**Deep Learning II**

**Tutorial SOLUTIONS**

1. What is the vanishing gradient problem? Why is it becoming the main obstacle when using Back Propagation to train deep neural networks?

**SOLUTIONS**:

Vanishing gradient problem: During each epoch of the neural network training, each weight value in the network receives an update proportional to the gradient of the error function. This has the effect of multiplying *n* of these small numbers to compute gradients of the "front" layers in an *n-layer* neural network, which means the error signal decreases exponentially with n.

Effect in Deep Neural Networks: in DNN the number n is large, so it will be very slow to adjust the weights of front layers.

2. What are the problems that deep learning particular good at solving?

SOLUTIONS: Deep Learning is good at solving pattern recognition problems.

3. Deep Neural Networks are the only architecture to perform deep learning, is it true? Explain your answer.

**SOLUTIONS**: No. There are alternative architectures proposed for deep learning, for instance, Deep Forest and deep SVM (<http://deeplearning.net/wp-content/uploads/2013/03/dlsvm.pdf>).

4. Deep Learning can achieve unsupervised learning tasks, and give one example.

**SOLUTIONS**: Deep Autoencoder

5. Increase the size of a convolutional kernel (in the filtering process) would necessarily increase the performance of a convolutional neural network.

~~A) TRUE~~ B) FALSE

6. Given an n-character English word, we want to predict which character would be the n+1th character in this word. For example, our input is “Edinburg” (which is a 8 character word) and we would like to predict what would be the 8th character.

Which neural network architecture would be suitable to complete this task?

SOLUTION: RNN (e.g. LSTM)

7. Suppose you have 6 convolutional kernels (or filters) of size 3 x 3 and stride 1 in the first layer of a convolutional neural network. You pass an input of dimension 100 x 100 x 3 through this layer (100\*100 means the size of the image, and 3 means RGB channels). What are the dimensions of the data which the next layer will receive?

~~A) 100 x 100 x 3~~

~~B) 33 x 33 x 5~~

C) 98 x 98 x 6

~~D) 300 x 300 x 7~~

TIPS:

1. Look at the number of kernels (filters).
2. Note the filter size 3\*3 is an abbreviation for 3\*3\*3 as the third dimension always equals to the number of channels (RGB) of input: 3. The filter actually sums up all three RGB channels, so the output will be 98\*98\*6, not 98\*98\*3\*6. Check the following links for more details:

<https://cs231n.github.io/convolutional-networks/>

<https://stackoverflow.com/questions/47982594/how-a-convolutional-neural-net-handles-channels>

BTW, this is a popular job interview question.