



deeplearning.ai

Hyperparameter tuning

Tuning process

1

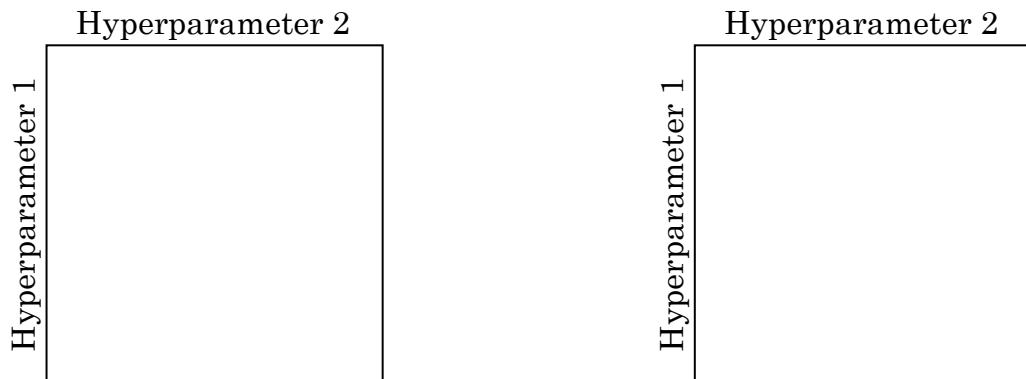
Hyperparameters

Andrew Ng

2

1

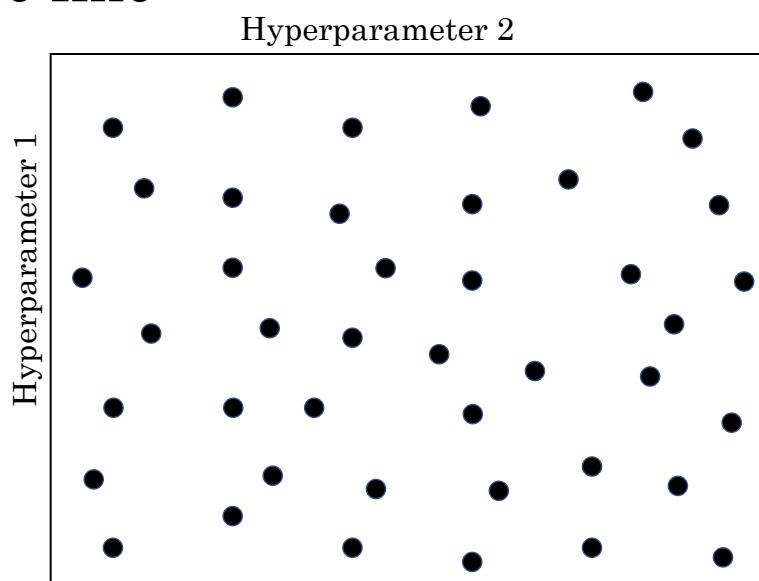
Try random values: Don't use a grid



Andrew Ng

3

Coarse to fine



Andrew Ng

4

2



deeplearning.ai

Hyperparameter tuning

Using an appropriate
scale to pick
hyperparameters

5

Picking hyperparameters at random

Andrew Ng

6

3

Appropriate scale for hyperparameters

Andrew Ng

7

Hyperparameters for exponentially weighted averages

Andrew Ng

8



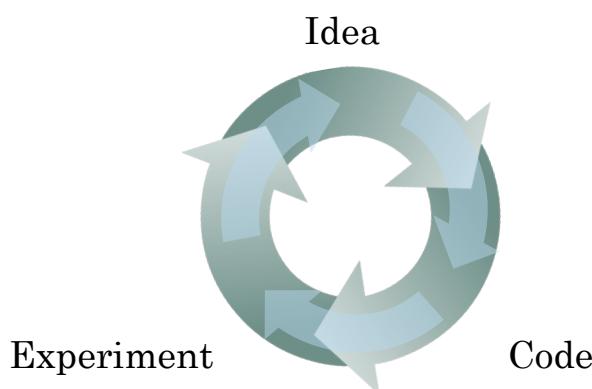
deeplearning.ai

Hyperparameters tuning

Hyperparameters tuning in practice: Pandas vs. Caviar

9

Re-test hyperparameters occasionally



- NLP, Vision, Speech, Ads, logistics,
- Intuitions do get stale.
Re-evaluate occasionally.

Andrew Ng

10

Babysitting one
model



Panda

Training many
models in parallel



Caviar

Andrew Ng

11



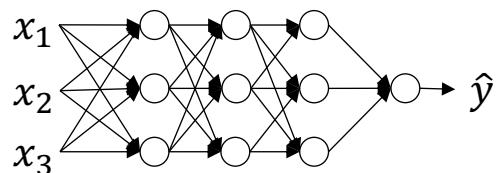
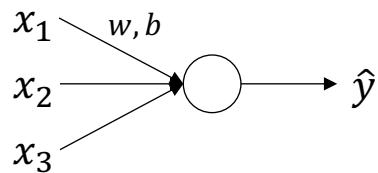
deeplearning.ai

Batch
Normalization

Normalizing activations
in a network

12

Normalizing inputs to speed up learning



Andrew Ng

13

Implementing Batch Norm

Andrew Ng

14



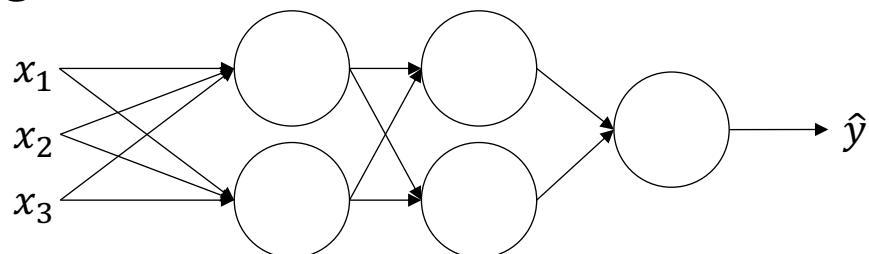
deeplearning.ai

Batch Normalization

Fitting Batch Norm into a neural network

15

Adding Batch Norm to a network



Andrew Ng

16

Working with mini-batches

Andrew Ng

17

Implementing gradient descent

Andrew Ng

18



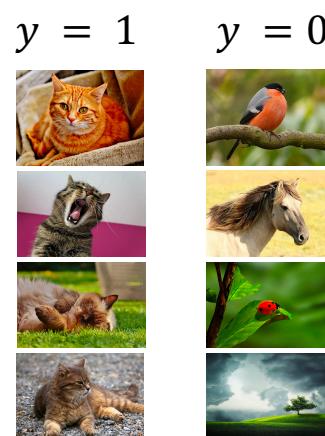
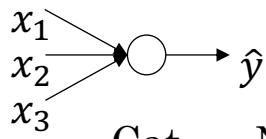
deeplearning.ai

Batch Normalization

Why does Batch Norm work?

19

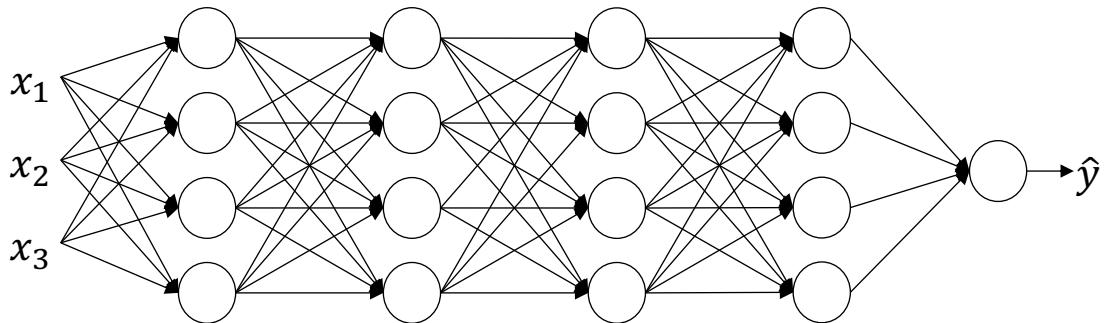
Learning on shifting input distribution



Andrew Ng

20

Why this is a problem with neural networks?



Andrew Ng

21

Batch Norm as regularization

- Each mini-batch is scaled by the mean/variance computed on just that mini-batch.
- This adds some noise to the values $z^{[l]}$ within that minibatch. So similar to dropout, it adds some noise to each hidden layer's activations.
- This has a slight regularization effect.

Andrew Ng

22



deeplearning.ai

Batch Normalization

Batch Norm at test time

Batch Norm at test time

$$\mu = \frac{1}{m} \sum_i z^{(i)}$$

$$\sigma^2 = \frac{1}{m} \sum_i (z^{(i)} - \mu)^2$$

$$z_{\text{norm}}^{(i)} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$

$$\tilde{z}^{(i)} = \gamma z_{\text{norm}}^{(i)} + \beta$$

Andrew Ng



deeplearning.ai

Multi-class classification

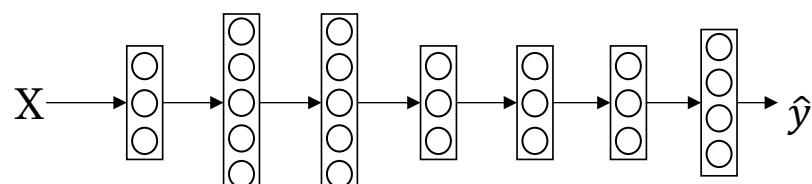
Softmax regression

25

Recognizing cats, dogs, and baby chicks



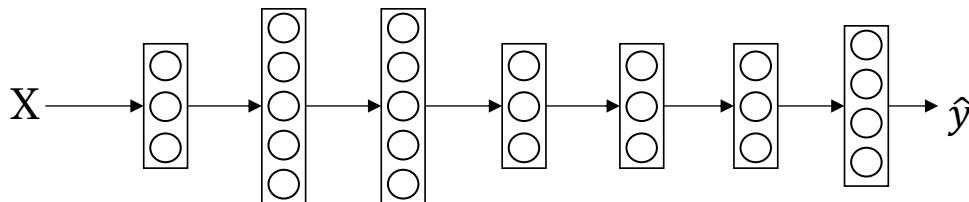
3 1 2 0 3 2 0 1



Andrew Ng

26

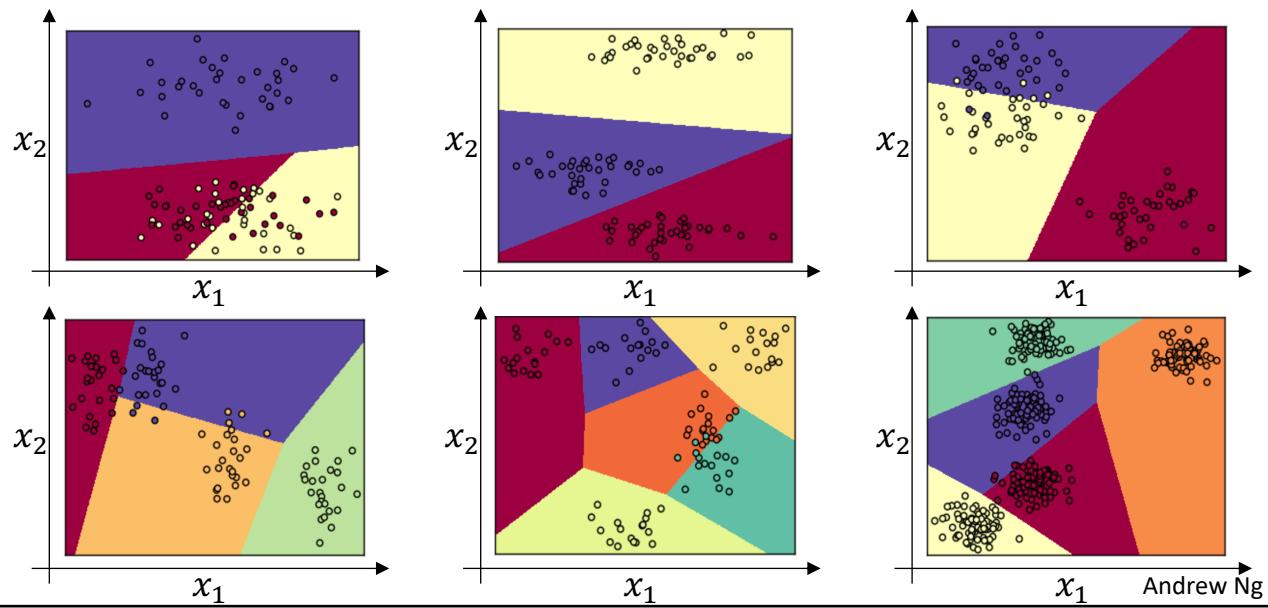
Softmax layer



Andrew Ng

27

Softmax examples



28

14



deeplearning.ai

Multi-class
classification

Trying a softmax
classifier

29

Understanding softmax

Andrew Ng

30

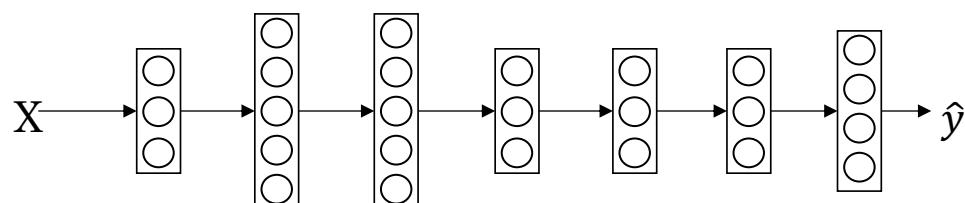
15

Loss function

Andrew Ng

31

Summary of softmax classifier



Andrew Ng

32



deeplearning.ai

Programming Frameworks

Deep Learning frameworks

33

Deep learning frameworks

- Caffe/Caffe2
- CNTK
- DL4J
- Keras
- Lasagne
- mxnet
- PaddlePaddle
- TensorFlow
- Theano
- Torch

- Choosing deep learning frameworks
- Ease of programming (development and deployment)
 - Running speed
 - Truly open (open source with good governance)

Andrew Ng

34



Programming Frameworks

TensorFlow

35

Motivating problem

Andrew Ng

36

18

Code example

```
import numpy as np
import tensorflow as tf

coefficients = np.array([[1], [-20], [25]])

w = tf.Variable([0], dtype=tf.float32)
x = tf.placeholder(tf.float32, [3,1])
cost = x[0]*w**2 + x[1]*w + x[2]*0    # (w-5)**2
train = tf.train.GradientDescentOptimizer(0.01).minimize(cost)
init = tf.global_variables_initializer()

session = tf.Session()
session.run(init)
print(session.run(w))

with tf.Session() as session:
    session.run(init)
    print(session.run(w))

for i in range(1000):
    session.run(train, feed_dict={x:coefficients})
    print(session.run(w))
```

Andrew Ng