Assignment 4 ML

J L Gouws
19G4436

April 19, 2022

Task 1

1. The weights from the first layer to the second layer(first hidden layer) are supposed to represent the contribution of the pixels to little line segments. The sencond layer is supposed to pick up on small “edges" or line segments.

2. The weights from the second layer to the third layer represent collection of the “edges" making up larger structures. These weights are supposed to represent the contribution of the small line segments to lines and circles. The weights for input into the final layer are supposed to represent how the lines and circles get added together to form the numbers.

3. The weights are not coded or worked out by hand to represent this. A computer works out the weights. The network needs some starting values for the weights, these are set randomly for ease of use. Since the network starts randomly there is no reason for the training network to follow a human’s semantic structure. The learning tries to find patterns that are natural for the network to recognize, which might not be natural/sensical to humans.

Task 2

1. Machine learning is the study of finding or making a function f that takes input x(feature vector) from some set D and gives output y, where nothing is known about the relationship between x and y a priori. Supervised learning is the case where y is known for some subset of D. Unsupervised learning is the case where nothing is known about the ouput of the function. An example would be taking a picture, x, of a dog or cat and finding a function, f, to classify the picture as a picture of a dog or cat, y.

2. (a) Feature scaling is changing the values of (numeric)features so that they conform to some restriction, without significantly changing the information encoded within the features.

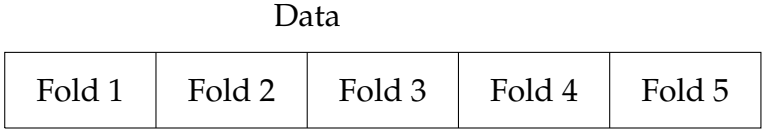
   (b) Distance based learning algorithms.

   - K Nearest Neighbors
   - Neural networks
   - State Vector Machines

   (c) Tree and ensemble based algorithms

   - Random forest
   - XGB

3. (a)



The feature/training data set is partitioned into various folds, as seen in the diagram above for five folds. The cross validation iterates through the partitions. For each iteration one fold is chosen as a test set and the other partitions are grouped into one training set. The cross validation then trains a model on the training partitions and evaluates the trained model on the test partition. This gives an estimate of generalization error of the model.

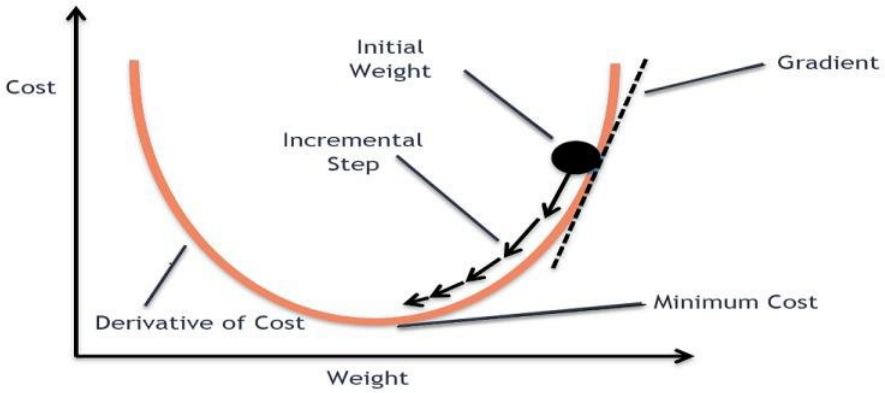
(b) Large number of folds:

Pro Bias of true error rate is small.
Con Computationally expensive.
Con Large variance in estimator.

Small number of folds:

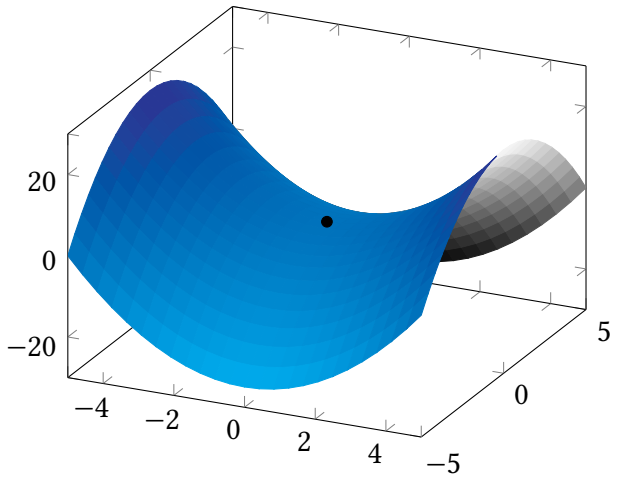
Pro Computationally cheaper than large number of folds.
Pro Low variance of estimator.
Con Error estimator has high bias.

4. (a) Gradient descent tries to find the minimum in some function, see the figure below.



From `https://www.researchgate.net/figure/Gradient-Descent-Algorithm-26_fig1_352019480`

This is used in machine learning to minimize a cost function, the cost function should be low when the machine learning model performs well and high when the model performs badly. The gradient descent method finds the direction in which the cost function decreases the fastest, and takes a step in that direction. The gradient descent method might only find a (poor) local minimum and not the global minimum. The gradient descent method might also find a saddle point in the cost function, see black dot in the figure below.



A potential solution to this problem is to find a cost function with one minimum, but this could be very complicated, if not impossible, for a particular problem. Another problem is that one has to worry about setting a learning rate for model updates. Small

learning rates result in slow solution and high learning rates can miss the minima in the cost function. This can be solved by giving the gradient descent solver a momentum which essentially adaptively modifies the learning rate to a suitable value.

(b) By looking at the error data, overfitting and underfitting of the model can be seen. If the error of the model on training data decreases, but the generalization error increases the model is overfitting. If the model has a high overall error on both training and test sets, and the error does not decrease with training, the model is probably too simple and so underfits. If the model is overfitting then the model is too complex, a lower degree polynomial is required. If the model is underfitting the model is too simple and a higher order polynomial is required.

(c) By looking at the error data, overfitting and underfitting of the model can be seen. If the model is overfitting then more regularization is needed. If the model is starting to underfit then less regularization is needed.

Task 3

I could not register my phone number on kaggle. I did everything on google colab.
`https://colab.research.google.com/drive/1XWlv3AMF0u4PEOcB884lvLg9KlY_jZ3Q?usp=sharing`

Task 4

`https://colab.research.google.com/drive/16MJyzzhjJo2CmgCHRiXDv1VCaXmCH8C7`