

## Exercise 6

1. Implement Adaline algorithm to predict second column from the first column in **rabbit\_data.txt**. First column represents age of a Wild Australian Rabbit in days and second column represents weights of their eye lenses. Do you think the prediction results are good, and why / why not? Do not use existing implementations of Adaline for this task. Hint: If you implemented Perceptron algorithm last week, it is a good place to start from.
2. Using existing implementations, or your own implementation, of multi-layer neural networks<sup>1</sup> to predict the weights of the eye lenses from rabbit's age. i.e. **scikit-learn** (Python) and **nnstart** (Matlab) offer such implementations. Do the follows:
  - Create separate training and test sets from the data.
  - Train the NN on training set, and evaluate accuracy on training and testing set. Is this better than with Adaline? Visualize prediction results (draw data-points and the networks' predictions).
  - Originally data was modelled with non-linear regression line
$$y = 233.836 \cdot (1 - e^{-0.00604x}) + \epsilon,$$
where  $\epsilon$  is the error term. Compare your neural network model against this model. Which one is better?
  - By modifying the architecture of neural network or training setup, reach *overfit* and *underfit*. How can you tell it is overfit/underfit? Visualize one example of overfitting and underfitting.
  - Apply normalization on the data (i. e.  $x = \frac{(x - \mu_x)}{\sigma_x}$ ), and rerun training/evaluation. Do you get lower prediction error? Was the training faster (required less iterations to reach same accuracy)? Why so?

---

<sup>1</sup>Sidenote: Avoid using "multi-layer perceptron": a) One of the field leaders acknowledge it should have never called that (<http://www.cs.toronto.edu/~hinton/coursera/lecture3/lec3.pdf>, slide 2), and b) You may get unexpected search results for "MLP"