GROUP 13, GALL & CORONEL

What is image classification?

Image classification is the task of assigning an image to exactly one class from all possible classes. In image classification, an image is classified to a class according to a previous set of images with known class. This comparison might be simply comparing the image in question with all others from the training dataset (as we did with KNN in this assignment) or using tasks of higher complexity, where the training dataset is used to obtain the weights to fit a model (e.g. neural network).

What is the purpose of the training, validation, and test sets and why do we need all of them?

The training set is the dataset of which a model learns from. The weights of the model, in the case of Neural Networks, are adjusted according to the training set. Or in another way to say it, the training set is use to fit the model. The validation set is used to evaluate the model during training. The results of the validation will give an idea if the training is going bad or not. For example, when the accuracy is going up in training but decreasing in the validation set, then the model is overfitted. Furthermore, hyperparameters are fine-tuned by evaluating different sets of hyperparameters by testing the model with the validation test sets. After training the final model using the training and validation sets, the test set is then used to test the model. It provides information on the model performance in practice without bias as the test set is a new set of data that the model has never seen before.

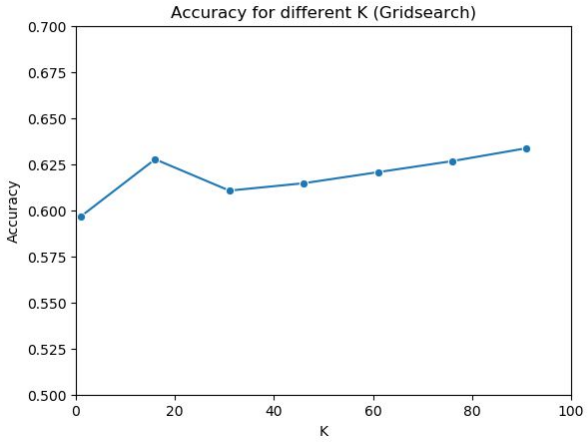
How do KNN-Classifiers Work

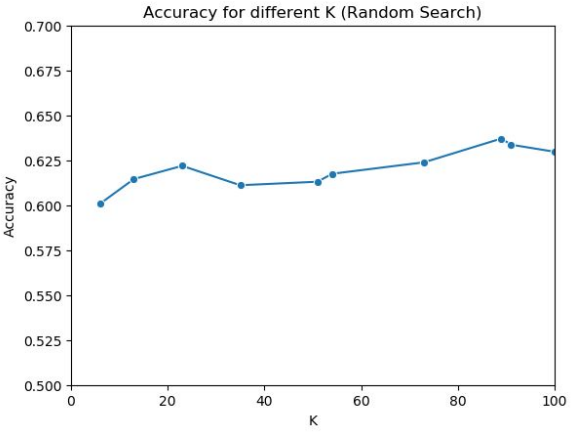
The idea of KNN is that a sample is classified according to the majority vote of the class of its *K* neighbours where neighbours are the samples nearest to it. And nearness is defined by the distance measure like L1-, or L2-norm (other distance measures are available).

For example, a sample is described by its feature vector with dimension n. And this vector is a point in an n-dimensional space. Given all samples are in this n-dimensional space, KNN follows the principle that samples of the same class should have a smaller distance to each other than to those samples belonging to another class. When classifying a new sample of unknown class, the new sample is compared to every training sample thus training in KNN is just storing all the training dataset. Comparing means measuring the distance between two feature vectors using some distance measure like L1-, or L2-norm. The K samples (=neighbours) from the training set with the smallest distances to the new sample are used to calculate the new sample's class. The usual implementation is to assign the new sample's class by majority vote of the K neighbours' class. Thus using an uneven K for binary classification makes sense because for tied results e.g. k=4, classes={0,0,1,1} an additional decision rule has to be implemented.

Applying this to cats vs dog image classification, each image from its 32x32x3 dimension (BGR) is transformed to an feature vector of 3072 in length. Distance was measured using L1-Norm. For example, using K=4, a new sample has 3 dogs and 1 cats as neighbours, therefore the class of the new sample is dog. However, instead of just the frequency of each class among the K neighbours, we calculated the probability distribution by using the softmax function.

Also include your results obtained from knn_cats_dogs.py. Include the validation accuracies for the different k you considered as a table or (better) a plot. Also include the final test accuracy, compare the best validation accuracy and the final test accuracy, and discuss the results.





By Grid Search

K	1	16	31	46	61	76	**91**
Val. Acc.	0.597	0.628	0.611	0.615	0.621	0.627	**0.634**

Best k = 91, Final test accuracy = 0.616

By Random Search

K	6	13	23	35	51	54	73	**89**	91	100
Val.Acc	0.601	0.615	0.622	0.611	0.613	0.618	0.624	**0.637**	0.634	0.630

Best k = 89, Final test accuracy = 0.615

From the 7 values of K tested by grid search, the best K is 91. Using random search to find the best K, K is 89. A larger value of K tends to generalize more (smoother boundary) while a lower value of K tends to be more sensitive to noise. However, at some point with a high value of K the model is underfitting and accuracy would go down as well. Comparing the validation accuracy to the final test accuracy, there's not much of a difference, the model performed similarly to the validation and test dataset but ultimately, KNN does not solve the problem of classification between cats and dogs very well.

In this example, we have a very high dimensional space of n=3072. There just isn't enough data to counteract the curse of dimensionality. Given the results of accuracy in the 0.6 range, it's obvious that KNN is not the best classification method for this type of problem. KNN has no understanding of the features and measuring the distance between the raw pixel values is not enough. The distance measure treats each feature the same way, be it a pixel belonging to the background or a pixel belonging to the cat or dog. This means, if two images have the same colored background the distance to each other will be close.