Solve edge AI problems with foundation models

Daniel Situnayake
Hello, I’m Daniel Situnayake!

- Director of Machine Learning at Edge Impulse
- Wrote *AI at the Edge* and *TinyML* (O’Reilly)
- Previously worked on *TensorFlow Lite* and *TFLM* (Google)
- *Superficial Intelligence* newsletter (dansitu.substack.com)
Foundation models

- Pre-trained models
- Trained on broad datasets
- Applied to tasks outside their training
- Tend to be large! Hundreds of megabytes to terabytes.

Generative AI

- Create data in addition to consuming
- Can be implemented using foundation models
- Size can vary greatly depending on task

Text  Audio  Image  Code

Genomics  Time series

Writing  Speech  Denoising  Code

Images  Music
Model sizes

Large Language Model Evolution

<table>
<thead>
<tr>
<th>Model</th>
<th>N Params</th>
<th>Max Tokens</th>
<th>HF Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phi-2</td>
<td>2.7B</td>
<td>2048</td>
<td>61</td>
</tr>
<tr>
<td>TinyLlama</td>
<td>1.1B</td>
<td>2048</td>
<td>53</td>
</tr>
<tr>
<td>Rocket</td>
<td>3B</td>
<td>1024</td>
<td>51</td>
</tr>
<tr>
<td>Mamba GPT</td>
<td>3B</td>
<td>2048</td>
<td>44</td>
</tr>
<tr>
<td>Guanaco Uncensored</td>
<td>3B</td>
<td>2048</td>
<td>39</td>
</tr>
<tr>
<td>Incite</td>
<td>3B</td>
<td>2048</td>
<td>39</td>
</tr>
<tr>
<td>OpenLLlama</td>
<td>3B</td>
<td>196K</td>
<td>36</td>
</tr>
<tr>
<td>Orca</td>
<td>3B</td>
<td>1024</td>
<td>35</td>
</tr>
<tr>
<td>Pythia</td>
<td>1.4B</td>
<td>2048</td>
<td>35</td>
</tr>
<tr>
<td>OPT</td>
<td>1.3B</td>
<td>2048</td>
<td>35</td>
</tr>
<tr>
<td>Lamini Neo</td>
<td>1.3B</td>
<td>2048</td>
<td>35</td>
</tr>
<tr>
<td>Lamini GPT</td>
<td>1.5B</td>
<td>1024</td>
<td>35</td>
</tr>
<tr>
<td>Lamini GPT</td>
<td>774M</td>
<td>1024</td>
<td>32</td>
</tr>
<tr>
<td>Pythia</td>
<td>410M</td>
<td>2048</td>
<td>31</td>
</tr>
<tr>
<td>Lamini Cerebras</td>
<td>1.3B</td>
<td>2048</td>
<td>30</td>
</tr>
<tr>
<td>Pythia</td>
<td>160M</td>
<td>2048</td>
<td>29</td>
</tr>
<tr>
<td>Lamini Neo</td>
<td>125M</td>
<td>2048</td>
<td>29</td>
</tr>
<tr>
<td>Pythia</td>
<td>70B</td>
<td>2048</td>
<td>28</td>
</tr>
<tr>
<td>Lamini GPT</td>
<td>124M</td>
<td>1024</td>
<td>28</td>
</tr>
<tr>
<td>Lamini Cerebras</td>
<td>590M</td>
<td>2048</td>
<td>28</td>
</tr>
<tr>
<td>Lamini Cerebras</td>
<td>256M</td>
<td>2048</td>
<td>28</td>
</tr>
<tr>
<td>Lamini Cerebras</td>
<td>111M</td>
<td>2048</td>
<td>28</td>
</tr>
</tbody>
</table>

https://medium.com/@daniellefranca96/battle-of-the-smallest-llms-e923e2cac1ff
Where we’re headed (warning, unscientific chart)

Model FLOPs for good performance
Compute efficiency

Time
“Large” models will eventually arrive on cheap, low power devices
“Large” models will eventually arrive on cheap, low power devices

But we don’t need to wait.
Four **key capabilities** of foundation models

- Zero-shot learning
- Reasoning
- Information retrieval
- Data generation
Zero-shot learning

Zero-shot time series forecasting

Zero-shot image classification with multimodal LLM

Prompt: “Classify this image as hotdog or not hotdog”

Response: “hotdog”

Zero-shot question answering with BERT

Prompt: “How do I change the batteries?”

Document: 

Response: “In order to change the batteries...”

https://blog.research.google/2024/02/a-decoder-only-foundation-model-for.html
Reasoning

Determining the right action

Prompt: “Plan a maintenance window based on the production line status”

Response: “A reasonable maintenance window is...”

Intent matching

Intents: dispense_drink, dispense_food

User: “I want a soda please”

Match: dispense_drink

Reasoning based on documents

Prompt: “Is the proposed solution legal?”

Document:

Response: “Yes, the solution proposed is...”
**Information retrieval**

Looking up facts with LLM + RAG

Prompt: “How can I treat this plant disease?”

Response: “This looks like <disease>, which can be treated with <treatment>.”

**Multimodal lookup**

Prompt: “Play a song with heavy guitar I have not heard before”

Response: [Song by Famous Prophets (Stars)]

**Question answering with BERT**

Prompt: “How do I change the batteries?”

Document: [Document icon]

Response: “In order to change the batteries…”
Data generation

Denoising and upscaling

Generating text and audio

Prompt: “Tell me a story about unicorns, with pictures”

Response: “Once upon a time...”

Video and audio generation  https://openai.com/sora

https://github.com/facebookresearch/denoiser
Are foundation models capable of these?

Yes.

- Zero-shot learning
- Reasoning
- Information retrieval
- Data generation
Are foundation models required?

No!

- Zero-shot learning
- Reasoning
- Information retrieval
- Data generation
Zero-shot learning on the edge

Benefits of large foundation models

- Reduces training data requirements
- Allows task to be adjusted on-the-fly

Alternatives

- Can implement in other ways (embeddings + nearest neighbor lookup, etc)
- Use smaller, domain-specific models (custom BERT)
- Can use zero-shot model for data labelling then train a conventional model
Reasoning on the edge

Benefits of large foundation models

- Understand complex user communication
- Match inputs to states
- Make sophisticated decisions

Alternatives

- Language - intent matching and slot filling
- State machines (game design)
- Smaller, domain-specific models (perhaps created via distillation)
Information retrieval on the edge

Benefits of large foundation models

- Convenient retrieval of information
- Language-based interface
- Answer any possible question

Alternatives

- Smaller, domain-specific models (custom BERT)
Data generation on the edge

Benefits of large foundation models

● Create and manipulate signals
● Generate multimodal content

Alternatives

● Smaller, domain-specific models
  ○ Visual question answering
  ○ Signal-to-signal for specific use cases
● Small, distilled generative models
Designing with foundation models at the edge
1. Frame your problem

● Which special capabilities do you require?
  ○ Zero-shot learning
  ○ Reasoning
  ○ Information retrieval
  ○ Data generation

● Can it be framed more simply? (classification, regression, clustering, etc.)
2. Determine your constraints

- Do you need to run on-device?
  - Bandwidth
  - Latency
  - Economics
  - Reliability
  - Privacy

- What are your hardware capabilities?
  - GPU
  - NPU
  - CPU
  - MCU
3. Is there a non-ML solution, or an existing solution, that works?

- **Algorithm choice**
  - Rule-based AI
  - Digital signal processing
  - State machines

- **Pre-trained deep learning models**
  - TinyBERT
  - Small LLMs
  - Quantization?
4. If you have to use an on-device model, make it simple

- Use a simple, non-foundation model where possible
  - For zero-shot can you just use embeddings and k-nearest neighbors?

- Transfer knowledge from foundation models to domain-specific simple ones
  - Label data with zero-shot learning models
  - Generate synthetic data with generative models
5. Increase complexity only when required

- Watch your costs and constraints
- Fine-tune instead of training from scratch
- Try to predict performance before spending money on training
How to design with foundation models at the edge

1. Frame your problem. Which capabilities do you require? (zero-shot, data generation, etc.)

2. Determine your constraints. Do you need to run on-device?

3. Look for a non-ML solution, or an existing solution that already works.

4. If you have to use an on-device model, make it simple.

5. Increase complexity only when required.
Foundation models in the edge AI toolchain
Labelling assistance
Labelling assistance
Synthetic data

- Text to image
  - Dall-E, stable diffusion, etc.

- Audio
  - Generate data for keyword spotting

- Many other things!
  - NeRF (2D to 3D)
    - https://blogs.nvidia.com/blog/instant-nerf-research-3d-ai/
  - 3D scene synthesis
    - https://machinelearning.apple.com/research/roomdreamer

https://docs.edgeimpulse.com/docs/tutorials/ml-and-data-engineering/generate-synthetic-datasets
Training!

VeLO: Training Versatile Learned Optimizers by Scaling Up

Edge AI and foundation models
in the future
1. Hardware-software crossover

- Model FLOPs for good performance
- Compute efficiency
2. Disconnectivity

No more subscriptions, models as IP
3. The curse of generality

Goodbye, GPT
3. Embodiment
Thank you!

edgeimpulse.com
dansitu.substack.com