Efficient data augmentation using graph imputation neural networks

I. Spinelli, S. Scardapane, M. Scarpiniti, and A. Uncini

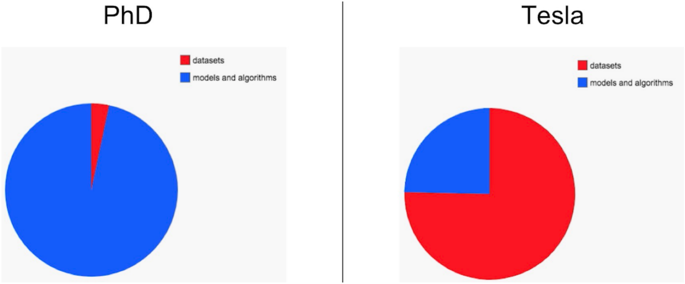
Sapienza University of Rome
ISPAMM Lab

Introduction

Data set importance in ML

The availability of high-quality training data sets is a key factor for running deep learning models in the real world.

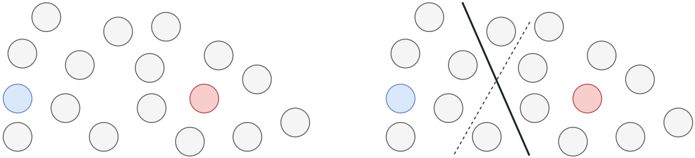
Amount of lost sleep by Andrej Karpathy over...



Semi-Supervised learning

Labelling the data set is the most difficult and expensive phase.

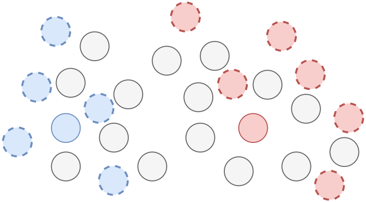
Semi-supervised learning (SSL) studies how to exploit vast amounts of unlabelled data to improve the performances of a model trained on a smaller number of labelled data points[1].

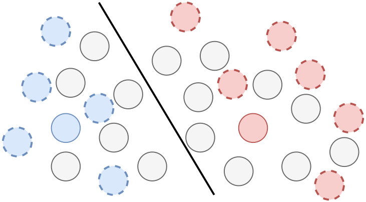


[1] Chapelle, Schlkopf, and Zien. *Semi-Supervised Learning*. 2010.

Data augmentation in SSL

Semi-Supervised data augmentation has the potential to provide significant boosts in accuracy for machine learning models.





Our contribution

Our contribution

We propose a new method to perform data augmentation for general vectorial data sets.

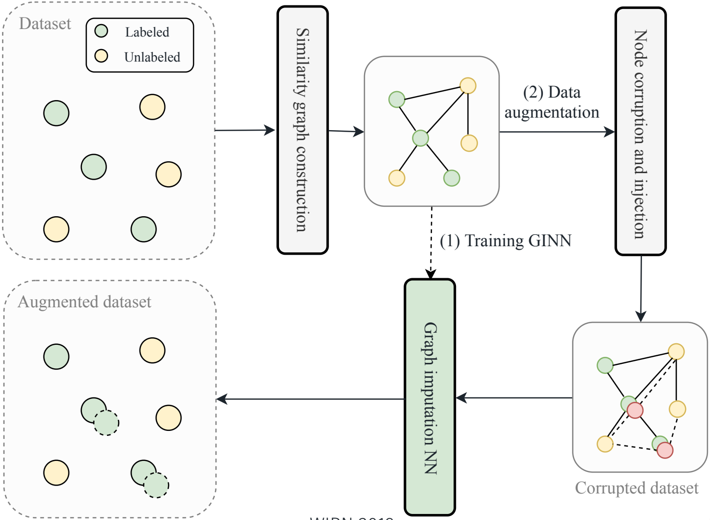
We reformulated the problem of data augmentation as a problem of data imputation under extreme level of noise.

With this reformulation we can use GINN[2](Graph Imputation Neural Network), our new framework for missing data imputation.

[2] Spinelli, Scardapane, and Uncini. “Missing Data Imputation with Adversarially-trained Graph Convolutional Networks”. 2019.

Our contribution

Overall schema of our data augmentation pipeline:



Graph Imputation Neural Network (GINN)

GINN's inner mechanism can be summarized in two main steps:

- build a similarity graph describing the structural proximity between samples.
- train adversarially a customized graph autoencoder to impute the missing values.

Similarity graph

We encode each feature vector as a node in a graph G.

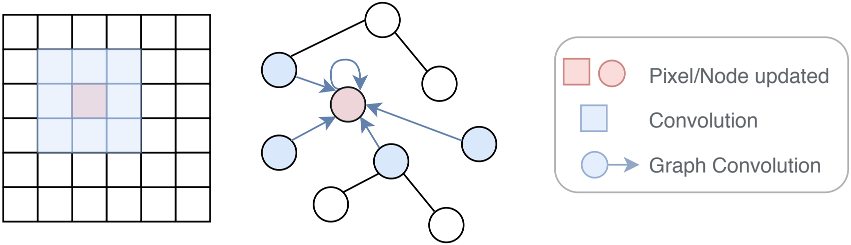
The adjacency matrix A of the graph is derived from a similarity matrix S containing the pairwise Euclidean distances of the feature vectors.

In order to keep only the most relevant edges, we apply a two-step pruning on S.

Graph convolutional layer

The graph convolutional layer[3] is the fundamental building block of our graph autoencoder.

2D Convolution vs Graph Convolution



[3] Kipf and Welling. “Semi-supervised classification with graph convolutional networks”. 2017.

Graph autoencoder

Graph imputation NN

$$
\begin{aligned}
\mathrm{H} &= \mathrm{ReLU}\left(\mathrm{L}\mathrm{X}\boldsymbol{\Theta}_1\right) \\
\widehat{\mathrm{X}} &= \mathrm{Sigmoid}\left(\mathrm{L}\mathrm{H}\boldsymbol{\Theta}_2 + \widetilde{\mathrm{L}}\mathrm{X}\boldsymbol{\Theta}_3\right)
\end{aligned}
$$

- X and $\widehat{\mathrm{X}}$ are the corrupted input and imputed output.
- H is the intermediate representation.
- L is a normalized version of the graph Laplacian.
- $\widetilde{\mathrm{L}}$ propagates the information like L, but without the self-loop.
- $\boldsymbol{\Theta}_*$ are the matrices of adaptable coefficients.

Corrupt, Inject, Impute

To perform the data augmentation we apply three additional steps:

Corrupt, Inject, Impute

To perform the data augmentation we apply three additional steps:

1. Corrupt randomly some of the labelled nodes by removing up to 80% of their features.

Corrupt, Inject, Impute

To perform the data augmentation we apply three additional steps:

1. **Corrupt** randomly some of the labelled nodes by removing up to 80% of their features.
2. **Inject** these new nodes in the graph recomputing on-the-fly their connections with unlabelled nodes. Only the non-zero elements of the corrupted nodes are used for the computation.

Corrupt, Inject, Impute

To perform the data augmentation we apply three additional steps:

1. **Corrupt** randomly some of the labelled nodes by removing up to 80% of their features.
2. **Inject** these new nodes in the graph recomputing on-the-fly their connections with unlabelled nodes. Only the non-zero elements of the corrupted nodes are used for the computation.
3. **Impute** the missing feature of these new nodes using the previously trained GINN architecture, generating new labelled samples that can be added to the data set.

Experimental evaluation

Experimental evaluation

Our experimental evaluation shows improvements in accuracy when training standard supervised learning algorithms on the augmented versions of the data set.

This happens for small augmentation up to increments of 10x the size of the original data set.

As will be shown later, these improvements range from less than a percentage point up to an increment of 24 percentage points.

Experimental evaluation

For the evaluation, we used 6 classification data set, taken from the UCI Machine Learning Repository[4], with numerical, categorical and mixed feature vectors.
We tracked the performances of 5 different classifiers.

Classifiers

- logistic regression
- k nearest neighbor
- support vector machine
- random forest
- neural network

Data sets

- abalone
- heart
- ionosphere
- phishing
- tic-tac-toe
- wine-quality

[4] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.

Experimental evaluation

In our experiments, we divide our data between the training set, 70%, and test set, 30%.

Only 10% of the training set has labels associated with feature vectors (SSL).

We create 3 different augmented training sets; respectively having 2x, 5x and 10x more labelled data.

We train the classifiers on this 4 different training sets and compute the accuracy over the test set.

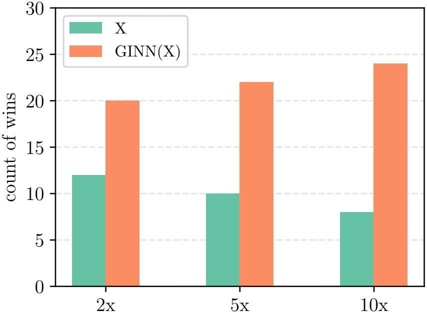
We repeat this procedure for 5 times an report the average in the results.

Results

Dataset	Classifier	Baseline	(2x)	(5x)	(10x)
abalone	LOG	52.87	52.54	52.47	**54.50**
	k-NN	**52.07**	**52.07**	**52.07**	**52.07**
	SVC	**52.87**	**52.87**	**52.87**	**52.87**
	RF	51.53	51.18	52.38	**53.67**
	MLP	50.53	52.66	52.03	**54.78**
heart	LOG	**76.92**	70.77	70.77	66.59
	k-NN	58.24	**64.40**	**64.40**	62.56
	SVC	55.88	**56.04**	**56.04**	**56.04**
	RF	79.56	**81.10**	80.44	78.02
	MLP	65.71	**66.81**	63.96	61.32
ionosphere	LOG	78.30	**80.94**	79.06	78.49
	k-NN	66.04	**90.57**	**90.57**	**90.57**
	SVC	64.15	85.09	84.34	**85.28**
	RF	83.96	**87.92**	85.47	86.23
	MLP	**90.57**	88.87	86.04	86.04

Results

Here we show the number of times the default and the augmented data sets had a better classification performances considering all data sets and all classifiers in the benchmark.



Future work

Future work

The method lends itself to a variety of improvements:

Future work

The method lends itself to a variety of improvements:

- extension to images and audio

Future work

The method lends itself to a variety of improvements:

- extension to images and audio
- data augmentation to fix unbalanced classes

Future work

The method lends itself to a variety of improvements:

- extension to images and audio
- data augmentation to fix unbalanced classes
- data augmentation as regularization strategy

Future work

The method lends itself to a variety of improvements:

- extension to images and audio
- data augmentation to fix unbalanced classes
- data augmentation as regularization strategy

Bridging two different fields, data augmentation and data imputation, has high potential for cross-fertilization.

Thank you!