deeplearning.ai

Setting up your optimization problem

Gradient Checking

Gradient check for a neural network

This is Done so we can ↓ error from gradients calculated

Take $W^{[1]}, b^{[1]}, \ldots, W^{[L]}, b^{[L]}$ and reshape into a big vector θ.

This is a matrix ⇒ Need to reshape to vector

$J(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]} \ldots W^{[L]}, b^{[L]}) = J(\theta)$

Is Already a vector

↳ Concatenate all of these

Take $dW^{[1]}, db^{[1]}, \ldots, dW^{[L]}, db^{[L]}$ and reshape into a big vector $d\theta$.

Big Question

Is $d\theta$ the gradient of $J(\theta)$?

Gradient checking (Grad check)

$J(\theta) = J(\theta_1, \theta_2, \theta_3 \dots)$

for each i: → i^{th} component of θ [the level of w]

$$d\theta_{approx}[i] = \frac{J(\theta_1, \theta_2 \dots \theta_i + \varepsilon, \dots) - J(\theta_1, \theta_2, \dots \theta_i - \varepsilon, \dots)}{2\varepsilon}$$

} large Δ calculation

Also $d\theta[i] = \frac{\partial J}{\partial \theta_i}$ (By traditional calculus)

What we hope is

$d\theta[i] = d\theta_{approx}[i]$

How to check if they are approx equal?

Take Euclidean dist & normalize by size of vectors

$$Blah = \frac{\| d\theta_{approx} - d\theta \|_2}{\| d\theta_{approx} \|_2 + \| d\theta \|_2}$$

} Euclidean dist

} size of vectors

What is the role of normalizing by length?

If Euclidean dist is large/small, the denominator scales it Accordingly to small/large (Respectively)

If $\varepsilon = 10^{-7}$ Above in the calculation & suppose Blah $= 10^{-7}$, then no error

If Blah $= 10^{-5}$, double check components of θ, none of them should be too large, if large, then possible bug

If Blah $= 10^{-3}$, there is a bug will have to track down which one made it large

Andrew Ng

deeplearning.ai

Setting up your optimization problem

Gradient Checking implementation notes

Gradient checking implementation notes

- Don't use in training – only to debug

→ Cuz of $d\theta_{approx}[i]$ is very slow

- If algorithm fails grad check, look at components to try to identify bug.

look at values of the i^{th} $d\theta_{approx}$ & $d\theta$, say for layer 5, we see large diff
then look at decomposition of θ
θ = vectorized(w) + vectorized(b) [In this case, for layer 5]
So if the large diff is being caused by the "b" component
look into how "b" is calculated

[0·1, 0·1, ·01, ·8, ·9]
w part b part
⇒ b is contributing to the large diff

- Remember regularization.

If you Add $\frac{\lambda}{2m}\sum \|w\|_2$ → Remember to include this when finding $\frac{\partial J}{\partial \theta}$

- Doesn't work with dropout.

→ We're eliminating nodes Randomly
⇒ can't find consistent diff as there is no consistent J
J = Some Random subset of J

- Run at random initialization; perhaps again after some training.

Andrew Ng