kernDeepStackNet: An R package for fitting kernel deep stacking networks

**Speaker:**
Thomas Welchowski (M. Sc.)

**Institution:**
Institute of Medical Biometry, Informatics and Epidemiology
University Hospital Bonn
Table of contents

1 Introduction deep learning

2 Kernel deep stacking networks

3 Tuning of KDSNs

4 Sparse kernel deep stacking networks
Applications of deep learning in biology
Overview of deep learning

- Variations and extensions of artificial neural networks
- Works better with large amounts of observations
- Applications outside biology (pattern recognition with images, videos, speech data, natural language processing or mastering the ancient game of Go)
Deep learning: Advantages and disadvantages

Advantages
- Multiple non-linear transformations
- Higher level feature abstractions
- Almost no assumptions

Disadvantages
- Large number of parameters and tuning parameters
- No standard systematic tuning approach available
- Difficult interpretation of results
Kernel deep stacking networks (KDSNs): Overview

\[ X = \text{input data} \]

\[ K_j = \text{random Fourier transformation} \]

\[ p_j = \text{prediction} \]

1st level

2nd level

3rd level

Output

Kernel ridge regression
Comparison of KDSN with artificial neural networks

Similarities
- Infinite series expansion with kernel representation
- \( \approx \) Large hidden layer of neural network
- Universal function approximation with radial basis functions
- \( \approx \) Generalization ability of neural networks

Advantages of KDSN
- Convex closed form solution available
- Computationally faster estimation
Data: $y \in \mathbb{R}$; $X \in \mathbb{R}^{n \times d}$ is assumed to be standardized \hspace{1cm} (1)

Kernel matrix: $K(x_i, x_j) := \exp \left( -\|x_i - x_j\|_2^2 / (2\tau^2) \right)$ \hspace{1cm} (2)

$K = (K(x_i, x_j))_{i,j=1,...,n} \in \mathbb{R}^{n \times n}$ \hspace{1cm} (3)

Random Fourier transformation (RFT): $\tilde{K} := Z^\top Z$ \hspace{1cm} (4)

$Z := \left( \cos(W^\top X^\top), \sin(W^\top X^\top) \right)^\top \in \mathbb{R}^{2D \times n}$ \hspace{1cm} (5)

$W \in \mathbb{R}^{d \times D}; \quad W_{i,..} \sim \text{iid} \ N(0, I/\sigma^2)$ \hspace{1cm} (6)

Estimation: $Zy \approx (ZZ^\top) \cdot (Z\beta)$ \hspace{1cm} (7)

$\Rightarrow \tilde{\beta} = \left( (ZZ^\top) + \lambda I \right)^{-1} (Zy); \quad I \in \mathbb{R}^{2D \times 2D}, \; \lambda > 0$ \hspace{1cm} (8)
Tuning parameters of KDSN

- $D$: Dimension of random Fourier transformation
- $\sigma^2$: Precision parameter of the random weights
- $\lambda$: Regularization parameter of ridge regression
- $L$: Number of KDSN levels

Traditional approach

- Cross validation based on grid search
- Computationally infeasible with many tuning parameters
Model based Optimization (MBO): General approach

1. Define loss function
2. Construct initial design of tuning parameters given level
3. Evaluate design
4. Estimate metamodel
5. Optimize meta loss to choose new point
6. Evaluate new point and add to the sample
7. Choose meta loss for next point evaluation
8. Iterate until convergence criterion is satisfied

1. Hill climbing to select level
2. MBO multiple times with different levels
MBO with application to KDSN

Latin hypercube maximin sampling

![Latin Hypercube Sampling Diagram](image)

Universal kriging with noisy observations

- Estimates conditional expectation and variances based on Gaussian process
- All parameters can be inferred by maximum likelihood

Expected improvement criterion

- Balances predicted improvement compared to best solution and variability
- Closed form available in combination with kriging model
Application: Readmission risk of diabetes patients

**Goals:**
- Better allocation of medical care interventions
- Quality metric for hospital performance and cost management

- 71,518 observed patients
- Data collected from 130 hospitals in the US between 1999 and 2008

**Response:** Binary variable, which indicates if a patient has been readmitted to hospital in less than 30 days after the first visit

**Covariates:**
- Demographic factors (e.g. age and gender)
- Clinical variables (e.g. number of diagnoses)
- Medical features (e.g. information on diabetic medication)
- Hospital characteristics (e.g. number of emergency visits)
Tuning of KDSN vs random choice of tuning parameters

![Box plot of AUC values across different levels]

- **AUC** values range from 0.48 to 0.62.
- **Level** values range from 1 to 10.
- The box plots show the distribution of AUC values at each level, with the median values indicated by horizontal lines.

**Graph Analysis**

- The median AUC values generally increase with level, indicating better performance.
- The interquartile range (IQR) also increases with level, suggesting greater variability in performance.

**Conclusion**

- Tuning of KDSNs parameters shows consistently better AUC values compared to random choice.
- Further investigation into the specific tuning parameters could optimize performance.

---

**Note:**

- This graph is a visual representation of the performance of KDSN models tuned versus random choice of tuning parameters.
- The performance metric chosen is AUC, which is commonly used for evaluating binary classification models.
Performance: KDSN vs other machine learning algorithms

- KDSN: AUC=0.6095
- H2O: AUC=0.5741
- RF: AUC=0.5338
- darch: AUC=0.5941
Current research: Sparse KDSN

Issues

- Non-influential covariates degrades performance
- Random Fourier transformations may contain irrelevant noise
- Errors made earlier in the network propagate to higher levels

Key extensions of KDSN

- Variable selection based on nonlinear association measure
  - Preselection with genetic algorithm
  - Internal variable selection in MBO tuning
- Dropout in random Fourier transformation
SKDSN vs other machine learning approaches

Data sets with binary responses

- **p53**: Mutated p53 transcriptional activity from in vivo experiments (31159 observations and 5408 continuous covariates)
- **Arcene**: Cancer patterns in mass-spectrometric data (100 observations and $10^4$ continuous variables)
- **Dorothea**: Binding to thrombin with chemical compounds (800 observations and $10^5$ binary covariates)

Comparison algorithms

- Random forest
- Gradient boosting with linear and penalized B-Spline base learners
p53 performance results

Performance
- Boost: AUC=0.95 [0.90 – 1]
- RF: AUC=0.94 [0.90 – 0.99]
- SKDSN: AUC=0.93 [0.89 – 0.98]
Arcene performance results

Performance
- RF: AUC=0.88 [0.82 – 0.94]
- SKDSN: AUC=0.84 [0.77 – 0.92]
- Boost: AUC=0.77 [0.68 – 0.86]
Dorothea performance results

Performance
- RF: AUC=0.91 [0.86 − 0.96]
- SKDSN: AUC=0.90 [0.82 − 0.97]
- Boost: AUC=0.86 [0.77 − 0.95]
Summary

- KDSN estimation is faster than deep artificial neural networks
- MBO is more efficient than traditional grid search tuning
- Sparse regularization improves performance in higher dimensions
- Implemented in R-package *kernDeepStackNet*

**Reference:** Thomas Welchowski and Matthias Schmid, 2016, A framework for parameter estimation and model selection in kernel deep stacking networks, Artificial Intelligence in Medicine, Volume 70, Pages 31-40