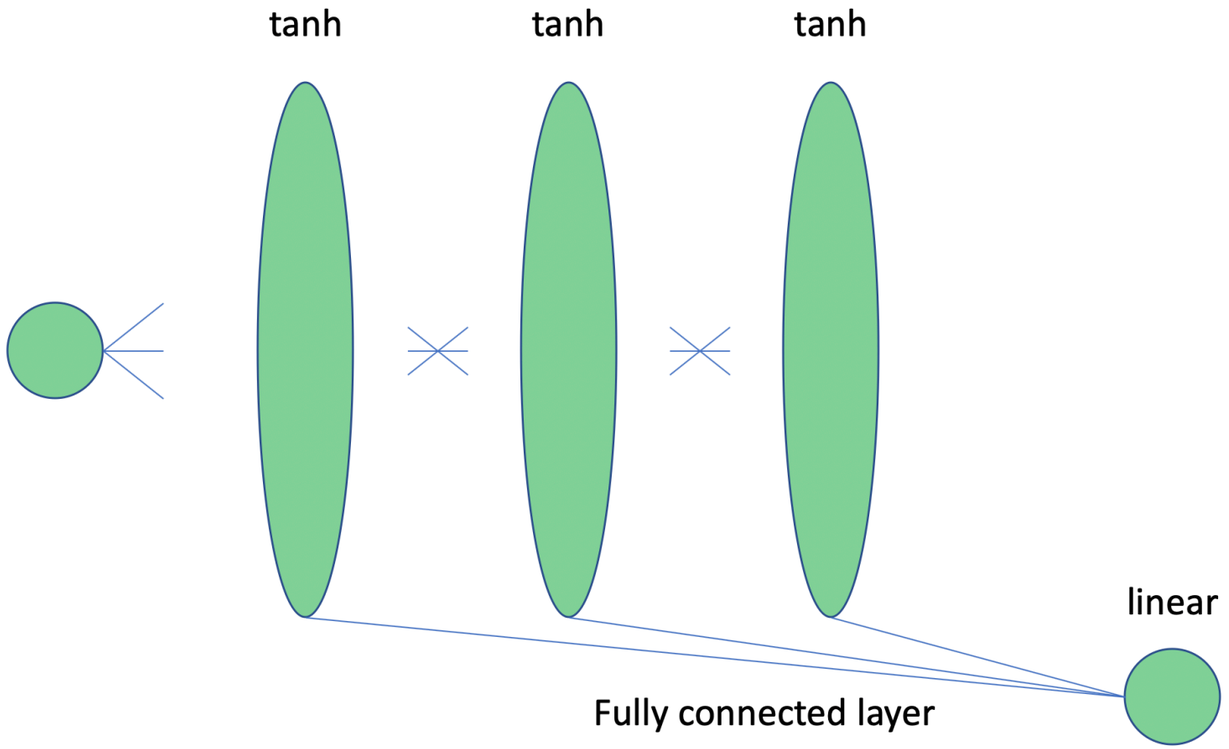
CME 216, ME 343 - Spring 2020

Eric Darve, ICME

Let us now assume that we now want to code up the following DNN.


tanh
tanh
tanh
linear
Fully connected layer

Hidden layers 1, 2, and 3 are fully connected with the previous layer.

But then, the output layer is fully connected to hidden layers 1, 2, and 3.

This cannot be expressed using a sequential DNN.

We need to use the functional API for that.

It will look very similar to the previous syntax.

```
from tensorflow.keras import layers, Model
input_ = layers.Input(shape=1)
hidden1 = layers.Dense(4, activation="tanh")(input_)
hidden2 = layers.Dense(4, activation="tanh")(hidden1)
hidden3 = layers.Dense(4, activation="tanh")(hidden2)
concat = layers.Concatenate()([hidden1, hidden2, hidden3])
output = layers.Dense(1, activation="linear")(concat)
model = Model(inputs=[input_], outputs=[output])
```

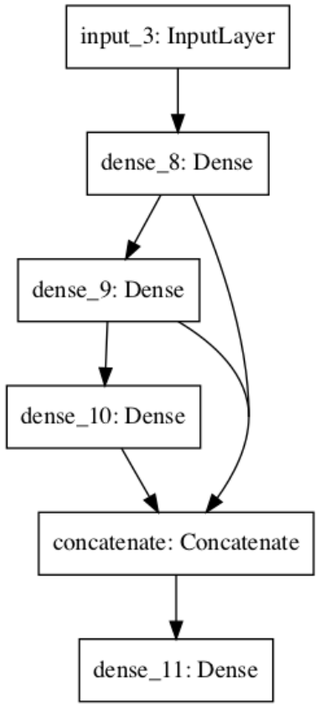
We recognize a new command

```
layers.Concatenate()([hidden1, hidden2, hidden3])
```

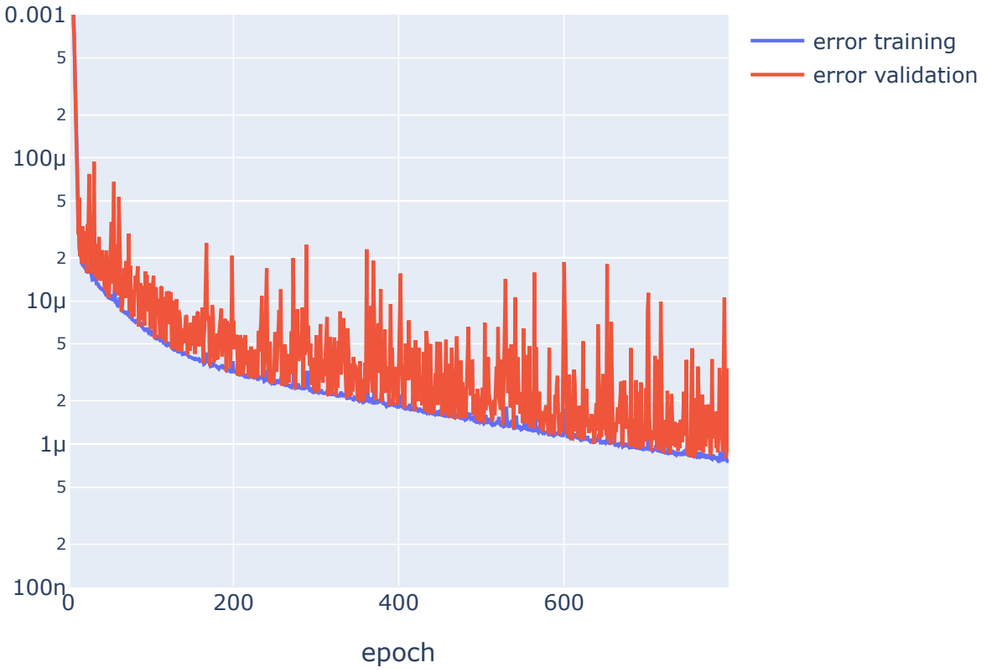
which concatenates together the output of multiple layers.

We can plot the model using

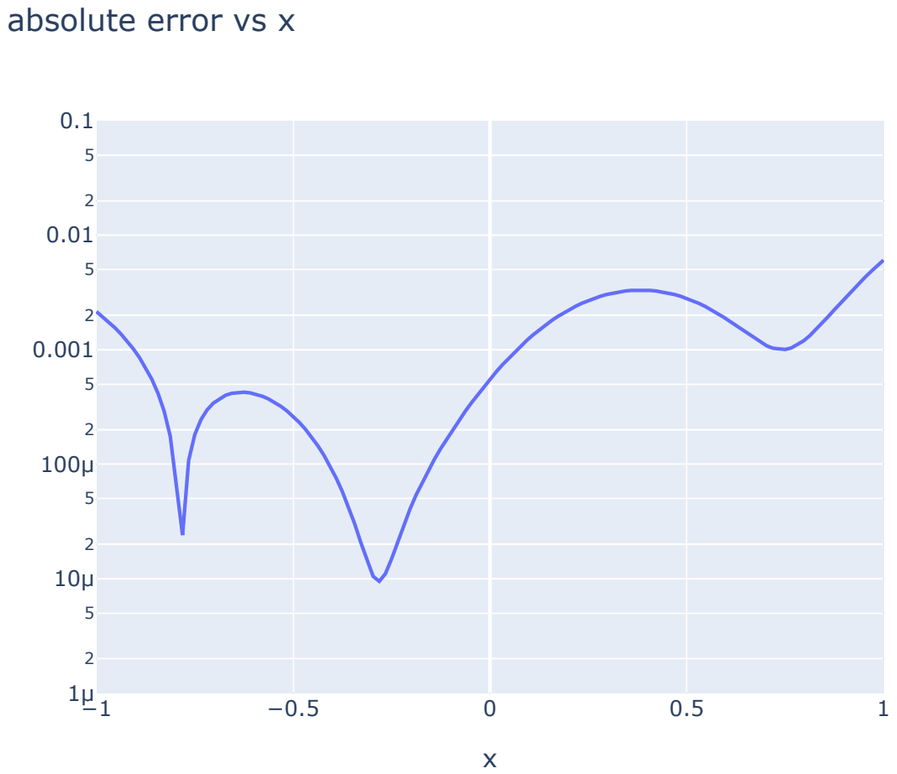
```
keras.utils.plot_model(model)
```


input_3: InputLayer
dense_8: Dense
dense_9: Dense
dense_10: Dense
concatenate: Concatenate
dense_11: Dense

Convergence



Error


absolute error vs x
0.1
5
2
0.01
5
2
0.001
5
2
100μ
5
2
10μ
5
2
1μ
−1
−0.5
0
0.5
1
x