

```
1 n_ukrytych = 2
2 enkoder_wejscie = Input(shape=(szerokosc, wysokosc))
3 x = Flatten()(enkoder_wejscie)
4 x = Dense(500, activation='relu')(x)
5 x = Dense(120, activation='relu')(x)
6
7 mu = Dense(n_ukrytych)(x)
8 log_sigma = Dense(n_ukrytych)(x)
9
10 def probka(args):
11     i_mu, i_log_sigma = args
12     eps = K.random_normal(shape=(K.shape(i_mu)[0],
13                                     K.shape(i_mu)[1]))
14     return i_mu + K.exp(i_log_sigma) * eps
15
16 z = Lambda(probka, output_shape=(n_ukrytych,))([mu, log_sigma])
17 enkoder = Model(enkoder_wejscie, [mu, log_sigma, z])
18
19 dekoderek_wejscie = Input(shape=(n_ukrytych,))
20 x = Dense(120, activation='relu')(dekoder_wejscie)
21 x = Dense(500, activation='relu')(x)
22 x = Dense(szerokosc * wysokosc, activation='sigmoid')(x)
23 dekoderek = Model(dekoder_wejscie, x)
24
25 z_odkodowane = dekoderek(z)
```