Machine Learning
Assignment 7

Mao Xiqing (tls868)

January 18, 2021

Contents

1 Neural network

The implementation of the neural network with one hidden layer and 20 neurons cna be found in `NN.ipynb`. The neural network uses the following activation function:

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

1.1 Gradient verification

The part of gradient verification can be found at `check(x, y)`. The gradients are calculated by a single backpropagation. The numerically estimated gradients are computed by all data points of `sincTrain25.dt` and the following calculation, with $\epsilon = 10^{-5}$, and the difference threshold is 10^{-8}.

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

They have been confirmed to be very close.

1.2 Training

I trained the data form `sincTrain25.dt` and tested it by using `sincValidate10.dt`, with learning rate at 0.5, 0.1, 0.01, 0.001, iterated 250000 times. The error is measured by mean-squared error:

$$\frac{1}{2n} \sum_{i=1}^{n} \left(Y - Y_{pred}\right)^2, \tag{3}$$

and the norm of gradient is measured by following equation:

$$\|\nabla E(\boldsymbol{w})\| = \sqrt{\sum_i \left(\frac{\partial E(\boldsymbol{w})}{\partial [\boldsymbol{w}]_i}\right)^2}. \tag{4}$$

Their performance are shown on Figure 1.

It is clear that when learning rates are 0.001 and 0.01, the learning speed would be slow (after 250000 times iteration their MSEs are still greater than 10^{-3}). When learning rate is too large, such as 0.5, the lines of performance will be a 'band', meaning that it 'jumped' repeatedly when using gradient descent.

It seems that 0.1 is a decent learning rate, because it learned faster than other rates: MSE decreased shortly after the beginning, and it produced lowest error on both data set. A good way to prevent over-fitting is to apply early stopping. The norm of the error

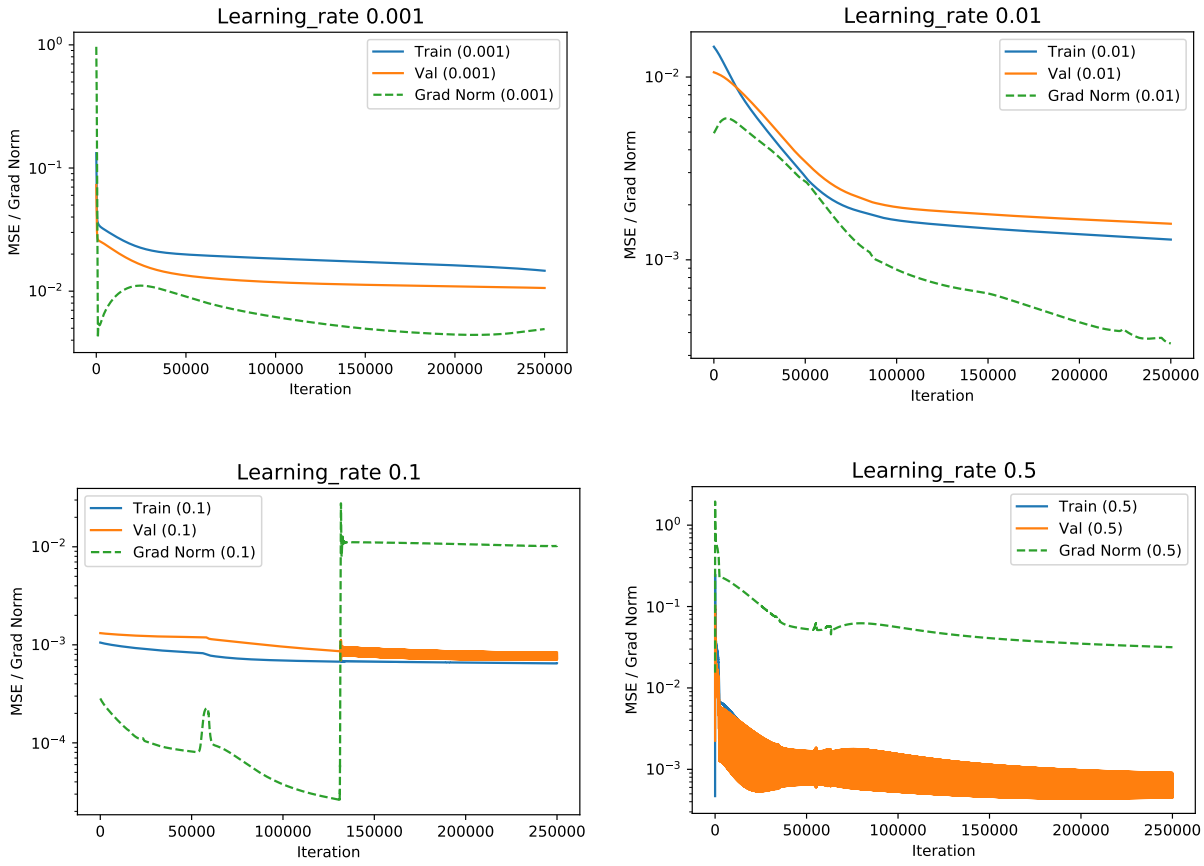

Figure 1: This figure shows the performance of different learning rate.

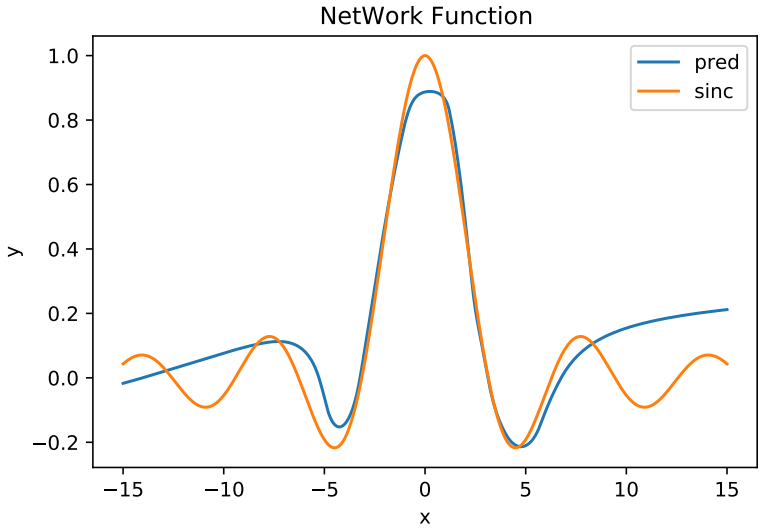

Figure 2: This figure compares Neural Networks function with true sin(x) function with data [-15, -14.95, ... ,15].

gradient suggests that it is better to stop at around 100000 times iteration and we should not do more than 130000 iteration.

Figure 2 shows the output of trained Neural Networks function under learning rate 0.1 and the real sin(x) function over the interval [-15, 15].

2 Early Stopping

1. In case (a), $\hat{L}(h_{t^*}, S_{val})$ is an unbiased estimate of $L(h_{t^*})$. However, in case (b) and (c) $\hat{L}(h_{t^*}, S_{val})$ is an biased estimate of $L(h_{t^*})$. Because in case (a) we do not select best hypnotises, the training procedure always stops after 100 epochs it return $h_{t^*} = h_{100}$ anyway, in case (b) and (c) we select best hypnotises according to some criteria: lowest validation error or no improvement.

2. (1) Bound for case (a):

$$\mathbb{P}\left(L(h_{100}) \leq \hat{L}(h_{100}, S_{val}) + \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}\right) \geq 1 - \delta. \tag{5}$$

(2) Bound for case (b):

$$\mathbb{P}\left(\exists h_{t^*} \in \mathcal{H} : L(h_{t^*}) \leq \hat{L}(h_{t^*}, S_{val}) + \sqrt{\frac{\ln \frac{M}{\delta}}{2n}}\right) \geq 1 - \delta, \tag{6}$$

where h_{t^*} is best hypnotises with the lowest validation error observed during the training process, M is the number of epochs T.

(3) Bound for case (c): We know that $|\mathcal{H}| = T$, where T is the significant number of epoch that no improvement in $\hat{L}(h_t, S_{val})$ is observed. Let $\pi(h) = \sum_{T=1}^{\infty} \frac{1}{T(T+1)} \frac{1}{T}$. Let h_{t^*} be the best hypnotises.

$$\mathbb{P}\left(\exists h_{t^*} \in \mathcal{H} : L(h_{t^*}) \leq \hat{L}(h_{t^*}, S_{val}) + \sqrt{\frac{\ln(T(T+1) \times T)/\delta)}{2n}}\right) \geq 1 - \delta. \tag{7}$$

3. The loss $L(h_{t^*}) - \hat{L}(h_{t^*}, S_{val})$ is bounded by 1, so the square root term can also be bounded by 1:

$$\begin{aligned} \sqrt{\frac{\ln(T(T+1) \times T)/\delta)}{2n}} &\leq 1 \\ \ln(T(T+1)) + \ln(T) - \ln(\delta) &\leq 2n \\ \ln(T) &\leq 2n + \ln(\delta) - \ln(T(T+1)) \end{aligned} \tag{8}$$

4. If we define $\pi(h) = \sum_{T=1}^{\infty} \frac{1}{2^T} \frac{1}{T}$, then

$$\begin{aligned} \sqrt{\frac{\ln(T2^T/\delta)}{2n}} &\leq 1 \\ \ln(T) + \ln(2^T) - \ln(\delta) &\leq 2n \\ \ln(T) &\leq 2n + \ln(\delta) - \ln(2^T). \end{aligned} \tag{9}$$

It is obvious that $\ln(2^T)$ is significantly larger than $\ln(T(T+1))$ when $T > 5$, so under the series $\sum_{T=1}^{\infty} \frac{1}{2^T}$ our model can run significantly less epochs than the series $\sum_{T=1}^{\infty} \frac{1}{T(T+1)}$.

5. ...