Machine Learning for Networking: Workflow, Advances and Opportunities

Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang

ABSTRACT

Recently, machine learning has been used in every possible field to leverage its amazing power. For a long time, the networking and distributed computing system is the key infrastructure to provide efficient computational resources for machine learning. Networking itself can also benefit from this promising technology. This article focuses on the application of MLN, which can not only help solve the intractable old network questions but also stimulate new network applications. In this article, we summarize the basic workflow to explain how to apply machine learning technology in the networking domain. Then we provide a selective survey of the latest representative advances with explanations of their design principles and benefits. These advances are divided into several network design objectives and the detailed information of how to solve them is also shared.

In many scenarios, MLN algorithms are used to perform classification or regression tasks from labeled data, while USL algorithms focus on classifying the sample sets into different groups (i.e., clusters) with unlabeled data. In RL algorithms, agents learn to find the best action series to maximize the cumulated reward (i.e., objective function) by interacting with the environment. The latest breakthroughs, including deep learning (DL), transfer learning and generative adversarial networks (GAN), also provide potential research and application directions in an unimaginable fashion.

Dealing with complex problems is one of the most important advantages of machine learning. For some tasks requiring classification, regression and decision making, machine learning may perform close to or even better than human beings. Some examples are facial recognition and game artificial intelligence. Since the network field often sees complex problems that demand efficient solutions, it is promising to bring machine learning into the networking domain.
## Overview

**Workflow:**
- Steps to use ML in networking
- From problem formulation to deployment

**Survey:**
- Summarize latest works
  - Design principles
  - Benefits

**Opportunities:**
- Solve intractable old questions
- Stimulate new applications
Developing algorithms and systems to deal with complex problems is a challenging task.

Networks are hard to manage, optimize, and secure, and not necessarily improving.

ML is good in that...
Recently ML techniques have made breakthroughs in:

bioinformatics
speech recognition
computer vision
Recently ML techniques have made breakthroughs in:

bioinformatics

speech recognition

computer vision
Recently ML techniques have made breakthroughs in:

- bioinformatics
- speech recognition
- computer vision
Recently ML techniques have made breakthroughs in:

bioinformatics

speech recognition

computer vision
Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player
Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov

- **2015** AlphaGo beats the first professional Go player

- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match
Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov

- **2015** AlphaGo beats the first professional Go player

- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match

- **2017** AlphaMaster beats Ke Jie, the Go world champion
Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player
- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match
- **2017** AlphaMaster beats Ke Jie, the Go world champion
- **2018** AlphaZero crashes AlphaMaster 100-0 (self-play)
Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player
- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match
- **2017** AlphaMaster beats Ke Jie, the Go world champion
- **2018** AlphaZero crashes AlphaMaster 100-0 (self-play)
- **2018** AlphaZero in Chess, Go, Shogi
Reaching super-human skills

- **1997** IBM deep blue beats the chess world champion Kasparov
- **2015** AlphaGo beats the first professional Go player
- **2016** AlphaGo beats 9-dan Lee Sedol in a five-game match
- **2017** AlphaMaster beats Ke Jie, the Go world champion
- **2018** AlphaZero crashes AlphaMaster 100-0 (self-play)
- **2018** AlphaZero in Chess, Go, Shogi
- DeepMind stops AlphaZero, concluding that it beats at any “perfect information” game
StarCraft is a deep, complicated war strategy game. Google’s AlphaStar AI crushed it.

DeepMind has conquered chess and Go and moved on to complex real-time games. Now it’s beating pro gamers 10-1.

By Kelsey Piper | Updated Jan 24, 2019, 7:04pm EST
OpenAI Bot Crushes Dota 2 Champions And This is Just the Beginning

Machines, like humans, learn best when they’re beaten
“AI is capable of vastly more than anyone on earth can even imagine
And its rate of improvement is exponential…”

— Elon Musk, CEO OpenAI
NIPS Attendance

Source: @ML_Hipster
AI papers on arxiv

Source: Andrej Karpathy
AlexNet to AlphaGo Zero: A 300,000x Increase in Compute

Source: OpenAI
Algorithmic innovation, much more data, and **funding**

Source: OpenAI
## Machine learning refresher

ML aims to construct algorithms and models that can learn to make decisions directly from data.

<table>
<thead>
<tr>
<th>Supervised</th>
<th>Classification or regression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Labeled data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unsupervised</th>
<th>Clustering samples into groups</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Unlabeled data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reinforcement</th>
<th>Agent learns the best action series</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- To maximize a cumulative reward</td>
</tr>
<tr>
<td></td>
<td>- Interacting with the environment</td>
</tr>
</tbody>
</table>
Machine learning refresher

ML aims to construct algorithms and models that can learn to make decisions directly from data.

**supervised**
- classification or regression
  - labeled data

**unsupervised**
- clustering samples into groups
  - unlabeled data

**reinforcement**
- agent learns the best action series
  - to maximize a cumulative reward
  - interacting with the environment
Supervised learning

Housing price prediction

price in 1000s CHF

Examples from Machine Learning, Andrew Ng
Supervised learning

Housing price prediction

How much can I get for a 750 feet$^2$ house?
Supervised learning

Housing price prediction

Price in 1000s CHF

Size in feet$^2$
Supervised learning

Housing price prediction

Learning algorithm 1 (linear function)

price in 1000s CHF

size in feet²
Supervised learning

Housing price prediction

Learning algorithm 2 (quadratic function)
Supervised learning

Housing price prediction

supervised

‘right answers’ (labels) given for the training dataset

regression

predict continuous valued output (price)
Supervised learning

Breast cancer (malignant, benign)
Supervised learning

Breast cancer (malignant, benign)

Will this tumor be malignant or benign?
Supervised learning

Breast cancer (malignant, benign)
Supervised learning

Breast cancer (malignant, benign)

Will this tumor be malignant or benign?
Supervised learning

Breast cancer (malignant, benign)

Learning algorithm 1 (linear function)
Supervised learning

Breast cancer (malignant, benign)

- Supervised: ‘right answers’ (labels) given for the training dataset
- Classification: the output is a discrete value
  - large number of features (SVM)
Machine learning refresher

ML aims to construct algorithms and models that can learn to make decisions directly from data.

<table>
<thead>
<tr>
<th>Supervised</th>
<th>Classification or regression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>labled data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unsupervised</th>
<th>Clustering samples into groups</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>unlabeled data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reinforcement</th>
<th>Agent learns the best action series</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>to maximize a cumulative reward</td>
</tr>
<tr>
<td></td>
<td>interacting with the environment</td>
</tr>
</tbody>
</table>
Unsupervised learning

supervised learning

unsupervised learning

‘right answers’ (labels) given for the training dataset

data is not labeled
Unsupervised learning
Unsupervised learning

Here is a dataset. Can you find some structure?
Unsupervised learning

Here is a dataset. Can you find some structure?
Machine learning refresher

ML aims to construct algorithms and models that can learn to make decisions directly from data.

- **supervised** classification or regression
  - labeled data

- **unsupervised** clustering samples into groups
  - unlabeled data

- **reinforcement** agent learns the best action series to maximize a cumulative reward
  - interacting with the environment
Reinforcement learning (Agent and Environment)

At each step $t$, the agent:
- observes state $s_t$
- receives reward $r_t$
- executes action $a_t$

The environment:
- receives action $a_t$
- emits state $s_t$
- emits reward $r_t$

RL in a nutshell:
- the agent selects actions to maximize future reward

Example from DRL, David Silver (Google DeepMind)
Why is ML a good match?

<table>
<thead>
<tr>
<th>ML is good in:</th>
<th>Which is useful for:</th>
</tr>
</thead>
<tbody>
<tr>
<td>classification and prediction</td>
<td>intrusion detection</td>
</tr>
<tr>
<td>decision making</td>
<td>performance prediction</td>
</tr>
<tr>
<td>interaction with complex</td>
<td>network scheduling</td>
</tr>
<tr>
<td>environments</td>
<td>parameter adaptation</td>
</tr>
<tr>
<td></td>
<td>load changing patterns</td>
</tr>
<tr>
<td></td>
<td>network state</td>
</tr>
</tbody>
</table>
Workflow

Step 1: Problem formulation (prediction, regression, clustering, decision making)

Step 2: Data collection (e.g., traffic traces, performance logs, etc.)

Step 3: Data analysis (preprocessing, feature extraction)

Step 4: Model construction (offline training and tuning)

Step 5: Model validation (cross validation, error analysis)

Step 6: Deployment and inference (tradeoff on speed, memory, stability and accuracy of inference)

Meet requirements? Yes → Step 6

No → Step 5
Step 1. Problem formulation

- formulate and abstract the problem
  - classification problem
  - clustering problem
  - decision making

- define and classify
to help us decide:
  - the best learning model
  - type of data required
  - amount of data required

Training is time consuming, so better make a good decision.
Step 2. Data collection

Gather a large amount of representative network data without bias

<table>
<thead>
<tr>
<th>purpose</th>
<th>training and evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>traffic traces, logs</td>
</tr>
<tr>
<td></td>
<td>network, application level</td>
</tr>
<tr>
<td>support</td>
<td>labels, feature extraction</td>
</tr>
</tbody>
</table>
Step 3. Data analysis

Pre-process and clean raw data

feature engineering which parameters impact the most on the target performance

Example from pForest, Coralie Busse-Grawitz et. al.
Step 4. Model construction and validation

- Model selection
  - problem category
  - size of the dataset
  - characteristics of scenario

- Tune parameters
  - training and tuning

- Adapt model
  - lack of theoretical guidelines
Step 4. Model construction and validation

- Support vector machines
- Random forest
- Naive Bayes
- Logistic regression
- Shallow neural networks
- Hidden Markov model
- K-nearest neighbour
- Convolutional feed-forward deep neural networks
- Recurrent deep neural networks
- Fully-connected feed-forward deep neural networks
- Deep belief networks
- Stacked auto-encoders
- Deep belief networks
- Deep neural networks
- Recurrent deep neural networks
- Deep neural networks
- Clustering
- Association
Step 5. Model validation and deployment

model validation
- test accuracy (over/under fitting)
  - increase data volume
  - reduce model complexity
- analyze wrong samples to find errors
- iterate multiple times!

deployment
- check resource constraints
- accuracy vs. overhead
- consider fault tolerance
Example 1. Next–generation firewall

We want to build an ML–based system to detect and mitigate DDoS attacks
**Example 2. Next-generation routing**

We want to build an ML-based system to predict failures and reroute traffic proactively.