Lecture 8: *"Introduction to Artificial Neural Networks with Keras"*

Book Chapter

Please read the *Chapter10* **"Introduction to Artificial Neural Networks with Keras"** till page 307. 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

Additional Installation

Please install keras and Tensorflow 2.0 as described on pages 296-297.

Questions

The start of the chapter is left out. The questions begin with the section "The perceptron".

1. Describe the architecture of a Perceptron.
2. How does a Perceptron work?
3. What is a Fully Connected Layer or also called Dense Layer?
4. Which problems are solvable with single Layer Perceptrons?
5. What is a Multilayer Perceptron (Feed Forward Network)?
6. Explain the backpropagation algorithm.
7. Why do we need an activation function with a non-zero derivative?
8. What is the default activation function commonly used?
9. How many output neurons are needed for regression tasks?
10. What type of activation functions should you use for the last layer in a regression task? Describe the different situations and the corresponding activation functions.
11. What type of activation functions should you use for the last layer in a classification task? Describe the different situations and the corresponding activation functions. Note: In Tensorflow does not exist a logistic activation function, use sigmoid instead (see: https://www.tensorflow.org/api_docs/python/tf/keras/activations)
12. How is the loss function defined for classification?

13. Describe why the formula for binary cross entropy works well as a loss function for binary classification. You find a good explanation here: https://towardsdatascience.com/cross-entropy-for-classification-d98e7f974451
14. What steps are required to set up a neural system to the point of evaluation?
15. In a multilayer perceptron the weights are initialized randomly. Why is this important?
16. What are the different settings in a feed-forward neural network in a regression problem compared to a classification problem?

Homework Assignment

Code the exercises in 08-ANN.zip .

Answers

1. A Perceptron consists of a set of weights equal to the size of an input instance. It forms a weighted sum with the input. Then follows a step function called Heaviside function. It sets all negative values to 0 and all positive to 1.
2. All weights are iniatialized with a random number or with zero. As input all input instance are fed in one-by-one. For each instance the weighted sum is computed and the step function applied. For this output the difference to the real (true) output is evaluated. This size of error times the x_j and a learning rate is added to the old weight to update it. This update procedure is done for the whole dataset in one iteration.
3. All neurons are connected to all neurons of the preceding layer.
4. All binary problems which are linearly separable.
5. A Multilayer Perceptron has multiple Perceptron layers. They are fully connected.
6. The backpropagation algorithm updates the weigths of a Multilayer Perceptron. It has a forward and a backward pass. In the forward pass it gets several instances and evaluates the predictions of them. Then the error of the predictions - the so called loss - is computed. In the backward pass this error is divided back to the last layer. The contribution of each neuron to the loss is taken into account. This is done until the first layer is reached. Finally the weights are changed according to their contributions. This is done with a Gradient Descent Step.
7. One needs a non-zero derivative because of the Gradient Descent. If the derivative is zero, no changes are done for this step. The activation function brings non-linearity into the network. Without activation function a net would only by linear.
8. It is the ReLU Function.

9. Just one output neuron to predict the value of the regression function.
10. The choice of the activation function depends on the range of values the output function can have. If all real numbers are possible, no activation function is needed. To limit the values to the positive values one can take the ReLU function (or the smooth variant the softplus function). If the values are in a certain range, the tanh or the sigmoid multiplied by some constants are used.
11. For binary classification you use the logistic function, for multiclass classification the softmax function.
12. It is the cross entropy loss.
13. (p. 144, for more than two classes p. 149) for two classes: $-y \log (p) + (1-y) \log(1-p)$ with p prediction and y in $\{0,1\}$ real values, interpretation: if p is near 0 and the true value of y is 1 you get: -1 * high negative vaule + 0 = high positive value, if y is 0: -0 * high negative value +1 * value near 0 = small value, good explanation: https://towardsdatascience.com/cross-entropy-for-classification-d98e7f974451
14. Define a model (model = keras.models.Sequential then layers with "add") giving layers and activation function for each layer
    Compile the model giving the loss, the optimizer and metrics
    Fit the model , give training and validation data and number of epochs
    Control the metrics by plotting the "learning curves", the accuracy or mse
    Finally evaluate the model for the test set
15. If the weights are initialized with the same values also their contribution to the loss would be same and in backpropagation the correction would also the same. So they stay the same for the whole time.
16. p. 308: The output layer has just a single neuron and no activation function. The loss is the mse.