Regression Tutorial with the Keras Deep Learning Library in Python

by **Jason Brownlee** on June 9, 2016 in **Deep Learning**

Keras is a deep learning library that wraps the efficient numerical libraries Theano and TensorFlow.

In this post you will discover how to develop and evaluate neural network models using Keras for a regression problem.

After completing this step-by-step tutorial, you will know:

- How to load a CSV dataset and make it available to Keras.
- How to create a neural network model with Keras for a regression problem.
- How to use scikit-learn with Keras to evaluate models using cross-validation.
- How to perform data preparation in order to improve skill with Keras models.

- How to tune the network topology of models with Keras.

Let's get started.

- **Update Mar/2017**: Updated example for Keras 2.0.2, TensorFlow 1.0.1 and Theano 0.9.0.

Regression Tutorial with Keras Deep Learning Library in Python
Photo by Salim Fadhley, some rights reserved.

1. Problem Description

The problem that we will look at in this tutorial is the Boston house price dataset.

You can download this dataset and save it to your current working directly with the file name housing

Get Your Start in Machine Learning

The dataset describes 13 numerical properties of houses in Boston suburbs and is concerned with modeling the price of houses in those suburbs in thousands of dollars. As such, this is a regression predictive modeling problem. Input attributes include things like crime rate, proportion of nonretail business acres, chemical concentrations and more.

This is a well-studied problem in machine learning. It is convenient to work with because all of the input and output attributes are numerical and there are 506 instances to work with.

Reasonable performance for models evaluated using Mean Squared Error (MSE) are around 20 in squared thousands of dollars (or $4,500 if you take the square root). This is a nice target to aim for with our neural network model.

2. Develop a Baseline Neural Network Model

In this section we will create a baseline neural network model for the regression problem.

Let's start off by including all of the functions and objects we will need for this tutorial.

```
1 import numpy
2 import pandas
3 from keras.models import Sequential
4 from keras.layers import Dense
5 from keras.wrappers.scikit_learn import KerasRegressor
6 from sklearn.model_selection import cross_val_score
7 from sklearn.model_selection import KFold
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
```

We can now load our dataset from a file in the local directory.

The dataset is in fact not in CSV format in the UCI Machine Learning Repository, the attributes are instead separated by whitespace. We can load this easily using the pandas library. We can then split the input (X) and output (Y) attributes so that they are easier to model with Keras and scikit-learn.

```
# load dataset
dataframe = pandas.read_csv("housing.csv", delim_whitespace=True, header=None)
dataset = dataframe.values
# split into input (X) and output (Y) variables
X = dataset[:,0:13]
Y = dataset[:,13]
```

We can create Keras models and evaluate them with scikit-learn by using handy wrapper objects pr use scikit-learn excels at evaluating models and will allow us to use powerful data preparation and mode

The Keras wrappers require a function as an argument. This function that we must define is respons evaluated.

Below we define the function to create the baseline model to be evaluated. It is a simple model that same number of neurons as input attributes (13). The network uses good practices such as the recti activation function is used for the output layer because it is a regression problem and we are interes transform.

The efficient ADAM optimization algorithm is used and a mean squared error loss function is optimiz evaluate the performance of the model. It is a desirable metric because by taking the square root giv the context of the problem (thousands of dollars).

```
# define base model
def baseline_model():
    # create model
    model = Sequential()
    model.add(Dense(13, input_dim=13, kernel_initializer='normal', activation='relu'))
    model.add(Dense(1, kernel_initializer='normal'))
    # Compile model
    model.compile(loss='mean_squared_error', optimizer='adam')
    return model
```

The Keras wrapper object for use in scikit-learn as a regression estimator is called KerasRegressor. We create an instance and pass it both the name of the function to create the neural network model as well as some parameters to pass along to the fit() function of the model later, such as the number of epochs and batch size. Both of these are set to sensible defaults.

We also initialize the random number generator with a constant random seed, a process we will repeat for each model evaluated in this tutorial. This is an attempt to ensure we compare models consistently.

```
# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
# evaluate model with standardized dataset
estimator = KerasRegressor(build_fn=baseline_model, nb_epoch=100, batch_size=5, verbose=0)
```

The final step is to evaluate this baseline model. We will use 10-fold cross validation to evaluate the model.

```
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(estimator, X, Y, cv=kfold)
print("Results: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

Running this code gives us an estimate of the model's performance on the problem for unseen data. ng the average and standard deviation (average variance) across all 10 folds of the cross validation eva

```
Baseline: 31.64 (26.82) MSE
```


3. Modeling The Standardized Dataset

An important concern with the Boston house price dataset is that the input attributes all vary in their s

It is almost always good practice to prepare your data before modeling it using a neural network mod

Continuing on from the above baseline model, we can re-evaluate the same model using a standardized version of the input dataset.

We can use scikit-learn's Pipeline framework to perform the standardization during the model evaluation process, within each fold of the cross validation. This ensures that there is no data leakage from each testset cross validation fold into the training data.

The code below creates a scikit-learn Pipeline that first standardizes the dataset then creates and evaluate the baseline neural network model

```
# evaluate model with standardized dataset
numpy.random.seed(seed)
estimators = []
estimators.append(('standardize', StandardScaler()))
estimators.append(('mlp', KerasRegressor(build_fn=baseline_model, epochs=50, batch_size=5, verbose=0)))
pipeline = Pipeline(estimators)
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(pipeline, X, Y, cv=kfold)
print("Standardized: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

Running the example provides an improved performance over the baseline model without standardized data, dropping the error.

```
Standardized: 29.54 (27.87) MSE
```

A further extension of this section would be to similarly apply a rescaling to the output variable such as normalizing it to the range of 0-1 and use a Sigmoid or similar activation function on the output layer to narrow output predictions to the same range.

4. Tune The Neural Network Topology

There are many concerns that can be optimized for a neural network model.

Perhaps the point of biggest leverage is the structure of the network itself, including the number of la

In this section we will evaluate two additional network topologies in an effort to further improve the p deeper and a wider network topology.

4.1. Evaluate a Deeper Network Topology

One way to improve the performance a neural network is to add more layers. This might allow the m es embedded in the data.

In this section we will evaluate the effect of adding one more hidden layer to the model. This is as easy as defining a new function that will create this deeper model, copied from our baseline model above. We can then insert a new line after the first hidden layer. In this case with about half the number of neurons.

```
# define the model
def larger_model():
    # create model
    model = Sequential()
```

```
    model.add(Dense(13, input_dim=13, kernel_initializer='normal', activation='relu'))
    model.add(Dense(6, kernel_initializer='normal', activation='relu'))
    model.add(Dense(1, kernel_initializer='normal'))
    # Compile model
    model.compile(loss='mean_squared_error', optimizer='adam')
    return model
```

Our network topology now looks like:

```
13 inputs -> [13 -> 6] -> 1 output
```

We can evaluate this network topology in the same way as above, whilst also using the standardization of the dataset that above was shown to improve performance.

```
numpy.random.seed(seed)
estimators = []
estimators.append(('standardize', StandardScaler()))
estimators.append(('mlp', KerasRegressor(build_fn=larger_model, epochs=50, batch_size=5,
pipeline = Pipeline(estimators)
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(pipeline, X, Y, cv=kfold)
print("Larger: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

Running this model does show a further improvement in performance from 28 down to 24 thousand

```
Larger: 22.83 (25.33) MSE
```

4.2. Evaluate a Wider Network Topology

Another approach to increasing the representational capability of the model is to create a wider netw

In this section we evaluate the effect of keeping a shallow network architecture and nearly doubling t

Again, all we need to do is define a new function that creates our neural network model. Here, we have increased the number of neurons in the hidden layer compared to the baseline model from 13 to 20.

```
# define wider model
def wider_model():
    # create model
    model = Sequential()
    model.add(Dense(20, input_dim=13, kernel_initializer='normal', activation='relu'))
    model.add(Dense(1, kernel_initializer='normal'))
```

```
    # Compile model
    model.compile(loss='mean_squared_error', optimizer='adam')
    return model
```

Our network topology now looks like:

```
13 inputs -> [20] -> 1 output
```

We can evaluate the wider network topology using the same scheme as above:

```
numpy.random.seed(seed)
estimators = []
estimators.append(('standardize', StandardScaler()))
estimators.append(('mlp', KerasRegressor(build_fn=wider_model, epochs=100, batch_size=5, verbose=0)))
pipeline = Pipeline(estimators)
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(pipeline, X, Y, cv=kfold)
print("Wider: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

Building the model does see a further drop in error to about 21 thousand squared dollars. This is not

```
Wider: 21.64 (23.75) MSE
```

It would have been hard to guess that a wider network would outperform a deeper network on this pr f empirical testing when it comes to developing neural network models.

Summary

In this post you discovered the Keras deep learning library for modeling regression problems.

Through this tutorial you learned how to develop and evaluate neural network models, including:

- How to load data and develop a baseline model.
- How to lift performance using data preparation techniques like standardization.
- How to design and evaluate networks with different varying topologies on a problem.

Do you have any questions about the Keras deep learning library or about this post? Ask your questions in the comments and I will do my best to answer.

About Jason Brownlee

Dr. Jason Brownlee is a husband, proud father, academic researcher, author, professional developer and a machine learning practitioner. He is dedicated to helping developers get started and get good at applied machine learning.
Learn more.

View all posts by Jason Brownlee →

‹ Save and Load Machine Learning Models in Python with scikit-learn

Your First Machine Learning Project in Python Step-By-Step ›

269 Responses to *Regression Tutorial with the Keras Deep Learning Library in Python*

Gautam Karmakar June 25, 2016 at 4:19 pm #

REPLY

Hi did you handle string variables in cross_val_score module?

Jason Brownlee June 26, 2016 at 6:00 am #

The dataset is numeric, no string values.

Ramya December 9, 2017 at 2:34 am #

How do we handle string values

Jason Brownlee December 9, 2017 at 5:43 am #

Great question, I have a whole section on the topic:
https://machinelearningmastery.com/start-here/#nlp

Erika December 12, 2017 at 7:22 am #

REPLY

One hot encoder is an option.

Paul June 30, 2016 at 2:28 am #

REPLY

Hi Jason,

Great tutorial(s) they have been very helpful as a crash course for me so far.

Is there a way to have the model output the estimated Ys in this example? I would like to evaluate the model a little more directly while I'm still learning Keras.

Thanks!

Jason Brownlee June 30, 2016 at 6:48 am #

Hi Paul, you can make predictions by calling model.predict()

Rahul November 22, 2016 at 7:23 pm #

Hey Paul,
How are you inserting the function model.predict() in the above code to run in on test data? Please l

DataScientistPM May 9, 2017 at 11:23 pm #

REPLY

Hi,

Is this how you insert predict and then get predictions in the model?

```
def mymodel():
model = Sequential()
```

```
model.add(Dense(13, input_dim=13, kernel_initializer='normal', activation='relu'))
model.add(Dense(6, kernel_initializer='normal', activation='relu'))
model.add(Dense(1, kernel_initializer='normal'))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(X,y, nb_epoch=50, batch_size=5)
predictions = model.predict(X)
return model
```

I actually want to write the predictions in a file?

Chris July 23, 2016 at 6:24 am #

REPLY

Hi, Great post thank you, Could you please give a sample on how to use Keras LSTM layer for

Thanks

Jason Brownlee July 23, 2016 at 1:29 pm #

Thanks Chris.

You can see an example of LSTMs on this dataset here:
http://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-ke

Chris July 25, 2016 at 9:21 pm #

REPLY

That was Awesome, thank you Json.

Jason Brownlee July 26, 2016 at 5:56 am #

You're welcome Chris.

Marc Huertas-Company July 28, 2016 at 4:50 am #

REPLY

Hi,

Thanks for the tutorial. I have a regression problem with bounded outputs (0-1). Is there an opitmal way to deal with this?
Thanks!
Marc

Jason Brownlee July 28, 2016 at 5:49 am #

Hi Marc, I think a linear activation function on the output layer will be just fun.

James August 5, 2016 at 6:50 am #

This is a good example. However, it is not relevant to Neural networks when over-fitting is cons
inside the fit() function to monitor over-fitting status. Moreover, early stopping can be used based on the
applicable for large data compared to the number of all weights of input and hidden nodes.

Jason Brownlee August 5, 2016 at 8:04 am #

REPLY

Great feedback, thanks James I agree.

It is intended as a good example to show how to develop a net for regression, but the dataset is indeed a bit small.

Amir October 24, 2016 at 11:08 am #

REPLY

Thanks Jason and James! A few questions (and also how to implement in python):

1) How can we monitor the over-fitting status in deep learning

2) how can we include the cross-validation process inside the fit() function to monitor the over-fitting status

3) How can we use early stopping based on the internal validation step

4) Why is this example only applicable for a large data set? What should we do if the data set is small?

Jason Brownlee October 25, 2016 at 8:21 am #

REPLY

Great questions Amir!

1. Monitor the performance of the model on the training and a standalone validation dataset. validation set goes down and skill on training goes up or keeps going up, you are overlearni
2. Cross validation is just a method for estimating the performance of a model on unseen da and your model, it does not go inside fit.
3. Monitor skill on a validation dataset as in 1, when skill stops improving on the validation s
4. Generally, neural nets need a lot more data to train than other methods.

Here's a tutorial on checkpointing that you can use to save "early stopped" models:
http://machinelearningmastery.com/check-point-deep-learning-models-keras/

Salem August 5, 2016 at 2:44 pm #

REPLY

Hi,
How once can predict new data point on a model while during building the model the training data has been standardised using sklearn.

Jason Brownlee August 6, 2016 at 2:09 pm #

You can save the object you used to standardize the data and later reuse it to standardize new data before making a prediction. This might be the MinMaxScaler for example.

Guy August 25, 2016 at 10:52 am #

REPLY

Hi,

I am not using the automatic data normalization as you show, but simply compute the mean and stdev for each feature (data column) in my training data and manually perform zscore ((data – mean) / stdev). By normalization I mean bringing the data to 0-mean, 1-stdev. I know there are several names for this process but let's call it "normalization" for the sake of this argument.

So I've got 2 questions:

1) Should I also normalize the output column? Or just leave it as it is in my train/test?

2) I take the mean, stdev for my training data and use them to normalize the test data. But it seems that , and no matter that each mini-batch is balanced (has the same distribution of output values). What am I r

Jason Brownlee August 26, 2016 at 10:30 am #

Hi Guy, yeah this is normally called standardization.

Generally, you can get good results from applying the same transform to the output column. Try and e performance measure, you may need to be careful with the interpretation of the results as the scale

Yep, this is a common problem. Ideally, you want a very large training dataset to effectively estimate these values. You could try using bootstrap on the training dataset (or within a fold of cross validation) to create a more robust estimate of these terms. Bootstrap is just the repeated subsampling of your dataset and estimation of the statistical quantities, then take the mean from all the estimates. It works quite well.

I hope that helps.

Pranith Kumar Pola September 2, 2016 at 3:52 am #

Get Your Start in Machine Learning

Hello Jason,

How should i load multiple finger print images into keras.

Can you please advise further.

Best Regards,
Pranith

REPLY

Luciano September 10, 2016 at 3:32 am #

Hi Jason, great tutorial. The best out there for free.

Can I use R^2 as my metric? If so, how?

Regards

Jason Brownlee September 10, 2016 at 7:11 am #

Thanks Luciano.

You can use R^2, see this list of metrics you can use:
http://scikit-learn.org/stable/modules/model_evaluation.html

REPLY

sumon October 1, 2016 at 2:38 am #

shouldn't results.mean() print accuracy instead of error?

REPLY

Jason Brownlee October 1, 2016 at 8:03 am #

Get Your Start in Machine Learning

We summarize error for regression problems instead of accuracy (x/y correct). I hope that helps.

David October 19, 2016 at 7:34 pm #

REPLY

Hi,

if I have a new dataset, X_new, and I want to make a prediction, the model.predict(X_new) shows the error ”NameError: name model is not defined’ and estimator.predict(X_test) shows the error message ‘KerasRegressor object has no attribute model’.

Do you have any suggestion? Thanks.

Jason Brownlee October 20, 2016 at 8:35 am #

Hi David, this post will get you started with the lifecycle of a Keras model:
http://machinelearningmastery.com/5-step-life-cycle-neural-network-models-keras/

Heinz Hemken January 3, 2017 at 8:23 am #

Hi Jason,

That page does not use KerasRegressor. How can we save the model and its weights in the coc

Thanks!

Avhirup October 22, 2016 at 11:19 pm #

REPLY

I’m getting more error by standardizing dataset using the same seed.What must be the reason behind it?

Get Your Start in Machine Learning

Avhirup October 22, 2016 at 11:25 pm #

also deeper network topology seems not to help .It increases the MSE

REPLY

Avhirup October 22, 2016 at 11:32 pm #

deeper network without standardisation gives better results.Somehow standardisation is adding more noise

REPLY

Michele Vascellari November 2, 2016 at 9:28 pm #

Hey great tutorial. I tried to use both Theano and Tensorflow backend, but I obtained very differ
obtained results very similar to you, but with Tensorflow I have MSE larger than 100.

Do you have any clue?

Michele

Jason Brownlee November 3, 2016 at 7:59 am #

Great question Michele,

Off the cuff, I would think it is probably the reproducibility problems we are seeing with Python deep
the random number generators used to get repeatable results.

I would not rule out a bug in one implementation or another, but I would find this very surprising for such a simple network.

Kenny November 7, 2016 at 4:21 pm #

REPLY

Get Your Start in Machine Learning

hi, i have a question about sklearn interface.
although we sent the NN model to sklearn and evaluate the regression performance, how can we get the exactly predictions of the input data X, like usually when we r using Keras we can call the model.predict(X) function in keras. btw, I mean the model is in sklearn right?

Jason Brownlee November 8, 2016 at 9:49 am #

REPLY

Hi Kenny,

You can use the sklearn model.predict() function in the same way to make predictions on new input data.

Silvan Mühlemann November 23, 2016 at 6:48 am #

Hi Jason

I bought the book “Deep Learning with Python”. Thanks for your great work!

I see the question about “model.predict()” quite often. I have it as well. In the code above “model” is not a model? I tried “estimator.predict()” but there I get the following error:

> ‘KerasRegressor’ object has no attribute ‘model’

I think it would help many readers

Jason Brownlee November 23, 2016 at 9:06 am #

REPLY

Thanks for your support Silvan.

With a keras model, you can train the model, assign it to a variable and call model.predict(). See this post:
http://machinelearningmastery.com/5-step-life-cycle-neural-network-models-keras/

In the above example, we use a pipeline, which is also a sklearn Estimator. We can call estimator.predict() directly (same function name, different API), more here:

Get Your Start in Machine Learning

http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html#sklearn.pipeline.Pipeline.predict

Does that help?

Dee November 24, 2016 at 9:18 am #

Hey Jason,

Is there anyway for you to provide a direct example of using the model.predict() for the example shown in this post? I've been following your posts for a couple months now and have gotten much more comfortable with Keras. However, I still cannot seem to be able to use .predict() on this example.

Thanks!

Jason Brownlee November 24, 2016 at 10:44 am #

Hi Dee,

There info on the predict function here:
http://machinelearningmastery.com/5-step-life-cycle-neural-network-models-keras/

There's an example of calling predict in this post:
http://machinelearningmastery.com/tutorial-first-neural-network-python-keras/

Does that help?

Silvan Mühlemann November 25, 2016 at 7:20 am #

Hi Dee

Jason, correct me if I am wrong: If I understand correctly the sample above does *not* provide a trained model as output. So you won't be able to use the .predict() function immediately.

Get Your Start in Machine Learning

Instead you have to train the pipeline:

pipeline.fit(X,Y)

Then only you can do predictions:

pipeline.predict(numpy.array([[0.0273, 0. , 7.07 , 0. , 0.469 , 6.421 ,
78.9 , 4.9671, 2. , 242. , 17.8 , 396.9 ,
9.14]]))

will return array(22.125564575195312, dtype=float32)

Jason Brownlee November 25, 2016 at 9:34 am #

Yes, thanks for the correction.

Sorry, for the confusion.

Dee November 28, 2016 at 12:07 pm #

Hey Silvan,

Thanks for the tip! I had a feeling that the crossval from SciKit did not output the fitted m
function.

I'll give it a go with the .fit()!

Thanks!

Sud March 17, 2017 at 1:03 am #

Hi Jason & Silvan,

REPLY

Could you pls tell me whether I am given “pipeline.fit(X,Y)” in correct position?
pls correct me if I am wrong.

```
numpy.random.seed(seed)
estimators = []
estimators.append((‘standardize’, StandardScaler()))
estimators.append((‘mlp’, KerasRegressor(build_fn=larger_model, nb_epoch=50, batch_size=5, verbose=0)))
pipeline = Pipeline(estimators)
pipeline.fit(X,Y)
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(pipeline, X, Y, cv=kfold)
print(“Larger: %.2f (%.2f) MSE” % (results.mean(), results.std()))
```

Thank you!

Jason Brownlee March 17, 2017 at 8:29 am #

pipeline.fit is not needed as you are evaluating the pipeline using kfold cross va

Rahul November 18, 2016 at 3:28 pm #

Dear Jason,
I have a few questions. I am running the wider neural network on a dataset that corresponds to modellin
in and out of a store. I get Wider: 24.73 (7.64) MSE. <– Can you explain exactly what those values mean?

Also can you suggest any other method of improving the neural network? Do I have to keep re-iterating and tuning according to different topological methods?

Also what exact function do you use to predict the new data with no ground truth? Is it the sklearn model.predict(X) where X is the new dataset with one lesser dimension because there is no output? Could you please elaborate and explain in detail. I would be really grateful to you.

Thank you

Jason Brownlee November 19, 2016 at 8:45 am #

REPLY

Hi Rahul,

The model reports on Mean Squared Error (MSE). It reports both the mean and the standard deviation of performance across 10 cross validation folds. This gives an idea of the expected spread in the performance results on new data.

I would suggest trying different network configurations until you find a setup that performs well on your problem. There are no good rules for net configuration.

You can use model.predict() to make new predictions. You are correct.

Rishabh Agrawal September 16, 2017 at 1:24 pm #

Hey! Jason.

Great work on machine learning. I have learned everything from here.

One question.
When we say that we have to train the model first and then predict, are we trying to determine w
other Keras attributes, to get the best fit…and then use the same attributes on prediction datase

Bottom line: are we trying to determine what keras attributes fits our model while we are training

Jason Brownlee September 17, 2017 at 5:23 am #

Generally, we want a model that makes good predictions on new data where we don't know the answer.

We evaluate different models and model configurations on test data to get an idea of how the models will perform when making predictions on new data, so that we can pick one or a few that we think will work well.

Does that help?

Kim December 31, 2016 at 5:45 pm #

REPLY

Hi Jason,

Thank you for the great tutorial.
I redo the code on a Ubuntu machine and run them on TITAN X GPU. While I get similar results for experiment in section 4.1, my results in section 4.2 is different from yours:

Larger: 103.31 (236.28) MSE

no_epoch is 50 and batch_size is 5.

Jason Brownlee January 1, 2017 at 5:23 am #

This can happen, it is hard to control the random number generators in Keras.

See this post:
http://machinelearningmastery.com/randomness-in-machine-learning/

A. Batuhan D. January 20, 2017 at 8:43 pm #

Hi Jason,

Thanks for sharing these useful tutorials. Two questions:

1) If regression model calculates the error and returns as result (no doubt for this) then what is those ‘accuracy’ values printed for each epoch when ‘verbose=1’?

2) With those predicted values (fit.predict() or cross_val_predict), is it meaningful to find the closest value(s) to predicted result and calculate an accuracy? (This way, more than one accuracy can be calculated: accuracy for closest 2, closest 3, …)

Jason Brownlee January 21, 2017 at 10:28 am #

Get Your Start in Machine Learning

Hi A. Batuhan D.,

1. You cannot print accuracy for a regression problem, it does not make sense. It would be loss or error.

2. Again, accuracy does not make sense for regression. It sounds like you are describing an instance based regression model like kNN?

A. Batuhan D. January 23, 2017 at 7:36 pm #

REPLY

Hi jason,

1. I know, it doesn't make any sense to calculate accuracy for a regression problem but when using Keras library and set verbose=1, function prints accuracy values also alongside with loss values. I'd like to ask the reason of this situation. It is confusing. In your example, verbose parameter is set to 0.

2. What i do is to calculate some vectors. As input, i'm using vectors (say embedded word vecto
word prediction) as an output (may not belong to any known vector in dictionary and probably nc
dictionary to one calculated by network by cosine distance approach. Counting model predicted
(say next words vector) than others in dictionary may lead to a reasonable accuracy in my opinic
not related to instance based regression models.

Thanks.

Jason Brownlee January 24, 2017 at 11:03 am #

That is very odd that accuracy is printed for a regression problem. I have not seen i

Are you able to paste a short code + output example?

Partha January 24, 2017 at 7:08 am #

REPLY

Hi,

I tried this tutorial – but it crashes with the following:

Traceback (most recent call last):
File “Riskind_p1.py”, line 132, in
results = cross_val_score(estimator, X, Y, cv=kfold)
File “C:\Python27\lib\site-packages\sklearn\model_selection_validation.py”, line 140, in cross_val_score
for train, test in cv_iter)
File “C:\Python27\lib\site-packages\sklearn\externals\joblib\parallel.py”, line 758, in __call__
while self.dispatch_one_batch(iterator):
File “C:\Python27\lib\site-packages\sklearn\externals\joblib\parallel.py”, line 603, in dispatch_one_batch
tasks = BatchedCalls(itertools.islice(iterator, batch_size))
File “C:\Python27\lib\site-packages\sklearn\externals\joblib\parallel.py”, line 127, in __init__
self.items = list(iterator_slice)
File “C:\Python27\lib\site-packages\sklearn\model_selection_validation.py”, line 140, in
for train, test in cv_iter)
File “C:\Python27\lib\site-packages\sklearn\base.py”, line 67, in clone
new_object_params = estimator.get_params(deep=False)
TypeError: get_params() got an unexpected keyword argument ‘deep’

Some one else also got this same error and posted a question on StackOverflow.

Any help is appreciated.

Jason Brownlee January 24, 2017 at 11:07 am #

Sorry to hear that.

What versions of sklearn, Keras and tensorflow or theano are you using?

REPLY

David January 25, 2017 at 12:23 am #

I have the same problem after an update to Keras 1.2.1. In my case: theano is 0.8.2 and sklearn is 0.18.1.

I could be wrong, but this could be a problem with the latest version of Keras…

David January 25, 2017 at 3:01 am #

REPLY

Ok, I think I have managed to solve the issues. I think the problem are crashess between different version of the packages. What it solves everything is to create an evironment. I have posted in stack overflow a solution, @Partha, here:
http://stackoverflow.com/questions/41796618/python-keras-cross-val-score-error/41832675#41832675

Partha January 25, 2017 at 4:31 am #

My versions are 0.8.2 for theano and 0.18.1 for sklearn and 1.2.1 for keras.
I did a new anaconda installation on another machine and it worked there.
Thanks,

Jason Brownlee January 25, 2017 at 10:08 am #

Thanks David, I'll take a look at the post.

Jason Brownlee January 25, 2017 at 10:58 am #

Hi David, I have reproduced the fault and understand the cause.

The error is caused by a bug in Keras 1.2.1 and I have two candidate fixes for the issue.

I have written up the problem and fixes here:
http://stackoverflow.com/a/41841066/78453

Jason Brownlee January 25, 2017 at 10:06 am #

Get Your Start in Machine Learning

Thanks, I will investigate and attempt to reproduce.

David January 25, 2017 at 8:51 pm #

Hi,
yes, Jason's solution is the correct one. My solution works because in the environment the Keras version installed is 1.1.1, not the one with the bug (1.2.1).

Andy January 25, 2017 at 5:05 am #

REPLY

Great tutorial, many thanks!

Just wondering how do you train on a standardaised dataset (as per section 3), but produce actual (i.e. I
Pipeline?

Jason Brownlee January 25, 2017 at 10:10 am #

Great question Andy,

The standardization occurs within the pipeline which can invert the transforms as needed. This is on

AndyS January 25, 2017 at 7:32 am #

REPLY

Great tutorial, many thanks!

How do I recover actual predictions (NOT standardized ones) having fit the pipeline in section 3 with pipeline.fit(X,Y)? I believe pipeline.predict(testX) yields a standardised predictedY?

I see there is an inverse_transform method for Pipeline, however appears to be for only reverting a trans

James Bond January 26, 2017 at 1:39 am #

REPLY

Thanks for you post..

I am currently having some problems with an regression problem, as such you represent here.

you seem to both normal both input and output, but what do you do if if the output should be used by a different component?… unnormalize it? and if so, wouldn't the error scale up as well?

I am currently working on mapping framed audio to MFCC features.
I tried a lot of different network structures.. cnn, multiple layers..

I just recently tried adding a linear layer at the end… and wauw.. what an effect.. it keeps declining.. how

Jason Brownlee January 26, 2017 at 4:47 am #

Hi James, yes the output must be denormalized (invert any data prep process) before use.

If the data prep processes are separate, you can keep track of the Python object (or coefficients) an

Sarick January 27, 2017 at 6:59 pm #

Is there any way to use pipeline but still be able to graph MSE over epochs for kerasregressor?

Jason Brownlee January 28, 2017 at 7:35 am #

REPLY

Not that I have seen Sarick. If you figure a way, let me know.

Aritra January 28, 2017 at 9:33 pm #

REPLY

Can you tell me how to do regression with convolutional neural network?

Jason Brownlee February 1, 2017 at 10:09 am #

REPLY

Great question Aritra.

You can use the standard CNN structure and modify the example to use a linear output function and a suitable regression loss function.

kono January 29, 2017 at 4:37 pm #

Hi Jason,

Could you tell me how to decide batch_size? Is there a rule of thumb for this?

Jason Brownlee February 1, 2017 at 10:15 am #

Great question kono.

Generally, I treat it like a parameter to be optimized for the problem, like learning rate.

These posts might help:

How large should the batch size be for stochastic gradient descent?
http://stats.stackexchange.com/questions/140811/how-large-should-the-batch-size-be-for-stochastic-gradient-descent

What is batch size in neural network?
http://stats.stackexchange.com/questions/153531/what-is-batch-size-in-neural-network

kono January 29, 2017 at 4:53 pm #

REPLY

Hi Jason,

I see some people use fit_generator to train a MLP. Could you tell me when to use fit_generator() and when to use fit()?

Jason Brownlee February 1, 2017 at 10:16 am #

REPLY

Hi kono, fit_generator() is used when working with a Data Generator, such as is the case with image augmentation:
http://machinelearningmastery.com/image-augmentation-deep-learning-keras/

Pratik Patil February 2, 2017 at 12:39 am #

Hi Jason,
Thank you for the post. I used two of your post this and one on GridSearchCV to get a keras regression
My question is how to get weight matrices and bias vectors of keras regressor in a fit, that is on the pipe
(My posts keep getting rejected/disappear, am I breaking some protocol/rule of the site?)

Jason Brownlee February 2, 2017 at 1:59 pm #

Comments are moderated, that is why you do not seem the immediately.

To access the weights, I would recommend training a standalone Keras model rather than using the KerasClassifier and sklearn Pipeline.

Pedro February 18, 2017 at 7:57 am #

REPLY

Hi,

Get Your Start in Machine Learning

Thank you for the excelent example! as a beginner, it was the best to start with.
But I have some questions:

In the wider topology, what does it mean to have more neurons?

e.g., in my input layer I “receive” 150 dimensions/features (input_dim) and output 250 dimensions (output_dim). What is in those 100 “extra” neurons (that are propagated to the next hidden layers) ?

Best,
Pedro

Jason Brownlee February 18, 2017 at 8:47 am #

REPLY

Hi Pedro,

A neuron is a single learning unit. A layer is comprised of neurons.

The size of the input layer must match the number of input variables. The size of the output layer mu
classes in the case of classification.

The number of hidden layers can vary and the number of neurons per hidden layer can vary. This is .

Does that help?

Pedro Fialho February 20, 2017 at 6:27 am #

Hi,

In your wider example, the input layer does not match/output the number of input variables/features:

model.add(Dense(20, input_dim=13, init=’normal’, activation=’relu’))

so my question is: apart from the 13 input features, what’s in the 7 neurons, output by this (input) layer?

Jason Brownlee February 20, 2017 at 9:33 am #

REPLY

Hi Pedro, I'm not sure I understand, sorry.

The example takes as input 13 features. The input layer (input_dim) expects 13 input values. The first hidden layer combines these weighted inputs 20 times or 20 different ways (20 neurons in the layer) and each neuron outputs one value. These are combined into one neuron (poor guy!) which outputs a prediction.

Pedro Fialho February 21, 2017 at 9:14 pm #

Hi,

Yes, now I understand (I was not confident that the input layer was also an hidden layer)

Jason Brownlee February 22, 2017 at 10:00 am #

The input layer is separate from the first hidden layer. The Keras API makes thi
line.

Bartosz February 19, 2017 at 11:42 am #

Hi Jason,

You've said that an activation function is not necessary as we want a numerical value as an output of our network. I've been looking at recurrent network and in particular this guide: https://deeplearning4j.org/lstm . It recommended using an identity activation function at the output. I was wondering is there any difference between your approach: using Dense(1) as the output layer, and adding an identity activation function at the output of the network: Activation('linear') ? are there any situations when I should use the identity activation layer? Could you elaborate on this?

In case of this tutorial the network would look like this with the identity function:
model = Sequential()
model.add(Dense(13, input_dim=13, init='normal', activation='relu'))

Get Your Start in Machine Learning

model.add(Dense(6, init='normal', activation='relu'))
model.add(Dense(1, init='normal'))
model.add(Activation('linear'))

Regards,
Bartosz

Jason Brownlee February 20, 2017 at 9:25 am #

REPLY

Indeed, the example uses a linear activation function by default.

Dan March 18, 2017 at 7:23 am #

Hi Jason,
my current understanding is that we want to fit + transform the scaling only on our training set and transf
pipeline in the cv like you did. Do we ensure that for each cv the scaling fit only takes place for the 9 trai
set?

Thanks very much

Jason Brownlee March 18, 2017 at 7:55 am #

Top question.

The Pipeline does this for us. It is fit then applied to the training set each CV fold, then the fit transforms are applied to the test set to evaluate the model on the fold. It's a great automatic pattern built into sklearn.

Paula March 21, 2017 at 11:46 pm #

Get Your Start in Machine Learning

Hi! I ran your code with your data and we got a different MSE. Should I be concerned? Thanks for help!

Jason Brownlee March 22, 2017 at 8:07 am #

REPLY

Generally no, machine learning algorithms are stochastic.

More details here:
http://machinelearningmastery.com/randomness-in-machine-learning/

Annanya March 29, 2017 at 4:23 am #

Hi Jason

while running this above code i found the error as

Y = dataset[:,25]
IndexError: index 25 is out of bounds for axis 1 with size 1

i had declared X and Y as

X = dataset[:,0:25]
Y = dataset[:,25]

help me for solving this

Sagar March 29, 2017 at 10:54 am #

REPLY

Hi Jason, Thanks for your great article !

I am working with same problem [No of samples: 460000 , No of Features:8] but my target column output has too big values like in between 20000 to 90000 !

I tried different NN architecture [larger to small] with different batch size and epoch but still not getting g

Get Your Start in Machine Learning

should i have to normalize my target column ? Please help me for solving this !

Thanks for your time !

Jason Brownlee March 30, 2017 at 8:45 am #

REPLY

Yes, you must rescale your input and output data.

Sagar March 31, 2017 at 4:22 pm #

REPLY

Hi Jason, Thanks for your reply !

Yes i tried different ways to rescale my data using

http://machinelearningmastery.com/prepare-data-machine-learning-python-scikit-learn/

url but i still i only got 20% accuracy !

I tried different NN topology with different batch size and epoch but not getting good results !

My code :

inputFilePath = “path-to-input-file”

dataframe = pandas.read_csv(inputFilePath, sep=”\t”, header=None)
dataset = dataframe._values

# split into input (X) and output (Y) variables
X = dataset[:,0:8]
Y = dataset[:,8]

scaler = StandardScaler().fit(X)
X = scaler.fit_transform(X)

maxnumber = max(Y) #Max number i got is : 79882.0

```
Y=Y / maxnumber

# create model
model = Sequential()
model.add(Dense(100, input_dim=8, init='normal', activation='relu'))
model.add(Dense(100, init='normal', activation='relu'))
model.add(Dense(80, init='normal', activation='relu'))
model.add(Dense(40, init='normal', activation='relu'))
model.add(Dense(20, init='normal', activation='relu'))
model.add(Dense(8, init='normal', activation='relu'))
model.add(Dense(6, init='normal', activation='relu'))
model.add(Dense(6, init='normal', activation='relu'))
model.add(Dense(1, init='normal',activation='relu'))

model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['accuracy'])
# checkpoint
model.fit(X, Y,nb_epoch=100, batch_size=400)
# 4. evaluate the network
loss, accuracy = model.evaluate(X, Y)
print("\nLoss: %.2f, Accuracy: %.2f%%" % (loss, accuracy*100))
```

I tried MSE and MAE in loss with adam and rmsprop optimizer but still not getting accuracy !

Please help me ! Thanks

Jason Brownlee April 1, 2017 at 5:51 am #

100 epochs will not be enough for such a deep network. It might need millions.

sagar April 6, 2017 at 11:29 pm #

Hello Jason, Thanks for your reply !

How can i ensure that i will get output after millions of epoch because after 10000 epoch accuracy is still 0.2378 !

How can i dynamically decide the number of layers and Neurons size in my neural network ? Is there any way ?

I already used neural network checkpoint mechanism to ensure its accuracy on validation spilt !
My code looks like

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

checkpoint = ModelCheckpoint(save_file_path, monitor='val_acc', verbose=1, save_best_only=True, mode='max')

callbacks_list = [checkpoint]

model.fit(X_Feature_Vector, Y_Output_Vector,validation_split=0.33, nb_epoch=1000000, batch_size=1300, callbacks=callbacks_list, verbose=0)

Let me know if i miss something !

Jason Brownlee April 9, 2017 at 2:43 pm #

Looks good.

There are neural net growing and pruning algorithms but I do not have tutorials sorry.

See the book: Neural Smithing http://amzn.to/2oOfXOz

Charlotte March 30, 2017 at 8:58 am #

Hi Jason,

Thanks for this great tutorial.

I do believe that there is a small mistake, when giving as parameters the number of epochs, the documentations shows that it should be given as:
estimator = KerasRegressor(build_fn=baseline_model, epochs=100, batch_size=5, verbose=0).

When giving:
estimator = KerasRegressor(build_fn=baseline_model, nb_epoch=100, batch_size=5, verbose=0)

the function doesn't recognise the argument and just ignore it.

Can you confirm?

I'm using your 'How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras' tutorial and have trouble tuning the number of epochs. If I checked one of the results of the GridSearchCv with a simple cross validation with the same number of folds I don't obtain the same results at all. There might be a similar mistake there?

Thank your for your time!

Jason Brownlee March 30, 2017 at 9:01 am #

REPLY

You can pass through any parameters you wish:
https://keras.io/scikit-learn-api/

You will get different results on each run because neural network behavior is stochastic. this post wil
http://machinelearningmastery.com/randomness-in-machine-learning/

Charlotte March 30, 2017 at 9:14 am #

https://keras.io/scikit-learn-api/ precises that number of epochs should be given as epo
the function will ignore the argument. As an example:

```
np.random.seed(seed)
estimators = []
estimators.append(('standardize', StandardScaler()))
estimators.append(('mlp', KerasRegressor(build_fn=baseline_model, nb_epoch='hi', batch_size=50, verbose=0)))
pipeline = Pipeline(estimators)
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(pipeline, X1, Y, cv=kfold)
print("Standardized: %.5f (%.2f) MSE" % (results.mean(), results.std()))
```

will not raise any error.

Get Your Start in Machine Learning

Am I missing something?

The results I get are strongly different and I don't think that this can be due to the stochasticity of the NN behaviour.

Jason Brownlee March 31, 2017 at 5:49 am #

REPLY

Thanks Charlotte, that looks like a recent change for Keras 2.0. I will update the examples soon.

Caleb Everett April 25, 2017 at 7:50 am #

REPLY

Thank you!

Jens April 16, 2017 at 7:59 am #

Hey Jason,

I tried the first part and got a different result for the baseline.
I figured that the
estimator = KerasRegressor(build_fn=baseline_model, nb_epoch=100, batch_size=5, verbose=0)

is not working as expected for me as it takes the default epoch of 10. When I change it to epochs=100 it

I just read the above comment, it seems like they changed that in the API

Jason Brownlee April 16, 2017 at 9:34 am #

REPLY

Neural networks are a stochastic algorithm that gives different results each time they are run (unless you fix the seed and make everything else the same).

See this post:
http://machinelearningmastery.com/randomness-in-machine-learning/

Martin April 19, 2017 at 11:35 pm #

REPLY

Hi Jason,

how can i get regression coefficients?

Jason Brownlee April 20, 2017 at 9:26 am #

Use an optimization algorithm to “find them”.

Stochastic gradient descent with linear regression may be a place to start:
http://machinelearningmastery.com/simple-linear-regression-tutorial-for-machine-learning/

Luca April 27, 2017 at 12:34 am #

Dear Jason,

Thanks for your tutorials!!
I made it work in a particle physics example I’m working on, and I have 2 questions.
1) Imagine my target is T=a/b (T=true_value/reco_value). If I give to the regression both “a” and “b” as features, then it should be able to find exactly the correct solution every time, right? Or there is some procedure that try to avoid overtraining, and do not allow to give a results precise at 100%? I ask because I tried, and I got “good” performances, not optimal as I would expect (if it has “a” and “b” it should be able to find the correct T in the test too at 100%). If I remove b from the regression, and I add other features, then y_hat/y_test is peaking at 0.75, meaning the the regression is biassed. Could you help me understanding these two facts?
2) I want to save the regression in order to use it later. After the training I do: a) estimator.model.save_weights and b) open(‘models/’+model_name, ‘w’).write(estimator.model.to_json()).

Estimator is “estimator = KerasRegressor(build_fn=baseline_model, nb_epoch=100, batch_size=50, verbose=1)” . How can I later use those 2 files to directly make predictions?

Thanks a lot,
Luca

Jason Brownlee April 27, 2017 at 8:42 am #

REPLY

Sorry, I’m not sure I follow your first question, perhaps you can restate it briefly?

See this post on saving and loading keras models:
http://machinelearningmastery.com/save-load-keras-deep-learning-models/

Luca April 28, 2017 at 1:30 am #

Hi Jason,
my point is the following. The regression is trained on a set of features (a set of floats), and it pro
training the regression learn how to guess the target as a function of the features.
Of course the target should not be function of the features, otherwise the problem is trivial, but I
did (as a test) is to define a target that is division of 2 features, i.e. I’m giving to the regression “a
In that simple case, the regression should be smart enough to understand during the training tha
able to find the correct value with 100% precision, i.e. dividing the 2 features. What I found is tha
close to a/b, but not exactly a/b. So I was wondering why the regression is behaving like that.

Thanks,
Luca

Jason Brownlee April 28, 2017 at 7:49 am #

REPLY

This is a great question.

Get Your Start in Machine Learning

At best machine learning can approximate a function, some approximations are better than others.

That is the best that I can answer it.

Ignacio April 27, 2017 at 12:36 am #

REPLY

Hi Jason,

thanks for your posts, I really enjoy them. I have a quick question: If I want to use sklearn's GridSearchCV and :
model.compile(loss='mean_squared_error'
in my model, will the highest score correspond to the combination with the *highest* mse?
If that's the case I assume there is a way to invert the scoring in GridSearchCV?

Jason Brownlee April 27, 2017 at 8:43 am #

When using MSE you will want to find the config that results in the lowest error, e.g. lowest

Navdeep May 2, 2017 at 10:37 pm #

Dear Jason

I have datafile with 7 variables, 6 inputs and 1 output

#from sklearn.cross_validation import train_test_split
#rain, test = train_test_split(data2, train_size = 0.8)

#train_y= train['Average RT']
#train_x= train[train.columns.difference(['Average RT'])]

##test_y= test['Average RT']
#est_x= test[test.columns.difference(['Average RT'])]

Get Your Start in Machine Learning

x=data2[data2.columns.difference([‘Average RT’])]
y=data2[‘Average RT’]

print x.shape
print y.shape

(1035, 6)
(1035L,)
# define base model
def baseline_model():
# create model
model = Sequential()
model.add(Dense(7, input_dim=7, kernel_initializer=’normal’, activation=’relu’))
model.add(Dense(1, kernel_initializer=’normal’))
# Compile model
model.compile(loss=’mean_squared_error’, optimizer=’adam’)
return model

fix random seed for reproducibility
#seed = 7
#numpy.random.seed(seed)
# evaluate model with standardized dataset
estimator = KerasRegressor(build_fn=baseline_model, nb_epoch=100, batch_size=5, verbose=0)

kfold = KFold(n_splits=5, random_state=seed)
results = cross_val_score(estimator, x,y, cv=kfold)
print(“Results: %.2f (%.2f) MSE” % (results.mean(), results.std()))

but getting error below
ValueError: Error when checking input: expected dense_15_input to have shape (None, 7) but got array with shape (828, 6)

Also i tried changing

model.add(Dense(7, input_dim=7, kernel_initializer=’normal’, activation=’relu’))
to
model.add(Dense(6, input_dim=6, kernel_initializer=’normal’, activation=’relu’))

because total i have 7 variables out of which 6 are input, 7th Average RT is output

Get Your Start in Machine Learning

could u help pls

could you help pls
there is non linear relationship also bw o/p and i/p, as ai am trying keras neural to develop relationship that is non linear by itself

Jason Brownlee May 3, 2017 at 7:39 am #

REPLY

If you have 6 inputs and 1 output, you will have 7 rows.

You can separate your data as:

```
X = data[:, 0:6]
y = datap[:, 6]
```

Then, you can configure the input layer of your neural net to expect 6 inputs by setting the “input_dim” argument to 6.

Does that help?

Aghiles June 12, 2017 at 1:40 am #

Dear Jason

and if I have 2 output, can I write

y = data[:, 0:6]
y = data[:, 6:7]

?

Jason Brownlee June 12, 2017 at 7:11 am #

REPLY

Not quite.

You can retrieve the 2 columns from your matrix and assign them to y so that y is now 2 columns and n rows.

```
y = data[:, 6:]
```

Perhaps get more comfortable with numpy array slicing first?

amit kumar May 10, 2017 at 4:23 am #

REPLY

sir plz give me code of “to calculayte cost estimation usin back prpoation technique uses simodial activation function”

Jason Brownlee May 10, 2017 at 8:52 am #

See this post:
http://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/

Francis May 11, 2017 at 5:22 pm #

Hi Jason,

I’m new in deep learning and thanks for this impressive tutorial. However, I have an important question a

How can we interpret these features just like lasso or other feature selection methods?

In my project, I have about 20000 features and I want to selected or ranking these features using deep learning methods. How can we do this?

Thank you!

Jason Brownlee May 12, 2017 at 7:36 am #

REPLY

Great question.

I would recommend performing feature selection as a pre-processing step.

Here's more information on feature selection:
http://machinelearningmastery.com/an-introduction-to-feature-selection/

Alogomining May 15, 2017 at 1:02 pm #

REPLY

Hi,
Thks a lot for this post.
is there a way to implement a Tweedie regression in thsi framework ?
A

Jason Brownlee May 16, 2017 at 8:33 am #

Sorry, I have not heard of "tweedie regression".

Inge May 23, 2017 at 10:41 pm #

Hi,

Thank you for the sharing.
I met a problem, and do not know how to deal with it.
When it goes to "results = cross_val_score(estimator, X, Y, cv=kfold)", I got warnings shown as below:
C:\Program Files\Anaconda3\lib\site-packages\ipykernel__main__.py:11: UserWarning: Update your Dense call to the Keras 2 API: Dense(13, input_dim=13, kernel_initializer="normal", activation="relu")
C:\Program Files\Anaconda3\lib\site-packages\ipykernel__main__.py:12: UserWarning: Update your Dense call to the Keras 2 API: Dense(1, kernel_initializer="normal")
C:\Program Files\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:2289: UserWarning: Expected no kwargs, you passed 1

Get Your Start in Machine Learning

kwargs passed to function are ignored with Tensorflow backend
warnings.warn('\n'.join(msg))

I've tried to update Anaconda and its all packages,but cannot fix it.

Huyen May 27, 2017 at 2:41 am #

REPLY

Hi Jason,

I have a classic question about neural network for regression but I haven't found any crystal answer. I have seen the very good performances of neural network on classification for image and so on but still doubt about its performances on regression. In fact, I have tested with 2 cases of data linear and non linear, 2 input and 1 output with random bias but the performances were not good in comparison with oth r Gradient Boosting… So for regression, which kind of data we should apply neural network? Whether the better?

Thank you for your answer in advance. Hope you have a good day 🙂

Jason Brownlee June 2, 2017 at 11:56 am #

Deep learning will work well for regression but requires larger/harder problems with lots mo

Small problems will be better suited to classical linear or even non-linear methods.

Huyen June 7, 2017 at 4:20 pm #

REPLY

Thank you Jason,

Return in your examples, I have one question about the appropriate number of neurons should be in each hidden layer and the number of hidden layers in a network. I have read some recommendations such that number of hidden layer neurons are 2/3 of size of input layer and the number of neurons it should (a) be between the input and output layer size, (b) set to something near (inputs+outputs) * 2/3, or (c) never larger than twice the size of the input layer to prevent the overfitting. I doubt about these constraints because I haven

With your example, I increase the number of layers to 7 and with each layer, I use a large number of neurons (approximately 300-200) and it gave MSQ to 0.1394 through 5000 epochs. So do you have any conditions about these number when you build a network?

Jason Brownlee June 8, 2017 at 7:38 am #

REPLY

No, generally neural network configuration is trial and error with a robust test harness.

Ali June 2, 2017 at 4:12 pm #

REPLY

Hi jason.Can i apply regression for Autoencoders?

Jason Brownlee June 3, 2017 at 7:21 am #

Yes, but I do not have examples sorry.

KK June 3, 2017 at 4:42 am #

Hi Jason

Thank you for the great tutorial code! I have some questions regarding regularization and kenel initializer.

I'd like to add L1/L2 regularization when updating the weights. Where should I put the commands?

I also have a question abut assigning " kernel_initializer='normal'," Is it necessary to initialize normal kernel?

Thanks!

KK

Jason Brownlee June 3, 2017 at 7:26 am #

REPLY

Here is an example of weight regularization:
http://machinelearningmastery.com/use-weight-regularization-lstm-networks-time-series-forecasting/

I would recommend evaluating different weight initialization schemes on your problem.

KK June 5, 2017 at 5:16 pm #

REPLY

Thanks Jason.

I have one more question. I will use convolution2D with dropout. Do I still need to use L1/L2 regu

Thanks!

KK

Jason Brownlee June 6, 2017 at 9:23 am #

Try with and without and compare performance.

Kid June 8, 2017 at 6:31 pm #

Dear Dr.,

I need you favor on how to use pre trained Keras based sequential model for NER with input text.

Example if “word1 word2 word3.” is a sentence with three words, how I can convert it to numpy array expected by Keras to predict each words NE tag set from the loaded pretrained Keras model.

With regards,

Jason Brownlee June 9, 2017 at 6:20 am #

REPLY

Convert the words to integers first.

Sayak Paul June 15, 2017 at 4:56 am #

REPLY

I am getting a rate of more than 58 every time.

Here's the exact code being used:

```
#Dependencies

from numpy.random import seed
seed(1)

import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

# load dataset
dataframe = pandas.read_csv("housing.csv", delim_whitespace=True, header=None)
dataset = dataframe.values
# split into input (X) and output (Y) variables
X = dataset[:,0:13]
Y = dataset[:,13]

# Basic NN model using Keras
```


Get Your Start in Machine Learning

```
def baseline_model():
# create model
model = Sequential()
model.add(Dense(13, input_dim=13, kernel_initializer='normal', activation='relu'))
model.add(Dense(1, kernel_initializer='normal'))
# Compile model
model.compile(loss='mean_squared_error', optimizer='adam')
return model

#seed = 1
# evaluate model with standardized dataset
estimator = KerasRegressor(build_fn=baseline_model, nb_epoch=100, batch_size=5, verbose=0)

#kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(estimator, X, Y, cv=10)
print("Results: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

Jason Brownlee June 15, 2017 at 8:51 am #

What do you mean by "a rate of more than 58"?

Hossein June 16, 2017 at 3:53 am #

Thank you very much,
Cant we use CNN instead of Dense layers? in case we want to use CNN, should we use conv2d or simply conv?
In regression problems using deep architectures, can we use AlexNet, VGGNet, and the likes just like how we use them with images?
I would appreciate if you could have an example in this regard as well
Best Regards

Jason Brownlee June 16, 2017 at 8:05 am #

REPLY

I would not recommend a CNN for regression. I would recommend a MLP.

The shape of your input data (1d, 2d, …) will define the type of CNN to use.

Jose June 18, 2017 at 1:34 pm #

REPLY

Great tutorial! I liked to save the weight that I adjusted in training, how can I do it?

Jason Brownlee June 19, 2017 at 8:33 am #

This tutorial will show you how to save network weights:
http://machinelearningmastery.com/save-load-keras-deep-learning-models/

Jacob June 20, 2017 at 11:14 am #

Thank you very much.
I have a question.
Is this tutorial suitable for wind speed prediction?

Jason Brownlee June 21, 2017 at 8:08 am #

REPLY

Try it and see.

Roy July 2, 2017 at 3:04 pm #

REPLY

Hi, Thank you for the tutorial. Few questions here.

1. What is the differences when we use
KerasRegressor(build_fn=baseline_model, nb_epoch=100, batch_size=5, verbose=0)

and with

model.fit(x_train, y_train, batch_size=batch_size,epochs=epochs,verbose=1,validation_data=(x_test, y_test))? AFAIK, with when using KerasRegressor, we can do CV while can't on model.fit. Am I right? Will both result in the same MSE etc?

2. How do create a neural network that predict two continuous output using Keras? Here, we only predict one output, how about two or more output? How do we implement that? (Multioutput regression problem?)

Jason Brownlee July 3, 2017 at 5:30 am #

Correct, using the sklearn wrapper lets us use tools like CV on small models.

You can have two outputs by changing the number of nodes in the output layer to 2.

Roy July 4, 2017 at 2:35 pm #

Thanks for the reply.

Does that mean that with sklearn wrapper model and with model.fit(without sklearn) model are able to get the same mse if both are given same train, valid, and test dataset (assume sklearn wrapper only run 1st fold)? Or there are some differences behind the model?

I read about the Keras Model class (functional API) (https://keras.io/models/model/). Is the implementation of the Model class,

model = Model(inputs=a1, outputs=[output1, output2])

the same as adding 1 node more at the output layer? If no, what's the differences?

Jason Brownlee July 6, 2017 at 10:10 am #

REPLY

Same keras model under the covers.

Nandini July 4, 2017 at 5:17 pm #

REPLY

```
from keras.layers.core import Dense,Activation,Dropout
from json import load,dump
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from keras.models import Sequential
from keras2pmml import keras2pmml
from pyspark import SparkContext,SparkConf
from pyspark.mllib.linalg import Matrix, Vector
from elephas.utils.rdd_utils import to_simple_rdd,to_labeled_point
from elephas import optimizers as elephas_optimizers
from elephas.spark_model import SparkModel
from CommonFunctions import DataRead,PMMLGenaration,ModelSave,LoadModel,UpdateDictionary,M
from keras import regularizers
from sklearn.metrics import r2_score
from keras.optimizers import SGD
#from keras.models import model_from_config
#from keras.utils.generic_utils import get_from_module

class SNNReg:

def train(self,sc,xml,data,hdfs_path):
## Variable initialization ##
hActivation = xml.HiddenActivation
#print hActivation
nodeList = map(int,(xml.NodeList.split(“,”)))
#print nodeList
```

```
Accuracy = xml.Accuracy
#print Accuracy
lossFn = xml.LossFunction
#print nodeList
optimi = xml.Optimizer
#print optimi
hCount = int(xml.HiddenNodeCount)
#print hCount
inputDim = int(xml.InputDimension)
opNodes = int(xml.OutputNodes)
print ('opNodes',opNodes)
nbEpoch = int(xml.NumEpoch)
batchSize = int(xml.BatchSize)
#settings default paramerters if not the provided the values for it
if hActivation=="":
hActivation="relu"
if lossFn=="":
lossFn="mean_squared_error"
if optimi=="":
optimi="adam"
if Accuracy=="":
Accuracy="Accuracy"
print "now going to read "
#print("lossFn",lossFn)
X,Y = DataRead(self,xml.NeuralNetCategory,xml.NeuralNetType,data,xml.ColumnNames,xml.TargetCol

# Creating a sequential model for simple neural network
model=Sequential()

model.add(Dense(nodeList[0],input_dim = inputDim,init='normal',activation =hActivation ))

# Creating hidden model nodes based on the hidden layers count
if hCount > 1:
for x in range(1,hCount):
model.add(Dense(nodeList[x],init='normal',activation = hActivation))
```

```
model.add(Dense(opNodes,activation='linear'))
# Compile model
print "model complilation stage"
model.compile(loss = lossFn, optimizer=optimi)
rdd =to_simple_rdd(sc,X,Y)
print rdd
#adam= elephas_optimizers.Adam()
adam = elephas_optimizers.Adam()
#adagrad = elephas_optimizers.Adagrad()
#adadelta = elephas_optimizers.Adadelta()
#print ("type of rdd",type(rdd))
print "now going to create spark model using elphass"
# Creating Spark elephas model from the spark model
print("no of workers",int(sc._conf.get('spark.pangea.ae.workers')))

sparkModel = SparkModel(sc,
model,
optimizer=adam,
frequency='epoch',
mode='asynchronous',
master_loss=lossFn,
num_workers=int(sc._conf.get('spark.pangea.ae.workers')))

# Train Spark model

print "now it is going to run train fucntion"
sparkModel.train(rdd,nb_epoch=nbEpoch, batch_size=batchSize)
```


i am trying to implement regression in Neural networks usign elphas and keras in python in a distributed way,but while trianing the i am getting to much high loss values , what i have to do ,give me any suggestions for go further.

Jason Brownlee July 6, 2017 at 10:12 am #

REPLY

Sorry I cannot help with distributing a Keras model.

Get Your Start in Machine Learning

Foad July 6, 2017 at 9:35 am #

REPLY

two small points:
1. please mention in the text that it is required to have TensorFlow installed
2. CSV, means comma separated file, but data in the file are not separated by commas. not a big deal though

Jason Brownlee July 6, 2017 at 10:27 am #

REPLY

Thanks for the suggestions.

Timothy Yan July 6, 2017 at 1:28 pm #

Thank you for the nice tutorial! In the post, you used “relu”, but I was wondering how to customi

Jason Brownlee July 9, 2017 at 10:22 am #

You can use sigmoid or tanh if you prefer.

Don July 8, 2017 at 10:09 am #

REPLY

Hi Jason,

Thanks for the great tutorial!

What are the advantages of the deep learning library Keras (with let’s say TensorFlow as the backend) c
MLPRegressor? In both cases, the procedure (input) is very similar, where you have to decide which arc

Get Your Start in Machine Learning

use.

Thanks,
Don

Jason Brownlee July 9, 2017 at 10:50 am #

REPLY

Speed of development and size of community.

Don July 10, 2017 at 7:43 am #

Thanks for the quick reply!

Can you please elaborate a little bit more?

When you are writing speed of development, can you please give a few practical examples for w
are writing size of community, do you mean that the Keras/TensorFlow community is larger than

In addition, can you please add a few words on the epochs and batch_size parameters? Why is
more sense to me? Does it make sense that sometimes when I increase the the epocks value, t

Thanks a lot!
Don

Jason Brownlee July 11, 2017 at 10:24 am #

REPLY

Yes, I believe it is easier/faster to develop models with Keras than other tools currently available.

I believe the Keras community is active and this is important to having the library stay current and useful. Keras is complementary to sklearn, tensorflow and theano.

One epoch is one pass through all training samples. One epoch is comprised of one or more batches. One batch involves a pass through one or more samples before updating the network weights.

Adam July 9, 2017 at 9:35 am #

REPLY

I'm a little confused. If you define x as:
X = dataset[:,0:13]

then the last column in X is the same as Y. Shouldn't X be:
X = dataset[:,0:12]
and then
Y = dataset[:,13]

If you define X to include the outputs, why wouldn't it just set all the weights for dataset[0:12] to zero the
answer?

Justtestityourselfnexttime July 12, 2017 at 12:55 am #

> X = [0,1,2,3,4]
> print(X[0:3])
[0, 1, 2]

End index is exclusive.

Jason Brownlee July 12, 2017 at 9:48 am #

REPLY

Yes. The more questions I get like this, the more I feel I need a post on basic numpy syntax.

Get Your Start in Machine Learning

Nandu July 11, 2017 at 7:54 pm #

REPLY

What are methods to validate the regression model in keras?Please can you help in that

Jason Brownlee July 12, 2017 at 9:42 am #

REPLY

You can estimate the skill of a model on unseen data using a validation dataset when fitting the model.

See the validation_split and validation_data arguments to the fit() function:
https://keras.io/models/sequential/

Ambika July 11, 2017 at 7:58 pm #

how can we recognize the keras regresssion model and classification model with code.

Jason Brownlee July 12, 2017 at 9:43 am #

By the choice of activation function on the output layer and the number of nodes.

Regression will use a linear activation, have one output and likely use a mse loss function.
Classification will use a softmax, tanh or sigmoid activation function, have one node per class (or on
function.

FrankLu July 13, 2017 at 11:24 am #

REPLY

Thanks for your tutotials and I find it helpful. However, I have a question.
When I use checkpoint callbacks in estimator.fit, it save a best trained weights, as a hd5 file.

Get Your Start in Machine Learning

But I cant load this pre trained weights, caz estimator does not have the method of load_weights which is one in keras models. What can I do, thank you!!!

REPLY

Jason Brownlee July 13, 2017 at 4:56 pm #

You must load the weights as a Keras model.

Learn more in this tutorial:
http://machinelearningmastery.com/check-point-deep-learning-models-keras/

Paul July 13, 2017 at 3:20 pm #

Hello, Jason
Thanks for the amazing tutorial! I learned alot from your blogs.
I have a question about np.random.seed.
What does the 'np.random.seed' actually do?
You explained that it is for reproducibility above but I didn't understand what it means..

Thank you and hope you have a great one!

Best,
Paul

REPLY

Jason Brownlee July 13, 2017 at 4:59 pm #

Many machine learning algorithms are stochastic by design:
http://machinelearningmastery.com/randomness-in-machine-learning/

We can remove this randomness in tutorials (purely for demonstration purposes) by ensuring we have the same amount of randomness each time the code is run:
http://machinelearningmastery.com/reproducible-results-neural-networks-keras/

Get Your Start in Machine Learning

This is not recommended for evaluating models in practice:
http://machinelearningmastery.com/evaluate-skill-deep-learning-models/

Does that help Paul?

nandu July 14, 2017 at 7:37 pm #

REPLY

Could you suggest the hidden activation functions for regression Neural networks other than relu.

Jason Brownlee July 15, 2017 at 9:41 am #

Yes, sigmoid and tanh were used for decades before relu came along.

Nandini July 18, 2017 at 3:04 pm #

I have given tanh to regression model usign keras,i am not getting good results,you sai
any suggesstions.

Jason Brownlee July 18, 2017 at 5:02 pm #

No, use a “linear” activation on the output layer for regression problems.

NANDINI July 19, 2017 at 3:14 pm #

yeah of course , for output layer i have given the linear activation function only,I
given relu,if i would give tanh i am not getting good results.

Jason Brownlee July 19, 2017 at 4:10 pm #

I'm sorry to hear that. I have a list of ideas to try in this post:
http://machinelearningmastery.com/improve-deep-learning-performance/

Wayne Tobin July 15, 2017 at 7:17 pm #

REPLY

Hi Jason,

Now that Keras and Tensorflow are available in R (RStudio) do you have any plans on doing above tutor
Boston Housing dataset using cubist and would love to see/run a direct comparison to get a sense of wh

Jason Brownlee July 16, 2017 at 7:58 am #

Perhaps in the future, thanks for the suggestion Wayne.

Ronald Levine July 19, 2017 at 5:58 am #

My problem is that everything is hidden in the Pipeline object. How do I pull out the components, such as the model predict method, then to pull out the predicted values to plot against the input values.

Jason Brownlee July 19, 2017 at 8:30 am #

REPLY

Don't use the Pipeline and pass data between the objects manually.

nandu July 19, 2017 at 4:32 pm #

REPLY

I have train the keras model, i need the logic for model.predict() ,how we are predicting the the values on test data,i have logic for predict_classes,but i don't have logic for predict ,Please can you tell me logic for model.predict.

```
def predict_proba(self, X):
a = X
for i in range(self._num_layers):
g = self._activations[i]
W = self._weights[i]
b = self._biases[i]
a = g(np.dot(a, W.T) + b)
print(len(a))
return a

def predict_classes(self, X):
probs = self.predict_proba(X)
print(np.argmax(probs,1))
return np.argmax(probs,1)
```

predict_classes for classification.
i need predict logic for regression.

Jason Brownlee July 20, 2017 at 6:17 am #

You can use the predict() function:

```
yhat = model.predict()
```

Mustafa July 26, 2017 at 7:05 am #

Get Your Start in Machine Learning

Hi Jason,

Thanks for the blog. I am trying to use the example for my case where I try to build a model and evaluate it for audio data. I use only spectrum data. Original data are in .wav format.

However, I am getting an error

“TypeError: can’t pickle NotImplementedType objects”

in line results = cross_val_score(pipeline, X, Y, cv=kfold)
My data is very small, only 5 samples.

Do you have any idea for this error?

Best,
Mustafa

Jason Brownlee July 26, 2017 at 8:03 am #

I would recommend talking to the people from which you got the pickled data.

Heringsalat August 9, 2017 at 12:15 am #

Hello Mustafa,

how is your pipeline initialized/defined? I had the exactly same error message at a line where I used
When you use something like

estimator = KerasRegressor(build_fn=myModel, nb_epoch=100, batch_size=5, verbose=0)

with a Keras-Model “myModel” and NOT with a function called “myModel” to return the model after compiling it like in the tutorial at the beginning you should get the same Pickle error. You can reproduce it with the tutorial code via myModel=baseline_model().

I hope this is helpful…

Best regards,
Heringsalat

ambika July 26, 2017 at 9:20 pm #

REPLY

why we are caluculating error rather than accuracy in regression problem,why accuracy does not make sence regression ,Please can you explain it.

Jason Brownlee July 27, 2017 at 8:04 am #

REPLY

We are not using accuracy. We are calculating error, specifically mean squared error (MSE)

ambika July 27, 2017 at 2:39 pm #

why we are calculating mse rather than accuracy sir?

Jason Brownlee July 28, 2017 at 8:27 am #

Because it is a regression problem and accuracy is only for classification problems.

ambika July 28, 2017 at 3:38 pm #

REPLY

while i am calulating loss and mse i am getting same values for regression,is that loss and mse are same in regression or different,if it is different ,how it is different,please can you explain it.

Get Your Start in Machine Learning

Jason Brownlee July 29, 2017 at 8:05 am #

REPLY

Loss is the objective minimized by the network. If you use mse as the loss, then you will not need to track mse as a metric as well. They will be the same thing.

Masuk July 28, 2017 at 9:28 pm #

REPLY

Hello!
I am trying to train a ppg signal to estimate the heart rate i.e BPM.
Do you think it is appropriate to follow this structure?
If not please kindly help me by suggesting better methods.

Thank You!

Jason Brownlee July 29, 2017 at 8:12 am #

Perhaps. Also consider a time series formulation. Evaluate every framing you can think of.

Paul August 7, 2017 at 1:37 pm #

Hi Jason! 🙂
Thank you for great post! 🙂 I have a question about StandardScaler and Normalization.
What is difference between them? Also, can I use Min Max scaler instead of StandardScaler?

Thanks in advance.

Best,
Paul

Jason Brownlee August 8, 2017 at 7:42 am #

REPLY

Normalization via the MinMaxScaler scales data between 0-1. Standardization via the StandardScaler subtracts the mean to give the distribution a mean of 0 and a standard deviation of 1.

Standardization is good for Gaussian distributions, normalization is good otherwise.

Paul August 9, 2017 at 1:57 pm #

REPLY

Ah ha! Thanks for replying me back! 🙂
I'll try MinMaxScaler()

Best,
Paul

Jason Brownlee August 10, 2017 at 6:45 am #

Good luck Paul.

Vipul September 3, 2017 at 7:41 pm #

Hi Jason,

Again, its very informative blog. I am implementing keras in R but I couldn´t find keras regressor to fit the model. Do you have workaround for this or could you please suggest what can be used as an alternative?

Jason Brownlee September 4, 2017 at 4:30 am #

REPLY

Sorry, I don't know about Keras in R.

James September 6, 2017 at 2:49 am #

REPLY

Hey Jason – Thanks for the post.

I'd love to hear about some other regression models Keras offers and your thoughts on their use-cases.

Thanks,
James

Jason Brownlee September 7, 2017 at 12:46 pm #

What do you mean James? Do you have an example?

Mohit Jain September 16, 2017 at 5:34 pm #

Hi David,

Thanks of the tutorials. These have been very helpful both for the implementation side to getting an insig
various fields.

I was trying to run the code in section 2 and came across the following error:

……
Traceback (most recent call last):
File "regression.py", line 48, in
results = cross_val_score(estimator, X, Y, cv=kfold)
File "/home/mjennet/anaconda2/lib/python2.7/site-packages/sklearn/model_selection/_validation.py", line 321, in cross_val_score
pre_dispatch=pre_dispatch)
File "/home/mjennet/anaconda2/lib/python2.7/site-packages/sklearn/model_selection/_validation.py", lin

for train, test in cv.split(X, y, groups))
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/sklearn/externals/joblib/parallel.py”, line 779, in __call__
while self.dispatch_one_batch(iterator):
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/sklearn/externals/joblib/parallel.py”, line 625, in dispatch_one_batch
self._dispatch(tasks)
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/sklearn/externals/joblib/parallel.py”, line 588, in _dispatch
job = self._backend.apply_async(batch, callback=cb)
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/sklearn/externals/joblib/_parallel_backends.py”, line 111, in apply_async
result = ImmediateResult(func)
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/sklearn/externals/joblib/_parallel_backends.py”, line 332, in __init__
self.results = batch()
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/sklearn/externals/joblib/parallel.py”, line 131, in __call__
return [func(*args, **kwargs) for func, args, kwargs in self.items]
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/sklearn/model_selection/_validation.py”, lin
estimator.fit(X_train, y_train, **fit_params)
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/keras/wrappers/scikit_learn.py”, line 137, in
self.model = self.build_fn(**self.filter_sk_params(self.build_fn))
File “regression.py”, line 35, in baseline_model
model.add(Dense(13, input_dim=13, kernel_initializer=’normal’, activation=’relu’))
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/keras/layers/core.py”, line 686, in __init__
super(Dense, self).__init__(**kwargs)
File “/home/mjennet/anaconda2/lib/python2.7/site-packages/keras/engine/topology.py”, line 307, in __ini
assert kwarg in allowed_kwargs, ‘Keyword argument not understood: ‘ + kwarg
AssertionError: Keyword argument not understood: kernel_initializer
……..

I tried to get more insight about the problem and came across the problem described by Mr. Partha which seems to be similar as mine and hence checked the version of Keras. The version of keras that I am using is 1.1.1 with tensorflow(1.2.1) as backend. Can you help me with this?

Jason Brownlee September 17, 2017 at 5:26 am #

REPLY

My name is Jason.

Get Your Start in Machine Learning

It looks like you need to update to Keras 2.

Mohit Jain November 26, 2017 at 8:30 pm #

REPLY

Apologies Mr. Jason

I tried to upgrade Keras as well as other dependencies but again the same error pops up. I am currently working on Keras 2.1.1 with Numpy 1.13.3 and scipy 1.0.0

Jason Brownlee November 27, 2017 at 5:49 am #

I am surprised as your error suggests an older version of Keras.

I have a good list of places to get help with Keras here that you could try:
https://machinelearningmastery.com/get-help-with-keras/

Bharath December 11, 2017 at 5:24 pm #

Hi Jason,

Thanks for the example. I get the same error too. Keras/Theano/sklearn: 2.1.2/0.90/0.19

Jason Brownlee December 12, 2017 at 5:23 am #

Sorry to hear that, I normally think it would be a version issue, but you look up to date.

I don't have any good ideas, let me know if you learn more?

asyraf September 26, 2017 at 4:58 pm #

REPLY

Hello Jason,

I used r2 metric on above code and figured that wider model has better score than deeper model. does this mean wider model is better than deeper? is r2 score a good metric to rate a regression model in this case?

Jason Brownlee September 27, 2017 at 5:39 am #

REPLY

Generally, neural network models are stochastic, meaning that they can give different results each time they are run:
https://machinelearningmastery.com/randomness-in-machine-learning/

I generally recommend this process to effectively evaluate neural networks:
https://machinelearningmastery.com/evaluate-skill-deep-learning-models/

tim September 28, 2017 at 12:00 pm #

Hello Jason, I am using your code from section

”
import numpy
import pandas
…
”

to this section

”
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(estimator, X, Y, cv=kfold)
print(“Results: %.2f (%.2f) MSE” % (results.mean(), results.std()))
”

but mean and std values are always higher than your result

”

Results: 60.40 (41.96) MSE

”

where is the problem??

Jason Brownlee September 28, 2017 at 4:44 pm #

REPLY

Machine learning algorithms are stochastic, it may simply be different results on different hardware/library versions.

See this post:
https://machinelearningmastery.com/randomness-in-machine-learning/

tim September 29, 2017 at 1:32 pm #

The result of cross_val_score

”

[10.79553125, 7.68724794, 11.24587975, 27.62757629,
10.6425943 , 8.12384602, 4.93369368, 91.03362441,
13.37441713, 21.56249909]

”

are “Mean square error” ?? or something else??

If they are MSE,
can I say this prediction model is very bad??

Jason Brownlee September 30, 2017 at 7:35 am #

The score are MSE. You could take the sqrt to convert them to RMSE.
How good a score is, depends on the skill of a baseline model (e.g. they're relative) on the problem and domain knowledge (e.g. their interpretation).

Gabby October 5, 2017 at 7:17 pm #

REPLY

I am having issues with cross_val_score. Whenever I run the code, I get the error:

#TypeError: The added layer must be an instance of class Layer. Found:

Suggestions?

Thank you!

The full output:
Traceback (most recent call last):
File "Y:\Tutorials\Keras_Regression_Tutorial\Keras_Regression_Tutorial\module1.py", line 39, in
results = cross_val_score(estimator, X, Y, cv=kfold)
File "C:\Users\Gabby\y35\lib\site-packages\sklearn\model_selection_validation.py", line 321, in cross_v
pre_dispatch=pre_dispatch)
File "C:\Users\Gabby\y35\lib\site-packages\sklearn\model_selection_validation.py", line 195, in cross_v
for train, test in cv.split(X, y, groups))
File "C:\Users\Gabby\y35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 779, in __call__
while self.dispatch_one_batch(iterator):
File "C:\Users\Gabby\y35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 625, in dispatch_one
self._dispatch(tasks)
File "C:\Users\Gabby\y35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 588, in _dispatch
job = self._backend.apply_async(batch, callback=cb)
File "C:\Users\Gabby\y35\lib\site-packages\sklearn\externals\joblib_parallel_backends.py", line 111, in apply_async
result = ImmediateResult(func)
File "C:\Users\Gabby\y35\lib\site-packages\sklearn\externals\joblib_parallel_backends.py", line 332, in __init__
self.results = batch()
File "C:\Users\Gabby\y35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in __call__
return [func(*args, **kwargs) for func, args, kwargs in self.items]

File “C:\Users\Gabby\y35\lib\site-packages\sklearn\externals\joblib\parallel.py”, line 131, in
return [func(*args, **kwargs) for func, args, kwargs in self.items]
File “C:\Users\Gabby\y35\lib\site-packages\sklearn\model_selection_validation.py”, line 437, in _fit_and_score
estimator.fit(X_train, y_train, **fit_params)
File “C:\Users\Gabby\y35\lib\site-packages\tensorflow\contrib\keras\python\keras\wrappers\scikit_learn.py”, line 157, in fit
self.model = self.build_fn(**self.filter_sk_params(self.build_fn))
File “Y:\Tutorials\Keras_Regression_Tutorial\Keras_Regression_Tutorial\module1.py”, line 25, in baseline_model
model.add(Dense(13, input_dim=13, kernel_initializer=’normal’, activation=’relu’))
File “C:\Users\Gabby\y35\lib\site-packages\tensorflow\contrib\keras\python\keras\models.py”, line 460, in add
‘Found: ‘ + str(layer))
TypeError: The added layer must be an instance of class Layer. Found:
Press any key to continue . . .

Jason Brownlee October 6, 2017 at 5:36 am #

Sorry, I have not seen this error before.

Confirm that you Python libraries including Keras and sklearn are up to date.

Confirm that you copied all of the code from the example.

Piotr October 11, 2017 at 1:42 am #

Hello! Thanks for really great tutorial. It’s help a lot!

But I have a question: Do you know how Can I use StandarsScaler in a pipeline, when I deal with CNN and 2D images? My X data has shape e.g. (39, 256, 256, 1).

It works perfectly without StandardScaler, but with StandardScaler I’ve got following error:
ValueError: Found array with dim 4. StandardScaler expected <= 2.

Do you know how can I convert my input data and where in order to work with CNN, 2D images and Sta

Jason Brownlee October 11, 2017 at 7:57 am #

REPLY

I would recommend using the built-in data scaling features for images built into Keras:
https://machinelearningmastery.com/image-augmentation-deep-learning-keras/

Piotr October 11, 2017 at 7:24 pm #

REPLY

Thanks a lot, for quick response! It's good to know that Keras has already ImageDataGenerator for augmenting images.

I have one more question, do you know how can I rescale back outputs from NN to original scale? Is there any DataGenerator that can do something similar to StandardScaler.inverse_transform() from sci-kit learn?

Jason Brownlee October 12, 2017 at 5:27 am #

I'm not sure, I don't think so.

If the image is an input, why would you need to reverse the operation?

Piotr October 12, 2017 at 9:50 pm #

In my case output of my network is based on actual values of pixels. I think, that in my case I will simply omit standardization. But thank you for mentioning ImageDataGenerator, it will help me much in other cases 🙂

Tony October 24, 2017 at 3:35 am #

REPLY

Hello,
Thank you very much for your post
I use the data you uploaded.
However, when I print the MSE, it noticed that : Found input variables with inconsistent numbers of sample [506, 1]. It is the final sample in the data.
please help me

Thank you very much

REPLY

Jason Brownlee October 24, 2017 at 5:37 am #

Sorry, I don't follow, can you restate the issue please?

Tony October 24, 2017 at 11:57 am #

At the end of step 2, evaluate the baseline model, I could't print because that error:
" Found input variables with inconsistent numbers of sample [506, 1]. It is the final sample in the data."

Jason Brownlee October 24, 2017 at 3:59 pm #

Perhaps double check that you copied all of the code exactly?

REPLY

Tony October 24, 2017 at 5:27 pm #

My mistake. I splitted data into columns already in Excel by "Text to Columns" function.

Thank you so much 🙂

Get Your Start in Machine Learning

Harry October 25, 2017 at 2:22 am #

REPLY

Thank you so much for these articles. Two questions:

1) You state “a mean squared error loss function is [used]….This will be the same metric that we will use to evaluate the performance of the model.” I see where ‘mean_squared_error’ is passed as the ‘loss’, but there no ‘metrics=[…]’ arg passed. Does Keras simply use the ‘loss’ function as the metric, if no metric is specified?

2) I recreated this experiment and added the arg “shuffle=True” to the KFold function. This appears to improve performance down to 13.52 (6.99) MSE (wider_model). Any thoughts on this potential optimization? It seemed almost “too good to be true”.

Thanks!

Jason Brownlee October 25, 2017 at 6:51 am #

Yes, Keras will report the loss and the metrics during training, this post might help you unde
https://machinelearningmastery.com/custom-metrics-deep-learning-keras-python/

Great work.

The result might not be real (e.g. not statistically significant), consider this methodology for evaluatin
https://machinelearningmastery.com/evaluate-skill-deep-learning-models/

Tony October 26, 2017 at 3:34 am #

Hello
The NN model you created contains 1 output.

I have a problem with more than 1 output.
I want to apply this code by modifying it. Is it ok ?
Can you suggest some solutions or notice to solve the problem?

Thank you very much

Get Your Start in Machine Learning

Jason Brownlee October 26, 2017 at 5:31 am #

REPLY

Yes, you can change the number of outputs.

Brence October 26, 2017 at 6:26 pm #

REPLY

Hey Jason,

I'm getting an error when running this code. I have karas 2, and scikit learn .17 installed. I keep getting this error:

Connected to pydev debugger (build 172.3968.37)
Using TensorFlow backend.
Traceback (most recent call last):
File "/home/b/pycharm-community-2017.2.3/helpers/pydev/pydevd.py", line 1599, in
globals = debugger.run(setup['file'], None, None, is_module)
File "/home/b/pycharm-community-2017.2.3/helpers/pydev/pydevd.py", line 1026, in run
pydev_imports.execfile(file, globals, locals) # execute the script
File "/home/b/PycharmProjects/ANN1a/ANN2-Keras1a", line 6, in
from sklearn.model_selection import cross_val_score
ImportError: No module named model_selection
Backend TkAgg is interactive backend. Turning interactive mode on.

Is it saying I have no module for Sklearn because I only have .17 instead of the current version which i th
scikit-learn package.

Jason Brownlee October 27, 2017 at 5:18 am #

REPLY

You will need to update your sklearn to 0.18 or higher.

Get Your Start in Machine Learning

Brence October 27, 2017 at 11:06 pm #

REPLY

Hey Jason I need some help with this error message. I'm not sure whats going on with it.

'ValueError: could not convert string to float: Close'

I think it may be talking about one of my columns in my dataset.csv file which is named 'Close'.

Here is the code:

```
import numpy
import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

# load dataset
dataframe = pandas.read_csv("PTNprice.csv", delim_whitespace=True, header=None, usecols=[1,2,3,4]
dataset = dataframe.values
# split into input (X) and output (Y) variables
X = dataset[:,0:4]
Y = dataset[:,1]

# define the model
def larger_model():
# create model
model = Sequential()
model.add(Dense(100, input_dim=4, kernel_initializer='normal', activation='relu'))
model.add(Dense(50, kernel_initializer='normal', activation='relu'))
model.add(Dense(1, kernel_initializer='normal'))
# Compile model
model.compile(loss='mean_squared_error', optimizer='adam')
return model
```

```
# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)

# evaluate model with standardized dataset
numpy.random.seed(seed)
estimators = []
estimators.append((‘standardize’, StandardScaler()))
estimators.append((‘mlp’, KerasRegressor(build_fn=larger_model, epochs=50, batch_size=5, verbose=0)))
pipeline = Pipeline(estimators)
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(pipeline, X, Y, cv=kfold)
print(“Standardized: %.2f (%.2f) MSE” % (results.mean(), results.std()))
```

Jason Brownlee October 28, 2017 at 5:14 am #

Are you using the code and the data from the tutorial?

Did you copy it all exactly, including indenting?

Brence October 28, 2017 at 6:27 am #

Jason,

No my code is modified to try and handle a new data text. I felt the data was very similar to the original dataset. I actually got it to work with no errors. I just changed header=none to header=1

code:

```
# load dataset
dataframe = pandas.read_csv(“PTNprice.csv”, delim_whitespace=True, header=1, usecols=[1,2,3,4])
dataset = dataframe.values
# split into input (X) and output (Y) variables
```

X = dataset[:,0:4]
Y = dataset[:,1]

It took longer then expected for the script to finish. I'm trying to get it to make a prediction as well, but my output is less the satisfactory. Here is what the out gave me. What does this mean do you think?

output:

Larger: 0.00 (0.00) MSE
[0.78021598 0.79241288 0.81000006 …, 3.64232779 3.59621549
3.79605269]

My data is just stock prices from a 10 year period example: 0.75674 0.9655 3.753 1.0293 columns set up like this.

Jason Brownlee October 29, 2017 at 5:48 am #

You may need to tune the model for your specific problem, here are some ideas on
http://machinelearningmastery.com/improve-deep-learning-performance/

Duccio November 4, 2017 at 4:41 am #

Hi Jason,

thank you so much, these courses are great, and very helpful !

I have written the code, following yours, but with the only difference that I have not used Pipeline, and take care of the scaling separately

seed = 7
np.random.seed(seed)

X = (X – X.mean(axis=0))/X.std(axis=0)

estimator = KerasRegressor(build_fn=baseline_model, nb_epoch = 100, batch_size=5,verbose=0)

Get Your Start in Machine Learning

kfold = KFold(n_splits=10,random_state=seed)

results = cross_val_score(estimator, X, y, cv=kfold)

For some reason, I don't understand, your method constantly produces better results. Any idea why it performs better ?

Thanks a lot

Jason Brownlee November 4, 2017 at 5:33 am #

REPLY

It might be a statistical fluke, try varying the random seed and repeat the experiment 10 to 30 times and take the average score for each model.

This post has more ideas on effective ways to evaluate stochastic algorithms:
https://machinelearningmastery.com/evaluate-skill-deep-learning-models/

Duccio November 6, 2017 at 9:46 am #

Thank you very much. I will do that.

Brence November 7, 2017 at 7:13 am #

Jason,

Just wanted to stop by and say thanks again!

I've been tweaking my learning models for the past three days and this is what I got

Larger: 0.12 (0.36) MSE

That's pretty good right? I'm using a different data set from you, but it is very similar in structure. I was having a underfitting problem with my model for a while and I was getting like 500%(500%) error. I realized that I needed to make it a bit more complex. So I tripled my features, made my layers deeper and wider at the same time, as well as up the amount of epochs to 50000 and batch size to 10000. I also changed the

Question: Will I be able to get a smaller error% or is “Larger: 0.12 (0.36) MSE” about the lowest I can expect?

thanks again Jason,

Brence

Jason Brownlee November 7, 2017 at 9:56 am #

REPLY

Nice work.

I’m not sure of the limits of this problem, push as much as you have time/interest. In practice “good” is relative to what you have achieved previously. This is a good lesson in applied ML!

Tony November 10, 2017 at 7:22 pm #

Dear Sir.

I applied your code and used it to predict successfully.

As same as my last question. I want to add 1 more output: the age of house: has built in 5 years, 7 years independent.
This is not a classify problem as I know.
So, would you suggest a code or what should I do next to solve the problem, please ?

Thank you very much

Brence November 14, 2017 at 11:20 am #

REPLY

Is it possible to do a recursive multi step forecast prediction with this regression model?

I’m not sure how this code would fit into this.

Get Your Start in Machine Learning

prediction(t+1) = model(obs(t-1), obs(t-2), …, obs(t-n))
prediction(t+2) = model(prediction(t+1), obs(t-1), …, obs(t-n))

Jason Brownlee November 15, 2017 at 9:44 am #

REPLY

Yes, perhaps this post could be used a template:
https://machinelearningmastery.com/multi-step-time-series-forecasting-long-short-term-memory-networks-python/

Toby November 18, 2017 at 3:45 am #

REPLY

Hi Jason,
Thank you for your tutorial. I just tried running your sample code for step 2 but unfortunately obtained a r

Results: -57.82 (42.31) MSE

Any ideas?

The code is exactly the same with minor exception that I had to changed
model.compile(loss='mean_square_error',optimizer='adam')
to:
model.compile(loss='mse',optimizer='adam')

Thanks
Toby

Jason Brownlee November 18, 2017 at 10:24 am #

REPLY

Yes, sklearn inverts minimizing scores to make them maximizing for optimization. Just take the absolute value.

Toby November 22, 2017 at 7:04 am #

Great thanks. Yes I found this out after I posed the question.

REPLY

Jason Brownlee November 22, 2017 at 11:16 am #

Nice.

REPLY

Mohit Jain November 27, 2017 at 2:15 am #

Hi Jason,
Thank you this amazing tutorial. I just wanted to know what are the ways such that we can predict the ou
and compare the performance by plotting the predicted and actual value

Thanks

Mohit

Jason Brownlee November 27, 2017 at 5:51 am #

It really depends on the application as to what and how to plot.

Moritz December 1, 2017 at 8:12 pm #

REPLY

Hi Jason,
first: thanks for this and all your other amazing tutorials. Really helps a lot as a beginner to get actual useful advice
However, I reached a point where I'm looking for further advice – hope you can help me out!

I understand the concept of regression and MSE – in my case, I try to predict two values based on various other parameters. It's really not complicated and the correlation between the values is pretty clear, so I think this shouldn't be a problem.
Now when having a value predicted, I don't want to know the MSE but I'd rather know, if the prediction is within a certain range from the original value.
Example:
'accepted' range: y +/- 0,1
y = 1
y^ = 1,08
y – y^ = | 0,08| –> OK, because it's within y +/- 0,1.

Is there a way to do this in Python or KERAS? I just started working with it, so any advice would be helpful. Thanks!

Jason Brownlee December 2, 2017 at 8:54 am #

You can calculate a confidence interval for linear models.

I have an example for linear regression on time series here that might give you ideas:
https://machinelearningmastery.com/time-series-forecast-uncertainty-using-confidence-intervals-pyth

Moritz December 4, 2017 at 12:59 am #

Thanks! I will look into that.

chenys December 12, 2017 at 7:29 pm #

REPLY

Hi Jason,
Thank you for your tutorial.
I want to know

if a regression problem dataset a 10000 feature. the input_dim is so big …….but all the feature are meaningful(it's an procedure data) and can't be delete

how to change this example to handle my problem,and what should i care,is there any trick?

Jason Brownlee December 13, 2017 at 5:30 am #

REPLY

Perhaps you can use a projection such as PCA? SVD? or others?

Steve December 13, 2017 at 5:32 am #

REPLY

Hi Jason – Thank you for all these tutorials. These are awesome!!

Since the NN architecture is black box. Is there a way to access hidden layer data for debugging? When
different numbers. Thx again!

Jason Brownlee December 13, 2017 at 5:47 am #

You will get different numbers every time you run the same algorithm on the same data Ste
https://machinelearningmastery.com/randomness-in-machine-learning/

You can access the layers on the model as an array: model.layers I think.

Brence December 28, 2017 at 11:36 am #

Hey Jason,

Is it possible to get an prediction output for each column used in the dataset? Like for example the dataset was made up of
12 1 22 45
2 34 55 8 like this. Could I get it to give me four output numbers, one for each column in the dataset?

Get Your Start in Machine Learning

Jason Brownlee December 28, 2017 at 2:11 pm #

REPLY

Yes, you can call model.predict()

Brence December 28, 2017 at 10:17 pm #

REPLY

but how do I predict for more then one column in the dataset? My dataset has 6 of them, and my output always has just 5 columns. Which leads me to believe that its just predicting for one column instead of all 6. Could I accomplish this by setting the output layer to have more then one neuron?

Jason Brownlee December 29, 2017 at 5:22 am #

You could configure the model to predict a vector via the number of neurons in the

You could configure the model output one column at a time via an encoder-decoder model.

I have examples of each on the blog.

Brence December 29, 2017 at 9:00 am #

That is exactly what I was looking for. I found your examples on the blog. Thanl

Jason Brownlee December 29, 2017 at 2:35 pm #

Glad to hear it.

Jack January 2, 2018 at 6:48 pm #

REPLY

Hi Jason,
Thank you for your tutorial! I'm not a programmer or anything, in fact, I've never wriiten a line of code my entire life. But I find your tutorial very helpful. Recently I came acoss a regression problem and I tried to solve it using deep learning. So I followed this article and step by step I got Keras up and running and got a result. The problem is I don't know how to tune the neural network and optimize it. The result I got is far from satisfactory. Do I have to adjust the parameter of the model one by one and see how it goes or is there a quicker way to optimize the neural network?
Also I see there's a mini-course here, and I tried so sign up for it but I didn't get the email. Maybe because I'm from China or anything, I don't know. Is there any crash course I can get? cause I know nothing about Python yet…

Jason Brownlee January 3, 2018 at 5:33 am #

Well done!

This post will show you how to tune a network:
http://machinelearningmastery.com/improve-deep-learning-performance/

You can access the mini course here:
http://machinelearningmastery.com/applied-deep-learning-in-python-mini-course/

fatma January 3, 2018 at 7:26 pm #

Hello, I need to ask for this line X = dataset[:,0:13], as I can see from the data set it contains 14 columns (0 to 13) and the last column is the labels column then this line should be X = dataset [:,0:12]. is it correct or I'm wrong?

Jason Brownlee January 4, 2018 at 8:09 am #

REPLY

No. You can learn more about slicing arrays here:
https://machinelearningmastery.com/index-slice-reshape-numpy-arrays-machine-learning-python/

Tanya January 4, 2018 at 7:43 am #

REPLY

Hi Jason,

I am trying to apply the code in this tutorial to forecast my time series data. But since the beginning when I am trying to split the data into X and Y, I am getting an error “TypeError: unhashable type: ‘slice'”. Unfortunately I cannot find the source of it.
Can you help me?
Thanks in advance!

runfile(‘D:/LOCAL_DROPBOX/MasterArbeit_Sammlung_V01/Python/MasterArbeit/ARIMA/Test/BaselineRegressionKNN.py’,
wdir=’D:/LOCAL_DROPBOX/MasterArbeit_Sammlung_V01/Python/MasterArbeit/ARIMA/Test’)
[[‘3,6’ ‘20,3’ ‘0’ …, 173 1136 0]
[‘11,4’ ‘18,8’ ‘15,2’ …, 105 1676 0]
[‘8,9’ ‘15,3’ ‘1,4’ …, 372 733 0]
…,
[‘-2,3’ ‘4,5’ ‘0’ …, 0 0 0]
[‘0,2’ ‘7,9’ ‘0’ …, 0 0 0]
[‘-3,5’ ‘4,4’ ‘0’ …, 0 0 0]]
Traceback (most recent call last):

File “”, line 1, in
runfile(‘D:/LOCAL_DROPBOX/MasterArbeit_Sammlung_V01/Python/MasterArbeit/ARIMA/Test/Baseline
wdir=’D:/LOCAL_DROPBOX/MasterArbeit_Sammlung_V01/Python/MasterArbeit/ARIMA/Test’)

File “C:\Users\Tanya\Anaconda3\lib\site-packages\spyder\utils\site\sitecustomize.py”, line 710, in runfile
execfile(filename, namespace)

File “C:\Users\Tanya\Anaconda3\lib\site-packages\spyder\utils\site\sitecustomize.py”, line 101, in execfile
exec(compile(f.read(), filename, ‘exec’), namespace)

File “D:/LOCAL_DROPBOX/MasterArbeit_Sammlung_V01/Python/MasterArbeit/ARIMA/Test/BaselineRegressionKNN.py”, line 25, in
X = dataset[:,0:8]

File “C:\Users\Tanya\Anaconda3\lib\site-packages\pandas\core\frame.py”, line 2139, in __getitem__
return self._getitem_column(key)

Get Your Start in Machine Learning

File “C:\Users\Tanya\Anaconda3\lib\site-packages\pandas\core\frame.py”, line 2146, in _getitem_column
return self._get_item_cache(key)

File “C:\Users\Tanya\Anaconda3\lib\site-packages\pandas\core\generic.py”, line 1840, in _get_item_cache
res = cache.get(item)

TypeError: unhashable type: ‘slice’

Jason Brownlee January 4, 2018 at 8:17 am #

REPLY

You can learn more about array slicing here:
https://machinelearningmastery.com/index-slice-reshape-numpy-arrays-machine-learning-python/

Leave a Reply

Name (required)

Email (will not be published) (required)

Get Your Start in Machine Learning

Website

SUBMIT COMMENT

Welcome to Machine Learning Mastery

Hi, I'm Dr. Jason Brownlee.
My goal is to make practitioners like YOU awesome at applied machine learning.

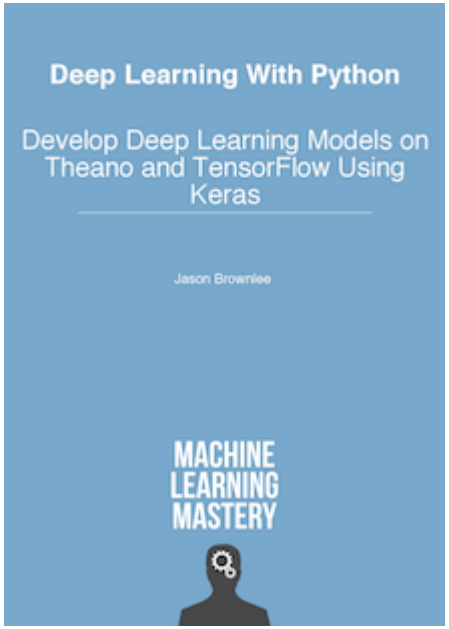
Read More

Finally Get Started With Deep Learning

Sick of the fancy math and need for super computers?
Looking for step-by-step tutorials?
Want end-to-end projects?

Get Your Start in Machine Learning

Get Started With Deep Learning in Python Today!

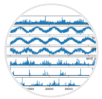


POPULAR


Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras
JULY 21, 2016


Your First Machine Learning Project in Python Step-By-Step
JUNE 10, 2016


Develop Your First Neural Network in Python With Keras Step-By-Step
MAY 24, 2016


Multivariate Time Series Forecasting with LSTMs in Keras
AUGUST 14, 2017

How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda
MARCH 13, 2017

Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**.
Find out how in this *free* and *practical* email course.

Email Address

START MY EMAIL COURSE

Get Your Start in Machine Learning

Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras
JULY 26, 2016

Time Series Forecasting with the Long Short-Term Memory Network in Python
APRIL 7, 2017

Regression Tutorial with the Keras Deep Learning Library in Python
JUNE 9, 2016

Multi-Class Classification Tutorial with the Keras Deep Learning Library
JUNE 2, 2016

How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras
AUGUST 9, 2016

Privacy | Contact | About