INTRODUCTION TO NEURAL NETWORKS 67103 EX 1 - THEORETICAL PART ANSWERS

Question 1

1.1) Given two functions, f and g, a Convolution operator is defiened:
$(f * g)[n] = \sum_{k=-\infty}^{\infty} f(k) \cdot g(n-k)$
From the definition above we get:
$(f(k+t) * g(k))[n] = \sum_{k=-\infty}^{\infty} f(k+t) \cdot g(n-k) = \sum_{i=-\infty}^{\infty} f(i) \cdot g(n+t-i) = (f(i) * g(i))[n+t]$
where $i = k+t$.

A convolution can be expressed as a dot product by defining $h(k) = g(n-k)$ such that:
$(f * g)[n] = \sum_{k=-\infty}^{\infty} f(k) \cdot g(n-k) = \sum_{k=-\infty}^{\infty} f(k) \cdot h(k) = f \cdot h$.

1.2) The length of the output signal of a when concolving a signal of length n with a filter of length $k \leq n$ is: $n-k+1$.
when $n = k$ the output length is 1.

Question 2

2.1) Using the chain rule we get that the partial derivatives of the loss term with respect to f and b are:
$\frac{\partial F}{\partial f} = \frac{\partial F}{\partial Relu} \cdot \frac{\partial Relu}{\partial f} = \frac{\partial F}{\partial Relu} \cdot \frac{\partial Relu}{\partial (f*I+b)} \cdot \frac{\partial (f*I+b)}{\partial f}$
$\frac{\partial F}{\partial b} = \frac{\partial F}{\partial Relu} \cdot \frac{\partial Relu}{\partial b} = \frac{\partial F}{\partial Relu} \cdot \frac{\partial Relu}{\partial (f*I+b)} \cdot \frac{\partial (f*I+b)}{\partial b}$
It is given that the filter f and the bias b produce a negative response on an example datapoint I for all n examples.
Therfore the term $\frac{\partial Relu}{\partial (f*I+b)}$ always equals 0.
2.2) This means that in a gradient-decent process consisting solely on examples which produce negative responses, the gradients will always be zero and the filters and biases will not change during the run of the algorithm.

Question 3

3.1) We can look at a convolution as a multipication with a weight matrix that is mostly zeros except for the blocks that participate.
Furthermore, weights in many of the blocks are equal due to parameter sharing.
So we can write a given convolution as the following matrix:

$$W = \begin{matrix} f_k & f_{k-1} & \cdots & f_1 & 0 & 0 & \cdots & 0 \\ 0 & f_k & f_{k-1} & \cdots & f_1 & 0 & \cdots & 0 \\ 0 & 0 & f_k & f_{k-1} & \cdots & f_1 & \cdots & 0 \\ \cdots & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ & & \cdots & \cdots & & & & \\ & & & & & & & \\ & & & f_k & f_{k-1} & \cdots & f_1 & 0 \\ 0 & 0 & 0 & \cdots & f_k & f_{k-1} & \cdots & f_1 \end{matrix}$$

We have that the output is a matrix multipication of the weight matrix with an input signal I of length n:
$W \cdot I$, with only k pixels not zeroed in each row.
Deriving this term with respect to f leaves us with the pixels that weren't zeroed.
So we get:

$$\frac{\partial f * I}{\partial f} = \begin{array}{cccccc} I_k & I_{k-1} & \cdots & & & I_1 \\ I_{k+1} & I_k & I_{k-1} & \cdots & & I_2 \\ \cdots & & & & & \cdots \\ \cdots & & & & & \\ & & \cdots & \cdots & & \\ & & & & & \\ I_n & I_{n-1} & \cdots & & & I_{n-k+1} \end{array}$$

3.2) Now the jacobian matrix above gets multiplied by a vector v, coming from the previous layer:
$\left(v \cdot \frac{\partial f * I}{\partial f}\right)[i] = \sum_y^{n-k+1} v(y) \cdot I_{y+k-i}$.

Question 4

4.1) Convolutional layer: $(5 \cdot 5 \cdot 3 + 1)\, 96 = 7296$
4.2) Fully connected layer: $(4 \cdot 4 \cdot 256)^2 + 4 \cdot 4 \cdot 256 = 16,781,312$
4.3) Constraints: $100,000 \cdot 1,000 = 10,000,000$