AutoML Systems in Action

Xia “Ben” Hu

Associate Professor and Lynn '84 and Bill Crane '83 Faculty Fellow
Department of Computer Science and Engineering
Texas A&M University
Email: xiahu@tamu.edu
What Is Automated Machine Learning (AutoML) ?

To make machine learning accessible

- To people with *limited ML background* such experts from other domains, as well as those *experienced data scientists*
- By automating the *end-to-end process from data to the results*
- Shown competitive performance on supervised image and text classifications
AutoML: System Components & Search Loop

1. Generate
   Generate the next architecture for observation

2. Observe
   Train the architecture and evaluate its performance

3. Update
   Update the surrogate model with training history (architectures and their performances)

---

Road Map

1. AutoML System for Deep Learning

   AutoKeras
   Tests passing, codecov 100%, pypi package 1.0.6

2. AutoML System for Recommendation

   Item_ID #23
   Harry Potter and the Chamber of Secrets
   Author
   J.K. Rowling
   User_ID #76
   Item_ID #2
   Harry Potter and the Sorcerer's Stone

3. AutoML System for Outlier Detection

   Reconstruction based
   Density based
   One-class based
   Cluster based
Road Map

1. AutoML System for Deep Learning

   AutoKeras

2. AutoML System for Recommendation

   Item_ID #23
   Harry Potter and the Chamber of Secrets

   Author
   J.K. Rowling

   User_ID #76

   Item_ID #2
   Harry Potter and the Sorcerer's Stone

3. AutoML System for Outlier Detection

   Reconstruction based

   Density based

   One-class based

   Cluster based
Given a dataset, the AutoML system searches for the best neural architecture and hyperparameters.
Automation of Deep Learning (1/3)

3. Update
Update the surrogate model with training history (architectures and their performances)

1. Generate
Generate the next architecture for observation

2. Observe
Train the architecture and evaluate its performance

Number of Iterations
Average Observation Time

$O(n\bar{t})$
Automation of Deep Learning (2/3)

3. **Update**
   Update the surrogate model with training history (architectures and their performances)

1. **Generate**
   Generate the next architecture for observation

2. **Observe**
   Train the architecture and evaluate its performance

 Challenge to **Evaluation**
 Training neural networks from scratch takes a long time

\[ O(\Delta t) \]

 Challenge to **Search Algorithm**
 Evolutionary algorithm [1], reinforcement learning [2] need to go through the NAS loop too many times

---

3. **Update**
Update the surrogate model with training history (architectures and their performances)

**Solution to Evaluation**
Network morphism makes use of the weights in trained architectures

1. **Generate**
Generate the next architecture for observation

**Solution to Search Algorithm**
Bayesian optimization requires less number of observations

2. **Observe**
Train the architecture and evaluate its performance
Efficient Search Algorithm

Bayesian-guided network morphism in AutoKeras [1]

- Propose a new neural architecture search (NAS) method based on **Bayesian optimization** and **network morphism**

- Bayesian Optimization (BO) is widely used in AutoML (model selection, hyper-param tuning)

- We want to explore the capability of BO on NAS to make it more efficient


Speed Up the Evaluation Process

Network morphism (AutoKeras)

- Change the neural architecture while preserve the functionality
- Support 3 operations: wider layer/deeper layer/skip connection
1. User call from the *AutoKeras API*
2. The Bayesian *search* is conducted on **CPU**
3. The *training* of the neural network is on **GPU**
4. All *searched models are stored* on a storage device (hard disk)
**Searcher and Model Parallelization**

- GPU and CPU run *in parallel*
- Searcher passes a neural network to **GPU for training**
- Concurrently, searcher runs BO to **generate new neural architectures on CPU**
Experiment: Effectiveness

Datasets:
MNIST, CIFAR10, FASHION-MNIST.

Baseline methods:
Simple: grid search, random search
Traditional: SMAC\(^1\), SPMT\(^2\)
State-of-the-art: SEAS\(^3\), NASBOT\(^4\)
Variant: BFS, BO
Our Method: NASNM

Setup:
Single GPU, 12 hours.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MNIST</th>
<th>CIFAR10</th>
<th>FASHION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANDOM</td>
<td>1.79%</td>
<td>16.86%</td>
<td>11.36%</td>
</tr>
<tr>
<td>GRID</td>
<td>1.68%</td>
<td>17.17%</td>
<td>10.28%</td>
</tr>
<tr>
<td>SPMT</td>
<td>1.36%</td>
<td>14.68%</td>
<td>9.62%</td>
</tr>
<tr>
<td>SMAC</td>
<td>1.43%</td>
<td>15.04%</td>
<td>10.87%</td>
</tr>
<tr>
<td>SEAS</td>
<td>1.07%</td>
<td>12.43%</td>
<td>8.05%</td>
</tr>
<tr>
<td>NASBOT</td>
<td>NA</td>
<td>12.30%</td>
<td>NA</td>
</tr>
<tr>
<td>BFS</td>
<td>1.56%</td>
<td>13.84%</td>
<td>9.13%</td>
</tr>
<tr>
<td>BO</td>
<td>1.83%</td>
<td>12.90%</td>
<td>7.99%</td>
</tr>
<tr>
<td>AK</td>
<td>0.55%</td>
<td>11.44%</td>
<td>7.42%</td>
</tr>
<tr>
<td>AK-DP</td>
<td>0.60%</td>
<td>3.60%</td>
<td>6.72%</td>
</tr>
</tbody>
</table>

## Experiment: Effectiveness

Datasets:
MNIST, CIFAR10, FASHION-MNIST.

Baseline methods:
Simple: grid search, random search
Traditional: SMAC\(^1\), SPMT\(^2\)
State-of-the-art: SEAS\(^3\), NASBOT\(^4\)
Variant: BFS, BO
Our Method: NASNM

### Table 1: Classification Error Rate

<table>
<thead>
<tr>
<th>Methods</th>
<th>MNIST</th>
<th>CIFAR10</th>
<th>FASHION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANDOM</td>
<td>1.79%</td>
<td>16.86%</td>
<td>11.36%</td>
</tr>
<tr>
<td>GRID</td>
<td>1.68%</td>
<td>17.17%</td>
<td>10.28%</td>
</tr>
<tr>
<td>SPMT</td>
<td>1.36%</td>
<td>14.68%</td>
<td>9.62%</td>
</tr>
<tr>
<td>SMAC</td>
<td>1.43%</td>
<td>15.04%</td>
<td>10.87%</td>
</tr>
<tr>
<td>SEAS</td>
<td>1.07%</td>
<td>12.43%</td>
<td>8.05%</td>
</tr>
<tr>
<td>NASBOT</td>
<td>NA</td>
<td>12.30%</td>
<td>NA</td>
</tr>
<tr>
<td>BFS</td>
<td>1.56%</td>
<td>13.84%</td>
<td>9.13%</td>
</tr>
<tr>
<td>BO</td>
<td>1.83%</td>
<td>12.90%</td>
<td>7.99%</td>
</tr>
<tr>
<td>AK</td>
<td>0.55%</td>
<td>11.44%</td>
<td>7.42%</td>
</tr>
<tr>
<td>AK-DP</td>
<td>0.60%</td>
<td>3.60%</td>
<td>6.72%</td>
</tr>
</tbody>
</table>

Experiment: Effectiveness

Datasets:
MNIST, CIFAR10, FASHION-MNIST.

Baseline methods:
Simple: grid search, random search
Traditional: SMAC\textsuperscript{1}, SPMT\textsuperscript{2}
State-of-the-art: SEAS\textsuperscript{3}, NASBOT\textsuperscript{4}
Variant: BFS, BO
Our Method: NASNM

Setup:
Single GPU, 12 hours.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MNIST</th>
<th>CIFAR10</th>
<th>FASHION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANDOM</td>
<td>1.79%</td>
<td>16.86%</td>
<td>11.36%</td>
</tr>
<tr>
<td>GRID</td>
<td>1.68%</td>
<td>17.17%</td>
<td>10.28%</td>
</tr>
<tr>
<td>SPMT</td>
<td>1.36%</td>
<td>14.68%</td>
<td>9.62%</td>
</tr>
<tr>
<td>SMAC</td>
<td>1.43%</td>
<td>15.04%</td>
<td>10.87%</td>
</tr>
<tr>
<td>SEAS</td>
<td>1.07%</td>
<td>12.43%</td>
<td>8.05%</td>
</tr>
<tr>
<td>NASBOT</td>
<td>NA</td>
<td>12.30%</td>
<td>NA</td>
</tr>
<tr>
<td>BFS</td>
<td>1.56%</td>
<td>13.84%</td>
<td>9.13%</td>
</tr>
<tr>
<td>BO</td>
<td>1.83%</td>
<td>12.90%</td>
<td>7.99%</td>
</tr>
<tr>
<td>AK</td>
<td>0.55%</td>
<td>11.44%</td>
<td>7.42%</td>
</tr>
<tr>
<td>AK-DP</td>
<td>0.60%</td>
<td>3.60%</td>
<td>6.72%</td>
</tr>
</tbody>
</table>

**Experiment: Efficiency**

**Figure 4: Evaluation of Efficiency.** The two figures plot the same result with different X-axis. BFS uses network morphism. BO uses Bayesian optimization. AK uses both.
AutoKeras: An Open-source AutoML System

- AutoKeras provides easy-to-use solutions to deep learning tasks

Visit autokeras.com for more information
Road Map

1. AutoML System for Deep Learning

AutoKeras

2. AutoML System for Recommendation

User_ID #76
Harry Potter and the Sorcerer's Stone

Item_ID #2
Harry Potter and the Chamber of Secrets

Item_ID #23
Harry Potter and the Chamber of Secrets

Author
J.K. Rowling

3. AutoML System for Outlier Detection

Reconstruction based
Density based
One-class based
Cluster based

Data Analytics at Texas A&M Lab
Why Automated Recommendation

It is difficult to decide the proper architecture due to diverse feature interactions, heterogeneous data, and high data volume, etc.

Our Goal — Automate architectural design, and with better performance

- Abstract and modularize virtual blocks to formulate a generalizable search space for recommendation tasks, namely CTR and rating prediction
- Better search algorithm which improves performance

Scenario-Based Abstract Search Space

Selectable search algorithms for BOTH model selection and HP tuning

- **Searcher**
  - **Tuner:** Random/Greedy/Bayesian

- **Model & HP search in Recommender**
  - **Model search:** hyper interaction
  - Types of interaction and ways to stack
  - **HP search:** interactor-specific
  - E.g., units, layers, dropout for MLP
Road Map

1. AutoML System for Deep Learning

AutoKeras

2. AutoML System for Recommendation

Item_ID #23
Harry Potter and the Chamber of Secrets

Author
J.K. Rowling

User_ID #76

Item_ID #2
Harry Potter and the Sorcerer’s Stone

3. AutoML System for Outlier Detection

Reconstruction based
Density based
One-class based
Cluster based
Why Automated Outlier Detection

Existing NAS methods are unsuitable for automating OD because they ignore the properties of outliers and the data is imbalanced.

Our Goal — Consider outlier def. in search and data distribution in training

- **Search for definition-hypothesis**, which decides objective function
- **Explore untapped search space** and exploit useful past sample
Conclusions

1. AutoML System for Deep Learning
   - *Bayesian optimization* speeds up the *NAS loop* by reducing the number of training iterations, while *network morphism* speeds up *evaluation* by using the weights of trained architecture.

2. AutoML System for Recommendation
   - *Abstract virtual blocks* create flexible *search space* to accommodate different recommendation scenarios (inputs, interactions, and objectives).
   - *Multi-objective evolutionary search algorithm* considers architecture *evaluation* and *computational complexity* and produced new SOTA.

3. AutoML System for Outlier Detection
   - *Curiosity-guided search strategy* addresses local optimality and *self-imitation learning* improves sample efficiency.
Future Directions

1. **AutoML Infrastructure**
   - Modules of AutoML systems are complicated. How can we have a unified and well-organized infrastructure for different AutoML systems? A potential solution is DARPA D3M infrastructure.

2. **Explainability**
   - Enabling explainability would be important for data scientists to easily make use of the outputted models by AutoML systems.

3. **Parallel Computation**
   - Further improve data/model/search efficiency in AutoML.
Acknowledgements

❖ DATA Lab and Collaborators

Data Analytics at Texas A&M (DATA Lab)

❖ Funding Agencies

--- Defense Advanced Research Projects Agency (DARPA)

--- National Science Foundation (NSF)

--- Industrial Sponsors (Samsung, Adobe, Apple, LinkedIn, etc.)

❖ Everyone attending the talk!