System Demo for Transfer Learning from Vision to Language using Domain Specific CNN Accelerator for On-Device NLP Applications

Anonymous Authors[1]

Abstract

Power-efficient CNN Domain Specific Accelerator (CNN-DSA) chips are currently available for wide use in mobile devices. These chips are mainly used in computer vision applications. However, the recent work of Super Characters method for text classification and sentiment analysis tasks using two-dimensional CNN models has also achieved state-of-the-art results through the method of transfer learning from vision to text. In this paper, we implemented the text classification and sentiment analysis applications on mobile devices using CNN-DSA chips. Compact network representations using one-bit and three-bits precision for coefficients and five-bits for activations are used in the CNN-DSA chip with power consumption less than 300mW. For edge devices under memory and compute constraints, the network is further compressed by approximating the external Fully Connected (FC) layers within the CNN-DSA chip. At the workshop, we have two system demonstrations for NLP tasks. The first demo classifies the input English Wikipedia sentence into one of the 14 ontologies. The second demo classifies the Chinese online-shopping review into positive or negative.

1. Introduction

Power-efficient CNN Domain Specific Accelerator (CNN-DSA) chips are currently available for wide use. Sun et al. (Sun et al., 2018c;a) designed a two-dimensional CNN-DSA accelerator which achieved a power consumption of less than 300mW and an ultra power-efficiency of 9.3TOPS/Watt. All the processing is in internal memory instead of external DRAM. Demos on mobile and embedded systems show its applications in real-world implementations. The 28nm CNN-DSA accelerator attains a 140fps for 224x224 RGB image inputs at an accuracy comparable to that of the VGG (Simonyan & Zisserman, 2014).

Figure 1. Efficient On-Device Natural Language Processing system demonstration. The CNN-DSA chip is connected to Raspberry Pi through the USB interface. Keyboard sends the typing text input to Raspberry Pi through USB. A monitor is connected to Raspberry Pi through HDMI for display. On the monitor, it shows the introduction for the demo (zoom in to see details). There are two demos. The first demo classifies the input English Wikipedia sentence into one of the 14 ontologies. The second demo classifies the Chinese online-shopping review into positive or negative.

For Natural Language Processing tasks, RNN and LSTM models (Tang et al., 2015; Lai et al., 2015) are widely used, which are different network architectures from the two-dimensional CNN. However, the recent work of Super Characters method (Sun et al., 2018b; 2019b) using two-dimensional word embedding achieved state-of-the-art result in text classification and sentiment analysis tasks, showcasing the promise of this new approach. The Super Characters method is a two-step method. In the first step, the characters of the input text are drawn onto a blank image, so that an image of the text is generated with each of its characters embedded by the pixel values in the two-dimensional space. The resulting image is called the Super Characters image. In the second step, the generated Super Characters image is fed into a two-dimensional CNN models for classification. The two-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

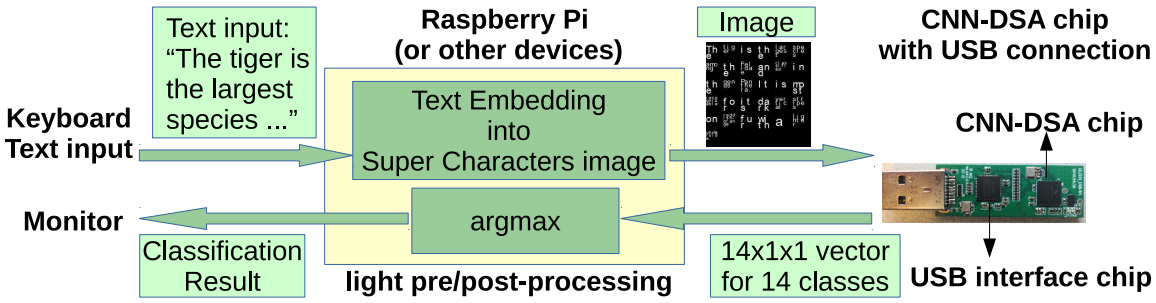

Figure 2. Data flow for the system demonstration of efficient on-device NLP using Super Characters method and CNN-DSA chip.

dimensional CNN models are trained for the text classification task through the method of Transfer Learning, which finetunes the pretrained models on large image dataset, e.g. ImageNet (Deng et al., 2009), with the labeled Super Characters images for the text classsification task.

In this paper, we implemented NLP applications on mobile devices using the Super Characters method on a CNN-DSA chip as shown in Figure 1. It takes arbitrary text input from keyboard connecting to a mobile device (e.g. Raspberry Pi). And then the text is pre-processed into a Super Characters image and sent to the CNN-DSA chip to classify. After post-processing at the mobile device, the final result will be displayed on the monitor.

2. System Design and Data Flow

As shown in Figure 2, the keyboard text input is pre-processed by the Raspberry Pi (or other mobile/embedded devices) to convert into a Super Characters image. This pre-processing is only a memory-write operation, which requires negligible computation and memory resources.

The Super Characters method works well for Asian languages which has characters in squared shapes, such as Chinese, Japanese, and Korean. These glyphs are easier for CNN models to recognize than Latin languages such as English, which is alphabets-based in a rectangular shape and may have to break the words at line-changing. To improve the performance for English, a method of Squared English Word (SEW) is proposed to cast English word in a squared shape as a glyph (Sun et al., 2019a). Figure 3 shows an example of this method. Basically, each word takes the same size of a square space lxl. Words with longer alphabets will have smaller space for each alphabet. Within the lxl space, the word with N alphabets will have each of its alpha in the square area of $\{l/ceil[sqrt(N)]\}^2$, where $sqrt(.)$ stands for square root, and $ceil[.]$ is rounding to the top.

The CNN-DSA chip receives the Super Characters image through the USB connection to the mobile device. It outputs the classification scores for the 14 classes in the Wikipedia text classification demo. The classification scores mean the probabilities for classification but before softmax. The mobile device only calculate the argmax to display final classification result on the monitor, which is also negligible computations. The CNN-DSA chip completes the complex CNN computations with low power less than 300mw.

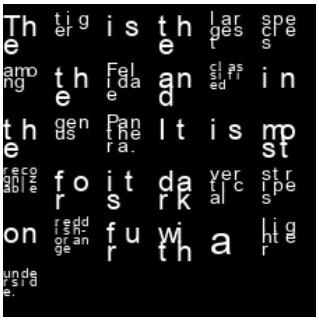

Figure 3. An example for Squared English Word (SEW) method. The two-dimensional embedding in this image corresponds to the text of ”The tiger is the largest species among the Felidae and classified in the genus Panthera. It is most recognizable for its dark vertical stripes on reddish-orange fur with a lighter underside.”

3. Compact Network Representations for Efficient Inference

3.1. Approximating FC layers for On-Device Applications under Memory and Computation Constraints

The CNN-DSA chip is a fast and low-power coprocessor. However, it does not directly support inner-product operations of the FC layers. It only supports 3x3 convolution, Relu, and max pooling. If the FC layers are executed on

the mobile device, there will be increased requirements for memory, computation, and storage for the FC coefficients. And it will also spend more interface time with the CNN-DSA chip for transmitting the activation map from the chip, and also cost relative high power consumption for the mobile device to execute the inner-product operations.

In order to address this problem, we proposed the GnetFC model, which approximates the FC layers using multiple layers of 3x3 convolutions. This is done by adding a sixth major layer with three sub-layers as shown in Figure 4. The

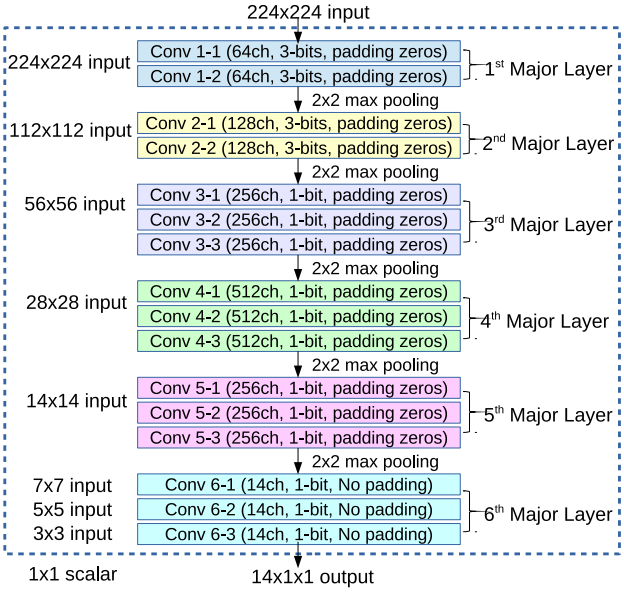

Figure 4. Model architecture. The input is of size 224x224 with multiple channels, and the output is of size 14x1x1. The architecture within blue dashed square is the model loaded into the CNN-DSA chip.

model is similar to VGG architecture except that it has six major layers, and the channels in the fifth major layer is reduced to 256 from the original 512 in order to save memory for the sixth layer due to the limitation of the on-chip memory. The sub-layers in each major layer has the same color. Each sub-layer name is followed by the detailed information in brackets, indicating the number of channels, bits-precision, and padding. The first five major layers has zero paddings at the image edge by one-pixel. But the sixth major layer has no padding for the three sublayers, which reduces the activation map from 7x7 through 5x5 and 3x3 and finally to 1x1. The output is of size 14x1x1, which is equal to an array of 14 scalars. The final classification result can be simply obtained by an argmax operation on the 14 scalars. This reduces the system memory footprint on the mobile device and accelerate the inference speed.

3.2. Low-precision Inference in the Chip

The memory of the CNN-DSA chip is built within the accelerator, so it is very power-efficient without wasting the energy for moving the bits from external DDR into internal SRAM. Thus the on-chip memory is very limited, which supports maximum 9MB for coefficients and activation map. As shown in Figure 4, the first two major layers uses 3-bits precision and the other four major layers uses 1-bit precision. All activations are presented by 5-bits in order to save on-chip data memory. The representation mechanism inside the accelerator supports up to four times compression with the 1-bit precision, and two times compression with the 3-bits precision. Due to the high compression rate, the convolutional layers in VGG16 with 58.9MB coefficients in floating precision could be compressed into only about 5.5MB within the chip. This is a more than 10x compression of the convolution layers. This compact representation has been proved to be successful on ImageNet (Deng et al., 2009) standard training and testing data and achieved the same level of accuracy as floating point models with 71% Top1 accuracy. The compact CNN representation without accuracy loss is because of the redundancy in the original network.

To efficiently use the on-chip memory, the model coefficients from the third major layers are only using 1-bit precision. For the first two major layers, 3-bits model coefficients are used as fine-grained filters from the original input image. And the cost on memory is only a quarter for the first major layer and a half for the second major layer if using the same 3-bits precision.

The total model size is 2.8MB, which is more than 200x compression from the original VGG model with FC layers. It completes all the convolution and FC processing within the CNN-DSA chip for the classification task with little accuracy drop. The GnetFC model on the CNN-DSA chip on the Wikipedia demo obtains an accuracy of 97.4%, while the number for the original VGG model is 97.6%. The accuracy drop is mainly brought by the approximation in GnetFC model, and also partially because of the bit-precision compression. The accuracy drop is very little, but the savings on power consumption and increasing on the inference speed is significant. It consumes less than 300mw on the CNN-DSA chip, and the power for pre/post-processing is negligible. The CNN-DSA chip processing time is 15ms, and the pre-processing time on mobile device is about 6ms. The time for post-processing is negligible, so the total text classification time is 21ms. It can process nearly 50 sentences in one second, which satisfies more than real-time requirement for NLP applications.

4. Conclusion

We implemented efficient on-device NLP applications on a 300mw CNN-DSA chip by employing the two-dimensional embedding used in the Super Characters method. The two-dimensional embedding converts text into images, which is ready to be fed into CNN-DSA chip for two-dimensional CNN computation. The demonstration system minimizes the power consumption of deep neural networks for text classification, with less than 0.2% accuracy drop from the original VGG model. The potential use cases for this demo system could be the intension recognition in a local-processing smart speaker or Chatbot.

References

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Lai, S., Xu, L., Liu, K., and Zhao, J. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pp. 2267–2273, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sun, B., Liu, D., Yu, L., Li, J., Liu, H., Zhang, W., and Torng, T. Mram co-designed processing-in-memory cnn accelerator for mobile and iot applications. *arXiv preprint arXiv:1811.12179*, 2018a.

Sun, B., Yang, L., Dong, P., Zhang, W., Dong, J., and Young, C. Super characters: A conversion from sentiment classification to image classification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 309–315, 2018b.

Sun, B., Yang, L., Dong, P., Zhang, W., Dong, J., and Young, C. Ultra power-efficient cnn domain specific accelerator with 9.3 tops/watt for mobile and embedded applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1677–1685, 2018c.

Sun, B., Yang, L., Chi, C., Zhang, W., and Lin, M. Squared english word: A method of generating glyph to use super characters for sentiment analysis. *arXiv preprint arXiv:1902.02160*, 2019a.

Sun, B., Yang, L., Zhang, W., Lin, M., Dong, P., Young, C., and Dong, J. Supertml: Two-dimensional word embedding and transfer learning using imagenet pretrained cnn models for the classifications on tabular data. *arXiv preprint arXiv:1903.06246*, 2019b.

Tang, D., Qin, B., and Liu, T. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1422–1432, 2015.