Convolutional Neural Networks for 3D MNIST Image Classification

Jeremy Irvin
Stanford University

Motivation

Visual object recognition is an essential skill for autonomous robots to function in real world environments. Robust classification of 3D digits is a crucial step towards this goal.

Problem Definition

Given a dataset of 10000 rotated and translated 3D point clouds with their digit labels from 0 to 9, automatically assign the correct digit to the 3D image without manual feature engineering.

Background: Voxelization and Conv Nets

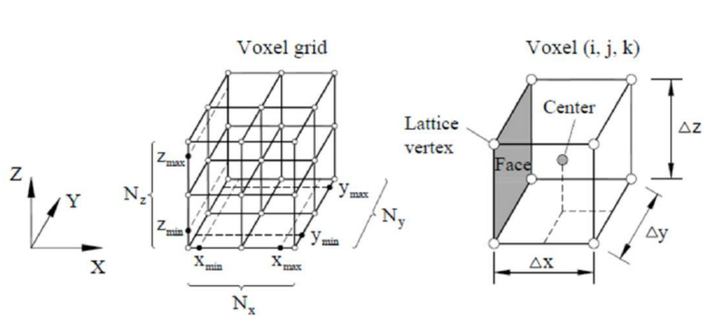

Figure 1 : Example of Occupancy Grid [1]. Each (x, y, z) in the point cloud is assigned a single (i, j, k) voxel. The image is then represented by a $N_x \times N_y \times N_z$ matrix where the ijkth entry contains the number of points assigned to that voxel.

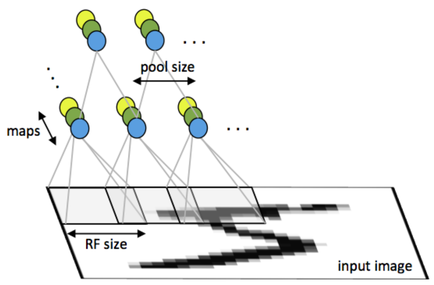

Figure 2 : Example of the 2D convolution and pooling operations [2]. Units of the same color in the convolution layer share the same weights.

Data: 3D MNIST

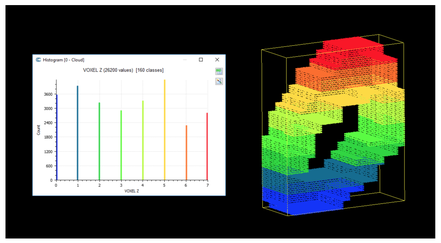

Figure 3 : Example of an image split into 8 voxels along the z-axis (each color corresponds to a single voxel) [1]. The graph shows a count of points in each voxel in the z-dimension.

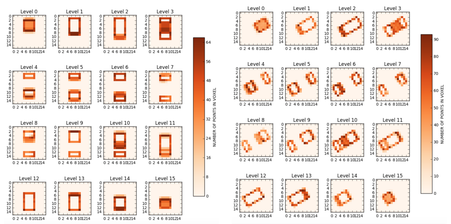

Figure 4 : Example of rotation clockwise by 60° along the z-axis [1]. The left panel shows a projection into the x dimension (image axes are z, y dimensions and levels are the x dimension). The right panel shows the image rotated clockwise by 60° in the z dimension. The colors denote the number of points within a voxel.

Model: 3D Conv Net (VoxNet [3])

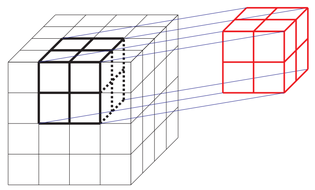

Figure 5 : 3D Filter and Pooling operations on a simple $4 \times 4 \times 4$ image [4]. The $2 \times 2 \times 2$ cube on the right can be thought of as either the filter or the pool. This figure does not show the stride parameter - this is just the size of the shift of the filter or pool (a sliding window).

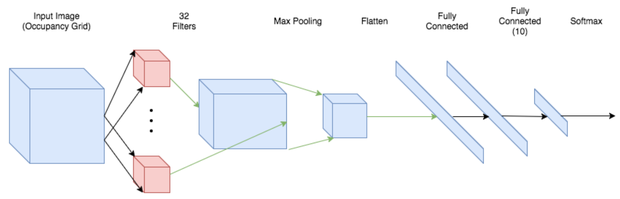

Figure 6 : Basic network architecture. One convolutional layer, max pooling layer, fully connected layer, and last fully connected layer into the 10 classes, where a final softmax is performed. Cross entropy loss is used and trained in Tensorflow with an Adam optimizer.

Experiments and Results

Model	Validation Accuracy
Linear multiclass ovr SVM: L2 Regularization, Squared Hinge Loss	0.5635
Linear multiclass ovr SVM: L1 Regularization, Squared Hinge Loss	0.5645
Linear multiclass ovr SVM: L2 Regularization, Hinge Loss	**0.566**
RBF kernel multiclass ovr SVM	0.5295
Polynomial kernel multiclass ovr SVM	0.107
Sigmoid kernel multiclass ovr SVM	0.496
ovr Logistic Regression:	**0.582**
Multinomial Logistic Regression	0.5805
2 Layer Neural Network, 512 Hidden Dimension	0.667
2 Layer Neural Network, 1024 Hidden Dimension	**0.6785**
Oracle (Vanilla CNN)	0.992

Figure 7 : Summary of baseline results. ovr stands for *one-versus-rest* (10 binary classifiers). In the SVM case, ovr is contrasted with a *one-versus-one* scheme ($\binom{10}{2} = 45$ binary classifiers). In the Logistic Regression case, ovr is contrasted with Multinomial Logistic Regression, which is softmax regression with cross-entropy loss. The best SVM, logistic regression, and neural network classifiers are bolded. The oracle is a vanilla convolutional neural network which uses the 2D image that was used to generate the 3D point clouds.

Model	Validation Accuracy
3DConvNet: 1 layer, 8 filters, 1024 hidden dim	**0.69**

Figure 8 : Summary of results.

Future Work

- Further experiments using more complex architectures
- Visualize cross sections of filters and activation of fully connected layers

References

[1] https://www.kaggle.com/daavoo/d/daavoo/3d-mnist/
[2] http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/
[3] http://www.dimatura.net/extra/voxnet_maturana_scherer_iros15.pdf
[4] https://ai2-s2-pdfs.s3.amazonaws.com/3c86/dfdbdf37060d5adcff6c4d7d453ea5a8b08f.pdf