About Keras models

There are two types of models available in Keras: the Sequential model and the Model class used with functional API.

These models have a number of methods in common:

- `model.summary()`: prints a summary representation of your model.
- `model.get_config()`: returns a dictionary containing the configuration of the model. The model can be reinstantiated from its config via:

```
config = model.get_config()
model = Model.from_config(config)
# or, for Sequential:
model = Sequential.from_config(config)
```

- `model.get_weights()`: returns a list of all weight tensors in the model, as Numpy arrays.
- `model.set_weights(weights)`: sets the values of the weights of the model, from a list of Numpy arrays. The arrays in the list should have the same shape as those returned by `get_weights()`.
- `model.to_json()`: returns a representation of the model as a JSON string. Note that the representation does not include the weights, only the architecture. You can reinstantiate the same model (with reinitialized weights) from the JSON string via:

```
from models import model_from_json

json_string = model.to_json()
model = model_from_json(json_string)
```

- `model.to_yaml()`: returns a representation of the model as a YAML string. Note that the representation does not include the weights, only the architecture. You can reinstantiate the same model (with reinitialized weights) from the YAML string via:

```
from models import model_from_yaml

yaml_string = model.to_yaml()
model = model_from_yaml(yaml_string)
```

- `model.save_weights(filepath)`: saves the weights of the model as a HDF5 file.
- `model.load_weights(filepath)`: loads the weights of the model from a HDF5 file (created by `save_weights`).