Pascal Gerhards
Deriving an Efficient Processing Chain for Radar
With Spiking Neural Networks

TUD*press*

Pascal Gerhards

Deriving an Efficient Processing Chain for Radar With Spiking Neural Networks

TUD*press*
2025

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

ISBN 978-3-95908-731-5

Thelem Universitätsverlag & Buchhandlung GmbH & Co. KG
http://www.thelem.de

TUDpress ist ein Imprint von Thelem

Gesetzt vom Autor.
Printed in Germany.

Technische Universität Dresden

Deriving an Efficient Processing Chain for Radar With Spiking Neural Networks

M. Sc.
Pascal Gerhards

der Fakultät Elektrotechnik und Informationstechnik der Technischen Universität Dresden

zur Erlangung des akademischen Grades

Doktoringenieur
(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender:	Prof. Dr.-Ing. Rafael Felix Schaefer	Tag der Einreichung	08.04.2024
Gutachter:	Prof. Dr.-Ing. habil. Christian Georg Mayr	Tag der Verteidigung	23.10.2024
Gutachter:	Prof. Dr. Federico Corradi		

Acknowledgements

First, I would like to thank Prof. Dr. Mayr for supervising my thesis and giving me the opportunity to work with the SPiNNaker 2 system, as well as his guidance during my PhD. I also thank Prof. Dr. Federico Corradi for taking over the second referee of my thesis.

I also want to thank my colleagues at Infineon's Development Center in Dresden. In particular, I wish to thank Uwe Gäbler for providing the opportunity to pursue my research in the company and Dr. Klaus Knobloch for our fruitful discussions, his insights, and feedback throughout the whole thesis, without, this work wouldn't have been possible. Furthermore, thank you Felix Kreutz, Jiaxin Huang, Daniel Scholz, and Julian Hille for all our discussions and collaboration on the implementation of SNNs in TensorFlow and on SpiNNaker 2 as well as capturing the radar gesture set. Also thanks to Dr. Evgeny Kusmenko and Mattis Hoppe for proof-reading my thesis.

I furthermore thank Martin Weih, Bernhard Vogginger, and Florian Kelber for their support with implementing neural networks on SpiNNaker 2 and setting up the energy measurements.

My special thanks goes to Carmen Weigelt for her unconditional support, fruitful discussions, and proof-reading my thesis.

Last but not least, I wish to thank Julia Stirnat and Martin Striegler for proof-reading my thesis and my family and friends for their support throughout the years.

Abstract

Artificial Intelligence significantly enhanced signal processing for radars and achieves unmatched results in classification performances and resolution, such as object detection or angle of arrival estimation. However, deep neural networks and especially recurrent Long Short-Term Memory networks for time dependent tasks require large amounts of energy for execution. Energy is a limited resource for battery driven devices such as automobiles, drones, smartphones, and smart watches which requires best energy efficiencies to increase their lifetime. Brain-inspired neuromorphic computing promises high energy efficiencies by sparse, event-based computation and communication with spiking neural networks. The aim of this work is to find optimal neural network architectures and preprocessing techniques for accurate and energy efficient processing chains for radar applications with the example of gesture recognition. Different network architectures were first trained in TensorFlow with a standard desktop computer on two pre-processed radar-based gesture recognitions datasets; one self-recorded and one publicly available. Afterwards, networks were evaluated for classification performance and computational complexity as estimate of their energy requirements. Then, networks were mapped to the SpiNNaker 2 neuromorphic hardware to evaluate their accuracy and energy costs at the edge. Furthermore, simulations were performed to analyze which network is most efficient depending on network sizes and input sparsity. Lastly, experiments were conducted to analyze whether neural networks can substitute traditional pre-processing techniques. For the given datasets, spiking neural networks achieved competitive accuracies to reference deep recurrent neural networks while requiring up to two orders of magnitude less operations. A combination of classical networks with brain inspired spiking networks achieved slightly better accuracies than the spiking neural network at the cost of additional complexity. Energy measurements revealed that spiking networks are most efficient; closely followed by hybrid networks with one third larger energy consumptions. Simulations revealed that the efficiency of spiking neural networks is correlated with their input activity and are Simulations revealed that spiking networks are most efficient when less than 13 % of all inputs emit spikes. At larger rates, hybrid networks are most efficient. Experiments with time-domain radar data revealed that neural networks achieve slightly worse results than networks based on pre-processed data. Best case estimations of their energy consumptions showed that they further have slightly worse efficiencies. It is concluded, that spiking networks are recommended when data is sparse. Otherwise, hybrid neural networks are better than deep neural networks and spiking network. Therefore, replacing recurrent layers of deep neural networks with spiking neuron layers is beneficial for both energy efficiency and accuracy, in radar-based gesture recognition. While neural networks also showed good results for time-domain processing, it is not recommended to substitute the standard FFT pre-processing due to the loss in accuracy and efficiency compared to the standard approach. This thesis demonstrated the high potential of brain inspired computing for radar processing on battery driven devices with increased runtime, i.e. lower energy consumption, and classification performance.

Zusammenfassung

Künstliche Intelligenz hat die Signalverarbeitung von Radardaten signifikant verbessert und dabei höchste Präzision erreicht, wie zum Beispiel bei der Winkelabschätzung oder der Objekterkennung. Auf der Kehrseite benötigen neuronale Netzwerke und insbesondere rekurrente Netze, die für zeitabhängige Aufgaben benutzt werden, Unmengen an Energie. In batteriebetriebenen Geräten, wie zum Beispiel Smartphones, Smart Watches oder Elektroautos ist Energie jedoch limitiert. Die Effizienz der Netze hat also unmittelbaren Einfluss auf die Lebensdauer beziehungsweise Reichweite. Neuromorphes Rechnen, das die Funktionsweise des menschlichen Gehirns imitiert, verspricht besonders hohe Energieeffizienzen durch eventbasierte pulsverarbeitende neuronale Netze. Das Ziel dieser Arbeit ist es optimale Netzwerkarchitekturen und Datenvorverarbeitungsmethoden zu ermitteln, die für Radar-gestützte Gestenerkennung maximale Klassifizierungsgenauigkeit und Energieeffizienz erreichen. Die Netzwerke wurden erst in TensorFlow auf einem üblichen Computer trainiert. Es wurde sowohl ein öffentlich verfügbarer Datensatz als auch einen selbst aufgenommener Datensatz benutzt. Die Netze wurden dann hinsichtlich ihrer Genauigkeit und ihrer Komplexität, die eine Abschätzung der benötigten Energie erlaubt, untersucht. Anschließend wurden die Netze auf SpiNNaker 2, einer digitalen neuromorphen Platform, transferiert und hinsichtlich des realen Energieverbrauchs evaluiert. Außerdem wurden Simulationen durchgeführt, die analysieren wie Netze verschiedener Größen und Eingangsdaten skalieren. Weiterhin wurden Experimente durchgeführt, die untersuchten ob die standard Vorverarbeitung der Radardaten durch neuronale Netze ersetzt werden kann. Für die beiden vorliegenden Datensätze erreichten die pulsverarbeitenden Netze jeweils vergleichbare Genauigkeiten wie die rekurrenten Netzwerke, brauchten dabei jedoch ein bis zwei Größenordnungen weniger Rechenoperationen. Eine Kombination aus traditionellen und pulsverarbeitenden Netzen erreichte noch bessere Genauigkeiten als die reinen Pulsverarbeitenden, jedoch zu Kosten der Energieeffizienz. Die Energiemessungen auf SpiNNaker 2 bestätigten, dass pulsverarbeitende Netze die höchste Effizienz aufweisen, wobei die hybriden Netze nur knapp schlechter abschneiden. Die Simulationen ergaben, dass die pulsverarbeitenden Netzwerke nur effizienter sind, wenn die Eingangsneuronen maximal zu 13 % aktiv sind. Bei höheren Raten sind dann die hybriden Netze am effizientesten. Die Experimente mit Radarrohdaten ergaben, dass die Netze zwar hohe Klassifizierungsgenauigkeiten erzielen, aber weder so präzise noch so effizient sind wie solche Netze, die vorverarbeitete Daten nutzen. Es wird daher nicht empfohlen die Vorverarbeitung durch neuronal Netze zu ersetzen. Weiterhin können folgende Schlussfolgerungen gezogen werden: Pulsverarbeitende Netze werden empfohlen, wenn die Eingangsdaten spärlich sind, also wenige Events produzieren. Andernfalls sind hybride Netze zu empfehlen, da diese genauere Klassifizierungen erlauben und gleichzeitig effizienter sind. Die Ergebnisse dieser Arbeit zeigten das große Potential von pulsverarbeitenden Netzwerken für die Verarbeitung von Radardaten und demonstrieren, dass batteriebetriebene Geräte dadurch höhere Laufzeiten erreichen, mit anderen Worten weniger Energie verbrauchen, und dabei hohe Erkennungsgenauigkeiten aufweisen.

Contents

List of Figures

List of Tables

Glossary

ADC	Analog to Digital Conversion
AI	Artificial Intelligence
ASIC	Application Specific Integrated Circuit
CFAR	Constant False Alarm Rate
CLSTM	Convolutional Long Short-Term Memory
CNN	Convolutional Neural Network
DL	Deep Learning
DMA	Direct Memory Access
DML	Deterministic Maximum-Likelihood
DNN	Deep Neural Network
ESPRIT	Estimation of Signal Parameters via Rotational Invariant Techniques
FFT	Fast Fourier Transform
FLOP	Floating Point Operation
FMCW	Frequency-Modulated Continuous Wave
HNN	Hybrid Neural Network
IFD	Infineon Dresden
LIF	Leaky Integrate-and-Fire
LSTM	Long Short-Term Memory
MAC	Multiply ACcumulate
MLA	Machine Learning Accelerator
MPT	Master Population Table
MUSIC	MUltiple SIgnal Classification
NoC	Network on Chip
PE	Processing Element
QPE	Quad Procesing Element
RAM	Range-Angle Map
RDM	Range-Doppler Map

ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SNN	Spiking Neural Network
VMM	Vector Matrix Multiplication

1 Introduction

Artificial Intelligence (AI) is considered as the next revolution [1]. It has and is undoubtedly continuing to change our life. Huge networks like Chat-GPT from OpenAI have millions of users, rising constantly [2]. Afar from such widely known applications, AI is used in many more areas. One such use-case is signal processing without which modern technology would not be possible. The following sections give an overview why and how AI is used for signal processing in general, how the recent Spiking Neural Networks (SNNs), heavily inspired by the efficiency of the human brain, can overcome some of the drawbacks inherent to Deep Neural Networks (DNNs), and finally, how AI can enhance the radar processing chain.

1.1 AI-enhanced Signal Processing

In order to discuss where, why, and how AI is able to improve signal processing, it is required to understand the basics and limitations of traditional signal processing. The following section first gives an overview why signal processing is so important in our every-day life and how it is defined.

Signal processing is a crucial part of life and modern technology in various applications. Today's children might not be able to imagine a world without televisions, telephones or smartphones, the internet, basically autonomously flying planes, or voice activated digital assistants like Siri or Alexa. All these and many more applications would not exist without signal processing. They have in common that humans want to interact with each other or the environment. For that, it is required to capture information, analyze and finally utilize it.

All life forms have the ability to capture and process various types of signals in order to navigate through the world and communicate with each other. For instance, when listening to sound sources, the ears measure pressure differences in the air. The measured signals are transformed into voltage pulses that are sent to the brain. The brain then extracts the time difference of the signal's arrival at both ears as well as the intensity difference to locate the sound source [3]. With technological advancements, humanity adapted and created

artificial signal processing that enables today's modern lifestyle, such as watching television or listening to podcasts. Digital signals of video and audio are captured by cameras and microphones, respectively, then, signal processing is used to compress and transmit those signals over wireless or wired connections to a receiving device, decompress it and display the video or playback the audio stream. Besides signals that humans can directly sense, for instance vision and hearing, humanity extended signal processing capabilities to many more signals, such as radar [4, 5] and sonar [4].

Signal processing can be generally described as the study and manipulation of signals in order to retrieve relevant information or features [6]. Various kinds of techniques were developed for different types of signals, applications and the required level of detail. While some signals are easy to analyze with one or few simple processing steps, other signal types require more and sophisticated processing steps. For example, estimating the direction of arrival of an object with a radar sensor can be done with simple but imprecise methods like a Fourier transform or with complex and resource intensive but accurate techniques such as the MUltiple SIgnal Classification (MUSIC) [7] or Deterministic Maximum-Likelihood (DML) algorithms.

Signal processing has undoubtedly advanced over the last centuries, achieved astonishing results, and realized amazing technologies [8, 6]. However, signal processing certainly has its limitations [9]. One is the difficulty to predict future changes based on current data. A good example is medical diagnosis. Today's methods can easily diagnose illnesses, such as diabetes by performing and analyzing tests and screenings. But it is nearly impossible with traditional processing techniques to predict if the diabetes test gets positive in the future which could save both lives and costs for treatment. Deep learning can significantly enhance the detection, prediction, and risk assessment [10]. A second limitation is the vulnerability to noise. Under noisy conditions, traditional algorithms sometimes fail to detect a weak signal that is only slightly stronger than the noise even when humans can see them with bare eyes [11]. But such low signal-to-noise regimes do occur regularly in real-world applications. One way to deal with such conditions is time averaging of the signal which is also done by human's subconscious [11]. However, time averaging is not feasible for all algorithms. For instance MUSIC requires hundreds or thousands of snapshots which is too slow for real-time predictions in automotive angle of arrival estimations. Another limitation is the runtime of an algorithm. Many applications require precise results and real-time operation for example for safety reasons like in autonomous driving. However, oftentimes a precise, high resolution algorithm is very resource intensive and is not real-time capable. The MUSIC and DML algorithms for high resolution direction of arrival with radar mentioned above are good examples for his. While achieving very good results, they are slow to compute and require data from multiple measurements which adds even more latency [12].

The previous paragraphs explained what signal processing is and why it is important as well as where it has its limitations. In the following, it is outlined why and how AI and more specifically DNNs, can improve certain aspects of signal processing.

With technological advancements in parallel computing devices, such as GPUs, AI had its breakthrough with the introduction of AlexNet that outperformed state of the art approaches for the ImageNet benchmark [13] by a good margin [14]. Since then, DNNs have been applied to many different problems and proved that they can solve some tasks better than classical signal processing, for example signal denoising of seismic signals [15], images [16], or lidar signals [17]. They even solve tasks that cannot be solved by classical processing techniques due to missing mathematical models, such as predictions of future medical conditions [10], predictive maintenance [18, 19, 20], or time series classifications [21] and predictions [22]. DNNs can do this because they are basically universal function approximators when they are sufficiently large [23]. This is due to their structure of layered units, commonly called neurons, that integrate weighted information from the previous layer and apply a non-linearity after integration. By algorithmically adjusting these weights to minimize the difference between the output of a given input and its desired output, it is possible to learn the underlying function that maps inputs and outputs. This eliminates the need of highly trained specialists with large domain knowledge, because the network learns the important features on its own [24, 25]. Continuous adjustment of parameters during operation also enables automatic adjustment to new conditions such as aging of sensors and actuators which leads to discrepancies in static settings. Another advantage of DNNs is that it uses Multiply ACcumulate (MAC) operations and activation functions. Especially the MACs are easy to accelerate on hardware which allows fast and relatively low energy execution, in contrast to some complex traditional algorithms that lack of efficient hardware acceleration.

Deep neural networks bring another opportunity rather than supporting traditional signal processing. Due to their capability to learn basically any relation between input and output data, they can be used to totally replace whole processing chains with multiple complex processing steps. For example deep learning is already used for end-to-end video segmentation. A single DNN can be used to detect, locate, and classify objects in each single frame and track them over multiple frames [26]. DNNs for such tasks typically do not use simple feed-forward layers but more specialized operations. Most famous is certainly the Convolutional Neural Network (CNN) that was also used in the above mentioned AlexNet [14]. CNNs are performant feature extractors from structured 2D data, such as images. They use so-called filter kernels of certain sizes that perform convolutions on individual image parts and is shifted over the whole data. Because the filter uses the same weights for all locations, it has only few trainable parameters. Also, the learned features are translation invariant which means they are found anywhere in the image, no matter its location. This is basically impossible with feed-forward networks and is the reason why CNNs are nowadays in many DNNs in some way or another. CNNs are also capable to detect features over time in time series problems [27, 28, 29]. However, Recurrent Neural Networks (RNNs), and its variants Long Short-Term Memory (LSTM) [30] and gated recurrent unit [31], as well as the recently developed transformer networks [32] were specifically developed to target tasks with an additional time complexity.

It is safe to say that AI has achieved astonishing results. However, there is still much room for improvement. One major concern of deep learning is its power requirements. Training of neural networks costs a lot of power. While small and medium sized networks can be trained on consumer GPUs, large attention networks like the one used for Chat-GPT require huge servers and days to months of training time [32]. But even inferences can be expensive to perform because of the large servers required to run models like Chat-GPT [33]. In energy-critical environments like battery-powered devices or electric vehicles, every bit of power matters in terms of time to recharge or its range. Thus, it is crucial to not only achieve best results but also reduce the inference power to a minimum. Many machine learning accelerator hardware blocks have been reported in the last decade with increasing throughput per Watt, for instance shown by Sanchez-Flores et al. [34] for RISC-V based architectures. These accelerators usually use integer arithmetic because it is much more energy efficient than floating point operations. Therefore, networks are typically quantized to a fixed-point integer format that allows inference with reduced bit width and enhanced energy efficiency [35].

Technology advances every year with more performance and less energy consumptions. However, deep learning is yet far away from efficiencies seen in biology. For instance, the human brain shows unmatched general computing performance with a power budget equivalent of around 20 W [36, 37, 38]. It achieves this by efficiently encoding information in the precise timing of short voltage pulses as well as massively parallel computing [39]. Researchers and engineers started already in the late 1990s with the investigation of spiking neural networks for computer science [40, 41]. Energy efficiency gains of up to 100-fold compared to classical neural networks were reported [41]. However, due to hardware limitations, there was a stronger focus on DNNs which were much easier to simulate and train compared to SNNs. With the advancement of technology and the applications of surrogate functions to overcome the non-differentiability of spiking neurons, SNNs gained again strong interest in the last decade. Many works demonstrate that the gap between DNNs and SNNs is closing and SNNs are even outperforming classical DNNs [42, 43, 44]. SNNs are inherently recurrent which makes them more suited for temporal tasks rather than "static" tasks such as image recognition [43]. For example, when a person should be tracked in a video sequence (cf. Fig. 1.1), the DNN has to evaluate each pixel of the frame to extract the exact location of the person. This processing also includes the processing of objects which are not moving and not important. In contrast, the SNN focuses on the changes between two frames and only processes changes and ignores redundant information.

SNNs promise huge boosts in energy efficiency and latency compared to DNNs. But to fully unleash their potential, new hardware concepts are required that support great locality, sparse communication, and near-memory computing. Especially the latter is a major bottleneck in classical von-Neumann architectures, i.e computers and smartphones with separated processors and memory. Computing performance of processors and speed as well as size of memory greatly improved during the last decades, following Moore's law [46]. The interconnect between them, however, did not experience the same improvements. Tricks, such as smart pipelining, circumvent this bottleneck to a certain degree but do not

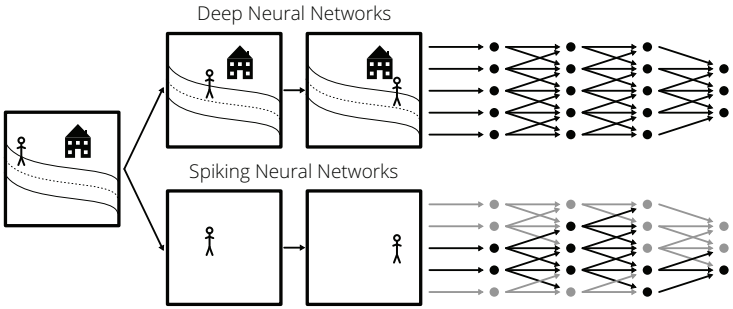

Figure 1.1: Processing paradigms of DNNs and SNNs on the example of video sequence. DNNs process each frame in its entirety no matter how much static content is present. SNNs focus on changes over time by omitting static, irrelevant parts as shown by gray parts that are inactive connections. Based and adapted from GrAI Matter Labs as shown by Fig.8 in [45].

solve it. The brain on the other hand couples memory and compute unit which removes latencies and energy overheads. Based on this model, neuromorphic computing architectures aim to imitate this behavior. Since the 2010s, a multitude of neuromorphic hardware platforms have emerged, each with its own design choices and strengths. Choices are, for example, the systems size, ranging from flexible large scale systems to small-scale Application Specific Integrated Circuit (ASIC), their used technology, i.e. digital, analog, or mixed signal, or supported functionality, such as static synapses and neurons, fully programmable neuron and synapse functions, or on-chip learning (also called online or on-line learning). In the following some of the commonly known architectures and their design choices are shown.

First, full digital systems are discussed. Fully digital systems simulate both neuron and synapse functionality. This simulation or emulation approach gives huge flexibility because neuron populations can be programmed individually. SpiNNaker 1 [47] and 2 [48], Loihi 1 [49] and 2 [50], TrueNorth [51], ODIN [52], Tianjic [53, 54], and μBrain [55] are all fully digital. μBrain is a fully event-driven ASIC without a global clock that is synthesized for each individual network which means that the chip itself is a representation of the network. Therefore, it can only execute this single network. On the other hand, it is highly optimized for that network. This optimization enables low latency and energy consumption. The authors claim fast development-to-deployment with their synthesis from a given network but after production no adaption is possible [55]. ODIN is another small-scale digital system with spike-driven, non-programmable plasticity. Those small-scale systems are limited in terms of network size and flexibility but offer the largest efficiencies and are thus the best when low latency and low energy consumption is most important. The other platforms are middle to large-scale systems. This means that they are designed for scalability with many chips per board and the option to interconnect boards to create systems with millions of cores. Tianjic is a platform that is designed with hybrid architectures in mind. By

that means, it features flexible and reconfigurable building blocks that allow the execution of DNNs, SNNs, and mixtures of both. It is scalable to support larger networks but does not feature on-chip learning. TrueNorth is a many-core on-chip system. Neuron cores are interconnected either on a single chip or between multiple boards in a grid layout. This allows simulation of networks with millions of neurons. TrueNorth does not support on-line learning. The SpiNNaker and Loihi platforms are similar systems with fully-programmable processing cores per chip, large-scale connectivity potential, and the support of on-chip learning. Differences include but are not limited to their spike routing and communication, chip connectivity and synchronization, and included hardware accelerators, such as the MAC arrays of SpiNNaker 2 that allow efficient acceleration of SNNs as well as DNNs.

Mixed-signal systems are another often used approach for neuromorphic platforms. They are used for small-scale systems such as ROLLS [56] to large-scale systems like NeuroGrid [57], Braindrop [58], HiAER-IFAT [59], DYNAPS [60], and BrainScales 2 [61]. Like the digital systems, also mixed-signal systems differ in their functionality. From the above mentioned, BrainScales 2 supports on-line learning with fully programmable and configurable plasticity while ROLLS supports only non-programmable plasticity. Analog synapses enable low-power operation with short latencies. BrainScales 2 for instance can accelerate SNN simulation up to 10^3-fold compared to real-time, enabling large-scale simulations of biological systems [61].

For this thesis, the SpiNNaker 2 platform is chosen as exemplary target platform because of its flexibility and ease of use which is provided by using standard ARM cores, and its MAC array for accelerating convolutions and matrix multiplications. These properties make SpiNNaker 2 well suited for evaluating network architectures to find efficient processing chains for radar with SNNs.

1.2 Radar Processing

The previous section outlined why AI and especially SNNs are interesting and powerful tools to enhance signal processing in general. This section motivates the importance of radar as sensor technology and the challenges of processing radar data. Finally, it discusses how SNN-based AI could potentially enhance the radar processing chain.

Radar is a sensing technology that relies on electromagnetic waves with typical frequencies between 1 GHz and 100 GHz. Using these frequency ranges has several advantages over other sensing technologies like cameras. First, radar can precisely and simultaneously measure distance, velocity, and angle of multiple objects with ranges of up to hundreds of kilometers [62]. Second, radar works independently of light and weather conditions [63, 64, 65]. Third, radar waves can penetrate thin, non-metallic objects and can thus easily be hidden behind plastic covers [66, 63]. Fourth, it is privacy conform [63]. And fifth, it requires very little power which makes it suitable for battery-powered devices such as handhelds and electric vehicles [67]. For that reasons, radar is used in various applications such as maritime [68], aviation [69], automotive field [70, 12, 71], remote monitoring of human vital signs [72], and human machine interfaces [67, 63, 73, 74, 75].

The data from a radar device is typically a complex valued signal that is retrieved from subtracting the emitted wave from the received wave. This signal contains all information about distance, velocity, and angle of any object in front of the device. To extract this information, several processing steps are required. A typical processing chain contains two Fast Fourier Transforms (FFTs) to extract ranges and velocities, angle calculation, and target detection, classification, and tracking, as shown in Fig. 1.2. The final result is called object list which contains all detected objects with information about their distance, velocity and angle.

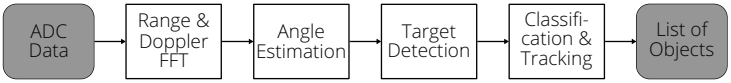

Figure 1.2: Simplified radar processing chain from radar data to a full list of objects and their properties such as range, velocity, and angle of arrival. Adapted from [65].

Radars are known and used for decades and many signal processing techniques of varying complexity have been successfully applied depending on the requirements of the task. Nevertheless, radar signal processing still faces several challenges. For instance, the processing chain is relatively long with more than four individual steps whose inputs and outputs are basically not interpretable by humans. Also, traditional signal processing techniques have intrinsic limitations imposed by the radar devices, such as the resolution of the direction of arrival that is limited by the size of the antenna array. Super resolution algorithms such as MUSIC, Estimation of Signal Parameters via Rotational Invariant Techniques (ESPRIT) [76], and DML use clever techniques to artificially increase the resolution, circumventing the limited resolution to a certain degree. These algorithms on the other hand are mostly expensive to calculate. Either because each target is tested whether it is a single target or multiple close targets (DML) or because a covariance matrix has to be calculated and analyzed (MUSIC, ESPRIT). These operations are typically not suitable for real-time execution.

This is where AI offers potential solutions. During the last decades, many research groups investigated DNN-based signal processing for radar. State-of-the-art results have been reported for direction of arrival estimations [71], classifications of hand gestures [67, 44], and object detection [77], classification [78], and tracking [79]. Most of these solutions are based on RNNs or its variants, for the reasons mentioned in Sec. 1.1. In recent years, transformer networks are on the rise and show very good results [32, 80, 81]. The disadvantage of transformers is their size and thus computational complexity and energy requirement. For energy-critical task like-battery driven devices or electric vehicles, this disadvantage outweighs their advantages. Especially in the automotive sector, RNNs and LSTMs are often used for temporal tasks [82, 83, 84, 85, 86]. However, LSTMs have complex internal dynamics which cannot be accelerated very well [87]. SNNs on the other hand have simple internal dynamics and sparse event communication which enables low power consumption of up to 100-fold less energy consumption than DNNs [41]. A second motivation for SNN-based radar signal processing is the sparsity of radar data. A radar frame after range and doppler

processing as well as target detection (cf. Fig. 1.2) consists of targets and background noise. Instead of processing whole radar frames with DNNs, SNNs can benefit from efficient encodings that encode the important features in spike events while ignoring the noise. This selective encoding will reduce the number of spikes in the network which directly reduces the required energy budget.

Recent literature showed that DNNs solve the radar-based gesture recognition task with large precisions. Grobelny et al. [88] reported 94.3 % accuracy of LSTM-based networks on a radar gesture recognition task with twelve classes and Choi et al. [89] reported 98.48 % accuracy at a ten class problem. In recent years, SNNs have been demonstrated that closed the gap to DNNs in terms of accuracy. Many research groups demonstrated accuracies beyond 90 % for their own and publicly available datasets such as the Google Soli gesture recognition task with eleven fine grained gestures [67]. Arsalan et al. [90] showed 99.5 % for their own dataset with eight gestures, Safa et al. [91] report 93 % for their own eight class dataset and 93 % for the Google Soli dataset. Yin et al [92] and Tsang et al. [44] achieved 91.9 % and 98.02 %, respectively, on the Soli dataset. Both further compare these results against LSTMs and conclude that SNN-based architectures achieve competitive accuracies and have orders of magnitude less computational complexity, i.e. power consumption. However, no direct comparisons were based on one-to-one replacements of LSTMs with SNNs of the same size and structure. Additionally, energy comparisons were only based on estimating the computational cost by calculating and counting Floating Point Operations (FLOPs) and synaptic events. This estimation does indeed give an overview how complex a network is to calculate but does not consider how well a certain architecture can be accelerated on hardware. But this information is crucial because some architectures are much better to accelerate than others as shown by Jouppi et al. for feed-forward, convolutional, and LSTM networks [87]. Some works estimate and compare energy costs of LSTMs and SNNs by first counting FLOPs and synaptic events and then use measured energy requirements for both operations to give energy estimations, for example Safa et al. [91] based on measurements from the µBrain chip by Stuijt et al. [55]. Davies et al. [43] benchmarked different algorithms on their Loihi platform versus each algorithm's reference architecture. The authors showed that feed-forward networks have no benefits in terms of energy-delay ratio, meaning they are neither more energy efficient nor faster to execute on Loihi, whereas RNNs and other non-feed-forward networks perform much better on Loihi. Nevertheless, direct comparisons of LSTMs and SNNs based on leaky integrate-and-fire neurons that state which network is better suited for radar-based gesture recognition is missing.

For that reasons, the main focus of this thesis is to derive an efficient processing chain with spiking neural networks for radar-based gesture recognition which includes the following research questions:

1. What is the optimal neural network architecture for energy efficient radar processing chains?
2. Can neural networks outperform the traditional radar processing chain by substituting the FFT pre-processing?

In order to target those questions, the TensorFlow framework is used for network training and estimation of their computational complexity. Evaluation of the network's energy efficiencies is performed on the SpiNNaker 2 platform because it delivers a high flexibility that arises from fully simulated neurons and synapses on microprocessors with dedicated accelerators. This allows simple implementation of the different network architectures that are investigated throughout this thesis. Furthermore, the SpiNNaker 2 system allows to measure power and therefore energy consumption of networks for benchmarking network architectures. The gesture recognition task is an interesting application for this research. Radar-based gesture recognition is used for brain-machine interfaces which requires low latency algorithms, i.e. real-time capable, and low-power operation for battery-powered devices. It is further easy to collect large amounts of data for training purposes. Results from this task can then also be transfererd to other, more complex tasks with similar real-time and low-power requirements where data collection is not that easy such as autonomous driving.

This thesis is structured as follows: Ch. 2 introduces the fundamental backgrounds about artificial intelligence, deep learning, spiking neural networks, and the SpiNNaker 2 neuromorphic hardware platform. Ch. 3 first discusses requirements and methods to train SNN-based networks on radar-based gesture recognition tasks. It then explores potential entry points for SNN-based processing in the radar processing chain and finishes by explaining how energy measurements are performed on the SpiNNaker 2 system. Ch. 4 demonstrates the benefits of SNN-based radar data processing in terms of classification accuracy and computational complexity. In Ch. 5, networks are ported to the SpiNNaker 2 platform for energy and latency measurements and highlights the strength of SNNs in processing sparse radar data. Ch. 6 discusses the obtained results with respect to the literature. Finally, Ch. 7 concludes the contributions of this thesis.

1.3 Publications

- **P. Gerhards**, M. Weih, J. Huang, K. Knobloch, C. Mayr, "Hybrid Spiking and Artificial Neural Networks for Radar-Based Gesture Recognition", in *2023 8thInternational Conference of Frontiers of Signal Processing (ICFSP)*, Oct. 2023,
  DOI: 10.1109/ICFSP59764.2023.10372930

- **P. Gerhards**, F. Kreutz, K. Knobloch, C. Mayr, "Radar-Based Gesture Recognition with Spiking Neural Networks", in *2022 7th International Conference on Frontiers in Signal Processing (ICFSP)*, Sep. 2022,
  DOI: 10.1109/ICFSP55781.2022.9924676

- D. Scholz, F. Kreutz, **P. Gerhards**, J. Huang, F. Hauer, K. Knobloch, C. Mayr, "Augmenting Radar Data via Sampling from Learned Latent Space", in *2023 IEEE 3rd International Conference on Computer Communication and Artificial Intelligence (CCAI)*, May 2023,
  DOI: 10.1109/CCAI57533.2023.10201307

- J. Huang, F. Kelber, B. Vogginger, B. Wu, F. Kreutz, **P. Gerhards**, D. Scholz, K. Knobloch, C. Mayr, "Efficient Algorithms for Accelerating Spiking Neural Networks on MAC Array of SpiNNaker 2", in *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Jun. 2023,
  DOI: 10.1109/AICAS57966.2023.10168559
- J. Huang, F. Kelber, B. Vogginger, C. Liu, F. Kreutz, **P. Gerhards**, D. Scholz, K. Knobloch, C. Mayr, "Efficient SNN multi-cores MAC array acceleration on SpiNNaker 2", *Frontiers in Neuroscience*, vol. 17, Aug. 2023,
  DOI: 10.3389/fnins.2023.1223262
- J. Huang, B. Vogginger, **P. Gerhards**, F. Kreutz, F. Kelber, D. Scholz, K. Knobloch, C. Mayr, "Real-time Radar Gesture Classification with Spiking Neural Network on SpiNNaker 2 Prototype", in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Jun. 2022,
  DOI: 10.1109/AICAS54282.2022.9869987
- F. Kreutz, **P. Gerhards**, B. Vogginger, K. Knobloch, C. Mayr, "Applied Spiking Neural Networks for Radar-based Gesture Recognition", in *2021 7th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*, Jun. 2021,
  DOI: 10.1109/EBCCSP53293.2021.9502357