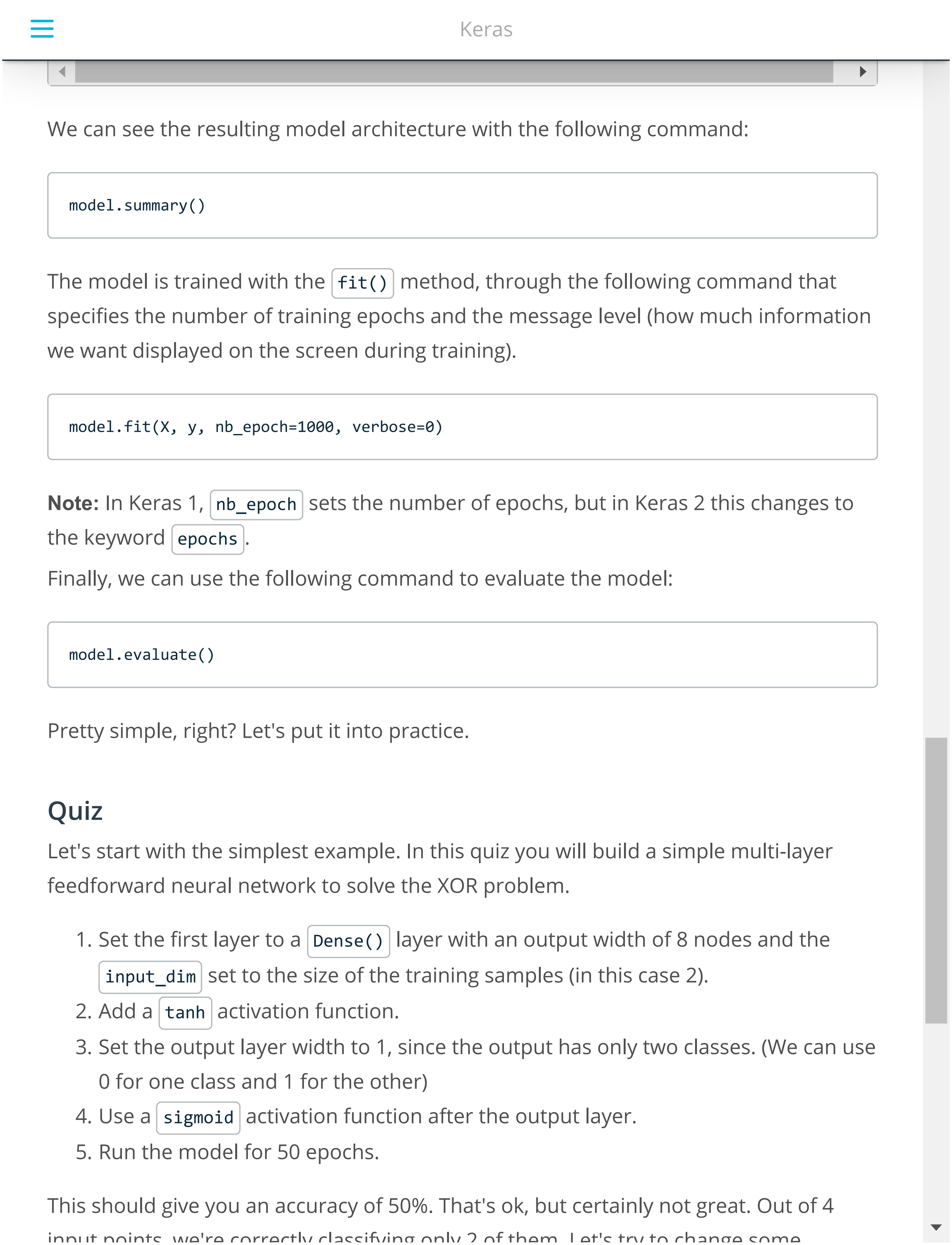

We can see the resulting model architecture with the following command:

```
model.summary()
```

The model is trained with the `fit()` method, through the following command that specifies the number of training epochs and the message level (how much information we want displayed on the screen during training).

```
model.fit(X, y, nb_epoch=1000, verbose=0)
```

Note: In Keras 1, `nb_epoch` sets the number of epochs, but in Keras 2 this changes to the keyword `epochs`.

Finally, we can use the following command to evaluate the model:

```
model.evaluate()
```

Pretty simple, right? Let's put it into practice.

Quiz

Let's start with the simplest example. In this quiz you will build a simple multi-layer feedforward neural network to solve the XOR problem.

1. Set the first layer to a `Dense()` layer with an output width of 8 nodes and the `input_dim` set to the size of the training samples (in this case 2).
2. Add a `tanh` activation function.
3. Set the output layer width to 1, since the output has only two classes. (We can use 0 for one class and 1 for the other)
4. Use a `sigmoid` activation function after the output layer.
5. Run the model for 50 epochs.

This should give you an accuracy of 50%. That's ok, but certainly not great. Out of 4 input points, we're correctly classifying only 2 of them. Let's try to change some