ECS708 Machine Learning

Feedback sheet for Assignment 1

1. General Comments

There were several questions where we ask to change a value and comment. Some students use only a few values, all in the same range (for instance lambda=0.001 or 0.002 or 0.01) Do not hesitate to try a very wide range of values to make the effects visible (for instance lambda=0.001 or lambda=1 or lambda=100). Typically one should initially start with values of lambda that are powers of 10 to get to the right range.

Some students did not utilise “for” loops, or vector computations. At the extreme students copy-pasted code instead of creating loops – this is not good programming practice. The best practice is to use vector computations instead of iterating over all the elements of a vector. Having said that, performance-related practices were not penalised.

Some students didn’t adhere to the instructions and did not include the code, or included only part of it, or didn’t include figures and graphs when they were instructed to.

2. General Comments

Task 2

When predicting values for the price of houses, don’t forget to normalise the input data using the same normalisation as done during training.

Task 3

The report should have highlighted:

- for the alpha value:
    - alpha too small: converges too slowly
    - alpha OK: good convergence within 100 iterations
    - alpha too big: doesn’t converge at all
- for the lambda value:
    - lambda too small: the curve goes through all the points, bad generalisation (overfitting)
    - lambda OK: the curve still follows the points, but not exactly, better generalisation
    - lambda too big: the curve is flat and doesn’t follow the points anymore (underfitting)

Task 5

When asked to draw the boundary, many students forgot to add the bias term.

Task 7

When more features are added, the error is supposed to decrease, as we have a model that uses non-linear features and, generally speaking, more parameters.

Task 8

We should observe the following behaviours:

- Small training dataset: the test error is stable, but high. The system generalises badly because there are too few training examples, it overfits on the few examples that were picked for training.
- Medium training dataset: the test error is stable and lower, there is better generalisation
- Large training dataset: the test error increases throughout iterations, there is overfitting.

When adding third-order features, we see the same behaviour, but there is more overfitting (due to an even more complex model)

Task 12

The output of a regression unit is binary. To discriminate between 3 classes using a regression unit, we would have to make 3 classifiers, to classify each of the 3 classes against the 2 others, in other words, 1-vs-all. We could also have used three 1-vs-1 classifiers. Several students suggested using 1-of-K encoding – this is a way of dealing with categorical input and not dealing with categorical output.

With neural networks, the output is also binary, but we can have as many outputs as we want, so we can have 3 output neurons, each one solving a binary classification problem corresponding to each of the three classes.

Task 13

When backpropagation was implemented properly, we should observe that the error with 1 neuron is high, but the errors with 2 or more neurons are comparable. A good choice is then to use a model with 2 or 3 neurons to be computationally efficient and reduce the risk of overfitting.