POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Scuola di Ingegneria Industriale e dell'Informazione

Predictive Analysis of Deep Neural Networks in Cloud-Edge Computing Continuum

Relatore: Danilo Ardagna
Correlatore: Marco Lattuada

Tesi di Laurea di:
Giacomo Bossi
Matricola 883758

Anno Accademico 2018-2019

Sommario

Nell'ultima decade l'utilizzo di deep neural network all'interno di applicazioni intelligenti ha avuto un successo crescente e i dispositivi mobili e IoT hanno avuto un ruolo fondamentale in questo sviluppo. A causa delle limitazioni tecniche di questi device, inizialmente il paradigma computazionale si è indirizzato sullo sviluppo di un approccio cloud-only, approccio basato sulla concentrazione della computazione sul server cloud e sul flusso dei dati non elaborati dai dispositivi mobili al provider del servizio. Negli ultimi anni la crescente potenza di questi device e il progresso delle tecnologie di comunicazione mobili ha dato spazio ad un nuovo modello: il computing continuum. In questo scenario, la collaborazione computazionale tra i dispositivi mobili e i server in cloud apre a nuovi possibilità di sviluppo. Grazie all'utilizzo delle risorse dei dispositivi mobili è possibile ridurre sia la latenza del servizio sia il carico sul server e migliorare la privacy sui dati scambiati.

Lo scopo di questa tesi è di esplorare le opportunità offerte dal computing continuum per implementare reti neurali atte al riconoscimento delle immagini, cercando di ottenere il miglior trade-off tra le performance e la privacy di generiche reti neurali pre-allenate su differenti dispositivi mobili. Per raggiungere tale scopo abbiamo sviluppato il sistema distribuito DiNeSys che, sfruttando le potenzialità del framework Apache Thrift e di Tensorflow, è in grado di creare un ambiente di testing per effettuare in modo automatizzato la profilazione delle applicazioni per studiarne le prestazioni. In questo ambiente distribuito abbiamo testato differenti coppie di reti neurali, generate dal frazionamento di una singola rete neurale pre-allenata in due parti consecutive e allocando la prima sottorete al client mobile e la seconda sottorete al server, muovendo in questo modo una porzione del carico computazionale vicino al sensore sul nodo mobile. Infine, sono stati sviluppati modelli di machine learning per effettuare la predizione delle prestazioni delle due sottoreti fornendo informazioni utili per valutare i compromessi computazionali.

Abstract

In the last decade, the use of deep neural networks within intelligent applications had a steadily increased while IoT and mobile devices had played a fundamental role in this development. Due to the initial technical limitations of these devices, the computational paradigm adopted was based on a cloud-only deployment approach, a model based on the centralization of the computing capacity on the cloud server while the flow of raw data was generated by mobile devices and sent to the service provider. In the last few years, the growing power of these devices and the progress of mobile communication technologies led to a new model: the computing continuum. In this scenario, the computational collaboration between mobile devices and cloud servers opens up new possibilities for development. Thanks to the use of mobile device resources, it is possible to reduce both the latency of the service and the load on the server side and improve the privacy of the data exchanged. The goal of this thesis is to explore the opportunities provided by the computing continuum to implement neural networks for image recognition, trying to obtain the best trade-off between the performance and privacy of generic pre-trained neural networks on different mobile devices. To achieve this objective we developed the DiNeSys distributed system which, exploiting the potentiality of the Apache Thrift and Tensorflow frameworks, is able to create a testing environment able to automate an application profiling in order to study its performance. In this distributed environment we have tested different pairs of neural networks, generated by the splitting of a single pre-trained neural network into two consecutive sub-networks where the first sub-network is deployed on the mobile client while the second one on the server, thus moving a portion of the computational load near the sensor on the mobile node. Finally, machine learning models have been developed to predict the performance of the two sub-networks, providing useful insights to study the computational trade-offs.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The growing spread of IoT devices in the last decade and the need to make these smart devices through the use of complex algorithms has been one of the major problems faced by the Information Technology industry.

The model initially designed to provide these features to devices with low computing power was to use a computing paradigm based entirely on the cloud.

But the growing number of IoT devices and technical advances, which have greatly increased the computing capacity of these devices, have allowed the introduction of a new paradigm, computing continuum, a distributed collaborative computing system.

The systems deployed in a collaborative computing scenario open up new possibilities for developing new applications, in which the computational load is partitioned optimally according to the technical specifications of the mobile device and the network bandwidth available. Continuum computing can expand the range of action and improve the performance of a cloud-only system and manages to mitigate the defects of a system based exclusively on the edge.

This collaboration can also be extended to the inference of the Convolutional Neural Network, where a high number of operations are required to classify an image through these networks. Currently, the status quo approach is to perform all Deep Neural Network processing in the cloud, leaving the mobile device the only task of acquiring the data through a sensor and sending it raw to the cloud.

To demonstrate that the continuum computing paradigm can also be implemented in this scenario, we have therefore built a collaborative distributed system DiNeSys - Distributed Neural System - capable of dividing a pre-trained convolutional neural network into two different parts and dividing the computational load between a mobile device and a cloud server.

Therefore we tested some well-known Convolutional Neural Networks in every possible cut that could be made on them, thus dividing the weight of the layer calculation in each possible variant. In this scenario, we have therefore tried to give a series of insights on which the optimal computational load is, having different devices and mobile connections available.

After analyzing the performance dataset, given that the convolution layers

were the part that most weighed on the total inference times on the mobile side, through a series of Machine Learning models we tried to estimate the calculation times.

Besides, by partially computing the input acquired by the sensor on the mobile device, they are no longer sent to the cloud server, therefore to the end-point of the classification service for raw data. In fact, by applying convolution functions, this data is "filtered" by losing the part of information not necessary for the provision of the service, therefore to the classification of the image while preserving the privacy of the person requesting the service.

The thesis chapters are organized as follows: Chapter 2 outlines the state of the art. Chapter 3 provides the formulation of the problem, indicating the problems that we would have addressed and what characteristics the distributed system had to have in the development phase. Chapter 4 shows how the DiNeSys system was developed, what components it has, and what features it can provide. Chapter 5, the experimental chapter, contains the analysis of the performances obtained with different devices, the prediction of the average inference times of the convolutional layers, and the analysis of the preservation of privacy obtained according to the cut performed on CNN.

Chapter 6

Conclusions

This thesis started the exploration of the opportunity to run Convolutional Neural Networks in computing continuum systems, trying to show and demonstrate the capacity that this technology can offer. The system we developed and tested has provided some useful insights about the possible development of collaborative neural computing. We tested the system with three different boards, a low-end board (Raspberry Pi 3), a mid-end board (Odroid N2) and a high-end board (Nvidia Jetson X2) in a computing continuum scenario with a Cloud Server. From the results of the tests obtained on the different boards, we have created a dataset and we have used it to train a series of machine learning models to estimate the processing times of the convolutional layers. The machine learning models we used were the Decision Tree Regressor, Random Forest Regressor and the XGBoost Regressor.

We have seen in our prediction tests how a low power board like the Raspberry Pi 3 is not indicated in a computing continuum scenario, due to the instability in the calculation times and the poor performance. The performances obtained by this board have been sub optimal in the case in which neural networks with a high number of floating-point operations are used, while we have noticed excellent inference times in networks optimized for mobile devices (e.g. MobileNet). In the prediction of the performance of the convolutional layer inference time, we obtained an average MAPE - Mean Average Percentage Error - on the estimation time of the convolutional layers of the VGG19 of approximately 40% using the results of the VGG16 as the dataset and using as ML method a Random Forest Regressor properly tuned.

Odroid N2 board has a calculation capacity similar to that of a smartphone. Indeed, This board appears to be more exploited in continuous computing scenario, obtaining convincing performance. The Odroid N2 board has achieved remarkable performance increases on partial inference times with a medium and high throughput connection, saving about 60% of the total computing time compared to an edge-only scenario. Since it was not possible to test this board with the VGG19, we will therefore report the prediction of the inference times on the VGG16 using a part of it as a dataset. The average error using the

MAPE index on the prediction of the inference times of a convolutional layer was approximately 50%.

Nvidia Tegra X2 was found to be a board not particularly suitable for a computing continuum scenario due to its performance, given that in the tests carried out it performs better if used exclusively, that is, by computing the entire inference and sending to the cloud server the result of the image recognition application. The inference times on this board, given that with a dedicated GPU, were very stable, and with them the errors on the inference times on the convolutional layers were very low. In the prediction of the performance of the convolutional layer inference time, we obtained an average MAPE - Mean Average Percentage Error - on the estimation time of the convolutional layers of the VGG19 below 20% using the results of the VGG16 as the dataset. Also in this case the Machine Learning model that outperformed the other was the Random Forest Regressor.

Finally, we created a metric based on the Mean Structural Similarity index (MSSIM), an index used to evaluate the quality of an image following a transformation or compression, to identify the amount of privacy preserved. In fact, breaking the neural network into an intermediate position allows to send the service provider a pre-processed image containing only the information necessary for the image recognition and classification service.

The results were excellent with decay on all the classes of images used of the overall similarity after the first convolutional block of the VGG16, to then be very dissimilar after the second block (maximum MSSIM index on the output of the layer less than 10%, average MSSIM index on the output of the layer below 2%).

The computing continuum is an early technology, it has numerous advantages with which some disadvantages can be associated. The search for stability and a compromise between them is what needs to be the focus of future studies. With the expansion of the test sample both through the use of different mobile devices and through the experimentation of new neural networks, it will be possible to compose a more comprehensive sample of what this technology is able to offer.

The next step is to create a tool that automates the computational load on the edge and cloud in an automated way, to exploit all the available capacities in the best possible way. A dynamic resource allocator will be indeed a necessary step for this system to become more mature. Indeed, if a congested connection, a throughput loss or a particularly busy cloud server are present, it is possible to reduce the weight of the data sent by increasing the calculation part reserved for the cloud edge. Otherwise, if a low-powered mobile edge or with a very high data throughput are present, it is possible to move a greater part of the computation to the cloud.

We can imagine a dynamic resource allocator that also is able to measure the amount of information that is exchanged with the service provider, in order to understand how much privacy is preserved. Another possible optimization may also be to create ad hoc Deep Neural Networks, combining neural networks

optimized for mobile devices, through the quantization of operations on floating points, and a second part optimized to exploit the GPUs of a possible Cloud Server.

Bibliography

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

F. Chollet. *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.

E. Gianniti, L. Zhang, and D. Ardagna. Performance prediction of gpu-based deep learning applications. In *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 167–170. IEEE, 2018.

J. Hauswald, T. Manville, Q. Zheng, R. Dreslinski, C. Chakrabarti, and T. Mudge. A hybrid approach to offloading mobile image classification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8375–8379. IEEE, 2014.

D. O. Hebb. *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall, 1949.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

D. Kahanwal, D. T. Singh, et al. The distributed computing paradigms: P2p, grid, cluster, cloud, and jungle. *arXiv preprint arXiv:1311.3070*, 2013.

Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News*, 45(1):615–629, 2017.

Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995, 1995.

W. Magalhaes, H. Gomes, L. Marinho, G. Aguiar, and P. Silveira. Investigating mobile edge-cloud trade-offs of object detection with yolo. In *Anais do VII Symposium on Knowledge Discovery, Mining and Learning*, pages 49–56. SBC, 2019.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

S. A. Osia, A. S. Shamsabadi, S. Sajadmanesh, A. Taheri, K. Katevas, H. R. Rabiee, N. D. Lane, and H. Haddadi. A hybrid deep learning architecture for privacy-preserving mobile analytics. *IEEE Internet of Things Journal*, 2020.

W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

M. Slee, A. Agarwal, and M. Kwiatkowski. Thrift: Scalable cross-language services implementation. *Facebook White Paper*, 5(8), 2007.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.