



```
loss_weights = loss_weights + loss_weights * 0.95
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

input_a_np = np.random.random((10, 4))
input_b_np = np.random.random((10, 3))

output_a_np = np.random.random((10, 4))
output_b_np = np.random.random((10, 3))

# Fit and train on batches
out = model.train_on_batch([input_a_np, input_b_np],
                           [output_a_np, output_b_np])
out = model.train_on_batch({'input_a': input_a_np, 'input_b':
                           [output_a_np, output_b_np]})
out = model.train_on_batch({'input_a': input_a_np, 'input_b':
                           [output_a_np, output_b_np]},
                           {'dense_1': output_a_np, 'dropout': output_b_np})

# Test fit
out = model.fit([input_a_np, input_b_np],
               [output_a_np, output_b_np], nb_epochs=5, batch_size=4)
out = model.fit({'input_a': input_a_np, 'input_b': input_b_np},
               [output_a_np, output_b_np], nb_epochs=5, batch_size=4)
out = model.fit({'input_a': input_a_np, 'input_b': input_b_np},
               {'dense_1': output_a_np, 'dropout': output_b_np},
               nb_epochs=1, batch_size=4)

# Evaluate
out = model.evaluate([input_a_np, input_b_np],
                    [output_a_np, output_b_np],
                    validation_split=0.5)
out = model.evaluate({'input_a': input_a_np,
                    'input_b': input_b_np},
                    [output_a_np, output_b_np],
                    validation_split=0.5)
out = model.evaluate({'input_a': input_a_np,
                    'input_b': input_b_np},
                    {'dense_1': output_a_np,
                    'dropout': output_b_np},
                    validation_split=0.5)
```

VOLUME 4 | NUMBER 3 | JUNE 2023