پیاده سازی تقریبی شبکه های عصبی کپسولی

دانشجو: روزبه بستاندوست

استاد راهنما: دکتر خونساری

دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران

مقدمه / خلاصه

ایده اصلی شبکه کپسولی هنگامی مطرح شد که ناتوانی شبکه های عصبی کانولوشنی در رابطه با در نظر گرفتن تناسب محل قرار گرفتن اجزای تصویر برای تشخیص تصویر، ثابت شد. یکی از خصوصیات شبکه های کپسولی،استفاده از محاسبات برداری به جای محاسبات عددی می‌باشد که باعث بالا رفتن حجم محاسبات می‌شود. در این پروژه، روش های تقریبی ارائه شد که سربار محاسبات برداری کمتر شود تا پیاده سازی این شبکه(بدون تغییر دقت) بر روی نرم افزار و سخت افزار آسان‌تر و از لحاظ مصرف منابع و مساحت(برای پیاده سازی سخت افزاری) بهینه‌تر گردد.

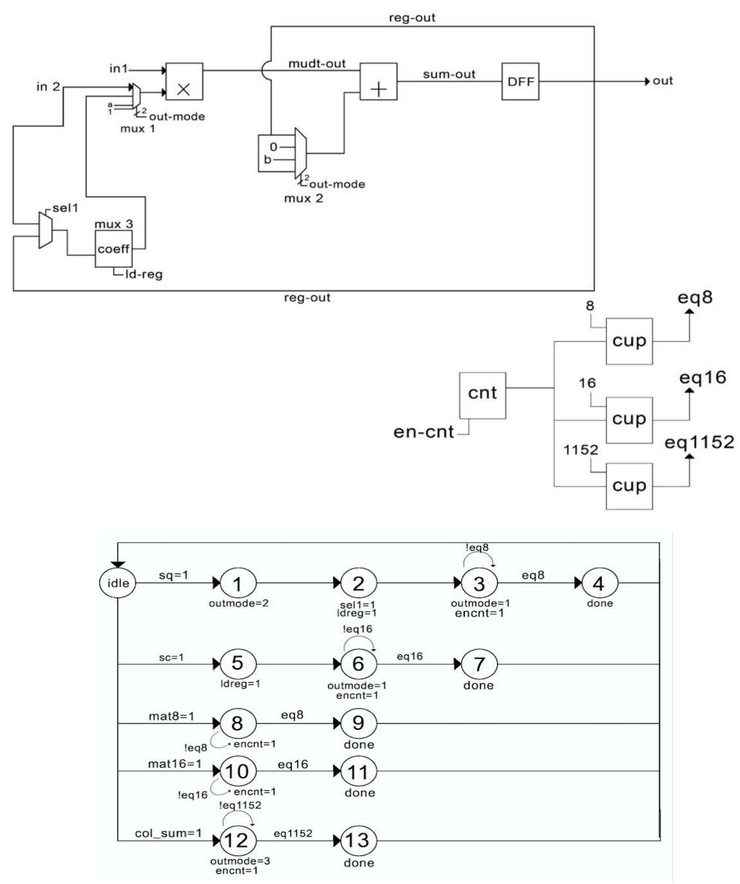
- **نوع پروژه: پیاده سازی**
- **اهداف پروژه: بهینه سازی و تقریب زدن در محاسبات برداری شبکه کپسولی برای کاهش هزینه پیاده سازی نرم افزاری و سخت افزاری**
- **سوال اصلی: چگونه می‌توان از روش های تقریب زدن بدون کاهش دقت شبکه استفاده کرد؟ آیا قسمت محاسباتی شبکه کپسولی را می‌توان بهشنه کرد و بر روی سخت افزار پیاده سازی کرد؟**
- **روش پاسخ به سوال: مطالعه روش های ریاضی استفاد شده و خوانده مقالات مربوط و آزمودن روش های مختلف و انتخاب بهترین روش با توجه به دقت شبکه و محدودیت های سخت افزاری و نرم افزاری**
- **دستاورد ها و خروجی ها: کاهش توان و مساحت سخت افزاری واحد محاسباتی شیکه و هم چنین کاهش زمان اجرای نرم افزاری شبکه با حفظ دقت شبکه**

روش/ساختار/مدل پیشنهادی

دو تابع محاسباتی که پیاده سازی آن ها از لحاظ سخت افزاری پیچیده است به نام های squash(خروجی تابع یک بردار است) و softmax(خروجی تابع یک عدد است) وجود دارند که فرمولsquash را در سمت چپ و فرمول softmax را در سمت راست مشاهده می‌کنید.

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1+||\mathbf{s}_j||^2}\frac{\mathbf{s}_j}{||\mathbf{s}_j||} \qquad c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$$

تابع squash، یک مقدار عددی را که در محاسبه آن از اندازه بردار استفاده می‌شود، در بردار ضرب می‌کند. برای تقریب زدن آن مقدار عددی، در ابتدا حدود ۱۱۵۲۰۰ نمونه از محاسبه تابع squash در هنگام آموزش شبکه جمع آوری شد و با روش رگرسیون خطی، ضرایب a و b در فرمول y = ax+b محاسبه شد که x اولین مولفه از بردار و y همان مقدار عددی است که باید محاسبه شود تا در کل بردار ضرب شود.
تابع softmax، در محاسبات خود از تابع exp استفاده می‌کند که هزینه بالایی در پیاده سازی سخت افزاری دارد. روش انجام شده برای بهینه سازی این تابع به این صورت است که در هنگام آموزش شبکه ، مقدار کمینه و بیشینه ورودی این تابع اندازه گیری می‌شود و به ۲۵۶ بازه تقسیم می‌شود و مقدار هر بازه مقدار خروجی تابع softmax به ازای ورودی عدد وسط هر بازه تعیین می‌شود و در حقیقت محاسبه دقیق تابع با LUT جایگزین شده است.
با توجه به بهینه سازی های انجام شده، واحد محاسباتی شبکه کپسولی به نحوی طراحی شده است که علاوه بر محاسبه توابع محاسباتی فوق، ضرب داخلی و خارجی دو ماتریس با تعداد ستون های ۸ و ۱۶ را نیز انجام می دهد.



شکل۱-مسیر داده و کنترلر واحد محاسباتی شبکه کپسولی

نتایج

نتایج نرم افزاری:

شبکه عصبی بهینه شده با مجموعه داده MNIST و CIFAR10 آزمایش شده است، که نتایج آن در ذیل گزارش شده است و مشاهده شد که دقت شبکه تغییر یافته با شبکه اصلی تفاوتی نمی‌کند.

جدول ۱ - جدول دقت شبکه

	دقت
MNIST	99.5%
CIFAR10	85%

نتایج سخت افزاری:

مسیر داده و کنترلر طراحی شده را با دو ابزار Design Compiler(برای سنتز ASIC) و ISE Design Suite(برای سنتز بر روی FPGA) سنتز شده است و نتایج در ذیل گزارش شده است و مشاهده شد که توان حدود ۶۰٪ و مساحت حدود ۷۰٪ کاهش یافته است.

Hierarchy	Switch Power	Int Power	Leak Power	Total Power
Top_Module	5.66e-03	2.25e-02	5.83e+05	2.87e-02

Hierarchical cell	Global cell area: Absolute Total	Global cell area: Percent Total	Local cell area: Combi-national	Local cell area: Noncombi-national	Local cell area: Black boxes	Design
Top_Module	56.3920	100.0	0.0000	0.0000	0.0000	Top_Module
cu	56.3920	100.0	37.2400	19.1520	0.0000	Controller
Total			37.2400	19.1520	0.0000	

Analysis & Synthesis Status	Successful - Sun Sep 08 13:27:31 2019
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	MAC_Synthesis
Top-level Entity Name	Top_Module
Family	Cyclone IV GX
Total logic elements	489
Total combinational functions	488
Dedicated logic registers	73
Total registers	73
Total pins	72

شکل۲-نتایج بدست آمده از DC و ISE Design Suite

جمع بندی

بهینه سازی های انجام شده بر روی شبکه کپسولی، از لحاظ سخت افزاری باعث کاهش مساحت و توان و از لحاظ نرم افزاری باعث کاهش زمان اجرا شده است در حالی که دقت شبکه تغییری نکرده است. هم‌چنین تا به حال واحد محاسباتی شبکه کپسولی به صورت سخت افزاری پیاده سازی نشده است و می‌توان از این پروژه در واحد های تشخیص تصاویر که نیاز به سرعت و دقت بالا دارند استفاده کرد.

مراجع اصلی

[1] S. Sabour, N. Frosst and G. Hinton, "Dynamic Routing Between Capsules", *arXiv.org*, 2019.
[2] G. E. Hinton, A. Krizhevsky, and S. D. Wang, “Transforming Auto-Encoders,” *Lecture Notes in Computer Science Artificial Neural Networks and Machine Learning – ICANN 2011*, pp. 44–51, 2011
[3] B. Gagana, H. A. U. Athri, and S. Natarajan, “Activation Function Optimizations for Capsule Networks,” *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018.

شماره پوستر: 68