Mohamed Ehab Fathy
2002597

PAGE
DATE

Assignment 2

1) w. shape = (500 x 500 x 3 , 100)

b. shape = 100

2) 5x5x10 = 250 parameters

3) derivative filters For edge detection (Sobel)

* left image's Filter (vertical edges)

1	0	-1
2	0	-2
1	0	-1

* right image's Filter (horizontal edges)

1	2	1
0	0	0
-1	-2	-1

4) exp. moving avg. ⟶ momentum term in Adam.
(β_1) ($V_{dw} = \beta_1 V_{dw} + (1-\beta_1) dw$)

* as $\beta_1 \uparrow$, V_{dw} becomes more affected by the previous values (has longer memory) so it becomes smoother and more sluggish if the gradient is noisy (changing sign) and will become faster if gradients are in one direction

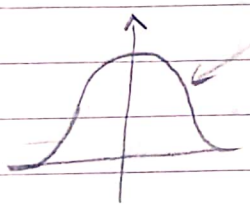
Hi . Star

5) $$\mu_{B_m} = \frac{1}{m} \sum_{i=1}^{m} z^i$$

$$\sigma^2_{B_m} = \frac{1}{m} \sum_{i=1}^{m} (z^i - \mu_{B_m})^2$$

$$\hat{z}_i = \frac{(z_i - \mu_{B_m})}{\sqrt{\sigma^2_{B_m} + \varepsilon}}$$, small value for numerical stability

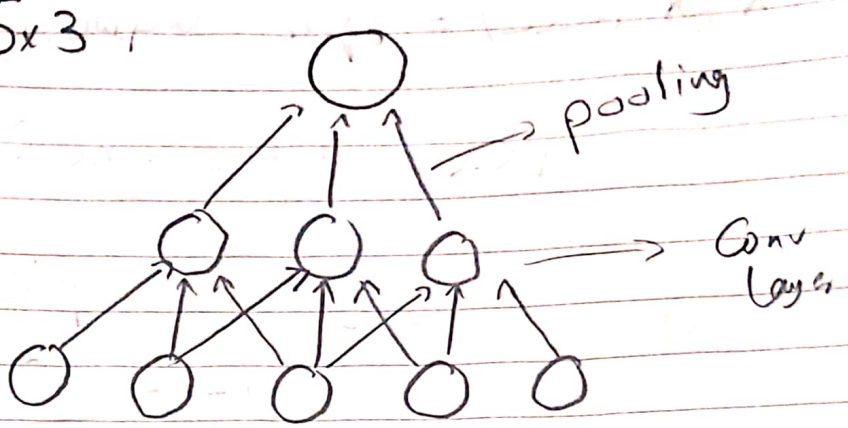
* improves gradients in the network, since activations gradients becomes zero in high value, hence avoids layers death (stopped learning)



* Allows higher learning rate

PAGE
DATE

6) $5 \times 5 \times 3$



7)

output.shape = (128, 1, x, x)

$$x = \frac{128 - 7 + 2*3}{2} + 1 = 64.5$$

↳ Floor → 64

8) inverted drop out scales (during training) the output of activations by inverse of keep probability $q = 1 - p$, so, it's not required to scale network's output at testing or evaluation

9) because it's not shift invariant, accordingly, during training, if the required classes changes it's location in the image, network will produce

different output, preventing the learning algorithm from finding the weights that detects the features of the specific class, regardless to the requirement of large weights size

10) $a[0] = -2*0 + 1*4 = 4$

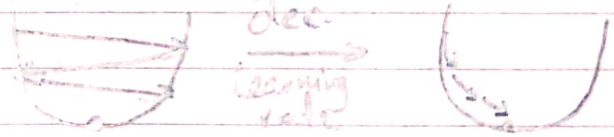
$a[1] = -2*4 + 1*1 = -7$

$a[2] = -2*1 + 1*-1 = -3$

$a[3] = -2*-2 + 1*3 = 7$

$a[4] = -2*3 + 1*0 = -6$

11) learning rate is decreased, since constant error is probably due to overshooting around local minima,



PAGE
DATE

12- Since they are shift invariant, so the same Filter will search For a Feature across the whole image regardless to the Feature location, this also decreases network's size, since the whole image's pixels shares the same Filters (weights)

13- since during each training iteration, some of network's weights are disabled by probability (1-p), but during prediction we can't do the same since we want to get benefit of the whole network, also we can't activate all weights without any change, since this will produce output value different From training's, so we will scale the weights by "p".

PAGE
DATE

14)

* momentum

$$\theta_{t+1} = \theta_t + v_{t+1}$$

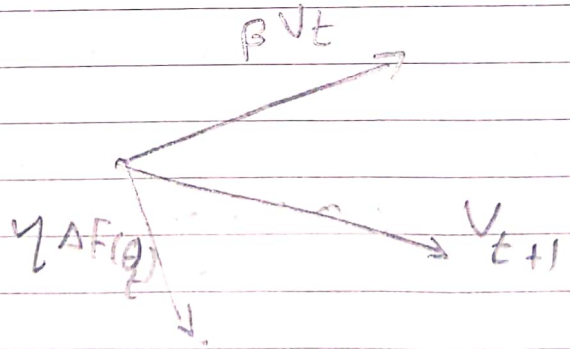
$$v_{t+1} = \beta v_t - \eta \Delta F(\theta_t)$$

* NAG

$$\theta_{t+1} = \theta_t + v_{t+1}$$

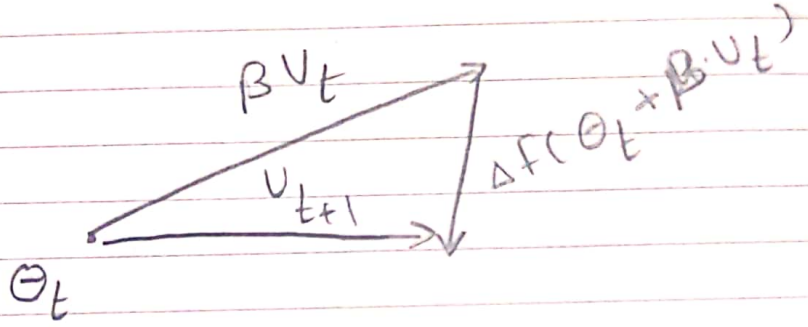
$$v_{t+1} = \beta v_t - \eta \Delta F(\theta_t + \beta v_t)$$

* in momentum gradient is Compute at the point before update



PAGE
DATE

* in NAG the gradient is computed after applying the velocity term to the previous parameter, so if this velocity term is bad (driving model into higher loss), we will be aware of and correct it



15) $G_t = \sum_{i=0}^{n} \Delta f(t) \; \Delta f(t)^T$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{Diag(G_t) + \varepsilon I}} \Delta f$$

* the learning rate decreases over time due to accumulation of squared gradients summation.

Hi . Star