**What is the difference between Multi Label and Multi-class classification problem?**

Multiclass classificationmeans a classification task with more than two classes; e.g., classify a set of images of fruits which may be oranges, apples, or pears. Multiclass classification makes the assumption that each sample is assigned to one and only one label: a fruit can be either an apple or a pear but not both at the same time.

Multilabel classification assigns to each sample a set of target labels. This can be thought of as predicting properties of a data-point that are not mutually exclusive, such as topics that are relevant for a document. A text might be about any of religion, politics, finance or education at the same time or none of these.

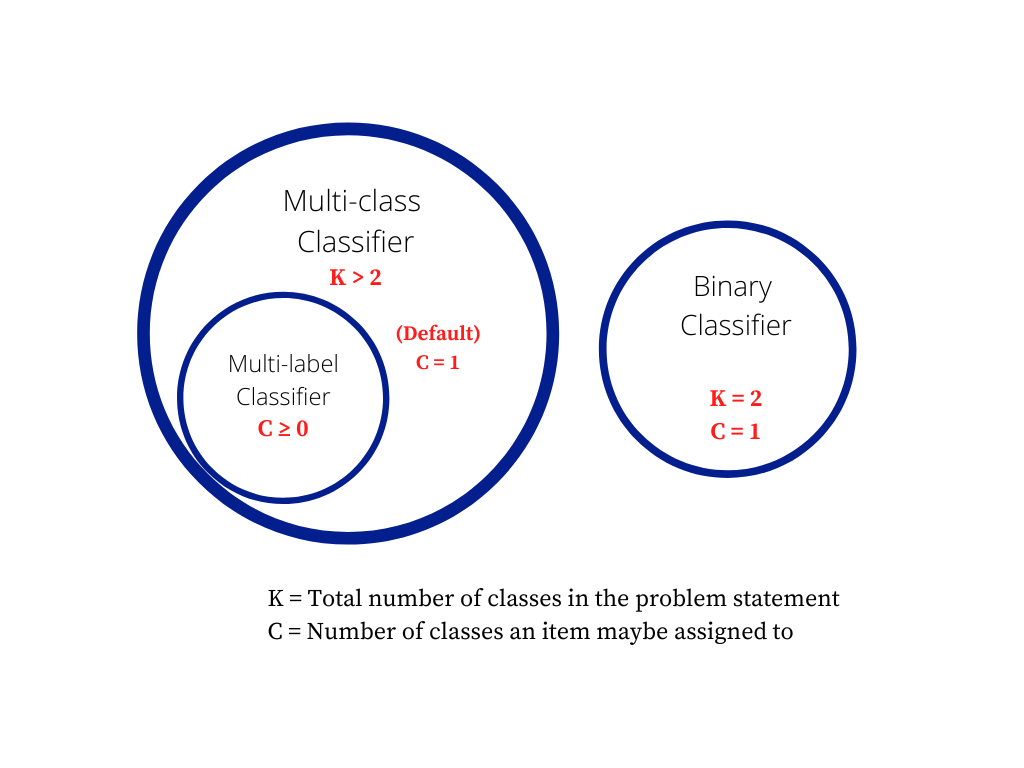
Taken from http://scikit-learn.org/stable/modules/multiclass.html

Look at these contrasts:

* Multi-class vs Binary-class is the question of the number of classes your classifier is modeling. In theory, a binary classifier is much simpler than multi-class, so it's important to make this distinction. For example, the Support vector machine (SVM) can trivially learn a hyperplane to separate two classes, but 3 or more classes makes it complex. In the neural networks, we commonly use Sigmoid for binary, but Softmax for multi-class as the last layer of the model.
* Multi-label vs Single-Label is the question of how many classes any object or example can belong to. In the neural networks, if single label is needed we use a single Softmax layer as the last layer, thus learning a single probability distribution that spans across all classes. If the multi-label classification is needed, we use multiple Sigmoids on the last layer, thus learning separate distribution for each class.

Remarks: we combine multilabel with multiclass, in fact, it is safe to assume that all multi-label are multi-class classifiers. When we have a binary classifier (say positive v/s negative classes), we wouldn't usually assign both labels or no-label at the same time! We usually convert such scenarios to a multi-class classifier where classes are one of {positive, negative, both, none}. Hence multi-label AND binary classifier is not practical, and it is safe to assume all multilabel are multiclass.

On the other side, not all Multi-class classifiers are multi-label classifiers and we shouldn't assume it unless explicitly stated.

****

**What should the final activation layer be?**

The probabilities produced by a sigmoid are independent, and are not constrained to sum to one: 0.37 + 0.77 + 0.48 + 0.91 = 2.53. That’s because the sigmoid looks at each raw output value separately.

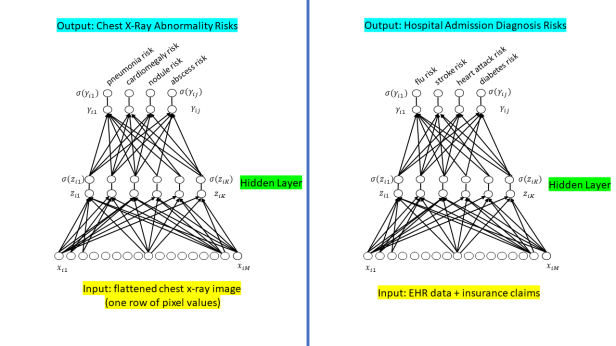
In contrast, the outputs of a softmax are all interrelated. The probabilities produced by a softmax will always sum to one by design: 0.04 + 0.21 + 0.05 + 0.70 = 1.00. Thus, if we are using a softmax, in order for the probability of one class to increase, the probabilities of at least one of the other classes has to decrease by an equivalent amount.

**Sigmoid Examples: Chest X-Rays and Hospital Admission**

Chest X-Rays: A single chest x-ray could show many different medical conditions at the same time. If we build a classifier for chest x-rays, we want that classifier to be able to indicate that multiple conditions are present. Here’s a chest x-ray image showing both pneumonia and abscess, and the corresponding label, which you’ll notice has multiple “ones” in it:

Hospital Admission: Given a patient’s health records, we might want to predict whether that patient will be admitted to the hospital in the future. We can frame this as a classification problem: classify a patient’s past health record according to their future hospital admission diagnoses (if any.) The patient might be admitted for multiple diseases, so there is possibly more than one right answer.

Diagrams: The picture below shows two feedforward neural networks, corresponding to these two classification problems. At the end, a sigmoid function is applied to the raw output values to obtain the final probabilities and allow for more than one correct answer – because a chest x-ray can contain multiple abnormalities, and a patient might be admitted to the hospital for multiple diseases.



**Softmax Examples: Handwritten Digits and Irises**

Handwritten Digits: If we are classifying images of handwritten digits (the MNIST data set), we want to force the classifier to choose only one identity for the digit by using the softmax function. After all, a picture of the number 8 is only the number 8; it cannot be the number 7 at the same time.

***Based on the above explanation the choice for final activation layer for our problem will be Sigmoid***

For a multi-class classification problem we use softmax activation function. It is because we want to maximize the probability of a single class and softmax ensures that the sum of the probabilities is one. However we use sigmoid activation function for the output layer in multi-label classification setting. What sigmoid does is that it allows you to have a high probability for all your classes or some of them, or none of them.

**What should the loss function for a multi class classification problem using neural networks?**

For a multi-class classification problem we often use *categorical\_crossentropy* loss. This is useful since we are interested to approximate the true data distribution (where only one class is true). However in multi label classification setting we formulate the objective function like a binary classifier where each neuron(*y\_train.shape[1]*) in the output layer is responsible for one vs all class classification. *binary\_crossentropy* is suited for binary classification and thus used for multi-label classification.

**How should the multi classes in a multi label of the Y-variable ( Target variable) be encoded for multi label classification problem for neural networks?**

We usually one hot encode our labels for multi-class classification problem. By one hot encoding we represent the categorical variables as binary vectors. We first map categorical values to integer values. Then, each integer value is represented as a binary vector where all values are zero except the index of the integer, which is marked with a 1.

However we know that multi-label classification problem we can have any number of classes associated with it. We strongly assume that the labels are mutually exclusive. Thus instead of one hot encoding we do **multi label binarization**. Here the label (can have multiple classes) is transformed into a binary vector such that all values are zero except the indexes associated for each class in that label, which is marked with a 1.

We can easily implement this as shown below:

from sklearn.preprocessing import MultiLabelBinarizer

# Create MultiLabelBinarizer object

mlb = MultiLabelBinarizer()

# One-hot encode data

mlb.fit\_transform(y)