Karafan
Quarterly Scientific Journal of National University of Skills
Spring 2025, Vol. 22, No. 1, p.105-126
Journal Homepage: https://karafan.nus.ac.ir/article_229121.html
https://doi.org/10.48301/kssa.2025.481769.3025

Presenting a method to reduce crosstalk in 3D on-chip networks using convolutional neural networks

Ali Fard[1], Zahra Shirmohammadi[2*]

[1] Bsc Student in Computer Engineering, Department of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran.

[2] *Associate Professor, Department of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran.

ARTICLE INFO

Received: 28.10.2024
Revised: 15.03.2025
Accepted: 25.06.2025

Keyword:
Crosstalk,
Communication Networks On A Chip,
Multi-core Systems,
Coding,
Deep Learning,
Artificial Intelligence

***Corresponding Author:**
Zahra Shirmohammadi
Email:
shirmohammadi@sru.ac.ir

ABSTRACT

With the advancement of technology and the increasing density of transistors in chips, crosstalk has become one of the main challenges in Network-on-Chip. This phenomenon causes unwanted interference between adjacent wires, leading to issues such as increased latency, reduced reliability, and higher energy consumption. Traditional methods for mitigating crosstalk, such as coding techniques and the use of repeaters, face limitations including increased overhead, high complexity, and higher energy consumption. Therefore, coding methods are considered effective approaches. The goal of this paper is to propose a novel encoding method based on deep learning and convolutional neural networks to reduce the effects of crosstalk in Network-on-Chip. In this method, the proposed algorithm, by analyzing and learning harmful and incompatible patterns in data, can accurately identify and eliminate these patterns. Simulation results demonstrate that the proposed method not only improves the performance of processing circuits but can also serve as an effective solution for designing the next generation of multi-core chips. Following a review of related work and the research background, the proposed method is thoroughly explained, and simulation results are presented. Finally, the paper concludes with a summary and suggestions for future research.

©2024 the authors. Published by National University of Skills, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-Noncommercial 4.0 International (CC BY-NC License) (https://creativecommons.org/licenses/by-nc/4.0/)

E-ISSN: 2538-4430
ISSN: 2382-9796

EXTENDED ABSTRACT

Introduction

In the increasingly complex field of multicore systems, designing and implementing advanced processing circuits comes with numerous challenges, one of the most critical being crosstalk. Crosstalk, which is electromagnetic interference between parallel signal paths, has been a major issue in modern integrated circuits (ICs), especially in 3D on-chip networks where signal lines are closely packed. Traditional methods of mitigating crosstalk, such as coding techniques and the use of transistor repeaters, are associated with significant overhead, increased complexity, and higher power consumption. These limitations have prompted the exploration of more innovative approaches, particularly those incorporating artificial intelligence (AI).

This paper presents a novel approach to address crosstalk using convolutional neural networks (CNNs). By leveraging the pattern recognition capabilities of CNNs, the proposed method aims to detect and mitigate the effects of crosstalk, thereby improving performance in terms of processing speed and power consumption. The CNN-based approach can dynamically analyze signal patterns, identify interference, and adaptively filter out harmful patterns, providing a more efficient solution for multicore system design. This article contributes to the growing body of knowledge on the application of AI in hardware optimization, offering a promising solution to one of the most persistent problems in modern chip design.

Methodology

The methodology employed in this study revolves around the use of CNNs to analyze signal patterns within a 3D on-chip network and to detect crosstalk. CNNs, a type of deep learning architecture, are particularly well-suited for tasks that involve recognizing patterns in grid-like structures, making them ideal for processing the signal waveforms commonly found in chip interconnLects.

The first step in the methodology involves collecting data from simulated multicore systems, where signals affected by crosstalk and unaffected signals are generated. This data is pre-processed and labeled, with each signal categorized as either "clean" or "affected by crosstalk." The dataset is then split into training, validation, and test sets to train the CNN model.

The CNN is designed with several layers: convolutional layers to extract features, pooling layers to reduce the dimensionality of the data, and fully connected layers that output a classification indicating whether crosstalk is present. The training process involves optimizing the network by adjusting hyperparameters, such as the number of layers, filter sizes, and the learning rate. The aim is to maximize the CNN's ability to classify the signals while minimizing computational overhead correctly.

After training, the CNN model is tested against the unseen test data to assess its performance. Metrics such as accuracy, power consumption, and processing speed are used to compare the effectiveness of the CNN-based method with traditional techniques, including coding methods and transistor repeaters. The results from these tests are then used to fine-tune the model, ensuring optimal performance in real-world applications.

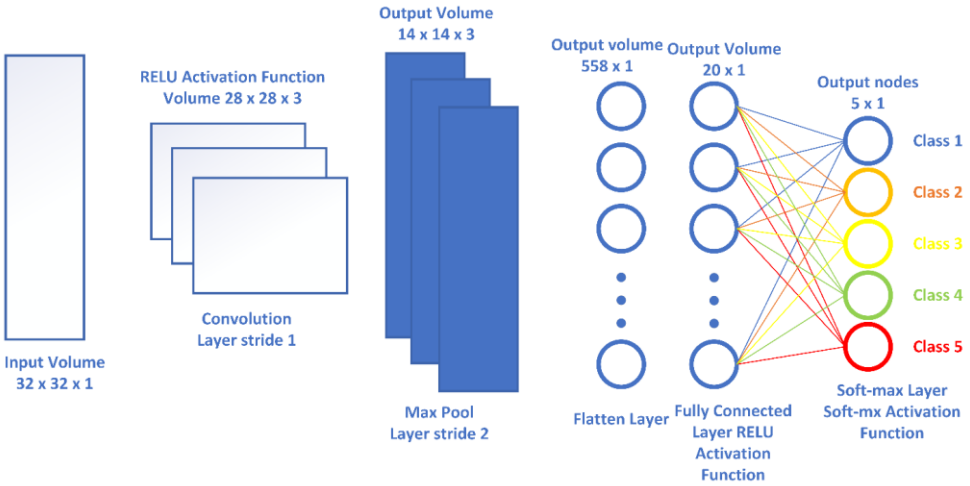


Figure 1. Layers of Convolutional Neural Networks.

Results and Discussion

The research presented in this article provides a novel approach to mitigating crosstalk in 3D on-chip networks by leveraging convolutional neural networks (CNNs). The proposed method demonstrates the potential for improving signal integrity by dynamically analyzing and filtering harmful patterns that cause interference, ultimately enhancing the performance of multicore systems.

Compared to traditional methods such as physical repeaters and coding techniques, this CNN-based approach offers several advantages. It operates without the need for additional hardware components, which can reduce the power consumption and overall complexity of the system. Moreover, the scalability of CNNs allows this approach to adapt to increasingly complex multicore environments, making it well-suited for future generations of chips.

The study emphasizes the importance of integrating AI-driven techniques in hardware design, providing a foundation for further exploration in this field. Future research could focus on optimizing the CNN architecture for real-time applications and investigating its applicability to other forms of signal interference beyond crosstalk. The adaptability of CNNs holds great promise for transforming chip design, leading to more efficient and reliable systems in the years to come.

The discussion also touches on potential applications of the CNN-based method beyond crosstalk mitigation. The underlying principles of this approach—dynamic pattern recognition and adaptive filtering—could be applied to other types of signal interference, such as electromagnetic interference (EMI) or noise in wireless communication systems. Furthermore, the adaptability of CNNs means they could be used in dynamic environments where signal conditions fluctuate, offering a robust solution for real-time hardware optimization.

Conclusions

The proposed method presented in this paper offers a promising new approach to mitigating crosstalk in 3D on-chip networks using convolutional neural networks. By leveraging the powerful pattern recognition capabilities of CNNs, the proposed method can accurately detect and filter out crosstalk, leading to improved signal integrity, reduced power consumption, and enhanced scalability.

Compared to traditional methods, the CNN-based approach not only increases accuracy but also

offers a more energy-efficient solution that does not require additional hardware components. This makes it particularly well-suited for the next generation of multicore systems, where power efficiency and scalability are paramount. Furthermore, the adaptability of the CNN allows it to handle increasingly complex systems, providing a versatile tool for future chip designs.

This article contributes to the growing field of AI-driven hardware optimization and paves the way for further exploration into the use of machine learning in chip design. Future work could focus on refining the CNN model to improve its performance further, exploring its applicability to other forms of signal interference, and integrating it into real-time systems for dynamic, on-the-fly optimization. As AI continues to evolve, its potential to transform hardware design will only grow, making it an essential tool for engineers and designers.

کارافن
فصلنامه علمی دانشگاه ملی مهارت
بهار۱٤۰٤، دوره ۲۲، شماره ۱، ۱۲٦-۱۰۵
آدرس نشریه: https://karafan.nus.ac.ir/article_229121.html
https://doi.org/10.48301/kssa.2025.481769.3025

ارائه روشی برای کاهش اشکال همشنوایی در شبکه‌های روی تراشه سه-بعدی با استفاده از شبکه‌های عصبی پیچشی

علی فرد[۱] زهرا شیرمحمدی*[۲]

۱- دانشجوی کارشناسی مهندسی کامپیوتر، گروه مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران.[۱]
۲*- دانشیار ، گروه مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران.[۲]

اطلاعات مقاله

دریافت مقاله: ۱۴۰۳/۰۸/۰۷
بازنگری مقاله : ۱۴۰۳/۱۲/۲۵
پذیرش مقاله: ۱۴۰۴/۰۲/۰۶

کلید واژگان:
همشنوایی،
شبکه‌های ارتباطی روی تراشه،
سیستم‌های چند هسته‌ای،
کدینگ،
یادگیری عمیق،
هوش مصنوعی

***نویسنده مسئول:** زهرا شیرمحمدی
پست الکترونیکی:
shirmohammadi@sru.ac.ir

چکیده

با پیشرفت فناوری و افزایش تراکم ترانزیستورها در تراشه‌ها، همشنوایی به یکی از چالش‌های اصلی در شبکه‌های روی تراشه تبدیل شده است. این پدیده باعث ایجاد تداخل ناخواسته بین سیم‌های مجاور می‌شود و مشکلاتی مانند افزایش تأخیر، کاهش قابلیت اطمینان و مصرف انرژی بیشتر را به همراه دارد. روش‌های سنتی مقابله با همشنوایی، مانند تکنیک‌های کدینگ و استفاده از تکرارکننده‌ها، با محدودیت‌هایی نظیر سربار افزایشی، پیچیدگی بالا و مصرف انرژی بیشتر مواجه هستند. لذا روش‌های کدینگ روش های کدینگ روش‌های موثری هستند. هدف این مقاله ارائه یک روش کدگذاری نوین مبتنی بر یادگیری عمیق و شبکه‌های عصبی پیچشی برای کاهش اثرات همشنوایی در شبکه‌های روی تراشه است. در این روش، الگوریتم پیشنهادی با تحلیل و یادگیری الگوهای ناسازگار و مضر در داده‌ها، قادر است این الگوها را با دقت بالا شناسایی و حذف کند. نتایج شبیه‌سازی‌ها نشان می‌دهد که روش پیشنهادی نه تنها عملکرد مدارهای پردازشی را بهبود می‌بخشد، بلکه می‌تواند به عنوان یک راهکار مؤثر در طراحی نسل جدید تراشه‌های چند هسته‌ای مورد استفاده قرار گیرد. در ادامه، پس از بیان پیشینه تحقیق و مرور کارهای مرتبط، روش پیشنهادی به‌طور دقیق تشریح شده و نتایج شبیه‌سازی‌ها ارائه می‌شود. در نهایت، مقاله با جمع‌بندی و پیشنهاداتی برای تحقیقات آینده به پایان می‌رسد.

شاپای الکترونیکی: 2538-4430
شاپای چاپی: 2382-9796

©2024 the authors. Published by National University of Skills, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-Noncommercial 4.0 International (CC BY-NC License) (https://creativecommons.org/licenses/by-nc/4.0/)

۱. مقدمه

کوچک‌تر شدن فناوری ساخت مدار، امکان تعبیه تعداد بسیار زیادی ترانزیستور در یک مدار را ممکن می-سازد و بدین صورت کارایی مدارات با استفاده چندین هسته پردازشی در یک تراشه فراهم شده است.[۱] طراحی پیمانه‌ای باعث شده تا پیچیدگی سیستم‌های چند پردازشی کاهش پیدا کند. مهم‌ترین چالش در تراشه‌های چندهسته‌ای متصل‌کردن هسته‌ها به یکدیگر است که برای حل این مشکل شبکه‌های روی تراشه به-عنوان یک راه‌حل مفید و قابل‌اجرا ارائه شده‌اند.[۱] اما این فناوری نیز منجر به ایجاد مشکلاتی از قبیل به‌خطرافتادن قابلیت سیستم‌های مبتنی بر تراشه شده است.[۲]

مشکل همشنوایی یکی از مهم-ترین چالش-هایی است که باعث شده سیستم-های مبتی بر شبکه-های روی تراشه کمتر قابل اطمینان باشد. علت این مسئله ایجاد خازن‌های تزویج و تزویج سلفی بین سیم‌های گذرگاه می‌باشد. این تاثیرات که شامل ایجاد ولتاژ گذرناخواسته، تسریع و یا تاخیر در سیم-های قربانی می‌شود؛ روی داده‌های در حال انتقال سیم‌های ارتباطی کانال رخ می‌دهد.اهمیت رسیدگی به این موضوع زمانی بیشتر حس خواهد شد که با توجه به اینکه فناوری به سمت طراحی عمیق زیر میکرون حرکت می‌کند، با افزایش طول مجموع سیم-های گذرگاه ارتباطی میزان اشکال همشنوایی نیز بیشتر خواهد شد.[۲] شدت اشکال همشنوایی به داده-های متوالی که برروی خطوط عمودی ظاهر می‌شود، بستگی دارد. این داده‌ها بنا به شدت خط به کلاس-هایی به نام-های کلاس ۰ تا ۸ تقسیم می شوند.[۳]

برای حل مشکل همشنوایی، در روش های سنتی مشکل در سه سطح فیزیکی، ترانزیستوری و کدگذاری مورد بررسی و مقابله قرار می‌گیرد. روش هایی که در سطح فیزیکی و ترانزیستوری اجرا می‌شوند؛ علی-رغم مفید بودن بدلیل تحمیل سربارهایی همچون مساحت زیاد و افزونگی سخت‌افزاری به مدار، نمی-توان از آنها استفاده کرد. لذا روش‌های در سطح کدگذاری داده‌ها، روش‌هایی هستند که اغلب برای حذف کلاس-های اشکال هم-شنوایی به-کار می‌رود. اما این روش ها عمدتا در مدارهای سری پیاده می‌شوند و این موضوع باعث تاخیر در مدار می‌شود. برای بهبود این مشکل و کاهش تاخیر در پردازش بهترین روش مراجعه به روش-های هوشمند خواهد بود. اگرچه این کدگذاری-ها اشکال را برطرف می-کنند، اما حذف الگوهای اشکال همشنوایی بصورت سنتی دقت و سرعت کافی ندارند. از طرفی تا کنون از هوش مصنوعی و الگوریتم-های یادگیری ماشین برای کد گذاری و مقابله با اشکال همشنوایی استفاده نشده است، درحالیکه روش-های هوشمند اغلب از هوش مصنوعی، شبکه های یادگیر و یادگیری ماشین استفاده می‌کنند که به دلیل فاز یادگیری آنها، دقت و سرعت بسیار بالایی دارند.[۲]

در این مقاله، با به کار گیری الگوریتم شبکه‌های عصبی پیچشی روشی ارائه خواهیم کرد که با استفاده از یادگیری ماشین موجب حذف الگوهای گذار نامناسب و مضر می‌شود. الگوریتم‌های شبکه عصبی پیچشی ویژگی‌های مهم را از داده‌های استخراج می‌کنند و می‌توانند با دقت بالایی که دارند الگوهای پیچیده تری را شناسایی کنند.[۴] این الگوریتم‌ها انعطاف پذیر بوده و می‌توانند با طیف گسترده‌تری از داده‌های ورودی کارکند. الگوریتم-های شبکه عصبی پیچشی با استفاده از یادگیری عمیق، به طور مداوم مدل را بهبود داده

و باعث کاهش خطاهای ناشی از طراحی ویژگی‌ها می‌شوند.[۵] و در نتیجه می‌توانند تمامی کلاس‌ها را مورد بررسی و کاهش قرار دهند.

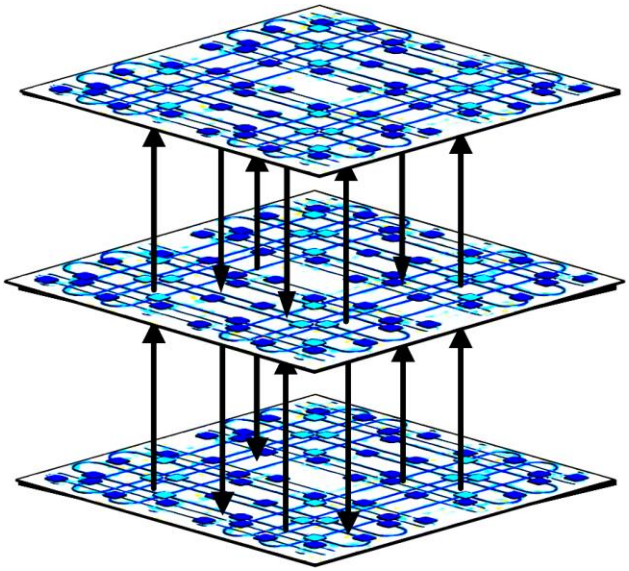
طبق بررسی هایی که صورت گرفته است، شدت اشکال هم شنوایی به داده‌های متوالی که برروی خطوط عمودی ظاهری بستگی دارد.[۶] در این روش الگوریتم هایی ارائه می‌شود که موجب حذف الگوهای گذار نامناسب و مضر در سیم های شبکه‌های روی تراشه سه بعدی می‌شود؛ تا در فرآیند کاهش اشکال شاهد افزایش سرعت و کارایی مدار باشیم.[۷; ۸]

در ادامه، در بخش دوم، پیش‌زمینه ای از موضوع بیان می‌شود؛ سپس در بخش سوم کارهای پیشین و روش‌هایی برای رفع اشکال بررسی میشود و در بخش چهارم با توجه به مزایا و معایب روش های بیان شده، روشی بهینه پیشنهاد شده و در پایان، نتیجه گیری مقاله ارائه خواهد شد.

۱. پیش زمینه

به دلیل پیچیده‌ترشدن کارهای پردازشی و همچنین نیاز به پردازشگرهای قدرمند برای پاسخ به کاربردهای پردازشی پیچیده و در عین حال کوچک شدن بی وقفه ابعاد پردازشگرها برای دستیابی به توان پردازشی و سرعت بیشتر و توان مصرفی کمتر، نیاز به معرفی یک فناوری جدیدتر و قدرتمند است؛ تا محدودیت-هایی که روش های فیزیکی ایجاد می-کند را برطرف کنیم. بنابراین باید به دنبال راهکاری برای افزایش تعداد و چگالی ترانزیستورهای روی تراشه بود و درعین‌حال از مساحت موجود تا حد امکان بهره بگریم تا بتوانیم به ویژگی های مطلوب یک سامانه روی تراشه که شامل کارایی زیاد، توان پردازشی بیشتر و همچنین فشرده‌سازی قابل قبول در عین صرف هزینه ی کم و توان مصرفی پایین دست یابیم.[۹]

همانطور که در مقدمه ذکر شد، سیستم‌های مورد استفاده در پردازشگرها، از شبکه‌های روی تراشه‌های دوبعدی استفاده می‌کنند. قطر زیاد شبکه در تراشه‌های دوبعدی منجر به افزایش تاخیر در سطح سامانه می-شود. همچنین با توجه افزایش تعداد هسته‌های پردازشی در آینده به صدها هسته و نیاز به ارتباط-دهی میان هسته-ها؛ این طراحی و پیاده سازی قابل اطمینان نخواهد بود.[۷] به همین منظور شبکه‌های روی تراشه- سه-بعدی معرفی شده و مورد استفاده قرار گرفتند. نمای کلی شبکه-های روی تراشه-های سه-بعدی در شکل۱ قابل مشاهده است.[۱۰]



شکل۱. نمای کلی شبکه-های روی تراشه سه-بعدی

همانطور که در شکل نشان داده شده، یک تراشه سه-بعدی از چندین لایه‌ی سیلیکونی که به صورت عمودی روی هم قرار گرفته-اند، تشکیل شده است. با قرا گرفتن لایه های مختلف بر روی هم، میانگین طول اتصالات نسبت به تراشه های دو بعدی کاهش یافته و در نتیجه توان مصرفی و زمان انتقال بسته‌های داده کاهش یافته و مسیر های بحرانی بهتر مدیریت می‌شود.

باید توجه داشت که علی‌رغم مزایای مذکور، ارتباط بین لایه های سیلیکون، از طریق گذرگاه‌های ارتباط بین لایه‌ای برقرار می‌شوند که هزینه ساخت بالایی دارد و همچنین این اتصالات باید از درون لایه سیلیکون عبور کنند و ابعاد آن‌ها در مقایسه با دیگر اتصالات در سطح شبکه روی تراشه، بسیار بزرگ‌تر است. اگرچه این تراشه‌ها با تعداد زیادی هسته دارای قدرت پردازشی بالایی هستند، اما وجود چندین هسته پردازشی در یک تراشه، منجر به بروز مشکلات جدیدی شده است.[۱۱]

یکی از مهم‌ترین چالش‌های قابلیت اطمینان در شبکه‌های روی تراشه، اشکال همشنوایی است. این اشکال که در کانال‌های ارتباطی مابین هسته‌های پردازشی رخ می‌دهد، متناسب با افزایش طول مجموع سیم‌های کانال روی تراشه افزایش می‌یابد.[۷] این افزایش طول در آینده بسیار جدی خواهد بود. پیش-بینی شده است که مجموع طول سیم‌های ارتباطی روی تراشه در سال ۲۰۲۰ به حدود۷۰۰۰ متر بر سانتی‌متر مربع برسد.[۱۲]

با توجه به میزان اهمیت مقابله با اشکال همشنوایی در شبکه‌های روی تراشه سه بعدی، باید رویکردهایی برای افزایش قابلیت اطمینان در کانال-ها و کاهش اشکال همشنوایی در نظر گرفته شود.

جدول ۱.الگوهای همشنوایی بین گذرگاههای عمودی در شبکه روی تراشه سه-بعدی[۳]

کلاس همشنوایی	الگوی گذار	کلاس همشنوایی	الگوی گذار
کلاس ۰		کلاس ۱	
کلاس ۲		کلاس ۳	
کلاس ۴		کلاس ۴	
کلاس ۵		کلاس ۵	
کلاس ۶		کلاس ۶	
کلاس ۷		کلاس ۸	

اکثر پژوهش‌های پیشین از یک شبکه‌ی مش دوبعدی ۳×۳ از گذرگاه‌های عمودی بین لایه-ای همانند جدول ۲ استفاده می کنند به نحوی که از اثرات تزویج سلفی صرف نظر شده است و همچنین از اثر تزویج خازنی بین یک اتصال عمودی با همسایه‌های قطری به دلیل فاصله بیشتر آن نسبت به همسایه‌های مستقیم چشم پوشی شده‌است. همسایه-های افقی و عمودی به دلیل فاصله کمتر از سیم قربانی دارای اثر خازنی حدود بیش از پنج برابری نسبت به همسایه-های قطری میباشد.[۱۳]

می‌توان نه کلاس مختلف را مطابق الگوهای جدول۲ برای شبکه-های روی تراشه سه بعدی در نظر گرفت. برای مثال در کلاس ۸ تمامی سیم‌های اطراف سیم قربانی در جهت عکس آن تغییر می‌کنند که این مو جب ایجاد خازن تزویجی متعلق به کلاس ۸ خواهد شد. البته در شکل تمامی حالات ممکن بیرای شبکه روی تراشه سه بعدی آورده نشده است و جایگشت-های مختلف از این کلاس-ها نیز باید در نظر گرفته شود.

۲. کارهای پیشین

از ساده‌ترین روش‌های مقابله با اشکال در سطح فیزیکی، حفاظ‌گذاری سیم‌هاست که به دلیل سربار مساحت و هزینه زیاد چندان مورد توجه قرار نمی-گیرد.[۱۴; ۱۵]همچنین روش‌های در سطح ترانزیستوری

مانند روش تغییر زمان‌بندی ارسال سیگنال با توجه به کم‌بودن کارایی و عملکرد مورد استفاده قرار نمی‌گیرند.[۱۲; ۱۶]

در روش درج تکرارکننده‌ها با تقسیم بندی سیم بهN قسمت و قرار دادن یک وارونگر یا میانگیر در هر قسمت، میتوان تأخیر کل سیستم را کاهش داد.[۱۷] این میانگیر یا وارونگر را تکرار کننده می‌نامند .البته برای اینکه بتوان رابطه‌ی تأخیر و طول سیم را به صورت خطی کاهش داد ،انتخاب مکان مناسب برای درج تکرارکننده بسیارمهم می‌باشد. این روش نیز با سربار مساحت زیادی که ناشی از افزودن وارونگر یا میانگیر است ،همراه می‌باشد.

با استفاده از کدگذاری اجتناب از همشنوایی مبتنی بر سیستم عددی می‌توان داده‌ای که بر روی گذرگاه جابه جا می‌شود را به صورتی کد کرد که الگوهایی که تاخیر بیشتری را بر آن تحمیل می‌کنند حذف شوند.[۱۸] با این وجود، ضعف الگوریتم نگاشت مدارات کدگذار و کدگشا در تولید کلمه‌های کد مشکلی است که کدگذاری‌های اجتناب از همشنوایی مبتنی بر سیستم عددی دودویی دارند،[۱۹] به طوری‌که سربار سیم‌های اضافی را به سیستم تحمیل می‌کنند و این باعث افزایش عرض کانال به‌صورت نمایی می‌گردد. [۷; ۱۹]

اگرچه تاکنون کدهای مختلفی برای بهبود این شرایط ارائه شدند، ولی هیچکدام نتوانستند جلوی این رشد نرخ اندازه‌ی عرض گذرگاه را بگیرند و تنها در بهترین حالت توانستند این رشد نمایی را به رشد خطی برسانند.

با توجه به مشکلات موجود در روش‌های قبلی کاهش اشکال مثل سربار مساحت بسیار زیاد برخی از روش‌ها، روش کدگذاری اجتناب از تداخل پیشنهاد شده‌است که با سربار مساحت کم‌تر اقدام به حذف گذارهای مضر می‌کند. دسته‌ای از کدهای اجتناب از تداخل با نام کدگذاری-های الگو ممنوعه وجود دارند.[۲۰; ۲۱]

در این کدگذاری-ها الگوهای ۱۰۱ و ۰۱۰ غیر مجاز می‌باشد و با توجه به نوع کدگذاری که انجام می‌دهیم به-دنبال این هستیم که الگوهای غیرمجاز و مضر را حذف کنیم.

حالات معتبر و نامعتبر کدهای الگوی ممنوعه با عرض گذرگاه ۳ و ۴ بیتی در جدول ۱ آورده شده‌است. انواع کدگذاری‌های الگوی ممنوعه عبارتند از FPF-CAC، WI-OLC و LO-OLC که به اختصار به معرفی هرکدام می‌پردازیم.[۲۲]

$$f_m = \begin{cases} 0, & if\ m = 0 \\ 1, & if\ m = 1 \\ f_{m-1} + f_{m-2} & if\ m \geq 2. \end{cases} \quad (۱)$$

برای تولید کدهای الگوی ممنوعه معتبر از سیستم‌های عددی مختلف استفاده می‌شود که آن سیستم عددی باید کامل باشد و همچنین سیستم عددی باید بتواند هر عدد موجود را به یک کد الگوی ممنوعه تبدیل کند. یکی از سیستم‌های عددی قابل استفاده برای تولید کدهای الگو ممنوعه، دنباله فیبوناچی است که در رابطه۱ ارائه شده است. در این رابطه(f_m) برابر مقدار عدد فیبوناچی در موقعیت(m) و (m) اندیس عدد فیبوناچی در دنباله می‌باشد. از اعداد این دنباله به عنوان یک سیستم عددی برای کدگذاری و کاهش اشکال

همشنوایی در شبکه‌های روی تراشه‌ها به‌کار می رود. برای مثال اعداد ۱ ۱ ۲ ۳ ۵ برای تولید کدها در کانال ۵ بیتی و اعداد ۱ ۱ ۲ ۳ ۵ ۸ برای تولید کدها در کانال ۶ بیتی استفاده می‌شوند. به عنوان نمونه در کدگذاری با استفاده از دنباله فیبوناچی، کدهای ۰ ۱ ۱ ۱ ۱ ۰ و ۱ ۰ ۰ ۱ ۱ ۰ ، بیانگر عدد ۱۱ است.

استفاده از سیستم عددی فیبوناچی با مشکلات و نقاط ضعفی همراه است. ضعف اول پیچیدگی زیاد کدگشا/کدگذارهایی است که در این روش استفاده می‌شود و ضعف دوم ابهام این سیستم عددی است که همانطور که در مثال بالا دیده شد در برخی موارد برای یک مقدار مشخص بیش از یک کدگذاری تولید می‌شود.

روش دیگر کدگذاری الگوی ممنوعه، تولید کدها با استفاده از پایه-هایی است که از سیستم-های عددی Wu-OLC به دست می-آیند. در این روش کدهای جدید با استفاده از سیستم عددی Wu-OLC تولید می‌شوند که الگوهای ۰۱۰ ، ۱۰۱ ، ۱۰۰۱ و ۰۱۱۰ نمی‌توانند در آن-ها ظاهر شوند. برای مثال در گذرگاه ۶ بیتی سیستم عددی تولید شده ۱۰۱۱۱۴ می‌باشد که به عنوان نمونه برای کلمه داده ۵، کد ۰۰۰۰۱۱ را تولید می-کند.[۲۳]

یکی از بهترین و اثرگذارترین روش‌های کدگذاری برای کاهش اشکال استفاده از روش کدگذاری LO-OLC است. دلایلی که استفاده از این روش را نسبت به سایر روش-های کدگذاری متمایز ساخته این است که اولا برخلاف سایر روش‌ها بر مبنای مدل سه‌سیمی ارائه شده بودند، کدگذاری LO-OLC بر مبنای مدل پنج‌سیمی ارائه شده است و همچنین سربار مساحت و توان مصرفی کمتری را به مدار تحمیل می‌کند.[۱۳]

جدول۲. حالت-های معتبر و نامعبتر کدهای الگوی ممنوعه سه و چهار بیتی [۲۳]

عرض گذرگاه	حالات معتبر کدهای الگو ممنوعه	حالات نامعتبر
۳	۰۰۰ ، ۰۰۱ ، ۱۰۰ ، ۱۱۰ ، ۰۱۱ ، ۱۱۱	۰۱۰ ، ۱۰۱
٤	۰۰۰۰ ، ۰۰۰۱ ، ۰۰۱۱ ، ۰۱۱۰ ، ۰۱۱۱ ، ۱۰۰۰ ، ۱۰۰۱ ، ۱۱۰۰ ، ۱۱۱۰ ، ۱۱۱۱	۱۰۱۰ ، ۰۱۰۱ ، ۱۰۱۱ ، ۱۱۰۱ ، ۰۱۰۰ ، ۰۰۱۰

۳. روش پیشنهادی

الگوریتم رگرسیون خطی برای پیش‌بینی مقادیر پیوسته توسط یافتن یک رابطه خطی بین متغیر وابسته و یک یا چند متغیر مستقل، استفاده می-شود.[۲۴; ۲۵]

شبکه عصبی نیز یک مدل محاسباتی است که عملکرد آن، الهام گرفته از شبکه-های عصبی بیولوژیکی موجود در مغز انسان است. این شبکه-ها، سیستم-هایی پیچیده از تعداد نسبتا زیادی واحد هـای پردازشـی به- نام نورون تشکیل شده-اند.[۳; ۴; ۲۶]

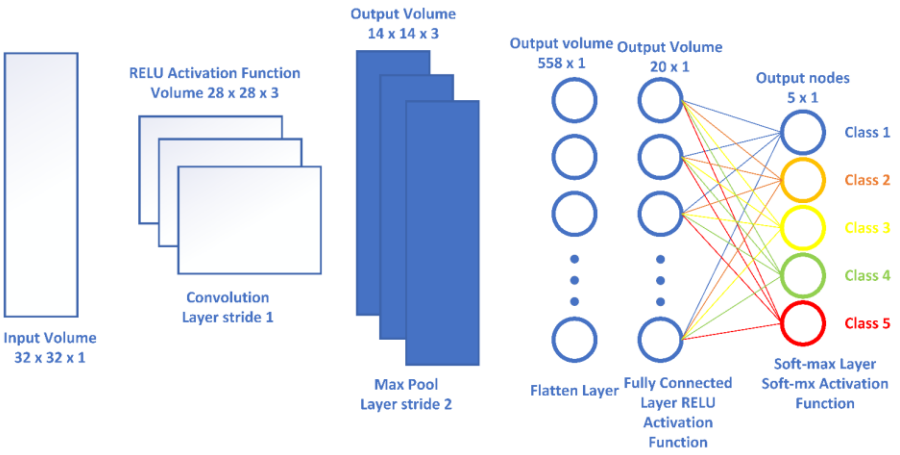
در این مقاله از معماری الگوریتم شبکه‌های عصبی پیچشی یا به اختصـار CNN[۱] بهـره بـرده ایـم کـه باعث کاهش تاخیر و مصرف انرژی در شبکه های روی تراشه شده و عملکرد پردازشی را افزایش مـی-دهـد. همانطور که در شکل ۲ نمایش داده شده، این معماری از لایه-هایی به شرح ذیل تشکیل شده-اند.[۲۷; ۲۸]

لایه واحد خطی یکسوسازی شده: این شبکه‌ها دارای یک لایه ۲[ReLU] برای انجـام عملیـات روی عناصر هستند. خروجی آن، یک نگاشت ویژگی یکسوسازی شده است.

لایه جمع آوری کننده: نگاشت ویژگی یکسوسازی شده، در ادامه به یک لایه پولینـگ داده مـی‌شـود. پولینگ، یک عملیات نمونه برداری کاهشی است که ابعاد نگاشت ویژگی را کاهش می‌دهد. این لایه، سـپس آرایه‌های دو بعدی حاصل از پولینگ نگاشت ویژگی را مسطح یا هموار) کرده و به یک بردار واحد، طـولانی، پیوسته و خطی تبدیل می‌کند.

لایه تماما متصل : یک لایه تماما متصل، هنگامی ایجـاد مـی‌شـود کـه مـاتریس همـوار شـده از لایـه پولینگ، به عنوان ورودی داده شود، که تصاویر را دسته بندی و شناسایی می‌کند.

لایه خروجی (ReLU / Sigmoid): در طبقـه‌بنـدی‌هـای چندکلاسـه: از ReLU بـرای نرمـال‌سـازی خروجـی اسـتفاده شـده اسـت همچنـین در تشـخیص دوکلاسـه (Binary Classification): از Sigmoid برای پیش‌بینی خروجی استفاده شده است.

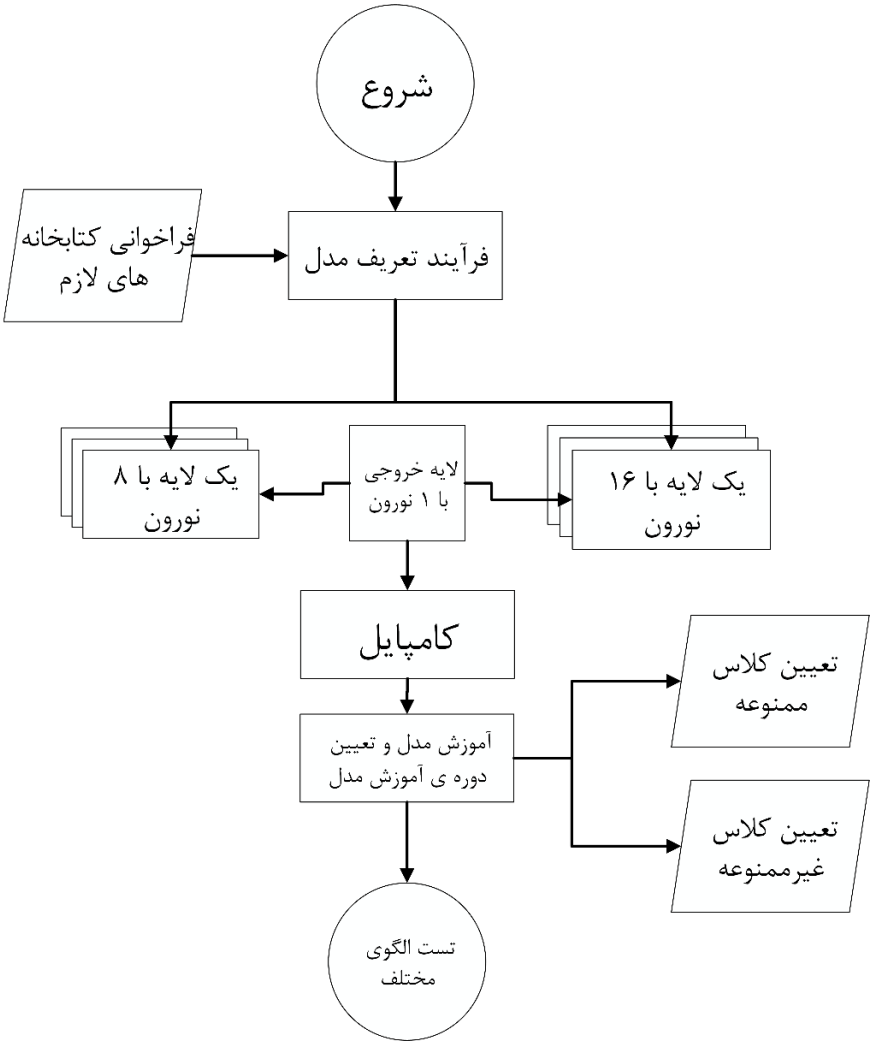


شکل۲. لایه های شبکه عصبی پیچشی[۲۹]

داده های مورد استفاده از داده های واقعی بوده و ۷۰ درصد از داده ها برای آموزش، ۲۰ درصـد بـرای تست ، و۱۰ درصد نیز برای اعتبار سنجی در نظر گرفته و استفاده شده است.

الگوریتم پیاده-سازی کد هوشمند پیاده شده به صورت زیر است:

- [شروع]
- [تعریف مدل]
  - تعریف مدل شبکه عصبی:
    - ورودی با اندازه ۵ تعریف می-شود.
    - یک لایه با ۱۶ نورون و فعال-سازی relu ایجاد می-شود.
    - یک لایه خروجی با ۱ نورون و فعال-سازی relu ایجاد می-شود.
    - یک لایه خروجی با ۱ نورون و فعال-سازی sigmoid تعریف می‌شود.
    - مدل کامپایل می-شود با optimizer='adam'، loss='binary_crossentropy' و معیار 'accuracy'.
- [تعریف داده‌ها]
  - تعریف داده‌های ورودی برای آموزش مدل:
    - الگوی ممنوعه با ورودی های [۱, ۱, ۱, ۱, ۰] و برچسب ۱ تعریف می‌شود.
    - الگوی غیر ممنوعه با ورودی های [۰, ۰, ۰, ۰, ۰] و برچسب ۰ تعریف می‌شود.
    - داده‌های دو کلاس ترکیب می شوند.
- [آموزش مدل]
  - آموزش مدل با ۲۰۰ دوره آموزش:
    - داده های آموزش برای مدل داده می‌شوند.
    - مدل آموزش داده می‌شود.
- [تست الگو]
  - الگوی ممنوعه با ورودی-های [۱, ۱, ۱, ۰, ۱] تست می-شود:
    - پیش‌بینی مدل برای این الگو انجام می-شود.
    - اگر پیش‌بینی بیشتر یا مساوی با ۰,۵ باشد:
      - الگوی ممنوعه شناسایی می-شود.
    - در غیر این صورت:
      - الگوی ممنوعه شناسایی نمی-شود.
- [پایان]



شکل ۳. فلوچارت هندسی الگوریتم

در تعریف مدل عصبی از دو تابع فعال-ساز ReLU[۲۸] و sigmoid [۳۰]استفاده کرده-ایم: لایه با ۱۶ نورون و فعال-سازی ReLU به معنای این است که در شبکه عصبی، این لایه شامل ۱۶ واحد یا نورون می‌شود که هر نورون ورودی های مخصوص خود را دارد و وظیفه آنها انتقال و پردازش اطلاعات است. تابع فعال-سازی ReLU به شکل رابطه ۲ تعریف می-شود[۳۱]:

$$f(x)=\max(0,x) \qquad (۲)$$

به این معنا که اگر ورودی(x) که می‌تواند مقدار هر نورون در یک لایه از شبکه عصبی باشد، بزرگتر از صفر باشد، مقدار خروجی(f (x)) برابر با خود ورودی (x) خواهد بود؛ اما اگر ورودی کوچکتر یا مساوی صفر باشد، مقدار خروجی برابر با صفر خواهد بود.

تابع فعال‌سازی sigmoid یک تابع ریاضی است که معمولا در شبکه‌های عصبی به عنوان تابع فعال-سازی در لایه‌های آخر برای تبدیل خروجی به مقادیر احتمالی بین ۰ و ۱ استفاده می-شود.

تابع فعال‌سازی sigmoid به صورت رابطه ۳ تعریف می‌شود[۳۱]:

$$f(x) = \frac{1}{1+e^{-x}} \quad (۳)$$

که (x) نمایانگر مقدار وزن دار خروجی یک نورون یا جمع وزن دار خروجی های یک لایه از نورون-ها است و (e) نمایانگر عدد اورولر است.

تابع سیگموید با تبدیل کردن اعداد ورودی به مقادیر بین ۰ و ۱، به شبکه عصبی اجازه می‌دهد که احتمال یا اطمینان در مورد یک ویژگی یا خروجی را محاسبه کند به صورتی که اگر مقدار خروجی بسیار نزدیک به ۰ باشد، به معنای اطمینان پایین یا تشخصی منفی و بالعکس وقتی مقدار خروجی بسیار نزدیک به ۱ باشد، به معنای اطمینان بالا در مورد وجود ویژگی یا تشخیص مثبت است.

نمی‌توانیم با رگرسیون لجستیک این کد را پیاده‌سازی کنیم زیرا الگوریتم رگرسیون لجستیک باید حداقل دو کلاس برای آموزش داشته باشد و آن هم بدین صورت که یک کلاس برای الگوی ممنوعه و دیگری الگوی مثبت یا غیرممنوعه باشد.[۳۲] ولی ما جایگشت های مطلوب در شبکه‌های تراشه سه-بعدی را در اختیار نداریم؛ پس کار را با الگوریتم شبکه‌های عصبی ادامه می دهیم.

این الگوریتم ها بر روی کامپیوتری با مشخصات فنی به شرح زیر و با استفاده از زبان برنامه نویسی python و کتابخانه های مربوطه(tensorflow,numpy,...) پیاده سازی و اجرا شده و نتایج آن در بخش نتایج ارائه شده است.

مشخصات فنی کامپیوتر:

- پردازنده مرکزی (CPU): Intel Core i7-۹۷۰۰ k ۳.۶ GHz
- پردازنده گرافیکی: NVIDIA GTX ۱۶۶۰ Ti
- حافظه(RAM): 16GB DDR4
- حافظه ذخیره سازی: SSD NVMe 512GB
- سیستم عامل(OS): Ubuntu 18.4

۵. نتایج

با استفاده از الگوریتم‌های یادگیری عمیق، روش ارائه شده در این مقاله قادر است الگوهای پیچیده‌تر را نیز شناسایی کند و انعطاف‌پذیری بیشتری در مواجهه با داده‌های ورودی مختلف از خود نشان دهد. این

ویژگی باعث می‌شود که روش پیشنهادی در مقایسه با روش‌های سنتی کدینگ، که با محدودیت‌هایی مانند سربار افزایشی، پیچیدگی بالا و مصرف انرژی بیشتر مواجه هستند، عملکرد بهتری داشته باشد.

برای ارزیابی میزان کارایی الگوریتم های بکار رفته در روش ارائه شده، میزان پیش‌بینی الگوهای مضر نسبت به میزان واقعی آن در هر کلاس را مورد بررسی قرار داده‌ایم. نتایج حاصله که در جدول۳ ارائه شده‌است، نشان می‌دهد در کلاس های ۱و۲ که کلاس‌های کم اهمیت‌تری نسبت به کلاس‌های خطرناک‌تر است،الگوریتم دقت شناسایی پایین‌تری دارد که این موضوع می‌تواند به دلیل تمرکز الگوریتم بر شناسایی کلاس های مهم‌تر به وجود آمده باشد. در کلاس‌های ۳ تا ۵ الگوریتم توانسته تا ۹۰ درصد دقت پیش‌بینی خود را بالا ببرد و عملکرد قابل قبولی داشته باشه. درکلاس‌های ۶ تا ۸ که شدت همشنوایی بالاتر است و خطرناک ترین و با اهمیت ترین کلاس‌های همشنوایی محسوب می‌شوند، الگوریتم عملکرد بسیار مطلوب نشان داده و دقت پیش‌بینی آن تا میزان ۹۶ درصد افزایش یافته است.

این نتایج نشان می‌دهد که الگوریتم پیشنهادی به طور کلی عملکرد خوبی در شناسایی و حذف الگوهای مضر داشته است و می‌تواند به عنوان یک راهکار مؤثر برای کاهش همشنوایی در شبکه‌های روی تراشه-های سه‌بعدی مورد استفاده قرار گیرد.

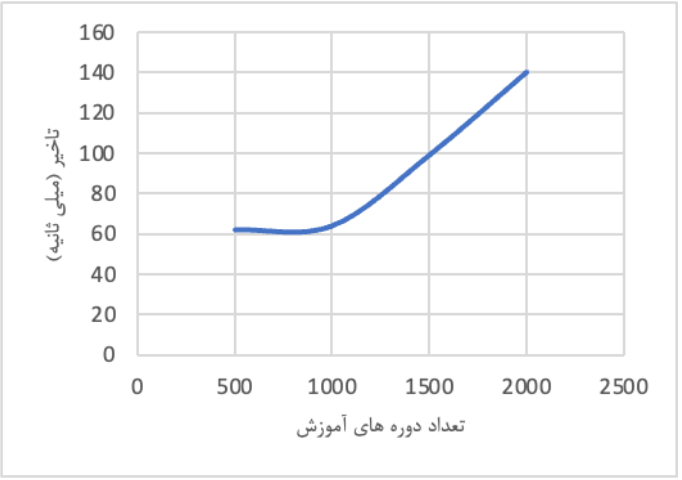
جدول۳. مقایسه‌ی میزان پیش‌بینی و مقادیر واقعی کاهش الگوهای مضر در هر کلاس از هم‌شنوایی

شماره کلاس	میزان پیش‌بینی شده	میزان واقعی
۱	۰	۱/۵
۲	۰	۱
۳	۵	۴/۵
٤	۱۳	۱۳/۲
۵	۱۰	۷/۸
٦	۳۲	۳۰
۷	۲۵	۲۴
۸	۴۳	۴۵

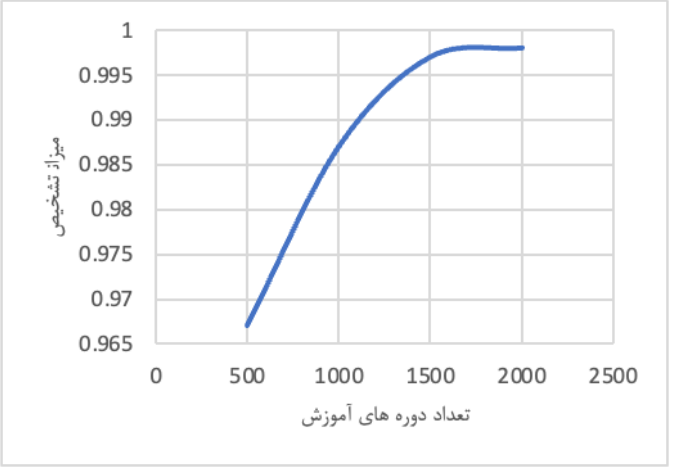
از طرفی با توجه به نمودار ۱ با افزایش تعداد دوره‌های آموزش، میزان تاخیر اجرای این کد بسیار بالا خواهد بود؛ به‌طوری که هسته‌ها سربار محاسباتی زیادی خواهند داشت و از این رو آنچه در ادامه پیشنهاد می‌شود پیاده سازی کدهایی با حجم بسیار کم و زمان اجرای بسیار مطلوب خواهد بود.

همچنین، طبق نمودار ۲ با افزایش تعداد دوره، قابلیت تشخیص شبکه‌های عصبی به طور قابل توجهی افزایش می‌یابد و با افزایش بیشتر دوره‌ها تغییرات قابلیت تشخیص کاهش می‌یابد.

نمودار۱. تاخیر بر اساس دوره-های آموزش



نمودار۲. تشخیص بر اساس دوره-های آموزش



۶. نتیجه‌گیری

در این مقاله با استفاده از الگوریتم شبکه‌های عصبی توانستیم برای مدیریت کلاس هشت به ویژه کلاس‌های با شدت همشنوایی بالا(کلاس‌های ۶ تا ۸) در شبکه تراشه‌های سه‌بعدی کدی با ضریب دقت بالا تولید کنیم که با دقت بیش از ۹۰ درصد در شناسایی و حذف الگوهای مضر موفق بوده است. با این حال، چالش‌هایی نیز در پیاده‌سازی این روش وجود دارد. به عنوان مثال، الگوریتم در کلاس‌های با شدت همشنوایی پایین دقت (کلاس‌های ۱و۲) دقت کمتری نشان داده‌است که می‌تواند در آینده با بهبود مدل یادگیری عمیق، برطرف شود. همچنین با افزایش تعداد دوره‌های آموزش، زمان اجرای الگوریتم افزایش می‌یابد که این موضوع می‌تواند باعث ایجاد سربار محاسباتی شود. برای رفع این مشکل، پیشنهاد می‌شود که در آینده از روش‌های بهینه‌سازی مانند استفاده از جداول کارنو یا پیاده‌سازی الگوریتم‌های کم‌حجم‌تر برای کاهش زمان اجرا و بهبود کارایی سیستم استفاده شود.

در نهایت، این روش می‌تواند به عنوان یک راهکار مؤثر در طراحی تراشه‌های آینده مورد استفاده قرار گیرد. با بهبود بیشتر الگوریتم‌ها و کاهش زمان اجرا، این روش می‌تواند به کاهش تأخیر و بهبود عملکرد کلی سیستم‌های پردازشی کمک کند. با توجه پردازش طولانی الگوریتم های این مقاله، می‌توان با استفاده از پیاده سازی با جدول کارنو و به‌دست آوردن فرمول جبری، سرباز زمانی الگوریتم را بسیار کاهش دهیم یا با پیاده‌سازی الگوریتم CNN کدهای خود را بهینه کنیم؛ تا در پارامتر های زمان اجرا و تشخیص نیز به سرعت و دقت لازم دست یابیم. بهینه‌سازی الگوریتم‌ها و تنظیم تعداد دوره‌های آموزشی می‌تواند به کاهش تأخیر و بهبود عملکرد کلی سیستم کمک کند. این امر می‌تواند منجر به افزایش کارایی و بهره‌وری در کاربردهای واقعی شود.

فهرست منابع

[1] Sheibanyrad, A., Pétrot, F., & Jantsch, A. (2010). *3D Integration for NoC-based SoC Architectures*. Springer New York, NY. https://doi.org/10.1007/978-1-4419-7618-5

[2] Mahdavi, Z. (2016). *Reliability Improvement in Network on Chips against Crosstalk Fault Considering Five Wire Latency Model* [Master's thesis, Sharif University of Technology]. https://library.sharif.ir/parvan/resource/439878/%D8%A7%D9%81%D8%B2%D8%A7%DB%8C%D8%B4-%D9%82%D8%A7%D8%A8%D9%84%DB%8C%D8%AA-%D8%A7%D8%B7%D9%85%DB%8C%D9%86%D8%A7%D9%86-%D8%B4%D8%A8%DA%A9%D9%87-%D9%87%D8%A7%DB%8C-%D8%B1%D9%88%DB%8C-%D8%AA%D8%B1%D8%A7%D8%B4%D9%87-%D8%AF%D8%B1-%D8%A8%D8%B1%D8%A7%D8%A8%D8%B1-%D8%A7%D8%B4%DA%A9%D8%A7%D9%84-

%D9%87%D9%85%D8%B4%D9%86%D9%88%D8%A7%DB%8C%DB%8C-%D8%A8%D8%A7-%D8%AF%D8%B1-%D9%86%D8%B8%D8%B1-%DA%AF%D8%B1%D9%81%D8%AA%D9%86-%D9%85%D8%AF%D9%84-%D8%AA%D8%A3%D8%AE%DB%8C%D8%B1-%D9%BE%D9%86%D8%AC%E2%80%8C%D8%B3%DB%8C%D9%85%DB%8C/&from=search&&query=49234&count=20&execute=true

[3] Mirosanlou, R. (2015). *Reliability Improvement in 3D Network on Chips Against Crosstalk Fault* [Master's degree, Sharif University of Technology]. https://library.sharif.ir/parvan/resource/421553/%D8%A7%D9%81%D8%B2%D8%A7%DB%8C%D8%B4-%D9%82%D8%A7%D8%A8%D9%84%DB%8C%D8%AA-%D8%A7%D8%B7%D9%85%DB%8C%D9%86%D8%A7%D9%86-%D8%B4%D8%A8%DA%A9%D9%87%E2%80%8C%D9%87%D8%A7%DB%8C-%D8%B1%D9%88%DB%8C-%D8%AA%D8%B1%D8%A7%D8%B4%D9%87-%D8%B3%D9%87-%D8%A8%D8%B9%D8%AF%DB%8C-%D8%AF%D8%B1-%D8%A8%D8%B1%D8%A7%D8%A8%D8%B1-%D8%A7%D8%B4%DA%A9%D8%A7%D9%84-%D9%87%D9%85%D8%B4%D9%86%D9%88%D8%A7%DB%8C%DB%8C/&from=search&&query=Mirosanlou&field=authorMain&count=20&execute=true

[4] Noori, A. (2022). A New Method for Detecting Influential Nodes in Social Network Graphs Using Deep Learning Techniques. *Karafan Quarterly Scientific Journal*, *19*(1), 607–628. https://doi.org/10.48301/kssa.2022.310565.1786

[5] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

[6] Nguyen, H., Kim, C.-H., & Kim, J. (2018). Effective Prediction of Bearing Fault Degradation under Different Crack Sizes Using a Deep Neural Network. *Applied Sciences*, *8*(11), 2332. https://doi.org/10.3390/app8112332

[7] Flayyih, W. N., Samsudin, K., Hashim, S. J., Ismail, Y. I., & Rokhani, F. Z. (2016). Adaptive Multibit Crosstalk-Aware Error Control Coding Scheme for On-Chip Communication. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *63*(2), 166–170. https://doi.org/10.1109/TCSII.2015.2483379

[8] Chen, G., & Nooshabadi, S. (2017, 30 April–3 May 2017). *Analysis and design of serial error correction code with crosstalk avoidance technique* 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), https://doi.org/10.1109/CCECE.2017.7946606

[9] Ramakrishnan, N., Sanju, V., & Farhanaaz. (2023, 16–17 June 2023). *Analysis of Network on Chip Topologies* 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), https://doi.org/10.1109/ICAISC58445.2023.10200489

[10] Cheah, B., Kong, J., Yong, K. C., Lo, L., & Yaw, P. (2014). Crosstalk study of high speed on-package interconnects for multi-chip package. *2014 IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 381–385. https://doi.org/10.1109/ISEMC.2014.6899001

[11] Reza, M. F. (2023). Machine Learning Enabled Solutions for Design and Optimization Challenges in Networks-on-Chip based Multi/Many-Core Architectures. *ACM Journal on Emerging Technologies in Computing Systems*, *19*(3), 1–26. https://doi.org/10.1145/3591470

[12] Höfflinger, B. (2012). *Chips 2020 : : a guide to the future of nanoelectronics*. Springer. https://doi.org/10.1007/978-3-642-23096-7

[13] Abed, H., & Flayyih, W. (2023). Comparative Reliability Analysis between Horizontal-Vertical-Diagonal Code and Code with Crosstalk Avoidance and Error Correction for NoC Interconnects. *Journal of Engineering*, *29*(7), 120–136. https://doi.org/10.31026/j.eng.2023.07.08

[14] Mehri, M., & Masoumi, N. (2015). A thorough investigation into active and passive shielding methods for nano-VLSI interconnects against EMI and crosstalk. *AEU - International Journal of Electronics and Communications*, *69*(9), 1199–1207. https://doi.org/https://doi.org/10.1016/j.aeue.2015.04.018

[15] Partin-Vaisband, I., Popovich, M., Köse, S., Jakushokas, R., Mezhiba, A., & Friedman, E. G. (2016). *On-Chip power delivery and management, fourth edition*. Springer Cham. https://doi.org/10.1007/978-3-319-29395-0

[16] Hirose, K., & Yasuura, H. (2000, 27–30 March 2000). *A bus delay reduction technique considering crosstalk* Proceedings Design, Automation and Test in Europe, Paris, France. https://doi.org/10.1109/DATE.2000.840308

[17] Lu, Q., Zhu, Z., Yang, Y., & Ding, R. (2016). Analysis of propagation delay and repeater insertion in single-walled carbon nanotube bundle interconnects. *Microelectronics Journal*, *54*, 85–92. https://doi.org/https://doi.org/10.1016/j.mejo.2016.05.012

[18] Melo, D. R., Zeferino, C. A., Bezerra, E. A., & Dilillo, L. (2021, 28–30 June 2021). *Design and Evaluation of Implementation Impact on a Fault-Tolerant Network-on-Chip Router* 2021 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), Montpellier, France. https://doi.org/10.1109/DTIS53253.2021.9505053

[19] Mutyam, M. (2012). Fibonacci Codes for Crosstalk Avoidance. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *20*(10), 1899–1903. https://doi.org/10.1109/TVLSI.2011.2162010

[20] Shirmohammadi, Z., Khorami, A., & Omana, M. (2021). ST-CAC: a low-cost crosstalk avoidance coding mechanism based on three-valued numerical system. *The Journal of Supercomputing*, *77*, 6692–6713. https://doi.org/10.1007/s11227-020-03527-0

[21] Mercier, R. (2021). *Multiple fault mitigation in network-on-chip architectures through a bit-shuffling method* (Publication Number 2021REN1S123) Université de Rennes]. https://theses.hal.science/tel-03716432

[22] Shirmohammadi, Z. (2020). Crosstalk Fault Tackling by Providing an Efficient Tri-Value Crosstalk Avoidance Coding Mechanism for NoCs. *TABRIZ JOURNAL OF ELECTRICAL ENGINEERING*, *49*(4), 1721–1731. https://tjee.tabrizu.ac.ir/article_10429_5d036c2cf601df943fe70410909f20a3.pdf

[23] Taali, M., Shirmohammadi, Z., Danish, M. S. S., & Khosravy, M. (2022). JCI-CAC: An Efficient Crosstalk Avoidance Code Considering Joint Capacitive and Inductive Effects. *IEEE Access*, *10*, 98348–98359. https://doi.org/10.1109/ACCESS.2022.3206039

[24] Kamel, A. R., & Abonazel, M. (2023). A Simple Introduction to Regression Modeling using R. *Computational Journal of Mathematical and Statistical Sciences*, *2*(1), 52–79. https://doi.org/10.21608/cjmss.2023.189834.1002

[25] Tahmasebi, M., & gohari, M. (2023). Design and Simulation of an Adaptive Neuro-controller for a Wire Driven Flexible Arm Robot. *Karafan Journal*, *20*(1), 243–262. https://doi.org/10.48301/kssa.2023.361179.2280

[26] Alipour, M., & Jafari, M. (2022). Estimating the Dynamic Margin of Voltage Stability in Power Systems Using Machine Learning. *Karafan Journal*, *19*(3), 221–245. https://doi.org/10.48301/kssa.2022.299963.1677

[27] Gangsar, P., Bajpei, A. R., & Porwal, R. (2022). A review on deep learning based condition monitoring and fault diagnosis of rotating machinery. *Noise & Vibration Worldwide*, *53*(11), 550–578. https://doi.org/10.1177/09574565221139638

[28] O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *ArXiv e-prints*. https://doi.org/10.48550/arXiv.1511.08458

[29] Purwono, I., Ma'arif, A., Rahmaniar, W., Imam, H., Fathurrahman, H. I. K., Frisky, A., & Haq, Q. M. U. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, *2*(4), 739–748. https://doi.org/10.31763/ijrcs.v2i4.888

[30] Pratiwi, H., Windarto, A. P., Susliansyah, S., Aria, R. R., Susilowati, S., Rahayu, L. K., Fitriani, Y., Merdekawati, A., & Rahadjeng, I. R. (2020). Sigmoid Activation Function in Selecting the Best Model of Artificial Neural Networks. *Journal of Physics: Conference Series*, *1471*(1), 012010. https://doi.org/10.1088/1742-6596/1471/1/012010

[31] Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, *11*(3), 53. https://doi.org/10.3390/computation11030052

[32] Shalev, M. (2007). *Limits and Alternatives to Multiple Regression in Comparative Research* (Vol. 24). Emerald Group Publishing Limited. https://doi.org/10.1016/s0195-6310(06)24006-7