Street View Housing Number Digit Recognition

Deep Learning

Reported by

Phuong Tran

September 28, 2023

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

Business Problem:

- This analysis aims to improve the map quality by automatically transcribing the address numbers from pixels in street-level photos into text. Each of the buildings on the Google map will be attached to a corresponding street number label. The aim of this report is to evaluate which model has the best performance in terms of predicting the address numbers given labeled data as inputs.

Business Objective:

- The business objective of this analysis is to predict the number depicted inside the street-level photos by using neural network models with the best performance.

Approach (Methodology):

- I will be using the two methods Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN), and select the one with the best performance.
- The best performance is determined by the accuracy level of the models measured in precision, recall, and F1 tests in different data subsets.
- The results will be displayed in a graphical comparison of overall accuracy of the models, and in more details in tabular view and confusion matrix.

Data Overview

- This Street View Housing Number (SVHN) dataset contains over 600,000 labeled digits cropped from the street-level photos in 2D pixel form. The dimension of each image is 32 x 32 pixels. The pixel data is stored under .h5 file, and 10% of the dataset will be used in this analysis.
- There are total of 60,000 images split into two parts:
    - 42,000 images in the training dataset
    - 18,000 images in the testing dataset
- The dataset comes with labeled data, so there will be further divisions in features and pre-existing labeled outputs. The independent variables called features are stored under x and the dependent variables called target values are stored under y (predicted values).
- The methodology used in this analysis include:
    - Artificial or Fully Connected Feed Forward Neural Networks (ANN)
    - Convolutional Neural Networks (CNN)

Data Pre-Processing for ANNs

Before feeding the data in to the neural networks' models, I need to manipulate and transform the pixel data into the right frameworks in the below steps. These data transformation pre-processing steps are necessary to make ANNs runnable, and ensure the data input fit and in the right framework.

- First, I need to reshape the 2D image data into 1D array. The data was given in a 2D format, this process is called Flatten.

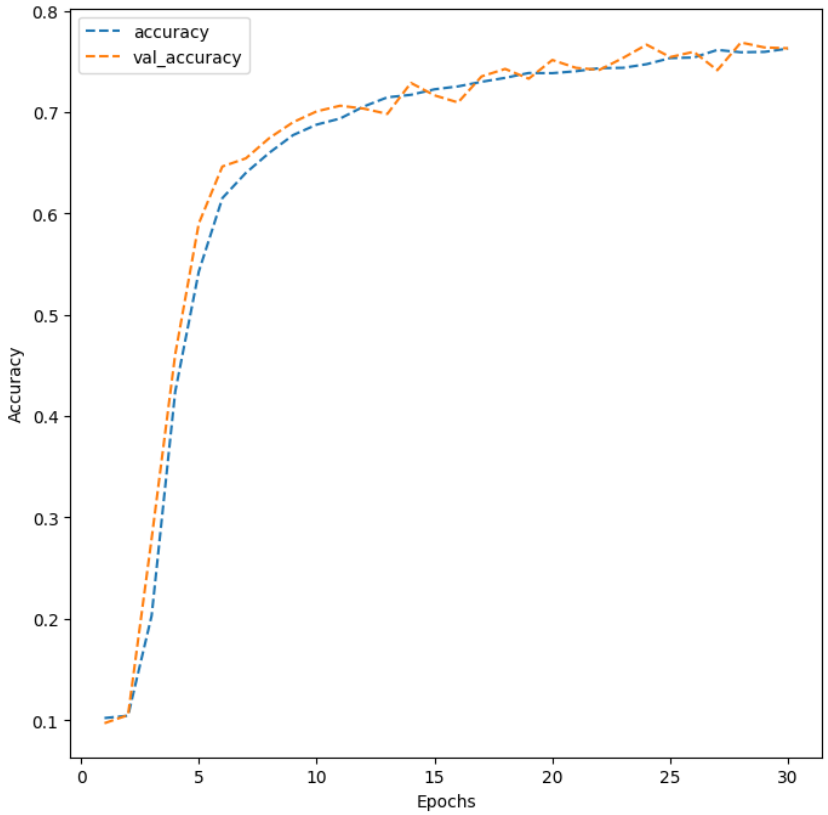
- Second, the data needs to be normalized from a 0 to 255 range of pixel units to a 0 to 1 scale.

- Third, the last step is to one-hot encode. One-hot encoding means converting the vectors into binary class matrix of 0 and 1s to form the training target vector.

ANN Model Performance Summary

Artificial Neural Networks:

- Two models of ANN were built in this analysis. The main differences are:
  - Model 1 has 20 epochs, and 3 hidden layers.
  - Model 2 has 30 epochs, and 8 hidden layers.
- The training versus validation accuracy plot for the ANN model 2 which is more complex and more accurate. The overall accuracy of the model is 76%, which is 10% points increase from model 1 when added more layers and more epochs. The consistency of the ANN model 2 is great. The results in the validation dataset is aligned with the results in the training dataset.

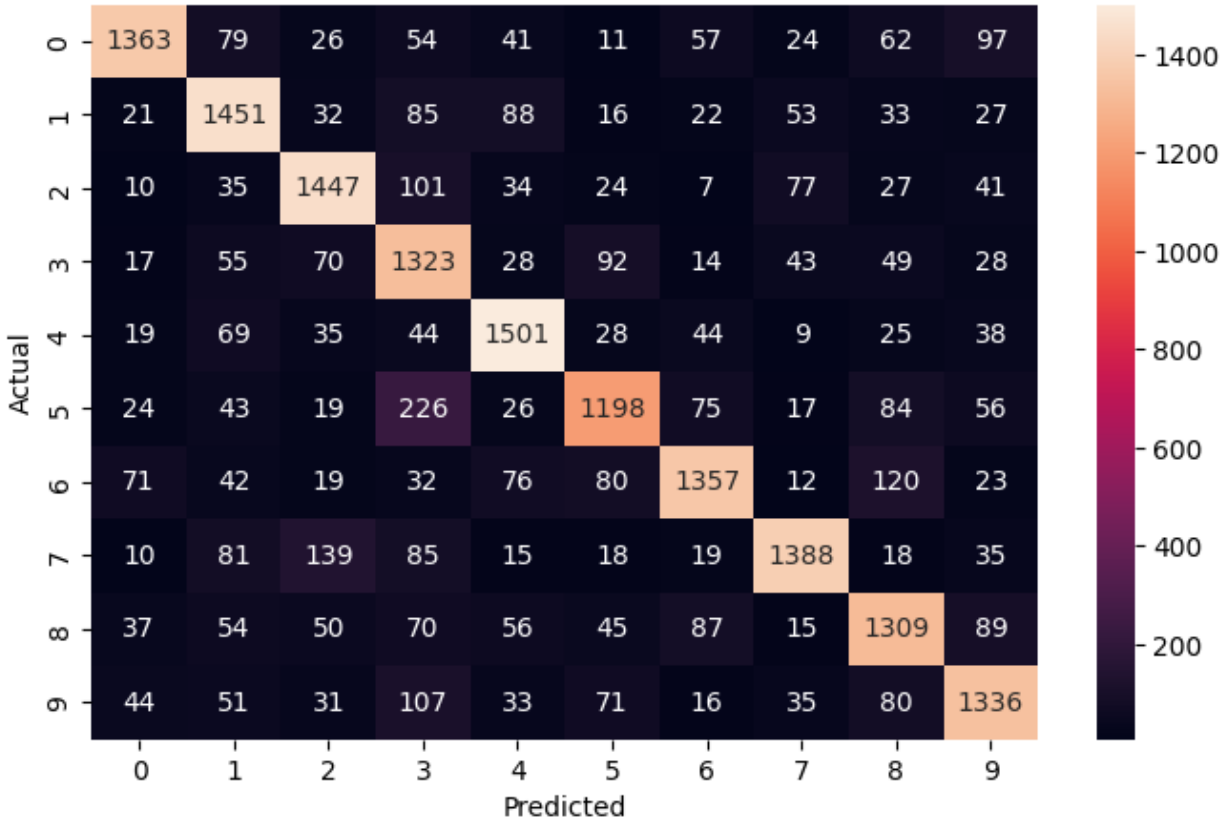


- Below are a detailed table of the accuracy metrics in precision rate and F1 score break down by each numerical digit of the target labels from 0 to 9.

digit	precision	recall	f1-score	support
0	0.8434	0.7514	0.7948	1814
1	0.7403	0.7938	0.7661	1828

2	0.7746	0.8026	0.7883	1803
3	0.6220	0.7696	0.6880	1719
4	0.7908	0.8284	0.8092	1812
5	0.7568	0.6776	0.7150	1768
6	0.7992	0.7407	0.7688	1832
7	0.8296	0.7677	0.7975	1808
8	0.7244	0.7224	0.7234	1812
9	0.7548	0.7406	0.7476	1804

- The confusion matrix reflects the high number of accurate predictions provided by the ANN model, especially for digit number 4 that yields the highest accuracy rate of 81% reflected by the lighted color on the scale. The remaining predicting errors are reflected on the black squares that are the darkest color in the color scale.



- Overall observations:
  - Accuracy rate is moderate at the upper 70%, with only one digit around 80%. There is still room for improvement.

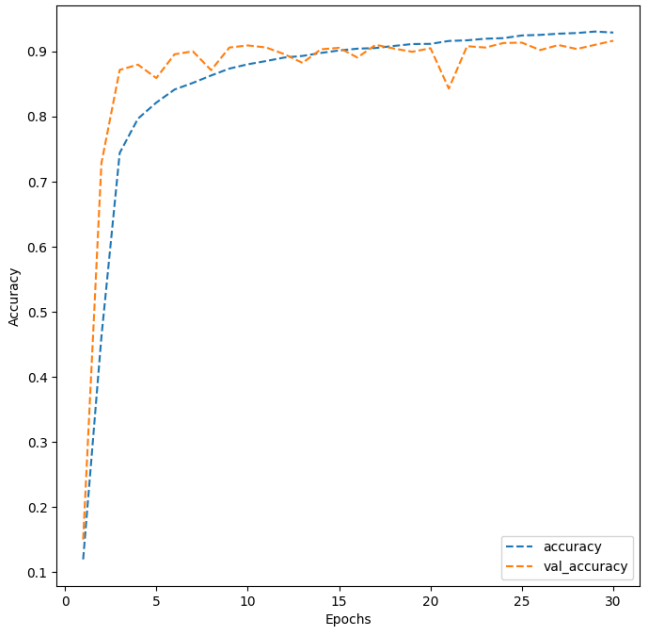
Data Pre-Processing for CNNs

- Before training the data, I had to perform the below preprocessing steps required for making the training data usable for CNNs.
    - Load and explore the dataset
    - Explore the shape of the dataset and reshape the 4D arrays as input to CNN
    - Normalize the inputs that were stored as 256 pixel values into value range from 0 to 1.
- The next data preprocessing steps required for the labels is called one-hot encoding. A function is created to one-hot encode labels in target variable y_train and y_test as in 0s-and-1s matrix.

CNN Model Performance Summary

Convolutional Neural Networks:

- The main differences between the two models built are:
    - The first model (model 1) has 20 epochs and 9 hidden layers
    - The second model (model 2) has 30 epochs and 16 hidden layers.
- After reviewing the results of the two models, I chose the CNN model 2 that ran on 30 epochs and 16 hidden layers. Even though the overall accuracy rates of the two models are similar at above 90%, the second model performed better on the validation dataset, as shown in the graph below where the two lines emerge. This result demonstrates higher consistency of accuracy in the CNN model 2 when added more complexity.
- Below is the training versus validation accuracy plot for the second model

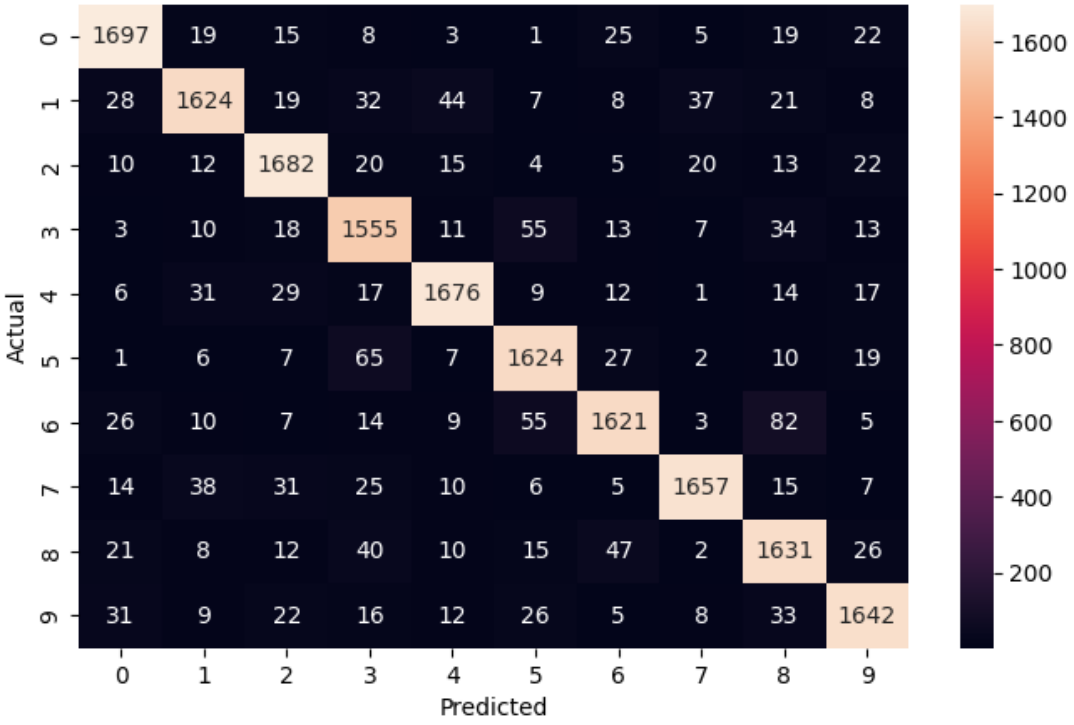


- The accuracy rate of the chosen model is reflected under precision and F1 score for each digit. Most numerical digits from 0 to 1 are predicted correctly; most f1-scores are above 90%, except for digit number 3 and 8. The two numbers look very similar compared to the rest. Specifically,

number 3 is visually more than one half of number 8 when slicing number 8 in half vertically.

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

- The number of accurate predictions that match with the actual value are highlighted in the color scale. Lightest color means the highest level of accurate predictions, while the black-colored are errors in predictions.



- Overall observations
  - The CNN model 2 is a better choice compared to the CNN model 1. The second model has better performance in terms of accuracy on training and validation datasets, around 91% for both.

Conclusion: Choosing the Final Model

- After carefully running the analysis on two different approaches, I would recommend CNN model 2 with 16 hidden layers and 30 epochs are the final solution. The model provides highest accuracy (91%) in training dataset among all 4 models, and the accuracy level stays consistent when tested on the validation dataset (90-91%) during 30 epochs.

- Side note, even though the CNN model 2 yield the highest accuracy rate among the four, it takes the longest time and the most power to compute. The total running time for the CNN model 2 is 75 minutes, while ANN models take <2 minutes, and CNN model 11 takes 33 minutes. The final model, CNN model with more complexity and more iterations, is designed for projects that is important, but not urgent. If timing and computing power is not a concern, then the more complex CNN model is the best.

Appendix

Steps of building ANN models

Step 1	Load the dataset
Step 2	Visualizing images
Step 3	Data preparation
Step 4	Normalize the train and the test data.
Step 5	Model Building
Step 6	Model Architecture
Step 7	Build and train an ANN model as per the above-mentioned architecture.
Step 8	Plotting the validation and training accuracies
Step 9	Second Model Architecture
Step 10	Build and train the new ANN model as per the above-mentioned architecture.
Step 11	Plotting the validation and training accuracies
Step 12	Predictions on the test data
Step 13	Print the classification report and the confusion matrix for the test predictions. Write your observations on the final results.

Steps of building CNN models

Step 1	Using Convolutional Neural Networks
Step 2	Data preparation
Step 3	One-hot encode the labels in the target variable y_train and y_test.
Step 4	Model Building
Step 5	Model Architecture
Step 6	Build and train a CNN model as per the above-mentioned architecture.
Step 7	Plotting the validation and training accuracies
Step 8	Second Model Architecture
Step 9	Build and train the second CNN model as per the above-mentioned architecture.
Step 10	Plotting the validation and training accuracies
Step 11	Predictions on the test data
Step 12	Make predictions on the test data using the second model.
Step 13	Write your final observations on the performance of the model on the test data.