Exercise Sheet 2

Very Deep Learning WS 2018/2019

Deadline: November 25th, 2018

1 Getting started

1.1 Dependencies

If you want to use your own computer, install Anaconda3 or pure Python 3.6 and following packages which will be necessary to solve this and the following exercise sheets:

- numpy
- PyTorch
- matplotlib
- jupyter

Alternatively, you can use the MindGarage computers which should have all required packages installed.

1.2 Loading the exercise

Clone the repository : *https://github.com/kumar-shridhar/Very-Deep-Learning-CNN*
Start a Jupyter Notebook. In Linux, this is done by typing *jupyter notebook* in a shell. Navigate to *CNN.ipynb* and start it. The basic code to understand and run the notebook is already present there. Start attempting the exercise only when you are done executing the notebook once.

2 Questions

2.1 Get familiarized with Convolutions

Apply what you have learned about the convolution operation in the sheet: *conv-example.xlsx*. An image of size 28*28 is present with two filters of 3*3. Apply the convolution operation and the results are in the next cells. Further, max pooling and dense layers are present. There are empty cells in the sheet marked in yellow. Fill those values. See how the values change the prediction in the end cell. Feel free to play around with filter values.

- Min Submission: Complete sheet with no values missing

2.2 Backpropagation

Learn the concept of Backpropagation from *backprop.py* file. For the lines (line 35,39,42,43), write the missing expression for it. Understand how the updation of gradient is made and the intuition behind it.

- Min Submission: Updated backprop.py file with no missing equations

2.3 Playing around with datasets

Run the same code with *CIFAR10* and *Dog Breed Datasets* (Download the dataset from here: *https://www.kaggle.com/c/dog-breed-identification/data*). Try using the Kaggle API for downloading it. Write the code in the *dataset.py* file and pass *'dog-breed'* as an argument from *CNN.ipynb* file. Train the model and submit the results on Kaggle (https://www.kaggle.com/c/dog-breed-identification) and check your standings.
Tip: Create a val set and run all experiments on it before submitting the results online.

- Min Submission: Kaggle submission screenshot with the TeamName as Kaggle user name.

2.4 Improvement

Improve the results with augmenting the dataset. For CIFAR-10 check all augmentation techniques listed in the *autoaugment.py* file. Create set of augmentations for *Dog Breed* dataset. Compare the results with the previous section.
Now add more layers in the ALexNet model (Conv and Linear). See the effects of adding more layers. Run the augmented dataset with *ResNet* architecture. Compare the training and val accuracies and verify if the model is underfitting. If yes, what is the reason for it?

- Min Submission: Augmentation parameters for Dog Breeds updated in autoaugment.py file. Answer to theory question about underfitting.

2.5 Overfitting

When using a new architecture, can you see if the model is overfitting? What are the different ways to solve the issue? *Hint: Regularization and Batch Norm.*
Play around with transfer learning techniques (Use a pre-trained model on ImageNet and fine tune it over Dog Breed dataset.) Compare the results. Find the reasons why the results are pretty comparable.

- Min Submission: Answer to theory question about overfitting. Comparison of results with normal approach to transfer learning.

2.6 Advanced Users (*Optional*)

Try to move up the rankings by playing around with different *architectures*, *Learning rates*, *data augmentation* methods and *ensembling* techniques. Feel free to experiment.
Read about other Image Classification approaches : CapsNet, Bayesian CNN.

- https://arxiv.org/abs/1710.09829
- https://arxiv.org/abs/1506.02158

2.7 Submission instructions

Write all the theory answers in a text file with section number mention (like 2.4 for answer to underfitting question). Kaggle names should be same as GroupName. No individual submission on Kaggle is needed. Team submission will work. Also, on OLAT please submit one solution per team. No individual submission needed. Best Kaggle Rank team will get a chance to present their solution in the Exercise Class and a surprise.