DOI: 10.1007/978-3-319-46687-3 58.

Implementation of a Modular Growing When Required Neural Gas Architecture for Recognition of Falls

Frederico B. Klein[1(✉)], Karla Štěpánová[2], and Angelo Cangelosi[1]

[1] School of Computing, Electronics and Mathematics,
Plymouth University, Plymouth, UK
{frederico.klein,a.cangelosi}@plymouth.ac.uk
[2] Department of Cybernetics, Czech Technical University, Prague, Czech Republic
stepakar@fel.cvut.cz
http://www.plymouth.ac.uk, http://www.fel.cvut.cz

Abstract. In this paper we aim for the replication of a state of the art architecture for recognition of human actions using skeleton poses obtained from a depth sensor. We review the usefulness of accurate human action recognition in the field of robotic elderly care, focusing on fall detection. We attempt fall recognition using a chained Growing When Required neural gas classifier that is fed only skeleton joints data. We test this architecture against Recurrent SOMs (RSOMs) to classify the TST Fall detection database ver. 2, a specialised dataset for fall sequences. We also introduce a simplified mathematical model of falls for easier and faster bench-testing of classification algorithms for fall detection.

The outcome of classifying falls from our mathematical model was successful with an accuracy of $97.12 \pm 1.65\,\%$ and from the TST Fall detection database ver. 2 with an accuracy of $90.2 \pm 2.68\,\%$ when a filter was added.

Keywords: Action recognition · Falls · Neural networks · Neural gas · Topological classifiers · Socially assistive robotics

1 Introduction

In the field of robotics, activity detection [11] is a fundamental concept if the robots are used in a setting where they are expected to cooperate with humans. The initial approaches to the detection of human actions involved the processing of RGB images, and as such were proven to be a hard problem due to the difficulty in segmenting the human body from the background and accurately processing pose information. Recently however this task was made considerably easier with the introduction of skeleton tracking based on depth-sensing cameras as implemented by the Microsoft Kinect and as it steadily improves, it also allows for more serious tasks that depend on activity recognition to be tackled, such as activity detection. We will focus on its use in the context of socially assistive robotics for social elderly care.

A. Hirose et al. (Eds.): ICONIP 2016, Part I, LNCS 9947, pp. 1–9, 2016.
DOI: 10.1007/978-3-319-46687-3_58

1.1 Ageing Population

With the ageing of populations around the world, elderly care is a field of growing concern. Many different technological aids [3] are being developed specifically for this population and robotics has emerged as a possible solution as the mobilisation of human caretakers for such a large amount of persons seems infeasible. While robots in regards to human-robot-interaction are yet to find a particular field in which it is undeniably useful, an interesting approach [18] to their use is finding newer areas in which they can nothing but excel, simply because there are no persons nor other technology available to perform that task. One of such tasks is around the clock health monitoring for independent living.

1.2 Our Task of Interest: Fall Detection

It is medical fact [9] that diseases that can present themselves as a loss of consciousness, such as strokes and heart infarctions - those two, the leading causes of death world wide - can have excellent prognosis if treated within 3 h. Particularly cardiac arrests present a survival rate of about one in each three subjects, if CPR and defibrillation are initiated in less than 5 min, whereas the probability of survival without any help is virtually zero [19]. Also other diseases such as pneumonia or COPD exacerbations do tend to have better prognosis [20] if treated promptly. One must take care as to not make bold assumptions, even more under the light that major reviews [5] are yet to reveal clear benefits of telemedicine, but some interesting recent results [4,8] demonstrate COPD as a likely candidate to benefit from remote monitoring.

The specific task of fall detection has recently attracted a lot of research, with a primary focus on smart home environments. In fact most fall detection systems [21] involve wearing special sensors device with accelerometers or detectors built on the floor or a combination of video and wearable devices with a sensor fusion approach in order to increase the accuracy of detection even further. These approaches although more simple (and therefore robust) have however the limitation of needing either a sensor to be worn at all times, or that the person's house to be adapted for this, which in practice will vastly limit its adherence. We see coding fall detection into some sort of a multi-functional robotic companion-that could have as one of its many functionalities: fall detection-as a reasonable solution to this problem. A robot can follow the user in different environment, position itself in order to prevent image occlusion and we avoid the need to renovate someone's house or remember always to wear a sensor.

1.3 Our Approach: Use the Parisi's Multilayer GWR Classifier

Using an unsupervised method for topological description [6] of tasks is not a new idea, since these methods have many possible advantages such as the ability to "operate autonomously, on-line or life-long, and in a non-stationary environment". We chose to replicate the infrastructure implemented by [15] for it's overall performance in the CAD60 database and the theoretical generality of

the method. In this paper we describe our implementation of a classifier based on an unsupervised Growing When Required Neural Gas with a sliding window scheme for time integration and chained in multiple layers to implement noise removal.

Source code (available in www.github.com/frederico-klein/ICONIP2016) of the Growing When Required Neural Gas implementation (in Matlab and Julia) is provided, as well as the full classification architecture and the inverted pendulum model (only in Matlab).

2 Materials and Methods

2.1 Justification for the Chosen Architecture: A Chained GWR Sliding Window Topological Classifier

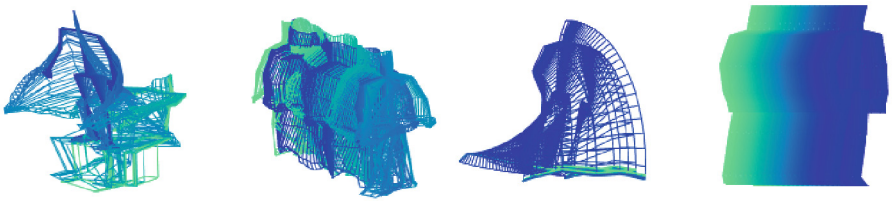
A detailed discussion of different types of neural gases is outside of the scope of this text. For a more in depth understanding one should probably first refer to Martinetz's paper [12] that implemented the first neural gas and later to Marsland's paper [13], that implemented the Growing When Required neural gas (GWR). The justification of using multiple chained gases (as opposed to one) is, first the biological plausibility reviewed extensively by Parisi but secondly probably due to necessity regarding the way too long execution time of a gas with a high number of dimensions. Finally one must add that, although neural gases, due to their nature, adjust to data that changes over time, this feature does not seem useful in tracking movement. For this function a sliding window scheme was used.

The dataset we chose to test our implementation was the TST v2 dataset contains skeleton positions Microsoft Kinect v2 and IMU data for 11 subjects performing either ADLs (activities daily living) and simulated falls. The subjects were between 22 and 39 years old, with different height (1.62–1.97 m) and build. Each of the two main groups (ADLs or Falls) contains 4 activities that are repeated three times by each subject [7]. For our present study only the skeleton joints in depth and skeleton space and time information were used. The accelerometer, as well as other data, were not used for our algorithm.

In addition to the real dataset a simplified stick model was developed to test the ability of a Growing When Required multilayer with a sliding window classifier to discriminate between action sequences that included a fall. We modelled 2 different activities, a fall and a walk as the movement of a stick in a 3D space and then simply substituted the stick for a typical skeleton.

Fall. We simulated fall of a person by a free falling inverted pendulum rod with a random initial pitch angular velocity θ and perfect slippage. It can be shown [16] that the kinematics differential equations that describe angle and position changes for a rod are:

$$-mg\frac{L}{2}cos\theta = \left(I_c + \frac{mL^2}{4}cos^2\theta\right)\ddot{\theta} - \frac{mL^2}{4}cos\theta sin\dot{\theta^2} \tag{1}$$



(a) (b) (c) (d)

Fig. 1. (a) A typical fall from TST v2 dataset. (b) one of the ADLs from the TST v2 dataset, a walk. (c) a typical fall from our model. (d) a "walk" from our model.

$$0 = m\ddot{x_c} \tag{2}$$

And, approximating a person by a slender rod, one has $I_c = \frac{mL^2}{3}$. The model also was given simulation parameters to add random noise in the variables of height (1.6–1.9 m), initial position (within a square area) and any initial yaw angle.

Walk. The simulation of a person's walk was done by simply doing a linear space of displacements inside the area that would be covered by the Kinect sensor, with random initial positions and walking angle (Fig. 1).

2.2 Skeleton Data

The algorithm presented uses skeleton data and not RGB-D raw images. A more thorough descriptions [17] of the data obtained from the depth sensor should be referenced, but in short it is a set of J points (where J is the number of joints) with x, y and z coordinates, each representing a landmark on the body in time [10] in a 3D space. We represent thus a particular pose as the concatenation of these J points, such as that for each time frame k we have a pose p represented by the matrix:

$$p(k) = \begin{bmatrix} j_{1x}(k) & j_{1y}(k) & j_{1z}(k) \\ j_{2x}(k) & j_{2y}(k) & j_{2z}(k) \\ \cdots & & \\ j_{Jx}(k) & j_{Jy}(k) & j_{Jz}(k) \end{bmatrix} \tag{3}$$

An action sequence represented on discrete time steps 1...K could therefore represented as the multidimensional array resulting of the sequential concatenation of the k-th pose matrices. To use the pose information with a gas we change the representation of the pose matrix $p(k)$ into a vector size $3 * J$ and the action sequence is the horizontal concatenation of the all the k-th, $p(k)$ matrices. One may thus understand the pose vector as a single point in a high dimensional space and an action sequence as a necessarily continuous trajectory in that space.

2.3 Construction and Randomisation of Training and Validation Sets

The dataset was separated into training and validation sets containing 80 % and 20 % of data respectively, before each training similarly to a repeated learning-testing method [1,2]. They were separated by subject, so that each subject had all of its actions belonging exclusively to one set. This was done to describe a more realistic testing scenario, in which the subject performing the activities is completely new having no activity data of himself in the training set, preventing bias in accuracy estimation due to overfitting the training set.

2.4 Preconditioning

The GWR algorithm is not translation invariant, so the first action performed on the data was to select a joint - based on our reference algorithm we used the hips and subtracted the offset from the hips joint in both the z and x coordinates from all other joint vectors. Secondly, we normalised (scaled) the data so that after scaling variance of the data would be equal to 1. The final step was to implement a centroid generating function, so to generate a smaller dimensionality representation of the skeleton poses, in a similar fashion to the function tested by Parisi. We created a model of 3 centroids that were the average position of the skeleton points such that the upper centroid was composed by the joints: head, neck, left shoulder, right shoulder, left elbow, right elbow; middle centroid corresponded to torso and lower centroid: left knee, right knee, left hip, right hip. Many other preconditioning functions are available on the supplied code and maybe be tried by the interested reader.

2.5 Classifier Architecture

The classifier was implemented as a serial chaining of gas subunits. This was done to enable different structures to be tried with minimal effort. All classification attempts in this text were done using 5 gas subunits linked in manner as to implement the architecture in Parisi's [15] paper (see Fig. 2), that is, 2 parallel sets of 2 gas subunits in series, each stream dealing with either pose positions or pose velocities and a last gas that integrates both.

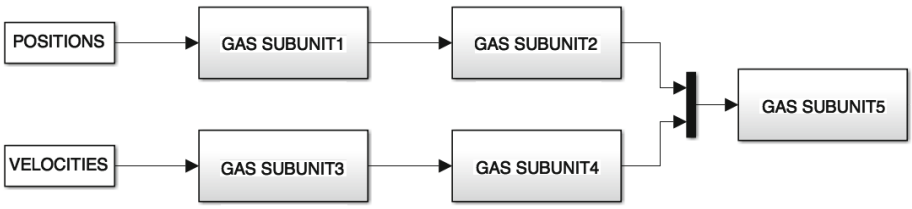


Fig. 2. Diagram of the classifier architecture.

For each gas subunit there are 5 main chained elements that are responsible for implementation estimation and classification:

- Sliding Window: implements the temporal concatenation of sample (also implements concatenation of multiple streams in case they exist).
- Gas Creator: receives data samples $p(k)$ or concatenated poses $W_l(k)$ and implements the learning algorithm for either the Growing When Required neural gas or the Growing Neural Gas.
- Mapping: finds the best matching pose from the nodes matrix A corresponding to each sample from the dataset.
- Labeller: simple labelling function that assigns the label of estimated concatenated pose as the same as the label of the pose to which it best matches.
- Activation checker: during training, checks to see if points are able to be well represented by the gas, and if not removes them from the sample.

3 Results and Discussion

For all the results here presented, the simulation parameters for the GWR neural gas are the same as in our reference paper [15].

3.1 Cornell CAD60 Dataset

As a means of comparing our implementation with that of Parisi, we also tested our architecture on the CAD60 dataset. Apparently our implementation does a lot of overfitting, as it reaches 99.6 % accuracy on the training set (average: 99.38 ± 0.2 % for 8 trials) but only reaches 71.7 % on the validation set. We noticed however that misclassifications were limited to some specific actions, with most having the same accuracy (greater than 90 %) in both sets. It is our conjecture that this difference reflects that the CAD60 dataset is too small to allow our stricter cross-validation method to produce generalization.

3.2 Falling Stick Model

Our algorithm, even with a much smaller network (100 nodes), seems to be quite consistently capable of classifying our faux fall/walk model. We simulated 20 subjects performing either a fall or a walk. The peak accuracy on the validation set of our implementation was 98.33 % (average: 97.12 ± 1.65 % for 8 trials).

3.3 TST Fall Detection ver.2 Dataset

Learning Across Layers. In order to understand how learning happens across layers we analyse the output classification from 5 gases with 1000 nodes run over 10 epochs (see Table 1) the results reflect what we would expect: there is a steady increase as we progress through the layers and there is a gain in accuracy.

Table 1. Progression of classification accuracy within different layers for chained GWR Neural Gas classifier with 1000 nodes and run over 10 epochs (for 8 trials).

Gas element	Validation set	Training set
GWR gas 1 Pos	$67.69 \pm 0.73\,\%$	$93.04 \pm 0.12\,\%$
GWR gas 2 Vel	$64.80 \pm 0.93\,\%$	$69.43 \pm 0.66\,\%$
GWR gas 3 Pos	$66.93 \pm 1.02\,\%$	$90.45 \pm 0.18\,\%$
GWR gas 4 Vel	$65.14 \pm 0.77\,\%$	$70.66 \pm 0.65\,\%$
GWR gas 5 STS	$73.99 \pm 1.16\,\%$	$88.35 \pm 0.41\,\%$

Mode Filter. With the intention of performing some sort of temporal filtering, we implemented a moving mode filter. The moving mode filter had an important positive effect on the classification results of the TST v2 database (see Table 2). Highest classification accuracy achieved (See Table 3) was 94.2 % on the validation set by 3^{rd} gas with 1000 nodes and 10 epochs with mode filter length of 35 data samples. One must note that adding a moving mode filter of size 35 means a delay of 11.67 s (since we need $(9 + 1) * 35$ samples @30 Hz) much more than the 0.6 s Parisi reported.

Table 2. Accuracies (in %) after applying the moving mode filter on classification results of the final gas unit of the classifier (for 8, 8 and 1 trials respectively).

Epochs	10		20		30	
Filter length	Val	Train	Val	Train	Val	Train
5	81.25 ± 1.44	95.05 ± 0.69	79.72 ± 2.09	79.72 ± 2.09	82.7	94.9
10	85.95 ± 2.13	95.74 ± 0.54	83.87 ± 2.93	96.00 ± 0.38	86.9	95.5
15	88.57 ± 2.48	95.31 ± 0.36	85.91 ± 2.93	95.46 ± 0.30	88.6	97.0
20	89.95 ± 2.36	95.34 ± 0.23	86.70 ± 4.08	95.26 ± 0.31	88.6	95.0
25	90.16 ± 2.28	94.32 ± 0.27	87.37 ± 3.79	94.47 ± 0.25	89.2	95.4
35	90.20 ± 2.68	92.29 ± 0.37	88.49 ± 3.83	92.44 ± 0.38	91.5	93.3
40	89.28 ± 2.68	91.23 ± 0.32	87.75 ± 3.82	91.33 ± 0.35	91.5	92.2
50	87.39 ± 2.06	88.84 ± 0.33	85.35 ± 3.39	88.80 ± 0.49	91.3	89.9

Comparison with RSOM. As a means of comparing the performance of our implementation, we also classified the TST v2 dataset using an RSOM implementation. The RSOM used the same preconditioning as we did for the chained gas classifier and a set of 3 consecutive poses $(p(k), p(k-1), p(k-2))$. The simulation parameters were: 900 nodes, 30 epochs, method 'RSOMHebbV01'. The peak accuracy on the validation set of the RSOM with these parameters was 78.76 % (average: 77.67 ± 0.77 % for 5 trials).

Table 3. Confusion matrix for our most accurate gas classifier. The calculated accuracy for the validation set is 94.2 %, higher than the 92.0 % training set.

Validation set		Target		Training set		Target	
		1	2			1	2
Output	1	1532	36	Output	1	4006	258
	2	124	1054		2	328	2746

4 Conclusion

The resulting classification scheme does the task which we want, that is, discriminate falls within the TST v2 dataset, it does it better than the RSOM and it does it consistently with around 90.2 ± 2.68 % accuracy while using the mode filter. We believed we achieved our goal and we have now a classifier of falls with openly accessible code that will hopefully encourage persons into designing experiments using fall detection or using neural gases for classification of hard to classify data.

Acknowledgment. This work was partially supported by CNPq Brazil (scholarship 232590/2014-1) and by SGS grant No. 10/279/OHK3/3T/13, sponsored by the CTU in Prague, Czech Republic.

References

1. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. Stat. Surv. **4**, 40–79 (2010). doi:10.1214/09-SS054
2. Burman, P.: A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. Biometrika **76**(3), 503–514 (1989)
3. Chan, M., Estéve, D., Escriba, C., Campo, E.: A review of smart homes-present state and future challenges. Comput. Methods Prog. Biomed. **91**, 55–81 (2008). doi:10.1016/j.cmpb.2008.02.001
4. Fernandez-Granero, M.A., Sanchez-Morillo, D., Leon-Jimenez, A.: Computerised analysis of telemonitored respiratory sounds for predicting acute exacerbations of copd. Sensors (Basel) **15**, 26978–26996 (2015). doi:10.3390/s151026978
5. Flodgren, G., Rachas, A., Farmer, A.J., Inzitari, M., Shepperd, S.: Interactive telemedicine: effects on professional practice and health care outcomes. In: Cochrane Database of Systematic Reviews. Wiley (2015)
6. Furao, S., Hasegawa, O.: An incremental network for on-line unsupervised classification and topology learning. Neural Netw. **19**, 90–106 (2006). doi:10.1016/j.neunet.2005.04.006
7. Gasparrini, S., Cippitelli, E., Gambi, E., Spinsante, S., Wåhslén, J., Orhan, I., Lindh, T.: Proposal and experimental evaluation of fall detection solution basedon wearable and depth data fusion. In: Loshkovska, S., Koceski, S. (eds.) ICTInnovations 2015, Advances in Intelligent Systems and Computing, pp. 99–108. Springer International Publishing, Switzerland (2016)

8. Ho, T.-W., Huang, C.-T., Chiu, H.-C., Ruan, S.-Y., Tsai, Y.-J., Yu, C.-J., Lai, F.: Effectiveness of telemonitoring in patients with chronic obstructive pulmonary disease in Taiwan-a randomized controlled Trial. Sci. Rep. **6** (2016). doi:10.1038/srep23797
9. Jauch, E.C., Saver, J.L., Demaerschalk, B.M., Khatri, P., McMullan Jr., P.W., Qureshi, A.I., Rosenfield, K., Scott, P.A., Summers, D.R., Wang, D.Z.: AHA/ASA Guideline. Stroke (2013)
10. JointType enumeration [WWW Document], n.d. https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx. Accessed 14 May 16
11. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. IEEE Trans. Pattern Anal. Mach. Intell. **38**, 14–29 (2016)
12. Martinetz, T.M., Schulten, K.J.: A "Neural Gas" networklearns topologies. In: Kohonen, T., Mäkisara, K., Simula, O., Kangas, J. (eds.) Proceedings of the International Conference on Artificial Neural Networks 1991, Espoo, Finland, pp. 397–402, Amsterdam, North-Holland, New York (1991)
13. Marsland, S., Shapiro, J., Nehmzow, U.: A self-organising network that grows when required. Neural Netw. **15**, 1041–1058 (2002)
14. Parisi, G., Wermter, S., others.: Hierarchical SOM-based detection of novel behavior for 3D human tracking. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2011)
15. Parisi, G.I., Weber, C., Wermter, S.: Self-organizing neural integration of pose-motion features for human action recognition. Front. Neurorobotics **9**, (2015). doi:10.3389/fnbot.2015.00003
16. Peacock, T., Hadjiconstantinou, N.: Course materials for 2.003J/1.053J dynamics and control I, Spring (2007). MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Accessed 13 May 2016
17. Prime sensor^{TM} NITE 1.3 framework programmer's guide - NITE.pdf. http://pr.cs.cornell.edu/humanactivities/data/NITE.pdf. Accessed 14 May 2016
18. Rabbitt, S.M., Kazdin, A.E., Scassellati, B.: Applications and recommendations for expanded use. Clin. Psychol. Rev. 35, 35–46. doi:10.1016/j.cpr.2014.07.001
19. Valenzuela, T.D., Roe, D.J., Cretin, S., Spaite, D.W., Larsen, M.P.: Estimating effectiveness of cardiac arrest interventions: a logistic regression survival model. Circulation **96**, 3308–3313 (1997). doi:10.1161/01.CIR.96.10.3308
20. Wilkinson, T.M.A., Donaldson, G.C., Hurst, J.R., Seemungal, T.A.R., Wedzicha, J.A.: Early therapy improves outcomes of exacerbations of chronic obstructive pulmonary disease. Am. J. Respir. Crit. Care Med. **169**, 1298–1303 (2004). doi:10.1164/rccm.200310-1443OC
21. Yongli, G., Yin, O.S., Han, P.Y.: State of the art: a study on fall detection. World Acad. Sci. Eng. Technol. **62**, 294–298 (2012)