



Universiteit
Leiden
The Netherlands

System-level design for efficient execution of CNNs at the edge

Minakova, S.

Citation

Minakova, S. (2022, November 24). *System-level design for efficient execution of CNNs at the edge*. Retrieved from <https://hdl.handle.net/1887/3487044>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/3487044>

Note: To cite this publication please use the final published version (if applicable).

Samenvatting

Convolutionele Neurale Netwerken (CNN's) zijn biologisch geïnspireerde modellen die in staat zijn grote hoeveelheden ongestructureerde data te verwerken. Door dit vermogen blinken CNN's uit in taken zoals beeldclassificatie, beeldsegmentatie en natuurlijke taalverwerking, en worden ze veel gebruikt om dergelijke taken uit te voeren in toepassingen zoals navigatie, gezichtsherkenning, medische beeldanalyse en meer. Tegenwoordig worden veel op CNN's gebaseerde applicaties uitgevoerd op zogeheten edge-platformen: mobiele telefoons, tablets, camera's, enz. Dit staat in contrast met de meer gebruikelijke aanpak waarbij de applicaties worden uitgevoerd op datacenters (in de cloud). In tegenstelling tot uitvoering in de cloud, vereist uitvoering op edge-platformen geen overdracht van de verzamelde gegevens (bv. beelden van een beveiligingscamera) via het internet, en garandeert zo een betere reactiesnelheid en veiligheid.

De uitvoering van op CNN's gebaseerde applicaties op edge-platformen is echter uitdagend vanwege de vereisten die aan de netwerken worden gesteld door de applicatie en het beoogde edge-platform. Van deze vereisten zijn de meest voorkomende: hoge nauwkeurigheid, hoge doorvoer, lage latentie, lage geheugenkosten en lage energiekosten. Deze vereisten maken het ontwerp van een CNN uitgevoerd op een edge-platform een complexe taak. Meestal wordt deze taak uitgevoerd met behulp van het state-of-the-art (SOTA) ontwerpproces. Het SOTA-ontwerpproces verkent CNN's met verschillende architecturen en parameters en probeert een CNN te vinden dat voldoet aan alle eisen die eraan worden gesteld.

Hoewel het SOTA-ontwerpproces goed uitzoekt *wat* moet worden uitgevoerd op het edge-platform (d.w.z. welke architectuur en welke parameters een CNN moet hebben), onderzoekt het niet *hoe* CNN's of applicaties gebaseerd op CNN's moeten worden uitgevoerd (d.w.z. hoe het de rekenkracht, geheugen en energie gebruikt die beschikbaar zijn op een edge-platform). In plaats daarvan past het SOTA-ontwerpproces principes toe die een negatief effect hebben op het ontwerp van CNN's en op CNN's gebaseerde applicaties uit-

gevoerd op edge-platformen. De eerste beperking is dat een CNN altijd laag-voor-laag wordt uitgevoerd. Deze sequentiële manier van CNN-uitvoering is wijdverbreid vanwege zijn eenvoud, maar kan het efficiënt gebruik van de beschikbare middelen op het edge-platform niet garanderen. Daardoor kan een CNN dat is ontworpen met behulp van het SOTA-ontwerpproces de beperkte middelen van een edge-platform inefficiënt gebruiken. De tweede beperking is dat een CNN-gebaseerde applicatie slechts één CNN gebruikt om zijn taak uit te voeren. Hierdoor mist het SOTA-ontwerpproces de middelen voor inter-CNN-optimalisaties en aanpassingsvermogen tijdens operatie (run-time adaptivity), welke belangrijk zijn voor bepaalde op CNN's gebaseerde applicaties.

In dit proefschrift richten we ons op het versoepelen van de twee bovengenoemde beperkingen, en het verminderen van de negatieve impact daarvan op het ontwerp van op CNN's gebaseerde applicaties uitgevoerd op edge-platformen. Hiervoor breiden we het SOTA-ontwerpproces uit en stellen we vier nieuwe methodologieën voor binnen het uitgebreide ontwerpproces.

De eerste twee methodieken richten zich op het versoepelen van de eerste beperking. Deze methodologieën vinden en handhaven een niet-sequentiële manier van CNN-uitvoering om efficiënt gebruik van middelen op een beoogd platform te garanderen. De eerste methodologie verdeelt de berekeningen van een CNN efficiënt onder de rekenkernen van een beoogd edge-platform en verhoogt daardoor de doorvoer. De tweede methode splitst de gegevens die tussen CNN-lagen worden uitgewisseld in delen en hergebruikt het platformgeheugen tussen deze delen, waardoor het benodigde geheugen voor een CNN wordt verminderd.

De laatste twee methodieken richten zich op het versoepelen van de tweede beperking. Deze methodologieën optimaliseren applicaties gebaseerd op CNN's, buiten het optimaliseren van individuele CNN's om. De derde methode introduceert aanpassingsvermogen tijdens operatie in een CNN applicatie. Dit maakt het mogelijk een applicatie te ontwerpen en efficiënte uit te voeren, dat zich kan aanpassen aan de noden van het moment. De vierde methodologie voert gezamenlijke geheugenoptimalisatie uit over een multi-CNN-applicatie (een applicatie die meerdere CNN's gebruikt om een taak uit te voeren). De methodologie maakt hoge geheugencompressie mogelijk om zo de multi-CNN-applicatie in de beperkte geheugencapaciteiten van een edge-platform te passen.