Lecture 9: *"Introduction to Artificial Neural Networks with Keras"*

Book Chapter

Please read the *Chapter10* **"Introduction to Artificial Neural Networks with Keras"** starting from the subsection "Building Complex Models …". Leave out the subsections “Using Subclassing API…” and “Using TensorBoard …”.

Answer the following Questions.

Keep in mind: If you answer these questions and write a detailed summary, you won't need to read these chapters again while preparing for the exam

Questions

Subsection “Building Complex Models …”

1. Describe the general architecture of a Wide&Deep model. What is the advantage of this architecture?
2. Let us assume we want to predict a property of an image, e.g. if it looks calm or not. We decide to calculate some features like sharpness, White portion of the image and give them as input in addition to the pixels. What architecture of the model would you recommend?

(No questions to “Using Subclassing API …”)

Subsection "Saving and Restoring a Model…" and “Using Callbacks”

1. What can you do to avoid having to adjust a model multiple times? How can you do it automatically?
2. How to avoid optimizing the number of epochs?

(No questions to “Using TensorBoard …”)

“Fine-Tuning Neural Networks Parameters” and following subsections

1. Which ways exist to fine-tune the hyperparameters of a neural network automatically? (Note: edition 2nd describes how to wrap a keras model

into a regular scikit-learn regressor and using Randomized Search for this, in edition 3rd they recommend to use the Keras Tuner library instead)
2. Theoretically, a neural network can learn complex tasks with only one layer. Why does it still make sense to use multiple hidden layers instead?
3. How can the layers be interpreted? What do the lower layers learn, what do the upper layers learn?
4. What is transfer learning?
5. Why is it useful to use transfer learning?
6. What is a good distribution of the number of neurons across the different hidden layers?
7. What is the recommended alternative to gradually adjusting the number of neurons to avoid overfitting?
8. Which hyperparameter is indicated as the most important?
9. Name advantages for using large batch sizes and explain what needs to be considered.
10. What strategy for finding the best batch-size for a specific model is recommended?

Homework Assignment

Please work on the exercises given in 09-ANN2.zip .

Answers

“Building Complex Models…”

1. The Wide&Deep Networks allows it to transfer the input directly to a concatenation layer. In a parallel path the layers of a neural network can be establishs. The outputs come to together in the concatenation layer. This allows the complete network to learn simple patterns and also complex rules.
2. We have to define two input set and handle the separately with different layers. At the end we concatenate the outputs and add some additional dense layers.

“Saving …”

1. You can save the best model. One option is to store it manually with the save command, another is to use callbacks and the early stopping option.
2. Using early stopping instead

Fine-Tuning

3. One way is to use GridSearchCV or RandomizedSearchCV by wrapping a keras model in a scikit-learn regressor. The other way is to use a python library to optimize the hyperparameters with an optimization strategy.
4. The neural network with multiple hidden layers is more efficient.
5. The lower layer learn the low-level structures, e. g. lines oder shapes, the upper layer the high-level structures, e.g. squares and circles.
6. In transfer learning you use a final neural network with pretrained weights to apply it to a new similar task. To the neural network are added some new upper layers and it is trained with the dataset for the new task.
7. With the pretrained weights you need less training data to find a good result.
8. A good distribution is, when the neurons form a pyramid to the end. To use the same number of neurons for each layer also give good results.
9. It should be used a bigger network with more neurons and then using early stopping.
10. The learning rate
11. One can use a batch size as large as the GPU or CPU RAM can handle. Then one should warm-up the learning rate.
12. First start with a large batch-size and a warm-up learning rate. If the model is instable or bad then try training again with a small batch-size.