Efficient and Robust Automated Machine Learning

Robert Feldhans

12. Juli 2018

Seminar Musterklassifikation

Inhalt

1. Motivation
2. Automated Machine Learning in a Nutshell
3. Meta Learning
4. Ensembles
5. Anwendung und Ergebnisse
6. Fragerunde

Motivation

Interlude: Hyperparameter I

Was sind Hyperparameter?

- Werden *vor* dem Lernen definiert
- Sind in der Regel Zahlen oder Funktionen

Interlude: Hyperparameter I

Was sind Hyperparameter?

- Werden *vor* dem Lernen definiert
- Sind in der Regel Zahlen oder Funktionen

Allgemein

Alles was in irgendeiner Art austauschbar ist in einem speziellen ML-Verfahren und während des Trainings konstant bleibt

Interlude: Hyperparameter II

Beispiele für Hyperparameter

- Lernrate
- Gewichte jeglicher Form
- Anzahl der Cluster in k-means clustering
- Aktivierungsfunktionen
- Anzahl der Hidden Layers in einem Netz
- Breite der Layers in einem Netz

Motivation

- Ein gutes neuronales Netz zu trainieren ist schwer, braucht viel Arbeitszeit und Erfahrung
- Jeder sollte in der Lage sein NN zu trainieren (im besten Fall sogar Maschinen!)

Motivation

- Ein gutes neuronales Netz zu trainieren ist schwer, braucht viel Arbeitszeit und Erfahrung
- Jeder sollte in der Lage sein NN zu trainieren (im besten Fall sogar Maschinen!)

Lösung: Ein automatisches (und effizientes) System, welches gute Hyperparameter auswählt, muss her!

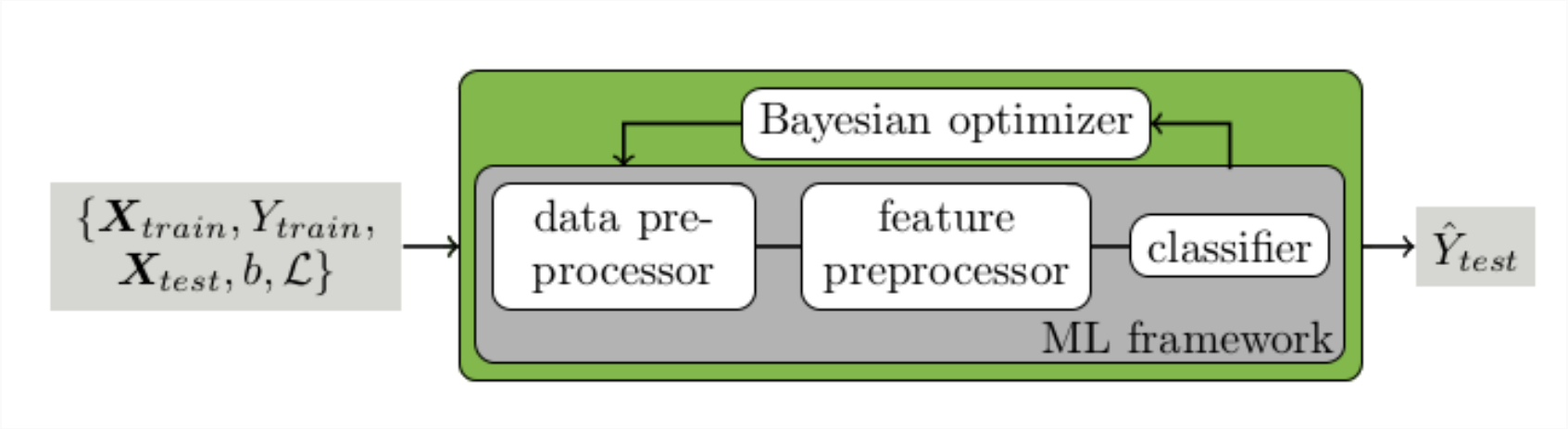
Automated Machine Learning in a Nutshell

Auto-ML I

Grundlegende Idee

- Starten mit *irgendwie* ausgewählten Hyperparametern
- Classifier trainieren
- Classifier evaluieren
- Hyperparametertuning mithilfe eines Bayesian optimizer
- Wiederholung bis zu einem zufriedenstellenden Ergebnis

Auto-ML II


$\{\boldsymbol{X}_{train}, Y_{train}, \boldsymbol{X}_{test}, b, \mathcal{L}\}$
Bayesian optimizer
data pre-processor
feature preprocessor
classifier
ML framework
$\hat{Y}_{test}$

Initialisierung der Hyperparameter

Bild zu problemen mit der initialisierung von hyperparametern

Rapidly Exploring Random Tree (RTT) I

Idee

- Werte zufällig wählen
- So oft wiederholen, bis man einen guten Überblick über den Searchspace hat

Rapidly Exploring Random Tree (RTT) I

Idee

- Werte zufällig wählen
- So oft wiederholen, bis man einen guten Überblick über den Searchspace hat

Vorteil

Bietet beliebig guten Überblick über den Searchspace

Rapidly Exploring Random Tree (RTT) I

Idee

- Werte zufällig wählen
- So oft wiederholen, bis man einen guten Überblick über den Searchspace hat

Vorteil

Bietet beliebig guten Überblick über den Searchspace

Achtung

RTT bietet einige Fallstricke. Think before use!

Rapidly Exploring Random Tree (RTT) II

Probleme

- Ausgesprochen Rechenintensiv
- Unterschiedliche Lernverfahren?
- Es gibt kein “best” Lernverfahren, nur “best at”
- Manche ML-Verfahren erfordern intensive Hyperparameteroptimisierung
- Bayes optimization sollte sich jedoch um dieses Problem kümmern

Meta Learning

Meta Learning

Idee

- Das richte ML-Verfahren für ein bestimmtes Datenset hängt vom Datenset selbst ab
- Ein bestimmter Klassifikator sollte auf ähnlichen Datensets ähnlich gute Ergebnisse liefern

Meta Learning

Idee

- Das richte ML-Verfahren für ein bestimmtes Datenset hängt vom Datenset selbst ab
- Ein bestimmter Klassifikator sollte auf ähnlichen Datensets ähnlich gute Ergebnisse liefern

$\Rightarrow$ Also bauen wir uns einen Klassifikator, der uns anhand eines neuen Datensets sagt, welche Art von Klassifikator wir trainieren sollten

Im Detail

Erstellung des Meta-Klassifikators

- Für eine (große) Menge an bereits bekannten Sets: Metafeatures berechnen
  $\Rightarrow$ class-probability, categorical-numerical-ratio, number of classes/features/instances (missing)
- Einen Klassifikator (SMAC) auf diesen Meta-Features trainieren

Im Detail

Erstellung des Meta-Klassifikators

- Für eine (große) Menge an bereits bekannten Sets: Metafeatures berechnen
  $\Rightarrow$ class-probability, categorical-numerical-ratio, number of classes/features/instances (missing)
- Einen Klassifikator (SMAC) auf diesen Meta-Features trainieren

Auswertung

- Für ein neues Datenset werden anhand der L_1 Distanz zu den bereits bekannten Datensets Klassifikatoren ausgewählt

Ensembles

Bestandsaufnahme

- Mehrere vielversprechende ML-Verfahren ausgewählt
- Jedes davon aufwändig mit Bayesian optimizer im Hinblick auf Hyperparameter getunt
- Das beste der Besten herausgepickt und die anderen weggeworfen

Bestandsaufnahme

- Mehrere vielversprechende ML-Verfahren ausgewählt
- Jedes davon aufwändig mit Bayesian optimizer im Hinblick auf Hyperparameter getunt
- Das beste der Besten herausgepickt und die anderen weggeworfen

Warum eigentlich?

Ensembles

Idee

- Anstatt teuer optimierte Klassifikatoren wegzuwerfen, Kombination der Besten

Ensembles

Idee

- Anstatt teuer optimierte Klassifikatoren wegzuwerfen, Kombination der Besten

Aber wie kombinieren?

- Alle ungewichtet aufsummieren?
- Stacking?
- gradient-free numerical optimization?

Ensemble Selection

Wie baut man ein Ensemble?

- Starte mit einem leeren Ensemble
- Füge den Klassifikator dem Ensemble hinzu, der das Ensemble am besten ergänzt
- Wiederhole bis alle Klassifikatoren enthalten sind oder X mal
- Durchschnitt über alle Predictions bilden für Resultat

Ensemble Selection

Wie baut man ein Ensemble?

- Starte mit einem leeren Ensemble
- Füge den Klassifikator dem Ensemble hinzu, der das Ensemble am besten ergänzt
- Wiederhole bis alle Klassifikatoren enthalten sind oder X mal
- Durchschnitt über alle Predictions bilden für Resultat
- Alle Einträge sind ungewichtet
- Duplikationen sind erlaubt

Anwendung und Ergebnisse

Anwendung und Evaluation

Setup

- Meta-learning mit 38 Features für 24h auf 140 OpenML Datensets (2/3 zu 1/3)
- Bayesian optimizer auf den 25 besten, Ensemble mit Größe 50

Anwendung und Evaluation

Setup

- Meta-learning mit 38 Features für 24h auf 140 OpenML Datensets (2/3 zu 1/3)
- Bayesian optimizer auf den 25 besten, Ensemble mit Größe 50

Auffälligkeiten und Ergebnisse

- Meta-learning und Ensemble Selection verbessert bisherige Ansätze deutlich
- Besonders bei kurzer Rechenzeit performt die Kombination ML+ES signifikant besser
- Mithilfe von ML+ES wird in der Regel ein hinreichend optimaler Klassifikator gefunden

Hinreichende Optimalität

OpenML dataset ID	AUTO-SKLEARN	AdaBoost	Bernoulli naïve Bayes	decision tree	extreml. rand. trees	Gaussian naïve Bayes	gradient boosting	kNN	LDA	linear SVM	kernel SVM	multinomial naïve Bayes	passive aggresive	QDA	random forest	Linear Class. (SGD)
38	2.15	2.68	50.22	2.15	18.06	11.22	**1.77**	50.00	8.55	16.29	17.89	46.99	50.00	8.78	2.34	15.82
46	3.76	4.65	-	5.62	4.74	7.88	**3.49**	7.57	8.67	8.31	5.36	7.55	9.23	7.57	4.20	7.31
179	**16.99**	17.03	19.27	18.31	17.09	21.77	17.00	22.23	18.93	17.30	17.57	18.97	22.29	19.06	17.24	17.01
184	**10.32**	10.52	-	17.46	11.10	64.74	10.42	31.10	35.44	15.76	12.52	27.13	20.01	47.18	10.98	12.76
554	1.55	2.42	-	12.00	2.91	10.52	3.86	2.68	3.34	2.23	**1.50**	10.37	100.00	2.75	3.08	2.50
772	46.85	49.68	47.90	47.75	**45.62**	48.83	48.15	48.00	46.74	48.38	48.66	47.21	48.75	47.67	47.71	47.93
917	10.22	9.11	25.83	11.00	10.22	33.94	10.11	11.11	34.22	18.67	**6.78**	25.50	20.67	30.44	10.83	18.33
1049	12.93	**12.53**	15.50	19.31	17.18	26.23	13.38	23.80	25.12	17.28	21.44	26.40	29.25	21.38	13.75	19.92
1111	23.70	23.16	28.40	24.40	24.47	29.59	**22.93**	50.30	24.11	23.99	23.56	27.67	43.79	25.86	28.06	23.36
1120	13.81	**13.54**	18.81	17.45	13.86	21.50	13.61	17.23	15.48	14.94	14.17	18.33	16.37	15.62	13.70	14.66
1128	4.21	4.89	4.71	9.30	3.89	4.77	4.58	4.59	4.58	4.83	4.59	4.46	5.65	5.59	**3.83**	4.33
293	2.86	4.07	24.30	5.03	3.59	32.44	24.48	4.86	24.40	14.16	100.00	24.20	21.34	28.68	**2.57**	15.54
389	19.65	22.98	-	33.14	19.38	29.18	19.20	30.87	19.68	**17.95**	22.04	20.04	20.14	39.57	20.66	17.99

Abbildung 1: Median balanced test error rate (BER) of optimizing AUTO - SKLEARN subspaces for each classification SVM method

Recap

Errungenschaften

- Automatisiertes Maschinenlernen!
- Gute Ergebnisse sind auch ohne Vorwissen oder genaue Kenntnisse der einzelnen Verfahre möglich

Recap

Errungenschaften

- Automatisiertes Maschinenlernen!
- Gute Ergebnisse sind auch ohne Vorwissen oder genaue Kenntnisse der einzelnen Verfahre möglich

bleibende/ ungelöste Probleme

- Rechenkosten
- Für sehr rechenaufwendige Verfahren (z.b. Deep Learning) nur eingeschränkt zu gebrauchen

Recap

Errungenschaften

- Automatisiertes Maschinenlernen!
- Gute Ergebnisse sind auch ohne Vorwissen oder genaue Kenntnisse der einzelnen Verfahre möglich

bleibende/ ungelöste Probleme

- Rechenkosten
- Für sehr rechenaufwendige Verfahren (z.b. Deep Learning) nur eingeschränkt zu gebrauchen
- Wird aber durch recht gute Parallelisierbarkeit teilweise wieder ausgeglichen

Vielen Dank für eure Aufmerksamkeit!

Fragerunde

Quellen


M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2962–2970, Curran Associates, Inc., 2015.