**Description of Neural\_net.py**

**Forward pass**: variable a1 was implemented by multiplying the input by weight and adding bias. ReLU was used as the activation function of the first fully -connected layer. In order to use softamax loss, the output value from the ReLU function was changed to probability through the softmax function, and then put into the log so that the output value could be compared with the class label. In addition, the weight was adjusted through L2 regularization in the process.

**Backward pass**: Backpropagation was used to calculate loss. After finding the derivative of log and activation function, the gradient was obtained by multiplying the upstream gradient by the local gradient, and the derivative of L2 norm was added to adjust the weight and stored in the dictionary.

**Train**: In addition, to optimize parameters using the SGD method, create random mini batch of training data. and update parameter values through the learning rate, which is the hype-parameter. At this time, the direction of the slope is adjusted to the direction in which the loss decreases.

**Result**

When the hype-parameter was tuned with batch\_size=50, learning rate=0.000600, hidden size=200.0000, 44% of valid accuracy was obtained.

When this was applied to the test set, the test accuracy was 0.441.

스크린샷, 사각형, 다채로움, 직사각형이(가) 표시된 사진

자동 생성된 설명

visualize the weights of the best network

**Discussion :**

1. How much performance will be improved if parameters are updated by Adam method rather than SGD method?
2. How much performance will improve when using He initialization rather than normal distribution as the initial value of the weight