COSE474 Deeplearning Project1 Report

2019320098 김어진

1. description of my code

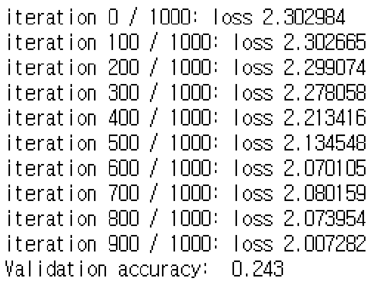
   a) neural_net.py

   In neural_net file, I put z1 as a pre-activation of first layer(z = wx + b), a1 as an activation of first layer, scores as a pre-activation of second layer. I builded a2 value (regularization of W1 and W2 data) and loss value (with softmax). To compute backward pass, I made dscores and hidden values that can be used in grads dictionary. I created X_batch, y_batch and stored random indices with function np.random.choice(). Using stochastic gradient descent, I updated gradients from W2 to W1, from b2 to b1. Finally, I put a1, scores, y_pred to final predict result.

   b) Hyperparameter tuning in two_layer_net.ipynb

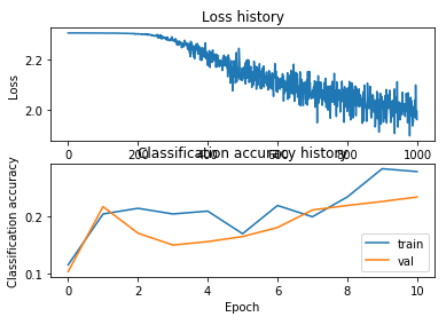
   In hyperparameter tuning part, I tried to find the optimal learning rate and regularization strengths. The learning rates I tested were 1e-2, 1e-3, 1e-4 and the regularization strenghts were 0.3, 0.4, 0.5. I used a for loop to test all cases. I applied each parameter to TwoLayerNet and stored the highest accuracy in the best_val variable.

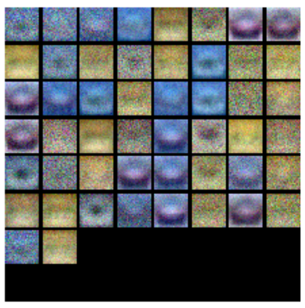
2. results



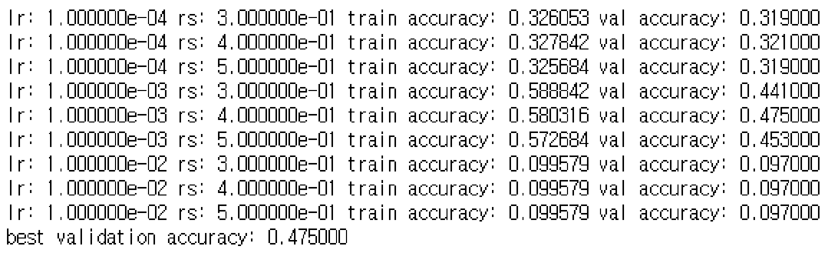
```
iteration 0 / 1000: loss 2.302984
iteration 100 / 1000: loss 2.302665
iteration 200 / 1000: loss 2.299074
iteration 300 / 1000: loss 2.278058
iteration 400 / 1000: loss 2.213416
iteration 500 / 1000: loss 2.134548
iteration 600 / 1000: loss 2.070105
iteration 700 / 1000: loss 2.080159
iteration 800 / 1000: loss 2.073954
iteration 900 / 1000: loss 2.007282
Validation accuracy:  0.243
```

train a network

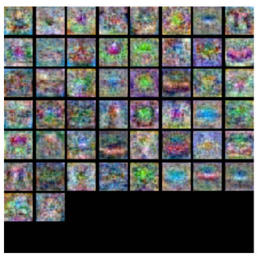




debug the training



```
lr: 1.000000e-04 rs: 3.000000e-01 train accuracy: 0.326053 val accuracy: 0.319000
lr: 1.000000e-04 rs: 4.000000e-01 train accuracy: 0.327842 val accuracy: 0.321000
lr: 1.000000e-04 rs: 5.000000e-01 train accuracy: 0.325684 val accuracy: 0.319000
lr: 1.000000e-03 rs: 3.000000e-01 train accuracy: 0.588842 val accuracy: 0.441000
lr: 1.000000e-03 rs: 4.000000e-01 train accuracy: 0.580316 val accuracy: 0.475000
lr: 1.000000e-03 rs: 5.000000e-01 train accuracy: 0.572684 val accuracy: 0.453000
lr: 1.000000e-02 rs: 3.000000e-01 train accuracy: 0.099579 val accuracy: 0.097000
lr: 1.000000e-02 rs: 4.000000e-01 train accuracy: 0.099579 val accuracy: 0.097000
lr: 1.000000e-02 rs: 5.000000e-01 train accuracy: 0.099579 val accuracy: 0.097000
best validation accuracy: 0.475000
```



tune your hyperparameters

3. Discussion

   My optimal hyperparameters were learning rates 0.001, regularization strengths 0.4, batch size 200, learning rate decay 0.95.

   Best validation accuracy achieved during cross validation was 0.475.

   Final test accuracy was 0.477.