Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Histogram-based Deep Neural Network for Quantification

LQ2021 - Workshop on Learning to Quantify

University of Oviedo

Pablo González   Juan José del Coz

November 1, 2021

Deep Neural Networks for Quantification

What is HistNet?

HistNet

HistNet is a **deep neural network** designed for solving **quantification** problems

Quantification (binary version)

We are trying to learn a model that is able to **estimate the proportion** of the positive and negative classes. This is, given a training set $D^{tr} = \{(x_i^{tr}, y_i^{tr})\}_{i=1}^{n} \sim S$,

where, $\quad x_i^{tr} \in \mathcal{X}, \quad y_i^{tr} \in \mathcal{Y} = \{-1, +1\}$,

we try to learn a model $\bar{h}$ that:

$$\bar{h} : \mathbb{N}^{\mathcal{X}} \longrightarrow [0, 1]$$

Previous work

There are two attemps to use deep learning for quantifying:

- QuaNet[1]
- DQN[2]

[1] Andrea Esuli, Alejandro Moreo Fernández, and Fabrizio Sebastiani. "A Recurrent Neural Network for Sentiment Quantification". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. CIKM '18. Torino, Italy: Association for Computing Machinery, 2018, pp. 1775–1778. ISBN: 9781450360142. DOI: 10.1145/3269206.3269287.

[2] Lei Qi et al. "A Framework for Deep Quantification Learning". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Frank Hutter et al. Cham: Springer International Publishing, 2021, pp. 232–248. ISBN: 978-3-030-67658-2.

QuaNet [Esuli et al. 2018]

QuaNet

QuaNet is a **recurrent neural network** for binary quantification

How does it work?

It sorts the examples in a sample based on a **classifier** output and then uses an LSTM to learn the **cut point** between the positives and the negatives

DQN [Qi et al. 2021]

DQN

Is a framework for deep quantification learning

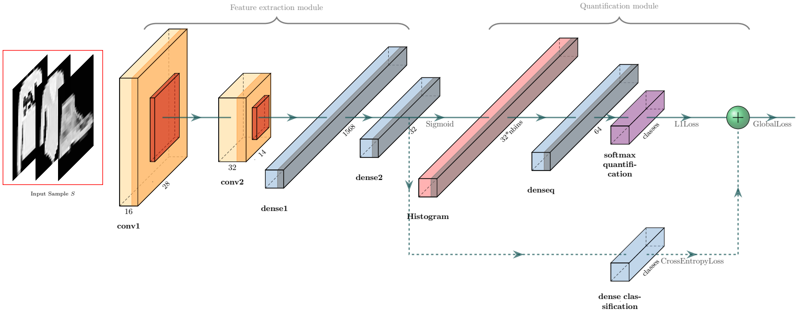
How does it work?

There is a first part of the network that transforms the input into features (**feature extraction layer**)

The second part of the network, **summarizes all the examples in a sample**, using a function like:

- CAT
- AVG, MIN, MAX
- NN

HistNet Architecture



Differentiable Histograms

Histnet uses a **histogram** as the sample representation layer

But wait, histograms are not **differentiable**, aren't they?

HistNet uses an approximation, called **differentiable histograms**

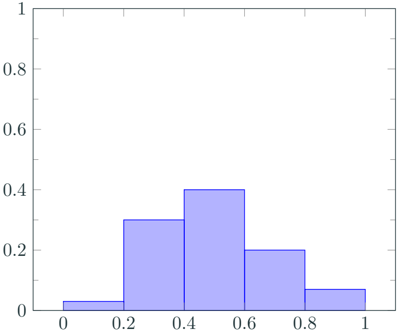


Figure 1: One histogram per feature

Histogram types

Soft vs hard binning

Histograms with **soft** binning where each value can contribute to more than one bin as opposed to **hard binning** where each value contributes with 1 to only one bin

Fixed vs variable bins

Histogram with **fixed bins** where each bin has a fixed centre as opposed to **variable bins**, where the bins centres and widths are learned in the construction process

Histogram as CNN layers

Soft binning version[3]

$$\phi_{k,b}(f_{k,i}) = \max\left(0, 1 - \frac{1}{w_{k,b}} \times |f_{k,i} - \mu_{k,b}|\right) \quad (1)$$

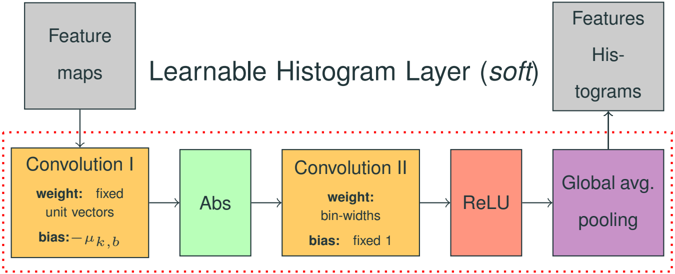


Figure 2: *soft*: Learnable histogram layer with soft binning and variable bin centers and widths

[3] Zhe Wang et al. *Learnable Histogram: Statistical Context Features for Deep Neural Networks*. 2018. arXiv: `1804.09398 [cs.CV]`.

Histogram as CNN layers

Hard binning version[4]

$$\phi_{k,b}(f_{k,i}) = \begin{cases} 0, & \text{if } 1.01^{w_{k,b} - |f_{k,i} - \mu_{k,b}|} \leq 1 \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

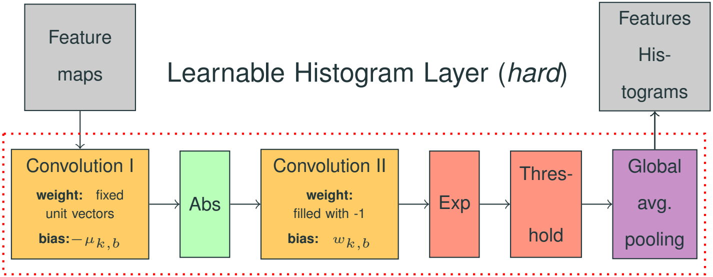


Figure 3: *hard:* Learnable histogram layer with hard binning and variable bin centers and widths

[4] Ibrahim Yusuf, George Igwegbe, and Oluwafemi Azeez. *Differentiable Histogram with Hard-Binning*. 2020. arXiv: 2012.06311 [cs.LG].

Sample generation

- HistNet needs **samples** to train
- Samples can be part of the training set (E.g. the plankton problem) or artificially generated
- When artificially generated, we use a **uniform** distribution of the prevalences
- In each **training epoch**, **500 samples** are generated randomly

HistNet is capable of using **labels** (if available), adding an extra connection in the network

Loss function

In this case the loss function to optimize is a combination of losses

$$L1Loss = \frac{\sum_i^C |\hat{p_i} - p_i|}{C} \tag{3}$$

$$CELoss = -\sum_i^C y_i \log s_i \tag{4}$$

where y_i and s_i are the groundtruth and the HistNet score for each class.

$$HistNetCLoss = L1Loss + CELoss \tag{5}$$

Experiments: Evaluating the performance of HistNet

Objective

- **Compare** HistNet with the other NN for quantification and also with traditional quantification algorithms
- **QuaNet** is implemented in the new quantification library **QuaPy**[5]
- For **DQN**, even though source code is provided[6], we were not able to reproduce the results in reasonable time

[5] Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. *QuaPy: A Python-Based Framework for Quantification*. 2021. arXiv: `2106.11057 [cs.LG]`.

[6] https://github.com/cml-cs-iastate/CyText

DATASETS

Binary datasets

- Text quantification $\rightarrow$ IMDB Dataset
- Image quantification $\rightarrow$ Fashion-MNIST reduced to two classes

Multiclass datasets

- Image quantification $\rightarrow$ Full Fashion-MNIST dataset

Experimental setup

Hyperparameters

- AdamW optimizer with early stopping over a validation dataset (20% of training data)
- Dropout of 0.3 in the quantification module to avoid overfitting
- Batch size of 16 samples (each epoch contains 500 samples)

Testing

- 2000 testing samples with prevalences between 0.01 and 0.99
- All experiments are **fully reproducible** [7]

[7] https://github.com/pglez82/histnet

Results IMDB Dataset

Method	AE	RAE	KLD
CC	0.07821	0.53489	0.04038
PCC	0.11627	0.75876	0.07737
AC	0.04686	0.31471	0.01728
PAC	0.04436	0.29587	0.01608
HDy	0.02907	0.17497	0.00719
QuaNet	0.02485	0.10224	0.00702
DQN-MAX	0.13777	0.83748	0.10068
HistNet	0.03905	0.22669	0.01140
HistNetC	**0.02034**	**0.09332**	**0.00346**

Table 1: Results over the IMDB dataset using the *softrbf* histogram for HistNet and HistNetC

Results Fashion-MNIST (Binary version)

Method	AE	RAE	KLD
CC	0.08250	0.37291	0.04354
PCC	0.09775	0.49963	0.05742
AC	0.04512	0.19669	0.01744
PAC	0.04079	0.17997	0.01443
HDy	0.04250	0.14525	0.02071
QuaNet	0.04077	0.13565	0.01237
DQN-MAX	0.07301	0.22351	0.04791
HistNet	0.04035	0.26046	0.01392
HistNetC	**0.03267**	**0.13243**	**0.00804**

Table 2: Results over the Fashion-MNIST dataset using the *softrbf* histogram for HistNet and HistNetC

Results Histogram Comparison

Histogram	IMDB		Fashion-MNIST	
	HistNet	HistNetC	HistNet	HistNetC
soft	0.03233	0.01908	0.03936	0.03292
softrbf	0.03905	0.02034	0.04035	0.03267
hard	0.06503	0.02761	0.04000	**0.03016**
sigmoid	0.03405	**0.01741**	0.23850	0.03256

Table 3: Comparison between different types of differentiable histograms.

Results Fashion-MNIST Multiclass

Method	AE
CC	0.02606
PCC	0.02602
AC	0.00936
PAC	0.00934
HDy	0.00933
HistNet	0.00687
HistNetC	**0.00614**

Table 4: Results over the complete Fashion-MNIST dataset using the *softrbf* histogram for HistNet and HistNetC

CONCLUSIONS

Some advantages

- Competitive method for both binary and multiclass quantification
- Works with and without labels
- Does not rely in a classifier
- Can handle different sample sizes
- Can be applied to different problems, changing the feature extraction layer

Also disadvantages

- Risk of overfitting
- Difficult to tune hyperparameters
- Complexity

Future work

What's next?

- Testing with **more datasets**
- Investigate how the different modules of the network work separated $\rightarrow$ **understand better the network architecture**
- Further investigate the **differences** between the different **histogram types**
- Can we use other types of **representations**? E.g. quantiles.