Image classification on the cifar10 dataset using Tensorflow”
Working Paper for CS4487 Machine Learning

BOHNSTEDT Timo , LÖHR Tim

***Abstract*—Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam ac enim vulputate ipsum pellentesque bibendum imperdiet id nibh. Mauris non varius odio. Pellentesque eu libero porta, porttitor lectus ut, dictum neque. Curabitur maximus, justo non faucibus tristique, tellus turpis ornare elit, et sagittis dui justo convallis ex. Pellentesque a libero dui. In vel lobortis nunc. dui. Vivamus congue nulla.**

***Index Terms*—cifar10, machine learning, data science, image classification**

1. Problem Description

TIn our course Machine Learning at the City University of Hong Kong, Dr Kede teched us the fundamental mathematical knowledge to solve machine learning tasks. Furthermore, we improved our ability to use this knowledge while working on Jupyter Notebook Tutorials, which were provided by Dr Kede and his assistant PhD students. To proof our learning progress in the theoretical and practical field, we are going to work on a Group Project. To solve an image classification task, we use the widely used dataset ”cifar10”. The original dataset consists of 60000 coloured images of objects from 10 classes, with 6000 images per category. There are 50, 000 training images and 10, 000 test images. To compare our work with other groups, we are using the following evaluation criteria:

$$Acc(f, D) = \frac{1}{m} \sum_{m}^{i=1} \mathbb{I} \left[y^{(i)} = f(x^{()i)}) \right] \quad (1)$$

We can choose the Tensorflow Framework from Google and the Facebook pendant Pytorch. To have a quick start, we got two tutorials which are focusing on bothe frameworks and how to use them while solving an image classification task. Because of the broader Community and after a first evaluation based on the provided evaluation criteria, the group decided to use Tensorflow instead of Pytorch.

2. Literature Survey

3. Technical Details

3.1 Preprocessing

3.1.a Data Augmentation: Data augmentation describes the method of creating new data points by transforming the original data. If we look at images, this can be done for example by resizing, rotating, shifting, flipping and more. With this process we can artificially increase the amount of data, eventhough we have limited access to only 50000 training data. This leads to a improved performance for our VGG model.
We use the *ImageDataGenerator class* built-in function from Keras itself to perform this transformations.

3.1.b Normalization: The goal of normalization is to make every datapoint have the same scale so each feature is equally important. There are many possible ways to scale data e.g. MinMax-Scaling or Z-Score and all of them follow certain purposes. The MinMax-Scaling suffers from the so called *outlier issue*. We decided to choose the Z-Scoring, because the Z-score normalization is a strategy of normalizing data that avoids this outlier issue [1].

3.1.c Categorical Output:

3.2 Model

3.2.a Convolutional Layers:
3.2.b Kernel Regularization:
3.2.c Batch Normalization:
3.2.d Activation Functions:
3.2.e Max Pooling:
3.2.f Flatten:
3.2.g Dense Layer:

4. Result Analysis

Proin scelerisque odio dolor, id volutpat lacus hendrerit nec. Nam sit amet consectetur libero. Mauris semper maximus dui, vitae maximus neque rutrum et. Integer odio arcu, porta quis consectetur non, consequat non ex. Donec commodo varius odio. Pellentesque facilisis diam et pretium sollicitudin. Nam fringilla nisi in est accumsan accumsan. Phasellus erat metus, sollicitudin quis dolor vitae, ullamcorper molestie urna. Nulla faucibus lacus nibh, eget fermentum enim luctus sit amet.

In rhoncus ligula vel magna malesuada iaculis. Donec hendrerit non ipsum ac tempus. Praesent sit amet aliquam lacus. Fusce vulputate tempor lacus, eget fermentum ex laoreet id. Morbi congue cursus ligula ut pretium. Sed quis velit quis justo elementum tincidunt nec nec mauris. Aliquam sed ipsum non lorem placerat vestibulum quis sed arcu. Mauris eu lorem dui. Vivamus congue nulla.result analysis

References

[1] T. a. Dr.A.Santhakumaran, “Statistical normalization and back propagation for classification,” 2011. [Online]. Available: https://pdfs.semanticscholar.org/4cbd/5fcd6081cfcd16e9111f1bcc17b6e283d439.pdf