Deep Learning - Homework 3

Brendon Boldt

Nov/27/2017

1 Autoencoder Results

The autoencoder used for the experiment had each layer pretrained for 400 epochs and fine tuned for 50 epochs. We used the dataset our class collected which consisted of 60 training, validation, and test images with a resolution of 256×256 split evenly between 6 classes. Table 1 contains the result of varying the number of neurons in each of the three layers of the autoencoder.

Table 1: Testing accuracy of the standard autoencoder; number of neurons was varied for each data point.

Neurons	Accuracy
$128 \times 64 \times 32$	0.2333
$256 \times 64 \times 32$	0.2500
$256 \times 128 \times 32$	0.2500
$256 \times 128 \times 64$	0.2333
$512 \times 64 \times 32$	0.2833
$512 \times 128 \times 32$	0.2333
$512 \times 128 \times 64$	0.2333
$512 \times 256 \times 32$	0.2000
$512 \times 256 \times 32$	0.2667
$512 \times 256 \times 64$	0.2667

All configurations showed no significant variance in testing accuracy considering the testing dataset only contained 60 images. The best configuration (by a small margin) was $512 \times 64 \times 32$ with an accuracy of 0.2833 which is marginally better than randomly guessing (given that there are 6 classes making for an accuracy of 0.167). Table 2 contains the results of varying the learning rate for the $512 \times 64 \times 32$ configuration.

Table 2: Testing accuracy of the $512 \times 64 \times 32$ varying the learning rate for each data point.

Learning Rate	Accuracy
0.0005	0.3167
0.0001	0.2833
0.00005	0.2167
0.00001	0.2500

The highest learning rate (0.0005) had the highest testing accuracy at 0.3167 which is a marginal increase from the base learning rate (0.0001) with an accuracy of 0.2833. All learning rates greater than or equal to 0.001 resulted in operations on `NaN` suggesting overflowing values, division by zero, or poor neural net architecture.

The inconclusive results are likely due to the structure of the dataset. As a matter of comparison, the CIFAR10 dataset (a standard reference point) contains $60,000$ 32×32 images for 10 categories whereas out dataset contains 180 256×256 images (60 of which were used for training) to account for 6 categories. The resulting dataset is too small and too noisy. Furthermore, given that the training accuracies were often > 0.9 while the validation and testing accuracies remained low suggests extreme overfitting. Thus, given the small sample size and high image resolution, one would not expect a model to generalize off of this our dataset.

2 Autoencoder with RBM Results

The experiment above was repeated using an autoencoder pretrained with RBMs. The RBMs were trained for 400 epochs and the autoencoder was fine tuned for 50 epochs. We used the same dataset as above. The modifications on the RBM Python file were relatively minor since only the dataset had to be changed. Instead of loading the MNIST dataset, we used the `read_data_sets` function to read our dataset and feed that in in the same way as the MNIST data. Additionally, test data was used to measure accuracy at the end of the training epochs.

Table 3: RBM-pretrained autoencoder varying number of neurons.

Neurons	Accuracy
$128 \times 64 \times 32$	0.167
$256 \times 64 \times 32$	0.167
$256 \times 128 \times 32$	0.000
$256 \times 128 \times 64$	0.000
$512 \times 64 \times 32$	0.167
$512 \times 128 \times 32$	0.000
$512 \times 128 \times 64$	0.167
$512 \times 256 \times 32$	0.167
$512 \times 256 \times 64$	0.000

Since accuracies were either 0.0 or $0.167 \approx 1/6$, the model never performed better than randomly guessing or selecting only one category. That being said, we tried varying the learning rates with the $128 \times 64 \times 32$ autoencoder (you know, because as Mies van der Rohe said, “less is more”).

Table 4: RBM-pretrained $128 \times 64 \times 32$ autoencoder varying learning rate.

Learning Rate	Accuracy
0.000001	0.167
0.00001	0.000
0.0001	0.167
0.001	0.167
0.01	0.167
0.1	0.167
1.0	0.167

Varying the learning also showed no sign of improvement with the model. It is likely the case that one or more of the hyperparameters of the model was far from an effective value which cause the model to fail consistently even when varying the number of neurons and the learning rate.