[S24] Introduction to Machine Learning. Bonus assignment

March 2024

Introduction

In this assignment you are going to experiment with different learning approaches on CIFAR-10 dataset. Labeling of even such a small dataset is a time and money demanding process. Therefore, most of the times you will be forced to obtain a reasonably good results on a very small portion of the data.

In this assignment you will be introduced to several approaches:

1. Self-supervised learning with the use of pre-trained autoencoder;
2. Multi-task learning - you will use input reconstruction task as a secondary task to improve the performance of the model on a major task;
3. Finally, you will implement the simplest ensemble of the neural network to see how weak models perform together.

In each task we provide you with only a general idea of the approach you are supposed to use. Further, we provide you with a suggested reference materials. Feel free to consult other sources. We give you a freedom to choose the appropriate hyper-parameters as well as model architecture, pre-processing steps and etc.

Each task is graded in all-or-nothing fashion. So do your best to complete all prescribed steps. If you miss one of those - you get 0 for entire task.

HINT: make sure that you read an entire assignment before starting solving it so that you can plan the software architecture in advance and avoid the necessity to rewrite it from scratch or duplicate too much code.

Keep your code neat and tidy and make sure that you have necessary comments. For each task keep only the best model.

Expected learning outcomes:

1. Learn how to train the model with a small labeled dataset;
2. Understand how to organize an ensemble of neural networks.

Task 1. Baseline model [0pt]

Find out the way to save and load your CNN model from the second task of the main part of the assignment. We are going to use this model as one of the classifiers in the ensemble in the Task 4. Refer to a used framework documentation to find the most appropriate way.

Task 2. Self-supervised learning [12pts]

In this task you are going to use an Autoencoder - model that learns to predict its input.

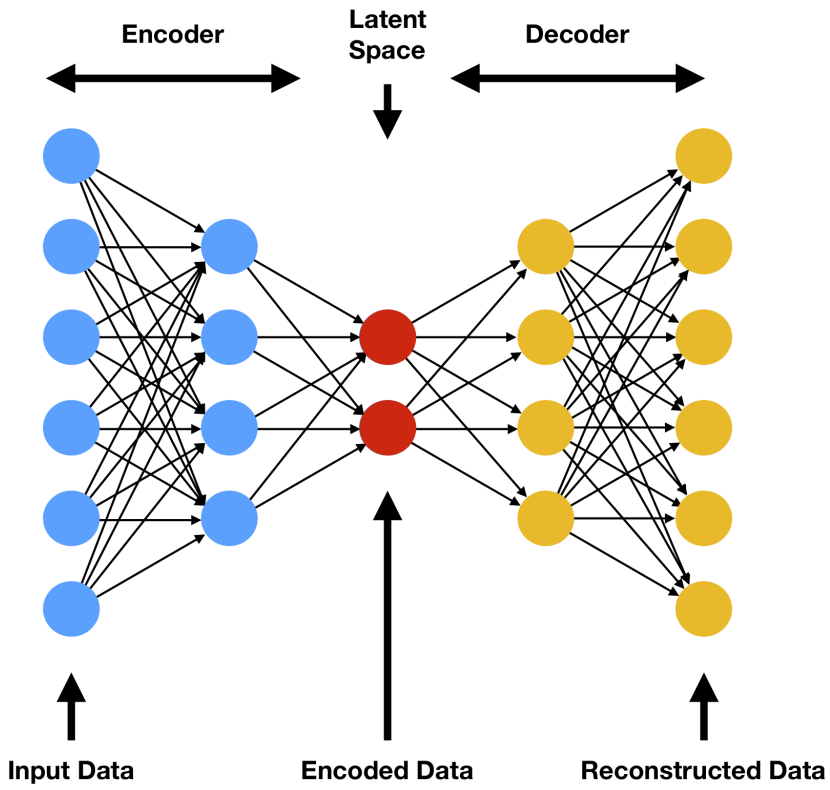

Figure 1: Autoencoder architecture

On a Figure 1 you can see a basic Autoencoder model. As you could see it has two parts: encoder and a decoder. The vector z is a vector of a so-called *latent* features. The goal of the autoencoder is to find a latent features that allow it to reconstruct the input image. In a learning process we aim to

minimize the distance between the input image and its reconstructed version - and there are many possible distance metric which you could try to use.

Refer to the Chapter 15 of the classical "Deep Learning" book by Ian Goodfellow, Yoshua Bengio, and Aaron Courville for a detailed description of autoencoders.

This task has two major steps:

1. Train an Autoencoder model on a whole CIFAR-10 dataset. Don't use labels.
2. Take the encoder part of the autoencoder, fix its weights (make them non-trainable) and attach a classification head. Train this model on a 10% sample of CIFAR-10 dataset stratified on a class label.

Make sure that you perform model evaluation on an entire test set.

Plot train and test losses of your model during the training as well as train and test accuracies.

Task 3. Auxiliary learning [12 pts]

In this part you will try to implement a neural network that solves two problems simultaneously: classification and image reconstruction. Use the same sample of the data as in the 2nd task to train that model.

In your solution:

1. Define an appropriate architecture
2. Select appropriate loss functions for both tasks (classification and reconstruction) and find out a way to efficiently combine them.
3. Train your network on a data sample described in Task 2

Make sure that you perform model evaluation on an entire test set.

Plot train and test losses (classification, reconstruction and combined) of your model during the training as well as train and test accuracies in a classification problem.

Refer to `https://link.springer.com/chapter/10.1007/978-3-319-46493-0_36` for more details on an approach.

Task 4. Ensemble [6pts]

Implement stacking ensemble. Don't forget to freeze parameters of base models. Use the following 3 models: baseline model, the one you trained using auxiliary learning approach and the one you trained in a self-supervised fashion. You can find an intuitive description of stacking (as well as of other ensemble methods) in the following blog-post: `https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205`

Make sure that you perform model evaluation on an entire test set.

Plot train and test losses of your model during the training as well as train and test accuracies.

Summarize the performance of all four models (baseline, self-supervised, auxiliary learning, and ensemble) in a table and try to reason about efficiency of used techniques in terms of:

1. Achieved accuracy
2. Used number of labeled data
3. Used number of unlabeled data
4. Total size of the model (you might use a torchsummary package for that)
5. Inference time - average time taken to classify single image from a testing set

Submission

You should submit a single IPython notebook (.ipynb) for all tasks of the bonus part of the assignment.. Make sure that the outputs of all cells are preserved.