

# Assignment No 12 : TensorFlow and Keras: Build various MLP architectures for MNIST Dataset

## Resources :

- Thanks AAIC Team
- Google Search ,Kaggle,Sklearn
- KrushitReddy
- <https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/3407/exercise-try-different-mlp-architectures-on-mnist-dataset/8/module-8-neural-networks-computer-vision-and-deep-learning>
- <https://github.com/krushithreddy>
- <https://scikit-learn.org/stable/index.html>
- <https://www.kaggle.com/>

In [1]:

```
# Importing Libraries
```

```
import tensorflow as tf
from tensorflow.python.client import device_lib
print(device_lib.list_local_devices())
```

```
C:\Users\Sohail\Anaconda3\lib\site-packages\h5py\__init__.py:36: FutureWarning: Conversion of the
second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be
treated as `np.float64 == np.dtype(float).type`.
    from ._conv import register_converters as _register_converters
```

```
[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 1783339991702152395
]
```

In [2]:

```
from keras.utils import np_utils
from keras.datasets import mnist
from keras.initializers import RandomNormal
import seaborn as sns
```

```
Using TensorFlow backend.
```

In [4]:

```
import matplotlib.pyplot as plt
import numpy as np
import time
```

In [5]:

```
# this function is used to update the plots for each epoch and error
def plt_dynamic(x, vy, ty, ax, colors=['b']):
    ax.plot(x,vy,'b',label="Validation Loss")
    ax.plot(x,ty,'r',label="Train Loss")
    plt.legend()
    plt.grid()
    fig.canvas.draw()
```

In [6]:

```
(X train, y train), (X test, y test) = mnist.load data()
```

```
Downloading data from https://s3.amazonaws.com/img-datasets/mnist.npz
11493376/11490434 [=====] - 149s 13us/step
```

In [7]:

```
print("Number of training examples:",X_train.shape[0],"; each image is of shape (%d,%d)"%(X_train.  
shape[1],X_train.shape[2]))  
print("Number of training examples:",X_test.shape[0],"; each image is of shape  
(%d,%d)"%(X_test.shape[1],X_test.shape[2]))
```

Number of training examples: 60000 ; each image is of shape (28,28)  
Number of training examples: 10000 ; each image is of shape (28,28)

In [8]:

```
# if you observe the input shape its 3 dimensional vector
# for each image we have a (28*28) vector
# we will convert the (28*28) vector into single dimensional vector of 1 * 784

X_train = X_train.reshape(X_train.shape[0], X_train.shape[1]*X_train.shape[2])
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1]*X_test.shape[2])
```

In [9]:

```
# after converting the input images from 3d to 2d vectors

print("Number of training examples :", X_train.shape[0], "and each image is of shape (%d)%"(X_train.shape[1]))
print("Number of training examples :", X_test.shape[0], "and each image is of shape (%d)%"(X_test.shape[1]))
```

Number of training examples : 60000 and each image is of shape (784)  
 Number of training examples : 10000 and each image is of shape (784)

In [10]:

```
# if we observe the above matrix each cell is having a value between 0-255
# before we move to apply machine learning algorithms lets try to normalize the data
#  $X \Rightarrow (X - X_{min}) / (X_{max} - X_{min}) = X / 255$ 
# since  $X_{min} = 0$ 
X_train = X_train/255
X_test = X_test/255
```

In [11]:

```
print(X_train[0])
```

[illegible]

|            |            |            |            |            |            |
|------------|------------|------------|------------|------------|------------|
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.01176471 | 0.07058824 | 0.07058824 | 0.07058824 |
| 0.49411765 | 0.53333333 | 0.68627451 | 0.10196078 | 0.65098039 | 1.         |
| 0.96862745 | 0.49803922 | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.11764706 | 0.14117647 | 0.36862745 | 0.60392157 |
| 0.66666667 | 0.99215686 | 0.99215686 | 0.99215686 | 0.99215686 | 0.99215686 |
| 0.88235294 | 0.6745098  | 0.99215686 | 0.94901961 | 0.76470588 | 0.25098039 |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.19215686 |
| 0.93333333 | 0.99215686 | 0.99215686 | 0.99215686 | 0.99215686 | 0.99215686 |
| 0.99215686 | 0.99215686 | 0.99215686 | 0.98431373 | 0.36470588 | 0.32156863 |
| 0.32156863 | 0.21960784 | 0.15294118 | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.07058824 | 0.85882353 | 0.99215686 |
| 0.99215686 | 0.99215686 | 0.99215686 | 0.99215686 | 0.77647059 | 0.71372549 |
| 0.96862745 | 0.94509804 | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.31372549 | 0.61176471 | 0.41960784 | 0.99215686 |
| 0.99215686 | 0.80392157 | 0.04313725 | 0.         | 0.16862745 | 0.60392157 |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.05490196 | 0.00392157 | 0.60392157 | 0.99215686 | 0.35294118 |
| 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.         | 0.54509804 | 0.99215686 | 0.74509804 | 0.00784314 | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.04313725 |
| 0.74509804 | 0.99215686 | 0.2745098  | 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.         | 0.1372549  | 0.94509804 |
| 0.88235294 | 0.62745098 | 0.42352941 | 0.00392157 | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.31764706 | 0.94117647 | 0.99215686 |
| 0.99215686 | 0.46666667 | 0.09803922 | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.17647059 | 0.72941176 | 0.99215686 | 0.99215686 |
| 0.58823529 | 0.10588235 | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.0627451  | 0.36470588 | 0.98823529 | 0.99215686 | 0.73333333 |
| 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.         | 0.97647059 | 0.99215686 | 0.97647059 | 0.25098039 | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.18039216 | 0.50980392 | 0.71764706 | 0.99215686 |
| 0.99215686 | 0.81176471 | 0.00784314 | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.15294118 | 0.58039216 |
| 0.89803922 | 0.99215686 | 0.99215686 | 0.99215686 | 0.98039216 | 0.71372549 |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.09411765 | 0.44705882 | 0.86666667 | 0.99215686 | 0.99215686 | 0.99215686 |
| 0.99215686 | 0.78823529 | 0.30588235 | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |
| 0.         | 0.         | 0.         | 0.         | 0.         | 0.         |

In [12]:

# 1.MLP + Batch-Norm on hidden Layers + AdamOptimizer

In [15]:

```
from keras.layers.normalization import BatchNormalization

model_batch = Sequential()

model_batch.add(Dense(512, activation='relu', input_shape=(input_dim,), kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(256, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(output_dim, activation='softmax'))

model_batch.summary()
```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/op\_def\_library.py:263: colocate\_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version. Instructions for updating: Colocations handled automatically by placer.

| Layer (type)                                | Output Shape | Param # |
|---|--------------|---------|
| dense_1 (Dense)                             | (None, 512)  | 401920  |
| batch_normalization_1 (Batch Normalization) | (None, 512)  | 2048    |
| dense_2 (Dense)                             | (None, 256)  | 131328  |
| batch_normalization_2 (Batch Normalization) | (None, 256)  | 1024    |
| dense_3 (Dense)                             | (None, 10)   | 2570    |
| Total params: 538,890                       |              |         |
| Trainable params: 537,354                   |              |         |
| Non-trainable params: 1,536                 |              |         |

In [16]:

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

history = model_batch.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, validation_data=(X_test, Y_test))
```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/math\_ops.py:3066: to\_int32 (from tensorflow.python.ops.math\_ops) is deprecated and will be removed in a future version. Instructions for updating:

Use tf.cast instead.

Train on 60000 samples, validate on 10000 samples

Epoch 1/60

60000/60000 [=====] - 7s 114us/step - loss: 0.1799 - acc: 0.9454 - val\_loss: 0.1045 - val\_acc: 0.9657

Epoch 2/60

60000/60000 [=====] - 5s 90us/step - loss: 0.0731 - acc: 0.9780 - val\_loss: 0.0836 - val\_acc: 0.9746

Epoch 3/60

60000/60000 [=====] - 5s 91us/step - loss: 0.0486 - acc: 0.9852 - val\_loss: 0.0749 - val\_acc: 0.9762

Epoch 4/60

60000/60000 [=====] - 6s 92us/step - loss: 0.0352 - acc: 0.9889 - val\_loss: 0.0753 - val\_acc: 0.9784

Epoch 5/60

60000/60000 [=====] - 5s 91us/step - loss: 0.0274 - acc: 0.9914 - val\_loss: 0.0767 - val\_acc: 0.9761

Epoch 6/60

60000/60000 [=====] - 5s 91us/step - loss: 0.0230 - acc: 0.9928 -

```
val_loss: 0.0727 - val_acc: 0.9773
Epoch 7/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0209 - acc: 0.9931 -
val_loss: 0.0792 - val_acc: 0.9769
Epoch 8/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0151 - acc: 0.9951 -
val_loss: 0.0827 - val_acc: 0.9784
Epoch 9/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0160 - acc: 0.9949 -
val_loss: 0.0810 - val_acc: 0.9786
Epoch 10/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0145 - acc: 0.9950 -
val_loss: 0.0830 - val_acc: 0.9782
Epoch 11/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0127 - acc: 0.9960 -
val_loss: 0.0940 - val_acc: 0.9757
Epoch 12/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0127 - acc: 0.9958 -
val_loss: 0.0726 - val_acc: 0.9824
Epoch 13/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0102 - acc: 0.9967 -
val_loss: 0.0725 - val_acc: 0.9821
Epoch 14/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0089 - acc: 0.9973 -
val_loss: 0.0767 - val_acc: 0.9815
Epoch 15/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0091 - acc: 0.9970 -
val_loss: 0.0764 - val_acc: 0.9800
Epoch 16/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0101 - acc: 0.9965 -
val_loss: 0.0881 - val_acc: 0.9788
Epoch 17/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0096 - acc: 0.9969 -
val_loss: 0.0832 - val_acc: 0.9797
Epoch 18/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0090 - acc: 0.9971 -
val_loss: 0.0836 - val_acc: 0.9808
Epoch 19/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0061 - acc: 0.9981 -
val_loss: 0.0825 - val_acc: 0.9804
Epoch 20/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0071 - acc: 0.9978 -
val_loss: 0.1030 - val_acc: 0.9766
Epoch 21/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0072 - acc: 0.9975 -
val_loss: 0.0848 - val_acc: 0.9811
Epoch 22/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0069 - acc: 0.9979 -
val_loss: 0.0831 - val_acc: 0.9818
Epoch 23/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0051 - acc: 0.9985 -
val_loss: 0.0869 - val_acc: 0.9822
Epoch 24/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0049 - acc: 0.9987 -
val_loss: 0.0840 - val_acc: 0.9822
Epoch 25/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0059 - acc: 0.9980 -
val_loss: 0.0876 - val_acc: 0.9802
Epoch 26/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0066 - acc: 0.9978 -
val_loss: 0.0867 - val_acc: 0.9805
Epoch 27/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0052 - acc: 0.9983 -
val_loss: 0.0804 - val_acc: 0.9822
Epoch 28/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0041 - acc: 0.9986 -
val_loss: 0.0836 - val_acc: 0.9828
Epoch 29/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0066 - acc: 0.9976 -
val_loss: 0.0960 - val_acc: 0.9807
Epoch 30/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0051 - acc: 0.9984 -
val_loss: 0.0912 - val_acc: 0.9804
Epoch 31/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0048 - acc: 0.9984 -
val_loss: 0.0843 - val_acc: 0.9833
Epoch 32/60
```

```
-
60000/60000 [=====] - 6s 92us/step - loss: 0.0056 - acc: 0.9981 -
val_loss: 0.0936 - val_acc: 0.9819
Epoch 33/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0048 - acc: 0.9985 -
val_loss: 0.0823 - val_acc: 0.9819
Epoch 34/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0039 - acc: 0.9988 -
val_loss: 0.0846 - val_acc: 0.9827
Epoch 35/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0033 - acc: 0.9989 -
val_loss: 0.0862 - val_acc: 0.9841
Epoch 36/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0039 - acc: 0.9987 -
val_loss: 0.0961 - val_acc: 0.9824
Epoch 37/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0052 - acc: 0.9983 -
val_loss: 0.0974 - val_acc: 0.9820
Epoch 38/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0039 - acc: 0.9987 -
val_loss: 0.0865 - val_acc: 0.9832
Epoch 39/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0040 - acc: 0.9986 -
val_loss: 0.0872 - val_acc: 0.9825
Epoch 40/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0052 - acc: 0.9984 -
val_loss: 0.0917 - val_acc: 0.9831
Epoch 41/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0040 - acc: 0.9987 -
val_loss: 0.0878 - val_acc: 0.9832
Epoch 42/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0036 - acc: 0.9988 -
val_loss: 0.0809 - val_acc: 0.9843
Epoch 43/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0041 - acc: 0.9988 -
val_loss: 0.0782 - val_acc: 0.9842
Epoch 44/60
60000/60000 [=====] - 5s 92us/step - loss: 0.0034 - acc: 0.9989 -
val_loss: 0.0896 - val_acc: 0.9834
Epoch 45/60
60000/60000 [=====] - 5s 92us/step - loss: 0.0028 - acc: 0.9992 -
val_loss: 0.0973 - val_acc: 0.9819
Epoch 46/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0032 - acc: 0.9990 -
val_loss: 0.0880 - val_acc: 0.9827
Epoch 47/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0041 - acc: 0.9987 -
val_loss: 0.0859 - val_acc: 0.9817
Epoch 48/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0040 - acc: 0.9986 -
val_loss: 0.0873 - val_acc: 0.9818
Epoch 49/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0040 - acc: 0.9985 -
val_loss: 0.0904 - val_acc: 0.9824
Epoch 50/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0041 - acc: 0.9986 -
val_loss: 0.1009 - val_acc: 0.9813
Epoch 51/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0025 - acc: 0.9992 -
val_loss: 0.0899 - val_acc: 0.9825
Epoch 52/60
60000/60000 [=====] - 5s 92us/step - loss: 0.0019 - acc: 0.9994 -
val_loss: 0.0843 - val_acc: 0.9837
Epoch 53/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0020 - acc: 0.9994 -
val_loss: 0.0881 - val_acc: 0.9835
Epoch 54/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0033 - acc: 0.9989 -
val_loss: 0.0901 - val_acc: 0.9838
Epoch 55/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0036 - acc: 0.9988 -
val_loss: 0.1018 - val_acc: 0.9805
Epoch 56/60
60000/60000 [=====] - 5s 90us/step - loss: 0.0033 - acc: 0.9991 -
val_loss: 0.0835 - val_acc: 0.9841
Epoch 57/60
60000/60000 [=====] - 5s 92us/step - loss: 0.0017 - acc: 0.9993 -
val_loss: 0.0866 - val_acc: 0.9846
```

```
Epoch 58/60
60000/60000 [=====] - 6s 92us/step - loss: 0.0030 - acc: 0.9989 -
val_loss: 0.0944 - val_acc: 0.9830
Epoch 59/60
60000/60000 [=====] - 5s 91us/step - loss: 0.0032 - acc: 0.9991 -
val_loss: 0.0851 - val_acc: 0.9830
Epoch 60/60
60000/60000 [=====] - 6s 93us/step - loss: 0.0025 - acc: 0.9992 -
val_loss: 0.0919 - val_acc: 0.9830
```

In [17]:

```
score = model_batch.evaluate(X_test, Y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

fig, ax = plt.subplots(1, 1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# list of epoch numbers
x = list(range(1, nb_epoch+1))

# print(history.history.keys())
# dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
# history = model_drop.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, va
lvalidation_data=(X_test, Y_test))

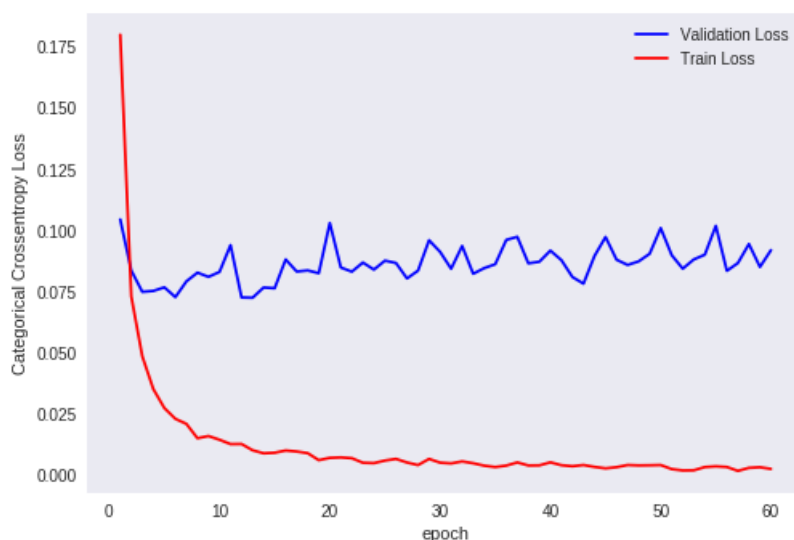
# we will get val_loss and val_acc only when you pass the paramter validation_data
# val_loss : validation loss
# val_acc : validation accuracy

# loss : training loss
# acc : train accuracy
# for each key in history.history we will have a list of length equal to number of epochs

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax)
```

Test score: 0.09192510834604554

Test accuracy: 0.983



## 2. MLP + Dropout + AdamOptimizer

In [18]:

```
from keras.layers import Dropout

model_drop = Sequential()

model_drop.add(Dense(512, activation='relu', input_shape=(input_dim,), kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))
```



```

model_drop.add(BatchNormalization())
model_drop.add(Dropout(0.5))

model_drop.add(Dense(256, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55,
seed=None)) )
model_drop.add(BatchNormalization())
model_drop.add(Dropout(0.5))

model_drop.add(Dense(output_dim, activation='softmax'))

model_drop.summary()

```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:3445: calling dropout (from tensorflow.python.ops.nn\_ops) with keep\_prob is deprecated and will be removed in a future version.  
Instructions for updating:  
Please use `rate` instead of `keep\_prob`. Rate should be set to `rate = 1 - keep\_prob`.

| Layer (type)                                | Output Shape | Param # |
|---|--------------|---------|
| dense_4 (Dense)                             | (None, 512)  | 401920  |
| batch_normalization_3 (Batch Normalization) | (None, 512)  | 2048    |
| dropout_1 (Dropout)                         | (None, 512)  | 0       |
| dense_5 (Dense)                             | (None, 256)  | 131328  |
| batch_normalization_4 (Batch Normalization) | (None, 256)  | 1024    |
| dropout_2 (Dropout)                         | (None, 256)  | 0       |
| dense_6 (Dense)                             | (None, 10)   | 2570    |
| Total params: 538,890                       |              |         |
| Trainable params: 537,354                   |              |         |
| Non-trainable params: 1,536                 |              |         |

In [19]:

```

model_drop.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

history = model_drop.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, validation_data=(X_test, Y_test))

```

Train on 60000 samples, validate on 10000 samples

Epoch 1/60  
60000/60000 [=====] - 6s 106us/step - loss: 0.4528 - acc: 0.8622 - val\_loss: 0.1598 - val\_acc: 0.9525

Epoch 2/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.2362 - acc: 0.9277 - val\_loss: 0.1314 - val\_acc: 0.9566

Epoch 3/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.1886 - acc: 0.9425 - val\_loss: 0.1041 - val\_acc: 0.9666

Epoch 4/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.1674 - acc: 0.9488 - val\_loss: 0.0946 - val\_acc: 0.9697

Epoch 5/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.1496 - acc: 0.9544 - val\_loss: 0.0869 - val\_acc: 0.9725

Epoch 6/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.1377 - acc: 0.9582 - val\_loss: 0.0780 - val\_acc: 0.9757

Epoch 7/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.1264 - acc: 0.9608 - val\_loss: 0.0774 - val\_acc: 0.9756

Epoch 8/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.1155 - acc: 0.9639 - val\_loss: 0.0738 - val\_acc: 0.9773

Epoch 9/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.1102 - acc: 0.9665 -

```
00000/00000 [=====] - 6s 94us/step - loss: 0.1057 - acc: 0.9671 -  
val_loss: 0.0678 - val_acc: 0.9790  
Epoch 10/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.1057 - acc: 0.9671 -  
val_loss: 0.0697 - val_acc: 0.9782  
Epoch 11/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0981 - acc: 0.9696 -  
val_loss: 0.0650 - val_acc: 0.9803  
Epoch 12/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0938 - acc: 0.9703 -  
val_loss: 0.0655 - val_acc: 0.9788  
Epoch 13/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0892 - acc: 0.9722 -  
val_loss: 0.0639 - val_acc: 0.9808  
Epoch 14/60  
60000/60000 [=====] - 6s 93us/step - loss: 0.0837 - acc: 0.9740 -  
val_loss: 0.0607 - val_acc: 0.9815  
Epoch 15/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0862 - acc: 0.9732 -  
val_loss: 0.0595 - val_acc: 0.9818  
Epoch 16/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0807 - acc: 0.9743 -  
val_loss: 0.0621 - val_acc: 0.9823  
Epoch 17/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0758 - acc: 0.9757 -  
val_loss: 0.0637 - val_acc: 0.9812  
Epoch 18/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.0729 - acc: 0.9770 -  
val_loss: 0.0573 - val_acc: 0.9832  
Epoch 19/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0711 - acc: 0.9783 -  
val_loss: 0.0594 - val_acc: 0.9827  
Epoch 20/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.0654 - acc: 0.9790 -  
val_loss: 0.0590 - val_acc: 0.9837  
Epoch 21/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.0666 - acc: 0.9787 -  
val_loss: 0.0568 - val_acc: 0.9839  
Epoch 22/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0643 - acc: 0.9794 -  
val_loss: 0.0607 - val_acc: 0.9828  
Epoch 23/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.0581 - acc: 0.9813 -  
val_loss: 0.0546 - val_acc: 0.9848  
Epoch 24/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.0586 - acc: 0.9814 -  
val_loss: 0.0568 - val_acc: 0.9840  
Epoch 25/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.0577 - acc: 0.9814 -  
val_loss: 0.0598 - val_acc: 0.9827  
Epoch 26/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0571 - acc: 0.9812 -  
val_loss: 0.0544 - val_acc: 0.9841  
Epoch 27/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0547 - acc: 0.9823 -  
val_loss: 0.0537 - val_acc: 0.9838  
Epoch 28/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0518 - acc: 0.9831 -  
val_loss: 0.0554 - val_acc: 0.9850  
Epoch 29/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0487 - acc: 0.9844 -  
val_loss: 0.0588 - val_acc: 0.9839  
Epoch 30/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0513 - acc: 0.9831 -  
val_loss: 0.0544 - val_acc: 0.9846  
Epoch 31/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.0499 - acc: 0.9840 -  
val_loss: 0.0562 - val_acc: 0.9849  
Epoch 32/60  
60000/60000 [=====] - 6s 94us/step - loss: 0.0469 - acc: 0.9850 -  
val_loss: 0.0565 - val_acc: 0.9839  
Epoch 33/60  
60000/60000 [=====] - 6s 95us/step - loss: 0.0462 - acc: 0.9849 -  
val_loss: 0.0557 - val_acc: 0.9855  
Epoch 34/60  
60000/60000 [=====] - 6s 93us/step - loss: 0.0463 - acc: 0.9845 -  
val_loss: 0.0555 - val_acc: 0.9856  
Epoch 35/60
```

```
Epoch 35/60
60000/60000 [=====] - 6s 95us/step - loss: 0.0440 - acc: 0.9854 -
val_loss: 0.0533 - val_acc: 0.9853
Epoch 36/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0447 - acc: 0.9856 -
val_loss: 0.0532 - val_acc: 0.9845
Epoch 37/60
60000/60000 [=====] - 6s 95us/step - loss: 0.0417 - acc: 0.9865 -
val_loss: 0.0596 - val_acc: 0.9839
Epoch 38/60
60000/60000 [=====] - 6s 95us/step - loss: 0.0414 - acc: 0.9868 -
val_loss: 0.0572 - val_acc: 0.9849
Epoch 39/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0407 - acc: 0.9869 -
val_loss: 0.0597 - val_acc: 0.9840
Epoch 40/60
60000/60000 [=====] - 6s 95us/step - loss: 0.0405 - acc: 0.9867 -
val_loss: 0.0555 - val_acc: 0.9850
Epoch 41/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0401 - acc: 0.9867 -
val_loss: 0.0516 - val_acc: 0.9871
Epoch 42/60
60000/60000 [=====] - 6s 93us/step - loss: 0.0396 - acc: 0.9869 -
val_loss: 0.0538 - val_acc: 0.9849
Epoch 43/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0363 - acc: 0.9880 -
val_loss: 0.0543 - val_acc: 0.9851
Epoch 44/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0357 - acc: 0.9881 -
val_loss: 0.0581 - val_acc: 0.9845
Epoch 45/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0337 - acc: 0.9888 -
val_loss: 0.0612 - val_acc: 0.9836
Epoch 46/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0348 - acc: 0.9884 -
val_loss: 0.0520 - val_acc: 0.9855
Epoch 47/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0360 - acc: 0.9882 -
val_loss: 0.0561 - val_acc: 0.9851
Epoch 48/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0355 - acc: 0.9883 -
val_loss: 0.0537 - val_acc: 0.9862
Epoch 49/60
60000/60000 [=====] - 6s 95us/step - loss: 0.0335 - acc: 0.9890 -
val_loss: 0.0583 - val_acc: 0.9847
Epoch 50/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0328 - acc: 0.9888 -
val_loss: 0.0595 - val_acc: 0.9850
Epoch 51/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0341 - acc: 0.9893 -
val_loss: 0.0568 - val_acc: 0.9856
Epoch 52/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0341 - acc: 0.9884 -
val_loss: 0.0559 - val_acc: 0.9847
Epoch 53/60
60000/60000 [=====] - 6s 95us/step - loss: 0.0338 - acc: 0.9890 -
val_loss: 0.0540 - val_acc: 0.9857
Epoch 54/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0318 - acc: 0.9897 -
val_loss: 0.0533 - val_acc: 0.9857
Epoch 55/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0307 - acc: 0.9898 -
val_loss: 0.0566 - val_acc: 0.9841
Epoch 56/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0292 - acc: 0.9906 -
val_loss: 0.0551 - val_acc: 0.9851
Epoch 57/60
60000/60000 [=====] - 6s 95us/step - loss: 0.0285 - acc: 0.9905 -
val_loss: 0.0543 - val_acc: 0.9847
Epoch 58/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0295 - acc: 0.9900 -
val_loss: 0.0546 - val_acc: 0.9857
Epoch 59/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0294 - acc: 0.9902 -
val_loss: 0.0559 - val_acc: 0.9863
Epoch 60/60
60000/60000 [=====] - 6s 94us/step - loss: 0.0259 - acc: 0.9912 -
val_loss: 0.0567 - val_acc: 0.9855
```

val\_loss: 0.0567    val\_acc: 0.9855

In [20]:

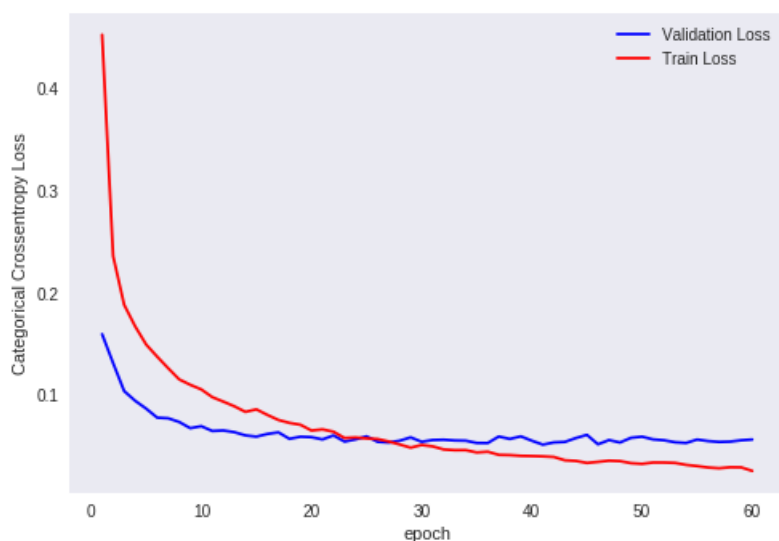
```
score = model_drop.evaluate(X_test, Y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax)
```

Test score: 0.05668292018135808

Test accuracy: 0.9855



## Architecture-2: input (784) - ReLu(1024) - ReLu(512)-Relu(256)-sigmoid

### 1.MLP + Batch-Norm on hidden Layers + AdamOptimizer

In [22]:

```
model_batch = Sequential()

model_batch.add(Dense(1024, activation='relu', input_shape=(input_dim,), kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(512, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(256, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(output_dim, activation='softmax'))

model_batch.summary()
```

| Layer (type)                                | Output Shape | Param # |
|---|--------------|---------|
| dense_7 (Dense)                             | (None, 1024) | 803840  |
| batch_normalization_5 (Batch Normalization) | (None, 1024) | 4096    |

|   |             |        |
|---|-------------|--------|
| batch_normalization_6 (Batch Normalization) | (None, 512) | 1024   |
| dense_8 (Dense)                             | (None, 512) | 524800 |
| batch_normalization_6 (Batch Normalization) | (None, 512) | 2048   |
| dense_9 (Dense)                             | (None, 256) | 131328 |
| batch_normalization_7 (Batch Normalization) | (None, 256) | 1024   |
| dense_10 (Dense)                            | (None, 10)  | 2570   |
| =====                                       |             |        |
| Total params: 1,469,706                     |             |        |
| Trainable params: 1,466,122                 |             |        |
| Non-trainable params: 3,584                 |             |        |

In [23]:

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

history = model_batch.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, validation_data=(X_test, Y_test))
```

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/60
60000/60000 [=====] - 9s 154us/step - loss: 0.1723 - acc: 0.9483 -
val_loss: 0.1034 - val_acc: 0.9691
Epoch 2/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0668 - acc: 0.9799 -
val_loss: 0.0872 - val_acc: 0.9740
Epoch 3/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0433 - acc: 0.9868 -
val_loss: 0.0809 - val_acc: 0.9742
Epoch 4/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0344 - acc: 0.9887 -
val_loss: 0.0768 - val_acc: 0.9786
Epoch 5/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0257 - acc: 0.9920 -
val_loss: 0.0860 - val_acc: 0.9752
Epoch 6/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0228 - acc: 0.9925 -
val_loss: 0.0829 - val_acc: 0.9747
Epoch 7/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0195 - acc: 0.9933 -
val_loss: 0.0635 - val_acc: 0.9815
Epoch 8/60
60000/60000 [=====] - 8s 135us/step - loss: 0.0204 - acc: 0.9930 -
val_loss: 0.0769 - val_acc: 0.9782
Epoch 9/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0176 - acc: 0.9939 -
val_loss: 0.0711 - val_acc: 0.9805
Epoch 10/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0141 - acc: 0.9951 -
val_loss: 0.0884 - val_acc: 0.9756
Epoch 11/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0145 - acc: 0.9951 -
val_loss: 0.0785 - val_acc: 0.9792
Epoch 12/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0128 - acc: 0.9955 -
val_loss: 0.0659 - val_acc: 0.9825
Epoch 13/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0132 - acc: 0.9956 -
val_loss: 0.0788 - val_acc: 0.9805
Epoch 14/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0111 - acc: 0.9965 -
val_loss: 0.0804 - val_acc: 0.9811
Epoch 15/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0113 - acc: 0.9965 -
val_loss: 0.0796 - val_acc: 0.9796
Epoch 16/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0119 - acc: 0.9962 -
val_loss: 0.0711 - val_acc: 0.9819
Epoch 17/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0097 - acc: 0.9967 -
val_loss: 0.0771 - val_acc: 0.9803
```

```
Epoch 18/60
60000/60000 [=====] - 8s 134us/step - loss: 0.0085 - acc: 0.9972 -
val_loss: 0.0653 - val_acc: 0.9830
Epoch 19/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0070 - acc: 0.9976 -
val_loss: 0.0752 - val_acc: 0.9823
Epoch 20/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0092 - acc: 0.9971 -
val_loss: 0.0811 - val_acc: 0.9818
Epoch 21/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0083 - acc: 0.9973 -
val_loss: 0.0860 - val_acc: 0.9800
Epoch 22/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0066 - acc: 0.9979 -
val_loss: 0.0853 - val_acc: 0.9785
Epoch 23/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0076 - acc: 0.9972 -
val_loss: 0.0761 - val_acc: 0.9813
Epoch 24/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0078 - acc: 0.9976 -
val_loss: 0.0874 - val_acc: 0.9812
Epoch 25/60
60000/60000 [=====] - 8s 135us/step - loss: 0.0061 - acc: 0.9981 -
val_loss: 0.0832 - val_acc: 0.9810
Epoch 26/60
60000/60000 [=====] - 8s 138us/step - loss: 0.0066 - acc: 0.9979 -
val_loss: 0.0859 - val_acc: 0.9818
Epoch 27/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0049 - acc: 0.9984 -
val_loss: 0.0885 - val_acc: 0.9812
Epoch 28/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0061 - acc: 0.9981 -
val_loss: 0.0774 - val_acc: 0.9825
Epoch 29/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0062 - acc: 0.9979 -
val_loss: 0.0813 - val_acc: 0.9826
Epoch 30/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0041 - acc: 0.9987 -
val_loss: 0.0800 - val_acc: 0.9827
Epoch 31/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0041 - acc: 0.9987 -
val_loss: 0.0757 - val_acc: 0.9832
Epoch 32/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0060 - acc: 0.9981 -
val_loss: 0.0827 - val_acc: 0.9811
Epoch 33/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0073 - acc: 0.9978 -
val_loss: 0.0818 - val_acc: 0.9823
Epoch 34/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0053 - acc: 0.9985 -
val_loss: 0.0774 - val_acc: 0.9847
Epoch 35/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0037 - acc: 0.9987 -
val_loss: 0.0880 - val_acc: 0.9824
Epoch 36/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0036 - acc: 0.9990 -
val_loss: 0.0791 - val_acc: 0.9842
Epoch 37/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0051 - acc: 0.9986 -
val_loss: 0.0821 - val_acc: 0.9824
Epoch 38/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0060 - acc: 0.9980 -
val_loss: 0.0833 - val_acc: 0.9820
Epoch 39/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0038 - acc: 0.9987 -
val_loss: 0.0761 - val_acc: 0.9842
Epoch 40/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0027 - acc: 0.9991 -
val_loss: 0.0931 - val_acc: 0.9815
Epoch 41/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0051 - acc: 0.9983 -
val_loss: 0.0788 - val_acc: 0.9843
Epoch 42/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0049 - acc: 0.9983 -
val_loss: 0.0789 - val_acc: 0.9852
Epoch 43/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0049 - acc: 0.9986 -
```

```

val_loss: 0.0837 - val_acc: 0.9830
Epoch 44/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0037 - acc: 0.9988 -
val_loss: 0.0732 - val_acc: 0.9853
Epoch 45/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0034 - acc: 0.9989 -
val_loss: 0.0799 - val_acc: 0.9839
Epoch 46/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0047 - acc: 0.9987 -
val_loss: 0.0881 - val_acc: 0.9840
Epoch 47/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0038 - acc: 0.9987 -
val_loss: 0.0899 - val_acc: 0.9837
Epoch 48/60
60000/60000 [=====] - 8s 138us/step - loss: 0.0035 - acc: 0.9988 -
val_loss: 0.0850 - val_acc: 0.9834
Epoch 49/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0027 - acc: 0.9991 -
val_loss: 0.0801 - val_acc: 0.9849
Epoch 50/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0030 - acc: 0.9990 -
val_loss: 0.0803 - val_acc: 0.9845
Epoch 51/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0042 - acc: 0.9986 -
val_loss: 0.0747 - val_acc: 0.9845
Epoch 52/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0044 - acc: 0.9986 -
val_loss: 0.0836 - val_acc: 0.9827
Epoch 53/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0052 - acc: 0.9985 -
val_loss: 0.0794 - val_acc: 0.9845
Epoch 54/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0030 - acc: 0.9990 -
val_loss: 0.0834 - val_acc: 0.9843
Epoch 55/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0023 - acc: 0.9992 -
val_loss: 0.0895 - val_acc: 0.9843
Epoch 56/60
60000/60000 [=====] - 8s 135us/step - loss: 0.0027 - acc: 0.9990 -
val_loss: 0.0872 - val_acc: 0.9842
Epoch 57/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0027 - acc: 0.9990 -
val_loss: 0.0873 - val_acc: 0.9846
Epoch 58/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0032 - acc: 0.9990 -
val_loss: 0.0937 - val_acc: 0.9844
Epoch 59/60
60000/60000 [=====] - 8s 137us/step - loss: 0.0036 - acc: 0.9988 -
val_loss: 0.0850 - val_acc: 0.9845
Epoch 60/60
60000/60000 [=====] - 8s 136us/step - loss: 0.0023 - acc: 0.9992 -
val_loss: 0.0934 - val_acc: 0.9826

```

In [24]:

```

score = model_drop.evaluate(X_test, Y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

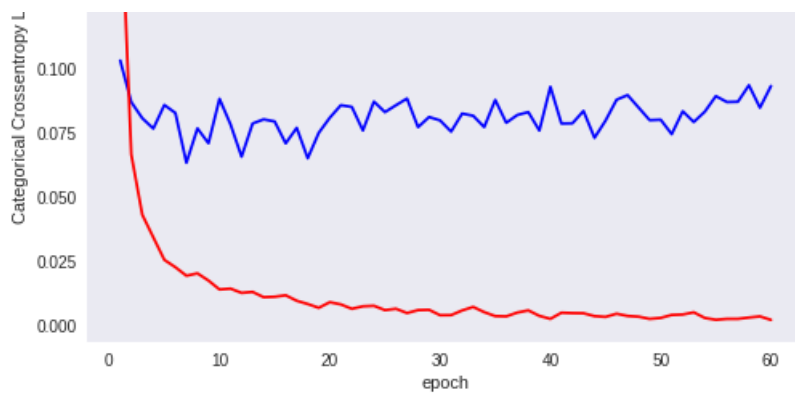
fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax)

```

Test score: 0.05668292018135808  
Test accuracy: 0.9855





## 2. MLP + Dropout + AdamOptimizer

In [25]:

```
model_drop = Sequential()

model_drop.add(Dense(1024, activation='relu', input_shape=(input_dim,), kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))
model_drop.add(BatchNormalization())
model_drop.add(Dropout(0.5))

model_drop.add(Dense(512, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_drop.add(BatchNormalization())
model_drop.add(Dropout(0.5))

model_drop.add(Dense(256, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_drop.add(BatchNormalization())
model_drop.add(Dropout(0.5))

model_drop.add(Dense(output_dim, activation='softmax'))

model_drop.summary()
```

| Layer (type)                                 | Output Shape | Param # |
|--|--------------|---------|
| dense_11 (Dense)                             | (None, 1024) | 803840  |
| batch_normalization_8 (Batch Normalization)  | (None, 1024) | 4096    |
| dropout_3 (Dropout)                          | (None, 1024) | 0       |
| dense_12 (Dense)                             | (None, 512)  | 524800  |
| batch_normalization_9 (Batch Normalization)  | (None, 512)  | 2048    |
| dropout_4 (Dropout)                          | (None, 512)  | 0       |
| dense_13 (Dense)                             | (None, 256)  | 131328  |
| batch_normalization_10 (Batch Normalization) | (None, 256)  | 1024    |
| dropout_5 (Dropout)                          | (None, 256)  | 0       |
| dense_14 (Dense)                             | (None, 10)   | 2570    |
| Total params: 1,469,706                      |              |         |
| Trainable params: 1,466,122                  |              |         |
| Non-trainable params: 3,584                  |              |         |

In [26]:

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

history = model_drop.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, validation_data=(X_val, Y_val))
```



```
history = model_dropout.fit(X_train, y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, validation_data=(X_test, Y_test))
```

Train on 60000 samples, validate on 10000 samples

Epoch 1/60

60000/60000 [=====] - 10s 164us/step - loss: 0.5917 - acc: 0.8182 - val\_loss: 0.1716 - val\_acc: 0.9440

Epoch 2/60

60000/60000 [=====] - 8s 141us/step - loss: 0.2722 - acc: 0.9175 - val\_loss: 0.1379 - val\_acc: 0.9550

Epoch 3/60

60000/60000 [=====] - 8s 140us/step - loss: 0.2157 - acc: 0.9360 - val\_loss: 0.1088 - val\_acc: 0.9644

Epoch 4/60

60000/60000 [=====] - 8s 140us/step - loss: 0.1818 - acc: 0.9452 - val\_loss: 0.1062 - val\_acc: 0.9663

Epoch 5/60

60000/60000 [=====] - 8s 139us/step - loss: 0.1587 - acc: 0.9517 - val\_loss: 0.0955 - val\_acc: 0.9708

Epoch 6/60

60000/60000 [=====] - 8s 141us/step - loss: 0.1448 - acc: 0.9564 - val\_loss: 0.0857 - val\_acc: 0.9725

Epoch 7/60

60000/60000 [=====] - 8s 140us/step - loss: 0.1328 - acc: 0.9600 - val\_loss: 0.0751 - val\_acc: 0.9782

Epoch 8/60

60000/60000 [=====] - 8s 141us/step - loss: 0.1251 - acc: 0.9610 - val\_loss: 0.0744 - val\_acc: 0.9776

Epoch 9/60

60000/60000 [=====] - 8s 141us/step - loss: 0.1139 - acc: 0.9641 - val\_loss: 0.0705 - val\_acc: 0.9794

Epoch 10/60

60000/60000 [=====] - 8s 141us/step - loss: 0.1058 - acc: 0.9681 - val\_loss: 0.0713 - val\_acc: 0.9777

Epoch 11/60

60000/60000 [=====] - 9s 143us/step - loss: 0.1018 - acc: 0.9686 - val\_loss: 0.0687 - val\_acc: 0.9798

Epoch 12/60

60000/60000 [=====] - 8s 141us/step - loss: 0.0941 - acc: 0.9704 - val\_loss: 0.0660 - val\_acc: 0.9820

Epoch 13/60

60000/60000 [=====] - 8s 142us/step - loss: 0.0904 - acc: 0.9720 - val\_loss: 0.0642 - val\_acc: 0.9812

Epoch 14/60

60000/60000 [=====] - 8s 140us/step - loss: 0.0848 - acc: 0.9738 - val\_loss: 0.0603 - val\_acc: 0.9823

Epoch 15/60

60000/60000 [=====] - 8s 140us/step - loss: 0.0833 - acc: 0.9738 - val\_loss: 0.0621 - val\_acc: 0.9822

Epoch 16/60

60000/60000 [=====] - 8s 141us/step - loss: 0.0790 - acc: 0.9755 - val\_loss: 0.0608 - val\_acc: 0.9826

Epoch 17/60

60000/60000 [=====] - 8s 141us/step - loss: 0.0748 - acc: 0.9764 - val\_loss: 0.0638 - val\_acc: 0.9823

Epoch 18/60

60000/60000 [=====] - 8s 140us/step - loss: 0.0730 - acc: 0.9775 - val\_loss: 0.0610 - val\_acc: 0.9821

Epoch 19/60

60000/60000 [=====] - 8s 139us/step - loss: 0.0672 - acc: 0.9793 - val\_loss: 0.0593 - val\_acc: 0.9835

Epoch 20/60

60000/60000 [=====] - 8s 140us/step - loss: 0.0632 - acc: 0.9803 - val\_loss: 0.0568 - val\_acc: 0.9836

Epoch 21/60

60000/60000 [=====] - 8s 141us/step - loss: 0.0619 - acc: 0.9801 - val\_loss: 0.0596 - val\_acc: 0.9841

Epoch 22/60

60000/60000 [=====] - 8s 140us/step - loss: 0.0636 - acc: 0.9798 - val\_loss: 0.0578 - val\_acc: 0.9840

Epoch 23/60

60000/60000 [=====] - 8s 141us/step - loss: 0.0573 - acc: 0.9814 - val\_loss: 0.0567 - val\_acc: 0.9842

Epoch 24/60

60000/60000 [=====] - 8s 141us/step - loss: 0.0570 - acc: 0.9819 - val\_loss: 0.0525 - val\_acc: 0.9859

Epoch 25/60

```
-
60000/60000 [=====] - 9s 144us/step - loss: 0.0548 - acc: 0.9823 -
val_loss: 0.0530 - val_acc: 0.9851
Epoch 26/60
60000/60000 [=====] - 9s 144us/step - loss: 0.0532 - acc: 0.9829 -
val_loss: 0.0567 - val_acc: 0.9843
Epoch 27/60
60000/60000 [=====] - 9s 146us/step - loss: 0.0533 - acc: 0.9832 -
val_loss: 0.0508 - val_acc: 0.9859
Epoch 28/60
60000/60000 [=====] - 9s 148us/step - loss: 0.0481 - acc: 0.9843 -
val_loss: 0.0573 - val_acc: 0.9850
Epoch 29/60
60000/60000 [=====] - 9s 144us/step - loss: 0.0476 - acc: 0.9847 -
val_loss: 0.0548 - val_acc: 0.9852
Epoch 30/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0469 - acc: 0.9852 -
val_loss: 0.0530 - val_acc: 0.9861
Epoch 31/60
60000/60000 [=====] - 9s 146us/step - loss: 0.0443 - acc: 0.9857 -
val_loss: 0.0510 - val_acc: 0.9855
Epoch 32/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0426 - acc: 0.9860 -
val_loss: 0.0542 - val_acc: 0.9849
Epoch 33/60
60000/60000 [=====] - 9s 147us/step - loss: 0.0423 - acc: 0.9863 -
val_loss: 0.0531 - val_acc: 0.9860
Epoch 34/60
60000/60000 [=====] - 9s 147us/step - loss: 0.0409 - acc: 0.9865 -
val_loss: 0.0507 - val_acc: 0.9862
Epoch 35/60
60000/60000 [=====] - 9s 148us/step - loss: 0.0436 - acc: 0.9858 -
val_loss: 0.0534 - val_acc: 0.9853
Epoch 36/60
60000/60000 [=====] - 9s 143us/step - loss: 0.0416 - acc: 0.9867 -
val_loss: 0.0550 - val_acc: 0.9850
Epoch 37/60
60000/60000 [=====] - 9s 146us/step - loss: 0.0401 - acc: 0.9876 -
val_loss: 0.0557 - val_acc: 0.9846
Epoch 38/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0375 - acc: 0.9881 -
val_loss: 0.0516 - val_acc: 0.9864
Epoch 39/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0371 - acc: 0.9877 -
val_loss: 0.0540 - val_acc: 0.9849
Epoch 40/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0359 - acc: 0.9884 -
val_loss: 0.0537 - val_acc: 0.9860
Epoch 41/60
60000/60000 [=====] - 9s 146us/step - loss: 0.0336 - acc: 0.9890 -
val_loss: 0.0548 - val_acc: 0.9863
Epoch 42/60
60000/60000 [=====] - 9s 146us/step - loss: 0.0331 - acc: 0.9892 -
val_loss: 0.0625 - val_acc: 0.9851
Epoch 43/60
60000/60000 [=====] - 9s 146us/step - loss: 0.0356 - acc: 0.9885 -
val_loss: 0.0583 - val_acc: 0.9848
Epoch 44/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0340 - acc: 0.9891 -
val_loss: 0.0557 - val_acc: 0.9858
Epoch 45/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0327 - acc: 0.9892 -
val_loss: 0.0516 - val_acc: 0.9860
Epoch 46/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0301 - acc: 0.9900 -
val_loss: 0.0552 - val_acc: 0.9860
Epoch 47/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0331 - acc: 0.9898 -
val_loss: 0.0548 - val_acc: 0.9854
Epoch 48/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0302 - acc: 0.9901 -
val_loss: 0.0546 - val_acc: 0.9861
Epoch 49/60
60000/60000 [=====] - 9s 144us/step - loss: 0.0318 - acc: 0.9899 -
val_loss: 0.0562 - val_acc: 0.9863
Epoch 50/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0292 - acc: 0.9905 -
val_loss: 0.0580 - val_acc: 0.9852
```

```

Epoch 51/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0286 - acc: 0.9906 -
val_loss: 0.0596 - val_acc: 0.9863
Epoch 52/60
60000/60000 [=====] - 9s 142us/step - loss: 0.0285 - acc: 0.9898 -
val_loss: 0.0592 - val_acc: 0.9860
Epoch 53/60
60000/60000 [=====] - 8s 141us/step - loss: 0.0272 - acc: 0.9911 -
val_loss: 0.0548 - val_acc: 0.9865
Epoch 54/60
60000/60000 [=====] - 8s 141us/step - loss: 0.0296 - acc: 0.9905 -
val_loss: 0.0532 - val_acc: 0.9864
Epoch 55/60
60000/60000 [=====] - 9s 144us/step - loss: 0.0276 - acc: 0.9909 -
val_loss: 0.0541 - val_acc: 0.9862
Epoch 56/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0252 - acc: 0.9918 -
val_loss: 0.0499 - val_acc: 0.9869
Epoch 57/60
60000/60000 [=====] - 9s 145us/step - loss: 0.0263 - acc: 0.9914 -
val_loss: 0.0509 - val_acc: 0.9874
Epoch 58/60
60000/60000 [=====] - 9s 144us/step - loss: 0.0254 - acc: 0.9919 -
val_loss: 0.0536 - val_acc: 0.9868
Epoch 59/60
60000/60000 [=====] - 9s 144us/step - loss: 0.0247 - acc: 0.9917 -
val_loss: 0.0574 - val_acc: 0.9863
Epoch 60/60
60000/60000 [=====] - 9s 144us/step - loss: 0.0245 - acc: 0.9919 -
val_loss: 0.0597 - val_acc: 0.9864

```

In [27]:

```

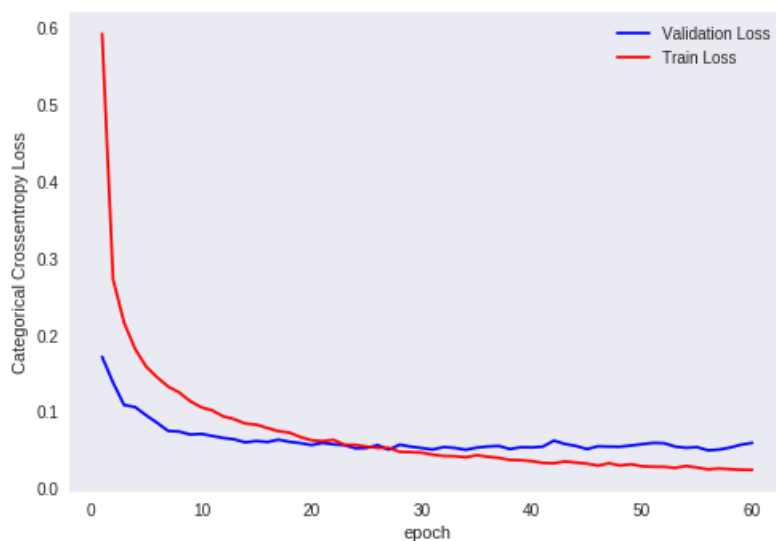
score = model_drop.evaluate(X_test, Y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax)

```

Test score: 0.059668766928038346  
Test accuracy: 0.9864



**Architecture 3: input (784) - ReLu(1024) - ReLu(512)-Relu(256)-Relu(128)-Relu(64)- sigmoid**

# 1.MLP + Batch-Norm on hidden Layers + AdamOptimizer

In [28]:

```
model_batch = Sequential()

model_batch.add(Dense(1024, activation='relu', input_shape=(input_dim,), kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(512, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(256, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(128, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(64, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_batch.add(BatchNormalization())

model_batch.add(Dense(output_dim, activation='softmax'))

model_batch.summary()
```

| Layer (type)                                 | Output Shape | Param # |
|--|--------------|---------|
| dense_15 (Dense)                             | (None, 1024) | 803840  |
| batch_normalization_11 (Batch Normalization) | (None, 1024) | 4096    |
| dense_16 (Dense)                             | (None, 512)  | 524800  |
| batch_normalization_12 (Batch Normalization) | (None, 512)  | 2048    |
| dense_17 (Dense)                             | (None, 256)  | 131328  |
| batch_normalization_13 (Batch Normalization) | (None, 256)  | 1024    |
| dense_18 (Dense)                             | (None, 128)  | 32896   |
| batch_normalization_14 (Batch Normalization) | (None, 128)  | 512     |
| dense_19 (Dense)                             | (None, 64)   | 8256    |
| batch_normalization_15 (Batch Normalization) | (None, 64)   | 256     |
| dense_20 (Dense)                             | (None, 10)   | 650     |
| Total params: 1,509,706                      |              |         |
| Trainable params: 1,505,738                  |              |         |
| Non-trainable params: 3,968                  |              |         |

In [29]:

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

history = model_batch.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, validation_data=(X_test, Y_test))
```

Train on 60000 samples, validate on 10000 samples  
Epoch 1/60  
60000/60000 [=====] - 13s 224us/step - loss: 0.2293 - acc: 0.9339 - val\_loss: 0.1111 - val\_acc: 0.9653  
Epoch 2/60  
60000/60000 [=====] - 11s 187us/step - loss: 0.0847 - acc: 0.9743 - val\_loss: 0.1000 - val\_acc: 0.9660

```
oss: 0.1002 - val_acc: 0.9690
Epoch 3/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0593 - acc: 0.9812 - val_l
oss: 0.0974 - val_acc: 0.9710
Epoch 4/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0459 - acc: 0.9853 - val_l
oss: 0.0895 - val_acc: 0.9747
Epoch 5/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0373 - acc: 0.9883 - val_l
oss: 0.0884 - val_acc: 0.9751
Epoch 6/60
60000/60000 [=====] - 11s 183us/step - loss: 0.0317 - acc: 0.9899 - val_l
oss: 0.0764 - val_acc: 0.9782
Epoch 7/60
60000/60000 [=====] - 11s 186us/step - loss: 0.0276 - acc: 0.9911 - val_l
oss: 0.0823 - val_acc: 0.9769
Epoch 8/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0225 - acc: 0.9928 - val_l
oss: 0.0813 - val_acc: 0.9757
Epoch 9/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0223 - acc: 0.9925 - val_l
oss: 0.0929 - val_acc: 0.9751
Epoch 10/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0234 - acc: 0.9919 - val_l
oss: 0.0817 - val_acc: 0.9783
Epoch 11/60
60000/60000 [=====] - 11s 191us/step - loss: 0.0180 - acc: 0.9944 - val_l
oss: 0.0889 - val_acc: 0.9767
Epoch 12/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0140 - acc: 0.9951 - val_l
oss: 0.0831 - val_acc: 0.9788
Epoch 13/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0140 - acc: 0.9955 - val_l
oss: 0.0790 - val_acc: 0.9789
Epoch 14/60
60000/60000 [=====] - 11s 186us/step - loss: 0.0173 - acc: 0.9941 - val_l
oss: 0.0830 - val_acc: 0.9788
Epoch 15/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0137 - acc: 0.9953 - val_l
oss: 0.0791 - val_acc: 0.9805
Epoch 16/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0114 - acc: 0.9963 - val_l
oss: 0.0773 - val_acc: 0.9805
Epoch 17/60
60000/60000 [=====] - 11s 186us/step - loss: 0.0129 - acc: 0.9955 - val_l
oss: 0.0819 - val_acc: 0.9789
Epoch 18/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0117 - acc: 0.9959 - val_l
oss: 0.0789 - val_acc: 0.9794
Epoch 19/60
60000/60000 [=====] - 11s 186us/step - loss: 0.0108 - acc: 0.9963 - val_l
oss: 0.0804 - val_acc: 0.9783
Epoch 20/60
60000/60000 [=====] - 11s 186us/step - loss: 0.0102 - acc: 0.9966 - val_l
oss: 0.0785 - val_acc: 0.9803
Epoch 21/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0116 - acc: 0.9964 - val_l
oss: 0.0954 - val_acc: 0.9768
Epoch 22/60
60000/60000 [=====] - 11s 184us/step - loss: 0.0089 - acc: 0.9970 - val_l
oss: 0.0829 - val_acc: 0.9794
Epoch 23/60
60000/60000 [=====] - 11s 181us/step - loss: 0.0086 - acc: 0.9972 - val_l
oss: 0.0871 - val_acc: 0.9801
Epoch 24/60
60000/60000 [=====] - 11s 184us/step - loss: 0.0073 - acc: 0.9976 - val_l
oss: 0.0859 - val_acc: 0.9807
Epoch 25/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0099 - acc: 0.9965 - val_l
oss: 0.0921 - val_acc: 0.9798
Epoch 26/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0086 - acc: 0.9971 - val_l
oss: 0.0905 - val_acc: 0.9801
Epoch 27/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0074 - acc: 0.9976 - val_l
oss: 0.0863 - val_acc: 0.9814
Epoch 28/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0088 - acc: 0.9968 - val_l
oss: 0.0888 - val_acc: 0.9808
```

```
60000/60000 [=====] - 11s 188us/step - loss: 0.0091 - acc: 0.9968 - val_l
oss: 0.0805 - val_acc: 0.9826
Epoch 29/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0069 - acc: 0.9979 - val_l
oss: 0.0916 - val_acc: 0.9791
Epoch 30/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0061 - acc: 0.9981 - val_l
oss: 0.0808 - val_acc: 0.9822
Epoch 31/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0069 - acc: 0.9978 - val_l
oss: 0.0841 - val_acc: 0.9809
Epoch 32/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0077 - acc: 0.9974 - val_l
oss: 0.0786 - val_acc: 0.9822
Epoch 33/60
60000/60000 [=====] - 11s 186us/step - loss: 0.0059 - acc: 0.9982 - val_l
oss: 0.0855 - val_acc: 0.9807
Epoch 34/60
60000/60000 [=====] - 11s 185us/step - loss: 0.0059 - acc: 0.9982 - val_l
oss: 0.0867 - val_acc: 0.9804
Epoch 35/60
60000/60000 [=====] - 11s 186us/step - loss: 0.0064 - acc: 0.9979 - val_l
oss: 0.0878 - val_acc: 0.9822
Epoch 36/60
60000/60000 [=====] - 11s 185us/step - loss: 0.0057 - acc: 0.9981 - val_l
oss: 0.0827 - val_acc: 0.9811
Epoch 37/60
60000/60000 [=====] - 11s 183us/step - loss: 0.0062 - acc: 0.9978 - val_l
oss: 0.0854 - val_acc: 0.9818
Epoch 38/60
60000/60000 [=====] - 11s 183us/step - loss: 0.0049 - acc: 0.9984 - val_l
oss: 0.0819 - val_acc: 0.9822
Epoch 39/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0056 - acc: 0.9982 - val_l
oss: 0.0874 - val_acc: 0.9816
Epoch 40/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0061 - acc: 0.9980 - val_l
oss: 0.0807 - val_acc: 0.9834
Epoch 41/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0054 - acc: 0.9983 - val_l
oss: 0.0953 - val_acc: 0.9806
Epoch 42/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0057 - acc: 0.9981 - val_l
oss: 0.0835 - val_acc: 0.9834
Epoch 43/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0051 - acc: 0.9983 - val_l
oss: 0.0882 - val_acc: 0.9818
Epoch 44/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0053 - acc: 0.9982 - val_l
oss: 0.1100 - val_acc: 0.9801
Epoch 45/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0050 - acc: 0.9986 - val_l
oss: 0.0804 - val_acc: 0.9842
Epoch 46/60
60000/60000 [=====] - 11s 185us/step - loss: 0.0034 - acc: 0.9990 - val_l
oss: 0.0871 - val_acc: 0.9824
Epoch 47/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0051 - acc: 0.9983 - val_l
oss: 0.0976 - val_acc: 0.9829
Epoch 48/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0052 - acc: 0.9985 - val_l
oss: 0.0895 - val_acc: 0.9840
Epoch 49/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0047 - acc: 0.9986 - val_l
oss: 0.0804 - val_acc: 0.9838
Epoch 50/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0047 - acc: 0.9986 - val_l
oss: 0.0800 - val_acc: 0.9829
Epoch 51/60
60000/60000 [=====] - 11s 186us/step - loss: 0.0033 - acc: 0.9991 - val_l
oss: 0.0766 - val_acc: 0.9842
Epoch 52/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0042 - acc: 0.9986 - val_l
oss: 0.0804 - val_acc: 0.9820
Epoch 53/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0043 - acc: 0.9985 - val_l
oss: 0.0886 - val_acc: 0.9820
```

```

Epoch 54/60
60000/60000 [=====] - 11s 187us/step - loss: 0.0055 - acc: 0.9982 - val_loss: 0.0893 - val_acc: 0.9822
Epoch 55/60
60000/60000 [=====] - 11s 185us/step - loss: 0.0036 - acc: 0.9987 - val_loss: 0.0907 - val_acc: 0.9818
Epoch 56/60
60000/60000 [=====] - 11s 181us/step - loss: 0.0031 - acc: 0.9992 - val_loss: 0.0855 - val_acc: 0.9828
Epoch 57/60
60000/60000 [=====] - 11s 182us/step - loss: 0.0043 - acc: 0.9987 - val_loss: 0.0825 - val_acc: 0.9823
Epoch 58/60
60000/60000 [=====] - 11s 182us/step - loss: 0.0029 - acc: 0.9990 - val_loss: 0.0818 - val_acc: 0.9844
Epoch 59/60
60000/60000 [=====] - 11s 182us/step - loss: 0.0027 - acc: 0.9993 - val_loss: 0.0832 - val_acc: 0.9834
Epoch 60/60
60000/60000 [=====] - 11s 181us/step - loss: 0.0039 - acc: 0.9988 - val_loss: 0.0933 - val_acc: 0.9820

```

In [30]:

```

score = model_drop.evaluate(X_test, Y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

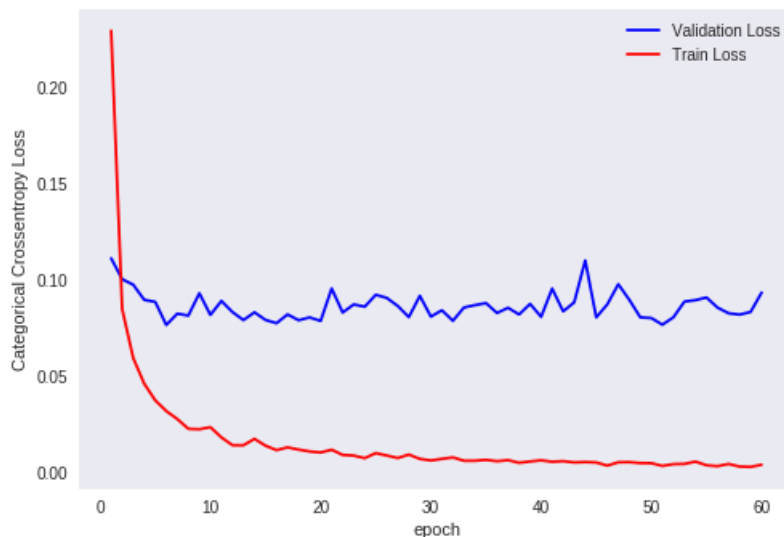
fig, ax = plt.subplots(1, 1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax)

```

Test score: 0.059668766928038346

Test accuracy: 0.9864



## 2. MLP + Dropout + AdamOptimizer

In [31]:

```

model_drop = Sequential()

model_drop.add(Dense(1024, activation='relu', input_shape=(input_dim,), kernel_initializer=RandomNormal(mean=0.0, stddev=0.039, seed=None)))
model_drop.add(BatchNormalization())
model_drop.add(Dropout(0.5))

model_drop.add(Dense(512, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55, seed=None)))
model_drop.add(BatchNormalization())

```

```

model_drop.add(Dropout(0.5))

model_drop.add(Dense(256, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55,
seed=None)) )
model_drop.add(BatchNormalization())
model_drop.add(Dropout(0.5))

model_drop.add(Dense(128, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55,
seed=None)) )
model_drop.add(BatchNormalization())
model_drop.add(Dropout(0.5))

model_drop.add(Dense(64, activation='relu', kernel_initializer=RandomNormal(mean=0.0, stddev=0.55,
seed=None)) )
model_drop.add(BatchNormalization())
model_drop.add(Dropout(0.5))

model_drop.add(Dense(output_dim, activation='softmax'))

model_drop.summary()

```

| Layer (type)                                 | Output Shape | Param # |
|--|--------------|---------|
| dense_21 (Dense)                             | (None, 1024) | 803840  |
| batch_normalization_16 (Batch Normalization) | (None, 1024) | 4096    |
| dropout_6 (Dropout)                          | (None, 1024) | 0       |
| dense_22 (Dense)                             | (None, 512)  | 524800  |
| batch_normalization_17 (Batch Normalization) | (None, 512)  | 2048    |
| dropout_7 (Dropout)                          | (None, 512)  | 0       |
| dense_23 (Dense)                             | (None, 256)  | 131328  |
| batch_normalization_18 (Batch Normalization) | (None, 256)  | 1024    |
| dropout_8 (Dropout)                          | (None, 256)  | 0       |
| dense_24 (Dense)                             | (None, 128)  | 32896   |
| batch_normalization_19 (Batch Normalization) | (None, 128)  | 512     |
| dropout_9 (Dropout)                          | (None, 128)  | 0       |
| dense_25 (Dense)                             | (None, 64)   | 8256    |
| batch_normalization_20 (Batch Normalization) | (None, 64)   | 256     |
| dropout_10 (Dropout)                         | (None, 64)   | 0       |
| dense_26 (Dense)                             | (None, 10)   | 650     |
| Total params: 1,509,706                      |              |         |
| Trainable params: 1,505,738                  |              |         |
| Non-trainable params: 3,968                  |              |         |

In [36]:

```

model_drop.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

history = model_drop.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, validation_data=(X_test, Y_test))

```

```

Train on 60000 samples, validate on 10000 samples
Epoch 1/60
60000/60000 [=====] - 14s 234us/step - loss: 1.6017 - acc: 0.4863 - val_loss: 0.3765 - val_acc: 0.9044
Epoch 2/60
60000/60000 [=====] - 11s 189us/step - loss: 0.6338 - acc: 0.8033 - val_loss: 0.3765 - val_acc: 0.9044

```



```
oss: 0.2279 - val_acc: 0.9319
Epoch 3/60
60000/60000 [=====] - 11s 189us/step - loss: 0.4424 - acc: 0.8710 - val_l
oss: 0.1846 - val_acc: 0.9499
Epoch 4/60
60000/60000 [=====] - 11s 189us/step - loss: 0.3453 - acc: 0.9036 - val_l
oss: 0.1547 - val_acc: 0.9566
Epoch 5/60
60000/60000 [=====] - 11s 189us/step - loss: 0.2850 - acc: 0.9216 - val_l
oss: 0.1354 - val_acc: 0.9623
Epoch 6/60
60000/60000 [=====] - 11s 188us/step - loss: 0.2492 - acc: 0.9336 - val_l
oss: 0.1176 - val_acc: 0.9680
Epoch 7/60
60000/60000 [=====] - 11s 189us/step - loss: 0.2317 - acc: 0.9369 - val_l
oss: 0.1143 - val_acc: 0.9702
Epoch 8/60
60000/60000 [=====] - 11s 189us/step - loss: 0.2115 - acc: 0.9440 - val_l
oss: 0.1074 - val_acc: 0.9720
Epoch 9/60
60000/60000 [=====] - 11s 189us/step - loss: 0.1979 - acc: 0.9475 - val_l
oss: 0.0971 - val_acc: 0.9738
Epoch 10/60
60000/60000 [=====] - 11s 188us/step - loss: 0.1833 - acc: 0.9519 - val_l
oss: 0.0942 - val_acc: 0.9761
Epoch 11/60
60000/60000 [=====] - 11s 188us/step - loss: 0.1692 - acc: 0.9553 - val_l
oss: 0.0955 - val_acc: 0.9752
Epoch 12/60
60000/60000 [=====] - 11s 188us/step - loss: 0.1573 - acc: 0.9579 - val_l
oss: 0.0907 - val_acc: 0.9769
Epoch 13/60
60000/60000 [=====] - 11s 188us/step - loss: 0.1501 - acc: 0.9609 - val_l
oss: 0.0879 - val_acc: 0.9780
Epoch 14/60
60000/60000 [=====] - 11s 189us/step - loss: 0.1453 - acc: 0.9617 - val_l
oss: 0.0870 - val_acc: 0.9779
Epoch 15/60
60000/60000 [=====] - 11s 188us/step - loss: 0.1363 - acc: 0.9649 - val_l
oss: 0.0829 - val_acc: 0.9789
Epoch 16/60
60000/60000 [=====] - 11s 188us/step - loss: 0.1321 - acc: 0.9650 - val_l
oss: 0.0776 - val_acc: 0.9805
Epoch 17/60
60000/60000 [=====] - 11s 188us/step - loss: 0.1267 - acc: 0.9672 - val_l
oss: 0.0788 - val_acc: 0.9798
Epoch 18/60
60000/60000 [=====] - 11s 188us/step - loss: 0.1187 - acc: 0.9692 - val_l
oss: 0.0792 - val_acc: 0.9812
Epoch 19/60
60000/60000 [=====] - 11s 189us/step - loss: 0.1161 - acc: 0.9697 - val_l
oss: 0.0741 - val_acc: 0.9813
Epoch 20/60
60000/60000 [=====] - 11s 188us/step - loss: 0.1135 - acc: 0.9710 - val_l
oss: 0.0792 - val_acc: 0.9810
Epoch 21/60
60000/60000 [=====] - 11s 190us/step - loss: 0.1111 - acc: 0.9712 - val_l
oss: 0.0704 - val_acc: 0.9823
Epoch 22/60
60000/60000 [=====] - 11s 189us/step - loss: 0.1038 - acc: 0.9724 - val_l
oss: 0.0742 - val_acc: 0.9823
Epoch 23/60
60000/60000 [=====] - 11s 189us/step - loss: 0.1015 - acc: 0.9733 - val_l
oss: 0.0715 - val_acc: 0.9834
Epoch 24/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0974 - acc: 0.9740 - val_l
oss: 0.0733 - val_acc: 0.9820
Epoch 25/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0952 - acc: 0.9753 - val_l
oss: 0.0716 - val_acc: 0.9819
Epoch 26/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0927 - acc: 0.9758 - val_l
oss: 0.0696 - val_acc: 0.9824
Epoch 27/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0928 - acc: 0.9758 - val_l
oss: 0.0679 - val_acc: 0.9833
Epoch 28/60
```

```
60000/60000 [=====] - 11s 189us/step - loss: 0.0894 - acc: 0.9764 - val_1
oss: 0.0681 - val_acc: 0.9833
Epoch 29/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0831 - acc: 0.9785 - val_1
oss: 0.0690 - val_acc: 0.9838
Epoch 30/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0851 - acc: 0.9774 - val_1
oss: 0.0662 - val_acc: 0.9846
Epoch 31/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0792 - acc: 0.9796 - val_1
oss: 0.0721 - val_acc: 0.9844
Epoch 32/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0815 - acc: 0.9785 - val_1
oss: 0.0703 - val_acc: 0.9833
Epoch 33/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0770 - acc: 0.9796 - val_1
oss: 0.0651 - val_acc: 0.9852
Epoch 34/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0740 - acc: 0.9805 - val_1
oss: 0.0663 - val_acc: 0.9853
Epoch 35/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0735 - acc: 0.9808 - val_1
oss: 0.0672 - val_acc: 0.9862
Epoch 36/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0721 - acc: 0.9809 - val_1
oss: 0.0621 - val_acc: 0.9856
Epoch 37/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0689 - acc: 0.9821 - val_1
oss: 0.0624 - val_acc: 0.9866
Epoch 38/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0700 - acc: 0.9812 - val_1
oss: 0.0676 - val_acc: 0.9854
Epoch 39/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0657 - acc: 0.9821 - val_1
oss: 0.0699 - val_acc: 0.9846
Epoch 40/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0629 - acc: 0.9830 - val_1
oss: 0.0662 - val_acc: 0.9851
Epoch 41/60
60000/60000 [=====] - 11s 188us/step - loss: 0.0608 - acc: 0.9833 - val_1
oss: 0.0663 - val_acc: 0.9857
Epoch 42/60
60000/60000 [=====] - 11s 191us/step - loss: 0.0635 - acc: 0.9835 - val_1
oss: 0.0674 - val_acc: 0.9846
Epoch 43/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0644 - acc: 0.9829 - val_1
oss: 0.0654 - val_acc: 0.9847
Epoch 44/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0595 - acc: 0.9846 - val_1
oss: 0.0641 - val_acc: 0.9857
Epoch 45/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0556 - acc: 0.9853 - val_1
oss: 0.0690 - val_acc: 0.9851
Epoch 46/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0612 - acc: 0.9836 - val_1
oss: 0.0660 - val_acc: 0.9854
Epoch 47/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0584 - acc: 0.9844 - val_1
oss: 0.0694 - val_acc: 0.9850
Epoch 48/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0562 - acc: 0.9856 - val_1
oss: 0.0612 - val_acc: 0.9863
Epoch 49/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0595 - acc: 0.9844 - val_1
oss: 0.0614 - val_acc: 0.9854
Epoch 50/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0558 - acc: 0.9850 - val_1
oss: 0.0630 - val_acc: 0.9859
Epoch 51/60
60000/60000 [=====] - 11s 191us/step - loss: 0.0546 - acc: 0.9855 - val_1
oss: 0.0626 - val_acc: 0.9865
Epoch 52/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0528 - acc: 0.9862 - val_1
oss: 0.0619 - val_acc: 0.9858
Epoch 53/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0521 - acc: 0.9862 - val_1
oss: 0.0667 - val_acc: 0.9850
```

```

Epoch 54/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0531 - acc: 0.9856 - val_loss: 0.0654 - val_acc: 0.9860
Epoch 55/60
60000/60000 [=====] - 11s 191us/step - loss: 0.0469 - acc: 0.9878 - val_loss: 0.0643 - val_acc: 0.9852
Epoch 56/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0489 - acc: 0.9871 - val_loss: 0.0689 - val_acc: 0.9851
Epoch 57/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0456 - acc: 0.9881 - val_loss: 0.0673 - val_acc: 0.9857
Epoch 58/60
60000/60000 [=====] - 11s 189us/step - loss: 0.0496 - acc: 0.9867 - val_loss: 0.0657 - val_acc: 0.9864
Epoch 59/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0498 - acc: 0.9871 - val_loss: 0.0620 - val_acc: 0.9862
Epoch 60/60
60000/60000 [=====] - 11s 190us/step - loss: 0.0483 - acc: 0.9874 - val_loss: 0.0626 - val_acc: 0.9873

```

In [37]:

```

score = model_drop.evaluate(X_test, Y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

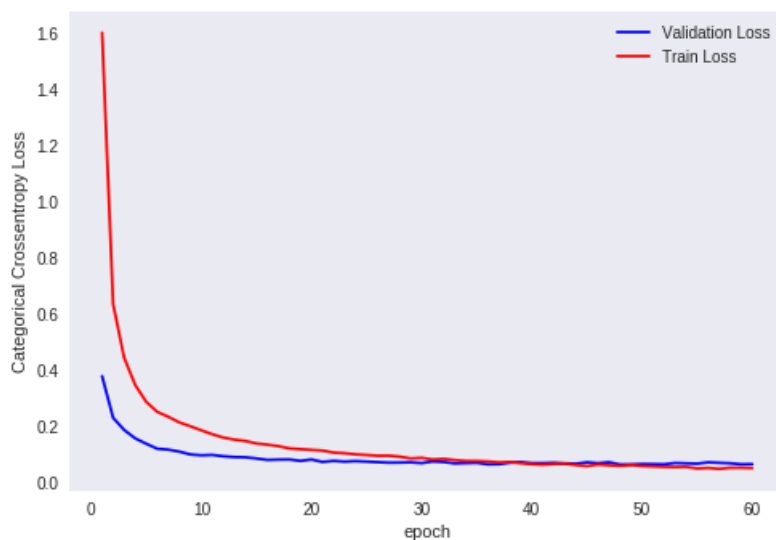
fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax)

```

Test score: 0.06261478376485174

Test accuracy: 0.9873



## Conclusion :

In [17]:

```
from prettytable import PrettyTable
```

In [24]:

```

x = PrettyTable()

x.field_names = ["Architecture", "Model", "Test_Score", "Test_Accuracy", "Epochs"]

x.add_row(["1 : 784-512-256 Sigmoid", "MLP + BN + Adam Opt", "0.091", "0.983", ">60"])

```

```

x.add_row(["1 : 784-512-256 Sigmoid", "MLP + AdamOpt + Dropout", "0.056", "0.985", "24"])
x.add_row(["2 : 784-1024-512-256 Sigmoid", "MLP + BN + Adam Opt", "0.056", "0.985", ">60"])
x.add_row(["2 : 784-1024-512-256 Sigmoid", "BN with Dropout", "0.059", "0.986", "23"])
x.add_row(["3 : 784-1024-512-256-128 Sigmoid", "MLP + BN + Adam Opt", "0.059", "0.986", ">60"])
x.add_row(["3 : 784-1024-512-256-128 Sigmoid", "MLP + AdamOpt + Dropout", "0.062", "0.987", "31"])

print(x)

```

| Architecture                     | Model                   | Test_Score | Test_Accuracy | Epochs |
|----------------------------------|-------------------------|------------|---------------|--------|
| 1 : 784-512-256 Sigmoid          | MLP + BN + Adam Opt     | 0.091      | 0.983         | >60    |
| 1 : 784-512-256 Sigmoid          | MLP + AdamOpt + Dropout | 0.056      | 0.985         | 24     |
| 2 : 784-1024-512-256 Sigmoid     | MLP + BN + Adam Opt     | 0.056      | 0.985         | >60    |
| 2 : 784-1024-512-256 Sigmoid     | BN with Dropout         | 0.059      | 0.986         | 23     |
| 3 : 784-1024-512-256-128 Sigmoid | MLP + BN + Adam Opt     | 0.059      | 0.986         | >60    |
| 3 : 784-1024-512-256-128 Sigmoid | MLP + AdamOpt + Dropout | 0.062      | 0.987         | 31     |