Exercise 3.2 Deep learning

e12045110 Maria de Ronde
e12040873 Quentin Andre
e11921655 Fabian Holzberger

July 21, 2021

Datasets

For exercise 3.2 Deep Learning we decided to apply deep learning on image classification. The data sets that we will use are CIFAR-10 [INCLUDE REFERENCE] and Tiny-ImagenNet[Include REFERENCE. With these two datasets we have variation in the classes represented in the data. This enables us to explore the difference in performance when the number of classes increase. In sections and both datasets are described in more detail.

CIFAR-10

CIFAR-10 is a dataset which consists of 60000 images, of which 50000 training images and 10000 test images. Each image has 32x32 colored pixels. There are 10 different classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck) each class has exactly 5000 images in the training data and 1000 images in the test data.Each image only belongs to one class. There are no multi-label images.

Tiny ImageNet

Tiny ImageNet is a dataset containing of 100000 training images, divided in 200 different classes. There are 500 images per class in the training data. Next to the training data there are 10000 testing and 10000 validation images as well. each picture has 64x64 pixels.

Traditional classifier

In order to have a baseline for our deep classifier some traditional classifiers have been executed. The following traditional classifiers have been trained:

1. **Multinomial Naive Bayes**: alpha = 1.0, fit_prior= True, class_prior= None
2. **Random forest**: n_estimators = 100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes = None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, cc_alpha=0.0, max_samples=None
3. **Single layer perceptron**: penalty=None, alpha=0.0001, l1_ratio=0.15, fit_intercept=True, max_iter=1000, tol=0.001, shuffle=True, verbose=0, eta0=1.0, n_jobs=None, random_state=0,

early_stopping=False, validation_fraction=0.1, n_iter_no_change=5, class_weight=None, warm_start=False

4. **Multi layer perceptron**: 2 Relu activation layers 256, 1 softmax activation 10 epochs=15, batch_size=32, verbose=0

Before we could train the traditional classifiers, we extracted features from our images. We performed two type of feature extraction.

1. Color histogram
2. SIFT

Color histograms

Color histograms is one of the simplest feature extraction method for images. It counts the frequency of pixels with a certain color. The bins are based on the RBG coding. Spatial information get lost completely during this feature extraction.
In FIGURE REF!!! a color histogram for both datasets is given.
We created 4 different datasets, two based on one dimensional histograms (256 bins per channel and 64 bins per channel), one on two dimensional histograms (16 bins per channel) and one of 3 dimensional histograms (8 bins per channel). This based on the example shown in simple-image-feature-extraction INCLUDE REFERENCE https://tuwel.tuwien.ac.at/course/view.php?id=35929 !!!. The color histograms have been created using OpenCV.

Sift back of visual words

First the images are converted into grey scale images. With use of SIFT the keypoints are detected. Afterwards 20 clusters are created with use of Kmeans.
Visual words are created and vectorized in a histogram (frequency of visual words).
Scale the histogram
Classify

Results

Deep learning

Res50Net

The first architecture we used for deep learning is the Res50Net [1]. The ResNet is pre-trained and transfer learning is applied. A front layer is added in order to have the correct input size. We froze the first 169 layers and re-trained the last XXXX layers. Re-training has been done in XXXX epochs.

Results

squeezNet

For the second architecture we decided on unsing SqueezeNet [2]. Squeezenet claims to have an accuracy equal to AlexNet but has a lot less parameters to learn. For SqeeuzeNet no weights are available, therefore we recreated the model and re-trained it.

Results

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing ren, Jian Sun, *Deep Residual Learning for Image Recognition* (2015b). arXiv:1512.03385.

[2] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Kreutzer. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡0.5MB model size*,(2016) arXiv:1602.07360.