Hybrid and Convolutional Neural Networks for Locomotion Recognition

Aomar OSMANI and Massinissa HAMIDI

Introduction and Context

Defining the context of displacements and trajectories of an individual, in particular, whether he is on a bicycle, in the bus or the subway at any given time, then opens perspectives to many applications and above all, enables technologies that provide smarter assistance and context-aware ambient intelligence.

We explore the relevance of an approach based exclusively on deep neural networks for locomotion recognition. This work is done within the Sussex-Huawei Locomotion-Transportation (SHL) recognition challenge as team *Power of Things*. More than 500 different convolutional and hybrid architectures are evaluated, and a Bayesian optimization procedure is used for hyper-parameters space exploration. The influence of these hyper-parameters on performances is analyzed using the fANOVA framework. Best models achieve a recognition rate of about 92% measured by the f1 score.

Approach

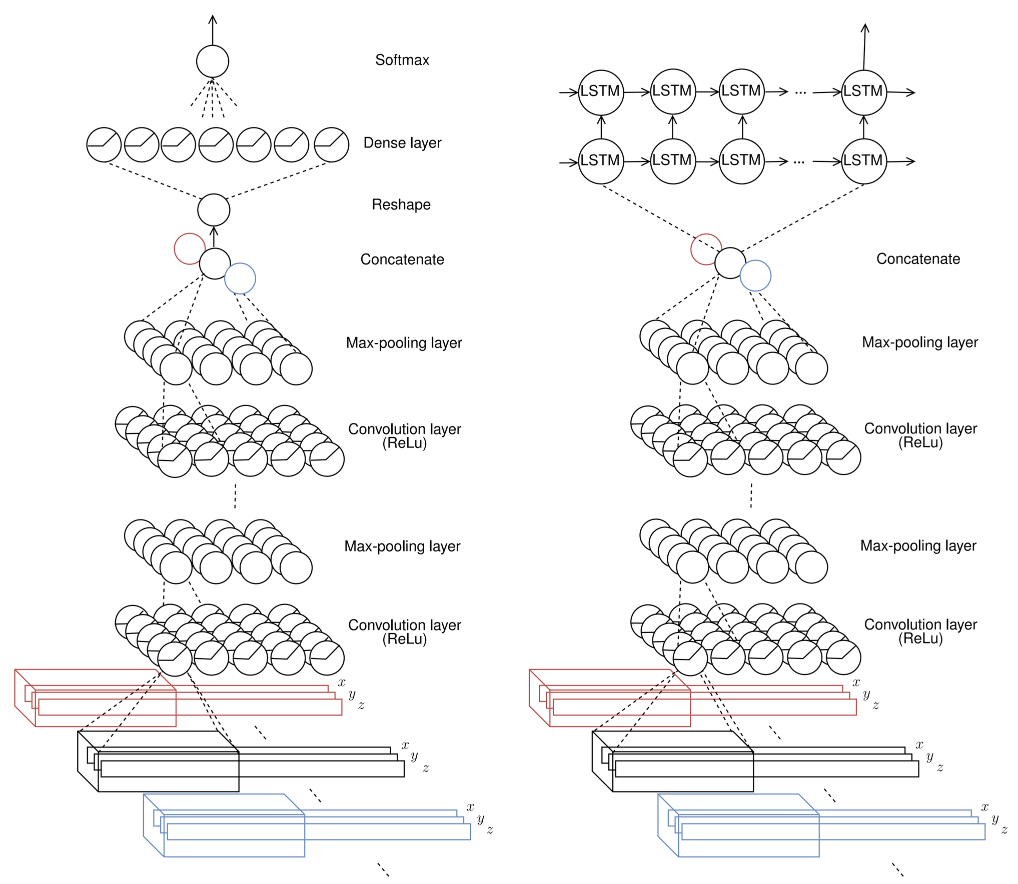


Figure 1: Schematic representation of; (a) a convolutional neural network that is used in our experiments. It encompasses a set of convolutional layers followed by a dense layer. (b) A hybrid neural network which consists of a set of convolutional layers followed by two recurrent layers of type long short-term memory (LSTM). Case where each modality is convolved apart from the others.

Convolutional modes

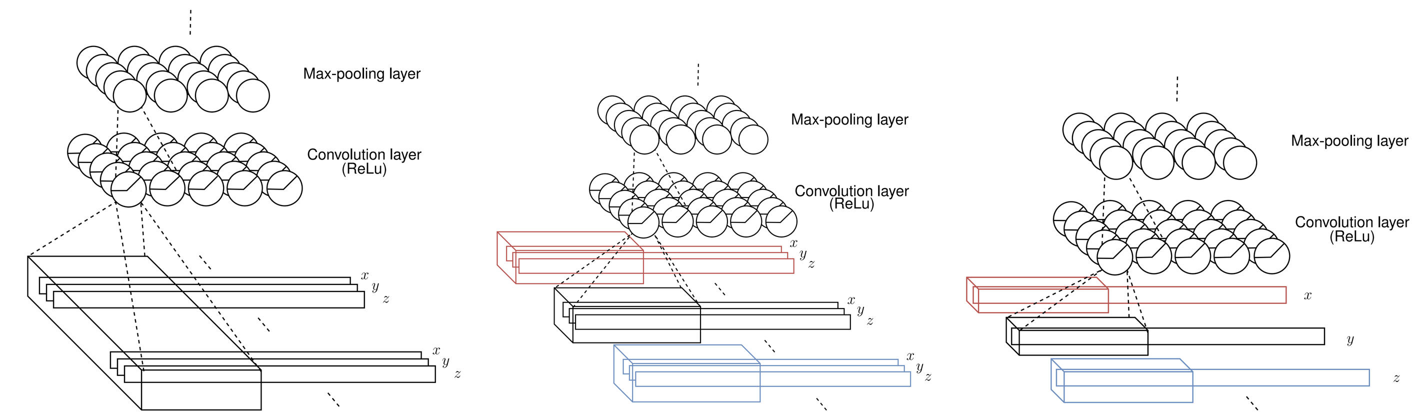
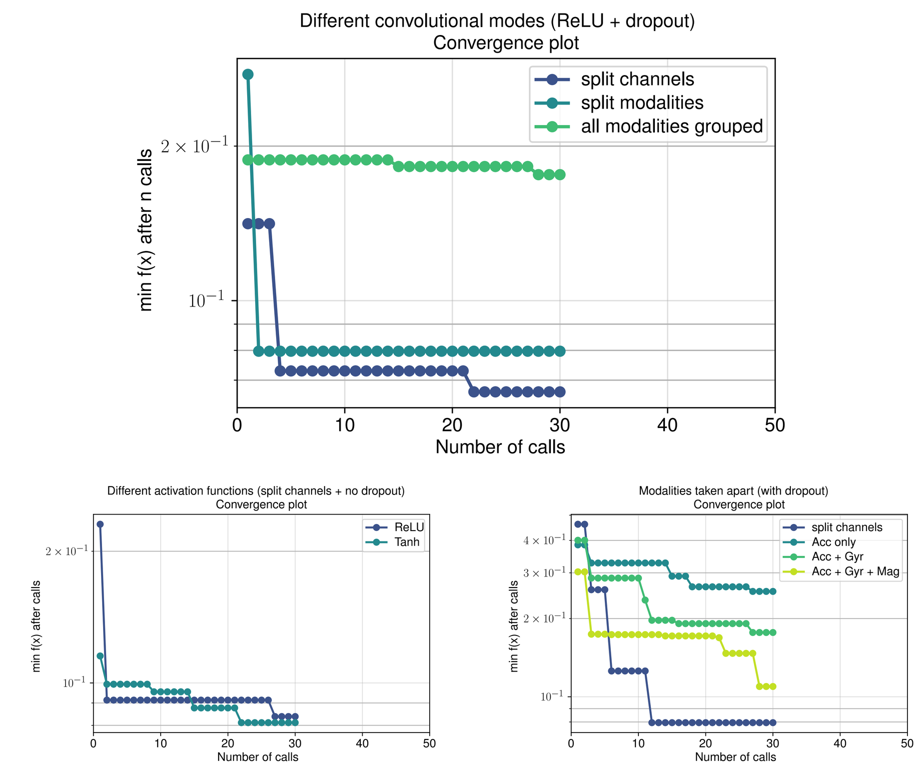


Figure 2: Schematic representation of the different *convolutional modes* of input data. (a) Modalities are grouped together and convolved with the filters. (b) Modalities are taken apart from each other. (c) Each channel is convolved alone.

Bayesian Optimization of Hyper-parameters

Param.	low	high	prior
l_r	0.001	0.1	log
ks_1	9	15	-
ks_2	9	15	-
ks_3	9	12	-
n_f	16	28	-
s	0.5	0.6	log
p_d	0.1	0.5	log
n_u	64	2048	-
n_{hu1}	64	384	-
n_{hu2}	64	384	-
p_{in}	0.5	1	log
p_{ou}	0.5	1	log
p_{st}	0.5	1	log



Results

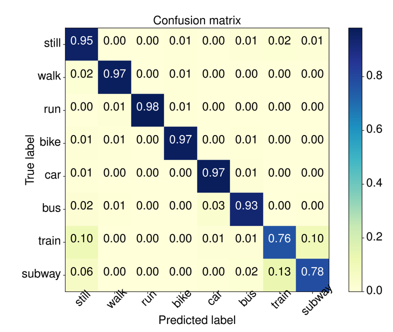


Figure 3: Confusion matrix for the 8 activities obtained with a hybrid model trained on all modalities. Best model's hyper-parameters are: n_{hu1}=176, n_{hu2}=384, l_r=0.1, n_f=20, $ks_i \in \{10, 13, 10\}$, s=0.558.

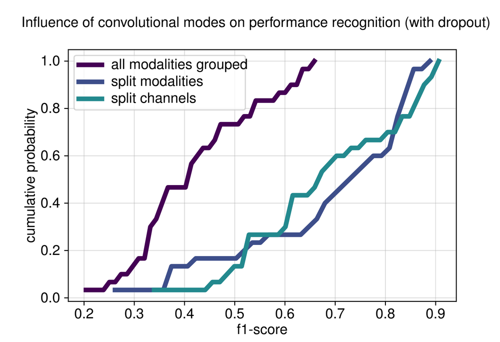


Figure 4: Cumulative distribution of recognition performances of hybrid models along with different configurations showing the influence of the different convolutional modes with regularisation.

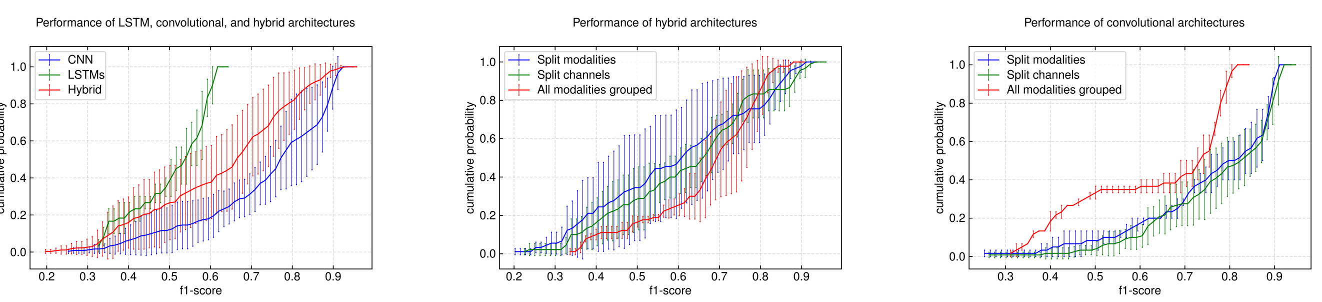


Figure 5: Cumulative distributions from the evaluation of the proposed architectures.

Hyper-parameters Impact Assessment

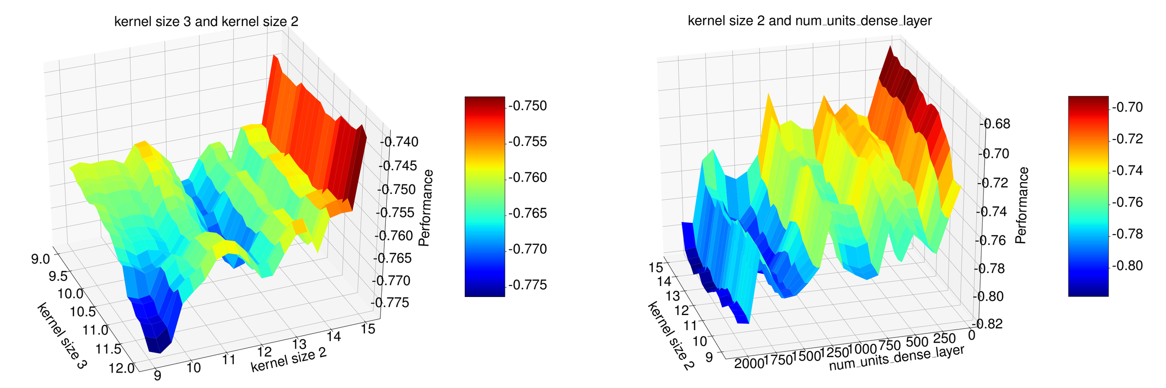


Figure 6: Pairwise marginal plots produced via fANOVA framework (3) for convolutional architectures. (a) kernel size 2 and kernel size 3 of convolutional layers 2 and 3 respectively, (b) number of units of the dense layer and kernel size 2 of convolutional layer 2.

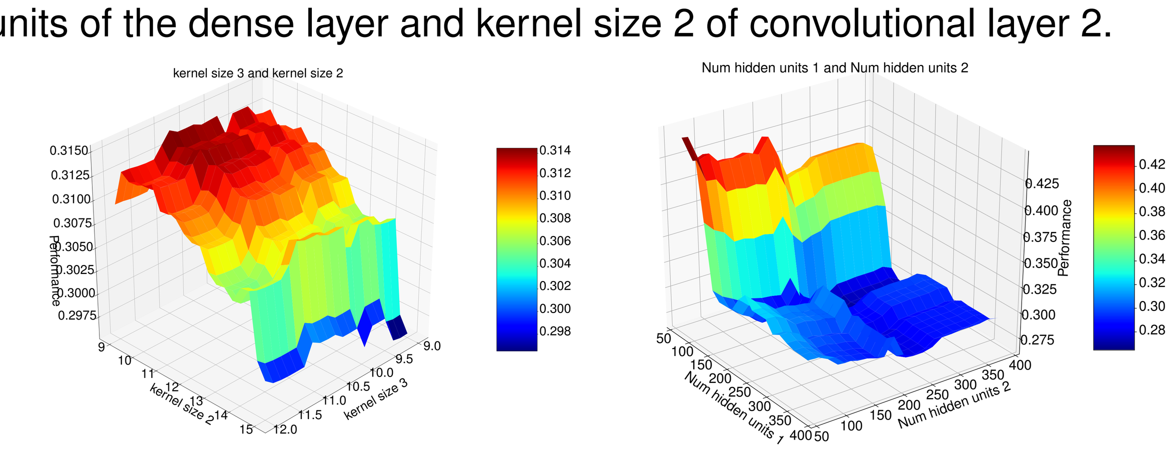


Figure 7: Pairwise marginal plots produced via fANOVA framework (3) for hybrid architectures. (a) kernel size 2 and kernel size 3 of convolutional layers 2 and 3 respectively, (b) number of hidden units in LSTM layers 1 and 2 respectively. This framework has been used by (2) in the context of activity recognition.

Implementation and Computational Aspects

As part of our experiments, we use the computing resources of a dedicated cluster called *Magi*. This grid is composed of several nodes of 40 cores each with an execution speed of 2.30 GHz and a working memory of 64 GB.

The execution time of the Bayesian optimization procedure for a given configuration of the hyper-parameters takes on average 5.77±2.7 hours. This execution time includes training the whole model on the training dataset for 13 successive epochs as well as the time taken by the aquisition function to determine the new hyper-parameter configuration to test next.

Models development and Bayesian optimization procedures are based on off-the-shelf implementations, all of which are free software. In this matter, we use the Tensorflow framework (1) for building the deep neural networks but also the `scikit-learn` library (4) and the `scikit-optimize` library specialized in optimizing cost functions. The implementation of the framework fANOVA is also free software and publicly available along with a user-guide.

References

[1] Abadi, M., et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, vol. 16 (2016), 265–283.

[2] Hammerla, N. Y., Halloran, S., and Plötz, T. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *International Joint Conference on Artificial Intelligence* (2016), 1533–1540.

[3] Hoos, H., and Leyton-Brown, K. An efficient approach for assessing hyperparameter importance. In *International Conference on Machine Learning* (2014), 754–762.

[4] Pedregosa, F., et al. Scikit-learn: Machine learning in python. *Journal of machine learning research 12*, Oct (2011), 2825–2830.