Setting up your
optimization problem

Gradient Checking

deeplearning.ai

Gradient check for a neural network

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

Concatenate

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

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

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:

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

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

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

Check

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

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

$$\approx \boxed{10^{-7} \; - \; great!} \leftarrow$$

$$10^{-5}$$

$$\rightarrow 10^{-3} \; - \; worry. \leftarrow$$

Andrew Ng

Gradient checking implementation notes

- Don't use in training – only to debug

$$d\Theta_{approx}[i] \longleftrightarrow \frac{d\Theta[i]}{}$$

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

$$db^{[\ell]} \qquad dW^{[\ell]}$$

- Remember regularization.

$$J(\Theta) = \frac{1}{m} \sum_i \ell(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} \sum_\ell \|W^{[\ell]}\|_F^2$$

$$d\Theta = \text{gradt of } J \text{ wrt. } \Theta$$

- Doesn't work with dropout. J keep-prob $= 1.0$

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

$$W, b \approx 0$$

Andrew Ng