Assignment 3

Bendik Nordeng (Student ID: 478128, user name: bendikno)
Peder Møyner Lund (Student ID: 478109, user name: pederml)

October 2019

1 Theory

1.1 Deep learning

Deep learning is a field within machine learning inspired by the structure of the human brain. When we deal with deep learning, we usually talk about artificial neural networks. Deep learning involves multiple levels of representation and multiple layers of non-linear processing units, often called neurons. The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. In contrast, all non deep learning approaches can be qualified as shallow learning. The earliest neural networks, which didn't have the capacities and potential for learning as today's networks, often contained few hidden layers (typically just 1). Hence, these were shallow neural networks. Since there is no universally agreed upon threshold of depth that divides shallow learning from deep learning, the term is somewhat simplistic and abstract. Deep learning networks usually takes longer time to train, increasing with amount of data and number of layers in the network. Deep models are also able to extract better features than shallow models.

1.2 Comparison of different machine learning techniques

1.2.1 K-NN

K-nearest neighbours (K-NN) algorithm is a non-parametric unsupervised machine learning algorithm that can be used to solve classification problems. It is intuitive and easy to implement and requires no training period as it is a lazy learner. K-NN is suitable when the task involves labelling continuous data into different groups on small data sets. An example could be to find

different categories of consumers based on shopping history. However, k-NN has its disadvantages. Firstly, k-NN does not work well with large data sets. Secondly, k-NN does not work well with complex problems with many dimensions as its similarity measures are based on a simple distance measure. Thus, k-NN would not be suitable for unstructured learning problems with vast amounts of image or video data sets.w

1.2.2 Decision tree

Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Like k-NN, decision trees have its advantages of being simple and intuitive. The method is very useful when one tries to discover a result given a range of factors or the outcome based on a sequential process. An example of this could be to evaluate whether or not a person should be given a bank loan based on factors such as income, gender, loan defaults etc. Using decision tree on this problem would output intuitive patterns that could help to understand certain outcomes. A disadvantage of decision trees is that decision tree training is relatively expensive, and bias in the training set can cause big changes to the decision tree during training. Thus, decision trees would not be suitable on problems that does not have concrete input and output data, or problems with too many input parameters due to computation complexity. An example of an unsuitable problem for decision trees could be unstructured learning problems with vast amounts of image or video data sets.

1.2.3 SVM

A Support Vector Machine (SVM) is a non-parametric, supervised classifier that classify based on a separating hyperplane. The algorithm outputs an optimal hyperplane which categorizes new examples. It can perform both linear and non-linear categorization. SVM works well for classification in higher dimensions and when there is a clear margin of separation between the cases. It can be applied on problems related to text categorization. As other supervised learning methods, it is dependent on labelled data. Thus, a disadvantage of the method is that it is not able to find natural clustering of the data to groups. Furthermore, SVM is not as intuitive as k-NN and logistic regression, especially in higher dimensions. Moreover, SVM does not perform well when there is noise and overlap in the training data. Thus when

it is important to understand the match between input variables and effect on the output, decision trees might be more useful, but SVM has proved to give better results in higher dimensions than other supervised learning methods.

1.2.4 Deep learning

As mentioned in task 1, deep learning involves multiple levels of representation and multiple layers of non-linear processing units, often called neurons. The method uses multiple layers to gradually extract higher level features from the raw input. The most crucial advantage of deep learning is its ability to utilize unseen patterns of unstructured data. By filtering and treating data through many layers, deep learning is able to uncover hidden patterns in a way that the previous discussed supervised learning methods are not able to in the same extent. An example of this is how different layers in a deep neural network can discover different kind of patterns in a picture. One level could focus on finding the edges while another may identify faces in a good way. Furthermore, deep learning models are very general and good models can be used in a vast number of different applications. However, deep learning have some disadvantages. Firstly, it may be complex to configure. Secondly, the training time could be extensive. Thirdly, it requires large amounts of data. In summary, deep learning should be used on complex problems with unstructured data while it would be more practical with the discussed supervised methods on simpler clustering and classification problems.

1.3 Ensemble methods

What are ensemble methods and when do we use them? Ensemble methods are machine learning techniques that combines several base models in order to improve the overall performance. The idea originates from the idea that no single algorithm performs best for all kind of problems. Thus, by combining multiple independent methods, the correct decisions are reinforced, and the random errors diminish. 3 types of ensemble machine learning methods

1.3.1 Random forest

Random forest is a supervised learning algorithm used for classification problems. The algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means

of voting. It is an ensemble method which is aimed to reduce overfitting and variance using the general technique of bagging to tree learners.

In brief, the forest is based on the following steps:

- Generate a bootstrapped dataset
- Generate a decision tree using the bootstrapped dataset, but only use a random subset of variables at each step. Do this on a wide range of subsets.
- Then for the prediction step, for each example find the prediction result for each of the generated trees in step 2.
- Find the aggregated voting scores from the different trees in order to find the final prediction result

This bootstrapping procedure leads to better model performance than single decision trees because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated.

1.3.2 AdaBoost

AdaBoost is a popular boosting technique which combines multiple weak classifiers into a single strong classifier by adjusting the error metric over time. The algorithm is characterized by having stumps (one level decision trees) as weak classifiers, some stumps get more say in the classification than others and each stump is formed by taking the previous stump's mistakes into account. In contrast to random forest, AdaBoost are made up of several stumps (one variable with two outcomes) and not different number of variables. Furthermore, in random forest, each tree has an equal saying in the final voting. This is not the case in AdaBoost. Lastly, in a random forest, each decision tree is made independent of the others. In contrast, in a forest of stumps, the order is important. In brief, the algorithm consists of the following steps:

- Chose a chosen set of iterations (T) and set a uniform weight of the training examples
- Iterate T times and for each iteration
  - Pick the stump that minimizes the classification error

  - Compute the weight of the classifier chosen
  - Update the weights of the training examples based on the classification of the

- Combine the classifiers. Each boosting iteration learns a new (simple) classifier on the weighed dataset. These classifiers are weighed to combine them into a single powerful classifier.

1.3.3 Gradient Boosting

Like with AdaBoost, the key with Gradient boosting is learning from previous mistakes, in this case from residual error directly rather than updating the weights of data points. The algorithm consists of the following steps:

- Train a decision tree
- Use the trained tree to predict and save the residual errors as the new y (the next decision tree will be trained on this)
- Repeat until the number of trees we set to train is reached
- Make the final prediction

The Gradient boosting algorithm thus makes a new prediction by simply adding up the predictions (of all trees).