Project proposal: Recognızıng dıgıts

Duygu Can

MEF BDA’18

# proposal

The goal of the project is to correctly identify digits from a dataset of tens of thousands of handwritten images. Different machine learning algorithms will be studied on the toy dataset to learn from first-hand what works well and how techniques compare. Later on, the most successful algorithm can further be applied to a real life image dataset.

# Methodology

In this project both classification methods such as SVM and K-nearest neighbors will be studied. Dimensionality reduction will be provided with PCA. Later on performance of those classification methods will be compared by simple convolutional neural networks (CNN). CNN model will be developed with the help of Keras library working on Tensorflow framework. Throughout the project data augmentation techniques will be employed and several regularization methods such as *dropout* and *batch normalization* will be used extensively.

# Data

* The MNIST (Modified National Institute of Standards and Technology) database is a large and mixed dataset composed of hand-written digits by high school students and US Census Bureu. It is composed of 70000 images (60000 for training and 10000 for test). This data set is already available in Scikit-Learn library with the same structure of the creators LeCun et al. [2]. The dataset is made available under a [Creative Commons Attribution-Share Alike 3.0 license](https://creativecommons.org/licenses/by-sa/3.0/).
* Depending on the progress of the project, the data set can be enlarged to real life data of images.

# references

1. [MNIST database](https://en.wikipedia.org/wiki/MNIST_database)

2. Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.".

3. [THE MNIST DATABASE of handwritten digits](http://yann.lecun.com/exdb/mnist/)

4. [Introduction to CNN Keras - 0.997 (top 6%)](https://www.kaggle.com/yassineghouzam/introduction-to-cnn-keras-0-997-top-6)

5. [Neural Networks and Deep Learning](http://neuralnetworksanddeeplearning.com/index.html)

6. [A Beginner's Guide To Understanding Convolutional Neural Networks](https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/)

7. [Visual Cortex Cell Recording](https://www.youtube.com/watch?v=Cw5PKV9Rj3o)

8. Xavier Glorot, Antoine Bordes and Yoshua Bengio (2011). [Deep sparse rectifier neural networks](http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf). AISTATS.

9. [Usage of Initializers](https://keras.io/initializers/#truncatednormal)

10. [Dropout: A Simple Way to Prevent Neural Networks from Overfitting](https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf)

11. [Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift](https://arxiv.org/pdf/1502.03167v3.pdf)

12. [Adam: A Method for Stochastic Optimization](https://arxiv.org/abs/1412.6980)

13. [An overview of gradient descent optimization algorithms](http://ruder.io/optimizing-gradient-descent/index.html#conclusion)

14. [keras/keras/optimizers.py](https://github.com/keras-team/keras/blob/master/keras/optimizers.py)

15. [Welcome to deep learning (CNN 99%)](https://www.kaggle.com/toregil/welcome-to-deep-learning-cnn-99)

16. [How can I explain the fact that test accuracy is much higher than train accuracy?](https://www.quora.com/How-can-I-explain-the-fact-that-test-accuracy-is-much-higher-than-train-accuracy)

17. [Why is the training loss much higher than the testing loss?](https://keras.io/getting-started/faq/#why-is-the-training-loss-much-higher-than-the-testing-loss)

18. [How can I obtain reproducible results using Keras during development?](https://keras.io/getting-started/faq/#how-can-i-obtain-reproducible-results-using-keras-during-development)