COMP 90051 Statistical Machine Learning Project 2
Kaggle Competition: Digits Recognition

Student Name: Yueni Chang (ID: 884622, Kaggle Name: nikiki)
Date: Oct 09 2017
Word Count: 399 words.

Data Pre-process

1. Classes are well-balanced in training set (see Fig.1).
2. Most digits located in the middle. Different levels of blur, rotation, shifting and scaling issues exist. To deal with them and improve model's tolerance of noises, random rotation, zooming and shifting is applied on training data.

Model Selection

1. Application Package: Tensorflow, Keras
2. Model: Convolutional neural network(CNN)
3. Model Architecture: See Table.1 for details. The final chosen model is model 5.

Model Optimisation and Evaluation

1. Enhancement:
   - Using cross entropy instead of classification accuracy since it reflects how good the weights are chosen, not just how many correct predictions are made.
   - Increasing the epoch numbers from 30-100
   - Adding more convolutional layers to implement a simplified version of VGG-16.
   - Skipping the normalisation after each pooling
   - Dropout before final layer to avoid over-fitting

2. Evaluation and Analysis

All models and tests are run on the same computer with identical environment settings. Time consumption reflects the computational complexity computation and its cost. Model comparison shows (Table. 1):

- Original VGG-16 model hires 16 layers in total. Adding convolutional layers from 3 to 6, then 8 attracted significant improvement. Smaller filers and consecutive convolutional layers provide a higher level of non-linearity and allow extraction of deeper features.
- Comparing model 4 and 5, simply going deeper does not always improve the model much. The simpler model learns faster in our case. Features of higher levels are constructed from the lower level ones, if going too deep, the model may take time to extract useful features (See Fig.3 and Fig.4 for learning rate comparison).
- Model 3 and model 5 have identical settings except the normalisation and epoch numbers. From Fig.4, model 5 learns pattern very fast under such architecture, the improvement from 10 epochs to 100 epochs is not very high. Actually, in our observation, validation accuracy stabilises at 99.5 after epoch 30 while training accuracy experiences graduate increase, which reveals the risk of over-fitting (See Fig.2 to compare the prediction distribution).
- Removing normalisation allows higher level layers to extract exact weight information from lower layers, avoiding the loss of feature weights.
- The final submitted model is model 5, however, considering the randomness and average score, without giving more test case, performance of the best three model are very close. The bottleneck may be posed by the design of underlying structure.

Appendix

Model	Adding Layers	Max Pool	Epochs/ Avg.time	Pub/s Prv/s	Train/a Test/a
1	Conv2D(32,3,3) Conv2D(64,3,3) Conv2D(128,3,3) FC-256, FC-256, FC-10 paras: 2,258,442	(2,2) (2,2) (2,2) with norm	30 23s	88.80 94.00	92.99 93.43
2	Conv2D(32,3,3) × 2 Conv2D(64,3,3) × 2 Conv2D(128,3,3) × 2 FC-256, FC-256, FC-10 paras: 2,452,202	(2,2) (2,2) (2,2) with norm	70 36s	No Submiss-ion	94.21 95.28
3 Best	Conv2D(32,3,3) × 2 Conv2D(64,3,3) × 2 Conv2D(128,3,3) × 2 Conv2D(256,3,3) × 2 FC-256, FC-256, FC-10 paras: 2,288,874	(2,2) (2,2) (2,2) (2,2) with norm	10 40s	97.20 98.00	97.56 98.53
			30 42s	98.00 98.40	98.19 99.06
4	Conv2D(32,3,3) × 2 Conv2D(64,3,3) × 2 Conv2D(128,3,3) × 2 Conv2D(256,3,3) × 2 Conv2D(512,3,3) × 2 FC-1024,FC-256,FC-10 paras: 7,074,794	(2,2) (2,2) (2,2) (2,2) (2,2) with norm	70 53s	98.00 98.00	98.79 99.52
5 Best	Conv2D(32,3,3) × 2 Conv2D(64,3,3) × 2 Conv2D(128,3,3) × 2 Conv2D(256,3,3) × 2 FC-256, FC-256, FC-10 paras: 2,288,874	(2,2) (2,2) (2,2) (2,2) w/o norm	100 42s	98.80 98.80	99.31 99.61

-paras: Number of parameters
-norm: Normalisation after pooling
-avg. time: average time for a single epoch
-pub/s , prv/s : public score, private score
-train/a, trest/a : training-accuracy, test accuracy.

-Conv2d (N, a, b): Convolutional layer with N filters, filter size is a by b.
-MaxPool(a,b): Apply MaxPooling on the layer with pool size of a by b.

Fig. 1

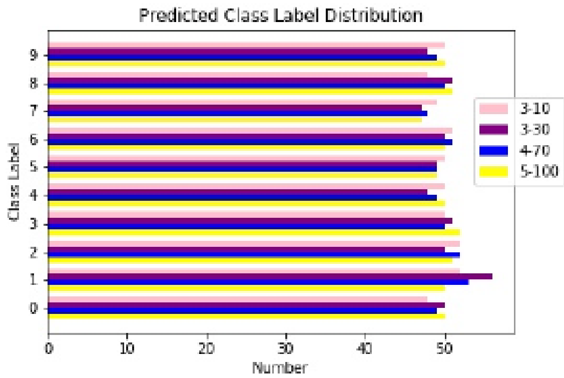

Fig. 2 Predicted class label distribution, labelled with model number and epoch number. Prediction by different models are quite balanced among classes excpet 3-30 on class 1 and 3-10 on class 0 .

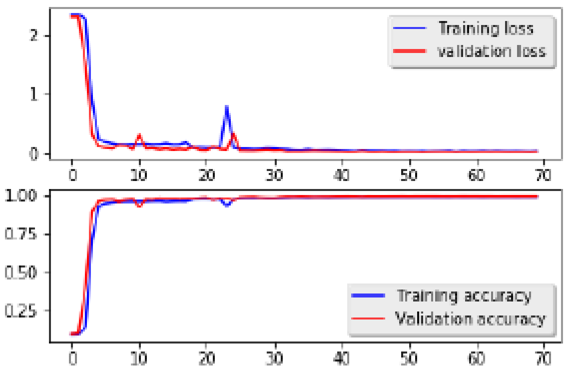

Fig. 3 Model 4 learning rate

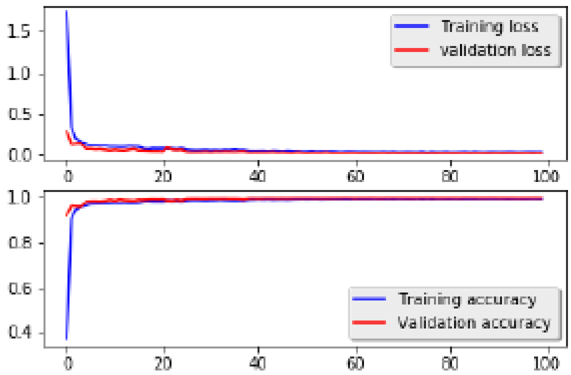

Fig. 4 Model 5 learning rate

Bibliography

-Videos & Examples

ML Lecture 10: Convolutional Neural Network."YouTube. N.p., 2017. Web. 1 Oct. 2017.
Convolutional Neural Networks | Tensorflow."TensorFlow. N.p., 2017. Web. 2 Oct. 2017.
Cs231n Convolutional Neural Networks For Visual Recognition."Cs231n.github.io. N.p., 2017. Web. 2 Oct. 2017.
Lqkhoo/Cs231n-Spring-2017."GitHub. N.p., 2017. Web. 2 Oct. 2017.

-Reading materials

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Sze, V., Chen, Y.H., Yang, T.J. and Emer, J., 2017. Efficient processing of deep neural networks: A tutorial and survey. arXiv preprint arXiv:1703.09039.