Assignment 1

Due Date: October 4^{th}, 2023, 10:59:00 pm
Total: 160 marks

General Instructions:

- You are allowed to work directly with one other person to discuss the questions. However, you are still expected to write the solutions/code/report in your own words; i.e. no copying. If you choose to work with someone else, you must indicate this in your assignment submission. For example, on the first line of your report file (after your own name and information, and before starting your answer to Q1), you should have a sentence tha *t, I worked together with my classr* *ave written the solutions/code/rep*
- Your submissi F), with the answers to the specific ation and discussion of your results. o MarkUs directly.
- Submit docum r results separately. Please store al the folder and then submit the file e a **README.txt** file (inside the es.
- Do not worry i your zip file; you can submit multipl

Part I: Theoret

[Question 1] Conv

[1.a] (5 marks) Calculate and plot the correlation **and** the convolution of $x[n]$ and $h[n]$ specified below:

$$x[n] = \begin{cases} 2 & -2 \leq n \leq 4 \\ 0 & \text{otherwise} \end{cases} \qquad h[n] = \begin{cases} 1 & -3 \leq n \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

[1.b] (5 marks) Calculate and plot the correlation **and** the convolution of $x[n]$ and $h[n]$ specified below:

$$x[n] = \begin{cases} 2 & -2 \leq n \leq 4 \\ 0 & \text{otherwise} \end{cases} \qquad h[n] = \begin{cases} 2 - |n| & -3 \leq n \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

[Question 2] Polynomial Multiplication and Convolution (5 marks)

Vectors can be used to represent polynomials. For example, 3^{rd}-degree polynomial $(a_3x^3 + a_2x^2 + a_1x + a_0)$ can by represented by vector $[a_3, a_2, a_1, a_0]$.
If $\mathbf{u}$ and $\mathbf{v}$ are vectors of polynomial coefficients, prove that convolving them is equivalent to multiplying the two polynomials they each represent.

Hint: You need to assume proper zero-padding to support the full-size convolution.

[Question 3] Laplacian Operator (10 marks)

The Laplace operato mensional Euclidean
space, defined as the a twice-differentiable
real-valued function,

where the latter not

Now, consider a 2D i I_{yy}. Here the second
partial derivatives a ables x, y associated
with the image grid n invariant. In other
words, show that ΔI al directions.

Hint: Start by usin (x, y). Then use the
chain rule.

[Question 4] Computational Complexity (5 marks)

Assume that we have a convolution implementation $J = conv2(F, I)$ that takes two images ($F_{k\times k}$ and $I_{n\times n}$) and returns the output in $O(k^2n^2)$ time. Given an image $I_{n\times n}$ and two filters $F_{k\times k}$ and $G_{k\times k}$, we want to compute $G * (F * I)$. Is it more efficient to call $conv2(G, conv2(F, I))$ or $conv2(conv2(G, F), I)$? Briefly justify your answer.

[Question 5] Image Pyramids (10 marks)

In Gaussian pyramids, the image at each level I_k is constructed by blurring the image at the previous level I_{k-1} and downsampling it by a factor of 2. A Laplacian pyramid, on the other hand, consists of the difference between the image at each level (I_k) and the upsampled version of the image in the next level of the Gaussian pyramid (I_{k+1}).

Given an image of size $2^n \times 2^n$ denoted by I_0, and its Laplacian pyramid representation denoted by $L_0, ..., L_{n-1}$, show how we can reconstruct the original image, using the minimum information from the Gaussian pyramid. Specify the minimum information required from the Gaussian pyramid and a closed-form expression for reconstructing I_0.
Hint: The reconstruction follows a recursive process; What is the base case that contains the minimum information?

Hint: Express the output of a network as a function of its inputs and its weights of layers.

[Question 6] Back Propagation (10 marks)

Consider a neural ne

where x_i denotes inp logistic function:

Suppose the loss fun **2** loss, i.e. $L(y, \hat{y}) = (y - \hat{y})^2$. Assume th

$(w_1, w$ $0,\ 0.3)$

[3.a] (5 marks) Dra fine appropriate intermediate variables unction into smaller components.)

[3.b] (5 marks) Given an input data point $(x_1, x_2, x_3, x_4) = (-1.1,\ 1.5,\ -1.5,\ 2.0)$ with true label of 0.0, compute the partial derivative $\frac{\partial L}{w_4}$, by using the back-propagation algorithm. Indicate the partial derivatives of your intermediate variables on the computational graph. Round all your calculations to 4 decimal places.

Hint: For any vector (or scalar) $\mathbf{x}$, we have $\frac{\partial}{\partial \mathbf{x}}(||\mathbf{x}||_2^2) = 2\mathbf{x}$. Also, you do not need to write any code for this question! You can do it by hand.

[Question 7] CNN FLOPs (10 marks)

In this problem, our goal is to estimate the computation overhead of CNNs by counting the FLOPs (floating point operations). Consider a convolutional layer C followed by a max pooling layer P. The input of layer C has 50 channels, each of which is of size 12×12. Layer C has 20 filters, each of which is of size 4×4. The convolution padding is 1 and the stride is

2. Layer P performs max pooling over each of the C's output feature maps, with 3×3 local receptive fields, and stride 1.

Given scalar inputs x_1, x_2, ..., x_n, we assume:

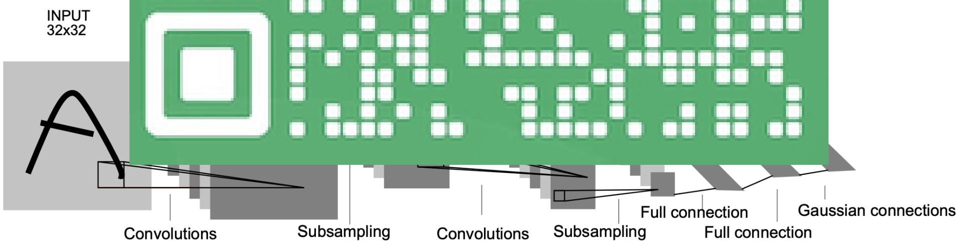
- A scalar multiplication $x_i.x_j$ accounts for one FLOP.
- A scalar addition $x_i + x_j$ accounts for one FLOP.
- A max operation $\max(x_1, x_2, ..., x_n)$ accounts for $n - 1$ FLOPs.
- All other operations do not account for FLOPs.

How many FLOPs layer C and P conduct in total during one forward pass, with and without accounting for bias?

Hint: Find the size [illegible] nsion, first calculate the number of multi[illegible]). For layer P, follow the same procedure.

[Question 8] Train[illegible]

The following CNN [illegible] ctures that was presented in the 90s. C[illegible] this network. Note that the Gaussian c[illegible] fully connected layer similar to $F6$.



[Question 9] Logistic Activation Function (10 marks)

For backpropagation in a node with logistic activation function, show that, in order to compute the gradient, as long as we have the output of the node, there is no need for the input.

Hint: Find the derivative of a neuron's output with respect to its inputs.

Part II: Implementation Tasks (80 marks)

In this question, we train (or fine-tune) a few different neural network models to classify dog breeds. We also investigate their dataset bias and cross-dataset performances. All the tasks should be implemented using Python with a deep learning package of your choice, e.g. PyTorch or TensorFlow.

We use two datasets in this assignment.

1. Stanford Dogs Dataset
2. Dog Breed Images

The Stanford Dogs Dataset (SDD) contains over 20,000 images of 120 different dog breeds. The annotations available for this dataset include class labels (i.e. dog breed name) and bounding boxes. In els. Further, we will only use a small port rain your models on Colab. Dog Breed I s of 10 different dog breeds.

To prepare the data

1- Download bot that appear in both datasets:

- Bernese mount
- Border collie
- Chihuahua
- Golden retriev
- Labrador retri
- Pug
- Siberian husky

2- Delete the folders associated with the remaining dog breeds in both datasets. You can also delete the folders associated with the bounding boxes in the SDD.

3- For the 7 breeds that are present in both datasets, the names might be written slightly differently (e.g. Labrador Retriever vs. Labrador). Manually rename the folders so the names match (e.g. make them both *labrador_retriever*).

4- Rename the folders to indicate that they are subsets of the original datasets (to avoid potential confusion if you later want to use them for another project). For example, *SDDsubset* and *DBIsubset*. Each of these should now contain 7 subfolders (e.g. *border_collie*, *pug*, etc.) and the names should match.

5- Zip the two folders (e.g. *SDDsubset.zip* and *DBIsubset.zip*) and upload them to your Google Drive (if you want to use Google Colab).

You can find sample code working with the SDD on the internet. If you want, you are welcome to look at these examples and use them as your starting code or use code snippets from them. You will need to modify the code as our questions are asking you to do different tasks, which are not the same as the ones in these online examples. But using and copying code snippets from these resources is fine. If you choose to use one of these online examples as your starting code, please acknowledge them in your submission. We also suggest that before starting to modify the starting code, you run them as is on your data (e.g. DBIsubset) to 1) make sure your dataset setup is correct and 2) to make sure you fully understand the starter code before you start modifying it.

Task I - Inspectio

Look at the images any systematic differences between ima

Task II - simple C

Construct a simple the images in DBI. For example, you ca

- convolutional l
- batch normaliz
- convolutional l
- max pooling (2
- convolutional l
- batch normaliz
- convolutional l
- max pooling (2
- dropout (e.g. 0.5)
- fully connected (32)
- dropout (0.5)
- softmax

If you want, you can change these specifications; but if you do so, please specify them in your submission. Use RELU as your activation function, and cross-entropy as your cost function. Train the model with the optimizer of your choice, *e.g.*, SGD, Adam, RMSProp, etc. Use random cropping, random horizontal flipping, random colour jitter, and random rotations for augmentation. Make sure to tune the parameters of your optimizer for getting the best performance on the validation set.

Plot the training, and test accuracy over the first 10 epochs. Note that the accuracy is different from the loss function; the accuracy is defined as the percentage of images classified correctly.

Train the same CNN model again; this time, without dropout. Plot the training and test accuracy over the first 10 epochs; and compare them with the model trained with dropout. Report the impact of dropout on the training and its generalization to the test set.

Task III - ResNet Training on the DBI (15 marks):

[**III.a**] (10 marks) ResNet models were proposed in the "Deep Residual Learning for Image Recognition" paper. These models have had great success in image recognition on benchmark datasets. In this task, we use the ResNet-18 model for the classification of the images in the DBI dataset. To do so, use the ResNet-18 model from PyTorch, modify the input/output layers to match you

do not use the pre-trained ResNet. Plo

compare those with the results of your C

[**III.b**] (5 marks) Ru

report the accuracy. Compare the accura

acy obtained on the SDD. Which is highe

briefly, in one or two sentences.

Task IV - Fine-tu

Similar to the previ

(within torchvision): ResNet18, ResNet34

del of your choosing from torchvision or t

volutional networks and Swin is a transf

need to replace the final layer so the ou

et. **Hint:** The final layer might have a d

This time you are supposed to use the pre-trained models and fine-tune the input/output layers on DBI training data. Report the accuracy of these fine-tuned models on DBI test dataset, and also the entire SDD dataset.

Discuss the cross-dataset performance of these trained models. Which models generalized to the new dataset better? For example, are there cases in which two different models perform equally well on the test portion of the DBI but have significant performance differences when evaluated on the SDD? Are there models for which the performance gap between the SSD and test portion of DBI are very small?

Task V - Dataset detection (15 marks):
Train a model that – instead of classifying dog breeds – can distinguish whether a given image is more likely to belong to SDD or DBI. To do so, first, you need to divide your data

into training and test data (and possibly validation if you need those for tuning the hyperparameters of your model). You need to either reorganize the datasets (to load the images using *torchvision.datasets.ImageFolder*) or write your own data loader function. You can start from a pre-trained model (of your choice) and fine-tune it on the training portion of the dataset. Include your network model specifications in the report, and make sure to include your justifications for that choice. Report your model's accuracy on the test portion of the dataset.

Task VI - How to improve performance on SDD? (10 marks):
If our goal were to have good performance on the SDD dataset, briefly discuss how to work towards this goal in each of the following cases: (you don't need to implement these, just briefly discuss each case in 2-3 sentences)

- At training tim [illegible] of the SDD dataset. All we know is [illegible] with DBI (similar to the answer you [illegible]
- At training tim [illegible] all portion (e.g. 10%) of the SDD da [illegible]
- At training tin [illegible] a small portion (e.g. 10%) of the SI [illegible] set.

Task VII - Discus [illegible]
Briefly discuss how s [illegible] se can have implications in real applica [illegible] xample, consider the case where available [illegible] university) and the goal is to deploy tra [illegible] me).

Summary of impl [illegible]

The train/val/test split is up to you. You can use the same split used in the sample code linked, i.e. train: 60%. validation 10%, test: 30%. But you can do other (reasonable) splits too if you want. Just specify in your report what you did.

	what we want to do	Train	Validation	Test
Task II	dog breed classification	DBI	DBI	DBI
Task III.a	dog breed classification	DBI	DBI	DBI
Task III.b	dog breed classification	DBI	DBI	SDD
Task IV	dog breed classification	DBI	DBI	both
Task V	dataset classification			