Identifying Wrongly Predicted Samples: A Method for Active Learning

Rahaf Aljundi Nikolay Chumerin Daniel Olmeda Reino

Toyota Motor Europe

Abstract

While unlabelled data can be largely available and even abundant, the annotation process can be quite expensive and limiting. Under the assumption that some samples are more important for a given task than others, active learning targets the problem of identifying the most informative samples that one should acquire annotations for. In this work we propose a simple sample selection criterion that moves beyond the conventional reliance on model uncertainty as proxy to leverage new labels. By first accepting the model prediction and then judging its effect on the generalization error, we can better identify wrongly predicted samples. We also present a very efficient approximation to our criterion, providing a similarity-based interpretation. In addition to evaluating our method on the standard benchmarks of active learning, we consider the challenging yet realistic imbalanced data scenario. We show state-of-the-art results, especially on the imbalanced setting, and achieve better rates at identifying wrongly predicted samples than existing active learning methods. Our method is simple, model agnostic and relies on the current model status without the need for re-training from scratch.

1. Introduction

The success of deep learning relies on the availability of large annotated data in the order of millions or more, however, obtaining annotations for such a scale can be a time consuming and a very expensive procedure. Besides, in many applications, *e.g.*, semantic segmentation, samples are not equally important for the task being learned. Many samples can be redundant or easily predicted and annotating them would be a waste of resources. With the goal of improving the data labelling efficiency, active learning is a sub-field of machine learning that aims at identifying the most informative data points in a stream or a large pool of unannotated samples. Those identified samples are then annotated and added to the existing training set, which would contribute to a substantial performance gain upon retraining, a process that can be repeated until reaching a certain level of performance or consuming the annotation budget. A large body of research has been dedicated to active learning (see [21] for a survey), however, a need emerged for active learning methods targeting deep models. In this regard, methods for selecting labelling candidates rely either on the uncertainty in their current predictions [6, 25, 7] or on how representative the selected candidates are to the rest of the samples [20], or, alternatively, how different selected samples are from the current training data [24]. The last two methods [24, 20] show state-of-the-art results when extracting large batches of samples from balanced pools of unannotated data. However, the first requires solving a mixed integer optimization problem over all pool samples and the later trains a VAE and a discriminator on the pool and training samples prior to the selection process. As such, practitioners usually resort to the uncertainty sampling [6, 25, 7], however training a Bayesian neural network or at least a network with dropout layer(s) is still required which might not always be applicable or favorable. In this work, we are interested in developing a lightweight model agnostic sample selection method. We suggest that informative samples are those that the current trained model has predicted their output (label) wrongly and by acquiring their true labels, the model will gain access to new bits of knowledge. Now the question we aim at answering is if there is a better way of pointing at wrongly predicted samples other than the model uncertainty. We propose to identify those wrongly predicted samples by first hypothesizing that the model prediction is correct and attempt at increasing the confidence of the model in its initial prediction. We then measure the effect of this hypothesis on the model performance on a small holdout set. As increasing the confidence on wrong predictions would harm the model performance in contrary to correct predictions, we use the relative change in the model performance (or alternatively the error) as a criterion for selecting samples that are likely to be wrongly predicted. Our method is generic, efficient and requires no changes on the current model.

Aside from these aspects, we point that active learning methods have been mostly tested in settings where the pool of unannotated samples is artificially balanced over the different categories as in the case of most standard datasets. This

assumption is unrealistic in many cases and hides the potential of the different approaches, *e.g*., random sampling is only outperformed by a small margin. We argue that real life applications often face the problem of imbalanced set of samples and the condition where samples are balanced among different classes is solely met in existing benchmarks. In this paper, we consider the challenging setting of imbalanced pool of unannotated samples where not all categories are equally represented. We show that random selection is no longer a competitive baseline and requires significant extra amount of annotations in comparison to our method which targets regions where most mistakes occur and surpasses the imbalanced nature of the data.

Our contributions are as follows: 1) we propose a novel approach for sample selection based on their plausibility of being wrongly predicted by the current trained model. 2) We present an approximated variant of our method and demonstrate an interesting link with kernel based similarity measures, here from the trained model perspective. 3) We achieve state-of-the-art results especially on the realistic yet challenging imbalanced setting. In the following, we discuss closely related works in Section 2 and describe our proposed approach in Section 3. We evaluate our method on image classification problem, Section 4.1 and semantic segmentation problem, Section 4.3, we conclude in Section 5.

2. Related Work

In this paper, we consider a pool-based active learning setting, where annotation candidates are selected from a big pool of unlabelled data [21]. Under this setting, most dominant lines of work focus either on identifying current uncertain samples or a set of diverse and representative samples [5]. In contrast, our work comes closest to approaches that aim at selecting samples which, once annotated, would have the largest effect on the trained model. These include *largest expected model change* (LEMC) and *expected error reduction* (EER) methods. A prominent example of LEMC methods, EGL [22], selects samples based on an approximation to the expected value of the sample's gradient given the current predicted output distribution. Samples with largest gradient magnitude are selected for annotation. In our method approximation we do not depend only on the loss gradient's magnitude, but also on the angle between the loss gradient estimated on a pool sample and that estimated on a holdout set. Instead of estimating the model change by the expected gradient length, variance reduction methods [11, 12] aim at implicitly reducing the generalization error by selecting candidates that would minimize the model output variance through the reliance on Fisher information. Closer to our approach, EER methods, estimate explicitly how much the generalization error will be reduced as in [19]. For each candidate, the model is trained on each possible label and the generalization error is computed on other pool samples, approximated with the current model output distribution and further averaged over the different possible labels of the candidate. In our work, we also aim at reducing the generalization error of the model when the selected samples are correctly annotated, however, we select samples that affect most *negatively* the generalization error when using their *current* predicted labels as ground truth. We use this as a proxy to identify wrongly predicted labels. Aside from the novel deployed criterion, in this work, we introduce a series of steps to make such approaches applicable to deep learning. Instead of estimating an expectation of the loss on the pool set, we deploy the typically required validation set to estimate the generalization error and instead of considering all possible labels while estimating the expected model update we rely on pseudo labels. More importantly, we present an efficient approximation and show how our criterion can be interpreted as selecting samples that are dissimilar to those in the holdout set.

When considering active learning methods designed for deep learning, several paradigms have emerged such as uncertainty based sampling [6, 7, 14], representation based sampling [20, 24] and query by committee using ensemble of models [8, 23]. Uncertainty based methods are closer in nature to our approach. Gal *et al*. in [6] showed that MC-Dropout can be used to perform approximate Bayesian inference in deep neural networks, and applied it to high-dimensional image data in [14] to estimate uncertainty as a selection criterion. [25] combine the obtained annotations of uncertain samples and the pseudo labels of the most certain samples. Our approach moves beyond uncertainty by selecting samples that are likely to be wrongly predicted through deploying pseudo labels as a proxy to estimate the change in generalization error. Very recently, [1] propose to rely on the gradient magnitude as a measure of uncertainty while selecting diverse candidates. In our approximation, we operate in the gradient space where not only the magnitude of the gradients is in effect but also the angle with the holdout set gradient.

3. Our Approach

We consider the pool based active learning setting where access is only assumed to a small initial set X^t of labelled training data along with a much larger set of unannotated data X^p (pool). The active learning method $\mathcal{M}$ should identify a set of K most informative samples X^a to be annotated and added to the training set which should contribute to a maximal gain in the trained model performance. K is the size of the annotation step s. This process can be repeated until reaching a certain performance level or exhausting annotation resources with S annotation steps performed. Given a model parameterized by θ, we want to learn a function $f(x;\theta)$ that maps the input data x to their corresponding labels y. We start by training the model on the initial training

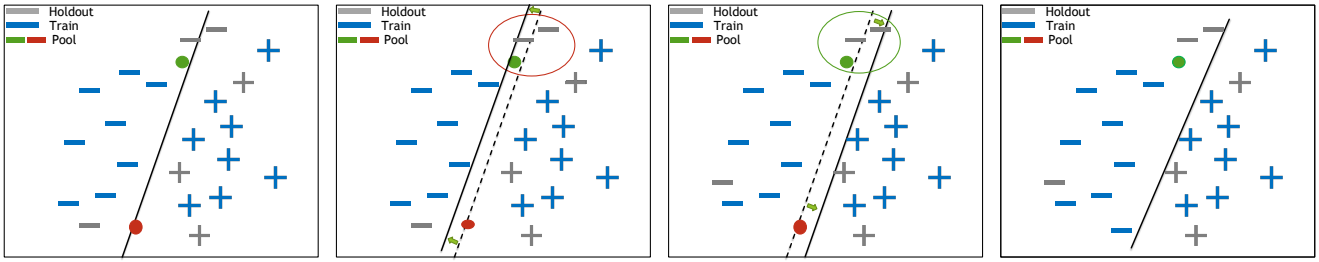


Figure 1: Illustration of the main idea of our approach (best viewed in color). (a) Shows the initial training set in blue, the validation set in gray, used here as a holdout set, and the pool samples (circles) in addition to the current learned decision boundary. The two pool samples (with ground truth of negative labels) are uncertain with one (in green) correctly predicted and the other (in red) wrongly predicted. (b) Illustrates how the decision boundary would move if we make small training step(s) with the prediction of the red pool sample as a true label, while (c) illustrates the effect of utilizing the correctly predicted output of the green pool sample instead. Using the (wrongly predicted) label of the red pool sample would harm the generalization performance as some correctly predicted samples will be misclassified as opposed to minimizing the loss on the correctly predicted sample. Hence, our method selects the red point for annotation in order to obtain its correct label. Finally, (d) shows the decision boundary after adding the newly annotated sample to the training set.

set X^t, considered as a starting step $s = 0$:

$$\theta^s = \arg\min_{\theta} \ell\left(f(X^t;\theta), Y^t\right). \quad (1)$$

The goal is to estimate each pool sample's utility, which is related to the amount of information conveyed when pairing the sample with the correct output. Providing already known labels might not be beneficial compared to correcting current mistakes. While a popular line of works [6, 25, 7] rely on uncertainty, we argue that a model can be uncertain about a sample and yet predict its output correctly. In this work, we propose a new method to spot wrong predictions.

We consider a standard classification problem, where the target class y is estimated from the model output $f(x;\theta)$. Starting from the initial predictions of the model, the entropy of these predictions can be used as a measure of uncertainty. Minimizing this entropy, which is usually deployed in unsupervised or semi-supervised learning [9, 2, 10], would push the prediction towards the most probable label and suppress other labels probabilities. Assume that the predicted label is correct, then maximizing the confidence in this prediction through maximizing the log-likelihood of the predicted label or alternatively minimizing the cross-entropy loss using the first predicted label as a pseudo label could help improving the model performance. However, if the initial model prediction is wrong, then minimizing the loss on a wrong label could harm the model performance as we are injecting false information in the model. We use this reasoning as a base for designing a measure to select wrongly predicted samples. Figure 1 illustrates the main idea behind our approach.
Formally, for each candidate sample $x_i^p \in X^p$, we first obtain its prediction $\hat{y}_i$ by the current model (with $\theta = \theta^s$). In unsupervised learning methods, $\hat{y}_i$ can be seen as a pseudo label and used to train the model [17]. Here, we first assume that the prediction given by the pseudo label $\hat{y}_i$ is correct and through minimizing the loss ℓ on the current sample x_i^p given the pseudo label $\hat{y}_i$ we obtain θ_i^p as:

$$\theta_i^p = \arg\,\mathrm{loc}\,\min_{\theta,\theta^s} \ell\left(f(x_i^p;\theta), \hat{y}_i\right). \quad (2)$$

Here $\arg\,\mathrm{loc}\,\min_{\theta,\theta_s}$ denotes the argument-result of the local minimization w.r.t. θ starting from θ^s. Now, as explained earlier, our hypothesis suggests that if the current model prediction is wrong, the minimization of the model loss given that prediction as a target label would harm the model performance. This can be due to moving the decision boundary in the wrong direction or to relying on a feature that is apparent or representative of another category. We use this rule as a proxy to identify the samples that are likely to be wrongly predicted. We propose to select samples using the estimate of the change in the model generalization error, *i.e.*, the change in the model prediction error on unseen samples:

$$\mathcal{S}(x_i^p) = \ell\left(f(X^v;\theta_i^p), Y^v\right) - \ell\left(f(X^v;\theta^s), Y^v\right). \quad (3)$$

To measure the generalization error, we employ a small set X^v, this can be a small holdout set or the validation set used for setting hyper-parameters and estimating the model performance. We then select K samples with largest values of $\mathcal{S}$ and request their annotations. The newly labelled data are to be added to the training set on which the model will be trained again and then a new active learning step can be carried out. It can be noted that in general the last K samples whose updated model loss decreases are likely to be correctly predicted by the current model and could be combined with the training pool to be learned in an "unsupervised" manner as in [25].
To summarize, instead of relying only on the model uncertainty to find the annotation candidates, we use the change

in the generalization error on a holdout set to identify the wrongly predicted samples.

3.1. First Order Approximation

Our criterion involves an estimation of the updated model loss on a holdout set after the loss minimization on each pool sample given its pseudo label. As we shall see in the experiments, Section 4.1, the holdout set can be very small and its loss can be estimated in one forward pass. Nonetheless, to account for scenarios with extreme constraints on computational cost and more importantly to gain better insights on the proposed method behaviour, we analyze and present an approximation to the selection criterion in (3). Let's define:

$$\ell_v(\theta) = \ell\left(f(X^v;\theta), Y^v\right) = \sum_j \ell\left(f(x_j^v;\theta), y_j^v\right). \quad (4)$$

Then (3) can be rewritten as follows:

$$\mathcal{S}(x_i^p) = \ell_v(\theta_i^p) - \ell_v(\theta^s). \quad (5)$$

We expand the first term about θ^s using first order Taylor series approximation.

$$\ell_v(\theta_i^p) \approx \ell_v(\theta^s) + \nabla_\theta \ell_v(\theta^s) \cdot (\theta_i^p - \theta^s), \quad (6)$$

where $\nabla_\theta \ell_v(\theta^s)$ denotes the gradient of $\ell_v(\theta)$ w.r.t. the parameter θ at point $\theta = \theta^s$. Instead of doing local optimization of the loss (mentioned in (2)), we propose to estimate θ_i^p by a single step of gradient descent from θ^s with a learning rate η and the loss gradient estimated at sample x_i^p:

$$\theta_i^p \approx \theta^s - \eta \nabla_\theta \ell(f(x_i^p;\theta^s), \hat{y}_i). \quad (7)$$

Then, using this estimate in the right-hand part of (6), we obtain:

$$\begin{aligned}\ell_v(\theta_i^p) &\approx \ell_v(\theta^s) + \nabla_\theta \ell_v(\theta^s) \cdot (\theta^s - \eta \nabla_\theta \ell(f(x_i^p;\theta^s), \hat{y}_i) - \theta^s) \\ &= \ell_v(\theta^s) - \eta \nabla_\theta \ell_v(\theta^s) \cdot \nabla_\theta \ell(f(x_i^p;\theta^s), \hat{y}_i), \quad (8)\end{aligned}$$

$$\begin{aligned}\mathcal{S}(x_i^p) &\approx \ell_v(\theta^s) - \eta \nabla_\theta \ell_v(\theta^s) \cdot \nabla_\theta \ell(f(x_i^p;\theta^s), \hat{y}_i) - \ell_v(\theta^s) \\ &= -\eta \nabla_\theta \ell_v(\theta^s) \cdot \nabla_\theta \ell(f(x_i^p;\theta^s), \hat{y}_i). \quad (9)\end{aligned}$$

The positive constant η in (9) has no influence on the order of x_i^p sorted by decreasing $\mathcal{S}(x_i^p)$, and, therefore, can be simply dropped. We propose the following alternative criterion and consider both criteria in the experiment Section 4.1.

$$\mathcal{S}_a(x_i^p) = -\nabla_\theta \ell_v(\theta^s) \cdot \nabla_\theta \ell(f(x_i^p;\theta^s), \hat{y}_i). \quad (10)$$

The estimation of the selection criterion $\mathcal{S}_a(x_i^p)$ does not involve the loss minimization of (2) as in the original criterion $\mathcal{S}(x_i^p)$, but uses only the estimation of the pool sample gradient. It also replaces the estimation of the holdout set loss in (3) for each pool sample with a single prior computation of the loss gradient on the holdout set.

3.1.1 A Similarity Based Interpretation

Here we want to present our criterion as a measure of dissimilarity between a given pool sample and samples of the holdout set based on the currently trained model. Let's define the following kernel:

$$K_\theta(x_i, x_j) = \nabla_\theta \ell(f(x_i;\theta), y_i) \cdot \nabla_\theta \ell(f(x_j;\theta), y_j). \quad (11)$$

Each term in (11) can be expanded using the chain rule as follows:

$$\nabla_\theta \ell(\theta) = (\nabla_\theta f(x_i;\theta))^\intercal \, \ell', \quad (12)$$

where $\ell' \in \mathbb{R}^C$ denotes the derivative of ℓ w.r.t. $f(x_i;\theta)$, and $\nabla_\theta f(x_i;\theta) \in \mathbb{R}^{C \times P}$ with C the number of categories and P the size of the parameter vector. The kernel can be written as follows:

$$K_\theta(x_i, x_j) = ((\nabla_\theta f(x_i;\theta))^\intercal \, \ell') \cdot ((\nabla_\theta f(x_j;\theta))^\intercal \, \ell'). \quad (13)$$

This kernel is related to the Neural Tangent Kernel (NTK) [13], $K_{NTK}(x_i, x_j) = \nabla_\theta f(x_i;\theta) \cdot \nabla_\theta f(x_j;\theta)$, studied from an optimization point of view in the infinite width limit; it describes how changing the network function at one point affects its output on another. A kernel similar to NTK was proposed in [3] to measure the similarity between samples from a trained network perspective for dataset self denoising. In the same study it was shown that from $\nabla_\theta f(x_i;\theta) = \nabla_\theta f(x_j;\theta)$ follows $f(x_i;\theta) = f(x_j;\theta)$, and samples with dissimilar features have orthogonal gradient directions and kernels value close to zero. The kernel, defined in (13), considers additionally the gradient of the loss function accounting for the sample/label pair. For example, in the case of a cross-entropy loss, we have $K_\theta(x_i, x_j) = (\nabla_\theta f(x_i;\theta)^\intercal \, (p_i - y_i)) \cdot (\nabla_\theta f(x_j;\theta)^\intercal (p_j - y_j))$ with y_i here constructed as a one-hot label vector and p_i the output probability, which can be seen as weighting the function gradient by the difference between the predicted class probabilities and the target/pseudo labels. See supplementary materials for details on the derivation.

Finally, given that $\nabla_\theta \ell_v(\theta) = \sum_j \nabla_\theta \ell(f(x_j^v;\theta), y_j^v)$, where j is the index of the holdout samples, our approximated criterion can be rewritten as follows:

$$\mathcal{S}_a(x_i^p) = -\sum_j K_\theta(x_i^p, x_j^v). \quad (14)$$

Following that, our criterion allows to select the samples that are dissimilar to those in the holdout set according to the kernel defined in (11).

Binary classification example. Let us demonstrate the proposed sample selection method on a binary classification problem employing a single-layer neural network parametrized by θ. Assume the input to the network is a feature vector $\phi(x) \in \mathbb{R}^n_{\geq 0}$ extracted from a sample x with a fixed (non-trainable) feature extractor ϕ. The function being

learned is $f_\theta(\phi(x)) = \theta^\intercal \phi(x)$. By defining $z = f_\theta(\phi(x))$, and the loss $\ell(z, y) = -y \log(\sigma(z)) - (1-y)\log(1-\sigma(z))$, where $y \in \{0, 1\}$ is the binary label, and $\sigma(z) = \frac{1}{1+e^{-z}}$; the gradient of the loss w.r.t. θ can be derived using the chain rule:

$$\nabla_\theta \ell(z, y) = \frac{\partial \ell(z, y)}{\partial \theta} = \frac{\partial \ell}{\partial \sigma}\frac{\partial \sigma}{\partial z}\frac{\partial z}{\partial \theta} = (\sigma(z) - y)\phi(x). \tag{15}$$

Following this definition, our criterion for selecting pool samples is:

$$\mathcal{S}_a(x_i^p) = -\sum_j (\sigma(z_i^p) - \hat{y}_i)\phi(x_i) \cdot (\sigma(z_j^v) - y_j^v)\phi(x_j^v). \tag{16}$$

Let us analyze the kernel value w.r.t. a pool sample x_i^p and a sample from the holdout set x_j^v.

$$\begin{aligned} K_\theta(x_i^p, x_j^v) &= c\,\phi(x_i^p) \cdot \phi(x_j^v), \text{with scalar } c \\ &= (\sigma(z_i^p) - \hat{y}_i)(\sigma(z_j^v) - y_j^v). \end{aligned} \tag{17}$$

Consider the following cases: 1) the feature vectors $\phi(x_i^p)$ and $\phi(x_j^v)$ are different and $\phi(x_i^p) \cdot \phi(x_j^v) \approx 0$, resulting in $K_\theta(x_i^p, x_j^v) \approx 0$; 2) x_i^p and x_j^v are close in the feature space and $\phi(x_i^p) \cdot \phi(x_j^v) \gg 0$. In the latter case, either $\hat{y}_i \neq y_j^v$ and $c < 0$, causing $K_\theta(x_i^p, x_j^v) \ll 0$ (case 2a), or $\hat{y}_i = y_j^v$ and $c > 0$, leading to $K_\theta(x_i^p, x_j^v) \gg 0$ (case 2b). Therefore, if x_i^p differs significantly from all holdout samples (case 1), then $\mathcal{S}_a(x_i^p) \approx 0$. However, if x_i^p is similar to some holdout samples and it is predicted incorrectly (case 2b), then $\mathcal{S}_a(x_i^p)$ is likely to be positive, otherwise, when the prediction is correct (case 2b), the corresponding $\mathcal{S}_a(x_i^p)$ is likely to be negative. Consequently, the pool samples from both former cases would get greater (than the samples from the later case) values of the selection criterion and, therefore, will be selected for annotation.

In a nutshell, our method aims at selecting pool samples that differ from the holdout samples firstly in their (probably wrongly) predicted label or in their feature representation.

4. Experiments

To evaluate the effectiveness of our approach in various active learning scenarios, we perform a wide set of experiments on both image classification (Section 4.1), and image segmentation (Section 4.3).

4.1. Image Classification

We first describe the details of our experiments:

Datasets. We consider MNIST [16] dataset for handwritten digit recognition, KMNIST [4] an MNIST style dataset for Kanji characters composed of 10 classes, SVHN [18] Google street view house numbers dataset and CIFAR10 [15].

Compared methods. - `Random`: a random subset of the pool is selected for annotation at each step.

-`Err-Reduction` [19]: an implementation of the error reduction approach using pseudo labels and error estimation on subset of the pool.

- `MC-Dropout` [6]: uses as a criterion the model uncertainty of each pool sample.

- `Coreset` [20]: selects a set of representative samples covering the rest of the pool. The method presents a mixed integer programming solution and a greedy alternative that is only $1 - 2\%$ inferior, we employ this efficient alternative.

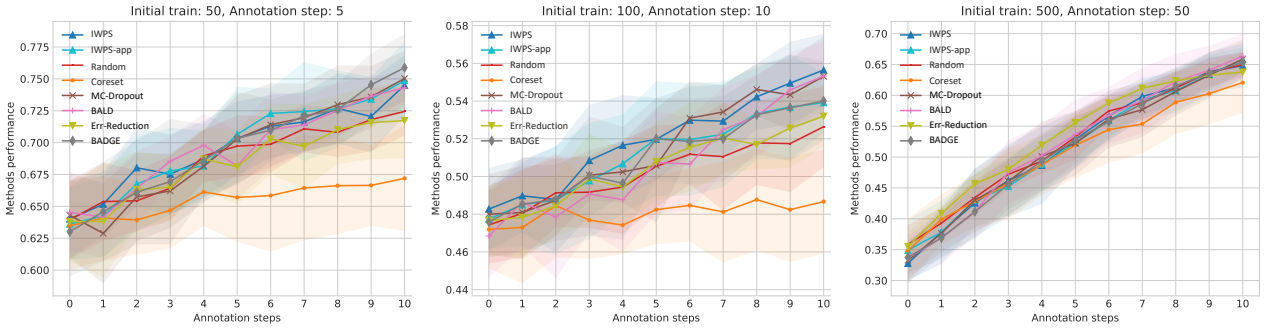
- `BALD` [7]: it is based on the mutual information between the prediction and the model posterior.

-`BADGE` [1] selects diverse and uncertain samples in a gradient space of pool samples based on their the pseudo labels.

Implementation. We deploy a two-layer fully connected network for MNIST and KMNIST datasets, and ResNet18 for SVHN and CIFAR10. All methods were trained using ADAM with early stopping on the validation set. We don't retrain the model from scratch after each annotation step, we rather continue training the model and only reset the optimizer parameters. This makes more sense since the new samples are selected based on a criterion linked to the previously trained model. We apply this to all methods and observe consistent improvement. For our method, Identifying Wrongly Predicted Samples, we consider both the criterion in (3) and refer to it as `IWPS`, and the first order approximation criterion in (10) and refer to it as `IWPS-app`. We use the initial validation set as a holdout set to estimate the criterion of both `IWPS` and `IWPS-app`. Note that in our experiments we keep the validation set fixed to the initial setting while in practice one can augment it as new labels are obtained. We limit the optimization of the loss in (2) to the last layer parameters and similarly for the gradients estimation of (10) of `IWPS-app` criterion. This has a valuable advantage computationally and shows no significant effect on the performance, see supplementary materials . The minimization of (2) in `IWPS` is performed with SGD and limited to 3 iterations with learning rate $\eta = 10^{-3}$ on the fully connected model and $\eta = 10^{-4}$ on ResNet. We report the average of 10 runs with different random seeds along with the standard deviation. We refer to the supplementary materials for a discussion on the computational cost of `IWPS` and `IWPS-app`. Note that our `IWPS-app` has a cost of approximately one forward pass, the best computation cost after the random selection baseline.

4.1.1 Identifying Wrongly Predicted Samples

A main question of this work is if there is a better way in identifying wrongly predicted samples than the conventional use of uncertainty. Those wrongly predicted pool samples, which, once identified and annotated, should improve the model performance, as it is being trained on previously unknown cases. We first validate how well our method can do that by inspecting the percentage of selected samples with wrong predictions in comparison to `BALD` and `MC-Dropout`. Table 1 reports the wrong predictions rate



(a) (b) (c)

Figure 2: Mean accuracy and std.dev. for (a) MNIST, (b) KMNIST and (c) SVHN on balanced setting.

among the selected samples at the indicated steps based on MNIST benchmark. Both our criteria are best at picking wrongly predicted samples, with `IWPS` achieving a higher mistake selection rate, 10%–20%, than others.
Having shown this, we next study the active learning performance of our selection criteria compared to other methods.

	Step 1	Step 5	Step 10
BALD	72.2%	57.6%	56.6%
MC-Dropout	70.4%	62.2%	60.6%
IWPS-app	91.8%	67.2%	62.0%
IWPS	90.0%	74.0%	70.0%

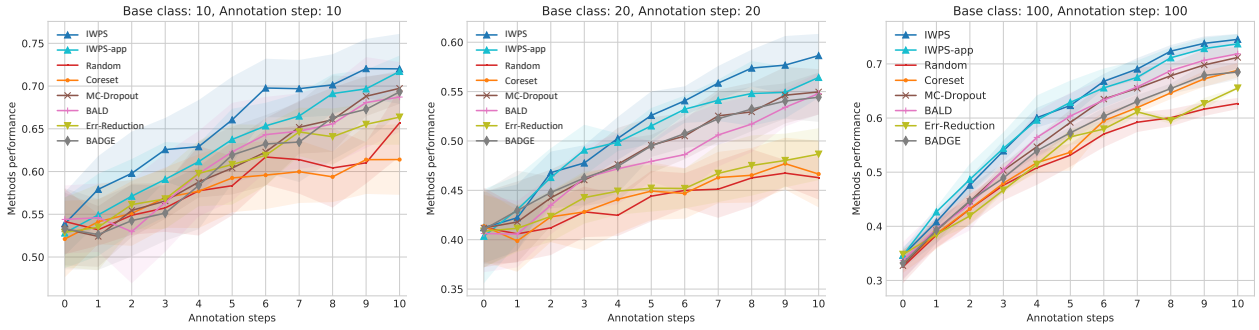
Table 1: Percentage of wrong prediction among selected samples, on MNIST with 50 initial train and 50 annotation step.

4.1.2 Balanced Setting

The used datasets are standard in the field and they are composed of a similar number of samples per category. As such, the pool and the randomly sampled initial training set represent equally each category. We refer to this setting as balanced setting. Here, we use a validation set of the same size as the initial training set. Each active learning round is composed of 10 annotation steps with each step being 10% of the initial training set size. The initial training set size is relative to each dataset difficulty and the amount of samples needed to obtain a reasonable initial performance. We use the following initial sizes 50, 100, 500, 5000 for MNIST, KMNIST, SVHN and CIFAR10 respectively. Figures 2a, 2b, 2c and 4a report the accuracy of the compared methods after each annotation step on MNIST, KMNIST, SVHN and CIFAR10 datasets respectively. See supplementary materials for more results. While most methods improve over random sampling, the margin of improvement is limited (2%–3%). When considering all the studied datasets, our proposed method `IWPS` and its approximation `IWPS-app` perform similarly to `BALD`, `MC-Dropout`, and `BADGE`. Our `IWPS-app` is the fastest to compute, as discussed in the supplementary materials.

4.1.3 Imbalanced Setting

In the standard balanced setting, `Random` appears to be a competitive baseline and only outperformed by a small margin as also shown in [24]. Here, we argue that random sampling cannot be taken for granted as a competitive method in the cases where there are dominant categories that are not of much importance to the task at hand as opposed to the under-represented ones. For example, in autonomous driving applications most images contain examples of road, sky or buildings, but other categories like cyclists or trains are much less frequent. We simulate this scenario by constructing a pool of samples in which half of the categories are under-represented with number of samples equals to $1/10$ of other categories samples (base number of samples per class). Since the compared methods start with initial training set, we also construct this set with the same imbalanced setting. Regarding the validation set, we keep it balanced but limit its size to $1/5$ of the initial training size, note that the test set on which we report the accuracy remains balanced. We apply this setup to the 4 studied datasets. As this is a much harder setting, for each dataset we double the base class size of the initial training set compared to the expected per category size in balanced setting, we also double the annotation step size.
Figures 3a, 3b, 3c, 4b report the test accuracy of each of the compared methods after each annotation step on the described imbalanced setting of MNIST, KMNIST, SVHN and CIFAR10 respectively, see supplementary materials for more results with larger imbalanced rates and different annotation steps. This setting shows larger differences between the compared methods. It is clear that `Random` baseline fails to compete here, for example, on MNIST `IWPS` with only 5 annotation steps achieves the same accuracy of `Random` after 10 annotation steps. This is an important reduction of half of the annotation resources. Uncertainty and information based methods `BALD`, `BADGE` and `MC-Dropout` continue to improve over `Random` with a larger margin in this setting. `Coreset`, in the contrary, has consistently lower perfor-



(a) (b) (c)

Figure 3: Mean accuracy and std.dev. for (a) MNIST, (b) KMNIST and (c) SVHN on imbalanced setting. Our `IWPS` & `IWPS-app` show significantly better performance.

mance, we think that this method is more suitable for extracting much larger batches of data. Our method `IWPS` achieves the best performance on MNIST, KMNIST and SVHN settings. We believe that this is a significant improvement that shows our method ability to identify those underrepresented categories where most mistakes occur, and thereof selects most informative samples contributing to higher gains in the trained model performance. Notably, `IWPS` is constantly outperforming `Err-Reduction`. This indicates that using the largest generalization error as a proxy for identifying wrongly predicted samples provides the model with previously unknown knowledge and results in a larger reduction of the newly trained model generalization error than when aiming at explicitly reducing it.

For CIFAR10 dataset, all methods perform closely; we think it is due to the low intra class similarity of this dataset which reduces the potential impact of the individual selected samples. `IWPS-app` improves over other methods on MNIST, KMNIST and SVHN benchmarks with slightly less margin than `IWPS` and achieves comparable performance on CIFAR10. This is still outstanding given its low complexity compared to other competitors.

4.2. Ablation

Having shown the effectiveness of our method especially on the important case of imbalanced data pool, in this section we discuss our design choices e.g., the holdout set selection and size.

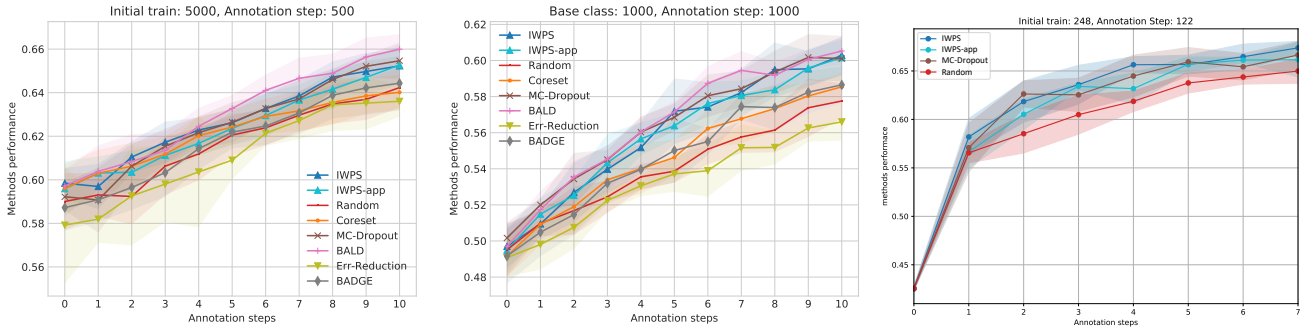
4.2.1 Holdout Set Size

Our criterion is based on the generalization error of the model $f(x;\theta)$, for $\theta = \theta_i^p$ – the updated parameters of each pool sample, estimated on a holdout set regarded as representative of the different concepts (categories). In the previous experiments, we have used the validation set for estimating our selection criterion. In this section, we study the effect of the used set choice on the behaviour of our selection criteria. We first answer the question of how many samples are needed for our method to make a reasonable selection. Figure 5a reports the performance of `IWPS` on imbalanced SVHN benchmark for different sizes of the holdout set $(1-40)$ per class, see supplementary for MNIST imbalanced and balanced setting. However, there are no significant differences for larger sizes. As it can be seen, only the very smallest holdout set sizes decay our method performance. This empirical evidence suggests that with only few samples (> 2) per class, our method can reach its best performance and, in spite of the important role of this holdout set, its size does not affect significantly the performance of our method. While one would expect that the more complex the ground-truth concept is the more holdout data is needed to estimate the generalization error; we argue that this is also the case of the typically deployed validation set. We indeed require similar and even smaller size to rank the pool samples than what needed to tune the hyper-parameters of a deep neural networks. We should note that each class is required to have a representative sample(s) and when a category is under-represented or over-represented, this would affect its contribution to the generalization error estimation and hence the ranking of the pool samples.

4.2.2 Holdout Set Alternative

The next discussion point is whether a subset of the training set instead of a holdout set can be used for the error change estimation. Here, the error will no longer be a generalization error. However, the interpretation of the selection criteria behaviour is still valid. Pool samples will be selected if utilizing their pseudo labels as ground-truth would harm the performance on seen data here instead of holdout data. Following our approximation criterion discussion, samples different from those in the training set either in their predicted labels or features, will be selected first. We examine our methods' (`IWPS` and `IWPS-app`) performance change, when deploying a random subset of the training data in the selection strategy instead of a holdout set.

Figure 5b reports on SVHN imbalanced setting the per-



(a) (b) (c)

Figure 4: Mean accuracy and std.dev. for (a) CIFAR10 balanced setting and (b) CIFAR10 imbalanced setting. (c) Semantic segmentation results, mIoU and std.dev. on Cityscapes.

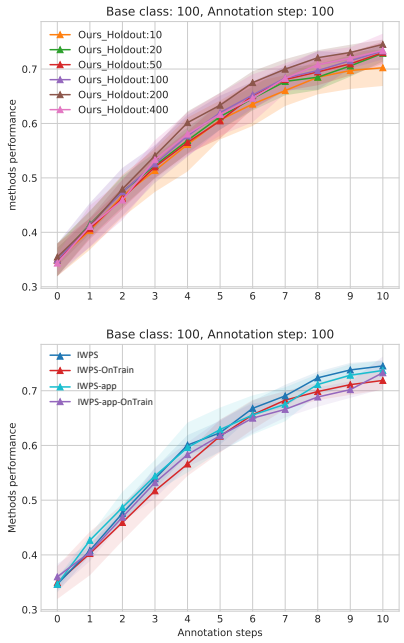


Figure 5: Top (5a) holdout set size ablation, bottom (5b) reports the performance when using a training subset vs a holdout set, both on the imbalanced SVHN setting.

formance of our both variants when using a holdout set (`IWPS`, `IWPS-app`) and when using a subset of the training data (`IWPS-OnTrain`, `IWPS-app-OnTrain`). The performance suffers from a small drop $1-2\%$ when using the training set. Empirically, it seems that our method can still select informative samples even when only relying on the same training samples for estimating the criteria. In supplementary materials we report the use of a training subset in other benchmarks. The only noticeable performance drop was on Imbalanced MNIST. It is a simple dataset and the training error can get to zero $\ell_v(\theta^s) \rightarrow 0$ which would lead to a close to zero norm of the training subset gradient and a noisy direction, thus hindering our selection criteria.

Finally, we note that the use of validation sets is unavoidable with deep neural networks as other strategies are prohibitively expensive and that deep learning based AL works usually deploy relatively big validation sets.

4.3. Semantic Segmentation Experiment

In the previous experiments, each sample has only one possible true class. We are interested in the case were each sample contributes to multiple and possibly conflicting hypotheses, thus we consider semantic segmentation.
We mainly compare to `Random` and `MC-Dropout` described previously, Section 4.1. For `IWPS` and `IWPS-app`: we only optimize the parameters or compute the gradients on the last two convolutional layers. We adapt `IWPS` to the case where an imperfect model produces both correct and incorrect predictions for one sample. We average pixel predictions $\mathcal{S}$ (3), where $\ell\left(f(X^v;\theta_i^p), Y^v\right) > \ell\left(f(X^v;\theta^s), Y^v\right)$ holds, and by doing so we only consider the subset of the frame that indicates that the model has been negatively impacted by the pool sample. See supplementary for further details and results. Fig. 4c shows the mean Intersection over Union (mIoU) after each annotation step. `IWPS` and `MC-Dropout` performs closely, with `IWPS` having slightly higher mIoU scores towards the end.

5. Conclusion

We propose a new solution to the problem of active learning that first accepts the hypothesis of the current model prediction on each pool sample and then judges the effect of increasing this hypothesis confidence on the performance on a holdout set. We use the change in the model generalization error as an indication of how likely the prediction of a given sample is to be mistaken. We further develop an approximation of our selection criterion and show that it targets sample/prediction pairs that are dissimilar to those in the holdout set from the current model perspective. We evaluate our approach on several benchmarks and achieve comparable performance to state-of-the-art methods. Additionally, we setup for the first time a systematic comparison on the more realistic imbalanced setting where we show significant improvements. Our method is computationally efficient and requires no changes on the available model. As a future work, we will explore possible alternatives to the holdout set in the case of very low data regime.

References

[1] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020.

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019.

[3] Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. Input similarity from the neural network perspective. In *Advances in Neural Information Processing Systems*, pages 5343–5352, 2019.

[4] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

[5] Yifan Fu, Xingquan Zhu, and Bin Li. A survey on instance selection for active learning. *Knowledge and information systems*, 35(2):249–283, 2013.

[6] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[7] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

[8] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. Query by committee made real. In *Advances in neural information processing systems*, pages 443–450, 2006.

[9] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[10] Alicia Guerrero-Curieses and Jesus Cid-Sueiro. An entropy minimization principle for semi-supervised terrain classification. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 3, pages 312–315. IEEE, 2000.

[11] Hamed Hassanzadeh and MohammadReza Keyvanpour. A variance based active learning approach for named entity recognition. In *International Conference on Intelligent Computing and Information Science*, pages 347–352. Springer, 2011.

[12] Steven CH Hoi, Rong Jin, and Michael R Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, pages 633–642, 2006.

[13] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[14] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc., 2017.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2, 2013.

[18] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[19] N Roy and A McCallum. Toward optimal active learning through sampling estimation of error reduction. int. conf. on machine learning, 2001.

[20] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[21] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[22] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296, 2008.

[23] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294, 1992.

[24] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5972–5981, 2019.

[25] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.