Multi-Class Classification Tutorial with the Keras Deep Learning Library

by **Jason Brownlee** on June 2, 2016 in **Deep Learning**

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

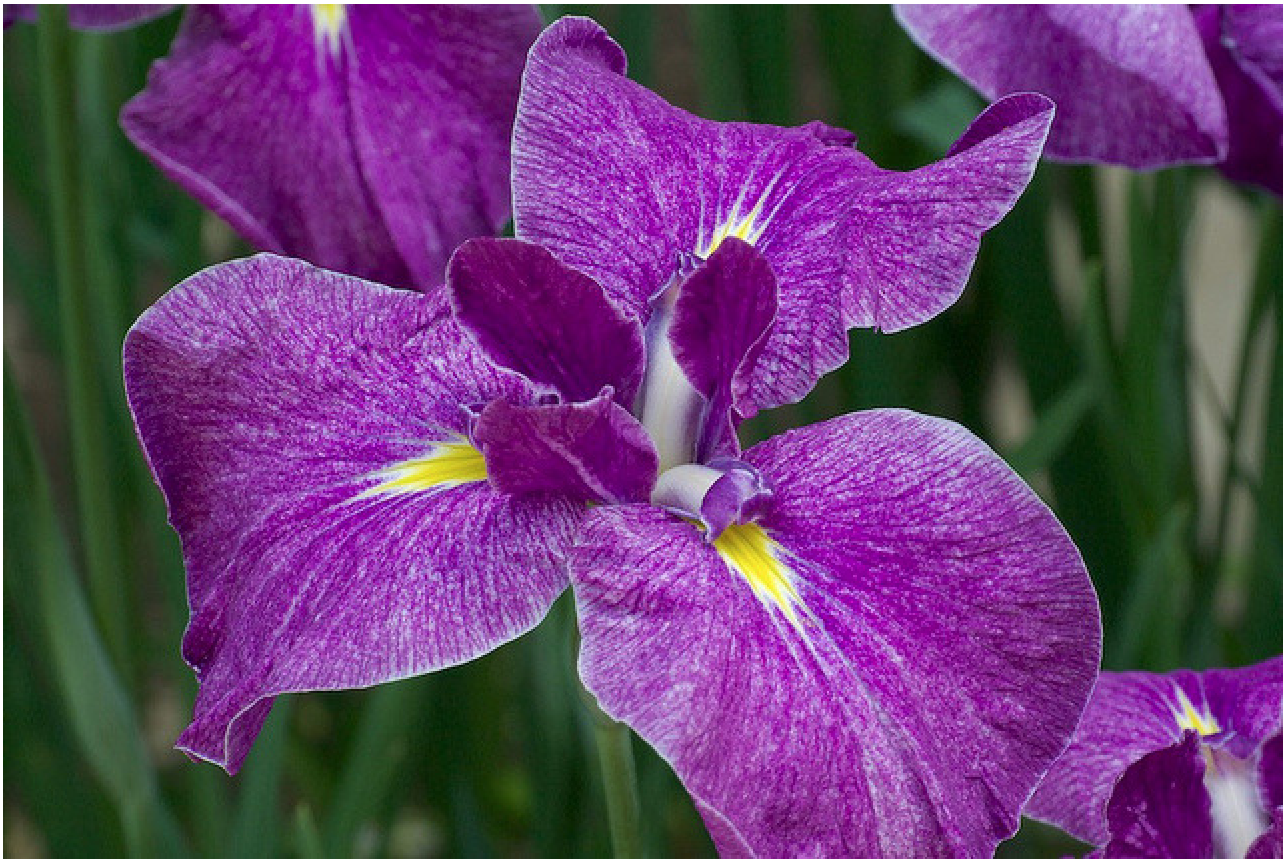
In this tutorial, you will discover how you can use Keras to develop and evaluate neural network models for multi-class classification problems.

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

- How to load data from CSV and make it available to Keras.
- How to prepare multi-class classification data for modeling with neural networks.
- How to evaluate Keras neural network models with scikit-learn.

Let's get started.

- **Update Oct/2016**: Updated examples for Keras 1.1.0 and scikit-learn v0.18.
- **Update Mar/2017**: Updated example for Keras 2.0.2, TensorFlow 1.0.1 and Theano 0.9.0.
- **Update Jun/2017**: Updated example to use softmax activation in output layer, larger hidden layer, default weight initialization.



Multi-Class Classification Tutorial with the Keras Deep Learning Library
Photo by houroumono, some rights reserved.

1. Problem Description

In this tutorial, we will use the standard machine learning problem called the iris flowers dataset.

This dataset is well studied and is a good problem for practicing on neural networks because all of the 4 input variables are numeric and have the same scale in centimeters. Each instance describes the properties of an observed flower measurements and the output variable is specific iris species.

This is a multi-class classification problem, meaning that there are more than two classes to be predicted, in fact there are three flower species. This is an important type of problem on which to practice with neural networks because the three class values require specialized handling.

The iris flower dataset is a well-studied problem and a such we can expect to achieve a model accuracy in the range of 95% to 97%. This provides a good target to aim for when developing our models.

You can download the iris flowers dataset from the UCI Machine Learning repository and place it in your current working directory with the filename “*iris.csv*“.

Need help with Deep

2. Import Classes and Functions

We can begin by importing all of the classes and functions we will need in this tutorial.

This includes both the functionality we require from Keras, but also data loading from pandas as well as data preparation and model evaluation from scikit-learn.

```
import numpy
import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
```

3. Initialize Random Number Generator

Next, we need to initialize the random number generator to a constant value (7).

This is important to ensure that the results we achieve from this model can be achieved again precisely. It ensures that the stochastic process of training a neural network model can be reproduced.

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

4. Load The Dataset

The dataset can be loaded directly. Because the output variable contains strings, it is easiest to load the data using pandas. We can then split the attributes (columns) into input variables (X) and output variables (Y).

```
# load dataset
dataframe = pandas.read_csv("iris.csv", header=None)
dataset = dataframe.values
X = dataset[:,0:4].astype(float)
Y = dataset[:,4]
```

5. Encode The Output Variable

The output variable contains three different string values.

When modeling multi-class classification problems using neural networks, it is good practice to reshape the output attribute from a vector that contains values for each class value to be a matrix with a boolean for each class value and whether or not a given instance has that class value or not.

This is called one hot encoding or creating dummy variables from a categorical variable.

For example, in this problem three class values are Iris-setosa, Iris-versicolor and Iris-virginica. If we had the observations:

```
Iris-setosa
Iris-versicolor
Iris-virginica
```

We can turn this into a one-hot encoded binary matrix for each data instance that would look as follows:

```
Iris-setosa,    Iris-versicolor,    Iris-virginica
1,      0,          0
0,      1,          0
0,      0,          1
```

We can do this by first encoding the strings consistently to integers using the scikit-learn class LabelEncoder. Then convert the vector of integers to a one hot encoding using the Keras function to_categorical().

```
# encode class values as integers
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
# convert integers to dummy variables (i.e. one hot encoded)
dummy_y = np_utils.to_categorical(encoded_Y)
```

6. Define The Neural Network Model

The Keras library provides wrapper classes to allow you to use neural network models developed with Keras in scikit-learn.

There is a KerasClassifier class in Keras that can be used as an Estimator in scikit-learn, the base type of model in the library. The KerasClassifier takes the name of a function as an argument. This function must return the constructed neural network model, ready for training.

Below is a function that will create a baseline neural network for the iris classification problem. It creates a simple fully connected network with one hidden layer that contains 8 neurons.

The hidden layer uses a rectifier activation function which is a good practice. Because we used a one-hot encoding for our iris dataset, the output layer m e output value with the largest value will be taken as the class predicted by the model.

The network topology of this simple one-layer neural network can be summarized as:

```
4 inputs -> [8 hidden nodes] -> 3 outputs
```

Note that we use a “*softmax*” activation function in the output layer. This is to ensure the output values are in the range of 0 and 1 and may be used as predicted probabilities.

Finally, the network uses the efficient Adam gradient descent optimization algorithm with a logarithmic loss function, which is called “*categorical_crossentropy*” in Keras.

```
# define baseline model
def baseline_model():
    # create model
    model = Sequential()
    model.add(Dense(8, input_dim=4, activation='relu'))
    model.add(Dense(3, activation='softmax'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

We can now create our KerasClassifier for use in scikit-learn.

We can also pass arguments in the construction of the KerasClassifier class that will be passed on to the fit() function internally used to train the neural network. Here, we pass the number of epochs as 200 and batch size as 5 to use when training the model. Debugging is also turned off when training by setting verbose to 0.

```
estimator = KerasClassifier(build_fn=baseline_model, epochs=200, batch_size=5, verbose=0)
```

7. Evaluate The Model with k-Fold Cross Validation

We can now evaluate the neural network model on our training data.

The scikit-learn has excellent capability to evaluate models using a suite of techniques. The gold standard for evaluating machine learning models is k-fold cross validation.

First we can define the model evaluation procedure. Here, we set the number of folds to be 10 (an excellent default) and to shuffle the data before partitioning it.

```
kfold = KFold(n_splits=10, shuffle=True, random_state=seed)
```

Now we can evaluate our model (estimator) on our dataset (X and dummy_y) using a 10-fold cross-validation procedure (kfold).

Evaluating the model only takes approximately 10 seconds and returns an object that describes the evaluation of the 10 constructed models for each of the splits of the dataset.

```
results = cross_val_score(estimator, X, dumm
print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
```

The results are summarized as both the mean and standard deviation of the model accuracy on the dataset. This is a reasonable estimation of the performance of the model on unseen data. It is also within the realm of known top results for this problem.

```
Accuracy: 97.33% (4.42%)
```

Summary

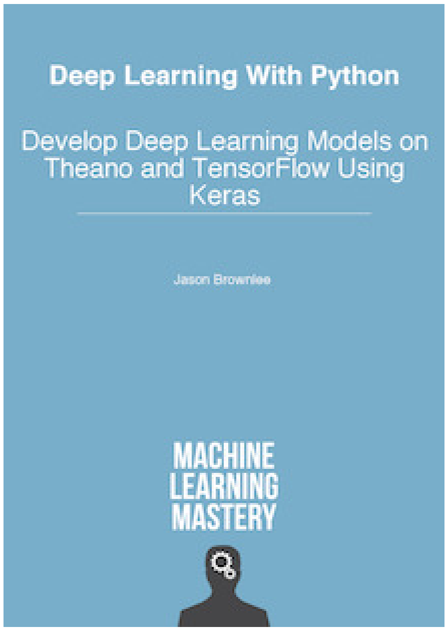
In this post you discovered how to develop and evaluate a neural network using the Keras Python library for deep learning.

By completing this tutorial, you learned:

- How to load data and make it available to Keras.
- How to prepare multi-class classification data for modeling using one hot encoding.
- How to use Keras neural network models with scikit-learn.
- How to define a neural network using Keras for multi-class classification.
- How to evaluate a Keras neural network model using scikit-learn with k-fold cross validation

Do you have any questions about deep learning with Keras or this post?

Ask your questions in the comments below and I will do my best to answer them.



Deep Learning With Python

Develop Deep Learning Models on Theano and TensorFlow Using Keras

Jason Brownlee

MACHINE LEARNING MASTERY

About Jason Brownlee

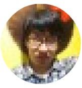
Dr. Jason Brownlee is a husband, proud father, academic researcher, author, professional developer and a machine learning practitioner. He is dedicated to helping developers get started and get good at applied machine learning. Learn more.

View all posts by Jason Brownlee →

‹ How To Compare Machine Learning Algorithms in Python with scikit-learn

Ensemble Machine Learning Algorithms in Python with scikit-learn ›

223 Responses to *Multi-Class Classification Tutorial with the Keras Deep Learning Library*



Jack June 19, 2016 at 3:12 pm #

REPLY

Thanks for this cool tutorial! I have a question about the input data. If the datatypes of input variables are different (i.e. string and numeric). How to preprocess the train data to fit keras?

Jason Brownlee June 20, 2016 at 5:41 am #

REPLY

Great question. Eventually, all of the data need to be turned into real values.

With categorical variables, you can create dummy variables and use one-hot encoding. For string data, you can use word embeddings.

Shraddha February 10, 2017 at 8:32 pm #

REPLY

Could you please let me know how to convert string data into word embeddings in large datasets?
Would really appreciate it
Thanks so much

Jason Brownlee February 11, 2017 at 5:01 am # REPLY ↰

Hi Shraddha,

First, convert the chars to vectors of integers. You can then pad all vectors to the same length. Then away you go.

I hope that helps.

Shraddha Sunil February 13, 2017 at 4:52 pm #

Thanks so much Jason!

Jason Brownlee February 14, 2017 at 10:04 am #

You're welcome.

sasi August 5, 2017 at 7:51 pm #

can you give an example for that..

Jason Brownlee August 6, 2017 at 7:38 am #

I have many tutorials for encoding and padding sequences on the blog. Please use the search.

Aakash Nain July 4, 2016 at 2:25 pm # REPLY ↰

Hello Jason,
It's a very nice tutorial to learn. I implemented the same model but on my work station I achieved a score of 88.67% only. After modifying the number of hidden layers, I achieved an accuracy of 93.04%. But I am not able to achieve the score of 95% or above. Any particular reason behind it ?

Jason Brownlee July 6, 2016 at 6:27 am # REPLY ↰

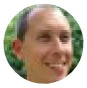
Interesting Aakash.

I used the Theano backend. Are you using the same?

Are all your libraries up to date? (Keras, Theano, NumPy, etc…)

Aakash Nain July 7, 2016 at 12:03 am # REPLY ↰

Yes Jason . Backend is theano and all libraries are up to date.



Jason Brownlee July 7, 2016 at 9:40 am # REPLY ↰

Interesting. Perhaps seeding the random number generator is not having the desired effect for reproducibility. It perhaps it has different effects on different platforms.

Perhaps re-run the above code example a few times and see the spread of accuracy scores you achieve?

La Tuan Nghia July 6, 2016 at 1:29 am # REPLY ↰

Hello Jason,

In chapter 10 of the book “Deep Learning With Python”, there is a fraction of code:

```
estimator = KerasClassifier(build_fn=baseline_model, nb_epoch=200, batch_size=5, verbose=0)
kfold = KFold(n=len(X), n_folds=10, shuffle=True, random_state=seed)
results = cross_val_score(estimator, X, dummy_y, cv=kfold)
print(“Accuracy: %.2f%% (%.2f%%)” % (results.mean()*100, results.std()*100))
```

How to save this model and weights to file, then how to load these file to predict a new input data?

Many thanks!

Jason Brownlee July 6, 2016 at 6:26 am #

REPLY

Really good question.

Keras does provide functions to save network weights to HDF5 and network structure to JSON or YAML. The problem is, once you wrap the network in a scikit-learn classifier, how do you access the model and save it. Or can you save the whole wrapped model.

Perhaps a simple but inefficient place to start would be to try and simply pickle the whole classifier?
https://docs.python.org/2/library/pickle.html

Constantin Weisser July 30, 2016 at 4:30 am #

REPLY

I tried doing that. It works for a normal sklearn classifier, but apparently not for a Keras Classifier:

```
import pickle
with open(“name.p”,”wb”) as fw:
pickle.dump(clf,fw)

with open(name+”.p”,”rb”) as fr:
clf_saved = pickle.load(fr)
print(clf_saved)

prob_pred=clf_saved.predict_proba(X_test)[:,1]
```

This gives:

```
theano.gof.fg.MissingInputError: An input of the graph, used to compute DimShuffle{x,x}(keras_learning_phase), was not provided and not given a value.Use the Theano flag exception_verbosity=’high’,for more information on this error.

Backtrace when the variable is created:
File “nn_systematics_I_evaluation_of_optimised_classifiers.py”, line 6, in
import classifier_eval_simplified
File “../../../../classifier_eval_simplified.py”, line 26, in
from keras.utils import np_utils
File “/usr/local/lib/python2.7/site-packages/keras/__init__.py”, line 2, in
from . import backend
File “/usr/local/lib/python2.7/site-packages/keras/backend/__init__.py”, line 56, in
from .theano_backend import *
File “/usr/local/lib/python2.7/site-packages/keras/backend/theano_backend.py”, line 17, in
_LEARNING_PHASE = T.scalar(dtype=’uint8′, name=’keras_learning_phase’) # 0 = test, 1 = train
```

Jason Brownlee July 30, 2016 at 7:12 am # REPLY ↰

I provide examples of saving and loading Keras models here:
http://machinelearningmastery.com/save-load-keras-deep-learning-models/

Sorry, I don’t have any examples of saving/loading the wrapped Keras classifier. Perhaps the internal model can be seralized and later deserialized and put back inside the wrapper.

Sally July 15, 2016 at 4:10 am # REPLY ↰

Dear Dr. Jason,

Thanks very much for this great tutorial . I got extra benefit from it, but I need to calculate precision, recall and confusion matrix for such multi-class classification. I tried to did it but each time I got a different problem. could you please explain me how to do this

Jason Brownlee July 15, 2016 at 9:04 am # REPLY ↰

Hi Sally, you could perhaps use the tools in scikit-learn to summarize the performance of your model.

For example, you could use sklearn.metrics.confusion_matrix() to calculate the confusion matrix for predictions, etc.

See the metrics package:
http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Fabian Leon July 31, 2016 at 4:12 am # REPLY ↰

Hi jason, Reading the tutorial and the same example in your book, you still don’t tell us how can use the model to make predictions, you have only show us how to train and evaluate it but I would like to see you using this model to make predictions on at least one example of iris flowers data no matters if is dummy data.

I would like to see how can I load my own instance of an iris-flower and use the above model to predict what kind is the flower?

could you do that for us?

Jason Brownlee July 31, 2016 at 7:31 am REPLY ↰

Hi Fabian, no problem.

In the tutorial above, we are using the scikit-learn wrapper. That means we can use the standard model.predict() function to make predictions from scikit-learn.

For example, below is an an example adapted from the above where we split the dataset, train on 67% and make predictions on 33%. Remember that we have encoded the output class value as integers, so the predictions are integers. We can then use encoder.inverse_transform() to turn the predicted integers back into strings.

```
# Train model and make predictions
import numpy
import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
from sklearn.cross_validation import train_test_split
from sklearn.preprocessing import LabelEncoder
# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
# load dataset
dataframe = pandas.read_csv("iris.csv", header=None)
dataset = dataframe.values
X = dataset[:,0:4].astype(float)
Y = dataset[:,4]
# encode class values as integers
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
# convert integers to dummy variables (i.e. one hot encoded)
dummy_y = np_utils.to_categorical(encoded_Y)
# define baseline model
def baseline_model():
    # create model
    model = Sequential()
    model.add(Dense(4, input_dim=4, init='normal', activation='relu'))
    model.add(Dense(3, init='normal', activation='sigmoid'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accu
    return model
estimator = KerasClassifier(build_fn=baseline_model, nb_epoch=200, batch_size=5, ve
X_train, X_test, Y_train, Y_test = train_test_split(X, dummy_y, test_size=0.33, ran
estimator.fit(X_train, Y_train)
predictions = estimator.predict(X_test)
print(predictions)
print(encoder.inverse_transform(predictions))
```

Running this example prints:

```
[2 1 0 1 2 0 1 1 0 1 2 1 0 2 0 2 2 2 0 0 1 2 1 2 2 2 1 1 2 2 2 1 0 2 1 0 0
 0 0 2 2 1 2 2 1 0 1 1 2 0]
['Iris-virginica' 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor'
 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-virginica' 'Iris-setosa' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'I
 'Iris-virginica' 'Iris-virginica' '
'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
'Iris-virginica' 'Iris-setosa']
```

I hope that is clear and useful. Let me know if you have any more questions.

Devendra November 27, 2016 at 9:40 pm #

REPLY

Hi Jason,

I was facing error while converting string to float and so I had to make a minor correction to my code
X = dataset[1:,0:4].astype(float)
Y = dataset[1:,4]

However, I am still unable to run since I am getting the following error for line

“—-> 1 results = cross_val_score(estimator, X, dummy_y, cv=kfold)”
………………..
“Exception: Error when checking model target: expected dense_4 to have shape (None, 3) but got array with shape (135L, 22L)”

I would appreciate your help. Thanks.

Devendra November 28, 2016 at 5:41 am #

REPLY

I found the issue. It was with with the indexes.
I had to take [1:,1:5] for X and [1:,5] for Y.

I am using Jupyter notebook to run my code.
The index range seems to be different in my case.

Jason Brownlee November 28, 2016 at 8:47 am #

I’m glad you worked it out Devendra.

Cristina March 24, 2017 at 2:23 am #

REPLY

For some reason, when I run this example I get 0 as prediction value for all the samples. What could be happening?

I've the same problem on prediction with other code I'm executing, and decided to run yours to check if i could be doing something wrong?

I'm lost now, this is very strange.

Thanks a in advance!

Cristina March 24, 2017 at 2:42 am #

REPLY ↰

Hello again,

This is happening with Keras 2.0, with Keras 1 works fine.

Thanks,

Cristina

Jason Brownlee March 24, 2017 at 8:00 am #

Thanks for the note.

Jason Brownlee March 24, 2017 at 7:57 am #

REPLY ↰

Very strange.

Maybe check that your data file is correct, that you have all of the code and that your environment is installed and is working correctly.

Tanvir. March 27, 2017 at 7:43 am #

REPLY ↰

Hi Jason,
Thanks for your awesome tutorials. I had a curious question:
As we are using KerasClassifier or KerasRegressor of Scikit-Learn wrapper, then how to save them as a file after fitting ?

For example, I am predicting regression or multiclass classification. I have to use KerasRegressor or KerasClassifier then. After fitting a large volume of data, I want to save the trained neural network model to use it for prediction purpose only. How to save them and how to restore them from saved files ? Your answer will help me a lot.

Jason Brownlee March 27, 2017 at 8:00 am #

REPLY

Great question, I'm not sure you can easily do this. You might be better served fitting the Keras model directly then using the Keras API to save the model:
http://machinelearningmastery.com/save-load-keras-deep-learning-models/

Reinier May 4, 2017 at 2:04 am #

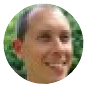
REPLY

Hi Jason, Thank your very much for those nice explainations.
I'm having some problems and I trying very hard to get it solved but it wont work..
If I simply copy-past your code from your comment on 31-july 2016 I keep getting the following Error:

Traceback (most recent call last): File "/Users/reinier/PycharmProjects/Test-IRIS/TESTIRIS.py", line 43, in estimator.fit(X_train, Y_train) File "/Users/reinier/Library/Python/3.6/lib/python/site-packages/keras/wrappers/scikit_learn.py", line 206, in fit return super(KerasClassifier, self).fit(x, y, **kwargs) File "/Users/reinier/Library/Python/3.6/lib/python/site-packages/keras/wrappers/scikit_learn.py", line 149, in fit history = self.model.fit(x, y, **fit_args) File "/Users/reinier/Library/Python/3.6/lib/python/site-packages/keras/models.py", line 856, in fit initial_epoch=initial_epoch) File "/Users/reinier/Library/Python/3.6/lib/python/site-packages/keras/engine/training.py", line 1429, in fit batch_size=batch_size) File "/Users/reinier/Library/Python/3.6/lib/python/site-packages/keras/engine/training.py", line 1309, in _standardize_user_data exception_prefix='target') File "/Users/reinier/Library/Python/3.6/lib/python/site-packages/keras/engine/training.py", line 139, in _standardize_input_data str(array.shape)) ValueError: Error when checking target: expected dense_2 to have shape (None, 3) but got array with shape (67, 40)

It seems like something is wrong with the fit function. Is this the cause of a new Keras version?
Thanks you very much in advance,

Reinier



Jason Brownlee May 4, 2017 at 8:09 am #

REPLY

Sorry, it is not clear what is going on.

Does the example in the blog post work as expected?

Priyesh July 12, 2017 at 3:02 am #

REPLY

Hello Jason,

Thank you for such a wonderful and detailed explanation. Please can guide me on how to plot the graphs for clustering for this data set and code (both for training and predictions).

Thanks.

Jason Brownlee July 12, 2017 at 9:50 am # REPLY ↰

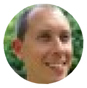
Sorry, I do not have examples of clustering.

Priyesh July 12, 2017 at 5:12 am # REPLY ↰

Hi Jason,

Thank you so much for such an elegant and detailed explanation. I wanted to learn on how to plot graphs for the same. I went through the comments and you said we can't plot accuracy but I wish to plot the graphs for input data sets and predictions to show like a cluster (as we show K-means like a scattered plot). Please can you guide me with the same.

Thank you.



Jason Brownlee July 12, 2017 at 9:53 am # REPLY ↰

Sorry I do not have any examples for clustering.

Prash August 14, 2016 at 9:15 pm # REPLY ↰

Jason, boss you are too good! You have really helped me out especially in implementation of Deep learning part. I was rattled and lost and was desperately looking for some technology and came across your blogs. thanks a lot.

Jason Brownlee August 15, 2016 at 12:38 pm # REPLY ↰

I'm glad I have helped in some small way Prash.

Harsha August 18, 2016 at 7:03 pm # REPLY ↰

It is a great tutorial Dr. Jason. Very clear a doubt.

Is it necessary to use scikit-learn. Can we solve the same problem using basic keras?

Jason Brownlee August 19, 2016 at 5:25 am # REPLY ↰

You can use basic Keras, but scikit-learn make Keras better. They work very well together.

Harsha August 19, 2016 at 11:06 pm # REPLY ↰

Thank You Jason for your prompt reply

Jason Brownlee August 20, 2016 at 6:05 am # REPLY ↰

You're welcome Harsha.

jokla January 12, 2017 at 7:30 am # REPLY ↰

Hi Jason, nice tutorial!

I have a question. You mentioned that scikit-learn make Keras better, why?

Thanks!

Jason Brownlee January 12, 2017 at 9:40 am # REPLY ↰

Hi jokla, great question.

The reason is that we can access all of sklearn's features using the Keras Wrapper classes. Tools like grid searching, cross validation, ensembles, and more.

moeyzf August 21, 2016 at 10:17 am # REPLY ↰

Hi Jason,

I'm a CS student currently studying sentiment analysis and was wondering how to use keras for multi classification of text, ideally I would like the functior vector representation against a given vocabulary is classification.

I am having trouble understanding the initial steps in transforming and feeding word data into vector representations. Can you help me out with some basic code examples of this first step in the sense that say I have a text file with 5000 words for example, which also include emoji (to use as the vocabulary), how can I feed in a training file in csv format text,sentiment and convert each text into a one hot representation then feed it into the neural net, for a final output vector of size e.g 1×7 to denote the various class labels.

I have tried to find help online and most of the solutions use helper methods to load in text data such as imdb, while others use word2vec which isnt what i need.

Hope you can help, I would really appreciate it!

Cheers,

Mo

Qichang September 12, 2016 at 3:01 pm #

REPLY

Hi Jason,

Thanks for the great tutorial!

Just one question regarding the output variable encoding. You mentioned that it is a good practice to convert the output variable to one hot encoding matrix. Is this a necessary step? If the output varible consists of discrete integters, say 1, 2, 3, do we still need to to_categorical() to perform one hot encoding?

I check some example codes in keras github, it seems this is required. Can you please kindly shed some lights on it?

Thanks in advance.

Jason Brownlee September 13, 2016 at 8:09 am #

REPLY

Hi Qichang, great question.

A one hot encoding is not required, you can train the network to predict an integer, it is just a MUCH harder problem.

By using a one hot encoding, you greatly simplify the prediction problem making it easier to train for and achieve better performance.

Try it and compare the results.

Pedro A. Castillo September 16, 2016 at 12:31 am #

REPLY

Hello,
I have followed your tutorial and I get an error in the following line:

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

```
Traceback (most recent call last):
File “k.py”, line 84, in
results = cross_val_score(estimator, X, dummy_y, cv=kfold)
File “/Library/Python/2.7/site-packages/scikit_learn-0.17.1-py2.7-macosx-10.9-intel.egg/sklearn/cross_validation.py”, line 1433, in cross_val_score
for train, test in cv)
File “/Library/Python/2.7/site-packages/scikit_learn-0.17.1-py2.7-macosx-10.9-intel.egg/sklearn/externals/joblib/parallel.py”, line 800, in __call__
while self.dispatch_one_batch(iterator):
File “/Library/Python/2.7/site-packages/scikit_learn-0.17.1-py2.7-macosx-10.9-intel.egg/sklearn/externals/joblib/parallel.py”, line 658, in dispatch_one_batch
self._dispatch(tasks)
File “/Library/Python/2.7/site-packages/scikit_learn-0.17.1-py2.7-macosx-10.9-intel.egg/sklearn/externals/joblib/parallel.py”, line 566, in _dispatch
job = ImmediateComputeBatch(batch)
File “/Library/Python/2.7/site-packages/scikit_learn-0.17.1-py2.7-macosx-10.9-intel.egg/sklearn/externals/joblib/parallel.py”, line 180, in __init__
self.results = batch()
File “/Library/Python/2.7/site-packages/scikit_learn-0.17.1-py2.7-macosx-10.9-intel.egg/sklearn/externals/joblib/parallel.py”, line 72, in __call__
return [func(*args, **kwargs) for func, args, kwargs in self.items]
File “/Library/Python/2.7/site-packages/scikit_learn-0.17.1-py2.7-macosx-10.9-intel.egg/sklearn/cross_validation.py”, line 1531, in _fit_and_score
estimator.fit(X_train, y_train, **fit_params)
File “/Library/Python/2.7/site-packages/keras/wrappers/scikit_learn.py”, line 135, in fit
**self.filter_sk_params(self.build_fn.__call__))
TypeError: __call__() takes at least 2 arguments (1 given)
```

Do you have received this error before? do you have an idea how to fix that?

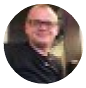
Jason Brownlee September 16, 2016 at 9:07 am #

REPLY

I have not seen this before Pedro.

Perhaps it is something simple like a copy-paste error from the tutorial?

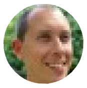
Are you able to double check the code matches the tutorial exactly?



Victor October 8, 2016 at 10:15 pm #

REPLY ↩

I have exactly the same problem.
Double checked the code,
have all the versions of keras etc, updated.
🙁



Jason Brownlee October 9, 2016 at 6:50 am #

REPLY ↩

Hi Victor, are you able to share your version of Keras, scikit-learn, TensorFlow/Theano?

Yunita September 25, 2016 at 12:17 am #

REPLY ↩

Hi Jason,

Thanks for the great tutorial.
But I have a question, why did you use sigmoid activation function together with categorical_crossentropy loss function?
Usually, for multiclass classification problem, I found implementations always using softmax activation function with categorical_cross entropy.
In addition, does one-hot encoding in the output make it as binary classification instead of multiclass classification? Could you please give some explanations on it?

Jason Brownlee September 25, 2016 at 8:04 am #

REPLY ↩

Yes, you could use a softmax instead of sigmoid. Try it and see.

The one hot encoding creates 3 binary output features. This too would be required with the softmax activation function.

Preston September 12, 2017 at 11:14 pm # REPLY ↩

Jason,

Great site, great resource. Is it possible to see the old example with the one hot encoding output? I'm interested in creating a network with multiple binary outputs and have been searching around for an example.

Many thanks.

Jason Brownlee September 13, 2017 at 12:31 pm # REPLY ↩

I have many examples on the blog of categorical outputs from LSTMs, try the search.

Preston September 14, 2017 at 5:40 am #

Thank you.

Marcus September 26, 2016 at 6:49 am # REPLY ↩

For Text classification or to basically assign them a category based on the text. How would the baseline_model change????

I'm trying to have an inner layer of 24 nodes and an output of 17 categories but the input_dim=4 as specified in the tutorial wouldn't be right cause the text length will change depending on the number of words.

I'm a little confused. Your help would be much appreciated.

```
model.add(Dense(24, init='normal', activation='relu'))
```

```
def baseline_model():
# create model
model = Sequential()
model.add(Dense(24, init='normal', activation='relu'))
model.add(Dense(17, init='normal', activation='sigmoid'))
# Compile model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
return model
```

Jason Brownlee September 26, 2016 at 7:01 am # REPLY ↰

You will need to use padding on the input vectors of encoded words.

See this post for an example of working with text:
http://machinelearningmastery.com/predict-sentiment-movie-reviews-using-deep-learning/

Vishnu October 19, 2016 at 9:07 pm # REPLY ↰

Hi Jason,

Thank you for your tutorial. I was really interested in Deep Learning and was looking for a place to start, this helped a lot.

But while I was running the code, I came across two errors. The first one was, that while loading the data through pandas, just like your code i set “header= None” but in the next line when we convert the value to float i got the following error message.

“ValueError: could not convert string to float: ‘Petal.Length'”.

This problem went away after I took the header=None condition off.

The second one came at the end, during the Kfold validation. during the one hot encoding it’s binning the values into 22 categories and not 3. which is causing this error:

“Exception: Error when checking model target: expected dense_2 to have shape (None, 3) but got array with shape (135, 22)”

I haven’t been able to get around this. Any suggestion would be appreciated.

Jason Brownlee October 20, 2016 at 8:36 am # REPLY ↰

That is quite strange Vishnu, I think perhaps you have the wrong dataset.

You can download the CSV here:
http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data

Homagni Saha October 20, 2016 at 10:39 am # REPLY ↰

Hello, I tried to use the exact same code for another dataset , the only difference being the dataset had 78 columns and 100000 rows . I had to predict the last column taking the remaining 77 columns as features . I must also say that the last column has 23 different classes.(types basically) and the 23 different classes are all integers not strings like you have used

model = Sequential()

```
model.add(Dense(77, input_dim=77, init='normal', activation='relu'))
model.add(Dense(10, init='normal', activation='relu'))
model.add(Dense(23, init='normal', activation='sigmoid'))
```

also I used nb_epoch=20 and batch_size=1000

also in estimator I changed the verbose to 1, and now the accuracy is a dismal of 0.52% at the end. Also while running I saw strange outputs in the verbose as :

```
93807/93807 [==============================] – 0s – loss: nan – acc: 0.0052
```

why is the loss always as loss: nan ??

Can you please tell me how to modify the code to make it run correctly for my dataset?(remaining everything in the code is unchanged)

Jason Brownlee October 21, 2016 at 8:30 am # REPLY

Hi Homagni,

That is a lot of classes for 100K records. If you can reduce that by splitting up the problem, that might be good.

Your batch size is probably too big and your number of epochs is way too small. Dramatically increase the number of epochs bu 2-3 orders of magnitude.

Start there and let me know how you go.

AbuZekry October 30, 2016 at 12:02 am # REPLY

Hi Jason,

I've edited the first layer's activation to 'softplus' instead of 'relu' and number of neurons to 8 instead of 4 Then I edited the second layer's activation to 'softmax' instead of sigmoid and I got 97.33% (4.42%) performance. Do you have an explanation to this enhancement in performance ?

Jason Brownlee October 30, 2016 at 8:55 am # REPLY

Well done AbuZekry.

Neural nets are infinitely configurable.

Panand November 7, 2016 at 3:58 am # REPLY

Hello Jason,

Is there a error in your code? You said the network has 4 input neurons , 4 hidden neurons and 3 output neurons.But in the code you haven't added the hidden neurons.You just specified only the input and output neurons… Will it effect the output in anyway?

Jason Brownlee November 7, 2016 at 7:18 am #

REPLY

Hi Panand,

The network structure is as follows:

```
4 inputs -> [4 hidden nodes] -> 3 outputs
```

Line 5 of the code in section 6 adds both the input and hidden layer:

```
model.add(Dense(4, input_dim=4, init='normal', activation='relu'))
```

The input_dim argument defines the shape of the input.

JD November 13, 2016 at 5:28 pm #

REPLY

Hi Jason,
I have a set of categorical features and continuous features, I have this model:
model = Sequential()
model.add(Dense(117, input_dim=117, init='normal', activation='relu'))
model.add(Dense(10, activation='softmax'))

I am getting a dismal : ('Test accuracy:', 0.43541752685249119) :
Details:
Total records 45k, 10 classes to predict
batch_size=1000, nb_epoch=25

Any improvements also I would like to put LSTM how to go about doing that as I am getting errors if I add
model.add(Dense(117, input_dim=117, init='normal', activation='relu'))
model.add(LSTM(117,dropout_W=0.2, dropout_U=0.2, return_sequences=True))
model.add(Dense(10, activation='softmax'))
Error:
Exception: Input 0 is incompatible with layer lstm_6: expected ndim=3, found ndim=2

Jason Brownlee November 14, 2016 at 7:41 am #

REPLY

Hi JD,

Here is a long list of ideas to improve the skill o http://machinelearningmastery.com/improve-deep-learning-performance/

Not sure about the exception, you may need to double check the input dimensions of your data and confirm that your model definition matches.

YA November 17, 2016 at 7:00 pm #

REPLY ↩

Hi Jason,

I have a set of categorical features(events) from a real system, and i am trying to build a deep learning model for event prediction.
The event's are not appears equally in the training set and one of them is relatively rare compared to the others.
event count in training set
1 22000
2 6000
3 13000
4 12000
5 26000

Should i continue with this training set? or should i restructure the training set?
What is your recommendation?

Jason Brownlee November 18, 2016 at 8:20 am #

REPLY ↩

Hi YA, I would try as many different "views" on your problem as you can think of and see which best exposes the problem to the learning algorithms (gets the best performance when everything else is held constant).

Tom December 9, 2016 at 12:13 am #

REPLY ↩

Hello Jason,
Great work on your website and tuturials! I was wondering if you could show a multi hot encoding, I think you can call it al multi label classification.
Now you have (only one option on and the rest off)
[1,0,0]
[0,1,0]
[0,0,1]

And do like (each classification has the option on or off)
[0,0,0]
[0,1,1]

[1,0,1]
[1,1,0]
[1,1,1]
etc..

This would really help for me
Thanks!!

Tom December 9, 2016 at 1:07 am #

REPLY ↩

Extra side note, with k-Fold Cross Validation. I got it working with binary_crossentropy with quite bad results. Therefore I wanted to optimize the model and add cross validation which unfortunately didn't work.

Martin December 26, 2016 at 6:02 pm #

REPLY ↩

Hi, Jason: Regarding this, I have 2 questions:
1) You said this is a "simple one-layer neural network". However, I feel it's still 3-layer network: input layer, hidden layer and output layer.

4 inputs -> [4 hidden nodes] -> 3 outputs

2) However, in your model definition:
model.add(Dense(4, input_dim=4, init='normal', activation='relu'))
model.add(Dense(3, init='normal', activation='sigmoid'))

Seems that only two layers, input and output, there is no hidden layer. So this is actually a 2-layer network. Is this right?

Jason Brownlee December 27, 2016 at 5:24 am #

REPLY ↩

Hi Martin, yes. One hidden layer. I take the input and output layers as assumed, the work happens in the hidden layer.

The first line defines the number of inputs (input_dim=4) AND the number of nodes in the hidden layer:

```
model.add(Dense(4, input_dim=4, init='normal', activation='relu'))
```

I hope that helps.

Seun January 16, 2017 at 3:58 pm #

REPLY ↩

Hi, Jason: I ran this same code but got this error:

```
Traceback (most recent call last):

File “”, line 1, in
runfile(‘C:/Users/USER/Documents/keras-master/examples/iris_val.py’, wdir=’C:/Users
/USER/Documents/keras-master/examples’)

File “C:\Users\USER\Anaconda2\lib\site-packages\spyder\utils\site\sitecustomize.py”, line 866, in runfile
execfile(filename, namespace)

File “C:\Users\USER\Anaconda2\lib\site-packages\spyder\utils\site\sitecustomize.py”, line 87, in execfile
exec(compile(scripttext, filename, ‘exec’), glob, loc)

File “C:/Users/USER/Documents/keras-master/examples/iris_val.py”, line 46, in
results = cross_val_score(estimator, X, dummy_y, cv=kfold)

File “C:\Users\USER\Anaconda2\lib\site-packages\sklearn\model_selection\_validation.py”, line 140, in
cross_val_score
for train, test in cv_iter)

File “C:\Users\USER\Anaconda2\lib\site-packages\sklearn\externals\joblib\parallel.py”, line 758, in
__call__
while self.dispatch_one_batch(iterator):

File “C:\Users\USER\Anaconda2\lib\site-packages\sklearn\externals\joblib\parallel.py”, line 603, in
dispatch_one_batch
tasks = BatchedCalls(itertools.islice(iterator, batch_size))

File “C:\Users\USER\Anaconda2\lib\site-packages\sklearn\externals\joblib\parallel.py”, line 127, in
__init__
self.items = list(iterator_slice)

File “C:\Users\USER\Anaconda2\lib\site-packages\sklearn\model_selection\_validation.py”, line 140, in
for train, test in cv_iter)

File “C:\Users\USER\Anaconda2\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’
```

Please, I need your help on how to resolve this.

Jason Brownlee January 17, 2017 at 7:35 am #

REPLY ↩

Hi Seun, it is not clear what is going on here.

You may have added an additional line or whitespace or perhaps your environment has a problem?

David January 25, 2017 at 3:07 am # REPLY

Hello Seun, perhaps this could help you: http://stackoverflow.com/questions/41796618/python-keras-cross-val-score-error/41832675#41832675

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

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

shazz January 25, 2017 at 7:36 am # REPLY

I have the same issue….
File "/usr/local/lib/python3.5/dist-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'

Looks to be an old issue fixed last year so I don't understand which lib is in the wrong version…
https://github.com/fchollet/keras/issues/1385

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

Hi shazz,

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

Seun January 25, 2017 at 10:13 pm # REPLY

Hi Jasson,
Thanks so much. The second fix worked for me.

Jason Brownlee January 26, 2017 at 4:45 am # REPLY ↩

Glad to hear it Seun.

Sulthan January 31, 2017 at 3:08 am # REPLY ↩

Dear Jason,

With the help of your example i am trying to use the same for handwritten digits pixel data to classify the no input is 5000rows with example 20*20 pixels so totally x matrix is (5000,400) and Y is (5000,1), i am not able to successfully run the model getting error as below in the end of the code.

```
#importing the needed libraries
import scipy.io
import numpy
from sklearn.preprocessing import LabelEncoder
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
```

In [158]:

```
#Intializing random no for reproductiblity
seed = 7
numpy.random.seed(seed)
```

In [159]:

```
#loading the dataset from mat file
mat = scipy.io.loadmat(‘C:\\Users\\Sulthan\\Desktop\\NeuralNet\\ex3data1.mat’)
print(mat)
```

```
{‘X’: array([[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.],
…,
[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.]]), ‘__header__’: b’MATLAB 5.0 MAT-file, Platform: GLNXA64, Created on: Sun
Oct 16 13:09:09 2011′, ‘__version__’: ‘1.0’, ‘y’: array
[10],
[10],
…,
[ 9],
[ 9],
[ 9]], dtype=uint8), '__globals__': []}
```

Type Markdown and LaTeX:
α
2
α2
In [ ]:

In [ ]:

In [160]:

```
#Splitting of X and Y of DATA
X_train = mat['X']
```

In [161]:

```
X_train
```

Out[161]:

```
array([[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.],
…,
[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.]])
```

In [162]:

```
Y_train = mat['y']
```

In [163]:

```
Y_train
```

Out[163]:

```
array([[10],
[10],
[10],
…,
[ 9],
[ 9],
[ 9]], dtype=uint8)
```

```
In [164]:

X_train.shape

Out[164]:
(5000, 400)
In [165]:

Y_train.shape

Out[165]:
(5000, 1)
In [166]:

data_trainX = X_train[2500:,0:400]

In [167]:

data_trainX

Out[167]:
array([[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.],
…,
[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.],
[ 0., 0., 0., …, 0., 0., 0.]])
In [168]:

data_trainX.shape

Out[168]:
(2500, 400)
In [256]:

data_trainY = Y_train[:2500,:].reshape(-1)

In [257]:

data_trainY
data_trainY.shape

Out[257]:
(2500,)
In [284]:

#enocode class values as integers
encoder = LabelEncoder()
encoder.fit(data_trainY)
encoded_Y = encoder.transform(data_trainY)
# convert integers to dummy variables
dummy_Y= np_utils.to_categorical(encoded_Y)
```

In [285]:

```
dummy_Y
```

Out[285]:

```
array([[ 0., 0., 0., 0., 1.],
[ 0., 0., 0., 0., 1.],
[ 0., 0., 0., 0., 1.],
…,
[ 0., 0., 0., 1., 0.],
[ 0., 0., 0., 1., 0.],
[ 0., 0., 0., 1., 0.]])
```

In [298]:

```
newy = dummy_Y.reshape(-1,1)
```

In [300]:

```
newy
```

Out[300]:

```
array([[ 0.],
[ 0.],
[ 0.],
…,
[ 0.],
[ 1.],
[ 0.]])
```

In [293]:

```
#define baseline model
def baseline_model():
#create model
model = Sequential()
model.add(Dense(15,input_dim=400,init='normal',activation='relu'))
model.add(Dense(10,init='normal',activation='sigmoid'))
#compilemodel
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
return model

estimator = KerasClassifier(build_fn=baseline_model, nb_epoch=200,batch_size=5,verbose=0)
print(estimator)
```

In [295]:

```
kfold = KFold(n_splits=10, shuffle=True, random_st
results = cross_val_score(estimator, data_trainX, newy, cv=kfold)
print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
```

```
ValueError Traceback (most recent call last)
in ()
—-> 1 results = cross_val_score(estimator, data_trainX, newy, cv=kfold)
2 print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))

C:\Users\Sulthan\Anaconda3\lib\site-packages\sklearn\model_selection\_validation.py in
cross_val_score(estimator, X, y, groups, scoring, cv, n_jobs, verbose, fit_params, pre_dispatch)
126
127 """
–> 128 X, y, groups = indexable(X, y, groups)
129
130 cv = check_cv(cv, y, classifier=is_classifier(estimator))

C:\Users\Sulthan\Anaconda3\lib\site-packages\sklearn\utils\validation.py in indexable(*iterables)
204 else:
205 result.append(np.array(X))
–> 206 check_consistent_length(*result)
207 return result
208

C:\Users\Sulthan\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_consistent_length(*arrays)
179 if len(uniques) > 1:
180 raise ValueError("Found input variables with inconsistent numbers of"
–> 181 " samples: %r" % [int(l) for l in lengths])
182
183

ValueError: Found input variables with inconsistent numbers of samples: [2500, 12500]
```

Jason Brownlee February 1, 2017 at 10:26 am #

REPLY

Hi Sulthan, the trace is a little hard to read.

Sorry, I have no off the cuff ideas.

Perhaps try cutting your example back to the minimum to help isolate the fault?

Linmu February 3, 2017 at 2:13 am #

REPLY

Hi Jason,

Thanks for your tutorial!

Just one question regarding the output. In this problem, we got three classes (setosa, versicolor and virginica), and since each data instance should be classified into only one category, the problem is more specifically “single-lable, multi-class classification”. What if each data instance belonged to multiple categories. Then we are facing “multi-lable, multi-class classification”. In our case, each flower belongs to at least two species (Let’s just forget the biology 🙂).

My solution is to modify the output variable (Y) with mutiple ‘1’ in it, i.e. [1 1 0], [0 1 1], [1 1 1]……. This is definitely not one-hot encoding any more (maybe two or three-hot?)

Will my method work out? If not, how do you think the problem of “multi-lable, multi-class classification” should be solved?

Thanks in advance

Jason Brownlee February 3, 2017 at 10:07 am #

REPLY ↩

Your method sounds very reasonable.

You may also want to use sigmoid activation functions on the output layer to allow binary class membership to each available class.

solarenqu February 19, 2017 at 9:28 pm #

REPLY ↩

Hello, how can I use the model to create predictions?

if i try this: print(‘predict: ‘,estimator.predict([[5.7,4.4,1.5,0.4]])) i got this exception:

AttributeError: ‘KerasClassifier’ object has no attribute ‘model’
Exception ignored in: <bound method BaseSession.__del__ of >
Traceback (most recent call last):
File “/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-packages/tensorflow/python/client/session.py”, line 581, in __del__
AttributeError: ‘NoneType’ object has no attribute ‘TF_DeleteStatus’

Jason Brownlee February 20, 2017 at 9:29 am #

REPLY ↩

I have not seen this error before.

What versions of Keras/TF/sklearn/Python are you using?

REPLY ↩

Suvam March 1, 2017 at 7:34 am #

Hi,

Thanks for the great tutorial.
It would be great if you could outline what changes would be necessary if I want to do a multi-class classification with text data: the training data assigns scores to different lines of text, and the problem is to infer the score for a new line of text. It seems that the estimator above cannot handle strings. What would be the fix for this?

Thanks in advance for the help.

Jason Brownlee March 1, 2017 at 8:47 am #

REPLY ↰

Consider encoding your words as integers, using a word embedding and a fixed sequence length.

See this tutorial:
http://machinelearningmastery.com/predict-sentiment-movie-reviews-using-deep-learning/

Sweta March 1, 2017 at 9:10 pm #

REPLY ↰

This was a great tutorial to enhance the skills in deep learning. My question: is it possible to use this same dataset for LSTM? Can you please help with this how to solve in LSTM?

Jason Brownlee March 2, 2017 at 8:15 am #

REPLY ↰

Hi Sweta,

You could use an LSTM, but it would not be appropriate because LSTMs are intended for sequence prediction problems and this is not a sequence prediction problem.

Akash March 22, 2017 at 5:47 pm #

REPLY ↰

Hi Jason,

I have this problem where I have 1500 features as input to my DNN and 2 output classes, can you explain how do I decide the size of neurons in my hidden layer and how many hidden layers I need to process such high features with accuracy.

Jason Brownlee March 23, 2017 at 8:47 am #

REPLY

Lots of trial and error.

Start with a small network and keep adding neurons and layers and epochs until no more benefit is seen.

Ananya Mohapatra March 24, 2017 at 9:39 pm #

REPLY

sir, the following code is showing an error message.. could you help me figure it out. i am trying to do a multi class classification with 5 datasets combined in one(4 non epileptic patients and 1 epileptic) …500 x 25 dataset and the 26th column is the class.

```
# Train model and make predictions
import numpy
import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
from sklearn.model_selection import cross_val_score
from sklearn.cross_validation import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import KFold

# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
# load dataset
dataframe = pandas.read_csv("DemoNSO.csv", header=None)
dataset = dataframe.values
X = dataset[:,0:25].astype(float)
Y = dataset[:,25]
# encode class values as integers
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
# convert integers to dummy variables (i.e. one hot encoded)
dummy_y = np_utils.to_categorical(encoded_Y)
# define baseline model
def baseline_model():
# create model
model = Sequential()
model.add(Dense(700, input_dim=25, init='normal',
model.add(Dense(2, init=’normal’, activation=’sigmoid’))

# Compile model
model.compile(loss=’categorical_crossentropy’, optimizer=’adam’, metrics=[‘accuracy’])
return model

estimator = KerasClassifier(build_fn=baseline_model, nb_epoch=50, batch_size=20)

kfold = KFold(n_splits=5, shuffle=True, random_state=seed)

results = cross_val_score(estimator, X, dummy_y, cv=kfold)
print(“Baseline: %.2f%% (%.2f%%)” % (results.mean()*100, results.std()*100))

X_train, X_test, Y_train, Y_test = train_test_split(X, dummy_y, test_size=0.55, random_state=seed)
estimator.fit(X_train, Y_train)
predictions = estimator.predict(X_test)

print(predictions)
print(encoder.inverse_transform(predictions))
```

error message:

```
str(array.shape))
ValueError: Error when checking model target: expected dense_56 to have shape (None, 2) but got array with shape (240, 3)
```

Jason Brownlee March 25, 2017 at 7:36 am #

REPLY

Confirm the size of your output (y) matches the dimension of your output layer.

Alican March 28, 2017 at 4:05 am #

REPLY

Hello Jason,

I got your model to work using Python 2.7.13, Keras 2.0.2, Theano 0.9.0.dev…, by copying the codes exactly, however the results that I get are not only very bad (59.33%, 48.67%, 38.00% on different trials), but they are also different.

I was under the impression that using a fixed seed would allow us to reproduce the same results.

Do you have any idea what could have caused such bad results?

Thanks

Alican March 28, 2017 at 4:28 am #

REPLY

edit: I was re-executing only the result listed above.

Running the whole script over and over generates the same result: “Baseline: 59.33% (21.59%)”

Jason Brownlee March 28, 2017 at 8:26 am #

REPLY

Glad to hear it.

Jason Brownlee March 28, 2017 at 8:25 am #

REPLY

Not sure why the results are so bad. I’ll take a look.

The fixed seed does not seem to have an effect on the Theano or TensorFlow backends. Try running examples multiple times and take the average performance.

Alican April 2, 2017 at 2:30 am #

REPLY

Did you have time to look into this?

I had my colleague run this script on Theano 1.0.1, and it gave the expected performance of 95.33%. I then installed Theano 1.0.1, and got the same result again.

However, using Theano 2.0.2 I was getting 59.33% with seed=7, and similar performances with different seeds. Is it possible the developers made some crucial changes with the new version?

Jason Brownlee April 2, 2017 at 6:30 am #

REPLY

The most recent version of Theano is 0.9: https://github.com/Theano/Theano/releases

Do you mean Keras versions?

It may not be the Keras version causing the difference in the run. The fixed random seed may not be having an effect in general, or may not be having when a Theano backend is being used.

Neural networks are stochastic algorithms and will produce a different result each run: http://machinelearningmastery.com/randomness-in-machine-learning/

Alican April 2, 2017 at 6:59 an

Yes I meant Keras, sorry.

There is no issue with the seed, I'm getting the same result with you on multiple computers using Keras 1.1.1. But with Keras 2.0.2, the results are absymally bad.

Jonathan July 11, 2017 at 4:28 am #

not sure if this was every resolved, but I'm getting the same thing with most recent versions of Theano and Keras

59.33% with seed=7

Jason Brownlee July 11, 2017 at 10:33 am #

Try running the example a few times with different seeds.

Neural networks are stochastic:
http://machinelearningmastery.com/randomness-in-machine-learning/

Nalini March 29, 2017 at 3:13 am #

REPLY ↩

Hi Jason

in this code for multiclass classification can u suggest me how to plot graph to display the accuracy and also what should be the axis represent

Jason Brownlee March 29, 2017 at 9:10 am #

REPLY ↩

No, we normally do not graph accuracy, unless you want to graph it over training epochs?

Nalini March 31, 2017 at 1:42 am #

REPLY ↩

thanks

Jason Brownlee March 31, 2017 at 5:55 am # REPLY

You're welcome.

Frank April 6, 2017 at 8:47 pm # REPLY

Dear Jason,
I have found this tutorial very interesting and helpful.
What I wanted to ask is, I am currently trying to classify poker hands as this kaggle competition: https://www.kaggle.com/c/poker-rule-induction (For a school project) I wish to create a neural network as you have created above. What do you suggest for me to start this?
Your help would be greatly appreciated!
Thanks.

Jason Brownlee April 9, 2017 at 2:39 pm # REPLY

This process will help you work through your modeling problem:
http://machinelearningmastery.com/start-here/#process

shiva April 8, 2017 at 12:28 pm # REPLY

Hi Jason,
Its an awesome tutorial. It would be great if you can come up with a blog post on multiclass medical image classification with Keras Deep Learning library. It would serve as a great asset for researchers like me, working with medical image classification. Looking forward.

Jason Brownlee April 9, 2017 at 2:56 pm # REPLY

Thanks for the suggestion.

Toby April 9, 2017 at 4:38 am # REPLY

Thanks for the great tutorial!
I duplicated the result using Theano as backend.
However, using Tensorflow yield a worse accuracy, 88.67%.
Any explanation?

Thanks!

Jason Brownlee April 9, 2017 at 3:00 pm #

REPLY ↰

It may be related to the stochastic nature of neural nets and the difficulty of making results with the TF backend reproducible.

You can learn more about the stochastic nature of machine learning algorithms here:
http://machinelearningmastery.com/randomness-in-machine-learning/

Anupam April 11, 2017 at 6:11 pm #

REPLY ↰

Hi Jason, How to find the Precision, Recall and f1 score of your example?

Case-1 I have used like :

model.compile(loss='categorical_crossentropy', optimizer='Nadam', metrics=['acc', 'fmeasure', 'precision', 'recall'])

Case-2 and also used :

```
def score(yh, pr):
coords = [np.where(yhh > 0)[0][0] for yhh in yh]
yh = [yhh[co:] for yhh, co in zip(yh, coords)]
ypr = [prr[co:] for prr, co in zip(pr, coords)]
fyh = [c for row in yh for c in row]
fpr = [c for row in ypr for c in row]
return fyh, fpr
```

```
pr = model.predict_classes(X_train)
yh = y_train.argmax(2)
fyh, fpr = score(yh, pr)
print 'Training accuracy:', accuracy_score(fyh, fpr)
print 'Training confusion matrix:'
print confusion_matrix(fyh, fpr)
precision_recall_fscore_support(fyh, fpr)
```

```
pr = model.predict_classes(X_test)
yh = y_test.argmax(2)
fyh, fpr = score(yh, pr)
print 'Testing accuracy:', accuracy_score(fyh, fpr)
print 'Testing confusion matrix:'
print confusion_matrix(fyh, fpr)
precision_recall_fscore_support(fyh, fpr)
```

What I have observed is that, accuracy of case-1 ar

Any solution?

Jason Brownlee April 12, 2017 at 7:52 am # REPLY ↩

You can make predictions on your test data and use the tools from sklearn:
http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Raynier van Egmond April 15, 2017 at 12:19 pm # REPLY ↩

Hi Jason,

Like a student earlier in the comments my accuracy results are exactly the same as his:

********** Baseline: 88.67% (21.09%)

and I think this is related to having Tensorflow as the backend rather than the Theano backend.

I am working this through in a Jupyter notebook

I went through your earlier tutorials on setting up the environment:

```
scipy: 0.18.1
numpy: 1.11.3
matplotlib: 2.0.0
pandas: 0.19.2
statsmodels: 0.6.1
sklearn: 0.18.1
theano: 0.9.0.dev-c697eeab84e5b8a74908da654b66ec9eca4f1291
tensorflow: 1.0.1
Using TensorFlow backend.
keras: 2.0.3
```

The Tensorflow is a Python3.6 recompile picked up from the web at:

http://www.lfd.uci.edu/~gohlke/pythonlibs/#tensorflow

Do you know have I can force the Keras library to take Theano as a backend rather than the Tensorflow library?

Thanks for the great work on your tutorials… for beginners it is such in invaluable thing to have tutorials that actually work !!!

Looking forward to get more of your books

Rene

Raynier van Egmond April 15, 2017 at 12:42 pm #

REPLY

Changing to the Theano backend doesn't change the results:

Managed to change to a Theano backend by setting the Keras config file:

```
{
“image_data_format”: “channels_last”,
“epsilon”: 1e-07,
“floatx”: “float32”,
“backend”: “theano”
}
```

as instructed at: https://keras.io/backend/#keras-backends

The notebook no longer reports it is using Tensorflow so I guess the switch worked but the results are still:

****** Baseline: 88.67% (21.09%)

Will need to look a little deeper and play with the actual architecture a bit.

All the same great material to get started with

Thanks again

Rene

Raynier van Egmond April 15, 2017 at 1:26 pm #

REPLY

Confirmed that changes to the model as someone above mentioned

```
model.add(Dense(8, input_dim=4, kernel_initializer=’normal’, activation=’relu’))
model.add(Dense(3, kernel_initializer=’normal’, activation=’softmax’))
```

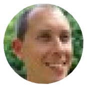
nodes makes a substantial difference:

**** Baseline: 96.67% (4.47%)

but there is no difference between the Tensorflow and Theano backend results. I guess that's as far as I can take this for now.

Take care,

Rene



Jason Brownlee April 16, 2017 at 9:27 am #

REPLY

Nice.

Also, note that MLPs are stochastic. This means that if you don't fix the random seed, you will get different results for each run of the algorithm.

Ideally, you should take the average performance of the algorithm across multiple runs to evaluate its performance.

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

Jason Brownlee April 16, 2017 at 9:22 am #

REPLY ↩

You can change the back-end used by Keras in the Kersas config file. See this post:
http://machinelearningmastery.com/introduction-python-deep-learning-library-keras/

Tursun April 16, 2017 at 9:18 pm #

REPLY ↩

Jason,
Thank you very much first. These tutorials are excellent. They are very practical. Your are an excellent educator.
I want classify my data into multiple classes of 25-30. Your IRIS example is nearest classification. They DL4J previously has IRIS classification with DBN; but disappeared in new community version.
I have following issues:
1.>
It takes so long. My laptop is TOSHIBA L745, 4GB RAM, i3 processor. it has CUDA.
My classification problem is solved with SVM in very short time. I'd say in split second.
Do you think speed would increase if we use DBN or CNN something ?
2.>
My result :
Baseline: 88.67% (21.09%),
Once I have installed Docker (tensorflow in it),then run IRIS classification. It shows 96%.
I wish similar or better accuracy. How to reach that level ?

Thank you

Jason Brownlee April 17, 2017 at 5:13 am #

REPLY ↩

MLP is the right algorithm for multi-class classification algorithms.

If it is slow, consider running it on AWS:
http://machinelearningmastery.com/develop-evaluate-large-deep-learning-models-keras-amazon-web-services/

There are many things you can do to lift perforr http://machinelearningmastery.com/improve-deep-learning-performance/

Chris April 17, 2017 at 5:13 am # REPLY

Hello Jason,
first of all, your tutorials are really well done when you start working with keras.

I have a question about the epochs and batch_size in this tutorial. I think I haven't understood it correctly.

I loaded the record and it contains 150 entries.

You choose 200 epochs and batch_size=5. So you use 5*200=1000 examples for training. So does keras use the same entries multiple times or does it stop automatically?

Thanks!

Jason Brownlee April 18, 2017 at 8:23 am # REPLY

One epoch involves exposing each pattern in the training dataset to the model.

One epoch is comprised of one or more batches.

One batch involves showing a subset of the patterns in the training data to the model and updating weights.

The number of patterns in the dataset for one epoch must be a factor of the batch size (e.g. divide evenly).

Does that help?

Chris April 22, 2017 at 3:43 am # REPLY

Hi,
thank you for the explanation.
The explanation helped me, and in the meantime I have read and tried several LSTM tutorials from you and it became much clearer to me.
greetings, Chris

Jason Brownlee April 22, 2017 at 9:28 am # REPLY

I'm glad to hear that Chris.

Abhilash Menon April 17, 2017 at 1:27 pm #

REPLY ↰

Hey Jason,

I have been following your tutorials and they have been very very helpful!. Especially, the most useful section being the comments where people like me get to ask you questions and some of them are the same ones I had in my mind.

Although, I have one that I think hasn’t been asked before, at least on this page!

What changes should I make to the regular program you illustrated with the “pima_indians_diabetes.csv” in order to take a dataset that has 5 categorical inputs and 1 binary output.

This would be a huge help! Thanks in advance!

Jason Brownlee April 18, 2017 at 8:30 am #

REPLY ↰

Great question.

Consider using an integer encoding followed by a binary encoding of the categorical inputs.

This post will show you how:
http://machinelearningmastery.com/data-preparation-gradient-boosting-xgboost-python/

Abhilash Menon July 18, 2017 at 12:47 pm #

REPLY ↰

Hello Dr. Brownlee,

The link that you shared was very helpful and I have been able to one hot encode and use the data set but at this point of time I am not able to find relevant information regarding what the perfect batch size and no. of epochs should be. My data has 5 categorical inputs and 1 binary output (2800 instances). Could you tell me what factors I should take into consideration before arriving at a perfect batch size and epoch number? The following are the configuration details of my neural net:

```
model.add(Dense(28, input_dim=43, init=’uniform’, activation=’relu’))
model.add(Dense(28, init=’uniform’, activation=’relu’))
model.add(Dense(1, init=’uniform’, activation=’sigmoid’))
model.compile(loss=’binary_crossentropy’, optimizer=’adam’, metrics=[‘accuracy’])
```

Jason Brownlee July 18, 2017 at 5:01 pm #

REPLY ↰

I recommend testing a suite of

I have a post this friday with advice on tuning the batch size, watch out for it.

Tuba April 18, 2017 at 8:43 am #

REPLY

Hi Jason,

First of all, your tutorials are really very interesting.

I was facing error this when i run it . I'm work with python 3 and the same file input .

```
Error :
ImportError: Traceback (most recent call last):
File "/home/indatacore/anaconda3/lib/python3.5/site-packages/tensorflow/python/__init__.py", line 61, in
from tensorflow.python import pywrap_tensorflow
File "/home/indatacore/anaconda3/lib/python3.5/site-packages/tensorflow/python/pywrap_tensorflow.py", line 28, in
_pywrap_tensorflow = swig_import_helper()
File "/home/indatacore/anaconda3/lib/python3.5/site-packages/tensorflow/python/pywrap_tensorflow.py", line 24, in swig_import_helper
_mod = imp.load_module('_pywrap_tensorflow', fp, pathname, description)
File "/home/indatacore/anaconda3/lib/python3.5/imp.py", line 242, in load_module
return load_dynamic(name, filename, file)
File "/home/indatacore/anaconda3/lib/python3.5/imp.py", line 342, in load_dynamic
return _load(spec)
ImportError: libcudart.so.8.0: cannot open shared object file: No such file or directory

Failed to load the native TensorFlow runtime.

See https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/get_started/os_setup.md#import_error

for some common reasons and solutions. Include the entire stack trace
above this error message when asking for help.
```

Jason Brownlee April 19, 2017 at 7:44 am #

REPLY

Ouch. I have not seen this error before.

Consider trying the Theano backend and see if that makes a difference.

Tursun April 21, 2017 at 2:17 am #

REPLY

Jason,

Thank you. I got your notion: there is no key which

Here, I have multi class classification problem.
My data can be downloaded from here:
https://www.dropbox.com/s/w2en6ewdsed69pc/tursun_deep_p6.csv?dl=0

size of my data set : 512*16, last column is 21 classes, they are digits 1-21
note: number of samples (rows in my data) for each class is different. mostly 20 rows, but sometimes 17 or 31 rows
my network has:
first layer (input) has 15 neurons
second layer (hidden) has 30 neurons
last layer (output) has 21 neurons
in last layer I used “softmax” based on this recommendation from
https://github.com/fchollet/keras/issues/1013
“The softmax function transforms your hidden units into probability scores of the class labels you have; and thus is more suited to classification problems ”
error message:
alueError: Error when checking model target: expected dense_8 to have shape (None, 21) but got array with shape (512, 1)

I would be thankful if you can help me to run this code.

I modified this code from yours:

```
———–keras code start ———–
from keras.models import Sequential
from keras.layers import Dense
import numpy
# fix random seed for reproducibility
numpy.random.seed(7)
# load pima indians dataset
dataset = numpy.loadtxt(“tursun_deep_p6.csv”, delimiter=”,”)
# split into input (X) and output (Y) variables
X = dataset[:,0:15]
Y = dataset[:,15]

# create model
model = Sequential()
model.add(Dense(30, input_dim=15, activation=’relu’)) # not sure if 30 too much. not sure #about lower and upper limits
#model.add(Dense(25, activation=’relu’)) # think about to add one more hidden layer
model.add(Dense(21, activation=’softmax’)) # they say softmax at last L does classification
# Compile model
model.compile(loss=’categorical_crossentropy’, optimizer=’adam’, metrics=[‘accuracy’])
# Fit the model
model.fit(X, Y, epochs=150, batch_size=5)
# evaluate the model
scores = model.evaluate(X, Y)
print(“\n%s: %.2f%%” % (model.metrics_names[1],
```

———keras code start ———

Jason Brownlee April 21, 2017 at 8:40 am #

REPLY

I see the problem, your output layer expects 8 columns and you only have 1.

You need to transform your output variable int 8 variables. You can do this using a one hot encoding.

Shiva April 23, 2017 at 5:54 am #

REPLY

Hi jason, I am following your book deep learning with python and i have an issue with the script. I have succesfully read my .csv datafile through pandas and trying to adopt a decay based learning rate as discussed in the book. I define the initial lrate, drop, epochs_drop and the formula for lrate update as said in the book. I then created the model like this (works best for my problem) and started creating a pipeline in contrary to the model fitting strategy used by you in the book:

```
def baseline_model():
# create model
model = Sequential()
model.add(Dense(50, input_dim=15, kernel_initializer=’normal’, activation=’relu’))
model.add(Dense(3, kernel_initializer=’normal’, activation=’sigmoid’))
sgd = SGD(lr=0.0, momentum=0.9, decay=0, nesterov=False)
model.compile(loss=’categorical_crossentropy’, optimizer=sgd, metrics=[‘accuracy’])
return model
#learning schedule callback
lrate = LearningRateScheduler(step_decay)
callbacks_list = [lrate]

estimators = []
estimators.append((‘standardize’, StandardScaler()))
estimators.append((‘mlp’, KerasClassifier(build_fn=baseline_model, epochs=100,
batch_size=5, callbacks=[lrate], verbose=1)))
pipeline = Pipeline(estimators)
kfold = StratifiedKFold(n_splits=2, shuffle=True, random_state=seed)
results = cross_val_score(pipeline, X, encoded_Y, cv=kfold)
```

I’m getting the error “Cannot clone object , as the constructor does not seem to set parameter callbacks”. According to keras documentation, I can see that i can pass callbacks to the kerasclassifier wrapper. kindly suggest what to do in this occasion. Looking forward.

Jason Brownlee April 24, 2017 at 5:29 am #

REPLY

I have not tried to use callbacks with the sklearn wrapper sorry.

Perhaps it is a limitation that you can't? Though, I'd be surprised.

you may have to use the keras API directly.

Shiva April 25, 2017 at 6:23 am #

REPLY

Hi Jason,
I'm trying to apply the image augmentation techniques discussed in your book to the data I have stored in my system under C:\images\train and C:\images\test. Could you help me with the syntax on how to load my own data with a modification to the syntax available in the book:

```
# load data
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

Thanks in advance.

Jason Brownlee April 25, 2017 at 7:52 am #

REPLY

Sorry, I don't have an example of how to load image data from disk, I hope to cover it in the future.

This post may help as a start:
https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

Michael Ng April 28, 2017 at 12:49 am #

REPLY

Hi,

By implementing neural network in Keras, how can we get the associated probabilities for each predicted class?'

Many Thanks!
Michael Ng

Jason Brownlee April 28, 2017 at 7:47 am #

REPLY

Review the outputs from the softmax, although not strictly probabilities, they can be used as such.

Also see the keras function model.predict_proba() for predicting probabilities directly.
https://keras.io/models/sequential/

Michael Ng April 30, 2017 at 11:55 am #

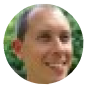
REPLY

Hi Jason,

'Note that we use a sigmoid activation function in the output layer. This is to ensure the output values are in the range of 0 and 1 and may be used as predicted probabilities.'

Instead of using softmax function, how do I review the sigmoidal outputs (as per the tutorial) for each of 3 output nodes? Mind to share the code to list the sigmoidal outputs?

Regards,
Michael Ng



Jason Brownlee May 1, 2017 at 5:52 am #

REPLY

I would recommend softmax for multi-class classification.

You can learn more about sigmoid here:
https://en.wikipedia.org/wiki/Logistic_function

Ann April 28, 2017 at 2:08 am #

REPLY

Hi, Jason! I'm exactly newbie to Keras, and I want to figure out confusion matrix by using sklearn.confusion_matrix(y_test, predict). But I was facing error this when i run it .

```
ValueError Traceback (most recent call last)
in ()
—-> 1 confusion_matrix(y_test, predict)

C:\Users\Ann\Anaconda3\envs\py27\lib\site-packages\sklearn\metrics\classification.pyc in
confusion_matrix(y_true, y_pred, labels, sample_weight)
240 y_type, y_true, y_pred = _check_targets(y_true, y_pred)
241 if y_type not in ("binary", "multiclass"):
–> 242 raise ValueError("%s is not supported" % y_
```

243
244 if labels is None:

ValueError: multilabel-indicator is not supported

I've checked that y_test and predict have same shape (231L, 2L).
Any solution?
Your help would be greatly appreciated!
Thanks.

Jason Brownlee April 28, 2017 at 7:50 am #

REPLY

Consider checking the dimensionality of both y and yhat to ensure they are the same (e.g. print the shape of them).

Mohammed Zahran April 30, 2017 at 4:49 am #

REPLY

can we use the same approach to classify MNIST in (0,1…) and the same time classify the numbers to even and odd numbers ?

Jason Brownlee April 30, 2017 at 5:35 am #

REPLY

Machine learning is not needed to check for odd and even numbers, just a little math.

TAM.G April 30, 2017 at 4:46 pm #

REPLY

but if we too it as a simple try to learn about multi-labeling ,, how could we do this

Moh May 1, 2017 at 10:45 am #

REPLY

@Jason Brownlee I totally agree with you. We are using this problem as proxy for more complex problems like classifying a scene with multiple cars and we want to classify the models of these cars. The same approach is needed in tackling neurological images

TAM.G April 30, 2017 at 3:22 pm #

REPLY

first this is a great tutorial , but , am confused a little ,, am i loading my training files and labeling files or what ??
as i tried to apply this tutorial to my case ,, I've about 10 folder each has its own images these images are related together for one class ,, but i need to make multi labeling for each folder of them for example folder number 1 has about 1500 .png imgs of owl bird , here i need to make a multi label for this to train it as a bird and owl , and here comes the problem ,, as i'm seraching for a tool to make labeling for all images in each folder and label them as [owl, bird] together … any idea about how to build my own multi label classifier ?

Jason Brownlee May 1, 2017 at 5:53 am # REPLY ↰

I would recommend using a CNN instead of an MLP for image classification, see this post:
http://machinelearningmastery.com/object-recognition-convolutional-neural-networks-keras-deep-learning-library/

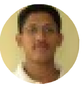
Ik.O May 14, 2017 at 10:58 pm # REPLY ↰

I implemented the same code on my system and achieved a score of 88.67% at seed = 7 and 96.00% at seed = 4. Any particular reason for this?

Jason Brownlee May 15, 2017 at 5:52 am # REPLY ↰

Nice work!

Yes, deep learning algorithms are stochastic:
http://machinelearningmastery.com/randomness-in-machine-learning/



Anupam May 18, 2017 at 4:58 pm # REPLY ↰

Hi Jason, Just gone through your blog http://machinelearningmastery.com/ .Just to know as a beginner in Deep learning, can you give any hint to do the task sequence learning for word language identification problem.
Here each word is a variable sequence of characters and the id of each word must be classified with a language tag.
Like, Suppose if we have a dataset like:

hello/L1 bahiya/L2 hain/L2 brother/L1 ,/L3 :)/L4

where L1,L2,L3 and L4 are the Language-tag

Jason Brownlee May 19, 2017 at 8:14 am # REPLY

Hi Anupam, that sounds like a great problem.

I would suggest starting with a high-quality dataset, then consider modeling the problem using a seq2seq architecture.

A.Malathi May 19, 2017 at 7:30 pm # REPLY

Hi Jason,

Your tutorials are great and very helpful to me. Have you written any article on Autoencoder.
I have constructed an autoencoder network for a dataset with labels. The output is a vector of errors(Euclidean Distance). From that errors, classification or prediction on the test set is possible since labels are given??

Jason Brownlee May 20, 2017 at 5:37 am # REPLY

Sorry, I don't currently have any material on autoencoders.

J. A. Gildea May 22, 2017 at 2:57 am # REPLY

Hi Jason, thank you so much for your helpful tutorials.
I have one question regarding one-hot encoding:
I am working on using a CNN for sentiment analysis and I have a total of six labels for my output variable, string values (P+, P, NONE, NEU, N, N+) representing sentiments.
I one-hot encoded my output variable the same way as you showed in this tutorial, but the shape after one-hot encoding appears to be (, 7). Shouln't it be 6 instead of 7? Any idea what might be going on? I checked for issues in my dataset such as null values in a certain row, and got rid of all of them yet this persists.
Thanks!

Jason Brownlee May 22, 2017 at 7:54 am # REPLY

It should be 7.

Consider loading your data in Python and printing the set of values in the column to get an idea of what is in your data.

J. A. Gildea May 22, 2017 at 6:39 pm # REPLY ↰

I checked my data a bit deeper and it seems it had a couple of null values that I removed.
I am however getting very poor results, could this be due to the fact that my data is a bit unbalanced? Some of the classes appear twice as others, so I imagine I would have to change the metrics in my compile function (using accuracy at the moment).
Can a slight imbalance in the dataset yield such poor results (under 40% validation accuracy)?

Thanks.

Jason Brownlee May 23, 2017 at 7:50 am # REPLY ↰

With multiple classes, it might be better to use another metric like log loss (cross entropy) or AUC.

Accuracy will not capture the true performance of the model.

Also, imbalanced classes can be a problem. You could look at removing some classes or rebalancing the data:
http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

Nalini May 24, 2017 at 6:10 pm # REPLY ↰

Hi Jason!
I can't seem to add more layers in my code.
model.add(Dense(12, input_dim=25, init='normal', activation='relu'))
model.add(Dense(5, init='normal', activation='sigmoid'))
This is a part of the existing code. if i try to add more layers along with them i get a warning for indentation fault.
can you please specify which one of the above layers is the input layer and which one is hidden….

Jason Brownlee June 2, 2017 at 11:32 am # REPLY ↰

This is a Python issue. Ensure you understand the role of whitespace in Python:
http://www.diveintopython.net/getting_to_know_python/indenting_code.html

Michael May 28, 2017 at 4:01 am # REPLY ↰

Hi Jason,
I have two questions:

1. I didn't see the code in this post calling the fit method. Is the fitting process executed in KerasClassifier?

2. I have only one dataset as training set (No dedicated test set).
Is the KFold method using this single dataset for evaluation in the KerasClassifier class?
Or should I use the "validation_split parameter in the fit method?

Thank's

Jason Brownlee June 2, 2017 at 12:06 pm # REPLY ↰

Hi Michael,

Yes, we use the sklearn infrastructure to fit and evaluate the model.

You can try both methods. The best evaluation test harness is really problem dependent. k-fold cross validation generally gives a less biased estimate of performance and is often recommended.

Nimesh May 29, 2017 at 4:20 pm # REPLY ↰

I am classifying mp3s into 7 genre classes. I have 1200 mp3 files dataset with 7 features as input. I got basic Neural network as your example shows and it gives nearly 60% of accuracy. Any suggestions on how to improve accuracy? your suggestions will be very helpful for me.

Jason Brownlee June 2, 2017 at 12:22 pm # REPLY ↰

Yes, see this post:
http://machinelearningmastery.com/machine-learning-performance-improvement-cheat-sheet/

And this post:
http://machinelearningmastery.com/improve-deep-learning-performance/

J. A. Gildea June 9, 2017 at 3:35 am # REPLY ↰

Hello Jason,

I posted here a while back and I'm back for more wisdom!

I have my own model and dataset for text classifica
not sure on how to evaluate it, I have tried using k f

accuracy which I assume is not the reality.
Just using model.fit() I obtain a result of 99%, which also makes me think I am not evaluating my model correctly.
I have been looking for a way to do this and apparently a good approach is to use a confusion matrix. Is this necessary to evaluate a multiclass model for text classification, or will other methods suffice?

Thanks

Jason Brownlee June 9, 2017 at 6:30 am #

REPLY ↩

Generally, I would recommend this process to work through your problem systematically:
http://machinelearningmastery.com/start-here/#process

I would recommend this post to get a robust estimate of the skill of a deep learning model on unseen data:
http://machinelearningmastery.com/evaluate-skill-deep-learning-models/

For multi-class classification, I would recommend a confusion matrix, but also measures like logloss.

zakaria June 11, 2017 at 3:47 am #

REPLY ↩

Hi Jason, I need your help I use tensorflow and keras to classify cifar10 images. My question is how to make prediction (make prediction for only one image)

Jason Brownlee June 11, 2017 at 8:26 am #

REPLY ↩

Like this:

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

zakaria June 12, 2017 at 6:35 pm #

REPLY ↩

Hi Jason,
To make the prediction I used this function Y_pred = model.predict (x_test)
print (Y_pred)
Y_pred = np.argmax (Y_pred, axis = 1)
print (y_pred)

And I got these results
[[0, 0, …, 0, 0, 0]]
[0, 1, 0, …, 0, 0, 0]

```
[1. 0. 0. …, 0. 0. 0.]
…
[0, 0, 0, …, 0, 0, 0]]
[1. 0. 0. …, 0. 0. 0.]
[0. 0. 0. …, 1. 0. 0.]]
[0 1 0 …, 5 0 7]
```

What these results mean
And how to display for example the first 10 images of the test database to see if the model works well

Jason Brownlee June 13, 2017 at 8:18 am # REPLY ↰

The prediction result may be an outcome (probability-like value) for each class.

You can take an argmax() of each vector to find the selected class.

Alternately, you can call predict_classes() to predict the class directly.

Huong June 12, 2017 at 11:55 pm # REPLY ↰

Dear @Jason,
Thank you for your useful post. I have a issues.
My dataset have 3 columns (features) for output data. Each column has multi-classes. So how can I process in this case?
Thanks.

Jason Brownlee June 13, 2017 at 8:22 am # REPLY ↰

I don't have a great answer for you off the cuff. I would suggest doing a little research to see how this type of problem has been handled in the literature.

Maybe you can model each class separately?

Maybe you can one-hot encode each output variable and use a neural network to output everyone directly.

Let me know how you go.

Anastasios June 17, 2017 at 10:05 pm # REPLY ↰

Hello Jason,

great post on multiclass classification. I am trying tc t

I get an error when calling the fit function on the gridsearch. Can we apply gridsearch on a multiclass dataset ?

My code looks like: https://pastebin.com/eB35aJmW

And the error I get is: https://pastebin.com/C1ch7709

Jason Brownlee June 18, 2017 at 6:31 am #

REPLY

Yes, I believe you can grid search a multi-class classification problem.

Sorry, it is not clear to me what the cause of the error might be. You will need to cut your example back to a minimum case that still produces the error.

Anupam Samanta June 29, 2017 at 3:42 am #

REPLY

Hi Jason,

Excellent tutorials! I have been able to learn a lot reading your articles.

I ran into some problem while implementing this program
My accuracy was around Accuracy: 70.67% (12.00%)
I dont know why the accuracy is so dismal!
I tried changing some parameters, mostly that are mentioned in the comments, such as removing kernel_initializer, changing activation function, also the number of hidden nodes. But the best I was able to achieve was 70 %

Any reason something is going wrong here in my code?!

```
# Modules
import numpy
import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import np_utils
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder
from keras import backend as K
import os

def set_keras_backend(backend):
if K.backend() != backend:
os.environ['KERAS_BACKEND'] = backend
reload(K)
```

```
assert K.backend() == backend

set_keras_backend(“theano”)
# seed
seed = 7
numpy.random.seed(seed)

# load dataset
dataFrame = pandas.read_csv(“iris.csv”, header=None)
dataset = dataFrame.values

X = dataset[:, 0:4].astype(float)
Y = dataset[:, 4]

# encode class values
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)

dummy_Y = np_utils.to_categorical(encoded_Y)

# baseline model
def baseline_model():
# create model
model = Sequential()
model.add(Dense(8, input_dim=4, kernel_initializer=’normal’, activation=’softplus’))
model.add(Dense(3, kernel_initializer=’normal’, activation=’softmax’))
# compile model
model.compile(loss=’categorical_crossentropy’, optimizer=’adam’, metrics=[‘accuracy’])
return model

estimator = KerasClassifier(build_fn=baseline_model, nb_epoch=200, batch_size=5, verbose=0)
kfold = KFold(n_splits=10, shuffle=True, random_state=seed)

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

print(“Accuracy: %.2f%% (%.2f%%)” % (results.mean() * 100, results.std() * 100))
```

Anupam Samanta June 29, 2017 at 3:45 am #

REPLY

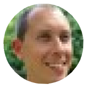
I added my code here: https://pastebin.com/3Kr7P6Kw
Its better formatted here!

Jason Brownlee June 29, 2017 at 6:39 am # REPLY ↰

There are more ideas here:
http://machinelearningmastery.com/deploy-machine-learning-model-to-production/

Anupam Samanta June 30, 2017 at 3:36 am # REPLY ↰

But isnt it strange, that when I use the same code as yours, my program in my machine returns such bad results!
Is there anything I am doing wrong in my code?!



Jason Brownlee June 30, 2017 at 8:14 am # REPLY ↰

No. Try running the example a few times. Neural networks are stochastic and give different results each time they are run.

See this post on why:
http://MachineLearningMastery.com/randomness-in-machine-learning/

See this post on how to address it and get a robust estimate of model performance:
http://MachineLearningMastery.com/evaluate-skill-deep-learning-models/

Zefeng Wu June 30, 2017 at 11:05 pm # REPLY ↰

Hi, my codes is as followings, but keras gave a extremebad results,

```
import numpy
import pandas
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
# load dataset
dataframe = pandas.read_csv(“iris.csv”, heade
dataset = dataframe.values
X = dataset[:,0:4].astype(float)
Y = dataset[:,4]
# encode class values as integers
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
# convert integers to dummy variables (i.e. one hot encoded)
dummy_y = np_utils.to_categorical(encoded_Y)
# define baseline model
def baseline_model():
# create model
model = Sequential()
model.add(Dense(8, input_dim=4 , activation= "relu" ))
model.add(Dense(3, activation= "softmax" ))
# Compile model
model.compile(loss= "categorical_crossentropy" , optimizer= "adam" , metrics=["accuracy"])
return model
estimator = KerasClassifier(build_fn=baseline_model, nb_epoch=200, batch_size=5, verbose=0)
kfold = KFold(n_splits=10, shuffle=True, random_state=seed)
results = cross_val_score(estimator, X, dummy_y, cv=kfold)
print("Accuracy: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
```

```
Using Theano backend.
Accuracy: 64.67% (15.22%)
```

Nunu July 4, 2017 at 12:13 am #

REPLY

Dear Jason,
How can I increase the accuracy while training ? I am always getting an accuracy arround 68% and 70%!! even if i am chanching the optimizer, the loss function and the learning rate.
(I am using keras and CNN)

Jason Brownlee July 6, 2017 at 10:02 am #

REPLY

Here are many ideas:
http://machinelearningmastery.com/machine-learning-performance-improvement-cheat-sheet/

Nunu July 8, 2017 at 12:06 am # REPLY

Thanks a lot it is very useful 🙂

Jason Brownlee July 9, 2017 at 10:47 am # REPLY

Glad to hear it.

Nunu July 12, 2017 at 7:27 pm # REPLY

Dear Jason,
I have a question: my model should classify every image in one of the 4 classes that I have, should I use "categorical cross entropy" or I can use instead the "Binary cross entropy" ? because I read a lot that when there is n classes it is better to use categorical cross entropy, but also the binary one is used for the same cases. I am too much confused 🙁 can you help me in understanding this issue better!!
Thanks in advance,
Nunu

Jason Brownlee July 13, 2017 at 9:53 am #

When you have more than 2 classes, use categorical cross entropy.

Nunu July 19, 2017 at 12:47 am #

oh ok thanks a lot 🙂 I have another question : I used Rmsprop with different learning rates such that 0.0001, 0.001 and 0.01 and with softmax in the last dense layer everything was good so far. Then i changed from softmax to sigmoid and i tried to excuted the same program with the same learning rates used in the cas of softmax, and here i got the problem : using learning rate 0.001 i got loss and val loss NAN after 24 epochs !! In your opinion what is the reason of getting such values??
Thanks in advance,
have a nice day,
Nunu

Jason Brownlee July 19,

Ensure you have scaled your input/output data to the bounds of the input/output activation functions.

Nunu July 19, 2017 at 5:49 pm #

Thanksssss 🙂

Sriram July 5, 2017 at 5:12 pm #

REPLY ↩

HI Jason,

Thanks for the awesome tutorial. I have a question regarding your first hidden layer which has 8 neurons. Correct me if I'm wrong, but shouldn't the number of neurons in a hidden layer be upperbounded by the number of inputs? (in this case 4).

Thanks,
Sriram

Jason Brownlee July 6, 2017 at 10:24 am #

REPLY ↩

No. There are no rules for the number of neurons in the hidden layer. Try different configurations and go with whatever robustly gives the best results on your problem.

Nunu July 13, 2017 at 8:17 pm #

REPLY ↩

ok thanks a lot,

have a nice day 🙂

riya July 5, 2017 at 10:33 pm #

REPLY ↩

i ran the above program and got error
Import error: bad magic numbers in 'keras':b'\xf3\r\n'

Jason Brownlee July 6, 2017 at 10:25 am #

REPLY ↰

You may have a copy-paste example. Check your code file.

riya July 6, 2017 at 9:43 pm #

REPLY ↰

actually a pyc file was created in the same directory due to which this error occoured.After deleting the file,error was solved

Jason Brownlee July 9, 2017 at 10:29 am #

REPLY ↰

Glad to hear it.

riya July 7, 2017 at 9:44 pm #

REPLY ↰

Hello jason,
how is the error calculated to adjust weights in neural network?does the classifier uses backpropgation or anything else for error correction and weight adjustment?

Jason Brownlee July 9, 2017 at 10:44 am #

REPLY ↰

Yes, the backpropgation algorithm is used.

riya July 9, 2017 at 7:15 pm #

REPLY ↰

Thanks jason

Jason Brownlee July 11, 2017 at 10:15 am #

REPLY ↰

You're welcome.

Nunu July 19, 2017 at 6:27 pm #

REPLY ↰

Dear Jason,
In my classifier I have 4 classes and as I know the last Dense layer should also have 4 outputs correct me please if i am wrong :). Now I want to change the number of classes from 4 to 2 !! my dataset is labeled as follows :
1) BirdYES_TreeNo
2) BirdNo_TreeNo
3)BirdYES_TreeYES
4)BirdNo_TreeYES
At the begining my output vector that i did was [0,0,0,0] in such a way that it can take 1 in the first place and all the rest are zeros if the image labeled as BirdYES_TreeNo and it can take 1 in the second place if it is labeled as BirdNo_TreeNo and so on…

Can you give me any hint inorder to convert these 4 classes into only 2 (is there a function in Python that can do this ?) class Bird and class Tree in which every class takes 2 values 1 and 0 (1 indicates the exsistance of a Bird/Tree and 0 indicates that there is no Bird/Tree). I hope that my explanation is clear.
I will appreciate so much any answer from your side.
Thanks in advance,
have a nice day,
Nunu

Jason Brownlee July 20, 2017 at 6:18 am # REPLY

Yes, the number of nodes in the output layer should match the number of classes.

Unless the number of classes is 2, in which case you can use a sigmoid activation function with a single neuron. Remember to change loss to binary_crossentropy.

Nunu July 20, 2017 at 6:07 pm # REPLY

Thanks a lot for your help i will try it.

Have a nice day,
Nunu

Jason Brownlee July 21, 2017 at 9:32 am # REPLY

Good luck!

Prathm July 26, 2017 at 8:32 am # REPLY kfold = KFold(n_splits=10, shuffle=True, random_state=seed)

This line is giving me follwing error:

File “C:\Users\pratmerc\AppData\Local\Continuum\Anaconda3\lib\site-packages\pandas\core\indexing.py”, line 1231, in _convert_to_indexer raise KeyError(‘%s not in index’ % objarr[mask])

KeyError: ‘[41421 7755 11349 16135 36853] not in index’

Can you please help ?

Jason Brownlee July 26, 2017 at 3:58 pm #

REPLY ↩

I’m sorry to hear that, perhaps check the data that you have loaded?

Q. I. August 5, 2017 at 5:20 am #

REPLY ↩

Hi,

Thanks for a great site. New visitor. I have a question. In line 38 in your code above, which is “print(encoder.inverse_transform(predictions))”, don’t you have to do un-one-hot-encoded or reverse one-hot-encoded first to do encoder.inverse_transform(predictions)?

Thanks.

Jason Brownlee August 5, 2017 at 5:49 am #

REPLY ↩

Normally yes, here I would guess that the learn wrapper predicted integers directly (I don’t recall the specifics off hand).

Try printing the outcome of predict() to confirm.

Hernando Salas August 11, 2017 at 5:16 am #

REPLY ↩

Hi Jason,

I really enjoy your tutorials awesome at presenting the material. I’m a little bit puzzled by the results of this project as I get %44 rather than %95 which is a huge difference. I have used your code as follows in ipython notebook online:

import numpy
import pandas

```
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
from sklearn.cross_validation import cross_val_score, KFold
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline

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

# load dataset
dataframe = pandas.read_csv(“iris.csv”, header=None)
dataset = dataframe.values
X = dataset[:,0:4].astype(float)
Y = dataset[:,4]

#encode class values as integers
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)

# convert integers to dummy variables (hot encoded)
dummy_y = np_utils.to_categorical(encoded_Y)

# define baseline model
def baseline_model():
# create model
model = Sequential()
model.add(Dense(4, input_dim=4, init=’normal’, activation=’relu’))
model.add(Dense(3, init=’normal’, activation=’sigmoid’))
# Compile model
model.compile(loss=’categorical_crossentropy’, optimizer=’adam’, metrics=[‘accuracy’])

return model

estimator = KerasClassifier(build_fn=baseline_model, nb_epoch=200, batch_size=5, verbose=0)
kfold = KFold(n=len(X), n_folds=10, shuffle=True, random_state=seed)
results = cross_val_score(estimator, X, dummy_y, cv=kfold)

print(“Accuracy: %.2f%% (%.2f%%)” % (results.mean()*100, results.std()*100))
```

Jason Brownlee August 11, 2017 at 6:46 am #

REPLY ↰

The algorithm is stochastic, so you will get different results each time it is run, try running it multiple times and take the average.

More about the stochastic nature of the algorithms here:
http://machinelearningmastery.com/randomness-in-machine-learning/

Hernando Salas August 15, 2017 at 5:41 am #

REPLY ↰

Hi Jason,

Thanks for the reply. Run several times and got the same result. Any ideas?

Hernando Salas August 15, 2017 at 5:43 am #

REPLY ↰

https://notebooks.azure.com/hernandosalas/libraries/deeplearning/html/main.ipynb

Jason Brownlee August 15, 2017 at 6:45 am #

REPLY ↰

You could try varying the configuration of the network to see if that has an effect?

Hernando Salas August 16, 2017 at 5:02 am #

REPLY ↰

If I set it to:

```
# create model
model = Sequential()
model.add(Dense(4, input_dim=4, init=’normal’, activation=’relu’))
model.add(Dense(3, init=’normal’, activation=’sigmoid’))
```

I get Accuracy: 44.00% (17.44%) everytime

If I set it to:

```
# create model
model = Sequential()
model.add(Dense(8, input_dim=4, init=’normal’, activation=’relu’))
model.add(Dense(3, init=’normal’, activation=’softmax’))
```

I get Accuracy: 64.00% (10.83%) everytime

Jason Brownlee August 16, 2017 at 6:40 am # REPLY

Interesting. Thanks for sharing.

Akash August 22, 2017 at 12:42 am # REPLY

Hi Jason,

Thank you for your wonderful tutorial and it was really helpful. I just want to ask if we can perform grid search cv also the similar way because I am not able to do it right now?

Jason Brownlee August 22, 2017 at 6:44 am # REPLY

Yes, see this post:
https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/

Alexander September 9, 2017 at 6:56 am # REPLY

Hi, Jason. Thank you for beautiful work.
Help me please.
Where (in which folder, directory) should i save file "iris.csv" to use this code? Now system doesn't see this file, when I write "dataframe=pandas.read_csv…."

4. Load The Dataset
The dataset can be loaded directly. Because the output variable contains strings, it is easiest to load the data using pandas. We can then split the attributes (columns) into input variables (X) and output variables (Y).
# load dataset
dataframe = pandas.read_csv("iris.csv", header=None)
dataset = dataframe.values
X = dataset[:,0:4].astype(float)
Y = dataset[:,4]

Jason Brownlee September 9, 2017 at 12:01 pm # REPLY

Download it and place it in the same directory as your Python code file.

Alexander September 9, 2017 at 5:59 pm #

REPLY

Thank you, Jason. I'll try.

Leave a Reply

Name (required)

Email (will not be published) (required)

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.

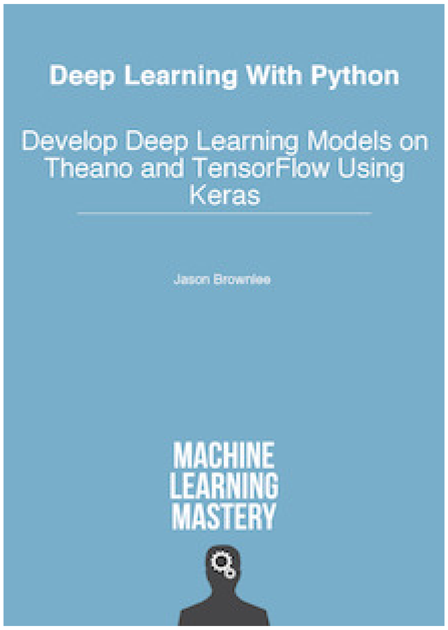
Read More

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 Started With Deep

Get Your Start in Machine Learning



Deep Learning With Python

Develop Deep Learning Models on Theano and TensorFlow Using Keras

Jason Brownlee

MACHINE LEARNING MASTERY

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

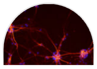
How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda
MARCH 13, 2017

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

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

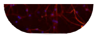
Time Series Forecasting with the Long Short-Term Memory Network in Python
APRIL 7, 2017

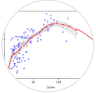
Regression Tutorial with the Keras Deep Learning Library in Python
JUNE 9, 2016



How to Implement the Backpropagation Algorit
NOVEMBER 7, 2016

Get Your Start in Machine Learning





A Tour of Machine Learning Algorithms
NOVEMBER 25, 2013

Privacy | Contact | About