Machine Learning for Image Analysis, SoSe 2022
Exercise Sheet #3: Random Forest and Neural Networks

---

**Problem 1** (Random Forests: calculating impurities) (20 pt).
Consider a two class classification problem ($C = 2$). At the current node there are $N = 400$ data points of each class (denoted by $(400, 400)$). Evaluate two possible splits:

- Split A: Create two nodes with $(300, 100)$ and $(100, 300)$ data points respectively.
- Split B: Create two nodes with $(200, 0)$ and $(200, 400)$ data points respectively.

Calculate for each split the misclassification rate, the Gini impurity as well as the entropy. Which split would each criterion prefer? Remember:

$$\text{Gini impurity: } H = 1 - \sum_{c=1}^{C} p(y=c)^2, \text{ Entropy: } H = -\sum_{c=1}^{C} p(y=c) \log p(y=c)$$

**Problem 2** (Perceptrons as logic gates) (20 pts).
Perceptrons with step activation functions can implement logic gates. The step-function is given by $a(z) = \left\{ \begin{array}{l} 1 \text{ if } z \geq 0 \\ 0 \text{ if } z < 0 \end{array} \right\}$. E.g., a NOT gate for a single input $x \in \{0, 1\}$ is implemented by a perceptron with weight -1, bias 0.5, and step activation function, as $a(-1 \cdot x + 0.5) = \neg x$. Construct the following logic gates with n inputs $x_1, .., x_n \in \{0, 1\}$:

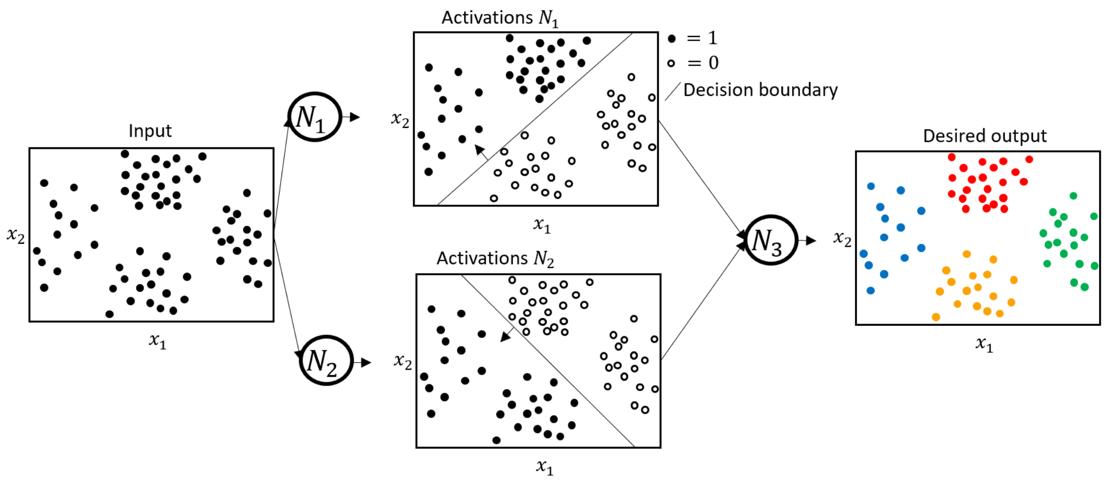
(a) $\mathrm{OR}(x_1, .., x_n)$

(b) $\mathrm{AND}(x_1, .., x_n)$

Below is sketch of a two-layer Perceptron with two Neurons $N_1, N_2$ in the first layer and their (step-) activations. Find weights and bias for neuron $N_3$ in the second layer such that $N_3$ (with step activation function) outputs

(c) 1 for the green dots and 0 for all others.

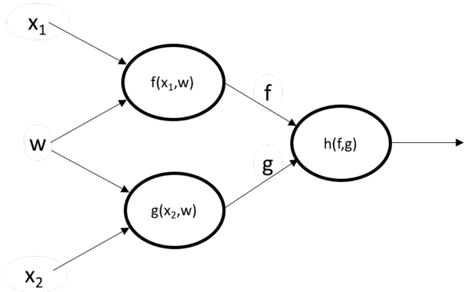
(d) 1 for the red dots and 0 for all others.




Now imagine a generalization of the above sketch into a two-layer perceptron wih M neurons in the 1st layer. As above, each such neuron defines a decision boundary in input feature space.

(e) Specify a general rule for constructing weights and bias for a neuron in the 2nd layer that outputs 1 for samples that lie inside one specific polygon of the input feature space, and 0 for all others. Hint: The firing pattern of the M layer 1 neurons for a particular polygon of the input feature space can be interpreted as a corner of an M-dimensional hyper-cube (see PR2012 Lecture 4.4). What linear decision boundary cuts off a single corner? Remember that the weight vector is orthogonal to the decision plane. Intuitively, the weight vector should point from the center of the hyper-cube to the sought corner. This will help you to construct appropriate weights. Now given these weights, what bias makes sure that only one corner is cut?

**Problem 3** (Weight sharing in backpropagation) (15 pt).
Instead of having only one node of a compute graph depend on a certain weight w, some neural networks have multiple nodes of their compute graph depend on the same w, as illustrated here:




This is referred to as *weight sharing.*

(a) What is the partial derivative of the output h with respect to the weight w?

(b) Which rule of calculus applies here?

(c) Derive the partial derivative of h w.r.t. w explicitly, as an expression over the input variables, via backprop on the compute graph, for $f(a, b) = g(a, b) = h(a, b) = a \cdot b$

(d) Derive the partial derivative of h w.r.t. w explicitly, as an expression over the input variables, via backprop on the compute graph, for $f(a, b) = g(a, b) = a + b$ and $h(a, b) = a^b$