University of Essex

Computer Science Department

Artificial Intelligence MSc

Machine Learning module

Unit 11 Assessment

January, 2025

Audio transcription from the presentation

Slide 1

Good day. Here is the assessment from Unit 11 of Machine Learning module.

Slide 2

Agenda is the following.

First, I am going to generally outline a task and given dataset. Then I will elaborate on the theoretical essentials: validation set, convolutional neural network, architecture of the model with particular attention on activation function and loss function.

Later, I am going to present efforts undertaken in order to find an optimal model, coupled with this model outcomes and conclusions.

Finally, I share my general reflections and lessons learnt.

Slide 3

General introduction.

The task is to create an artificial neural network recognizing pictures from ten different categories based on CIFAR-10 dataset.

CIFAR-10 is a renowed dataset consisting of 60,000 RGB pictures of 32px x 32px. Each picture is assigned to one of ten given categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck (distributed equally for each category).

As for the Explainatory Data Analysis: the dataset initially is divided into two groups: training set (50,000 elements) and test set (10,000 elements).

The data is structured in four arrays (or two tuples) as depicted below: for training we have 50,000 pictures and 50,000 labels, and for testing 10,000, accordingly.

The dataset structure is consistent; there are no gaps. The dataset is complete, there are no missing values.

As the data has been manually labelled, it may be trusted, however, after examining randomly small subset of it, because of a low resolution, some instances categorization may rise controversion.

Slide 4

As mentioned, the original set is separated in groups for training and testing purposes. Such division allows to assess the model accuracy independently. Nonetheless, for more sophisticated datasets—as the training process can often be time-consuming – it is important to have an independent indicator of advancement in gaining accuracy by the model. Once we independently confirm the model's enough accuracy or lack of advancements in subsequently taken epochs, we stop training process. For such purposes, another set called „Validation set" is established.

The importance of maintaining a separate validation set:

- Independent indicator of a accuracy gaining rate by the model,
- A trigger to finish the training procedure after a number of ineffective training epochs.
- Hence, it is an efficient method of overtraining detecting.

The proportion of training / validation / testing sets:

- M. Kubat in an essential handbook „An Introduction to Machine Learning" stated that „To train multilayer perceptrons is more art than science" having in mind also a hyperparameters tuning (including „ training / validation / testing" problem). As there is no general rule, the scholars seem to generally agree that the bigger the dataset, the share of training part should tower above the rest.
- However, the proportion for this task would be 2/3 (66%) training, 1/6 (17%) validation, 1/6 (17%) testing.

Slide 5

For the task given in the assignment, we will use Convolutional Neural Network – a type of neural network where multi-layer perceptron is followed by a number of convolutional filters and spatial reduction. These layers preprocess input image for subsequent deep network: detect important features, reduce a noise and downscale where the amount of provided details is excessive (so called „pooling”).

In details, in the beginning input values are being processed with a convolutional, „walking” smaller window, called kernel. This kernel applies adaptive filters to the input creating a number of layers, each with a different filter. Filters detect most important features in the input. Convolution layers mathematically are kind of artificial neurons, and they also use an activation functions, which we will elaborate on later in the presentation. For the sake of efficiency, these layers are later scaled down („pooling”). The steps taken can be multiplied, so that many filters are applied and scaled down in loop.

After that an entire set sub-layers subjected to filters is merged in a single array (flattening), and feeds multi-layered perceptron with potentially several hidden layers. Finally reaching an output layer, the output is frequently represented as possibility range. Convolution Neural Networks, as part of neural networks family, also can use backpropagation with loss function. On this topic we also will elaborate later in the presentation.

Slide 6

Having described the general idea behind Convolutional Neural Network, here we are turning to real implementation of the model.

We begin with a baseline architecture: input layer, single convolution and pooling layer followed by flattening, single hidden neural layer and output. In the next step we extend the model with secondary convolution and pooling. Moving forward, as a second modification we added another hidden layer of neural network.

Once Albert Einstein was asked by one of his fellow professor: „Dr. Einstein, how we should teach physics?” Einstein replied: „Explain things as simple as it can be, but not simpler.” Choosing a proper architecture for a model is similarly a very subtle task, and gaining an optimal performance often means finding a golden proportion between oversimplified and overcomplicated structure. The so-to-speak sophistication ladder depicted on the slide captures in the nutshell those two wings: baseline model does not meet the required accuracy criteria (0.607), while the 2nd modification model with all its complexity lacks any additional value compared to its 1st modification precedestor (insignificant difference between 0.662 vs 0.659 accuracy), which turns out to be an optimal choice.

Obviously, we can dig deeper as variety of the structures and hyperparameters may evolve like a Mandelbrot sets; however, we have to stay within reasonable complexity boundaries.

Slide 7

Activation function is a mathematical processing of amplification or dumping of a sum of weighted neuron’s inputs. As name suggests, its role is to activate or not a neuron, imitating functioning of biological counterparts. Frequently, artificial neural networks use one of the following activation functions.

ReLU remains the most popular activation function for input or middle layers in CNN, while softmax is preferably chosen for output layer. The advantage of softmax lies in its convenience: each output is represented by probability, and the sum of them is equal to 1.

For the given task, after trying a few different functions (see the following slides), I chose the most promising functions: ReLU for an internal part of network, and softmax for an output layer.

Slide 8

Describe the loss function implemented.

The loss function represents a difference between calculated and expected output. It holds an important role both in indicating the accuracy, and also in backpropagation process. Hence, its computational complexity is of extreme importance. In the picture below we can find a general neural network scheme with a loss function depiction

In general, there are many types of loss functions, e.g.:

- $\log_2$ loss function,
- Mean Squared Error
- Mean Absolute Error

With the softmax activation function applied in the last layer, the loss function we use equals:

$L_i = -\log_2 p_i$.

As described earlier, softmax function outcomes are possibilities (here: p_i). Low loss function means that model is accurate, accordingly.

Slide 9

After deciding on the architecture of the network in the previous slide, we are able to continue our endeavors with hyperparameters tuning.

In the slide you can see the „tuning knobs" we can use. These are: kernel filters in two convolutional layers, activation function of kernels and hidden layer, and number of neuron in hidden layer. At this point and in this case, there is no other solution than trying and correcting manually. I started with a basic hyperparameters set which resulted in a relatively decent accuracy (0.636) and computing costs (5 epochs). This was on the ReLU activation function. Switching to different function – sigmoid – results in accuracy decrease (0.608), and even more significant performance dropdown (17 epochs). Combination of sigmoid and ReLU in the next version demonstrates slight improvement in accuracy and reduction in number of epochs. Fully returning to the ReLU in the version 4, I increased numbers of filters in convolutional second stage (to 64), which in fact slightly affected computation efficiency (7 epochs), yet increased the accuracy but more than 10% (0.683).

Following this successful path, I doubled the number of filters in both convolutional steps obtaining the most accurate level so far with only 6 epochs. While continuing to expand the number of filters would be too far-fetched, in the next, final step I decided to double the number of neurons in the hidden layer. Though more efficient computationally (5 epochs), the accuracy does not change from previous, less sophisticated architecture. Therefore version V – the one with increased number of filters in convolutional layers, use of ReLU activation function and 256 neurons in hidden layer – is the optimal one at this level. It took 6 epochs to train the model until it started loosing accuracy (become overtrained)

Slide 10

To conclude, we chose a fifth version of network with given hyperparameters.

This model produced the best accuracy among the others. In order to assess a deeper quality of the model, researchers use another performance metrics like precision, recall, and F1-score. Precision is a ratio between correctly classified positives and the everything classified as positives. A perfect model will have a numerator and denominator equal, so precision of 1 means model perfection in positives classification. Here, we obtain 0.67 corresponding with the accuracy level.

Recall in turn is a ratio between correctly classified positives and every positive in the set. Here also ideal model will score 1.0. In our task we gained 0.66.

F1-score combines in a single metrics recall and precision, so that F1-score of 1 means perfect recall and precision, and conversely F1-score of 0 means both recall and precision of 0. In our case it equals 0.66.

More advanced tools like confusion matrix is provided as an appendix.

Finally, those metrics give us a theoretical understanding of the model performance, yet to see it in real life application is a keystone to our endeavors. In the slide we can see five random examples of classification, where model labels mistakenly in one of them. Realizing how low-quality the examples are, achieved accuracy presents much better.

Slide 11

As a conclusion, allow me also to share my reflections on the learnings acquired while tacking this task and lessons learnt.

First of all:

- The importance of profound familiarity with field research prior constructing the model structure

My observation is that learning through rich experiences of researchers and seasoned practitioners can save considerable amount of time and efforts, and avoid potential mistakes. Especially as the training of neural network is very time-consuming, being aware of good practices can be priceless. This is my biggest lessons learnt.

- Second observation is that proficency in understanding/sensing of when to stop searching for a more optimal model is crucial

Another lesson learnt for me is that adjusting hyperparameters is not a one-way task; increasing network architecture size does not always come hand in hand with accuracy gain, and stopping at the peak of local maximum of accuracy/complexity value is consequential.

- Third observation: crucial role of validation set

Validation set acts as an nearly independent judge in observing real accuracy during the training. As seen in the plot below visualising the accuracy gain within epochs, prediction accuracy of validation set increases only until certain point, while the one for test set keeps rising (misleadingly).

- Fourth observation is on the notions of Recall and Precision

It is never too much of appreciation for those two. These two metrics give a good understanding of model performance and its advantages and weaker sides. In turn, F1-score provides handy combination of those.

There are also more lessons learnt; all of them remain as a part of my growing experience.