XAI-INCREMENT: A NOVEL APPROACH LEVERAGING LIME EXPLANATIONS FOR IMPROVED INCREMENTAL LEARNING

Arnab Neelim Mazumder*,†, Niall Lyons†, Anand Dubey†, Ashutosh Pandey†, Avik Santra†

* University of Maryland Baltimore County, Baltimore, MD, USA
†Infineon Technologies, Irvine, CA, USA

ABSTRACT

Explainability of neural network prediction is essential to understand feature importance and gain interpretable insight into neural network performance. In this work, model explanations are fed back to the feed-forward training to help the model generalize better. To this extent, a custom weighted loss where the weights are generated by considering the Euclidean distances between true LIME (Local Interpretable Model-Agnostic Explanations) explanations and model-predicted LIME explanations is proposed. Also, in practical training scenarios, developing a solution that can help the model learn sequentially without losing information on previous data distribution is imperative due to the unavailability of all the training data at once. Thus, the framework known as XAI-Increment incorporates the custom weighted loss developed with elastic weight consolidation (EWC), to maintain performance in sequential testing sets. Finally, the training procedure involving the custom weighted loss shows around 1% accuracy improvement compared to the traditional loss based training for the keyword spotting task on the Google Speech Commands dataset and also shows low loss of information when coupled with EWC in the incremental learning setup.

Index Terms— keyword spotting, incremental learning, LIME explanations, elastic weight consolidation, weighted loss, explainable methods

1. INTRODUCTION

Human-Machine interface via voice has become omnipresent in nowadays society. A distinctive feature of voice assistants is that, in order to be used, they first have to be activated by means of a spoken keyword spotting (KWS), thereby avoiding computational expenses when it is not required. Thus, KWS can be defined as the task of identification of keywords in audio streams comprising speech. The earliest approaches are based on continuous KWS. One of the advantages of this approach is the flexibility to deal with changing/non-predefined keywords. Therefore it is usually desired to have algorithms to do incremental learning (IL), commonly referred as continual learning [1,2]. In a continual learning setup, a continuously learning agent at a time step $t$ is trained to recognize the tasks $1, \ldots, t-1, t$ while the data $D_1, D_2, \ldots, D_{t-1}$, for the tasks $1, \ldots, t-1$ may or may not be available. Such a learning paradigm has two fundamental trade-offs (i.e., knowledge transfer (KT) and semantic transfer (ST)) to overcome. A positive KT suggests that the agent should deliver better accuracy on the task $t$ if allowed to learn it incrementally through tasks $1, \ldots, t-1$ while achieving a low validation error on all of these tasks. On the other hand, a positive ST means that learning a new task $t$ would increase the model’s performance on the previously learned tasks $1, \ldots, t-1$. To find the trade-off between KT and ST, multiple methods are proposed in the literature (architecture-based [3], memory-based [4], etc.). In contrast to these methods, which are computationally extensive, the regularization method (in this case, elastic weight consolidation) typically assumes a fixed network size and learns a new task while trying to avoid changes to parameters sensitive to previous tasks. In this work, a regularization-based elastic weight consolidation (EWC) approach is utilized for continual learning where parameter $\theta_{t-1}^*$ configuration is achieved at the end of the dataset $i$, which is expected to solve all the datasets $D_{i+1}$. Such an objective can be minimized by adding a regularization loss, which prevents $\theta_{t-1}^*$ from veering too far away from $\theta_{t-1}^*$. Since this regularization loss should preserve closeness to the previous solution, the KL-divergence between $p(\theta | D_{i+1})$ and $p(\theta | D_{i+1})$ as the regularization loss is used. In practice, EWC proposes using the second-order approximation of this KL-divergence:

$$KL(p(\theta | D_{i+1}) || p(\theta | D_{i+1})) \approx \frac{1}{2} \sum_j F_{jj} (\theta_j - \theta_{t-1,j}^*)^2$$

(1)

Here, $F$ refers to the empirical Fisher matrix, only the diagonal of which is used in the approximation.

As the network is optimized using maximum likelihood estimation, the semantic transfer highly depends on the examples used during training and their similarity coefficient. That is, dataset components with less similarity suffer more semantic loss. To avoid data-dependent optimization, LIME (Local Interpretable Model-Agnostic Explanations) [6,7] based continual learning where the important semantics are learned using weighted LIME scores in combination with EWC is proposed. The contributions of this paper are:

• A novel and general framework where EWC regularization is combined with model explainability to enhance classification performance of any neural network is proposed.

• Further, LIME scores of miss-classified samples from the previous task are used as a weighting factor during model optimization to have better semantic transfer learning between tasks in an IL setting and to build more generalized models.

Section 2 details the proposed approach of LIME-based weighted loss IL with Section 3 describing the experimental setup and elaborations of the experiments in Section 4.

2. BACKGROUND & PROPOSED FRAMEWORK

The generalizability of much XAI published research is problematic and leads to a design-by-data approach [6]. Moreover, XAI is very scarcely used in the context of IL, and minimal research has been conducted on the use of incorporating XAI and IL to create robust, reliable, non-human-in-the-loop AI systems. The proposed solution
utilizes recent XAI approaches introduced in [6,9] and incorporates them into an IL framework.

The XAI-Increment framework shown in Figure 1 aims to bridge this gap and provide a training methodology to enhance the classification accuracy for the KWS task in-midst of an explainable pipeline that can augment IL for the same task. The situation where adding new data to the training regime negatively impacts the learned distribution is known as catastrophic forgetting. In order to prevent this, [5] proposed EWC, which forces the model to retain previous information on top of adding new data. However, although EWC acts as a regularizer to prevent catastrophic forgetting, it also limits the model to learn information from the new data. To address this, we propose using a weighted loss during model retraining where the weights from the samples come from the difference between LIME visuals of the true and predicted classes. As a result, the model will focus more on rectifying the incorrect predictions with higher weights, allowing the network to learn the new data better.

2.1. LIME Visualizations and Feature Scores

Algorithm 1: Feature Score Generation Procedure with LIME

Require: \( X_{\text{in}}, \theta \)

\[ C = \text{Cluster}(X_{\text{in}}) \]
\[ \text{var} \leftarrow [\text{null}], \sigma = 0.25 \]
\[ \text{for } i \in [1, 2, \ldots, n] \text{ do} \]
\[ V = \text{Perturbation}(C) \]
\[ \text{var}.\text{append}(V) \]
\[ \text{end for} \]
\[ \text{pred} = \text{Predict}(\theta^*, \text{var}) \]
\[ \text{dist} = \text{Cosine Distance}(X_{\text{in}}, \text{var}) \]
\[ \text{wt} = \sqrt{\frac{\text{dist}^2}{\sigma^2 + \text{dist}^2}} \]
\[ \text{reg} = \text{Linear Regression}(\text{var}, \text{pred}, \text{wt}) \]
\[ \text{score} = \text{coeff}(\text{reg}) \]

Saliency map-based methods like GradCAM [7], ScoreCAM [10], etc., can provide explainable visuals of the heatmap overlayed on top of the original image according to either weight or gradient activation. However, in the context of enhancing the accuracy of the KWS task, determining the importance of segments within a spectrogram to isolate activity regions is highly important. To this end, the LIME-based visual explanations [6] to generate sample weight for the weighted loss for the weighted loss is adopted in this work and detailed in Algorithm 2. Initially, the input is segmented with the slic [11] clustering algorithm. The clustered input then goes through a perturbation process where the different segments are turned on or off according to the binomial distribution. Each instance of the clustered input thus generated is referred to as a variation. Next, the trained model (for which the explanations are generated) is used to predict the classes of all these variations. Finally, the scores are created for the segments by fitting a linear regression classifier on the variations and their corresponding predictions, where the cosine distance between the variations and the original input act as weights. Since LIME explanations are based on segmentations created on the spectrogram, it tries to fit a linear classifier for generating an importance score for the segment itself. As a result, LIME provides a qualitative and a quantitative metric to explain the model prediction on an example as shown in Figure 2(A) and (B).

Fig. 1: A high-level overview of the proposed training (XAI-Increment) methodology that integrates explainable weight generation with EWC regularization to maintain model performance in sequential testing sets.

Fig. 2: A qualitative representation of LIME explanations for an accurate classification (A) and a misclassification (B). (A) illustrates the five most influential slic clusters (greyed out clusters) for making the prediction ‘yes’ which coincides with the LIME explanation for the true class. (B) demonstrates the two most influential slic clusters for making the prediction ‘no’ which is different compared to the important slic clusters of the true class ‘go’.

2.2. Weighted Loss

In a traditional training setup, all samples are provided a similar influence/weight during loss generation. However, some samples are more difficult to predict than others. More importantly, in an IL setting, it is imperative to learn the incoming data rigorously while preserving the already learned information from the distribution. In this context, using a weighted loss function will force the model to prioritize learning the new samples in back-propagation during the training. For example, if a sample is \( X_i \) with the corresponding model to be \( \theta(x_i, \theta) \) with \( \theta \) parameter weights, and the sample loss to be \( L(y_i, y) \) then the weighted loss function for \( N \) batches in a feed-forward network is given by

\[
\text{Loss} = \frac{1}{N} \sum_{i=1}^{N} w_i L(y_i, y) \tag{2}
\]
The straightforward way to generate these above-mentioned sample weights would be to assign each incorrectly predicted sample with a high value. However, that process might not represent which samples need more focus than others. To address this, sample weights are generated from the explainable LIME visuals. The weights are considered to be the Euclidean distance between the LIME explanations for the true class ($E_i$) and the predicted class ($E_p$) as given by the following:

$$w = \sum_{i=1}^{n} (E_p^i - E_t^i)^2$$  \hspace{1cm} (3)

Here, $n$ denotes the number of segments. The choice behind using Euclidean distance is further justified through Table 1 where it is evident that Euclidean distance-based LIME weights outperform Manhattan and Cosine distance-based LIME weights in terms of classification accuracy.

| Distance Metric | Accuracy (%) |
|-----------------|--------------|
| Euclidean       | 90.92 ±0.23  |
| Manhattan       | 90.69 ±0.28  |
| Cosine          | 90.65 ±0.24  |

### 2.3. Elastic Weight Consolidation (EWC)

To address catastrophic forgetting in an IL setting, it is imperative to have some regularization (usually L1 or L2) during training to maintain information from the previous distributions. However, as demonstrated in [5], L1 or L2 normalization most often constrains each weight with the same coefficient, meaning the model can only remember the previous task at the expense of not learning the new one. EWC provides an alternative to this predicament where learning is slowed down on certain weights based on their importance to the previous distribution with a quadratic penalty on the loss, as shown in Equation (4). This scenario further enhances the learning of both tasks and maintains the model performance on each separately. For example, if the loss for the current task is $L_{curr}$, then EWC regularization is given by the following:

$$\text{Loss} = L_{curr} + \frac{\lambda}{2} \sum_i F_i (\theta_i - \theta_{A,i}^*)^2$$  \hspace{1cm} (4)

Here, $\theta_{A,i}$ are the parameters from the previous task, and $\theta_i$ are the parameters from the current task. $F_i$ represents the previous task’s Fisher information matrix (FIM) parameters. $\lambda$ controls the amount of EWC regularization applied to the current loss.

### 3. EXPERIMENTAL SETUP

The dataset used for the experiments is Google Speech Commands [12]. We start by making all the audios have 16,000 samples with zero padding. Next, we create spectrograms of these files, which significantly reduces the compute requirement for the later use of deep neural networks (DNNs). Finally, the spectrograms enable us to create LIME explanations and use that insight as feedback to complement the learning process. A VGG-like framework is utilized with only four blocks and fewer filters in all the layers than the actual VGG architecture. We denote this as this work’s adapted VGG architecture (shown in Equation 5). Here $2 \times (8 \text{ Conv}_3 \times 3)$ means 2 sequential convolutional layers with 8 filters and $3 \times 3$ kernel shape, $32 \text{ pool}_{2 \times 2}$ means a maxpooling layer of $2 \times 2$ pooling window with 32 filters, and 1000 $FC$ means a fully-connected layer with 1000 neurons respectively.

\[(\text{in}) - 2 \times (8 \text{ Conv}_3 \times 3) - 2 \times (16 \text{ Conv}_3 \times 3) - 3 \times (32 \text{ Conv}_3 \times 3) - (32 \text{ pool}_{2 \times 2}) - 3 \times (64 \text{ Conv}_3 \times 3) - (64 \text{ pool}_{2 \times 2}) - (1000 \text{ FC}) - (\text{out})\]

(5)

The overall training procedure is illustrated in Figure 1 and described through Algorithm 2. To accommodate retraining in sequential sessions, a training, validation, and test split of 80%, 10%, and 10%, respectively, is created. Here, the data splits are based on speaker information. All utterances from a specific speaker are used in a particular set only, with no leakage of the samples in other sets. Only during the sequential retraining are the incorrectly predicted samples added to the training set. The weight of the incorrectly predicted samples is generated by taking the Euclidean distance between the LIME explanation of the predicted class and the correctly predicted samples is generated by taking the Euclidean distance between the true class and the predicted class.

#### Algorithm 2: Incremental Training Procedure with LIME-based Weighted Loss and EWC.

**Require:** $D^m$, $D^n$, $D^f$

**Initial Training:**
1: Initialize model $\theta^0$
2: $\theta^0 = \text{train} (D^m, \theta^0, D^n)$
3: $\text{Acc}^m = \text{eval} (\theta^0, D^f)$

**Generate LIME weights**
4: for $D^n_i$ in [$D^{n_1}, D^{n_2}, ..., D^{n_{\text{d}}}$] do
5: $D_i = \{(d, y) \in D^n_i | \text{predicted}(\theta^0, d) \neq y\}$
6: $\text{LIME}_{\text{pred}}^i = \text{lime}(\theta^0, D_i)$
7: $\text{LIME}_{\text{true}}^i = \text{lime}(\theta^0, D_i)$
8: $w^i = \text{Lwf}(\text{LIME}_{\text{pred}}^i, \text{LIME}_{\text{true}}^i)$
9: end for

**Incremental Training:**
10: Initialize $D = D^m$, $\theta^0 = \theta^m$, $w = 1$
11: for $D_i$ in [$D^{i_1}, D^{i_2}, ..., D^{i_{\text{d}}}$] do
12: Add new data: $D_{\text{new}} = D \cup D_i$, $w = w + w^i$
13: $\text{FIM} = \text{gen}_{\text{fim}}(\theta^{i-1}, w \% \text{of } D_{\text{new}})$
14: $\theta^i = \text{EWC}_{\text{fim}}(\theta^{i-1}, D_{\text{new}}, w, D^{vi}, \text{FIM})$
15: $\text{Acc}^i = \text{eval}(\theta^i, D^f)$
16: end for

### 4. RESULTS & DISCUSSION

We perform six separate runs on the complete dataset for weighted loss training and regular loss training to create an analogy between...
the two methods. The weighted loss method results in 90.9% Top-1 accuracy, which is around 1% better than regular loss-based training (90.2%). Figure 3 shows the performance of the first ten classes during testing in the form of confusion matrices. The weighted loss training method adopted in this work performs better and gets more predictions right in most classes than traditional loss-based training, proving that our custom loss enhances performance. To further accommodate EWC according to Equation 4, we aim to choose the optimal parameter (\(\lambda\)), which controls the influence of EWC regularization while generating the loss. A higher \(\lambda\) value allows aggressive learning of the newer task with the caveat of forgetting the older ones. Hence we experiment with six identical EWC and weighted loss-based IL setups for six different \(\lambda\) values. As per the line graph shown in Figure 3 it is apparent that \(\lambda = 1\) provides us with the most stable performance in terms of test set accuracy throughout all six sessions of IL.

![Fig. 3](image)

---

**Table 2:** Standard Error Comparison among traditional loss, weighted loss and weighted loss with EWC for six stage incremental learning over six separate runs

| # of Sess. | Trad. Loss | W_Loss | W_Loss + EWC |
|-----------|------------|--------|--------------|
| 1         | 89.90 ±0.27| 90.93 ±0.86| 90.06 ±0.20 |
| 2         | 90.60 ±0.24| 90.51 ±0.48| 90.79 ±0.22 |
| 3         | 90.48 ±0.23| 90.58 ±0.22| 90.56 ±0.33 |
| 4         | 90.60 ±0.65| 90.70 ±0.48| 90.51 ±0.18 |
| 5         | 90.64 ±0.29| 90.68 ±0.40| 90.88 ±0.46 |
| 6         | 90.91 ±0.53| 91.34 ±0.20| 91.45 ±0.23 |

Using explainable insights during learning can significantly improve the performance of DNNs. To that end, this work looks to take advantage of the LIME visuals of a model prediction. By differentiating the explanations for actual and predicted classes, proportional weights are generated for false predictions and used during the consequent IL stage retraining. On top of it, the enhancement brought upon by EWC regularization to maintain neural network performance across IL scenarios is explored. The experiments suggest that the weighted loss training coupled with EWC reaches an accuracy level of 91.45%, which is close to a 0.5% improvement over the incremental training with the traditional sparse categorical cross-entropy loss (90.91%). Our future work will expand the scope of this methodology to include weights from explainable visuals in the context of metric learning to further bolster the classification network’s performance.

5. CONCLUSION

Using explainable insights during learning can significantly improve the performance of DNNs. To that end, this work looks to take advantage of the LIME visuals of a model prediction. By differentiating the explanations for actual and predicted classes, proportional weights are generated for false predictions and used during the consequent IL stage retraining. On top of it, the enhancement brought upon by EWC regularization to maintain neural network performance across IL scenarios is explored. The experiments suggest that the weighted loss training coupled with EWC reaches an accuracy level of 91.45%, which is close to a 0.5% improvement over the incremental training with the traditional sparse categorical cross-entropy loss (90.91%). Our future work will expand the scope of this methodology to include weights from explainable visuals in the context of metric learning to further bolster the classification network’s performance.

6. REFERENCES

[1] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, “Learning to learn without forgetting by maximizing transfer and minimizing interference,” 2018. [Online]. Available: [https://arxiv.org/abs/1810.11910](https://arxiv.org/abs/1810.11910)
[2] S. Jha, M. Schiemer, and J. Ye, “Continual learning in human activity recognition: an empirical analysis of regularization,” 2020. [Online]. Available: https://arxiv.org/abs/2007.03032

[3] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” 2016. [Online]. Available: https://arxiv.org/abs/1606.04671

[4] J. Knoblauch, H. Husain, and T. Diethe, “Optimal continual learning has perfect memory and is np-hard,” 2020. [Online]. Available: https://arxiv.org/abs/2006.05188

[5] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska et al., “Overcoming catastrophic forgetting in neural networks,” Proceedings of the national academy of sciences, vol. 114, no. 13, pp. 3521–3526, 2017.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, “" why should i trust you?" explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.

[7] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-cam: Why did you say that?” arXiv preprint arXiv:1611.07450, 2016.

[8] G. Vilone and L. Longo, “Explainable artificial intelligence: a systematic review,” arXiv preprint arXiv:2006.00093, 2020.

[9] H. Sun, L. Servadei, H. Feng, M. Stephan, R. Wille, and A. Santra, “Utilizing explainable ai for improving the performance of neural networks,” 2022. [Online]. Available: https://arxiv.org/abs/2210.04686

[10] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, “Score-cam: Score-weighted visual explanations for convolutional neural networks,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 24–25.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels,” Tech. Rep., 2010.

[12] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” arXiv preprint arXiv:1804.03209, 2018.