

AI-Friendly.com: Artificial Intelligence Made Friendly

Ezequiel Alvarez¹, Cesar Miquel², and Gaston Mazzei^{1,3}

¹ International Center for Advanced Studies, San Martin, PBA 1650, AR

² EasytechGreen, Washington, DC 20009, USA

³ Universidad de Buenos Aires, Ciudad Universitaria, CABA 1428, AR

Abstract. AI-Friendly is a web application that allows to easily perform binary classification with a neural network. It is free and allows network customization, making it a useful tool for machine learning courses.

Keywords: neural network · classification · artificial intelligence .

“AI-Friendly.com” is an open-source website[1] that allows the customization of a neural network and its application to user-defined binary classification tasks. As it is free of charge, trained inside a server, and requires Microsoft Excel files, it is accesible to generic unspecialized users. There exist several offline software for creating Artificial Neural Networks via a GUI [2–4], as well as free online tools that use fixed toy datasets [5, 6]. Nevertheless, the authors do not know about any free online embeddings of the former, nor generalizations to user-provided datasets of the latter.

As shown in Fig. 1 (left), only three steps are required to use *ai-friendly.com*: (1) provide a “LearnCard”, i.e. a labeled dataset used for training the neural network, (2) an “InCard”, i.e. an unlabeled dataset over which to make predictions, and (3) an e-mail address to which a link to a website with results will be sent to. As shown in Fig. 1 (right), the results displayed include a ROC curve computed over a portion of the training dataset exclusively reserved for validation purposes. Other results include the training and validation curves for the accuracy and loss as a function of the number of epochs.

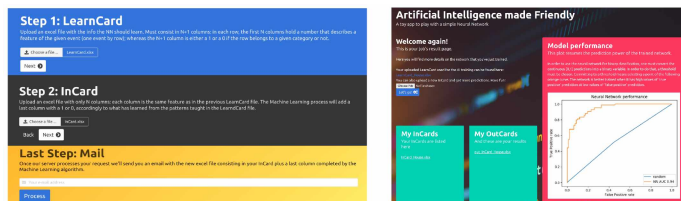


Fig. 1. (left) Zoomed-out view of “AI-Friendly.com” showing that three steps are required to use it. (right) A section of the website with results, where a ROC curve is shown together with a text that assists the interpretation by unspecialized users.

2 Ezequiel Alvarez, Cesar Miquel, and Gaston Mazzei

In the Fig. 2 the advanced customization menu is displayed. Users can choose the optimization algorithm, perform operations over the features, set the amount of neurons per layer, choose the activation function, set the number of Epochs, the batch size and the learning rate. Available optimization algorithms are SGD and Adam[7], while the available operations over the features, in the order in which they will be applied, consist of the following: “*Sin(x)*” duplicates the features by adding the result of applying the sine function to each column, “*Quad*” duplicates the features by adding the result of elevating to the power of two each column, and “*Norm*” removes the feature’s dimensions by dividing by its standard deviation. After all the selected operations, a MinMax scaling is performed to set the range of the data from 0 to 1. If “*Noisy*” is selected, normally distributed noise with zero mean and 50% each feature’s standard deviation is added. Finally “*Conv*” allows to add two convolutional 1-Dimensional layers with zero-padding, 6 filters, a stride of 1 and a kernel size of 3 which corresponds to interpreting the input features as a one-dimensional time series.

Running and editing the app locally, which requires knowledge of several open-source technologies such as Docker, Python, and to a lesser extent Nginx, Jenkins, and uWSGI, allows users to become contributors as the authors are willing to update the website with new properly-tested code. Suggested ideas for future contributions include adding network evaluation metrics to the results and allowing both a variable number of layers and regularization via dropout.

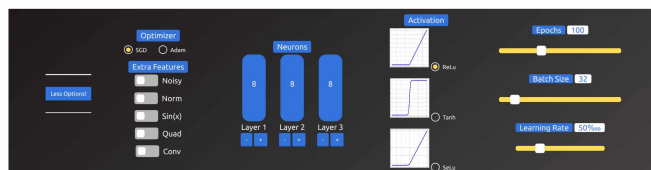


Fig. 2. AI-Friendly.com’s advanced network customization menu.

References

1. Miquel C., Alvarez. E, Mazzei G. (2020). “<https://github.com/GastonMazzei/ai-friendly.com>”, retrieved 20 June 2021
2. Hrnjica B., “<https://github.com/bhrnjica/anndotnet>”, retrieved 16 August 2021
3. Deep Learning Studio [Computer software]. “<https://deecognition.ai/deep-learning-studio/>”, retrieved 16 August 2021
4. Deep Network Designer [Computer software]. “<https://www.mathworks.com/help/deeplearning/gs/get-started-with-deep-network-designer.html>”, retrieved 16 August 2021
5. Karpathy A., “<https://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html>”, retrieved 20 June 2021
6. Smilkov D., Carter S., “<https://playground.tensorflow.org>”, retrieved 20 June 2021
7. Zhang, M. et al. (2019). Lookahead Optimizer: k steps forward, 1 step back.