Homework 3

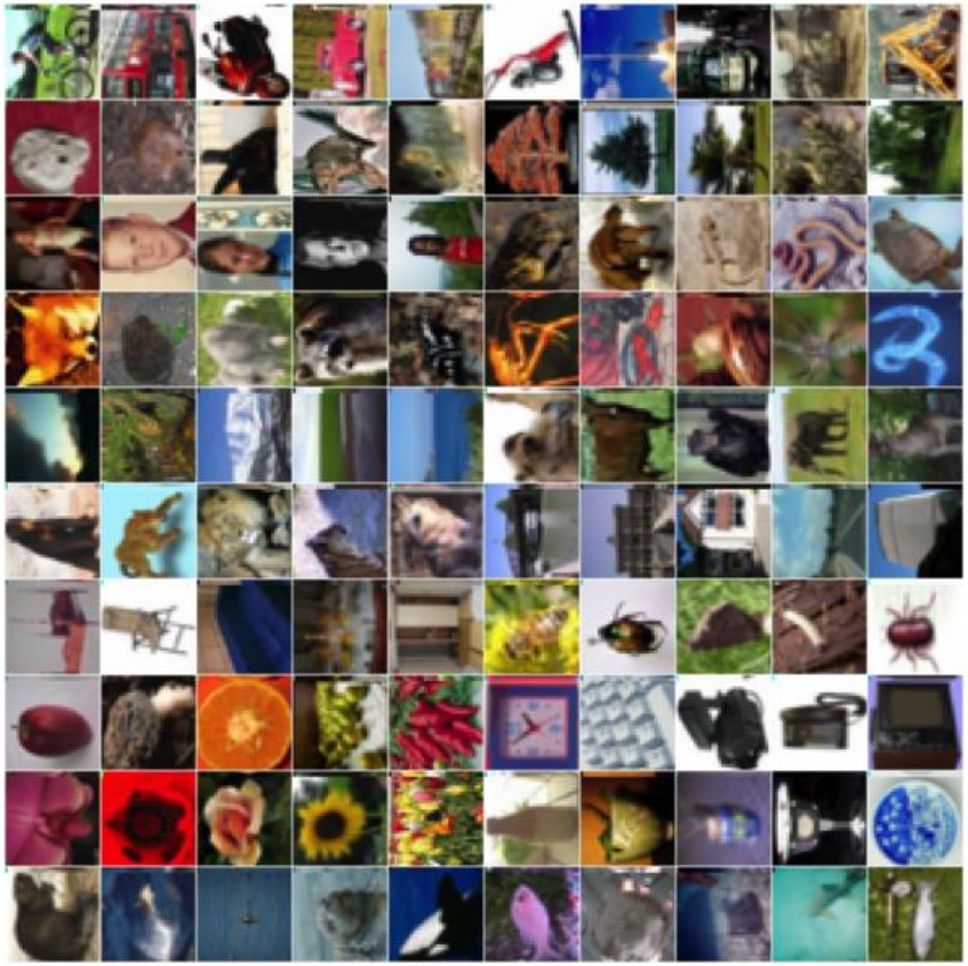
Deep Learning homework

- Going to implement a Convolutional Neural Network on a big image dataset
- You can work with any D.L. framework you like, but...

Deep Learning homework

- Going to implement a Convolutional Neural Network on a big image dataset
- Training and testing on CIFAR 100 dataset, already included in Pytorch (or download separately…)
- You can work with any D.L. framework you like, but…
- I heavily suggest Pytorch ;-)

CIFAR 100



Deep Learning homework

- You are going to start from a pre-defined code that implement a basic NN and CNN, the data loading, the training phase and the evaluation(testing) phase.
- Feel free to use it or to use every code you like
- Submit code to paolo.russo@iit.it with 3 network classes: the traditional NN already included, the CNN trained from scratch that gives you the best accuracy, and the ResNet finetuned model

Deep Learning homework

- Problem: many of you could not have an NVIDIA card and a well done Deep Learning framework setup...how to do the homework?!

Deep Learning homework

- Solution: Google Colab

Deep Learning homework

- A very user friendly python notebook from Google
- You can install python packages, download datasets, plot images, and above all…
- You have a free GPU to do trainings!
- Drawback: you must stay online (can reconnect to the page after 90 minutes max)

Colab

homework3.ipynb

File Edit View Insert Runtime Tools Help

CODE TEXT CELL CELL

```
!pip3 install -q http://download.pytorch.org/whl/cu90/torch-0.4.0-cp36-cp36m-linux_x86_64.whl
!pip3 install torchvision
```

```
[ ]  %matplotlib inline
     import torch
     import torchvision
     from torchvision import models
     import torchvision.transforms as transforms
     from torchvision.transforms import ToPILImage
     import torch.optim as optim

     import torch.nn as nn
     import torch.nn.functional as F

     import matplotlib.pyplot as plt

     import numpy as np

     # function to show an image
     def imshow(img):
         img = img / 2 + 0.5     # unnormalize
         npimg = img.numpy()
         plt.imshow(np.transpose(npimg, (1, 2, 0)))
         plt.show()

     def plot_kernel(model):
         model_weights = model.state_dict()
         fig = plt.figure()
         plt.figure(figsize=(10,10))
         for idx, filt  in enumerate(model_weights['conv1.weight']):
         #print(filt[0, :, :])
             if idx >= 32: continue
             plt.subplot(4,8, idx + 1)
             plt.imshow(filt[0, :, :], cmap="gray")
             plt.axis('off')

         plt.show()

     def plot_kernel_output(model,images):
         fig1 = plt.figure()
```

Deep Learning homework: 1/6

- Train a traditional 2 hidden layers + last FC layer network on the CIFAR 100 train set (layers parameters provided)
- Parameters: 256 batch size, 20 epochs, 32x32 resolution, Adam solver with learning rate 0.0001
- At each epoch calculate and store the accuracy of the current network on the test set
- Plot the training loss and the calculated accuracy curves
- Write your comment about the final accuracy. Did you expect it? Try to make an hypothesis on that behaviour.

Deep Learning homework: 2/6

- Train the simple CNN architecture provided on the CIFAR 100 train set
- Parameters: the same of 1/6.
- At each epoch calculate and store the accuracy of the current network on the test set
- Plot the training loss and the calculated accuracy curves
- Write your comment about the final accuracy. Did you expect it? Why?

Deep Learning homework: 3/6

- Repeat 2/4 step but change the number of convolutional filters from 32/32/32/64 to 128/128/128/256, 256/256/256/512, 512/512/512/1024 (**slow training**)
- Parameters: the same of 1/6.
- Plot the training loss and the calculated accuracy curves
- Write your comment about the final accuracy. Any particular behaviour?Try to make an hypothesis on it. What about computational time?

Deep Learning homework: 4/6

- Start from the network with 128/128/128/256 filters, repeat 2/4 analysis but do the following modifications:
- 4a) Batch Normalization (*every convolutional layer)*
- 4b) BN + FC1 wider (8192 neurons)
- 4c) BN + Dropout 0.5 on FC1 (4096 neurons)
- Parameters: the same of 1/6.
- Plot the training loss and the calculated accuracy curves
- Write your comment about the final accuracy. Any particular behaviour?Try to make an hypothesis on it.

Deep Learning homework: 5/6

- Start from the network with 128/128/128/256 filters, repeat 2/4 analysis but with the following data augmentation: 4a) Random horizontal flipping; 4b) random crop
- Parameters: the same of 1/6. To do random crop, resize to 40x40 and do random crop 32x32.
- Plot the training loss and the calculated accuracy curves
- Write your comment about the final accuracy. Any particular behaviour?Try to make an hypothesis on it.

Deep Learning homework: 6/6

- Load ResNet18 pretrained on ImageNet and finetune it on our CIFAR 100 training set as usual. (**slow training**)
- Parameters: 128 batch size, 10 epochs, 224x224 resolution, Adam solver with learning rate 0.0001.
- Use the best data augmentation schema found in previous step.
- Plot the training loss and the calculated accuracy curves
- Write your comment about the final accuracy. Any particular behaviour?Compare with previous results and try to make an hypothesis on it.

Deep Learning homework: random thoughts

- Adam solver does not need any LR change while training and finetuning, and has been chosen as stabler solution w.r.t plain SGD
- Slow training means 2-4 hours
- Dropout needs to be added in the forward after the fully connected which you wanna affect
- Batch Normalization has one parameters: set it equal the the number of feature maps of the previous layer :)
- Transforms class is very useful to apply transformations and data normalization

Deep Learning homework: random thoughts

- OPTIONAL: some cool functions to plot conv1 kernel values and conv1 output (feature maps). Look at how different they are between a simple network from scratch and the ResNet18 pretrained!
- (Ignore ConnectionResetError: a weird bug of pytorch 0.4)
- A lot of things are not easily explained in D.L: use your knowledge + insight to make hypotheses :)