# 人工智能行业大师访谈

## 吴恩达采访Geoffrey Hinton

## 吴恩达采访Pieter Abbeel

## 吴恩达采访Ian Goodfellow

## 吴恩达采访Yoshua Bengio

## 吴恩达采访林元庆

## 吴恩达采访Andrej Karpathy

## 吴恩达采访Ruslan Salakhutdinov

# 课程1——神经网络与深度学习

## 第一周 深度学习概论

### 1.1 欢迎来到深度学习工程师微专业

### 1.2 什么是神经网络？

### 1.3 用神经网络进行监督学习

### 1.4 为什么深度学习会兴起？

### 1.5 关于这门课

### 1.6 课程资源

Quiz:

### Week 1 Quiz - Introduction to deep learning

1. What does the analogy “AI is the new electricity” refer to?
   *  AI is powering personal devices in our homes and offices, similar to electricity.
   *  Through the “smart grid”, AI is delivering a new wave of electricity.
   *  AI runs on computers and is thus powered by electricity, but it is letting computers do things not possible before.
   *  Similar to electricity starting about 100 years ago, AI is transforming multiple industries.

Note: Andrew illustrated the same idea in the lecture.

1. Which of these are reasons for Deep Learning recently taking off? (Check the two options that apply.)
   *  We have access to a lot more computational power.
   *  Neural Networks are a brand new field.
   *  We have access to a lot more data.
   *  Deep learning has resulted in significant improvements in important applications such as online advertising, speech recognition, and image recognition.
2. Recall this diagram of iterating over different ML ideas. Which of the statements below are true? (Check all that apply.)
   *  Being able to try out ideas quickly allows deep learning engineers to iterate more quickly.
   *  Faster computation can help speed up how long a team takes to iterate to a good idea.
   *  It is faster to train on a big dataset than a small dataset.
   *  Recent progress in deep learning algorithms has allowed us to train good models faster (even without changing the CPU/GPU hardware).

Note: A bigger dataset generally requires more time to train on a same model.

1. When an experienced deep learning engineer works on a new problem, they can usually use insight from previous problems to train a good model on the first try, without needing to iterate multiple times through different models. True/False?
   *  True
   *  False

Note: Maybe some experience may help, but nobody can always find the best model or hyperparameters without iterations.

1. Which one of these plots represents a ReLU activation function?
   * Check [here](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)).
2. Images for cat recognition is an example of “structured” data, because it is represented as a structured array in a computer. True/False?
   *  True
   *  False
3. A demographic dataset with statistics on different cities' population, GDP per capita, economic growth is an example of “unstructured” data because it contains data coming from different sources. True/False?
   *  True
   *  False
4. Why is an RNN (Recurrent Neural Network) used for machine translation, say translating English to French? (Check all that apply.)
   *  It can be trained as a supervised learning problem.
   *  It is strictly more powerful than a Convolutional Neural Network (CNN).
   *  It is applicable when the input/output is a sequence (e.g., a sequence of words).
   *  RNNs represent the recurrent process of Idea->Code->Experiment->Idea->....
5. In this diagram which we hand-drew in lecture, what do the horizontal axis (x-axis) and vertical axis (y-axis) represent?
   * x-axis is the amount of data
   * y-axis (vertical axis) is the performance of the algorithm.
6. Assuming the trends described in the previous question's figure are accurate (and hoping you got the axis labels right), which of the following are true? (Check all that apply.)
   *  Increasing the training set size generally does not hurt an algorithm’s performance, and it may help significantly.
   *  Increasing the size of a neural network generally does not hurt an algorithm’s performance, and it may help significantly.
   *  Decreasing the training set size generally does not hurt an algorithm’s performance, and it may help significantly.
   *  Decreasing the size of a neural network generally does not hurt an algorithm’s performance, and it may help significantly.

## 第二周 神经网络基础

### 2.1 二分分类

### 2.2 logistic 回归

### 2.3 logistic 回归损失函数

### 2.4 梯度下降法

### 2.5 导数

### 2.6 更多导数的例子

### 2.7 计算图

### 2.8 计算图的导数计算

### 2.9 logistic 回归中的梯度下降法

### 2.10 m个样本的梯度下降

### 2.11 向量化

### 2.12 向量化的更多例子

### 2.13 向量化 logistic 回归

### 2.14 向量化 logistic 回归的梯度输出

### 2.15 Python 中的广播

### 2.16 关于 python / numpy 向量的说明

### 2.17 Jupyter / Ipython 笔记本的快速指南

### 2.18 （选修）logistic 损失函数的解释

### Week 2 Quiz - Neural Network Basics

1. What does a neuron compute?
   *  A neuron computes an activation function followed by a linear function (z = Wx + b)
   *  A neuron computes a linear function (z = Wx + b) followed by an activation function
   *  A neuron computes a function g that scales the input x linearly (Wx + b)
   *  A neuron computes the mean of all features before applying the output to an activation function

Note: The output of a neuron is a = g(Wx + b) where g is the activation function (sigmoid, tanh, ReLU, ...).

1. Which of these is the "Logistic Loss"?

L(i)(y^(i),y(i))=y(i)log(y^(i))+(1−y(i))log(1−y^(i))

* + Check [here](https://en.wikipedia.org/wiki/Cross_entropy#Cross-entropy_error_function_and_logistic_regression).

Note: We are using a cross-entropy loss function.

1. Suppose img is a (32,32,3) array, representing a 32x32 image with 3 color channels red, green and blue. How do you reshape this into a column vector?
   * x = img.reshape((32 \* 32 \* 3, 1))
2. Consider the two following random arrays "a" and "b":

a = np.random.randn(2, 3) # a.shape = (2, 3)

b = np.random.randn(2, 1) # b.shape = (2, 1)

c = a + b

What will be the shape of "c"?

b (column vector) is copied 3 times so that it can be summed to each column of a. Therefore, c.shape = (2, 3).

1. Consider the two following random arrays "a" and "b":

a = np.random.randn(4, 3) # a.shape = (4, 3)

b = np.random.randn(3, 2) # b.shape = (3, 2)

c = a \* b

What will be the shape of "c"?

"\*" operator indicates element-wise multiplication. Element-wise multiplication requires same dimension between two matrices. It's going to be an error.

1. Suppose you have n\_x input features per example. Recall that X=[x^(1), x^(2)...x^(m)]. What is the dimension of X?

(n\_x, m)

Note: A stupid way to validate this is use the formula Z^(l) = W^(l)A^(l) when l = 1, then we have

* + A^(1) = X
  + X.shape = (n\_x, m)
  + Z^(1).shape = (n^(1), m)
  + W^(1).shape = (n^(1), n\_x)

1. Recall that np.dot(a,b) performs a matrix multiplication on a and b, whereas a\*b performs an element-wise multiplication.

Consider the two following random arrays "a" and "b":

a = np.random.randn(12288, 150) # a.shape = (12288, 150)

b = np.random.randn(150, 45) # b.shape = (150, 45)

c = np.dot(a, b)

What is the shape of c?

c.shape = (12288, 45), this is a simple matrix multiplication example.

1. Consider the following code snippet:

# a.shape = (3,4)

# b.shape = (4,1)

for i in range(3):

for j in range(4):

c[i][j] = a[i][j] + b[j]

How do you vectorize this?

c = a + b.T

1. Consider the following code:

a = np.random.randn(3, 3)

b = np.random.randn(3, 1)

c = a \* b

What will be c?

This will invoke broadcasting, so b is copied three times to become (3,3), and ∗ is an element-wise product so c.shape = (3, 3).

1. Consider the following computation graph.

What is the output J?

J = u + v - w

= a \* b + a \* c - (b + c)

= a \* (b + c) - (b + c)

= (a - 1) \* (b + c)

Answer: (a - 1) \* (b + c)

## 第三周 浅层神经网络

### 3.1 神经网络概览

### 3.2 神经网络表示

### 3.3 计算神经网络的输出

### 3.4 向量化示例

### 3.5 向量化实现的解释

### 3.6 激活函数

### 3.7 为什么需要非线性激活函数？

### 3.8 激活函数的导数

### 3.9 神经网络的梯度下降法

### 3.10 （选修）直观理解反向传播

### 3.11 随机初始化

### Week 3 Quiz - Shallow Neural Networks

1. Which of the following are true? (Check all that apply.) **Notice that I only list correct options.**
   * X is a matrix in which each column is one training example.
   * a^[2]\_4 is the activation output by the 4th neuron of the 2nd layer
   * a^[2](12) denotes the activation vector of the 2nd layer for the 12th training example.
   * a^[2] denotes the activation vector of the 2nd layer.

Note: If you are not familiar with the notation used in this course, check [here](https://www.coursera.org/learn/neural-networks-deep-learning/resources/YsZjP).

1. The tanh activation usually works better than sigmoid activation function for hidden units because the mean of its output is closer to zero, and so it centers the data better for the next layer. True/False?
   *  True
   *  False

Note: You can check [this post](https://stats.stackexchange.com/a/101563/169377) and (this paper)[<http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf>].

As seen in lecture the output of the tanh is between -1 and 1, it thus centers the data which makes the learning simpler for the next layer.

1. Which of these is a correct vectorized implementation of forward propagation for layer l, where 1≤l≤L?
   * Z^[l]=W^[l]A^[l−1]+b^[l]
   * A^[l]=g^[l](Z^[l])
2. You are building a binary classifier for recognizing cucumbers (y=1) vs. watermelons (y=0). Which one of these activation functions would you recommend using for the output layer?
   *  ReLU
   *  Leaky ReLU
   *  sigmoid
   *  tanh

Note: The output value from a sigmoid function can be easily understood as a probability.

Sigmoid outputs a value between 0 and 1 which makes it a very good choice for binary classification. You can classify as 0 if the output is less than 0.5 and classify as 1 if the output is more than 0.5. It can be done with tanh as well but it is less convenient as the output is between -1 and 1.

1. Consider the following code:

A = np.random.randn(4,3)

B = np.sum(A, axis = 1, keepdims = True)

What will be B.shape?

B.shape = (4, 1)

we use (keepdims = True) to make sure that A.shape is (4,1) and not (4, ). It makes our code more rigorous.

1. Suppose you have built a neural network. You decide to initialize the weights and biases to be zero. Which of the following statements are True? (Check all that apply)
   *  Each neuron in the first hidden layer will perform the same computation. So even after multiple iterations of gradient descent each neuron in the layer will be computing the same thing as other neurons.
   *  Each neuron in the first hidden layer will perform the same computation in the first iteration. But after one iteration of gradient descent they will learn to compute different things because we have “broken symmetry”.
   *  Each neuron in the first hidden layer will compute the same thing, but neurons in different layers will compute different things, thus we have accomplished “symmetry breaking” as described in lecture.
   *  The first hidden layer’s neurons will perform different computations from each other even in the first iteration; their parameters will thus keep evolving in their own way.
2. Logistic regression’s weights w should be initialized randomly rather than to all zeros, because if you initialize to all zeros, then logistic regression will fail to learn a useful decision boundary because it will fail to “break symmetry”, True/False?
   *  True
   *  False

Logistic Regression doesn't have a hidden layer. If you initialize the weights to zeros, the first example x fed in the logistic regression will output zero but the derivatives of the Logistic Regression depend on the input x (because there's no hidden layer) which is not zero. So at the second iteration, the weights values follow x's distribution and are different from each other if x is not a constant vector.

1. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large values, using np.random.randn(..,..)\*1000. What will happen?
   *  It doesn’t matter. So long as you initialize the weights randomly gradient descent is not affected by whether the weights are large or small.
   *  This will cause the inputs of the tanh to also be very large, thus causing gradients to also become large. You therefore have to set α to be very small to prevent divergence; this will slow down learning.
   *  This will cause the inputs of the tanh to also be very large, causing the units to be “highly activated” and thus speed up learning compared to if the weights had to start from small values.
   *  This will cause the inputs of the tanh to also be very large, thus causing gradients to be close to zero. The optimization algorithm will thus become slow.

tanh becomes flat for large values, this leads its gradient to be close to zero. This slows down the optimization algorithm.

1. Consider the following 1 hidden layer neural network:
   * b[1] will have shape (4, 1)
   * W[1] will have shape (4, 2)
   * W[2] will have shape (1, 4)
   * b[2] will have shape (1, 1)

Note: Check [here](https://user-images.githubusercontent.com/14886380/29200515-7fdd1548-7e88-11e7-9d05-0878fe96bcfa.png) for general formulas to do this.

1. In the same network as the previous question, what are the dimensions of Z^[1] and A^[1]?
   * Z[1] and A[1] are (4,m)

Note: Check [here](https://user-images.githubusercontent.com/14886380/29200515-7fdd1548-7e88-11e7-9d05-0878fe96bcfa.png) for general formulas to do this.

## 第四周 深层神经网络

### 4.1 深层神经网络

### 4.2 深层网络中的前向传播

### 4.3 核对矩阵的维数

### 4.4 为什么使用深层表示？

### 4.5 深层神经网络的搭建

### 4.6 前向和反向传播

### 4.7 参数 VS 超参数

### 4.8 这和大脑有什么关系？

### Week 4 Quiz - Key concepts on Deep Neural Networks

1. What is the "cache" used for in our implementation of forward propagation and backward propagation?
   *  It is used to cache the intermediate values of the cost function during training.
   *  We use it to pass variables computed during forward propagation to the corresponding backward propagation step. It contains useful values for backward propagation to compute derivatives.
   *  It is used to keep track of the hyperparameters that we are searching over, to speed up computation.
   *  We use it to pass variables computed during backward propagation to the corresponding forward propagation step. It contains useful values for forward propagation to compute activations.

the "cache" records values from the forward propagation units and sends it to the backward propagation units because it is needed to compute the chain rule derivatives.

1. Among the following, which ones are "hyperparameters"? (Check all that apply.) **I only list correct options.**
   * size of the hidden layers n[l]
   * learning rate α
   * number of iterations
   * number of layers L in the neural network

Note: You can check [this Quora post](https://www.quora.com/What-are-hyperparameters-in-machine-learning) or [this blog post](http://colinraffel.com/wiki/neural_network_hyperparameters).

1. Which of the following statements is true?
   *  The deeper layers of a neural network are typically computing more complex features of the input than the earlier layers. Correct
   *  The earlier layers of a neural network are typically computing more complex features of the input than the deeper layers.

Note: You can check the lecture videos. I think Andrew used a CNN example to explain this.

1. Vectorization allows you to compute forward propagation in an L-layer neural network without an explicit for-loop (or any other explicit iterative loop) over the layers l=1, 2, …,L. True/False?
   *  True
   *  False

Note: We cannot avoid the for-loop iteration over the computations among layers.

1. Assume we store the values for n^[l] in an array called layers, as follows: layer\_dims = [n\_x, 4,3,2,1]. So layer 1 has four hidden units, layer 2 has 3 hidden units and so on. Which of the following for-loops will allow you to initialize the parameters for the model?

for(i in range(1, len(layer\_dims))):

parameter[‘W’ + str(i)] = np.random.randn(layers[i], layers[i - 1])) \* 0.01

parameter[‘b’ + str(i)] = np.random.randn(layers[i], 1) \* 0.01

1. Consider the following neural network.
   * The number of layers L is 4. The number of hidden layers is 3.

Note: The input layer (L^[0]) does not count.

As seen in lecture, the number of layers is counted as the number of hidden layers + 1. The input and output layers are not counted as hidden layers.

1. During forward propagation, in the forward function for a layer l you need to know what is the activation function in a layer (Sigmoid, tanh, ReLU, etc.). During backpropagation, the corresponding backward function also needs to know what is the activation function for layer l, since the gradient depends on it. True/False?
   *  True
   *  False

During backpropagation you need to know which activation was used in the forward propagation to be able to compute the correct derivative.

1. There are certain functions with the following properties:

(i) To compute the function using a shallow network circuit, you will need a large network (where we measure size by the number of logic gates in the network), but (ii) To compute it using a deep network circuit, you need only an exponentially smaller network. True/False?

* +  True
  +  False

Note: See lectures, exactly same idea was explained.

1. Consider the following 2 hidden layer neural network:

Which of the following statements are True? (Check all that apply).

* + W^[1] will have shape (4, 4)
  + b^[1] will have shape (4, 1)
  + W^[2] will have shape (3, 4)
  + b^[2] will have shape (3, 1)
  + b^[3] will have shape (1, 1)
  + W^[3] will have shape (1, 3)

Note: See [this image](https://user-images.githubusercontent.com/14886380/29200515-7fdd1548-7e88-11e7-9d05-0878fe96bcfa.png) for general formulas.

1. Whereas the previous question used a specific network, in the general case what is the dimension of W^[l], the weight matrix associated with layer l?
   * W^[l] has shape (n^[l],n^[l−1])

Note: See [this image](https://user-images.githubusercontent.com/14886380/29200515-7fdd1548-7e88-11e7-9d05-0878fe96bcfa.png) for general formulas.

# 课程2——改善深层神经网络：超参数调试、正则化以及优化

## 第一周 深度学习的实用层面

### 1.1 训练 / 开发 / 测试集

### 1.2 偏差 / 方差

### 1.3 机器学习基础

### 1.4 正则化

### 1.5 为什么正则化可以减少过拟合？

### 1.6 Dropout 正则化

### 1.7 理解 Dropout

### 1.8 其他正则化方法

### 1.9 正则化输入

### 1.10 梯度消失与梯度爆炸

### 1.11 深度网络的权重初始化

### 1.12 梯度的数值逼近

### 1.13 梯度检验

### 1.14 关于梯度检验实现的注记

### Week 1 Quiz - Practical aspects of deep learning

1. If you have 10,000,000 examples, how would you split the train/dev/test set?
   * 98% train . 1% dev . 1% test
2. The dev and test set should:
   * Come from the same distribution
3. If your Neural Network model seems to have high variance, what of the following would be promising things to try?
   * Add regularization
   * Get more training data

Note: Check [here](https://user-images.githubusercontent.com/14886380/29240263-f7c517ca-7f93-11e7-8549-58856e0ed12f.png).

1. You are working on an automated check-out kiosk for a supermarket, and are building a classifier for apples, bananas and oranges. Suppose your classifier obtains a training set error of 0.5%, and a dev set error of 7%. Which of the following are promising things to try to improve your classifier? (Check all that apply.)
   * Increase the regularization parameter lambda
   * Get more training data

Note: Check [here](https://user-images.githubusercontent.com/14886380/29240263-f7c517ca-7f93-11e7-8549-58856e0ed12f.png).

1. What is weight decay?
   * A regularization technique (such as L2 regularization) that results in gradient descent shrinking the weights on every iteration.
2. What happens when you increase the regularization hyperparameter lambda?
   * Weights are pushed toward becoming smaller (closer to 0)
3. With the inverted dropout technique, at test time:
   * You do not apply dropout (do not randomly eliminate units) and do not keep the 1/keep\_prob factor in the calculations used in training
4. Increasing the parameter keep\_prob from (say) 0.5 to 0.6 will likely cause the following: (Check the two that apply)
   * Reducing the regularization effect
   * Causing the neural network to end up with a lower training set error
5. Which of these techniques are useful for reducing variance (reducing overfitting)? (Check all that apply.)
   * Dropout
   * L2 regularization
   * Data augmentation
6. Why do we normalize the inputs x?
   * It makes the cost function faster to optimize

## 第二周 优化算法

### 2.1 Mini-batch 梯度下降法

### 2.2 理解 mini-batch 梯度下降法

### 2.3 指数加权平均

### 2.4 理解指数加权平均

### 2.5 指数加权平均的偏差修正

### 2.6 动量梯度下降法

### 2.7 RMSprop

### 2.8 Adam 优化算法

### 2.9 学习率衰减

### 2.10 局部最优的问题

### Week 2 Quiz - Optimization algorithms

1. Which notation would you use to denote the 3rd layer’s activations when the input is the 7th example from the 8th minibatch?
   * a^[3]{8}(7)

Note: **[i]{j}(k)** superscript means **i-th layer**, **j-th minibatch**, **k-th example**

1. Which of these statements about mini-batch gradient descent do you agree with?
   *  You should implement mini-batch gradient descent without an explicit for-loop over different mini-batches, so that the algorithm processes all mini-batches at the same time (vectorization).
   *  Training one epoch (one pass through the training set) using mini-batch gradient descent is faster than training one epoch using batch gradient descent.
   *  One iteration of mini-batch gradient descent (computing on a single mini-batch) is faster than one iteration of batch gradient descent.

Note: Vectorization is not for computing several mini-batches in the same time.

1. Why is the best mini-batch size usually not 1 and not m, but instead something in-between?
   * If the mini-batch size is 1, you lose the benefits of vectorization across examples in the mini-batch.
   * If the mini-batch size is m, you end up with batch gradient descent, which has to process the whole training set before making progress.
2. Suppose your learning algorithm’s cost ***J***, plotted as a function of the number of iterations, looks like this:
   * If you’re using mini-batch gradient descent, this looks acceptable. But if you’re using batch gradient descent, something is wrong.

Note: There will be some oscillations when you're using mini-batch gradient descent since there could be some noisy data example in batches. However batch gradient descent always guarantees a lower ***J*** before reaching the optimal.

1. Suppose the temperature in Casablanca over the first three days of January are the same:

Jan 1st: θ\_1 = 10

Jan 2nd: θ\_2 = 10

Say you use an exponentially weighted average with β = 0.5 to track the temperature: v\_0 = 0, v\_t = βv\_t−1 + (1 − β)θ\_t. If v\_2 is the value computed after day 2 without bias correction, and v^corrected\_2 is the value you compute with bias correction. What are these values?

* + v\_2 = 7.5, v^corrected\_2 = 10

1. Which of these is NOT a good learning rate decay scheme? Here, t is the epoch number.
   * α = e^t \* α\_0

Note: This will explode the learning rate rather than decay it.

1. You use an exponentially weighted average on the London temperature dataset. You use the following to track the temperature: v\_t = βv\_t−1 + (1 − β)θ\_t. The red line below was computed using β = 0.9. What would happen to your red curve as you vary β? (Check the two that apply)
   * Increasing β will shift the red line slightly to the right.
   * Decreasing β will create more oscillation within the red line.
2. Consider this figure:

These plots were generated with gradient descent; with gradient descent with momentum (β = 0.5) and gradient descent with momentum (β = 0.9). Which curve corresponds to which algorithm?

(1) is gradient descent. (2) is gradient descent with momentum (small β). (3) is gradient descent with momentum (large β)

1. Suppose batch gradient descent in a deep network is taking excessively long to find a value of the parameters that achieves a small value for the cost function J(W[1],b[1],...,W[L],b[L]). Which of the following techniques could help find parameter values that attain a small value forJ? (Check all that apply)
   *  Try using Adam
   *  Try better random initialization for the weights
   *  Try tuning the learning rate α
   *  Try mini-batch gradient descent
   *  Try initializing all the weights to zero
2. Which of the following statements about Adam is False?
   * Adam should be used with batch gradient computations, not with mini-batches.

Note: Adam could be used with both.

## 第三周 超参数调试、Batch 正则化和程序框架

### 3.1 调试处理

### 3.2 为超参数选择合适的范围

### 3.3 超参数训练的实践：Pandas VS Caviar

### 3.4 正则化网络的激活函数

### 3.5 将 Batch Norm 拟合进神经网络

### 3.6 Batch Norm 为什么奏效？

### 3.7 测试时的 Batch Norm

### 3.8 Softmax 回归

### 3.9 训练一个 Softmax 分类器

### 3.10 深度学习框架

### 3.11 TensorFlow

### Week 3 Quiz - Hyperparameter tuning, Batch Normalization, Programming Frameworks

1. If searching among a large number of hyperparameters, you should try values in a grid rather than random values, so that you can carry out the search more systematically and not rely on chance. True or False?
   *  False
   *  True

Note: Try random values, don't do grid search. Because you don't know which hyperparameters are more important than others.

And to take an extreme example, let's say that hyperparameter two was that value epsilon that you have in the denominator of the Adam algorithm. So your choice of alpha matters a lot and your choice of epsilon hardly matters.

1. Every hyperparameter, if set poorly, can have a huge negative impact on training, and so all hyperparameters are about equally important to tune well. True or False?
   *  False
   *  True

We've seen in lecture that some hyperparameters, such as the learning rate, are more critical than others.

1. During hyperparameter search, whether you try to babysit one model (“Panda” strategy) or train a lot of models in parallel (“Caviar”) is largely determined by:
   *  Whether you use batch or mini-batch optimization
   *  The presence of local minima (and saddle points) in your neural network
   *  The amount of computational power you can access
   *  The number of hyperparameters you have to tune
2. If you think β (hyperparameter for momentum) is between on 0.9 and 0.99, which of the following is the recommended way to sample a value for beta?

r = np.random.rand()

beta = 1 - 10 \*\* (-r - 1)

1. Finding good hyperparameter values is very time-consuming. So typically you should do it once at the start of the project, and try to find very good hyperparameters so that you don’t ever have to revisit tuning them again. True or false?
   *  False
   *  True

Note: Minor changes in your model could potentially need you to find good hyperparameters again from scratch.

1. In batch normalization as presented in the videos, if you apply it on the lth layer of your neural network, what are you normalizing?
   * z^[l]
2. In the normalization formula, why do we use epsilon?
   * To avoid division by zero
3. Which of the following statements about γ and β in Batch Norm are true? **Only correct options listed**
   * They can be learned using Adam, Gradient descent with momentum, or RMSprop, not just with gradient descent.
   * They set the mean and variance of the linear variable z^[l] of a given layer.
4. After training a neural network with Batch Norm, at test time, to evaluate the neural network on a new example you should:
   * Perform the needed normalizations, use μ and σ^2 estimated using an exponentially weighted average across mini-batches seen during training.
5. Which of these statements about deep learning programming frameworks are true? (Check all that apply)
   *  A programming framework allows you to code up deep learning algorithms with typically fewer lines of code than a lower-level language such as Python.
   *  Even if a project is currently open source, good governance of the project helps ensure that the it remains open even in the long term, rather than become closed or modified to benefit only one company.
   *  Deep learning programming frameworks require cloud-based machines to run.

# 课程3——结构化机器学习项目

## 第一周 机器学习（ML）策略（1）

### 1.1 为什么是 ML 策略

### 1.2 正交化

### 1.3 单一数字评估指标

### 1.4 满足和优化指标

### 1.5 训练 / 开发 / 测试集划分

### 1.6 开发集合测试集的大小

### 1.7 什么时候该改变开发 / 测试集和指标

### 1.8 为什么是人的表现

### 1.9 可避免偏差

### 1.10 理解人的表现

### 1.11 超过人的表现

### 1.12 改善模型表现

## 第二周 机器学习（ML）策略（2）

### 2.1 进行误差分析

### 2.2 清楚标注错误的数据

### 2.3 快速搭建你的第一个系统，并进行迭代

### 2.4 在不同的划分上进行训练并测试

### 2.5 不匹配数据划分的偏差和方差

### 2.6 定位数据不匹配

### 2.7 迁移学习

### 2.8 多任务学习

### 2.9 什么是端到端的深度学习

### 2.10 是否要使用端到端的深度学习

# 课程4——卷积神经网络

## 第一周 卷积神经网络

### 1.1 计算机视觉

### 1.2 边缘检测示例

### 1.3 更多边缘检测内容

### 1.4 Padding

### 1.5 卷积步长

### 1.6 卷积为何有效

### 1.7 单层卷积网络

### 1.8 简单卷积网络示例

### 1.9 池化层

### 1.10 卷积神经网络示例

### 1.11 为什么使用卷积？

### Quiz 1 - The basics of ConvNets

1. What do you think applying this filter to a grayscale image will do?

[[0 1 -1 0][ 1 3 -3 -1][ 1 3 -3 -1][ 0 1 -1 0]]

Ans: Detect vertical edges

1. Suppose your input is a 300 by 300 color (RGB) image, and you are not using a convolutional network. If the first hidden layer has 100 neurons, each one fully connected to the input, how many parameters does this hidden layer have (including the bias parameters)?

Ans: 27,000,100

1. Suppose your input is a 300 by 300 color (RGB) image, and you use a convolutional layer with 100 filters that are each 5x5. How many parameters does this hidden layer have (including the bias parameters)?

Ans: 2600

1. You have an input volume that is 63x63x16, and convolve it with 32 filters that are each 7x7, using a stride of 2 and no padding. What is the output volume?

Ans: 29x29x32

1. You have an input volume that is 15x15x8, and pad it using “pad=2.” What is the dimension of the resulting volume (after padding)?

Ans: 19x19x8

1. You have an input volume that is 63x63x16, and convolve it with 32 filters that are each 7x7, and stride of 1. You want to use a “same” convolution. What is the padding?

Ans: 3

1. You have an input volume that is 32x32x16, and apply max pooling with a stride of 2 and a filter size of 2. What is the output volume?

Ans: 16x16x16

1. Because pooling layers do not have parameters, they do not affect the backpropagation (derivatives) calculation.

Ans: False

1. In lecture we talked about “parameter sharing” as a benefit of using convolutional networks. Which of the following statements about parameter sharing in ConvNets are true? (Check all that apply.)

It reduces the total number of parameters, thus reducing overfitting.

It allows a feature detector to be used in multiple locations throughout the whole input image/input volume.

1. In lecture we talked about “sparsity of connections” as a benefit of using convolutional layers. What does this mean?

Each activation in the next layer depends on only a small number of activations from the previous layer.

## 第二周 深度卷积网络：实例探究

### 2.1 为什么要进行实例探究？

### 2.2 经典网络

### 2.3 残差网络

### 2.4 残差网络为什么有用？

### 2.5 网络中的网络以及 1×1 卷积

### 2.6 谷歌 Inception 网络简介

### 2.7 Inception 网络

### 2.8 使用开源的实现方案

### 2.9 迁移学习

### 2.10 数据扩充

### 2.11 计算机视觉现状

### Quiz 2 - **Deep-convolutional-models**

1. Which of the following do you typically see as you move to deeper layers in a ConvNet?

Ans: nH and nW decrease, while nC increases

1. Which of the following do you typically see in a ConvNet? (Check all that apply.)

Ans:

Multiple CONV layers followed by a POOL layer

FC layers in the last few layers

1. In order to be able to build very deep networks, we usually only use pooling layers to downsize the height/width of the activation volumes while convolutions are used with “valid” padding. Otherwise, we would downsize the input of the model too quickly.

Ans: False

1. Training a deeper network (for example, adding additional layers to the network) allows the network to fit more complex functions and thus almost always results in lower training error. For this question, assume we’re referring to “plain” networks.

Ans: False

1. The following equation captures the computation in a ResNet block. What goes into the two blanks above?

a[l+2]=g(W[l+2]g(W[l+1]a[l]+b[l+1])+b[l+2]+\_\_\_\_\_\_\_ )+\_\_\_\_\_\_\_

Ans: a[l] and 0, respectively

1. Which ones of the following statements on Residual Networks are true? (Check all that apply.)

Ans:

Using a skip-connection helps the gradient to backpropagate and thus helps you to train deeper networks

The skip-connection makes it easy for the network to learn an identity mapping between the input and the output within the ResNet block.

1. Suppose you have an input volume of dimension 64x64x16. How many parameters would a single 1x1 convolutional filter have (including the bias)?

Ans: 17

1. Suppose you have an input volume of dimension nH x nW x nC. Which of the following statements do you agree with? (Assume that “1x1 convolutional layer” below always uses a stride of 1 and no padding.)

Ans:

You can use a 1x1 convolutional layer to reduce nC but not nH, nW.

You can use a pooling layer to reduce nH, nW, but not nC.

1. Which ones of the following statements on Inception Networks are true? (Check all that apply.)

Ans:

Inception blocks usually use 1x1 convolutions to reduce the input data volume’s size before applying 3x3 and 5x5 convolutions.

A single inception block allows the network to use a combination of 1x1, 3x3, 5x5 convolutions and pooling.

1. Which of the following are common reasons for using open-source implementations of ConvNets (both the model and/or weights)? Check all that apply.

Ans:

Parameters trained for one computer vision task are often useful as pretraining for other computer vision tasks.

It is a convenient way to get working an implementation of a complex ConvNet architecture.

## 第三周 目标检测

### 3.1 目标定位

### 3.2 特征点检测

### 3.3 目标检测

### 3.4 卷积的滑动窗口实现

### 3.5 Bounding Box 预测

### 3.6 交并比

### 3.7 非极大值抑制

### 3.8 Anchor Boxes

### 3.9 YOLO 算法

### 3.10 （选修）RPN 网络

### Quiz Week 3 - Detection algorithms

1. You are building a 3-class object classification and localization algorithm. The classes are: pedestrian (c=1), car (c=2), motorcycle (c=3). What would be the label for the following image? Recall y=[pc,bx,by,bh,bw,c1,c2,c3]

Ans: y=[1,0.3,0.7,0.3,0.3,0,1,0]

1. Continuing from the previous problem, what should y be for the image below? Remember that “?” means “don’t care”, which means that the neural network loss function won’t care what the neural network gives for that component of the output. As before, y=[pc,bx,by,bh,bw,c1,c2,c3].

Ans: y=[0,?,?,?,?,?,?,?]

1. You are working on a factory automation task. Your system will see a can of soft-drink coming down a conveyor belt, and you want it to take a picture and decide whether (i) there is a soft-drink can in the image, and if so (ii) its bounding box. Since the soft-drink can is round, the bounding box is always square, and the soft drink can always appears as the same size in the image. There is at most one soft drink can in each image. Here’re some typical images in your training set:

What is the most appropriate set of output units for your neural network?

Ans: Logistic unit, bx, by, bh (since bw = bh)

1. If you build a neural network that inputs a picture of a person’s face and outputs N landmarks on the face (assume the input image always contains exactly one face), how many output units will the network have?

Ans: 2N

1. When training one of the object detection systems described in lecture, you need a training set that contains many pictures of the object(s) you wish to detect. However, bounding boxes do not need to be provided in the training set, since the algorithm can learn to detect the objects by itself.

Ans: True

1. Suppose you are applying a sliding windows classifier (non-convolutional implementation). Increasing the stride would tend to increase accuracy, but decrease computational cost.

Ans: False

1. In the YOLO algorithm, at training time, only one cell ---the one containing the center/midpoint of an object--- is responsible for detecting this object.

Ans: True

1. What is the IoU between these two boxes? The upper-left box is 2x2, and the lower-right box is 2x3. The overlapping region is 1x1.

Ans: 1/9

1. Suppose you run non-max suppression on the predicted boxes above. The parameters you use for non-max suppression are that boxes with probability ≤ 0.4 are discarded, and the IoU threshold for deciding if two boxes overlap is 0.5. How many boxes will remain after non-max suppression?

Ans: 4

1. Suppose you are using YOLO on a 19x19 grid, on a detection problem with 20 classes, and with 5 anchor boxes. During training, for each image you will need to construct an output volume y as the target value for the neural network; this corresponds to the last layer of the neural network. (y may include some “?”, or “don’t cares”). What is the dimension of this output volume?

Ans: 19x19x(5x25)

## 第四周 特殊应用：人脸识别和神经风格转换

### 4.1 什么是人脸识别？

### 4.2 One-Shot 学习

### 4.3 Siamese 网络

### 4.4 Triplet 损失

### 4.5 面部验证与二分类

### 4.6 什么是神经风格转换？

### 4.7 深度卷积网络在学什么？

### 4.8 代价函数

### 4.9 内容代价函数

### 4.10 风格损失函数

### 4.11 一维到三维推广

### Quiz Week 4 – Face Recognition & Neural Style Transfer

1. Face verification requires comparing a new picture against one person’s face, whereas face recognition requires comparing a new picture against K person’s faces.

Ans: True

1. Why do we learn a function d(img1,img2) for face verification? (Select all that apply.)

# Ans:

- We need to solve a one-shot learning problem.

- This allows us to learn to recognize a new person given just a single image of that person.

1. In order to train the parameters of a face recognition system, it would be reasonable to use a training set comprising 100,000 pictures of 100,000 different persons.

Ans: False

1. Which of the following is a correct definition of the triplet loss? Consider that ```a>0```. (We encourage you to figure out the answer from first principles, rather than just refer to the lecture.)

Ans: max(||f(A)-f(P)||^2-||f(A)-f(N)||^2+a,0)

1. Consider the following Siamese network architecture:

The upper and lower neural networks have different input images, but have exactly the same parameters.

Ans: True

1. You train a ConvNet on a dataset with 100 different classes. You wonder if you can find a hidden unit which responds strongly to pictures of cats. (I.e., a neuron so that, of all the input/training images that strongly activate that neuron, the majority are cat pictures.) You are more likely to find this unit in layer 4 of the network than in layer 1.

Ans: True

1. Neural style transfer is trained as a supervised learning task in which the goal is to input two images (x), and train a network to output a new, synthesized image (y).

Ans: False

1. In the deeper layers of a ConvNet, each channel corresponds to a different feature detector. The style matrix G^{[l]} measures the degree to which the activations of different feature detectors in layer l vary (or correlate) together with each other.

Ans: True

1. In neural style transfer, what is updated in each iteration of the optimization algorithm?

Ans: The pixel values of the generated image G

1. You are working with 3D data. You are building a network layer whose input volume has size 32x32x32x16 (this volume has 16 channels), and applies convolutions with 32 filters of dimension 3x3x3 (no padding, stride 1). What is the resulting output volume?

Ans: 30x30x30x32

# 课程5——序列模型

## 第一周 循环序列模型

### 1.1 为什么选择序列模型

### 1.2 数学符号

### 1.3 循环神经网络模型

### 1.4 通过时间的反向传播

### 1.5 不同类型的循环神经网络

### 1.6 语言模型和序列生成

### 1.7 对新序列采样

### 1.8 带有神经网络的梯度消失

### 1.9 GRU 单元

### 1.10 长短期记忆（LSTM）

### 1.11 双向神经网络

### 1.12 深层循环神经网络

## 第二周 自然语言处理与词嵌入

### 2.1 词汇表征

### 2.2 使用词嵌入

### 2.3 词嵌入的特性

### 2.4 嵌入矩阵

### 2.5 学习词嵌入

### 2.6 Word2Vec

### 2.7 负采样

### 2.8 GloVe 词向量

### 2.9 情绪分类

### 2.10 词嵌入除偏

## 第三周 序列模型和注意力机制

### 3.1 基础模型

### 3.2 选择最可能的句子

### 3.3 定向搜索

### 3.4 改进定向搜索

### 3.5 定向搜索的误差分析

### 3.6 Bleu得分（选修）

### 3.7 注意力模型直观理解

### 3.8 注意力模型

### 3.9 语音辨识

### 3.10 触发字检测

### 3.11 结论和致谢