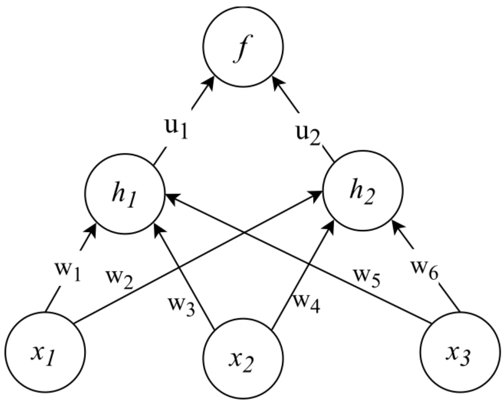
LAS Group, Institute for Machine Learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `http://las.inf.ethz.ch/teaching/lis-s16/`
Email questions to:
Felix Berkenkamp, `befelix@inf.ethz.ch`

Series 3, Apr 5th, 2017 (ANNs)

It is not mandatory to submit solutions and sample solutions will be published in two weeks. If you choose to submit your solution, please send an e-mail from your `ethz.ch` **address with subject** `Exercise3` **containing a PDF (LaTeX or scan) to** `lis2016@lists.inf.ethz.ch` **until Sunday, April 17th 2017.**

Problem 1 (Neural network derivatives):

Consider the following neural network with two logistic hidden units h_1, h_2, and three inputs x_1, x_2, x_3. The output neuron f is a linear unit, and we are using the squared error cost function $E = (y - f)^2$. The logistic function is defined as $\rho(x) = 1/\left(1 + e^{-x}\right)$.



(a) Consider a single training example $\boldsymbol{x} = [x_1, x_2, x_3]$ with target output (label) y. Write down the sequence of calculations required to compute the squared error cost (called forward propagation).

(a) A way to reduce the number of parameters to avoid overfitting is to tie certain weights together, so that they share a parameter. Suppose we decide to tie the weights w_1 and w_4, so that $w_1 = w_4 = w_{\text{tied}}$. What is the derivative of the error E with respect to w_{tied}, i.e. $\nabla_{w_{\text{tied}}} E$?

Problem 2 (Building an RBF Network):

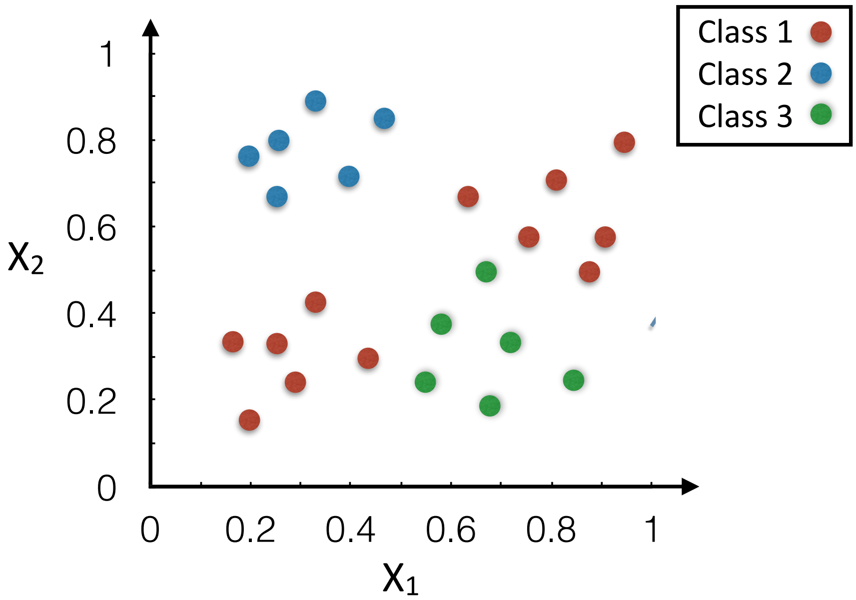
Radial basis function (RBF) networks are artificial neural networks that use radial basis functions as activation functions. They typically have three layers: an input layer, a hidden layer with a RBF activation function and a *linear* output layer. Hence, the output of the network is a linear combination of radial basis functions of the inputs and neuron parameters.

The input can be modeled as a vector of real numbers $\mathbf{x} \in \mathbb{R}^n$. Each output of the network $Y_j : \mathbb{R}^n \to \mathbb{R}$ is then given by

$$Y_j = \sum_{i=1}^{N} w_{ij} \exp(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)),$$

where N is the number of neurons in the hidden layer, μ_i and Σ_i are the mean vector and covariance matrix for neuron i, and w_{ij} is the weight of neuron i in the linear output neuron. In the basic form all inputs are connected to each hidden neuron.

Now, let us consider the following dataset:



(a) Draw an RBF network that perfectly classifies the given data points. Determine suitable values for the mean and covariance of each neuron in the hidden layer (μ_i, Σ_i and the appropriate weights w_{ij}) in the network.

Hint: You can assume that Σ_i *is a multiple of the identity matrix, so that* $Y_j = \sum_{i=1}^{N} w_{ij} \exp(-\frac{||\mathbf{x}-\mu_i||^2}{2\sigma_i^2})$.

(b) Argue why your network classifies the data points correctly. Pick one one of the data points and calculate the network output.