Supplementary material *for* Detection of out-of-distribution samples using binary neuron activation patterns

Bartłomiej Olber[1,2], Krystian Radlak[1,2], Adam Popowicz[2],
Michal Szczepankiewicz[3], Krystian Chachuła[1]
[1]Warsaw University of Technology [2]Silesian University of Technology [3]NVIDIA
{bartlomiej.olber.stud, krystian.radlak, krystian.chachula}@pw.edu.pl
adam.popowicz@polsl.pl msz@nvidia.com

1. Introduction

This document is a supplementary material that presents:

- theoretical explanation why binary patterns can be used as efficient feature in out-of-distribution (OOD) samples detection problem
- some additional experiments which were summarized in the main manuscript.

2. Theoretical explanation

In most of state-of-the-art networks, the chosen activation function is the ReLU [2]:

$$R(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ z & \text{otherwise.} \end{cases} \tag{1}$$

It is noteworthy to mention that

$$\frac{\partial R(z)}{\partial z} = \begin{cases} 0 & \text{if } z \leq 0 \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Let assume that we are considering a simple fully connected network, where o_n^l is the n^{th} component of the output of layer l, a_n^l is the output of neuron n of layer l, x_i is the input sample and w_{nm}^l is the weight vector of the n^{th} neuron of layer l defined as

$$o_n^l = a_n^l = \begin{cases} R(o^{l-1} * w_n^l) & \text{if } l > 1 \\ R(x * w_n^l) & \text{if } l = 1. \end{cases} \tag{3}$$

Let also assume that ${w_{mn}^l}^t$ is m^{th} weight of n^th neuron of layer l at iteration t then for fully connected network with ReLU function we can prove Theorem 1 [1].

Theorem 1 *In every layer l, for each neuron n, the gradients $\frac{\partial E^i}{\partial w_{nm}^i}$ will be zero if the neuron is not activated.*

Proof: The primary method used to adjust a network's parameters is known as back-propagation, which relies on stochastic gradient descent. This involves defining an error function, denoted by $E(X, \theta^t)$, where X refers to a batch of training data pairs x^i, y^i and y^i represents the expected output for sample x^i. The parameter θ^t denotes the network's parameters at iteration t, and the objective of stochastic gradient descent is to optimize the θ^t that minimize the error function $E(X, \theta)$. During each iteration, the weights are adjusted as:

$${w_{mn}^l}^{t+1} = {w_{mn}^l}^t - \alpha \frac{\partial E^i}{\partial w_{nm}^i} \tag{4}$$

It can be noticed that $E(X, \theta^t)$ can be formulated as the average of the error E^i of each sample x^i

$$E^i = -\sum_{k=1}^{c} y_k^i \ln(\hat{y}_k^i) \tag{5}$$

then

$${w_{nm}^l}^{t+1} = {w_{nm}^l}^t - \alpha \frac{1}{N} \sum_{i=1}^{N} \frac{\partial E^i}{\partial w_{nm}^l} \tag{6}$$

Then we can rewrite the gradient as

$$\frac{\partial E^i}{\partial w_{nm}^l} = \frac{\partial E^i}{\partial a_n^l} \frac{\partial a_n^l}{\partial w_{nm}^l}, \tag{7}$$

For $l > 0$ we have

$$\frac{\partial a_n^l}{\partial w_{nm}^l} = R'(\sum_{j=0}^{N} o_j^{l-1} w_{nj}^l) o_m^{l-1} =$$

$$= R'(z_n^l) o_m^{l-1} = \begin{cases} o_m^{l-1} & \text{if neuron is activated } (z_n^l > 0) \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

To simplify notation let assume that

$$\gamma_n^l = \frac{\partial E^i}{\partial a_n^l} \tag{9}$$

then

$$\frac{\partial E^i}{\partial a_n^l} = \sum_{j=1}^{N} \frac{\partial E^i}{\partial a_j^{l+1}} \frac{\partial a_j^{l+1}}{\partial a_n^l} = \sum_{j=1}^{N} \gamma_j^{l+1} \frac{\partial a_j^{l+1}}{\partial a_n^l}. \quad (10)$$

Since

$$a_j^{l+1} = R(\sum_{k=0}^{c} w_{jk}^{l+1} a_k^l) \quad (11)$$

then we obtain

$$\frac{\partial a_j^{l+1}}{\partial a_n^l} = R'(z_j^{l+1}) w_j^{l+1} \quad (12)$$

what leads to

$$\frac{\partial E^i}{\partial w_{nm}^i} = R'(z_n^l) \gamma_n^l w_{jn}^{l+1}. \quad (13)$$

3. Parameters evaluation

This section presents some results of experiments which were performed to tune the parameters of the proposed NAP method. In our experiments, we identified that there is no single optimal layer or configuration of the parameters that will give the best results for all databases. In order to establish the most optimal configuration of our method, we had conducted several experiments aimed at selecting the single most promising layer and its configuration. The summary of the optimized parameters in presented in Tab. 1.

Table 1. Summary of the parameters that need to be tuned in the proposed NAP based OOD detector

Paramemer	Description
p	Percentile threshold for binarization in a convolutional layer
t	Type of pooling
τ	Hamming distance threshold to decide if a sample is in or out-of-distribution
k	number of layer in network

Fig. 1 depicts exemplary OOD performance results of various NAP configurations for VGG-16 network [3] trained on FashionMNIST [4] dataset and evaluated on other OOD datasets. Subsequent bars correspond to subsequent activation layers in the network and parameters configurations.

Each pair of blue and orange bars refers to a single combination of the NAP configuration parameters presented in Tab. 1. Blue bars correspond to averaged accuracy of separation FashionMNIST against all other validation OOD datasets. Orange bars correspond to averaged accuracy on test OOD datasets. While separating training against test distributions, we use uncertainty threshold set on validation distribution according to the OOD evaluation protocol described in the main manuscript. As can be observed, activation patterns approach is capable of achieving very high OOD detection accuracy but the performance varies strongly across the selected layers and configurations.

Noticeably in Fig. 1 validation accuracy is correlated with accuracy on test dataset. It is a worthwhile conclusion which we have successfully incorporated in our auto-configuration strategy by choosing validation accuracy as a main optimization criterion.

We delved further into single-layer activation patterns, analyzing distributions of Hamming distances generated for samples that come from known and unknown sets of images. In Fig.2 (for the same network setup as in Fig. 1) we present the impact of the selected configuration on the achieved distribution separability. Setting the value of binarization percentile p either to 30 or 90 leads to mutually exclusive improvement of OOD detection accuracy on two OOD datasets - MNIST and NormalNoise. This phenomenon is one of the main reasons why selection of parameters in our method demands so much attention. For the most of possible pairs of D_s, D_v datasets considered in this work, exists such a combination of layer index, pooling and binarization parameters that allow to separate very accurately samples from these datasets by setting a Hamming distance threshold. Fig. 3 shows examples of these combinations for various dataset pairs. Acknowledging that selection of the most optimal NAP configurations is not straightforward, we designed the auto-configuration scheme based on exhaustive grid-search.

References

[1] Victor Demuysere. Investigating layer activation patterns in neural networks for classification error detection. Master's thesis, 2018. 1

[2] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. of the 27th Int. Conf. on on Machine Learning*, ICML'10, page 807–814, 2010. 1

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 2

[4] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 2

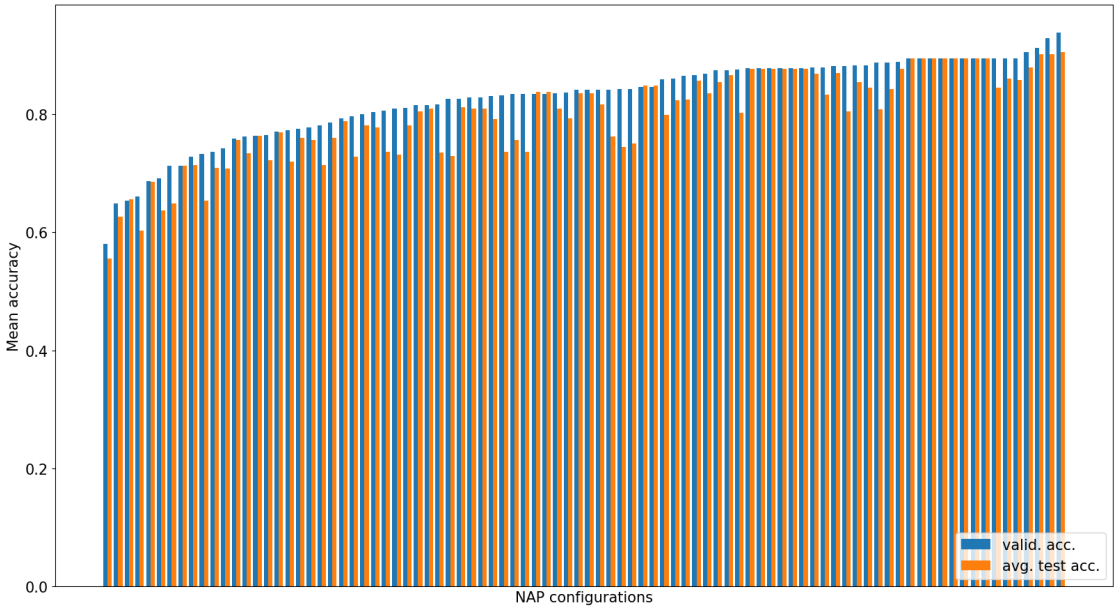
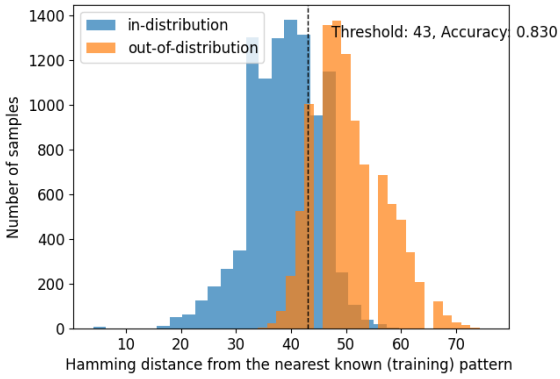
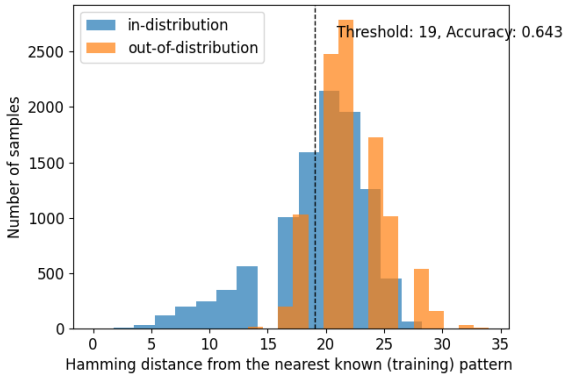


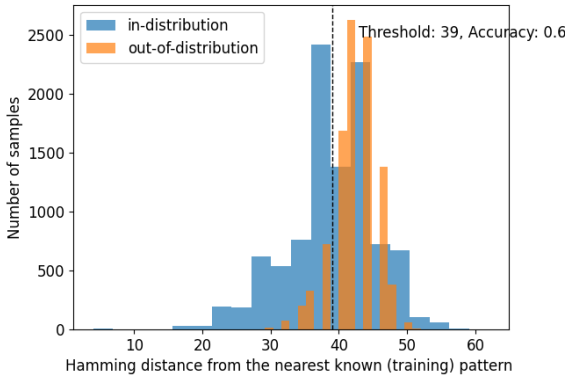
Figure 1. Accuracy of all considered single-layer NAP configurations averaged over all D_v for VGG trained on FashionMNIST.



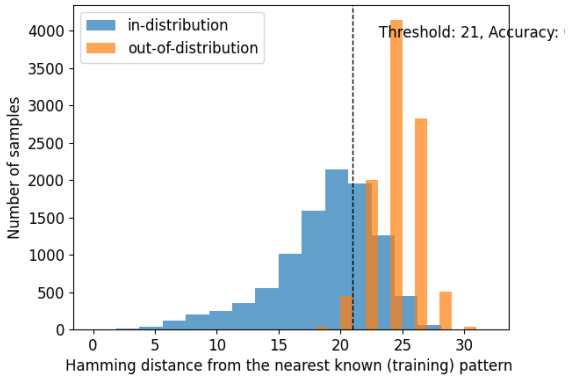
(a) D_v - MNIST, p - 30 (accuracy - 83%).



(b) D_v - MNIST, p - 90 (accuracy - 64%).

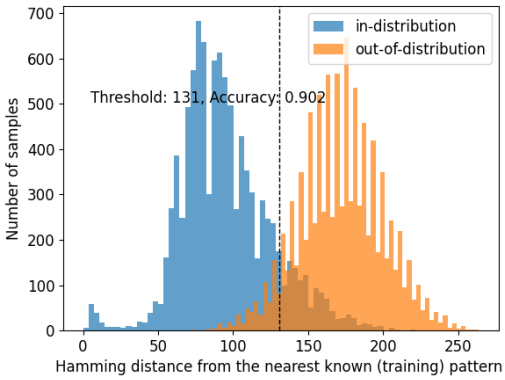


(c) D_v - NormalNoise, p - 30 (accuracy - 67%).

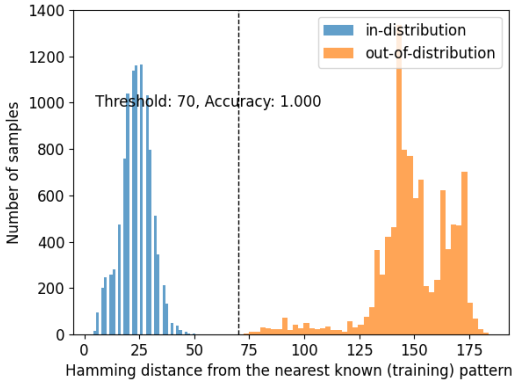


(d) D_v - NormalNoise, p - 90 (accuracy - 79%).

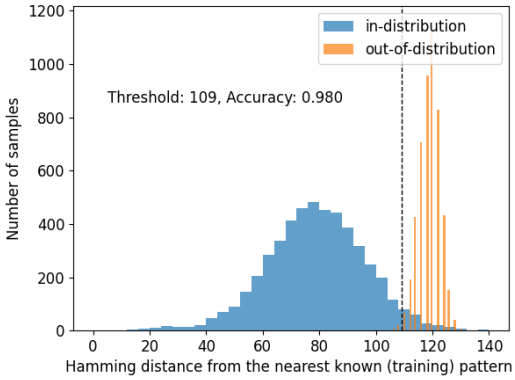
Figure 2. Hamming distance distributions histograms of selected (5th) layer for VGG trained on FashionMNIST.



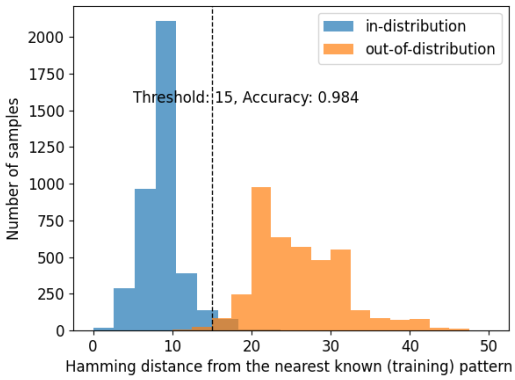
(a) Model - Resnet D_s - CIFAR10, D_v - FashionMNIST, p - 90, pool type - avg, layer id. - 1 (accuracy - 90%).



(b) Model - Resnet D_s - MNIST, D_v - CIFAR10, p - 30, pool type - avg, , layer id. - 9 (accuracy - 100%).



(c) Model - Resnet D_s - TinyImagenet, D_v - NormalNoise, p - 50, pool type - avg, layer id. - 10 (accuracy - 98%).



(d) Model - VGG D_s - STL10, D_v - CIFAR100, p - 30, pool type - max, layer id. - 12 (accuracy - 98%).

Figure 3. Selected Hamming distance distributions histograms - examples of high separability.