Explain to Not Forget: Defending Against Catastrophic Forgetting with XAI

Sami Ede\textsuperscript{1,†}, Serop Baghdadlian\textsuperscript{1,†}, Leander Weber\textsuperscript{1}, An Nguyen\textsuperscript{2}, Dario Zanca\textsuperscript{2}, Wojciech Samek\textsuperscript{1,3,4,*}, and Sebastian Lapuschkin\textsuperscript{1,*}

\textsuperscript{1} Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany
\textsuperscript{2} Friedrich Alexander-Universität Erlangen-Nürnberg, 91052 Erlangen, Germany
\textsuperscript{3} Technische Universität Berlin, 10587 Berlin, Germany
\textsuperscript{4} BIFOLD – Berlin Institute for the Foundations of Learning and Data, 10587 Berlin, Germany

\textsuperscript{†} contributed equally

\textsuperscript{*} corresponding authors: \{sebastian.lapuschkin, wojciech.samek\}@hhi.fraunhofer.de

Abstract. The ability to continuously process and retain new information like we do naturally as humans is a feat that is highly sought after when training neural networks. Unfortunately, the traditional optimization algorithms often require large amounts of data available during training time and updates w.r.t. new data are difficult after the training process has been completed. In fact, when new data or tasks arise, previous progress may be lost as neural networks are prone to catastrophic forgetting. Catastrophic forgetting describes the phenomenon when a neural network completely forgets previous knowledge when given new information. We propose a novel training algorithm called Relevance-based Neural Freezing in which we leverage Layer-wise Relevance Propagation in order to retain the information a neural network has already learned in previous tasks when training on new data. The method is evaluated on a range of benchmark datasets as well as more complex data. Our method not only successfully retains the knowledge of old tasks within the neural networks but does so more resource-efficiently than other state-of-the-art solutions.

Keywords: Explainable AI · Layer-wise Relevance Propagation (LRP) · Neural Network Pruning · Catastrophic Forgetting

1 Introduction

While neural networks achieve extraordinary results in a wide range of applications, from the medical field to computer vision or successfully beating humans on a variety of games [37,41], the established training process typically relies on a large amount of data that is present at training time to learn a specific task. For example, the famous ImageNet dataset [7] consists of more than 14 million images which results in a size of more than 150 GB, while the authors of [30] collected a dataset of 400 million images that make up more than 10 TB of data [35]. Large amounts of samples can help models generalize better by avoiding overfitting in single examples, but in turn make model training extremely expensive. If more data is later added, and the model should be able to correctly predict on both new and old data, usually it has to be fine-tuned or trained from scratch with the expanded dataset — as opposed to only the new data — as otherwise, catastrophic forgetting can occur when learning multiple consecutive tasks or from non-stationary data. If more data is later added, and the model should be able to correctly predict on both new and old data, usually it has to be fine-tuned or trained from scratch with the expanded dataset as opposed to only the new data. Otherwise, catastrophic forgetting [14] can occur when learning multiple consecutive tasks or from non-stationary data. One prominent example of this is reinforcement learning, in which an agent continuously interacts with its environment, using a stream of observations as training data. As the observations change with the agents actions in the environment, the data distribution becomes non-i.i.d., leading to catastrophic forgetting in the agent that is usually countered with an “experience buffer”, in which earlier observations are saved and randomly repeated. Other applications would also benefit from solutions to continuous or lifelong learning, e.g., medical applications such as skin cancer detection, where more targets could be added after additional samples have been obtained.
The issue of catastrophic forgetting is especially pronounced when previous data is not accessible anymore or is proprietary, making retraining impossible.

Recently, techniques of Explainable Artificial Intelligence (XAI) [33] have been proposed which are able to identify the elements of a neural network model crucial for solving the problem the model has been optimized for. One such method is Layer-wise Relevance Propagation (LRP) [3], which assigns relevance scores to latent network structures through modified backpropagation. In the past, this information has been used to great success to efficiently reduce neural network complexity without sacrificing performance by [5,44].

In this paper, we are proposing Relevance-based Neural Freezing (RNF), a novel approach to alleviate catastrophic forgetting that builds upon the aforementioned pruning technique. Instead of compressing the network, the information about unit importance is used to freeze the knowledge represented by the learned network parameters by inhibiting or completely stopping the training process for those parts of the network that are relevant for a specific task, while the remaining units are free to learn further tasks. We evaluate our method on several commonly used datasets, i.e., MNIST [8], CIFAR10 and CIFAR100 [22], ImageNet [7], and the challenging Adience dataset for facial categorization [9], which is a dataset of photos shot under real-world conditions, meaning different variations in pose, lighting conditions and image quality.

2 Related Work

In this section we briefly review the theoretical background of explainable AI and catastrophic forgetting.

2.1 Explainable Artificial Intelligence

In recent years, XAI has gotten more and more attention as the urgency to understand how “black box” neural networks arrive at their predictions has become more apparent. Especially in applications that have far-reaching consequences for humans, like the prediction of cancer (e.g., [6,11,18]), it is not only important to know what the network predicted, but also why the decision was made. Generally, methods from XAI can be roughly divided into two categories:

Global explanations provide general knowledge about the model, its feature sensitivities and concept encodings. Some approaches rank the importance of specific features, concepts or data transformations (e.g., [15,16,19]) by analyzing the model’s reaction to an amount of real or synthetic data, while others try to assess their model by finding important neurons and their interactions [17] or finding the concepts encoded by hidden filters through synthesizing the input that maximizes their activation [10,26,27].

Instead of providing insight of the models general understanding of the data, local explanation methods aim at making individual model predictions interpretable, i.e., by ranking the importance of features with respect to specific samples. By attributing importance scores to the input variables, these explanations can be illustrated as heatmaps with the same dimensions as the input space. Among the local explanation methods, there are again multiple approaches, some still treating the model as a “black box”, approximating the local explanations via separately trained proxy models [31,45] or otherwise applying perturbation or occlusion techniques [13,47]. Other methods use (augmented) backpropagation in order to compute the importance ranking of the input or latent features, such as [3,4,25,39]. Our proposed method leverages the advantages of Layer-wise Relevance Propagation [3] as the method’s ability to measure the per-prediction usefulness and involvement of network elements has recently shown great success [5,44].

2.2 Catastrophic Forgetting

Unlike humans or animals, neural networks do not have the inherent ability to retain previously attained knowledge when they are presented with new information while being optimized. This effect is characterized by a drastic performance decrease on tasks trained earlier when progressing
on a new task or dataset. This phenomenon is described by the term catastrophic forgetting [14]. As neural networks are generally assumed to train on i.i.d. data, adding a new task to be learned can violate this assumption, causing the gradient updates to override the weights that have been learned for the previous tasks and causing the aforementioned loss of old knowledge. One way to combat catastrophic forgetting is experience replay [23], in which the old data is interspersed with the new data, simulating i.i.d. data such that the network retains the old knowledge. However, this approach is inefficient, does not allow online-learning and may even be impossible if access to the old data is not available. Therefore, numerous approaches have been proposed to tackle this problem more efficiently. The approach of [42] learns masks defining subnetworks in untrained models that are responsible for a given task while [36] concurrently learn binary attention vectors to retain the knowledge obtained in previous tasks. Other approaches [20, 46] propose constraints on weight updates for neurons that have been identified as being pertinent for a previous task. Dynamically Expandable Networks [24] increase the network capacity when training a new task.

In this paper, we propose a training algorithm that — motivated by the successful XAI-based pruning method described in [44]— uses LRP in order to identify those neurons that are relevant for a given task. After finding the important neurons, they are given a lower elasticity for learning subsequent tasks, such that the network efficiently retains the knowledge from previous tasks, outperforming other state-of-the-art methods.

3 Relevance-based Neural Freezing

As a local attribution method, LRP has shown [32,44] to not only deliver accurate and interpretable explanations about the input variables, the conservatory property of the local distribution rules also allows to gain insights on the importance of individual latent neurons and their associated filters. Additionally, LRP is scalable w.r.t. network depth, easy to implement through existing frameworks (e.g., [1,2]), and efficient with a linear computational cost w.r.t. a backpropagation pass. It works by treating the prediction of the model \( f(x) \) w.r.t. a network output of interest as the total sum of importance, or relevance \( R \), that is then redistributed towards the input variables: After the forward pass, the layers of a classifier are reversely iterated, redistributing the relevance among its neurons proportionally to their contributions in the preceding forward pass. This redistribution process follows a conservatory constraint, meaning that the sum of relevance in each layer is equal to the total amount of relevance at the model head:

\[
 f(x) = \cdots = \sum_{d_i \in (l+1)} R_{d_i}^{(l+1)} = \sum_{d_j \in (l)} R_{d_j}^{(l)} = \cdots = \sum_{d_k \in (l_0)} R_{d_k}^{(l_0)},
 \]

where \( f(x) \) is model output and \( R_{d_i} \) is the relevance score of unit \( d \) in layer \( l \). Depending on the application and layer type, variously purposed propagation rules have been proposed. For example, the LRP\( \varepsilon \) rule [34] is defined as

\[
 R_{j-k} = \sum_k \frac{a_j w_{jk}}{\sum_{d_j} a_j w_{jk} + \varepsilon} R_k,
\]

with \( a_j \) being the input activation at neuron \( j \), \( w_{jk} \) being the learned weights between neurons \( j \) in the lower- and \( k \) in the upper layer and \( \varepsilon \) being a small positive term. This rule is typically used in upper layers to filter out weaker or contradictory values via the added \( \varepsilon \) in the denominator, which results in smoother, less noisy heatmaps. A discussion of other rules and their applications can be found in [34]. In this paper, we use the LRP\( \varepsilon \) rule for fully-connected layers and the LRP\( \varepsilon^+ \) rule for convolutional layers, as recommended in [21].

The proposed method aims to prevent catastrophic forgetting by decreasing the plasticity of neurons rated as important for a given, already optimized task. The general procedure is as follows: After training the model on the first task, the relevant units are identified by using LRP on a small, randomly sampled subset of the data, from here on called the reference dataset, similarly as in [44]. Until the model performance on the test set decreases by a certain set threshold, the units
with the lowest relevance (computed on the reference dataset) are repeatedly selected and then pruned by setting their outgoing connections to zero. Once the threshold is reached, the remaining units are assigned a lower learning rate for any subsequent tasks, as they were the most important for the current task. To completely freeze the units, the learning rate is set to zero, but it is also possible to just lower the elasticity to a fraction of the original learning rate. To continue training, the connections to the less relevant units are restored to their state before the pruning. This algorithm is outlined in Algorithm 1; an intuitive illustration can be found in Figure 1.

Algorithm 1 Relevance-based Neural Freezing

Require: untrained model net, reference data x_r, task specific training data x_t, pruning threshold t, pruning ratio r, task number N_t, learning rate lr, learning rate for relevant units lr_frozen, learning rate for irrelevant units lr_irrelevant, epochs N_e, network units, with unit ∈ neurons, filters.

for task in N_t do
  for epoch in N_e do
    ▷ train net on x_t using lr
  end for
  for all layer in net do
    for all units in layer do
      ▷ compute importance of units using LRP
    end for
  end for
  ▷ sort units in descending order w.r.t. their global importance to the task
  while t not reached do
    ▷ zero out r units from net where importance is minimal
  end while
  ▷ lower elasticity of relevant units for current task
  lr_relevant units ← lr_frozen
  lr_irrelevant units ← lr_irrelevant
  ▷ restore zeroed out connections to continue training.
end for

The stopping criterion for the unit selection can be determined freely and is not limited to network performance. For instance, it could also be the number of remaining free units in the network or even the energy consumption of the network when running inference. In the following experiments, the learning rate for all frozen units is set to zero.

4 Experiments

We show the effectiveness of the proposed technique on a number of increasingly difficult datasets and tasks. We start by using well-known toy and benchmark datasets, namely MNIST [8] and a combination of CIFAR10/CIFAR100 [22], to illustrate the method and its conceptual functionality before also showing effectiveness in larger benchmark and real-world datasets. We set the pruning threshold to 2%, which we found to be optimal during our experiments. The optimal value for this parameter can be determined using grid search. Using 2% results in enough free network capacity to learn the additional tasks while keeping the accuracy of the classifier as high as possible. Results on MNIST and CIFAR10/100 are averaged between 20 seeds, while results on the ImageNet and the Adience dataset are averaged over five seeds. In addition to the accuracy, we also evaluate the free capacity of the model as the percentage of unfrozen nodes after each task. Details on each experimental setup can be found in the Appendix A. In all experiments, new tasks are introduced with a task-incremental [Task-IL] setup [28]. In a task-incremental setup, the neural network is additionally informed about the task identity during training as well as during inference. Each task has its own (separate) head, while the rest of the neurons are shared among all tasks.
Fig. 1: Illustration of our method to identify relevant neurons and freeze their training progress after training on a task. (a) Forward pass through the network trained on all previous tasks with a reference image of the current task (left) and computation of unit importance using LRP (right). (b) Iteratively set connections of irrelevant neurons/units to zero until performance threshold is reached. (c) Freezing the knowledge about the current task by setting the learning rate of the remaining units to zero and training on the next task. The process is repeated after training each task.
4.1 MNIST-Split

The first series of experiments is performed on the popular MNIST benchmark dataset. The dataset is split up into five tasks, each task being the classification of two digits, e.g., the first task consists of the digits 0 and 1, task two contains the digits 2 and 3 and so on. The model is trained on the first tasked and then fine-tuned sequentially using a task-incremental setup, which we refer to as naive finetuning. We also establish the upper bound, in which the model is trained on the entire dataset as one task. Since the samples in the task-incremental setup have a binary label, all samples are grouped according to their label, which results in a binary classification task with the two classes \([0, 2, 4, 6, 8]\) and \([1, 3, 5, 7, 9]\) (each pair has a separate network head).

Figure 2a shows the effect of Relevance-based Neural Freezing on the MNIST-Split dataset. The mean test accuracy over all tasks is increased by about 4%, which is the most evident in the accuracy for both task one and task two compared to the baseline. Instead of a drop of 30% in accuracy, the model can still classify task one with an accuracy of almost 90% and retains an accuracy for task two of over 90%. The increase in baseline accuracy after task five can be attributed to a similarity between the shapes of digits in tasks one and five: both the digits 8 and 9 have rounded forms which makes it easier to distinguish between a 0 and a 1.

4.2 MNIST-Permuted

The MNIST-Permuted setup increases the complexity of the MNIST dataset by introducing random pixel permutations to the digits. It is commonly used \[12\] in a ten-task configuration such that the unpermuted dataset poses as the first task while the remaining nine tasks are made up of different permutations of the original digits. The permutations are the same for all classes but change between tasks.

Even though applying Relevance-based Neural Freezing slightly lowers the mean accuracy from 76.99% to 75% compared to the naive finetuning baseline and does not reach the 95.85% of the upper bound, Figure 2b shows that the average accuracy over all seen tasks is above the naive finetuning baseline during training. For a closer inspection of task performance, Figure 2c shows the individual task accuracy over the training. It can be seen that especially task one, two and three benefited the most from Relevance-based Neural Freezing, whereby the small capacity model did not suffice to successfully learn more tasks. Nevertheless, it suggests that parameters can be re-used for new tasks: even though the free capacity drops to below ten percent after the first two tasks, the model can still learn the remaining tasks with reasonable accuracy by an apparent re-utilization of the frozen filters that have been deemed relevant for the previous tasks.

4.3 CIFAR10-100

For this dataset, the networks’ first task is the entire CIFAR-10 dataset, after which five further tasks are trained sequentially, each containing 10 randomly selected classes from the CIFAR-100 dataset. We expand on this approach and instead split CIFAR-100 into ten tasks, each containing ten random classes. Using Relevance-based Neural Freezing, we achieve a 40% increase of mean test accuracy compared to the baseline. This effect is also displayed in Figure 3a. Even though the network capacity limit seems to be reached at task 5, as in previous experiments, the models ability to still learn various tasks suggests that the knowledge attained in the previous tasks is enough to facilitate the learning of the remaining classes due to the random assignment of classes to the individual tasks, which prevents a semantic bias towards the underlying concepts of the classes within the tasks. Again, the ability of the model to learn new tasks despite a limited residual free capacity signals a high amount of filter re-use of the already frozen filters.

Another, more complex experiment is performed by manually ordering the classes in semantic groups with seven superclasses, containing several subclasses each: Flowers and trees, animals, aquatic mammals and fish, random objects, small insects, nature scenes and vehicles. Like in the random grouping experiment, applying Relevance-based Neural Freezing gains almost 30% in accuracy compared to the default baseline, as can be seen in Figure 3b. Again, it can be seen that the
Fig. 2: Results on the MNIST dataset. The left Y-axis shows the mean test accuracy over all already seen tasks. The X-axis shows task progression. The right Y-axis shows the model’s free capacity. (a) shows the mean test accuracy progression for each task on the MNIST Split dataset. (b) shows the test accuracy progression for both lower-bound default method and Relevance-based Neural Freezing on the MNIST Permuted dataset. (c) shows the mean mean test accuracy progression over all previous tasks after introducing each new task for the MNIST Permuted dataset.

model’s free capacity is getting close to 0% after training the first four tasks, but the model is able to re-use previous abstractions from previous tasks to still learn the remaining tasks despite the difficulty of the semantic grouping. While the network is not able to achieve an initial accuracy for new tasks that is as high as the default baseline (as shown in Figure 3c), Relevance-based Neural Freezing can strongly mitigate the loss of accuracy that default finetuning displays when even more tasks are learned.

4.4 ImageNet Split

For the ImageNet split the default baseline displays significant catastrophic forgetting even after training each task for only 10 epochs, as is evident in Figure 4a. On this dataset, our method not only preserves the performance of the model on previous tasks but even leads to an increase in accuracy, which in turn results in an overall gain of 35.31% of accuracy over all tasks.

4.5 Adience

The Adience benchmark dataset of unfiltered faces is a dataset made up about 26,000 photos of human faces with binary gender- and eight different age group labels. The images are shot under real-world conditions, meaning different variations in pose, lighting conditions and image quality. We performed experiments on Adience in two scenarios:
Fig. 3: CIFAR100 is split into 10 tasks for the random split and 8 tasks for the semantic split. In the semantic split setup, each task contains a different number of semantically similar classes. The left Y-axis shows the mean test accuracy over all already seen tasks. The X-axis shows task progression. The right Y-axis shows the model’s free capacity. For (a) CIFAR100 was split randomly into 10 tasks each containing 10 classes. The plot shows the mean test accuracy progression over all previous tasks after introducing each new task. (b) shows the test accuracy progression for both lower-bound default method and Relevance-based Neural Freezing on the semantic split on CIFAR 100. (c) shows the mean test accuracy progression over all previous tasks after introducing each new task on CIFAR 100.

- **Split**: The dataset is split into two tasks, each consisting of a four-class classification problem of different age groups. The classes are grouped in a mixed (Adience-Mixed, classes [0, 2, 4, 6] and [1, 3, 5, 7]) and an ordered (Adience-Ordered, classes [0-3] and [4-7]) setup. The baseline is established by finetuning the model for three epochs per sequential task.

- **Entire Dataset**: In this scenario, the pretrained model was first pruned to retain the knowledge of ImageNet and then trained on the entire Adience dataset. As Adience is a very complex task, the pruning threshold is increased to 5% in order to increase the amount of free network capacity in the pretrained model. The model was finetuned on Adience for six epochs.

Similar to the experiments on the benchmark datasets, the results of the proposed method display a lower individual accuracy of the second task compared to the default baseline for both Split tasks, which can be seen in Table 1. Nevertheless, the baseline displays significant catastrophic forgetting, especially for the ordered setup with a drop of almost 46% for the previously learned task that is almost completely prevented when using Relevance-based Neural Freezing. The lower accuracy scores on task two can be explained by the stability/plasticity dilemma: decreasing the plasticity of parts of the network can increase stability for already acquired knowledge, but can slow down learning of new tasks, so that with the same amount of training epochs, the task is not learned to the
same degree. Even though the accuracy of the second task is lower, Relevance-based Neural Freezing still shows that the application of the method is advantageous for sequential tasks regardless of complexity of tasks and size of the model as the mean accuracy over both tasks increases by about 18%. In both setups, the model retains about 30% of capacity, making it possible to learn further tasks.

As can be seen in Figure 4b, preserving the knowledge of the ImageNet dataset in the pretrained model requires about 73% of the models full capacity. The results of Relevance-based Neural Freezing on the Adience-ImageNet split can be found in Table 2. While the accuracy of task two after our method is again lower than the default baseline, the accuracy of task one only drops about 2% compared to almost 20% in the default case, granting a mean accuracy increase of 5.6% over both tasks while still retaining about 15% free capacity in the model that can be used to learn further tasks.

4.6 Qualitative Results

Alongside the experiments, we also observe changes in the heatmaps computed by LRP before and after the application of Relevance-based Neural Freezing. Figure 5a shows heatmaps computed w.r.t. different age groups as targets alongside the original image. Red pixels denote positive relevance towards the target class while blue pixels denote negative relevance. The model learns to associate different facial features with specific age groups, which is especially apparent in the first image, where positive relevance is assigned to the glasses for the age group [48 - 53] while negative relevance is assigned in the upper area of the face when the target age group is [8 - 13]. The effect of catastrophic forgetting on the relevance distribution can be observed in Figure 5b. Computing the relevance after training the second task shows that using the default baseline, the model now assigns negative relevance to the upper area of the face that was previously considered a positive class trait. Relevance-based Neural Freezing retains the initial relevance assignments and still focuses on the glasses of the woman. Similar behavior is shown in the other images: while the assignment of relevance changes...
Table 1: Average test accuracy on Adience for both the ordered and the mixed setup in two tasks alongside the model’s free capacity after each freezing stage. The first test accuracy column represents the accuracy of the respective task after training the first task. The second test accuracy column tracks the change in test accuracy of the respective task after training the second task.

| Ordered Approach | Current Task | Test Accuracy | Free Capacity |
|------------------|--------------|---------------|---------------|
| Default fine Tuning | Task 1       | 64.38(±3.4)   | 18.7(±6.24)   | 100            |
| Default fine Tuning | Task 2       | -             | 58.15(±9.76)  | 100            |
| Average over both |              | 38.43(±8)     |               |                |
| Relevance-based Neural Freezing | Task 1     | 62.52(±5.6)   | 62.87(±6.3)   | 54.53(±9.16)   |
| Relevance-based Neural Freezing | Task 2     | -             | 50.02(±7.23)  | 29.4(±9.4)     |
| Average accuracy over both tasks |                      | 56.45(±6.7)   |               |                |

| Mixed Approach | Current Task | Test Accuracy | Free Capacity |
|----------------|--------------|---------------|---------------|
| Default Fine-Tuning | ImageNet     | 68.3(±4.33)   | 52.71(±9.6)   | 100            |
| Default Fine-Tuning | Adience      | -             | 80.42(±4.47)  | 100            |
| Average accuracy over both tasks |                      | 66.57(±7)     |               |                |
| Relevance-based Neural Freezing | ImageNet     | 70.89(±4.33)  | 70.11(±3.63)  | 47.57(±2.8)    |
| Relevance-based Neural Freezing | Adience      | -             | 75.85(±4.24)  | 28.8(±3.76)    |
| Average accuracy over both tasks |                      | 72.98(±3.9)   |               |                |

Table 2: Average test accuracy on splitting ImageNet and Adience into two tasks alongside the model’s free capacity after each pruning stage. The first test accuracy column represents the test accuracy after training ImageNet. The second column tracks the change in test accuracy of ImageNet after training on Adience and the test accuracy on Adience.

| Approach | Current Task | Test Accuracy | Free Capacity |
|----------|--------------|---------------|---------------|
| Default Fine-Tuning | ImageNet | 71.5(±0) | 51.51(±3.42) | 100            |
| Default Fine-Tuning | Adience | - | 51.02(±1.66) | 100            |
| Average accuracy over both tasks |                      | 61.27(±2.54) |               |                |
| Relevance-based Neural Freezing | ImageNet | 71.5(±0) | 69.05(±0.16) | 26.5(±0)       |
| Relevance-based Neural Freezing | Adience | - | 44.66(±1.61) | 14.9(±0.02)    |
| Average accuracy over both tasks |                      | 56.86(±0.89) |               |                |

after training with the default baseline, the model that used Relevance-based Neural Freezing is still focusing on the areas that were relevant before training the second task and keeps the sign of the relevance consistent. Finally, the generated heatmaps are consistent with the test accuracy results displayed in the Figures 4a and 4b and illustrate how previously learned features are preserved despite the introduction of new classes when Relevance-based Neural Freezing is employed.

5 Conclusion

Overcoming catastrophic forgetting is one key obstacle towards achieving reliable lifelong learning. Retraining the model from scratch every time new data or tasks are added is sometimes possible, but very inefficient. In order to prevent the model from forgetting previously learned information, the plasticity of important neurons can be lowered so they retain the ability to solve earlier tasks. We present an effective algorithm that uses LRP to identify the neurons that contribute the most to a given task, achieving comparable results as state-of-the-art methods. Evaluation of the proposed method on the CIFAR10-100 split achieved an increase in accuracy of about 40% compared to the
Fig. 5: (a): Images from the Adience dataset alongside their explanations. The red colored regions in the heatmaps show relevant features used by the model for recognizing the chosen class, while blue corresponds to negative relevance, marking contradicting evidence. The relevance is computed w.r.t the target class labels indicated on the left of each heatmap. Choosing different target classes produces different explanations, as different class outputs of the model utilize the presented input information differently. (b): Original images from the Adience dataset alongside their explanations for the true class, after either using default finetuning over several tasks, or the proposed method. The Figure shows samples from task 1 before and after learning task 2 in the ordered split experiment. This demonstrates that our Relevance-based Neural Freezing approach prevents a drift in the reasoning of the model under continued training for tasks already optimized, which is occurring under default fine tuning and is known as catastrophic forgetting.

In addition to the benchmark datasets, we also evaluated three scenarios on the Adience dataset. We could show that our methods not only work with difficult and unbalanced data but also in a multi-dataset scenario. Retaining the knowledge after training on ImageNet to learn the entire Adience dataset conserved 18% of accuracy compared to the baseline, achieving a net gain of about 11% over both tasks. We were able to show that Relevance-based Neural Freezing is scalable, efficient and resilient against catastrophic forgetting in sequential learning setups. As LRP is applicable to a wide range of network architectures, this technique can also be applied to arbitrary domains, e.g. neural language processing. An additional benefit our technique introduces is functional annotation of neural networks. After identifying the relevant parts of the network for a specific task, it is possible to save the learning rate mask, which then serves as an annotation for other researchers or practitioners.

6 TODO

- Reviewer 1
  - 1. In MNIST-split data, before training the neural network for the third task (and the subsequent tasks), which data are used for LRP? If the the authors only use the data for task 2, then how can the catastrophic forgetting for task 1 be achieved. But if both data for task1 and task 2 are used, then this contradicts one of the main objective in that the original data may not be available. this should be cleared up as we mention and define the reference dataset in chapter 3. I made it bold to make more clear. It’s true that it contradicts the notion of not having the original data available, so maybe we should clarify that this can also be a dataset that is similar to the (assumed) data distribution?
Fig. 2(a): there is rebound for task 1’s curve after the learning of task 5. Please give an explanation on this.

Adding to MNIST-Split

The increase in baseline accuracy after task five can be attributed to a similarity between the shapes of digits in tasks one and five: both the digits 8 and 9 have rounded forms which makes it easier to distinguish between a 0 and a 1.

1. the graphs’ legends are too small to see.
2. inconsistent usage of "wrt" and "w.r.t"

Reviewer 2

✗ I would suggest, if the paper will be accepted, to reframe the title.
✓ I wonder if the proposed approach is model agnostic (LRP should be as far as I understand) and can be used to deal with catastrophic forgetting in networks and tasks not related to image classification. This is not discussed in the paper. add a sentence about this in Conclusion? Something like: As LRP is applicable to a wide range of network architectures, this technique can be applied to arbitrary domains, e.g. neural language processing.
✓ Some parts about the experiments should be made clearer. For instance, the free capacity of the network / model is used everywhere but it is not properly introduced. In the more up to date version of the paper, this is explained in Section 3.
✓ It is not clear to me how the baseline (the lower bound) is determined. It is said that to establish a baseline, the model is first pre-trained. For what task is the model pre-trained? Is the lower bound the Default Fine Tuning referred in the Figures?
□ It is also not clear to me why the upper bound is mentioned but almost not used in the comparison. In the figures it does not appear. True, the upper bound is only mentioned once. Serop uses it in the paper to compare MNIST Split with other methods, but since they are able to reach higher scores, I omitted that comparison, see Figure ??

| Approach               | Experiment          | Task-IL       |
|------------------------|---------------------|---------------|
| Baselines [?]          | Random-6 Tasks      | 62.5(±1.2)    |
| Baselines [Ours]       | Random-6 Tasks      | 41.82(±3.72)  |
| Training By Explaining| Random-6 Tasks      | 72.16(±5.46)  |
| Synaptic Intelligence  | Random-6 Tasks      | 73.52(±1.04)  |
| Baselines [Ours]       | Random-11 Tasks     | 28.37(±3.1)   |
|                        | Semantic-8 Tasks    | 28.62(±1.9)   |
| Training By Explaining| Random-11 Tasks     | 69.74(± 6.36) |
|                        | Semantic-8 Tasks    | 56.70(±3.2)   |

Table 3: Comparison between the achieved average test accuracy on CIFAR10-100 Split of 6 sequential tasks with the literature mentioned method, synaptic intelligence. The experiments are repeated five times, and the results are presented as mean (±STD). The test accuracy was also compared between our baseline and when training by explaining was applied to splitting the entire CIFAR100 either randomly or semantically.

EWC works by determining importance of parameters by calculating the Fisher information per parameter after training every task and using the Fisher Information matrix as a constraint in the loss function. The authors claim to have a linear runtime with respect to the parameters and the data samples but it appears to have need for the dataset to be present.
Context Dependend Gating (XdG): Combines EWC and Synaptic Intelligence with random gatings per task identity, this requires the task identity at runtime though.

☑ If the task-incremental setup is used in (almost) all the experiments, it would be maybe better to reserve a subsection at the beginning of Section 4 to describe the experiments' setup. Generally, I would suggest to describe all relevant knowledge at the beginning of Section 4. Move explanation about task incremental setup to preamble of section 4.

☐ It is claimed (in the conclusion) that comparable results w.r.t. state-of-the-art methods are obtained. Are these methods those described in Section 2.2 (to mitigate Catastrophic Forgetting)? Then it seems to me that this comparison is not available in the paper. If you refer to some papers where experiments using the same datasets are provided please add it, or if already present, make it more explicit. **Removed because not using**

☒ Minor points:

☑ Page 1, Section 1: the sentence 'If more data is later added, and the model should be able to correctly predict [...] from non-stationary data.' is too long and should be rephrased. **If more data is later added, and the model should be able to correctly predict on both new and old data, usually it has to be fine-tuned or trained from scratch with the expanded dataset as opposed to only the new data. Otherwise, catastrophic forgetting** can occur when learning multiple consecutive tasks or from non-stationary data.

☑ Page 4, Section 4 (experiments):

- to establish a baseline, the model is first pre-trained to perform what task? **Clarified that the model was trained on the first task and then finetuned thereafter after talking to Serop**

☑ the free capacity of the network should be defined.

☑ Page 5, Section 4.3:

- 'and vehicles Like' -ि 'and vehicles. Like'
- 'While [...] as Figure 3c shows that Training by Explaining can [...]’ -ि 'While [...] (as shown in Figure 3c), Training by Explaining can [...]’

☑ Page 7, Section 4.4:

- ‘[...] catastrophic forgetting even after 10 epochs, as is evident in Figure 4a’. I am not sure epochs are represented in Fig 4a. **Clarified that each task was trained for only 10 epochs**

☑ Page 10, Table 1 caption: ‘task task’ -ि ‘task’

☒ The manuscript does not appear to be in the correct LNCS format - please be sure to check, correct and also ensure an even number of pages!

☑ Figures 2 and 3 appear very difficult to read in the paper printout, perhaps a solution can be found here to support the reader. **might have been solved because we made this larger already?**

☑ Section 2.1. in addition to references 6,17, the authors could add the work of Evans et al. (2022) - as a benefit for the interested reader, who would get a practical example in addition to the work of Hägele et al. see: https://doi.org/10.1016/j.future.2022.03.009

☒ GNN background? **This is out of scope for this paper. While we are working on implementing this functionality in a range of neural network architectures, and there exists an LRP Framework for GNNs, this will be explored in future work.**

☑ 3) Methodology: the main arguments that elements that are important freeze to a certain adaptable extend and others are there for learning is robust. The set threshold though needs
to be defined; at the moment it seems that it is empirically set, it would be desirable to give strategies, directions, search methodologies to set this. \textbf{Added: The optimal value for this parameter can be determined using grid search.}

\checkmark How was the “semantically similar classes” in figure 3 uncovered? \textbf{Add explanation that this was done manually? Added that the classes we ordered manually}

\xmark 4) Results: The fact that the free capacity is represented by the number of unfrozen nodes is reasonable, but it depends on the size of the neural network somehow. I would like to see how bigger or smaller networks differ from each other when trained with this method. “Training by Explaining slightly lowers the mean accuracy” in section 4.2. - why that happens is roughly explained, but how do you think that you can change the method a bit to tackle this issue – and preferably automatically? Similarly for 4.3. The figures are very good. Do you consider to change automatically the architecture because of the results, initiate an xAI-driven architecture search? \textbf{We’re working on this - currently exploring capability of dynamic network growth during training. In reference to [44] the motivation is a post processing architecture search - here, we’re trying to achieve an orthogonal goal, identifying what works in order to fit as much information into the network as possible}

– Stuff to put in there

\checkmark Relevance based neural-freezing

\square \textbf{Redo all images to rename method as well}

\checkmark Annotation of networks

\checkmark rename lower bound to naive finetuning

\textbf{Acknowledgment}

This work was supported by the German Ministry for Education and Research as BIFOLD (ref. 01IS18025A and ref. 01IS18037A), the European Union’s Horizon 2020 programme (grant no. 965221 and 957059), and the Investitionsbank Berlin under contract No. 10174498 (Pro FIT programme).

\textbf{References}

1. Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K.T., Montavon, G., Samek, W., Müller, K.R., Dühne, S., Kindermans, P.J.: innvestigate neural networks! Journal of Machine Learning Research \textbf{20}(93), 1–8 (2019)

2. Anders, C.J., Neumann, D., Samek, W., Müller, K.R., Lapuschkin, S.: Software for dataset-wide xai: From local explanations to global insights with Zennit, CoRelAy, and ViRelAy. arXiv preprint arXiv:2106.13200 (2021)

3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE \textbf{10}(7), e0130140 (2015)

4. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Mueller, K.R.: How to explain individual classification decisions (2010)

5. Becking, D., Dreyer, M., Samek, W., Müller, K., Lapuschkin, S.: Ecqx: Explainability-driven quantization for low-bit and sparse dnn. In: xxAI - Beyond Explainable AI, LNAI, vol. 13200, pp. 271–296. Springer (2022)

6. Chereda, H., Bleckmann, A., Menek, K., Perera-Bel, J., Stegmaier, P., Auer, F., Kramer, F., Leh, A., Beißbarth, T.: Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer. Genome medicine \textbf{13}(1), 1–16 (2021)

7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)

8. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine \textbf{29}(6), 141–142 (2012)

9. Eidinger, E., Enbar, R., Hassner, T.: Age and gender estimation of unfiltered faces. IEEE Transactions on Information Forensics and Security \textbf{9}(12), 2170–2179 (2014)
10. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. Technical Report, Université de Montréal (01 2009)

11. Evans, T., Retzlaff, C.O., Geißler, C., Kargl, M., Plass, M., Müller, H., Kiehl, T.R., Zerbe, N., Holzinger, A.: The explainability paradox: Challenges for xai in digital pathology. Future Generation Computer Systems 133, 281–296 (2022). [https://doi.org/https://doi.org/10.1016/j.future.2022.03.009](https://www.sciencedirect.com/science/article/pii/S0167739X22000838)

12. Farquhar, S., Gal, Y.: Towards robust evaluations of continual learning (2018)

13. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. 2017 IEEE International Conference on Computer Vision (ICCV) (2017)

14. French, R.: Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences 3, 128–135 (05 1999)

15. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)

16. Guyon, I., Weston, J., Barnhill, S., Vapnič, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (01 2002)

17. Hohman, F., Park, H., Robinson, C., Chau, D.H.: Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. arXiv preprint arXiv:1904.02323 (2019)

18. Hägele, M., Seegerer, P., Lapuschkin, S., Bockmayer, M., Samek, W., Klauschen, F., Müller, K.R., Binder, A.: Resolving challenges in deep learning-based analyses of histopathological images using explanation methods. Scientific Reports 10, 6423 (2020)

19. Kim, B., Wattenberg, M., Gilmer, J., Cai, C.J., Wexler, J., Viégas, F.B., Sayres, R.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: International Conference on Machine Learning. pp. 2668–2677. PMLR (2018)

20. Kirkpatrick, J., Pascanu, R., Rabinovitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R.: Overcoming catastrophic forgetting in neural networks (2017)

21. Kohlbrenner, M., Bauer, A., Nakajima, S., Binder, A., Samek, W., Lapuschkin, S.: Towards best practice in explaining neural network decisions with lrp. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–7 (2020)

22. Krizhevsky, A.: Learning multiple layers of features from tiny images. Master’s thesis, University of Toronto, Department of Computer Science (2009)

23. Lange, M.D., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G.G., Tuytelaars, T.: Continual learning: A comparative study on how to defy forgetting in classification tasks. arXiv preprint arXiv:1909.08383 (2019)

24. Lee, J., Yoon, J., Yang, E., Hwang, S.J.: Lifelong learning with dynamically expandable networks. arXiv preprint arXiv:1708.01547 (2017)

25. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition 65, 211–222 (May 2017)

26. Nguyen, A.M., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. arXiv preprint arXiv:1605.09304 (2016)

27. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill 2(11), e7 (2017)

28. Oren, G., Wolf, L.: In defense of the learning without forgetting for task incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 2209–2218 (2021)

29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. pp. 8024–8035 (2019)

30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021)

31. Ribeiro, M.T., Singh, S., Guestrin, C.: "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. pp. 1155–1164 (2016)

32. Samek, W., Binder, A., Montavon, G., Bach, S., Müller, K.R.: Evaluating the visualization of what a deep neural network has learned. IEEE Transactions on Neural Networks and Learning Systems 28(11), 2660–2673 (2017)

33. Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R.: Explaining deep neural networks and beyond: A review of methods and applications. Proceedings of the IEEE 109(3), 247–278 (2021)

34. Samek, W., Wiegand, T., Müller, K.R.: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. ITU Journal: ICT Discoveries 1(1), 39–48 (2018)
35. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021)
36. Serrà, J., Surís, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. arXiv preprint arXiv:1801.01423 (2018)
37. Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., et al.: Mastering the game of go with deep neural networks and tree search. Nature 529, 484–489 (01 2016)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (ICLR) (2015)
39. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International Conference on Machine Learning. pp. 3319–3328. PMLR (2017)
40. van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. CoRR abs/1904.07734 (2019), http://arxiv.org/abs/1904.07734
41. Wilm, F., Benz, M., Bruns, V., Baghdadiian, S., Dexl, J., Hartmann, D., et al.: Fast whole-slide cartography in colon cancer histology using superpixels and CNN classification. Journal of Medical Imaging 9 (2022)
42. Wortsman, M., Ramanujan, V., Liu, R., Kembhavi, A., Rastegari, M., Yosinski, J., Farhadi, A.: Super-masks in superposition. arXiv preprint arXiv:2006.14769 (2020)
43. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. CoRR abs/1905.13260 (2019), http://arxiv.org/abs/1905.13260
44. Yeom, S.K., Seegerer, P., Lapuschkin, S., Binder, A., Wiedemann, S., Müller, K.R., Samek, W.: Pruning by explaining: A novel criterion for deep neural network pruning. Pattern Recognition 115, 107899 (2021)
45. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. arXiv preprint arXiv:1311.2901 (2013)
46. Zenke, F., Poole, B., Ganguli, S.: Improved multitask learning through synaptic intelligence. arXiv preprint arXiv:1703.04200 (2017)
47. Zintgraf, L.M., Cohen, T.S., Adel, T., Welling, M.: Visualizing deep neural network decisions: Prediction difference analysis. arXiv preprint arXiv:1702.04595 (2017)
A Appendix

A.1 MNIST-Split

The model architecture was taken from [40], which compares multiple methods for mitigating catastrophic forgetting. It consists of two hidden layers with 400 neurons each and ReLU activations. As for most experiments except the real-world dataset, the pruning threshold is set to 2%, meaning that the accuracy can drop by up to 2% before the pruning procedure is halted. We use the Adam optimizer with a learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with a batch size of 128.

A.2 MNIST-Permuted

For this experiment, the architecture from [34] was adapted by increasing the number of hidden layer units to 1000 per layer to match the increased complexity of the task. Additionally, the learning rate was decreased to 0.0001 and the model was trained for ten instead of four epochs per task.

A.3 CIFAR10-100

In this experiment, we adopted architecture and experimental setup from [46].

A.4 ImageNet Split

Here, we replicate the conditions from [43] but establish our baseline after ten instead of 70 epochs, which we also use when Training by Explaining.

A.5 Adience

As is state-of-the art for this dataset [9], we normalize to zero mean and unit standard deviation during training and apply data augmentation by randomly cropping to 244x244 as well as horizontal flipping. For inference, each sample is randomly cropped five times to 244x244, where each crop is additionally mirrored. The ground truth is then compared to the mean of the Softmax activations of the ten samples. As the dataset is strongly imbalanced, we additionally employ a resampling strategy during training that undersamples classes with a high number of samples and oversamples classes with a low number of samples by computing the class probabilities and then sampling from a multinomial distribution.

In this experiment, we employ a VGG-16 network architecture [38] that has been pretrained on ImageNet (from the PyTorch [29] model zoo), as well as an Adam optimizer and L2 regularization.

- **Split**: We used a learning rate of 0.0001, L2 regularization with $\lambda = 0.01$ and a batch size of 32.

- **Entire Dataset**: The model was trained with a learning rate of 0.00001 and $\lambda = 0.001$. 