# Project VIII Part 1: Linear Perceptron

In the lab, we work with the linear perceptron. You will implement the binary linear perceptron and test it on a toy dataset.

## 1 Linear perceptron

In this part, you are given a set of emoji samples and will implement a binary perceptron to classify them. Your task will be to complete the implementation of the 'PrimalPerceptron' class in 'LinearPerceptron.ipynb'.

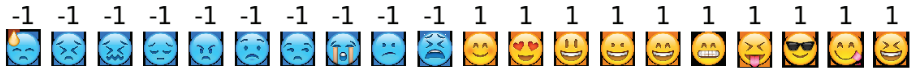

Figure 1: Example of emoji samples.

Linear perceptron uses the sign of a linear model to indicate the class of the input sample. Given a sample $\mathbf{x}$, the output labels will be either 1 or -1.

$$y = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$$

Your tasks are to:

- initialize the perceptron weights self.w and bias self.b in the __init__ function of the class.
- implement **predict(self)**, which should return both the values of $\mathbf{w}^T\mathbf{x}+b$ and the labels. The former one is named the 'preds' and the latter 'y_hat'.
- complete the **update(self)**, which searches for the optimal parameters of the linear perceptron. In the function, you enumerate the supplied samples and update the parameters. In one call of the function, you shoule enumerate the samples and update the weigths and bias.

To test your implementation, run the 'LinearPerceptron.ipynb' to get the correct classification like that in Fig. 1.