Neural Random Forest

G. Biau, E. Scornet, J. Welbl

Sankhyā A: The Indian Journal of Statistics
21. Juni 2018

Problemstellung und Ziel

- Darstellung des Random Forest von Breiman (2001)[2] als eine Menge von Neuronalen Netzwerken.
- Dadurch sollen die Nachteile beider Methoden verringert und die Vorteile hervorgehoben werden.

Neuronale Netzwerke (NN):
Viele Parameter müssen angepasst werden
⇒ Ermöglicht komplexe Modellierungen
⇒ Hohes Overfitting-Risiko

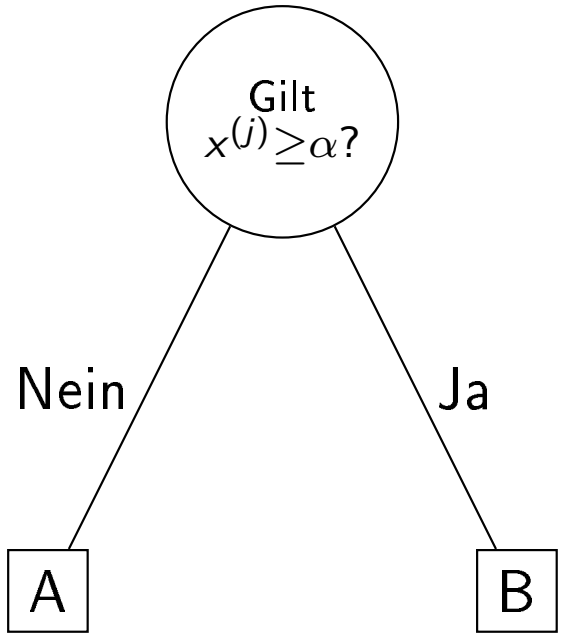
Random Forest (RF):
Die Art zu Unterteilen wirkt sich auf einige Datenmenge gut aus, jedoch ist sie für andere besonders ungeeignet.

Bisherige Ansätze

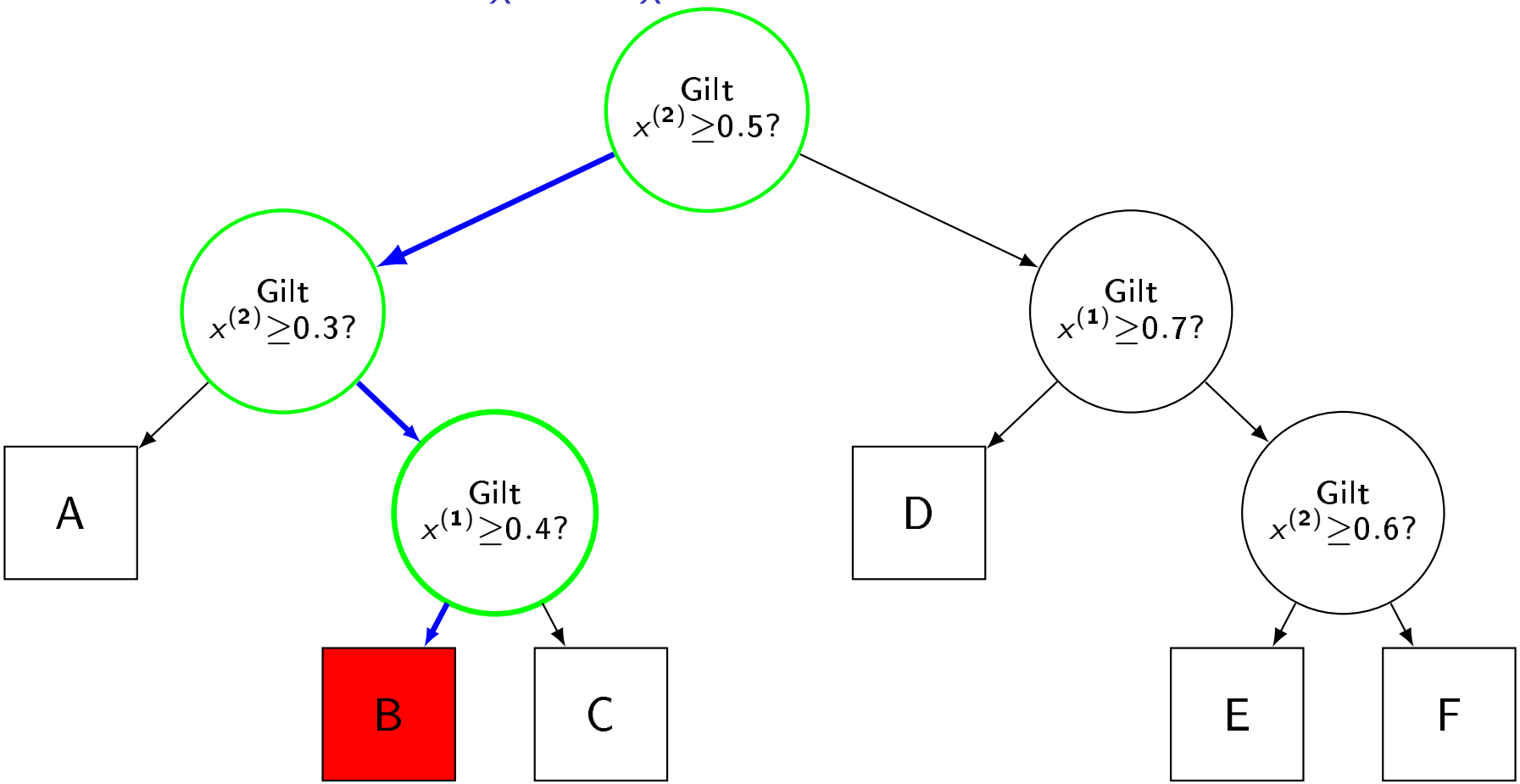
- Mehrere Ansätze versuchten bereits eine Verbindung zwischen Entscheidungsbäumen und Neuronales Netzwerken herzustellen
  - Sethi (1990[9]/1991[8])
  - Brent (1991)[3]
  - Devroye et al. (1996, Kapitel 30)[5]
  - Kontschieder et al. (2015)[7]
  - Ioammou et al. (2016)[6]
- Nicht bekannt, dass eine theoretische Studie über die Verbindung zwischen RF und NN existiert

Tree

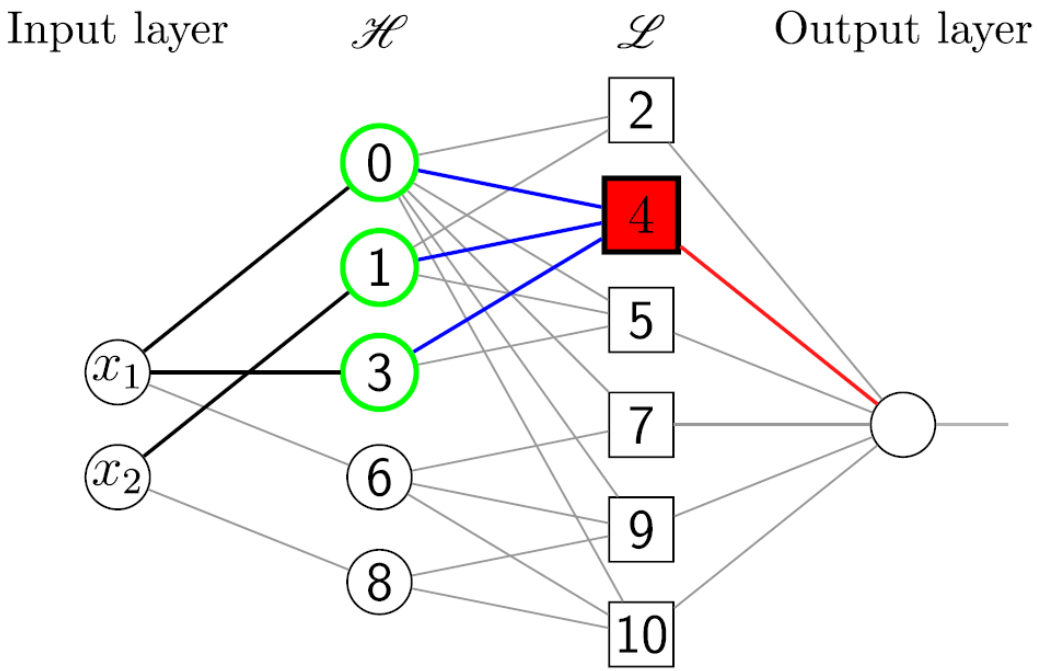
- Input der Form $x \in [0,1]^d$
- $\alpha \in [0,1], j \in \{1, \ldots, d\}$



Tree (Beispiel: $x = (\underbrace{0.3}_{x^{(1)}}, \underbrace{0.3}_{x^{(2)}})$)



Neuronales Netzwerk[1]

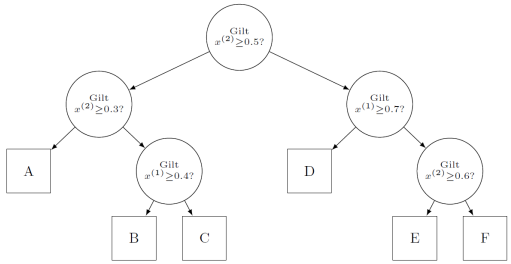


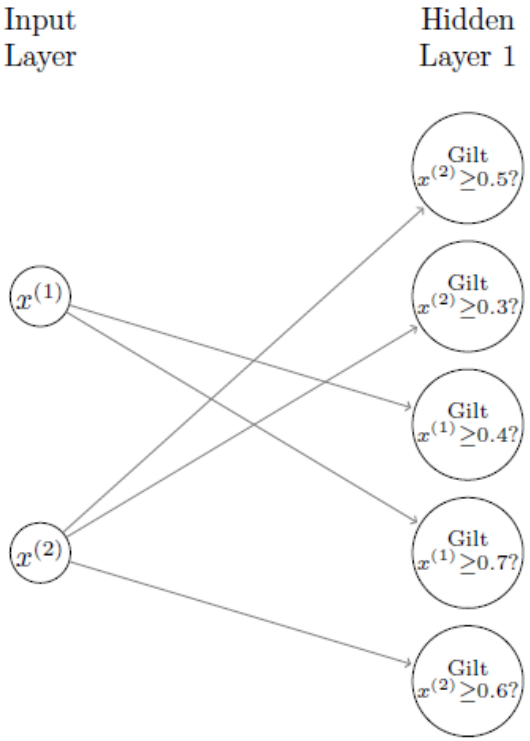
Aktivierungsfunktion: $\tau(u) = 2\mathbb{1}_{u \geq 0} - 1$

First Hidden Layer (HL1)

- Enthält $K - 1$ Neuronen, wobei K die Anzahl der Blätter im Baum ist
- Jedes Neuron repräsentiert einen inneren Knoten und damit seine „Frage“
- Ein Neuron ist nur mit dem zugehörigen Eingabewert verbunden
- Kanten ungewichtet
- Ausgabe 1, wenn $x^{j_k} \geq \alpha_{j_k}$ und -1 wenn nicht

First Hidden Layer: Beispiel

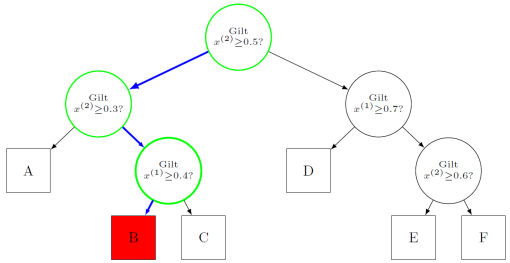


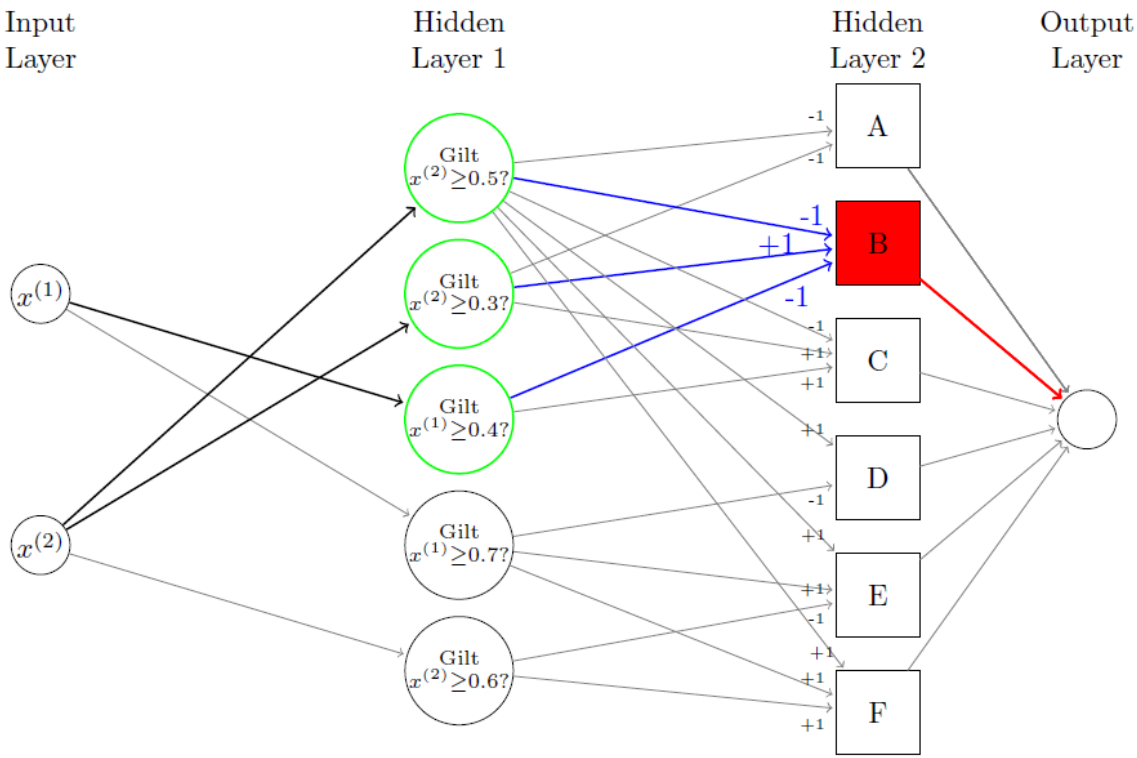


Second Hidden Layer (HL2)

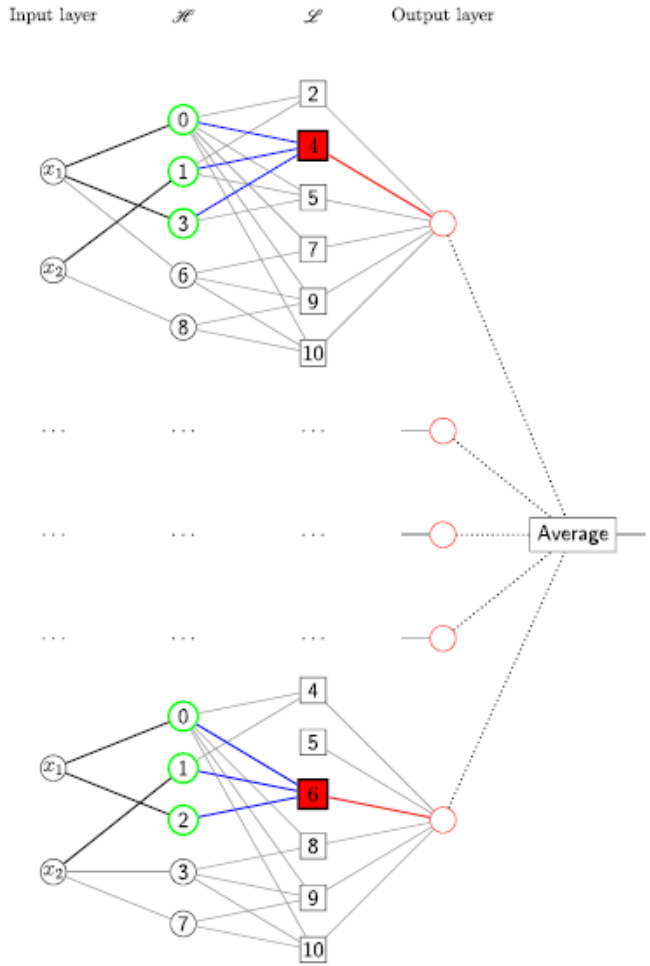
- Enthält K Neuronen
- Jedes Neuron repräsentiert ein Blatt
- Verbunden mit Neuronen aus HL1, die Knoten auf dem Pfad zum Blatt repräsentieren
- Kanten haben den Wert 1 bzw. -1, wenn auf dem Pfad nach rechts bzw. links gegangen wird
- Ausgabe 1, wenn die Eingabe zu dem repräsentierten Blatt gehört und -1 wenn nicht

Tree → NN: Beispiel

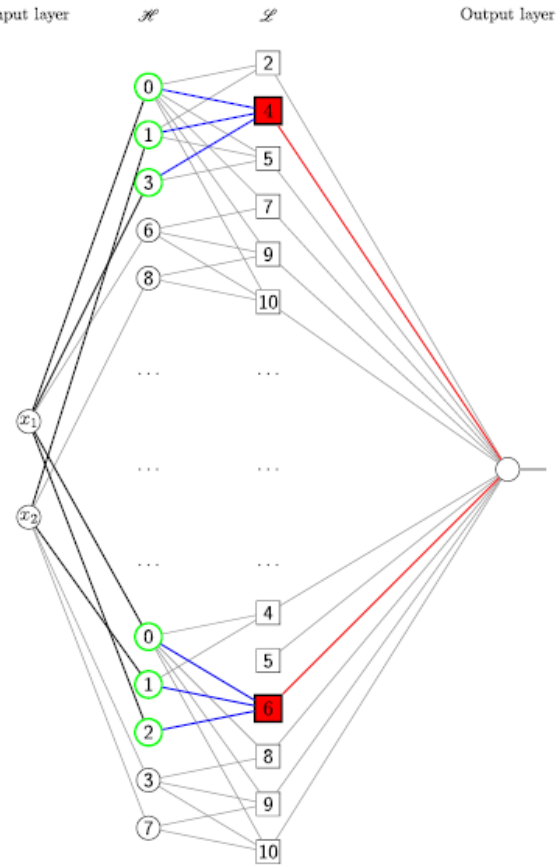




Trainingsarten[1]



Idependent Training



Joint Training

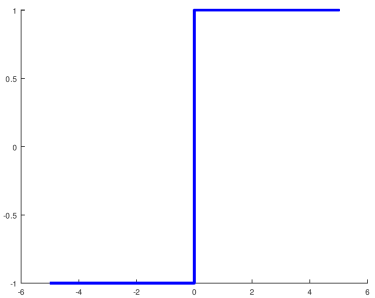
Backpropagation

- Fehler des Output-Neurons bestimmen
- Nutzen um Fehler des vorherigen Layers zu bestimmen
- Nutzen um Fehler des vorherigen Layers zu bestimmen

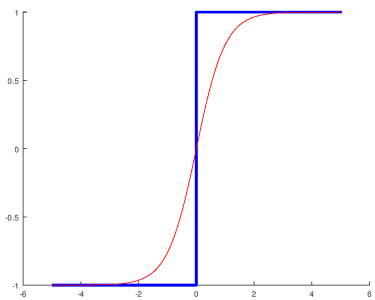
  ⋮
- Wiederholen bis man beim Input-Layer ankommt

Anmerkung zu Backpropagation

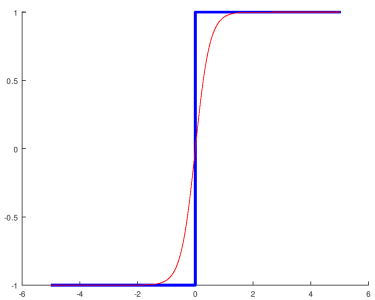
- Bisherige Aktivierungsfunktion: $\tau(u) = 2\mathbb{1}_{u \geq 0} - 1$
- Problem: Nicht differenzierbar
- Neue Funktion: $\sigma(u) = \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} = \frac{e^{2u} - 1}{e^{2u} + 1}$
- Genauer $\sigma(\gamma u)$



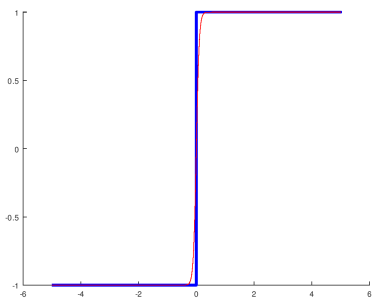
$\tau(u)$



$\gamma = 1$



$\gamma = 2$



$\gamma = 10$

Experimente

- Jedes Experiment wir 10mal durchgeführt
- Unterteilung der Daten: 50/25/25 (training/validation/test)
- Vergleich mit
  - Standard RF
  - Standard NNs (1-3 Layer)
  - BART[4]
- NRF wird mit beiden Methoden auf zwei Weisen getestet
  - sparsely connected
  - fully connected[1]

[1] Jedes Neuron ist mit allen aus dem vorherigen Layer verbunden

Daten[1]

- Nicht-numerischen Features werden entfernt
- Exemplare mit fehlenden Einträgen wurden zuvor entfernt
- Vor jeder Durchführung zufällig gemischt

Data set	Number of samples	Number of features
Auto MPG	398	7
Housing	506	13
Communities and Crime	1994	101
Forest Fires	517	10
Wisconsin	194	32
Concrete	1030	8
Protein	45730	9

Root-mean-squared-error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$$

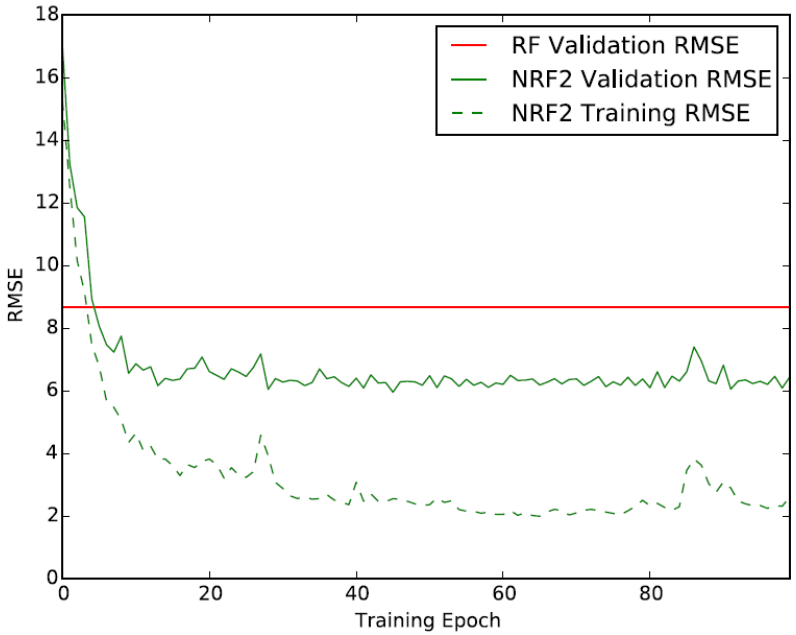
- T Anzahl der vorhergesagten Werte
- $\hat{y}_t$ Tatsächlicher Wert
- y_t Ausgegebener Wert

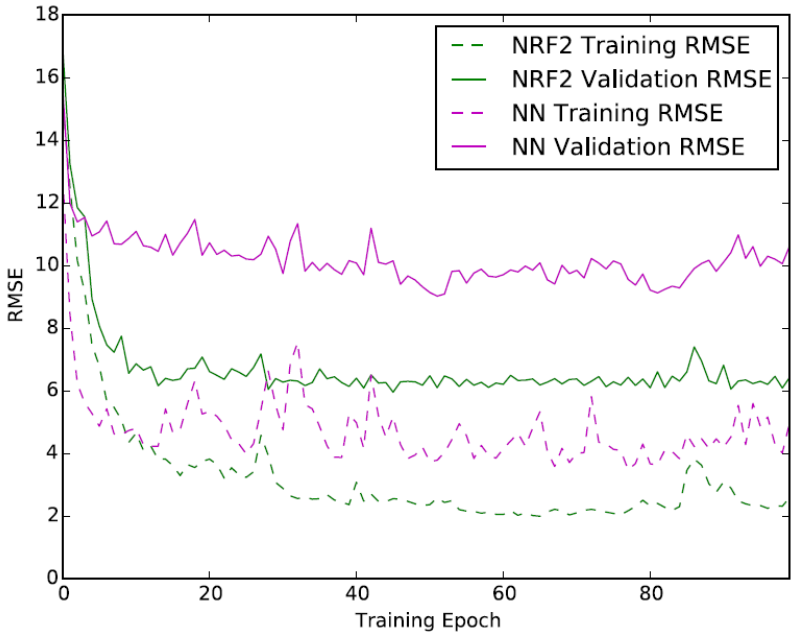
RMSE-Vergleich[1]

Data set	NN1	NN2	NN3	RF
Auto MPG	4.56 (0.83)	3.95 (0.39)	5.02 (0.72)	3.44 (0.38)
Housing	9.06 (0.85)	7.81 (0.71)	12.52 (1.08)	4.78 (0.88)
Crime	5.39 (0.42)	6.78 (0.47)	6.64 (0.31)	0.17 (0.01)
Forest Fires	96.7 (0.20)	*54.87 (34.33)*	97.9 (0.90)	95.47 (43.52)
Wisconsin	36.91 (0.88)	*34.71 (2.36)*	38.43 (2.88)	45.63 (3.53)
Concrete	10.18 (0.49)	10.21 (0.68)	11.52 (0.88)	8.39 (0.62)
Protein	6.12 (0.02)	6.11 (0.02)	6.12 (0.02)	5.06 (0.03)

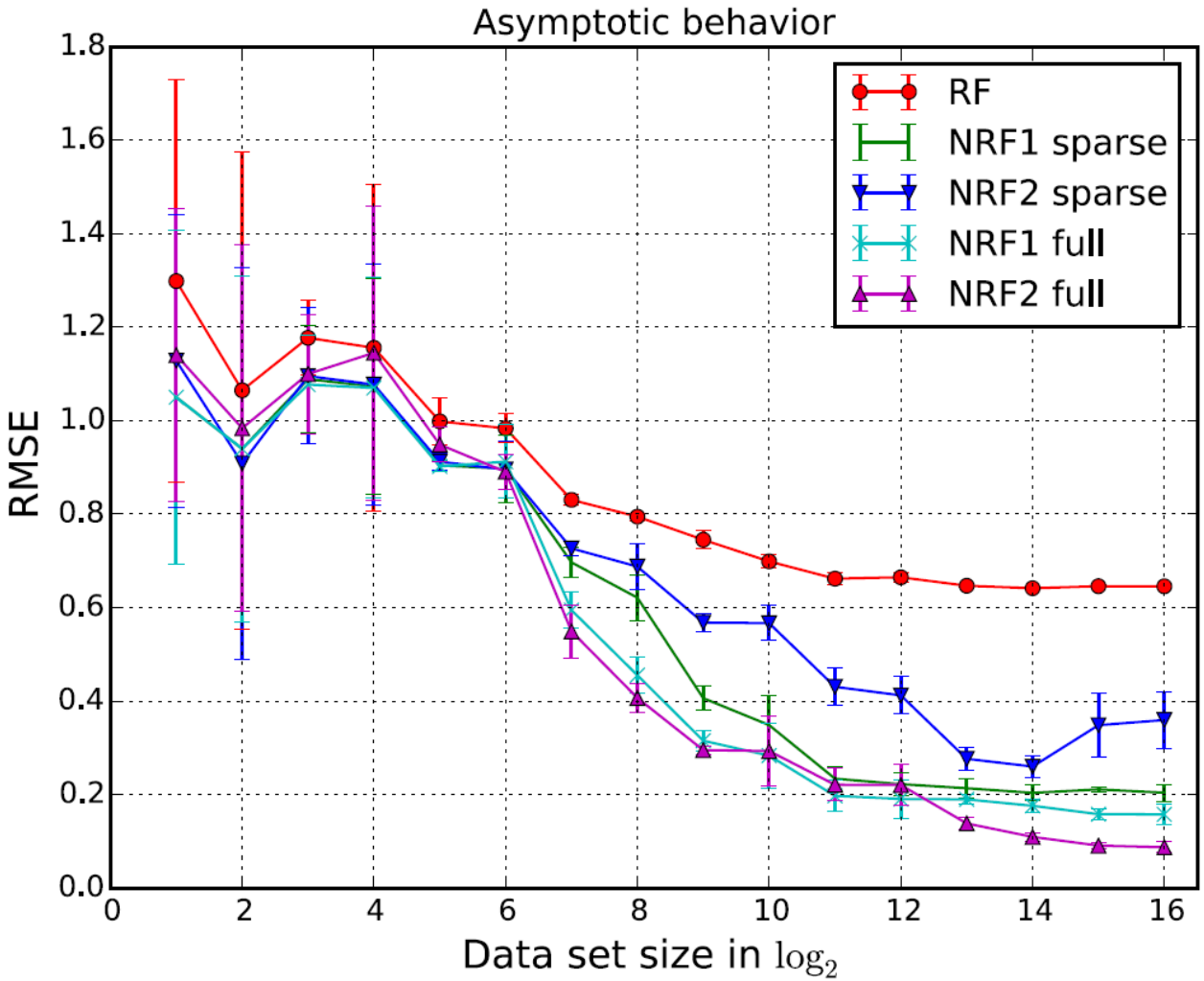
NRF1 full	NRF2 full	NRF1 sparse	NRF2 sparse	BART
3.20(0.39)	3.35 (0.46)	3.28 (0.41)	3.28 (0.42)	**2.90 (0.33)**
4.34 (0.85)	4.68 (0.88)	4.59 (0.91)	4.62 (0.88)	**3.78 (0.51)**
0.16 (0.01)	0.16 (0.01)	*0.16 (0.01)*	0.16 (0.01)	**0.14 (0.01)**
54.47 (34.64)	78.60 (28.17)	68.51 (35.56)	82.80 (32.07)	55.04 (16.40)
37.12 (2.89)	41.22 (3.05)	40.70 (2.51)	38.03 (3.95)	**33.50 (2.26)**
6.28 (0.40)	6.44 (0.37)	7.42 (0.56)	7.78 (0.56)	**5.20 (0.34)**
4.82 (0.04)	*4.77 (0.05)*	4.82 (0.04)	4.77 (0.05)	**4.57 (0.04)**

Entwicklung[1]





Einfluss der Datensatzgröße[1]



Fazit

- NRF ist auch auf größeren Datensätzen besser als der ursprüngliche RF
- NRF scheint eine bessere Toleranz gegenüber Overfitting zu haben als Standard-NNs
- Aufgrund des Erfolges von BART, sollten die dargestellten Methoden darauf getestet werden

Literatur I

[1] Biau, G. ; Scornet, E. ; Welbl, J. : Neural Random Forests. In: *Sankhya A* (2018), Jun. http://dx.doi.org/10.1007/s13171-018-0133-y. – DOI 10.1007/s13171–018–0133–y. – ISSN 0976–8378

[2] Breiman, L. : Random Forests. In: *Machine Learning* 45 (2001), Oct, Nr. 1, 5–32. http://dx.doi.org/10.1023/A:1010933404324. – DOI 10.1023/A:1010933404324. – ISSN 1573–0565

[3] Brent, R. P.: Fast training algorithms for multilayer neural nets. In: *IEEE Transactions on Neural Networks* 2 (1991), Nr. 3, S. 346–354

[4] Chipman, H. A. ; George, E. I. ; McCulloch, R. E. u. a.: BART: Bayesian additive regression trees. In: *The Annals of Applied Statistics* 4 (2010), Nr. 1, S. 266–298

[5] Devroye, L. ; Györfi, L. ; Lugosi, G. : *A probabilistic theory of pattern recognition*. Bd. 31. Springer Science & Business Media, 2013

Literatur II

[6] Ioannou, Y. ; Robertson, D. ; Zikic, D. ; Kontschieder, P. ; Shotton, J. ; Brown, M. ; Criminisi, A. : Decision forests, convolutional networks and the models in-between. In: *arXiv preprint arXiv:1603.01250* (2016)

[7] Kontschieder, P. ; Fiterau, M. ; Criminisi, A. ; Rota Bulo, S. : Deep neural decision forests. In: *Proceedings of the IEEE international conference on computer vision*, 2015, S. 1467–1475

[8] Sethi, I. K.: Decision tree performance enhancement using an artificial neural network implementation. In: *Machine Intelligence and Pattern Recognition* Bd. 11. Elsevier, 1991, S. 71–88

[9] Sethi, I. K.: Entropy nets: from decision trees to neural networks. In: *Proceedings of the IEEE* 78 (1990), Nr. 10, S. 1605–1613