Replication of paper CRITICAL LEARNING PERIODS IN DEEP NETWORKS by Alessandro Achille, Matteo Rovere, Stefano Soatto

Bartłomiej Krzepkowski

About

Researchers are trying to present, using the example of neural networks and the image classification task, the existence of a period (critical period) in which an adaptive information processing system develops associations that are difficult to modify in the later period of its development.

In all experiments, for computational reasons, the number of intermediate checkpoints was limited while maintaining the main effect which can be received.

MNIST Setting

Model:

Fully Connected Neural Network consistent with the description of the article.

Optimizer: SGD, lr = 0.005

Transformation of image: padding to 32x32, normalization (0.5, 0.5)

Random Seed = 42

```python
class MnistFC(nn.Module):
    def __init__(self):
        super().__init__()
        self.net = nn.Sequential(nn.Linear(1024, 2500), nn.BatchNorm1d(2500), nn.ReLU(),
                                 nn.Linear(2500, 2000), nn.BatchNorm1d(2000), nn.ReLU(),
                                 nn.Linear(2000, 1500), nn.BatchNorm1d(1500), nn.ReLU(),
                                 nn.Linear(1500, 1000), nn.BatchNorm1d(1000), nn.ReLU(),
                                 nn.Linear(1000, 500), nn.BatchNorm1d(500), nn.ReLU(),
                                 nn.Linear(500, 10))

    def forward(self, x):
        x = x.reshape(x.shape[0], -1)
        x = self.net(x)
        return x
```

CIFAR10 Setting

Model:

AllCNN model consistent with the description of the article.

Optimizer: SGD, lr = 0.05, weight decay = 0.001

Exponential annealing schedule:gamma = 0.97

Transformation of image: random translations up to 4 pixels, random horizontal flipping, normalization (0.5, 0.5, 0.5)

Random Seed = 42

```python
class AllCNN(nn.Module):
    def __init__(self):
        super().__init__()
        self.net = nn.Sequential(nn.Conv2d(3, 96, kernel_size=3, padding=2),  # 34
                                 nn.BatchNorm2d(96),
                                 nn.ReLU(),
                                 nn.Conv2d(96, 96, kernel_size=3, padding=1),  # 34
                                 nn.BatchNorm2d(96),
                                 nn.ReLU(),
                                 nn.Conv2d(96, 192, kernel_size=3, stride=2),  # 16
                                 nn.BatchNorm2d(192),
                                 nn.ReLU(),
                                 nn.Conv2d(192, 192, kernel_size=3, padding=1),  # 16
                                 nn.BatchNorm2d(192),
                                 nn.ReLU(),
                                 nn.Conv2d(192, 192, kernel_size=3, stride=2),  # 7
                                 nn.BatchNorm2d(192),
                                 nn.ReLU(),
                                 nn.Conv2d(192, 192, kernel_size=3),  # 5
                                 nn.BatchNorm2d(192),
                                 nn.ReLU(),
                                 nn.Conv2d(192, 192, kernel_size=1),  # 5
                                 nn.Conv2d(192, 10, kernel_size=1)  # 5
                                 )
        self.avg = nn.AvgPool2d(kernel_size=5)

    def forward(self, x):
        x = self.net(x)
        x = self.avg(x)
        x = torch.squeeze(x)
        return x
```

Deficit Removal

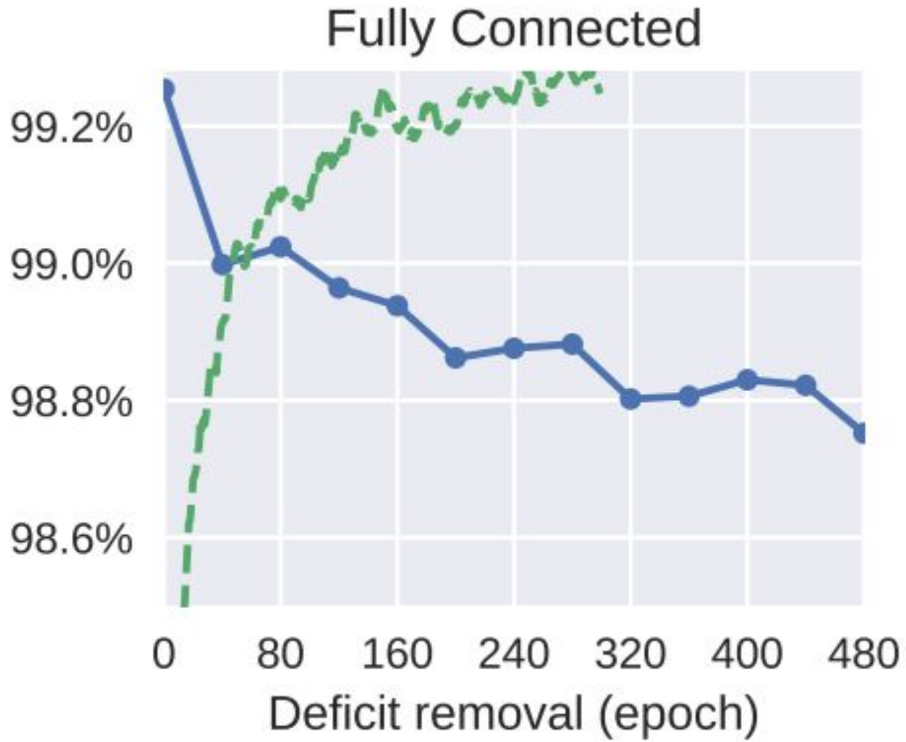
Investigation of the impact of the length of access to deficit data on the accuracy of further normal learning of the neural network.

A model was first trained over a certain number of epochs on the perturbed data (data with deficit), then learned over N epochs on the correct data.

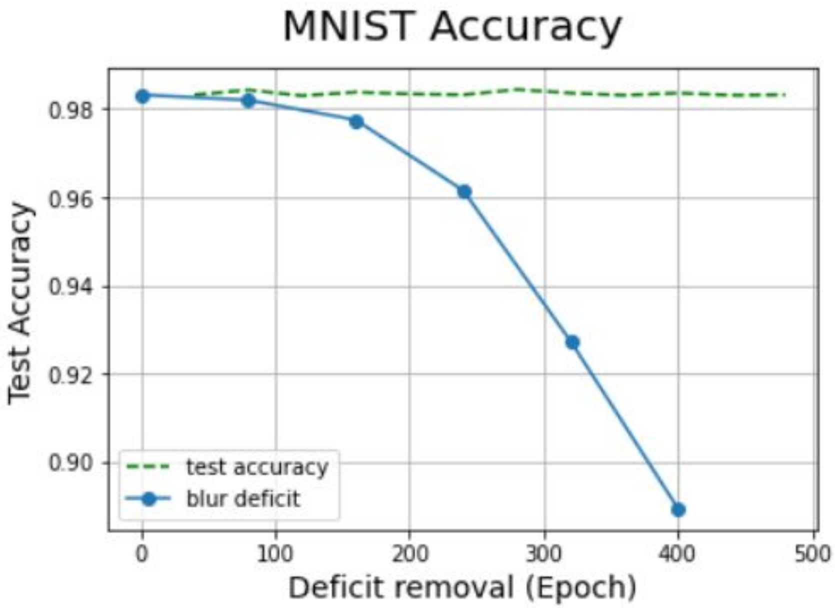
In the case of MNIST: N=480

In the case of CIFAR10: N=140

Results on MNIST

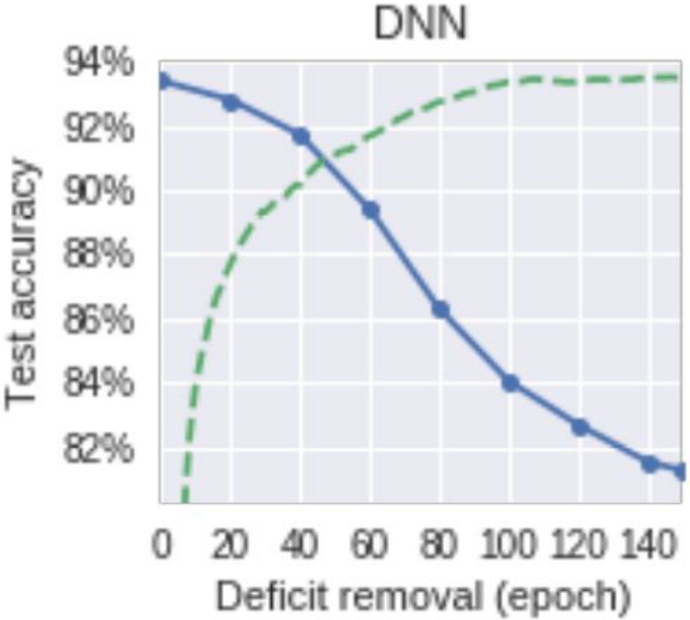


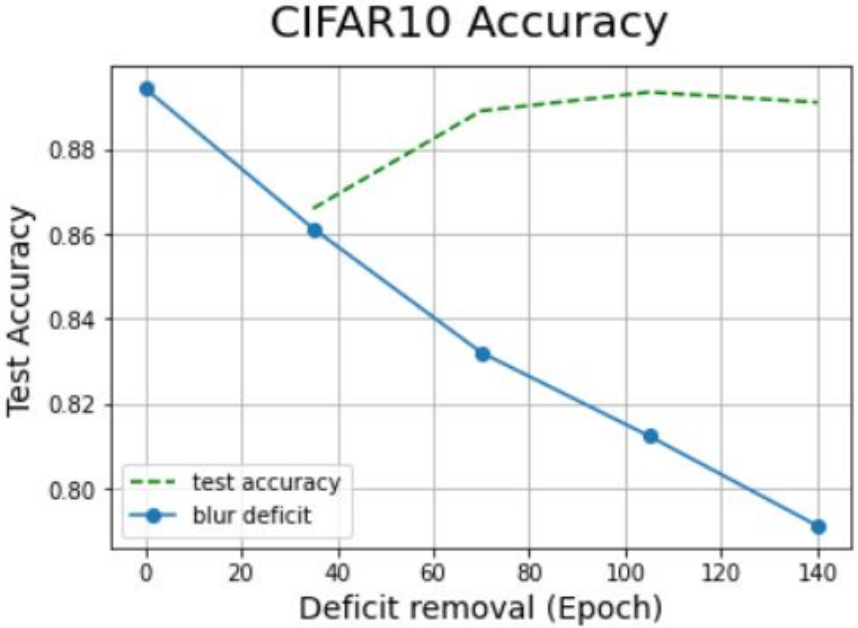
Paper



Replication

Results on CIFAR10





Paper

Replication

Comment

It can be seen that it failed to accurately replicate the models that were used in the original article, but the trend is being maintained.

In the experiment with MNIST, however, we can observe a significant decrease in the accuracy of the classification compared to the result in the article.

The dashed line represents the accuracy test of the model without a deficit.

Other deficits

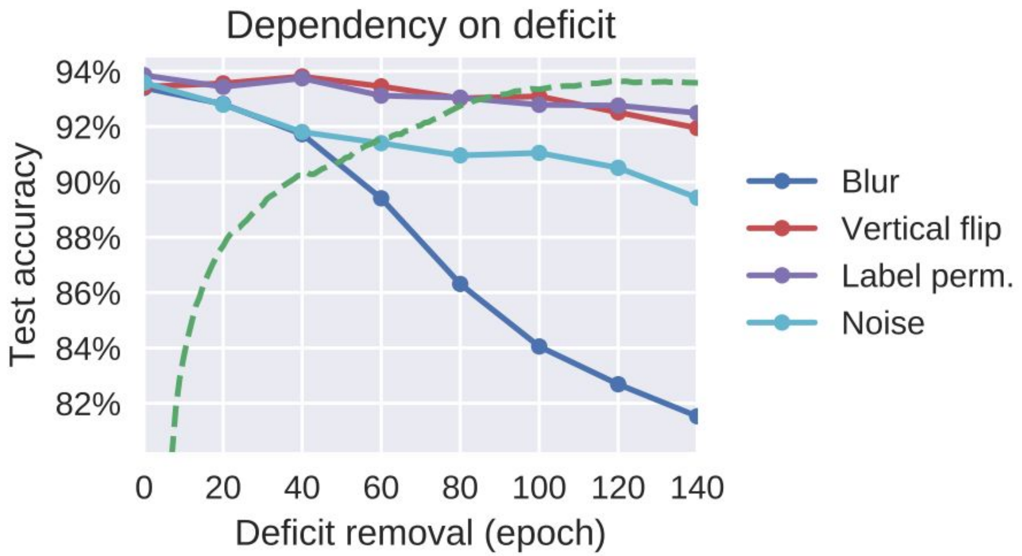
To compare the selection of a deficit, which is blur, the quality of other deficits, named as higher levels, was examined, at which the model was to return to the initial level of classification efficiency more efficiently.

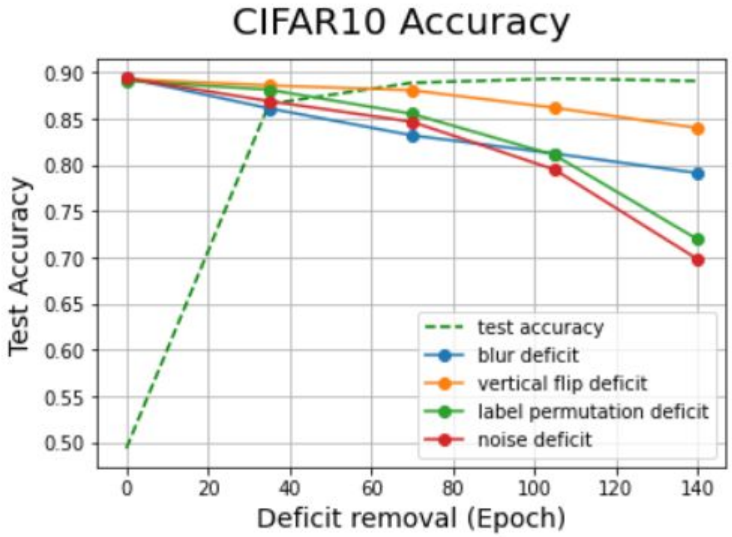
These higher-level deficits have been named:

- Random permutation of labels
- Random Vertical Flip
- Noise as an input signal to the model

in the training set.

Results on CIFAR10





Paper

Replication

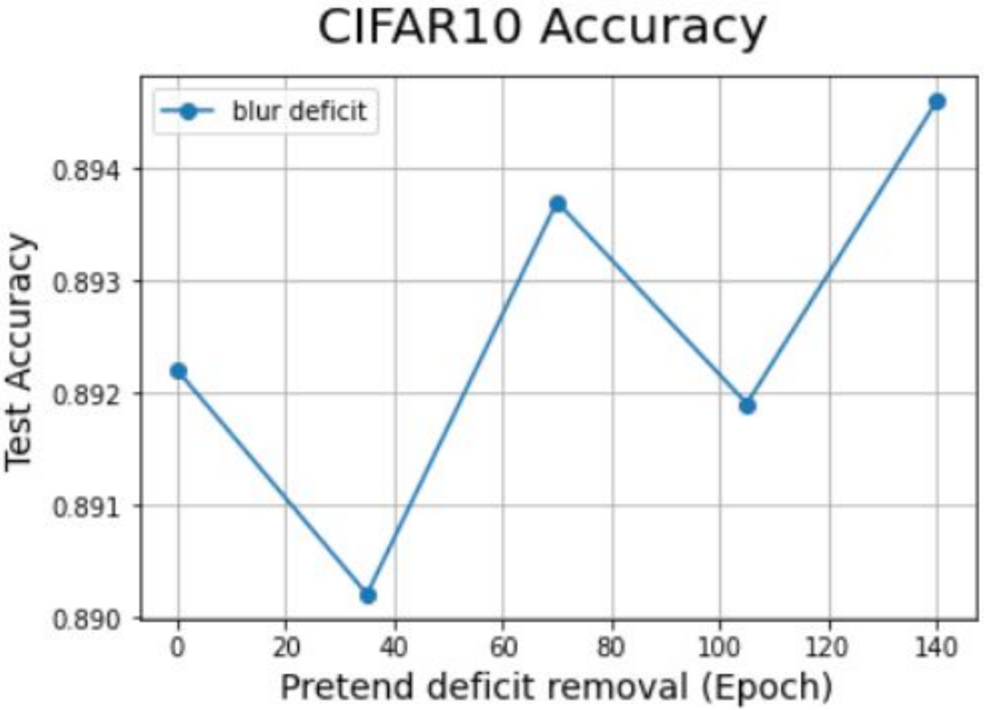
Comment

As can be seen, the trend is maintained in the case of vertical flipped deficit and blurred deficit, although in the case of the former it seems that the decrease is too large, because the highest level deficits were supposed to be almost independent of the deficit. The second of the highest-level deficits has the most drastic decrease along with the noise deficit. This is the opposite of what is described in the article.

The dashed line represents the accuracy test of the model without a deficit.

Comment

To test the correctness of the created framework, an experiment was carried out in which there was a transition from correct data to correct data in training loop.



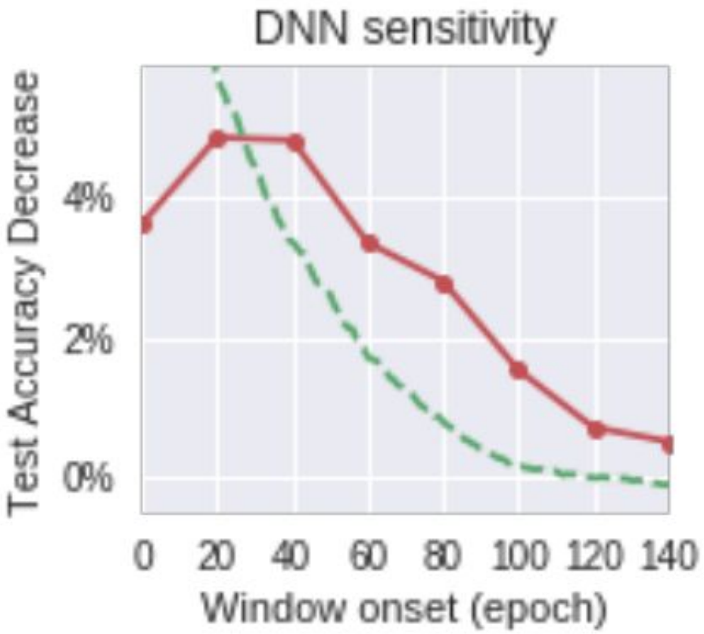
Which excludes the argument that the very moment of transition between datasets reduces the effectiveness of the classification.

Sensitivity during learning

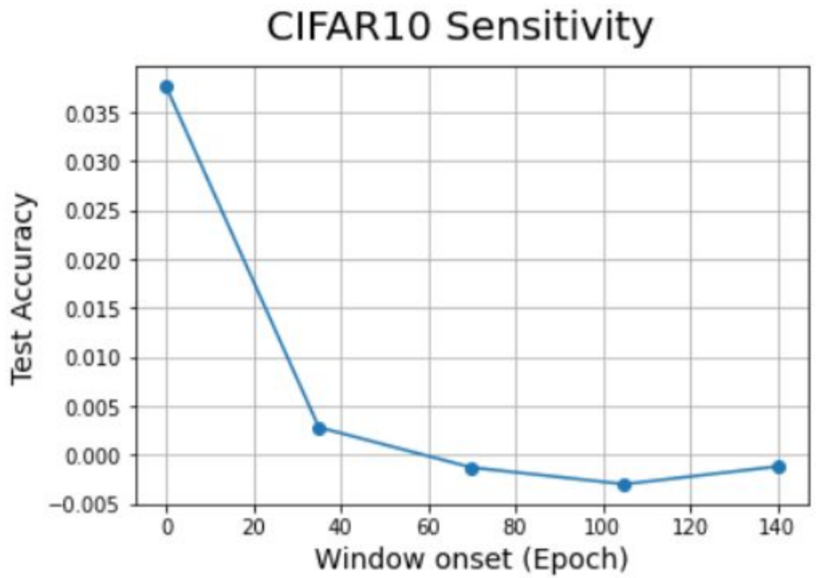
Investigation of the impact of the arrangement of the access window to disturbed data on the period of specific number of epochs on the effectiveness of further network learning.

The model was first trained on valid data for a specified number of epochs, then trained on perturbed data for 35 (in paper 40) epochs, and then again on correct data 140 epochs. The decrease in the classification efficiency depending on the window arrangement was investigated.

Results on Cifar10



Paper



Replication

Comment

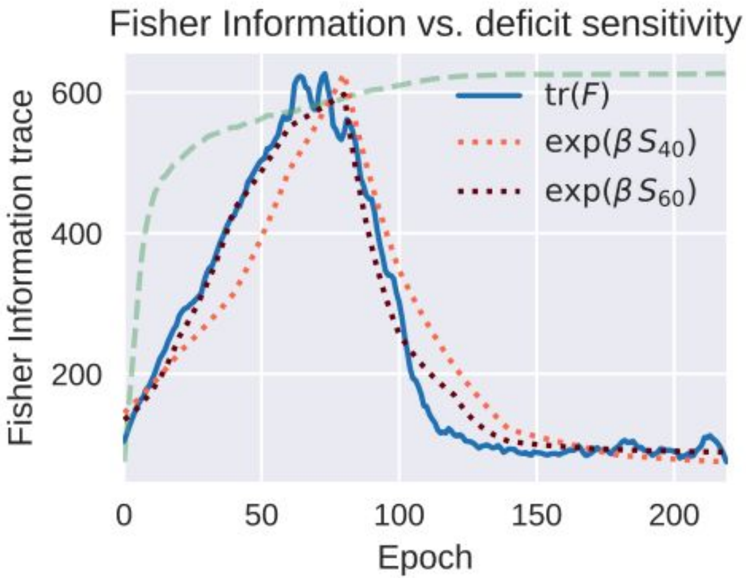
As can be seen, it failed to replicate the bell-shaped shape of the curve. Moreover, the replicated values also take negative values, which are supposed to suggest that the disturbed data later in the training should improve the final classification efficiency.

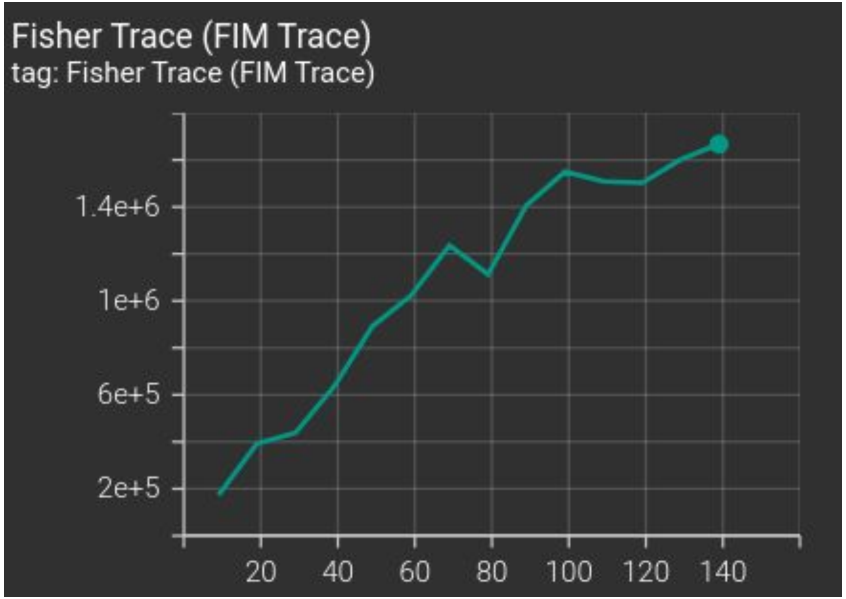
Fisher Information Matrix Trace (Trace of FIM)

The main idea to investigate critical periods is to investigate the variance of local information change during the learning process on the models and data already described.

Unfortunately, the author of this replication is not convinced about his own implementation of this fragment, so he only attaches the obtained replication and will remain silent until he receives advice.

Results on Cifar10





Paper

Replication

Summary

Unfortunately, the author of this replication cannot be sure of the results, because he considers the description of the procedure described by the authors to be vague (partly, especially in context of FIM), and he did not find any repository with the replication of this study on the Internet.

It is also possible that in the code it was placed in the repository

https://github.com/BartekKrzepkowski/-Replication--Critical-Learning-Periods-in-deep-networks

there is an error that was not found.

Thank you for your attention!