Machine Learning Worksheet Solution 9

Julius Jankowski

1 Activation functions

Problem 1:

The basis functions (also called activation functions) transform the features into a new space in a nonlinear way. If we would not use nonlinear basis functions, we would not be able to seperate data sets which are seperable but not in a linear way. As a supporting fact: No matter how deep a network without nonlinear basis functions is, we can simplify the calculation to an equivalent one perceptron operation.

Problem 2:

We can show that the sigmoid function can be considered as a shifted and scaled tanh function:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = -\frac{e^x + e^{-x} - 2e^x}{e^x + e^{-x}} \tag{1.1}$$

$$\implies tanh(x) = 2\frac{1}{1 + e^{-2x}} - 1 = 2\sigma(2x) - 1 \tag{1.2}$$

$$\implies \sigma(x) = 0.5(1 + tanh(\frac{x}{2})) \tag{1.3}$$

Now we can replace the sigmoid function by the tanh function and we know that there is a linear relation. When we adapt the new weights accordingly, we obtain a network which behaves similar to the sigmoid network.

Problem 3:

We know the relation between the tanh function and the sigmoid function:

$$tanh(x) = 2\sigma(2x) - 1 \tag{1.4}$$

Hence, according to the chain rule, the derivative w.r.t. x is derived as follows:

$$\frac{dtanh}{dx} = \frac{dtanh}{d\sigma} \cdot \frac{d\sigma(2x)}{d2x} \cdot \frac{d2x}{dx} \tag{1.5}$$

And we can plug in the derivation for $\sigma(x)$, since we know that this can be written as a function of itself:

$$\implies \frac{dtanh}{dx} = 4\sigma(2x)(1 - \sigma(2x)) \tag{1.6}$$

Now we can resubsitute $\sigma(2x)$ by the expression depending on the tanh function:

$$\implies \frac{dtanh}{dx} = 2(1 + tanh(x))(1 - \frac{1}{2}(1 + tanh(x))) = 1 - tanh(x)^2 \tag{1.7}$$

Problem 4:

Since the error function is supposed to behave similar for whatever activation function, the simplest solution for this problem is to linearly scale the output values down to the range of the sigmoid output range. Thus we obtain a scaling function:

$$g(x) = \frac{1}{2}(1 + x) \tag{1.8}$$

Consequently, the new error function looks like the following:

$$E_{new}(\mathbf{W}) = -\sum_{i=1}^{N}[g(y_i)log(g(f(\mathbf{x}_i, \mathbf{W}))) + (1 - g(y_i))log(1 - g(f(\mathbf{x}_i, \mathbf{W})))] \tag{1.9}$$

This ouput range is generated by the tanh function, because $-1 \leq tanh(x) \leq 1$.

2 Optimization

Problem 5:

$$\frac{\delta E}{\delta \mathbf{w}} = \frac{1}{m}\sum_{i=1}^{m} -\frac{\delta l}{\delta \eta}\mathbf{x}_i + \lambda \mathbf{w} \tag{2.1}$$

$$\frac{\delta l}{\delta \eta} = \begin{cases} \eta, & \text{if } |\eta| < 1. \\ sgn(\eta), & \text{otherwise.} \end{cases} \tag{2.2}$$

where $sgn(\eta)$ is the signum function.

$$\implies \frac{\delta E}{\delta \mathbf{w}} = -\frac{1}{m}\sum_{i=1}^{m} min(\eta_i, sgn(\eta_i))\mathbf{x}_i + \lambda \mathbf{w} \tag{2.3}$$

with $\eta_i = y_i - \mathbf{w}^T\mathbf{x}_i$.

Problem 6:

The best point to stop the training is after approximately 50 updates. This is the lowest point of the validation curve. The data was splitted into a training and validation set on one hand which is used for tuning the wheights and on the other hand a set of completely independant test samples which is used to obtain a measure for the generalization performance of the network. The validation error can be interpreted as a mixture of training and test error since the training error gives a measure for overfitting and test error gives a measure for generalization. Both of them are not completely representing the performance of the network, this is why to use validation error for the network assessment.

3 Numerical stability

Problem 7:

$$a + ln \sum_{i=1}^{N} e^{x_i - a} = a + ln \sum_{i=1}^{N} e^{x_i} e^{-a} = a + ln(e^{-a}) + ln \sum_{i=1}^{N} e^{x_i} = ln \sum_{i=1}^{N} e^{x_i} \tag{3.1}$$

Problem 8:

$$\frac{e^{x_i - a}}{\sum_{i=1}^{N} e^{x_i - a}} = \frac{e^{-a}}{e^{-a}} \frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}} = \frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}} \tag{3.2}$$

Problem 9:

$$max(x, 0) - xy + log(1 + e^{-abs(x)}) = \begin{cases} -xy + log(1 + e^{x}), & \text{if } x < 0. \\ x - xy + log(1 + e^{-x}), & \text{otherwise.} \end{cases} \tag{3.3}$$

Since the original equation does not differ between positive and negative x, both of the above equations have to be equal to the original loss function.

$$-ylog(\sigma(x)) - (1-y)log(1-\sigma(x)) = ylog(1+e^{-x}) + (1-y)log(1+e^{-x}) + (1-y)x = x - xy + log(1+e^{-x}) \tag{3.4}$$

$$x - xy + log(1 + e^{-x}) = x - xy + log(e^{-x}) + log(1 + e^{x}) = -xy + log(1 + e^{x}) \tag{3.5}$$