# ARTIFICIAL INTELLIGENCE: HINTS FROM THE PAST

IAN F. BLAKE LECTURE SERIES

## Contents



## 1. Introduction

A couple of words about the Ian F. Blake lecture series. It started in 2019 with a gift made to UBC by Dr. Vijay Bhargava. Ian F. Blake is a leading authority in the areas of coding theory, cryptography, and information theory.

Today's lecture is given by Yaser S. Abu-Mostafa from the California Institute of Technology on *Artificial Intelligence: Hints from the Past*.

Outline of the talk: the story, hints, and the future. There are 30 slides. He is going to give his perspective on AI and its effect on education in particular.

## 2. The story

*How to imitate the brain?* We are looking into an existence proof of intelligence. We looked at the brain and found neurons. Then we said: let us try to imitate it. There were different models in that era. One that survived in modern AI was back-propagation; another important direction was the Hopfield model.

At that point, this simple imitation was still very speculative. The analogy is with birds and flight. We did not imitate birds literally, but studying them eventually led to airplanes.

Next came a paradigm shift in the 1980s: from pure design to a learning approach from data. Key ingredients:

---

*Date*: Speaker(s): Professor Yaser S. Abu-Mostafa
April 23, 2026.

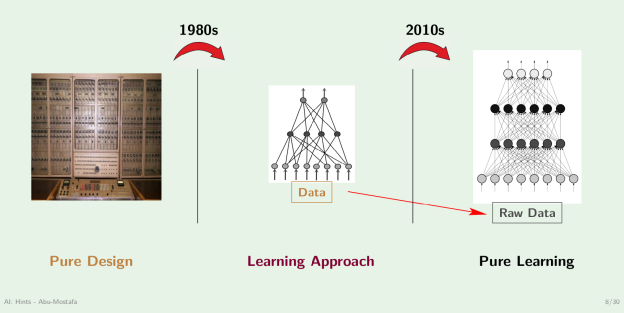


FIGURE 1. Two major breakthroughs

- learning; no one comes to your brain and puts information exactly where it should be;
- collective computation; the old approach is easier to describe because the machine already has an answer within it, while in a neural network the answer is spread out within the network weights.

Examples where neural networks won:

- a neural network that learned to play backgammon,
- neural networks in financial engineering.

*Neural networks are the second-best way to solve just about anything.* So what was the missing ingredient?

The missing ingredient came around 2010: learning from raw data at scale with pure learning.

Now we are going to go through three aspects that came together around 2012: data, protocols, and computing.

2.0.1. *Data — Self-supervised learning.* We need labels, and we need a lot of data. What is a fast way to label data? Take *War and Peace*, for example, and look at 7 consecutive words. If the phrase is

*Was the first to arrive at her,*

then

*Was the first arrive at her*

is the input and *to* is the output.

2.0.2. *Protocols — transfer learning.* With a deep network, the hidden layers learn higher-level representations of objects in an image. We are interested in the layers that take an image and produce a higher-level representation of it. Then we can move to a new problem, for example detecting cancer from X-ray scans. The idea is that if you learned something useful from a previous problem, then when you move to a similar new problem, the system can learn it faster.

2.0.3. *Protocols — pre-training.* A big part of this is the foundation-model idea. It is like a baby trying to understand an environment: pre-trained on a huge amount of crude data. After that, you get a system that is familiar with the environment. We go from a baby to a teenager. To get a specialist, we need to fine-tune, handle more complex input, polish the output, and then we get a specialist. In my lab, we are creating a foundation model for cardiology.

2.0.4. *Computing — GPU.* Ilya (?) wrote a compiler that implemented back-propagation on GPU chips. Observation: academia versus industry.

Some landmark contributions of AI: AlexNet (2012) and GANs (2014). Fast-forward 10 years: AlphaFold (2021) and ChatGPT (2022).

What AI offers: doing what we already do, and doing what we cannot do.

## 3. Hints

A hint is a known property of the target function. For example, invariance:

$$x \mapsto x' \qquad \text{where} \qquad f(x) = f(x').$$

Create a new labeled example to encode a desired property: data augmentation. Take an image, move it by 5 pixels, and it is still a cat. Create a new example $(x', f(x))$ from $(x, f(x))$, and add to the error minimization

$$(g(x') - f(x))^2.$$

Learn the property with no labels: hints. Create an unlabeled virtual example $(x \mapsto x')$ and minimize

$$(g(x') - g(x))^2.$$

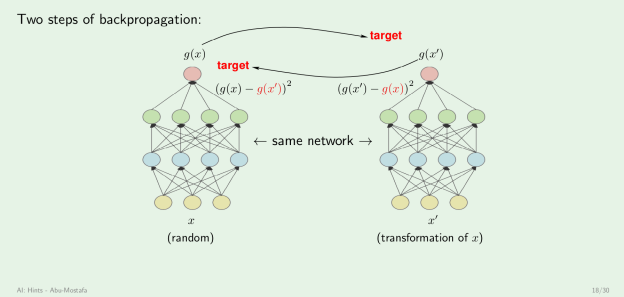


FIGURE 2. Hints in action

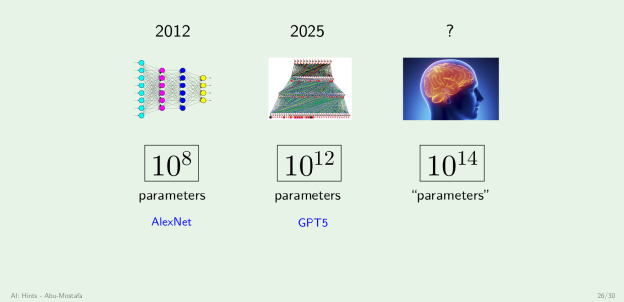


FIGURE 3. Number of neurons comparison

How to implement it? Use two steps of back-propagation: in

$$(g(x') - g(x))^2,$$

make them a target of each other and ask to minimize it again with the switched roles as targets.

Theoretically, hints should be superior.

Data augmentation: danger of overfitting, since we are just creating more labeled data in input space. A mixed objective enforces both the label and the hint.

Hints: no direct overfitting in the same way. In this case, we have independent objectives.

Results? Mixed. There is one application where it worked great: financial forecasting. For computer vision, there was no comparable success at first. The missing ingredient was the input distribution. If you try to learn the hint on random patterns, then outputting a constant is also invariant. It works better for finance because the data is already closer to random.

What do we do in computer vision to make it work? We need to mimic an input distribution. This is where a generative model, such as a GAN, comes in.

## 4. Where is this going?

How far can we go? AlexNet had about $10^8$ parameters, ChatGPT about $10^{12}$ parameters, and the human brain about $10^{14}$ parameters. What capabilities will this achieve? Emergent abilities.

With about $10^{22}$ FLOPS, many emergent abilities arise for different models.

More good than evil? Healthcare and education have positive impact. Bad directions are warfare and cybercrime.

Speaking of education: AI is the new calculator. What are the essential skills now? Is AI-assisted cheating intractable?

Thank you!

Notes by Maksym Zubkov