CST 463 - Advanced Machine Learning

Dr. Glenn Bruns

# Lab: Siamese networks

Please work with your team. For a Siamese network you will probably want to generate training batches on the fly, using a batch generator.

1. Modify your CIFAR10 model so that it uses a batch generation function.

Check the [Keras documentation for fit()](https://www.tensorflow.org/api_docs/python/tf/keras/Model#fit). An easy option is to just create a Python generator function that returns a batch. Keras' fit() method will work with Python generators.

I recommend that you start by writing a function make\_batch() that, given X and y, returns X\_batch, y\_batch. Then write a Python generator batch\_generator() that calls it.

Check to make sure make\_batch() works. Then modify your .fit() call to provide batch\_generator in place of the training data.

You can also use your batch\_generator to generate validation data.

1. If you achieve this, try adding data augmentation to your batch generator.

### 

### Hints:

1. For the make\_batch() function, a simple approach is like this:

def make\_batch(X, y, batch\_size):

rows = np.random.choice(X.shape[0], batch\_size)

return X[rows], y[rows]

You can choose to use replace=True or replace=Fase in the random.choice() call.

For the batch\_generator() function, this works:

def batch\_generator(X, y, batch\_size=32):

while True:

yield make\_batch(X, y, batch\_size)

For the .fit() call, you can do something like:

.fit(batch\_generator(X\_train, y\_train), steps\_per\_epoch=32, …)