Unsupervised post-tuning of deep neural networks

Christophe Cerisara, Paul Caillon, Guillaume Le Berre

Unsupervised Post-Tuning of Deep Neural Networks

Christophe Cerisara
Université de Lorraine, CNRS, LORIA
F-54000 Nancy, France
cerisara@loria.fr

Paul Caillon
Université de Lorraine, CNRS, LORIA
F-54000 Nancy, France
paul.caillon@loria.fr

Guillaume Le Berre
Université de Lorraine, CNRS, LORIA
F-54000 Nancy, France
guillaume.le-berre@loria.fr

***Abstract*—We propose in this work a new unsupervised training procedure that is most effective when it is applied after supervised training and fine-tuning of deep neural network classifiers. While standard regularization techniques combat overfitting by means that are unrelated to the target classification loss, such as by minimizing the L2 norm or by adding noise either in the data, model or process, the proposed unsupervised training loss reduces overfitting by optimizing the true classifier risk. The proposed approach is evaluated on several tasks of increasing difficulty and varying conditions: unsupervised training, post-tuning and anomaly detection. It is also tested both on simple neural networks, such as small multi-layer perceptron, and complex Natural Language Processing models, e.g., pretrained BERT embeddings. Experimental results confirm the theory and show that the proposed approach gives the best results in post-tuning conditions, i.e., when applied after supervised training and fine-tuning.**

***Index Terms*—deep learning, unsupervised training, regularization, natural language processing**

I. Introduction

A major challenge for deep learning classifiers is to move beyond traditional supervised training and exploit the large quantity of unlabeled data available. A particularly successful approach in this direction consists in training models with auxiliary tasks for which labels are easily available, such as re-generating the observations. This may be realized for instance by plugging the classifier into an autoencoder architecture [1], or by predicting masked tokens or future events in time series, e.g., with language models [2]. Then, the parameters of these models may be transferred to target tasks with limited amount of annotated data, but such a fine-tuning process is likely to overfit [3]. Most of these methods still exploit supervised training losses, because both the observations in autoencoders and the masked words in language models are given at training time. Unsupervised losses that do not depend on the class labels are rather often used in different contexts, for instance to cluster [4] the data or to pretrain a classification model [5] in order to facilitate the subsequent supervised training process. We propose in this work an unsupervised loss that is designed to optimize the classifier risk without relying on any labeled training instance. This is made possible by replacing the standard empirical approximation of the risk, which approximates the full data distribution by a limited labeled training set, by another type of approximation that does not require any label. However, this approximation relies on an assumption about the shape of the distribution of the model outputs that is not always fulfilled in practice. We show in this work that carefully initializing the parameters of the model enables to fulfill this necessary assumption and leads to an analytical derivation of the unsupervised loss that can be applied to deep neural networks. Concretely, such an initialization of the parameters may be achieved by first training or fine-tuning the model on the small amount of available labeled data. Then, the unsupervised loss may be applied in a post-training way, i.e., as soon as the model's parameters are in a neighborhood of the optimum. This additional final training stage then improves the generalization capability of the classifier without impacting its performances on the target supervised task. Indeed, the proposed unsupervised risk gives theoretical guarantees that the model converges towards the same optimum than with the supervised classification risk. This is in contrast with most standard regularization methods, such as the L2 norm, dropout and priors: all of these methods push the model's parameters in a direction that improves generalization but that is not related to the target task-dependent classification risk.

We present in Section III the proposed unsupervised training method and evaluate its performances in Section IV. Section V discusses the experimental results, while Section II reviews the related work and Section VI concludes the paper.

II. Related works

Regularization is a key component of most deep neural networks, because of their large number of paremeters and the ease with which they might overfit [19]. The standard stochastic gradient descent has an *implicit regularization* property, which is still not well understood theoretically [20], but is essential to the success of deep learning. In addition to this implicit regularization, many regularization methods have been proposed for deep neural networks [21] but, to the best of our knowledge, only a few published methods propose to modify the parameters after the supervised training phase. One of such methods generate new latent data representations that are hard to classify to improve generalization [22], but although this data augmentation procedure occurs after the main training step, it is still supervised while our loss is unsupervised. Another classical post-training technique in deep learning is quantization, which aims at compressing the model layers with smaller bit widths [23]. Although such methods may be viewed as post-training regularizers, because drastic compres-

This work was performed using HPC resources from GENCI-IDRIS (Grant 2020-AD011011668).

sion introduces noise in the model, such a regularization is not based on any approximation of the classifier risk. The authors of [24] propose another post-training strategy for deep neural networks, for training the final layer alone while freezing the rest of the network. They report improvements in generalization and explain them from the kernel theory point of view, but contrary to our proposal, they still use the supervised objective for this post-training step.

Learning with Label Proportions is a different weakly supervised paradigm where the only supervision comes from the known proportion of classes in subsets, or bags, of observed samples [25]. This situation occurs in many important applications, for instance to train a neural network on quantum physics experiments, where assigning a precise label to a given instance is not possible [26]. [27] further study the minimum amount of unlabeled sets with different label proportions required to train any binary classifier, and prove that three such sets are sufficient. In our approach, we rather use a single such subset and exploit two additional assumptions to train our deep neural networks. More generic weakly supervised learning methods may also exploit uncertainty on labels, such as [28] or class-conditional label noise, which considers training a classifier with labels that have been corrupted with different probability [29]. Another approach that is related to our work are one-class deep neural networks, in particular the Deep-SVDD [30] and CVDD [16] models. We compare our method with CVDD in Section IV. Although such one-class models and our post-processing approach differ in their theoretical motivation, both optimization algorithms end up being very similar. For instance both approaches rely on a first step to split the classifier scores into two subsets according to a given quantile, followed by a second optimization step. The main difference lies in the loss function that is used during this optimization step. Text anomaly detection methods may also exploit generative models, for instance, [31] adapts generative adversarial networks, which have already shown remarkable results for image anomalies, to generate normal sentences and then compute an anomaly score for texts. Non-deep learning methods are also used for text anomaly detection, for instance, [32] rely on the low-rank matrix factorization of the document-term matrices.

III. Proposed unsupervised approach

A. Motivation

Let us consider a binary classifier with parameters θ that outputs a scalar score $f(x) \in \mathbb{R}$ for input x. Let us call $y \in \{0, 1\}$ the true class of x. The classification decision is class $\hat{y} = 0$ iff $f(x) \leq 0$, and $\hat{y} = 1$ iff $f(x) > 0$. Supervised training of $f(x)$ classically aims at minimizing the classifier risk $\theta^* = \arg\min R(\theta)$, which is the expected error of the classifier over the full (unknown) distribution of the data. With a hinge loss, it is:

$$\begin{aligned} R(\theta) &= E_{p(x,y)}\left[(1 - f(x) \cdot (2y-1))_+\right] \quad (1) \\ &= P(y=0) \int p(f(x)=\alpha|y=0)(1+\alpha)_+ d\alpha + \\ &\quad P(y=1) \int p(f(x)=\alpha|y=1)(1-\alpha)_+ d\alpha \end{aligned}$$

where $(x)_+ = \max(0, x)$. The distribution $p(x, y)$ is usually unknown, and $R(\theta)$ is commonly approximated by the empirical risk $\hat{R}_{emp}(\theta)$, which averages the error over a finite dataset, given the supervised labels y. [6] propose another approximation $\hat{R}(\theta)$ of $R(\theta)$ that does not require the knowledge of y, hence leading to an unsupervised training algorithm, which depends on 2 assumptions:

- **H1**: The class-conditional distribution of the scores $p(f(x)|y)$ is Gaussian;
- **H2**: The class-marginal prior $P(y)$ is known;

They thus propose an unsupervised training method for a binary linear classifier, and for three types of error functions: hinge, exponential and log loss. We propose in this work to extend the domain of application of their approximation of the risk to a much larger and powerful class of binary classifiers: any deep neural network (DNN) with a final linear (or softmax) classification layer. To achieve this, we introduce a new simplifying assumption, derive an end-to-end differentiable equation of the risk, analyze the resulting function and propose a new post-tuning regularization strategy.

B. Limitations and proposed solutions

While **H2** is an assumption that is met in several common tasks, e.g., the prevalence of a disease is known, as well as the ratio of occurrence of a word in a language, **H1** is critical for the validity of the previous approximation of the risk. [6] justifies **H1** with a central limit theorem for non-iid features: intuitively, a linear classifier computes a sum of random variables that tends towards a Gaussian distribution as the feature vector dimension tends towards infinity. When replacing the linear classifier with a DNN, this theorem may not be used any more. Furthermore, even with a linear classifier, we have observed that **H1** does not hold in several practical situations, for instance when initializing the classifier with random parameters. Even after training has started, the fact that the feature vector dimension is small may break the theorem. Therefore, we rather propose a third assumption:

- **H3:** Both distributions $p(f(x)|y=0)$ and $p(f(x)|y=1)$ are well separated when θ is close to an optimum $\theta^*_{emp} = \arg\min_\theta \hat{R}_{emp}(\theta)$

Our rationale is to not assume any more that **H1** is always true, but rather to restrict ourselves to contexts where **H1** is valid: **H3** serves to define such contexts, and **H3** further enables us to derive an end-to-end closed-form solution of the unsupervised loss. Note that, in a neighborhood of θ^*_{emp} and when the DNN has enough capacity, the Universal Approximation Theorem [7] guarantees that the scores at the output of the DNN of the samples that can be correctly classified are well separated. The samples that can not be correctly classified, i.e., the ones for which the input features do not provide enough information to decide on one class or another, are arbitrarily assigned to the two partitions. In practical situations, the input features are rich enough and this Bayes Risk is small compared to both distributions of well-classified samples. Then, the scores in each partition may be approximated with a Gaussian, which

fulfills **H1**. We have done many experiments that confirm this, and report a few next.

The obvious question is why would we need to optimize the model with our unsupervised loss, when the model is already trained ? Because θ^*_{emp} is prone to overfitting, and compensating this issue with excessive regularization during the supervised training stage might move away θ^*_{emp} from θ^*. Indeed, conversely to standard regularization, our approximation of Eq- 1 is designed to converge towards θ^*. Furthermore, it is more robust to overfitting than empirical risk minimization, because only 4 parameters (the two Gaussian parameters) are estimated from the data, compared to the large number of parameters classically found in recent DNNs. Our proposed approach may thus be viewed as a post-tuning stage that is applied after fine-tuning the DNN to the target task and that will a posteriori reduce the impact of overfitting.

C. Derivation of the loss

We propose to rewrite Eq- 1 as follows:

$$R(\mu,\sigma) = P(y=0)\int N(\alpha;\mu_0,\sigma_0)(1+\alpha)_+ d\alpha + P(y=1)\int N(\alpha;\mu_1,\sigma_1)(1-\alpha)_+ d\alpha$$

where $\mu = (\mu_0,\mu_1) \in \mathbb{R}^2$ and $\sigma^2 = (\sigma_0^2,\sigma_1^2) \in \mathbb{R}^{+2}$ are respectively the means and variances of the corresponding Gaussians $p(f(x)|y=0;\theta)$ and $p(f(x)|y=1;\theta)$, and

$$N(\alpha;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\alpha-\mu)^2}{2\sigma^2}}$$

Removing the non-linearity, we get:

$$\begin{aligned} R(\mu,\sigma) &= P(y=0)\int_{-1}^{+\infty} N(\alpha;\mu_0,\sigma_0)d\alpha + \\ & P(y=0)\int_{-1}^{+\infty} \alpha N(\alpha;\mu_0,\sigma_0)d\alpha + \\ & P(y=1)\int_{-\infty}^{1} N(\alpha;\mu_1,\sigma_1)d\alpha - \\ & P(y=1)\int_{-\infty}^{1} \alpha N(\alpha;\mu_1,\sigma_1)d\alpha \end{aligned}$$

We know that:

$$\int_a^b N(x;\mu,\sigma)dx = \frac{1}{2}\left(\text{erf}\left(\frac{b-\mu}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{a-\mu}{\sigma\sqrt{2}}\right)\right)$$

and:

$$\int_a^b xN(x;\mu,\sigma)dx = \mu\int_a^b N(x;\mu,\sigma)dx - \sigma^2\left[N(x;\mu,\sigma)\right]_a^b$$

So,

$$\begin{aligned} \int_a^b xN(x;\mu,\sigma)dx &= \tfrac{\mu}{2}\left(\text{erf}\left(\tfrac{b-\mu}{\sigma\sqrt{2}}\right) - \text{erf}\left(\tfrac{a-\mu}{\sigma\sqrt{2}}\right)\right) - \\ & \sigma^2\left(N(b;\mu,\sigma) - N(a;\mu,\sigma)\right) \end{aligned}$$

Which gives:

$$\begin{aligned} R(\mu,\sigma) &= \frac{P(y=0)}{2}(1+\mu_0)\left(1-\text{erf}\left(\frac{-1-\mu_0}{\sigma_0\sqrt{2}}\right)\right) + \\ & P(y=0)\sigma_0^2 N(-1;\mu_0,\sigma_0) + \\ & \frac{P(y=1)}{2}(1-\mu_1)\left(1+\text{erf}\left(\frac{1-\mu_1}{\sigma_1\sqrt{2}}\right)\right) + \\ & P(y=1)\sigma_1^2 N(1;\mu_1,\sigma_1) \end{aligned} \quad (2)$$

In order to use Eq- 2 as a loss function for Stochastic Gradient Descent in deep learning toolkits, we need to rewrite this expression as a differentiable function that depends on the parameters θ. We analyze next the function $R(\mu,\sigma)$ and finally derive the missing analytical expression to build the end-to-end loss.

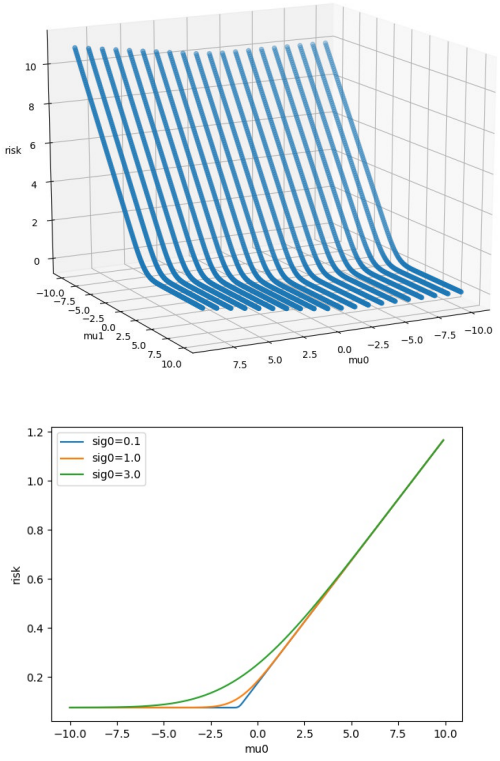

Fig. 1. Risk as a function of both (μ_0,μ_1) (top), and only μ_0 (bottom) for $\mu_1 = 2$, $\sigma_1 = 1$ and $\sigma_0 \in \{0.1, 1, 3\}$

D. Analysis of the loss

Let us note $p_0 = P(Y=0)$ and plot Equation 2 as a function of (μ_0,μ_1) in Figure 1 (top), for $p_0 = 0.1$ and $\sigma_0 = \sigma_1 = 1$. When we fix μ_1, we can see in Figure 1 (bottom) that the risk as a function of μ_0 may be well approximated by a scaled and translated rectified linear function, as long as the variances are small enough. Furthermore, the lower σ_0 (and σ_1) is, the better the risk is. Varying μ_1 and σ_1 only translates this curve vertically, above the horizontal axis. So, assuming that the risk has first been minimized with respect to μ_1, then the global minimum of the risk may be obtained by decreasing

linearly μ_0. Conversely, lower risks are obtained when μ_1 is increasing. In conclusion, the risk is lower when μ_0 is small, μ_1 is large and when σ_0 and σ_1 are small, which supports the validity of **H3** on reasonably good areas of the parameter space, i.e., when θ^* and θ^*_{emp} are not too far away one from the other.

E. Approximation of the bi-Gaussian distribution

Given the previous analysis, we may now assume **H3** in a neighborhood of θ^*_{emp}: both modes (μ_0, σ_0) and (μ_1, σ_1) of the scores distribution are well separated with a small overlap. Then, a good approximation of μ_0 and μ_1 can be computed by splitting all the scores $f(x)$ according to the p_0-quantile x_{p_0}. Let us call X^- the subset of size N^- of all data points that are on the left side of the p_0-quantile:

$$X^- = \{x \in X \quad s.t. \quad f(x) < f(x_{p_0})\}$$

and similarly for the other side:

$$X^+ = \{x \in X \quad s.t. \quad f(x) \geq f(x_{p_0})\}$$

We can now approximate the Gaussian parameters deterministically:

$$\mu_0 \simeq \frac{1}{N^-} \sum_{x \in X^-} f(x) \qquad \mu_1 \simeq \frac{1}{N^+} \sum_{x \in X^+} f(x) \tag{3}$$

$$\sigma_0^2 \simeq \left(\frac{1}{N^-} \sum_{x \in X^-} f(x)^2 \right) - \left(\frac{1}{N^-} \sum_{x \in X^-} f(x) \right)^2$$

$$\sigma_1^2 \simeq \left(\frac{1}{N^+} \sum_{x \in X^+} f(x)^2 \right) - \left(\frac{1}{N^+} \sum_{x \in X^+} f(x) \right)^2$$

We can now plug equations 3 into Eq 2 to get a differentiable loss $\hat{R}(\theta)$ with respect to the network parameters θ.

Algorithm 1 summarizes the optimization procedure.

IV. Experimental validation

The proposed unsupervised loss is coded in PyTorch [8] and is freely distributed [1].

A. Unsupervised conditions

We first evaluate our proposed optimization algorithm in unsupervised conditions, i.e., without any label in the training corpus, and thus without any first stage of supervised training or fine-tuning of the model: the models' parameters are initialized randomly before applying algorithm 1. These experiments will help us to evaluate, on various tasks, the impact of not fulfilling assumption **H3**.

1) Breast Cancer detection: The first task is to detect cancer on the standard Wisconsins Breast Cancer benchmark [9] with a 2-layers Multi-Layer Perceptron (MLP). We compare it with k-means clustering in Table I.

This experiment suggests that, on a relatively easy dataset, the proposed optimization procedure may be used as an unsupervised training algorithm and outperforms a simple clustering baseline. We investigate next its unsupervised performances on more complex tasks and models.

[1] https://github.com/cerisara/unsuprisk.git

Algorithm 1 Unsupervised optimization

- Initialization:
  - Let consider a binary classification task, for which we assume that the proportion of class-0 elements p_0 is known approximately;
  - Let $\{x_i\}_{1 \leq i \leq N}$ be a corpus of observations without labels;
  - Let $g_\phi(x)$ be a deep neural network with parameters ϕ that computes a vectorial representation of an input x, which is fed to a final linear classification layer $f_\theta(g_\phi(x))$ with parameters θ; ϕ and θ should be pretrained.
- Iterate:
  - Run a forward pass on the dataset $\{x_i\}_{1 \leq i \leq N}$ with the current parameters ϕ, θ.
  - Compute all classifier scores $\{s_i = f_\theta(g_\phi(x_i))\}_{1 \leq i \leq n}$ over the full corpus N, or over a batch of observations n that is large enough to assume that the distribution of classes in the batch is representative of the distribution in the whole corpus.
  - Sort the list of scores $(s_i)_{1 \leq i \leq n}$ to compute the p_0-quantile x_{p_0}.
  - Compute the Gaussian parameters $\mu = (\mu_0, \mu_1), \sigma = (\sigma_0, \sigma_1)$ with Equations 3.
  - Compute the risk (Eq 2) with these Gaussian parameters.
  - Apply automatic differentiation to compute $\nabla_\theta R(\mu, \sigma)$, and optionally $\nabla_\phi R(\mu, \sigma)$;
  - Run a step of Gradient Descent to update θ, and optionally ϕ.

TABLE I
UNSUPERVISED EXPERIMENTS ON THE WISCONSINS BREAST CANCER BENCHMARK.

	2-layers MLP	K-means clustering
Accuracy	**91%**	85%

2) SentEval tasks: We evaluate our unsupervised training algorithm on four recent Natural Language Processing binary classification datasets:

- **Movie Review** (MR): classification of positive vs. negative movie reviews;
- **Product Review** (CR): classification of positive vs. negative product reviews;
- **Subjectivity status** (SUBJ): classification of subjective vs. objective movie reviews;
- **Opinion polarity** (MPQA): classification of positive vs. negative movie reviews.

These datasets as well as the experimental evaluation protocol that we have used are described in details in [10]. This protocol first computes a sentence representation with the **InferSent** [11] sentence embeddings, and then passes these sentence embeddings into a feed-forward network that is

trained on each dataset.

We have adopted the same experimental protocol and the same hyper-parameters, except that we do not train the final feed-forward network with supervised labels and the cross-entropy loss, but we rather train it without any label and with our proposed unsupervised loss. Note that in these experiments, the parameters of the InferSent embeddings are pretrained on general English texts, while the parameters of the final classification layer are initialized randomly, because we assume that we do not have access to any task-specific label. Table II summarizes the accuracy of two baselines: the state-of-the-art supervised models trained on the full corpus and on only 100 instances, as well as the accuracy of the proposed unsupervised model.

TABLE II
PURELY UNSUPERVISED TRAINING: ACCURACY ON 4 NLP TASKS (BOTTOM LINE). SUPERVISED MODEL TRAINED ON ALL DATA (TOP, FROM [11]), AND ON ONLY 100 TRAINING SAMPLES (MIDDLE).

System	CR	SUBJ	MPQA	MR
InferSent fully supervised	*86.3*	*92.4*	*90.2*	*81.1*
InferSent supervised 100 obs	63.8	62.5	70.1	53.9
Proposed unsupervised training	66.8	83.0	70.9	59.7

We can observe that the proposed unsupervised method performs much worse than the fully supervised models. We interpret this as a consequence of the fact that **H3** is not fulfilled. However, our unsupervised model always matches the performances of a supervised model trained on 100 reviews, and outperforms them (+20% accuracy) on the subjectivity classification task.

B. Post tuning conditions

We evaluate next our unsupervised loss in "post-tuning" conditions: the models are first trained in a supervised way to their optimum with standard regularization methods, and this first training phase is followed by our unsupervised optimization procedure. In order to initialize a model that is as close as possible to its supervised optimum, we exploit the scripts and hyperparameters from state-of-the-art published papers, and report their published results. Two binary classification NLP tasks are considered, plus another text anomaly detection task.

1) Supervised classification: The two classification tasks are MRPC, which goal is to decide whether two sentences are paraphrases, and CoLA, where the model decides whether a sentence is grammatical or not. 400 sentences are extracted from the training sets to find p_0 and early stopping threshold. The state-of-the-art accuracy is 93% on MRPC [12] and 75% on CoLA [13], which are obtained with advanced models, such as StructBERT and ensembling. We rather use the reference and ubiquitous BERT pretrained model, which is easier to experiment with and gives 84% on MRPC and 56% on CoLA when trained on the full training corpus [14]. We report in Table III the best accuracy (among all training epochs) obtained on the validation set with these baselines, when trained on our reduced training set. For our post-tuned model, we find the best epoch with early stopping on the 400 development sentences, and compute the accuracy on the validation set with this model.

TABLE III
POST TUNING EXPERIMENTS WITH A PRETRAINED BERT MODEL ON THE MRPC AND COLA BENCHMARKS.

	MRPC	
	Supervised fine-tuning	Proposed post-tuning
Accuracy	78.4	**82.3**
F1	85.7	**86.7**
	CoLA	
	Supervised fine-tuning	Proposed post-tuning
Matthew's corr.	47.1	**49.6**

The proposed loss improves the accuracy of the fine-tuned BERT model on both tasks.

2) Unsupervised anomaly detection: The post-tuning approach may also be applied on a model previously trained with another unsupervised loss function. We demonstrate this on the text anomaly detection task, where models are classically trained in an unsupervised way, because of the lack of labeled anomaly samples. Unsupervised text anomaly detection is performed on the Reuters-21578 corpus [15], with the model and experimental conditions described in [16]. The standard Area Under Curve (AUC) metric is used to evaluate the quality of the text anomaly detection task. The anomaly detection model transforms a variable-length sequence of Glove [17] pre-trained word embeddings into a fixed-size document embedding matrix M with multi-head self-attention [18]. Each of the r heads, or columns in the resulting matrix M, is a linear combination of the original word embeddings.

The anomaly score is then the average cosine distance between these r sentence embeddings and a set of r context vectors $\{c_k\}_{1\leq k\leq r}$. These context vectors are trained in [16] with the unsupervised CVDD loss, which minimizes the cosine distance on the (unlabeled) training corpus.

We propose to reformulate this last step within the framework of neural networks, by encoding each c_k as the parameters θ of an unbiased linear layer, which outputs the dot product between its weights θ and the input x, normalized by both θ and x Frobenius norms:

$$f_{cos}(x) = \frac{\sum_i \theta_i x_i}{(\sum_i \theta_i^2)\,(\sum_i x_i^2)} \tag{4}$$

This neural version of the CVDD model is illustrated in Figure 2.

In our experiments, we initialize the parameters of the model in Figure 2 with a standard CVDD training, as described in [16]. This first step enables to get parameters that are relatively close to a good solution, with well-separated Gaussians in the score space, hence fulfilling our third assumption. Then, in a second phase, we remove the CVDD loss and replace it with our unsupervised loss.

The version of the Reuters-21578 dataset distributed with the CVDD code is composed of news articles with a single label (topic) per article. Table IV presents the experimental

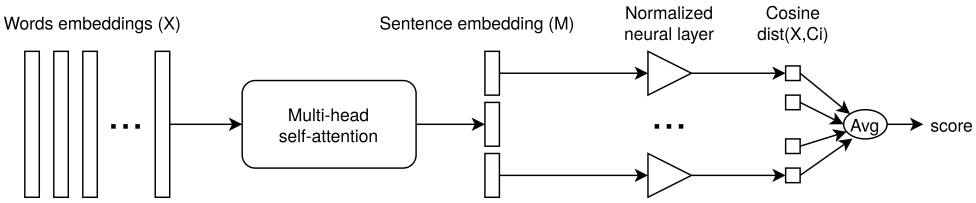

Fig. 2. Text anomaly detection model from [16], encoded in the form of a neural network. The word embeddings sequence is encoded into M fixed-size embeddings. Each of these M sentence embeddings is passed to a normalized linear layer that outputs the cosine distance between the sentence embedding and its parameter vector. These r normalized neural layers implement equation 4. The final anomaly score is the average of these distances.

results on this corpus: each of the seven labels listed in the first column is considered as the normal class one after the other, while all the other labels are representatives of the anomaly class. The italic numbers between parenthesis are taken from the paper [16], while the plain CVDD numbers are obtained by re-running the code distributed by the authors of [16]: there may be some differences, which are likely due to different runtime environments (type of GPUs, library versions...). The columns "Eq 2" correspond to our proposed unsupervised loss. The model and number of parameters are the same between CVDD and our approach: only the loss changes. Following the experimental conditions chosen in [16], the training corpus only contains normal samples.

TABLE IV
AUC ON ANOMALY DETECTION FOR TEXTS: "EQ 2" IS OUR PROPOSED UNSUPERVISED POST-TUNING METHOD.

Normal class	r=3 **CVDD**	r=3 **Eq 2**	r=5 **CVDD**	r=5 **Eq 2**	r=10 **CVDD**	r=10 **Eq 2**
earn	93.9 (*94.0*)	**96.1**	92.7 (*92.8*)	**95.1**	88.2 (*91.8*)	**96.5**
acq	90.1 (*90.2*)	**92.2**	88.6 (*88.7*)	**92.7**	91.5 (*91.5*)	**94.2**
crude	89.7 (*89.6*)	**92.2**	92.5 (*92.5*)	**96.2**	96.4 (*95.5*)	**98.1**
trade	97.9 (*98.3*)	**98.2**	98.1 (*98.2*)	**98.1**	**99.6** (*99.2*)	99.4
money-fx	81.9 (*82.5*)	**89.5**	78.0 (*76.7*)	**91.3**	**83.1** (*82.8*)	81.1
interest	92.4 (*92.3*)	**94.2**	92.1 (*91.7*)	**95.5**	97.2 (*97.7*)	**98.4**
ship	96.8 (*97.6*)	**98.8**	92.8 (*96.9*)	**98.7**	96.1 (*95.6*)	**97.0**

The experimental results with both the original CVDD loss and the model fine-tuned with our unsupervised loss are reported in Table IV. All models in this table have the same number of parameters. Fine-tuning is achieved with the Adam criterion over 1000 epochs with a learning rate of 10^{-4} and momentum of 10^{-6}. The only remaining hyperparameter is our p_0 value, which represents our expected proportion of anomaly instances. We optimize p_0 with a grid search between 0.1 and 0.5 and pick the value with the lowest risk on a development set. The development set is equal to the training set (with normal samples only) merged and shuffled with 50% of random outliers from the training set. There is no label in the development set.

We can observe a quasi-systematic improvement obtained when using our proposed unsupervised loss, as compared to the CVDD objective. The average relative reduction of the error rate obtained with our proposed loss is -32% with $r = 3$ and **-43%** with $r = 5$, which means that our proposed loss nearly halves the number of errors made by CVDD, and -22% with $r = 10$. A qualitative analysis of the detected anomalies is given in the next section.

V. DISCUSSION OF EXPERIMENTAL RESULTS

We now discuss our experimental results on topic outliers detection on the Reuters corpus, in the light of our modeling assumptions.

a) **H1** *and* **H3***:* The first assumption concerns the bi-gaussianity of the scores. To get an intuition about this assumption, we plot in Figure 3 the histogram of the scores $f(x)$ over all training samples just before (left) and after (right) unsupervised post-tuning. We can also see in Figure 3 the effect of post-tuning, which increases the interval between both modes from 0.14 to 0.17, as expected. This plot also illustrates a case that is compatible with our third assumption, i.e., that the initial distribution is already composed of two modes that are relatively well separated.

b) **H2***:* Regarding our second assumption about the known class marginal p_0, we have plotted in Figure 4 the test AUC for the full range of possible values for p_0.

Note that at training time, following the experimental conditions in [16], there is no outlier in the training corpus, so in theory the true p_0 should be null. However, in every realistic corpus, there are always samples that are not prototypical and somehow differ from the bulk of the distribution. Such samples may be considered as outliers, as illustrated in the following qualitative analysis.

Figure 4 shows that the influence of p_0 is significant, but with a moderate degradation within the whole range of possible values. p_0 should then preferably be tuned on a development corpus as we have done in our experiments, whenever it is possible.

c) Qualitative analysis: Because the test corpus is created by choosing a single topic as the normal class, and pooling together all other topics as outliers, the resulting corpus is very unbalanced. Figure 5 plots the distribution of the classifier scores on the test corpus, showing with a different color and opacity the normal and anomaly samples.

We observe the following:

- At the extreme left of the distribution, we find all "normal" class examples, e.g.:
  A 24-hour strike by Belgian public employees protesting against a government pay ...
- The large set of "non-normal" samples is distributed over a large span covering the mid-left, central and right of the

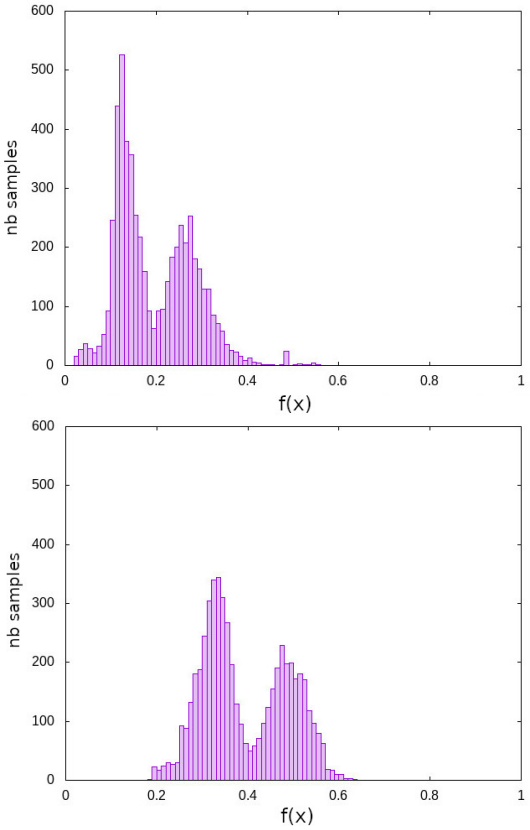

Fig. 3. Histograms of the classifier scores on the training corpus, before (top) and after (bottom) unsupervised training with our loss.

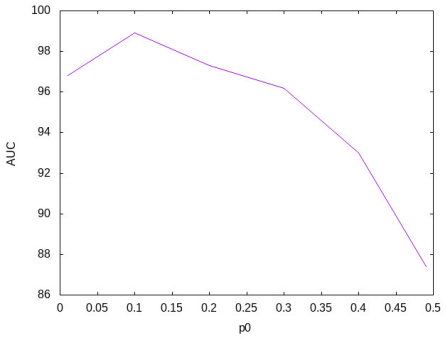

Fig. 4. Test AUC for varying p_0, with $r = 3$ and $nc =$"ship". Because of the symmetry of unsupervised binary classification, p_0 may only vary between 0 and 0.5

distribution. The model then splits this large span itself into several levels of outliers:

- At the extreme right, outliers composed of only one or two words, typically just months: *august*, *july*...
- At the center of the distribution, standard-length documents with one of the "non-normal" topics, e.g.: *Video Jukebox Network inc said it signed a letter of intent to purchase up to 3.5 mln shares of the four mln shares of the company's common stock...*

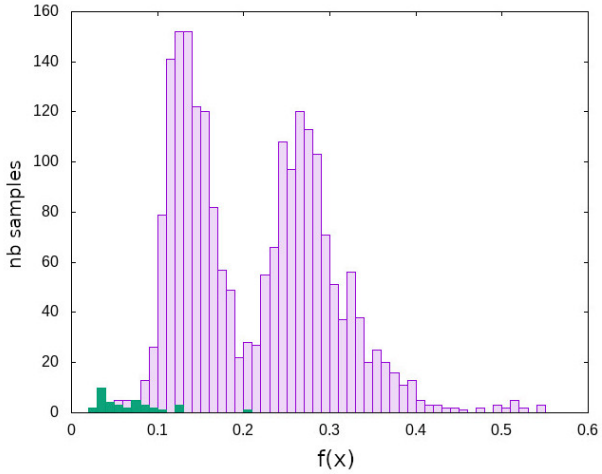

Fig. 5. Histogram of the scores of the normal (solid green on the bottom left) and anomaly (translucent pink dominating between $x = 0.1$ and $x = 0.6$) test samples.

VI. Discussion and conclusion

Supervised learning aims at minimizing the classifier error for a specific task, for instance predicting the sentiment from short texts or recognizing persons in pictures. The most commonly used function to approximate this objective is the empirical risk, which computes the classifier error on a given labeled training corpus. [6] proposes another approximation for linear classifiers, which provably converges towards the optimum using a training corpus without labels, i.e., without a precise and detailed information about the task itself. The success of the approach depends on an assumption about the Gaussianity of the marginal scores distribution, which is theoretically fulfilled when the observations have infinite dimension. Experiments confirm that it is also *often* fulfilled in practice, although regions of the parameter space might exist where this assumption is broken. During training, when the classifier enters such a region, convergence is not any more guaranteed. Therefore, we propose in this work a new constraint that positions the model in regions of the parameter space that are likely to fulfill the Gaussianity assumption, even with non-linear deep neural networks.

Just like other regularization methods, the proposed loss combats overfitting: this is mainly because it does not follow the common empirical risk approximation, which is the root cause of overfitting. Still, one could argue that a finite dataset, even unlabeled, is actually used and that the training process may overfit the distribution of the data in this dataset. This is correct, but the dataset is only used to estimate four parameters: the means and variances of two Gaussians, which strongly limit the risk of overfitting, as compared to the millions of parameters that are trained in the neural network in the preceding supervised training phase. So the proposed loss is extremely robust to overfitting: it will lead the model away from the overfitted local optima and towards the optimum of the classifier risk with much better generalization properties.

Hence, although it may seem counter-intuitive to apply unsupervised optimization after supervised training, because of the possibility to destroy information gained from the training labels, this final optimization step is designed to still converge towards the minimum task error rate.

We experimentally validate our proposed approach on tasks and models of increasing complexity: from two-layers MLP on the Wisconsis Breast Cancer dataset, to middle-sized neural networks for anomaly detection on texts and large BERT-based fine-tuned models. We also experiment with models that do not fulfill our proposed constraint on SentEval benchmarks, and models that do, such as CVDD. We show that using our proposed constraint leads to better results, giving a new state-of-the-art performance for text anomaly detection on the Reuters corpus.

REFERENCES

[1] M. Tschannen, O. Bachem, and M. Lucic, "Recent Advances in Autoencoder-Based Representation Learning," *arXiv:1812.05069 [cs, stat]*, Dec. 2018, arXiv: 1812.05069. [Online]. Available: http://arxiv.org/abs/1812.05069

[2] Q. Liu, M. J. Kusner, and P. Blunsom, "A Survey on Contextual Embeddings," *arXiv:2003.07278 [cs]*, Apr. 2020, arXiv: 2003.07278. [Online]. Available: http://arxiv.org/abs/2003.07278

[3] R. Wang, S. Si, G. Wang, L. Zhang, L. Carin, and R. Henao, "Integrating Task Specific Information into Pretrained Language Models for Low Resource Fine Tuning," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 3181–3186. [Online]. Available: https://www.aclweb.org/anthology/2020.findings-emnlp.285

[4] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture," *IEEE Access*, vol. 6, pp. 39 501–39 514, 2018, conference Name: IEEE Access.

[5] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?" in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.

[6] K. Balasubramanian, P. Donmez, and G. Lebanon, "Unsupervised supervised learning II: Margin-based classification without labels," *Journal of Machine Learning Research*, vol. 12, pp. 3119–3145, 2011.

[7] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/32cbf687880eb1674a07bf717761dd3a-Paper.pdf

[8] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Proc. of the NIPS Workshop Autodiff*, 2017.

[9] W. H. Wolberg, W. N. Street, and O. L. Mangasarian, "Breast cancer wisconsin (diagnostic) data set," *UCI Machine Learning Repository [http://archive. ics. uci. edu/ml/]*, 1992.

[10] A. Conneau and D. Kiela, "SentEval: An evaluation toolkit for universal sentence representations," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1269

[11] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 670–680. [Online]. Available: https://www.aclweb.org/anthology/D17-1070

[12] S. Ruder, "Nlp-progress," 2020. [Online]. Available: {http://nlpprogress.com/english/semantic_textual_similarity.html}

[13] A. Warstadt, A. Singh, and S. R. Bowman, "Neural network acceptability judgments," *CoRR*, vol. abs/1805.12471, 2018. [Online]. Available: http://arxiv.org/abs/1805.12471

[14] Huggingface, "Transformers results," 2020. [Online]. Available: https://huggingface.co/transformers/v2.3.0/examples.html

[15] C. Apté, F. Damerau, and S. M. Weiss, "Automated learning of decision rules for text categorization," *ACM Transactions on Information Systems (TOIS)*, vol. 12, no. 3, pp. 233–251, 1994.

[16] L. Ruff, Y. Zemlyanskiy, R. Vandermeulen, T. Schnake, and M. Kloft, "Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4061–4071. [Online]. Available: https://www.aclweb.org/anthology/P19-1398

[17] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://www.aclweb.org/anthology/D14-1162

[18] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=BJC_jUqxe

[19] B. Neyshabur, R. Tomioka, R. Salakhutdinov, and N. Srebro, "Geometry of Optimization and Implicit Regularization in Deep Learning," *arXiv:1705.03071 [cs]*, May 2017, arXiv: 1705.03071. [Online]. Available: http://arxiv.org/abs/1705.03071

[20] N. Razin and N. Cohen, "Implicit Regularization in Deep Learning May Not Be Explainable by Norms," *arXiv:2005.06398 [cs, stat]*, Oct. 2020, arXiv: 2005.06398. [Online]. Available: http://arxiv.org/abs/2005.06398

[21] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for Deep Learning: A Taxonomy," *arXiv:1710.10686 [cs, stat]*, Oct. 2017, arXiv: 1710.10686 version: 1. [Online]. Available: http://arxiv.org/abs/1710.10686

[22] A. Khan and K. Fraz, "Post-training iterative hierarchical data augmentation for deep networks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[23] P. Kluska and M. Zieba, "Post-training Quantization Methods for Deep Learning Models," in *Intelligent Information and Database Systems*, ser. Lecture Notes in Computer Science, N. T. Nguyen, K. Jearanaitanakij, A. Selamat, B. Trawiński, and S. Chittayasothorn, Eds. Cham: Springer International Publishing, 2020, pp. 467–479.

[24] T. Moreau and J. Audiffren, "Post Training in Deep Learning with Last Kernel," *arXiv:1611.04499 [cs, stat]*, Oct. 2017, arXiv: 1611.04499. [Online]. Available: http://arxiv.org/abs/1611.04499

[25] Z. Qi, F. Meng, Y. Tian, L. Niu, Y. Shi, and P. Zhang, "Adaboost-LLP: A Boosting Method for Learning With Label Proportions," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3548–3559, Aug. 2018, conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[26] T. Cohen, M. Freytsis, and B. Ostdiek, "(Machine) learning to do more with less," *Journal of High Energy Physics*, vol. 2018, no. 2, p. 34, Feb. 2018. [Online]. Available: https://doi.org/10.1007/JHEP02(2018)034

[27] N. Lu, G. Niu, A. K. Menon, and M. Sugiyama, "On the minimal supervision for training any binary classifier from only unlabeled data," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=B1xWcj0qYm

[28] V. Zantedeschi, R. Emonet, and M. Sebban, "Beta-risk: a new surrogate risk for learning from weakly labeled data," in *Advances in Neural Information Processing Systems*, 2016, pp. 4365–4373.

[29] H. Reeve *et al.*, "Classification with unknown class-conditional label noise on non-compact feature spaces," in *Conference on Learning Theory*. PMLR, 2019, pp. 2624–2651.

[30] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*, 2018, pp. 4390–4399.

[31] T. Y. Yap, "Text anomaly detection with arae-anogan," 2020. [Online]. Available: https://digitalcommons.iwu.edu/cs_honproj/22

[32] R. Kannan, H. Woo, C. C. Aggarwal, and H. Park, "Outlier detection for text data," in *Proceedings of the 2017 siam international conference on data mining*. SIAM, 2017, pp. 489–497.