

Plan:

1. Neural networks calculating. Taking look at the math behind NN
 - a. Cost function
 - b. Minimization, backpropagation algorithm
2. Backpropagation on practice. Intuitional understanding of what is forward and backpropagation.
 - a. Gradient checking
 - b. Random initialization
3. Application of neural networks. Some real-world examples of neural network applications.
 - a. Autonomous driving

Questions:

1. What's the difference between forward propagation and backpropagation?
2. Why it's a bad idea to initialize network weights with zeros?
3. When and why are we should use gradient checking?

Glossary:

Backpropagation algorithm - gradient calculating algorithm actively used in neural networks - *To run Backpropagation we need to run forward propagation first*

Gradient checking - numerical gradient calculating algorithm that we use as an extra step to check if our backpropagation implementation is correct - *Don't forget to turn off gradient checking before starting learning your network.*

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log((h_{\Theta}(x^{(i)}))_k) + (1 - y_k^{(i)}) \log(1 - (h_{\Theta}(x^{(i)}))_k) + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (\Theta_{j,i}^{(l)})^2$$

And of course, some memes to compensate the fact that submission is almost a week late.

Me after reading the documentation
for 2 seconds



