deeplearning.ai

# Setting up your optimization problem

---

# Gradient Checking

# Gradient check for a neural network

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

concatenate

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

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

concatenate

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

# Gradient checking (Grad check)

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

For each $i$:

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

$d\theta_{approx}[i] \approx d\theta[i] = \frac{\partial J}{\partial \theta_i}$

$d\theta_{approx} \overset{?}{\approx} d\theta$

Check

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

$\varepsilon = 10^{-7}$

$\approx$ $10^{-7}$ — great! ←

$10^{-5}$

→ $10^{-3}$ — worry. ←

Andrew Ng

deeplearning.ai

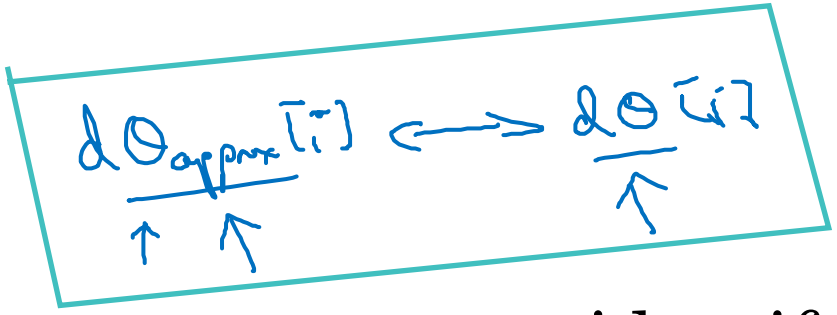
# Setting up your optimization problem

---

# Gradient Checking implementation notes

# Gradient checking implementation notes

- Don't use in training – only to debug




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

$db^{[l]}$ $dW^{[l]}$

- Remember regularization.

$J(\theta) = \frac{1}{m}\sum_i \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\sum_l \|W^{[l]}\|_F^2$

$d\theta$ = grad of $J$ wrt. $\theta$

- Doesn't work with dropout.

$J$ keep-prob = 1.0

Since the cost function J() cannot be well-expressed

Better set keep_prob=1, i.e. turn off dropout, then do gradient checking.

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

$W, b \approx 0$

Andrew Ng