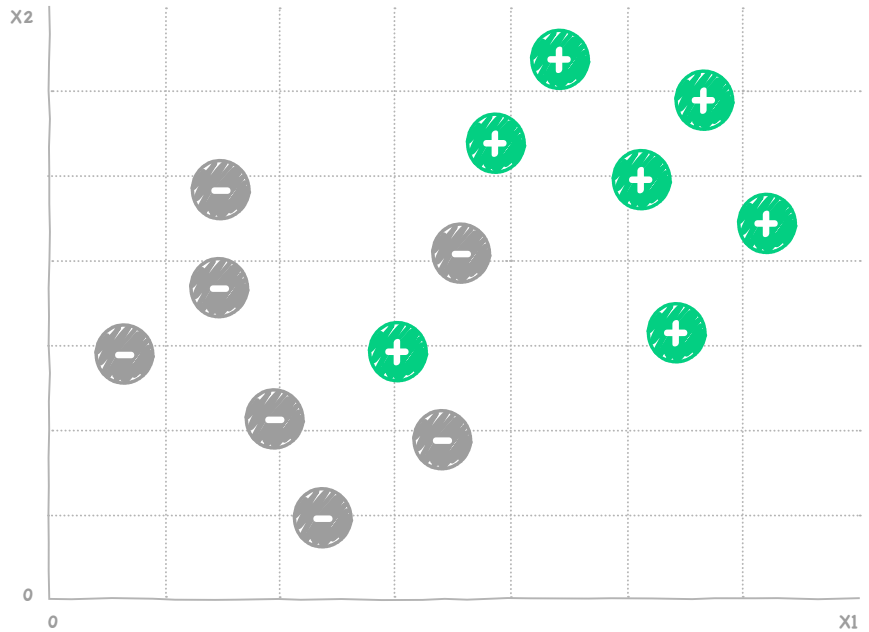
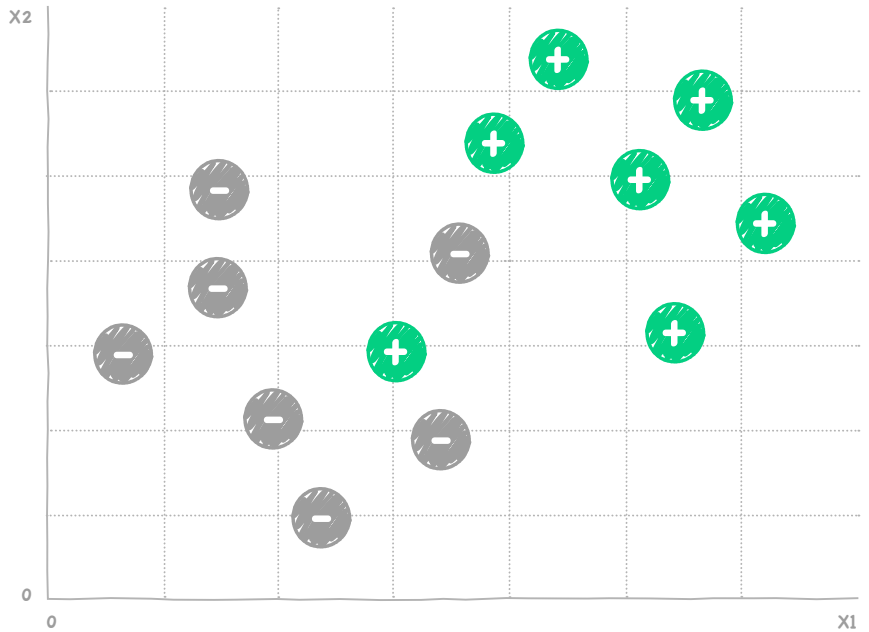
LECTURE 10-3

NN OVERFITTING : REGULARIZATION

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml

Overfitting





Am I Overfitting?

- Very high accuracy on the training dataset (eg: 0.99)
- Poor accuracy on the test data set (0.85)

Solutions for Overfitting

- More training data!
- Reduce the number of features
- Regularization

Regularization

- Let's not have too big numbers in the weight

Regularization

- Let's not have too big numbers in the weight

$$\text{Cost} + \lambda \sum w^2$$

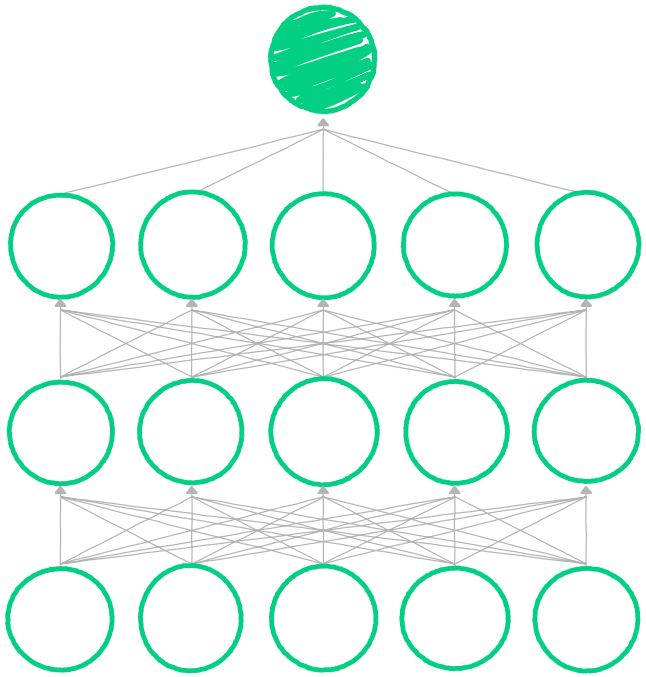
Regularization Strength

```
l2reg = 0.001 * tf.reduce_sum(tf.square(W))
```

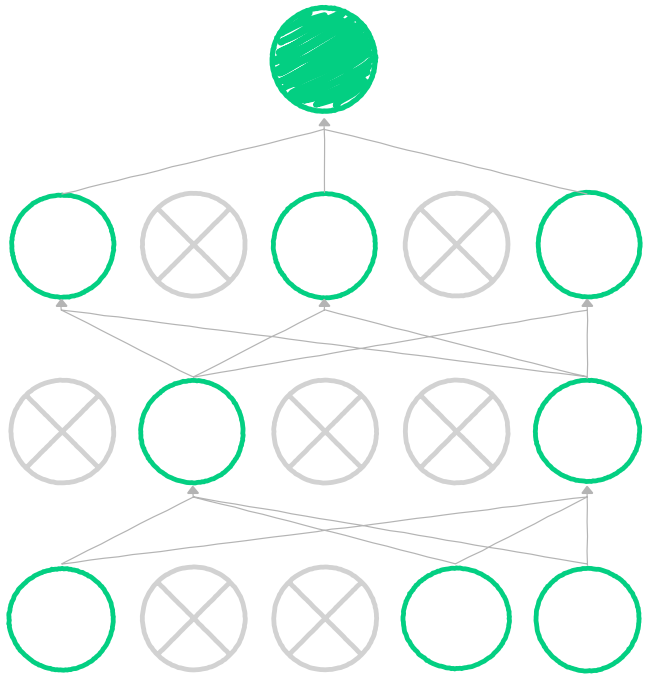
Dropout

A Simple Way to Prevent
Neural Networks from Overfitting

Srivastava et al., 2014



(A) Standard Neural Net

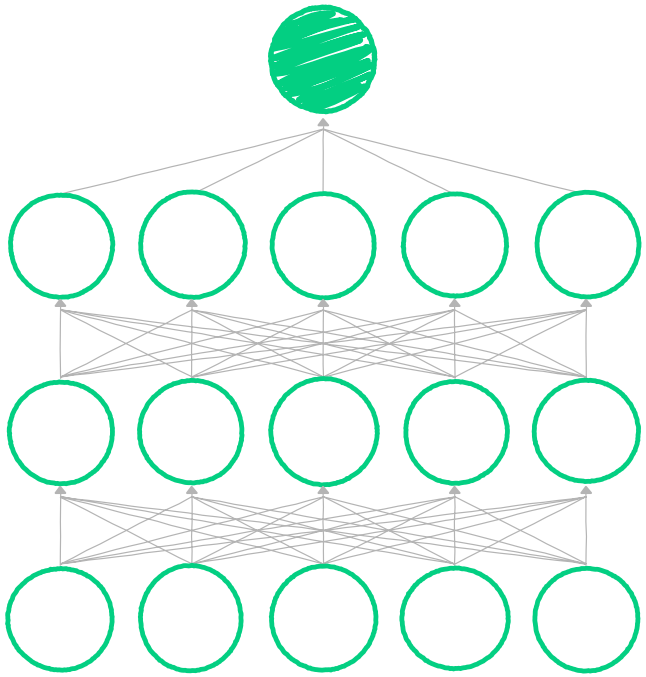


(B) After Applying Dropout

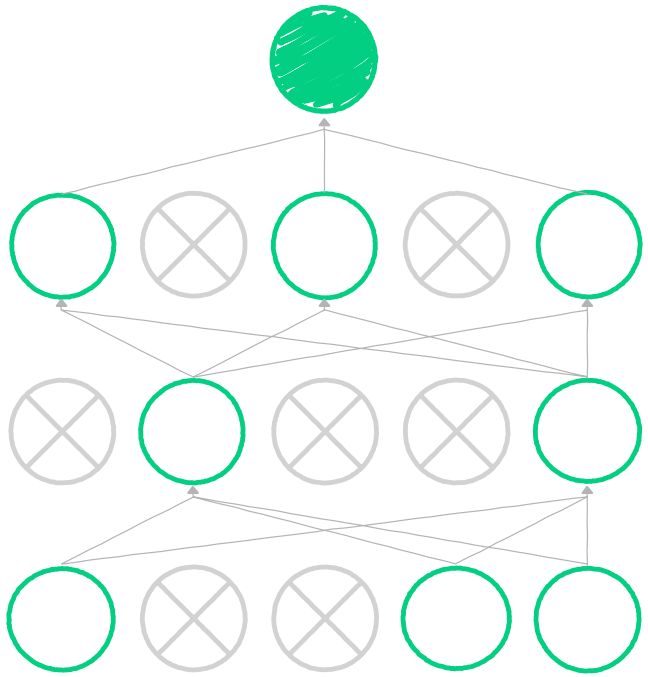
Dropout

Regularization :
Randomly Set Some Neurons to Zero in the forward Pass

Srivastava et al., 2014



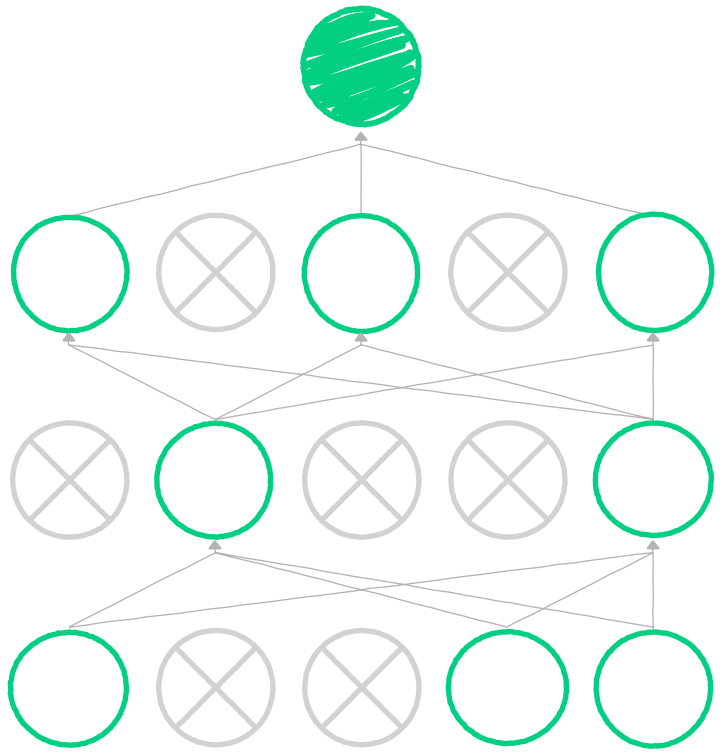
(A) Standard Neural Net



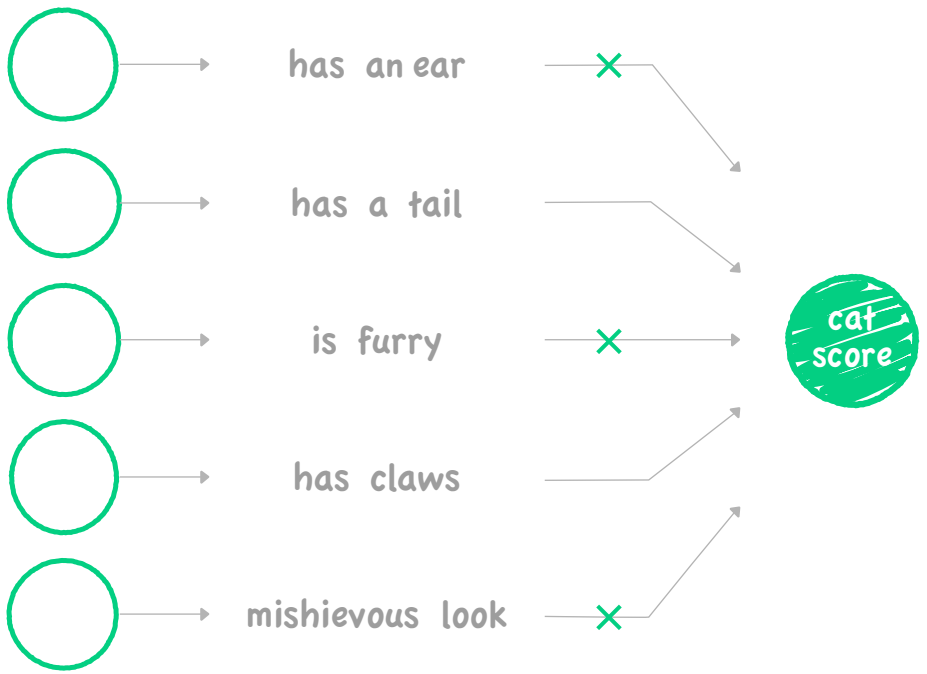
(B) After Applying Dropout

Dropout

How could this possibly be a good idea?



Forces the Network to Have a Redundant Representation.



TensorFlow Implementation

```
dropout_rate = tf.placeholder("float")
_L1 = tf.nn.relu(tf.add(tf.matmul(X, W1), B1))
L1 = tf.nn.dropout(_L1, dropout_rate)
```

TRAIN:

```
sess.run(optimizer, feed_dict={X: batch_xs, Y: batch_ys, dropout_rate: 0.7})
```

EVALUATION:

```
print "Accuracy:", accuracy.eval({X: mnist.test.images, Y: mnist.test.labels,
dropout_rate: 1})
```

NEXT LECTURE

NN LEGO PLAY!