Towards Automatically-Tuned Neural Networks

Hector Mendoza, Aaron Klein, Matthias Feurer, Tobias Jost Springenberg, Frank Hutter
{mendozah, kleinaa, feurerm, springj, fh} @cs.uni-freiburg.de
Department of Computer Science, University of Freiburg

In a nutshell

- **Deep Learning** has become a powerful machine learning tool, but is still **complicated to use** for non-experts on the field.
- Requires human expert input in the **setting of hyperparamenters**, but is an expensive and not straightforward task.
- **Auto-sklearn** has been used in the past for automated configuration of algorithms
- We include **neural networks** as an extra classifier and regressor to use inside auto-sklearn machinery
- Automatically configured networks proved to be reliably and robust, winning three datasets on the AutoML Challenge

Inside auto-sklearn

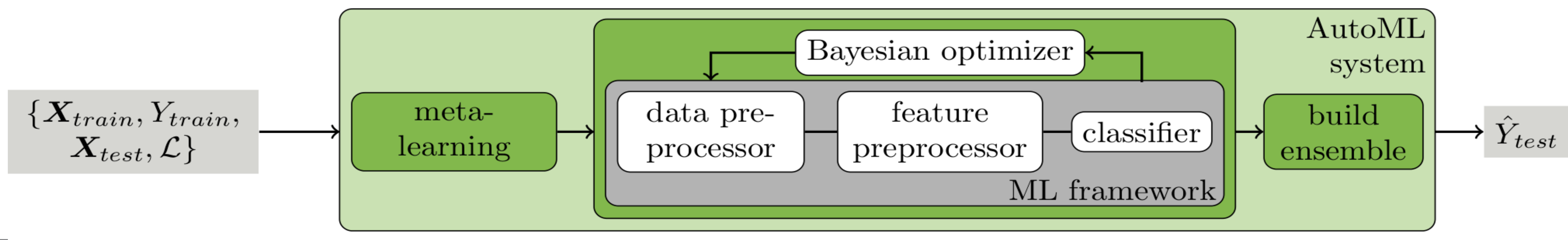
- Bayesian optimization is a powerful hyperparameter optimization tool.
- Specially around highly conditional, mixed discrete/continuous hyperparameter space, just like the one of neural networks
- Configurable machine learning pipeline, originally built around scikit-learn.
- Uses **SMAC**, a tree based method for BO instead of *GPs*

Auto-sklearn combines:

- Preprocessing methods: Feature selection or dimensionality reduction that speeds up neural network training or improve performance.
- Data preprocessing: Imputation, balancing and rescaling of the input data
- Ensemble selection: Better prediction than single models. Chosen greedily allowing repetitions of models.

- Methods that are particulary helpful to nets:
- To handle sparsity
  - **Truncated SVD**
  - Densifier
- To reduce dimensionality
  - **Gaussian Random Embedding**
  - Nystroem Sampler
- But there's **no silver bullet**, it depends on the dataset and task type

Bottom: The AutoML workflow as used in auto-sklearn. We use four data preprocessors and choose between 13 feature preprocessors to help Autonet



- **We do this to include only neural networks, but also as a complement to the 16 classifiers and regressors inside scikit-learn**

Networks' Plug & Play

Implement **Autonet** as a component inside auto-sklearn system

- Independent of model implementation
- Initial neural network model using Theano and Lasagne python libraries
- Most of the cases are already implemented on lasagna package. Only *smorms* solver was specially implemented.
- Handles sparse datasets out of the box, multilabel, regression and binary and multiclass classification.
- Several conditional parameters based on the number of layers. e.g. Units on layer 4 only active if number of layers is 4 or bigger

Possible extensions:

- Add more hyperparameters such as loss functions or L1-regularization
- Include custom learning rate policies or solvers or parameterized layers

Available on `github.com/hmendozap/auto-sklearn@development_java`

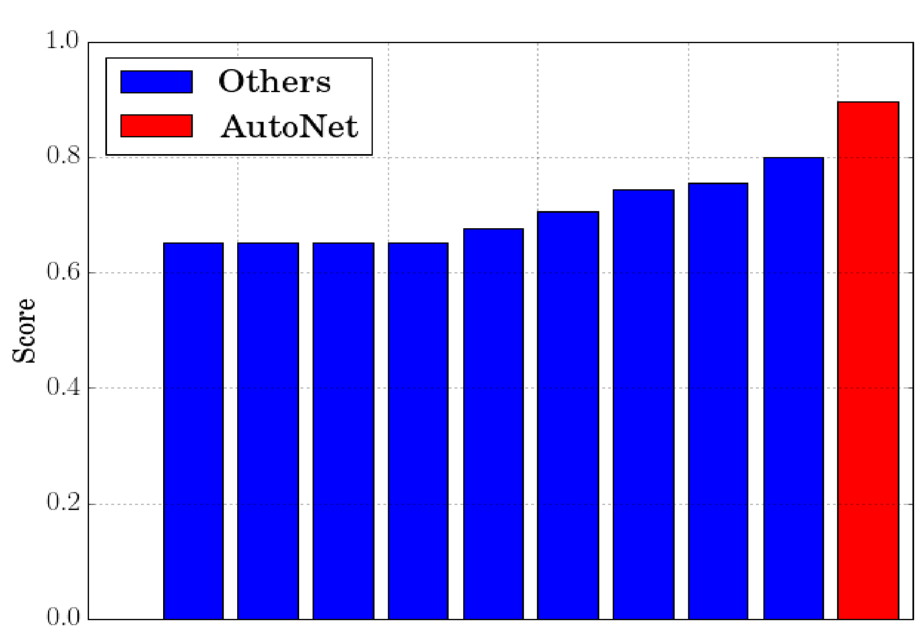
Neural Network Hyperparameter space: A total of 57 hyperparameters with combinations of categorical, integers, floats and layer dependent.

	Name	Range	Default	log scale	Type	Conditional
Network hyperparameters	batch size	$[32, 4096]$	32	✓	float	-
	number of updates	$[50, 2500]$	200	✓	int	-
	number of layers	$[1, 6]$	1	-	int	-
	learning rate	$[10^{-6}, 1.0]$	10^{-2}	✓	float	-
	L_2 regularization	$[10^{-7}, 10^{-2}]$	10^{-4}	✓	float	-
	dropout output layer	$[0.0, 0.99]$	0.5	✓	float	-
	solver type	{SGD, Momentum, Adam, Adadelta, Adagrad, smorm, Nesterov }	smorm3s	-	cat	-
	lr-policy	{Fixed, Inv, Exp, Step}	fixed	-	cat	-
Conditioned on solver type	β_1	$[10^{-4}, 10^{-1}]$	10^{-1}	✓	float	✓
	β_2	$[10^{-4}, 10^{-1}]$	10^{-1}	✓	float	✓
	ρ	$[0.05, 0.99]$	0.95	✓	float	✓
	momentum	$[0.3, 0.999]$	0.9	✓	float	✓
Conditioned on lr-policy	γ	$[10^{-3}, 10^{-1}]$	10^{-2}	✓	float	✓
	k	$[0.0, 1.0]$	0.5	-	float	✓
	s	$[2, 20]$	2	-	int	✓
Per-layer hyperparameters	activation-type	{Sigmoid, TanH, ScaledTanH, ELU, ReLU, Leaky, Linear}	ReLU	-	cat	✓
	number of units	$[64, 4096]$	128	✓	int	✓
	dropout in layer	$[0.0, 0.99]$	0.5	-	float	✓
	weight initialization	{Constant, Normal, Uniform, Glorot-Uniform, Glorot-Normal, He-Normal, He-Uniform, Orthogonal, Sparse}	He-Normal	-	cat	✓
	std. normal init.	$[10^{-7}, 0.1]$	0.0005	-	float	✓
	leakiness	$[0.01, 0.99]$	$\frac{1}{3}$	-	float	✓
	tanh scale in	$[0.5, 1.0]$	2/3	-	float	✓
	tanh scale out	$[1.1, 3.0]$	1.7159	✓	float	✓

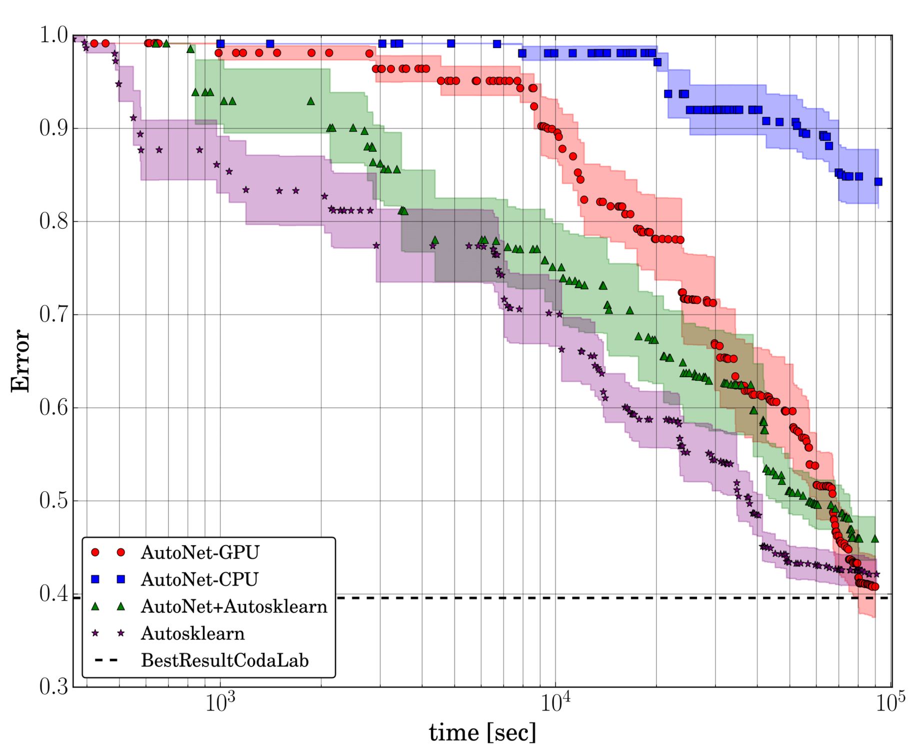
Baseline and AutoML ChallengeResults

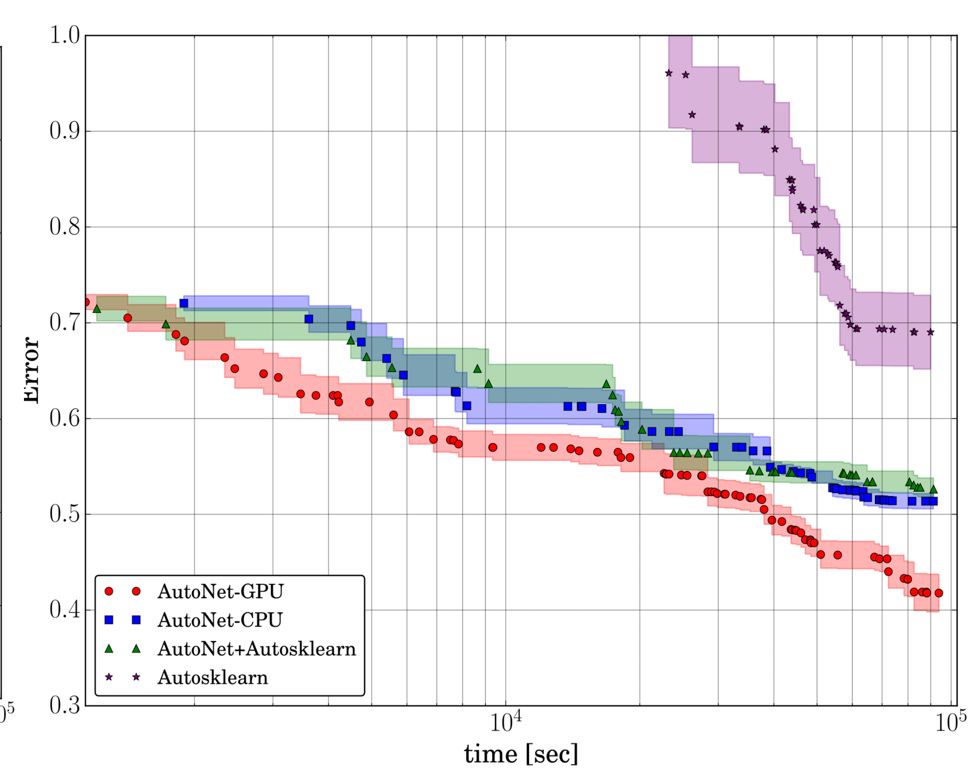
- Compare CPU-, GPU-Autonet, auto-sklearn and Autonet + autosklearn
- Tested on five datasets of phase-0 in AutoML Challenge
- GPU version a **order** of magnitude faster
- GPU-Autonet better on one dataset, tied on other 3 and worst on only one.
- GPU-Autonet won datasets on Phase 4 (alexis) and Phase 5 (tania) of AutoML Challenge.

Official competition results for dataset `alexis` (bottom).



- 3rd. Place on GPU track
- To our knowledge first automatically tuned neural network to win a competition dataset
- Ensemble winning consisted of 8 1-layer networks, 2 2-layer networks and logistic regressor trained with SGD
- Autonet-GPU and -CPU outperformed autosklearn on a `tania` dataset.
- The combination Autonet + autosklearn saw an increase in performance despite the increase in the configuration space.





Ensemble performance on dataset `tania` (right) and `newsgroups` (left) over time. Cross-validation performance on training set due to lack of test set availaibility.