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 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

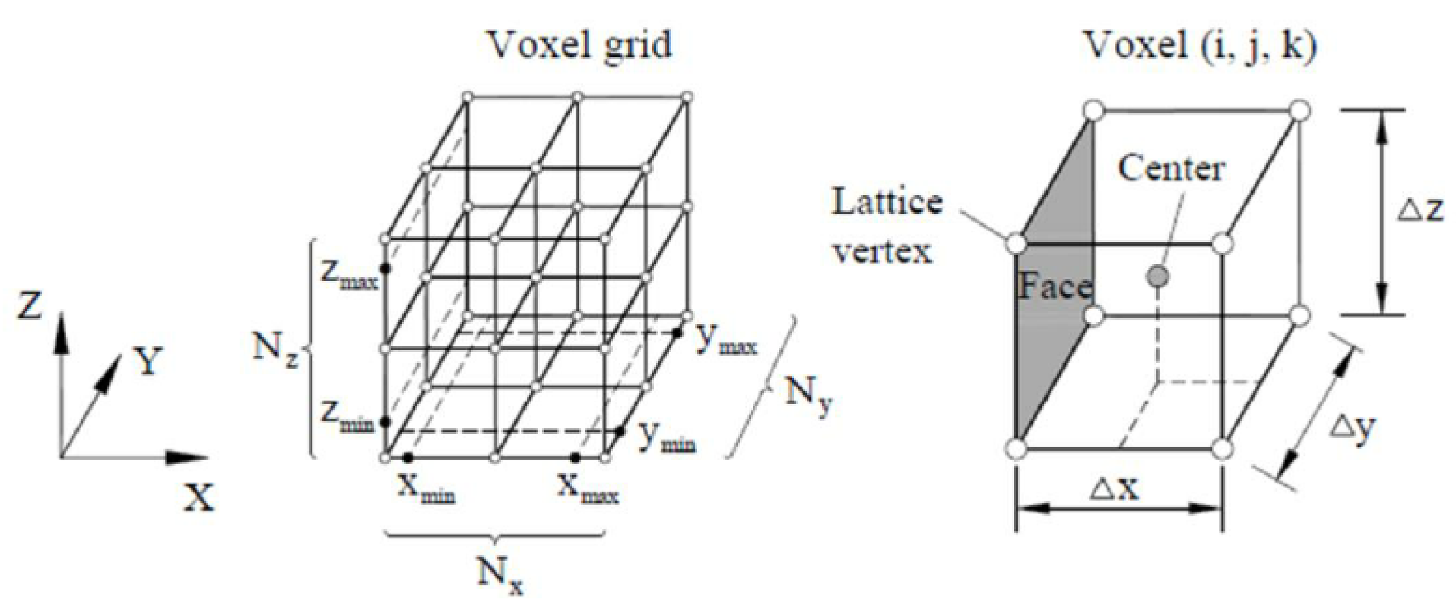

Figure 1 : Example of Occupancy Grid [1]. Each (x, y, z) in the point cloud is assigned a single (i, j, k) voxel. The image is then represented by a $N_x \times N_y \times N_z$ matrix where the ijkth entry contains the number of points assigned to that voxel.

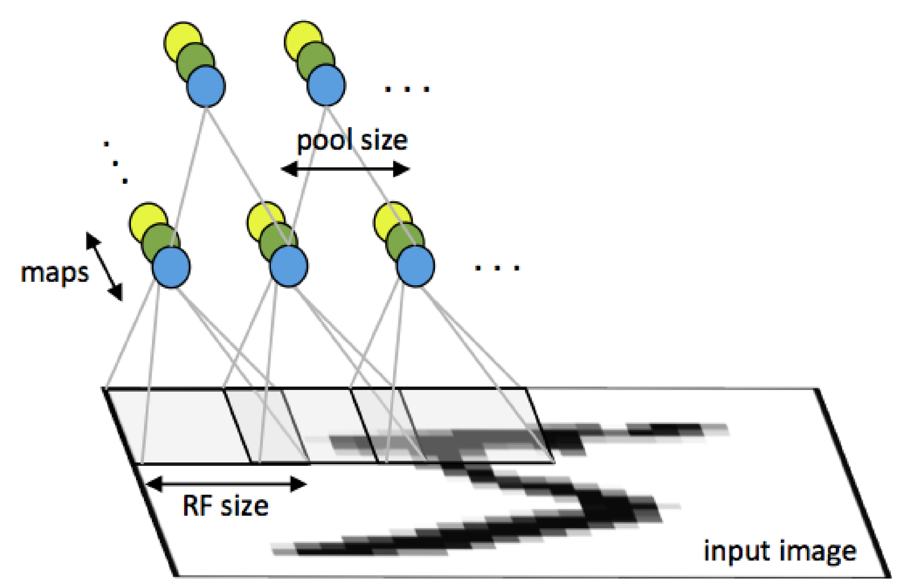

Figure 2 : Example of the 2D convolution and pooling operations [2]. Units of the same color in the convolution layer share the same weights.

Data: 3D MNIST

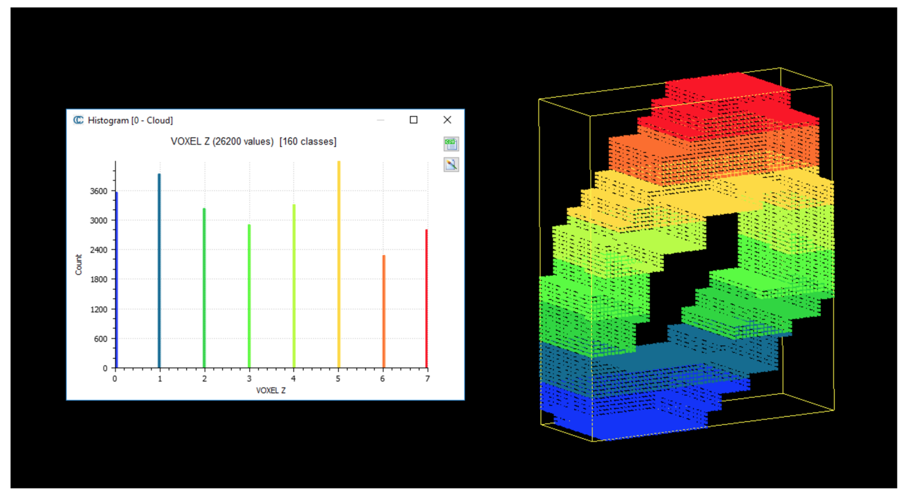

Figure 3 : Example of an image split into 8 voxels along the z-axis (each color corresponds to a single voxel) [1]. The graph shows a count of points in each voxel in the z-dimension.

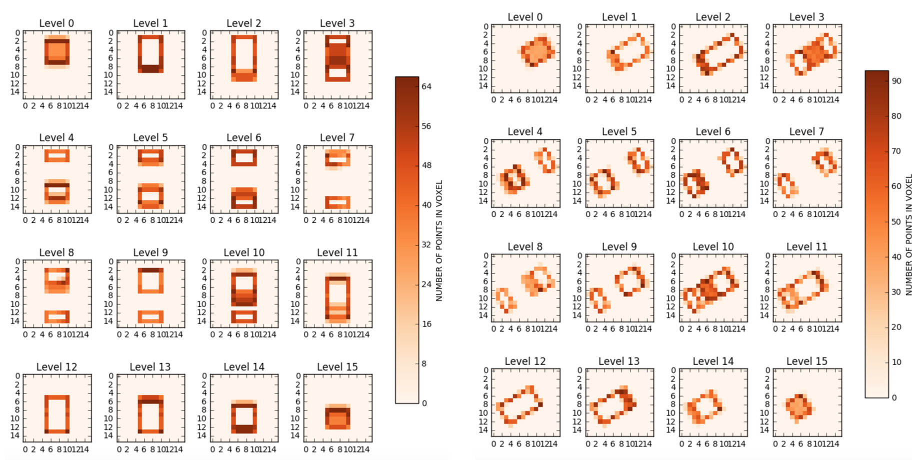

Figure 4 : Example of rotation clockwise by 60° along the z-axis [1]. The left panel shows a projection into the x dimension (image axes are z, y dimensions and levels are the x dimension). The right panel shows the image rotated clockwise by 60° in the z dimension. The colors denote the number of points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

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

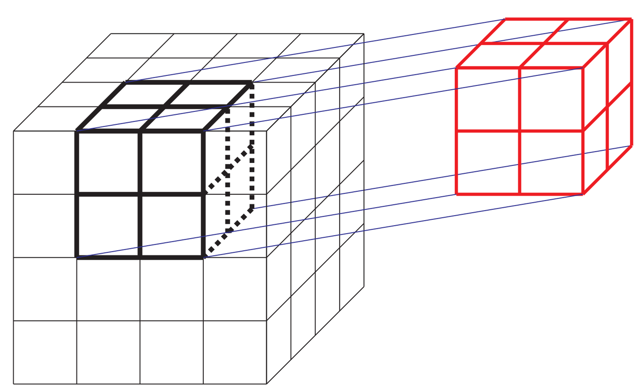

Figure 5 : 3D Filter and Pooling operations on a simple $4 \times 4 \times 4$ image [4]. The $2 \times 2 \times 2$ cube on the right can be thought of as either the filter or the pool. This figure does not show the stride parameter - this is just the size of the shift of the filter or pool (a sliding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

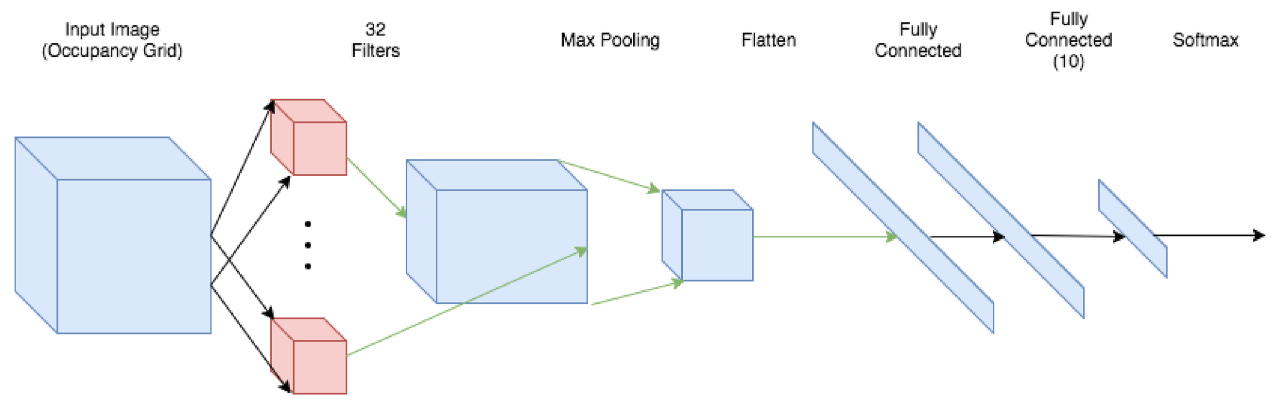

Figure 6 : Basic network architecture. One convolutional layer, max pooling layer, fully connected layer, and last fully connected layer into the 10 classes, where a final softmax is performed. Cross entropy loss is used and trained in Tensorflow with an Adam optimizer.

Experiments and Results

Model	Test Accuracy
Linear multiclass ovr SVM: L2 Regularization, Squared Hinge Loss	0.554
Linear multiclass ovr SVM: L1 Regularization, Squared Hinge Loss	**0.566**
RBF kernel multiclass ovr SVM	0.542
Polynomial kernel multiclass ovr SVM	0.126
Sigmoid kernel multiclass ovr SVM	0.51
ovr Logistic Regression:	**0.5905**
Multinomial Logistic Regression	0.583
2 Layer Neural Network, 128 Hidden Dimension, Sigmoid Nonlinearity	0.622
2 Layer Neural Network, 256 Hidden Dimension, Sigmoid Nonlinearity	**0.634**
2 Layer Neural Network, 512 Hidden Dimension, Sigmoid Nonlinearity	0.6315
2 Layer Neural Network, 1024 Hidden Dimension, Sigmoid Nonlinearity	0.6285
Oracle (Vanilla CNN)	0.992

Figure 7 : Summary of baseline results on test set (2000 examples). ovr stands for *one-versus-rest* in contrast with a *one-versus-one* scheme. 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 (Sigmoid)	Test Accuracy
1 layer, 32 filters, 128 hidden dim	0.6655
1 layer, 32 filters, 256 hidden dim	0.6915
1 layer, 32 filters, 512 hidden dim	0.6475
2 layer, 32 filters, 128 hidden dim	0.7125
2 layer, 32 filters, 256 hidden dim	0.7275
2 layer, 32 filters, 512 hidden dim	**0.728**
2 layer, 32 filters, 1024 hidden dim	0.7225

Figure 8 : Summary of results of different 3D Convolutional Neural Network architectures.

Model (2 layer, 32 filters, 256 hidden dim)	Test Accuracy
ReLU	0.7055
Sigmoid	**0.7275**
Tanh	0.719
ELU [5]	0.7105
None	0.689

Figure 9 : Comparison of nonlinearities between fully connected layers.

Future Work

- Further experiments using different architectures and more/less fine-grained voxelization
- 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] D. Maturana and S. Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. *IROS*, 2015.
[4] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *PAMI*, 2013.
[5] Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). *ICLR*, 2016.