Deep Learning Project #1 Report

2016160311 이재윤

Forward Pass

To implement forward pass, 3 variables are set as *z1*, *p1*, and *z2*, which are corresponded to $z_1 = W_1x + b_1,\ p_1 = relu(z_1),\ z_2 = W_2x + b_2$. Since *b1* and *b2* are vectors, the input *x* is calculated *N* times, which is the number of inputs. The output *z2* is set as *scores* to return it if *y* is None.

Loss

p2 is a softmax function which is applied on *z2* and *dl* is a data loss of softmax classifier. *R* is L2 regularization for *W1* and *W2*. *loss* is a total loss of data and regularization.

Backward Pass

Variable *new_y* is a changed form of y which is consisted of 0 and 1 to calculate $p_2 - y$. Each variables dz2, dw2, db2, dp1, dz1, dw1, and db1 are corresponded to $\frac{\partial L}{\partial z_2}, \frac{\partial L}{\partial W_2}, \frac{\partial L}{\partial b_2}, \frac{\partial L}{\partial p_1}, \frac{\partial L}{\partial z_1}, \frac{\partial L}{\partial W_1}$, and $\frac{\partial L}{\partial b_1}$. Since the number of inputs is N, weights are divided by *N* and biases are calculated as average. To implement the derivative of relu function, *dp1*'s negative values are all transformed as 0, and it is copied to *dz1*. Variables *dw2*, *db2*, *dw1*, and *db1* are stored in *grads* dictionary.

Minibatch

Each batches are constructed by randomly chosen values from original data. It considered number of training data and batch size.

Updating Parameters

Variable *parameters* is a dictionary of parameters which contains 'W1', 'W2', 'b1', and 'b2'. By updating rule, *learning_rate* is corresponded to α, and *grads[param]* is corresponded to $\frac{\partial}{\partial w}L(w)$.

Predict

To perform forward pass, the input *X* is put in *loss* function and the return value is stored in *scores*. The argmax function is used to find the index of max value to make *y_pred* which has index of maximum scores.

Result and Discussion

The forward pass and computing loss were successful, but the gradient had some error related to the numerical gradient. Because of this error, the training of network was not also successful. Although the training loss history showed some decreasing trend, the actual loss was not improved. For this reason, it was impossible to get above 36% of test accuracy. The problem is definitely on the formula of gradients in the backpropagation, but it was hard to find the specific point of mistake. In spite of numerous trials, nothing was effective to improve the result.