Artificial Intelligence and Neural Networks based approach for signal denoising in non-linear spectroscopy applications

Luca Polenta (Project Author), Matteo Matera (Project Author), Christian Napoli (Professor) and Stefano Giagu (Professor)

Abstract

A project in collaboration with INFN (National Institute of Nuclear Physics). It is based on the development of neural network based algorithms and signal denoising for Raman spectroscopy and related applications. This project involves the development of artificial intelligence algorithms and neural networks that can isolate the Raman signal from spurious non-linear contributions.

Keywords

Residual Learning Neural Network, One-Dimensional Signal Denoising, Raman Spectrum, Time-series Signal

1. HIGHLIGHTS

- Using residuals learning to denoise Raman spectroscopy signals
- Use of parallel networks to better study and combine different features
- Generic denoise since we do not focus on features peculiar to the origin of the signals
- Use of non-distorting signal pre-processing
- Excellent results both qualitatively and quantitatively

2. Introduction

Denoising methods are both a classic and contemporary topic, as they are indispensable in many scientific fields. The goal of these methods is to remove the noise that is added to the object being studied, whether it is a signal, an image or something else. In the case of the current project, the noise affects a one-dimensional signal obtained from Raman spectroscopy. This is a non-destructive material analysis technique that provides detailed information about the chemical structure and characteristics of an analyzed material. In particular, it is based on the investigation of the scattering of a monochromatic electromagnetic radiation by the analyzed sample. Raman spectroscopy can be used for different types of materials that can be in a solid, liquid or gaseous state. It is crucial for the study of carbon-based materials such as graphite, graphene and similar materials. In addition to geological investigations, it is also used in a wide variety of fields, including industrial process control, planetary exploration, internal security and even in the medical field since, combined with artificial intelligence algorithms and neural networks, it is used to study cancer cells, melanomas and similar diseases for diagnostic purposes.

This emphasizes the importance of having a signal that is as clean and correct as possible to allow a realistic and reliable study of the material on which it is applied. Furthermore, this relevance becomes more pronounced as this technique has several problems that compromise the reliability of the signals obtained. First, in a normal environment, Raman spectroscopy encounters noise that often hides the peaks, making it difficult to interpret the spectrum. In addition, when performing Raman spectroscopy, the baseline may not always be constant and this can blur the signals, as the peaks do not appear with the right intensity, which deteriorates the analytical results. Among the various alterations that can corrupt the signal, the ones of greatest importance are baseline alteration and high peak noise because when analyzing signals obtained with this technique, it is more important to look at the peaks, the times at which they occur and their intensity, rather than the entire signal and the noise in it. Therefore, in this project, more attention will be paid to baseline correction and spikes, as well as their related characteristics, and only secondarily an attempt will be made to reduce all the noise along the signal.

To achieve this goal, several methodologies have been developed over the past decade and many of them do not involve artificial intelligence or neural networks. For example, one of the first and most efficient was the EMD [1] (Empirical Mode Decomposition) algorithm. This is a method of noise reduction in a signal that has led to good results, although only in special cases as it has several

polenta.1794787@studenti.uniroma1.it (L. Polenta); matera.1795339@studenti.uniroma1.it (M. Matera); c.napoli@uniroma1.it (C. Napoli); stefano.giagu@uniroma1.it (S. Giagu)

https://www.linkedin.com/in/luca-polenta/ (L. Polenta); https://www.linkedin.com/in/matteo-matera-86a611233/ (M. Matera); https://sites.google.com/diag.uniroma1.it/napoli/home (C. Napoli); https://stefanogiagu.site.uniroma1.it/home (S. Giagu)

CEUR Workshop Proceedings (CEUR-WS.org)

necessary restrictions for its proper functioning. There are also other wavelet noise reduction methods that can be used to solve this problem, including Wiener's estimation, Empirical Bayes, James-Stein Block, Minimax Estimation, Stein's Unbiased Risk Estimate, Universal Threshold G and so on but all are less efficient than more modern techniques that instead rely on artificial intelligence.

More recent projects, in fact, make use of neural network-based approaches to improve denoising performance, and so a similar technique will be developed in this project as well. In particular, it has been developed a DNN network based on discriminative learning model called "DnCNN". It consists of a set of feed-forward convolutional neural networks (CNNs) which recognizes noise in the signal and subtracts it from the signal to clean it of unwanted noise. In addition, in order to create a procedure that is as universal as possible, in this project no attention will be paid to the origin of the signal and therefore studies will be performed regardless of the features of the signal origin. Finally, special attention will be paid to optimizing the model, as it is desired to produce the lightest possible network.

3. Related Works

As mentioned earlier, the first studies that attempted a Raman spectrum denoising approach took advantage of wavelet analysis algorithms, as in the case of the papers [2] and [3]. However, these methodologies, although interesting at the time, were quickly overtaken by more complex and targeted algorithms in the field of natural non-stationary signals. For example, one of the earliest techniques that led to advances in the state-of-the-art Raman spectrum denoising was the EMD[4] algorithm. This is a method of reducing a signal without leaving the time domain and is useful for analyzing natural signals, which are most often non-linear and non-stationary. This method does not need to analyze and study the unknown signal before denoising, but through an iterative process decomposes a signal x(t) into intrinsic mode functions (IMFs), which form a complete basis that is nearly orthogonal to the original signal. However, this technique can present two issues: first, the algorithm works optimally if the number of zero crossing points and the number of peaks is equal, and this characteristic is not necessarily present in a signal obtained by Raman spectroscopy. Therefore, the algorithm cannot iterate more than once and is therefore less effective. The second problem is that in Raman spectroscopy the average value of the upper envelope and the lower envelope is zero, i.e. the lines of the upper and lower envelope are locally symmetrical to the time axis, and this compromises the proper functioning of the technique. Therefore, this technique is useful only in special cases where the Raman spectroscopy signals have all the necessary characteristics for the algorithm to work properly.

Then there are further techniques which make use of procedure that do not use neural networks. In this other paper [5] the use of Wiener's estimation has been proposed to numerically calibrate the dataset and decrease the presence of noise in the signal. Moreover, in this particular case, the procedure is carried out without the need of experimental measurements (as happened in other work) and theoretically this brings several advantages since it can lead to a solution that disregards the peculiarities present in specific Raman signals for certain types of materials. In any case, the results obtained from this project with three types of samples (including a phantom sample, a human fingernail, and leukemic cells) show huge improvements in performance over two commonly used denoising methods, namely moving average filtering and Savitzky-Golay filtering. This method provides comparable or even better denoising performance in cases with low signal-to-noise ratios. Moreover, the performance of the proposed method is significantly less sensitive to parameter choices.

Furthermore, in more recent techniques the signal denoising problem is addressed using neural networks of varying complexity. For example, a fairly simple and high-performance DNN (Deep Neural Network) model was developed in the article [6]. It is obtained by joining two convolutional neural networks in parallel where the two CNNs are structured in a fairly simple way: they are a series of convolutional layers alternating with MaxPooling layers and that uses the "ReLU" as activation function. At the end the results of the two parallel networks are concatenated by a fully Connected layer before being passed to the regression output layer. According to their studies, the use of two CNNs in parallel allows to better study the features of the signal and then recognize it when there is a lot of noise. The results obtained from this procedure are much better than previous denoising methods that did not use neural networks and better performing than simpler neural networks such as a single CNN.

Then there are several studies that make use of more complex neural networks to accomplish our goal. For example, in the paper [7] an unsupervised deep learning neural network has been developed. It is able to perform both denoising and segmentation of the Raman signal. The network named "UHRED" is able to learn with a "one-shot" technique, i.e. it learns from a single image and without any requirement for training on previously labeled datasets or images. In addition, then it applies a k-means clustering algorithm to the processed data and segment them. The results obtained show how UHRED unequivocally improves hyper-spectral contrast in low SNR images. The implementation of the k-means clustering algorithm in latent space has also led to excellent results in the case of automated and unsupervised image segmentation. This leads to an intuitive map of chemical species, even if limited to some cases of them. In fact, multi-mode signals such as harmonic generation, fluorescence, thermal lens and others are not included. The researchers of that project believe that as computing power continues to improve, the UHRED scaffolding can be further generalized to include more cases.

4. Dataset and Pre-Processing

There are four datasets used for testing in this project. For each dataset there is both a noisy and a clean version. The first dataset has only samples with low noise, the second with mixed noise and the last with high noise. Each has 5000 samples of 801 timesteps each. Finally, there is a fourth mixed dataset, dated August 17, composed of 15000 samples, each of 801 timesteps.

Before processing the datasets with the neural network, each of them can undergo various types of pre-processing to improve overall performance of the model. The pre-processing tested and implemented in this project are three. The first allows to normalize the noisy samples of the dataset between 0 and 1. Normalization brings the values of the noisy signal closer to those of the corrected signal, although it can lead to a change in baseline angle by either increasing or decreasing it. An example of the result obtained is in figure 1 below:

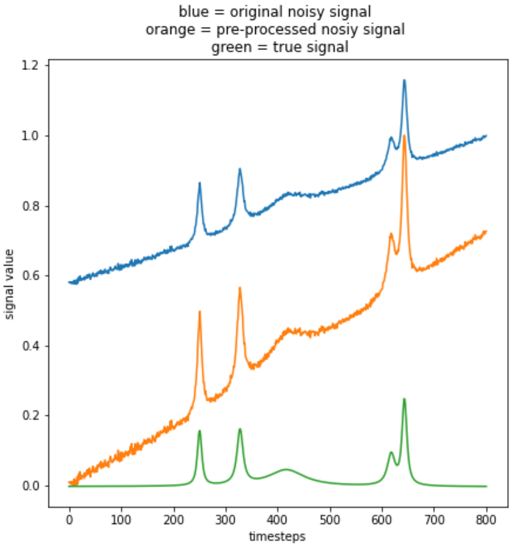

Figure 1: Signal normalization between 0 and 1

The second pre-processing allows to shift the dataset in such a way that the minimum value of the noisy signal is at zero and the other values are shifted by the same amount and in the same direction. This is achieved by finding the minimum value of each sample among the 801 timesteps and subtracting that minimum from all timesteps. This is the only pre-processing that neither distorts the shape of the signal nor causes the baseline angle to vary. The only things that it does is to shift the curve: if it is above zero, it will be shifted down and vice versa if it is below zero, it will be shifted up. An example of the result obtained is in figure 2 below:

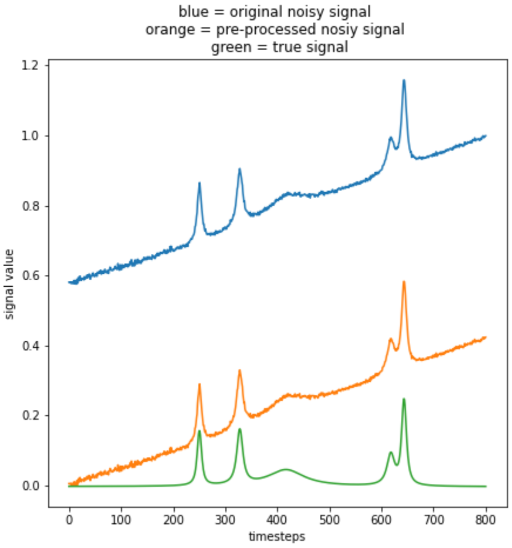

Figure 2: Shift the noisy signal so that the minimum value is zero

Finally, the last pre-processing that was tested on the noisy samples is the airPLS [8] algorithm. It was implemented by importing the algorithm from the original project repository [9]. airPLS is the acronym for adaptive iteratively re-weighted Penalized Least Squares algorithm. It iteratively changes weights of sum squares errors (SSE) between the fitted baseline and original signals, and the weights of SSE are obtained adaptively using between previously fitted baseline and original signals. This baseline estimator is fast and flexible in fitting baselines. It also brings the noisy signal much closer to the clean signal, although it very often causes signal distortion. In fact, recalculating the signal values for each timestep can slightly change the shape of the signal and the intensity of the peaks. It can also happen that it fails to properly redefine the extremes of the signal along the baseline because, instead of getting a constant baseline, it produces a linear increase or decrease. This is the pre-processing that could distort the signal trend and peak intensity. An example of the result obtained is in figure 3 below:

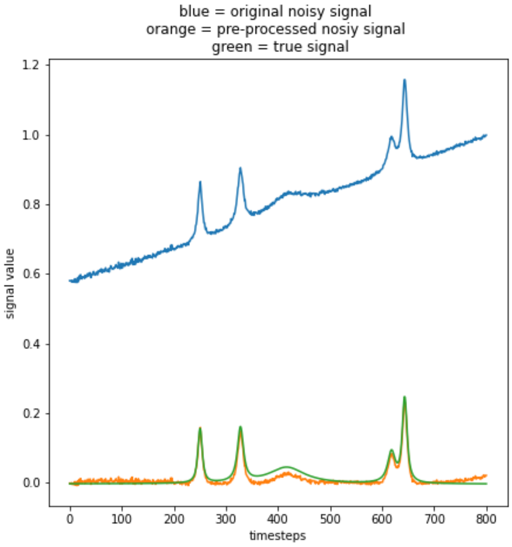

Figure 3: Application of airPLS algorithm on noisy signal

In general, all of the pre-processing used have different advantages and disadvantages, but they bring the noisy curve closer to the trend it should have, so it was deemed necessary to attempt the use of each and have the neural network correct any disadvantages obtained.

Finally, another pre-processing that is performed is a modification of the shape of the dataset array. In fact, to process datasets with convolutional layers it is necessary to add a last axis corresponding to the number of features, i.e. equal to 1, so the shape of the data changes from (sample,timestep) to (sample,timestep,features)=(sample,timestep,1).

5. Model

The model developed in this project proposes a DNN composed of a series of parallel CNN residual learning networks that study the noise features that affect the signal. Each of them produces a result that is merged into a single result through an Average layer. Finally it is processed by a new network before being subtracted from the noisy signal, thus obtaining the clean one. This particularly articulated structure was necessary because the signals we have to analyze are composed of various features, both in terms of the trend of the signal itself and in terms of the various types of noise to which they may be subject. In this way each network identifies with greater accuracy the features of each distortion and the final one combines the various results in a coherent way with respect to what we need to obtain. In particular, each network is obtained from the implementation of the same network with some parameter variations from case to case. The general structure of the network used in each branch is inspired by an image denoising project that uses a DNN Residual Learning network [10]. Residual learning is implemented through a skip connection in which the input is both processed by the network and used at the end of the model to subtract the result obtained by the network. This way it can be reused at the end in the subtraction so that it is clean of noise. The part of the model that deals with the study of noise is composed of a first convolutional layer that receives the input and its number of filters is equal to 1. Then there are a series of convolutional layers with a customizable number of filters, whose number corresponds to the depth of the network. The depth is also customizable when creating the model. In addition, it is possible to decide whether to insert Batch Normalization layers between intermediate convolution layers. Finally, there is an additional convolutional layer at the end of the intermediate part and its number of filters is 1. The last function of the model is a Subtract that performs the difference between the input signal and the output of the last convolutional layer.

In the current project, a model that applies this procedure several times has been implemented, but many changes and customizations have been made to both the parameters used and new ones. First of all, it should be noted that, having only one-dimensional inputs, all layers of the network were designed in a 1D dimension. Another modification concerns the activation function, which originally was "ReLU" and has been changed to "LeakyReLU". This change allowed for a slight improvement in performance because nodes that are generally not activated during training are still subject to a small gradient that allows them to improve and be used in the future if they perform again. In addition, in the convolutional layers the optimal values identified by the original paper were kept, but some additions were made on the unused parameters. In particular, the dilation_rate value has been added that allows to manage the distance between values in the kernel by inserting gaps (zero values) in it every timestep. Through it the network learns better the general characteristics of the signal without counting the smallest variations from one timestep to another that could be corrupted by noise. Moreover, always as a parameter of the convolutional layers, the L2 regularization was inserted to improve the denoising along the signal. It was inserted in the parameters kernel_regularizer, bias_regularizer and activity_regularizer which are applied to the kernel weight matrix, bias vector and to the output of the layer respectively. In the end, what differentiates the 3 networks in parallel and the last one before the end of the model are the size of the kernels, the presence or not of the batch normalization layers and the number of filters per convolution and the depth of the network, i.e. the amount of convolutions and batch layers in the core of the network. A scheme of the model is in figure 4 below:

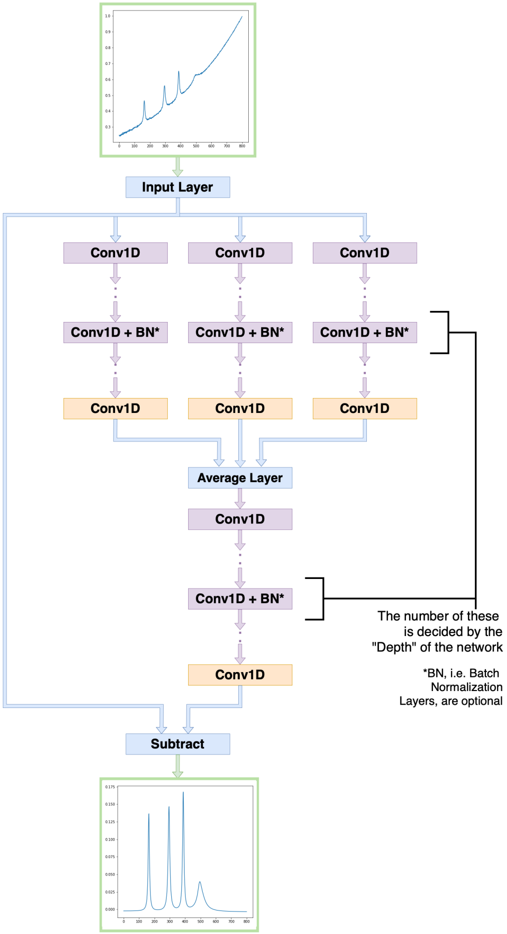

Figure 4: parallel DnCNN residual learning network model

Furthermore, the model is compiled using Adam as optimizer and residual sum of squares (RSS) as loss. In addition, mean square error (MSE) and mean absolute error (MAE) were added as metrics in the training. Both are often used in this field to observe when the predicted curve deviates from the actual curve. The formulas of the metrics used to compute the losses are:

$$\mathrm{RSS} = \sum_{i=1}^{n} (x_i - \widehat{x}_i)^2$$

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n} (x_i - \widehat{x}_i)^2}{n}$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |x_i - \widehat{x}_i|}{n}$$

where $\widehat{x}_i$ is the prediction and x_i is the true value.
Finally, four callbacks are added to both models in order to improve the training. The first is the ReduceLROnPlateau which reduces the value of the learning rate by a factor of 0.5 when there is no improvement in validation loss for more than 2 epochs. This allows training to be refined when the learning rate is too high. The second callback is earlystopping which interrupts the training in advance if there are no improvements in the validation loss for more than 7 epochs. There is improvement if the validation loss improves more than to 0.5, otherwise it is considered insufficient. The third callback is the ModelCheckpoint that allows to save all the model only when there are improvements in the validation loss. Finally, the last callback is TensorBoard which provides a tool for visualization and logging of the model.

6. Experiments and Results

The first experiments were focused on optimizing the model to find the right combination of parameters for each network to achieve the best performance. After performing many empirical tests, the optimal parameters identified are the following: the optimal network depth is [17,12,7,3] and is handled by the "depth" parameter; the optimal number of filters for each of the convolutions inserted by the "depth" parameter is [96,64,32,96]; the kernel sizes are [5,15,30,7]; the optimal value for dilation_rate is equal to 5 in all convolutions of the networks, except for the last one which has a one filter and that is inserted at the end of each network before the average layer or the subtraction layer; the optimal results were obtained by not inserting the Batch Normalization Layer within the depth of all networks; the initial learning rate relative to the Adam optimizer is 5e-4; the optimal value of the L2 regularizer is 1e-5 or 1e-6. For simplicity, the variations between values of the same parameter between different networks have been reported in an array of values where the first 3 refer to each of the parallel networks and the last to the final network.
Afterwards, different pre-processings were evaluated to assess which one leads to better results. To compare the various techniques, two approaches will be used: the first is qualitative in which some samples will be compared graphically in order to discard the techniques that appear clearly worse than others; the second is a quantitative objective approach that measures precisely the number of peaks, the timestep in which they occur and the intensity they have in order to obtain an overall perspective of the performance achieved. Analyzing first the most relevant samples through the qualitative method, the first preprocessing that has been tested is airPLS and the results

obtained are in the following figures 5 and 6:

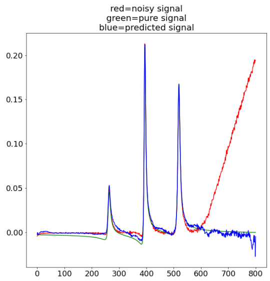

Figure 5: airPLS on sample 1

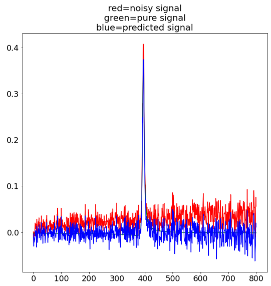

Figure 6: airPLS on sample 2

Applying the airPLS algorithm to the dataset does not bring any benefit to the training of the network. In the case of the second sample, where the noise was high, there seems to have been no correction of the noise along the curve, although there seems to have been a correction of the baseline, which in any case was not very distorted. In an attempt to improve this result several variants of the model have been tried and some simpler ones are able to do a better job with respect to the denoising of noise along the signal, but these models were less effective in all other datasets, so it was decided not to continue along this way and to seek further solutions. Moreover, in the first sample, the airPLS presented the baseline growth defect described above. This distortion did not lead to excessive model failure, even the results are not very accurate. What can be deduced is that airPLS is not a sufficiently adequate pre-processing because, even if it brings the noisy signal very close to the original one, the latter is not always done correctly and distortions of the signals and peaks can compromise the correct functioning of the model.
Subsequently, the second pre-processing that was tested is the normalization between 0 and 1 of each noisy sample and the results obtained on two samples belonging to the same dataset as before are in the following figures 7 and 8:

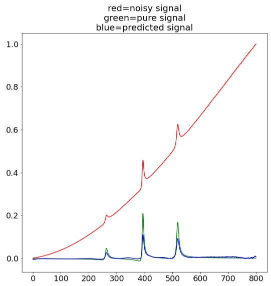

Figure 7: Normalization on sample 1

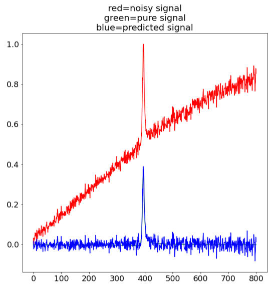

Figure 8: Normalization on sample 2

In the first sample, where the noise was slight, it is possible to notice that the prediction followed the correct trend, although it did not correctly predict the values of the peaks. This is a quite relevant problem because the peaks are fundamental for the study of Raman spectroscopy signals. Probably the excessive distortion of the baseline angle and intensity of some peaks (as also observed in figure 1), has compromised the functioning of the network that is not able to reposition them correctly in the new baseline. In the second case, instead, the peak appears to have been correctly predicted, but the noise along the curve did not change significantly and remained in high amounts. Probably also in this case, as in the previous pre-processing, the distortion of the signal obtained from the pre-processing makes the network learn how to correctly reposition the signal, i.e. correct the baseline, but not how to correct the noise along it.
So, in the end, the third preprocessing that was tested is the shifting of the minimum of the signal to 0 and the results obtained on two samples belonging to the same dataset as before are in the following figures 9 and 10:

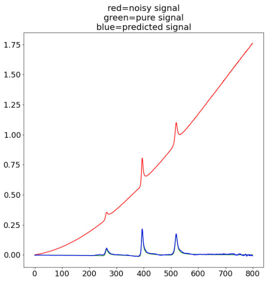

Figure 9: Minimum of signal to zero on sample 1

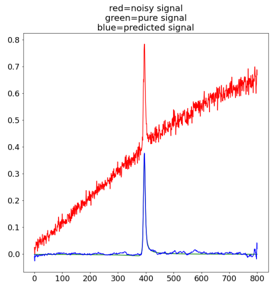

Figure 10: Minimum of signal to zero on sample 2

The results obtained from this latest test appear incredibly good. The model was able to correct the baseline optimally, repositioning the peaks to their optimal value and denoising along the signal itself even when the noise is very high. The latter therefore appears to be the best pre-processing that will be used in quantitative evaluations of all samples in the datasets. To do this, a test set was manually created. The first test carried out reports the following quantities: the number of peaks in the clean signal; the number of peaks correctly predicted, i.e. occurring at the same timestep as the peaks in the clean signal; the number of non-predicted peaks; the number of predicted peaks not corresponding to any peak in the clean signal. Subsequently, a second test was performed on the correctly predicted peaks. Of these, it is verified that the intensity of the peak, i.e. the value along y, was correctly predicted. The prediction is considered correct if it occurs with a maximum error along the x-axis of timesteps ± 4 with respect to the correct value. Similarly,

Table 1
Quantitative results of model trainings with pre-processing shift to zero

Type of dataset	Num. of real peak	Peak correctly predicted	Peak not predicted	Peak incorrectly predicted (added)	Accuracy peak intensity THR 10%-20%	Accuracy peak intensity THR 15%-30%	RMSE	MAPE
Low Noise	3244	2754 (84.90%)	490 (15.10%)	89	83.51%	92.05%	1.15e-2	4.14
Mixed Noise	3278	2775 (84.66%)	503 (15.34%)	100	78.99%	89.23%	1.32e-2	9.13
High Noise	3222	2627 (81.53%)	595 (18.47%)	125	78.72%	90.67%	1.30e-2	17.82
Mixed Noise (17 Aug)	9219	7719 (83.72%)	1500 (16.27%)	1070	83.61%	92.54%	1.55e-2	4.72

the intensity of a prediction is also considered correct if it occurs with a maximum error along the y-axis within a certain threshold. To be exact, there are two thresholds: one for very intense peaks and one for mild peaks. It was decided to use two thresholds because the most relevant peaks for Raman Spectroscopy studies are the higher ones and for them it is necessary that the value obtained is as close as possible to the original one, while for the milder peaks a lighter threshold was set because it is less relevant that they are predicted more accurately. It is not possible to determine a priori how low a peak must be to be considered mild because each signal has different intensities. Therefore, it was decided that a peak is considered mild if its intensity is less than 1/6 of the intensity of the maximum peak within the same sample. To specify these thresholds in the results, they are preceded by the words "THR" and the two values are separated by a "-" where the first value indicates the maximum error for high peaks and the second for mild ones. Finally, root mean square error (RMSE) and mean absolute percent error (MAPE) were included to assess how much the predicted and true signals diverged from each other and to evaluate the accuracy of the predictive model, respectively. They are two alternative metrics to the MSE and MAE used during training, but they compute the same concepts and can be used to compare with work done in other papers. The formulas for these metrics are:

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \widehat{x}_i)^2}{n}}$$

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{x_i - \widehat{x}_i}{x_i}\right|$$

where $\widehat{x}_i$ is the prediction and x_i is the true value.

The results that were obtained in this project through the previously described model and the pre-processing that shifts the minimum noisy signals to 0 can be seen in Table 1 and they are excellent. In each dataset there is a percentage of correctly predicted peaks higher than 82%. Moreover, among them the percentage of peaks with the correct intensity varies from 78% to 92% depending on the dataset and the type of threshold used. The lowest results are obtained in the noisy signal, but this is more normal and acceptable since this type of signal is the most difficult to clean. In addition, incorrectly predicted peaks, i.e. those that do not correspond to any peak in the cleaned signal, were also reported. These peaks are few in almost all cases, but in general it should be noted that such incorrect additional predictions often do not really correspond to a peak in the predicted signal. In fact, the function implemented to detect peaks in the predicted signal is quite sensitive to variations in the signal and is equally complex to set and it is necessary to find the right trade-off to make it detect real peaks and reject false ones. Multiple tests have been performed to find the optimal combination of parameters, and changing them to decrease the number of false predictions would have also compromised the prediction of the correct ones, so it was left the optimal trade-off achieved. As further evidence that it is not the predicted signal that is wrong, we can observe the last two metrics. In fact, the RMSE and MAPE metrics report that the divergence between the predicted signal and the cleaned signal is minimal and the accuracy is high in all datasets. This means that the trends of the predicted signals are very faithful to the corresponding cleaned signals and therefore do not present additional peaks created by a bad denoising of the signal.

Finally, it is possible to compare the results obtained using the RMSE and MAPE metrics. In particular, Table 2 reports the results of the paper [6] where RMSE and MAPE were calculated for their parallel convolutional network and different wavelet noise reduction methods.

Table 2
Quantitative results of papers [6]

Models	RMSE	MAPE
CNN	183.75	13.79
parallelCNNs	129.96	9.50
Empirical Bayes	422.59	27.07
Block James-Stein	489.30	27.57
False Discovery Rate	467.36	27.79
Minimax Estimation	512.98	28.15
Stein's Unbiased Risk Estimate	416.64	34.80
Universal Threshold G	674.92	29.48

By comparing tables 1 and 2, it is evident that the results obtained from our model lead to a large improvement in state-of-the-art. While in table 2 the RMSE always reports values higher than 10^2, all the values obtained in table 1 report results in the order of 10^{-2} and this means that the error between the predicted curve and the real one is much lower. Also the MAPE, which measures the average percentage accuracy of the prediction system, is better in the case of our model. The only dataset that reports worse results with this metric is the one with high noise. However, this behavior can always be justified by the fact that this situation is very particular and difficult to denoise correctly. In addition, we don't know exactly what kind of noise afflicted the dataset in paper 2, so we don't know if it is correct to directly compare the case of high noise with it. In any case, considering that the RMSE metric reports a much smaller error, it is possible to notice that a very good job has been done also in this last case.
Therefore it is possible to deduce that the results obtained are excellent in all the analyzed cases, showing the high performance of the network in very different situations.

7. Conclusion

In this paper, a parallel convolutional neural network model is proposed for denoising a one-dimensional signal that is highly varied in both signal tendencies and the types of noise that affect it. Each network exploits residual learning to separate the noise from the signal and each of them focuses on different features in order to optimize the final results. In addition, pre-processing the dataset using a non-distorting technique of the signal, such as signal shifting near zero, accelerates and greatly aids the model in achieving excellent performance. Unlike other discriminative models that train networks on specific datasets or for specific noise levels, our model has the ability to handle denoising of various types of noise. In fact, experimental results have shown that the proposed method provides excellent signal denoising performance both quantitatively and qualitatively.

Acknowledgments

We would like to thank professors Christian Napoli, Stefano Giagu and tutor Giorgio De Magistris for the support given to this project through their advice and tips which contributed to the achievement of our results.

References

[1] M. D. B. B. Max Lambert, Andrew Engroff, Empirical mode decomposition, https://www.clear.rice.edu/elec301/Projects02/empiricalMode/index.html, 2018.

[2] N. Kumar, A. Siddiqi, K. Alam, Raman spectral data de-noising based on wavelet analysis, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.662.4653&rep=rep1&type=pdf, 2014.

[3] C. Chen, F. Peng, Q. Cheng, D. Xu, Raman spectral data denoising based on wavelet analysis, 2009.

[4] F. León-Bejarano, M. Ramírez-Elías, M. O. Mendez, G. Dorantes-Méndez, M. d. C. Rodríguez-Aranda, A. Alba, Denoising of raman spectroscopy for biological samples based on empirical mode decomposition, https://www.worldscientific.com/doi/abs/10.1142/S0129183117501169, 2017.

[5] Y. Bai, Q. Liu, Denoising raman spectra by wiener estimation with a numerical calibration dataset, https://opg.optica.org/boe/fulltext.cfm?uri=boe-11-1-200&id=423997, 2020.

[6] L. Pan, P. Pipitsunthonsan, P. Zhang, C. Daengngam, A. Booranawong, M. Chongcheawchamnan, Noise reduction technique for raman spectrum using deep learning network, https://ieeexplore.ieee.org/abstract/document/9325766, 2020.

[7] P. Abdolghader, A. Ridsdale, T. Grammatikopoulos, G. Resch, F. Légaré, A. Stolow, A. F. Pegoraro, I. Tamblyn, Unsupervised hyperspectral stimulated raman microscopy image enhancement: denoising and segmentation via one-shot deep learning, https://opg.optica.org/oe/fulltext.cfm?uri=oe-29-21-34205&id=460202, 2021.

[8] Z.-M. Zhang, S. Chen, Y.-Z. Liang, Baseline correction using adaptive iteratively reweighted penalized least squares, https://pubs.rsc.org/en/content/articlelanding/2010/an/b922045c/unauth, 2010.

[9] S. C. Renato Lombardo, Zhi-Min Zhang, Y.-Z. Liang, adaptive iteratively reweighted penalized least squares (airpls), https://github.com/zmzhang/airPLS, 2016. [Online; accessed November-2021].

[10] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising, https://ieeexplore.ieee.org/abstract/document/7839189, 2017.