ARCHITECTING AI APPLICATIONS

O’REILLY AI CONF
LONDON, 2018

DR. MIKIO BRAUN
@mikiobraun
AI ARCHITECT
ZALANDO SE

11-OCT-2018
ARTIFICIAL INTELLIGENCE

- Symbolic AI
- Rule-Based Systems
- Expert Systems
- Perceptron
- Artificial Neural Networks
- Statistics
- Machine Learning
- Shallow Models
AI TIMELINE PAST 20 YEARS (that’s 1998 till 2018)

1997: Chess
2011: Jeopardy
2016: Go
2017: Image Recognition
...
Initially approached AI like any other problem computers could solve.

Alternatively, using an approach inspired by human biology.

Machine Learning added a statistical approach to the mix.

Recently, Deep Learning has led to impressive improvements.
• Classical approach is to specify what the input/output relation is, then devise programs to solve that.

• Machine Learning replaces that with examples (+ a cost function).

• Training then means to infer a model that generalizes well on future data.
• Artificial Intelligence is the overarching goal or challenge.
• Machine Learning is one approach that has proven very successful if the problem itself cannot be specified easily.
Many reasons why: convenience, security, disrupting mobility.

Current approaches are a mix of many systems, some of which make heavy use of machine learning.

Deep Learning very successful for computer vision and image analysis.
Autonomous driving is based on a mix of sensors with quite different capabilities to improve reliability.

- **Sonar/Radar**: Cheap, low resolution, works well under extreme weather and in darkness, can estimate velocity.
- **Camera**: Cheap, very high resolution, similar to what we humans see.
- **Lidar (light detection and ranging)**: expensive, very accurate depth maps.
Companies like Waymo do extensive data collection and simulation to evaluate and tune the system.

Not just for training ML methods, also for overall systems testing.

ML inspired approach to defining the problem, but mix of ML and explicit solutions.

(Waymo lecture at MIT)
Chatbots

Booking a flight:

Frame:
- when: Date
- start, end: Location
- how many persons: Number

Example 1:
A: “I’d like to book a flight tomorrow”
B: “From where to where do you want to fly?”
A: “From London to Berlin.”
B: “With how many passengers?”
A: “Just me.”
B: “Okay, so I have one passenger for a flight from London to Berlin tomorrow. Is that correct?”
A: “Yes.”
B: …

Example 2:
A: “I’d like to book a flight for me tomorrow from London to Berlin.”
B: …

- Not that much ML in there (at least right now).
- Dialog is done through Frames that capture a piece of information required and an analysis part that maps user input to fields.
- ML used for understanding speech2text, named entity recognition, analysis
• Machine Learning used especially on “perception” part.

• Core is rule based system.

• Potential to improve those based on examples, too, same for text2speech.
Recommendations as an AI problem:

- Understand what the user is looking for right now. What is his intent, what is in his mind?
- Technically, predict next action.
- Quite involved, dealing with real-time data, etc.
From computer science’s point of view, strategy games are “easy” if you know the value of each state.

Cleverly simulating “plausible actions” leads to speedup (Monte Carlo tree search).
ALPHA GO: CONVOLUTIONAL NEURAL NETWORKS FOR POLICY AND VALUE PREDICTION

Training
1. policy from expert
2. policy against itself
3. value

Play:
Simulate with policy, evaluate with V

Board positions as image
COMMON PIECES IN AI APPLICATIONS

- Some form of data collection.
- “Perception”, turning data into features.
- Use of ML to generalize beyond concrete training data.
- Strong focus on using data to define problems & evaluate.
- Combine ML components with rule based, classical control & optimization components.
- Deliver predictions from learned models in production.
ARCHITECTING AI APPLICATIONS
Core Machine Learning
—how to train, evaluate, etc.

Serving
—access predictions in real-time

Data Preprocessing and Features
—how to deal with preprocessing

Automation & Monitoring
—making it more production ready

Machine Learning Integration
—how to fit it into a larger picture
Core Patterns:

- Preprocess data to extract features, transform to fit to ML model.
- Training/test split for model selection and evaluation.

Leaving out how to do this part effectively (notebooks, moving from exploration to construction, etc.)
SERVING PATTERNS

• How to provide the predictions of the machine learning model to the application.
• If the domain is small, precomputing predictions might be the easiest way to serve.
• If you have processing on the data that is adjusted to the data, these become part of your model, too, and their state need to be saved.

• If you process the outputs, you need to have a **reverse** transformation in prediction.
• If predictions are precomputed, no special treatment of feature computation is necessary.

• If it is served online, you can either precompute features in a Feature Store. But you don’t have updates for new data.

• Or you can use the Feature Store and update features online as well.
• Automation is required to adjust models to changes in the data.
• Everything needs to be automated: data gathering, feature extraction, preprocessing, training, etc.
• Easy: cron jobs, AWS Scheduled Event and Lambda
• Eventually you want:
  • Fine grained jobs
  • Dependency between jobs
  • Monitoring and automatic restart, etc.
• Project like Airflow are very popular. Netflix is one of the leader in automating everything.
Monitoring is essential for production.

Technical monitoring checks for latencies, error rates, general technical health.

Functional monitoring compares core statistics for your algorithms.

If performance cannot be readily observed, you can monitor data statistics to watch for changes.

(L. Weichbrodt, Measuring Operational Quality of Recommendations—Quality Oriented Service Level Objectives)

(Continuous Live Monitoring of Machine Learning Models with Delayed Label Feedback)
• Depending on how ML is used, there are different integration patterns.

• Keep in mind that the ML problem and the application problem are different and have different metrics.

• Beware of interacting ML models if the overall training data is used.
• Modern AI Applications are a mix of ML and programmed components.
• It takes considerable effort to “productionize” ML applications.
• Recurring design patterns are evolving.
• Still a lot of room for new products & open source projects.
MIKIO BRAUN
INFORMATION
AI ARCHITECT

mikio.braun@zalando.de

11-10-2018
This presentation and its contents are strictly confidential. It may not, in whole or in part, be reproduced, redistributed, published or passed on to any other person by the recipient.

The information in this presentation has not been independently verified. No representation or warranty, express or implied, is made as to the accuracy or completeness of the presentation and the information contained herein and no reliance should be placed on such information. No responsibility is accepted for any liability for any loss howsoever arising, directly or indirectly, from this presentation or its contents.