AutoML: Neural Architecture Search (NAS)

Practical Recommendations for NAS and HPO

Bernd Bischl   Frank Hutter   Lars Kotthoff
Marius Lindauer   Joaquin Vanschoren

Maturity of the Fields of NAS and HPO

- Hyperparameter optimization is a mature field
  - Blackbox HPO has been researched for decades; there are many software packages
  - Multi-fidelity HPO has also become quite mature

Maturity of the Fields of NAS and HPO

- Hyperparameter optimization is a mature field
  - Blackbox HPO has been researched for decades; there are many software packages
  - Multi-fidelity HPO has also become quite mature
- Neural architecture search is still a very young field

Maturity of the Fields of NAS and HPO

- Hyperparameter optimization is a mature field
  - Blackbox HPO has been researched for decades; there are many software packages
  - Multi-fidelity HPO has also become quite mature

- Neural architecture search is still a very young field
  - Blackbox is quite mature, but slow
  - Multi-fidelity NAS is also quite mature and faster

Maturity of the Fields of NAS and HPO

- Hyperparameter optimization is a mature field
  - Blackbox HPO has been researched for decades; there are many software packages
  - Multi-fidelity HPO has also become quite mature
- Neural architecture search is still a very young field
  - Blackbox is quite mature, but slow
  - Multi-fidelity NAS is also quite mature and faster
  - Meta-learning + multi-fidelity NAS is fast, but is still a very young field

Maturity of the Fields of NAS and HPO

- Hyperparameter optimization is a mature field
  - Blackbox HPO has been researched for decades;
    there are many software packages
  - Multi-fidelity HPO has also become quite mature

- Neural architecture search is still a very young field
  - Blackbox is quite mature, but slow
  - Multi-fidelity NAS is also quite mature and faster
  - Meta-learning + multi-fidelity NAS is fast, but is still a very young field
  - Gradient-based NAS is fast, but can have failure modes with terrible performance
  - Gradient-based NAS hasn't reached the hands-off AutoML stage yet

Maturity of the Fields of NAS and HPO

- Hyperparameter optimization is a mature field
  - Blackbox HPO has been researched for decades;
    there are many software packages
  - Multi-fidelity HPO has also become quite mature
- Neural architecture search is still a very young field
  - Blackbox is quite mature, but slow
  - Multi-fidelity NAS is also quite mature and faster
  - Meta-learning + multi-fidelity NAS is fast, but is still a very young field
  - Gradient-based NAS is fast, but can have failure modes with terrible performance
  - Gradient-based NAS hasn't reached the hands-off AutoML stage yet
- NAS is mainly used to create new architectures that many others can reuse

Maturity of the Fields of NAS and HPO

- Hyperparameter optimization is a mature field
  - Blackbox HPO has been researched for decades;
    there are many software packages
  - Multi-fidelity HPO has also become quite mature

- Neural architecture search is still a very young field
  - Blackbox is quite mature, but slow
  - Multi-fidelity NAS is also quite mature and faster
  - Meta-learning + multi-fidelity NAS is fast, but is still a very young field
  - Gradient-based NAS is fast, but can have failure modes with terrible performance
  - Gradient-based NAS hasn't reached the hands-off AutoML stage yet

- NAS is mainly used to create new architectures that many others can reuse

- Given a new dataset, HPO is crucial for good performance; NAS may not be necessary
  - The biggest gains typically come from tuning key hyperparameters (learning rate, etc)
  - Reusing a previous achitecture often yields competitive results

Practical Recommendations for NAS and HPO

- Recommendations for a new dataset
  - → Always run HPO
  - → Try NAS if you can

Practical Recommendations for NAS and HPO

- Recommendations for a new dataset
  - → Always run HPO
  - → Try NAS if you can

- How to combine NAS & HPO
  - If the compute budget suffices, optimize them jointly, e.g., using BOHB
    - + Auto-PyTorch Tabular [Zimmer, Lindauer & Hutter, 2020]
    - + Auto-RL [Runge et al. 2019]

Practical Recommendations for NAS and HPO

- Recommendations for a new dataset
  - → Always run HPO
  - → Try NAS if you can
- How to combine NAS & HPO
  - If the compute budget suffices, optimize them jointly, e.g., using BOHB
    - + Auto-PyTorch Tabular [Zimmer, Lindauer & Hutter, 2020]
    - + Auto-RL [Runge et al. 2019]
  - Else
    - + If you have decent hyperparameters:
      run NAS, followed by HPO for fine-tuning [Saikat et al. 2019]

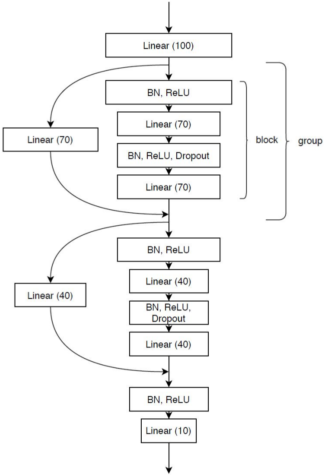
Practical Recommendations for NAS and HPO

- Recommendations for a new dataset
  - → Always run HPO
  - → Try NAS if you can

- How to combine NAS & HPO
  - If the compute budget suffices, optimize them jointly, e.g., using BOHB
    - + Auto-PyTorch Tabular [Zimmer, Lindauer & Hutter, 2020]
    - + Auto-RL [Runge et al. 2019]
  - Else
    - + If you have decent hyperparameters: run NAS, followed by HPO for fine-tuning [Saikat et al. 2019]
    - + If you don't have decent hyperparameters: first run HPO to get competitive

Case Study: NAS & HPO in Auto-PyTorch Tabular [Zimmer, Lindauer & Hutter, 2020]

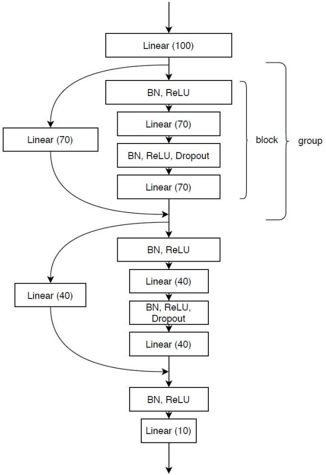
- Joint Architecture Search and Hyperparameter Optimization



	Name	Range	log	type	cond.
Archi-tecture	network type	[ResNet, MLPNet]	-	cat	-
	num layers (MLP)	[1, 6]	-	int	✓
	max units (MLP)	[64, 1024]	✓	int	✓
	max dropout (MLP)	[0, 1]	-	float	✓
	num groups (Res)	[1, 5]	-	int	✓
	blocks per group (Res)	[1, 3]	-	int	✓
	max units (Res)	[32, 512]	✓	int	✓
	use dropout (Res)	[F, T]	-	bool	✓
	use shake drop	[F, T]	-	bool	✓
	use shake shake	[F, T]	-	bool	✓
	max dropout (Res)	[0, 1]	-	float	✓
	max shake drop (Res)	[0, 1]	-	float	✓
Hyper-para-meters	batch size	[16, 512]	✓	int	-
	optimizer	[SGD, Adam]	-	cat	-
	learning rate (SGD)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (SGD)	[1e-5, 1e-1]	-	float	✓
	momentum	[0.1, 0.999]	-	float	✓
	learning rate (Adam)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (Adam)	[1e-5, 1e-1]	-	float	✓
	training technique	[standard, mixup]	-	cat	-
	mixup alpha	[0, 1]	-	float	✓
	preprocessor	[none, trunc. SVD]	-	cat	-
	SVD target dim	[10, 256]	-	int	✓

Case Study: NAS & HPO in Auto-PyTorch Tabular [Zimmer, Lindauer & Hutter, 2020]

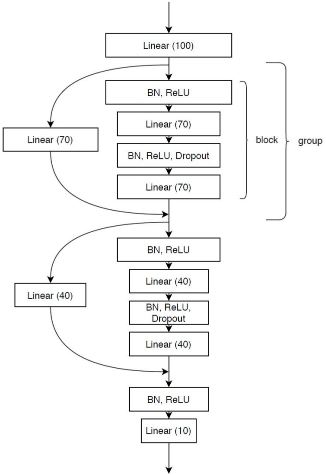
- Joint Architecture Search and Hyperparameter Optimization
  - Purely using HPO techniques: very similar methods as in Auto-sklearn 2.0



	Name	Range	log	type	cond.
Archi-tecture	network type	[ResNet, MLPNet]	-	cat	-
	num layers (MLP)	[1, 6]	-	int	✓
	max units (MLP)	[64, 1024]	✓	int	✓
	max dropout (MLP)	[0, 1]	-	float	✓
	num groups (Res)	[1, 5]	-	int	✓
	blocks per group (Res)	[1, 3]	-	int	✓
	max units (Res)	[32, 512]	✓	int	✓
	use dropout (Res)	[F, T]	-	bool	✓
	use shake drop	[F, T]	-	bool	✓
	use shake shake	[F, T]	-	bool	✓
	max dropout (Res)	[0, 1]	-	float	✓
	max shake drop (Res)	[0, 1]	-	float	✓
Hyper-para-meters	batch size	[16, 512]	✓	int	-
	optimizer	[SGD, Adam]	-	cat	-
	learning rate (SGD)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (SGD)	[1e-5, 1e-1]	-	float	✓
	momentum	[0.1, 0.999]	-	float	✓
	learning rate (Adam)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (Adam)	[1e-5, 1e-1]	-	float	✓
	training technique	[standard, mixup]	-	cat	-
	mixup alpha	[0, 1]	-	float	✓
	preprocessor	[none, trunc. SVD]	-	cat	-
	SVD target dim	[10, 256]	-	int	✓

Case Study: NAS & HPO in Auto-PyTorch Tabular [Zimmer, Lindauer & Hutter, 2020]

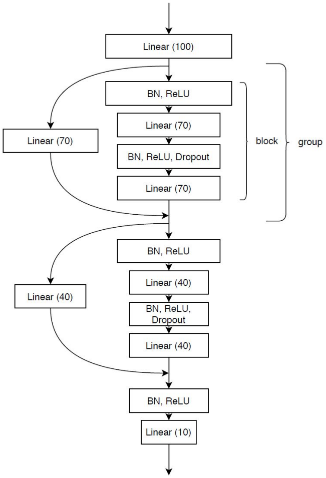
- Joint Architecture Search and Hyperparameter Optimization
  - Purely using HPO techniques: very similar methods as in Auto-sklearn 2.0
  - Multi-fidelity optimization with BOHB
  - Meta-learning with task-independent recommendations



	Name	Range	log	type	cond.
Archi-tecture	network type	[ResNet, MLPNet]	-	cat	-
	num layers (MLP)	[1, 6]	-	int	✓
	max units (MLP)	[64, 1024]	✓	int	✓
	max dropout (MLP)	[0, 1]	-	float	✓
	num groups (Res)	[1, 5]	-	int	✓
	blocks per group (Res)	[1, 3]	-	int	✓
	max units (Res)	[32, 512]	✓	int	✓
	use dropout (Res)	[F, T]	-	bool	✓
	use shake drop	[F, T]	-	bool	✓
	use shake shake	[F, T]	-	bool	✓
	max dropout (Res)	[0, 1]	-	float	✓
	max shake drop (Res)	[0, 1]	-	float	✓
Hyper-para-meters	batch size	[16, 512]	✓	int	-
	optimizer	[SGD, Adam]	-	cat	-
	learning rate (SGD)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (SGD)	[1e-5, 1e-1]	-	float	✓
	momentum	[0.1, 0.999]	-	float	✓
	learning rate (Adam)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (Adam)	[1e-5, 1e-1]	-	float	✓
	training technique	[standard, mixup]	-	cat	-
	mixup alpha	[0, 1]	-	float	✓
	preprocessor	[none, trunc. SVD]	-	cat	-
	SVD target dim	[10, 256]	-	int	✓

Case Study: NAS & HPO in Auto-PyTorch Tabular [Zimmer, Lindauer & Hutter, 2020]

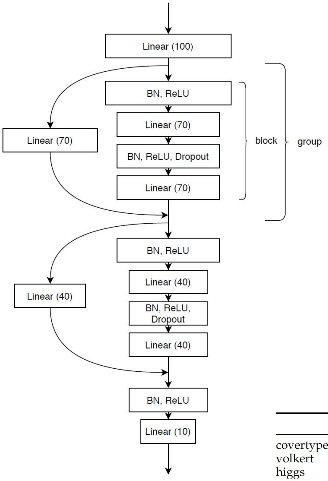
- Joint Architecture Search and Hyperparameter Optimization
  - Purely using HPO techniques: very similar methods as in Auto-sklearn 2.0
  - Multi-fidelity optimization with BOHB
  - Meta-learning with task-independent recommendations
  - Ensembling of neural nets and traditional ML



	Name	Range	log	type	cond.
Archi-tecture	network type	[ResNet, MLPNet]	-	cat	-
	num layers (MLP)	[1, 6]	-	int	✓
	max units (MLP)	[64, 1024]	✓	int	✓
	max dropout (MLP)	[0, 1]	-	float	✓
	num groups (Res)	[1, 5]	-	int	✓
	blocks per group (Res)	[1, 3]	-	int	✓
	max units (Res)	[32, 512]	✓	int	✓
	use dropout (Res)	[F, T]	-	bool	✓
	use shake drop	[F, T]	-	bool	✓
	use shake shake	[F, T]	-	bool	✓
	max dropout (Res)	[0, 1]	-	float	✓
	max shake drop (Res)	[0, 1]	-	float	✓
Hyper-para-meters	batch size	[16, 512]	✓	int	-
	optimizer	[SGD, Adam]	-	cat	-
	learning rate (SGD)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (SGD)	[1e-5, 1e-1]	-	float	✓
	momentum	[0.1, 0.999]	-	float	✓
	learning rate (Adam)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (Adam)	[1e-5, 1e-1]	-	float	✓
	training technique	[standard, mixup]	-	cat	-
	mixup alpha	[0, 1]	-	float	✓
	preprocessor	[none, trunc. SVD]	-	cat	-
	SVD target dim	[10, 256]	-	int	✓

Case Study: NAS & HPO in Auto-PyTorch Tabular [Zimmer, Lindauer & Hutter, 2020]

- Joint Architecture Search and Hyperparameter Optimization
  - Purely using HPO techniques: very similar methods as in Auto-sklearn 2.0
  - Multi-fidelity optimization with BOHB
  - Meta-learning with task-independent recommendations
  - Ensembling of neural nets and traditional ML



	Name	Range	log	type	cond.
Architecture	network type	[ResNet, MLPNet]	-	cat	-
	num layers (MLP)	[1, 6]	-	int	✓
	max units (MLP)	[64, 1024]	✓	int	✓
	max dropout (MLP)	[0, 1]	-	float	✓
	num groups (Res)	[1, 5]	-	int	✓
	blocks per group (Res)	[1, 3]	-	int	✓
	max units (Res)	[32, 512]	✓	int	✓
	use dropout (Res)	[F, T]	-	bool	✓
	use shake drop	[F, T]	-	bool	✓
	use shake shake	[F, T]	-	bool	✓
	max dropout (Res)	[0, 1]	-	float	✓
	max shake drop (Res)	[0, 1]	-	float	✓
Hyper-parameters	batch size	[16, 512]	✓	int	-
	optimizer	[SGD, Adam]	-	cat	-
	learning rate (SGD)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (SGD)	[1e-5, 1e-1]	-	float	✓
	momentum	[0.1, 0.999]	-	float	✓
	learning rate (Adam)	[1e-4, 1e-1]	✓	float	✓
	L2 reg. (Adam)	[1e-5, 1e-1]	-	float	✓
	training technique	[standard, mixup]	-	cat	-
	mixup alpha	[0, 1]	-	float	✓
	preprocessor	[none, trunc. SVD]	-	cat	-
	SVD target dim	[10, 256]	-	int	✓

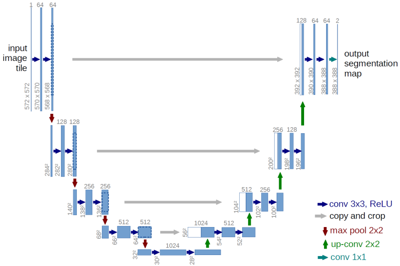
	Auto-PyTorch	AutoGluon	AutoKeras	Auto-Sklearn	hyperopt-sklearn
covertype	**96.86 ± 0.41**	-	61.61 ± 3.52	-	-
volkert	**79.46 ± 0.43**	68.34 ± 0.10	44.25 ± 2.38	67.32 ± 0.46	-
higgs	**73.01 ± 0.09**	72.6 ± 0.00	71.25 ± 0.29	72.03 ± 0.33	-
car	**99.22 ± 0.02**	97.19 ± 0.35	93.39 ± 2.82	98.42 ± 0.62	98.95 ± 0.96
mfeat-factors	**99.10 ± 0.18**	98.03 ± 0.23	97.73 ± 0.23	98.64 ± 0.39	97.88 ± 38.48
apsfailure	99.32 ± 0.01	**99.5 ± 0.03**	-	99.43 ± 0.04	-
phoneme	**90.59 ± 0.13**	89.62 ± 0.06	86.76 ± 0.12	89.26 ± 0.14	89.79 ± 4.54
dibert	**99.04 ± 0.15**	98.17 ± 0.05	96.51 ± 0.62	98.14 ± 0.47	-

Case Study: NAS & HPO in Auto-DispNet

- Problem: disparity estimation
  - Estimate depth from stereo images

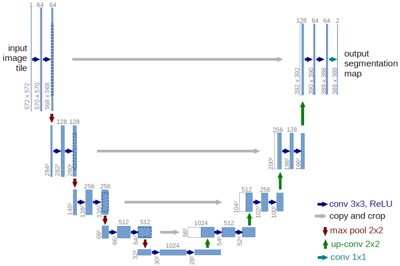
Case Study: NAS & HPO in Auto-DispNet

- Problem: disparity estimation
  - Estimate depth from stereo images
- Background: U-Net
  - Skip connections from similar spatial resolution to avoid loosing information



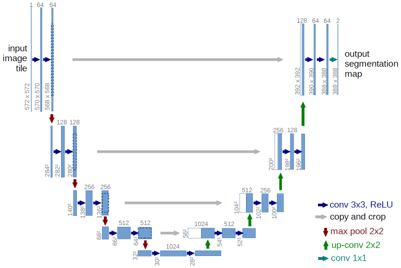
Case Study: NAS & HPO in Auto-DispNet

- Problem: disparity estimation
  - Estimate depth from stereo images
- Background: U-Net
  - Skip connections from similar spatial resolution to avoid loosing information
- Search space for DARTS
  - 3 cells: keeping spatial resolution, downsampling, and new upsampling cell that supports U-Net like skip connections



Case Study: NAS & HPO in Auto-DispNet

- Problem: disparity estimation
  - Estimate depth from stereo images
- Background: U-Net
  - Skip connections from similar spatial resolution to avoid loosing information
- Search space for DARTS
  - 3 cells: keeping spatial resolution, downsampling, and new upsampling cell that supports U-Net like skip connections
- Both NAS and HPO improved the state of the art [Saikat et al. 2019]:
  - End-point-error (EPE) on Sintel dataset: 2.36 $\rightarrow$ 2.14 (by DARTS)
  - Subsequent HPO: 2.14 $\rightarrow$ 1.94 (by BOHB)



Questions to Answer for Yourself / Discuss with Friends

- Repetition:
  If you want to use both HPO and NAS for your problem, how could you proceed?
- Discussion:
  Think of a problem of your particular interest. For that problem, which approach would you use to combine HPO and NAS, and why?