Neuromorphic Computing

Dr. Charles Kind

Computational Neuroscience Group, SCEEMS, Bristol University

October 2022

Lecture series contents

- The state of computer hardware today.
- Challenges in hardware driven by societal needs (data science).
- What future solutions could answer current (and near future) data processing needs?
- How are brains like computers?
- How are computers like brains?
- What lessons learnt about human brains can we apply to computers?
- What is neuromorphic computing?
- The three main approaches to neuromorphic computing.
- Examples of the state of the art in neuromorphic computing.

Lecture review of last week

- There are important, physically fundamental, reasons why transistor and therefore general computing improvements have slowed
- We have tried two main hardware solutions to answer this issue:
  - More cores: but there are limits to paralellism in many arenas
  - Specialisation of components: GPUs with many parallel arithmetic processors and task specific components such as video encoder/decoders
- Computers and brains operate in very different power regimes:
  - Computers draw very high power loads for extreme precision and reliability
  - Brains have very low power requirements for low precision but highly parallel tasks
- Computers and brains therefore excel at different tasks
- Why bother emulating brains with computers? Why bother emulating computers with brains?

Neuromorphic Computing

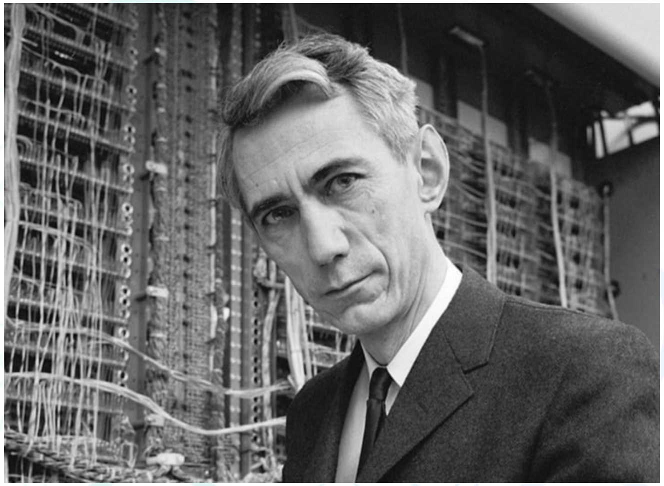
Goals

- Scalable architecture designed to run brain like computations
- Replace virtual neurons/synapses with physical, analog devices
- ‘In memory’ devices
- Reduce power consumption
- Enable ‘edge computing’
- Sit in the gap between high accuracy, high energy classical computing and low accuracy, low energy neural computation
- Become standard CPU/hardware extension like SIMD SSE
- Learn more about how the human brain works

Shannon Capacity Theorem

Claude Shannon, an engineer and mathematician, worked on cryptanalysis in the second World War and developed useful theories on information theory that where published in 1948.

Shannons capacity theorem provides us with a method to analyse communication energy efficiency with respect to errors.



Claude Shannon

Shannon's theorem (in words):

- A given communication system has a maximum rate of information C known as the channel capacity.
- If the information rate R is less than C, then one can approach arbitrarily small error probabilities by using intelligent coding techniques.
- To get lower error probabilities, the encoder has to work on longer blocks of signal data. This entails longer delays and higher computational requirements.

Thus, if $R \leq C$ then transmission may be accomplished without error in the presence of noise.

Shannon Capacity Theorem

$$C(B,S) = B \log_2(1 + S/N)$$

Where:
C is the channel capacity in bits per second
B is the bandwidth of the communication channel
S is the average received signal power
N is the average noise power

What relationships can we see from the equation?

- Channel capacity is linearly proportional to bandwidth
- If signal far exceeds noise (S>>N) the logarithm term can grow without bound
- If noise far exceeds signal (N>>S) then, in the limit, the logarithm goes to zero

Shannon Capacity Theorem

$$C = B \log_2(1 + S/N)$$

As $B \to \infty$, the the channel capacity does not become infinite since, with an increase in bandwidth, the noise power also increases. If the noise power spectral density is $\eta/2$, then the total noise power is $N = \eta B$, so the theorem becomes

$$\begin{aligned} C &= B \log_2(1 + S/\eta B) \\ &= S/\eta(\eta B/S)\log_2(1 + S/\eta B) \end{aligned}$$

Which simplifies in the limit as $B \to \infty$ to $C_\infty = \log_2(e)(S/\eta)$

This gives the maximum information transmission rate possible for a system of given power but no bandwidth limitations.

Shannon Capacity Theorem example

$$C = B \log_2(1 + S/N)$$

Let us compare the channel capacity for two different signal strengths and fixed bandwidth. For case one S=3N and for case two S=15N. We therefore have:

One: C = B log2 (1+3N/N) = 2B
Two: C = B log2 (1+15N/N) = 4B

If we define energy efficiency (F) as capacity divided by total energy (F = C/(S+N)) then for case one F = C/4N and for case two F = C/16N. Hence energy efficiency doubles as signal energy decreases from 15N to 3N!

Without reference to other energy costs it is clearly more efficient to have multiple low capacity connections than one high capacity connection.

So where does this fit in the question of computer vs brains vs computers?

Brains vs computers vs brains

Power vs accuracy

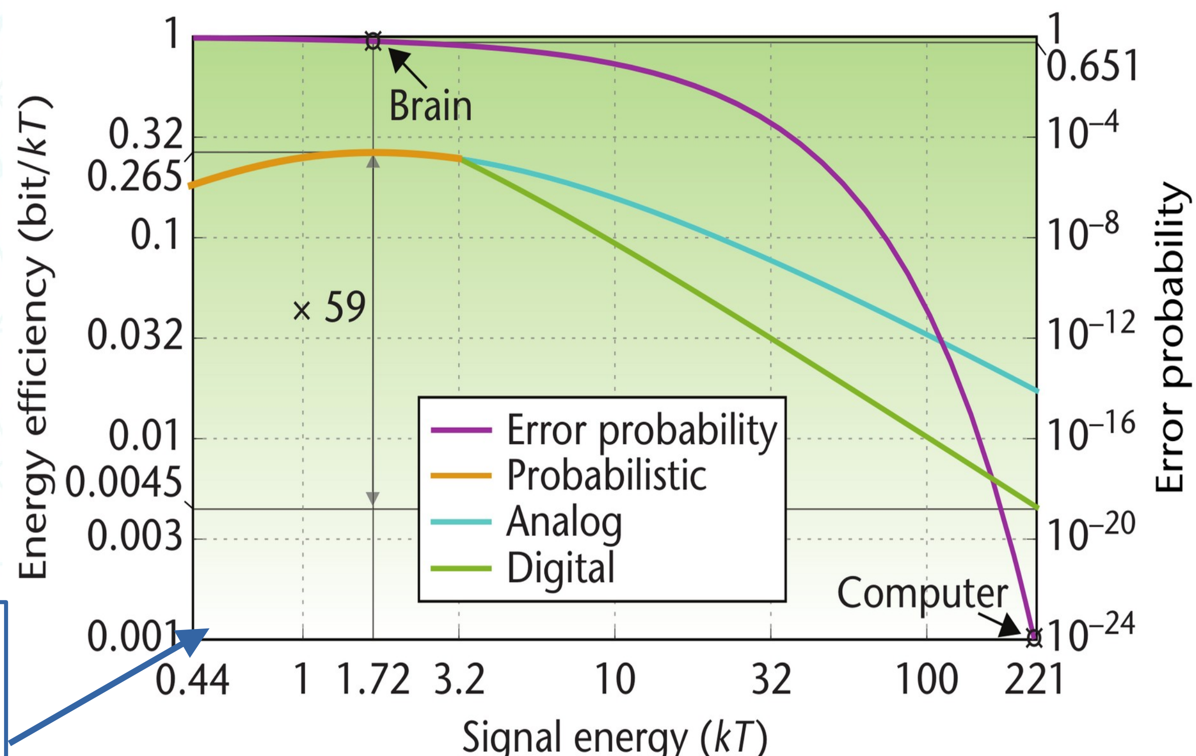
- Computer's digital computations cannot tolerate errors and pay a high energy cost to minimise it. If a computer makes one error every 10 days then at current speeds it has an error rate of around p = 1e-24 and must operate with an energy greater than 220N (probability of error is given by exp(-0.25(E/N)). This is ~60 times lower than the maximum possible.
- By using analog computation, which degrades gracefully, brains tolerate errors. It is estimated (C Allen et al. 1994) that voltage spikes in the human brain fail to trigger ligand release 2/3 of the time and carry around one bit of information (S Panzeri et al. 2001)

Shannon capacity theorem provides us with a method to analyse communication energy efficiency with respect to errors. Note here N=kT.

Brain: Analog signals, high energy efficiency at low precision

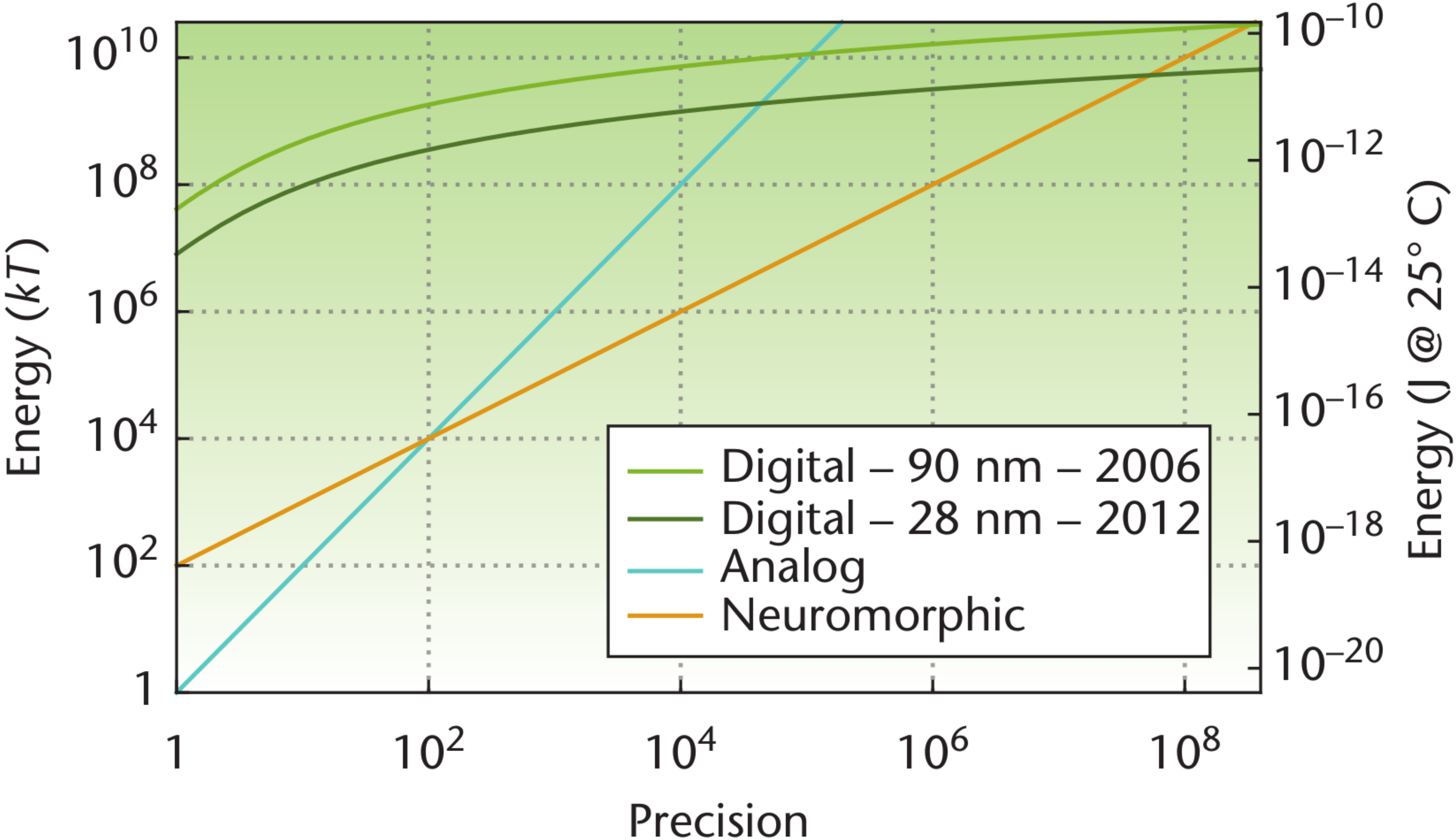
Computer: Digital signals, high energy efficiency at high precision

A signal with energy E conveys $b = ½\log_2(1 + E/kT)$ bits of information with an energy efficiency of $b/(E + kT)$ bits per joule.



Kwabena Boahen (2017)

The energy cost of computation



The neuromorphic curve assumes neuron counts scale linearly with precision. Here we see far more clearly the costs associated with computers simulating brains and brains simulating computers. Is there a middle ground?

Where does Shannon leave us?

When should we use computers?

When should we use brains?

How much precision does a task require? Consider:

Task		Precision?
Driving a car		Metres to centimetres
Calculating the digits of pi		Abritrary but can be very high
Facial recognition		Rarely sub-pixel
Flying a rocket to the moon		High say ~ 32 bit FP
Natural language processing		Low
Travelling salesperson problem		Low but potentially high number of nodes

Neuromorphic Computing

Thank you :)