

1. initialize - parameters - deep (layer dims)

return parameters $\{W^{[1]} \dots W^{[L]}$
 $b^{[1]} \dots b^{[L]}$

2. linear - forward (A, W, b)

return Z, cache

$Z = \text{np.dot}(W, A) + b$
cache (A, W, b)

forward
propagation

3. linear - activation - forward (A_prev, w, b, activation)

if activation = "sigmoid":

Z, linear-cache = linear-forward (A_prev, w, b)

A, activation-cache = sigmoid(Z)

Way
of
activation

elif activation = "relu":

=
=

cache (linear-cache, activation-cache)

return A, cache (A_prev, w, b, Z)

4. L - model - forward (X, parameter)

A = X, len(parameters) / 2 → how many layers.

initial

for $1 \sim L-1$ { A, cache = linear-activation-forward (A_prev, w, b, "relu")
caches.append(cache)

L : A_L, cache = linear-activation-forward (A, w, b, "sigmoid")
caches.append(cache)

5. cost function

6. linear-backward (dz, cache)

return dA_prev, dw, db

7. linear-activation-backward (dA, cache, activation)

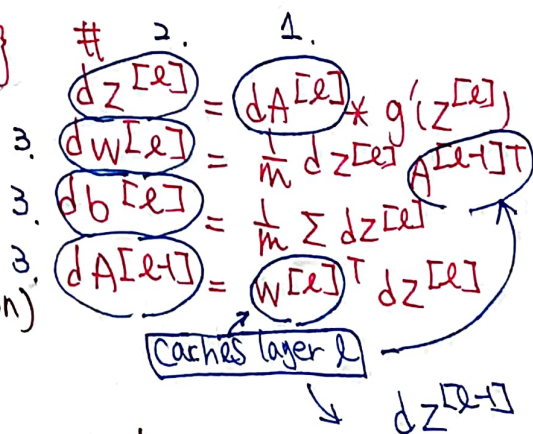
① linear-cache, activation-cache = cache

② if activation = "relu"

= "sigmoid" different way to calculate dz

③ return dA_prev, dw, db

8. L - model - backward (A_L, Y, caches)



L → linear-backward (dz, linear-cache)