دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

گزارش پروژه نهایی درس طراحی سیستم‌های قابل بازپیکربندی

طراحی و شبیه‌سازی شبکه عصبی CNN با هدف تشخیص ارقام دست‌نویس به‌وسیله HLS

نگارش

رضا آدینه پور

استاد درس

جناب آقای دکتر صاحب‌الزمانی

بهمن ۱۴۰۳


بسم الله الرحمن الرحیم

سپاس

از استاد گرانقدر خود، جناب آقای دکتر صاحب‌الزمانی، به خاطر ارائه‌های بی‌نظیرشان در طول ترم خالصانه تشکر و قدردانی می‌نمایم. همچنین از جناب آقای دکتر ملکوتی، تدریس‌یار محترم درس نیز به دلیل راهنمایی‌های بی‌نظیر و حمایت‌های بی‌دریغ ایشان در طول این پروژه، صمیمانه تشکر می‌نمایم. بازخوردها و کمک‌های سازنده ایشان نقش بسزایی در شکل‌گیری این پروژه داشته است.

چکیده

شبکه‌های عصبی پیچشی یکی از پرکاربردترین مدل‌ها در حوزه یادگیری عمیق هستند که در بسیاری از کاربردها مانند شناسایی تصاویر و پردازش داده‌های بصری مورد استفاده قرار می‌گیرند. با توجه به نیاز روزافزون به پردازش سریع و بهینه، استفاده از سخت‌افزارهایی مانند FPGA به دلیل قابلیت پردازش موازی و توان مصرفی پایین، گزینه‌ای ایده‌آل برای پیاده‌سازی این شبکه‌ها محسوب می‌شود.

در این پروژه، هدف پیاده‌سازی یک شبکه عصبی پیچشی برای شناسایی ارقام دست‌نویس بر روی FPGA با استفاده از روش سنتز سطح‌بالا است. فرآیند پیاده‌سازی شامل دو فاز اصلی بود: در فاز نرم‌افزاری، شبکه مورد نظر آموزش داده شد و وزن‌های آن ذخیره گردید. سپس در فاز سخت‌افزاری، وزن‌های ذخیره‌شده به FPGA منتقل شده و داده‌های ورودی به شبکه ارسال شدند. نتایج خروجی به منظور ارزیابی عملکرد و صحت شناسایی پردازش شدند. این پیاده‌سازی ترکیبی از کارایی بالا و انعطاف‌پذیری FPGA را با قدرت یادگیری عمیق ادغام کرده و امکان بهره‌وری بیشتر در کاربردهای عملی را فراهم می‌کند.

کلیدواژه‌ها: شبکه‌های عصبی، یادگیری عمیق، شبکه عصبی پیچشی، FPGA

فهرست مطالب

فهرست جداول

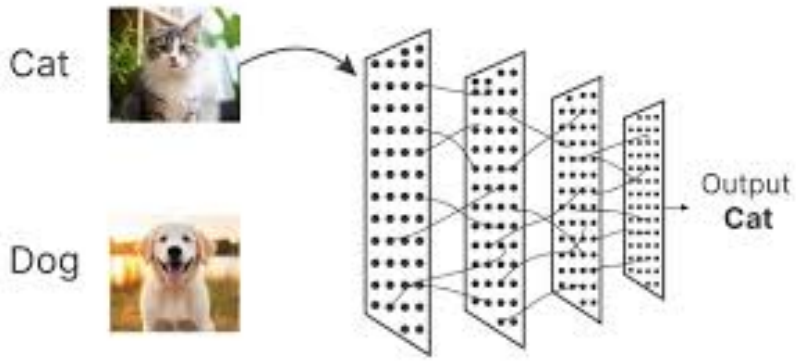
فهرست تصاویر

فصل ۱

مقدمه

۱-۱ تعریف مسئله

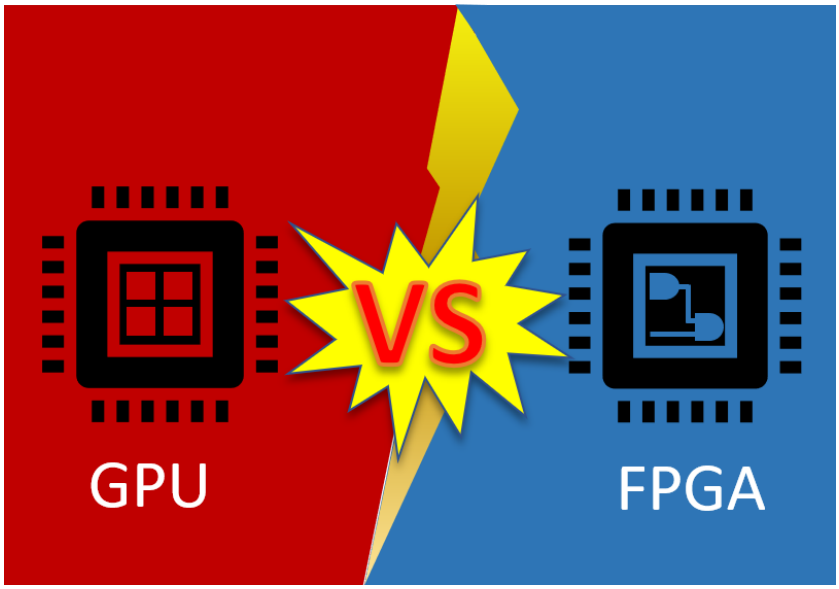
طبقه‌بندی[۱] یکی از مسائل اصلی در حوزه یادگیری ماشین[۲] است که هدف آن تخصیص ورودی‌ها به یکی از دسته‌های از پیش تعریف‌شده می‌باشد. شبکه‌های عصبی پیچشی[۳] (CNN) به دلیل توانایی بالای خود در استخراج ویژگی‌های سلسله‌مراتبی از داده‌های خام، در بسیاری از مسائل طبقه‌بندی، از جمله شناسایی تصاویر عملکرد بسیار خوبی داشته‌اند. مسئله طبقه‌بندی ارقام دست‌نویس به عنوان یک مسئله مرجع، نقش مهمی در نشان دادن توانایی شبکه‌های عصبی در پردازش داده‌های بصری دارد و به طور گسترده برای ارزیابی روش‌ها و مدل‌های مختلف استفاده می‌شود.



شکل ۱-۱: مسئله طبقه‌بندی [۱]

[1] Classification
[2] Machine Learning
[3] Convolutional Neural Network

با این حال، اجرای مدل‌های CNN در کاربردهای عملی چالش‌هایی مانند پیچیدگی محاسباتی بالا و نیاز به منابع سخت‌افزاری کارآمد را به همراه دارد. در حالی که GPUها به دلیل توان عملیاتی بالا گزینه‌ای مناسب برای آموزش و استنتاج[۴] مدل‌ها هستند، مصرف انرژی بالا و محدودیت‌های آن‌ها در کاربردهای نهفته[۵] و محیط‌هایی با منابع محدود، آن‌ها را برای برخی کاربردها نامناسب می‌سازد. در مقابل، FPGAها با قابلیت پردازش موازی، مصرف انرژی کمتر و قابلیت بازپیکربندی[۶]، گزینه‌ای ایده‌آل برای پیاده‌سازی مدل‌های CNN در کاربردهایی هستند که نیاز به پردازش بی‌درنگ[۷] و بهره‌وری بالا[۸] دارند.



شکل ۱-۲: مسئله طبقه‌بندی [۲]

۱-۲ اهمیت پژوهش

پیاده‌سازی شبکه‌های عصبی پیچشی بر روی FPGA نه تنها به دلیل چالش‌های فنی موجود در ترکیب یادگیری عمیق با سخت‌افزارهای نهفته اهمیت دارد، بلکه از جنبه‌های کاربردی نیز تأثیر بسزایی دارد. این پروژه امکان استفاده از مدل‌های یادگیری عمیق در محیط‌هایی با منابع محدود و نیاز به مصرف انرژی کم، مانند دستگاه‌های IoT، سیستم‌های صنعتی بی‌درنگ و تجهیزات پزشکی قابل حمل[۹] را فراهم می‌کند. علاوه بر این، FPGAها به دلیل انعطاف‌پذیری در طراحی و تطبیق‌پذیری با کاربردهای متنوع، می‌توانند بستری مناسب برای توسعه سامانه‌های هوشمند با کارایی بالا باشند. نتایج این پژوهش می‌تواند راهگشای کوچکی برای پژوهشگران و

[4] Inference
[5] Embedded
[6] Reconfigurability
[7] Real-Time
[8] High Performance
[9] Portable

مهندسان در کاهش هزینه‌های طراحی، بهبود سرعت پردازش و افزایش بهره‌وری سیستم‌های مبتنی بر یادگیری عمیق باشد.

۱-۳ اهداف پژوهش

هدف اصلی این پژوهش، شتاب‌دهی سخت‌افزاری فاز استنتاج[۱۰] شبکه‌های عصبی پیچشی با بهینه‌سازی مصرف توان و انرژی است. با توجه به نیاز روزافزون به پردازش سریع و کارآمد داده‌ها در کاربردهای بی‌درنگ و نهفته، استفاده از FPGA به عنوان بستری مناسب برای تحقق این هدف در اولویت قرار گرفته است. این پروژه به دنبال دستیابی به معماری سخت‌افزاری است که علاوه بر ارائه سرعت بالا در پردازش، مصرف انرژی را به حداقل برساند و قابلیت پیاده‌سازی در محیط‌های محدود به منابع مانند سیستم‌های IoT، دستگاه‌های قابل حمل و کاربردهای صنعتی را فراهم کند.

۱-۴ ساختار پژوهش

این پژوهش در ۴ فصل انجام شده است. در فصل ۱ به مقدمه و اهمیت موضوع پژوهش پرداخته شده است. در فصل ۲ به مفاهیم اولیه و پیش‌نیاز ها پرداخته شده است. در ادامه در فصل ؟؟ پژوهش به بررسی کارهای پیشین انجام شده در این زمینه پرداخت شده است. و در فصل پایانی، جمع‌بندی و نتیجه گیری پژوهش ارائه شده است.

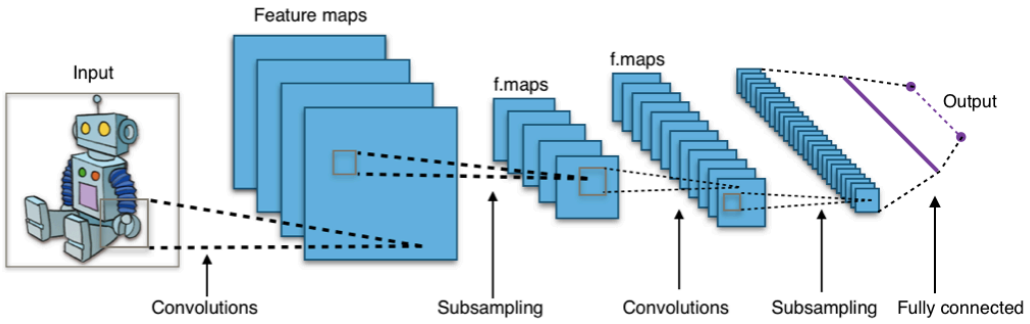
[10] Inference

فصل ۲

مفاهیم اولیه

۲-۱ شبکه عصبی CNN

شبکه‌های عصبی پیچشی (Convolutional Neural Networks یا CNN) نوعی از شبکه‌های عصبی مصنوعی هستند که به طور خاص برای پردازش داده‌های با ساختار شبکه‌ای مانند تصاویر طراحی شده‌اند. این شبکه‌ها به دلیل توانایی بالای خود در شناسایی الگوها و ویژگی‌ها، به طور گسترده در مسائلی مانند طبقه‌بندی تصاویر [۱] و تشخیص اشیاء[۲] استفاده می‌شوند. CNNها از معماری سلسله‌مراتبی[۳] برای استخراج ویژگی‌ها از داده‌های ورودی استفاده می‌کنند و قادرند ویژگی‌های سطح پایین (مانند لبه‌ها) تا ویژگی‌های سطح بالا (مانند اشکال پیچیده) را به طور خودکار شناسایی کنند.



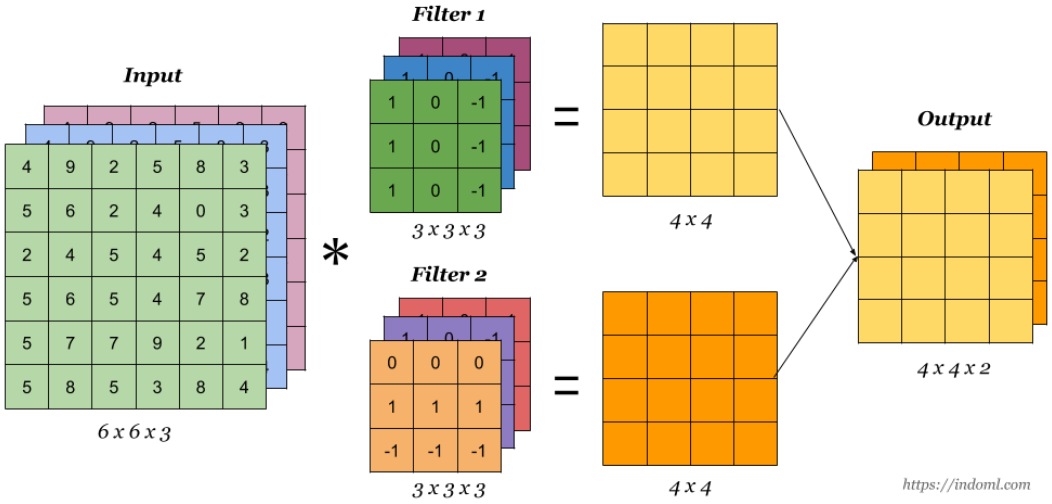
شکل ۲-۱: ساختار یک شبکه CNN [۳]

[1] Image Classification
[2] Object Detection
[3] Hierarchical

۲-۲ اجزای اصلی شبکه CNN

۲-۲-۱ لایه کانولوشن:

این لایه وظیفه استخراج ویژگی‌ها از داده‌های ورودی را بر عهده دارد. در این لایه، یک یا چند فیلتر[۴] کوچک بر روی داده‌های ورودی حرکت کرده و عملیات کانولوشن را انجام می‌دهند. نتیجه این عملیات ویژگی‌های مکانی[۵] است که اطلاعات مهم را حفظ کرده و داده‌های غیرضروری را حذف می‌کند.



شکل ۲-۲: عملیات کانولوشن [۴]

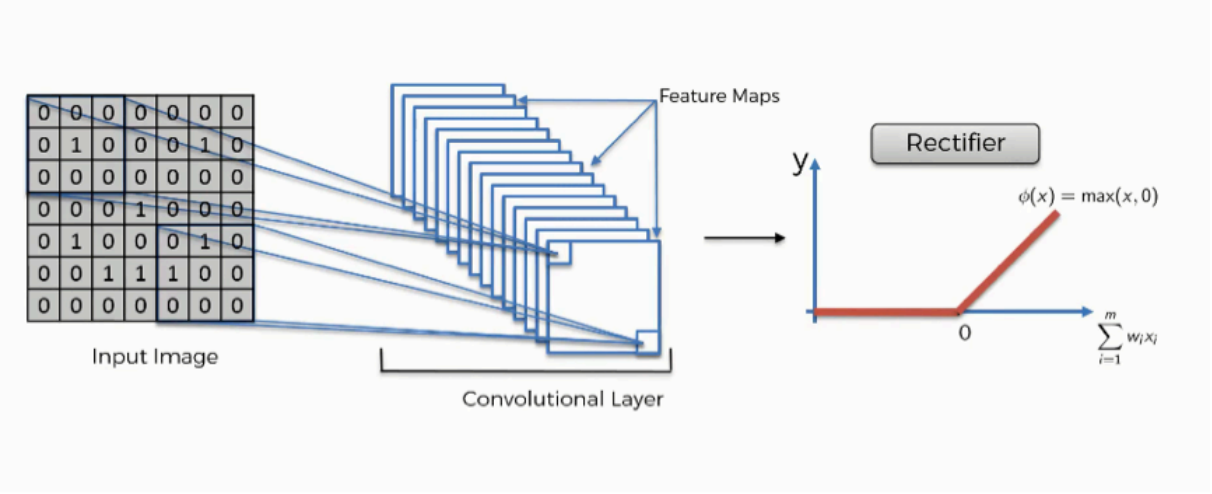
۲-۲-۲ لایه فعال‌سازی:

پس از هر لایه کانولوشن، از یک تابع فعال‌ساز[۶] غیرخطی (مانند ReLU)[۷] استفاده می‌شود. این تابع باعث می‌شود مدل بتواند روابط پیچیده و غیرخطی را یاد بگیرد. تابع ReLU معمولاً بیشترین استفاده را دارد و با نگه‌داشتن مقادیر مثبت و صفر کردن مقادیر منفی، سرعت و کارایی مدل را افزایش می‌دهد.

۲-۲-۳ لایه تجمیع:

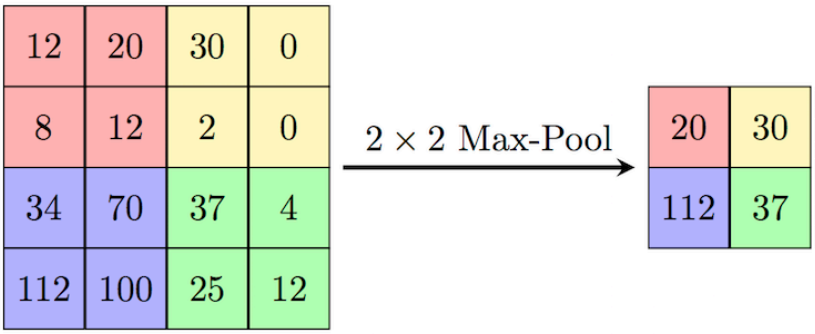
لایه تجمیع[۸] وظیفه کاهش ابعاد داده‌ها را دارد تا تعداد پارامترها و پیچیدگی محاسباتی کاهش یابد. معمولاً از روش Max Pooling استفاده می‌شود، که در آن بزرگ‌ترین مقدار در هر ناحیه انتخاب می‌شود. این فرآیند

[4] Kernel
[5] Spatial Features
[6] Activation Function
[7] Rectified Linear Unit
[8] Pooling



شکل ۲-۳: تابع فعال‌ساز ReLU [۵]

باعث افزایش مقاومت مدل در برابر تغییرات جزئی در داده‌های ورودی (مانند انتقال یا چرخش) می‌شود.



شکل ۲-۴: لایه Max_pooling [۶]

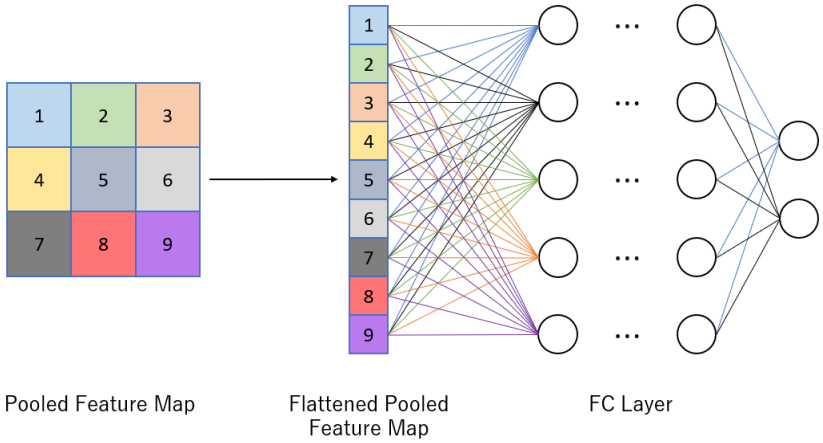
۲-۲-۴ لایه تمام‌متصل:

لایه تمام‌متصل[۹]، ویژگی‌های استخراج‌شده توسط لایه‌های قبلی را به صورت یک بردار مسطح درمی‌آیند و به نرون‌های خروجی متصل می‌شوند. این لایه وظیفه تصمیم‌گیری نهایی (مانند طبقه‌بندی) را بر عهده دارد.

۲-۲-۵ لایه خروجی:

این لایه از یک تابع فعال‌سازی مانند Softmax یا Sigmoid برای تولید خروجی استفاده می‌کند. خروجی این لایه معمولاً احتمال تعلق ورودی به هر کلاس در مسئله طبقه‌بندی است.

[9] Fully Connected



شکل ۲-۵: لایه تمام‌متصل [۷]

۲-۳ نحوه عملکرد شبکه CNN

CNNها با دریافت داده‌های ورودی (مانند تصاویر)، آن‌ها را از طریق لایه‌های مختلف عبور داده و ویژگی‌های مهم را مرحله به مرحله استخراج می‌کنند. در هر مرحله، ویژگی‌ها پیچیده‌تر و خاص‌تر می‌شوند. لایه‌های کانولوشنی ویژگی‌ها را استخراج می‌کنند، لایه‌های تجمیع ابعاد داده‌ها را کاهش می‌دهند و در نهایت، لایه‌های تمام‌متصل و خروجی تصمیم‌گیری نهایی را انجام می‌دهند. این معماری به CNNها اجازه می‌دهد تا در کاربردهایی مانند شناسایی چهره، تشخیص اشیاء و تحلیل تصاویر پزشکی عملکردی بسیار دقیق و مؤثر داشته باشند.

۲-۴ ساختار FPGA

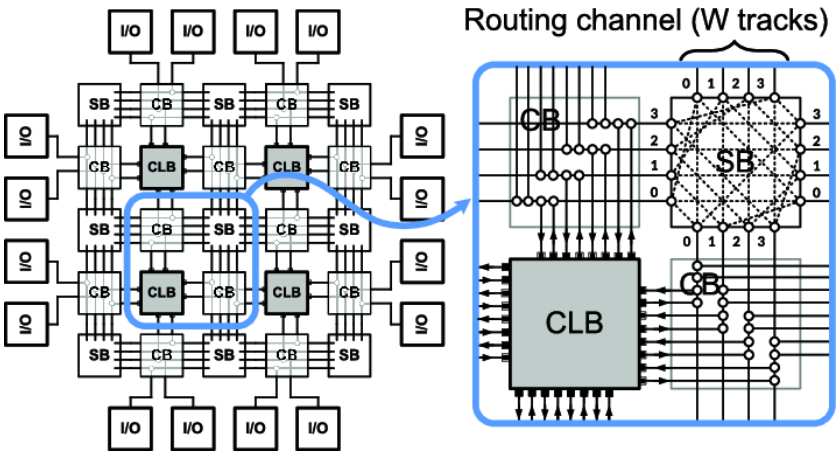
FPGA[۱۰] یک تراشه سخت‌افزاری قابل‌برنامه‌ریزی است که به کاربران اجازه می‌دهد ساختار داخلی آن را پس از تولید تغییر دهند و برای کاربردهای خاص طراحی کنند. این قابلیت بازپیکربندی[۱۱] یکی از ویژگی‌های کلیدی FPGA است که امکان اجرای مجموعه‌ای از عملکردهای منطقی و موازی را با انعطاف‌پذیری بالا فراهم می‌کند. FPGAها از ساختارهایی شامل بلوک‌های منطقی قابل‌پیکربندی (CLB)[۱۲]، منابع اتصالات قابل‌برنامه‌ریزی، و منابع ورودی/خروجی تشکیل شده‌اند که با یکدیگر کار می‌کنند تا مدارهای دلخواه را پیاده‌سازی کنند.

[10] Field-Programmable Gate Array
[11] Reconfigurable
[12] Combinational Logic Block

۲-۵ اجزای مهم FPGA

۲-۵-۱ بلوک‌های منطقی قابل‌پیکربندی (CLB):

این بلوک‌ها هسته اصلی FPGA هستند و از ترکیب LUT، فلیپ‌فلاپ‌ها، و عناصر منطقی تشکیل شده‌اند. LUTها امکان پیاده‌سازی توابع منطقی را فراهم می‌کنند و فلیپ‌فلاپ‌ها برای ذخیره مقادیر استفاده می‌شوند. با استفاده از CLBها، می‌توان انواع گیت‌های منطقی و توابع پیچیده‌تر را پیاده‌سازی کرد.



شکل ۲-۶: CLB ها در FPGA

۲-۵-۲ منابع اتصالات (Routing Resources):

FPGA دارای شبکه‌ای از مسیرهای قابل‌برنامه‌ریزی است که بلوک‌های منطقی را به یکدیگر و به پورت‌های ورودی/خروجی متصل می‌کند. این منابع شامل سوئیچ‌ها و ماتریس‌های اتصالات است که امکان تنظیم مسیرهای داده در FPGA را فراهم می‌کنند.

۲-۵-۳ بلوک‌های ورودی/خروجی (I/O Blocks):

این بلوک‌ها مسئول ارتباط FPGA با دنیای خارجی هستند و امکان تبادل داده با سایر دستگاه‌ها را فراهم می‌کنند. I/Oها برای پشتیبانی از پروتکل‌های مختلف ارتباطی قابل‌پیکربندی هستند.

۴-۵-۲ حافظه‌های داخلی:

FPGAها شامل حافظه‌هایی مانند RAM بلوکی[۱۳] و حافظه‌های توزیع‌شده[۱۴] هستند که برای ذخیره داده‌ها و متغیرها در طول اجرای عملیات استفاده می‌شوند.

۵-۵-۲ واحدهای DSP:

اکثر FPGAهای مدرن دارای واحدهای پردازش سیگنال دیجیتال[۱۵] هستند که برای عملیات ریاضی پیچیده مانند ضرب و جمع بهینه‌سازی شده‌اند. این واحدها نقش مهمی در کاربردهایی مانند پردازش سیگنال و یادگیری عمیق ایفا می‌کنند.

۶-۲ قابلیت بازپیکربندی FPGA:

FPGAها به کاربران اجازه می‌دهند مدار داخلی خود را با استفاده از ابزارهای سنتز سخت‌افزاری مانند Verilog، VHDL، یا HLS تغییر دهند. این قابلیت به معنای انعطاف‌پذیری بالا برای تغییر یا بهبود طراحی است، حتی پس از ساخت سخت‌افزار. علاوه بر این، برخی از FPGAها از بازپیکربندی پویا[۱۶] پشتیبانی می‌کنند که امکان تغییر بخشی از طراحی را در زمان اجرا بدون اختلال در عملکرد سایر بخش‌ها فراهم می‌کند.

۷-۲ ابزار HLS

Xilinx Vitis HLS (High-Level Synthesis) یک پلتفرم توسعه نرم‌افزاری است که به مهندسان این امکان را می‌دهد تا کدهای نرم‌افزاری نوشته شده در زبان‌های سطح بالا مانند C، C++ یا OpenCL را به سخت‌افزار FPGA تبدیل کنند. هدف اصلی این ابزار تسهیل فرآیند طراحی سخت‌افزار است، به‌گونه‌ای که توسعه‌دهندگان نیازی به درک عمیق از جزئیات معماری FPGA نداشته باشند. با استفاده از Vitis HLS، طراحی‌های سخت‌افزاری به طور خودکار از کدهای سطح بالا استخراج شده و به طراحی‌های RTL[۱۷] که قابل‌سنتز در FPGA هستند، تبدیل می‌شوند.

[13] BRAM
[14] Distributed RAM
[15] DSP
[16] Dynamic Reconfiguration
[17] Register-Transfer Level

۱-۷-۲ مزایای استفاده از Xilinx Vitis HLS

- کاهش زمان توسعه: استفاده از زبان‌های سطح بالا برای نوشتن کد، به طراحان این امکان را می‌دهد که زمان بیشتری برای الگوریتم‌ها و منطق طراحی صرف کنند و به‌جای پرداختن به جزئیات معماری FPGA، تمرکز بیشتری بر روی ویژگی‌های عملکردی داشته باشند.
- ارتقاء عملکرد: ابزار Vitis HLS این امکان را فراهم می‌کند که طراحی‌های سخت‌افزاری بهینه‌سازی شوند، به‌ویژه در زمینه‌هایی مانند پردازش موازی، سرعت بالا و کارایی انرژی.
- سادگی پیاده‌سازی: بدون نیاز به دانش تخصصی در زمینه زبان‌های سخت‌افزاری مانند VHDL یا Verilog، مهندسان می‌توانند به راحتی از C/C++ یا OpenCL برای ایجاد مدارهای پیچیده استفاده کنند.

Bibliography

[1] A. Vidhya. Beginner-friendly project: Cat and dog classification using cnn, 2021. Accessed: 2025-01-23.

[2] InAccel. Gpus vs fpgas: Which one is better in dl and data centers applications?, 2020. Accessed: 2025-01-23.

[3] W. contributors. Convolutional neural network. https://en.wikipedia.org/wiki/Convolutional_neural_network, 2025. Accessed: 2025-01-23.

[4] IndoML. Student notes: Convolutional neural networks (cnn) introduction, 2018. Accessed: 2025-01-23.

[5] SuperDataScience. Convolutional neural networks (cnn) step 1b: Relu layer, n.d. Accessed: 2025-01-23.

[6] P. with Code. Max pooling, n.d. Accessed: 2025-01-23.

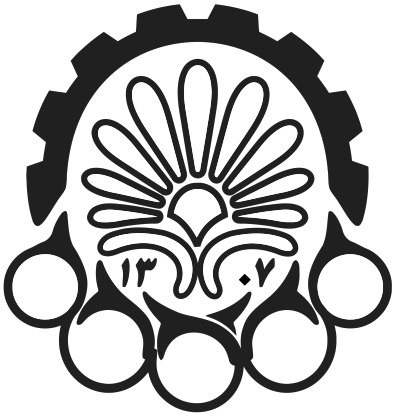
[7] A. I. Aramendia. Convolutional neural networks (cnns): A complete guide, n.d. Accessed: 2025-01-23.

Abstract

Convolutional Neural Networks (CNNs) are among the most widely used models in the field of deep learning, particularly in applications such as image recognition and visual data processing. Given the growing demand for fast and efficient processing, hardware platforms like FPGA have become an ideal choice for implementing these networks due to their parallel processing capabilities and low power consumption.

In this project, the goal was to implement a Convolutional Neural Network for handwritten digit recognition on an FPGA using High-Level Synthesis (HLS). The implementation process consisted of two main phases: In the software phase, the network was trained, and its weights were stored. In the hardware phase, the stored weights were transferred to the FPGA, and the input data was fed into the network. The outputs were then processed to evaluate the performance and accuracy of recognition. This implementation combines the high efficiency and flexibility of FPGA with the power of deep learning, enabling enhanced productivity in practical applications.

Keywords: Neural Networks, Deep Learning, CNN, FPGA



Amirkabir University of Technology

(Tehran Polytechnic)

Department of Computer Engineering

Reconfigurable Systems Design Final Project Report

Design and Simulation of CNN Neural Network for Hand Written Digit Recognition Using HLS

By:

Reza Adinepour

Supervisor:

Prof. Saheb Zamani

Jan 2025