Table of Contents

Surya || 16th Dec 2021

FOUNDATION MODELS

- Machine Learning pre 2010s – domain-specific feature extraction
- Post 2010s with Deep Learning (large data / compute / resources):
  - **Homogenization** : similar building blocks [Transformers]
  - Devise your problem as some form of sequence learning
    - Predict next sentence [NLP]
    - Predict next action [RL]
    - Predict next nucleotide [Protein / alphafold]
    - Predict next bounding box coordinate [Object Detection]
    - Combine any of these – prepare new tasks (protein + clinical report text + x-ray image as some sequence)
  - **Homogenization** : similar architecture, framing

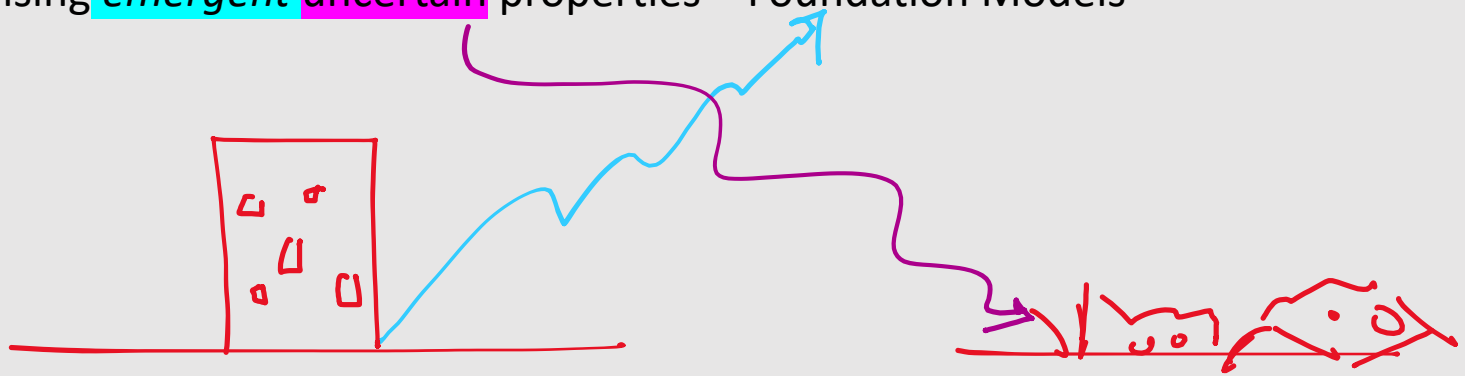
https://arxiv.org/abs/2108.07258

Table of Contents

Surya

FOUNDATION MODELS

Super large (param) models that trained on various forms of large-scale data displaying surprising *emergent* uncertain properties = Foundation Models



Everyone builds on top of these models – think xyzBERT, BERTology, CLIP embeddings

Emergent

- Ability to predict/answer things that we haven't even trained the models for
- Example : GPT-3 Prompts
- VQGAN-CLIP outputs [shown later]

Uncertain

- Don't know what outputs to expect
- Why are we seeing the emergent properties?
- Does it hurt a certain group of people more?
- Is it wise to spend so much money / compute on "uncertain" things?

Table of Contents

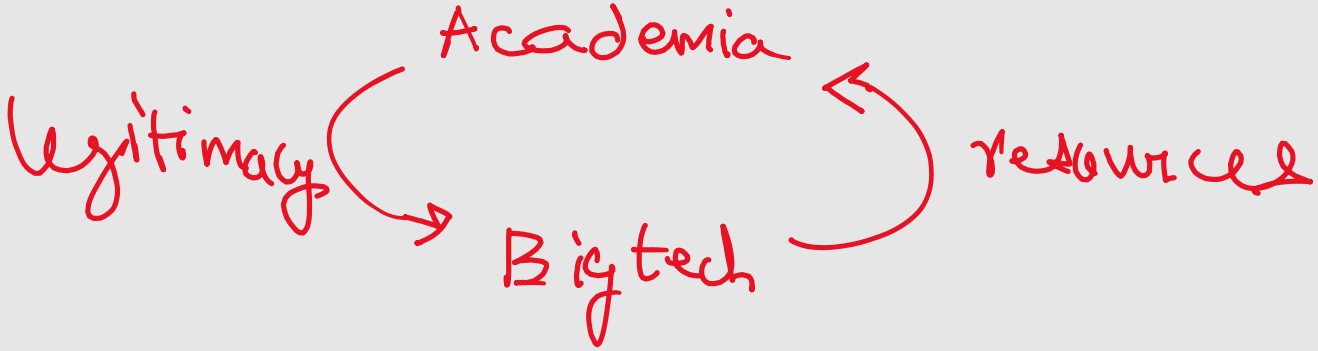
Surya

FOUNDATION MODELS

Why should big-tech have all the fun?

- Currently only they have the resources to train these models
- They probably won't study the properties well
- Lots of harms possible due to this

Thus, it makes sense that academia (Stanford) gets resources to train and test these models in an inter-disciplinary mode (CS, Philosophy, Sociology, Ethics)



Criticism

- **Name**: Why call it foundation? Some sort of PR hack? [fundamental / "morally" strong word]
- What if academia (Stanford) also makes it **inaccessible** to train / test?
  - Other open-source implementations like hface / eleutherAI [gpt-jax] are there
- Why should we assume that these models are useful?
  - ***Why should we build them in the first place?***

https://interactions.acm.org/archive/view/november-december-2021/the-steep-cost-of-capture

https://twitter.com/emilymbender/status/1420175904337121281

Surya

FOUNDATION MODELS

ViT – An Image is Worth 16x16 Words [Google AI ; ICLR 2020]

- Influential paper; highly cited
- Code is provided / Detailed description of dataset
- Every requirement is written clearly

Problem?

- Need to learn JAX to implement it or to understand the code
- Need to learn how to use TPUs [Google Cloud credits required]
- Or need ColabPRO to get TPU Pods
- The main dataset of the paper (based on Google Photos) is not available publicly

We just use their embeddings now. No way to test other than re-implement from scratch in some other language; train it on a (slower) hardware.

But, but, reproducibility checklist is satisfied, right?
At least, they gave us some details. They really didn't have to.

Maybe a marketing trick to bring people to JAX / TPU?

Table of Contents

Surya

FOUNDATION MODELS

Tesla – End-to-end system

Data: Roads, grounds, terrains, cars, drivers, cameras – entirely in their hands

Curation / Annotation: Hire people to annotate – in their hands

Training: Specialized chips, neural architectures, servers, GPUs (DOJO) – in their hands

Deployment: Cars, Software, etc. – in their hands

Get errors/mismatches from trial deployment. Send feedback to the model. Improve forever. **Continual Learning.**

An interesting very large system – wherein the entire stack is novel. Emergent / uncertain properties exist. Foundation?

Not free to build on top though.

https://www.youtube.com/watch?v=NSDTZQdo6H8

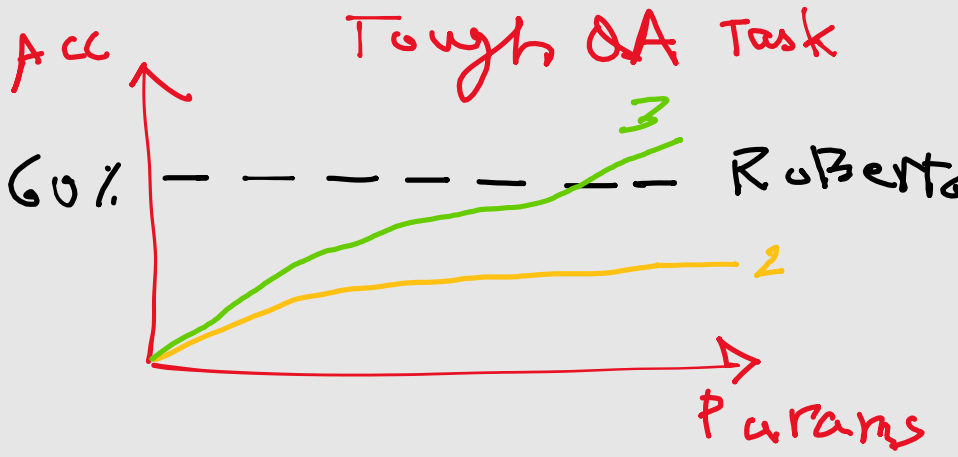
Table of Contents

Surya

FOUNDATION MODELS

Prompt Engineering = Rapid Prototyping

```
import gpt_2

# read questions from file
questions = read_file()

model = gpt_2.create()
answers = model.run(questions)
```




```
import gpt_3

# read questions from file
questions = read_file()

prompt = """
Q: Who is PM? A: Modi
Q: Capital of India? A: Delhi
...
"""

model = gpt_3.create(prompt)
answers = model.run(questions)
```

```
import gpt_3

# read questions from file
questions = read_file()

prompt = """
I: Average Movie O: Negative
I: Super fun? O: Positive
...
"""

model = gpt_3.create(prompt)
sentiments = model.run(questions)
```

Semantic parameters! Just with a few lines of prompt, we created:
- A question answering model ; - A sentiment classifier model

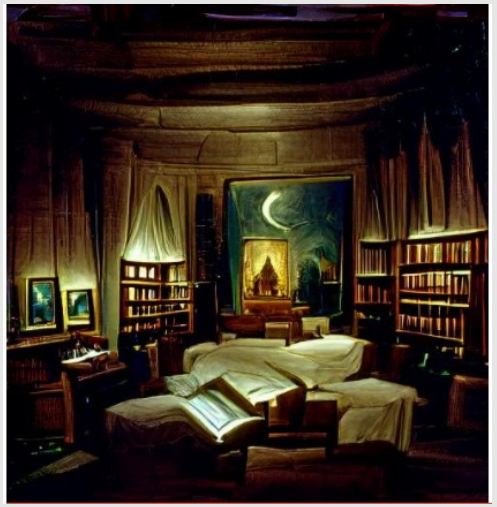
Table of Contents

Surya

FOUNDATION MODELS

```
import clip

# read images
images = read_images()

prompt = [“is photoshopped”, ”is not photoshopped”]
model = clip.create(label = prompt)
is_im_photoshopped = model.run(images)
```

```
import vqgan-clip

prompt = [“a cosmic room of seismic surprises”]

image = vqgan-clip.generate(prompt)
```



A fake-image detector in just few lines of code

- No retraining; no finetuning
- Replace labels to make your own model == prompt engineering

Emergent

- We didn’t even train for any of these
- Just some self-supervised cosine similarity

Uncertain

- What if I change order of prompt sentences
- What if I decrease prompts
- What if I use different words
- What are the right words to use

Table of Contents

Surya

FOUNDATION MODELS

https://moultano.wordpress.com/2021/07/20/tour-of-the-sacred-library/

No training
No finetuning

Just identify the right prompts
[takes a lot of effort]

And run the model