

Contents

1	Working with Unlabeled Data – Clustering Analysis	1
1.1	Grouping objects by similarity using k-means	1
1.1.1	k-means clustering using scikit-learn	1
1.2	Organizing clusters as a hierarchical tree	2
1.2.1	Grouping clusters in a bottom-up fashion	2
1.3	Locating regions of high density via DBSCAN	2

Chapter 1

1.1

1.1.1 Estimating class probabilities in multiclass classification via the softmax function

The softmax function is a soft form of the argmax function; instead of giving a single class index, it provides the probability of each class. Therefore, it allows us to compute meaningful class probabilities in multiclass settings (multinomial logistic regression).

In softmax, the probability of a particular sample with net input z belonging to the i th class can be computed with a normalization term in the denominator, that is, the sum of the exponentially weighted linear functions:

$$p(z) = \sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^M e^{z_j}} \quad (1.1)$$

1.1.2 Broadening the output spectrum using a hyperbolic tangent

Another sigmoidal function that is often used in the hidden layers of artificial NNs is the hyperbolic tangent (commonly known as tanh), which can be interpreted as a rescaled version of the logistic function:

$$\begin{aligned} \sigma_{\text{logistic}}(z) &= \frac{1}{1 + e^{-z}} \\ \sigma_{\text{tanh}}(z) &= 2 \times \sigma_{\text{logistic}}(2z) - 1 = \frac{e^z - e^{-z}}{e^z + e^{-z}} \end{aligned} \quad (1.2)$$

The advantage of the hyperbolic tangent over the logistic function is that it has a broader output spectrum ranging in the open interval $(-1, 1)$, which can improve the convergence of the backpropagation algorithm.

Note that using `torch.sigmoid(x)` produces results that are equivalent to `torch.nn.Sigmoid()(x)`. `torch.nn.Sigmoid` is a class to which you can pass in parameters to construct an object in order to control the behavior. In contrast, `torch.sigmoid` is a function.

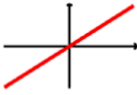
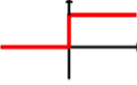
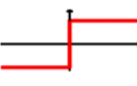


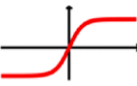

Activation function	Equation	Example	1D graph
Linear	$\sigma(z) = z$	Adaline, linear regression	
Unit step (Heaviside function)	$\sigma(z) = \begin{cases} 0 & z < 0 \\ 0.5 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	
Sign (signum)	$\sigma(z) = \begin{cases} -1 & z < 0 \\ 0 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	
Piece-wise linear	$\sigma(z) = \begin{cases} 0 & z \leq -\frac{1}{2} \\ z + \frac{1}{2} & -\frac{1}{2} \leq z \leq \frac{1}{2} \\ 1 & z \geq \frac{1}{2} \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\sigma(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, multilayer NN	
Hyperbolic tangent (tanh)	$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multilayer NN, RNNs	
ReLU	$\sigma(z) = \begin{cases} 0 & z < 0 \\ z & z > 0 \end{cases}$	Multilayer NN, CNNs	

Figure 1.1: The activation functions covered