SVHN Digit Recognition

Deep Learning

Prepared by Daniel Holloman

Contents / Agenda

- Business Problem Overview and Solution Approach
- Data Preprocessing for ANNs
- Model Performance Summary for ANNs
- Data Preprocessing for CNNs
- Model Performance Summary for CNNs
- Conclusion

Business Problem Overview

Google's Street View mapping capabilities are extensive. With more than 220 billion images (Google, 2022), Google can see many of the world's house numbers in 360 degree images. Street View collects GPS data of every image, and makes a record of the coordinates. Along with this wealth of data comes a massive labeling problem. Performing manual data entry and human supervised learning on such a large dataset would be far too slow, laborious, and expensive. As the size of the database increases, so too does the need for structured and labeled housing numbers. This task cannot realistically be done with human assistance alone. However, labeling automation using deep learning and computer vision can help categorize digits in a huge dataset.

Deep learning can help alleviate the labeling problem by using image processing. Massive amounts of housing numbers can be broken into single digits and labeled using object recognition. Google combines self-driving cars, cameras, deep learning, and neural networks in ensemble to analyze and classify cropped digits from all their Street View data, and labels them with the proper street address.

Deep learning models of the SVHN can handle large datasets faster than human supervised learning. This method provides labels of house digits with an acceptable level of accuracy, effectively cleaning the dataset. This report focuses on testing and examining the accuracy, results, and methodology used by Google. It offers an analysis of the Street View Housing Number Database (SVHN) using fully-connected and convolutional neural networks. The analysis highlights the key findings of testing these neural networks with various layer types, attempting to find the model with the highest correlation between training and testing data.

Data Overview

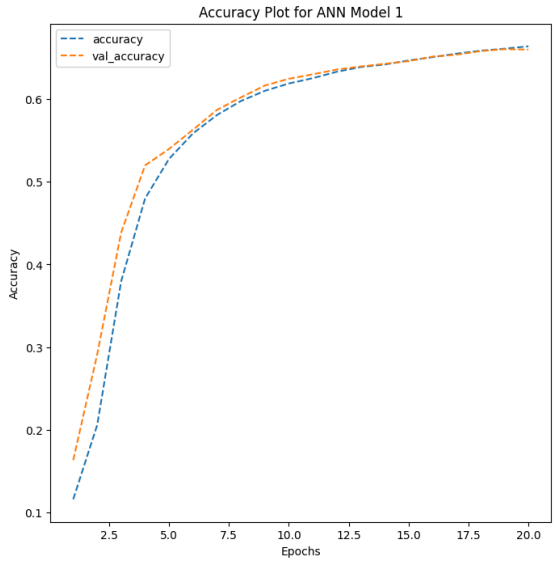
The SVHN database contains more than 600,000 cropped digits from house and business numbers. It can effectively capture these digits and compile them together to improve the accuracy and efficiency of Google Maps for end users. The database has captured digits from “more than 10 million miles” of roads across the world (Google, 2022).

Data Pre-Processing for ANNs

First, the data must be cleaned to make it usable by the neural networks. For an ANN, data is split into training, testing, and validation sets. It must be scaled and flattened into a 1-dimensional array, encoded in categories, and the features engineered by creating several dense layers. ANN Model 2 also includes a dropout layer and batch normalization layer. Data will be shuffled, batch normalized, and compiled.

In preprocessing the labels, the data must be classified with one-hot encoding. Data normalizing creates upper and lower limits for inputs, speeds up gradient descent, and reduces compute and costs. Then the shape is checked to see if it is an appropriately flattened vector. The model can then perform the gradient descent and calculate the accuracy over epochs.

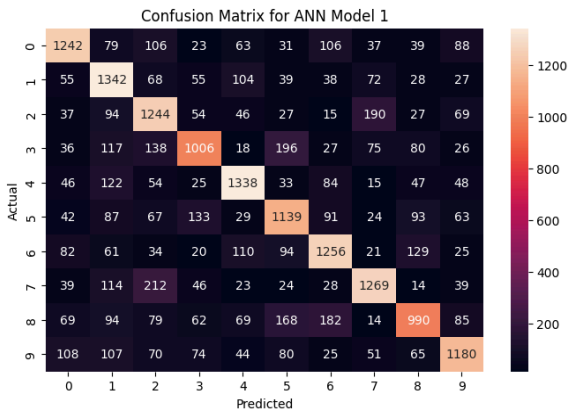
Model Performance Summary: ANN Model 1



ANN Model 1 has been identified as the most useful ANN tested. The main difference between the ANN models was that ANN Model 1 utilizes a series of dense layers, while ANN Model 2 used dense layers, but also mixed in a dropout layer, and a batch normalization layer. The validation accuracy of the first fully-connected neural network fits the training accuracy curve extremely well, and even overlaps at times. This is a sign that the testing accuracy is successfully predicting digits and ignores much of the noise in the dataset. The testing and validation accuracy rises steeply at first and begins to flatten out after about the 8th epoch mark, which corresponds with approximately 60% accuracy. It is better to have a well-fit model than to have a high rate of accuracy. Batch normalization and dropout were not used, but still, the validation set learned very quickly at the beginning and tapered off in later epochs, with both curves appearing very similar.

Classification Report for ANN Model 1:

	precision	recall	f1-score	support
0	0.71	0.68	0.70	1814
1	0.61	0.73	0.66	1828
2	0.60	0.69	0.64	1803
3	0.67	0.59	0.63	1719
4	0.73	0.74	0.73	1812
5	0.62	0.64	0.63	1768
6	0.68	0.69	0.68	1832
7	0.72	0.70	0.71	1808
8	0.65	0.55	0.60	1812
9	0.72	0.65	0.68	1804
accuracy			0.67	18000
macro avg	0.67	0.67	0.67	18000
weighted avg	0.67	0.67	0.67	18000



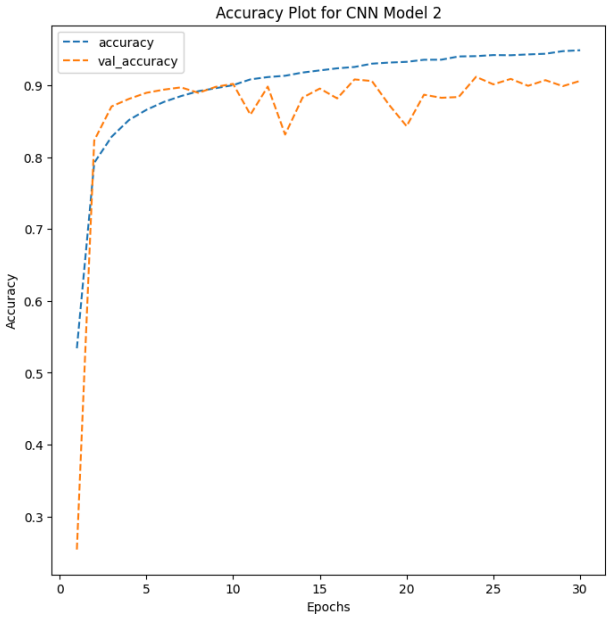
Harmonic mean averages about 67%, as all F1 scores remain between 0.60 and 0.73. The precision score closely follows the same pattern, showing the same average, 67%.

Data Pre-Processing for CNNs

In order to preprocess the training data, it must be printed and the array must be shaped. Testing and training data must be reshaped into a 4D array. Data needs to be normalized by dividing by 255. The new shapes of the datasets can be viewed. Finally, one-hot encoding helps simplify the data.

To preprocess the labels, the CNN architectures must be modeled with a mixture of Leaky ReLUs, convolutional layers, max-pooling, and flattening.

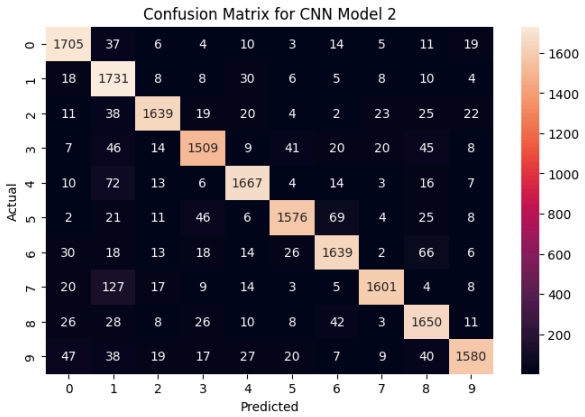
Model Performance Summary: CNN Model 2



CNN Model 2 is the most effective of the two CNN models observed. There are more Leaky ReLUs and convolutional layers in CNN 2, when compared to CNN 1. CNN 2 also utilizes a dropout layer. In this model, training accuracy starts very high (around 53%) and both accuracy curves show a sharply upward trend, which starts to flatten out around the 90% accuracy level, after only about 5 epochs. The validation accuracy seems to be overfitting to noise in the data. The learning rate is suspiciously high at first, then flattens out. The training and validation data starts off pretty close, but in later epochs the validation curve stays consistently below the accuracy curve, bouncing around from noise.

Classification Report for CNN Model 2:

	precision	recall	f1-score	support
0	0.91	0.94	0.92	1814
1	0.80	0.95	0.87	1828
2	0.94	0.91	0.92	1803
3	0.91	0.88	0.89	1719
4	0.92	0.92	0.92	1812
5	0.93	0.89	0.91	1768
6	0.90	0.89	0.90	1832
7	0.95	0.89	0.92	1808
8	0.87	0.91	0.89	1812
9	0.94	0.88	0.91	1804
accuracy			0.91	18000
macro avg	0.91	0.91	0.91	18000
weighted avg	0.91	0.91	0.91	18000



Precision, recall, and the F1 score is high in CNN Model 2, with an average of 91% in all three categories. This may look like a good model, but as mentioned previously, the validation accuracy does not fit well to the training accuracy.

Choosing the Final Model:

Artificial neural networks are typically better for tabular data, but ANN Model 1 is the best option for this use case. The training and validation accuracy lines nearly overlap, signaling consistent fitting. Although the accuracy curves begin to flatten around 60%, and peak around 68%, they are products of the model with the best fit and highest consistency. This model will be the most effective across future epochs. Sometimes the simplest solution is the best.

CNN Model 2 is naturally more apt to handle image data, but overfits on a lot of noise, and displays too much variation to be considered effective in this case. After about 4 epochs it begins to flatten out and exhibits a zig-zag trajectory. It does not fit well to the training data. It does not generalize well. This could lead to large accuracy errors if the size of the dataset increases, and Google Street View is a massive dataset.

There is an art and a science to using deep learning to identify digits from images. The analysis in this report may work well in small use cases, but results can vary depending on the size of the dataset. Results can also vary greatly depending on the chosen architecture of each neural network.

Conclusion:

By analyzing and employing several different neural networks, Google's labeling problem can largely be solved through automation. Neural networks prove to be an effective tool, not only in summarizing data, but also in categorizing and labeling the dataset.

The analysis proposes that the simplest solution is best. Although CNNs are typically better at processing and identifying data from images, in this case the simpler ANN models outperformed the rest.

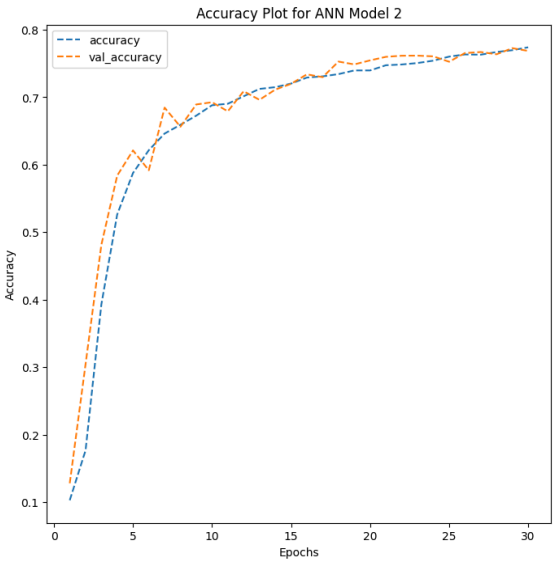
Data science can be considered an artform, and it should be noted that the methods employed can vary and results of the analysis can vary based on randomization techniques which are inherent in the models. Through careful analysis and rigorous testing cases, we have identified ANN Model 1 as the best solution to Google Street View's labeling problem.

Appendix:

Key differences in preparing ANNs and CNNs

	Artificial Neural Network	**Convolutional Neural Network**
Input Shape	Flat vector	Multi-dimensional
Spatial Info	Destroyed unless engineered	Preserved
Feature Engineering	Often required	Minimal
Typical Tasks	Tabular data, generic tasks	Images, audio, time series
Preprocessing Effort	Typically higher	Low to Moderate

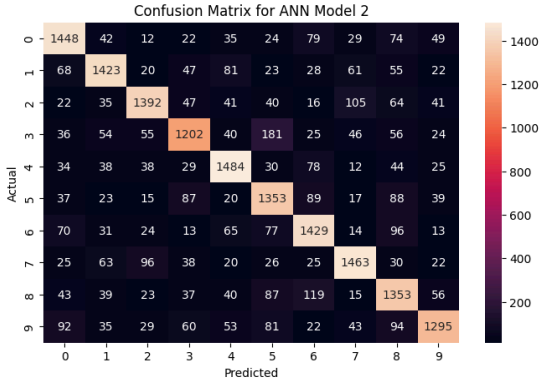
Supplemental Analysis: ANN Model 2



The second fully-connected model shows a validation accuracy line that zig-zags around the testing accuracy line. This could be caused by overfitting the testing accuracy to noise. Dropout rate is set to 20%, and batch normalization is employed to speed up and stabilize training.

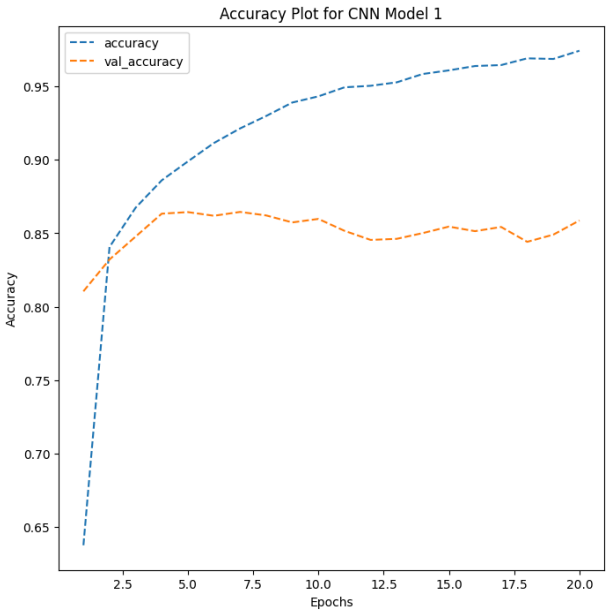
Classification Report for ANN Model 2:

	precision	recall	f1-score	support
0	0.77	0.80	0.79	1814
1	0.80	0.78	0.79	1828
2	0.82	0.77	0.79	1803
3	0.76	0.70	0.73	1719
4	0.79	0.82	0.80	1812
5	0.70	0.77	0.73	1768
6	0.75	0.78	0.76	1832
7	0.81	0.81	0.81	1808
8	0.69	0.75	0.72	1812
9	0.82	0.72	0.76	1804
accuracy			0.77	18000
macro avg	0.77	0.77	0.77	18000
weighted avg	0.77	0.77	0.77	18000



F1 stays flat, with an average of 0.77. Precision varies with each digit, with a possible central limit also near 0.77. Many of the predictions were correct. But there still remains noise affecting the analysis.

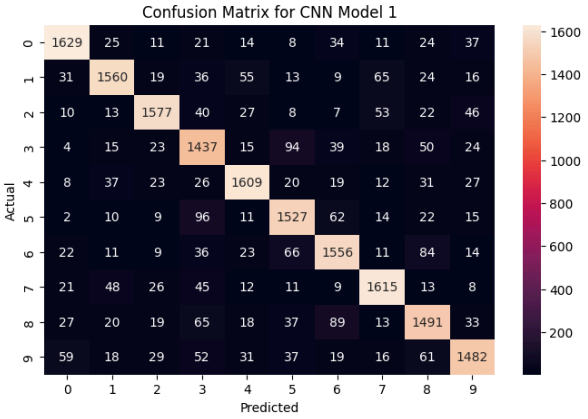
Supplemental Analysis: CNN Model 1



Validation accuracy starts off very high, around 81%. As the epochs increase, the validation line plateaus and then begins to decrease slightly. This is likely due to overfitting. The model may be simply memorizing the noise, creating this type of relationship, with very little correlation between training and testing accuracy.

Classification Report for CNN Model 1:

	precision	recall	f1-score	support
0	0.90	0.90	0.90	1814
1	0.89	0.85	0.87	1828
2	0.90	0.87	0.89	1803
3	0.78	0.84	0.80	1719
4	0.89	0.89	0.89	1812
5	0.84	0.86	0.85	1768
6	0.84	0.85	0.85	1832
7	0.88	0.89	0.89	1808
8	0.82	0.82	0.82	1812
9	0.87	0.82	0.85	1804
accuracy			0.86	18000
macro avg	0.86	0.86	0.86	18000
weighted avg	0.86	0.86	0.86	18000



The harmonic mean stays relatively stable over all 10 digits (0-9). Precision stays somewhat stable as well. The confusion matrix shows a lot of correct predictions. Overall, it seems to be predicting well, but the correlation between training and testing sets is very low.

Citations

Ahmed, A. (2018, February 22). *Getting started with "street view house numbers" (SVHN) dataset*. Cerenaut. https://cerenaut.ai/2018/01/31/getting-started-street-view-house-numbers-svhn-dataset/

Google. (2022). *Celebrate 15 years of exploring your world on Street View*. Google. https://www.google.com/streetview/anniversary/#:~:text=Street%20View%20began%20in%202007,100%20countries%20and%20territories%20together.

Ibarz, J., & Banerjee, S. (2017, May 3). *Updating google maps with deep learning and Street View*. Google Research. https://research.google/blog/updating-google-maps-with-deep-learning-and-street-view/

Li, J. (2018, May 11). *Street View House Numbers (SVHN)*. Kaggle. https://www.kaggle.com/datasets/stanfordu/street-view-house-numbers

Stanford University. (n.d.). The Street View House Numbers (SVHN) Dataset. Stanford University. http://ufldl.stanford.edu/housenumbers/