Machine Learning - Assignment 5

Christoffer Thrysøe - dfv107

February 20, 2017

1. Neural Networks

The implementation for the multilayer neural network is located in the file `NN.py`, the network uses the following activation function for forward propagation:

$$h(a) = \frac{a}{1 + |a|} \tag{1}$$

The derivative of (**??**) is defined as the following and is used for backward propagation:

$$h(a) = \frac{1}{(1 + |a|^2)}$$

The implementation uses a single hidden layer, and takes as input the number of hidden units at the layer. The implementation uses gradient decent as described in the assignment text.

Gradient verification

The gradient computation has been verified by calculating the numerically estimated partial derivatives using the following RHS calculation, and verifying the proximity.

$$\frac{\partial E(\mathbf{w})}{\partial [\mathbf{w}]_i} \approx \frac{E(\mathbf{w} + \epsilon \mathbf{e_i}) - E(\mathbf{w})}{\epsilon} \tag{2}$$

The gradient verification has been implemented in the function `gradient_verify`, the gradient has been compared using the first 10 data points of the file `sincTrain25.dt`, the gradients have been estimated performing a single backpropagation iteration. The result from (**??**) has been confirmed to be less than 10^{-8}, when setting $\epsilon = 0.00000001$.
Different plots are given in the next section.

Neural network training

The neural network has been trained using the training data `sincTrain25.dt`. Batch gradient learning has been applied when training the model, using backpropagation. Besides training the data, the resulting weights of the backpropagation procedure is used for evaluating the validation error, which is done by performing a forward propagation using the validation data from the set `sincValidate10.dt` and the weights of the network. A network with 2 and 20 hidden units have been tested, each with a learning rate of

$0.001, 0.01, 0.1$. Figure **??** shows the different learning rates with 2 hidden units, figure **??** shows the same with 20 hidden units.

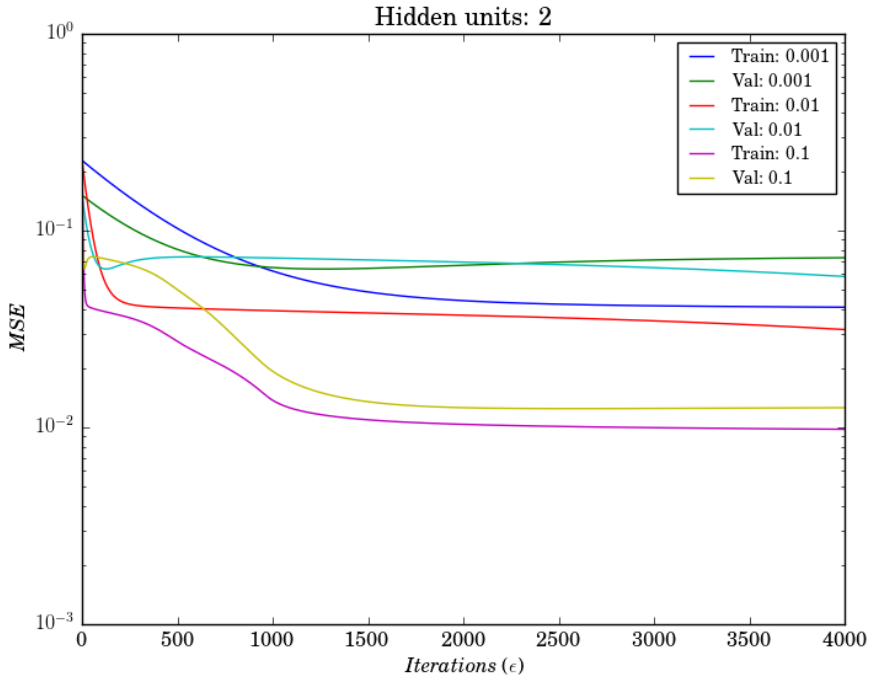

Figure 1: MSE for different learning rates using 2 hidden units

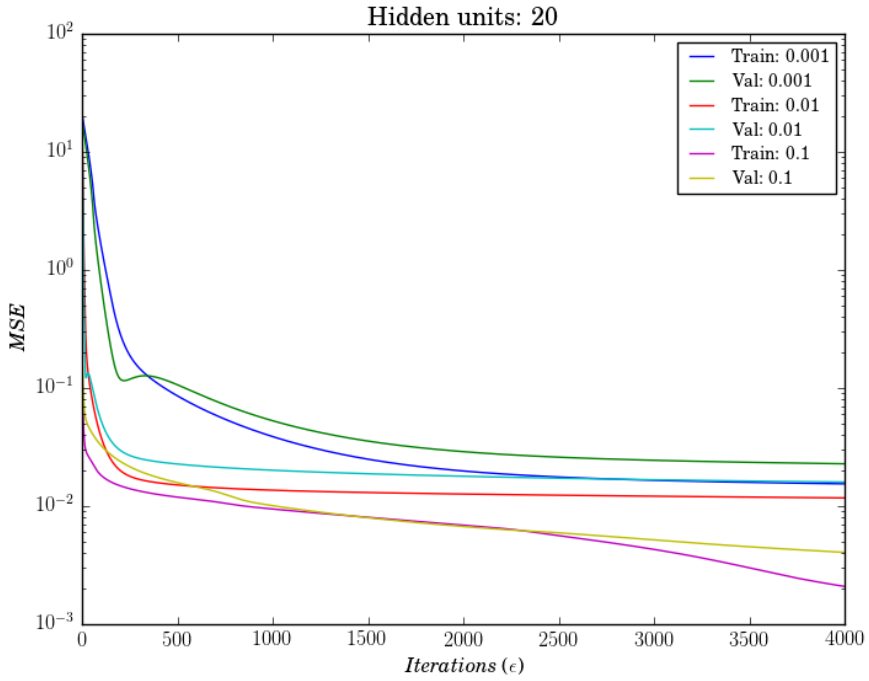

Figure 2: MSE for different learning rates using 20 hidden units

As evident from figure **??** and **??** the validation error converges nicely in a decreasing manner, this is good as it indicates a better generalization of the data. It's clear that the network using 20 hidden units leads to a better generalization of the data as the validation error using a learning rate of 0.1 produces the lowest error of all the validation sets in both figures. It is also noticeable from the figures that choosing a very low learning rate causes the network with 2 hidden units to barely learn at all, while a decrease in the validation error is evident when using 20 hidden units event for low learning rate. The stopping criteria for the model is simply when all iterations have been performed. This could have been changed to early stopping by identifying when the validation error stops decreasing/starts increasing over a number of iterations. However as no apparent over fitting is present this was not implemented. A concern regarding applying early stopping was when to stop the iterations as the error function decreases twice over the iterations, which may be the result of the gradient decent procedure being located in a local minima, and we want to wait for the procedure to locate the global minima, which is not possible if the early stopping procedure forces a stop at a local minima.

Figure **??** shows a plot of the network functions, with different learning rates with 2 hidden units, the learning rate of 2.0 has been introduced to visualize what happens with big learning rates, also the learning rate of 0.01 has not been plotted for clarity. The underlying function which the toy data was generated upon is also plotted. We want to produce a model which captures the underlying structure of the data, thus the model which fits the function the best. Clearly small and large learning rates do not produce a well generalization. Choosing a low learning rate causes the function to capture close to nothing of the structure of the data, thus under fitting the data. A large learning rate captures the data nicely to start with, however it diverges away from the data at the end. This is probably caused by the gradient being stuck at a local minima which it is not able to escape from / jumping between local minima, due to the high learning rate. From figure **??** we see that the learning rate which fits the data the best is 0.1.

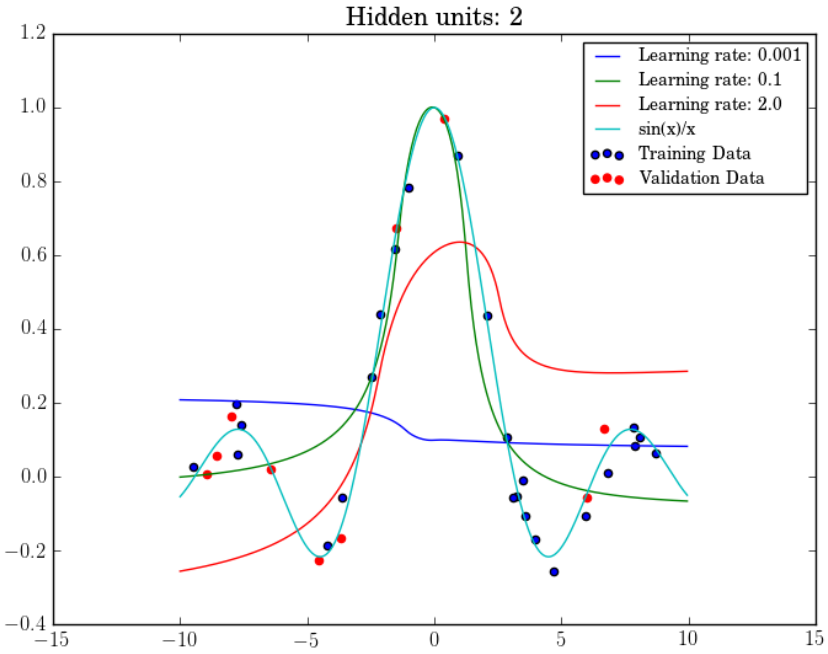

Figure 3: Different network functions plotted amongst the training data and the validation data, using 2 hidden units. The function from which the data was generated is also visualized.

Figure **??** shows the same interval plotted using 20 hidden units, this has been added to show the learning effect of applying more hidden units.

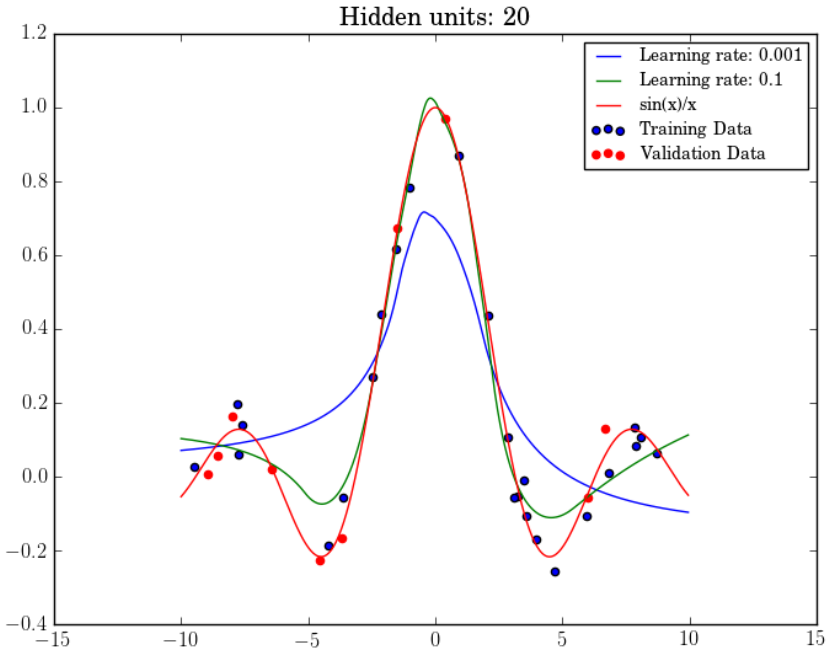

Figure 4: Different network functions plotted amongst the training data and the validation data, using 20 hidden units. The function from which the data was generated is also visualized.

We see that the learning function now tracks the underlying distribution even better. It is also evident from the training and validation points that an over-fitting is very unlikely as the points don't have a lot of noise and are very close.

2. The growth function

1. We have that $|\mathcal{H}| = M$, the number of all possible dichotomies is 2^N given a set of size N. If $M \leq 2^N$ then M bounds the growth function as M hypothesis can generate at most M dichotomies, and each hypothesis can generate at most one unique dichotomy, therefore $m_{\mathcal{H}}(N) \leq M$.
   If $M \geq 2^N$ then the number of hypothesis is larger than the number of possible dichotomies. If the hypothesis shatters the sample set, then there are 2^N dichotomies, as some hypotheses generate identical dichotomies, therefore $m_{\mathcal{H}}(N) \leq 2^N$. Which gives the following bound:
   $$m_{\mathcal{H}}(N) \leq min\ \{M, 2^N\} \tag{3}$$
   thus concluding the proof.
   The VC-dimension of $\mathcal{H}$ is the largest sample size for which the data is shattered by M hypothesis. We wish to provide an upper bound on the dimension as we do not know the break point (if any) of the data. Thus we take the largest number of sample size which can be shattered by M hypotheses. We get the following bound:
   $$d_{VC} \leq lg(M) \tag{4}$$
2. We wish to show that $m_{\mathcal{H}}(2N) \leq m_{\mathcal{H}}(N)^2$. First we note that the growth can be described in terms of the VC-dimension:
   $$m_{\mathcal{H}}(k) = \begin{cases} 2^k & k \leq d_{\text{VC}} \\ 2^k - c(k) & \text{otherwise} \end{cases}$$

that is the growth function takes the value 2^k until a break point is met. We want to show that for different sizes of d_{VC} the following is met, for some increasing cost function c:

$$\begin{aligned} m_{\mathcal{H}}(2N) &= \max_{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_{2N}\in\mathbf{X}} |\mathcal{H}(\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_{2N})| \\ &\leq \left(\max_{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_N\in\mathbf{X}} |\mathcal{H}(\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_N)|\right)^2 = m_{\mathcal{H}}(N)^2 \end{aligned}$$

We must consider the following cases:
If $d_{VC} \leq N$ we have that the breakpoint is is in the first N elements:

$$m_{\mathcal{H}}(2N) = 2^{2N} - c(2N) \leq (2^N - c(N))^2 = m_{\mathcal{H}}(N)^2$$

if $N < d_{VC} \leq 2N$ we have that that the breakpoint is between N+1 and 2N:

$$m_{\mathcal{H}}(2N) = 2^{2N} - c(2N) \leq 2^{2N} \leq 2^{2^N} = m_{\mathcal{H}}(N)^2$$

Finally, if $2N < d_{VC}$ we have that the breakpoint is after the first 2N elements, or there is no breakpoint:

$$m_{\mathcal{H}}(2N) = 2^{2N} \leq 2^{2^N} = m_{\mathcal{H}}(N)^2$$

For $N > 1$, thus concluding the proof.

3. TBD

4. Rewriting the following theorem:

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{k-1} \binom{N}{i} \tag{5}$$

in terms of the VC dimension, and applying the inequality yields the following:

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{d} \binom{N}{i} \leq N^{d_{VC}} + 1 \tag{6}$$

5. Using the VC generalization bound:

$$Pr\left[\forall h \in \mathcal{H} : L(h) \leq L(\hat{h}, S) + \sqrt{\frac{8}{N} ln \frac{4m_{\langle}(2N)}{\delta}}\right] \geq 1 - \delta \tag{7}$$

we can substitute the result from (**??**) into the bound which yields the following bound:

$$Pr\left[\forall h \in \mathcal{H} : L(h) \leq L(\hat{h}, S) + \sqrt{\frac{8}{N} ln \frac{8N^{d_{VC}+4}}{\delta}}\right] \geq 1 - \delta \tag{8}$$

3. VC-dimension

1. It's easy to verify that that the VC-dimension is at least 3 as any triangle created by three points can be shattered by a circle. Collinear points can too be shattered by $\mathcal{H}_+$. It is however not possible to shatter a set of 4 points using $\mathcal{H}_+$ to see this, we consider two cases. 1) When all four point form a quadrilateral convex hull, choosing furthest diagonal points as positive and the closest as negative is a dichotomy which can never be realised. 2) The convex hull is a triangle, thus one of the points is inside the triangle. Classifying the inner point as negative and the outer as positive is a dichotomy which can never be realised. Thus arguing that we have: $m_{\mathcal{H}_+}(k) = 2^k$ for $k = \{1, 2, 3\}, m_{\mathcal{H}_+}(4) < 2^4$,thus we have $d_{VC}(\mathcal{H}_+) = 3$

2. With the new hypothesis set: $\mathcal{H} = \mathcal{H}_+ \cup \mathcal{H}_-$ dichotomies can now be generated from assignment 1, thus we focus on the sample size of 5. Again the reasoning is made regarding convex hulls, the following scenarios are: 1) The convex case, the same arguments follows from question 1 and is therefore not allowed. 2) When 1 point lie within the convex hull, the same argument follows from question 1 and is therefore not allowed. When two points lie within the convex hull, having two setting one to positive and one to negative will the outside too can have interchanging classifications is not possible. Thus arguing that we have: $m_{\mathcal{H}}(k) = 2^k$ for $k = \{1, 2, 3, 4\}, m_{\mathcal{H}}(5) < 2^5$,thus we have $d_{VC}(\mathcal{H}) = 4$