**University of Oslo**
**Department of Mathematics**

Fall, 2024
Mandatory Assignment 1

Author: Jonas Semprini Næss

# TEK5040 - Deep Learning for Autonomous Systems

## Answers and Analysis

*You may observe that the accuracy is quite high already after one epoch, what is the main reason for this?*

**Answer:**

By observation of the images present both in training and validation it is understood that a majority of pixels belong to the NO-ROAD class. As a consequence, the model thus predicts the classification rather "well" since by initialisation it presumes that every pixel belongs to NO-ROAD. Though this is by no means a reasonable prediction since it generalises way too heavily on ONE feature.

*The training loss should be decreasing from the start, but it might take some time before the accuracy increases after the first epoch, why do you think this is?*

**Answer:**

From how the classification is constructed, the threshold for a pixel/prediction to be labelled to class 1 or ROAD is given by

$$\mathbb{P}(y_{\text{pred}}) \geq 0.5.$$

Whereas these probabilities continue to rise from initialisation, but take time before reaching the desired values.

*How many training steps are there per epoch?*

**Answer:**

The expression for calculation the number of training steps per epoch is given by

$$\text{SPE} = \frac{|T|}{|B_t|}$$

where $|T|$ refers to the size of the training data set and $|B_t|$ is the size of the batches used in training. That yields the following

$$\text{SPE} = \frac{274}{4} = 68$$

*How many training steps are there in total?*

**Answer:**

The general expression yields

$$\text{TTS} = \text{SPE} \cdot E$$

where $E$ naturally describes the number of epochs. That then gives

$$\text{TTS} = 68 \cdot 12 = 816$$

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

**Answer:**

In cases where one class is highly present or dominant the general model could achieve reasonably high accuracy by simply constructing it to classify all predictions to the dominant class.

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

**Answer:**

Dice-Sørensen coefficient, Precision and Recall, F1 score and Intersection over Union.

*Briefly describe transposed convolution also known as deconvolution.(Hint: It may be easier to consider 1D-case.)*

**Answer:**

Transposed convolution is a method of unsampling a given input map to create a greater output map. It can be understood as a form of inverse operation to the standard convolution where a single input pixel is mapped to a larger set of pixels in the output. In this way the method aims to upscale/increase the resolution of a given image.

*How many trainable parameters do your model have?*

**Answer:**

To find the number of trainable parameters we recall that the expression

$$p_j = (k_h^j \cdot k_w^j \cdot C_{in}^j + 1) \cdot C_{out}^j$$

calculates the number of parameters available for training in an arbitrary layer $j$, where

- $k_h^j$: Height of the convolution kernel (filter)
- $k_w^j$: Width of the convolution kernel (filter)
- $C_{in}^j$: Number of input channels (or depth of the input feature map)
- $C_{out}^j$: Number of output channels (or number of filters)
- $+1$: Bias term for each filter.

Hence will

$$p = \sum_{j=1}^{N} \left( (k_h^j \cdot k_w^j \cdot C_{in}^j + 1) \cdot C_{out}^j \right)$$

determine the total number, which for this model gives $p = 436857$ parameters.

*Do you expect your model to behave differently under training and test modes (i.e. for a given input do you get the same output from the model in train and test modes)? State the reason for your answer.*

**Answer:**

The model should not behave differently seeing as neither Dropout or Batch Normalisation is applied to any layers during training.

*If your task was to perform segmentation with more than two* (eg : four classes $\{\text{ROAD}, \text{BUILDINGS}, \text{CARS}, \text{OTHER}\}$), *how would you change the activation function and number of output channels?*

**Answer:**

The number of output channels need to equal the number of classes, meaning it has to have four channels. Moreover, would the Softmax activation be appropriate since we are working with multi-class classification and it produces a probability distribution over all classes.

*Briefly explain why skip connections, shown in gray arrows in Figure 1, are useful in segmentation.*

**Answer:**

Skip connections address the vanishing gradient problem by ensuring uninterrupted gradient flow from early to late layers. During downsampling, local information is often lost as the network generalises on broader, more global information and features (though hopefully more important). In segmentation this proves to be a challenge since each pixel is classified separately and local information might prove useful. Skip connections are thus a method of addressing the issue of loosing local information in the downsampling process.

*If your task is classification of a given image, how do you modify the model architecture and the loss function?*

**Answer:**

The activation function should as mentioned have as many channels as classes. The loss function for a classification problem should be either binary cross-entropy for binary classification (Sigmoid), or categorical cross-entropy for multi-class classification (Softmax), and the up-scaling/up-sampling layers are not needed.

*If your task is to transform each input image to an image similar to another given image, how do you modify the loss function?*

**Answer:**

An option to opt for would be to calculate a pixel-wise loss function. Namely through either Mean Squared Error (MSE) or Mean Absolute Error (MAE), which would look at the differences in pixels from the input image to the transformed.

*Minimizing common loss functions can be interpreted as maximizing the log likelihood. Give arguments for or against such an interpretation for the loss functions in the previous two (i.e. loss functions in classification and image-to-image transform).*

**Answer:**

**Classification (Binary and multi-class)**

In the case of classification problems where the applied loss function is cross-entropy, minimising it, is equivalent to maximising the log-likelihood. Here is why

Suppose we have data $y_i \in \{0, 1\}$ (this is analgous for multi-class classification, just extend the class of labels $y_i$) that follow a Bernoulli model with paramter $\theta$. Then the likelihood is given by:

$$L(\theta|y_i) = \prod_{i=1}^{n} \hat{y}_i^{y_i} \cdot (1 - \hat{y}_i)^{1-y_i}$$

where: $\hat{y}_i = p(y_i = 1|x_i, \theta)$ is the predicted probability that the $i$-th data point $x_i$ belongs to class 1, given the model parameters $\theta$ and $(1 - \hat{y}_i)$ is the predicted probability that the $i$-th data point belongs to class 0. Then the log-likelihood follows

$$\ell(\theta|y_i) = \sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

where we remember that the binary cross-entropy is

$$L(\theta|y_i) = -\frac{1}{n} \sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

meaning $\ell(\theta|y_i) = -nL(\theta|y_i)$. And maximizing the log-likelihood is equivalent to minimizing its negative (here the cross entropy-loss

**Mean Squared error (MSE)**

Minimizing the Mean Squared Error (MSE) is not always equivalent to maximizing the log-likelihood, since it relies on assumptions about the distribution that the data follow. Namely

If the errors $\epsilon_i = y_i - \hat{y}_i$ follow a standard normal distribution $\sim \mathcal{N}(0, \sigma^2)$ then:

$$p(y_i|x_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \hat{y}_i)^2}{2\sigma^2}\right)$$

and the log-likelihood by proportionality gives,

$$\ell(\theta|y_i) \propto -\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where the equivalence follows by

$$\ell(\theta|y_i) \propto -n \cdot \text{MSE}$$

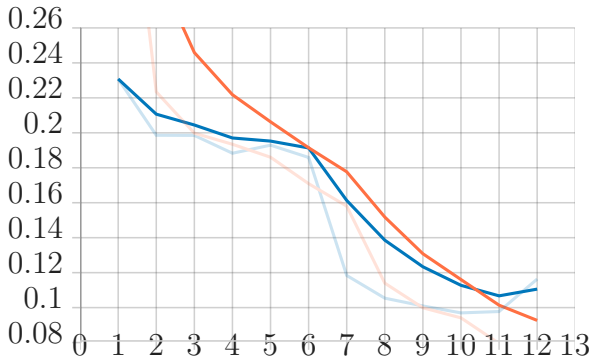
A counterexample arises if say the errors are not normally distributed, but rather Laplace distributed, where maximising the log-likelihood would be equavialent to minimising the Mean Absolute Error (MAE).

*Train your model. Did you notice any improvements over the orignal model?*

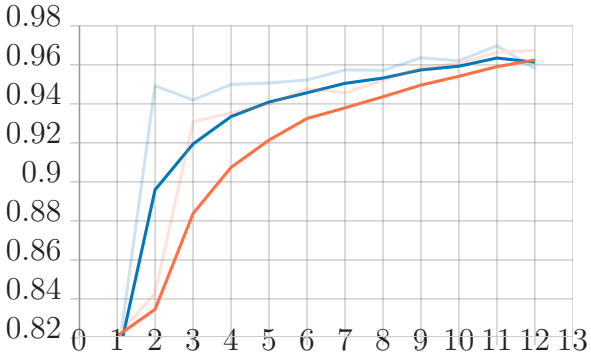
**Answer:**

The U-net model is superior in terms of accuracy, though at the expensive of computational effort.

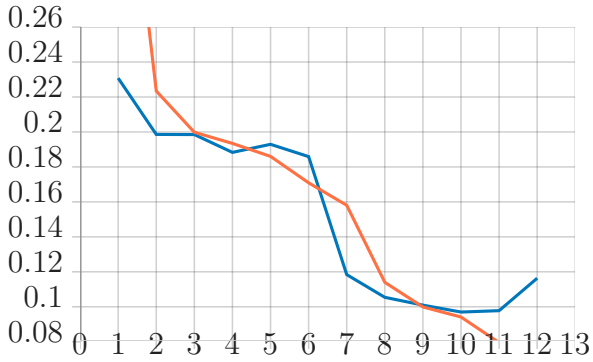
# Plots



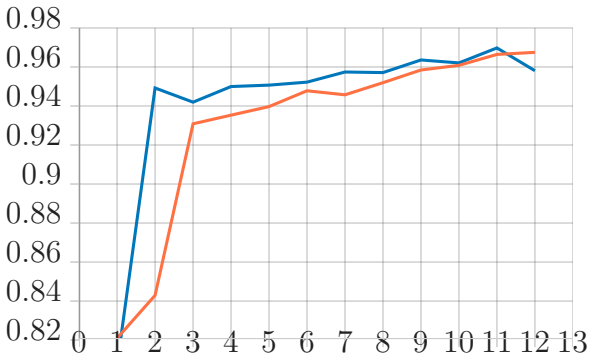

(a) Loss with smoothing




(b) Accuracy with smoothing

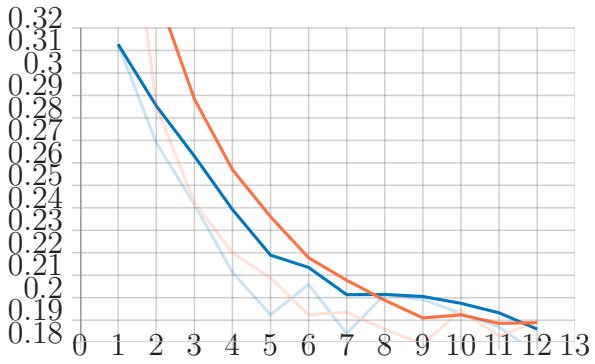



(c) Loss without smoothing

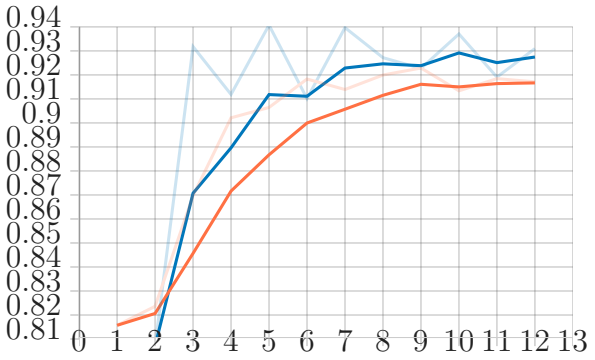



(d) Accuracy without smoothing

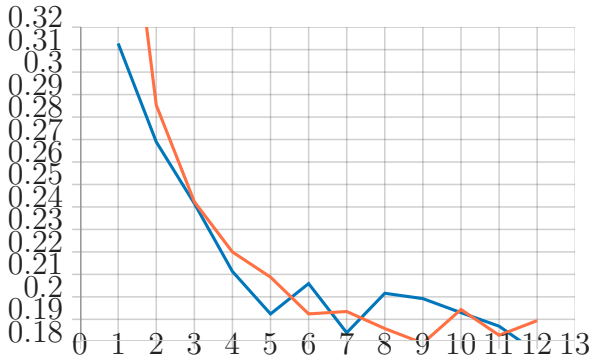
Figur 1: Unet accuracy and loss



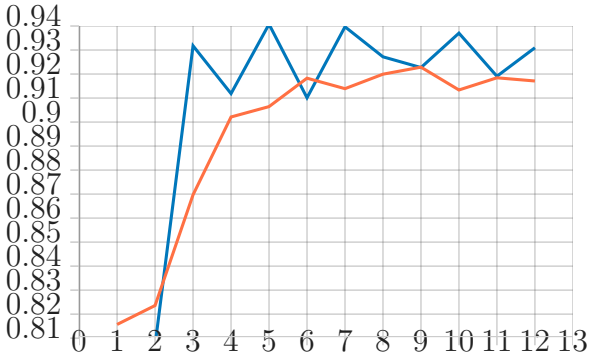

(a) Loss with smoothing




(b) Accuracy with smoothing




(c) Loss without smoothing




(d) Accuracy without smoothing

Figur 2: Simple model accuracy and loss