International Scientific Conference 20 UNITECH 26

INTERNATIONAL SCIENTIFIC CONFERENCE
19-21 November 2026
GABROVO

МАТЕМАТИЧЕН МОДЕЛ И МЕТОД ЗА ОПТИМИЗАЦИЯ НА ЕНЕРГИЙНАТА ЕФЕКТИВНОСТ ПРИ НЕВРОМОРФНИ EDGE AI АРХИТЕКТУРИ

Мирослав Дянков[1], Горан Горанов[2]

[1]Технически университет – Габрово, катедра „Електроника“, Габрово, България
[2]Технически университет – Габрово, катедра „Електроника“, Габрово, България
**кореспондиращ автор: e-mail Miroslavdyankovmor@gmail.com*

A MATHEMATICAL MODEL AND METHOD FOR ENERGY-EFFICIENCY OPTIMIZATION OF NEUROMORPHIC EDGE AI ARCHITECTURES

Miroslav Dyankov[1], Goran Goranov[2]

Abstract
The paper starts from a practical question: what actually consumes the energy during one SNN inference? A single power value for the chip does not answer this well. Spike activity, fan-out, memory accesses, communication between blocks and the baseline consumption all take part. We therefore use a hardware-oriented model in which these terms are kept separate. The activity counters can come from an SNN simulator, from counters in an RTL/FPGA prototype or later from EDA analysis. The same model is then used as an objective in an optimization problem with limits on accuracy, latency and memory. A communication-related regularization term is also included during training. The model is not treated as a substitute for measurement. Its purpose is to give a clear structure for the measurements and to make later FPGA and ASIC results easier to compare.

Keywords*:* neuromorphic computing, Edge AI, spiking neural networks, energy model, hardware-aware optimization, memory locality, FPGA, low-power design.

ВЪВЕДЕНИЕ

При Edge AI бързодействието е само част от задачата. В малък сензорен възел, носимо устройство или автономен модул обикновено има твърда граница за консумацията и паметта. Има и още един проблем - движението на данни. При много цифрови архитектури прехвърлянето между паметта и изчислителните блокове струва повече енергия от част от самите аритметични операции [2,3]. Във невроморфните архитектури е по интересно, защото се съхранява локално информацията близо до мястото в което се обработват и извършват изчисления основно при настъпване на събития. Така от локалното съхранение на данни се пести енергия от техния пренос и от самото активиране на импулс и не е нужно всички блокове да работят. [1,4,5].

SNN не означава автоматично ниска консумация. Ако кодиращата схема генерира твърде много импулси, ако fan-out е голям или ако теглата постоянно се четат от далечна памет, очакваното предимство може да изчезне [6-10]. Базовата консумация на платформата също остава. Затова число в mW, дадено без конкретна задача и време за инференция, не е достатъчно за сравнение.

Енергията на една инференция се разделя на компоненти, които могат да се оценяват по отделно синаптична обработка, импулси, памет, комуникация и

базов разход. Моделът е приложим последователно при софтуерна симулация, при FPGA прототип с хардуерни броячи и при EDA анализ за конкретна CMOS библиотека. По този начин архитектурните промени могат да бъдат оценени още преди преминаване към ASIC реализация.

СЪСТОЯНИЕ НА ПРОБЛЕМА

Horowitz систематизира относителната цена на аритметичните операции и достъпа до памет и показва, че преносът на данни може да доминира над самото изчисление [3]. Този резултат е особено важен за Edge AI, където сравнително малък брой операции умножение и натрупване (MAC) или синаптично натрупване могат да бъдат придружени от голям брой паметни трансфери. Sze et al. също определят движението на данни като ключов енергиен фактор при невронните ускорители и аргументират необходимостта от съвместно проектиране на алгоритъма и паметната йерархия [2].

Horowitz [3] показва, че паметният достъп може да изисква повече енергия от самата аритметична операция. При Edge AI този ефект се усилва, тъй като дори сравнително малка мрежа може да генерира многократни четения и записи. Поради това броят MAC или SynOp операции не е достатъчен за пълна енергийна оценка.

При невронните ускорители този въпрос се разглежда чрез паметната йерархия и движението на данни [2]. За невроморфен хардуер Ostrau et al. [7] използват параметричен енергиен модел, а Blouw et al. [6] сравняват енергия, латентност и точност върху конкретна задача. И двата подхода показват необходимостта алгоритъмът и хардуерът да се оценяват съвместно.

Резултатите в [9] и [10] показват, че промяната на импулсната активност изменя броя цикли, паметния трафик и енергията. Асинхронната обработка дава предимство при разреден вход, докато при висока активност ефектът намалява. FPGA обзорите показват същата зависимост: една и съща SNN може да има различна хардуерна цена според разрядността, времевия прозорец и картографирането [11,12].

При енергийното оценяване обикновено се използват два крайни подхода. Единият прилага средна енергия на операция и е подходящ за ранно сравнение, но не отчита подробно паметта и комуникацията. Другият използва директно измерване върху конкретен чип и дава по-точна оценка за съответната платформа, но резултатът трудно се пренася към друга реализация. Междинният подход запазва една и съща структура на модела, а коефициентите се определят отделно за всяка платформа.

От гледна точка на научния проблем остава празнина между два типа оценки. Първият тип използва общи стойности за енергия на операция и е удобен за ранно алгоритмично сравнение, но често не отчита реалното преместване на данни. Вторият тип използва измервания върху конкретен чип и е точен за дадената платформа, но трудно се пренася към нова архитектура. Предложеният модел цели междинно ниво: да бъде достатъчно подробен, за да отчита памет и комуникация, но параметричен, така че да може постепенно да се калибрира при преминаване от софтуер към FPGA и ASIC.

ПРЕДЛОЖЕН МАТЕМАТИЧЕН МОДЕЛ

Под „синаптична операция“ (SynOp) разбираме натрупване, задействано от входен импулс. Хардуерът може да го реализира по различен начин, но за модела ни трябва еднакъв брояч. Ако S_l е броят изходни импулси от слой l, а F_l е средният ефективен fan-out след картографирането, броят активирани синаптични операции се определя с:

$$N_{syn} = \Sigma_l S_l \cdot F_l \qquad (1)$$

За една инференция са нужни няколко разхода на енергия. e_{syn} е средната

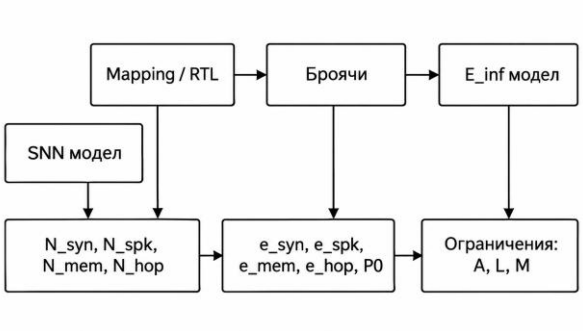
енергия за активирана SynOp, e_spk е допълнителният разход за генериране и обработка на импулс, e_mem е цената на паметен достъп, а e_hop - на една маршрутна стъпка. P_0 е базовата мощност на активната конфигурация. N_mem, N_hop и t_inf се измерват за конкретното изпълнение, а E_IO включва входното кодиране и изходния интерфейс.

$$E_inf = e_syn \cdot N_syn + e_spk \cdot N_spk + e_mem \cdot N_mem + e_hop \cdot N_hop + P_0 \cdot t_inf + E_IO \quad (2)$$

Формата на модела позволява N-величините да се получат от симулатор или хардуерни броячи, а e-коефициентите да се калибрират отделно за конкретната реализация. При FPGA те могат да се оценят чрез измерване на мощността при контролирани микробенчмаркове, а при бъдещ ASIC – чрез power analysis след синтез и физическо разполагане. Така алгоритмичната активност и технологичният процес остават разделени в модела.

N-величините характеризират извършената работа, а e-величините задават нейната енергийна цена върху конкретния хардуер. Броячите могат да бъдат получени от симулатор или директно от FPGA, докато енергийните коефициенти се калибрират отделно. При промяна на технологичния процес структурата на модела се запазва и се определя нов набор от коефициенти.

Ако локалната и отдалечената памет имат ясно различна изразходвана енергия, паметният член се разделя: $E_mem = e_LN_L + e_RN_R$. N_L и N_R са броят локални и отдалечени достъпи. При нормална микроархитектура e_R е по-голям, защото включва допълнителен пренос. По същия начин N_hop може да брои само междуядрената комуникация, а локалната да остане в e_syn. Детайлността не е фиксирана; тя се избира според реалния хардуер.



Фиг. 1. *Структура на предложения модел и източници за калибриране на параметрите*

МОДЕЛ НА РАЗРЕДЕНОСТТА И ЛОКАЛНОСТТА

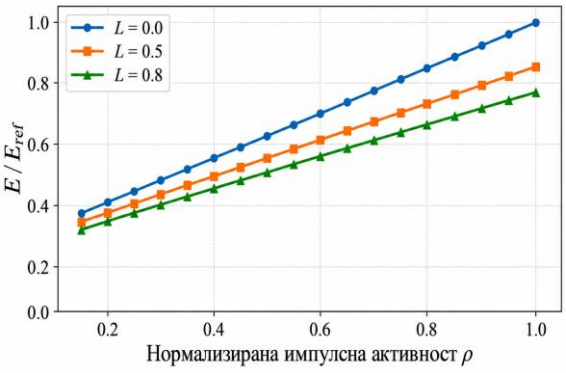
За първоначалния анализ се използват две управляеми величини: импулсната активност и локалността. С $\rho \in (0,1]$ се означава активността спрямо базова конфигурация. При една и съща топология N_syn и N_spk приблизително следват ρ, както и част от паметните и комуникационните събития. С $L \in [0,1]$ се задава делът на работата, който остава локално. L=0 съответства на базовото картографиране, а по-голямо L означава по-малко отдалечени достъпи и по-малко междуядрени прехвърляния.

$$E(\rho,L) = \rho K_c + \rho(1 - \alpha L) K_m + \rho(1 - \beta L) K_n + E_b \quad (3)$$

В уравнение (3) K_c представя събитийните изчисления, K_m – паметните разходи, K_n – комуникацията, а E_b е компонентът, който остава при много малка импулсна активност. Параметрите α и β определят степента, в която подобреното картографиране намалява паметните и комуникационните разходи. При голяма стойност на E_b намаляването на броя импулси не води до пропорционално намаляване на общата енергия.

При фиксирани t_inf и периферни разходи, когато ρ клони към нула, остава E_b. Производната по ρ определя изменението на енергията при допълнителна импулсна активност за зададена локалност. Смесената производна $\partial^2 E/(\partial\rho\partial L) = -(\alpha K_m + \beta K_n) < 0$ показва взаимното влияние между двата параметъра. При по-висока локалност енергий-

ната цена на допълнителния импулс намалява, което обосновава съвместното настройване на ρ и L.



Фиг. 2. *Чувствителност на енергия към активността ρ и локалността L.*

МНОГОКРИТЕРИАЛНА ОПТИМИЗАЦИЯ

Минимизирането на енергията не е достатъчно като единствен критерий. Edge AI системата трябва едновременно да запази необходимата точност, допустимата латентност и ограниченията на наличната памет. Оптимизационната задача се задава с A_min, L_max и M_max. Конфигурация с близка енергия, но със значителна загуба на точност, не се приема за по-добро решение.

$$\min_\theta E_{inf}(\theta), \quad \text{при } A(\theta) \geq A_{min}, \; t_{inf}(\theta) \leq L_{max}, \; M(\theta) \leq M_{max} \qquad (4)$$

При наличие на противоречиви критерии се използва множество на Парето вместо предварително зададени тегловни коефициенти. Така енергийното намаление може да се сравнява пряко със загубата на точност или увеличението на латентността. Подходът позволява съпоставка между няколко архитектурни варианта без предварително фиксиране на относителната тежест на критериите.

Оптимизацията може да започне още по време на обучението. Към L_task добавяме наказание за прекомерна импулсна активност, за комуникация и за разрядност. Комуникационният член е нужен, защото два неврона с еднакъв брой импулси могат да имат различна хардуерна цена, ако единият изпраща събития само локално, а другият към няколко ядра.

$$L_{total} = L_{task} + \lambda_s R_{spike} + \lambda_c R_{comm} + \lambda_q R_{quant} \qquad (5)$$

R_spike е нормализираният брой импулси. R_quant описва ограничението на разрядността. За комуникацията се използва $R_{comm}=(1/Z)\Sigma_i S_i F_i D_i$, където S_i е броят импулси, F_i е ефективният fan-out, а D_i е оценка за комуникационното разстояние след картографиране. Изразът служи като регуляризационен член, който задава по-висока цена на събитията, преминаващи през повече хардуерни ресурси.

ИДЕНТИФИКАЦИЯ И КАЛИБРИРАНЕ НА МОДЕЛА

За реална платформа коефициентите трябва да се намерят от измервания. При m конфигурации в матрицата X записваме N_syn, N_spk, N_mem, N_hop, t_inf и константен член. В y е измерената енергия след изваждане на известния E_IO. Неизвестните e_syn, e_spk, e_mem, e_hop, P_0 и постоянният член образуват вектора c. Така калибрацията се свежда до регресионна задача.

$$\hat{c} = \arg\min_{c \geq 0} \|Xc - y\|^2 + \gamma\|c - c_{prior}\|^2 \qquad (6)$$

Ограничението $c \geq 0$ следва от физическия смисъл на параметрите, тъй като енергията на операция не може да бъде отрицателна. c_prior може да задава начална оценка от литература или EDA анализ, без тези стойности да се приемат за универсални. Микробенчмарковете трябва да натоварват различни компоненти на модела. Ако паметта, комуникацията и SynOp се изменят едновременно във всички тестове, отделните коефициенти трудно могат да бъдат идентифицирани. Подобна зависимост възниква и при P_0, когато t_inf почти не се изменя.

Част от конфигурациите се изключват от калибрацията и се използват за вали-

дация. За тях прогнозната E_inf се сравнява с измерената чрез RMSE, нормализирана RMSE или MAPE. Оценката на модела се извършва и при активности, разрядности или картографирания, които не са участвали при определянето на коефициентите.

Табл. 1. *Параметри на модела и последователност за тяхното калибриране.*

ПАРАМЕТЪР	СМИСЪЛ	ПРЯКА РОЛЯ	СЛЕД FPGA/EDA
N_SPK	ГЕНЕРИРАНИ ИМПУЛСИ	SNN ОПЕРАТОР	ХАРДУЕРНИ COUNTER
N_SYN	СИНАПТИЧНА АКТИВНОСТ	SPIKE/FAN OUT	RTL TRACE
N_MEM	ПАМЕТНИ ДОСТЪПИ	ТЕГЛОВИ МАСИВ	BRAM/SRAM
N_HOP	МЕЖДУБЛОКОВИ ПРЕХОДИ	NOC МРЕЖА	ROUTER/COUNTERS
E	ЕНЕРГИЙНИ КОЕФИЦЕНТИ	НОРМАЛИЗИРАНЕ	SILICON/BENCHMARKS
P_0	БАЗОВА МОЩНОСТ	ПАРАМЕТЪР	ИЗМЕРВАНЕ/EDA

МОДЕЛНО ИЗСЛЕДВАНЕ НА ЧУВСТВИТЕЛНОСТТА

Преди реалните измервания е извършен илюстративен анализ на уравнение (3), без да се моделира конкретен чип. Проверява се поведението на модела при промяна на ρ и L. За E_b и останалите коефициенти са използвани нормализирани примерни стойности, поради което резултатите от табл. 3 представляват моделно изследване, а не измерване върху силиций.

Табл. 3. *Илюстративна чувствителност. Стойностите са резултат от модела, не от измерване на силиций.*

СЦЕНАРИЙ	P	L	E/E-REF	
A : БАЗОВ	1.00	0.00	1.00	0%
B:	0.50	0.00	0.625	37.5%
C:	0.20	0.80	0.511	48.9%
D	0.50	0.80	0.436	56.4%

Табл. 3 показва, че намаляването на ρ от 1 до 0.5 не води автоматично до 50% по-ниска обща енергия, тъй като E_b и част от I/O разхода остават. Увеличаването на локалността допълнително намалява паметния и комуникационния член. Най-ниската стойност на енергията в разгледаните сценарии се получава при едновременно намаляване на активността и увеличаване на локалността.

Моделът позволява предварителна оценка на ефекта от отделни оптимизации. Ако измерването покаже, че P_0·t_inf формира повече от половината от E_inf, допълнителното разреждане на SNN ще има ограничено влияние върху общата консумация. В такъв случай по-голям ефект може да се очаква от намаляване на времето за инференция, оптимизация на тактовото управление или периферните блокове.

ЕКСПЕРИМЕНТАЛЕН ПРОТОКОЛ ЗА FPGA

Експерименталната проверка се предвижда върху FPGA с компактно цифрово невроморфно ядро, локална BRAM и проста събитийна шина. В RTL описанието се добавят броячи за входни импулси, активирани синапси, паметни достъпи, маршрутизирани събития и активни тактови цикли. По този начин един тест осигурява едновременно функционален резултат и измерими величини за енергийния модел.

При измерването се фиксират тактовата честота, захранващото напрежение, наборът от данни, batch size, периодът за установяване и точката на измерване. Всяка конфигурация се изпълнява многократно, като се записват средната стойност и разсейването. Мощността в покой се отделя от динамичната компонента. При наличие на външен процесор неговият разход се включва в E_IO или се отчита отделно, за да се запази съпоставимостта между конфигурациите.

При EDA анализ същото RTL описание може да се синтезира за стандартноклетъчна библиотека.

Табл. 4. *Предложен протокол за преминаване от софтуерен SNN към калибриран FPGA/EDA модел.*

Eman	*Използвани величини*	*критерий*
1.SNN	*Activiti,spikes*	$A \geq A\text{-}min$
2.Mapping	*N-mem*	$M \leq M - max$
3.FPGA	*L-inf*	*E-mean*
4.Calibration	*PO*	*RMSE*
5.Optimizacion	*P,L*	*E_A_L*

От симулацията след синтез или физическо разполагане се извлича комутационната активност, а power анализът определя статичната и динамичната компонента. За e_mem, e_hop и e_syn се използват отделни микробенчмаркове, така че коефициентите да бъдат определени за конкретната технология, а не чрез стойности в pJ от несъпоставима система.

ПРИЛОЖЕНИЕ КЪМ ASIC

След калибрацията моделът може да се използва при търсене на архитектурни параметри. Входните величини включват SNN графа, наличната локална памет и допустимата латентност. Оптимизират се картографирането, разрядността, праговете и политиката за изключване на блокове при минимална E_inf и зададените ограничения. При малък брой параметри е възможно пълно дискретно обхождане, а при по-голямо пространство могат да се използват еволюционни или Байесови методи.

Обучението и картографирането могат да бъдат включени в общ итеративен цикъл. Мрежата се разпределя върху хардуера и се изчислява R_comm, след което се дообучава с уравнение (5) и картографирането се обновява. Процесът се повтаря до достигане на малка промяна между последователните итерации. Намаляването на N_hop при запазена точност и без съществено увеличение на N_syn може да бъде проверено върху FPGA и чрез EDA анализ.

При ASIC към критериите се добавя и площта. По-голямата локална SRAM може да намали N_hop и част от паметния разход, но увеличава заеманата площ и leakage. Задачата може да включва ограничение $\text{Area}(\theta) \leq \text{Area_max}$ или площта да се разглежда като допълнителен критерий. Така оптимизацията се формулира едновременно по енергия, площ и време.

Моделът разделя алгоритмичната активност от енергийната цена на конкретната хардуерна реализация. Две SNN могат да имат еднакъв N_spk, но различен N_syn поради различен fan-out. При еднакъв N_syn е възможен и различен N_mem, когато архитектурите използват различна локалност на теглата. Поради това броят импулси сам по себе си не е достатъчен критерий за енергийна ефективност.

R_comm свързва обучението с физическото картографиране. Импулс, който остава в локалното ядро, има различна хардуерна цена от импулс, преминаващ през няколко маршрутизатора. Включването на този ефект в loss функцията насочва обучението към структура с по-нисък комуникационен разход. Размерът на ефекта подлежи на експериментална проверка.

Основното ограничение на текущия етап е липсата на калибрирани коефициенти за конкретна платформа. Поради това числените резултати представляват анализ на чувствителността, а не измерена консумация на готов чип. Експерименталната валидация изисква FPGA прототип, хардуерни броячи и измерване на енергията.

Сравнение с DNN или микроконтролер е коректно само при една и съща задача и близка точност. Публикувана минимална мощност на невроморфен чип не може директно да се съпоставя с енергията на система с различен набор от данни, времеви прозорец и периферия. Подобно сравнение не осигурява достатъчно основание за оценка на енергийната ефективност.

От модела могат да се проверят няколко конкретни твърдения. H1: при фиксирани картографиране и време за инференция $E(\rho)$ е афинна, а не строго пропорционална на импулсната активност, когато E_b е различно от нула. Това се проверява с няколко нива на ρ при еднакъв хардуер и после се сравнява линеен модел през началото с афинен модел.

H2: регуляризацията, която отчита комуникацията, трябва да намали N_hop по-силно от вариант, който наказва само броя импулси, при близка точност. За

това са нужни поне три обучения - без регуляризация, само с R_spike и с R_spike+R_comm - върху една и съща задача и еднакъв seed набор.

H3: след калибрация моделът трябва да предсказва енергията и за конфигурации, които не са участвали при определянето на коефициентите. Наборът за проверка се отделя предварително. Ако грешката там е близка до грешката върху калибрационните точки, моделът вече има практическа стойност.

H4: структурата на модела може да се пренася между различни цифрови събитийни реализации, но числените коефициенти трябва да се калибрират отново. Това може да се провери върху две FPGA конфигурации или FPGA и цифров ASIC модел. BrainScaleS-2 не е подходящ пряк тест за тази хипотеза, защото използва смесено-сигнална ускорена динамика [15].

Линейността по броячите улеснява калибрирането на модела, но не описва всички физически ефекти. DVFS, температурната зависимост, конфликтите в паметта или нелинейното clock gating могат да изискват допълнителни членове. Остатъчната грешка след калибрацията трябва да се анализира, за да се оцени дали избраната форма на модела е достатъчна.

Резултатът зависи и от характера на задачата. Keyword spotting, поток от събитийна камера и индустриален времеви сигнал могат да имат различна импулсна статистика. Калибрацията не трябва автоматично да се пренася между тях; параметрите се проверяват за конкретния клас натоварване при запазена обща структура на модела.

За възпроизводимост е необходимо освен крайната стойност за mJ/inference да се публикуват броячите, версията на SNN модела, картографирането, тактовата честота, разрядността и броят времеви стъпки. Тези данни позволяват резултатът да бъде проследен и сравнен от друг екип.

Различните невроморфни платформи имат различни граници на приложимост. TrueNorth е цифрова система с ниска консумация [13,14], BrainScaleS-2 използва ускорена смесено-сигнална динамика [15], а SpiNNaker2 е многоядрена програмируема архитектура [16]. Енергийните метрики трябва да се сравняват при съпоставими условия. Предложеният модел е насочен основно към цифрови събитийни реализации, при които броячите могат да се измерят пряко.

ЗАКЛЮЧЕНИЕ

Предложеният модел разделя енергията на синаптична обработка, импулси, памет, комуникация и базов/I/O разход. Броячите за активност могат да бъдат запазени при различни хардуерни реализации, докато енергийните коефициенти се калибрират за всяка конкретна платформа. Редуцираният модел показва взаимната зависимост между импулсната активност и локалността. FPGA валидацията включва измерване на E_inf и калибриране на коефициентите, след което моделът може да се използва при търсене на ASIC архитектурни параметри.

ЛИТЕРАТУРА

[1] Mead, C. Neuromorphic Electronic Systems. Proceedings of the IEEE, 78(10), 1629–1636, 1990. doi:10.1109/5.58356.

[2] Sze, V., Chen, Y.-H., Yang, T.-J., Emer, J. S. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. Proceedings of the IEEE, 105(12), 2295–2329, 2017. doi:10.1109/JPROC.2017.2761740.

[3] Horowitz, M. 1.1 Computing's Energy Problem (and What We Can Do About It). 2014 IEEE International Solid-State Circuits Conference (ISSCC), 10–14, 2014. doi:10.1109/ISSCC.2014.6757323.

[4] Davies, M. et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. IEEE Micro, 38(1), 82–99, 2018. doi:10.1109/MM.2018.112130359.

[5] Davies, M., Wild, A., Orchard, G. et al. Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook.

Proceedings of the IEEE, 109(5), 911–934, 2021. doi:10.1109/JPROC.2021.3067593.

[6] Blouw, P., Choo, X., Hunsberger, E., Eliasmith, C. Benchmarking Keyword Spotting Efficiency on Neuromorphic Hardware. Proceedings of NICE 2019, Article 1, 1–8. doi:10.1145/3320288.3320304.

[7] Ostrau, C., Klarhorst, C., Thies, M., Rückert, U. Benchmarking Neuromorphic Hardware and Its Energy Expenditure. Frontiers in Neuroscience, 16:873935, 2022. doi:10.3389/fnins.2022.873935.

[8] Xu, Y. et al. Optimizing Event-Based Neural Networks on Digital Neuromorphic Architecture: A Comprehensive Design Space Exploration. Frontiers in Neuroscience, 18:1335422, 2024. doi:10.3389/fnins.2024.1335422.

[9] Barchi, F., Parisi, E., Zanatta, L. et al. Energy Efficient and Low-Latency Spiking Neural Networks on Embedded Microcontrollers Through Spiking Activity Tuning. Neural Computing and Applications, 36, 18897–18917, 2024. doi:10.1007/s00521-024-10191-5.

[10] Yao, M., Richter, O., Zhao, G. et al. Spike-Based Dynamic Computing with Asynchronous Sensing-Computing Neuromorphic Chip. Nature Communications, 15, 4464, 2024. doi:10.1038/s41467-024-47811-6.

[11] Ferreira, P. M., Wang, S., Gao, Y., Benlarbi-Delai, A. A Comparative Review of Deep and Spiking Neural Networks for Edge AI Neuromorphic Circuits. Frontiers in Neuroscience, 19:1676570, 2025. doi:10.3389/fnins.2025.1676570.

[12] Karamimanesh, M., Abiri, E., Shahsavari, M., Hassanli, K., van Schaik, A., Eshraghian, J. Spiking Neural Networks on FPGA: A Survey of Methodologies and Recent Advancements. Neural Networks, 186, 107256, 2025. doi:10.1016/j.neunet.2025.107256.

[13] Merolla, P. A. et al. A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface. Science, 345(6197), 668–673, 2014. doi:10.1126/science.1254642.

[14] Akopyan, F. et al. TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34(10), 1537–1557, 2015. doi:10.1109/TCAD.2015.2474396.

[15] Pehle, C. et al. The BrainScaleS-2 Accelerated Neuromorphic System With Hybrid Plasticity. Frontiers in Neuroscience, 16:795876, 2022. doi:10.3389/fnins.2022.795876.

[16] Scholze, S. et al. The SpiNNaker2 Chip: A Many-Core Platform for Flexible and Scalable Brain-Inspired Computing. IEEE Open Journal of Circuits and Systems, 7, 465–478, 2026. doi:10.1109/OJCAS.2026.3714974.