“We Rock the Hizzle and Stuff” hints on how to write a nice research essay

Bryan Lucchetta†, Luca Parolari‡

Abstract—**Future vehicular communication networks call for new solutions to support their capacity demands, by leveraging the potential of the millimeter-wave (mm-wave) spectrum. Mobility, in particular, poses severe challenges in their design, and as such shall be accounted for. A key question in mm-wave vehicular networks is how to optimize the trade-off between directive Data Transmission (DT) and directional Beam Training (BT), which enables it. In this paper, learning tools are investigated to optimize this trade-off. In the proposed scenario, a Base Station (BS) uses BT to establish a mm-wave directive link towards a Mobile User (MU) moving along a road. To control the BT/DT trade-off, a Partially Observable (PO) Markov Decision Process (MDP) is formulated, where the system state corresponds to the position of the MU within the road link. The goal is to maximize the number of bits delivered by the BS to the MU over the communication session, under a power constraint. The resulting optimal policies reveal that adaptive BT/DT procedures significantly outperform common-sense heuristic schemes, and that specific mobility features, such as user position estimates, can be effectively used to enhance the overall system performance and optimize the available system resources.**

Index Terms—**Human Activity Recognition, Convolutional Neaural Networks, Autoencoders**

I. Introduction

Recognition of the activity that a user is currently doing is important not only for health care monitoring or security concerns, but also for developing mobile apps that are able to use this information to improve our day life.

Developing mobile apps that are capable of tracking user activities that are used ”in the wild” contexts, leads important challenges that need to be tackle down. As stated in [1] many variables are involved both coming from users and smartphones. Users are demographically different (age, stature, weight, ...) and perform activities in different ways, using their devices in their own ways. Devices instead share among them different operating-systems, hardware and sensing capabilities.

Different solution were proposed in literature to perform Human Activity Recognition (HAR)[citare tutti i paper] capable of reaching high performance, but when they are used in real scenarios the performance usually decrease a lot.

TODO: Citare inoltre il paper [2] recente che dice parecchie cose sugli algoritmi di learning attuali e tutti i problemi che ci sono in questo caso

†Department of Mathematics, University of Padova, email: {bryan.lucchetta}@studenti.unipd.it

†Department of Mathematics, University of Padova, email: {luca.parolari}@studenti.unipd.it

Special thanks / acknowledgement go here.

II. Related Work

Different solution were pubblished in litterature to perform HAR. Many deep learning and machine learning has been used, starting from PCA CITE and SVM CITE, through.

Se si vuole fare una introduzione piu’ ampia si puo’ parlare dei lavori sulle CNN e l’uso di stacked autoencoders TODO Luca!

We start our work from [3] e spiegare le buone caratteristiche che ha questa rete e i contributi che hanno dato.

Spiegare perche’ questa rete non e’ sufficiente in contesti piu’ reali, come la rotazione dello smartphone e i problemi dei bias dei sensori, citando il paper [4].

Questi paper sono stati migliorati combinando i contributi di entrambi, abbiamo aggiunto un sistema per rendere il modello predittivo indipendente dalla posizione e rotazione dello smartphone e abbiamo aggiunto features piu’ roboste al modello presentato in [3]

III. Processing Pipeline

The task of HAR in real use case scenarios is a difficult task, and many aspects need to be considered when these applications are brought in a mobile environment. As reported in [1] the three major types of heterogeneties which yield impairments in HAR are:

- **Sensor Biases**: To keep the overall cost of a smartphone low, low costs accelerometer and gyroscope sensor are used, yielding a poor-calibrated, inaccurate an of limited granularity and range acquired signals. So, among this type of sensors we could observe differences in precision, resolution, range and also biases. Usually an initial sensors calibration are made by smartphones manufactures, but due to rotation or misalignment of the sensor to the circuit board of the final product, this could introduce errors. Furthermore, if a device experience shock, e.g falling on the ground, the sensor can be misaligned causing unwanted biases.
- **Sampling Rate Heterogeneity**: Often popular smartphones vary in terms of the default and supported sampling frequencies for accelerometer and gyroscope sensor. In the dataset TODO riportare link used for this experiment for example we are dealing with smartphone where the sampling frequency varies from 50Hz to 200Hz. See Fig. TODO so the actual number of devices used in this dataset, with their corresponding sampling frequencies.
- **Sampling Rate Instability**: This phenomenon is specific to a single device and regards the regularity of

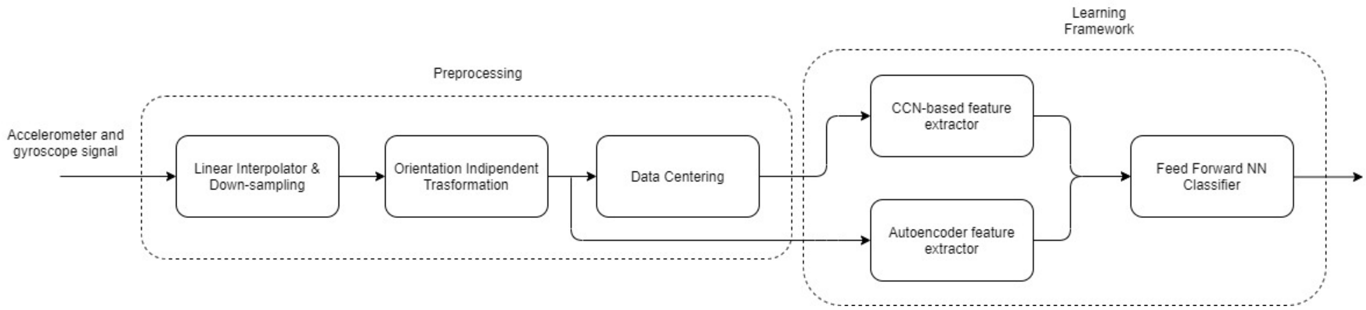

Fig. 1: Processing pipeline

time span between successive measurements. Different factors could accentuate this problem, including heavy multitasking or high I/O load in the mobile device. Multitasking effect in particular is a major problem: smartphone usually prioritizes among various running tasks and doing so could extremely affects the sensor sampling of a HAR application running on the device. In our collected dataset with a 100 Hz sampling rate,for example we observe a time span between consecutive measurement of TODO, even if the smartphone was hold in *airplane mode* to reduce at the minimum this effect. The Fig. TODO shows the amount of different time-span present in our dataset.

Furthermore, if we considered a real use case scenarios of a HAR mobile application we must also consider that smartphones can be positioned and oriented in different ways in human body. For example a smartphone could lay in trouser pockets (back and front) or maybe inside accessories like in a pouch or in a bag with different orientations. These different initial model settings have huge effects in prediction accuracy of an HAR predictor, especially if the model has been trained on a dataset that consist of activity measurement coming from only one fixed position and orientation of the smartphone, as is usually the case dealing with dataset collected in a controlled environment that could be found on Internet.

To tackle all these problems we decided to adopt in our preprocessing pipeline as show in Fig. TODO, 3 main blocks.

The first block called Linear Interpolator is in charge of mitigate the problems regarding the Sampling Rate Heterogeneity and Sampling Rate Instabilities as discussed previously. It main purpose is to down-sample the input data to a fixed sampling rate, in our case 50 samples/second.

The second block called Orientation Independent Transformation is used to represent data coming from different orientation of the smartphone in a rotation independent space. In this way all the signals are projected in a new space whose orientation is independent of that of the smartphone and aligned with gravity and the direction of motion. In this way an user can place his smartphone in whatever position he or she wants, mitigating the problem of different position and rotations of smartphone, as we will see in the Experiment section TODO reference.

Our last block consists of a data centering operation, applied for centering signals among y-axis that are presented only for the Covolutional Layer and not for the Autoencoder block. The reason of this choice would be clear in the section TODO. As reported in [3], time series centering standardize the input data, making the task for the CNN easier. Data normalization instead must be avoided because does not help in this situation since it significantly distorts time series shape, removing magnitude information which is critical for activities differentiation.

After the preprocessing pipeline we adopt a novel Learning framework. It is composed of CNN augmented with features coming from an autoencoder. As discussed in [3] CNNs learns filters that are applied to small sub-regions of the data, and therefore they are able to capture local data pattern and their variations. Additionally, due to a small number of connections and high parallelism the amount of computations and running time of CNNs is significantly lower compared to other deep learning algorithm. This yield these model perfect for real-time HAR apps, where these models could also run in a restrict environment as one like smartphones where computation resources are limited. The only drawback of CCNs is that they fall behind in capturing global properties of the signal, and as proposed in [3] they resolve this problem by augmenting CNNs with some basic statistical features that comprise this aspects of the data. But as opposed to what done in this latter work, where they used manual extracted features, we decided to opt for a autoencoder features extractors which can provide more robust feature. For this reason we train an auto-encoder separately on the training data and then use the encoder part to augment the CNN features used for the last classification Feed Forward Neural Network.

IV. Signals and Features

Parlare del nostro dataset usato Heterogenity, cioe come sono strutturati i nostri dati partenza.

A. Dataset & Meausurement Setup

Parlare di come abbiamo splittato il dataset, quindi escludendo gli utenti a e b per fare in modo di testare le performance cambiando compltamente utente.

Introduzione e spiegazione del dataset eterogeneo, preso da: riportare link. e riportare come abbiamo tirato fuori le time window con overlapping window!

Accenare dell'acquisizione di un nostro dataset alla stessa maniera, a 100 HZ in varie posizioni per testare poi anche tutto il dataset!

B. Signal preprocessing

In questo caso si va nel dettaglio (con formule e forse se neccessario anche grafici) della parte di preprocessing (parte trattegiata nel mio schema come preprocessing):

Notation: With $\boldsymbol{x}$ we mean a column vector $\boldsymbol{x} = (x_1, x_2, ..., x_n)^T$. With $||\boldsymbol{x}||$ we mean the L2-norm operator and with $\bar{x} = \sum_{i=1}^{n} x_i/n$ the mean of a vector. With $\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^T \boldsymbol{y}$ we mean the inner product of two vectors. With $\vec{\boldsymbol{x}}$ we mean a 3D vector $\vec{\boldsymbol{x}} = (x_1, x_2, x_3)^T$ and with $\hat{\boldsymbol{x}}$ the corresponding 3D versor $\hat{\boldsymbol{x}} = \vec{\boldsymbol{x}}/||\vec{\boldsymbol{x}}||$. For any two 3D vector $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{y}}$ we indicate their cross-product as $\vec{\boldsymbol{x}} \times \vec{\boldsymbol{y}}$. We represent with $\boldsymbol{a_x}, \boldsymbol{a_y}, \boldsymbol{a_z}$ the x, y and z components of the accellerometer signal, as well as the gyroscope signal is represented with $\boldsymbol{g_x}, \boldsymbol{g_y}, \boldsymbol{g_z}$. We define also matrices with uppercase and bold letters. For example to define a $3 \times n$ matrix composed of 3 vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n$ we use the notation $\boldsymbol{M} = [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}]^T$

Linear Interpolation & Down-sampling

Orientation Indipendent Transform

To project the signals acquired during an activity in a new space wich is indipendent of the rotation of the smartphone, 3 orthogonal versors need to be found. We decided to adopt the technique proposed in [5], although many other works proposed a similar solution as in [6], [7]. In summary we have to find these 3 orthogonal versor namely vertical versor $\hat{\boldsymbol{v}}$, horizontal versor $\hat{\boldsymbol{h}}$ and lateral versor $\hat{\boldsymbol{l}}$. As the name suggest the vertical versor is aligned with user torso pointing up, the horizontal versor is aligned with the direction of motion, pointing foraward, and the later versor tracks lateral movements and it is othogonal to the other two.

Starting with the vertical versor we need to find where the gravity vector $\vec{\boldsymbol{p}}$ ly in the original space. Although the gravity vector is a constant vector in stationary conditions, during a user activity it continuosly changes in the original cordinate system of the smartphone, therefore we are only able to consider the mean direction of the gravity within the current user activity. So, to estimate it we must considered only the acceleremoter signal: $\vec{\boldsymbol{p}} = (\bar{a_x}, \bar{a_y}, \bar{a_z})^T$ and we now could find the vertical versor with $\hat{\boldsymbol{v}} = \vec{\boldsymbol{p}}/||\vec{\boldsymbol{p}}||$

This is the first axis of our new space, and we can project the datas onto this new versor to obtain the first component axis. To do we define the acceleration matrix $\boldsymbol{A} = [\boldsymbol{a_x}, \boldsymbol{a_y}, \boldsymbol{a_z}]^T$ and the gyroscope matrix $\boldsymbol{G} = [\boldsymbol{g_x}, \boldsymbol{g_y}, \boldsymbol{g_z}]^T$. Now we could project the data onto $\hat{\boldsymbol{v}}$ by:

$$\boldsymbol{a_v} = \boldsymbol{A} \cdot \hat{\boldsymbol{v}}, \;\; \boldsymbol{g_v} = \boldsymbol{G} \cdot \hat{\boldsymbol{v}}$$

Now we have to find an horizontal plane, parallel to the floor, where the activity motion mostly occurs. To do so we have to remove the $\boldsymbol{a_v}$ component from the original data. We represent the accelerometer data lying on this new plane as $\boldsymbol{M}$ representing the so called motion plane. To find $\boldsymbol{M}$ we have to: $\boldsymbol{M} = \boldsymbol{A} - \hat{\boldsymbol{v}}\boldsymbol{a_v}^T$

In this new plane, we could see that the direction with the largest variance of projected data, represents the main direction of motion, ie in which direction the user is currently performing the activity with respect to the current smartphone orientation. By aplying PCA [8], we are able to find the direction along which the variance of measurements is maximized. This new extracted vector it is called horizantal vector $\vec{\boldsymbol{h}}$. We now could compute the horizontal versor: $\hat{\boldsymbol{h}} = \vec{\boldsymbol{h}}/||\vec{\boldsymbol{h}}||$ and we are now able to project our data onto this second new axis by:

$$\boldsymbol{a_h} = \boldsymbol{A} \cdot \hat{\boldsymbol{h}}, \;\; \boldsymbol{g_v} = \boldsymbol{G} \cdot \hat{\boldsymbol{h}}$$

To find the last axis its sufficient to apply a cross product between the two last axis found, so $\hat{\boldsymbol{l}} = \hat{\boldsymbol{v}} \times \hat{\boldsymbol{h}}$ and the obtain our last projection of the original data into this new space by:

$$\boldsymbol{a_l} = \boldsymbol{A} \cdot \hat{\boldsymbol{l}}, \;\; \boldsymbol{g_l} = \boldsymbol{G} \cdot \hat{\boldsymbol{l}}$$

Our final accelerometer and gyroscope trasformed vectors lying in this new orientation indipendent space is $(\boldsymbol{a_v}, \boldsymbol{a_h}, \boldsymbol{a_l})$ and $(\boldsymbol{g_v}, \boldsymbol{g_h}, \boldsymbol{g_l})$

Centering

- interpolazione lineare / downsampling
- rotational invariant citare tutti i paper e come funziona
- centering

C. Feature vector

- window strategy
- basic feature extraction
- autoencoder feature extraction

V. Learning Framework

Desctivere qui invece la parte tratteggiata come learning framework

Descivere prima l'autoencoder, come Ã¨ stato costruito e come viene allenato. TODO luca

Passare alla decrizione della mia archittetura, come sono stai selezionati gli iper-parametri, ecc ecc.

VI. Results

Io direi che in questa parte partiamo con il dataset eterogeneo, facendo vedere come si comporta anche con le posizione sit e stand, e dopodichÃ¨ passando ai risultati ottenuti con il rotational indipendent far vedere quello che abbiamo ottenuto nel dataset nostro, dove le attivita' sit e stand sono state compressate in no_activity!

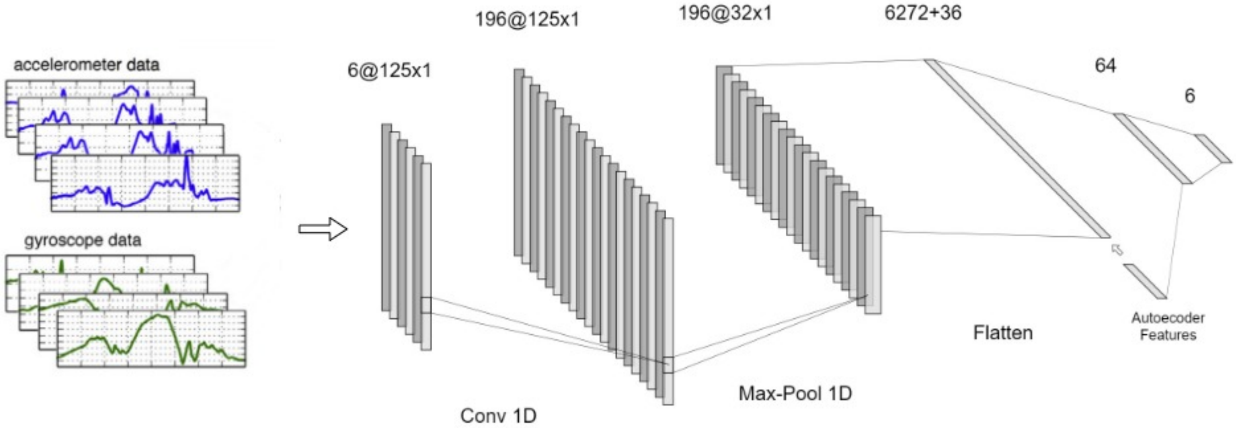

Fig. 2: Learning Framework

A. Original settings

Parlare dei risultati ottenuti da luca con un semplice autoencoder sia K-NN classifier che con FFNN alla fine. Un concetto alla volta!

Passare alla mia archittetura e spiegare come sono stati scelti i paramentri, poi far vedere che togliende le basic feature il modello decrementa la sua accuracy, e aggiungendo invece quelle de"autoencode il modello migliora.

Calcolare anche la F1 score e prbabilmente abbiamo migliorato di molto il modello proposto in [1]

B. Advance settings

Passare al nostro dataset e far vedere l'importanza del rotational invariant!!! Che migliora sensibilmente le performance ovviamente dato che la rete Ã¨ allenata in posizione fisse.

RIPORTARE TUTTE LE METRICHE, anche confusion matrix e argomentarle

VII. Concluding Remarks

This section should take max half a page, I personally find it difficult to come up with really useful observations, I mean ones that bring a new contribution with respect to what you have already expounded in the "Results" section. In case you have some serious stuff to write, you may also extend the section to 3/4 of a page :-).

In many papers, here you find a summary of what done. It is basically an abstract where instead of using the present tense you use the past participle, as you refer to something that you have already developed in the previous sections. While I did it myself in the past, I now find it rather useless.

What I would like to see here is:

1) a very short summary of what done,
2) some (possibly) intelligent observations on the relevance and *applicability* of your algorithms / findings,
3) what is still missing, and can be added in the future to extend your work.

The idea is that this section should be *useful* and not just a repetition of the abstract (just re-phrased and written using a different tense...).

Moreover: being a project report, I would also like to see a specific paragraph stating

4) what you have learned, and
5) any difficulties you may have encountered.

References

[1] H. Blunck, N. O. Bouvin, T. Franke, K. Grønbæk, M. B. Kjaergaard, P. Lukowicz, and M. Wüstenberg, "On heterogeneity in mobile sensing applications aiming at representative data collection," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pp. 1087–1098, 2013.

[2] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: overview, challenges and opportunities," *arXiv preprint arXiv:2001.07416*, 2020.

[3] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[4] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition," in *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pp. 127–140, 2015.

[5] M. Gadaleta and M. Rossi, "Idnet: Smartphone-based gait recognition with convolutional neural networks," *Pattern Recognition*, vol. 74, pp. 25–37, 2018.

[6] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole, "Which way am i facing: Inferring horizontal device orientation from an accelerometer signal," in *2009 International Symposium on Wearable Computers*, pp. 149–150, IEEE, 2009.

[7] A. Henpraserttae, S. Thiemjarus, and S. Marukatat, "Accurate activity recognition using a mobile phone regardless of device orientation and location," in *2011 International Conference on Body Sensor Networks*, pp. 41–46, IEEE, 2011.

[8] C. R. Rao, "The use and interpretation of principal component analysis in applied research," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 329–358, 1964.