

---

# Gradientzoo Documentation

*Release 0.8.8*

**Eric Florenzano**

Sep 27, 2017



---

# Contents

---

<b>1</b>	<b>Using Keras with Gradientzoo</b>	<b>3</b>
1.1	Initialization . . . . .	3
1.2	Loading a Model . . . . .	4
1.3	Saving a Model . . . . .	4
1.4	Creating a Keras callback that will save automatically . . . . .	4
<b>2</b>	<b>Using Tensorflow with Gradientzoo</b>	<b>5</b>
2.1	Initialization . . . . .	5
2.2	Loading a Model . . . . .	6
2.3	Saving a Model . . . . .	6
<b>3</b>	<b>Using Lasagne with Gradientzoo</b>	<b>7</b>
3.1	Initialization . . . . .	7
3.2	Loading a Model . . . . .	8
3.3	Saving a Model . . . . .	8
<b>4</b>	<b>Using plain old Python with Gradientzoo</b>	<b>9</b>
4.1	Initialization . . . . .	9
4.2	Loading a Model . . . . .	10
4.3	Saving a Model . . . . .	10
<b>5</b>	<b>Indices and tables</b>	<b>11</b>



Visit [Gradientzoo.com](http://Gradientzoo.com)

Contents:



---

## Using Keras with Gradientzoo

---

This doc will demonstrate how to save and load your trained neural network in Python using Keras.

### Initialization

```
from gradientzoo import KerasGradientzoo
```

This is the client that you'll create to save and load your model to the service.

**param username** The gradientzoo username of the project you want to connect to

**param model\_slug** The slug (short url-safe name) of the model

**param api\_base** *Optional* URL prefix to specify where the gradientzoo API is located, which is especially useful if you are running your own instance of gradientzoo

**param auth\_token\_id** Your authentication token as provided by the service

**param default\_dir** *Optional* When a download directory is asked for later, this is what it will default to if you pass None

**Note:** For convenience and aesthetics, you can pass both the username and model\_slug as username/model\_slug to the username parameter.

```
# Examples of how to instantiate the client
zoo = KerasGradientzoo('exampleuser/modelslug')
zoo = KerasGradientzoo('exampleuser', 'modelslug', auth_token_id='12345af')
zoo = KerasGradientzoo('exampleuser/modelslug', api_base='https://api2.gradientzoo.com
↪')
```

### Examples

The next few examples will assume `zoo`, an instantiated `KerasGradientzoo` client:

```
from gradientzoo import KerasGradientzoo
zoo = KerasGradientzoo('exampleuser/modelslug')
```

## Loading a Model

If you want to load a model from the remote cloud, that can be done by calling the `load` method on the client.

- param model** The Keras model you want to load the weights into
- param filename** *Optional* The filename of the thing you would like to download
- param id** *Optional* If there is a specific version of the file you want to download, provide the file id (as seen on the website) here
- param dir** *Optional* The directory to download the file to
- returns file\_model** A dictionary with metadata about the file you downloaded

```
# Example of how to load a model
zoo.load(your_keras_model)
```

## Saving a Model

If you want to save your model to the remote cloud, simply call the `upload_file` method on the client.

- param model** The Keras model you want to save the weights for
- param filename** *Optional* The name of the file you're uploading
- param metadata** *Optional* A bag of key/value pairs to be associated with this file (must be JSON encodable)
- param dir** *Optional* The temporary directory to save the model weights to before uploading them

```
# Example of how to save a model
zoo.save(your_keras_model)
```

## Creating a Keras callback that will save automatically

Often times you don't want to manage saving the model yourself. To make this easier, we provide a Keras callback that will save every N batches or every N epochs. To get a callback, call the `make_callback` method on the client.

- param model** The Keras model you want to save the weights for
- param after\_batches** *Optional* The number of batches after which to save
- param after\_epochs** *Optional* The number of epochs after which to save (defaults to 1)

```
# Example of how to automatically save a model using a callback
zoo_callback = zoo.make_callback(your_model)
your_model.fit(X_train, Y_train, # ...
               callbacks=[zoo_callback])
```

---

## Using Tensorflow with Gradientzoo

---

This doc will demonstrate how to save and load your trained neural network in Python using Tensorflow.

### Initialization

```
from gradientzoo import TensorflowGradientzoo
```

This is the client that you'll create to save and load your model to the service.

**param username** The gradientzoo username of the project you want to connect to

**param model\_slug** The slug (short url-safe name) of the model

**param api\_base** *Optional* URL prefix to specify where the gradientzoo API is located, which is especially useful if you are running your own instance of gradientzoo

**param auth\_token\_id** Your authentication token as provided by the service

**param default\_dir** *Optional* When a download directory is asked for later, this is what it will default to if you pass None

**Note:** For convenience and aesthetics, you can pass both the username and model\_slug as username/model\_slug to the username parameter.

```
# Examples of how to instantiate the client
zoo = TensorflowGradientzoo('exampleuser/modelslug')
zoo = TensorflowGradientzoo('exampleuser', 'modelslug', auth_token_id='12345af')
zoo = TensorflowGradientzoo('exampleuser/modelslug', api_base='https://api2.
↳gradientzoo.com')
```

### Examples

The next few examples will assume `zoo`, an instantiated `TensorflowGradientzoo` client:

```
from gradientzoo import TensorflowGradientzoo
zoo = TensorflowGradientzoo('exampleuser/modelslug')
```

## Loading a Model

If you want to load a model from the remote cloud, that can be done by calling the `load` method on the client.

**param session** The Tensorflow session you want to load the weights into

**param filename** *Optional* The filename of the thing you would like to download

**param id** *Optional* If there is a specific version of the file you want to download, provide the file id (as seen on the website) here

**param dir** *Optional* The directory to download the file to

**returns file\_model** A dictionary with metadata about the file you downloaded

```
# Example of how to load a Tensorflow session
with tf.Session() as sess:
    # Load latest weights from Gradientzoo
    zoo.load(sess)
```

## Saving a Model

If you want to save your model to the remote cloud, simply call the `upload_file` method on the client.

**param session** The Tensorflow session you want to save the weights for

**param filename** *Optional* The name of the file you're uploading

**param metadata** *Optional* A bag of key/value pairs to be associated with this file (must be JSON encodable)

**param dir** *Optional* The temporary directory to save the model weights to before uploading them

```
# Example of how to save a Tensorflow session
zoo.save(sess)
```

---

## Using Lasagne with Gradientzoo

---

This doc will demonstrate how to save and load your trained neural network in Python using Lasagne.

### Initialization

```
from gradientzoo import LasagneGradientzoo
```

This is the client that you'll create to save and load your model to the service.

**param username** The gradientzoo username of the project you want to connect to

**param model\_slug** The slug (short url-safe name) of the model

**param api\_base** *Optional* URL prefix to specify where the gradientzoo API is located, which is especially useful if you are running your own instance of gradientzoo

**param auth\_token\_id** Your authentication token as provided by the service

**param default\_dir** *Optional* When a download directory is asked for later, this is what it will default to if you pass None

**Note:** For convenience and aesthetics, you can pass both the username and model\_slug as username/model\_slug to the username parameter.

```
# Examples of how to instantiate the client
zoo = LasagneGradientzoo('exampleuser/modelslug')
zoo = LasagneGradientzoo('exampleuser', 'modelslug', auth_token_id='12345af')
zoo = LasagneGradientzoo('exampleuser/modelslug', api_base='https://api2.gradientzoo.
↪com')
```

### Examples

The next few examples will assume `zoo`, an instantiated `LasagneGradientzoo` client:

```
from gradientzoo import LasagneGradientzoo
zoo = LasagneGradientzoo('exampleuser/modelslug')
```

## Loading a Model

If you want to load a model from the remote cloud, that can be done by calling the `load` method on the client.

**param network** The Lasagne network you want to load the weights into

**param filename** *Optional* The filename of the thing you would like to download

**param id** *Optional* If there is a specific version of the file you want to download, provide the file id (as seen on the website) here

**param dir** *Optional* The directory to download the file to

**returns file\_model** A dictionary with metadata about the file you downloaded

```
# Example of how to load a Lasagne network
zoo.load(your_lasagne_network)
```

## Saving a Model

If you want to save your model to the remote cloud, simply call the `upload_file` method on the client.

**param network** The Lasagne network you want to save the weights for

**param filename** *Optional* The name of the file you're uploading

**param metadata** *Optional* A bag of key/value pairs to be associated with this file (must be JSON encodable)

**param dir** *Optional* The temporary directory to save the model weights to before uploading them

```
# Example of how to save a Lasagne network
zoo.save(your_lasagne_network)
```

---

## Using plain old Python with Gradientzoo

---

This doc will demonstrate how to save and load your trained neural network in Python. This is useful either for writing your own library integration, or for doing something outside of what the standard integrations expect.

### Initialization

```
from gradientzoo import Gradientzoo
```

This is the client that you'll create to save and load your model to the service.

**param username** The gradientzoo username of the project you want to connect to

**param model\_slug** The slug (short url-safe name) of the model

**param api\_base** *Optional* URL prefix to specify where the gradientzoo API is located, which is especially useful if you are running your own instance of gradientzoo

**param auth\_token\_id** Your authentication token as provided by the service

**param default\_dir** *Optional* When a download directory is asked for later, this is what it will default to if you pass None

**Note:** For convenience and aesthetics, you can pass both the username and model\_slug as username/model\_slug to the username parameter.

```
# Examples of how to instantiate the client
zoo = Gradientzoo('exampleuser/modelslug')
zoo = Gradientzoo('exampleuser', 'modelslug', auth_token_id='12345af')
zoo = Gradientzoo('exampleuser/modelslug', api_base='https://api2.gradientzoo.com')
```

### Examples

The next few examples will assume `zoo`, an instantiated Gradientzoo client:

```
from gradientzoo import Gradientzoo
zoo = Gradientzoo('exampleuser/modelslug')
```

## Loading a Model

If you want to load a model from the remote cloud, that can be done by calling the `download_file` method on the client.

**param filename** The filename of the thing you would like to download

**param id** *Optional* If there is a specific version of the file you want to download, provide the file id (as seen on the website) here

**param dir** *Optional* The directory to download the file to

**param chunk\_size** *Optional* The size of the chunks to read and flush to disk

**returns file\_model** A dictionary with metadata about the file you downloaded

```
# Example of how to download a file
filepath, file_model = zoo.download_file('model.npz')

with open(filepath, 'r') as f:
    data = f.read()

# Now load the data into your model however your framework or library works
```

## Saving a Model

If you want to save your model to the remote cloud, simply call the `upload_file` method on the client.

**param filename** The name of the file you're uploading

**param f** A file object that can be read to upload to the server

**param metadata** *Optional* A bag of key/value pairs to be associated with this file (must be JSON encodable)

```
# Example of how to upload a file, first by saving it out from your library
filename = '/tmp/model.npz'
numpy.savez(filename, your_model_tensor)

# Then sending it up to gradientzoo
with open(filename, 'r') as f:
    zoo.upload_file(filename, f, {'loss': your_model_loss})
```

## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`