



Encouraging an appropriate representation simplifies training of neural networks

Krisztian BUZA^{a,b}

^a Faculty of Informatics
Eötvös Loránd University (ELTE)
Budapest, Hungary

^b Center for the Study of Complexity
Babeş-Bolyai University
Cluj Napoca, Romania
email: buza@biointelligence.hu

Abstract. A common assumption about neural networks is that they can learn an appropriate internal representation on their own, see e.g. end-to-end learning. In this work we challenge this assumption. We consider two simple tasks and show that the state-of-the-art training algorithm fails, although the model itself is able to represent an appropriate solution. We will demonstrate that encouraging an appropriate internal representation allows the same model to solve these tasks. While we do not claim that it is impossible to solve these tasks by other means (such as neural networks with more layers), our results illustrate that integration of domain knowledge in form of a desired internal representation may improve the generalization ability of neural networks.

1 Introduction

Traditionally, the applications of machine learning algorithms were closely coupled with careful feature engineering requiring extensive domain knowledge. In contrast, in the era of deep learning, a common assumption is that neural

Computing Classification System 1998: I.2.6

Mathematics Subject Classification 2010: 68T07

Key words and phrases: neural networks, representation, integration of domain knowledge

networks are able to learn appropriate representations that may be better than the features defined by domain experts, see end-to-end learning for self-driving cars [4], speech recognition [2] and other applications [7, 14].

Recent success stories related to neural networks include mastering board games [15], the diagnosis of various diseases, such as skin cancer [5], retinal disease [6] and mild cognitive impairment [9]. Despite these (and many other) spectacular results, there is increasing evidence indicating that neural networks do not learn the underlying concepts: minor alterations of images, that are invisible to humans, may lead to erroneous recognition [16, 12], e.g., slight modifications of traffic signs “can completely fool machine learning algorithms” [1].

In this paper we will demonstrate that encouraging an appropriate representation may substantially simplify the training of (deep) neural networks. In particular, we consider two simple tasks and show that the state-of-the-art training algorithm fails, although the model itself is able to represent an appropriate solution. Subsequently, we will see that encouraging an appropriate internal representation allows the same model to solve the same tasks. The resulting networks not only have good generalization abilities, but they are more understandable to human experts as well.

The remainder of the paper is organised as follows: next, we will explain what we mean by “encouraging a representation” (Section 2). In Section 3 and Section 4 we will demonstrate in the context of two transformation tasks that the proposed idea may substantially improve the accuracy of the model. Finally, we discuss the implications of our observations to other applications in Section 5.

2 Encouraging a representation

With “encouraging a representation” we mean to train a neural network in a way that some of the hidden units correspond to predefined concepts. We would like to emphasize that this requirement is meant for a relatively small subset of all the hidden units. For example, if we want to train a network for the recognition of traffic signs in images, the network is likely to have thousands of hidden units and we may require that some of those hidden nodes recognize particular shapes or letters, i.e., their activations should be related to the presence of a triangle, an octagon, or particular letters like ‘S’, ‘T’, ‘O’ and ‘P’.

The idea of encouraging a representation may be implemented in various ways: for example, in the cost function, we may include a regularisation term that penalises the situation if the activation of some given nodes is inappropriate. In one of our previous works, in the context of matrix factorisation for drug-target interaction prediction, we encouraged a lower dimensional representation so that the distances between drugs and targets are in accordance with their chemical and genomic similarities [13].

In this paper, we consider encoder-decoder networks. In particular, we will “encourage” the encoder to learn a pre-defined representation. This representation is the output of the encoder and serves as the input of the decoder. We demonstrate that encouraging an appropriate representation may lead to a substantial increase of the accuracy.

3 TraNet: a network for translation and transcription

We consider two tasks, Translation and Transcription (see Section 3.1). We try to solve both of these tasks with the same neural network, called *TraNet* (as “Tran” is a common prefix of the names of these two tasks). When solving the two tasks, the only difference is the input layer in accordance with the input of the tasks.

We implemented all the experiments in Python using numpy, matplotlib and TensorFlow with the keras API. In order to assist reproducibility, our implementation is available at <http://www.biointelligence.hu/encourage/>.

3.1 Benchmark tasks: translation and transcription

As benchmark tasks, we consider the translation of written numbers from English to German (e.g. *twenty-five* should be translated to *funfundzwanzig*¹), and recognition of 4-digit handwritten numbers where the desired output is the number written in English, see also Fig 1. For simplicity, we will refer to these tasks as *Translation* and *Transcription*.

As neural networks are used for substantially more complex tasks, see e.g. [18] for a review, we expect them to have an excellent performance in case of our tasks.

¹For simplicity, instead of the German special letters 'ä', 'ö', 'ü' and 'ß', we use 'a', 'o', 'u' and 'ss' respectively.

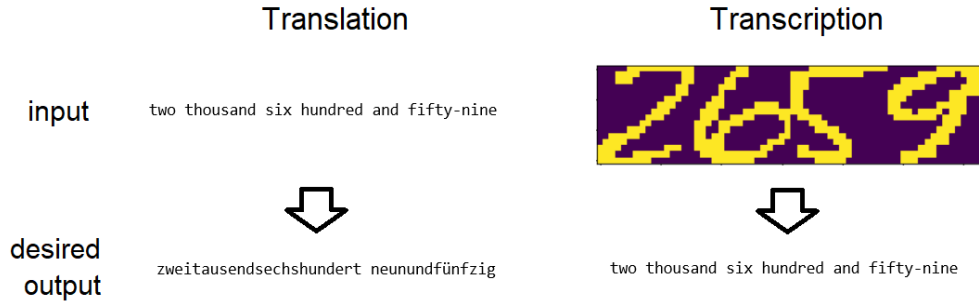


Figure 1: Illustration of the considered tasks, Translation (left) and Transcription (right).

3.2 Architecture

TraNet is a feed-forward encoder-decoder network. For strings (i.e. numbers written in English or German) we use letter-wise one-hot encoding. For example, 'a' is coded as $(1, 0, 0, 0, \dots, 0)$, 'b' is coded as $(0, 1, 0, 0, \dots, 0)$, etc. The entire string is coded as the concatenation of the codes of each letter. We allow at most a length of 50 letters, therefore, the output layer of TraNet contains $50 \times 29 = 1450$ units. Similarly, in the case of the Translation task, the input layer of TraNet contains 1450 units as well. When TraNet is used for Transcription, the input is a binary image with $64 \times 16 = 1024$ pixels, therefore, the input layer contains 1024 units, each one corresponding to one of the pixels. The input layer does not perform any operation, its sole purpose is to represent the input data.

An appropriate internal representation, which *could* be learned by the network, is the *digit-wise one-hot encoding* of the 4-digit number that was shown to the network. Each digit between 0 and 9 is coded as a binary vector of length 10. In particular, digit '0' is coded as $(1, 0, 0, 0, \dots, 0)$, '1' is coded as $(0, 1, 0, 0, \dots, 0)$, etc. The digit-wise one-hot encoding of a 4-digit number is the concatenation of the vectors corresponding to the digits of the number, therefore, it has a total length of $4 \times 10 = 40$.

TraNet contains 3 hidden layers. The first and third layers contain 1000 units with ReLU activation function. The second hidden layer contains 40 units so that it might potentially learn the aforementioned digit-wise one-hot encoding. In the second layer, we use the sigmoid activation function, as it may be suited to the digit-wise one-hot encoding.

As loss function, we use binary cross-entropy. We trained TraNet for 100 epochs using the ADAM optimizer [8]. We performed the computations on CPU (i.e., no GPU/TPU support was used).

4 Results

Next, we compare conventional training and training with encouraging the digit-wise one-hot encoding as internal representation.

4.1 Translation

In the case of Translation task, we considered the numbers between 0 and 9999, written in English (input) and German (desired output). A randomly selected subset of 100 numbers was used as test data, while the remaining 9900 instances were used as training data. We repeated the experiments 5-times with a different initial split of the data into training and test data.

In case of conventional training, denoted as *Conventional TraNet*, the resulting network was not able to give an exact translation for any of the numbers of the test set. While in some cases the translations generated by the network were at least partially understandable to humans, in other cases the network failed to translate the numbers, see Tab 1 for some examples.

Although there are many possibilities to improve the model, next, we show that there is nothing wrong with the model: TraNet itself is able to represent a function that gives a reasonably good translation, the problem is the conventional training.

In order to encourage the model to learn an appropriate representation, we train the encoder and decoder separately. The encoder consists of the input layer and the first two hidden layers. It is trained to translate from English to the digit-wise one-hot encoding (described in Section 3.2), i.e., we expect the output of the encoder to be the digit-wise one-hot encoding. The input of the decoder is the second hidden layer of TraNet. Additionally, the third hidden layer of TraNet and its output layer belong to the decoder. The decoder is trained to translate from digit-wise one-hot encoding to German.

When training the encoder and decoder separately, so that the digit-wise one-hot encoding is encouraged, we observed that the network was able to translate on average 95.8 % of the numbers perfectly. From the practical point of view, the quality of the translation is even better, because in many cases when the translation was not perfect, we observed only minor spelling mistakes, such as “einhundert einundneenzig” instead of “einhundert einundneun-

Input	Output of conventional TraNet	Output of encouraged TraNet
one hundred and ninety one	einaaudenaaaiaaaaaaaaaa	ein hundred einundneenzig
four thousand	vietausenazieihundert	viertausendzweihundert
two hundred and twenty-five	aieaanaaaanai	funfundzwanzig
eight thousand	acattausenaaaeihundert	achttausendachthundert
eight hundred and sixty	aieaaaaa	sechzig
seven hundred and sixty-six	aaebenaunaeaaaaaanan- aaaaiaaea	siebenhundert sechshundsechzig

Table 1: Examples for translation of numbers from English to German *with* and *without* encouraging the digit-wise one-hot encoding as internal representation, denoted as “encouraged TraNet” and “conventional TraNet”, respectively.

zig”, or “siebenhundert sechshundsechzig” instead of “siebenhundert sechshundsechzig”. Consequently, encouraging an appropriate representation lead to an accurate model for the translation of numbers from English to German.

4.2 Transcription

In order to obtain 4-digit handwritten numbers, we considered the *Semeion* dataset², which is a publicly available dataset containing images of handwritten digits. As each of the images shows a single digit, in order to obtain an image of a four-digit handwritten number, we choose four images randomly and stack them horizontally, see Fig. 1 for an example. The last 100 images of the Semeion dataset are used to obtain test images, whereas the training data is obtained from the first 1493 images. In total, we obtained 100 000 training images and 1000 test images of 4-digit numbers. We repeated the experiment 5-times with different training and test images.

Conventional training of TraNet, i.e., training without encouraging the digit-wise one-hot encoding as intermediate representation, lead to a network that could not transcribe any of the images correctly. Instead, the output of TraNet was always a number-like phrase without clear meaning, such as “fivet huusadd ne hundded and” or “tivethhusand ne hundred and”. In contrast, encouraging the digit-wise one-hot encoding, i.e., training the encoder and decoder sepa-

²<https://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit>



Figure 2: Examples for transcription *with* and *without* encouraging the digit-wise one-hot encoding as internal representation, denoted as “encouraged TraNet” (bottom) and “conventional TraNet” (center), respectively.

rately, lead to a network that was able to perfectly transcribe on average 74.9 % numbers of the test data, while it made mostly minor spelling mistakes in the remaining cases, see Fig. 2 for some illustrative examples.

5 Discussion

While we do not claim that there are no other ways to find an acceptable solution, in fact, we observed that using more deep neural networks achieved comparable performance to that of encouraged TraNet³, our results clearly demonstrate the power of encouraging the digit-wise one-hot encoding.

Besides allowing a relatively simple model to achieve good generalisation ability, our encouraged models are better understandable to domain experts. On the one hand, this may increase the trust in the model, and it allows to “debug” the model: if the output is incorrect, one could check whether the activations in the 2nd hidden layer are close to that of in case of the desired internal representation.

Although the fact that a neural network may learn concepts that are substantially different from human concepts, may be considered as an advantage in many cases, we argue that it may be worth to use encouraged models together with conventional models. For example, in critical applications (credit scoring, medical diagnosis, etc.), it may be worth to carefully consider cases when conventional and encouraged models disagree.

The computational cost of training conventional and encouraged TraNet is similar: while performing forward propagation, after the calculation of the

³We note that, due to its simplicity, our translation task can be solved perfectly using techniques from the classic theory of formal languages.

activities in the second layer (which is necessary in both cases in order to determine the gradients of the connections in the previous layers), in case of encouraged TraNet, the desired activations (i.e., activations corresponding to the digit-wise one-hot encoding in our case) are propagated to the third layer. In case of backpropagation, the only difference between conventional and encouraged training is that in case of encouraged training, the back-propagated activations of the second layer do not need to be calculated: the desired activations should be used instead. Therefore, the cost of performing one training iteration is asymptotically same in both cases. This is consistent with our observations, according to which encouraged training was slightly ($\approx 5\%$) faster.

Note however that we considered illustrative examples, in which conventional training failed in the sense that it did not lead to any reasonable model. In other cases, in order to achieve comparable accuracy, conventional training may require much more training epochs compared to encouraged training. Thus, encouraged training may be beneficial in terms of the total computational cost.

Although the appropriate representation depends on the underlying task, we believe that the general idea of encouraging models to learn concepts that are similar to human concepts may be beneficial in various applications ranging from the analysis of biomedical signals [10], such as ECG [17] to image classification [11]. Furthermore, in many cases, see e.g. intrusion detection, recent studies used models other than neural networks [3]. Encouraging appropriate representations might allow neural networks to be applied in such cases too.

Last, but not least, we point out that defining an appropriate representation may be seen as a way of integrating domain knowledge into the training procedure, thus it can be seen as a way of collaboration between human intelligence and machine intelligence.

Acknowledgements

K. Buza was supported by the project ED_18-1-2019-0030. Project no. ED_18-1-2019-0030 (Application domain specific highly reliable IT solutions subprogramme) has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme funding scheme. K. Buza received the “Professor Ferencz Radó” Fellowship of the Babeş-Bolyai University, Cluj Napoca, Romania.

References

- [1] E. Ackerman, Slight street sign modifications can completely fool machine learning algorithms, *IEEE Spectrum* 6 (2019). [⇒ 103](#)
- [2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., [Deep speech 2: end-to-end speech recognition in english and mandarin](#), in: *International conference on machine learning*, 2016, pp. 173–182. [⇒ 103](#)
- [3] M. Antal, E. Egyed-Zsigmond, Intrusion detection using mouse dynamics *IET Biometrics* 8 (5) (2019) 285–294. [⇒ 109](#)
- [4] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al., End to end learning for self-driving cars, *arXiv* (2016) arXiv:1604.07316. [⇒ 103](#)
- [5] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, *Nature* 542 (2017) 115. [⇒ 103](#)
- [6] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O’Donoghue, D. Visentin, et al., Clinically applicable deep learning for diagnosis and referral in retinal disease, *Nature medicine* 24 (2018) 1342. [⇒ 103](#)
- [7] A. Gordo, J. Almazan, J. Revaud, D. Larlus, End-to-end learning of deep visual representations for image retrieval, *International Journal of Computer Vision* 124 (2017) 237–254. [⇒ 103](#)
- [8] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv* (2014) arXiv:1412.6980. [⇒ 106](#)
- [9] R. J. Meszlényi, K. Buza, Z. Vidnyánszky, Resting state fmri functional connectivity-based classification using a convolutional neural network architecture, *Frontiers in neuroinformatics* 11 (2017) 61. [⇒ 103](#)
- [10] K. Miok, D. Nguyen-Doan, M. Robnik-Sikonja, D. Zaharie, [Multiple Imputation for Biomedical Data using Monte Carlo Dropout Autoencoders](#), in: *E-Health and Bioengineering Conference (EHB)* (2019) [⇒ 109](#)
- [11] T. Nyíri, A. Kiss, [Novel Ensembling Methods for Dermatological Image Classification](#), in: *International Conference on Theory and Practice of Natural Computing* (2018) [⇒ 109](#)
- [12] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, A. Swami, [Practical black-box attacks against machine learning](#), in: *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, ACM, 2017, pp. 506–519. [⇒ 103](#)
- [13] L. Peška, K. Buza, J. Koller, Drug-target interaction prediction: A bayesian ranking approach, *Computer methods and programs in biomedicine* 152 (2017) 15–21. [⇒ 104](#)

-
- [14] D. Silver, H. van Hasselt, M. Hessel, T. Schaul, A. Guez, T. Harley, G. Dulac-Arnold, D. Reichert, N. Rabinowitz, A. Barreto, et al., [The predictron: end-to-end learning and planning](#), in: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 3191–3199. ⇒ 103
 - [15] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, *Nature* 529 (2016) 484. ⇒ 103
 - [16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, *arXiv* (2013) arXiv:1312.6199. ⇒ 103
 - [17] S. M. Szilagyi, L. Szilagyi, D. Iclanzan, Z. Benyó, [Unified Neural Network Based Adaptive ECG Signal Analysis and Compression](#), *Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Automatic Control and Computer Science* 51.65 (2006) 27–36. ⇒ 109
 - [18] J. Zhang, C. Zong, et al., Deep neural networks in machine translation: An overview, *IEEE Intelligent Systems* 30 (2015) 16–25. ⇒ 104

Received: March 26, 2020 • Revised: June 7, 2020