Journal of Progress in Electronics and Communication Engineering, ISSN: 3048-7625 Vol. 2, No. 1, 2025 (pp. 39-47)

RESEARCH ARTICLE WWW.PECEJOURNAL.COM

Detailed Guide to Machine Learning Techniques in Signal Processing

El Manaa Barhoumi[1], Y. Charabi[2], S. Farhani[3]

[1-3]College of Applied Science, University of Technology and Applied Sciences, Ibri, Sultanate of Oman

KEYWORDS:
Machine learning;
Signal processing;
Deep learning;
Feature extraction;
Pattern recognition;
Noise reduction;
Convolutional neural networks (CNNs)

ARTICLE HISTORY:
Received : 09.11.2024
Revised : 09.12.2024
Accepted : 13.03.2025

https://doi.org/10.31838/ECE/02.01.04

Abstract

This article provides an in-depth exploration of the intersection between machine learning and signal processing, two pivotal areas in modern technology. Signal processing involves the analysis, interpretation, and manipulation of signals, which can be in the form of audio, video, or sensor data. Machine learning, on the other hand, involves the development of algorithms that allow systems to learn from and make decisions based on data. This guide covers a broad spectrum of techniques and applications where these fields converge.The guide begins with foundational concepts, detailing the basics of signal processing and machine learning, and progresses to advanced topics such as deep learning, convolutional neural networks, and reinforcement learning. It emphasizes the application of machine learning algorithms to enhance signal processing tasks, including noise reduction, feature extraction, and pattern recognition. Practical implementations are discussed, highlighting real-world scenarios in areas such as telecommunications, healthcare, and multimedia. The guide addresses the challenges and future directions in this dynamic field, providing insights into the integration of emerging technologies like quantum computing and edge AI. By bridging the gap between theory and practice, this comprehensive guide serves as an essential resource for researchers, practitioners, and students aiming to harness the power of machine learning to advance signal processing technologies.

Author's e-mail: el.manaa.bar@gmail.com, charbi.y@gmail.com, farhani.s@gmail.com

How to cite this article: Barhoumi ME, Charabi Y, Farhani S. Detailed Guide to Machine Learning Techniques in Signal Processing, Journal of Progress in Electronics and Communication Engineering Vol. 2, No. 1, 2025 (pp. 39-47).

Introduction:

Signal processing has been an invaluable field for the past 50 years, providing tools like autocorrelation, convolution, Fourier and wavelet transforms, adaptive filtering, linear estimators, and compressed sensing to extract information from real-world signals.[1] However, as signals are often corrupted by noise or partially unavailable, machine learning techniques like deep neural networks have emerged as powerful alternatives for unveiling hidden structures and recovering desired information from complex data.[2]

Machine learning for signal processing leverages models like convolutional neural networks and transformers to process diverse signals, including electrical, mechanical, audio, speech, images, and biological data.[2] This comprehensive guide explores the growing intersection of machine learning and signal processing, covering deep learning workflows, long short-term memory models for human activity recognition, GPU and FPGA acceleration, and the rising importance of neural networks and deep learning in various signal processing applications [1], [2]. Signal processing is an electrical engineering subfield that focuses on analyzing, modifying, and synthesizing signals, such as sound, images, potential fields, seismic signals, altimetry processing, and scientific measurements.[3] A signal is a function x(t), where this function is either deterministic or a realization of a stochastic process $(X_t)_{t \in T}$.[3] Signal processing techniques are used to optimize transmissions, digital storage efficiency, correct distorted signals, enhance subjective video quality, and detect or pinpoint components of interest in a measured signal,[3] as mentioned Fig. 1.

Analog signal processing involves linear and nonlinear electronic circuits for signals that have not been digitized, as in most 20th-century radio, telephone, and television systems.[3] Digital signal processing is the processing of digitized discrete-time sampled signals by

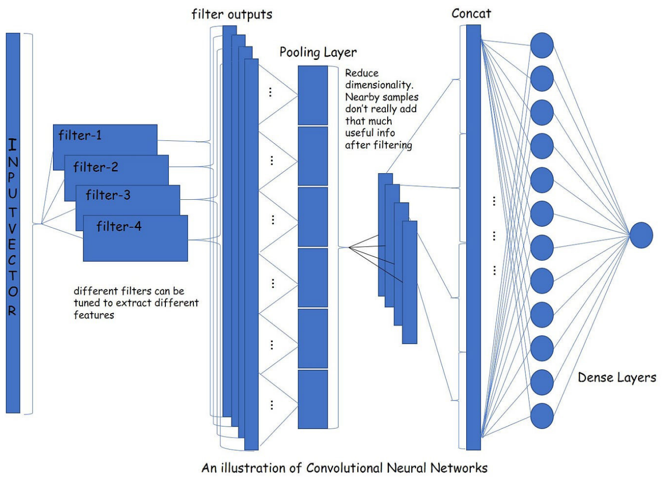

Fig. 1: Machine Learning and Signal Processing

general-purpose computers, digital circuits like ASICs, FPGAs, or specialized digital signal processors (DSP chips).[3]

Importance of Signal Processing

1. Signal processing allows engineers and scientists to analyze, optimize, and correct signals, including scientific data, audio streams, images, and video.[4-6]
2. Companies invest heavily in test equipment and manpower to conduct tests for understanding how their products will perform in the real world, which requires capturing and analyzing high-quality, objective data through signal processing.[4-6]
3. Signal processing transforms data in a way that allows us to see things that are not possible by direct observation or comparison.[6]

Applications of Signal Processing

Signal processing is applied across numerous industries and applications, including:[4-6]

1. Audio compression and signal processing
2. Data acquisition and signal processing
3. Digital image and graphics processing
4. Video compression and signal processing
5. Speech recognition and processing
6. RADAR, SONAR, and LiDAR signal processing and optimization
7. Seismic studies and data analysis
8. Geophysical applications, including oil exploration
9. Data transmission, including error detection and error correction
10. Economic modeling and analysis
11. Medical applications, especially imagery (CAT and MRI)
12. Weather forecasting
13. Oceanography, including undersea acoustic performance predictions

Some specific applications of signal processing include: [5]

- Speech recognition
- Hearing aids
- Autonomous driving
- Image processing and analysis
- Wearables
- Data science
- Communications systems and networks

Challenges of Signal Data

Deep learning for signal data presents unique challenges compared to applying deep learning techniques to other data types.[11] Signal data is often plagued by noise, variability, wideband noise, jitters, and distortions, making it difficult to obtain high-quality data.[11] These unwanted characteristics found in most signal data can hinder the performance of deep learning models.

Importance of Data Quantity and Computational Power

As with all deep learning projects, the success of applying deep learning to signal data heavily depends on the availability of large amounts of labeled data and substantial computational power.[12] Deep learning models require immense amounts of training data to achieve an acceptable level of accuracy, which was not easily accessible until the era of big data and cloud computing.[12]

Moreover, the hardware requirements for deep learning models are significant. Multicore high-performing graphics processing units (GPUs) and other similar processing units are necessary to ensure improved efficiency and decreased time consumption.[12] However, these units are expensive and consume large amounts of energy, in addition to requiring ample RAM and solid-state drives.[12]

The more powerful and accurate the deep learning model, the more parameters it requires, which in turn necessitates even larger datasets.[12] Therefore, a good deep learning workstation with substantial computational resources is highly recommended when applying deep learning techniques to signal data.[11]

While deep learning offers advantages like high recognition accuracy, crucial for applications where safety is a critical factor (e.g., autonomous vehicles, medical devices), its data and computational requirements can be limiting [12]. In cases where these requirements cannot be met, a thorough understanding of signal data and signal processing may be needed to employ machine learning techniques that rely on less data than deep learning [11].

Data Preprocessing and Feature Extraction

Preprocessing data is a crucial step in signal processing that lays the foundation for accurate and meaningful analysis. Depending on the data and the analysis, this may involve dealing with irregular or missing data through resampling and interpolation methods or smoothing the data using various filters,[15] as mentioned Fig. 2.

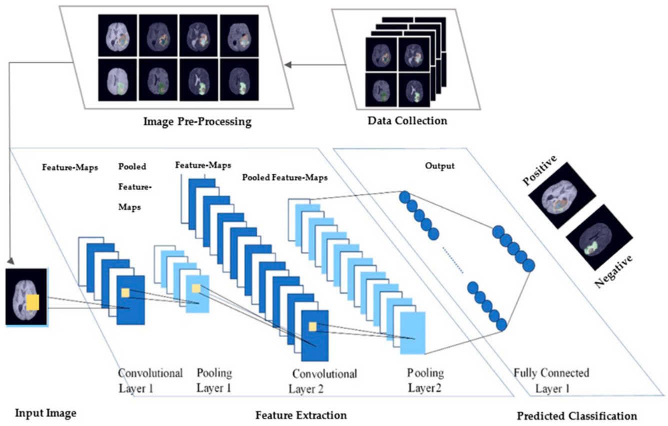

Fig. 2: Data Preprocessing and Feature Extraction

Noise and outliers are common occurrences in signal data, and they can make it challenging to obtain insights.[15] Outliers can be managed through techniques like data truncation or winsorization, which involve capping extreme values at a certain threshold.[15] One approach to handling noise is smoothing to reduce the impact of random fluctuations.[15] Moving averages and rolling windows are simple yet effective filtering techniques used to smooth time-series data and reduce the impact of noise.[15]

Signal Processing Toolbox™ provides functionality to perform signal labeling, feature engineering, and dataset generation for machine learning and deep learning workflows [16]. Tools like Signal Labeler allow users to label signal attributes, regions, and points of interest, and extract features for tasks like classification.[16]

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.[14] The extraction of features allows reducing the amount of data to be processed while transforming it into a smaller dataset with the same level of completeness and accuracy as the original data.[14] Good features depicting the most suitable representations of the data help in building effective machine learning models.[14]

Data Visualization

Visualizations are a powerful tool in signal processing, providing a clear understanding of the data patterns and the effectiveness of the applied techniques.[15] Time-domain plots, frequency spectra, and spectrograms are all common visualizations for signals.[15]

Classical Time Series Analysis Techniques

Several classical time series analysis techniques can be employed before applying deep learning models to signal data:

1. **Auto-correlation**: Auto-correlation measures the similarity between a time-series and a lagged version of itself, helping identify repeating patterns or cyclic behavior within the data.[15]
2. **Cross-correlation**: Cross-correlation explores the relationship between two different signals, useful in finding correlations and lagged associations between variables.[15]
3. **Trend Analysis**: Trend analysis is useful for understanding the underlying long-term behavior of a signal, representing the general direction in which the data is moving over an extended period.[15] Detrending methods are applied to separate the underlying trend from the signal, helping focus the analysis on the remaining components like seasonality and irregular fluctuations.[15]
4. **Fourier Transform**: The Fourier Transform is a mathematical technique used to convert a time-domain signal into its corresponding frequency-domain representation, decomposing the original signal into a sum of sinusoidal functions of different frequencies.[15]
5. **Power Spectral Density (PSD)**: The PSD plot displays the power (or magnitude squared) of each frequency component, with peaks indicating dominant frequencies in the data, which can reveal underlying patterns or periodic behavior.[15]
6. **Spectrogram**: The spectrogram is a valuable visualization technique used to examine how the frequency content of a signal changes over time,

providing a time-frequency representation of the data.[15]

A. Long Short-Term Memory Models (LSTMs) for Human Activity Recognition

Sensor-based and External Device-based HAR

Sensor-based human activity recognition (HAR) involves a series of steps: sensor selection, data collection, feature extraction, model training, and model testing [20]. Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Support Vector Machines (SVMs) are prominent techniques utilized for HAR [20].

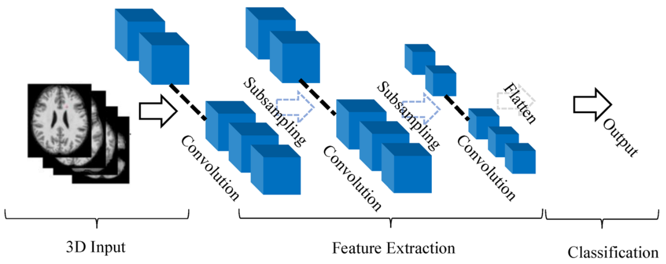

Fig. 3: 3D Deep Learning on Medical Image

Zhu et al. [17] proposed a semi-supervised deep learning approach that employs deep long-short term memory (DLSTM) on labeled and unlabeled data. Smartphone inertial sensors were used to collect data, and deep neural networks (DNNs) were utilized to obtain local dependencies from the data characteristics,[17] as mentioned Fig. 3.

Pan et al.[28] aimed to address the issues of independent and identically distributed (I.I.D.) data by employing the Gated Recurrent Unit (GRU) network, which collects valuable moments and temporal attention to minimize model attributes for HAR.[28]

Luwe et al.[29] proposed a hybrid model combining one-dimensional CNN with bidirectional LSTM (1D-CNN-BiLSTM) to recognize individual actions using wearable sensors. The 1D-CNN converts visible features gathered by sensors to indicative features, while BiLSTM encodes broad dependencies through a gating process.[29]

Deep RNN Architectures for HAR

Deep learning (DL) methods like Convolutional Neural Networks (CNNs), Deep Belief Networks (DBNs), and Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM), have been widely used for activity recognition based on time series data.[19] In DL, a multilayered architecture (deep) is constructed for various objects, including feature selection and classification, with each layer performing a non-linear conversion on the previous layer's output.[19]

CNNs are particularly potent for recognizing activities based on time series data.[19]. Researchers have focused on DL and fuzzy computing to address challenges associated with windowing, feature selection, and recognition, with LSTM networks playing a significant role in feature selection and activity recognition by incorporating previous results into current decisions.[19]

End-to-End Mapping from Raw Sensor Data to Activity Labels

Researchers have proposed end-to-end models that map raw sensor data directly to activity labels, overcoming the need for explicit feature extraction.[23-24] These models often involve convolutional-recurrent neural network architectures with LSTM layers for real-time processing and classification of digital sensor data.[23-24]

The convolutional-recurrent architecture offers practical advantages, such as flexible sequence lengths for training or prediction and the possibility of real-time implementation on various platforms.[23] By training a neural network to perform all stages of feature extraction and classification, performance far superior to what is possible from heart rate features alone can be achieved.[23]

LSTM networks have been particularly effective in capturing long-term temporal dynamics in the context of activity recognition.[21, 22] They have been used to capture performed motion information and its dependencies from recorded sensor data.[21, 22] LSTMs can also be employed in the context of skeleton-based activity analysis by considering the spatial and temporal motion information of human bone points as features.[21, 22]

While traditional Recurrent Neural Networks (RNNs) struggle with long-term dependencies found in human activity data, Hochreiter and Schmidhuber[16] proposed LSTMs by providing a gating mechanism to hold long-term memory and avoid the long-dependency problem.[16, 21, 22]

B. Signal Processing on GPUs

Performance Benefits of GPUs

Graphics Processing Units (GPUs) have emerged as powerful accelerators for signal processing tasks due to their ability to perform parallel computations efficiently .[27, 28] GPUs offer several performance advantages over traditional CPUs:

1. **Speed and Parallel Processing:** GPUs leverage their parallel architecture and numerous cores to execute the computationally intensive matrix operations and complex calculations involved in signal processing algorithms significantly faster than CPUs.[26] This accelerates training and inference processes, making signal processing tasks more efficient.

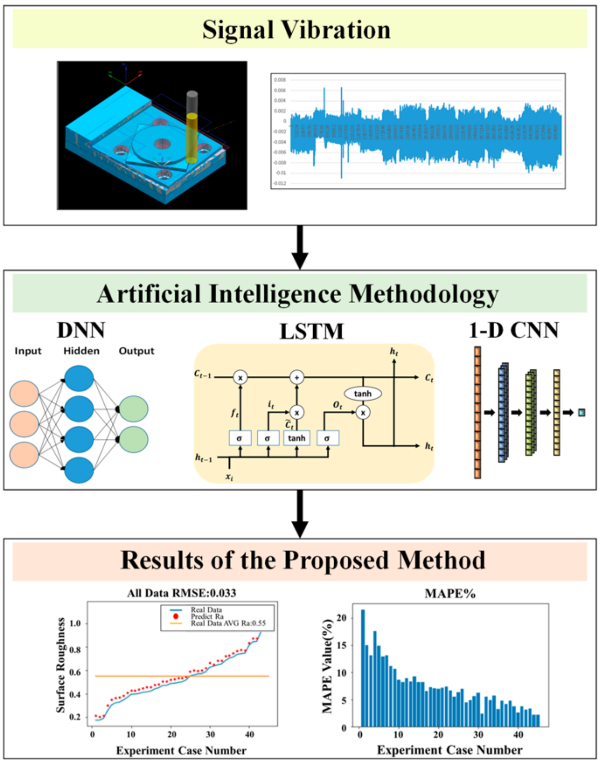

Fig. 4: **Evaluation of Deep Learning Neural Networks**

2. **Optimized Libraries and APIs**: GPU manufacturers provide specialized libraries and APIs, such as CUDA for NVIDIA GPUs or ROCm for AMD GPUs, designed to accelerate signal processing frameworks [26]. These tools optimize algorithms and provide functionalities tailored for signal processing tasks, enhancing both performance and efficiency, as mentioned Fig. 4.
3. **Dedicated Hardware for Signal Processing Workloads:** Modern GPUs incorporate dedicated hardware components, like tensor cores, specifically designed to accelerate signal processing workloads.[26] These tensor cores perform matrix multiplication operations used heavily in signal processing with significantly higher throughput, further enhancing performance while maintaining efficiency.
4. **Reduced Processing Times:** The use of hardware-accelerated signal processing on GPUs significantly reduces the time required for processing complex signals [26]. Tasks that might take weeks on a CPU can often be accomplished in hours or even minutes on a GPU, improving productivity and enabling faster experimentation and model iteration.
5. **Energy Efficiency:** Although GPUs consume more power compared to CPUs, they offer higher performance per watt in signal processing tasks due to their ability to handle parallel computations efficiently.[26] This means that while they consume more power overall, they can complete tasks faster, potentially leading to better energy efficiency in terms of computations per unit of power consumed.

C. GPU-accelerated Libraries for Signal Processing

To leverage the computational power of GPUs for signal processing, various GPU-accelerated libraries have been developed:

1. **cuFFT:** NVIDIA's cuFFT library provides GPU acceleration for Fast Fourier Transform (FFT) computations, which are essential for many signal processing tasks.[28] It outperforms CPU-based FFT libraries, with benchmarks showing a 3x performance improvement over CPU-based libraries like Intel Math Kernel Library (MKL).[26]
2. **cuBLAS, cuSOLVER, and cuSPARSE:** These NVIDIA libraries accelerate matrix solvers, decompositions, and sparse matrix operations, which are crucial for various signal processing algorithms.[28]
3. **cuSignal:** NVIDIA's cuSignal library offers GPU-accelerated implementations of common signal processing operations, such as filtering, windowing, and modulation.[26] It utilizes techniques like zero-copy memory mapping to minimize data transfer overhead between CPU and GPU, improving overall workflow performance.
4. **GPUFFTW:** The GPUFFTW library claims to be the fastest FFT library on GPUs, running on NVIDIA cards [26]. It reportedly achieves a 3x performance improvement over other GPU-based FFT libraries like libgpufft and an empirical computational performance of 29 GFLOPS on an NVIDIA 8800 GTX GPU.[26]
5. **Hardware Implementations:** In addition to software libraries, hardware implementations of signal processing algorithms like FFT can be achieved using Field-Programmable Gate Arrays (FPGAs), Digital Signal Processors (DSPs),

and custom integrated circuits (ICs).[26] These hardware implementations offer ultra-low latency and high accuracy, making them suitable for real-time signal processing applications.

These GPU-accelerated libraries and hardware implementations enable signal processing engineers to leverage the computational power of GPUs, leading to significant performance improvements and reduced processing times compared to traditional CPU-based approaches.[27, 28]

Signal Processing on FPGAs

Field-Programmable Gate Arrays (FPGAs) are integrated circuits designed with configurable logic blocks and interconnects that can be programmed to implement custom hardware functionality.[29] Unlike CPUs and GPUs, which are software-programmable fixed architectures, FPGAs are reconfigurable, and their compute engines are defined by the user.[31] When writing software targeting an FPGA, compiled instructions become hardware components that are laid out on the FPGA fabric in space, and those components can all execute in parallel.[31 Because of this, FPGA architecture is sometimes referred to as a spatial architecture.

An FPGA is a massive array of small processing units consisting of up to millions of programmable 1-bit Adaptive Logic Modules (each can function like a one-bit ALU), up to tens of thousands of configurable memory blocks, and tens of thousands of math engines, known as digital signal processing (DSP) blocks, that support variable precision floating-point and fixed-point operations.[31 All these resources are connected by a mesh of programmable wires that can be activated on an as-needed basis as mentioned Fig. 5.

When software is "executed" on the FPGA, it is not executing in the same sense that compiled and assembled instructions execute on CPUs and GPUs.[31 Instead, data flows through customized deep pipelines on the FPGA that match the operations expressed in the software.[31 Because the dataflow pipeline hardware matches the software, control overhead is eliminated, which results in improved performance and efficiency.[31

Advantages of FPGAs for Signal Processing

FPGAs offer several advantages for signal processing applications:

1. **Efficiency**: Data processing pipelines can be tuned exactly to the needs of the software, eliminating the need for control units, instruction fetch units, register writeback, and other execution overhead.[31
2. **Custom Instructions**: Instructions not natively supported by CPUs/GPUs can be easily implemented and efficiently executed on FPGAs (e.g., bit manipulations).[31
3. **Data Dependencies across Parallel Work**: Data dependencies across parallel work can be resolved without stalls to the pipeline.[31

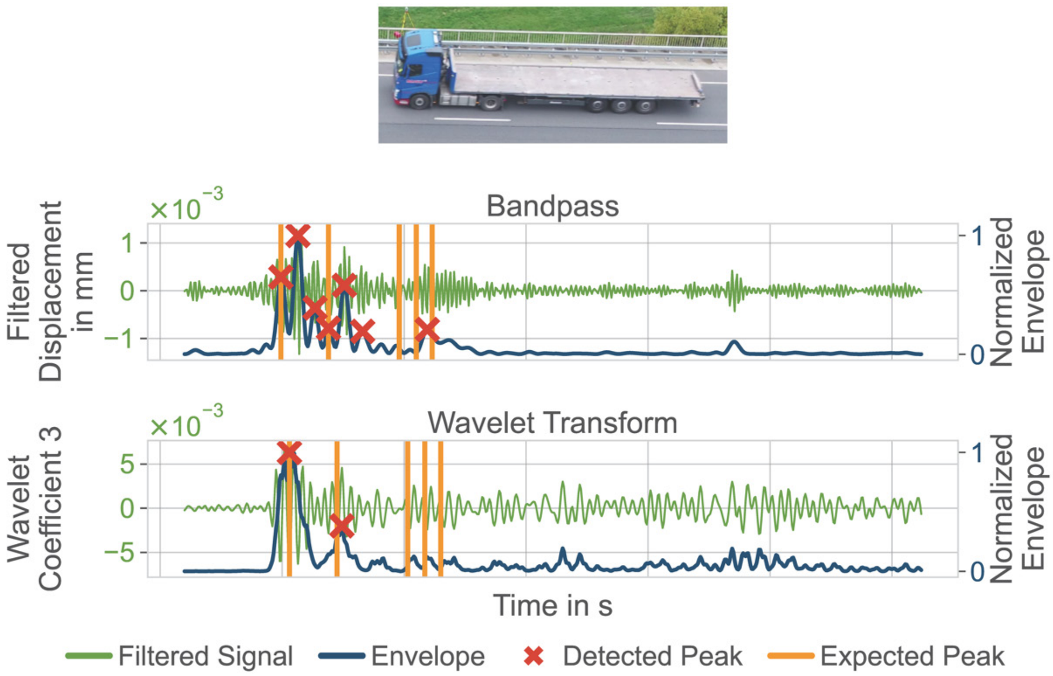

Fig. 5: Machine Learning and Signal Processing for Bridge Traffic Classification

4. **Flexibility:** FPGAs can be reconfigured to accommodate different functions and data types, including non-standard data types .[31].
5. **Custom On-Chip Memory Topology**: On-chip memory can be built to accommodate the access pattern of the algorithm, minimizing or eliminating stalls.[31
6. **Rich I/O**: The FPGA core can interact directly with various network, memory, and custom interfaces and protocols, resulting in low and deterministic latency solutions.[31

FPGAs are well-suited for applications that require low-latency and real-time signal processing, such as digital signal processing, radar systems, software-defined radios, and telecommunications.[32] Their programmability empowers developers to adapt the hardware to meet the specific requirements of their applications.[32]

The ability to change the internal circuitry of FPGAs makes them an excellent choice for prototyping and development, as engineers can iterate quickly, testing different hardware configurations until they find the most efficient solution for their problem.[32] FPGAs often outshine GPUs in terms of latency and power usage, especially when fine-tuned for certain tasks, as developers can implement custom hardware accelerators tailored to specific tasks that may not be well-suited for the fixed architectures of GPUs.[32]

The Growing Importance of Signal Processing

The notion of "signal processing" might seem impenetrably complex, even to scientists. However, the fact is that most of them have already been doing it for a long time, albeit in an unconscious way.[33] Acquiring, shaping and transforming data, cleaning it for improved analysis and extraction of useful information – all of this is what experimental science is about.[33]

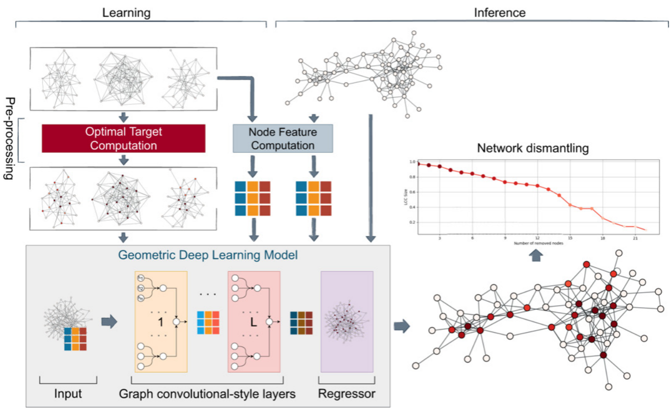

Fig. 6: Machine learning dismantling and early-warning signals of disintegration in complex systems

A signal is essentially the physical support of information: audio recordings, images, videos, data collected by sensors of different nature.[33] It includes almost all forms of data that are or can be digitized, including texts.[33] Our world is full of signals that can be processed, coming from nature as well as created by past and current technology,[33] as mentioned Fig. 6.

Signal Processing as an Interface Discipline

Mathematics is a necessary transit point for the formalization of processing and the evaluation of its performance.[33] Informatics offers possibilities of efficient implementation of algorithms.[33] Interacting with each of these fields, yet irreducible to any of them, signal processing is by nature an interface discipline.[33] Developing its own methodologies in a constant dialog with applications, from acquisition to interpretation, it stands as both singular and central in the landscape of information sciences.[33]

Origins and Early Development

Tracing back to the origins of signal processing, one finds Joseph Fourier (1768-1830), one of the field's pioneers.[33] To establish the equations governing heat propagation, Fourier developed a mathematical method – now called the "Fourier transform" – that replaces the description of a signal in the time or space domain by another one in the frequency domain.[33]

Even before World War II, signal processing began to develop, mostly for military purposes related to sonar and radar.[33] This led to new approaches aimed at solidifying the theoretical bases of then-vague notions such as message, signal, noise, transmission and control.[33] Within a few years emerged the communication theory of Dennis Gabor, the cybernetics of Norbert Wiener and, of course, the information theory of Claude E. Shannon.[33]

In parallel, the French physicist and mathematician André Blanc-Lapierre developed a "theory of random functions" that allowed the modeling and analysis of the "background noise" observed in underwater acoustics.[33] This collaboration between academia and Navy gave birth to the GRETSI association, which, in 1967, organized the first-ever congress on the emerging field of signal processing.[33]

Widespread Applications

Signal processing is concerned today by a variety of applications that go far beyond its origins, investing progressively more and more domains of science and technology [33]. As a recognition of the importance of signal processing, the IEEE Signal Processing Society doesn't hesitate to define its domain as "the science behind our digital life".[33]

The health field offers many examples of applications of signal processing, handling waveforms, images or video sequences under different modalities in problems as diverse as tracking of heartbeats, localization of epileptic sources in the brain, echography, or magnetic-resonance imaging (MRI).[33] The same applies to major areas such as energy or transportation, where the ever-growing deployment of sensors and the need for exploiting the collected information puts signal processing at the center of global issues such as smart grids and smart cities.[33]

A common denominator of many signal processing methods is their interest in "denoising" data, "disentangling" and "reconstructing" them while taking into account the limited resolution of sensing devices.[33] This concern can also be found in other domains such as seismics, with the purpose of imaging the underground thanks to returning echoes from emitted vibrations.[33]

In astronomy and astrophysics, two recent examples in which signal processing proved instrumental are the European Space Agency's Planck mission, where advanced source-separation techniques permitted the reconstruction of the "oldest picture" of the Universe, and the first direct detections on Earth of gravitational waves by the LIGO-Virgo collaborationsv These extraordinary achievements were made possible by advanced signal-processing algorithms that were based in part on wavelet analysis, which can be seen as a powerful extension of Fourier's work.[33]

Emerging Challenges and Future Directions

The digital world in which we live is an ever-growing source of data: hyperspectral imaging with hundreds of frequency channels, networks with thousands of sensors (for example, in environmental science), contact data on social networks.[33] These new configurations can lead signal processing to re-invent itself, in connection with techniques from distributed computing, optimization or machine learning.[33]

Yet signal processing must keep its identity and specificities, and guarantee the development of methods that are both generalizable and computationally efficient [33]. It must also be based on an in-depth understanding of data and be fundamentally aware of its potential impact on our society.[33]

Enabling Technology for Modern Devices and Systems

Signal processing – the enabling technology for the generation, transformation, extraction and interpretation of information via electronic signals – is essential for our smartphones and wearable devices, as well as the latest healthcare technologies, digital cameras and our digital assistants like Amazon Echo and Google Home.[33]

Taking stock of the immense power and promise of signal processing, it's not difficult to see how it can be the ideal vocational path for a person with an interest in science, technology, or math, and a desire to change the world.[33] Signal processing allows for the expansion of computing power and data storage capabilities, making signal processing engineers indispensable for understanding and tackling our biggest global problems.[33] A career in this field isn't just about employment opportunities or guarding against your job being automated. It's about contributing to improving the world.[33]

Conclusion

The intersection of machine learning and signal processing presents incredible opportunities to extract valuable insights from complex and noisy data. Advanced techniques like convolutional neural networks, LSTMs, and GPU/FPGA acceleration are revolutionizing signal analysis across various domains. As data continues to grow exponentially, the ability to leverage these powerful tools will be critical for scientific breakthroughs and technological advancements.Signal processing has emerged as a central discipline, enabling groundbreaking applications in fields ranging from healthcare and energy to astronomy and seismics. Its methodologies, rooted in mathematics and computer science, continue to evolve, incorporating emerging technologies like distributed computing and machine learning. With its focus on efficient data processing and interpretation, signal processing will undoubtedly play a pivotal role in shaping our digital future and tackling global challenges.

References:

1. Bishop, C. M. (2006).Pattern Recognition and Machine Learning. Springer.
2. Bousseljot, R., Kreiseler, D., & Schnabel, A. (1995). Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet.Biomedizinische Technik/Biomedical Engineering, 40(s1), 317-318.
3. Bruna, J., &Mallat, S. (2013). Invariant scattering convolution networks.IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1872-1886.

4. Cichocki, A., & Amari, S. (2002).Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. Wiley.
5. Coates, A., Ng, A. Y., & Wang, A. (2011). Large-scale hierarchical convolutional deep learning. InProceedings of the 30th International Conference on Machine Learning (ICML-13)(pp. 1387-1395).
6. Duda, R. O., Hart, P. E., & Stork, D. G. (2000).Pattern Classification(2nd ed.). Wiley.
7. Goodfellow, I., Bengio, Y., & Courville, A. (2016).Deep Learning. MIT Press.
8. Haykin, S. (1994).Neural Networks: A Comprehensive Foundation. Prentice Hall.
9. Hyvarinen, A., Karhunen, J., & Oja, E. (2001).Independent Component Analysis. Wiley.
10. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects.Science, 349(6245), 255-260.
11. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
12. Mallat, S. (2008).A Wavelet Tour of Signal Processing: The Sparse Way. Academic Press.
13. Reed, R. D., & Marks, R. J. (1999).Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press.
14. Ripley, B. D. (1996).Pattern Recognition and Neural Networks. Cambridge University Press.
15. Schmidhuber, J. (2015). Deep learning in neural networks: An overview.Neural Networks, 61, 85-117.
16. Schölkopf, B., &Smola, A. J. (2002).Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.
17. Smola, A. J., &Vishwanathan, S. V. N. (2008).Introduction to Machine Learning. Cambridge University Press.
18. Rani, B.M.S., et al., “Road Identification Through Efficient Edge Segmentation Based on Morphological Operations,” Traitement du Signal, 38(5), 2021.
19. Vijay, V. and Srinivasulu, A., “A novel square wave generator using second-generation differential current conveyor,” Arabian Journal for Science and Engineering, 42(12), 2017, pp.4983-4990.
20. Vapnik, V. N. (1998).Statistical Learning Theory. Wiley.
21. Widrow, B., & Lehr, M. A. (1990). 30 years of adaptive neural networks: Perceptron, Madaline, and backpropagation. Proceedings of the IEEE, 78(9), 1415-1442.
22. Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H.,... & Steinberg, D. (2008). Top 10 algorithms in data mining.Knowledge and Information Systems, 14(1), 1-37.
23. Zhang, Z. (2000). A flexible new technique for camera calibration.IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), 1330-1334.
24. Zhou, S. K., Chellappa, R., & Moghaddam, B. (2006).Visual Tracking and Recognition Using Appearance-Adaptive Models. Springer.
25. Pittala, C.S., et al., “1-Bit FinFET carry cells for low voltage high-speed digital signal processing applications,” Silicon, 15(2), 2023, pp.713-724.
26. Nizam, Taaha, et al. “Novel all-pass section for high-performance signal processing using CMOS DCCII.” TENCON 2021-2021 IEEE Region 10 Conference (TENCON). IEEE, 2021.
27. Babu, D. Vijendra, et al. “Digital code modulation-based MIMO system for underwater localization and navigation using MAP algorithm.” Soft Computing (2023): 1-9.
28. Selvam, L., et al. “Collaborative autonomous system based wireless security in signal processing using deep learning techniques.” Optik 272 (2023): 170313.
29. Rani, B. M. S., et al. “Disease prediction based retinal segmentation using bi-directional ConvLSTMU-Net.” Journal of Ambient Intelligence and Humanized Computing (2021): 1-10.
30. McFee, B., Salamon, J., Li, P., & Ellis, D. P. (2015).Research Advances in Machine Learning for Signal Processing. IEEE Signal Processing Magazine, 32(4), 128-135.
31. Mitchell, T. M. (1997).Machine Learning. McGraw-Hill.
32. O’Shea, T. J., &Hoydis, J. (2017). An introduction to deep learning for the physical layer.IEEE Transactions on Cognitive Communications and Networking, 3(4), 563-575.
33. Rabiner, L. R., & Schafer, R. W. (2010).Theory and Applications of Digital Speech Processing. Prentice Hall.