

# 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 [84]: 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
```

### 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).

## Without Noise

```
In [85]: # the data, shuffled and split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

```
In [86]: 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 [87]: batch_size = 128
num_classes = 10
epochs = 20

# 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])
```

Model: "sequential\_20"

Layer (type)	Output Shape	Param #
dense_60 (Dense)	(None, 512)	401920
dropout_40 (Dropout)	(None, 512)	0
dense_61 (Dense)	(None, 512)	262656
dropout_41 (Dropout)	(None, 512)	0
dense_62 (Dense)	(None, 10)	5130

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

Epoch 1/20

2022-11-06 19:29:05.261298: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.

469/469 [=====] - ETA: 0s - loss: 0.2439 - accuracy: 0.9252

2022-11-06 19:29:10.250477: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.

```
469/469 [=====] - 5s 11ms/step - loss: 0.2439 - accuracy: 0.925
2 - val_loss: 0.1192 - val_accuracy: 0.9625
Epoch 2/20
469/469 [=====] - 5s 11ms/step - loss: 0.1021 - accuracy: 0.968
7 - val_loss: 0.0747 - val_accuracy: 0.9765
Epoch 3/20
469/469 [=====] - 5s 11ms/step - loss: 0.0738 - accuracy: 0.977
9 - val_loss: 0.0707 - val_accuracy: 0.9793
Epoch 4/20
469/469 [=====] - 5s 11ms/step - loss: 0.0591 - accuracy: 0.982
2 - val_loss: 0.0833 - val_accuracy: 0.9768
Epoch 5/20
469/469 [=====] - 5s 11ms/step - loss: 0.0495 - accuracy: 0.985
3 - val_loss: 0.0747 - val_accuracy: 0.9812
Epoch 6/20
469/469 [=====] - 5s 11ms/step - loss: 0.0430 - accuracy: 0.987
9 - val_loss: 0.0777 - val_accuracy: 0.9819
Epoch 7/20
469/469 [=====] - 5s 11ms/step - loss: 0.0362 - accuracy: 0.989
1 - val_loss: 0.0825 - val_accuracy: 0.9827
Epoch 8/20
469/469 [=====] - 5s 11ms/step - loss: 0.0321 - accuracy: 0.990
4 - val_loss: 0.1025 - val_accuracy: 0.9805
Epoch 9/20
469/469 [=====] - 5s 11ms/step - loss: 0.0273 - accuracy: 0.992
0 - val_loss: 0.0868 - val_accuracy: 0.9820
Epoch 10/20
469/469 [=====] - 5s 11ms/step - loss: 0.0264 - accuracy: 0.992
6 - val_loss: 0.1013 - val_accuracy: 0.9825
Epoch 11/20
469/469 [=====] - 5s 11ms/step - loss: 0.0252 - accuracy: 0.992
5 - val_loss: 0.1171 - val_accuracy: 0.9801
Epoch 12/20
469/469 [=====] - 5s 11ms/step - loss: 0.0240 - accuracy: 0.993
6 - val_loss: 0.1003 - val_accuracy: 0.9837
Epoch 13/20
469/469 [=====] - 5s 11ms/step - loss: 0.0220 - accuracy: 0.993
6 - val_loss: 0.1131 - val_accuracy: 0.9821
Epoch 14/20
469/469 [=====] - 5s 11ms/step - loss: 0.0204 - accuracy: 0.994
1 - val_loss: 0.1110 - val_accuracy: 0.9826
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 0.0203 - accuracy: 0.994
9 - val_loss: 0.1176 - val_accuracy: 0.9828
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.0168 - accuracy: 0.995
6 - val_loss: 0.1453 - val_accuracy: 0.9819
Epoch 17/20
469/469 [=====] - 5s 11ms/step - loss: 0.0181 - accuracy: 0.995
2 - val_loss: 0.1350 - val_accuracy: 0.9820
Epoch 18/20
469/469 [=====] - 5s 11ms/step - loss: 0.0156 - accuracy: 0.995
6 - val_loss: 0.1681 - val_accuracy: 0.9816
Epoch 19/20
469/469 [=====] - 5s 11ms/step - loss: 0.0178 - accuracy: 0.995
6 - val_loss: 0.1368 - val_accuracy: 0.9836
Epoch 20/20
469/469 [=====] - 5s 11ms/step - loss: 0.0160 - accuracy: 0.995
9 - val_loss: 0.1512 - val_accuracy: 0.9819
Test loss: 0.1512499749660492
Test accuracy: 0.9819000363349915
```

## With Noise

```
In [88]: import numpy as np  
import matplotlib.pyplot as plt
```

```
In [120... (x_train, y_train), (x_test, y_test) = mnist.load_data()  
x_train[2]
```

```
Out[120]: array([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 67, 232, 39, 0, 0, 0],
 [0, 0, 0, 62, 81, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 126, 163, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 220, 163, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 222, 163, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 46, 245, 163, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 120, 254, 163, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 159, 254, 120, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 159, 254, 67, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 159, 254, 85, 0, 0, 0, 47, 49, 116, 144,
 150, 241, 243, 234, 179, 241, 252, 40, 0, 0, 0, 0],
 [0, 0, 0, 150, 253, 237, 207, 207, 207, 253, 254, 250, 240,
 198, 143, 91, 28, 5, 233, 250, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 119, 177, 177, 177, 177, 177, 98, 56, 0,
 0, 0, 0, 102, 254, 220, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 169, 254, 137, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 169, 254, 57, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 169, 254, 57, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```

0, 0, 0, 0, 169, 255, 94, 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, 169, 254, 96, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 169, 254, 153, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 169, 255, 153, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 96, 254, 153, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0]], dtype=uint8)

```

```

In [121... 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

x_train_noise = np.random.normal(x_train)
x_test_noise = np.random.normal(x_test)

print(x_train_noise.shape[0], 'train samples')
print(x_test_noise.shape[0], 'test samples')

60000 train samples
10000 test samples

```

```

In [122... x_train_noise[2]

```

```
Out[122]: array([ 1.22289552e-01,  9.37949546e-01,  1.25482326e+00,  1.63183460e-01,
 2.66003182e+00,  1.23812908e+00,  5.65104027e-01, -2.28444325e-01,
-1.55362444e+00, -4.12486185e-01, -1.64996292e+00,  9.13524556e-01,
 1.41618773e+00,  1.03575641e+00,  3.14567410e-01,  1.15772854e-01,
-5.21060430e-02,  1.03025597e-01,  1.16196142e+00, -4.16944948e-01,
-8.58160319e-01, -3.14299343e-01,  1.69538194e-01, -2.02897417e-01,
 7.52219218e-01,  3.84351218e-01,  6.28754769e-01,  6.24037690e-01,
 2.05509390e+00,  1.53772080e+00, -1.20658422e+00, -3.81010924e-01,
-9.68826727e-01, -1.03803353e-01, -1.57274432e+00, -5.03517550e-01,
-1.35099565e+00, -3.94380860e-01,  1.94724637e+00,  1.28039156e+00,
-1.30575467e+00, -6.09353185e-01, -1.42702398e+00, -1.75340062e+00,
 8.94263061e-02, -1.10475863e+00,  7.52091750e-02,  4.54132398e-01,
-3.88847761e-01, -6.60370366e-03, -1.19085127e+00,  7.47452977e-01,
 1.50777946e+00,  2.56612051e+00,  7.14668830e-01,  5.40708429e-01,
 4.90809318e-01, -9.59630718e-01, -3.84783741e-01,  6.70821237e-01,
 4.50103966e-02,  2.05761685e+00,  9.49105408e-02,  2.75106952e-01,
-4.06956166e-01, -1.20830330e+00,  1.43792538e+00, -1.54013469e+00,
-1.54915389e-01,  1.86150468e+00,  1.23301382e+00,  2.31028404e+00,
-4.06719951e-01, -2.12848076e-02, -4.84698398e-01,  6.66704755e-01,
 3.46299109e-01,  2.04578102e-01, -1.40677942e+00,  1.46109888e+00,
-8.09657962e-01, -6.83461742e-01,  9.00886629e-01, -3.09626213e-01,
 5.15347013e-01, -1.10485398e+00, -8.17067443e-02,  3.62040094e-01,
 7.66799470e-01,  6.01922544e-01,  1.89987300e+00,  5.10782615e-02,
 8.09943318e-01,  4.74408796e-01,  9.75311106e-01, -4.95788918e-01,
-1.24269393e+00,  1.25108186e-01,  1.15263478e-01, -7.22851296e-01,
-1.45259585e+00, -2.56918974e-01,  1.28476277e-01,  3.29572601e-01,
-1.25154630e+00,  3.83214921e-01,  2.43096076e+00, -9.45369357e-01,
 8.79114177e-01,  3.43121437e-01, -8.89952181e-02, -7.32641111e-01,
-8.62839326e-01, -1.62044517e-01,  2.36600333e-01,  4.49692357e-01,
-7.47531841e-01, -1.06258417e-01,  1.46550655e+00,  4.28692442e-01,
-3.60080417e-01,  2.59683340e-01,  1.44973148e+00, -7.17108971e-01,
 9.89038318e-01, -8.67474277e-01,  3.10602142e-01,  3.06758596e-01,
 1.99308580e+00,  9.03267746e-02,  8.64819034e-02, -1.81045498e-01,
-1.04272159e+00,  8.26709073e-02,  8.74912676e-01, -1.62215048e-01,
 7.32367715e-01, -2.47864411e-02, -1.09847638e+00,  8.74890872e-02,
 4.19933596e-01,  1.40705454e+00,  4.21692816e-01, -1.40872423e+00,
-3.91847359e-02, -7.03147469e-01,  9.35502118e-01, -1.14068492e+00,
-1.08323597e+00, -5.92333409e-01,  6.45868835e-01,  4.81187341e-01,
 1.23551213e+00,  4.01181227e-01, -8.54438190e-01,  1.49197382e-01,
-7.02343890e-01, -8.41865809e-01, -1.48111280e+00,  1.65290116e-01,
 2.24890109e+00,  1.31634015e+00,  8.23640752e-01,  9.50992957e-01,
 3.31885730e-01,  9.69300346e-01,  2.55389109e-01,  1.09116412e+00,
 1.02802335e-01,  9.50321189e-01,  4.36730610e-01, -2.71308017e-01,
 4.63192372e-01,  1.06142710e+00,  6.11990641e-02,  1.00224558e-02,
 8.81640051e-01,  5.89292621e-01,  2.04657055e+00,  1.45822744e+00,
-1.62858206e+00,  1.15824603e+00,  1.57465267e+00,  5.67876092e-01,
-5.19805844e-01,  6.98877698e-01,  1.35869526e-02,  2.10695629e-01,
-9.68050980e-02,  1.71209693e+00, -2.01756009e-01, -4.24003607e-01,
 1.21430626e-01,  2.04930196e+00, -7.72385657e-01,  3.40422335e-01,
 4.13593262e-01, -1.03657202e-01,  3.23737564e-01, -2.27492865e+00,
-5.15202062e-01,  4.26197557e-01, -2.28937724e-01,  1.91753732e+00,
-3.68562090e-01, -1.14957386e+00, -4.94111394e-01,  2.89264788e-01,
-1.12982033e+00,  6.73831582e-01,  1.25271454e+00,  1.49935953e+00,
-1.25572380e+00, -1.00309330e+00, -3.24276904e-01, -2.14452615e+00,
 1.04064881e+00,  6.68879149e-01,  2.35008036e-01, -3.17726626e-01,
 3.45489843e-02,  9.68481483e-01,  1.40360949e-01,  2.90192531e-01,
 2.77538956e-01,  1.87538870e+00,  8.31710564e-01,  1.00310948e+00,
 8.29036874e-01,  1.00367927e+00,  1.04253521e+00, -4.60812716e-01,
 1.28570340e+00,  9.00216452e-01, -1.82115098e-01, -2.57324476e-01,
-1.09082782e+00, -1.64269257e+00,  8.73078465e-01, -2.85756420e-01,
-3.39879247e-01,  2.40508597e-01,  7.73682146e-01,  2.00285915e+00,
```



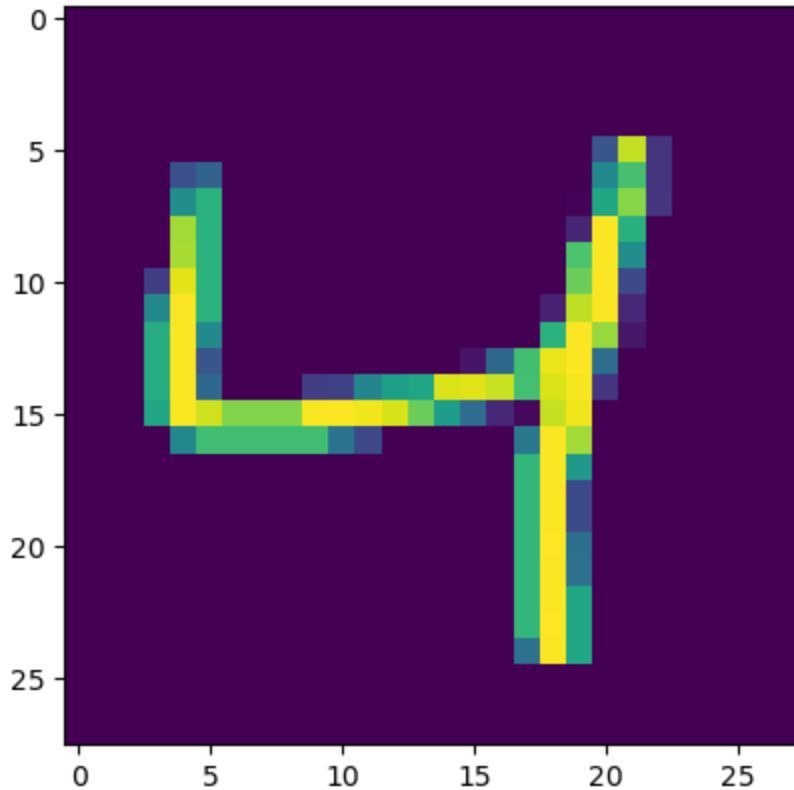
2.18264846e+00,	8.96257599e-01,	2.45655644e+00,	2.83699681e-02,
-1.28058449e+00,	8.84636606e-01,	3.23922017e-01,	1.40301371e+00,
-3.05480732e-01,	9.58030917e-01,	2.99165419e-01,	1.36362443e-01,
5.61652986e-01,	1.62765116e+00,	-6.75803616e-01,	5.40350966e-01,
1.65321806e+00,	1.51587933e+00,	-5.68827409e-01,	-8.52339045e-01,
-2.82518957e-01,	8.10114872e-01,	-9.71028582e-01,	5.20239495e-02,
-2.08649158e+00,	1.98930758e-02,	1.66010485e-01,	1.70169102e+00,
3.57642695e-02,	-3.28009917e-01,	9.03586237e-01,	1.67156234e-01,
-5.97057208e-01,	-1.32779177e+00,	-1.74483877e+00,	-1.59872046e-02,
-5.67440620e-01,	1.09920970e+00,	3.71478799e-01,	5.40809238e-01,
-7.39760670e-01,	6.28322575e-01,	3.38935915e-01,	-1.49959840e-01,
-8.93492927e-02,	1.13468949e+00,	1.36374433e+00,	-9.51020419e-01,
6.84684466e-01,	2.39923285e-01,	1.62834050e-02,	1.67972707e+00,
6.09295996e-01,	-8.12727730e-01,	-1.32659465e+00,	1.04929102e+00,
-4.58774091e-01,	6.54740428e-01,	-2.55473205e+00,	-1.24129052e-01,
-2.86560926e-02,	-1.25265949e-01,	6.19099616e-01,	4.65008995e-01,
-1.56401418e+00,	-3.96554504e-01,	3.38008792e-01,	1.04988356e+00,
9.93021974e-01,	2.60704493e-01,	-1.88352010e+00,	-2.75628418e+00,
-1.63917424e-01,	4.60222871e-01,	-6.39043885e-01,	5.29255716e-01,
1.02193652e+00,	-3.98306499e-01,	-4.31309134e-01,	6.81092953e-02,
-8.75837886e-01,	5.92083644e-01,	-9.73420685e-01,	9.47212254e-01,
1.38888210e+00,	8.49725404e-02,	-9.64770143e-01,	1.31612328e+00,
7.00824409e-01,	-1.90336306e+00,	-9.37977310e-01,	-8.36206586e-01,
1.54694963e-01,	6.02460190e-01,	-3.77538348e-01,	1.28223773e-01,
2.16152446e+00,	-1.81119334e+00,	1.66256551e+00,	8.78766532e-01,
7.21463504e-01,	5.88930681e-01,	-1.85370892e+00,	1.21826103e+00,
5.29345514e-01,	3.41970618e-01,	-1.23292933e+00,	7.37699713e-02,
-1.52182689e-01,	1.09839125e+00,	8.03095421e-01,	-3.80580532e-01,
5.70314977e-02,	-5.02343193e-01,	-4.25476357e-01,	1.59168178e-01,
-2.90456978e-01,	2.23570811e-01,	2.40819833e+00,	4.84580910e-03,
-8.32999165e-01,	-6.71313896e-01,	3.07504643e-01,	7.97723490e-01,
2.29169719e+00,	9.30120850e-01,	1.08074998e-02,	-1.56794905e+00,
7.04551384e-01,	1.55849860e+00,	-6.60681383e-02,	1.40865475e+00,
1.35706005e+00,	-5.80428545e-01,	-2.09143279e+00,	-8.75598134e-01,
-9.71665584e-01,	9.85393668e-02,	1.24846228e+00,	1.26167573e+00,
-6.69802732e-01,	-5.03142689e-01,	-4.45587569e-02,	-1.16006896e+00,
1.44544223e+00,	-9.63402618e-01,	9.67005762e-02,	5.06727078e-01,
-6.19335289e-01,	-4.52951147e-01,	-1.19969017e+00,	4.88178396e-01,
-4.29253339e-02,	1.61345794e+00,	-2.78855841e-01,	-5.52188954e-01,
-1.27660030e+00,	2.18218572e+00,	-8.85300518e-01,	9.90861579e-01,
1.31227172e+00,	-6.11007146e-01,	2.60076740e+00,	2.41751488e+00,
8.50625600e-01,	1.67854229e+00,	1.17429457e-01,	9.23714387e-01,
-4.47498145e-01,	4.55082203e-01,	-1.00240294e-01,	1.51097529e+00,
-3.22423296e-01,	1.78671989e+00,	-1.13058221e+00,	-7.35369316e-01,
4.07941438e-01,	-1.14108496e+00,	2.24596677e+00,	-2.65210705e-02,
1.49031146e+00,	1.89438380e-01,	1.01866035e+00,	-1.93248721e-01,
1.08269545e+00,	2.18956690e+00,	4.78850351e-01,	1.08212399e+00,
2.62648234e+00,	-1.43539588e+00,	-8.06004822e-01,	-2.50378125e-01,
9.43881386e-01,	3.50260012e-01,	2.00854022e+00,	1.45202284e+00,
-2.83732710e+00,	7.32618855e-02,	7.30789216e-01,	1.63860726e-01,
1.41609452e+00,	1.20049447e+00,	-3.13788395e-01,	2.20415132e-02,
1.49218056e+00,	1.23304266e+00,	1.70107630e-01,	7.28607632e-01,
1.54690180e+00,	1.16459597e+00,	2.99340556e-01,	7.54361433e-01,
-2.67512805e-01,	8.20343812e-01,	2.03287731e+00,	-1.31348476e-01,
-1.85270925e+00,	1.50766655e+00,	-1.12855451e+00,	-5.64600876e-01,
1.35274212e+00,	7.13403114e-01,	3.37088724e-01,	1.46569005e+00,
6.67435397e-01,	9.90611415e-01,	1.54746931e+00,	2.43840277e-01,
-1.65423965e+00,	9.89399779e-01,	1.65513470e+00,	6.38358915e-01,
7.03800714e-02,	3.24330383e-01,	-1.09239969e+00,	4.71709641e-01,
7.78098700e-01,	1.30061881e+00,	-2.08880834e-02,	1.58130878e+00,
9.35864091e-01,	-1.21160080e+00,	2.58528013e-01,	-7.60057218e-02,
-7.07078580e-01,	-1.39691407e+00,	1.53224774e+00,	-9.95394642e-01,

-9.67637937e-01, -4.66167269e-01, 1.47633935e+00, 6.21675856e-01,  
-7.28235746e-01, 1.05248861e+00, 5.27371244e-01, 4.32113848e-01,  
1.76520744e+00, -1.34643066e+00, -1.37949707e+00, 1.88489795e+00,  
-2.42484820e-02, 2.22811248e+00, -2.00716729e+00, -9.94668817e-01,  
1.29084072e+00, 1.52289150e+00, 4.24580997e-01, -8.95774393e-01,  
1.14197763e-02, 1.95377644e+00, -3.97666701e-01, 9.25764587e-02,  
-1.43114922e+00, -2.05326221e+00, -4.47498546e-01, 1.45793620e+00,  
9.04558649e-01, 1.35853803e+00, 7.07617672e-01, 2.46426004e-01,  
8.25646575e-01, 1.91440349e+00, 1.21335151e-01, 4.10418344e-01,  
2.32330496e-01, -5.43412655e-01, 1.02270556e+00, 8.73591093e-01,  
-5.27925159e-01, -4.48072492e-01, -3.66393726e-01, -6.70514787e-01,  
-1.69429432e+00, -1.50511332e+00, -1.29971126e+00, 1.49926884e+00,  
1.93730711e+00, 1.45399577e-01, -1.23377466e-01, -1.08232655e+00,  
1.56010331e+00, -2.05670127e+00, -3.59044511e-01, -3.41188634e-01,  
-1.56684667e+00, 2.81373404e+00, 2.22294722e+00, -3.17768650e-01,  
9.10477595e-01, -9.25921229e-01, -6.40564901e-01, 1.09449649e+00,  
-4.57120482e-02, 5.14249743e-01, 2.54579517e-01, -8.17780724e-01,  
-1.82811418e+00, -1.53795429e+00, 6.49577632e-01, 1.74052057e+00,  
-7.25703397e-01, 2.67349229e-01, -6.51786524e-01, 7.79043178e-03,  
2.90278275e-01, -6.10422531e-01, 3.89658274e-01, 1.95100153e+00,  
-8.62415459e-01, -1.41857578e+00, -1.26262585e-02, -9.33689608e-01,  
7.25593785e-01, -8.94930875e-01, -1.74756317e-01, -1.32433548e+00,  
1.32602703e-01, 5.03784794e-02, 1.38448243e-01, 4.63514254e-01,  
1.13294533e+00, -2.27740852e-01, 4.16673536e-02, 2.36472735e-01,  
-1.50908167e+00, -4.05518795e-01, -6.85029416e-01, -3.22181265e-01,  
3.73841598e-02, -1.05014001e+00, -1.20425761e+00, -1.02510603e+00,  
9.58678404e-01, 9.53315169e-01, 1.15408456e+00, -8.17744571e-01,  
-1.12319723e+00, 2.15164927e+00, -1.29854111e+00, 9.70190140e-01,  
-1.49972183e+00, -1.89924762e-01, 2.08664004e-01, -1.94919484e-01,  
9.34236878e-01, -7.03909694e-02, -3.79660028e-01, -1.19022414e-01,  
3.51517327e-01, 1.37035944e-01, -6.23013327e-01, -2.56985138e-01,  
-4.50288843e-01, 1.97515802e+00, -1.02276581e+00, 1.20767295e+00,  
3.35669306e-01, -3.97857589e-01, -7.38132388e-01, 5.52249765e-01,  
-1.94319847e+00, 7.06283473e-01, -2.01513889e-01, 1.61435715e-01,  
1.25333078e+00, -2.99498268e-01, 4.31575341e-01, -1.16735954e+00,  
9.81085381e-01, -2.24802871e-01, -6.44652927e-01, 6.36152982e-01,  
1.27955014e+00, -1.77113241e+00, 1.59011074e+00, 9.61991996e-01,  
-4.60827824e-01, 5.06186786e-01, -7.21569703e-02, 6.02651771e-01,  
7.81105511e-01, 1.51427353e-01, 2.46778765e-01, -4.98391572e-01,  
2.62358465e-01, -6.03302215e-01, -8.73458767e-01, 1.05567681e+00,  
-1.56851580e-01, 1.29020323e-01, -1.33165257e+00, 1.62022895e+00,  
6.22509941e-01, -1.69516677e-01, 1.68961240e-01, -1.13286667e+00,  
1.02811677e+00, -6.59263547e-01, 4.42466187e-02, 2.04327544e+00,  
1.68134480e+00, 5.07330007e-01, -6.48399775e-01, 8.08680939e-01,  
-8.13157957e-03, 4.61076118e-02, -1.64908258e+00, -6.48327866e-01,  
1.83879047e+00, 1.06288394e+00, 1.28851164e+00, 5.86149478e-01,  
1.22647956e+00, -8.37495212e-03, -4.74478257e-01, 1.43004107e+00,  
1.70045152e+00, -2.06887767e+00, -5.74644811e-01, 4.99944797e-01,  
1.35128072e+00, -1.94777337e+00, -7.66115563e-01, 4.99319805e-01,  
5.12065309e-01, -3.63831485e-01, 3.82813623e-01, -1.52103870e-01,  
-2.55306917e-01, -4.58109983e-01, -9.19469708e-01, 9.72778672e-01,  
-3.77819693e-01, -2.43431119e-01, 1.70274425e+00, 2.01990784e+00,  
1.32378192e+00, -3.95671827e-01, -4.67466755e-01, -4.35720398e-01,  
-1.88018731e+00, 3.83706036e-01, 7.92676953e-01, -1.18824877e+00,  
-2.44674779e+00, -2.43625190e+00, -2.68628580e-01, 1.27239965e+00,  
1.50631788e-01, -1.43916701e-01, -1.81859170e+00, 1.84272290e-01,  
7.21056570e-01, -9.47518926e-01, -6.90603513e-01, -2.71649510e-01,  
-1.49361631e-01, -6.74859744e-01, 2.52879328e+00, 1.04043404e+00,  
8.50580174e-01, -6.55539292e-01, 1.55211738e-03, 5.55165307e-01,  
-1.63514644e+00, 1.15672479e+00, 1.68130021e+00, -3.11958151e-01,  
-1.53111285e+00, 1.91849849e-01, -3.32935828e-01, -1.01536907e+00,  
4.26555075e-02, -8.04745482e-01, 4.12738883e-01, 3.08137073e-02,

```
-1.12522993e+00, -1.43700059e+00, -2.31589704e-01, -4.45391244e-01,
-8.25144223e-01, 8.58421394e-01, 1.41441576e+00, 1.09023612e+00,
6.51983097e-01, 1.18739162e-01, -7.92453691e-01, -2.76228613e-01,
-2.01132362e+00, -1.25466408e-01, -7.27036655e-01, -3.84368424e-01,
1.96253570e-01, 6.85296774e-01, -1.98876826e+00, -3.23426971e-01,
7.60924880e-01, 5.01533864e-01, 5.09681289e-02, -1.06775675e-01,
-1.13398861e-01, -2.10595376e+00, -5.80044284e-02, -2.60382391e-02,
8.77931272e-01, -8.45623035e-01, -6.94791917e-02, -1.55839509e+00,
-3.89434417e-02, -5.22312942e-01, -3.36369955e-01, 7.21778807e-01,
5.52268374e-01, 2.28992273e+00, 5.59955225e-01, 5.78782554e-01,
1.52389000e+00, 8.39272966e-01, 5.49536704e-01, -9.38366883e-01])
```

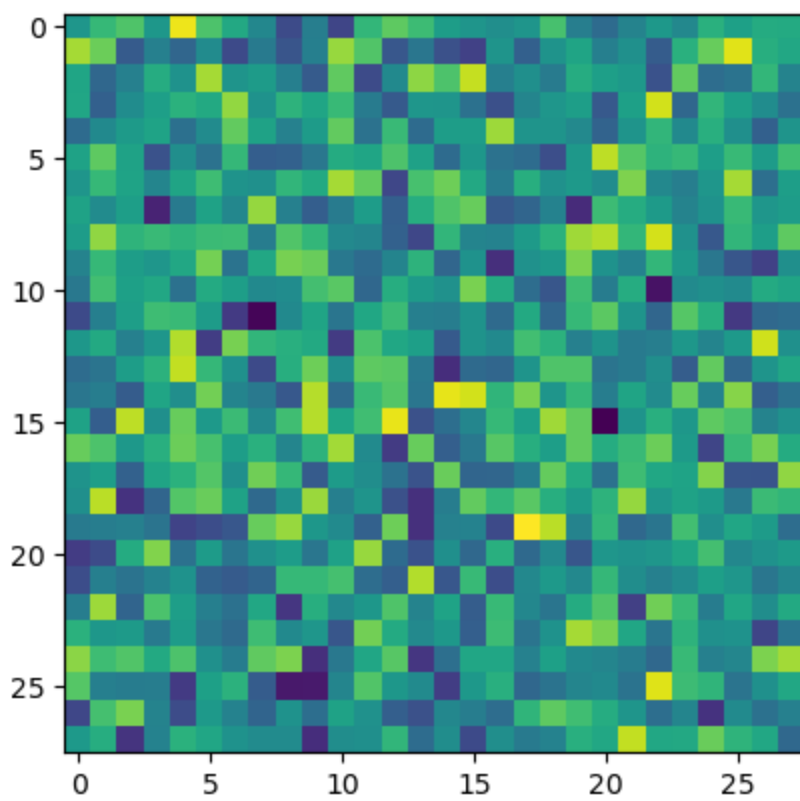
```
In [123]: plt.imshow(x_train[2].reshape(28,28))
```

```
Out[123]: <matplotlib.image.AxesImage at 0x3952cc940>
```



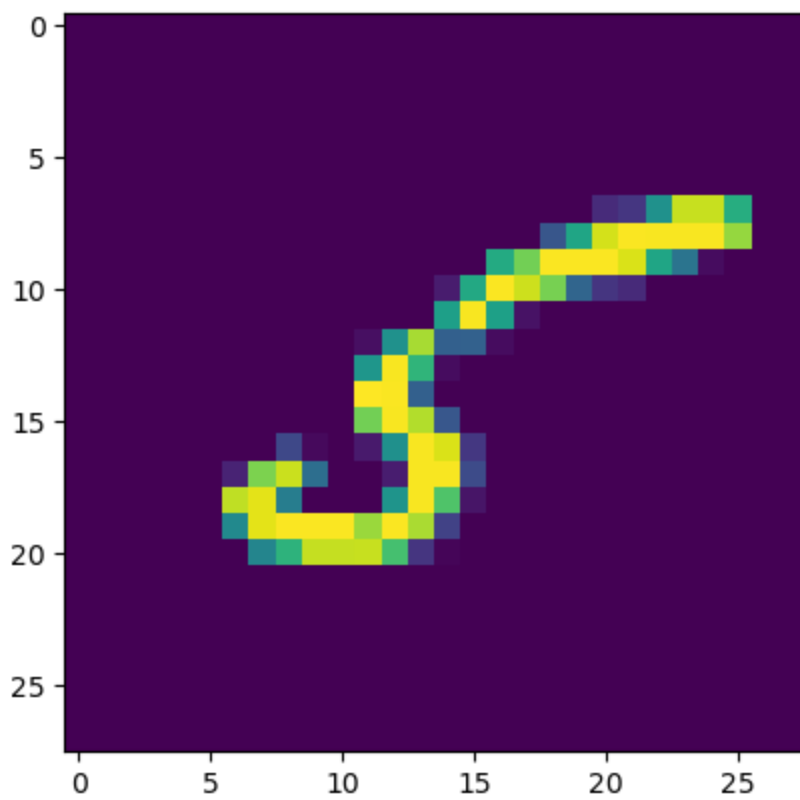
```
In [124]: plt.imshow(x_train_noise[2].reshape(28,28))
```

```
Out[124]: <matplotlib.image.AxesImage at 0x2cd0ad310>
```



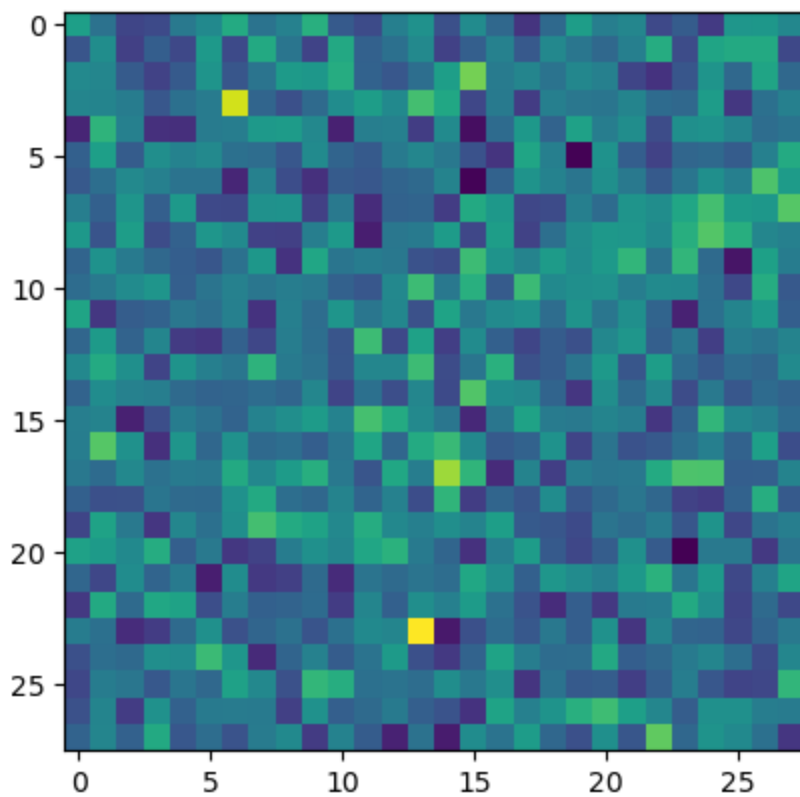
```
In [125... plt.imshow(x_train[11].reshape(28,28))
```

```
Out[125]: <matplotlib.image.AxesImage at 0x2cd9dd820>
```



```
In [126... plt.imshow(x_train_noise[11].reshape(28,28))
```

```
Out[126]: <matplotlib.image.AxesImage at 0x2cbf9d790>
```



```
In [127... batch_size = 128
num_classes = 10
epochs = 20

# 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_noise = model.fit(x_train_noise, y_train,
                          batch_size=batch_size,
                          epochs=epochs,
                          verbose=1,
                          validation_data=(x_test_noise, y_test))
score = model.evaluate(x_test_noise, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

Model: "sequential\_29"

Layer (type)	Output Shape	Param #
dense_87 (Dense)	(None, 512)	401920
dropout_58 (Dropout)	(None, 512)	0
dense_88 (Dense)	(None, 512)	262656
dropout_59 (Dropout)	(None, 512)	0
dense_89 (Dense)	(None, 10)	5130
Total params: 669,706		
Trainable params: 669,706		
Non-trainable params: 0		

Epoch 1/20

2022-11-06 20:22:22.848332: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.

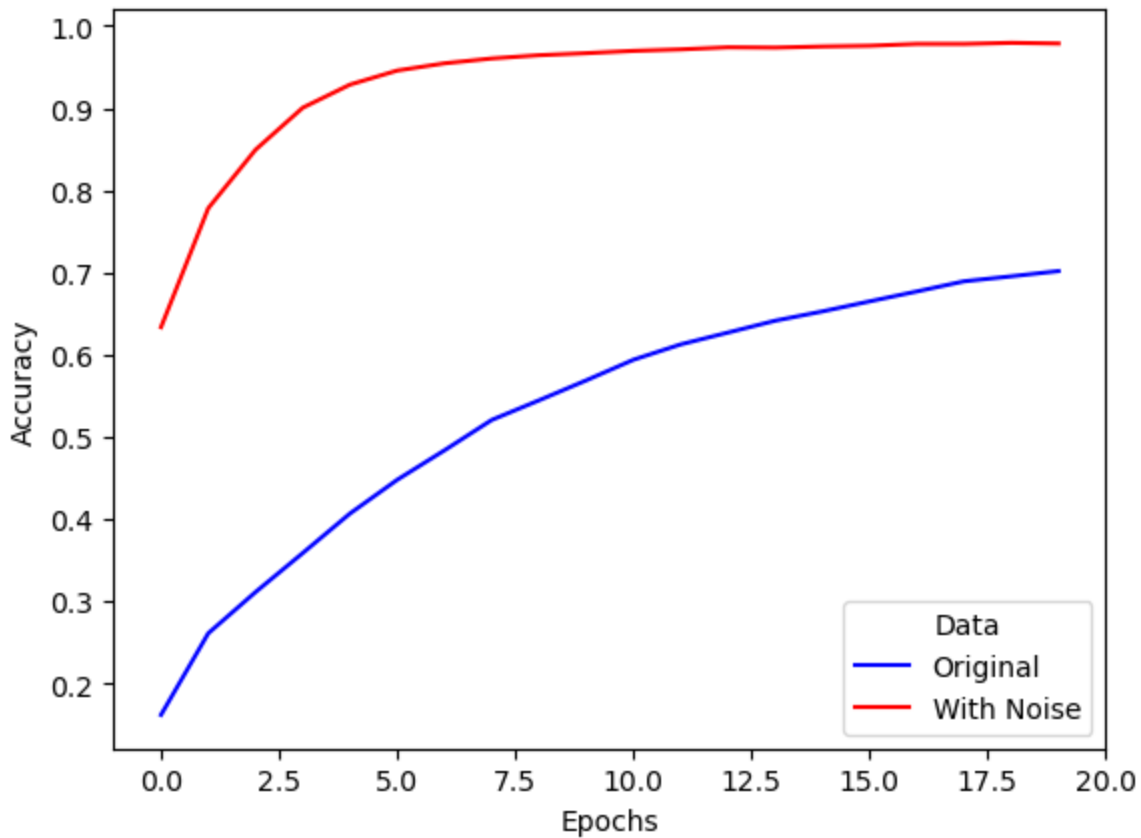
469/469 [=====] - ETA: 0s - loss: 1.0831 - accuracy: 0.6335

2022-11-06 20:22:27.902059: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.

```
469/469 [=====] - 5s 11ms/step - loss: 1.0831 - accuracy: 0.633
5 - val_loss: 0.8155 - val_accuracy: 0.7235
Epoch 2/20
469/469 [=====] - 5s 11ms/step - loss: 0.6630 - accuracy: 0.778
3 - val_loss: 0.7184 - val_accuracy: 0.7635
Epoch 3/20
469/469 [=====] - 5s 11ms/step - loss: 0.4463 - accuracy: 0.849
4 - val_loss: 0.7602 - val_accuracy: 0.7601
Epoch 4/20
469/469 [=====] - 5s 11ms/step - loss: 0.2959 - accuracy: 0.900
5 - val_loss: 0.8732 - val_accuracy: 0.7541
Epoch 5/20
469/469 [=====] - 5s 11ms/step - loss: 0.2077 - accuracy: 0.928
7 - val_loss: 0.9877 - val_accuracy: 0.7560
Epoch 6/20
469/469 [=====] - 5s 11ms/step - loss: 0.1598 - accuracy: 0.945
8 - val_loss: 1.1200 - val_accuracy: 0.7511
Epoch 7/20
469/469 [=====] - 5s 11ms/step - loss: 0.1330 - accuracy: 0.954
6 - val_loss: 1.2151 - val_accuracy: 0.7544
Epoch 8/20
469/469 [=====] - 5s 11ms/step - loss: 0.1172 - accuracy: 0.960
4 - val_loss: 1.3201 - val_accuracy: 0.7549
Epoch 9/20
469/469 [=====] - 5s 11ms/step - loss: 0.1095 - accuracy: 0.964
4 - val_loss: 1.3898 - val_accuracy: 0.7504
Epoch 10/20
469/469 [=====] - 5s 11ms/step - loss: 0.1018 - accuracy: 0.966
9 - val_loss: 1.4722 - val_accuracy: 0.7538
Epoch 11/20
469/469 [=====] - 5s 11ms/step - loss: 0.0936 - accuracy: 0.969
7 - val_loss: 1.5332 - val_accuracy: 0.7538
Epoch 12/20
469/469 [=====] - 5s 11ms/step - loss: 0.0888 - accuracy: 0.971
5 - val_loss: 1.5908 - val_accuracy: 0.7576
Epoch 13/20
469/469 [=====] - 5s 11ms/step - loss: 0.0820 - accuracy: 0.974
0 - val_loss: 1.7233 - val_accuracy: 0.7539
Epoch 14/20
469/469 [=====] - 5s 11ms/step - loss: 0.0845 - accuracy: 0.973
7 - val_loss: 1.6860 - val_accuracy: 0.7578
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 0.0795 - accuracy: 0.975
0 - val_loss: 1.8100 - val_accuracy: 0.7578
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.0786 - accuracy: 0.975
9 - val_loss: 1.8647 - val_accuracy: 0.7545
Epoch 17/20
469/469 [=====] - 5s 11ms/step - loss: 0.0728 - accuracy: 0.978
2 - val_loss: 1.9180 - val_accuracy: 0.7614
Epoch 18/20
469/469 [=====] - 5s 11ms/step - loss: 0.0747 - accuracy: 0.978
1 - val_loss: 1.9488 - val_accuracy: 0.7589
Epoch 19/20
469/469 [=====] - 5s 11ms/step - loss: 0.0691 - accuracy: 0.979
5 - val_loss: 2.0260 - val_accuracy: 0.7575
Epoch 20/20
469/469 [=====] - 5s 11ms/step - loss: 0.0726 - accuracy: 0.978
7 - val_loss: 2.1107 - val_accuracy: 0.7549
Test loss: 2.1107349395751953
Test accuracy: 0.7549000382423401
```

In [139...

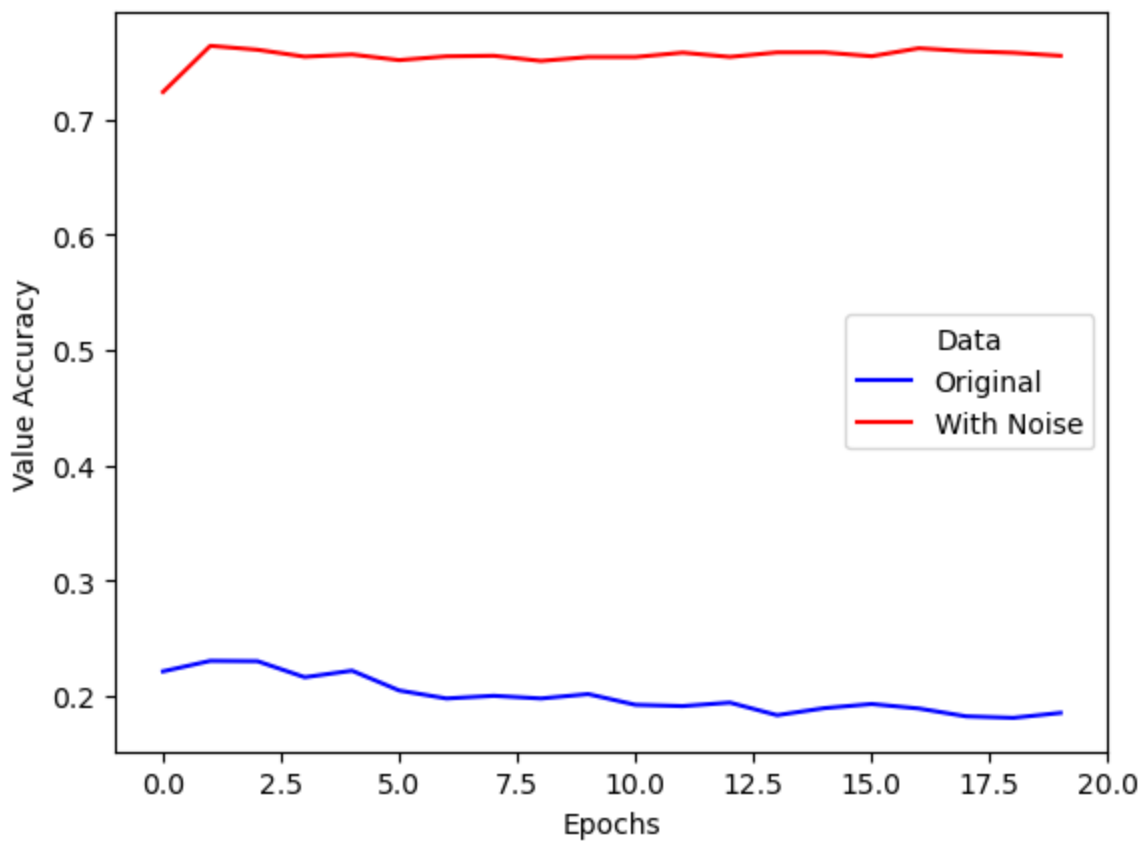
```
plt.plot(history.history['accuracy'],c='b',label='Original')
plt.plot(history_noise.history['accuracy'],c='red',label='With Noise')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.xlim(-1,20)
plt.legend(title='Data')
plt.show()
```



In [138...

```
plt.plot(history.history['val_accuracy'],c='b',label='Original')
plt.plot(history_noise.history['val_accuracy'],c='red',label='With Noise')
plt.ylabel('Value Accuracy')
plt.xlabel('Epochs')
plt.xlim(-1,20)
plt.legend(title='Data')
plt.show()
```





## Varying scales

```
In [130... scales = [.1, .5, 1.0, 2.0, 4.0]
accuracies = []
vals = []

for scale in scales:
    (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

    x_train = np.random.normal(x_train, scale=scale)
    x_test = np.random.normal(x_test, scale=scale)

    batch_size = 128
    num_classes = 10
    epochs = 20
    # 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_noise, y_test, verbose=0)

accuracies.append(history.history['accuracy'])
vals.append(history.history['val_accuracy'])

```

Model: "sequential\_30"

Layer (type)	Output Shape	Param #
dense_90 (Dense)	(None, 512)	401920
dropout_60 (Dropout)	(None, 512)	0
dense_91 (Dense)	(None, 512)	262656
dropout_61 (Dropout)	(None, 512)	0
dense_92 (Dense)	(None, 10)	5130

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

Epoch 1/20

1/469 [.....] - ETA: 2:31 - loss: 2.3611 - accuracy: 0.0625

2022-11-06 20:28:17.200792: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.

469/469 [=====] - ETA: 0s - loss: 0.2609 - accuracy: 0.9201

2022-11-06 20:28:22.184542: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.

```

469/469 [=====] - 6s 11ms/step - loss: 0.2609 - accuracy: 0.920
1 - val_loss: 0.1267 - val_accuracy: 0.9604
Epoch 2/20
469/469 [=====] - 5s 11ms/step - loss: 0.0983 - accuracy: 0.969
3 - val_loss: 0.1036 - val_accuracy: 0.9642
Epoch 3/20
469/469 [=====] - 5s 11ms/step - loss: 0.0590 - accuracy: 0.981
3 - val_loss: 0.1026 - val_accuracy: 0.9739
Epoch 4/20
469/469 [=====] - 5s 11ms/step - loss: 0.0400 - accuracy: 0.987
6 - val_loss: 0.1160 - val_accuracy: 0.9729
Epoch 5/20
469/469 [=====] - 5s 11ms/step - loss: 0.0307 - accuracy: 0.990
2 - val_loss: 0.1170 - val_accuracy: 0.9742
Epoch 6/20
469/469 [=====] - 5s 11ms/step - loss: 0.0217 - accuracy: 0.993
0 - val_loss: 0.1351 - val_accuracy: 0.9731
Epoch 7/20
469/469 [=====] - 5s 11ms/step - loss: 0.0194 - accuracy: 0.994
1 - val_loss: 0.1665 - val_accuracy: 0.9734
Epoch 8/20
469/469 [=====] - 5s 11ms/step - loss: 0.0183 - accuracy: 0.994
6 - val_loss: 0.1526 - val_accuracy: 0.9737
Epoch 9/20
469/469 [=====] - 6s 12ms/step - loss: 0.0153 - accuracy: 0.995
6 - val_loss: 0.1671 - val_accuracy: 0.9741
Epoch 10/20
469/469 [=====] - 5s 11ms/step - loss: 0.0125 - accuracy: 0.996
3 - val_loss: 0.1922 - val_accuracy: 0.9754
Epoch 11/20
469/469 [=====] - 5s 11ms/step - loss: 0.0119 - accuracy: 0.996
3 - val_loss: 0.1873 - val_accuracy: 0.9743
Epoch 12/20
469/469 [=====] - 5s 11ms/step - loss: 0.0122 - accuracy: 0.996
4 - val_loss: 0.2039 - val_accuracy: 0.9760
Epoch 13/20
469/469 [=====] - 5s 11ms/step - loss: 0.0107 - accuracy: 0.997
0 - val_loss: 0.2190 - val_accuracy: 0.9736
Epoch 14/20
469/469 [=====] - 5s 11ms/step - loss: 0.0090 - accuracy: 0.997
3 - val_loss: 0.2350 - val_accuracy: 0.9735
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 0.0095 - accuracy: 0.997
4 - val_loss: 0.2286 - val_accuracy: 0.9750
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.0100 - accuracy: 0.997
4 - val_loss: 0.2220 - val_accuracy: 0.9764
Epoch 17/20
469/469 [=====] - 5s 11ms/step - loss: 0.0087 - accuracy: 0.997
7 - val_loss: 0.2528 - val_accuracy: 0.9752
Epoch 18/20
469/469 [=====] - 5s 11ms/step - loss: 0.0093 - accuracy: 0.997
4 - val_loss: 0.2513 - val_accuracy: 0.9741
Epoch 19/20
469/469 [=====] - 5s 11ms/step - loss: 0.0081 - accuracy: 0.998
0 - val_loss: 0.2660 - val_accuracy: 0.9755
Epoch 20/20
469/469 [=====] - 5s 11ms/step - loss: 0.0086 - accuracy: 0.997
7 - val_loss: 0.2659 - val_accuracy: 0.9742
Model: "sequential_31"

```

---

Layer (type)

Output Shape

Param #

dense_93 (Dense)	(None, 512)	401920
dropout_62 (Dropout)	(None, 512)	0
dense_94 (Dense)	(None, 512)	262656
dropout_63 (Dropout)	(None, 512)	0
dense_95 (Dense)	(None, 10)	5130

```

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

```

---

Epoch 1/20

```

2022-11-06 20:30:05.851009: I tensorflow/core/grappler/optimizers/custom_graph_optimizer
_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
469/469 [=====] - ETA: 0s - loss: 0.5430 - accuracy: 0.8232
2022-11-06 20:30:11.026604: I tensorflow/core/grappler/optimizers/custom_graph_optimizer
_registry.cc:114] Plugin optimizer for device_type GPU is enabled.

```

```

469/469 [=====] - 6s 12ms/step - loss: 0.5430 - accuracy: 0.823
2 - val_loss: 0.3222 - val_accuracy: 0.8943
Epoch 2/20
469/469 [=====] - 5s 11ms/step - loss: 0.2372 - accuracy: 0.925
8 - val_loss: 0.2819 - val_accuracy: 0.9117
Epoch 3/20
469/469 [=====] - 5s 11ms/step - loss: 0.1235 - accuracy: 0.958
4 - val_loss: 0.3077 - val_accuracy: 0.9149
Epoch 4/20
469/469 [=====] - 5s 11ms/step - loss: 0.0684 - accuracy: 0.977
7 - val_loss: 0.3625 - val_accuracy: 0.9139
Epoch 5/20
469/469 [=====] - 5s 11ms/step - loss: 0.0485 - accuracy: 0.983
5 - val_loss: 0.4249 - val_accuracy: 0.9129
Epoch 6/20
469/469 [=====] - 5s 11ms/step - loss: 0.0383 - accuracy: 0.986
9 - val_loss: 0.4637 - val_accuracy: 0.9143
Epoch 7/20
469/469 [=====] - 5s 11ms/step - loss: 0.0316 - accuracy: 0.989
3 - val_loss: 0.5539 - val_accuracy: 0.9096
Epoch 8/20
469/469 [=====] - 6s 13ms/step - loss: 0.0321 - accuracy: 0.989
6 - val_loss: 0.5289 - val_accuracy: 0.9153
Epoch 9/20
469/469 [=====] - 5s 11ms/step - loss: 0.0288 - accuracy: 0.990
7 - val_loss: 0.5721 - val_accuracy: 0.9147
Epoch 10/20
469/469 [=====] - 5s 12ms/step - loss: 0.0256 - accuracy: 0.991
3 - val_loss: 0.5939 - val_accuracy: 0.9144
Epoch 11/20
469/469 [=====] - 5s 12ms/step - loss: 0.0218 - accuracy: 0.993
1 - val_loss: 0.6941 - val_accuracy: 0.9134
Epoch 12/20
469/469 [=====] - 6s 12ms/step - loss: 0.0218 - accuracy: 0.993
7 - val_loss: 0.6806 - val_accuracy: 0.9174
Epoch 13/20
469/469 [=====] - 5s 11ms/step - loss: 0.0201 - accuracy: 0.993
8 - val_loss: 0.7130 - val_accuracy: 0.9162
Epoch 14/20
469/469 [=====] - 5s 11ms/step - loss: 0.0207 - accuracy: 0.993
7 - val_loss: 0.7420 - val_accuracy: 0.9172
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 0.0201 - accuracy: 0.994
5 - val_loss: 0.7711 - val_accuracy: 0.9137
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.0195 - accuracy: 0.994
3 - val_loss: 0.7422 - val_accuracy: 0.9190
Epoch 17/20
469/469 [=====] - 5s 11ms/step - loss: 0.0157 - accuracy: 0.995
3 - val_loss: 0.7749 - val_accuracy: 0.9172
Epoch 18/20
469/469 [=====] - 5s 11ms/step - loss: 0.0180 - accuracy: 0.994
7 - val_loss: 0.8248 - val_accuracy: 0.9176
Epoch 19/20
469/469 [=====] - 5s 11ms/step - loss: 0.0179 - accuracy: 0.995
1 - val_loss: 0.7848 - val_accuracy: 0.9237
Epoch 20/20
469/469 [=====] - 5s 11ms/step - loss: 0.0157 - accuracy: 0.995
7 - val_loss: 0.8265 - val_accuracy: 0.9202
Model: "sequential_32"

```

---

Layer (type)

Output Shape

Param #

dense_96 (Dense)	(None, 512)	401920
dropout_64 (Dropout)	(None, 512)	0
dense_97 (Dense)	(None, 512)	262656
dropout_65 (Dropout)	(None, 512)	0
dense_98 (Dense)	(None, 10)	5130

```

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

```

---

Epoch 1/20

```

2022-11-06 20:31:56.258092: I tensorflow/core/grappler/optimizers/custom_graph_optimizer
_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
469/469 [=====] - ETA: 0s - loss: 1.0770 - accuracy: 0.6356
2022-11-06 20:32:01.551294: I tensorflow/core/grappler/optimizers/custom_graph_optimizer
_registry.cc:114] Plugin optimizer for device_type GPU is enabled.

```

```

469/469 [=====] - 6s 12ms/step - loss: 1.0770 - accuracy: 0.635
6 - val_loss: 0.8246 - val_accuracy: 0.7266
Epoch 2/20
469/469 [=====] - 5s 11ms/step - loss: 0.6665 - accuracy: 0.776
1 - val_loss: 0.7537 - val_accuracy: 0.7516
Epoch 3/20
469/469 [=====] - 5s 11ms/step - loss: 0.4492 - accuracy: 0.847
2 - val_loss: 0.7612 - val_accuracy: 0.7578
Epoch 4/20
469/469 [=====] - 5s 11ms/step - loss: 0.2977 - accuracy: 0.898
6 - val_loss: 0.8683 - val_accuracy: 0.7586
Epoch 5/20
469/469 [=====] - 5s 11ms/step - loss: 0.1997 - accuracy: 0.931
1 - val_loss: 1.0700 - val_accuracy: 0.7537
Epoch 6/20
469/469 [=====] - 5s 11ms/step - loss: 0.1554 - accuracy: 0.946
2 - val_loss: 1.1856 - val_accuracy: 0.7503
Epoch 7/20
469/469 [=====] - 5s 11ms/step - loss: 0.1356 - accuracy: 0.954
1 - val_loss: 1.2733 - val_accuracy: 0.7516
Epoch 8/20
469/469 [=====] - 5s 11ms/step - loss: 0.1177 - accuracy: 0.960
0 - val_loss: 1.4104 - val_accuracy: 0.7448
Epoch 9/20
469/469 [=====] - 5s 11ms/step - loss: 0.1080 - accuracy: 0.963
4 - val_loss: 1.4598 - val_accuracy: 0.7504
Epoch 10/20
469/469 [=====] - 5s 11ms/step - loss: 0.1023 - accuracy: 0.966
8 - val_loss: 1.5741 - val_accuracy: 0.7529
Epoch 11/20
469/469 [=====] - 5s 11ms/step - loss: 0.0943 - accuracy: 0.969
9 - val_loss: 1.5956 - val_accuracy: 0.7517
Epoch 12/20
469/469 [=====] - 5s 11ms/step - loss: 0.0881 - accuracy: 0.971
1 - val_loss: 1.6693 - val_accuracy: 0.7502
Epoch 13/20
469/469 [=====] - 5s 11ms/step - loss: 0.0868 - accuracy: 0.972
7 - val_loss: 1.7562 - val_accuracy: 0.7483
Epoch 14/20
469/469 [=====] - 5s 11ms/step - loss: 0.0821 - accuracy: 0.974
2 - val_loss: 1.8092 - val_accuracy: 0.7465
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 0.0765 - accuracy: 0.976
0 - val_loss: 1.8871 - val_accuracy: 0.7507
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.0752 - accuracy: 0.976
6 - val_loss: 1.9469 - val_accuracy: 0.7539
Epoch 17/20
469/469 [=====] - 5s 11ms/step - loss: 0.0737 - accuracy: 0.977
7 - val_loss: 1.9877 - val_accuracy: 0.7485
Epoch 18/20
469/469 [=====] - 5s 11ms/step - loss: 0.0714 - accuracy: 0.978
5 - val_loss: 2.0140 - val_accuracy: 0.7493
Epoch 19/20
469/469 [=====] - 5s 11ms/step - loss: 0.0728 - accuracy: 0.978
8 - val_loss: 2.0763 - val_accuracy: 0.7494
Epoch 20/20
469/469 [=====] - 5s 11ms/step - loss: 0.0640 - accuracy: 0.980
4 - val_loss: 2.1983 - val_accuracy: 0.7472
Model: "sequential_33"

```

---

Layer (type)

Output Shape

Param #

dense_99 (Dense)	(None, 512)	401920
dropout_66 (Dropout)	(None, 512)	0
dense_100 (Dense)	(None, 512)	262656
dropout_67 (Dropout)	(None, 512)	0
dense_101 (Dense)	(None, 10)	5130

```

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

```

---

Epoch 1/20

```

2022-11-06 20:33:45.149831: I tensorflow/core/grappler/optimizers/custom_graph_optimizer
_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
469/469 [=====] - ETA: 0s - loss: 1.8757 - accuracy: 0.3575
2022-11-06 20:33:50.338433: I tensorflow/core/grappler/optimizers/custom_graph_optimizer
_registry.cc:114] Plugin optimizer for device_type GPU is enabled.

```



```

469/469 [=====] - 6s 12ms/step - loss: 1.8757 - accuracy: 0.357
5 - val_loss: 1.6040 - val_accuracy: 0.4425
Epoch 2/20
469/469 [=====] - 5s 11ms/step - loss: 1.4649 - accuracy: 0.496
0 - val_loss: 1.5831 - val_accuracy: 0.4463
Epoch 3/20
469/469 [=====] - 5s 11ms/step - loss: 1.2627 - accuracy: 0.565
0 - val_loss: 1.6168 - val_accuracy: 0.4446
Epoch 4/20
469/469 [=====] - 5s 11ms/step - loss: 1.0621 - accuracy: 0.632
3 - val_loss: 1.6911 - val_accuracy: 0.4354
Epoch 5/20
469/469 [=====] - 5s 11ms/step - loss: 0.8884 - accuracy: 0.694
6 - val_loss: 1.8169 - val_accuracy: 0.4317
Epoch 6/20
469/469 [=====] - 5s 11ms/step - loss: 0.7546 - accuracy: 0.739
3 - val_loss: 1.9546 - val_accuracy: 0.4201
Epoch 7/20
469/469 [=====] - 5s 11ms/step - loss: 0.6426 - accuracy: 0.776
1 - val_loss: 2.0978 - val_accuracy: 0.4203
Epoch 8/20
469/469 [=====] - 5s 11ms/step - loss: 0.5708 - accuracy: 0.803
4 - val_loss: 2.2288 - val_accuracy: 0.4170
Epoch 9/20
469/469 [=====] - 5s 11ms/step - loss: 0.5187 - accuracy: 0.820
5 - val_loss: 2.3700 - val_accuracy: 0.4138
Epoch 10/20
469/469 [=====] - 5s 11ms/step - loss: 0.4673 - accuracy: 0.840
5 - val_loss: 2.5076 - val_accuracy: 0.4115
Epoch 11/20
469/469 [=====] - 5s 11ms/step - loss: 0.4387 - accuracy: 0.851
1 - val_loss: 2.6452 - val_accuracy: 0.4152
Epoch 12/20
469/469 [=====] - 5s 11ms/step - loss: 0.4077 - accuracy: 0.861
2 - val_loss: 2.8045 - val_accuracy: 0.4139
Epoch 13/20
469/469 [=====] - 5s 11ms/step - loss: 0.3938 - accuracy: 0.867
3 - val_loss: 2.8839 - val_accuracy: 0.4085
Epoch 14/20
469/469 [=====] - 5s 11ms/step - loss: 0.3764 - accuracy: 0.875
8 - val_loss: 2.9741 - val_accuracy: 0.4103
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 0.3588 - accuracy: 0.880
4 - val_loss: 3.1129 - val_accuracy: 0.4065
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.3518 - accuracy: 0.885
1 - val_loss: 3.2472 - val_accuracy: 0.4021
Epoch 17/20
469/469 [=====] - 5s 11ms/step - loss: 0.3352 - accuracy: 0.891
6 - val_loss: 3.2669 - val_accuracy: 0.3998
Epoch 18/20
469/469 [=====] - 5s 11ms/step - loss: 0.3261 - accuracy: 0.897
1 - val_loss: 3.4021 - val_accuracy: 0.4065
Epoch 19/20
469/469 [=====] - 5s 11ms/step - loss: 0.3221 - accuracy: 0.898
3 - val_loss: 3.4084 - val_accuracy: 0.4065
Epoch 20/20
469/469 [=====] - 5s 11ms/step - loss: 0.3089 - accuracy: 0.902
7 - val_loss: 3.6304 - val_accuracy: 0.4056
Model: "sequential_34"

```

---

Layer (type)

Output Shape

Param #

dense_102 (Dense)	(None, 512)	401920
dropout_68 (Dropout)	(None, 512)	0
dense_103 (Dense)	(None, 512)	262656
dropout_69 (Dropout)	(None, 512)	0
dense_104 (Dense)	(None, 10)	5130

```

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

```

---

Epoch 1/20

```

2022-11-06 20:35:33.396996: I tensorflow/core/grappler/optimizers/custom_graph_optimizer
_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
469/469 [=====] - ETA: 0s - loss: 2.4128 - accuracy: 0.1618
2022-11-06 20:35:38.627213: I tensorflow/core/grappler/optimizers/custom_graph_optimizer
_registry.cc:114] Plugin optimizer for device_type GPU is enabled.

```

```

469/469 [=====] - 5s 12ms/step - loss: 2.4128 - accuracy: 0.161
8 - val_loss: 2.1558 - val_accuracy: 0.2211
Epoch 2/20
469/469 [=====] - 5s 11ms/step - loss: 2.0679 - accuracy: 0.261
0 - val_loss: 2.1365 - val_accuracy: 0.2302
Epoch 3/20
469/469 [=====] - 5s 11ms/step - loss: 1.9355 - accuracy: 0.311
1 - val_loss: 2.1508 - val_accuracy: 0.2299
Epoch 4/20
469/469 [=====] - 5s 11ms/step - loss: 1.8047 - accuracy: 0.358
6 - val_loss: 2.1962 - val_accuracy: 0.2160
Epoch 5/20
469/469 [=====] - 5s 11ms/step - loss: 1.6830 - accuracy: 0.406
7 - val_loss: 2.2615 - val_accuracy: 0.2217
Epoch 6/20
469/469 [=====] - 5s 11ms/step - loss: 1.5700 - accuracy: 0.447
8 - val_loss: 2.3431 - val_accuracy: 0.2045
Epoch 7/20
469/469 [=====] - 5s 11ms/step - loss: 1.4728 - accuracy: 0.483
6 - val_loss: 2.4042 - val_accuracy: 0.1976
Epoch 8/20
469/469 [=====] - 5s 11ms/step - loss: 1.3775 - accuracy: 0.520
6 - val_loss: 2.5135 - val_accuracy: 0.1999
Epoch 9/20
469/469 [=====] - 5s 11ms/step - loss: 1.3032 - accuracy: 0.544
4 - val_loss: 2.5508 - val_accuracy: 0.1976
Epoch 10/20
469/469 [=====] - 5s 11ms/step - loss: 1.2414 - accuracy: 0.568
7 - val_loss: 2.6785 - val_accuracy: 0.2014
Epoch 11/20
469/469 [=====] - 5s 11ms/step - loss: 1.1798 - accuracy: 0.593
9 - val_loss: 2.7600 - val_accuracy: 0.1921
Epoch 12/20
469/469 [=====] - 5s 11ms/step - loss: 1.1278 - accuracy: 0.612
5 - val_loss: 2.8840 - val_accuracy: 0.1910
Epoch 13/20
469/469 [=====] - 5s 11ms/step - loss: 1.0886 - accuracy: 0.626
9 - val_loss: 2.9658 - val_accuracy: 0.1940
Epoch 14/20
469/469 [=====] - 5s 11ms/step - loss: 1.0557 - accuracy: 0.641
4 - val_loss: 3.0084 - val_accuracy: 0.1831
Epoch 15/20
469/469 [=====] - 5s 11ms/step - loss: 1.0236 - accuracy: 0.652
7 - val_loss: 3.0336 - val_accuracy: 0.1892
Epoch 16/20
469/469 [=====] - 5s 11ms/step - loss: 0.9866 - accuracy: 0.664
9 - val_loss: 3.1815 - val_accuracy: 0.1928
Epoch 17/20
469/469 [=====] - 5s 11ms/step - loss: 0.9570 - accuracy: 0.677
0 - val_loss: 3.2386 - val_accuracy: 0.1889
Epoch 18/20
469/469 [=====] - 5s 11ms/step - loss: 0.9284 - accuracy: 0.689
3 - val_loss: 3.2799 - val_accuracy: 0.1821
Epoch 19/20
469/469 [=====] - 5s 11ms/step - loss: 0.9144 - accuracy: 0.695
4 - val_loss: 3.3267 - val_accuracy: 0.1808
Epoch 20/20
469/469 [=====] - 5s 11ms/step - loss: 0.8940 - accuracy: 0.701
8 - val_loss: 3.4046 - val_accuracy: 0.1851

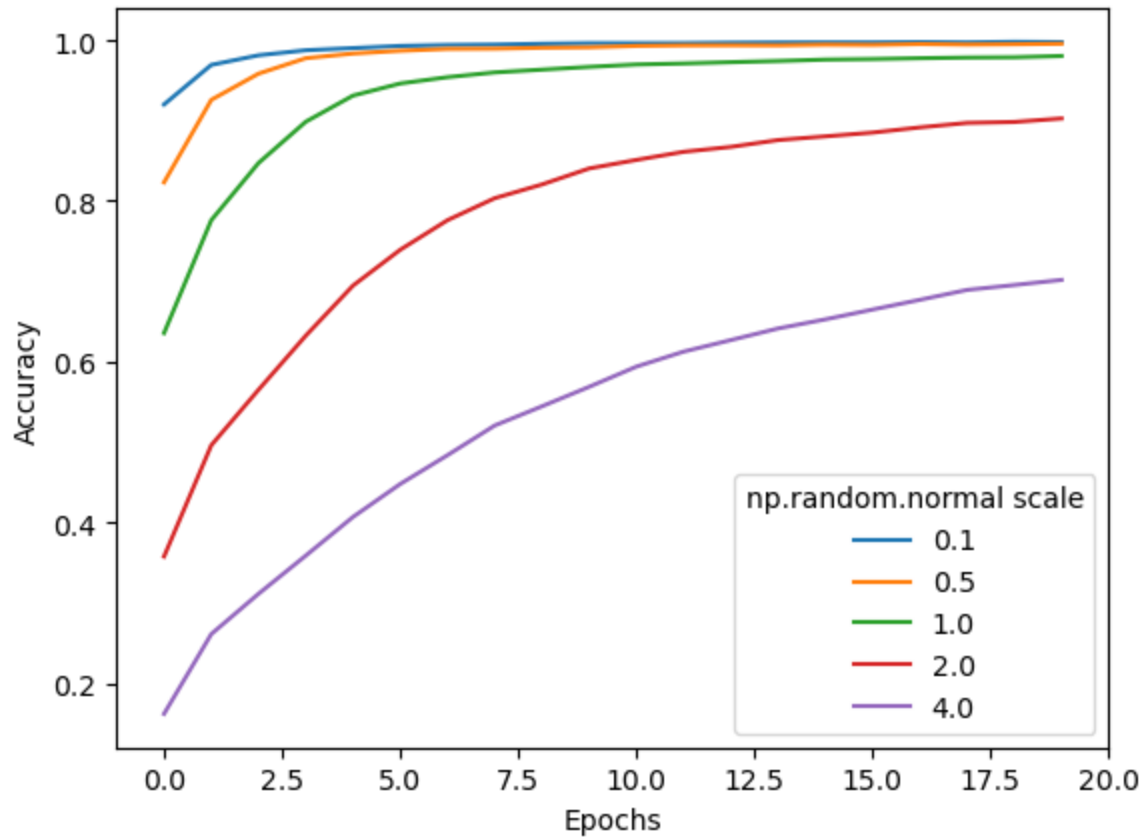
```

```

In [136... for i in range(0, len(accuracies)):
            plt.plot(accuracies[i], label=scales[i])

```

```
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.xlim(-1,20)
plt.legend(title='np.random.normal scale')
plt.show()
```



```
In [137... for i in range(0,len(vals)):
    plt.plot(vals[i],label=scales[i])
plt.ylabel('Value Accuracy')
plt.xlabel('Epochs')
plt.xlim(-1,20)
plt.legend(title='np.random.normal scale')
plt.show()
```

