Carlo Cena

Homework 3

Machine Learning and AI
January 10, 2020

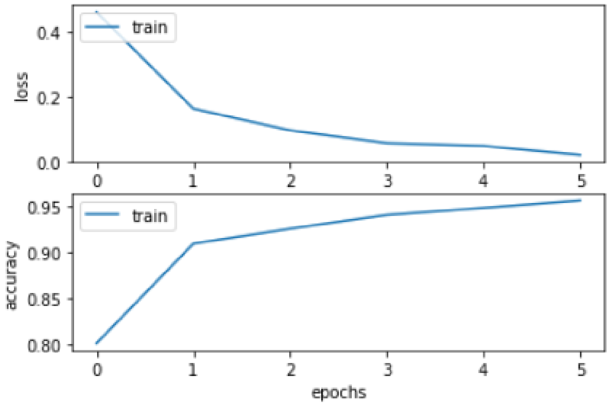


Figure 1: Train loss and accuracy without adaptation, learning rate 8e-3, batch size 64.

Introduction

In this homework we are going to implement DANN, a domain adaptation algorithm, on PACS dataset with AlexNet.

PAC is made of 7 classes and 4 domains, Photo, Art painting, Cartoon, Sketch.
We will train the network on photo and then test it on a classification task on art paintings.

Training and testing without adaptation

Initially, we're going to train the network without adaptation, therefore we won't use the discriminator.

Figure1 represents loss and accuracy for each epoch.

The accuracy reaches 98% on the photo domain in only a few epochs, because we are using pretrained weights of AlexNet.

I then tested the network on a different domain, art paintings, and it reached an accuracy between 43.9 and 44.5%.

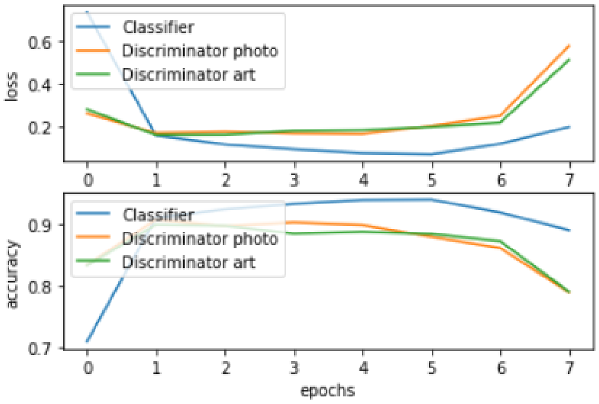


Figure 2: Train loss and accuracy without adaptation, 5e-4 learning rate and batch size 64, after 8 epochs, alpha=0.5

Training and testing with adaptation

As in the previous section I trained the network on the photo domain and then tested it on the art paintings' domain.

However, this time, I used adaption, training results are shown in Figure2.
During training the model reaches an accuracy of 97% on the photo domain and the discriminator is able to classify the domain of an image with an accuracy of 88% for photos and 89% for art paintings. Test accuracy on art paintings is around 46.3%.

Validation for domain adaptation

Validation for domain adaptation is an open problem, in this homework we are going to tune the hyper-parameters by performing domain adaptation in two different settings: photo to cartoon and photo to sketch.

Validation without adaptation

Initially I performed hyper-parameters tuning without adaptation; I tuned learning rate and batch size, starting from values uniformly distributed between 10^{-5} and 10^{5}, for the first one, and 32, 64, 128 or 256 for the batch size.
The results for this initial search are shown in Figure3.

For the finer search the learning rate was drawn from a uniform distribution U(10^{-5},10^{5}), this led to the losses represented in Figure4.

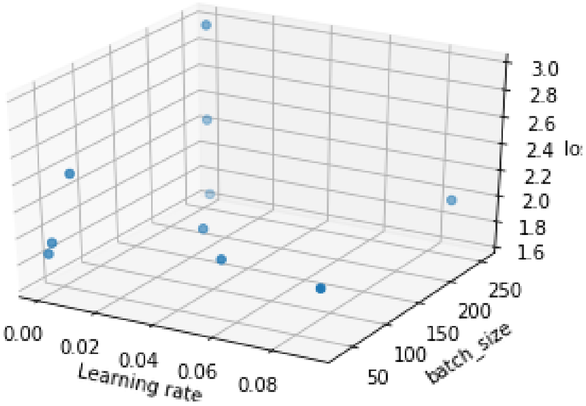


Figure 3: Learning rate and batch size tuning, first search.

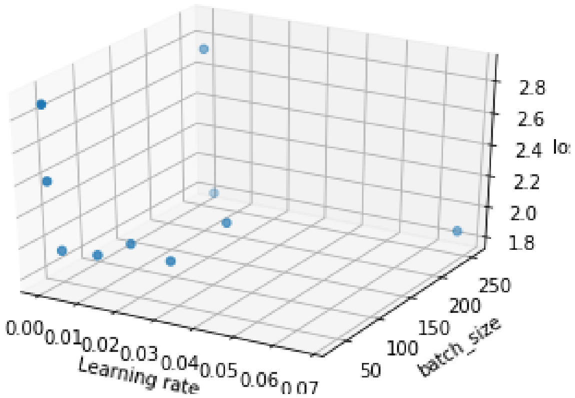


Figure 4: Learning rate and batch size tuning, second search.

As can be seen the best results are obtained with batch size equal to 64 and small learning rate, in particular the best couple of hyper-parameters found is batch size = 64 and learning rate = 3e-4.

With this values the model reached an accuracy between 44.7% and 47.2% on the art paintings' domain.

Validation with adaptation

Lastly, I trained the model with adaptation, by validating on photo to cartoon and on photo to sketch domain adaptation and then computing the mean of the results.
The coarse search for learning rate and alpha led to the values represented in Figure5; the

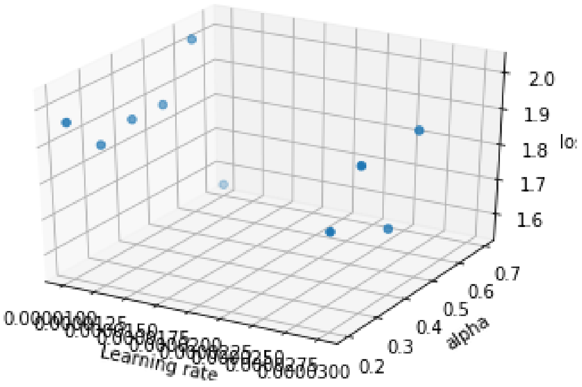


Figure 5: Learning rate and alpha tuning, first search with adaptation.

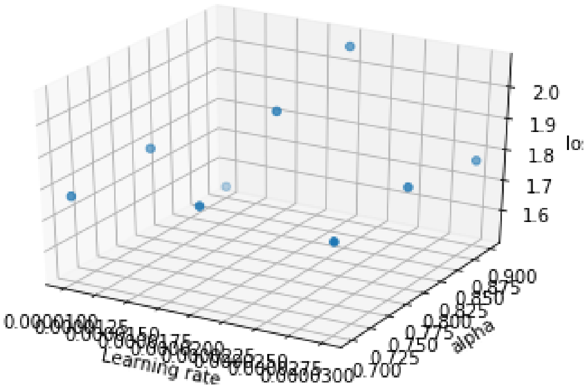


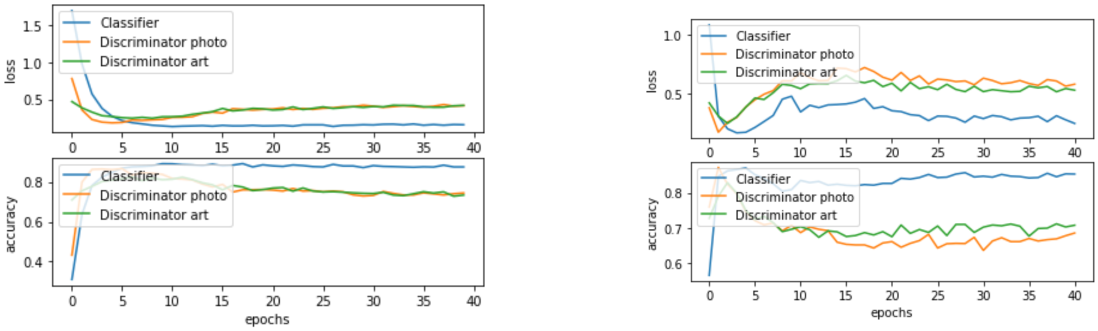
Figure 6: Learning rate and alpha tuning, second search with adaptation.

learning rate was drawn from [1e-5, 3e-5, 1e-4, 3e-4, 1e-3], while alpha from [0.2, 0.3, 0.4, 0.5, 0.6, 0.7], I decided to use values smaller than 1, because I previously found that with alpha equal or bigger than 1 the loss explodes.

As can be seen the best results were obtained with alpha equal to 0.7 and learning rate between 1e-5 and 3e-5, I then decided to perform a finer search with learning rate drawn from [1e-5, 2e-5, 3e-5] and with alphas taken from [0.7, 0.8, 0.9], I added 0.8 and 0.9 as possible values, because 0.7, the best value found, was the largest possible.
This second search led to Figure6.

The best values found were lr=1e-5, alpha=0.9 and lr=3e-5, alpha=0.9, I decided to train two models on photo and test them on art paintings averaging the outputs, this led to an accuracy of about 48.2%.

Train losses and accuracies for the two models are represented on figure7.



(a) Train loss and accuracy for lr=1e-5 and alpha=0.9

(b) Train loss and accuracy for lr=3e-5 and alpha=0.9

Figure 7: Train losses and accuracies for the best learning rates and alpha found.