Report A3 data challenge kaggle MVA 2023 (sketch images Classification)

Anonymous CVPR submission

Paper ID *****

Abstract

The goal of this challenge is to classify data sketches in a test (mistery) set with the best possible score (accuracy between 0 and 1) on Kaggle, a well-known data challenge competition website within the community of novices and experts in ML. Thus, we focus ourselves on a vision classification task. In order to achieve our goal, we have a github repository at our disposal with principal python scripts : the architecture of a basic CNN named "model.py" (and "model factory.py") along with train, test and validation dataset of sketches, a code to pre-process our data before entering in the NN using "data.py" . Additionally, we have a "main.py" script and an"evaluate.py" script to train our neural network architecture and evaluate it with the test set generating a csv file deposited on Kaggle.Afterwards, it affects us a score based on 30 percent of the mystery set for the initial evaluation (public section) and then a score based on the remaining 70 percent of the same set in an another evaluation. From my perspective it is a very interesting challenge because we study and learn a lot of stuff. For instance I have learned how to build an effective strategy to overcome the "poor" performances of a model in ML. In this one-page report I will discuss briefly what choices I have made to achieve a pretty high level accuracy for such a non-trivial classification !

1. Approach

1.1. Forward propagation

1.1.1 Choice of the model

I used the pre-trained model DeiT (Data efficient Transformer) distilled with patch of 16 and 384 x 384 resolution of images in the library timm. I opted not to use a CNN for this specific task due to recent advancements.Vision Transformers (ViTs) have demonstrated excellent accuracies in image classification tasks. Currently, Data-efficient Transformers (DeiTs) particularly variants like DeiT distilled, represent a significant improvement over ViTs according to the literature. The concept behind DeiT distilled involves transferring information from a 'teacher model' to a 'student model,' which is computationally less expensive. This approach provides a solid foundation for tackling the challenge. In addition to the pretrained model, I introduced an extra linear layer to produce an output of nclasses=250. This is followed by a dropout layer (0.5) to prevent overfitting with the data. Above all, I selected 9 epochs (it takes some time to achieve the desired precision, especially with data augmentation) and a batch size of 16 for computational efficiency.

1.1.2 Pre-processing data

Accessing the data emphasized the need for crucial data augmentation. For the training dataset, I applied a random resized crop (size=384, scale=(0.9,1.0)) followed by normalization. The validation dataset underwent simpler preprocessing, including resizing and normalization. Both datasets were transformed into tensors.

1.2. Backward propagation

Stochastic Gradient Descent (SGD) with Momentum and Weight Decay (W-D) was employed to add some regularization.

1.2.1 Parametrization of the learning rate , the momentum and the W-D

I selected a learning rate of 0.0095, a momentum of 0.85, and a weight decay of 0.0005. This specific parametrization facilitates the model's convergence during backward propagation.

2. Results

The model, with the specified parameters/hyperparameters, achieved 98 percent accuracy on the training set and 81 percent on the validation set over the course of 9 epochs. This is a highly relevant performance for the task. I ensured that the accuracy consistently increased on both sets during each epoch to mitigate overfitting.

References