Image classification with CNNs

David Štych
Aleksandra Jamróz

October 25, 2022

Baseline performance

Improvements

- We have used data augumentation to randomize training data to reduce overfitting and achieve better performance [**?**]

  To the training dataset we applied following transformations:

  - Random rotation of the image by $\pm 30°$
    torchvision: RandomRotation
  - Crop a random portion of image and resize it to a given size
    torchvision: RandomResizedCrop[**?**]
  - Randomly flipping the image horizontaly with probability of 50%
    torchvision: RandomHorizontalFlip[**?**]
  - Converting to PyTorch tensor and normalization

  To the validation and test dataset we applied following transformations:

  - Resize the image to specific size
    torchvision: Resize[**?**]
  - Crop the image at the center to a specific size
    torchvision: CenterCrop[**?**]
  - Converting to PyTorch tensor and normalization

- Class balancing

Label	**Number of samples in the dataset**
nevus	1372
melanoma	374
keratosis	254

  Table above shows sample distributuion in the dataset. Clearly nevus is much more prevelant in the dataset, which is what we would expect as it is more common medical problem. [**?**] Nevertheless, we used class weights as an argument in the loss function and it lead to an improvement on the validation dataset. For each class we calculated weight with the following formula:

$$\text{Class weight} = 1 - \frac{\text{Number of samples in the dataset}}{\text{Total number of samples}}$$

- We have used Adam optimizer resultin in faster learning.
- We changed *maxSize* argument to 0 meaning we have used the entire dataset. This was possible thanks to Czech Technical University in Prague (home university of one the authors of this report), which allowed us to use their computational resources. [**?**]