierungsfunktion
 - Ähnliche Problematik wie bei Regression

Neuronale Netze – Probleme 3

Probleme der Interpretation

- Black-Box-Charakter
 - Nutzung von vorkonfigurierter Software ist praktisch, erlaubt aber nicht alle Formen der Analyse und Optimierung
 - Sensitivitätsanalyse verschachtelter nicht-linearer Regression ist schwierig
 - Interpretation erhaltener Ergebnisse („Warum ist das so?“) ist durch den verschachtelten nicht-linearen Charakter oft mehr Vermutung als fundierte Erklärung

Neuronale Netze - Gesamtwürdigung

Abschließenden Bemerkungen

- Neuronale Netze werden z.T. als die *Lösung aller Probleme* gefeiert, z.T. als *alter Wein in neuen Schläuchen* bezeichnet
- Neutrale Herangehensweise ist empfehlenswert
- Kritischer Ansatz bei der Anwendung ist absolut wichtig
- Zwingend notwendig:
 - Domänenwissen (=> es braucht den Aktuar!)
 - Vergleich zu einfachen, bewährten Methoden
 - Out-of-Sample-Überprüfung (Ausführliche Testphase)

Vielfach sind dem Aktuar die linearen Varianten bereits bekannt, er sollte sich den nicht-linearen Erweiterungen nicht verschließen

Neuronale Netze – Erst mal Schluss ...

Vielen Dank für die
Aufmerksamkeit !

Referenzen (kleine Auswahl ...)

- J. Braun, M. Griebel (2009) On a constructive proof of Kolmogorov's superposition theorem, *Constructive Approximation*, 30: 653.
- G. Gao, M.V. Wüthrich (2018) Feature extraction from telematics car driving heat maps, *EAI*, 8(2):383-406.
- D. Hainault (2016) A neural-network analyzer for mortality forecast, *ASTIN Bull* 48(2), 481-508.
- T. Hastie, R. Tibshirani, J. Friedman (2009) *The elements of statistical learning – Data mining, inference, and prediction*, 2nd edition, Springer.
- A. N. Kolmogorov (1957) On the representation of continuous functions of many variables by superpositions of continuous functions of one variable and addition. *Doklady Akademii Nauk USSR*, 14(5):953–956.
- K. Hornik, M. Stinchcombe, H. White (1989) Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359-366.
- M. Wiese (2019) *Deep generative modeling of financial time series*, Masterarbeit, TU Kaiserslautern.

Referenzen (kleine Auswahl ...)

R. Richman, M.V. Wüthrich (2018) A neural network extension of the Lee-Carter model to Multiple Populations. <http://dx.doi.org/10.2139/ssrn.3270877>

M.V. Wüthrich (2018) Neural networks applied to chain ladder reserving, EAJ 8(2):407-436.

Beste Referenz für Versicherungsanwendungen:

Homepage von Mario Wüthrich (ETH Zürich): <https://people.math.ethz.ch/~wueth/>