HW5

Wednesday, April 13, 2022 6:41 PM

1) Explain why it is important to reduce the dimension and remove irrelevant features of data (e.g., using PCA) for Instance-Based Learning such as kNN? (5 points)

High dimensional data can cause problems for many machine learnings algorithms, including Instance-Based ones like kNN. The curse of dimensionality shows how randomly sampled high-dimensional vectors are usually sparse, which can increase the risk of overfitting. A way to solve this is by increasing the amount of training data. However, this is generally not feasible which is why dimensionality reduction is the next best alternative. Instance-Based Learning algorithms are noise-sensitive, so doing dimensionality-reduction to remove irrelevant features & noise allows the algorithm to focus on only the relevant data.

2) One limitation with K-Means is the variability issue. Explain how to address this problem. (5 points)

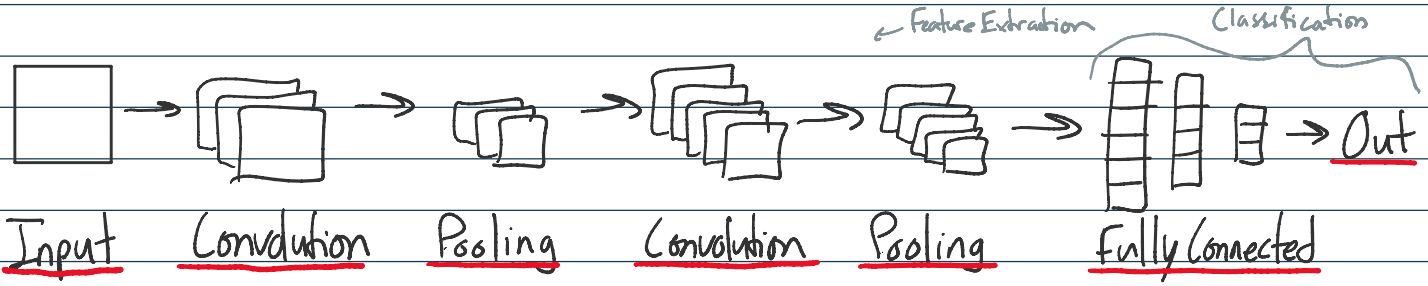
The K-Means algorithm works best under certain conditions. These conditions include the clusters having equal variances & similar sizes & shapes (ideally spherical). Without these conditions the algorithm will fail to converge to an optimal solution. Depending on the location of the initial centroids, the algorithm may find non-ideal clusters. A way to address this issue is to adjust the location of the initially centroids. One can run the algorithm several times with different centroids to see which will give the best solution.

3) Please explain the technique of Gaussian Mixture and how it is used for anomaly detection. (5 points)

A Gaussian mixture model (GMM) is a probabilistic model that assumes that the instances were generated from a mixture of several Gaussian distributions whose parameters are unknown. You assume that the data is grouped into a finite number of clusters, each with an ellipsoidal shape, & we do not know which cluster each image belongs to. The Expectation-Maximization (EM) algorithm is used to find the ideal parameters in an interative process. First, random Gaussians are initialized with equal variances. Then, the posterior probabilities (responsibilities) are evaluated from the current Gaussians (E-Step). Next, new Gaussians are re-estimated using the current responsibilities (M-Step). The E+M steps are repeated until convergence. It is essentially a generalization of K-Means. GMM can be

estimated using the current responsibilities (M-step). The E and M steps are repeated until convergence. It is essentially a generalization of K-Means. GMM can be used for many applications, such as anomaly detection. With GMM, anomalies would be any instance located in a low-density region.

4) Please draw the diagram of Convolutional Neural Networks (CNN). Then explain the functionality of each layer of CNN. Name several latest algorithms of CNN (e.g., AlexNet etc.). (5 points)



- Convolution: Used to extract features from the input. Convolutions are performed between the input/previous layer & a filter of a particular size. The filter slides over the input & the convolution product between the feature & each portion of the input is calculated. This results in a feature map, which tells us where the features are in the input. These features are learned during training (the filter kernels are updated during backpropagation).
- Pooling: Layer that usually follows convolution. It receives several feature maps & applies pooling to each, which shrinks the input in order to reduce computational load, memory usage, & the number of parameters. It subsamples while also preserving important characteristics. The output has the same number of feature maps as the input but they are much smaller.
- Fully Connected: The last layers of a CNN (or any neural network). Receives an input vector & applies a linear combination/activation function to produce a new output vector. The matrices from the previous layer are flattened into a vector & the relationship between the position of features in the input & a class are determined for classification.

There are many different CNN architectures, including:
- LeNet-5
- AlexNet
- GoogLeNet
- VGGNet
- ResNet
- Xception
- SENet

5) What are the vanishing and exploding gradients problems in Backpropagation? Name several techniques to address these problems. (5 points)

The vanishing gradient problem is when the gradient get smaller & smaller toward lower (earlier) layers, leaving these layers' weight virtually unchanged, resulting in the training never converging to a good solution. The exploding gradient problem is similar, except here the gradient grow bigger & bigger & the weight updates become very large, resulting in the algorithm diverging. There are many ways to fix this:

- Improving weight initialization → several techniques can be employed to help make the variance of the outputs of each layer to be equal to the variance of its inputs.
- Using non-saturating activation functions → ReLU, Leaky ReLU, ELU, & SELU can all be used to help prevent the activation functions from saturating easily at the top layers (where the gradients are very small).
- Batch Normalization → makes inputs of each layer zero-centered & normalized by adding an operation before or after the activation function at each layer.
- Gradient Clipping → clip the gradients during backpropagation so they never exceed some threshold.

2) Consider a learned hypothesis, *h*, for some Boolean concept. When *h* is tested on a set of 100 examples, it classifies 80 correctly. What is the 95% confidence interval for the true error rate for $Error_D(h)$?

95% CI → $1-\alpha = 0.95$ → $\alpha/2 = 0.025$

$Z_{\alpha/2} = Z_{0.025} = 1.96$ ★ From Z-tables

Standard Error $E = \sqrt{\frac{Error_S(h)\cdot(1-Error_S(h))}{n}}$

$= \sqrt{\frac{0.8\cdot(1-0.8)}{100}} = 0.04$

★ $error_S(h) \pm Z\sqrt{\frac{error_S(h)(1-error_S(h))}{n}}$

Lower Bound = $0.2-(1.96\cdot 0.04) = 0.1216$

Upper Bound = $0.2+(1.96\cdot 0.04) = 0.2784$

=> 95% CI for $Error_D(h) = (0.1216 < Error_D(h) < 0.2784)$