Implementing Neural Networks in TensorFlow for the Task of Character Recognition

How Approaches Differ, and What Inferences Can Be Drawn Regarding More Complex Problems

Yannik Glaser
University of North Georgia

ABSTRACT

This project aims to analyze and present the discrepancies in performance of different implementations of neural networks. A basic feed-forward neural networks, a feed-forward neural network with convolutional layers and lastly a recurrent convolutional neural network will be the subjects of comparison, being used for the in the task of character recognition. Performance will be measured in terms of maximum accuracy achieved for the MNIST character dataset and training speed. To implement these neural networks, Python and TensorFlow will be utilized. The collected data will be used as a framework to make predictions regarding solutions for more elaborate deep learning utilizations, for instance object recognition. A conclusion about the potential held by different implementations for presenting viable solutions to problems the deep learning research community is currently concerned with will be presented at the end.

The MNIST Character Set

- Handwritten digits 28x28 pixels
- Utilized for its ease of use due to the dataset's uniformity in size and positioning of the characters
- Consists of 60,000 training images and 10,000 test images

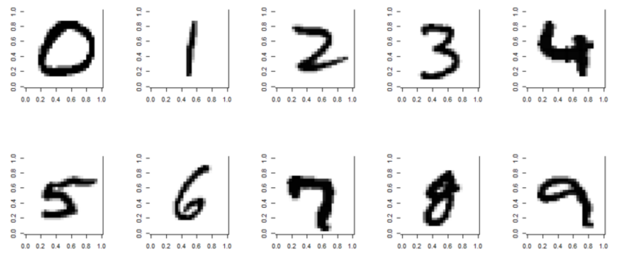

Fig.1: MNIST character sample (characters 0-9)

Neural Networks Utilized for this Project

Feed-forward neural network

- This network consists of three fully connected layers with 500 nodes each
- The 28x28 images are being resized into flat 784 vector to then be passed through the computation graph
- In each layer, the vector is multiplied by the weights and biases unique to the individual nodes.
- The Softmax function is applied and then a ReLU function acts as threshold for the neurons to fire (except for the output layer)
- Cross entropy is used to evaluate the loss or cost for each training iteration
- Cost is then being minimized through a gradient descent algorithm

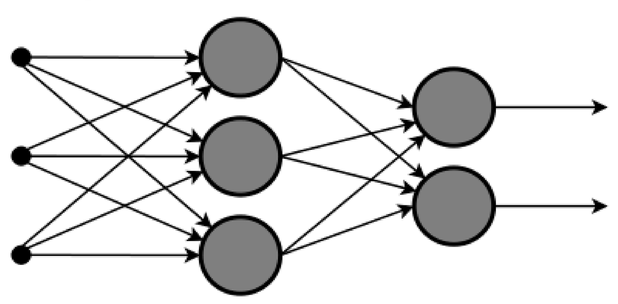

Fig.2: Feed-forward neural network sample structure (left to right: 3 input nodes, 3 hidden layer nodes, 2 output nodes)

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j)}$$

Fig.3: Softmax function being applied to nodes to normalize output

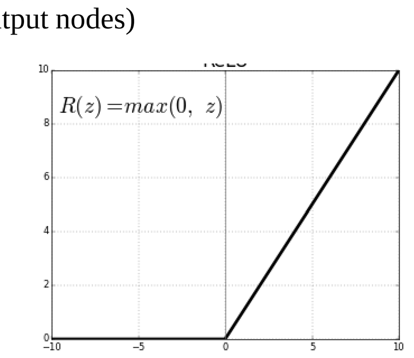

Fig.4: Softplus function (ReLU), which serves as activation function for the networks

Convolutional feed-forward neural network
2 convolutional layers added before the fully connected layers
After each layer the images will be down-sampled through max pooling with a 2x2 stride
Then one fully connected layer is added that functions like a single layer in the first model
For training, the dropout rate will be 25%

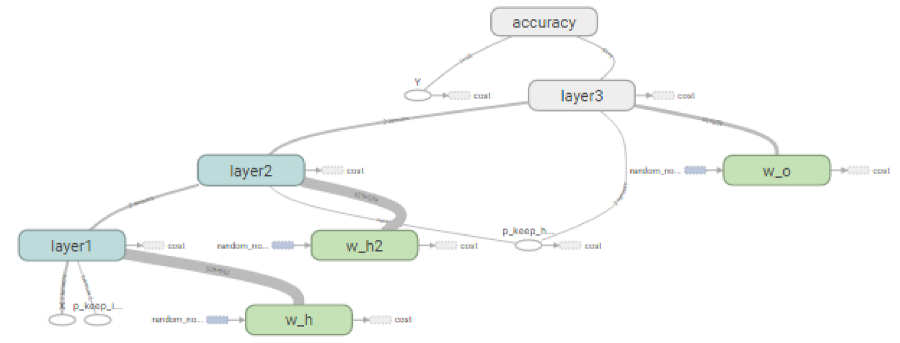

Fig.6: Computation graph for feed-forward neural network (created with TensorBoard)

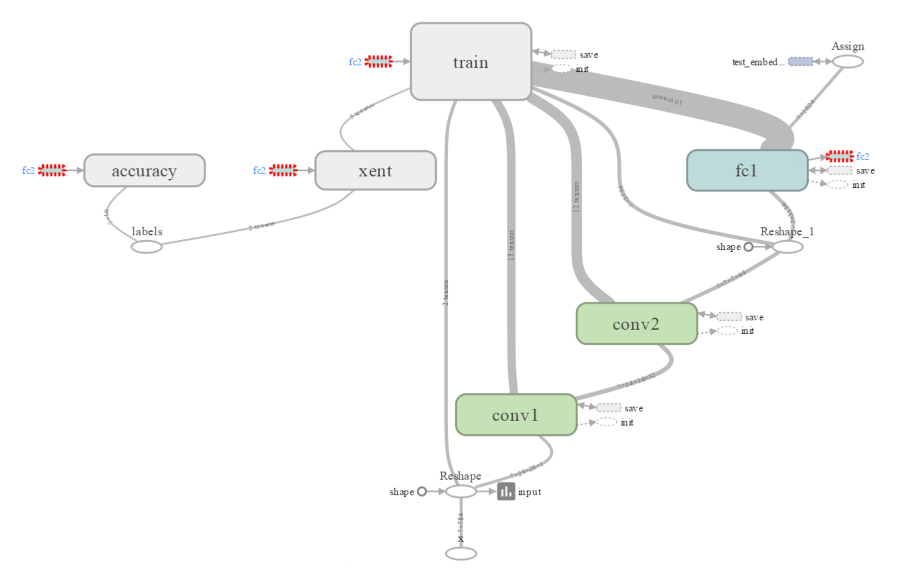

Fig.7: Computation graph for convolutional neural network (created with TensorBoard)

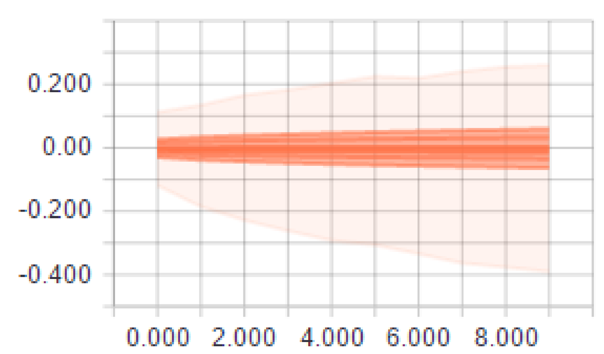

Fig.8: Network 1 weight distribution in the first layer over 9000 iterations

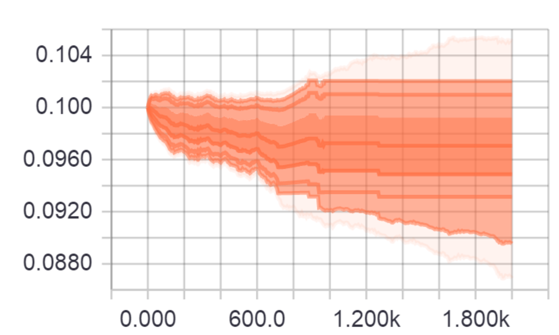

Fig.10: Network 2 bias distribution in the fully connected layer

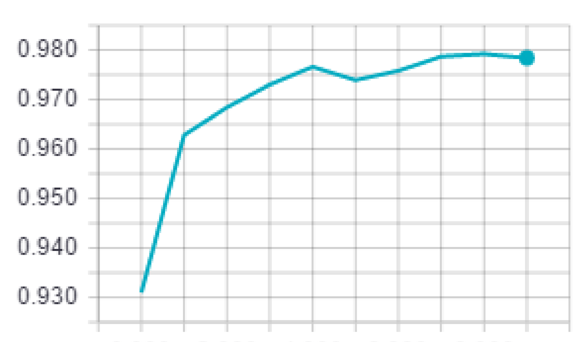

Fig.9: Achieved accuracy of network 1 over 9000 iterations (time taken: ~22min)

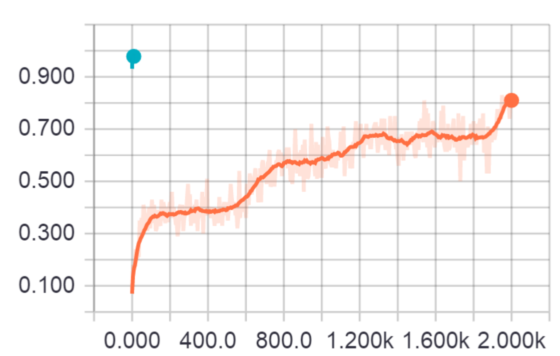

Fig.11: Achieved accuracy of network 2 over 200,000 iterations (time taken ~80 minutes)

Recurrent neural network

- TensorFlow implementation still being worked on
- For the purpose of this project however, the network analyzed in "Recurrent Convolutional Neural Network for Object Recognition" (Liang, et al.) will be considered
- This network is comprised of one convolutional layer, four recurrent convolutional layers, and one Softmax layer in that order
- After L1, L3, and L5 a max pooling layer is employed
- In the recurrent layers consists of and unfolding of the input over four steps, each one being influenced by the previous through a recurrent input
- The first convolutional layer and the Softmax layer are non-recurrent.

Applications for Neural Networks

- Character recognition, especially in datasets as controlled as MNIST, is not the most difficult machine learning task
- Other methods, including statistical classifiers (based on Bayes decision theory) like a k nearest neighbor classifier perform almost equally well on this task [LeCunn et al]
- Neural networks show the best results by minimal margins [LeCunn et al] on character recognition, however, have proven extremely efficient in more complex tasks.
- The reason character recognition has been chosen as a baseline for this task is the similarity it has to many of these more advanced applications for neural networks, including object recognition, scene detection, and even gesture of face recognition.

What the results indicate for the task of scene analysis

- Scene analysis searches to analyze a picture (often of outdoor scenes) for certain structures by interpreting low level structures
- This is employed in self-driving cars and in even mars rovers [Castano et al. 1] and an area of continuing research
- Benefits neural networks offer based on the testing results:
  - Enormous precision, unmatched by most other machine learning methods employed for similar tasks, which makes them the most reliable solution.
  - The models are very flexible, with a similar architectural approach to most problems. Limits are mostly set by training data available.
  - Different architectures can easily be combined, as with the example networks, where one network always builds on the previous one and expands their models.
- Potential drawbacks indicated by testing results
  - Enormously time consuming to train, with even the simplest model taking around 22 minutes to walk through one training cycle. This is a result of the complex operations performed on the computation graph (but certainly also affected by hardware limits). While this problem, to an extent, can be resolved through more sophisticated hardware (i.e. performing vector computations by using V-RAM rather than the CPU), some tasks inherently limit the hardware that can be used (for instance, room on a mars rover is very limited). However, after the training is done, classification will be done almost instantaneously.
  - Precision depends highly on vast training data. While this is less of a problem in the age of 'big data', for novel tasks, like navigating on Mars' surface, this can present a serious issue.

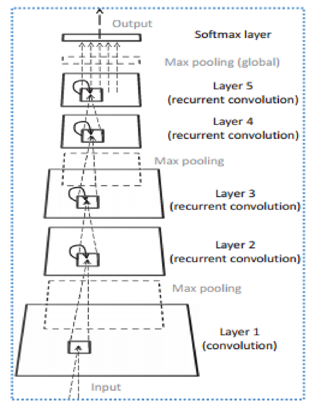

Fig.12: Layers in the recurrent neural network as proposed by Liang et al.

Contemporary Implementations of Neural Networks for Scene Analysis

- 'Salience-based visual attention model', proposed by Itti et al. is based off of visual systems found in nature and a strictly hierarchical [Itti et al. 1254].
  - In this model, several different convolutions and feature extraction methods are employed to extract prominent features in the scene. This is employed in a hierarchical structure not unlike that of the visual cortex, with basic feature detectors (e.g. for color) on the bottom of the hierarchy and more complex detectors on the top, culminating in a "salience map" that is then analyzed by a winner-takes-all type fully connected layer [Itti et al. 1254].
  - This combination of more sophisticated convolutions culminating in a single feature map is something that could easily be achieved by building on the basic code provided in the second neural net in this project
  - This network, being built for rapid scene analysis, has near-human performance in pop-out tasks [Itti et al. 1258] that require the fast analyzing of a given scene, making this type of network especially effective for environments that require fast computation like navigating traffic as a self-driving car
- 'Region-based convolutional networks', as proposed by Girshick et al., again work in a hierarchical fashion.
  - These networks first go over an image to segment it in a recurrent fashion. Segments are then handled by specialized sub-networks to allow for good domain-specificity [Girshick 2].
  - While this approach requires very diligent and extensive training, it is so far the best performing model overall [Girshick 14] and widely employed for object recognition tasks that require high accuracy.

Conclusion

This project is still a work in progress. So far this project has successfully shown that even very simple neural network implementations bear great potential for simple classification problems on one hand, and, if correctly recombined, for more sophisticated problems on the other hand.

In the future, this project will be expanded to include an implementation of the recurrent neural network in tensorflow, as well as a recombination of these networks to tackle a tougher problem and demonstrate the inherent power if neural nets in general.

Referenceses

- Castano, R., Judd, M. Anderson, R. and Estling, T. Machine Learning Challenges in Mars Rover Traverse ScienceWeb. 3 Mar. 2017. http://ml.jpl.nasa.gov/papers/castano/castano-ICML03.pdf.
- Girshick, R. Donahue, J. Darrell, T. and Malik, J. Region-based Convolutional Networks for Accurate Object Detection and Segmentation. Web. 3 Mar. 2017. https://people.eecs.berkeley.edu/~rbg/papers/pami/rcnn_pami.pdf.
- Itti, L. Koch, C. and Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.11 (1998): 1254-259. Web. 3 Mar. 2017.
- LeCunn, Y. Cortes, C. and Burges, C. "THE MNIST DATABASE of handwritten digits." *THE MNIST DATABASE of handwritten digits*. N.p., n.d. Web. 3 Mar. 2017. http://yann.lecun.com/exdb/mnist/.
- Liang, M. and Hu, X. Recurrent Convolutional Neural network for Object Recognition . Web. 3 Mar. 2017. http://www.cv-foundation.org/openaccess/content_cvpr_2015/app/2B_004.pdf.

Image sources

- Fig. 1: http://cdn-ak.f.st-hatena.com/images/fotolife/T/TJO/20150225/20150225130052.png
- Fig. 2: https://upload.wikimedia.org/wikipedia/commons/thumb/0/00/Multi-Layer_Neural_Network-Vector-Blank.svg/400px-Multi-Layer_Neural_Network-Vector-Blank.svg.png
- Fig. 4: http://csci431.artifice.cc/images/relu.png
- Fig. 12: http://www.cv-foundation.org/openaccess/content_cvpr_2015/app/2B_004.pdf