



Picked for you:



Your First Deep Learning Project in Python
with Keras Step-By-Step

[Click to Take the FREE Deep Learning Crash-Course](#)

Search...



Dropout Regularization in Deep Learning Models With Keras



Regression Tutorial with the Keras Deep Learning Library in Python
Brownlee on June 20, 2016 in Deep Learning

Tweet

Tweet

Share

Share



Multi-Class Classification Tutorial with the Keras Deep Learning Library

A simple and powerful regularization technique for neural networks and deep learning models is **dropout**.



I [polbyda](#) [Solved](#) [Discover](#) [Your](#) [Keras](#) [Dropout](#) regularization technique and how to apply it to your models in Python with [Learning](#) [Model](#)

After reading this post you will know:

Loving the Tutorials?

- How the dropout regularization technique works.
- How to use dropout on your input layers.
- How to use dropout on your hidden layers.
- How to tune the dropout level on your problem.

[>> SEE WHAT'S INSIDE](#)

Kick-start your project with my new book [Deep Learning With Python](#), including *step-by-step tutorials* and the *Python source code* files for all examples.

Let's get started.

- **Update Oct/2016:** Updated for Keras 1.1.0, TensorFlow 0.10.0 and scikit-learn v0.18.
- **Update Mar/2017:** Updated for Keras 2.0.2, TensorFlow 1.0.1 and Theano 0.9.0.
- **Update Sep/2019:** Updated for Keras 2.2.5 API.

Never miss a tutorial:



Picked for you:



Your First Deep Learning Project in Python with Keras Step-By-Step



How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras



Regression Tutorial with the Keras Deep Learning Library in Python



Multi-Class Classification Tutorial with the Keras Deep Learning Library

Dropout Regularization in Deep Learning
Photo by [Trekking Rinjani](#), some rights reserved



How to Save and Load Your Keras Deep Learning Model

Dropout Regularization For Neural Networks

Dropout is a regularization technique for neural network models proposed by Nitish Srivastava et al.

[Dropout: A Simple Way to Prevent Neural Networks from Overfitting \(download the PDF\)](#).

Loving the Tutorials?

Dropout is a technique where randomly selected neurons are ignored during training. They are “dropped-out” randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.

[>> SEE WHAT'S INSIDE](#)

As a neural network learns, neuron weights settle into their context within the network. Weights of neurons are tuned for specific features providing some specialization. Neighboring neurons become to rely on this specialization, which if taken too far can result in a fragile model too specialized to the training data. This reliant on context for a neuron during training is referred to complex co-adaptations.

You can imagine that if neurons are randomly dropped out of the network during training, that other neurons will have to step in and handle the representation required to make predictions for the missing neurons. This is believed to result in multiple independent internal representations being learned by the network.

The effect is that the network becomes less sensitive to the specific weights of neurons. This in turn results in a network that is capable of better generalization and is less likely to overfit the training data.

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

[START MY EMAIL COURSE](#)

Need help with Deep Learning in Python?

Take my free 2-week email course and discover ML, NLP, Computer Vision, and Reinforcement Learning with code).

[Start Machine Learning](#)





Start Your FREE Mini-Course Now

Picked for you:

Dropout Regularization in Keras

Your First Deep Learning Project in Python
with Keras Step-By-Step

Dropout is easily implemented by randomly selecting nodes to be dropped-out with a given probability (e.g. 20%) each weight update cycle. This is how Dropout is implemented in Keras. Dropout is only used during the training of a model not used when evaluating the skill of the model.

 How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras

Next we will explore a few different ways of using Dropout in Keras.

 Regression Tutorial with the Keras Deep Learning Library in Python

Samples will use the Sonar dataset. This is a binary classification problem identifying rocks and mock-mines from sonar chirp returns. It is a good problem because input values are numerical and have the same scale.

 Multi-Class Classification Tutorial with the Keras Deep Learning Library

A dataset can be downloaded from the UCI Machine Learning Repository and saved to the current working directory with the file name sonar.csv.

 We will evaluate the developed models using scikit-learn with 10-fold cross-validation to see the results.

There are 60 input values and a single output value and the input values are scaled between 0 and 1. The network has two hidden layers, the first with 60 units and the second with 30. Stochastic gradient descent is used to train the model with a relatively low learning rate and momentum.

The Deep Learning with Python EBook is available for purchase at [RealPython.com](#).

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

```

1 # Baseline Model for the Sonar Dataset
2 from pandas import read_csv
3 from keras.models import Sequential
4 from keras.layers import Dense
5 from keras.wrappers.scikit_learn import KerasClassifier
6 from keras.optimizers import SGD
7 from sklearn.model_selection import cross_val_score
8 from sklearn.preprocessing import LabelEncoder
9 from sklearn.model_selection import StratifiedKFold
10 from sklearn.preprocessing import StandardScaler
11 from sklearn.pipeline import Pipeline
12 # load dataset
13 dataframe = read_csv("sonar.csv", header=None)
14 dataset = dataframe.values
15 # split into input (X) and output (Y) variables
16 X = dataset[:,0:60].astype(float)
17 Y = dataset[:,60]
18 # encode class values as integers
19 encoder = LabelEncoder()
20 encoder.fit(Y)
21 encoded_Y = encoder.transform(Y)
22
23 # baseline
24 def create_baseline():
25     # create model
26     model = Sequential()

```

Start Machine Learning

```

27 model.add(Dense(60, input_dim=60, activation='relu'))
28 model.add(Dense(30, activation='relu'))
29 model.add(Dense(1, activation='sigmoid'))
30 # Compile model
31 sgd = SGD(lr=0.01, momentum=0.8)
32 model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['accuracy'])
33 return model
34
35 estimators = []
36 estimators.append('standardize', StandardScaler())
37 estimators.append('baseline', KerasClassifier(build_fn=create_baseline, epochs=300, batch_size=16, verbose=0))
38 pipeline = Pipeline(estimators)
39 kfold = StratifiedKFold(n_splits=10, shuffle=True)
40 results = cross_val_score(pipeline, X, encoded_Y, cv=kfold)
41 print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))

```

 How to Grid Search Hyperparameters for your Deep Learning Model. Your results may vary given the stochastic nature of the algorithm or numerical precision. Consider running the example a few times and compare the results.

Running the example generates an estimated classification accuracy:

 [Regression Tutorial with the Keras Deep Learning Library in Python](#)

Baseline: 86.04% (4.58%)

Using Dropout on the Visible Layer

 [Multi-Class Classification Tutorial with the Keras Deep Learning Library](#)

Dropout can be applied to input neurons called the visible layer.

In the example below we add a new Dropout layer between the input layer and the hidden layer. We set the rate to 0.2, meaning one in 5 inputs will be randomly dropped.

Additionally, as recommended in the original paper on Dropout, a weight constraint is applied to the hidden layer. This ensures that the maximum norm of the weights does not exceed a value of 3. This is done by setting the kernel_constraint argument on the Dense class when constructing the layers.

The Deep Learning with Python EBook is where you can find the [Really Good Stuff](#). The learning rate was lifted by one order of magnitude and the momentum was increased to 0.9. These increases in the learning rate were also recommended in the original Dropout paper.

[>> SEE WHAT'S INSIDE](#)

Continuing on from the baseline example above, the code below exercises the same network with input dropout.

```

1 # Example of Dropout on the Sonar Dataset: Visible Layer
2 from pandas import read_csv
3 from keras.models import Sequential
4 from keras.layers import Dense
5 from keras.layers import Dropout
6 from keras.wrappers.scikit_learn import KerasClassifier
7 from keras.constraints import maxnorm
8 from keras.optimizers import SGD
9 from sklearn.model_selection import cross_val_score
10 from sklearn.preprocessing import LabelEncoder
11 from sklearn.model_selection import StratifiedKFold
12 from sklearn.preprocessing import StandardScaler
13 from sklearn.pipeline import Pipeline
14 # load dataset
15 dataframe = read_csv("sonar.csv", header=None)
16 dataset = dataframe.values
17 # split into input (X) and output (Y) variables
18 X = dataset[:,0:60].astype(float)
19 Y = dataset[:,60]
20 # encode class values as integers
21 encoder = LabelEncoder()
22 encoder.fit(Y)

```

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

[START MY EMAIL COURSE](#)

[Start Machine Learning](#)

```

23 encoded_Y = encoder.transform(Y)
24
25 # dropout in the input layer with weight constraint
26 def create_model():
27     # create model
28     model = Sequential()
29     model.add(Dropout(0.2, input_shape=(60,)))
30     model.add(Dense(60, activation='relu', kernel_constraint=maxnorm(3)))
31     model.add(Dense(30, activation='relu', kernel_constraint=maxnorm(3)))
32     model.add(Dense(1, activation='sigmoid'))
33     #Compile model
34     sgd = SGD(lr=0.1, momentum=0.9)
35     model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['accuracy'])
36     return model
37
38 estimators = []
39 estimators.append(('standardize', StandardScaler()))
40 estimators.append(('mlp', KerasClassifier(build_fn=create_keras_model,
41                                         verbose=0)))
42 pipeline = Pipeline(estimators)
43 kfold = StratifiedKFold(n_splits=10, shuffle=True)
44 results = cross_val_score(pipeline, X, encoded_Y, cv=kfold)
45 print("Visible: %.2f%% (%.2f%%)" % (results.mean()*100, results.std() * 100))

```

 Learning Library in Python

 Your results may vary given the stochastic nature of the algorithm and numerical precision. Consider running the example a few times and compare the results.

 Multi-Class Classification Tutorial with the Keras Deep Learning Library

1 Visible: 83.52% (7.68%)

 How to Save and Load Your Keras Deep Learning Model

Using Dropout on Hidden Layers

Dropout can be applied to hidden neurons in the body of your network.

Loving the Tutorials?

In the example below Dropout is applied between the two hidden layers and between the last hidden layer and the output layer. Again a dropout rate of 20% is used as is a weight constraint on those layers.

The Deep Learning with Python EBook is where you'll find the *Really Good* stuff.

```

1 # Example of Dropout on the Sonar Dataset: Hidden Layer
2 from pandas import read_csv
3 from keras.models import Sequential
4 from keras.layers import Dense
5 from keras.layers import Dropout
6 from keras.wrappers.scikit_learn import KerasClassifier
7 from keras.constraints import maxnorm
8 from keras.optimizers import SGD
9 from sklearn.model_selection import cross_val_score
10 from sklearn.preprocessing import LabelEncoder
11 from sklearn.model_selection import StratifiedKFold
12 from sklearn.preprocessing import StandardScaler
13 from sklearn.pipeline import Pipeline
14 # load dataset
15 dataframe = read_csv("sonar.csv", header=None)
16 dataset = dataframe.values
17 # split into input (X) and output (Y) variables
18 X = dataset[:,0:60].astype(float)
19 Y = dataset[:,60]
20 # encode class values as integers
21 encoder = LabelEncoder()
22 encoder.fit(Y)
23 encoded_Y = encoder.transform(Y)
24
25 # dropout in hidden layers with weight constraint
26 def create_model():
27     # create model

```

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

[Start Machine Learning](#)

```

28 model = Sequential()
29 #ever miss a tutorial!
30 model.add(Dense(60, input_dim=60, activation='relu', kernel_constraint=maxnorm(3)))
31 model.add(Dropout(0.2))
32 model.add(Dense(30, activation='relu', kernel_constraint=maxnorm(3)))
33 model.add(Dropout(0.2))
34 model.add(Dense(1, activation='sigmoid'))
35 # Compile model
36 sgd = SGD(lr=0.1, momentum=0.9)
37 model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['accuracy'])
38
39 #estimators
40 estimators.append('standardize', StandardScaler())
41 estimators.append('mlp', KerasClassifier(build_fn=create_model, epochs=300, batch_size=16, verbose=0))
42 pipeline = Pipeline(estimators)
43 kfold = StratifiedKFold(n_splits=10, shuffle=True)
44 results = cross_val_score(pipeline, X, encoded_Y, cv=kfold)
45 print("Hidden: %.2f% (%.2f%) %%" % (results.mean()*100, r

```

Note: Your [results may vary](#) given the stochastic nature of the algorithm. Consider running the example a few times and comparing the results.

 [Regression Tutorial with the Keras Deep Learning Library in Python](#)

We can see that for this problem and for the chosen network configuration, dropout did not lift performance. In fact, performance was worse than the baseline.

 [Multi-Class Classification Tutorial with the Keras Deep Learning Library](#)

It is possible that additional training epochs are required or that further tuning is required.

1 Hidden: 83.59% (7.31%)

 [How to Save and Load Your Keras Deep Learning Model](#)

For Using Dropout

The original paper on Dropout provides experimental results on a number of datasets. As a result they provide a number of useful heuristics to consider when using dropout in practice.

Loving the Tutorials?

- Generally use a small dropout value of 20%-50% of neurons with 20% providing a good starting point. A probability too low has [noisy](#) effect and a value too high results in under-learning by the network.
- Use a larger network. You are likely to get better performance when dropout is used on a larger network, giving the network the opportunity to learn independent representations.
- Use dropout on incoming (visible) as well as hidden units. Application of dropout at each layer of the network has shown good results.
- Use a large learning rate with decay and a large momentum. Increase your learning rate by a factor of 10 to 100 and use a high momentum value of 0.9 or 0.99.
- Constrain the size of network weights. A large learning rate can result in very large network weights. Imposing a constraint on the size of network weights such as max-norm regularization with a size of 4 or 5 has been shown to improve results.

More Resources on Dropout

Below are some resources that you can use to learn more about dropout in neural network and deep learning models.

- [Dropout: A Simple Way to Prevent Neural Networks from Overfitting](#) (original paper).
- [Improving neural networks by preventing co-adaptation of feature detectors](#).
- [How does the dropout method work in deep learning?](#) on Quora.

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Summary tutorial:

is part of your social network and the dropout regularization technique for deep learning models. You learned:

- What dropout is and how it works.

Picked for you: How you can use dropout on your own deep learning models.

- Tips for getting the best results from dropout on your own models.

[Your First Deep Learning Project in Python](#)

[with Keras Step-By-Step](#)

have any questions about dropout or about this post? Ask your questions in the comments and I will do my best to answer.



How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras

Develop Deep Learning Projects



Regression Tutorial with the Keras Deep Learning Library in Python



Develop Deep Learning Models on Multi-Class Classification with the Keras Deep Learning Library



How to Save and Load Your Keras Deep Learning Model



Loving the Tutorials?

The Deep Learning with Python EBook is where you'll find the **Really Good** stuff.

[>> SEE WHAT'S INSIDE](#)

[Twitter](#)

[Twitter](#)

[Facebook](#)

[LinkedIn](#)

More On This Topic

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

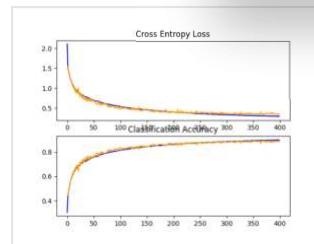
I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

[START MY EMAIL COURSE](#)

Skip the Academics. Just Results.

[SEE WHAT'S INSIDE](#)

Never miss a tutorial:



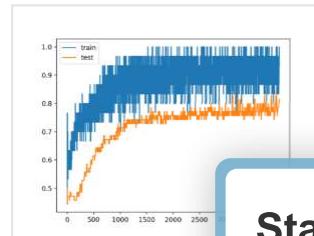
Picked for you:



How to Develop a CNN From Scratch for CIFAR-10 Photo...
[Your First Deep Learning Project in Python with Keras Step-By-Step](#)



How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras



How to Reduce Overfitting
[Regression Tutorial with the Keras Deep Learning Library in Python](#)

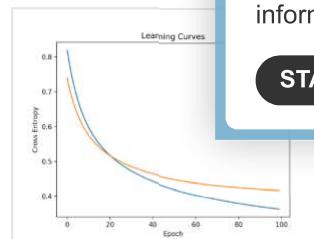


Multi-Class Classification Tutorial with the Keras Deep Learning Library



How to Save and Load Your Keras Deep Learning Model

A Gentle Introduction to Dropout



Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?

The [Deep Learning with Python](#) EBook is

where you'll find the *Really Good* TensorFlow 2 Tutorial: Get Started in Deep Learning...

[>> SEE WHAT'S INSIDE](#)



Dropout with LSTM Networks for Time Series Forecasting



Multi-Label Classification of Satellite Photos of...

About Jason Brownlee

[Start Machine Learning](#)





Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

[View all posts by Jason Brownlee →](#)



Picked for you:

< [Display Deep Learning Model Training History in Keras](#)
[Your First Deep Learning Project in Python with Keras Step-By-Step](#)

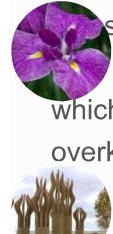
[A Tour of the Weka Machine Learning Workbench](#) >

 [How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras](#)

 **sheinis** September 7, 2016 at 6:27 am # [Regression Tutorial with the Keras Deep Learning Library in Python](#)
Hi,

thanks for the very useful examples!!

question: the goal of the dropouts is to reduce the risk of overfitting
Keras Deep Learning Library
wondering whether the accuracy is the best way to measure the which is by itself a way to reduce overfitting. As you are performing overkill? Maybe the drop in accuracy is actually a drop in amount of How to Save and Load Your Keras Deep Learning Model

 **Jason Brownlee** September 7, 2016 at 10:31 am #
Loving the Tutorials?
Great question.

The [Deep Learning with Python](#) EBook is Yes dropout is a technique to reduce the overfitting of the network to the training data. where you'll find the **Really Good** stuff.

k-fold cross-validation is a robust technique to estimate the skill of a model. It is well suited to determine whether a specific model has over or under fit the problem.

You could also look at diagnostic plots of loss over epoch on the training and validation datasets to determine how overlearning has been affected by different dropout configurations.

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE



Aditya September 23, 2016 at 8:44 am #

REPLY ↗

Very good post . Just one question why the need for increasing the learning rate in combination with setting max norm value?



Jason Brownlee September 24, 2016 at 8:01 am #

REPLY ↗

Great question. Perhaps less nodes being updated with dropout requires more change/update each batch.

Start Machine Learning

^

REPLY ↵

Star October 13, 2016 at 1:17 pm #



The LSTM performs well among the training dataset, while does not do well in the testing dataset, i.e. prediction.
Could you give me some advice for this problem?

Picked for you:



Jason Brownlee October 13, 2016 at 8:57 am #

[Keras Step-By-Step](#)

It sounds like overlearning.

REPLY ↵



Consider using a regularization technique like dropout discussed in this post above.

[How to Grid Search Hyperparameters for](#)

[Deep Learning Models in Python With](#)

Keras



Yuanliang Meng November 2, 2016 at 7:06 am #

[Regression Tutorial with the Keras Deep](#)

Some people mentioned that applying dropout on the LSTM

Here: <https://arxiv.org/abs/1508.03720>



I wonder if anyone has any comment on this.

[Multi-Class Classification Tutorial with the](#)

[Keras Deep Learning Library](#)



Jason Brownlee November 2, 2016 at 9:10 am #

[How to Save and Load Your Keras Deep](#)

Learning Model

Thanks for the link Yuanliang. I do often see worse results in the dense layer before output.

X

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

The [Deep Learning with Python](#) EBook is where you'll find the **Really Good** stuff.

Happy December 6, 2016 at 1:50 am #

REPLY ↵

>> SEE WHAT'S INSIDE

First of all, thanks to you for making machine learning fun to learn.

I have a query related to drop-outs.

Can we use drop-out even in case we have selected the optimizer as adam and not sgd?

In the examples, sgd is being used and also in the tips section, it has been mentioned "

Use a large learning rate with decay and a large momentum." As far as I see adam does not have the momentum. So what should be the parameter to adam if we use dropouts.

`keras.optimizers.Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0)`



Jason Brownlee December 6, 2016 at 9:52 am #

REPLY ↵

Yes, you can use dropout with other optimization algorithms. I would suggest experimenting with the parameters and see how to balance learning and regularization provided by dropout.

Start Machine Learning

^

REPLY ↵

Never miss a tutorial · January 27, 2017 at 7:58 am #



In a recent post I asked how does one deal with rare events when building a Deep Learning Model? For shallow Machine Learning, I can add a utility or a cost function, but here I'm to see if a more elegant approach has been developed.

Picked for you:

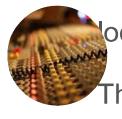


Your First Deep Learning Project in Python

With Keras Step-By-Step

Jason Brownlee January 27, 2017 at 12:26 pm #

REPLY ↵



You may also oversample the rare events, perhaps with data augmentation (randomness) to make them look different.

[How to Grid Search Hyperparameters for](#)

[Deep Learning Models in Python With](#)

There's some discussion of handling imbalanced datasets here.

<http://machinelearningmastery.com/tactics-to-combat-imbalance/>



Regression Tutorial with the Keras Deep Learning Library in Python

Junaid Effendi January 30, 2017 at 8:06 am #



Multi-Class Classification Tutorial with the

[Keras Deep Learning Library](#)

Dropout will work in these cases? It can lower down the complexity of the network. I have been experimenting with the dropout on ANN and now on RNN (LSTM). It seems to work well between the recurrent layers.. but the accuracy remains the same.

[How to Save and Load Your Keras Deep Learning Model](#)

Any comments ?

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?

REPLY ↵

The Deep Learning with Python EBOOK is available now!

where you'll find the **Really Good** stuff.
Hi Junaid,

I have [>> SEE WHAT'S INSIDE <<](#) Ns myself.

Perhaps you need more drop-out and less training to impact the skill or generalization capability of your network.



Shin April 15, 2017 at 11:48 pm #

REPLY ↵

Is the dropout layer stored in the model when it is stored?..

If so why?.. it doesn't make sense to have a dropout layer in a model? besides when training?



Jason Brownlee April 16, 2017 at 9:28 am #

REPLY ↵

It is a function with no weights. There is nothing to store, other than the fact that it exists in a specific point in the network topology.

Start Machine Learning



Never miss a tutorial:

QW April 21, 2017 at 12:53 am #

REPLY ↵



Is it normal practice to fix a random seed for training?

Picked for you:



Jason Brownlee April 21, 2017 at 8:37 am #

Your First Deep Learning Project in Python

REPLY ↵

with Keras Step-By-Step

I see two options, to fix the random seed and make the code reproducible or to run the model n times (30?) and take the mean performance from all runs.



The latter is more robust

How to Search Hyperparameters for

Deep Learning Models in Python With

For more on the stochastic nature of machine learning algorithms

Keras

[http://machinelearningmastery.com/randomness-in-machine-le](http://machinelearningmastery.com/randomness-in-machine-learning/)



Regression Tutorial with the Keras Deep Learning Library in Python

Punit Singh June 11, 2017 at 4:21 am #



Multi-Class Classification Tutorial with the

Keras Deep Learning Library

I am a beginner in Machine Learning and trying to learn new concepts. In this post, I understand all the concepts of dropout, but the use of

from sklearn.model_selection import cross_val_score

from sklearn.model_selection import StratifiedKFold

How to Save and Load Your Keras Deep

Learning Model

sklearn.preprocessing import StandardScaler

sklearn.pipeline import Pipeline

is making code difficult for me to understand. Would you suggest a

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?



Jason Brownlee June 11, 2017 at 8:28 am #

REPLY ↵

where you'll find the **Really Good** stuff.

Yes, you can use keras directly. I offer many tutorials on the topic, try the search feature at the top of the page >> SEE WHAT'S INSIDE



Yitzhak June 18, 2017 at 8:23 pm #

REPLY ↵

thanks Jason, this is so useful !!



Jason Brownlee June 19, 2017 at 8:44 am #

REPLY ↵

You're welcome!



Corina July 10, 2017 at 6:18 pm #

REPLY ↵

Hi Jason,

Start Machine Learning



Thanks for the awesome materials you provide!

Never miss a tutorial:

I have a question, I saw that when using dropout for the hidden layers, you applied it for all of them.

My question is if dropout is applied to the hidden layers then, should it be applied to all of them? Or better yet how do we choose where to apply the dropout?

Thanks ! 😊

Picked for you:



Your First Deep Learning Project in Python

with **Jason Brownlee**

July 11, 2017 at 10:28 am #

REPLY ↗



Great question. I would recommend testing every variation you can think of for your network and see what works best on your specific problem.

How to Get Your Specific Problem for Deep Learning Models in Python With Keras



Dikshika July 10, 2017 at 8:28 pm #

Regression Tutorial with the Keras Deep

Learning Library in Python

My cat dog classifier with Keras is over-fitting for Dog. How



Multi-Class Classification Tutorial with the Keras Deep Learning Library

Jason Brownlee

July 11, 2017 at 10:30 am #



Consider augmentation on images in the cat class in order to help my classifier. I am using the CIFAR-10 dataset.

How to Save and Load Your Keras Deep

Learning Model



Dikshika July 17, 2017 at 2:54 pm #

Loving the Tutorials?

I have already augmented the train data set. But it's not helping. Here is my code snippet.

The [Deep learning API](#) correctly classifies 246 out of 254 dogs and 83 out of 246 cats.

where you'll find the **Really Good** stuff.

[>> SEE WHAT'S INSIDE](#)



Jason Brownlee July 18, 2017 at 8:41 am #

REPLY ↗

Sorry, I don't have good advice off the cuff.



Dikshika July 17, 2017 at 2:55 pm #

REPLY ↗

```
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras import backend as K

# dimensions of our images.
img_width, img_height = 150, 150
```

train_data_dir = r'E:\Interns ! Projects\Positive Integers\CatDogKeras\data\train'

validation_data_dir = r'E:\Interns ! Projects\Positive Integers\CatDogKeras\data\validation'

[Start Machine Learning](#)

^

nb_train_samples = 18000

Never miss a tutorial:
nb_validation_samples = 7000



epoch = 20
batch_size = 16

Picked for you:

if K.image_data_format() == 'channels_first':
 input_shape = (3, img_width, img_height)
else:
 input_shape = (img_width, img_height, 3)



[Your First Deep Learning Project in Python with Keras Step-By-Step](#)

model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=input_shape))

[How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras](#)



model.add(Activation('relu'))

[Regression Tutorial with the Keras Deep Learning Library in Python](#)



model.add(MaxPooling2D(pool_size=(2, 2)))

[Multi-Class Classification Tutorial with the Keras Deep Learning Library](#)



model.add(Flatten())

model.add(Dense(128))

[How to Save and Load Your Keras Deep Learning Model](#)



model.add(Dropout(0.5))

model.add(Dense(1))

model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy',

Loving the Tutorials?

optimizer='rmsprop',

The Deep learning with Python EBook is

where you'll find the **Really Good** stuff.

this is the augmentation configuration we will use for training

train_datagen = ImageDataGenerator(rescale=1. / 255,

>> SEE WHAT'S INSIDE <<

rescale=1. / 255,

shear_range=0.2,

zoom_range=0.2,

horizontal_flip=True)

this is the augmentation configuration we will use for testing:

only rescaling

test_datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = train_datagen.flow_from_directory(

train_data_dir,

target_size=(img_width, img_height),

batch_size=batch_size,

class_mode='binary')

validation_generator = test_datagen.flow_from_directory(

validation_data_dir,

target_size=(img_width, img_height),

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Start Machine Learning

batch_size=batch_size,
Never miss a tutorial:
class_mode='binary')



```
model.fit_generator(  
    train_generator,  
    steps_per_epoch=nb_train_samples // batch_size,  
    epochs=epochs,  
    validation_data=validation_generator,  
    validation_steps=nb_validation_samples // batch_size)  
input("Press enter to exit")
```

 model.save_weights('first_try_v2.h5')
[How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras](#)

 Regression Tutorial with the Keras Deep Learning Library [Dikshika](#) July 1, 2017 at 6:17 pm #

Also, is it possible to get the probability of each training sa

 Multi-Class Classification Tutorial with the Keras Deep Learning Library [Dikshika](#) July 1, 2017 at 6:17 pm #

 [Jason Brownlee](#) July 18, 2017 at 8:42 am #

[How to Save and Load Your Keras Deep Learning Model](#)
Yes, make a probability prediction for each sample at t

[1 results = model.predict_proba\(X\)](#)

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?

 [Dikshika](#) July 18, 2017 at 2:48 pm #

REPLY ↗

where you'll find the *Really Good* stuff.

Thanks a lot. This blog and your suggestions have been really helpful.

>> SEE WHAT'S INSIDE



[Jason Brownlee](#) July 18, 2017 at 5:01 pm #

REPLY ↗

You're welcome, I'm glad to hear that.



[Dikshika](#) July 18, 2017 at 9:05 pm #

I am struck here. I am using binary cross entropy. I want to see probabilities as the actual ones between 0 and 1. But I am getting only maximum probabilities, ie, 0 or 1. For both test and training samples.



[Jason Brownlee](#) July 19, 2017 at 8:22 am #

Start Machine Learning

^



 Thong Bui August 17, 2017 at 4:09 am #

REPLY ↗

Picked for you:

Thanks for the great insights on how dropout works. I have 1 question: what is the difference between adding a dropout layer (like your examples here) and setting the dropout parameter of a layer, for example:

 [Your First Deep Learning Project in Python](#)

[with Keras Step-By-Step](#)

```
model.add(SimpleRNN(..., dropout=0.5))
```

```
model.add(LSTM(..., dropout=0.5))
```

 [How to Optimize Hyperparameters with](#)

[Deep Learning Models in Python With](#)

Keras

 [Regression Tutorial with the Keras Deep](#)

[Learning Library in Python](#)

Jason Brownlee August 17, 2017 at 6:48 am #

Nothing really, they are equivalent.

 [Multi-Class Classification Tutorial with the](#)

[Keras Deep Learning Library](#)

 [Guillaume August 31, 2017 at 11:29 pm #](#)

[How to Save and Load Your Keras Deep](#)

[Learning Model](#)

Thanks for your tutorials 😊

I have a question considering the implementation of the dropout.

Loving the Tutorials?

I am using an LSTM to predict values of a sin wave. Without, the NN is able to catch quite correctly the frequency and the [The Deep Learning with Python](#) EBook is

where you'll find the **Really Good** stuff.

However, implementing dropout like this:

model >> SEE WHAT'S INSIDE

```
model.add(LSTM(neuron, input_shape=(1,1)))
model.add(Dropout(0.5))
model.add(Dense(1))
```

does not lead to the same results as with:

```
model = Sequential()
model.add(LSTM(neuron, input_shape=(1,1), dropout=0.5))
model.add(Dense(1))
```

In the first case, the results are also great. But in the second, the amplitude is reduce by 1/4 of its original value..

Any idea why ?

Thank you !

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE



Jason Brownlee September 1, 2017 at 6:48 am #

REPLY ↗

Start Machine Learning



I would think that they are the same thing, I guess my intuition is wrong.

Never miss a tutorial:

I'm not sure what is going on.



Picked for you:

James September 27, 2017 at 3:15 am #

REPLY ↗



Your First Deep Learning Project in Python
with Keras Step-By-Step

...anks for all your posts, they are great!

My main question is a general one about searching for optimal hyper-parameters; is there a methodology you prefer (i.e. How to Grid Search Hyperparameters for Keras's grid/random search methods)? Or do you generally just plug and chug?

In addition, I found this code online and had a number of questions benefit from:

Regression Tutorial with the Keras Deep Learning Library in Python

Input layer with dimension 1 and hidden layer i with 128 neurons.

model.add(Dense(128, input_dim=1, activation='relu'))
dropout of 20% of the neurons and activation layer.

model.add(Dropout(.2))

model.add(Activation("linear"))

"hidden layer j with 64 neurons plus activation layer.

model.add(Dense(64, activation='relu'))

model.add(Activation("linear"))

Hidden layer k with 64 neurons.

model.add(Dense(64, activation='relu'))

Loving the Tutorials?

Output Layer.

model.add(Dense(1))

The Deep Learning with Python EBook is

where you'll find the Really Good stuff

Model is derived and compiled using mean square error as loss

function, using the Adam gradient descent optimizer.

model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])

>> SEE WHAT'S INSIDE

Training model with train data. Fixed random seed:

numpy.random.seed(3)

model.fit(X_train, y_train, nb_epoch=256, batch_size=2, verbose=2) "

1) I was under the impression that the input layer should be the number of features (i.e. columns – 1) in the data, but this code defines it as 1.

2) defining the activation function twice for each layer seems odd to me, but maybe I am misunderstanding the code, but doesn't this just overwrite the previously defined activation function.

3) For regression problems, shouldn't the last activation function (before the output layer) be linear?

Source: <http://gonzalopla.com/deep-learning-nonlinear-regression/#comment-290>

Thanks again for all the great posts!

James

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Start Machine Learning

Never miss a tutorial:

Jason Brownlee

September 27, 2017 at 5:47 am #

REPLY ↵



f

I'm going to be learning myself. I try to be as systematic as possible, see here:
<https://machinelearningmastery.com/plan-run-machine-learning-experiments-systematically/>

Picked for you:

The code you pasted has many problems.



Your First Deep Learning Project in Python
with Keras Step-By-Step

Azim September 28, 2017 at 2:11 pm #

REPLY ↵



Hi Jason, Thanks for nicely articulated blog. I have a question. Is it that dropout is not applied on the output layer, where we have used softmax function? If so, what is the rationale?

Keras

Regards,



Regression Tutorial with the Keras Deep
Learning Library in Python



Multi-Class Classification Tutorial with the
Keras Deep Learning Library



No, we don't use dropout on output only on input and hidden layers.

The rationale is that we do not want to corrupt the output from the final layer. Instead, we want to corrupt the input features. This is done by applying dropout to the input layer and hidden layers. We can also apply dropout to the output layer if we want to. For example, if we are doing a multi-class classification task, we might want to apply dropout to the output layer to prevent overfitting.



Alex October 2, 2017 at 7:53 am #

REPLY ↵

Loving the Tutorials?
great explanation, Jason!

The Deep Learning with Python EBook is
where you'll find the *Really Good* stuff.

>> SEE WHAT'S INSIDE

October 2, 2017 at 9:40 am #

REPLY ↵

Thanks Alex.



Animesh October 6, 2017 at 10:36 am #

REPLY ↵

How do I plot this code. I have tried various things but get a different error each time.
What is the correct syntax to plot this code?

Jason Brownlee October 6, 2017 at 11:05 am #

REPLY ↵

What do you mean plot the code?

You can run the code by copying it and pasting it into a new file, saving it with a .py extension and running it with the Python interpreter.

Start Machine Learning

^

If you are new to Python, I recommend learning some basics of the language first.
Never miss a tutorial!



Saman December 16, 2017 at 3:13 pm #

REPLY ↵

Picked for you:

hi

I want to know that emerging any two kernels in convolutional layer is dropout technique?
[Your First Deep Learning Project in Python with Keras Step-By-Step](#)



How to Choose Parameters for Deep Learning Models in Python With Keras

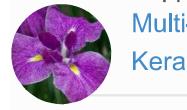
REPLY ↵

Sorry, I don't follow, can you please restate your question?



Regression Tutorial with the Keras Deep Learning Library in Python

amir May 11, 2018 at 7:12 pm #



i run your code on my laptop,, but every time result changes
Multi-Class Classification Tutorial with the Keras Deep Learning Library



Jason Brownlee May 12, 2018 at 6:28 am #
How to Save and Load Your Keras Deep Learning Model

This is a common question that I answer here:

<https://machinelearningmastery.com/faq/single-faq/why-do-i-get-different-results-every-time-i-run-my-deep-learning-code/>

Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?

The Deep Learning with Python EBook is #

REPLY ↵

where you'll find the **Really Good** stuff.

Hi Jason,

I've read >> SEE WHAT'S INSIDE << they are generally pretty good.

I question your statements here:

"You can imagine that if neurons are randomly dropped out of the network during training, that other neurons will have to step in and handle the representation required to make predictions for the missing neurons. This is believed to result in multiple independent internal representations being learned by the network.

The effect is that the network becomes less sensitive to the specific weights of neurons."

I don't think it is correct. The goal is to not create MORE representations but a smaller number of robust representations.

(I've never really seen a specific plausible explanation of co-adaptation. It's all hand-waving.)

Small note: The paper you cite as the "original" paper on dropout is not, it is their 2nd paper. The original one is the one with "co-adaptation" in the title.

Craig Will

Jason Brownlee June 18, 2018 at 3:06 pm #

REPLY ↵

Thanks for the note Craig.

Start Machine Learning

Never miss a tutorial:



at 6:28 am #

REPLY ↵

Jason, thanks for the example. Apparently it has been working for everyone. However, I get the following error

Picked for you:

when I run your code

Your First Deep Learning Project in Python
with Keras Step-By-Step
neError: name 'create_baseline' is not defined

I was hoping you could help me figure this out, as I haven't been able to find anything online nor solve it myself

How to Grid Search Hyperparameters for

Deep Learning Models in Python With

Keras

Jason Brownlee June 21, 2018 at 6:45 am #

Regression Tutorial with the Keras Deep

Learning Library in Python

This might help:

<https://machinelearningmastery.com/faq/single-faq/how-do-i-create-a-multi-class-classification-tutorial-with-the-keras-deep-learning-library/>

Multi-Class Classification Tutorial with the

Keras Deep Learning Library

Guillermo June 21, 2018 at 7:42 am #

How to Save and Load Your Keras Deep

Learning Model

Thanks for the reply! I'm going to try to use it with

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about
services and special offers by email. For more
information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?



Jason Brownlee June 21, 2018 at 4:49 pm #

REPLY ↵

The Deep Learning with Python EBook is

where you'll find the *Glad to hear that!* *Really Good stuff.*

>> SEE WHAT'S INSIDE



Gurudev July 21, 2018 at 4:30 am #

REPLY ↵

Simple and clearly explainedThanks for such articles



Jason Brownlee July 21, 2018 at 6:39 am #

REPLY ↵

I'm glad it helped.



Shabad July 24, 2018 at 4:21 pm #

REPLY ↵

Hi Jason,

Thanks for your articles. I am learning a lot from them.

Start Machine Learning



Btw I ran your code on the same dataset and I got 81.66% (6.89%) accuracy without the dropout layer and a whooping increase to 87.54% (7.07%) accuracy with just dropout at input layer. What I am not able to understand is why the accuracy increased for the same dataset and same model for me and not for you? Is it overfitting in my case? and how do I fix for it



Thank you in advance.

Picked for you:



Your First Deep Learning Project in Python

with Keras Step-By-Step

Jason Brownlee July 25, 2018 at 6:12 am #

REPLY ↵



Nice work.

How to Grid Search Hyperparameters for

Dropout Reduces Overfitting Python With

Keras



Regression Tutorial with the Keras Deep

Patsy Coate July 25, 2018 at 3:27 am #



Multi-Class Classification Tutorial with the

Keras Deep Learning Library



How to Save and Load Your Keras Deep

Learning Model

The idea is to evaluate a model many times and calculate

here:

<https://machinelearningmastery.com/evaluate-skill-deep-learning/>

Loving the Tutorials?

The Deep Learning with Python EBook is



where you'll find the **Really Good** stuff.

Heider August 17, 2018 at 3:04 am #

REPLY ↵

[>> SEE WHAT'S INSIDE](#)

About the dropout on the visible layer, in your example it means that in one batch of 60 images, 12 of them (20%) will be set to zero?

Thank you!



Jason Brownlee August 17, 2018 at 6:36 am #

REPLY ↵

It will be input-variable-wise, e.g. per-pixel, not the whole sample.



Anam Habib August 29, 2018 at 4:26 pm #

REPLY ↵

Hi Jason,

I have observed in my dataset that when i used dropout to reduce the overfitting of deep learning model then it reduces

[Start Machine Learning](#)



the overfitting but it has an effect on the accuracy that it decreases the accuracy of my model. So, how can i increase the accuracy of a model in the process of reducing overfitting?



Picked for you:



Jason Brownlee August 30, 2018 at 6:28 am #

REPLY ↵

Perhaps try different dropout levels?
[Your First Deep Learning Project in Python with Keras Step-By-Step](#)



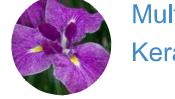
Anam August 30, 2018 at 2:44 pm #

REPLY ↵

Deep Learning Models in Python With Keras
Hy Jason,
If I got 96% training accuracy and 86% testing accuracy the



[Regression Tutorial with the Keras Deep Learning Library in Python](#)



Jason Brownlee August 30, 2018 at 4:53 pm #

REPLY ↵

[Multi-Class Classification Tutorial with the Keras Deep Learning Library](#)
Look at learning curves on train/validation sets



Kern September 8, 2018 at 2:17 am #

REPLY ↵

How to Save and Load Your Keras Deep Learning Model
Can someone please explain the kernel_constraint to me

I get that it works as some kind of regularization.

Acc to keras docs, maxnorm(m) will, if the L2-Norm of your weights exceeds m, scale your whole weight matrix by a factor

Loving the Tutorials?

The [Deep Learning with Python](#) EBook is

where you'll find the **Really Good** stuff.

So, how does it differ from the regular normalisation?

From [here](#) >> SEE WHAT'S INSIDE [-networks-2/#reg:](#)

Max norm constraints. Another form of regularization is to enforce an absolute upper bound on the magnitude of the weight vector for every neuron and use projected gradient descent to enforce the constraint

how does this bound / scaling on weight matrix work?



Jason Brownlee September 8, 2018 at 6:14 am #

REPLY ↵

Good question.

There is weight decay that pushes all weights in a node to be small, e.g. using L1 or L2 or the vector norm (magnitude). Keras calls this kernel regularization I think.

Then there is weight constraint, which imposes a hard rule on the weights. A common example is max norm that forces the vector norm of the weights to be below a value, like 1, 2, 3. Once exceeded all weights in the node are made small enough to meet the constraint.

[Start Machine Learning](#)



It's a subtle difference. Decay is a penalty in the loss function (soft), constraint is a if-then statement in the optimization procedure (hard).



Picked for you:



Ilya September 20, 2018 at 12:03 pm #
Your First Deep Learning Project in Python

REPLY ↗

with Keras Step-By-Step
Thank you for posting, this is very helpful



How to Grid Search Hyperparameters for
Deep Learning Models in Python With
Jason Brownlee September 20, 2018 at 2:28 pm #
Keras

X

I'm happy that it helped.



Regression Tutorial with the Keras Deep
Learning Library in Python



Gavin November 22, 2018 at 6:59 am #
Multi-Class Classification Tutorial with the
Hi Jason Keras Deep Learning Library

Thanks for these awesome tutorials, they are just unbelievable!!



How to Save and Load Your Keras Deep
Learning Model

t

Will "import [moduleX]" not just import the entire library, or are there ways to do this to save memory?

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about
services and special offers by email. For more
information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?

The Deep Learning with Python EBook is

where you'll find the Really Good stuff
Jason Brownlee November 22, 2018 at 2:06 pm #

REPLY ↗

>> SEE WHAT'S INSIDE taught to only import what is required.



ravi gurnatham November 26, 2018 at 10:35 am #

REPLY ↗

Hi Jason,

you mentioned that dropouts are not used while predicting but its not, we use the dropouts as well while predicting.

we simply multiply each output of activation function by the probability rate which we used in every layer.



Jason Brownlee November 26, 2018 at 2:02 pm #

REPLY ↗

Dropout is only used during training.

Start Machine Learning

^

Never miss a tutorial:

Sinan Ozdemir December 5, 2018 at 9:11 am #

REPLY ↗



I applied the above technique and other techniques mentioned in your book to reduce the overfitting which worked fine.

Picked for you: model performed very poorly on a dataset that it hadn't seen before.

Do you think based on your experience, not applying any overfitting reducing methods might be the optimal way to train your dataset? I know this might not be possible to generalize, but do you also think this has something to do with the dataset that we are dealing with. In my case, I am working with healthcare dataset.

Thank you so much for these wonderful tutorials and the books.



[How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras](#)



[Regression Tutorial with the Keras Deep Learning Library in Python](#)

Some ideas:



Perhaps the out of sample dataset is not representative?
[Multi-Class Classification Tutorial with the Keras Deep Learning Library](#)
Perhaps other regularization methods are needed?
Perhaps the training process needs tuning too?



Let me know how you go.
[How to Save and Load Your Keras Deep Learning Model](#)



Hamed December 8, 2018 at 6:53 am #

Loving the Tutorials?
Hello Dear Jason,

The [Deep Learning with Python](#) EBook is
Thanks a lot for your great post.
where you'll find the **Really Good** stuff.

Just I have a question regarding over-fitting.
Could you tell me how can I know my CNN over fitted?
To the [>> SEE WHAT'S INSIDE](#) there is a considerable difference between training accuracy and validation accuracy over-fitting happened.

I am asking this because I designed a CNN and I set the dropout to 0.5.

My training accuracy was around 99% and my maximum validation accuracy was 89% and when I reduced the dropout to 0.3 my training accuracy was fixed but the validation accuracy surprisingly increased to 95%.

I don't know I can trust this accuracy or not.

Do you think over-fitting happened to my CNN or not.



Jason Brownlee December 8, 2018 at 7:14 am #

REPLY ↗

You can review the learning curves during training on the training dataset and validation dataset.

If validation continues to get worse and training continues to get better, you are overfit.

Vandana March 2, 2019 at 5:51 am #

[Start Machine Learning](#)

REPLY ↗

Hi Jason,

Never miss a tutorial:

Can we use drop out before the first convolution layer? In the examples you have shown here, drop out is done prior to the first convolution layer. I found a piece of code where they have used drop out before the 1st convolution layer.



<https://github.com/tzirakis/Multimodal-Emotion-Recognition/blob/master/models.py>

Picked for you:

If I did not interpret this wrong, in this code, inside the 'audio_model', 'net' is having first layer as drop out followed by a conv2D layer.

[Your First Deep Learning Project in Python with Keras Step-By-Step](#)



REPLY ↗

Jason Brownlee March 2, 2019 at 9:36 am #

[How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras](#)



X

[Regression Tutorial with the Keras Deep Learning Library in Python](#)

Shanzeb Haider April 28, 2019 at 6:14 pm #



Hi Jason, setting dropout to zero has any impact on our net

[Multi-Class Classification Tutorial with Keras Deep Learning Library](#)



Jason Brownlee April 29, 2019 at 8:18 am #



A dropout rate of 0.0 will have no effect in Keras.

Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

REPLY ↗

Loving the Tutorials?

Pranith May 5, 2019 at 12:34 am #

The [Deep Learning with Python EBook](#) is

where you'll find the **Really Good** stuff.

Hi, I have trained a sequential model with layers LSTM, Dropout, LSTM, Dropout and a Dense layer. And tuning the hyperparameter [>> SEE WHAT'S INSIDE <](#) out value to be 1. What does it mean? I couldn't make sense out of it.

REPLY ↗

Jason Brownlee May 5, 2019 at 6:31 am #

REPLY ↗

In Keras, a dropout of 1, is full/always dropout, which might not work:

<https://machinelearningmastery.com/how-to-reduce-overfitting-with-dropout-regularization-in-keras/>

JG May 10, 2019 at 7:28 pm #

REPLY ↗

Hi Jason!

I come back to your wise, wide and deep machine learning knowledge!

My questions are related to understand the limits values of dropout rate, so:

1) when dropout rate is set = 0., is it equivalent to not add the dropout layer? Is it correct?

[Start Machine Learning](#)

^

2) when dropout rate is set = 1. , is it equivalent to break the whole network so the model is not longer able to learn? Is it correct?



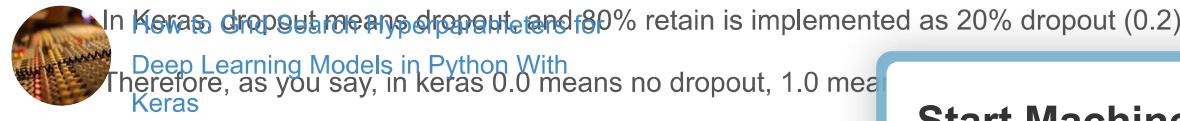
Picked for you:



Your First Deep Learning Project in Python
Jason Brownlee May 11, 2019 at 6:11 am #
with Keras Step-By-Step

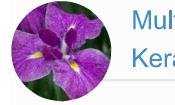
REPLY ↵

In the paper dropout is a “retain” percentage, e.g. 80% means dropout 20%.



Regression Tutorial with the Keras Deep
Learning Library in Python
Jg May 11, 2019 at 6:51 pm #

Thks Jason ! it is more clear now!



Multi-Class Classification Tutorial with the
Keras Deep Learning Library



Jason Brownlee May 12, 2019 at 6:39 am #
How to Save and Load Your Keras Deep
Learning Model

Very happy to hear that.



Loving the Tutorials?
Vanessa December 5, 2019 at 11:07 pm #

REPLY ↵

The Deep Learning with Python EBook is
Hi Jason,
where you'll find the **Really Good** stuff.

Where can I get the data you used?

[>> SEE WHAT'S INSIDE](#)

Or could you tell me what's the type of data, as I'm getting some errors, and I don't know why. Specially in the way I provide the y to the model, it's complaining about dimensions.

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about
services and special offers by email. For more
information, see the [Privacy Policy](#).

START MY EMAIL COURSE



Jason Brownlee December 6, 2019 at 5:17 am #

REPLY ↵

All datasets are available here:

<https://github.com/brownlee/Datasets>



Patxi February 8, 2020 at 3:54 am #

REPLY ↵

Hi, I have used ModelCheckpoint, the same epoch and batchsize. The best configuration found is at the 86th epoch.

I have plotted the loss function and accuracy and I see that there is an important gap between the train and the test curve.
The loss curve corresponding to the test data increases.

Start Machine Learning

^

What does it mean?
Never miss a tutorial:

Thanks!!



Picked for you:

Jason Brownlee February 8, 2020 at 7:16 am #

REPLY ↵



Your First Deep Learning Project in Python

This will help:

with Keras Step-By-Step

<https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>



How to Grid Search Hyperparameters for

Deep Learning Models in Python With

Patxi February 10, 2020 at 1:24 am #

Keras

Hi Jason,



Regression Tutorial with the Keras Deep

Thank you for your kick help! According to the loss function
Learning Library in Python
corresponding to the unrepresentative data set.



I have reduced the train-test Split from 30% to 20%. Accuracy

[Multi-Class Classification Tutorial with the](#)

Keras Deep Learning Library



How to Save and Load Your Keras Deep

Jason Brownlee

MACHINE LEARNING MASTERY

February 10, 2020 at 6:32 am #

Perhaps try 50/50 split?

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about
services and special offers by email. For more
information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?

The Deep Learning with Python EBook is

where you'll find the **Really Good** stuff.

Hi Jason,

>> SEE WHAT'S INSIDE

You for your patience. The results remains the same: the gap between the train and test
loss functions remains, and there is a small drop of accuracy (now it is 86%)

If the training set is smaller (drop from 80% to 50%), how can it be more representative?



Jason Brownlee February 10, 2020 at 1:19 pm #

Perhaps the model is slightly overfit the training dataset?



Patxi February 15, 2020 at 3:16 am #

REPLY ↵

Hi Jason,

This is the model that I'm using:

Start Machine Learning



create model
Never miss a tutorial:
model = Sequential()


model.add(Dense(60, input_dim=59, kernel_initializer='uniform', kernel_constraint=max_norm(3.), activation='relu'))
model.add(Dense(30, kernel_initializer='uniform', kernel_constraint=max_norm(1.), activation='relu'))
model.add(Dropout(0.2))

Picked for you:

[Your First Deep Learning Project in Python](#)
Compile model
with Keras Step-By-Step
Epochs = 150

learning_rate = 0.1
decay_rate = learning_rate / epochs

[How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras](#)
sgd = SGD(lr=learning_rate, momentum=momentum, decay=decay_rate)
model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=[


[Regression Tutorial with the Keras Deep Learning Library](#)
filepath='weights.best.hdf5',
monitor='val_accuracy',
callbacks_list = [checkpoint]


[Multi-Class Classification Tutorial with the Keras Deep Learning Library](#)
try:
model.fit(X_train, Y_train, validation_data=(X_test, Y_test),
callbacks=callbacks_list, verbose=0)


[How to Save and Load Your Keras Deep Learning Model](#)
Jason Brownlee February 15, 2020 at 6:35 am #

Well done!

Loving the Tutorials?

The Deep Learning with Python EBook is
where you'll find the **Really Good** stuff.

Nitin Sharma February 17, 2020 at 2:50 pm #

REPLY ↗

[>> SEE WHAT'S INSIDE](#)

Can you please let us know, while we have already trained a neural network with dropouts, how that will be taken care when we are using the same model weights predicting (Serving Model) ?


Jason Brownlee February 18, 2020 at 6:16 am #

REPLY ↗

Dropout is only used during training as far I recall.


Nitin Sharma February 18, 2020 at 4:35 pm #

REPLY ↗

Hi Jason, thanks for the reply, but let's say we have 6 neurons dropout and we have a probability of 0.5 so the weightage of signals going down in the next layer will be 1/5 strength , but when we have the prediction network, it will have all 6 neurons passing on the signal, it can cause an additional input magnitude increase, which in turn will have an impact in subsequent layers. So will the output be accordingly ?

[Start Machine Learning](#)



Start Machine Learning

You can master applied Machine Learning without **math or fancy degrees**.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

[START MY EMAIL COURSE](#)

Never miss a tutorial:



wnlee February 19, 2020 at 7:57 am #

REPLY ↵

No, training under dropout causes the nodes to share the load, balance out.

Picked for you:

Recall that dropout is probabilistic.



Your First Deep Learning Project in Python with Keras Step-By-Step

Amit January 26, 2022 at 5:21 am #

REPLY ↵

How to Grid Search Hyperparameters for Deep Learning Models in Python With

Keras



Regression Tutorial with the Keras Deep Learning Library in Python Rohit Khanna April 29, 2020 at 7:42 pm #



How to implement dropout on the test set during prediction

Basically, I want to model the uncertainty in the model prediction or

Multi-Class Classification Tutorial with the Keras Deep Learning Library



Jason Brownlee April 30, 2020 at 6:42 am # How to Save and Load Your Keras Deep



Dropout is only used during training.

I believe you can force dropout during testing. I might have an

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?

The Deep Learning with Python EBook is tiairr June 12, 2020 at 9:23 am # where you'll find the **Really Good** stuff.

REPLY ↵

Hi Jason how to use dropout with 3 different dataset?

>> SEE WHAT'S INSIDE



Jason Brownlee June 12, 2020 at 11:13 am #

REPLY ↵

Sorry, I don't understand your question, can you please elaborate?



manju badiger August 17, 2020 at 5:08 pm #

REPLY ↵

Hi Jason can we build a Customized drop out layer? if we build customized drop out layer then can we add or change functions in customized drop out layer? So that we can compare with the regular drop out layer ? is it practically possible? or better we can use a regular drop out layer can you give me some good explanation? it would help more



Jason Brownlee August 18, 2020 at 5:59 am #

REPLY ↵

Start Machine Learning

^

I don't see why not.

Never miss a tutorial:

Sorry, I don't have examples of creating custom dropout layers. I recommend checking the API and



Picked for you:



Vishal S. September 28, 2020 at 5:21 am #
Your First Deep Learning Project in Python

with Keras Step-By-Step

Here's a customized dropout, very simple to implement. See CS231 Stanford lec online and github for more specifics.

REPLY ↗

```
1 class DropoutLayer(nn.Module):
2     def __init__(self, p):
3         super().__init__()
4         self.p = p
5
6     def forward(self, input):
7         if self.training:
8             u1 = (torch.rand(*input.shape) < self.p) / s
9             output = input * u1
10
11         return output
12     else:
13         return input
Keras Deep Learning Library
```



How to Save and Load Your Keras Deep
Jason Brownlee September 28, 2020 at 6:28 am #

It's for torch, not keras.

Why are you sharing it?

Loving the Tutorials?

The Deep Learning with Python EBook is
where you'll find the **Really Good** stuff.
walid October 24, 2020 at 8:02 pm #

REPLY ↗

>> SEE WHAT'S INSIDE

i'm trying to test some of your codes with my data. i tried to use dropout in my code but unfortunately my validation loss is lower than my training loss despite the mse for train and test seems the same 0.007 and 0.008 (without drop).

also the mse with dropout is more than without dropout as mentioned in the figures below.

<https://ibb.co/ZKxB7fL>

<https://ibb.co/N3vpDCZ>

i tried to fund the reasons like mentioned here:

<https://www.pyimagesearch.com/2019/10/14/why-is-my-validation-loss-lower-than-my-training-loss/>

reason1: Regularization applied during training, but not during validation/testing (normally and by default the dropout is not used in validation/testing)

reason2: Training loss is measured during each epoch while validation loss is measured after each epoch

reason3: The validation set may be easier than the training set (or there may be leaks)

I think in my case that the reason 2 is the more logical.

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about
services and special offers by email. For more
information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Start Machine Learning

X

^

X_train=174200 samples

Never miss a tutorial:

X_test=85800



the code:

Picked for you:

X = dataset[:,0:20].astype(float)

Use [F120 Deep Learning Project in Python](#)

[StandardScaler Step-by-Step](#)

X = scaler.fit_transform(X)

y = scaler.fit_transform(y)

[How to Grid Search Hyperparameters for](#)

[Deep Learning Models in Python With](#)

[Keras](#)

print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

define the keras model

[Sequential\(\) Tutorial with the Keras Deep](#)

[Layer Dense\(20\) in Python](#) dim=20,kernel_initializer='normal'))

model.add(LeakyReLU(alpha=0.1))

#model.add(Dropout(0.2))

[Multi-Class Classification Tutorial with the](#)

[Keras Deep Learning Library](#)

#model.add(Dropout(0.2))

model.add(Dense(2, activation='linear'))

[How to Save and Load Your Keras Deep](#)

[Learning Model](#)

Compile the keras model

model.compile(loss='mean_squared_error', optimizer=opt, metrics=

fit the keras model on the dataset

history=model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=25, verbose=0)

evaluate the model

_ = model.evaluate(X_train, y_train, verbose=0)

_ = model.evaluate(X_test, y_test, verbose=0)

print('Train: %.2f Test: %.2f' % (train_mse, test_mse))

#plot loss

[>> SEE WHAT'S INSIDE](#)

pyplot.title('Loss / Mean Squared Error')

pyplot.plot(history.history['loss'], label='train')

pyplot.plot(history.history['val_loss'], label='test')

pyplot.legend()

pyplot.show()

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about
services and special offers by email. For more
information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Jason Brownlee October 25, 2020 at 7:00 am #

REPLY ↗

This is common when your validation dataset is too small or not representative of the training dataset.

walid October 25, 2020 at 1:44 pm #

REPLY ↗

Start Machine Learning

^

thank you for your answer.

Never miss a tutorial!

i increase the test_size to 50% but it's the same thing. the printed train_mse and test_mse are identical(seems good).

why the plot of loss function of the test loss is lower than the train?

what you advise me?

thanks

Picked for you:



Your First Deep Learning Project in Python

with Keras Step-By-Step

Jason Brownlee

October 26, 2020 at 6:48 am #

REPLY ↵



Perhaps conform you are plotting what you believe you are plotting (debug).

[How to Grid Search Hyperparameters for](#)

Perhaps try an alternate sample of your dataset.

[Deep Learning Models in Python With](#)

Perhaps try an alternate model.

Keras



Regression Tutorial with the Keras Deep

Learning Library in Python

wand

October 27, 2020 at 3:25 pm #



thank you jason for your answer.

[Multi-Class Classification Tutorial with the](#)

i tried to change the dataset.

[Keras Deep Learning Library](#)



i tried to increase the number of nodes in the hidden layer to

i also reduce the model, just i kept one hidden layer:

[How to Save and Load Your Keras Deep](#)

Learning Model

model = Sequential()

model.add(Dense(20, input_dim=20, kernel_constraint=maxnorm(3)))

model.add(LeakyReLU(alpha=0.1))

model.add(Dropout(0.2))

Loving the Tutorials?

The [Deep Learning with Python](#) EBook is

opt = Adadelta(lr=0.01)

where you'll find the **Really Good** stuff.

compile the keras model

>> SEE WHAT'S INSIDE >> model.compile(loss='mean_squared_error', optimizer=opt, metrics=['mse'])

....

the obtained result :Train: 0.002, Test: 0.003

but the graph of loss function is unchanged. always the validation loss is lower than my training loss.

the code to plot the loss function is:

```
pyplot.title('Loss / Mean Squared Error')
pyplot.plot(history.history['loss'], label='train')
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
pyplot.show()
```

Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about
services and special offers by email. For more
information, see the [Privacy Policy](#).

START MY EMAIL COURSE



Jason Brownlee October 28, 2020 at 6:42 am #

REPLY ↵

Start Machine Learning

^



Picked for you: [October 27, 2020 at 5:44 pm #](#)

REPLY ↵



thank you jason.
Your First Deep Learning Project in Python
with Keras Step-By-Step
codes are very helpful.

i found the error .



How to Grid Search Hyperparameters for
Deep Learning Models in Python With
Keras
model.fit(X, y, validation_data=(X_test, y_test), epochs=25,)

X,y==>X_train ,y_train



Regression Tutorial with the Keras Deep
Learning Library in Python



Jason Brownlee October 28, 2020 at 6:43 am #
Hi-Class Classification Tutorial with the
Keras Deep Learning Library



How to Save and Load Your Keras Deep
Learning Model

Sunmesh December 4, 2020 at 5:12 pm #

Hi Jason,

Thanks for the clear explanation! I do have a doubt though. In one of the models that I have created, I'm getting pretty good (~99%) validation accuracy with a minimalistic baseline CNN (just 4 layers of conv+maxpool). However, when I increase the number of layers, the validation does an early-stopping as it tends to plateau. Does this mean the network is going deeper and studying things that aren't positively contributing to the model? Since it isn't a case of overfitting (validation and training goes pretty much hand in hand), I'm not inclined to use dropout as well. Does it make sense to augment >> SEE WHAT'S INSIDE << see if the accuracy increases? With an accuracy of 99.xxx, I'm not sure if there is a real need to do so. Would like to hear your thoughts on this.

Regards,
James



Jason Brownlee December 5, 2020 at 8:04 am #

REPLY ↵

You're welcome.

Changing the capacity of the model will often require a corresponding adjustment of the learning hyperparameters (learning rate, batch size, etc.)

Try dropout and see. Try augmentation and see. Experiments are cheap.

An accuracy of 99% on a hold out dataset might suggest your prediction task is trivial and might be solved using simpler methods.

Start Machine Learning



Never miss a tutorial:

Sumesh December 5, 2020 at 12:13 pm #

REPLY ↵



Tha Jas Are there simpler methods when it comes to running classification for images? I typically opt for CNNs when it comes to images. I could try doing the plain regression or Random Forests to see how it fares though.

Picked for you:



Your First Deep Learning Project in Python with Keras Step-By-Step



Jason Brownlee December 5, 2020 at 1:22 pm #

REPLY ↵



How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras



Sumesh December 5, 2020 at 3:33 pm #
Regression Tutorial with the Keras Deep Learning Library in Python



Multi-Class Classification Tutorial with the Keras Deep Learning Library

Angela July 28, 2021 at 10:35 am #



Hi Jason,
How to Save and Load Your Keras Deep Learning Model

Start Machine Learning

You can master applied Machine Learning without math or fancy degrees.

Find out how in this free and practical course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE



Loving the Tutorials?

Jason Brownlee July 29, 2021 at 5:08 am #

REPLY ↵

The Deep Learning with Python EBook is Sure, you can add layers to the network, but I think you will have to re-connect all layers in the network (a where you'll find the **Really Good** stuff. pain).

>> SEE WHAT'S INSIDE

Leave a Reply

Name (required)

Email (will not be published) (required)

Start Machine Learning

^



Welcome!

Picked for you I'm Jason Brownlee PhD

and I help developers get results with machine learning.

Your First Deep Learning Project in Python

with Keras Step-By-Step



How to Grid Search Hyperparameters for
Deep Learning Models in Python With
Keras



Regression Tutorial with the Keras Deep
Learning Library in Python



Multi-Class Classification Tutorial with the
Keras Deep Learning Library



How to Save and Load Your Keras Deep
Learning Model



Start Machine Learning

You can master applied Machine Learning
without math or fancy degrees.

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about
services and special offers by email. For more
information, see the [Privacy Policy](#).

START MY EMAIL COURSE



Loving the Tutorials?

The [Deep Learning with Python](#) EBook is
where you'll find the **Really Good** stuff.

[>> SEE WHAT'S INSIDE](#)





Picked for you:



[Your First Deep Learning Project in Python with Keras Step-By-Step](#)



[How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras](#)



[Regression Tutorial with the Keras Deep Learning Library in Python](#)



[Multi-Class Classification Tutorial with the Keras Deep Learning Library](#)



[How to Save and Load Your Keras Deep Learning Model](#)

Start Machine Learning

You can master applied Machine Learning **without math or fancy degrees.**

Find out how in this *free* and *practical* course.

Email Address

I consent to receive information about services and special offers by email. For more information, see the [Privacy Policy](#).

START MY EMAIL COURSE

Loving the Tutorials?

The [Deep Learning with Python](#) EBook is where you'll find the ***Really Good*** stuff.

[>> SEE WHAT'S INSIDE](#)