Huffman Coding Homework

National Chiao Tung University
Chun-Jen Tsai
3/21/2023

Homework Goal

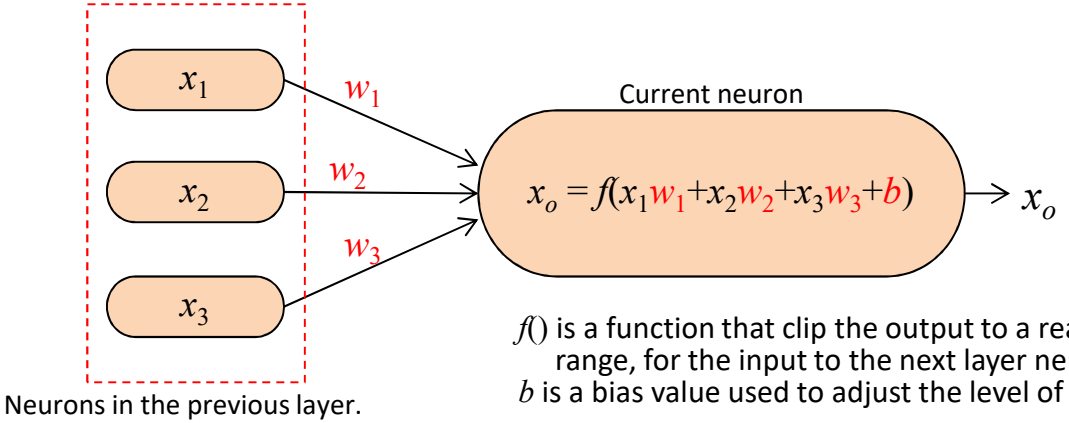
- For this homework, you will write a program to test if the adaptive Huffman coding algorithm can improve compression on a large binary file
  - Your code must be written in either C++ or Python
  - You must upload the source to E3 for TA to check your result
  - You must write a short report to summarize your discovery. Only PDF format, please.
- Upload your code and report to E3 by 4/7, 17:00pm.

About the Data File

- We will use a neural network model file, `alexnet.pth`, as the target binary data for compression
  - The file format is the Python pickle file generated using the neural network framework PyTorch
  - The pretrained model is quantized to reduce the entropy
  - The file is available at the following link: https://drive.google.com/file/d/1fZBC3OLHhDmcUT4KprdswK3KjOMCSbVQ/view?usp=share_link
- PyTorch is used to generate the data file
  - PyTorch is a library for the Python programming language for neural network experiments: https://pytorch.org/
  - You do NOT have to use PyTorch for this homework

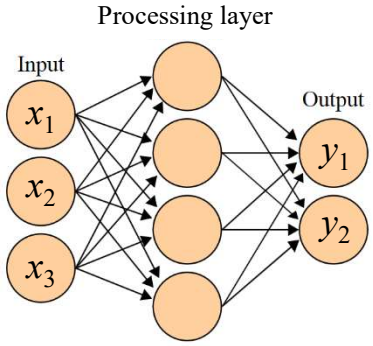
Neural Network Model (1/2)

- Computation model for a single neuron:



$f()$ is a function that clip the output to a reasonable range, for the input to the next layer neurons.
b is a bias value used to adjust the level of the inner product.

- A three-layer neural network (NN):

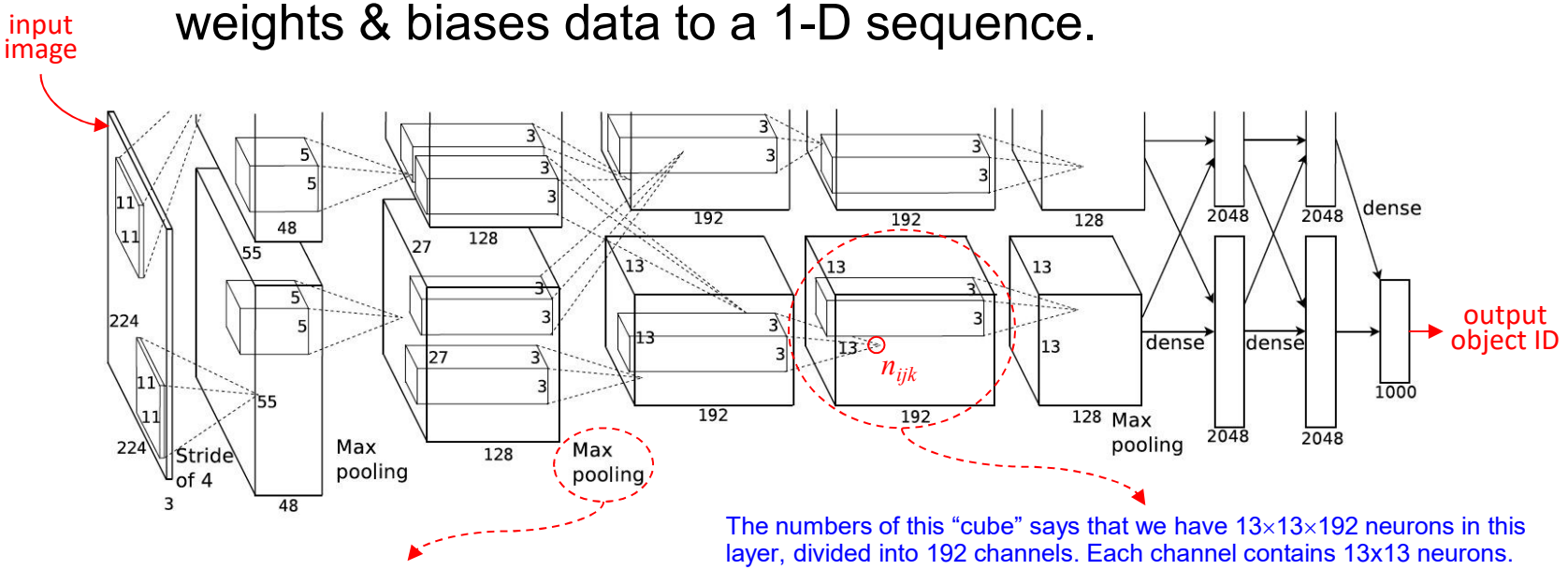


Neural Network Model (2/2)

- An NN model contains the input weights w_i of all the neuron links and the biases b of each neuron
  - These numbers are encoded in 32-bit floating-point format in our data file
- The neural network training process tries to find the weights and biases to satisfy the given input-output relations (specified by the training data set)

AlexNet Model Architecture

- AlexNet† can be used to recognize 1000 different types of objects given a 224×224 input image:
  - The pickle file, `alexnet.pth`, serialize the multi-dimensional weights & biases data to a 1-D sequence.



A neuron layer with max pooling means that the output layer width and height will be subsampled to a different size.

The numbers of this "cube" says that we have 13×13×192 neurons in this layer, divided into 192 channels. Each channel contains 13x13 neurons.

Each neuron value n_{ijk} is computed by a 3D inner product of previous layer neurons with a vector size 3×3×13 plus a bias value, then passing the result through an activation function (see page 4).

† A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *In Advances in Neural Info. Processing Systems*, 2012, pp. 1097-1105.

Goals of the HW

- You can treat the ANN model, `alexnet.pth`, as a data source
  - The "native" alphabet size should be 32-bit (the files contains 32-bit header fields and 32-bit floating-point numbers)
  - But, you can also assume that it is a 8-bit data source
- You must
  - Implement 8-bit and 32-bit Huffman coders to compress the data file
  - Implement a 8-bit Adaptive Huffman coder to compress the data file
  - Conduct some experiments based on your coders

Experiments You Should Do

- Try the basic Huffman algorithm for data compression
  - Performance measure: the average codeword length $E[l(x_i)]$
  - Treat the file as an 8-bit data source first, then treat it as a 32-bit source to see the differences in $E[l(x_i)]$
  - Check whether the p.m.f. changes throughout the data source
    - For example, you can estimate the p.m.f. on the entire file first, and then for every 40MB of the data, for example.
- Try the adaptive 8-bit Huffman algorithm, see if $E[l(x_i)]$ can be reduced, compare to the basic algorithms
- Try other Huffman algorithms (such as the extended Huffman code) if you have time

Coding Instructions

- You must use C/C++ or Python for the HW, using open-source library is fine
  - You have to upload your source code in an archive file (in zip or tgz format).
  - You should put a readme file in the package with detail instructions about how to build and run your programs.
  - If the readme file is not written clearly enough for the TAs to reproduce your results, you will get 0 points for this HW!

Evaluation of the Homework

- The report shall be in two-column format, single-space text, font size 10, with up to 3 pages (hard threshold). You can use the report template (in both word and pdf format) on E3
- Your report must contains the following sections
  - Abstract, Introduction, Implementation Details, Open-Source Usage, Experimental Results, Discussions
  - The section "Open-Source Usage" must provides the details on the libraries and LLM models you use for this HW.
- Your grade will be based on
  - Thoroughness of your experiments
  - Implementation difficulties of your code
  - Clarity and completeness of information