deeplearning.ai

# Setting up your optimization problem

## Gradient Checking

# Gradient check for a neural network

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

concatenate

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

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

concatenate

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 :

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

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

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

Check

i

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

$$\varepsilon = 10^{-7}$$

$\approx$ $10^{-7}$ – 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

$$\boxed{\underline{d\theta_{approx}}[i] \longleftrightarrow \underline{d\theta}[i]}$$

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

$\underline{db^{[l]}} \quad \underline{dW^{[l]}}$

- Remember regularization.

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

$$d\theta = \text{grad of } J \text{ wrt. } \theta$$

- Doesn't work with dropout.

$\underline{J}$ $\quad$ $\underline{\text{keep-prob} = 1.0}$

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.

$\underline{w, b \approx 0}$