

# Neural Networks image recognition - MultiLayer Perceptron

Use both MLNN for the following problem.

1. Add random noise (see below on `size` parameter on `np.random.normal`) to the images in training and testing. **Make sure each image gets a different noise feature added to it. Inspect by printing out several images. Note - the `size` parameter should match the data.**
2. Compare the `accuracy` of train and val after N epochs for MLNN with and without noise.
3. Vary the amount of noise by changing the `scale` parameter in `np.random.normal` by a factor. Use `.1, .5, 1.0, 2.0, 4.0` for the `scale` and keep track of the `accuracy` for training and validation and plot these results.

## np.random.normal

### Parameters

#### loc

Mean ("centre") of the distribution.

#### scale

Standard deviation (spread or "width") of the distribution. Must be non-negative.

#### size

Output shape. If the given shape is, e.g., (m, n, k), then m n k samples are drawn. If size is None (default), a single value is returned if loc and scale are both scalars. Otherwise, `np.broadcast(loc, scale).size` samples are drawn.

# Neural Networks - Image Recognition

```
In [13]: import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.optimizers import RMSprop
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend
import matplotlib.pyplot as plt
```

```
import numpy as np
plt.rcParams['figure.figsize'] = (24, 16)
plt.rcParams['font.size'] = 14
```

## Multi Layer Neural Network

Trains a simple deep NN on the MNIST dataset. Gets to 98.40% test accuracy after 20 epochs (there is *a lot* of margin for parameter tuning).

```
In [2]: # the data, shuffled and split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train.reshape(60000, 784)
x_test = x_test.reshape(10000, 784)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')
```

```
60000 train samples
10000 test samples
```

```
In [3]: def generate_noise(data, scale = 0.1):
    new_data = data + np.random.normal(0, scale, (784,))
    return new_data
```

```
In [4]: def run_model(x_train, x_test, y_train, y_test, scale = 0):
    batch_size = 128
    num_classes = 10
    epochs = 20

    x_train = x_train.copy()
    x_test = x_test.copy()
    y_train = y_train.copy()
    y_test = y_test.copy()

    x_train = generate_noise(x_train, scale)
    x_test = generate_noise(x_test, scale)

    plt.imshow(x_train[12].reshape(28,28))
    plt.show()
    plt.close()

    # convert class vectors to binary class matrices
    y_train = keras.utils.to_categorical(y_train, num_classes)
    y_test = keras.utils.to_categorical(y_test, num_classes)

    model = Sequential()
    model.add(Dense(512, activation='relu', input_shape=(784,)))
    model.add(Dropout(0.2))
    model.add(Dense(512, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(10, activation='softmax'))
```

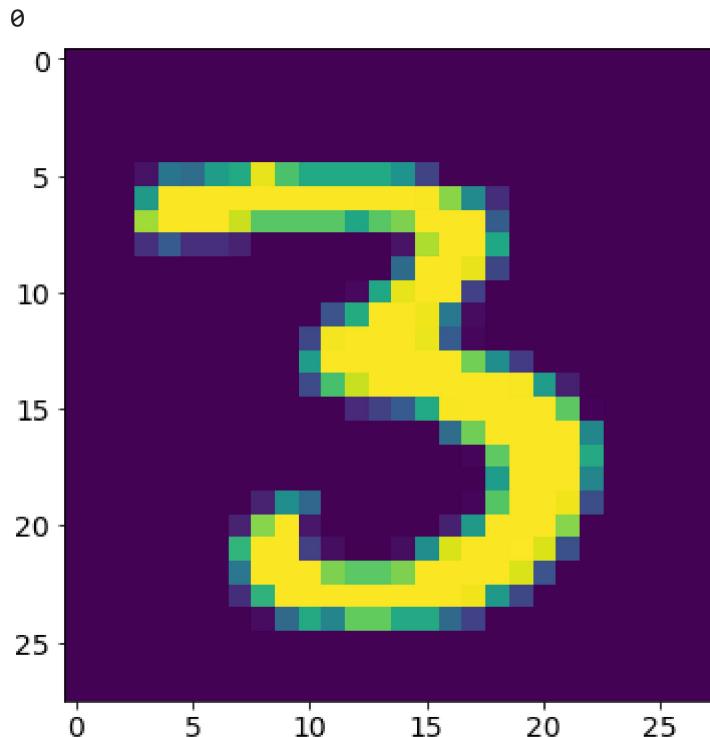
```
model.summary()

model.compile(loss='categorical_crossentropy',
                optimizer=RMSprop(),
                metrics=['accuracy'])

history = model.fit(x_train, y_train,
                      batch_size=batch_size,
                      epochs=epochs,
                      verbose=1,
                      validation_data=(x_test, y_test))
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
return[
    history, score
]
```

```
In [6]: historys = []
scores = []
scales = [0, .1, .5, 1.0, 2.0, 4.0]

for i in scales:
    print(i)
    history, score = run_model(x_train, x_test, y_train, y_test, scale = i)
    historys.append(history)
    scores.append(score)
```

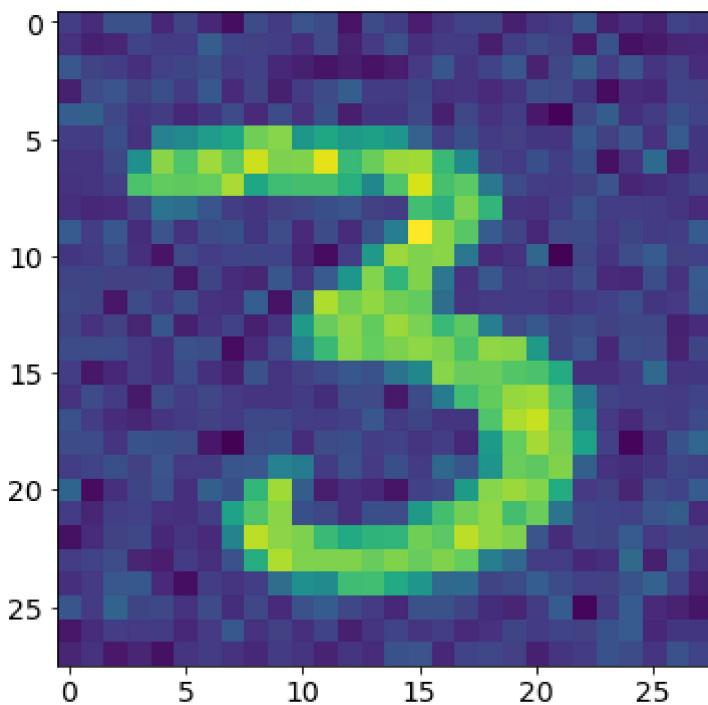


Model: "sequential\_1"

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 512)	401920
dropout_2 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 10)	5130
<hr/>		
Total params: 669,706		
Trainable params: 669,706		
Non-trainable params: 0		

Epoch 1/20  
469/469 [=====] - 6s 12ms/step - loss: 0.2452 - accuracy: 0.9253 - val\_loss: 0.1222 - val\_accuracy: 0.9627  
Epoch 2/20  
469/469 [=====] - 6s 12ms/step - loss: 0.1042 - accuracy: 0.9685 - val\_loss: 0.0827 - val\_accuracy: 0.9754  
Epoch 3/20  
469/469 [=====] - 6s 12ms/step - loss: 0.0756 - accuracy: 0.9769 - val\_loss: 0.0823 - val\_accuracy: 0.9768  
Epoch 4/20  
469/469 [=====] - 6s 13ms/step - loss: 0.0601 - accuracy: 0.9827 - val\_loss: 0.0760 - val\_accuracy: 0.9785  
Epoch 5/20  
469/469 [=====] - 5s 12ms/step - loss: 0.0511 - accuracy: 0.9845 - val\_loss: 0.0698 - val\_accuracy: 0.9820  
Epoch 6/20  
469/469 [=====] - 5s 12ms/step - loss: 0.0452 - accuracy: 0.9863 - val\_loss: 0.0817 - val\_accuracy: 0.9801  
Epoch 7/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0378 - accuracy: 0.9890 - val\_loss: 0.0849 - val\_accuracy: 0.9802  
Epoch 8/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0345 - accuracy: 0.9899 - val\_loss: 0.0916 - val\_accuracy: 0.9811  
Epoch 9/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0310 - accuracy: 0.9905 - val\_loss: 0.0930 - val\_accuracy: 0.9816  
Epoch 10/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0282 - accuracy: 0.9918 - val\_loss: 0.0956 - val\_accuracy: 0.9834  
Epoch 11/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0277 - accuracy: 0.9920 - val\_loss: 0.0976 - val\_accuracy: 0.9835  
Epoch 12/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0275 - accuracy: 0.9923 - val\_loss: 0.1102 - val\_accuracy: 0.9800  
Epoch 13/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0249 - accuracy: 0.9929 - val\_loss: 0.0924 - val\_accuracy: 0.9837  
Epoch 14/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0235 - accuracy: 0.

```
9934 - val_loss: 0.1051 - val_accuracy: 0.9828
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 0.0223 - accuracy: 0.
9938 - val_loss: 0.1104 - val_accuracy: 0.9823
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.0218 - accuracy: 0.
9939 - val_loss: 0.1190 - val_accuracy: 0.9815
Epoch 17/20
469/469 [=====] - 5s 12ms/step - loss: 0.0191 - accuracy: 0.
9947 - val_loss: 0.1344 - val_accuracy: 0.9816
Epoch 18/20
469/469 [=====] - 5s 11ms/step - loss: 0.0208 - accuracy: 0.
9947 - val_loss: 0.1342 - val_accuracy: 0.9804
Epoch 19/20
469/469 [=====] - 5s 11ms/step - loss: 0.0188 - accuracy: 0.
9952 - val_loss: 0.1286 - val_accuracy: 0.9827
Test loss: 0.12867088615894318
Test accuracy: 0.983299970626831
0.1
```

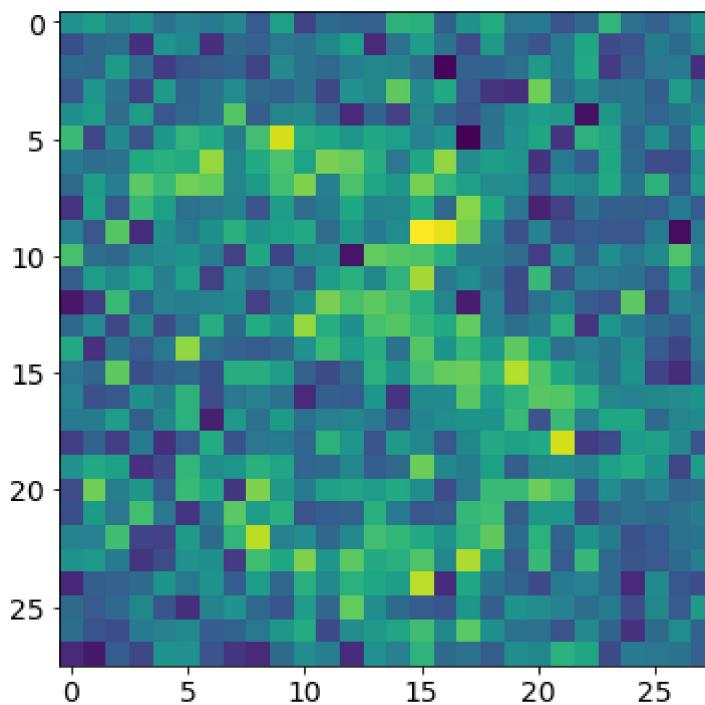


Model: "sequential\_2"

Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 512)	401920
dropout_4 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 512)	262656
dropout_5 (Dropout)	(None, 512)	0
dense_8 (Dense)	(None, 10)	5130
<hr/>		
Total params: 669,706		
Trainable params: 669,706		
Non-trainable params: 0		

Epoch 1/20  
469/469 [=====] - 6s 11ms/step - loss: 0.2707 - accuracy: 0.9154 - val\_loss: 0.1269 - val\_accuracy: 0.9564  
Epoch 2/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1163 - accuracy: 0.9651 - val\_loss: 0.1226 - val\_accuracy: 0.9627  
Epoch 3/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0898 - accuracy: 0.9730 - val\_loss: 0.1261 - val\_accuracy: 0.9666  
Epoch 4/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0723 - accuracy: 0.9788 - val\_loss: 0.1378 - val\_accuracy: 0.9692  
Epoch 5/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0619 - accuracy: 0.9815 - val\_loss: 0.1853 - val\_accuracy: 0.9640  
Epoch 6/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0577 - accuracy: 0.9820 - val\_loss: 0.1539 - val\_accuracy: 0.9735  
Epoch 7/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0506 - accuracy: 0.9853 - val\_loss: 0.2067 - val\_accuracy: 0.9712  
Epoch 8/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0486 - accuracy: 0.9857 - val\_loss: 0.2891 - val\_accuracy: 0.9672  
Epoch 9/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0430 - accuracy: 0.9875 - val\_loss: 0.3204 - val\_accuracy: 0.9675  
Epoch 10/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0430 - accuracy: 0.9882 - val\_loss: 0.3791 - val\_accuracy: 0.9656  
Epoch 11/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0433 - accuracy: 0.9882 - val\_loss: 0.3905 - val\_accuracy: 0.9694  
Epoch 12/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0382 - accuracy: 0.9895 - val\_loss: 0.3879 - val\_accuracy: 0.9715  
Epoch 13/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0356 - accuracy: 0.9902 - val\_loss: 0.8026 - val\_accuracy: 0.9471  
Epoch 14/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0360 - accuracy: 0.

```
9906 - val_loss: 0.6580 - val_accuracy: 0.9611
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 0.0359 - accuracy: 0.
9904 - val_loss: 0.7126 - val_accuracy: 0.9617
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.0385 - accuracy: 0.
9906 - val_loss: 0.8616 - val_accuracy: 0.9549
Epoch 17/20
469/469 [=====] - 5s 11ms/step - loss: 0.0364 - accuracy: 0.
9906 - val_loss: 0.7333 - val_accuracy: 0.9645
Epoch 18/20
469/469 [=====] - 5s 11ms/step - loss: 0.0371 - accuracy: 0.
9913 - val_loss: 0.7475 - val_accuracy: 0.9678
Epoch 19/20
469/469 [=====] - 5s 11ms/step - loss: 0.0329 - accuracy: 0.
9921 - val_loss: 0.9150 - val_accuracy: 0.9633
Test loss: 1.013909935951233
Test accuracy: 0.9603000283241272
0.5
```



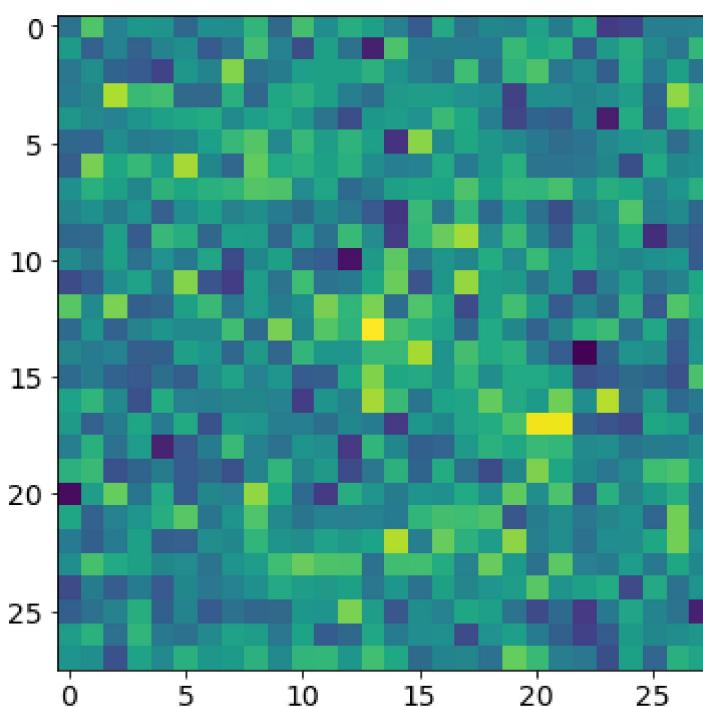
Model: "sequential\_3"

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 512)	401920
dropout_6 (Dropout)	(None, 512)	0
dense_10 (Dense)	(None, 512)	262656
dropout_7 (Dropout)	(None, 512)	0
dense_11 (Dense)	(None, 10)	5130

Total params: 669,706  
Trainable params: 669,706  
Non-trainable params: 0

Epoch 1/20  
469/469 [=====] - 6s 11ms/step - loss: 0.5054 - accuracy: 0.8396 - val\_loss: 0.6250 - val\_accuracy: 0.8061  
Epoch 2/20  
469/469 [=====] - 5s 11ms/step - loss: 0.2127 - accuracy: 0.9334 - val\_loss: 0.4462 - val\_accuracy: 0.8713  
Epoch 3/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1653 - accuracy: 0.9486 - val\_loss: 0.2612 - val\_accuracy: 0.9305  
Epoch 4/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1407 - accuracy: 0.9571 - val\_loss: 0.3001 - val\_accuracy: 0.9356  
Epoch 5/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1287 - accuracy: 0.9614 - val\_loss: 0.3524 - val\_accuracy: 0.9347  
Epoch 6/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1189 - accuracy: 0.9644 - val\_loss: 0.4286 - val\_accuracy: 0.9392  
Epoch 7/20  
469/469 [=====] - 6s 12ms/step - loss: 0.1128 - accuracy: 0.9676 - val\_loss: 0.5733 - val\_accuracy: 0.9364  
Epoch 8/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1097 - accuracy: 0.9692 - val\_loss: 0.7186 - val\_accuracy: 0.9344  
Epoch 9/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1038 - accuracy: 0.9701 - val\_loss: 0.8739 - val\_accuracy: 0.9316  
Epoch 10/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1063 - accuracy: 0.9702 - val\_loss: 1.2201 - val\_accuracy: 0.9195  
Epoch 11/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1032 - accuracy: 0.9720 - val\_loss: 1.6300 - val\_accuracy: 0.9094  
Epoch 12/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1036 - accuracy: 0.9717 - val\_loss: 1.7962 - val\_accuracy: 0.9094  
Epoch 13/20  
469/469 [=====] - 5s 11ms/step - loss: 0.1009 - accuracy: 0.9730 - val\_loss: 3.0379 - val\_accuracy: 0.8721  
Epoch 14/20  
469/469 [=====] - 5s 11ms/step - loss: 0.0988 - accuracy: 0.

```
9734 - val_loss: 3.8523 - val_accuracy: 0.8515
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 0.0975 - accuracy: 0.
9738 - val_loss: 3.5439 - val_accuracy: 0.8710
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.0957 - accuracy: 0.
9745 - val_loss: 4.3479 - val_accuracy: 0.8560
Epoch 17/20
469/469 [=====] - 6s 12ms/step - loss: 0.0964 - accuracy: 0.
9749 - val_loss: 5.1644 - val_accuracy: 0.8488
Epoch 18/20
469/469 [=====] - 7s 16ms/step - loss: 0.0939 - accuracy: 0.
9757 - val_loss: 6.8909 - val_accuracy: 0.8289
Epoch 19/20
469/469 [=====] - 7s 14ms/step - loss: 0.0968 - accuracy: 0.
9749 - val_loss: 5.2245 - val_accuracy: 0.8576
Epoch 20/20
469/469 [=====] - 7s 15ms/step - loss: 0.0938 - accuracy: 0.
9763 - val_loss: 7.0371 - val_accuracy: 0.8259
Test loss: 7.037131309509277
Test accuracy: 0.8259000182151794
1.0
```



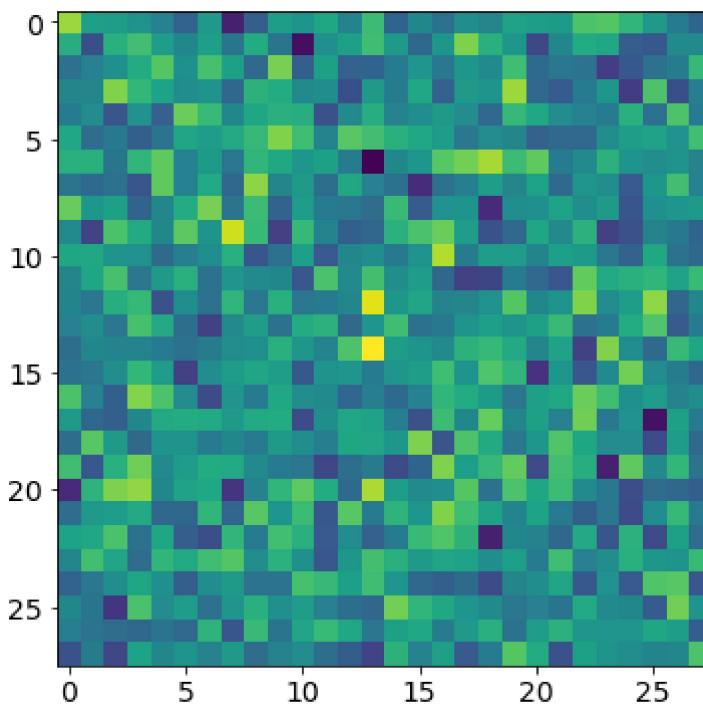
Model: "sequential\_4"

Layer (type)	Output Shape	Param #
dense_12 (Dense)	(None, 512)	401920
dropout_8 (Dropout)	(None, 512)	0
dense_13 (Dense)	(None, 512)	262656
dropout_9 (Dropout)	(None, 512)	0
dense_14 (Dense)	(None, 10)	5130

Total params: 669,706  
Trainable params: 669,706  
Non-trainable params: 0

Epoch 1/20  
469/469 [=====] - 6s 12ms/step - loss: 0.8314 - accuracy: 0.7313 - val\_loss: 1.3429 - val\_accuracy: 0.5868  
Epoch 2/20  
469/469 [=====] - 5s 11ms/step - loss: 0.3739 - accuracy: 0.8816 - val\_loss: 1.3398 - val\_accuracy: 0.5838  
Epoch 3/20  
469/469 [=====] - 7s 15ms/step - loss: 0.2834 - accuracy: 0.9116 - val\_loss: 1.0153 - val\_accuracy: 0.6954  
Epoch 4/20  
469/469 [=====] - 9s 19ms/step - loss: 0.2394 - accuracy: 0.9258 - val\_loss: 0.9295 - val\_accuracy: 0.7397  
Epoch 5/20  
469/469 [=====] - 7s 14ms/step - loss: 0.2123 - accuracy: 0.9344 - val\_loss: 1.4301 - val\_accuracy: 0.6559  
Epoch 6/20  
469/469 [=====] - 6s 12ms/step - loss: 0.2004 - accuracy: 0.9380 - val\_loss: 1.3252 - val\_accuracy: 0.6953  
Epoch 7/20  
469/469 [=====] - 6s 14ms/step - loss: 0.1849 - accuracy: 0.9441 - val\_loss: 1.7504 - val\_accuracy: 0.6590  
Epoch 8/20  
469/469 [=====] - 6s 13ms/step - loss: 0.1802 - accuracy: 0.9464 - val\_loss: 1.8345 - val\_accuracy: 0.6955  
Epoch 9/20  
469/469 [=====] - 6s 13ms/step - loss: 0.1727 - accuracy: 0.9478 - val\_loss: 1.6017 - val\_accuracy: 0.7337  
Epoch 10/20  
469/469 [=====] - 6s 13ms/step - loss: 0.1664 - accuracy: 0.9504 - val\_loss: 2.1592 - val\_accuracy: 0.7152  
Epoch 11/20  
469/469 [=====] - 7s 14ms/step - loss: 0.1630 - accuracy: 0.9521 - val\_loss: 1.9089 - val\_accuracy: 0.7636  
Epoch 12/20  
469/469 [=====] - 7s 14ms/step - loss: 0.1572 - accuracy: 0.9538 - val\_loss: 2.2313 - val\_accuracy: 0.7515  
Epoch 13/20  
469/469 [=====] - 6s 13ms/step - loss: 0.1540 - accuracy: 0.9538 - val\_loss: 2.6745 - val\_accuracy: 0.7534  
Epoch 14/20  
469/469 [=====] - 6s 13ms/step - loss: 0.1573 - accuracy: 0.

```
9540 - val_loss: 2.8385 - val_accuracy: 0.7676
Epoch 15/20
469/469 [=====] - 6s 13ms/step - loss: 0.1580 - accuracy: 0.
9545 - val_loss: 3.6951 - val_accuracy: 0.7467
Epoch 16/20
469/469 [=====] - 6s 13ms/step - loss: 0.1548 - accuracy: 0.
9566 - val_loss: 4.1483 - val_accuracy: 0.7501
Epoch 17/20
469/469 [=====] - 6s 13ms/step - loss: 0.1520 - accuracy: 0.
9564 - val_loss: 4.2347 - val_accuracy: 0.7626
Epoch 18/20
469/469 [=====] - 6s 13ms/step - loss: 0.1546 - accuracy: 0.
9571 - val_loss: 5.1134 - val_accuracy: 0.7451
Epoch 19/20
469/469 [=====] - 6s 13ms/step - loss: 0.1510 - accuracy: 0.
9581 - val_loss: 6.6173 - val_accuracy: 0.7350
Epoch 20/20
469/469 [=====] - 6s 13ms/step - loss: 0.1580 - accuracy: 0.
9570 - val_loss: 6.7910 - val_accuracy: 0.7409
Test loss: 6.791043758392334
Test accuracy: 0.7408999800682068
2.0
```



Model: "sequential\_5"

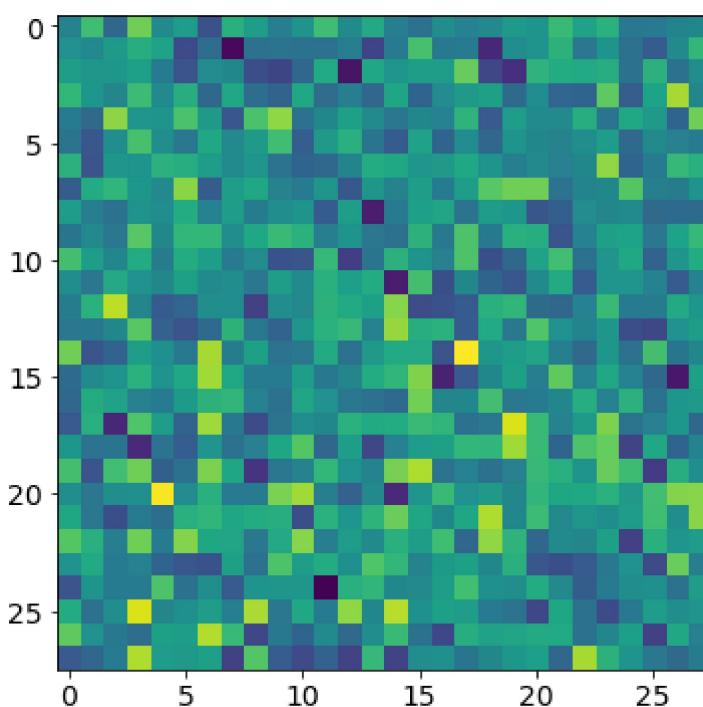
Layer (type)	Output Shape	Param #
dense_15 (Dense)	(None, 512)	401920
dropout_10 (Dropout)	(None, 512)	0
dense_16 (Dense)	(None, 512)	262656
dropout_11 (Dropout)	(None, 512)	0
dense_17 (Dense)	(None, 10)	5130

Total params: 669,706  
Trainable params: 669,706  
Non-trainable params: 0

---

Epoch 1/20  
469/469 [=====] - 7s 13ms/step - loss: 1.3635 - accuracy: 0.  
5443 - val\_loss: 4.0867 - val\_accuracy: 0.2197  
Epoch 2/20  
469/469 [=====] - 6s 13ms/step - loss: 0.7103 - accuracy: 0.  
7623 - val\_loss: 4.8570 - val\_accuracy: 0.2453  
Epoch 3/20  
469/469 [=====] - 6s 13ms/step - loss: 0.5625 - accuracy: 0.  
8162 - val\_loss: 4.3887 - val\_accuracy: 0.3384  
Epoch 4/20  
469/469 [=====] - 6s 14ms/step - loss: 0.4791 - accuracy: 0.  
8435 - val\_loss: 6.2039 - val\_accuracy: 0.2153  
Epoch 5/20  
469/469 [=====] - 6s 13ms/step - loss: 0.4284 - accuracy: 0.  
8604 - val\_loss: 5.5282 - val\_accuracy: 0.2565  
Epoch 6/20  
469/469 [=====] - 6s 13ms/step - loss: 0.3953 - accuracy: 0.  
8732 - val\_loss: 4.5698 - val\_accuracy: 0.3228  
Epoch 7/20  
469/469 [=====] - 6s 13ms/step - loss: 0.3672 - accuracy: 0.  
8819 - val\_loss: 5.5075 - val\_accuracy: 0.2711  
Epoch 8/20  
469/469 [=====] - 6s 13ms/step - loss: 0.3508 - accuracy: 0.  
8877 - val\_loss: 5.5597 - val\_accuracy: 0.2673  
Epoch 9/20  
469/469 [=====] - 6s 13ms/step - loss: 0.3362 - accuracy: 0.  
8932 - val\_loss: 5.0232 - val\_accuracy: 0.3007  
Epoch 10/20  
469/469 [=====] - 6s 13ms/step - loss: 0.3217 - accuracy: 0.  
8971 - val\_loss: 5.2766 - val\_accuracy: 0.2984  
Epoch 11/20  
469/469 [=====] - 6s 13ms/step - loss: 0.3101 - accuracy: 0.  
9026 - val\_loss: 5.6141 - val\_accuracy: 0.2646  
Epoch 12/20  
469/469 [=====] - 7s 14ms/step - loss: 0.3082 - accuracy: 0.  
9042 - val\_loss: 6.7024 - val\_accuracy: 0.2049  
Epoch 13/20  
469/469 [=====] - 7s 14ms/step - loss: 0.2971 - accuracy: 0.  
9061 - val\_loss: 4.7633 - val\_accuracy: 0.3545  
Epoch 14/20  
469/469 [=====] - 6s 14ms/step - loss: 0.2903 - accuracy: 0.

```
9083 - val_loss: 5.0706 - val_accuracy: 0.3034
Epoch 15/20
469/469 [=====] - 7s 15ms/step - loss: 0.2878 - accuracy: 0.
9091 - val_loss: 5.1036 - val_accuracy: 0.3518
Epoch 16/20
469/469 [=====] - 8s 17ms/step - loss: 0.2798 - accuracy: 0.
9137 - val_loss: 6.0068 - val_accuracy: 0.3101
Epoch 17/20
469/469 [=====] - 10s 21ms/step - loss: 0.2738 - accuracy: 0.
9152 - val_loss: 5.6070 - val_accuracy: 0.3448
Epoch 18/20
469/469 [=====] - 8s 18ms/step - loss: 0.2732 - accuracy: 0.
9157 - val_loss: 5.0956 - val_accuracy: 0.3437
Epoch 19/20
469/469 [=====] - 7s 15ms/step - loss: 0.2668 - accuracy: 0.
9185 - val_loss: 6.8707 - val_accuracy: 0.2836
Test loss: 5.012186050415039
Test accuracy: 0.3684000074863434
4.0
```



Model: "sequential\_6"

Layer (type)	Output Shape	Param #
dense_18 (Dense)	(None, 512)	401920
dropout_12 (Dropout)	(None, 512)	0
dense_19 (Dense)	(None, 512)	262656
dropout_13 (Dropout)	(None, 512)	0
dense_20 (Dense)	(None, 10)	5130

Total params: 669,706  
Trainable params: 669,706  
Non-trainable params: 0

Epoch 1/20  
469/469 [=====] - 7s 14ms/step - loss: 2.4299 - accuracy: 0.  
1123 - val\_loss: 6.0913 - val\_accuracy: 0.1028  
Epoch 2/20  
469/469 [=====] - 6s 13ms/step - loss: 2.3015 - accuracy: 0.  
1125 - val\_loss: 6.7602 - val\_accuracy: 0.1028  
Epoch 3/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3015 - accuracy: 0.  
1124 - val\_loss: 7.2550 - val\_accuracy: 0.1028  
Epoch 4/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3017 - accuracy: 0.  
1124 - val\_loss: 7.3985 - val\_accuracy: 0.1028  
Epoch 5/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3015 - accuracy: 0.  
1124 - val\_loss: 6.8070 - val\_accuracy: 0.1028  
Epoch 6/20  
469/469 [=====] - 6s 13ms/step - loss: 2.3015 - accuracy: 0.  
1125 - val\_loss: 7.9412 - val\_accuracy: 0.1028  
Epoch 7/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3015 - accuracy: 0.  
1123 - val\_loss: 7.5547 - val\_accuracy: 0.1028  
Epoch 8/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3015 - accuracy: 0.  
1124 - val\_loss: 7.6026 - val\_accuracy: 0.1028  
Epoch 9/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3014 - accuracy: 0.  
1123 - val\_loss: 7.1865 - val\_accuracy: 0.1028  
Epoch 10/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.  
1124 - val\_loss: 7.1165 - val\_accuracy: 0.1028  
Epoch 11/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.  
1124 - val\_loss: 7.1161 - val\_accuracy: 0.1028  
Epoch 12/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.  
1124 - val\_loss: 7.0311 - val\_accuracy: 0.1028  
Epoch 13/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.  
1124 - val\_loss: 7.0967 - val\_accuracy: 0.1028  
Epoch 14/20  
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.

```
1124 - val_loss: 7.0700 - val_accuracy: 0.1028
Epoch 15/20
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.
1124 - val_loss: 7.0723 - val_accuracy: 0.1028
Epoch 16/20
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.
1124 - val_loss: 7.0979 - val_accuracy: 0.1028
Epoch 17/20
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.
1124 - val_loss: 7.0687 - val_accuracy: 0.1028
Epoch 18/20
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.
1124 - val_loss: 7.0596 - val_accuracy: 0.1028
Epoch 19/20
469/469 [=====] - 6s 13ms/step - loss: 2.3013 - accuracy: 0.
1124 - val_loss: 7.0655 - val_accuracy: 0.1028
Epoch 20/20
469/469 [=====] - 6s 12ms/step - loss: 2.3013 - accuracy: 0.
1124 - val_loss: 7.0642 - val_accuracy: 0.1028
Test loss: 7.064211845397949
Test accuracy: 0.10279999673366547
```

```
In [22]: fig, ax = plt.subplots(2,2)
for i in range(len(scales)):
    ax[0,0].plot(historys[i].history['loss'], label = scales[i])
    ax[0,0].set_xlabel('Epochs')
    ax[0,0].set_ylabel('Loss')
    ax[0,0].legend(loc = "upper right", title = "Noise Factor")
    ax[0,0].set_title("Plot of Training Loss")

    ax[1,0].plot(historys[i].history['val_loss'], label = scales[i])
    ax[1,0].set_xlabel('Epochs')
    ax[1,0].set_ylabel('Loss')
    ax[1,0].legend(loc = "upper right", title = "Noise Factor")
    ax[1,0].set_title("Plot of Test Loss")

    ax[0,1].plot(historys[i].history['accuracy'], label = scales[i])
    ax[0,1].set_xlabel('Epochs')
    ax[0,1].set_ylabel('Accuracy')
    ax[0,1].legend(loc = "upper right", title = "Noise Factor")
    ax[0,1].set_title("Plot of Training Accuracy")

    ax[1,1].plot(historys[i].history['val_accuracy'], label = scales[i])
    ax[1,1].set_xlabel('Epochs')
    ax[1,1].set_ylabel('Accuracy')
    ax[1,1].legend(loc = "upper right", title = "Noise Factor")
    ax[1,1].set_title("Plot of Test Accuracy")
```

