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.

Neural Networks - Image Recognition

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import tensorflow as tf
import keras
from keras.datasets import mnist
from keras.models import Sequential
from tensorflow.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

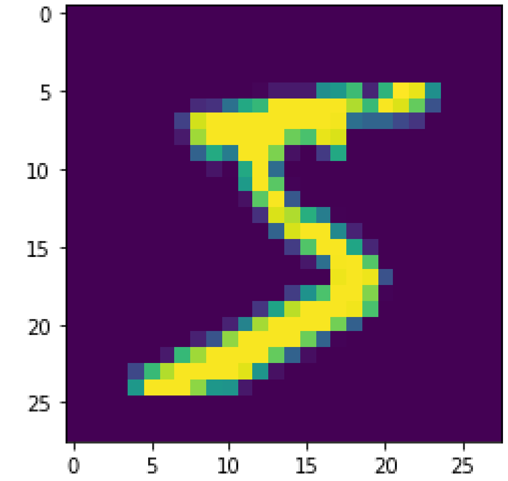
Trains a simple deep NN on the MNIST dataset. Gets to 98.40% test accuracy after 20 epochs (there is *a lot* of margin for parameter tuning).

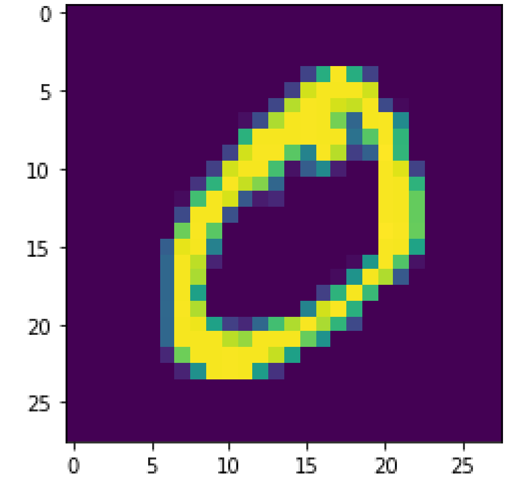
In [2]:

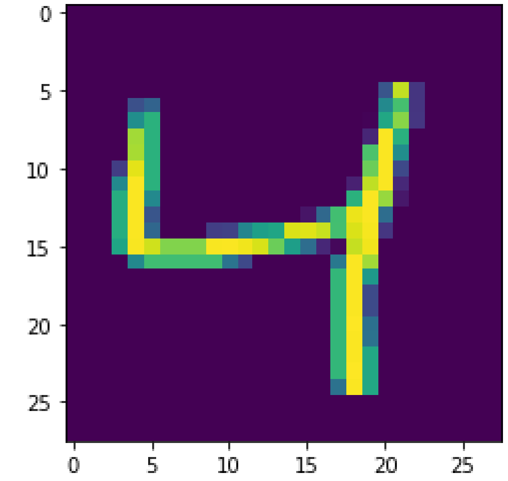
```
# the data, shuffled and split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

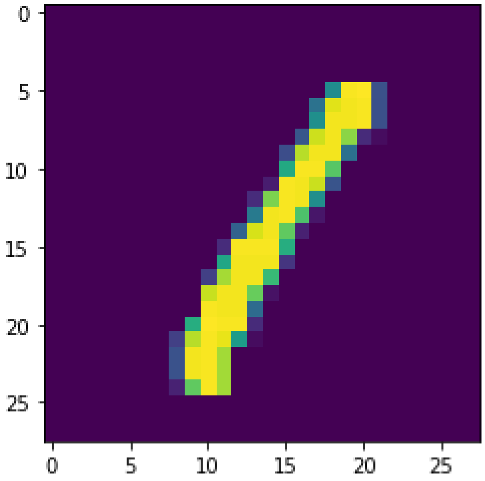
In [3]:

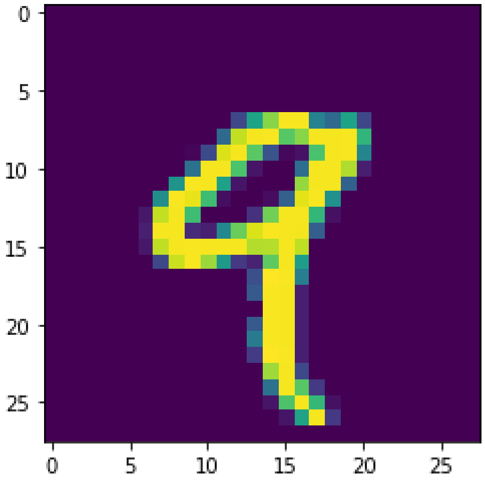
```
images = [-1, 0, 1, 2, 3, 4]
for i in images:
    plt.imshow(x_train[i])
    plt.figure(i + 1)
```


0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25





```
<Figure size 432x288 with 0 Axes>
```

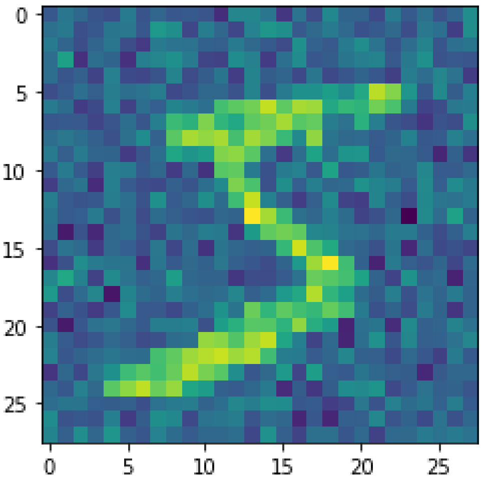
In [4]:
```
print(y_train[0])
print(y_train[1])
print(y_train[2])
print(y_train[3])
print(y_train[4])
```

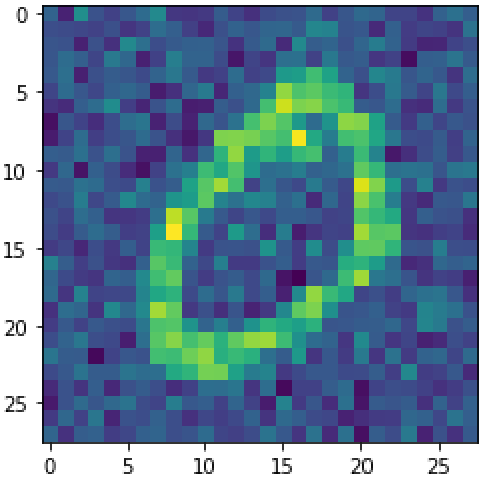
```
5
0
4
1
9
```

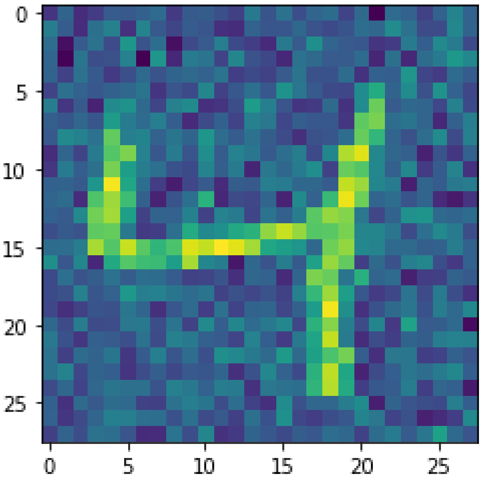
1) Add random noise (see below on size parameter on np.random.normal) to the images in training and testing. *Make sure each image gets a different noise feature added to it. Inspect by printing out several images. Note - the size parameter should match the data.*

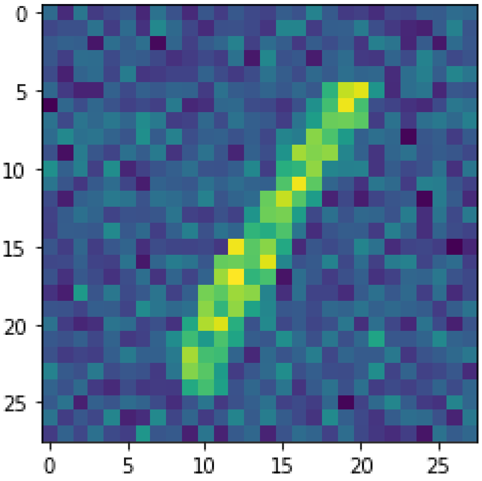
In [5]:
```
x_train_noise = x_train + np.random.normal(scale = 50, size = x_train.shape)
# Setting scale to 50 appears to "mildly to heavily" distort the image
```

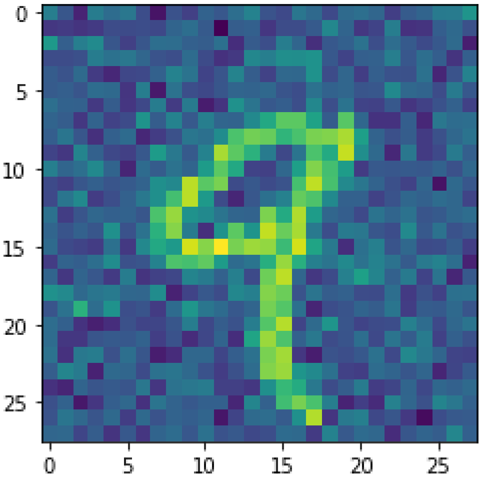
In [6]:
```
images = [-1, 0, 1, 2, 3, 4]
for i in images:
    plt.imshow(x_train_noise[i])
    plt.figure(i + 1)
```


0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25


0
5
10
15
20
25
0
5
10
15
20
25





```
<Figure size 432x288 with 0 Axes>
```

In [7]:
```
print(y_train[0])
print(y_train[1])
print(y_train[2])
print(y_train[3])
print(y_train[4])
```

```
5
0
4
1
9
```

2) Compare the `accuracy` of train and val after N epochs for MLNN with and without noise.

In [8]:
```
# Add noise, confirm the distortions are "mild to heavy" and without noise = 0.
x_train_noise = x_train + np.random.normal(scale = 0.7, size = x_train.shape)
print("Image 1 with noise:", x_train_noise[0, :5])
print("Image 1 without noise:", x_train[0, :5])
print("Image 2 with noise:", x_train_noise[1, :5])
print("Image 2 without noise:",x_train[1, :5])
print("Image 3 with noise:", x_train_noise[2, :5])
print("Image 3 without noise:", x_train[2, :5])
print("Image 4 with noise:", x_train_noise[3, :5])
print("Image 4 without noise:", x_train[3, :5])
print("Image 5 with noise:", x_train_noise[4, :5])
print("Image 5 without noise:", x_train[5, :5])
```

```
Image 1 with noise: [[ 0.38248305 -0.26618816  0.04660875  0.02019968  0.00283
725  1.08243348
  -0.19294936  0.24220481 -1.15797956 -0.7961585   0.92057683  0.62217734
  -1.17690773 -0.58672908 -0.62252674 -0.87304044 -0.75054664 -0.14183476
  -0.72143164 -0.04985499  1.5549931  -0.25140015 -0.13262035  0.63733253
   0.7648549   0.51082122 -0.39684768  0.2147539 ]
 [ 1.47230661 -0.21868641 -0.23775632  0.55347024  0.64144495 -0.72344572
   0.28860551  0.08588662  0.72119549  0.88335475 -0.6748685   1.37625366
  -1.01445929 -0.18940671 -0.44642431  0.21844896 -0.14258855 -0.0416632
  -0.25216904  0.52498239 -0.301516   -1.36956365  0.3974975  -0.33753768
  -0.06454998 -0.48257323 -0.43382264 -0.24656412]
 [ 0.10666124  0.03272304 -0.14629471  0.1945251  -0.31673246 -0.34488001
   0.18715389  0.43726696 -0.25859383  1.14035186  0.73620484  0.10559565
   0.97090719  0.24998227 -0.4451798   1.39637862  0.33714875 -0.25786243
  -0.01418439  0.47853183 -0.18459838 -0.68473961  1.14954562 -0.22209253
  -0.40731848  0.21723372  1.17082643  0.90525499]
 [-0.37569506 -0.12937971  0.14347893  0.5960726   0.22992756  0.57442808
  -0.0599187   0.06732069  0.36269539  0.63946839  0.56813622  0.14483011
   0.48680752  0.86414432  0.07373945  0.14948557  0.15938853 -0.06500146
   0.01457346  0.12538754 -0.36221252 -0.42130958  0.05044384 -0.13493664
   0.01875011 -0.96674867  0.84270945 -0.05046923]
 [-0.83943349 -0.47062168 -0.22496926  0.68298257 -0.33493059  0.03181197
  -0.67001932  0.24364084 -0.56514409  0.52928956 -0.50214644  0.1332547
   0.12534891  0.79244259 -1.2425055   0.81553012 -0.62082815 -0.54372344
   0.56412769  0.76614196 -0.77803416  0.91955598 -0.95693709  0.22284648
  -0.40366354 -0.89986696 -0.11498492  0.74006436]]
Image 1 without noise: [[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 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]]
Image 2 with noise: [[ 5.12426091e-01 -4.43466747e-01  8.51294262e-02  3.47016
448e-01
   3.72599722e-01  3.03474495e-01  4.76221384e-01 -1.33413806e-01
   2.53895636e-01  1.10631417e-01  9.17705329e-01  5.14566448e-02
  -2.70932256e-04  1.04232175e-01  3.86965739e-01  3.74064887e-01
  -4.41521746e-01 -9.48835811e-01 -9.82069054e-02  1.55601092e+00
  -1.28206642e-01  5.24854742e-01 -1.15611060e-01 -1.78119501e-01
   1.42193320e+00  8.00285837e-01  5.38879197e-02  1.78964634e+00]
 [ 7.03503366e-01  2.90057708e-01 -7.05135024e-01 -8.59249709e-01
  -9.63210932e-01 -9.35677690e-01  2.05512149e-01  7.29722479e-01
   1.02932067e+00  7.88491480e-01  6.15096918e-01 -5.40223184e-01
  -1.17858730e-01  1.02746030e+00  5.74813096e-02  4.30876580e-02
  -7.22333217e-01 -1.77843887e+00  5.17210900e-01 -5.43427355e-01
   5.35541040e-01 -5.05622531e-01  1.41224601e+00  7.57865652e-01
   3.88445876e-01  1.32542665e-01  9.61467029e-02 -4.25259971e-01]
 [-1.13798932e-01  5.03895849e-01 -2.46001251e-01 -3.82676424e-02
  -3.96387952e-01  1.42930789e-01  1.23867172e-01  9.70737999e-01
   1.18930928e-01  1.52498500e+00 -2.72093922e-01 -1.02784938e+00
  -3.42769707e-01  2.50454279e-01  1.04959974e+00  6.34079156e-01
   1.17456978e-01  3.11146607e-01  6.94218891e-01 -2.81477814e-01
  -5.60934130e-01  1.61785218e-02 -3.04722144e-01  1.50492625e-01
  -9.13827624e-01 -1.76742862e-02 -8.26941937e-01 -1.84808150e-01]
 [-5.93229876e-01 -2.11118418e-01 -1.98263191e-01 -5.30714351e-01
   8.77266847e-03 -1.48260136e-01  6.56929895e-01 -8.91410650e-01
   8.46121205e-01  9.25071164e-01  1.44106875e+00 -3.41265805e-02
  -1.14633290e+00  1.36719696e+00  3.69631051e-01 -2.13302260e-01
  -8.99885502e-01 -6.00590272e-01  6.18267095e-01  6.38900898e-01
  -5.08239024e-01 -2.69914384e-01 -1.99074391e-01  8.45794765e-01
  -1.21717134e-01  6.36544434e-01  1.68797738e-01 -1.64204995e+00]
 [-1.24872983e-01 -1.44323113e+00 -3.99298757e-02 -4.66247869e-01
  -2.92092517e-01 -3.04732379e-02 -1.26037344e-01 -3.47261210e-01
   5.19338609e-01 -8.45060652e-01  1.42716853e+00 -1.52178758e+00
   2.40644460e-01  5.70337801e-01 -1.57032921e+00  5.09937164e+01
   1.58914194e+02  2.51561222e+02  1.58784202e+02  4.83619032e+01
   8.25679240e-02  6.81131476e-01 -4.53326882e-01 -7.82424216e-01
  -4.39301272e-01  4.00097331e-03  4.37072084e-02 -3.19988318e-01]]
Image 2 without noise: [[  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   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  51 159 253
  159  50   0   0   0   0   0   0   0   0]]
Image 3 with noise: [[-0.30820529 -0.08676325  0.22433134  0.60927653  0.52295
842  0.64013516
   0.34583674 -1.43646068 -0.50100331 -0.97331328 -0.15058898 -0.56293527
  -0.00229614  0.50316325 -0.15817972 -1.1422455  -0.20240477  0.4616825
   0.16334309 -0.66669209  0.41849942 -0.14456132  1.42989065  0.03731867
   0.20762325  0.0311032   0.61735637 -0.19906409]
 [-0.0701715   0.76331108 -1.10243321 -0.63340188 -1.55592314  1.53995526
   0.17330933  0.72236674 -0.90376597  0.76046982  0.68412496  0.49600427
   1.07467654  0.11519968 -0.2006     -0.18308568  0.69169539 -0.25834325
  -0.13458547 -0.2173864   0.6664338   1.03573803 -0.3532459  -0.80378976
  -0.45993015 -0.7588067  -0.12735844 -0.98766236]
 [-0.71026509 -0.68860736  0.36190757  0.28530817 -0.17397271 -1.03074254
   0.63966756 -0.50295437  1.38204511  1.20931151  0.00906655  1.26869206
   0.03990913 -0.52808922  0.29701047 -0.50077227 -1.31700192  0.00312434
  -1.08622012  0.4333194  -0.41465099 -0.44134168 -0.23258014 -1.02939426
  -0.93571945  0.39667432 -0.38663144  0.05248347]
 [ 1.43902473  1.36077956 -0.92514115 -0.43011435  0.23068301  0.06195872
  -1.24274771  0.21265686  0.03177813 -0.53649241 -0.31367859 -0.60873176
   0.83005243 -0.16643757 -0.0569717   0.76806451 -0.08099571 -0.28441946
  -1.52166683  1.25781335 -1.55198579 -0.38893087  0.52276242  1.61772035
   1.26386088 -0.1379572  -0.13521288 -0.37947738]
 [ 0.94882294  0.75808412 -0.37279962 -0.3947643  -0.23734174 -1.15583191
   0.04282615 -0.00840809  0.74475258 -0.76698362  1.25647621 -1.85062675
  -0.01173101 -0.18336279 -0.04254168  0.56337315 -1.26943106  0.07384581
   0.23408127 -0.67729961  0.19554502  1.19076431 -0.29009475  0.08331865
   0.71683944  0.04400135  0.4814366  -0.49364109]]
Image 3 without noise: [[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 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]]
Image 4 with noise: [[ 0.13855389 -0.55581539 -0.31702563 -0.31205092  0.25953
278 -0.63435912
   0.71103544 -1.34091314  1.00831037 -0.47700754 -0.39351022 -0.48256628
   1.06371682  0.70686418 -0.0655369  -0.13345211 -0.92379658  0.09097473
   0.87312645 -1.66518971 -0.11103157  0.15667794 -0.45955882 -0.76273829
   0.54761521 -0.14810999  2.00567746  0.79511186]
 [ 0.27484679  0.39184572  0.12571926  0.49566721  1.26053565 -0.52129756
   0.42665546 -0.13919242  0.35002878 -0.45741811 -0.20995785  0.25538566
  -0.04654647 -0.62480216  0.716585   -1.11967559  0.26411588  0.39507185
  -0.11535788 -0.20023877  0.0031039  -0.98629569 -0.63185737  0.5725129
  -0.02448182 -0.34865639  0.39863305  0.79854469]
 [-0.1638594  -1.42648388  0.98812786 -0.7524803  -0.29518974 -0.8410821
   0.15245528  0.50400769 -0.17102217 -0.90994704 -0.41384026 -0.48254243
  -0.41391247  0.82221281 -0.11609179 -0.49004198 -1.1780761   0.65557475
  -0.65327394 -1.55029155 -0.35930447  0.08675392 -0.25159994  0.51243518
  -0.98420305 -0.16915077 -1.04547278  0.24129861]
 [-0.1295721  -0.40201517  1.04731324  0.71302612 -0.44450704  0.05685471
   0.70874502  0.21082547  0.64722043  0.8730733   0.43498157 -0.79511945
  -0.33542248  0.15057828 -0.69785293 -0.63534603 -0.44632237 -1.30468496
   0.20980344 -1.0910391  -0.43991423 -0.47573998 -0.56596472  0.59616275
  -0.33958476  0.1064719   0.34911164  0.8698763 ]
 [ 1.78790451  0.6079066  -0.64969356 -1.06315287  0.84551781 -0.675535
   0.2280227  -0.81257852  0.30535316 -0.6330847   0.94139423 -0.81417179
   0.58587657 -1.10147601  0.17757179 -0.46128476  1.06854446  0.33550634
   0.14952127  0.28873324 -0.4873436   0.20489323 -0.40784082 -0.16584226
   0.50396436 -0.61646552 -0.87038893  1.29201602]]
Image 4 without noise: [[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 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]]
Image 5 with noise: [[ 7.93793117e-01 -2.54731421e-01 -1.24612328e-01  7.12854
705e-01
   6.20513744e-01  7.00566766e-01 -5.98353407e-02  6.31078614e-02
   1.13588915e+00  6.57107202e-01  1.62491974e+00 -2.11609704e+00
  -5.73855729e-01 -8.50237268e-01 -1.36006911e-01  8.01614065e-01
  -3.41393266e-01 -3.13715808e-02 -7.28341904e-01  9.24551551e-01
  -5.41382119e-01 -3.94438882e-02 -8.13760597e-01 -1.66064424e+00
  -2.43324329e-01 -2.92128870e-01  1.10303938e+00 -7.23749721e-01]
 [-7.33425424e-01  5.60420639e-01 -7.24877326e-02 -2.61845700e-01
   3.63469107e-02 -1.20237466e+00  1.23128149e-01  1.25814162e+00
  -3.64308940e-02  1.75532860e-01  8.90992094e-02 -4.90346177e-01
  -1.21556910e-01 -5.38912917e-01  5.70896022e-01  2.68222643e-01
  -5.66786106e-01  4.18141603e-04 -1.20911012e-01  1.09233893e+00
   6.31862130e-01  4.34370466e-01 -6.73185537e-01 -1.61744674e-02
  -9.80837392e-02  2.79889949e-01  2.14447931e-01  6.59995143e-01]
 [-9.76286543e-02  1.48206966e-01 -1.32805141e+00 -4.54847777e-01
   5.50185332e-01 -9.48723597e-01 -3.23455755e-01 -4.83383049e-02
  -7.44839600e-01  1.64004752e-01 -1.86185761e-01  8.83992541e-01
  -9.02287973e-01 -3.88634157e-01 -3.03659538e-01 -3.66215193e-01
  -5.39610429e-01 -2.09940545e-01  1.29300887e+00  1.93725827e-01
  -5.63876644e-01 -1.38487070e+00  9.90704652e-01  3.13802049e-01
   1.13451755e+00  3.42897455e-01  7.87911398e-01 -7.83768313e-02]
 [-7.63588614e-01  5.62571712e-01 -6.46385084e-01 -1.22861425e+00
   1.89319593e-01 -6.97601892e-01  1.14679511e+00  1.38883349e+00
  -5.96222039e-01 -7.19409512e-01  2.32057654e-01 -1.13576915e-01
  -6.61235695e-01  8.10085406e-01 -7.89501642e-01 -1.88501325e+00
   7.86273202e-01 -6.11162323e-01  3.50793954e-01  4.21308866e-01
  -1.74918617e-01 -1.35309888e+00 -5.63477019e-01 -5.61105576e-01
   2.55171962e-01  5.40732508e-02  1.51669876e-01 -6.47548195e-01]
 [ 1.40680345e-01 -1.00353517e+00 -2.50727595e-01  4.65583302e-01
   6.34586361e-01 -1.05780602e+00 -6.25264101e-01 -1.17836324e+00
  -5.20722262e-01  9.10030334e-01 -1.69796780e-01  6.60193088e-02
   2.62438337e-01  4.03475372e-01 -1.24981380e+00  2.58077116e-01
   4.94308199e-02  7.85553448e-02  1.03809285e+00  2.57362830e-01
  -5.46315733e-01 -4.73721767e-01 -8.12851491e-01  6.93913404e-01
  -3.30333130e-01 -4.55685054e-01  5.44717140e-01  9.11216753e-01]]
Image 5 without noise: [[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 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]]
```

In [9]:

```
# Reshape
(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
```

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.

In [10]:

```
scales = [.1, .5, 1.0, 2.0, 4.0]
train_acc = []
test_acc = []

batch_size = 128
num_classes = 10
epochs = 20

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

for scale in scales:
    x_train_noise = x_train + np.random.normal(scale = scale, size = x_train.sh
    x_test_noise = x_test + np.random.normal(scale = scale, size = x_test.shape

    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_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)
    train_acc.append(history.history['accuracy'][-1])
    test_acc.append(score[1])
```

```
Metal device set to: Apple M1 Pro
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 dense (Dense)               (None, 512)               401920    
                                                                 
 dropout (Dropout)           (None, 512)               0         
                                                                 
 dense_1 (Dense)             (None, 512)               262656    
                                                                 
 dropout_1 (Dropout)         (None, 512)               0         
                                                                 
 dense_2 (Dense)             (None, 10)                5130      
                                                                 
=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
_________________________________________________________________
Epoch 1/20
```

```
2022-04-09 16:28:34.877125: I tensorflow/core/common_runtime/pluggable_device/
pluggable_device_factory.cc:305] Could not identify NUMA node of platform GPU
ID 0, defaulting to 0. Your kernel may not have been built with NUMA support.
2022-04-09 16:28:34.877223: I tensorflow/core/common_runtime/pluggable_device/
pluggable_device_factory.cc:271] Created TensorFlow device (/job:localhost/rep
lica:0/task:0/device:GPU:0 with 0 MB memory) -> physical PluggableDevice (devi
ce: 0, name: METAL, pci bus id: <undefined>)
2022-04-09 16:28:35.024176: W tensorflow/core/platform/profile_utils/cpu_util
s.cc:128] Failed to get CPU frequency: 0 Hz
```

```
  1/469 [..............................] - ETA: 3:09 - loss: 2.3798 - accurac
y: 0.1484
```

```
2022-04-09 16:28:35.285051: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

```
468/469 [============================>.] - ETA: 0s - loss: 0.2639 - accuracy:
0.9186
```

```
2022-04-09 16:28:39.767945: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

```
469/469 [==============================] - 5s 10ms/step - loss: 0.2636 - accur
acy: 0.9187 - val_loss: 0.1242 - val_accuracy: 0.9606
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1001 - accur
acy: 0.9693 - val_loss: 0.0995 - val_accuracy: 0.9698
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0601 - accur
acy: 0.9815 - val_loss: 0.1012 - val_accuracy: 0.9731
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0432 - accur
acy: 0.9866 - val_loss: 0.1146 - val_accuracy: 0.9703
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0298 - accur
acy: 0.9903 - val_loss: 0.1287 - val_accuracy: 0.9719
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0244 - accur
acy: 0.9920 - val_loss: 0.1278 - val_accuracy: 0.9747
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0181 - accur
acy: 0.9942 - val_loss: 0.1767 - val_accuracy: 0.9694
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0189 - accur
acy: 0.9944 - val_loss: 0.1406 - val_accuracy: 0.9756
Epoch 9/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0144 - accur
acy: 0.9956 - val_loss: 0.1496 - val_accuracy: 0.9759
Epoch 10/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0130 - accur
acy: 0.9959 - val_loss: 0.1668 - val_accuracy: 0.9740
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0129 - accur
acy: 0.9959 - val_loss: 0.1681 - val_accuracy: 0.9753
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0116 - accur
acy: 0.9967 - val_loss: 0.1686 - val_accuracy: 0.9776
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0108 - accur
acy: 0.9968 - val_loss: 0.1914 - val_accuracy: 0.9750
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0105 - accur
acy: 0.9972 - val_loss: 0.2127 - val_accuracy: 0.9754
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0103 - accur
acy: 0.9972 - val_loss: 0.2013 - val_accuracy: 0.9786
Epoch 16/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0106 - accur
acy: 0.9971 - val_loss: 0.2081 - val_accuracy: 0.9764
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0090 - accur
acy: 0.9976 - val_loss: 0.2468 - val_accuracy: 0.9741
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0090 - accur
acy: 0.9977 - val_loss: 0.2167 - val_accuracy: 0.9772
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0084 - accur
acy: 0.9976 - val_loss: 0.2408 - val_accuracy: 0.9756
Epoch 20/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0064 - accur
acy: 0.9982 - val_loss: 0.2494 - val_accuracy: 0.9771
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      
                                                                 
=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
_________________________________________________________________
Epoch 1/20
  7/469 [..............................] - ETA: 4s - loss: 2.0079 - accuracy:
0.3002
```

```
2022-04-09 16:30:10.932032: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

```
469/469 [==============================] - ETA: 0s - loss: 0.5429 - accuracy:
0.8233
```

```
2022-04-09 16:30:15.327191: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

```
469/469 [==============================] - 5s 10ms/step - loss: 0.5429 - accur
acy: 0.8233 - val_loss: 0.3267 - val_accuracy: 0.8969
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.2389 - accur
acy: 0.9233 - val_loss: 0.2679 - val_accuracy: 0.9144
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1222 - accur
acy: 0.9603 - val_loss: 0.2996 - val_accuracy: 0.9160
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0693 - accur
acy: 0.9769 - val_loss: 0.3562 - val_accuracy: 0.9169
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0488 - accur
acy: 0.9837 - val_loss: 0.3948 - val_accuracy: 0.9169
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0364 - accur
acy: 0.9875 - val_loss: 0.4615 - val_accuracy: 0.9147
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0333 - accur
acy: 0.9895 - val_loss: 0.4894 - val_accuracy: 0.9181
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0301 - accur
acy: 0.9900 - val_loss: 0.5298 - val_accuracy: 0.9164
Epoch 9/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0247 - accur
acy: 0.9918 - val_loss: 0.5562 - val_accuracy: 0.9199
Epoch 10/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0266 - accur
acy: 0.9923 - val_loss: 0.5780 - val_accuracy: 0.9197
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0224 - accur
acy: 0.9932 - val_loss: 0.6674 - val_accuracy: 0.9148
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0217 - accur
acy: 0.9936 - val_loss: 0.6759 - val_accuracy: 0.9196
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0215 - accur
acy: 0.9936 - val_loss: 0.6976 - val_accuracy: 0.9164
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0202 - accur
acy: 0.9938 - val_loss: 0.7141 - val_accuracy: 0.9178
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0189 - accur
acy: 0.9942 - val_loss: 0.7511 - val_accuracy: 0.9194
Epoch 16/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0204 - accur
acy: 0.9944 - val_loss: 0.7691 - val_accuracy: 0.9195
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0185 - accur
acy: 0.9947 - val_loss: 0.7990 - val_accuracy: 0.9179
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0172 - accur
acy: 0.9951 - val_loss: 0.7901 - val_accuracy: 0.9227
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0168 - accur
acy: 0.9953 - val_loss: 0.8156 - val_accuracy: 0.9207
Epoch 20/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0177 - accur
acy: 0.9949 - val_loss: 0.8196 - val_accuracy: 0.9227
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      
                                                                 
=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
_________________________________________________________________
Epoch 1/20
  6/469 [..............................] - ETA: 4s - loss: 2.6800 - accuracy:
0.1615
```

```
2022-04-09 16:31:47.003491: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

```
468/469 [============================>.] - ETA: 0s - loss: 1.0758 - accuracy:
0.6348
```

```
2022-04-09 16:31:51.370412: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

```
469/469 [==============================] - 5s 10ms/step - loss: 1.0750 - accur
acy: 0.6351 - val_loss: 0.7961 - val_accuracy: 0.7342
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.6531 - accur
acy: 0.7798 - val_loss: 0.7241 - val_accuracy: 0.7615
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 0.4403 - accur
acy: 0.8521 - val_loss: 0.7801 - val_accuracy: 0.7562
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.2899 - accur
acy: 0.9007 - val_loss: 0.8869 - val_accuracy: 0.7614
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1984 - accur
acy: 0.9319 - val_loss: 1.0222 - val_accuracy: 0.7520
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1540 - accur
acy: 0.9477 - val_loss: 1.1511 - val_accuracy: 0.7570
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1365 - accur
acy: 0.9534 - val_loss: 1.2285 - val_accuracy: 0.7539
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1183 - accur
acy: 0.9604 - val_loss: 1.3839 - val_accuracy: 0.7503
Epoch 9/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1064 - accur
acy: 0.9639 - val_loss: 1.4590 - val_accuracy: 0.7532
Epoch 10/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1000 - accur
acy: 0.9680 - val_loss: 1.4895 - val_accuracy: 0.7570
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0946 - accur
acy: 0.9696 - val_loss: 1.5861 - val_accuracy: 0.7547
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0895 - accur
acy: 0.9716 - val_loss: 1.6674 - val_accuracy: 0.7564
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0842 - accur
acy: 0.9734 - val_loss: 1.7161 - val_accuracy: 0.7582
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0801 - accur
acy: 0.9750 - val_loss: 1.8114 - val_accuracy: 0.7579
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0780 - accur
acy: 0.9762 - val_loss: 1.9046 - val_accuracy: 0.7601
Epoch 16/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0819 - accur
acy: 0.9758 - val_loss: 1.8849 - val_accuracy: 0.7566
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0699 - accur
acy: 0.9783 - val_loss: 2.0562 - val_accuracy: 0.7552
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0743 - accur
acy: 0.9773 - val_loss: 2.0543 - val_accuracy: 0.7600
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0718 - accur
acy: 0.9789 - val_loss: 2.1010 - val_accuracy: 0.7594
Epoch 20/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0708 - accur
acy: 0.9792 - val_loss: 2.1432 - val_accuracy: 0.7579
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
  6/469 [..............................] - ETA: 4s - loss: 3.9505 - accuracy:
0.1198
```

```
2022-04-09 16:33:23.581108: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

```
468/469 [============================>.] - ETA: 0s - loss: 1.8635 - accuracy:
0.3590
```

```
2022-04-09 16:33:27.992604: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

```
469/469 [==============================] - 5s 10ms/step - loss: 1.8632 - accur
acy: 0.3591 - val_loss: 1.5765 - val_accuracy: 0.4569
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 1.4564 - accur
acy: 0.4988 - val_loss: 1.5714 - val_accuracy: 0.4550
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 1.2543 - accur
acy: 0.5680 - val_loss: 1.6014 - val_accuracy: 0.4527
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 1.0614 - accur
acy: 0.6331 - val_loss: 1.6739 - val_accuracy: 0.4450
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.8799 - accur
acy: 0.6963 - val_loss: 1.8049 - val_accuracy: 0.4389
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.7494 - accur
acy: 0.7405 - val_loss: 1.9696 - val_accuracy: 0.4289
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.6427 - accur
acy: 0.7746 - val_loss: 2.1054 - val_accuracy: 0.4189
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.5687 - accur
acy: 0.8036 - val_loss: 2.2510 - val_accuracy: 0.4206
Epoch 9/20
469/469 [==============================] - 5s 10ms/step - loss: 0.5093 - accur
acy: 0.8250 - val_loss: 2.3922 - val_accuracy: 0.4178
Epoch 10/20
469/469 [==============================] - 5s 10ms/step - loss: 0.4665 - accur
acy: 0.8397 - val_loss: 2.4919 - val_accuracy: 0.4178
Epoch 11/20
469/469 [==============================] - 5s 10ms/step - loss: 0.4392 - accur
acy: 0.8504 - val_loss: 2.6283 - val_accuracy: 0.4224
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 0.4050 - accur
acy: 0.8624 - val_loss: 2.7597 - val_accuracy: 0.4189
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3966 - accur
acy: 0.8668 - val_loss: 2.8320 - val_accuracy: 0.4128
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3715 - accur
acy: 0.8762 - val_loss: 2.9321 - val_accuracy: 0.4193
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3556 - accur
acy: 0.8825 - val_loss: 3.0714 - val_accuracy: 0.4147
Epoch 16/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3395 - accur
acy: 0.8889 - val_loss: 3.2275 - val_accuracy: 0.4102
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3306 - accur
acy: 0.8921 - val_loss: 3.2569 - val_accuracy: 0.4158
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3232 - accur
acy: 0.8967 - val_loss: 3.2644 - val_accuracy: 0.4157
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3133 - accur
acy: 0.9001 - val_loss: 3.4029 - val_accuracy: 0.4106
Epoch 20/20
469/469 [==============================] - 5s 10ms/step - loss: 0.3070 - accur
acy: 0.9034 - val_loss: 3.5665 - val_accuracy: 0.4127
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
  6/469 [..............................] - ETA: 4s - loss: 7.4754 - accuracy: 
0.0951
```

```
2022-04-09 16:35:01.787564: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

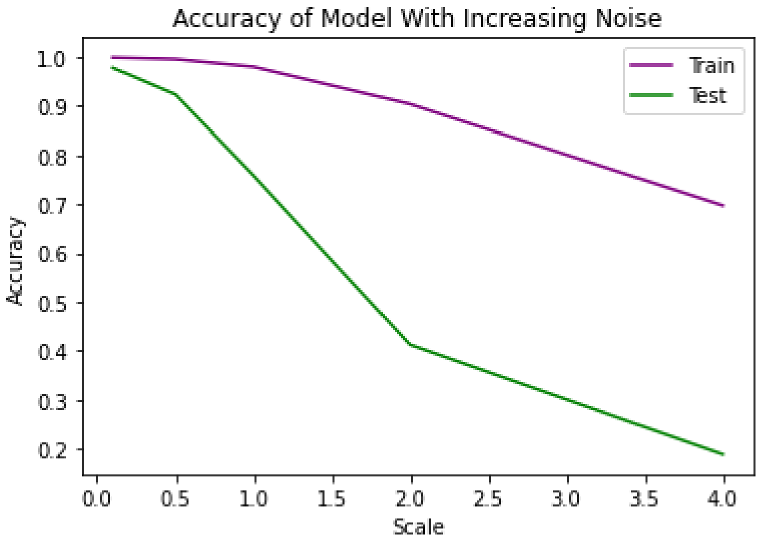
```
467/469 [============================>.] - ETA: 0s - loss: 2.3981 - accuracy: 
0.1583
```

```
2022-04-09 16:35:06.261999: I tensorflow/core/grappler/optimizers/custom_graph
_optimizer_registry.cc:113] Plugin optimizer for device_type GPU is enabled.
```

```
469/469 [==============================] - 5s 10ms/step - loss: 2.3974 - accur
acy: 0.1585 - val_loss: 2.1581 - val_accuracy: 0.2123
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 2.0790 - accur
acy: 0.2546 - val_loss: 2.1296 - val_accuracy: 0.2276
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 1.9425 - accur
acy: 0.3121 - val_loss: 2.1475 - val_accuracy: 0.2301
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 1.8133 - accur
acy: 0.3582 - val_loss: 2.1881 - val_accuracy: 0.2221
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 1.6906 - accur
acy: 0.4047 - val_loss: 2.2456 - val_accuracy: 0.2199
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 1.5795 - accur
acy: 0.4459 - val_loss: 2.3179 - val_accuracy: 0.2120
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 1.4755 - accur
acy: 0.4817 - val_loss: 2.3824 - val_accuracy: 0.2097
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 1.3904 - accur
acy: 0.5158 - val_loss: 2.4535 - val_accuracy: 0.2039
Epoch 9/20
469/469 [==============================] - 5s 11ms/step - loss: 1.3223 - accur
acy: 0.5394 - val_loss: 2.5286 - val_accuracy: 0.2009
Epoch 10/20
469/469 [==============================] - 5s 10ms/step - loss: 1.2533 - accur
acy: 0.5661 - val_loss: 2.6404 - val_accuracy: 0.1996
Epoch 11/20
469/469 [==============================] - 5s 11ms/step - loss: 1.2001 - accur
acy: 0.5863 - val_loss: 2.6576 - val_accuracy: 0.1971
Epoch 12/20
469/469 [==============================] - 5s 10ms/step - loss: 1.1602 - accur
acy: 0.6000 - val_loss: 2.7331 - val_accuracy: 0.1956
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 1.1148 - accur
acy: 0.6181 - val_loss: 2.8115 - val_accuracy: 0.1963
Epoch 14/20
469/469 [==============================] - 5s 10ms/step - loss: 1.0787 - accur
acy: 0.6303 - val_loss: 2.8985 - val_accuracy: 0.1926
Epoch 15/20
469/469 [==============================] - 5s 10ms/step - loss: 1.0409 - accur
acy: 0.6476 - val_loss: 2.9607 - val_accuracy: 0.1969
Epoch 16/20
469/469 [==============================] - 5s 10ms/step - loss: 1.0164 - accur
acy: 0.6578 - val_loss: 3.0516 - val_accuracy: 0.1912
Epoch 17/20
469/469 [==============================] - 5s 10ms/step - loss: 0.9847 - accur
acy: 0.6681 - val_loss: 3.0511 - val_accuracy: 0.1893
Epoch 18/20
469/469 [==============================] - 5s 10ms/step - loss: 0.9628 - accur
acy: 0.6784 - val_loss: 3.1893 - val_accuracy: 0.1942
Epoch 19/20
469/469 [==============================] - 5s 10ms/step - loss: 0.9401 - accur
acy: 0.6867 - val_loss: 3.2666 - val_accuracy: 0.1951
Epoch 20/20
469/469 [==============================] - 5s 10ms/step - loss: 0.9158 - accur
acy: 0.6972 - val_loss: 3.2986 - val_accuracy: 0.1895
```

In [11]:

```
plt.figure()
plt.plot(scales, train_acc, label = 'Train', c = "purple")
plt.plot(scales, test_acc, label = 'Test', c = "green")
plt.xlabel('Scale')
plt.ylabel('Accuracy')
plt.title('Accuracy of Model With Increasing Noise')
plt.legend()
plt.show()
```



As evidenced above the accuracy of the model decreases steadily as the noise increases, which makes sense.

In [ ]: