Learning What Data to Learn

Yang Fan et al.

Recent Trends in Automated Machine Learning

Muhammad Shahbal Khan
Introduction

- SGD is used to train Neural Networks.
- Often multiple epochs are required to achieve an effective model.
- Traversing more data takes more effort.
Introduction

- Would an intelligent data selection strategy be useful?
Introduction

- Would an intelligent data selection strategy be useful?
Previous Works

- **Hardness** as a Heuristic
Previous Works

- **Hardness as a Heuristic**
  - Curriculum Learning (Bengio et al.)
    - Based on heuristic understandings of the data.
Previous Works

- **Hardness as a Heuristic**
  - Curriculum Learning (Bengio et al.)
    - Based on heuristic understandings of the data.

- Self Paced Learning (Kumar et al.)
  - Uses training loss as selection strategy.
Previous Works

- **Hardness** as a **Heuristic**
  - Curriculum Learning (Bengio et al.)
    - Based on heuristic understandings of the data.

- Self Paced Learning (Kumar et al.)
  - Uses training loss as selection strategy.

- Not Dynamic...
Contribution

- Neural Data Filter (NDF)
  - Based on Deep Reinforcement Learning, (REINFORCE)

\[ \Theta \leftarrow \Theta + \alpha v_t \sum_m \frac{\partial \log P_\Theta(a_m | s_m)}{\partial \Theta} \]

- A binary classification algorithm as a policy function.
Neural Data Filter

NDF acts as a teacher to the base SGD model
NDF State Representation

The state representation includes:
NDF State Representation

The state representation includes:

- **Data features:**
  - label categories (one-hot encodings), sentence length, semantic features, histograms etc.
NDF State Representation

The state representation includes:

- **Data features:**
  - label categories(one-hot encodings), sentence length, semantic features, histograms etc.

- **Base model features:**
  - mini-batch number/iteration number, average historical training loss, validation accuracy etc.
The state representation includes:

- **Data features:**
  - label categories (one-hot encodings), sentence length, semantic features, histograms etc.

- **Base model features:**
  - mini-batch number/iteration number, average historical training loss, validation accuracy etc.

- **Combined data and model features:**
  - predicted class probabilities, loss on current mini-batch, margin between predicted and actual values etc.
Training process

Algorithm 1 SGD Training with Neural Data Filter.

Input: Training data $D$.
1. Randomly sample a subset of NDF training data $D'$ from $D$.
2. Optimize NDF policy network $A(s; \Theta)$ based on $D'$ by policy gradient (details in Algorithm 2).
3. Apply $A(s; \Theta)$ to full dataset $D$ to train the base machine learning model by SGD.

Output: The base machine learning model.
Learning the NDF Policy

**Algorithm 1** Train NDF policy.

**Input:** Training data $D'$. Episodes $L$. Mini-batch size $M$. Discount factor $\gamma \in [0, 1]$.

Randomly split $D'$ into two disjoint subsets: $D'_{\text{train}}$ and $D'_{\text{dev}}$.

Initialize NDF data filtration policy $A(s, a; \Theta)$, i.e., $P_{\theta}(a|s)$.

for each episode $l = 1, 2, \cdots, L$ do

Initialize the base machine learning model.

Shuffle $D'_{\text{train}}$ to get the mini-batches sequence $\{D_1, D_2, \cdots\}$.

$T = 0$.

while stopping criteria is not met do

$T = T + 1$.

Sample data filtration action for each data instance in $D_T = \{d_1, \cdots, d_M\}$: $a = \{a_m\}_{m=1}^M$, $a_m \propto P_{\theta}(a|s_m)$.

Update base model by SGD based on the selected data in $D_T$.

Receive reward $r_T$ computed on $D'_{\text{dev}}$.

end while

for $t = 1, \cdots, T$ do

Compute cumulative reward $v_t = r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t} r_T$.

$$\Theta \leftarrow \Theta + \alpha v_t \sum_m \frac{\partial \log P_{\theta}(a_m|s_m)}{\partial \Theta}$$ (1)

end for

end for

**Output:** The NDF policy $A(s, a; \Theta)$. 

---

Muhammad Shahbal Khan | Recent Trends in Automated Machine Learning
Learning the NDF Policy

- Subtract a baseline from reward to reduce estimation variance.

\[ \Theta \leftarrow \Theta + \alpha (r_t - b_t) \sum_m \frac{\partial \log P_\Theta(a|s_m)}{\partial \Theta} . \]
Learning the NDF Policy

- Subtract a baseline from reward to reduce estimation variance.

\[ \Theta \leftarrow \Theta + \alpha (r_t - b_t) \sum_m \frac{\partial \log P_\Theta(a|s_m)}{\partial \Theta}. \]

- The baseline is a running average of previous rewards.
Learning the NDF Policy

- Subtract a baseline from reward to reduce estimation variance.

\[ \Theta \leftarrow \Theta + \alpha (r_t - b_t) \sum_m \frac{\partial \log P_\Theta(a|s_m)}{\partial \Theta} \]

- The baseline is a running average of previous rewards.
- The reward is validation accuracy of the base model.
Experiments Setup

- Performance comparisons conducted with 3 techniques
  - Self Paced Learning
  - RandDrop
  - Unfiltered SGD
Experiments Setup

- Performance comparisons conducted with 3 techniques
  - Self Paced Learning
  - RandDrop
  - Unfiltered SGD

- NDF a 3 layer NN.
Experiments Setup

- Performance comparisons conducted with 3 techniques
  - Self Paced Learning
  - RandDrop
  - Unfiltered SGD

- NDF a 3 layer NN.
- Models only updated after M samples are accumulated.
  - Ensures that convergence speed is only affected by data quality.
Experiment 1: MLP for MNIST

A 3-layer FF NN with tanh activation and cross-entropy loss.
Experiment 2: CNN for CIFAR-10

ResNet with Momentum-SGD.
Exp 3: RNN for IMDB sentiment classification

An LSTM fed into Logistic Regression for sentiment classification with Adadelta optimizer.
Discussion

- A good data selection mechanism can accelerate model convergence.
Discussion

- A good data selection mechanism can accelerate model convergence.
- Data selection policy can be case dependent, it needs to adapt to the data and model.
Discussion

- A good data selection mechanism can accelerate model convergence.
- Data selection policy can be case dependent, NDF adapts to the data and model.
- Compared to heuristic techniques NDF performs better.
Discussions

- Can the NDF policy learning be improved?
Discussions

- Can the NDF policy learning be improved?
  - ‘Proximal Policy Optimization’ (Schulman et al.),
  - ‘Actor-Critic’ (Konda et al.)
  - ‘Q-Learning’ (Watkins et al.).
Questions?