The Rat Race to Deploy AI Models: The Problem of Being on Top of the Gartner Hype Cycle

Varshanth R Rao

CS846 Course Project
Agenda

1. Introduction to SDLC
2. GHC and Effects on SDLC
3. Basic Concepts
4. Case Study
5. Post Mortem/Take Aways
6. Conclusion

10 Phases of SDLC:
The Rosy Picture

1. Initiation: Identification of Opportunity
2. System Concept Development: Scope & Boundaries of Concept
3. Planning to Acquire Resources
4. Requirements Analysis
5. Design: The How Part
6. Development: Design->System
7. Integration & Testing
8. Implementation: Lab->Production
9. Operation & Maintenance
10. Disposal : End of Life
Gartner Hype Cycle

Gartner Hype Cycle 2018

Source: https://www.fourquadrant.com/gartner-hype-cycles-magic-quadrants/
Peak of GHC & Effect on SDLC

Why Innovate at the Peak??

1) Spearhead Innovation
2) Boost Competitive Edge
3) Capture Market Share Early

Effect of Investment At The Peak:

1) Surge of Pressure Top (Mgmt) to Down (Engg)
2) Increases risk of breaking safe SDLC
10 Phases of SDLC:

The Effect of Investing at The Peak of Inflated Expectations

1. Initiation: Identification of Opportunity
2. System Concept Development: Scope & Boundaries
3. Planning to Acquire A LOT of Resources
4. Requirements Analysis
5. Design: The How Part
6. Development: Design->System
7. Integration & Testing
8. Implementation: Lab->Production
9. Operation & Maintenance
10. Disposal: End of Life

Maximum Focus. This is where we need the work done..... Fast..

What is going on here?
Who wrote this code & why?
Why is this not working !!!!

Maximum Focus? What Boundaries? Everything is Perfect

To bring in $$ we invest $$$$$$$

We can do it !! Even though we don’t understand it completely...

Maximum Focus.

This is where we need the work done ..... Fast..

What is going on here?
Who wrote this code & why?
Why is this not working !!!!
Where is AI?

- In last 20y, academic papers increased by 9x
- AI startups increased from 2000 by 14x
- Annual investment in AI has increased by 4x since 2013

Source:
Basic Concepts

Object Detection:
- Task of localizing and detecting objects
- Traditional methods overtaken by AI

Face Detection:
- Subset of Object Detection: Localize & Detect Faces

Source:
https://paperswithcode.com/task/face-detection/latest
**Basic Concepts**

**Object Detection:**
Different AI solutions come with tradeoffs:

<table>
<thead>
<tr>
<th></th>
<th>a) Faster RCNN:</th>
<th>b) YOLO v3:</th>
<th>c) SSD:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow</td>
<td>Slow -&gt; 2 Stage Network</td>
<td>Fast</td>
<td>Fast</td>
</tr>
<tr>
<td>Accurate &amp; Reliable</td>
<td>- Sacrifices accuracy for speed</td>
<td>- Versatile as network is modularized</td>
<td>- Versatile as network is modularized</td>
</tr>
<tr>
<td></td>
<td>- Not versatile as network is intangible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Basic Concepts

Evaluating Object Detection:

mean Average Precision @ IoU Threshold

- Area under the Precision-Recall Curve averaging over all classes and/or IoU values

\[ \text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \]
Basic Concepts

Common AI Development Libraries:

- TensorFlow
  - Open Sourced by Google
  - Steep Learning Curve
  - Large community support

- Caffe
  - Open Sourced by UCB
  - Closer to device
  - Framework - Not a Dev Platform
  - Large community support

- PyTorch
  - Open Sourced by FB
  - Easy to learn/Fast Prototyping
  - Models can be converted to Caffe2
  - Large community support
Case Study:

Deploy Face Detection on Smartphone

Supplier Co-MP -> Client Cu-MP
Requirements Discussion with Cu-MP

1. Should work on Cu-MP smartphones with specified hardware
2. Should not increase camera application’s memory beyond 768MB
3. OpenCV source code provided but modification through review process
4. Will provide annotated dataset for training
5. Will provide testing API to test on Cu-MP hidden test set (Max 24 submissions a day)
6. Should achieve mAP@0.5 of 70% at camera rate of 30FPS
7. Will provide 3 target smartphones for testing
Co-MP Implementation Plan

1. Engg team 2 members to work on optimizing OpenCV code

2. 2 Researchers allocated to find most stable and reliable deep learning algorithm

3. 4 Researchers allocated for implementation in PyTorch (favored by recent researchers) & Tensorflow (favored by senior researchers) (2 each) due to lack of clearness of which platform is better.

4. PyTorch & TF code can be converted to Caffe, so 2 Engineers allocated to optimize Caffe codebase
Progress Checkpoint

1. MobileNet v2 SSD Lite was selected. Used a more recent operation called Depthwise Separable Convolution used for model compression
2. Both (PyTorch & TF) teams achieved mAP of 76% at 25FPS
3. Major optimizations committed to OpenCV source code
4. Caffe source code optimizations in progress
Disaster Week Prior to Customer Demo

1. During integration camera application kept crashing ->
   Group convolution operation for depthwise separable convolution not
   implemented by Cu-MP compiler team although the SoC support present ->
   MobileNet v2 SSD Lite cannot be used

2. Plan B -> Implement SqueezeNet SSD in PyTorch (rapid prototyping)

3. 2 out of 4 researchers not skilled at PyTorch, hence were given minor
   tasks and mandatory participation in code reviews to ramp up quickly

4. Implementation occupied 830MB (62MB greater than reqm) but
   achieved mAP @ 0.5 of 65% at 23FPS
Salt on the Wound: Customer Demo

1. Cu-MP announces it recently added 10K more images to hidden test set
   - Would give Co-MP 4 week extension
2. SqueezeNet SSD achieves meagre 55% mAP @ 0.5
3. Cu-MP presses that agreed reqm are strict
4. Cu-MP agrees to attempt to deliver the group convolution implementation but without guarantee
5. Cu-MP iterates deadlines are strict & does not heed to Co-MP extension requests
Firefight: Triage & Diagnosis

1. mAP drop due to lack of detection of small faces due to
   a) SSD architecture drawback
   b) Lack of small faces data points in training set

2. Engineering team interfaces with compiler team for group convolution operation

3. 2 researchers put in charge of performing architecture modifications for Squeezenet SSD. 2 researchers for Mobilenet v2 SSD Lite in case group convolution operation is successful

4. 2 engineering members assigned to collect & annotate small face & low light data (which was found to also be in small numbers)
Sigh of Relief: Final Product Delivery

1) Engg <-> Compiler team successful. Group convolution implemented

2) The customized Mobilenet v2 SSDLite achieved 72% map @ 0.5 at 30FPS

3) Success through architecture modifications due to research efforts, data augmentation & optimizations by engineering team
Requirements Analysis: The Blank Cheque Problem

1) Not well researched technologies cannot be holistically evaluated through academic metrics
   e.g) Low light photography samples

2) Assumptions from successes of similar experiments cannot be extrapolated to other experiments
   e.g) mAP @ 0.5 of 70% blindly agreed even though mAP is a co-property of the dataset!

3) Requirements analysis must be conducted in the presence of senior research scientists with prior domain experience

4) Requirements analysis cannot be treated like a Blank Cheque!
Bias Towards Productionizing State of The Art

1) R & D team: Population(Researchers) > Population(Engineers)

2) Media hype focus on SOTA -> Translates to Expectations of Product

3) Leads to pushing SOTA into production without understanding limitations & environment requirements of the tech

   e.g) Using MobileNet v2 SSDLite (SOTA for lightweight object detection) without confirming if “Group Convolution” is implemented
Lack of Data First Mentality

1) Problem? Think of solution -> No!
   e.g) Small faces & low light images absent -> Last minute find. High pressure.

2) Noisy data (Garbage) In -> Garbage Out

3) Data driven testing should be in parallel with solution scoping.

4) Encourages rail guarding of development & coverage oriented testing

Would **HIGHLY** recommend reading:
Divide & Conquer: A Double Edged Sword

1) Software Project Management Process is different for software dev vs R&D projects i.e. Research projects are open ended & more flexible.

2) Requirement for effective task division ->
Delicate balance to handle cross-talent knowledge flow + task completion

e.g) Bifurcation into TF & PyTorch teams -> Redundant work, Talent Waste
Could have instead worked with toy examples to compare performance

Would HIGHLY recommend reading:

Bridging Knowledge Gaps

1) High Reward & Low Risk Investment

2) Researchers are hired for their unique skill & hence R & D teams have silo-like skillsets (highly specialized)

3) Research -> Product involves convergence of highly skilled researchers with diverse experience & high variance of egos

4) Becomes critical to build ecosystem to bridge knowledge gaps:
   a) Formal training  b) KT Sessions  c) Pair Programming

   e.g) PyTorch/TF training sessions given to researchers for cross collaboration
Inducing Explicit Accountability Through Implicit Explainability

1) New Tech. Limitations & Strengths not fully known

2) Question everything-> Why is it working? Why is it not working? Has the entire solution space been covered? Under what conditions is it or is it not working?

3) Accountability can clarify liability, minimize risks & increase trust

4) Implicit Explainability through plug and play explainable modules like decision heat maps. Eliminate black box nature.

   e.g) Early detection of small face failure or low light under representation
Dearth of Specialized Management

1) Project Management (Software Dev) ≠ Project Management (R & D)

2) Usually Senior Dev Managers recruited for Project Management (R & D) -> May pose long term risks as research mgmt responsibilities vary

3) Suggest a co-management structure with sufficient collective experience in managing dev+test, possess domain expertise to make technically sound assumptions & uncover hidden requirements & must be open to technological transitions

   e.g) Co-MP management lacked in handling most of the prev mentioned points and also buckled to Cu-MP deadline pressures
Post Mortem Summary:

R & D @ Peak of Inflated Expectations Should Watch Out For...

1. Meticulous Requirements Analysis: Don’t Sign on a Blank Cheque!
2. Productionize SOTA only when safe
3. Have a data first mentality
4. Manage Talent Pool Wisely
5. Do not cut back on bridging knowledge gaps
6. Add modules to perform implicit explainability to induce explicit accountability
7. Choose the right people to manage R&D
Conclusion:

Link to RE - Does this look familiar?

Source:

Thank You!

Q & A