AN EMPIRICAL STUDY OF EXAMPLE FORGETTING DURING DEEP NEURAL NETWORK LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Inspired by the phenomenon of catastrophic forgetting, we investigate the learning dynamics of neural networks as they train on single classification tasks. Our goal is to understand whether a related phenomenon occurs when data does not undergo a clear distributional shift. We define a “forgetting event” to have occurred when an individual training example transitions from being classified correctly to incorrectly over the course of learning. Across several benchmark data sets, we find that: (i) certain examples are forgotten with high frequency, and some not at all; (ii) a data set’s (un)forgettable examples generalize across neural architectures; and (iii) based on forgetting dynamics, a significant fraction of examples can be omitted from the training data set while still maintaining state-of-the-art generalization performance.

1 INTRODUCTION

Many machine learning models, in particular neural networks, cannot perform continual learning. They have a tendency to forget previously learned information when trained on new tasks, a phenomenon usually called catastrophic forgetting [Kirkpatrick et al., 2017; Ritter et al., 2018]. One of the hypothesized causes of catastrophic forgetting in neural networks is the shift in the input distribution across different tasks—i.e., a lack of common factors or structure in the inputs of different tasks might lead standard optimization techniques to converge to radically different solutions each time a new task is presented. In this paper, we draw inspiration from this phenomenon and investigate the extent to which a related forgetting process occurs as a model learns examples traditionally considered to belong to the same task.

Similarly to the continual learning setting, in stochastic gradient descent (SGD) optimization, each mini-batch can be considered as a mini-“task” presented to the network sequentially. In this context, we are interested in characterizing the learning dynamics of neural networks by analyzing (catastrophic) example forgetting events. These occur when examples that have been “learnt” (i.e., correctly classified) at some time $t$ in the optimization process are subsequently misclassified — or in other terms forgotten — at a time $t' > t$. We thus switch the focus from studying interactions between sequentially presented tasks to studying interactions between sequentially presented dataset examples during SGD optimization. Our starting point is to understand whether there exist examples that are consistently forgotten across subsequent training presentations and, conversely, examples that are never forgotten. We will call the latter unforgettable examples. We hypothesize that specific examples consistently forgotten between subsequent presentations, if they exist, must not share commonalities with other examples from the same task. We therefore analyze the proportion of forgettable/unforgettable examples for a given task and what effects these examples have on a model’s decision boundary and generalization error.

The goal of our investigation is two-fold. First, we attempt to gain insight into the optimization process by analyzing interactions among examples during learning and their influence on the final decision boundary. We are particularly interested in whether we can glean insight on the compressibility of a dataset, and thereby increase data efficiency without compromising generalization accuracy. It is a timely problem that has been the recent focus of few-shot learning approaches via meta-learning [Finn et al., 2017; Ravi & Larochelle, 2017]. Second, we aim to characterize whether forgetting statistics can be used to identify “important” samples and detect outliers and examples with noisy labels [John, 1995; Brodley & Friedl, 1999; Sukhbaatar et al., 2014; Jiang et al., 2018].
Identifying important, or most informative examples is an important line of work and was extensively studied in the literature. Techniques of note — among others — are predefined curricula of examples (Bengio & LeCun, 2007), self-paced learning (Kumar et al., 2010), and more recently meta-learning (Fan et al., 2017). These research directions usually define “hardness” or “commonality” of an example as a function of the loss on that particular example at some point during training (or possibly at convergence). They do not consider whether some examples are consistently forgotten throughout learning. Very recently, Chang et al. (2017) consider re-weighting examples by accounting for the variance of their predictive distribution. This is related to our definition of forgetting events, but the authors provide little analysis of the extent to which the phenomenon occurs in their proposed tasks. Our purpose is to study this phenomenon from an empirical standpoint and characterize its prevalence in different datasets and across different model architectures.

Our experimental findings suggest that: a) there exist a large number of unforgettable examples, i.e., examples that are never forgotten once learnt, and those examples are correlated from one neural architecture to another; b) examples with noisy labels are among the most forgotten examples, along with images with “uncommon” features, visually complicated to classify; c) training a neural network on a dataset where a very large fraction of the least forgotten examples have been removed still results in extremely competitive performance on the test set.

2 RELATED WORK

Curriculum Learning and Sample Weighting  Curriculum learning is a paradigm that favors learning along a curriculum of examples of increasing difficulty (Bengio et al., 2009). This general idea has found success in a variety of areas since its introduction (Kumar et al., 2010; Lee & Grauman, 2011; Schaul et al., 2015). Kumar et al. (2010) implemented their curriculum by identifying easy examples as those which have a smaller loss. In our experiments, we will empirically validate that unforgettable examples can be safely removed without compromising strong generalization performance. Zhao & Zhang (2015); Katharopoulos & Fleuret (2018) relate sample importance to the norm of the loss gradient computed on the examples with respect to the parameters of the network. More recently, Fan et al. (2017); Kim & Choi (2018); Jiang et al. (2018) aim to learn a curriculum directly from data in order to minimize the task loss. Jiang et al. (2018) also study the robustness of their method in the context of noisy examples. This relates to a rich literature on the detection of outliers and the removal of examples with noisy labels (John, 1995; Brodley & Friedl, 1999; Sukhbaatar et al., 2014; Jiang et al., 2018). We will provide evidence that noisy examples rank higher with respect to the number of forgetting events.

Deep Generalization  The study of the generalization properties of deep neural networks when trained by stochastic gradient descent has been the focus of several recent publications (Zhang et al., 2016; Keskar et al., 2016; Chaudhari et al., 2016; Advani & Saxe, 2017). These studies suggest that the generalization error does not depend solely on the complexity of the hypothesis space. For instance, it has been demonstrated that over-parameterized models with many more parameters than training points can still achieve low test error (Huang et al., 2017; Wang et al., 2018) while being complex enough to fit a dataset with completely random labels (Zhang et al., 2016). A possible explanation for this phenomenon is a form of implicit regularization performed by stochastic gradient descent: deep neural networks trained with sgd have been recently shown to converge to the maximum margin solution in the linearly separable case (Soudry et al., 2017; Xu et al., 2018). In our work, we provide empirical evidence that generalization can be maintained when removing a substantial portion of the training examples and without restricting the complexity of the hypothesis class. This goes along the support vector interpretation provided by Soudry et al. (2017).

3 DEFINING AND COMPUTING EXAMPLE FORGETTING

Our general case study for example forgetting is a standard classification setting. Given a dataset \( D = \{ (x_i, y_i) \} \) of observation/label pairs, we wish to learn the conditional probability distribution \( p(y|x; \theta) \) using a deep neural network with parameters \( \theta \). The network is trained to minimize the empirical risk \( R = \frac{1}{|D|} \sum_{i} L(p(y|x_i; \theta), y_i) \), where \( L \) denotes the cross-entropy loss. The minimization is performed using variations of stochastic gradient descent, starting from initial random parameters \( \theta^0 \), and by sampling examples at random from the dataset \( D \).
Forgetting and learning events We denote $\hat{y}_t^i = \arg\max_k p(y_t^i | x_t^i; \theta^t)$ the predicted label (out of $K$ possible labels) for the example $x_t^i$ obtained after $t$ steps of SGD. We also let $\text{acc}^i_t = \mathbb{1}_{\hat{y}_t^i = y_t^i}$ be a binary variable indicating that the example is correctly classified at time step $t$. Example $i$ undergoes a forgetting event when $\text{acc}^i_t$ decreases between two consecutive updates: $\text{acc}^i_t > \text{acc}^i_{t+1}$. In other words, example $i$ is misclassified at step $t+1$ after having been correctly classified at step $t$. Conversely, a learning event has occurred if $\text{acc}^i_t < \text{acc}^i_{t+1}$.

Classification margin We will also be interested in analyzing the classification margin. Our predictors have the form $p(y_t^i | x_t^i; \theta) = \sigma(\beta(x_t^i))$, where $\sigma$ is a sigmoid (softmax) activation function in the case of binary (categorical) classification. The classification margin $m$ is defined as the difference between the logit of the correct class and the largest logit among the other classes, i.e. $m = \beta_k - \arg\max_{k \neq k'} \beta_k', \ k \in [K-1]$ where $K$ is the index corresponding to the correct class.

Unforgettable examples We qualify examples as unforgettable if they are learnt at some point and experience no forgetting events during the whole course of training: example $i$ is unforgettable if the first time it is learnt $t^*$ verifies $t^* < \infty$ and for all $k \geq t^*$, $\text{acc}^i_k = 1$. Note that, according to this definition, examples that are never learnt during training do not qualify as unforgettable. We refer to examples that have been forgotten at least once as forgettable.

3.1 Procedural Description and Experimental Setting

Following the previous definitions, monitoring forgetting events entails computing the prediction for all examples in the dataset at each model update, which would be prohibitively expensive. In practice, for each example, we subsample the full sequence of forgetting events by computing forgetting statistics only when the example is included in the current mini-batch; that is, we compute forgetting across presentations of the same example in subsequent mini-batches. This gives a lower bound on the number of forgetting events an example undergoes during training.

We train a classifier on a given dataset and record the forgetting events for each example when they are sampled in the current mini-batch. For the purposes of further analysis, we then sort the dataset’s examples based on the number of forgetting events they undergo. Ties are broken at random when sampling from the ordered data. Samples that are never learnt are considered forgotten an infinite number of times for sorting purposes.

We perform our experimental evaluation on three datasets of increasing complexity: 

- **MNIST** (LeCun et al., 1999), a permuted version of MNIST, and 
- **CIFAR-10** (Krizhevsky, 2009). We use various model architectures and training schemes that yield test errors comparable with the current state-of-the-art on the respective datasets. In particular, the MNIST-based experiments use a network comprised of two convolutional layers followed by a fully connected one, trained using SGD with momentum and dropout. This network achieves 0.8% test error. For CIFAR-10, we use a ResNet with cutout (DeVries & Taylor, 2017) trained using SGD and momentum with a particular learning rate schedule. This network achieves a competitive 3.99% test error. For full details of the experimentation, see the Supp. Mat.

4 Characterizing Example Forgetting

4.1 Distribution of Forgetting Events

Number of forgetting events We estimate the number of forgetting events of all the training examples for the three different datasets (MNIST, permutedMNIST and CIFAR-10) across 5 random seeds. The histograms of forgetting events computed from one seed are shown in Figure 1. Across

---

**Algorithm 1** Computing forgetting statistics.

```
initialize prev_\text{acc}_i = 0, i \in \mathcal{D}
initialize forgetting T[i] = 0, i \in \mathcal{D}
while not training done do
    B \sim \mathcal{D} # sample a minibatch
    for example $i \in B$ do
        compute $\text{acc}_i$
        if $\text{prev}_\text{acc}_i < \text{acc}_i$ then
            $T[i] = T[i] + 1$
            $\text{prev}_\text{acc}_i = \text{acc}_i$
        end
    end
    gradient update classifier on $B$
return $T$
```
Figure 1: Histograms of forgetting events on (from left to right) MNIST, permutedMNIST and CIFAR-10. Rescaled view where the number of examples have been capped between 0 and 2000. The full view is available in Supplementary Figure 8.

Figure 2: Left 2D-histogram of the number of forgetting events and mean misclassification margin across all examples of CIFAR-10. There is significant negative correlation (-0.74, Spearman rank correlation) between mean misclassification margin and the number of forgetting events. Middle and Right Pictures of unforgettable (Middle) and forgettable examples (Right). Forgettable examples seem to exhibit peculiar or uncommon features.

5 seeds, there are 55,012, 45,181 and 15,628 unforgettable examples, which represent respectively 91.7%, 75.3%, and 31.3% of the corresponding training sets. Note that datasets with less complexity and diversity of examples, such as MNIST, seem to contain significantly more unforgettable examples. permutedMNIST exhibits a complexity balanced between MNIST (easiest) and CIFAR-10 (hardest). This finding seems to suggest a correlation between forgetting statistics and the intrinsic dimension of the learning problem, as recently formalized by [Li et al., 2018].

First learning events We investigate whether unforgettable and forgettable examples need to be presented different numbers of times in order to be learnt for the first time (i.e. for the first learning event to occur, as defined in Sec 3). The distributions of the presentation numbers at which first learning events occur across all datasets can be seen in Supplementary Fig. 9. We observe that, while both unforgettable and forgettable sets contain many examples that are learnt during the first 3-4 presentations, the forgettable examples contain a larger number of examples that are first learnt later in training. The Spearman rank correlation between the first learning event presentations and the number of forgetting events across all training examples is 0.56, indicating a moderate relationship.

Misclassification margin The definition of forgetting events is binary and as such fairly crude compared to more sophisticated estimators of example relevance [Zhao & Zhang, 2015, Chang et al., 2017]. In order to qualify its validity, we compute the misclassification margin of forgetting events. The misclassification margin of an example is defined as the mean classification margin (defined in Sec 3) over all its forgetting events, a negative quantity by definition. The Spearman rank correlation between forgetting events and misclassification margin is -0.74 (computed over 5 seeds). Examples which are frequently forgotten have a large misclassification margin.

Visual inspection We visualize some of the unforgettable examples in Figure 2 along with some examples that have been most forgotten in the CIFAR-10 dataset. Unforgettable samples are easily recognizable and contain the most obvious class attributes or centered objects, e.g. a plane on a
clear sky. On the other hand, the most forgotten examples exhibit more ambiguous characteristics (as in the center image, a truck on a brown background) that may not align with the learning signal common to other examples from the same class.

Detection of noisy examples We further investigate the observation that the most forgettable examples seem to exhibit atypical characteristics. We would expect that if highly forgettable examples have atypical class characteristics, then noisily-labeled examples will undergo more forgetting events. We randomly change the labels of 20% of CIFAR-10 and record the number of forgetting events of both the noisy and regular examples through training. The distributions of forgetting events across noisy and regular examples are shown in Figure 3. We observe that the most forgotten examples are those with noisy labels and that no noisy examples are unforgettable. We also compare the forgetting events of the noisy examples to that of the same set of examples with original labels and observe a much higher degree of forgetting in the noisy case. The results of these synthetic experiments support the hypothesis that highly forgettable examples exhibit atypical class characteristics.

4.2 Continual Learning Setup

We observed that in harder tasks such as CIFAR-10, a significant portion of examples are forgotten at least once during learning. This leads us to believe that catastrophic forgetting may be observed, to some extent, even when considering examples coming from the same task distribution. To test this hypothesis, we perform an experiment inspired by the standard continual learning setup (McCloskey & Cohen, 1989; Kirkpatrick et al., 2017). We create two tasks by randomly sampling 10k examples from the CIFAR-10 training set and dividing them in two equally-sized partitions (5k examples each). We then train a classifier for 20 epochs on each partition sequentially, while tracking performance on both partitions. The results are reported in Figure 4 (a). We observe some forgetting of the second task when we only train on the first task (panel (a.2)). This is somewhat surprising as the two tasks contain examples from the same underlying distribution.

In Figure 4 (b), we report results obtained by randomly sampling examples based on forgetting statistics. That is, we first compute the forgetting events based on Algorithm 1 and we create our tasks by sampling 5k examples that have zero forgetting events (named f0) and 5k examples that have non-zero forgetting events (named fN). We observe that examples that have been forgotten at least once suffer a more drastic form of forgetting than those included in a random split (compare (a.2) with (b.2)). In panel (b.3), we can observe that examples from task f0 suffer very mild forgetting when training on task fN. This suggests that examples that have been forgotten at least once may be able to “support” those that have never been forgotten.
5 REMOVING UNFORGETTABLE EXAMPLES

As shown in the previous section, learning on examples that have been forgotten at least once minimally impacts performance on those that are unforgettable. This appears to indicate that unforgettable examples are less informative than others, and, more generally, that the more an example is forgotten during training, the more useful it may be to the classification task. This seems to align with the observations done in Chang et al. (2017), where the authors re-weight training examples by accounting for the variance of their predictive distribution. Here, we test whether it is possible to completely remove a given subset of examples during training.

In Fig. 5 (Left), we show the evolution of the generalization performance in CIFAR-10 when we artificially remove examples from the training dataset. We choose the examples to remove by increasing number of forgetting events. Each point in the figure corresponds to retraining the model from scratch on an increasingly smaller subset of the training data (with the same hyper-parameters as the base model). We observe that when removing a random subset of the dataset, performance rapidly decreases. Comparatively, by removing examples ordered by number of forgetting events, 30% of the dataset can be removed while maintaining comparable generalization performance as the base model trained on the full dataset, and up to 35% can be removed with marginal degradation (less than 0.2%). The results on the other datasets are similar, a large fraction of training examples can be ignored without hurting the final generalization performance of the classifiers (Figure 6).

In Figure 5 (Right), we show the evolution of the generalization error when we remove from the dataset 5,000 examples with increasing forgetting statistics. Each point in the figure corresponds to the generalization error of a model trained on the full dataset minus 5,000 examples as a function of the average number of forgetting events in those 5,000 examples. As can be seen, removing the same number of examples with increasingly more forgetting events results in worse generalization for most of the curve. It is interesting to notice the rightmost part of the curve moving up, this suggests that some of the most forgotten examples actually hurt performance. Those can correspond to outliers or mislabeled examples (also see Sec. 4.1). Finding a way to separate those points from very informative ones is an ancient but still active area of research (John, 1995; Jiang et al., 2018).

Support vectors Various explanations of the implicit generalization of deep neural networks (Zhang et al., 2016) have been offered: flat minima generalize better and stochastic gradient descent converges towards them (Hochreiter & Schmidhuber, 1997; Kleinberg et al., 2018), gradient descent protects against overfitting (Advani & Saxe, 2017; Tachet et al., 2018), deep networks’ structure biases learning towards simple functions (Neyshabur et al., 2014; Perez et al., 2018). But it remains a poorly understood phenomenon. An interesting direction of research is to study the convergence properties of gradient descent in terms of maximum margin classifiers. It has been shown recently (Soudry et al., 2017) that on separable data, a linear network will learn such a maximum margin classifier. This supports the idea that stochastic gradient descent implicitly converges to solutions that maximally separate the dataset, and additionally, that some data points are more relevant than

Figure 4: The synthetic continual learning setup for CIFAR-10 highlights that examples that have been forgotten at least once can “support” those that have never been forgotten.
Intrinsic dataset dimension As mentioned above, the datasets we study have various fractions of unforgettable events (91.7% for MNIST, 75.3% for permutedMNIST and 31.3% for CIFAR-10). We also see in Figure 6 that performance on those datasets starts to degrade at different fractions of removed examples, the number of support vectors varies from one dataset to the other, based on the complexity of the underlying data distribution. This last statement can be put in perspective with the intrinsic dataset dimension defined by Li et al. (2018) as the codimension in the parameter space of the solution set: for a given architecture, the higher the intrinsic dataset dimension, the larger the number of support vectors, and the fewer the number of unforgettable examples.
6 Transferable Forgetting Events

Our definition of forgetting events is based on training a given architecture, with a given optimizer for a given number of time steps. We investigate to what extent the forgetting statistics of examples depend on those factors.

Across seeds To test the stability of our metric with respect to the variance generated by stochastic gradient descent, we compute the number of forgetting events per example for 10 different random seeds and measure their correlation. From one seed to another, the average Pearson correlation is 89.2%. When randomly splitting the 10 different seeds into two sets of 5, the cumulated number of forgetting events within those two sets shows a high correlation of 97.6%.

Throughout training We compute the Spearman rank correlation between the ordering obtained at the end of training (200 epochs) and the ordering after various number of epochs. As seen in Fig. 7 (Left), the ordering is very stable after 75 epochs, and we found a reasonable number of epochs to get a good correlation to be 25 (see the Supp. Mat. for precision-recall plots).

Between architectures A limitation of our method is that it requires computing the ordering from a previous run. An interesting question is whether that ordering could be obtained from a simpler architecture than residual networks. We train a network with two convolutional layers followed by two fully connected ones (see Supp. Mat. for the full architecture) and compare the resulting ordering with the one obtained with the ResNet18. Figure 7 (Middle) shows a precision-recall plot of the unforgettable examples computed with the residual network. We see a reasonably strong agreement between the unforgettable examples of the convolutional neural network and the ones of the ResNet18. Finally, we train a WideResNet (Zagoruyko & Komodakis, 2016) on truncated data sets (using the example ordering from ResNet18) and plot its generalization performance in Figure 7 (Right): the network still performs near optimally with 30% of the dataset removed. This opens up promising avenues where forgetting statistics could be computed by smaller architectures.

7 Conclusion and Future Work

In this paper, inspired by the phenomenon of catastrophic forgetting, we investigate the learning dynamics of neural networks when training on single classification tasks. We show that catastrophic forgetting can occur in the context of what is usually considered to be a single task. Inspired by this result, we find that some examples within a task are more prone to being forgotten, while others are consistently unforgettable. We also find that forgetting statistics seem to be fairly stable with respect to the various characteristics of training, hinting that they actually uncover intrinsic properties of the data rather than idiosyncrasies of the training schemes. Furthermore, these unforgettable examples seem to play little part in the final performance of the classifier as they can be removed from the training set without hurting generalization. This supports recent research interpreting deep neural networks as max margin classifiers in the linear case.

Future work involves understanding forgetting events better from a theoretical perspective, exploring potential applications to other areas of supervised learning, such as speech or text and to reinforcement learning where forgetting is prevalent due to the continual shift of the underlying distribution.
REFERENCES


Under review as a conference paper at ICLR 2019


8 EXPERIMENTATION DETAILS

Detailed distributions

Figure 8: Full view of forgetting event counts on (from left to right) MNIST, permutedMNIST and CIFAR-10.

Figure 9: From left to right, distributions of the first presentation at which each unforgettable and forgettable example was learned in MNIST, permutedMNIST and CIFAR-10 respectively. Rescaled view where the number of examples have been capped between 0 and 1500 for visualization purposes. Unforgettable examples are generally learnt early during training, thus may be considered as “easy” in the sense of Kumar et al. (2010), i.e. may have a low loss during most of the training.

permutedMNIST The permutedMNIST data set is obtained by applying a fixed random permutation of the pixels to all the images of the standard MNIST data set. This typically makes the data set harder to learn for convolutional neural networks as local patterns, e.g. the horizontal bar of the 7, get shuffled. This statement is supported by the two following facts:

- The number of unforgettable examples for permutedMNIST is 45181 versus 55012 for MNIST.
- The intrinsic data set dimension (Li et al., 2018) of permutedMNIST is 1400 compared to 290 for the untouched data set.

Network Architectures

We use a variety of different architectures in the main text. Below are their specifications.

The architecture for the MNIST and permutedMNIST experiments is the following:

1. a first convolutional layer with 5 by 5 filters and 10 feature maps,
2. a second convolutional layer with 5 by 5 filters and 20 feature maps,
3. a fully connected layer with 50 hidden units
4. the output layer, with 10 logits, one for each class.

We apply ReLU nonlinearities to the feature maps and to the hidden layer. The last layer is passed through a softmax to output probabilities for each class of the data set.

The ResNet18 architecture used for CIFAR-10 is described thoroughly in DeVries & Taylor (2017), its implementation can be found at https://github.com/uoguelph-mlrg/Cutout.

The second one is a WideResNet (Zagoruyko & Komodakis, 2016), with a depth of 28 and a widen factor of 10. We used the implementation found at https://github.com/meliketoy/wide-resnet.pytorch.
The convolutional architecture used in Section 6 is the following:

1. a first convolutional layer with 5 by 5 filters and 6 feature maps,
2. a 2 by 2 max pooling layer
3. a second convolutional layer with 5 by 5 filters and 16 feature maps,
4. a first fully connected layer with 120 hidden units
5. a second fully connected layer with 84 hidden units
6. the output layer, with 10 logits, one for each class.

Optimization

The MNIST networks are trained to minimize the cross-entropy loss using stochastic gradient descent with a learning rate of 0.01 and a momentum of 0.5.

The ResNet18 is trained using cutout, data augmentation and stochastic gradient descent with a 0.9 Nesterov momentum and a learning rate starting at 0.1 and divided by 5 at epochs 60, 120 and 160.

The WideResNet is trained using Adam (Kingma & Ba, 2014) and a learning rate of 0.001.

9 Noising THE DATA SETS

Figure 10: Distribution of forgetting events across all training examples in MNIST when 20% of labels are randomly changed. **Right.** The number of examples are clipped to a maximum of 1000 for the plot so the distribution is visible. Half of the noisy examples are never learned (have 200 forgetting events). 2% of noisy examples are unforgettable.

10 Stability of the FORGETTING EVENTS

In this section, we plot precision-recall diagrams for the unforgettable and most forgotten examples of CIFAR-10 obtained on ResNet18 after 200 epochs and various prior time steps. We see in particular that at 75 epochs, the examples on both side of the spectrum can be retrieved with very high precision and recall.
Figure 11: Right: precision and recall of retrieving the unforgettable examples from a full run of ResNet18 (200 epochs), using the example ordering after 25, 50, and 75 epochs. The unforgettable examples are retrieved with high precision and recall after 50 epochs.