

Take-home Exam

DAT340: Applied Machine Learning

Part 1:

Question 1.

a) This is an image classification task, which calls for a convolutional neural net (CNN). We need to determine if an image shows two, one or no hands on the steering wheel. This means it is a multiclass (3) classification problem. This can be reduced into two binary classification problems; hands on or not, if on: one or two. However, since we are using a non-linear classifier, we are capable to deal with all classes simultaneously. A CNN requires some preprocessing, we need to make sure all input has the same shape, i.e. dimensions and channels. Some resizing or reformatting might be needed if different cameras or settings are used. The size of the input will affect training time and computational demand, so this is one parameter to adjust.

Next is the architecture. This requires some experimentation, but initially a few sets of convolutional and max-pooling layers make for a good starting point. After the spatial image manipulation, a flattening layer followed by some fully connected layers follows. Perhaps some dropout and/or noise layers are useful to prevent overfitting, validation on the test set will reveal this. All layers up to this point has Relu activation. Finally, a dense layer with 3 nodes (one for each class) with softmax activation is added. The output can be interpreted as the probability that an input corresponds to each of the 3 classes. The 3x1 output vector sum to 1. One proposed architecture is as follows:

```
model = models.Sequential(layers = [
    layers.Conv2D(filters = 32, kernel_size = (4, 4), strides = (1, 1), input_shape = (img_size, img_size, 3)),
    layers.MaxPool2D(pool_size = (4, 4)),
    layers.Conv2D(filters = 64, kernel_size = (4, 4), strides = (1, 1)),
    layers.MaxPool2D(pool_size = (2, 2)),
    layers.Flatten(),
    layers.Dropout(rate = 0.2),
    layers.Dense(128),
    layers.Dense(32),
    layers.Dense(3, activation = "softmax")
])
```

Before training the network, a loss-function and an optimizer need to be chosen. Categorical cross entropy loss is standard for multiclass classification. Note that in Keras CategoricalCrossentropy expects one-hot encoded labels, if the labels of our images are not on this format they must be converted. If the labels are integer (0 – 2), then SparseCategoricalCrossentropy can be used directly instead. The adam optimizer which is an efficient stochastic gradient descent implementation is generally a good choice.

Before training on the collected image data set a few remarks can be made. Since there are a lot of images keeping them all in memory is expensive, therefore a generator loading the images by demand is practical. At this stage some manipulation, including resizing, can be implemented. Some of the issues discussed in b) can be addressed here. Also, the labels need to be encoded correctly at this point. If there are too few images to have a large enough training set, some variations in the images can be made here and thereby producing more input. A test set and/or validation set is also needed to be created from the available images. An 80 – 20 split is OK.

b) Some inspection of the training set can reveal some issues, however we only have 4 images from the training set available. So, this part contains some speculation. The images are all well-lit and taken from the same angle. Perhaps all cameras will be positioned in this way, but it is fair to assume that different trucks can have different view-angles for the cameras. If the training set does not reflect this, then the model might generalize poorly. Since there are thousands of hours of footage, there probably is a decent amount of poorly lit images as well. A potential issue is how the system perform if the driver is wearing dark clothes and gloves (or even dark skin tone), making it more difficult to detect hands from dark surroundings of a typical dashboard. Another issue might be an unbalanced dataset. A decent accuracy might be achieved from always predicting two hands on the wheel. Also, from this angle it might be difficult to determine if the hands are on or in front of the steering wheel. Perhaps another view is needed as well for depth perception.

To conclude, perhaps a sensor in the steering wheel is a better way to go about this problem.

Question 2.

	Model 1		Model 2		Model 3	
Accuracy	0.976		0.976		0.967	
<i>Prediction</i>	Cancer	Non-Cancer	Cancer	Non-Cancer	Cancer	Non-Cancer
Precision	0.850	0.987	0.792	0.996	0.720	0.997
Recall	0.850	0.987	0.950	0.978	0.963	0.967

Question 3.

Let's use the SelectK-best feature selector and the PCA dimensionality reducer to explain this. When using the feature selector, we chose a number of features from the original feature set using some criteria. We are then left with a subset of the full feature set. When using the PCA dimensionality reduction we compute the principal vectors of our data set. The principal vectors describe the "directions" in our feature space which have the most variance. This is equivalent to saying what combination of features account for the most spread in our data. These feature combinations are more interesting for classification as we are interested in makes our data points different. When select some of these vectors by some criteria (more variance than some threshold, Kbest etc), we end up with a set of linear combinations of our original feature set.

Question 4.

a) Bikes, scooters and such mobility tools are generally easily recognizable and require no special knowledge to be able to differentiate between, so it is expected to get a big kappa score. 0.95 is a great kappa score, however personally I would always do some inspection of the annotation. Image annotation requires boxes to be drawn around objects to be recognized, here there can be some discrepancies between the annotators. E.g. the boxes are drawn very large.

b) Usually, internet annotation pays poorly and can be viewed as an outsourcing to third world countries in the same way sweatshops are. Also, the quality is expected to be worse than what professional annotation firms can provide. For autonomous driving security is very important, and therefore quality in every step of the development is required. This includes annotation. Finally, if cheap internet labor becomes the standard in research because of costs, future projects might receive less money. This might lead to annotation being done by underqualified people.

Question 5.


a) At the very least we need to make sure that the input always has the same shape. (Image size and channels). There might also be some reshaping due to batch processing needed. Also, perhaps gray

scaling the image is necessary (i.e. make it 1 channel). Some architectures might demand data in the range 0-1 instead of integers in 0-255, so a division by 255 might be necessary.

b) This kernel detects vertical edges. Right edges will be positive and left edges negative. If combined with relu, only right edges will be detected.

c) A 3x3 max pooling layer returns the biggest number in each 3x3 region. With strides of 1, each region is separated by a move of 1 pixel in any direction. Pooling layers give some robustness to the position of features and helps against overfitting.

11	14	4	5	8
34	5	9	15	27
32	6	18	3	13
4	29	14	18	20
1	5	9	13	15



34	18	27
34	29	27
32	29	20

d) For the hidden layers ReLU is fine, however both sigmoid and tanh can also be used. Typically, ReLU will give faster training as more neurons will take zero values and fewer operations are needed. As there are multiple output classes softmax is appropriate for the output layer.

Question 6.

a) The easiest way to avoid overfitting in decision trees is to limit the depth of the tree. This forces the tree to take decisions for larger groups of data, limiting the overfitting. There are multiple ways of doing this. We can preset a depth of the tree and return the majority class when the depth is reached, this is called prepruning. Or we can grow the entire tree, and then cut the less important subtrees afterwards. This is called postpruning. This approach is computationally more expensive but give better results as some branches might require more depth than others.

b) Tree ensembles are collection of decision trees with minor individual differences. The result of the entire ensemble is determined by a majority vote. Ensembles are more robust against overfitting since the individual trees different, thus generalize the data. Sort of like asking many people a question instead of one is more likely to help you find the correct answer. Random forest is one tree ensemble method, here each tree is trained on a randomly picked subset of the training set.

Question 7.

a) My answer to this would be dependent on how intrusive the second round of screening is, and how common and dangerous the cancer is. Let's assume the ratios of the training set is accurate for the people being sent to screening 1, i.e. 8% have cancer. (Since the screening is described as simple and cheap, I would assume the actual number is far lower). Let A be that you have cancer, $P(A) = 0.08$. Let B that the screening says that you have cancer.

Using Bayes theorem, the probability of a person having cancer when the screening says so is given by:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, P(A) = 0.08$$

Model 1

$$P(B) = 0.08, P(B|A) = 0.85$$

$$P(A|B) = \frac{0.85 * 0.08}{0.08} = 0.85$$

Model 2

$$P(B) = 0.096, P(B|A) = 0.95$$

$$P(A|B) = \frac{0.95 * 0.08}{0.096} = 0.79$$

Model 3

$$P(B) = 0.107, P(B|A) = 0.963$$

$$P(A|B) = \frac{0.963 * 0.08}{0.107} = 0.72$$

From these statistics it can be seen that model 1 is the most accurate, it has the highest cancer prediction precision. So, if the screening round 2 is very intrusive this might be the model to go with. However, as can be seen from the recall computed in Q2, it has the lowest recall. That is, of the 3 models it correctly classifies the fewest with cancer. As the cancer is very dangerous, obviously far more than the screening round 2, it is most likely better to use method 2 or 3. The combination of an intrusive screening and dangerous cancer calls for method 2.

b) How early the cancer can be detected is one medical factor. From the machine learning point of view interpretability is important.

Question 8.

We can use semi-supervised learning. Semi-supervised learning is a mix of supervised and unsupervised learning. When only part of the training data is annotated, we use this to learn to predict the outcome, given some assumptions on the data. The assumptions are like in the unsupervised case: spatial position, clustering etc. The assumptions are extracted from the small, annotated data set. Another option is to do self-training. Create a baseline predictor on the annotated data and predict the unannotated. Giving each label a confidence score makes it possible to extract the most confident predicted labels and retrain the model on these as well.

Question 9.

a) The code shows the 10 most important features measured by mutual information. Mutual information is measure of dependence between two random variables. Here the first random variable is the feature to compute the importance of, and the second is the labels. If the labels are strongly dependent on a feature, then that feature is helpful for predicting labels. Since we have both continuous and categorical data, we use a dictvectorizer to vectorize the strings. One should probably standardize the data, but maybe not by mean as there might be very many categorical variables which gives a lot of zeros in the unstandardized dataset. Many zeros make for efficient computation, but subtracting the mean will skew the zeros to some small negative value instead.

b) We see these labels in the output list since the strings are vectorized. This means that every unique string per feature is given its own feature. So, the string E2 is given the feature E=E2 which is 0 when our data does not include the string in E, and 1 if it does.

Question 10.

Despite the name, logistic regression is a linear classification method. The linear classifiers are all binary in nature, meaning we can only have 2 output classes. To extend them to multiple classes multiple classifiers must be trained. One can either go for a One-Vs-Rest approach, or a One-Vs-One approach. The first approach needs to train N classifiers for N classes. The second must train $N(N-1)/2$. In the first approach data is classified to belong to either one class or one of the rest. This is repeated until the correct class is found. For the One-Vs-One each duel is scored, in the end the high scoring class is the predicted as the correct class.

Question 11.

a) This scheme is targeted at young people, however almost no current youth use Facebook. But for the sake of the task, let's pretend this is the case.

Some questions:

- Is there actually a link between grammar and driving ability?
- Are there better indicators?
- Can we get data from other sources as well? LinkedIn, credit scores, parents driving habits etc?
- Will this lead to more customers and more revenue?
- Can the money for this project be better invested elsewhere?
- Any better/good ideas?

I would not let this project go ahead.

b) I don't think there exists a significant link between grammar, sentence structure and driving ability. Probably some discrimination of language/dialect also.