Lab 2 introduction

General observations from lab 1

- Code quality generally good (and well commented)
- Discussion forum worked well
  - Please continue to use it for lab 2
  - We encourage both asking and answering questions
- Common programming errors from previous years avoided
  - Normalizing vectors correctly
  - sum(W) vs sum(W, ‘all’)
- Errors mostly related to
  - Warmup exercise (finding filter and threshold)
  - Gradients and Gradient histograms
- 1.13 (discuss how to select radius automatically)
  - A new type of error: Verbose and/or circular answers
  - Suspected LLM answers passed on to Fredrik
- See canvas announcement for more details on common issues

Using Python

- You are welcome to do so
- It will be your own responsibility
  - To translate provided matlab functions
  - To format your Jupyter Notebooks as well the MATLAB MLX notebooks

Lab 2

- Part 1: Learning a linear classifier
  - Challenging part: compute correct gradients
  - Use the chain rule

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial p}\frac{\partial p}{\partial y}\frac{\partial y}{\partial \theta}, \qquad \text{where } \theta \in \{w, w_0\}$$

  - L is the loss, p is the sigmoid activation, w is the filter (a matrix) and w_0 is the bias (a scalar)
  - Do one step at a time (first ∂L/∂p, then ∂p/∂y, …)
- Part 2: Training a CNN
  - Pretty straight-forward
  - Avoid common error by reading Lecture Notes Chapter 4-6
  - Reasoning about computational costs: Lecture notes 6.1
  - Recall and precision: Lecture notes 4.1

Bonus questions

- Lab 2, 3, and 4 have bonus question sets:
  - 2x3=6 points per set
  - Get 12/18 points to raise your grade by +1
- Sharing of answers or LLM usage will be reported.
  - Chalmers rules regarding cheating, disciplinary action & suspension
- Risking academic suspension in an attempt to get +1 to your grade is a very bad idea.

Bonus questions

- Everything you need to know is in
  - The lecture notes
  - Fredriks slides
  - The Deep Learning Book (chapter 9 especially)

Tips for computing the receptive field

- You get zero points for simply applying an equation you found googling.
- You will get at least partial points if you demonstrate you can reason properly regarding how receptive fields work.
- Try to make it simple:
  - First evaluate the receptive field in 1D (see image →)
  - Then think about the symmetry in the 2D setting
  - Assume a stride of 1 for convolutional layers
  - Assume a stride of 2 (and kernel size 2x2) for maxpooling layers
  - Remember that reducing both width and height by ½ reduces number of neurons by a factor 1/4

From: [The Deep Learning Book, ch9, page 332]

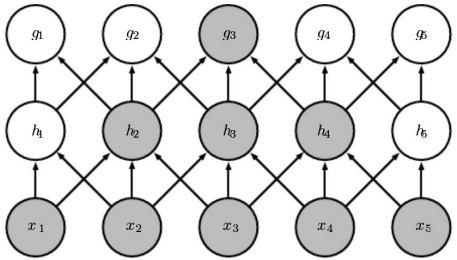

Figure 9.4: The receptive field of the units in the deeper layers of a convolutional network is larger than the receptive field of the units in the shallow layers. This effect increases if the network includes architectural features like strided convolution (figure 9.12) or pooling (section 9.3). This means that even though *direct* connections in a convolutional net are very sparse, units in the deeper layers can be *indirectly* connected to all or most of the input image.