INTRODUCTION TO NEURAL NETWORKS 67103 EX 2

1. Theoretical Part Answers

Answer 1

(1) conv1 7×7
Receptive field: 7×7
Filter weights: 7×7 weights + 1 bias weight. total of 50 weights
Multiplication: 49 multipications per pixel (number of pixels depands on the input and padding size).
(2) conv1 3×3, conv2 3×3, conv3 3×3
Receptive field: $(3+3-1)\times(3+3-1) = 5\times5 \rightarrow (5+3-1)\times(5+3-1) = 7 \times 7$
(3) conv1 5×5, conv2 3×3
Receptive field: $(5+3-1) \times (5+3-1) = 7 \times 7$
Filter weights: 5×5 weights + 1 bias weight for the first convolution and 3×3 weights + 1 bias weight for the scond concolution. total of 36 weights.
Multiplication: 25 + 9 multipications per pixel (number of pixels depands on the input and padding size).
(4) conv1 5×5 with stride 2, conv2 3×3
Receptive field: $(w-5)/2+1-3+1 = 1 \rightarrow (w-5)/2 = 2 \rightarrow w = 9$
so $w \times h = 9 \times 9$.
OR
conv1 5×5, stride 2 (pooling), conv2 3×3
Receptive field: $((w-5)+1)/2-3+1 = 1 \rightarrow (w-4)/2 = 2 \rightarrow w \times h = 10\times10$

Support as a function of the stride rate (convolution stride):
By taking a filter of size f and zero-interpolating it at the size of the stride, s, we get that the support of a single convolution (in 1 dimension) is:
$support = f \cdot s - 1 + 1 = f \cdot s$
For two convolutions the support is:
$support = (f_1 \cdot s - 1) + (f_2 \cdot s - 1) + 1 \approx s\,(f_1 + f_2)\,.$
We can conclude that the support grows linearly as a function of the convolution stride rate.

Answer 2

Given a shallow network: $f(x) = \sum_{i=1}^{n} \alpha_i (w_i x + b_i)$, we would like to describe it by a deep network with $O(n)$ neurons.
Our deep network will be similar to the construction we saw in class.
It is given that $\alpha_i s$ can be negative, so we will add layers that do the same propagation and summation to to all negative α_is, as we did in the first network for the positive $\alpha_i s$.
Notice that we are forwarding $-\alpha_i$s for negative α_is, so they are actualy positive values and the relu functions won't zero them.
Our last layer will calculate the output of the negative α_is minus the output of the originaly positve α_is as follows:

$\sum_{\alpha_i^+} \alpha\sigma\left(w_i x + b_i\right) - \sum_{\alpha_i^-} \alpha_i\sigma\left(w_i x + b_i\right) = \sum_{i=1}^{n} \alpha_i\sigma\left(w_i x + b_i\right) = f\left(x\right)$.
We can conclude that our deep network's output equals the shallow network's output with $\mathrm{O}(n)$ neurons.

Answer 3

No,
Even though we can find a local minimum in a fixed number of operations, the number of local minimums (or at least the search space) is exponential in the number of parameters.
hence the network training problem is still np-complete.