Building A Text Recognition AI #2

TRANSLUE

Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

3rd Meeting – Monday, September 12th 2022

Review - How could we build our own text recognition AI?

Proposed Ingredients:

- Image Scaler
    - Resize or crop images to have uniform dimensions (AI *often* needs uniform input) [1]
- Feature Extraction Unit
    - Convolutional Neural Network to locate image patches with text [2] [3]
- Integral Embedding Extractor
    - Learns visual and contextual feature embeddings for each detected integral text unit [4]
- Contextual Text Block Generator
    - Groups and arranges the detected integral texts in reading order to produce contextual text blocks [4]
- Character Classification Unit
    - Convolutional Neural Network to find characters in obtained image patches [2] [3]

→ How could this structure be improved?

→ What has to be considered while running on smartphones?

How could this structure be improved?

Proposed Ingredients:

- Image Scaler
  - Genuninely needed, because [5] and need for noise reduction
  - Resize or crop images to have uniform dimensions (AI *often* needs uniform input) [1]
- Feature Extraction Unit
  - Convolutional Neural Network to locate image patches with text [2] [3]
- Integral Embedding Extractor
  - Learns visual and contextual feature embeddings for each detected integral text unit [4]
- Contextual Text Block Generator
  - Groups and arranges the detected integral texts in reading order to produce contextual text blocks [4]
- Character Classification Unit
  - Convolutional Neural Network to find characters in obtained image patches [2] [3]

→ How could this structure be improved?

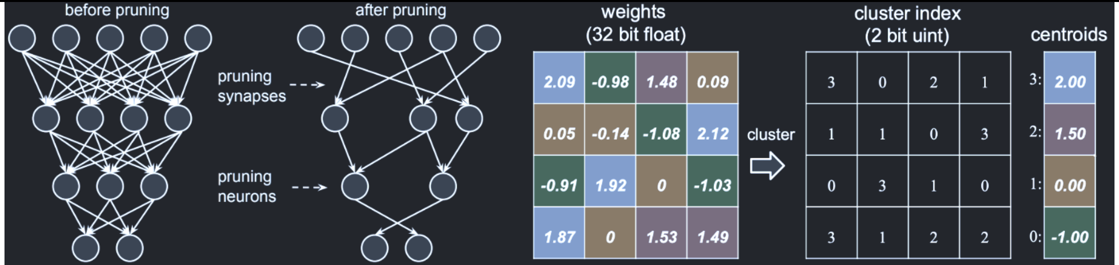
→ What has to be considered while running on smartphones?

Considerations for the Smartphone environment

- Computation workloads can’t be delegated to the cloud
  -> <u>Translue is highly sensitive to latency changes</u>
- Reducing complexity means boosting processing speed [6]
  - Pruning (Removes the redundant elements in neural networks)
  - Truncated Singular Value Decomposition (simplifies layers of CNNs)
  - Knowledge distillation (learn findings of bigger prototype system)

  → Can achieve 98.6+% accuracy for a „lightweight“ model [7]
- Reducing system size reduces memory needs [6]
  - Quantization (Compressing learned parameters into small data types)

Pruning Goals:
Reduce the model size and computation cost



weights (32 bit float)

2.09	-0.98	1.48	0.09
0.05	-0.14	-1.08	2.12
-0.91	1.92	0	-1.03
1.87	0	1.53	1.49

cluster index (2 bit uint)

3	0	2	1
1	1	0	3
0	3	1	0
3	1	2	2

centroids

3:	2.00
2:	1.50
1:	0.00
0:	-1.00

[6]

References

[1] - https://medium.com/mindboard/image-classification-with-variable-input-resolution-in-keras-cbfbe576126f, 30.08.22

[2] - Yoshihashi, Ryota, et al. "Context-Free TextSpotter for real-time and mobile end-to-end text detection and recognition.“ (Yahoo)

[3] - Bartz, Christian, Haojin Yang, and Christoph Meinel. "STN-OCR: A single neural network for text detection and text recognition.“ (HPI)

[4] - Xue, Chuhui, et al. "Contextual Text Block Detection towards Scene Text Understanding.“ (ByteDance)

[5] - Parés Sabatés, Ferran, et al. "Training CNNs using high-resolution images of variable shape." (Barcelona Supercomputing Center)

[6] - Cai, Han, et al. "Enable deep learning on mobile devices: Methods, systems, and applications.“ (MIT)

[7] – Hinton, Geoffrey, et al. “Distilling the Knowledge in a Neural Network“ (Google)