UiT

THE ARCTIC UNIVERSITY OF NORWAY

Home Exam

Home Exam in	**FYS-3033 - Deep Learning**
Hand-out:	**Monday March 27, 2023, 09:00**
Hand-in:	**Wednesday April 26, 2023, 10:00**

The Home Exam contains 7 pages including this cover page

Contact person:	**Michael Kampffmeyer**
Email:	**michael.c.kampffmeyer@uit.no**

UiT

THE ARCTIC UNIVERSITY OF NORWAY

Before You Start

Portfolio instructions

Your code should be submitted together with your report (see instructions below). **Further, please include a discussion of the results obtained and a discussion of the implementation in your report, which show that you understand what you are doing.**

The code should be commented in such a way that any person with programming knowledge should be able to understand how the program works. Like your report, the code must be your own individual work.

You are permitted to use deep learning frameworks such as Pytorch and Tensorflow. As there is a lot of code available online, please make sure that your report and code clearly show that you understand what you are doing.

Hand-in format

Please submit your report (in pdf format) to WISEflow and attach *one* single `.zip` file that contains two folders, one called `doc` that contains your report, and another one called `src` containing the code. The file name of the `.zip` file should follow the format `homeexam_candidate`*`XX`*`.zip` (replace *`XX`* with your candidate number obtained from WISEflow) for anonymity.

Please include your candidate number and the course name on the frontpage of your report.

Follow the hand-in instruction in Wiseflow. Upload the pdf as the main file and then attach your zip as an attachment. Note, the reports will be processed by a plagiarism checker and the **pdf file size must not exceed 15MB**.

UNIVERSITY OF TROMSØ

Problem 1

In this problem you will derive and discuss some of the theoretical results that are found in deep learning.

(1a) Describe the importance of random initialization in deep learning. Derive the He initialization (1), stating and discussing the underlying assumptions and their reasonability.

(1b) Discuss how the He initialization differs from the Xavier initialization (2).

(1c) Derive the expression for the KL divergence between two Bernoulli distributions. Discuss how this result might be used in Variational Autoencoders.

(1d) The goal of generative models is to sample from the underlying distribution, p_{data}, of data $\mathbf{x}$. Generative Adversarial Networks (GANs) achieve this goal by constructing a generator G, which tries to fool a discriminator D. The generator transforms a sample $\mathbf{z}$ from a prior noise distribution, $p_{\mathbf{z}}$, into a new distribution p_g. Given the loss that the discriminator is tasked to maximize

$$V(G, D) = \int_x p_{\text{data}(x)} \log(D(x))dx + \int_z p_z(z) \log(1 - D(g(z)))dz \tag{1}$$

and assuming that the optimal solution has been reached ($p_g = p_{\text{data}}$). Show that the optimal solution for the discriminator given a fixed generator g is

$$D(x) = \frac{1}{2} \, . \tag{2}$$

Finally, illustrate that this optimum corresponds to a value of $-\log(4)$.

Problem 2

In this problem you will train an image classifier on the provided traffic sign dataset (3) and explore several approaches to explain the models predictions and detect backdoors. The data can be found on Canvas in `Canvas/Files/homeexam/problem2.zip`.

(2a) Explain one approach that can be used to interpret/explain a specific prediction of a deep learning model and one that can be used to interpret/explain the model as a whole.

(2b) Train a ResNet-18 to classify the data. Describe the network and your implementation and report the obtained accuracy for the training and validation data. For this, use the data contained in the "train" folder and split it into a 80%/20% train/validation split. Split the data per class so that you keep the same class distribution.

(2c) Both quantitatively and qualitatively inspect the results on the provided test images (in the "test" folder) and compare these results to what you obtained on the validation set. Discuss what you observe.

(2d) For the class where the model fails on the testset, produce Class Saliency Maps for the training images as described in (4) (Section 3.1) to discover what the model bases its predictions on. Provide the saliency maps of 10 training images and comment on the observed results.

(2e) For the same training images, perform also an Occlusion analysis as described in (5) (Section 4.2) where you occlude a 10x10 region and monitor the classifier output as you occlude different parts of the image. Provide illustrations of the same examples as in 2c) that mimic Figure 7 (e) in (5) and comment on the observed results.

(2f) Based on your observations, propose an approach to modify the training data to remove the backdoor in the training dataset.

(2g) Retrain your classifier on the modified data and report the new results for the test images. Discuss the results.

(2h) Provide the saliency maps and the occlusion analysis maps for the new model for the same images as in 2d) and 2e). Discuss the results.

Problem 3

In this problem, you are given two datasets of images, SVHN and MNIST, where the SVHN dataset contains labeled images of house numbers, and the MNIST dataset contains images of handwritten digits that are unlabeled. You want to train a deep learning model that is trained on the labeled SVHN dataset that is able to classify the handwritten MNIST digits. Scripts for loading both datasets are provided on Canvas in `Canvas/Files/homeexam/problem3.zip`. Hint: The labels of the SVHN dataset range from 1-10 instead of 0-9, which is the case for the MNIST dataset. The label '10' actually corresponds to the digit 0 in the images. Depending on how you implement the different networks, this might result in some difficulties. The simplest solution is to create a new label vector where the digit '0' is labeled with a '0', such that the labels range from 0-9 instead of 1-10.

(3a) Implement and train a network to classify the SVHN images. The network architecture is given in Figure 1. Note, this is a wider version of the model that you implemented in the mandatory assignment.

(3b) Use the network trained in Problem (3a) directly to classify the images of the MNIST dataset. Report the achieved accuracy.

You find that the performance of the model is significantly better on the SVHN dataset than on the MNIST dataset, even though the numbers in both datasets are the same. To improve performance on the MNIST dataset, we will in the following design a MNIST classifier based on the Adversarial Discriminative Domain Adaptation (ADDA) (6) approach illustrated in Figure 2.

(3c) In the ADDA framework, a discriminator is used to train the target feature extractor by aligning its extracted features with the features from the source feature extractor (the CNN trained in Problem 3a without the output layer). In Figure 3, the discriminator in the framework has been replaced with a mean squared error loss. Discuss, why this is not a good idea.

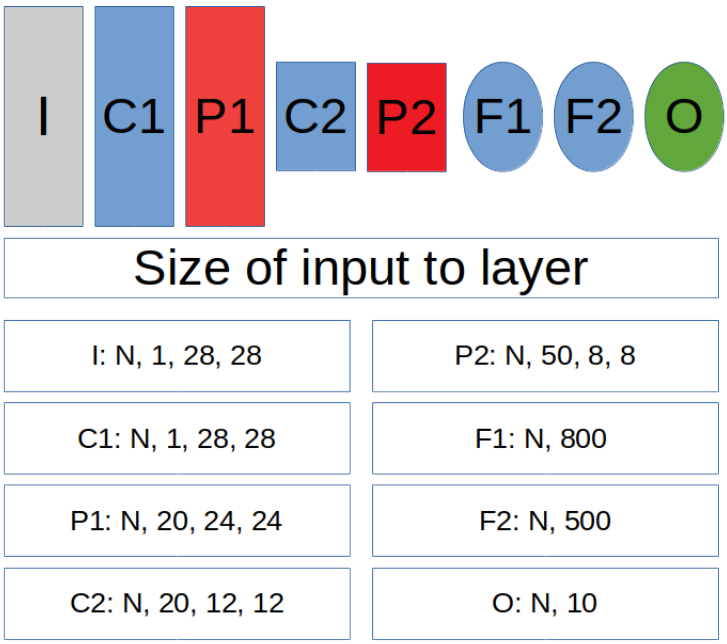

Figure 1: *Illustration of a wide LeNet-5 architecture. Layers labeled with a 'C' refers to a convolutional layer, which performs a convolution operation followed by an activation function. Layers labeled with a 'P' refers to a pooling layer, which performs a max pooling operation. Layers labeled with a 'F' refers to a fully connected layer, which performs matrix multiplication followed by an activation function. The 'I' and 'O' layer indicate the input and the output layer, respectively. Below the architecture itself is a description of the size of the input to each layer. There is no padding in the convolutional layers, stride is set to 1, and the filters are of size* 5×5*. The pooling layers apply* 2×2 *filters with a stride of 2. Assume a ReLU activation function.*

(3d) Train the target feature extractor by aligning its extracted features with the features from the source feature extractor using a discriminator. Note, both feature extractors should have an identical architecture and it is recommended to initialize the target feature extractor with the weights of the source feature extractor. The discriminator should consist of two layers with 500 units, each followed by ReLU-nonlinearities, and the final output layer.

Test the learned feature extractor (followed by the source classifier) on the MNIST dataset. Compare the results to the results from Problem (3b) when no domain adaptation is performed.

Disclaimer: As mentioned in the lecture, training generative adversarial networks can be challenging.

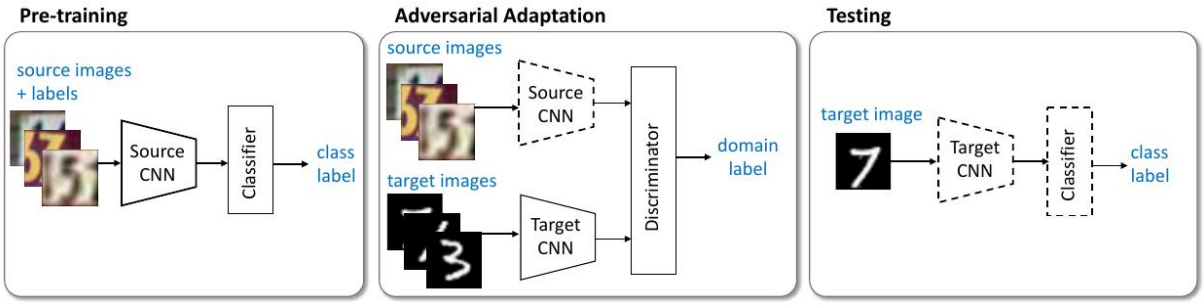

Figure 2: *Domain adaptation framework ADDA (6). The source encoder CNN is first pre-trained using labeled source image examples. Adversarial adaptation is performed by learning a target feature extractor CNN such that a discriminator that sees source features and target features cannot reliably predict their domain label. During testing, features are extracted for target images and classified by the source classifier. Dashed lines indicate fixed network parameters.*

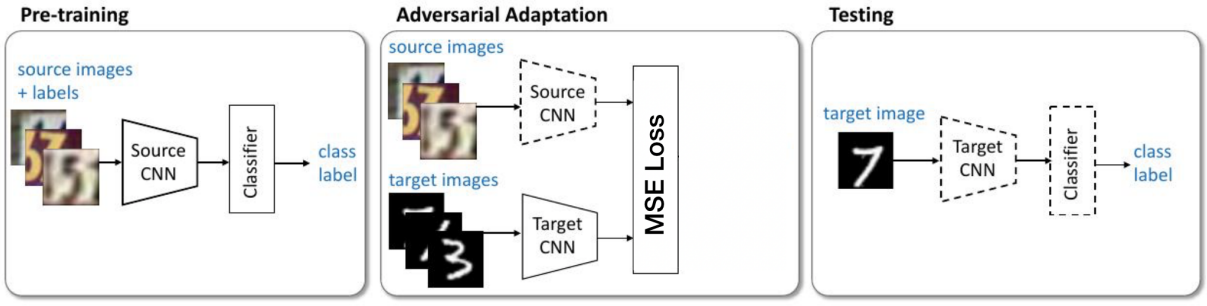

Figure 3: *Modified version of the domain adaptation framework ADDA (6), where the target CNN is trained with a mean squared error loss.*

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[4] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[5] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[6] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.