Discussion on Model Training

annicas

September 21, 2023

3.3- 3.5 Observations and Questions

3.3a High Accuracy After One Epoch

You may observe that the accuracy is quite high already after one epoch, what is the main reason for this? Hint: Look at the images in TensorBoard, where the pixels in the image that are predicted as road (more than 0.5 score from the network) are colored in red.

The images include most NO-ROAD pixels and when they are evaluated with the labels, most of the predictions align with them. Therefore the accuracy will be illustrated very high right after one epoch because of imbalanced class distribution, even though most of the roads are wrongly predicted as NO-ROAD (making false negatives).

3.3b Training Loss vs. Accuracy

The training loss should be decreasing from the start, but it might take some time before the accuracy increases after the first epoch.

The reason for this is; even though the model is learning to make better predictions in the beginning and consequently decreasing the loss, the model still doesn't classify correctly. The loss starts off very high (¡0.3), because the model's weights are randomly initialized at first and predictions are way off. Then it takes at least one epoch for the model's weights to adjust, learn and improve the model. The loss decreases to ..., getting closer to more correct answers. Nonetheless, the model keeps predicting incorrectly, hence accuracy remains low for a while.

3.4 Epoch and Steps

Each epoch contains of 70 steps and in total evaluates 820 steps.

3.5 Good Accuracy vs. Learning

Can you think of any cases where you can get a good accuracy result without the model having learned anything clever?

In the case of overfitting, the model will specify its training to the given images and will output a high accuracy despite not learning any more new. The model will not be generalized enough to give as such good results for new images.

3.5b Additional Metrics

Can you think of any other metrics that might shed some additional light on the performance of your model

The task of the problem is to find the actual correct ROAD-pixels, so it's sufficient to know the correct ROAD-pixels (positive predictions). To avoid predicting NO-ROAD pixels wrongly, we can observe and improve precision. When precision is very good, it's possible to miss positive instances. Therefore recall is also great to use in order to successfully identify all ROAD-pixels. Furthermore, to get an even more robust model, we can implement F1 Score that depicts characteristics from both precision and recall.

3.6 Implement U-net questions

3.6a Transposed Convolution (Deconvolution)

Transposed convolution is a part of the UNET that consists of upsampling and increasing the spatial resolution of feature maps. In a 1D-case, during upsampling the kernel inserts values between the existing elements in the input image. This expansion results in doubling the numbers of feature channels in the input data.

3.6b Trainable Parameters

The model has a total of 529,457 trainable parameters.

3.6c Behavior in Training and Test Modes

I expect the model to behave consistently in both training and test modes since it does not include dropout layers. Therefore, during training, all nodes are utilized in the same way as during testing.

3.6d Segmentation with Multiple Classes

Since this is a multi-class classification, soft-max suits well as the activation function because it provides a feature map of pixels representating probability

distributions over the classes. Also, the number of output channels should correspond to the number of classes, because the output is one-hot encoded and there should be one class per channel.

3.6e Importance of Skip Connections

The main reason of skip connections is to maintain the features or the spatial information that are lost when increasing depth in upsampling. Additionally, they result in smoother loss function and avoid Vanishing gradient problem.

3.6f Classification Task Modification

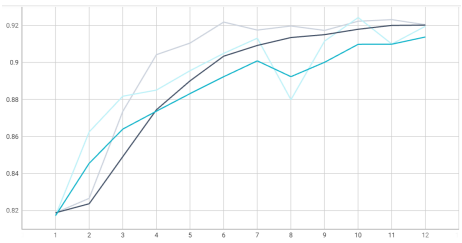
You will have to change the architecture to remove the upsampling layers and replace the final layer of the model with a fully connected multi-layer perceptron. The activation function should be a soft-max. For the loss function we can use cross-entropy loss.

3.6g Image-to-Image Transformation Loss

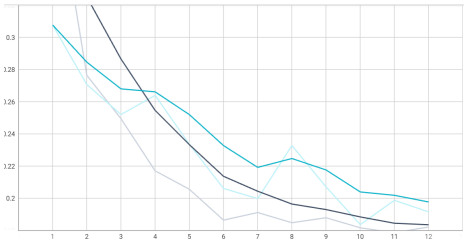
In an image-to-image transformation task, where the goal is to make each input image similar to another given image, the loss function should be modified to Mean Square Error (MSE)

4 Loss and Accuracy

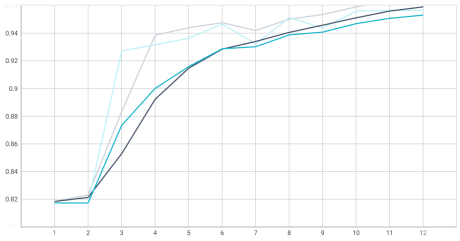
Here are screenshots of the plots for both simple and U-net model.



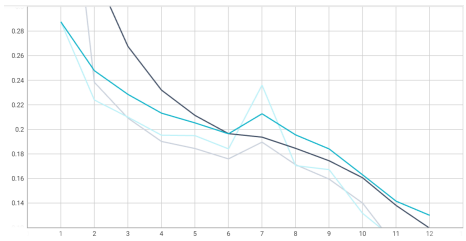
(a) Accuracy Simple



(b) Loss Simple



(c) Accuracy U-Net



(d) Loss U-net