Towards Automatically-Tuned Neural Networks
Hector Mendoza, Aaron Klein, Matthias Feurer, Tobias Jost Springenberg, Frank Hutter
{mendoza, kleinaa, feurerm, spring, 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 hyperparameters, 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 GP

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

- Methods that are particularly helpful to nets:
  - To handle sparsity
    - Truncated SVD
    - Densifier
  - To reduce dimensionality
    - Gaussian Random Embedding
    - Nyström Sampler

  But there’s no silver bullet, it depends on the dataset and task type

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

Baseline and AutoML ChallengeResults

- Compare CPU-, GPU-Autonet, auto-sklearn and Autonet + autosklear
- 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 (tanía) of AutoML Challenge.

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 tanía dataset.

The combination Autonet + autosklear saw an increase in performance despite the increase in the configuration space.

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 amorns 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/mendoza-a/auto-sklearn-development_java

Networks’ Plug & Play

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

**Baseline and AutoML ChallengeResults**

- Compare CPU-, GPU-Autonet, auto-sklearn and Autonet + autosklear
- 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 (tanía) of AutoML Challenge.

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 tanía dataset.

The combination Autonet + autosklear saw an increase in performance despite the increase in the configuration space.

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 amorns 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/mendoza-a/auto-sklearn-development_java

Networks’ Plug & Play

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

**Baseline and AutoML ChallengeResults**

- Compare CPU-, GPU-Autonet, auto-sklearn and Autonet + autosklear
- 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 (tanía) of AutoML Challenge.

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 tanía dataset.

The combination Autonet + autosklear saw an increase in performance despite the increase in the configuration space.