

# CS 464

## Introduction Machine Learning

### Spring 2019

### Homework 3

**Due: May 21, 2019 23:59**

## Instructions

- This homework contains both written and programming questions about neural networks. You should implement your programming questions on this notebook. Your plots should also be produced in this notebook. Each programming question has its own cell for your answer. You can implement your code directly in these cells. You should upload this .ipynb file with the computed output cells for each code cell.
- For questions that you need to plot, your plot results have to be included in cell output.
- It is **NOT ALLOWED** to use different libraries than given libraries which are defined in the required packages section.
- It is **NOT ALLOWED** to use a different deep learning framework than PyTorch.
- In this homework, you have to implement programming questions on **Python 3** version programming language.
- While submitting the homework file, please package your notebook(".ipynb") files as a gzipped TAR file or a ZIP file with the name CS464\_HW3\_Firstname\_Lastname. Please do not use any Turkish letters for any of your files including code files and report file. Upload your homework to Moodle.
- Any violation of these rules may lead to significant grade deduction.
- If your code cell does not return outputs correctly, you will lose points for the corresponding question.
- You can send an email for your questions to **can.uner@bilkent.edu.tr** address.
- You can use Google Colab or your own local environment for this homework. You can prefer Google Colab since you have access to GPUs. You do not have to use GPUs to complete this homework though. If you prefer to use your local system, you should follow the below "Environmental Setup" instructions.

## Environment Setup

### Anaconda Installation

- Download anaconda from <https://www.anaconda.com/download>
- Follow the instructions provided in <https://conda.io/docs/user-guide/install/index.html#regular-installation>

## Creation of Virtual Environment

- Create python3.7 virtual environment for your hw3 using follow command from the command line  
> `conda create -n HW3 python=3.7 anaconda`
- Activate your virtual environment  
> `source activate HW3`
- To install auxiliary libraries create "requirements.txt" file which contains given required packages and run following command in the activated "hw3" environment  
> `pip install -r requirements.txt`
- When you create your virtual environment with "anaconda" metapackage, jupyter notebook should be installed. Try:  
> `jupyter notebook`

## Pytorch Installation

You should install PyTorch to your virtual environment which is created for the hw3. Therefore, you should activate your homework virtual environment before to start PyTorch installation.

- > `source activate HW3`

After you have activated the virtual environment, then use one of the following commands to install pytorch for CPU for your system. See <https://pytorch.org/> (<https://pytorch.org/>) for help.

- For MacOS:  
> `conda install pytorch torchvision -c pytorch`
- For Linux:  
> `conda install pytorch-cpu torchvision-cpu -c pytorch`
- For Windows:  
> `conda install pytorch-cpu torchvision-cpu -c pytorch`

If your system has appropriate GPU Driver, CUDA and CuDNN installations, then you can use GPU supported PyTorch as well.

## Required Packages

Additional required packages except PyTorch and Torchvision are defined below.

In [ ]:

```
numpy
pandas
Pillow
matplotlib
scikit-image
scipy
```

## Question 1 - 35 pts

In this question, you are going to implement a Convolutional Neural Network (CNN) to solve the indoor scene recognition problem. The provided subset of [MIT Indoor67 dataset](http://web.mit.edu/torralba/www/indoor.html) (<http://web.mit.edu/torralba/www/indoor.html>) contains images of 10 indoor categories which are bakery, toyshop, dining room, bathroom, library, pool inside, gym, bowling, computer room, hospital room.

In this question, you are going to compare transfer learning and from scratch training approaches.

### Q1.a. Data Loader - 10 pts

We provide you a subset of MIT Indoor dataset. Download our version of MIT Indoor10 dataset from Moodle. You have to implement a custom data loader for the indoor dataset. You can visit [link](https://pytorch.org/tutorials/beginner/data_loading_tutorial.html) ([https://pytorch.org/tutorials/beginner/data\\_loading\\_tutorial.html](https://pytorch.org/tutorials/beginner/data_loading_tutorial.html)) to implement such a data loader. There will be a comparison of training approaches in this question. **Both approaches have to use exactly the same splits (training, test, and validation) to make their performance results comparable.**

1. [Code](#)  
Split the dataset as training, test, and validation sets. For each class, 20% of its samples should be in test set and 10% of its samples should be in the validation set.
2. [Code](#)  
Apply appropriate augmentations to the set(s) for which it is essential.
3. [Written](#)  
Give details about data loading and augmentation steps. How many training, test and validation samples do exist for each class? Do you need to apply augmentations for training, test, and validation sets? Why or why not?

In [1]:

```
# USE THIS CODE CELL TO SPECIFY ROOT PATH FOR THE DATASET.

# IF YOU USE GOOGLE COLAB, DATA PATH EXAMPLE:
# root_dir = '/content/gdrive/My\ Drive/CS464/HW3/indoor_data'
# -----

# IF YOU USE LOCAL ENVIRONMENT, DATA PATH EXAMPLE:
# root_dir = '/home/user/CS464/HW3/indoor_data'
# -----

root_dir = '' # change the value of root_dir variable as your root path
```

In [19]:

```
# USE THIS CODE CELL TO IMPLEMENT YOUR DATA LOADER
```

Answer for Q1.a.3:

USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.

[Double click to this text to write your answer !!!](#)

### Q1.b. Convolutional Neural Network (CNN) - 5 pts

The torchvision package contains of popular datasets, model architectures, and common image transformations for computer vision. In this question, you are going to use one of the well known CNN models which already exist in torchvision. Visit [link \(https://pytorch.org/docs/stable/torchvision/models.html\)](https://pytorch.org/docs/stable/torchvision/models.html) to explore models in torchvision. Remember that using deeper or wider networks may increase your training time.

1. *Code*

Write your own CNN class which is inherited from torch.nn.Module class. This class should have a network that you select among torchvision models as an instance member.

2. *Written*

State the model you selected and explain why you picked that model. Did you need to modify any part of this model to adopt to your task?

*In [8]:*

```
# USE THIS CODE CELL TO DEFINE CNN CLASS
```

**Answer Q1.b.2:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

*Double click to this text to write your answer !!!*

### **Q1.c. Transfer Learning vs Training from Scratch - 13 pts**

In this question, are going to compare two different learning approaches: Transfer learning and Learning from scratch. Thanks to using the already implemented CNN model from torchvision, you can easily convert your randomly initialized network to a pretrained network. Do not forget that you are going to solve a multiclass classification problem. According to that, you need to carefully select the appropriate loss function. You need to select appropriate values for hyper-parameters. You need to use SGD optimizer for this question. Use the training, validation and test splits you created Q1.a. Note that you will NOT perform cross validation. Evaluate your network performance after each epoch by using the validation set. Use matplotlib library for plotting figures.

1. *Code*  
Implement training and validation functions for both strategies. You have to print out average values of training loss, training accuracy, validation loss and validation accuracy for each epoch.
2. *Figure*  
For the from-scratch-trained model, plot training loss and training accuracy of each iteration (update). Also plot validation loss and accuracy at each epoch.
3. *Figure*  
For the transfer learning applied model, plot training loss and training accuracy of each iteration (each update). Also plot validation loss and accuracy at each epoch.
4. *Code*  
Your models are going to run up to the max epoch parameter. According to the validation results, pick the best models for both training approaches. You are going to use these models at the test phase. You need to save them in ".pth" files.
5. *Written*  
Which loss function did you use? Why did you choose it?
6. *Written*  
What are the differences between transfer learning and from scratch learning.
7. *Written*  
Explain how you selected hyperparameters for both strategies. Did you need to make any hyperparameter changes?

In [16]:

```
# USE THIS CODE CELL TO SPECIFY YOUR BEST MODEL PATHS THAT WILL BE USED TO SAVE MODELS.

FROM_SCRATCH_MODEL_PATH = '' # change the value of FROM_SCRATCH_MODEL_PATH variable as your path
TRANSFER_LEARNING_MODEL_PATH = '' # change the value of TRANSFER_LEARNING_MODEL_PATH variable as your path
```

In [32]:

```
# USE THIS CODE CELL TO WRITE TRAINING AND VALIDATION FUNCTIONS WHICH ARE COMMON FOR BOTH LEARNING APPROACH.
# PARAMETERS OF TRAINING CODE MAY DIFFER ACCORDING TO LEARNING APPROACH.

# --- Representative Code Snippet ---

# def train(**kwargs):

# def evaluation(**kwargs):
```

*In [ ]:*

```
# RUN TRAIN FUNCTION FOR THE FROM SCRATCH APPROACH IN THIS CELL.
# PRINT OUT REQUIRED METRICS FOR EACH EPOCH.
# EVALUATE YOUR MODEL AND SAVE THE BEST MODEL ACCORDING TO VALIDATION METRICS.

# --- Representative Code Snippet ---

# from_scratch_hyperparameters = {...}
# for epoch_id in range(max_epoch):
#     train(from_scratch_hyperparameters)
#     val_acc = evaluate(from_scratch_hyperparameters)
#     if val_acc > best_acc:
#         save_model()
```

*In [ ]:*

```
# RUN TRAIN FUNCTION FOR THE TRANSFER LEARNING APPROACH IN THIS CELL.
# PRINT OUT REQUIRED METRICS FOR EACH EPOCH.
# EVALUATE YOUR MODEL AND SAVE THE BEST MODEL ACCORDING TO VALIDATION METRICS.

# --- Representative Code Snippet ---

# transfer_learning_hyperparameters = {...}
# for epoch_id in range(max_epoch):
#     train(transfer_learning_hyperparameters)
#     val_acc = evaluate(transfer_learning_hyperparameters)
#     if val_acc > best_acc:
#         save_model()
```

*In [38]:*

```
# USE THIS CODE CELL TO PLOT METRICS FOR THE FROM SCRATCH LEARNING [Answer: Q1.
c.2]
```

*In [39]:*

```
# USE THIS CODE CELL TO PLOT METRICS FOR THE TRANSFER LEARNING [Answer: Q1.c.3]
```

**Answer for Q1.c.5:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

*Double click to this text to write your answer !!!*

**Answer for Q1.c.6:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

*Double click to this text to write your answer !!!*

**Answer for Q1.c.7:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

[Double click to this text to write your answer !!!](#)

## **Q1.d. Test Classifiers - 7 pts**

You should have obtained two trained networks so far. One of them comes from a transfer learning approach and the other network is from from-scratch-trained approach. In this case, you need to compare these two trained networks by using the test set.

1. [Code](#)  
Load saved '.pth' file for the model that trained with the transfer learning approach. Test this model by using the test set.
2. [Code](#)  
Load saved '.pth' file for the model that trained from scratch. Test this model by using the test set.
3. [Figure](#)  
Provide confusion matrix for the results of [1.] question.
4. [Figure](#)  
Provide confusion matrix for the results of [2.] question.
5. [Figure](#)  
Display randomly selected 20 images from the test set with the predicted and ground truth labels. Show these images in a grid (4x5).
6. [Written](#)  
Give the test set accuracy values for these two models. Which training approach does work better than the other? Explain why?

*In [44]:*

```
# USE THIS CODE CELL TO IMPLEMENT TEST FUNCTION WHICH IS COMMON FOR BOTH MODEL

# --- Representative Code Snippet ---
# def test(**kwargs):
```

*In [12]:*

```
# USE THIS CODE CELL TO LOAD BEST MODEL WHICH USES TRANSFER LEARNING APPROACH
# LOAD TEST SAMPLES
# GET THE TEST RESULTS, PRINT OUT TEST METRICS

# --- Representative Code Snippet ---
# testloader = torch.utils.data.DataLoader(test_dataset, batch_size=n_batch, num
# _workers=n_workers)
# model = YourCNNClass(**kwargs)
# model.load_state_dict(torch.load(TRANSFER_LEARNING_MODEL_PATH))
# model.eval()
# test(model)
```

In [13]:

```
# USE THIS CODE CELL TO LOAD BEST MODEL WHICH IS TRAINED FROM SCRATCH
# LOAD TEST SAMPLES
# GET THE TEST RESULTS, PRINT OUT TEST METRICS

# --- Representative Code Snippet ---
# testloader = torch.utils.data.DataLoader(test_dataset, batch_size=n_batch, num
# _workers=n_workers)
# model = YourCNNClass(**kwargs)
# model.load_state_dict(torch.load(FROM_SCRATCH_MODEL_PATH))
# model.eval()
# test(model)
```

In [14]:

```
# USE THIS CODE CELL TO CREATE CONFUSION MATRIX [Answer: Q1.d.3]
# YOU CAN USE sklearn.metrics package to compute confusion matrix
```

In [15]:

```
# USE THIS CODE CELL TO CREATE CONFUSION MATRIX [Answer: Q1.d.4]
# YOU CAN USE sklearn.metrics package to compute confusion matrix
```

In [58]:

```
# USE THIS CODE CELL TO DISPLAY RANDOMLY SELECTED TEST SAMPLES WITH THEIR PREDIC
TED AND TARGET LABELS [Answer: Q1.d.5]
```

**Answer for Q1.d.6:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

*Double click to this text to write your answer !!!*

## Question 2 - 65 pts

In this question, you are going to implement both a Convolutional Neural Network (CNN) architecture and a Multilayer Perceptron (MLP) to solve the bird type classification problem. As the dataset, you will use [Caltech-UCSD Birds-200-2011](http://www.vision.caltech.edu/visipedia/CUB-200-2011.html) (<http://www.vision.caltech.edu/visipedia/CUB-200-2011.html>) dataset. The dataset consists of 11.788 photos of 200 bird species. The dataset provides you bounding boxes, image labels, and training-test splits. DO NOT FORGET THAT YOU WILL NOT USE TRAINING-TEST SPLIT WHICH IS PROVIDED BY DATASET. YOU WILL SPLIT THE DATASET INTO TRAINING, TEST AND VALIDATION SETS YOURSELF.

You are free to choose your neural network architecture in this question but there will be a minimum performance threshold and your network's performance will affect your grade for this question. The thresholds and metric types are specified in corresponding questions.

**Note: You should be able to achieve these goals with training time < a day on a GPU.**

### Q2.a. Data Loader - 15 pts



In this question you are going to use [Caltech-UCSD Birds-200-2011](http://www.vision.caltech.edu/visipedia/CUB-200-2011.html) (<http://www.vision.caltech.edu/visipedia/CUB-200-2011.html>) dataset. Download the dataset from given [link](http://www.vision.caltech.edu/visipedia/CUB-200-2011.html) (<http://www.vision.caltech.edu/visipedia/CUB-200-2011.html>). You have to implement a custom data loader. Notice that photos are not tightly cropped. You may choose to crop images by using bounding boxes which exist in the dataset. Note that you will build CNN and MLP architectures and input types should be different for these networks. You should prepare data loaders by considering this situation. You may need to convert images to grayscale format to train the MLP model. **MLP and CNN architectures have to use exactly the same splits (training, test, and validation) to make their performance results comparable.**

1. Code  
Split the dataset as training, test, and validation sets. For each class, 20% of its samples should be in test set and 10% of its samples should be in the validation set.
2. Code  
Apply appropriate augmentations to the set(s) for which it is essential.
3. Written  
Explain your augmentation method differences for two models?

In [4]:

```
# USE THIS CODE CELL TO SPECIFY ROOT PATH FOR THE DATASET.

# IF YOU USE GOOGLE COLAB, DATA PATH EXAMPLE:
# root_dir = '/content/gdrive/My\ Drive/CS464/HW3/CUB_200_2011'
# -----

# IF YOU USE LOCAL ENVIRONMENT, DATA PATH EXAMPLE:
# root_dir = '/home/user/CS464/HW3/CUB_200_2011'
# -----

root_dir = '' # change the value of root_dir variable as your root path
```

In [5]:

```
# USE THIS CODE CELL TO IMPLEMENT YOUR DATA LOADER
```

**Answer for Q2.a.3:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

Double click to this text to write your answer !!!

**Q2.b. Multilayer Perceptron - 5 pts**

Implement the MLP model to solve this multiclass classification problem. For the MLP model, **accuracy for the top-5 classes on the test set has to be minimum 50%**. You have to design your network according to this limitation.

1. [Code](#)

Write your own MLP class which is inherited from torch.nn.Module class.

2. [Written](#)

How many layers are there in your network? How did you choose the number of layers? How many neurons are there in each layer? Did you use any activation functions?

[In \[24\]:](#)

```
# USE THIS CODE CELL TO DEFINE MLP CLASS
```

**Answer Q2.b.2:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

[Double click to this text to write your answer !!!](#)

## Q2.c. Convolutional Neural Network (CNN) - 8 pts

Implement the CNN model to solve this multiclass classification problem. For the CNN model, **accuracy for the top-5 classes on the test set has to be minimum 75%**. You have to design your network according to that limitation.

1. [Code](#)

Write your own CNN class which is inherited from torch.nn.Module class.

2. [Written](#)

How many layers are there in your network? How did you choose the number of layers? How did you choose kernel sizes for each convolutional layer? Did you use any activation functions?

[In \[25\]:](#)

```
# USE THIS CODE CELL TO DEFINE CNN CLASS
```

**Answer for Q2.c.2:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

[Double click to this text to write your answer !!!](#)

## Q2.d. Train MLP & CNN Networks - 13 pts

You are going to train MLP and CNN networks by using same training, validation and test samples. You have to solve a multiclass classification problem and you need to carefully select the appropriate loss function. You have minimum performance thresholds for both models. You need to select appropriate values for hyperparameters to achieve these thresholds. You need to use SGD optimizer for this question. So far, you should have created three dataset splits for training, validation and testing. You will need to load these splits at this phase. Note that you will NOT perform cross validation. Evaluate your network performance after each epoch by using the validation set. Use matplotlib library for plotting figures.

1. *Code*  
Implement training and validation functions for both network architectures. You have to print out average values of training loss, training top-1 accuracy (accuracy of the most accurately predicted class), validation loss and validation top-1 accuracy for each epoch.
2. *Figure*  
For the MLP model, plot training loss and training top-1 accuracy of each iteration (each update). Also plot validation loss and validation top-1 accuracy at each epoch.
3. *Figure*  
For the CNN model, plot training loss and training top-1 accuracy of each iteration (each update). Also plot validation loss and validation top-1 accuracy at each epoch.
4. *Code*  
Your models are going to run up to the max epoch parameter. According to validation results pick the best models for both architectures. You are going to use these models at the test phase. You need to save them in a ".pth" files.
5. *Written*  
Which loss function did you use? Why did you choose it?
6. *Written*  
Explain how you selected the hyperparameters of the training phases for both MLP and CNN models.

In [47]:

```
# USE THIS CODE CELL TO SPECIFY YOUR BEST MODEL PATHS THAT WILL BE USED TO SAVE MODELS.

BEST_MLP_PATH = '' # change the value of BEST_MLP_PATH variable as your path
BEST_CNN_PATH = '' # change the value of BEST_CNN_PATH variable as your path
```

In [48]:

```
# USE THIS CODE CELL TO WRITE TRAINING AND VALIDATION FUNCTIONS WHICH ARE COMMON FOR BOTH NETWORK ARCHITECTURES.
# PARAMETERS OF TRAINING CODE MAY DIFFER ACCORDING TO NETWORK ARCHITECTURE.

# --- Representative Code Snippet ---

# def train(**kwargs):

# def evaluation(**kwargs):
```

In [36]:

```
# RUN TRAIN FUNCTION FOR THE MLP MODEL IN THIS CELL.
# PRINT OUT REQUIRED METRICS FOR EACH EPOCH.
# EVALUATE YOUR MODEL AND SAVE THE BEST MODEL ACCORDING TO VALIDATION METRICS.

# --- Representative Code Snippet ---

# mlp_hyperparameters = {...}
# for epoch_id in range(max_epoch):
#     train(mlp_hyperparameters)
#     val_acc = evaluate(mlp_hyperparameters)
#     if val_acc > best_acc:
#         save_model()
```

In [37]:

```
# RUN TRAIN FUNCTION FOR THE CNN MODEL IN THIS CELL.
# PRINT OUT REQUIRED METRICS FOR EACH EPOCH.
# EVALUATE YOUR MODEL AND SAVE THE BEST MODEL ACCORDING TO VALIDATION METRICS.

# --- Representative Code Snippet ---

# cnn_hyperparameters = {...}
# for epoch_id in range(max_epoch):
#     train(cnn_hyperparameters)
#     val_acc = evaluate(cnn_hyperparameters)
#     if val_acc > best_acc:
#         save_model()
```

In [40]:

```
# USE THIS CODE CELL TO PLOT METRICS FOR THE MLP MODEL [Answer: Q2.d.2]
```

In [41]:

```
# USE THIS CODE CELL TO PLOT METRICS FOR THE CNN MODEL [Answer: Q2.d.3]
```

**Answer for Q2.d.5:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

*Double click to this text to write your answer !!!*

**Answer for Q2.d.6:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

*Double click to this text to write your answer !!!*

**Q2.e. Test MLP & CNN Networks - 9 pts**

You should have obtained two trained networks so far. One of them comes from the MLP model and the other is from the CNN model. In this part, you need to compare these two trained networks by using the test set.

1. [Code](#)

Load saved '.pth' file for the MLP model. Test this model by using the test set.

2. [Code](#)

Load saved '.pth' file for the CNN model. Test this model by using the test set.

3. [Figure](#)

Display randomly selected 20 images from the test set with the predicted and ground truth labels. Show these images in a grid (4x5).

4. [Written](#)

Give the top-1 and top-5 accuracy values on the test set for the MLP and the CNN models. Which architecture does work better than the other? Explain why.

In [60]:

```
# USE THIS CODE CELL TO IMPLEMENT TEST FUNCTION WHICH IS COMMON FOR BOTH MODEL

# --- Representative Code Snippet ---
# def test(**kwargs):
```

In [49]:

```
# USE THIS CODE CELL TO LOAD BEST MLP MODEL
# LOAD TEST SAMPLES
# GET THE TEST RESULTS, PRINT OUT TEST METRICS

# --- Representative Code Snippet ---
# testloader = torch.utils.data.DataLoader(test_dataset, batch_size=n_batch, num
_workers=n_workers)
# model = MLP(**kwargs)
# model.load_state_dict(torch.load(BEST_MLP_PATH))
# model.eval()
# test(model)
```

In [54]:

```
# USE THIS CODE CELL TO LOAD BEST CNN MODEL
# LOAD TEST SAMPLES
# GET THE TEST RESULTS, PRINT OUT TEST METRICS

# --- Representative Code Snippet ---
# testloader = torch.utils.data.DataLoader(test_dataset, batch_size=n_batch, num
_workers=n_workers)
# model = CNN(**kwargs)
# model.load_state_dict(torch.load(BEST_CNN_PATH))
# model.eval()
# test(model)
```

In [59]:

```
# USE THIS CODE CELL TO DISPLAY RANDOMLY SELECTED TEST SAMPLES WITH THEIR PREDIC
TED AND TARGET LABELS [Answer: Q2.e.3]
```

**Answer for Q2.e.4:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

[Double click to this text to write your answer !!!](#)

## **Q2.f. Activation and Feature Map Visualization - 15 pts**

You have trained your CNN model. For this trained network, you will visualize the weights and outputs of the randomly selected (at most) 100 filters (feature maps) from the last convolutional layer, as images in a square grid format (10x10). You should select a single image from the test set to display its activation maps extracted from selected 100 filters.

1. *Figure*  
Select an image from the test set and display it.
2. *Figure*  
Display the weights of randomly selected at most 100 filters from the last convolutional layer. Each filter weight should be represented as an image in a square grid.
3. *Figure*  
For the selected sample, compute the activation maps for the 100 filters (from 2nd question). Display each activation map as an image in a square grid (10x10).
4. *Written*  
Explain what these activation maps represent and why they are important.

*In [61]:*

```
# USE THIS CODE CELL TO DISPLAY SELECTED SINGLE SAMPLE FROM TEST SET [Answer: Q2.f.1]
```

*In [1]:*

```
# USE THIS CODE CELL TO DISPLAY WEIGHTS OF THE SELECTED KERNELS [Answer: Q2.f.2]
```

*In [64]:*

```
# USE THIS CODE CELL TO DISPLAY OUTPUTS OF THE SELECTED KERNELS [Answer: Q2.f.3]
```

**Answer for Q2.f.4:**

**USE THIS MARKDOWN CELL TO GIVE YOUR ANSWER FOR THE WRITTEN QUESTION.**

[Double click to this text to write your answer !!!](#)