Biomedical Data Science & AI

Exercise sheet 9

SoSe2021

Prof. Dr. Holger Fröhlich, Mohamed Aborageh, Vinay Bharadhwaj, Yasamin Salimi

Due date: June 29th

Questions

Exercise 1 - Basics of NN (9 points)

From the MNIST database load the handwritten digits dataset.

1. Normalize your dataset before training your model. **(1 point)**
2. Train a neural network once using Adam and once using AdaGrad optimizer. **Hint:** Set epochs = 20, neurons of hidden layer = 100, activation function = ReLU for reproducibility. **(2 points)**
3. Plot the SparseCategoricalCrossentropy loss for both models. Plot the computed accuracy for both models. Which model performed better while training? **(2 points)**
4. Compute the model accuracy on the test set for both optimizers. Which model performed better? **(1 point)**
5. Familiarize yourself with Layer Normalization and explain how it works. **(1 point)**
6. Using the same dataset to train a neural network with Layer Normalization. **Hint:** Set epochs = 20, neurons of hidden layer = 100, activation function = ReLU for reproducibility.
    a. Compute the SparseCategoricalCrossentropy loss and model accuracy. **(1 point)**
    b. Evaluate the model performance using the test dataset. **(1 point)**

Exercise 2 - Hyper Parameter Optimization (9 points)

1. What are the main challenges with hyper-parameter optimization for neural networks? **(1 point)**
2. Inform yourself about variants of Bayesian-HPO and explain them in detail **(2 points)**
3. Using the same MNIST dataset, optimize the activation function for the output layer and the number of dropout units in the NN model using the following methods. **(6 points)**
    a. Grid search
    b. Random search
    c. Bayesian Hyper-parameter optimization

Exercise 3 - Transfer Learning & CNNs (7 points)

1. Load the VGG16 pre-trained model using Keras Applications API. Use the model to classify the dog images in *canines.zip* after pre-processing each image by doing the following: **(2 points)**
    a. Load each image and set the size to 224 x 224 pixels
    b. Convert the image pixels to a numpy array and reshape it according to the model's input requirements
    c. Use the model to print out the predicted class and its probability for each image
2. Downscale the given matrix by applying the following pooling operations:
    a. Max Pool **(1 point)**
    b. Average Pool **(1 point)**

1	4	1	5
4	9	4	8
4	5	4	3
6	5	7	4

3. Load the CIFAR10 dataset using Keras datasets API and normalize the images' pixel values. Train a convolutional neural network to classify the dataset images with the following architecture: **(3 points)**
    a. Convolutional Base:
        i. An input convolution layer with 32 filters and a kernel size of (3,3). Adjust your input shape to that of the CIFAR images' format
        ii. 2 convolution layers, each with 64 filters and a kernel size of (3,3)
        iii. 2 Max Pool layers, with a pool size of 2x2
    b. 2 dense layers, with 64 and 10 units respectively. Adjust the output of the convolutional base such that it satisfies the input requirements of the dense layers.
    c. Use the following parameters to train the network:
        i. Sparse categorical cross entropy as your loss function
        ii. Adam optimizer
        iii. 10 epochs
        iv. ReLU activation for your layers

    Compile your model, then plot the accuracy across each epoch.