EDAN95
Applied Machine Learning
http://cs.lth.se/edan95/
Lecture 9: Autoencoders and Generative Learning

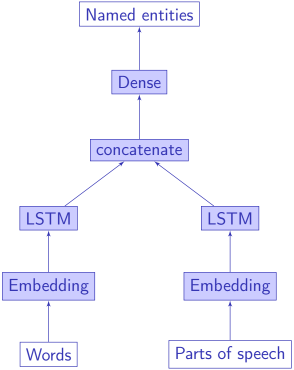
Pierre Nugues

Lund University
Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

December 3, 2018

The Functional Model

So far, we have used the `Sequential` model to build networks
These models correspond to pipelines with one input and one output



To build graphs, we need to use the functional model.

Comparing the Models

Sequential:

```
seq_model = Sequential()
seq_model.add(layers.Dense(32, activation='relu',
  input_shape=(64,)))
seq_model.add(layers.Dense(32, activation='relu'))
seq_model.add(layers.Dense(10, activation='softmax'))
```

Functional:

```
input_tensor = Input(shape=(64,))
x = layers.Dense(32, activation='relu')(input_tensor)
x = layers.Dense(32, activation='relu')(x)
output_tensor = layers.Dense(10, activation='softmax')(x)
model = Model(input_tensor, output_tensor)
```

From Chollet, page 237

Building a Multi Input Model

To build a multi input, we need the functional model and at a certain point, merge the branches with `layers.concatenate()` function

Code Example

We will now build a NER tagger that uses two inputs: the words and parts of speech and we will compare it to a sequential model

Jupyter Notebooks: `5.2-multiinput.ipynb` and `5.3-monoinput.ipynb`

Multiple Outputs

???