

[Start Here](#)[Blog](#)[Books](#)[About](#)[Contact](#)

Need help with Deep Learning? [Take the FREE Mini-Course](#)

# 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.

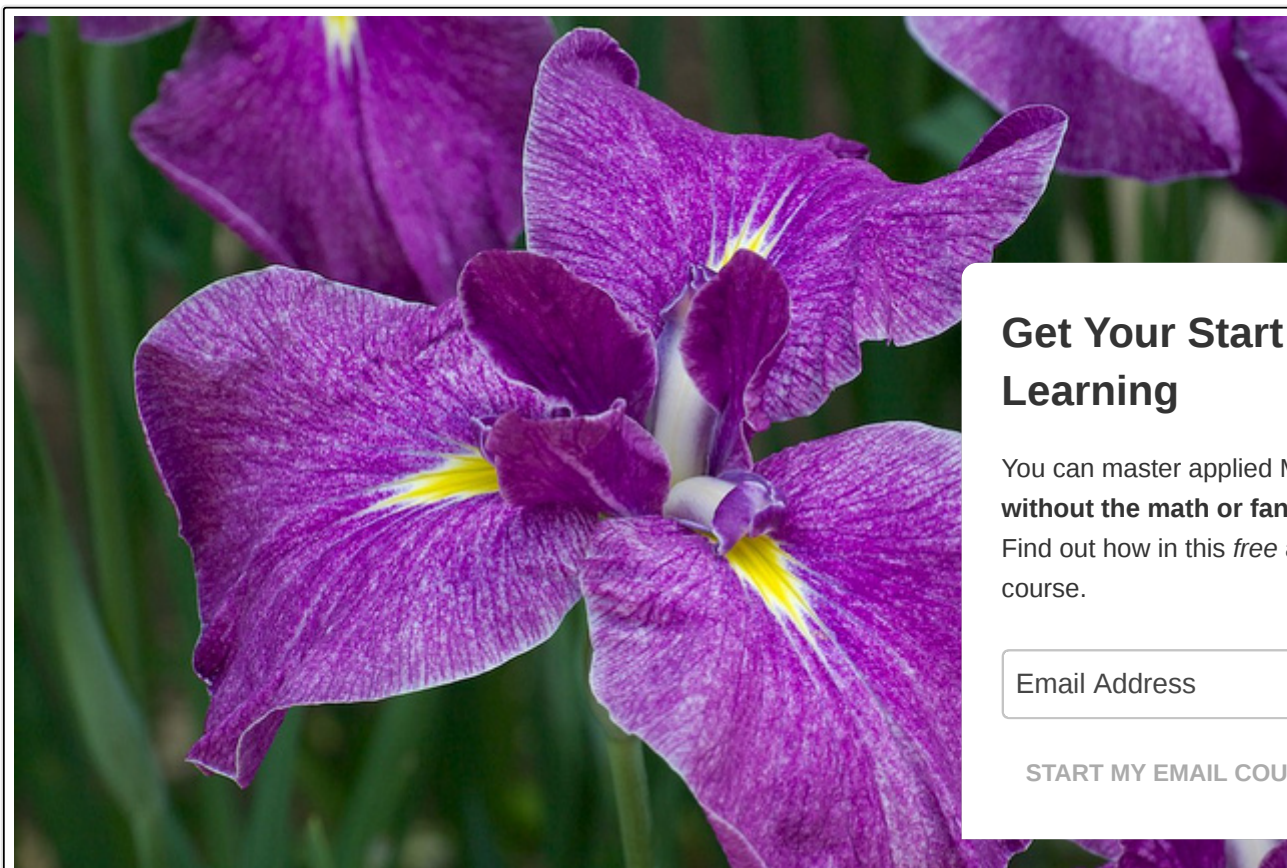
After completing this step-by-step tutorial, you will know:

- 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.

[Get Your Start in Machine Learning](#)

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.

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

## 1. Problem Description

In this tutorial, we will use the standard machine learning problem called the [iris flowers dataset](#).

Get Your Start in Machine Learning

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 as 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 Learning in Python?

Take my free 2-week email course and discover MLPs, CNNs and LSTMs

Click to sign-up now and also get a free PDF Ebook version of the course

Start Your FREE Mini-Course Now!

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

## 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](#).

```
1 import numpy
2 import pandas
3 from keras.models import Sequential
```

Get Your Start in Machine Learning

```
4 from keras.layers import Dense
5 from keras.wrappers.scikit_learn import KerasClassifier
6 from keras.utils import np_utils
7 from sklearn.model_selection import cross_val_score
8 from sklearn.model_selection import KFold
9 from sklearn.preprocessing import LabelEncoder
10 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.

```
1 # fix random seed for reproducibility
2 seed = 7
3 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 attributes (columns) into input variables (X) and output variables (Y).

```
1 # load dataset
2 dataframe = pandas.read_csv("iris.csv", header=None)
3 dataset = dataframe.values
4 X = dataset[:,0:4].astype(float)
5 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.

#### Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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

```
1 Iris-setosa
2 Iris-versicolor
3 Iris-virginica
```

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

```
1 Iris-setosa,   Iris-versicolor,   Iris-virginica
2 1,           0,                 0
3 0,           1,                 0
4 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()`.

```
1 # encode class values as integers
2 encoder = LabelEncoder()
3 encoder.fit(Y)
4 encoded_Y = encoder.transform(Y)
5 # convert integers to dummy variables (i.e. one hot encoded)
6 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

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

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 must create 3 output values, one for each class. The 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:

### Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

```
1 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.

```
1 # define baseline model
2 def baseline_model():
3     # create model
4     model = Sequential()
5     model.add(Dense(8, input_dim=4, activation='relu'))
6     model.add(Dense(3, activation='softmax'))
7     # Compile model
8     model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
9     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 model. Here, we pass the number of epochs as 200 and batch size as 5 to use when training the model. We also set verbose to 0.

```
1 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.

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

### Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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.

```
1 results = cross_val_score(estimator, X, dummy_y, cv=kfold)
2 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.

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

## Summary

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

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.

### Get Your Start in Machine Learning

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

START MY EMAIL COURSE

## Frustrated With Your Progress In Deep Learning?

What If You Could Develop A Network in Min

Get Your Start in Machine Learning



...with just a few lines of Python

Discover how in my new Ebook: [Deep Learning With Python](#)

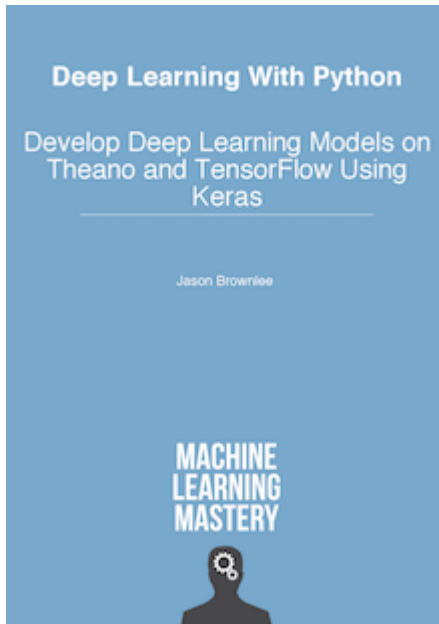
It covers **self-study tutorials** and **end-to-end projects** on topics like:

*Multilayer Perceptrons, Convolutional Nets and Recurrent Neural Nets*, and more...

## Finally Bring Deep Learning To Your Own Projects

Skip the Academics. Just Results.

[Click to learn more.](#)



### About Jason Brownlee

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

[Learn more.](#)

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

## Get Your Start in Machine Learning

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

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

[START MY EMAIL COURSE](#)

[◀ How To Compare Machine Learning Algorithms in Python with scikit-learn](#)

[Ensemble Machine Learning Algorithms in Python with scikit-learn ▶](#)

[Get Your Start in Machine Learning](#)



## 266 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

**Shraddha** February 10, 2017 at 8:32 pm #

Could you please let me know how to convert string data into word embeddings in large  
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.

### Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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

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

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 ↩

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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 #

Interesting. Perhaps seeding the random number generator is not having the desired effects on different platforms.

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

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

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!

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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 #

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

```
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 = 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\_value).Use the Theano flag exception\_verbosity='high',for more information on this error.

Backtrace when the variable is created:

File “nn\_systematics | 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

## Get Your Start in Machine Learning

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

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

Email Address

**START MY EMAIL COURSE**

## Get Your Start in Machine Learning

```
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 model was not properly serialized and put back inside the wrapper.

**Sally** July 15, 2016 at 4:10 am #

Dear Dr. Jason,

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

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

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>

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

**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 #

Hi Fabian, no problem.

In the tutorial above, we are using the scikit-learn wrapper. That means we can use the standard model from scikit-learn.

For example, below is an example adapted from the above where we split the dataset, train on 60% and test on 40%. We have encoded the output class value as integers, so the predictions are integers. We can then use the `LabelEncoder` to convert the integers back into strings.

```
1 # Train model and make predictions
2 import numpy
3 import pandas
4 from keras.models import Sequential
5 from keras.layers import Dense
6 from keras.wrappers.scikit_learn import KerasClassifier
7 from keras.utils import np_utils
8 from sklearn.cross_validation import train_test_split
9 from sklearn.preprocessing import LabelEncoder
10 # fix random seed for reproducibility
11 seed = 7
12 numpy.random.seed(seed)
13 # load dataset
14 dataframe = pandas.read_csv("iris.csv", header=None)
15 dataset = dataframe.values
16 X = dataset[:,0:4].astype(float)
17 Y = dataset[:,4]
18 # encode class values as integers
```

REPLY ↩

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```

19 encoder = LabelEncoder()
20 encoder.fit(Y)
21 encoded_Y = encoder.transform(Y)
22 # convert integers to dummy variables (i.e. one hot encoded)
23 dummy_y = np_utils.to_categorical(encoded_Y)
24 # define baseline model
25 def baseline_model():
26     # create model
27     model = Sequential()
28     model.add(Dense(4, input_dim=4, init='normal', activation='relu'))
29     model.add(Dense(3, init='normal', activation='sigmoid'))
30     # Compile model
31     model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
32     return model
33 estimator = KerasClassifier(build_fn=baseline_model, nb_epoch=200, batch_size=5, verbose=0)
34 X_train, X_test, Y_train, Y_test = train_test_split(X, dummy_y, test_size=0.33, random_state=seed)
35 estimator.fit(X_train, Y_train)
36 predictions = estimator.predict(X_test)
37 print(predictions)
38 print(encoder.inverse_transform(predictions))

```

Running this example prints:

```

1  [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
2   0 0 2 2 1 2 2 1 0 1 1 2 0]
3  ['Iris-virginica' 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor'
4   'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
5   'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
6   'Iris-setosa' 'Iris-virginica' 'Iris-setosa' 'Iris-virginica'
7   'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
8   'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica'
9   'Iris-virginica' 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor'
10  'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor'
11  'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
12  'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
13  'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
14  'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
15  'Iris-virginica' 'Iris-setosa']

```

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

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE



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 #

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 c

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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 #

Very strange.

Maybe check that your data file is correct, that you have all of the code and that your environ

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

---

**Andrea** December 12, 2017 at 7:17 am #

Jason, I'm getting the same prediction (all zeroes) with Keras 2. If we could be able to nail the cause, it would be great. After all, as of now it's more than likely that people will try to run your great examples with keras 2.

Plus, a couple of questions:

Get Your Start in Machine Learning

1. why did you use a sigmoid for the output layer instead of a softmax?
2. why did you provide initialization even for the last layer?

Thanks a lot.

---

**Jason Brownlee** December 12, 2017 at 4:02 pm #

The example does use softmax, perhaps check that you have copied all of the code from the post?

---

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

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 the model?  
For example, I am predicting regression or multiclass classification. I have to use KerasRegressor or KerasClassifier.  
If I have a large amount of data, I want to save the trained neural network model to use it for prediction purpose only. How to save the model to a file?  
files ? Your answer will help me a lot.

---

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

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 ↩

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

Hi Jason, Thank your very much for those nice explanations.

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?

Reinier

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

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 #

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.

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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 #

Sorry I do not have any examples for clustering.

**Prash** August 14, 2016 at 9:15 pm #

Jason, boss you are too good! You have really helped me out especially in implementation of D 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.

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**Harsha** August 18, 2016 at 7:03 pm #

REPLY ↩

It is a great tutorial Dr. Jason. Very clear and crispy. I am a beginner in Keras. I have a small 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 #

Thank You Jason for your prompt reply

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

You're welcome Harsha.

**jokla** January 12, 2017 at 7:30 am #

Hi Jason, nice tutorial!

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

Thanks!

## Get Your Start in Machine Learning



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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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 class classification. I used the Tfidfvectoriser from sklearn so a one hot vector representation against a given vocabulary is used for classification.

I am having trouble understanding the initial steps in transforming and feeding word data into vector representation. I have code examples of this first step in the sense that say I have a text file with 5000 words for example, which I can feed in a training file in csv format text,sentiment and convert each text into a one hot representation vector of size e.g 1x7 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 Tfidfvectoriser.

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!

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning



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 variable consists of discrete integers, 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

Try it and compare the results.

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

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

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/external

Get Your Start in Machine Learning

```
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 #

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 #

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 #

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

REPLY ↩

Get Your Start in Machine Learning

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\_crossentropy.

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 #

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

**Preston** September 12, 2017 at 11:14 pm #

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 ↩

Get Your Start in Machine Learning

## Get Your Start in Machine Learning



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

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

START MY EMAIL COURSE

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. The input 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
```

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

---

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

REPLY ↩

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

Get Your Start in Machine Learning

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 data causing this error:

"Exception: Error when checking model target: expected dense\_2 to have shape (None, 3) but got array of shape (None, 4)"

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

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

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>

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE



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

REPLY ↩

Get Your Start in Machine Learning

Hello, I tried to use the exact same code for another dataset , the only difference being the dataset had 78 columns and 10000 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 everyt

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

Hi Homagni,

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

Your batch size is probably too big and your number of epochs is way too small. Dramatically increase th

Start there and let me know how you go.

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

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

Hi Jason,

REPLY ↩

Get Your Start in Machine Learning

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 #

Hello Jason,

Is there a error in your code? You said the network has 4 input neurons , 4 hidden neurons and 3 output 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 #

Hi Panand,

The network structure is as follows:

```
1 4 inputs -> [4 hidden nodes] -> 3 outputs
```

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

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

The input\_dim argument defines the shape of the input.

## Get Your Start in Machine Learning



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

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

START MY EMAIL COURSE



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 #

Hi JD,

Here is a long list of ideas to improve the skill of your deep learning model:

<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,

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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 #

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

**Tom** December 9, 2016 at 12:13 am #

Hello Jason,

Great work on your website and tutorials! I was wondering if you could show a multi hot encoding, I think

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]

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

[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 #

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

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

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

**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:

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

Get Your Start in Machine Learning

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  
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)

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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 #

Hello Seun, perhaps this could help you: <http://stackoverflow.com/questions/41796618/python-error-get-params-got-an-unexpected-keyword-argument-deep>

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

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'

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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 #

Hi Jasson,

Thanks so much. The second fix worked for me.

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

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 28\*28 pixels as totally x matrix is (5000,400) and Y is (5000,1), i am not able to successfully run the model getting error

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
#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 reproductibility
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],
 [10],
```

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning



```
...,
[ 9],
[ 9],
[ 9]], dtype=uint8), '__globals__': {}}
```

Type Markdown and LaTeX:

$\alpha$

2

$\alpha^2$

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]:

## Get Your Start in Machine Learning ×

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

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

START MY EMAIL COURSE

```
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.]])
```

## Get Your Start in Machine Learning ×

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
[ 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]:
```

```
#encode 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.],
```

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
[ 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_state=seed)
```

## Get Your Start in Machine Learning

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

**START MY EMAIL COURSE**

```
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]
```

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

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) but the output is only one category, the problem is more specifically “single-label, multi-class classification” categories. Then we are facing “multi-label, multi-class classification”. In our case, each flower belongs to one or more categories (e.g. [1 1 0], [0 1 1], [1 1 1]..... This is a multi-label problem (right?).

My solution is to modify the output variable (Y) with multiple ‘1’ in it, i.e. [1 1 0], [0 1 1], [1 1 1]..... This is a multi-label problem (right?).

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

Thanks in advance

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

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?

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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?

**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 problem where each line of text has a score. I want to predict scores to different lines of text, and the problem is to infer the score for a new line of text. It seems that the solution would be the fix for this?

Thanks in advance for the help.

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

**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/>

Get Your Start in Machine 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 problem.

**Akash** March 22, 2017 at 5:47 pm #

Hi Jason,

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

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

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 #

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning



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', activation='relu'))
model.add(Dense(2, init='normal', activation='sigmoid'))
```

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
# 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
```

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

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

**Alican** March 28, 2017 at 4:05 am #

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.

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

REPLY ↩

Get Your Start in Machine Learning

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 `results=cross_val_score(...)` line to get different results I 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 #

Glad to hear it.

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

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 performance.

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

**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.

Get Your Start in Machine Learning

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 am #

Yes I meant Keras, sorry.

There is no issue with the seed, I'm getting the same result with you on multiple computations. My results are abysmally bad.

---

**Jonathan** July 11, 2017 at 4:28 am #

not sure if this was ever resolved, but I'm getting the same thing with most recent versions of Theano and Keras  
59.33% with seed=7

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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

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

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

**Nalini** March 31, 2017 at 1:42 am #

thanks

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

REPLY ↩

You're welcome.

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

**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 #

This process will help you work through your modeling problem:

<http://machinelearningmastery.com/start-here/#process>

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

Hi Jason,

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

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

Thanks for the suggestion.

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

REPLY ↩

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

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 #

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'])
```

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)
```

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
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 and case-2 are different?

Any solution?

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

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 #

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

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning



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

Thanks for the great work on your tutorials... for beginners it is such in invaluable thing to have tutorials

Looking forward to get more of your books

Rene

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

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"
}
```

## Get Your Start in Machine Learning ×

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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 #

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

Take care,

Rene

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

**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 ev

Get Your Start in Machine Learning

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 Keras config file. See this post:

<http://machinelearningmastery.com/introduction-python-deep-learning-library-keras/>



**Tursun** April 16, 2017 at 9:18 pm #

Jason,

Thank you very much first. These tutorials are excellent. They are very practical. Your are an excellent e  
I want classify my data into multiple classes of 25-30. Your IRIS example is nearest classification. They I  
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

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

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

Get Your Start in Machine Learning

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 performance, see this post:

<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  $5 \times 200 = 1000$  examples for training. So does keras automatically?

Thanks!

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

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?

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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 #

Hey Jason,

I have been following your tutorials and they have been very very helpful!. Especially, the most useful sections are the ones where you ask 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" dataset? It has 8 categorical inputs and 1 binary output.

This would be a huge help! Thanks in advance!

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

**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.

Get Your Start in Machine Learning

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 #

I recommend testing a suite of different batch sizes.

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 .

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

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#im](https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/get_started/os_setup.md#im)

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 #

Ouch. I have not seen this error before.

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

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE



**Tursun** April 21, 2017 at 2:17 am #

Jason,

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

REPLY ↩

Get Your Start in Machine Learning

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](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 this is more suited to classification problems."

error message:

ValueError: Error when checking model target: expected dense\_8 to have shape (None, 21) but got array

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
```

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning



```
# 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], scores[1]*100))

————keras code start ————
```

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

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

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

**Shiva** April 23, 2017 at 5:54 am #

Hi Jason, I am following your book deep learning with python and I have an issue with the script pandas and trying to adopt a decay based learning rate as discussed in the book. I define the initial rate as said in the book. I then created the model like this (works best for my problem) and started creating a 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
```

REPLY ↩

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
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 #

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 #

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.

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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

Many Thanks!

Michael Ng

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

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

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.'

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

Instead of using softmax function, how do I review the sigmoidal outputs (as per the tutorial) for each of 5 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](https://en.wikipedia.org/wiki/Logistic_function)

---

**Andrea** December 12, 2017 at 7:59 am #

Jason,

may you elaborate further (or provide a link) about “the outputs from the softmax, although not s

I thought they were probabilities even in the most formal sense.

Thanks!

---

**Jason Brownlee** December 12, 2017 at 4:04 pm #

REPLY ↩

No, they are normalized to look like probabilities.

This might be a good place to start:

[https://en.wikipedia.org/wiki/Softmax\\_function](https://en.wikipedia.org/wiki/Softmax_function)

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

---

Get Your Start in Machine Learning

**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\_type)

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 #

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 nun

Get Your Start in Machine Learning

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

**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 #

@Jason Brownlee I totally agree with you. We are using this problem as proxy for multiple cars and we want to classify the models of these cars. The same approach is needed

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

first this is a great tutorial , but , am confused a little ,, am i loading my training files and labeling as i tried to apply this tutorial to my case ,, I've about 10 folder each has its own images these images are multi labeling for each folder of them for example folder number 1 has about 1500 .png imgs of owl bird a bird and owl , and here comes the problem ,, as i'm searching for a tool to make labeling for all images ... 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/>

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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 #

Hi Jason, Just gone through your blog <http://machinelearningmastery.com/> .Just to know as a b 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 l 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

## Get Your Start in Machine Learning



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

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

START MY EMAIL COURSE

**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.

Get Your Start in Machine Learning

**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 #

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 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 is (6, 6) instead of (6, 7)? Any idea what might be going on? I checked for issues in my dataset such as null values in the output variable and it 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 values are present.

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning



**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 #

With multiple classes, it might be better to use another metric like log loss (cross entropy).  
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 re-sampling. See <http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-model/>

**Nalini** May 24, 2017 at 6:10 pm #

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....

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

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

Get Your Start in Machine Learning

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](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 #

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 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.

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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 classification (6 labels representing sentiment of tweets). I am fold just as in your example and it yields 100% 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. I have been looking for a way to do this and apparently a good approach is to use a confusion matrix. Is classification, or will other methods suffice?

Thanks

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

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.

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

**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:

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

**zakaria** June 12, 2017 at 6:35 pm #

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

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**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 pro  
Thanks.

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

I don't have a great answer for you off the cuff. I would suggest doing a little research to see what  
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 d

Let me know how you go.

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE



**Anastasios** June 17, 2017 at 10:05 pm #

REPLY ↩

Hello Jason,

Get Your Start in Machine Learning

great post on multiclass classification. I am trying to do a gridsearch on a multiclass dataset i created, but 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 #

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 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

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
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'))
```

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
# 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 #

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

---

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

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

---

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

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 ↩

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

REPLY ↩

Get Your Start in Machine Learning



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", 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)
```

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
# 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 #

Dear Jason,

How can I increase the accuracy while training ? I am always getting an accuracy around 68% and 70% function and the learning rate.  
(I am using keras and CNN)

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

Here are many ideas:

<http://machinelearningmastery.com/machine-learning-performance-improvement-cheat-sheet/>

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

REPLY ↩

Get Your Start in Machine Learning



**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 #

Dear Jason,

I have a question: my model should classify every image in one of the 4 classes that I have, instead the “Binary cross entropy” ? because I read a lot that when there is n classes it is be binary one is used for the same cases. I am too much confused 😞 can you help me in unde  
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 so

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

program with the same learning rates used in the case of softmax, and here I got the problem: using learning rate 0.001 I got loss and various 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, 2017 at 8:27 am #

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 #

Hi Jason,

Thanks for the awesome tutorial. I have a question regarding your first hidden layer which has 8 neurons. Can 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 see which one gives the best results on your problem.

**Get Your Start in Machine Learning**

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE



**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 #

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

**riya** July 6, 2017 at 9:43 pm #

actually a pyc file was created in the same directory due to which this error occurred. After deleting it, the error was resolved.

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

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

REPLY ↩

Glad to hear it.

Get Your Start in Machine Learning

**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 #

Thanks jason

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

You're welcome.

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

REPLY ↩



**Nunu** July 19, 2017 at 6:27 pm #

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

Get Your Start in Machine Learning

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 exsistence 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 #

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 `binary_crossentropy`.



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

Thanks a lot for your help i will try it.

Have a nice day,  
Nunu

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

Good luck!

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

REPLY ↩

Get Your Start in Machine Learning

**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 #

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

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

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))", should I 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.

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning



**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)
```

## Get Your Start in Machine Learning ×

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
# 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 #

The algorithm is stochastic, so you will get different results each time it is run, try running it

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 #

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 #

## Get Your Start in Machine Learning ×

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

START MY EMAIL COURSE

REPLY ↩

---

Get Your Start in Machine Learning

<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

## Get Your Start in Machine Learning



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

[START MY EMAIL COURSE](#)**Jason Brownlee** August 16, 2017 at 6:40 am #

REPLY ↩

Interesting. Thanks for sharing.

[Get Your Start in Machine Learning](#)

**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 #

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 t

#### 4. Load The Dataset

The dataset can be loaded directly. Because the output variable contains strings, it is easiest to load the (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]
```

## Get Your Start in Machine Learning



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

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

START MY EMAIL COURSE

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

REPLY ↩

Get Your Start in Machine Learning

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.

**Tran Minh** September 18, 2017 at 4:29 pm #

REPLY ↩

Hi Jason, thank you for your great instruction

I follow your code but unfortunately, I get only 68%~70% accuracy rate.

I use Tensorflow backend and modified seed as well as the number of hidden units but I still can't reach

Do you have any idea how to improve it

**Jason Brownlee** September 19, 2017 at 7:32 am #

Perhaps try running the example a few times, see this post:

<https://machinelearningmastery.com/randomness-in-machine-learning/>

**Greg** September 21, 2017 at 8:23 am #

Jason,

First thanks so much for a great post.

I cut and pasted the code above and got the following run times with a GTX 1060

real 2m49.436s

user 4m46.852s

## Get Your Start in Machine Learning



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

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

START MY EMAIL COURSE

REPLY ↩

Get Your Start in Machine Learning

```
sys 0m21.944s
```

and running without the GPU

```
124.93 user 25.74 system 1:04.90 elapsed 232% CPU
```

Is this reasonable? It seems slow for a toy problem.

---

**Jason Brownlee** September 21, 2017 at 4:19 pm <#>

REPLY 

Thanks for sharing.

Yes, LSTMs are slower than MLPs generally.

**Bee** September 27, 2017 at 1:20 am <#>

Hi Dr. Jason,

It's a great tutorial. Do you have any similar tutorials for Unsupervised classification too?

Thanks,  
Bee

---

**Jason Brownlee** September 27, 2017 at 5:43 am <#>

Unsupervised methods cannot be used for classification, only supervised methods.

---

**Bee** October 2, 2017 at 5:08 am <#>

REPLY 

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

---

Get Your Start in Machine Learning

Sorry, it was my poor choice of words. What I meant was clustering data using unsupervised methods when I don't have labels. Is that possible with Keras?

Thanks,  
Bee

---

**Jason Brownlee** October 2, 2017 at 9:40 am #

REPLY ↩

It may be, but I do not have examples of working with unsupervised methods, sorry.

---

**Miqueias** October 3, 2017 at 8:48 am #

Hi Jason,

Thanks for your work describing in a very nice way how to use Keras! I've a question about the performance. Suppose you have a class for something you call your signal and, then, many other classes which you want to be efficient to work on Keras: merging the different background classes and considering all of them as just one class for classification or use a categorical one to account all the classes? In other words, is one way more sensible than the other from all the classes?

---

**Jason Brownlee** October 3, 2017 at 3:44 pm #

Great question.

It really depends on the specific data. I would recommend designing some experiments to see what works best.

---

**Miqueias** October 3, 2017 at 10:52 pm #

REPLY ↩

## Get Your Start in Machine Learning ×

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

Thanks for fast replay Jason!

I'll try that to see what I get.

I'm wondering if in categorical classification Keras can build up independent functions inside it. Because, since the background classes may exist in different phase space regions (what would be more truthfully described by separated functions), training the net with all of them together for binary classification may not extract all the features from each one. In principle, that could be done with a single net but, it would probably require more neurons (which increases the over-fitting issue).

By the way, what do you think about training different nets for signal vs. each background? Could they be combined in the end?

**Jason Brownlee** October 4, 2017 at 5:45 am #

REPLY ↩

If the classes are separable I would encourage you to model them as separate problems.

Nevertheless, the best advice is always to test each idea and see what works best on your problem.

**Dave** October 11, 2017 at 5:22 am #

Hi Jason! I have a question about multi classification

I would like to classify the 3 class of sleep disordered breathing.

I designed the LSTM network. but it works like under the table.

What is this situation?

Train matrix: precision recall f1-score support

0 0.00 0.00 0.00 1749

1 0.46 1.00 0.63 2979

2 0.00 0.00 0.00 1760

avg / total 0.21 0.46 0.29 6488

Train matrix: precision recall f1-score support

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning



0 0.00 0.00 0.00 441  
1 0.46 1.00 0.63 750  
2 0.00 0.00 0.00 431

avg / total 0.21 0.46 0.29 1622

**sasi** October 13, 2017 at 10:48 pm #

REPLY ↩

Hi Jason,

Does this topic will match for this tutorial??

“Deep learning based multiclass classification tutorial”

**Jason Brownlee** October 14, 2017 at 5:46 am #

Yes.

**zaheer** October 16, 2017 at 11:48 pm #

This tutorial is awsom. thanks for your time.

My data is

404. instances

2. class label. A/B

20. attribute columns.

i have tried the this example gives me 58% acc.

```
model = Sequential()
```

```
model.add(Dense(200, input_dim=20, activation='relu'))
```

```
model.add(Dense(2, activation='softmax'))
```

```
# Compile model
```

## Get Your Start in Machine Learning



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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
return model

#Classifier invoking
estimator = KerasClassifier(build_fn=baseline_model, epochs=200, batch_size=5, verbose=0)

what should i do, how to increase the acc of the system
```

**Jason Brownlee** October 17, 2017 at 5:47 am #

REPLY ↩

See this post for a ton of ideas:

<http://machinelearningmastery.com/improve-deep-learning-performance/>

**Curious\_Kid** October 24, 2017 at 1:00 am #

Hi Jason,

My training data consists of lines of characters with each line corresponding to a label.

E.g. afhafajkfajkfahkahfahk 6

fafafafafaftuiowjtwojdfafanfa 8

dakworfwajanfnafjajfahifqnfqqfnq 4

Here, 6,8 and 4 are labels for each line of the training data.

.....

I have first done the integer encoding for each character and then done the one hot encoding. To keep the integer encoding consistent, I first looked for the unique letters in all the rows and then did the integer encoding. e.g. that's why letter h will always be encoded as 7 in all the lines.

For a better understanding, consider a simple example where my training data has 3 lines(each line has some label):

af

fa

nf

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

It will be one hot encoded as:

0 [[1.0, 0.0, 0.0], [0.0, 1.0, 0.0]]

1 [[0.0, 1.0, 0.0], [1.0, 0.0, 0.0]]

2 [[0.0, 0.0, 1.0], [0.0, 1.0, 0.0]]

I wanted to do the classification for the unseen data(which label does the new line belong to) by training a neural network on this one hot encoded training data.

I am not able to understand how my model should look like as I want the model to learn from each one hot encoded character for each line. Could you please suggest me something in this case? Please let me know if you need more information to understand the problem.

**Jason Brownlee** October 24, 2017 at 5:33 am #

This is a sequence classification task.

Perhaps this post will give you a template to get started:

<https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/>

**Curious\_Kid** October 24, 2017 at 6:10 am #

Thanks Jason for the reply.

However, I am not dealing with words. I just have characters in a line and I am doing one hot encoding as explained above. What I am confused with is the shapes that I have to give to the layers of my neural network.

**Jason Brownlee** October 24, 2017 at 3:56 pm #

I see, perhaps this post will help with reshaping your data:

<https://machinelearningmastery.com/reshape-input-data-long-short-term-memory-networks-keras/>

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

REPLY ↩

Get Your Start in Machine Learning

**philippe** November 6, 2017 at 9:10 pm #

REPLY ↩

Hello Jason,

very clear tutorial. one quick question, how do you decide on the number of hidden neurons (in classification case). it seems to follow (Hidden neurons = input \* 2) , how about \* 1 or \*3 is there a rule. same goes for epoch ; how do you choose nbr of iterations;

thanks.

**Jason Brownlee** November 7, 2017 at 9:49 am #

REPLY ↩

There are no good rules, use trial and error or a robust test harness and a grid search.

**Niklas Wilke** November 13, 2017 at 11:26 pm #

Hi Jason,

great tutorial!

I've got a multi class classification problem. I try to classify different kind of bills into categories (that are e.g. restaurants, hotels and so on).

I got a couple files in PDF which i transform in PNG to make it processable by MC Computer Vision using pytesseract. After that i come out with a .txt or .csv file of the plain text.

Now i used sklearn's vectorizers to create a bag of words and fit the single bills/documents.

Ending up with numpy-arrays looking like this (sample data i used to create the code while i was gathering data):

```
[[3 0 1 1 0 0 0 0 2 0 2 2 1 3 1 1 0 3 0 0 3 2 1 0 1 3 1 0 0 5 0 0 1 1 0 1 0
 0 1 1 1 1 0 1 0 1 0 1 0 2 0 2 1 0 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 0 0 1 1 1
 0 1 1 0 0 0 0 1 0 0 0 1 0 0 1 1 1 2 1 0 0 0 0 0 0 0 2 1 0 0 0 2 1 0 1 0 1
 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 2 0 0 0 0 0 0 2 0 1
 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1 1 1 0 2 0 0
 0 1 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 2 3 0 1 0 0 0 0 0 1 0 3 0 1 0
```

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
1 1 0 0 0 0 0 0 1 2 0 0 0 3 0 0 0 1 0 0 0 1 1 0 2 0 0 0 0 1 0 1 1 0 0 1 0
1 1 0 1 0 0 1 0 0 0 0 1 0]]
```

How do i categoryze or transform this to something like the iris dataset ?

Isn't it basically the same ? Just with way more numbers and bigger arrays ?

I tried to iterate through the array to print every single number in a .csv-file and then just append the category at the back with some for loops but sadly you can't iterate through numpy-arrays ... + i can't imagine that's the intended way of labeling data ...

Thanks for reading through this way too long comment , help is highly apreciated.

**Jason Brownlee** November 14, 2017 at 10:18 am #

REPLY ↩

Yes, the vectorized documents become input to ML algorithms.

I'd love to hear how you go, post your results!

**Niklas Wilke** November 30, 2017 at 1:27 am #

Finally solved all my preprocessing problems and today i was able to perform my first train  
buffer\_y = dummy\_y)

```
1 def createModell():
2     #8137 words = input shape
3     #14 categorys = output shape
4
5     model = Sequential()
6     model.add(Dense(8137, input_dim = 8137, activation='relu'))
7     model.add(Dense(2250, activation='relu'))
8     #model.add(Dense(581, activation='relu'))
9     model.add(Dense(14, activation='softmax'))
10
11     model.compile(loss= 'categorical_crossentropy', optimizer='adam', metrics = ['accuracy'])
12     return model
13
14 estimator = KerasClassifier(build_fn = createModell, epochs = 10, batch_size = 64)
15 crossvalidation_data = KFold(n_splits = 39, shuffle = True)
16 results = cross_val_score(estimator, X, buffer_y, cv= crossvalidation_data)
```

## Get Your Start in Machine Learning



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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

And hell am i overfitting.

0.98 acuraccy , which can't be because my dataset is horribly unbalanced. (maybe thats the issue?)

Anyhow, i enabled the print option and for me it only displays 564/564 sample files for every epoche even though my dataset contains 579 ... i check for you example and it also only displays 140/140 even though the iris dataset is 150 files big.

Are the splits to high ?

and what is a good amount of nodes for such a high input shape ./ tried to split it up to multiple layers so its not 8139 -> 4000-> 14

Cheers

Niklas

**Jason Brownlee** November 30, 2017 at 8:20 am #

Well done!

Consider the options in this post for imbalanced data:

<https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine>

The count is wrong because you are using cross-validation (e.g. not all samples for each run)

You must use trial and error to explore alternative configurations, here are some ideas:

<http://machinelearningmastery.com/improve-deep-learning-performance/>

I hope that helps as a start.

**Niklas Wilke** November 30, 2017 at 6:52 pm #

Ah ok , good point. When i create 10 splits it only uses 521 files => 90% of 579

Will look into it and post my hopefully sucessfull results here.

Given that i had no issue with the imbalance of my dataset, is the general amount of nodes or layers alright ? I have literally no clue because all the tipps ive found so far refer to way smaller input shapes like 4 or 8.

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

**Jason Brownlee** December 1, 2017 at 7:28 am #

There are no good rules of thumb, I recommend testing a suite of configurations to see what works best for your problem.

**Niklas Wilke** November 30, 2017 at 7:34 pm #

I read you mentioned other classifiers like decision trees performing well on imbalanced datasets.

Is there some way i can use other classifiers INSIDE of my NN ?

for example could i implement naive bayes into my NN ?

**Jason Brownlee** December 1, 2017 at 7:29 am #

Not that I am aware.

You could combine the predictions from multiple models into an ensemble though.

**Niklas Wilke** November 30, 2017 at 6:59 pm #

Btw, even though i tell it to run 10 epoches , after the 10 epoches it just starts again with slightly

Epoch 1/10

521/521 [=====] – 12s – loss: 2.0381 – acc: 0.4952

Epoch 2/10

521/521 [=====] – 10s – loss: 0.3139 – acc: 0.9443

Epoch 3/10

521/521 [=====] – 10s – loss: 0.0748 – acc: 0.9866

Epoch 4/10

## Get Your Start in Machine Learning



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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
521/521 [=====] - 11s - loss: 0.0578 - acc: 0.9942
Epoch 5/10
521/521 [=====] - 11s - loss: 0.0434 - acc: 0.9962
Epoch 6/10
521/521 [=====] - 11s - loss: 0.0352 - acc: 0.9962
Epoch 7/10
521/521 [=====] - 11s - loss: 0.0321 - acc: 0.9981
Epoch 8/10
521/521 [=====] - 11s - loss: 0.0314 - acc: 0.9981
Epoch 9/10
521/521 [=====] - 11s - loss: 0.0312 - acc: 0.9981
Epoch 10/10
521/521 [=====] - 11s - loss: 0.0311 - acc: 0.9981
58/58 [=====] - 0s
Epoch 1/10
521/521 [=====] - 13s - loss: 1.9028 - acc: 0.4722
Epoch 2/10
521/521 [=====] - 11s - loss: 0.2883 - acc: 0.9463
Epoch 3/10
521/521 [=====] - 11s - loss: 0.1044 - acc: 0.9770
Epoch 4/10
521/521 [=====] - 11s - loss: 0.0543 - acc: 0.9942
```

**Niklas Wilke** November 30, 2017 at 11:38 pm #

Hi Jason,  
could you please comment on this blog entry :  
<http://www.alfredo.motta.name/cross-validation-done-wrong/>

Sounds pretty logical to me and isnt that exactly what we are doing here ?  
If we ignore the feature selection part, we also split the data first and afterwards train the model ....

Thanks in advance

## Get Your Start in Machine Learning ×

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

START MY EMAIL COURSE

Get Your Start in Machine Learning



**Summer Cassidy** December 16, 2017 at 12:12 am #

REPLY ↩

Hello Jason Brownlee,

When I run the code I get an error. I have checked multiple times whether I have copied the code correctly. I am unable to trace why the error is occurring. Can you please help me out?

The error is:

Traceback (most recent call last):

File "F:/7th semester/machine language/thesis work/python/iris2.py", line 36, in

results = cross\_val\_score(estimator, X, dummy\_y, cv=kfold)

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\model\_selection\\_validation.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\model\_selection\\_validation.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in \_\_call\_\_

File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in \_\_call\_\_

## Get Your Start in Machine Learning

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

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

START MY EMAIL COURSE

Get Your Start in Machine Learning

```
File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\keras\wrappers\scikit_learn.py", line 147, in fit
history = self.model.fit(x, y, **fit_args)
File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\keras\models.py", line 960, in fit
validation_steps=validation_steps)
File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\keras\engine\training.py", line 1581, in fit
batch_size=batch_size)
File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\keras\engine\training.py", line 1418, in _standardize_user_data
exception_prefix='target')
File "C:\Users\ratul\AppData\Local\Programs\Python\Python35\lib\site-packages\keras\engine\training.py", line 153, 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 (90, 40)
```

**Jason Brownlee** December 16, 2017 at 5:32 am #

Looks like you might be using different data.

**Summer Cassidy** December 16, 2017 at 6:30 am #

Thanks for looking into the problem. I downloaded the iris flower dataset but from a different Learning repository solved my problem.

**Jason Brownlee** December 16, 2017 at 9:21 am #

Glad to hear it!

**Pubudu** January 1, 2018 at 4:52 pm #

## Get Your Start in Machine Learning

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

START MY EMAIL COURSE

REPLY ↩

Get Your Start in Machine Learning

Hey Jason:

Thanks for the tute. BTW, how do you planning to void dummy variable trap. You don't need all three types. Can you explain why you didn't use `train_test_split` method?

**Jason Brownlee** January 2, 2018 at 5:34 am #

REPLY ↩

The example uses k-fold cross validation instead of a train/test split.

The results are less biased with this method and I recommend it for smaller models.

## Leave a Reply

 Name (required) Email (will not be published) (required) Website

## Get Your Start in Machine Learning

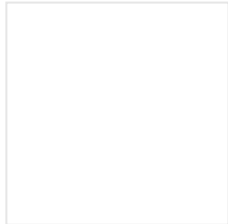


You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

[SUBMIT COMMENT](#)

## Welcome to Machine Learning Mastery



Hi, I'm Dr. Jason Brownlee.

My goal is to make practitioners like YOU awesome at applied machine learning.

[Read More](#)

### Finally Get Started With Deep Learning

Sick of the fancy math and need for super computers?

Looking for step-by-step tutorials?

Want end-to-end projects?

## Get Your Start in Machine Learning



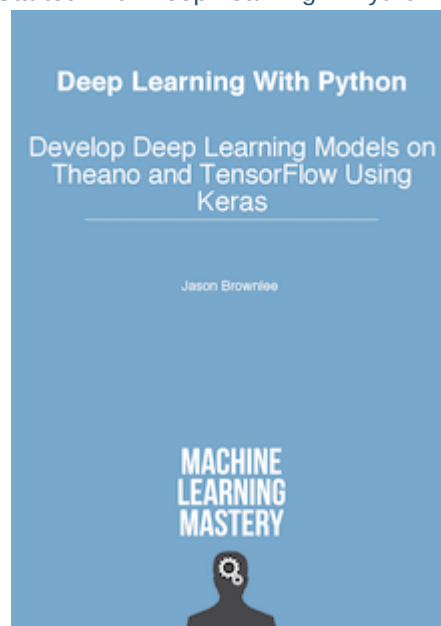
You can master applied Machine Learning **without the math or fancy degree.**

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

[START MY EMAIL COURSE](#)

[Get Your Start in Machine Learning](#)

Get Started With Deep Learning in Python Today!



## POPULAR

**Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras**

JULY 21, 2016

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

JUNE 10, 2016

**Develop Your First Neural Network in Python With Keras Step-By-Step**

MAY 24, 2016

**Multivariate Time Series Forecasting with LSTMs in Keras**

AUGUST 14, 2017

**How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda**

MARCH 13, 2017

## Get Your Start in Machine Learning



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

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

[START MY EMAIL COURSE](#)[Get Your Start in Machine Learning](#)



### Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras

JULY 26, 2016



### Time Series Forecasting with the Long Short-Term Memory Network in Python

APRIL 7, 2017



### Regression Tutorial with the Keras Deep Learning Library in Python

JUNE 9, 2016



### Multi-Class Classification Tutorial with the Keras Deep Learning Library

JUNE 2, 2016



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

AUGUST 9, 2016

## Get Your Start in Machine Learning ×

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

START MY EMAIL COURSE

© 2018 Machine Learning Mastery. All Rights Reserved.

[Privacy](#) | [Contact](#) | [About](#)

Get Your Start in Machine Learning