Polymer 193 (2020) 122341

Contents lists available at ScienceDirect

Polymer

journal homepage: http://www.elsevier.com/locate/polymer

From chemical structure to quantitative polymer properties prediction through convolutional neural networks

Luis A. Miccio [a,b,c,*], Gustavo A. Schwartz [a,b]

[a] Centro de Física de Materiales (CSIC-UPV/EHU) - Materials Physics Center (MPC), P. M. de Lardizábal 5, 20018, San Sebastián, Spain
[b] Donostia International Physics Center, P. M. de Lardizábal 4, 20018, San Sebastián, Spain
[c] Institute of Materials Science and Technology (INTEMA), National Research Council (CONICET), Colón 10850, 7600 Mar del Plata, Buenos Aires, Argentina

ARTICLE INFO

Keywords:
QSPR
Properties prediction
Deep learning
Neural network
Smart design

ABSTRACT

In this work convolutional-fully connected neural networks were designed and trained to predict the glass transition temperature of polymers based only on their chemical structure. This approach has shown to successfully predict the T_g of unknown polymers with average relative errors as low as 6%. Several networks with different architecture or hiperparameters were successfully trained using a previously studied glass transition temperatures dataset for validation, and then the same method was employed for an extended dataset, with larger T_g dispersion and polymer's structure variability. This approach has shown to be accurate and reliable, and does not require any time consuming or expensive measurements and calculations as inputs. Furthermore, it is expected that this method can be easily extended to predict other properties. The possibility of predicting the properties of polymers not even synthesized will save time and resources for industrial development as well as accelerate the scientific understanding of structure-properties relationships in polymer science.

1. Introduction

During the last decades, considerable efforts have been spent in the development of Quantitative Structure Property Relationship (QSPR) models. Clearly, the possibility of predicting the behavior of novel materials (even before synthesizing them) and of understanding how structure is related with the macroscopic properties is of utmost importance for materials designers [1–4]. In particular, the possibility of obtaining these predictions by using only the chemical structure of the polymer's repeating unit (monomer) is especially interesting because no data from experimental measurements or complex calculations are needed. Among the many QSPR modelling methods, the use of artificial neural networks (ANN) has arisen as a very promising and suitable approach for establishing structure-properties relationships, especially with the fast-paced development of computers and graphical processing units (GPU). Since the beginning of artificial neural networks in the 1950s, and especially during their explosion from 1990 onwards [2–6], ANN-based methods have shown successful results in a wide range of problems like atomization energy, band gaps, density and viscosity, and glass transition temperature, among others [5–10]. ANN are also very powerful tools for complementing group contribution correlations, molecular dynamics (MD) simulations and DFT calculations, by saving time and helping to focus calculation efforts on promising candidates. In the last years, convolutional neural networks (CNN) have gained renown as a very powerful approach for solving many exciting technical problems like face recognition, style transfer, medical images recognition and driverless cars, among many others [11–15]. When coupled with fully connected layers (FC) and appropriate activation functions, and by being provided of suitable data on a given test subject, these networks can mine the critical information (features) from a given data set, fit them to the desired property (output) during training, and then generalize the result to other (previously unknown) input data. This ability, and the fact that ANN-based approaches do not involve any additional experimental measurements, can save a considerable amount of time and resources to both researchers and companies.

In this work we focus on the prediction of the glass transition temperature (T_g) of polymers only based on the chemical structure of their monomers. We chose to use the T_g for validating our approach because it is a relevant property of polymers and therefore it is well known and well documented. However, it is worth noting here that this approach can be easily generalized to predict other properties like polarizability or polymer fragility among others.

Several approaches based on ANN models have been published in the last years to predict the T_g of different glass formers. On the one hand,

* Corresponding author. Centro de Física de Materiales (CSIC-UPV/EHU) - Materials Physics Center (MPC), P. M. de Lardizábal 5, 20018, San Sebastián, Spain.
E-mail addresses: luisalejandro_miccio@ehu.es (L.A. Miccio), gustavo.schwartz@csic.es (G.A. Schwartz).

https://doi.org/10.1016/j.polymer.2020.122341
Received 26 November 2019; Received in revised form 28 January 2020; Accepted 27 February 2020
Available online 2 March 2020
0032-3861/

Liu and Cao [6] proposed a model to correlate the T_g of different polymers with four physical quantities (polarizability, orbital energy, thermal energy and total entropy) obtained from density functional theory (DFT) calculations. On the other hand, Cassar et al. [16] have recently published a different approach where they use an ANN to predict the T_g of oxide glasses based on their chemical composition (i.e. the relative amount of each atom in the glass). From a chemical point of view, these two approaches do not explicitly take into account the molecular structure neither the interactions between atoms. In this sense, Joyce et al. [17] presented a fully connected network-based approach, with up to three hidden layers, where the chemical structure of the monomer was partially introduced in the model through a numeric version of a Simplified Molecular Input Line Entry System (SMILES). Thus, they were able to predict the T_g of different polymers based on inputs related to the monomer structure. However, this approach required adding several numbers to each monomer code in order to consider other information such as element period or position in the structure, dramatically increasing the difficulty of the encoding work and automation. In addition, the use of fully connected networks made the increased number of input parameters more costly.

Here we present an ANN model which encode all the composition and structure information contained in the SMILES code into an easily readable binary image in order to predict the T_g of different polymers. The model uses a combination of convolutional neural networks (CNN) coupled with fully connected (FC) layers. This network architecture is able to predict the T_g of unknown polymers with an average relative error of about 6% for both training and tests sets which shows a great level of generalization.

2. Method

In this section we explain the origin and characteristics of the datasets employed for training and testing ANNs capable of predicting T_g. In addition, an explanation on how the data are processed before being fed to the ANN, and how ANN's architecture, parameters and hyperparameters are tuned is also provided.

Datasets. In this work we used two datasets for training and testing ANNs: 1) a dataset composed of approximately 100 polymers (mainly polystyrenes and polyacrylates) and their corresponding T_g values, originally published by Liu et al. [6] (see Table SI1) and 2) an extended version of the former, which includes more than 200 polymers from different families [18–20] (see Table SI2).

Data treatment. In order to take into account the structure and composition of the monomeric units, we added an encoded representation of the monomers by using a Simplified Molecular Input Line Entry System. The SMILES code is a very useful line notation for encoding molecular structures [21,22], that allows transforming the chemical structures into linear strings of SMILES characters. As a second step, we used a one-hot encoding algorithm [23] to transform these strings into a more readable input for the ANNs. Using a *dictionary* with all the existing characters in the SMILES code, we converted the SMILES strings into matrices with only zeros and ones: the row *i* of the matrix is filled with zeros except for the position of the dictionary which coincides with the same character on the *i-th* position in the monomer SMILES code. A one is placed in that case. Therefore, the number of characters in the dictionary (nd) and the length of each SMILES code (npos), define the columns and the rows of the matrix. The process is summarized in Fig. 1. We defined the dictionary as an ordered list of the SMILES characters, as shown below:

Dictionary = **[**'c', 'n', 'o', 'C', 'N', 'O', 'F', 'P', 'S', 'Cl', 'Br', 'I', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '.', '-', '=', '#', '$', ':', '/', '+', ')', '(', '@', '{', '}', '\', ' ', '[', ']'**]**

It is worth noting here that the order of the dictionary is arbitrary. However, this order could eventually affect the performance and the efficiency of the network. The influence on the ANN's performance of the dictionary's order was studied by creating several randomly shuffled dictionaries and repeating the ANN's training and testing processes. This is discussed later in this work.

The matrices obtained from the data encoding process can also be interpreted as a set of mx binary images of dimensions nd x $npos_{max}$ (where $npos_{max}$ is the number of positions in the longest monomer SMILES code and mx is the total number of polymers in the dataset). Fig. 1 shows the encoding process and the resulting binary images dataset. Other examples of the obtained images are shown in Fig. SI1.

ANN's architecture. We employed convolutional [14,24,25] neural networks, fed with the previously constructed images (or matrices) as input. Before passing through the network, it is important to mention that zero padding (a frame of zeroes added to each image) was applied to the data with the aim of ensuring equal weighting of all the pixels during convolutions. Since these binary images directly represent the encoded monomer structure, it can be assumed that some parts of the image (i.e. the structure) are more relevant than others for predicting the T_g. We call features to these parts that contribute more to the structure-T_g relationship, being much more important for the ANN's training process than the rest of the image. Fig. 2 shows a schematic view of the ANN's architecture: the hidden features were extracted from the data by two convolutional layers with varying window size and different number of filters (convolution in Fig. 2), along with ReLU activations and max-poolings [26]. Batch normalization was employed before each activation function in order to reduce the covariance shift. Usually, as a network trains, early layers weights change and as a result, the inputs of later layers suffer large variations. Each layer must then readjust its weights to the varying distribution of every batch of inputs, in turn slowing the training process. The use of batch normalization mitigates this effect. In addition, by normalizing the output of the neurons, the activation function will only receive inputs close to zero, thus ensuring a

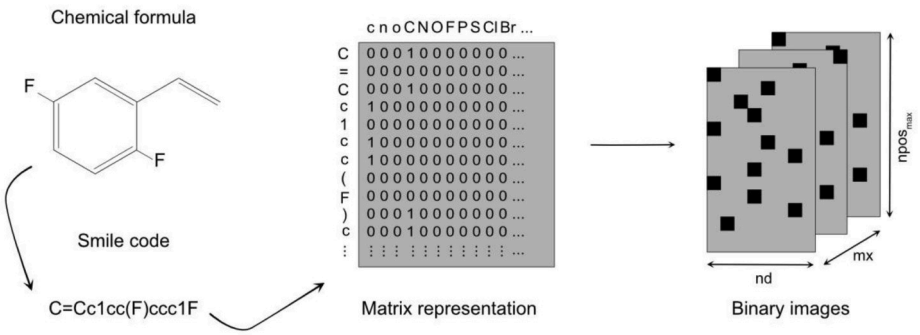

Fig. 1. Each monomer structure is converted into a SMILES string, which in turn is converted into a binary image by employing a dictionary of SMILES characters and one-hot encoding. The dimensions of the images are nd columns (number of SMILES characters in the dictionary) and $npos_{max}$ rows (number of positions in the longest string of the dataset). In addition, mx stands for the number of polymers in the dataset.

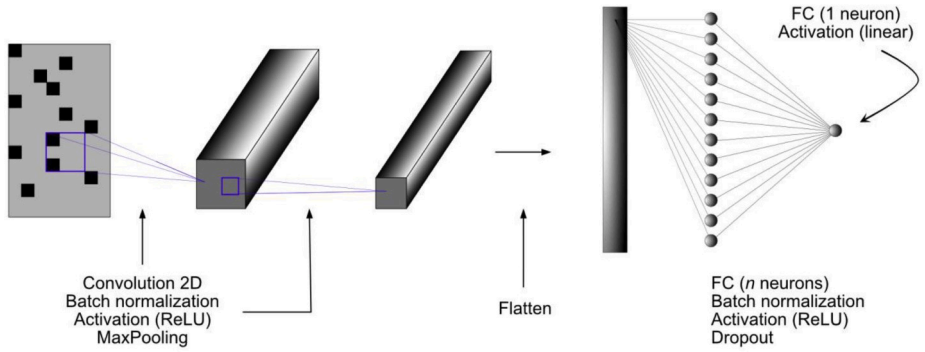

Fig. 2. Schematic picture of the artificial neural network employed for predicting polymer T_g.

non-vanishing gradient.

In order to obtain glass transition temperatures values as the final output, the resulting data tensor was flattened to a vector and passed through a fully connected layers of 100 neurons (see Fig. 2). Batch normalization for covariance shift reducing and ReLU activation functions were also employed in this layer. In addition, the dropout [27] algorithm was employed in order to further prevent overfitting by forcing a spreading of the weights through all the neurons, with dropping probabilities ranging from 0,2 to 0,3. Finally, this layer was connected to a single neuron with a linear activation function responsible of providing the glass transition temperature value.

To ensure equal weighting of lower and higher T_g data values during training, the loss function was defined as the median relative error between the actual (A_i) and forecasted (F_i) glass transition temperatures.

$$Loss = \frac{100}{mx} \cdot \sum_{i=1}^{mx} \left(\frac{A_i - F_i}{A_i} \right)$$

An ADAM optimizer [26] with different learning rates (lr) ranging from 0,0001 to 0,1 was employed for speeding up the convergence during training (beta 1 and beta 2 ranged from 0,1 to 0,99 and 0,1 to 0,999, respectively). The calculations were done by using mini batches ranging from 1 to 256 images (being 256 equal to the gradient descent algorithm given the total amount of data).

ANN's optimization: in order to achieve the best possible performance for the ANNs, different values of the network hyperparameters were explored. In this way, several networks with varying parameter values were trained and compared. This comparison was done in two steps: 1) taking into account the performance achieved on each dataset (i.e: reduced and extended) and 2) by comparison with literature on T_g predictions.

As usual, the dataset was randomly divided into test and train subsets, and no enforcing of any preference in the way the data is divided was applied. The effect of the manner the data is split on the results was estimated by repeating the training and testing several times (parameter: #reps) with different shuffling. Similarly, the above-mentioned influence of the way each dictionary was constructed was also studied, in this case by repeating the training process of each network for several dictionary variations (parameter: #Dict). In summary, for each ANN configuration (#reps) x (#Dict) training repetitions were performed, and the differences in achieved relative errors and dispersion were compared.

Searching for the optimal performance of the ANNs, their hyperparameters were optimized in the following order (other important characteristics of the network were also explored, as shown in Table 1):

1. Learning rate
2. Beta 1 and beta 2
3. Mini batch size
4. # Hidden neurons in the fully connected layer
5. Best dictionary configuration (#Dict)

Table 1
Hyperparameters employed during the training of the different neural networks.

Item	Values
Data split ratio (train/test)	75/25; 80/20; 90/10
Zero padding (frame width in pixels)	0; 3; 5
Dropout probability	0; 0,2; 0,3
Mini batch size	0, 4, 10, 32, 64, 128, 256
Learning rate	0,0001 to 0,01
Beta1 (Beta2)	0,1 to 0,99 (0,1 to 0,999)
# filters 1	8, 16, 32, 64, 128, 256
# filters 2	8, 16, 32, 64, 128, 256
# reps	10, 20, 50
# Dict	20, 30, 50

From the above-mentioned exploration process, a confidence interval for the optimized ANN (typically larger than just the individual mean relative error) was obtained.

3. Discussion

We are working under the premise that the monomer's structure contains enough relevant information for obtaining an accurate prediction of certain polymer properties, provided enough training examples (i.e: the information contained in the monomer representation indirectly and partially accounts for other important parameters, like free volume, inter and intra-chain coupling or tacticity, among others, at least to the extent of the obtained prediction error). Moreover, we are also assuming that the SMILES encoded structure (and therefore its transformation into a binary image) contains roughly the same information. This kind of approach has recently proven to be successful in the analysis of biomolecules [28] and we will see that it is also highly suitable for the aims of the present work. On the other hand, we have chosen in this work to use the glass transition temperature of polymers (above the molecular weight saturation) as a suitable property to validate our approach due to the extensive amount of data and literature, as well as the existence of other trained neural networks examples to compare with [6,8,17].

For the first (restricted) data set, we found average relative errors for the prediction of the T_g using our ANN as low as about 6% for both the training and the test sets. We also found that the dispersion associated to the random splitting of the data (in training and test sets) can reach values of about 3%–5% along repetitions of the same ANN, probably because the generalization becomes harder when "difficult cases" fall in the test set only, i.e. the networks cannot train on these rare examples and therefore generalize poorly afterwards (this point will be further discussed later in this work). Some representative results of the relative error as a function of the number of epochs are presented in Fig. 3. As shown, an error of about 6% is achieved in the best case scenario for the proposed ANN main architecture (2 hidden convolutional layers coupled with 100 hidden neurons fully connected layer), and no

overfitting or relevant systematic difference between train and test errors is observed (while with more neurons overfitting cannot be avoided). The best observed configuration uses (3,3) maxpooling and convolutional windows and 256 filters in the first layer and (3,3) windows with 128 filters in the second. Zero padding of 5 pixels, dropout probability of 20% and mini batches of 15 samples were employed. It can be seen that the training process reaches a plateau just after 200 epochs. It is worth to remind here that we are predicting the T_g only based on the chemical structure as encoded in the SMILES code without any physical data (measured or calculated). Nevertheless, we are getting similar errors to those obtained by Liu et al. [6] (in turn calculated from four physical quantities of the monomers through DFT calculations).

In order to further expand these results, we employed the extended dataset, with several other (more than 200) polymers like Polycaprolactone, Poly(p-phenylene terephthalamide), Poly(phenylsulfone) and Nylon, among many others. As a result, due to the addition of very different chemical structures of the monomers the extended dataset presents a larger dispersion of T_g values (from 178 K up to nearly 600 K instead of the previous 198 K–389 K range). The extended list and their respective SMILES codification and T_g values are shown in Table SI 2. In a first approach, the same ANNs were trained and tested using this dataset. Despite the abovementioned larger dispersion and the presence of data from different distributions (not all the polymers can be treated as belonging to the same family and heterogeneities in the structure-T_g relationship cannot be discarded), the obtained results still present small relative errors in the order of 8%. The relative error value as a function of the epoch number for a representative ANN is shown in Fig. 4.

The error plateau is again reached slightly above 200 epochs, and the best configuration uses (3,3) maxpooling and convolutional windows with 256 filters in the first layer and (3,3) windows with 128 filters in the second. Zero padding of 5 pixels, dropout probability of 20% and batches of 10 images were employed.

Following the hyperparameter tuning order proposed in the method section, and by using these ANNs as starting point, we conducted a learning rate sweep for determining the optimum value. Fig. 5 shows the average relative error obtained for each lr value (the last 25 epochs of each training and test result were employed for calculating the average error and the dispersion).

As shown, the use of lr = 0.008 (with FC = 100 and mb = 10) appears to be the best approach (see arrow in Fig. 5) since it presents the lowest

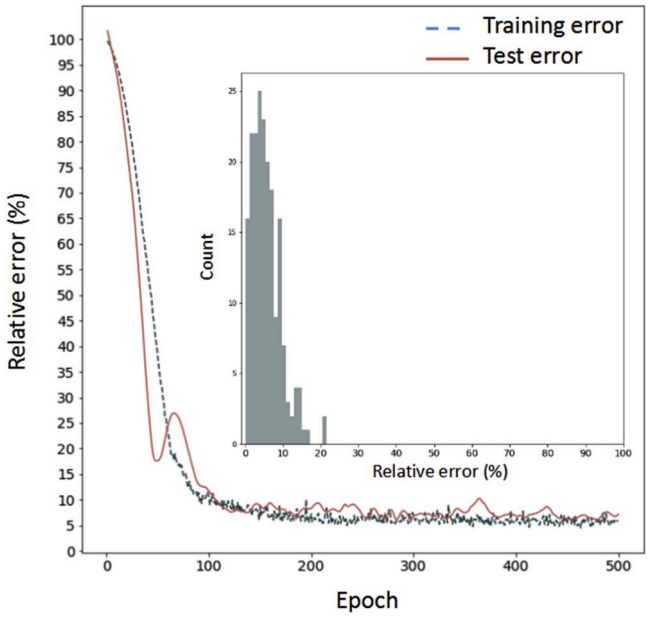

Fig. 3. Average relative error as a function of the epoch number for a selected neural network. Inset: number of counts (polymer subjects) vs relative error.

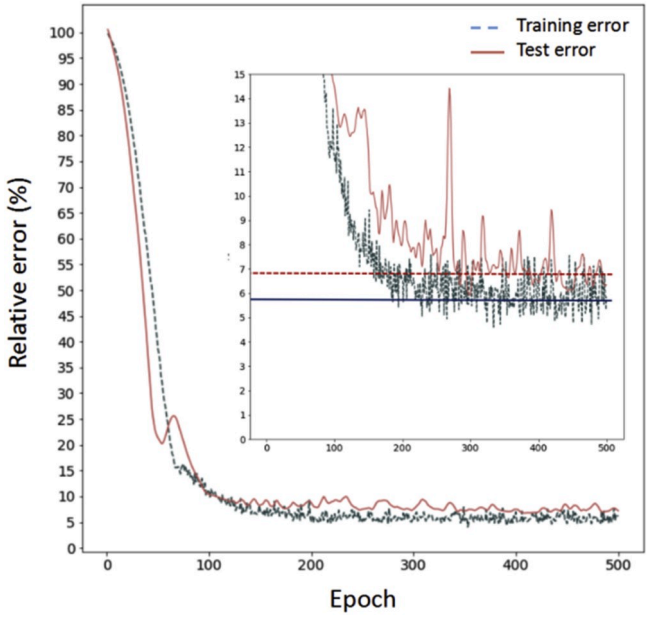

Fig. 4. Relative error as a function of the epoch number for a selected neural network for the extended dataset. The inset shows a zoom of the plateau region.

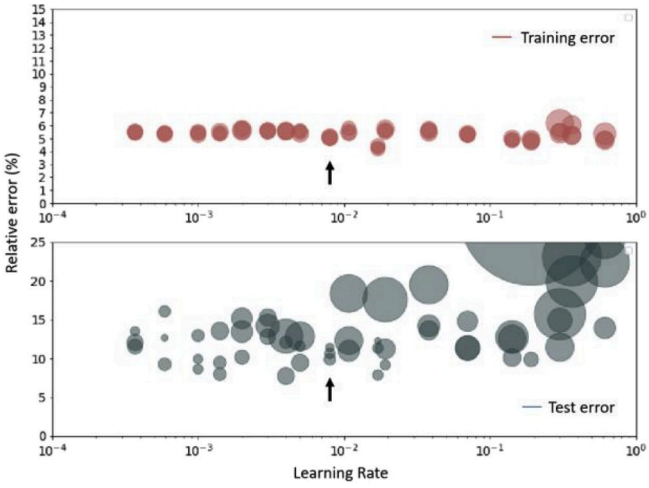

Fig. 5. Average relative error as a function of the learning rate. Data dispersion and size represent the influence of the different dictionaries and data shuffling between train and test sets along repetitions.

overfitting of the data (the difference between train and test errors and the data dispersion are lower than in the other cases). The same sweep process was performed for beta 1 and beta 2 parameters (the obtained results are shown in Fig. SI3). No substantial improvements were obtained from these parameters and the 0,99 and 0,999 values were then left unchanged.

The third optimization step was conducted by sweeping the number of images in the mini batch from 1 to 256. Fig. 6 shows the results of this process where we can observe that there is a noticeable change in the train error from using 1 image to 256 (i.e. all the images in the dataset, which is equivalent to normal gradient descent). There is lower overfitting of the data when using smaller mini batches. The best compromise between lower train and test errors is apparently obtained with mb = 10.

The same optimization procedure was employed for architecture related parameters like number of neurons in the FC layer and number of filters in the convolutional layers. The errors obtained as a function of

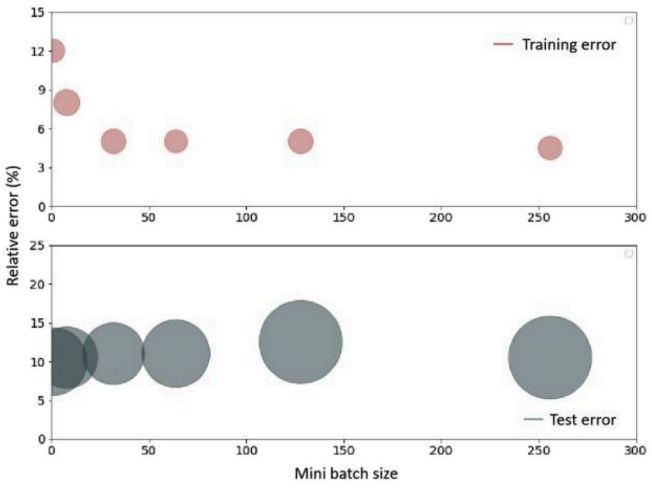

Fig. 6. Average relative error as a function of the number of images in the minibatch for both train and test processes. The size of the dots represents the standard deviation (SD).

these parameters are also presented in Fig. SI3.

The last studied parameters were the influence of the order of the dictionary, #Dict, and the number of repetitions, #reps (related to the sensitivity to the splitting of the dataset into train and test subsets). Fig. 7 shows the average relative error (of up to 5 repetitions of each ANN) as a function of the employed dictionary, for both the train and test sets.

As shown, there are very noticeably differences between the results obtained with, for instance, dictionaries 3 (test error of about 5.0 ± 0.1%) and 46 (test error of about 7 ± 3%). On the one hand, the comparison along the #Dict axis represents the influence of the different dictionaries on the observed performance (i.e.: the dispersion of the observed average relative errors reveals how much the performance can be influenced by the different dictionaries). Therefore, from Fig. 7 (and focusing on the test results) it can be estimated that there could be about 6% dispersion coming from the way the dictionary is constructed. On the other hand, the comparison of the results obtained with the same dictionary shows that there could be another 3% of dispersion in turn produced by the shuffling of the different polymers into the train and test sets.

It is interesting to analyze the position of each character in the dictionaries. As an example, dictionaries 0 and 20 are shown below:

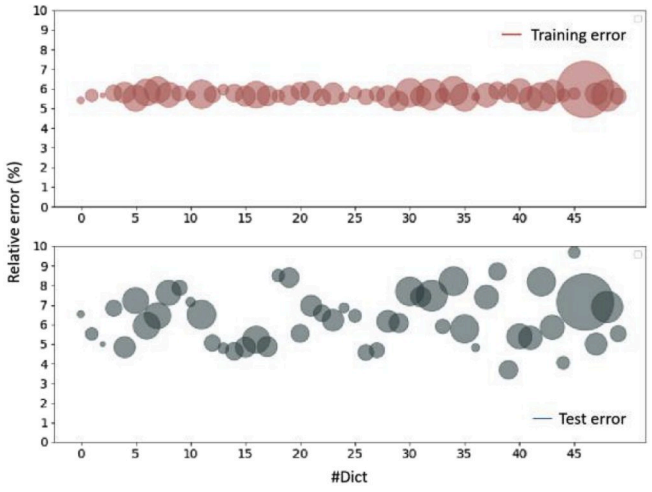

Fig. 7. Average relative error as a function of the dictionary for train and test processes. The size of the dots represents their standard deviation (SD).

Dictionary 0 = **[**‘(’, ‘4’, ‘F’, ‘ ’, ‘$’, ‘n’, ‘ = ’, ‘7’, ‘}’, ‘:’, ‘3’, ‘I’, ‘c’, ‘N’, ‘9’, ‘X’, ‘+’, ‘Y’, ‘2’, ‘@’, ‘P’, ‘d’, ‘5’, ‘1’, ‘{’, ‘#’, ‘O’, ‘]’, ‘o’, ‘8’, ‘0’, ‘6’, ‘C’, ‘S’, ‘-’, ‘/’, ‘[’, ‘)’, ‘.’**]**

Dictionary 20 = **[**‘}’, ‘1’, ‘P’, ‘8’, ‘ ’, ‘Y’, ‘.’, ‘I’, ‘d’, ‘@’, ‘4’, ‘/’, ‘9’, ‘(’, ‘S’, ‘2’, ‘c’, ‘ = ’, ‘6’, ‘7’, ‘-’, ‘o’, ‘)’, ‘O’, ‘$’, ‘n’, ‘F’, ‘{’, ‘X’, ‘3’, ‘[’, ‘:’, ‘5’, ‘C’, ‘0’, ‘]’, ‘+’, ‘N’, ‘#’**]**

The influence of the order of the dictionary can be rationalized when considering the nature of our encoding: features are defined by neighboring pixels. In this way, each change in the position of a given character in the dictionary modify the neighbors of each dot, therefore altering the appearance of these features and making more or less difficult their identification. This can also be interpreted as a comparison of different potential encoding biases (the CNN aims to learn the visual features of the images which better correlate with the output property and the visual structure of these images might be a biased construct). Similar effects were observed for the reduced dataset (see Fig. SI2).

Fig. 8 shows an example of real vs predicted T_g values (K) obtained from an ANN constructed by using dictionary 0. We can observe that most of the polymers lie on (or close to) the “Real = Predicted” line. Besides a few outliers, the average dispersion is lower than that observed in previous works [6] even considering that in this work we are using a broader temperature range and a larger data set. The observed trend appears to imply that the encoded monomer structure is slightly more efficient for capturing relevant features to predict T_g than for example, the predictors used in reference 6. On the other hand, it is worth noticing that the presence of outliers indicate some limitations of the method but at the same time provide clues about how to improve it by extracting more insight into the structure-property relationship from the rare cases. More details about this will be published in future works.

Table 2 shows a few examples of real and predicted values of the glass transition temperature for selected polymers. These values are for both the training and control datasets (polymers unseen by the ANN during training).

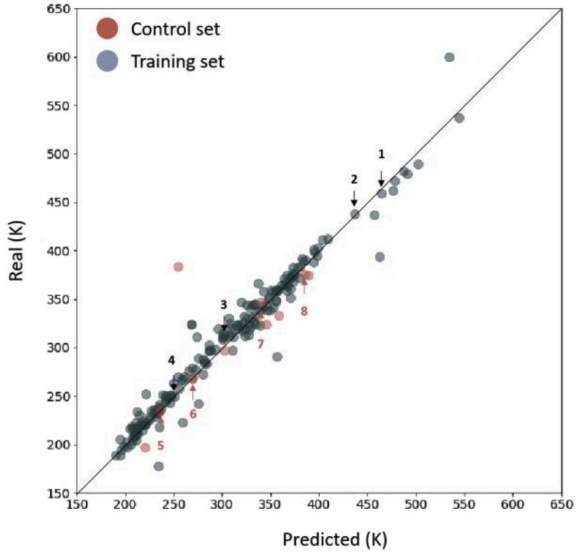

Fig. 8. Example of real vs predicted T_g values (K) obtained from an ANN trained by using dictionary 0. There are two different types of dots, red and blue, standing for control and training data, respectively. Control data have not previously been seen by the ANN, while training data have been employed for error minimization (i.e. training of the ANN). We have employed transparency as a way of highlighting the different density of points at certain image locations. Therefore, dark blue dots indicate higher concentration of data (superposition of several pale blue dots). As guide for the eyes, the black line indicates the “real = predicted” region of the plot. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

In comparison with a pure fully connected neural network approach, and in spite of not needing experimental measurements nor calculations but only the monomer structure for the input, there are other advantages in this convolutional-fully connected approach. The number of parameters is lower due to the use of the convolutional layers and thus the number of calculations is greatly decreased. In addition, the use of max pooling allows further dimension reducing and a more robust feature extraction.

The use of this method can be easily extrapolated to other prediction scenarios, provided that the polymer structure is the main responsible for the property of interest. In addition, this approach can be also used for accelerating industrial materials design as well as scientific understanding of the materials structure-properties relationships. This method can be also of particular interest to any industrial player interested in polymer dynamics, like rubber and tires companies. It is well known that the development of new materials is a process that can be as long as 10 years from first research to first use [4]. The here proposed approach can save a considerable amount of time, resources and money in the development and optimization of polymers and polymer composites.

4. Conclusions

The feasibility of using a mixed convolutional-fully connected neural network architecture for predicting the glass transition temperature of polymers has been demonstrated. Relative errors of about 6% have been observed for both training and test sets and the ANN has shown an excellent level of generalization. This approach relies only on the repeating unit chemical formula and does not require any kind of experimental measurements or calculations as input, therefore becoming a powerful designing tool for material scientists and engineers.

5. Associated content

Additional information on reduced and extended datasets with SMILES codes and glass transition temperatures, codified images, average relative error as a function of the dictionary's order for train and test processes and the corresponding dictionaries can be found in the Supplementary Information (SI) file.

Author contributions

The manuscript was written by LAM and GAS. Both authors have given approval to the final version of the manuscript.

Funding sources

We acknowledge the financial support provided by the Spanish Government "Ministerio de Ciencia e Innovacion" (PID2019-104650 GB-C21) and the Basque Government (IT-1175-19).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Luis A. Miccio: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization, Supervision, Project administration. **Gustavo A. Schwartz:** Conceptualization, Methodology, Formal analysis, Investigation, Writing - review & editing, Visualization, Supervision, Project administration, Funding acquisition.

Table 2
Examples of real and predicted values of the glass transition temperature for selected polymers from both, training (1–4) and control (5–8) datasets.

Monomer	Real T_g [K]	Predicted T_g [K]
1. Poly(bisphenol A isophthalate)	462	465
2. Poly(acrylamide)	438	437
3. Poly(4-methoxymethylstyrene)	350	350
4. Poly(*sec*-butyl acrylate)	250	251
5. Poly(2-methylpentyl acrylate)	235	236
6. Poly(1,2-butadiene)	268	270
7. Poly(2-tertbutylphenyl acrylate)	345	342
8. Poly(4-fluorostyrene)	376	385

Acknowledgment

We gratefully acknowledge the financial support from the Spanish Government "Ministerio de Ciencia e Innovación" (PID2019-104650 GB-C21) and the Basque Government (IT-1175-19). We also acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.polymer.2020.122341.

References

[1] Tu Le, V. Chandana Epa, Frank R. Burden, David A. Winkler, Quantitative structure–property relationship modeling of diverse materials properties | chemical reviews, Chem. Rev. 112 (5) (2012) 2889–2919.
[2] Y. Liu, T. Zhao, W. Ju, S. Shi, Materials discovery and design using machine learning, J. Materiomics 3 (3) (2017) 159–177, https://doi.org/10.1016/j.jmat.2017.08.002.
[3] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, von Lilienfeld, O. Anatole, Klaus-Robert Müller, Alexandre Tkatchenko, Machine learning predictions of molecular properties: accurate many-body potentials and nonlocality in chemical space, J. Phys. Chem. Lett. 6 (12) (2015) 2326–2331.
[4] Nicholas E. Jackson, Michael A. Webb, de Pablo, J. Juan, Recent advances in machine learning towards multiscale soft materials design, Curr. Opin. Chem. Eng. 23 (March) (2019) 106–114.
[5] Z. Zhang, K. Friedrich, Artificial neural networks applied to polymer composites: a review, Compos. Sci. Technol. 63 (14) (2003) 2029–2044, https://doi.org/10.1016/S0266-3538(03)00106-4.
[6] W. Liu, C. Cao, Artificial neural network prediction of glass transition temperature of polymers, Colloid Polym. Sci. 287 (7) (2009) 811–818, https://doi.org/10.1007/s00396-009-2035-y.
[7] F. Hou, Z. Wu, Z. Hu, Z. Xiao, L. Wang, X. Zhang, G. Li, Comparison study on the prediction of multiple molecular properties by various neural networks, J. Phys. Chem. 122 (46) (2018) 9128–9134, https://doi.org/10.1021/acs.jpca.8b09376.
[8] B.E. Mattioni, P.C. Jurs, Prediction of glass transition temperatures from monomer and repeat unit structure using computational neural networks, J. Chem. Inf. Comput. Sci. 42 (2) (2002) 232–240, https://doi.org/10.1021/ci010062o.
[9] Z. Jiang, Z. Zhang, K. Friedrich, Prediction on wear properties of polymer composites with artificial neural networks, Compos. Sci. Technol. 67 (2) (2007) 168–176, https://doi.org/10.1016/j.compscitech.2006.07.026.
[10] G.A. Schwartz, Prediction of rheometric properties of compounds by using artificial neural networks, Rubber Chem. Technol. 74 (1) (2001) 116–123, https://doi.org/10.5254/1.3547632.
[11] S.-C.B. Lo, H.-P. Chan, J.-S. Lin, H. Li, M.T. Freedman, S.K. Mun, Artificial convolution neural network for medical image pattern recognition, Neural Network. 8 (7) (1995) 1201–1214, https://doi.org/10.1016/0893-6080(95)00061-5.
[12] S.B. Lo, S.A. Lou, J.S. Lin, M.T. Freedman, M.V. Chien, S.K. Mun, Artificial convolution neural network techniques and applications for lung nodule detection, IEEE Trans. Med. Imag. 14 (4) (1995) 711–718, https://doi.org/10.1109/42.476112.
[13] Y. LeCun, K. Kavukcuoglu, C. Farabet, Convolutional networks and applications in vision, in: Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2010, pp. 253–256, https://doi.org/10.1109/ISCAS.2010.5537907.
[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: F. Pereira, C.J.C. Burges, L. Bottou, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.
[15] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017. ArXiv170404861 Cs.

[16] D.R. Cassar, A.C.P.L.F. de Carvalho, E.D. Zanotto, Predicting glass transition temperatures using neural networks, Acta Mater. 159 (2018) 249–256, https://doi.org/10.1016/j.actamat.2018.08.022.
[17] S. Joyce, D. Osguthorpe, J. Padget, G. Price, Neural network prediction of glass-transition temperatures from monomer structure, J. Chem. Soc. Faraday. Trans. 91 (1995) 2491–2496, https://doi.org/10.1039/FT9959102491.
[18] Plastic Library, Chemical retrieval on the web, CROW. https://polymerdatabase.com. (Accessed 17 August 2019).
[19] D.J. Plazek, K.L. Ngai, The glass temperature, in: J.E. Mark (Ed.), Physical Properties of Polymers Handbook, Springer New York, New York, NY, 2007, pp. 187–215, https://doi.org/10.1007/978-0-387-69002-5_12.
[20] George Wypych, Handbook of Polymers, second ed., Elsevier, 2016 https://doi.org/10.1016/C2015-0-01462-9.
[21] D. Weininger, SMILES, a chemical language and information System. 1. Introduction to methodology and encoding rules, J. Chem. Inf. Comput. Sci. 28 (1) (1988) 31–36, https://doi.org/10.1021/ci00057a005.
[22] N.M. O'Boyle, Towards a universal SMILES representation - a standard method to generate canonical SMILES based on the InChI, J. Cheminf. 4 (1) (2012) 22, https://doi.org/10.1186/1758-2946-4-22.
[23] H. Alkharusi, Categorical variables in regression analysis: a comparison of dummy and effect coding, Int. J. Educ. 4 (2) (2012) 202–210, https://doi.org/10.5296/ije.v4i2.1962.
[24] Matthew D. Zeiler, Rob Fergus, Visualizing and understanding convolutional networks, in: Lecture Notes in Computer Science, vol. 8689, Springer, Cham, 2014, https://doi.org/10.1007/978-3-319-10590-1_53.
[25] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014. ArXiv14091556 Cs.
[26] H. Wu, X. Gu, Max-pooling Dropout for Regularization of Convolutional Neural Networks, 2015. ArXiv151201400 Cs.
[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Dropout Salakhutdinov, A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[28] M. Hirohara, Y. Saito, Y. Koda, K. Sato, Y. Sakakibara, Convolutional neural network based on SMILES representation of compounds for detecting chemical motif, BMC Bioinf. 19 (19) (2018) 526, https://doi.org/10.1186/s12859-018-2523-5.