Technion – Israel Institute of Technology

Electrical Engineering Department

Laboratory of Control, Robotics and Machine Learning

**CNN Spatial Optimizations**

Winter Semester of 2019

Presenters: Ido Imanuel and Inna Batenkov

Supervisor: Gil Shomron

Table of Contents

[Table of Contents 2](#_Toc1902113)

[1. Abstract 3](#_Toc1902114)

[2. Project Specification 4](#_Toc1902115)

[2.1 Project Targets 4](#_Toc1902116)

[3. Introduction 5](#_Toc1902117)

[3.1 CNN Overview 5](#_Toc1902118)

[4. Architecture 7](#_Toc1902119)

[5. Optimization 8](#_Toc1902120)

[6. Results 9](#_Toc1902121)

[7. Further Work 10](#_Toc1902122)

[8. Bibolography 12](#_Toc1902123)

[9. Collaterals 13](#_Toc1902124)

# Abstract

Convolutional neural networks (CNNs) compute their output using weighted-sums of adjacent input elements. This method enables CNNs to achieve state-of-the-art results in a wide range of applications such as computer vision and speech recognition. However, it also comes with the cost of **high computational intensity**. Shomron et al [1] purposed exploiting the **spatial correlation** inherent in CNNs and predict activation values, thus reducing the needed computations in the network. They introduced a heuristic that predicts which activations are zero-valued according to nearby activation values, in a scheme they call **cross-neuron prediction**.

In this work, we further generalize the work done by Shomron et al, with the following steps:

1. Implement a custom CNN layer which we dub a “**Spatial**” layer, allowing for a fast and efficient prediction statistics collection on any chosen CNN architecture.
2. Create a hierarchal framework that allows for the cross-neuron predictions on **any CNN architecture** and with **any dataset** chosen.
3. Implement the class structure into

# Project Specification

In this section we will briefly explain the

## Project Targets

# Introduction

## CNN Overview

Deep Convolutional Neural Networks (CNNs)

perform billions of operations for classifying a single input. To

reduce these computations, this paper offers a solution that leverages

a combination of runtime information and the algorithmic structure

of CNNs. Specifically, in numerous modern CNNs, the outputs of

compute-heavy convolution operations are fed to activation units

that output zero if their input is negative. By exploiting this unique

algorithmic property, we propose a predictive early activation

technique, dubbed SnaPEA. This technique cuts the computation

of convolution operations short if it determines that the output will

be negative. SnaPEA can operate in two distinct modes, exact and

predictive. In the exact mode, with no loss in classification accuracy,

SnaPEA statically re-orders the weights based on their signs and

periodically performs a single-bit sign check on the partial sum.

Once the partial sum drops below zero, the rest of computations can

simply be ignored, since the output value will be zero in any case.

In the predictive mode, which trades the classification accuracy

for larger savings, SnaPEA speculatively cuts the computation

short even earlier than the exact mode. To control the accuracy, we

develop a multi-variable optimization algorithm that thresholds the

degree of speculation. As such, the proposed algorithm exposes a

knob to gracefully navigate the trade-offs between the classification

accuracy and computation reduction. Compared to a state-of-the-art

CNN accelerator, SnaPEA in the exact mode, yields, on average,

28% speedup and 16% energy reduction in various modern CNNs

without affecting their classification accuracy. With 3% loss in

classification accuracy, on average, 67.8% of the convolutional

layers can operate in the predictive mode. The average speedup and

energy saving of these layers are 2.02× and 1.89×, respectively. The

benefits grow to a maximum of 3.59× speedup and 3.14× energy

reduction. Compared to static pruning approaches, which are

complimentary to the dynamic approach of SnaPEA, our proposed

technique offers up to 63% speedup and 49% energy reduction

across the convolution layers with no loss in classification accuracy.

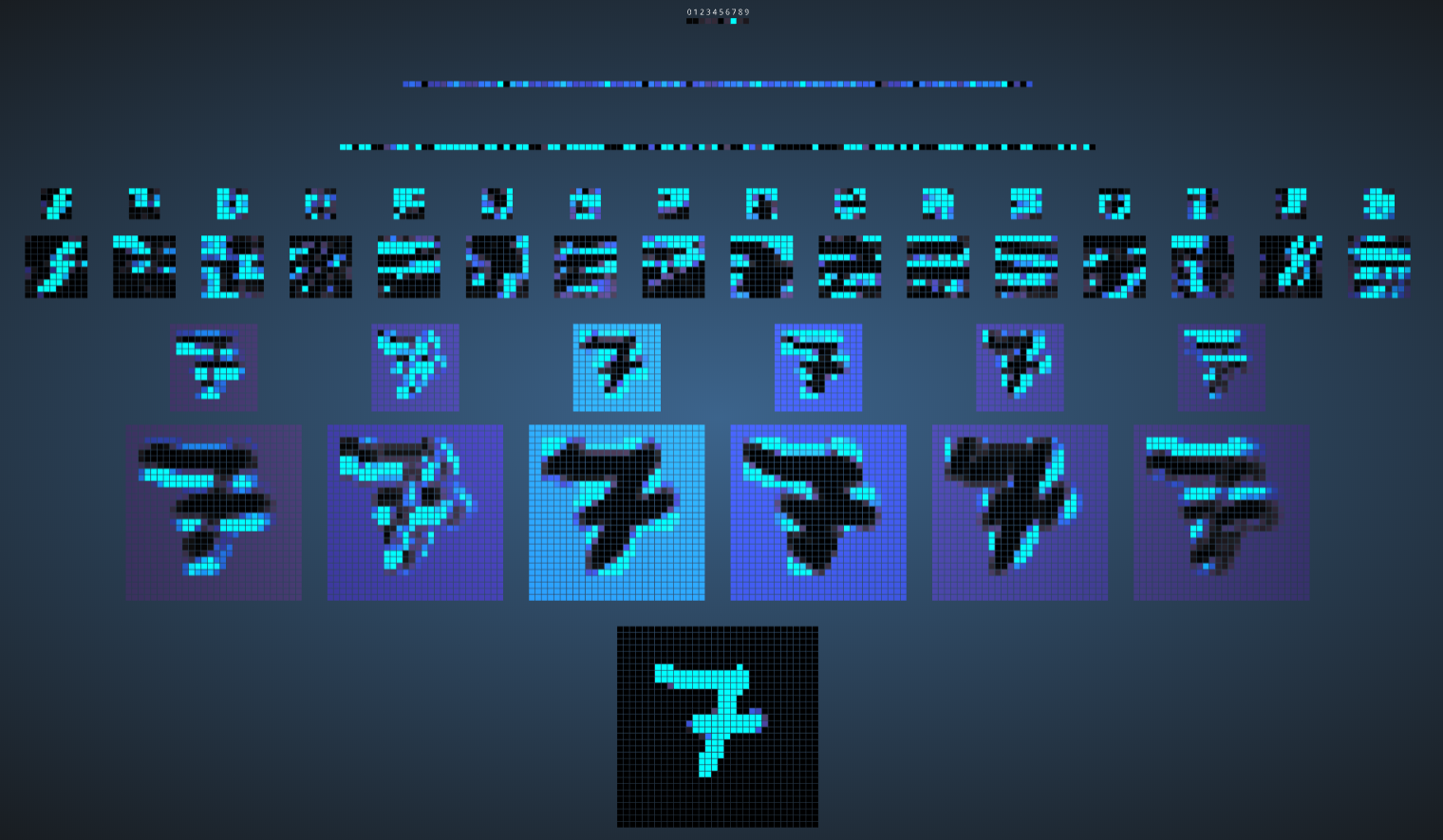


Figure 1: CNN Visualization on MNIST trained CNN classifier for the figure 7 [2]

# Architecture

# Optimization

# Results

# Further Work

# Bibolography

# Collaterals