UNIVERSITY OF SOUTHAMPTON

Efficient Teacher-Student Architectures for Human Activity Recognition via Soft Labels and Binarization

by

Yipeng Shen

(ORCID: orcid.org/0009-0007-4363-9460)

A thesis submitted in partial fulfillment for the
degree of Master of Philosophy

in the
Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

March 2024

Research Thesis: Declaration of Authorship

Print name: YIPENG SHEN

Title of thesis: Efficient Teacher-Student Architectures for Human Activity Recognition via Soft Labels and Binarization

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Delete as appropriate None of this work has been published before submission;
8. or Parts of this work have been published as: [please list references below]

Signature: ____________________

Date: ____________________

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Master of Philosophy

by Yipeng Shen
(ORCID: orcid.org/0009-0007-4363-9460)

Human Activity Recognition (HAR) applications are most commonly deployed on embedded systems with limited computational resources. Our work focuses on applying deep learning methods to HAR and developing compact architectures.

The first chapter of this report introduces a novel label representation for HAR in which we introduce the soft label shown to be capable of inducing better representation performance than that of the one-hot label. We will further investigate the teacher-student architecture for HAR. In our approach, we incorporate soft label by the teacher that supervises the next generation of training via the students. Experiments on 3 benchmark datasets, widely used in the community, which confirm that after a few generations of training, the model's performance surpasses that of the one-hot label. We also introduce the ECE, to avoid over-confident predictions, we use ECE as a performance metric, to evaluate the calibration performance of the HAR models. The experimental results also confirm that the teacher-student architecture effectively reduces the ECE and trains well-calibrated networks.

In the second chapter of this report, we evaluate the application of Binary Neural Networks (BNNs) in Human Activity Recognition (HAR) more suitable for constraints of embedded systems the features of embedded systems. Our goal is to significantly reduce the storage requirements and forward propagation latency of the model. We use XNOR-Net as the backbone architecture, where the weights, activation functions, and inputs to the convolutional layer are binary. The most crucial aspect is that the convolution operation is replaced by XNOR, resulting in a 32-fold reduction in memory usage and a 58-fold reduction in convolution operation latency. This enables operations to be performed on CPUs with limited computing power, rather than powerful GPUs, in most cases. We also examine the impact of using BNNs on the model's performance and the potential for transfer learning. Our findings show that these benefits do not come at the cost of accuracy or Expected Calibration Error (ECE) performance. However, the dataset we used has different sensors in different body parts, making transfer learning challenging.

In the third chapter, we study the application of a hybrid XNOR-Net and teacher-student architecture in HAR. The teacher network is first trained with a hard label that supervises the BNN student networks. Our approach improves the performance of future generations (i.e., the students of the student).

Finally, as part of our previous research, we participated in the OU-ISIR Wearable Sensor-based Gait Challenge in 2019 as part of an international competition in HAR and finished as runners-up. This involved gender and age-related multitasking learning. The gradient normalization algorithm was used in conjunction with the hybrid ResNet and BLSTM blocks. However, we no longer use it in subsequent research as the employed dataset is a single classification challenge rather than multi-task learning.

This research contributes to the ongoing advancement in HAR, offering insights and methodologies that may inspire future research in this field.

Contents

List of abbreviations

Terminology

HAR	Human Activity Recognition
KD	Knowledge Distillation
ECE	Expected Calibration Error
MAE	Mean Absolute Error
MSE	Mean Square Error
SGD	Stochastic Gradient Descent
SGDM	Stochastic Gradient Descent with Moment
ADAM	Adaptive Moment Estimation
CPU	Central Processing Unit
GPU	Graphics Processing Unit
DSP	Digital Signal Processors
CWTM	Confidence Weighted by Teacher Max
DKPP	Dark Knowledge Permuted Predictions
IMU	Inertial Measurement Unit
BERT	Bidirectional Encoder Representations from Transformers
NAS	Neural Architecture Search

Datasets

FOG	The Daphnet Freezing of Gait Dataset
PAMAP2	Physical Activity Monitoring DataSet

Models

CNN	Convolutional Neural Network
FCN	Fully Convolutional Network
ResNet	Residual Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BNN	Binary Neural Networks
BANS	Born Again Neural Networks

Chapter 1

Introduction

Over the past decade, deep learning technology has emerged as an effective tool for a wide range of applications, particularly in computer vision and natural language processing (Baghezza et al., 2020).

Human Activity Recognition (HAR) has received extensive attention in recent years due to its significant impact and diverse societal applications, such as health care (Sagha et al., 2011), gaming (Kang et al., 2004), and assistive technologies (Bächlin et al., 2010). HAR is primarily based on sensor data and collecting such data requires specialized hardware. However, recent technological advancements have made it easier to access such data due to the integration of various sensors into smartphones, such as accelerometers, gyroscopes, and other tracking sensors. These sensors have enabled the development of convenient and ubiquitous, yet affordable, health monitoring. In the past, the typical approach to HAR was limited to extracting hand-crafted features from raw sensor data, and the performance of these HAR classification methods was directly tied to the relevance of these hand-crafted features, which is a time-consuming and resource-intensive task.

Deep learning technology has become the mainstream technology in natural language processing, data mining, and computer vision (LeCun et al., 2015). A typical example of such technology is AlexNet (Krizhevsky et al., 2012), which achieved great success and gained state-of-the-art performance on the ImageNet LSVRC-2010 competition (Hammerla et al., 2016).

Deep learning has also been applied to HAR, eliminating the need for extracting specific hand-crafted features (Guan and Plötz, 2017; Edel and Köppe, 2016; Hammerla et al., 2016). As HAR is often based on time series data, deep learning architectures such as RNN and LSTM components are well-suited to enhance its performance. These architectures model the inherent time-series characteristics of sensor data, which reflect different human activities (Alharbi and Farrahi, 2018; Yao et al., 2017; Davarci et al., 2017; Steven Eyobu and Han, 2018; Delgado-Escaño et al., 2019). Studies have shown that the automatic feature extraction from inertial sensors by deep neural networks outperforms hand-crafted features. The temporal dynamics in such data can also be extracted by convolution windows and recurrent components, which obtain the relationship between different time intervals (Yao et al., 2017). Furthermore, the combination of multiple RNNs with convolution layers has been citep to be an effective approach for inertial sensor data (Yao et al., 2017; Delgado-Escaño et al., 2019).

Multi-task learning is utilized in HAR. In the competition we participated in, the task was to predict the age and gender of subjects simultaneously based on wearable sensor data (Delgado-Escaño et al., 2019; Ruder, 2017). These studies show that a neural network that outputs multiple predictions provides better performance than splitting them into separate single-task predictions. This is due to the generation of more general features. However, in some cases, different targets have different criteria.

In the competition mentioned earlier, the goal was to predict the age and gender based on sensor data. Usually, cross-entropy is used as the loss function for classification, while MAE and MSE are used for regression. However, as MSE is often much larger than MAE, the network tends to focus on the aging task and neglect the gender goal. To address this issue, we modified the gradient normalization algorithm in (Chen et al., 2017) to automatically balance the training process in deep multitask models through dynamic adjustment of gradient magnitudes. These approaches aim to improve the model architecture with a given one-hot label. However, the one-hot label has limited information, as it cannot represent the potential relationship between different categories. To address this, we explore the use of soft labels.

While deep learning has greatly improved model accuracy, in real-world classification systems, accuracy is not the only measure of performance. Confidence is also crucial, as it ensures the correctness of a model's predictions. For instance, in automated healthcare, if the confidence level of a diagnosis result is low, the result must be referred to human doctors (Jiang et al., 2012). Another example is self-driving cars, where neural networks are used to detect pedestrians and other obstacles. To ensure safety, tasks such as these require a high confidence level in detecting the presence or absence of immediate obstacles (Bojarski et al., 2016). Hence, in some applications, the deep learning network should provide a calibrated confidence measure in addition to its predictions.

In other words, the probability associated with the predicted class label should accurately reflect its likelihood of being correct (Guo et al., 2017). Additionally, a high-confidence prediction provides valuable information that contributes to its trustworthiness, which is particularly important for deep neural networks whose classification processes can be difficult to interpret.

As demonstrated in (Guo et al., 2017), despite the remarkable improvement in accuracy of modern neural networks in recent years, they are often no longer well-calibrated. The authors compare a 5-layer LeNet (LeCun et al., 1998) and a 110-layer ResNet (Al-Shedivat et al., 2017) on the CIFAR-100 dataset and find that while the LeNet is well-calibrated and has a high confidence level, the ResNet is not well-calibrated but has

much higher accuracy. Furthermore, (Müller et al., 2019) have shown that label smoothing can be used to calibrate a network and reduce its ECE through a teacher-student architecture. In our work, we adopt this architecture to develop a well-calibrated and high-performing network for HAR.
The size of the model is crucial in HAR as these applications are often deployed on small devices with limited computational resources. Knowledge distillation, a model compression method, is applied by training a smaller model to mimic a pre-trained, larger model (or ensemble of models). This is known as the teacher-student architecture, where the large model serves as the teacher and the small model is the student. The teacher is often a high-capacity model with high performance, while the student is a compact model. The goal of knowledge distillation is to transfer knowledge from the teacher to the student and benefit from the compactness of the student model without sacrificing too much performance.

In recent years, a new perspective on knowledge distillation has emerged, referred to as Born Again Neural Networks (BANS), in which the student models are trained to be identically parameterized with their corresponding teachers. This approach has been applied successfully in computer vision and language modeling and is expected to be a good fit for the HAR task.

In real-world classification systems, accuracy is not the only factor that matters. Confidence is equally important as it ensures the correctness of the model. For example, in automated healthcare, if the confidence level of a diagnosis result is low, it must be reviewed by human doctors.

1.1 List of contributions

The contributions of this thesis as follows:

1. To the best of our knowledge, we are the first to introduce the ECE metric in HAR and investigate the significance of model calibration. We also explore how ECE relates to the classification performance of various types of activities.

2. We propose the teacher-student architectures to generate the soft label for training the next generations of students. This approach has rich information and a more powerful representation. By employing KD and BANS, this results in an ensemble model for more efficiency and improved ECE performance obtaining state-of-the-art performance for HAR on the Daphnet dataset.

3. We apply the XNOR-Net architecture as the backbone in BNN. This approach achieves a significant reduction in the memory size and forward propagation latency with only partial loss of F1 score.

4. We combine the teacher-student architectures with the XNOR-Net to train a high-performance binary student network. This also reduces the ECE and calibrates the model by the noise introduced via binarization.

5. We evaluate the performance of transfer learning on BNN with the Daphnet, Opportunity, and PAMAP2 dataset.

Chapter 2

Literature review of human activity recognition

2.1 Human activity recognition

Human activity recognition is a challenging time-series classification task that aims to provide information about human body activities and detect simple or complex movements in the real world. This technology can improve the quality of life in various areas, such as geriatric care, rehabilitation, daily life documentation, personal health, and assistance to people with cognitive impairments (Golestani and Moghaddam, 2020). Traditional methods for HAR require deep domain expertise and techniques from signal processing to design raw data features that are suitable for machine learning models. The two main methods for deploying Human Activity Recognition (HAR) systems are external sensors and wearable sensors (Lara and Labrador, 2012). The external approach involves setting up a monitoring device at a fixed point, and the user is expected to interact with it (Wang et al., 2015). This approach often uses vision-based techniques and has been extensively studied for human activity analysis, but it faces several challenges such as coverage, accuracy, privacy, and cost. For example, it requires infrastructure support, like installing cameras in surveillance areas, which can be expensive. Additionally, if the user goes out of the camera's range, the device will not be able to capture any data (Bodor et al., 2003).

On the other hand, wearable sensors, such as accelerometers, gyroscopes, and magnetometers, are used to convert human movement into signal patterns for activity recognition (Kumari et al., 2017). Advances in embedded sensor technology have made it possible to monitor user activity using smart devices. Body sensors are designed to capture the state of the user and their environment, and they utilize information from heterogeneous sensors that are connected to the subject's body. This allows for continuous monitoring of numerous physiological signals and is useful for authentication, health and aging, and activity recognition in sports and exercise monitoring applications (Chetty and White, 2016; Lu et al., 2017). The use of smartwatches and smartphones in human activity monitoring has been reported, and satisfactory performance has been achieved (Lu et al., 2017).

Prior research (Banos et al., 2014) has shown that for capturing the significant motion variation found in activities with intricate details, like domestic chores, extended window sizes are typically necessary. However, a number of activities could take advantage of smaller window sizes. Activities engaging the entire body or multiple parts can be identified more readily and also allow for the adjustment of the window duration. Walking, jogging, running, and various other sports exercises are examples of such activities. These activities provide a more comprehensive description compared to those that involve only certain body parts, such as particular jumps and isolated limb movements. In these instances, some data windows obtained from specific body parts may not significantly contribute to differentiation, causing the recognition process to depend on a smaller set of informative windows. To counteract this, it's necessary to gather more data from the parts that yield larger, more informative windows.

2.2 Datasets Overview

Three datasets related to HAR are employed in the experiments. Figure 2.1 illustrates a visualization of label distribution, which is unbalanced. Hence using the weighted label strategy is necessary. Also, Figure 2.2 presents a visualization of a 45-frame segment from the Daphnet Gait Dataset, depicting the Normalized Data Over Frame. This visualization distinctly highlights the intense fluctuations observed during freeze states, in contrast to the relative stability in no-freeze conditions. The other two datasets are similarly visualized, employing the same methodology but encompassing more dimensions.

Tables 2.1, 2.2, and 2.3 detail the key information of the Daphnet Freezing of Gait, OPPORTUNITY, and PAMAP2 datasets respectively.

Daphnet Gait Dataset: This dataset (Bachlin et al., 2010) is a binary classification dataset consisting of recordings from 10 participants diagnosed with Parkinson's disease (PD). Dataset activities correspond to recognizing whether or not gait freeze occurs based on wearable acceleration sensors. The dataset was recorded in a lab environment with the subjects were instructed to carry out activities with a high likelihood of inducing freezing of gait, which is a common motor complication in PD.

Opportunity Dataset: This dataset (Roggen et al., 2010) contains recordings from various wearables and environment sensors from four participants who carry out common kitchen activities, such as Open/Close Door, Dishwasher, and Fridge, via Inertial Measurement Units (IMUs) at 30Hz. Each participant is recorded in five different runs.

Detail	**Description**
Wearable Sensors	3 wireless sensors (ankle, thigh, hip), 3D acceleration
Sampling Frequency	64 Hz
Sample Length	151,987
Number of different classes	9
Recording Sessions	10 subjects, 1-3 runs each
Annotations	0: not experiment, 1: no freeze, 2: freeze
Data Format	Sensor readings and annotations in text matrix format
Instances	Combined data from all runs for each subject

Table 2.1: Daphnet Freezing of Gait Dataset Details

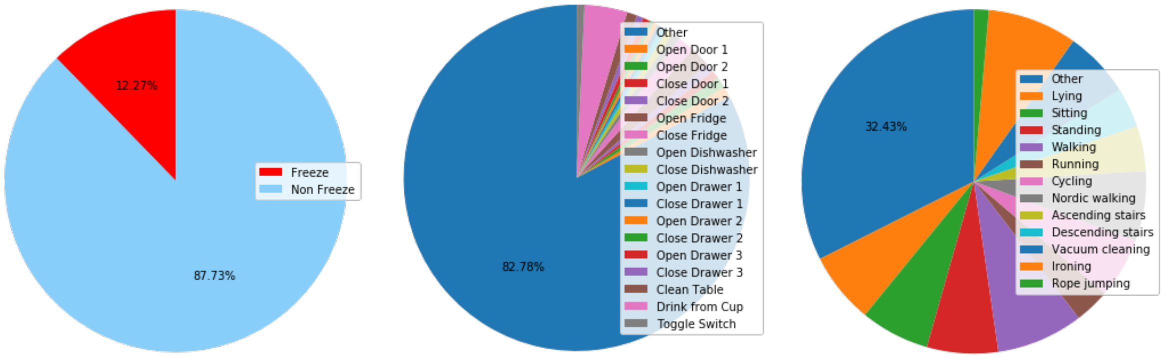
PAMAP2 Dataset: The physical activity monitoring dataset (Reiss and Stricker, 2012) is similar to the opportunity dataset, consisting of nine participants performing 12 kinds of daily physical activities, such as cycling, walking, sitting. The sensors used in the inertial measurement units (IMUs) include accelerometers, gyroscopes, magnetometers, temperature, and heart rate.

Detail	**Description**
Wearable Sensors	7 IMUs, 12 3D accelerometers
Sampling Frequency	30 Hz
Sample Length	51,116
Number of different classes	77
Recording Sessions	4 subjects, 6 runs each (5 ADL, 1 drill)
Annotations	Locomotion modes, actions, objects, gestures, activities
Data Format	Sensor readings and annotations in text matrix format
Instances	All subjects/recordings; Locomotion: 3653, Gestures: 2551

Table 2.2: OPPORTUNITY Dataset Details

Detail	**Description**
Wearable Sensors	3 Colibri wireless IMUs, positions: wrist, chest, ankle
Sampling Frequency	IMUs: 100Hz, HR Monitor: 9Hz
Sample Length	319,352
Number of different classes	52
Recording Sessions	9 subjects, various activities including optional ones
Annotations	Activity labeling via GUI
Data Format	Sensor readings in text matrix format
Instances	Each subject's data collected according to protocol

Table 2.3: PAMAP2 Physical Activity Monitoring Dataset Details



(a) Daphnet dataset (b) Opportunity dataset (c) PAMPA2 dataset

FIGURE 2.1: Label distribution of HAR datasets, the irrelevant labels are in the majority, and labels are grossly unbalanced.

2.3 The essential of model compression

In HAR, the original datasets are often quite large, therefore, features are usually computed on the segments of the available data using a sliding window. They are also often stacked with their derivatives (Lara and Labrador, 2013). Also, HAR applications are often used in wearable devices which are not computationally powerful.
However, the practical application of deep learning is often limited by its scale of storage and computation. For example, if the VGG-16 network contains about 140 million floating point parameters, the entire network needs more than 500 Megabytes of storage space (Simonyan and Zisserman, 2014). At present, Although in theory such computations could be executed by a Turing machine, practically, they are typically conducted using high-performance parallel devices due to the impracticality of achieving timely results otherwise. Therefore, the compression of neural network can be promising for HAR. According to machine learning theory and existing deep model compression methods, there are divided into four main categories:

- Network pruning
- Low-rank factorization
- Transferred/compact convolutional filters
- Knowledge distillation

Neural network pruning aims to remove the redundant parts of networks with good performance but high cost of resources. Although the learning ability of large neural networks is obvious, in fact, not all neural networks are useful after the training process, and the idea of neural network pruning is to remove these useless parts without affecting network performance. Low-rank factorization techniques use matrix/tensor factorization to estimate the information parameters of deep learning models. Moreover, A special

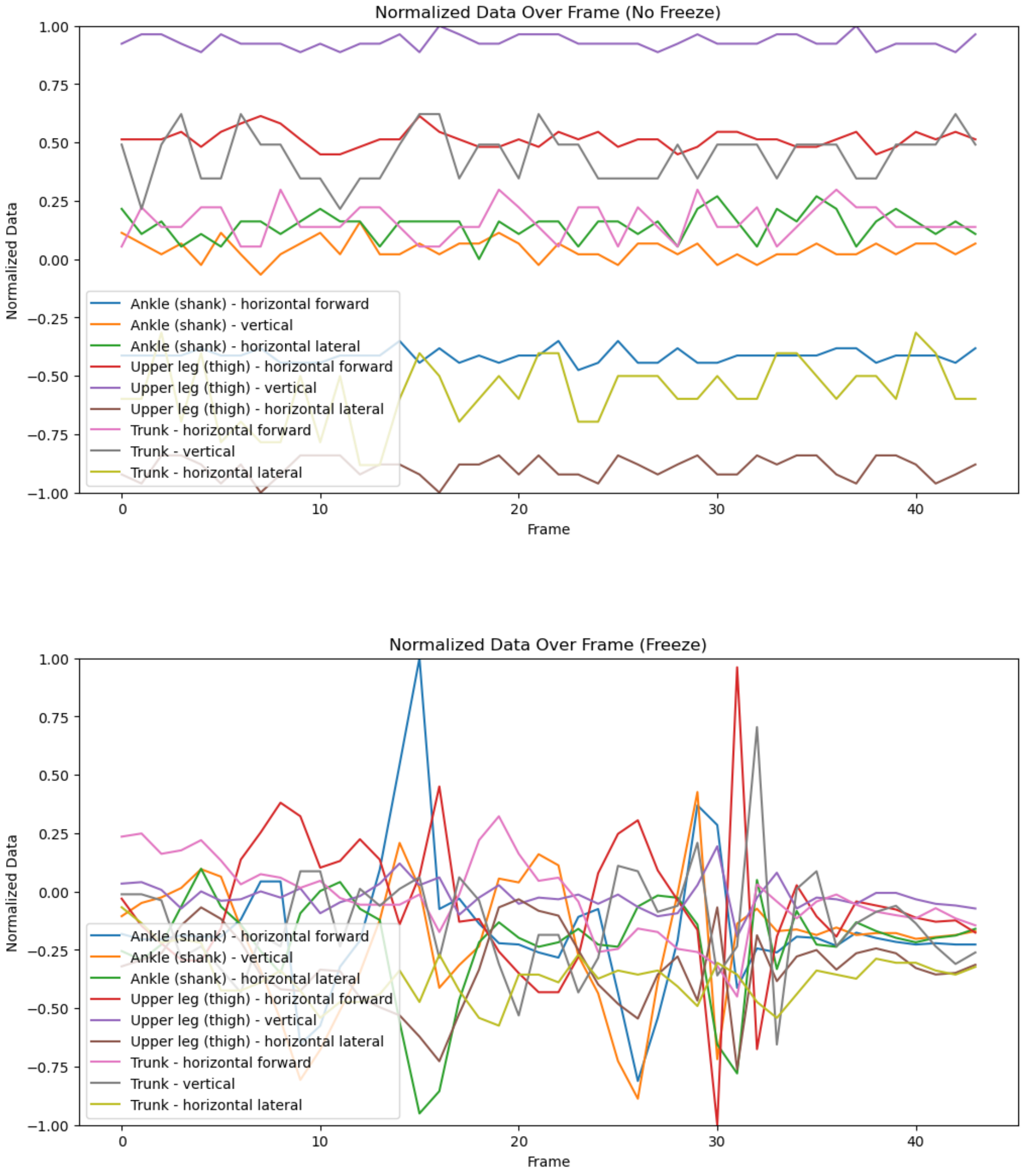


FIGURE 2.2: Visualizes a segment (length 45) of the Daphnet Gait Dataset, showing the Normalized Data Over Frame. It includes both freeze and no-freeze results

structured convolution filter is designed to reduce the storage and computational complexity.

Knowledge distillation, introduced by (Hinton et al., 2015), uses soft labels which contain potentially rich additional information. It introduces the teacher-student architecture while focusing on model compression and transferring knowledge from one machine-learning model to another. It trains a more compact neural network to reproduce the output of a larger network. The knowledge of one network can be transferred to another network, and those two networks can be isomorphic or heterogeneous, which can be used to transform the network from a large network into a small network and retain performance close to that of the large network. Chapter 3 will illustrate the further research on soft label, and the derived structures, Born again network (Furlanello et al., 2018; Yim et al., 2017), where a student model supervises itself to train a duplicate version in the iteration. It is then shown that the student eventually becomes the master after a few generations. After performing an ensemble of all students, (Bucilu et al., 2006) proposed a feasible method to compress the knowledge from the ensemble models into a single model.

In this thesis, we adopt the teacher-student method, where the same architecture is used for both the teacher and the student. We call this algorithm the soft-label generator and provide theoretical evidence to confirm its feasibility.

2.4 To Calibrate or not to calibrate

Following the definition by (Guo et al., 2017) the problem is set in supervised multi-class classification with the input $X \in \chi$ and label $Y \in \gamma$. Let h be the trained neural network with $h(X) = (\hat{Y}, \hat{P})$. Where $\hat{Y}$ is a class prediction and $\hat{P}$ is its associated confidence.To calibrate the confidence as true probability, we define the perfect calibration as Eq. 2.1

$$\mathcal{P}(\hat{Y} = Y | \hat{P} = p) = p, \quad \forall p \in [0, 1] \tag{2.1}$$

However, achieving the perfect calibration is impossible because of the finite number of samples. To estimate the empirical approximation, the samples are then grouped into M interval bins with $\frac{1}{M}$ size, depending on their confidence level. Let B_m be the set of indices of samples whose prediction confidence falls into the interval $I_m = (\frac{m-1}{M}, \frac{m}{M})$. Therefore, the accuracy of B_m is defined as in Eq. 2.2. where $\hat{y}_i$ and y_i are the predicted and true class labels for sample i respectively.

$$\mathrm{acc(B_m)} = \frac{1}{\mathrm{|B_m|}} \sum_{\mathrm{i \in B_m}} 1(\mathrm{\hat{y}_i = y_i}) \tag{2.2}$$

Furthermore, the average confidence within bin B_m is as in Eq. 2.3. where $\hat{p}_i$ is the confidence for sample i.

Back to Eq. 2.1, the left-hand and right-hand sides are exactly $acc(B_m)$ and $conf(B_m)$. Hence, the perfectly calibrated model will have $acc(B_m) = conf(B_m)$ for all indications

$$\mathrm{conf(B_m)} = \frac{1}{|\mathrm{B_m}|} \sum_{\mathrm{i \in B_m}} \mathrm{\hat{p}_i} \tag{2.3}$$

To measure the miscalculations of models, the difference in the expectation between the accuracy and confidence is employed as in Eq. 2.4. This term is also named expected calibration error (ECE) as in Naeini et al. (2015) which is the primary empirical metric to measure the calibration

$$E_{\hat{P}} \left[|\mathcal{P}(\hat{Y} = Y | \hat{P} = p) - p| \right] \tag{2.4}$$

For the finite number of samples, the approximation formula is Eq.2.5, in which the predictions are partitioned prediction into M bins with weights

$$\mathrm{ECE} = \sum_{\mathrm{m=1}}^{\mathrm{M}} \frac{|\mathrm{B_m}|}{\mathrm{n}} |\mathrm{acc(B_m)} - \mathrm{conf(B_m)}| \tag{2.5}$$

2.5 Rich information in the soft label

(Hinton et al., 2015) show that the soft label carries rich information, especially on the confidence on non-correct label. For example, Suppose the dataset space is four classification problem, and the confidence vector of trained neural network output is [0.7, 0.15, 0.14, 0.01], compare with the ground truth target [1, 0, 0, 0]. The predicated label of the trained neural network is the *argmax* selected term, so its actual output classification is correct. While, it is interesting on the incorrect label, which shows more information that: the trained neural network illustrates the second and third dimension are almost the same and very tiny possibility on the last dimension.

To review the foundation of machine learning, the cross-entropy loss function is widely employed. The first Derivative of it in Eq. 2.6:

$$\frac{\partial L}{\partial z_i} = \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial z_i} \tag{2.6}$$

Where L is the cross-entropy loss function, z_i is the gradient of neuron output and a_j is the Softmax function

For the first term:

$$\frac{\partial L}{\partial a_j} = \frac{-\sum_j y_i \ln a_j}{\partial a_j} = -\sum_j y_i \frac{1}{a_j}$$

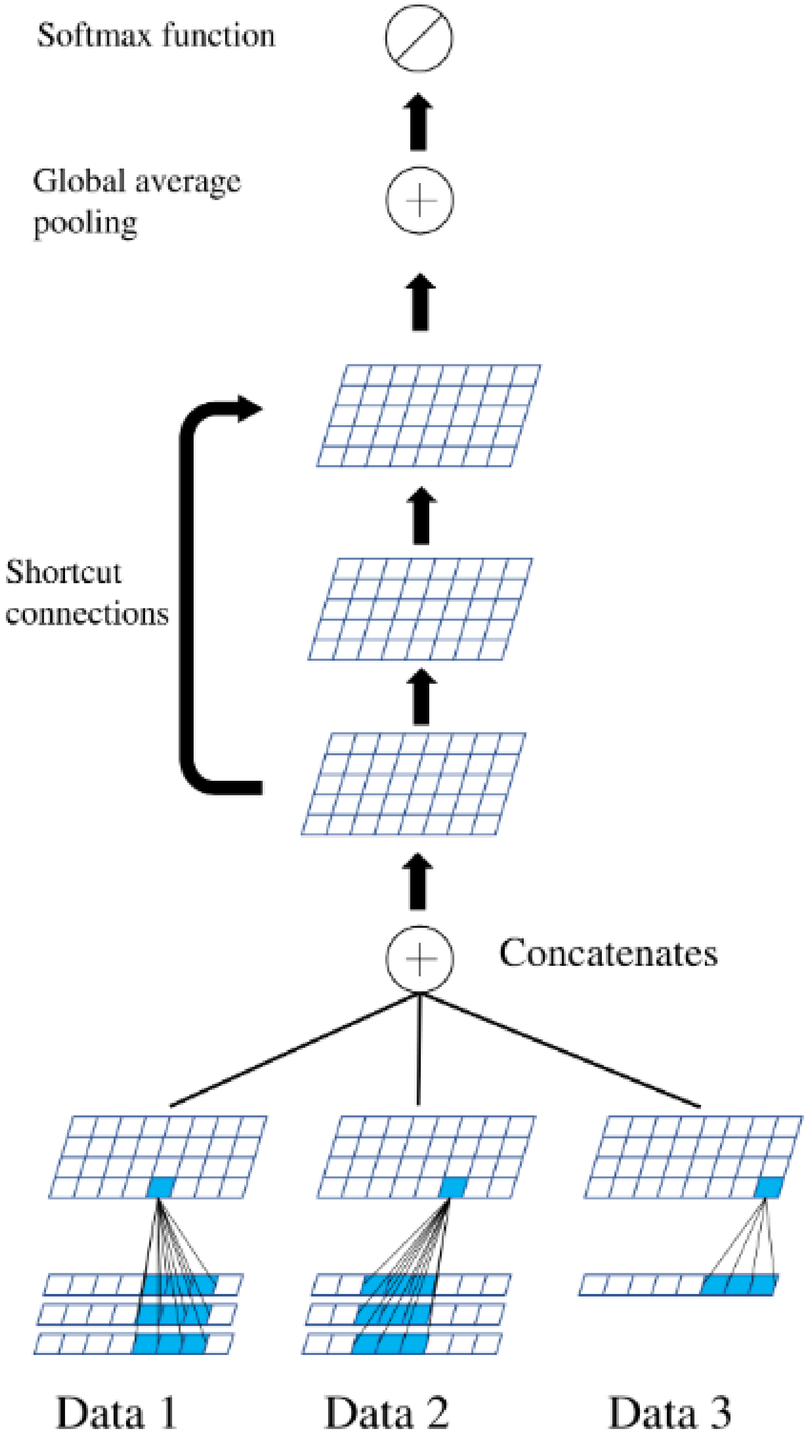


Figure 2.3: ResNet architecture the data is segmented to sensor-independent first. Then the x,y,z dimensions from the same accelerometer are grouped so they are distinguished from other sensors.

For the second term:
when $i = j$

$$\frac{\partial a_j}{\partial z_i} = \frac{\partial \frac{e^{z_i}}{\sum_k e^{z_k}}}{\partial z_i} = \frac{\sum_k e^{z_k} e^{z_i} - (e^{z_i})^2}{(\sum_k e^{z_k})^2} = (\frac{e^{z_i}}{\sum_k e^{z_k}})(1 - \frac{e^{z_i}}{\sum_k e^{z_k}}) = a_i(1 - a_i)$$

when $i \neq j$

$$\frac{\partial a_j}{\partial z_i} = \frac{\partial}{\partial z_i} \left(\frac{e^{z_j}}{\sum_k e^{z_k}} \right) = -e^{z_j} \left(\frac{1}{(\sum_k e^{z_k})^2} \right) = -a_i a_j$$

Thus:

$$\frac{\partial L}{\partial z_i} = (-\sum_j y_j \frac{1}{a_j}) \frac{\partial a_j}{\partial z_i} = -\frac{y_i}{a_i} a_i(1 - a_i) + \sum_{j \neq i} \frac{y_i}{a_j} a_i a_j = -y_i + y_i a_i + \sum_{j \neq i} \frac{y_i}{a_i} = -y_i + a_i \sum_j y_j$$

Where y_i is the ground truth with one-hot label. For the example above, the gradient $z_i = -1 + 0.7 = -0.3$
Obviously, it only employed the confidence on the correct label term and discards others confidence. Our interesting is that employ those 'discard' information to further boost the neural network. However, obtaining the soft label distribution of data on each label is a significantly challenging task by manual labeling.
It can be obtained using forwarding propagation on a trained model, namely a teacher model. The well-performing teacher model generates the correct label with high confidence (low probability of classifying in an incorrect category).
In distillation, knowledge is transferred from the teacher model to the student through a minimization loss function, where the target is the probability-like distribution predicted by the teacher model. That is – the output of the softmax function of the logarithm of the teacher model. However, in many cases, the probability of the correct category of this probability distribution is very high, while the probability of all other categories is very close to zero. Therefore, it does not provide much information beyond the basic fact labels already provided in the dataset. To solve this problem, (Hinton et al., 2015). introduced the concept of "Softmax temperature" in 2015. The probability of the class is calculated from the logarithm.

as in Eq. 2.7

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{2.7}$$

where T is the temperature parameter, and the higher the value of T, the softer the probability distribution.

When T=1, we get the standard Softmax function. As T increases, the probability distribution generated by softmax functions becomes softer, providing more information about which classes the teacher finds to be more similar to the predicted classes. Hinton calls this ”dark knowledge” embedded in the teacher model, and it’s this dark knowledge that we transfer into the student model in the distillation process. When calculating the loss function and the teacher’s soft goal, we use the same t-value to calculate the student’s logarithmic Softmax. We call this loss ”distillation loss”.

2.6 Experimental Setup and Result Presentation

We first select ResNet He et al. (2016) as the model backbone, on teacher-student architecture, for our experiments and apply that on all the HAR datasets. Furthermore, ResNet-16 and ResNet-64 are arranged to assess the impact of reducing the number of layers on the HAR performance. To enable a fair comparison with the results presented in (Hammerla et al., 2016), we also use the weighted F1 score in Eq. 2.8 for the opportunity dataset. For the other two datasets, we use the mean F1 score as in Eq. 2.9.

$$F_w = 2 \sum_c \frac{N_c}{N_{total}} \frac{\mathrm{prec_c} \times \mathrm{recall_c}}{\mathrm{prec_c} + \mathrm{recall_c}} \tag{2.8}$$

$$F_m = \frac{2}{|c|} \sum_c \frac{\mathrm{prec_c} \times \mathrm{recall_c}}{\mathrm{prec_c} + \mathrm{recall_c}} \tag{2.9}$$

Where $prec_c$, $recall_c$ represent the precision and recall in label c, respectively, N_c refers to the number of samples in class c and N_{total} is the number of samples in the dataset

Experiments were run on a machine with four GPU cluster (4x Nvidia GTX 1080Ti) on Iridis 5. In the software used section, PyTorch version: 2.0.1 CUDA version: 11.7 (suitable for NVIDIA GPU acceleration). The different hyper-parameters explored in this work are: batch size = 128, initial learning rate = 0.1, weigh decay factor=0.1, patience=10, for all introduced architecture. For teacher-student architecture, SGD optimizer momentum = 0.9, alpha=0.5 for KD and generation = 5 for maximum students. For the XNOR-Net Adam optimizer, betas = (0.9, 0.999) eps = 1e-8.

The code for calculating the Expected Calibration Error is written as follows.

```
# Define the computation ECE
def compute_ece(confidence, ground_truth):
    Histogram, Acc, Conf = np.zeros(10), np.zeros(10), np.zeros(10)

    bins = np.linspace(0, 1, 11)  # 10 bins
    bin_indices = np.digitize(confidence, bins) - 1

    for i, bin_index in enumerate(bin_indices):
        Histogram[bin_index] += 1
        Conf[bin_index] += confidence[i]
        if confidence[i] == ground_truth[i]:
            Acc[bin_index] += 1
    valid_bins = Histogram > 0
    Acc[valid_bins] = Acc[valid_bins] / Histogram[valid_bins]
    Conf[valid_bins] = Conf[valid_bins] / Histogram[valid_bins]

    ECE = np.abs(Acc - Conf) * (Histogram / Histogram.sum())

    return ECE.sum()
```

Chapter 3

Born Again Network

3.1 Principle

(Furlanello et al., 2018) propose a new perspective based on the KD that is referred to as born-again networks. It employs a powerful teacher model to generate a soft label to supervise the light student structure with a hard label. In this case, the learning task is described by Eq. 3.1:

$$\theta_2^* = \operatorname*{argmin}_{\theta_2} \mathcal{L}(f(x, \operatorname*{argmin}_{\theta_1} \mathcal{L}(y, f(x, \theta_1))), f(x, \theta_2)) \tag{3.1}$$

where θ_2, θ_1 represent the student and teacher network parameters, respectively. Once the student network finishes training, it becomes a teacher to support the next generation of the student iteration.

Here we experiment with the teacher-student architecture. After the first model, the teacher is trained within hard-label, and we initialize another network (the student) with the same architecture. The student, however, now has a soft-label which is the output provided by the teacher, as well as a weighted hard-label. We examine how to combine these labels. Fig. 4.1 illustrates the structure of our method. We first experiment with the temperature parameter T as 1 and the weights w_1, w_2 in Eq. 3.2. These contain information on the output of both the teacher and the one-hot label.

$$\begin{split}\theta_2^* = w_1 \operatorname*{argmin}_{\theta_2} \mathcal{L}(f(x, \operatorname*{argmin}_{\theta_1} \mathcal{L}(y, f(x, \theta_1))), f(x, \theta_2)) \\ + w_2 \operatorname*{argmin}_{\theta_2} \mathcal{L}(y, f(x, \theta_2))\end{split} \tag{3.2}$$

In the backpropagation processes, the first derivative between the loss function and the logits output is given by Eq. 3.3:

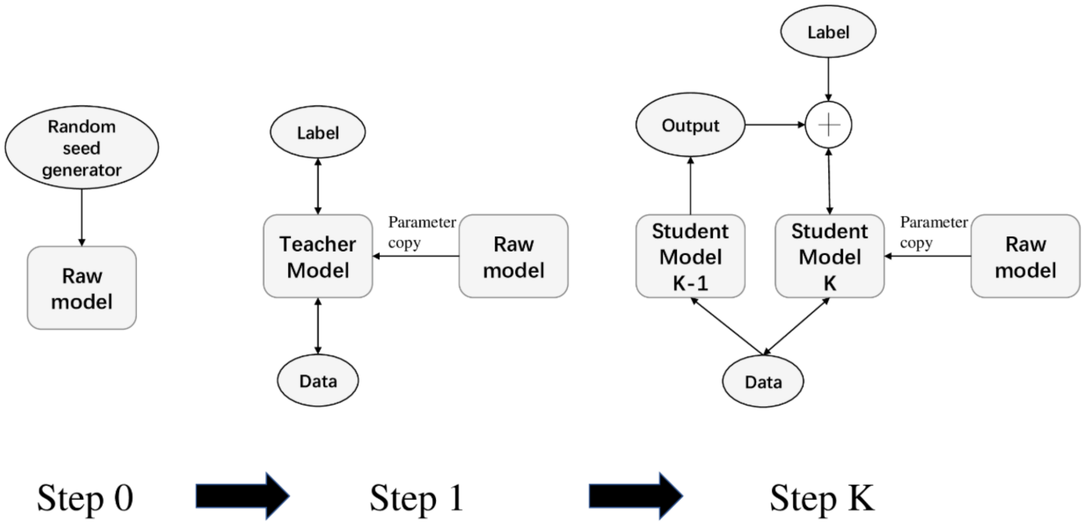


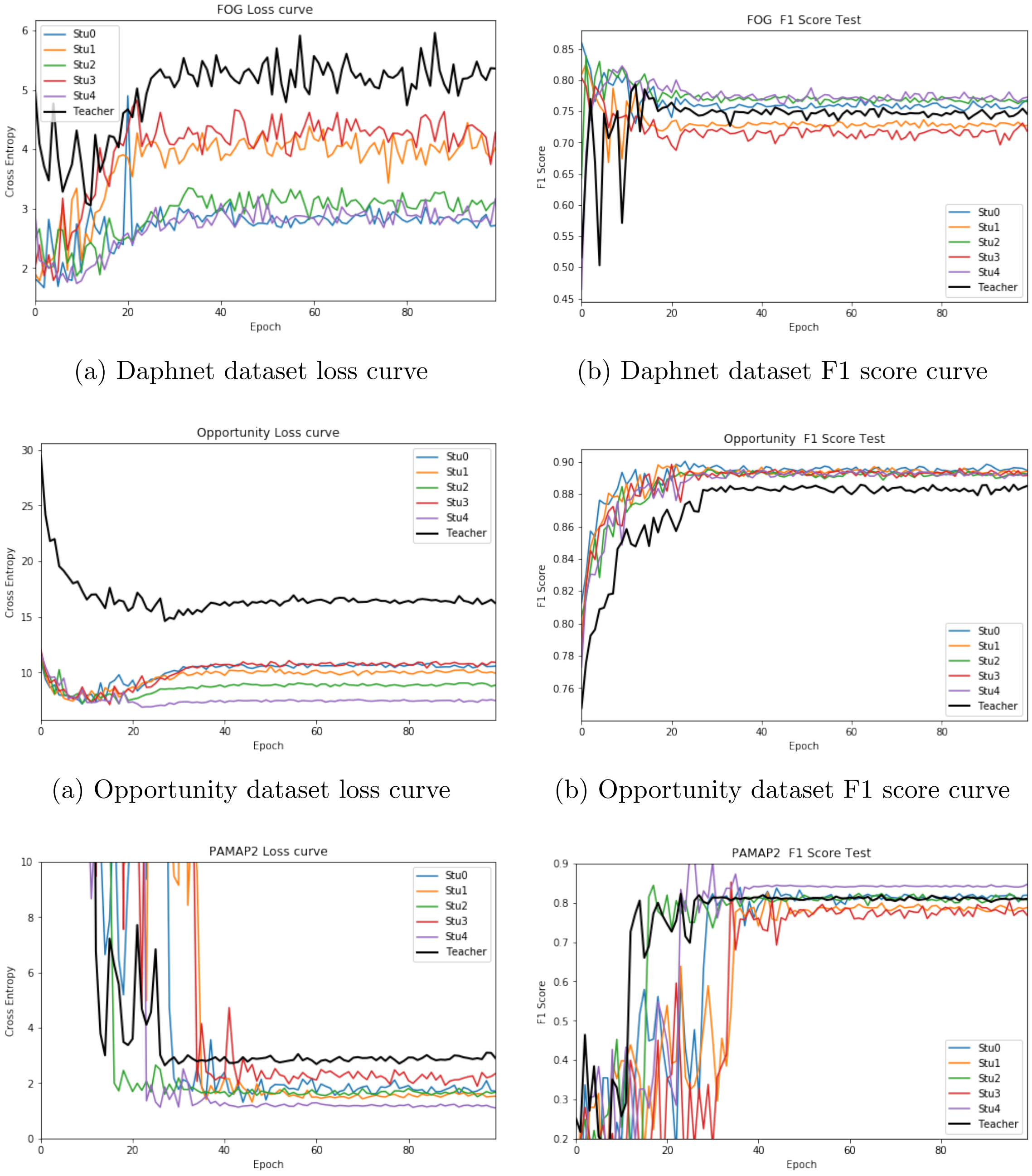
Figure 3.1: Graphical representation of the employed architecture. In the first step, we initialize the random-seed generator and a sample model. Then, the prototype model is trained in the usual way by copying the parameter from the sample model. In the next consecutive step, an identical model is created with the copied parameter and trained from the supervision of both the previous generation and the hard label. Note that the one-way arrows represent the forward propagation, and two-way arrows represent both forward and backpropagation.

$$\frac{\partial \mathcal{L}}{\partial l} = \frac{1}{2b} \sum_{n=1}^{b} \left(\frac{\exp(l_*)}{\sum_{i=1}^{n} \exp(l_i)} - 1 + \frac{\exp(l_i)}{\sum_{i=1}^{n} \exp(l_i)} - \frac{\exp(t_i)}{\sum_{i=1}^{n} \exp(t_i)} \right) \tag{3.3}$$

where t is the teacher's output. The first term is the cross-entropy with the hard label and the second term represents the cross-entropy with the soft label.
In the experimental setup, we test 1D ResNets on the datasets described in Section 4. We use the fixed evaluation test and training set to follow the previous works by (Edel and Köppe, 2016; Guan and Plötz, 2017; Reiss and Stricker, 2012) for comparison. We use the same architectures and parameters, except for the last dense layer in which a different number of classification targets are considered.

The details of the structure are as follows: 64 filters convolution layer with 15 kernel size at the beginning. This is followed by the basic ResBlocks (3,4,4,5) and a global-average pooling layer, SoftMax activation, and the dense layer. For the network, there are five generations trained and each generation is trained for 400 epochs using stochastic gradient descent with moment. We set the momentum to 0.9. and the learning rate is 0.1.

Following the idea of (Furlanello et al., 2018) for the BANs, here, we consider the same architecture for the student and teacher. Figure 3.2 shows the results on the considered HAR datasets, where at least one of the student networks achieves a better F1 score than that of the initial model (the teacher). This student is then selected to present the final result. In the opportunity dataset, all of the students achieve better F1 scores than the teacher. Similar results can be observed in Figure 3.5, where at least one student achieves better ECE performance than the teacher.



(a) Daphnet dataset loss curve

(b) Daphnet dataset F1 score curve

(a) Opportunity dataset loss curve

(b) Opportunity dataset F1 score curve

(a) PAMAP2 dataset loss curve

(b) PAMAP2 dataset F1 score curve

FIGURE 3.2: Loss and F1 score results on the HAR datasets, based on ResNet-16 architecture. A student performs better than the teacher

In Table 3.1 we present the number of parameters for each model and the number of required floating-point operations. As can be seen, due to the reduction of the number of layers in ResNet, the numbers of parameters and floating-point operations are approximately halved. In Table 3.2 we also present comparisons with the state-of-the-art methods. Since the datasets are imbalanced, we use F1-score for comparisons. The ensemble student model is selected, and as seen, its performance is better than the teacher.

Metric	Model	Daphnet	Opportunity	PAMAP2
Parameters	`ResNet-16`	1,974,850	2,040,130	2,016,130
	`ResNet-64`	7,889,400	8,140,520	8,044,520
FLOPs	`ResNet-16`	0.126 Gb	0.063 Gb	0.032 Gb
	`ResNet-64`	0.502 Gb	0.251 Gb	0.121 Gb

Table 3.1: The number of models' parameters (above) and the number of floating-point operations (below) on each dataset.

It is also seen the teacher-student networks achieve a better performance of generation of students. Nevertheless, by increasing the number of generations, the ECE is also increased. Considering that our application is focused on healthcare, an uncalibrated model might result in a lower confidence level for the medices.

Figure 3.3 and 3.4 illustrate how ECE relates to the classification performance of actions in Opportunity and PAMPA2 dataset. In Opportunity, it is less accurate recognition of actions with high similarities, such as switching on or off an appliance. However, with a wide variety of activities, our approach works well on the PAMAP2 dataset. These evidences may suggest that our method may not be so sensitive to temporal order, but works well on very different actions.

Metric	Daphnet	Opportunity	PAMAP2
LSTM baseline	-	0.659	0.756
LSTM Ensemble	-	0.726±0.008	0.854±0.026
DeepConvLSTM	-	-	0.917
Binarized-BLSTM	-	0.78±0.002	0.93±0.002
CNN	0.684± 0.122	0.894±0.104	0.937±0.071
LSTM-F	0.637±0.281	0.908±0.156	0.929±0.10
LSTM-S	0.76±0.297	0.912±0.168	0.882±0.128
b-LSTM-S	0.741±0.221	**0.927±0.172**	0.868±0.087
ResNet-16 Teacher	0.737±0.032	0.889±0.021	0.835±0.351
ResNet-16 Student-0	0.751±0.023	0.895±0.011	0.826±0.012
ResNet-16 Student-1	0.726±0.021	0.901±0.019	0.835±0.047
ResNet-16 Student-2	0.771±0.014	0.875±0.028	0.819±0.018
ResNet-16 Student-3	0.711±0.001	0.894±0.017	0.829±0.051
ResNet-16 Student-4	0.755±0.031	0.894±0.028	0.858±0.021
ResNet-16 Student-Ensamble	0.773±0.012	0.891±0.026	0.823±0.022
ResNet-64 Teacher	0.764±0.012	0.885±0.014	0.901±0.024
ResNet-64 Student-0	0.762±0.017	0.892±0.018	0.914±0.018
ResNet-64 Student-1	0.771±0.024	0.901±0.021	0.894±0.016
ResNet-64 Student-2	0.758±0.031	0.904±0.019	0.906±0.022
ResNet-64 Student-3	0.752±0.019	0.889±0.024	0.912±0.026
ResNet-64 Student-4	0.757±0.021	0.891±0.026	0.914±0.032
ResNet-64 Student-Ensamble	**0.778±0.017**	0.891±0.019	**0.941±0.024**

Table 3.2: Comparison with the state-of-the-art for each model and dataset using F1 scores. The results confirm that our method achieves the best results on Daphnet, and achieves high performance on the opportunity and PAMAP2 datasets.

Metric	Daphnet	Opportunity	PAMAP2
ResNet-16 Teacher	**5.287±1.235**	**5.264±0.158**	4.321±0.351
ResNet-16 Student-0	6.285±0.193	5.971±0.024	**3.922±1.052**
ResNet-16 Student-1	6.473±1.861	5.869±0.349	5.158±2.446
ResNet-16 Student-2	5.369±1.974	6.138±0.389	6.096±3.674
ResNet-16 Student-3	7.181±0.657	6.399±0.074	6.251±2.824
ResNet-16 Student-4	5.883±5.987	6.032±1.233	5.851±6.638

Table 3.3: The expected calibration error on HAR datasets, where the original teacher achieves the minimal ECE, except for the PAMAP2 with the first generation of the student, although the variance is higher.

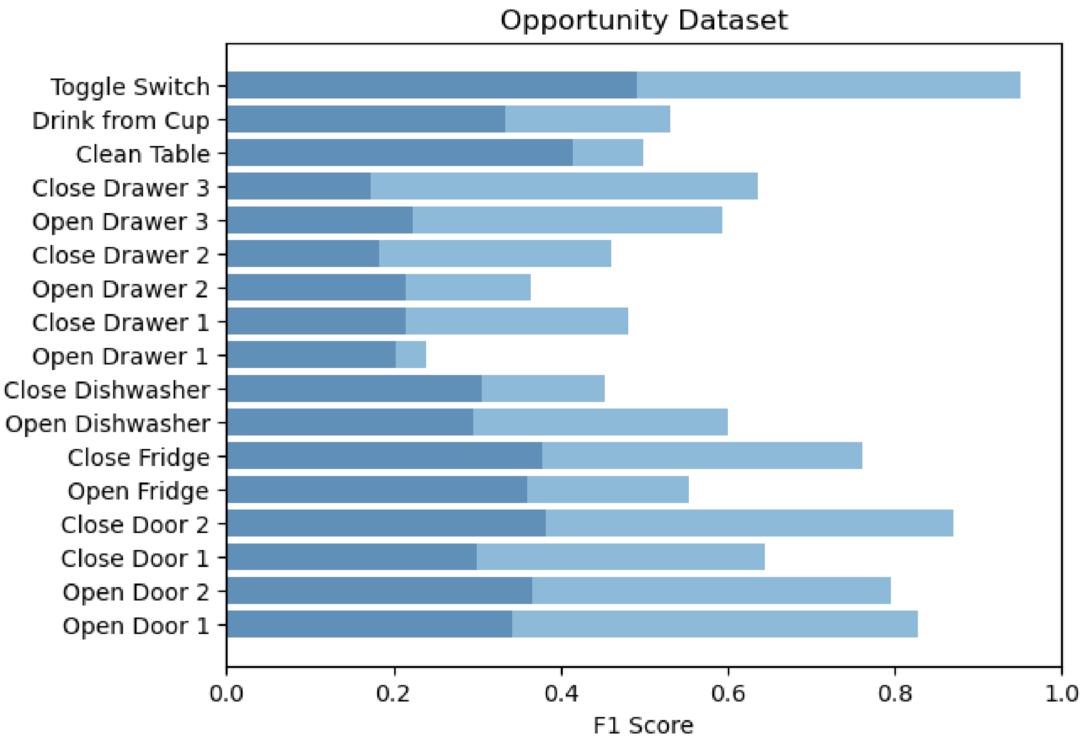


Figure 3.3: The confidence(Deep blue) and F1 score(sky blue) comparison on Opportunity dataset

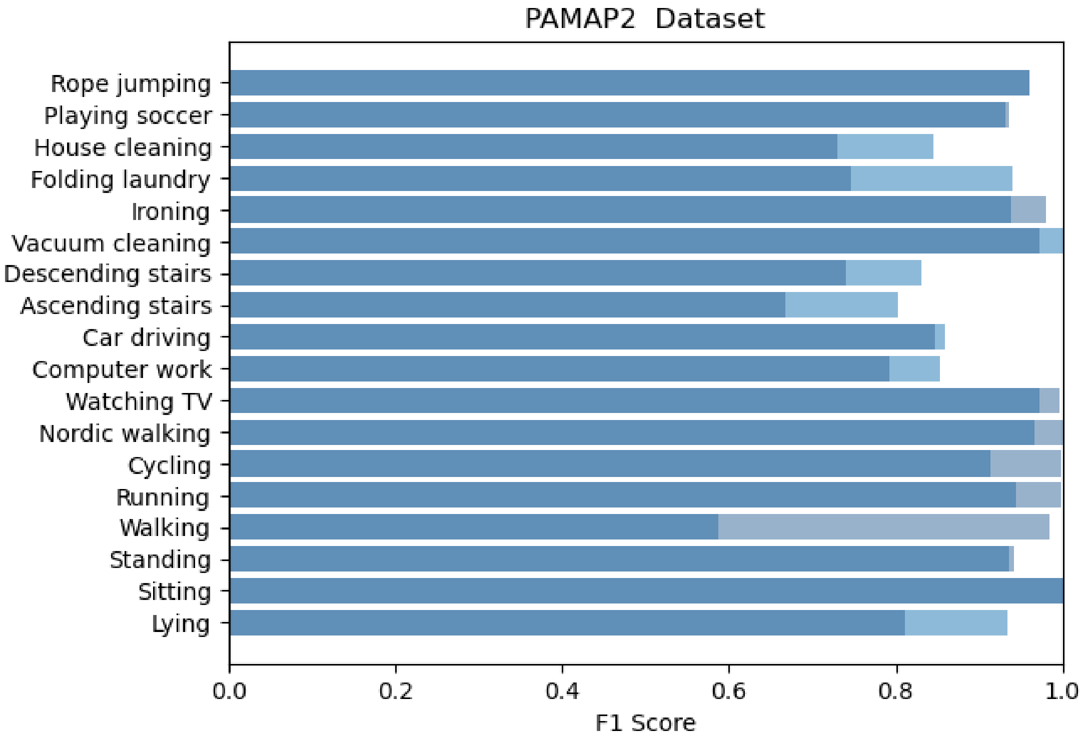
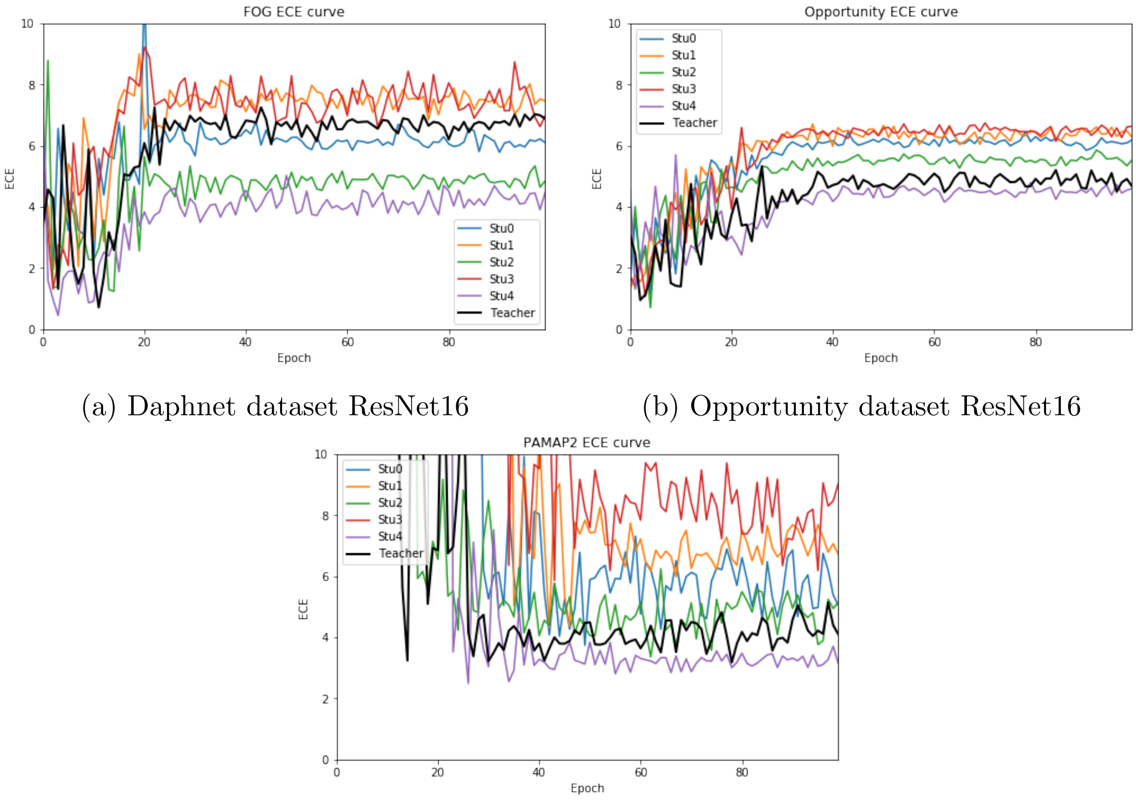


Figure 3.4: The confidence(Deep blue) and F1 score(sky blue) comparison on PAMPA2 dataset



(a) Daphnet dataset ResNet16

(b) Opportunity dataset ResNet16

(c) PAMAP2 dataset ResNet16

Figure 3.5: ECE results on the HAR datasets, based on ResNet-16 architecture. A student performs better than the teacher

Chapter 4

Binary Neural Networks

4.1 Deterministic and Stochastic Binarization

In this section, we present the details of the binarization function and its impact on the computation of the parameter gradient on backpropagation. The BNN constrains the weights and the activation function to either +1 or 1. Those two values are suitable for FPGA hardware implementation. In general, the float-32 formula is widely used to save both weights and activation. To binarize these parameters, two types of functions are presented as in Eq. 4.1 and Eq. 4.2 for deterministic and stochastic binarization.

$$x^b = \begin{cases} +1 & \text{if} \quad x \geq 0, \\ -1 & \text{otherwise.} \end{cases} \tag{4.1}$$

where x^b is the binarized variable and x is the float-32 value.

$$x^b = \begin{cases} +1 & \text{withprobability} \quad p = \delta(x), \\ -1 & \text{withprobability} \quad 1 - p. \end{cases} \tag{4.2}$$

The stochastic binarization is more complicated, and $\delta(x)$ is the hard sigmoid function as shown in Eq. 4.3

$$\delta(x) = \text{clip}(\frac{x+1}{2}, 0, 1) = \max(0, \min(1, \frac{x+1}{2})) \tag{4.3}$$

The stochastic binarization is often more appealing than that of the deterministic function. However, stochastic binarization is more challenging because it needs to generate

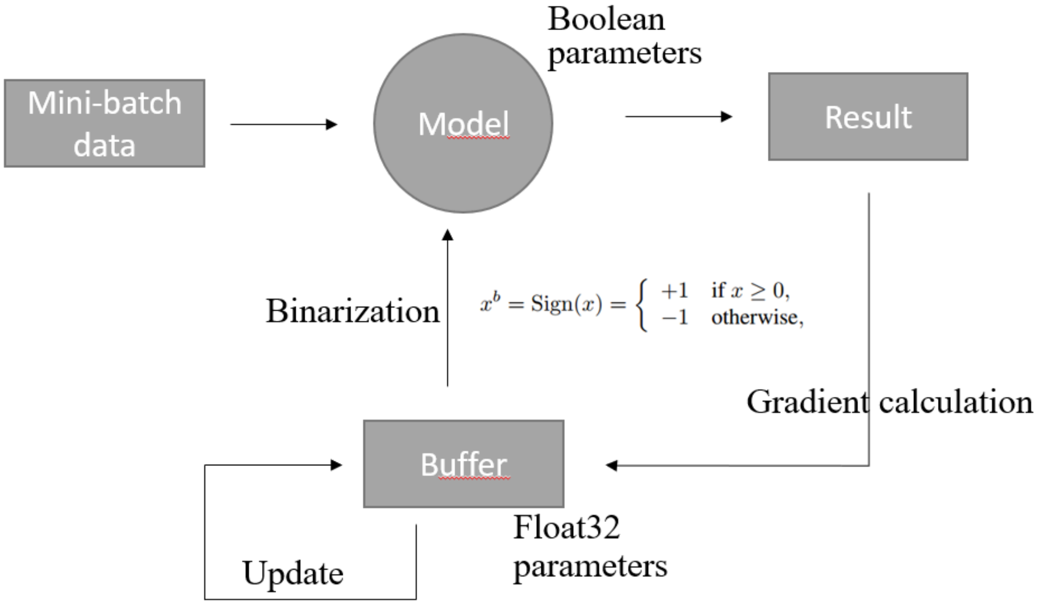


Figure 4.1: Graphical representation of the Straight-through estimator: the extra buffer is employed to temporarily store the float-32 type of weights, also with the gradients. There is no change with forwarding propagation, but the weights are binary after gradients are updated through the backpropagation

random bits, which is computationally complex. Thus, here we use the former for quantification.

4.1.1 Straight-through estimator

In the quantification step with a deterministic function, the derivative of the sign function is zero almost everywhere. This makes quantization incompatible with backpropagation as the gradients of weights are accumulated in real values. It is, therefore, essential to maintain sufficient resolution for the accumulator to maintain the high precision results.

Table 4.1 shows the comparison between the regular network training progress, BNN, and the combination of the BANS with BNN. As seen, for the BNN, the network size is significantly reduced (90%) with a reasonable reduction of the F1 score (5%). It is also seen that combining BANS and BNN and selecting the ensemble results from students after 5 iterations, results in a slight increase of their corresponding F1 scores, while their ECE values are further decreased. This means that in this setting, the calibration in our model outperforms the other models.

Method	Metric	Daphnet	Opportunity	PAMAP2
BANS	`F1 score`	**0.737±0.032**	**0.889±0.001**	**0.895±0.351**
	`ECE`	0.09±0.02	0.08±0.02	0.08±0.01
	`Size`	7.44 MB	7.96 MB	7.77 MB
XNOR-Net	`F1 score`	0.692±0.062	0.842±0.031	0.734±0.521
	`ECE`	0.07±0.02	0.07±0.01	0.07±0.02
	`Size`	**0.98 MB**	**1.03 MB**	**0.96 MB**
BANS + XNOR-Net	`F1 score`	0.712±0.052	0.853±0.001	0.753±0.272
	`ECE`	**0.06±0.01**	**0.06±0.01**	**0.06±0.01**
	`Size`	**0.98 MB**	**1.03 MB**	**0.96 MB**

Table 4.1: Comparisons between the regular teacher network training (above), BNN (middle), and the combination of BANS and BNN (below).

4.2 Transfer learning Results

Table 4.2 illustrates the result of transfer learning for the datasets. For the Opportunity dataset across users, with 15 channels from the accelerometers. We also try to transfer across the datasets, from Opportunity to PAMAP2 with sensor signal at the same location as the participant. The results indicate a great potential for transfer learning performance across datasets in HAR. However, the FOG dataset only has 3 IMU sensors on the ankle, leg, and trunk. However due to the large dimensional difference, training is not possible, we tried filling missing dimensions with zeros, however this would not lead to network convergence. In the experiment, due to the number sample being insufficient, we performed data augmentation by oversampling the windows, increasing data by 4 times. The result illustrates that with transfer learning, the performance decreased approximately 13-15%. Transfer learning still works in binarization, even if the location of the sensors on the data has changed, which suggests that sensor shift migration learning has some potential for HAR. Moreover, due to the noise introduced by binarisation, the ECE instead improves, indicating that the model is more calibrated.

Method	Metric	Result

Without transfer
Without binarization
	`F1 score`
	`ECE`
		0.911±0.014
		0.08±0.01

Without transfer
With binarization
	`F1 score`
	`ECE`
		0.831±0.034
		0.07±0.01

With transfer
Without binarization
	`F1 score`
	`ECE`
		0.779±0.067
		0.06±0.02

With transfer
With binarization
	`F1 score`
	`ECE`
		0.729±0.037
		0.05±0.02

TABLE 4.2: The comparison between with and without transfer learning and binarization, from Opportunity to PAMPA2 dataset.

Chapter 5

Conclusions

Our research has thoroughly examined the application of soft labels and demonstrated its effectiveness in enhancing classification performance in Human Activity Recognition (HAR), confirming that our proposed teacher-student network model achieves great performance, In particular, it achieves an F1 performance of 0.778 ± 0.017, surpassing the results reported in (Hammerla et al., 2016) by 0.684 (CNN), LSTM-F (0.637), LSTM-S (0.76) and b-LSTM-S (0.741) on the Daphnet dataset. Moreover, it achieves 0.941 ± 0.024, surpassing the results reported in LSTM baseline (Guan and Plötz, 2017) by 0.756, LSTM Ensemble (0.854), DeepConvLSTM (Ordóñez and Roggen, 2016) by 0.917, Binarized-BLSTM (Edel and Köppe, 2016) by 0.93, CNN (Hammerla et al., 2016) (0.937), LSTM-F (0.929), LSTM-S (0.882) and b-LSTM-S (0.868) on the PAMAP2 datase. However, in the Opportunity dataset, it does not achieve the highest score, specifically at 0.891 ± 0.019. Performance inferior to (Hammerla et al., 2016) CNN's (0.894), LSTM-F (0.908), LSTM-S (0.912) and the b-LSTM-S (0.927). Although it beats LSTM baseline (Guan and Plötz, 2017) by 0.659, LSTM Ensemble (0.726) and Binarized-BLSTM (Edel and Köppe, 2016) by 0.78.

A key strategy in our methodology involved using the Born again networks (BANS) technique to aggregate student models and finalize the selection of optimal parameters.

Moreover, we incorporated Expected Calibration Error (ECE) into HAR, underlining the crucial role of model calibration in applications where the F1 score is not the sole performance measure. With our proposed method, it's possible to train an ensemble network to achieve superior performance.

The binary Neural Networks (BNN) proved to be highly efficient in processing HAR datasets, reducing the model size by an impressive 90%. When BNN was paired with BANS, the F1 scores experienced a slight decrease, but ECE saw a substantial reduction.

As for transfer learning performance across different datasets as well as within the same dataset, the results were quite satisfactory. However, it's worth noting that the sensor placement can impose significant constraints.

5.1 Future work

Following this line of research, our future research plans are as the following:
Attention maybe is our need : The Attention mechanism was proposed by Vaswani et al. (2017) in 2017 and has been widely applied in various areas of deep learning in recent years, such as in computer vision for capturing perceptual fields in images, or the NLP for locating key tokens or features. Google team's (Devlin et al., 2018) proposed BERT algorithm for generating word vectors has achieved a significant improvement in the effectiveness of 11 tasks in NLP, which is the most exciting news in deep learning in 2018. The most important part of the BERT algorithm is the concept of Transformer proposed in this paper, in which the traditional CNN and RNN are abandoned, and the entire network structure is composed entirely of the Attention mechanism. In other words, the Transformer only consists of a self-attention mechanism and feed forward Neural network. This may also apply to our research by introducing a transformer structure in our proposed architecture. One of the motivation is that, the attention mechanism solves the problem of very long-time series issues, especially in NLP. Our current research finds that the Opportunity dataset, it is not sensitive to very long time series signals. This may be helpful for our further research.

Neural Architecture Search : Convolutional neural networks are usually developed with a fixed resource. They are scaled up to obtain better performance, if more resources are available, by increasing the network depth, width, and input resolution. However, the combination space is significantly large that it is difficult to tune by manual. Among them, the most famous achievement belongs to Efficientnet(Tan and Le, 2019). At the beginning of its release, it amazed the entire CV class with its various SOTA results in image classification, and it was completed fast and accurately. From the NAS, the optimal set of parameters: depth, width, resolution can be obtained. We would like to research how NAS works on our BNN model to get better performances. Also, we build the end-to-end network to produce the soft label, which is highly correlated with ECE we introduced. An optimal parameter network that may help us explain the interpretability of neural networks, better understand the role of ECE in it and the nature of neural network black box.

Appendix A

OU-ISIR Wearable Sensor-based Gait Challenge

A.1 Deep Convolutional BLSTM

This appendix presents a report of our participation in the international competition, the "OU-ISIR Wearable Sensor-based Gait Challenge". Our task in the competition was to predict the age and gender of subjects based on the data collected from wearable sensors. To accomplish this task, we employed a residual neural network (ResNet) with bidirectional long short-term memory (BLSTM) blocks in combination with multitask learning, which enabled us to achieve favorable results and secure the runners-up position in the competition.

This project was a collaborative effort with Fangfei Liu and Takuya Yaguchi. My specific contributions included the design of the backbone architecture of the model and the implementation of the gradient normalization (GradNorm) (Chen et al., 2018) algorithm. To separate the dataset and ensemble the predictions, we utilized 5-fold cross-validation. The dataset comprised a total of 610 subjects for model training and an additional 150 for validation. The network was trained over 500 epochs using the Adam optimizer. The initial learning rate was set to 0.1, and was decreased by a factor of 10 at the 3rd, 100th, and 200th epochs

Figure A.1 illustrates the structure of our proposed deep neural network, which is inspired by (Ordóñez and Roggen, 2016). We have modified the original architecture by using a bidirectional LSTM (BLSTM) instead of a simple LSTM. This modification was made to better handle the mixed flatland-, uphill-, and downhill-walking environments present in the dataset, to which the bidirectional structure is more sensitive. The effectiveness of the bidirectional structure has also been proven in the field of Natural Language Processing (NLP), particularly for handling sequences. Moreover, our model accepts multiple inputs and has been adapted to perform multitasking for both age and

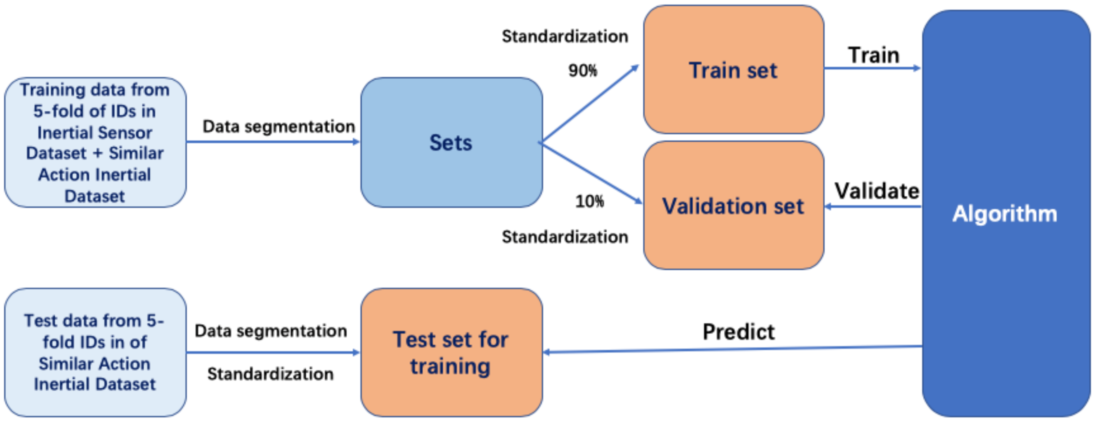


Figure A.1: Data pre-processing steps

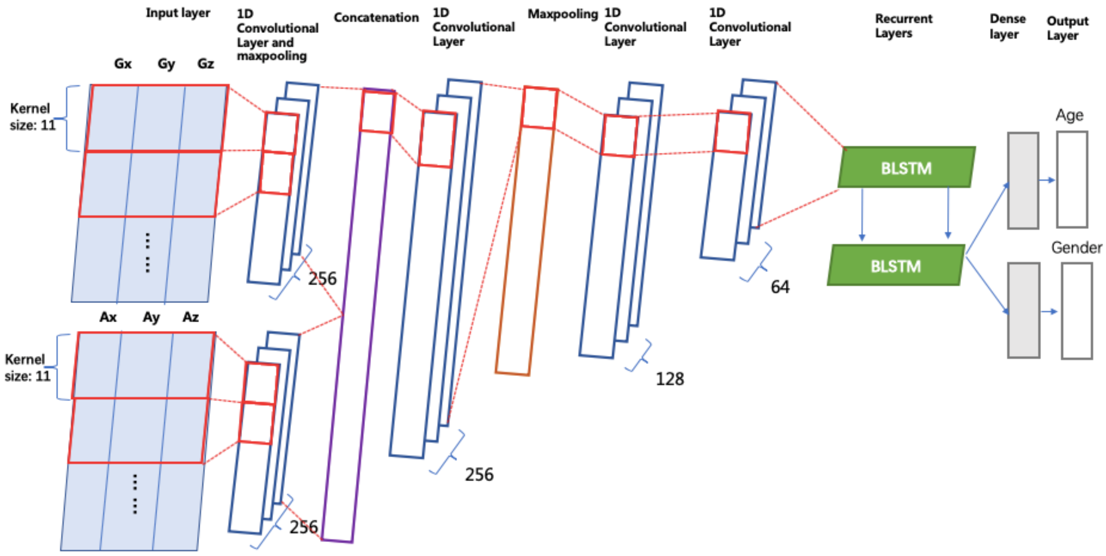


Figure A.2: The structure of the multi-task deep convolutional BLSTM

gender prediction tasks.

The input to the network is divided into two parts: angular velocity (Gx, Gy, Gz) and acceleration (Ax, Ay, Az), both of which can capture essential connection information. These inputs are concatenated after passing through the first convolutional layer. Subsequently, the data is processed by three additional convolutional layers and two bidirectional LSTMs. To enable multi-output prediction, the network branches into two paths, each leading to an output. Before reaching the output layer, the data passes through a fully connected layer. Additionally, we have implemented batch normalization and max-pooling operations after each convolutional layer.

Our model operates as a multi-task network that simultaneously predicts gender and age. The gender prediction task is considered a binary classification problem and, consequently, we adopt binary cross-entropy as its loss function. On the other hand, for the age estimation task, we use mean absolute error (MAE) as the loss function. Typically,

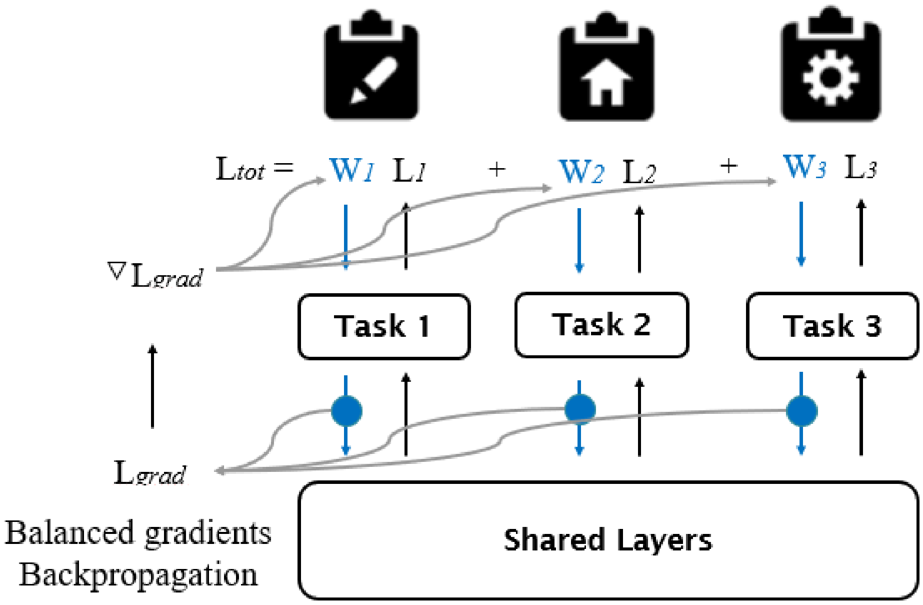


Figure A.3: The structure of the Gradient Normalization

the MAE loss is much larger than the binary cross-entropy loss. To balance these different magnitudes, we manually assign weights of 5 and 1 to the binary cross-entropy and MAE losses, respectively.

Multitask learning presents a particular challenge in terms of balancing the weights of different tasks for the network parameters, ensuring features can be effectively shared across tasks. Additionally, within a Balanced Academic Network (BAN), the student model needs to maintain a balanced loss between the teacher model's predictions and the hard labels. GradNorm (Chen et al., 2017) provides a novel solution to this problem. It automatically balances training in deep multitask models by dynamically tuning the gradient magnitudes, as illustrated in Figure A.2.

A.2 Competition Result

We concluded the competition as the runners-up. Our primary oversight was not taking into account the sensor orientation, which significantly affects the performance of the network. As the sensor orientation differed between the training and test sets, this information was lost. Despite this, we achieved reasonably good results, with 75% accuracy on gender prediction and a mean absolute error of 7 on age estimation. The results of the competition are illustrated in Table A.1.

Category	Gender(Prediction error)	Age(Mean absoulte error)
`Ours`	30.41	7.54
`Champion`	24.22	5.39
`Mean`	39.71	9.81

Table A.1: GAGP2019 competition result

Bibliography

Al-Shedivat, M., Wilson, A. G., Saatchi, Y., Hu, Z., and Xing, E. P. (2017). Learning scalable deep kernels with recurrent structure. *The Journal of Machine Learning Research*, 18(1):2850–2886.

Alharbi, F. and Farrahi, K. (2018). A convolutional neural network for smoking activity recognition. In *2018 IEEE 20th International Conference on E-Health Networking, Applications and Services (Healthcom)*, pages 1–6. IEEE.

Bächlin, M., Plotnik, M., Roggen, D., Giladi, N., Hausdorff, J. M., and Tröster, G. (2010). A wearable system to assist walking of parkinson's disease patients. *Methods of Information in Medicine*, 49(01):88–95.

Bachlin, M., Plotnik, M., Roggen, D., Maidan, I., Hausdorff, J. M., Giladi, N., and Troster, G. (2010). Wearable assistant for parkinson disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):436–446.

Baghezza, R., Bouchard, K., Bouzouane, A., and Gouin-Vallerand, C. (2020). Activity recognition in the city using embedded systems and anonymous sensors. *Procedia Computer Science*, 170:67–74.

Banos, O., Galvez, J.-M., Damas, M., Pomares, H., and Rojas, I. (2014). Window size impact in human activity recognition. *Sensors*, 14(4):6474–6499.

Bodor, R., Jackson, B., and Papanikolopoulos, N. (2003). Vision-based human tracking and activity recognition. In *Proc. of the 11th Mediterranean Conf. on Control and Automation*, volume 1. Citeseer.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Bucilu, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541. ACM.

Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. (2017). Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *ArXiv Preprint ArXiv:1711.02257*.

Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. (2018). Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR.

Chetty, G. and White, M. (2016). Body sensor networks for human activity recognition. In *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 660–665. IEEE.

Davarci, E., Soysal, B., Erguler, I., Aydin, S. O., Dincer, O., and Anarim, E. (2017). Age group detection using smartphone motion sensors. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 2201–2205. IEEE.

Delgado-Escaño, R., Castro, F. M., Cózar, J. R., Marín-Jiménez, M. J., and Guil, N. (2019). An end-to-end multi-task and fusion cnn for inertial-based gait recognition. *IEEE Access*, 7:1897–1908.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Edel, M. and Köppe, E. (2016). Binarized-blstm-rnn based human activity recognition. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE.

Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., and Anandkumar, A. (2018). Born again neural networks. *ArXiv Preprint ArXiv:1805.04770*.

Golestani, N. and Moghaddam, M. (2020). Human activity recognition using magnetic induction-based motion signals and deep recurrent neural networks. *Nature communications*, 11(1):1–11.

Guan, Y. and Plötz, T. (2017). Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):11.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*.

Hammerla, N. Y., Halloran, S., and Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. *ArXiv Preprint ArXiv:1604.08880*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *ArXiv Preprint ArXiv:1503.02531*.

Jiang, X., Osl, M., Kim, J., and Ohno-Machado, L. (2012). Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274.

Kang, H., Lee, C. W., and Jung, K. (2004). Recognition-based gesture spotting in video games. *Pattern Recognition Letters*, 25(15):1701–1714.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.

Kumari, P., Mathew, L., and Syal, P. (2017). Increasing trend of wearables and multimodal interface for human activity monitoring: A review. *Biosensors and Bioelectronics*, 90:298–307.

Lara, O. D. and Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209.

Lara, O. D. and Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & tutorials*, 15(3):1192–1209.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lu, Y., Wei, Y., Liu, L., Zhong, J., Sun, L., and Liu, Y. (2017). Towards unsupervised physical activity recognition using smartphone accelerometers. *Multimedia Tools and Applications*, 76(8):10701–10719.

Müller, R., Kornblith, S., and Hinton, G. E. (2019). When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4694–4703.

Naeini, M. P., Cooper, G., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Ordóñez, F. J. and Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115.

Reiss, A. and Stricker, D. (2012). Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pages 108–109. IEEE.

Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Förster, K., Tröster, G., Lukowicz, P., Bannach, D., Pirkl, G., Ferscha, A., et al. (2010). Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pages 233–240. IEEE.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *ArXiv Preprint ArXiv:1706.05098*.

Sagha, H., Digumarti, S. T., Millán, J. d. R., Chavarriaga, R., Calatroni, A., Roggen, D., and Tröster, G. (2011). Benchmarking classification techniques using the opportunity human activity dataset. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pages 36–40. IEEE.

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

Steven Eyobu, O. and Han, D. (2018). Feature representation and data augmentation for human activity classification based on wearable imu sensor data using a deep lstm neural network. *Sensors*, 18(9):2892.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wang, W., Liu, A. X., Shahzad, M., Ling, K., and Lu, S. (2015). Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st annual international conference on mobile computing and networking*, pages 65–76.

Yao, S., Hu, S., Zhao, Y., Zhang, A., and Abdelzaher, T. (2017). Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web*, pages 351–360. International World Wide Web Conferences Steering Committee.

Yim, J., Joo, D., Bae, J., and Kim, J. (2017). A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141.