

Report for the COSE474 project1

Implementing two-layer neural net

2016320187 컴퓨터학과 홍성윤

1. Code description

(1) 'loss' function (computing scores and loss)

- Since the outputs of the second fully-connected layer are the scores for each class, z_2 should be returned for scores.
- Loss consists of two terms – Log Likelihood loss term and regularization term.
- LL Loss term can be computed by accumulating the log probability of true label.
- Since Regularization term uses L2 regularization and considers only W_1 , W_2 , it can be computed by element-wise power operation.

(2) 'loss' function (computing gradients)

- Since regularization term is derived by W_1 , W_2 , this term can be ignored when computing gradients with respect to the variables other than W_1 , W_2 . So, dL/db_1 , dL/db_2 can be calculated as $quiz_2$ (using chain rule).
- Since dL/db_1 , dL/db_2 should be the average gradients of N samples, get average by using `np.sum`. Likewise, gradients of LL Loss term with respect to W_1 , W_2 respectively also should be the average gradients of N samples.
- But dL/dW_2 , dL/dW_1 gradients must also consider regularization term. In other words, we need dR/dW_2 , dR/dW_1 (let R be the regularization term). dR/dW_2 , dR/dW_1 are equivalent to $W_2 * 2 * reg$, $W_1 * 2 * reg$. Then, add this gradient of regularization term and gradient of LL Loss term.

(3) 'train' function (Implement stochastic gradient descent)

(4) 'predict' function (Use argmax function to get the highest score)

2. Result

- By hyperparameter tuning, I improved the validation accuracy up to 0.38. I used simple loop to check and modify hyperparameters.
- My final hyperparameters: `hidden_size = 50`, `num_iters=1500`, `batch_size=300`, `learning_rate=0.0002`, `learning_rate_decay=0.95`, `reg=0.25`

3. Discussions

- I realized again that hyperparameter tuning process is very important and hard although professor said so.
- 'Learning_rate' parameter seems to have big effect on the model.
- From plot, we can know that tuning process made convergence faster (not decrease linearly) and accuracy bigger
- I used tedious method for tuning, but I think the process can be improved by other method.