

The Morse Code Room

Applicability of the Chinese Room Argument to Spiking Neural Networks

Masterarbeit
zur Erlangung des Hochschulgrades
Master of Arts
im Master-Studiengang Philosophie

vorgelegt von

Johannes Brinz

Institut für Philosophie
Philosophische Fakultät
Bereich Geistes- und Sozialwissenschaften
Technische Universität Dresden
2022

Eingereicht am 08.August 2022

1. Gutachter: Prof. Dr. Moritz Schulz
2. Gutachter: PD. Dr. Rico Hauswald

Summary

Abstract:

The Chinese room argument (CRA) was first stated in 1980. Since then computer technologies have improved and today spiking neural networks (SNNs) are “arguably the only viable option if one wants to understand how the brain computes.” (Tavane et.al. 2019: 47) SNNs differ in various important respects from the digital computers the CRA was directed against. The objective of the present work is to explore whether the CRA applies to SNNs. In the first chapter I am going to discuss computationalism, the Chinese room argument and give a brief overview over spiking neural networks. The second chapter is going to be considered with five important differences between SNNs and digital computers: (1) Massive parallelism, (2) subsymbolic computation, (3) machine learning, (4) analogue representation and (5) temporal encoding. I am going to finish by concluding that, besides minor limitations, the Chinese room argument can be applied to spiking neural networks.

Kurzzusammenfassung:

Das Argument vom chinesischen Zimmer wurde erstmals 1980 veröffentlicht. Seit dieser Zeit hat sich die Computertechnologie stark weiterentwickelt und die heute viel beachteten gepulsten neuronalen Netze ähneln stark dem Aufbau und der Arbeitsweise biologischer Gehirne. Gepulste neuronale Netze unterscheiden sich in verschiedenen wichtigen Aspekten von den digitalen Computern, gegen die die CRA gerichtet war. Das Ziel der vorliegenden Arbeit ist es, zu untersuchen, ob das Argument vom chinesischen Zimmer auf gepulste neuronale Netze anwendbar ist. Im ersten Kapitel werde ich den Computer-Funktionalismus und das Argument des chinesischen Zimmers erörtern und einen kurzen Überblick über gepulste neuronale Netze geben. Das zweite Kapitel befasst sich mit fünf wichtigen Unterschieden zwischen gepulsten neuronalen Netzen und digitalen Computern: (1) Massive Parallelität, (2) subsymbolische Berechnung, (3) maschinelles Lernen, (4) analoge Darstellung und (5) zeitliche Kodierung. Ich werde schlussfolgern, dass das Argument des chinesischen Zimmers, abgesehen von geringfügigen Einschränkungen, auf gepulste neuronale Netze angewendet werden kann.

Contents

Introduction	3
Theoretical background	7
I Strong AI: Computationalism	7
II The Chinese room argument	14
III Spiking neural networks	22
Applicability to spiking neural networks	29
I Massive parallelism	29
II Subsymbolic computation	31
III Machine learning	37
IV Analogue representation	41
V Temporal encoding	47
VI The Morse code room and its replies	50
VII Some more general considerations regarding hardware and software	54
Conclusion	57
Bibliography	59

Introduction

The Chinese room argument (CRA) counts as one of the most influential arguments of contemporary philosophy. It was first published in 1980 by American philosopher John Searle in his paper *Minds, Brains, and Programs* and has sparked a heated debate in the field of artificial intelligence (AI), and beyond, about whether it is possible to program intelligent computers. In 1980, when the article first appeared, computer science was a new and exciting research area. The first department of computer science had been formed only 18 years before in 1962 and the first person to receive a PhD from a computer science department was Richard Wexelblat in 1965. (see Shallit 1995) The moon landing in 1969 was supported by the *Apollo Guidance Computer*, a system that was processing landing information in real time, the first micro processor was developed around the same time in the laboratories of Intel and the first modern programming languages like C and Pascal arose. Also in theoretical informatics pioneering articles appeared in the fields of complexity theory, algorithm development and cryptography. The RSA-crypto system still used today for data encryption was developed in 1976, the same year in which Steve Wozniak and Steve Jobs founded Apple Computers. In 1979, three graduate students developed a distributed news server that later became Usenet, a predecessor of the internet. (see Shallit 1995)

The rapid progress in the IT sector gave rise to a growing interest in the field of AI. In 1965 Gordon Moore, an American engineer and co-founder of Intel, stated that the number of transistors, and thus the computational speed, of computer chips is going double every 12 months. Around the same time machine learning algorithms improved and programmers got better at solving certain problems. Early algorithms such as Joseph Weizenbaum's ELIZA showed promise towards understanding and interpreting natural languages. This lead government agencies to invest large amounts of money in the emerging field of AI. The US government, for example, was particularly interested in computer programs that could translate and transcribe spoken language. In the 1970s optimism about machine intelligence was high. (see Anyoha 2017) In 1970 leading cognitive and computer scientist Marvin Minsky told the Life-magazine "from three to eight years we will have a machine with the general intelligence of an average human being." (Minsky as in Anyoha 2017)

Also in philosophy the late developments were greeted with excitement. In 1967 Hilary Putnam presented the first version of what should soon become the leading paradigm in cognitive science: *computationalism*. (see Rescorla 2015: 10) Computationalism is the idea

that the mind should best be explained as a computer program. Basically, what philosophers claimed they had found was a solution to one of the deepest philosophical mysteries: *the mind body problem*. Since the ancient Greek thinkers were wondering how mentality can be integrated into the physical world of natural sciences. How can a chunk of matter, like our brain, give rise to conscious thought, feelings and emotions? What happens in our brains and how does it relate to the subjective feeling we get of it? Computationalists claimed that they had found the solution. *Our brain carries out computations and our mind relates to them like the software of a digital computer relates to its hardware*. Our brain is essentially a information processing machine that derives the experience of a piece of music, for example, from the mechanical motion of the sounds that stimulate our eardrums. And whereas it seems mysterious how the mind relates to the brain, we indeed know how the software relates to the hardware. This relation is constantly being explored by computer scientists around the world. (see Searle 2004: 63-66)

It was in this agitated time that Searle published his CRA as a critique of computationalism, or *strong AI* as Searle has baptised it. The CRA can be briefly summarised as follows. Imaging a man who does not speak Chinese inside a room which contains nothing but a desk, a rule book and some scratch paper. Imagine further that this man is being handed bunches of Chinese symbols, which he does not understand, manipulates them according to the rules in his book and returns the result of this manipulations to the outside. Unknown to him, the first set of symbols he received were questions in Chinese and the rule book was written such that the set of symbols he returned were the appropriate answers. But although it looks as if he was speaking Chinese, he did not understand anything, he was just shuffling cards of meaningless symbols. And now Searle's point is: *This is all computers do. They manipulate symbols, i.e. 0s and 1s, according to a program, i.e. a set of rules*. Therefore, if the man in the Chinese room does not understand Chinese, neither does the digital computer. So there is at least one thing that computers cannot do which average human beings can, namely understand Chinese. Searle's argument provoked strong objections from the AI community as well as from proponents of computationalism and has been cited more than 9000 times as for today.¹

However, the CRA is clearly directed against digital computers. Just as a digital computer the Chinese room operates on symbolic representations, manipulates them according to a program and works in a step-wise manner returning one symbol after another. However, since the 1980s computer technology has made tremendous process. Even the most powerful supercomputers of that time can hardly compete with the computational power of a smartphone. But not only have digital computers become significantly faster but also new technologies have evolved. In 1957 Frank Rosenblatt developed the *perceptron*, a brain like architecture that uses neurons and synapses as computational units. The *Mark-1 Perceptron* was the first hardware implementation of Roesnblatt's neural net. It was designed for image recognition tasks and

¹According to Google Scholar: https://scholar.google.de/scholar?hl=de&as_sdt=0,5&q=minds+brains+and+programs (Accessed: 02.08.2022)

consisted of 400 photocells arranged in a 20×20 array and randomly connected to the artificial neurons via potentiometers that represented the synaptic connection strengths. The Mark-1 perceptron was capable of *learning* to discriminate shapes by adjusting the potentiometers in the right way using electric motors. (see Bishop & Nasrabadi 2006: 167) The development of artificial neural nets (ANNs) created new opportunities in machine learning and today most modern AI applications are based on neuronal architectures.

Neural networks have been evolving towards higher bio-fidelity. In recent years a lot of research resources have been put into the development of *spiking neural networks (SNNs)*. Biological brains use patterns of spikes in neuronal action potentials to process information and, unlike ANNs, SNNs resemble brains in this respect. Like the perceptron, they also use neurons and synapses to process information but instead of encoding information in the action potential, i.e. the voltage of the neurons they use *spikes*, i.e. sudden jumps in the potential to process information. The main difference between ANNs and SNNs is that the latter incorporate the concept of *time*, they use the timing of spikes to compute. Due to their high degree of bio-fidelity SNNs are often used for large scale brain models and some consider SNNs to be "the only viable option if one wants to understand how the brain computes." (Tavanaei et.al. 2019: 47) SNNs are still under development and in particular learning presents a challenge to modern research. However, substantial amount of research goes into the growing field of spiking computation.

SNNs differ in various ways from digital computers. Instead of a central processing unit (CPU), a memory and an input/output unit (I/O), SNNs compute using neurons and synapses. And while digital computers need to be programmed, SNNs are hard-wired to produce a certain input-output relation. Since the CRA is tailored to digital computers many have argued that it only presents a counter argument against standard computers and that modern computational technologies are not effected by it. The question of this thesis, therefore, is: *Can the CRA be applied to SNNs*. In answering this question I am going to proceed as follows. The first chapter is devoted to the theoretical background of the CRA. First, I am going to present *computationalism*, since it is the thesis that the CRA is supposed to refute. I will try not to commit myself to any particular form of computationalism but rather focus on the claim that is at the heart of all variants: *That implementing the right kind of computation alone is sufficient for the possession of a mind*. An important concept that is going to be discussed is that of implementation. What does it mean for a physical system like a computer to implement a computation. Next, I am going to discuss the actual CRA and the associated thought experiment. The third section of chapter one consists in a brief overview about SNNs. I am going to elaborate the relevant differences between SNNs and digital computers that are going to play a role in the main part of this work.

Chapter two is concerned with the main question of this thesis: Can the CRA be modified such that it applies to SNNs? Five differences between digital computers and SNNs are going

to be discussed. (1) Parallel processing. Digital computers usually use one central processing unit (CPU) to process information. SNNs on the other hand use large numbers of neurons that serve as computational units and that work *simultaneously*. While digital computers compute in series, i.e. perform one step after another, SNNs are capable of parallel processing. (2) Subsymbolic representation. Digital computers operate over *symbols* that have a syntactic and a semantic level. They are part of a rule-full system and they are used to represent things. The computational tokens of SNNs, on the other hand, don't refer to anything. Only the combination of many neurons can be interpreted as a meaning full representation. (3) Machine learning. Digital computers carry out fixed programs to relate the input to the output. in contrast to this SNNs successively adjust their synaptic connection strengths and, thus, *learn* their input-output relation. (4) Analogue representation. Digital computers, unsurprisingly, are digital. They use 0s and 1s to encode all other numbers. SNNs on the other hand are analogue, they can directly *represent any number* in the spiking patterns of different neurons. (5) Temporal encoding. SNNs use the precise *timing* of spikes to encode information. This distinguishes them from all other computers that use static state variables as representations.

There upon I am going to discuss in how far those differences are relevant with respect to the CRA. At the end of every section I am going to modify the Chinese room thought experiment such that the CRA applies to computers with the respective property. My considerations are going to culminate in the *Morse code room* argument, a modification of the Chinese room thought experiment such that it applies to SNNs. In the sixth section of the second chapter I am going to further discuss the Morse code room and evaluate whether some replies stated against the original CRA gain force against the Morse code room argument. The last section is going to be about some general remarks regarding hardware and software. *I am going to conclude that, with certain restrictions, the CRA can be applied to SNNs.*

Theoretical background

I Strong AI: Computationalism

At the beginning of the twentieth century philosophers have been trying to integrate the study of the mind into a scientific world view. The driving force behind cognitive science has been to free the study of the mind from all subjective and unscientific influences making psychology part of the natural sciences. Many philosophers, thus, turned to some form of materialism. Soon people claimed that neurobiology had discovered what mental states really are: Brain states of neuronal activity. (see Searle 2004: 38) Being in a certain type of mental state is identical to being in a certain type of brain state. An example often cited is that of pains and C-fibre firing: Being in pain is nothing but the activity of certain kinds of axons, the C-fibres. This so called *reductionism*, however, faced a problem. If being in a certain kind of mental state is nothing but a certain kind of neuronal stimulation having the right kind of neuronal basis is necessary for having that mental state. But this seems to be *neuronal chauvinism*: It restricts the ascription of mental states like ours to beings with the same neuronal substrate. But what about animals with a very different kind of brain like reptiles, birds or molluscs? (see Bickle 1998: 7) It seems implausible that animals with a different neuronal architecture cannot have human-like mental states such as pains, feelings of thirst or visual experiences.

Many, therefore, believed that reductionism, or type-identity theory, should be replaced by *token-identity theory*. Type identity theory states that types of mental states (e.g. pains) are identical to types of brain states (e.g. C-fibre excitation). Token identity theory, however, claims that every token of a mental state is identical to a token of a brain state. This change towards particular realisations of mental states avoids chauvinism. Mental states still are nothing but physical states of the brain, but the restriction on what kinds of brain states could serve as a neuronal basis has been lifted. This makes token-identity theory much more plausible. Imagine you and I both have the belief that Dresden is the capital of Saxony. It seems unnecessarily strict to suppose that we both are in exactly the same neurobiological state. (see Searle 2004: 42)

But token-identity theory faces a different problem: What individuates mental states? What fact about your mental state and my mental state makes them both the belief that Dresden is the capitol of Saxony? Token-identity theorists cannot claim that it must be some

irreducible mental property like the content of the belief, since the aim was to reduce all mental properties to physical properties. Neither can they claim that it must be the type of the brain states, since that again would yield type-identity theory. (see Searle 2004: 43) However, there must be *something* that characterises them as a belief of a certain kind. And the answer many philosophers gave was that it must be the functional role they play in the overall behaviour of a cognitive agent. This idea lays at the hart of *functionalism*.

Functionalism in the philosophy of mind individuates mental states in terms of their causes and effects. Pain, for example, is caused by tissue damage or trauma to bodily regions, and in turn causes specific beliefs (e.g., that one is in pain), desires (e.g., that one relieves the pain), and behaviours such as crying out, nursing the damaged area, and seeking out pain relieving drugs. (Bickle 1998: 7)

Your and my belief that Dresden is the capitol of Saxony, thus, are the same belief because they have the same function in our overall causal architecture. They are that entity that causes us to utter "Dresden" when being asked what the capitol of Saxony is, at least if we have the intention to be honest and a certain desire to speak. Two things are worth noting. First, mental states (e.g. a belief) are defined via their relation to a certain input (e.g. a question) and output (e.g. an answer) *as well as to other mental states (e.g. to desires and intentions)*. Second, mental states are defined as the function they serve in an abstract causal organisation. Functionalism makes no claim about the concrete physical realisation of this functional architecture. This is analogous to how many other things are defined, such as thermostats or clocks. A thermostat is any entity that serves the function of regulating the room temperature. A clock is any entity that tells the time. The physical structure is not important to the definition of clock and thermostat. "A clock, for example, can be made out of gears and wheels, it can be made out of an hourglass with sand in it, it can be made out of quartz oscillators, it can be made out of any number of physical mechanism that enables us to tell the time." (Searle 2004: 44)

This makes functionalism sit particularly comfortable with *multiple realisability*, i.e. the claim that mental states can be realised in a variety of physical systems. Biology suggests that mental states such as pains are realised in different neuronal substrates. And from multiple *realisation* follows multiple *realisability*: Mental states can in principle be realised in any kind of physical substrates such as human brains, mollusc brains, extraterrestrial brains or silicon circuits.

Combining functionalism with the claim that the functional architecture is best captured computationally yields *computationalism* or *strong AI*. According to computationalism mental states are individuated by their causal function and the causal function can be understood as the role they play in a certain computation. The theory of computation defines the abstract

functional organisation and the brain is the physical system in which it is implemented. Now, we seem to have discovered a solution to the mind-body problem: The mind relates to the brain as the software relates to the hardware in which it is implemented. Describing the brain on a mental level is the same as describing a computer on the software level: What matters is not the physical substrate, i.e. the hardware but the abstract functioning. Whilst the mind-body problem seems mysterious, the relation between software and hardware is well understood. (see Searle 2004: 45-46) This view is attractive for many reasons. First, it postulates nothing over and above the physical. All there is are physical substances, i.e. particles arranged in a certain way. But secondly, computationalism is more than just a crude identity thesis. Software and hardware are not simply the same, they are different levels of description. We can describe a computer either on the level of molecules and atoms, i.e. on the hardware level or on the level of abstract causal organisation, i.e. on the software level. In the same way we can understand the brain either on the physical level of neurons and neuro-transmitters or on the mental level of its functional architecture. We do not simply claim that pain was nothing but C-fibre firing, we can *explain* why it is: Because it serves the relevant causal function. And third, we get multiple realisability for free. Mental states can be implemented in different physical substrates just as the same program can be implemented in different hardware. Just as the Word program can be run on Mac as well as on a PC, pains can be implemented in mammal brains or mollusc brains. We might even think of more extravagant hardware such as artificial neural networks or extraterrestrial brains. Any system that is capable of implementing the right kind of computation will do. (see Rescorla 2015: 8-31)

We can, thus, sum up the main thesis of strong AI:

Strong AI: Implementing the right kind of computation suffices for having a mind.²

Any system that implements the same computation as my brain, therefore, must have the same mental states as I do, be it another human brain or a silicon circuit. "On the Strong AI view, the appropriately programmed digital computer does not just simulate having a mind; it literally has a mind." (Searle 2004: 46) However, we still need to further clarify the notion *computation*, *implementation* and what the *right kind* of computation might be.

A usefull tool to formalise the concept of computation is the *Turing machine*. A Turing machine is, contrary to what its name might suggest, not a real machine but a mathematical model introduced by Alan Turing in his 1936 paper *On computable numbers, with an appli-*

²Strictly speaking strong AI contains the additional claim that computation serves as an explanatory framework for cognition, i.e. the fact that a system implements the right kind of computation *explains* why it has the mental states that it does. The Chinese room argument is directed against both claims. (see Searle 1980: 417) Since falsifying one of the two claims is enough to falsify the conjunction of both and in order to keep things simple we are going to focus on the sufficiency claim, however, keeping in mind that computationalism also aims at *explaining* cognitive phenomena.

cation to the *Entscheidungsproblem* (Turing 1936). Turing's goal was to formalise the notion of computation and what it means for a function to be computable. Unlike today, in Turing's times computations were mostly performed by humans. Those human computers worked in the government, in business or in research establishments and used so called *effective methods* in order to calculate functions. An effective method is a finite set of rules that, if carried out correctly, yields the desired result in a finite number of steps. It was this kind of human computation by means of effective methods that Turing had in mind. (see Copeland 1997: 1-7) Turing's thought was that, due to the rule full structure of effective methods, it must in principle be possible to design a machine that performed the necessary computations, i.e. goes through the same steps as a human computer. The Turing-machine can be understood as exactly such a mechanical computer³. It consists of a read-write head and an infinitely long tape divided into squares. Any square contains exactly one or no symbol and the head is always aligned over a field. It can perform exactly three possible actions: (1) Write sign S , move to the right and transition to state q , (2) write sign S , move to the left and transition to state q , or (3) write sign S , do not move and transition to state q . Which of the three actions are carried out, which symbol the machine writes and to which state it transitions is governed by the so called *machine table* or *program*. The program states what the read-write head will do next given the state it is in now and the sign it reads. When computing a function f it takes the input x encoded in the symbols of the tape and goes through a series of steps according to the program in which it manipulates the symbols on the tape. The machine halts if it enters a halting state and the symbols on the tape are the output $f(x)$. Any function that can be computed by means of an effective method, i.e. by a human being following a finite set of rules, can also be computed by a Turing machine. This claim is called the *Church-Turing thesis*. (see De Mol 1999: 5-10) *A computation, therefore, is an ordered set of symbol manipulations according to a set of well defined rules that transforms one or more inputs into one or more results.*

Before moving on, let's note some parallels between Turing machines and computationalism which make the idea that the mental is essentially computational particularly interesting. Just like in the functionalistic understanding of mental processes, a Turing machine is not defined only via its input-output relation but also via the states it transitions through in the course of the computation. According to functionalism, the mental processes of a person who has been hit on the hand with a hammer should be reconstructed as follows: The person receives as an input a certain tissue damage, which in turn causes a belief that she has been hit (mental state 1) along with a desire to relieve the pain (mental state 2) which finally yields the behaviour (output) of searching a cooling pad. This is analogous to the functioning of a Turing machine which takes an input, goes through a series of states and state transitions and finally yields an output. (see Rescorla 2015: 2-8)

³Where the expression "mechanical" should not be understood as referring to a physical device but to the rule-following character of the Turing-machine.

But a Turing machine, just as any other model of computation, is an abstract object, brains and computers on the other hand are physical devices. We therefore need some kind of nexus between abstract computational models and concrete physical systems. This nexus is provided by the concept of *implementation*. A physical system can be said to implement a computation if and only if for any abstract state (e.g. the state of a Turing machine) there is a physical state (e.g. the state of the hard drive of a digital computer) and the state transition of the physical system mirrors that of the abstract model. Implementation describes an isomorphism between the formal structure of the computation and the causal structure of the physical system. David Chalmers gives the following definition:

A physical system implements a given computation when there exists a grouping of physical states of the system into state-types and a one-to-one mapping from formal states of the computation to physical state-types, such that formal states related by an abstract state-transition relation are mapped onto physical state-types related by a corresponding causal state-transition relation. (Chalmers 1994: 392)

The property of implementing a computation, thus, specifies conditions strict enough such that not *any* system can be said to implement any computation but wide enough that there can be a variety of physical systems that implement the same computation, e.g. human brains, digital computers, human computers, etc. Two systems, then, implement the same computation if for both systems there is a mapping from physical states of the system onto the computational states of the model and if the physical states evolve in accordance with the state transitions of the computation. This not only specifies an isomorphism between each of the systems with the computational model but also between the *physical systems themselves*. Two system that implement the same computation share a common *causal organisation*, i.e. their components interact according to a shared causal pattern. Not only do the states of the systems mirror the states of the computation, but both physical systems evolve in accordance with each other. "What do all implementations of a given computation have in common? Precisely their causal organization." (Chalmers 1994: 401). According to computationalism, it is this causal structure that determines whether a physical system has a mind or not. A system that implements the same computation as my brain has the same causal organisation and, therefore, the same mental properties as myself. The abstract notion of computation thus specifies the causal organisation of a physical system by way of the latter implementing the former.

Having clarified the notions of computation and implementation we need to discuss what it means to implement *the right kind* of computation. The question, however, which computational model is most appropriate for describing the mind goes beyond the scope of philosophy.

Some have argued that the mind is, or at least is similar, to a Turing machine. (see Putnam 1967) Others found that a more brain-like model was better suited. (see Churchland 1989) In order to stay as general as possible we are not going to commit ourselves to any of the specific forms of computationalism but rather consider the fundamental ideas shared by all those positions. According to Chalmers the claim of computationalism "is simply that some computational framework can *explain* and *replicate* human cognitive processes." (Chalmers 2011: 350) All computationalist theories share the fundamental view that computation is sufficient for mentality.

This follows from the facts that computation captures the general patterns of causal organisation and that mentality is something Chalmers calls an "organizational invariant".

Call a property P an organizational invariant if it is invariant with respect to causal topology: that is, if any change to the system that preserves the causal topology preserves P. The sort of changes in question include: (a) moving the system in space; (b) stretching, distorting, expanding and contracting the system; (c) replacing sufficiently small parts of the system with parts that perform the same local function (e.g. replacing a neuron with a silicon chip with the same I/O properties); (d) replacing the causal links between parts of a system with other links that preserve the same pattern of dependencies (e.g., we might replace a mechanical link in a telephone exchange with an electrical link); and (e) any other changes that do not alter the pattern of causal interaction among parts of the system. (Chalmers 2011: 337-338)

Most properties are not organisationally invariant, they depend essentially on features that are not part of the causal organisation.

Flying depends on height, digestion depends on a particular physiochemical makeup, tubes of toothpaste depend on shape and physiochemical makeup, and so on. Change the features in question enough and the property in question will change, even though causal topology might be preserved throughout. (Chalmers 2011: 338)

There are, however, certain properties that are organisationally invariant. The property of implementing a computation, for example, does not depend on anything but the causal dependency relations of parts of the system. Moving a computational system in space, stretching it or replacing causal links with causally equivalent other links is not going to change the computation carried out or making it a non-computable system all together. (see Chalmers 2016: 40) The question at hand now is: Are mental properties organisational invariants? Chalmers justifies his affirmative answer with the argument from *fading* and dancing qualia: Imagine a healthy human brain being step by step replaced by silicon chips that locally perform the same input-output function as the original parts of the brain. After some time the entire brain has

been replaced by a silicon circuit that, by definition, has the same functional architecture, i.e. the same causal topology, as the original brain. Since the input-output relations have been preserved throughout the replacement process, the behaviour of the person in question would be the same with the silicon as with the biological brain. This simply follows from the claim that the functional architecture has been preserved. Now, Chalmers argues: Imagine mental properties were not organisationally invariant. That would mean that during the replacement process they would either at one point suddenly disappear or gradually *fade*. However, *without having any impact on the subject's behaviour*. Chalmers considers this highly implausible and argues that we would expect the gradually (or abruptly) going blind, deaf and numb to have at least *some* impact on the person's behaviour. Even more so if we imagine that the original brain had not been destroyed in the replacement process. We then could build in a switch that switches back and forth between the silicon and the biological brain. If mental properties were not organisationally invariant, i.e. if the silicon and the biological brain had different mental properties, flipping the switch on and off would cause the mental experience to *dance* before the eyes. Again however, *without having any impact on the outward behaviour*. It then implausible that a person who experiences such dancing qualia would act as if nothing had happened. (see Chalmers 1995: 309-328) Chalmers arguments render it unlikely that two systems with the same causal organisation could have different mental experiences. "If this is right, we can say that consciousness is an organizational invariant: that is, systems with the same patterns of causal organization have the same states of consciousness, no matter whether that organization is implemented in neurons, in silicon, or in some other substrate." (Chalmers 2016: 40) In some sense the fading and dancing qualia arguments are the functionalistic counterpart of the CRA. Both compare two systems with the same causal organisation, the fading and dancing qualia argument making it plausible that they must have the same mental properties and the CRA appealing to our intuition that they don't.

Computationalism is, therefore, particularly well suited for the purpose of AI. If mentality and cognition are essentially computational, it does not matter whether the relevant computation is carried out in neuronal "wetware" or in the silicon hardware of a digital computer. According to computationalism, finding a mapping from the physical states of the computer to the computational states of the mental model is sufficient for the computer to have mental states. But this means that all we need to do is *simulate* a human brain up to a sufficient degree of coarse graining. The simulation, if success full, computes the same function as the original brain and following the claim of strong AI, therefore, has the same mental states.

[I]f a property is an organizational invariant, we should expect it to be preserved in a computer simulation (a simulated computer is a computer). So given that consciousness is an organizational invariant, we should expect a good enough computer simulation of a conscious system to be conscious, and to have the same sorts

of conscious states as the original system. (Chalmers 2011: 40)

This claim deserves some emphasis. Computer simulations are used in all kinds of areas from engineering to finance. But unlike simulations of plains or stock markets *the simulation of a mind actually is a mind*, according to computationalism. No one would claim something similar about a plane simulation. This follows from the belief that mental properties are organisationally invariant and the property of flying is not.

Digital computers are particularly well suited for simulation tasks since they are *universal*, i.e. they can compute any function that is computable.⁴ A digital computer, therefore, can simulate any (computable) process up to arbitrary good approximation. It is for this reason that digital computers still play a key role in artificial intelligence: *What ever the relevant computations of the mind might be, it is possible to program a digital computer such that it simulates them up to an arbitrary degree of precision.* It is no wonder, therefore, that the setup of the Chinese room resembles the architecture of digital computers in several ways.

But before turning to the CRA, let's sum up the key ideas of strong AI: According to functionalism, mental states are individuated by their causes and effects, i.e. by the causal role they play in an functional architecture of an agent. Having the right kind of causal organisation, whatever that might be, therefore, suffices for having a mind. In addition to that Computationalists claim that the causal organisation is best captured computationally. All systems that implement the same computation share a certain causal pattern. Therefore, implementing the right kind of computation suffices for having the right kind of causal structure and thus for having a mind. It is this claim of *computational sufficiency* that lays at the hart of Strong AI and that the CRA is directed against.

II The Chinese room argument

"If what has gone before is correct, this establishes the thesis of computational sufficiency, and therefore the the view that Searle has called "strong artificial intelligence": that there exists some computation such that any implementation of the computation possesses mentality." (Chalmers 2011: 341) The Chinese room argument⁵ (see Searle 1980: 417-419) is directed against the claim of strong AI: That implementing the right kind of computation suffices for having a mind. The CRA is build around a thought experiment: Imagine Searle, who does not speak Chinese, inside a room which contains nothing but boxes full of paper, a desk with a pen and a rule book. (see fig. 1)

⁴Where the term "computable" needs to be understood in its strict sense, i.e. as *computable by effective methods* or equivalently as *Turing computable*.

⁵The explications in this section regarding the CRA are taken from Searle's original paper (1980) along with remarks in (Searle 1990a, 1990b, 1993, 2004).

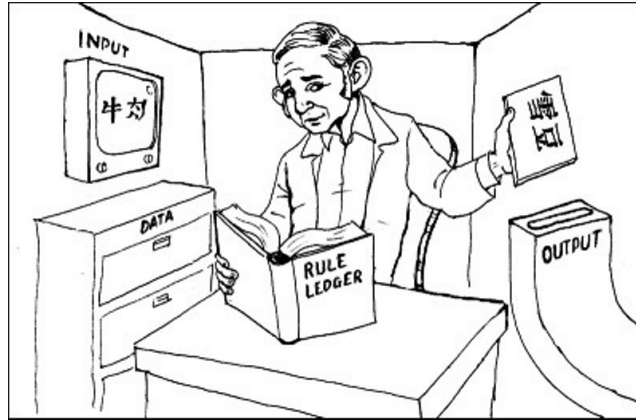


Figure 1: Searle in the Chinese room manipulating symbols. (Novella: 2015)

Through a screen on the wall Searle receives Chinese symbols which he does not understand and is supposed to return cards with symbols he does not understand either through a mail slot in the wall. The rule book, however, is written in English and states a set of rules which tell Searle exactly what symbols to return upon receiving a certain input from the screen. So, when the screen shows a particular sign Searle takes his rule book and looks up which symbol he has to write on the card and pass through the mail slot.⁶ Unknown to Searle the symbols he was receiving were questions written in Chinese and the cards he was returning the appropriate answers.

This means that from an outside perspective it looks as if Searle indeed was having a conversation in Chinese. He gave the same answers that a native Chinese speaker would have given, i.e. Searle was passing the *Turing test* for speaking Chinese. The Turing test in his original form was first introduced by Turing in 1950 in the context of machine intelligence. (see Turing 1950) Turing found the question whether machines could think too meaningless to deserve discussion and proposed to replace it with the more precise question whether a machine could do well in the *imitation game*. The imitation game, as Turing called the test later named after himself, goes as follows. Suppose there is a person, a machine and an interrogator. The interrogator is located inside a room separated from the machine and the person such that she cannot tell which is which. Her objective is to determine which is the machine and which is the person. All she can do to find out is ask questions like: "Could test subject 1 please tell me whether he likes to play tennis?" or "Which is test subject 2's favourite colour?" The machine is designed to imitate a human person in all these respects. After a certain amount of time the interrogator has to decide which of the two test subjects is the machine and which the person. If she miss labels them or is unable to decide, the machine has passed the Turing test.⁷ (see Oppy and Dowe 2003: 1-3)

⁶In the original thought experiment Searle only talks about "batch[es] of Chinese symbols". (Searle 1980: 418) The use of an input screen and a mail slot were introduced by myself for illustrational purposes as in (Novella: 2015)

⁷The claim that passing the Turing test is equivalent to being intelligent or to the possession of mental features

The rule book in the Chinese room is written such that Searle always replies with the appropriate Chinese symbols. From an outside perspective the Chinese room is, thus, indistinguishable from a Chinese native speaker and an interrogator would be unable to reliably decide whether she was talking to a native speaker or not. Therefore, Searle passes the Turing test for speaking Chinese *without understanding what the symbols mean*. Unlike a native speaker Searle lacks the mental feature of understanding.

But how is this related to computationalism? The Chinese room thought experiment is designed such that Searle implements the same computation as the brain of a native speaker. *Whatever the computations are that are relevant for speaking Chinese Searle's rule book is written such that he implements them*, i.e. he goes through the same series of steps. According to computationalism both systems, the Chinese room and the brain of the native speaker, therefore, must have the same mental features. But according to Searle this is obviously false. *Unlike the Chinese native speaker Searle lacks the mental property of understanding*. The Chinese room implements the computation for understanding Chinese, however, it lacks the corresponding mental feature. Thus, implementing the right kind of computation, is *not* sufficient for having a mind. There are at least some mental features that are not grounded in the causal architecture of the system, understanding Chinese is one of them.

All Searle does in the Chinese room is manipulate symbols that are completely meaningless to him. Searle is not able to *refer* to the outside world, his symbol manipulations are not *about* anything. This shows that computation alone is not able to account for the mental feature of *intentionality*. Intentionality is the capability to *represent* or *to be about* states of the world. "In philosophy, intentionality is the power [...] to be about, to represent, or to stand for, things, properties and states of affairs." (Jacob 2003: 1) The English sentence "Dresden is the capitol of Saxony." for example refers to a city in Germany and its being a capitol of a Bundesland called "Saxony". But the sentence itself, i.e. the series of words written in letters of the Latin alphabet, carries its meaning only because *I used it* to express something. If a cat were to jump on the keyboard of my laptop and by chance type the words "Dresden is the capitol of Saxony" this sequence of letters would not refer to the city of Dresden. Sentences along with sings, utterances, symbols, emojis etc. only have *derived intentionality*, they only are about anything because they are used to express a thought, a belief, a worry, etc. Mental states on the other hand have *intrinsic intentionality*. The intentionality of my *belief* that Dresden is the capitol of Saxony is not derived from something else, it is intrinsic to the belief itself.

Since sentences - the sounds that come out of one's mouth or the marks that one makes on paper - are, considered in one way, just objects in the world like any other objects, their capacity to represent is not intrinsic but is derived from the

was not intended by Turing himself.

Intentionality of the mind. The Intentionality of mental states, on the other hand, is not derived from some more prior forms of Intentionality but is intrinsic to the states themselves. An agent uses a sentence to make a statement or ask a question, but he does not in that way use his beliefs and desires, he simply has them. (Searle 1983: vii)

The CRA, now, shows that computation alone, i.e. a series of symbol manipulations, cannot account for intrinsic intentionality. Many mental states, however, are essentially intentional. The belief that Dresden is the capitol of Saxony, is is the belief that it is because it's about Dresden, Saxony and the former being the capitol of the latter. The hope that it will rain later is essentially about a certain weather phenomenon, a thirst for water is essentially about H_2O , my regret of having stepped on your foot is essentially about your foot and me having stepped on it and so on. Arguably, there are certain non-intentional mental states like feelings of anxiety or joy that are not directed at anything in particular. But if computation is not able to account for intentional mental states it certainly is not sufficient for mental states in general.

On a standard view computation is defined syntactically as a series of symbol manipulations. A Turing machine for example computes by erasing and writing numbers on a tape. In a digital computer symbols are encoded in bit strings implemented in varying voltage levels and operated on by a central processing unit (CPU). In many respects the Chinese room is analogous to a standard digital computer. (1) Both operate over symbols. The symbols in the Chinese room are Chinese letters written on cards or displayed on a screen respectively. The symbols that a digital computer uses are encoded in strings of bits which again are implemented in varying levels of voltage. (2) Both have a central processing unit. It is one physical object that carries out the computations. In the case of the Chinese room it is Searle, in the case of the digital computer it is the CPU. (3) Both operate according to a given set of rules. In the digital computer those rules are stated in the program, in the Chinese room the rules are written in the rule book. Digital computers, just as Searle in the Chinese room, carry out computations by way of manipulating symbols. "As far as the Chinese is concerned, I simply behave like a computer; I perform computational operations on formally specified elements." (Searle 1980: 418)

Searle summarises the idea that computation is unable to account for intentionality under the slogan *syntax is not semantics*. The CRA shows that manipulating symbols according to syntactic rules is not sufficient for attaching meaning to them, for making them about anything. And here is the difference between the Chinese room and the native speaker: Both use the same symbols and manipulate them according to the same syntactic rules, *but for Searle they have no meaning, they lack semantic content*. The CRA shows that implementing a certain computation, i.e. manipulating symbols according to syntactic rules, is not sufficient for attaching meaning to them. Mental processes have intrinsic intentionality, symbols on the

other hand only are about anything if they are being used by someone to refer to states of the world.

The CRA, thus, can be summarised as follows:

- (1) Computation is defined purely formally (syntactically).
- (2) Mental states have content (semantics).
- (3) Syntax alone is neither constitutive nor sufficient for semantics.
- (C) Therefore, implementing the right kind of computation is neither constitutive nor sufficient for having a mind. (see Churchland and Churchland 1990: 33)

Searle does not argue for the first two premises but assumes them to be uncontroversial. The Chinese room thought experiment is presented in order to support premise (3) and the conclusion is the anti-thesis of strong AI.

In order to avoid some confusion I next want to mention a couple of theses that are *not* contested by the CRA. (1) Computers or machines in general cannot have a mind. This claim actually is trivially false, since human brains are biological machines and can be understood as computers. The CRA only implies that appropriately programmed computers must not *necessarily* have mental states. (2) Only biological systems can have a mind. Although Searle seems to favour such a view it is not supported by the CRA. All the CRA suggests is that implementing the right kind of computation alone is not sufficient for having a mind. Thus, there is room for artificial intelligence. Computation alone might not be a sufficient condition for the possession of a mind, but it might well be a necessary one. For example, implementing the right kind of computation *in the appropriate hardware*, might be sufficient for mental features.⁸ Or, implementing the right kind of computation *and being embedded in the environment in the correct way* might suffice for having a mind. Strictly speaking, those possibilities are not excluded by the CRA. (3) An appropriately programmed computer cannot have a mind. As already mentioned, human brains can be understood as appropriately programmed computers, thus, making the claim trivially true. The thesis of strong AI is not that appropriately programmed computers can have mental states, that there is some undiscovered feature that explains their mental properties, but rather that they *must be thinking* because they implement the right kind of computation. The CRA is directed solely against the claim of strong AI as stated here. (see Searle 1990a: 27)

Next, I want to discuss a couple of objections that are commonly raised against the CRA and that have already been mentioned by Searle in his original paper.⁹ (see Searle 1980:

⁸In some sense this claim is trivially true. If we understand "the right kind of computation" as "the computation that a human brain implements" and "the appropriate hardware" as "the human brain" it means no more than that human brains have mental states.

⁹Searle considers three further replies which we are going to omit here for reasons of space and because they are not relevant for the further discussion.

419-424)

I. The systems reply: While it is true that a single person shuffling symbols is not going to understand Chinese, this is not what computationalists claim. Searle is only a part of the computer, i.e. the CPU, and no one claims that CPUs alone can understand. It is the entire Chinese room with all the cards and boxes, with the rule book and the scratch paper that implements the computation, not the person inside it alone. Strong AI does not ascribe understanding to Searle, but rather to the entire system of which he is merely a part.

Searle's answer to the systems reply is twofold. First, he considers the idea that a combination of a person together with some sheets of paper and a pencil might be said to understand Chinese implausible. "It is not easy for me to imagine how someone who was not in the grip of an ideology would find the idea at all plausible." (Searle 1980: 419) If anything in the Chinese room might be even a candidate for understanding it must be Searle himself. Second, Searle claims that he can get rid of everything in the Chinese room except himself without altering the argument. Imagine that Searle had memorised his rule book as well as all the symbols he had received so far, that he would do all the computations in his head and that he would work outdoors. "All the same, he understands nothing of the Chinese, and a fortiori neither does the system, because there isn't anything in the system that isn't in him." (Searle 1980: 419)

II. The robot reply: Suppose we put a computer inside a robot and connect it to the environment via sensors and cameras that enable it to perceive and via actuators and mechanical limbs that enable it to act. Such a robot would not only manipulate formal symbols but eat, sleep, move around and act in any other way just like a human. If such a robot were to engage in a conversation with a Chinese person and pass the Turing test it would have genuine understanding. The CRA only focuses on the computer, i.e. the brain and leaves the other relevant parts unconsidered.

Again Searle provides two answers. First, he points out that the robot reply goes beyond the claim of strong AI. It is not solely the fact that a system implements a certain computation that accounts for its mentality. It also has to be connected to the environment in the appropriate way. The robot reply adds some kind of causal connection to the environment to the conditions for having mental features. But, secondly, "the answer to the robot reply is that the addition of such "perceptual" and "motor" capacities adds nothing by way of understanding, in particular, or intentionality, in general to [the] original problem." (Searle 1980: 420) Suppose a large enough robot that has inside his head not a computer but the Chinese room. Just like in the original thought experiment Searle receives Chinese symbols, manipulates them according to a rule book and feeds back different Chinese symbols to the outside. The claim that some of the input symbols might be coming from a camera or that some of the outgoing symbols might be fed into an actuator adds nothing to the original thought experiment. All

Searle does is shuffling symbols that carry no meaning to him.

III. The brain simulator reply: Suppose we wrote a program that does not operate on Chinese symbols itself and defines which answers to give when receiving certain questions. But rather, imagine a program that simulates the actual brain processes of a person engaging in a Chinese conversations, i.e. the sequences of neurons firing, neurotransmitters being released, synaptic weights adjusted etc. Such a system would not only reproduce the input-output behaviour of the Chinese person but simulate the information processing of her brain down to the neuronal level. If we refuse to attribute understanding to such a system how can we say that the Chinese person understands, given that both process information in exactly the same way.

Searle's comeback is the famous water pipe-brain thought experiment.

[I]magine that instead of a monolingual man inside a room shuffling symbols we have the man operate an elaborate set of water pipes with valves connecting them. When the man receives the Chinese symbols, he looks up in the program, written in English, which valves he has to turn on and off. Each water connection corresponds to a synapse in the Chinese brain, and the whole system is rigged up so that after doing all the right firing, that is after turning on all the right faucets, the Chinese answers pop out at the output end of the series of pipes.

Now where is the understanding in this system? It takes Chinese as input, it simulates the formal structure of the synapses of the Chinese brain, and it gives Chinese as output. But the man certainly doesn't understand Chinese, and neither do the water pipes [...]. (Searle 1980: 421)

Even a perfect simulation of the human brain still is just a simulation. Solely recreating its abstract causal relations is not sufficient for the possession of a mind.

Next, I want to consider an argument that is closely related, however, different from the CRA: *The argument from observer relativity of computation*. Searle thinks that from multiple realisability follows *universal* realisability, i.e. the claim that a given computation can not only be implemented in a variety of different hardware but in fact *in any physical system*. (see Searle 1990a: 26-28) He argues that any object counts as a digital computer that admits the assignment of 0s and 1s. That follows from the claim that computation is multiple realisable. Nothing about bits or bit strings makes it necessary that they be implemented in voltage levels. Wholes in a paper cards, magnets being aligned in certain directions or levers being in an upward or downward position would work just as well. But then again any physical object admits for such a interpretation. We could interpret a cup as a 0 when it is whole and as a 1 when it is broken. Or we could interpret a plane as a 0 when it is in the air and as a 1

when it is on the ground. Basically any object would do the job. Thus, any object that is sufficiently complex can be interpreted as implementing any computation. All that is required is an interpretation of the object under which its states are isomorphic to the formal states of the computation.

For any program there is some sufficiently complex object such that there is some description of the object under which it is implementing the program. Thus for example the wall behind my back is right now implementing the Wordstar program, because there is some pattern of molecule movements which is isomorphic with the formal structure of Wordstar. But if the wall is implementing Wordstar then if it is a big enough wall it is implementing any program, including any program implemented in the brain. (Searle 1990a: 27)

I think Searle is wrong here. I do not think that it is possible to find a set of a wall's physical states, e.g. phase space states of molecule motion, that are isomorphic to and evolve in accordance with the computational states of the Wordstar program. Chalmers estimates the chance of an arbitrary system implementing a given complex computation to be significantly lower than one to $(10^{1000})^{10^{1000}}$. (see Chalmers 1994: 396) Universal realisability does not follow from multiple realisability.

However, I think, Searle is right in pointing out that *computation is an observer relative notion*. "Computational states are not *discovered within* the physics, they are *assigned* to the physics." (Searle 1990a: 27) Nothing about the intrinsic physical features of a PC, like mass, voltage, momentum etc. make it a computer. It is us who interpret it as such. It is us who assign 0s and 1s to certain voltage levels. Chalmers admits this: "In general, there is no canonical mapping from a physical object to "the" computation it is performing. We might say that within every physical system, there are numerous computational systems. To this very limited extent, the notion of implementation is "interest-relative"." (Chalmers 1994: 397) Mental features on the other hand are not observer relative but intrinsic properties of a given system. The question whether a certain human being, a lower animal or a digital computer has mental states is not a matter of interpretation. Either it has beliefs, desires, pains, etc. or it hasn't. To put it crudely: If by tomorrow there were no human beings computers would cease to exist, but animals would still feel pain.

We can turn this into an argument against strong AI. The claim that computation is sufficient for mentality is strictly speaking not just false, *it is ill defined*. There is no way we could *discover* that implementing a computation suffices for having a mind, because we could never *discover* that a system implements a computation in the first place. We only can *ascribe* a computation to a system. "The question "Is the brain a digital computer?" Is as ill defined as the questions "Is it an abacus?", "Is it a book?", "Is it a set of mathematical formulae?" (Searle 1990a: 35-36) When we are claiming that the brain was a computer of some sort we

are committing a fallacy: Without a homunculus inside our brain using it to compute there is not even a sense in which computation might be sufficient for the possession of a mind.¹⁰

III Spiking neural networks

Artificial neural networks (ANNs)¹¹ are computational devices that are inspired by the neuronal structure of human brains. They consist of interconnected nodes called "neurons". (see figure 4) The strength of the connections, or "synapses" regulates flow of information from one neuron to another. ANNs are used in various fields of machine learning and artificial intelligence. In cognitive science ANNs are usually referred to as *connectionist systems*. The explanations of this section largely follow (Hertz et.al. 2018).

The first model of an artificial neuron was presented in 1943 by Warren McCulloch and Walter Pitts. The so called McCulloch-Pitts cell is a highly simplified neuron model and operates on digital signals only. It takes a number of binary inputs, adds them up and returns a 1 if the sum exceeds a certain threshold and a 0 if it doesn't. (see McCulloch & Pitts 1943) A logical AND function, for example, can be described by taking two inputs and setting the threshold to 2. If both inputs are TRUE, i.e. equal to 1, the threshold is reached and a 1 is being returned, in all other cases the neuron returns a 0. (see figure 2) The synapses in

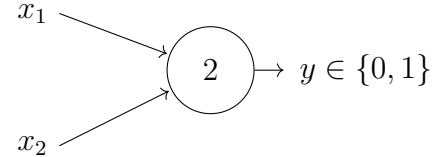
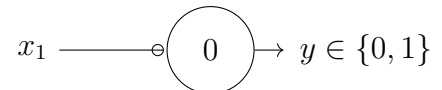


Figure 2: McCulloch-Pitts neuron computing the logical AND function.

the McCulloch and Pitts model can be either excitatory or inhibitory. That means that the input signal is multiplied by either $+1$ or -1 before being fed into the neuron. A inhibitory synapse is represented with an arrow with a circle. The logical NOT function, therefore, can be represented as seen in figure 3. The input is inhibited, i.e. multiplied by -1 and then compared to the threshold. When $x_1 = 0$ the threshold is reached and the neuron returns a 1. When $x_1 = 1$ the inhibited value is smaller than 0 and the neuron outputs a 0. The

Figure 3: McCulloch-Pitts neuron computing the logical NOT function.



¹⁰There is, however, a way around this argument. If we understand "implementing a computation" not as "being interpreted as a computer" but as "being *interpretable* as such" we might say that the brain computes a certain function. No homunculus needs to constantly observe it, there only needs to be a mapping from physical states to computational states that we potentially *could* assign to the system. In that sense computation *is* intrinsic to physics. Chalmers (1994: 399) argues for something along those lines. But this seems highly anthropocentric: Out of all the physical and bio-chemical processes that might account for mentality it is the property of *interpretability by human beings* that gives rise to consciousness and intentionality? To me this seems highly implausible.

¹¹We are later going to use the term "ANN" to stand for standard non-spiking neural nets as opposed to SNNs

combination of a NOT function and one that computes an AND function is called a NAND gate. Since NAND gates are universal, any Boolean function can be expressed by a series of interconnected McCulloch-Pitts neurons. (see Bayetkin & Akkaya 2000: 1)

Neural networks operating on digital signals and using step wise threshold functions are commonly referred to as *first generation* neural nets. Most ANNs used today are networks of the *second generation*. They operate on *analogue* in- and outputs, i.e. on real valued signals and use non-linear but smooth functions. Those functions are used to model the so called *membrane potential*. This term is derived from neuro-biology where it describes the activation of the neuronal membranes that can be excited to different degrees. Membrane potentials of artificial neurons are implemented in the voltage level of electronic components. Also the inhibition/excitation value of the synaptic strength can take any real number, not just +1 and -1. However, the basic idea stays the same: Information is processed via nets of neurons interconnected with synapses. Often ANNs have large numbers of computational elements, i.e. neurons and synapses. Each neuron can process information independently from and simultaneously with other neurons. Therefore, ANNs are said to be *parallel processing*, i.e. they perform a number of different computations at the same time. All ANNs have an input layer where the signals are fed into the system and an output layer from which the results of the computation is read out. If a system has additional layers in between, so called *hidden layers*, we call it a *deep neural network*. Figure 4 shows a deep neural net with three hidden layers. The input is encoded in the activation rates of the input neurons. This means that

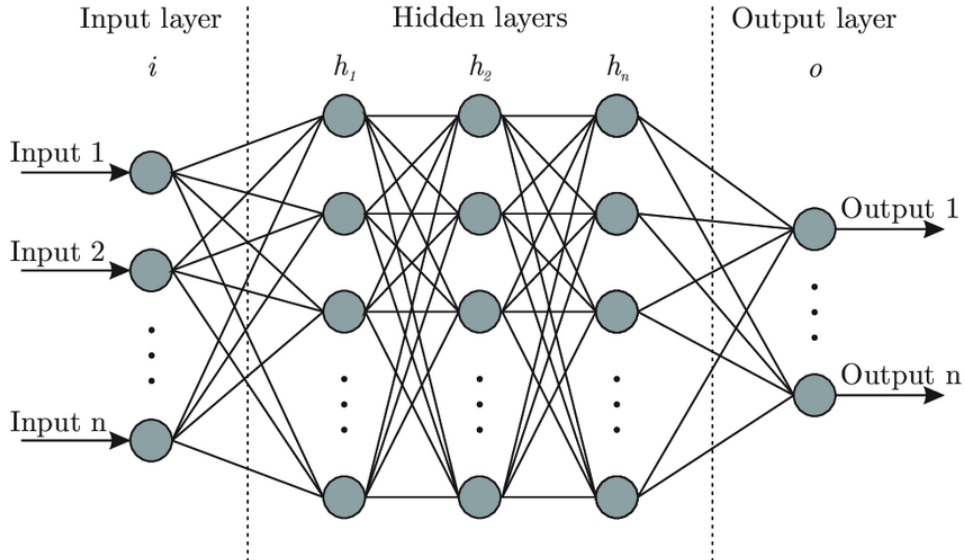


Figure 4: Schematic drawing of a deep neural network. (Bre et.al. 2018: 4)

ANNs, unlike digital computers, do not operate on bits and bit strings but on *patterns of activation*. A standard task for ANNs is hand written digit recognition. The handwritten sample is fed into the system by taking a picture of it and dividing the picture into pixels.

Then, one assigns a number to each pixel which represents its brightness. This number is then encoded in the activation level of one neuron. Thus, the entire picture of a 28×28 pixel picture is encoded in the activation levels of 784 neurons. (see figure 5) In neural networks

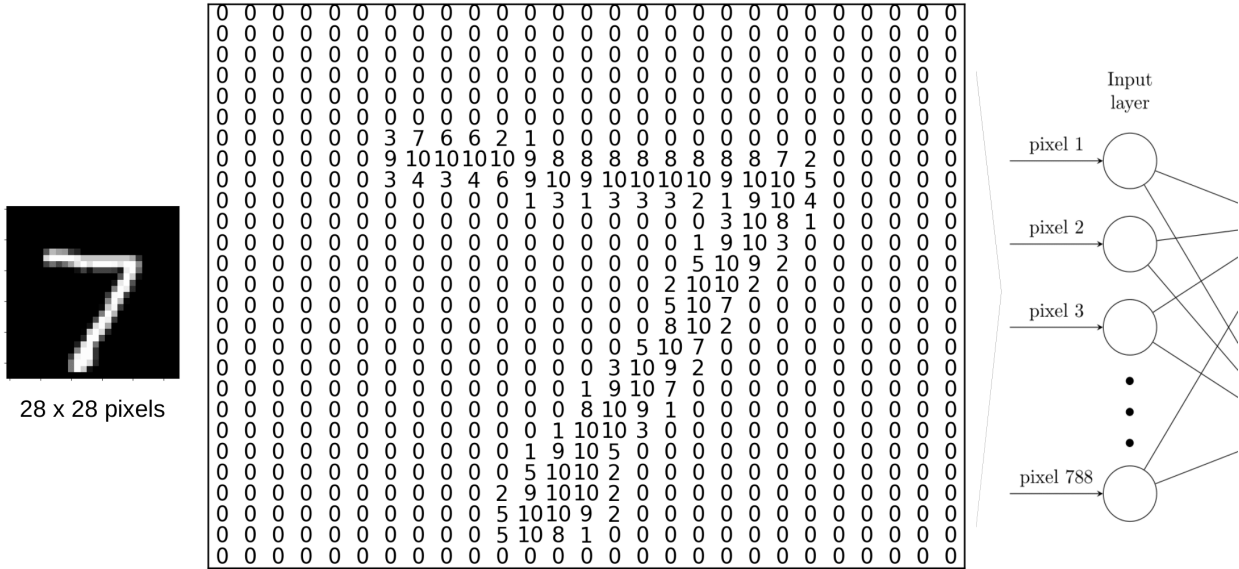


Figure 5: Encoding a handwritten digit into a neural network. Example taken from the MNIST data base.

the input-output relation is not determined by a program. There are no explicitly stated rules which tell the system to respond with a certain output to receiving a given input. Rather, *the input-output relation is determined by the synaptic weights*. Given an input, the activation value of each subsequent neuron is determined by the strength of the synaptic connections, they determine which activation levels arrive at the output layer. This makes artificial neural nets particularly well suited for *machine learning* tasks.

ANNs do not have a "hard-coded" program, that connects input and output. Rather, they *learn* the function they compute by iteratively updating the synaptic weights. To illustrate this, let's again consider the network for digit recognition. The output layer of such a system usually consist in ten neurons each standing for a number between zero and nine. If the third neuron has the highest activation level, the system has recognised a 3, if the fifth neuron is the one with the highest activation it has recognised a 5 and so on. In the example of figure 5 the output layer should consist of nine neurons with a low activation level and one, the seventh, neuron with a higher activation. This is what we would expect for a trained system. Before training, however, the synaptic weights are randomly initialised and the output of the system is more or less random. The system sees a three and outputs a one, it receives a five and outputs a zero. We therefore need to train it, such that it actually recognises the digits it is presented with. This happens by first giving the system some sort of feedback whether it's performance

was good or poor. This is achieved by defining a so called *error function*. The error function is a measure of how close the systems output is to the desired result. For example, we could define the error function in the digit recognition scenario as the difference between the number that was fed into the system and the output it returned. If, for example, it is presented with a seven and it returns a seven the error is zero. If it returns a three, however, the error would be four. In this way we can define a value that stands for how well the system is performing. The aim is of course to *minimise* the error. This is done by updating the synaptic weights such that the output is closer to the desired result. Effectively, this yields an optimisation problem with the synaptic weights being the parameters and the error the quantity that is to be minimised. There are different ways in which this can be achieved. Presumably, the most widely used method for ANNs is *error back propagation*. However, back propagation is neither likely to be used in biological brains nor particularly well suited for SNNs. (see Tavanaei et.al. 2019: 49) For the sake of simplicity technical details are not being discussed here. What matters is that in the course of every training sequence we present the system with an input, evaluate its output by calculating an error and update its synaptic weights such that it minimises the error. After a certain number of feedback-update rounds the parameters have been adjusted such that the system is able to recognise the handwritten digits that it is presented with and *that it has not encountered during training*. ANNs are being used for various different machine learning tasks but the basic idea is always the same: Present the system with a certain task, give it some feedback on its performance and update the synaptic weights in order to optimise the result.

ANNs from the second generation are more biologically realistic than their predecessors. And "[t]he third generation of neural networks once again raises the level of biological realism by using individual spikes. This allows incorporating spatial-temporal information in communication and computation, like real neurons do." (Vreeken 2003: 2) Neural networks from the third generation are *spiking neural network* and, therefore, the main subject of the present thesis. Just like ANNs from the second generation were akin to first generation ANNs, SNNs are in many respects similar to ANNs and most of what has been said so far about second generation ANNs is also true for SNNs. However, there is one important difference: SNNs do not use static values of neuronal "membrane potential" to process information but the temporal pattern of *neuronal spikes*.

So, first of all, what is a spike? Biological brains use spikes to process information. Incoming signals to a neuron alter its excitation level and when it reaches a certain threshold the neuron emits an action potential as a sudden increase in voltage. Those short pulses are called *spikes*. Spikes then travel down the axons, i.e. the "arms" of a neuron. When they reach a synapse, i.e. the gap to the next neuron, those spikes induce a so called "post synaptic potential". These potentials can be either positive (in case of an excitatory synapse) or negative (in case

of an inhibitory synapse) and serve as the incoming signal for the next neuron. They travel to the centre of the neuron where they again alter its voltage, potentially causing it to spike itself. Each neuron in the human brain receives signals from roughly 10,000 synapses. Single neurons, thus, emit patterns of abrupt pulses and in this way use the precise timing of spikes to encode their messages to other neurons. (see Vreeken 2003: 2-3)

Just like real brains, SNNs use patterns of spikes, so called *spike trains* to encode information. A spike train is a set of times at that the neuron has spiked. It is, therefore, the timing, i.e. the "rhythm", of spikes, rather than their precise shape that is used for neural information processing. (see Ponulak & Kasiński 2011: 410) Figure 6 shows different examples

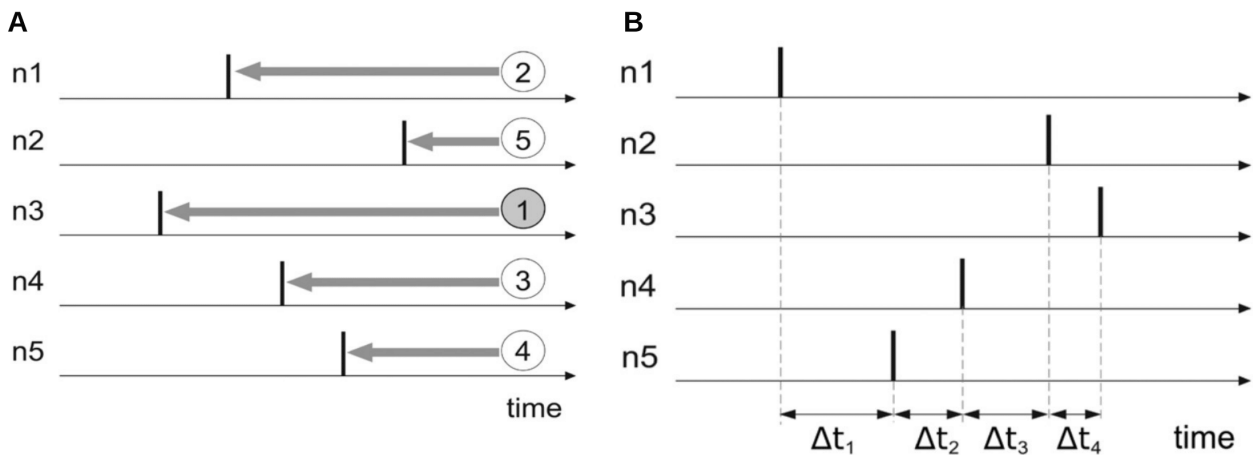


Figure 6: **A:** Rank order coding: Information is encoded in the order of spikes in the activity of different neurons. **B:** Latency coding: Neurons use the time interval between different spikes for information processing. (Ponulak & Kasiński 2011: 414)

of spike trains. The easiest method for encoding a number in the timing of spikes is the so called *frequency coding*. Here the number is encoded in the firing frequency of the neuron. If it spikes 137 times per second, for example, this encodes the number .137. Most encoding strategies, however, not only use the spike pattern of a single neuron but the relative spike times between different neurons. I want to give two examples of such coding strategies.

Rank order coding uses the order of spikes coming from different neurons of a population to represent data. (see figure 6A) Depending on which neuron spikes first, second and so on different symbols are encoded. Rank order coding assumes that each neuron emits only a single spike upon receiving an input. Ultra fast categorisation mechanism in primates visual system can be explained by rank order approaches. (see Ponulak & Kasiński 2011: 413)

Latency coding: Latency coding represents information in the precise timing of spikes from different neurons relative to each other. (see figure 6B) If one neuron spikes slightly earlier or slightly later, this changes the relative timing between neurons and thus the symbol that is being encoded. Latency coding is assumed to be used in a variety of different mechanisms in the brain. Especially brain like learning mechanisms rely heavily of the exact time intervals between different spikes. (see Ponulak & Kasiński 2011: 414)

The fact that different encoding strategies exist emphasises the observer relativity of computation. Depending on whether we use latency or rank order coding, or any other interpretation for that matter, one and the same pattern of spikes represents different data. We *interpret* the same system as encoding different information. What makes things even more complicated is that we do not necessarily have to use the same strategy for *encoding* and *decoding*. We might for example use latency coding to create spike trains that are fed into the input layer but rely on rank order coding to decode the output spikes. We can use different interpretations for one and the same system. There is nothing intrinsic to the spiking behaviour that would make it implement a certain computation and not another. We *chose* what the spike trains represent and, therefore, what computation the system implements.

To sum things up, SNNs are networks of interconnected units called neurons that use spiking patterns of single neurons or of populations of neurons to encode information. The input-output relation is governed by the strength of the connections, i.e. the synaptic weights. Training mechanisms optimise the systems behaviour by adjusting the connection strengths. Spiking neural networks are particularly bio-realistic and sometimes considered to be "the only viable option if one wants to understand how the brain computes." (Tavanaei et.al. 2019: 47)

Applicability to spiking neural networks

In the following chapter I am going to consider five differences between SNNs and digital computers: (1) massive parallelism, (2) Subsymbolic computation, (3) machine learning, (4) analogue representation and (5) temporal encoding. In each section I am first going to briefly examine some philosophical arguments why the respective difference might be an objection to the CRA. Then, I am going to discuss whether the CRA can be modified such that it takes into account those replies. My considerations are going to culminate in the *Morse code room argument*, a modification of the CRA such that it applies to SNNs. In the sixth section I am going to reconsider the systems, robot and brain simulator reply in the light of the newly developed thought experiment along with a new counter argument called the *luminous room*. I am going to finish with some more general remarks regarding computation and the CRA.

I Massive parallelism

SNNs often use large numbers of neurons. For example, the BioSpaun brain model by Eliasmith et.al. (2016) uses 2.5 million spiking neurons. As we have seen, every neuron can be regarded as a computational unit performing its own operations. Therefore, SNNs often are *massively parallel*, i.e. they consist of a large number of computational units that operate independently and simultaneously. Digital computers, on the other hand, usually consist of one CPU which computes *in series*, i.e. it executes different steps of a computation one after another, similar to a Turing machine that reads and writes one symbol at a time. Also the Chinese room consists of only a single person that carries out the computation. Just like a digital computer, Searle carries out the steps in the rule book one at a time.

This is why many people have argued that the CRA does not apply to parallel distributed systems like SNNs. Patricia and Paul Churchland (1990), for example, have argued that the CRA maintains its plausibility because it is one person doing the calculation. According to them, not only the correct input-output relation matters but also "how the input-output function is achieved [...]" (Churchland & Churchland 1990: 37) While they agree that the CRA renders implausible the idea that the brain is a digital computer, they think that parallel processing in connectionist networks might account for semantics. Although it seems intuitive that the Chinese room lacks intentionality, they argue that a system out of billions of people manipulating and interchanging Chinese symbols might well develop understanding. "[I]f

such a system were to be assembled in a suitably cosmic scale, with all its pathways faithfully modelled on the human case, we might then have a large, slow, oddly made but fully functional brain on our hands." (Churchland & Churchland 1990: 37)

Andy Clark (1991) argues that the brain thinks of its "variable and flexible substructures". (Clark as in Cole 2004: 33) Unlike digital computers, parallel distributed systems can be subdivided into smaller subsystems which perform computations on a lower level. The reason why digital computers, along with the Chinese room, are unable to account for cognition is not that computationalism is false, but that serial machines are unable to accommodate a fine grained model. Brains compute on a neuronal level and parallel computation comes a lot closer to being brain-like than serial machines. (see Cole 2004: 33-34)

Searle himself is aware of parallel processing and his first attempt to adapt the CRA is in the way mentioned above.

Imagine that instead of a Chinese room, I have a Chinese gym: a hall containing many monolingual, English-speaking men. These men would carry out the same operations as the nodes and synapses in a connectionist architecture [...], and the outcome would be the same as having one man manipulate symbols according to a rule book. No one in the gym speaks a word of Chinese, and there is no way for the system as a whole to learn the meanings of any Chinese words. (Searle 1990b: 28)

However, I agree with the Churchlands that in the case of the Chinese gym it becomes less obvious whether there can be any understanding present in any part of the system. The Chinese gym lacks the clear intuition of the Chinese room. While one person shuffling random symbols is most certainly not sufficient for understanding a large enough and heavily interconnected system like the Chinese gym might just as well understand Chinese as does the brain of a Chinese native speaker. With respect to the Chinese gym, the systems reply seems to gain plausibility.

However, the true reason why parallelism is not a threat to the CRA is because "the parallel, "brain like" character of the processing [...] is irrelevant to the purely computational aspects of the process. Any function that can be computed on a parallel machine can also be computed on a serial machine." (Searle 1990b: 29) Most processors can emulate a Turing-machine. This means that digital computers can compute any function that is computable and, in particular, any function that can be computed by a connectionist network¹². Actually, the vast majority of the neural nets that are use in practice are implemented in some digital hardware.¹³ Even large scale SNNs can be implemented on a system with a single processing unit. (see Nageshwaran et.al. 2009)

¹²Setting aside any potential super-Turing powers of connectionist networks. For further discussion see section IV.

¹³The reason for this is that implementing neural nets in digital hardware allows for greater flexibility regarding

To see this, let's again consider the Chinese gym. In the beginning there is a vast number of human computers that carry out calculations and pass the results among each other. But nothing about those calculations makes it necessary that they be computed by different persons. We can imagine one human computer taking on the tasks of his neighbour as well. This would change nothing about the computation as long as the same steps are being carried out. But then again we could continue with the next neighbour, letting one person do the job of three. And then the next and ever so on until we end up with only one busy human computer doing all the work.¹⁴ Chalmers has a very similar replacement process in mind when he talks about the computations of the brain being implemented by a "little daemons".

In principle, we can move from the brain to the Chinese room simulation in small steps, replacing neurons at each step by little demons doing the same causal work, and then gradually cutting down labor by replacing two neighboring demons by one who does the same work. Eventually we arrive at a system where a single demon is responsible for maintaining the causal organization [...]. (Chalmers 2011: 345)

The claim of strong AI is that implementing the right kind of computation alone suffices for having a mind, it does not make any appeal to whether the computation should be implemented in a serial or a parallel hardware. Indeed, computationalists see it as an advantage that they are able to account for *multiple realisability*. As long as it is the same computation, i.e. the same series of steps, it does not matter whether those steps are carried out by one processor or by many. Thus, the same neural net can be implemented in serial or parallel hardware and since the physical substrate in which the computation is implemented is irrelevant to the claim of strong AI, *the fact that SNNs are massively parallel is no objection to the CRA*.

II Subsymbolic computation

Symbols usually can be described on two level: First, they are part of a set of rules that describe its relation to other symbols and to other linguistic elements. This is the level of *syntax*. The syntax of the English language, for example, states rules about the order in which individual words need to be arranged such that they form a correct sentence. For computational symbols the syntax is given by the program or the set of rules that govern it. They state which symbol needs to be replaced with what other symbol such that correct steps are being carried out. But symbols also have an *semantic level*, i.e. they mean something. The English word "pen"

the architecture of the neural net. Changing the number of neurons, for example, would require rewiring the entire system. For digital computers all that is required is rewriting the program in order to adjust the simulation.

¹⁴He probably would have to be carefully not to mix up the order in which he has to carry out the steps since between every step of his "own" computation there would be a number of calculations he has to do for his neighbours. However, as long as the overall order stays the same, so does the computation that is being implemented.

as well as the Chinese symbol "笔" both refer to pens. We can use words and symbols to talk about the world. Symbols in the strong sense have both: a syntactic relation to other linguistic elements *and* a semantic content.

Usually the syntactic and the semantic level coincide. English words, for example, carry meaning *and* are rule-fully arranged into sentences. The same is true for the binary symbols used in digital computers. Digital computers use strings of 0s and 1s as their computational units. The English word "pen", for example, is represented in binary code as "01110000 01100101 01101110". Just as "pen" or "笔", this bit string represents pens. Whenever this symbol appears in the program it is being operated on as a whole, i.e. it's a *computational token*. Computational tokens are "atomic objects that are manipulated for the computation to take place." (Chalmers 1992: 33) For example, we might imagine a program that translates from English to Chinese. If such a translator would be presented with the input "pen" (encoded as 01110000 01100101 01101110) it would yield the output 笔. The rules that are stated in the program, thus, operate on *words*, i.e. linguistic elements that have *meaning*.

This changes when we consider neural networks. The computational tokens of a ANN are the membrane potential of the units and those of SNNs are the spiking patterns of individual neurons. They are the atomic objects that govern the computational process. However, neither the membrane potential, nor the spiking behaviour of a single neuron means anything. In neural networks information is encoded in *patterns* of activation, i.e. in patterns of membrane potential or in patterns of spike trains. Let us recall how the number "7" was represented in the digit recognition software. (see figure 5) The same system could be realised in a SNN. The brightness level would, then, not be encoded in the membrane potential of the input neurons but in their spiking behaviour. The simplest way to encode an activation level of let's say .37 would be to use a spike train that spikes 37 times per second. In total 788 neurons with different activation levels were used in order to encode one symbol. The single neurons, then, have no semantic content, only the input layer as a whole could be interpret as a meaning full representation. *In connectionist networks the syntactic falls below the semantic level.* The representations on the higher level have meaning, but play no role in the syntactic operations. The computational tokens are part of the syntax, but carry no semantic contents.¹⁵ Systems in which syntactic and semantic level diverge are called *subsymbolic systems*.

In cognitive science there is an ongoing debate between *symbolicism* and *subsymbolicism* about whether subsymbolic representation is needed for modelling mental processes. Symbolicism is the view that the mind computes over some sort of a language of thought, i.e. that the atomic objects of neuronal computation directly represent objects of the world.

¹⁵This is particularly true for SNNs, since here representation is distributed over neurons in a dual sense. First, just like in ANNs, usually a whole number of neurons is interpret as representing some entity. But additionally, under certain encoding strategies, the computational tokens are themselves spiking patterns of *groups* of neurons. While in ANNs computational tokens are the membrane potentials of single neurons, SNNs use the spiking behaviour of clusters of neurons as computational primitives.

To a first approximation, we can cash out this view as the claim that the computational primitives in a computational description of cognition are also representational primitives. That is to say, the basic syntactic entities between which state-transitions are defined are themselves bearers of semantic content, and are therefore symbols. (Chalmers 2011: 351)

Representatives of subsymbolicism hold against this view that most cognitive processes do not follow strict rules, but rather are associative and creative and that subsymbolic representation, therefore, serves for a better model of the mind, since it can account for loose mental operations on the higher level without giving up computational strictness at the basis. (see Rescorla 2015: 21-22) Cognitive representations on this view are build up *subsymbols*, i.e. smaller entities that themselves carry no semantic content and are thus not symbols in the full sense. "Entities that are typically represented in the symbolic paradigm by symbols are typically represented in the subsymbolic paradigm by a large number of subsymbols." (Smolensky 1988: 3)

The debate about whether mental computation is symbolic or not also carries over to the CRA. Let's first note that the computation in the Chinese room is *symbolic*. The computational primitives are Chinese symbols and, thus, carriers of semantic content. It might be for this reason that the Chinese room does not understand.

For example, the Churchlands (1990) argue that rule following is only one of many mental capacities that a connectionist network might learn and has nothing to do with its "mode of operation". The bare fact that the Chinese room is a symbolic system, thus, prevents the CRA from transferring to subsymbolic computation.

[I]t is important to note that the parallel system described is not manipulating symbols according to structure sensitive rules. Rather symbol manipulation appears to be just one of many cognitive skills that a network may or may not learn to display. Rule-governed symbol manipulation is not its basic mode of operation. Searle's argument is directed against rule-governed [...] machines: [systems] of the kind we describe are therefore not threatened by his Chinese room argument [...]" (Churchland and Churchland 1990: 36)

Similarly, Smolensky (1988) has argued that the fact that digital computers can simulate connectionist models (and vice versa) does not mean that the symbolic and the subsymbolic paradigm are equivalent approaches to cognitive representation. The former is a fact about syntax, the latter a question regarding semantics.

If one cavalierly characterizes the two approaches only syntactical [...], then indeed the issue - connectionist or not connectionist - appears to be "one of AI's wonderful red herrings."

It is a mistake to claim that the connectionist approach has nothing new to offer cognitive science. The issue at stake is a central one: Does the complete formal account of cognition lie at the conceptual level? The position taken by the subsymbolic paradigm is: No- it lies at the subconceptual level. (Smolensky 1988: 7)

The fact that Searle could simulate, i.e. follow the same syntactic rules as, a neural network is not enough to make it equivalent to a neural net. Searle manipulates symbols, a complete formal account of cognition, however, needs to consider subsymbolic representation. Mental states are intentional, i.e. they relate to the world. It is, therefore, not enough to focus on the syntax of computation, i.e. symbol manipulation, the semantic properties are just as important. The reason why Searle does not understand Chinese is rooted in the fact that he manipulates symbols, whereas the representations of the mind are build up of a complex structure of subsymbols. It is this composite structure of representation that accounts for intentionality.

Chalmers (1992) further clarifies this point. He argues that in subsymbolic systems the meaning of the higher level representations *emerges* from a complicated relations between subsymbols on the lower level. Much like thermodynamics properties like temperature and pressure emerge from particle motion, semantics emerges from the manipulation of subsymbols. Since there are no subsymbols being manipulated in the Chinese room it is not surprising that understanding does not emerge. All that the CRA can show is that the *computational tokens* are meaning less to the computing system. But that is what the subsymbolic paradigm postulates anyway. The computational primitives themselves carry no meaning, only the representations on the higher level might relate to the world. "Because the levels of syntax and semantics are distinct, connectionist systems are safe from the [Chinese room] argument." (Chalmers 1992: 17)

I think that all those arguments are mistaken and am going to argue in the following for what I call the *irrelevance thesis I*.

Irrelevance thesis I: The symbolic/subsymbolic distinction is of no importance to cognitive science. Whether a system is symbolic or subsymbolic is *irrelevant* to the questions of the mind.

The reason for this is that *any given system can be interpret as both, symbolic and subsymbolic*. Let's begin by noticing that "the symbolic/subsymbolic distinction completely cross-classifies the architectural distinction between Turing machines, say, and neural networks. Turing machines can be used to implement both symbolic and subsymbolic models, as can neural networks." (Chalmers 1992: 34) The output of above mentioned digit recognition system, for example, has been decoded symbolically. One neuron of the output layer corresponds to one

digit and, therefore, has a direct semantic content. On the other hand, if we were to simulate the input layer of that system on a digital computer it would still be subsymbolic. The simulated neurons still have no semantic content on their own.

Whether a system is regarded as symbolic or subsymbolic, therefore, has nothing to do with its physical or computational structure. All that matters is how we *interpret* the computational tokens. Do we interpret them as symbols with semantic content or do we interpret them as subsymbols that are merely manipulated for the computation to take place? There is nothing intrinsic to the system that would necessitate any of those interpretations. To see this let us consider once again the input layer of our digit recognition net as an archetypal example of a subsymbolic system. We can either interpret the 788 activation levels as encoding one digit. Or we can interpret the same 788 neurons as describing the gray scale of 788 pixels. Any computational token then carries meaning, viz. the brightness of the corresponding pixel. One might argue that the fact that *this* system can be interpreted as symbolic is not sufficient to show that *any* subsymbolic system can be. It might well be due to the conditions of the example that, by chance, the activation levels also can be interpreted as different shades of gray. This is true and the example is only given to illustrate my point. There is, however, one interpretation that we can always use and that makes the computational tokens bear semantic content. We can always interpret the activation levels as real numbers. Nothing keeps me from interpreting an membrane potential of, let's say, .37V or a spike train with 37 spikes per second *as the number .37*. Under this interpretation, the same system would not be recognising handwritten digits but compute a mathematical function $f : \mathbb{R}^{788} \rightarrow \mathbb{N}$ that maps from a 788 dimensional vector space to the natural numbers. The reverse direction works as well. Symbols in digital computers are encoded in bit strings. "Pen", for example, is encoded as "01110000 01100101 01101110". But this symbol itself has an internal structure very similar to that of a subsymbolic representation. Again, nothing prevents me from regarding the individual bits as computational tokens making them mere subsymbols, i.e. elements that carry meaning only in conjunction with each other.¹⁶ Any physical symbol can be interpreted at various levels and nothing intrinsic to the physics favours one interpretation over any other. Mental features, on the other hand, are *intrinsic* properties of a system. It is a fact of nature that I am conscious and that the desk in front of me is not. This is not up to our interpretation.

Therefore, we have established the following to claims.

- (1) Mental properties are intrinsic to physics.
- (2) The symbolic/subsymbolic distinction does not make any statement regarding the intrinsic nature of a system.

¹⁶Chalmers (1992: 11) argues that a system can only count as subsymbolic if there is no interpretation such that it is symbolic. This would necessarily render the present bit string a symbol. This demand, however, is completely ad hoc. Also, we have shown that *any* system can be interpreted as symbolic, thus making the set of subsymbolic systems so defined the null set.

From (1) and (2) follows the conclusion

(C) that the symbolic/subsymbolic distinction is *irrelevant* to the question whether a given system possess mental properties or not (irrelevance thesis I).

Let us next discuss whether the CRA can be applied to subsymbolic systems. The easiest way would be to interpret the Chinese room as a subsymbolic system. We might take the pixels of the input screen to be the subsymbols of the Chinese representations that are being fed into the Chinese room. Likewise, we interpret the spots on Searle's cards to be the computational primitives. Just like in any other subsymbolic systems the computational tokens, then, don't carry meaning and the syntactic and the semantic level diverge. *The fact that the rules are stated with respect to the symbols need not bother us, since any explicit rule about a symbol is an implicit rule about a subsymbol. It is not possible to manipulate a Chinese symbol without also manipulating the pixels on the screen or the spots on Searle's cards.* Due to the irrelevance thesis I we could leave it at that. However, we can modify the CRA such that it more straight forwardly applies to subsymbolic systems. Imagine, Searle would not be carrying out computations over Chinese symbols but over activation levels, i.e. real numbers. Just like in a neural net the Chinese representations are encoded in those activation numbers. We might for example imagine a similar encoding strategy as in the digit recognition case: use the pixels of the input screen and to every pixel assign one neuron which encodes the respective brightness in its membrane potential/spiking behaviour. The rules in Searle's book are stated in English and tell him how to translate activation levels to activation levels. In effect, the situation is equivalent to the Chinese room except for the fact that Searle is not carrying out computations over levels of activation and not over Chinese symbols. Such a system would certainly have to be called subsymbolic. Now, imagine further that those activation levels are not written as Arabic but as *Chinese numbers*. We, then, arrive at exactly the same thought experiment as in the original CRA. Searle manipulates Chinese symbols without understanding what they mean, however, passing the Turing test for having a Chinese conversation. The only difference is that in this case the subsymbolic interpretation is more obvious.

The subsymbolic/symbolic distinction is not a matter of the intrinsic properties of a system. Any computational system can be interpreted as symbolic *and* as subsymbolic. Cognitive science, however, is concerned with intrinsic mental properties and, therefore, should be indifferent about the question whether we use a system to implement a symbolic or a subsymbolic computation. The Chinese room, just as any other symbolic system, can be understood as performing computations over subsymbols. *The fact that SNNs favour a subsymbolic interpretation, therefore, is no objection to the CRA.*

III Machine learning

Most connectionist networks are specifically designed for machine learning tasks. Through a training process of iteratively updating the connection strengths neural networks learn the function they compute. With sufficient amount of training data at hand connectionist networks can simulate almost any functional correlation without the need of hard-coding the rules that govern it. This feature makes them particularly well suited for tasks in which the relationship between in- and output is either not known or very complex, since we need not "tell" the system beforehand which function to compute but rather let it figure it out itself via try and error. Let's take as an example once again the digit recognition system. Explicitly stating the function between the brightness level of 788 pixels and the digit they represent would be a hopeless endeavour. We simply do not know the hard rules that make our example a writing sample of a 7 rather than that of a 1. In this respect learning in artificial neural networks much resembles human learning. When children learn to read and write numbers they are not told hard rules like: "A seven consists of two lines with an angle of no more than 88° between them and the first being at least 34% shorter than the second." They learn to recognise digits by being presented with them and being corrected when they mistake a 7 for a 1.

Most ANNs use a mathematical process called "error back propagation" for updating their synaptic weights. Although back propagation is particularly easy to implement it lacks bio plausibility. Biological brains most certainly use different mechanisms in order to adjust their synaptic connections. (see Rescorla 2015: 23) SNNs, on the other hand, allow for a learning mechanism with a high degree of bio fidelity: Spike-timing-dependent plasticity (STDP). STDP uses the precise spike timing of neighbouring neurons to adapt the synaptic strength between them. It, therefore, has no counterpart in non-spiking ANNs. On the other hand, biological brains are likely to use STDP. (see Tavanaei 2019: 49) Learning in SNNs is, thus, particularly akin to human learning processes.

Bechtel and Abrahamsen (1993) argue that it is this capability to learn from experience that distinguishes connectionist networks from the symbolic approach to cognition. The "most distinctively human capabilities for dealing with information" (Bechtel & Abrahamsen 1993: 65) are best represented by the natural learning mechanisms of neural networks. Many cognitive skills like the ability to read and write numbers or to understand a language are learned through interaction with the environment and connectionist learning is best used to model this process. "[L]earning a language or learning to do arithmetics is awkward to model symbolically, but is natural (although challenging) to model using networks." (Bechtel & Abrahamsen 1993: 65)

What lies at the heart of the CRA is what Sven Harnard (1990) has called the *symbol grounding problem*. Whereas mental states like beliefs or desires are intrinsically intentional, computational symbols only have derived intentionality, i.e. they only mean something if they

are used to refer to the world by somebody. How is it then possible that computation, i.e. the mere manipulation of symbols, is able to account for mentality? “How is symbol meaning to be grounded in something other than just other meaningless symbols?” (Harnard 1990: 340)

Symbol grounding problem: Mental contents have intentionality but symbols have no intrinsic meaning independent from a homunculus that interprets them. But what then grounds the meaning of the symbols in a computationalist account of the mind? (see Harnard 1990: 339)

Many have argued that machine learning might provides a solution to the symbols grounding problem.

Tim Crane (1991) agrees with Searle that the Chinese room in the original thought experiment does not understand Chinese. Simply shuffling symbols according to rules does not endow them with meaning. Syntax indeed is not semantics. However, Crane thinks that experience might provide a causal link between the system and the environment and thus is able to account for understanding.

“Searle’s assumption, none the less, seems to me correct [...]. I argued that the proper response to Searle’s argument is: sure, Searle-in-the-room, or the room alone, cannot understand Chinese. But, if you let the outside world have some impact on the room, meaning or ‘semantics’ might begin to get a foothold.” (Crane 2003: 128)

Also Fodor agrees that the Chinese room does not understand Chinese because Searle lacks the appropriate causal link to the environment.

Given that there are the right kinds of causal linkages between the symbols that the device manipulates and things in the world [...] it is quite unclear that intuition rejects ascribing propositional attitudes to it. All that Searle’s example shows is that the kind of causal linkage he imagines [...] is, unsurprisingly, not the right kind. (Fodor 1980: 431)

Machine learning might just be the right mechanism to account for such an impact. The feedback-update strategy of training provides a causal link between the environment and the program. If the system receives a signal that its operation was not optimal, it changes the synaptic connections. The in- and output along with the update signal, thus, *have a causal impact on the system and its behaviour*. Causal theories of meaning are very prominent in the philosophy of language and causal machine learning mechanisms might ground the meaning of the computational symbols. If we, therefore, were to introduce a training mechanism to the Chinese room thought experiment, Searle might indeed *learn* to speak Chinese by interacting with his environment.

The idea that machine learning processes might account for the intentionality of mental representation gave rise to a new understanding of AI sometimes called epigenic robotics.

Those who agreed with Searle that there is a problem, but were still committed to the functional theory of mind, developed a hybrid response by using connectionist nets to ‘hook’ the symbolic or representational domain onto the world. A dominant idea is that connectionist systems can be readily connected to the outside world by ‘learning’ a mapping between sensors and actuators. (Sharkey and Ziemke 2001: 260)

Harnard (1990) himself offers a solution to the symbol grounding problem that relies on connectionist learning. He is one of the few to spell out a mechanism that could explain understanding. His account relies on the notion of *iconic representations*. "These are internal analog transforms of the projections of distal objects on our sensory surface. In the case of horses (and vision), they would be analogs of the many shapes that horses cast on our retinas." (Harnard 1990: 342) He does not explain what he means by saying that iconic representations are "analog copies of the sensory projection" (Harnard 1990: 342) but he seems to have something in mind like the changes in brain states that are caused by sensory inputs. He describes the causal chain as follows: "It is a purely causal connection, based on the relation between distal objects, proximal sensory projections and the acquired internal changes that result from a history of behavioral interactions with them." (Harnard 1990: 343) Those iconic representations do not mean anything yet. They stand in a relation to the objects of which they are a representation of like a image in a camera. Connectionist learning mechanisms, then, are used to categorise different iconic representations that represent the same object, for example different "pictures" of horses, under one common name. Harnard, thus, seems to be suggesting a mechanism similar to that of our well known digit recognition system. But instead of digits the system learns to recognise horses from the impressions that real horses leave on the input layer of the neural net. Through a process of training the system learns to attach the name "horse" to certain objects and not others, i.e. *it learns the meaning of the word "horse"*.

I think that, despite its intuitive plausibility, this approach fails due to what Searle has called the *homunculus fallacy*. In order to make sense it has to assume a little man inside the brain that constantly interprets the physical states as symbols.

The idea is always to treat the brain as if there were some agent inside it using it to compute with. [...] Typical homunculus questions in cognitive science are such as the following: "How does the visual system compute shape from shading; how does it compute object distance from size of retinal image?" A parallel question would be, "How do nails compute the distance they are to travel in the board from the impact of the hammer and the density of the wood?" [...] If we are talking about how the system works intrinsically neither nails nor visual systems compute anything. We as outside homunculi might describe them computationally [...]. But you do not understand hammering by supposing that nails are somehow

intrinsically implementing hammering algorithms and you do not understand vision by supposing the system is implementing, e.g., the shape from shading algorithm." (Searle 1990a: 28-29)

Harnard's solution to the symbol grounding problem is a generic example of the homunculus fallacy. In order for the explanation to work Harnard must assume a homunculus that interprets the system as performing some sort of image recognition. This interpretation is not intrinsic to the system itself.¹⁷ Harnard must be assuming a homunculus that asks the question "What are horses?" and that somehow perceives the iconic representations. Otherwise the system would not be attaching names to objects but only patterns of activation to patterns of activation. Without the homunculus all that is left is a system manipulating meaningless symbols. For real computers there is no homunculus fallacy, since there always is a user who interprets them. We interpret the activation of the neurons in the input layer as gray-scale levels of pixels and that of the output neurons as the recognised numbers. But those are ascriptions of the user, not intrinsic properties of the system. "Without the homunculus there is no computation, just an electronic circuit." (Searle 1990a: 33)

We can illustrate this point by incorporating learning into the Chinese room. Imagine the same scenario as in the original thought experiment but with one slight difference: In addition to his existing equipment Searle receives a *second rule book* along with an electrical sign that tells him whether he was successfully imitating a native speaker or had been detected as an impostor. Further we imagine that the first rule book initially serves as a poor translation table and the Chinese room performs very badly at conversing in Chinese. Upon being asked “你叫什么名字” (“What is your name?”) Searle answers “笔是蓝色的” (“The pen is blue”) immediately failing the Turing test. We then use the electrical sign as a feedback mechanism which tells Searle that he had the wrong rule book. *Whenever Searle receives the information that he had been detected he alters the first rule book according to the rules stated in the second.* He then tries again, fails and updates the program. After some rounds he figures out that when receiving the input “你叫什么名字” (“What is your name?”) he has to respond with “约翰塞尔” (“John Searle”). He then keeps on playing the imitation game always updating his rule book when failing. After some time he gets so good at answering Chinese questions that he always responds with the correct symbols keeping up a proper conversation. He passes the Turing test thus completing his training. With the correct rule book in hand he can keep on engaging in conversations still not understanding a word of Chinese.

Effectively, *the original thought experiment is the best case scenario for the Chinese room with integrated learning mechanism*: After completing successful training he arrives at the same rule book that he had to begin with in the original thought experiment. The fact that

¹⁷Just as in the case of digit recognition we might imagine a homunculus that interprets the same system as computing a complex mathematical function.

Searle *learns* to answer with meaningless symbols to meaningless symbols adds nothing to his understanding. Even under ideal training conditions Searle understands no more Chinese than he did in the original case.

This is also true if we imagine subsymbolic learning like Harnard does. Instead of Chinese symbols Searle might be manipulating Chinese numbers. Through a successfully training process he learns to answer with patterns of activation upon receiving patterns of activation. Under ideal conditions he learns to manipulate those Chinese numbers in the same way as described in the previous chapter. Nothing about the fact that he has *learned* his program is able to account for meaning. The apparent plausibility of Harnard's solution to the symbol grounding problem derives from the implicit assumption of a homunculus that perceives the input pattern as degrees of brightness or as real numbers and that interprets the output pattern as names of objects. Without the homunculus all the system does is learn to attach meaning less symbols to meaning less symbols, or meaning less patterns of activation to meaning less patterns.

Despite the apparent plausibility of the claim that causally linking the computational system to the environment by machine learning techniques might account for understanding introducing training mechanisms to the Chinese room yields no relevant changes. Even under ideal training conditions the room learns nothing more than it already could do in the original thought experiment. Once the homunculus is removed the plausibility of learning approaches to the symbol grounding problem vanishes. *The fact that SNNs are particularly well suited for machine learning tasks is, thus, no objection to the CRA.*

IV Analogue representation

Digital computers use bit strings to encode information. A bit, or *binary digit*, is either 0 or 1 and is most commonly represented in voltage levels of electrical circuits. The most straight forward way to encode digital information would be to represent 0s by a negative and 1s by a positive voltage level.¹⁸ All information that is being processed in digital computers, thus, is encoded in a series of 0s and 1s or of positive and negative voltage levels. SNNs on the other hand are commonly referred to as being *analogue*, i.e. they do not operate on bits that are either 0 or 1 but on a continuous variety of spiking patterns.

In philosophy there are two different understandings of the analogue/digital distinction. The most prominent account has been put forward by Nelson Goodman in his 1969 monograph *Languages of Art*. Goodman understands analogue representation as "dense" as opposed to digital representations which are "differentiated". Representations are physical objects or

¹⁸The most common way, however, would be to interpret a change from positive to negative voltages as 0s and the change from negative to positive as 1s.

states that represent something. Examples are pictures, thermometers, voltage levels or spike trains. *If for any two numbers that are not identical and two physical states that represent those numbers one can find a state that represents an intermediate number, the system is called analogue.* Analogue representations are dense in the sense that no matter how close two of them are, as long as they are not copies, there is an intermediate representation. A system is digital if it is not analogue. (see Thompson-Jones & Moser 2015: 4-6) For example, a mercury thermometer is analogue. For any two temperatures that it displays, for example 24.6°C and 24.7°C, there is an intermediate representation. The same is true for a mechanical watch. For any two distinct times it tells there is an intermediate time that could be represented as well. A digital watch, on the other hand, (unsurprisingly) is digital. It can display the times 12:22 32 and 12:22 33 but nothing in between. Also tally marks are digital. We can represent a 5 as ~~||||~~ and a 6 as ~~||||~~/ but no 5.5.

David Lewis (1971) challenges this view and argues the analogue/digital distinction is not a matter of continuity and discreteness but rather that an analogue representation is a "representation of a number by physical magnitudes of a special kind. Resistances, voltages, amounts of fluid, for instance, are physical magnitudes of the proper kind for analog representation." (Lewis 1971: 324) Physical magnitudes of the proper kind are *primitive magnitudes*, i.e. magnitudes that are expressed by primitive physical terms, such as voltage, mass, velocity, etc. Lewis gives an example of a representation that is digital in Goodman's sense, but should be regarded as analogue according to his view. He thinks of an electrical computer that uses resistors to encode numbers. "A setting of 137 ohms represents the number 137, and so on." (Lewis 1971: 322) We might imagine that such a computer can represent any arbitrary real number by using a smoothly varying resistor that can be continuously tuned to any resistance between (almost) 0 and, let's say, 200 ohm. Or we might imagine a step wise variation of resistance by connecting in series a number of resistors with constant value. According to Lewis, both computers should be regarded as analogue since they use a primitive magnitude (resistance) in order to represent numbers.

Digital computers are digital in both senses. They use voltage levels that either encode a 0 or a 1, no value in between can be represented.¹⁹ And they are also digital in Lewis' sense, since numbers are represented by complex variations of voltages rather than by mere voltage levels. ANNs, on the other hand, are analogue in both senses. They are able to represent any real number in the voltage level of the membrane potential. Representations in ANNs

¹⁹It is of course possible to implement numbers between 0 and 1. Digital computers use so called floating point numbers which are approximations of real numbers. Usually they consist of a series of 16 bits that represent the decimals of the number to be represented. However, the representation of floating point numbers is digital as well. 16 bits allow for the representation of 5 decimals and thus restrict the resolution detail. We can represent the numbers .13701 and .13702 but nothing in between. A finer resolution could be achieved by using longer bit strings. Any fixed number of bits, however, is only able to approximate reals up to a certain number of decimals.

are, thus, *dense* and *primitive*. For SNNs the story becomes a bit more complicated. They are, however, clearly capable of analogue representation in Goodman's sense. Representations in SNNs are dense, since for any two numbers we always can find a spiking behaviour that represents an intermediate number. The firing frequency of a neuron, for example, can vary continuously between 0 and a given number. Whether SNNs are analogue on Lewis' account seems to depend on the encoding strategy that is being used. The firing frequency of a neuron might be regarded as a primitive magnitude such that frequency coding SNNs would be analogue. Unfortunately, Lewis is not particularly clear about what counts as a primitive magnitude. For other coding strategies like rank order and latency coding SNNs are digital on Lewis' understanding. While in ANNs, for example, a membrane potential of 137V directly encodes the number 137, the voltage level of a spiking neuron has no direct representation. Only its time evolution in combination with that of other neurons is used for implementation. The precise time interval between the spikes of different neurons certainly is not a primitive physical term like voltage, mass or resistance. Therefore, when we discuss the applicability of the Chinese room argument to analogue computers, we understand analogue in Goodman's sense as dense representation, since SNNs are not necessarily analogue on Lewis' account.

Let us next discuss whether the Chinese room is digital. So far we have used the analogue/digital distinction with respect to representations of numbers, the Chinese room however computes over Chinese symbols more generally. Let's first note that Chinese numbers are represented digitally. There is a symbol 五 for five and there is a symbol 六 for six but there is no smooth transition between them that is used to represent intermediate numbers. Chinese numbers just as Arabic and binary numbers are differentiated and thus digital. There is, however, some discussion whether it makes sense to speak of written words as being digital. It does not make much sense to ask whether there is a word between "pen" and "paper", since there is no intuitive way to define a distance between them. (see Maley 2011: 125) Written words, just as letters of the Chinese alphabet, however are *differentiated*, i.e. there is no smooth transition between them.

[L]etters of the alphabet, written words, and poker chips are all discrete [...]. In these cases, the representation is either wholly present or it is not—half of a letter does not count as a letter. Furthermore, there is no representation between any two representations that are adjacent according to some ordering. There is no letter between 'a' and 'b', and if we sort English words by length and then alphabetically, there is no word between 'cam' and 'can'. (Maley 2011: 125)

On Goodman's account the Chinese symbols which Searle uses to compute, therefore, are digital.

Let's now discuss whether the CRA can be applied to analogue computers. In the following, I am going to argue for two seemingly contradictory theses. First, I am going to show that

there is a (fairly limited) sense in which it cannot. Then, I am going to defend the *irrelevance thesis II*: The analogue/digital distinction is irrelevant to the questions of the mind. I hope that in the course of my explanations it will become clear that the two theses are not mutually exclusive.

The Church-Turing thesis states that any function that is computable by an effective method can also be computed by a Turing-machine and vice versa.

Church-Turing thesis: The set of functions that can be computed by an effective method is equivalent to the set of functions that can be computed by a Turing machine. All (effectively) computable functions are Turing-computable. (see Copeland 1997: 1-2)

Most programming languages are Turing-complete, i.e. they can simulate any Turing machine. The set of functions that can be computed by a digital computer is, thus, equivalent to the set of functions that can be computed by effective methods. According to Copeland (1997) a method M is called effective if and only if:

1. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
2. M will, if carried out without error, produce the desired result in a finite number of steps;
3. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
4. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method. (Copeland 1997: 2)

From this definition it becomes clear that the method which Searle uses in the Chinese room is an effective method. It is set out in a finite number of instructions written in the rule book, it produces the desired result in a finite number of steps, it actually is carried out by a human and it demands no insight or intuition. And since digital computers are Turing-complete, the set of functions that can be computed by the Chinese room is equivalent to that which can be computed by a digital computer. *Digital computers and the Chinese room are computationally equivalent.*

There are, however, functions that cannot be computed by effective methods²⁰ and, thus, not by Searle in his Chinese room. With regard to digital computers this is not a problem. If it cannot be computed by the Chinese room, it cannot be computed by a digital computer. Analogue computers, however, are potentially capable of performing *super-Turing computation*, i.e.

²⁰The proof for this claim is so elegant that I want to briefly sketch its outline. The set of possible Turing machines is countable. The set of functions, however, is uncountable. Therefore, there must be functions for which there is no Turing machine that computes them. Together with the Church-Turing thesis this yields the desired claim.

they can compute functions that are not Turing-computable. (see Siegelmann 1995) In particular analogue neural networks are proposed to have super-Turing computational power. (see Siegelmann 2003) The reason why analogue neural nets potentially are capable of computation beyond the Turing regime is that they can implement irrational (possibly uncomputable) numbers. (see Siegelmann & Sonntag 1994: 3)²¹ Whether such models can be physically realised is an open question. A thesis that is often confused with the Church-Turing thesis is the claim that Copeland has called the *maximality thesis*:

Maximality thesis: All functions that can be generated by machines (working in accordance with a finite program of instructions) are computable by effective methods. (Copeland 1997: 24)

Whereas the Church-Turing thesis is uncontroversial it is not clear yet whether the Maximality thesis holds. "At the present time, it remains unknown whether hypercomputation is permitted or excluded by the contingencies of the actual universe. It is, therefore, an open empirical question whether or not the [...] the maximality thesis is true." (Copeland 1997: 32) Therefore, it is at least possible that analogue neural networks are capable of implementing functions that the Chinese room cannot compute.

Searle typically assumes that brain processes can be simulated by digital computers. "Can the operations of the brain be simulated on a digital computer? [...] The answer seems to me [...] demonstrably 'Yes' [...]." (Searle 1992: 200) I want to call this claim the *simulation thesis*:

Simulation thesis: Any brain process relevant for cognition can be simulated by a digital computer.

Searle justifies the simulation thesis with a claim that he mislabels "Church's thesis": "[A]nything that can be given a precise enough characterization as a set of steps can be simulated on a digital computer [...]." This claim as it stands is simply false. For example, the halting problem has a precise characterisation as a set of steps, however, cannot be solved by a Turing machine. Since Turing machines and digital computers are computationally equivalent, the halting function (just as any other uncomputable function) cannot be simulated by a digital computer. (see Copeland 1997: 30) Not all processes are effectively computable and it is the task of neuro science to find out whether all *brain* process can be simulated by a digital computer. "It is an open question whether a completed neuroscience will need to employ functions that are not effectively calculable." (Copeland 1997: 37)

But for now let's assume that the simulation thesis is true and all relevant brain processes are Turing-computable. How would we have to modify the CRA such that it applies to analogue

²¹On an interesting side note: The fact that the analogue supremacy derives from the theoretical ability to encode irrational reals is an argument for understanding "analogue" in Goodman's sense. For digital on Lewis' account allows for the implementation of irrational numbers as well. We might, for example, interpret the square of the time interval between spikes of two different neurons as a number x . This representation is certainly not primitive, thus digital, but capable of implementing irrational numbers.

computers such as SNNs? First of all, we would not have to modify it at all. Whatever the right kind of computation might be, as long as it is Turing-computable it can be implemented in the Chinese room and the CRA applies. However, for reasons of clarity we might imagine the same thought experiment as in section II but with the difference that the numbers Searle uses to compute are not represented by Chinese symbols but by the "brightness" of uni coloured gray cards. The higher the number that is to be represented, the brighter the cards. A 0 for example would be encoded as a black card, a .7 as light gray and a white card would represent a 1. All other details are exactly as in the previous example. Searle computes according to his rule book, passes the Turing test for speaking Chinese, however, understands nothing. The only difference is that *the Chinese room now acts as an analogue computer*. Analogue representation alone is, therefore, no objection to the CRA.

Both claims, the maximality and the simulation thesis are likely to be true. (see Sandberg & Bostrom: 38) However, they are not proven, and it is, thus, possible that some brain processes are computable by analogue computers but not by effective methods. If this is the case the reason why Searle does not understand Chinese might be because he cannot implement the right computations. An analogue neural net, however, might be able to simulate the relevant brain process and, thus, to understand the meaning of the Chinese symbols.

In the remainder of this section I want to argue that, even though the CRA is not applicable to the full extent, the fact that a computer might be analogue rather than digital is *irrelevant* to the questions of mentality.

Irrelevance thesis II: The digital/analogue distinction is of no importance to cognitive science. Whether a system uses differentiated or dense representations is *irrelevant* to the questions of the mind.

The argument is going to be in the spirit of that in section II. Any given system can be interpreted as both, analogue *and* digital. Let's begin by considering the generic case of a digital computer. Digital computers use varying levels of voltage to encode strings of 0s and 1s. A possible way might be to implement 0s as negative and 1s positive voltages. Since no other number can be directly represented, digital computers are clearly differentiated. However, nothing intrinsic to physics necessitates this interpretation. We might just as well interpret the same levels of voltage as real numbers. Imagine a voltage level varying in a step wise manner between -1V and +1V. Nothing intrinsic to the laws of electrodynamics makes this a bit string. We might just as well understand the same voltage curve as representing a continuously varying real number between -1 and 1. Under this interpretation the same system would have a dense representation, implementing all real values from -1.0 over -.3 and .7 to 1.0. Digital computers are only digital because we use them as such.

There also is an interpretation under which the Chinese room is analogue. Instead of interpreting the lines of paint on Searle's cards as Chinese symbols we might interpret them as

representing real numbers. The amount of ink in ml on a given card implements the corresponding real number. It might, for example, take .3ml to draw a 什 but .7ml for a 笔. One and the same physical object can be used to represent pens and numbers. Under this representation Searle would not be having a Chinese conversation but computing a mathematical function.

The reverse direction works just as well. Consider, for example, an ANN with neuronal membrane potentials between -1.0V and +1.0V. We can either interpret them as representing real numbers or we can interpret them as encoding strings of binary digits. After every second check whether the membrane potential is positive or negative. If it is negative interpret it as a 0, if it is positive interpret it as a 1. After 8 seconds a string of 8 bits has been implemented. If we change interpretation ANNs are digital.

We also have already met a system that uses analogue encoding and digital decoding: our well known digit recogniser. The input representations, be it membrane potentials or spike times, are dense. For any two levels of gray scale there is a representation for an intermediate brightness. The output representation, however, is digital. The spiking of a certain output neuron represents a *digit*, i.e. a natural number between 0 and 9. No number in between can be represented. *Whether a system is analogue or digital completely depends on the way we use it.* For any given system there always is a digital *and* an analogue representation. If we want to avoid the homunculus fallacy, the conclusion follows: Whether a system is digital or analogue is irrelevant to the question whether it has mental features (irrelevance thesis II).

To sum things up, while the CRA is applicable to SNNs given that either the maximality or the simulation thesis is true, the argument from observer relativity applies to computation in general, be it digital or analogue. Even if relevant brain processes could be simulated by a SNN but not by a digital computer, the fact that a system implements this simulation is not sufficient for cognition. Since once the homunculus is removed, it is not a fact after all. Any given system is an analogue computer only by way of being used as such.²² *The CRA is, therefore, applicable to SNNs under the condition that either the maximality or the simulation thesis is true. Analogue representation alone is no objection to the CRA.*

V Temporal encoding

All considerations so far do not only apply to SNNs but to neural nets in general. This changes now. What distinguishes SNNs from ANNs (and digital computers) is that they use time to

²²Once again a proponent of strong AI might insist that a system implements a certain (analogue) computation not by way of being interpreted but by way of being *interpretable* as such. With this understanding of implementation no homunculus is required. However, strong AI would then amount to the bold claim that out of all possible facts of nature it is human interpretability that accounts for cognition. It seems unreasonable to assume that nature cares about how we interpret it. Also, any system is *interpretable* as an analogue computer.

encode information. The standard understanding of computation in the Turing machine model makes no reference to time. Whether a human computer works fast or slow does not matter to the computation that she implements. As long as she goes through the same series of steps in the same order speed is of no importance to the calculations that she carries out. The same is true for digital computers. The same program can be run on computers with different processor speed. The Word program runs on fast and slow computers alike. On a classical computational account time is irrelevant. "One could physically implement the same abstract Turing machine with a silicon-based device, or a slower vacuum-tube device, or an even slower pulley-and-lever device." (Rescorla 2015: 61)

Therefore, some have argued that (classical) computationalism is unable to appropriately represent the temporal aspects of cognition. Van Gelder, for example, stresses the fact that "*natural cognition happens in real time*. [...] Every cognitive process unfolds in continuous time, and the fine temporal detail calls out for scientific accounting." (van Gelder 1998: 622) Some cognitive tasks, like dodging a punch or catching a ball, even depend *essentially* on time. If the timing is not right you will get hit or miss the ball. Dodging and catching are only achieved when carried out fast enough. Furthermore, he argues that cognition is embedded in (1) a neural substrate, (2) a body and (3) an environment. Therefore, there should be a way to relate cognition to the time evolution of brain processes, the speed of bodily movement and the conditions of the surrounding. A purely computational model of the mind is unable to account for the temporal aspects of cognition. Van Gelder, thus, argues that cognition, as any other natural process, is best described by a *dynamical account*. (see van Gelder 1998: 622)

Also late neuro scientific approaches stress the importance of time in an appropriate model of the mind. Northoff et.al. (2020a) suggest that tempo-spacial dynamics might be the "common currency" between the mind and the brain. They postulate an "inner time and space" (Northoff et.al. 2020a: 1) and argue that in healthy cognition the inner time is in sync with physical real time. Depressed patients, for example, often experience the world around them as too fast. They feel that they cannot cope with the speed of the surroundings. Conversely, patients suffering from mania experience the world as too slow. Northoff et.al., thus, argue for replacing "the view of the brain as information processing device [...]" by its characterization as enabling space-time transformation." (Northoff et.al. 2020b: 80)

But also computationalists start to incorporate time into their models. Piccinini for example argues that one need to specify a time scale over which a computational step is being carried out, just as a physicist needs to specify whether the time that appears in his calculations needs to be measured in nanoseconds or years. (see Piccinini 2010: 857) Chris Eliasmith develops a "theory, which essentially includes precisely these kinds of dynamics, shows how representation, computation, and dynamics can be integrated in order to tell a unified story about how the mind works." (Eliasmith 2003: 518)

The CRA, on the other hand, makes no reference to time. It might be for this reason that the Chinese room does not understand. "One is tempted to complain that Searle's thought experiment is unfair because his Rube Goldberg system will compute with absurd slowness." (Churchland & Churchland 1990: 34) The reason that Searle does not understand Chinese is, therefore, that he is implementing the wrong kind of computation. What the CRA, then, shows is not that computation was not sufficient for mental features, but that a computation that makes no reference to time cannot be the right kind of computation. A natural way to include time into a computational model is to use SNNs. Eliasmith et.al., for example, use spiking networks for their large scale brain models in (Eliasmith et.al. 2012) and (Eliasmith et.al. 2016). By encoding information in the timing of spikes the relevant temporal aspects of cognition might be captured computationally. The question, therefore, is: Can the CRA be applied to computers that use temporal encoding? And the answer is yes. In the following I want to present a thought experiment which I call the *Morse code room* that resembles the Chinese room in many aspects but is able to account for temporal encoding.

Imagine myself, as I speak no Morse code, in a room which contains nothing but a loud speaker, a rule book, some scratch paper and a Morse key. Around my wrist I have a watch that precisely tells the time. I have instructions to translate incoming clicking noises from the loud speaker to clicks on the Morse key. The rule book is written in English such that I can understand it and states precise rules how to answer with clicks upon receiving a certain pattern of sounds from the speaker. In order to remember the incoming pattern I am allowed to write down the times at which the speaker had clicked. I keep on following the rules although to me all those noises seem completely arbitrary and the clicking patterns that I enter on the Morse key are absolutely meaning less. Unknown to me the incoming signals were questions in a new form of Morse code and the clicks I returned were taken to be my answers. Let's further suppose that after a while I get so good at following the instructions, precisely timing the clicks on my key, that I always respond with the appropriate answer. Every incoming clicking pattern is precisely converted to clicks on the Morse key according to the rules stated in my book. From an outside perspective it seems as if I was having a conversation in Morse code, as my behaviour is indistinguishable from that of a person who actually spoke Morse code. The instructions are written such that if I were asked the same questions in German or English I would be giving the exact same answers. Therefore, I pass the Turing test for speaking Morse code. Unlike in the case of German or English, however, *I do not know what those sounds mean*, I lack understanding. As far as my speaking Morse code is concerned I behave exactly like a SNN. I perform operations over temporally encoded symbols.

The claim of strong AI is that any system that implements the right kind of computation actually has mental states, be it the human brain, a SNN or myself in the Morse code room. *But even though I am implementing the same computation as the brain of someone who speaks (this new form of) Morse code, I lack the mental feature of understanding.* As long as brain

processes are computable I can simulate them and as long as analogue computers are not super-Turing I can implement any computation that a SNNs carries out. *Temporal encoding is, thus, no objection to the CRA.*

VI The Morse code room and its replies

In this section I want to further discuss the Morse code room and its relation to SNNs. Then, I am going to reconsider some of the replies against the CRA in the light of the new thought experiment.

Note that the Morse code room is akin to SNNs in all regards discussed in this thesis.

(1) It could be used to compute *in parallel*. Just like in the case of the Chinese gym we might imagine a large enough building that houses a vast number of Morse code rooms that are interconnected via Morse keys and speakers and trade information in the precise timing of clicks. We, then, again could start cutting down labour by replacing neighbouring rooms with only one person that computes the same function. In order to make things easier for the person we could imagine that the speakers of the different rooms made sounds in a different pitch. The speaker in room one emits a deep humming noise, whereas the speaker in room two peeps in a high pitched tone. Also the Morse keys have different colours. The key from room one is blue, whereas room two has a red key. After having replaced all other human computers we end up with one person doing the entire work. His room contains a number of speakers that all click in different pitches and a series of differently coloured Morse keys. The rules in his book then tell him how to relate incoming series of clicks from different loud speakers to clicks on the various Morse keys.

(2) It seems natural to interpret such a system as *subsymbolic* and to understand a whole series of incoming patterns from different speakers as one single representation. For example could our new form of Morse code use different pitches to encode words. "Pen", for example, might be represented by a deep humming followed by two high-pitched sounds.

(3) We could also introduce a learning mechanism into the Morse code room in very much the same way as in section III. We add a second rule book according to which I have to change the rules in the first book upon receiving a signal that I had failed the Turing test. After a certain amount of repetitions I *learn* to answer with the correct clicking patterns, however, without understanding Morse code.

(4) And even though the representation of the words in Morse code is differentiated and thus digital, we might interpret the various clicking noises as implementing an analogue computation. We might, for example, add numbers to our Morse code alphabet. We, thus, define the time interval between a certain deep and a particular high-pitched noise measured in seconds to represent that number. If a deep tone is followed by a high tone after 3.2 seconds this implements the real number 3.2. Unlike Chinese or Arabic numbers, Morse code representa-

tion of numbers is analogue. In an indirect way we then might even represent words using reals. Imagine a TV screen that displays words in English. We can represent any of the screens pixel by a real number that stands for its brightness and is implemented in the time interval between clicks. Under this representation even words are represented using analogue implementations of numbers. The Morse code room, thus, can be interpreted as an analogue computer.

Next, I want to reconsider the systems, robot and brain simulator reply in the light of the Morse code room.

The systems reply: The argument could be restated as following. Although it might be true that I do not understand Morse code, strong AI states that it is *the whole system* that does. Let's consider whether this reply is any more plausible now than in the original thought experiment. The whole system consists in a number of Morse keys, some scratch paper, a watch and a room. Just as in the CRA it seems implausible to assume that the combination of those elements might be the right kind of entity that could possess mentality. And just as in the original thought experiment we can dispose of everything except myself. I could memorise the rule book, do all computations in the head, use claps instead of Morse keys to communicate and operate outdoors. Nothing about the Morse code room makes the systems reply any more plausible.

The robot reply: Might equipping SNNs with sensors and actuators yields understanding? The idea seems plausible on first glance, since in combination with an appropriate training mechanism it might just yield the right kind of causal connection to the objects of the environment. Just as the human brain learns to understand Morse code by processing information derived from the senses, a SNN might learn what the spikes represent by acting in an environment which it perceives via its sensors. Since it might even use STDP based update rules, learning seems very much like in the human brain. So why wouldn't it be possible to learn Morse code in much the same way that people do? On a closer look, however, adding sensors and actuators yields no relevant changes. The fact that the clicking noises in the Morse code room might be produced by a microphone or a camera changes nothing about the fact that they are mere noise to me. Also feeding the output from the Morse keys into artificial limbs changes nothing. Even in combination with machines learning techniques all I am doing is manipulating acoustic symbols that mean nothing to me according to rules that are generated in a way that is completely meaning less from my point of view. Even the implementation of a leaning mechanism does not render the robot reply a good objection to the Morse code room argument.

The brain simulator reply: If we were to simulate the brain of a person that spoke Morse code down to the neuronal level (or even below) in an appropriate SNN, wouldn't we have to admit that such a system actually understands? What might justify attributing understanding to the one system but not the other? Unlike a digital computer or ANNs we

might even reproduce the precise spiking behaviour of the human brain. Also this reply seems tempting. But again it fails. The same program could be implemented in the Morse code room. We could imagine that I operate on a vast amount of speakers and Morse keys, representing the synapses of the human brain. When I receive an acoustic signal I look up which of the many Morse keys I need to press in what particular order and the entire system is build such that it outputs the right Morse code pattern in the end. I could be simulating the entire human brain by receiving signals, looking up the rules and pressing Morse keys. Still, I would not understand what any of the noises and clicks mean. Actually, the Morse code room is an even better counter argument to the brain simulator reply than the water-pipe brain since it is able to capture not only the static organisational architecture of the human brain, but also its dynamical behaviour. Instead of simulating just the order in which neurons are activated the Morse code room can also implement the exact timing of the neuronal spikes.

The systems, robot and brain simulator reply are no more of an objection against the Morse code room than they were against the CRA. However, there is another counter argument against the CRA that seems to be gaining additional force when applied to the Morse code room: *The luminous room argument*. The luminous room is an argument from analogy presented by Patricia and Paul Churchland (1990) that is to show that the CRA is unsound. The argument goes as follow: (1) If the CRA is sound, then so is an analogous argument, the luminous room argument. (2) The luminous room argument is obviously unsound. (C) Thus, the CRA is unsound as well. The conclusion obviously follows from the premises. But in order to see, if they are true, let me first present the luminous room.

Its formal structure is analogous to that of the CRA (see chapter section II), however, yielding not the conclusion that strong AI is false but that electricity is insufficient for light.

- (1) Electricity and magnetism are forces.
 - (2) The essential property of light is luminance.
 - (3) Forces by themselves are neither constitutive of nor sufficient for luminance.
 - (C) Electricity and magnetism are neither constitutive nor sufficient for light.
- (Churchland & Churchland 1990: 33)

Since we know that light is essentially an oscillating electromagnetic field, this argument is obviously unsound. However, having a similar formal structure is not enough to establish equivalence with the CRA. The Churchlands, thus, develop a thought experiment analogous to the Chinese room in order to justify premise (3). Imagine a man in a dark room holding a magnet in his hand and pumping it up and down. According to the laws of electrodynamics such a moving magnet would be emitting electromagnetic waves into the dark room. So far so good. But now imagine further that this man might argue: "Your theory of light tells me that a moving magnet emits electromagnetic waves and that electromagnetic waves are light. So

why then is the room still dark? Obviously your theory must be false. Forces by themselves are neither constitutive of nor sufficient for luminance." The Churchlands claim that this line of reasoning, though obviously flawed, is analogous to Searle arguing that strong AI must be false just because formal symbol manipulation *in the Chinese room* is unable to produce understanding. Magnet manipulation in the luminous room is not sufficient for light and symbol manipulation in the Chinese room is not sufficient for understanding. However, this proves neither that, forces are not sufficient for luminance nor that syntax is not sufficient for semantics. The reason why Searle doesn't understand is the same why the luminous room does not illuminate: the (thought) experimental setup is flawed. (see Churchland & Churchland 1990: 34)

Since the luminous room argument obviously is unsound the question is whether it also is analogous to the CRA. And although they seem alike I think that the answer is clearly "no". Strong AI claims that formal symbol manipulation alone constitute understanding. Electrodynamics, on the other hand, states that electromagnetic fields that oscillate *with the right frequency* are light. This with-the-right-frequency condition breaks the analogy. The man in the luminous room is simply wrong when asserting that electromagnetic waves are light. Only electromagnetic waves with a frequency *between 468THz and 789THz* are. According to electrodynamics the man in the room would have to pump the magnet at least 468 billion times per millisecond in order to produce light. Strong AI makes no such claim about the speed of computation. "[S]peed is strictly irrelevant here. A slow thinker should still be a real thinker. Everything essential to the duplication of thought, as per classical AI, is [...] present in the Chinese room." (Churchland & Churchland 1990: 34)

However, this seems no longer to be true for SNNs and the Morse code room. For systems that use temporal encoding time is relevant and the computation changes as we change the computation frequency. Let's remember how our Morse code represents a real number. A 3.2 for example is represented as the time interval between two spikes. If we were to increase or decrease the spiking frequency the number and, thus, the computation would change. The same is true for any other symbol that is encoded in the precise timing of spikes or clicks. Using temporal encoding seems to be adding the with-the-right-frequency condition to the thesis of strong AI, thus, keeping up the analogy to the Chinese room. Changing the frequency also changes the computation that is being implemented. Implementing the right kind of computation, therefore, seems to be equivalent to implementing the right kind of computation *with the right frequency*. The reason why I don't understand Morse code is not because computationalism is false but because I am way too slow. I don't implement the same computation as the brain of someone who understands Morse code since the human brain emits millions of spikes per second. Just as the luminous room does not light up because the man pumps the magnet too slow, the Morse code room does not understand because my clicking frequency is too low.

Speed, now, *is* relevant and a slow thinker implements a different computation than a fast thinker.

Although this line of reasoning seems tempting I am going to show that

for any high-frequency system (like the human brain or a SNN) there is a low frequency system (like the Morse code room) that *implements the same function*.

The reason for this is actually very simple. All one has to do is uniformly "stretch time". What I mean by this is that we need to change the interpretation when moving from the high- to the low-frequency system such that any time interval is multiplied by a given factor. For example, if a SNN encodes the real number 3.2 as a time interval between spikes of $3.2 \mu\text{s}$ than we can simply change the interpretation such that the Morse code room implements the same number as a time interval of 3.2s. The SNN spikes with a million times higher frequency than the Morse code room, however, both implement the same number. If we were to change the interpretation of all representations, not only that of real numbers, in the same way and let the Morse code room carry out the same patterns just a million times slower both systems, the SNN and the Morse code room, *implement the same computation. For any high-frequency system that implements a given computation under a certain interpretation there is a different interpretation such that a low-frequency system implements the same computation.* The luminous room, therefore, is *not analogous* to the Morse code room. We can dispose of the with-the-right-frequency clause in the latter, but not in the former. For any high-frequency system there is a low-frequency system that implements the same computation but if we change the frequency of a electromagnetic wave we change its constituting light.

VII Some more general considerations regarding hardware and software

So far we have used the term "neural network" ambiguously. It can either refer to a directed graph with sets of units and sets of connections, i.e. *a mathematical model*. Or it can mean physical system with electronic neurons connected with real synaptic connections. i.e. *a computing device*. Whenever we were asking whether the CRA applies to SNNs we were talking about the *physical hardware* and not the mathematical model. It makes no sense to say that an abstract model is parallel distributed, i.e. that it has many computational units that compute in parallel. Only physical devices can have spatially separated processors. Likewise, there is no meaning in asking whether a mathematical model uses digital or analogue representations. Representations are physical objects like cards, varying voltage levels or neuronal membrane potentials. There are digital and analogue computers but there are no digital or analogue models. And even if there might be a sense in which spiking *models* might be subsymbolic, capable of learning and employing temporal encoding the relevant differences

to digital computers all refer to the *hardware*. Meaning might emerge from an actual physical structure that is used for subsymbolic representation, certainly not from some abstract concept. Learning might yield the right *causal* connection to the environment, only if it is regarded as a physical process. Mathematical algorithms cannot connect a system to the environment. And temporal encoding might be relevant because it links the system to the *physical real time* of the environment in which it is placed. Abstract intervals are not the kind of entity that could be "in sync" with external time. Whenever we were considering the applicability of the CRA to SNNs we were talking about *spiking hardware*.

If we were to ask the question whether the CRA applies to *spiking models* the answer would be very simple. It does not matter which program the Chinese room implements, whether it is that of a bot answering questions in Chinese or that of a SNN simulation. All the processor of such a system does is manipulating meaningless symbols according to some rules. Even the most ambitious and bio-realistic models like BioSpaun (Eliasmith et.al. 2016), SpiNNaker2 (Mayer et.al. 2019) or the model of the Human Brain project (Markram et.al. 2015) use more or less standard digital hardware to simulate spiking models. The CRA applies trivially to them. All we have to do is change the program such that Searle no longer simulates a Chinese speaker but rather implements a brain model. Nothing about the general set up of the Chinese room changes.

On the other hand, *multiple realisability* states that the hardware is strictly irrelevant. It does not matter whether the same model is implemented on a digital computer, a human brain or a SNN. So if we ask whether the CRA applies to *spiking hardware* the answer again is trivially "yes". Strong AI is a thesis about the right kind of computation, i.e. the correct mathematical model. It is completely insensitive to matters of physical realisation.

So actually the question whether the CRA applies to spiking neural networks is trivial.²³ In my opinion, the reason why there has been so much confusion about this issue is that the terms "connectionist system", "neural network", etc. were used ambiguously as meaning both, mathematical models and physical devices. What cognitive scientists were doing is developing *abstract models* of the mind in order to find out what "the right kind of computation" is. But they used *physical features* to argue for the supremacy of the respective model. The Churchlands, for example, thought that the fact that *physical* neural nets are parallel distributed was an argument for the supremacy of the connectionist *model*. Chalmers argued that meaning might emerge from a certain pattern of *physical* representations. And that this was a reason to use subsymbolic *models*. Harnard argued that learning might yield the appropriate *causal* connection between computer and environment and that machine learning *algorithms*, therefore, were the key to mentality. Eliasmith thought that we had to adjust the time scale in which computers yield a result to the *physical real time* of the environment and that this was

²³Except for the possibility of analogue hardware that is able to implement abstract models that are not Turing-computable.

reason enough to chose spiking *models*. The ambiguity of the term "neural net" as meaning computational model *and* physical device is ubiquitous in cognitive science.

I think that most of the arguments discussed above actually are good arguments, and I believe that SNNs are more likely to be capable of creating understanding in the future than digital computers. The reason for this is however not that *spiking models* are better models of the mind but that *spiking hardware* is more biologically realistic. There is a growing field of so called *neuromorphic hardware* that is concerned with the development of physical devices that not only *simulate* but *replicate* the human brain in various respects including parallel processing, analogue representation and temporal encoding. Hardware implementations for SNNs that focused on bio-emulation in analogue electronics are a topic of recent research. (see Bouvier et.al. 2019) It think that cognitive science should take more interest in neuromorphic engineering and incorporate considerations regarding hardware into its theories. What the CRA shows is that implementing the right kind of computation alone is not sufficient for having a mind, but implementing the right kind of computation *in the right kind of hardware* just might be.

Conclusion

In the present paper I have addressed the question of whether the CRA can be applied to SNNs despite the several differences between them and classical digital computers. After some considerations regarding the theoretical background about computationalism, implementation and the Chinese room I have been discussing 5 major differences between SNNs and digital computers and their relation to the CRA. (1) Parallel processing. SNNs are parallel computers, i.e. they have multiple processing units. Just like the human brain they process information not in one location but distributed amongst many neurons. We found out that due to the Turing completeness of digital computers any neural network can be simulated on a standard hardware. The CRA can, therefore, be applied to parallel machines. (2) Subsymbolic computation. In SNNs the semantic and the symbolic level diverge, i.e. the computational tokens no longer carry meaning. In contrast to digital computers which operate over meaning full symbols, the basic computational units, viz. the neurons, only count as representations if used in entire layers. I have been arguing for the *irrelevance thesis*: The symbolic/subsymbolic distinction is irrelevant to cognitive science. For this the reason was that any system can be interpret as implementing symbolic *and* subsymbolic computations. We found that by letting Searle operate over Chinese numbers instead of symbols we can modify the CRA such that it applies to subsymbolic computation. (3) Machine learning. Unlike digital computers, SNNs usually have build-in machine learning mechanisms. Some have argued that this opens up the possibility the environment having a causal impact on the system and, thus, endowing computational symbols with meaning. In order to account for machine learning we introduced a training mechanism into the Chinese room, telling Searle when to update his rule book. We found that even after successful training, Searle understands no more than in the original Chinese room. The CRA, thus, can be applied to systems with machine learning mechanisms. (4) Analogue representations. We have been discussing two understandings of the term "analogue". First, in the sense of continuous representation and, second, as meaning representation by physical primitives. We saw that SNNs are not necessarily analogue in the second sense and, thus, focused on the first understanding. We found out that the CRA can be applied to analogue computers only under the condition that either the maximality or the simulation thesis is true. That means that either SNNs and digital computers are computationally equivalent or that the brain processes relevant for cognition are Turing computable. Both claims are likely to be true. If one of the two is the CRA is applicable to analogue computers such as

SNNs. However, there are some restrictions regarding analogue computing to keep in mind. Additionally, I have been arguing for the *irrelevance thesis II*: The digital/analogue distinction is irrelevant to cognitive science. The reason for this was once more that any analogue system can be interpreted digitally and vice versa. Unlike the CRA, the irrelevance thesis holds universally for all computing devices. (5) Temporal encoding. SNNs do not use static system parameters such as voltage levels but rather encode information in the timing of spikes. This makes them attractive for dynamical approaches to cognitive science since it defines a time scale over which computation occurs. However, we have found out that for any high-frequency system there is a low frequency system that implements the same computation. Regarding computational issues this time scale, thus, becomes irrelevant and the CRA can be applied to systems that use temporal encoding. We used the *Morse code room* thought experiment to illustrate how temporal encoding can be implemented in the Chinese room. Several replies against the Morse code room have been considered. In summary, we conclude that, *apart from considerations regarding possible super-Turing computational power, the Chinese room argument can be applied to spiking neural networks.*

In the last section we have been discussing some general remarks regarding hardware and software. We found out that many of the confusions in cognitive science can be traced back to the ambiguous use of SNN and connectionist network in general, meaning either physical hardware device *or* abstract computational model. I have been arguing that most objections to the CRA use arguments concerning *hardware* issues to motivate new *models* of the mind. I consider many of those arguments to be good arguments. However, they do not support strong AI but rather the different claim *that implementing the right kind of computation in the right kind of hardware is sufficient for mentality.* This points the way to a new understanding of AI which focuses more on neuromorphic hardware in addition to brain-like computational models. In my opinion, a computational system that not only simulates but rather replicates the human brain in important respects is far more likely to reach the general intelligence of an average human being.

Bibliography

- [1] Amin, H. (2006). Spiking Neural Networks Learning, Applications, and Analysis.
- [2] Anyoha, R. (2017). The History of Artificial Intelligence. *Harvard University*: <https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/> (Accessed: 01.08.2022)
- [3] Baytekin, H. T., & Akkaya, E. U. (2000). A molecular NAND gate based on Watson Crick base pairing. *Organic Letters*, 2(12): 1725-1727.
- [4] Bechtel, W., & Abrahamsen, A. (1993). Connectionism and the mind: An introduction to parallel processing in networks. *Basil Blackwell*.
- [5] Bickle, J. (1998). Multiple realizability. *The Stanford Encyclopedia of Philosophy*, (Summer 2020), Edward N. Zalta (ed.)
- [6] Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning. *Springer*.
- [7] Bre, F., Gimenez, J. M., & Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and buildings*, 158: 1429-1441.
- [8] Chalmers, D. J. (1992). Subsymbolic computation and the Chinese room. *The symbolic and connectionist paradigms: Closing the gap*: 25-48.
- [9] Chalmers, D. J. (1994). On implementing a computation. *Minds and Machines*, 4(4).
- [10] Chalmers, D. J. (1995). Absent Qualia, Fading Qualia, Dancing Qualia. in *Conscious Experience*. Thomas Metzinger (ed.): 309-328.
- [11] Chalmers, D. J. (2011). A computational foundation for the study of cognition. *Journal of Cognitive Science*, 12(4): 325-359.
- [12] Chalmers, D. J. (2016). The singularity: A philosophical analysis. *Science fiction and philosophy: From time travel to superintelligence*: 171-224.

-
- [13] Churchland, P.M. (1989). A Neurocomputational Perspective: The Nature of Mind and the Structure of Science, *MIT Press*.
 - [14] Churchland, P. M., & Churchland, P. S. (1990). Could a machine think?. *Scientific american*, 262(1): 32-39.
 - [15] Clark, A. (1989). Microcognition: Philosophy, cognitive science, and parallel distributed processing. *MIT Press*.
 - [16] Cole, D. (2004). The Chinese room argument. *The Stanford Encyclopedia of Philosophy*, (Winter 2020), Edward N. Zalta (ed.)
 - [17] Copeland, B. J. (1997). The church-turing thesis. *The Stanford Encyclopedia of Philosophy*, (Summer 2020), Edward N. Zalta (ed.)
 - [18] Crane, T. (1996). The Mechanical Mind: A Philosophical Introduction to Minds, Machines and Mental Representation. *Penguin*.
 - [19] De Mol, L. (2018). Turing machines. *The Stanford Encyclopedia of Philosophy*, (Summer 2020), Edward N. Zalta (ed.)
 - [20] Eliasmith, C. (2003). Moving beyond metaphors: Understanding the mind for what it is. *The Journal of philosophy*, 100(10): 493-520.
 - [21] Eliasmith, C.; Stewart, T.C.; Choo, X.; Bekolay, T.; DeWolf, T.; Tang, Y.; Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science*. 338 (6111): 1202–1205.
 - [22] Eliasmith, C., Gosmann, J., Choo, X. (2016). BioSpaun: A large-scale behaving brain model with complex neurons. *arXiv preprint arXiv:1602.05220*.
 - [23] Fodor, J. A. (1980). Searle on what only brains can do. *Behavioral and Brain Sciences*, 3(3): 431-432.
 - [24] Hertz, J., Krogh, A., & Palmer, R. G. (2018). Introduction to the theory of neural computation. *CRC Press*.
 - [25] Goodman, N. (1968). Languages of art. *Hackett Publishing*.
 - [26] Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3): 335-346.
 - [27] Jacob, P. (2003). Intentionality. *The Stanford Encyclopedia of Philosophy*, (Winter 2019), Edward N. Zalta (ed.)
 - [28] Lewis, D. (1971). Analog and digital. *Nous*: 321-327.

-
- [29] Maley, C. J. (2011). Analog and digital, continuous and discrete. *Philosophical Studies*, 155(1): 117-131.
- [30] McCulloch, M. & Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, (5): 115–133.
- [31] Nageswaran, J.M., Dutt N., Krichmar, J. L., Nicolau, A., Veidenbaum, A.V. (2009). A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors. *Neural Networks*, Volume 22, Issues 5–6: 791-800.
- [32] Northoff, G., Wainio-Theberge, S., & Evers, K. (2020a). Is temporo-spatial dynamics the “common currency” of brain and mind? In Quest of “Spatiotemporal Neuroscience”. *Physics of Life Reviews*, 33: 34-54.
- [33] Northoff, G., Wainio-Theberge, S., & Evers, K. (2020b). Spatiotemporal neuroscience—what is it and why we need it. *Physics of Life Reviews*, 33: 78-87.
- [34] Novella, S. (2015). AI and the Chinese room argument [Online image]. <https://theness.com/neurologicablog/index.php/ai-and-the-chinese-room-argument/> (Accessed: 01.08.2022)
- [35] Oppy, G. and Dowe, D. (2003) The Turing Test. *The Stanford Encyclopedia of Philosophy*, (Winter 2021), Edward N. Zalta (ed.)
- [36] Piccinini, G. (2010). The resilience of computationalism. *Philosophy of Science*, 77(5): 852-861.
- [37] Ponulak, F., & Kasinski, A. (2011). Introduction to spiking neural networks: Information processing, learning and applications. *Acta neurobiologiae experimentalis*, 71(4): 409-433.
- [38] Putnam, H. (1967). Psychophysical Predicates, in: *Art, Mind, and Religion*, W. Capitan and D. Merrill (eds), *University of Pittsburgh Press*: 429–440.
- [39] Rescorla, M. (2015). The computational theory of mind. *The Stanford Encyclopedia of Philosophy*, (Fall 2020), Edward N. Zalta (ed.)
- [40] Sandberg, A., & Bostrom, N. (2008). Whole Brain Emulation: A Roadmap, Technical Report 2008-3, *Future of Humanity Institute*, Oxford University.
- [41] Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, 3(3): 417-424.
- [42] Searle, J. R. (1983). Intentionality: An essay in the philosophy of mind. *Cambridge university press*.

- [43] Searle, J. R. (1990a). Is the brain a digital computer?. *Proceedings and addresses of the American Philosophical Association*, Vol. 64, No. 3: 21-37.
- [44] Searle, J. R. (1990b). Is the brain's mind a computer program?. *Scientific American*, 262(1), 25-31.
- [45] Searle, J. R. (1992). The rediscovery of the mind. *MIT press*.
- [46] Searle, J.R. (2004). Mind: A brief introduction. *Oxford university press*.
- [47] Shallit, J. (1995) A very brief history of computer science. *University of Waterloo*: <https://cs.uwaterloo.ca/~shallit/Courses/134/history.html> (Accessed: 01.08.2022)
- [48] Sharkey, N. E., & Ziemke, T. (2001). Mechanistic versus phenomenal embodiment: Can robot embodiment lead to strong AI?. *Cognitive Systems Research*, 2(4): 251-262.
- [49] Siegelmann, H. T. (1995). Computation beyond the Turing limit. *Science*, 268(5210): 545-548.
- [50] Siegelmann, H. T. (2003). Neural and super-Turing computing. *Minds and Machines*, 13(1): 103-114.
- [51] Siegelmann, H. T., & Sontag, E. D. (1994). Analog computation via neural networks. *Theoretical Computer Science*, 131(2): 331-360.
- [52] Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and brain sciences*, 11(1): 1-23.
- [53] Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T. and Maida, A. (2019). Deep learning in spiking neural networks. *Neural networks*, 111: 47-63.
- [54] Thomson-Jones, K., & Moser, S. (2015). The philosophy of digital art. *The Stanford Encyclopedia of Philosophy*, (Spring 2021), Edward N. Zalta (ed.)
- [55] Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *J. of Math* 58.345-363.
- [56] Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, 59 (236): 433-460.
- [57] Van Gelder, T. (1998). The dynamical hypothesis in cognitive science. *Behavioral and brain sciences*, 21(5): 615-628.
- [58] Vreeken, J. (2003). Spiking neural networks, an introduction.

Selbstständigkeitserklärung

Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich reiche sie erstmals als Prüfungsleistung ein. Mir ist bekannt, dass ein Betrugsversuch mit der Note "nicht ausreichend" (5,0) geahndet wird und im Wiederholungsfall zum Ausschluss von der Erbringung weiterer Prüfungsleistungen führen kann.

Johannes Brinz

Dresden, August 2022