Today: AutoML

Logistics

(Tuesday: Whiteboard via iPad)

Thursday: Discussion Sessions (Presenters will lead discussion + Theo)

Today: AutoML

Last class: Automated Feature Selection and how to frame this problem as search over the set of possible features.

How to search over the space of possible models.

Deep Learning:

- Vision (CNNs)
- Audio analytics
- Text (RNNs, Transformers)
- Images (CNNs)
- Tabular data.

Problem

The performance of DL: depends on

- Units in each layer/# layers
- Hyperparameters

- Learning rate, cross-validation parameters

Two types of HP (hyperparameters)

- Architectural HP
- Algorithmic HP (optimization)

Expansion of parameters

SD design decisions

Hyperparameter optimization evaluation costs
Current solution

Data → decision on hyper-parameters
   → user design a model.
   → trained model/eval.

Promise of AutoML

Data. → iterate to maximize performance.
   Meta-level learning/optimization
   Learning box
   → not just a trained model (learning pipeline)
   → never ending learning
   7 define this box

Learning box
\{ data cleaning \}
\{ "pre-proc" \}
\{ feature selection \}
\{ training algo selection \}
\{ model selection \}
\{ etc. \}

Formal Problem statement

AutoML: it is a hyper-parameter optim problem.

Grid search: eliminate parts of the grid

I: hyperparameters of a ML algorithm A has a domain \( \Delta \) (valid values that HR I can take)
Find $A^*$ s.t. some utility is maximized

Utility function for ML: minimize our generalization

$L(A_2, D_{train}, D_{valid}) \to \min_{A_2} \text{loss}$

loss of $A_2$ using HP, $A_2$

trained on $D_{train}$ and evaluated on $D_{valid}$ (simulating generalization error)

$HPO: A^* = \arg \min_A L(A_2, D_{train}, D_{valid})$

Q: What is this domain $\Lambda$? What kind of variable types do we have?

$LR$: continuous variable

# of h.units: discrete variable

ReLU or sigmoid: categorical (binary) variable (finite domain)

What optimizer to use: ADAM or SGD?

Some of the $A$s are "unlocked" depending on specific configuration for other parameter $A$s.

W. Adam $\rightarrow$ momentum

SGD

Choose the ML model $\leftarrow$ SVM (kernel, conditional HPs)
Instead of a single Algorithm $A \to A$
I have access to a set of As
$A = \{ A^{(1)}, \ldots, A^{(n)} \}$
$A^{(i)}$ the HP space of $A^{(i)} \quad \forall i=1\cdots n$

$L(A^{(i)}, D_{\text{train}}, D_{\text{valid}})$

$A^* = \arg\min_{A^{(i)}} L(A^{(i)}, D_{\text{train}}, D_{\text{valid}})$

$A^{(i)} \in A^{(i)} \quad \forall A^{(i)} \in A$

---

Analyze function $L$ (costs)

$\left( \sum_{A^{(i)}} L(A^{(i)}) \right) \left( \sum_{z \in A^{(i)}} z \in A^{(i)} \right)$

Train over all $A^{(i)}$ (size of search space I have to consider)

$A$ is a NN $\sim |A^{(i)}| = 50$
Search is expensive due to exponential explosion.

→ Evaluation at a single point?

Operations during Eval: \( A^{(i)} \) using \( A^{(i)} \) on \( D_{train} \) to evaluate on \( D_{valid} \)

Depending on the model

\( |D_{train}|, |D_{valid}| \rightarrow \) expensive op

How to solve this opt problem

Grid search

Blackbox opt

Grid points 12 configs

→ parallel

Learning rate (cont)
Blackbox opt Case 2:  \[
\text{Fix } A \text{ } \sim \text{ N}
\]

\[
\arg\max_L (A_2, D_{\text{train}}, D_{\text{valid}})
\]

\#layer

Neighborhood

lr

\[
D_{\text{train}} \text{ lr} 0.13
\]

Explores this space to find a \((\#\text{layers, lr})\) config that maximizes my utility.

Random search considers only very recent information.
Type 3 of BlackBox opt

Bayesian optimization

High-level

Fit a probabilistic model to your function evaluations:

\[
\mathcal{D} \left< f(x), f(x) \right>
\]

\( f \): prob. distribution

\( f \): parameter values \( \theta_1, \theta_2, \ldots \)

\[
\frac{1}{D} \sum_{i=1}^{D} f(\theta_i), f(\theta_2), \ldots f(\theta_D)
\]
I have a point \( \mathbf{f}(x) \). \( \mathbf{f}(x) \to \) hidden random variable.

\[
P(\mathbf{f}(x) | \mathbf{f}(x_1), \mathbf{f}(x_2), \ldots, \mathbf{f}(x_d))
\]

use this to estimate utility \( \mathbf{f} \) on unknown points.

given a config param \( \mathbf{f} \)

instead of running my model to find \( \mathbf{f}(x) \)

\[
\begin{align*}
\text{#layer} & \quad \text{prob distr.} \\
2 & \quad \text{gradient step} \\
1 & \quad \text{gradient step}
\end{align*}
\]

\[
0.43 \quad \mathbf{f}(x_1) \quad 0.62 \quad (2 \text{ optimization problems to find the best performing model})
\]

\[
\sum (\text{combined})
\]
\[
\hat{a} = \arg\max P(f(a) | t(x)) \\
\text{depending on Gaussian distribution}
\]

Utility

very expensive evaluations

starting point

exp. evaluations with cheap inference

\Rightarrow \text{give me one good point to evaluate on}

Bayesian opt.: exchange