УНИВЕРЗИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“ – БИТОЛА

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ И КОМУНИКАЦИСКИ ТЕХНОЛОГИИ – БИТОЛА

Информатички науки и компјутерско инженерство

АКЦЕЛЕРАЦИЈА НА ХАРДВЕРСКИ БАЗИРАНИ АРХИТЕКТУРИ НА КОНВОЛУЦИСКИ НЕВРОНСКИ МРЕЖИ ВО ВГРАДЛИВИ СИСТЕМИ (EMBEDDED SYSTEMS)

- докторски проект -

Кандидат
Дарко Пајковски
Број на индекс: 10/19/III

Ментор
проф. д-р Никола Рендевски

Содржина

Апстракт

Изминатата деценија донесе огромен развој на вештачката интелигенција и нејзините под области, посебно машинското учење. Се повеќе алгоритми од машинското учење, најдоа широка примена и надвор од своите рамки. Развојот и примената во различните сфери на целокупната под област е огромна, а една од најзначајните е се однесува на обработка на сликата (класификација по разни критериуми, локализација на објектите, сегментација итн). Во оваа проблематика најголем придонес односно најистакнати се конволуциските невронски мрежи. Единствен недостаток на овој тип на мрежи е количината на операции која што е потребно да се изврши за да се процесира сликата. Првобитно оваа комплексност на некој начин ја ограничување нивната примена, но од друга страна се повлече развојот на специјализирани акцелератори за одредени компактни електронски системи. За да се надминат ограничувањата на ресурсите кои постојат во овие системи, потребно е да се развива архитектура која ефикасно ќе ги користи расположливите ресурси. Темата на докторскиот проект е развивање на хардверски акцелератор, наменет за процесирање на конволуциските невронски мрежи. Бидејќи станува збор за развој на компактен акцелератор, конволуциските невронски мрежи нема да се процесираат во изворен формат, туку истите ќе се оптимизираат односно кастрат (анг. Pruning) со цел комплексноста да се редуцира. Според анализата од претходните вакви решенија, може да се заклучи дека развојот и оптимизацијата на ваквите алгоритми често е независен од развојот на хардверската архитектура, но тоа може да ги ограничи крајните резултати на акцелераторот. Ваквиот исход најчесто е последица на фактот дека оптимизирањето на алгоритамот не е направено во согласност расположливите хардверски ресурси кои ќе бидат користени при развој на акцелераторот. Развојот и оптимизацијата на алгоритмите во овој докторски проект ќе се прави согласност карактеристиките на крајниот програмабилен систем (FPGA, ang. Field-programmable-gate-array).

Клучни зборови: конволуциски невронски мрежи, хардвер, акцелератор, FPGA

1. Вовед

Во последната деценија, сведоци сме на интензивниот развој на вештачката интелигенција вклучувајќи ги и нејзините под области од кои една е и машинското учење (анг. machine learning). Брзиот развој во областа првично е поттикнат од развојот на хардвер кој што бил во можност во разумно време да обработи големи количества на податоци. Со текот на времето, хардверот станува сé помоќен овозможувајќи понатамошен развој на алгоритмите. Поради квалитетот на резултатите и можностите за широка примена во најразлични области стана јасно дека е исклучително важно извршувањето на комплексните алгоритми од машинското учење на вградливите електронски системи (анг.embedded systems). Овој тип на електронски системи, денес е достапен во огромен број уреди за крајните корисници во делот на апликации кои овозможуваат пресметување на работ (анг. edge computing), така што претставува еден мотив плус за забрзувањето на развојот на хардверските акцелератори, но во исто време и намалување на комплексноста на алгоритмите. Во овој тим на системи, од круцијална важност е да се креира специјализиран хардверски модул кој што ефикасно ќе ги користи расположливите ресурси, со цел да се искористи целиот достапен хардверски потенцијал. Темата и идејата на овој докторски проект е да се развие еден таков хардверски модул (акцелератор) кој што ќе биде ефикасен во поглед на извршувањето на еден алгоритам од машинското учење како конвулациските невронски мрежи (CNN). Иако, областа бележи огромен и интензивен развој, сеуште не можеме да кажеме дека се спојува со некој правец, така што конволуциските невронски мрежи се еден од најшироко користени алгоритми во областа на апликациите за машинско учење.

Покрај извонредните перформанси, CNN претставуваат модели на машинско учење кои бараат голема процесорска моќ на системите на кои се извршуваат. На пример, за класификација на една слика со помош на VGG-16 [1] CNN, хардверот извршува околу 31 милијарда операции. За обработка на големи количества на податоци CNN најчесто користат генерални решенија како што се графичките процесори (анг. Graphic Processing Unit, GPU) и единици за процесирање на тензор (анг. Tensor Processing Unit, TPU). На жалост, овие хардверски платформи не се (GPU и TPU) не се применливи во системите во кои ресурсите се ограничени, како и во вградливите системи (embedded systems). Потребата за извршување на CNN на вакви системи е огромно со оглед на фактот дека во последните години публикувани се повеќе од стотина трудови на теми за хардверски CNN акцелератори, како ASIC и FPGA технологии.

Покрај развојот на акцелераторите за специфични намени, забрзување при извршување на CNN може да се постигне на дополнителни два начини:

1. Оптимизација на комплексноста на моделот.
2. Кастрење (анг. Pruning) CNN.

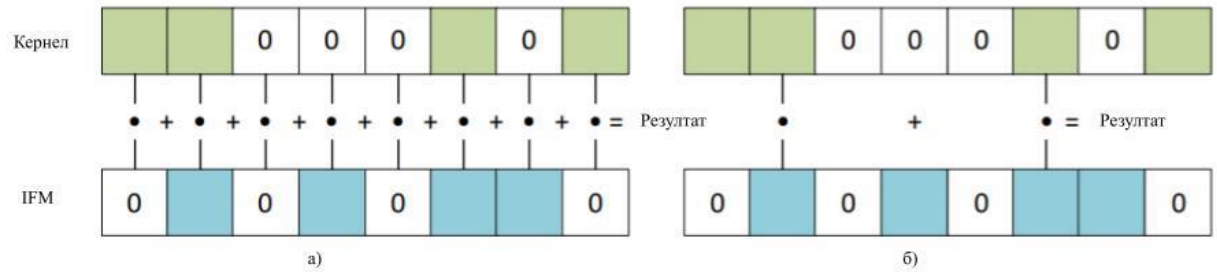
Оптимизацијата на комплексноста на моделот најчесто опфаќа редукција на моделот на макро ниво. Постојат CNN како што се MobileNet v1 [2] кои се направени со значајно намалување на комплексноста на мрежата во поглед на операции и слоеви притоа задржувајќи ги перформансите на некои драстично сложени модели. Друга,

најчеста оптимизација е со намалување на битовите кои се користат за репрезентација на броевите со подвижна децимала на фиксна децимала.

Кастрењето на CNN претставува процес во кој голем број на параметри на кернелот во конволуциските слоеви се поставува на вредност нула. Со оваа постапка се прекинуваат непотребните врски во CNN а како бенефит се добива мрежа која има помалку параметри кои влијаат на крајниот резултат. Редукцијата на параметрите може да се искористи на повеќе начини, а притоа главни придобивки се зголемувањето на перформансите на акцелераторот, намалување на потребата од мемориски ресурси за чување на параметрите на кернелот и намалување на потрошувачката на енергија за процесирање на CNN.

Развојот на алгоритмите за кастрење најчесто е раздвоен од развојот на архитектурата на акцелераторите. Исходот од ваквиот пристап за оптимизација е често архитектурата на акцелераторот кој не може да искористи голем дел од предностите кои алгоритамот за кастрење ги пружи. За да ги искористат сите предности на кастрениот CNN, акцелераторите мора да имаат значајно поголема комплексност во однос на архитектурите кои извршуваат не кастрени CNN. За да се намалат ваквите недостатоци, потребно е да се паралелно развиваат архитектурата и алгоритамот за CNN имајќи ги во предвид карактеристиките на расположливите хардверски ресурси на кои што акцелераторот ќе биде развиен и имплементиран.

Како што е познато, основна предност при процесирањето на кастрените мрежи е тоа што е потребно да се извршат помалку операции во однос на не кастрените CNN. Еден илустративен пример е прикажан на слика 1 на кој што се претставени два вектори со податоци над кои е потребно да се пресмета скаларен производ. Скаларниот производ се пресметува како збир на сите производи на елементите со зелена и сина боја кои се наоѓаат на иста позиција во двата вектори. Квадратите во боја претставуваат елементи различни од нула. Акцелераторот кој што нема можност за процесирање на кастрените мрежи треба да ги изврши сите операции (осум множења и седум собирања) како би се пресметала конечната вредност (слика 1а). Од сликата забележуваме дека само два производа помеѓу елементите во векторот имаат вредност различна од нула. Доколку акцелераторот има можност да ги прескокни непотребните операции (множење и собирање) крајниот резултат би можел да биди значајно побрзо пресметан (само две множења и собирање на крајниот резултат). Покрај перформансите за прескокнување на некои операции, во овој случај ќе се намали и потрошувачката на енергија.

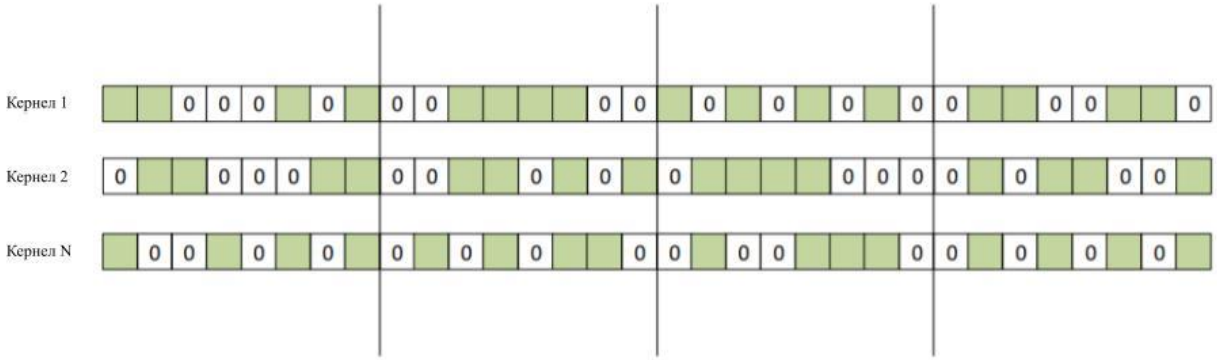


Слика бр. 1: Процесирање на не кастрени (а) и кастрени мрежа (б)

Двата вектори прикажани на слика 1, можат да го претставуваат кернелот и делот на влезната мапа на атрибути (анг. Input Feature, IFM) во конволуцискиот слој CNN над кој што треба да се пресмета скаларниот производ. Во најголем дел од случаевите нулите во IFM потекнуваат од активационите функции на претходните слоеви (најчесто ReLU) и нивниот распоред е непредвидлив. Многу од fine-grained алгоритмите за кастрење додаваат нули во кернелот и без претходно познат шаблон во смисла на позиција на преостанатите вредности во кернелот. Ова значи дека хардверската архитектура која процесира кастрени CNN во општ случај не смее да биде зависна од распоредот на елементите во векторот кои што се различни од нула. Комплексноста на ваквите архитектури е значајно поголема од комплексноста на архитектурите кои процесираат елементи со претходно познат редослед. Во општ случај, архитектурата мора да биде во состојба да прескокни произволен број параметри во кернелот и/или IFM што значајно ќе го усложни меморискиот дел на акцелераторот кој што бара дополнителна логика која одредува кој е следниот елемент кој што треба да се пребарува. Дополнителен проблем кој што се јавува потекнува од фактот дека сите кернели не се кастрат на ист начин. Исто така, процесирањето на кернелите е паралелно од страна на акцелераторот се со цел да се постигнат подобри перформанси, резултирајќи со балансирање на оптеретувањето помеѓу процесорските елементи.

Сите овие проблеми, во голема мера можат да се решат со усложнување на акцелераторот. Најчесто комплексните акцелератори не се погодни за вградливите системи (embedded systems). Покрај големината, постои и аспект на искористеност на расположливите хардверски ресурси во овие системи, а комплексните акцелератори немаат добри карактеристики во тој поглед. На пример во случаеви кога имаме FPGA базирани акцелератори, често се случува логиката за прескокнување на непотребните аритметички операции, баферите за балансирање на оптеретувањето на процесорските елементи и мемориски блокови на акцелераторот да зафатат скоро 100% од расположливите LUT (анг. Look-up-table), додека DSP блоковите (анг. Digital signal processing) останат неискористени. Не балансираната потрошувачка на ресурси најчесто доведува до ограничувања во скалирањето на акцелераторот што доведува до лимитирање во поглед на перформансите.

Непредвидливиот распоред на параметрите различни од нула се зема како основна причина за усложнување на архитектурата на акцелераторот. Она што не е познато е како да се влијае на распоредот на нулите внатре во IFM. Ако се намали влијанието во IFM, можно е да се редуцира ефикасноста на архитектурата така што истата нема да може да ги прескокнува нулите внатре во IFM. Исто така, доколку дополнително се ограничи алгоритамот за кастрење, можно е во целост да се исфрли наведената појава. На алгоритамот е можно да се влијае така што би се вовеле шаблони на позициите на параметрите кои што после кастрењето се различни од нула. На слика 2 е прикажано едно од ограничувањата како би се избегнала потполната варијабилност на позициите на значајните параметри.



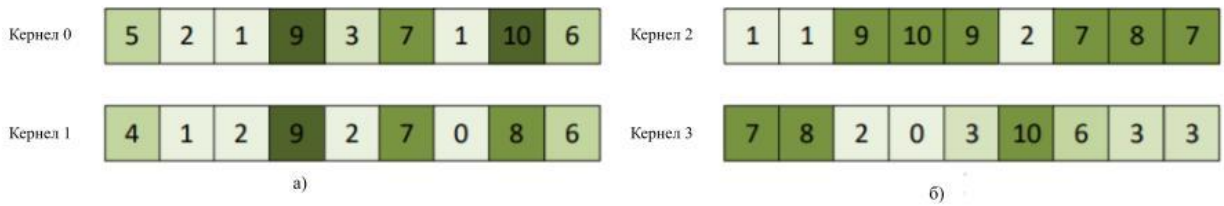
Слика бр. 2: – Кастрење на кернелот (оптимизирање)

Доколку параметрите на кернелот се кастрат во групи од 8 последователни параметри, така што во секоја група би останале однапред познат број на параметри, тогаш можно е да се проектира хардвер кој што ќе е во можност ефикасно да ги процесира CNN со намалена потрошувачка на ресурси. Оваа идеја е првично презентирана во [2]. Со овој пристап, прозорецот во кој што се бараат параметрите различни од нула се ограничува на 8 последователни елементи. Ваквите мали прозорци можат да се процесираат со прилично едноставен хардвер. Секој прозорец се обработува со идентичен број на операции. Резултатот ќе претставува компактна архитектура со високи перформанси, посебно во поглед на добиени перформанси по искористен хардверски ресурс, што е од огромна важност кај вградливите системи.

2. Развој на алгоритмите за кластерирање на кернелите во CNN

Идејата на овој докторски проект е да се развие нов алгоритам за кастрење (анг. Pruning) конволуциски невронски мрежи (анг. Convolutional Neural Networks, CNN), како и дигитална архитектура која ќе биде во можност да ги искористи предностите на новиот алгоритам при процесирање на кастрените CNN. Акцелераторот ќе биде развиен со посебен акцент на карактеристиките на расположливите хардверски ресурси на FPGA (Field Programmable Gate Array) технологијата.

Целта на алгоритмите за кластерирање е да се групираат кернелите во конволуциските слоеви во групи (кластери) по две или повеќе. Првична идеја е кернелите во еден кластер да се кастрат на ист начин во поглед на позицијата на параметрите кои ќе бидат кастрени. Сличноста на кернелите во кластерот се мери според позицијата на параметрите со најголеми вредности. Слика 3а илустрира пример кога се кернелите слични и кога треба да се постават во ист кластер, додека 3б е случај кога не треба да бидат во ист кластер. Нијансата на зелена боја одговара на амплитудата на вредностите во кернелот (потемната е поголема вредност).



Слика бр. 3: Пример слични вектори а) и помалку слични по позиција на големите вредности б)

Од сликата се забележува дека вредностите на кернелот 0 и 1 се далеку послични по амплитуда од вредностите на кернел 2 и 3. Бидејќи поголемите параметри на кернелот на CNN се сметаат за поважни за точност на мрежата, очекувано е да се добие помала деградација на точност на CNN доколку кернелот 0 и 1 ги кастриме на ист начин отколку кернелот 2 и 3. Прашањето е, каков критериум треба да се примени при формирање на маска за кастрење на кернел 2 и 3 притоа да се исфрли што е можно помал број важни параметри од двата кернели, а во исто време да бидат кастрени на ист начин во поглед на позициите.

Резултатот од работата на алгоритмите за кластерирање не е кастрен CNN туку само кластерите на кернелот кои и понатака даваат можност за процесирање на алгоритмите за кастрирање притоа давајќи бенефит кернелите во секој кластер да имаат параметри со големи амплитуди на исти позиции. Големи резултати постигнати при кастрење на CNN се показател на висока редундантност на параметрите. Целта е да се постигне баланс помеѓу нивоата на кастрење и комплексноста на процесирањето на CNN. Секогаш треба да се тежнее кон задржување на прифатливо кастрење, а притоа да не се деградира точноста на CNN. Со овој пристап би се намалила комплексноста на хардверскиот акцелератор во поглед на потребните ресурси. Причина за редуцирање на комплексноста на акцелераторот произлегува од фактот дека е потребен еден хардверски блок кој што селектира парови параметри/влезни атрибути чиј производ е вредност поголема од нула по кластер. Авторите на [4] ја објаснуваат оваа проблематика со помош на групирања на соседните кернели на CNN моделот.

3. Оптимизација и подобрување на постоечките алгоритми за кастрење на CNN со посебен акцент на FPGA

Предложениот алгоритам во трудот [3] е еден од ретките од групата на fine-gained алгоритми кои што на излез имаат кастрен CNN, така што позициите на параметрите различни од нула се однапред предвидени, односно на однапред познати места. Ваков пристап доведува до значаен придонес за намалување на комплексноста на хардверскиот акцелератор кој процесира оптимизиран CNN. Предности на овој систем во однос на останатите произлегуваат од тоа што на пример низа од по 8 параметри во кернелот

обработуваат идентичен број на MAC операции (слика 2). Ова значи дека, ако еден процесорски елемент е способен да процесира една ваква група во еден такт, а групите имаат идентичен број на MAC операции, многу лесно би се склопиле процесорските елементи без потреба од хардвер за балансирање на оптеретувањето.

Детталната анализа на алгоритамот од [3], покажува дека одредени конфигурации бараат широки мултиплексери кој што пак не се достапни на модерните FPGA архитектури. Алгоритамот го решава проблемот, ги оптимизира позициите на параметрите на CNN и со тоа допринесува за поедноставување на хардверот, но тие широки мултиплексери кои се користат во блоковите чии што производ треба да е поголем од нула ја поништуваат предноста посебно кога станува збор за имплементација на FPGA чиповите. Идејата на овој докторски проект е да ги подобри овие алгоритми односно да ја редуцира ширината на мултиплексерите за најмалку 50%, се со цел да се добие ефикасно мапирање на голем број модерни FPGA чипови.

3.1 *Развој на хардверските архитектури и можности за процесирање на оптимизирани и не оптимизирани CNN*

За да се постигне висока ефикасност архитектурата треба да се подели на повеќе специјализирани блокови. Најкомплексен блок во ваквата архитектура секогаш е конволуционото јадро (анг. Conv Core, CC) кое е специјализирано за процесирање на конволуционите FC слоеви. Покрај ова јадро постојат и блокови зза процесирање на останатите слоеви. Секое CC е составено од процесорски елементи (анг. Processing Element, PE), кои што можат паралелно да процесираат повеќе кернели од истиот конволуциски слој. За да акцелераторот биде комплетен CC блокот секогаш е во можност покрај CNN оптимизираните мрежи, да процесира и не оптимизирани. Ако мрежите се оптимизирани со некој друг алгоритам, архитектурата нема да биде во можност да ги користи предностите од оптимизираната мрежа, но ќе биде во можност да ја испроцесира како не оптимизирана.

4. Машинско учење (Machine Learning)

Машинското учење е област од вештачката интелигенција која бележи голем напредок од 90-тите години на минатиот век. Постојат мноштво на дефиниции за машинското учење, некои од нив се:

1. Машинското учење е процес во кој што компјутерите ги модифицираат своите акции станувајќи се поточни, а точноста претставува отстапка преземена од саканата акција [5]

2. Машинското учење е гранка од вештачката интелигенција која систематски ги применува алгоритмите како ви се извлекле информации од расположливите податоци [6].

3. Машинското учење е збир од методи за конструкција на математички модели кои можат автоматски да препознаваат шаблони во податоците [7].

Според авторите на [5], алгоритмите за машинско учење може да се поделат и спрема начинот на кој го креираат моделот:

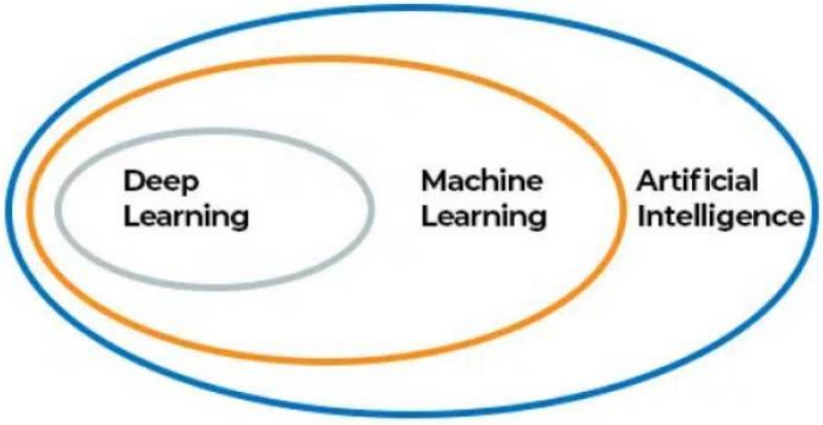
1. Надгледувано учење (анг. Supervised leaning) – Алгоритамот го генерализира одзивот на основа на тренинг множествата кои што содржат примери на влезни податоци заедно со точните одзиви.

2. Не надгледувано учење (анг. Unsupervised leaning) – Алгоритамот нема на располагање точни одзиви, туку користи само примери. Во процесот на тренирање се обидува да најде сличност помеѓу влезните податоци и креира модел кон може да ги категоризира влезните податоци спрема припадноста на некоја група.

3. Појачано учење (анг. Reinforcement leaning) – Алгоритамот добива повратна информација за тоа дали одговорот е точен или не, но не добива информација на кој начин да го поправи предвидувањето. Алгоритамот мора самостојно да ги истражи различните можности се додека не пронајде адекватен начин за да даде точен одговор.

4. Еволутивно учење (анг. Evolutionary learning) – Биолошката еволуција може да се смета како процес на учење. Организмите се прилагодуваат како би ја подобриле шансата за преживување.

Најзастапен тип на учење е надгледаното учење кое ќе биде во фокус на овој докторски проект. Подобласт на машинското учење е т.н. длабоко учење (анг. Deep Leaning). На слика 4 се претставени односот на вештачката интелигенција, спрема машинското односно длабокото учење.

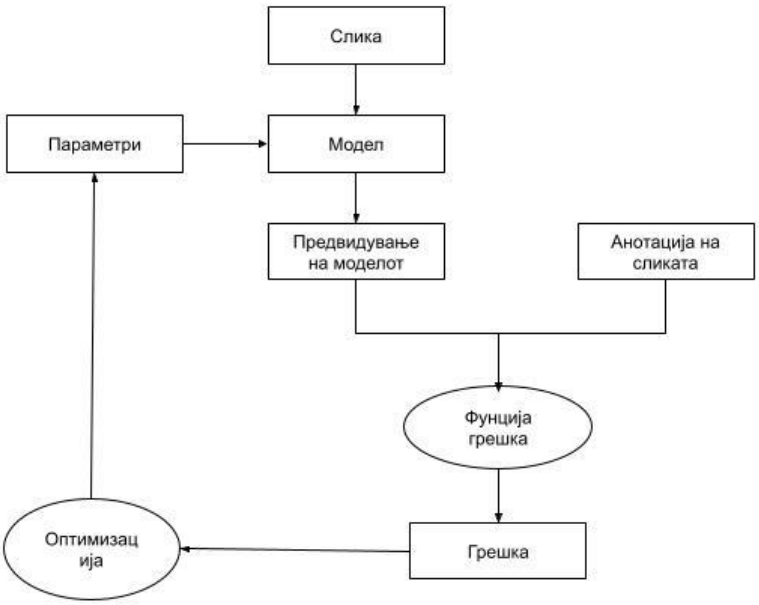


Слика бр. 4: Вештачката интелигенција во однос на машинското односно длабокото учење (Извор [8])

Доколку треба да се креира систем кој што ќе распределува слики во две групи, во зависност од тоа дали на сликата се наоѓа куче или мече, потребно е:

- Влезни податоци (слики на кои има куче, односно мече).
- Анотација на одреден број слики
- Функција на грешка која ќе покажува колку добро (лошо) моделот ги класифицира сликите

Процесот на учење е прикажан на слика 5. За решавање на одреден проблем е потребно да се одреди кој модел на машинско учење ќе се користи и да се постават почетните вредности на параметрите на моделот. Доколку станува збор на невронска мрежа, параметрите најчесто се иницијализираат на вредности кои се random генерирани. Во процесот на учење, моделот ги трансформира влезните податоци во репрезентација која ги претставува предвидените податоци на моделот. Предвидувањето се споредува со точната анотација на податоците, притоа функцијата ја враќа грешката за тековниот пример. Грешката се проследува до процесот кој што врши оптимизација на параметрите во мрежата кој што користи алгоритам за пропагација на грешката низ слоевите на невронската мрежа наназад. Како резултат од оптимизацијата, добиваме промена на параметрите на моделот. Вака променетите параметри најчесто резултираат со квалитетен модел во поглед на предвидувањата.



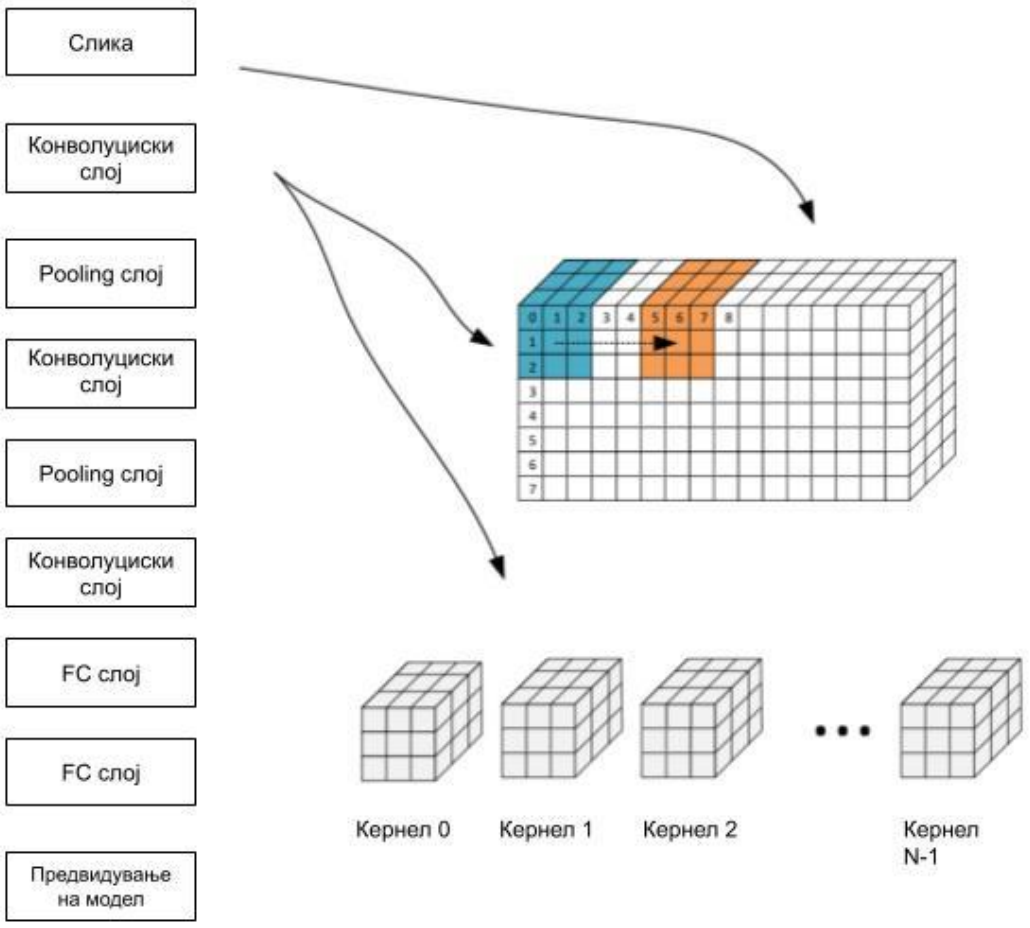
Слика бр. 5: Процесот на тренинг на модел со машинско учење

4.1 *Конволуциски невронски мрежи*

Како што е познато, почетокот на интензивниот развој на длабокото учење и конволуциските невронски мрежи се поврзува со откривањето на AlexNet CNN во 2012 година. Иако главните теоретски корени се поставени уште кон крајот на минатиот век [9]

недостатокот на хардверските ресурси го спречувале развојот и примената. На слика 6 е прикажан пример на конволуциска невронска мрежа која може да се користи за едноставни задачи, како на пример за класификација на цифри. Покрај конволуциските слоеви, CNN ја чинат и Pooling слоевите, потполно порвзани (анг. Fully connected, FC), слоеви за собирање (анг. Adding layer).

Конвулациските слоеви го сочинуваат кернелот кој што е најчесто 3D облик со големина од 3x3 каде длабочината е најчесто еднаква на влезната мапа на атрибутот. Првиот конволуциски слој врши обработка на сликата и доколку таа е во RGB формат, со длабочина 3, тогаш и кернелот на првиот слој има длабочина 3. На слика 6 е прикажана конволуција на првиот слој CNN чија што влезна слика ги има наведените карактеристики. Излезната вредност за секоја положба на кернелот се однесува на влезната мапа на атрибути (IFM) и се пресметува како скаларен производ на кернелот и дадениот прозорец на IFM. Пресметката на комплетната излезна мапа на атрибутот (анг. Output Feature Map, OFM) се врши така што кернелите се "лизгаат" по IFM. Двата прозорци означени со сина и портокалова боја учествуваат во креирање на точките OFM на две позиции. Чекорот во овој случај е еднаков на еден.



Слика бр. 6: Едноставна конволуциска невронска мрежа

5. Архитектура на акцелераторот наменет за FPGA платформа

Почетна точка при постапката за креирање на хардверски акцелератор е алгоритамот кој што треба хардверски да се акцелерира. Над алгоритамот треба да се применат техники за оптимизација, расчленување на циклусите, нивна детална анализа, оптимизација и верификација на истите. Со најголема комплексност во поглед на времето за процесирање и преносот на податоци се CNN слоевите кои што одземаат до 90% од вкупното време на процесирање [10]. Од наведеното е јасно дека крајните перформанси на акцелераторот ќе бидат во корелација со квалитетот на архитектурата што ќе се изгради за извршување на конволуциските слоеви.

Развојот на архитектурата на акцелераторот ќе се развива редоследно, почнувајќи од процесирањето на конволуциските слоеви. Чекор по чекор ќе се расчленуваат вгнездените циклуси и ќе се трасферираат (мапираат) на хардверската имплементација. Секој хардверски блок одделно ќе се прикаже со акцент на најважните детали при неговото градење и оптимизирање. После секоја успешна имплементација ќе се врши верификација со која што се потврдува коректноста на имплементацијата на акцелераторот.

6. Заклучок

Во овој труд, прикажани се концептите за развивање на нов алгоритам за оптимизирање на CNN мрежа и комплетен акцелератор наменет за употреба на FPGA развојна плоча. Целта е да се покаже дека со паралелен развој на алгоритмите за оптимизација и архитектурата на акцелераторот може да се креира јадро кое би имало подобри перформанси во однос на вкупните расположливи ресурси на FPGA плочите од досегашните решенија. Почетната фаза од алгоритамот е концепирана, но истата дополнително ќе се оптимизира и намени за FPGA архитектура. Исто така, треба дополнителната логика за исфрлање на непотребните операции да се подобри и со тоа значајно би се подобрил вкупниот перформанс. Акцелераторот е во можност да обработува како оптимизирани така и неоптимизирани мрежи, користејќи ги бенефитите на алгоритамот и балансирано користејќи ги расположливите хардверски ресурси се со цел подобрување на перформансите на FPGA плочите.

За демонстрација на перформансите на алгоритмот, ќе се оптимизираат повеќе архитектурно различни CNN модели како на пример ResNet50, VGG-16, MobileNet итн. Секој од овие модели има своја специфичност како што е компактноста, разноликоста во слоевите, перформансите, распространетоста итн.

7. Референци

1. Karen Simonyan, Andrew Zisserman, „Very Deep Convolutional Networks for Large Scale Image Recognition,“ u arXiv:1409.1556, 2014.
2. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijum Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, „MobileNets: Efficient Convolutional Neural Networks for Mobile Vision,“ arXiv:1704.04861, 2017
3. Hyeon-Ju Kang, „Accelerator-Aware Pruning for Convolutional Neural Networks,“ IEEE Transactions on Circuits and Systems for Video Technology, t. 30, pp. 2093-2103, 2020.
4. Xuda Zhou, Zidong Du, Qi Guo, Shaoli Liu, Chengsi Liu, Chao Wang, Xuehai Zhou, Ling Li, „Cambricon-S: Addressing Irregularity in Sparse Neural Networks through A Cooperative Software/Hardware Approach,“ u 2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Fukuoka, Japan, 2018.
5. Stephen Marsland, Machine Learning: An Algorithmic Perspective, Second Edition, Chapman & Hall/CRC, 2014.
6. Mariette Awad, Rahul Khanna, Machine Learning. In: Efficient Learning Machines, Apress, Berkeley, CA, 2015
7. Vranjković Vuk, Doktorska disertacija - Rekonfigurabilne arhitekture za hardversku akceleraciju prediktivnih modela mašinskog učenja, Novi Sad: Fakultet tehničkih nauka, 2015.
8. https://datamites.com/blog/machine-learning-vs-deep-learning/
9. Yann LeCun, Patrick Haffner, Léon Bottou, Yoshua Bengio, Object Recognition with Gradient-Based Learning, t. 1681, Berlin: Springer, Berlin, Heidelberg, 1999.
10. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, „Deep Residual Learning for Image Recognition,“ u IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016